Nios II Embedded Design Suite Release Notes

© May 2013 RN-EDS-11.0

About These Release Notes

These release notes cover versions 12.0 through 13.0 of the Altera® Nios® II Embedded Design Suite (EDS). These release notes describe the revision history for the Nios II EDS.

For the most recent list of errata for the Nios® II EDS, refer to the Knowledge Base page of the Altera website. Also, you can use the Knowledge Base to search for errata based on the product version affected and other criteria.

Product Revision History

Table 1 shows the revision history for the Nios II EDS.

Table 1. Nios II Embedded Design Suite Revision History

Version	Date	Description
13.0	May 2013	Verified with the Quartus II software v13.0
12.1	December 2012	■ Verified with the Quartus II software v12.1 ■ Nios II C-to-Hardware (C2H) Acceleration Compiler discontinued
12.0	June 2012	Verified with the Quartus II software v12.0

For more information about Nios II EDS features, refer to the *Nios II Processor Reference Handbook* and the *Nios II Software Developer's Handbook*.

Nios II IDE and Nios II C2H Compiler End of Life

The Quartus II software no longer includes the Nios II IDE or the Nios II C-to-Hardware (C2H) Acceleration Compiler. These products are discontinued. The Nios II Software Build Tools (SBT), including the Nios II SBT for Eclipse, is the recommended flow for all Nios II software development.

You can download past versions of the Nios II IDE and the Nios II C2H Compiler legacy tools from the Altera Download Center. Altera will support Nios II C2H Compiler licensees for up to one year after v12.1. To obtain support, you must have a current license for the Nios II C2H Compiler v12.0 or v11.1.

Nios II Embedded Design Suite Release Notes

© December 2012 RN-EDS-10.1

About These Release Notes

These release notes cover versions 11.1 through 12.1 of the Altera® Nios® II Embedded Design Suite (EDS). These release notes describe the revision history for the Nios II EDS.

For the most recent list of errata for the Nios® II EDS, refer to the Knowledge Base page of the Altera website. Also, you can use the Knowledge Base to search for errata based on the product version affected and other criteria.

Product Revision History

Table 1 shows the revision history for the Nios II EDS.

Table 1. Nios II Embedded Design Suite Revision History

Version	Date	Description
12.1	December 2012	■ Verified with the Quartus II software v12.1 ■ Nios II C-to-Hardware (C2H) Acceleration Compiler discontinued
12.0	June 2012	Verified with the Quartus II software v12.0
11.1	November 2011	Verified with the Quartus II software v11.0

For more information about Nios II EDS features, refer to the *Nios II Processor Reference Handbook* and the *Nios II Software Developer's Handbook*.

Nios II IDE and Nios II C2H Compiler End of Life

The Quartus II software no longer includes the Nios II IDE or the Nios II C-to-Hardware (C2H) Acceleration Compiler. These products are discontinued. The Nios II Software Build Tools (SBT), including the Nios II SBT for Eclipse, is the recommended flow for all Nios II software development.

You can download past versions of the Nios II IDE and the Nios II C2H Compiler legacy tools from the Altera Download Center. Altera will support Nios II C2H Compiler licensees for up to one year after v12.1. To obtain support, you must have a current license for the Nios II C2H Compiler v12.0 or v11.1.

Notice about Environment Variables

The `QUARTUS_ROOTDIR` and `SOPC_KIT_NIOS2` environment variables will be deprecated in a future release. If you have custom scripts that use these variables, you should be prepared to change them to be independent of these variables, or to set the environment variables yourself.

Nios II Embedded Design Suite Release Notes and Errata

© July 2012 RN-EDS-10.0

About These Release Notes

These release notes cover versions 11.0 through 12.0 of the Altera® Nios® II Embedded Design Suite (EDS). These release notes describe the revision history and errata for the Nios II EDS.

Errata are functional defects or errors, which might cause the product to deviate from published specifications. Documentation issues include errors, unclear descriptions, or omissions from current published specifications or product documents.

For the most recent list of errata for the Nios® II EDS, refer to the Knowledge Base page of the Altera website. Also, you can use the Knowledge Base to search for errata based on the product version affected and other criteria.

Product Revision History

Table 1 shows the revision history for the Nios II EDS, including the Nios II C2H Compiler.

Table 1. Nios II Embedded Design Suite Revision History

Version	Date	Description
12.0	June 2012	Maintenance release
11.1	November 2011	Maintenance release
11.0	May 2011	■ The Nios II EDS by default supports the GCC 4.1.2 version of the GNU Compiler Collection (GCC). GCC 3.4.6, the Nios II IDE, and Nios II C2H Compiler are optional features. They are available only if you enable **Legacy Package: Nios II IDE / GCC3 Toolchain / C2H Compiler** when you install the Altera Complete Design Suite. ■ Projects created with version 10.1 and earlier of the Nios II SBT need to be re-imported to create the Nios II launch configuration correctly.

For more information about Nios II EDS features, refer to the *Nios II Processor Reference Handbook*, the *Nios II Software Developer's Handbook*, and the *Nios II C2H Compiler User Guide*.

Notice about Environment Variables

The `QUARTUS_ROOTDIR` and `SOPC_KIT_NIOS2` environment variables will be deprecated in a future release. If you have custom scripts that use these variables, you should be prepared to change them to be independent of these variables, or to set the environment variables yourself.

Errata

Table 2.

Problem	Published Date	Version Found In	Version Fixed In	Knowledge Base Link
Launch SOPC Builder Button in C2H View	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr205439.html
Method for Accessing MMU and MPU Registers in the Debugger	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr271017.html
Error Message when Downloading .elf File	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr274039.html
Unaligned Transfers of Small Payloads Fail on SG-DMA	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr273638.html
Java Heap Space Exception if Quartus II Compilation is Enabled	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr241217.html
Must Enable Support for Nios II ModelSim Flow in GUI for DDR2 and DDR3 SDRAM Controller with UniPHY and RLDRAM II Controller with UniPHY	23-Nov-11	Version 11.0	Version 11.1	http://www.altera.com/support/kdb/solutions/spr371218.html
Nios II SBT for Eclipse Unable to Create New Application and BSP from Template	14-Sep-11	Version 10.1		http://www.altera.com/support/kdb/solutions/spr361380.html
NicheStack TCP/IP Stack - Nios II Edition Does Not Support TFTP	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr324002.html
Incorrect Results From Logical or Conditional Operation With Side-Effects	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr201796.html
Quartus II Compilation Warnings for Nios II Stratix II 2S60 ROHS Example	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr356312.html
Pre-7.1 Systems Are Not Supported	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr252367.html
BSP Not Updated for Memory Size Changes in SOPC Builder	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr332637.html
SOPC Builder Not Supported for DDR SDRAM Controller with ALTMEMPHY	1-Dec-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr377815.html
Incorrect Stack and Heap Free Space Report	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr212712.html
Memory Window Sets Control Register Values Incorrectly	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr237683.html
GNU Assembler Does Not Accept the --defsym Flag	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr220772.html
Cannot Locate Source Code in Driver Files Shared by Multiple Projects	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr238891.html
Nios II IDE Reports Problems Without Displaying Error in Console	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr225102.html
Trace Debug Does Not Support the JMPI Instruction	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr253697.html
Non-System-Wide Reset Can Cause Improper Initialization of Mailbox Core	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr237093.html

Table 2.

Problem	Published Date	Version Found In	Version Fixed In	Knowledge Base Link
AN543 Contains Incorrect Information about Updating the Flash	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr344660.html
ISS Fails on Designs Containing Triple Speed Ethernet MAC or SG-DMA Components	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr239173.html
nios2-console Does Not Work in the GCC 4 Command Shell	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr352235.html
Linker Errors with Dual-Port Memories	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr239208.html
IDE Cannot Display Imported Profiling Data	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr271045.html
Intermittent Failures While Accessing CompactFlash Card	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr378928.html
Nios II IDE shows “Source not found” during debugging	22-Sep-11	Version 11.0		http://www.altera.com/support/kdb/solutions/spr374173.html
C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr387111.html
SBT Fails if Nios II EDS is Installed in a Path Containing Spaces	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr253132.html
Variable Casting Unsupported in ISS	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr267696.html
--src-dir SBT Argument Does Not Work With C2H	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr249030.html
Nios II IDE Help Content Missing on Linux Platforms	1-Dec-11	Version 11.0		http://www.altera.com/support/kdb/solutions/spr385755.html
BSP Editor Does Not Show Command-Line Help	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr336931.html
Compiler Flags for Building Custom Newlib	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr302227.html
Incorrect IRQ Information in system.h	22-May-12	Version 10.1	Version 11.1 SP1	http://www.altera.com/support/kdb/solutions/spr375970.html
Compilation Error with Separate Exception Stack Option	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr226411.html
Nios II IDE Online Help Expand Buttons Do Not Work	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr252392.html
Nios II EDS Requires Administrative Permissions on Windows Vista and Windows 7	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr341734.html
Nios II IDE Cannot Find stdio.h in Outline View	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr288467.html
JTAG UART Interactive Window for Simulation Does Not Work	14-Sep-11	Version 11.0		http://www.altera.com/support/kdb/solutions/spr374124.html
The C2H Compiler Regenerates an Accelerator Unnecessarily	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr241781.html

Table 2.

Problem	Published Date	Version Found In	Version Fixed In	Knowledge Base Link
Nios II Application Properties Settings Not Saved	22-May-12	Version 12.0	Version 11.1 SP2	http://www.altera.com/support/kdb/solutions/fb31565.html
Build Errors on Software for Pre-Existing Design with SG-DMA	13-Sep-11	Version 11.0		http://www.altera.com/support/kdb/solutions/spr370447.html
Pointer Dereferences to Volatile Types	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr235077.html
Watchpoints Do Not Work on Certain Variables	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr178267.html
Stop on Startup Option in Run Configuration Has No Effect	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr320455.html
System Timestamp for System ID Peripheral Is Always 0	22-Sep-11	Version 11.0		http://www.altera.com/support/kdb/solutions/spr373315.html
Valid Range of hal.log_flags is -1 to 3	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr330972.html
Programs That Interact With a Terminal Console on Windows Do Not Work	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr226096.html
The Restart Command on the Run Menu Does Not Work	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr223992.html
Nios II IDE Hangs With UNC Project Path	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr269656.html
C2H Compiler Does Not Work in the Nios II Command Shell with GCC 4	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr362413.html
The Restart Button in the Debugger Does Not Work	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr349888.html
Functions Declared Without a Return Type Are Not Supported	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr274111.html
Array Elements in Structures Do Not Copy Correctly	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr205992.html
Nios II Launch Configurations Not Visible after Service Pack Installation	24-Aug-11	Version 11.0		http://www.altera.com/support/kdb/solutions/spr364701.html
Missing .dat File Error Message: “Failed to open VHDL file”	14-Sep-11	Version 11.0		http://www.altera.com/support/kdb/solutions/spr374122.html
Missing Design Examples Mentioned in Nios II Software Developer's Handbook	1-Dec-11	Version 11.0		http://www.altera.com/support/kdb/solutions/spr381144.html
Cannot Launch Nios II SBT for Eclipse on Red Hat Linux	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr324012.html
Error: c2h_accelerator_base_addresses.h: No such file or directory	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr241685.html
Console Window Is Not Updated After ISS Error	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr239064.html
Networking Examples	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr387664.html
Windows Vista: Limited Support in Nios II EDS	22-Sep-11	Version 10.0	Version 10.0	http://www.altera.com/support/kdb/solutions/spr257471.html

Table 2.

Problem	Published Date	Version Found In	Version Fixed In	Knowledge Base Link
Error Marker Persists on BSP Project After Build Error Corrected	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr317261.html
stdio Does Not Work with MicroC/OS-II and Small C Library	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr225241.html
Hardware Accelerators Remain After Deleting the Software Project	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr189600.html
elf2flash File Size Limit	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr220774.html
Nios II IDE Freezes While Displaying the Splash Screen	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr227063.html
Errors Converting Nios II IDE Multiprocessor Project	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr325340.html
Build Command Not Functional for BSPs Created With the Nios II SBT	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr240221.html
C++ Exceptions Unsupported by the Nios II GCC 4 Toolchain	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr345499.html
User-Managed BSP Settings Not Supported in Nios II IDE	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr238889.html
Windows/Cygwin: Nios II Processor Generation Failure	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr255473.html
Internal Error When Double-Clicking on a Large Objdump File	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr227686.html
DMA Controller Always Busy in Burst Mode	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr240514.html
C++ Exceptions Unsupported in a Multi-Threaded Environment	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr232407.html
Error Messages for UniPHY External Memory Interfaces in ModelSim Flow for Eclipse	18-Nov-11	Version 11.0		http://www.altera.com/support/kdb/solutions/spr375252.html
Trace Debug Does Not Support Instruction-Related Exceptions	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr269276.html
Accelerator Generation Failure If Tools Are Installed in Path With Spaces	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr239007.html
C/C++ Scanner Does Not Support Certain C/C++ Constructs	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr174004.html
Clean Build Causes Build Failure	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr205910.html
EIC-Based Nios II System Crashes With NicheStack	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr324305.html
“Run as ModelSim” in the Nios II IDE Fails	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr241280.html
Build Failure with Nios II Advanced Exceptions, MMU, and MPU	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr273521.html
Run as Nios II Hardware Fails	1-Dec-11	Version 11.0		http://www.altera.com/support/kdb/solutions/spr373493.html

Table 2.

Problem	Published Date	Version Found In	Version Fixed In	Knowledge Base Link
Error Message After Renaming Project: “Resource is out of sync with the system”	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr325603.html
Error Running Nios II Project: ‘Downloading ELF Process failed’	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr344863.html
Warning Message: “'pragma_reverse_bitfields' attribute directive ignored”	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr357595.html
cout From MicroC/OS-II Task Does Not Send Data to stdout	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr178104.html
Error “UNC paths are not supported” Launching ModelSim	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr227904.html
Error Messages on Console When Debugging	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr319862.html
‘Exclude from build’ Not Supported	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr387097.html
Error Building Imported Project: ‘target pattern contains no %’	13-Sep-11	Version 10.0	Version 10.0	http://www.altera.com/support/kdb/solutions/spr341302.html
Nios II IDE Command-Line Tools Hang on Windows	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr254931.html
nios2-download Does Not Recognize CPU name in Qsys Subsystems	30-Nov-11	Version 11.0		http://www.altera.com/support/kdb/solutions/spr376945.html
Incorrect information about Embedded C++	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr361482.html
memcpy() Optimization Misalignment	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr304041.html
Uninitialized .bss Variables in Simulation	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr145474.html
Missing Nios II Perspective	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr343803.html
Nios II IDE Command-Line Tools Select Wrong Workspace on Linux	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr258212.html
URL Project Location Causes Project Creation Error	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr240598.html
Build Errors on Interrupt API Calls	22-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr320691.html
Error Building Imported Project: 'No rule to make target'	13-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr341689.html
Changing Build Configurations Produces Unexpected Results	14-Sep-11	Version 10.0		http://www.altera.com/support/kdb/solutions/spr205422.html

Nios II Embedded Design Suite Release Notes and Errata

© December 2011 RN-EDS-9.1

About These Release Notes

These release notes cover versions 10.1 through 11.1 of the Altera® Nios® II Embedded Design Suite (EDS). These release notes describe the revision history and errata for the Nios II EDS.

Errata are functional defects or errors, which might cause the product to deviate from published specifications. Documentation issues include errors, unclear descriptions, or omissions from current published specifications or product documents.

For the most recent list of errata for the Nios® II EDS, refer to the Knowledge Base page of the Altera website. Also, you can use the Knowledge Base to search for errata based on the product version affected and other criteria.

Product Revision History

Table 1 shows the revision history for the Nios II EDS, including the Nios II C2H Compiler.

Table 1. Nios II Embedded Design Suite Revision History

Version	Date	Description
11.1	November 2011	Maintenance release
11.0	May 2011	■ The Nios II EDS by default supports the GCC 4.1.2 version of the GNU Compiler Collection (GCC). GCC 3.4.6, the Nios II IDE, and Nios II C2H Compiler are optional features. They are available only if you enable **Legacy Package: Nios II IDE / GCC3 Toolchain / C2H Compiler** when you install the Altera Complete Design Suite. ■ Projects created with version 10.1 and earlier of the Nios II SBT need to be re-imported to create the Nios II launch configuration correctly.
10.1	December 2010	Maintenance release

For more information about Nios II EDS features, refer to the *Nios II Processor Reference Handbook*, the *Nios II Software Developer's Handbook*, and the *Nios II C2H Compiler User Guide*.

Notice about Environment Variables

The `QUARTUS_ROOTDIR` and `SOPC_KIT_NIOS2` environment variables will be deprecated in a future release. If you have custom scripts that use these variables, you should be prepared to change them to be independent of these variables, or to set the environment variables yourself.

Errata

Nios II IDE Help Content Missing on Linux Platforms

Description

You might be unable to view Nios II IDE Help content when running on newer versions of the Linux operating system. The Nios II IDE requires an outdated version of SWT library **libswt-mozilla-gtk-3235.so**, which is not included in newer versions of Linux.

Workaround

Altera recommends migrating your projects to the Nios II Software Build Tools for Eclipse.

Missing Design Examples Mentioned in Nios II Software Developer's Handbook

Description

The *Nios II Software Developer's Handbook* refers to design files that are no longer bundled with the current Nios II EDS installation. The Overview of Nios II Embedded Development and Nios II Software Build Tools Reference chapters refer to design files that are no longer part of the installation.

Workaround

Obtain the design examples directly from the Design Examples page on the Altera website.

SOPC Builder Not Supported for DDR SDRAM Controller with ALTMEMPHY

Description

SOPC Builder does not support the DDR SDRAM Controller with ALTMEMPHY in version 10.1. The DDR2 SDRAM Controller with ALTMEMPHY is supported.

Workaround

No workaround is available at this time. Consider migrating your system to Qsys.

nios2-download Does Not Recognize CPU name in Qsys Subsystems

Description

You cannot download **.elf** files to Nios II processors in Qsys subsystems if all the processors have the same name.

Workaround

Rename the Nios II processors in the Qsys subsystems with unique names, such as **cpu_one** and **cpu_two**.

Run as Nios II Hardware Fails

Description

Sometimes you cannot run a design as Nios II Hardware.

Workaround

Select **Run Configurations** from the Run menu, point to **Apply**, and click **Run**.

Uninitialized .bss Variables in Simulation

Description

If your program reads the value of an uninitialized `.bss` variable during HDL simulation, and the BSP (system library) is compiled with the **ModelSim only, no hardware support** property enabled in Nios II IDE, a warning appears about unfiltered data being `'x'`. This warning appears because when this property is enabled, the code that clears the `.bss` memory region is omitted to speed up HDL simulation so this memory region is uninitialized. The `.bss` region contains global and static local variables that are not initialized by the application so they default to a value of zero. When the Nios II processor reads uninitialized variables, it displays a warning and converts any of the bits of the uninitialized data to zero which correctly mimics the effect of the missing `.bss` clearing code. The Hardware Abstraction Layer (HAL) code that executes before and after `main()` might use `.bss` variables, so these warnings might appear even if your application does not use the `.bss` section.

Workaround

No workaround is available.

C/C++ Scanner Does Not Support Certain C/C++ Constructs

Description

The C/C++ scanner performs C/C++ Search, navigation, open declaration, and parts of content assist. Due to limitations of the C/C++ scanner, these features do not work with C code constructs not supported by the C++ language. An example is functions that take a function pointer as an argument.

Workaround

If the C/C++ Search fails, use the File Search facility.

cout From MicroC/OS-II Task Does Not Send Data to stdout

Description

If neither `printf()` or `cout` is used from `main()` before tasks are started, `cout` does not work from a task.

Workaround

Add the following C++ code to the beginning of `main()`:

```
std::ios_base::sync_with_stdio(false);
```

Watchpoints Do Not Work on Certain Variables

Description

Watchpoints do not work on a variable whose size is not 32 bits.

Workaround

Change the types of global and static local variables to `int`, `long`, or `unsigned long` before setting watchpoints on them.

Hardware Accelerators Remain After Deleting the Software Project

Description

If a system contains C2H accelerators, deleting the software project that defines the accelerators does not remove the accelerators from the hardware system, and the accelerator logic remains in the SOPC Builder system.

Workaround

To remove an accelerator from a system, delete the accelerator from the C2H view in the Nios II IDE first, and then recompile the software project. The C2H Compiler then removes the accelerator from the SOPC Builder system. Once the compilation is complete then the software application can be deleted from the workspace.

Incorrect Results From Logical or Conditional Operation With Side-Effects

Description

The C2H Compiler always evaluates both operands of logical (&&, ||) and conditional (?:) operators. This is different from expected American National Standards Institute (ANSI) C behavior, where operands are evaluated left-to-right, and unnecessary operands are skipped.

For example, in the expression (i-- && j--), if the value of i is zero, ANSI C does not evaluate the right-hand-side (RHS) expression, and j is not decremented. By contrast, the C2H Compiler evaluates both sides, decrementing j.

The following expressions are other examples that might be affected by this issue:

```
if (i-- || j++)
    ...;


a = ((cond == 1)?  i++ : j++);
```

Workaround

Use logical and conditional operations whose operators have no side effects. Operations with side effects include pre- and post- increment and decrement operations (++, --), memory operations (*, [], ., ->), and function calls.

Changing Build Configurations Produces Unexpected Results

Description

The C2H Compiler does not support multiple build configurations (for example Release or Debug) in the Nios II IDE. After creating one or more accelerators in a particular configuration, the C2H Compiler produces undefined results if you switch to a different build configurations and create more accelerators.

Workaround

For a specific SOPC Builder system and Nios II IDE project, specify C2H accelerators in only one build configuration. You can use multiple build configurations, as long as only one configuration specifies C2H Compiler settings.

Launch SOPC Builder Button in C2H View

Description

When the Nios II IDE workspace contains multiple projects with multiple system libraries, the incorrect SOPC Builder system might open when you click **Launch SOPC Builder** in the C2H view.

Workaround

Launch SOPC Builder from the Quartus II software. Alternatively, keep only one system library project open at a time while using the C2H Compiler.

Clean Build Causes Build Failure

Description

Performing a clean build on a Nios II IDE project that contains a hardware accelerator can cause the next build to fail in the IDE, because the clean build erroneously deletes a file required by the C2H Compiler.

Workaround

Do not perform a clean build on projects that use hardware accelerators. If you have already performed a clean build, recompile with option **Build software, generate SOPC Builder system, and run Quartus II compilation** to regenerate the necessary files.

Array Elements in Structures Do Not Copy Correctly

Description

C2H accelerators do not correctly copy array elements that are elements of structures.

In "Array Elements of Structs", the a and b elements of the structure copy correctly, but the buf element does not. After this assignment, struct_a equals {9, 8, {3, 3, 3, 3}}.

Example 1. Array Elements of Structs

```
typedef struct my_struct {
  int a;
  int b;
  int buf[BUF_SIZE];
}MY_STRUCT;
MY_STRUCT struct_a = {1, 2, {3, 3, 3, 3}};
MY_STRUCT struct_b = {9, 8, {7, 7, 7, 7}};
struct_a = struct_b;
```

Workaround

Copy the array elements explicitly, as shown in "Copying Array Elements Explicitly".

Example 2. Copying Array Elements Explicitly

```
{
  int i=0;
  do
    {
      struct_a.buf[i] = struct_b.buf[i];
      i++;
    } while (i<LENGTH_OF_BUF_ELEMENT)
}
```

Incorrect Stack and Heap Free Space Report

Description

The makefile reports an incorrect number of bytes free for the stack and heap, if the heap and stack are in different memory regions.

Workaround

No workaround is available.

GNU Assembler Does Not Accept the --defsym Flag

Description

According to the GNU documentation, you can set an assembler definition by using the `--defsym` flag, but it does not work in the following form: `--defsym MY_VAR=1`.

Workaround

There is no workaround.

elf2flash File Size Limit

Description

The **elf2flash** utility supports **.elf** files up to approximately 24 MBytes in size. The **elf2flash** utility might fail on files larger than 24 MBytes, with the error message `java.lang.OutOfMemoryError`.

Workaround

Lower the number of symbols in your **.elf** file by turning off debug symbols.

Alternatively, specify less initialized data in the application.

The Restart Command on the Run Menu Does Not Work

Description

The **Restart** command on the **Run** menu does not work.

Workaround

Stop the program, then debug it again. If the debugger is hung in an endless loop, use the following **bash** alias to break the target, then stop it:

```
alias break="kill -2 \'ps -a | grep nios2-elf-gdb | cut -f6 -d' '\'"
```

Nios II IDE Reports Problems Without Displaying Error in Console

Description

When building a project, the Nios II IDE reports problems, but the build output in the console does not contain any errors.

The Nios II IDE incorrectly reports some linker warnings as errors, with a dialog box saying **Errors exist in a required project**. The Dhrystone software example exhibits this behavior, and recompiling the project again makes the issue go away.

Workaround

If the Console output does not contain errors, then the project built correctly. On subsequent builds, the linker step is skipped and the errors do not appear.

stdio Does Not Work with MicroC/OS-II and Small C Library

Description

`stdin`, `stdout`, and `stderr` do not work in MicroC/OS-II applications built with the Small C library option.

Workaround

Disable the small C library option.

Programs That Interact With a Terminal Console on Windows Do Not Work

Description

Programs with this behavior work in version 6.0 and earlier, but do not work in Nios II IDE version 6.1 and later.

The Eclipse platform in version 6.1 and later of the IDE (on Windows only) sends the string `\r\n` instead of just `\n` when running or debugging using the Terminal. This behavior can break existing software designs, and it is inconsistent with `nios2-terminal`, which still just sends `\n`.

Workaround

Change the software running on the Nios II processor to parse for `\r\n` as well as `\n`.

Compilation Error with Separate Exception Stack Option

Description

Choosing the **Use a separate exception stack option** might cause the following compilation error when building a project:

```
VARIABLE %STACK_POINTER%"
```

This error occurs if the exception stack is larger than the memory available for it.

Workaround

On the system library properties page for the project, turn off the separate exception stack or reduce the **Maximum exception stack size** setting.

Nios II IDE Freezes While Displaying the Splash Screen

Description

After clicking **Switch Workspace** on the File menu on a Windows machine, a Nios II IDE splash screen appears. Unfortunately, this splash screen obscures the dialog box that asks you to specify the new workspace. As a result, the IDE appears to freeze.

Workaround

Press Alt-Tab to switch applications. Two relevant application icons appear: an Eclipse icon associated with the splash screen and a Nios II IDE icon associated with the **Workspace** dialog box. Select the Nios II icon to bring the dialog box to the foreground.

Internal Error When Double-Clicking on a Large Objdump File

Description

On Windows when opening a large **objdump** file in the Nios II IDE, you might get the following error message:

```
Unable to create this part due to an internal error.
Reason for the failure: Editor could not be initialized.
```

Workaround

Adjust the Windows launch arguments for the Nios II IDE editor. Perform the following steps:

1. On the Windows Start menu, browse to the **Nios II EDS** program icon, right-click it, then click **Properties**. The **Windows Properties** dialog box appears.
2. In the **Target** field, append `"vmargs -Xmx1024m"` to the end of the path to the Nios II IDE executable. For example:

   ```
   C:\altera\72\nios2eds\bin\eclipse\nios2-ide.exe -vmargs -Xmx1024m
   ```

Error “UNC paths are not supported” Launching ModelSim

Description

If you launch the ModelSim®-Altera simulation software from a working directory that is mapped via a universal naming convention (UNC) path (a path that starts with `//` instead of drive letter), you receive the following error message in SOPC Builder: **UNC paths are not supported. Defaulting to Windows directory**. This error occurs because ModelSim is calling a command shell, which does not support UNC paths.

Workaround

Map the UNC path to a drive letter and use the drive letter to reference the working directory in the launching shell.

C++ Exceptions Unsupported in a Multi-Threaded Environment

Description

C++ exceptions are only supported in a single-threaded environment. They cannot be used in a multi-threaded environment such as MicroC/OS-II.

Workaround

In a multi-threaded environment, you need to guard C++ exceptions with semaphores.

Pointer Dereferences to Volatile Types

Description

The C2H Compiler treats pointer dereferences to a volatile type as if they alias all other pointer dereferences. Pointers that are restrict-qualified are treated the same way.

The two loops in Example 1 cannot be scheduled concurrently because the `volatile` qualification overrides the `__restrict__` pragma.

Example 1. Non-Concurrent Loops

```
volatile int * __restrict__ fifo_rd = FIFO_RD_BASE;
volatile int * __restrict__ fifo_wr = FIFO_WR_BASE;
for ()
{
  *fifo_wr = ....;
}
for ()
{
  ... = *fifo_rd;
}
```

Workaround

Divide the function into multiple interrupt request (IRQ)-enabled accelerators that are launched concurrently from the processor, and use FIFO buffers to communicate between them.

Non-System-Wide Reset Can Cause Improper Initialization of Mailbox Core

Description

The altera_avalon_mailbox peripheral might not be initialized properly when a soft (non-system-wide) reset occurs. In this condition, mailbox contents (read and write pointers) are not reinitialized and might show potentially stale data.

Workaround

Ensure that a system-wide reset event occurs by asserting the reset_n input to the SOPC Builder system containing the mailbox. This resets all peripherals and Nios II processors in the system.

Memory Window Sets Control Register Values Incorrectly

Description

The memory window might incorrectly set values in memory-mapped control registers. For example, writing 0x1234 to a byte addressed register results in the value 0x3434 being stored in the register. The memory window shows this incorrect value.

Workaround

Use the GNU debugger (GDB) console window in the IDE, instead of the memory window, to write to the registers. For example, type the following command:

```
set {int} <register address>=0x1234↵
```

You must refresh the memory window for it to correctly display the target value.

User-Managed BSP Settings Not Supported in Nios II IDE

Description

For projects created with the Nios II SBT and imported to the Nios II IDE, the IDE configuration settings have no effect.

For example, objdump, compiler, and linker settings made in the IDE are ignored. This behavior occurs because Nios II SBT projects are not IDE-managed projects. In addition, the make-related preferences do not pertain to imported Nios II SBT projects. The IDE ignores these options during the build process.

Workaround

Make these settings in the project's makefile.

Cannot Locate Source Code in Driver Files Shared by Multiple Projects

Description

If you hit a breakpoint in a driver file, and that driver file is shared with another project that is closed, the Nios II IDE might indicate that it cannot locate the source code.

Workaround

Open the closed system library project and resume debugging.

Accelerator Generation Failure If Tools Are Installed in Path With Spaces

Description

If the path to your installation of the Altera Design Suite contains spaces, the C2H Compiler fails to generate the accelerators.

Workaround

Reinstall the tools to a path containing no spaces.

Console Window Is Not Updated After ISS Error

Description

After performing a **Run as ISS**, if you receive an ISS error in the console window, the console is not updated subsequently.

Workaround

Close the console window after receiving an ISS error. A new console window opens when a new message is available.

ISS Fails on Designs Containing Triple Speed Ethernet MAC or SG-DMA Components

Description

You receive an Internal Error when attempting to perform an ISS simulation of designs containing the Altera Triple Speed Ethernet media access control (MAC) or SG-DMA components because the Nios II ISS does not support these components.

Workaround

Remove the Triple Speed Ethernet MAC and SG-DMA components from your system and perform ISS simulation on the simplified system. You can also simulate the design in ModelSim or test it on hardware.

Before removing the Triple Speed Ethernet MAC and SG-DMA components, make a copy of the unmodified system to ensure that you can return to the original configuration.

Linker Errors with Dual-Port Memories

Description

If your instruction master and data master ports are connected to the same dual-port memory and the ports have different addresses, your code fails to run or you experience a linker error. The Nios II IDE does not warn you of the addressing violation.

Workaround

Assign the same address to both ports of the dual-port memory.

Build Command Not Functional for BSPs Created With the Nios II SBT

Description

The build option in the Nios II IDE menu does not rebuild BSPs imported to the IDE.

Workaround

The Nios II SBT for Eclipse can both build and debug projects created on the command line. The Nios II SBT for Eclipse is the preferred tool for debugging Nios II SBT projects.

For information about the Nios II SBT for Eclipse, refer to the Getting Started with the Graphical User Interface chapter of the *Nios II Software Developer's Handbook*.

In the Nios II IDE, you can build the BSP by building the associated application project.

DMA Controller Always Busy in Burst Mode

Description

The DMA controller component (`altera_avalon_dma`), when enabled for burst transactions, does not perform transfers at widths other than its full data width. The DMA controller is always busy.

Workaround

When bursting is enabled, the DMA controller must be programmed to perform transactions at its full data width.

URL Project Location Causes Project Creation Error

Description

When you try to create a new project in an existing workspace, you might see an error dialog box saying:

```
Project cannot be created. Reason: Internal Error
```

This error might occur if the path to any project in the workspace is a URL location, for example **file:/F:/Design**. To view the path, right-click on the project and select **Properties**.

Workaround

Import your existing application and system library projects to a new workspace.

Java Heap Space Exception if Quartus II Compilation is Enabled

Description

If your design contains a C2H accelerator, and you select **Build software, generate SOPC Builder system, and run Quartus II compilation** in the C2H view, you might see the following error during Quartus II compilation:

```
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
make: *** [c2h_hdl-t] Error 1
```

Workaround

Select the Build software and generate SOPC builder system option in the C2H Compiler settings window, and then manually launch the Quartus II software to compile the design.

“Run as ModelSim” in the Nios II IDE Fails

Description

The **Run as ModelSim** command might fail on launch configurations created in version 7.0 or earlier of the IDE. This problem does not occur for new launch configurations.

Workaround

Select a location for the ModelSim tool from the launch configuration dialog box. You can use the **Browse** button next to the **ModelSim path** group, or type in a path to the ModelSim directory (for example **c:/altera/71/modelsim_ae/win32aloem**).

Error: c2h_accelerator_base_addresses.h: No such file or directory

Description

When a C2H accelerator is compiled for the first time, the following compile-time error can result if the **Analyze all accelerators** option is selected:

```
c2h_accelerator_base_addresses.h: No such file or directory.
```

Workaround

Click **Build software and generate SOPC Builder system** and build once before building with the **Analyze all accelerators** option.

The C2H Compiler Regenerates an Accelerator Unnecessarily

Description

The C2H Compiler might regenerate an accelerator even when the accelerated function is unchanged. This problem can result from changes to files included by the C file containing the accelerated function. The C2H Compiler fails to check that the generated hardware description language (HDL) matches the previously generated HDL causing the system to be regenerated.

Workaround

To avoid this issue move the accelerator-specific information from the include file to a separate include file. This Workaroundprevents regeneration of the system when the HDL is unchanged.

--src-dir SBT Argument Does Not Work With C2H

Description

When using the C2H Compiler with Nios II SBT, an error occurs when using the `nios2-app-generate-makefile` argument `--src-dir` to specify a directory of source files. An example of the error is:

```
multiple definition of 'my_accelerated_function'
```

Workaround

Specify source files individually using the `--src-files` argument.

Pre-7.1 Systems Are Not Supported

Description

Starting in version 7.1, the C2H Compiler does not support SOPC Builder systems created in SOPC Builder version 7.0 or earlier (systems based on a **class.ptf** file).

Workaround

Before using the C2H Compiler, you must upgrade your SOPC Builder system. Open the system in SOPC Builder version 7.1 or later. SOPC Builder prompts you to convert the system, creating a ***_hw.tcl** file compatible with the C2H Compiler.

Nios II IDE Online Help Expand Buttons Do Not Work

Description

The expand text arrow and the **Show All** option in the online help do not work in Internet Explorer.

Workaround

In Internet Explorer, carry out the following actions:

1 Click the Refresh/Show Current Topic icon (yellow arrows icon at the top of the search pane). The table of contents appears with the selected topic highlighted.

2 Click the highlighted topic to refresh the browser frame. Expanded text works.

Alternatively, use Firefox 2.0.0.6.

SBT Fails if Nios II EDS is Installed in a Path Containing Spaces

Description

The Nios II Command Line SBT fails if you install the Nios II EDS in a path containing spaces.

Workaround

Reinstall the Nios II EDS to a path that does not contain spaces.

Trace Debug Does Not Support the JMPI Instruction

Description

The JMPI instruction is not supported by the Nios II trace tools.

Workaround

There is no workaround.

Nios II IDE Command-Line Tools Hang on Windows

Description

On Windows systems, the Nios II IDE command-line tools are sometimes unable to locate the project workspace. When this happens, the tools might hang.

This issue potentially affects the following tools:

- `nios2-create-system-library`
- `nios2-create-application-project`
- `nios2-build-project`
- `nios2-import-project`
- `nios2-delete-project`

Workaround

Explicitly supply the Nios II IDE command-line tools with a workspace location, by means of the `-data` command-line argument. The syntax of the `-data` argument is as follows:

`-data` *<path to workspace>*

The path to the workspace must be absolute, and must not contain whitespace. Aside from these restrictions, any valid file system path can be used for the workspace. The workspace shown in Example 1 is the default workspace used by the Nios II IDE in version 9.0.

Example 1. Providing a Workspace Location

```
nios2-create-project -data c:/altera/90/nios2eds/\
    bin/eclipse/nios2-ide-workspace-9.0 <other arguments> ↵
```

Windows/Cygwin: Nios II Processor Generation Failure

Description

You might see the following error message when you generate your Nios II system in SOPC Builder:

```
Error: Generator program for module 'cpu_fpoint' did NOT run
successfully
```

This error might occur on a Windows-based system when all of the following conditions are true:

- You have installed a version of Cygwin other than the one distributed with the Quartus II software.
- You launch the Quartus II software and SOPC Builder from a Nios II Command Shell.
- You enable the floating point custom instruction (FPCI) in the Nios II processor core.

Workaround

Launch the Quartus II software and SOPC Builder from the Windows Start menu, and regenerate your system.

Windows Vista: Limited Support in Nios II EDS

Description

The Quartus II software introduced Windows Vista (32-bit and 64-bit) support in version 7.2. However, several ensuing releases of the Nios II Embedded Design Suite support only the following platforms:

- Windows 2000
- Windows XP
- Windows XP (64)
- Windows Vista Business Edition (32-bit)
- SUSE 9 (32-bit)
- SUSE 9 (64-bit)
- Red Hat Linux version 3.0 (32-bit)
- Red Hat Linux version 3.0 (64-bit)
- Red Hat Linux version 4.0 (32-bit)
- Red Hat Linux version 4.0 (64-bit)

Workaround

Upgrade to a more recent version of the Nios II EDS.

To see the current operating system support for the Altera Complete Design Suite tools, refer to the Operating System Support page of the Altera website.

Nios II IDE Command-Line Tools Select Wrong Workspace on Linux

Description

On Linux systems, the Nios II IDE command-line tools use the default Eclipse project workspace, regardless what workspace you are using in the IDE. The default workspace is at `$SOPC_KIT_NIOS2/eclipse/nios2-ide-workspace-<version>`.

This issue affects the following tools:

- `nios2-create-system-library`
- `nios2-create-application-project`
- `nios2-build-project`
- `nios2-import-project`
- `nios2-delete-project`

Workaround

Explicitly supply the Nios II IDE command-line tools with a workspace location, by means of the `-data` command-line argument. The syntax of the `-data` argument is as follows:

`-data` <*path to workspace*>

The path to the workspace must be absolute, and must not contain whitespace. Aside from these restrictions, any valid file system path can be used for the workspace. Example 1 shows how to specify a workspace path.

Example 1. Providing a Non-Default Workspace Location in Linux

```
nios2-create-project -data $HOME/myworkspace<other arguments>↵
```

Variable Casting Unsupported in ISS

Description

Casting variables in the Variables view in the Debug perspective when using the Nios II ISS might cause an exception dialog box to open.

This exception might occur if you select **Debug as ISS** in the Nios II IDE and try to cast variables via the Variables view.

Workaround

There is no workaround.

Trace Debug Does Not Support Instruction-Related Exceptions

Description

The instruction-related exception handler is not supported by the Nios II trace tools.

Workaround

There is no workaround.

Nios II IDE Hangs With UNC Project Path

Description

In the **New Project** dialog box, if you turn on **Specify Location** and specify the path in UNC form, the IDE might hang.

Workaround

Map the UNC path to a remote drive which looks like a Windows drive to the Nios II IDE.

Method for Accessing MMU and MPU Registers in the Debugger

Description

You might observe an error if you try to read or write an MMU or MPU register via the Nios II Debugger.

Workaround

To read an MPU region, execute the following steps:

1 Set region `INDEX` in the `MPUBASE` register.

2 Exit and reenter debug mode, that is, single step.

3 Set the `RD` bit in the `MPUACC` register.

4 Exit and reenter debug mode, that is, single step.

5 Read back `MPUBASE` for pertinent information.

6 Read back `MPUACC` for pertinent information.

For a system with an MMU or MPU, this Workaroundallows you to read and write the current values of the registers. However, you cannot use it to control MPU regions or MMU TLB entries.

The debug core copies the Nios II processor’s register values to its internal memory when the processor enters debug mode. The debug core writes register values back to the processor only when the processor leaves debug mode. Therefore, if you attempt to set an MPU region with several consecutive values, only the last one, when leaving debug mode, is committed to the processor.

IDE Cannot Display Imported Profiling Data

Description

If you create a software project in the Nios II SBT command line flow, generate profiling data to **gmon.out** in the Nios II command shell, and then import the project to the IDE, you cannot use the Profiling perspective to view the profiling data. When you attempt to view **gmon.out**, the IDE displays the following error message:

```
nios2-elf-gprof: ../: not in a.out format.
```

Workaround

Run `nios2-elf-gprof` from the Nios II command shell to generate a profiler report.

Build Failure with Nios II Advanced Exceptions, MMU, and MPU

Description

Projects created in the Nios II IDE for an SOPC Builder system containing a Nios II processor configured with advanced exceptions, the memory management unit (MMU), or the memory protection unit (MPU) fail to build, and generate the following error messages:

```
ERROR - Classic build flow for Nios II system library does not support
the Nios II advanced exceptions.
Use the non-classic Nios II Board Support Package instead.
<timestamp> - (SEVERE) generate: java.lang.IllegalStateException:
java.lang.IllegalStateException:
com.altera.ingenuous.GTFElement.GTFElementErrorException: <error>
element in GTF script
make[1]: *** [system_description/../obj/system.h-t] Error 1
make: *** [system_project] Error 2
```

Workaround

Create the software project with the Nios II SBT command-line development flow. Alternatively, use the Nios II SBT for Eclipse.

Unaligned Transfers of Small Payloads Fail on SG-DMA

Description

The Scatter Gather DMA SOPC Builder peripheral does not correctly handle unaligned transfers with small payloads. A payload length smaller than the data width causes erroneous data transfers.

Workaround

Avoid using DMA devices to transfer small payloads.

If absolutely necessary, for a 32-bit SG-DMA, a minimum length of 4 bytes guarantees that data is transferred correctly.

Error Message when Downloading .elf File

Description

If the **Build Automatically** option is turned on in the IDE, you might get the following error when downloading a **.elf**:

```
"2 [main] sh 5736 fork: child 3892 - died waiting; \
    for longjmp before initialization, errno 11"
```

The **Build Automatically** option is off by default, as recommended by the Eclipse help system.

Workaround

Turn off the **Build Automatically** option and download the **.elf** file again.

Functions Declared Without a Return Type Are Not Supported

Description

The C2H compiler does not support functions without an explicitly declared return type.

Workaround

If you are using the implicit `int` return type, declare the return type explicitly. If your function has no return value, declare it as `void`.

Nios II IDE Cannot Find stdio.h in Outline View

Description

If you create a new project using the Nios II IDE and try to open the **stdio.h** file from the Outline view before building the project, the IDE displays the error message **No include files were found that matched that name**.

Workaround

Build the project before attempting to open **stdio.h**.

Compiler Flags for Building Custom Newlib

Description

"Common BSP Tasks" in the Nios II Software Build Tools chapter of the *Nios II Software Developer's Handbook* describes how to compile a custom version of the newlib library. However, it does not list the compiler flags that must be selected to generate the correct library.

Workaround

The missing information is as follows:

The Nios II EDS provides a number of precompiled newlib implementations. The provided libraries are precompiled with every viable combination of the GCC compiler flags shown in Table 1.

When you create a BSP with a precompiled newlib, the Nios II SBT selects the newlib matching your BSP's compiler settings. When you create a custom newlib, you must ensure that the compiler flags listed in Table 1 match your BSP's settings.

Table 1. GCC Compiler Options for Newlib

Option Name	Purpose
`-pg`	Link for profiling with gprof
`-EB`	Generate big-endian code
`-mstack-check`	Enable stack checking
`-mno-hw-mul`	Disable use of the `mul` family of instructions
`-mhw-mulx`	Enable use of the `mulx` family of instructions
`-mcustom-fpu-cfg`	Use a floating point custom instruction

The Nios II-specific compiler flag `-mcustom-fpu-cfg` requires the Nios II processor core to be implemented with the floating-point custom instruction as specified in the flag argument. The compiler flag can have one of the following two argument values:

- `-mcustom-fpu-cfg=60-1`--Use the Nios II floating-point custom instruction without divider
- `-mcustom-fpu-cfg=60-2`--Use the Nios II floating-point custom instruction with divider

For details about the Nios II-specific compiler flags `-mstack-check`, `-mno-hw-mul`, and `-mhw-mulx`, refer to "Altera Nios II Options" in *Using the GNU Compiler Collection (GCC)*, installed with the Nios II EDS. For details about the `-pg` compiler flag, refer to "Compiling a Program for Profiling" in *GNU Profiler*, installed with the Nios II EDS. To find the installed documentation, on the Windows Start menu, click **Programs** > **Altera** > **Nios II EDS** <*version*> > **Nios II** <*version*> **Documentation**, and then click **Literature**.

memcpy() Optimization Misalignment

Description

When optimization is turned on (`-O1` or higher), if you use `memcpy()` and the source pointer is aligned to a 32-bit boundary, the compiler implements `memcpy()` with word-oriented instructions as part of the optimization process. This optimization technique causes unexpected results in your software if `memcpy()` is used on a misaligned address.

Workaround

Take steps to ensure that an optimized implementation of `memcpy()` is called only with aligned data pointers.

Error Marker Persists on BSP Project After Build Error Corrected

Description

In the Nios II SBT for Eclipse, if an application project fails to build, error markers appear in the BSP project, if the build error message contains a path to BSP project source files. Even after you have rebuilt the application project without errors, the error markers persist in the BSP project.

Workaround

Rebuild your BSP project.

Alternatively, if your application will always be associated with the same BSP, set the Eclipse project references by following these steps:

1. Right-click your application project, point to **Properties** and click **Project References**.
2. Select the desired BSP for your application.
3. Build your application project again to clear the error markers in the BSP project.

If you later link your application project with a different BSP, you must manually correct the Eclipse project references.

Error Messages on Console When Debugging

Description

When debugging, you might see the following messages in the Nios II SBT for Eclipse console:

```
No symbol "auto" in current context.
No symbol "new" in current context.
```

These are normal messages from the GNU Debugger (GDB) console, not error messages. This issue does not impact the debug process.

Workaround

Disregard the messages.

Stop on Startup Option in Run Configuration Has No Effect

Description

Turning on **Stop on Startup at** in the **Debugger** tab of a run configuration has no effect.

Workaround

You can manually set any breakpoint in the debug perspective.

Build Errors on Interrupt API Calls

Description

You might observe build errors if your application project installs an ISR that calls the `alt_irq_register()` function.

The Nios II EDS version 9.1 includes support for an optional External Interrupt Controller (EIC) interface on the Nios II processor. To support the EIC interface, the HAL includes an enhanced interrupt API.

When you create a BSP, the SBT determines which interrupt API(s) the device drivers in your system support. If all registered drivers in the BSP identify themselves as supporting the enhanced API, the SBT implements the enhanced API. All Altera device drivers in version 9.1 and later support the enhanced API.

When the enhanced API is implemented, the legacy interrupt API, including `alt_irq_register()`, is not available. Therefore, if the application code contains a call to a legacy API function, a linker error occurs.

This issue might affect application projects that call `alt_irq_register()` or other legacy interrupt API functions directly, rather than depending on drivers in the BSP for interrupt support.

If your Nios II application project's source code registers an ISR using `alt_irq_register()`, or includes an ISR, your application might fail to compile in the Nios II EDS version 9.1, due to undefined references to `alt_irq_register()`, or to a change to the ISR function prototype.

Workaround

Modify your application code to use the new enhanced interrupt API. To use the enhanced API, you must modify several function calls and all ISR function prototypes.

For information about supporting the enhanced interrupt API, refer to the Exception Handling chapter of the *Nios II Software Developer's Handbook*, or to AN 595: Vectored Interrupt Controller Usage and Applications.

You must upgrade to the enhanced API if you wish to use an EIC in your hardware design to accelerate interrupt response time. The enhanced API also works with the Nios II processor's internal interrupt controller.

NicheStack TCP/IP Stack - Nios II Edition Does Not Support TFTP

Description

The Nios II documentation should include the following information:

Versions of the NicheStack TCP/IP Stack other than the Nios II Edition include optional Trivial File Transfer Protocol (TFTP) client and server applications. However, the TFTP client and server are not thread safe. They cannot be used in systems with the MicroC/OS-II RTOS.

TFTP is not available with the NicheStack TCP/IP Stack - Nios II Edition. The TFTP client and server features are disabled.

Cannot Launch Nios II SBT for Eclipse on Red Hat Linux

Description

You might be unable to launch Nios II SBT for Eclipse on the Red Hat Linux operating system. This is an issue with the version of XULRunner in Eclipse 3.4. The issue is resolved in XULRunner 1.9.1, which is available with Eclipse 3.5.

Workaround

Upgrade XULRunner to a newer version. To determine what version of XULRunner you have, type the following command at the command prompt:

```
xulrunner -vr
```

To upgrade XULRunner using the **yum** software package manager, type the following command:

```
sudo yum update xulrunnerr
```

EIC-Based Nios II System Crashes With NicheStack

Description

When you attempt to run software based on the NicheStack TCP/IP Stack - Nios II Edition on a hardware design incorporating an external interrupt controller such as the VIC, the Nios II system crashes during initialization, with unpredictable results.

Workaround

There is no workaround.

Errors Converting Nios II IDE Multiprocessor Project

Description

If you try use the **nios2-convert-ide2sbt** utility to convert an IDE multiprocessor project to the Nios II SBT, error messages appear, and the project fails to generate properly.

Workaround

Create new Nios II SBT application and BSP projects using project settings equivalent to the original Nios II IDE project settings.

Error Message After Renaming Project: “Resource is out of sync with the system”

Description

The Getting Started with the Graphical User Interface chapter of the *Nios II Software Developer’s Handbook* does not explain the correct procedure for renaming a project.

Workaround

The missing information is as follows:

To rename a project in the Nios II SBT for Eclipse, execute the following steps:

1 Right click the BSP project and click **Rename**.

2 Type the new BSP name.

3 Right click the BSP project and click **Refresh**.

If you neglect to refresh the project, you might see the following error message when you attempt to build it:

```
Resource <original_bsp_name> is out of sync with the system
```

Valid Range of hal.log_flags is -1 to 3

Description

The Nios II EDS documentation incorrectly states the valid range for the `hal.log_flags` BSP setting.

Workaround

Valid values of `hal.log_flags` range from -1 through 3.

BSP Not Updated for Memory Size Changes in SOPC Builder

Description

If you change the size of a memory in SOPC Builder, the memory region size in any previously created BSP is no longer correct. Regenerating the BSP does not update the BSP's memory region size.

Workaround

In the BSP Editor, on the **Linker Script** tab, you can correct the memory region size either of the following ways:

- Edit the memory region size manually.
- Click **Restore Defaults** to rerun the default Tcl script. The default Tcl script reads the updated memory region size from the SOPC information file (**.sopcinfo**) and updates the BSP.

For detailed information about keeping your BSP consistent with changes in the underlying SOPC Builder system, refer to "Revising Your BSP" in the Nios II Software Build Tools chapter of the *Nios II Software Developer's Handbook*.

BSP Editor Does Not Show Command-Line Help

Description

The following commands in the Nios II Command Shell should display command-line help:

- `nios2-bsp-editor --help`
- `nios2-bsp-editor --extended-help`

Instead, these commands launch the BSP Editor.

Workaround

For help with the BSP Editor, refer to "Using the BSP Editor" in the Getting Started with the Graphical User Interface chapter of the *Nios II Software Developer's Handbook*.

Error Building Imported Project: ‘target pattern contains no %’

Description

If your application or library makefile uses an absolute path and is generated with the GCC 3 toolchain, and you import it to the version 10.0 Nios II SBT for Eclipse using the GCC 4 toolchain, you get the following build error:

```
makefile:587: *** target pattern contains no '%'. Stop.
```

Workaround

Recreate the project using the SBT for Eclipse version 10.0.

Alternatively, use the GCC 3 toolchain.

Error Building Imported Project: 'No rule to make target'

Description

If an application or library project is created with version 9.1 SP2 or earlier (using the GCC 3 toolchain), and you import it to the version 10.0 Nios II SBT for Eclipse using the GCC 4 toolchain, you might see the build error shown in Example 1.

Example 1. Build Error in GCC 4

```
make all
Info: Building ../hw_bsp
make --no-print-directory -C ../hw_bsp
make[1]: *** No rule to make target
'/cygdrive/c/tmp/verilog_niosII_stratixII_2s60_standard/software/hw_bs
p/alt_sys_init.c', needed by 'all'. Stop.
make: *** [../hw_bsp-recurs-make-lib] Error 2
```

This error can also happen if you import a GCC 3 project to the version 10.0 SBT for Eclipse using the GCC 3 toolchain, and later change the toolchain to **MinGW Nios II GCC4**.

This issue only affects platforms running the Windows operating system.

Workaround

To avoid this error, follow these steps:

- Edit the application project and library project makefiles and add the inserted code as shown in Example 2.
- Right-click the BSP project, point to **Nios II** and click **Generate BSP**.
- Clean and rebuild the application and library projects.

Example 2. Makefile Correction for GCC 4 Build Error

```
#------------------------------------------------------------------------------
# The adjust-path macro
#
# If COMSPEC is defined, Make is launched from Windows through
# Cygwin. This adjust-path macro will call 'cygpath -u' on all
# paths to ensure they are readable by Make.
#
# If COMSPEC is not defined, Make is launched from *nix, and no adjustment
# is necessary
#------------------------------------------------------------------------------

# Beginning of inserted code
ifndef COMSPEC
ifdef ComSpec
COMSPEC = $(ComSpec)
endif # ComSpec
endif # !COMSPEC
#end of inserted code

ifdef COMSPEC
adjust-path = $(shell cygpath -u "$1")
adjust-path-mixed = $(shell cygpath -m "$1")
else
adjust-path = $1
adjust-path-mixed = $1
endif
```

Alternatively, install the Altera Complete Design Suite version 11.0 or later.

Nios II EDS Requires Administrative Permissions on Windows Vista and Windows 7

Description

On the Windows Vista and Windows 7 operating systems, the Nios II EDS requires your user account to have administrative permissions.

Attempts to write to a directory other than the user home directory might fail, especially writing to an installation directory. For example, you might be unable to create a software project under an example design directory. Error messages might vary.

Workaround

Obtain an administrator account on your workstation.

Alternatively, start Nios II EDS applications in administrator mode. For example, when launching the Nios II SBT for Eclipse, right-click **Nios II** <*version*> **Software Build Tools for Eclipse** and click **Run As Administrator**.

Missing Nios II Perspective

Description

If you launch the Nios II SBT for Eclipse from a universal naming convention (UNC) path, the Nios II perspective does not appear in Eclipse.

This issue appears only on platforms running the Windows operating system.

Workaround

Do not use a UNC path to launch the Nios II SBT for Eclipse. For example, if the SBT for Eclipse is installed on a network drive, map the network path to a Windows drive letter.

AN543 Contains Incorrect Information about Updating the Flash

Description

"Updating the Flash Option Bits on the Cyclone® III Development Board" in AN543, Debugging Nios II Software Using the Lauterbach Debugger contains incorrect information about updating the flash.

Workaround

The correct information is as follows:

To restore the flash factory image, refer to "Restoring the Factory Design to the Flash Device" in the Cyclone III Development Kit User Guide.

Error Running Nios II Project: ‘Downloading ELF Process failed’

Description

If the Nios II processor′s cpu.data_master port is not connected to all program memories (memories to which the **.elf** file is downloaded) the software project fails to run on Nios II hardware.

Failure to connect cpu.data_master to all program memories is a design error that the Nios II SBT does not detect.

Workaround

Connect cpu.data_master to all program memories.

C++ Exceptions Unsupported by the Nios II GCC 4 Toolchain

Description

With the GCC 4 toolchain, C++ exceptions are not supported.

Workaround

If your single-threaded project requires C++ exceptions, use the GCC 3 toolchain.

The Restart Button in the Debugger Does Not Work

Description

When debugging in the Nios II SBT for Eclipse, if you click the **Restart** icon, execution is suspended, with the following error message:

```
Don't know how to run. Try "help target."
```

Workaround

Terminate the program, download it again, and start the debugger.

nios2-console Does Not Work in the GCC 4 Command Shell

Description

If you try to run the **nios2-console** command in the GCC 4 Command Shell, you see the following error message:

```
Can't locate strict.pm in @INC ...
```

Workaround

Use the GCC 3 Command Shell.

Quartus II Compilation Warnings for Nios II Stratix II 2S60 ROHS Example

Description

You might see the following warnings if you try to compile the Nios II Stratix® II 2S60 ROHS example design, installed at <*Nios II EDS install path*>**/examples/vhdl/niosII_stratixII_2s60/standard** or downloaded from the Altera Wiki (www.alterawiki.com):

```
Warning (10541): VHDL Signal Declaration warning at
NiosII_stratixII_2s60_standard.vhd(59): used implicit default value
for signal
"cpu_data_master_read_data_valid_NiosII_stratixII_2s60_standard_clock_
0_in" because signal was never assigned a value or an explicit default
value. Use of implicit default value may introduce unintended design
optimizations.
```

```
Warning (10542): VHDL Variable Declaration warning at
altera_europa_support_lib.vhd(340): used initial value expression for
variable "arg_copy" because variable was never assigned a value
```

```
Warning (10542): VHDL Variable Declaration warning at
altera_europa_support_lib.vhd(344): used initial value expression for
variable "arg_length" because variable was never assigned a value
```

You can safely ignore these warnings.

The Nios II Stratix II 2S60 ROHS example is deprecated.

Workaround

None.

Warning Message: “'pragma_reverse_bitfields' attribute directive ignored”

Description

The Nios II GCC 4 tool chain does not support the C compiler pragma and flag for reverse bit fields. If you try to use the compiler pragma `reverse_bitfields`, you see the following warning:

```
'pragma_reverse_bitfields' attribute directive ignored
```

Similarly, the C compiler flag `-mreverse-bitfields` is unsupported.

Workaround

Use the GCC 3 tool chain.

Nios II SBT for Eclipse Unable to Create New Application and BSP from Template

Description

You should be able to create a new application and BSP from a template with the following steps:

1 Select **Nios II Application and BSP from Template**.

2 Choose your .**sopcinfo** file and select a template.

3 Click **Select an existing BSP project from your workspace.**

4 Click **Create**.

5 Choose the desired BSP options.

6 Click **Finish**.

However, if you try to create a new application and BSP with this method, the SBT creates a BSP, but no application, and the GUI becomes unresponsive.

Workaround

Perform the following steps:

1 Click **Cancel** to close the previous GUI.

2 Select **Nios II Application and BSP from Template** again.

3 Choose your .**sopcinfo** file and select a template.

4 Click **Select an existing BSP project from your workspace.**

5 From the BSP list, select the BSP created previously.

6 Click **Finish**.

Incorrect information about Embedded C++

Description

The Embedded Design Handbook contains the following incorrect statement about C++ support:

> The HAL supports only the standard Embedded C++ subset of the full C++ language. C++ programs that use features beyond this subset fail in the HAL environment. C++ features not available in Embedded C++ include polymorphism, templates, and single and multiple object inheritance. In general, features that consume a large amount of memory are not included in Embedded C++. Catch/throw exceptions fail in the MicroC/OS-II environment.

Nios II C++ support is not restricted to the obsolete Embedded C++ specification.

Workaround

In place of the incorrect paragraph, refer to the following correct information:

Nios II C++ language support depends on the GCC tool chain. The Nios II GCC 4 C++ tool chain supports the following features:

- Polymorphism
- Friendship and inheritance
- Multiple inheritance
- Virtual base classes
- Run-time type information (`typeid`)
- The `mutable` type qualifier
- Namespaces
- Templates
- New-and-delete style dynamic memory allocation
- Operator overloading
- Standard Template Library (STL)

Exceptions and new-style dynamic casts are not supported.

The Nios II Software Build Tools for Eclipse generate errors for Qsys (Beta)

Description

If you generate the simple socket server and web server Nios II example designs that contain tristate components, the Nios II Software Build Tools for Eclipse generates errors similar to the following:

```
[Error ] ‘EXT_FLASH_BASE’ undeclared
[Error ] ‘EXT_FLASH_BASE’ undeclared
```

Workaround

In the network_utilities.c file, on lines 213 and 419 change

```
EXT_FLASH_NAME
```

to

```
AV_TRI_S1_EXT_FLASH_0_NAME
```

and on line 290 change

```
EXT_FLASH_BASE
```

to

```
AV_TRI_S1_EXT_FLASH_0_BASE
```

C2H Compiler Does Not Work in the Nios II Command Shell with GCC 4

Description

The Nios II Command Shell with GCC 4 does not support the C2H Compiler.

Workaround

Use the Nios II Command Shell with GCC 3.

Nios II Launch Configurations Not Visible after Service Pack Installation

Description

After installing Altera Complete Design Suite version 10.1 Service Pack 1 over an existing version 10.1 installation, the first time you start the Nios II SBT for Eclipse, you might not see Nios II-specific launch configurations, such as **Run As Nios II Hardware** and **Run As Nios II ModelSim**. Only the **Run As Local C/C++ Application** configuration might be visible.

Workaround

Close and restart the Nios II SBT for Eclipse to see the Nios II-specific launch configurations.

This issue will be fixed in a future release of the Nios II EDS.

Build Errors on Software for Pre-Existing Design with SG-DMA

Description

If you have a Nios II system generated with SOPC Builder version 10.1 or earlier, and it contains the Scatter-Gather DMA (SG-DMA) component, you cannot build software for it with the Nios II SBT for Eclipse version 11.0 or later. If you attempt to do so, the Nios II compiler reports errors.

The SG-DMA component is updated for version 11.0 and the software driver is not compatible with older versions of hardware. The minimum compatible hardware version of a component is defined in the component's **_sw.tcl** file, for example `set_sw_property min_compatible_hw_version 11.0`.

Workaround

Regenerate the design with SOPC Builder version 11.0 or later before attempting to compile the software in the Nios II SBT for Eclipse.

System Timestamp for System ID Peripheral Is Always 0

Description

When the System ID Peripheral is not added to the top level of Qsys system, the system timestamp for the System ID peripheral is always 0.

Workaround

Remove the System ID Peripheral from any of the Qsys subsystems if you have them and add the System ID Peripheral only in the top level of Qsys system.

Missing .dat File Error Message: “Failed to open VHDL file”

Description

When you simulate a Qsys hardware design in VHDL that contains the JTAG UART core, and you run the simulation using the `ld_debug` command, you might see the following error message:

```
# ** Error: (vsim-7) Failed to open VHDL file
"system_tb_system_inst_jtag_input_stream.dat" in r mode.
```

You can safely ignore this error, because it does not affect the `stdout` output of the JTAG UART.

This error message does not appear when simulating a hardware design in the Verilog HDL.

Workaround

Run the simulation using the `ld` command, and the error is not displayed.

Alternatively, ignore the error message.

JTAG UART Interactive Window for Simulation Does Not Work

Description

When you simulate a Qsys system, the JTAG UART interactive window might not work.

Workaround

There is no workaround.

Nios II IDE shows “Source not found” during debugging

Description

When you are debugging a project in Nios II IDE and attempt to step into the `printf()` function, you might see an error message saying `Source not found`.

Workaround

Port your project to the Nios II SBT for Eclipse for future software development and debugging purposes, as described in the Using the Nios II Integrated Development Environment appendix of the *Nios II Software Developer's Handbook*.

Incorrect IRQ Information in system.h

Description

For Qsys hardware designs, the Vectored Interrupt Controller (VIC) is not supported properly by the Nios II SBT. The SBT publishes incorrect information to the **system.h** file. The `ALT_ENHANCED_INTERRUPT_API_PRESENT` definition is missing, and IRQ priorities are defined as -1.

Workaround

Use one of the following techniques to work around this issue:

- Create a set of `#define` statements in a new header (**.h**) file. This technique is recommended because your definitions are preserved if you regenerate the BSP. Ensure that the following information is defined properly:

  ```
  #define ALT_ENHANCED_INTERRUPT_API_PRESENT
  #define <component>_IRQ <interrupt priority from Qsys system>
  ```

- Manually code the correct `#define` statements in **system.h**. However, if you use this technique, you will lose your changes every time the BSP is regenerated and **system.h** is updated.
- Generate your system with SOPC Builder.
- Use the internal interrupt controller.

Intermittent Failures While Accessing CompactFlash Card

Description

The Nios II Development Kit version 5.0 and later includes a CompactFlash controller peripheral suitable for interfacing to CompactFlash cards in True IDE mode on Nios development boards. For True IDE mode to operate, CompactFlash cards require that the `ATASEL_N` input be driven to ground during power-up.

The CompactFlash controller peripheral includes a configurable power register, used to cycle power to CompactFlash cards from Nios II software through a metal oxide semiconductor field-effect transistor (MOSFET) on the Nios development boards. However, in certain development boards, power to the CompactFlash card does not turn off completely during this power cycle operation. Because of this condition, the CompactFlash might not sample the `ATASEL_N` pin during the power-cycle operation after FPGA configuration when this pin is driven to ground. Instead, the CompactFlash card might sample the `ATASEL_N` pin when power is first applied to the development board, when I/O is not yet driven by the FPGA (before FPGA configuration).

Workaround

If you encounter errors with CompactFlash when using the Nios development boards, try one of the following solutions:

- Use a different CompactFlash card. Certain cards are more susceptible to the power-cycling issue than others.
- Modify the Nios development board. This is recommended if you are familiar and comfortable with board-level modifications. Disconnect pin 9 (`ATASEL_N`) on the CompactFlash socket on your Nios development board and tie this pin to ground.

  The CompactFlash socket uses a staggered numbering on the pins (starting from pin 1: 1, 26, 2, 27, ...); refer to the CompactFlash Association specification for right-angle surface-mount connectors for exact specifications on this connector. This modification permanently enables True IDE mode operation.

‘Exclude from build’ Not Supported

Description

In the Nios II SBT for Eclipse version 9.1 SP2 and earlier, you can use the **Exclude from build** command to omit a source file from your project at build time. Starting with version 10.0, **Exclude from build** is replaced by the **Remove from Nios II build** and **Add to Nios II build** commands.

Due to an Eclipse platform limitation, **Exclude from build** still appears in the project context menu in version 10.0. However, it has no effect.

Workaround

To exclude a source file from your project when it is built, on the project context menu, click **Remove from Nios II build**. To restore it to the build, click **Add to Nios II build**.

C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File

Description

When accelerating a function in a file, the C2H Compiler cannot link subfunctions that are defined in a different file.

Workaround

Include all subfunctions called by the accelerated function within the same source code file.

Networking Examples

Description

If you are running a networking software example, you might be asked for a nine-digit number. You are directed to find this number on a sticker on your Nios development board, identified by the prefix **ASJ**. Not all Nios development boards have this sticker.

Workaround

If your Nios development board does not have a sticker with the letters **ASJ** followed by a nine-digit number, enter a unique nine-digit number when prompted. To avoid network address conflicts, ensure that this number is unique to each Nios board connected to your network.

The Qsys (Beta) Board Support Package (BSP) does not correctly specify the required driver when migrating a Common Flash Interface (CFI) memory component (altera_avalon_cfi_flash) that requires a Nios II software driver

Description

If you use Qsys upgrade transforms to migrate a Common Flash Interface (CFI) memory component (altera_avalon_cfi_flash), which requires a Nios II software driver with initialization, the Board Support Package (BSP) does not correctly specify the required driver. Although the BSP compiles and no error message is issued, the component does not function correctly.

Workaround

Instantiate the legacy Flash Memory Interface (CFI) component (altera_avalon_cfi_flash). To transfer the required driver assignments to an upgraded Generic Tristate Controller:

1 On the **System** menu, select **Run Qsys Upgrade Transforms**.

2 In the Generic Tristate Controller GUI, enter `embeddedsw.configuration.softwareDriver` in the **Key** column of the **Module Assignments** table and enter `altera_avalon_flash_driver` in the **Value** column.

3 Modify the **alt_sys_init.c** file created during BSP generation by following these steps:
a. In the Device Headers section, change the include statement from `altera_generic_tristate_controller.h` to `altera_avalon_cfi_flash.h`.
b. In the Device Storage section, change `ALTERA_GENERIC_TRISTATE_CONTROLLER_INSTANCE (<instance_name_caps>, <instance_name>)` to `ALTERA_AVALON_CFI_FLASH_INSTANCE(<instance_name_caps>, <instance_name>)`.

In the alt_sys_init function, change `ALTERA_GENERIC_TRISTATE_CONTROLLER_INIT (<instance_name_caps>, <instance_name>)` to `ALTERA_AVALON_CFI_FLASH_INIT (<instance_name_caps>, <instance_name>)`.

The Qsys (Beta) Board Support Package (BSP) might fail to compile if your system requires a Nios II software driver with initialization

Description

If your system includes a custom component that requires a Nios II software driver with initialization, the driver might not be found and the Board Support Package might fail to compile with errors about missing identifiers and assignments. Components are affected if the **_sw.tcl** file sets the property “`set_sw_property auto_initialize true`” to **request alt_sys_init.c** initialization.

Workaround

Instantiate the custom component in SOPC Builder and then open the system in Qsys. To transfer the required driver assignments to an upgraded Generic Tristate Controller, on the **System** menu, select **Run Qsys Upgrade Transforms**. Then, in the Generic Tristate Controller GUI, enter `embeddedsw.configuration.softwareDriver` in the **Key** column of the **Module Assignments** table and, in the Value column, enter the driver name that is listed in the component’s **_sw.tcl** file. You might need to add an entry into the `alt_irq_init` function, depending on the exact requirements of the component’s driver. You might also need to add additional driver-specific header files to the Device Headers section.

The Qsys (Beta) Board Support Package (BSP) fails to compile if you instantiate a Generic Tristate Controller for CFI

Description

If you instantiate a Generic Tristate Controller for CFI in Qsys (which requires a Nios II software driver with initialization) the Board Support Package fails to compile with errors regarding missing identifiers and assignments.

Workaround

Modify the **alt_sys_int.c** file created during BSP generation by following these steps:

1. In the Device Headers section, change the include statement from `altera_generic_tristate_controller.h` to `altera_avalon_cfi_flash.h`.
2. In the Device Storage section, change `ALTERA_GENERIC_TRISTATE_CONTROLLER_INSTANCE (<instance_name_caps>, <instance_name>)` to `ALTERA_AVALON_CFI_FLASH_INSTANCE (<instance_name_caps>, <instance_name>)`.

In the `alt_sys_init` function, change `ALTERA_GENERIC_TRISTATE_CONTROLLER_INIT (<instance_name_caps>, <instance_name>)` to `ALTERA_AVALON_CFI_FLASH_INIT (<instance_name_caps>, <instance_name>)`

Cannot simulate Nios II software with Qsys (Beta) systems

Description

The Nios II Software Build Tools (SBT) for Eclipse do not support the Run as ModelSim simulation flow to simulate Nios II software code with Qsys systems.

Workaround

Run your simulation outside the Eclipse environment. You can use the generated ModelSim **script mti_setup.tcl** as an example for your testbench and simulation environment.

To simulate a Nios II processor running software code, you must generate a Nios II Memory Initalization File (**.mif**). To generate the **.mif** file:

1 In Eclipse, right-click the application project, point to **Make Targets**, and then click **Build**.

2 Select **mem_init_install** and then click **Build.**

For the latest simulation recommendations, refer to New Qsys Issues.

101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support
www.altera.com/support

Nios II Embedded Design Suite Release Notes and Errata

© June 2011 RN-EDS-9.0

About These Release Notes

These release notes cover versions 10.0 through 11.0 of the Altera® Nios® II Embedded Design Suite (EDS). These release notes describe the revision history and errata for the Nios II EDS.

Errata are functional defects or errors, which might cause the product to deviate from published specifications. Documentation issues include errors, unclear descriptions, or omissions from current published specifications or product documents.

For the most up-to-date errata for this release, refer to the latest version of the *Nios II Embedded Design Suite Release Notes and Errata* on the Altera website. You can refer to release notes and errata for the Nios II processor core in the *Nios II Processor* chapter of the *MegaCore IP Library Release Notes and Errata*.

Revision History

Table 1 shows the revision history for the Nios II EDS, including the Nios II C2H Compiler.

Table 1. Nios II Embedded Design Suite Revision History (Part 1 of 2)

Version	Date	Description
11.0	May 2011	■ The Nios II EDS by default supports the GCC 4.1.2 version of the GNU Compiler Collection (GCC). GCC 3.4.6, the Nios II IDE, and Nios II C2H Compiler are optional features. They are available only if you enable **Legacy Package: Nios II IDE / GCC3 Toolchain / C2H Compiler** when you install the Altera Complete Design Suite. ■ Projects created with version 10.1 and earlier of the Nios II SBT need to be re-imported to create the Nios II launch configuration correctly.
10.1	December 2010	Maintenance release

Table 1. Nios II Embedded Design Suite Revision History (Part 2 of 2)

Version	Date	Description
10.0	July 2010	■ Nios II Software Build Tools (SBT) for Eclipse ■ Updated Eclipse to version 3.5 and Eclipse C/C++ development toolkit (CDT) to version 6.0 ■ Improved Source File management ■ Supports Multi-CPU launch ■ Updated to Nios II GCC 4.1.2 tool chain ■ Supports both Nios II SBT for Eclipse and Nios II Command Shell ■ GCC 4 improves compilation time compared to GCC 3 on Windows ■ Host platform ■ Supports Windows 7 ■ Supports SUSE Linux Enterprise 11 ■ Support for SUSE Linux Enterprise 9 discontinued ■ New Nios II software design examples for Reduced Gigabit Media Independent Interface (RGMII) ■ Web Server (RGMII) ■ Simple Socket Server (RGMII) ■ Legacy hardware design examples (Stratix II and Cyclone II) are removed ■ Installer changes ■ Nios II EDS must be installed next to Quartus II

For more information on new features and changes, refer to the *Nios II Processor Reference Handbook*, the *Nios II Software Developer's Handbook*, and the *Nios II C2H Compiler User Guide*. For information about changes to the Nios II processor core, refer to the *Nios II Processor* chapter of the *MegaCore IP Library Release Notes and Errata*.

Deprecated Features

C2H SBT Command-Line Flow

The C2H command-line flow using **nios2-c2h-generate-makefile** is deprecated as of version 10.0.

Notice about Environment Variables

The `QUARTUS_ROOTDIR` and `SOPC_KIT_NIOS2` environment variables will be deprecated in a future release. If you have custom scripts that use these variables, you should be prepared to change them to be independent of these variables, or to set the environment variables yourself.

Errata Summary

Table 2 summarizes the issues that affect the Nios II Embedded Design Suite versions 10.0 through 11.0. For a detailed description of each issue, click on the issue name.

Refer to the Altera Knowledge Database for older errata and solutions.

Table 2. Nios II EDS Errata *(1)* (Part 1 of 4)

Added or Updated	Issue	Page	Affected Version		
			11.0	10.1	10.0
	Nios II SBT for Eclipse				
Jun 2011	Nios II Launch Configurations Not Visible after Service Pack Installation	7	✓	—	—
	Build Errors on Software for Pre-Existing Design with SG-DMA	7	✓	—	—
	No .sopcinfo File Name Shown in Nios II BSP Properties Page	7	Fixed	✓	—
	Only One Thread Visible When Debugging MicroC/OS-II Application	8	—	Fixed	—
	Nios II Consoles Do Not Work With Multiprocessor Project	8	Fixed	✓	✓
	Nios II SBT for Eclipse Hangs When Project Run as Nios II Hardware	8	Fixed	✓	—
	Run As Nios II ModelSim Does Not Work	9	Fixed	✓	—
	Error Building Imported Project: 'No rule to make target'	9	✓	✓	✓
	Debugger Breaks in crt0.s Instead of main()	10	—	—	Fixed
	Nios II Options Do Not Appear in Eclipse	11	Fixed	✓	✓
	Error Building Imported Project: 'target pattern contains no %'	11	—	—	Fixed
Jan 2011	Nios II SBT for Eclipse Unable to Create New Application and BSP from Template	12	✓	✓	—
	Run Configuration Cannot Find Imported Custom Makefile Project	12	—	Fixed	✓
	Errors Creating or Importing Software Projects	13	—	Fixed	✓
Sep 2010	The Restart Button in the Debugger Does Not Work	13	✓	✓	✓
July 2010	Missing Nios II Perspective	14	✓	✓	✓
	Error Marker Persists on BSP Project After Build Error Corrected	14	✓	✓	✓
	'Exclude from build' Not Supported	14	✓	✓	✓
Nov 2009	Error Messages on Console When Debugging	15	✓	✓	✓
	Errors Converting Nios II IDE Multiprocessor Project	15	✓	✓	✓
	Stop on Startup Option in Run Configuration Has No Effect	15	✓	✓	✓
	C2H Compiler				
Jan 2011	C2H Compiler Does Not Work in the Nios II Command Shell with GCC 4	16	✓	✓	✓
Jul 2008	Functions Declared Without a Return Type Are Not Supported	16	✓	✓	✓
Oct 2007	Pre-7.1 Systems Are Not Supported	16	✓	✓	✓
	--src-dir SBT Argument Does Not Work With C2H	16	✓	✓	✓

Table 2. Nios II EDS Errata *(1)* (Part 2 of 4)

Added or Updated	Issue	Page	Affected Version		
			11.0	10.1	10.0
May 2007	Accelerator Generation Failure If Tools Are Installed in Path With Spaces	17	✓	✓	✓
	The C2H Compiler Regenerates an Accelerator Unnecessarily	17	✓	✓	✓
	Error: c2h_accelerator_base_addresses.h: No such file or directory	17	✓	✓	✓
	Java Heap Space Exception if Quartus II Compilation is Enabled	18	✓	✓	✓
	Pointer Dereferences to Volatile Types	18	✓	✓	✓
Dec 2006	C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File	18	✓	✓	✓
Jun 2006	Array Elements in Structures Do Not Copy Correctly	19	✓	✓	✓
	Clean Build Causes Build Failure	19	✓	✓	✓
	Changing Build Configurations Produces Unexpected Results	20	✓	✓	✓
	Hardware Accelerators Remain After Deleting the Software Project *(2)*	20	✓	✓	✓
	Incorrect Results From Logical or Conditional Operation With Side-Effects	20	✓	✓	✓
	Launch SOPC Builder Button in C2H View	21	✓	✓	✓
	Development Boards				
Jun 2006	Intermittent Failures While Accessing CompactFlash Card	21	✓	✓	✓
	Documentation Issues				
Jun 2011	Incorrect information about Embedded C++	22	✓	✓	✓
	Error Building Project: 'No rule to make target'	23	—	Fixed	✓
	Eclipse CDT Features Not Supported by Nios II Plugins	23	Fixed	✓	✓
	Incorrect Information about Nested Exceptions	25	—	—	Fixed
July 2010	AN543 Contains Incorrect Information about Updating the Flash	26	✓	✓	✓
Feb 2010	Valid Range of hal.log_flags is –1 to 3	26	✓	✓	✓
Nov 2009	NicheStack TCP/IP Stack - Nios II Edition Does Not Support TFTP	27	✓	✓	✓
	Error Message After Renaming Project: "Resource is out of sync with the system"	27	✓	✓	✓
Mar 2009	Compiler Flags for Building Custom Newlib	27	✓	✓	✓
Oct 2007	Nios II IDE Online Help Expand Buttons Do Not Work	28	✓	✓	✓
	Hardware Abstraction Layer				
Jun 2011	Missing Structure Member Errors in alt_log_printf.c with Small JTAG UART Driver	29	—S	—	Fixed
	Hardware Example Designs				
Jan 2011	Quartus II Compilation Warnings for Nios II Stratix II 2S60 ROHS Example	30	✓	✓	✓
	Software Examples				
Jun 2006	Networking Examples	30	✓	✓	✓
	Flash Programmer				
Dec 2006	elf2flash File Size Limit	31	✓	✓	✓

Table 2. Nios II EDS Errata *(1)* (Part 3 of 4)

Added or Updated	Issue	Page	Affected Version		
			11.0	10.1	10.0
	Hardware Simulation				
Jun 2011	Missing .dat File Error Message: “Failed to open VHDL file”	31	✓	—	—
	JTAG UART Interactive Window for Simulation Does Not Work	31	✓	—	—
Dec 2006	Error “UNC paths are not supported” Launching ModelSim	32	✓	✓	✓
Jun 2006	Uninitialized .bss Variables in Simulation	32	✓	✓	✓
	Host Platform				
Jan 2011	Nios II EDS Requires Administrative Permissions on Windows Vista and Windows 7	32	✓	✓	✓
Nov 2009	Cannot Launch Nios II SBT for Eclipse on Red Hat Linux	33	✓	✓	✓
Dec 2007	Windows/Cygwin: Nios II Processor Generation Failure	33	✓	✓	✓
Oct 2007	Windows Vista: Limited Support in Nios II EDS	34	—	—	—
	Nios II IDE				
	Building Projects				
Mar 2009	Nios II IDE Command-Line Tools Select Wrong Workspace on Linux	34	✓	✓	✓
	Nios II IDE Command-Line Tools Hang on Windows	35	✓	✓	✓
Nov 2008	Nios II IDE Cannot Find stdio.h in Outline View	36	✓	✓	✓
Jul 2008	Nios II IDE Hangs With UNC Project Path	36	✓	✓	✓
	Build Failure with Nios II Advanced Exceptions, MMU, and MPU	36	✓	✓	✓
May 2007	Build Command Not Functional for BSPs Created With the Nios II SBT	37	✓	✓	✓
	Linker Errors with Dual-Port Memories	37	✓	✓	✓
	User-Managed BSP Settings Not Supported in Nios II IDE	37	✓	✓	✓
	URL Project Location Causes Project Creation Error	38	✓	✓	✓
Dec 2006	Compilation Error with Separate Exception Stack Option	38	✓	✓	✓
	Incorrect Stack and Heap Free Space Report	38	✓	✓	✓
	Nios II IDE Reports Problems Without Displaying Error in Console	38	✓	✓	✓
	Debugging Projects				
Jun 2011	Nios II IDE shows “Source not found” during debugging	39	✓	—	—
Nov 2009	Method for Accessing MMU and MPU Registers in the Debugger	39	✓	✓	✓
Nov 2008	IDE Cannot Display Imported Profiling Data	40	✓	✓	✓
Jul 2008	Error Message when Downloading .elf File	40	✓	✓	✓
	Variable Casting Unsupported in ISS	40	✓	✓	✓
	Trace Debug Does Not Support Instruction-Related Exceptions	41	✓	✓	✓
Oct 2007	Trace Debug Does Not Support the JMPI Instruction	41	✓	✓	✓
May 2007	Cannot Locate Source Code in Driver Files Shared by Multiple Projects	41	✓	✓	✓
	Console Window Is Not Updated After ISS Error	41	✓	✓	✓
	ISS Fails on Designs Containing Triple Speed Ethernet MAC or SG-DMA Components	41	✓	✓	✓
	Memory Window Sets Control Register Values Incorrectly	42	✓	✓	✓

Table 2. Nios II EDS Errata *(1)* (Part 4 of 4)

Added or Updated	Issue	Page	Affected Version		
			11.0	10.1	10.0
Dec 2006	Programs That Interact With a Terminal Console on Windows Do Not Work	42	✓	✓	✓
	"Run as ModelSim" in the Nios II IDE Fails	42	✓	✓	✓
	The Restart Command on the Run Menu Does Not Work	43	✓	✓	✓
Jun 2006	Watchpoints Do Not Work on Certain Variables	43	✓	✓	✓
Navigating Projects					
Dec 2006	Nios II IDE Freezes While Displaying the Splash Screen	43	✓	✓	✓
	Internal Error When Double-Clicking on a Large Objdump File	44	✓	✓	✓
Jun 2006	C/C++ Scanner Does Not Support Certain C/C++ Constructs	44	✓	✓	✓
Nios II SBT					
Jun 2011	Incorrect IRQ Information in system.h	44	✓	✓	—
	Nios II SBT Creates Spurious BSP for System with MMU	45	Fixed	✓	—
July 2010	Error Running Nios II Project: 'Downloading ELF Process failed'	45	✓	✓	✓
May 2010	BSP Editor Does Not Show Command-Line Help	46	✓	✓	✓
Feb 2010	BSP Not Updated for Memory Size Changes in SOPC Builder	46	✓	✓	✓
Nov 2009	Build Errors on Interrupt API Calls	46	✓	✓	✓
Oct 2007	SBT Fails if Nios II EDS is Installed in a Path Containing Spaces	47	✓	✓	✓
May 2007	User-Managed BSP Settings Not Supported in Nios II IDE	37	✓	✓	✓
Peripherals					
Jun 2011	System Timestamp for System ID Peripheral Is Always 0	48	✓	—	—
Jul 2008	Unaligned Transfers of Small Payloads Fail on SG-DMA	48	✓	✓	✓
May 2007	DMA Controller Always Busy in Burst Mode	48	✓	✓	✓
	Non-System-Wide Reset Can Cause Improper Initialization of Mailbox Core	48	✓	✓	✓
Target Software					
Nov 2009	EIC-Based Nios II System Crashes With NicheStack	49	✓	✓	✓
Dec 2006	stdio Does Not Work with MicroC/OS-II and Small C Library	49	✓	✓	✓
Jun 2006	cout From MicroC/OS-II Task Does Not Send Data to stdout	49	✓	✓	✓
Toolchain and Utilities					
Jun 2011	nios2-download Cannot Find JTAG Instance IDs	50	Fixed	✓	—
Jan 2011	Warning Message: "'pragma_reverse_bitfields' attribute directive ignored"	50	✓	✓	✓
Sep 2010	nios2-console Does Not Work in the GCC 4 Command Shell	50	✓	✓	✓
July 2010	C++ Exceptions Unsupported in a Multi-Threaded Environment	51	✓	✓	✓
	C++ Exceptions Unsupported by the Nios II GCC 4 Toolchain	51	✓	✓	✓
	memcpy() Optimization Misalignment	51	✓	✓	✓
Dec 2006	GNU Assembler Does Not Accept the --defsym Flag	51	✓	✓	✓

Notes to Table 2:

(1) Refer to the Altera Knowledge Database for older errata and solutions.

(2) For further information, refer to the *Nios II C2H Compiler User Guide*.

Nios II SBT for Eclipse Errata

This section describes in detail the issues related to the Nios II SBT for Eclipse.

Nios II Launch Configurations Not Visible after Service Pack Installation

After installing Altera Complete Design Suite version 10.1 Service Pack 1 over an existing version 10.1 installation, the first time you start the Nios II SBT for Eclipse, you might not see Nios II-specific launch configurations, such as **Run As Nios II Hardware** and **Run As Nios II ModelSim**. Only the **Run As Local C/C++ Application** configuration might be visible.

Workaround

Close and restart the Nios II SBT for Eclipse to see the Nios II-specific launch configurations.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Build Errors on Software for Pre-Existing Design with SG-DMA

If you have a Nios II system generated with SOPC Builder version 10.1 or earlier, and it contains the Scatter-Gather DMA (SG-DMA) component, you cannot build software for it with the Nios II SBT for Eclipse version 11.0 or later. If you attempt to do so, the Nios II compiler reports errors.

The SG-DMA component is updated for version 11.0 and the software driver is not compatible with older versions of hardware. The minimum compatible hardware version of a component is defined in the component's **_sw.tcl** file, for example `set_sw_property min_compatible_hw_version 11.0`.

Workaround

Regenerate the design with SOPC Builder version 11.0 or later before attempting to compile the software in the Nios II SBT for Eclipse.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

No .sopcinfo File Name Shown in Nios II BSP Properties Page

In the Nios II SBT for Eclipse, the **SOPC Info** field in the Nios II BSP properties page should display the **.sopcinfo** filename for the underlying hardware design. In the Nios II SBT for Eclipse running on the Windows operating system, this field is empty.

Design Impact

The **SOPC Info** field is a read-only field. This issue does not affect functionality.

Workaround

None needed.

Solution Status

Fixed in version 11.0 of the Nios II EDS

Only One Thread Visible When Debugging MicroC/OS-II Application

When you use GCC 4 tool chain, the Nios II SBT for Eclipse debugs a multi-threaded application as if it were a single-threaded application.

Workaround

Use GCC 3 tool chain.

Alternatively, install the Altera Complete Design Suite version 10.1 or later.

Solution Status

Fixed in version 10.1 of the Nios II EDS

Nios II Consoles Do Not Work With Multiprocessor Project

Launch groups do not work with multiple JTAG UARTs in multiprocessor projects.

When you launch a multiprocessor software project with a launch group, if the design uses multiple JTAG UARTs, multiple Nios II consoles appear, but only one of them is active.

Workaround

Launch the software projects individually.

Alternatively, install the Altera Complete Design Suite version 11.0 or later.

Solution Status

Fixed in version 11.0 of the Nios II EDS

Nios II SBT for Eclipse Hangs When Project Run as Nios II Hardware

On Windows, the Nios II SBT for Eclipse sometimes becomes unresponsive when you try to run (or debug) a project with the **Run as Nios II Hardware** (or **Debug as Nios II Hardware**) command.

Workaround

Use one of the following approaches to work around this issue:

- In the **Run Configuration** (or **Debug Configuration**) dialog box, on the **Target Connection** tab, click **Refresh Connections** before running (or debugging) the application. Each time you restart the SBT for Eclipse, you must perform this step for each application you wish to run or debug.
- Visit the MySupport website and request Quartus® II 10.1 patch 0.18.
- Install the Altera Complete Design Suite version 11.0 or later.

Solution Status

Fixed in version 11.0 of the Nios II EDS

Run As Nios II ModelSim Does Not Work

In the Windows operating system, if you try to run an application with the **Run As Nios II ModelSim** command, you see the following error message:

```
No such file or directory
```

Workaround

Visit the MySupport website and request Quartus II 10.1 patch 0.18.

Alternatively, install the Altera Complete Design Suite version 11.0 or later.

Solution Status

Fixed in version 11.0 of the Nios II EDS

Error Building Imported Project: ‘No rule to make target’

If an application or library project is created with version 9.1 SP2 or earlier (using the GCC 3 toolchain), and you import it to the version 10.0 Nios II SBT for Eclipse using the GCC 4 toolchain, you might see the build error shown in Example 1.

Example 1.

```
make all
Info: Building ../hw_bsp
make --no-print-directory -C ../hw_bsp
make[1]: *** No rule to make target `/cygdrive/c/tmp/
verilog_niosII_stratixII_2s60_standard/software/hw_bsp/
alt_sys_init.c', needed by `all'. Stop.
make: *** [../hw_bsp-recurs-make-lib] Error 2
```

This error can also happen if you import a GCC 3 project to the version 10.0 SBT for Eclipse using the GCC 3 toolchain, and later change the toolchain to **MinGW Nios II GCC4**.

This issue only affects platforms running the Windows operating system.

Workaround

To avoid this error, follow these steps:

1. Edit the application project and library project makefiles and add the italicized code as shown in Example 2.

Example 2.

```
#------------------------------------------------------------------------------
# The adjust-path macro
#
# If COMSPEC is defined, Make is launched from Windows through
# Cygwin. This adjust-path macro will call 'cygpath -u' on all
# paths to ensure they are readable by Make.
#
# If COMSPEC is not defined, Make is launched from *nix, and no adjustment
# is necessary
#------------------------------------------------------------------------------

# Beginning of inserted code
ifndef COMSPEC
ifdef ComSpec
COMSPEC = $(ComSpec)
endif # ComSpec
endif # !COMSPEC
#end of inserted code

ifdef COMSPEC
adjust-path = $(shell cygpath -u "$1")
adjust-path-mixed = $(shell cygpath -m "$1")
else
adjust-path = $1
adjust-path-mixed = $1
endif
```

2. Right-click the BSP project, point to **Nios II** and click **Generate BSP**.
3. Clean and rebuild the application and library projects.

Alternatively, install the Altera Complete Design Suite version 11.0 or later.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Debugger Breaks in crt0.s Instead of main()

If you attempt to launch a debug session by right clicking the project, without first creating a run configuration, the debugger breaks in the initialization code in **crt0.s**, rather than at the top of the `main()` function. You must explicitly create the run configuration before launching the debug session.

Workaround

You can work around this issue with either of the following methods:

- Create the run configuration with the **Run Configuration** dialog box. You can then debug with this run configuration by right-clicking the project, clicking **Debug As**, and selecting the debug configuration. The program correctly breaks at `main()`.
- Manually set a breakpoint at `main()`.

Alternatively, install the Altera Complete Design Suite version 10.0 or later.

Solution Status

Fixed in version 10.0 of the Nios II EDS

Nios II Options Do Not Appear in Eclipse

When Nios II SBT for Eclipse starts, the Nios II plugins might fail to load, resulting in the following symptoms:

- The Nios II perspective is not available
- No Nios II items appear when you click **New** in the File menu

Eclipse can behave this way if you have installed the Nios II EDS in the same directory as a previous installation.

When the Altera Complete Design Suite installer installs the Nios II EDS, you specify a root directory for the EDS file structure. For example, if the Nios II EDS 9.1 is installed on the Windows operating system, the root directory might be **c:\altera\91\nios2eds**. For simplicity, Altera documentation refers to this directory as <*Nios II EDS install path*>.

When the Altera Complete Design Suite installer removes the Nios II EDS, it leaves behind some files in <*Nios II EDS install path*>. If you reinstall the Nios II EDS in the same directory, these leftover files might cause the Nios II SBT for Eclipse to work incorrectly.

Workaround

To correct this problem, execute the following steps:

1. Uninstall the Nios II EDS.
2. Delete <*Nios II EDS install path*>, including all subdirectories, before reinstalling the Nios II EDS. Or reinstall the Nios II EDS in a different directory.

Alternatively, install the Altera Complete Design Suite version 11.0 or later.

Solution Status

Fixed in version 11.0 of the Nios II EDS

Error Building Imported Project: ‘target pattern contains no %’

If your application or library makefile uses an absolute path and is generated with the GCC 3 toolchain, and you import it to the version 10.0 Nios II SBT for Eclipse using the GCC 4 toolchain, you get the following build error:

```
makefile:587: *** target pattern contains no '%'. Stop.
```

Workaround

Recreate the project using the SBT for Eclipse version 10.0.

Alternatively, use the GCC 3 toolchain.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Nios II SBT for Eclipse Unable to Create New Application and BSP from Template

You should be able to create a new application and BSP from a template with the following steps:

1. Select **Nios II Application and BSP from Template**.
2. Choose your .**sopcinfo** file and select a template.
3. Click **Select an existing BSP project from your workspace.**
4. Click **Create**.
5. Choose the desired BSP options.
6. Click **Finish**.

However, if you try to create a new application and BSP with this method, the SBT creates a BSP, but no application, and the GUI becomes unresponsive.

Workaround

Perform the following steps:

1. Click **Cancel** to close the previous GUI.
2. Select **Nios II Application and BSP from Template** again.
3. Choose your .**sopcinfo** file and select a template.
4. Click **Select an existing BSP project from your workspace.**
5. From the BSP list, select the BSP created previously.
6. Click **Finish**.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Run Configuration Cannot Find Imported Custom Makefile Project

After you import a project using the **Import Custom Makefile for Nios II Software Build Tools Project** option, the Nios II SBT fails to recognize the imported custom makefile as a Nios II C/C++ application project. As a result, the project name does not appear in the list in the run configuration.

Workaround

1. In the **Run Configuration** dialog box, select any visible Nios II C/C++ project. The project need not be related to your custom makefile project.

   ☞ If there is no Nios II C/C++ project in your workspace, create a dummy project, such as Hello World, through the File menu, by clicking **New**, **Nios II Application and BSP from Template**.

2. Turn on **Enable browse for file system ELF file**.
3. Browse to the correct **.elf** file for your custom makefile project to complete creation of a run configuration.

Alternatively, install the Altera Complete Design Suite version 10.1 or later.

Solution Status

Fixed in version 10.1 of the Nios II EDS

Errors Creating or Importing Software Projects

If you place software projects in your Eclipse workspace directory, you might experience project-related errors. For example, you might see one of the following error messages:

-
  ```
  Unable to create project
  Reason: Unable to create project in workspace directory
  ```

-
  ```
  Failed to import
  Reason: Unable to import project
  ```

Workaround

Switch to a different workspace, or create a new workspace, separate from your project files. To switch workspaces or create a workspace, on the File menu, click **Switch Workspace**.

Alternatively, install the Altera Complete Design Suite version 10.0 or later.

Solution Status

Fixed in version 10.1 of the Nios II EDS

The Restart Button in the Debugger Does Not Work

When debugging in the Nios II SBT for Eclipse, if you click the **Restart** icon, execution is suspended, with the following error message:

```
Don't know how to run. Try "help target."
```

Workaround

Terminate the program, download it again, and start the debugger.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Missing Nios II Perspective

If you launch the Nios II SBT for Eclipse from a universal naming convention (UNC) path, the Nios II perspective does not appear in Eclipse.

This issue appears only on platforms running the Windows operating system.

Workaround

Do not use a UNC path to launch the Nios II SBT for Eclipse. For example, if the SBT for Eclipse is installed on a network drive, map the network path to a Windows drive letter.

Solution Status

Not fixed

Error Marker Persists on BSP Project After Build Error Corrected

In the Nios II SBT for Eclipse, if an application project fails to build, error markers appear in the BSP project, if the build error message contains a path to BSP project source files. Even after you have rebuilt the application project without errors, the error markers persist in the BSP project.

Workaround

Rebuild your BSP project.

Alternatively, if your application will always be associated with the same BSP, set the Eclipse project references by following these steps:

1. Right-click your application project, point to **Properties** and click **Project References**.
2. Select the desired BSP for your application.
3. Build your application project again to clear the error markers in the BSP project.

☞ If you later link your application project with a different BSP, you must manually correct the Eclipse project references.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

‘Exclude from build’ Not Supported

In the Nios II SBT for Eclipse version 9.1 SP2 and earlier, you can use the **Exclude from build** command to omit a source file from your project at build time. Starting with version 10.0, **Exclude from build** is replaced by the **Remove from Nios II build** and **Add to Nios II build** commands.

Due to an Eclipse platform limitation, **Exclude from build** still appears in the project context menu in version 10.0. However, it has no effect.

Workaround

To exclude a source file from your project when it is built, on the project context menu, click **Remove from Nios II build**. To restore it to the build, click **Add to Nios II build**.

Error Messages on Console When Debugging

When debugging, you might see the following messages in the Nios II SBT for Eclipse console:

```
No symbol "auto" in current context.
No symbol "new" in current context.
```

These are normal messages from the GNU Debugger (GDB) console, not error messages. This issue does not impact the debug process.

Workaround

No workaround needed.

Solution Status

None needed.

Errors Converting Nios II IDE Multiprocessor Project

If you try use the **nios2-convert-ide2sbt** utility to convert an IDE multiprocessor project to the Nios II SBT, error messages appear, and the project fails to generate properly.

Workaround

Create new Nios II SBT application and BSP projects using project settings equivalent to the original Nios II IDE project settings.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Stop on Startup Option in Run Configuration Has No Effect

Turning on **Stop on Startup at** in the **Debugger** tab of a run configuration has no effect.

Workaround

You can manually set any breakpoint in the debug perspective.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

C2H Compiler Errata

This section describes in detail the issues related to the Nios II C2H Compiler.

C2H Compiler Does Not Work in the Nios II Command Shell with GCC 4

The Nios II Command Shell with GCC 4 does not support the C2H Compiler.

Workaround

Use the Nios II Command Shell with GCC 3.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Functions Declared Without a Return Type Are Not Supported

The C2H compiler does not support functions without an explicitly declared return type.

Workaround

If you are using the implicit `int` return type, declare the return type explicitly. If your function has no return value, declare it as `void`.

Solution Status

Not fixed

Pre-7.1 Systems Are Not Supported

Starting in version 7.1, the C2H Compiler does not support SOPC Builder systems created in SOPC Builder version 7.0 or earlier (systems based on a **class.ptf** file).

Workaround

Before using the C2H Compiler, you must upgrade your SOPC Builder system. Open the system in SOPC Builder version 7.1 or later. SOPC Builder prompts you to convert the system, creating a ***_hw.tcl** file compatible with the C2H Compiler.

Solution Status

Not fixed

--src-dir SBT Argument Does Not Work With C2H

When using the C2H Compiler with Nios II SBT, an error occurs when using the `nios2-app-generate-makefile` argument `--src-dir` to specify a directory of source files. An example of the error is:

```
multiple definition of 'my_accelerated_function'
```

Workaround

Specify source files individually using the `--src-files` argument.

Solution Status

Not fixed

Accelerator Generation Failure If Tools Are Installed in Path With Spaces

If the path to your installation of the Altera Design Suite contains spaces, the C2H Compiler fails to generate the accelerators.

Workaround

Reinstall the tools to a path containing no spaces.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

The C2H Compiler Regenerates an Accelerator Unnecessarily

The C2H Compiler might regenerate an accelerator even when the accelerated function is unchanged. This problem can result from changes to files included by the C file containing the accelerated function. The C2H Compiler fails to check that the generated hardware description language (HDL) matches the previously generated HDL causing the system to be regenerated.

Workaround

To avoid this issue move the accelerator-specific information from the include file to a separate include file. This workaround prevents regeneration of the system when the HDL is unchanged.

Solution Status

Not fixed

Error: c2h_accelerator_base_addresses.h: No such file or directory

When a C2H accelerator is compiled for the first time, the following compile-time error can result if the **Analyze all accelerators** option is selected:

```
c2h_accelerator_base_addresses.h: No such file or directory.
```

Workaround

Click **Build software and generate SOPC Builder system** and build once before building with the **Analyze all accelerators** option.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Java Heap Space Exception if Quartus II Compilation is Enabled

If your design contains a C2H accelerator, and you select **Build software, generate SOPC Builder system, and run Quartus II compilation** in the C2H view, you might see the following error during Quartus II compilation:

```
Exception in thread "main" java.lang.OutOfMemoryError: Java
heap space

make: *** [c2h_hdl-t] Error 1
```

Workaround

Select the **Build software and generate SOPC builder system** option in the C2H Compiler settings window, and then manually launch the Quartus II software to compile the design.

Solution Status

Not fixed

Pointer Dereferences to Volatile Types

The C2H Compiler treats pointer dereferences to a volatile type as if they alias all other pointer dereferences. Pointers that are restrict-qualified are treated the same way.

The two loops in Example 3 cannot be scheduled concurrently because the `volatile` qualification overrides the `__restrict__` pragma.

Example 3. Non-Concurrent Loops

```
volatile int * __restrict__ fifo_rd = FIFO_RD_BASE;
volatile int * __restrict__ fifo_wr = FIFO_WR_BASE;
for ()
{
  *fifo_wr = ....;
}
for ()
{
  ... = *fifo_rd;
}
```

Workaround

Divide the function into multiple interrupt request (IRQ)-enabled accelerators that are launched concurrently from the processor, and use FIFO buffers to communicate between them.

Solution Status

Not fixed

C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File

When accelerating a function in a file, the C2H Compiler cannot link subfunctions that are defined in a different file.

Workaround

Include all subfunctions called by the accelerated function within the same source code file.

Solution Status

Not fixed

Array Elements in Structures Do Not Copy Correctly

C2H accelerators do not correctly copy array elements that are elements of structures.

In Example 4, the a and b elements of the structure copy correctly, but the buf element does not. After this assignment, struct_a equals {9, 8, {3, 3, 3, 3}}.

Example 4. Array Elements of Structs

```
typedef struct my_struct {
  int a;
  int b;
  int buf[BUF_SIZE];
}MY_STRUCT;
MY_STRUCT struct_a = {1, 2, {3, 3, 3, 3}};
MY_STRUCT struct_b = {9, 8, {7, 7, 7, 7}};
struct_a = struct_b;
```

Workaround

Copy the array elements explicitly, as shown in Example 5.

Example 5. Copying Array Elements Explicitly

```
{
  int i=0;
  do
    {
      struct_a.buf[i] = struct_b.buf[i];
      i++;
    } while (i<LENGTH_OF_BUF_ELEMENT)
}
```

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Clean Build Causes Build Failure

Performing a clean build on a Nios II IDE project that contains a hardware accelerator can cause the next build to fail in the IDE, because the clean build erroneously deletes a file required by the C2H Compiler.

Workaround

Do not perform a clean build on projects that use hardware accelerators. If you have already performed a clean build, recompile with option **Build software, generate SOPC Builder system, and run Quartus II compilation** to regenerate the necessary files.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Changing Build Configurations Produces Unexpected Results

The C2H Compiler does not support multiple build configurations (for example Release or Debug) in the Nios II IDE. After creating one or more accelerators in a particular configuration, the C2H Compiler produces undefined results if you switch to a different build configurations and create more accelerators.

Workaround

For a specific SOPC Builder system and Nios II IDE project, specify C2H accelerators in only one build configuration. You can use multiple build configurations, as long as only one configuration specifies C2H Compiler settings.

Solution Status

Not fixed

Hardware Accelerators Remain After Deleting the Software Project

If a system contains C2H accelerators, deleting the software project that defines the accelerators does not remove the accelerators from the hardware system, and the accelerator logic remains in the SOPC Builder system.

Workaround

To remove an accelerator from a system, delete the accelerator from the C2H view in the Nios II IDE first, and then recompile the software project. The C2H Compiler then removes the accelerator from the SOPC Builder system. Once the compilation is complete then the software application can be deleted from the workspace.

Solution Status

Not fixed

Incorrect Results From Logical or Conditional Operation With Side-Effects

The C2H Compiler always evaluates both operands of logical (&&, `||`) and conditional (`?:`) operators. This is different from expected American National Standards Institute (ANSI) C behavior, where operands are evaluated left-to-right, and unnecessary operands are skipped.

For example, in the expression `(i-- && j--)`, if the value of `i` is zero, ANSI C does not evaluate the right-hand-side (RHS) expression, and `j` is not decremented. By contrast, the C2H Compiler evaluates both sides, decrementing `j`.

The following expressions are other examples that might be affected by this issue:

```
if (i-- || j++)
    ...;


a = ((cond == 1)?  i++ : j++);
```

Workaround

Use logical and conditional operations whose operators have no side effects. Operations with side effects include pre- and post- increment and decrement operations (++, --), memory operations (*, [], ., ->), and function calls.

Solution Status

Not fixed

Launch SOPC Builder Button in C2H View

When the Nios II IDE workspace contains multiple projects with multiple system libraries, the incorrect SOPC Builder system might open when you click **Launch SOPC Builder** in the C2H view.

Workaround

Launch SOPC Builder from the Quartus II software. Alternatively, keep only one system library project open at a time while using the C2H Compiler.

Solution Status

Not fixed

Development Board Errata

This section describes in detail the Nios II EDS issues related to Nios development boards.

Intermittent Failures While Accessing CompactFlash Card

The Nios II Development Kit version 5.0 and later includes a CompactFlash controller peripheral suitable for interfacing to CompactFlash cards in True IDE mode on Nios development boards. For True IDE mode to operate, CompactFlash cards require that the ATASEL_N input be driven to ground during power-up.

The CompactFlash controller peripheral includes a configurable power register, used to cycle power to CompactFlash cards from Nios II software through a metal oxide semiconductor field-effect transistor (MOSFET) on the Nios development boards. However, in certain development boards, power to the CompactFlash card does not turn off completely during this power cycle operation. Because of this condition, the CompactFlash might not sample the ATASEL_N pin during the power-cycle operation after FPGA configuration when this pin is driven to ground. Instead, the CompactFlash card might sample the ATASEL_N pin when power is first applied to the development board, when I/O is not yet driven by the FPGA (before FPGA configuration).

Workaround

If you encounter errors with CompactFlash when using the Nios development boards, try one of the following solutions:

- Use a different CompactFlash card. Certain cards are more susceptible to the power-cycling issue than others.
- Modify the Nios development board. This is recommended if you are familiar and comfortable with board-level modifications. Disconnect pin 9 (`ATASEL_N`) on the CompactFlash socket on your Nios development board and tie this pin to ground.

☞ The CompactFlash socket uses a staggered numbering on the pins (starting from pin 1: 1, 26, 2, 27, ...); refer to the CompactFlash Association specification for right-angle surface-mount connectors for exact specifications on this connector. This modification permanently enables True IDE mode operation.

Solution Status

Not fixed

Documentation Errata

This section describes in detail the Nios II EDS documentation issues.

Incorrect information about Embedded C++

The *Embedded Design Handbook* contains the following incorrect statement about C++ support:

> The HAL supports only the standard Embedded C++ subset of the full C++ language. C++ programs that use features beyond this subset fail in the HAL environment. C++ features not available in Embedded C++ include polymorphism, templates, and single and multiple object inheritance. In general, features that consume a large amount of memory are not included in Embedded C++. Catch/throw exceptions fail in the MicroC/OS-II environment.

Nios II C++ support is not restricted to the obsolete Embedded C++ specification.

Workaround

In place of the incorrect paragraph, refer to the following correct information:

Nios II C++ language support depends on the GCC tool chain. The Nios II GCC 4 C++ tool chain supports the following features:

- Polymorphism
- Friendship and inheritance
- Multiple inheritance
- Virtual base classes
- Run-time type information (`typeid`)
- The `mutable` type qualifier
- Namespaces

- Templates
- New-and-delete style dynamic memory allocation
- Operator overloading
- Standard Template Library (STL)

Exceptions and new-style dynamic casts are not supported.

Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

Error Building Project: ‘No rule to make target’

The *Getting Started with Graphical User Interface* and *Getting Started from the Command Line* chapters of the *Nios II Software Developer’s Handbook* do not include the following information:

You cannot link Nios II GCC 3 projects with Nios II GCC 4 projects. Your application, library and BSP projects must all use the same Nios II GCC toolchain.

If you switch between Nios II GCC 3 and Nios II GCC 4, make sure you run **make clean** on your application, library and BSP projects before rebuilding.

When importing a GCC 3 BSP, such as a project created with version 9.1 SP2 or earlier, to version 10.0 SBT for Eclipse using the GCC 4 toolchain, after importing the BSP, regenerate the makefile.

If you do not follow these rules, you might see the following error:

```
make[1]: *** No rule to make target `/cygdrive/c/.../bsp/
alt_sys_init.c', needed by 'all'. Stop.
make: *** [../bsp/-recurs-make-lib] Error 2
```

Solution Status

Fixed in version 10.1 documentation

Eclipse CDT Features Not Supported by Nios II Plugins

The following information is missing from the *Getting Started with Graphical User Interface* chapter of the *Nios II Software Developer’s Handbook*:

The features listed in the left column of Table 3 are supported by the Eclipse CDT plugins, but are not supported by Nios II plugins. The right column lists alternative features supported by the Nios II plugins.

Table 3. Eclipse CDT Features Not Supported by the Nios II Plugins (Part 1 of 2)

Unsupported CDT Feature	Alternative Nios II Feature
New Project Wizard	
C/C++	
■ C Project	To create a new project, use one of the following Nios II wizards:
■ C++ Project	■ Nios II Application
	■ Nios II Application and BSP from Template
	■ Nios II Board Support Package
	■ Nios II Library
■ Convert to a C/C++ Project	
■ Source Folder	
Build configurations	
■ Right-click project and point to **Build Configurations**	The Nios II plugins only support a single build configuration.
■ **Debugger** tab	
■ **Stop on startup**	This feature is supported only at the top of `main()`.
Exclude from Build (from version 10.0 onwards)	
Right-click source files	Use **Remove from Nios II Build** and **Add to Nios II Build**.
Project Properties	
C/C++ Build	
■ Builder Settings	
■ Makefile generation	By default, the Nios II SBT generates makefiles automatically.
■ Build location	The build location is determined with the **Nios II Application Properties** or **Nios II BSP Properties** dialog box.
■ Behavior	
■ Build on resource save (Auto build)	
■ Build Variables	
■ Discovery Options	
■ Environment	
■ Settings	
■ Tool Chain Editor	
■ Current builder	
■ Used tools	To change the toolchain, use the **Current tool chain** option
Project Properties, continued	
C/C++ General	
■ Enable project specific settings	
■ Documentation tool comments	
■ Documentation	
■ File Types	
■ Indexer	
■ Build configuration for the indexer	The Nios II plugins only support a single build configuration.

Table 3. Eclipse CDT Features Not Supported by the Nios II Plugins (Part 2 of 2)

Unsupported CDT Feature	Alternative Nios II Feature
- Language Mappings - Paths and Symbols	Use **Nios II Application Properties** and **Nios II Application Paths**
Window Preferences	
C/C++	
- Build scope	
- Build project configurations	The Nios II plugins only support a single build configuration.
- Build Variables	
- Environment	
- File Types	
- Indexer	
- Build configuration for the indexer	The Nios II plugins only support a single build configuration.
- Language Mappings	
- New CDT project wizard	

Workaround

Refer to the information in Table 3.

Alternatively, refer to the most recent version of the *Getting Started with Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*.

Solution Status

Fixed in version 11.0 documentation

Incorrect Information about Nested Exceptions

"Exception Processing" in the *Programming Model* chapter of the *Nios II Processor Reference Handbook* incorrectly states that multiple interrupts with different requested interrupt levels (RILs) must not be assigned to the same shadow register set.

Workaround

The correct information is as follows:

Multiple interrupts with different RILs can be assigned to a single shadow register set. However, with multiple register sets, you must not allow the RILs assigned to one shadow register set to overlap the RILs assigned to another register set.

Table 4 and Table 5 illustrate the validity of register set assignments when preemption within a register set is enabled.

Table 4. Example of Illegal RIL Assignment

RIL	Register Set 1	Register Set 2
1	IRQ0	
2	IRQ1	
3		IRQ2
4	IRQ3	
5		IRQ4
6		IRQ5
7		IRQ6

Table 5. Example of Legal RIL Assignment

RIL	Register Set 1	Register Set 2
1	IRQ0	
2	IRQ1	
3	IRQ3	
4		IRQ2
5		IRQ4
6		IRQ5
7		IRQ6

Solution Status

Fixed in version 10.0 documentation

AN543 Contains Incorrect Information about Updating the Flash

"Updating the Flash Option Bits on the Cyclone® III Development Board" in AN543, *Debugging Nios II Software Using the Lauterbach Debugger* contains incorrect information about updating the flash.

Workaround

The correct information is as follows:

Updating the Flash on the Cyclone III Development Board

To restore the flash factory image, refer to "Restoring the Factory Design to the Flash Device" in the *Cyclone III Development Kit User Guide*.

Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

Valid Range of hal.log_flags is –1 to 3

The Nios II EDS documentation incorrectly states the valid range for the `hal.log_flags` BSP setting. Valid values of `hal.log_flags` range from –1 through 3.

Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

NicheStack TCP/IP Stack - Nios II Edition Does Not Support TFTP

The Nios II documentation should include the following information:

Versions of the NicheStack TCP/IP Stack other than the Nios II Edition include optional Trivial File Transfer Protocol (TFTP) client and server applications. However, the TFTP client and server are not thread safe. They cannot be used in systems with the MicroC/OS-II RTOS.

TFTP is not available with the NicheStack TCP/IP Stack - Nios II Edition. The TFTP client and server features are disabled.

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Error Message After Renaming Project: “Resource is out of sync with the system”

The *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer’s Handbook* does not explain the correct procedure for renaming a project.

Workaround

The missing information is as follows:

To rename a project in the Nios II SBT for Eclipse, execute the following steps:

1. Right click the BSP project and click **Rename**.
2. Type the new BSP name.
3. Right click the BSP project and click **Refresh**.

☞ If you neglect to refresh the project, you might see the following error message when you attempt to build it:

```
Resource <original_bsp_name> is out of sync with the system
```

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Compiler Flags for Building Custom Newlib

“Common BSP Tasks” in the *Using the Nios II Software Build Tools* chapter of the *Nios II Software Developer’s Handbook* describes how to compile a custom version of the newlib library. However, it does not list the compiler flags that must be selected to generate the correct library.

Workaround

The missing information is as follows:

The Nios II EDS provides a number of precompiled newlib implementations. The provided libraries are precompiled with every viable combination of the GCC compiler flags shown in Table 6.

When you create a BSP with a precompiled newlib, the Nios II SBT selects the newlib matching your BSP's compiler settings. When you create a custom newlib, you must ensure that the compiler flags listed in Table 6 match your BSP's settings.

Table 6. GCC Compiler Options for Newlib

Option Name	Purpose
`-pg`	Link for profiling with gprof
`-EB`	Generate big-endian code
`-mstack-check`	Enable stack checking
`-mno-hw-mul`	Disable use of the `mul` family of instructions
`-mhw-mulx`	Enable use of the `mulx` family of instructions
`-mcustom-fpu-cfg`	Use a floating point custom instruction

☞ The Nios II-specific compiler flag `-mcustom-fpu-cfg` requires the Nios II processor core to be implemented with the floating-point custom instruction as specified in the flag argument. The compiler flag can have one of the following two argument values:

- `-mcustom-fpu-cfg=60-1`—Use the Nios II floating-point custom instruction without divider
- `-mcustom-fpu-cfg=60-2`—Use the Nios II floating-point custom instruction with divider

For details about the Nios II-specific compiler flags `-mstack-check`, `-mno-hw-mul`, and `-mhw-mulx`, refer to "Altera Nios II Options" in *Using the GNU Compiler Collection (GCC)*, installed with the Nios II EDS. For details about the `-pg` compiler flag, refer to "Compiling a Program for Profiling" in *GNU Profiler*, installed with the Nios II EDS. To find the installed documentation, on the Windows Start menu, click **Programs** > **Altera** > **Nios II EDS <*version*>** > **Nios II <*version*> Documentation**, and then click **Literature**.

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Nios II IDE Online Help Expand Buttons Do Not Work

The expand text arrow and the **Show All** option in the online help do not work in Internet Explorer.

Workaround

In Internet Explorer, carry out the following actions:

1. Click the **Refresh/Show Current Topic** icon (yellow arrows icon at the top of the search pane). The table of contents appears with the selected topic highlighted.
2. Click the highlighted topic to refresh the browser frame. Expanded text works.

Alternatively, use Firefox 2.0.0.6.

Solution Status

Not fixed

Hardware Abstraction Layer Errata

This section describes in detail the Nios II Hardware Abstraction Layer issues.

Missing Structure Member Errors in alt_log_printf.c with Small JTAG UART Driver

In a BSP, if you enable the small JTAG UART driver (`altera_avalon_jtag_uart_driver.enable_small_driver`), and enable Altera logging (`hal.log_port`) using the JTAG UART as the log port, when you build the project, the compiler reports missing structure members in **alt_log_printf.c**. The project fails to build.

Workaround

To avoid this issue, execute the following steps:

1. In function `alt_log_jtag_uart_print_control_reg()` in **alt_log_printf.c**, insert the italicized code as shown in Example 6.

Example 6. ALT_LOG_PRINTF() Correction

```
ALT_LOG_PRINTF(
    "%s SW CirBuf = %d, HW FIFO wspace=%d AC=%d WI=%d RI=%d WE=%d RE=%d\r\n",
    header,

/* Beginning of inserted code */
#ifndef ALTERA_AVALON_JTAG_UART_SMALL
    (dev->tx_out-dev->tx_in),
#else
    0,
#endif
/* End of inserted code */

    space,ac,wi,ri,we,re);
```

2. Rebuild the project.

Alternatively, install the Altera Complete Design Suite version 11.0 or later.

Solution Status

Fixed in version 11.0 of the Nios II EDS

Hardware Example Design Errata

This section describes in detail the Nios II EDS hardware example design issues.

Quartus II Compilation Warnings for Nios II Stratix II 2S60 ROHS Example

You might see the following warnings if you try to compile the Nios II Stratix® II 2S60 ROHS example design, installed at *<Nios II EDS install path>***/examples/vhdl/niosII_stratixII_2s60/standard** or downloaded from the Altera Wiki (**www.alterawiki.com**):

```
Warning (10541): VHDL Signal Declaration warning at\
    NiosII_stratixII_2s60_standard.vhd(59): used implicit default value for signal\
    "cpu_data_master_read_data_valid_NiosII_stratixII_2s60_standard_clock_0_in"\
    because signal was never assigned a value or an explicit default value. Use of\
    implicit default value may introduce unintended design optimizations.
Warning (10542): VHDL Variable Declaration warning at\
    altera_europa_support_lib.vhd(340): used initial value expression for variable\
    "arg_copy" because variable was never assigned a value
Warning (10542): VHDL Variable Declaration warning at\
    altera_europa_support_lib.vhd(344): used initial value expression for variable\
    "arg_length" because variable was never assigned a value
```

You can safely ignore these warnings.

☞ The Nios II Stratix II 2S60 ROHS example is deprecated.

Workaround

None.

Solution Status

Not fixed.

Software Example Errata

This section describes in detail the Nios II EDS software example issues.

Networking Examples

If you are running a networking software example, you might be asked for a nine-digit number. You are directed to find this number on a sticker on your Nios development board, identified by the prefix **ASJ**. Not all Nios development boards have this sticker.

Workaround

If your Nios development board does not have a sticker with the letters **ASJ** followed by a nine-digit number, enter a unique nine-digit number when prompted. To avoid network address conflicts, ensure that this number is unique to each Nios board connected to your network.

Solution Status

Not fixed

Flash Programmer Errata

This section describes in detail the Nios II EDS issues related to the flash programmer.

elf2flash File Size Limit

The **elf2flash** utility supports **.elf** files up to approximately 24 MBytes in size. The **elf2flash** utility might fail on files larger than 24 MBytes, with the error message `java.lang.OutOfMemoryError`.

Workaround

Lower the number of symbols in your **.elf** file by turning off debug symbols.

Alternatively, specify less initialized data in the application.

Solution Status

Not fixed

Hardware Simulation Errata

This section describes in detail the Nios II EDS issues related to hardware simulation.

Missing .dat File Error Message: “Failed to open VHDL file”

When you simulate a Qsys hardware design in VHDL that contains the JTAG UART core, and you run the simulation using the `ld_debug` command, you might see the following error message:

```
# ** Error: (vsim-7) Failed to open VHDL file
"system_tb_system_inst_jtag_input_stream.dat" in r mode.
```

You can safely ignore this error, because it does not affect the `stdout` output of the JTAG UART.

This error message does not appear when simulating a hardware design in the Verilog HDL.

Workaround

Run the simulation using the `ld` command, and the error is not displayed.

Alternatively, ignore the error message.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

JTAG UART Interactive Window for Simulation Does Not Work

When you simulate a Qsys system, the JTAG UART interactive window might not work.

Workaround

There is no workaround.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Error "UNC paths are not supported" Launching ModelSim

If you launch the ModelSim®-Altera simulation software from a working directory that is mapped via a universal naming convention (UNC) path (a path that starts with `//` instead of drive letter), you receive the following error message in SOPC Builder: **UNC paths are not supported. Defaulting to Windows directory**. This error occurs because ModelSim is calling a command shell, which does not support UNC paths.

Workaround

Map the UNC path to a drive letter and use the drive letter to reference the working directory in the launching shell.

Solution Status

Not fixed

Uninitialized .bss Variables in Simulation

If your program reads the value of an uninitialized `.bss` variable during HDL simulation, and the BSP (system library) is compiled with the **ModelSim only, no hardware support** property enabled in Nios II IDE, a warning appears about unfiltered data being `'x'`. This warning appears because when this property is enabled, the code that clears the `.bss` memory region is omitted to speed up HDL simulation so this memory region is uninitialized. The `.bss` region contains global and static local variables that are not initialized by the application so they default to a value of zero. When the Nios II processor reads uninitialized variables, it displays a warning and converts any of the bits of the uninitialized data to zero which correctly mimics the effect of the missing `.bss` clearing code. The Hardware Abstraction Layer (HAL) code that executes before and after `main()` might use `.bss` variables, so these warnings might appear even if your application does not use the `.bss` section.

Solution Status

Not fixed

Host Platform Errata

This section describes in detail the Nios II EDS issues related to the host development platform.

Nios II EDS Requires Administrative Permissions on Windows Vista and Windows 7

On the Windows Vista and Windows 7 operating systems, the Nios II EDS requires your user account to have administrative permissions.

Attempts to write to a directory other than the user home directory might fail, especially writing to an installation directory. For example, you might be unable to create a software project under an example design directory. Error messages might vary.

Workaround

Obtain an administrator account on your workstation.

Alternatively, start Nios II EDS applications in administrator mode. For example, when launching the Nios II SBT for Eclipse, right-click **Nios II <*version*> Software Build Tools for Eclipse** and click **Run As Administrator**.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Cannot Launch Nios II SBT for Eclipse on Red Hat Linux

You might be unable to launch Nios II SBT for Eclipse on the Red Hat Linux operating system. This is an issue with the version of XULRunner in Eclipse 3.4. The issue is resolved in XULRunner 1.9.1, which is available with Eclipse 3.5.

Workaround

Upgrade XULRunner to a newer version. To determine what version of XULRunner you have, type the following command at the command prompt:

```
xulrunner -v↵
```

To upgrade XULRunner using the **yum** software package manager, type the following command:

```
sudo yum update xulrunner↵
```

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Windows/Cygwin: Nios II Processor Generation Failure

You might see the following error message when you generate your Nios II system in SOPC Builder:

```
Error: Generator program for module 'cpu_fpoint' did NOT run
successfully
```

This error might occur on a Windows-based system when all of the following conditions are true:

- You have installed a version of Cygwin other than the one distributed with the Quartus II software.
- You launch the Quartus II software and SOPC Builder from a Nios II Command Shell.
- You enable the floating point custom instruction (FPCI) in the Nios II processor core.

Workaround

Launch the Quartus II software and SOPC Builder from the Windows Start menu, and regenerate your system.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Windows Vista: Limited Support in Nios II EDS

The Quartus II software introduced Windows Vista (32-bit and 64-bit) support in version 7.2. However, several ensuing releases of the Nios II Embedded Design Suite support only the following platforms:

- Windows 2000
- Windows XP
- Windows XP (64)
- Windows Vista Business Edition (32-bit)
- SUSE 9 (32-bit)
- SUSE 9 (64-bit)
- Red Hat Linux version 3.0 (32-bit)
- Red Hat Linux version 3.0 (64-bit)
- Red Hat Linux version 4.0 (32-bit)
- Red Hat Linux version 4.0 (64-bit)

Workaround

Upgrade to a more recent version of the Nios II EDS.

To see the current operating system support for the Altera Complete Design Suite tools, refer to the Operating System Support page of the Altera website.

Solution Status

Fixed in version 8.1 of the Nios II EDS

Nios II IDE Errata: Building Projects

This section describes in detail the issues related to building projects in the Nios II IDE.

Nios II IDE Command-Line Tools Select Wrong Workspace on Linux

On Linux systems, the Nios II IDE command-line tools use the default Eclipse project workspace, regardless what workspace you are using in the IDE. The default workspace is at `$SOPC_KIT_NIOS2/eclipse/nios2-ide-workspace-<`*`version`*`>`.

This issue affects the following tools:

- `nios2-create-system-library`
- `nios2-create-application-project`
- `nios2-build-project`

- `nios2-import-project`
- `nios2-delete-project`

Workaround

Explicitly supply the Nios II IDE command-line tools with a workspace location, by means of the `-data` command-line argument. The syntax of the `-data` argument is as follows:

```
-data <path to workspace>
```

The path to the workspace must be absolute, and must not contain whitespace. Aside from these restrictions, any valid file system path can be used for the workspace. Example 7 shows how to specify a workspace path.

Example 7. Providing a Non-Default Workspace Location in Linux

```
nios2-create-project -data $HOME/myworkspace <other arguments>↵
```

Solution Status

Not fixed

Nios II IDE Command-Line Tools Hang on Windows

On Windows systems, the Nios II IDE command-line tools are sometimes unable to locate the project workspace. When this happens, the tools might hang.

This issue potentially affects the following tools:

- `nios2-create-system-library`
- `nios2-create-application-project`
- `nios2-build-project`
- `nios2-import-project`
- `nios2-delete-project`

Workaround

Explicitly supply the Nios II IDE command-line tools with a workspace location, by means of the `-data` command-line argument. The syntax of the `-data` argument is as follows:

```
-data <path to workspace>
```

The path to the workspace must be absolute, and must not contain whitespace. Aside from these restrictions, any valid file system path can be used for the workspace. The workspace shown in Example 8 is the default workspace used by the Nios II IDE in version 9.0.

Example 8. Providing a Workspace Location

```
nios2-create-project -data c:/altera/90/nios2eds/\
    bin/eclipse/nios2-ide-workspace-9.0 <other arguments>
```

Solution Status

Not fixed

Nios II IDE Cannot Find stdio.h in Outline View

If you create a new project using the Nios II IDE and try to open the **stdio.h** file from the Outline view before building the project, the IDE displays the error message **No include files were found that matched that name**.

Workaround

Build the project before attempting to open **stdio.h**.

Solution Status

Not fixed

Nios II IDE Hangs With UNC Project Path

In the **New Project** dialog box, if you turn on **Specify Location** and specify the path in UNC form, the IDE might hang.

Workaround

Map the UNC path to a remote drive which looks like a Windows drive to the Nios II IDE.

Solution Status

Not fixed

Build Failure with Nios II Advanced Exceptions, MMU, and MPU

Projects created in the Nios II IDE for an SOPC Builder system containing a Nios II processor configured with advanced exceptions, the memory management unit (MMU), or the memory protection unit (MPU) fail to build, and generate the following error messages:

```
ERROR - Classic build flow for Nios II system library does not support
the Nios II \
    advanced exceptions.
Use the non-classic Nios II Board Support Package instead.
<timestamp> - (SEVERE) generate: java.lang.IllegalStateException: \
    java.lang.IllegalStateException: \
    com.altera.ingenuous.GTFElement.GTFElementErrorException: \
    <error> element in GTF script
make[1]: *** [system_description/../obj/system.h-t] Error 1
make: *** [system_project] Error 2
```

Workaround

Create the software project with the Nios II SBT command-line development flow. Alternatively, use the Nios II SBT for Eclipse.

Solution Status

Not fixed

Build Command Not Functional for BSPs Created With the Nios II SBT

The build option in the Nios II IDE menu does not rebuild BSPs imported to the IDE.

Workaround

The Nios II SBT for Eclipse can both build and debug projects created on the command line. The Nios II SBT for Eclipse is the preferred tool for debugging Nios II SBT projects.

For information about the Nios II SBT for Eclipse, refer to the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*.

In the Nios II IDE, you can build the BSP by building the associated application project.

Solution Status

Not fixed

Linker Errors with Dual-Port Memories

If your instruction master and data master ports are connected to the same dual-port memory and the ports have different addresses, your code fails to run or you experience a linker error. The Nios II IDE does not warn you of the addressing violation.

Workaround

Assign the same address to both ports of the dual-port memory.

Solution Status

Not fixed

User-Managed BSP Settings Not Supported in Nios II IDE

For projects created with the Nios II SBT and imported to the Nios II IDE, the IDE configuration settings have no effect.

For example, objdump, compiler, and linker settings made in the IDE are ignored. This behavior occurs because Nios II SBT projects are not IDE-managed projects. In addition, the make-related preferences do not pertain to imported Nios II SBT projects. The IDE ignores these options during the build process.

Workaround

Make these settings in the project's makefile.

Solution Status

Not fixed

URL Project Location Causes Project Creation Error

When you try to create a new project in an existing workspace, you might see an error dialog box saying:

```
Project cannot be created. Reason: Internal Error
```

This error might occur if the path to any project in the workspace is a URL location, for example **file:/F:/Design**. To view the path, right-click on the project and select **Properties**.

Workaround

Import your existing application and system library projects to a new workspace.

Solution Status

Not fixed

Compilation Error with Separate Exception Stack Option

Choosing the **Use a separate exception stack option** might cause the following compilation error when building a project:

```
VARIABLE %STACK_POINTER%"
```

This error occurs if the exception stack is larger than the memory available for it.

Workaround

On the system library properties page for the project, turn off the separate exception stack or reduce the **Maximum exception stack size** setting.

Solution Status

Not fixed

Incorrect Stack and Heap Free Space Report

The makefile reports an incorrect number of bytes free for the stack and heap, if the heap and stack are in different memory regions.

Workaround

No workaround available.

Solution Status

Not fixed

Nios II IDE Reports Problems Without Displaying Error in Console

When building a project, the Nios II IDE reports problems, but the build output in the console does not contain any errors.

The Nios II IDE incorrectly reports some linker warnings as errors, with a dialog box saying **Errors exist in a required project**. The Dhrystone software example exhibits this behavior, and recompiling the project again makes the issue go away.

Workaround

If the Console output does not contain errors, then the project built correctly. On subsequent builds, the linker step is skipped and the errors do not appear.

Solution Status

Not fixed

Nios II IDE Errata: Debugging Projects

This section describes in detail the issues related to debugging projects in the Nios II IDE.

Nios II IDE shows “Source not found” during debugging

When you are debugging a project in Nios II IDE and attempt to step into the `printf()` function, you might see an error message saying `Source not found`.

Workaround

Port your project to the Nios II SBT for Eclipse for future software development and debugging purposes, as described in the *Using the Nios II Integrated Development Environment* appendix of the *Nios II Software Developer’s Handbook*.

Solution Status

Not fixed

Method for Accessing MMU and MPU Registers in the Debugger

You might observe an error if you try to read or write an MMU or MPU register via the Nios II Debugger.

Workaround

To read an MPU region, execute the following steps:

1. Set region `INDEX` in the `MPUBASE` register.
2. Exit and reenter debug mode, that is, single step.
3. Set the `RD` bit in the `MPUACC` register.
4. Exit and reenter debug mode, that is, single step.
5. Read back `MPUBASE` for pertinent information.
6. Read back `MPUACC` for pertinent information.

For a system with an MMU or MPU, this workaround allows you to read and write the current values of the registers. However, you cannot use it to control MPU regions or MMU TLB entries.

The debug core copies the Nios II processor’s register values to its internal memory when the processor enters debug mode. The debug core writes register values back to the processor only when the processor leaves debug mode. Therefore, if you attempt to set an MPU region with several consecutive values, only the last one, when leaving debug mode, is committed to the processor.

Solution Status

Not fixed

IDE Cannot Display Imported Profiling Data

If you create a software project in the Nios II SBT command line flow, generate profiling data to **gmon.out** in the Nios II command shell, and then import the project to the IDE, you cannot use the Profiling perspective to view the profiling data. When you attempt to view **gmon.out**, the IDE displays the following error message:

```
nios2-elf-gprof: ../: not in a.out format.
```

Workaround

Run `nios2-elf-gprof` from the Nios II command shell to generate a profiler report.

Solution Status

Not fixed

Error Message when Downloading .elf File

If the **Build Automatically** option is turned on in the IDE, you might get the following error when downloading a **.elf**:

```
"2 [main] sh 5736 fork: child 3892 - died waiting; \
    for longjmp before initialization, errno 11"
```

The **Build Automatically** option is off by default, as recommended by the Eclipse help system.

Workaround

Turn off the **Build Automatically** option and download the **.elf** file again.

Solution Status

Not fixed

Variable Casting Unsupported in ISS

Casting variables in the Variables view in the Debug perspective when using the Nios II ISS might cause an exception dialog box to open.

This exception might occur if you select **Debug as ISS** in the Nios II IDE and try to cast variables via the Variables view.

Workaround

There is no workaround.

Solution Status

Not fixed

Trace Debug Does Not Support Instruction-Related Exceptions

The instruction-related exception handler is not supported by the Nios II trace tools.

Workaround

There is no workaround.

Solution Status

Not fixed

Trace Debug Does Not Support the JMPI Instruction

The JMPI instruction is not supported by the Nios II trace tools.

Workaround

There is no workaround.

Solution Status

Not fixed

Cannot Locate Source Code in Driver Files Shared by Multiple Projects

If you hit a breakpoint in a driver file, and that driver file is shared with another project that is closed, the Nios II IDE might indicate that it cannot locate the source code.

Workaround

Open the closed system library project and resume debugging.

Solution Status

Not fixed

Console Window Is Not Updated After ISS Error

After performing a **Run as ISS**, if you receive an ISS error in the console window, the console is not updated subsequently.

Workaround

Close the console window after receiving an ISS error. A new console window opens when a new message is available.

Solution Status

Not fixed

ISS Fails on Designs Containing Triple Speed Ethernet MAC or SG-DMA Components

You receive an Internal Error when attempting to perform an ISS simulation of designs containing the Altera Triple Speed Ethernet media access control (MAC) or SG-DMA components because the Nios II ISS does not support these components.

Workaround

Remove the Triple Speed Ethernet MAC and SG-DMA components from your system and perform ISS simulation on the simplified system. You can also simulate the design in ModelSim or test it on hardware.

☞ Before removing the Triple Speed Ethernet MAC and SG-DMA components, make a copy of the unmodified system to ensure that you can return to the original configuration.

Solution Status

Not fixed

Memory Window Sets Control Register Values Incorrectly

The memory window might incorrectly set values in memory-mapped control registers. For example, writing 0x1234 to a byte addressed register results in the value 0x3434 being stored in the register. The memory window shows this incorrect value.

Workaround

Use the GNU debugger (GDB) console window in the IDE, instead of the memory window, to write to the registers. For example, type the following command:

```
set {int} <register address>=0x1234↵
```

☞ You must refresh the memory window for it to correctly display the target value.

Solution Status

Not fixed

Programs That Interact With a Terminal Console on Windows Do Not Work

Programs with this behavior work in version 6.0 and earlier, but do not work in Nios II IDE version 6.1 and later.

The Eclipse platform in version 6.1 and later of the IDE (on Windows only) sends the string `\r\n` instead of just `\n` when running or debugging using the Terminal. This behavior can break existing software designs, and it is inconsistent with `nios2-terminal`, which still just sends `\n`.

Workaround

Change the software running on the Nios II processor to parse for `\r\n` as well as `\n`.

Solution Status

Not fixed

"Run as ModelSim" in the Nios II IDE Fails

The **Run as ModelSim** command might fail on launch configurations created in version 7.0 or earlier of the IDE. This problem does not occur for new launch configurations.

Workaround

Select a location for the ModelSim tool from the launch configuration dialog box. You can use the **Browse** button next to the **ModelSim path** group, or type in a path to the ModelSim directory (for example **c:/altera/71/modelsim_ae/win32aloem**).

Solution Status

Not fixed

The Restart Command on the Run Menu Does Not Work

The **Restart** command on the **Run** menu does not work.

Workaround

Stop the program, then debug it again. If the debugger is hung in an endless loop, use the following **bash** alias to break the target, then stop it:

```
alias break="kill -2 \'ps -a | grep nios2-elf-gdb | cut -f6 -d' '\'"
```

Solution Status

Not fixed

Watchpoints Do Not Work on Certain Variables

Watchpoints do not work on a variable whose size is not 32 bits.

Workaround

Change the types of global and static local variables to `int`, `long`, or `unsigned long` before setting watchpoints on them.

Solution Status

Not fixed

Nios II IDE Errata: Navigating Projects

This section describes in detail the issues related to navigating projects in the Nios II IDE.

Nios II IDE Freezes While Displaying the Splash Screen

After clicking **Switch Workspace** on the File menu on a Windows machine, a Nios II IDE splash screen appears. Unfortunately, this splash screen obscures the dialog box that asks you to specify the new workspace. As a result, the IDE appears to freeze.

Workaround

Press Alt-Tab to switch applications. Two relevant application icons appear: an Eclipse icon associated with the splash screen and a Nios II IDE icon associated with the **Workspace** dialog box. Select the Nios II icon to bring the dialog box to the foreground.

Solution Status

Not fixed

Internal Error When Double-Clicking on a Large Objdump File

On Windows when opening a large **objdump** file in the Nios II IDE, you might get the following error message:

```
Unable to create this part due to an internal error.
Reason for the failure: Editor could not be initialized.
```

Workaround

Adjust the Windows launch arguments for the Nios II IDE editor. Perform the following steps:

1. On the Windows Start menu, browse to the **Nios II EDS** program icon, right-click it, then click **Properties**. The **Windows Properties** dialog box appears.
2. In the **Target** field, append `"vmargs -Xmx1024m"` to the end of the path to the Nios II IDE executable. For example:

   ```
   C:\altera\72\nios2eds\bin\eclipse\nios2-ide.exe -vmargs -Xmx1024m
   ```

Solution Status

Not fixed

C/C++ Scanner Does Not Support Certain C/C++ Constructs

The C/C++ scanner performs C/C++ Search, navigation, open declaration, and parts of content assist. Due to limitations of the C/C++ scanner, these features do not work with C code constructs not supported by the C++ language. An example is functions that take a function pointer as an argument.

Workaround

If the C/C++ Search fails, use the File Search facility.

Solution Status

Not fixed

Nios II SBT Errata

This section describes in detail the issues in the Nios II SBT.

Incorrect IRQ Information in system.h

For Qsys hardware designs, the Vectored Interrupt Controller (VIC) is not supported properly by the Nios II SBT. The SBT publishes incorrect information to the **system.h** file. The `ALT_ENHANCED_INTERRUPT_API_PRESENT` definition is missing, and IRQ priorities are defined as –1.

Workaround

Use one of the following techniques to work around this issue:

- Create a set of `#define` statements in a new header (**.h**) file. Ensure that the following information is defined properly:

  ```
  #define ALT_ENHANCED_INTERRUPT_API_PRESENT
  #define <component>_IRQ <interrupt priority from Qsys system>
  ```

  This technique is recommended because your definitions are preserved if you regenerate the BSP.
- Manually code the correct `#define` statements in **system.h**. However, if you use this technique, you will lose your changes every time the BSP is regenerated and **system.h** is updated.
- Generate your system with SOPC Builder.
- Use the internal interrupt controller.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Nios II SBT Creates Spurious BSP for System with MMU

The Nios II SBT is not intended to generate BSPs for the Nios II MMU. The SBT incorrectly lets you create a BSP for a Qsys-generated Nios II system with an MMU. This BSP includes a **system.h** file with incorrect MMU information.

You cannot use the Nios II SBT to create a BSP for the Nios II MMU.

Affected Configurations

Qsys designs with the Nios II MMU

Workaround

None.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Error Running Nios II Project: ‘Downloading ELF Process failed’

If the Nios II processor’s cpu.data_master port is not connected to all program memories (memories to which the **.elf** file is downloaded) the software project fails to run on Nios II hardware.

Failure to connect cpu.data_master to all program memories is a design error that the Nios II SBT does not detect.

Workaround

Connect cpu.data_master to all program memories.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

BSP Editor Does Not Show Command-Line Help

The following commands in the Nios II Command Shell should display command-line help:

- `nios2-bsp-editor --help`
- `nios2-bsp-editor --extended-help`

Instead, these commands launch the BSP Editor.

Workaround

For help with the BSP Editor, refer to "Using the BSP Editor" in the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

BSP Not Updated for Memory Size Changes in SOPC Builder

If you change the size of a memory in SOPC Builder, the memory region size in any previously created BSP is no longer correct. Regenerating the BSP does not update the BSP's memory region size.

Workaround

In the BSP Editor, on the **Linker Script** tab, you can correct the memory region size either of the following ways:

- Edit the memory region size manually.
- Click **Restore Defaults** to rerun the default Tcl script. The default Tcl script reads the updated memory region size from the SOPC information file (**.sopcinfo**) and updates the BSP.

For detailed information about keeping your BSP consistent with changes in the underlying SOPC Builder system, refer to "Revising Your BSP" in the *Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook*.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Build Errors on Interrupt API Calls

You might observe build errors if your application project installs an ISR that calls the `alt_irq_register()` function.

The Nios II EDS version 9.1 includes support for an optional External Interrupt Controller (EIC) interface on the Nios II processor. To support the EIC interface, the HAL includes an enhanced interrupt API.

When you create a BSP, the SBT determines which interrupt API(s) the device drivers in your system support. If all registered drivers in the BSP identify themselves as supporting the enhanced API, the SBT implements the enhanced API. All Altera device drivers in version 9.1 and later support the enhanced API.

When the enhanced API is implemented, the legacy interrupt API, including `alt_irq_register()`, is not available. Therefore, if the application code contains a call to a legacy API function, a linker error occurs.

This issue might affect application projects that call `alt_irq_register()` or other legacy interrupt API functions directly, rather than depending on drivers in the BSP for interrupt support.

If your Nios II application project's source code registers an ISR using `alt_irq_register()`, or includes an ISR, your application might fail to compile in the Nios II EDS version 9.1, due to undefined references to `alt_irq_register()`, or to a change to the ISR function prototype.

Workaround

Modify your application code to use the new enhanced interrupt API. To use the enhanced API, you must modify several function calls and all ISR function prototypes.

For information about supporting the enhanced interrupt API, refer to the *Exception Handling* chapter of the *Nios II Software Developer’s Handbook*, or to *AN 595: Vectored Interrupt Controller Usage and Applications*.

You must upgrade to the enhanced API if you wish to use an EIC in your hardware design to accelerate interrupt response time. The enhanced API also works with the Nios II processor’s internal interrupt controller.

Solution Status

Not fixed

SBT Fails if Nios II EDS is Installed in a Path Containing Spaces

The Nios II Command Line SBT fails if you install the Nios II EDS in a path containing spaces.

Workaround

Reinstall the Nios II EDS to a path that does not contain spaces.

Solution Status

Not fixed

User-Managed BSP Settings Not Supported in Nios II IDE

This issue is described on page 37.

Peripheral Errata

This section describes in detail the Nios II EDS issues related to peripheral components.

System Timestamp for System ID Peripheral Is Always 0

When the System ID Peripheral is not added to the top level of Qsys system, the system timestamp for the System ID peripheral is always 0.

Workaround

Remove the System ID Peripheral from any of the Qsys subsystems if you have them and add the System ID Peripheral only in the top level of Qsys system.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Unaligned Transfers of Small Payloads Fail on SG-DMA

The Scatter Gather DMA SOPC Builder peripheral does not correctly handle unaligned transfers with small payloads. A payload length smaller than the data width causes erroneous data transfers.

Workaround

Avoid using DMA devices to transfer small payloads.

If absolutely necessary, for a 32-bit SG-DMA, a minimum length of 4 bytes guarantees that data is transferred correctly.

Solution Status

Not fixed

DMA Controller Always Busy in Burst Mode

The DMA controller component (`altera_avalon_dma`), when enabled for burst transactions, does not perform transfers at widths other than its full data width. The DMA controller is always busy.

Workaround

When bursting is enabled, the DMA controller must be programmed to perform transactions at its full data width.

Solution Status

Not fixed

Non-System-Wide Reset Can Cause Improper Initialization of Mailbox Core

The `altera_avalon_mailbox` peripheral might not be initialized properly when a soft (non-system-wide) reset occurs. In this condition, mailbox contents (read and write pointers) are not reinitialized and might show potentially stale data.

Workaround

Ensure that a system-wide reset event occurs by asserting the `reset_n` input to the SOPC Builder system containing the mailbox. This resets all peripherals and Nios II processors in the system.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Target Software Errata

This section describes in detail the Nios II EDS issues related to target software packages.

EIC-Based Nios II System Crashes With NicheStack

When you attempt to run software based on the NicheStack TCP/IP Stack - Nios II Edition on a hardware design incorporating an external interrupt controller such as the VIC, the Nios II system crashes during initialization, with unpredictable results.

Workaround

There is no workaround.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

stdio Does Not Work with MicroC/OS-II and Small C Library

`stdin`, `stdout`, and `stderr` do not work in MicroC/OS-II applications built with the Small C library option.

Workaround

Disable the small C library option.

Solution Status

Not fixed

cout From MicroC/OS-II Task Does Not Send Data to stdout

If neither `printf()` or `cout` is used from `main()` before tasks are started, `cout` does not work from a task.

Workaround

Add the following C++ code to the beginning of `main()`:

```
std::ios_base::sync_with_stdio(false);
```

Solution Status

Not fixed

Toolchain and Utilities Errata

This section describes in detail the Nios II EDS issues related to the Nios II and GNU compiler toolchains, including tools such as **gcc**, **gdb**, and **sof2flash**.

nios2-download Cannot Find JTAG Instance IDs

Because Qsys uses different instance naming conventions compared to SOPC Builder, if you use the `--jdi` and `--cpu_name` flags with the **nios2-download** command, **nios2-download** cannot properly parse the Qsys instance names to find the correct JTAG debug module instance ID in the **.jdi** file.

Design Impact

Qsys designs with multiple JTAG debug modules

Workaround

Manually find the JTAG debug module ID in the **.jdi** file. Pass this value to **nios2-download with** the `--instance` flag.

☞ The JTAG debug module ID might change after each Quartus II compilation.

Solution Status

Fixed in version 11.0 of the Nios II EDS

Warning Message: “'pragma_reverse_bitfields' attribute directive ignored”

The Nios II GCC 4 tool chain does not support the C compiler pragma and flag for reverse bit fields. If you try to use the compiler pragma `reverse_bitfields`, you see the following warning:

```
'pragma_reverse_bitfields' attribute directive ignored
```

Similarly, the C compiler flag `-mreverse-bitfields` is unsupported.

Workaround

Use the GCC 3 tool chain.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

nios2-console Does Not Work in the GCC 4 Command Shell

If you try to run the **nios2-console** command in the GCC 4 Command Shell, you see the following error message:

```
Can't locate strict.pm in @INC ...
```

Workaround

Use the GCC 3 Command Shell.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

C++ Exceptions Unsupported in a Multi-Threaded Environment

C++ exceptions are only supported in a single-threaded environment. They cannot be used in a multi-threaded environment such as MicroC/OS-II.

Also see "C++ Exceptions Unsupported by the Nios II GCC 4 Toolchain".

Workaround

In a multi-threaded environment, you need to guard C++ exceptions with semaphores.

Solution Status

Not fixed

C++ Exceptions Unsupported by the Nios II GCC 4 Toolchain

With the GCC 4 toolchain, C++ exceptions are not supported.

Also see "C++ Exceptions Unsupported in a Multi-Threaded Environment".

Workaround

If your single-threaded project requires C++ exceptions, use the GCC 3 toolchain.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

memcpy() Optimization Misalignment

When optimization is turned on (`-O1` or higher), if you use `memcpy()` and the source pointer is aligned to a 32-bit boundary, the compiler implements `memcpy()` with word-oriented instructions as part of the optimization process. This optimization technique causes unexpected results in your software if `memcpy()` is used on a misaligned address.

Workaround

Take steps to ensure that an optimized implementation of `memcpy()` is called only with aligned data pointers.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

GNU Assembler Does Not Accept the --defsym Flag

According the GNU documentation, you can set an assembler definition by using the `--defsym` flag, but it does not work in the following form: `--defsym MY_VAR=1`.

Workaround

There is no workaround.

Solution Status

Not fixed

How to Contact Altera

For the most up-to-date information about Altera products, refer to Table 7.

Table 7. Contact Information

Contact (1)	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Altera literature services	Email	literature@altera.com
Non-technical support (General)	Email	nacomp@altera.com
(Software Licensing)	Email	authorization@altera.com

Note:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

Table 8 shows the typographic conventions that this document uses.

Table 8. Typographic Conventions (Part 1 of 2)

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Indicates command names, dialog box titles, dialog box options, and other graphical user interface (GUI) labels. For example, **Save As** dialog box. For GUI elements, capitalization matches the GUI.
bold type	Indicates directory names, project names, disk drive names, file names, file name extensions, dialog box options, software utility names, and other GUI labels. For example, **\qdesigns** directory, **d:** drive, and **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Indicates document titles. For example, *AN 519: Stratix IV Design Guidelines.*
Italic type	Indicates variables. For example, *n* + 1. Variable names are enclosed in angle brackets (< >). For example, <*file name*> and <*project name*>**.pof** file.
Initial Capital Letters	Indicates keyboard keys and menu names. For example, Delete key and the Options menu.
“Subheading Title”	Quotation marks indicate references to sections within a document and titles of Quartus II Help topics. For example, “Typographic Conventions.”

Table 8. Typographic Conventions (Part 2 of 2)

Visual Cue	Meaning
`Courier type`	Indicates signal, port, register, bit, block, and primitive names. For example, `data1`, `tdi`, and `input`. Active-low signals are denoted by suffix `n`. For example, `resetn`. Indicates command line commands and anything that must be typed exactly as it appears. For example, `c:\qdesigns\tutorial\chiptrip.gdf`. Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword `SUBDESIGN`), and logic function names (for example, `TRI`).
1., 2., 3., and a., b., c., and so on.	Numbered steps indicate a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ■	Bullets indicate a list of items when the sequence of the items is not important.
☞	The hand points to information that requires special attention.
⚠ CAUTION	A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.
⚠ WARNING	A warning calls attention to a condition or possible situation that can cause you injury.
↵	The angled arrow instructs you to press Enter.
👣	The feet direct you to more information about a particular topic.

101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support
www.altera.com/support

Nios II Embedded Design Suite Release Notes and Errata

© January 2011 RN-EDS-8.0

About These Release Notes

These release notes cover versions 9.1 through 10.1 of the Altera® Nios® II Embedded Design Suite (EDS). These release notes describe the revision history and errata for the Nios II EDS.

Errata are functional defects or errors, which might cause the product to deviate from published specifications. Documentation issues include errors, unclear descriptions, or omissions from current published specifications or product documents.

For the most up-to-date errata for this release, refer to the latest version of the *Nios II Embedded Design Suite Release Notes and Errata* on the Altera website. You can refer to release notes and errata for the Nios II processor core in the *Nios II Processor* chapter of the *MegaCore IP Library Release Notes and Errata*.

Revision History

Table 1 shows the revision history for the Nios II EDS, including the Nios II C2H Compiler.

Table 1. Nios II Embedded Design Suite Revision History (Part 1 of 2)

Version	Date	Description
10.1	December 2010	Maintenance release
10.0	July 2010	■ Nios II Software Build Tools (SBT) for Eclipse ■ Updated Eclipse to version 3.5 and Eclipse C/C++ development toolkit (CDT) to version 6.0 ■ Improved Source File management ■ Supports Multi-CPU launch ■ Updated to Nios II GCC 4.1.2 tool chain ■ Supports both Nios II SBT for Eclipse and Nios II Command Shell ■ GCC 4 improves compilation time compared to GCC 3 on Windows ■ Host platform ■ Supports Windows 7 ■ Supports SUSE Linux Enterprise 11 ■ Support for SUSE Linux Enterprise 9 discontinued ■ New Nios II software design examples for Reduced Gigabit Media Independent Interface (RGMII) ■ Web Server (RGMII) ■ Simple Socket Server (RGMII) ■ Legacy hardware design examples (Stratix® II and Cyclone® II) are removed ■ Installer changes ■ Nios II EDS must be installed next to Quartus® II

Table 1. Nios II Embedded Design Suite Revision History (Part 2 of 2)

Version	Date	Description
9.1 SP2	March 2010	Maintenance release
9.1 SP1	January 2010	Maintenance release
9.1	November 2009	■ The Nios II SBT for Eclipse, an integrated development environment based on Eclipse. Recommended for all new Nios II projects. ■ The Nios II Integrated Development Environment (IDE) is available in the Nios II EDS, but not recommended for new projects. ■ The Nios II Studio is discontinued. ■ The vectored interrupt controller (VIC) core offers high-performance, low-latency interrupt handling. ■ Nios II Design Examples with memory management unit (MMU): ■ Nios II 3C120 Hardware Reference Design with MMU, Cyclone III Edition ■ Nios II 4S230 Hardware Reference Design with MMU, Stratix IV GX Edition Commercial and open-source support for embedded Linux is available for these systems. ■ The Embedded Systems Lab for DE1 for Quartus and Nios II EDS v9.1 is available for download from the Nios Community Wiki (www.nioswiki.com). ■ Tcl scripting for software packages and drivers enhanced to support callback functions.

For more information on new features and changes, refer to the *Nios II Processor Reference Handbook*, the *Nios II Software Developer's Handbook*, and the *Nios II C2H Compiler User Guide*. For information about changes to the Nios II processor core, refer to the *Nios II Processor* chapter of the *MegaCore IP Library Release Notes and Errata*.

Deprecated Features

C2H SBT Command-Line Flow

The C2H command-line flow using **nios2-c2h-generate-makefile** is deprecated as of v10.0.

Notice about Environment Variables

The `QUARTUS_ROOTDIR` and `SOPC_KIT_NIOS2` environment variables will be deprecated in a future release. If you have custom scripts that use these variables, you should be prepared to change them to be independent of these variables, or to set the environment variables yourself.

Revision Details

This section provides details about some of the Nios II EDS revisions shown in Table 1.

Nios II Software Build Tools for Eclipse

The Nios II Software Build Tools (SBT) for Eclipse is a set of plugins based on the popular Eclipse framework and the Eclipse C/C++ development toolkit (CDT) plugins. The Nios II SBT for Eclipse provides a consistent development platform that works for all Nios II processor systems. You can accomplish most Nios II software development tasks within Eclipse, including creating, editing, building, running, debugging, and profiling programs. The Nios II SBT for Eclipse is built from the ground up as a set of plugins to standard Eclipse. The Nios II SBT for Eclipse is based on the Nios II Software Build Tools, a set of powerful commands, utilities and scripts for creating applications, board support packages and software libraries, and managing build options.

Table 2 compares the Nios II Integrated Development Environment (IDE) with the Nios II SBT for Eclipse.

Table 2. Differences Between Nios II SBT for Eclipse and Nios II IDE (Part 1 of 2)

Feature	Nios II IDE	Nios II SBT for Eclipse
Types of projects	■ Application ■ Board support package (BSP)—Called a system library in the IDE ■ User Library (optional)	■ Application ■ BSP ■ User Library (optional)
Handling updates to the SOPC Builder system	Updates the BSP automatically	You must initiate the BSP update (in the Nios II menu, click Generate BSP)
Options for creating projects	Application from template. BSP is created automatically as part of application.	■ New Application and BSP from template ■ Standalone application ■ Standalone BSP ■ Standalone library
Handling updates to BSP	Entire BSP is rebuilt. This process can be time-consuming.	Smart rebuild only recompiles changed files
Compatibility with command-line flow	None	Full compatibility with the Nios II SBT in the Nios II Command Shell
Modifying BSP properties	Limited control over BSP properties through the System Library page of the Properties dialog box.	Full control over BSP properties through the BSP Editor

Table 2. Differences Between Nios II SBT for Eclipse and Nios II IDE (Part 2 of 2)

Feature	Nios II IDE	Nios II SBT for Eclipse
Nios II C2H Compiler	Supported	Not supported
Nios II Flash Programmer	Limited control over the contents of the flash image	■ Allows you to create complex flash images containing Executable and Linking Format Files (.elf), SRAM Object Files (.sof), and binary data files in specified locations ■ Allows you to save flash programmer settings for repeatable flash programming
Scripting and regression testing	Not Available	Project creation and build can be scripted using the Nios II SBT command line

Converting IDE Projects to the Nios II SBT for Eclipse

If you have a pre-existing Nios II IDE project, before you can continue development in the Nios II SBT for Eclipse, you must convert it to the SBT and import it to the Eclipse environment. To convert and import an IDE project to the Nios II SBT for Eclipse, execute the following steps:

1. Convert the IDE project to an SBT-based project with the **nios2-convert-ide2sbt** command line utility.
2. Import the converted project to the Nios II SBT for Eclipse.

For a detailed description of the conversion process, refer to "Porting Nios II IDE Projects to the Software Build Tools" in the *Using the Nios II Integrated Development Environment* appendix of the *Nios II Software Developer's Handbook*. For a description of the process of importing a project, refer to "Importing a Command-Line Project" in the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*.

Learning about the Nios II SBT for Eclipse

Refer to the following documents to help you get acquainted with the Nios II SBT for Eclipse and related tools:

- *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*
- *Nios II Hardware Development Tutorial*
- The Comprehensive Nios II Embedded Design Suite support page
- The Nios II Software Build Tools for Eclipse support page

Errata Summary

Table 3 summarizes the issues that affect the Nios II Embedded Design Suite versions 9.1 through 10.1. For a detailed description of each issue, click on the issue name.

Refer to the Altera Knowledge Database for older errata and solutions.

Table 3. Nios II EDS Errata *(1)* (Part 1 of 6)

Added or Updated	Issue	Page	Affected Version				
			10.1	10.0	9.1 SP2	9.1 SP1	9.1
	Nios II SBT for Eclipse						
Jan 2011	No .sopcinfo File Name Shown in Nios II BSP Properties Page	10	✓	—	—	—	—
	Only One Thread Visible When Debugging MicroC/OS-II Application	10	✓	—	—	—	—
	Nios II SBT for Eclipse Unable to Create New Application and BSP from Template	11	✓	—	—	—	—
	Nios II Consoles Do Not Work With Multiprocessor Project	11	✓	✓	—	—	—
	Nios II SBT for Eclipse Hangs When Project Run as Nios II Hardware	11	✓	—	—	—	—
	Run As Nios II ModelSim Does Not Work	12	✓	—	—	—	—
	Run Configuration Cannot Find Imported Custom Makefile Project	12	Fixed	✓	—	—	—
	SBT for Eclipse Creates NicheStack BSP Incorrectly	13	—	Fixed	—	—	—
	Nios II SBT for Eclipse Cannot Step into GCC 3 Library Source Code	13	—	Fixed	✓	✓	✓
	Errors Creating or Importing Software Projects	14	Fixed	✓	✓	✓	✓
	Build Errors on Software for Pre-Existing Design with SG-DMA	14	—	Fixed	✓	✓	✓
	Spurious System ID Mismatch Error	14	—	Fixed	✓	✓	✓
	Errors Debugging as Local C/C++ Application	15	—	Fixed	✓	✓	✓
	java.lang.NullPointerException Error When Running Application	15	—	Fixed	✓	✓	✓
	Update Failure When Setting Empty Properties in the Nios II Properties Page	16	—	—	—	—	Fixed
	DHCP Time-Out on EP3C120	16	—	Fixed	✓	✓	✓
	No Nios II Multiprocessor Collection Run Configuration	17	—	Fixed	✓	✓	✓
	Project Imported From Command Line Fails to Build Correctly	17	—	Fixed	✓	✓	✓
Sep 2010	The Restart Button in the Debugger Does Not Work	17	✓	✓	✓	✓	✓

Table 3. Nios II EDS Errata *(1)* (Part 2 of 6)

Added or Updated	Issue	Page	Affected Version				
			10.1	10.0	9.1 SP2	9.1 SP1	9.1
July 2010	Error Building Imported Project: 'No rule to make target'	18	✓	✓	—	—	—
	Error Building Imported Project: 'target pattern contains no %'	19	✓	✓	—	—	—
	Missing Nios II Perspective	19	✓	✓	✓	✓	✓
	Error Debugging Imported Project: 'Can't find a source file'	19	—	Fixed	✓	✓	✓
	Error Marker Persists on BSP Project After Build Error Corrected	20	✓	✓	✓	✓	✓
	Assembly Language Source File Not Found	20	—	Fixed	✓	✓	✓
	'Exclude from build' Not Supported	20	✓	✓	—	—	—
May 2010	Nios II Options Do Not Appear in Eclipse	21	✓	✓	✓	✓	✓
Nov 2009	Error Messages on Console When Debugging	21	✓	✓	✓	✓	✓
	Errors Converting Nios II IDE Multiprocessor Project	22	✓	✓	✓	✓	✓
	Stop on Startup Option in Run Configuration Has No Effect	22	✓	✓	✓	✓	✓
	Debugger Breaks in crt0.s Instead of main()	22	✓	✓	✓	✓	✓
C2H Compiler							
Jan 2011	C2H Compiler Does Not Work in the Nios II Command Shell with GCC 4	22	✓	✓	—	—	—
Jul 2008	Functions Declared Without a Return Type Are Not Supported	23	✓	✓	✓	✓	✓
Oct 2007	Pre-7.1 Systems Are Not Supported	23	✓	✓	✓	✓	✓
	--src-dir SBT Argument Does Not Work With C2H	23	✓	✓	✓	✓	✓
May 2007	Accelerator Generation Failure If Tools Are Installed in Path With Spaces	23	✓	✓	✓	✓	✓
	The C2H Compiler Regenerates an Accelerator Unnecessarily	24	✓	✓	✓	✓	✓
	Error: c2h_accelerator_base_addresses.h: No such file or directory	24	✓	✓	✓	✓	✓
	Java Heap Space Exception if Quartus II Compilation is Enabled	24	✓	✓	✓	✓	✓
	Pointer Dereferences to Volatile Types	25	✓	✓	✓	✓	✓
Dec 2006	C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File	25	✓	✓	✓	✓	✓
Jun 2006	Array Elements in Structures Do Not Copy Correctly	25	✓	✓	✓	✓	✓
	Clean Build Causes Build Failure	26	✓	✓	✓	✓	✓
	Changing Build Configurations Produces Unexpected Results	26	✓	✓	✓	✓	✓
	Hardware Accelerators Remain After Deleting the Software Project *(2)*	27	✓	✓	✓	✓	✓
	Incorrect Results From Logical or Conditional Operation With Side-Effects	27	✓	✓	✓	✓	✓
	Launch SOPC Builder Button in C2H View	28	✓	✓	✓	✓	✓

Table 3. Nios II EDS Errata *(1)* (Part 3 of 6)

Added or Updated	Issue	Page	Affected Version				
			10.1	10.0	9.1 SP2	9.1 SP1	9.1
Development Boards							
Jun 2006	Intermittent Failures While Accessing CompactFlash Card	28	✓	✓	✓	✓	✓
Documentation Issues							
Jan 2011	Incorrect information about Embedded C++	29	✓	✓	✓	✓	✓
	Missing Documentation of Interrupt API Properties	30	—	Fixed	✓	✓	✓
	Error Message on Invalid Exception or Reset Vector	30	—	—	—	—	Fixed
July 2010	Error Building Project: ‘No rule to make target’	31	✓	✓	—	—	—
	AN543 Contains Incorrect Information about Updating the Flash	32	✓	✓	✓	✓	✓
	Incorrect Information about Floating-Point Instruction Precision	32	—	Fixed	✓	✓	✓
	Eclipse CDT Features Not Supported by Nios II Plugins	33	✓	✓	✓	✓	✓
May 2010	Incorrect Information about Nested Exceptions	35	✓	✓	✓	✓	✓
Feb 2010	Valid Range of hal.log_flags is –1 to 3	35	✓	✓	✓	✓	✓
Nov 2009	NicheStack TCP/IP Stack - Nios II Edition Does Not Support TFTP	36	✓	✓	✓	✓	✓
	Error Message After Renaming Project: “Resource is out of sync with the system”	36	✓	✓	✓	✓	✓
Mar 2009	Compiler Flags for Building Custom Newlib	36	✓	✓	✓	✓	✓
Oct 2007	Nios II IDE Online Help Expand Buttons Do Not Work	37	✓	✓	✓	✓	✓
Hardware Abstraction Layer							
Jan 2011	“unused variable” Warning When Building alt_main.c	38	—	Fixed	✓	✓	✓
	Build Warnings in Performance Counter Driver Files	38	—	Fixed	✓	✓	✓
Feb 2010	Missing Structure Member Errors in alt_log_printf.c with Small JTAG UART Driver	39	✓	✓	✓	✓	✓
Hardware Example Designs							
Jan 2011	Quartus II Compilation Warnings for Nios II Stratix II 2S60 ROHS Example	39	✓	✓	✓	—	—
Software Examples							
Jan 2011	Hardware Tutorial Software Example Hangs on Some Boards	40	—	Fixed	✓	✓	✓
Jun 2006	Networking Examples	40	✓	✓	✓	✓	✓
Flash Programmer							
Feb 2010	Unable to Configure FPGA from Flash with Parallel Flash Loader	41	—	—	—	Fixed	✓
Dec 2006	elf2flash File Size Limit	41	✓	✓	✓	✓	✓
Hardware Simulation							
Jan 2011	Vectored Interrupt Controller Does Not Support VHDL Simulation Models	42	—	—	—	—	Fixed
Dec 2006	Error “UNC paths are not supported” Launching ModelSim	42	✓	✓	✓	✓	✓
Jun 2006	Uninitialized .bss Variables in Simulation	42	✓	✓	✓	✓	✓

Table 3. Nios II EDS Errata *(1)* (Part 4 of 6)

Added or Updated	**Issue**	**Page**	**Affected Version**				
			10.1	**10.0**	**9.1 SP2**	**9.1 SP1**	**9.1**
Host Platform							
Jan 2011	Nios II EDS Requires Administrative Permissions on Windows Vista and Windows 7	43	✓	✓	✓	✓	✓
Nov 2009	Cannot Launch Nios II SBT for Eclipse on Red Hat Linux	43	✓	✓	✓	✓	✓
Dec 2007	Windows/Cygwin: Nios II Processor Generation Failure	43	✓	✓	✓	✓	✓
Oct 2007	Windows Vista: Limited Support in Nios II EDS	44	✓	✓	✓	✓	✓
Nios II IDE							
Building Projects							
Mar 2009	Nios II IDE Command-Line Tools Select Wrong Workspace on Linux	45	✓	✓	✓	✓	✓
	Nios II IDE Command-Line Tools Hang on Windows	45	✓	✓	✓	✓	✓
Nov 2008	Nios II IDE Cannot Find stdio.h in Outline View	46	✓	✓	✓	✓	✓
Jul 2008	Nios II IDE Hangs With UNC Project Path	46	✓	✓	✓	✓	✓
	Build Failure with Nios II Advanced Exceptions, MMU, and MPU	47	✓	✓	✓	✓	✓
May 2007	Build Command Not Functional for BSPs Created With the Nios II SBT	47	✓	✓	✓	✓	✓
	Linker Errors with Dual-Port Memories	47	✓	✓	✓	✓	✓
	User-Managed BSP Settings Not Supported in Nios II IDE	48	✓	✓	✓	✓	✓
	URL Project Location Causes Project Creation Error	48	✓	✓	✓	✓	✓
Dec 2006	Compilation Error with Separate Exception Stack Option	48	✓	✓	✓	✓	✓
	Incorrect Stack and Heap Free Space Report	49	✓	✓	✓	✓	✓
	Nios II IDE Reports Problems Without Displaying Error in Console	49	✓	✓	✓	✓	✓
Debugging Projects							
Nov 2009	Method for Accessing MMU and MPU Registers in the Debugger	49	✓	✓	✓	✓	✓
Nov 2008	IDE Cannot Display Imported Profiling Data	50	✓	✓	✓	✓	✓
Jul 2008	Error Message when Downloading .elf File	50	✓	✓	✓	✓	✓
	Variable Casting Unsupported in ISS	50	✓	✓	✓	✓	✓
	Trace Debug Does Not Support Instruction-Related Exceptions	51	✓	✓	✓	✓	✓
Oct 2007	Trace Debug Does Not Support the JMPI Instruction	51	✓	✓	✓	✓	✓
May 2007	Cannot Locate Source Code in Driver Files Shared by Multiple Projects	51	✓	✓	✓	✓	✓
	Console Window Is Not Updated After ISS Error	51	✓	✓	✓	✓	✓
	ISS Fails on Designs Containing Triple Speed Ethernet MAC or SG-DMA Components	52	✓	✓	✓	✓	✓
	Memory Window Sets Control Register Values Incorrectly	52	✓	✓	✓	✓	✓

Table 3. Nios II EDS Errata *(1)* (Part 5 of 6)

Added or Updated	Issue	Page	Affected Version				
			10.1	10.0	9.1 SP2	9.1 SP1	9.1
Dec 2006	Programs That Interact With a Terminal Console on Windows Do Not Work	52	✓	✓	✓	✓	✓
	"Run as ModelSim" in the Nios II IDE Fails	53	✓	✓	✓	✓	✓
	The Restart Command on the Run Menu Does Not Work	53	✓	✓	✓	✓	✓
Jun 2006	Watchpoints Do Not Work on Certain Variables	53	✓	✓	✓	✓	✓
Navigating Projects							
Dec 2006	Nios II IDE Freezes While Displaying the Splash Screen	53	✓	✓	✓	✓	✓
	Internal Error When Double-Clicking on a Large Objdump File	54	✓	✓	✓	✓	✓
Jun 2006	C/C++ Scanner Does Not Support Certain C/C++ Constructs	54	✓	✓	✓	✓	✓
Nios II SBT							
Jan 2011	Nios II SBT Creates Spurious BSP for System with MMU	55	✓	—	—	—	—
July 2010	Error Running Nios II Project: 'Downloading ELF Process failed'	55	✓	✓	✓	✓	✓
May 2010	BSP Editor Does Not Show Command-Line Help	55	✓	✓	✓	✓	✓
Feb 2010	BSP Not Updated for Memory Size Changes in SOPC Builder	56	✓	✓	✓	✓	✓
Nov 2009	Build Errors on Interrupt API Calls	56	✓	✓	✓	✓	✓
Oct 2007	SBT Fails if Nios II EDS is Installed in a Path Containing Spaces	57	✓	✓	✓	✓	✓
May 2007	User-Managed BSP Settings Not Supported in Nios II IDE	48	✓	✓	✓	✓	✓
Peripherals							
Jan 2011	Vectored Interrupt Controller Does Not Support VHDL Simulation Models	42	—	—	—	—	Fixed
Jul 2008	Unaligned Transfers of Small Payloads Fail on SG-DMA	57	✓	✓	✓	✓	✓
May 2007	DMA Controller Always Busy in Burst Mode	58	✓	✓	✓	✓	✓
	Non-System-Wide Reset Can Cause Improper Initialization of Mailbox Core	58	✓	✓	✓	✓	✓
Target Software							
Jan 2011	NicheStack tcp_wake() Function Does Not Awaken All select() Sleepers	58	—	Fixed	✓	✓	✓
	malloc(), realloc() Failures With MicroC/OS-II	59	—	Fixed	✓	✓	✓
Nov 2009	EIC-Based Nios II System Crashes With NicheStack	59	✓	✓	✓	✓	✓
Dec 2006	stdio Does Not Work with MicroC/OS-II and Small C Library	60	✓	✓	✓	✓	✓
Jun 2006	cout From MicroC/OS-II Task Does Not Send Data to stdout	60	✓	✓	✓	✓	✓
Toolchain and Utilities							
Jan 2011	Warning Message: "'pragma_reverse_bitfields' attribute directive ignored"	60	✓	✓	—	—	—
	nios2-download Cannot Find JTAG Instance IDs	60	✓	—	—	—	—
Sep 2010	nios2-console Does Not Work in the GCC 4 Command Shell	61	✓	✓	—	—	—

Table 3. Nios II EDS Errata *(1)* (Part 6 of 6)

Added or Updated	Issue	Page	Affected Version				
			10.1	10.0	9.1 SP2	9.1 SP1	9.1
July 2010	C++ Exceptions Unsupported in a Multi-Threaded Environment	61	✓	✓	✓	✓	✓
	C++ Exceptions Unsupported by the Nios II GCC 4 Toolchain	61	✓	✓	—	—	—
	memcpy() Optimization Misalignment	62	✓	✓	✓	✓	✓
Feb 2010	Unable to Configure FPGA from Flash with Parallel Flash Loader	41	—	—	—	Fixed	✓
Dec 2006	GNU Assembler Does Not Accept the --defsym Flag	62	✓	✓	✓	✓	✓

Notes to Table 3:

(1) Refer to the Altera Knowledge Database for older errata and solutions.

(2) For further information, refer to the *Nios II C2H Compiler User Guide*.

Nios II SBT for Eclipse Errata

This section describes in detail the issues related to the Nios II SBT for Eclipse.

No .sopcinfo File Name Shown in Nios II BSP Properties Page

In the Nios II SBT for Eclipse, the **SOPC Info** field in the Nios II BSP properties page should display the **.sopcinfo** filename for the underlying hardware design. In the Nios II SBT for Eclipse running on the Windows operating system, this field is empty.

Design Impact

The **SOPC Info** field is a read-only field. This issue does not affect functionality.

Workaround

None needed.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Only One Thread Visible When Debugging MicroC/OS-II Application

When you use GCC 4 tool chain, the Nios II SBT for Eclipse debugs a multi-threaded application as if it were a single-threaded application.

Workaround

Use GCC 3 tool chain.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Nios II SBT for Eclipse Unable to Create New Application and BSP from Template

You should be able to create a new application and BSP from a template with the following steps:

1. Select **Nios II Application and BSP from Template**.
2. Choose your .**sopcinfo** file and select a template.
3. Click **Select an existing BSP project from your workspace.**
4. Click **Create**.
5. Choose the desired BSP options.
6. Click **Finish**.

However, if you try to create a new application and BSP with this method, the SBT creates a BSP, but no application, and the GUI becomes unresponsive.

Workaround

Perform the following steps:

1. Click **Cancel** to close the previous GUI.
2. Select **Nios II Application and BSP from Template** again.
3. Choose your .**sopcinfo** file and select a template.
4. Click **Select an existing BSP project from your workspace.**
5. From the BSP list, select the BSP created previously.
6. Click **Finish**.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Nios II Consoles Do Not Work With Multiprocessor Project

Launch groups do not work with multiple JTAG UARTs in multiprocessor projects.

When you launch a multiprocessor software project with a launch group, if the design uses multiple JTAG UARTs, multiple Nios II consoles appear, but only one of them is active.

Workaround

Launch the software projects individually.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Nios II SBT for Eclipse Hangs When Project Run as Nios II Hardware

On Windows, the Nios II SBT for Eclipse sometimes becomes unresponsive when you try to run (or debug) a project with the **Run as Nios II Hardware** (or **Debug as Nios II Hardware**) command.

Workaround

In the **Run Configuration** (or **Debug Configuration**) dialog box, on the **Target Connection** tab, click **Refresh Connections** before running (or debugging) the application. Each time you restart the SBT for Eclipse, you must perform this step for each application you wish to run or debug.

Alternatively, visit the MySupport website and request Quartus® II 10.1 patch 0.18.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Run As Nios II ModelSim Does Not Work

In the Windows operating system, if you try to run an application with the **Run As Nios II ModelSim** command, you see the following error message:

```
No such file or directory
```

Workaround

Visit the MySupport website and request Quartus II 10.1 patch 0.18.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Run Configuration Cannot Find Imported Custom Makefile Project

After you import a project using the **Import Custom Makefile for Nios II Software Build Tools Project** option, the Nios II SBT fails to recognize the imported custom makefile as a Nios II C/C++ application project. As a result, the project name does not appear in the list in the run configuration.

Workaround

1. In the **Run Configuration** dialog box, select any visible Nios II C/C++ project. The project need not be related to your custom makefile project.

   ☞ If there is no Nios II C/C++ project in your workspace, create a dummy project, such as Hello World, through the File menu, by clicking **New**, **Nios II Application and BSP from Template**.

2. Turn on **Enable browse for file system ELF file**.
3. Browse to the correct **.elf** file for your custom makefile project to complete creation of a run configuration.

Solution Status

Fixed in v10.1 of the Nios II EDS

SBT for Eclipse Creates NicheStack BSP Incorrectly

NicheStack applications encounter runtime failures if you use the following steps to create a NicheStack BSP:

1. Create a new Nios II Board Support Package with the BSP type selected as Micrium MicroC/OS-II. Notice that the HAL component generates **alt_syscall.h**.
2. Using the Nios II BSP Editor, in the **Software Packages** tab, select **altera_iniche**. Save and build the BSP. Notice that the NicheStack (iniche) component generates another **alt_syscall.h**.

public.mk includes the HAL header files before the NicheStack header files. This causes the HAL **alt_syscall.h** to be used in the build, instead of the required **alt_syscall.h** from the NicheStack component.

If a NicheStack application links to such a BSP, the application builds successfully, but encounters runtime failures that are difficult to debug.

Workaround

Manually edit **public.mk**. Under `Software Component & Driver Include Paths`, rearrange the include paths to include the NicheStack header files before the HAL header files.

In general, include paths should appear in the following order:

1. Drivers
2. Software packages
3. Extended BSP type (e.g., MicroC/OS-II)
4. Base BSP type (HAL)

Example 1 shows include path directories added in the correct order.

Example 1. Include Path Ordering in **public.mk**

```
ALT_INCLUDE_DIRS += $(ALT_LIBRARY_ROOT_DIR)/drivers/inc
ALT_INCLUDE_DIRS += $(ALT_LIBRARY_ROOT_DIR)/drivers/inc/iniche
ALT_INCLUDE_DIRS += $(ALT_LIBRARY_ROOT_DIR)/iniche/inc
ALT_INCLUDE_DIRS += $(ALT_LIBRARY_ROOT_DIR)/iniche/src/h
ALT_INCLUDE_DIRS += $(ALT_LIBRARY_ROOT_DIR)/iniche/src/h/nios2
ALT_INCLUDE_DIRS += $(ALT_LIBRARY_ROOT_DIR)/iniche/src/nios2
ALT_INCLUDE_DIRS += $(ALT_LIBRARY_ROOT_DIR)/UCOSII/inc
ALT_INCLUDE_DIRS += $(ALT_LIBRARY_ROOT_DIR)/HAL/inc
```

Solution Status

Fixed in v10.0 of the Nios II EDS

Nios II SBT for Eclipse Cannot Step into GCC 3 Library Source Code

If your project uses the Nios II GCC 3 toolchain, when debugging with the Nios II SBT for Eclipse, you might be unable to step into library source code. A message such as the following message appears:

```
Resource '/xxx/xxx.c' does not exist.
```

Workaround

Upgrade to the Nios II EDS v10.0 or later and use the Nios II GCC 4 toolchain.

Solution Status

Fixed in v10.0 of the Nios II EDS

Errors Creating or Importing Software Projects

If you place software projects in your Eclipse workspace directory, you might experience project-related errors. For example, you might see one of the following error messages:

-
  ```
  Unable to create project
  Reason: Unable to create project in workspace directory
  ```

-
  ```
  Failed to import
  Reason: Unable to import project
  ```

Workaround

Switch to a different workspace, or create a new workspace, separate from your project files. To switch workspaces or create a workspace, on the File menu, click **Switch Workspace**.

Solution Status

Fixed in v10.0 of the Nios II EDS

Build Errors on Software for Pre-Existing Design with SG-DMA

If you have a Nios II system generated with SOPC Builder v. 9.0 or earlier, and it contains the Scatter-Gather DMA (SG-DMA) component, you cannot build software for it with the Nios II SBT for Eclipse v. 9.1 or later. If you attempt to do so, the Nios II compiler reports errors.

Workaround

Regenerate the design with SOPC Builder v. 9.1 or later before attempting to compile the software in the Nios II SBT for Eclipse.

Solution Status

Fixed in v10.0 of the Nios II EDS

Spurious System ID Mismatch Error

You might see the following error message when attempting to run or debug a Nios II software project, or to edit a launch configuration:

```
[Target Connection]:System ID mismatch - actual: "0xffffffff", expected: "<system id>"
```

<system id> is replaced by the system ID in your software project. When this error occurs, you cannot run or debug software.

This error normally means that the FPGA is unconfigured, or configured with a **.sof** file that does not correspond to your software project. However, the error occasionally appears even when the FPGA is configured with the correct **.sof**.

Workaround

After verifying that the FPGA is configured with the correct **.sof**, you can use one of the following methods to work around this issue:

- In the **Run Configuration** or **Debug Configuration** dialog box, on the **Target Connection** tab, click **Refresh Connections** repeatedly until the error message disappears.
- Delete the launch configuration and create a new launch configuration.

Solution Status

Fixed in v10.0 of the Nios II EDS

Errors Debugging as Local C/C++ Application

If you try to debug a Nios II software project as the Local C/C++ Application launch configuration type, you see one of the following errors, depending on your platform:

- Windows: "Error creating session"
- Linux: "Execution is suspended because of error"

The Nios II debug perspective fails to open.

This is expected CDT behavior in the Eclipse platform. Local C/C++ Application is the launch configuration type for a standard CDT project. To invoke the Nios II plugins, you must use a Nios II launch configuration type.

Workaround

Always debug Nios II software projects as the Nios II Hardware launch configuration type.

Solution Status

Fixed in v10.0 of the Nios II EDS

java.lang.NullPointerException Error When Running Application

When you attempt to run or debug an application, the Nios II SBT for Eclipse might show an internal error message similar to the following:

```
Launching New_configuration java.lang.NullPointerException
```

This error occurs in either of the following circumstances:

- You attempt to run your project without rebuilding after performing a `make clean`.
- You delete a run configuration belonging to your project while the project is running.

You must restart Nios II SBT for Eclipse to recover from this error.

Workaround

Build the project before attempting to run or debug it. Do not delete a run configuration while it is in use.

Solution Status

Fixed in v10.0 of the Nios II EDS

Update Failure When Setting Empty Properties in the Nios II Properties Page

If you use the Nios II Properties page to update more than one board support package (BSP) setting to the empty string (""), the BSP is not updated.

Workaround

You can use either of the following workarounds to set multiple BSP settings to the empty string:

- Use the keyword none instead of "" to set settings to the empty string.
- Use the BSP Editor instead of the Nios II Properties page.

Solution Status

Fixed in v9.1 of the Nios II EDS

DHCP Time-Out on EP3C120

If you create a project based on the web server template targeting the Altera Embedded Systems Development Kit, Cyclone® III Edition (EP3C120) board, the system is unable to get an IP address at startup. DHCP times out and the system reverts to the default IP address.

Workaround

Create the project in the Nios II Command Shell and then import it to the Nios II SBT for Eclipse, by executing the following steps:

1. In the Nios II Command Shell, navigate to the following path:

   *<Nios II EDS install path>***/examples/***<language>***/niosII_cycloneIII_3c120/tserd/software_examples/app/web_server_tse_3c120_rgmii**

   where *<language>* is either **vhdl** or **verilog**.

2. Type the following command to create a web server example for the EP3C120 kit:

   ```
   create-this-app↵
   ```

3. Import the command-line project to the Nios II SBT for Eclipse.

   For information about importing command-line projects, refer to "Importing a Command-Line Project" in the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*.

Solution Status

Fixed in v10.0 of the Nios II EDS

No Nios II Multiprocessor Collection Run Configuration

The multiprocessor collection run configuration is not available in the Run menu. Therefore, you cannot use a single run configuration to download Executable and Linking Format Files (**.elf**) to run on multiple Nios II processors in the Nios II SBT for Eclipse.

Workaround

Create a separate run configuration for each Nios II processor in the hardware design, and launch them one at a time

Solution Status

Fixed in v10.0 of the Nios II EDS

Project Imported From Command Line Fails to Build Correctly

If you try to import a command-line project to the Nios II SBT for Eclipse, and your project has source files located outside the project folder, the Nios II SBT for Eclipse fails to build your project because the makefile does not include source files located outside the project folder.

Workaround

You can work around this issue with either of the following methods:

- When importing the project, disable source management through the **Nios II Application Properties** or **Nios II Library Properties** dialog box.

  ☞ You must manually keep your makefile updated to reflect source files added to or removed from the project.

- Manually link the affected source files to the project. Right-click the project, click **Properties**, expand **C/C++ General**, select **Paths and Symbols** and then **Source Location**, and click **Link Folder**.

Solution Status

Fixed in v10.0 of the Nios II EDS

The Restart Button in the Debugger Does Not Work

When debugging in the Nios II SBT for Eclipse, if you click the **Restart** icon, execution is suspended, with the following error message:

```
Don't know how to run. Try "help target."
```

Workaround

Terminate the program, download it again, and start the debugger.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Error Building Imported Project: ‘No rule to make target’

If an application or library project is created with v9.1 SP2 or earlier (using the GCC 3 toolchain), and you import it to the v10.0 Nios II SBT for Eclipse using the GCC 4 toolchain, you might see the build error shown in Example 2.

Example 2.

```
make all
Info: Building ../hw_bsp
make --no-print-directory -C ../hw_bsp
make[1]: *** No rule to make target `/cygdrive/c/tmp/
verilog_niosII_stratixII_2s60_standard/software/hw_bsp/
alt_sys_init.c', needed by `all'. Stop.
make: *** [../hw_bsp-recurs-make-lib] Error 2
```

This error can also happen if you import a GCC 3 project to the v10.0 SBT for Eclipse using the GCC 3 toolchain, and later change the toolchain to **MinGW Nios II GCC4**.

This issue only affects platforms running the Windows operating system.

Workaround

To avoid this error, follow these steps:

1. Edit the application project and library project makefiles and add the italicized code as shown in Example 3.

Example 3.

```
#------------------------------------------------------------------------------
# The adjust-path macro
#
# If COMSPEC is defined, Make is launched from Windows through
# Cygwin. This adjust-path macro will call 'cygpath -u' on all
# paths to ensure they are readable by Make.
#
# If COMSPEC is not defined, Make is launched from *nix, and no adjustment
# is necessary
#------------------------------------------------------------------------------

# Beginning of inserted code
ifndef COMSPEC
ifdef ComSpec
COMSPEC = $(ComSpec)
endif # ComSpec
endif # !COMSPEC
#end of inserted code

ifdef COMSPEC
adjust-path = $(shell cygpath -u "$1")
adjust-path-mixed = $(shell cygpath -m "$1")
else
adjust-path = $1
adjust-path-mixed = $1
endif
```

2. Right-click the BSP project, point to **Nios II** and click **Generate BSP**.
3. Clean and rebuild the application and library projects.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Error Building Imported Project: ‘target pattern contains no %’

If your application or library makefile uses an absolute path and is generated with the GCC 3 toolchain, and you import it to the v10.0 Nios II SBT for Eclipse using the GCC 4 toolchain, you get the following build error:

```
makefile:587: *** target pattern contains no '%'. Stop.
```

Workaround

Recreate the project using the SBT for Eclipse v10.0.

Alternatively, use the GCC 3 toolchain.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Missing Nios II Perspective

If you launch the Nios II SBT for Eclipse from a universal naming convention (UNC) path, the Nios II perspective does not appear in Eclipse.

This issue appears only on platforms running the Windows operating system.

Workaround

Do not use a UNC path to launch the Nios II SBT for Eclipse. For example, if the SBT for Eclipse is installed on a network drive, map the network path to a Windows drive letter.

Solution Status

Not fixed

Error Debugging Imported Project: ‘Can't find a source file’

If you create a project using the Nios II SBT in the Nios II Command Shell, and then import the project to Nios II SBT for Eclipse for debugging, the debugger cannot find source files located outside of project folder.

Example of error message:

```
Can't find a source file at "../src/test.c"
```

Workaround

To link the affected source files to the project, follow these steps:

1. Right-click the project and click **Properties** to open the **Properties** dialog box.
2. Click **+** to expand **C/C++ General**, and click the **Paths and Symbols** tab.
3. Click the **Source Location** tab.
4. Click **Link Folder** and navigate to the folder containing the source file.

Solution Status

Fixed in v10.0 of the Nios II EDS

Error Marker Persists on BSP Project After Build Error Corrected

In the Nios II SBT for Eclipse, if an application project fails to build, error markers appear in the BSP project, if the build error message contains a path to BSP project source files. Even after you have rebuilt the application project without errors, the error markers persist in the BSP project.

Workaround

Rebuild your BSP project.

Alternatively, if your application will always be associated with the same BSP, set the Eclipse project references by following these steps:

1. Right-click your application project, point to **Properties** and click **Project References**.
2. Select the desired BSP for your application.
3. Build your application project again to clear the error markers in the BSP project.

☞ If you later link your application project with a different BSP, you must manually correct the Eclipse project references.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Assembly Language Source File Not Found

The Nios II SBT for Eclipse does not support assembly language files with the upper case file extension **.S**. If your project includes a file with an upper case extension, the compiler is not able to find the assembly language file during build.

Workaround

Rename the assembly language files with the lower-case extension **.s**.

Solution Status

Fixed in v10.0 of the Nios II EDS

‘Exclude from build’ Not Supported

In the Nios II SBT for Eclipse v9.1 SP2 and earlier, you can use the **Exclude from build** command to omit a source file from your project at build time. Starting with v10.0, **Exclude from build** is replaced by the **Remove from Nios II build** and **Add to Nios II build** commands.

Due to an Eclipse platform limitation, **Exclude from build** still appears in the project context menu in v10.0. However, it has no effect.

Workaround

To exclude a source file from your project when it is built, on the project context menu, click **Remove from Nios II build**. To restore it to the build, click **Add to Nios II build**.

Nios II Options Do Not Appear in Eclipse

When Nios II SBT for Eclipse starts, the Nios II plugins might fail to load, resulting in the following symptoms:

- The Nios II perspective is not available
- No Nios II items appear when you click **New** in the File menu

Eclipse can behave this way if you have installed the Nios II EDS in the same directory as a previous installation.

When the Altera Complete Design Suite (ACDS) installer installs the Nios II EDS, you specify a root directory for the EDS file structure. For example, if the Nios II EDS 9.1 is installed on the Windows operating system, the root directory might be **c:\altera\91\nios2eds**. For simplicity, Altera documentation refers to this directory as <*Nios II EDS install path*>.

When the ACDS installer removes the Nios II EDS, it leaves behind some files in <*Nios II EDS install path*>. If you reinstall the Nios II EDS in the same directory, these leftover files might cause the Nios II SBT for Eclipse to work incorrectly.

Workaround

To correct this problem, execute the following steps:

1. Uninstall the Nios II EDS.
2. Delete <*Nios II EDS install path*>, including all subdirectories, before reinstalling the Nios II EDS.

   Alternatively, reinstall the Nios II EDS in a different directory.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Error Messages on Console When Debugging

When debugging, you might see the following messages in the Nios II SBT for Eclipse console:

```
No symbol "auto" in current context.
No symbol "new" in current context.
```

These are normal messages from the GNU Debugger (GDB) console, not error messages. This issue does not impact the debug process.

Workaround

No workaround needed.

Solution Status

None needed.

Errors Converting Nios II IDE Multiprocessor Project

If you try use the **nios2-convert-ide2sbt** utility to convert an IDE multiprocessor project to the Nios II SBT, error messages appear, and the project fails to generate properly.

Workaround

Create new Nios II SBT application and BSP projects using project settings equivalent to the original Nios II IDE project settings.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Stop on Startup Option in Run Configuration Has No Effect

Turning on **Stop on Startup at** in the **Debugger** tab of a run configuration has no effect.

Workaround

You can manually set any breakpoint in the debug perspective.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Debugger Breaks in crt0.s Instead of main()

If you attempt to launch a debug session by right clicking the project, without first creating a run configuration, the debugger breaks in the initialization code in **crt0.s**, rather than at the top of the `main()` function. You must explicitly create the run configuration before launching the debug session.

Workaround

You can work around this issue with either of the following methods:

- Create the run configuration with the **Run Configuration** dialog box. You can then debug with this run configuration by right-clicking the project, clicking **Debug As**, and selecting the debug configuration. The program correctly breaks at `main()`.
- Manually set a breakpoint at `main()`.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

C2H Compiler Errata

This section describes in detail the issues related to the Nios II C2H Compiler.

C2H Compiler Does Not Work in the Nios II Command Shell with GCC 4

The Nios II Command Shell with GCC 4 does not support the C2H Compiler.

Workaround

Use the Nios II Command Shell with GCC 3.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Functions Declared Without a Return Type Are Not Supported

The C2H compiler does not support functions without an explicitly declared return type.

Workaround

If you are using the implicit `int` return type, declare the return type explicitly. If your function has no return value, declare it as `void`.

Solution Status

Not fixed

Pre-7.1 Systems Are Not Supported

Starting in v. 7.1, the C2H Compiler does not support SOPC Builder systems created in SOPC Builder 7.0 or earlier (systems based on a **class.ptf** file).

Workaround

Before using the C2H Compiler, you must upgrade your SOPC Builder system. Open the system in SOPC Builder v. 7.1 or later. SOPC Builder prompts you to convert the system, creating a ***_hw.tcl** file compatible with the C2H Compiler.

Solution Status

Not fixed

--src-dir SBT Argument Does Not Work With C2H

When using the C2H Compiler with Nios II SBT, an error occurs when using the `nios2-app-generate-makefile` argument `--src-dir` to specify a directory of source files. An example of the error is:

```
multiple definition of 'my_accelerated_function'
```

Workaround

Specify source files individually using the `--src-files` argument.

Solution Status

Not fixed

Accelerator Generation Failure If Tools Are Installed in Path With Spaces

If the path to your installation of the Altera Design Suite contains spaces, the C2H Compiler fails to generate the accelerators.

Workaround

Reinstall the tools to a path containing no spaces.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

The C2H Compiler Regenerates an Accelerator Unnecessarily

The C2H Compiler might regenerate an accelerator even when the accelerated function is unchanged. This problem can result from changes to files included by the C file containing the accelerated function. The C2H Compiler fails to check that the generated hardware description language (HDL) matches the previously generated HDL causing the system to be regenerated.

Workaround

To avoid this issue move the accelerator-specific information from the include file to a separate include file. This workaround prevents regeneration of the system when the HDL is unchanged.

Solution Status

Not fixed

Error: c2h_accelerator_base_addresses.h: No such file or directory

When a C2H accelerator is compiled for the first time, the following compile-time error can result if the **Analyze all accelerators** option is selected:

```
c2h_accelerator_base_addresses.h: No such file or directory.
```

Workaround

Click **Build software and generate SOPC Builder system** and build once before building with the **Analyze all accelerators** option.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Java Heap Space Exception if Quartus II Compilation is Enabled

If your design contains a C2H accelerator, and you select **Build software, generate SOPC Builder system, and run Quartus II compilation** in the C2H view, you might see the following error during Quartus II compilation:

```
Exception in thread "main" java.lang.OutOfMemoryError: Java
heap space

make: *** [c2h_hdl-t] Error 1
```

Workaround

Select the **Build software and generate SOPC builder system** option in the C2H Compiler settings window, and then manually launch the Quartus II software to compile the design.

Solution Status

Not fixed

Pointer Dereferences to Volatile Types

The C2H Compiler treats pointer dereferences to a volatile type as if they alias all other pointer dereferences. Pointers that are restrict-qualified are treated the same way.

The two loops in Example 4 cannot be scheduled concurrently because the `volatile` qualification overrides the `__restrict__` pragma.

Example 4. Non-Concurrent Loops

```
volatile int * __restrict__ fifo_rd = FIFO_RD_BASE;
volatile int * __restrict__ fifo_wr = FIFO_WR_BASE;
for ()
{
  *fifo_wr = ....;
}
for ()
{
  ... = *fifo_rd;
}
```

Workaround

Divide the function into multiple interrupt request (IRQ)-enabled accelerators that are launched concurrently from the processor, and use FIFO buffers to communicate between them.

Solution Status

Not fixed

C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File

When accelerating a function in a file, the C2H Compiler cannot link subfunctions that are defined in a different file.

Workaround

Include all subfunctions called by the accelerated function within the same source code file.

Solution Status

Not fixed

Array Elements in Structures Do Not Copy Correctly

C2H accelerators do not correctly copy array elements that are elements of structures.

In Example 5, the `a` and `b` elements of the structure copy correctly, but the `buf` element does not. After this assignment, `struct_a` equals `{9, 8, {3, 3, 3, 3}}`.

Example 5. Array Elements of Structs

```
typedef struct my_struct {
  int a;
  int b;
  int buf[BUF_SIZE];
}MY_STRUCT;
MY_STRUCT struct_a = {1, 2, {3, 3, 3, 3}};
MY_STRUCT struct_b = {9, 8, {7, 7, 7, 7}};
struct_a = struct_b;
```

Workaround

Copy the array elements explicitly, as shown in Example 6.

Example 6. Copying Array Elements Explicitly

```
{
  int i=0;
  do
    {
      struct_a.buf[i] = struct_b.buf[i];
      i++;
    } while (i<LENGTH_OF_BUF_ELEMENT)
}
```

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Clean Build Causes Build Failure

Performing a clean build on a Nios II IDE project that contains a hardware accelerator can cause the next build to fail in the IDE, because the clean build erroneously deletes a file required by the C2H Compiler.

Workaround

Do not perform a clean build on projects that use hardware accelerators. If you have already performed a clean build, recompile with option **Build software, generate SOPC Builder system, and run Quartus II compilation** to regenerate the necessary files.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Changing Build Configurations Produces Unexpected Results

The C2H Compiler does not support multiple build configurations (for example Release or Debug) in the Nios II IDE. After creating one or more accelerators in a particular configuration, the C2H Compiler produces undefined results if you switch to a different build configurations and create more accelerators.

Workaround

For a specific SOPC Builder system and Nios II IDE project, specify C2H accelerators in only one build configuration. You can use multiple build configurations, as long as only one configuration specifies C2H Compiler settings.

Solution Status

Not fixed

Hardware Accelerators Remain After Deleting the Software Project

If a system contains C2H accelerators, deleting the software project that defines the accelerators does not remove the accelerators from the hardware system, and the accelerator logic remains in the SOPC Builder system.

Workaround

To remove an accelerator from a system, delete the accelerator from the C2H view in the Nios II IDE first, and then recompile the software project. The C2H Compiler then removes the accelerator from the SOPC Builder system. Once the compilation is complete then the software application can be deleted from the workspace.

Solution Status

Not fixed

Incorrect Results From Logical or Conditional Operation With Side-Effects

The C2H Compiler always evaluates both operands of logical (&&, `||`) and conditional (`?:`) operators. This is different from expected American National Standards Institute (ANSI) C behavior, where operands are evaluated left-to-right, and unnecessary operands are skipped.

For example, in the expression `(i-- && j--)`, if the value of `i` is zero, ANSI C does not evaluate the right-hand-side (RHS) expression, and `j` is not decremented. By contrast, the C2H Compiler evaluates both sides, decrementing `j`.

The following expressions are other examples that might be affected by this issue:

```
if (i-- || j++)
    ...;


a = ((cond == 1)?  i++ : j++);
```

Workaround

Use logical and conditional operations whose operators have no side effects. Operations with side effects include pre- and post- increment and decrement operations (`++`, `--`), memory operations (`*`, `[]`, ., `->`), and function calls.

Solution Status

Not fixed

Launch SOPC Builder Button in C2H View

When the Nios II IDE workspace contains multiple projects with multiple system libraries, the incorrect SOPC Builder system might open when you click **Launch SOPC Builder** in the C2H view.

Workaround

Launch SOPC Builder from the Quartus II software. Alternatively, keep only one system library project open at a time while using the C2H Compiler.

Solution Status

Not fixed

Development Board Errata

This section describes in detail the Nios II EDS issues related to Nios development boards.

Intermittent Failures While Accessing CompactFlash Card

The Nios II Development Kit version 5.0 and later includes a CompactFlash controller peripheral suitable for interfacing to CompactFlash cards in True IDE mode on Nios development boards. For True IDE mode to operate, CompactFlash cards require that the ATASEL_N input be driven to ground during power-up.

The CompactFlash controller peripheral includes a configurable power register, used to cycle power to CompactFlash cards from Nios II software through a metal oxide semiconductor field-effect transistor (MOSFET) on the Nios development boards. However, in certain development boards, power to the CompactFlash card does not turn off completely during this power cycle operation. Because of this condition, the CompactFlash might not sample the ATASEL_N pin during the power-cycle operation after FPGA configuration when this pin is driven to ground. Instead, the CompactFlash card might sample the ATASEL_N pin when power is first applied to the development board, when I/O is not yet driven by the FPGA (before FPGA configuration).

Workaround

If you encounter errors with CompactFlash when using the Nios development boards, try one of the following solutions:

- Use a different CompactFlash card. Certain cards are more susceptible to the power-cycling issue than others.
- Modify the Nios development board. This is recommended if you are familiar and comfortable with board-level modifications. Disconnect pin 9 (ATASEL_N) on the CompactFlash socket on your Nios development board and tie this pin to ground.

☞ The CompactFlash socket uses a staggered numbering on the pins (starting from pin 1: 1, 26, 2, 27, ...); refer to the CompactFlash Association specification for right-angle surface-mount connectors for exact specifications on this connector. This modification permanently enables True IDE mode operation.

Solution Status

Not fixed

Documentation Errata

This section describes in detail the Nios II EDS documentation issues.

Incorrect information about Embedded C++

The *Nios II Software Developer's Handbook* contains the following incorrect statement about C++ support:

> The HAL supports only the standard Embedded C++ subset of the full C++ language. C++ programs that use features beyond this subset fail in the HAL environment. C++ features not available in Embedded C++ include polymorphism, templates, and single and multiple object inheritance. In general, features that consume a large amount of memory are not included in Embedded C++. Catch/throw exceptions fail in the MicroC/OS-II environment.

Nios II C++ support is not restricted to the obsolete Embedded C++ specification.

Workaround

In place of the incorrect paragraph, refer to the following correct information:

Nios II C++ language support depends on the GCC tool chain. The Nios II GCC 4 C++ tool chain supports the following features:

- Polymorphism
- Friendship and inheritance
- Multiple inheritance
- Virtual base classes
- Run-time type information (`typeid`)
- The `mutable` type qualifier
- Namespaces
- Templates
- New-and-delete style dynamic memory allocation
- Operator overloading
- Standard Template Library (STL)

Exceptions and new-style dynamic casts are not supported.

Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

Missing Documentation of Interrupt API Properties

“Tcl Commands” in the *Nios II Software Build Tools Reference* chapter of the *Nios II Software Developer’s Handbook* does not include information about two `set_sw_property` command properties related to the application programming interfaces (APIs) for interrupts.

Workaround

The missing information belongs with the `set_sw_property` command, in “Tcl Commands” in the *Nios II Software Build Tools Reference* chapter of the *Nios II Software Developer’s Handbook*.

The missing information follows:

- `supported_interrupt_apis`—Specifies the interrupt API that the device driver supports. The Nios II SBT analyzes this property for each driver in the system to determine the appropriate API to be used in the system.

  Specify `legacy_interrupt_api` if the device driver supports the legacy API only or `enhanced_interrupt_api` if the device driver supports the enhanced API only. Specify both using a quoted list if the device driver supports both APIs.

  If you do not specify which API your device driver supports, the Nios II SBT assumes that only the legacy interrupt API is supported.

  This property is only available for device drivers.

  For more information about the legacy and enhanced APIs, refer to “Exception Handling” in the *Nios II Software Build Tools* chapter of the *Nios II Software Developer’s Handbook*.

- `isr_preemption_supported`—Specify `true` if your device driver’s interrupt service routine (ISR) can be preempted by a higher priority ISR. If you do not specify whether the driver supports preemption, the Nios II SBT assumes that your device driver does not support preemption. If your driver does not have an ISR, but the associated device has an interrupt port, you can set this property to `true`.

  This property is valid for operating systems and device drivers.

Solution Status

Fixed in v10.0 of the Nios II EDS documentation

Error Message on Invalid Exception or Reset Vector

Nios II compiler error messages, added in v 8.1, are not documented in the *Nios II Software Developer’s Handbook*.

Workaround

The missing information is as follows:

While building a Nios II software project, the compiler produces an error message if the SOPC Builder system defines an invalid exception or reset vector. Table 4 shows possible error messages and their causes.

In the SOPC Builder system, ensure that the exception and reset vectors conform to all the criteria listed in Table 4.

Solution Status

Fixed in v9.1 of the Nios II EDS documentation

Table 4. Exception and Reset Vector Error Messages

Message	Cause
`The section mapping ".entry" is not mapped to a memory region.`	The `.entry` section is mapped to a nonexistent memory region.
`The section mapping ".exceptions" is not mapped to a memory region.`	The `.exceptions` section is mapped to a nonexistent memory region.
`The section mapping ".entry" maps to a memory region "<`*`region name`*`>" which is not the required length "32".`	The `.entry` section is mapped to a memory region that is smaller than 32 bytes.
`The section mapping ".entry" maps to a memory region "<`*`region name`*`>" which does not start at the SOPC Design specified address "<`*`base address`*`>".`	The `.entry` section is mapped to a memory region that does not start at the reset vector base address.
`The section mapping ".exceptions" maps to a memory region "<`*`region name`*`>" which does not start at the SOPC Design specified address "<`*`base address`*`>".`	The `.exceptions` section is mapped to a memory region that does not start at the exception vector base address.
`The SOPC design Nios II reset and exception vector addresses are mapped to the same memory. For HAL the exception address must be at least 32 bytes larger than the reset address.`	The `.entry` section and `.exceptions` section map to the same memory device, and the base of the `.exceptions` section is less than 32 bytes above the base of the `.entry` section.
`The section mapping ".exceptions" mapped device "<`*`memory 1`*`>" is not the same as the SOPC Design specified device "<`*`memory 2`*`>".`	The `.exceptions` section is mapped to a device that is different from the exception vector device specified in the SOPC Builder system.
`The section mapping ".entry" mapped device "<`*`memory 1`*`>" is not the same as the SOPC Design specified device "<`*`memory 2`*`>".`	The `.entry` section is mapped to a device that is different from the reset vector device specified in the SOPC Builder system.

Error Building Project: 'No rule to make target'

The *Getting Started with Graphical User Interface* and *Getting Started from the Command Line* chapters of the *Nios II Software Developer's Handbook* do not include the following information:

You cannot link Nios II GCC 3 projects with Nios II GCC 4 projects. Your application, library and BSP projects must all use the same Nios II GCC toolchain.

If you switch between Nios II GCC 3 and Nios II GCC 4, make sure you run **make clean** on your application, library and BSP projects before rebuilding.

When importing a GCC 3 BSP, such as a project created with v9.1 SP2 or earlier, to v10.0 SBT for Eclipse using the GCC 4 toolchain, after importing the BSP, regenerate the makefile.

If you do not follow these rules, you might see the following error:

```
make[1]: *** No rule to make target `/cygdrive/c/.../bsp/
alt_sys_init.c', needed by 'all'. Stop.
make: *** [../bsp/-recurs-make-lib] Error 2
```

Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

AN543 Contains Incorrect Information about Updating the Flash

"Updating the Flash Option Bits on the Cyclone III Development Board" in AN543, *Debugging Nios II Software Using the Lauterbach Debugger* contains incorrect information about updating the flash.

Workaround

The correct information is as follows:

Updating the Flash on the Cyclone III Development Board

To restore the flash factory image, refer to "Restoring the Factory Design to the Flash Device" in the *Cyclone III Development Kit User Guide*.

Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

Incorrect Information about Floating-Point Instruction Precision

In "Arithmetic Logic Unit" in the *Processor Architecture* chapter, the *Nios II Processor Reference Handbook* incorrectly states that the Nios II compiler treats floating-point constants as double-precision numbers.

Workaround

The correct information is as follow:

All the floating-point custom instructions are single-precision operations. Double-precision floating-point operations are implemented in software.

When the floating-point custom instructions are not present, the Nios II compiler treats floating-point constants as double-precision values. However, with the floating-point custom instructions, the Nios II compiler treats floating-point constants as single-precision numbers by default. This allows all floating-point expressions to be evaluated in hardware, at a possible cost in precision.

If you do not wish floating-point constants to be cast down to single precision values, append L to each constant value, to instruct the compiler to treat the constant as a double-precision floating-point value. In this case, if an expression contains a floating-point constant, each term in the expression is cast to double precision. As a result, the expression is computed with software-implemented double-precision arithmetic, at a possible cost in computation speed.

Table 5 shows code examples using floating-point constants, indicating how each computation is implemented.

Table 5. Floating-Point Constant Examples

Example Code	Floating-Point Custom Instructions Present?	Precision	Implementation
`b = a × 4.67`	No	Double	Software
`b = a × 4.67f`	No	Single	Software
`b = a × 4.67L`	No	Double	Software
`b = a × 4.67`	Yes	Single	Hardware
`b = a × 4.67f`	Yes	Single	Hardware
`b = a × 4.67L`	Yes	Double	Software

Solution Status

Fixed in v10.0 documentation

Eclipse CDT Features Not Supported by Nios II Plugins

The following information is missing from the *Getting Started with Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*:

The features listed in the left column of Table 6 are supported by the Eclipse CDT plugins, but are not supported by Nios II plugins. The right column lists alternative features supported by the Nios II plugins.

Table 6. Eclipse CDT Features Not Supported by the Nios II Plugins (Part 1 of 2)

Unsupported CDT Feature	Alternative Nios II Feature
New Project Wizard	
C/C++ ■ C Project ■ C++ Project ■ C/C++ ■ Convert to a C/C++ Project ■ Source Folder	To create a new project, use one of the following Nios II wizards: ■ Nios II Application ■ Nios II Application and BSP from Template ■ Nios II Board Support Package ■ Nios II Library

Table 6. Eclipse CDT Features Not Supported by the Nios II Plugins (Part 2 of 2)

Unsupported CDT Feature	Alternative Nios II Feature
Build configurations	
Right-click project and point to **Build Configurations**	The Nios II plugins only support a single build configuration.
Exclude from Build (from v10.0 onwards)	
Right-click source files	Use **Remove from Nios II Build** and **Add to Nios II Build.**
	See "'Exclude from build' Not Supported" on page 20.
Project Properties	
C/C++ Build	
■ Builder Settings	
■ Makefile generation	By default, the Nios II SBT generates makefiles automatically.
■ Build location	The build location is fixed.
■ Behaviour	
■ Build on resource save (Auto build)	
■ Build Variables	
■ Discovery Options	
■ Environment	
■ Settings	
■ Tool Chain Editor	
■ Current builder	
■ Used tool	To change the toolchain, use the **Current tool chain** option
C/C++ General	
■ Enable project specific settings	
■ Documentation	
■ File Types	
■ Indexer	
■ Build configuration for the indexer	The Nios II plugins only support a single build configuration.
■ Language Mappings	
■ Paths and Symbols	Use **Nios II Application Properties** and **Nios II Application Paths**
Window Preferences	
C/C++	
■ Build scope	
■ Build project configurations	The Nios II plugins only support a single build configuration.
■ Build Variables	
■ Environment	
■ File Types	
■ Indexer	
■ Build configuration for the indexer	The Nios II plugins only support a single build configuration.
■ Language Mappings	
■ New CDT project wizard	

Incorrect Information about Nested Exceptions

"Exception Processing" in the *Programming Model* chapter of the *Nios II Processor Reference Handbook* incorrectly states that multiple interrupts with different requested interrupt levels (RILs) must not be assigned to the same shadow register set.

Workaround

The correct information is as follows:

Multiple interrupts with different RILs can be assigned to a single shadow register set. However, with multiple register sets, you must not allow the RILs assigned to one shadow register set to overlap the RILs assigned to another register set.

Table 2–7 and Table 2–8 illustrate the validity of register set assignments when preemption within a register set is enabled.

Table 2–7. Example of Illegal RIL Assignment

RIL	Register Set 1	Register Set 2
1	IRQ0	
2	IRQ1	
3		IRQ2
4	IRQ3	
5		IRQ4
6		IRQ5
7		IRQ6

Table 2–8. Example of Legal RIL Assignment

RIL	Register Set 1	Register Set 2
1	IRQ0	
2	IRQ1	
3	IRQ3	
4		IRQ2
5		IRQ4
6		IRQ5
7		IRQ6

Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

Valid Range of hal.log_flags is –1 to 3

The Nios II EDS documentation incorrectly states the valid range for the `hal.log_flags` BSP setting. Valid values of `hal.log_flags` range from –1 through 3.

Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

NicheStack TCP/IP Stack - Nios II Edition Does Not Support TFTP

The Nios II documentation should include the following information:

Versions of the NicheStack TCP/IP Stack other than the Nios II Edition include optional Trivial File Transfer Protocol (TFTP) client and server applications. However, the TFTP client and server are not thread safe. They cannot be used in systems with the MicroC/OS-II RTOS.

TFTP is not available with the NicheStack TCP/IP Stack - Nios II Edition. The TFTP client and server features are disabled.

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Error Message After Renaming Project: "Resource is out of sync with the system"

The *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook* does not explain the correct procedure for renaming a project.

Workaround

The missing information is as follows:

To rename a project in the Nios II SBT for Eclipse, execute the following steps:

1. Right click the BSP project and click **Rename**.
2. Type the new BSP name.
3. Right click the BSP project and click **Refresh**.

☞ If you neglect to refresh the project, you might see the following error message when you attempt to build it:

```
Resource <original_bsp_name> is out of sync with the system
```

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Compiler Flags for Building Custom Newlib

"Common BSP Tasks" in the *Using the Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook* describes how to compile a custom version of the newlib library. However, it does not list the compiler flags that must be selected to generate the correct library.

Workaround

The missing information is as follows:

The Nios II EDS provides a number of precompiled newlib implementations. The provided libraries are precompiled with every viable combination of the GCC compiler flags shown in Table 7.

When you create a BSP with a precompiled newlib, the Nios II SBT selects the newlib matching your BSP's compiler settings. When you create a custom newlib, you must ensure that the compiler flags listed in Table 7 match your BSP's settings.

Table 7. GCC Compiler Options for Newlib

Option Name	Purpose
`-pg`	Link for profiling with gprof
`-EB`	Generate big-endian code
`-mstack-check`	Enable stack checking
`-mno-hw-mul`	Disable use of the `mul` family of instructions
`-mhw-mulx`	Enable use of the `mulx` family of instructions
`-mcustom-fpu-cfg`	Use a floating point custom instruction

☞ The Nios II-specific compiler flag `-mcustom-fpu-cfg` requires the Nios II processor core to be implemented with the floating-point custom instruction as specified in the flag argument. The compiler flag can have one of the following two argument values:

- `-mcustom-fpu-cfg=60-1`—Use the Nios II floating-point custom instruction without divider
- `-mcustom-fpu-cfg=60-2`—Use the Nios II floating-point custom instruction with divider

For details about the Nios II-specific compiler flags `-mstack-check`, `-mno-hw-mul`, and `-mhw-mulx`, refer to "Altera Nios II Options" in *Using the GNU Compiler Collection (GCC)*, installed with the Nios II EDS. For details about the `-pg` compiler flag, refer to "Compiling a Program for Profiling" in *GNU Profiler*, installed with the Nios II EDS. To find the installed documentation, on the Windows Start menu, click **Programs** > **Altera** > **Nios II EDS <*version*>** > **Nios II <*version*> Documentation**, and then click **Literature**.

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Nios II IDE Online Help Expand Buttons Do Not Work

The expand text arrow and the **Show All** option in the online help do not work in Internet Explorer.

Workaround

In Internet Explorer, carry out the following actions:

1. Click the **Refresh/Show Current Topic** icon (yellow arrows icon at the top of the search pane). The table of contents appears with the selected topic highlighted.
2. Click the highlighted topic to refresh the browser frame. Expanded text works.

Alternatively, use Firefox 2.0.0.6.

Solution Status

Not fixed

Hardware Abstraction Layer Errata

This section describes in detail the Nios II Hardware Abstraction Layer issues.

“unused variable” Warning When Building alt_main.c

If you set the `hal.enable_exit` BSP setting to `false`, when building the project you see the following warning in **alt_main.c**:

```
warning: unused variable 'result'
```

Design Impact

This warning is harmless.

Workaround

Set `hal.enable_exit` to `true`.

Alternatively, you can ignore this harmless warning.

Solution Status

Fixed in v10.0 of the Nios II EDS

Build Warnings in Performance Counter Driver Files

If your Nios II system contains a performance counter, you might see the following warning when you build a BSP:

```
warning: return makes pointer from integer without a cast
```

This warning appears in `alt_get_performance_counter_base()` in the software file **altera_avalon_performance_counter.c**.

This warning appears when the your BSP uses the Altera performance counter driver, and the C preprocessor symbol `PERFORMANCE_COUNTER_0_BASE` is defined.

This warning is harmless.

Workaround

In the `alt_get_performance_counter_base()` function in **altera_avalon_performance_counter.c**, add typecast `(void *)` to `PERFORMANCE_COUNTER_0_BASE`.

Solution Status

Fixed in v10.0 of the Nios II EDS

Missing Structure Member Errors in alt_log_printf.c with Small JTAG UART Driver

In a BSP, if you enable the small JTAG UART driver (`altera_avalon_jtag_uart_driver.enable_small_driver`), and enable Altera logging (`hal.log_port`) using the JTAG UART as the log port, when you build the project, the compiler reports missing structure members in **alt_log_printf.c**. The project fails to build.

Workaround

To avoid this issue, execute the following steps:

1. In function `alt_log_jtag_uart_print_control_reg()` in **alt_log_printf.c**, insert the italicized code as shown in Example 9.

Example 9. ALT_LOG_PRINTF() Correction

```
ALT_LOG_PRINTF(
    "%s SW CirBuf = %d, HW FIFO wspace=%d AC=%d WI=%d RI=%d WE=%d RE=%d\r\n",
    header,

/* Beginning of inserted code */
#ifndef ALTERA_AVALON_JTAG_UART_SMALL
    (dev->tx_out-dev->tx_in),
#else
    0,
#endif
/* End of inserted code */

    space,ac,wi,ri,we,re);
```

2. Rebuild the project.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Hardware Example Design Errata

This section describes in detail the Nios II EDS hardware example design issues.

Quartus II Compilation Warnings for Nios II Stratix II 2S60 ROHS Example

You might see the following warnings if you try to compile the Nios II Stratix® II 2S60 ROHS example design, installed at <*Nios II EDS install path*>**/examples/vhdl/niosII_stratixII_2s60/standard** or downloaded from the Altera Wiki (**www.alterawiki.com**):

```
Warning (10541): VHDL Signal Declaration warning at\
    NiosII_stratixII_2s60_standard.vhd(59): used implicit default value for signal\
    "cpu_data_master_read_data_valid_NiosII_stratixII_2s60_standard_clock_0_in"\
    because signal was never assigned a value or an explicit default value. Use of\
    implicit default value may introduce unintended design optimizations.
Warning (10542): VHDL Variable Declaration warning at\
    altera_europa_support_lib.vhd(340): used initial value expression for variable\
    "arg_copy" because variable was never assigned a value
Warning (10542): VHDL Variable Declaration warning at\
    altera_europa_support_lib.vhd(344): used initial value expression for variable\
    "arg_length" because variable was never assigned a value
```

You can safely ignore these warnings.

☞ The Nios II Stratix II 2S60 ROHS example is deprecated.

Workaround

None.

Solution Status

Not fixed.

Software Example Errata

This section describes in detail the Nios II EDS software example issues.

Hardware Tutorial Software Example Hangs on Some Boards

The **count binary** application used in the tutorial hangs when run on some Altera development boards. The software runs through the binary count once and stops responding.

This issue affects boards that do not have a display to handle the LCD output. The affected boards include the following:

- Embedded Systems Development Kit, Cyclone III Edition (EP3C120)
- Stratix IV GX FPGA Development Kit (EP4SGX230)

Workaround

Modify line 18 of the **count_binary.h** file, as follows:

```
# define LCD_PRINTF(lcd, args...) /* Do Nothing */
```

Rebuild and run the software again.

Solution Status

Fixed in v10.0 of the Nios II EDS

Networking Examples

If you are running a networking software example, you might be asked for a nine-digit number. You are directed to find this number on a sticker on your Nios development board, identified by the prefix **ASJ**. Not all Nios development boards have this sticker.

Workaround

If your Nios development board does not have a sticker with the letters **ASJ** followed by a nine-digit number, enter a unique nine-digit number when prompted. To avoid network address conflicts, ensure that this number is unique to each Nios board connected to your network.

Solution Status

Not fixed

Flash Programmer Errata

This section describes in detail the Nios II EDS issues related to the flash programmer.

Unable to Configure FPGA from Flash with Parallel Flash Loader

If you program an FPGA image to flash memory on a board that utilizes the Parallel Flash Loader, the FPGA subsequently fails to load from flash memory. This issue arises because the **sof2flash** utility generates a flash file with an incompatible Programmer Object File (**.pof**) bitstream format.

Affected Configurations

This issue impacts the following target hardware:

- The Stratix IV GX FPGA Development Kit
- The Arria® II GX FPGA Development Kit
- Any hardware using the Parallel Flash Loader

Workaround

1. Visit the MySupport website and request Nios II EDS patch 0.73.

   Alternatively, upgrade to the Nios II EDS v. 9.1 SP1 or later.

2. Invoke **sof2flash** with the following options:

   - `--pfl`
   - `--optionsbits=<`*`option bit address`*`>`

   With these options, **sof2flash** generates a flash file with a compatible **.pof** file bitstream format.

Solution Status

Fixed in v. 9.1 SP1 of the Nios II EDS

elf2flash File Size Limit

The **elf2flash** utility supports **.elf** files up to approximately 24 MBytes in size. The **elf2flash** utility might fail on files larger than 24 MBytes, with the error message `java.lang.OutOfMemoryError`.

Workaround

Lower the number of symbols in your **.elf** file by turning off debug symbols.

Alternatively, specify less initialized data in the application.

Solution Status

Not fixed

Hardware Simulation Errata

This section describes in detail the Nios II EDS issues related to hardware simulation.

Vectored Interrupt Controller Does Not Support VHDL Simulation Models

SOPC Builder reports errors when you try to generate VHDL simulation files for the Vectored Interrupt Controller.

Workaround

There is no workaround.

Solution Status

Fixed in 9.1 of the Nios II EDS

Error "UNC paths are not supported" Launching ModelSim

If you launch ModelSim® from a working directory that is mapped via a universal naming convention (UNC) path (a path that starts with `//` instead of drive letter), you receive the following error message in SOPC Builder: **UNC paths are not supported. Defaulting to Windows directory**. This error occurs because ModelSim is calling a command shell, which does not support UNC paths.

Workaround

Map the UNC path to a drive letter and use the drive letter to reference the working directory in the launching shell.

Solution Status

Not fixed

Uninitialized .bss Variables in Simulation

If your program reads the value of an uninitialized `.bss` variable during HDL simulation, and the BSP (system library) is compiled with the **ModelSim only, no hardware support** property enabled in Nios II IDE, a warning appears about unfiltered data being `'x'`. This warning appears because when this property is enabled, the code that clears the `.bss` memory region is omitted to speed up HDL simulation so this memory region is uninitialized. The `.bss` region contains global and static local variables that are not initialized by the application so they default to a value of zero. When the Nios II processor reads uninitialized variables, it displays a warning and converts any of the bits of the uninitialized data to zero which correctly mimics the effect of the missing `.bss` clearing code. The Hardware Abstraction Layer (HAL) code that executes before and after `main()` might use `.bss` variables, so these warnings might appear even if your application does not use the `.bss` section.

Solution Status

Not fixed

Host Platform Errata

This section describes in detail the Nios II EDS issues related to the host development platform.

Nios II EDS Requires Administrative Permissions on Windows Vista and Windows 7

On the Windows Vista and Windows 7 operating systems, the Nios II EDS requires your user account to have administrative permissions.

Attempts to write to a directory other than the user home directory might fail, especially writing to an installation directory. For example, you might be unable to create a software project under an example design directory. Error messages might vary.

Workaround

Obtain an administrator account on your workstation.

Alternatively, start Nios II EDS applications in administrator mode. For example, when launching the Nios II SBT for Eclipse, right-click **Nios II <*version*> Software Build Tools for Eclipse** and click **Run As Administrator**.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Cannot Launch Nios II SBT for Eclipse on Red Hat Linux

You might be unable to launch Nios II SBT for Eclipse on the Red Hat Linux operating system. This is an issue with the version of XULRunner in Eclipse 3.4. The issue is resolved in XULRunner 1.9.1, which is available with Eclipse 3.5.

Workaround

Upgrade XULRunner to a newer version. To determine what version of XULRunner you have, type the following command at the command prompt:

```
xulrunner -v↵
```

To upgrade XULRunner using the **yum** software package manager, type the following command:

```
sudo yum update xulrunner↵
```

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Windows/Cygwin: Nios II Processor Generation Failure

You might see the following error message when you generate your Nios II system in SOPC Builder:

```
Error: Generator program for module 'cpu_fpoint' did NOT run
successfully
```

This error might occur on a Windows-based system when all of the following conditions are true:

- You have installed a version of Cygwin other than the one distributed with the Quartus II software.
- You launch the Quartus II software and SOPC Builder from a Nios II Command Shell.
- You enable the floating point custom instruction (FPCI) in the Nios II processor core.

Workaround

Launch the Quartus II software and SOPC Builder from the Windows Start menu, and regenerate your system.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Windows Vista: Limited Support in Nios II EDS

The Quartus II software introduces Windows Vista (32-bit and 64-bit) support in version 7.2. However, the Nios II Embedded Design Suite supports only the following platforms:

- Windows 2000
- Windows XP
- Windows XP (64)
- Windows Vista Business Edition (32-bit)
- SUSE 9 (32-bit)
- SUSE 9 (64-bit)
- Red Hat Linux v3.0 (32-bit)
- Red Hat Linux v3.0 (64-bit)
- Red Hat Linux v4.0 (32-bit)
- Red Hat Linux v4.0 (64-bit)

Workaround

There is no workaround.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Nios II IDE Errata: Building Projects

This section describes in detail the issues related to building projects in the Nios II IDE.

Nios II IDE Command-Line Tools Select Wrong Workspace on Linux

On Linux systems, the Nios II IDE command-line tools use the default Eclipse project workspace, regardless what workspace you are using in the IDE. The default workspace is at `$SOPC_KIT_NIOS2/eclipse/nios2-ide-workspace-<`*`version`*`>`.

This issue affects the following tools:

- `nios2-create-system-library`
- `nios2-create-application-project`
- `nios2-build-project`
- `nios2-import-project`
- `nios2-delete-project`

Workaround

Explicitly supply the Nios II IDE command-line tools with a workspace location, by means of the `-data` command-line argument. The syntax of the `-data` argument is as follows:

```
-data <path to workspace>
```

The path to the workspace must be absolute, and must not contain whitespace. Aside from these restrictions, any valid file system path can be used for the workspace. Example 10 shows how to specify a workspace path.

Example 10. Providing a Non-Default Workspace Location in Linux

```
nios2-create-project -data $HOME/myworkspace <other arguments>↵
```

Solution Status

Not fixed

Nios II IDE Command-Line Tools Hang on Windows

On Windows systems, the Nios II IDE command-line tools are sometimes unable to locate the project workspace. When this happens, the tools might hang.

This issue potentially affects the following tools:

- `nios2-create-system-library`
- `nios2-create-application-project`
- `nios2-build-project`
- `nios2-import-project`
- `nios2-delete-project`

Workaround

Explicitly supply the Nios II IDE command-line tools with a workspace location, by means of the `-data` command-line argument. The syntax of the `-data` argument is as follows:

```
-data <path to workspace>
```

The path to the workspace must be absolute, and must not contain whitespace. Aside from these restrictions, any valid file system path can be used for the workspace. The workspace shown in Example 11 is the default workspace used by the Nios II IDE in version 9.0.

Example 11. Providing a Workspace Location

```
nios2-create-project -data c:/altera/90/nios2eds/\
    bin/eclipse/nios2-ide-workspace-9.0 <other arguments>
```

Solution Status

Not fixed

Nios II IDE Cannot Find stdio.h in Outline View

If you create a new project using the Nios II IDE and try to open the **stdio.h** file from the Outline view before building the project, the IDE displays the error message **No include files were found that matched that name**.

Workaround

Build the project before attempting to open **stdio.h**.

Solution Status

Not fixed

Nios II IDE Hangs With UNC Project Path

In the **New Project** dialog box, if you turn on **Specify Location** and specify the path in UNC form, the IDE might hang.

Workaround

Map the UNC path to a remote drive which looks like a Windows drive to the Nios II IDE.

Solution Status

Not fixed

Build Failure with Nios II Advanced Exceptions, MMU, and MPU

Projects created in the Nios II IDE for an SOPC Builder system containing a Nios II processor configured with advanced exceptions, the memory management unit (MMU), or the memory protection unit (MPU) fail to build, and generate the following error messages:

```
ERROR - Classic build flow for Nios II system library does not support
the Nios II \
    advanced exceptions.
Use the non-classic Nios II Board Support Package instead.
<timestamp> - (SEVERE) generate: java.lang.IllegalStateException: \
    java.lang.IllegalStateException: \
    com.altera.ingenuous.GTFElement.GTFElementErrorException: \
    <error> element in GTF script
make[1]: *** [system_description/../obj/system.h-t] Error 1
make: *** [system_project] Error 2
```

Workaround

Create the software project with the Nios II SBT command-line development flow. Alternatively, use the Nios II SBT for Eclipse.

Solution Status

Not fixed

Build Command Not Functional for BSPs Created With the Nios II SBT

The build option in the Nios II IDE menu does not rebuild BSPs imported to the IDE.

Workaround

The Nios II SBT for Eclipse can both build and debug projects created on the command line. The Nios II SBT for Eclipse is the preferred tool for debugging Nios II SBT projects.

For information about the Nios II SBT for Eclipse, refer to the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*.

In the Nios II IDE, you can build the BSP by building the associated application project.

Solution Status

Not fixed

Linker Errors with Dual-Port Memories

If your instruction master and data master ports are connected to the same dual-port memory and the ports have different addresses, your code fails to run or you experience a linker error. The Nios II IDE does not warn you of the addressing violation.

Workaround

Assign the same address to both ports of the dual-port memory.

Solution Status

Not fixed

User-Managed BSP Settings Not Supported in Nios II IDE

For projects created with the Nios II SBT and imported to the Nios II IDE, the IDE configuration settings have no effect.

For example, objdump, compiler, and linker settings made in the IDE are ignored. This behavior occurs because Nios II SBT projects are not IDE-managed projects. In addition, the make-related preferences do not pertain to imported Nios II SBT projects. The IDE ignores these options during the build process.

Workaround

Make these settings in the project's makefile.

Solution Status

Not fixed

URL Project Location Causes Project Creation Error

When you try to create a new project in an existing workspace, you might see an error dialog box saying:

```
Project cannot be created. Reason: Internal Error
```

This error might occur if the path to any project in the workspace is a URL location, for example **file:/F:/Design**. To view the path, right-click on the project and select **Properties**.

Workaround

Import your existing application and system library projects to a new workspace.

Solution Status

Not fixed

Compilation Error with Separate Exception Stack Option

Choosing the **Use a separate exception stack option** might cause the following compilation error when building a project:

```
VARIABLE %STACK_POINTER%"
```

This error occurs if the exception stack is larger than the memory available for it.

Workaround

On the system library properties page for the project, turn off the separate exception stack or reduce the **Maximum exception stack size** setting.

Solution Status

Not fixed

Incorrect Stack and Heap Free Space Report

The makefile reports an incorrect number of bytes free for the stack and heap, if the heap and stack are in different memory regions.

Workaround

No workaround available.

Solution Status

Not fixed

Nios II IDE Reports Problems Without Displaying Error in Console

When building a project, the Nios II IDE reports problems, but the build output in the console does not contain any errors.

The Nios II IDE incorrectly reports some linker warnings as errors, with a dialog box saying **Errors exist in a required project**. The Dhrystone software example exhibits this behavior, and recompiling the project again makes the issue go away.

Workaround

If the Console output does not contain errors, then the project built correctly. On subsequent builds, the linker step is skipped and the errors do not appear.

Solution Status

Not fixed

Nios II IDE Errata: Debugging Projects

This section describes in detail the issues related to debugging projects in the Nios II IDE.

Method for Accessing MMU and MPU Registers in the Debugger

You might observe an error if you try to read or write an MMU or MPU register via the Nios II Debugger.

Workaround

To read an MPU region, execute the following steps:

1. Set region `INDEX` in the `MPUBASE` register.
2. Exit and reenter debug mode, that is, single step.
3. Set the `RD` bit in the `MPUACC` register.
4. Exit and reenter debug mode, that is, single step.
5. Read back `MPUBASE` for pertinent information.
6. Read back `MPUACC` for pertinent information.

For a system with an MMU or MPU, this workaround allows you to read and write the current values of the registers. However, you cannot use it to control MPU regions or MMU TLB entries.

The debug core copies the Nios II processor's register values to its internal memory when the processor enters debug mode. The debug core writes register values back to the processor only when the processor leaves debug mode. Therefore, if you attempt to set an MPU region with several consecutive values, only the last one, when leaving debug mode, is committed to the processor.

Solution Status

Not fixed

IDE Cannot Display Imported Profiling Data

If you create a software project in the Nios II SBT command line flow, generate profiling data to **gmon.out** in the Nios II command shell, and then import the project to the IDE, you cannot use the Profiling perspective to view the profiling data. When you attempt to view **gmon.out**, the IDE displays the following error message:

```
nios2-elf-gprof: ../: not in a.out format.
```

Workaround

Run `nios2-elf-gprof` from the Nios II command shell to generate a profiler report.

Solution Status

Not fixed

Error Message when Downloading .elf File

If the **Build Automatically** option is turned on in the IDE, you might get the following error when downloading a **.elf**:

```
"2 [main] sh 5736 fork: child 3892 - died waiting; \
    for longjmp before initialization, errno 11"
```

The **Build Automatically** option is off by default, as recommended by the Eclipse help system.

Workaround

Turn off the **Build Automatically** option and download the **.elf** file again.

Solution Status

Not fixed

Variable Casting Unsupported in ISS

Casting variables in the Variables view in the Debug perspective when using the Nios II ISS might cause an exception dialog box to open.

This exception might occur if you select **Debug as ISS** in the Nios II IDE and try to cast variables via the Variables view.

Workaround

There is no workaround.

Solution Status

Not fixed

Trace Debug Does Not Support Instruction-Related Exceptions

The instruction-related exception handler is not supported by the Nios II trace tools.

Workaround

There is no workaround.

Solution Status

Not fixed

Trace Debug Does Not Support the JMPI Instruction

The JMPI instruction is not supported by the Nios II trace tools.

Workaround

There is no workaround.

Solution Status

Not fixed

Cannot Locate Source Code in Driver Files Shared by Multiple Projects

If you hit a breakpoint in a driver file, and that driver file is shared with another project that is closed, the Nios II IDE might indicate that it cannot locate the source code.

Workaround

Open the closed system library project and resume debugging.

Solution Status

Not fixed

Console Window Is Not Updated After ISS Error

After performing a **Run as ISS**, if you receive an ISS error in the console window, the console is not updated subsequently.

Workaround

Close the console window after receiving an ISS error. A new console window opens when a new message is available.

Solution Status

Not fixed

ISS Fails on Designs Containing Triple Speed Ethernet MAC or SG-DMA Components

You receive an Internal Error when attempting to perform an ISS simulation of designs containing the Altera Triple Speed Ethernet media access control (MAC) or SG-DMA components because the Nios II ISS does not support these components.

Workaround

Remove the Triple Speed Ethernet MAC and SG-DMA components from your system and perform ISS simulation on the simplified system. You can also simulate the design in ModelSim or test it on hardware.

☞ Before removing the Triple Speed Ethernet MAC and SG-DMA components, make a copy of the unmodified system to ensure that you can return to the original configuration.

Solution Status

Not fixed

Memory Window Sets Control Register Values Incorrectly

The memory window might incorrectly set values in memory-mapped control registers. For example, writing 0x1234 to a byte addressed register results in the value 0x3434 being stored in the register. The memory window shows this incorrect value.

Workaround

Use the GNU debugger (GDB) console window in the IDE, instead of the memory window, to write to the registers. For example, type the following command:

```
set {int} <register address>=0x1234↵
```

☞ You must refresh the memory window for it to correctly display the target value.

Solution Status

Not fixed

Programs That Interact With a Terminal Console on Windows Do Not Work

Programs with this behavior work in v. 6.0 and earlier, but do not work in Nios II IDE v. 6.1 and later.

The Eclipse platform in v. 6.1 and later of the IDE (on Windows only) sends the string `\r\n` instead of just `\n` when running or debugging using the Terminal. This behavior can break existing software designs, and it is inconsistent with `nios2-terminal`, which still just sends `\n`.

Workaround

Change the software running on the Nios II processor to parse for `\r\n` as well as `\n`.

Solution Status

Not fixed

“Run as ModelSim” in the Nios II IDE Fails

The **Run as ModelSim** command might fail on launch configurations created in v. 7.0 or earlier of the IDE. This problem does not occur for new launch configurations.

Workaround

Select a location for the ModelSim tool from the launch configuration dialog box. You can use the **Browse** button next to the **ModelSim path** group, or type in a path to the ModelSim directory (for example **c:/altera/71/modelsim_ae/win32aloem**).

Solution Status

Not fixed

The Restart Command on the Run Menu Does Not Work

The **Restart** command on the **Run** menu does not work.

Workaround

Stop the program, then debug it again. If the debugger is hung in an endless loop, use the following **bash** alias to break the target, then stop it:

```
alias break="kill -2 \'ps -a | grep nios2-elf-gdb | cut -f6 -d' '\'"
```

Solution Status

Not fixed

Watchpoints Do Not Work on Certain Variables

Watchpoints do not work on a variable whose size is not 32 bits.

Workaround

Change the types of global and static local variables to `int`, `long`, or `unsigned long` before setting watchpoints on them.

Solution Status

Not fixed

Nios II IDE Errata: Navigating Projects

This section describes in detail the issues related to navigating projects in the Nios II IDE.

Nios II IDE Freezes While Displaying the Splash Screen

After clicking **Switch Workspace** on the File menu on a Windows machine, a Nios II IDE splash screen appears. Unfortunately, this splash screen obscures the dialog box that asks you to specify the new workspace. As a result, the IDE appears to freeze.

Workaround

Press Alt-Tab to switch applications. Two relevant application icons appear: an Eclipse icon associated with the splash screen and a Nios II IDE icon associated with the **Workspace** dialog box. Select the Nios II icon to bring the dialog box to the foreground.

Solution Status

Not fixed

Internal Error When Double-Clicking on a Large Objdump File

On Windows when opening a large **objdump** file in the Nios II IDE, you might get the following error message:

```
Unable to create this part due to an internal error.
Reason for the failure: Editor could not be initialized.
```

Workaround

Adjust the Windows launch arguments for the Nios II IDE editor. Perform the following steps:

1. On the Windows Start menu, browse to the **Nios II EDS** program icon, right-click it, then click **Properties**. The **Windows Properties** dialog box appears.
2. In the **Target** field, append `"vmargs -Xmx1024m"` to the end of the path to the Nios II IDE executable. For example:

   ```
   C:\altera\72\nios2eds\bin\eclipse\nios2-ide.exe -vmargs -Xmx1024m
   ```

Solution Status

Not fixed

C/C++ Scanner Does Not Support Certain C/C++ Constructs

The C/C++ scanner performs C/C++ Search, navigation, open declaration, and parts of content assist. Due to limitations of the C/C++ scanner, these features do not work with C code constructs not supported by the C++ language. An example is functions that take a function pointer as an argument.

Workaround

If the C/C++ Search fails, use the File Search facility.

Solution Status

Not fixed

Nios II SBT Errata

This section describes in detail the issues in the Nios II SBT.

Nios II SBT Creates Spurious BSP for System with MMU

The Nios II SBT is not intended to generate BSPs for the Nios II MMU. The SBT incorrectly lets you create a BSP for a Qsys-generated Nios II system with an MMU. This BSP includes a **system.h** file with incorrect MMU information.

You cannot use the Nios II SBT to create a BSP for the Nios II MMU.

Affected Configurations

Qsys designs with the Nios II MMU

Workaround

None.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Error Running Nios II Project: 'Downloading ELF Process failed'

If the Nios II processor's cpu.data_master port is not connected to all program memories (memories to which the **.elf** file is downloaded) the software project fails to run on Nios II hardware.

Failure to connect cpu.data_master to all program memories is a design error that the Nios II SBT does not detect.

Workaround

Connect cpu.data_master to all program memories.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

BSP Editor Does Not Show Command-Line Help

The following commands in the Nios II Command Shell should display command-line help:

- `nios2-bsp-editor --help`
- `nios2-bsp-editor --extended-help`

Instead, these commands launch the BSP Editor.

Workaround

For help with the BSP Editor, refer to "Using the BSP Editor" in the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

BSP Not Updated for Memory Size Changes in SOPC Builder

If you change the size of a memory in SOPC Builder, the memory region size in any previously created BSP is no longer correct. Regenerating the BSP does not update the BSP's memory region size.

Workaround

In the BSP Editor, on the **Linker Script** tab, you can correct the memory region size either of the following ways:

- Edit the memory region size manually.
- Click **Restore Defaults** to rerun the default Tcl script. The default Tcl script reads the updated memory region size from the SOPC information file (**.sopcinfo**) and updates the BSP.

For detailed information about keeping your BSP consistent with changes in the underlying SOPC Builder system, refer to "Revising Your BSP" in the *Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook*.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Build Errors on Interrupt API Calls

You might observe build errors if your application project installs an ISR that calls the `alt_irq_register()` function.

The Nios II EDS v9.1 includes support for an optional External Interrupt Controller (EIC) interface on the Nios II processor. To support the EIC interface, the HAL includes an enhanced interrupt API.

When you create a BSP, the SBT determines which interrupt API(s) the device drivers in your system support. If all registered drivers in the BSP identify themselves as supporting the enhanced API, the SBT implements the enhanced API. All Altera device drivers in v9.1 and later support the enhanced API.

When the enhanced API is implemented, the legacy interrupt API, including `alt_irq_register()`, is not available. Therefore, if the application code contains a call to a legacy API function, a linker error occurs.

This issue might affect application projects that call `alt_irq_register()` or other legacy interrupt API functions directly, rather than depending on drivers in the BSP for interrupt support.

If your Nios II application project's source code registers an ISR using `alt_irq_register()`, or includes an ISR, your application might fail to compile in the Nios II EDS v. 9.1, due to undefined references to `alt_irq_register()`, or to a change to the ISR function prototype.

Workaround

Modify your application code to use the new enhanced interrupt API. To use the enhanced API, you must modify several function calls and all ISR function prototypes.

For information about supporting the enhanced interrupt API, refer to the *Exception Handling* chapter of the *Nios II Software Developer's Handbook*, or to *AN 595: Vectored Interrupt Controller Usage and Applications*.

☞ You must upgrade to the enhanced API if you wish to use an EIC in your hardware design to accelerate interrupt response time. The enhanced API also works with the Nios II processor's internal interrupt controller.

Solution Status

Not fixed

SBT Fails if Nios II EDS is Installed in a Path Containing Spaces

The Nios II Command Line SBT fails if you install the Nios II EDS in a path containing spaces.

Workaround

Reinstall the Nios II EDS to a path that does not contain spaces.

Solution Status

Not fixed

User-Managed BSP Settings Not Supported in Nios II IDE

This issue is described on page 48.

Peripheral Errata

This section describes in detail the Nios II EDS issues related to peripheral components.

Vectored Interrupt Controller Does Not Support VHDL Simulation Models

This issue is described on page 42.

Unaligned Transfers of Small Payloads Fail on SG-DMA

The Scatter Gather DMA SOPC Builder peripheral does not correctly handle unaligned transfers with small payloads. A payload length smaller than the data width causes erroneous data transfers.

Workaround

Avoid using DMA devices to transfer small payloads.

If absolutely necessary, for a 32-bit SG-DMA, a minimum length of 4 bytes guarantees that data is transferred correctly.

Solution Status

Not fixed

DMA Controller Always Busy in Burst Mode

The DMA controller component (`altera_avalon_dma`), when enabled for burst transactions, does not perform transfers at widths other than its full data width. The DMA controller is always busy.

Workaround

When bursting is enabled, the DMA controller must be programmed to perform transactions at its full data width.

Solution Status

Not fixed

Non-System-Wide Reset Can Cause Improper Initialization of Mailbox Core

The `altera_avalon_mailbox` peripheral might not be initialized properly when a soft (non-system-wide) reset occurs. In this condition, mailbox contents (read and write pointers) are not reinitialized and might show potentially stale data.

Workaround

Ensure that a system-wide reset event occurs by asserting the `reset_n` input to the SOPC Builder system containing the mailbox. This resets all peripherals and Nios II processors in the system.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Target Software Errata

This section describes in detail the Nios II EDS issues related to target software packages.

NicheStack tcp_wake() Function Does Not Awaken All select() Sleepers

This issue impacts software based on the NicheStack TCP/IP Stack - Nios II Edition, with multiple sleeping tasks that depend on the same `select()` event to wake up. The `tcp_sleep()` and `tcp_wake()` functions do not work properly for tasks that call `select()`.

Tasks can normally use `tcp_sleep()` and `tcp_wake()` to suspend and then awaken themselves upon specific events. The `select()` function passes an event pointer to `tcp_sleep()` that is identical for every task that might call `select()`. All other events that call `tcp_sleep()` pass it a unique event pointer that specifically identifies them as the task to be awakened at the next event.

The `tcp_wake()` function apparently assumes that the list of sleepers contains unique event pointers by which they can be identified. Therefore, when a `select()` event occurs, instead of awakening all tasks waiting for it, `tcp_wake()` awakens only the first task in the list. Therefore, when more than one task calls `select()` to await some outcome on a socket, only the highest-priority task in the sleepers list is awakened at `select()` events. Other sleeping tasks that depend on the same `select()` event do not wake up.

Workaround

There is no workaround.

Solution Status

Fixed in v10.0 of the Nios II EDS

malloc(), realloc() Failures With MicroC/OS-II

When you use the MicroC/OS-II RTOS, calls to `malloc()` and `realloc()` might fail if successive calls to `malloc()` or `realloc()` within a MicroC/OS-II task occur after changing the task priority of the task in which a memory block is originally allocated.

Workaround

Use one of the following workarounds:

- Allocate and/or reallocate memory blocks outside of MicroC/OS-II tasks, before task switching starts. Preallocated memory blocks make it possible to change thread priorities at runtime.
- Allocate fixed areas of memory using arrays (rather than using `malloc()`) before task switching starts. Fixed memory arrays make it possible to change thread priorities at runtime.
- Allocate memory using `malloc()` or `realloc()` from a MicroC/OS-II task. You can change task priorities at runtime, but only for tasks that have not used `malloc()` or `realloc()`.

Solution Status

Fixed in v10.0 of the Nios II EDS

EIC-Based Nios II System Crashes With NicheStack

When you attempt to run software based on the NicheStack TCP/IP Stack - Nios II Edition on a hardware design incorporating an external interrupt controller such as the VIC, the Nios II system crashes during initialization, with unpredictable results.

Workaround

There is no workaround.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

stdio Does Not Work with MicroC/OS-II and Small C Library

stdin, stdout, and stderr do not work in MicroC/OS-II applications built with the Small C library option.

Workaround

Disable the small C library option.

Solution Status

Not fixed

cout From MicroC/OS-II Task Does Not Send Data to stdout

If neither printf() or cout is used from main() before tasks are started, cout does not work from a task.

Workaround

Add the following C++ code to the beginning of main():

```
std::ios_base::sync_with_stdio(false);
```

Solution Status

Not fixed

Toolchain and Utilities Errata

This section describes in detail the Nios II EDS issues related to the Nios II and GNU compiler toolchains, including tools such as **gcc**, **gdb**, and **sof2flash**.

Warning Message: “'pragma_reverse_bitfields' attribute directive ignored”

The Nios II GCC 4 tool chain does not support the C compiler pragma and flag for reverse bit fields. If you try to use the compiler pragma reverse_bitfields, you see the following warning:

```
'pragma_reverse_bitfields' attribute directive ignored
```

Similarly, the C compiler flag -mreverse-bitfields is unsupported.

Workaround

Use the GCC 3 tool chain.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

nios2-download Cannot Find JTAG Instance IDs

Because Qsys uses different instance naming conventions compared to SOPC Builder, if you use the --jdi and --cpu_name flags with the **nios2-download** command, **nios2-download** cannot properly parse the Qsys instance names to find the correct JTAG debug module instance ID in the **.jdi** file.

Design Impact

Qsys designs with multiple JTAG debug modules

Workaround

Manually find the JTAG debug module ID in the **.jdi** file. Pass this value to **nios2-download with** the `--instance` flag.

The JTAG debug module ID might change after each Quartus II compilation.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

nios2-console Does Not Work in the GCC 4 Command Shell

If you try to run the **nios2-console** command in the GCC 4 Command Shell, you see the following error message:

```
Can't locate strict.pm in @INC ...
```

Workaround

Use the GCC 3 Command Shell.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

C++ Exceptions Unsupported in a Multi-Threaded Environment

C++ exceptions are only supported in a single-threaded environment. They cannot be used in a multi-threaded environment such as MicroC/OS-II.

Also see "C++ Exceptions Unsupported by the Nios II GCC 4 Toolchain".

Workaround

In a multi-threaded environment, you need to guard C++ exceptions with semaphores.

Solution Status

Not fixed

C++ Exceptions Unsupported by the Nios II GCC 4 Toolchain

With the GCC 4 toolchain, C++ exceptions are not supported.

Also see "C++ Exceptions Unsupported in a Multi-Threaded Environment".

Workaround

If your single-threaded project requires C++ exceptions, use the GCC 3 toolchain.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

memcpy() Optimization Misalignment

When optimization is turned on (`-O1` or higher), if you use `memcpy()` and the source pointer is aligned to a 32-bit boundary, the compiler implements `memcpy()` with word-oriented instructions as part of the optimization process. This optimization technique causes unexpected results in your software if `memcpy()` is used on a misaligned address.

Workaround

Take steps to ensure that an optimized implementation of `memcpy()` is called only with aligned data pointers.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Unable to Configure FPGA from Flash with Parallel Flash Loader

This issue is described on page 41.

GNU Assembler Does Not Accept the --defsym Flag

According the GNU documentation, you can set an assembler definition by using the `--defsym` flag, but it does not work in the following form: `--defsym MY_VAR=1`.

Workaround

There is no workaround.

Solution Status

Not fixed

How to Contact Altera

For the most up-to-date information about Altera products, refer to Table 8.

Table 8. Contact Information

Contact *(1)*	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Altera literature services	Email	literature@altera.com
Non-technical support (General)	Email	nacomp@altera.com
(Software Licensing)	Email	authorization@altera.com

Note:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

Table 9 shows the typographic conventions that this document uses.

Table 9. Typographic Conventions

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Indicates command names, dialog box titles, dialog box options, and other graphical user interface (GUI) labels. For example, **Save As** dialog box. For GUI elements, capitalization matches the GUI.
bold type	Indicates directory names, project names, disk drive names, file names, file name extensions, dialog box options, software utility names, and other GUI labels. For example, **\qdesigns** directory, **d:** drive, and **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Indicates document titles. For example, *AN 519: Stratix IV Design Guidelines.*
Italic type	Indicates variables. For example, *n* + 1. Variable names are enclosed in angle brackets (< >). For example, <*file name*> and <*project name*>**.pof** file.
Initial Capital Letters	Indicates keyboard keys and menu names. For example, Delete key and the Options menu.
“Subheading Title”	Quotation marks indicate references to sections within a document and titles of Quartus II Help topics. For example, “Typographic Conventions.”
`Courier type`	Indicates signal, port, register, bit, block, and primitive names. For example, `data1`, `tdi`, and `input`. Active-low signals are denoted by suffix `n`. For example, `resetn`. Indicates command line commands and anything that must be typed exactly as it appears. For example, `c:\qdesigns\tutorial\chiptrip.gdf`. Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword `SUBDESIGN`), and logic function names (for example, `TRI`).
1., 2., 3., and a., b., c., and so on.	Numbered steps indicate a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ■	Bullets indicate a list of items when the sequence of the items is not important.
☞	The hand points to information that requires special attention.
⚠ CAUTION	A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.
⚠ WARNING	A warning calls attention to a condition or possible situation that can cause you injury.
↵	The angled arrow instructs you to press Enter.
👣	The feet direct you to more information about a particular topic.

101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support
www.altera.com/support

Nios II Embedded Design Suite Release Notes and Errata

© September 2010 RN-EDS-7.1

About These Release Notes

These release notes cover versions 9.0 through 10.0 SP1 of the Altera® Nios® II Embedded Design Suite (EDS). These release notes describe the revision history and errata for the Nios II EDS.

Errata are functional defects or errors, which might cause the product to deviate from published specifications. Documentation issues include errors, unclear descriptions, or omissions from current published specifications or product documents.

For the most up-to-date errata for this release, refer to the latest version of the *Nios II Embedded Design Suite Release Notes and Errata* on the Altera website. You can refer to release notes and errata for the Nios II processor core in the *Nios II Processor* chapter of the *MegaCore IP Library Release Notes and Errata*.

Revision History

Table 1 shows the revision history for the Nios II EDS, including the Nios II C2H Compiler.

Table 1. Nios II Embedded Design Suite Revision History (Part 1 of 2)

Version	Date	Description
10.0	July 2010	■ Software Build Tools (SBT) for Eclipse ■ Updated Eclipse to version 3.5 and Eclipse C/C++ development toolkit (CDT) to version 6.0 ■ Improved Source File management ■ Supports Multi-CPU launch ■ Updated to Nios II GCC 4.1.2 tool chain ■ Supports both SBT for Eclipse and Nios II Command Shell ■ GCC 4 improves compilation time compared to GCC 3 on Windows ■ Host platform ■ Supports Windows 7 ■ Supports SUSE Linux Enterprise 11 ■ Support for SUSE Linux Enterprise 9 discontinued ■ New Nios II software design examples for Reduced Gigabit Media Independent Interface (RGMII) ■ Web Server (RGMII) ■ Simple Socket Server (RGMII) ■ Legacy hardware design examples (Stratix® II and Cyclone® II) are removed ■ Installer changes ■ Nios II EDS must be installed next to Quartus® II
9.1 SP2	March 2010	Maintenance release

Table 1. Nios II Embedded Design Suite Revision History (Part 2 of 2)

Version	Date	Description
9.1 SP1	January 2010	Maintenance release
9.1	November 2009	■ The Nios II SBT for Eclipse, an integrated development environment based on Eclipse. Recommended for all new Nios II projects. ■ The Nios II Integrated Development Environment (IDE) is available in the Nios II EDS, but not recommended for new projects. ■ The Nios II Studio is discontinued. ■ The vectored interrupt controller (VIC) core offers high-performance, low-latency interrupt handling. ■ Nios II Design Examples with memory management unit (MMU): ■ Nios II 3C120 Hardware Reference Design with MMU, Cyclone III Edition ■ Nios II 4S230 Hardware Reference Design with MMU, Stratix IV GX Edition Commercial and open-source support for embedded Linux is available for these systems. ■ The Embedded Systems Lab for DE1 for Quartus and Nios II EDS v9.1 is available for download from the Nios Community Wiki (www.nioswiki.com). ■ Tcl scripting for software packages and drivers enhanced to support callback functions.
9.0	March 2009	Added the `nios2-convert-ide2sbt` tool, which converts projects from the Nios II IDE development flow to the Nios II SBT development flow
8.1 SP1	January 2009	Nios Community Wiki URL changing (www.nioswiki.com)
8.1	November 2008	Maintenance release

For more information on new features and changes, refer to the *Nios II Processor Reference Handbook*, the *Nios II Software Developer's Handbook*, and the *Nios II C2H Compiler User Guide*. For information about changes to the Nios II processor core, refer to the *Nios II Processor* chapter of the *MegaCore IP Library Release Notes and Errata*.

Notice about Environment Variables

The `QUARTUS_ROOTDIR` and `SOPC_KIT_NIOS2` environment variables will be deprecated in a future release. If you have custom scripts that use these variables, you should be prepared to change them to be independent of these variables, or to set the environment variables yourself.

Revision Details

This section provides details about some of the Nios II EDS revisions shown in Table 1.

Nios II Software Build Tools for Eclipse

The Nios II Software Build Tools (SBT) for Eclipse is a set of plugins based on the popular Eclipse framework and the Eclipse C/C++ development toolkit (CDT) plugins. The Nios II SBT for Eclipse provides a consistent development platform that works for all Nios II processor systems. You can accomplish most Nios II software development tasks within Eclipse, including creating, editing, building, running, debugging, and profiling programs. The Nios II SBT for Eclipse is built from the ground up as a set of plugins to standard Eclipse. The Nios II SBT for Eclipse is based on the Nios II Software Build Tools, a set of powerful commands, utilities and scripts for creating applications, board support packages and software libraries, and managing build options.

Table 2 compares the Nios II Integrated Development Environment (IDE) with the Nios II SBT for Eclipse.

Table 2. Differences Between Nios II SBT for Eclipse and Nios II IDE

Feature	Nios II IDE	Nios II SBT for Eclipse
Types of projects	■ Application ■ Board support package (BSP)—Called a system library in the IDE ■ User Library (optional)	■ Application ■ BSP ■ User Library (optional)
Handling updates to the SOPC Builder system	Updates the BSP automatically	You must initiate the BSP update (in the Nios II menu, click Generate BSP)
Options for creating projects	Application from template. BSP is created automatically as part of application.	■ New Application and BSP from template ■ Standalone application ■ Standalone BSP ■ Standalone library
Handling updates to BSP	Entire BSP is rebuilt. This process can be time-consuming.	Smart rebuild only recompiles changed files
Compatibility with command-line flow	None	Full compatibility with the Nios II SBT in the Nios II Command Shell
Modifying BSP properties	Limited control over BSP properties through the System Library page of the Properties dialog box.	Full control over BSP properties through the BSP Editor
Nios II C2H Compiler	Supported	Not supported
Nios II Flash Programmer	Limited control over the contents of the flash image	■ Allows you to create complex flash images containing Executable and Linking Format Files (.elf), SRAM Object Files (.sof), and binary data files in specified locations ■ Allows you to save flash programmer settings for repeatable flash programming
Scripting and regression testing	Not Available	Project creation and build can be scripted using the SBT command line

Converting IDE Projects to the SBT for Eclipse

If you have a pre-existing Nios II IDE project, before you can continue development in the Nios II SBT for Eclipse, you must convert it to the SBT and import it to the Eclipse environment. To convert and import an IDE project to the Nios II SBT for Eclipse, execute the following steps:

1. Convert the IDE project to an SBT-based project with the **nios2-convert-ide2sbt** command line utility.
2. Import the converted project to the Nios II SBT for Eclipse.

For a detailed description of the conversion process, refer to “Porting Nios II IDE Projects to the Software Build Tools” in the *Using the Nios II Integrated Development Environment* appendix of the *Nios II Software Developer’s Handbook*. For a description of the process of importing a project, refer to “Importing a Command-Line Project” in the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer’s Handbook*.

Learning about the Nios II SBT for Eclipse

Refer to the following documents to help you get acquainted with the Nios II SBT for Eclipse and related tools:

- *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer’s Handbook*
- *Nios II Hardware Development Tutorial*
- The Comprehensive Nios II Embedded Design Suite support page
- The Nios II Software Build Tools for Eclipse support page

Errata Summary

Table 3 summarizes the issues that affect the Nios II Embedded Design Suite versions 8.1 through 9.1 SP2. For a detailed description of each issue, click on the issue name.

Refer to the Altera Knowledge Database for older errata and solutions.

Table 3. Nios II EDS Errata *(1)* (Part 1 of 6)

Added or Updated	Issue	Page	Affected Version				
			10.0	9.1 SP2	9.1 SP1	9.1	9.0
	Nios II SBT for Eclipse						
Sep 2010	Run Configuration Cannot Find Imported Custom Makefile Project	9	✓	—	—	—	—
	SBT for Eclipse Creates NicheStack BSP Incorrectly	10	✓	—	—	—	—
	The Restart Button in the Debugger Does Not Work	10	✓	✓	✓	✓	—
July 2010	Nios II SBT for Eclipse Cannot Step into GCC 3 Library Source Code	11	✓	✓	✓	✓	—
	Error Building Imported Project: ‘No rule to make target’	11	✓	—	—	—	—
	Error Building Imported Project: ‘target pattern contains no %’	12	✓	—	—	—	—
	Missing Nios II Perspective	13	✓	✓	✓	✓	—
	Error Debugging Imported Project: ‘Can't find a source file’	13	Fixed	✓	✓	✓	—
	Error Marker Persists on BSP Project After Build Error Corrected	13	✓	✓	✓	✓	—
	Assembly Language Source File Not Found	14	Fixed	✓	✓	✓	—
	‘Exclude from build’ Not Supported	14	✓	—	—	—	—
May 2010	Errors Creating or Importing Software Projects	14	✓	✓	✓	✓	—
	Nios II Options Do Not Appear in Eclipse	15	✓	✓	✓	✓	—

Table 3. Nios II EDS Errata *(1)* (Part 2 of 6)

Added or Updated	Issue	Page	Affected Version				
			10.0	9.1 SP2	9.1 SP1	9.1	9.0
Feb 2010	Build Errors on Software for Pre-Existing Design with SG-DMA	15	✓	✓	✓	✓	—
	Spurious System ID Mismatch Error	16	✓	✓	✓	✓	—
	Errors Debugging as Local C/C++ Application	16	✓	✓	✓	✓	—
Nov 2009	Error Messages on Console When Debugging	16	✓	✓	✓	✓	—
	java.lang.NullPointerException Error When Running Application	17	✓	✓	✓	✓	—
	Update Failure When Setting Empty Properties in the Nios II Properties Page	17	✓	✓	✓	✓	—
	Errors Converting Nios II IDE Multiprocessor Project	18	✓	✓	✓	✓	—
	DHCP Time-Out on EP3C120	18	✓	✓	✓	✓	—
	No Nios II Multiprocessor Collection Run Configuration	18	✓	✓	✓	✓	—
	Stop on Startup Option in Run Configuration Has No Effect	19	✓	✓	✓	✓	—
	Project Imported From Command Line Fails to Build Correctly	19	✓	✓	✓	✓	—
	Debugger Breaks in crt0.s Instead of main()	19	✓	✓	✓	✓	—
C2H Compiler							
Nov 2009	Error "java.lang.NoClassDefFoundError" During C2H Compilation	20	—	—	—	—	Fixed
Jul 2008	Functions Declared Without a Return Type Are Not Supported	20	✓	✓	✓	✓	✓
Oct 2007	Pre-7.1 Systems Are Not Supported	20	✓	✓	✓	✓	✓
	--src-dir SBT Argument Does Not Work With C2H	21	✓	✓	✓	✓	✓
May 2007	Accelerator Generation Failure If Tools Are Installed in Path With Spaces	21	✓	✓	✓	✓	✓
	The C2H Compiler Regenerates an Accelerator Unnecessarily	21	✓	✓	✓	✓	✓
	Error: c2h_accelerator_base_addresses.h: No such file or directory	22	✓	✓	✓	✓	✓
	Java Heap Space Exception if Quartus II Compilation is Enabled	22	✓	✓	✓	✓	✓
	Pointer Dereferences to Volatile Types	22	✓	✓	✓	✓	✓
Dec 2006	C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File	23	✓	✓	✓	✓	✓
Jun 2006	Array Elements in Structures Do Not Copy Correctly	23	✓	✓	✓	✓	✓
	Clean Build Causes Build Failure	24	✓	✓	✓	✓	✓
	Changing Build Configurations Produces Unexpected Results	24	✓	✓	✓	✓	✓
	Hardware Accelerators Remain After Deleting the Software Project *(2)*	24	✓	✓	✓	✓	✓
	Incorrect Results From Logical or Conditional Operation With Side-Effects	25	✓	✓	✓	✓	✓
	Launch SOPC Builder Button in C2H View	25	✓	✓	✓	✓	✓

Table 3. Nios II EDS Errata *(1)* (Part 3 of 6)

Added or Updated	Issue	Page	Affected Version				
			10.0	9.1 SP2	9.1 SP1	9.1	9.0
	Development Boards						
Jun 2006	Intermittent Failures While Accessing CompactFlash Card	25	✓	✓	✓	✓	✓
	Documentation Issues						
July 2010	Error Building Project: 'No rule to make target'	26	✓	—	—	—	—
	AN543 Contains Incorrect Information about Updating the Flash	27	✓	✓	✓	✓	—
	Incorrect Information about Floating-Point Instruction Precision	27	Fixed	✓	✓	✓	—
	Eclipse CDT Features Not Supported by Nios II Plugins	28	✓	✓	✓	✓	—
May 2010	Incorrect Information about Nested Exceptions	29	✓	✓	✓	✓	—
Feb 2010	Missing Documentation of Interrupt API Properties	30	✓	✓	✓	✓	—
	Valid Range of hal.log_flags is –1 to 3	31	✓	✓	✓	✓	—
Nov 2009	NicheStack TCP/IP Stack - Nios II Edition Does Not Support TFTP	31	✓	✓	✓	✓	✓
	Error Message After Renaming Project: "Resource is out of sync with the system"	32	✓	✓	✓	✓	—
	Obsolete Wiki URLs in Help Systems and Handbook	32	—	—	—	—	Fixed
	Missing Source Control Information	32	—	—	—	—	Fixed
Mar 2009	Compiler Flags for Building Custom Newlib	34	✓	✓	✓	✓	✓
Nov 2008	Error Message on Invalid Exception or Reset Vector	35	✓	✓	✓	✓	✓
Oct 2007	Nios II IDE Online Help Expand Buttons Do Not Work	36	✓	✓	✓	✓	✓
	Hardware Abstraction Layer						
Feb 2010	Missing Structure Member Errors in alt_log_printf.c with Small JTAG UART Driver	36	✓	✓	✓	✓	—
	"unused variable" Warning When Building alt_main.c	37	✓	✓	✓	✓	—
	Build Warnings in Performance Counter Driver Files	37	✓	✓	✓	✓	—
	Hardware Example Designs						
Mar 2007	Incorrect PLL Phase Shift in Stratix II and Cyclone II Designs	38	✓	✓	✓	✓	✓
	Software Examples						
Nov 2009	Hardware Tutorial Software Example Hangs on Some Boards	39	✓	✓	✓	✓	✓
Nov 2008	Simple Socket Server Example Displays Incorrect IP Address	39	✓	✓	✓	✓	✓
Jun 2006	Networking Examples	40	✓	✓	✓	✓	✓
	Flash Programmer						
Feb 2010	Unable to Configure FPGA from Flash with Parallel Flash Loader	40	Fixed	Fixed	Fixed	✓	✓
	Boot Failure or elf2flash Error Due to Incorrect Flash File Size	41	✓	✓	✓	✓	✓
May 2007	Delay When Creating New Flash Programmer Configuration	42	✓	✓	✓	✓	✓

Table 3. Nios II EDS Errata *(1)* (Part 4 of 6)

Added or Updated	Issue	Page	Affected Version				
			10.0	9.1 SP2	9.1 SP1	9.1	9.0
Dec 2006	elf2flash File Size Limit	42	✓	✓	✓	✓	✓
	Error Message “No such file or directory” When Programming Flash	42	✓	✓	✓	✓	✓
Hardware Simulation							
May 2010	Vectored Interrupt Controller Does Not Support VHDL Simulation Models	42	✓	✓	✓	✓	—
Dec 2006	Error “UNC paths are not supported” Launching ModelSim	43	✓	✓	✓	✓	✓
Jun 2006	Uninitialized .bss Variables in Simulation	43	✓	✓	✓	✓	✓
Host Platform							
July 2010	Nios II EDS Requires Administrative Permissions on Windows Vista	43	✓	✓	✓	✓	—
Nov 2009	Cannot Launch Nios II SBT for Eclipse on Red Hat Linux	44	✓	✓	✓	✓	—
Dec 2007	Windows/Cygwin: Nios II Processor Generation Failure	44	✓	✓	✓	✓	✓
Oct 2007	Windows Vista: Limited Support in Nios II EDS	45	✓	✓	✓	✓	✓
Nios II IDE							
Building Projects							
Mar 2009	Nios II IDE Command-Line Tools Select Wrong Workspace on Linux	45	✓	✓	✓	✓	✓
	Nios II IDE Command-Line Tools Hang on Windows	46	✓	✓	✓	✓	✓
Nov 2008	Nios II IDE Cannot Find stdio.h in Outline View	47	✓	✓	✓	✓	✓
Jul 2008	Nios II IDE Hangs With UNC Project Path	47	✓	✓	✓	✓	✓
	Build Failure with Nios II Advanced Exceptions, MMU, and MPU	47	✓	✓	✓	✓	✓
May 2007	Build Command Not Functional for BSPs Created With the Nios II SBT	47	✓	✓	✓	✓	✓
	Linker Errors with Dual-Port Memories	48	✓	✓	✓	✓	✓
	User-Managed BSP Settings Not Supported in Nios II IDE	48	✓	✓	✓	✓	✓
	URL Project Location Causes Project Creation Error	48	✓	✓	✓	✓	✓
Dec 2006	Compilation Error with Separate Exception Stack Option	49	✓	✓	✓	✓	✓
	Incorrect Stack and Heap Free Space Report	49	✓	✓	✓	✓	✓
	Nios II IDE Reports Problems Without Displaying Error in Console	49	✓	✓	✓	✓	✓
Debugging Projects							
Nov 2009	Method for Accessing MMU and MPU Registers in the Debugger	50	✓	✓	✓	✓	✓
Nov 2008	IDE Cannot Display Imported Profiling Data	50	✓	✓	✓	✓	✓
Jul 2008	Error Message when Downloading .elf File	51	✓	✓	✓	✓	✓
	Variable Casting Unsupported in ISS	51	✓	✓	✓	✓	✓
	Trace Debug Does Not Support Instruction-Related Exceptions	51	✓	✓	✓	✓	✓
Oct 2007	Trace Debug Does Not Support the JMPI Instruction	51	✓	✓	✓	✓	✓

Table 3. Nios II EDS Errata *(1)* (Part 5 of 6)

Added or Updated	Issue	Page	Affected Version				
			10.0	9.1 SP2	9.1 SP1	9.1	9.0
May 2007	Cannot Locate Source Code in Driver Files Shared by Multiple Projects	52	✓	✓	✓	✓	✓
	Console Window Is Not Updated After ISS Error	52	✓	✓	✓	✓	✓
	ISS Fails on Designs Containing Triple Speed Ethernet MAC or SG-DMA Components	52	✓	✓	✓	✓	✓
	Memory Window Sets Control Register Values Incorrectly	53	✓	✓	✓	✓	✓
Dec 2006	Programs That Interact With a Terminal Console on Windows Do Not Work	53	✓	✓	✓	✓	✓
	"Run as ModelSim" in the Nios II IDE Fails	53	✓	✓	✓	✓	✓
	The Restart Command on the Run Menu Does Not Work	53	✓	✓	✓	✓	✓
Jun 2006	Watchpoints Do Not Work on Certain Variables	54	✓	✓	✓	✓	✓
Navigating Projects							
Dec 2006	Nios II IDE Freezes While Displaying the Splash Screen	54	✓	✓	✓	✓	✓
	Internal Error When Double-Clicking on a Large Objdump File	54	✓	✓	✓	✓	✓
Jun 2006	C/C++ Scanner Does Not Support Certain C/C++ Constructs	55	✓	✓	✓	✓	✓
Nios II SBT							
July 2010	Error Running Nios II Project: 'Downloading ELF Process failed'	55	✓	✓	✓	✓	—
May 2010	BSP Editor Does Not Show Command-Line Help	56	✓	✓	✓	✓	—
Feb 2010	BSP Not Updated for Memory Size Changes in SOPC Builder	56	✓	✓	✓	✓	—
Nov 2009	Build Errors on Interrupt API Calls	56	✓	✓	✓	✓	—
	BSP Editor Incorrectly Allows Mixed Drivers	57	—	—	—	Fixed	✓
	nios2-bsp --help Displays Incorrect List of BSP Types	58	—	—	—	Fixed	✓
	Java Heap Space Error When Creating BSP	58	—	—	—	—	Fixed
Oct 2007	SBT Fails if Nios II EDS is Installed in a Path Containing Spaces	59	✓	✓	✓	✓	✓
May 2007	User-Managed BSP Settings Not Supported in Nios II IDE	48	✓	✓	✓	✓	✓
	Upper-Case File Extensions Not Supported	59	✓	✓	✓	✓	✓
Peripherals							
May 2010	Vectored Interrupt Controller Does Not Support VHDL Simulation Models	42	✓	✓	✓	✓	—
Jul 2008	Unaligned Transfers of Small Payloads Fail on SG-DMA	60	✓	✓	✓	✓	✓
May 2007	DMA Controller Always Busy in Burst Mode	60	✓	✓	✓	✓	✓
	Non-System-Wide Reset Can Cause Improper Initialization of Mailbox Core	60	✓	✓	✓	✓	✓
Target Software							
Nov 2009	NicheStack tcp_wake() Function Does Not Awaken All select() Sleepers	61	✓	✓	✓	✓	—
	EIC-Based Nios II System Crashes With NicheStack	61	✓	✓	✓	✓	—
Mar 2009	Gigabit Ethernet Performance Issues	61	—	—	—	—	Fixed

Table 3. Nios II EDS Errata *(1)* (Part 6 of 6)

Added or Updated	Issue	Page	Affected Version				
			10.0	9.1 SP2	9.1 SP1	9.1	9.0
Dec 2006	stdio Does Not Work with MicroC/OS-II and Small C Library	62	✓	✓	✓	✓	✓
Jun 2006	cout From MicroC/OS-II Task Does Not Send Data to stdout	62	✓	✓	✓	✓	✓
	malloc(), realloc() Failures With MicroC/OS-II	62	✓	✓	✓	✓	✓
Toolchain (gcc, gdb, etc.)							
Sep 2010	nios2-console Does Not Work in the GCC 4 Command Shell	63	✓	—	—	—	—
July 2010	C++ Exceptions Unsupported in a Multi-Threaded Environment	63	✓	✓	✓	✓	✓
	C++ Exceptions Unsupported by the Nios II GCC 4 Toolchain	64	✓	—	—	—	—
	memcpy() Optimization Misalignment	64	✓	✓	✓	✓	✓
Feb 2010	Unable to Configure FPGA from Flash with Parallel Flash Loader	40	—	—	Fixed	✓	✓
Dec 2006	GNU Assembler Does Not Accept the --defsym Flag	64	✓	✓	✓	✓	✓

Notes to Table 3:

(1) Refer to the Altera Knowledge Database for older errata and solutions.

(2) For further information, refer to the *Nios II C2H Compiler User Guide*.

Nios II SBT for Eclipse Errata

This section describes in detail the issues related to the Nios II SBT for Eclipse.

Run Configuration Cannot Find Imported Custom Makefile Project

After you import a project using the **Import Custom Makefile for Nios II Software Build Tools Project** option, the Nios II SBT fails to recognize the imported custom makefile as a Nios II C/C++ application project. As a result, the project name does not appear in the list in the run configuration.

Workaround

1. In the **Run Configuration** dialog box, select any visible Nios II C/C++ project. The project need not be related to your custom makefile project.

   ☞ If there is no Nios II C/C++ project in your workspace, create a dummy project, such as Hello World, through the File menu, by clicking **New**, **Nios II Application and BSP from Template**.

2. Turn on **Enable browse for file system ELF file**.
3. Browse to the correct **.elf** file for your custom makefile project to complete creation of a run configuration.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

SBT for Eclipse Creates NicheStack BSP Incorrectly

NicheStack applications encounter runtime failures if you use the following steps to create a NicheStack BSP:

1. Create a new Nios II Board Support Package with the BSP type selected as Micrium MicroC/OS-II. Notice that the HAL component generates **alt_syscall.h**.
2. Using the Nios II BSP Editor, in the **Software Packages** tab, select **altera_iniche**. Save and build the BSP. Notice that the NicheStack (iniche) component generates another **alt_syscall.h**.

public.mk includes the HAL header files before the NicheStack header files. This causes the HAL **alt_syscall.h** to be used in the build, instead of the required **alt_syscall.h** from the NicheStack component.

If a NicheStack application links to such a BSP, the application builds successfully, but encounters runtime failures that are difficult to debug.

Workaround

Manually edit **public.mk**. Under `Software Component & Driver Include Paths`, rearrange the include paths to include the NicheStack header files before the HAL header files.

In general, include paths should appear in the following order:

1. Drivers
2. Software packages
3. Extended BSP type (e.g., MicroC/OS-II)
4. Base BSP type (HAL)

Example 1 shows include path directories added in the correct order.

Example 1. Include Path Ordering in **public.mk**

```
ALT_INCLUDE_DIRS += $(ALT_LIBRARY_ROOT_DIR)/drivers/inc
ALT_INCLUDE_DIRS += $(ALT_LIBRARY_ROOT_DIR)/drivers/inc/iniche
ALT_INCLUDE_DIRS += $(ALT_LIBRARY_ROOT_DIR)/iniche/inc
ALT_INCLUDE_DIRS += $(ALT_LIBRARY_ROOT_DIR)/iniche/src/h
ALT_INCLUDE_DIRS += $(ALT_LIBRARY_ROOT_DIR)/iniche/src/h/nios2
ALT_INCLUDE_DIRS += $(ALT_LIBRARY_ROOT_DIR)/iniche/src/nios2
ALT_INCLUDE_DIRS += $(ALT_LIBRARY_ROOT_DIR)/UCOSII/inc
ALT_INCLUDE_DIRS += $(ALT_LIBRARY_ROOT_DIR)/HAL/inc
```

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

The Restart Button in the Debugger Does Not Work

When debugging in the SBT for Eclipse, if you click the **Restart** icon, execution is suspended, with the following error message:

```
Don't know how to run. Try "help target."
```

Workaround

Terminate the program, download it again, and start the debugger.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Nios II SBT for Eclipse Cannot Step into GCC 3 Library Source Code

If your project uses the Nios II GCC 3 toolchain, when debugging with the Nios II SBT for Eclipse, you might be unable to step into library source code. A message such as the following message appears:

```
Resource '/xxx/xxx.c' does not exist.
```

Workaround

Upgrade to the Nios II EDS v10.0 or later and use the Nios II GCC 4 toolchain.

Solution Status

Not fixed

Error Building Imported Project: 'No rule to make target'

If an application or library project is created with v9.1 SP2 or earlier (using the GCC 3 toolchain), and you import it to the v10.0 SBT for Eclipse using the GCC 4 toolchain, you might see the build error shown in Example 2.

Example 2.

```
make all
Info: Building ../hw_bsp
make --no-print-directory -C ../hw_bsp
make[1]: *** No rule to make target `/cygdrive/c/tmp/
verilog_niosII_stratixII_2s60_standard/software/hw_bsp/
alt_sys_init.c', needed by `all'. Stop.
make: *** [../hw_bsp-recurs-make-lib] Error 2
```

This error can also happen if you import a GCC 3 project to the v10.0 SBT for Eclipse using the GCC 3 toolchain, and later change the toolchain to **MinGW Nios II GCC4**.

This issue only affects platforms running the Windows operating system.

Workaround

To avoid this error, follow these steps:

1. Edit the application project and library project makefiles and add the italicized code as shown in Example 3.

Example 3.

```
#------------------------------------------------------------------------------
# The adjust-path macro
#
# If COMSPEC is defined, Make is launched from Windows through
# Cygwin. This adjust-path macro will call 'cygpath -u' on all
# paths to ensure they are readable by Make.
#
# If COMSPEC is not defined, Make is launched from *nix, and no adjustment
# is necessary
#------------------------------------------------------------------------------

# Beginning of inserted code
ifndef COMSPEC
ifdef ComSpec
COMSPEC = $(ComSpec)
endif # ComSpec
endif # !COMSPEC
#end of inserted code

ifdef COMSPEC
adjust-path = $(shell cygpath -u "$1")
adjust-path-mixed = $(shell cygpath -m "$1")
else
adjust-path = $1
adjust-path-mixed = $1
endif
```

2. Right-click the BSP project, point to **Nios II** and click **Generate BSP**.
3. Clean and rebuild the application and library projects.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Error Building Imported Project: ‘target pattern contains no %’

If your application or library makefile uses an absolute path and is generated with the GCC 3 toolchain, and you import it to the v10.0 SBT for Eclipse using the GCC 4 toolchain, you get the following build error:

```
makefile:587: *** target pattern contains no '%'. Stop.
```

Workaround

Recreate the project using the SBT for Eclipse v10.0.

Alternatively, use the GCC 3 toolchain.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Missing Nios II Perspective

If you launch the Nios II SBT for Eclipse from a universal naming convention (UNC) path, the Nios II perspective does not appear in Eclipse.

This issue appears only on platforms running the Windows operating system.

Workaround

Do not use a UNC path to launch the Nios II SBT for Eclipse. For example, if the SBT for Eclipse is installed on a network drive, map the network path to a Windows drive letter.

Solution Status

Not fixed

Error Debugging Imported Project: ‘Can't find a source file’

If you create a project using the Nios II SBT in the Nios II Command Shell, and then import the project to Nios II SBT for Eclipse for debugging, the debugger cannot find source files located outside of project folder.

Example of error message:

```
Can't find a source file at "../src/test.c"
```

Workaround

To link the affected source files to the project, follow these steps:

1. Right-click the project and click **Properties** to open the **Properties** dialog box.
2. Click **+** to expand **C/C++ General**, and click the **Paths and Symbols** tab.
3. Click the **Source Location** tab.
4. Click **Link Folder** and navigate to the folder containing the source file.

Solution Status

Fixed in v10.0 of the Nios II EDS

Error Marker Persists on BSP Project After Build Error Corrected

In the Nios II SBT for Eclipse, if an application project fails to build, error markers appear in the BSP project, if the build error message contains a path to BSP project source files. Even after you have rebuilt the application project without errors, the error markers persist in the BSP project.

Workaround

Rebuild your BSP project.

Alternatively, if your application will always be associated with the same BSP, set the Eclipse project references by following these steps:

1. Right-click your application project, point to **Properties** and click **Project References**.
2. Select the desired BSP for your application.
3. Build your application project again to clear the error markers in the BSP project.

☞ If you later link your application project with a different BSP, you must manually correct the Eclipse project references.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Assembly Language Source File Not Found

The Nios II SBT for Eclipse does not support assembly language files with the upper case file extension **.S**. If your project includes a file with an upper case extension, the compiler is not able to find the assembly language file during build.

Workaround

Rename the assembly language files with the lower-case extension **.s**.

Solution Status

Fixed in v10.0 of the Nios II EDS

‘Exclude from build’ Not Supported

In the Nios II SBT for Eclipse v9.1 SP2 and earlier, you can use the **Exclude from build** command to omit a source file from your project at build time. Starting with v10.0, **Exclude from build** is replaced by the **Remove from Nios II build** and **Add to Nios II build** commands.

Due to an Eclipse platform limitation, **Exclude from build** still appears in the project context menu in v10.0. However, it has no effect.

Workaround

To exclude a source file from your project when it is built, on the project context menu, click **Remove from Nios II build**. To restore it to the build, click **Add to Nios II build**.

Errors Creating or Importing Software Projects

If you place software projects in your Eclipse workspace directory, you might experience project-related errors. For example, you might see one of the following error messages:

-
  ```
  Unable to create project
  Reason: Unable to create project in workspace directory
  ```

-
  ```
  Failed to import
  Reason: Unable to import project
  ```

Workaround

Switch to a different workspace, or create a new workspace, separate from your project files. To switch workspaces or create a workspace, on the File menu, click **Switch Workspace**.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Nios II Options Do Not Appear in Eclipse

When Nios II SBT for Eclipse starts, the Nios II plugins might fail to load, resulting in the following symptoms:

- The Nios II perspective is not available
- No Nios II items appear when you click **New** in the File menu

Eclipse can behave this way if you have installed the Nios II EDS in the same directory as a previous installation.

When the Altera Complete Design Suite (ACDS) installer installs the Nios II EDS, you specify a root directory for the EDS file structure. For example, if the Nios II EDS 9.1 is installed on the Windows operating system, the root directory might be **c:\altera\91\nios2eds**. For simplicity, Altera documentation refers to this directory as <*Nios II EDS install path*>.

When the ACDS installer removes the Nios II EDS, it leaves behind some files in <*Nios II EDS install path*>. If you reinstall the Nios II EDS in the same directory, these leftover files might cause the Nios II SBT for Eclipse to work incorrectly.

Workaround

To correct this problem, execute the following steps:

1. Uninstall the Nios II EDS.
2. Delete <*Nios II EDS install path*>, including all subdirectories, before reinstalling the Nios II EDS.

   Alternatively, reinstall the Nios II EDS in a different directory.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Build Errors on Software for Pre-Existing Design with SG-DMA

If you have a Nios II system generated with SOPC Builder v. 9.0 or earlier, and it contains the Scatter-Gather DMA (SG-DMA) component, you cannot build software for it with the Nios II SBT for Eclipse v. 9.1 or later. If you attempt to do so, the Nios II compiler reports errors.

Workaround

Regenerate the design with SOPC Builder v. 9.1 or later before attempting to compile the software in the Nios II SBT for Eclipse.

Solution Status

Not fixed

Spurious System ID Mismatch Error

You might see the following error message when attempting to run or debug a Nios II software project, or to edit a launch configuration:

```
[Target Connection]:System ID mismatch - actual: "0xffffffff", expected: "<system id>"
```

<system id> is replaced by the system ID in your software project. When this error occurs, you cannot run or debug software.

This error normally means that the FPGA is unconfigured, or configured with a **.sof** file that does not correspond to your software project. However, the error occasionally appears even when the FPGA is configured with the correct **.sof**.

Workaround

After verifying that the FPGA is configured with the correct **.sof**, you can use one of the following methods to work around this issue:

- In the **Run Configuration** or **Debug Configuration** dialog box, on the **Target Connection** tab, click **Refresh Connections** repeatedly until the error message disappears.
- Delete the launch configuration and create a new launch configuration.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Errors Debugging as Local C/C++ Application

If you try to debug a Nios II software project as the Local C/C++ Application launch configuration type, you see one of the following errors, depending on your platform:

- Windows: “Error creating session”
- Linux: “Execution is suspended because of error”

The Nios II debug perspective fails to open.

This is expected CDT behavior in the Eclipse platform. Local C/C++ Application is the launch configuration type for a standard CDT project. To invoke the Nios II plugins, you must use a Nios II launch configuration type.

Workaround

Always debug Nios II software projects as the Nios II Hardware launch configuration type.

Error Messages on Console When Debugging

When debugging, you might see the following messages in the Nios II SBT for Eclipse console:

```
No symbol "auto" in current context.
No symbol "new" in current context.
```

These are normal messages from the GNU Debugger (GDB) console, not error messages. This issue does not impact the debug process.

Workaround

No workaround needed.

Solution Status

None needed.

java.lang.NullPointerException Error When Running Application

When you attempt to run or debug an application, the Nios II SBT for Eclipse might show an internal error message similar to the following:

```
Launching New_configuration java.lang.NullPointerException
```

This error occurs in either of the following circumstances:

- You attempt to run your project without rebuilding after performing a `make clean`.
- You delete a run configuration belonging to your project while the project is running.

You must restart Nios II SBT for Eclipse to recover from this error.

Workaround

Build the project before attempting to run or debug it. Do not delete a run configuration while it is in use.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Update Failure When Setting Empty Properties in the Nios II Properties Page

If you use the Nios II Properties page to update more than one board support package (BSP) setting to the empty string (`""`), the BSP is not updated.

Workaround

You can use either of the following workarounds to set multiple BSP settings to the empty string:

- Use the keyword `none` instead of `""` to set settings to the empty string.
- Use the BSP Editor instead of the Nios II Properties page.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Errors Converting Nios II IDE Multiprocessor Project

If you try use the **nios2-convert-ide2sbt** utility to convert an IDE multiprocessor project to the Nios II SBT, error messages appear, and the project fails to generate properly.

Workaround

Create new Nios II SBT application and BSP projects using project settings equivalent to the original Nios II IDE project settings.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

DHCP Time-Out on EP3C120

If you create a project based on the web server template targeting the Altera Embedded Systems Development Kit, Cyclone® III Edition (EP3C120) board, the system is unable to get an IP address at startup. DHCP times out and the system reverts to the default IP address.

Workaround

Create the project in the Nios II Command Shell and then import it to the Nios II SBT for Eclipse, by executing the following steps:

1. In the Nios II Command Shell, navigate to the following path:

   *<Nios II EDS install path>***/examples/***<language>***/niosII_cycloneIII_3c120/tserd/ software_examples/app/web_server_tse_3c120_rgmii**

   where *<language>* is either **vhdl** or **verilog**.

2. Type the following command to create a web server example for the EP3C120 kit:

   ```
   create-this-app↵
   ```

3. Import the command-line project to the Nios II SBT for Eclipse.

For information about importing command-line projects, refer to "Importing a Command-Line Project" in the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

No Nios II Multiprocessor Collection Run Configuration

The multiprocessor collection run configuration is not available in the Run menu. Therefore, you cannot use a single run configuration to download Executable and Linking Format Files (**.elf**) to run on multiple Nios II processors in the Nios II SBT for Eclipse.

Workaround

Create a separate run configuration for each Nios II processor in the hardware design, and launch them one at a time

Solution Status

Not fixed

Stop on Startup Option in Run Configuration Has No Effect

Turning on **Stop on Startup at** in the **Debugger** tab of a run configuration has no effect.

Workaround

You can manually set any breakpoint in the debug perspective.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Project Imported From Command Line Fails to Build Correctly

If you try to import a command-line project to the Nios II SBT for Eclipse, and your project has source files located outside the project folder, the Nios II SBT for Eclipse fails to build your project because the makefile does not include source files located outside the project folder.

Workaround

You can work around this issue with either of the following methods:

- When importing the project, disable source management through the **Nios II Application Properties** or **Nios II Library Properties** dialog box.

  ☞ You must manually keep your makefile updated to reflect source files added to or removed from the project.

- Manually link the affected source files to the project. Right-click the project, click **Properties**, expand **C/C++ General**, select **Paths and Symbols** and then **Source Location**, and click **Link Folder**.

Solution Status

This issue will be fixed in future release of the Nios II EDS.

Debugger Breaks in crt0.s Instead of main()

If you attempt to launch a debug session by right clicking the project, without first creating a run configuration, the debugger breaks in the initialization code in **crt0.s**, rather than at the top of the `main()` function. You must explicitly create the run configuration before launching the debug session.

Workaround

You can work around this issue with either of the following methods:

- Create the run configuration with the **Run Configuration** dialog box. You can then debug with this run configuration by right-clicking the project, clicking **Debug As**, and selecting the debug configuration. The program correctly breaks at `main()`.
- Manually set a breakpoint at `main()`.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

C2H Compiler Errata

This section describes in detail the issues related to the Nios II C2H Compiler.

Error "java.lang.NoClassDefFoundError" During C2H Compilation

C2H compilation fails with the following error:

```
java.lang.NoClassDefFoundError: com/altera/c2h/C2Hgenerate
Exception in thread "main" make: *** [c2h_hdl-t] Error 1
```

This error occurs when you invoke C2H from the Nios II IDE on the Windows operating system. This problem does not affect Linux installations, nor C2H compilation from the Nios II command shell.

Workaround

To correct the problem, visit mySupport and request the Nios II EDS 8.1 Patch 0.01 (PC version).

Solution Status

Fixed in v. 9.0 of the Nios II EDS

Functions Declared Without a Return Type Are Not Supported

The C2H compiler does not support functions without an explicitly declared return type.

Workaround

If you are using the implicit `int` return type, declare the return type explicitly. If your function has no return value, declare it as `void`.

Solution Status

Not fixed

Pre-7.1 Systems Are Not Supported

Starting in v. 7.1, the C2H Compiler does not support SOPC Builder systems created in SOPC Builder 7.0 or earlier (systems based on a **class.ptf** file).

Workaround

Before using the C2H Compiler, you must upgrade your SOPC Builder system. Open the system in SOPC Builder v. 7.1 or later. SOPC Builder prompts you to convert the system, creating a ***_hw.tcl** file compatible with the C2H Compiler.

Solution Status

Not fixed

--src-dir SBT Argument Does Not Work With C2H

When using the C2H Compiler with Nios II SBT, an error occurs when using the `nios2-app-generate-makefile` argument `--src-dir` to specify a directory of source files. An example of the error is:

```
multiple definition of 'my_accelerated_function'
```

Workaround

Specify source files individually using the `--src-files` argument.

Solution Status

Not fixed

Accelerator Generation Failure If Tools Are Installed in Path With Spaces

If the path to your installation of the Altera Design Suite contains spaces, the C2H Compiler fails to generate the accelerators.

Workaround

Reinstall the tools to a path containing no spaces.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

The C2H Compiler Regenerates an Accelerator Unnecessarily

The C2H Compiler might regenerate an accelerator even when the accelerated function is unchanged. This problem can result from changes to files included by the C file containing the accelerated function. The C2H Compiler fails to check that the generated hardware description language (HDL) matches the previously generated HDL causing the system to be regenerated.

Workaround

To avoid this issue move the accelerator-specific information from the include file to a separate include file. This workaround prevents regeneration of the system when the HDL is unchanged.

Solution Status

Not fixed

Error: c2h_accelerator_base_addresses.h: No such file or directory

When a C2H accelerator is compiled for the first time, the following compile-time error can result if the **Analyze all accelerators** option is selected:

```
c2h_accelerator_base_addresses.h: No such file or directory.
```

Workaround

Click **Build software and generate SOPC Builder system** and build once before building with the **Analyze all accelerators** option.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Java Heap Space Exception if Quartus II Compilation is Enabled

If your design contains a C2H accelerator, and you select **Build software, generate SOPC Builder system, and run Quartus II compilation** in the C2H view, you might see the following error during Quartus® II compilation:

```
Exception in thread "main" java.lang.OutOfMemoryError: Java
heap space

make: *** [c2h_hdl-t] Error 1
```

Workaround

Select the **Build software and generate SOPC builder system** option in the C2H Compiler settings window, and then manually launch the Quartus II software to compile the design.

Solution Status

Not fixed

Pointer Dereferences to Volatile Types

The C2H Compiler treats pointer dereferences to a volatile type as if they alias all other pointer dereferences. Pointers that are restrict-qualified are treated the same way.

The two loops in Example 4 cannot be scheduled concurrently because the `volatile` qualification overrides the `__restrict__` pragma.

Example 4. Non-Concurrent Loops

```
volatile int * __restrict__ fifo_rd = FIFO_RD_BASE;
volatile int * __restrict__ fifo_wr = FIFO_WR_BASE;
for ()
{
  *fifo_wr = ....;
}
for ()
{
  ... = *fifo_rd;
}
```

Workaround

Divide the function into multiple interrupt request (IRQ)-enabled accelerators that are launched concurrently from the processor, and use FIFO buffers to communicate between them.

Solution Status

Not fixed

C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File

When accelerating a function in a file, the C2H Compiler cannot link subfunctions that are defined in a different file.

Workaround

Include all subfunctions called by the accelerated function within the same source code file.

Solution Status

Not fixed

Array Elements in Structures Do Not Copy Correctly

C2H accelerators do not correctly copy array elements that are elements of structures.

In Example 5, the `a` and `b` elements of the structure copy correctly, but the `buf` element does not. After this assignment, `struct_a` equals `{9, 8, {3, 3, 3, 3}}`.

Example 5. Array Elements of Structs

```
typedef struct my_struct {
  int a;
  int b;
  int buf[BUF_SIZE];
}MY_STRUCT;
MY_STRUCT struct_a = {1, 2, {3, 3, 3, 3}};
MY_STRUCT struct_b = {9, 8, {7, 7, 7, 7}};
struct_a = struct_b;
```

Workaround

Copy the array elements explicitly, as shown in Example 6.

Example 6. Copying Array Elements Explicitly

```
{
  int i=0;
  do
    {
      struct_a.buf[i] = struct_b.buf[i];
      i++;
    } while (i<LENGTH_OF_BUF_ELEMENT)
}
```

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Clean Build Causes Build Failure

Performing a clean build on a Nios II IDE project that contains a hardware accelerator can cause the next build to fail in the IDE, because the clean build erroneously deletes a file required by the C2H Compiler.

Workaround

Do not perform a clean build on projects that use hardware accelerators. If you have already performed a clean build, recompile with option **Build software, generate SOPC Builder system, and run Quartus II compilation** to regenerate the necessary files.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Changing Build Configurations Produces Unexpected Results

The C2H Compiler does not support multiple build configurations (for example Release or Debug) in the Nios II IDE. After creating one or more accelerators in a particular configuration, the C2H Compiler produces undefined results if you switch to a different build configurations and create more accelerators.

Workaround

For a specific SOPC Builder system and Nios II IDE project, specify C2H accelerators in only one build configuration. You can use multiple build configurations, as long as only one configuration specifies C2H Compiler settings.

Solution Status

Not fixed

Hardware Accelerators Remain After Deleting the Software Project

If a system contains C2H accelerators, deleting the software project that defines the accelerators does not remove the accelerators from the hardware system, and the accelerator logic remains in the SOPC Builder system.

Workaround

To remove an accelerator from a system, delete the accelerator from the C2H view in the Nios II IDE first, and then recompile the software project. The C2H Compiler then removes the accelerator from the SOPC Builder system. Once the compilation is complete then the software application can be deleted from the workspace.

Solution Status

Not fixed

Incorrect Results From Logical or Conditional Operation With Side-Effects

The C2H Compiler always evaluates both operands of logical (&&, `||`) and conditional (`?:`) operators. This is different from expected American National Standards Institute (ANSI) C behavior, where operands are evaluated left-to-right, and unnecessary operands are skipped.

For example, in the expression `(i-- && j--)`, if the value of `i` is zero, ANSI C does not evaluate the right-hand-side (RHS) expression, and `j` is not decremented. By contrast, the C2H Compiler evaluates both sides, decrementing `j`.

The following expressions are other examples that might be affected by this issue:

```
if (i-- || j++)
    ...;


a = ((cond == 1)?  i++ : j++);
```

Workaround

Use logical and conditional operations whose operators have no side effects. Operations with side effects include pre- and post- increment and decrement operations (`++`, `--`), memory operations (`*`, `[]`, `.`, `->`), and function calls.

Solution Status

Not fixed

Launch SOPC Builder Button in C2H View

When the Nios II IDE workspace contains multiple projects with multiple system libraries, the incorrect SOPC Builder system might open when you click **Launch SOPC Builder** in the C2H view.

Workaround

Launch SOPC Builder from the Quartus II software. Alternatively, keep only one system library project open at a time while using the C2H Compiler.

Solution Status

Not fixed

Development Board Errata

This section describes in detail the Nios II EDS issues related to Nios development boards.

Intermittent Failures While Accessing CompactFlash Card

The Nios II Development Kit version 5.0 and later includes a CompactFlash controller peripheral suitable for interfacing to CompactFlash cards in True IDE mode on Nios development boards. For True IDE mode to operate, CompactFlash cards require that the `ATASEL_N` input be driven to ground during power-up.

The CompactFlash controller peripheral includes a configurable power register, used to cycle power to CompactFlash cards from Nios II software through a metal oxide semiconductor field-effect transistor (MOSFET) on the Nios development boards. However, in certain development boards, power to the CompactFlash card does not turn off completely during this power cycle operation. Because of this condition, the CompactFlash might not sample the `ATASEL_N` pin during the power-cycle operation after FPGA configuration when this pin is driven to ground. Instead, the CompactFlash card might sample the `ATASEL_N` pin when power is first applied to the development board, when I/O is not yet driven by the FPGA (before FPGA configuration).

Workaround

If you encounter errors with CompactFlash when using the Nios development boards, try one of the following solutions:

- Use a different CompactFlash card. Certain cards are more susceptible to the power-cycling issue than others.
- Modify the Nios development board. This is recommended if you are familiar and comfortable with board-level modifications. Disconnect pin 9 (`ATASEL_N`) on the CompactFlash socket on your Nios development board and tie this pin to ground.

☞ The CompactFlash socket uses a staggered numbering on the pins (starting from pin 1: 1, 26, 2, 27, ...); refer to the CompactFlash Association specification for right-angle surface-mount connectors for exact specifications on this connector. This modification permanently enables True IDE mode operation.

Solution Status

Not fixed

Documentation Errata

This section describes in detail the Nios II EDS documentation issues.

Error Building Project: ‘No rule to make target’

The *Getting Started with Graphical User Interface* and *Getting Started from the Command Line* chapters of the *Nios II Software Developer’s Handbook* do not include the following information:

You cannot link Nios II GCC 3 projects with Nios II GCC 4 projects. Your application, library and BSP projects must all use the same Nios II GCC toolchain.

If you switch between Nios II GCC 3 and Nios II GCC 4, make sure you run **make clean** on your application, library and BSP projects before rebuilding.

When importing a GCC 3 BSP, such as a project created with v9.1 SP2 or earlier, to v10.0 SBT for Eclipse using the GCC 4 toolchain, after importing the BSP, regenerate the makefile.

If you do not follow these rules, you might see the following error:

```
make[1]: *** No rule to make target `/cygdrive/c/.../bsp/
alt_sys_init.c', needed by 'all'. Stop.
```

```
make: *** [../bsp/-recurs-make-lib] Error 2
```

Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

AN543 Contains Incorrect Information about Updating the Flash

"Updating the Flash Option Bits on the Cyclone III Development Board" in AN543, *Debugging Nios II Software Using the Lauterbach Debugger* contains incorrect information about updating the flash.

Workaround

The correct information is as follows:

Updating the Flash on the Cyclone III Development Board

To restore the flash factory image, refer to "Restoring the Factory Design to the Flash Device" in the *Cyclone III Development Kit User Guide*.

Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

Incorrect Information about Floating-Point Instruction Precision

In "Arithmetic Logic Unit" in the *Processor Architecture* chapter, the *Nios II Processor Reference Handbook* incorrectly states that the Nios II compiler treats floating-point constants as double-precision numbers.

Workaround

The correct information is as follow:

All the floating-point custom instructions are single-precision operations. Double-precision floating-point operations are implemented in software.

When the floating-point custom instructions are not present, the Nios II compiler treats floating-point constants as double-precision values. However, with the floating-point custom instructions, the Nios II compiler treats floating-point constants as single-precision numbers by default. This allows all floating-point expressions to be evaluated in hardware, at a possible cost in precision.

If you do not wish floating-point constants to be cast down to single precision values, append `L` to each constant value, to instruct the compiler to treat the constant as a double-precision floating-point value. In this case, if an expression contains a floating-point constant, each term in the expression is cast to double precision. As a result, the expression is computed with software-implemented double-precision arithmetic, at a possible cost in computation speed.

Table 4 shows code examples using floating-point constants, indicating how each computation is implemented.

Table 4. Floating-Point Constant Examples

Example Code	Floating-Point Custom Instructions Present?	Precision	Implementation
`b = a × 4.67`	No	Double	Software
`b = a × 4.67f`	No	Single	Software
`b = a × 4.67L`	No	Double	Software
`b = a × 4.67`	Yes	Single	Hardware
`b = a × 4.67f`	Yes	Single	Hardware
`b = a × 4.67L`	Yes	Double	Software

Solution Status

Fixed in v10.0 documentation

Eclipse CDT Features Not Supported by Nios II Plugins

The following information is missing from the *Getting Started with Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*:

The features listed in the left column of Table 5 are supported by the Eclipse CDT plugins, but are not supported by Nios II plugins. The right column lists alternative features supported by the Nios II plugins.

Table 5. Eclipse CDT Features Not Supported by the Nios II Plugins (Part 1 of 2)

Unsupported CDT Feature	Alternative Nios II Feature
New Project Wizard C/C++ ■ C Project ■ C++ Project ■ C/C++ ■ Convert to a C/C++ Project ■ Source Folder	To create a new project, use one of the following Nios II wizards: ■ Nios II Application ■ Nios II Application and BSP from Template ■ Nios II Board Support Package ■ Nios II Library
Build configurations Right-click project and point to **Build Configurations**	The Nios II plugins only support a single build configuration.
Exclude from Build (from v10.0 onwards) Right-click source files	Use **Remove from Nios II Build** and **Add to Nios II Build.** See "'Exclude from build' Not Supported" on page 14.

Table 5. Eclipse CDT Features Not Supported by the Nios II Plugins (Part 2 of 2)

Unsupported CDT Feature	Alternative Nios II Feature
Project Properties	
C/C++ Build	
■ Builder Settings	
■ ■ Makefile generation	By default, the Nios II SBT generates makefiles automatically.
■ ■ Build location	The build location is fixed.
■ Behaviour	
■ ■ Build on resource save (Auto build)	
■ Build Variables	
■ Discovery Options	
■ Environment	
■ Settings	
■ Tool Chain Editor	
■ ■ Current builder	
■ ■ Used tool	To change the toolchain, use the **Current tool chain** option
C/C++ General	
■ Enable project specific settings	
■ Documentation	
■ File Types	
■ Indexer	
■ ■ Build configuration for the indexer	The Nios II plugins only support a single build configuration.
■ Language Mappings	
■ Paths and Symbols	Use **Nios II Application Properties** and **Nios II Application Paths**
Window Preferences	
C/C++	
■ Build scope	
■ Build project configurations	The Nios II plugins only support a single build configuration.
■ Build Variables	
■ Environment	
■ File Types	
■ Indexer	
■ ■ Build configuration for the indexer	The Nios II plugins only support a single build configuration.
■ Language Mappings	
■ New CDT project wizard	

Incorrect Information about Nested Exceptions

“Exception Processing” in the *Programming Model* chapter of the *Nios II Processor Reference Handbook* incorrectly states that multiple interrupts with different requested interrupt levels (RILs) must not be assigned to the same shadow register set.

Workaround

The correct information is as follows:

Multiple interrupts with different RILs can be assigned to a single shadow register set. However, with multiple register sets, you must not allow the RILs assigned to one shadow register set to overlap the RILs assigned to another register set.

Table 2–7 and Table 2–8 illustrate the validity of register set assignments when preemption within a register set is enabled.

Table 2–7. Example of Illegal RIL Assignment

RIL	Register Set 1	Register Set 2
1	IRQ0	
2	IRQ1	
3		IRQ2
4	IRQ3	
5		IRQ4
6		IRQ5
7		IRQ6

Table 2–8. Example of Legal RIL Assignment

RIL	Register Set 1	Register Set 2
1	IRQ0	
2	IRQ1	
3	IRQ3	
4		IRQ2
5		IRQ4
6		IRQ5
7		IRQ6

Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

Missing Documentation of Interrupt API Properties

"Tcl Commands" in the *Nios II Software Build Tools Reference* chapter of the *Nios II Software Developer's Handbook* does not include information about two `set_sw_property` command properties related to the application programming interfaces (APIs) for interrupts.

Workaround

The missing information belongs with the `set_sw_property` command, in "Tcl Commands" in the *Nios II Software Build Tools Reference* chapter of the *Nios II Software Developer's Handbook*.

The missing information follows:

- `supported_interrupt_apis`—Specifies the interrupt API that the device driver supports. The Nios II SBT analyzes this property for each driver in the system to determine the appropriate API to be used in the system.

  Specify `legacy_interrupt_api` if the device driver supports the legacy API only or `enhanced_interrupt_api` if the device driver supports the enhanced API only. Specify both using a quoted list if the device driver supports both APIs.

  If you do not specify which API your device driver supports, the Nios II SBT assumes that only the legacy interrupt API is supported.

  ☞ This property is only available for device drivers.

  For more information about the legacy and enhanced APIs, refer to "Exception Handling" in the *Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook*.

- `isr_preemption_supported`—Specify `true` if your device driver's interrupt service routine (ISR) can be preempted by a higher priority ISR. If you do not specify whether the driver supports preemption, the Nios II SBT assumes that your device driver does not support preemption. If your driver does not have an ISR, but the associated device has an interrupt port, you can set this property to `true`.

  ☞ This property is valid for operating systems and device drivers.

Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

Valid Range of hal.log_flags is –1 to 3

The Nios II EDS documentation incorrectly states the valid range for the `hal.log_flags` BSP setting. Valid values of `hal.log_flags` range from –1 through 3.

Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

NicheStack TCP/IP Stack - Nios II Edition Does Not Support TFTP

The Nios II documentation should include the following information:

Versions of the NicheStack TCP/IP Stack other than the Nios II Edition include optional Trivial File Transfer Protocol (TFTP) client and server applications. However, the TFTP client and server are not thread safe. They cannot be used in systems with the MicroC/OS-II RTOS.

TFTP is not available with the NicheStack TCP/IP Stack - Nios II Edition. The TFTP client and server features are disabled.

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Error Message After Renaming Project: "Resource is out of sync with the system"

The *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook* does not explain the correct procedure for renaming a project.

Workaround

The missing information is as follows:

To rename a project in the Nios II SBT for Eclipse, execute the following steps:

1. Right click the BSP project and click **Rename**.
2. Type the new BSP name.
3. Right click the BSP project and click **Refresh**.

☞ If you neglect to refresh the project, you might see the following error message when you attempt to build it:

```
Resource <original_bsp_name> is out of sync with the system
```

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Obsolete Wiki URLs in Help Systems and Handbook

The following documents contain obsolete URLs for the Nios Community Wiki:

- The *Nios II Software Developer's Handbook*
- The *Nios II Embedded Design Handbook*
- The Nios II IDE help system

Workaround

You can visit the Nios Community Wiki at www.nioswiki.com.

Solution Status

Fixed in the documentation for v. 9.0

Missing Source Control Information

The Quartus II Handbook refers to the *Using the Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook* for information about putting Nios II projects in source control. The referenced chapter only describes the process for Nios II SBT projects.

Workaround

The missing information follows.

Archiving Nios II IDE Software Projects

This section helps you identify the files you must include when archiving a Nios II IDE software project. With this information, you can archive a Nios II application project and its associated Nios II system library project.

You might want to archive your projects for one of the following reasons:

- To place them under source control
- To create backups
- To bundle the projects for transfer to another location

This section covers the following information:

- How to find and identify the files that you must include in an archived Nios II IDE software project.
- Which files must have write permission to allow the software projects to be built.

Table 6 and Table 7 list the files required by Nios II IDE software projects. This is the minimum set of files needed to completely rebuild a software project, including the **.elf**.

Archive your Nios II IDE software projects together with the SOPC Builder system on which they are based. You cannot rebuild a Nios II IDE software project without its associated SOPC Builder system.

The files listed in Table 6 are located in the Nios II application project directory.

Table 6. Files Required for a Nios II Application Project

File Description	File Name	Write Permission Required?
All source files	for example: **app.c**, **header.h**, **assembly.s**, **lookuptable.dat**	No
Eclipse project file	**.project**	No
C/C++ Development Toolkit project file	**.cdtproject**	Yes
C/C++ Development Toolkit option file	**.cdtbuild**	No
Software configuration file	**application.stf**	No

The files listed in Table 7 are located in the Nios II system library project directory.

Table 7. Files Required for a Nios II System Library Project

File Description	File Name	Write Permission Required?
Eclipse project file	**.project**	Yes
C/C++ Development Toolkit project file	**.cdtproject**	Yes
C/C++ Development Toolkit option file	**.cdtbuild**	No
System software configuration file	**system.stf**	Yes

☞ You must have write permission for certain files. The tools write to these files as part of the build process. If the files are not writable, the toolchain fails. Many source control tools mark local files read-only by default. In this case, you must override this behavior. You do not have to check the files out of source control unless you are modifying the Nios II software project.

Solution Status

In the documentation for v. 9.0, the missing information appears in the *Nios II Integrated Development Environment* chapter of the *Nios II Software Developer's Handbook*. In the documentation for v. 9.1, the missing information appears in the *Using the Nios II Integrated Development Environment* appendix of the *Nios II Software Developer's Handbook*.

Compiler Flags for Building Custom Newlib

"Common BSP Tasks" in the *Using the Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook* describes how to compile a custom version of the newlib library. However, it does not list the compiler flags that must be selected to generate the correct library.

Workaround

The missing information is as follows:

The Nios II EDS provides a number of precompiled newlib implementations. The provided libraries are precompiled with every viable combination of the GCC compiler flags shown in Table 8.

When you create a BSP with a precompiled newlib, the Nios II SBT selects the newlib matching your BSP's compiler settings. When you create a custom newlib, you must ensure that the compiler flags listed in Table 8 match your BSP's settings.

Table 8. GCC Compiler Options for Newlib

Option Name	Purpose
`-pg`	Link for profiling with gprof
`-EB`	Generate big-endian code
`-mstack-check`	Enable stack checking
`-mno-hw-mul`	Disable use of the `mul` family of instructions
`-mhw-mulx`	Enable use of the `mulx` family of instructions
`-mcustom-fpu-cfg`	Use a floating point custom instruction

☞ The Nios II-specific compiler flag `-mcustom-fpu-cfg` requires the Nios II processor core to be implemented with the floating-point custom instruction as specified in the flag argument. The compiler flag can have one of the following two argument values:

- `-mcustom-fpu-cfg=60-1`—Use the Nios II floating-point custom instruction without divider
- `-mcustom-fpu-cfg=60-2`—Use the Nios II floating-point custom instruction with divider

For details about the Nios II-specific compiler flags `-mstack-check`, `-mno-hw-mul`, and `-mhw-mulx`, refer to "Altera Nios II Options" in *Using the GNU Compiler Collection (GCC)*, installed with the Nios II EDS. For details about the `-pg` compiler flag, refer to "Compiling a Program for Profiling" in *GNU Profiler*, installed with the Nios II EDS. To find the installed documentation, on the Windows Start menu, click **Programs** > **Altera** > **Nios II EDS <*version*>** > **Nios II <*version*> Documentation**, and then click **Literature**.

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Error Message on Invalid Exception or Reset Vector

Nios II compiler error messages, added in v 8.1, are not documented in the *Nios II Software Developer's Handbook*.

Workaround

The missing information is as follows:

While building a Nios II software project, the compiler produces an error message if the SOPC Builder system defines an invalid exception or reset vector. Table 9 shows possible error messages and their causes.

In the SOPC Builder system, ensure that the exception and reset vectors conform to all the criteria listed in Table 9.

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Table 9. Exception and Reset Vector Error Messages (Part 1 of 2)

Message	Cause
`The section mapping ".entry" is not mapped to a memory region.`	The `.entry` section is mapped to a nonexistent memory region.
`The section mapping ".exceptions" is not mapped to a memory region.`	The `.exceptions` section is mapped to a nonexistent memory region.
`The section mapping ".entry" maps to a memory region "<`*`region name`*`>" which is not the required length "32".`	The `.entry` section is mapped to a memory region that is smaller than 32 bytes.
`The section mapping ".entry" maps to a memory region "<`*`region name`*`>" which does not start at the SOPC Design specified address "<`*`base address`*`>".`	The `.entry` section is mapped to a memory region that does not start at the reset vector base address.
`The section mapping ".exceptions" maps to a memory region "<`*`region name`*`>" which does not start at the SOPC Design specified address "<`*`base address`*`>".`	The `.exceptions` section is mapped to a memory region that does not start at the exception vector base address.
`The SOPC design Nios II reset and exception vector addresses are mapped to the same memory. For HAL the exception address must be at least 32 bytes larger than the reset address.`	The `.entry` section and `.exceptions` section map to the same memory device, and the base of the `.exceptions` section is less than 32 bytes above the base of the `.entry` section.

Table 9. Exception and Reset Vector Error Messages (Part 2 of 2)

Message	Cause
`The section mapping ".exceptions" mapped device "<`*`memory 1`*`>" is not the same as the SOPC Design specified device "<`*`memory 2`*`>".`	The `.exceptions` section is mapped to a device that is different from the exception vector device specified in the SOPC Builder system.
`The section mapping ".entry" mapped device "<`*`memory 1`*`>" is not the same as the SOPC Design specified device "<`*`memory 2`*`>".`	The `.entry` section is mapped to a device that is different from the reset vector device specified in the SOPC Builder system.

Nios II IDE Online Help Expand Buttons Do Not Work

The expand text arrow and the **Show All** option in the online help do not work in Internet Explorer.

Workaround

In Internet Explorer, carry out the following actions:

1. Click the **Refresh/Show Current Topic** icon (yellow arrows icon at the top of the search pane). The table of contents appears with the selected topic highlighted.
2. Click the highlighted topic to refresh the browser frame. Expanded text works.

Alternatively, use Firefox 2.0.0.6.

Solution Status

Not fixed

Hardware Abstraction Layer Errata

This section describes in detail the Nios II Hardware Abstraction Layer issues.

Missing Structure Member Errors in alt_log_printf.c with Small JTAG UART Driver

In a BSP, if you enable the small JTAG UART driver (`altera_avalon_jtag_uart_driver.enable_small_driver`), and enable Altera logging (`hal.log_port`) using the JTAG UART as the log port, when you build the project, the compiler reports missing structure members in **alt_log_printf.c**. The project fails to build.

Workaround

To avoid this issue, execute the following steps:

1. In function `alt_log_jtag_uart_print_control_reg()` in **alt_log_printf.c**, insert the italicized code as shown in Example 9.

Example 9. ALT_LOG_PRINTF() Correction

```
ALT_LOG_PRINTF(
    "%s SW CirBuf = %d, HW FIFO wspace=%d AC=%d WI=%d RI=%d WE=%d RE=%d\r\n",
    header,

/* Beginning of inserted code */
#ifndef ALTERA_AVALON_JTAG_UART_SMALL
    (dev->tx_out-dev->tx_in),
#else
    0,
#endif
/* End of inserted code */

    space,ac,wi,ri,we,re);
```

2. Rebuild the project.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

“unused variable” Warning When Building alt_main.c

If you set the `hal.enable_exit` BSP setting to `false`, when building the project you see the following warning in **alt_main.c**:

```
warning: unused variable 'result'
```

Design Impact

This warning is harmless.

Workaround

Set `hal.enable_exit` to `true`.

Alternatively, you can ignore this harmless warning.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Build Warnings in Performance Counter Driver Files

If your Nios II system contains a performance counter, you might see the following warning when you build a BSP:

```
warning: return makes pointer from integer without a cast
```

This warning appears in `alt_get_performance_counter_base()` in the software file **altera_avalon_performance_counter.c**.

This warning appears when the your BSP uses the Altera performance counter driver, and the C preprocessor symbol `PERFORMANCE_COUNTER_0_BASE` is defined.

This warning is harmless.

Workaround

In the `alt_get_performance_counter_base()` function in **altera_avalon_performance_counter.c**, add typecast `(void *)` to `PERFORMANCE_COUNTER_0_BASE`.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Hardware Example Design Errata

This section describes in detail the Nios II EDS hardware example design issues.

Incorrect PLL Phase Shift in Stratix II and Cyclone II Designs

In several example designs targeting the Nios II Cyclone II 2C35 and Nios II Stratix® II 2S60 RoHS development boards, the synchronous static random access memory (SSRAM) and legacy DDR SDRAM clocks are implemented with an incorrect phase-locked loop (PLL) phase shift.

This issue affects the following example designs:

- 2s60 RoHS Triple Speed Ethernet SG-DMA
- 2s60 RoHS Standard
- 2s60 RoHS Full Featured
- 2c35 Triple Speed Ethernet SG-DMA
- 2c35 Standard
- 2c35 Full Featured

This issue also affects any custom 2c35 or 2s60 RoHS design using SSRAM or legacy DDR SDRAM.

The incorrect phase shift might cause timing violations when reading from or writing to SSRAM or legacy DDR SDRAM.

Workaround

To correct the PLL settings, perform the following steps:

1. Open the standard design in SOPC Builder.
2. Double-click the component instance named `pll` to launch the PLL MegaWizard™ interface.
3. Click **Launch Altera's ALTPLL MegaWizard** to launch the MegaWizard interface.
4. Select the **Output Clocks** page and then the clock page used to generate the SSRAM clock. Depending on the particular design, this might be **clk c1**, **clk c2**, or another clock.

5. Ensure that **Clock phase shift** is –3.38 ns.
6. Click the clock page used to generate the write clock for the legacy DDR SDRAM, if any. Depending on the particular design, this might be **clk c1**, **clk c2**, or another clock.
7. Ensure that **Clock phase shift** is 270 degrees.
8. Click **Finish** to save changes and exit the PLL MegaWizard interface.
9. Click **Finish** to save settings to the PLL instance in SOPC Builder.
10. Regenerate the system in SOPC Builder and recompile in the Quartus II software. Refer to the **readme.txt** file in the design example folder for information on compiling designs containing the DDR controller.

For further information refer to **ssram_interface_readme.html**, located in the **<*Quartus II installation directory*>/sopc_builder/documents** folder. This document discusses the SSRAM timing analysis methodology in detail. Additionally, *AN 411: Understanding PLL Timing for Stratix II Devices* discusses clock phase shift calculations and assignments for PLLs in Stratix II devices.

Solution Status

Not fixed

Software Example Errata

This section describes in detail the Nios II EDS software example issues.

Hardware Tutorial Software Example Hangs on Some Boards

The **count binary** application used in the tutorial hangs when run on some Altera development boards. The software runs through the binary count once and stops responding.

This issue affects boards that do not have a display to handle the LCD output. The affected boards include the following:

- Embedded Systems Development Kit, Cyclone III Edition (EP3C120)
- Stratix IV GX FPGA Development Kit (EP4SGX230)

Workaround

Modify line 18 of the **count_binary.h** file, as follows:

```
# define LCD_PRINTF(lcd, args...) /* Do Nothing */
```

Rebuild and run the software again.

Solution Status

Not fixed

Simple Socket Server Example Displays Incorrect IP Address

The simple socket server networking software example displays an incorrect IP address for the incoming connection.

Workaround

There is currently no workaround for this problem. Ignore the IP address of any incoming connection that is displayed on the console.

Solution Status

Not fixed

Networking Examples

If you are running a networking software example, you might be asked for a nine-digit number. You are directed to find this number on a sticker on your Nios development board, identified by the prefix **ASJ**. Not all Nios development boards have this sticker.

Workaround

If your Nios development board does not have a sticker with the letters **ASJ** followed by a nine-digit number, enter a unique nine-digit number when prompted. To avoid network address conflicts, ensure that this number is unique to each Nios board connected to your network.

Solution Status

Not fixed

Flash Programmer Errata

This section describes in detail the Nios II EDS issues related to the flash programmer.

Unable to Configure FPGA from Flash with Parallel Flash Loader

If you program an FPGA image to flash memory on a board that utilizes the Parallel Flash Loader, the FPGA subsequently fails to load from flash memory. This issue arises because the **sof2flash** utility generates a flash file with an incompatible Programmer Object File (**.pof**) bitstream format.

Affected Configurations

This issue impacts the following target hardware:

- The Stratix IV GX FPGA Development Kit
- The Arria® II GX FPGA Development Kit
- Any hardware using the Parallel Flash Loader

Workaround

1. Visit the MySupport website and request Nios II EDS patch 0.73.

   Alternatively, upgrade to the Nios II EDS v. 9.1 SP1 or later.

2. Invoke **sof2flash** with the following options:

   - `--pfl`
   - `--optionsbits=<`*`option bit address`*`>`

   With these options, **sof2flash** generates a flash file with a compatible **.pof** file bitstream format.

Solution Status

Fixed in v. 9.1 SP1 of the Nios II EDS

Boot Failure or elf2flash Error Due to Incorrect Flash File Size

The `sof2flash` utility fails to recognize certain devices. As a result, it generates Motorola S-record (**.flash**) files of the wrong size.

With erasable programmable configurable serial (EPCS) memory, the `elf2flash --after` command fails to recognize the **.flash** file size. The resulting error prevents you from programming flash memory.

With Common Flash Interface (CFI) memory, you can program the **.flash** file, but the Nios II system fails to boot.

This error occurs when programming flash memory for the following devices:

- The following Stratix III devices:
  - EP3SL50
  - EP3SL50F484
  - EP3SL50F780
  - EP3SL110
  - EP3SL110F1152
  - EP3SL110F780
  - EP3SE80
  - EP3SE80F1152
  - EP3SE80F780
- All Stratix IV devices
- All Arria II GX devices

Workaround

Visit the MySupport website and request Nios II EDS 9.0 SP2 patch 2.01 and Quartus II 9.0 SP2 patch 2.63.

Alternatively, download and install the Altera Complete Design Suite v. 9.1 or later.

Solution Status

Fixed in v. 9.1 of the Nios II EDS

Delay When Creating New Flash Programmer Configuration

You might experience a delay of several seconds when creating a new Flash Programmer configuration in the Nios II IDE.

Workaround

There is no workaround.

Solution Status

Not fixed

elf2flash File Size Limit

The **elf2flash** utility supports **.elf** files up to approximately 24 MBytes in size. The **elf2flash** utility might fail on files larger than 24 MBytes, with the error message `java.lang.OutOfMemoryError`.

Workaround

Lower the number of symbols in your **.elf** file by turning off debug symbols.

Alternatively, specify less initialized data in the application.

Solution Status

Not fixed

Error Message “No such file or directory” When Programming Flash

When programming flash from a project stored in a path containing spaces, you might receive the spurious error message **No such file or directory**. The flash programmer does not correctly handle spaces in the directory path. However, this error is harmless, because flash programming completes successfully.

Workaround

None required.

Solution Status

Not fixed

Hardware Simulation Errata

This section describes in detail the Nios II EDS issues related to hardware simulation.

Vectored Interrupt Controller Does Not Support VHDL Simulation Models

SOPC Builder reports errors when you try to generate VHDL simulation files for the Vectored Interrupt Controller.

Workaround

There is no workaround.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Error “UNC paths are not supported” Launching ModelSim

If you launch ModelSim® from a working directory that is mapped via a universal naming convention (UNC) path (a path that starts with `//` instead of drive letter), you receive the following error message in SOPC Builder: **UNC paths are not supported. Defaulting to Windows directory**. This error occurs because ModelSim is calling a command shell, which does not support UNC paths.

Workaround

Map the UNC path to a drive letter and use the drive letter to reference the working directory in the launching shell.

Solution Status

Not fixed

Uninitialized .bss Variables in Simulation

If your program reads the value of an uninitialized `.bss` variable during HDL simulation, and the BSP (system library) is compiled with the **ModelSim only, no hardware support** property enabled in Nios II IDE, a warning appears about unfiltered data being `'x'`. This warning appears because when this property is enabled, the code that clears the `.bss` memory region is omitted to speed up HDL simulation so this memory region is uninitialized. The `.bss` region contains global and static local variables that are not initialized by the application so they default to a value of zero. When the Nios II processor reads uninitialized variables, it displays a warning and converts any of the bits of the uninitialized data to zero which correctly mimics the effect of the missing `.bss` clearing code. The Hardware Abstraction Layer (HAL) code that executes before and after `main()` might use `.bss` variables, so these warnings might appear even if your application does not use the `.bss` section.

Solution Status

Not fixed

Host Platform Errata

This section describes in detail the Nios II EDS issues related to the host development platform.

Nios II EDS Requires Administrative Permissions on Windows Vista

On the Windows Vista operating system, the Nios II EDS requires your user account to have administrative permissions.

Attempts to write to a directory other than the user home directory might fail, especially writing to an installation directory. For example, you might be unable to create a software project under an example design directory. Error messages might vary.

Workaround

Obtain an administrator account on your workstation.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Cannot Launch Nios II SBT for Eclipse on Red Hat Linux

You might be unable to launch Nios II SBT for Eclipse on the Red Hat Linux operating system. This is an issue with the version of XULRunner in Eclipse 3.4. The issue is resolved in XULRunner 1.9.1, which is available with Eclipse 3.5.

Workaround

Upgrade XULRunner to a newer version. To determine what version of XULRunner you have, type the following command at the command prompt:

```
xulrunner -v↵
```

To upgrade XULRunner using the **yum** software package manager, type the following command:

```
sudo yum update xulrunner↵
```

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Windows/Cygwin: Nios II Processor Generation Failure

You might see the following error message when you generate your Nios II system in SOPC Builder:

```
Error: Generator program for module 'cpu_fpoint' did NOT run
successfully
```

This error might occur on a Windows-based system when all of the following conditions are true:

- You have installed a version of Cygwin other than the one distributed with the Quartus II software.
- You launch the Quartus II software and SOPC Builder from a Nios II Command Shell.
- You enable the floating point custom instruction (FPCI) in the Nios II processor core.

Workaround

Launch the Quartus II software and SOPC Builder from the Windows Start menu, and regenerate your system.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Windows Vista: Limited Support in Nios II EDS

The Quartus II software introduces Windows Vista (32-bit and 64-bit) support in version 7.2. However, the Nios II Embedded Design Suite supports only the following platforms:

- Windows 2000
- Windows XP
- Windows XP (64)
- Windows Vista Business Edition (32-bit)
- SUSE 9 (32-bit)
- SUSE 9 (64-bit)
- Red Hat Linux v3.0 (32-bit)
- Red Hat Linux v3.0 (64-bit)
- Red Hat Linux v4.0 (32-bit)
- Red Hat Linux v4.0 (64-bit)

Workaround

There is no workaround.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Nios II IDE Errata: Building Projects

This section describes in detail the issues related to building projects in the Nios II IDE.

Nios II IDE Command-Line Tools Select Wrong Workspace on Linux

On Linux systems, the Nios II IDE command-line tools use the default Eclipse project workspace, regardless what workspace you are using in the IDE. The default workspace is at `$SOPC_KIT_NIOS2/eclipse/nios2-ide-workspace-<`*`version`*`>`.

This issue affects the following tools:

- `nios2-create-system-library`
- `nios2-create-application-project`
- `nios2-build-project`
- `nios2-import-project`
- `nios2-delete-project`

Workaround

Explicitly supply the Nios II IDE command-line tools with a workspace location, by means of the `-data` command-line argument. The syntax of the `-data` argument is as follows:

```
-data <path to workspace>
```

The path to the workspace must be absolute, and must not contain whitespace. Aside from these restrictions, any valid file system path can be used for the workspace. Example 10 shows how to specify a workspace path.

Example 10. Providing a Non-Default Workspace Location in Linux

```
nios2-create-project -data $HOME/myworkspace <other arguments>↵
```

Solution Status

Not fixed

Nios II IDE Command-Line Tools Hang on Windows

On Windows systems, the Nios II IDE command-line tools are sometimes unable to locate the project workspace. When this happens, the tools might hang.

This issue potentially affects the following tools:

- `nios2-create-system-library`
- `nios2-create-application-project`
- `nios2-build-project`
- `nios2-import-project`
- `nios2-delete-project`

Workaround

Explicitly supply the Nios II IDE command-line tools with a workspace location, by means of the `-data` command-line argument. The syntax of the `-data` argument is as follows:

```
-data <path to workspace>
```

The path to the workspace must be absolute, and must not contain whitespace. Aside from these restrictions, any valid file system path can be used for the workspace. The workspace shown in Example 11 is the default workspace used by the Nios II IDE in version 9.0.

Example 11. Providing a Workspace Location

```
nios2-create-project -data c:/altera/90/nios2eds/\
    bin/eclipse/nios2-ide-workspace-9.0 <other arguments>
```

Solution Status

Not fixed

Nios II IDE Cannot Find stdio.h in Outline View

If you create a new project using the Nios II IDE and try to open the **stdio.h** file from the Outline view before building the project, the IDE displays the error message **No include files were found that matched that name**.

Workaround

Build the project before attempting to open **stdio.h**.

Solution Status

Not fixed

Nios II IDE Hangs With UNC Project Path

In the **New Project** dialog box, if you turn on **Specify Location** and specify the path in UNC form, the IDE might hang.

Workaround

Map the UNC path to a remote drive which looks like a Windows drive to the Nios II IDE.

Solution Status

Not fixed

Build Failure with Nios II Advanced Exceptions, MMU, and MPU

Projects created in the Nios II IDE for an SOPC Builder system containing a Nios II processor configured with advanced exceptions, the memory management unit (MMU), or the memory protection unit (MPU) fail to build, and generate the following error messages:

```
ERROR - Classic build flow for Nios II system library does not support
the Nios II \
    advanced exceptions.
Use the non-classic Nios II Board Support Package instead.
<timestamp> - (SEVERE) generate: java.lang.IllegalStateException: \
    java.lang.IllegalStateException: \
    com.altera.ingenuous.GTFElement.GTFElementErrorException: \
    <error> element in GTF script
make[1]: *** [system_description/../obj/system.h-t] Error 1
make: *** [system_project] Error 2
```

Workaround

Create the software project with the Nios II SBT command-line development flow. Alternatively, use the Nios II SBT for Eclipse.

Solution Status

Not fixed

Build Command Not Functional for BSPs Created With the Nios II SBT

The build option in the Nios II IDE menu does not rebuild BSPs imported to the IDE.

Workaround

The Nios II SBT for Eclipse can both build and debug projects created on the command line. The Nios II SBT for Eclipse is the preferred tool for debugging Nios II SBT projects.

For information about the Nios II SBT for Eclipse, refer to the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*.

In the Nios II IDE, you can build the BSP by building the associated application project.

Solution Status

Not fixed

Linker Errors with Dual-Port Memories

If your instruction master and data master ports are connected to the same dual-port memory and the ports have different addresses, your code fails to run or you experience a linker error. The Nios II IDE does not warn you of the addressing violation.

Workaround

Assign the same address to both ports of the dual-port memory.

Solution Status

Not fixed

User-Managed BSP Settings Not Supported in Nios II IDE

For projects created with the Nios II SBT and imported to the Nios II IDE, the IDE configuration settings have no effect.

For example, objdump, compiler, and linker settings made in the IDE are ignored. This behavior occurs because Nios II SBT projects are not IDE-managed projects. In addition, the make-related preferences do not pertain to imported Nios II SBT projects. The IDE ignores these options during the build process.

Workaround

Make these settings in the project's makefile.

Solution Status

Not fixed

URL Project Location Causes Project Creation Error

When you try to create a new project in an existing workspace, you might see an error dialog box saying:

```
Project cannot be created. Reason: Internal Error
```

This error might occur if the path to any project in the workspace is a URL location, for example **file:/F:/Design**. To view the path, right-click on the project and select **Properties**.

Workaround

Import your existing application and system library projects to a new workspace.

Solution Status

Not fixed

Compilation Error with Separate Exception Stack Option

Choosing the **Use a separate exception stack option** might cause the following compilation error when building a project:

```
VARIABLE %STACK_POINTER%"
```

This error occurs if the exception stack is larger than the memory available for it.

Workaround

On the system library properties page for the project, turn off the separate exception stack or reduce the **Maximum exception stack size** setting.

Solution Status

Not fixed

Incorrect Stack and Heap Free Space Report

The makefile reports an incorrect number of bytes free for the stack and heap, if the heap and stack are in different memory regions.

Workaround

No workaround available.

Solution Status

Not fixed

Nios II IDE Reports Problems Without Displaying Error in Console

When building a project, the Nios II IDE reports problems, but the build output in the console does not contain any errors.

The Nios II IDE incorrectly reports some linker warnings as errors, with a dialog box saying **Errors exist in a required project**. The Dhrystone software example exhibits this behavior, and recompiling the project again makes the issue go away.

Workaround

If the Console output does not contain errors, then the project built correctly. On subsequent builds, the linker step is skipped and the errors do not appear.

Solution Status

Not fixed

Nios II IDE Errata: Debugging Projects

This section describes in detail the issues related to debugging projects in the Nios II IDE.

Method for Accessing MMU and MPU Registers in the Debugger

You might observe an error if you try to read or write an MMU or MPU register via the Nios II Debugger.

Workaround

To read an MPU region, execute the following steps:

1. Set region `INDEX` in the `MPUBASE` register.
2. Exit and reenter debug mode, that is, single step.
3. Set the `RD` bit in the `MPUACC` register.
4. Exit and reenter debug mode, that is, single step.
5. Read back `MPUBASE` for pertinent information.
6. Read back `MPUACC` for pertinent information.

For a system with an MMU or MPU, this workaround allows you to read and write the current values of the registers. However, you cannot use it to control MPU regions or MMU TLB entries.

The debug core copies the Nios II processor's register values to its internal memory when the processor enters debug mode. The debug core writes register values back to the processor only when the processor leaves debug mode. Therefore, if you attempt to set an MPU region with several consecutive values, only the last one, when leaving debug mode, is committed to the processor.

Solution Status

Not fixed

IDE Cannot Display Imported Profiling Data

If you create a software project in the Nios II SBT command line flow, generate profiling data to **gmon.out** in the Nios II command shell, and then import the project to the IDE, you cannot use the Profiling perspective to view the profiling data. When you attempt to view **gmon.out**, the IDE displays the following error message:

```
nios2-elf-gprof: ../: not in a.out format.
```

Workaround

Run `nios2-elf-gprof` from the Nios II command shell to generate a profiler report.

Solution Status

Not fixed

Error Message when Downloading .elf File

If the **Build Automatically** option is turned on in the IDE, you might get the following error when downloading a **.elf**:

```
"2 [main] sh 5736 fork: child 3892 - died waiting; \
    for longjmp before initialization, errno 11"
```

The **Build Automatically** option is off by default, as recommended by the Eclipse help system.

Workaround

Turn off the **Build Automatically** option and download the **.elf** file again.

Solution Status

Not fixed

Variable Casting Unsupported in ISS

Casting variables in the Variables view in the Debug perspective when using the Nios II ISS might cause an exception dialog box to open.

This exception might occur if you select **Debug as ISS** in the Nios II IDE and try to cast variables via the Variables view.

Workaround

There is no workaround.

Solution Status

Not fixed

Trace Debug Does Not Support Instruction-Related Exceptions

The instruction-related exception handler is not supported by the Nios II trace tools.

Workaround

There is no workaround.

Solution Status

Not fixed

Trace Debug Does Not Support the JMPI Instruction

The JMPI instruction is not supported by the Nios II trace tools.

Workaround

There is no workaround.

Solution Status

Not fixed

Cannot Locate Source Code in Driver Files Shared by Multiple Projects

If you hit a breakpoint in a driver file, and that driver file is shared with another project that is closed, the Nios II IDE might indicate that it cannot locate the source code.

Workaround

Open the closed system library project and resume debugging.

Solution Status

Not fixed

Console Window Is Not Updated After ISS Error

After performing a **Run as ISS**, if you receive an ISS error in the console window, the console is not updated subsequently.

Workaround

Close the console window after receiving an ISS error. A new console window opens when a new message is available.

Solution Status

Not fixed

ISS Fails on Designs Containing Triple Speed Ethernet MAC or SG-DMA Components

You receive an Internal Error when attempting to perform an ISS simulation of designs containing the Altera Triple Speed Ethernet media access control (MAC) or SG-DMA components because the Nios II ISS does not support these components.

Workaround

Remove the Triple Speed Ethernet MAC and SG-DMA components from your system and perform ISS simulation on the simplified system. You can also simulate the design in ModelSim or test it on hardware.

☞ Before removing the Triple Speed Ethernet MAC and SG-DMA components, make a copy of the unmodified system to ensure that you can return to the original configuration.

Solution Status

Not fixed

Memory Window Sets Control Register Values Incorrectly

The memory window might incorrectly set values in memory-mapped control registers. For example, writing 0x1234 to a byte addressed register results in the value 0x3434 being stored in the register. The memory window shows this incorrect value.

Workaround

Use the GNU debugger (GDB) console window in the IDE, instead of the memory window, to write to the registers. For example, type the following command:

```
set {int} <register address>=0x1234↵
```

☞ You must refresh the memory window for it to correctly display the target value.

Solution Status

Not fixed

Programs That Interact With a Terminal Console on Windows Do Not Work

Programs with this behavior work in v. 6.0 and earlier, but do not work in Nios II IDE v. 6.1 and later.

The Eclipse platform in v. 6.1 and later of the IDE (on Windows only) sends the string `\r\n` instead of just `\n` when running or debugging using the Terminal. This behavior can break existing software designs, and it is inconsistent with `nios2-terminal`, which still just sends `\n`.

Workaround

Change the software running on the Nios II processor to parse for `\r\n` as well as `\n`.

Solution Status

Not fixed

“Run as ModelSim” in the Nios II IDE Fails

The **Run as ModelSim** command might fail on launch configurations created in v. 7.0 or earlier of the IDE. This problem does not occur for new launch configurations.

Workaround

Select a location for the ModelSim tool from the launch configuration dialog box. You can use the **Browse** button next to the **ModelSim path** group, or type in a path to the ModelSim directory (for example **c:/altera/71/modelsim_ae/win32aloem**).

Solution Status

Not fixed

The Restart Command on the Run Menu Does Not Work

The **Restart** command on the **Run** menu does not work.

Workaround

Stop the program, then debug it again. If the debugger is hung in an endless loop, use the following **bash** alias to break the target, then stop it:

```
alias break="kill -2 \'ps -a | grep nios2-elf-gdb | cut -f6 -d' '\'"
```

Solution Status

Not fixed

Watchpoints Do Not Work on Certain Variables

Watchpoints do not work on a variable whose size is not 32 bits.

Workaround

Change the types of global and static local variables to `int`, `long`, or `unsigned long` before setting watchpoints on them.

Solution Status

Not fixed

Nios II IDE Errata: Navigating Projects

This section describes in detail the issues related to navigating projects in the Nios II IDE.

Nios II IDE Freezes While Displaying the Splash Screen

After clicking **Switch Workspace** on the File menu on a Windows machine, a Nios II IDE splash screen appears. Unfortunately, this splash screen obscures the dialog box that asks you to specify the new workspace. As a result, the IDE appears to freeze.

Workaround

Press Alt-Tab to switch applications. Two relevant application icons appear: an Eclipse icon associated with the splash screen and a Nios II IDE icon associated with the **Workspace** dialog box. Select the Nios II icon to bring the dialog box to the foreground.

Solution Status

Not fixed

Internal Error When Double-Clicking on a Large Objdump File

On Windows when opening a large **objdump** file in the Nios II IDE, you might get the following error message:

```
Unable to create this part due to an internal error.
Reason for the failure: Editor could not be initialized.
```

Workaround

Adjust the Windows launch arguments for the Nios II IDE editor. Perform the following steps:

1. On the Windows Start menu, browse to the **Nios II EDS** program icon, right-click it, then click **Properties**. The **Windows Properties** dialog box appears.
2. In the **Target** field, append `"vmargs -Xmx1024m"` to the end of the path to the Nios II IDE executable. For example:

   ```
   C:\altera\72\nios2eds\bin\eclipse\nios2-ide.exe -vmargs -Xmx1024m
   ```

Solution Status

Not fixed

C/C++ Scanner Does Not Support Certain C/C++ Constructs

The C/C++ scanner performs C/C++ Search, navigation, open declaration, and parts of content assist. Due to limitations of the C/C++ scanner, these features do not work with C code constructs not supported by the C++ language. An example is functions that take a function pointer as an argument.

Workaround

If the C/C++ Search fails, use the File Search facility.

Solution Status

Not fixed

Nios II SBT Errata

This section describes in detail the issues in the Nios II SBT.

Error Running Nios II Project: ‘Downloading ELF Process failed’

If the Nios II processor’s cpu.data_master port is not connected to all program memories (memories to which the **.elf** file is downloaded) the software project fails to run on Nios II hardware.

Failure to connect cpu.data_master to all program memories is a design error that the Nios II SBT does not detect.

Workaround

Connect cpu.data_master to all program memories.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

BSP Editor Does Not Show Command-Line Help

The following commands in the Nios II Command Shell should display command-line help:

- `nios2-bsp-editor --help`
- `nios2-bsp-editor --extended-help`

Instead, these commands launch the BSP Editor.

Workaround

For help with the BSP Editor, refer to "Using the BSP Editor" in the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

BSP Not Updated for Memory Size Changes in SOPC Builder

If you change the size of a memory in SOPC Builder, the memory region size in any previously created BSP is no longer correct. Regenerating the BSP does not update the BSP's memory region size.

Workaround

In the BSP Editor, on the **Linker Script** tab, you can correct the memory region size either of the following ways:

- Edit the memory region size manually.
- Click **Restore Defaults** to rerun the default Tcl script. The default Tcl script reads the updated memory region size from the SOPC information file (**.sopcinfo**) and updates the BSP.

For detailed information about keeping your BSP consistent with changes in the underlying SOPC Builder system, refer to "Revising Your BSP" in the *Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook*.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Build Errors on Interrupt API Calls

You might observe build errors if your application project installs an ISR that calls the `alt_irq_register()` function.

The Nios II EDS v9.1 includes support for an optional External Interrupt Controller (EIC) interface on the Nios II processor. To support the EIC interface, the HAL includes an enhanced interrupt API.

When you create a BSP, the SBT determines which interrupt API(s) the device drivers in your system support. If all registered drivers in the BSP identify themselves as supporting the enhanced API, the SBT implements the enhanced API. All Altera device drivers in v9.1 and later support the enhanced API.

When the enhanced API is implemented, the legacy interrupt API, including `alt_irq_register()`, is not available. Therefore, if the application code contains a call to a legacy API function, a linker error occurs.

This issue might affect application projects that call `alt_irq_register()` or other legacy interrupt API functions directly, rather than depending on drivers in the BSP for interrupt support.

If your Nios II application project's source code registers an ISR using `alt_irq_register()`, or includes an ISR, your application might fail to compile in the Nios II EDS v. 9.1, due to undefined references to `alt_irq_register()`, or to a change to the ISR function prototype.

Workaround

Modify your application code to use the new enhanced interrupt API. To use the enhanced API, you must modify several function calls and all ISR function prototypes.

For information about supporting the enhanced interrupt API, refer to the *Exception Handling* chapter of the *Nios II Software Developer's Handbook*, or to *AN 595: Vectored Interrupt Controller Usage and Applications*.

You must upgrade to the enhanced API if you wish to use an EIC in your hardware design to accelerate interrupt response time. The enhanced API also works with the Nios II processor's internal interrupt controller.

Solution Status

Not fixed

BSP Editor Incorrectly Allows Mixed Drivers

If multiple drivers are found for a single peripheral type, and you have multiple instances of the peripheral type in your SOPC Builder system, the Nios II BSP editor allows you to manually select a different driver for each instance of the peripheral or processor. For example, if you have two `altera_avalon_timer` devices, you can select two different timer drivers.

This behavior is not fully supported. You must choose the same driver for both instances of the peripheral type.

You can make driver assignments in one of the following ways:

- In the Nios II BSP editor, within the **Drivers** tab. The **Driver name** associated with each peripheral can be edited, if multiple driver types are found that support the peripheral. You must make the same driver type selection for each instance of a peripheral type in the list of peripherals as defined in the **Module Name** column.
- With the Nios II SBT in command-line mode (`nios2-bsp`, `nios2-bsp-create-settings`, `nios2-bsp-update-settings`): the **set_driver** command can be invoked (either from a Tcl script or with the `--cmd` option). The **set_driver** command accepts a peripheral module instance name and driver name as arguments. If you are using the command line development flow and wish to use this command, you must invoke it once for each instance of a peripheral type in your system, similar to the BSP editor flow described in the previous paragraph.

Workaround

There is no workaround.

Solution Status

Fixed in v. 9.1 of the Nios II EDS

nios2-bsp --help Displays Incorrect List of BSP Types

The Nios II SBT supports custom HAL-based operating systems, when they are added as Tcl-scripted software packages. However, the `nios2-bsp` tool help message indicates that there are only two BSP types, HAL and UCOSII. If other HAL-based operating system components are added they do not appear in the `nios2-bsp` help message.

Workaround

To get a correct, dynamic list of available BSP types, type the following command:

```
nios2-bsp-create-settings --type x --settings x --sopc=<existing sopcinfo path>↵
```

This command returns an error message that lists available BSP types:

```
SEVERE: Available BSP type values for the --type argument are: [<list of BSP types>]
```

Solution Status

Fixed in v. 9.1 of the Nios II EDS

Java Heap Space Error When Creating BSP

When creating a BSP, the Nios II SBT might hang with the following message:

`SEVERE: OutOfMemoryError: Java Heap Space`.

This problem occurs when your SOPC Builder design has connection cycles. A connection cycle consists of a group of components that master one another in a circular arrangement. For example, if `ddr_sdram_bridge` masters `lcd_sgdma`, and `lcd_sgdma` masters `ddr_sdram_bridge`, the two components form a simple connection cycle. In this situation, the SBT is unable to create a BSP.

Workaround

Avoid connection cycles in the SOPC Builder system.

Solution Status

Fixed in v. 9.0 of the Nios II EDS

SBT Fails if Nios II EDS is Installed in a Path Containing Spaces

The Nios II Command Line SBT fails if you install the Nios II EDS in a path containing spaces.

Workaround

Reinstall the Nios II EDS to a path that does not contain spaces.

Solution Status

Not fixed

User-Managed BSP Settings Not Supported in Nios II IDE

This issue is described on page 48.

Upper-Case File Extensions Not Supported

The Nios II SBT for applications and libraries (`nios2-app-generate-makefile` and `nios2-lib-generate-makefile` commands) does not support source files with certain upper-case extensions. If a file with an upper-case extension is included, the make command stops with no descriptive warning.

Only Nios II assembly language files built by the C preprocessor can have upper-case file extension (**.S**). All C language files must have the extension **.c** or **.h**. C++ language source files must have the extension **.cpp**, **.cxx**, **.cc,** or **.h**.

Workaround

Rename all C language files with the extension **.c** or **.h**. Rename all C++ language files with the extension **.cpp**, **.cxx**, **.cc,** or **.h**.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Peripheral Errata

This section describes in detail the Nios II EDS issues related to peripheral components.

Vectored Interrupt Controller Does Not Support VHDL Simulation Models

This issue is described on page 42.

Unaligned Transfers of Small Payloads Fail on SG-DMA

The Scatter Gather DMA SOPC Builder peripheral does not correctly handle unaligned transfers with small payloads. A payload length smaller than the data width causes erroneous data transfers.

Workaround

Avoid using DMA devices to transfer small payloads.

If absolutely necessary, for a 32-bit SG-DMA, a minimum length of 4 bytes guarantees that data is transferred correctly.

Solution Status

Not fixed

DMA Controller Always Busy in Burst Mode

The DMA controller component (`altera_avalon_dma`), when enabled for burst transactions, does not perform transfers at widths other than its full data width. The DMA controller is always busy.

Workaround

When bursting is enabled, the DMA controller must be programmed to perform transactions at its full data width.

Solution Status

Not fixed

Non-System-Wide Reset Can Cause Improper Initialization of Mailbox Core

The `altera_avalon_mailbox` peripheral might not be initialized properly when a soft (non-system-wide) reset occurs. In this condition, mailbox contents (read and write pointers) are not reinitialized and might show potentially stale data.

Workaround

Ensure that a system-wide reset event occurs by asserting the `reset_n` input to the SOPC Builder system containing the mailbox. This resets all peripherals and Nios II processors in the system.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Target Software Errata

This section describes in detail the Nios II EDS issues related to target software packages.

NicheStack tcp_wake() Function Does Not Awaken All select() Sleepers

This issue impacts software based on the NicheStack TCP/IP Stack - Nios II Edition, with multiple sleeping tasks that depend on the same `select()` event to wake up. The `tcp_sleep()` and `tcp_wake()` functions do not work properly for tasks that call `select()`.

Tasks can normally use `tcp_sleep()` and `tcp_wake()` to suspend and then awaken themselves upon specific events. The `select()` function passes an event pointer to `tcp_sleep()` that is identical for every task that might call `select()`. All other events that call `tcp_sleep()` pass it a unique event pointer that specifically identifies them as the task to be awakened at the next event.

The `tcp_wake()` function apparently assumes that the list of sleepers contains unique event pointers by which they can be identified. Therefore, when a `select()` event occurs, instead of awakening all tasks waiting for it, `tcp_wake()` awakens only the first task in the list. Therefore, when more than one task calls `select()` to await some outcome on a socket, only the highest-priority task in the sleepers list is awakened at `select()` events. Other sleeping tasks that depend on the same `select()` event do not wake up.

Workaround

There is no workaround.

Solution Status

This issue will be fixed in future release of the Nios II EDS.

EIC-Based Nios II System Crashes With NicheStack

When you attempt to run software based on the NicheStack TCP/IP Stack - Nios II Edition on a hardware design incorporating an external interrupt controller such as the VIC, the Nios II system crashes during initialization, with unpredictable results.

Workaround

There is no workaround.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Gigabit Ethernet Performance Issues

The NicheStack TCP/IP Stack - Nios II Edition might hang while sending transmission control protocol (TCP) packets to the host, or experience dropped packets when receiving TCP packets from the host. These issues can arise with version 3.1 of the NicheStack TCP/IP Stack, using the triple-speed Ethernet component to transmit over a Gigabit Ethernet point-to-point connection.

Workaround

Download and install the patches at www.altera.com/support/kdb/solutions/rd01132009_588.html.

Alternatively, upgrade to the Nios II EDS v. 9.0 or later.

Solution Status

Fixed in v. 9.0 of the Nios II EDS

stdio Does Not Work with MicroC/OS-II and Small C Library

`stdin`, `stdout`, and `stderr` do not work in MicroC/OS-II applications built with the Small C library option.

Workaround

Disable the small C library option.

Solution Status

Not fixed

cout From MicroC/OS-II Task Does Not Send Data to stdout

If neither `printf()` or `cout` is used from `main()` before tasks are started, `cout` does not work from a task.

Workaround

Add the following C++ code to the beginning of `main()`:

```
std::ios_base::sync_with_stdio(false);
```

Solution Status

Not fixed

malloc(), realloc() Failures With MicroC/OS-II

When you use the MicroC/OS-II RTOS, calls to `malloc()` and `realloc()` might fail if successive calls to `malloc()` or `realloc()` within a MicroC/OS-II task occur after changing the task priority of the task in which a memory block is originally allocated.

Workaround

Use one of the following workarounds:

- Allocate and/or reallocate memory blocks outside of MicroC/OS-II tasks, before task switching starts. Preallocated memory blocks make it possible to change thread priorities at runtime.
- Allocate fixed areas of memory using arrays (rather than using `malloc()`) before task switching starts. Fixed memory arrays make it possible to change thread priorities at runtime.
- Allocate memory using `malloc()` or `realloc()` from a MicroC/OS-II task. You can change task priorities at runtime, but only for tasks that have not used `malloc()` or `realloc()`.

Solution Status

Not fixed

Toolchain Errata

This section describes in detail the Nios II EDS issues related to the Nios II and GNU compiler toolchains, including tools such as **gcc**, **gdb**, and **sof2flash**.

nios2-console Does Not Work in the GCC 4 Command Shell

If you try to run the **nios2-console** command in the GCC 4 Command Shell, you see the following error message:

```
Can't locate strict.pm in @INC ...
```

Workaround

Use the GCC 3 Command Shell.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

C++ Exceptions Unsupported in a Multi-Threaded Environment

C++ exceptions are only supported in a single-threaded environment. They cannot be used in a multi-threaded environment such as MicroC/OS-II.

Also see "C++ Exceptions Unsupported by the Nios II GCC 4 Toolchain".

Workaround

In a multi-threaded environment, you need to guard C++ exceptions with semaphores.

Solution Status

Not fixed

C++ Exceptions Unsupported by the Nios II GCC 4 Toolchain

With the GCC 4 toolchain, C++ exceptions are not supported.

Also see “C++ Exceptions Unsupported in a Multi-Threaded Environment”.

Workaround

If your single-threaded project requires C++ exceptions, use the GCC 3 toolchain.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

memcpy() Optimization Misalignment

When optimization is turned on (`-O1` or higher), if you use `memcpy()` and the source pointer is aligned to a 32-bit boundary, the compiler implements `memcpy()` with word-oriented instructions as part of the optimization process. This optimization technique causes unexpected results in your software if `memcpy()` is used on a misaligned address.

Workaround

Take steps to ensure that an optimized implementation of `memcpy()` is called only with aligned data pointers.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Unable to Configure FPGA from Flash with Parallel Flash Loader

This issue is described on page 40.

GNU Assembler Does Not Accept the --defsym Flag

According the GNU documentation, you can set an assembler definition by using the `--defsym` flag, but it does not work in the following form: `--defsym MY_VAR=1`.

Workaround

There is no workaround.

Solution Status

Not fixed

How to Contact Altera

For the most up-to-date information about Altera products, refer to Table 10.

Table 10. Contact Information

Contact (1)	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Altera literature services	Email	literature@altera.com
Non-technical support (General)	Email	nacomp@altera.com
(Software Licensing)	Email	authorization@altera.com

Note:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

Table 11 shows the typographic conventions that this document uses.

Table 11. Typographic Conventions (Part 1 of 2)

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Indicates command names, dialog box titles, dialog box options, and other graphical user interface (GUI) labels. For example, **Save As** dialog box. For GUI elements, capitalization matches the GUI.
bold type	Indicates directory names, project names, disk drive names, file names, file name extensions, dialog box options, software utility names, and other GUI labels. For example, **\qdesigns** directory, **d:** drive, and **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Indicates document titles. For example, *AN 519: Stratix IV Design Guidelines.*
Italic type	Indicates variables. For example, *n* + 1. Variable names are enclosed in angle brackets (< >). For example, <*file name*> and <*project name*>**.pof** file.
Initial Capital Letters	Indicates keyboard keys and menu names. For example, Delete key and the Options menu.
“Subheading Title”	Quotation marks indicate references to sections within a document and titles of Quartus II Help topics. For example, “Typographic Conventions.”
`Courier type`	Indicates signal, port, register, bit, block, and primitive names. For example, `data1`, `tdi`, and `input`. Active-low signals are denoted by suffix `n`. For example, `resetn`. Indicates command line commands and anything that must be typed exactly as it appears. For example, `c:\qdesigns\tutorial\chiptrip.gdf`. Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword `SUBDESIGN`), and logic function names (for example, `TRI`).
1., 2., 3., and a., b., c., and so on.	Numbered steps indicate a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ■	Bullets indicate a list of items when the sequence of the items is not important.

Table 11. Typographic Conventions (Part 2 of 2)

Visual Cue	Meaning
☞	The hand points to information that requires special attention.
⚠ CAUTION	A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.
⚠ WARNING	A warning calls attention to a condition or possible situation that can cause you injury.
↵	The angled arrow instructs you to press Enter.
👣	The feet direct you to more information about a particular topic.

101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support
www.altera.com/support

Nios II Embedded Design Suite Release Notes and Errata

© July 2010 RN-EDS-7.0

About These Release Notes

These release notes cover versions 9.0 through 10.0 of the Altera® Nios® II Embedded Design Suite (EDS). These release notes describe the revision history and errata for the Nios II EDS.

Errata are functional defects or errors, which might cause the product to deviate from published specifications. Documentation issues include errors, unclear descriptions, or omissions from current published specifications or product documents.

For the most up-to-date errata for this release, refer to the latest version of the *Nios II Embedded Design Suite Release Notes and Errata* on the Altera website. You can refer to release notes and errata for the Nios II processor core in the *Nios II Processor* chapter of the *MegaCore IP Library Release Notes and Errata*.

Revision History

Table 1 shows the revision history for the Nios II EDS, including the Nios II C2H Compiler.

Table 1. Nios II Embedded Design Suite Revision History

Version	Date	Description
10.0	July 2010	■ Software Build Tools (SBT) for Eclipse ■ Updated Eclipse to version 3.5 and Eclipse C/C++ development toolkit (CDT) to version 6.0 ■ Improved Source File management ■ Supports Multi-CPU launch ■ Updated to Nios II GCC 4.1.2 tool chain ■ Supports both SBT for Eclipse and Nios II Command Shell ■ GCC 4 improves compilation time compared to GCC 3 on Windows ■ Host platform ■ Supports Windows 7 ■ Supports SUSE Linux Enterprise 11 ■ Support for SUSE Linux Enterprise 9 discontinued ■ New Nios II software design examples for Reduced Gigabit Media Independent Interface (RGMII) ■ Web Server (RGMII) ■ Simple Socket Server (RGMII) ■ Legacy hardware design examples (Stratix® II and Cyclone® II) are removed ■ Installer changes ■ Nios II EDS must be installed next to Quartus® II
9.1 SP2	March 2010	Maintenance release
9.1 SP1	January 2010	Maintenance release
9.1	November 2009	■ The Nios II SBT for Eclipse, an integrated development environment based on Eclipse. Recommended for all new Nios II projects. ■ The Nios II Integrated Development Environment (IDE) is available in the Nios II EDS, but not recommended for new projects. ■ The Nios II Studio is discontinued. ■ The vectored interrupt controller (VIC) core offers high-performance, low-latency interrupt handling. ■ Nios II Design Examples with memory management unit (MMU): ■ Nios II 3C120 Hardware Reference Design with MMU, Cyclone III Edition ■ Nios II 4S230 Hardware Reference Design with MMU, Stratix IV GX Edition Commercial and open-source support for embedded Linux is available for these systems. ■ The Embedded Systems Lab for DE1 for Quartus and Nios II EDS v9.1 is available for download from the Nios Community Wiki (www.nioswiki.com). ■ Tcl scripting for software packages and drivers enhanced to support callback functions.
9.0	March 2009	Added the `nios2-convert-ide2sbt` tool, which converts projects from the Nios II IDE development flow to the Nios II SBT development flow
8.1 SP1	January 2009	Nios Community Wiki URL changing (www.nioswiki.com)
8.1	November 2008	Maintenance release

For more information on new features and changes, refer to the *Nios II Processor Reference Handbook*, the *Nios II Software Developer's Handbook*, and the *Nios II C2H Compiler User Guide*. For information about changes to the Nios II processor core, refer to the *Nios II Processor* chapter of the *MegaCore IP Library Release Notes and Errata*.

Notice about Environment Variables

The `QUARTUS_ROOTDIR` and `SOPC_KIT_NIOS2` environment variables will be deprecated in a future release. If you have custom scripts that use these variables, you should be prepared to change them to be independent of these variables, or to set the environment variables yourself.

Revision Details

This section provides details about some of the Nios II EDS revisions shown in Table 1.

Nios II Software Build Tools for Eclipse

The Nios II Software Build Tools (SBT) for Eclipse is a set of plugins based on the popular Eclipse framework and the Eclipse C/C++ development toolkit (CDT) plugins. The Nios II SBT for Eclipse provides a consistent development platform that works for all Nios II processor systems. You can accomplish most Nios II software development tasks within Eclipse, including creating, editing, building, running, debugging, and profiling programs. The Nios II SBT for Eclipse is built from the ground up as a set of plugins to standard Eclipse. The Nios II SBT for Eclipse is based on the Nios II Software Build Tools, a set of powerful commands, utilities and scripts for creating applications, board support packages and software libraries, and managing build options.

Table 2 compares the Nios II Integrated Development Environment (IDE) with the Nios II SBT for Eclipse.

Table 2. Differences Between Nios II SBT for Eclipse and Nios II IDE (Part 1 of 2)

Feature	Nios II IDE	Nios II SBT for Eclipse
Types of projects	■ Application ■ Board support package (BSP)—Called a system library in the IDE ■ User Library (optional)	■ Application ■ BSP ■ User Library (optional)
Handling updates to the SOPC Builder system	Updates the BSP automatically	You must initiate the BSP update (in the Nios II menu, click Generate BSP)
Options for creating projects	Application from template. BSP is created automatically as part of application.	■ New Application and BSP from template ■ Standalone application ■ Standalone BSP ■ Standalone library
Handling updates to BSP	Entire BSP is rebuilt. This process can be time-consuming.	Smart rebuild only recompiles changed files
Compatibility with command-line flow	None	Full compatibility with the Nios II SBT in the Nios II Command Shell

Table 2. Differences Between Nios II SBT for Eclipse and Nios II IDE (Part 2 of 2)

Feature	Nios II IDE	Nios II SBT for Eclipse
Modifying BSP properties	Limited control over BSP properties through the System Library page of the Properties dialog box.	Full control over BSP properties through the BSP Editor
Nios II C2H Compiler	Supported	Not supported
Nios II Flash Programmer	Limited control over the contents of the flash image	■ Allows you to create complex flash images containing Executable and Linking Format Files (.elf), SRAM Object Files (.sof), and binary data files in specified locations ■ Allows you to save flash programmer settings for repeatable flash programming
Scripting and regression testing	Not Available	Project creation and build can be scripted using the SBT command line

Converting IDE Projects to the SBT for Eclipse

If you have a pre-existing Nios II IDE project, before you can continue development in the Nios II SBT for Eclipse, you must convert it to the SBT and import it to the Eclipse environment. To convert and import an IDE project to the Nios II SBT for Eclipse, execute the following steps:

1. Convert the IDE project to an SBT-based project with the **nios2-convert-ide2sbt** command line utility.
2. Import the converted project to the Nios II SBT for Eclipse.

For a detailed description of the conversion process, refer to “Porting Nios II IDE Projects to the Software Build Tools” in the *Using the Nios II Integrated Development Environment* appendix of the *Nios II Software Developer's Handbook*. For a description of the process of importing a project, refer to “Importing a Command-Line Project” in the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*.

Learning about the Nios II SBT for Eclipse

Refer to the following documents to help you get acquainted with the Nios II SBT for Eclipse and related tools:

- *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*
- *Nios II Hardware Development Tutorial*
- The Comprehensive Nios II Embedded Design Suite support page
- The Nios II Software Build Tools for Eclipse support page

Errata Summary

Table 3 summarizes the issues that affect the Nios II Embedded Design Suite versions 8.1 through 9.1 SP2. For a detailed description of each issue, click on the issue name.

Refer to the Altera Knowledge Database for older errata and solutions.

Table 3. Nios II EDS Errata *(1)* (Part 1 of 5)

Added or Updated	Issue	Page	Affected Version				
			10.0	9.1 SP2	9.1 SP1	9.1	9.0
Nios II SBT for Eclipse							
July 2010	Nios II SBT for Eclipse Cannot Step into GCC 3 Library Source Code	10	✓	✓	✓	✓	—
	Error Building Imported Project: ‘No rule to make target’	10	✓	—	—	—	—
	Error Building Imported Project: ‘target pattern contains no %’	11	✓	—	—	—	—
	Missing Nios II Perspective	12	✓	✓	✓	✓	—
	Error Debugging Imported Project: ‘Can't find a source file’	12	Fixed	✓	✓	✓	—
	Error Marker Persists on BSP Project After Build Error Corrected	12	✓	✓	✓	✓	—
	Assembly Language Source File Not Found	13	Fixed	✓	✓	✓	—
	‘Exclude from build’ Not Supported	13	✓	—	—	—	—
May 2010	Errors Creating or Importing Software Projects	13	✓	✓	✓	✓	—
	Nios II Options Do Not Appear in Eclipse	14	✓	✓	✓	✓	—
Feb 2010	Build Errors on Software for Pre-Existing Design with SG-DMA	14	✓	✓	✓	✓	—
	Spurious System ID Mismatch Error	15	✓	✓	✓	✓	—
	Errors Debugging as Local C/C++ Application	15	✓	✓	✓	✓	—
Nov 2009	Error Messages on Console When Debugging	15	✓	✓	✓	✓	—
	java.lang.NullPointerException Error When Running Application	16	✓	✓	✓	✓	—
	Update Failure When Setting Empty Properties in the Nios II Properties Page	16	✓	✓	✓	✓	—
	Errors Converting Nios II IDE Multiprocessor Project	17	✓	✓	✓	✓	—
	DHCP Time-Out on EP3C120	17	✓	✓	✓	✓	—
	No Nios II Multiprocessor Collection Run Configuration	17	✓	✓	✓	✓	—
	Stop on Startup Option in Run Configuration Has No Effect	18	✓	✓	✓	✓	—
	Project Imported From Command Line Fails to Build Correctly	18	✓	✓	✓	✓	—
	Debugger Breaks in crt0.s Instead of main()	18	✓	✓	✓	✓	—
C2H Compiler							
Nov 2009	Error "java.lang.NoClassDefFoundError" During C2H Compilation	19	—	—	—	—	Fixed
Jul 2008	Functions Declared Without a Return Type Are Not Supported	19	✓	✓	✓	✓	✓
Oct 2007	Pre-7.1 Systems Are Not Supported	19	✓	✓	✓	✓	✓
	--src-dir SBT Argument Does Not Work With C2H	20	✓	✓	✓	✓	✓

Table 3. Nios II EDS Errata *(1)* (Part 2 of 5)

Added or Updated	Issue	Page	Affected Version				
			10.0	9.1 SP2	9.1 SP1	9.1	9.0
May 2007	Accelerator Generation Failure If Tools Are Installed in Path With Spaces	20	✓	✓	✓	✓	✓
	The C2H Compiler Regenerates an Accelerator Unnecessarily	20	✓	✓	✓	✓	✓
	Error: c2h_accelerator_base_addresses.h: No such file or directory	21	✓	✓	✓	✓	✓
	Java Heap Space Exception if Quartus II Compilation is Enabled	21	✓	✓	✓	✓	✓
	Pointer Dereferences to Volatile Types	21	✓	✓	✓	✓	✓
Dec 2006	C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File	22	✓	✓	✓	✓	✓
Jun 2006	Array Elements in Structures Do Not Copy Correctly	22	✓	✓	✓	✓	✓
	Clean Build Causes Build Failure	23	✓	✓	✓	✓	✓
	Changing Build Configurations Produces Unexpected Results	23	✓	✓	✓	✓	✓
	Hardware Accelerators Remain After Deleting the Software Project *(2)*	23	✓	✓	✓	✓	✓
	Incorrect Results From Logical or Conditional Operation With Side-Effects	24	✓	✓	✓	✓	✓
	Launch SOPC Builder Button in C2H View	24	✓	✓	✓	✓	✓
Development Boards							
Jun 2006	Intermittent Failures While Accessing CompactFlash Card	24	✓	✓	✓	✓	✓
Documentation Issues							
July 2010	Error Building Project: ‘No rule to make target’	25	✓	—	—	—	—
	AN543 Contains Incorrect Information about Updating the Flash	26	✓	✓	✓	✓	—
	Incorrect Information about Floating-Point Instruction Precision	26	Fixed	✓	✓	✓	—
	Eclipse CDT Features Not Supported by Nios II Plugins	27	✓	✓	✓	✓	—
May 2010	Incorrect Information about Nested Exceptions	28	✓	✓	✓	✓	—
Feb 2010	Missing Documentation of Interrupt API Properties	29	✓	✓	✓	✓	—
	Valid Range of hal.log_flags is –1 to 3	30	✓	✓	✓	✓	—
Nov 2009	NicheStack TCP/IP Stack - Nios II Edition Does Not Support TFTP	30	✓	✓	✓	✓	✓
	Error Message After Renaming Project: “Resource is out of sync with the system”	31	✓	✓	✓	✓	—
	Obsolete Wiki URLs in Help Systems and Handbook	31	—	—	—	—	Fixed
	Missing Source Control Information	31	—	—	—	—	Fixed
Mar 2009	Compiler Flags for Building Custom Newlib	33	✓	✓	✓	✓	✓
Nov 2008	Error Message on Invalid Exception or Reset Vector	34	✓	✓	✓	✓	✓
Oct 2007	Nios II IDE Online Help Expand Buttons Do Not Work	35	✓	✓	✓	✓	✓

Table 3. Nios II EDS Errata *(1)* (Part 3 of 5)

Added or Updated	Issue	Page	Affected Version				
			10.0	9.1 SP2	9.1 SP1	9.1	9.0
Hardware Abstraction Layer							
Feb 2010	Missing Structure Member Errors in alt_log_printf.c with Small JTAG UART Driver	35	✓	✓	✓	✓	—
	“unused variable” Warning When Building alt_main.c	36	✓	✓	✓	✓	—
	Build Warnings in Performance Counter Driver Files	36	✓	✓	✓	✓	—
Hardware Example Designs							
Mar 2007	Incorrect PLL Phase Shift in Stratix II and Cyclone II Designs	37	✓	✓	✓	✓	✓
Software Examples							
Nov 2009	Hardware Tutorial Software Example Hangs on Some Boards	38	✓	✓	✓	✓	✓
Nov 2008	Simple Socket Server Example Displays Incorrect IP Address	38	✓	✓	✓	✓	✓
Jun 2006	Networking Examples	39	✓	✓	✓	✓	✓
Flash Programmer							
Feb 2010	Unable to Configure FPGA from Flash with Parallel Flash Loader	39	Fixed	Fixed	Fixed	✓	✓
	Boot Failure or elf2flash Error Due to Incorrect Flash File Size	40	✓	✓	✓	✓	✓
May 2007	Delay When Creating New Flash Programmer Configuration	41	✓	✓	✓	✓	✓
Dec 2006	elf2flash File Size Limit	41	✓	✓	✓	✓	✓
	Error Message “No such file or directory” When Programming Flash	41	✓	✓	✓	✓	✓
Hardware Simulation							
May 2010	Vectored Interrupt Controller Does Not Support VHDL Simulation Models	41	✓	✓	✓	✓	—
Dec 2006	Error “UNC paths are not supported” Launching ModelSim	42	✓	✓	✓	✓	✓
Jun 2006	Uninitialized .bss Variables in Simulation	42	✓	✓	✓	✓	✓
Host Platform							
July 2010	Nios II EDS Requires Administrative Permissions on Windows Vista	42	✓	✓	✓	✓	—
Nov 2009	Cannot Launch Nios II SBT for Eclipse on Red Hat Linux	43	✓	✓	✓	✓	—
Dec 2007	Windows/Cygwin: Nios II Processor Generation Failure	43	✓	✓	✓	✓	✓
Oct 2007	Windows Vista: Limited Support in Nios II EDS	44	✓	✓	✓	✓	✓
Nios II IDE							
Building Projects							
Mar 2009	Nios II IDE Command-Line Tools Select Wrong Workspace on Linux	44	✓	✓	✓	✓	✓
	Nios II IDE Command-Line Tools Hang on Windows	45	✓	✓	✓	✓	✓
Nov 2008	Nios II IDE Cannot Find stdio.h in Outline View	46	✓	✓	✓	✓	✓

Table 3. Nios II EDS Errata *(1)* (Part 4 of 5)

Added or Updated	Issue	Page	Affected Version				
			10.0	9.1 SP2	9.1 SP1	9.1	9.0
Jul 2008	Nios II IDE Hangs With UNC Project Path	46	✓	✓	✓	✓	✓
	Build Failure with Nios II Advanced Exceptions, MMU, and MPU	46	✓	✓	✓	✓	✓
May 2007	Build Command Not Functional for BSPs Created With the Nios II SBT	46	✓	✓	✓	✓	✓
	Linker Errors with Dual-Port Memories	47	✓	✓	✓	✓	✓
	User-Managed BSP Settings Not Supported in Nios II IDE	47	✓	✓	✓	✓	✓
	URL Project Location Causes Project Creation Error	47	✓	✓	✓	✓	✓
Dec 2006	Compilation Error with Separate Exception Stack Option	48	✓	✓	✓	✓	✓
	Incorrect Stack and Heap Free Space Report	48	✓	✓	✓	✓	✓
	Nios II IDE Reports Problems Without Displaying Error in Console	48	✓	✓	✓	✓	✓
Debugging Projects							
Nov 2009	Method for Accessing MMU and MPU Registers in the Debugger	49	✓	✓	✓	✓	✓
Nov 2008	IDE Cannot Display Imported Profiling Data	49	✓	✓	✓	✓	✓
Jul 2008	Error Message when Downloading .elf File	50	✓	✓	✓	✓	✓
	Variable Casting Unsupported in ISS	50	✓	✓	✓	✓	✓
	Trace Debug Does Not Support Instruction-Related Exceptions	50	✓	✓	✓	✓	✓
Oct 2007	Trace Debug Does Not Support the JMPI Instruction	50	✓	✓	✓	✓	✓
May 2007	Cannot Locate Source Code in Driver Files Shared by Multiple Projects	51	✓	✓	✓	✓	✓
	Console Window Is Not Updated After ISS Error	51	✓	✓	✓	✓	✓
	ISS Fails on Designs Containing Triple Speed Ethernet MAC or SG-DMA Components	51	✓	✓	✓	✓	✓
	Memory Window Sets Control Register Values Incorrectly	52	✓	✓	✓	✓	✓
Dec 2006	Programs That Interact With a Terminal Console on Windows Do Not Work	52	✓	✓	✓	✓	✓
	“Run as ModelSim” in the Nios II IDE Fails	52	✓	✓	✓	✓	✓
	The Restart Command on the Run Menu Does Not Work	52	✓	✓	✓	✓	✓
Jun 2006	Watchpoints Do Not Work on Certain Variables	53	✓	✓	✓	✓	✓
Navigating Projects							
Dec 2006	Nios II IDE Freezes While Displaying the Splash Screen	53	✓	✓	✓	✓	✓
	Internal Error When Double-Clicking on a Large Objdump File	53	✓	✓	✓	✓	✓
Jun 2006	C/C++ Scanner Does Not Support Certain C/C++ Constructs	54	✓	✓	✓	✓	✓
Nios II SBT							
July 2010	Error Running Nios II Project: ‘Downloading ELF Process failed’	54	✓	✓	✓	✓	—
May 2010	BSP Editor Does Not Show Command-Line Help	55	✓	✓	✓	✓	—
Feb 2010	BSP Not Updated for Memory Size Changes in SOPC Builder	55	✓	✓	✓	✓	—

Table 3. Nios II EDS Errata *(1)* (Part 5 of 5)

Added or Updated	Issue	Page	Affected Version				
			10.0	9.1 SP2	9.1 SP1	9.1	9.0
Nov 2009	Build Errors on Interrupt API Calls	55	✓	✓	✓	✓	—
	BSP Editor Incorrectly Allows Mixed Drivers	56	—	—	—	Fixed	✓
	nios2-bsp --help Displays Incorrect List of BSP Types	57	—	—	—	Fixed	✓
	Java Heap Space Error When Creating BSP	57	—	—	—	—	Fixed
Oct 2007	SBT Fails if Nios II EDS is Installed in a Path Containing Spaces	58	✓	✓	✓	✓	✓
May 2007	User-Managed BSP Settings Not Supported in Nios II IDE	47	✓	✓	✓	✓	✓
	Upper-Case File Extensions Not Supported	58	✓	✓	✓	✓	✓
Peripherals							
May 2010	Vectored Interrupt Controller Does Not Support VHDL Simulation Models	41	✓	✓	✓	✓	—
Jul 2008	Unaligned Transfers of Small Payloads Fail on SG-DMA	59	✓	✓	✓	✓	✓
May 2007	DMA Controller Always Busy in Burst Mode	59	✓	✓	✓	✓	✓
	Non-System-Wide Reset Can Cause Improper Initialization of Mailbox Core	59	✓	✓	✓	✓	✓
Target Software							
Nov 2009	NicheStack tcp_wake() Function Does Not Awaken All select() Sleepers	60	✓	✓	✓	✓	—
	EIC-Based Nios II System Crashes With NicheStack	60	✓	✓	✓	✓	—
Mar 2009	Gigabit Ethernet Performance Issues	60	—	—	—	—	Fixed
Dec 2006	stdio Does Not Work with MicroC/OS-II and Small C Library	61	✓	✓	✓	✓	✓
Jun 2006	cout From MicroC/OS-II Task Does Not Send Data to stdout	61	✓	✓	✓	✓	✓
	malloc(), realloc() Failures With MicroC/OS-II	61	✓	✓	✓	✓	✓
Toolchain (gcc, gdb, etc.)							
July 2010	C++ Exceptions Unsupported in a Multi-Threaded Environment	62	✓	✓	✓	✓	✓
	C++ Exceptions Unsupported by the Nios II GCC 4 Toolchain	62	✓	—	—	—	—
	memcpy() Optimization Misalignment	63	✓	✓	✓	✓	✓
Feb 2010	Unable to Configure FPGA from Flash with Parallel Flash Loader	39	—	—	Fixed	✓	✓
Dec 2006	GNU Assembler Does Not Accept the --defsym Flag	63	✓	✓	✓	✓	✓

Notes to Table 3:

(1) Refer to the Altera Knowledge Database for older errata and solutions.

(2) For further information, refer to the *Nios II C2H Compiler User Guide*.

Nios II SBT for Eclipse Errata

This section describes in detail the issues related to the Nios II SBT for Eclipse.

Nios II SBT for Eclipse Cannot Step into GCC 3 Library Source Code

If your project uses the Nios II GCC 3 toolchain, when debugging with the Nios II SBT for Eclipse, you might be unable to step into library source code. A message such as the following message appears:

```
Resource '/xxx/xxx.c' does not exist.
```

Workaround

Upgrade to the Nios II EDS v10.0 or later and use the Nios II GCC 4 toolchain.

Solution Status

Not fixed

Error Building Imported Project: ‘No rule to make target’

If an application or library project is created with v9.1 SP2 or earlier (using the GCC 3 toolchain), and you import it to the v10.0 SBT for Eclipse using the GCC 4 toolchain, you might see the build error shown in Example 1.

Example 1.

```
make all
Info: Building ../hw_bsp
make --no-print-directory -C ../hw_bsp
make[1]: *** No rule to make target `/cygdrive/c/tmp/
verilog_niosII_stratixII_2s60_standard/software/hw_bsp/
alt_sys_init.c', needed by `all'. Stop.
make: *** [../hw_bsp-recurs-make-lib] Error 2
```

This error can also happen if you import a GCC 3 project to the v10.0 SBT for Eclipse using the GCC 3 toolchain, and later change the toolchain to **MinGW Nios II GCC4**.

This issue only affects platforms running the Windows operating system.

Workaround

To avoid this error, follow these steps:

1. Edit the application project and library project makefiles and add the italicized code as shown in Example 2.

Example 2.

```
#------------------------------------------------------------------------------
# The adjust-path macro
#
# If COMSPEC is defined, Make is launched from Windows through
# Cygwin. This adjust-path macro will call 'cygpath -u' on all
# paths to ensure they are readable by Make.
#
# If COMSPEC is not defined, Make is launched from *nix, and no adjustment
# is necessary
#------------------------------------------------------------------------------

# Beginning of inserted code
ifndef COMSPEC
ifdef ComSpec
COMSPEC = $(ComSpec)
endif # ComSpec
endif # !COMSPEC
#end of inserted code

ifdef COMSPEC
adjust-path = $(shell cygpath -u "$1")
adjust-path-mixed = $(shell cygpath -m "$1")
else
adjust-path = $1
adjust-path-mixed = $1
endif
```

2. Right-click the BSP project, point to **Nios II** and click **Generate BSP**.
3. Clean and rebuild the application and library projects.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Error Building Imported Project: ‘target pattern contains no %’

If your application or library makefile uses an absolute path and is generated with the GCC 3 toolchain, and you import it to the v10.0 SBT for Eclipse using the GCC 4 toolchain, you get the following build error:

```
makefile:587: *** target pattern contains no '%'. Stop.
```

Workaround

Recreate the project using the SBT for Eclipse v10.0.

Alternatively, use the GCC 3 toolchain.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Missing Nios II Perspective

If you launch the Nios II SBT for Eclipse from a universal naming convention (UNC) path, the Nios II perspective does not appear in Eclipse.

This issue appears only on platforms running the Windows operating system.

Workaround

Do not use a UNC path to launch the Nios II SBT for Eclipse. For example, if the SBT for Eclipse is installed on a network drive, map the network path to a Windows drive letter.

Solution Status

Not fixed

Error Debugging Imported Project: 'Can't find a source file'

If you create a project using the Nios II SBT in the Nios II Command Shell, and then import the project to Nios II SBT for Eclipse for debugging, the debugger cannot find source files located outside of project folder.

Example of error message:

```
Can't find a source file at "../src/test.c"
```

Workaround

To link the affected source files to the project, follow these steps:

1. Right-click the project and click **Properties** to open the **Properties** dialog box.
2. Click **+** to expand **C/C++ General**, and click the **Paths and Symbols** tab.
3. Click the **Source Location** tab.
4. Click **Link Folder** and navigate to the folder containing the source file.

Solution Status

Fixed in v10.0 of the Nios II EDS

Error Marker Persists on BSP Project After Build Error Corrected

In the Nios II SBT for Eclipse, if an application project fails to build, error markers appear in the BSP project, if the build error message contains a path to BSP project source files. Even after you have rebuilt the application project without errors, the error markers persist in the BSP project.

Workaround

Rebuild your BSP project.

Alternatively, if your application will always be associated with the same BSP, set the Eclipse project references by following these steps:

1. Right-click your application project, point to **Properties** and click **Project References**.
2. Select the desired BSP for your application.
3. Build your application project again to clear the error markers in the BSP project.

If you later link your application project with a different BSP, you must manually correct the Eclipse project references.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Assembly Language Source File Not Found

The Nios II SBT for Eclipse does not support assembly language files with the upper case file extension **.S**. If your project includes a file with an upper case extension, the compiler is not able to find the assembly language file during build.

Workaround

Rename the assembly language files with the lower-case extension **.s**.

Solution Status

Fixed in v10.0 of the Nios II EDS

‘Exclude from build’ Not Supported

In the Nios II SBT for Eclipse v9.1 SP2 and earlier, you can use the **Exclude from build** command to omit a source file from your project at build time. Starting with v10.0, **Exclude from build** is replaced by the **Remove from Nios II build** and **Add to Nios II build** commands.

Due to an Eclipse platform limitation, **Exclude from build** still appears in the project context menu in v10.0. However, it has no effect.

Workaround

To exclude a source file from your project when it is built, on the project context menu, click **Remove from Nios II build**. To restore it to the build, click **Add to Nios II build**.

Errors Creating or Importing Software Projects

If you place software projects in your Eclipse workspace directory, you might experience project-related errors. For example, you might see one of the following error messages:

-
  ```
  Unable to create project
  Reason: Unable to create project in workspace directory
  ```

-
  ```
  Failed to import
  Reason: Unable to import project
  ```

Workaround

Switch to a different workspace, or create a new workspace, separate from your project files. To switch workspaces or create a workspace, on the File menu, click **Switch Workspace**.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Nios II Options Do Not Appear in Eclipse

When Nios II SBT for Eclipse starts, the Nios II plugins might fail to load, resulting in the following symptoms:

- The Nios II perspective is not available
- No Nios II items appear when you click **New** in the File menu

Eclipse can behave this way if you have installed the Nios II EDS in the same directory as a previous installation.

When the Altera Complete Design Suite (ACDS) installer installs the Nios II EDS, you specify a root directory for the EDS file structure. For example, if the Nios II EDS 9.1 is installed on the Windows operating system, the root directory might be **c:\altera\91\nios2eds**. For simplicity, Altera documentation refers to this directory as <*Nios II EDS install path*>.

When the ACDS installer removes the Nios II EDS, it leaves behind some files in <*Nios II EDS install path*>. If you reinstall the Nios II EDS in the same directory, these leftover files might cause the Nios II SBT for Eclipse to work incorrectly.

Workaround

To correct this problem, execute the following steps:

1. Uninstall the Nios II EDS.
2. Delete <*Nios II EDS install path*>, including all subdirectories, before reinstalling the Nios II EDS.

   Alternatively, reinstall the Nios II EDS in a different directory.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Build Errors on Software for Pre-Existing Design with SG-DMA

If you have a Nios II system generated with SOPC Builder v. 9.0 or earlier, and it contains the Scatter-Gather DMA (SG-DMA) component, you cannot build software for it with the Nios II SBT for Eclipse v. 9.1 or later. If you attempt to do so, the Nios II compiler reports errors.

Workaround

Regenerate the design with SOPC Builder v. 9.1 or later before attempting to compile the software in the Nios II SBT for Eclipse.

Solution Status

Not fixed

Spurious System ID Mismatch Error

You might see the following error message when attempting to run or debug a Nios II software project, or to edit a launch configuration:

```
[Target Connection]:System ID mismatch - actual: "0xffffffff", expected: "<system id>"
```

<system id> is replaced by the system ID in your software project. When this error occurs, you cannot run or debug software.

This error normally means that the FPGA is unconfigured, or configured with a **.sof** file that does not correspond to your software project. However, the error occasionally appears even when the FPGA is configured with the correct **.sof**.

Workaround

After verifying that the FPGA is configured with the correct **.sof**, you can use one of the following methods to work around this issue:

- In the **Run Configuration** or **Debug Configuration** dialog box, on the **Target Connection** tab, click **Refresh Connections** repeatedly until the error message disappears.
- Delete the launch configuration and create a new launch configuration.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Errors Debugging as Local C/C++ Application

If you try to debug a Nios II software project as the Local C/C++ Application launch configuration type, you see one of the following errors, depending on your platform:

- Windows: “Error creating session”
- Linux: “Execution is suspended because of error”

The Nios II debug perspective fails to open.

This is expected CDT behavior in the Eclipse platform. Local C/C++ Application is the launch configuration type for a standard CDT project. To invoke the Nios II plugins, you must use a Nios II launch configuration type.

Workaround

Always debug Nios II software projects as the Nios II Hardware launch configuration type.

Error Messages on Console When Debugging

When debugging, you might see the following messages in the Nios II SBT for Eclipse console:

```
No symbol "auto" in current context.
No symbol "new" in current context.
```

These are normal messages from the GNU Debugger (GDB) console, not error messages. This issue does not impact the debug process.

Workaround

No workaround needed.

Solution Status

None needed.

java.lang.NullPointerException Error When Running Application

When you attempt to run or debug an application, the Nios II SBT for Eclipse might show an internal error message similar to the following:

```
Launching New_configuration java.lang.NullPointerException
```

This error occurs in either of the following circumstances:

- You attempt to run your project without rebuilding after performing a `make clean`.
- You delete a run configuration belonging to your project while the project is running.

You must restart Nios II SBT for Eclipse to recover from this error.

Workaround

Build the project before attempting to run or debug it. Do not delete a run configuration while it is in use.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Update Failure When Setting Empty Properties in the Nios II Properties Page

If you use the Nios II Properties page to update more than one board support package (BSP) setting to the empty string (`""`), the BSP is not updated.

Workaround

You can use either of the following workarounds to set multiple BSP settings to the empty string:

- Use the keyword `none` instead of `""` to set settings to the empty string.
- Use the BSP Editor instead of the Nios II Properties page.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Errors Converting Nios II IDE Multiprocessor Project

If you try use the **nios2-convert-ide2sbt** utility to convert an IDE multiprocessor project to the Nios II SBT, error messages appear, and the project fails to generate properly.

Workaround

Create new Nios II SBT application and BSP projects using project settings equivalent to the original Nios II IDE project settings.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

DHCP Time-Out on EP3C120

If you create a project based on the web server template targeting the Altera Embedded Systems Development Kit, Cyclone® III Edition (EP3C120) board, the system is unable to get an IP address at startup. DHCP times out and the system reverts to the default IP address.

Workaround

Create the project in the Nios II Command Shell and then import it to the Nios II SBT for Eclipse, by executing the following steps:

1. In the Nios II Command Shell, navigate to the following path:

   *<Nios II EDS install path>***/examples/***<language>***/niosII_cycloneIII_3c120/tserd/software_examples/app/web_server_tse_3c120_rgmii**

   where *<language>* is either **vhdl** or **verilog**.

2. Type the following command to create a web server example for the EP3C120 kit:

   ```
   create-this-app↵
   ```

3. Import the command-line project to the Nios II SBT for Eclipse.

   For information about importing command-line projects, refer to "Importing a Command-Line Project" in the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

No Nios II Multiprocessor Collection Run Configuration

The multiprocessor collection run configuration is not available in the Run menu. Therefore, you cannot use a single run configuration to download Executable and Linking Format Files (**.elf**) to run on multiple Nios II processors in the Nios II SBT for Eclipse.

Workaround

Create a separate run configuration for each Nios II processor in the hardware design, and launch them one at a time

Solution Status

Not fixed

Stop on Startup Option in Run Configuration Has No Effect

Turning on **Stop on Startup at** in the **Debugger** tab of a run configuration has no effect.

Workaround

You can manually set any breakpoint in the debug perspective.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Project Imported From Command Line Fails to Build Correctly

If you try to import a command-line project to the Nios II SBT for Eclipse, and your project has source files located outside the project folder, the Nios II SBT for Eclipse fails to build your project because the makefile does not include source files located outside the project folder.

Workaround

You can work around this issue with either of the following methods:

- When importing the project, disable source management through the **Nios II Application Properties** or **Nios II Library Properties** dialog box.

  ☞ You must manually keep your makefile updated to reflect source files added to or removed from the project.

- Manually link the affected source files to the project. Right-click the project, click **Properties**, expand **C/C++ General**, select **Paths and Symbols** and then **Source Location**, and click **Link Folder**.

Solution Status

This issue will be fixed in future release of the Nios II EDS.

Debugger Breaks in crt0.s Instead of main()

If you attempt to launch a debug session by right clicking the project, without first creating a run configuration, the debugger breaks in the initialization code in **crt0.s**, rather than at the top of the `main()` function. You must explicitly create the run configuration before launching the debug session.

Workaround

You can work around this issue with either of the following methods:

- Create the run configuration with the **Run Configuration** dialog box. You can then debug with this run configuration by right-clicking the project, clicking **Debug As**, and selecting the debug configuration. The program correctly breaks at `main()`.
- Manually set a breakpoint at `main()`.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

C2H Compiler Errata

This section describes in detail the issues related to the Nios II C2H Compiler.

Error "java.lang.NoClassDefFoundError" During C2H Compilation

C2H compilation fails with the following error:

```
java.lang.NoClassDefFoundError: com/altera/c2h/C2Hgenerate
Exception in thread "main" make: *** [c2h_hdl-t] Error 1
```

This error occurs when you invoke C2H from the Nios II IDE on the Windows operating system. This problem does not affect Linux installations, nor C2H compilation from the Nios II command shell.

Workaround

To correct the problem, visit mySupport and request the Nios II EDS 8.1 Patch 0.01 (PC version).

Solution Status

Fixed in v. 9.0 of the Nios II EDS

Functions Declared Without a Return Type Are Not Supported

The C2H compiler does not support functions without an explicitly declared return type.

Workaround

If you are using the implicit `int` return type, declare the return type explicitly. If your function has no return value, declare it as `void`.

Solution Status

Not fixed

Pre-7.1 Systems Are Not Supported

Starting in v. 7.1, the C2H Compiler does not support SOPC Builder systems created in SOPC Builder 7.0 or earlier (systems based on a **class.ptf** file).

Workaround

Before using the C2H Compiler, you must upgrade your SOPC Builder system. Open the system in SOPC Builder v. 7.1 or later. SOPC Builder prompts you to convert the system, creating a ***_hw.tcl** file compatible with the C2H Compiler.

Solution Status

Not fixed

--src-dir SBT Argument Does Not Work With C2H

When using the C2H Compiler with Nios II SBT, an error occurs when using the `nios2-app-generate-makefile` argument `--src-dir` to specify a directory of source files. An example of the error is:

```
multiple definition of 'my_accelerated_function'
```

Workaround

Specify source files individually using the `--src-files` argument.

Solution Status

Not fixed

Accelerator Generation Failure If Tools Are Installed in Path With Spaces

If the path to your installation of the Altera Design Suite contains spaces, the C2H Compiler fails to generate the accelerators.

Workaround

Reinstall the tools to a path containing no spaces.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

The C2H Compiler Regenerates an Accelerator Unnecessarily

The C2H Compiler might regenerate an accelerator even when the accelerated function is unchanged. This problem can result from changes to files included by the C file containing the accelerated function. The C2H Compiler fails to check that the generated hardware description language (HDL) matches the previously generated HDL causing the system to be regenerated.

Workaround

To avoid this issue move the accelerator-specific information from the include file to a separate include file. This workaround prevents regeneration of the system when the HDL is unchanged.

Solution Status

Not fixed

Error: c2h_accelerator_base_addresses.h: No such file or directory

When a C2H accelerator is compiled for the first time, the following compile-time error can result if the **Analyze all accelerators** option is selected:

```
c2h_accelerator_base_addresses.h: No such file or directory.
```

Workaround

Click **Build software and generate SOPC Builder system** and build once before building with the **Analyze all accelerators** option.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Java Heap Space Exception if Quartus II Compilation is Enabled

If your design contains a C2H accelerator, and you select **Build software, generate SOPC Builder system, and run Quartus II compilation** in the C2H view, you might see the following error during Quartus® II compilation:

```
Exception in thread "main" java.lang.OutOfMemoryError: Java
heap space

make: *** [c2h_hdl-t] Error 1
```

Workaround

Select the **Build software and generate SOPC builder system** option in the C2H Compiler settings window, and then manually launch the Quartus II software to compile the design.

Solution Status

Not fixed

Pointer Dereferences to Volatile Types

The C2H Compiler treats pointer dereferences to a volatile type as if they alias all other pointer dereferences. Pointers that are restrict-qualified are treated the same way.

The two loops in Example 3 cannot be scheduled concurrently because the `volatile` qualification overrides the `__restrict__` pragma.

Example 3. Non-Concurrent Loops

```
volatile int * __restrict__ fifo_rd = FIFO_RD_BASE;
volatile int * __restrict__ fifo_wr = FIFO_WR_BASE;
for ()
{
  *fifo_wr = ....;
}
for ()
{
  ... = *fifo_rd;
}
```

Workaround

Divide the function into multiple interrupt request (IRQ)-enabled accelerators that are launched concurrently from the processor, and use FIFO buffers to communicate between them.

Solution Status

Not fixed

C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File

When accelerating a function in a file, the C2H Compiler cannot link subfunctions that are defined in a different file.

Workaround

Include all subfunctions called by the accelerated function within the same source code file.

Solution Status

Not fixed

Array Elements in Structures Do Not Copy Correctly

C2H accelerators do not correctly copy array elements that are elements of structures.

In Example 4, the `a` and `b` elements of the structure copy correctly, but the `buf` element does not. After this assignment, `struct_a` equals `{9, 8, {3, 3, 3, 3}}`.

Example 4. Array Elements of Structs

```
typedef struct my_struct {
  int a;
  int b;
  int buf[BUF_SIZE];
}MY_STRUCT;
MY_STRUCT struct_a = {1, 2, {3, 3, 3, 3}};
MY_STRUCT struct_b = {9, 8, {7, 7, 7, 7}};
struct_a = struct_b;
```

Workaround

Copy the array elements explicitly, as shown in Example 5.

Example 5. Copying Array Elements Explicitly

```
{
  int i=0;
  do
    {
      struct_a.buf[i] = struct_b.buf[i];
      i++;
    } while (i<LENGTH_OF_BUF_ELEMENT)
}
```

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Clean Build Causes Build Failure

Performing a clean build on a Nios II IDE project that contains a hardware accelerator can cause the next build to fail in the IDE, because the clean build erroneously deletes a file required by the C2H Compiler.

Workaround

Do not perform a clean build on projects that use hardware accelerators. If you have already performed a clean build, recompile with option **Build software, generate SOPC Builder system, and run Quartus II compilation** to regenerate the necessary files.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Changing Build Configurations Produces Unexpected Results

The C2H Compiler does not support multiple build configurations (for example Release or Debug) in the Nios II IDE. After creating one or more accelerators in a particular configuration, the C2H Compiler produces undefined results if you switch to a different build configurations and create more accelerators.

Workaround

For a specific SOPC Builder system and Nios II IDE project, specify C2H accelerators in only one build configuration. You can use multiple build configurations, as long as only one configuration specifies C2H Compiler settings.

Solution Status

Not fixed

Hardware Accelerators Remain After Deleting the Software Project

If a system contains C2H accelerators, deleting the software project that defines the accelerators does not remove the accelerators from the hardware system, and the accelerator logic remains in the SOPC Builder system.

Workaround

To remove an accelerator from a system, delete the accelerator from the C2H view in the Nios II IDE first, and then recompile the software project. The C2H Compiler then removes the accelerator from the SOPC Builder system. Once the compilation is complete then the software application can be deleted from the workspace.

Solution Status

Not fixed

Incorrect Results From Logical or Conditional Operation With Side-Effects

The C2H Compiler always evaluates both operands of logical (&&, ||) and conditional (?:) operators. This is different from expected American National Standards Institute (ANSI) C behavior, where operands are evaluated left-to-right, and unnecessary operands are skipped.

For example, in the expression (i-- && j--), if the value of i is zero, ANSI C does not evaluate the right-hand-side (RHS) expression, and j is not decremented. By contrast, the C2H Compiler evaluates both sides, decrementing j.

The following expressions are other examples that might be affected by this issue:

```
if (i-- || j++)
    ...;


a = ((cond == 1)?  i++ : j++);
```

Workaround

Use logical and conditional operations whose operators have no side effects. Operations with side effects include pre- and post- increment and decrement operations (++, --), memory operations (*, [], ., ->), and function calls.

Solution Status

Not fixed

Launch SOPC Builder Button in C2H View

When the Nios II IDE workspace contains multiple projects with multiple system libraries, the incorrect SOPC Builder system might open when you click **Launch SOPC Builder** in the C2H view.

Workaround

Launch SOPC Builder from the Quartus II software. Alternatively, keep only one system library project open at a time while using the C2H Compiler.

Solution Status

Not fixed

Development Board Errata

This section describes in detail the Nios II EDS issues related to Nios development boards.

Intermittent Failures While Accessing CompactFlash Card

The Nios II Development Kit version 5.0 and later includes a CompactFlash controller peripheral suitable for interfacing to CompactFlash cards in True IDE mode on Nios development boards. For True IDE mode to operate, CompactFlash cards require that the ATASEL_N input be driven to ground during power-up.

The CompactFlash controller peripheral includes a configurable power register, used to cycle power to CompactFlash cards from Nios II software through a metal oxide semiconductor field-effect transistor (MOSFET) on the Nios development boards. However, in certain development boards, power to the CompactFlash card does not turn off completely during this power cycle operation. Because of this condition, the CompactFlash might not sample the `ATASEL_N` pin during the power-cycle operation after FPGA configuration when this pin is driven to ground. Instead, the CompactFlash card might sample the `ATASEL_N` pin when power is first applied to the development board, when I/O is not yet driven by the FPGA (before FPGA configuration).

Workaround

If you encounter errors with CompactFlash when using the Nios development boards, try one of the following solutions:

- Use a different CompactFlash card. Certain cards are more susceptible to the power-cycling issue than others.
- Modify the Nios development board. This is recommended if you are familiar and comfortable with board-level modifications. Disconnect pin 9 (`ATASEL_N`) on the CompactFlash socket on your Nios development board and tie this pin to ground.

☞ The CompactFlash socket uses a staggered numbering on the pins (starting from pin 1: 1, 26, 2, 27, ...); refer to the CompactFlash Association specification for right-angle surface-mount connectors for exact specifications on this connector. This modification permanently enables True IDE mode operation.

Solution Status

Not fixed

Documentation Errata

This section describes in detail the Nios II EDS documentation issues.

Error Building Project: ‘No rule to make target’

The *Getting Started with Graphical User Interface* and *Getting Started from the Command Line* chapters of the *Nios II Software Developer’s Handbook* do not include the following information:

You cannot link Nios II GCC 3 projects with Nios II GCC 4 projects. Your application, library and BSP projects must all use the same Nios II GCC toolchain.

If you switch between Nios II GCC 3 and Nios II GCC 4, make sure you run **make clean** on your application, library and BSP projects before rebuilding.

When importing a GCC 3 BSP, such as a project created with v9.1 SP2 or earlier, to v10.0 SBT for Eclipse using the GCC 4 toolchain, after importing the BSP, regenerate the makefile.

If you do not follow these rules, you might see the following error:

```
make[1]: *** No rule to make target `/cygdrive/c/.../bsp/
alt_sys_init.c', needed by 'all'. Stop.
make: *** [../bsp/-recurs-make-lib] Error 2
```

Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

AN543 Contains Incorrect Information about Updating the Flash

"Updating the Flash Option Bits on the Cyclone III Development Board" in AN543, *Debugging Nios II Software Using the Lauterbach Debugger* contains incorrect information about updating the flash.

Workaround

The correct information is as follows:

Updating the Flash on the Cyclone III Development Board

To restore the flash factory image, refer to "Restoring the Factory Design to the Flash Device" in the *Cyclone III Development Kit User Guide*.

Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

Incorrect Information about Floating-Point Instruction Precision

In "Arithmetic Logic Unit" in the *Processor Architecture* chapter, the *Nios II Processor Reference Handbook* incorrectly states that the Nios II compiler treats floating-point constants as double-precision numbers.

Workaround

The correct information is as follow:

All the floating-point custom instructions are single-precision operations. Double-precision floating-point operations are implemented in software.

When the floating-point custom instructions are not present, the Nios II compiler treats floating-point constants as double-precision values. However, with the floating-point custom instructions, the Nios II compiler treats floating-point constants as single-precision numbers by default. This allows all floating-point expressions to be evaluated in hardware, at a possible cost in precision.

If you do not wish floating-point constants to be cast down to single precision values, append `L` to each constant value, to instruct the compiler to treat the constant as a double-precision floating-point value. In this case, if an expression contains a floating-point constant, each term in the expression is cast to double precision. As a result, the expression is computed with software-implemented double-precision arithmetic, at a possible cost in computation speed.

Table 4 shows code examples using floating-point constants, indicating how each computation is implemented.

Table 4. Floating-Point Constant Examples

Example Code	Floating-Point Custom Instructions Present?	Precision	Implementation
`b = a × 4.67`	No	Double	Software
`b = a × 4.67f`	No	Single	Software
`b = a × 4.67L`	No	Double	Software
`b = a × 4.67`	Yes	Single	Hardware
`b = a × 4.67f`	Yes	Single	Hardware
`b = a × 4.67L`	Yes	Double	Software

Solution Status

Fixed in v10.0 documentation

Eclipse CDT Features Not Supported by Nios II Plugins

The following information is missing from the *Getting Started with Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*:

The features listed in the left column of Table 5 are supported by the Eclipse CDT plugins, but are not supported by Nios II plugins. The right column lists alternative features supported by the Nios II plugins.

Table 5. Eclipse CDT Features Not Supported by the Nios II Plugins (Part 1 of 2)

Unsupported CDT Feature	Alternative Nios II Feature
New Project Wizard C/C++ ■ C Project ■ C++ Project ■ C/C++ ■ Convert to a C/C++ Project ■ Source Folder	To create a new project, use one of the following Nios II wizards: ■ Nios II Application ■ Nios II Application and BSP from Template ■ Nios II Board Support Package ■ Nios II Library
Build configurations Right-click project and point to **Build Configurations**	The Nios II plugins only support a single build configuration.
Exclude from Build (from v10.0 onwards) Right-click source files	Use **Remove from Nios II Build** and **Add to Nios II Build.** See "'Exclude from build' Not Supported" on page 13.

Table 5. Eclipse CDT Features Not Supported by the Nios II Plugins (Part 2 of 2)

Unsupported CDT Feature	Alternative Nios II Feature
Project Properties	
C/C++ Build	
■ Builder Settings	
■ Makefile generation	By default, the Nios II SBT generates makefiles automatically.
■ Build location	The build location is fixed.
■ Behaviour	
■ Build on resource save (Auto build)	
■ Build Variables	
■ Discovery Options	
■ Environment	
■ Settings	
■ Tool Chain Editor	
■ Current builder	
■ Used tool	To change the toolchain, use the **Current tool chain** option
C/C++ General	
■ Enable project specific settings	
■ Documentation	
■ File Types	
■ Indexer	
■ Build configuration for the indexer	The Nios II plugins only support a single build configuration.
■ Language Mappings	
■ Paths and Symbols	Use **Nios II Application Properties** and **Nios II Application Paths**
Window Preferences	
C/C++	
■ Build scope	
■ Build project configurations	The Nios II plugins only support a single build configuration.
■ Build Variables	
■ Environment	
■ File Types	
■ Indexer	
■ Build configuration for the indexer	The Nios II plugins only support a single build configuration.
■ Language Mappings	
■ New CDT project wizard	

Incorrect Information about Nested Exceptions

"Exception Processing" in the *Programming Model* chapter of the *Nios II Processor Reference Handbook* incorrectly states that multiple interrupts with different requested interrupt levels (RILs) must not be assigned to the same shadow register set.

Workaround

The correct information is as follows:

Multiple interrupts with different RILs can be assigned to a single shadow register set. However, with multiple register sets, you must not allow the RILs assigned to one shadow register set to overlap the RILs assigned to another register set.

Table 2–6 and Table 2–7 illustrate the validity of register set assignments when preemption within a register set is enabled.

Table 2–6. Example of Illegal RIL Assignment

RIL	Register Set 1	Register Set 2
1	IRQ0	
2	IRQ1	
3		IRQ2
4	IRQ3	
5		IRQ4
6		IRQ5
7		IRQ6

Table 2–7. Example of Legal RIL Assignment

RIL	Register Set 1	Register Set 2
1	IRQ0	
2	IRQ1	
3	IRQ3	
4		IRQ2
5		IRQ4
6		IRQ5
7		IRQ6

Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

Missing Documentation of Interrupt API Properties

“Tcl Commands” in the *Nios II Software Build Tools Reference* chapter of the *Nios II Software Developer’s Handbook* does not include information about two `set_sw_property` command properties related to the application programming interfaces (APIs) for interrupts.

Workaround

The missing information belongs with the `set_sw_property` command, in “Tcl Commands” in the *Nios II Software Build Tools Reference* chapter of the *Nios II Software Developer’s Handbook*.

The missing information follows:

- supported_interrupt_apis—Specifies the interrupt API that the device driver supports. The Nios II SBT analyzes this property for each driver in the system to determine the appropriate API to be used in the system.

  Specify legacy_interrupt_api if the device driver supports the legacy API only or enhanced_interrupt_api if the device driver supports the enhanced API only. Specify both using a quoted list if the device driver supports both APIs.

  If you do not specify which API your device driver supports, the Nios II SBT assumes that only the legacy interrupt API is supported.

  ☞ This property is only available for device drivers.

  For more information about the legacy and enhanced APIs, refer to "Exception Handling" in the *Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook*.

- isr_preemption_supported—Specify true if your device driver's interrupt service routine (ISR) can be preempted by a higher priority ISR. If you do not specify whether the driver supports preemption, the Nios II SBT assumes that your device driver does not support preemption. If your driver does not have an ISR, but the associated device has an interrupt port, you can set this property to true.

  ☞ This property is valid for operating systems and device drivers.

Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

Valid Range of hal.log_flags is –1 to 3

The Nios II EDS documentation incorrectly states the valid range for the hal.log_flags BSP setting. Valid values of hal.log_flags range from –1 through 3.

Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

NicheStack TCP/IP Stack - Nios II Edition Does Not Support TFTP

The Nios II documentation should include the following information:

Versions of the NicheStack TCP/IP Stack other than the Nios II Edition include optional Trivial File Transfer Protocol (TFTP) client and server applications. However, the TFTP client and server are not thread safe. They cannot be used in systems with the MicroC/OS-II RTOS.

TFTP is not available with the NicheStack TCP/IP Stack - Nios II Edition. The TFTP client and server features are disabled.

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Error Message After Renaming Project: “Resource is out of sync with the system”

The *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer’s Handbook* does not explain the correct procedure for renaming a project.

Workaround

The missing information is as follows:

To rename a project in the Nios II SBT for Eclipse, execute the following steps:

1. Right click the BSP project and click **Rename**.
2. Type the new BSP name.
3. Right click the BSP project and click **Refresh**.

☞ If you neglect to refresh the project, you might see the following error message when you attempt to build it:

```
Resource <original_bsp_name> is out of sync with the system
```

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Obsolete Wiki URLs in Help Systems and Handbook

The following documents contain obsolete URLs for the Nios Community Wiki:

- The *Nios II Software Developer’s Handbook*
- The *Nios II Embedded Design Handbook*
- The Nios II IDE help system

Workaround

You can visit the Nios Community Wiki at www.nioswiki.com.

Solution Status

Fixed in the documentation for v. 9.0

Missing Source Control Information

The Quartus II Handbook refers to the *Using the Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook* for information about putting Nios II projects in source control. The referenced chapter only describes the process for Nios II SBT projects.

Workaround

The missing information follows.

Archiving Nios II IDE Software Projects

This section helps you identify the files you must include when archiving a Nios II IDE software project. With this information, you can archive a Nios II application project and its associated Nios II system library project.

You might want to archive your projects for one of the following reasons:

- To place them under source control
- To create backups
- To bundle the projects for transfer to another location

This section covers the following information:

- How to find and identify the files that you must include in an archived Nios II IDE software project.
- Which files must have write permission to allow the software projects to be built.

Table 6 and Table 7 list the files required by Nios II IDE software projects. This is the minimum set of files needed to completely rebuild a software project, including the **.elf**.

Archive your Nios II IDE software projects together with the SOPC Builder system on which they are based. You cannot rebuild a Nios II IDE software project without its associated SOPC Builder system.

The files listed in Table 6 are located in the Nios II application project directory.

Table 6. Files Required for a Nios II Application Project

File Description	File Name	Write Permission Required?
All source files	for example: **app.c**, **header.h**, **assembly.s**, **lookuptable.dat**	No
Eclipse project file	**.project**	No
C/C++ Development Toolkit project file	**.cdtproject**	Yes
C/C++ Development Toolkit option file	**.cdtbuild**	No
Software configuration file	**application.stf**	No

The files listed in Table 7 are located in the Nios II system library project directory.

Table 7. Files Required for a Nios II System Library Project

File Description	File Name	Write Permission Required?
Eclipse project file	**.project**	Yes
C/C++ Development Toolkit project file	**.cdtproject**	Yes
C/C++ Development Toolkit option file	**.cdtbuild**	No
System software configuration file	**system.stf**	Yes

☞ You must have write permission for certain files. The tools write to these files as part of the build process. If the files are not writable, the toolchain fails. Many source control tools mark local files read-only by default. In this case, you must override this behavior. You do not have to check the files out of source control unless you are modifying the Nios II software project.

Solution Status

In the documentation for v. 9.0, the missing information appears in the *Nios II Integrated Development Environment* chapter of the *Nios II Software Developer's Handbook.* In the documentation for v. 9.1, the missing information appears in the *Using the Nios II Integrated Development Environment* appendix of the *Nios II Software Developer's Handbook.*

Compiler Flags for Building Custom Newlib

"Common BSP Tasks" in the *Using the Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook* describes how to compile a custom version of the newlib library. However, it does not list the compiler flags that must be selected to generate the correct library.

Workaround

The missing information is as follows:

The Nios II EDS provides a number of precompiled newlib implementations. The provided libraries are precompiled with every viable combination of the GCC compiler flags shown in Table 8.

When you create a BSP with a precompiled newlib, the Nios II SBT selects the newlib matching your BSP's compiler settings. When you create a custom newlib, you must ensure that the compiler flags listed in Table 8 match your BSP's settings.

Table 8. GCC Compiler Options for Newlib

Option Name	Purpose
`-pg`	Link for profiling with gprof
`-EB`	Generate big-endian code
`-mstack-check`	Enable stack checking
`-mno-hw-mul`	Disable use of the `mul` family of instructions
`-mhw-mulx`	Enable use of the `mulx` family of instructions
`-mcustom-fpu-cfg`	Use a floating point custom instruction

☞ The Nios II-specific compiler flag `-mcustom-fpu-cfg` requires the Nios II processor core to be implemented with the floating-point custom instruction as specified in the flag argument. The compiler flag can have one of the following two arguments values:

- `-mcustom-fpu-cfg=60-1`—Use the Nios II floating-point custom instruction without divider
- `-mcustom-fpu-cfg=60-2`—Use the Nios II floating-point custom instruction with divider

For details about the Nios II-specific compiler flags `-mstack-check`, `-mno-hw-mul`, and `-mhw-mulx`, refer to “Altera Nios II Options” in *Using the GNU Compiler Collection (GCC)*, installed with the Nios II EDS. For details about the `-pg` compiler flag, refer to “Compiling a Program for Profiling” in *GNU Profiler*, installed with the Nios II EDS. To find the installed documentation, on the Windows Start menu, click **Programs** > **Altera** > **Nios II EDS <*version*>** > **Nios II <*version*> Documentation**, and then click **Literature**.

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Error Message on Invalid Exception or Reset Vector

Nios II compiler error messages, added in v 8.1, are not documented in the *Nios II Software Developer’s Handbook*.

Workaround

The missing information is as follows:

While building a Nios II software project, the compiler produces an error message if the SOPC Builder system defines an invalid exception or reset vector. Table 9 shows possible error messages and their causes.

In the SOPC Builder system, ensure that the exception and reset vectors conform to all the criteria listed in Table 9.

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Table 9. Exception and Reset Vector Error Messages (Part 1 of 2)

Message	Cause
`The section mapping ".entry" is not mapped to a memory region.`	The `.entry` section is mapped to a nonexistent memory region.
`The section mapping ".exceptions" is not mapped to a memory region.`	The `.exceptions` section is mapped to a nonexistent memory region.
`The section mapping ".entry" maps to a memory region "<`*`region name`*`>" which is not the required length "32".`	The `.entry` section is mapped to a memory region that is smaller than 32 bytes.
`The section mapping ".entry" maps to a memory region "<`*`region name`*`>" which does not start at the SOPC Design specified address "<`*`base address`*`>".`	The `.entry` section is mapped to a memory region that does not start at the reset vector base address.
`The section mapping ".exceptions" maps to a memory region "<`*`region name`*`>" which does not start at the SOPC Design specified address "<`*`base address`*`>".`	The `.exceptions` section is mapped to a memory region that does not start at the exception vector base address.
`The SOPC design Nios II reset and exception vector addresses are mapped to the same memory. For HAL the exception address must be at least 32 bytes larger than the reset address.`	The `.entry` section and `.exceptions` section map to the same memory device, and the base of the `.exceptions` section is less than 32 bytes above the base of the `.entry` section.

Table 9. Exception and Reset Vector Error Messages (Part 2 of 2)

Message	Cause
`The section mapping ".exceptions" mapped device "<memory 1>" is not the same as the SOPC Design specified device "<memory 2>".`	The `.exceptions` section is mapped to a device that is different from the exception vector device specified in the SOPC Builder system.
`The section mapping ".entry" mapped device "<memory 1>" is not the same as the SOPC Design specified device "<memory 2>".`	The `.entry` section is mapped to a device that is different from the reset vector device specified in the SOPC Builder system.

Nios II IDE Online Help Expand Buttons Do Not Work

The expand text arrow and the **Show All** option in the online help do not work in Internet Explorer.

Workaround

In Internet Explorer, carry out the following actions:

1. Click the **Refresh/Show Current Topic** icon (yellow arrows icon at the top of the search pane). The table of contents appears with the selected topic highlighted.
2. Click the highlighted topic to refresh the browser frame. Expanded text works.

Alternatively, use Firefox 2.0.0.6.

Solution Status

Not fixed

Hardware Abstraction Layer Errata

This section describes in detail the Nios II Hardware Abstraction Layer issues.

Missing Structure Member Errors in alt_log_printf.c with Small JTAG UART Driver

In a BSP, if you enable the small JTAG UART driver (`altera_avalon_jtag_uart_driver.enable_small_driver`), and enable Altera logging (`hal.log_port`) using the JTAG UART as the log port, when you build the project, the compiler reports missing structure members in **alt_log_printf.c**. The project fails to build.

Workaround

To avoid this issue, execute the following steps:

1. In function `alt_log_jtag_uart_print_control_reg()` in **alt_log_printf.c**, insert the italicized code as shown in Example 8.

Example 8. ALT_LOG_PRINTF() Correction

```
ALT_LOG_PRINTF(
    "%s SW CirBuf = %d, HW FIFO wspace=%d AC=%d WI=%d RI=%d WE=%d RE=%d\r\n",
    header,

/* Beginning of inserted code */
#ifndef ALTERA_AVALON_JTAG_UART_SMALL
    (dev->tx_out-dev->tx_in),
#else
    0,
#endif
/* End of inserted code */

    space,ac,wi,ri,we,re);
```

2. Rebuild the project.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

"unused variable" Warning When Building alt_main.c

If you set the `hal.enable_exit` BSP setting to `false`, when building the project you see the following warning in **alt_main.c**:

```
warning: unused variable 'result'
```

Design Impact

This warning is harmless.

Workaround

Set `hal.enable_exit` to `true`.

Alternatively, you can ignore this harmless warning.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Build Warnings in Performance Counter Driver Files

If your Nios II system contains a performance counter, you might see the following warning when you build a BSP:

```
warning: return makes pointer from integer without a cast
```

This warning appears in `alt_get_performance_counter_base()` in the software file **altera_avalon_performance_counter.c**.

This warning appears when the your BSP uses the Altera performance counter driver, and the C preprocessor symbol `PERFORMANCE_COUNTER_0_BASE` is defined.

This warning is harmless.

Workaround

In the `alt_get_performance_counter_base()` function in **altera_avalon_performance_counter.c**, add typecast `(void *)` to `PERFORMANCE_COUNTER_0_BASE`.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Hardware Example Design Errata

This section describes in detail the Nios II EDS hardware example design issues.

Incorrect PLL Phase Shift in Stratix II and Cyclone II Designs

In several example designs targeting the Nios II Cyclone II 2C35 and Nios II Stratix® II 2S60 RoHS development boards, the synchronous static random access memory (SSRAM) and legacy DDR SDRAM clocks are implemented with an incorrect phase-locked loop (PLL) phase shift.

This issue affects the following example designs:

- 2s60 RoHS Triple Speed Ethernet SG-DMA
- 2s60 RoHS Standard
- 2s60 RoHS Full Featured
- 2c35 Triple Speed Ethernet SG-DMA
- 2c35 Standard
- 2c35 Full Featured

This issue also affects any custom 2c35 or 2s60 RoHS design using SSRAM or legacy DDR SDRAM.

The incorrect phase shift might cause timing violations when reading from or writing to SSRAM or legacy DDR SDRAM.

Workaround

To correct the PLL settings, perform the following steps:

1. Open the standard design in SOPC Builder.
2. Double-click the component instance named `pll` to launch the PLL MegaWizard™ interface.
3. Click **Launch Altera's ALTPLL MegaWizard** to launch the MegaWizard interface.
4. Select the **Output Clocks** page and then the clock page used to generate the SSRAM clock. Depending on the particular design, this might be **clk c1**, **clk c2**, or another clock.

5. Ensure that **Clock phase shift** is –3.38 ns.
6. Click the clock page used to generate the write clock for the legacy DDR SDRAM, if any. Depending on the particular design, this might be **clk c1**, **clk c2**, or another clock.
7. Ensure that **Clock phase shift** is 270 degrees.
8. Click **Finish** to save changes and exit the PLL MegaWizard interface.
9. Click **Finish** to save settings to the PLL instance in SOPC Builder.
10. Regenerate the system in SOPC Builder and recompile in the Quartus II software. Refer to the **readme.txt** file in the design example folder for information on compiling designs containing the DDR controller.

For further information refer to **ssram_interface_readme.html**, located in the **<*Quartus II installation directory*>/sopc_builder/documents** folder. This document discusses the SSRAM timing analysis methodology in detail. Additionally, *AN 411: Understanding PLL Timing for Stratix II Devices* discusses clock phase shift calculations and assignments for PLLs in Stratix II devices.

Solution Status

Not fixed

Software Example Errata

This section describes in detail the Nios II EDS software example issues.

Hardware Tutorial Software Example Hangs on Some Boards

The **count binary** application used in the tutorial hangs when run on some Altera development boards. The software runs through the binary count once and stops responding.

This issue affects boards that do not have a display to handle the LCD output. The affected boards include the following:

- Embedded Systems Development Kit, Cyclone III Edition (EP3C120)
- Stratix IV GX FPGA Development Kit (EP4SGX230)

Workaround

Modify line 18 of the **count_binary.h** file, as follows:

```
# define LCD_PRINTF(lcd, args...) /* Do Nothing */
```

Rebuild and run the software again.

Solution Status

Not fixed

Simple Socket Server Example Displays Incorrect IP Address

The simple socket server networking software example displays an incorrect IP address for the incoming connection.

Workaround

There is currently no workaround for this problem. Ignore the IP address of any incoming connection that is displayed on the console.

Solution Status

Not fixed

Networking Examples

If you are running a networking software example, you might be asked for a nine-digit number. You are directed to find this number on a sticker on your Nios development board, identified by the prefix **ASJ**. Not all Nios development boards have this sticker.

Workaround

If your Nios development board does not have a sticker with the letters **ASJ** followed by a nine-digit number, enter a unique nine-digit number when prompted. To avoid network address conflicts, ensure that this number is unique to each Nios board connected to your network.

Solution Status

Not fixed

Flash Programmer Errata

This section describes in detail the Nios II EDS issues related to the flash programmer.

Unable to Configure FPGA from Flash with Parallel Flash Loader

If you program an FPGA image to flash memory on a board that utilizes the Parallel Flash Loader, the FPGA subsequently fails to load from flash memory. This issue arises because the **sof2flash** utility generates a flash file with an incompatible Programmer Object File (**.pof**) bitstream format.

Affected Configurations

This issue impacts the following target hardware:

- The Stratix IV GX FPGA Development Kit
- The Arria® II GX FPGA Development Kit
- Any hardware using the Parallel Flash Loader

Workaround

1. Visit the MySupport website and request Nios II EDS patch 0.73.

   Alternatively, upgrade to the Nios II EDS v. 9.1 SP1 or later.

2. Invoke **sof2flash** with the following options:

   - `--pfl`
   - `--optionsbits=<option bit address>`

   With these options, **sof2flash** generates a flash file with a compatible **.pof** file bitstream format.

Solution Status

Fixed in v. 9.1 SP1 of the Nios II EDS

Boot Failure or elf2flash Error Due to Incorrect Flash File Size

The `sof2flash` utility fails to recognize certain devices. As a result, it generates Motorola S-record (**.flash**) files of the wrong size.

With erasable programmable configurable serial (EPCS) memory, the `elf2flash --after` command fails to recognize the **.flash** file size. The resulting error prevents you from programming flash memory.

With Common Flash Interface (CFI) memory, you can program the **.flash** file, but the Nios II system fails to boot.

This error occurs when programming flash memory for the following devices:

- The following Stratix III devices:
  - EP3SL50
  - EP3SL50F484
  - EP3SL50F780
  - EP3SL110
  - EP3SL110F1152
  - EP3SL110F780
  - EP3SE80
  - EP3SE80F1152
  - EP3SE80F780
- All Stratix IV devices
- All Arria II GX devices

Workaround

Visit the MySupport website and request Nios II EDS 9.0 SP2 patch 2.01 and Quartus II 9.0 SP2 patch 2.63.

Alternatively, download and install the Altera Complete Design Suite v. 9.1 or later.

Solution Status

Fixed in v. 9.1 of the Nios II EDS

Delay When Creating New Flash Programmer Configuration

You might experience a delay of several seconds when creating a new Flash Programmer configuration in the Nios II IDE.

Workaround

There is no workaround.

Solution Status

Not fixed

elf2flash File Size Limit

The **elf2flash** utility supports **.elf** files up to approximately 24 MBytes in size. The **elf2flash** utility might fail on files larger than 24 MBytes, with the error message `java.lang.OutOfMemoryError`.

Workaround

Lower the number of symbols in your **.elf** file by turning off debug symbols.

Alternatively, specify less initialized data in the application.

Solution Status

Not fixed

Error Message “No such file or directory” When Programming Flash

When programming flash from a project stored in a path containing spaces, you might receive the spurious error message **No such file or directory**. The flash programmer does not correctly handle spaces in the directory path. However, this error is harmless, because flash programming completes successfully.

Workaround

None required.

Solution Status

Not fixed

Hardware Simulation Errata

This section describes in detail the Nios II EDS issues related to hardware simulation.

Vectored Interrupt Controller Does Not Support VHDL Simulation Models

SOPC Builder reports errors when you try to generate VHDL simulation files for the Vectored Interrupt Controller.

Workaround

There is no workaround.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Error “UNC paths are not supported” Launching ModelSim

If you launch ModelSim® from a working directory that is mapped via a universal naming convention (UNC) path (a path that starts with `//` instead of drive letter), you receive the following error message in SOPC Builder: **UNC paths are not supported. Defaulting to Windows directory**. This error occurs because ModelSim is calling a command shell, which does not support UNC paths.

Workaround

Map the UNC path to a drive letter and use the drive letter to reference the working directory in the launching shell.

Solution Status

Not fixed

Uninitialized .bss Variables in Simulation

If your program reads the value of an uninitialized `.bss` variable during HDL simulation, and the BSP (system library) is compiled with the **ModelSim only, no hardware support** property enabled in Nios II IDE, a warning appears about unfiltered data being `'x'`. This warning appears because when this property is enabled, the code that clears the `.bss` memory region is omitted to speed up HDL simulation so this memory region is uninitialized. The `.bss` region contains global and static local variables that are not initialized by the application so they default to a value of zero. When the Nios II processor reads uninitialized variables, it displays a warning and converts any of the bits of the uninitialized data to zero which correctly mimics the effect of the missing `.bss` clearing code. The Hardware Abstraction Layer (HAL) code that executes before and after `main()` might use `.bss` variables, so these warnings might appear even if your application does not use the `.bss` section.

Solution Status

Not fixed

Host Platform Errata

This section describes in detail the Nios II EDS issues related to the host development platform.

Nios II EDS Requires Administrative Permissions on Windows Vista

On the Windows Vista operating system, the Nios II EDS requires your user account to have administrative permissions.

Attempts to write to a directory other than the user home directory might fail, especially writing to an installation directory. For example, you might be unable to create a software project under an example design directory. Error messages might vary.

Workaround

Obtain an administrator account on your workstation.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Cannot Launch Nios II SBT for Eclipse on Red Hat Linux

You might be unable to launch Nios II SBT for Eclipse on the Red Hat Linux operating system. This is an issue with the version of XULRunner in Eclipse 3.4. The issue is resolved in XULRunner 1.9.1, which is available with Eclipse 3.5.

Workaround

Upgrade XULRunner to a newer version. To determine what version of XULRunner you have, type the following command at the command prompt:

```
xulrunner -v↵
```

To upgrade XULRunner using the **yum** software package manager, type the following command:

```
sudo yum update xulrunner↵
```

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Windows/Cygwin: Nios II Processor Generation Failure

You might see the following error message when you generate your Nios II system in SOPC Builder:

```
Error: Generator program for module 'cpu_fpoint' did NOT run
successfully
```

This error might occur on a Windows-based system when all of the following conditions are true:

- You have installed a version of Cygwin other than the one distributed with the Quartus II software.
- You launch the Quartus II software and SOPC Builder from a Nios II Command Shell.
- You enable the floating point custom instruction (FPCI) in the Nios II processor core.

Workaround

Launch the Quartus II software and SOPC Builder from the Windows Start menu, and regenerate your system.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Windows Vista: Limited Support in Nios II EDS

The Quartus II software introduces Windows Vista (32-bit and 64-bit) support in version 7.2. However, the Nios II Embedded Design Suite supports only the following platforms:

- Windows 2000
- Windows XP
- Windows XP (64)
- Windows Vista Business Edition (32-bit)
- SUSE 9 (32-bit)
- SUSE 9 (64-bit)
- Red Hat Linux v3.0 (32-bit)
- Red Hat Linux v3.0 (64-bit)
- Red Hat Linux v4.0 (32-bit)
- Red Hat Linux v4.0 (64-bit)

Workaround

There is no workaround.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Nios II IDE Errata: Building Projects

This section describes in detail the issues related to building projects in the Nios II IDE.

Nios II IDE Command-Line Tools Select Wrong Workspace on Linux

On Linux systems, the Nios II IDE command-line tools use the default Eclipse project workspace, regardless what workspace you are using in the IDE. The default workspace is at `$SOPC_KIT_NIOS2/eclipse/nios2-ide-workspace-<`*`version`*`>`.

This issue affects the following tools:

- `nios2-create-system-library`
- `nios2-create-application-project`
- `nios2-build-project`
- `nios2-import-project`
- `nios2-delete-project`

Workaround

Explicitly supply the Nios II IDE command-line tools with a workspace location, by means of the `-data` command-line argument. The syntax of the `-data` argument is as follows:

```
-data <path to workspace>
```

The path to the workspace must be absolute, and must not contain whitespace. Aside from these restrictions, any valid file system path can be used for the workspace. Example 9 shows how to specify a workspace path.

Example 9. Providing a Non-Default Workspace Location in Linux

```
nios2-create-project -data $HOME/myworkspace <other arguments>↵
```

Solution Status

Not fixed

Nios II IDE Command-Line Tools Hang on Windows

On Windows systems, the Nios II IDE command-line tools are sometimes unable to locate the project workspace. When this happens, the tools might hang.

This issue potentially affects the following tools:

- `nios2-create-system-library`
- `nios2-create-application-project`
- `nios2-build-project`
- `nios2-import-project`
- `nios2-delete-project`

Workaround

Explicitly supply the Nios II IDE command-line tools with a workspace location, by means of the `-data` command-line argument. The syntax of the `-data` argument is as follows:

```
-data <path to workspace>
```

The path to the workspace must be absolute, and must not contain whitespace. Aside from these restrictions, any valid file system path can be used for the workspace. The workspace shown in Example 10 is the default workspace used by the Nios II IDE in version 9.0.

Example 10. Providing a Workspace Location

```
nios2-create-project -data c:/altera/90/nios2eds/\
    bin/eclipse/nios2-ide-workspace-9.0 <other arguments>
```

Solution Status

Not fixed

Nios II IDE Cannot Find stdio.h in Outline View

If you create a new project using the Nios II IDE and try to open the **stdio.h** file from the Outline view before building the project, the IDE displays the error message **No include files were found that matched that name**.

Workaround

Build the project before attempting to open **stdio.h**.

Solution Status

Not fixed

Nios II IDE Hangs With UNC Project Path

In the **New Project** dialog box, if you turn on **Specify Location** and specify the path in UNC form, the IDE might hang.

Workaround

Map the UNC path to a remote drive which looks like a Windows drive to the Nios II IDE.

Solution Status

Not fixed

Build Failure with Nios II Advanced Exceptions, MMU, and MPU

Projects created in the Nios II IDE for an SOPC Builder system containing a Nios II processor configured with advanced exceptions, the memory management unit (MMU), or the memory protection unit (MPU) fail to build, and generate the following error messages:

```
ERROR - Classic build flow for Nios II system library does not support
the Nios II \
    advanced exceptions.
Use the non-classic Nios II Board Support Package instead.
<timestamp> - (SEVERE) generate: java.lang.IllegalStateException: \
    java.lang.IllegalStateException: \
    com.altera.ingenuous.GTFElement.GTFElementErrorException: \
    <error> element in GTF script
make[1]: *** [system_description/../obj/system.h-t] Error 1
make: *** [system_project] Error 2
```

Workaround

Create the software project with the Nios II SBT command-line development flow. Alternatively, use the Nios II SBT for Eclipse.

Solution Status

Not fixed

Build Command Not Functional for BSPs Created With the Nios II SBT

The build option in the Nios II IDE menu does not rebuild BSPs imported to the IDE.

Workaround

The Nios II SBT for Eclipse can both build and debug projects created on the command line. The Nios II SBT for Eclipse is the preferred tool for debugging Nios II SBT projects.

For information about the Nios II SBT for Eclipse, refer to the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*.

In the Nios II IDE, you can build the BSP by building the associated application project.

Solution Status

Not fixed

Linker Errors with Dual-Port Memories

If your instruction master and data master ports are connected to the same dual-port memory and the ports have different addresses, your code fails to run or you experience a linker error. The Nios II IDE does not warn you of the addressing violation.

Workaround

Assign the same address to both ports of the dual-port memory.

Solution Status

Not fixed

User-Managed BSP Settings Not Supported in Nios II IDE

For projects created with the Nios II SBT and imported to the Nios II IDE, the IDE configuration settings have no effect.

For example, objdump, compiler, and linker settings made in the IDE are ignored. This behavior occurs because Nios II SBT projects are not IDE-managed projects. In addition, the make-related preferences do not pertain to imported Nios II SBT projects. The IDE ignores these options during the build process.

Workaround

Make these settings in the project's makefile.

Solution Status

Not fixed

URL Project Location Causes Project Creation Error

When you try to create a new project in an existing workspace, you might see an error dialog box saying:

```
Project cannot be created. Reason: Internal Error
```

This error might occur if the path to any project in the workspace is a URL location, for example **file:/F:/Design**. To view the path, right-click on the project and select **Properties**.

Workaround

Import your existing application and system library projects to a new workspace.

Solution Status

Not fixed

Compilation Error with Separate Exception Stack Option

Choosing the **Use a separate exception stack option** might cause the following compilation error when building a project:

```
VARIABLE %STACK_POINTER%"
```

This error occurs if the exception stack is larger than the memory available for it.

Workaround

On the system library properties page for the project, turn off the separate exception stack or reduce the **Maximum exception stack size** setting.

Solution Status

Not fixed

Incorrect Stack and Heap Free Space Report

The makefile reports an incorrect number of bytes free for the stack and heap, if the heap and stack are in different memory regions.

Workaround

No workaround available.

Solution Status

Not fixed

Nios II IDE Reports Problems Without Displaying Error in Console

When building a project, the Nios II IDE reports problems, but the build output in the console does not contain any errors.

The Nios II IDE incorrectly reports some linker warnings as errors, with a dialog box saying **Errors exist in a required project**. The Dhrystone software example exhibits this behavior, and recompiling the project again makes the issue go away.

Workaround

If the Console output does not contain errors, then the project built correctly. On subsequent builds, the linker step is skipped and the errors do not appear.

Solution Status

Not fixed

Nios II IDE Errata: Debugging Projects

This section describes in detail the issues related to debugging projects in the Nios II IDE.

Method for Accessing MMU and MPU Registers in the Debugger

You might observe an error if you try to read or write an MMU or MPU register via the Nios II Debugger.

Workaround

To read an MPU region, execute the following steps:

1. Set region `INDEX` in the `MPUBASE` register.
2. Exit and reenter debug mode, that is, single step.
3. Set the `RD` bit in the `MPUACC` register.
4. Exit and reenter debug mode, that is, single step.
5. Read back `MPUBASE` for pertinent information.
6. Read back `MPUACC` for pertinent information.

For a system with an MMU or MPU, this workaround allows you to read and write the current values of the registers. However, you cannot use it to control MPU regions or MMU TLB entries.

The debug core copies the Nios II processor's register values to its internal memory when the processor enters debug mode. The debug core writes register values back to the processor only when the processor leaves debug mode. Therefore, if you attempt to set an MPU region with several consecutive values, only the last one, when leaving debug mode, is committed to the processor.

Solution Status

Not fixed

IDE Cannot Display Imported Profiling Data

If you create a software project in the Nios II SBT command line flow, generate profiling data to **gmon.out** in the Nios II command shell, and then import the project to the IDE, you cannot use the Profiling perspective to view the profiling data. When you attempt to view **gmon.out**, the IDE displays the following error message:

```
nios2-elf-gprof: ../: not in a.out format.
```

Workaround

Run `nios2-elf-gprof` from the Nios II command shell to generate a profiler report.

Solution Status

Not fixed

Error Message when Downloading .elf File

If the **Build Automatically** option is turned on in the IDE, you might get the following error when downloading a **.elf**:

```
"2 [main] sh 5736 fork: child 3892 - died waiting; \
    for longjmp before initialization, errno 11"
```

The **Build Automatically** option is off by default, as recommended by the Eclipse help system.

Workaround

Turn off the **Build Automatically** option and download the **.elf** file again.

Solution Status

Not fixed

Variable Casting Unsupported in ISS

Casting variables in the Variables view in the Debug perspective when using the Nios II ISS might cause an exception dialog box to open.

This exception might occur if you select **Debug as ISS** in the Nios II IDE and try to cast variables via the Variables view.

Workaround

There is no workaround.

Solution Status

Not fixed

Trace Debug Does Not Support Instruction-Related Exceptions

The instruction-related exception handler is not supported by the Nios II trace tools.

Workaround

There is no workaround.

Solution Status

Not fixed

Trace Debug Does Not Support the JMPI Instruction

The JMPI instruction is not supported by the Nios II trace tools.

Workaround

There is no workaround.

Solution Status

Not fixed

Cannot Locate Source Code in Driver Files Shared by Multiple Projects

If you hit a breakpoint in a driver file, and that driver file is shared with another project that is closed, the Nios II IDE might indicate that it cannot locate the source code.

Workaround

Open the closed system library project and resume debugging.

Solution Status

Not fixed

Console Window Is Not Updated After ISS Error

After performing a **Run as ISS**, if you receive an ISS error in the console window, the console is not updated subsequently.

Workaround

Close the console window after receiving an ISS error. A new console window opens when a new message is available.

Solution Status

Not fixed

ISS Fails on Designs Containing Triple Speed Ethernet MAC or SG-DMA Components

You receive an Internal Error when attempting to perform an ISS simulation of designs containing the Altera Triple Speed Ethernet media access control (MAC) or SG-DMA components because the Nios II ISS does not support these components.

Workaround

Remove the Triple Speed Ethernet MAC and SG-DMA components from your system and perform ISS simulation on the simplified system. You can also simulate the design in ModelSim or test it on hardware.

☞ Before removing the Triple Speed Ethernet MAC and SG-DMA components, make a copy of the unmodified system to ensure that you can return to the original configuration.

Solution Status

Not fixed

Memory Window Sets Control Register Values Incorrectly

The memory window might incorrectly set values in memory-mapped control registers. For example, writing 0x1234 to a byte addressed register results in the value 0x3434 being stored in the register. The memory window shows this incorrect value.

Workaround

Use the GNU debugger (GDB) console window in the IDE, instead of the memory window, to write to the registers. For example, type the following command:

```
set {int} <register address>=0x1234↵
```

☞ You must refresh the memory window for it to correctly display the target value.

Solution Status

Not fixed

Programs That Interact With a Terminal Console on Windows Do Not Work

Programs with this behavior work in v. 6.0 and earlier, but do not work in Nios II IDE v. 6.1 and later.

The Eclipse platform in v. 6.1 and later of the IDE (on Windows only) sends the string `\r\n` instead of just `\n` when running or debugging using the Terminal. This behavior can break existing software designs, and it is inconsistent with `nios2-terminal`, which still just sends `\n`.

Workaround

Change the software running on the Nios II processor to parse for `\r\n` as well as `\n`.

Solution Status

Not fixed

"Run as ModelSim" in the Nios II IDE Fails

The **Run as ModelSim** command might fail on launch configurations created in v. 7.0 or earlier of the IDE. This problem does not occur for new launch configurations.

Workaround

Select a location for the ModelSim tool from the launch configuration dialog box. You can use the **Browse** button next to the **ModelSim path** group, or type in a path to the ModelSim directory (for example **c:/altera/71/modelsim_ae/win32aloem**).

Solution Status

Not fixed

The Restart Command on the Run Menu Does Not Work

The **Restart** command on the **Run** menu does not work.

Workaround

Stop the program, then debug it again. If the debugger is hung in an endless loop, use the following **bash** alias to break the target, then stop it:

```
alias break="kill -2 \'ps -a | grep nios2-elf-gdb | cut -f6 -d' '\'"
```

Solution Status

Not fixed

Watchpoints Do Not Work on Certain Variables

Watchpoints do not work on a variable whose size is not 32 bits.

Workaround

Change the types of global and static local variables to `int`, `long`, or `unsigned long` before setting watchpoints on them.

Solution Status

Not fixed

Nios II IDE Errata: Navigating Projects

This section describes in detail the issues related to navigating projects in the Nios II IDE.

Nios II IDE Freezes While Displaying the Splash Screen

After clicking **Switch Workspace** on the File menu on a Windows machine, a Nios II IDE splash screen appears. Unfortunately, this splash screen obscures the dialog box that asks you to specify the new workspace. As a result, the IDE appears to freeze.

Workaround

Press Alt-Tab to switch applications. Two relevant application icons appear: an Eclipse icon associated with the splash screen and a Nios II IDE icon associated with the **Workspace** dialog box. Select the Nios II icon to bring the dialog box to the foreground.

Solution Status

Not fixed

Internal Error When Double-Clicking on a Large Objdump File

On Windows when opening a large **objdump** file in the Nios II IDE, you might get the following error message:

```
Unable to create this part due to an internal error.
Reason for the failure: Editor could not be initialized.
```

Workaround

Adjust the Windows launch arguments for the Nios II IDE editor. Perform the following steps:

1. On the Windows Start menu, browse to the **Nios II EDS** program icon, right-click it, then click **Properties**. The **Windows Properties** dialog box appears.
2. In the **Target** field, append `"vmargs -Xmx1024m"` to the end of the path to the Nios II IDE executable. For example:

   ```
   C:\altera\72\nios2eds\bin\eclipse\nios2-ide.exe -vmargs -Xmx1024m
   ```

Solution Status

Not fixed

C/C++ Scanner Does Not Support Certain C/C++ Constructs

The C/C++ scanner performs C/C++ Search, navigation, open declaration, and parts of content assist. Due to limitations of the C/C++ scanner, these features do not work with C code constructs not supported by the C++ language. An example is functions that take a function pointer as an argument.

Workaround

If the C/C++ Search fails, use the File Search facility.

Solution Status

Not fixed

Nios II SBT Errata

This section describes in detail the issues in the Nios II SBT.

Error Running Nios II Project: 'Downloading ELF Process failed'

If the Nios II processor's cpu.data_master port is not connected to all program memories (memories to which the **.elf** file is downloaded) the software project fails to run on Nios II hardware.

Failure to connect cpu.data_master to all program memories is a design error that the Nios II SBT does not detect.

Workaround

Connect cpu.data_master to all program memories.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

BSP Editor Does Not Show Command-Line Help

The following commands in the Nios II Command Shell should display command-line help:

- `nios2-bsp-editor --help`
- `nios2-bsp-editor --extended-help`

Instead, these commands launch the BSP Editor.

Workaround

For help with the BSP Editor, refer to "Using the BSP Editor" in the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

BSP Not Updated for Memory Size Changes in SOPC Builder

If you change the size of a memory in SOPC Builder, the memory region size in any previously created BSP is no longer correct. Regenerating the BSP does not update the BSP's memory region size.

Workaround

In the BSP Editor, on the **Linker Script** tab, you can correct the memory region size either of the following ways:

- Edit the memory region size manually.
- Click **Restore Defaults** to rerun the default Tcl script. The default Tcl script reads the updated memory region size from the SOPC information file (**.sopcinfo**) and updates the BSP.

For detailed information about keeping your BSP consistent with changes in the underlying SOPC Builder system, refer to "Revising Your BSP" in the *Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook*.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Build Errors on Interrupt API Calls

You might observe build errors if your application project installs an ISR that calls the `alt_irq_register()` function.

The Nios II EDS v9.1 includes support for an optional External Interrupt Controller (EIC) interface on the Nios II processor. To support the EIC interface, the HAL includes an enhanced interrupt API.

When you create a BSP, the SBT determines which interrupt API(s) the device drivers in your system support. If all registered drivers in the BSP identify themselves as supporting the enhanced API, the SBT implements the enhanced API. All Altera device drivers in v9.1 and later support the enhanced API.

When the enhanced API is implemented, the legacy interrupt API, including `alt_irq_register()`, is not available. Therefore, if the application code contains a call to a legacy API function, a linker error occurs.

This issue might affect application projects that call `alt_irq_register()` or other legacy interrupt API functions directly, rather than depending on drivers in the BSP for interrupt support.

If your Nios II application project's source code registers an ISR using `alt_irq_register()`, or includes an ISR, your application might fail to compile in the Nios II EDS v. 9.1, due to undefined references to `alt_irq_register()`, or to a change to the ISR function prototype.

Workaround

Modify your application code to use the new enhanced interrupt API. To use the enhanced API, you must modify several function calls and all ISR function prototypes.

For information about supporting the enhanced interrupt API, refer to the *Exception Handling* chapter of the *Nios II Software Developer's Handbook*, or to *AN 595: Vectored Interrupt Controller Usage and Applications*.

You must upgrade to the enhanced API if you wish to use an EIC in your hardware design to accelerate interrupt response time. The enhanced API also works with the Nios II processor's internal interrupt controller.

Solution Status

Not fixed

BSP Editor Incorrectly Allows Mixed Drivers

If multiple drivers are found for a single peripheral type, and you have multiple instances of the peripheral type in your SOPC Builder system, the Nios II BSP editor allows you to manually select a different driver for each instance of the peripheral or processor. For example, if you have two `altera_avalon_timer` devices, you can select two different timer drivers.

This behavior is not fully supported. You must choose the same driver for both instances of the peripheral type.

You can make driver assignments in one of the following ways:

- In the Nios II BSP editor, within the **Drivers** tab. The **Driver name** associated with each peripheral can be edited, if multiple driver types are found that support the peripheral. You must make the same driver type selection for each instance of a peripheral type in the list of peripherals as defined in the **Module Name** column.
- With the Nios II SBT in command-line mode (`nios2-bsp`, `nios2-bsp-create-settings`, `nios2-bsp-update-settings`): the **set_driver** command can be invoked (either from a Tcl script or with the `--cmd` option). The **set_driver** command accepts a peripheral module instance name and driver name as arguments. If you are using the command line development flow and wish to use this command, you must invoke it once for each instance of a peripheral type in your system, similar to the BSP editor flow described in the previous paragraph.

Workaround

There is no workaround.

Solution Status

Fixed in v. 9.1 of the Nios II EDS

nios2-bsp --help Displays Incorrect List of BSP Types

The Nios II SBT supports custom HAL-based operating systems, when they are added as Tcl-scripted software packages. However, the `nios2-bsp` tool help message indicates that there are only two BSP types, HAL and UCOSII. If other HAL-based operating system components are added they do not appear in the `nios2-bsp` help message.

Workaround

To get a correct, dynamic list of available BSP types, type the following command:

```
nios2-bsp-create-settings --type x --settings x --sopc=<existing sopcinfo path>↵
```

This command returns an error message that lists available BSP types:

```
SEVERE: Available BSP type values for the --type argument are: [<list of BSP types>]
```

Solution Status

Fixed in v. 9.1 of the Nios II EDS

Java Heap Space Error When Creating BSP

When creating a BSP, the Nios II SBT might hang with the following message:

`SEVERE: OutOfMemoryError: Java Heap Space`.

This problem occurs when your SOPC Builder design has connection cycles. A connection cycle consists of a group of components that master one another in a circular arrangement. For example, if `ddr_sdram_bridge` masters `lcd_sgdma`, and `lcd_sgdma` masters `ddr_sdram_bridge`, the two components form a simple connection cycle. In this situation, the SBT is unable to create a BSP.

Workaround

Avoid connection cycles in the SOPC Builder system.

Solution Status

Fixed in v. 9.0 of the Nios II EDS

SBT Fails if Nios II EDS is Installed in a Path Containing Spaces

The Nios II Command Line SBT fails if you install the Nios II EDS in a path containing spaces.

Workaround

Reinstall the Nios II EDS to a path that does not contain spaces.

Solution Status

Not fixed

User-Managed BSP Settings Not Supported in Nios II IDE

This issue is described on page 47.

Upper-Case File Extensions Not Supported

The Nios II SBT for applications and libraries (`nios2-app-generate-makefile` and `nios2-lib-generate-makefile` commands) does not support source files with certain upper-case extensions. If a file with an upper-case extension is included, the make command stops with no descriptive warning.

Only Nios II assembly language files built by the C preprocessor can have upper-case file extension (**.S**). All C language files must have the extension **.c** or **.h**. C++ language source files must have the extension **.cpp**, **.cxx**, **.cc,** or **.h**.

Workaround

Rename all C language files with the extension **.c** or **.h**. Rename all C++ language files with the extension **.cpp**, **.cxx**, **.cc,** or **.h**.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Peripheral Errata

This section describes in detail the Nios II EDS issues related to peripheral components.

Vectored Interrupt Controller Does Not Support VHDL Simulation Models

This issue is described on page 41.

Unaligned Transfers of Small Payloads Fail on SG-DMA

The Scatter Gather DMA SOPC Builder peripheral does not correctly handle unaligned transfers with small payloads. A payload length smaller than the data width causes erroneous data transfers.

Workaround

Avoid using DMA devices to transfer small payloads.

If absolutely necessary, for a 32-bit SG-DMA, a minimum length of 4 bytes guarantees that data is transferred correctly.

Solution Status

Not fixed

DMA Controller Always Busy in Burst Mode

The DMA controller component (`altera_avalon_dma`), when enabled for burst transactions, does not perform transfers at widths other than its full data width. The DMA controller is always busy.

Workaround

When bursting is enabled, the DMA controller must be programmed to perform transactions at its full data width.

Solution Status

Not fixed

Non-System-Wide Reset Can Cause Improper Initialization of Mailbox Core

The `altera_avalon_mailbox` peripheral might not be initialized properly when a soft (non-system-wide) reset occurs. In this condition, mailbox contents (read and write pointers) are not reinitialized and might show potentially stale data.

Workaround

Ensure that a system-wide reset event occurs by asserting the `reset_n` input to the SOPC Builder system containing the mailbox. This resets all peripherals and Nios II processors in the system.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Target Software Errata

This section describes in detail the Nios II EDS issues related to target software packages.

NicheStack tcp_wake() Function Does Not Awaken All select() Sleepers

This issue impacts software based on the NicheStack TCP/IP Stack - Nios II Edition, with multiple sleeping tasks that depend on the same `select()` event to wake up. The `tcp_sleep()` and `tcp_wake()` functions do not work properly for tasks that call `select()`.

Tasks can normally use `tcp_sleep()` and `tcp_wake()` to suspend and then awaken themselves upon specific events. The `select()` function passes an event pointer to `tcp_sleep()` that is identical for every task that might call `select()`. All other events that call `tcp_sleep()` pass it a unique event pointer that specifically identifies them as the task to be awakened at the next event.

The `tcp_wake()` function apparently assumes that the list of sleepers contains unique event pointers by which they can be identified. Therefore, when a `select()` event occurs, instead of awakening all tasks waiting for it, `tcp_wake()` awakens only the first task in the list. Therefore, when more than one task calls `select()` to await some outcome on a socket, only the highest-priority task in the sleepers list is awakened at `select()` events. Other sleeping tasks that depend on the same `select()` event do not wake up.

Workaround

There is no workaround.

Solution Status

This issue will be fixed in future release of the Nios II EDS.

EIC-Based Nios II System Crashes With NicheStack

When you attempt to run software based on the NicheStack TCP/IP Stack - Nios II Edition on a hardware design incorporating an external interrupt controller such as the VIC, the Nios II system crashes during initialization, with unpredictable results.

Workaround

There is no workaround.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Gigabit Ethernet Performance Issues

The NicheStack TCP/IP Stack - Nios II Edition might hang while sending transmission control protocol (TCP) packets to the host, or experience dropped packets when receiving TCP packets from the host. These issues can arise with version 3.1 of the NicheStack TCP/IP Stack, using the triple-speed Ethernet component to transmit over a Gigabit Ethernet point-to-point connection.

Workaround

Download and install the patches at www.altera.com/support/kdb/solutions/rd01132009_588.html.

Alternatively, upgrade to the Nios II EDS v. 9.0 or later.

Solution Status

Fixed in v. 9.0 of the Nios II EDS

stdio Does Not Work with MicroC/OS-II and Small C Library

`stdin`, `stdout`, and `stderr` do not work in MicroC/OS-II applications built with the Small C library option.

Workaround

Disable the small C library option.

Solution Status

Not fixed

cout From MicroC/OS-II Task Does Not Send Data to stdout

If neither `printf()` or `cout` is used from `main()` before tasks are started, `cout` does not work from a task.

Workaround

Add the following C++ code to the beginning of `main()`:

```
std::ios_base::sync_with_stdio(false);
```

Solution Status

Not fixed

malloc(), realloc() Failures With MicroC/OS-II

When you use the MicroC/OS-II RTOS, calls to `malloc()` and `realloc()` might fail if successive calls to `malloc()` or `realloc()` within a MicroC/OS-II task occur after changing the task priority of the task in which a memory block is originally allocated.

Workaround

Use one of the following workarounds:

- Allocate and/or reallocate memory blocks outside of MicroC/OS-II tasks, before task switching starts. Preallocated memory blocks make it possible to change thread priorities at runtime.
- Allocate fixed areas of memory using arrays (rather than using `malloc()`) before task switching starts. Fixed memory arrays make it possible to change thread priorities at runtime.
- Allocate memory using `malloc()` or `realloc()` from a MicroC/OS-II task. You can change task priorities at runtime, but only for tasks that have not used `malloc()` or `realloc()`.

Solution Status

Not fixed

Toolchain Errata

This section describes in detail the Nios II EDS issues related to the Nios II and GNU compiler toolchains, including tools such as **gcc**, **gdb**, and **sof2flash**.

C++ Exceptions Unsupported in a Multi-Threaded Environment

C++ exceptions are only supported in a single-threaded environment. They cannot be used in a multi-threaded environment such as MicroC/OS-II.

Also see "C++ Exceptions Unsupported by the Nios II GCC 4 Toolchain".

Workaround

In a multi-threaded environment, you need to guard C++ exceptions with semaphores.

Solution Status

Not fixed

C++ Exceptions Unsupported by the Nios II GCC 4 Toolchain

With the GCC 4 toolchain, C++ exceptions are not supported.

Also see "C++ Exceptions Unsupported in a Multi-Threaded Environment".

Workaround

If your single-threaded project requires C++ exceptions, use the GCC 3 toolchain.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

memcpy() Optimization Misalignment

When optimization is turned on (`-O1` or higher), if you use `memcpy()` and the source pointer is aligned to a 32-bit boundary, the compiler implements `memcpy()` with word-oriented instructions as part of the optimization process. This optimization technique causes unexpected results in your software if `memcpy()` is used on a misaligned address.

Workaround

Take steps to ensure that an optimized implementation of `memcpy()` is called only with aligned data pointers.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Unable to Configure FPGA from Flash with Parallel Flash Loader

This issue is described on page 39.

GNU Assembler Does Not Accept the --defsym Flag

According the GNU documentation, you can set an assembler definition by using the `--defsym` flag, but it does not work in the following form: `--defsym MY_VAR=1`.

Workaround

There is no workaround.

Solution Status

Not fixed

How to Contact Altera

For the most up-to-date information about Altera products, refer to Table 10.

Table 10. Contact Information

Contact *(1)*	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Altera literature services	Email	literature@altera.com
Non-technical support (General)	Email	nacomp@altera.com
(Software Licensing)	Email	authorization@altera.com

Note:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

Table 11 shows the typographic conventions that this document uses.

Table 11. Typographic Conventions

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Indicates command names, dialog box titles, dialog box options, and other graphical user interface (GUI) labels. For example, **Save As** dialog box. For GUI elements, capitalization matches the GUI.
bold type	Indicates directory names, project names, disk drive names, file names, file name extensions, dialog box options, software utility names, and other GUI labels. For example, **\qdesigns** directory, **d:** drive, and **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Indicates document titles. For example, *AN 519: Stratix IV Design Guidelines.*
Italic type	Indicates variables. For example, *n* + 1. Variable names are enclosed in angle brackets (< >). For example, <*file name*> and <*project name*>**.pof** file.
Initial Capital Letters	Indicates keyboard keys and menu names. For example, Delete key and the Options menu.
"Subheading Title"	Quotation marks indicate references to sections within a document and titles of Quartus II Help topics. For example, "Typographic Conventions."
`Courier type`	Indicates signal, port, register, bit, block, and primitive names. For example, `data1`, `tdi`, and `input`. Active-low signals are denoted by suffix `n`. For example, `resetn`. Indicates command line commands and anything that must be typed exactly as it appears. For example, `c:\qdesigns\tutorial\chiptrip.gdf`. Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword `SUBDESIGN`), and logic function names (for example, `TRI`).
1., 2., 3., and a., b., c., and so on.	Numbered steps indicate a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ■	Bullets indicate a list of items when the sequence of the items is not important.
☞	The hand points to information that requires special attention.
⚠ CAUTION	A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.
⚠ WARNING	A warning calls attention to a condition or possible situation that can cause you injury.
↵	The angled arrow instructs you to press Enter.
👣	The feet direct you to more information about a particular topic.

101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support
www.altera.com/support

Nios II Embedded Design Suite Release Notes and Errata

© May 2010 RN-EDS-6.1

About These Release Notes

These release notes cover versions 8.1 through 9.1 SP2 of the Altera® Nios® II Embedded Design Suite (EDS). These release notes describe the revision history and errata for the Nios II EDS.

Errata are functional defects or errors, which might cause the product to deviate from published specifications. Documentation issues include errors, unclear descriptions, or omissions from current published specifications or product documents.

For the most up-to-date errata for this release, refer to the latest version of the *Nios II Embedded Design Suite Release Notes and Errata* on the Altera website. You can refer to release notes and errata for the Nios II processor core in the *Nios II Processor* chapter of the *MegaCore IP Library Release Notes and Errata*.

Revision History

Table 1 shows the revision history for the Nios II EDS, including the Nios II C2H Compiler.

Table 1. Nios II Embedded Design Suite Revision History

Version	Date	Description
9.1 SP2	March 2010	Maintenance release
9.1 SP1	January 2010	Maintenance release
9.1	November 2009	■ The Nios II Software Build Tools (SBT) for Eclipse, an integrated development environment based on Eclipse. Recommended for all new Nios II projects. ■ The Nios II Integrated Development Environment (IDE) is available in the Nios II EDS, but not recommended for new projects. ■ The Nios II Studio is discontinued. ■ The vectored interrupt controller (VIC) core offers high-performance, low-latency interrupt handling. ■ Nios II Design Examples with memory management unit (MMU): ■ Nios II 3C120 Hardware Reference Design with MMU, Cyclone® III Edition ■ Nios II 4S230 Hardware Reference Design with MMU, Stratix® IV GX Edition Commercial and open-source support for embedded Linux is available for these systems. ■ The Embedded Systems Lab for DE1 for Quartus® and Nios II EDS v9.1 is available for download from the Nios Community Wiki (www.nioswiki.com). ■ Tcl scripting for software packages and drivers enhanced to support callback functions.
9.0	March 2009	Added the `nios2-convert-ide2sbt` tool, which converts projects from the Nios II IDE development flow to the Nios II SBT development flow
8.1 SP1	January 2009	Nios Community Wiki URL changing (www.nioswiki.com)
8.1	November 2008	Maintenance release

For more information on new features and changes, refer to the *Nios II Processor Reference Handbook*, the *Nios II Software Developer's Handbook*, and the *Nios II C2H Compiler User Guide*. For information about changes to the Nios II processor core, refer to the *Nios II Processor* chapter of the *MegaCore IP Library Release Notes and Errata*.

Revision Details

This section provides details about some of the Nios II EDS revisions shown in Table 1.

Nios II Software Build Tools for Eclipse

The Nios II Software Build Tools (SBT) for Eclipse is a set of plugins based on the popular Eclipse framework and the Eclipse C/C++ development toolkit (CDT) plugins. The Nios II SBT for Eclipse provides a consistent development platform that works for all Nios II processor systems. You can accomplish most Nios II software development tasks within Eclipse, including creating, editing, building, running, debugging, and profiling programs. The Nios II SBT for Eclipse is built from the ground up as a set of plugins to standard Eclipse. The Nios II SBT for Eclipse is based on the Nios II Software Build Tools, a set of powerful commands, utilities and scripts for creating applications, board support packages and software libraries, and managing build options.

Table 2 compares the Nios II Integrated Development Environment (IDE) with the Nios II SBT for Eclipse.

Table 2. Differences Between Nios II SBT for Eclipse and Nios II IDE (Part 1 of 2)

Feature	Nios II IDE	Nios II SBT for Eclipse
Types of projects	■ Application ■ Board support package (BSP)—Called a system library in the IDE ■ User Library (optional)	■ Application ■ BSP ■ User Library (optional)
Handling updates to the SOPC Builder system	Updates the BSP automatically	You must initiate the BSP update (in the Nios II menu, click Generate BSP)
Options for creating projects	Application from template. BSP is created automatically as part of application.	■ New Application and BSP from template ■ Standalone application ■ Standalone BSP ■ Standalone library
Handling updates to BSP	Entire BSP is rebuilt. This process can be time-consuming.	Smart rebuild only recompiles changed files
Compatibility with command-line flow	None	Full compatibility with the Nios II SBT in the Nios II Command Shell
Modifying BSP properties	Limited control over BSP properties through the System Library page of the Properties dialog box.	Full control over BSP properties through the BSP Editor

Table 2. Differences Between Nios II SBT for Eclipse and Nios II IDE (Part 2 of 2)

Feature	Nios II IDE	Nios II SBT for Eclipse
Nios II C2H Compiler	Supported	Not supported
Nios II Flash Programmer	Limited control over the contents of the flash image	■ Allows you to create complex flash images containing Executable and Linking Format Files (.elf), SRAM Object Files (.sof), and binary data files in specified locations ■ Allows you to save flash programmer settings for repeatable flash programming
Scripting and regression testing	Not Available	Project creation and build can be scripted using the SBT command line

Converting IDE Projects to the SBT for Eclipse

If you have a pre-existing Nios II IDE project, before you can continue development in the Nios II SBT for Eclipse, you must convert it to the SBT and import it to the Eclipse environment. To convert and import an IDE project to the Nios II SBT for Eclipse, execute the following steps:

1. Convert the IDE project to an SBT-based project with the **nios2-convert-ide2sbt** command line utility.
2. Import the converted project to the Nios II SBT for Eclipse.

For a detailed description of the conversion process, refer to "Porting Nios II IDE Projects to the Software Build Tools" in the *Using the Nios II Integrated Development Environment* appendix of the *Nios II Software Developer's Handbook*. For a description of the process of importing a project, refer to "Importing a Command-Line Project" in the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*.

Learning about the Nios II SBT for Eclipse

Refer to the following documents to help you get acquainted with the Nios II SBT for Eclipse and related tools:

- *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*
- *Nios II Hardware Development Tutorial*
- The Comprehensive Nios II Embedded Design Suite support page
- The Nios II Software Build Tools for Eclipse support page

Errata Summary

Table 3 summarizes the issues that affect the Nios II Embedded Design Suite versions 8.1 through 9.1 SP2. For a detailed description of each issue, click on the issue name.

Refer to the Altera Knowledge Database for older errata and solutions.

Table 3. Nios II EDS Errata *(1)* (Part 1 of 5)

Added or Updated	Issue	Page	Affected Version				
			9.1 SP2	9.1 SP1	9.1	9.0	8.1
Nios II SBT for Eclipse							
May 2010	Errors Creating or Importing Software Projects	8	✓	✓	✓	—	—
	Nios II Options Do Not Appear in Eclipse	9	✓	✓	✓	—	—
Feb 2010	Build Errors on Software for Pre-Existing Design with SG-DMA	9	✓	✓	✓	—	—
	Spurious System ID Mismatch Error	9	✓	✓	✓	—	—
	Errors Debugging as Local C/C++ Application	10	✓	✓	✓	—	—
Nov 2009	Error Messages on Console When Debugging	10	✓	✓	✓	—	—
	java.lang.NullPointerException Error When Running Application	11	✓	✓	✓	—	—
	Update Failure When Setting Empty Properties in the Nios II Properties Page	11	✓	✓	✓	—	—
	Errors Converting Nios II IDE Multiprocessor Project	11	✓	✓	✓	—	—
	DHCP Time-Out on EP3C120	12	✓	✓	✓	—	—
	No Nios II Multiprocessor Collection Run Configuration	12	✓	✓	✓	—	—
	Stop on Startup Option in Run Configuration Has No Effect	12	✓	✓	✓	—	—
	Project Imported From Command Line Fails to Build Correctly	13	✓	✓	✓	—	—
	Debugger Breaks in crt0.s Instead of main()	13	✓	✓	✓	—	—
C2H Compiler							
Nov 2009	Error "java.lang.NoClassDefFoundError" During C2H Compilation	14	—	—	—	Fixed	✓
	C2H Cache Coherency Issue with Global Variables	14	—	—	—	—	Fixed
Jul 2008	Functions Declared Without a Return Type Are Not Supported	15	✓	✓	✓	✓	✓
Oct 2007	Pre-7.1 Systems Are Not Supported	15	✓	✓	✓	✓	✓
	--src-dir SBT Argument Does Not Work With C2H	15	✓	✓	✓	✓	✓
May 2007	Accelerator Generation Failure If Tools Are Installed in Path With Spaces	15	✓	✓	✓	✓	✓
	The C2H Compiler Regenerates an Accelerator Unnecessarily	16	✓	✓	✓	✓	✓
	Error: c2h_accelerator_base_addresses.h: No such file or directory	16	✓	✓	✓	✓	✓
	Java Heap Space Exception if Quartus II Compilation is Enabled	16	✓	✓	✓	✓	✓
	Pointer Dereferences to Volatile Types	17	✓	✓	✓	✓	✓
Dec 2006	C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File	17	✓	✓	✓	✓	✓

Table 3. Nios II EDS Errata *(1)* (Part 2 of 5)

Added or Updated	Issue	Page	Affected Version				
			9.1 SP2	9.1 SP1	9.1	9.0	8.1
Jun 2006	Array Elements in Structures Do Not Copy Correctly	17	✓	✓	✓	✓	✓
	Clean Build Causes Build Failure	18	✓	✓	✓	✓	✓
	Changing Build Configurations Produces Unexpected Results	18	✓	✓	✓	✓	✓
	Hardware Accelerators Remain After Deleting the Software Project *(2)*	19	✓	✓	✓	✓	✓
	Incorrect Results From Logical or Conditional Operation With Side-Effects	19	✓	✓	✓	✓	✓
	Launch SOPC Builder Button in C2H View	20	✓	✓	✓	✓	✓
Development Boards							
Jun 2006	Intermittent Failures While Accessing CompactFlash Card	20	✓	✓	✓	✓	✓
Documentation Issues							
May 2010	Incorrect Information about Nested Exceptions	21	✓	✓	✓	—	—
Feb 2010	Missing Documentation of Interrupt API Properties	22	✓	✓	✓	—	—
	Valid Range of hal.log_flags is –1 to 3	22	✓	✓	✓	—	—
Nov 2009	NicheStack TCP/IP Stack - Nios II Edition Does Not Support TFTP	23	✓	✓	✓	✓	✓
	Error Message After Renaming Project: “Resource is out of sync with the system”	23	✓	✓	✓	—	—
	Obsolete Wiki URLs in Help Systems and Handbook	23	—	—	—	Fixed	✓
	Missing Source Control Information	24	—	—	—	Fixed	✓
Mar 2009	Compiler Flags for Building Custom Newlib	25	✓	✓	✓	✓	✓
Nov 2008	Error Message on Invalid Exception or Reset Vector	26	✓	✓	✓	✓	✓
Oct 2007	Nios II IDE Online Help Expand Buttons Do Not Work	27	✓	✓	✓	✓	✓
Hardware Abstraction Layer							
Feb 2010	Missing Structure Member Errors in alt_log_printf.c with Small JTAG UART Driver	27	✓	✓	✓	—	—
	“unused variable” Warning When Building alt_main.c	28	✓	✓	✓	—	—
	Build Warnings in Performance Counter Driver Files	28	✓	✓	✓	—	—
Hardware Example Designs							
Mar 2007	Incorrect PLL Phase Shift in Stratix II and Cyclone II Designs	29	✓	✓	✓	✓	✓
Software Examples							
Nov 2009	Hardware Tutorial Software Example Hangs on Some Boards	30	✓	✓	✓	✓	✓
Nov 2008	Simple Socket Server Example Displays Incorrect IP Address	30	✓	✓	✓	✓	✓
Jun 2006	Networking Examples	31	✓	✓	✓	✓	✓
Flash Programmer							
Feb 2010	Unable to Configure FPGA from Flash with Parallel Flash Loader	31	Fixed	Fixed	✓	✓	—
	Boot Failure or elf2flash Error Due to Incorrect Flash File Size	32	✓	✓	✓	✓	✓

Table 3. Nios II EDS Errata *(1)* (Part 3 of 5)

Added or Updated	Issue	Page	Affected Version				
			9.1 SP2	9.1 SP1	9.1	9.0	8.1
May 2007	Delay When Creating New Flash Programmer Configuration	33	✓	✓	✓	✓	✓
Dec 2006	elf2flash File Size Limit	33	✓	✓	✓	✓	✓
	Error Message “No such file or directory” When Programming Flash	33	✓	✓	✓	✓	✓
Hardware Simulation							
May 2010	Vectored Interrupt Controller Does Not Support VHDL Simulation Models	33	✓	✓	✓	—	—
Dec 2006	Error “UNC paths are not supported” Launching ModelSim	34	✓	✓	✓	✓	✓
Jun 2006	Uninitialized .bss Variables in Simulation	34	✓	✓	✓	✓	✓
Host Platform							
Nov 2009	Cannot Launch Nios II SBT for Eclipse on Red Hat Linux	34	✓	✓	✓	—	—
Dec 2007	Windows/Cygwin: Nios II Processor Generation Failure	35	✓	✓	✓	✓	✓
Oct 2007	Windows Vista: Limited Support in Nios II EDS	35	✓	✓	✓	✓	✓
Nios II IDE							
Building Projects							
Mar 2009	Nios II IDE Command-Line Tools Select Wrong Workspace on Linux	36	✓	✓	✓	✓	✓
	Nios II IDE Command-Line Tools Hang on Windows	37	✓	✓	✓	✓	✓
Nov 2008	Nios II IDE Cannot Find stdio.h in Outline View	37	✓	✓	✓	✓	✓
Jul 2008	Nios II IDE Hangs With UNC Project Path	37	✓	✓	✓	✓	✓
	Build Failure with Nios II Advanced Exceptions, MMU, and MPU	38	✓	✓	✓	✓	✓
May 2007	Build Command Not Functional for BSPs Created With the Nios II SBT	38	✓	✓	✓	✓	✓
	Linker Errors with Dual-Port Memories	39	✓	✓	✓	✓	✓
	User-Managed BSP Settings Not Supported in Nios II IDE	39	✓	✓	✓	✓	✓
	URL Project Location Causes Project Creation Error	39	✓	✓	✓	✓	✓
Dec 2006	Compilation Error with Separate Exception Stack Option	40	✓	✓	✓	✓	✓
	Incorrect Stack and Heap Free Space Report	40	✓	✓	✓	✓	✓
	Nios II IDE Reports Problems Without Displaying Error in Console	40	✓	✓	✓	✓	✓
Debugging Projects							
Nov 2009	Method for Accessing MMU and MPU Registers in the Debugger	41	✓	✓	✓	✓	✓
Nov 2008	IDE Cannot Display Imported Profiling Data	41	✓	✓	✓	✓	✓
Jul 2008	Error Message when Downloading .elf File	42	✓	✓	✓	✓	✓
	Variable Casting Unsupported in ISS	42	✓	✓	✓	✓	✓
	Trace Debug Does Not Support Instruction-Related Exceptions	42	✓	✓	✓	✓	✓
Oct 2007	Trace Debug Does Not Support the JMPI Instruction	42	✓	✓	✓	✓	✓

Table 3. Nios II EDS Errata *(1)* (Part 4 of 5)

Added or Updated	Issue	Page	Affected Version				
			9.1 SP2	9.1 SP1	9.1	9.0	8.1
May 2007	Cannot Locate Source Code in Driver Files Shared by Multiple Projects	43	✓	✓	✓	✓	✓
	Console Window Is Not Updated After ISS Error	43	✓	✓	✓	✓	✓
	ISS Fails on Designs Containing Triple Speed Ethernet MAC or SG-DMA Components	43	✓	✓	✓	✓	✓
	Memory Window Sets Control Register Values Incorrectly	43	✓	✓	✓	✓	✓
Dec 2006	Programs That Interact With a Terminal Console on Windows Do Not Work	44	✓	✓	✓	✓	✓
	"Run as ModelSim" in the Nios II IDE Fails	44	✓	✓	✓	✓	✓
	The Restart Command on the Run Menu Does Not Work	44	✓	✓	✓	✓	✓
Jun 2006	Watchpoints Do Not Work on Certain Variables	45	✓	✓	✓	✓	✓
Navigating Projects							
Dec 2006	Nios II IDE Freezes While Displaying the Splash Screen	45	✓	✓	✓	✓	✓
	Internal Error When Double-Clicking on a Large Objdump File	45	✓	✓	✓	✓	✓
Jun 2006	C/C++ Scanner Does Not Support Certain C/C++ Constructs	46	✓	✓	✓	✓	✓
Nios II SBT							
May 2010	BSP Editor Does Not Show Command-Line Help	46	✓	✓	✓	—	—
Feb 2010	BSP Not Updated for Memory Size Changes in SOPC Builder	47	✓	✓	✓	—	—
Nov 2009	Build Errors on Interrupt API Calls	47	✓	✓	✓	—	—
	BSP Editor Incorrectly Allows Mixed Drivers	48	Fixed	Fixed	Fixed	✓	✓
	nios2-bsp --help Displays Incorrect List of BSP Types	48	Fixed	Fixed	Fixed	✓	✓
	nios2-bsp-create-settings Fails to List BSP Types	49	—	—	—	—	Fixed
	Makefile Error with Absolute Source Path	49	—	—	—	—	Fixed
	Incorrect Linker Errors	50	—	—	—	—	Fixed
	Java Heap Space Error When Creating BSP	50	—	—	—	Fixed	✓
Oct 2007	SBT Fails if Nios II EDS is Installed in a Path Containing Spaces	50	✓	✓	✓	✓	✓
May 2007	User-Managed BSP Settings Not Supported in Nios II IDE	39	✓	✓	✓	✓	✓
	Upper-Case File Extensions Not Supported	51	✓	✓	✓	✓	✓
Peripherals							
May 2010	Vectored Interrupt Controller Does Not Support VHDL Simulation Models	33	✓	✓	✓	—	—
Nov 2009	SBT Fails With UART Driver	51	—	—	—	—	Fixed
Jul 2008	Unaligned Transfers of Small Payloads Fail on SG-DMA	52	✓	✓	✓	✓	✓
May 2007	DMA Controller Always Busy in Burst Mode	52	✓	✓	✓	✓	✓
	Non-System-Wide Reset Can Cause Improper Initialization of Mailbox Core	52	✓	✓	✓	✓	✓

Table 3. Nios II EDS Errata (1) (Part 5 of 5)

Added or Updated	Issue	Page	Affected Version				
			9.1 SP2	9.1 SP1	9.1	9.0	8.1
Target Software							
Nov 2009	NicheStack tcp_wake() Function Does Not Awaken All select() Sleepers	53	✓	✓	✓	—	—
	EIC-Based Nios II System Crashes With NicheStack	53	✓	✓	✓	—	—
Mar 2009	Gigabit Ethernet Performance Issues	53	—	—	—	Fixed	✓
Dec 2006	stdio Does Not Work with MicroC/OS-II and Small C Library	54	✓	✓	✓	✓	✓
Jun 2006	cout From MicroC/OS-II Task Does Not Send Data to stdout	54	✓	✓	✓	✓	✓
	malloc(), realloc() Failures With MicroC/OS-II	54	✓	✓	✓	✓	✓
Toolchain (gcc, gdb, etc.)							
Feb 2010	Unable to Configure FPGA from Flash with Parallel Flash Loader	31	Fixed	Fixed	✓	✓	—
Nov 2009	sof2flash Names Output Files Incorrectly	55	—	—	—	—	Fixed
Dec 2006	GNU Assembler Does Not Accept the --defsym Flag	55	✓	✓	✓	✓	✓

Notes to Table 3:

(1) Refer to the Altera Knowledge Database for older errata and solutions.

(2) For further information, refer to the *Nios II C2H Compiler User Guide*.

Nios II SBT for Eclipse Errata

This section describes in detail the issues related to the Nios II SBT for Eclipse.

Errors Creating or Importing Software Projects

If you place software projects in your Eclipse workspace directory, you might experience project-related errors. For example, you might see one of the following error messages:

- `Unable to create project`
  `Reason: Unable to create project in workspace directory`
- `Failed to import`
  `Reason: Unable to import project`

Workaround

Switch to a different workspace, or create a new workspace, separate from your project files. To switch workspaces or create a workspace, in the File menu, click **Switch Workspace**.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Nios II Options Do Not Appear in Eclipse

When Nios II SBT for Eclipse starts, the Nios II plugins might fail to load, resulting in the following symptoms:

- The Nios II perspective is not available
- No Nios II items appear when you click **New** in the File menu

Eclipse can behave this way if you have installed the Nios II EDS in the same directory as a previous installation.

When the Altera Complete Design Suite (ACDS) installer installs the Nios II EDS, you specify a root directory for the EDS file structure. For example, if the Nios II EDS 9.1 is installed on the Windows operating system, the root directory might be **c:\altera\91\nios2eds**. For simplicity, Altera documentation refers to this directory as <*Nios II EDS install path*>.

When the ACDS installer removes the Nios II EDS, it leaves behind some files in <*Nios II EDS install path*>. If you reinstall the Nios II EDS in the same directory, these leftover files might cause the Nios II SBT for Eclipse to work incorrectly.

Workaround

To correct this problem, execute the following steps:

1. Uninstall the Nios II EDS.
2. Delete <*Nios II EDS install path*>, including all subdirectories, before reinstalling the Nios II EDS.

   Alternatively, reinstall the Nios II EDS in a different directory.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Build Errors on Software for Pre-Existing Design with SG-DMA

If you have a Nios II system generated with SOPC Builder v. 9.0 or earlier, and it contains the Scatter-Gather DMA (SG-DMA) component, you cannot build software for it with the Nios II SBT for Eclipse v. 9.1 or later. If you attempt to do so, the Nios II compiler reports errors.

Workaround

Regenerate the design with SOPC Builder v. 9.1 or later before attempting to compile the software in the Nios II SBT for Eclipse.

Solution Status

Not fixed.

Spurious System ID Mismatch Error

You might see the following error message when attempting to run or debug a Nios II software project, or to edit a launch configuration:

```
[Target Connection]:System ID mismatch - actual: "0xffffffff", expected: "<system id>"
```

<system id> is replaced by the system ID in your software project. When this error occurs, you cannot run or debug software.

This error normally means that the FPGA is unconfigured, or configured with a **.sof** file that does not correspond to your software project. However, the error occasionally appears even when the FPGA is configured with the correct **.sof**.

Workaround

After verifying that the FPGA is configured with the correct **.sof**, you can use one of the following methods to work around this issue:

- In the **Run Configuration** or **Debug Configuration** dialog box, on the **Target Connection** tab, click **Refresh Connections** repeatedly until the error message disappears.
- Delete the launch configuration and create a new launch configuration.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Errors Debugging as Local C/C++ Application

If you try to debug a Nios II software project as the Local C/C++ Application launch configuration type, you see one of the following errors, depending on your platform:

- Windows: “Error creating session”
- Linux: “Execution is suspended because of error”

The Nios II debug perspective fails to open.

This is expected CDT behavior in the Eclipse platform. Local C/C++ Application is the launch configuration type for a standard CDT project. To invoke the Nios II plugins, you must use a Nios II launch configuration type.

Workaround

Always debug Nios II software projects as the Nios II Hardware launch configuration type.

Error Messages on Console When Debugging

When debugging, you might see the following messages in the Nios II SBT for Eclipse console:

```
No symbol "auto" in current context.
```

```
No symbol "new" in current context.
```

These are normal messages from the GNU Debugger (GDB) console, not error messages. This issue does not impact the debug process.

Workaround

No workaround needed.

Solution Status

None needed.

java.lang.NullPointerException Error When Running Application

When you attempt to run or debug an application, the Nios II SBT for Eclipse might show an internal error message similar to the following:

```
Launching New_configuration java.lang.NullPointerException
```

This error occurs in either of the following circumstances:

- You attempt to run your project without rebuilding after performing a `make clean`.
- You delete a run configuration belonging to your project while the project is running.

You must restart Nios II SBT for Eclipse to recover from this error.

Workaround

Build the project before attempting to run or debug it. Do not delete a run configuration while it is in use.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Update Failure When Setting Empty Properties in the Nios II Properties Page

If you use the Nios II Properties page to update more than one board support package (BSP) setting to the empty string (`""`), the BSP is not updated.

Workaround

You can use either of the following workarounds to set multiple BSP settings to the empty string:

- Use the keyword `none` instead of `""` to set settings to the empty string.
- Use the BSP Editor instead of the Nios II Properties page.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Errors Converting Nios II IDE Multiprocessor Project

If you try use the **nios2-convert-ide2sbt** utility to convert an IDE multiprocessor project to the Nios II SBT, error messages appear, and the project fails to generate properly.

Workaround

Create new Nios II SBT application and BSP projects using project settings equivalent to the original Nios II IDE project settings.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

DHCP Time-Out on EP3C120

If you create a project based on the web server template targeting the Altera Embedded Systems Development Kit, Cyclone® III Edition (EP3C120) board, the system is unable to get an IP address at startup. DHCP times out and the system reverts to the default IP address.

Workaround

Create the project in the Nios II Command Shell and then import it to the Nios II SBT for Eclipse, by executing the following steps:

1. In the Nios II Command Shell, navigate to the following path:

   <*Nios II EDS install path*>**/examples/**<*language*>**/niosII_cycloneIII_3c120/tserd/software_examples/app/web_server_tse_3c120_rgmii**

   where <*language*> is either **vhdl** or **verilog**.

2. Type the following command to create a web server example for the EP3C120 kit:

   ```
   create-this-app↵
   ```

3. Import the command-line project to the Nios II SBT for Eclipse.

For information about importing command-line projects, refer to "Importing a Command-Line Project" in the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

No Nios II Multiprocessor Collection Run Configuration

The multiprocessor collection run configuration is not available in the Run menu. Therefore, you cannot use a single run configuration to download **.elf** files to run on multiple Nios II processors in the Nios II SBT for Eclipse.

Workaround

Create a separate run configuration for each Nios II processor in the hardware design, and launch them one at a time

Solution Status

Not fixed

Stop on Startup Option in Run Configuration Has No Effect

Turning on **Stop on Startup at** in the **Debugger** tab of a run configuration has no effect.

Workaround

You can manually set any breakpoint in the debug perspective.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Project Imported From Command Line Fails to Build Correctly

If you try to import a command-line project to the Nios II SBT for Eclipse, and your project has source files located outside the project folder, the Nios II SBT for Eclipse fails to build your project because the makefile does not include source files located outside the project folder.

Workaround

You can work around this issue with either of the following methods:

- When importing the project, disable source management through the **Nios II Application Properties** or **Nios II Library Properties** dialog box.

  ☞ You must manually keep your makefile updated to reflect source files added to or removed from the project.

- Manually link the affected source files to the project. Right-click the project, click **Properties**, expand **C/C++ General**, select **Paths and Symbols** and then **Source Location**, and click **Link Folder**.

Solution Status

This issue will be fixed in future release of the Nios II EDS.

Debugger Breaks in crt0.s Instead of main()

If you attempt to launch a debug session by right clicking the project, without first creating a run configuration, the debugger breaks in the initialization code in **crt0.s**, rather than at the top of the `main()` function. You must explicitly create the run configuration before launching the debug session.

Workaround

You can work around this issue with either of the following methods:

- Create the run configuration with the **Run Configuration** dialog box. You can then debug with this run configuration by right-clicking the project, clicking **Debug As**, and selecting the debug configuration. The program correctly breaks at `main()`.
- Manually set a breakpoint at `main()`.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

C2H Compiler Errata

This section describes in detail the issues related to the Nios II C2H Compiler.

Error "java.lang.NoClassDefFoundError" During C2H Compilation

C2H compilation fails with the following error:

```
java.lang.NoClassDefFoundError: com/altera/c2h/C2Hgenerate
Exception in thread "main" make: *** [c2h_hdl-t] Error 1
```

This error occurs when you invoke C2H from the Nios II IDE on the Windows operating system. This problem does not affect Linux installations, nor C2H compilation from the Nios II command shell.

Workaround

To correct the problem, visit mySupport and request the Nios II EDS 8.1 Patch 0.01 (PC version).

Solution Status

Fixed in v. 9.0

C2H Cache Coherency Issue with Global Variables

Cache coherency issues might be encountered when the following conditions are all true:

- A C2H-accelerated function writes to a global or static variable
- The Nios II processor is configured with a data cache
- The referenced global variable is located in cacheable memory

In this situation, the data in the global variable can be corrupted.

This issue can only arise if the accelerated function writes to the global or static variable. If the function merely reads from the variable, cache coherency is maintained correctly.

Workaround

Add a data cache flush operation in the accelerator's wrapper function. The wrapper can be found in **alt_c2h_<*function_name*>.c**. In this file, find the following comment:

```
/* Write 1 to address 0 starts the accelerator */
```

Directly after this comment, add the following function call:

```
alt_dcache_flush_all();
```

This function call ensures that the data cache is flushed immediately before the accelerator is started.

☞ Flushing the cache entails a performance penalty.

Alternatively, allocate your global and static variables in non-cacheable regions of memory. Placing these variables in non-cacheable regions eliminates any need to perform a data cache flush.

For information about allocating variables to non-cacheable regions, refer to the *Cache and Tightly-Coupled Memory* chapter of the *Nios II Software Developer's Handbook*.

Solution Status

Fixed in v. 8.1

Functions Declared Without a Return Type Are Not Supported

The C2H compiler does not support functions without an explicitly declared return type.

Workaround

If you are using the implicit `int` return type, declare the return type explicitly. If your function has no return value, declare it as `void`.

Solution Status

Not fixed

Pre-7.1 Systems Are Not Supported

Starting in v. 7.1, the C2H Compiler does not support SOPC Builder systems created in SOPC Builder 7.0 or earlier (systems based on a **class.ptf** file).

Workaround

Before using the C2H Compiler, you must upgrade your SOPC Builder system. Open the system in SOPC Builder v. 7.1 or later. SOPC Builder prompts you to convert the system, creating a ***_hw.tcl** file compatible with the C2H Compiler.

Solution Status

Not fixed

--src-dir SBT Argument Does Not Work With C2H

When using the C2H Compiler with Nios II SBT, an error occurs when using the `nios2-app-generate-makefile` argument `--src-dir` to specify a directory of source files. An example of the error is:

```
multiple definition of 'my_accelerated_function'
```

Workaround

Specify source files individually using the `--src-files` argument.

Solution Status

Not fixed

Accelerator Generation Failure If Tools Are Installed in Path With Spaces

If the path to your installation of the Altera Design Suite contains spaces, the C2H Compiler fails to generate the accelerators.

Workaround

Reinstall the tools to a path containing no spaces.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

The C2H Compiler Regenerates an Accelerator Unnecessarily

The C2H Compiler might regenerate an accelerator even when the accelerated function is unchanged. This problem can result from changes to files included by the C file containing the accelerated function. The C2H Compiler fails to check that the generated hardware description language (HDL) matches the previously generated HDL causing the system to be regenerated.

Workaround

To avoid this issue move the accelerator-specific information from the include file to a separate include file. This workaround prevents regeneration of the system when the HDL is unchanged.

Solution Status

Not fixed

Error: c2h_accelerator_base_addresses.h: No such file or directory

When a C2H accelerator is compiled for the first time, the following compile-time error can result if the **Analyze all accelerators** option is selected:

```
c2h_accelerator_base_addresses.h: No such file or directory.
```

Workaround

Click **Build software and generate SOPC Builder system** and build once before building with the **Analyze all accelerators** option.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Java Heap Space Exception if Quartus II Compilation is Enabled

If your design contains a C2H accelerator, and you select **Build software, generate SOPC Builder system, and run Quartus® II compilation** in the C2H view, you might see the following error during Quartus II compilation:

```
Exception in thread "main" java.lang.OutOfMemoryError: Java
heap space

make: *** [c2h_hdl-t] Error 1
```

Workaround

Select the **Build software and generate SOPC builder system** option in the C2H Compiler settings window, and then manually launch the Quartus II software to compile the design.

Solution Status

Not fixed

Pointer Dereferences to Volatile Types

The C2H Compiler treats pointer dereferences to a volatile type as if they alias all other pointer dereferences. Pointers that are restrict-qualified are treated the same way.

The two loops in Example 1 cannot be scheduled concurrently because the `volatile` qualification overrides the `__restrict__` pragma.

Example 1. Non-Concurrent Loops

```
volatile int * __restrict__ fifo_rd = FIFO_RD_BASE;
volatile int * __restrict__ fifo_wr = FIFO_WR_BASE;
for ()
{
  *fifo_wr = ....;
}
for ()
{
  ... = *fifo_rd;
}
```

Workaround

Divide the function into multiple interrupt request (IRQ)-enabled accelerators that are launched concurrently from the processor, and use FIFO buffers to communicate between them.

Solution Status

Not fixed

C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File

When accelerating a function in a file, the C2H Compiler cannot link subfunctions that are defined in a different file.

Workaround

Include all subfunctions called by the accelerated function within the same source code file.

Solution Status

Not fixed

Array Elements in Structures Do Not Copy Correctly

C2H accelerators do not correctly copy array elements that are elements of structures.

In Example 2, the `a` and `b` elements of the structure copy correctly, but the `buf` element does not. After this assignment, `struct_a` equals `{9, 8, {3, 3, 3, 3}}`.

Example 2. Array Elements of Structs

```
typedef struct my_struct {
  int a;
  int b;
  int buf[BUF_SIZE];
}MY_STRUCT;
MY_STRUCT struct_a = {1, 2, {3, 3, 3, 3}};
MY_STRUCT struct_b = {9, 8, {7, 7, 7, 7}};
struct_a = struct_b;
```

Workaround

Copy the array elements explicitly, as shown in Example 3.

Example 3. Copying Array Elements Explicitly

```
{
  int i=0;
  do
    {
      struct_a.buf[i] = struct_b.buf[i];
      i++;
    } while (i<LENGTH_OF_BUF_ELEMENT)
}
```

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Clean Build Causes Build Failure

Performing a clean build on a Nios II IDE project that contains a hardware accelerator can cause the next build to fail in the IDE, because the clean build erroneously deletes a file required by the C2H Compiler.

Workaround

Do not perform a clean build on projects that use hardware accelerators. If you have already performed a clean build, recompile with option **Build software, generate SOPC Builder system, and run Quartus II compilation** to regenerate the necessary files.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Changing Build Configurations Produces Unexpected Results

The C2H Compiler does not support multiple build configurations (for example Release or Debug) in the Nios II IDE. After creating one or more accelerators in a particular configuration, the C2H Compiler produces undefined results if you switch to a different build configurations and create more accelerators.

Workaround

For a specific SOPC Builder system and Nios II IDE project, specify C2H accelerators in only one build configuration. You can use multiple build configurations, as long as only one configuration specifies C2H Compiler settings.

Solution Status

Not fixed

Hardware Accelerators Remain After Deleting the Software Project

If a system contains C2H accelerators, deleting the software project that defines the accelerators does not remove the accelerators from the hardware system, and the accelerator logic remains in the SOPC Builder system.

Workaround

To remove an accelerator from a system, delete the accelerator from the C2H view in the Nios II IDE first, and then recompile the software project. The C2H Compiler then removes the accelerator from the SOPC Builder system. Once the compilation is complete then the software application can be deleted from the workspace.

Solution Status

Not fixed

Incorrect Results From Logical or Conditional Operation With Side-Effects

The C2H Compiler always evaluates both operands of logical (&&, `||`) and conditional (`?:`) operators. This is different from expected American National Standards Institute (ANSI) C behavior, where operands are evaluated left-to-right, and unnecessary operands are skipped.

For example, in the expression `(i-- && j--)`, if the value of `i` is zero, ANSI C does not evaluate the right-hand-side (RHS) expression, and `j` is not decremented. By contrast, the C2H Compiler evaluates both sides, decrementing `j`.

The following expressions are other examples that might be affected by this issue:

```
if (i-- || j++)
    ...;


a = ((cond == 1)?  i++ : j++);
```

Workaround

Use logical and conditional operations whose operators have no side effects. Operations with side effects include pre- and post- increment and decrement operations (`++`, `--`), memory operations (`*`, `[]`, `.`, `->`), and function calls.

Solution Status

Not fixed

Launch SOPC Builder Button in C2H View

When the Nios II IDE workspace contains multiple projects with multiple system libraries, the incorrect SOPC Builder system might open when you click **Launch SOPC Builder** in the C2H view.

Workaround

Launch SOPC Builder from the Quartus II software. Alternatively, keep only one system library project open at a time while using the C2H Compiler.

Solution Status

Not fixed

Development Board Errata

This section describes in detail the Nios II EDS issues related to Nios development boards.

Intermittent Failures While Accessing CompactFlash Card

The Nios II Development Kit version 5.0 and later includes a CompactFlash controller peripheral suitable for interfacing to CompactFlash cards in True IDE mode on Nios development boards. For True IDE mode to operate, CompactFlash cards require that the `ATASEL_N` input be driven to ground during power-up.

The CompactFlash controller peripheral includes a configurable power register, used to cycle power to CompactFlash cards from Nios II software through a metal oxide semiconductor field-effect transistor (MOSFET) on the Nios development boards. However, in certain development boards, power to the CompactFlash card does not turn off completely during this power cycle operation. Because of this condition, the CompactFlash might not sample the `ATASEL_N` pin during the power-cycle operation after FPGA configuration when this pin is driven to ground. Instead, the CompactFlash card might sample the `ATASEL_N` pin when power is first applied to the development board, when I/O is not yet driven by the FPGA (before FPGA configuration).

Workaround

If you encounter errors with CompactFlash when using the Nios development boards, try one of the following solutions:

- Use a different CompactFlash card. Certain cards are more susceptible to the power-cycling issue than others.
- Modify the Nios development board. This is recommended if you are familiar and comfortable with board-level modifications. Disconnect pin 9 (`ATASEL_N`) on the CompactFlash socket on your Nios development board and tie this pin to ground.

☞ The CompactFlash socket uses a staggered numbering on the pins (starting from pin 1: 1, 26, 2, 27, ...); refer to the CompactFlash Association specification for right-angle surface-mount connectors for exact specifications on this connector. This modification permanently enables True IDE mode operation.

Solution Status

Not fixed

Documentation Errata

This section describes in detail the Nios II EDS documentation issues.

Incorrect Information about Nested Exceptions

"Exception Processing" in the *Programming Model* chapter of the *Nios II Processor Reference Handbook* incorrectly states that multiple interrupts with different requested interrupt levels (RILs) must not be assigned to the same shadow register set.

Workaround

The correct information is as follows:

Multiple interrupts with different RILs can be assigned to a single shadow register set. However, with multiple register sets, you must not allow the RILs assigned to one shadow register set to overlap the RILs assigned to another register set.

Table 2–4 and Table 2–5 illustrate the validity of register set assignments when preemption within a register set is enabled.

Table 2–4. Example of Illegal RIL Assignment

RIL	Register Set 1	Register Set 2
1	IRQ0	
2	IRQ1	
3		IRQ2
4	IRQ3	
5		IRQ4
6		IRQ5
7		IRQ6

Table 2–5. Example of Legal RIL Assignment

RIL	Register Set 1	Register Set 2
1	IRQ0	
2	IRQ1	
3	IRQ3	
4		IRQ2
5		IRQ4
6		IRQ5
7		IRQ6

Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

Missing Documentation of Interrupt API Properties

“Tcl Commands” in the *Nios II Software Build Tools Reference* chapter of the *Nios II Software Developer’s Handbook* does not include information about two `set_sw_property` command properties related to the application programming interfaces (APIs) for interrupts.

Workaround

The missing information belongs with the `set_sw_property` command, in “Tcl Commands” in the *Nios II Software Build Tools Reference* chapter of the *Nios II Software Developer’s Handbook*.

The missing information follows:

- `supported_interrupt_apis`—Specifies the interrupt API that the device driver supports. The Nios II SBT analyzes this property for each driver in the system to determine the appropriate API to be used in the system.

  Specify `legacy_interrupt_api` if the device driver supports the legacy API only or `enhanced_interrupt_api` if the device driver supports the enhanced API only. Specify both using a quoted list if the device driver supports both APIs.

  If you do not specify which API your device driver supports, the Nios II SBT assumes that only the legacy interrupt API is supported.

  ☞ This property is only available for device drivers.

  For more information about the legacy and enhanced APIs, refer to “Exception Handling” in the *Nios II Software Build Tools* chapter of the *Nios II Software Developer’s Handbook*.

- `isr_preemption_supported`—Specify `true` if your device driver’s interrupt service routine (ISR) can be preempted by a higher priority ISR. If you do not specify whether the driver supports preemption, the Nios II SBT assumes that your device driver does not support preemption. If your driver does not have an ISR, but the associated device has an interrupt port, you can set this property to `true`.

  ☞ This property is valid for operating systems and device drivers.

Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

Valid Range of hal.log_flags is –1 to 3

The Nios II EDS documentation incorrectly states the valid range for the `hal.log_flags` BSP setting. Valid values of `hal.log_flags` range from –1 through 3.

Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

NicheStack TCP/IP Stack - Nios II Edition Does Not Support TFTP

The Nios II documentation should include the following information:

Versions of the NicheStack TCP/IP Stack other than the Nios II Edition include optional Trivial File Transfer Protocol (TFTP) client and server applications. However, the TFTP client and server are not thread safe. They cannot be used in systems with the MicroC/OS-II RTOS.

TFTP is not available with the NicheStack TCP/IP Stack - Nios II Edition. The TFTP client and server features are disabled.

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Error Message After Renaming Project: "Resource is out of sync with the system"

The *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook* does not explain the correct procedure for renaming a project.

Workaround

The missing information is as follows:

To rename a project in the Nios II SBT for Eclipse, execute the following steps:

1. Right click the BSP project and click **Rename**.
2. Type the new BSP name.
3. Right click the BSP project and click **Refresh**.

☞ If you neglect to refresh the project, you might see the following error message when you attempt to build it:

```
Resource <original_bsp_name> is out of sync with the system
```

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Obsolete Wiki URLs in Help Systems and Handbook

The following documents contain obsolete URLs for the Nios Community Wiki:

- The *Nios II Software Developer's Handbook*
- The *Nios II Embedded Design Handbook*
- The Nios II IDE help system

Workaround

You can visit the Nios Community Wiki at www.nioswiki.com.

Solution Status

Fixed in the documentation for v. 9.0.

Missing Source Control Information

The Quartus II Handbook refers to the *Using the Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook* for information about putting Nios II projects in source control. The referenced chapter only describes the process for Nios II SBT projects.

Workaround

The missing information follows.

Archiving Nios II IDE Software Projects

This section helps you identify the files you must include when archiving a Nios II IDE software project. With this information, you can archive a Nios II application project and its associated Nios II system library project.

You might want to archive your projects for one of the following reasons:

- To place them under source control
- To create backups
- To bundle the projects for transfer to another location

This section covers the following information:

- How to find and identify the files that you must include in an archived Nios II IDE software project.
- Which files must have write permission to allow the software projects to be built.

Table 4 and Table 5 list the files required by Nios II IDE software projects. This is the minimum set of files needed to completely rebuild a software project, including the executable and linking file (**.elf**).

Archive your Nios II IDE software projects together with the SOPC Builder system on which they are based. You cannot rebuild a Nios II IDE software project without its associated SOPC Builder system.

The files listed in Table 4 are located in the Nios II application project directory.

Table 4. Files Required for a Nios II Application Project

File Description	File Name	Write Permission Required?
All source files	for example: **app.c**, **header.h**, **assembly.s**, **lookuptable.dat**	No
Eclipse project file	**.project**	No
C/C++ Development Toolkit project file	**.cdtproject**	Yes
C/C++ Development Toolkit option file	**.cdtbuild**	No
Software configuration file	**application.stf**	No

The files listed in Table 5 are located in the Nios II system library project directory.

Table 5. Files Required for a Nios II System Library Project

File Description	File Name	Write Permission Required?
Eclipse project file	**.project**	Yes
C/C++ Development Toolkit project file	**.cdtproject**	Yes
C/C++ Development Toolkit option file	**.cdtbuild**	No
System software configuration file	**system.stf**	Yes

You must have write permission for certain files. The tools write to these files as part of the build process. If the files are not writable, the toolchain fails. Many source control tools mark local files read-only by default. In this case, you must override this behavior. You do not have to check the files out of source control unless you are modifying the Nios II software project.

Solution Status

In the documentation for v. 9.0, the missing information appears in the *Nios II Integrated Development Environment* chapter of the *Nios II Software Developer's Handbook*. In the documentation for v. 9.1, the missing information appears in the *Using the Nios II Integrated Development Environment* appendix of the *Nios II Software Developer's Handbook*.

Compiler Flags for Building Custom Newlib

"Common BSP Tasks" in the *Using the Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook* describes how to compile a custom version of the newlib library. However, it does not list the compiler flags that must be selected to generate the correct library.

Workaround

The missing information is as follows:

The Nios II EDS provides a number of precompiled newlib implementations. The provided libraries are precompiled with every viable combination of the GCC compiler flags shown in Table 6.

When you create a BSP with a precompiled newlib, the Nios II SBT selects the newlib matching your BSP's compiler settings. When you create a custom newlib, you must ensure that the compiler flags listed in Table 6 match your BSP's settings.

Table 6. GCC Compiler Options for Newlib

Option Name	Purpose
`-pg`	Link for profiling with gprof
`-EB`	Generate big-endian code
`-mstack-check`	Enable stack checking
`-mno-hw-mul`	Disable use of the `mul` family of instructions
`-mhw-mulx`	Enable use of the `mulx` family of instructions
`-mcustom-fpu-cfg`	Use a floating point custom instruction

The Nios II-specific compiler flag `-mcustom-fpu-cfg` requires the Nios II processor core to be implemented with the floating-point custom instruction as specified in the flag argument. The compiler flag can have one of the following two argument values:

- `-mcustom-fpu-cfg=60-1`—Use the Nios II floating-point custom instruction without divider
- `-mcustom-fpu-cfg=60-2`—Use the Nios II floating-point custom instruction with divider

For details about the Nios II-specific compiler flags `-mstack-check`, `-mno-hw-mul`, and `-mhw-mulx`, refer to “Altera Nios II Options” in *Using the GNU Compiler Collection (GCC)*, installed with the Nios II EDS. For details about the `-pg` compiler flag, refer to “Compiling a Program for Profiling” in *GNU Profiler*, installed with the Nios II EDS. To find the installed documentation, on the Windows Start menu, click **Programs** > **Altera** > **Nios II EDS <*version*>** > **Nios II <*version*> Documentation**, and then click **Literature**.

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Error Message on Invalid Exception or Reset Vector

Nios II compiler error messages, added in v 8.1, are not documented in the *Nios II Software Developer’s Handbook*.

Workaround

The missing information is as follows:

While building a Nios II software project, the compiler produces an error message if the SOPC Builder system defines an invalid exception or reset vector. Table 7 shows possible error messages and their causes.

In the SOPC Builder system, ensure that the exception and reset vectors conform to all the criteria listed in Table 7.

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Table 7. Exception and Reset Vector Error Messages (Part 1 of 2)

Message	Cause
`The section mapping ".entry" is not mapped to a memory region.`	The `.entry` section is mapped to a nonexistent memory region.
`The section mapping ".exceptions" is not mapped to a memory region.`	The `.exceptions` section is mapped to a nonexistent memory region.
`The section mapping ".entry" maps to a memory region "<`*`region name`*`>" which is not the required length "32".`	The `.entry` section is mapped to a memory region that is smaller than 32 bytes.
`The section mapping ".entry" maps to a memory region "<`*`region name`*`>" which does not start at the SOPC Design specified address "<`*`base address`*`>".`	The `.entry` section is mapped to a memory region that does not start at the reset vector base address.

Table 7. Exception and Reset Vector Error Messages (Part 2 of 2)

Message	Cause
`The section mapping ".exceptions" maps to a memory region "<region name>" which does not start at the SOPC Design specified address "<base address>".`	The `.exceptions` section is mapped to a memory region that does not start at the exception vector base address.
`The SOPC design Nios II reset and exception vector addresses are mapped to the same memory. For HAL the exception address must be at least 32 bytes larger than the reset address.`	The `.entry` section and `.exceptions` section map to the same memory device, and the base of the `.exceptions` section is less than 32 bytes above the base of the `.entry` section.
`The section mapping ".exceptions" mapped device "<memory 1>" is not the same as the SOPC Design specified device "<memory 2>".`	The `.exceptions` section is mapped to a device that is different from the exception vector device specified in the SOPC Builder system.
`The section mapping ".entry" mapped device "<memory 1>" is not the same as the SOPC Design specified device "<memory 2>".`	The `.entry` section is mapped to a device that is different from the reset vector device specified in the SOPC Builder system.

Nios II IDE Online Help Expand Buttons Do Not Work

The expand text arrow and the **Show All** option in the online help do not work in Internet Explorer.

Workaround

In Internet Explorer, carry out the following actions:

1. Click the **Refresh/Show Current Topic** icon (yellow arrows icon at the top of the search pane). The table of contents appears with the selected topic highlighted.
2. Click the highlighted topic to refresh the browser frame. Expanded text works.

Alternatively, use Firefox 2.0.0.6.

Solution Status

Not fixed

Hardware Abstraction Layer Errata

This section describes in detail the Nios II Hardware Abstraction Layer issues.

Missing Structure Member Errors in alt_log_printf.c with Small JTAG UART Driver

In a BSP, if you enable the small JTAG UART driver (`altera_avalon_jtag_uart_driver.enable_small_driver`), and enable Altera logging (`hal.log_port`) using the JTAG UART as the log port, when you build the project, the compiler reports missing structure members in **alt_log_printf.c**. The project fails to build.

Workaround

To avoid this issue, execute the following steps:

1. In function `alt_log_jtag_uart_print_control_reg()` in **alt_log_printf.c**, insert the italicized code as shown in Example 6.

Example 6. ALT_LOG_PRINTF() Correction

```
ALT_LOG_PRINTF(
    "%s SW CirBuf = %d, HW FIFO wspace=%d AC=%d WI=%d RI=%d WE=%d RE=%d\r\n",
    header,

/* Beginning of inserted code */
#ifndef ALTERA_AVALON_JTAG_UART_SMALL
    (dev->tx_out-dev->tx_in),
#else
    0,
#endif
/* End of inserted code */

    space,ac,wi,ri,we,re);
```

2. Rebuild the project.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

"unused variable" Warning When Building alt_main.c

If you set the `hal.enable_exit` BSP setting to `false`, when building the project you see the following warning in **alt_main.c**:

```
warning: unused variable 'result'
```

Design Impact

This warning is harmless.

Workaround

Set `hal.enable_exit` to `true`.

Alternatively, you can ignore this harmless warning.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Build Warnings in Performance Counter Driver Files

If your Nios II system contains a performance counter, you might see the following warning when you build a BSP:

```
warning: return makes pointer from integer without a cast
```

This warning appears in `alt_get_performance_counter_base()` in the software file **altera_avalon_performance_counter.c**.

This warning appears when the your BSP uses the Altera performance counter driver, and the C preprocessor symbol `PERFORMANCE_COUNTER_0_BASE` is defined.

This warning is harmless.

Workaround

In the `alt_get_performance_counter_base()` function in **altera_avalon_performance_counter.c**, add typecast `(void *)` to `PERFORMANCE_COUNTER_0_BASE`.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Hardware Example Design Errata

This section describes in detail the Nios II EDS hardware example design issues.

Incorrect PLL Phase Shift in Stratix II and Cyclone II Designs

In several example designs targeting the Nios II Cyclone II 2C35 and Nios II Stratix® II 2S60 RoHS development boards, the synchronous static random access memory (SSRAM) and legacy DDR SDRAM clocks are implemented with an incorrect phase-locked loop (PLL) phase shift.

This issue affects the following example designs:

- 2s60 RoHS Triple Speed Ethernet SG-DMA
- 2s60 RoHS Standard
- 2s60 RoHS Full Featured
- 2c35 Triple Speed Ethernet SG-DMA
- 2c35 Standard
- 2c35 Full Featured

This issue also affects any custom 2c35 or 2s60 RoHS design using SSRAM or legacy DDR SDRAM.

The incorrect phase shift might cause timing violations when reading from or writing to SSRAM or legacy DDR SDRAM.

Workaround

To correct the PLL settings, perform the following steps:

1. Open the standard design in SOPC Builder.
2. Double-click the component instance named `pll` to launch the PLL MegaWizard™ interface.
3. Click **Launch Altera's ALTPLL MegaWizard** to launch the MegaWizard interface.
4. Select the **Output Clocks** page and then the clock page used to generate the SSRAM clock. Depending on the particular design, this might be **clk c1**, **clk c2**, or another clock.

5. Ensure that **Clock phase shift** is –3.38 ns.
6. Click the clock page used to generate the write clock for the legacy DDR SDRAM, if any. Depending on the particular design, this might be **clk c1**, **clk c2**, or another clock.
7. Ensure that **Clock phase shift** is 270 degrees.
8. Click **Finish** to save changes and exit the PLL MegaWizard interface.
9. Click **Finish** to save settings to the PLL instance in SOPC Builder.
10. Regenerate the system in SOPC Builder and recompile in the Quartus II software. Refer to the **readme.txt** file in the design example folder for information on compiling designs containing the DDR controller.

For further information refer to **ssram_interface_readme.html**, located in the **<*Quartus II installation directory*>/sopc_builder/documents** folder. This document discusses the SSRAM timing analysis methodology in detail. Additionally, *AN 411: Understanding PLL Timing for Stratix II Devices* discusses clock phase shift calculations and assignments for PLLs in Stratix II devices.

Solution Status

Not fixed

Software Example Errata

This section describes in detail the Nios II EDS software example issues.

Hardware Tutorial Software Example Hangs on Some Boards

The **count binary** application used in the tutorial hangs when run on some Altera development boards. The software runs through the binary count once and stops responding.

This issue affects boards that do not have a display to handle the LCD output. The affected boards include the following:

- Embedded Systems Development Kit, Cyclone III Edition (EP3C120)
- Stratix IV GX FPGA Development Kit (EP4SGX230)

Workaround

Modify line 18 of the **count_binary.h** file, as follows:

```
# define LCD_PRINTF(lcd, args...) /* Do Nothing */
```

Rebuild and run the software again.

Solution Status

Not fixed

Simple Socket Server Example Displays Incorrect IP Address

The simple socket server networking software example displays an incorrect IP address for the incoming connection.

Workaround

There is currently no workaround for this problem. Ignore the IP address of any incoming connection that is displayed on the console.

Solution Status

Not fixed

Networking Examples

If you are running a networking software example, you might be asked for a nine-digit number. You are directed to find this number on a sticker on your Nios development board, identified by the prefix **ASJ**. Not all Nios development boards have this sticker.

Workaround

If your Nios development board does not have a sticker with the letters **ASJ** followed by a nine-digit number, enter a unique nine-digit number when prompted. To avoid network address conflicts, ensure that this number is unique to each Nios board connected to your network.

Solution Status

Not fixed

Flash Programmer Errata

This section describes in detail the Nios II EDS issues related to the flash programmer.

Unable to Configure FPGA from Flash with Parallel Flash Loader

If you program an FPGA image to flash memory on a board that utilizes the Parallel Flash Loader, the FPGA subsequently fails to load from flash memory. This issue arises because the **sof2flash** utility generates a flash file with an incompatible Programmer Object File (**.pof**) bitstream format.

Affected Configurations

This issue impacts the following target hardware:

- The Stratix IV GX FPGA Development Kit
- The Arria® II GX FPGA Development Kit
- Any hardware using the Parallel Flash Loader

Workaround

1. Visit the MySupport website and request Nios II EDS patch 0.73.

   Alternatively, upgrade to the Nios II EDS v. 9.1 SP1 or later.

2. Invoke **sof2flash** with the following options:

   - `--pfl`
   - `--optionsbits=`*`<option bit address>`*

   With these options, **sof2flash** generates a flash file with a compatible **.pof** file bitstream format.

Solution Status

Fixed in v. 9.1 SP1

Boot Failure or elf2flash Error Due to Incorrect Flash File Size

The `sof2flash` utility fails to recognize certain devices. As a result, it generates Motorola S-record (**.flash**) files of the wrong size.

With erasable programmable configurable serial (EPCS) memory, the `elf2flash --after` command fails to recognize the **.flash** file size. The resulting error prevents you from programming flash memory.

With Common Flash Interface (CFI) memory, you can program the **.flash** file, but the Nios II system fails to boot.

This error occurs when programming flash memory for the following devices:

- The following Stratix III devices:
  - EP3SL50
  - EP3SL50F484
  - EP3SL50F780
  - EP3SL110
  - EP3SL110F1152
  - EP3SL110F780
  - EP3SE80
  - EP3SE80F1152
  - EP3SE80F780
- All Stratix IV devices
- All Arria II GX devices

Workaround

Visit the MySupport website and request Nios II EDS 9.0 SP2 patch 2.01 and Quartus II 9.0 SP2 patch 2.63.

Alternatively, download and install the Altera Complete Design Suite v. 9.1 or later.

Solution Status

Fixed in v. 9.1

Delay When Creating New Flash Programmer Configuration

You might experience a delay of several seconds when creating a new Flash Programmer configuration in the Nios II IDE.

Workaround

There is no workaround.

Solution Status

Not fixed

elf2flash File Size Limit

The **elf2flash** utility supports **.elf** files up to approximately 24 MBytes in size. The **elf2flash** utility might fail on files larger than 24 MBytes, with the error message `java.lang.OutOfMemoryError`.

Workaround

Lower the number of symbols in your **.elf** file by turning off debug symbols.

Alternatively, specify less initialized data in the application.

Solution Status

Not fixed

Error Message “No such file or directory” When Programming Flash

When programming flash from a project stored in a path containing spaces, you might receive the spurious error message **No such file or directory**. The flash programmer does not correctly handle spaces in the directory path. However, this error is harmless, because flash programming completes successfully.

Workaround

None required.

Solution Status

Not fixed

Hardware Simulation Errata

This section describes in detail the Nios II EDS issues related to hardware simulation.

Vectored Interrupt Controller Does Not Support VHDL Simulation Models

SOPC Builder reports errors when you try to generate VHDL simulation files for the Vectored Interrupt Controller.

Workaround

There is no workaround.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Error “UNC paths are not supported” Launching ModelSim

If you launch ModelSim® from a working directory that is mapped via a universal naming convention (UNC) path (a path that starts with `//` instead of drive letter), you receive the following error message in SOPC Builder: **UNC paths are not supported. Defaulting to Windows directory**. This error occurs because ModelSim is calling a command shell, which does not support UNC paths.

Workaround

Map the UNC path to a drive letter and use the drive letter to reference the working directory in the launching shell.

Solution Status

Not fixed

Uninitialized .bss Variables in Simulation

If your program reads the value of an uninitialized `.bss` variable during HDL simulation, and the BSP (system library) is compiled with the **ModelSim only, no hardware support** property enabled in Nios II IDE, a warning appears about unfiltered data being `'x'`. This warning appears because when this property is enabled, the code that clears the `.bss` memory region is omitted to speed up HDL simulation so this memory region is uninitialized. The `.bss` region contains global and static local variables that are not initialized by the application so they default to a value of zero. When the Nios II processor reads uninitialized variables, it displays a warning and converts any of the bits of the uninitialized data to zero which correctly mimics the effect of the missing `.bss` clearing code. The Hardware Abstraction Layer (HAL) code that executes before and after `main()` might use `.bss` variables, so these warnings might appear even if your application does not use the `.bss` section.

Solution Status

Not fixed

Host Platform Errata

This section describes in detail the Nios II EDS issues related to the host development platform.

Cannot Launch Nios II SBT for Eclipse on Red Hat Linux

You might be unable to launch Nios II SBT for Eclipse on the Red Hat Linux operating system. This is an issue with the version of XULRunner in Eclipse 3.4. The issue is resolved in XULRunner 1.9.1, which is available with Eclipse 3.5.

Workaround

Upgrade XULRunner to a newer version. To determine what version of XULRunner you have, type the following command at the command prompt:

```
xulrunner -v↵
```

To upgrade XULRunner using the **yum** software package manager, type the following command:

```
sudo yum update xulrunner↵
```

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Windows/Cygwin: Nios II Processor Generation Failure

You might see the following error message when you generate your Nios II system in SOPC Builder:

```
Error: Generator program for module 'cpu_fpoint' did NOT run
successfully
```

This error might occur on a Windows-based system when all of the following conditions are true:

- You have installed a version of Cygwin other than the one distributed with the Quartus II software.
- You launch the Quartus II software and SOPC Builder from a Nios II Command Shell.
- You enable the floating point custom instruction (FPCI) in the Nios II processor core.

Workaround

Launch the Quartus II software and SOPC Builder from the Windows Start menu, and regenerate your system.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Windows Vista: Limited Support in Nios II EDS

The Quartus II software introduces Windows Vista (32-bit and 64-bit) support in version 7.2. However, the Nios II Embedded Design Suite supports only the following platforms:

- Windows 2000
- Windows XP
- Windows XP (64)
- Windows Vista Business Edition (32-bit)
- SUSE 9 (32-bit)
- SUSE 9 (64-bit)

- Red Hat Linux v3.0 (32-bit)
- Red Hat Linux v3.0 (64-bit)
- Red Hat Linux v4.0 (32-bit)
- Red Hat Linux v4.0 (64-bit)

Workaround

There is no workaround.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Nios II IDE Errata: Building Projects

This section describes in detail the issues related to building projects in the Nios II IDE.

Nios II IDE Command-Line Tools Select Wrong Workspace on Linux

On Linux systems, the Nios II IDE command-line tools use the default Eclipse project workspace, regardless what workspace you are using in the IDE. The default workspace is at `$SOPC_KIT_NIOS2/eclipse/nios2-ide-workspace-<version>`.

This issue affects the following tools:

- `nios2-create-system-library`
- `nios2-create-application-project`
- `nios2-build-project`
- `nios2-import-project`
- `nios2-delete-project`

Workaround

Explicitly supply the Nios II IDE command-line tools with a workspace location, by means of the `-data` command-line argument. The syntax of the `-data` argument is as follows:

```
-data <path to workspace>
```

The path to the workspace must be absolute, and must not contain whitespace. Aside from these restrictions, any valid file system path can be used for the workspace. Example 7 shows how to specify a workspace path.

Example 7. Providing a Non-Default Workspace Location in Linux

```
nios2-create-project -data $HOME/myworkspace <other arguments>↵
```

Solution Status

Not fixed

Nios II IDE Command-Line Tools Hang on Windows

On Windows systems, the Nios II IDE command-line tools are sometimes unable to locate the project workspace. When this happens, the tools might hang.

This issue potentially affects the following tools:

- `nios2-create-system-library`
- `nios2-create-application-project`
- `nios2-build-project`
- `nios2-import-project`
- `nios2-delete-project`

Workaround

Explicitly supply the Nios II IDE command-line tools with a workspace location, by means of the `-data` command-line argument. The syntax of the `-data` argument is as follows:

```
-data <path to workspace>
```

The path to the workspace must be absolute, and must not contain whitespace. Aside from these restrictions, any valid file system path can be used for the workspace. The workspace shown in Example 8 is the default workspace used by the Nios II IDE in version 9.0.

Example 8. Providing a Workspace Location

```
nios2-create-project -data c:/altera/90/nios2eds/\
    bin/eclipse/nios2-ide-workspace-9.0 <other arguments>
```

Solution Status

Not fixed

Nios II IDE Cannot Find stdio.h in Outline View

If you create a new project using the Nios II IDE and try to open the **stdio.h** file from the Outline view before building the project, the IDE displays the error message **No include files were found that matched that name**.

Workaround

Build the project before attempting to open **stdio.h**.

Solution Status

Not fixed

Nios II IDE Hangs With UNC Project Path

In the **New Project** dialog box, if you turn on **Specify Location** and specify the path in UNC form, the IDE might hang.

Workaround

Map the UNC path to a remote drive which looks like a Windows drive to the Nios II IDE.

Solution Status

Not fixed

Build Failure with Nios II Advanced Exceptions, MMU, and MPU

Projects created in the Nios II IDE for an SOPC Builder system containing a Nios II processor configured with advanced exceptions, the memory management unit (MMU), or the memory protection unit (MPU) fail to build, and generate the following error messages:

```
ERROR - Classic build flow for Nios II system library does not support
the Nios II \
    advanced exceptions.
Use the non-classic Nios II Board Support Package instead.
<timestamp> - (SEVERE) generate: java.lang.IllegalStateException: \
    java.lang.IllegalStateException: \
    com.altera.ingenuous.GTFElement.GTFElementErrorException: \
    <error> element in GTF script
make[1]: *** [system_description/../obj/system.h-t] Error 1
make: *** [system_project] Error 2
```

Workaround

Create the software project with the Nios II SBT command-line development flow. Alternatively, use the Nios II SBT for Eclipse.

Solution Status

Not fixed

Build Command Not Functional for BSPs Created With the Nios II SBT

The build option in the Nios II IDE menu does not rebuild BSPs imported to the IDE.

Workaround

The Nios II SBT for Eclipse can both build and debug projects created on the command line. The Nios II SBT for Eclipse is the preferred tool for debugging Nios II SBT projects.

For information about the Nios II SBT for Eclipse, refer to the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*.

In the Nios II IDE, you can build the BSP by building the associated application project.

Solution Status

Not fixed

Linker Errors with Dual-Port Memories

If your instruction master and data master ports are connected to the same dual-port memory and the ports have different addresses, your code fails to run or you experience a linker error. The Nios II IDE does not warn you of the addressing violation.

Workaround

Assign the same address to both ports of the dual-port memory.

Solution Status

Not fixed

User-Managed BSP Settings Not Supported in Nios II IDE

For projects created with the Nios II SBT and imported to the Nios II IDE, the IDE configuration settings have no effect.

For example, objdump, compiler, and linker settings made in the IDE are ignored. This behavior occurs because Nios II SBT projects are not IDE-managed projects. In addition, the make-related preferences do not pertain to imported Nios II SBT projects. The IDE ignores these options during the build process.

Workaround

Make these settings in the project's makefile.

Solution Status

Not fixed

URL Project Location Causes Project Creation Error

When you try to create a new project in an existing workspace, you might see an error dialog box saying:

```
Project cannot be created. Reason: Internal Error
```

This error might occur if the path to any project in the workspace is a URL location, for example **file:/F:/Design**. To view the path, right-click on the project and select **Properties**.

Workaround

Import your existing application and system library projects to a new workspace.

Solution Status

Not fixed

Compilation Error with Separate Exception Stack Option

Choosing the **Use a separate exception stack option** might cause the following compilation error when building a project:

```
"UNDEFINED VARIABLE %STACK_POINTER%"
```

This error occurs if the exception stack is larger than the memory available for it.

Workaround

On the system library properties page for the project, turn off the separate exception stack or reduce the **Maximum exception stack size** setting.

Solution Status

Not fixed

Incorrect Stack and Heap Free Space Report

The makefile reports an incorrect number of bytes free for the stack and heap, if the heap and stack are in different memory regions.

Workaround

No workaround available.

Solution Status

Not fixed

Nios II IDE Reports Problems Without Displaying Error in Console

When building a project, the Nios II IDE reports problems, but the build output in the console does not contain any errors.

The Nios II IDE incorrectly reports some linker warnings as errors, with a dialog box saying **Errors exist in a required project**. The Dhrystone software example exhibits this behavior, and recompiling the project again makes the issue go away.

Workaround

If the Console output does not contain errors, then the project built correctly. On subsequent builds, the linker step is skipped and the errors do not appear.

Solution Status

Not fixed

Nios II IDE Errata: Debugging Projects

This section describes in detail the issues related to debugging projects in the Nios II IDE.

Method for Accessing MMU and MPU Registers in the Debugger

You might observe an error if you try to read or write an MMU or MPU register via the Nios II Debugger.

Workaround

To read an MPU region, execute the following steps:

1. Set region `INDEX` in the `MPUBASE` register.
2. Exit and reenter debug mode, that is, single step.
3. Set the `RD` bit in the `MPUACC` register.
4. Exit and reenter debug mode, that is, single step.
5. Read back `MPUBASE` for pertinent information.
6. Read back `MPUACC` for pertinent information.

For a system with an MMU or MPU, this workaround allows you to read and write the current values of the registers. However, you cannot use it to control MPU regions or MMU TLB entries.

The debug core copies the Nios II processor’s register values to its internal memory when the processor enters debug mode. The debug core writes register values back to the processor only when the processor leaves debug mode. Therefore, if you attempt to set an MPU region with several consecutive values, only the last one, when leaving debug mode, is committed to the processor.

Solution Status

Not fixed

IDE Cannot Display Imported Profiling Data

If you create a software project in the Nios II SBT command line flow, generate profiling data to **gmon.out** in the Nios II command shell, and then import the project to the IDE, you cannot use the Profiling perspective to view the profiling data. When you attempt to view **gmon.out**, the IDE displays the following error message:

```
nios2-elf-gprof: ../: not in a.out format.
```

Workaround

Run `nios2-elf-gprof` from the Nios II command shell to generate a profiler report.

Solution Status

Not fixed

Error Message when Downloading .elf File

If the **Build Automatically** option is turned on in the IDE, you might get the following error when downloading an executable and linking formal file (**.elf**):

```
"2 [main] sh 5736 fork: child 3892 - died waiting; \
    for longjmp before initialization, errno 11"
```

The **Build Automatically** option is off by default, as recommended by the Eclipse help system.

Workaround

Turn off the **Build Automatically** option and download the **.elf** file again.

Solution Status

Not fixed

Variable Casting Unsupported in ISS

Casting variables in the Variables view in the Debug perspective when using the Nios II ISS might cause an exception dialog box to open.

This exception might occur if you select **Debug as ISS** in the Nios II IDE and try to cast variables via the Variables view.

Workaround

There is no workaround.

Solution Status

Not fixed

Trace Debug Does Not Support Instruction-Related Exceptions

The instruction-related exception handler is not supported by the Nios II trace tools.

Workaround

There is no workaround.

Solution Status

Not fixed

Trace Debug Does Not Support the JMPI Instruction

The JMPI instruction is not supported by the Nios II trace tools.

Workaround

There is no workaround.

Solution Status

Not fixed

Cannot Locate Source Code in Driver Files Shared by Multiple Projects

If you hit a breakpoint in a driver file, and that driver file is shared with another project that is closed, the Nios II IDE might indicate that it cannot locate the source code.

Workaround

Open the closed system library project and resume debugging.

Solution Status

Not fixed

Console Window Is Not Updated After ISS Error

After performing a **Run as ISS**, if you receive an ISS error in the console window, the console is not updated subsequently.

Workaround

Close the console window after receiving an ISS error. A new console window opens when a new message is available.

Solution Status

Not fixed

ISS Fails on Designs Containing Triple Speed Ethernet MAC or SG-DMA Components

You receive an Internal Error when attempting to perform an ISS simulation of designs containing the Altera Triple Speed Ethernet media access control (MAC) or SG-DMA components because the Nios II ISS does not support these components.

Workaround

Remove the Triple Speed Ethernet MAC and SG-DMA components from your system and perform ISS simulation on the simplified system. You can also simulate the design in ModelSim or test it on hardware.

☞ Before removing the Triple Speed Ethernet MAC and SG-DMA components, make a copy of the unmodified system to ensure that you can return to the original configuration.

Solution Status

Not fixed

Memory Window Sets Control Register Values Incorrectly

The memory window might incorrectly set values in memory-mapped control registers. For example, writing 0x1234 to a byte addressed register results in the value 0x3434 being stored in the register. The memory window shows this incorrect value.

Workaround

Use the GNU debugger (GDB) console window in the IDE, instead of the memory window, to write to the registers. For example, type the following command:

```
set {int} <register address>=0x1234↵
```

You must refresh the memory window for it to correctly display the target value.

Solution Status

Not fixed

Programs That Interact With a Terminal Console on Windows Do Not Work

Programs with this behavior work in v. 6.0 and earlier, but do not work in Nios II IDE v. 6.1 and later.

The Eclipse platform in v. 6.1 and later of the IDE (on Windows only) sends the string `\r\n` instead of just `\n` when running or debugging using the Terminal. This behavior can break existing software designs, and it is inconsistent with `nios2-terminal`, which still just sends `\n`.

Workaround

Change the software running on the Nios II processor to parse for `\r\n` as well as `\n`.

Solution Status

Not fixed

"Run as ModelSim" in the Nios II IDE Fails

The **Run as ModelSim** command might fail on launch configurations created in v. 7.0 or earlier of the IDE. This problem does not occur for new launch configurations.

Workaround

Select a location for the ModelSim tool from the launch configuration dialog box. You can use the **Browse** button next to the **ModelSim path** group, or type in a path to the ModelSim directory (for example **c:/altera/71/modelsim_ae/win32aloem**).

Solution Status

Not fixed

The Restart Command on the Run Menu Does Not Work

The **Restart** command on the **Run** menu does not work.

Workaround

Stop the program, then debug it again. If the debugger is hung in an endless loop, use the following **bash** alias to break the target, then stop it:

```
alias break="kill -2 \'ps -a | grep nios2-elf-gdb | cut -f6 -d' '\'"
```

Solution Status

Not fixed

Watchpoints Do Not Work on Certain Variables

Watchpoints do not work on a variable whose size is not 32 bits.

Workaround

Change the types of global and static local variables to `int`, `long`, or `unsigned long` before setting watchpoints on them.

Solution Status

Not fixed

Nios II IDE Errata: Navigating Projects

This section describes in detail the issues related to navigating projects in the Nios II IDE.

Nios II IDE Freezes While Displaying the Splash Screen

After clicking **Switch Workspace** on the File menu on a Windows machine, a Nios II IDE splash screen appears. Unfortunately, this splash screen obscures the dialog box that asks you to specify the new workspace. As a result, the IDE appears to freeze.

Workaround

Press Alt-Tab to switch applications. Two relevant application icons appear: an Eclipse icon associated with the splash screen and a Nios II IDE icon associated with the **Workspace** dialog box. Select the Nios II icon to bring the dialog box to the foreground.

Solution Status

Not fixed

Internal Error When Double-Clicking on a Large Objdump File

On Windows when opening a large **objdump** file in the Nios II IDE, you might get the following error message:

```
Unable to create this part due to an internal error.
Reason for the failure: Editor could not be initialized.
```

Workaround

Adjust the Windows launch arguments for the Nios II IDE editor. Perform the following steps:

1. On the Windows Start menu, browse to the **Nios II EDS** program icon, right-click it, then click **Properties**. The **Windows Properties** dialog box appears.
2. In the **Target** field, append `"vmargs -Xmx1024m"` to the end of the path to the Nios II IDE executable. For example:

```
C:\altera\72\nios2eds\bin\eclipse\nios2-ide.exe -vmargs -Xmx1024m
```

Solution Status

Not fixed

C/C++ Scanner Does Not Support Certain C/C++ Constructs

The C/C++ scanner performs C/C++ Search, navigation, open declaration, and parts of content assist. Due to limitations of the C/C++ scanner, these features do not work with C code constructs not supported by the C++ language. An example is functions that take a function pointer as an argument.

Workaround

If the C/C++ Search fails, use the File Search facility.

Solution Status

Not fixed

Nios II SBT Errata

This section describes in detail the issues in the Nios II SBT.

BSP Editor Does Not Show Command-Line Help

The following commands in the Nios II Command Shell should display command-line help:

- `nios2-bsp-editor --help`
- `nios2-bsp-editor --extended-help`

Instead, these commands launch the BSP Editor.

Workaround

For help with the BSP Editor, refer to “Using the BSP Editor” in the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer’s Handbook*.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

BSP Not Updated for Memory Size Changes in SOPC Builder

If you change the size of a memory in SOPC Builder, the memory region size in any previously created BSP is no longer correct. Regenerating the BSP does not update the BSP's memory region size.

Workaround

In the BSP Editor, on the **Linker Script** tab, you can correct the memory region size either of the following ways:

- Edit the memory region size manually.
- Click **Restore Defaults** to rerun the default Tcl script. The default Tcl script reads the updated memory region size from the SOPC information file (**.sopcinfo**) and updates the BSP.

For detailed information about keeping your BSP consistent with changes in the underlying SOPC Builder system, refer to "Revising Your BSP" in the *Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook*.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Build Errors on Interrupt API Calls

You might observe build errors if your application project installs an ISR that calls the `alt_irq_register()` function.

The Nios II EDS v9.1 includes support for an optional External Interrupt Controller (EIC) interface on the Nios II processor. To support the EIC interface, the HAL includes an enhanced interrupt API.

When you create a BSP, the SBT determines which interrupt API(s) the device drivers in your system support. If all registered drivers in the BSP identify themselves as supporting the enhanced API, the SBT implements the enhanced API. All Altera device drivers in v9.1 and later support the enhanced API.

When the enhanced API is implemented, the legacy interrupt API, including `alt_irq_register()`, is not available. Therefore, if the application code contains a call to a legacy API function, a linker error occurs.

This issue might affect application projects that call `alt_irq_register()` or other legacy interrupt API functions directly, rather than depending on drivers in the BSP for interrupt support.

If your Nios II application project's source code registers an ISR using `alt_irq_register()`, or includes an ISR, your application might fail to compile in the Nios II EDS v. 9.1, due to undefined references to `alt_irq_register()`, or to a change to the ISR function prototype.

Workaround

Modify your application code to use the new enhanced interrupt API. To use the enhanced API, you must modify several function calls and all ISR function prototypes.

For information about supporting the enhanced interrupt API, refer to the *Exception Handling* chapter of the *Nios II Software Developer's Handbook*, or to *AN 595: Vectored Interrupt Controller Usage and Applications*.

You must upgrade to the enhanced API if you wish to use an EIC in your hardware design to accelerate interrupt response time. The enhanced API also works with the Nios II processor's internal interrupt controller.

Solution Status

Not fixed

BSP Editor Incorrectly Allows Mixed Drivers

If multiple drivers are found for a single peripheral type, and you have multiple instances of the peripheral type in your SOPC Builder system, the Nios II BSP editor allows you to manually select a different driver for each instance of the peripheral or processor. For example, if you have two `altera_avalon_timer` devices, you can select two different timer drivers.

This behavior is not fully supported. You must choose the same driver for both instances of the peripheral type.

You can make driver assignments in one of the following ways:

- In the Nios II BSP editor, within the **Drivers** tab. The **Driver name** associated with each peripheral can be edited, if multiple driver types are found that support the peripheral. You must make the same driver type selection for each instance of a peripheral type in the list of peripherals as defined in the **Module Name** column.
- With the Nios II SBT in command-line mode (`nios2-bsp`, `nios2-bsp-create-settings`, `nios2-bsp-update-settings`): the **set_driver** command can be invoked (either from a Tcl script or with the `--cmd` option). The **set_driver** command accepts a peripheral module instance name and driver name as arguments. If you are using the command line development flow and wish to use this command, you must invoke it once for each instance of a peripheral type in your system, similar to the BSP editor flow described in the previous paragraph.

Workaround

There is no workaround.

Solution Status

Fixed in v. 9.1

nios2-bsp --help Displays Incorrect List of BSP Types

The Nios II SBT supports custom HAL-based operating systems, when they are added as Tcl-scripted software packages. However, the `nios2-bsp` tool help message indicates that there are only two BSP types, HAL and UCOSII. If other HAL-based operating system components are added they do not appear in the `nios2-bsp` help message.

Workaround

See "nios2-bsp-create-settings Fails to List BSP Types" to get a correct, dynamic list of available BSP types.

Solution Status

Fixed in v. 9.1

nios2-bsp-create-settings Fails to List BSP Types

The help message for `nios2-bsp-create-settings` incorrectly states that using the `--type` option with no value creates a list of available BSP types.

Workaround

To get a list of available BSP types, execute the following command:

```
nios2-bsp-create-settings --type x --settings x --sopc=<existing sopcinfo path>↵
```

The following command returns an error message that lists available BSP types:

```
SEVERE: Available BSP type values for the --type argument are: [<list of BSP types>]
```

Solution Status

Fixed in v. 8.1

Makefile Error with Absolute Source Path

You might receive the following error message from the Nios II SBT:

```
Makefile:: *** target pattern contains no '%'
```

This error message can appear if you are using Windows, you specify an absolute source path with the `--src-dir` option, and your application directory, specified with the `--app-dir` option, is the same as your source path.

Workaround

Use a relative path to the source directory, or organize your files so that your application directory is different from your source directory.

Solution Status

Fixed in v. 8.1

Incorrect Linker Errors

While building a HAL-based Nios II software project, the linker displays misleading error messages if the SOPC Builder system provides less than 32 bytes of memory at the reset vector. The linker errors are similar to the following:

```
/cygdrive/d/altera/80/nios2eds/bin/nios2-gnutools/H-i686-pc-cygwin/bin/../lib/gcc/ \
    nios2-elf/3.4.6/../../../../nios2-elf/bin/ld:/cygdrive/d/test3/software/ \
    hello_world_0_syslib/Debug/system_description/generated.x:186: warning: memory \
    region UNDEFINED not declared
/cygdrive/d/altera/80/nios2eds/bin/nios2-gnutools/H-i686-pc-cygwin/bin/../lib/gcc/ \
    nios2-elf/3.4.6/../../../../nios2-elf/bin/ld:/cygdrive/d/test3/software/ \
    hello_world_0_syslib/Debug/system_description/generated.x:186: parse error
collect2: ld returned 1 exit status
make: *** [hello_world_0.elf] Error 1
```

Workaround

In the SOPC Builder system, if the reset and exception vectors are in the same memory device, ensure that the exception vector is at least 32 bytes above the reset vector.

Solution Status

Fixed in v. 8.1

Java Heap Space Error When Creating BSP

When creating a BSP, the Nios II SBT might hang with the following message:

`SEVERE: OutOfMemoryError: Java Heap Space`.

This problem occurs when your SOPC Builder design has connection cycles. A connection cycle consists of a group of components that master one another in a circular arrangement. For example, if `ddr_sdram_bridge` masters `lcd_sgdma`, and `lcd_sgdma` masters `ddr_sdram_bridge`, the two components form a simple connection cycle. In this situation, the SBT is unable to create a BSP.

Workaround

Avoid connection cycles in the SOPC Builder system.

Solution Status

Fixed in v. 9.0

SBT Fails if Nios II EDS is Installed in a Path Containing Spaces

The Nios II Command Line SBT fails if you install the Nios II EDS in a path containing spaces.

Workaround

Reinstall the Nios II EDS to a path that does not contain spaces.

Solution Status

Not fixed

User-Managed BSP Settings Not Supported in Nios II IDE

This issue is described on page 39.

Upper-Case File Extensions Not Supported

The Nios II SBT for applications and libraries (`nios2-app-generate-makefile` and `nios2-lib-generate-makefile` commands) does not support source files with certain upper-case extensions. If a file with an upper-case extension is included, the make command stops with no descriptive warning.

Only Nios II assembly language files built by the C preprocessor can have upper-case file extension (**.S**). All C language files must have the extension **.c** or **.h**. C++ language source files must have the extension **.cpp**, **.cxx**, **.cc,** or **.h**.

Workaround

Rename all C language files with the extension **.c** or **.h**. Rename all C++ language files with the extension **.cpp**, **.cxx**, **.cc,** or **.h**.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Peripheral Errata

This section describes in detail the Nios II EDS issues related to peripheral components.

Vectored Interrupt Controller Does Not Support VHDL Simulation Models

This issue is described on page 33.

SBT Fails With UART Driver

If you use the SBT on a design containing the SOPC Builder UART peripheral, and if driver `ioctl()` support for the `altera_avalon_uart` component is enabled, then the BSP build process might fail. The **system.h** generated by the SBT contains an invalid type for the UART parameter.

Workaround

To correct **system.h**, carry out the following steps:

1. Manually edit the **system.h** file by changing the following line:

   ```
   #define UART1_PARITY "N"
   ```

   to:

   ```
   #define UART1_PARITY 'N'
   ```

   ☞ Change double quotes to single quotes.

2. Build the BSP.

Solution Status

Fixed in v. 8.1

Unaligned Transfers of Small Payloads Fail on SG-DMA

The Scatter Gather DMA SOPC Builder peripheral does not correctly handle unaligned transfers with small payloads. A payload length smaller than the data width causes erroneous data transfers.

Workaround

Avoid using DMA devices to transfer small payloads.

If absolutely necessary, for a 32-bit SG-DMA, a minimum length of 4 bytes guarantees that data is transferred correctly.

Solution Status

Not fixed

DMA Controller Always Busy in Burst Mode

The DMA controller component (`altera_avalon_dma`), when enabled for burst transactions, does not perform transfers at widths other than its full data width. The DMA controller is always busy.

Workaround

When bursting is enabled, the DMA controller must be programmed to perform transactions at its full data width.

Solution Status

Not fixed

Non-System-Wide Reset Can Cause Improper Initialization of Mailbox Core

The `altera_avalon_mailbox` peripheral might not be initialized properly when a soft (non-system-wide) reset occurs. In this condition, mailbox contents (read and write pointers) are not reinitialized and might show potentially stale data.

Workaround

Ensure that a system-wide reset event occurs by asserting the `reset_n` input to the SOPC Builder system containing the mailbox. This resets all peripherals and Nios II processors in the system.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Target Software Errata

This section describes in detail the Nios II EDS issues related to target software packages.

NicheStack tcp_wake() Function Does Not Awaken All select() Sleepers

This issue impacts software based on the NicheStack TCP/IP Stack - Nios II Edition, with multiple sleeping tasks that depend on the same `select()` event to wake up. The `tcp_sleep()` and `tcp_wake()` functions do not work properly for tasks that call `select()`.

Tasks can normally use `tcp_sleep()` and `tcp_wake()` to suspend and then awaken themselves upon specific events. The `select()` function passes an event pointer to `tcp_sleep()` that is identical for every task that might call `select()`. All other events that call `tcp_sleep()` pass it a unique event pointer that specifically identifies them as the task to be awakened at the next event.

The `tcp_wake()` function apparently assumes that the list of sleepers contains unique event pointers by which they can be identified. Therefore, when a `select()` event occurs, instead of awakening all tasks waiting for it, `tcp_wake()` awakens only the first task in the list. Therefore, when more than one task calls `select()` to await some outcome on a socket, only the highest-priority task in the sleepers list is awakened at `select()` events. Other sleeping tasks that depend on the same `select()` event do not wake up.

Workaround

There is no workaround.

Solution Status

This issue will be fixed in future release of the Nios II EDS.

EIC-Based Nios II System Crashes With NicheStack

When you attempt to run software based on the NicheStack TCP/IP Stack - Nios II Edition on a hardware design incorporating an external interrupt controller such as the VIC, the Nios II system crashes during initialization, with unpredictable results.

Workaround

There is no workaround.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Gigabit Ethernet Performance Issues

The NicheStack TCP/IP Stack - Nios II Edition might hang while sending transmission control protocol (TCP) packets to the host, or experience dropped packets when receiving TCP packets from the host. These issues can arise with version 3.1 of the NicheStack TCP/IP Stack, using the triple-speed Ethernet component to transmit over a Gigabit Ethernet point-to-point connection.

Workaround

Download and install the patches at www.altera.com/support/kdb/solutions/rd01132009_588.html.

Alternatively, upgrade to the Nios II EDS v. 9.0 or later.

Solution Status

Fixed in v. 9.0

stdio Does Not Work with MicroC/OS-II and Small C Library

stdin, stdout, and stderr do not work in MicroC/OS-II applications built with the Small C library option.

Workaround

Disable the small C library option.

Solution Status

Not fixed

cout From MicroC/OS-II Task Does Not Send Data to stdout

If neither printf() or cout is used from main() before tasks are started, cout does not work from a task.

Workaround

Add the following C++ code to the beginning of main():

```
std::ios_base::sync_with_stdio(false);
```

Solution Status

Not fixed

malloc(), realloc() Failures With MicroC/OS-II

When you use the MicroC/OS-II RTOS, calls to malloc() and realloc() might fail if successive calls to malloc() or realloc() within a MicroC/OS-II task occur after changing the task priority of the task in which a memory block is originally allocated.

Workaround

Use one of the following workarounds:

- Allocate and/or reallocate memory blocks outside of MicroC/OS-II tasks, before task switching starts. Preallocated memory blocks make it possible to change thread priorities at runtime.
- Allocate fixed areas of memory using arrays (rather than using `malloc()`) before task switching starts. Fixed memory arrays make it possible to change thread priorities at runtime.
- Allocate memory using `malloc()` or `realloc()` from a MicroC/OS-II task. You can change task priorities at runtime, but only for tasks that have not used `malloc()` or `realloc()`.

Solution Status

Not fixed

Toolchain Errata

This section describes in detail the Nios II EDS issues related to the Nios II and GNU compiler toolchains, including tools such as **gcc**, **gdb**, and **sof2flash**.

Unable to Configure FPGA from Flash with Parallel Flash Loader

This issue is described on page 31.

sof2flash Names Output Files Incorrectly

If **sof2flash** is called with the `--save` option and a relative path is specified for the output file (that is, including "./" or "../"), the raw binary file that is saved is incorrectly named **.rbf** instead of **<flash filename>.rbf**.

Workaround

Provide a full path to the output file(s), or change to the output directory before running the **sof2flash** command so the output file specification is `--output=<`*`filename`*`>.flash`.

Solution Status

Fixed in v. 8.1

GNU Assembler Does Not Accept the --defsym Flag

According the GNU documentation, you can set an assembler definition by using the `--defsym` flag, but it does not work in the following form: `--defsym MY_VAR=1`.

Workaround

There is no workaround.

Solution Status

Not fixed

How to Contact Altera

For the most up-to-date information about Altera products, refer to Table 8.

Table 8. Contact Information

Contact *(1)*	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Altera literature services	Email	literature@altera.com
Non-technical support (General)	Email	nacomp@altera.com
(Software Licensing)	Email	authorization@altera.com

Note:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

Table 9 shows the typographic conventions that this document uses.

Table 9. Typographic Conventions (Part 1 of 2)

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Indicates command names, dialog box titles, dialog box options, and other graphical user interface (GUI) labels. For example, **Save As** dialog box. For GUI elements, capitalization matches the GUI.
bold type	Indicates directory names, project names, disk drive names, file names, file name extensions, dialog box options, software utility names, and other GUI labels. For example, **\qdesigns** directory, **d:** drive, and **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Indicates document titles. For example, *AN 519: Stratix IV Design Guidelines.*
Italic type	Indicates variables. For example, *n* + 1. Variable names are enclosed in angle brackets (< >). For example, <*file name*> and <*project name*>**.pof** file.
Initial Capital Letters	Indicates keyboard keys and menu names. For example, Delete key and the Options menu.
"Subheading Title"	Quotation marks indicate references to sections within a document and titles of Quartus II Help topics. For example, "Typographic Conventions."
`Courier type`	Indicates signal, port, register, bit, block, and primitive names. For example, `data1`, `tdi`, and `input`. Active-low signals are denoted by suffix `n`. For example, `resetn`. Indicates command line commands and anything that must be typed exactly as it appears. For example, `c:\qdesigns\tutorial\chiptrip.gdf`. Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword `SUBDESIGN`), and logic function names (for example, `TRI`).
1., 2., 3., and a., b., c., and so on.	Numbered steps indicate a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ■	Bullets indicate a list of items when the sequence of the items is not important.

Table 9. Typographic Conventions (Part 2 of 2)

Visual Cue	Meaning
☞	The hand points to information that requires special attention.
⚠ CAUTION	A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.
⚠ WARNING	A warning calls attention to a condition or possible situation that can cause you injury.
↵	The angled arrow instructs you to press Enter.
👣	The feet direct you to more information about a particular topic.

101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support
www.altera.com/support

Nios II Embedded Design Suite Release Notes and Errata

© February 2010 RN-EDS-6.0

About These Release Notes

These release notes cover versions 8.1 through 9.1 SP1 of the Altera® Nios® II Embedded Design Suite (EDS). These release notes describe the revision history and errata for the Nios II EDS.

Errata are functional defects or errors, which might cause the product to deviate from published specifications. Documentation issues include errors, unclear descriptions, or omissions from current published specifications or product documents.

For the most up-to-date errata for this release, refer to the latest version of the *Nios II Embedded Design Suite Release Notes and Errata* on the Altera website. You can refer to release notes and errata for the Nios II processor core in the *Nios II Processor* chapter of the *MegaCore IP Library Release Notes and Errata*.

Revision History

Table 1 shows the revision history for the Nios II EDS, including the Nios II C2H Compiler.

Table 1. Nios II Embedded Design Suite Revision History

Version	Date	Description
9.1 SP1	January 2010	Maintenance release
9.1	November 2009	■ The Nios II Software Build Tools (SBT) for Eclipse, an integrated development environment based on Eclipse. Recommended for all new Nios II projects. ■ The Nios II Integrated Development Environment (IDE) is available in the Nios II EDS, but not recommended for new projects. ■ The Nios II Studio is discontinued. ■ The vectored interrupt controller (VIC) core offers high-performance, low-latency interrupt handling. ■ Nios II Design Examples with memory management unit (MMU): ■ Nios II 3C120 Hardware Reference Design with MMU, Cyclone® III Edition ■ Nios II 4S230 Hardware Reference Design with MMU, Stratix® IV GX Edition Commercial and open-source support for embedded Linux is available for these systems. ■ The Embedded Systems Lab for DE1 for Quartus® and Nios II EDS v9.1 is available for download from the Nios Community Wiki (www.nioswiki.com). ■ Tcl scripting for software packages and drivers enhanced to support callback functions.
9.0	March 2009	Added the `nios2-convert-ide2sbt` tool, which converts projects from the Nios II IDE development flow to the Nios II SBT development flow
8.1 SP1	January 2009	Nios Community Wiki URL changing (www.nioswiki.com)
8.1	November 2008	Maintenance release

For more information on new features and changes, refer to the *Nios II Processor Reference Handbook*, the *Nios II Software Developer's Handbook*, and the *Nios II C2H Compiler User Guide*. For information about changes to the Nios II processor core, refer to the *Nios II Processor* chapter of the *MegaCore IP Library Release Notes and Errata*.

Revision Details

This section provides details about some of the Nios II EDS revisions shown in Table 1.

Nios II Software Build Tools for Eclipse

The Nios II Software Build Tools (SBT) for Eclipse is a set of plugins based on the popular Eclipse framework and the Eclipse C/C++ development toolkit (CDT) plugins. The Nios II SBT for Eclipse provides a consistent development platform that works for all Nios II processor systems. You can accomplish most Nios II software development tasks within Eclipse, including creating, editing, building, running, debugging, and profiling programs. The Nios II SBT for Eclipse is built from the ground up as a set of plugins to standard Eclipse. The Nios II SBT for Eclipse is based on the Nios II Software Build Tools, a set of powerful commands, utilities and scripts for creating applications, board support packages and software libraries, and managing build options.

Table 2 compares the Nios II Integrated Development Environment (IDE) with the Nios II SBT for Eclipse.

Table 2. Differences Between Nios II SBT for Eclipse and Nios II IDE (Part 1 of 2)

Feature	Nios II IDE	Nios II SBT for Eclipse
Types of projects	■ Application ■ Board support package (BSP)—Called a system library in the IDE ■ User Library (optional)	■ Application ■ BSP ■ User Library (optional)
Handling updates to the SOPC Builder system	Updates the BSP automatically	You must initiate the BSP update (in the Nios II menu, click Generate BSP)
Options for creating projects	Application from template. BSP is created automatically as part of application.	■ New Application and BSP from template ■ Standalone application ■ Standalone BSP ■ Standalone library
Handling updates to BSP	Entire BSP is rebuilt. This process can be time-consuming.	Smart rebuild only recompiles changed files
Compatibility with command-line flow	None	Full compatibility with the Nios II SBT in the Nios II Command Shell
Modifying BSP properties	Limited control over BSP properties through the System Library page of the Properties dialog box.	Full control over BSP properties through the BSP Editor

Table 2. Differences Between Nios II SBT for Eclipse and Nios II IDE (Part 2 of 2)

Feature	Nios II IDE	Nios II SBT for Eclipse
Nios II C2H Compiler	Supported	Not supported
Nios II Flash Programmer	Limited control over the contents of the flash image	■ Allows you to create complex flash images containing Executable and Linking Format Files (.elf), SRAM Object Files (.sof), and binary data files in specified locations ■ Allows you to save flash programmer settings for repeatable flash programming
Scripting and regression testing	Not Available	Project creation and build can be scripted using the SBT command line

Converting IDE Projects to the SBT for Eclipse

If you have a pre-existing Nios II IDE project, before you can continue development in the Nios II SBT for Eclipse, you must convert it to the SBT and import it to the Eclipse environment. To convert and import an IDE project to the Nios II SBT for Eclipse, execute the following steps:

1. Convert the IDE project to an SBT-based project with the **nios2-convert-ide2sbt** command line utility.
2. Import the converted project to the Nios II SBT for Eclipse.

For a detailed description of the conversion process, refer to "Porting Nios II IDE Projects to the Software Build Tools" in the *Using the Nios II Integrated Development Environment* appendix of the *Nios II Software Developer's Handbook*. For a description of the process of importing a project, refer to "Importing a Command-Line Project" in the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*.

Learning about the Nios II SBT for Eclipse

Refer to the following documents to help you get acquainted with the Nios II SBT for Eclipse and related tools:

- *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*
- *Nios II Hardware Development Tutorial*
- The Comprehensive Nios II Embedded Design Suite support page
- The Nios II Software Build Tools for Eclipse support page

Errata Summary

Table 3 summarizes the issues that affect the Nios II Embedded Design Suite versions 8.1 through 9.1. For a detailed description of each issue, click on the issue name.

Refer to the Altera Knowledge Database for older errata and solutions.

Table 3. Nios II EDS Errata *(1)* (Part 1 of 5)

Added or Updated	Issue	Page	Affected Version			
			9.1 SP1	9.1	9.0	8.1
	Nios II SBT for Eclipse					
Feb 2010	Build Errors on Software for Pre-Existing Design with SG-DMA	8	✓	✓	—	—
	Spurious System ID Mismatch Error	8	✓	✓	—	—
	Errors Debugging as Local C/C++ Application	9	✓	✓	—	—
Nov 2009	Error Messages on Console When Debugging	9	✓	✓	—	—
	java.lang.NullPointerException Error When Running Application	9	✓	✓	—	—
	Update Failure When Setting Empty Properties in the Nios II Properties Page	10	✓	✓	—	—
	Errors Converting Nios II IDE Multiprocessor Project	10	✓	✓	—	—
	DHCP Time-Out on EP3C120	10	✓	✓	—	—
	No Nios II Multiprocessor Collection Run Configuration	11	✓	✓	—	—
	Stop on Startup Option in Run Configuration Has No Effect	11	✓	✓	—	—
	Project Imported From Command Line Fails to Build Correctly	12	✓	✓	—	—
	Debugger Breaks in crt0.s Instead of main()	12	✓	✓	—	—
	C2H Compiler					
Nov 2009	Error "java.lang.NoClassDefFoundError" During C2H Compilation	12	—	—	Fixed	✓
	C2H Cache Coherency Issue with Global Variables	13	—	—	—	Fixed
Jul 2008	Functions Declared Without a Return Type Are Not Supported	14	✓	✓	✓	✓
Oct 2007	Pre-7.1 Systems Are Not Supported	14	✓	✓	✓	✓
	--src-dir SBT Argument Does Not Work With C2H	14	✓	✓	✓	✓
May 2007	Accelerator Generation Failure If Tools Are Installed in Path With Spaces	14	✓	✓	✓	✓
	The C2H Compiler Regenerates an Accelerator Unnecessarily	15	✓	✓	✓	✓
	Error: c2h_accelerator_base_addresses.h: No such file or directory	15	✓	✓	✓	✓
	Java Heap Space Exception if Quartus II Compilation is Enabled	15	✓	✓	✓	✓
	Pointer Dereferences to Volatile Types	16	✓	✓	✓	✓
Dec 2006	C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File	16	✓	✓	✓	✓

Table 3. Nios II EDS Errata *(1)* (Part 2 of 5)

Added or Updated	Issue	Page	Affected Version			
			9.1 SP1	9.1	9.0	8.1
Jun 2006	Array Elements in Structures Do Not Copy Correctly	16	✓	✓	✓	✓
	Clean Build Causes Build Failure	17	✓	✓	✓	✓
	Changing Build Configurations Produces Unexpected Results	17	✓	✓	✓	✓
	Hardware Accelerators Remain After Deleting the Software Project *(2)*	18	✓	✓	✓	✓
	Incorrect Results From Logical or Conditional Operation With Side-Effects	18	✓	✓	✓	✓
	Launch SOPC Builder Button in C2H View	19	✓	✓	✓	✓
Development Boards						
Jun 2006	Intermittent Failures While Accessing CompactFlash Card	19	✓	✓	✓	✓
Documentation Issues						
Feb 2010	Missing Documentation of Interrupt API Properties	20	✓	✓	—	—
	Valid Range of hal.log_flags is –1 to 3	21	✓	✓	—	—
Nov 2009	NicheStack TCP/IP Stack - Nios II Edition Does Not Support TFTP	21	✓	✓	✓	✓
	Error Message After Renaming Project: “Resource is out of sync with the system”	21	✓	✓	—	—
	Obsolete Wiki URLs in Help Systems and Handbook	22	—	—	Fixed	✓
	Missing Source Control Information	22	—	—	Fixed	✓
Mar 2009	Compiler Flags for Building Custom Newlib	23	✓	✓	✓	✓
Nov 2008	Error Message on Invalid Exception or Reset Vector	24	✓	✓	✓	✓
Oct 2007	Nios II IDE Online Help Expand Buttons Do Not Work	25	✓	✓	✓	✓
Hardware Abstraction Layer						
Feb 2010	Missing Structure Member Errors in alt_log_printf.c with Small JTAG UART Driver	26	✓	✓	—	—
	“unused variable” Warning When Building alt_main.c	26	✓	✓	—	—
	Build Warnings in Performance Counter Driver Files	27	✓	✓	—	—
Hardware Example Designs						
Mar 2007	Incorrect PLL Phase Shift in Stratix II and Cyclone II Designs	27	✓	✓	✓	✓
Software Examples						
Nov 2009	Hardware Tutorial Software Example Hangs on Some Boards	28	✓	✓	✓	✓
Nov 2008	Simple Socket Server Example Displays Incorrect IP Address	29	✓	✓	✓	✓
Jun 2006	Networking Examples	29	✓	✓	✓	✓
Flash Programmer						
Feb 2010	Unable to Configure FPGA from Flash with Parallel Flash Loader	29	Fixed	✓	✓	—
	Boot Failure or elf2flash Error Due to Incorrect Flash File Size	30	✓	✓	✓	✓
May 2007	Delay When Creating New Flash Programmer Configuration	31	✓	✓	✓	✓

Table 3. Nios II EDS Errata *(1)* (Part 3 of 5)

Added or Updated	Issue	Page	Affected Version			
			9.1 SP1	9.1	9.0	8.1
Dec 2006	elf2flash File Size Limit	31	✓	✓	✓	✓
	Error Message “No such file or directory” When Programming Flash	31	✓	✓	✓	✓
Hardware Simulation						
Dec 2006	Error “UNC paths are not supported” Launching ModelSim	32	✓	✓	✓	✓
Jun 2006	Uninitialized .bss Variables in Simulation	32	✓	✓	✓	✓
Host Platform						
Nov 2009	Cannot Launch Nios II SBT for Eclipse on Red Hat Linux	32	✓	✓	—	—
Dec 2007	Windows/Cygwin: Nios II Processor Generation Failure	33	✓	✓	✓	✓
Oct 2007	Windows Vista: Limited Support in Nios II EDS	33	✓	✓	✓	✓
Nios II IDE						
Building Projects						
Mar 2009	Nios II IDE Command-Line Tools Select Wrong Workspace on Linux	34	✓	✓	✓	✓
	Nios II IDE Command-Line Tools Hang on Windows	35	✓	✓	✓	✓
Nov 2008	Nios II IDE Cannot Find stdio.h in Outline View	35	✓	✓	✓	✓
Jul 2008	Nios II IDE Hangs With UNC Project Path	35	✓	✓	✓	✓
	Build Failure with Nios II Advanced Exceptions, MMU, and MPU	36	✓	✓	✓	✓
May 2007	Build Command Not Functional for BSPs Created With the Nios II SBT	36	✓	✓	✓	✓
	Linker Errors with Dual-Port Memories	37	✓	✓	✓	✓
	User-Managed BSP Settings Not Supported in Nios II IDE	37	✓	✓	✓	✓
	URL Project Location Causes Project Creation Error	37	✓	✓	✓	✓
Dec 2006	Compilation Error with Separate Exception Stack Option	37	✓	✓	✓	✓
	Incorrect Stack and Heap Free Space Report	38	✓	✓	✓	✓
	Nios II IDE Reports Problems Without Displaying Error in Console	38	✓	✓	✓	✓
Debugging Projects						
Nov 2009	Method for Accessing MMU and MPU Registers in the Debugger	38	✓	✓	✓	✓
Nov 2008	IDE Cannot Display Imported Profiling Data	39	✓	✓	✓	✓
Jul 2008	Error Message when Downloading .elf File	39	✓	✓	✓	✓
	Variable Casting Unsupported in ISS	40	✓	✓	✓	✓
	Trace Debug Does Not Support Instruction-Related Exceptions	40	✓	✓	✓	✓
Oct 2007	Trace Debug Does Not Support the JMPI Instruction	40	✓	✓	✓	✓
May 2007	Cannot Locate Source Code in Driver Files Shared by Multiple Projects	40	✓	✓	✓	✓
	Console Window Is Not Updated After ISS Error	40	✓	✓	✓	✓
	ISS Fails on Designs Containing Triple Speed Ethernet MAC or SG-DMA Components	41	✓	✓	✓	✓
	Memory Window Sets Control Register Values Incorrectly	41	✓	✓	✓	✓

Table 3. Nios II EDS Errata *(1)* (Part 4 of 5)

Added or Updated	Issue	Page	Affected Version			
			9.1 SP1	9.1	9.0	8.1
Dec 2006	Programs That Interact With a Terminal Console on Windows Do Not Work	41	✓	✓	✓	✓
	“Run as ModelSim” in the Nios II IDE Fails	42	✓	✓	✓	✓
	The Restart Command on the Run Menu Does Not Work	42	✓	✓	✓	✓
Jun 2006	Watchpoints Do Not Work on Certain Variables	42	✓	✓	✓	✓
Navigating Projects						
Dec 2006	Nios II IDE Freezes While Displaying the Splash Screen	43	✓	✓	✓	✓
	Internal Error When Double-Clicking on a Large Objdump File	43	✓	✓	✓	✓
Jun 2006	C/C++ Scanner Does Not Support Certain C/C++ Constructs	43	✓	✓	✓	✓
Nios II SBT						
Feb 2010	BSP Not Updated for Memory Size Changes in SOPC Builder	44	✓	✓	—	—
Nov 2009	Build Errors on Interrupt API Calls	44	✓	✓	—	—
	BSP Editor Incorrectly Allows Mixed Drivers	45	Fixed	Fixed	✓	✓
	nios2-bsp --help Displays Incorrect List of BSP Types	46	Fixed	Fixed	✓	✓
	nios2-bsp-create-settings Fails to List BSP Types	46	—	—	—	Fixed
	Makefile Error with Absolute Source Path	46	—	—	—	Fixed
	Incorrect Linker Errors	47	—	—	—	Fixed
	Java Heap Space Error When Creating BSP	47	—	—	Fixed	✓
Oct 2007	SBT Fails if Nios II EDS is Installed in a Path Containing Spaces	47	✓	✓	✓	✓
May 2007	User-Managed BSP Settings Not Supported in Nios II IDE	37	✓	✓	✓	✓
	Upper-Case File Extensions Not Supported	48	✓	✓	✓	✓
Peripherals						
Nov 2009	SBT Fails With UART Driver	48	—	—	—	Fixed
Jul 2008	Unaligned Transfers of Small Payloads Fail on SG-DMA	49	✓	✓	✓	✓
May 2007	DMA Controller Always Busy in Burst Mode	49	✓	✓	✓	✓
	Non-System-Wide Reset Can Cause Improper Initialization of Mailbox Core	49	✓	✓	✓	✓
Target Software						
Nov 2009	NicheStack tcp_wake() Function Does Not Awaken All select() Sleepers	50	✓	✓	—	—
	EIC-Based Nios II System Crashes With NicheStack	50	✓	✓	—	—
Mar 2009	Gigabit Ethernet Performance Issues	50	—	—	Fixed	✓
Dec 2006	stdio Does Not Work with MicroC/OS-II and Small C Library	51	✓	✓	✓	✓
Jun 2006	cout From MicroC/OS-II Task Does Not Send Data to stdout	51	✓	✓	✓	✓
	malloc(), realloc() Failures With MicroC/OS-II	51	✓	✓	✓	✓

Table 3. Nios II EDS Errata (1) (Part 5 of 5)

Added or Updated	Issue	Page	Affected Version			
			9.1 SP1	9.1	9.0	8.1
	Toolchain (gcc, gdb, etc.)					
Feb 2010	Unable to Configure FPGA from Flash with Parallel Flash Loader	29	Fixed	✓	✓	—
Nov 2009	sof2flash Names Output Files Incorrectly	52	—	—	—	Fixed
Dec 2006	GNU Assembler Does Not Accept the --defsym Flag	52	✓	✓	✓	✓

Notes to Table 3:

(1) Refer to the Altera Knowledge Database for older errata and solutions.

(2) For further information, refer to the *Nios II C2H Compiler User Guide*.

Nios II SBT for Eclipse Errata

This section describes in detail the issues related to the Nios II SBT for Eclipse.

Build Errors on Software for Pre-Existing Design with SG-DMA

If you have a Nios II system generated with SOPC Builder v. 9.0 or earlier, and it contains the Scatter-Gather DMA (SG-DMA) component, you cannot build software for it with the Nios II SBT for Eclipse v. 9.1 or later. If you attempt to do so, the Nios II compiler reports errors.

Workaround

Regenerate the design with SOPC Builder v. 9.1 or later before attempting to compile the software in the Nios II SBT for Eclipse.

Solution Status

Not fixed.

Spurious System ID Mismatch Error

You might see the following error message when attempting to run or debug a Nios II software project, or to edit a launch configuration:

[Target Connection]:System ID mismatch - actual: "0xffffffff", expected: "<*system id*>"

<*system id*> is replaced by the system ID in your software project. When this error occurs, you cannot run or debug software.

This error normally means that the FPGA is unconfigured, or configured with a **.sof** file that does not correspond to your software project. However, the error occasionally appears even when the FPGA is configured with the correct **.sof**.

Workaround

After verifying that the FPGA is configured with the correct **.sof**, you can use one of the following methods to work around this issue:

- In the **Run Configuration** or **Debug Configuration** dialog box, on the **Target Connection** tab, click **Refresh Connections** repeatedly until the error message disappears.
- Delete the launch configuration and create a new launch configuration.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Errors Debugging as Local C/C++ Application

If you try to debug a Nios II software project as the Local C/C++ Application launch configuration type, you see one of the following errors, depending on your platform:

- Windows: "Error creating session"
- Linux: "Execution is suspended because of error"

The Nios II debug perspective fails to open.

This is expected CDT behavior in the Eclipse platform. Local C/C++ Application is the launch configuration type for a standard CDT project. To invoke the Nios II plugins, you must use a Nios II launch configuration type.

Workaround

Always debug Nios II software projects as the Nios II Hardware launch configuration type.

Error Messages on Console When Debugging

When debugging, you might see the following messages in the Nios II SBT for Eclipse console:

```
No symbol "auto" in current context.
No symbol "new" in current context.
```

These are normal messages from the GNU Debugger (GDB) console, not error messages. This issue does not impact the debug process.

Workaround

No workaround needed.

Solution Status

None needed.

java.lang.NullPointerException Error When Running Application

When you attempt to run or debug an application, the Nios II SBT for Eclipse might show an internal error message similar to the following:

```
Launching New_configuration java.lang.NullPointerException
```

This error occurs in either of the following circumstances:

- You attempt to run your project without rebuilding after performing a `make clean`.
- You delete a run configuration belonging to your project while the project is running.

You must restart Nios II SBT for Eclipse to recover from this error.

Workaround

Build the project before attempting to run or debug it. Do not delete a run configuration while it is in use.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Update Failure When Setting Empty Properties in the Nios II Properties Page

If you use the Nios II Properties page to update more than one board support package (BSP) setting to the empty string (`""`), the BSP is not updated.

Workaround

You can use either of the following workarounds to set multiple BSP settings to the empty string:

- Use the keyword `none` instead of `""` to set settings to the empty string.
- Use the BSP Editor instead of the Nios II Properties page.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Errors Converting Nios II IDE Multiprocessor Project

If you try use the **nios2-convert-ide2sbt** utility to convert an IDE multiprocessor project to the Nios II SBT, error messages appear, and the project fails to generate properly.

Workaround

Create new Nios II SBT application and BSP projects using project settings equivalent to the original Nios II IDE project settings.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

DHCP Time-Out on EP3C120

If you create a project based on the web server template targeting the Altera Embedded Systems Development Kit, Cyclone® III Edition (EP3C120) board, the system is unable to get an IP address at startup. DHCP times out and the system reverts to the default IP address.

Workaround

Create the project in the Nios II Command Shell and then import it to the Nios II SBT for Eclipse, by executing the following steps:

1. In the Nios II Command Shell, navigate to the following path:

   *<Nios II EDS install path>***/examples/***<language>***/niosII_cycloneIII_3c120/tserd/software_examples/app/web_server_tse_3c120_rgmii**

   where *<language>* is either **vhdl** or **verilog**.

2. Type the following command to create a web server example for the EP3C120 kit:

   ```
   create-this-app↵
   ```

3. Import the command-line project to the Nios II SBT for Eclipse.

For information about importing command-line projects, refer to "Importing a Command-Line Project" in the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

No Nios II Multiprocessor Collection Run Configuration

The multiprocessor collection run configuration is not available in the Run menu. Therefore, you cannot use a single run configuration to download **.elf** files to run on multiple Nios II processors in the Nios II SBT for Eclipse.

Workaround

Create a separate run configuration for each Nios II processor in the hardware design, and launch them one at a time

Solution Status

Not fixed

Stop on Startup Option in Run Configuration Has No Effect

Turning on **Stop on Startup at** in the **Debugger** tab of a run configuration has no effect.

Workaround

You can manually set any breakpoint in the debug perspective.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Project Imported From Command Line Fails to Build Correctly

If you try to import a command-line project to the Nios II SBT for Eclipse, and your project has source files located outside the project folder, the Nios II SBT for Eclipse fails to build your project because the makefile does not include source files located outside the project folder.

Workaround

You can work around this issue with either of the following methods:

- When importing the project, disable source management through the **Nios II Application Properties** or **Nios II Library Properties** dialog box.

  ☞ You must manually keep your makefile updated to reflect source files added to or removed from the project.

- Manually link the affected source files to the project. Right-click the project, click **Properties**, expand **C/C++ General**, select **Paths and Symbols** and then **Source Location**, and click **Link Folder**.

Solution Status

This issue will be fixed in future release of the Nios II EDS.

Debugger Breaks in crt0.s Instead of main()

If you attempt to launch a debug session by right clicking the project, without first creating a run configuration, the debugger breaks in the initialization code in **crt0.s**, rather than at the top of the `main()` function. You must explicitly create the run configuration before launching the debug session.

Workaround

You can work around this issue with either of the following methods:

- Create the run configuration with the **Run Configuration** dialog box. You can then debug with this run configuration by right-clicking the project, clicking **Debug As**, and selecting the debug configuration. The program correctly breaks at `main()`.
- Manually set a breakpoint at `main()`.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

C2H Compiler Errata

This section describes in detail the issues related to the Nios II C2H Compiler.

Error "java.lang.NoClassDefFoundError" During C2H Compilation

C2H compilation fails with the following error:

```
java.lang.NoClassDefFoundError: com/altera/c2h/C2Hgenerate
Exception in thread "main" make: *** [c2h_hdl-t] Error 1
```

This error occurs when you invoke C2H from the Nios II IDE on the Windows operating system. This problem does not affect Linux installations, nor C2H compilation from the Nios II command shell.

Workaround

To correct the problem, visit mySupport and request the Nios II EDS 8.1 Patch 0.01 (PC version).

Solution Status

Fixed in v. 9.0

C2H Cache Coherency Issue with Global Variables

Cache coherency issues might be encountered when the following conditions are all true:

- A C2H-accelerated function writes to a global or static variable
- The Nios II processor is configured with a data cache
- The referenced global variable is located in cacheable memory

In this situation, the data in the global variable can be corrupted.

This issue can only arise if the accelerated function writes to the global or static variable. If the function merely reads from the variable, cache coherency is maintained correctly.

Workaround

Add a data cache flush operation in the accelerator's wrapper function. The wrapper can be found in **alt_c2h_<*function_name*>.c**. In this file, find the following comment:

```
/* Write 1 to address 0 starts the accelerator */
```

Directly after this comment, add the following function call:

```
alt_dcache_flush_all();
```

This function call ensures that the data cache is flushed immediately before the accelerator is started.

☞ Flushing the cache entails a performance penalty.

Alternatively, allocate your global and static variables in non-cacheable regions of memory. Placing these variables in non-cacheable regions eliminates any need to perform a data cache flush.

For information about allocating variables to non-cacheable regions, refer to the *Cache and Tightly-Coupled Memory* chapter of the *Nios II Software Developer's Handbook*.

Solution Status

Fixed in v. 8.1

Functions Declared Without a Return Type Are Not Supported

The C2H compiler does not support functions without an explicitly declared return type.

Workaround

If you are using the implicit `int` return type, declare the return type explicitly. If your function has no return value, declare it as `void`.

Solution Status

Not fixed

Pre-7.1 Systems Are Not Supported

Starting in v. 7.1, the C2H Compiler does not support SOPC Builder systems created in SOPC Builder 7.0 or earlier (systems based on a **class.ptf** file).

Workaround

Before using the C2H Compiler, you must upgrade your SOPC Builder system. Open the system in SOPC Builder v. 7.1 or later. SOPC Builder prompts you to convert the system, creating a ***_hw.tcl** file compatible with the C2H Compiler.

Solution Status

Not fixed

--src-dir SBT Argument Does Not Work With C2H

When using the C2H Compiler with Nios II SBT, an error occurs when using the `nios2-app-generate-makefile` argument `--src-dir` to specify a directory of source files. An example of the error is:

```
multiple definition of 'my_accelerated_function'
```

Workaround

Specify source files individually using the `--src-files` argument.

Solution Status

Not fixed

Accelerator Generation Failure If Tools Are Installed in Path With Spaces

If the path to your installation of the Altera Design Suite contains spaces, the C2H Compiler fails to generate the accelerators.

Workaround

Reinstall the tools to a path containing no spaces.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

The C2H Compiler Regenerates an Accelerator Unnecessarily

The C2H Compiler might regenerate an accelerator even when the accelerated function is unchanged. This problem can result from changes to files included by the C file containing the accelerated function. The C2H Compiler fails to check that the generated hardware description language (HDL) matches the previously generated HDL causing the system to be regenerated.

Workaround

To avoid this issue move the accelerator-specific information from the include file to a separate include file. This workaround prevents regeneration of the system when the HDL is unchanged.

Solution Status

Not fixed

Error: c2h_accelerator_base_addresses.h: No such file or directory

When a C2H accelerator is compiled for the first time, the following compile-time error can result if the **Analyze all accelerators** option is selected:

```
c2h_accelerator_base_addresses.h: No such file or directory.
```

Workaround

Click **Build software and generate SOPC Builder system** and build once before building with the **Analyze all accelerators** option.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Java Heap Space Exception if Quartus II Compilation is Enabled

If your design contains a C2H accelerator, and you select **Build software, generate SOPC Builder system, and run Quartus® II compilation** in the C2H view, you might see the following error during Quartus II compilation:

```
Exception in thread "main" java.lang.OutOfMemoryError: Java
heap space

make: *** [c2h_hdl-t] Error 1
```

Workaround

Select the **Build software and generate SOPC builder system** option in the C2H Compiler settings window, and then manually launch the Quartus II software to compile the design.

Solution Status

Not fixed

Pointer Dereferences to Volatile Types

The C2H Compiler treats pointer dereferences to a volatile type as if they alias all other pointer dereferences. Pointers that are restrict-qualified are treated the same way.

The two loops in Example 1 cannot be scheduled concurrently because the `volatile` qualification overrides the `__restrict__` pragma.

Example 1. Non-Concurrent Loops

```
volatile int * __restrict__ fifo_rd = FIFO_RD_BASE;
volatile int * __restrict__ fifo_wr = FIFO_WR_BASE;
for ()
{
  *fifo_wr = ....;
}
for ()
{
  ... = *fifo_rd;
}
```

Workaround

Divide the function into multiple interrupt request (IRQ)-enabled accelerators that are launched concurrently from the processor, and use FIFO buffers to communicate between them.

Solution Status

Not fixed

C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File

When accelerating a function in a file, the C2H Compiler cannot link subfunctions that are defined in a different file.

Workaround

Include all subfunctions called by the accelerated function within the same source code file.

Solution Status

Not fixed

Array Elements in Structures Do Not Copy Correctly

C2H accelerators do not correctly copy array elements that are elements of structures.

In Example 2, the `a` and `b` elements of the structure copy correctly, but the `buf` element does not. After this assignment, `struct_a` equals `{9, 8, {3, 3, 3, 3}}`.

Example 2. Array Elements of Structs

```
typedef struct my_struct {
  int a;
  int b;
  int buf[BUF_SIZE];
}MY_STRUCT;
MY_STRUCT struct_a = {1, 2, {3, 3, 3, 3}};
MY_STRUCT struct_b = {9, 8, {7, 7, 7, 7}};
struct_a = struct_b;
```

Workaround

Copy the array elements explicitly, as shown in Example 3.

Example 3. Copying Array Elements Explicitly

```
{
  int i=0;
  do
    {
      struct_a.buf[i] = struct_b.buf[i];
      i++;
    } while (i<LENGTH_OF_BUF_ELEMENT)
}
```

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Clean Build Causes Build Failure

Performing a clean build on a Nios II IDE project that contains a hardware accelerator can cause the next build to fail in the IDE, because the clean build erroneously deletes a file required by the C2H Compiler.

Workaround

Do not perform a clean build on projects that use hardware accelerators. If you have already performed a clean build, recompile with option **Build software, generate SOPC Builder system, and run Quartus II compilation** to regenerate the necessary files.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Changing Build Configurations Produces Unexpected Results

The C2H Compiler does not support multiple build configurations (for example Release or Debug) in the Nios II IDE. After creating one or more accelerators in a particular configuration, the C2H Compiler produces undefined results if you switch to a different build configurations and create more accelerators.

Workaround

For a specific SOPC Builder system and Nios II IDE project, specify C2H accelerators in only one build configuration. You can use multiple build configurations, as long as only one configuration specifies C2H Compiler settings.

Solution Status

Not fixed

Hardware Accelerators Remain After Deleting the Software Project

If a system contains C2H accelerators, deleting the software project that defines the accelerators does not remove the accelerators from the hardware system, and the accelerator logic remains in the SOPC Builder system.

Workaround

To remove an accelerator from a system, delete the accelerator from the C2H view in the Nios II IDE first, and then recompile the software project. The C2H Compiler then removes the accelerator from the SOPC Builder system. Once the compilation is complete then the software application can be deleted from the workspace.

Solution Status

Not fixed

Incorrect Results From Logical or Conditional Operation With Side-Effects

The C2H Compiler always evaluates both operands of logical (&&, `||`) and conditional (`?:`) operators. This is different from expected American National Standards Institute (ANSI) C behavior, where operands are evaluated left-to-right, and unnecessary operands are skipped.

For example, in the expression `(i-- && j--)`, if the value of `i` is zero, ANSI C does not evaluate the right-hand-side (RHS) expression, and `j` is not decremented. By contrast, the C2H Compiler evaluates both sides, decrementing `j`.

The following expressions are other examples that might be affected by this issue:

```
if (i-- || j++)
    ...;


a = ((cond == 1)?  i++ : j++);
```

Workaround

Use logical and conditional operations whose operators have no side effects. Operations with side effects include pre- and post- increment and decrement operations (`++`, `--`), memory operations (`*`, `[]`, `.`, `->`), and function calls.

Solution Status

Not fixed

Launch SOPC Builder Button in C2H View

When the Nios II IDE workspace contains multiple projects with multiple system libraries, the incorrect SOPC Builder system might open when you click **Launch SOPC Builder** in the C2H view.

Workaround

Launch SOPC Builder from the Quartus II software. Alternatively, keep only one system library project open at a time while using the C2H Compiler.

Solution Status

Not fixed

Development Board Errata

This section describes in detail the Nios II EDS issues related to Nios development boards.

Intermittent Failures While Accessing CompactFlash Card

The Nios II Development Kit version 5.0 and later includes a CompactFlash controller peripheral suitable for interfacing to CompactFlash cards in True IDE mode on Nios development boards. For True IDE mode to operate, CompactFlash cards require that the `ATASEL_N` input be driven to ground during power-up.

The CompactFlash controller peripheral includes a configurable power register, used to cycle power to CompactFlash cards from Nios II software through a metal oxide semiconductor field-effect transistor (MOSFET) on the Nios development boards. However, in certain development boards, power to the CompactFlash card does not turn off completely during this power cycle operation. Because of this condition, the CompactFlash might not sample the `ATASEL_N` pin during the power-cycle operation after FPGA configuration when this pin is driven to ground. Instead, the CompactFlash card might sample the `ATASEL_N` pin when power is first applied to the development board, when I/O is not yet driven by the FPGA (before FPGA configuration).

Workaround

If you encounter errors with CompactFlash when using the Nios development boards, try one of the following solutions:

- Use a different CompactFlash card. Certain cards are more susceptible to the power-cycling issue than others.
- Modify the Nios development board. This is recommended if you are familiar and comfortable with board-level modifications. Disconnect pin 9 (`ATASEL_N`) on the CompactFlash socket on your Nios development board and tie this pin to ground.

☞ The CompactFlash socket uses a staggered numbering on the pins (starting from pin 1: 1, 26, 2, 27, ...); refer to the CompactFlash Association specification for right-angle surface-mount connectors for exact specifications on this connector. This modification permanently enables True IDE mode operation.

Solution Status

Not fixed

Documentation Errata

This section describes in detail the Nios II EDS documentation issues.

Missing Documentation of Interrupt API Properties

"Tcl Commands" in the *Nios II Software Build Tools Reference* chapter of the *Nios II Software Developer's Handbook* does not include information about two `set_sw_property` command properties related to the application programming interfaces (APIs) for interrupts.

Workaround

The missing information belongs with the `set_sw_property` command, in "Tcl Commands" in the *Nios II Software Build Tools Reference* chapter of the *Nios II Software Developer's Handbook*.

The missing information follows:

- `supported_interrupt_apis`—Specifies the interrupt API that the device driver supports. The Nios II SBT analyzes this property for each driver in the system to determine the appropriate API to be used in the system.

  Specify `legacy_interrupt_api` if the device driver supports the legacy API only or `enhanced_interrupt_api` if the device driver supports the enhanced API only. Specify both using a quoted list if the device driver supports both APIs.

  If you do not specify which API your device driver supports, the Nios II SBT assumes that only the legacy interrupt API is supported.

  This property is only available for device drivers.

  For more information about the legacy and enhanced APIs, refer to "Exception Handling" in the *Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook*.

- `isr_preemption_supported`—Specify `true` if your device driver's interrupt service routine (ISR) can be preempted by a higher priority ISR. If you do not specify whether the driver supports preemption, the Nios II SBT assumes that your device driver does not support preemption. If your driver does not have an ISR, but the associated device has an interrupt port, you can set this property to `true`.

  This property is valid for operating systems and device drivers.

Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

Valid Range of hal.log_flags is –1 to 3

The Nios II EDS documentation incorrectly states the valid range for the `hal.log_flags` BSP setting. Valid values of `hal.log_flags` range from –1 through 3.

Solution Status

This issue will be fixed in a future release of the Nios II EDS documentation.

NicheStack TCP/IP Stack - Nios II Edition Does Not Support TFTP

The Nios II documentation should include the following information:

Versions of the NicheStack TCP/IP Stack other than the Nios II Edition include optional Trivial File Transfer Protocol (TFTP) client and server applications. However, the TFTP client and server are not thread safe. They cannot be used in systems with the MicroC/OS-II RTOS.

TFTP is not available with the NicheStack TCP/IP Stack - Nios II Edition. The TFTP client and server features are disabled.

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Error Message After Renaming Project: "Resource is out of sync with the system"

The *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook* does not explain the correct procedure for renaming a project.

Workaround

The missing information is as follows:

To rename a project in the Nios II SBT for Eclipse, execute the following steps:

1. Right click the BSP project and click **Rename**.
2. Type the new BSP name.
3. Right click the BSP project and click **Refresh**.

☞ If you neglect to refresh the project, you might see the following error message when you attempt to build it:

```
Resource <original_bsp_name> is out of sync with the system
```

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Obsolete Wiki URLs in Help Systems and Handbook

The following documents contain obsolete URLs for the Nios Community Wiki:

- The *Nios II Software Developer's Handbook*
- The *Nios II Embedded Design Handbook*
- The Nios II IDE help system

Workaround

You can visit the Nios Community Wiki at www.nioswiki.com.

Solution Status

Fixed in the documentation for v. 9.0.

Missing Source Control Information

The Quartus II Handbook refers to the *Using the Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook* for information about putting Nios II projects in source control. The referenced chapter only describes the process for Nios II SBT projects.

Workaround

The missing information follows.

Archiving Nios II IDE Software Projects

This section helps you identify the files you must include when archiving a Nios II IDE software project. With this information, you can archive a Nios II application project and its associated Nios II system library project.

You might want to archive your projects for one of the following reasons:

- To place them under source control
- To create backups
- To bundle the projects for transfer to another location

This section covers the following information:

- How to find and identify the files that you must include in an archived Nios II IDE software project.
- Which files must have write permission to allow the software projects to be built.

Table 4 and Table 5 list the files required by Nios II IDE software projects. This is the minimum set of files needed to completely rebuild a software project, including the executable and linking file (**.elf**).

Archive your Nios II IDE software projects together with the SOPC Builder system on which they are based. You cannot rebuild a Nios II IDE software project without its associated SOPC Builder system.

The files listed in Table 4 are located in the Nios II application project directory.

Table 4. Files Required for a Nios II Application Project

File Description	File Name	Write Permission Required?
All source files	for example: **app.c**, **header.h**, **assembly.s**, **lookuptable.dat**	No
Eclipse project file	**.project**	No
C/C++ Development Toolkit project file	**.cdtproject**	Yes
C/C++ Development Toolkit option file	**.cdtbuild**	No
Software configuration file	**application.stf**	No

The files listed in Table 5 are located in the Nios II system library project directory.

Table 5. Files Required for a Nios II System Library Project

File Description	File Name	Write Permission Required?
Eclipse project file	**.project**	Yes
C/C++ Development Toolkit project file	**.cdtproject**	Yes
C/C++ Development Toolkit option file	**.cdtbuild**	No
System software configuration file	**system.stf**	Yes

☞ You must have write permission for certain files. The tools write to these files as part of the build process. If the files are not writable, the toolchain fails. Many source control tools mark local files read-only by default. In this case, you must override this behavior. You do not have to check the files out of source control unless you are modifying the Nios II software project.

Solution Status

In the documentation for v. 9.0, the missing information appears in the *Nios II Integrated Development Environment* chapter of the *Nios II Software Developer's Handbook*. In the documentation for v. 9.1, the missing information appears in the *Using the Nios II Integrated Development Environment* appendix of the *Nios II Software Developer's Handbook*.

Compiler Flags for Building Custom Newlib

"Common BSP Tasks" in the *Using the Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook* describes how to compile a custom version of the newlib library. However, it does not list the compiler flags that must be selected to generate the correct library.

Workaround

The missing information is as follows:

The Nios II EDS provides a number of precompiled newlib implementations. The provided libraries are precompiled with every viable combination of the GCC compiler flags shown in Table 6.

When you create a BSP with a precompiled newlib, the Nios II SBT selects the newlib matching your BSP's compiler settings. When you create a custom newlib, you must ensure that the compiler flags listed in Table 6 match your BSP's settings.

Table 6. GCC Compiler Options for Newlib

Option Name	Purpose
`-pg`	Link for profiling with gprof
`-EB`	Generate big-endian code
`-mstack-check`	Enable stack checking
`-mno-hw-mul`	Disable use of the `mul` family of instructions
`-mhw-mulx`	Enable use of the `mulx` family of instructions
`-mcustom-fpu-cfg`	Use a floating point custom instruction

The Nios II-specific compiler flag `-mcustom-fpu-cfg` requires the Nios II processor core to be implemented with the floating-point custom instruction as specified in the flag argument. The compiler flag can have one of the following two argument values:

- `-mcustom-fpu-cfg=60-1`—Use the Nios II floating-point custom instruction without divider
- `-mcustom-fpu-cfg=60-2`—Use the Nios II floating-point custom instruction with divider

For details about the Nios II-specific compiler flags `-mstack-check`, `-mno-hw-mul`, and `-mhw-mulx`, refer to "Altera Nios II Options" in *Using the GNU Compiler Collection (GCC)*, installed with the Nios II EDS. For details about the `-pg` compiler flag, refer to "Compiling a Program for Profiling" in *GNU Profiler*, installed with the Nios II EDS. To find the installed documentation, on the Windows Start menu, click **Programs** > **Altera** > **Nios II EDS <*version*>** > **Nios II <*version*> Documentation**, and then click **Literature**.

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Error Message on Invalid Exception or Reset Vector

Nios II compiler error messages, added in v 8.1, are not documented in the *Nios II Software Developer's Handbook*.

Workaround

The missing information is as follows:

While building a Nios II software project, the compiler produces an error message if the SOPC Builder system defines an invalid exception or reset vector. Table 7 shows possible error messages and their causes.

In the SOPC Builder system, ensure that the exception and reset vectors conform to all the criteria listed in Table 7.

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Table 7. Exception and Reset Vector Error Messages

Message	Cause
`The section mapping ".entry" is not mapped to a memory region.`	The `.entry` section is mapped to a nonexistent memory region.
`The section mapping ".exceptions" is not mapped to a memory region.`	The `.exceptions` section is mapped to a nonexistent memory region.
`The section mapping ".entry" maps to a memory region "<region name>" which is not the required length "32".`	The `.entry` section is mapped to a memory region that is smaller than 32 bytes.
`The section mapping ".entry" maps to a memory region "<region name>" which does not start at the SOPC Design specified address "<base address>".`	The `.entry` section is mapped to a memory region that does not start at the reset vector base address.
`The section mapping ".exceptions" maps to a memory region "<region name>" which does not start at the SOPC Design specified address "<base address>".`	The `.exceptions` section is mapped to a memory region that does not start at the exception vector base address.
`The SOPC design Nios II reset and exception vector addresses are mapped to the same memory. For HAL the exception address must be at least 32 bytes larger than the reset address.`	The `.entry` section and `.exceptions` section map to the same memory device, and the base of the `.exceptions` section is less than 32 bytes above the base of the `.entry` section.
`The section mapping ".exceptions" mapped device "<memory 1>" is not the same as the SOPC Design specified device "<memory 2>".`	The `.exceptions` section is mapped to a device that is different from the exception vector device specified in the SOPC Builder system.
`The section mapping ".entry" mapped device "<memory 1>" is not the same as the SOPC Design specified device "<memory 2>".`	The `.entry` section is mapped to a device that is different from the reset vector device specified in the SOPC Builder system.

Nios II IDE Online Help Expand Buttons Do Not Work

The expand text arrow and the **Show All** option in the online help do not work in Internet Explorer.

Workaround

In Internet Explorer, carry out the following actions:

1. Click the **Refresh/Show Current Topic** icon (yellow arrows icon at the top of the search pane). The table of contents appears with the selected topic highlighted.
2. Click the highlighted topic to refresh the browser frame. Expanded text works.

Alternatively, use Firefox 2.0.0.6.

Solution Status

Not fixed

Hardware Abstraction Layer Errata

This section describes in detail the Nios II Hardware Abstraction Layer issues.

Missing Structure Member Errors in alt_log_printf.c with Small JTAG UART Driver

In a BSP, if you enable the small JTAG UART driver (`altera_avalon_jtag_uart_driver.enable_small_driver`), and enable Altera logging (`hal.log_port`) using the JTAG UART as the log port, when you build the project, the compiler reports missing structure members in **alt_log_printf.c**. The project fails to build.

Workaround

To avoid this issue, execute the following steps:

1. In function `alt_log_jtag_uart_print_control_reg()` in **alt_log_printf.c**, insert the italicized code as shown in Example 4.

Example 4. ALT_LOG_PRINTF() Correction

```
ALT_LOG_PRINTF(
    "%s SW CirBuf = %d, HW FIFO wspace=%d AC=%d WI=%d RI=%d WE=%d RE=%d\r\n",
    header,

/* Beginning of inserted code */
#ifndef ALTERA_AVALON_JTAG_UART_SMALL
    (dev->tx_out-dev->tx_in),
#else
    0,
#endif
/* End of inserted code */

    space,ac,wi,ri,we,re);
```

2. Rebuild the project.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

"unused variable" Warning When Building alt_main.c

If you set the `hal.enable_exit` BSP setting to `false`, when building the project you see the following warning in **alt_main.c**:

```
warning: unused variable 'result'
```

Design Impact

This warning is harmless.

Workaround

Set `hal.enable_exit` to `true`.

Alternatively, you can ignore this harmless warning.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Build Warnings in Performance Counter Driver Files

If your Nios II system contains a performance counter, you might see the following warning when you build a BSP:

```
warning: return makes pointer from integer without a cast
```

This warning appears in `alt_get_performance_counter_base()` in the software file **altera_avalon_performance_counter.c**.

This warning appears when the your BSP uses the Altera performance counter driver, and the C preprocessor symbol `PERFORMANCE_COUNTER_0_BASE` is defined.

This warning is harmless.

Workaround

In the `alt_get_performance_counter_base()` function in **altera_avalon_performance_counter.c**, add typecast `(void *)` to `PERFORMANCE_COUNTER_0_BASE`.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Hardware Example Design Errata

This section describes in detail the Nios II EDS hardware example design issues.

Incorrect PLL Phase Shift in Stratix II and Cyclone II Designs

In several example designs targeting the Nios II Cyclone II 2C35 and Nios II Stratix® II 2S60 RoHS development boards, the synchronous static random access memory (SSRAM) and legacy DDR SDRAM clocks are implemented with an incorrect phase-locked loop (PLL) phase shift.

This issue affects the following example designs:

- 2s60 RoHS Triple Speed Ethernet SG-DMA
- 2s60 RoHS Standard
- 2s60 RoHS Full Featured
- 2c35 Triple Speed Ethernet SG-DMA
- 2c35 Standard
- 2c35 Full Featured

This issue also affects any custom 2c35 or 2s60 RoHS design using SSRAM or legacy DDR SDRAM.

The incorrect phase shift might cause timing violations when reading from or writing to SSRAM or legacy DDR SDRAM.

Workaround

To correct the PLL settings, perform the following steps:

1. Open the standard design in SOPC Builder.
2. Double-click the component instance named `pll` to launch the PLL MegaWizard™ interface.
3. Click **Launch Altera's ALTPLL MegaWizard** to launch the MegaWizard interface.
4. Select the **Output Clocks** page and then the clock page used to generate the SSRAM clock. Depending on the particular design, this might be **clk c1**, **clk c2**, or another clock.
5. Ensure that **Clock phase shift** is –3.38 ns.
6. Click the clock page used to generate the write clock for the legacy DDR SDRAM, if any. Depending on the particular design, this might be **clk c1**, **clk c2**, or another clock.
7. Ensure that **Clock phase shift** is 270 degrees.
8. Click **Finish** to save changes and exit the PLL MegaWizard interface.
9. Click **Finish** to save settings to the PLL instance in SOPC Builder.
10. Regenerate the system in SOPC Builder and recompile in the Quartus II software. Refer to the **readme.txt** file in the design example folder for information on compiling designs containing the DDR controller.

For further information refer to **ssram_interface_readme.html**, located in the **<*Quartus II installation directory*>/sopc_builder/documents** folder. This document discusses the SSRAM timing analysis methodology in detail. Additionally, *AN 411: Understanding PLL Timing for Stratix II Devices* discusses clock phase shift calculations and assignments for PLLs in Stratix II devices.

Solution Status

Not fixed

Software Example Errata

This section describes in detail the Nios II EDS software example issues.

Hardware Tutorial Software Example Hangs on Some Boards

The **count binary** application used in the tutorial hangs when run on some Altera development boards. The software runs through the binary count once and stops responding.

This issue affects boards that do not have a display to handle the LCD output. The affected boards include the following:

- Embedded Systems Development Kit, Cyclone III Edition (EP3C120)
- Stratix IV GX FPGA Development Kit (EP4SGX230)

Workaround

Modify line 18 of the **count_binary.h** file, as follows:

```
 # define LCD_PRINTF(lcd, args...) /* Do Nothing */
```

Rebuild and run the software again.

Solution Status

Not fixed

Simple Socket Server Example Displays Incorrect IP Address

The simple socket server networking software example displays an incorrect IP address for the incoming connection.

Workaround

There is currently no workaround for this problem. Ignore the IP address of any incoming connection that is displayed on the console.

Solution Status

Not fixed

Networking Examples

If you are running a networking software example, you might be asked for a nine-digit number. You are directed to find this number on a sticker on your Nios development board, identified by the prefix **ASJ**. Not all Nios development boards have this sticker.

Workaround

If your Nios development board does not have a sticker with the letters **ASJ** followed by a nine-digit number, enter a unique nine-digit number when prompted. To avoid network address conflicts, ensure that this number is unique to each Nios board connected to your network.

Solution Status

Not fixed

Flash Programmer Errata

This section describes in detail the Nios II EDS issues related to the flash programmer.

Unable to Configure FPGA from Flash with Parallel Flash Loader

If you program an FPGA image to flash memory on a board that utilizes the Parallel Flash Loader, the FPGA subsequently fails to load from flash memory. This issue arises because the **sof2flash** utility generates a flash file with an incompatible Programmer Object File (**.pof**) bitstream format.

Affected Configurations

This issue impacts the following target hardware:

- The Stratix IV GX FPGA Development Kit
- The Arria® II GX FPGA Development Kit
- Any hardware using the Parallel Flash Loader

Workaround

1. Visit the MySupport website and request Nios II EDS patch 0.73.

   Alternatively, upgrade to the Nios II EDS v. 9.1 SP1 or later.

2. Invoke **sof2flash** with the following options:

   - `--pfl`
   - `--optionsbits=<`*`option bit address`*`>`

   With these options, **sof2flash** generates a flash file with a compatible **.pof** file bitstream format.

Solution Status

Fixed in v. 9.1 SP1

Boot Failure or elf2flash Error Due to Incorrect Flash File Size

The `sof2flash` utility fails to recognize certain devices. As a result, it generates Motorola S-record (**.flash**) files of the wrong size.

With erasable programmable configurable serial (EPCS) memory, the `elf2flash --after` command fails to recognize the **.flash** file size. The resulting error prevents you from programming flash memory.

With Common Flash Interface (CFI) memory, you can program the **.flash** file, but the Nios II system fails to boot.

This error occurs when programming flash memory for the following devices:

- The following Stratix III devices:
  - EP3SL50
  - EP3SL50F484
  - EP3SL50F780
  - EP3SL110
  - EP3SL110F1152
  - EP3SL110F780
  - EP3SE80
  - EP3SE80F1152
  - EP3SE80F780
- All Stratix IV devices

- All Arria II GX devices

Workaround

Visit the MySupport website and request Nios II EDS 9.0 SP2 patch 2.01 and Quartus II 9.0 SP2 patch 2.63.

Alternatively, download and install the Altera Complete Design Suite v. 9.1.

Solution Status

Fixed in v. 9.1

Delay When Creating New Flash Programmer Configuration

You might experience a delay of several seconds when creating a new Flash Programmer configuration in the Nios II IDE.

Workaround

There is no workaround.

Solution Status

Not fixed

elf2flash File Size Limit

The **elf2flash** utility supports **.elf** files up to approximately 24 MBytes in size. The **elf2flash** utility might fail on files larger than 24 MBytes, with the error message `java.lang.OutOfMemoryError`.

Workaround

Lower the number of symbols in your **.elf** file by turning off debug symbols.

Alternatively, specify less initialized data in the application.

Solution Status

Not fixed

Error Message “No such file or directory” When Programming Flash

When programming flash from a project stored in a path containing spaces, you might receive the spurious error message **No such file or directory**. The flash programmer does not correctly handle spaces in the directory path. However, this error is harmless, because flash programming completes successfully.

Workaround

None required.

Solution Status

Not fixed

Hardware Simulation Errata

This section describes in detail the Nios II EDS issues related to hardware simulation.

Error "UNC paths are not supported" Launching ModelSim

If you launch ModelSim® from a working directory that is mapped via a universal naming convention (UNC) path (a path that starts with `//` instead of drive letter), you receive the following error message in SOPC Builder: **UNC paths are not supported. Defaulting to Windows directory**. This error occurs because ModelSim is calling a command shell, which does not support UNC paths.

Workaround

Map the UNC path to a drive letter and use the drive letter to reference the working directory in the launching shell.

Solution Status

Not fixed

Uninitialized .bss Variables in Simulation

If your program reads the value of an uninitialized `.bss` variable during HDL simulation, and the BSP (system library) is compiled with the **ModelSim only, no hardware support** property enabled in Nios II IDE, a warning appears about unfiltered data being `'x'`. This warning appears because when this property is enabled, the code that clears the `.bss` memory region is omitted to speed up HDL simulation so this memory region is uninitialized. The `.bss` region contains global and static local variables that are not initialized by the application so they default to a value of zero. When the Nios II processor reads uninitialized variables, it displays a warning and converts any of the bits of the uninitialized data to zero which correctly mimics the effect of the missing `.bss` clearing code. The Hardware Abstraction Layer (HAL) code that executes before and after `main()` might use `.bss` variables, so these warnings might appear even if your application does not use the `.bss` section.

Solution Status

Not fixed

Host Platform Errata

This section describes in detail the Nios II EDS issues related to the host development platform.

Cannot Launch Nios II SBT for Eclipse on Red Hat Linux

You might be unable to launch Nios II SBT for Eclipse on the Red Hat Linux operating system. This is an issue with the version of XULRunner in Eclipse 3.4. The issue is resolved in XULRunner 1.9.1, which is available with Eclipse 3.5.

Workaround

Upgrade XULRunner to a newer version. To determine what version of XULRunner you have, type the following command at the command prompt:

```
xulrunner -v↵
```

To upgrade XULRunner using the **yum** software package manager, type the following command:

```
sudo yum update xulrunner↵
```

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Windows/Cygwin: Nios II Processor Generation Failure

You might see the following error message when you generate your Nios II system in SOPC Builder:

```
Error: Generator program for module 'cpu_fpoint' did NOT run
successfully
```

This error might occur on a Windows-based system when all of the following conditions are true:

- You have installed a version of Cygwin other than the one distributed with the Quartus II software.
- You launch the Quartus II software and SOPC Builder from a Nios II Command Shell.
- You enable the floating point custom instruction (FPCI) in the Nios II processor core.

Workaround

Launch the Quartus II software and SOPC Builder from the Windows Start menu, and regenerate your system.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Windows Vista: Limited Support in Nios II EDS

The Quartus II software introduces Windows Vista (32-bit and 64-bit) support in version 7.2. However, the Nios II Embedded Design Suite supports only the following platforms:

- Windows 2000
- Windows XP
- Windows XP (64)
- Windows Vista Business Edition (32-bit)
- SUSE 9 (32-bit)
- SUSE 9 (64-bit)

- Red Hat Linux v3.0 (32-bit)
- Red Hat Linux v3.0 (64-bit)
- Red Hat Linux v4.0 (32-bit)
- Red Hat Linux v4.0 (64-bit)

Workaround

There is no workaround.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Nios II IDE Errata: Building Projects

This section describes in detail the issues related to building projects in the Nios II IDE.

Nios II IDE Command-Line Tools Select Wrong Workspace on Linux

On Linux systems, the Nios II IDE command-line tools use the default Eclipse project workspace, regardless what workspace you are using in the IDE. The default workspace is at `$SOPC_KIT_NIOS2/eclipse/nios2-ide-workspace-<version>`.

This issue affects the following tools:

- `nios2-create-system-library`
- `nios2-create-application-project`
- `nios2-build-project`
- `nios2-import-project`
- `nios2-delete-project`

Workaround

Explicitly supply the Nios II IDE command-line tools with a workspace location, by means of the `-data` command-line argument. The syntax of the `-data` argument is as follows:

```
-data <path to workspace>
```

The path to the workspace must be absolute, and must not contain whitespace. Aside from these restrictions, any valid file system path can be used for the workspace. Example 5 shows how to specify a workspace path.

Example 5. Providing a Non-Default Workspace Location in Linux

```
nios2-create-project -data $HOME/myworkspace <other arguments>↵
```

Solution Status

Not fixed

Nios II IDE Command-Line Tools Hang on Windows

On Windows systems, the Nios II IDE command-line tools are sometimes unable to locate the project workspace. When this happens, the tools might hang.

This issue potentially affects the following tools:

- `nios2-create-system-library`
- `nios2-create-application-project`
- `nios2-build-project`
- `nios2-import-project`
- `nios2-delete-project`

Workaround

Explicitly supply the Nios II IDE command-line tools with a workspace location, by means of the `-data` command-line argument. The syntax of the `-data` argument is as follows:

```
-data <path to workspace>
```

The path to the workspace must be absolute, and must not contain whitespace. Aside from these restrictions, any valid file system path can be used for the workspace. The workspace shown in Example 6 is the default workspace used by the Nios II IDE in version 9.0.

Example 6. Providing a Workspace Location

```
nios2-create-project -data c:/altera/90/nios2eds/\
    bin/eclipse/nios2-ide-workspace-9.0 <other arguments>
```

Solution Status

Not fixed

Nios II IDE Cannot Find stdio.h in Outline View

If you create a new project using the Nios II IDE and try to open the **stdio.h** file from the Outline view before building the project, the IDE displays the error message **No include files were found that matched that name**.

Workaround

Build the project before attempting to open **stdio.h**.

Solution Status

Not fixed

Nios II IDE Hangs With UNC Project Path

In the **New Project** dialog box, if you turn on **Specify Location** and specify the path in UNC form, the IDE might hang.

Workaround

Map the UNC path to a remote drive which looks like a Windows drive to the Nios II IDE.

Solution Status

Not fixed

Build Failure with Nios II Advanced Exceptions, MMU, and MPU

Projects created in the Nios II IDE for an SOPC Builder system containing a Nios II processor configured with advanced exceptions, the memory management unit (MMU), or the memory protection unit (MPU) fail to build, and generate the following error messages:

```
ERROR - Classic build flow for Nios II system library does not support
the Nios II \
    advanced exceptions.
Use the non-classic Nios II Board Support Package instead.
<timestamp> - (SEVERE) generate: java.lang.IllegalStateException: \
    java.lang.IllegalStateException: \
    com.altera.ingenuous.GTFElement.GTFElementErrorException: \
    <error> element in GTF script
make[1]: *** [system_description/../obj/system.h-t] Error 1
make: *** [system_project] Error 2
```

Workaround

Create the software project with the Nios II SBT command-line development flow. Alternatively, use the Nios II SBT for Eclipse.

Solution Status

Not fixed

Build Command Not Functional for BSPs Created With the Nios II SBT

The build option in the Nios II IDE menu does not rebuild BSPs imported to the IDE.

Workaround

The Nios II SBT for Eclipse can both build and debug projects created on the command line. The Nios II SBT for Eclipse is the preferred tool for debugging Nios II SBT projects.

For information about the Nios II SBT for Eclipse, refer to the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*.

In the Nios II IDE, you can build the BSP by building the associated application project.

Solution Status

Not fixed

Linker Errors with Dual-Port Memories

If your instruction master and data master ports are connected to the same dual-port memory and the ports have different addresses, your code fails to run or you experience a linker error. The Nios II IDE does not warn you of the addressing violation.

Workaround

Assign the same address to both ports of the dual-port memory.

Solution Status

Not fixed

User-Managed BSP Settings Not Supported in Nios II IDE

For projects created with the Nios II SBT and imported to the Nios II IDE, the IDE configuration settings have no effect.

For example, objdump, compiler, and linker settings made in the IDE are ignored. This behavior occurs because Nios II SBT projects are not IDE-managed projects. In addition, the make-related preferences do not pertain to imported Nios II SBT projects. The IDE ignores these options during the build process.

Workaround

Make these settings in the project's makefile.

Solution Status

Not fixed

URL Project Location Causes Project Creation Error

When you try to create a new project in an existing workspace, you might see an error dialog box saying:

```
Project cannot be created. Reason: Internal Error
```

This error might occur if the path to any project in the workspace is a URL location, for example **file:/F:/Design**. To view the path, right-click on the project and select **Properties**.

Workaround

Import your existing application and system library projects to a new workspace.

Solution Status

Not fixed

Compilation Error with Separate Exception Stack Option

Choosing the **Use a separate exception stack option** might cause the compilation error `"UNDEFINED VARIABLE %STACK_POINTER%"` when building a project. This error occurs if the exception stack is larger than the memory available for it.

Workaround

On the system library properties page for the project, turn off the separate exception stack or reduce the **Maximum exception stack size** setting.

Solution Status

Not fixed

Incorrect Stack and Heap Free Space Report

The makefile reports an incorrect number of bytes free for the stack and heap, if the heap and stack are in different memory regions.

Workaround

No workaround available.

Solution Status

Not fixed

Nios II IDE Reports Problems Without Displaying Error in Console

When building a project, the Nios II IDE reports problems, but the build output in the console does not contain any errors.

The Nios II IDE incorrectly reports some linker warnings as errors, with a dialog box saying **Errors exist in a required project**. The Dhrystone software example exhibits this behavior, and recompiling the project again makes the issue go away.

Workaround

If the Console output does not contain errors, then the project built correctly. On subsequent builds, the linker step is skipped and the errors do not appear.

Solution Status

Not fixed

Nios II IDE Errata: Debugging Projects

This section describes in detail the issues related to debugging projects in the Nios II IDE.

Method for Accessing MMU and MPU Registers in the Debugger

You might observe an error if you try to read or write an MMU or MPU register via the Nios II Debugger.

Workaround

To read an MPU region, execute the following steps:

1. Set region `INDEX` in the `MPUBASE` register.
2. Exit and reenter debug mode, that is, single step.

3. Set the RD bit in the MPUACC register.
4. Exit and reenter debug mode, that is, single step.
5. Read back MPUBASE for pertinent information.
6. Read back MPUACC for pertinent information.

For a system with an MMU or MPU, this workaround allows you to read and write the current values of the registers. However, you cannot use it to control MPU regions or MMU TLB entries.

The debug core copies the Nios II processor's register values to its internal memory when the processor enters debug mode. The debug core writes register values back to the processor only when the processor leaves debug mode. Therefore, if you attempt to set an MPU region with several consecutive values, only the last one, when leaving debug mode, is committed to the processor.

Solution Status

Not fixed

IDE Cannot Display Imported Profiling Data

If you create a software project in the Nios II SBT command line flow, generate profiling data to **gmon.out** in the Nios II command shell, and then import the project to the IDE, you cannot use the Profiling perspective to view the profiling data. When you attempt to view **gmon.out**, the IDE displays the error message **nios2-elf-gprof: ../: not in a.out format**.

Workaround

Run `nios2-elf-gprof` from the Nios II command shell to generate a profiler report.

Solution Status

Not fixed

Error Message when Downloading .elf File

If the **Build Automatically** option is turned on in the IDE, you might get the following error when downloading an executable and linking formal file (**.elf**):

```
"2 [main] sh 5736 fork: child 3892 - died waiting; \
    for longjmp before initialization, errno 11"
```

The **Build Automatically** option is off by default, as recommended by the Eclipse help system.

Workaround

Turn off the **Build Automatically** option and download the **.elf** file again.

Solution Status

Not fixed

Variable Casting Unsupported in ISS

Casting variables in the Variables view in the Debug perspective when using the Nios II ISS might cause an exception dialog box to open.

This exception might occur if you select **Debug as ISS** in the Nios II IDE and try to cast variables via the Variables view.

Workaround

There is no workaround.

Solution Status

Not fixed

Trace Debug Does Not Support Instruction-Related Exceptions

The instruction-related exception handler is not supported by the Nios II trace tools.

Workaround

There is no workaround.

Solution Status

Not fixed

Trace Debug Does Not Support the JMPI Instruction

The JMPI instruction is not supported by the Nios II trace tools.

Workaround

There is no workaround.

Solution Status

Not fixed

Cannot Locate Source Code in Driver Files Shared by Multiple Projects

If you hit a breakpoint in a driver file, and that driver file is shared with another project that is closed, the Nios II IDE might indicate that it cannot locate the source code.

Workaround

Open the closed system library project and resume debugging.

Solution Status

Not fixed

Console Window Is Not Updated After ISS Error

After performing a **Run as ISS**, if you receive an ISS error in the console window, the console is not updated subsequently.

Workaround

Close the console window after receiving an ISS error. A new console window opens when a new message is available.

Solution Status

Not fixed

ISS Fails on Designs Containing Triple Speed Ethernet MAC or SG-DMA Components

You receive an Internal Error when attempting to perform an ISS simulation of designs containing the Altera Triple Speed Ethernet media access control (MAC) or SG-DMA components because the Nios II ISS does not support these components.

Workaround

Remove the Triple Speed Ethernet MAC and SG-DMA components from your system and perform ISS simulation on the simplified system. You can also simulate the design in ModelSim or test it on hardware.

☞ Before removing the Triple Speed Ethernet MAC and SG-DMA components, make a copy of the unmodified system to ensure that you can return to the original configuration.

Solution Status

Not fixed

Memory Window Sets Control Register Values Incorrectly

The memory window might incorrectly set values in memory-mapped control registers. For example, writing 0x1234 to a byte addressed register results in the value 0x3434 being stored in the register. The memory window shows this incorrect value.

Workaround

Use the GNU debugger (GDB) console window in the IDE, instead of the memory window, to write to the registers. For example, type the following command:

```
set {int} <register address>=0x1234↵
```

☞ You must refresh the memory window for it to correctly display the target value.

Solution Status

Not fixed

Programs That Interact With a Terminal Console on Windows Do Not Work

Programs with this behavior work in v. 6.0 and earlier, but do not work in Nios II IDE v. 6.1 and later.

The Eclipse platform in v. 6.1 and later of the IDE (on Windows only) sends the string `\r\n` instead of just `\n` when running or debugging using the Terminal. This behavior can break existing software designs, and it is inconsistent with `nios2-terminal`, which still just sends `\n`.

Workaround

Change the software running on the Nios II processor to parse for `\r\n` as well as `\n`.

Solution Status

Not fixed

"Run as ModelSim" in the Nios II IDE Fails

The **Run as ModelSim** command might fail on launch configurations created in v. 7.0 or earlier of the IDE. This problem does not occur for new launch configurations.

Workaround

Select a location for the ModelSim tool from the launch configuration dialog box. You can use the **Browse** button next to the **ModelSim path** group, or type in a path to the ModelSim directory (for example **c:/altera/71/modelsim_ae/win32aloem**).

Solution Status

Not fixed

The Restart Command on the Run Menu Does Not Work

The **Restart** command on the **Run** menu does not work.

Workaround

Stop the program, then debug it again. If the debugger is hung in an endless loop, use the following **bash** alias to break the target, then stop it:

```
alias break="kill -2 \'ps -a | grep nios2-elf-gdb | cut -f6 -d' '\'"
```

Solution Status

Not fixed

Watchpoints Do Not Work on Certain Variables

Watchpoints do not work on a variable whose size is not 32 bits.

Workaround

Change the types of global and static local variables to `int`, `long`, or `unsigned long` before setting watchpoints on them.

Solution Status

Not fixed

Nios II IDE Errata: Navigating Projects

This section describes in detail the issues related to navigating projects in the Nios II IDE.

Nios II IDE Freezes While Displaying the Splash Screen

After clicking **Switch Workspace** on the File menu on a Windows machine, a Nios II IDE splash screen appears. Unfortunately, this splash screen obscures the dialog box that asks you to specify the new workspace. As a result, the IDE appears to freeze.

Workaround

Press Alt-Tab to switch applications. Two relevant application icons appear: an Eclipse icon associated with the splash screen and a Nios II IDE icon associated with the **Workspace** dialog box. Select the Nios II icon to bring the dialog box to the foreground.

Solution Status

Not fixed

Internal Error When Double-Clicking on a Large Objdump File

On Windows when opening a large **objdump** file in the Nios II IDE, you might get the following error message:

```
Unable to create this part due to an internal error.
Reason for the failure: Editor could not be initialized.
```

Workaround

Adjust the Windows launch arguments for the Nios II IDE editor. Perform the following steps:

1. On the Windows Start menu, browse to the **Nios II EDS** program icon, right-click it, then click **Properties**. The **Windows Properties** dialog box appears.
2. In the **Target** field, append `"vmargs -Xmx1024m"` to the end of the path to the Nios II IDE executable. For example:

   ```
   C:\altera\72\nios2eds\bin\eclipse\nios2-ide.exe -vmargs -Xmx1024m
   ```

Solution Status

Not fixed

C/C++ Scanner Does Not Support Certain C/C++ Constructs

The C/C++ scanner performs C/C++ Search, navigation, open declaration, and parts of content assist. Due to limitations of the C/C++ scanner, these features do not work with C code constructs not supported by the C++ language. An example is functions that take a function pointer as an argument.

Workaround

If the C/C++ Search fails, use the File Search facility.

Solution Status

Not fixed

Nios II SBT Errata

This section describes in detail the issues in the Nios II SBT.

BSP Not Updated for Memory Size Changes in SOPC Builder

If you change the size of a memory in SOPC Builder, the memory region size in any previously created BSP is no longer correct. Regenerating the BSP does not update the BSP's memory region size.

Workaround

In the BSP Editor, on the **Linker Script** tab, you can correct the memory region size either of the following ways:

- Edit the memory region size manually.
- Click **Restore Defaults** to rerun the default Tcl script. The default Tcl script reads the updated memory region size from the SOPC information file (**.sopcinfo**) and updates the BSP.

For detailed information about keeping your BSP consistent with changes in the underlying SOPC Builder system, refer to "Revising Your BSP" in the *Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook*.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Build Errors on Interrupt API Calls

You might observe build errors if your application project installs an ISR that calls the `alt_irq_register()` function.

The Nios II EDS v9.1 includes support for an optional External Interrupt Controller (EIC) interface on the Nios II processor. To support the EIC interface, the HAL includes an enhanced interrupt API.

When you create a BSP, the SBT determines which interrupt API(s) the device drivers in your system support. If all registered drivers in the BSP identify themselves as supporting the enhanced API, the SBT implements the enhanced API. All Altera device drivers in v9.1 and later support the enhanced API.

When the enhanced API is implemented, the legacy interrupt API, including `alt_irq_register()`, is not available. Therefore, if the application code contains a call to a legacy API function, a linker error occurs.

This issue might affect application projects that call `alt_irq_register()` or other legacy interrupt API functions directly, rather than depending on drivers in the BSP for interrupt support.

If your Nios II application project's source code registers an ISR using `alt_irq_register()`, or includes an ISR, your application might fail to compile in the Nios II EDS v. 9.1, due to undefined references to `alt_irq_register()`, or to a change to the ISR function prototype.

Workaround

Modify your application code to use the new enhanced interrupt API. To use the enhanced API, you must modify several function calls and all ISR function prototypes.

For information about supporting the enhanced interrupt API, refer to the *Exception Handling* chapter of the *Nios II Software Developer's Handbook*, or to *AN 595: Vectored Interrupt Controller Usage and Applications*.

You must upgrade to the enhanced API if you wish to use an EIC in your hardware design to accelerate interrupt response time. The enhanced API also works with the Nios II processor's internal interrupt controller.

Solution Status

Not fixed

BSP Editor Incorrectly Allows Mixed Drivers

If multiple drivers are found for a single peripheral type, and you have multiple instances of the peripheral type in your SOPC Builder system, the Nios II BSP editor allows you to manually select a different driver for each instance of the peripheral or processor. For example, if you have two `altera_avalon_timer` devices, you can select two different timer drivers.

This behavior is not fully supported. You must choose the same driver for both instances of the peripheral type.

You can make driver assignments in one of the following ways:

- In the Nios II BSP editor, within the **Drivers** tab. The **Driver name** associated with each peripheral can be edited, if multiple driver types are found that support the peripheral. You must make the same driver type selection for each instance of a peripheral type in the list of peripherals as defined in the **Module Name** column.
- With the Nios II SBT in command-line mode (`nios2-bsp`, `nios2-bsp-create-settings`, `nios2-bsp-update-settings`): the **set_driver** command can be invoked (either from a Tcl script or with the `--cmd` option). The **set_driver** command accepts a peripheral module instance name and driver name as arguments. If you are using the command line development flow and wish to use this command, you must invoke it once for each instance of a peripheral type in your system, similar to the BSP editor flow described in the previous paragraph.

Workaround

There is no workaround.

Solution Status

Fixed in v. 9.1

nios2-bsp --help Displays Incorrect List of BSP Types

The Nios II SBT supports custom HAL-based operating systems, when they are added as Tcl-scripted software packages. However, the `nios2-bsp` tool help message indicates that there are only two BSP types, HAL and UCOSII. If other HAL-based operating system components are added they do not appear in the `nios2-bsp` help message.

Workaround

See "nios2-bsp-create-settings Fails to List BSP Types" to get a correct, dynamic list of available BSP types.

Solution Status

Fixed in v. 9.1

nios2-bsp-create-settings Fails to List BSP Types

The help message for `nios2-bsp-create-settings` incorrectly states that using the `--type` option with no value creates a list of available BSP types.

Workaround

To get a list of available BSP types, execute the following command:

```
nios2-bsp-create-settings --type x --settings x --sopc=<existing sopcinfo path>↵
```

The following command returns an error message that lists available BSP types:

```
SEVERE: Available BSP type values for the --type argument are: [<list of BSP types>]
```

Solution Status

Fixed in v. 8.1

Makefile Error with Absolute Source Path

You might receive the following error message from the Nios II SBT:

```
Makefile:: *** target pattern contains no '%'
```

This error message can appear if you are using Windows, you specify an absolute source path with the `--src-dir` option, and your application directory, specified with the `--app-dir` option, is the same as your source path.

Workaround

Use a relative path to the source directory, or organize your files so that your application directory is different from your source directory.

Solution Status

Fixed in v. 8.1

Incorrect Linker Errors

While building a HAL-based Nios II software project, the linker displays misleading error messages if the SOPC Builder system provides less than 32 bytes of memory at the reset vector. The linker errors are similar to the following:

```
/cygdrive/d/altera/80/nios2eds/bin/nios2-gnutools/H-i686-pc-cygwin/bin/../lib/gcc/ \
    nios2-elf/3.4.6/../../../../nios2-elf/bin/ld:/cygdrive/d/test3/software/ \
    hello_world_0_syslib/Debug/system_description/generated.x:186: warning: memory \
    region UNDEFINED not declared
/cygdrive/d/altera/80/nios2eds/bin/nios2-gnutools/H-i686-pc-cygwin/bin/../lib/gcc/ \
    nios2-elf/3.4.6/../../../../nios2-elf/bin/ld:/cygdrive/d/test3/software/ \
    hello_world_0_syslib/Debug/system_description/generated.x:186: parse error
collect2: ld returned 1 exit status
make: *** [hello_world_0.elf] Error 1
```

Workaround

In the SOPC Builder system, if the reset and exception vectors are in the same memory device, ensure that the exception vector is at least 32 bytes above the reset vector.

Solution Status

Fixed in v. 8.1

Java Heap Space Error When Creating BSP

When creating a BSP, the Nios II SBT might hang with the following message:

`SEVERE: OutOfMemoryError: Java Heap Space`.

This problem occurs when your SOPC Builder design has connection cycles. A connection cycle consists of a group of components that master one another in a circular arrangement. For example, if `ddr_sdram_bridge` masters `lcd_sgdma`, and `lcd_sgdma` masters `ddr_sdram_bridge`, the two components form a simple connection cycle. In this situation, the SBT is unable to create a BSP.

Workaround

Avoid connection cycles in the SOPC Builder system.

Solution Status

Fixed in v. 9.0

SBT Fails if Nios II EDS is Installed in a Path Containing Spaces

The Nios II Command Line SBT fails if you install the Nios II EDS in a path containing spaces.

Workaround

Reinstall the Nios II EDS to a path that does not contain spaces.

Solution Status

Not fixed

User-Managed BSP Settings Not Supported in Nios II IDE

This issue is described on page 37.

Upper-Case File Extensions Not Supported

The Nios II SBT for applications and libraries (`nios2-app-generate-makefile` and `nios2-lib-generate-makefile` commands) does not support source files with certain upper-case extensions. If a file with an upper-case extension is included, the make command stops with no descriptive warning.

Only Nios II assembly language files built by the C preprocessor can have upper-case file extension (**.S**). All C language files must have the extension **.c** or **.h**. C++ language source files must have the extension **.cpp**, **.cxx**, **.cc,** or **.h**.

Workaround

Rename all C language files with the extension **.c** or **.h**. Rename all C++ language files with the extension **.cpp**, **.cxx**, **.cc,** or **.h**.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Peripheral Errata

This section describes in detail the Nios II EDS issues related to peripheral components.

SBT Fails With UART Driver

If you use the SBT on a design containing the SOPC Builder UART peripheral, and if driver `ioctl()` support for the `altera_avalon_uart` component is enabled, then the BSP build process might fail. The **system.h** generated by the SBT contains an invalid type for the UART parameter.

Workaround

To correct **system.h**, carry out the following steps:

1. Manually edit the **system.h** file by changing the following line:

   ```
   #define UART1_PARITY "N"
   ```

   to:

   ```
   #define UART1_PARITY 'N'
   ```

   ☞ Change double quotes to single quotes.

2. Build the BSP.

Solution Status

Fixed in v. 8.1

Unaligned Transfers of Small Payloads Fail on SG-DMA

The Scatter Gather DMA SOPC Builder peripheral does not correctly handle unaligned transfers with small payloads. A payload length smaller than the data width causes erroneous data transfers.

Workaround

Avoid using DMA devices to transfer small payloads.

If absolutely necessary, for a 32-bit SG-DMA, a minimum length of 4 bytes guarantees that data is transferred correctly.

Solution Status

Not fixed

DMA Controller Always Busy in Burst Mode

The DMA controller component (`altera_avalon_dma`), when enabled for burst transactions, does not perform transfers at widths other than its full data width. The DMA controller is always busy.

Workaround

When bursting is enabled, the DMA controller must be programmed to perform transactions at its full data width.

Solution Status

Not fixed

Non-System-Wide Reset Can Cause Improper Initialization of Mailbox Core

The `altera_avalon_mailbox` peripheral might not be initialized properly when a soft (non-system-wide) reset occurs. In this condition, mailbox contents (read and write pointers) are not reinitialized and might show potentially stale data.

Workaround

Ensure that a system-wide reset event occurs by asserting the `reset_n` input to the SOPC Builder system containing the mailbox. This resets all peripherals and Nios II processors in the system.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Target Software Errata

This section describes in detail the Nios II EDS issues related to target software packages.

NicheStack tcp_wake() Function Does Not Awaken All select() Sleepers

This issue impacts software based on the NicheStack TCP/IP Stack - Nios II Edition, with multiple sleeping tasks that depend on the same `select()` event to wake up. The `tcp_sleep()` and `tcp_wake()` functions do not work properly for tasks that call `select()`.

Tasks can normally use `tcp_sleep()` and `tcp_wake()` to suspend and then awaken themselves upon specific events. The `select()` function passes an event pointer to `tcp_sleep()` that is identical for every task that might call `select()`. All other events that call `tcp_sleep()` pass it a unique event pointer that specifically identifies them as the task to be awakened at the next event.

The `tcp_wake()` function apparently assumes that the list of sleepers contains unique event pointers by which they can be identified. Therefore, when a `select()` event occurs, instead of awakening all tasks waiting for it, `tcp_wake()` awakens only the first task in the list. Therefore, when more than one task calls `select()` to await some outcome on a socket, only the highest-priority task in the sleepers list is awakened at `select()` events. Other sleeping tasks that depend on the same `select()` event do not wake up.

Workaround

There is no workaround.

Solution Status

This issue will be fixed in future release of the Nios II EDS.

EIC-Based Nios II System Crashes With NicheStack

When you attempt to run software based on the NicheStack TCP/IP Stack - Nios II Edition on a hardware design incorporating an external interrupt controller such as the VIC, the Nios II system crashes during initialization, with unpredictable results.

Workaround

There is no workaround.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Gigabit Ethernet Performance Issues

The NicheStack TCP/IP Stack - Nios II Edition might hang while sending transmission control protocol (TCP) packets to the host, or experience dropped packets when receiving TCP packets from the host. These issues can arise with version 3.1 of the NicheStack TCP/IP Stack, using the triple-speed Ethernet component to transmit over a Gigabit Ethernet point-to-point connection.

Workaround

Download and install the patches at www.altera.com/support/kdb/solutions/rd01132009_588.html.

Alternatively, upgrade to the Nios II EDS v. 9.0 or later.

Solution Status

Fixed in v. 9.0

stdio Does Not Work with MicroC/OS-II and Small C Library

`stdin`, `stdout`, and `stderr` do not work in MicroC/OS-II applications built with the Small C library option.

Workaround

Disable the small C library option.

Solution Status

Not fixed

cout From MicroC/OS-II Task Does Not Send Data to stdout

If neither `printf()` or `cout` is used from `main()` before tasks are started, `cout` does not work from a task.

Workaround

Add the following C++ code to the beginning of `main()`:

```
std::ios_base::sync_with_stdio(false);
```

Solution Status

Not fixed

malloc(), realloc() Failures With MicroC/OS-II

When you use the MicroC/OS-II RTOS, calls to `malloc()` and `realloc()` might fail if successive calls to `malloc()` or `realloc()` within a MicroC/OS-II task occur after changing the task priority of the task in which a memory block is originally allocated.

Workaround

Use one of the following workarounds:

- Allocate and/or reallocate memory blocks outside of MicroC/OS-II tasks, before task switching starts. Preallocated memory blocks make it possible to change thread priorities at runtime.
- Allocate fixed areas of memory using arrays (rather than using `malloc()`) before task switching starts. Fixed memory arrays make it possible to change thread priorities at runtime.
- Allocate memory using `malloc()` or `realloc()` from a MicroC/OS-II task. You can change task priorities at runtime, but only for tasks that have not used `malloc()` or `realloc()`.

Solution Status

Not fixed

Toolchain Errata

This section describes in detail the Nios II EDS issues related to the Nios II and GNU compiler toolchains, including tools such as **gcc**, **gdb**, and **sof2flash**.

Unable to Configure FPGA from Flash with Parallel Flash Loader

This issue is described on page 29.

sof2flash Names Output Files Incorrectly

If **sof2flash** is called with the `--save` option and a relative path is specified for the output file (that is, including "./" or "../"), the raw binary file that is saved is incorrectly named **.rbf** instead of **<flash filename>.rbf**.

Workaround

Provide a full path to the output file(s), or change to the output directory before running the **sof2flash** command so the output file specification is `--output=<`*`filename`*`>.flash`.

Solution Status

Fixed in v. 8.1

GNU Assembler Does Not Accept the --defsym Flag

According the GNU documentation, you can set an assembler definition by using the `--defsym` flag, but it does not work in the following form: `--defsym MY_VAR=1`.

Workaround

There is no workaround.

Solution Status

Not fixed

How to Contact Altera

For the most up-to-date information about Altera products, refer to Table 8.

Table 8. Contact Information

Contact *(1)*	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Altera literature services	Email	literature@altera.com
Non-technical support (General)	Email	nacomp@altera.com
(Software Licensing)	Email	authorization@altera.com

Note:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

Table 9 shows the typographic conventions that this document uses.

Table 9. Typographic Conventions (Part 1 of 2)

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Indicates command names, dialog box titles, dialog box options, and other graphical user interface (GUI) labels. For example, **Save As** dialog box. For GUI elements, capitalization matches the GUI.
bold type	Indicates directory names, project names, disk drive names, file names, file name extensions, dialog box options, software utility names, and other GUI labels. For example, **\qdesigns** directory, **d:** drive, and **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Indicates document titles. For example, *AN 519: Stratix IV Design Guidelines.*
Italic type	Indicates variables. For example, *n* + 1. Variable names are enclosed in angle brackets (< >). For example, <*file name*> and <*project name*>**.pof** file.
Initial Capital Letters	Indicates keyboard keys and menu names. For example, Delete key and the Options menu.
“Subheading Title”	Quotation marks indicate references to sections within a document and titles of Quartus II Help topics. For example, “Typographic Conventions.”
`Courier type`	Indicates signal, port, register, bit, block, and primitive names. For example, `data1`, `tdi`, and `input`. Active-low signals are denoted by suffix `n`. For example, `resetn`. Indicates command line commands and anything that must be typed exactly as it appears. For example, `c:\qdesigns\tutorial\chiptrip.gdf`. Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword `SUBDESIGN`), and logic function names (for example, `TRI`).
1., 2., 3., and a., b., c., and so on.	Numbered steps indicate a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ■	Bullets indicate a list of items when the sequence of the items is not important.

Table 9. Typographic Conventions (Part 2 of 2)

Visual Cue	Meaning
☞	The hand points to information that requires special attention.
⚠ CAUTION	A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.
⚡ WARNING	A warning calls attention to a condition or possible situation that can cause you injury.
↵	The angled arrow instructs you to press Enter.
👣	The feet direct you to more information about a particular topic.

101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support
www.altera.com/support

Nios II Embedded Design Suite Release Notes and Errata

© November 2009 RN-EDS-5.0

About These Release Notes

These release notes cover versions 8.1 through 9.1 of the Altera® Nios® II Embedded Design Suite (EDS). These release notes describe the revision history and errata for the Nios II EDS.

Errata are functional defects or errors, which might cause the product to deviate from published specifications. Documentation issues include errors, unclear descriptions, or omissions from current published specifications or product documents.

For the most up-to-date errata for this release, refer to the latest version of the *Nios II Embedded Design Suite Release Notes and Errata* on the Altera website. You can refer to release notes and errata for the Nios II processor core in the *Nios II Processor* chapter of the *MegaCore IP Library Release Notes and Errata*.

Revision History

Table 1 shows the revision history for the Nios II EDS, including the Nios II C2H Compiler.

Table 1. Nios II Embedded Design Suite Revision History

Version	Date	Description
9.1.0	November 2009	■ The Nios II Software Build Tools for Eclipse, an integrated development environment based on Eclipse. Recommended for all new Nios II projects. ■ The Nios II Integrated Development Environment (IDE) is available in the Nios II EDS, but not recommended for new projects. ■ The Nios II Studio is discontinued. ■ The vectored interrupt controller (VIC) core offers high-performance, low-latency interrupt handling. ■ Nios II Design Examples with memory management unit (MMU): ■ Nios II 3C120 Hardware Reference Design with MMU, Cyclone® III Edition ■ Nios II 4S230 Hardware Reference Design with MMU, Stratix® IV GX Edition Commercial and open-source support for embedded Linux is now available for these systems. ■ Tcl scripting for software packages and drivers enhanced to support callback functions.
9.0.0	March 2009	Added the `nios2-convert-ide2sbt` tool, which converts projects from the Nios II IDE development flow to the Nios II Software Build Tools development flow
8.1.1	January 2009	Nios Community Wiki URL changing (www.nioswiki.com)
8.1.0	November 2008	Maintenance release

For more information on new features and changes, refer to the *Nios II Processor Reference Handbook*, the *Nios II Software Developer's Handbook*, and the *Nios II C2H Compiler User Guide*. For information about changes to the Nios II processor core, refer to the *Nios II Processor* chapter of the *MegaCore IP Library Release Notes and Errata*.

Revision Details

This section provides details about some of the Nios II EDS revisions shown in Table 1.

Nios II Software Build Tools for Eclipse

The Nios II Software Build Tools for Eclipse are a set of plug-ins based on the popular Eclipse framework and the Eclipse C/C++ development toolkit (CDT) plug-ins. The Nios II Software Build Tools for Eclipse provide a consistent development platform that works for all Nios II processor systems. You can accomplish most Nios II software development tasks within Eclipse, including creating, editing, building, running, debugging, and profiling programs. The Nios II Software Build Tools for Eclipse are built from the ground up as plug-ins to standard Eclipse. The Nios II Software Build Tools for Eclipse are based on the Nios II Software Build Tools, a set of powerful commands, utilities and scripts for creating applications, board support packages and software libraries, and managing build options.

Table 2 compares the Nios II Integrated Development Environment (IDE) with the Nios II Software Build Tools for Eclipse.

Table 2. Differences Between Nios II Software Build Tools for Eclipse and Nios II IDE (Part 1 of 2)

Feature	Nios II IDE	Nios II Software Build Tools for Eclipse
Types of projects	■ Application ■ Board support package (BSP)—Called a system library in the IDE ■ User Library (optional)	■ Application ■ BSP ■ User Library (optional)
Handling updates to the SOPC Builder system	Updates the BSP automatically	You must initiate the BSP update (in the Nios II menu, click Generate BSP)
Options for creating projects	Application from template. BSP is created automatically as part of application.	■ New Application and BSP from template ■ Standalone application ■ Standalone BSP ■ Standalone library
Handling updates to BSP	Entire BSP is rebuilt. This process can be time-consuming.	Smart rebuild only recompiles changed files
Compatibility with command-line flow	None	Full compatibility with the Nios II Software Build Tools in the Nios II Command Shell
Modifying BSP properties	Limited control over BSP properties through the System Library page of the Properties dialog box.	Full control over BSP properties through the BSP Editor

Table 2. Differences Between Nios II Software Build Tools for Eclipse and Nios II IDE (Part 2 of 2)

Feature	Nios II IDE	Nios II Software Build Tools for Eclipse
Nios II C2H Compiler	Supported	Not supported
Nios II Flash Programmer	Limited control over the contents of the flash image	■ Allows you to create complex flash images containing Executable and Linking Format Files (.elf), SRAM Object Files (.sof), and binary data files in specified locations ■ Allows you to save flash programmer settings for repeatable flash programming
Scripting and regression Testing	Not Available	Project creation and build can be scripted using the Software Build Tools command line

Converting IDE Projects into Software Build Tools for Eclipse

If you have a pre-existing Nios II IDE project, before you can continue development in the Nios II Software Build Tools for Eclipse, you must convert it to the Software Build Tools and import it into the Eclipse environment. To convert and import an IDE project to the Nios II Software Build Tools for Eclipse, execute the following steps:

1. Convert the IDE project to a Software Build Tools-based project with the **nios2-convert-ide2sbt** command line utility.
2. Import the converted project into the Nios II Software Build Tools for Eclipse.

For a detailed description of the conversion process, refer to "Porting Nios II IDE Projects to the Software Build Tools" in the *Using the Nios II Integrated Development Environment* appendix of the *Nios II Software Developer's Handbook*. For a description of the process of importing a project, refer to "Importing a Command-Line Project" in the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*.

Learning about the Nios II Software Build Tools for Eclipse

Refer to the following documents to help you get acquainted with the Nios II Software Build Tools for Eclipse and related tools:

- *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*
- *Nios II Hardware Development Tutorial*
- The Comprehensive Nios II Embedded Design Suite support page
- The Nios II Software Build Tools for Eclipse support page

Errata Summary

Table 3 summarizes the issues that affect the Nios II Embedded Design Suite versions 8.1 through 9.1. For a detailed description of each issue, click on the issue name.

Refer to the Altera Knowledge Database for older errata and solutions.

Table 3. Nios II EDS Errata *(1)* (Part 1 of 4)

Added or Updated	Issue	Page	Affected Version		
			9.1	9.0	8.1
	Nios II Software Build Tools for Eclipse				
Nov 2009	Error Messages on Console When Debugging	7	✓	—	—
	java.lang.NullPointerException Error When Running Application	8	✓	—	—
	Update Failure When Setting Empty Properties in the Nios II Properties Page	8	✓	—	—
	Errors Converting Nios II IDE Multiprocessor Project	8	✓	—	—
	DHCP Time-Out on EP3C120	9	✓	—	—
	No Nios II Multiprocessor Collection Run Configuration	9	✓	—	—
	Stop on Startup Option in Run Configuration Has No Effect	9	✓	—	—
	Project Imported From Command Line Fails to Build Correctly	10	✓	—	—
	Debugger Breaks in crt0.s Instead of main()	10	✓	—	—
	C2H Compiler				
Nov 2009	Error "java.lang.NoClassDefFoundError" During C2H Compilation	11	—	Fixed	✓
	C2H Cache Coherency Issue with Global Variables	11	—	—	Fixed
Jul 2008	Functions Declared Without a Return Type Are Not Supported	12	✓	✓	✓
Oct 2007	Pre-7.1 Systems Are Not Supported	12	✓	✓	✓
	--src-dir Software Build Tools Argument Does Not Work With C2H	12	✓	✓	✓
May 2007	Accelerator Generation Failure If Tools Are Installed in Path With Spaces	12	✓	✓	✓
	The C2H Compiler Regenerates an Accelerator Unnecessarily	13	✓	✓	✓
	Error: c2h_accelerator_base_addresses.h: No such file or directory	13	✓	✓	✓
	Java Heap Space Exception if Quartus II Compilation is Enabled	13	✓	✓	✓
	Pointer Dereferences to Volatile Types	14	✓	✓	✓
Dec 2006	C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File	14	✓	✓	✓
Jun 2006	Array Elements in Structures Do Not Copy Correctly	14	✓	✓	✓
	Clean Build Causes Build Failure	15	✓	✓	✓
	Changing Build Configurations Produces Unexpected Results	15	✓	✓	✓
	Hardware Accelerators Remain After Deleting the Software Project *(2)*	16	✓	✓	✓
	Incorrect Results From Logical or Conditional Operation With Side-Effects	16	✓	✓	✓
	Launch SOPC Builder Button in C2H View	17	✓	✓	✓
	Development Boards				
Jun 2006	Intermittent Failures While Accessing CompactFlash Card	17	✓	✓	✓

Table 3. Nios II EDS Errata *(1)* (Part 2 of 4)

Added or Updated	Issue	Page	Affected Version		
			9.1	9.0	8.1
	Documentation Issues				
Nov 2009	NicheStack TCP/IP Stack - Nios II Edition Does Not Support TFTP	18	✓	✓	✓
	Error Message After Renaming Project: "Resource is out of sync with the system"	18	✓	—	—
	Obsolete Wiki URLs in Help Systems and Handbook	19	—	Fixed	✓
	Missing Source Control Information	19	—	Fixed	✓
Mar 2009	Compiler Flags for Building Custom Newlib	20	✓	✓	✓
Nov 2008	Error Message on Invalid Exception or Reset Vector	21	✓	✓	✓
Oct 2007	Nios II IDE Online Help Expand Buttons Do Not Work	22	✓	✓	✓
	Hardware Example Designs				
Mar 2007	Incorrect PLL Phase Shift in Stratix II and Cyclone II Designs	23	✓	✓	✓
	Software Examples				
Nov 2009	Hardware Tutorial Software Example Hangs on Some Boards	24	✓	✓	✓
Nov 2008	Simple Socket Server Example Displays Incorrect IP Address	24	✓	✓	✓
Jun 2006	Networking Examples	25	✓	✓	✓
	Flash Programmer				
Aug 2009	Boot Failure or elf2flash Error Due to Incorrect Flash File Size	25	✓	✓	✓
May 2007	Delay When Creating New Flash Programmer Configuration	26	✓	✓	✓
Dec 2006	elf2flash File Size Limit	26	✓	✓	✓
	Error Message "No such file or directory" When Programming Flash	26	✓	✓	✓
	Hardware Simulation				
Dec 2006	Error "UNC paths are not supported" Launching ModelSim	27	✓	✓	✓
Jun 2006	Uninitialized .bss Variables in Simulation	27	✓	✓	✓
	Host Platform				
Nov 2009	Cannot Launch Nios II Software Build Tools for Eclipse on Red Hat Linux	27	✓	—	—
Dec 2007	Windows/Cygwin: Nios II Processor Generation Failure	28	✓	✓	✓
Oct 2007	Windows Vista: Limited Support in Nios II EDS	28	✓	✓	✓
	Nios II IDE				
	Building Projects				
Mar 2009	Nios II IDE Command-Line Tools Select Wrong Workspace on Linux	29	✓	✓	✓
	Nios II IDE Command-Line Tools Hang on Windows	30	✓	✓	✓
Nov 2008	Nios II IDE Cannot Find stdio.h in Outline View	30	✓	✓	✓
Jul 2008	Nios II IDE Hangs With UNC Project Path	30	✓	✓	✓
	Build Failure with Nios II Advanced Exceptions, MMU, and MPU	31	✓	✓	✓

Table 3. Nios II EDS Errata *(1)* (Part 3 of 4)

Added or Updated	Issue	Page	Affected Version		
			9.1	9.0	8.1
May 2007	Build Command Not Functional for BSPs Created With the Nios II Software Build Tools	31	✓	✓	✓
	Linker Errors with Dual-Port Memories	32	✓	✓	✓
	User-Managed BSP Settings Not Supported in Nios II IDE	32	✓	✓	✓
	URL Project Location Causes Project Creation Error	32	✓	✓	✓
Dec 2006	Compilation Error with Separate Exception Stack Option	32	✓	✓	✓
	Incorrect Stack and Heap Free Space Report	33	✓	✓	✓
	Nios II IDE Reports Problems Without Displaying Error in Console	33	✓	✓	✓
Debugging Projects					
Nov 2009	Method for Accessing MMU and MPU Registers in the Debugger	33	✓	✓	✓
Nov 2008	IDE Cannot Display Imported Profiling Data	34	✓	✓	✓
Jul 2008	Error Message when Downloading .elf File	34	✓	✓	✓
	Variable Casting Unsupported in ISS	35	✓	✓	✓
	Trace Debug Does Not Support Instruction-Related Exceptions	35	✓	✓	✓
Oct 2007	Trace Debug Does Not Support the JMPI Instruction	35	✓	✓	✓
May 2007	Cannot Locate Source Code in Driver Files Shared by Multiple Projects	35	✓	✓	✓
	Console Window Is Not Updated After ISS Error	35	✓	✓	✓
	ISS Fails on Designs Containing Triple Speed Ethernet MAC or SG-DMA Components	36	✓	✓	✓
	Memory Window Sets Control Register Values Incorrectly	36	✓	✓	✓
Dec 2006	Programs That Interact With a Terminal Console on Windows Do Not Work	36	✓	✓	✓
	"Run as ModelSim" in the Nios II IDE Fails	37	✓	✓	✓
	The Restart Command on the Run Menu Does Not Work	37	✓	✓	✓
Jun 2006	Watchpoints Do Not Work on Certain Variables	37	✓	✓	✓
Navigating Projects					
Dec 2006	Nios II IDE Freezes While Displaying the Splash Screen	38	✓	✓	✓
	Internal Error When Double-Clicking on a Large Objdump File	38	✓	✓	✓
Jun 2006	C/C++ Scanner Does Not Support Certain C/C++ Constructs	38	✓	✓	✓
Nios II Software Build Tools					
Nov 2009	Build Errors on Interrupt API Calls	39	✓	—	—
	BSP Editor Incorrectly Allows Mixed Drivers	40	Fixed	✓	✓
	nios2-bsp --help Displays Incorrect List of BSP Types	40	Fixed	✓	✓
	nios2-bsp-create-settings Fails to List BSP Types	40	—	—	Fixed
	Makefile Error with Absolute Source Path	41	—	—	Fixed
	Incorrect Linker Errors	41	—	—	Fixed
	Java Heap Space Error When Creating BSP	42	—	Fixed	✓
Oct 2007	Software Build Tools Fail if Nios II EDS is Installed in a Path Containing Spaces	42	✓	✓	✓

Table 3. Nios II EDS Errata (1) (Part 4 of 4)

Added or Updated	Issue	Page	Affected Version		
			9.1	9.0	8.1
May 2007	User-Managed BSP Settings Not Supported in Nios II IDE	32	✓	✓	✓
	Upper-Case File Extensions Not Supported	42	✓	✓	✓
	Peripherals				
Nov 2009	Software Build Tools Fail With UART Driver	43	—	—	Fixed
Jul 2008	Unaligned Transfers of Small Payloads Fail on SG-DMA	43	✓	✓	✓
May 2007	DMA Controller Always Busy in Burst Mode	43	✓	✓	✓
	Non-System-Wide Reset Can Cause Improper Initialization of Mailbox Core	44	✓	✓	✓
	Target Software				
Nov 2009	NicheStack tcp_wake() Function Does Not Awaken All select() Sleepers	44	✓	—	—
	EIC-Based Nios II System Crashes With NicheStack	45	✓	—	—
Mar 2009	Gigabit Ethernet Performance Issues	45	—	Fixed	✓
Dec 2006	stdio Does Not Work with MicroC/OS-II and Small C Library	45	✓	✓	✓
Jun 2006	cout From MicroC/OS-II Task Does Not Send Data to stdout	45	✓	✓	✓
	malloc(), realloc() Failures With MicroC/OS-II	46	✓	✓	✓
	Toolchain (gcc, gdb, etc.)				
Nov 2009	sof2flash Names Output Files Incorrectly	46	—	—	Fixed
Dec 2006	GNU Assembler Does Not Accept the --defsym Flag	47	✓	✓	✓

Notes to Table 3:

(1) Refer to the Altera Knowledge Database for older errata and solutions.

(2) For further information, refer to the *Nios II C2H Compiler User Guide*.

Nios II Software Build Tools for Eclipse Errata

This section describes in detail the issues related to the Nios II Software Build Tools for Eclipse.

Error Messages on Console When Debugging

When debugging, you might see the following messages in the Nios II Software Build Tools for Eclipse console:

```
No symbol "auto" in current context.
No symbol "new" in current context.
```

These are normal messages from the GNU Debugger (GDB) console, not error messages. This issue does not impact the debug process.

Workaround

No workaround needed.

Solution Status

None needed.

java.lang.NullPointerException Error When Running Application

When you attempt to run or debug an application, the Nios II Software Build Tools for Eclipse might show an internal error message similar to the following:

```
Launching New_configuration java.lang.NullPointerException
```

This error occurs in either of the following circumstances:

- You attempt to run your project without rebuilding after performing a `make clean`.
- You delete a run configuration belonging to your project while the project is running.

You must restart Nios II Software Build Tools for Eclipse to recover from this error.

Workaround

Build the project before attempting to run or debug it. Do not delete a run configuration while it is in use.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Update Failure When Setting Empty Properties in the Nios II Properties Page

If you use the Nios II Properties page to update more than one BSP setting to the empty string (`""`), the BSP is not updated.

Workaround

You can use either of the following workarounds to set multiple BSP settings to the empty string:

- Use the keyword `none` instead of `""` to set settings to the empty string.
- Use the BSP Editor instead of the Nios II Properties page.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Errors Converting Nios II IDE Multiprocessor Project

If you try use the **nios2-convert-ide2sbt** utility to convert an IDE multiprocessor project to the Nios II Software Build Tools, error messages appear, and the project fails to generate properly.

Workaround

Create new Nios II Software Build Tools application and BSP projects using project settings equivalent to the original Nios II IDE project settings.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

DHCP Time-Out on EP3C120

If you create a project based on the web server template targeting the Altera Embedded Systems Development Kit, Cyclone® III Edition (EP3C120) board, the system is unable to get an IP address at startup. DHCP times out and the system reverts to the default IP address.

Workaround

Create the project in the Nios II Command Shell and then import it to the Nios II Software Build Tools for Eclipse, by executing the following steps:

1. In the Nios II Command Shell, navigate to the following path:

   *<Nios II EDS install path>***/examples/***<language>***/niosII_cycloneIII_3c120/tserd/software_examples/app/web_server_tse_3c120_rgmii**

   where *<language>* is either **vhdl** or **verilog**.

2. Type the following command to create a web server example for the EP3C120 kit:

   ```
   create-this-app↵
   ```

3. Import the command-line project to the Nios II Software Build Tools for Eclipse.

   For information about importing command-line projects, refer to "Importing a Command-Line Project" in the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

No Nios II Multiprocessor Collection Run Configuration

The multiprocessor collection run configuration is not available in the Run menu. Therefore, you cannot use a single run configuration to download **.elf** files to run on multiple Nios II processors in the Nios II Software Build Tools for Eclipse.

Workaround

Create a separate run configuration for each Nios II processor in the hardware design, and launch them one at a time

Solution Status

Not fixed

Stop on Startup Option in Run Configuration Has No Effect

Turning on **Stop on Startup at** in the **Debugger** tab of a run configuration has no effect.

Workaround

You can manually set any breakpoint in the debug perspective.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Project Imported From Command Line Fails to Build Correctly

If you try to import a command-line project into Nios II Software Build Tools for Eclipse, and your project has source files located outside the project folder, the Nios II Software Build Tools for Eclipse fail to build your project because the makefile does not include source files located outside the project folder.

Workaround

You can work around this issue with either of the following methods:

- When importing the project, disable source management through the **Nios II Application Properties** or **Nios II Library Properties** dialog box.

  ☞ You must manually keep your makefile updated to reflect source files added to or removed from the project.

- Manually link the affected source files to the project. Right-click the project, click **Properties**, expand **C/C++ General**, select **Paths and Symbols** and then **Source Location**, and click **Link Folder**.

Solution Status

This issue will be fixed in future release of the Nios II EDS.

Debugger Breaks in crt0.s Instead of main()

If you attempt to launch a debug session by right clicking the project, without first creating a run configuration, the debugger breaks in the initialization code in **crt0.s**, rather than at the top of the `main()` function. You must explicitly create the run configuration before launching the debug session.

Workaround

You can work around this issue with either of the following methods:

- Create the run configuration with the **Run Configuration** dialog box. You can then debug with this run configuration by right-clicking the project, clicking **Debug As**, and selecting the debug configuration. The program correctly breaks at `main()`.
- Manually set a breakpoint at `main()`.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

C2H Compiler Errata

This section describes in detail the issues related to the Nios II C2H Compiler.

Error "java.lang.NoClassDefFoundError" During C2H Compilation

C2H compilation fails with the following error:

```
java.lang.NoClassDefFoundError: com/altera/c2h/C2Hgenerate
Exception in thread "main" make: *** [c2h_hdl-t] Error 1
```

This error occurs when you invoke C2H from the Nios II IDE on the Windows operating system. This problem does not affect Linux installations, nor C2H compilation from the Nios II command shell.

Workaround

To correct the problem, visit mySupport and request the Nios II EDS 8.1 Patch 0.01 (PC version).

Solution Status

Fixed in v. 9.0

C2H Cache Coherency Issue with Global Variables

Cache coherency issues might be encountered when the following conditions are all true:

- A C2H-accelerated function writes to a global or static variable
- The Nios II processor is configured with a data cache
- The referenced global variable is located in cacheable memory

In this situation, the data in the global variable can be corrupted.

This issue can only arise if the accelerated function writes to the global or static variable. If the function merely reads from the variable, cache coherency is maintained correctly.

Workaround

Add a data cache flush operation in the accelerator's wrapper function. The wrapper can be found in **alt_c2h_<*function_name*>.c**. In this file, find the following comment:

```
/* Write 1 to address 0 starts the accelerator */
```

Directly after this comment, add the following function call:

```
alt_dcache_flush_all();
```

This function call ensures that the data cache is flushed immediately before the accelerator is started.

☞ Flushing the cache entails a performance penalty.

Alternatively, allocate your global and static variables in non-cacheable regions of memory. Placing these variables in non-cacheable regions eliminates any need to perform a data cache flush.

For information about allocating variables to non-cacheable regions, refer to the *Cache and Tightly-Coupled Memory* chapter of the *Nios II Software Developer's Handbook*.

Solution Status

Fixed in v. 8.1

Functions Declared Without a Return Type Are Not Supported

The C2H compiler does not support functions without an explicitly declared return type.

Workaround

If you are using the implicit `int` return type, declare the return type explicitly. If your function has no return value, declare it as `void`.

Solution Status

Not fixed

Pre-7.1 Systems Are Not Supported

Starting in v. 7.1, the C2H Compiler does not support SOPC Builder systems created in SOPC Builder 7.0 or earlier (systems based on a **class.ptf** file).

Workaround

Before using the C2H Compiler, you must upgrade your SOPC Builder system. Open the system in SOPC Builder v. 7.1 or later. SOPC Builder prompts you to convert the system, creating a ***_hw.tcl** file compatible with the C2H Compiler.

Solution Status

Not fixed

--src-dir Software Build Tools Argument Does Not Work With C2H

When using the C2H Compiler with Nios II Software Build Tools, an error occurs when using the `nios2-app-generate-makefile` argument `--src-dir` to specify a directory of source files. An example of the error is:

```
multiple definition of 'my_accelerated_function'
```

Workaround

Specify source files individually using the `--src-files` argument.

Solution Status

Not fixed

Accelerator Generation Failure If Tools Are Installed in Path With Spaces

If the path to your installation of the Altera Design Suite contains spaces, the C2H Compiler fails to generate the accelerators.

Workaround

Reinstall the tools to a path containing no spaces.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

The C2H Compiler Regenerates an Accelerator Unnecessarily

The C2H Compiler might regenerate an accelerator even when the accelerated function is unchanged. This problem can result from changes to files included by the C file containing the accelerated function. The C2H Compiler fails to check that the generated hardware description language (HDL) matches the previously generated HDL causing the system to be regenerated.

Workaround

To avoid this issue move the information in the include file specific to the hardware accelerator into a separate include file. This workaround prevents regeneration of the system when the HDL is unchanged.

Solution Status

Not fixed

Error: c2h_accelerator_base_addresses.h: No such file or directory

When a C2H accelerator is compiled for the first time, the following compile-time error can result if the **Analyze all accelerators** option is selected:

```
c2h_accelerator_base_addresses.h: No such file or directory.
```

Workaround

Click **Build software and generate SOPC Builder system** and build once before building with the **Analyze all accelerators** option.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Java Heap Space Exception if Quartus II Compilation is Enabled

If your design contains a C2H accelerator, and you select **Build software, generate SOPC Builder system, and run Quartus® II compilation** in the C2H view, you might see the following error during Quartus II compilation:

```
Exception in thread "main" java.lang.OutOfMemoryError: Java
heap space

make: *** [c2h_hdl-t] Error 1
```

Workaround

Select the **Build software and generate SOPC builder system** option in the C2H Compiler settings window, and then manually launch the Quartus II software to compile the design.

Solution Status

Not fixed

Pointer Dereferences to Volatile Types

The C2H Compiler treats pointer dereferences to a volatile type as if they alias all other pointer dereferences. Pointers that are restrict-qualified are treated the same way.

The two loops in Example 1 cannot be scheduled concurrently because the `volatile` qualification overrides the `__restrict__` pragma.

Example 1. Non-Concurrent Loops

```
volatile int * __restrict__ fifo_rd = FIFO_RD_BASE;
volatile int * __restrict__ fifo_wr = FIFO_WR_BASE;
for ()
{
  *fifo_wr = ....;
}
for ()
{
  ... = *fifo_rd;
}
```

Workaround

Divide the function into multiple interrupt request (IRQ)-enabled accelerators that are launched concurrently from the processor, and use FIFO buffers to communicate between them.

Solution Status

Not fixed

C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File

When accelerating a function in a file, the C2H Compiler cannot link subfunctions that are defined in a different file.

Workaround

Include all subfunctions called by the accelerated function within the same source code file.

Solution Status

Not fixed

Array Elements in Structures Do Not Copy Correctly

C2H accelerators do not correctly copy array elements that are elements of structures.

In Example 2, the `a` and `b` elements of the structure copy correctly, but the `buf` element does not. After this assignment, `struct_a` equals `{9, 8, {3, 3, 3, 3}}`.

Example 2. Array Elements of Structs

```
typedef struct my_struct {
  int a;
  int b;
  int buf[BUF_SIZE];
}MY_STRUCT;
MY_STRUCT struct_a = {1, 2, {3, 3, 3, 3}};
MY_STRUCT struct_b = {9, 8, {7, 7, 7, 7}};
struct_a = struct_b;
```

Workaround

Copy the array elements explicitly, as shown in Example 3.

Example 3. Copying Array Elements Explicitly

```
{
  int i=0;
  do
    {
      struct_a.buf[i] = struct_b.buf[i];
      i++;
    } while (i<LENGTH_OF_BUF_ELEMENT)
}
```

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Clean Build Causes Build Failure

Performing a clean build on a Nios II IDE project that contains a hardware accelerator can cause the next build to fail in the IDE, because the clean build erroneously deletes a file required by the C2H Compiler.

Workaround

Do not perform a clean build on projects that use hardware accelerators. If you have already performed a clean build, recompile with option **Build software, generate SOPC Builder system, and run Quartus II compilation** to regenerate the necessary files.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Changing Build Configurations Produces Unexpected Results

The C2H Compiler does not support multiple build configurations (for example Release or Debug) in the Nios II IDE. After creating one or more accelerators in a particular configuration, the C2H Compiler produces undefined results if you switch to a different build configurations and create more accelerators.

Workaround

For a specific SOPC Builder system and Nios II IDE project, specify C2H accelerators in only one build configuration. You can use multiple build configurations, as long as only one configuration specifies C2H Compiler settings.

Solution Status

Not fixed

Hardware Accelerators Remain After Deleting the Software Project

If a system contains C2H accelerators, deleting the software project that defines the accelerators does not remove the accelerators from the hardware system, and the accelerator logic remains in the SOPC Builder system.

Workaround

To remove an accelerator from a system, delete the accelerator from the C2H view in the Nios II IDE first, and then recompile the software project. The C2H Compiler then removes the accelerator from the SOPC Builder system. Once the compilation is complete then the software application can be deleted from the workspace.

Solution Status

Not fixed

Incorrect Results From Logical or Conditional Operation With Side-Effects

The C2H Compiler always evaluates both operands of logical (&&, `||`) and conditional (`?:`) operators. This is different from expected American National Standards Institute (ANSI) C behavior, where operands are evaluated left-to-right, and unnecessary operands are skipped.

For example, in the expression `(i-- && j--)`, if the value of `i` is zero, ANSI C does not evaluate the right-hand-side (RHS) expression, and `j` is not decremented. By contrast, the C2H Compiler evaluates both sides, decrementing `j`.

The following expressions are other examples that might be affected by this issue:

```
if (i-- || j++)
    ...;


a = ((cond == 1)?  i++ : j++);
```

Workaround

Use logical and conditional operations whose operators have no side effects. Operations with side effects include pre- and post- increment and decrement operations (`++`, `--`), memory operations (`*`, `[]`, `.`, `->`), and function calls.

Solution Status

Not fixed

Launch SOPC Builder Button in C2H View

When the Nios II IDE workspace contains multiple projects with multiple system libraries, the incorrect SOPC Builder system might open when you click **Launch SOPC Builder** in the C2H view.

Workaround

Launch SOPC Builder from the Quartus II software. Alternatively, keep only one system library project open at a time while using the C2H Compiler.

Solution Status

Not fixed

Development Board Errata

This section describes in detail the Nios II EDS issues related to Nios development boards.

Intermittent Failures While Accessing CompactFlash Card

The Nios II Development Kit version 5.0 and later includes a CompactFlash controller peripheral suitable for interfacing to CompactFlash cards in True IDE mode on Nios development boards. For True IDE mode to operate, CompactFlash cards require that the `ATASEL_N` input be driven to ground during power-up.

The CompactFlash controller peripheral includes a configurable power register, used to cycle power to CompactFlash cards from Nios II software through a metal oxide semiconductor field-effect transistor (MOSFET) on the Nios development boards. However, in certain development boards, power to the CompactFlash card does not turn off completely during this power cycle operation. Because of this condition, the CompactFlash might not sample the `ATASEL_N` pin during the power-cycle operation after FPGA configuration when this pin is driven to ground. Instead, the CompactFlash card might sample the `ATASEL_N` pin when power is first applied to the development board, when I/O is not yet driven by the FPGA (before FPGA configuration).

Workaround

If you encounter errors with CompactFlash when using the Nios development boards, try one of the following solutions:

- Use a different CompactFlash card. Certain cards are more susceptible to the power-cycling issue than others.
- Modify the Nios development board. This is recommended if you are familiar and comfortable with board-level modifications. Disconnect pin 9 (`ATASEL_N`) on the CompactFlash socket on your Nios development board and tie this pin to ground.

☞ The CompactFlash socket uses a staggered numbering on the pins (starting from pin 1: 1, 26, 2, 27, ...); refer to the CompactFlash Association specification for right-angle surface-mount connectors for exact specifications on this connector. This modification permanently enables True IDE mode operation.

Solution Status

Not fixed

Documentation Errata

This section describes in detail the Nios II EDS documentation issues.

NicheStack TCP/IP Stack - Nios II Edition Does Not Support TFTP

The Nios II documentation should include the following information:

Versions of the NicheStack TCP/IP Stack other than the Nios II Edition include optional Trivial File Transfer Protocol (TFTP) client and server applications. However, the TFTP client and server are not thread safe. They cannot be used in systems with the MicroC/OS-II RTOS.

TFTP is not available with the NicheStack TCP/IP Stack - Nios II Edition. The TFTP client and server features are disabled.

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Error Message After Renaming Project: "Resource is out of sync with the system"

The *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook* does not explain the correct procedure for renaming a project.

Workaround

The missing information is as follows:

To rename a project in the Nios II Software Build Tools for Eclipse, execute the following steps:

1. Right click the BSP project and click **Rename**.
2. Type the new BSP name.
3. Right click the BSP project and click **Refresh**.

☞ If you neglect to refresh the project, you might see the following error message when you attempt to build it:

```
Resource <original_bsp_name> is out of sync with the system
```

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Obsolete Wiki URLs in Help Systems and Handbook

The following documents contain obsolete URLs for the Nios Community Wiki:

- The *Nios II Software Developer's Handbook*
- The *Nios II Embedded Design Handbook*
- The Nios II IDE help system

Workaround

You can visit the Nios Community Wiki at www.nioswiki.com.

Solution Status

Fixed in the documentation for v. 9.0.

Missing Source Control Information

The Quartus II Handbook refers to the *Using the Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook* for information about putting Nios II projects in source control. The referenced chapter only describes the process for Nios II Software Build Tools projects.

Workaround

The missing information follows.

Archiving Nios II IDE Software Projects

This section helps you identify the files you must include when archiving a Nios II IDE software project. With this information, you can archive a Nios II application project and its associated Nios II system library project.

You might want to archive your projects for one of the following reasons:

- To place them under source control
- To create backups
- To bundle the projects for transfer to another location

This section covers the following information:

- How to find and identify the files that you must include in an archived Nios II IDE software project.
- Which files must have write permission to allow the software projects to be built.

Table 4 and Table 5 list the files required by Nios II IDE software projects. This is the minimum set of files needed to completely rebuild a software project, including the executable and linking file (**.elf**).

Archive your Nios II IDE software projects together with the SOPC Builder system on which they are based. You cannot rebuild a Nios II IDE software project without its associated SOPC Builder system.

The files listed in Table 4 are located in the Nios II application project directory.

Table 4. Files Required for a Nios II Application Project

File Description	File Name	Write Permission Required?
All source files	for example: **app.c**, **header.h**, **assembly.s**, **lookuptable.dat**	No
Eclipse project file	**.project**	No
C/C++ Development Toolkit project file	**.cdtproject**	Yes
C/C++ Development Toolkit option file	**.cdtbuild**	No
Software configuration file	**application.stf**	No

The files listed in Table 5 are located in the Nios II system library project directory.

Table 5. Files Required for a Nios II System Library Project

File Description	File Name	Write Permission Required?
Eclipse project file	**.project**	Yes
C/C++ Development Toolkit project file	**.cdtproject**	Yes
C/C++ Development Toolkit option file	**.cdtbuild**	No
System software configuration file	**system.stf**	Yes

☞ You must have write permission for certain files. The tools write to these files as part of the build process. If the files are not writable, the toolchain fails. Many source control tools mark local files read-only by default. In this case, you must override this behavior. You do not have to check the files out of source control unless you are modifying the Nios II software project.

Solution Status

In the documentation for v. 9.0, the missing information appears in the *Nios II Integrated Development Environment* chapter of the *Nios II Software Developer's Handbook.* In the documentation for v. 9.1, the missing information appears in the *Using the Nios II Integrated Development Environment* appendix of the *Nios II Software Developer's Handbook.*

Compiler Flags for Building Custom Newlib

"Common BSP Tasks" in the *Using the Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook* describes how to compile a custom version of the newlib library. However, it does not list the compiler flags that must be selected to generate the correct library.

Workaround

The missing information is as follows:

The Nios II EDS provides a number of precompiled newlib implementations. The provided libraries are precompiled with every viable combination of the GCC compiler flags shown in Table 6.

When you create a BSP with a precompiled newlib, the Nios II Software Build Tools select the newlib matching your BSP's compiler settings. When you create a custom newlib, you must ensure that the compiler flags listed in Table 6 match your BSP's settings.

Table 6. GCC Compiler Options for Newlib

Option Name	Purpose
`-pg`	Link for profiling with gprof
`-EB`	Generate big-endian code
`-mstack-check`	Enable stack checking
`-mno-hw-mul`	Disable use of the `mul` family of instructions
`-mhw-mulx`	Enable use of the `mulx` family of instructions
`-mcustom-fpu-cfg`	Use a floating point custom instruction

The Nios II-specific compiler flag `-mcustom-fpu-cfg` requires the Nios II processor core to be implemented with the floating-point custom instruction as specified in the flag argument. The compiler flag can have one of the following two argument values:

- `-mcustom-fpu-cfg=60-1`—Use the Nios II floating-point custom instruction without divider
- `-mcustom-fpu-cfg=60-2`—Use the Nios II floating-point custom instruction with divider

For details about the Nios II-specific compiler flags `-mstack-check`, `-mno-hw-mul`, and `-mhw-mulx`, refer to "Altera Nios II Options" in *Using the GNU Compiler Collection (GCC)*, installed with the Nios II EDS. For details about the `-pg` compiler flag, refer to "Compiling a Program for Profiling" in *GNU Profiler*, installed with the Nios II EDS. To find the installed documentation, on the Windows Start menu, click **Programs** > **Altera** > **Nios II EDS <*version*>** > **Nios II <*version*> Documentation**, and then click **Literature**.

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Error Message on Invalid Exception or Reset Vector

Nios II compiler error messages, added in v 8.1, are not documented in the *Nios II Software Developer's Handbook*.

Workaround

The missing information is as follows:

While building a Nios II software project, the compiler produces an error message if the SOPC Builder system defines an invalid exception or reset vector. Table 7 shows possible error messages and their causes.

In the SOPC Builder system, ensure that the exception and reset vectors conform to all the criteria listed in Table 7.

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Table 7. Exception and Reset Vector Error Messages

Message	Cause
`The section mapping ".entry" is not mapped to a memory region.`	The `.entry` section is mapped to a nonexistent memory region.
`The section mapping ".exceptions" is not mapped to a memory region.`	The `.exceptions` section is mapped to a nonexistent memory region.
`The section mapping ".entry" maps to a memory region "<`*`region name`*`>" which is not the required length "32".`	The `.entry` section is mapped to a memory region that is smaller than 32 bytes.
`The section mapping ".entry" maps to a memory region "<`*`region name`*`>" which does not start at the SOPC Design specified address "<`*`base address`*`>".`	The `.entry` section is mapped to a memory region that does not start at the reset vector base address.
`The section mapping ".exceptions" maps to a memory region "<`*`region name`*`>" which does not start at the SOPC Design specified address "<`*`base address`*`>".`	The `.exceptions` section is mapped to a memory region that does not start at the exception vector base address.
`The SOPC design Nios II reset and exception vector addresses are mapped to the same memory. For HAL the exception address must be at least 32 bytes larger than the reset address.`	The `.entry` section and `.exceptions` section map to the same memory device, and the base of the `.exceptions` section is less than 32 bytes above the base of the `.entry` section.
`The section mapping ".exceptions" mapped device "<`*`memory 1`*`>" is not the same as the SOPC Design specified device "<`*`memory 2`*`>".`	The `.exceptions` section is mapped to a device that is different from the exception vector device specified in the SOPC Builder system.
`The section mapping ".entry" mapped device "<`*`memory 1`*`>" is not the same as the SOPC Design specified device "<`*`memory 2`*`>".`	The `.entry` section is mapped to a device that is different from the reset vector device specified in the SOPC Builder system.

Nios II IDE Online Help Expand Buttons Do Not Work

The expand text arrow and the **Show All** option in the online help do not work in Internet Explorer.

Workaround

In Internet Explorer, carry out the following actions:

1. Click the **Refresh/Show Current Topic** icon (yellow arrows icon at the top of the search pane). The table of contents appears with the selected topic highlighted.
2. Click the highlighted topic to refresh the browser frame. Expanded text works.

Alternatively, use Firefox 2.0.0.6.

Solution Status

Not fixed

Hardware Example Design Errata

This section describes in detail the Nios II EDS hardware example design issues.

Incorrect PLL Phase Shift in Stratix II and Cyclone II Designs

In several example designs targeting the Nios II Cyclone II 2C35 and Nios II Stratix® II 2S60 RoHS development boards, the synchronous static random access memory (SSRAM) and legacy DDR SDRAM clocks are implemented with an incorrect phase-locked loop (PLL) phase shift.

This issue affects the following example designs:

- 2s60 RoHS Triple Speed Ethernet Scatter-Gather DMA (SG-DMA)
- 2s60 RoHS Standard
- 2s60 RoHS Full Featured
- 2c35 Triple Speed Ethernet SG-DMA
- 2c35 Standard
- 2c35 Full Featured

This issue also affects any custom 2c35 or 2s60 RoHS design using SSRAM or legacy DDR SDRAM.

The incorrect phase shift might cause timing violations when reading from or writing to SSRAM or legacy DDR SDRAM.

Workaround

To correct the PLL settings, perform the following steps:

1. Open the standard design in SOPC Builder.
2. Double-click the component instance named `pll` to launch the PLL MegaWizard™ interface.
3. Click **Launch Altera's ALTPLL MegaWizard** to launch the MegaWizard interface.
4. Select the **Output Clocks** page and then the clock page used to generate the SSRAM clock. Depending on the particular design, this might be **clk c1**, **clk c2**, or another clock.
5. Ensure that **Clock phase shift** is –3.38 ns.
6. Click the clock page used to generate the write clock for the legacy DDR SDRAM, if any. Depending on the particular design, this might be **clk c1**, **clk c2**, or another clock.
7. Ensure that **Clock phase shift** is 270 degrees.
8. Click **Finish** to save changes and exit the PLL MegaWizard interface.
9. Click **Finish** to save settings to the PLL instance in SOPC Builder.
10. Regenerate the system in SOPC Builder and recompile in the Quartus II software. Refer to the **readme.txt** file in the design example folder for information on compiling designs containing the DDR controller.

For further information refer to **ssram_interface_readme.html**, located in the ***<Quartus II installation directory>*/sopc_builder/documents** folder. This document discusses the SSRAM timing analysis methodology in detail. Additionally, *AN 411: Understanding PLL Timing for Stratix II Devices* discusses clock phase shift calculations and assignments for PLLs in Stratix II devices.

Solution Status

Not fixed

Software Example Errata

This section describes in detail the Nios II EDS software example issues.

Hardware Tutorial Software Example Hangs on Some Boards

The **count binary** application used in the tutorial hangs when run on some Nios boards. The software runs through the binary count once and stops responding.

This issue affects boards that do not have a display to handle the LCD output. The affected boards include the following:

- Embedded Systems Development Kit, Cyclone III Edition (EP3C120)
- Stratix IV GX FPGA Development Kit (EP4SGX230)

Workaround

Modify line 18 of the **count_binary.h** file, as follows:

```
 # define LCD_PRINTF(lcd, args...) /* Do Nothing */
```

Rebuild and run the software again.

Solution Status

Not fixed

Simple Socket Server Example Displays Incorrect IP Address

The simple socket server networking software example displays an incorrect IP address for the incoming connection.

Workaround

There is currently no workaround for this problem. Ignore the IP address of any incoming connection that is displayed on the console.

Solution Status

Not fixed

Networking Examples

If you are running a networking software example, you might be asked for a nine-digit number. You are directed to find this number on a sticker on your Nios development board, identified by the prefix **ASJ**. Not all Nios development boards have this sticker.

Workaround

If your Nios development board does not have a sticker with the letters **ASJ** followed by a nine-digit number, enter a unique nine-digit number when prompted. To avoid network address conflicts, ensure that this number is unique to each Nios board connected to your network.

Solution Status

Not fixed

Flash Programmer Errata

This section describes in detail the Nios II EDS issues related to the flash programmer.

Boot Failure or elf2flash Error Due to Incorrect Flash File Size

The `sof2flash` utility fails to recognize certain devices. As a result, it generates Motorola S-record (**.flash**) files of the wrong size.

With erasable programmable configurable serial (EPCS) memory, the `elf2flash --after` command fails to recognize the **.flash** file size. The resulting error prevents you from programming flash memory.

With Common Flash Interface (CFI) memory, you can program the **.flash** file, but the Nios II system fails to boot.

This error occurs when programming flash memory for the following devices:

- The following Stratix III devices:
  - EP3SL50
  - EP3SL50F484
  - EP3SL50F780
  - EP3SL110
  - EP3SL110F1152
  - EP3SL110F780
  - EP3SE80
  - EP3SE80F1152
  - EP3SE80F780
- All Stratix IV devices
- All Arria® II GX devices

Workaround

There is no workaround at this time.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Delay When Creating New Flash Programmer Configuration

You might experience a delay of several seconds when creating a new Flash Programmer configuration in the Nios II IDE.

Workaround

There is no workaround.

Solution Status

Not fixed

elf2flash File Size Limit

The **elf2flash** utility supports **.elf** files up to approximately 24 MBytes in size. The **elf2flash** utility might fail on files larger than 24 MBytes, with the error message `java.lang.OutOfMemoryError`.

Workaround

Lower the number of symbols in your **.elf** file by turning off debug symbols.

Alternatively, specify less initialized data in the application.

Solution Status

Not fixed

Error Message “No such file or directory” When Programming Flash

When programming flash from a project stored in a path containing spaces, you might receive the spurious error message **No such file or directory**. The flash programmer does not correctly handle spaces in the directory path. However, this error is harmless, because flash programming completes successfully.

Workaround

None required.

Solution Status

Not fixed

Hardware Simulation Errata

This section describes in detail the Nios II EDS issues related to hardware simulation.

Error "UNC paths are not supported" Launching ModelSim

If you launch ModelSim® from a working directory that is mapped via a universal naming convention (UNC) path (a path that starts with `//` instead of drive letter), you receive the following error message in SOPC Builder: **UNC paths are not supported. Defaulting to Windows directory**. This error occurs because ModelSim is calling a command shell, which does not support UNC paths.

Workaround

Map the UNC path to a drive letter and use the drive letter to reference the working directory in the launching shell.

Solution Status

Not fixed

Uninitialized .bss Variables in Simulation

If your program reads the value of an uninitialized `.bss` variable during HDL simulation, and the BSP (system library) is compiled with the **ModelSim only, no hardware support** property enabled in Nios II IDE, a warning appears about unfiltered data being `'x'`. This warning appears because when this property is enabled, the code that clears the `.bss` memory region is omitted to speed up HDL simulation so this memory region is uninitialized. The `.bss` region contains global and static local variables that are not initialized by the application so they default to a value of zero. When the Nios II processor reads uninitialized variables, it displays a warning and converts any of the bits of the uninitialized data to zero which correctly mimics the effect of the missing `.bss` clearing code. The Hardware Abstraction Layer (HAL) code that executes before and after `main()` might use `.bss` variables, so these warnings might appear even if your application does not use the `.bss` section.

Solution Status

Not fixed

Host Platform Errata

This section describes in detail the Nios II EDS issues related to the host development platform.

Cannot Launch Nios II Software Build Tools for Eclipse on Red Hat Linux

You might be unable to launch Nios II Software Build Tools for Eclipse on the Red Hat Linux operating system. This is an issue with the version of XULRunner in Eclipse 3.4. The issue is resolved in XULRunner 1.9.1, which is available with Eclipse 3.5.

Workaround

Upgrade XULRunner to a newer version. To determine what version of XULRunner you have, type the following command at the command prompt:

```
xulrunner -v↵
```

To upgrade XULRunner using the **yum** software package manager, type the following command:

```
sudo yum update xulrunner↵
```

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Windows/Cygwin: Nios II Processor Generation Failure

You might see the following error message when you generate your Nios II system in SOPC Builder:

```
Error: Generator program for module 'cpu_fpoint' did NOT run
successfully
```

This error might occur on a Windows-based system when all of the following conditions are true:

- You have installed a version of Cygwin other than the one distributed with the Quartus II software.
- You launch the Quartus II software and SOPC Builder from a Nios II Command Shell.
- You enable the floating point custom instruction (FPCI) in the Nios II processor core.

Workaround

Launch the Quartus II software and SOPC Builder from the Windows Start menu, and regenerate your system.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Windows Vista: Limited Support in Nios II EDS

The Quartus II software introduces Windows Vista (32-bit and 64-bit) support in version 7.2. However, the Nios II Embedded Design Suite supports only the following platforms:

- Windows 2000
- Windows XP
- Windows XP (64)
- Windows Vista Business Edition (32-bit)
- SUSE 9 (32-bit)
- SUSE 9 (64-bit)

- Red Hat Linux v3.0 (32-bit)
- Red Hat Linux v3.0 (64-bit)
- Red Hat Linux v4.0 (32-bit)
- Red Hat Linux v4.0 (64-bit)

Workaround

There is no workaround.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Nios II IDE Errata: Building Projects

This section describes in detail the issues related to building projects in the Nios II IDE.

Nios II IDE Command-Line Tools Select Wrong Workspace on Linux

On Linux systems, the Nios II IDE command-line tools use the default Eclipse project workspace, regardless what workspace you are using in the IDE. The default workspace is at `$SOPC_KIT_NIOS2/eclipse/nios2-ide-workspace-<version>`.

This issue affects the following tools:

- `nios2-create-system-library`
- `nios2-create-application-project`
- `nios2-build-project`
- `nios2-import-project`
- `nios2-delete-project`

Workaround

Explicitly supply the Nios II IDE command-line tools with a workspace location, by means of the `-data` command-line argument. The syntax of the `-data` argument is as follows:

```
-data <path to workspace>
```

The path to the workspace must be absolute, and must not contain whitespace. Aside from these restrictions, any valid file system path can be used for the workspace. Example 4 shows how to specify a workspace path.

Example 4. Providing a Non-Default Workspace Location in Linux

```
nios2-create-project -data $HOME/myworkspace <other arguments>↵
```

Solution Status

Not fixed

Nios II IDE Command-Line Tools Hang on Windows

On Windows systems, the Nios II IDE command-line tools are sometimes unable to locate the project workspace. When this happens, the tools might hang.

This issue potentially affects the following tools:

- `nios2-create-system-library`
- `nios2-create-application-project`
- `nios2-build-project`
- `nios2-import-project`
- `nios2-delete-project`

Workaround

Explicitly supply the Nios II IDE command-line tools with a workspace location, by means of the `-data` command-line argument. The syntax of the `-data` argument is as follows:

```
-data <path to workspace>
```

The path to the workspace must be absolute, and must not contain whitespace. Aside from these restrictions, any valid file system path can be used for the workspace. The workspace shown in Example 5 is the default workspace used by the Nios II IDE in version 9.0.

Example 5. Providing a Workspace Location

```
nios2-create-project -data c:/altera/90/nios2eds/\
    bin/eclipse/nios2-ide-workspace-9.0 <other arguments>
```

Solution Status

Not fixed

Nios II IDE Cannot Find stdio.h in Outline View

If you create a new project using the Nios II IDE and try to open the **stdio.h** file from the Outline view before building the project, the IDE displays the error message **No include files were found that matched that name**.

Workaround

Build the project before attempting to open **stdio.h**.

Solution Status

Not fixed

Nios II IDE Hangs With UNC Project Path

In the **New Project** dialog box, if you turn on **Specify Location** and specify the path in UNC form, the IDE might hang.

Workaround

Map the UNC path into a remote drive which looks like a Windows drive to the Nios II IDE.

Solution Status

Not fixed

Build Failure with Nios II Advanced Exceptions, MMU, and MPU

Projects created in the Nios II IDE for an SOPC Builder system containing a Nios II processor configured with advanced exceptions, the memory management unit (MMU), or the memory protection unit (MPU) fail to build, and generate the following error messages:

```
ERROR - Classic build flow for Nios II system library does not support
the Nios II \
    advanced exceptions.
Use the non-classic Nios II Board Support Package instead.
<timestamp> - (SEVERE) generate: java.lang.IllegalStateException: \
    java.lang.IllegalStateException: \
    com.altera.ingenuous.GTFElement.GTFElementErrorException: \
    <error> element in GTF script
make[1]: *** [system_description/../obj/system.h-t] Error 1
make: *** [system_project] Error 2
```

Workaround

Create the software project with the Nios II Software Build Tools command-line development flow. Alternatively, use the Nios II Software Build Tools for Eclipse.

Solution Status

Not fixed

Build Command Not Functional for BSPs Created With the Nios II Software Build Tools

The build option in the Nios II IDE menu does not rebuild BSPs imported into the IDE.

Workaround

The Nios II Software Build Tools for Eclipse can both build and debug projects created on the command line. The Nios II Software Build Tools for Eclipse is the preferred tool for debugging Nios II Software Build Tools projects.

☞ For information about the Nios II Software Build Tools for Eclipse, refer to the *Getting Started with the Graphical User Interface* chapter of the *Nios II Software Developer's Handbook*.

In the Nios II IDE, you can build the BSP by building the associated application project.

Solution Status

Not fixed

Linker Errors with Dual-Port Memories

If your instruction master and data master ports are connected to the same dual-port memory and the ports have different addresses, your code fails to run or you experience a linker error. The Nios II IDE does not warn you of the addressing violation.

Workaround

Assign the same address to both ports of the dual-port memory.

Solution Status

Not fixed

User-Managed BSP Settings Not Supported in Nios II IDE

For projects created with the Nios II Software Build Tools and imported into the Nios II IDE, the IDE configuration settings have no effect.

For example, objdump, compiler, and linker settings made in the IDE are ignored. This behavior occurs because Nios II Software Build Tools projects are not IDE-managed projects. In addition, the make-related preferences do not pertain to imported Nios II Software Build Tools projects. The IDE ignores these options during the build process.

Workaround

Make these settings in the project's makefile.

Solution Status

Not fixed

URL Project Location Causes Project Creation Error

When you try to create a new project in an existing workspace, you might see an error dialog box saying:

```
Project cannot be created. Reason: Internal Error
```

This error might occur if the path to any project in the workspace is a URL location, for example **file:/F:/Design**. To view the path, right-click on the project and select **Properties**.

Workaround

Import your existing application and system library projects into a new workspace.

Solution Status

Not fixed

Compilation Error with Separate Exception Stack Option

Choosing the **Use a separate exception stack option** might cause the compilation error `"UNDEFINED VARIABLE %STACK_POINTER%"` when building a project. This error occurs if the exception stack is larger than the memory available for it.

Workaround

On the system library properties page for the project, turn off the separate exception stack or reduce the **Maximum exception stack size** setting.

Solution Status

Not fixed

Incorrect Stack and Heap Free Space Report

The makefile reports an incorrect number of bytes free for the stack and heap, if the heap and stack are in different memory regions.

Workaround

No workaround available.

Solution Status

Not fixed

Nios II IDE Reports Problems Without Displaying Error in Console

When building a project, the Nios II IDE reports problems, but the build output in the console does not contain any errors.

The Nios II IDE incorrectly reports some linker warnings as errors, with a dialog box saying **Errors exist in a required project**. The Dhrystone software example exhibits this behavior, and recompiling the project again makes the issue go away.

Workaround

If the Console output does not contain errors, then the project built correctly. On subsequent builds, the linker step is skipped and the errors do not appear.

Solution Status

Not fixed

Nios II IDE Errata: Debugging Projects

This section describes in detail the issues related to debugging projects in the Nios II IDE.

Method for Accessing MMU and MPU Registers in the Debugger

You might observe an error if you try to read or write an MMU or MPU register via the Nios II Debugger.

Workaround

To read an MPU region, execute the following steps:

1. Set region `INDEX` in the `MPUBASE` register.
2. Exit and reenter debug mode, that is, single step.

3. Set the RD bit in the MPUACC register.
4. Exit and reenter debug mode, that is, single step.
5. Read back MPUBASE for pertinent information.
6. Read back MPUACC for pertinent information.

For a system with an MMU or MPU, this workaround allows you to read and write the current values of the registers. However, you cannot use it to control MPU regions or MMU TLB entries.

The debug core copies the Nios II processor’s register values to its internal memory when the processor enters debug mode. The debug core writes register values back to the processor only when the processor leaves debug mode. Therefore, if you attempt to set an MPU region with several consecutive values, only the last one, when leaving debug mode, is committed to the processor.

Solution Status

Not fixed

IDE Cannot Display Imported Profiling Data

If you create a software project in the Nios II Software Build Tools command line flow, generate profiling data to **gmon.out** in the Nios II command shell, and then import the project into the IDE, you cannot use the Profiling perspective to view the profiling data. When you attempt to view **gmon.out**, the IDE displays the error message **nios2-elf-gprof: ../: not in a.out format**.

Workaround

Run `nios2-elf-gprof` from the Nios II command shell to generate a profiler report.

Solution Status

Not fixed

Error Message when Downloading .elf File

If the **Build Automatically** option is turned on in the IDE, you might get the following error when downloading an executable and linking formal file (**.elf**):

```
"2 [main] sh 5736 fork: child 3892 - died waiting; \
    for longjmp before initialization, errno 11"
```

The **Build Automatically** option is off by default, as recommended by the Eclipse help system.

Workaround

Turn off the **Build Automatically** option and download the **.elf** file again.

Solution Status

Not fixed

Variable Casting Unsupported in ISS

Casting variables in the Variables view in the Debug perspective when using the Nios II ISS might cause an exception dialog box to open.

This exception might occur if you select **Debug as ISS** in the Nios II IDE and try to cast variables via the Variables view.

Workaround

There is no workaround.

Solution Status

Not fixed

Trace Debug Does Not Support Instruction-Related Exceptions

The instruction-related exception handler is not supported by the Nios II trace tools.

Workaround

There is no workaround.

Solution Status

Not fixed

Trace Debug Does Not Support the JMPI Instruction

The JMPI instruction is not supported by the Nios II trace tools.

Workaround

There is no workaround.

Solution Status

Not fixed

Cannot Locate Source Code in Driver Files Shared by Multiple Projects

If you hit a breakpoint in a driver file, and that driver file is shared with another project that is closed, the Nios II IDE might indicate that it cannot locate the source code.

Workaround

Open the closed system library project and resume debugging.

Solution Status

Not fixed

Console Window Is Not Updated After ISS Error

After performing a **Run as ISS**, if you receive an ISS error in the console window, the console is not updated subsequently.

Workaround

Close the console window after receiving an ISS error. A new console window opens when a new message is available.

Solution Status

Not fixed

ISS Fails on Designs Containing Triple Speed Ethernet MAC or SG-DMA Components

You receive an Internal Error when attempting to perform an ISS simulation of designs containing the Altera Triple Speed Ethernet media access control (MAC) or SG-DMA components because the Nios II ISS does not support these components.

Workaround

Remove the Triple Speed Ethernet MAC and SG-DMA components from your system and perform ISS simulation on the simplified system. You can also simulate the design in ModelSim or test it on hardware.

☞ Before removing the Triple Speed Ethernet MAC and SG-DMA components, make a copy of the unmodified system to ensure that you can return to the original configuration.

Solution Status

Not fixed

Memory Window Sets Control Register Values Incorrectly

The memory window might incorrectly set values in memory-mapped control registers. For example, writing 0x1234 to a byte addressed register results in the value 0x3434 being stored in the register. The memory window shows this incorrect value.

Workaround

Use the GNU debugger (GDB) console window in the IDE, instead of the memory window, to write to the registers. For example, type the following command:

```
set {int} <register address>=0x1234↵
```

☞ You must refresh the memory window for it to correctly display the target value.

Solution Status

Not fixed

Programs That Interact With a Terminal Console on Windows Do Not Work

Programs with this behavior work in v. 6.0 and earlier, but do not work in Nios II IDE v. 6.1 and later.

The Eclipse platform in v. 6.1 and later of the IDE (on Windows only) sends the string `\r\n` instead of just `\n` when running or debugging using the Terminal. This behavior can break existing software designs, and it is inconsistent with `nios2-terminal`, which still just sends `\n`.

Workaround

Change the software running on the Nios II processor to parse for `\r\n` as well as `\n`.

Solution Status

Not fixed

"Run as ModelSim" in the Nios II IDE Fails

The **Run as ModelSim** command might fail on launch configurations created in v. 7.0 or earlier of the IDE. This problem does not occur for new launch configurations.

Workaround

Select a location for the ModelSim tool from the launch configuration dialog box. You can use the **Browse** button next to the **ModelSim path** group, or type in a path to the ModelSim directory (for example **c:/altera/71/modelsim_ae/win32aloem**).

Solution Status

Not fixed

The Restart Command on the Run Menu Does Not Work

The **Restart** command on the **Run** menu does not work.

Workaround

Stop the program, then debug it again. If the debugger is hung in an endless loop, use the following **bash** alias to break the target, then stop it:

```
alias break="kill -2 \'ps -a | grep nios2-elf-gdb | cut -f6 -d' '\'"
```

Solution Status

Not fixed

Watchpoints Do Not Work on Certain Variables

Watchpoints do not work on a variable whose size is not 32 bits.

Workaround

Change the types of global and static local variables to `int`, `long`, or `unsigned long` before setting watchpoints on them.

Solution Status

Not fixed

Nios II IDE Errata: Navigating Projects

This section describes in detail the issues related to navigating projects in the Nios II IDE.

Nios II IDE Freezes While Displaying the Splash Screen

After clicking **Switch Workspace** on the File menu on a Windows machine, a Nios II IDE splash screen appears. Unfortunately, this splash screen obscures the dialog box that asks you to specify the new workspace. As a result, the IDE appears to freeze.

Workaround

Press Alt-Tab to switch applications. Two relevant application icons appear: an Eclipse icon associated with the splash screen and a Nios II IDE icon associated with the **Workspace** dialog box. Select the Nios II icon to bring the dialog box to the foreground.

Solution Status

Not fixed

Internal Error When Double-Clicking on a Large Objdump File

On Windows when opening a large **objdump** file in the Nios II IDE, you might get the following error message:

```
Unable to create this part due to an internal error.
Reason for the failure: Editor could not be initialized.
```

Workaround

Adjust the Windows launch arguments for the Nios II IDE editor. Perform the following steps:

1. On the Windows Start menu, browse to the **Nios II EDS** program icon, right-click it, then click **Properties**. The **Windows Properties** dialog box appears.
2. In the **Target** field, append `"vmargs -Xmx1024m"` to the end of the path to the Nios II IDE executable. For example:

   ```
   C:\altera\72\nios2eds\bin\eclipse\nios2-ide.exe -vmargs -Xmx1024m
   ```

Solution Status

Not fixed

C/C++ Scanner Does Not Support Certain C/C++ Constructs

The C/C++ scanner performs C/C++ Search, navigation, open declaration, and parts of content assist. Due to limitations of the C/C++ scanner, these features do not work with C code constructs not supported by the C++ language. An example is functions that take a function pointer as an argument.

Workaround

If the C/C++ Search fails, use the File Search facility.

Solution Status

Not fixed

Nios II Software Build Tools Errata

This section describes in detail the issues in the Nios II Software Build Tools.

Build Errors on Interrupt API Calls

You might observe build errors if your application project installs an interrupt service routine (ISR) that calls the `alt_irq_register()` function.

The Nios II EDS v9.1 includes support for an optional External Interrupt Controller (EIC) interface on the Nios II processor. To support the EIC interface, the HAL includes an enhanced interrupt API.

When you create a BSP, the Software Build Tools determine which interrupt API(s) the device drivers in your system support. If all registered drivers in the BSP identify themselves as supporting the enhanced API, the Software Build Tools implement the enhanced API. All Altera device drivers in v9.1 and later support the enhanced API.

When the enhanced API is implemented, the legacy interrupt API, including `alt_irq_register()`, is not available. Therefore, if the application code contains a call to a legacy API function, there will be a linker error.

This issue might affect application projects that call `alt_irq_register()` or other legacy interrupt API functions directly, rather than depending on drivers in the BSP for interrupt support.

If your Nios II application project's source code registers an ISR using `alt_irq_register()`, or includes an ISR, your application might fail to compile in the Nios II EDS 9.1, due to undefined references to `alt_irq_register()`, or to a change to the ISR function prototype.

Workaround

Modify your application code to use the new enhanced interrupt API. To use the enhanced API, you must modify several function calls and all ISR function prototypes.

For information about supporting the enhanced interrupt API, refer to the *Exception Handling* chapter of the *Nios II Software Developer's Handbook*, or to *AN595: Vectored Interrupt Controller Usage and Applications*.

You must upgrade to the enhanced API if you wish to use an EIC in your hardware design to accelerate interrupt response time. The enhanced API also works with the Nios II processor's internal interrupt controller.

Solution Status

Not fixed

BSP Editor Incorrectly Allows Mixed Drivers

If multiple drivers are found for a single peripheral type, and you have multiple instances of the peripheral type in your SOPC Builder system, the Nios II BSP editor allows you to manually select a different driver for each instance of the peripheral or processor. For example, if you have two `altera_avalon_timer` devices, you can select two different timer drivers.

This behavior is not fully supported. You must choose the same driver for both instances of the peripheral type.

You can make driver assignments in one of the following ways:

- In the Nios II BSP editor, within the **Drivers** tab. The **Driver name** associated with each peripheral can be edited, if multiple driver types are found that support the peripheral. You must make the same driver type selection for each instance of a peripheral type in the list of peripherals as defined in the **Module Name** column.
- With the Nios II Software Build Tools in command-line mode (`nios2-bsp`, `nios2-bsp-create-settings`, `nios2-bsp-update-settings`): the **set_driver** command can be invoked (either from a Tcl script or with the `--cmd` option). The **set_driver** command accepts a peripheral module instance name and driver name as arguments. If you are using the command line development flow and wish to use this command, you must invoke it once for each instance of a peripheral type in your system, similar to the BSP editor flow described above.

Workaround

There is no workaround.

Solution Status

Fixed in v. 9.1

nios2-bsp --help Displays Incorrect List of BSP Types

The Nios II Software Build Tools support custom HAL-based operating systems, when they are added as Tcl-scripted software packages. However, the `nios2-bsp` tool help message indicates that there are only two BSP types, HAL and UCOSII. If other HAL-based operating system components are added they do not appear in the `nios2-bsp` help message.

Workaround

See "nios2-bsp-create-settings Fails to List BSP Types" to get a correct, dynamic list of available BSP types.

Solution Status

Fixed in v. 9.1

nios2-bsp-create-settings Fails to List BSP Types

The help message for `nios2-bsp-create-settings` incorrectly states that using the `--type` option with no value creates a list of available BSP types.

Workaround

To get a list of available BSP types, execute the following command:

```
nios2-bsp-create-settings --type x --settings x --sopc=<existing sopcinfo path>↵
```

The following command returns an error message that lists available BSP types:

```
SEVERE: Available BSP type values for the --type argument are: [<list of BSP types>]
```

Solution Status

Fixed in v. 8.1

Makefile Error with Absolute Source Path

You might receive the following error message from the Nios II Software build tools:

```
Makefile:: *** target pattern contains no '%'
```

This error message can appear if you are using Windows, you specify an absolute source path with the `--src-dir` option, and your application directory, specified with the `--app-dir` option, is the same as your source path.

Workaround

Use a relative path to the source directory, or organize your files so that your application directory is different from your source directory.

Solution Status

Fixed in v. 8.1

Incorrect Linker Errors

While building a HAL-based Nios II software project, the linker displays misleading error messages if the SOPC Builder system provides less than 32 bytes of memory at the reset vector. The linker errors are similar to the following:

```
/cygdrive/d/altera/80/nios2eds/bin/nios2-gnutools/H-i686-pc-cygwin/bin/../lib/gcc/ \
    nios2-elf/3.4.6/../../../../nios2-elf/bin/ld:/cygdrive/d/test3/software/ \
    hello_world_0_syslib/Debug/system_description/generated.x:186: warning: memory \
    region UNDEFINED not declared
/cygdrive/d/altera/80/nios2eds/bin/nios2-gnutools/H-i686-pc-cygwin/bin/../lib/gcc/ \
    nios2-elf/3.4.6/../../../../nios2-elf/bin/ld:/cygdrive/d/test3/software/ \
    hello_world_0_syslib/Debug/system_description/generated.x:186: parse error
collect2: ld returned 1 exit status
make: *** [hello_world_0.elf] Error 1
```

Workaround

In the SOPC Builder system, if the reset and exception vectors are in the same memory device, ensure that the exception vector is at least 32 bytes above the reset vector.

Solution Status

Fixed in v. 8.1

Java Heap Space Error When Creating BSP

When creating a BSP, the Nios II Software Build Tools might hang with the following message:

`SEVERE: OutOfMemoryError: Java Heap Space`.

This problem occurs when your SOPC Builder design has connection cycles. A connection cycle consists of a group of components that master one another in a circular arrangement. For example, if `ddr_sdram_bridge` masters `lcd_sgdma`, and `lcd_sgdma` masters `ddr_sdram_bridge`, the two components form a simple connection cycle. In this situation, the Software Build Tools are unable to create a BSP.

Workaround

Avoid connection cycles in the SOPC Builder system.

Solution Status

Fixed in v. 9.0

Software Build Tools Fail if Nios II EDS is Installed in a Path Containing Spaces

The Nios II Command Line Software Build Tools fail if you install the Nios II EDS in a path containing spaces.

Workaround

Reinstall the Nios II EDS into a path that does not contain spaces.

Solution Status

Not fixed

User-Managed BSP Settings Not Supported in Nios II IDE

This issue is described on page 32.

Upper-Case File Extensions Not Supported

The Nios II Software Build Tools for applications and libraries (`nios2-app-generate-makefile` and `nios2-lib-generate-makefile` commands) do not support source files with certain upper-case extensions. If a file with an upper-case extension is included, the make command stops with no descriptive warning.

Only Nios II assembly language files built by the C preprocessor can have upper-case file extension (**.S**). All C language files must have the extension **.c** or **.h**. C++ language source files must have the extension **.cpp**, **.cxx**, **.cc,** or **.h**.

Workaround

Rename all C language files with the extension **.c** or **.h**. Rename all C++ language files with the extension **.cpp**, **.cxx**, **.cc,** or **.h**.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Peripheral Errata

This section describes in detail the Nios II EDS issues related to peripheral components.

Software Build Tools Fail With UART Driver

If you use the Software Build Tools on a design containing the SOPC Builder UART peripheral, and if driver `ioctl()` support for the `altera_avalon_uart` component is enabled, then the BSP build process might fail. The **system.h** generated by the Software Build Tools contains an invalid type for the UART parameter.

Workaround

To correct **system.h**, carry out the following steps:

1. Manually edit the **system.h** file by changing the following line:

   ```
   #define UART1_PARITY "N"
   ```

   to:

   ```
   #define UART1_PARITY 'N'
   ```

   ☞ Change double quotes to single quotes.

2. Build the BSP.

Solution Status

Fixed in v. 8.1

Unaligned Transfers of Small Payloads Fail on SG-DMA

The Scatter Gather DMA SOPC Builder peripheral does not correctly handle unaligned transfers with small payloads. A payload length smaller than the data width causes erroneous data transfers.

Workaround

Avoid using DMA devices to transfer small payloads.

If absolutely necessary, for a 32-bit SGDMA, a minimum length of 4 bytes guarantees that data is transferred correctly.

Solution Status

Not fixed

DMA Controller Always Busy in Burst Mode

The DMA controller component (`altera_avalon_dma`), when enabled for burst transactions, does not perform transfers at widths other than its full data width. The DMA controller is always busy.

Workaround

When bursting is enabled, the DMA controller must be programmed to perform transactions at its full data width.

Solution Status

Not fixed

Non-System-Wide Reset Can Cause Improper Initialization of Mailbox Core

The `altera_avalon_mailbox` peripheral might not be initialized properly when a soft (non-system-wide) reset occurs. In this condition, mailbox contents (read and write pointers) are not reinitialized and might show potentially stale data.

Workaround

Ensure that a system-wide reset event occurs by asserting the `reset_n` input to the SOPC Builder system containing the mailbox. This resets all peripherals and Nios II processors in the system.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Target Software Errata

This section describes in detail the Nios II EDS issues related to target software packages.

NicheStack tcp_wake() Function Does Not Awaken All select() Sleepers

This issue impacts software based on the NicheStack TCP/IP Stack - Nios II Edition, with multiple sleeping tasks that depend on the same `select()` event to wake up. The `tcp_sleep()` and `tcp_wake()` functions do not work properly for tasks that call `select()`.

Tasks can normally use `tcp_sleep()` and `tcp_wake()` to suspend and then awaken themselves upon specific events. The `select()` function passes an event pointer to `tcp_sleep()` that is identical for every task that might call `select()`. All other events that call `tcp_sleep()` pass it a unique event pointer that specifically identifies them as the task to be awakened at the next event.

The `tcp_wake()` function apparently assumes that the list of sleepers contains unique event pointers by which they can be identified. Therefore, when a `select()` event occurs, instead of awakening all tasks waiting for it, `tcp_wake()` awakens only the first task in the list. Therefore, when more than one task calls `select()` to await some outcome on a socket, only the highest-priority task in the sleepers list is awakened at `select()` events. Other sleeping tasks that depend on the same `select()` event do not wake up.

Workaround

There is no workaround.

Solution Status

This issue will be fixed in future release of the Nios II EDS.

EIC-Based Nios II System Crashes With NicheStack

When you attempt to run software based on the NicheStack TCP/IP Stack - Nios II Edition on a hardware design incorporating an external interrupt controller such as the VIC, the Nios II system crashes during initialization, with unpredictable results.

Workaround

There is no workaround.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Gigabit Ethernet Performance Issues

The NicheStack TCP/IP Stack - Nios II Edition might hang while sending transmission control protocol (TCP) packets to the host, or experience dropped packets when receiving TCP packets from the host. These issues can arise with version 3.1 of the NicheStack TCP/IP Stack, using the triple-speed Ethernet component to transmit over a Gigabit Ethernet point-to-point connection.

Workaround

Download and install the patches at www.altera.com/support/kdb/solutions/rd01132009_588.html.

Alternatively, upgrade to the Nios II EDS v. 9.0 or later.

Solution Status

Fixed in v. 9.0

stdio Does Not Work with MicroC/OS-II and Small C Library

`stdin`, `stdout`, and `stderr` do not work in MicroC/OS-II applications built with the Small C library option.

Workaround

Disable the small C library option.

Solution Status

Not fixed

cout From MicroC/OS-II Task Does Not Send Data to stdout

If neither `printf()` or `cout` is used from `main()` before tasks are started, `cout` does not work from a task.

Workaround

Add the following C++ code to the beginning of `main()`:

```
std::ios_base::sync_with_stdio(false);
```

Solution Status

Not fixed

malloc(), realloc() Failures With MicroC/OS-II

When you use the MicroC/OS-II RTOS, calls to `malloc()` and `realloc()` might fail if successive calls to `malloc()` or `realloc()` within a MicroC/OS-II task occur after changing the task priority of the task in which a memory block is originally allocated.

Workaround

Use one of the following workarounds:

- Allocate and/or reallocate memory blocks outside of MicroC/OS-II tasks, before task switching starts. Preallocated memory blocks make it possible to change thread priorities at runtime.
- Allocate fixed areas of memory using arrays (rather than using `malloc()`) before task switching starts. Fixed memory arrays make it possible to change thread priorities at runtime.
- Allocate memory using `malloc()` or `realloc()` from a MicroC/OS-II task. You can change task priorities at runtime, but only for tasks that have not used `malloc()` or `realloc()`.

Solution Status

Not fixed

Toolchain Errata

This section describes in detail the Nios II EDS issues related to the GNU compiler toolchain (gcc, gdb, etc.).

sof2flash Names Output Files Incorrectly

If **sof2flash** is called with the `--save` option and a relative path is specified for the output file (that is, including "./" or "../"), the raw binary file that is saved is incorrectly named **.rbf** instead of **<flash filename>.rbf**.

Workaround

Provide a full path to the output file(s), or change to the output directory before running the **sof2flash** command so the output file specification is `--output=<`*`filename`*`>.flash`.

Solution Status

Fixed in v. 8.1

GNU Assembler Does Not Accept the --defsym Flag

According the GNU documentation, you can set an assembler definition by using the `--defsym` flag, but it does not work in the following form: `--defsym MY_VAR=1`.

Workaround

There is no workaround.

Solution Status

Not fixed

How to Contact Altera

For the most up-to-date information about Altera products, refer to Table 8.

Table 8. Contact Information

Contact *(1)*	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Altera literature services	Email	literature@altera.com
Non-technical support (General)	Email	nacomp@altera.com
(Software Licensing)	Email	authorization@altera.com

Note:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

Table 9 shows the typographic conventions that this document uses.

Table 9. Typographic Conventions (Part 1 of 2)

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Indicates command names, dialog box titles, dialog box options, and other graphical user interface (GUI) labels. For example, **Save As** dialog box. For GUI elements, capitalization matches the GUI.
bold type	Indicates directory names, project names, disk drive names, file names, file name extensions, dialog box options, software utility names, and other GUI labels. For example, **\qdesigns** directory, **d:** drive, and **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Indicates document titles. For example, *AN 519: Stratix IV Design Guidelines.*
Italic type	Indicates variables. For example, *n* + 1. Variable names are enclosed in angle brackets (< >). For example, <*file name*> and <*project name*>**.pof** file.
Initial Capital Letters	Indicates keyboard keys and menu names. For example, Delete key and the Options menu.
“Subheading Title”	Quotation marks indicate references to sections within a document and titles of Quartus II Help topics. For example, “Typographic Conventions.”

Table 9. Typographic Conventions (Part 2 of 2)

Visual Cue	Meaning
`Courier type`	Indicates signal, port, register, bit, block, and primitive names. For example, `data1`, `tdi`, and `input`. Active-low signals are denoted by suffix `n`. For example, `resetn`. Indicates command line commands and anything that must be typed exactly as it appears. For example, `c:\qdesigns\tutorial\chiptrip.gdf`. Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword `SUBDESIGN`), and logic function names (for example, `TRI`).
1., 2., 3., and a., b., c., and so on.	Numbered steps indicate a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ■	Bullets indicate a list of items when the sequence of the items is not important.
☞	The hand points to information that requires special attention.
⚠ CAUTION	A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.
⚡ WARNING	A warning calls attention to a condition or possible situation that can cause you injury.
↵	The angled arrow instructs you to press Enter.
👣	The feet direct you to more information about a particular topic.

101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support
www.altera.com/support

Nios II Embedded Design Suite Release Notes and Errata

© March 2009 RN-EDS-4.0

About These Release Notes

These release notes cover versions 8.0 through 9.0 of the Altera® Nios® II Embedded Design Suite (EDS). These release notes describe the revision history and errata for the Nios II EDS.

Errata are functional defects or errors, which might cause the product to deviate from published specifications. Documentation issues include errors, unclear descriptions, or omissions from current published specifications or product documents.

For the most up-to-date errata for this release, refer to the latest version of the *Nios II Embedded Design Suite Release Notes and Errata* on the Altera website. You can refer to release notes and errata for the Nios II processor core in the *Nios II Processor* chapter of the *MegaCore IP Library Release Notes and Errata*.

Prior to v7.2, Nios II release notes and errata were organized differently. You can refer to the documents from before v7.2 on the Altera website at:

- Nios II Embedded Design Suite Release Notes
- Nios II Embedded Design Suite Errata

Revision History

Table 1 shows the revision history for the Nios II EDS, including the Nios II C2H Compiler.

For more information on new features and changes, refer to the *Nios II Processor Reference Handbook*, the *Nios II Software Developer's Handbook*, and the *Nios II C2H Compiler User Guide*. For information about changes to the Nios II processor core, refer to the *Nios II Processor* chapter of the *MegaCore IP Library Release Notes and Errata*.

Revision Details

This section provides details about some of the Nios II EDS revisions shown in Table 1.

Table 1. Nios II Embedded Design Suite Revision History

Version	Date	Description
9.0.0	March 2009	Added the `nios2-convert-ide2sbt` tool, which converts projects from the Nios II integrated development environment (IDE) development flow to the Nios II software build tools development flow
8.1.1	January 2009	Nios Community Wiki URL changing (www.nioswiki.com)
8.1.0	November 2008	Maintenance release
8.0	May 2008	■ The Nios II BSP editor, a standalone graphical user interface (GUI), provides an easy-to-use front end for the Nios II software build tools. ■ The Altera Hardware Abstraction Layer (HAL) supports an advanced exception handler that can handle Nios II memory protection unit (MPU) exceptions. ■ The Nios II Studio is available as a Beta release. The Studio is an alternative to the Nios II IDE. Nios II projects created with the Nios II Studio are 100% compatible with projects created with the Nios II software build tools on the command line. ■ The Nios II software build tools `--sopc` option only accepts **.sopcinfo** files as arguments and not **.sopc** files. ■ The Nios II IDE command-line utilities, which form the infrastructure of the Nios II IDE, are deprecated; support will be discontinued in a future release. The deprecated tools include `nios2-create-system-library`, `nios2-create-application-project`, `nios2-build-project`, `nios2-import-project`, and `nios2-delete-project`. ■ The Lightweight IP (lwIP) transmission control protocol/Internet protocol (TCP/IP) suite is removed from the Nios II EDS. ■ The MicroC/OS-II real-time operating system (RTOS) is upgraded from v2.83 to v2.86. ■ The NicheStack TCP/IP stack — Nios II Edition is upgraded from v3.0 to v3.1. ■ Certain Nios II software examples are removed from the EDS, and are distributed on the Altera website instead. The affected designs and their new locations are listed in "Software Examples Moved to the Web" on page 2. ■ The NicheStack TCP/IP Stack macro definition `INICHE_DEFAULT_IF` is deprecated. For details, see "Deprecated NicheStack Setting". ■ The **nios_csh** file in **sopc_builder/bin** is removed from the Nios II EDS.

Software Examples Moved to the Web

Starting in v8.0, the software examples shown in Table 2 are removed from the Nios II EDS, and instead distributed on the Web, at the locations shown in the table. This applies to software examples in Nios II IDE template form, and in the form of Nios II software build tools `create-this-app` and `create-this-bsp` scripts.

Deprecated NicheStack Setting

The Nios II IDE and the Nios II software build tools contain a setting to specify a default Ethernet (MAC) interface when using the NicheStack TCP/IP stack. Starting in v8.0, the default MAC setting is deprecated. It is no longer used in Altera software application example code. The NicheStack TCP/IP stack has never depended on this setting.

Table 2. Software Examples Removed from the Nios II EDS

Software Example	Delivery Mechanism
Read-Only Zip File System	The ExampleDesigns page of the Nios Community Wiki (www.nioswiki.com)
MicroC/OS-II Message Box	The ExampleDesigns page of the Nios Community Wiki (www.nioswiki.com)
Hello_LED	The ExampleDesigns page of the Nios Community Wiki (www.nioswiki.com)
Host File System	The ExampleDesigns page of the Nios Community Wiki (www.nioswiki.com)
Dhrystone	■ The Literature: Nios II Processor page, with the *Nios II Performance Benchmark Data Sheet* ■ The Fast Nios II Hardware Design Example page
MicroC/OS II Tutorial	The Literature: Nios II Processor page, with the *Using MicroC/OS-II RTOS with the Nios II Processor Tutorial*
Tightly Coupled Memory (TCM) Tutorial	The Literature: Nios II Processor page, with the *Using Nios II Tightly Coupled Memory Tutorial*
Custom Instruction Tutorial	The Literature: Nios II Processor page, with the *Nios II Custom Instruction User Guide*

The default MAC setting generates the macro `INICHE_DEFAULT_IF` in the generated **system.h** file located in the board support package (BSP). If your software depends on this macro definition, modify it, for example, to define `INICHE_DEFAULT_IF` explicitly. The macro will be removed in a future Nios II EDS release.

Altera networking example applications do not support multiple Ethernet interfaces. If you wish to use the Web Server or Simple Socket Server networking examples in a hardware system with multiple Ethernet MAC interfaces, or with the multi-port Altera Triple Speed Ethernet MegaCore function, you need to modify the `get_ip_addr()` and `get_mac_addr()` functions to provide a unique MAC and IP address for each Ethernet interface in the system.

Errata Summary

Table 3 summarizes the issues that affect the Nios II Embedded Design Suite versions 8.0 through 9.0. For a detailed description of each issue, click on the issue name.

Table 3. Nios II EDS Errata (Part 1 of 5)

Added or Updated	Issue	Page	Affected Version		
			9.0	8.1	8.0
	C2H Compiler				
Nov 2008	Error "java.lang.NoClassDefFoundError" During C2H Compilation	7	✓	✓	—
Jul 2008	C2H Cache Coherency Issue with Global Variables	8	✓	✓	✓
Jul 2008	Functions Declared Without a Return Type Are Not Supported	9	✓	✓	✓
Oct 2007	Pre-7.1 Systems Are Not Supported	9	✓	✓	✓
Oct 2007	--src-dir Software Build Tools Argument Does Not Work With C2H	9	✓	✓	✓
May 2007	Accelerator Generation Failure If Tools Are Installed in Path With Spaces	9	✓	✓	✓
May 2007	C2H Accelerators Time Out Immediately When Untethered	*(1)*	✓	✓	✓
May 2007	The C2H Compiler Regenerates an Accelerator Unnecessarily	10	✓	✓	✓
May 2007	Error: c2h_accelerator_base_addresses.h: No such file or directory	10	✓	✓	✓

Table 3. Nios II EDS Errata (Part 2 of 5)

Added or Updated	Issue	Page	Affected Version		
			9.0	**8.1**	**8.0**
May 2007	Java Heap Space Exception if Quartus II Compilation is Enabled	10	✓	✓	✓
May 2007	Pointer Dereferences to Volatile Types	11	✓	✓	✓
Dec 2006	C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File	11	✓	✓	✓
Dec 2006	C2H Compiler Out-of-Memory Error	*(1)*	✓	✓	✓
Jun 2006	Array Elements in Structures Do Not Copy Correctly	11	✓	✓	✓
Jun 2006	Clean Build Causes Build Failure	12	✓	✓	✓
Jun 2006	Changing Build Configurations Produces Unexpected Results	12	✓	✓	✓
Jun 2006	Closed System Library While Working With the C2H Compiler	*(1)*	✓	✓	✓
Jun 2006	Hardware Accelerators Remain After Deleting the Software Project *(2)*	13	✓	✓	✓
Jun 2006	Incorrect Results From Logical or Conditional Operation With Side-Effects	13	✓	✓	✓
Jun 2006	Launch SOPC Builder Button in C2H View	14	✓	✓	✓
Jun 2006	Multiple Clock Domains Cause Hardware Accelerator to Fail	*(1)*	✓	✓	✓
Jun 2006	Simulating on ISS Not Supported	*(1)*	✓	✓	✓
	Development Boards				
Jun 2006	Intermittent Failures While Accessing CompactFlash Card	14	✓	✓	✓
	Documentation Issues				
Mar 2009	Compiler Flags for Building Custom Newlib	15	✓	✓	✓
Jan 2009	Obsolete Wiki URLs in Help Systems and Handbook	16	Fixed	✓	—
Nov 2008	Error Message on Invalid Exception or Reset Vector	16	✓	✓	—
Jul 2008	Missing Source Control Information	17	✓	✓	✓
Dec 2007	Incomplete newlib Documentation	*(1)*	✓	✓	✓
Oct 2007	Nios II IDE Online Help Expand Buttons Do Not Work	18	✓	✓	✓
	Download Cables and Debug Hardware				
Jun 2006	Communication Errors During Run/Debug Sessions Using Older Download Cables	*(1)*	✓	✓	✓
	Hardware Example Designs				
Mar 2007	Incorrect PLL Phase Shift in Stratix II and Cyclone II Designs	19	✓	✓	✓
	Software Examples				
Nov 2008	Simple Socket Server Example Displays Incorrect IP Address	20	✓	✓	—
Oct 2007	Hardware Tutorial Software Example Hangs on Some Boards	20	✓	✓	✓
Jun 2006	Networking Examples	21	✓	✓	✓
Jun 2006	RAM Test Failure When Running Memory Test Software Template on the ISS	*(1)*	✓	✓	✓
	Flash Programmer				
May 2007	Delay When Creating New Flash Programmer Configuration	21	✓	✓	✓
Dec 2006	elf2flash File Size Limit	21	✓	✓	✓
Dec 2006	Error Message “No such file or directory” When Programming Flash	22	✓	✓	✓

Table 3. Nios II EDS Errata (Part 3 of 5)

Added or Updated	Issue	Page	Affected Version		
			9.0	8.1	8.0
Dec 2006	Flash Programmer Error with Multiprocessor System	*(1)*	✓	✓	✓
Hardware Simulation					
Dec 2006	Error “UNC paths are not supported” Launching ModelSim	22	✓	✓	✓
Jun 2006	ModelSim Fails to Load Large Memory Models	*(1)*	✓	✓	✓
Jun 2006	Uninitialized .bss Variables in Simulation	22	✓	✓	✓
Host Platform					
Dec 2007	Windows/Cygwin: Nios II Processor Generation Failure	23	✓	✓	✓
Oct 2007	Windows Vista: Limited Support in Nios II EDS	23	✓	✓	✓
Jun 2006	Linux: Process Fails To Terminate While Debugging With the Nios II ISS Target	24	✓	✓	✓
Jun 2006	Linux: The Quartus® II Stand-Alone Programmer is Not Supported	*(1)*	✓	✓	✓
Jun 2006	Windows: Frisk Antivirus Software Causes SOPC Builder and Nios II Command Shell to Be Unresponsive	*(1)*	✓	✓	✓
Nios II IDE					
Building Projects					
Mar 2009	Nios II IDE Command-Line Tools Select Wrong Workspace on Linux	24	✓	✓	✓
Mar 2009	Nios II IDE Command-Line Tools Hang on Windows	25	✓	✓	✓
Nov 2008	Nios II IDE Cannot Find stdio.h in Outline View	25	✓	✓	✓
Jul 2008	Nios II IDE Hangs With UNC Project Path	26	✓	✓	✓
Jul 2008	Build Failure with Nios II Advanced Exceptions, MMU, and MPU	26	✓	✓	✓
May 2007	Build Command Not Functional for BSPs Created Using the Software Build Flow	26	✓	✓	✓
May 2007	Linker Errors with Dual-Port Memories	27	✓	✓	✓
May 2007	User-Managed BSP Settings Not Supported in Nios II IDE	27	✓	✓	✓
May 2007	URL Project Location Causes Project Creation Error	27	✓	✓	✓
Dec 2006	Compilation Error with Separate Exception Stack Option	28	✓	✓	✓
Dec 2006	Illegal Project Location Error When Creating New Project	*(1)*	✓	✓	✓
Dec 2006	Incorrect Address Assignment for Dual-Port Memory Mastered by Two Different Master Ports	*(1)*	✓	✓	✓
Dec 2006	Incorrect Stack and Heap Free Space Report	28	✓	✓	✓
Dec 2006	Nios II IDE Reports Problems Without Displaying Error in Console	28	✓	✓	✓
Jun 2006	Build Errors After Changing Component Names in SOPC Builder	*(1)*	✓	✓	✓
Jun 2006	Nios II IDE Unnecessarily Updates the SOPC Builder System File (.ptf)	*(1)*	✓	✓	✓
Debugging Projects					
Nov 2008	IDE Cannot Display Imported Profiling Data	29	✓	✓	—
Jul 2008	Error Message when Downloading .elf File	29	✓	✓	✓
Jul 2008	Variable Casting Unsupported in ISS	29	✓	✓	✓
Jul 2008	Trace Debug Does Not Support Instruction-Related Exceptions	30	✓	✓	✓
Jul 2008	Error Accessing MMU and MPU Registers in the Debugger	30	✓	✓	✓
Oct 2007	Trace Debug Does Not Support the JMPI Instruction	30	✓	✓	✓

Table 3. Nios II EDS Errata (Part 4 of 5)

Added or Updated	Issue	Page	Affected Version		
			9.0	8.1	8.0
May 2007	Cannot Locate Source Code in Driver Files Shared by Multiple Projects	31	✓	✓	✓
May 2007	Console Window Is Not Updated After ISS Error	31	✓	✓	✓
May 2007	ISS Fails on Designs Containing TSE MAC or SG-DMA Components	31	✓	✓	✓
May 2007	Memory Window Sets Control Register Values Incorrectly	31	✓	✓	✓
May 2007	Previous 7.1 Beta Install Might Cause Internal Error When Launching FS2 in IDE	*(1)*	✓	✓	✓
Dec 2006	Programs That Interact With a Terminal Console on Windows Do Not Work	32	✓	✓	✓
Dec 2006	"Run as ModelSim" in the Nios II IDE Fails	32	✓	✓	✓
Dec 2006	The Restart Command on the Run Menu Does Not Work	32	✓	✓	✓
Jun 2006	Breakpoints on Adjacent Lines of Assembly Fail to Halt the Processor	*(1)*	✓	✓	✓
Jun 2006	Cannot Use Watchpoints in the Nios II IDE When the FS2 Console Is Open	*(1)*	✓	✓	✓
Jun 2006	Debugger Cannot Step Into __sflags(), and Continues Execution Instead	*(1)*	✓	✓	✓
Jun 2006	Incorrect Breakpoint Filtering on Threads	*(1)*	✓	✓	✓
Jun 2006	Missing Address and Data in the Trace View	*(1)*	✓	✓	✓
Jun 2006	nios2-gdb-server Fails to Terminate After Setting a Watch Point	*(1)*	✓	✓	✓
Jun 2006	"Step failed. Target is not responding (timed out)" Error Message	*(1)*	✓	✓	✓
Jun 2006	Uninitialized Memory Error When Executing From ISS	*(1)*	✓	✓	✓
Jun 2006	Watchpoints Do Not Work on Certain Variables	33	✓	✓	✓
Navigating Projects					
Dec 2006	Nios II IDE Freezes While Displaying the Splash Screen	33	✓	✓	✓
Dec 2006	Internal Error When Double-Clicking on a Large Objdump File	33	✓	✓	✓
Jun 2006	C/C++ Scanner Does Not Support Certain C/C++ Constructs	34	✓	✓	✓
Nios II Studio					
Mar 2009	Incorrect Default Workspace Name	34	✓	—	—
Nov 2008	Error in nios2-studio Help System	34	✓	✓	—
Jul 2008	No System ID Validation Performed	35	✓	✓	✓
Jul 2008	Spurious Messages When Importing Pre-8.0 Projects	35	✓	✓	✓
Jul 2008	public.mk Error or System Failure After Hardware Modification	35	✓	✓	✓
Nios II Software Build Tools					
Mar 2009	Cannot Include C++ File in Driver or Software Package	36	—	—	✓
Nov 2008	Incorrect Linker Errors	36	✓	✓	✓
Nov 2008	Java Heap Space Error When Creating BSP	37	✓	✓	—
Jul 2008	BSP Editor Incorrectly Allows Mixed Drivers	37	✓	✓	✓
Jul 2008	nios2-bsp --help Displays Incorrect List of BSP Types	38	✓	✓	✓
Jul 2008	nios2-bsp-create-settings Fails to List BSP Types	38	✓	✓	✓
Jul 2008	Error Creating BSP for Pre-8.0 System	38	✓	✓	✓
Jul 2008	Makefile Error with Absolute Source Path	39	✓	✓	✓
Oct 2007	Software Build Tools Fail if Nios II EDS is Installed in a Path Containing Spaces	39	✓	✓	✓
May 2007	User-Managed BSP Settings Not Supported in Nios II IDE	27	✓	✓	✓

Table 3. Nios II EDS Errata (Part 5 of 5)

Added or Updated	Issue	Page	Affected Version		
			9.0	8.1	8.0
May 2007	Upper-Case File Extensions Not Supported	39	✓	✓	✓
Peripherals					
Jul 2008	DMA Transfers Truncated or Stalled	*(1)*	✓	✓	✓
Jul 2008	Unaligned Transfers of Small Payloads Fail on SG-DMA	40	✓	✓	✓
Jul 2008	IDT71V416 SRAM Component HardCopy Incompatibility	40	✓	✓	✓
Jul 2008	Software Build Tools Fail With UART Driver	40	✓	✓	✓
May 2007	DMA Controller Always Busy in Burst Mode	41	✓	✓	✓
May 2007	Non-System-Wide Reset Can Cause Improper Initialization of Mailbox Core	41	✓	✓	✓
Jun 2006	Using the DEV_CLRn Signal Can Cause JTAG UART Instability After Device-Wide Reset	*(1)*	✓	✓	✓
Target Software					
Mar 2009	Gigabit Ethernet Performance Issues	42	—	✓	✓
Jul 2008	Degraded TCP/IP Performance with NicheStack TCP/IP stack and TSE MAC	*(1)*	✓	✓	✓
Jul 2008	Slow TSE MAC Performance with NicheStack TCP/IP Stack	*(1)*	✓	✓	✓
Jul 2008	Default IP Address Assigned Incorrectly with NicheStack TCP/IP Stack	*(1)*	✓	✓	✓
Dec 2006	Lightweight IP (lwIP) Failure When Using DHCP	*(1)*	✓	✓	✓
Dec 2006	stdio Does Not Work with MicroC/OS-II and Small C Library	42	✓	✓	✓
Jun 2006	cout From MicroC/OS-II Task Does Not Send Data to stdout	42	✓	✓	✓
Jun 2006	Creating New Custom HAL Components in the Nios II IDE	*(1)*	✓	✓	✓
Jun 2006	malloc(), realloc() Failures With MicroC/OS-II	42	✓	✓	✓
Jun 2006	Problems Using HAL Drivers With Toshiba Flash	*(1)*	✓	✓	✓
Toolchain (gcc, gdb, etc.)					
Jul 2008	sof2flash Names Output Files Incorrectly	43	✓	✓	✓
Dec 2006	GNU Assembler Does Not Accept the --defsym Flag	43	✓	✓	✓
Jun 2006	Breakpoints in C++ Constructors Fail to Halt the Processor	*(1)*	✓	✓	✓

Notes to Table 3:

(1) This issue is documented online. To see the solution, click on the name of the issue, or search the Altera Knowledge Database.

(2) For further information, refer to the *Nios II C2H Compiler User Guide*.

C2H Compiler Errata

This section describes in detail the issues related to the Nios II C2H Compiler.

Error "java.lang.NoClassDefFoundError" During C2H Compilation

C2H compilation fails with the following error:

```
java.lang.NoClassDefFoundError: com/altera/c2h/C2Hgenerate
Exception in thread "main" make: *** [c2h_hdl-t] Error 1
```

This error occurs when you invoke C2H from the Nios II IDE on the Windows operating system. This problem does not affect Linux installations, nor C2H compilation from the Nios II command shell.

Workaround

To correct the problem, visit mySupport and request the Nios II EDS 8.1 Patch 0.01 (PC version).

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

C2H Cache Coherency Issue with Global Variables

Cache coherency issues might be encountered when the following conditions are all true:

- A C2H-accelerated function writes to a global or static variable
- The Nios II processor is configured with a data cache
- The referenced global variable is located in cacheable memory

In this situation, the data in the global variable can be corrupted.

This issue can only arise if the accelerated function writes to the global or static variable. If the function merely reads from the variable, cache coherency is maintained correctly.

Workaround

Add a data cache flush operation in the accelerator's wrapper function. The wrapper can be found in **alt_c2h_<*function_name*>.c**. In this file, find the following comment:

```
/* Write 1 to address 0 starts the accelerator */
```

Directly after this comment, add the following function call:

```
alt_dcache_flush_all();
```

This function call ensures that the data cache is flushed immediately before the accelerator is started.

☞ Flushing the cache entails a performance penalty.

Alternatively, allocate your global and static variables in non-cacheable regions of memory. Placing these variables in non-cacheable regions eliminates any need to perform a data cache flush.

For information about allocating variables to non-cacheable regions, refer to the *Cache and Tightly-Coupled Memory* chapter of the *Nios II Software Developer's Handbook*.

Solution Status

Not fixed

Functions Declared Without a Return Type Are Not Supported

The C2H compiler does not support functions without an explicitly declared return type.

Workaround

If you are using the implicit `int` return type, declare the return type explicitly. If your function has no return value, declare it as `void`.

Solution Status

Not fixed

Pre-7.1 Systems Are Not Supported

Starting in v. 7.1, the C2H Compiler does not support SOPC Builder systems created in SOPC Builder 7.0 or earlier (systems based on a **class.ptf** file).

Workaround

Before using the C2H Compiler, you must upgrade your SOPC Builder system. Open the system in SOPC Builder v. 7.1 or later. SOPC Builder prompts you to convert the system, creating a ***_hw.tcl** file compatible with the C2H Compiler.

Solution Status

Not fixed

--src-dir Software Build Tools Argument Does Not Work With C2H

When using the C2H Compiler with Nios II Software Build Tools, an error occurs when using the `nios2-app-generate-makefile` argument `--src-dir` to specify a directory of source files. An example of the error is:

```
multiple definition of 'my_accelerated_function'
```

Workaround

Specify source files individually using the `--src-files` argument.

Solution Status

Not fixed

Accelerator Generation Failure If Tools Are Installed in Path With Spaces

If the path to your installation of the Altera Design Suite contains spaces, the C2H Compiler fails to generate the accelerators.

Workaround

Reinstall the tools to a path containing no spaces.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

The C2H Compiler Regenerates an Accelerator Unnecessarily

The C2H Compiler might regenerate an accelerator even when the accelerated function is unchanged. This problem can result from changes to files included by the C file containing the accelerated function. The C2H Compiler fails to check that the generated hardware description language (HDL) matches the previously generated HDL causing the system to be regenerated.

Workaround

To avoid this issue move the information in the include file specific to the hardware accelerator into a separate include file. This workaround prevents regeneration of the system when the HDL is unchanged.

Solution Status

Not fixed

Error: c2h_accelerator_base_addresses.h: No such file or directory

When a C2H accelerator is compiled for the first time, the following compile-time error can result if the **Analyze all accelerators** option is selected:

```
c2h_accelerator_base_addresses.h: No such file or directory.
```

Workaround

Click **Build software and generate SOPC Builder system** and build once before building with the **Analyze all accelerators** option.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Java Heap Space Exception if Quartus II Compilation is Enabled

If your design contains a C2H accelerator, and you select **Build software, generate SOPC Builder system, and run Quartus II compilation** in the C2H view, you might see the following error during Quartus II compilation:

```
Exception in thread "main" java.lang.OutOfMemoryError: Java
heap space

make: *** [c2h_hdl-t] Error 1
```

Workaround

Select the **Build software and generate SOPC builder system** option in the C2H Compiler settings window, and then manually launch the Quartus II software to compile the design.

Solution Status

Not fixed

Pointer Dereferences to Volatile Types

The C2H Compiler treats pointer dereferences to a volatile type as if they alias all other pointer dereferences. Pointers that are restrict-qualified are treated the same way.

The two loops in Example 1 cannot be scheduled concurrently because the `volatile` qualification overrides the `__restrict__` pragma.

Example 1. Non-Concurrent Loops

```
volatile int * __restrict__ fifo_rd = FIFO_RD_BASE;
volatile int * __restrict__ fifo_wr = FIFO_WR_BASE;
for ()
{
  *fifo_wr = ....;
}
for ()
{
  ... = *fifo_rd;
}
```

Workaround

Divide the function into multiple IRQ-enabled accelerators that are launched concurrently from the processor, and use FIFOs to communicate between them.

Solution Status

Not fixed

C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File

When accelerating a function in a file, the C2H Compiler cannot link subfunctions that are defined in a different file.

Workaround

Include all subfunctions called by the accelerated function within the same source code file.

Solution Status

Not fixed

Array Elements in Structures Do Not Copy Correctly

C2H accelerators do not correctly copy array elements that are elements of structures.

In Example 2, the `a` and `b` elements of the structure copy correctly, but the `buf` element does not. After this assignment, `struct_a` equals `{9, 8, {3, 3, 3, 3}}`.

Example 2. Array Elements of Structs

```
typedef struct my_struct {
  int a;
  int b;
  int buf[BUF_SIZE];
}MY_STRUCT;
MY_STRUCT struct_a = {1, 2, {3, 3, 3, 3}};
MY_STRUCT struct_b = {9, 8, {7, 7, 7, 7}};
struct_a = struct_b;
```

Workaround

Copy the array elements explicitly, as shown in Example 3.

Example 3. Copying Array Elements Explicitly

```
{
  int i=0;
  do
    {
      struct_a.buf[i] = struct_b.buf[i];
      i++;
    } while (i<LENGTH_OF_BUF_ELEMENT)
}
```

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Clean Build Causes Build Failure

Performing a clean build on a Nios II IDE project that contains a hardware accelerator can cause the next build to fail in the IDE, because the clean build erroneously deletes a file required by the C2H Compiler.

Workaround

Do not perform a clean build on projects that use hardware accelerators. If you have already performed a clean build, recompile with option **Build software, generate SOPC Builder system, and run Quartus II compilation** to regenerate the necessary files.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Changing Build Configurations Produces Unexpected Results

The C2H Compiler does not support multiple build configurations (for example Release or Debug) in the Nios II IDE. After creating one or more accelerators in a particular configuration, the C2H Compiler produces undefined results if you switch to a different build configurations and create more accelerators.

Workaround

For a specific SOPC Builder system and Nios II IDE project, specify C2H accelerators in only one build configuration. You can use multiple build configurations, as long as only one configuration specifies C2H Compiler settings.

Solution Status

Not fixed

Hardware Accelerators Remain After Deleting the Software Project

If a system contains C2H accelerators, deleting the software project that defines the accelerators does not remove the accelerators from the hardware system, and the accelerator logic remains in the SOPC Builder system.

Workaround

To remove an accelerator from a system, delete the accelerator from the C2H view in the Nios II IDE first, and then recompile the software project. The C2H Compiler then removes the accelerator from the SOPC Builder system. Once the compilation is complete then the software application can be deleted from the workspace.

Solution Status

Not fixed

Incorrect Results From Logical or Conditional Operation With Side-Effects

The C2H Compiler always evaluates both operands of logical (&&, `||`) and conditional (`?:`) operators. This is different from expected American National Standards Institute (ANSI) C behavior, where operands are evaluated left-to-right, and unnecessary operands are skipped.

For example, in the expression `(i-- && j--)`, if the value of `i` is zero, ANSI C does not evaluate the right-hand-side (RHS) expression, and `j` is not decremented. By contrast, the C2H Compiler evaluates both sides, decrementing `j`.

The following expressions are other examples that might be affected by this issue:

```
if (i-- || j++)
    ...;


a = ((cond == 1)?  i++ : j++);
```

Workaround

Use logical and conditional operations whose operators have no side effects. Operations with side effects include pre- and post- increment and decrement operations (`++`, `--`), memory operations (`*`, `[]`, `.`, `->`), and function calls.

Solution Status

Not fixed

Launch SOPC Builder Button in C2H View

When the Nios II IDE workspace contains multiple projects with multiple system libraries, the incorrect SOPC Builder system might open when you click **Launch SOPC Builder** in the C2H view.

Workaround

Launch SOPC Builder from the Quartus II software. Alternatively, keep only one system library project open at a time while using the C2H Compiler.

Solution Status

Not fixed

Development Board Errata

This section describes in detail the Nios II EDS issues related to Nios development boards.

Intermittent Failures While Accessing CompactFlash Card

The Nios II Development Kit version 5.0 and later includes a CompactFlash controller peripheral suitable for interfacing to CompactFlash cards in True IDE mode on Nios development boards. For True IDE mode to operate, CompactFlash cards require that the `ATASEL_N` input be driven to ground during power-up.

The CompactFlash controller peripheral includes a configurable power register, used to cycle power to CompactFlash cards from Nios II software through a metal oxide semiconductor field-effect transistor (MOSFET) on the Nios development boards. However, in certain development boards, power to the CompactFlash card does not turn off completely during this power cycle operation. Because of this condition, the CompactFlash might not sample the `ATASEL_N` pin during the power-cycle operation after FPGA configuration when this pin is driven to ground. Instead, the CompactFlash card might sample the `ATASEL_N` pin when power is first applied to the development board, when I/O is not yet driven by the FPGA (before FPGA configuration).

Workaround

If you encounter errors with CompactFlash when using the Nios development boards, try one of the following solutions:

- Use a different CompactFlash card. Certain cards are more susceptible to the power-cycling issue than others.
- Modify the Nios development board. This is recommended if you are familiar and comfortable with board-level modifications. Disconnect pin 9 (`ATASEL_N`) on the CompactFlash socket on your Nios development board and tie this pin to ground.

☞ The CompactFlash socket uses a staggered numbering on the pins (starting from pin 1: 1, 26, 2, 27, ...); refer to the CompactFlash Association specification for right-angle surface-mount connectors for exact specifications on this connector. This modification permanently enables True IDE mode operation.

Solution Status

Not fixed

Documentation Errata

This section describes in detail the Nios II EDS documentation issues.

Compiler Flags for Building Custom Newlib

"Common BSP Tasks" in the *Using the Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook* describes how to compile a custom version of the newlib library. However, it does not list the compiler flags that must be selected to generate the correct library.

Workaround

The missing information is as follows:

The Nios II EDS provides a number of precompiled newlib implementations. The provided libraries are precompiled with every viable combination of the GCC compiler flags shown in Table 4.

When you create a BSP with a precompiled newlib, the Nios II software build tools select the newlib matching your BSP's compiler settings. When you create a custom newlib, you must ensure that the compiler flags listed in Table 4 match your BSP's settings.

☞ The Nios II-specific compiler flag `-mcustom-fpu-cfg` requires the Nios II processor core to be implemented with the floating-point custom instruction as specified in the flag argument. The compiler flag can have one of the following two argument values:

- `-mcustom-fpu-cfg=60-1`—Use the Nios II floating-point custom instruction without divider
- `-mcustom-fpu-cfg=60-2`—Use the Nios II floating-point custom instruction with divider

For details about the Nios II-specific compiler flags `-mstack-check`, `-mno-hw-mul`, and `-mhw-mulx`, refer to "Altera Nios II Options" in *Using the GNU Compiler Collection (GCC)*, installed with the Nios II EDS. For details about the `-pg` compiler flag, refer to "Compiling a Program for Profiling" in *GNU Profiler*, installed with the Nios II EDS. To find the installed documentation, on the Windows Start menu, click **Programs** > **Altera** > **Nios II EDS <*version*>** > **Nios II <*version*> Documentation**, and then click **Literature**.

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Table 4. GCC Compiler Options for Newlib

Option Name	Purpose
`-pg`	Link for profiling with gprof
`-EB`	Generate big-endian code
`-mstack-check`	Enable stack checking
`-mno-hw-mul`	Disable use of the `mul` family of instructions
`-mhw-mulx`	Enable use of the `mulx` family of instructions
`-mcustom-fpu-cfg`	Use a floating point custom instruction

Obsolete Wiki URLs in Help Systems and Handbook

The following documents contain obsolete URLs for the Nios Community Wiki:

- The *Nios II Software Developer's Handbook*
- The *Nios II Embedded Design Handbook*
- The Nios II IDE help system
- The Nios II Studio help system

Workaround

You can visit the Nios Community Wiki at www.nioswiki.com.

Solution Status

Fixed in v. 9.0.

Error Message on Invalid Exception or Reset Vector

New Nios II compiler error messages, added in v 8.1, are not documented in the *Nios II Software Developer's Handbook*.

Workaround

The missing information is as follows:

While building a Nios II software project, the compiler produces an error message if the SOPC Builder system defines an invalid exception or reset vector. Table 5 shows possible error messages and their causes.

Table 5. Exception and Reset Vector Error Messages (Part 1 of 2)

Message	Cause
`The section mapping ".entry" is not mapped to a memory region.`	The `.entry` section is mapped to a nonexistent memory region.
`The section mapping ".exceptions" is not mapped to a memory region.`	The `.exceptions` section is mapped to a nonexistent memory region.
`The section mapping ".entry" maps to a memory region "`*<region name>*`" which is not the required length "32".`	The `.entry` section is mapped to a memory region that is smaller than 32 bytes.

Table 5. Exception and Reset Vector Error Messages (Part 2 of 2)

Message	Cause
`The section mapping ".entry" maps to a memory region "`*<region name>*`" which does not start at the SOPC Design specified address "`*<base address>*`".`	The `.entry` section is mapped to a memory region that does not start at the reset vector base address.
`The section mapping ".exceptions" maps to a memory region "`*<region name>*`" which does not start at the SOPC Design specified address "`*<base address>*`".`	The `.exceptions` section is mapped to a memory region that does not start at the exception vector base address.
`The SOPC design Nios II reset and exception vector addresses are mapped to the same memory. For HAL the exception address must be at least 32 bytes larger than the reset address.`	The `.entry` section and `.exceptions` section map to the same memory device, and the base of the `.exceptions` section is less than 32 bytes above the base of the `.entry` section.
`The section mapping ".exceptions" mapped device "`*<memory 1>*`" is not the same as the SOPC Design specified device "`*<memory 2>*`".`	The `.exceptions` section is mapped to a device that is different from the exception vector device specified in the SOPC Builder system.
`The section mapping ".entry" mapped device "`*<memory 1>*`" is not the same as the SOPC Design specified device "`*<memory 2>*`".`	The `.entry` section is mapped to a device that is different from the reset vector device specified in the SOPC Builder system.

In the SOPC Builder system, ensure that the exception and reset vectors conform to all the criteria listed in Table 5.

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Missing Source Control Information

The Quartus II Handbook refers to the *Using the Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook* for information about putting Nios II projects in source control. This information does not appear in the referenced chapter.

Workaround

The missing information follows.

Archiving Nios II IDE Software Projects

This section helps you identify the files you must include when archiving a Nios II IDE software project. With this information, you can archive a Nios II application project and its associated Nios II system library project.

You might want to archive your projects for one of the following reasons:

- To place them under source control
- To create backups
- To bundle the projects for transfer to another location

This section covers the following information:

- How to find and identify the files that you must include in an archived Nios II IDE software project.
- Which files must have write permission to allow the software projects to be built.

Table 6 and Table 7 list the files required by Nios II IDE software projects. This is the minimum set of files needed to completely rebuild a software project, including the executable and linking file (**.elf**).

Archive your Nios II IDE software projects together with the SOPC Builder system on which they are based. You cannot rebuild a Nios II IDE software project without its associated SOPC Builder system.

The files listed in Table 6 are located in the Nios II application project directory.

Table 6. Files Required for a Nios II Application Project

File Description	File Name	Write Permission Required?
All source files	for example: **app.c**, **header.h**, **assembly.s**, **lookuptable.dat**	No
Eclipse project file	**.project**	No
C/C++ Development Toolkit project file	**.cdtproject**	Yes
C/C++ Development Toolkit option file	**.cdtbuild**	No
Software configuration file	**application.stf**	No

The files listed in Table 7 are located in the Nios II system library project directory.

Table 7. Files Required for a Nios II System Library Project

File Description	File Name	Write Permission Required?
Eclipse project file	**.project**	Yes
C/C++ Development Toolkit project file	**.cdtproject**	Yes
C/C++ Development Toolkit option file	**.cdtbuild**	No
System software configuration file	**system.stf**	Yes

☞ You must have write permission for certain files. The tools write to these files as part of the build process. If the files are not writable, the toolchain fails. Many source control tools mark local files read-only by default. In this case, you must override this behavior. You do not have to check the files out of source control unless you are modifying the Nios II software project.

Solution Status

Not fixed

Nios II IDE Online Help Expand Buttons Do Not Work

The expand text arrow and the **Show All** option in the online help do not work in Internet Explorer.

Workaround

In Internet Explorer, carry out the following actions:

1. Click the **Refresh/Show Current Topic** icon (yellow arrows icon at the top of the search pane). The table of contents appears with the selected topic highlighted.
2. Click the highlighted topic to refresh the browser frame. Expanded text works.

Alternatively, use Firefox 2.0.0.6.

Solution Status

Not fixed

Hardware Example Design Errata

This section describes in detail the Nios II EDS hardware example design issues.

Incorrect PLL Phase Shift in Stratix II and Cyclone II Designs

In several example designs targeting the Nios II Cyclone® II 2C35 and Nios II Stratix® II 2S60 RoHS development boards, the synchronous static random access memory (SSRAM) and legacy dual data rate synchronous dynamic random access memory (DDR SDRAM) clocks are implemented with an incorrect phase-locked loop (PLL) phase shift.

This issue affects the following example designs:

- 2s60 RoHS TSE SG-DMA
- 2s60 RoHS Standard
- 2s60 RoHS Full Featured
- 2c35 TSE SG-DMA
- 2c35 Standard
- 2c35 Full Featured

This issue also affects any custom 2c35 or 2s60 RoHS design using SSRAM or legacy DDR SDRAM.

The incorrect phase shift might cause timing violations when reading from or writing to SSRAM or legacy DDR SDRAM.

Workaround

To correct the PLL settings, perform the following steps:

1. Open the standard design in SOPC Builder.
2. Double-click the component instance named `pll` to launch the PLL MegaWizard interface.
3. Click **Launch Altera's ALTPLL MegaWizard** to launch the MegaWizard interface.
4. Select the **Output Clocks** page and then the clock page used to generate the SSRAM clock. Depending on the particular design, this might be **clk c1**, **clk c2**, or another clock.

5. Ensure that **Clock phase shift** is –3.38 ns.
6. Click the clock page used to generate the write clock for the legacy DDR SDRAM, if any. Depending on the particular design, this might be **clk c1**, **clk c2**, or another clock.
7. Ensure that **Clock phase shift** is 270 degrees.
8. Click **Finish** to save changes and exit the PLL MegaWizard interface.
9. Click **Finish** to save settings to the PLL instance in SOPC Builder.
10. Regenerate the system in SOPC Builder and recompile in the Quartus II software. Refer to the **readme.txt** file in the design example folder for information on compiling designs containing the DDR controller.

For further information refer to **ssram_interface_readme.html**, located in the **<*Quartus II installation directory*>/sopc_builder/documents** folder. This document discusses the SSRAM timing analysis methodology in detail. Additionally, *AN 411: Understanding PLL Timing for Stratix II Devices* discusses clock phase shift calculations and assignments for PLLs in Stratix II devices.

Solution Status

Not fixed

Software Example Errata

This section describes in detail the Nios II EDS software example issues.

Simple Socket Server Example Displays Incorrect IP Address

The simple socket server networking software example displays an incorrect IP address for the incoming connection.

Workaround

There is currently no workaround for this problem. Ignore the IP address of any incoming connection that is displayed on the console.

Solution Status

Not fixed

Hardware Tutorial Software Example Hangs on Some Boards

The **count binary** application used in the tutorial might hang when run on some designs that do not contain a liquid crystal display (LCD) peripheral.

Workaround

Modify line 18 of the **count_binary.h** file, as follows:

```
# define LCD_PRINTF(lcd, args...) /* Do Nothing */
```

Rebuild and run the software again.

Solution Status

Not fixed

Networking Examples

If you are running a networking software example, you might be asked for a nine-digit number. You are directed to find this number on a sticker on your Nios development board, identified by the prefix **ASJ**. Not all Nios development boards have this sticker.

Workaround

If your Nios development board does not have a sticker with the letters **ASJ** followed by a nine-digit number, enter a unique nine-digit number when prompted. To avoid network address conflicts, ensure that this number is unique to each Nios board connected to your network.

Solution Status

Not fixed

Flash Programmer Errata

This section describes in detail the Nios II EDS issues related to the flash programmer.

Delay When Creating New Flash Programmer Configuration

You might experience a delay of several seconds when creating a new Flash Programmer configuration in the Nios II IDE.

Workaround

There is no workaround.

Solution Status

Not fixed

elf2flash File Size Limit

The **elf2flash** utility supports **.elf** files up to approximately 24 MBytes in size. The **elf2flash** utility might fail on files larger than 24 MBytes, with the error message `java.lang.OutOfMemoryError`.

Workaround

Lower the number of symbols in your **.elf** file by turning off debug symbols.

Alternatively, specify less initialized data in the application.

Solution Status

Not fixed

Error Message "No such file or directory" When Programming Flash

When programming flash from a project stored in a path containing spaces, you might receive the spurious error message **No such file or directory**. The flash programmer does not correctly handle spaces in the directory path. However, this error is harmless, because flash programming completes successfully.

Workaround

None required.

Solution Status

Not fixed

Hardware Simulation Errata

This section describes in detail the Nios II EDS issues related to hardware simulation.

Error "UNC paths are not supported" Launching ModelSim

If you launch ModelSim from a working directory that is mapped via a universal naming convention (UNC) path (a path that starts with `//` instead of drive letter), you receive the following error message in SOPC Builder: **UNC paths are not supported. Defaulting to Windows directory**. This error occurs because ModelSim is calling a command shell, which does not support UNC paths.

Workaround

Map the UNC path to a drive letter and use the drive letter to reference the working directory in the launching shell.

Solution Status

Not fixed

Uninitialized .bss Variables in Simulation

If your program reads the value of an uninitialized `.bss` variable during HDL simulation, and the BSP (system library) is compiled with the **ModelSim only, no hardware support** property enabled in Nios II IDE, a warning appears about unfiltered data being `'x'`. This warning appears because when this property is enabled, the code that clears the `.bss` memory region is omitted to speed up HDL simulation so this memory region is uninitialized. The `.bss` region contains global and static local variables that are not initialized by the application so they default to a value of zero. When the Nios II processor reads uninitialized variables, it displays a warning and converts any of the bits of the uninitialized data to zero which correctly mimics the effect of the missing `.bss` clearing code. The HAL code that executes before and after `main()` might use `.bss` variables, so these warnings might appear even if your application does not use the `.bss` section.

Solution Status

Not fixed

Host Platform Errata

This section describes in detail the Nios II EDS issues related to the host development platform.

Windows/Cygwin: Nios II Processor Generation Failure

You might see the following error message when you generate your Nios II system in SOPC Builder:

```
Error: Generator program for module 'cpu_fpoint' did NOT run
successfully
```

This error might occur on a Windows-based system when all of the following conditions are true:

- You have installed a version of Cygwin other than the one distributed with the Quartus II software.
- You launch the Quartus II software and SOPC Builder from a Nios II SDK Shell.
- You enable the floating point custom instruction (FPCI) in the Nios II processor core.

Workaround

Launch the Quartus II software and SOPC Builder from the Windows Start menu, and regenerate your system.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Windows Vista: Limited Support in Nios II EDS

The Quartus II software introduces Windows Vista (32-bit and 64-bit) support in version 7.2. However, the Nios II Embedded Design Suite supports only the following platforms:

- Windows 2000
- Windows XP
- Windows XP (64)
- Windows Vista Business Edition (32-bit)
- SUSE 9 (32-bit)
- SUSE 9 (64-bit)
- RedHAT Linux v3.0 (32-bit)
- RedHAT Linux v3.0 (64-bit)
- RedHAT Linux v4.0 (32-bit)
- RedHAT Linux v4.0 (64-bit)

Workaround

There is no workaround.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Linux: Process Fails To Terminate While Debugging With the Nios II ISS Target

If you try to interrupt or terminate a debug session targeting the Nios II ISS, you might see an error message **Interrupt Failed or Terminate Failed**. This means that the `nios2-iss` process failed to terminate. The debug session appears to have terminated in the IDE, but the `nios2-iss` process still continues running.

Workaround

Open a command shell and kill the `nios2-iss` process as follows:

1. Type:

   `jobs`

   to get a list of process IDs.

2. Type:

   `kill -9` *<nios2-iss process ID>*

   to terminate the process.

Solution Status

Not fixed

Nios II IDE Errata: Building Projects

This section describes in detail the issues related to building projects in the Nios II IDE.

Nios II IDE Command-Line Tools Select Wrong Workspace on Linux

On Linux systems, the Nios II IDE command-line tools use the default Eclipse project workspace, regardless what workspace you are using in the IDE. The default workspace is at `$SOPC_KIT_NIOS2/eclipse/nios2-ide-workspace-`*<version>*.

This issue affects the following tools:

- `nios2-create-system-library`
- `nios2-create-application-project`
- `nios2-build-project`
- `nios2-import-project`
- `nios2-delete-project`

Workaround

Explicitly supply the Nios II IDE command-line tools with a workspace location, by means of the `-data` command-line argument. The syntax of the `-data` argument is as follows:

`-data` *<path to workspace>*

The path to the workspace must be absolute, and must not contain whitespace. Aside from these restrictions, any valid file system path can be used for the workspace. Example 4 shows how to specify a workspace path.

Example 4. Providing a Non-Default Workspace Location in Linux

```
nios2-create-project -data $HOME/myworkspace <other arguments>↵
```

Solution Status

Not fixed

Nios II IDE Command-Line Tools Hang on Windows

On Windows systems, the Nios II IDE command-line tools are sometimes unable to locate the project workspace. When this happens, the tools might hang.

This issue potentially affects the following tools:

- `nios2-create-system-library`
- `nios2-create-application-project`
- `nios2-build-project`
- `nios2-import-project`
- `nios2-delete-project`

Workaround

Explicitly supply the Nios II IDE command-line tools with a workspace location, by means of the `-data` command-line argument. The syntax of the `-data` argument is as follows:

`-data` *<path to workspace>*

The path to the workspace must be absolute, and must not contain whitespace. Aside from these restrictions, any valid file system path can be used for the workspace. The workspace shown in Example 5 is the default workspace used by the Nios II IDE in version 9.0.

Example 5. Providing a Workspace Location

```
nios2-create-project -data c:/altera/90/nios2eds/\
    bin/eclipse/nios2-ide-workspace-9.0 <other arguments>
```

Solution Status

Not fixed

Nios II IDE Cannot Find stdio.h in Outline View

If you create a new project using the Nios II IDE and try to open the **stdio.h** file from the Outline view before building the project, the IDE displays the error message **No include files were found that matched that name**.

Workaround

Build the project before attempting to open **stdio.h**.

Solution Status

Not fixed

Nios II IDE Hangs With UNC Project Path

In the **New Project** dialog box, if you turn on **Specify Location** and specify the path in UNC form, the IDE might hang.

Workaround

Map the UNC path into a remote drive which looks like a Windows drive to the Nios II IDE.

Solution Status

Not fixed

Build Failure with Nios II Advanced Exceptions, MMU, and MPU

Projects created in the Nios II IDE for an SOPC Builder system containing a Nios II processor configured with advanced exceptions, the memory management unit (MMU), or the MPU fail to build, and generate the following error messages:

```
ERROR - Classic build flow for Nios II system library does not support
the Nios II \
    advanced exceptions.
Use the non-classic Nios II Board Support Package instead.
<timestamp> - (SEVERE) generate: java.lang.IllegalStateException: \
    java.lang.IllegalStateException: \
    com.altera.ingenuous.GTFElement.GTFElementErrorException: \
    <error> element in GTF script
make[1]: *** [system_description/../obj/system.h-t] Error 1
make: *** [system_project] Error 2
```

Workaround

Create the software project with the Nios II software build tools command-line development flow. Alternatively, use the Nios II Studio.

Solution Status

Not fixed

Build Command Not Functional for BSPs Created Using the Software Build Flow

The build option in the Nios II IDE menu does not rebuild BSPs imported into the IDE.

Workaround

To build the BSP, build the associated application project.

Solution Status

Not fixed

Linker Errors with Dual-Port Memories

If your instruction master and data master ports are connected to the same dual-port memory and the ports have different addresses, your code fails to run or you experience a linker error. The Nios II IDE does not warn you of the addressing violation.

Workaround

Assign the same address to both ports of the dual-port memory.

Solution Status

Not fixed

User-Managed BSP Settings Not Supported in Nios II IDE

For projects created using the new software build flow and imported into the Nios II IDE, the IDE configuration settings have no effect.

For example, objdump, compiler, and linker settings made in the IDE are ignored. This behavior occurs because Nios II software build tools projects are not IDE-managed projects. In addition, the make-related preferences do not pertain to imported Nios II software build tools projects. The IDE ignores these options during the build process.

Workaround

Make these settings in the project's makefile.

Solution Status

Not fixed

URL Project Location Causes Project Creation Error

When you try to create a new project in an existing workspace, you might see an error dialog box saying:

```
Project cannot be created. Reason: Internal Error
```

This error might occur if the path to any project in the workspace is a Uniform Resource Locator (URL) location, for example **file:/F:/Design**. To view the path, right-click on the project and select **Properties**.

Workaround

Import your existing application and system library projects into a new workspace.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Compilation Error with Separate Exception Stack Option

Choosing the **Use a separate exception stack option** might cause the compilation error `"UNDEFINED VARIABLE %STACK_POINTER%"` when building a project. This error occurs if the exception stack is larger than the memory available for it.

Workaround

On the system library properties page for the project, turn off the separate exception stack or reduce the **Maximum exception stack size** setting.

Solution Status

Not fixed

Incorrect Stack and Heap Free Space Report

The makefile reports an incorrect number of bytes free for the stack and heap, if the heap and stack are in different memory regions.

Workaround

No workaround available.

Solution Status

Not fixed

Nios II IDE Reports Problems Without Displaying Error in Console

When building a project, the Nios II IDE reports problems, but the build output in the console does not contain any errors.

The Nios II IDE incorrectly reports some linker warnings as errors, with a dialog box saying **Errors exist in a required project**. The Dhrystone software example exhibits this behavior, and recompiling the project again makes the issue go away.

Workaround

If the Console output does not contain errors, then the project built correctly. On subsequent builds, the linker step is skipped and the errors do not appear.

Solution Status

Not fixed

Nios II IDE Errata: Debugging Projects

This section describes in detail the issues related to debugging projects in the Nios II IDE.

IDE Cannot Display Imported Profiling Data

If you create a software project in the Nios II software build tools command line flow, generate profiling data to **gmon.out** in the Nios II command shell, and then import the project into the IDE, you cannot use the Profiling perspective to view the profiling data. When you attempt to view **gmon.out**, the IDE displays the error message **nios2-elf-gprof: ../: not in a.out format**.

Workaround

Run `nios2-elf-gprof` from the Nios II command shell to generate a profiler report.

Solution Status

Not fixed

Error Message when Downloading .elf File

If the **Build Automatically** option is turned on in the IDE, you might get the following error when downloading an executable and linking formal file (**.elf**):

```
"2 [main] sh 5736 fork: child 3892 - died waiting; \
    for longjmp before initialization, errno 11"
```

The **Build Automatically** option is off by default, as recommended by the Eclipse help system.

Workaround

Turn off the **Build Automatically** option and download the **.elf** file again.

Solution Status

Not fixed

Variable Casting Unsupported in ISS

Casting variables in the Variables view in the Debug perspective when using the Nios II ISS might cause an exception dialog box to open.

This exception might occur if you select **Debug as ISS** in the Nios II IDE and try to cast variables via the Variables view.

Workaround

There is no workaround.

Solution Status

Not fixed

Trace Debug Does Not Support Instruction-Related Exceptions

The new instruction-related exception handling introduced in version 8.0 is not supported by the Nios II trace tools.

Workaround

There is no workaround.

Solution Status

Not fixed

Error Accessing MMU and MPU Registers in the Debugger

You might get the following error if you try to read or write an MMU or MPU register via the Nios II Debugger:

```
nios2-gdb-server-wrapped: nios2debug.cpp:538: virtual ADI_ERROR \
    NIOS2DEBUG::leave_debug_mode(): Assertion 'm_state == STATE_DEBUG' failed.
```

Workaround

To read an MPU region, execute the following steps:

1. Set region `INDEX` in the `MPUBASE` register.
2. Exit and reenter debug mode, that is, single step.
3. Set the `RD` bit in the `MPUACC` register.
4. Exit and reenter debug mode, that is, single step.
5. Read back `MPUBASE` for pertinent information.
6. Read back `MPUACC` for pertinent information.

To set an MPU region with consecutive value changes, only the last one, when leaving debug mode, is committed.

Solution Status

Not fixed

Trace Debug Does Not Support the JMPI Instruction

The new JMPI instruction introduced in version 7.2 is not supported by the Nios II trace tools.

Workaround

There is no workaround.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Cannot Locate Source Code in Driver Files Shared by Multiple Projects

If you hit a breakpoint in a driver file, and that driver file is shared with another project that is closed, the Nios II IDE might indicate that it cannot locate the source code.

Workaround

Open the closed system library project and resume debugging.

Solution Status

Not fixed

Console Window Is Not Updated After ISS Error

After performing a **Run as ISS**, if you receive an ISS error in the console window, the console is not updated subsequently.

Workaround

Close the console window after receiving an ISS error. A new console window opens when a new message is available.

Solution Status

Not fixed

ISS Fails on Designs Containing TSE MAC or SG-DMA Components

You receive an Internal Error when attempting to perform an ISS simulation of designs containing the Altera TSE MAC or SG-DMA components because the Nios II ISS does not support these components.

Workaround

Remove the TSE MAC and SG-DMA components from your system and perform ISS simulation on the simplified system. You can also simulate the design in ModelSim or test it on hardware.

☞ Before removing the TSE MAC and SG-DMA components, make a copy of the unmodified system to ensure that you can return to the original configuration.

Solution Status

Not fixed

Memory Window Sets Control Register Values Incorrectly

The memory window might incorrectly set values in memory-mapped control registers. For example, writing 0x1234 to a byte addressed register results in the value 0x3434 being stored in the register. The memory window shows this incorrect value.

Workaround

Use the GNU debugger (GDB) console window in the IDE, instead of the memory window, to write to the registers. For example, type the following command:

```
set {int} <register address>=0x1234↵
```

☞ You must refresh the memory window for it to correctly display the target value.

Solution Status

Not fixed

Programs That Interact With a Terminal Console on Windows Do Not Work

Programs with this behavior work in v. 6.0 and earlier, but do not work in Nios II IDE v. 6.1 and later.

The Eclipse platform in v. 6.1 and later of the IDE (on Windows only) sends the string `\r\n` instead of just `\n` when running or debugging using the Terminal. This behavior can break existing software designs, and it is inconsistent with `nios2-terminal`, which still just sends `\n`.

Workaround

Change the software running on the Nios II processor to parse for `\r\n` as well as `\n`.

Solution Status

Not fixed

“Run as ModelSim” in the Nios II IDE Fails

The **Run as ModelSim** command might fail on launch configurations created in v. 7.0 or earlier of the IDE. This problem does not occur for new launch configurations.

Workaround

Select a location for the ModelSim tool from the launch configuration dialog box. You can use the **Browse** button next to the **ModelSim path** group, or type in a path to the ModelSim directory (for example **c:/altera/71/modelsim_ae/win32aloem**).

Solution Status

Not fixed

The Restart Command on the Run Menu Does Not Work

The **Restart** command on the **Run** menu does not work.

Workaround

Stop the program, then debug it again. If the debugger is hung in an endless loop, use the following **bash** alias to break the target, then stop it:

```
alias break="kill -2 \'ps -a | grep nios2-elf-gdb | cut -f6 -d' '\'"
```

Solution Status

Not fixed

Watchpoints Do Not Work on Certain Variables

Watchpoints do not work on a variable whose size is not 32 bits.

Workaround

Change the types of global and static local variables to `int`, `long`, or `unsigned long` before setting watchpoints on them.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Nios II IDE Errata: Navigating Projects

This section describes in detail the issues related to navigating projects in the Nios II IDE.

Nios II IDE Freezes While Displaying the Splash Screen

After clicking **Switch Workspace** on the File menu on a Windows machine, a Nios II IDE splash screen appears. Unfortunately, this splash screen obscures the dialog box that asks you to specify the new workspace. As a result, the IDE appears to freeze.

Workaround

Press Alt-Tab to switch applications. Two relevant application icons appear: an Eclipse icon associated with the splash screen and a Nios II IDE icon associated with the **Workspace** dialog box. Select the Nios II icon to bring the dialog box to the foreground.

Solution Status

Not fixed

Internal Error When Double-Clicking on a Large Objdump File

On Windows when opening a large **objdump** file in the Nios II IDE, you might get the following error message:

```
Unable to create this part due to an internal error.
Reason for the failure: Editor could not be initialized.
```

Workaround

Adjust the Windows launch arguments for the Nios II IDE editor. Perform the following steps:

1. On the Windows Start menu, browse to the **Nios II EDS** program icon, right-click it, then click **Properties**. The **Windows Properties** dialog box appears.

2. In the **Target** field, append `"vmargs -Xmx1024m"` to the end of the path to the Nios II IDE executable. For example:

```
C:\altera\72\nios2eds\bin\eclipse\nios2-ide.exe -vmargs -Xmx1024m
```

Solution Status

Not fixed

C/C++ Scanner Does Not Support Certain C/C++ Constructs

The C/C++ scanner performs C/C++ Search, navigation, open declaration, and parts of content assist. Due to limitations of the C/C++ scanner, these features do not work with C code constructs not supported by the C++ language. An example is functions that take a function pointer as an argument.

Workaround

If the C/C++ Search fails, use the File Search facility.

Solution Status

Not fixed

Nios II Studio Errata

This section describes in detail the issues related to the Nios II Studio.

Incorrect Default Workspace Name

When first launched, the Nios II Studio prompts for a workspace folder name. The default workspace folder name is **nios2-studio-workspace-8.0**. If you accept the default workspace, and v8.0 of the Nios II EDS was previously installed on your system, the Studio might modify project information from the previous installation.

Workaround

To keep your Studio 9.0 projects separate from your Studio 8.0 projects, manually insert a different workspace name, such as **nios2-studio-workspace-9.0**.

Solution Status

Not fixed

Error in nios2-studio Help System

In the Windows operating system, with the Nios II Studio help system viewed in the internal browser, the following error message appears when you click any link:

A Runtime Error has occurred

Workaround

Use an external browser. To configure Nios II Studio to display the help system in an external browser, perform the following steps:

1. On the Window menu, click **Preferences**.

2. In the **Preferences** dialog box, click **Help**, and turn on **Use external browser**.
3. Expand **General** and click **Web Browser**.
4. Select **Use external Web browser**.

Solution Status

Not fixed

No System ID Validation Performed

The **Validate Nios II system ID before software download** option does not perform System ID validation on **.elf** files built outside Nios II Studio unless the System ID is specified.

Workaround

Specify the `--sidp` and `--id` arguments in the **Tcl** extra arguments field of the **Main** tab of your launch configuration.

Solution Status

Not fixed

Spurious Messages When Importing Pre-8.0 Projects

Nios II Studio gives the following message when you import a project from a pre-8.0 version.

```
The project nature will be imported unchanged. If you wish to change the project \
    to a command-line project, remove the .project and .cdtproject files before \
    finishing this import.
```

The message is incorrect. The project is actually upgraded to a Nios II Studio application or BSP. You do not need to remove the specified files.

The following errors might also appear and can be ignored.

```
Unable to invoke: nios2-app-update-makefile --app-dir <app_dir> \
    --add-src-rdir .settings obj
SEVERE: nios2-app-update-makefile failed.
SEVERE: UserErrorException: The following option has the incorrect number of arguments \
    "--add-src-rdir".
```

Workaround

No action needed.

Solution Status

Not fixed

public.mk Error or System Failure After Hardware Modification

If you modify your SOPC Builder system and do not regenerate your BSP in the Nios II Studio, you might receive the following error:

```
make: *** [public.mk] Error 1
```

or see system failures. System failures occur because there is a system ID mismatch that is not detected by the tools.

Workaround

Always regenerate your BSP project if you modify your SOPC Builder system.

Solution Status

Not fixed

Nios II Software Build Tools Errata

This section describes in detail the issues in the Nios II software build tools.

Cannot Include C++ File in Driver or Software Package

In the Nios II software build tools prior to v. 8.1, you cannot configure a device driver or software package to include a C++ source file. You can include a C source file with the `add-sw-property` Tcl command, using the `c_source` property. However, there is no equivalent property for C++ source files.

Workaround

Upgrade to the Nios II EDS v. 8.1 or later, and use the `cpp_source` property.

Solution Status

Fixed in v. 8.1

Incorrect Linker Errors

While building a HAL-based Nios II software project, the linker displays misleading error messages if the SOPC Builder system provides less than 32 bytes of memory at the reset vector. The linker errors are similar to the following:

```
/cygdrive/d/altera/80/nios2eds/bin/nios2-gnutools/H-i686-pc-cygwin/bin/../lib/gcc/ \
    nios2-elf/3.4.6/../../../../nios2-elf/bin/ld:/cygdrive/d/test3/software/ \
    hello_world_0_syslib/Debug/system_description/generated.x:186: warning: memory \
    region UNDEFINED not declared
/cygdrive/d/altera/80/nios2eds/bin/nios2-gnutools/H-i686-pc-cygwin/bin/../lib/gcc/ \
    nios2-elf/3.4.6/../../../../nios2-elf/bin/ld:/cygdrive/d/test3/software/ \
    hello_world_0_syslib/Debug/system_description/generated.x:186: parse error
collect2: ld returned 1 exit status
make: *** [hello_world_0.elf] Error 1
```

Workaround

In the SOPC Builder system, if the reset and exception vectors are in the same memory device, ensure that the exception vector is at least 32 bytes above the reset vector.

Solution Status

Fixed in v. 8.1

Java Heap Space Error When Creating BSP

When creating a BSP, the Nios II software build tools might hang with the following message:

`SEVERE: OutOfMemoryError: Java Heap Space`.

This problem occurs when your SOPC Builder design has connection cycles. A connection cycle consists of a group of components that master one another in a circular arrangement. For example, if `ddr_sdram_bridge` masters `lcd_sgdma`, and `lcd_sgdma` masters `ddr_sdram_bridge`, the two components form a simple connection cycle. In this situation, the software build tools are unable to create a BSP.

Workaround

Avoid connection cycles in the SOPC Builder system.

Solution Status

Not fixed

BSP Editor Incorrectly Allows Mixed Drivers

If multiple drivers are found for a single peripheral type, and you have multiple instances of the peripheral type in your SOPC Builder system, the Nios II BSP editor allows you to manually select a different driver for each instance of the peripheral or processor. For example, if you have two `altera_avalon_timer` devices, you can select two different timer drivers.

This behavior is not fully supported. You must choose the same driver for both instances of the peripheral type.

You can make driver assignments in one of the following ways:

- In the Nios II BSP editor, within the **Drivers** tab. The **Driver name** associated with each peripheral can be edited, if multiple driver types are found that support the peripheral. You must make the same driver type selection for each instance of a peripheral type in the list of peripherals as defined in the **Module Name** column.
- With the Nios II software build tools in command line mode (`nios2-bsp`, `nios2-bsp-create-settings`, `nios2-bsp-update-settings`): the **set_driver** command can be invoked (either from a Tcl script or with the `--cmd` option). The **set_driver** command accepts a peripheral module instance name and driver name as arguments. If you are using the command line development flow and wish to use this command, you must invoke it once for each instance of a peripheral type in your system, similar to the BSP editor flow described above.

Workaround

There is no workaround.

Solution Status

Not fixed

nios2-bsp --help Displays Incorrect List of BSP Types

In version 8.0, the Nios II software build tools support additional HAL-based operating systems, when they are added as Tcl-scripted software packages. However, the `nios2-bsp` tool help message indicates that there are only two BSP types, HAL and UCOSII. If other HAL-based operating system components are added they do not appear in the `nios2-bsp` help message.

Workaround

See "nios2-bsp-create-settings Fails to List BSP Types" to get a correct, dynamic list of available BSP types.

Solution Status

Not fixed

nios2-bsp-create-settings Fails to List BSP Types

The help message for `nios2-bsp-create-settings` incorrectly states that using the `--type` option with no value creates a list of available BSP types.

Workaround

To get a list of available BSP types, execute the following command:

```
nios2-bsp-create-settings --type x --settings x --sopc=<existing sopcinfo path>↵
```

The following command returns an error message that lists available BSP types:

```
SEVERE: Available BSP type values for the --type argument are: [<list of BSP types>]
```

Solution Status

Not fixed

Error Creating BSP for Pre-8.0 System

You receive the following error message if you try to create a BSP with the v8.0 software build tools using a hardware design from v7.2 or prior.

```
SEVERE: ErrorLogException: Attempt to load "NiosII_stratixII_2s60_standard.sopc" \
    failed. The --sopc option no longer accepts *.sopc files from SOPC Builder. \
    Re-generate your system in SOPC Builder (version 8.0 or later) to create a \
    *.sopcinfo file and use that instead.
SEVERE: nios2-bsp-update-settings failed.
nios2-bsp: nios2-bsp-update-settings failed
```

Workaround

You must regenerate your v7.2 or older design in SOPC Builder and recompile in the Quartus II software before creating a BSP using the software build tools v8.0.

Solution Status

Not fixed

Makefile Error with Absolute Source Path

You might receive the following error message from the Nios II Software build tools:

```
Makefile:: *** target pattern contains no '%'
```

This error message can appear if you are using Windows, you specify an absolute source path with the `--src-dir` option, and your application directory, specified with the `--app-dir` option, is the same as your source path.

Workaround

Use a relative path to the source directory, or organize your files so that your application directory is different from your source directory.

Solution Status

Not fixed

Software Build Tools Fail if Nios II EDS is Installed in a Path Containing Spaces

The Nios II Command Line Software Build Tools fail if you install the Nios II EDS in a path containing spaces.

Workaround

Reinstall the Nios II EDS into a path that does not contain spaces.

Solution Status

Not fixed

User-Managed BSP Settings Not Supported in Nios II IDE

This issue is described on page 27.

Upper-Case File Extensions Not Supported

The Nios II software build tools for applications and libraries (`nios2-app-generate-makefile` and `nios2-lib-generate-makefile` commands) do not support source files with certain upper-case extensions. If a file with an upper-case extension is included, the make command stops with no descriptive warning.

Only Nios II assembly language files built by the C preprocessor can have upper-case file extension (**.S**). All C language files must have the extension **.c** or **.h**. C++ language source files must have the extension **.cpp**, **.cxx**, **.cc,** or **.h**.

Workaround

Rename all C language files with the extension **.c** or **.h**. Rename all C++ language files with the extension **.cpp**, **.cxx**, **.cc,** or **.h**.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Peripheral Errata

This section describes in detail the Nios II EDS issues related to peripheral components.

Unaligned Transfers of Small Payloads Fail on SG-DMA

The Scatter Gather DMA SOPC Builder peripheral does not correctly handle unaligned transfers with small payloads. A payload length smaller than the data width causes erroneous data transfers.

Workaround

Avoid using DMA devices to transfer small payloads.

If absolutely necessary, for a 32-bit SGDMA, a minimum length of 4 bytes guarantees that data is transferred correctly.

Solution Status

Not fixed

IDT71V416 SRAM Component HardCopy Incompatibility

The IDT71V416 SRAM SOPC Builder component, when generated in a clock domain running at 50 MHz or less, contains logic that triggers the following Design Assistant warning, and thus is not compatible with HardCopy devices.

```
Warning: (Medium) Rule C106: Clock signal source should not drive registers that are \
    triggered by different clock edges. Found 1 node(s) related to this rule.
```

This warning appears because a negedge-clocked register is created for the SRAM write signal, to reduce latency at lower clock speeds.

Workaround

There is no workaround.

Solution Status

Not fixed

Software Build Tools Fail With UART Driver

If you use the software build tools on a design containing the SOPC Builder UART peripheral, and if driver ioctl() support for the `altera_avalon_uart` component is enabled, then the BSP build process might fail. The **system.h** generated by the software build tools contains an invalid type for the UART parameter.

Workaround

To correct **system.h**, carry out the following steps:

1. Manually edit the **system.h** file by changing the following line:

   ```
   #define UART1_PARITY "N"
   ```

   to:

   ```
   #define UART1_PARITY 'N'
   ```

   ☞ Change double quotes to single quotes.

2. Build the BSP.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

DMA Controller Always Busy in Burst Mode

The DMA controller component (`altera_avalon_dma`), when enabled for burst transactions, does not perform transfers at widths other than its full data width. The DMA controller is always busy.

Workaround

When bursting is enabled, the DMA controller must be programmed to perform transactions at its full data width.

Solution Status

Not fixed

Non-System-Wide Reset Can Cause Improper Initialization of Mailbox Core

The `altera_avalon_mailbox` peripheral might not be initialized properly when a soft (non-system-wide) reset occurs. In this condition, mailbox contents (read and write pointers) are not reinitialized and might show potentially stale data.

Workaround

Ensure that a system-wide reset event occurs by asserting the `reset_n` input to the SOPC Builder system containing the mailbox. This resets all peripherals and Nios II processors in the system.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Target Software Errata

This section describes in detail the Nios II EDS issues related to target software packages.

Gigabit Ethernet Performance Issues

The NicheStack TCP/IP Stack - Nios II Edition might hang while sending transmission control protocol (TCP) packets to the host, or experience dropped packets when receiving TCP packets from the host. These issues can arise with version 3.1 of the NicheStack TCP/IP Stack, using the triple-speed Ethernet component to transmit over a Gigabit Ethernet point-to-point connection.

Workaround

Download and install the patches at www.altera.com/support/kdb/solutions/rd01132009_588.html.

Alternatively, upgrade to the Nios II EDS v. 9.0 or later.

Solution Status

Fixed in v. 9.0

stdio Does Not Work with MicroC/OS-II and Small C Library

`stdin`, `stdout`, and `stderr` do not work in MicroC/OS-II applications built with the Small C library option.

Workaround

Disable the small C library option.

Solution Status

Not fixed

cout From MicroC/OS-II Task Does Not Send Data to stdout

If neither `printf()` or `cout` is used from `main()` before tasks are started, `cout` does not work from a task.

Workaround

Add the following C++ code to the beginning of `main()`:

```
std::ios_base::sync_with_stdio(false);
```

Solution Status

Not fixed

malloc(), realloc() Failures With MicroC/OS-II

When you use the MicroC/OS-II RTOS, calls to `malloc()` and `realloc()` might fail if successive calls to `malloc()` or `realloc()` within a MicroC/OS-II task occur after changing the task priority of the task in which a memory block is originally allocated.

Workaround

Use one of the following workarounds:

- Allocate and/or reallocate memory blocks outside of MicroC/OS-II tasks, before task switching starts. Preallocated memory blocks make it possible to change thread priorities at runtime.
- Allocate fixed areas of memory using arrays (rather than using `malloc()`) before task switching starts. Fixed memory arrays make it possible to change thread priorities at runtime.
- Allocate memory using `malloc()` or `realloc()` from a MicroC/OS-II task. You can change task priorities at runtime, but only for tasks that have not used `malloc()` or `realloc()`.

Solution Status

Not fixed

Toolchain Errata

This section describes in detail the Nios II EDS issues related to the GNU compiler toolchain (gcc, gdb, etc.).

sof2flash Names Output Files Incorrectly

If **sof2flash** is called with the `--save` option and a relative path is specified for the output file (that is, including "./" or "../"), the raw binary file that is saved is incorrectly named **.rbf** instead of **<flash filename>.rbf**.

Workaround

Provide a full path to the output file(s), or change to the output directory before running the **sof2flash** command so the output file specification is `--output=`*<filename>*`.flash`.

Solution Status

Not fixed

GNU Assembler Does Not Accept the --defsym Flag

According the GNU documentation, you can set an assembler definition by using the `--defsym` flag, but it does not work in the following form: `--defsym MY_VAR=1`.

Workaround

There is no workaround.

Solution Status

Not fixed

How to Contact Altera

For the most up-to-date information about Altera products, refer to the following table.

Contact *(1)*	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Altera literature services	Email	literature@altera.com
Non-technical support (General)	Email	nacomp@altera.com
(Software Licensing)	Email	authorization@altera.com

Note:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

The following table shows the typographic conventions that this document uses.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Indicates command names, dialog box titles, dialog box options, and other GUI labels. For example, **Save As** dialog box. For GUI elements, capitalization matches the GUI.
bold type	Indicates directory names, project names, disk drive names, file names, file name extensions, dialog box options, software utility names, and other GUI labels. For example, **\qdesigns** directory, **d:** drive, and **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Indicates document titles. For example, *AN 519: Stratix IV Design Guidelines.*
Italic type	Indicates variables. For example, *n* + 1. Variable names are enclosed in angle brackets (< >). For example, <*file name*> and <*project name*>**.pof** file.
Initial Capital Letters	Indicates keyboard keys and menu names. For example, Delete key and the Options menu.
"Subheading Title"	Quotation marks indicate references to sections within a document and titles of Quartus II Help topics. For example, "Typographic Conventions."
`Courier type`	Indicates signal, port, register, bit, block, and primitive names. For example, `data1`, `tdi`, and `input`. Active-low signals are denoted by suffix `n`. For example, `resetn`. Indicates command line commands and anything that must be typed exactly as it appears. For example, `c:\qdesigns\tutorial\chiptrip.gdf`. Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword `SUBDESIGN`), and logic function names (for example, `TRI`).
1., 2., 3., and a., b., c., and so on.	Numbered steps indicate a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ■	Bullets indicate a list of items when the sequence of the items is not important.
☞	The hand points to information that requires special attention.

Visual Cue	Meaning
⚠ CAUTION	A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.
⚠ WARNING	A warning calls attention to a condition or possible situation that can cause you injury.
↵	The angled arrow instructs you to press Enter.
👣	The feet direct you to more information about a particular topic.

101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support
www.altera.com/support

Nios II Embedded Design Suite Release Notes and Errata

RN-EDS-3.1

About These Release Notes

These release notes cover versions 7.1 through 8.1 of the Altera® Nios® II Embedded Design Suite (EDS). These release notes describe the revision history and errata for the Nios II EDS.

Errata are functional defects or errors, which might cause the product to deviate from published specifications. Documentation issues include errors, unclear descriptions, or omissions from current published specifications or product documents.

For the most up-to-date errata for this release, refer to the latest version of the *Nios II Embedded Design Suite Release Notes and Errata* on the Altera website.

You can refer to release notes and errata for the Nios II processor core in the *Nios II Processor* chapter of the *MegaCore IP Library Release Notes and Errata*.

Prior to v7.2, Nios II release notes and errata were organized differently. You can refer to the documents from before v7.2 at:

- http://www.altera.com/support/ip/processors/nios2/rn/ips-niosii-rn.html—Release notes
- http://www.altera.com/support/ip/processors/nios2/er/ips-niosii-er.html—Errata

Revision History

Table 1 shows the revision history for the Nios II EDS, including the Nios II C2H Compiler.

For more information on new features and changes, refer to the *Nios II Processor Reference Handbook*, the *Nios II Software Developer's Handbook*, and the *Nios II C2H Compiler User Guide*. For information about changes to the Nios II processor core, refer to the *Nios II Processor* chapter of the *MegaCore IP Library Release Notes and Errata*.

Table 1. Nios II Embedded Design Suite Revision History (Part 1 of 2)

Version	Date	Description
8.1.1	January 2009	Nios Community Wiki URL changing
8.1.0	November 2008	Maintenance release
8.0	May 2008	● The Nios II BSP editor, a standalone graphical user interface (GUI), provides an easy-to-use front end for the Nios II software build tools. ● The Altera Hardware Abstraction Layer (HAL) supports an advanced exception handler that can handle Nios II memory protection unit (MPU) exceptions. ● The Nios II Studio is available as a Beta release. The Studio is an alternative to the Nios II integrated development environment (IDE). Nios II projects created with the Nios II Studio are 100% compatible with projects created with the Nios II software build tools on the command line. ● The Nios II software build tools `--sopc` option only accepts **.sopcinfo** files as arguments and not **.sopc** files. ● The Nios II IDE command-line utilities, which form the infrastructure of the Nios II IDE, are deprecated; support will be discontinued in a future release. The deprecated tools include `nios2-create-system-library`, `nios2-create-application-project`, `nios2-build-project`, `nios2-import-project`, and `nios2-delete-project`. ● The Lightweight IP (lwIP) transmission control protocol/Internet protocol (TCP/IP) suite is removed from the Nios II EDS. ● The MicroC/OS-II real-time operating system (RTOS) is upgraded from v2.83 to v2.86. ● The NicheStack TCP/IP stack — Nios II Edition is upgraded from v3.0 to v3.1. ● Certain Nios II software examples are removed from the EDS, and are distributed on the Altera website instead. The affected designs and their new locations are listed in "Software Examples Moved to the Web" on page 4. ● The NicheStack TCP/IP Stack macro definition `INICHE_DEFAULT_IF` is deprecated. For details, see "Deprecated NicheStack Setting". ● The **nios_csh** file in **sopc_builder/bin** is removed from the Nios II EDS.
7.2 SP1	December 2007	● Maintenance release

Table 1. Nios II Embedded Design Suite Revision History (Part 2 of 2)

Version	Date	Description
7.2	October 2007	● Nios II software build tools: Library support. ● Nios II software build tools: Support for precompiled libraries (.a files) as device drivers. ● Nios II software build tools: Recompiles newlib when new compiler flags introduced. ● Nios II software build tools: command line options to specify component search paths ● Nios II GNU compiler: targets floating point hardware when performing conversion (to and from integer) and comparison operations. ● Nios II IDE: Lauterbach instruction set simulator (ISS) debug option. ● Nios II IDE: **altera.components** project shortens project creation time. ● NicheStack TCP/IP Stack – Nios II Edition: Simplified licensing mechanism. ● NicheStack TCP/IP Stack – Nios II Edition: Increased Dynamic Host Configuration Protocol (DHCP) timeout to 130 seconds. ● Triple Speed Ethernet MegaCore® function hardware design example: Includes tightly-coupled memories; other changes. Refer to the Nios II Documentation Launchpad for details. ● Nios II processor and EPCS Flash Controller: Cyclone® III boot loader support. ● Stratix® III and Cyclone III Fast hardware example design removed. Designs targeting those device families are available with Stratix III and Cyclone III development kits. ● Legacy software development kit (SDK) utilities `nios2-build` and `nios2-run` removed.
7.1	May 2007	● Nios II software build tools design flow ● Arria™ GX device family support ● Design examples including new SOPC Builder components ● Altera Triple Speed Ethernet (TSE) Media Access Controller (MAC) MegaCore function ● Altera Scatter-Gather Direct Memory Access (SG-DMA) Controller

Revision Details

This section provides details about some of the Nios II EDS revisions shown in Table 1.

Software Examples Moved to the Web

Starting in v8.0, the software examples shown in Table 2 are removed from the Nios II EDS, and instead distributed on the Web, at the locations shown in the table. This applies to software examples in Nios II IDE template form, and in the form of Nios II software build tools `create-this-app` and `create-this-bsp` scripts.

Table 2. Software Examples Removed from the Nios II EDS

Software Example	Delivery Mechanism
Read-Only Zip File System	Nios Wiki
MicroC/OS-II Message Box	Nios Wiki
Hello_LED	Nios Wiki
Host File System	Nios Wiki
Dhrystone	● Nios II Literature Page—with the *Nios II Performance Benchmark Data Sheet* ● Fast Nios II Hardware Design Example Page
MicroC/OS II Tutorial	Nios II Literature Page—with the *Using MicroC/OS-II RTOS with the Nios II Processor Tutorial*
Tightly Coupled Memory (TCM) Tutorial	Nios II Literature Page—with the *Using Nios II Tightly Coupled Memory Tutorial*
Custom Instruction Tutorial	Nios II Literature Page—with the *Nios II Custom Instruction User Guide*

Deprecated NicheStack Setting

The Nios II IDE and the Nios II software build tools contain a setting to specify a default Ethernet (MAC) interface when using the NicheStack TCP/IP stack. Starting in v8.0, this setting is deprecated. It is no longer used in Altera software application example code. The NicheStack TCP/IP stack has never depended on this setting.

This setting generates the macro `INICHE_DEFAULT_IF` in the generated **system.h** file located in the board support package (BSP). If your software depends on this macro definition, modify it, for example, to define `INICHE_DEFAULT_IF` explicitly. The macro will be removed in a future Nios II EDS release.

Altera networking example applications do not support multiple Ethernet interfaces. If you wish to use the Web Server or Simple Socket Server networking examples in a hardware system with multiple

Ethernet MAC interfaces, or with the multi-port Altera Triple Speed Ethernet MegaCore function, you need to modify the `get_ip_addr()` and `get_mac_addr()` functions to provide a unique MAC and IP address for each Ethernet interface in the system.

Errata Summary

Table 3 summarizes the issues that affect the Nios II Embedded Design Suite versions 7.1 through 8.1. For a detailed description of each issue, click on the issue name.

Table 3. Nios II EDS Errata (Part 1 of 9)

Added or Updated	Issue	Page	Affected Version					
			8.1	8.0	7.2 SP1	7.2	7.1 SP1	7.1
C2H Compiler								
Nov 2008	Error "java.lang.NoClassDefFoundError" During C2H Compilation	13	✓	—	—	—	—	—
Jul 2008	C2H Cache Coherency Issue with Global Variables	14	✓	✓	—	—	—	—
Jul 2008	Functions Declared Without a Return Type Are Not Supported	15	✓	✓	—	—	—	—
Oct 2007	Pre-7.1 Systems Are Not Supported	15	✓	✓	✓	✓	—	—
Oct 2007	--src-dir Software Build Tools Argument Does Not Work With C2H	16	✓	✓	✓	✓	—	—
May 2007	Accelerator Generation Failure If Tools Are Installed in Path With Spaces	16	✓	✓	✓	✓	✓	✓
May 2007	C2H Accelerators Time Out Immediately When Untethered	*(1)*	✓	✓	✓	✓	✓	✓
May 2007	The C2H Compiler Regenerates an Accelerator Unnecessarily	16	✓	✓	✓	✓	✓	✓
May 2007	Error: c2h_accelerator_base_addresses.h: No such file or directory	17	✓	✓	✓	✓	✓	✓
May 2007	Java Heap Space Exception if Quartus II Compilation is Enabled	17	✓	✓	✓	✓	✓	✓
May 2007	Pointer Dereferences to Volatile Types	18	✓	✓	✓	✓	✓	✓
Dec 2006	C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File	18	✓	✓	✓	✓	✓	✓
Dec 2006	C2H Compiler Out-of-Memory Error	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Array Elements in Structures Do Not Copy Correctly	19	✓	✓	✓	✓	✓	✓
Jun 2006	Build Clean Causes Build Failure	19	✓	✓	✓	✓	✓	✓

Table 3. Nios II EDS Errata (Part 2 of 9)

Added or Updated	Issue	Page	Affected Version						
			8.1	8.0	7.2 SP1	7.2	7.1 SP1	7.1	
Jun 2006	Changing Build Configurations Produces Unexpected Results	20	✓	✓	✓	✓	✓	✓	
Jun 2006	Closed System Library While Working With the C2H Compiler	*(1)*	✓	✓	✓	✓	✓	✓	
Jun 2006	Comma Operators Are Not Supported	20	—	—	—	—	—	Fixed	
Jun 2006	Hardware Accelerators Remain After Deleting the Software Project *(2)*	21	✓	✓	✓	✓	✓	✓	
Jun 2006	Incorrect Results From Logical or Conditional Operation With Side-Effects	21	✓	✓	✓	✓	✓	✓	
Jun 2006	Launch SOPC Builder Button in C2H View	22	✓	✓	✓	✓	✓	✓	
Jun 2006	Multiple Clock Domains Cause Hardware Accelerator to Fail	*(1)*	✓	✓	✓	✓	✓	✓	
Jun 2006	Simulating on ISS Not Supported	*(1)*	✓	✓	✓	✓	✓	✓	
Development Boards									
Jun 2006	Intermittent Failures While Accessing CompactFlash Card	23	✓	✓		✓	✓	✓	
Documentation Issues									
Jan 2009	Obsolete Wiki URLs in Help Systems and Handbook	24	✓	—	—	—	—	—	294739, 294744
Nov 2008	Error Message on Invalid Exception or Reset Vector	24	✓	—	—	—	—	—	288159
Jul 2008	Missing Source Control Information	25	✓	✓	—	—	—	—	
Dec 2007	Incomplete newlib Documentation	*(1)*	✓	✓	✓	✓	—	—	
Oct 2007	Nios II IDE Online Help Expand Buttons Do Not Work	27	✓	✓	✓	✓	—	—	
Oct 2007	New Control Register Names Not Listed	28	Fixed	Fixed	✓	✓	—	—	
May 2007	Arria GX Support Incorrectly Listed in Documentation HTML Page	28	—	—	—	Fixed	✓	✓	
May 2007	Nios II IDE Help Topic Does Not Work With Linux Mozilla	28	✓	✓	✓	✓	✓	✓	
Mar 2007	GNU Documentation is Missing	*(1)*	—	—	—	—	—	Fixed	
Mar 2007	Nios II Processor Reference Handbook Refers Erroneously to the Data Bus	29	—	—	—	—	—	Fixed	

Table 3. Nios II EDS Errata (Part 3 of 9)

Added or Updated	Issue	Page	Affected Version					
			8.1	8.0	7.2 SP1	7.2	7.1 SP1	7.1
Download Cables and Debug Hardware								
Jun 2006	Communication Errors During Run/Debug Sessions Using Older Download Cables	*(1)*	✓	✓	✓	✓	✓	✓
Hardware Example Designs								
Mar 2007	Incorrect PLL Phase Shift in Stratix II and Cyclone II Designs	30	✓	✓	✓	✓	✓	✓
Software Examples								
Nov 2008	Simple Socket Server Example Displays Incorrect IP Address	31	✓	—	—	—	—	—
Oct 2007	Hardware Tutorial Software Example Hangs on Some Boards	31	✓	✓	✓	✓	—	—
May 2007	Dhrystone BSP Example Fails When Run on the Fast Hardware Example	32	—	—	—	Fixed	✓	✓
May 2007	Software Template Error With Non-Ethernet Hardware Design	33	—	—	—	Fixed	✓	✓
Jun 2006	Networking Examples	34	✓	✓	✓	✓	✓	✓
Jun 2006	RAM Test Failure When Running Memory Test Software Template on the ISS	*(1)*	✓	✓	✓	✓	✓	✓
Flash Programmer								
Oct 2007	elf2flash Warning With Projects That Share a .ptf File	34	—	—	—	Fixed	✓	✓
May 2007	Delay When Creating New Flash Programmer Configuration	35	✓	✓	✓	✓	✓	✓
Dec 2006	elf2flash File Size Limit	35	✓	✓	✓	✓	✓	✓
Dec 2006	Error Message “No such file or directory” When Programming Flash	35	✓	✓	✓	✓	✓	✓
Dec 2006	Flash Programmer Generates Error with Override File	36	—	—	—	—	—	Fixed
Dec 2006	Spurious Flash Verify Failure With nios2-flash-programmer	36	—	—	—	—	—	Fixed
Dec 2006	Flash Programmer Error with Multiprocessor System	*(1)*	✓	✓	✓	✓	✓	✓
Hardware Simulation								
Dec 2007	Nios II Processor Fails in Simulation	*(1)*	—	—	Fixed	✓	✓	—

275067

Table 3. Nios II EDS Errata (Part 4 of 9)

Added or Updated	Issue	Page	Affected Version					
			8.1	8.0	7.2 SP1	7.2	7.1 SP1	7.1
Oct 2007	bash: vsim: command not found	37	Fixed	Fixed	✓	✓	—	—
Oct 2007	Simulation Fails Due to Incorrect Port Properties	37	Fixed	Fixed	✓	✓	✓	✓
Dec 2006	Error "UNC paths are not supported" Launching ModelSim	38	✓	✓	✓	✓	✓	✓
Jun 2006	ModelSim Fails to Load Large Memory Models	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Simulation Failure if Reset Address is Set to EPCS	38	—	—	—	Fixed	✓	✓
Jun 2006	Uninitialized .bss Variables in Simulation	38	✓	✓	✓	✓	✓	✓
Host Platform								
Dec 2007	Windows/Cygwin: Nios II Processor Generation Failure	39	✓	✓	✓	✓	—	—
Oct 2007	Windows Vista: Limited Support in Nios II EDS	40	✓	✓	✓	✓	—	—
Oct 2007	Linux: Nios II IDE Fails to Find File	40	Fixed	Fixed	✓	✓	—	—
Dec 2006	Windows XP: Internal Error When Repeatedly Stepping Into With Debugger	41	—	—	—	—	—	Fixed
Jun 2006	Linux: Process Fails To Terminate While Debugging With the Nios II ISS Target	41	✓	✓	✓	✓	✓	✓
Jun 2006	Linux: The Quartus® II Stand-Alone Programmer is Not Supported	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Windows: Frisk Antivirus Software Causes SOPC Builder and Nios II Command Shell to Be Unresponsive	*(1)*	✓	✓	✓	✓	✓	✓
Nios II IDE								
Building Projects								
Nov 2008	Nios II IDE Cannot Find stdio.h in Outline View	42	✓	✓	✓	✓	—	—
Jul 2008	Nios II IDE Hangs With UNC Project Path	42	✓	✓	—	—	—	—
Jul 2008	Build Failure with Nios II Advanced Exceptions, MMU, and MPU	43	✓	✓	—	—	—	—
Dec 2007	First Build in the Nios II IDE Takes Several Minutes	43	—	—	Fixed	✓	—	—
Dec 2007	Nios II IDE Command-Line Tools Hang	44	✓	✓	✓	✓	✓	✓
Oct 2007	Read-Only Zip File System Base Address Fixed at 0	44	—	—	—	Fixed	✓	✓

263073

Table 3. Nios II EDS Errata (Part 5 of 9)

Added or Updated	Issue	Page	Affected Version					
			8.1	8.0	7.2 SP1	7.2	7.1 SP1	7.1
May 2007	Build Command Not Functional for BSPs Created Using the Software Build Flow	45	✓	✓	✓	✓	✓	✓
May 2007	Linker Errors with Dual-Port Memories	45	✓	✓	✓	✓	✓	✓
May 2007	User-Managed BSP Settings Not Supported in Nios II IDE	45	✓	✓	✓	✓	✓	✓
May 2007	Error Message: 'Internal error occurred during "Launching"'	46	—	—	—	—	✓	✓
May 2007	URL Project Location Causes Project Creation Error	46	✓	✓	✓	✓	✓	✓
Mar 2007	Incorrect Stack Checking When Interrupt Vector Custom Instruction Is Used	47	—	—	—	—	—	Fixed
Mar 2007	Unable to Save Files Containing Copyright Symbol	47	—	—	—	—	—	Fixed
Dec 2006	Compilation Error with Separate Exception Stack Option	48	✓	✓	✓	✓	✓	✓
Dec 2006	Illegal Project Location Error When Creating New Project	*(1)*	✓	✓	✓	✓	✓	✓
Dec 2006	Incorrect Address Assignment for Dual-Port Memory Mastered by Two Different Master Ports	*(1)*	✓	✓	✓	✓	✓	✓
Dec 2006	Incorrect Stack and Heap Free Space Report	48	✓	✓	✓	✓	✓	✓
Dec 2006	Nios II IDE Reports Problems Without Displaying Error in Console	48	✓	✓	✓	✓	✓	✓
Jun 2006	Build Errors After Changing Component Names in SOPC Builder	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Nios II IDE Unnecessarily Updates the SOPC Builder System File (.ptf)	*(1)*	✓	✓	✓	✓	✓	✓
	Debugging Projects							
Nov 2008	IDE Cannot Display Imported Profiling Data	49	✓	—	—	—	—	—
Jul 2008	Error Message when Downloading .elf File	49	✓	✓	—	—	—	—
Jul 2008	Variable Casting Unsupported in ISS	50	✓	✓	—	—	—	—
Jul 2008	Trace Debug Does Not Support Instruction-Related Exceptions	50	✓	✓	—	—	—	—
Jul 2008	Error Accessing MMU and MPU Registers in the Debugger	50	✓	✓	—	—	—	—
Oct 2007	Trace Debug Does Not Support the JMPI Instruction	51	✓	✓	✓	✓	—	—

271045

Table 3. Nios II EDS Errata (Part 6 of 9)

Added or Updated	Issue	Page	Affected Version					
			8.1	8.0	7.2 SP1	7.2	7.1 SP1	7.1
May 2007	Cannot Locate Source Code in Driver Files Shared by Multiple Projects	51	✓	✓	✓	✓	✓	✓
May 2007	Console Window Is Not Updated After ISS Error	51	✓	✓	✓	✓	✓	✓
May 2007	Extra Arguments in Debugger for Multiple Launch Configurations	52	—	—	—	Fixed	✓	✓
May 2007	GDB Error 256 When Launching Debugger	52	—	—	—	Fixed	✓	✓
May 2007	ISS Fails on Designs Containing TSE MAC or SG-DMA Components	53	✓	✓	✓	✓	✓	✓
May 2007	Memory Window Sets Control Register Values Incorrectly	53	✓	✓	✓	✓	✓	✓
May 2007	Previous 7.1 Beta Install Might Cause Internal Error When Launching FS2 in IDE	*(1)*	✓	✓	✓	✓	✓	✓
May 2007	Nios II IDE Debugger Cannot Display Driver Source Code for Custom Component	54	—	—	—	Fixed	✓	✓
Dec 2006	“Additional nios2-terminal Arguments” Field Is Not Functional	55	—	—	—	—	—	Fixed
Dec 2006	MicroC/OS-II “Source Not Found” Error Occurs at Breakpoint	55	—	—	—	—	—	Fixed
Dec 2006	Programs That Interact With a Terminal Console on Windows Do Not Work	56	✓	✓	✓	✓	✓	✓
Dec 2006	“Run as ModelSim” in the Nios II IDE Fails	57	✓	✓	✓	✓	✓	✓
Dec 2006	The Restart Command on the Run Menu Does Not Work	57	✓	✓	✓	✓	✓	✓
Jun 2006	Breakpoints on Adjacent Lines of Assembly Fail to Halt the Processor	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Cannot Use Watchpoints in the Nios II IDE When the FS2 Console Is Open	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Debugger Cannot Step Into __sflags(), and Continues Execution Instead	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Incorrect Breakpoint Filtering on Threads	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Missing Address and Data in the Trace View	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	nios2-gdb-server Fails to Terminate After Setting a Watch Point	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	“Step failed. Target is not responding (timed out)” Error Message	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Uninitialized Memory Error When Executing From ISS	*(1)*	✓	✓	✓	✓	✓	✓

Table 3. Nios II EDS Errata (Part 7 of 9)

Added or Updated	Issue	Page	Affected Version						
			8.1	8.0	7.2 SP1	7.2	7.1 SP1	7.1	
Jun 2006	Watchpoints Do Not Work on Certain Variables	57	✓	✓	✓	✓	✓	✓	
Navigating Projects									
May 2007	C/C++ Search Returns Nothing	58	—	—	—	Fixed	✓	✓	
May 2007	Nios II IDE Import System Library Dialog Box Does Not Close After Import	58	Fixed	Fixed	✓	✓	✓	✓	
Dec 2006	Nios II IDE Freezes While Displaying the Splash Screen	58	✓	✓	✓	✓	✓	✓	
Dec 2006	Internal Error When Double-Clicking on a Large Objdump File	59	✓	✓	✓	✓	✓	✓	
Jun 2006	C/C++ Scanner Does Not Support Certain C/C++ Constructs	59	✓	✓	✓	✓	✓	✓	
Jun 2006	Nios II IDE Does Not Recognize When an Object File is Built Outside the IDE	60	—	—	—	—	—	Fixed	
Nios II Studio									
Nov 2008	Error in nios2-studio Help System	60	✓	—	—	—	—	—	287087
Jul 2008	No System ID Validation Performed	61	✓	✓	—	—	—	—	
Jul 2008	Spurious Messages When Importing Pre-8.0 Projects	61	✓	✓	—	—	—	—	
Jul 2008	public.mk Error or System Failure After Hardware Modification	62	✓	✓	—	—	—	—	
Nios II Software Build Tools									
Nov 2008	Incorrect Linker Errors	62	✓	✓	—	—	—	—	278716
Nov 2008	Java Heap Space Error When Creating BSP	63	✓	—	—	—	—	—	284272
Jul 2008	BSP Editor Incorrectly Allows Mixed Drivers	63	✓	✓	—	—	—	—	
Jul 2008	nios2-bsp --help Displays Incorrect List of BSP Types	64	✓	✓	—	—	—	—	
Jul 2008	nios2-bsp-create-settings Fails to List BSP Types	65	✓	✓	—	—	—	—	
Jul 2008	Error Creating BSP for Pre-8.0 System	65	✓	✓	—	—	—	—	
Jul 2008	Makefile Error with Absolute Source Path	65	✓	✓	—	—	—	—	
Dec 2007	Space in --set Option Value Causes Error	66	—	—	Fixed	✓	—	—	
Oct 2007	Software Build Tools Fail if Nios II EDS is Installed in a Path Containing Spaces	66	✓	✓	✓	✓	—	—	
Oct 2007	JTAG Instance Conflict for Multiprocessor Systems Imported Into IDE	67	Fixed	Fixed	✓	✓	—	—	

Table 3. Nios II EDS Errata (Part 8 of 9)

Added or Updated	Issue	Page	Affected Version					
			8.1	8.0	7.2 SP1	7.2	7.1 SP1	7.1
Oct 2007	system.h File Does Not Reflect Pre-7.1 Custom Instruction Name Changes	67	Fixed	Fixed	✓	✓	—	—
May 2007	"make download-elf" Fails for Stratix II 2S60 RoHS Design Examples	68	—	—	—	Fixed	✓	✓
May 2007	User-Managed BSP Settings Not Supported in Nios II IDE	45	✓	✓	✓	✓	✓	✓
May 2007	Improper Quote Format in system.h	69	—	—	—	Fixed	✓	✓
May 2007	Upper-Case File Extensions Not Supported	69	✓	✓	✓	✓	✓	✓
Peripherals								
Jul 2008	DMA Transfers Truncated or Stalled	*(1)*	✓	✓	✓	✓	✓	✓
Jul 2008	Unaligned Transfers of Small Payloads Fail on SG-DMA	70	✓	✓	—	—	—	—
Jul 2008	IDT71V416 SRAM Component HardCopy Incompatibility	70	✓	✓	—	—	—	—
Jul 2008	Software Build Tools Fail With UART Driver	71	✓	✓	—	—	—	—
Dec 2007	Flash Memory (CFI) Component Has Incorrect Default Timing Parameters	71	Fixed	Fixed	✓	—	—	—
Oct 2007	Software Build Errors Due to _classic Suffix	72	—	—	—	Fixed	✓	✓
Oct 2007	Software Build Tools Select Inappropriate Memories	73	—	—	—	Fixed	✓	✓
May 2007	DMA Controller Always Busy in Burst Mode	73	✓	✓	✓	✓	✓	✓
May 2007	Non System Wide Reset Can Cause Improper Initialization of Mailbox Core	73	✓	✓	✓	✓	✓	✓
May 2007	Triple Speed Ethernet Designs Require a Manual Reset on Subsequent Software Downloads	74	—	—	—	Fixed	✓	✓
Jun 2006	Using the DEV_CLRn Signal Can Cause JTAG UART Instability After Device-Wide Reset	*(1)*	✓	✓	✓	✓	✓	✓
Target Software								
Jul 2008	Degraded TCP/IP Performance with NicheStack TCP/IP stack and TSE MAC	*(1)*	✓	✓	✓	✓	—	—
Jul 2008	Slow TSE MAC Performance with NicheStack TCP/IP Stack	*(1)*	✓	✓	—	—	—	—
Jul 2008	Default IP Address Assigned Incorrectly with NicheStack TCP/IP Stack	*(1)*	✓	✓	—	—	—	—
Oct 2007	System Clock Rollover	74	—	—	—	Fixed	✓	✓

Table 3. Nios II EDS Errata (Part 9 of 9)

Added or Updated	Issue	Page	Affected Version					
			8.1	8.0	7.2 SP1	7.2	7.1 SP1	7.1
Dec 2006	Compiling an Existing MicroC/OS-II Application Generates Errors	75	—	—	—	—	—	—
Dec 2006	Lightweight IP (lwIP) Failure When Using DHCP	*(1)*	✓	✓	✓	✓	✓	✓
Dec 2006	stdio Does Not Work with MicroC/OS-II and Small C Library	75	✓	✓	✓	✓	✓	✓
Jun 2006	cout From MicroC-OS/II Task Does Not Send Data to stdout	76	✓	✓	✓	✓	✓	✓
Jun 2006	Creating New Custom HAL Components in the Nios II IDE	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	malloc(), realloc() Failures With MicroC/OS-II	76	✓	✓	✓	✓	✓	✓
Jun 2006	Problems Using HAL Drivers With Toshiba Flash	*(1)*	✓	✓	✓	✓	✓	✓
Toolchain (gcc, gdb, etc.)								
Jul 2008	sof2flash Names Output Files Incorrectly	77	✓	✓	—	—	—	—
Dec 2006	GNU Assembler Does Not Accept the --defsym Flag	77	✓	✓	✓	✓	✓	✓
Jun 2006	Breakpoints in C++ Constructors Fail to Halt the Processor	*(1)*	✓	✓	✓	✓	✓	✓

Notes to Table 3:

(1) This issue is documented in Altera's online Knowledge Database. To see the solution, click on the name of the issue, or search the Knowledge Database at **www.altera.com/support/kdb/kdb-index.jsp**.

(2) For further information, refer to the *Nios II C2H Compiler User Guide*.

C2H Compiler Errata

This section describes in detail the issues related to the Nios II C2H Compiler.

Error "java.lang.NoClassDefFoundError" During C2H Compilation

C2H compilation fails with the following error:

```
java.lang.NoClassDefFoundError: com/altera/c2h/C2Hgenerate
Exception in thread "main" make: *** [c2h_hdl-t] Error 1
```

This error occurs when you invoke C2H from the Nios II IDE on the Windows operating system. This problem does not affect Linux installations, nor C2H compilation from the Nios II command shell.

Workaround

To correct the problem, visit mysupport.altera.com/eservice and request the Nios2 EDS 8.1 Patch 0.01 (PC version).

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

C2H Cache Coherency Issue with Global Variables

Cache coherency issues might be encountered when the following conditions are all true:

- A C2H-accelerated function writes to a global or static variable
- The Nios II processor is configured with a data cache
- The referenced global variable is located in cacheable memory

In this situation, the data in the global variable can be corrupted.

This issue can only arise if the accelerated function writes to the global or static variable. If the function merely reads from the variable, cache coherency is maintained correctly.

Workaround

Add a data cache flush operation in the accelerator's wrapper function. The wrapper can be found in **alt_c2h_<*function_name*>.c**. In this file, find the following comment:

```
/* Write 1 to address 0 starts the accelerator */
```

Directly after this comment, add the following function call:

```
alt_dcache_flush_all();
```

This function call ensures that the data cache is flushed immediately before the accelerator is started.

☞ Flushing the cache entails a performance penalty.

Alternatively, allocate your global and static variables in non-cacheable regions of memory. Placing these variables in non-cacheable regions eliminates any need to perform a data cache flush.

For information about allocating variables to non-cacheable regions, refer to the *Cache and Tightly-Coupled Memory* chapter of the *Nios II Software Developer's Handbook*.

Solution status

Not fixed

Functions Declared Without a Return Type Are Not Supported

The C2H compiler does not support functions without an explicitly declared return type.

Workaround

If you are using the implicit `int` return type, declare the return type explicitly. If your function has no return value, declare it as `void`.

Solution status

Not fixed

Pre-7.1 Systems Are Not Supported

Starting in v. 7.1, the C2H Compiler does not support SOPC Builder systems created in SOPC Builder 7.0 or earlier (systems based on a **class.ptf** file).

Workaround

Before using the C2H Compiler, you must upgrade your SOPC Builder system. Open the system in SOPC Builder v. 7.1 or later. SOPC Builder prompts you to convert the system, creating a ***_hw.tcl** file compatible with the C2H Compiler.

Solution Status

Not fixed

--src-dir Software Build Tools Argument Does Not Work With C2H

When using the C2H Compiler with Nios II Software Build Tools, an error occurs when using the `nios2-app-generate-makefile` argument `--src-dir` to specify a directory of source files. An example of the error is:

```
multiple definition of 'my_accelerated_function'
```

Workaround

Specify source files individually using the `--src-files` argument.

Solution Status

Not fixed

Accelerator Generation Failure If Tools Are Installed in Path With Spaces

If the path to your installation of the Altera Design Suite contains spaces, the C2H Compiler fails to generate the accelerators.

Workaround

Reinstall the tools to a path containing no spaces.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

The C2H Compiler Regenerates an Accelerator Unnecessarily

The C2H Compiler might regenerate an accelerator even when the accelerated function is unchanged. This problem can result from changes to files included by the C file containing the accelerated function. The C2H Compiler fails to check that the generated hardware description language (HDL) matches the previously generated HDL causing the system to be regenerated.

Workaround

To avoid this issue move the information in the include file specific to the hardware accelerator into a separate include file. This workaround prevents regeneration of the system when the HDL is unchanged.

Solution Status

Not fixed

Error: c2h_accelerator_base_addresses.h: No such file or directory

When a C2H accelerator is compiled for the first time, the following compile-time error can result if the **Analyze all accelerators** option is selected:

```
c2h_accelerator_base_addresses.h: No such file or directory.
```

Workaround

Click **Build software and generate SOPC Builder system** and build once before building with the **Analyze all accelerators** option.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Java Heap Space Exception if Quartus II Compilation is Enabled

If your design contains a C2H accelerator, and you select **Build software, generate SOPC Builder system, and run Quartus II compilation** in the C2H view, you might see the following error during Quartus II compilation:

```
    Exception in thread "main"
    java.lang.OutOfMemoryError: Java heap space

    make: *** [c2h_hdl-t] Error 1
```

Workaround

Select the **Build software and generate SOPC builder system** option in the C2H Compiler settings window, and then manually launch the Quartus II software to compile the design.

Solution Status

Not fixed

Pointer Dereferences to Volatile Types

The C2H Compiler treats pointer dereferences to a volatile type as if they alias all other pointer dereferences. Pointers that are restrict-qualified are treated the same way.

The two loops in Example 1 cannot be scheduled concurrently because the volatile qualification overrides the __restrict__ pragma.

Example 1. Non-Concurrent Loops

```
volatile int * __restrict__ fifo_rd = FIFO_RD_BASE;
volatile int * __restrict__ fifo_wr = FIFO_WR_BASE;
for ()
{
 *fifo_wr = ....;
}
for ()
{
 ... = *fifo_rd;
}
```

Workaround

Divide the function into multiple IRQ-enabled accelerators that are launched concurrently from the processor, and use FIFOs to communicate between them.

Solution Status

Not fixed

C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File

When accelerating a function in a file, the C2H Compiler cannot link subfunctions that are defined in a different file.

Workaround

Include all subfunctions called by the accelerated function within the same source code file.

Solution Status

Not fixed

Array Elements in Structures Do Not Copy Correctly

C2H accelerators do not correctly copy array elements that are elements of structures.

In Example 2, the `a` and `b` elements of the structure copy correctly, but the `buf` element does not. After this assignment, `struct_a` equals `{9, 8, {3, 3, 3, 3}}`.

Example 2. Array Elements of Structs

```
typedef struct my_struct {
  int a;
  int b;
  int buf[BUF_SIZE];
}MY_STRUCT;
MY_STRUCT struct_a = {1, 2, {3, 3, 3, 3}};
MY_STRUCT struct_b = {9, 8, {7, 7, 7, 7}};
struct_a = struct_b;
```

Workaround

Copy the array elements explicitly, as shown in Example 3.

Example 3. Copying Array Elements Explicitly

```
{
  int i=0;
  do
    {
      struct_a.buf[i] = struct_b.buf[i];
      i++;
    } while (i<LENGTH_OF_BUF_ELEMENT)
}
```

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Build Clean Causes Build Failure

Performing a clean build on a Nios II IDE project that contains a hardware accelerator can cause the next build to fail in the IDE, because the clean build erroneously deletes a file required by the C2H Compiler.

Workaround

Do not perform a clean build on projects that use hardware accelerators. If you have already performed a clean build, recompile with option **Build software, generate SOPC Builder system, and run Quartus II compilation** to regenerate the necessary files.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Changing Build Configurations Produces Unexpected Results

The C2H Compiler does not support multiple build configurations (for example Release or Debug) in the Nios II IDE. After creating one or more accelerators in a particular configuration, the C2H Compiler produces undefined results if you switch to a different build configurations and create more accelerators.

Workaround

For a specific SOPC Builder system and Nios II IDE project, specify C2H accelerators in only one build configuration. You can use multiple build configurations, as long as only one configuration specifies C2H Compiler settings.

Solution Status

Not fixed

Comma Operators Are Not Supported

The C2H Compiler does not support comma operators, such as in Example 4.

Example 4. Comma Operators

```
for(i = 0, j = 3; i < 10; i++, j++)
{
  /* statements */
}
```

Workaround

You can code the same functionality in separate statements, as shown in Example 5.

Example 5. Simulated Comma Operators

```
j = 3;
for(i = 0; i < 10; i++)
{
  /* statements */
  j++;
}
```

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in release 7.1.

Hardware Accelerators Remain After Deleting the Software Project

If a system contains C2H accelerators, deleting the software project that defines the accelerators does not remove the accelerators from the hardware system, and the accelerator logic remains in the SOPC Builder system.

Workaround

To remove an accelerator from a system, delete the accelerator from the C2H view in the Nios II IDE first, and then recompile the software project. The C2H Compiler then removes the accelerator from the SOPC Builder system. Once the compilation is complete then the software application can be deleted from the workspace.

Solution Status

Not fixed

Incorrect Results From Logical or Conditional Operation With Side-Effects

The C2H Compiler always evaluates both operands of logical (`&&`, `||`) and conditional (`?:`) operators. This is different from expected American National Standards Institute (ANSI) C behavior, where operands are evaluated left-to-right, and unnecessary operands are skipped.

For example, in the expression `(i-- && j--)`, if the value of `i` is zero, ANSI C does not evaluate the right-hand-side (RHS) expression, and `j` is not decremented. By contrast, the C2H Compiler evaluates both sides, decrementing `j`.

The following expressions are other examples that might be affected by this issue:

```
if (i -- || j++)

a = ((cond == 1)?  i++ : j++);
```

Workaround

Use logical and conditional operations whose operators have no side effects. Operations with side effects include pre- and post- increment and decrement operations (`++`, `--`), memory operations (`*`, `[]`, `.`, `->`), and function calls.

Solution Status

Not fixed

Launch SOPC Builder Button in C2H View

When the Nios II IDE workspace contains multiple projects with multiple system libraries, the incorrect SOPC Builder system might open when you click **Launch SOPC Builder** in the C2H view.

Workaround

Launch SOPC Builder from the Quartus II software. Alternatively, keep only one system library project open at a time while using the C2H Compiler.

Solution Status

Not fixed

Development Board Errata

This section describes in detail the Nios II EDS issues related to Nios development boards.

Intermittent Failures While Accessing CompactFlash Card

The Nios II Development Kit version 5.0 and later includes a CompactFlash controller peripheral suitable for interfacing to CompactFlash cards in True IDE mode on Nios development boards. For True IDE mode to operate, CompactFlash cards require that the ATASEL_N input be driven to ground during power-up.

The CompactFlash controller peripheral includes a configurable power register, used to cycle power to CompactFlash cards from Nios II software through a metal oxide semiconductor field-effect transistor (MOSFET) on the Nios development boards. However, in certain development boards, power to the CompactFlash card does not turn off completely during this power cycle operation. Because of this condition, the CompactFlash might not sample the ATASEL_N pin during the power-cycle operation after FPGA configuration when this pin is driven to ground. Instead, the CompactFlash card might sample the ATASEL_N pin when power is first applied to the development board, when I/O is not yet driven by the FPGA (before FPGA configuration).

Workaround

If you encounter errors with CompactFlash when using the Nios development boards, try one of the following solutions:

- Use a different CompactFlash card. Certain cards are more susceptible to the power-cycling issue than others.
- Modify the Nios development board. This is recommended if you are familiar and comfortable with board-level modifications. Disconnect pin 9 (ATASEL_N) on the CompactFlash socket on your Nios development board and tie this pin to ground.

☞ The CompactFlash socket uses a staggered numbering on the pins (starting from pin 1: 1, 26, 2, 27, ...); refer to the CompactFlash Association specification for right-angle surface-mount connectors for exact specifications on this connector. This modification permanently enables True IDE mode operation.

Solution Status

Not fixed

Documentation Errata

This section describes in detail the Nios II EDS documentation issues.

Obsolete Wiki URLs in Help Systems and Handbook

The following documents contain obsolete URLs for the Nios Community Wiki:

- The *Nios II Software Developer's Handbook*
- The *Nios II Embedded Design Handbook*
- The Nios II IDE help system
- The Nios II Studio help system

Workaround

You can visit the Nios Community Wiki at www.nioswiki.com.

Solution Status

Not fixed

Error Message on Invalid Exception or Reset Vector

New Nios II compiler error messages, added in v 8.1, are not documented in the *Nios II Software Developer's Handbook: Studio Edition*.

Workaround

The missing information is as follows:

While building a Nios II software project, the compiler produces an error message if the SOPC Builder system defines an invalid exception or reset vector. Table 2–1 shows possible error messages and their causes.

Table 2–1. Exception and Reset Vector Error Messages (Part 1 of 2)

Message	Cause
`The section mapping ".entry" is not mapped to a memory region.`	The `.entry` section is mapped to a nonexistent memory region.
`The section mapping ".exceptions" is not mapped to a memory region.`	The `.exceptions` section is mapped to a nonexistent memory region.
`The section mapping ".entry" maps to a memory region "`*<region name>*`" which is not the required length "32".`	The `.entry` section is mapped to a memory region that is smaller than 32 bytes.

Table 2–1. Exception and Reset Vector Error Messages (Part 2 of 2)

Message	Cause
`The section mapping ".entry" maps to a memory region "`*<region name>*`" which does not start at the SOPC Design specified address "`*<base address>*`".`	The `.entry` section is mapped to a memory region that does not start at the reset vector base address.
`The section mapping ".exceptions" maps to a memory region "`*<region name>*`" which does not start at the SOPC Design specified address "`*<base address>*`".`	The `.exceptions` section is mapped to a memory region that does not start at the exception vector base address.
`The SOPC design Nios II reset and exception vector addresses are mapped to the same memory. For HAL the exception address must be at least 32 bytes larger than the reset address.`	The `.entry` section and `.exceptions` section map to the same memory device, and the base of the `.exceptions` section is less than 32 bytes above the base of the `.entry` section.
`The section mapping ".exceptions" mapped device "`*<memory 1>*`" is not the same as the SOPC Design specified device "`*<memory 2>*`".`	The `.exceptions` section is mapped to a device that is different from the exception vector device specified in the SOPC Builder system.
`The section mapping ".entry" mapped device "`*<memory 1>*`" is not the same as the SOPC Design specified device "`*<memory 2>*`".`	The `.entry` section is mapped to a device that is different from the reset vector device specified in the SOPC Builder system.

In the SOPC Builder system, ensure that the exception and reset vectors conform to all the criteria listed in Table 2–1.

Solution Status

This issue will be fixed in a future release of Nios II EDS documentation.

Missing Source Control Information

The Quartus II Handbook refers to the *Using the Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook* for information about putting Nios II projects in source control. This information does not appear in the referenced chapter.

Workaround

The missing information follows.

Archiving Nios II IDE Software Projects

This section helps you identify the files you must include when archiving a Nios II IDE software project. With this information, you can archive a Nios II application project and its associated Nios II system library project.

You might want to archive your projects for one of the following reasons:

- To place them under source control
- To create backups
- To bundle the projects for transfer to another location

This section covers the following information:

- How to find and identify the files that you must include in an archived Nios II IDE software project.
- Which files must have write permission to allow the software projects to be built.

Table 3 and Table 4 list the files required by Nios II IDE software projects. This is the minimum set of files needed to completely rebuild a software project, including the executable and linking file (**.elf**).

Archive your Nios II IDE software projects together with the SOPC Builder system on which they are based. You cannot rebuild a Nios II IDE software project without its associated SOPC Builder system.

The files listed in Table 3 are located in the Nios II application project directory.

Table 3. Files Required for a Nios II Application Project

File Description	File Name	Write Permission Required?
All source files	for example: **app.c**, **header.h**, **assembly.s**, **lookuptable.dat**	No
Eclipse project file	**.project**	No
C/C++ Development Toolkit project file	**.cdtproject**	Yes
C/C++ Development Toolkit option file	**.cdtbuild**	No
Software configuration file	**application.stf**	No

The files listed in Table 4 are located in the Nios II system library project directory.

Table 4. Files Required for a Nios II System Library Project

File Description	File Dame	Write Permission Required?
Eclipse project file	**.project**	Yes
C/C++ Development Toolkit project file	**.cdtproject**	Yes
C/C++ Development Toolkit option file	**.cdtbuild**	No
System software configuration file	**system.stf**	Yes

☞ You must have write permission for certain files. The tools write to these files as part of the build process. If the files are not writable, the toolchain fails.

Many source control tools mark local files read-only by default. In this case, you must override this behavior. You do not have to check the files out of source control unless you are modifying the Nios II software project.

Solution status

Not fixed

Nios II IDE Online Help Expand Buttons Do Not Work

The expand text arrow and the **Show All** option in the online help do not work in Internet Explorer.

Workaround

In Internet Explorer, carry out the following actions:

1. Click the **Refresh/Show Current Topic** icon (yellow arrows icon at the top of the search pane). The table of contents appears with the selected topic highlighted.

2. Click the highlighted topic to refresh the browser frame. Expanded text works.

Alternatively, use Firefox 2.0.0.6.

Solution Status

Not fixed

New Control Register Names Not Listed

New control register names, added to the assembler in v. 7.2, are not documented in the *Nios II Processor Reference Handbook*.

Workaround

The new control register names are shown in Table 5.

Table 5. New Control Registers

Control Register	Name
8	`pteaddr`
9	`tlbacc`
10	`tlbmisc`

Solution Status

Fixed in v. 8.0.

Arria GX Support Incorrectly Listed in Documentation HTML Page

In the Nios II 7.1 Documentation Launchpad, the **Example Designs** page incorrectly states that a fast design exists targeting the Arria GX family. Although v. 7.1 and later of the Nios II processor supports the Arria GX family, it does not include a fast design for Arria GX devices.

Workaround

Use an existing fast design targeting a Stratix II device and change the device settings.

For information about device settings, refer to the “Device Page” topic in the Quartus II software online help.

Solution Status

Fixed in v. 7.2.

Nios II IDE Help Topic Does Not Work With Linux Mozilla

The expandable Help text for the **Nios II C/C++ Application** (About Nios II IDE Projects) does not function correctly with Mozilla 1.4.3 under Linux.

Workaround

The missing information is as follows:

> A Nios II C/C++ application project contains a C/C++ program, usually including a project's `main()` function. Building a Nios II C/C++ application project results in an executable linkable file (**.elf**) that you can run on target hardware, the Nios II ISS, and the ModelSim® hardware simulator. A Nios II C/C++ application project depends on a single system library project and might reference functions in a Nios II C/C++ library project.
>
> A Nios II C/C++ application project is a Nios II IDE managed make project. The Nios II IDE creates the necessary makefiles and manages the project for you.

Refer to **www.altera.com/literature/ug/ug_nios2_ide_help.pdf** for complete text of the Nios II help system.

Solution Status

Not fixed

Nios II Processor Reference Handbook Refers Erroneously to the Data Bus

In the *Instantiating the Nios II Processor in SOPC Builder* chapter of the *Nios II Processor Reference Handbook*, under “Caches and Tightly Coupled Memories Tab”, the following text appears under “Instruction Settings”: “Usually you enable bursts on the processor's data bus when processor data is stored in DRAM, and disable bursts when processor data is stored in SRAM.” This sentence should read “Usually you enable bursts on the processor's instruction master port when instructions are stored in DRAM, and disable bursts when instructions are stored in SRAM.”

Solution Status

Fixed in v. 7.1.

Hardware Example Design Errata

This section describes in detail the Nios II EDS hardware example design issues.

Incorrect PLL Phase Shift in Stratix II and Cyclone II Designs

In several example designs targeting the Nios II Cyclone II 2C35 and Nios II Stratix II 2S60 RoHS development boards, the synchronous static random access memory (SSRAM) and legacy dual data rate synchronous dynamic random access memory (DDR SDRAM) clocks are implemented with an incorrect phase-locked loop (PLL) phase shift.

This issue affects the following example designs:

- 2s60 RoHS TSE SG-DMA
- 2s60 RoHS Standard
- 2s60 RoHS Full Featured
- 2c35 TSE SG-DMA
- 2c35 Standard
- 2c35 Full Featured

This issue also affects any custom 2c35 or 2s60 RoHS design using SSRAM or legacy DDR SDRAM.

The incorrect phase shift might cause timing violations when reading from or writing to SSRAM or legacy DDR SDRAM.

Workaround

To correct the PLL settings, perform the following steps:

1. Open the standard design in SOPC Builder.

2. Double-click the component instance named `pll` to launch the PLL MegaWizard interface.

3. Click **Launch Altera's ALTPLL MegaWizard** to launch the MegaWizard interface.

4. Select the **Output Clocks** page and then the clock page used to generate the SSRAM clock. Depending on the particular design, this might be **clk c1**, **clk c2**, or another clock.

5. Ensure that **Clock phase shift** is –3.38 ns.

6. Click the clock page used to generate the write clock for the legacy DDR SDRAM, if any. Depending on the particular design, this might be **clk c1**, **clk c2**, or another clock.

7. Ensure that **Clock phase shift** is 270 degrees.

8. Click **Finish** to save changes and exit the PLL MegaWizard interface.

9. Click **Finish** to save settings to the PLL instance in SOPC Builder.

10. Regenerate the system in SOPC Builder and recompile in the Quartus II software. Refer to the **readme.txt** file in the design example folder for information on compiling designs containing the DDR controller.

For further information refer to **ssram_interface_readme.html**, located in the <*Quartus II installation directory*>**/sopc_builder/documents** folder. This document discusses the SSRAM timing analysis methodology in detail. Additionally, *AN 411: Understanding PLL Timing for Stratix II Devices* discusses clock phase shift calculations and assignments for PLLs in Stratix II devices.

Solution Status

Not fixed

Software Example Errata

This section describes in detail the Nios II EDS software example issues.

Simple Socket Server Example Displays Incorrect IP Address

The simple socket server networking software example displays an incorrect IP address for the incoming connection.

Workaround

There is currently no workaround for this problem. Ignore the IP address of any incoming connection that is displayed on the console.

Solution Status

Not fixed

Hardware Tutorial Software Example Hangs on Some Boards

The **count binary** application used in the tutorial might hang when run on some designs that do not contain a liquid crystal display (LCD) peripheral.

Workaround

Modify line 18 of the **count_binary.h** file, as follows:

```
# define LCD_PRINTF(lcd, args...) /* Do Nothing */
```

Rebuild and run the software again.

Solution Status

Not fixed

Dhrystone BSP Example Fails When Run on the Fast Hardware Example

You might see the following output when running the Dhrystone BSP example in the Nios II IDE after creating and building it via command line.

```
  Int_Comp:          36608
        should be:   18
  Str_Comp:          ·"
nios2-terminal: exiting due to ^D on remote
```

This error is caused by settings in the `create-this-bsp` script that cause the application to overrun the amount of memory available in the Fast design.

Workaround

To run the Dhrystone example successfully with the Software Build Tools, edit the `create-this-bsp` script located in the *<hardware example design>***/software_examples/bsp/hal_dhrystone** folder. Carry out the following steps:

1. If the BSP already exists, remove the **settings.bsp** file and run `make clean` in the BSP folder.

2. Change the following line:

```
--set hal.enable_lightweight_device_driver_api true \
```

to:

```
--set hal.enable_reduced_device_drivers true \
```

3. Run the `create-this-bsp` script. (Alternatively, run the `create-this-app` script, found in the Dhrystone application folder.)

The BSP is regenerated with the new setting.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Software Template Error With Non-Ethernet Hardware Design

You might get an internal error if your hardware design does not contain an Ethernet MAC, and you try to create a software example from one of the following templates:

- host_filesystem
- simple_socket_server
- web_server
- zip_filesystem

These templates check for components in the system library and incorrectly test for the existence of a MAC.

Workaround

Create the system library project for your hardware design before creating your application project from one of the templates. In the system library properties, turn on the Host File System feature by carrying out the following steps:

1. Right-click the system library project and select **Properties**.

2. Click the **Software Components** button, located in the bottom left corner of the **System Library Properties** page.

3. Select **Altera Host Based File System.**

4. Turn on **Add this software component**.

5. Accept the default mount point shown in the **Mount-point** box.

When creating your application project, select this system library.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Networking Examples

If you are running a networking software example, you might be asked for a nine-digit number. You are directed to find this number on a sticker on your Nios development board, identified by the prefix **ASJ**. Not all Nios development boards have this sticker.

Workaround

If your Nios development board does not have a sticker with the letters **ASJ** followed by a nine-digit number, enter a unique nine-digit number when prompted. To avoid network address conflicts, ensure that this number is unique to each Nios board connected to your network.

Solution Status

Not fixed

Flash Programmer Errata

This section describes in detail the Nios II EDS issues related to the flash programmer.

elf2flash Warning With Projects That Share a .ptf File

You might have difficulty burning flash with more than one flash programmer configuration. This issue can arise when switching between projects that share an SOPC Builder configuration file (**.ptf**). A typical example is a multi-processor project.

When you change flash configurations, and try to burn flash, you might see a message similar to the following warning:

```
(WARNING) elf2flash: Supplied reset address: 0x0
differs from the elf file reset vector: 0x500000.
```

This happens because the contents of the custom offset fields are improperly carried over from one flash programmer configuration to the next.

Workaround

Upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Delay When Creating New Flash Programmer Configuration

You might experience a delay of several seconds when creating a new Flash Programmer configuration in the Nios II IDE.

Workaround

There is no workaround.

Solution Status

Not fixed

elf2flash File Size Limit

The **elf2flash** utility supports **.elf** files up to approximately 24 MBytes in size. The **elf2flash** utility might fail on files larger than 24 MBytes, with the error message `java.lang.OutOfMemoryError`.

Workaround

Lower the number of symbols in your **.elf** file by turning off debug symbols.

Alternatively, specify less initialized data in the application.

Solution Status

Not fixed

Error Message "No such file or directory" When Programming Flash

When programming flash from a project stored in a path containing spaces, you might receive the spurious error message **No such file or directory**. The flash programmer does not correctly handle spaces in the directory path. However, this error is benign, because flash programming completes successfully.

Workaround

None required.

Solution Status

Not fixed

Flash Programmer Generates Error with Override File

Using the Nios II Flash Programmer with an override file stored in <*Nios II EDS path*>**/bin** might generate the following error:

```
4 [main] nios2-flash-programmer 4440 _cygtls::handle_exceptions: Error while dumping state
(probably corrupted stack)  Segmentation fault (core dumped)
```

Workaround

Move the override file from the <*Nios II EDS path*>/bin directory to the working directory of your Nios II IDE project. Then, specify the override file to `nios2-flash-programmer` using the `--override=` parameter. In Nios II IDE, you can do this in the **Additional nios2-flash-programmer arguments** field in the **Flash Programmer** dialog box. At the command line, the override file can be specified by adding the `--override=` argument to the command line, as follows:

```
nios2-flash-programmer --base-0x0 --override=my_fp_override.txt standard.flash
```

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Spurious Flash Verify Failure With nios2-flash-programmer

Using the `nios2-flash-programmer` command line utility to verify flash contents using the `--verify` argument might result in a verify failure even though flash contents are correct. The failure message is similar to the following message:

```
Verifying 00000000 ( 0%)Failed to verify at around
00000000  Verify failed
```

Workaround

To work around this issue, avoid using the `--verify` argument with the `nios2-flash-programmer` command-line utility. A verification of flash contents can be done by reading the flash contents into a file using the `--read` argument, then comparing the file to the input file used to program flash.

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Solution Status

Not fixed

Hardware Simulation Errata

This section describes in detail the Nios II EDS issues related to hardware simulation.

bash: vsim: command not found

You might get this error when you try to launch ModelSim from the Nios II command shell. This can happen if you have a separate Cygwin installation on your machine. Windows cannot find the path to the ModelSim executable.

Workaround

Add the path to your ModelSim installation to your Path variable.

Alternatively, upgrade to v. 8.0 or later.

Solution Status

Fixed in v. 8.0.

Simulation Fails Due to Incorrect Port Properties

An issue between SOPC Builder and the Nios II IDE causes custom instruction ports to be assigned as input, std_logic, and with width 1. This causes your system to fail to generate or simulate in ModelSim.

Workaround

After adding your custom instruction, close SOPC Builder and relaunch it.

Alternatively, upgrade to v. 8.0 or later.

Solution Status

Fixed in v. 8.0.

Error “UNC paths are not supported” Launching ModelSim

If you launch ModelSim from a working directory that is mapped via a universal naming convention (UNC) path (a path that starts with // instead of drive letter), you receive the following error message in SOPC Builder: **UNC paths are not supported. Defaulting to Windows directory**. This error occurs because ModelSim is calling a command shell, which does not support UNC paths.

Workaround

Map the UNC path to a drive letter and use the drive letter to reference the working directory in the launching shell.

Solution Status

Not fixed

Simulation Failure if Reset Address is Set to EPCS

Running ModelSim RTL simulation of a Nios II system fails if the reset address of the Nios II processor is set to an EPCS Serial Flash Controller because there is no simulation model for it.

Workaround

To simulate your system, temporarily set the Reset Address of the Nios II processor to the memory in which your application code resides (for example, SDRAM), then regenerate the system in SOPC Builder and run RTL simulation again. Before booting the Nios II processor from EPCS flash on your target board, change the Nios II Reset Address back to the EPCS Controller peripheral and regenerate the system in SOPC Builder and recompile in the Quartus II software to produce an updated SRAM object file (**.sof**) with the Nios II processor booting from EPCS flash.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Uninitialized .bss Variables in Simulation

If your program reads the value of an uninitialized `.bss` variable during HDL simulation, and the BSP (system library) is compiled with the **ModelSim only, no hardware support** property enabled in Nios II IDE, a warning appears about unfiltered data being `'x'`. This warning appears because when this property is enabled, the code that clears the `.bss` memory region is omitted to speed up HDL simulation so this memory region is uninitialized. The `.bss` region contains global and static local variables that are not initialized by the application so they default to a value of zero. When the Nios II processor reads uninitialized variables, it displays a warning and converts any of the bits of the uninitialized data to zero which correctly mimics the effect of the missing `.bss` clearing code. The HAL code that executes before and after `main()` might use `.bss` variables, so these warnings might appear even if your application does not use the `.bss` section.

Solution Status

Not fixed

Host Platform Errata

This section describes in detail the Nios II EDS issues related to the host development platform.

Windows/Cygwin: Nios II Processor Generation Failure

You might see the following error message when you generate your Nios II system in SOPC Builder:

```
Error: Generator program for module 'cpu_fpoint'
did NOT run successfully
```

This error might occur on a Windows-based system when all of the following conditions are true:

- You have installed a version of Cygwin other than the one distributed with the Quartus II software.
- You launch the Quartus II software and SOPC Builder from a Nios II SDK Shell.
- You enable the floating point custom instruction (FPCI) in the Nios II processor core.

Workaround

Launch the Quartus II software and SOPC Builder from the Windows Start menu, and regenerate your system.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Windows Vista: Limited Support in Nios II EDS

The Quartus II software introduces Windows Vista (32-bit and 64-bit) support in version 7.2. However, the Nios II Embedded Design Suite supports only the following platforms:

- Windows 2000
- Windows XP
- Windows XP (64)
- Windows Vista Business Edition (32-bit)
- SUSE 9 (32-bit)
- SUSE 9 (64-bit)
- RedHAT Linux v3.0 (32-bit)
- RedHAT Linux v3.0 (64-bit)
- RedHAT Linux v4.0 (32-bit)
- RedHAT Linux v4.0 (64-bit)

Workaround

There is no workaround.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Linux: Nios II IDE Fails to Find File

The Nios II IDE fails to open project files on Linux and reports that the **GCC** compiler is not found.

Workaround

To avoid this issue, make a link to `nios2-elf-gcc` by performing the following steps:

1. Set `QUARTUS_ROOTDIR` to the path of the Quartus II installation.

2. Run **/altera/72/nios2eds/sdk_shell**

3. Make a link to `nios2-elf-gcc` by executing the following command:

```
ln -s $SOPC_KIT_NIOS2/bin/nios2-gnutools/\
   H-i686-pc-linux-gnu/bin/nios2-elf-gcc /tmp/gcc
```

4. Add this link to the path as follows:

   ```
   export PATH=$PATH:/tmp
   ```

Alternatively, upgrade to v. 8.0 or later.

Solution Status

Fixed in v. 8.0

Windows XP: Internal Error When Repeatedly Stepping Into With Debugger

If you repeatedly use the **Step Into** command while debugging on Windows XP, you might receive the following error:

```
Retrieving Children: An internal error occurred
during: Retrieving Children
```

Workaround

End your debug session, then reopen it and resume debugging. Limit your use of the **Step Into** command.

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Linux: Process Fails To Terminate While Debugging With the Nios II ISS Target

If you try to interrupt or terminate a debug session targeting the Nios II ISS, you might see an error message **Interrupt Failed or Terminate Failed**. This means that the `nios2-iss` process failed to terminate. The debug session appears to have terminated in the IDE, but the `nios2-iss` process still continues running.

Workaround

Open a command shell and kill the `nios2-iss` process as follows:

1. Type:

   ```
   jobs
   ```

   to get a list of process IDs.

2. Type:

   ```
   kill -9 <nios2-iss process ID>
   ```

   to terminate the process.

Solution Status

Not fixed

Nios II IDE Errata: Building Projects

This section describes in detail the issues related to building projects in the Nios II IDE.

Nios II IDE Cannot Find stdio.h in Outline View

If you create a new project using the Nios II IDE and try to open the **stdio.h** file from the Outline view before building the project, the IDE displays the error message **No include files were found that matched that name**.

Workaround

Build the project before attempting to open **stdio.h**.

Solution Status

Not fixed

Nios II IDE Hangs With UNC Project Path

In the **New Project** dialog box, if you turn on **Specify Location** and specify the path in UNC form, the IDE might hang.

Workaround

Map the UNC path into a remote drive which looks like a Windows drive to the Nios II IDE.

Solution Status

Not fixed

Build Failure with Nios II Advanced Exceptions, MMU, and MPU

Projects created in the Nios II IDE for an SOPC Builder system containing a Nios II processor configured with advanced exceptions, the memory management unit (MMU), or the MPU fail to build, and generate the following error messages:

```
ERROR - Classic build flow for Nios II system library does not support the Nios II \
    advanced exceptions.
Use the non-classic Nios II Board Support Package instead.
<timestamp> - (SEVERE) generate: java.lang.IllegalStateException: \
    java.lang.IllegalStateException: \
    com.altera.ingenuous.GTFElement.GTFElementErrorException: \
    <error> element in GTF script
make[1]: *** [system_description/../obj/system.h-t] Error 1
make: *** [system_project] Error 2
```

Workaround

Create the software project with the Nios II software build tools command-line development flow. Alternatively, use the Nios II Studio.

Solution status

Not fixed

First Build in the Nios II IDE Takes Several Minutes

After you install the Nios II EDS, on a 32-bit Windows Vista system, building your first Nios II IDE project might take several minutes. This is due to the time required to index the **altera.components** project.

Workaround

Upgrade to the Nios II EDS v. 7.2 SP1 or later.

Solution Status

Fixed in version 7.2 SP1.

Nios II IDE Command-Line Tools Hang

On Windows systems, the Nios II IDE command-line tools are sometimes unable to locate the project workspace automatically. When this happens, the tools might hang.

This issue potentially affects the following tools:

- `nios2-create-system-library`
- `nios2-create-application-project`
- `nios2-build-project`
- `nios2-import-project`
- `nios2-delete-project`

Workaround

Explicitly supply the Nios II IDE command-line tools with a workspace location, by means of the `-data` command-line argument. The syntax of the `-data` argument is as follows:

`-data` *<path to workspace>*

The path to the workspace must be absolute, and must not contain whitespace. Aside from these restrictions, any valid file system path can be used for the workspace. The workspace shown in Example 6 is the default workspace used by the Nios II IDE in version 7.2.

Example 6. Providing a Workspace Location

```
nios2-create-project -data c:/altera/72/nios2eds/\
    bin/eclipse/nios2-ide-workspace-7.2 <other arguments>
```

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Read-Only Zip File System Base Address Fixed at 0

The symbol `ALTERA_RO_ZIPFS_BASE` must be set to the base address of the selected flash memory. In designs created in the Nios II IDE, `ALTERA_RO_ZIPFS_BASE` is hard-coded to 0.

Workaround

Develop your software project using the Nios II software build tools.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Build Command Not Functional for BSPs Created Using the Software Build Flow

The build option in the Nios II IDE menu does not rebuild BSPs imported into the IDE.

Workaround

To build the BSP, build the associated application project.

Solution Status

Not fixed

Linker Errors with Dual-Port Memories

If your instruction master and data master ports are connected to the same dual-port memory and the ports have different addresses, your code fails to run or you experience a linker error. The Nios II IDE does not warn you of the addressing violation.

Workaround

Assign the same address to both ports of the dual-port memory.

Solution Status

Not fixed

User-Managed BSP Settings Not Supported in Nios II IDE

For projects created using the new software build flow and imported into the Nios II IDE, the IDE configuration settings have no effect.

For example, objdump, compiler, and linker settings made in the IDE are ignored. This behavior occurs because Nios II software build tools projects are not IDE-managed projects. In addition, the make-related preferences do not pertain to imported Nios II software build tools projects. The IDE ignores these options during the build process.

Workaround

Make these settings in the project's makefile.

Solution Status

Not fixed

Error Message: 'Internal error occurred during "Launching"'

If you installed a Beta release of Nios II version 7.1 prior to installing the full release, you might encounter the error **An internal error occurred during "Launching"**, when building, running or debugging a software application project in Nios II IDE.

Workaround

Close Nios II IDE, open a Nios II command shell, and type `nios2-ide-clean`. The Nios II IDE launches again, and the error does not recur. Launching Nios II IDE with the `-clean` option does not affect any IDE or project settings you have made.

Altera strongly recommends that you uninstall Beta versions of the 7.1 tools before installing the released version of the Nios II EDS v. 7.1. Also, remove or rename the existing installation directories.

Solution Status

Fixed in v. 7.2.

URL Project Location Causes Project Creation Error

When you try to create a new project in an existing workspace, you might see an error dialog box saying:

```
Project cannot be created. Reason: Internal Error
```

This error might occur if the path to any project in the workspace is a Uniform Resource Locator (URL) location, for example **file:/F:/Design**. To view the path, right-click on the project and select **Properties**.

Workaround

Import your existing application and system library projects into a new workspace.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Incorrect Stack Checking When Interrupt Vector Custom Instruction Is Used

If you enable run-time stack checking in systems that include the interrupt vector custom instruction, the Nios II IDE might erroneously report a stack overflow or might fail to report an actual stack overflow.

Workaround

Do not enable the run-time stack checking and interrupt vector custom instruction features in the same application.

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Unable to Save Files Containing Copyright Symbol

On computers using a character encoding other than Cp1252, the copyright symbol included in the **board_diag.c** file is not recognized. Because of this, the file can not be saved using the Nios II IDE.

Workaround

Change the default text file encoding setting in the Nios II IDE to Cp1252 by performing the following steps:

1. In the Nios II IDE, select **Preferences** from the **Window** menu.

2. In the **General** settings, select **Workspace.**

3. Select **Default (Cp 1252)** in the **Text File Encoding** section and click **OK**.

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Compilation Error with Separate Exception Stack Option

Choosing the **Use a separate exception stack option** might cause the compilation error "`UNDEFINED VARIABLE %STACK_POINTER%`" when building a project. This error occurs if the exception stack is larger than the memory available for it.

Workaround

On the system library properties page for the project, turn off the separate exception stack or reduce the **Maximum exception stack size** setting.

Solution Status

Not fixed

Incorrect Stack and Heap Free Space Report

The makefile reports an incorrect number of bytes free for the stack and heap, if the heap and stack are in different memory regions.

Workaround

No workaround available.

Solution Status

Not fixed

Nios II IDE Reports Problems Without Displaying Error in Console

When building a project, the Nios II IDE reports problems, but the build output in the console does not contain any errors.

The Nios II IDE incorrectly reports some linker warnings as errors, with a dialog box saying **Errors exist in a required project**. The Dhrystone software example exhibits this behavior, and recompiling the project again makes the issue go away.

Workaround

If the Console output does not contain errors, then the project built correctly. On subsequent builds, the linker step is skipped and the errors do not appear.

Solution Status

Not fixed

Nios II IDE Errata: Debugging Projects

This section describes in detail the issues related to debugging projects in the Nios II IDE.

IDE Cannot Display Imported Profiling Data

If you create a software project in the Nios II software build tools command line flow, generate profiling data to **gmon.out** in the Nios II command shell, and then import the project into the IDE, you cannot use the Profiling perspective to view the profiling data. When you attempt to view **gmon.out**, the IDE displays the error message **nios2-elf-gprof: ../: not in a.out format**.

Workaround

Run `nios2-elf-gprof` from the Nios II command shell to generate a profiler report.

Solution Status

Not fixed

Error Message when Downloading .elf File

If the **Build Automatically** option is turned on in the IDE, you might get the following error when downloading an executable and linking formal file (**.elf**):

```
"2 [main] sh 5736 fork: child 3892 - died waiting; \
    for longjmp before initialization, errno 11"
```

The **Build Automatically** option is off by default, as recommended by the Eclipse help system.

Workaround

Turn off the **Build Automatically** option and download the **.elf** file again.

Solution status

Not fixed

Variable Casting Unsupported in ISS

Casting variables in the Variables view in the Debug perspective when using the Nios II ISS might cause an exception dialog box to open.

This exception might occur if you select **Debug as ISS** in the Nios II IDE and try to cast variables via the Variables view.

Workaround

There is no workaround.

Solution Status

Not fixed

Trace Debug Does Not Support Instruction-Related Exceptions

The new instruction-related exception handling introduced in version 8.0 is not supported by the Nios II trace tools.

Workaround

There is no workaround.

Solution Status

Not fixed

Error Accessing MMU and MPU Registers in the Debugger

You might get the following error if you try to read or write an MMU or MPU register via the Nios II Debugger:

```
nios2-gdb-server-wrapped: nios2debug.cpp:538: virtual ADI_ERROR \
    NIOS2DEBUG::leave_debug_mode(): Assertion 'm_state == STATE_DEBUG' failed.
```

Workaround

To read an MPU region, execute the following steps:

1. Set region `INDEX` in the `MPUBASE` register.
2. Exit and reenter debug mode, that is, single step.
3. Set the `RD` bit in the `MPUACC` register.

4. Exit and reenter debug mode, that is, single step.

5. Read back `MPUBASE` for pertinent information.

6. Read back `MPUACC` for pertinent information.

To set an MPU region with consecutive value changes, only the last one, when leaving debug mode, is committed.

Solution status

Not fixed

Trace Debug Does Not Support the JMPI Instruction

The new JMPI instruction introduced in version 7.2 is not supported by the Nios II trace tools.

Workaround

There is no workaround.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Cannot Locate Source Code in Driver Files Shared by Multiple Projects

If you hit a breakpoint in a driver file, and that driver file is shared with another project that is closed, the Nios II IDE might indicate that it cannot locate the source code.

Workaround

Open the closed system library project and resume debugging.

Solution Status

Not fixed

Console Window Is Not Updated After ISS Error

After performing a **Run as ISS**, if you receive an ISS error in the console window, the console is not updated subsequently.

Workaround

Close the console window after receiving an ISS error. A new console window opens when a new message is available.

Solution Status

Not fixed

Extra Arguments in Debugger for Multiple Launch Configurations

If you have multiple debug launch configurations, you might have issues with the **Additional nios2-download arguments** field. If you delete the contents of the **Additional nios2-download arguments** field in one debug launch configuration, view a second launch configuration, with a nonempty string, in the **Additional nios2-download arguments** text box, and return to the first launch configuration, the value from the second configuration appear in the first configuration.

Workaround

To remove all arguments, use a single space rather than deleting the entire contents of the field.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

GDB Error 256 When Launching Debugger

You might see an `Error 256` message when launching the debugger if you have environment variables that are longer than 1200 characters.

Workaround

Ensure that you do not have any environment variable that exceed 1200 characters.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

ISS Fails on Designs Containing TSE MAC or SG-DMA Components

You receive an Internal Error when attempting to perform an ISS simulation of designs containing the Altera TSE MAC or SG-DMA components because the Nios II ISS does not support these components.

Workaround

Remove the TSE MAC and SG-DMA components from your system and perform ISS simulation on the simplified system. You can also simulate the design in ModelSim or test it on hardware.

☞ Before removing the TSE MAC and SG-DMA components, make a copy of the unmodified system to ensure that you can return to the original configuration.

Solution Status

Not fixed

Memory Window Sets Control Register Values Incorrectly

The memory window might incorrectly set values in memory-mapped control registers. For example, writing 0x1234 to a byte addressed register results in the value 0x3434 being stored in the register. The memory window shows this incorrect value.

Workaround

Use the GNU debugger (GDB) console window in the IDE, instead of the memory window, to write to the registers. For example, type:

```
set {int} <register address>=0x1234↵
```

in the GDB console window.

☞ You must refresh the memory window for it to correctly display the target value.

Solution Status

Not fixed

Nios II IDE Debugger Cannot Display Driver Source Code for Custom Component

When debugging a custom component in the Nios II IDE, the debugger might not find the software driver files for that component.

Workaround

You must add a linked folder to an IDE project so that the IDE can find source code files associated with the component. The locations which you must add appear in a generated file called **install2.ptf** in the directory containing the **install.ptf** for the hardware design. Examine this file to find the directory path which you must add. For example, a custom component called `altera_avalon_pwm` is stored in **/data/pwm_component_test**. Example 7 shows an excerpt from **install2.ptf**.

Example 7. Excerpt from install2.ptf

```
PACKAGE install2
{
   COMPONENT altera_avalon_pwm
   {
      VERSION 2.0
      {
         local = "/data/pwm_component_test";
      }
   }
}
```

To add the path **/data/pwm_component_test** to the IDE, carry out the following steps:

1. Right-click the system library project for your design.
2. Point to **New** > **Other** > **General** > **Folder**, and click **Next**.
3. Specify the parent project (for example **syslib**) in the **Enter or select the parent folder** field.
4. Click **Advanced**, turn on **Link to folder in the file system**, and browse to or type in the path to the component.
5. Click **Finish** to link the folder.

☞ Do not link a custom component to more than one system library in the IDE. The breakpoint manager sometimes fails to handle the aliases, which results in breakpoints being hit even when removed from the breakpoint manager.

Alternatively, upgrade to v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

"Additional nios2-terminal Arguments" Field Is Not Functional

On the **Target Connection** tab of the **Run** or **Debug** launch configuration dialog box, the **Additional nios2-terminal arguments** field is not functional. Entering commands in these fields has no effect.

The **Additional nios2-download arguments** and **Additional nios2-flash-programmer arguments** fields are functional.

Workaround

Upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

MicroC/OS-II "Source Not Found" Error Occurs at Breakpoint

You might receive a **Source not found** error message if you step into MicroC/OS-II code or break on a program entry point while debugging your application.

Workaround

To eliminate this error, you must specify the proper source path mapping in the Nios II IDE by performing the following steps.

1. On the Windows menu, click **Preferences**. The **Preferences** dialog box appears.

2. Expand the **C/C++** group, then expand the **Debug** group, then click **Common Source Lookup Path**.

3. Click **Add...**. The **Add Source** dialog box appears.

4. Select **Path Mapping**, then click **OK**.
5. In the **Preferences** dialog box, select **Path Mapping: New Mapping** and click **Edit**. The **Path Mappings** dialog box appears.
6. Enter `cygdrive` in the **Name** field.
7. Click **Add...**. A dialog box appears, prompting you to specify the path mapping.
8. Enter `/cygdrive/c` in the **Compilation path** field and `c:/` in the **Local file system path** field.
9. Click **OK** in all open dialog boxes to return to the IDE workbench.

This process sets up the source path mapping for the entire workspace, which is recommended. You can also create a similar mapping for individual projects by clicking **Edit Source Lookup Path** that appears below the **Source not found** error message and following steps 4 through 8.

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Programs That Interact With a Terminal Console on Windows Do Not Work

Programs with this behavior work in v. 6.0 and earlier, but do not work in Nios II IDE v. 6.1 and later.

The Eclipse platform in v. 6.1 and later of the IDE (on Windows only) sends the string `\r\n` instead of just `\n` when running or debugging using the Terminal. This behavior can break existing software designs, and it is inconsistent with `nios2-terminal`, which still just sends `\n`.

Workaround

Change the software running on the Nios II processor to parse for `\r\n` as well as `\n`.

Solution Status

Not fixed

“Run as ModelSim” in the Nios II IDE Fails

The **Run as ModelSim** command might fail on launch configurations created in v. 7.0 or earlier of the IDE. This problem does not occur for new launch configurations.

Workaround

Select a location for the ModelSim tool from the launch configuration dialog box. You can use the **Browse** button next to the **ModelSim path** group, or type in a path to the ModelSim directory (for example **c:/altera/71/modelsim_ae/win32aloem**).

Solution Status

Not fixed

The Restart Command on the Run Menu Does Not Work

The **Restart** command on the **Run** menu does not work.

Workaround

Stop the program, then debug it again. If the debugger is hung in an endless loop, use the following **bash** alias to break the target, then stop it:

```
alias break="kill -2 \’ps -a | grep nios2-elf-gdb | cut -f6 -d' '\’"
```

Solution Status

Not fixed

Watchpoints Do Not Work on Certain Variables

Watchpoints do not work on a variable whose size is not 32 bits.

Workaround

Change the types of global and static local variables to `int`, `long`, or `unsigned long` before setting watchpoints on them.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Nios II IDE Errata: Navigating Projects

This section describes in detail the issues related to navigating projects in the Nios II IDE.

C/C++ Search Returns Nothing

The C/C++ Search in the Nios II IDE does not return results because a search index is not created at start up.

Workaround

Manually invoke indexing for projects you need to search. Refer to the "Indexing and Searching Project Content" IDE online help topic for instructions.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Nios II IDE Import System Library Dialog Box Does Not Close After Import

When importing a system library project, the Nios II IDE wizard does not close after clicking the **Finish** button. The wizard imports the project successfully after clicking **Finish** once. Clicking **Finish** again results in an error message stating that the project is already created.

Workaround

Click **Cancel** in the **Import Wizard** dialog box after clicking **Finish**.

Alternatively, upgrade to v. 8.0 or later.

Solution Status

Fixed in v. 8.0.

Nios II IDE Freezes While Displaying the Splash Screen

After clicking **Switch Workspace** on the File menu on a Windows machine, a Nios II IDE splash screen appears. Unfortunately, this splash screen obscures a dialog box which asks you to specify the new workspace. This makes the IDE appear to freeze.

Workaround

Press Alt-Tab to switch applications. Two relevant application icons appear: an Eclipse icon associated with the splash screen and a Nios II IDE icon associated with the **Workspace** dialog box. Select the Nios II icon to bring the dialog box to the foreground.

Solution Status

Not fixed

Internal Error When Double-Clicking on a Large Objdump File

On Windows when opening a large **objdump** file in the Nios II IDE, you might get the following error message:

```
Unable to create this part due to an internal error.
Reason for the failure: Editor could not be initialized.
```

Workaround

Adjust the Windows launch arguments for the Nios II IDE editor. Perform the following steps:

1. On the Windows Start menu, browse to the **Nios II EDS** program icon, right-click it, then click **Properties**. The **Windows Properties** dialog box appears.

2. In the **Target** field, append “`vmargs -Xmx1024m`” to the end of the path to the Nios II IDE executable. For example:

```
C:\altera\72\nios2eds\bin\eclipse\nios2-ide.exe -vmargs -Xmx1024m
```

Solution Status

Not fixed

C/C++ Scanner Does Not Support Certain C/C++ Constructs

The C/C++ scanner performs C/C++ Search, navigation, open declaration, and parts of content assist. Due to limitations of the C/C++ scanner, these features do not work with C code constructs not supported by the C++ language. An example is functions that take a function pointer as an argument.

Workaround

If the C/C++ Search fails, use the File Search facility.

Solution Status

Not fixed

Nios II IDE Does Not Recognize When an Object File is Built Outside the IDE

If you use a build process external to the Nios II IDE, the IDE might not recognize when you have generated or regenerated the **.elf** file for your project. In this case, the IDE debugger might report an error that the **.elf** file does not exist.

Workaround

Right-click on the project name and click **Refresh** to allow the IDE to recognize the **.elf** file.

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Nios II Studio Errata

This section describes in detail the issues related to the Nios II Studio.

Error in nios2-studio Help System

In the Windows operating system, with the Nios II Studio help system viewed in the internal browser, the following error message appears when you click any link:

A Runtime Error has occurred

Workaround

Use an external browser. To configure Nios II Studio to display the help system in an external browser, perform the following steps:

1. On the Window menu, click **Preferences**.

2. In the **Preferences** dialog box, click **Help**, and turn on **Use external browser**.

3. Expand **General** and click **Web Browser**.

4. Select **Use external Web browser**.

Solution Status

Not fixed

No System ID Validation Performed

The **Validate Nios II system ID before software download** option does not perform System ID validation on **.elf** files built outside Nios II Studio unless the System ID is specified.

Workaround

Specify the `--sidp` and `--id` arguments in the **Tcl** extra arguments field of the **Main** tab of your launch configuration.

Solution status

Not fixed

Spurious Messages When Importing Pre-8.0 Projects

Nios II Studio gives the following message when you import a project from a pre-8.0 version.

```
The project nature will be imported unchanged. If you wish to change the project \
    to a command-line project, remove the .project and .cdtproject files before \
    finishing this import.
```

The message is incorrect. The project is actually upgraded to a Nios II Studio application or BSP. You do not need to remove the specified files.

The following errors might also appear and can be ignored.

```
Unable to invoke: nios2-app-update-makefile --app-dir <app_dir> \
    --add-src-rdir .settings obj
SEVERE: nios2-app-update-makefile failed.
SEVERE: UserErrorException: The following option has the incorrect number of arguments \
    "--add-src-rdir".
```

Workaround

No action needed.

Solution status

Not fixed

public.mk Error or System Failure After Hardware Modification

If you modify your SOPC Builder system and do not regenerate your BSP in the Nios II Studio, you might receive the following error:

```
make: *** [public.mk] Error 1
```

or see system failures. System failures occur because there is a system ID mismatch that is not detected by the tools.

Workaround

Always regenerate your BSP project if you modify your SOPC Builder system.

Solution status

Not fixed

Nios II Software Build Tools Errata

This section describes in detail the issues in the Nios II software build tools.

Incorrect Linker Errors

While building a HAL-based Nios II software project, the linker displays misleading error messages if the SOPC Builder system provides less than 32 bytes of memory at the reset vector. The linker errors are similar to the following:

```
/cygdrive/d/altera/80/nios2eds/bin/nios2-gnutools/H-i686-pc-
cygwin/bin/../lib/gcc/nios2-elf/3.4.6/../../../../nios2-
elf/bin/ld:/cygdrive/d/test3/software/hello_world_0_syslib/Debug/system_d
escription/generated.x:186: warning: memory region UNDEFINED not declared
/cygdrive/d/altera/80/nios2eds/bin/nios2-gnutools/H-i686-pc-
cygwin/bin/../lib/gcc/nios2-elf/3.4.6/../../../../nios2-
elf/bin/ld:/cygdrive/d/test3/software/hello_world_0_syslib/Debug/system_d
escription/generated.x:186: parse error
collect2: ld returned 1 exit status
make: *** [hello_world_0.elf] Error 1
```

Workaround

In the SOPC Builder system, if the reset and exception vectors are in the same memory device, ensure that the exception vector is at least 32 bytes above the reset vector.

Solution Status

Fixed in v. 8.1

Java Heap Space Error When Creating BSP

When creating a BSP, the Nios II software build tools might hang with the following message:

```
SEVERE: OutOfMemoryError: Java Heap Space.
```

This problem occurs when your SOPC Builder design has connection cycles. A connection cycle consists of a group of components that master one another in a circular arrangement. For example, if `ddr_sdram_bridge` masters `lcd_sgdma`, and `lcd_sgdma` masters `ddr_sdram_bridge`, the two components form a simple connection cycle. In this situation, the software build tools are unable to create a BSP.

Workaround

Avoid connection cycles in the SOPC Builder system.

Solution Status

Not fixed

BSP Editor Incorrectly Allows Mixed Drivers

If multiple drivers are found for a single peripheral type, and you have multiple instances of the peripheral type in your SOPC Builder system, the Nios II BSP editor allows you to manually select a different driver for each instance of the peripheral or processor. For example, if you have two altera_avalon_timer devices, you can select two different timer drivers.

This behavior is not fully supported. You must choose the same driver for both instances of the peripheral type.

You can make driver assignments in one of the following ways:

- In the Nios II BSP editor, within the **Drivers** tab. The **Driver name** associated with each peripheral can be edited, if multiple driver types are found that support the peripheral. You must make the same driver type selection for each instance of a peripheral type in the list of peripherals as defined in the **Module Name** column.
- With the Nios II software build tools in command line mode (`nios2-bsp`, `nios2-bsp-create-settings`, `nios2-bsp-update-settings`): the **set_driver** command can be invoked (either from a Tcl script or with the `--cmd` option). The **set_driver** command accepts a peripheral module instance name and driver name as arguments. If you are using the command line development flow and wish to use this command, you must invoke it once for each instance of a peripheral type in your system, similar to the BSP editor flow described above.

Workaround

There is no workaround.

Solution status

Not fixed

nios2-bsp --help Displays Incorrect List of BSP Types

In version 8.0, the Nios II software build tools support additional HAL-based operating systems, when they are added as Tcl-scripted software packages. However, the `nios2-bsp` tool help message indicates that there are only two BSP types, HAL and UCOSII. If other HAL-based operating system components are added they do not appear in the `nios2-bsp` help message.

Workaround

See "nios2-bsp-create-settings Fails to List BSP Types" to get a correct, dynamic list of available BSP types.

Solution status

Not fixed

nios2-bsp-create-settings Fails to List BSP Types

The help message for `nios2-bsp-create-settings` incorrectly states that using the `--type` option with no value creates a list of available BSP types.

Workaround

To get a list of available BSP types, execute the following command:

```
nios2-bsp-create-settings --type x --settings x --sopc=<existing sopcinfo path>↵
```

This command returns an error message that lists available BSP types:

```
SEVERE: Available BSP type values for the --type argument are: [<list of BSP types>]
```

Solution status

Not fixed

Error Creating BSP for Pre-8.0 System

You receive the following error message if you try to create a BSP with the v8.0 software build tools using a hardware design from v7.2 or prior.

```
SEVERE: ErrorLogException: Attempt to load "NiosII_stratixII_2s60_standard.sopc" \
    failed. The --sopc option no longer accepts *.sopc files from SOPC Builder. \
    Re-generate your system in SOPC Builder (version 8.0 or later) to create a \
    *.sopcinfo file and use that instead.
SEVERE: nios2-bsp-update-settings failed.
nios2-bsp: nios2-bsp-update-settings failed
```

Workaround

You must regenerate your v7.2 or older design in SOPC Builder and recompile in the Quartus II software before creating a BSP using the software build tools v8.0.

Solution status

Not fixed

Makefile Error with Absolute Source Path

You might receive the following error message from the Nios II Software build tools:

```
Makefile:: *** target pattern contains no '%'
```

This error message can appear if you are using Windows, you specify an absolute source path with the `--src-dir` option, and your application directory, specified with the `--app-dir` option, is the same as your source path.

Workaround

Use a relative path to the source directory, or organize your files so that your application directory is different from your source directory.

Solution status

Not fixed

Space in --set Option Value Causes Error

When using `nios2-bsp-create-settings` or `nios2-bsp-update-settings`, or the `nios2-bsp` script, you might receive the following error message if you include spaces in the `--set` option value:

```
SEVERE: nios2-bsp-create-settings failed.
SEVERE: CommandLineErrorException: The following
option has the incorrect number of arguments
"--set".
```

Workaround

Enclose the setting value in double quotes. For example:

```
--set hal.make.BSP_CFLAGS_USER_FLAGS=\
   "-O1 -ffunction-sections -fverbose-asm \
   -fno-inline"
```

Alternatively, upgrade to the Nios II EDS v. 7.2 SP1 or later.

Solution Status

Fixed in version 7.2 SP1.

Software Build Tools Fail if Nios II EDS is Installed in a Path Containing Spaces

The Nios II Command Line Software Build Tools fail if you install the Nios II EDS in a path containing spaces.

Workaround

Reinstall the Nios II EDS into a path that does not contain spaces.

Solution Status

Not fixed

JTAG Instance Conflict for Multiprocessor Systems Imported Into IDE

When importing a multiprocessor project created using the Software Build Tools into the Nios II IDE, the instance IDs for the Nios II processor might be incorrect, causing JTAG errors during debugging.

Workaround

In the Nios II IDE **Debug** launch configuration, inspect each Nios II processor's instance number in the **Additional nios2-download arguments** field. Manually change the instance ID of each processor to the correct value. The correct instance ID is located in the <*SOPC Builder system name*>**.jdi** file. Search for the processor name. The instance ID is located in the "`node hpath`" line for that processor instance.

☞ The Nios II IDE allows you to set the instance ID only for the currently selected processor. To correct the instance ID on another processor, first select it.

Alternatively, upgrade to v. 8.0 or later.

Solution Status

Fixed in v. 8.0.

system.h File Does Not Reflect Pre-7.1 Custom Instruction Name Changes

If you instantiate a **class.ptf**-based custom instruction (created in SOPC Builder v. 7.0 or earlier) or upgrade a design containing a **class.ptf**-based custom instruction, subsequent custom instruction name changes are lost when you close and relaunch SOPC Editor. The **system.h** file generated by the Software Build Tools does not reflect the name change.

Workaround

Manually change the name in the **system.h** file.

Alternatively, upgrade to v. 8.0 or later.

Solution Status

Fixed in v. 8.0.

“make download-elf” Fails for Stratix II 2S60 RoHS Design Examples

You might see error messages when running the 2S60 RoHS example applications with the Software Build Tools. These messages appear due to an incorrect system ID value in the BSP makefile. The error messages are similar to the following:

```
Reading System ID at address 0x021208B8:
    ID value does not match: read 0x072DA598; expected 0x7D1BD35D
    Timestamp value was not verified: value was not specified
The software you are downloading may not run on the system which is
currently configured into the device.  Please download the correct SOF
or recompile
```

Workaround

Remove the system ID check from the application makefile. To remove the system ID check, carry out the following steps:

1. Open the application **Makefile,** located in the application project directory.

2. Search for "`download-elf`" to find the section that needs modification.

3. Remove the `$(SOPC_SYSID_FLAG)` argument from the `$(DOWNLOAD)` command in the following section:

```
.PHONY : download-elf
download-elf : $(ELF)
      @echo Info: Downloading $(ELF)
      $(DOWNLOAD) --go --cpu_name=$(CPU_NAME) $(SOPC_SYSID_FLAG)
$(WRITE_GMON_OPTION) $(ELF)
```

The resulting section looks like:

```
.PHONY : download-elf
download-elf : $(ELF)
      @echo Info: Downloading $(ELF)
      $(DOWNLOAD) --go --cpu_name=$(CPU_NAME) $(WRITE_GMON_OPTION) $(ELF
```

Rebuild the application.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

User-Managed BSP Settings Not Supported in Nios II IDE

This issue is described on page 45.

Improper Quote Format in system.h

The Nios II software build tools tcl command

```
add_sw_setting character system_h_define character_system_h \
    character_system_h a "character_system_h"
```

generates a **system.h** file with contents in double quotes (that is, `"a"`) instead of the proper single quotes (that is, `'a'`).

Workaround

Manually edit the **system.h** file to fix the appropriate line by changing double quotes to single quotes.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Upper-Case File Extensions Not Supported

The Nios II software build tools for applications and libraries (`nios2-app-generate-makefile` and `nios2-lib-generate-makefile` commands) do not support source files with certain upper-case extensions. If a file with an upper-case extension is included, the make command stops with no descriptive warning.

Only Nios II assembly language files built by the C preprocessor can have upper-case file extension (**.S**). All C language files must have the extension **.c** or **.h**. C++ language source files must have the extension **.cpp**, **.cxx**, **.cc,** or **.h**.

Workaround

Rename all C language files with the extension **.c** or **.h**. Rename all C++ language files with the extension **.cpp**, **.cxx**, **.cc,** or **.h**.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Peripheral Errata

This section describes in detail the Nios II EDS issues related to peripheral components.

Unaligned Transfers of Small Payloads Fail on SG-DMA

The Scatter Gather DMA SOPC Builder peripheral does not correctly handle unaligned transfers with small payloads. A payload length smaller than the data width causes erroneous data transfers.

Workaround

Avoid using DMA devices to transfer small payloads.

If absolutely necessary, for a 32-bit SGDMA, a minimum length of 4 bytes guarantees that data is transferred correctly.

Solution Status

Not fixed

IDT71V416 SRAM Component HardCopy Incompatibility

The IDT71V416 SRAM SOPC Builder component, when generated in a clock domain running at 50 MHz or less, contains logic that triggers the following Design Assistant warning, and thus is not compatible with HardCopy devices.

```
Warning: (Medium) Rule C106: Clock signal source should not drive registers that are \
    triggered by different clock edges. Found 1 node(s) related to this rule.
```

This warning appears because a negedge-clocked register is created for the SRAM write signal, to reduce latency at lower clock speeds.

Workaround

There is no workaround.

Solution status

Not fixed

Software Build Tools Fail With UART Driver

If you use the software build tools on a design containing the SOPC Builder UART peripheral, and if driver ioctl() support for the altera_avalon_uart component is enabled, then the BSP build process might fail. The **system.h** generated by the software build tools contains an invalid type for the UART parameter.

Workaround

To correct **system.h**, carry out the following steps:

1. Manually edit the **system.h** file by changing the following line:

   ```
   #define UART1_PARITY "N"
   ```

   to:

   ```
   #define UART1_PARITY 'N'
   ```

   ☞ Change double quotes to single quotes.

2. Build the BSP.

Solution status

This issue will be fixed in a future release of the Nios II EDS.

Flash Memory (CFI) Component Has Incorrect Default Timing Parameters

The Intel P30 presets for the Flash Memory (CFI) component in SOPC Builder contain slightly incorrect timing parameters. There is a small chance that these timing parameters might cause the Nios II Embedded Evaluation Kit (NEEK) on-board flash memory to function improperly with some designs.

Workaround

Instead of using the Intel 128P30 or 256P30 presets for the Flash Memory (CFI) component, use the options shown in Table 6.

Table 6. CFI Component Options for NEEK

Option Name	Option Value	
	128P30	256P30
Presets	Custom	Custom
Address Width (bits)	23	24
Data Width (bits)	16	16
Setup	25	25
Wait	100	100
Hold	20	20
Units	ns	ns

Alternatively, upgrade to v. 8.0 or later.

Solution Status

Fixed in v. 8.0.

Software Build Errors Due to _classic Suffix

This issue arises when you integrate a device driver for “classic” SOPC Builder IP, that is, IP described by a **class.ptf** file, rather than the newer **_hw.tcl** script. In this situation, the Nios II Software Build Tools append “_classic” to the class name that you specify for your custom component. The tools make these modifications, for example, to macros in **system.h** and to header file names in **system.h** and **alt_sys_init.c**. The modified names cause software build errors.

Workaround

Upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Software Build Tools Select Inappropriate Memories

When selecting default memory devices for memory regions, the Nios II software build tools sometimes select a memory that is not mastered by an appropriate master port on the Nios II IDE.

Workaround:

Upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

DMA Controller Always Busy in Burst Mode

The DMA controller component (`altera_avalon_dma`), when enabled for burst transactions, does not perform transfers at widths other than its full data width. The DMA controller is always busy.

Workaround

When bursting is enabled, the DMA controller must be programmed to perform transactions at its full data width.

Solution Status

Not fixed

Non System Wide Reset Can Cause Improper Initialization of Mailbox Core

The **altera_avalon_mailbox** peripheral might not be initialized properly when a soft (non-system-wide) reset occurs. In this condition, mailbox contents (read and write pointers) are not reinitialized and might show potentially stale data.

Workaround

Ensure that a system-wide reset event occurs by asserting the `reset_n` input to the SOPC Builder system containing the mailbox. This resets all peripherals and Nios II processors in the system.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Triple Speed Ethernet Designs Require a Manual Reset on Subsequent Software Downloads

You might experience issues in designs containing the Altera Triple Speed Ethernet MegaCore function. These issues might prevent the TSE MAC from operating properly after repeated downloads of the Nios II processor's **.elf** file. When this bug occurs, even though the Nios II processor executes the correct initialization sequence to reset the Triple Speed Ethernet component, a complete reset never occurs. The result is that the TSE MAC is unable to transmit or receive Ethernet data.

Workaround

Before downloading the **.elf** file to the Nios II processor, push the board's reset button. This action resets the Altera Triple Speed Ethernet MAC into a known good state.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Target Software Errata

This section describes in detail the Nios II EDS issues related to target software packages.

System Clock Rollover

The `gettimeofday()` function in the Nios II HAL can experience clock rollover. After a few thousand system clock ticks, the count of microseconds since system startup overflows, resulting in an incorrect function return value.

Workaround

Upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Compiling an Existing MicroC/OS-II Application Generates Errors

The version of MicroC/OS-II is upgraded as of the v. 6.1 release, which can cause this issue. After importing a Nios II IDE project that uses MicroC/OS-II created with version 6.0 SP1 or earlier, building the project in the Nios II IDE generates errors, such as

```
#error "OS_CFG.H, Missing OS_FLAGS_NBITS: Determine
#bits used for event flags, MUST be either 8, 16 or 32'
```

Workaround

Resetting the MicroC/OS-II software package settings to their defaults fixes this issue. To reset the component settings, carry out the following steps:

1. Right-click on the application project in the Nios II C/C++ view.
2. Click **System Library Properties**.
3. Click **RTOS Options...** to display the **MicroC/OS-II RTOS Options** dialog box.
4. Expand the **MicroC/OS-II** group in the upper left hand corner of the dialog box.
5. Examine each page of options, and make note of your settings.
6. Click **Restore Defaults.**
7. Reexamine each page of options, and reenter any settings that changed.
8. Click **OK** to dismiss the **MicroC/OS-II RTOS Options** dialog box.
9. Click **OK** again to dismiss the Properties dialog box.

Solution Status

This issue only affects upgrades from v. 6.0 SP1 or earlier.

stdio Does Not Work with MicroC/OS-II and Small C Library

`stdin`, `stdout`, and `stderr` do not work in MicroC/OS-II applications built with the Small C library option.

Workaround

Disable the small C library option.

Solution Status

Not fixed

cout From MicroC-OS/II Task Does Not Send Data to stdout

If neither `printf()` or `cout` is used from `main()` before tasks are started, `cout` does not work from a task.

Workaround

Add the following C++ code to the beginning of `main()`:

```
std::ios_base::sync_with_stdio(false);
```

Solution Status

Not fixed

malloc(), realloc() Failures With MicroC/OS-II

When you use the MicroC/OS-II RTOS, calls to `malloc()` and `realloc()` might fail if successive calls to `malloc()` or `realloc()` within a MicroC/OS-II task occur after changing the task priority of the task in which a memory block is originally allocated.

Workaround

Use one of the following workarounds:

- Allocate and/or reallocate memory blocks outside of MicroC/OS-II tasks, before task switching starts. This makes it possible to change thread priorities at runtime.
- Allocate fixed areas of memory using arrays (rather than using `malloc()`) before task switching starts. This makes it possible to change thread priorities at runtime.
- Allocate memory using `malloc()` or `realloc()` from a MicroC/OS-II task. You can change task priorities at runtime, but only for tasks that have not used `malloc()` or `realloc()`.

Solution Status

Not fixed

Toolchain Errata

This section describes in detail the Nios II EDS issues related to the GNU compiler toolchain (gcc, gdb, etc.).

sof2flash Names Output Files Incorrectly

If **sof2flash** is called with the `--save` option and a relative path is specified for the output file (that is, including "./" or "../"), the raw binary file that is saved is incorrectly named **.rbf** instead of **<flash filename>.rbf**.

Workaround

Provide a full path to the output file(s), or change to the output directory before running the **sof2flash** command so the output file specification is `--output=<filename>.flash`.

Solution Status

Not fixed

GNU Assembler Does Not Accept the --defsym Flag

According the GNU documentation, you can set an assembler definition by using the `--defsym` flag, but it does not work in the following form: `--defsym MY_VAR=1`

Workaround

There is no workaround.

Solution Status

Not fixed

How to Contact Altera

For the most up-to-date information about Altera® products, refer to the following table.

Contact (1)	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Altera literature services	Email	literature@altera.com
Non-technical support (General)	Email	nacomp@altera.com
(Software Licensing)	Email	authorization@altera.com

Note:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

The following table shows the typographic conventions that this document uses.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Command names, dialog box titles, checkbox options, and dialog box options are shown in bold, initial capital letters. Example: **Save As** dialog box.
bold type	External timing parameters, directory names, project names, disk drive names, file names, file name extensions, and software utility names are shown in bold type. Examples: $\mathbf{f_{MAX}}$, **\qdesigns** directory, **d:** drive, **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Document titles are shown in italic type with initial capital letters. Example: *AN 75: High-Speed Board Design*.
Italic type	Internal timing parameters and variables are shown in italic type. Examples: t_{PIA}, $n + 1$. Variable names are enclosed in angle brackets (< >) and shown in italic type. Example: *<file name>*, *<project name>***.pof** file.
Initial Capital Letters	Keyboard keys and menu names are shown with initial capital letters. Examples: Delete key, the Options menu.
"Subheading Title"	References to sections within a document and titles of on-line help topics are shown in quotation marks. Example: "Typographic Conventions."

Visual Cue	Meaning
`Courier type`	Signal and port names are shown in lowercase Courier type. Examples: `data1`, `tdi`, `input`. Active-low signals are denoted by suffix `n`, for example, `resetn`. Anything that must be typed exactly as it appears is shown in Courier type. For example: `c:\qdesigns\tutorial\chiptrip.gdf`. Also, sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword `SUBDESIGN`), as well as logic function names (for example, `TRI`) are shown in Courier.
1., 2., 3., and a., b., c., etc.	Numbered steps are used in a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ● •	Bullets are used in a list of items when the sequence of the items is not important.
✓	The checkmark indicates a procedure that consists of one step only.
☞	The hand points to information that requires special attention.
⚠ CAUTION	A caution calls attention to a condition or possible situation that can damage or destroy the product or the user's work.
⚠ WARNING	A warning calls attention to a condition or possible situation that can cause injury to the user.
↵	The angled arrow indicates you should press the Enter key.
👣	The feet direct you to more information on a particular topic.

Nios II Embedded Design Suite Release Notes and Errata

RN-EDS-2.0

About These Release Notes

These release notes cover versions 7.0 through 8.0 of the Altera® Nios® II Embedded Design Suite (EDS). These release notes describe the revision history and errata for the Nios II EDS.

☞ From v7.2 onwards, the contents of the Nios II release notes and errata sheet that Altera formerly published with each software release are superseded by two documents:

- *Nios II Embedded Design Suite Release Notes and Errata* (this document)
- The *Nios II Processor* chapter of the *MegaCore IP Library Release Notes and Errata*.

Errata are functional defects or errors, which might cause the product to deviate from published specifications. Documentation issues include errors, unclear descriptions, or omissions from current published specifications or product documents.

For the most up-to-date errata for this release, refer to the latest version of the *Nios II Embedded Design Suite Release Notes and Errata* on the Altera website.

Revision History

Table 1 shows the revision history for the Nios II EDS, including the Nios II C2H Compiler.

For more information on new features and changes, refer to the *Nios II Processor Reference Handbook*, the *Nios II Software Developer's Handbook*, and the *Nios II C2H Compiler User Guide*. For information about changes to the Nios II processor core, refer to the *MegaCore IP Library Release Notes and Errata*.

Table 1. Nios II Embedded Design Suite Revision History

Version	Date	Description
8.0	May 2008	● The Nios II BSP editor, a standalone graphical user interface (GUI), provides an easy-to-use front end for the Nios II software build tools. ● The Altera Hardware Abstraction Layer (HAL) supports an advanced exception handler that can handle Nios II memory protection unit (MPU) exceptions. ● The Nios II Studio is available as a Beta release. The Studio is an alternative to the Nios II integrated development environment (IDE). Nios II projects created with the Nios II Studio are 100% compatible with projects created with the Nios II software build tools on the command line. ● The Nios II software build tools `--sopc` option only accepts **.sopcinfo** files as arguments and not **.sopc** files. ● The Nios II IDE command-line utilities, which form the infrastructure of the Nios II IDE, are deprecated; support will be discontinued in a future release. The deprecated tools include **nios2-create-system-library**, **nios2-create-application-project**, **nios2-build-project**, **nios2-import-project**, and **nios2-delete-project**. ● The Lightweight IP (lwIP) transmission control protocol/Internet protocol (TCP/IP) suite is removed from the Nios II EDS. ● The MicroC/OS-II real-time operating system (RTOS) is upgraded from v2.83 to v2.86. ● The NicheStack TCP/IP stack — Nios II Edition is upgraded from v3.0 to v3.1. ● Certain Nios II example designs are removed from the EDS, and are distributed on the Altera website instead. The affected designs and their new locations are listed in "Example Designs Moved to the Web" on page 4. ● The NicheStack TCP/IP Stack macro definition `INICHE_DEFAULT_IF` is deprecated. For details, see "Deprecated NicheStack Setting". ● The **nios_csh** file in **sopc_builder/bin** is removed from the Nios II EDS.
7.2 SP1	December 2007	● Maintenance release

Table 1. Nios II Embedded Design Suite Revision History

Version	Date	Description
7.2	October 2007	● Nios II software build tools: Library support. ● Nios II software build tools: Support for pre-compiled libraries (.a files) as device drivers. ● Nios II software build tools: Recompiles **newlib** when new compiler flags introduced. ● Nios II software build tools: command line options to specify component search paths ● Nios II GNU compiler: targets floating point hardware when performing conversion (to and from integer) and comparison operations. ● Nios II IDE: Lauterbach instruction set simulator (ISS) debug option. ● Nios II IDE: **altera.components** project shortens project creation time. ● NicheStack TCP/IP Stack – Nios II Edition: Simplified licensing mechanism. ● NicheStack TCP/IP Stack – Nios II Edition: Increased Dynamic Host Configuration Protocol (DHCP) timeout to 130 seconds. ● Triple Speed Ethernet MegaCore® function hardware design example: Includes tightly-coupled memories; other changes. Refer to the Nios II Documentation Launchpad for details. ● Nios II processor and EPCS Flash Controller: Cyclone® III boot-loader support. ● Stratix® III and Cyclone III Fast hardware example design removed. Designs targeting those device families are available with Stratix III and Cyclone III development kits. ● Legacy software development kit (SDK) utilities **nios2-build** and **nios2-run** removed.
7.1	May 2007	● Nios II software build tools design flow ● Arria™ GX device family support ● Design examples including new SOPC Builder components ● Altera Triple Speed Ethernet (TSE) Media Access Controller (MAC) MegaCore function ● Altera Scatter-Gather Direct Memory Access (SG-DMA) Controller
7.0	March 2007	● Cyclone III device family support ● Burst support on the instruction master port

Revision Details

This section provides details about some of the Nios II EDS revisions shown in Table 1.

Example Designs Moved to the Web

Starting in v8.0, the example designs shown in Table 2 are removed from the Nios II EDS, and instead distributed on the Web, at the locations shown in the table. This applies to example designs in Nios II IDE template form, and in the form of Nios II software build tools create-this-app and create-this-bsp scripts.

Table 2. Example Designs Removed from the Nios II EDS

Software Example	Delivery Mechanism
Read-Only Zip File System	Nios Wiki
MicroC/OS-II Message Box	Nios Wiki
Hello_LED	Nios Wiki
Host File System	Nios Wiki
Dhrystone	Nios II Literature Page with the *Nios II Performance Benchmark Data Sheet*
MicroC/OS II Tutorial	Nios II Literature Page with the *Using MicroC/OS-II RTOS with the Nios II Processor Tutorial*
Tightly Coupled Memory (TCM) Tutorial	Nios II Literature Page with the *Using Nios II Tightly Coupled Memory Tutorial*
Custom Instruction Tutorial	Nios II Literature Page with the *Nios II Custom Instruction User Guide*

Deprecated NicheStack Setting

The Nios II IDE and the Nios II software build tools contain a setting to specify a default Ethernet (MAC) interface when using the NicheStack TCP/IP stack. Starting in v8.0, this setting is deprecated, and is removed from Altera software application example code that used it. The NicheStack TCP/IP stack has never depended on this setting.

This setting generates the macro `INICHE_DEFAULT_IF` in the generated **system.h** file located in the board support package (BSP). If your software depends on this macro definition, modify it, for example, to define `INICHE_DEFAULT_IF` explicitly. The macro will be removed in a future Nios II EDS release.

Altera networking example applications do not support multiple Ethernet interfaces. If you wish to use the Web Server or Simple Socket Server networking examples in a hardware system with multiple

Ethernet MAC interfaces, or with the multi-port Altera Triple Speed Ethernet MegaCore function, you need to modify the `get_ip_addr()` and `get_mac_addr()` functions to provide a unique MAC and IP address for each Ethernet interface in the system.

Errata Summary

Table 3 summarizes the issues that affect the Nios II Embedded Design Suite versions 7.0 through 8.0. For a detailed description of each errata issue, click on the issue name.

Table 3. Nios II EDS Errata

Added or Updated	Issue	Page	Affected Version					
			8.0	7.2 SP1	7.2	7.1 SP1	7.1	7.0
	C2H Compiler							
Jul 2008	C2H Cache Coherency Issue with Global Variables	13	✓	—	—	—	—	—
Jul 2008	Functions Declared Without a Return Type Are Not Supported	14	✓	—	—	—	—	—
Oct 2007	Pre-7.1 Systems Are Not Supported	14	✓	✓	✓	—	—	—
Oct 2007	--src-dir Software Build Tools Argument Does Not Work With C2H	15	✓	✓	✓	—	—	—
May 2007	Accelerator Generation Failure If Tools Are Installed in Path With Spaces	15	✓	✓	✓	✓	✓	—
May 2007	C2H Accelerators Time Out Immediately When Untethered	*(1)*	✓	✓	✓	✓	✓	—
May 2007	The C2H Compiler Regenerates an Accelerator Unnecessarily	15	✓	✓	✓	✓	✓	—
May 2007	Error: c2h_accelerator_base_addresses.h: No such file or directory	16	✓	✓	✓	✓	✓	—
May 2007	Java Heap Space Exception if Quartus II Compilation is Enabled	16	✓	✓	✓	✓	✓	—
May 2007	Pointer Dereferences to Volatile Types	17	✓	✓	✓	✓	✓	—
Dec 2006	C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File	17	✓	✓	✓	✓	✓	✓
Dec 2006	C2H Compiler Out-of-Memory Error	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Array Elements of structs Do Not Copy Correctly	18	✓	✓	✓	✓	✓	✓
Jun 2006	Build Clean Causes Build Failure	18	✓	✓	✓	✓	✓	✓
Jun 2006	Changing Build Configurations Produces Unexpected Results	19	✓	✓	✓	✓	✓	✓

Table 3. Nios II EDS Errata

Added or Updated	Issue	Page	Affected Version					
			8.0	7.2 SP1	7.2	7.1 SP1	7.1	7.0
Jun 2006	Closed System Library While Working With the C2H Compiler	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Comma Operators Are Not Supported	19	—	—	—	—	Fixed	✓
Jun 2006	Hardware Accelerators Remain After Deleting the Software Project *(2)*	20	✓	✓	✓	✓	✓	✓
Jun 2006	Incorrect Results From Logical or Conditional Operation With Side-Effects	20	✓	✓	✓	✓	✓	✓
Jun 2006	Launch SOPC Builder Button in C2H View	21	✓	✓	✓	✓	✓	✓
Jun 2006	Multiple Clock Domains Cause Hardware Accelerator to Fail	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Simulating on ISS Not Supported	*(1)*	✓	✓	✓	✓	✓	✓
Development Boards								
Jun 2006	Intermittent Failures While Accessing CompactFlash Card	22	✓	✓	✓	✓	✓	✓
Documentation Issues								
Jul 2008	Missing Source Control Information	23	✓	—	—	—	—	—
Dec 2007	Incomplete newlib Documentation	*(1)*	✓	✓	✓	—	—	—
Oct 2007	Nios II IDE Online Help Expand Buttons Do Not Work	24	✓	✓	✓	—	—	—
Oct 2007	New Control Register Names Not Listed	25	Fixed	✓	✓	—	—	—
May 2007	Arria GX Support Incorrectly Listed in Documentation HTML Page	25	—	—	Fixed	✓	✓	—
May 2007	Nios II IDE Help Topic Does Not Work With Linux Mozilla	26	✓	✓	✓	✓	✓	—
Mar 2007	GNU Documentation is Missing	*(1)*	—	—	—	—	Fixed	✓
Mar 2007	Nios II Processor Reference Handbook Refers Erroneously to the Data Bus	26	—	—	—	—	Fixed	✓
Download Cables and Debug Hardware								
Jun 2006	Communication Errors During Run/Debug Sessions Using Older Download Cables	*(1)*	✓	✓	✓	✓	✓	✓
Hardware Example Designs								
Mar 2007	Incorrect PLL Phase Shift in Stratix II and Cyclone II Designs	27	✓	✓	✓	✓	✓	✓

Table 3. Nios II EDS Errata

Added or Updated	Issue	Page	Affected Version					
			8.0	7.2 SP1	7.2	7.1 SP1	7.1	7.0
Software Example Designs								
Oct 2007	Hardware Tutorial Software Example Hangs on Some Boards	28	✓	✓	✓	—	—	—
May 2007	Dhrystone BSP Example Fails When Run on the Fast Hardware Example	29	—	—	Fixed	✓	✓	—
May 2007	Software Template Error With Non-Ethernet Hardware Design	30	—	—	Fixed	✓	✓	—
Jun 2006	Networking Examples	31	✓	✓	✓	✓	✓	✓
Jun 2006	RAM Test Failure When Running Memory Test Software Template on the ISS	*(1)*	✓	✓	✓	✓	✓	✓
Flash Programmer								
Oct 2007	elf2flash Warning With Projects That Share a .ptf File	31	—	—	Fixed	✓	✓	✓
May 2007	Delay When Creating New Flash Programmer Configuration	32	✓	✓	✓	✓	✓	—
Dec 2006	elf2flash File Size Limit	32	✓	✓	✓	✓	✓	✓
Dec 2006	“No such file or directory” Error for a Project Stored in a Path Containing Spaces	32	✓	✓	✓	✓	✓	✓
Dec 2006	Flash Programmer Generates Error with Override File	33	—	—	—	—	Fixed	✓
Dec 2006	Spurious Flash Verify Failure With nios2-flash-programmer	33	—	—	—	—	Fixed	✓
Dec 2006	Flash Programmer Error with Multiprocessor System	*(1)*	✓	✓	✓	✓	✓	✓
Hardware Simulation								
Dec 2007	Nios II Processor Fails in Simulation	*(1)*	—	Fixed	✓	✓	—	—
Oct 2007	bash: vsim: command not found	34	Fixed	✓	✓	—	—	—
Oct 2007	Simulation Fails Due to Incorrect Port Properties	34	Fixed	✓	✓	✓	✓	—
Dec 2006	Error: “UNC paths are not supported. Defaulting to Windows directory.”	35	✓	✓	✓	✓	✓	✓
Jun 2006	ModelSim Fails to Load Large Memory Models	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Simulation Failure if Reset Address is Set to EPCS	35	—	—	Fixed	✓	✓	✓

Table 3. Nios II EDS Errata

Added or Updated	Issue	Page	Affected Version					
			8.0	7.2 SP1	7.2	7.1 SP1	7.1	7.0
Jun 2006	Uninitialized .bss Variables in Simulation	35	✓	✓	✓	✓	✓	✓
Host Platform								
Dec 2007	Windows/Cygwin: Nios II Processor Generation Failure	36	✓	✓	✓	—	—	—
Oct 2007	Windows Vista: Limited Support in Nios II EDS	37	✓	✓	✓	—	—	—
Oct 2007	Linux: Nios II IDE Fails to Find File	37	Fixed	✓	✓	—	—	—
Dec 2006	Windows XP: Internal Error When Repeatedly Stepping Into With Debugger	38	—	—	—	—	Fixed	✓
Jun 2006	Linux: Process Fails To Terminate While Debugging With the Nios II ISS Target	38	✓	✓	✓	✓	✓	✓
Jun 2006	Linux: The Quartus® II Stand-Alone Programmer is Not Supported	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Windows: Frisk Antivirus Software Causes SOPC Builder and Nios II Command Shell to Be Unresponsive	*(1)*	✓	✓	✓	✓	✓	✓
Nios II IDE								
Building Projects								
Jul 2008	Nios II IDE Hangs With UNC Project Path	39	✓	—	—	—	—	—
Jul 2008	Build Failure with Nios II Advanced Exceptions, MMU, and MPU	39	✓	—	—	—	—	—
Dec 2007	First Build in the Nios II IDE Takes Several Minutes	40	—	Fixed	✓	—	—	—
Dec 2007	Nios II IDE Command-Line Tools Hang	40	✓	✓	✓	✓	✓	✓
Oct 2007	Read-Only Zip File System Base Address Fixed at 0	41	—	—	Fixed	✓	✓	—
May 2007	Build Command Not Functional for BSPs Created Using the Software Build Flow	41	✓	✓	✓	✓	✓	—
May 2007	Linker Errors with Dual-Port Memories	41	✓	✓	✓	✓	✓	—
May 2007	User-Managed BSP Settings Not Supported in Nios II IDE	42	✓	✓	✓	✓	✓	—
May 2007	Error Message: ‘Internal error occurred during “Launching”’	42	—	—	—	✓	✓	—
May 2007	URL Project Location Causes Project Creation Error	43	✓	✓	✓	✓	✓	—

Table 3. Nios II EDS Errata

Added or Updated	Issue	Page	Affected Version					
			8.0	7.2 SP1	7.2	7.1 SP1	7.1	7.0
Mar 2007	Incorrect Stack Checking When Interrupt Vector Custom Instruction Is Used	43	—	—	—	—	Fixed	✓
Mar 2007	Unable to Save Files Containing Copyright Symbol	44	—	—	—	—	Fixed	✓
Dec 2006	Compilation Error with Separate Exception Stack Option	44	✓	✓	✓	✓	✓	✓
Dec 2006	Illegal Project Location Error When Creating New Project	*(1)*	✓	✓	✓	✓	✓	✓
Dec 2006	Incorrect Address Assignment for Dual-Port Memory Mastered by Two Different Master Ports	*(1)*	✓	✓	✓	✓	✓	✓
Dec 2006	Incorrect Stack and Heap Free Space Report	44	✓	✓	✓	✓	✓	✓
Dec 2006	Nios II IDE Reports Problems Without Displaying Error in Console	45	✓	✓	✓	✓	✓	✓
Jun 2006	Build Errors After Changing Component Names in SOPC Builder	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Nios II IDE Unnecessarily Updates the SOPC Builder System File (.ptf)	*(1)*	✓	✓	✓	✓	✓	✓
Debugging Projects								
Jul 2008	Error Message when Downloading .elf File	45	✓	—	—	—	—	—
Jul 2008	Variable Casting Unsupported in ISS	46	✓	—	—	—	—	—
Jul 2008	Trace Debug Does Not Support Instruction-Related Exceptions	46	✓	—	—	—	—	—
Jul 2008	Error Accessing MMU and MPU Registers in the Debugger	46	✓	—	—	—	—	—
Oct 2007	Trace Debug Does Not Support the JMPI Instruction	47	✓	✓	✓	—	—	—
May 2007	Cannot Locate Source Code in Driver Files Shared by Multiple Projects	47	✓	✓	✓	✓	✓	—
May 2007	Console Window Is Not Updated After ISS Error	48	✓	✓	✓	✓	✓	—
May 2007	Extra Arguments in Debugger for Multiple Launch Configurations	48	—	—	Fixed	✓	✓	—
May 2007	GDB Error 256 When Launching Debugger	48	—	—	Fixed	✓	✓	—
May 2007	ISS Fails on Designs Containing TSE MAC or SG-DMA Components	49	✓	✓	✓	✓	✓	—
May 2007	Memory Window Sets Control Register Values Incorrectly	49	✓	✓	✓	✓	✓	—

Table 3. Nios II EDS Errata

Added or Updated	Issue	Page	Affected Version					
			8.0	7.2 SP1	7.2	7.1 SP1	7.1	7.0
May 2007	Previous 7.1 Beta Install Might Cause Internal Error When Launching FS2 in IDE	*(1)*	✓	✓	✓	✓	✓	—
May 2007	Nios II IDE Debugger Cannot Display Driver Source Code for Custom Component	50	—	—	Fixed	✓	✓	—
Dec 2006	“Additional nios2-terminal Arguments” Field Is Not Functional	51	—	—	—	—	Fixed	✓
Dec 2006	MicroC/OS-II “Source Not Found” Error Occurs at Breakpoint	51	—	—	—	—	Fixed	✓
Dec 2006	Programs That Interact With a Terminal Console on Windows Do Not Work	52	✓	✓	✓	✓	✓	✓
Dec 2006	“Run as ModelSim” in the Nios II IDE Fails	53	✓	✓	✓	✓	✓	—
Dec 2006	The Restart Command on the Run Menu Does Not Work	53	✓	✓	✓	✓	✓	✓
Jun 2006	Breakpoints on Adjacent Lines of Assembly Fail to Halt the Processor	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Cannot Use Watchpoints in the Nios II IDE When the FS2 Console Is Open	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Debugger Cannot Step Into __sflags(), and Continues Execution Instead	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Incorrect Breakpoint Filtering on Threads	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Missing Address and Data in the Trace View	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	nios2-gdb-server Fails to Terminate After Setting a Watch Point	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	“Step failed. Target is not responding (timed out)” Error Message	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Uninitialized Memory Error When Executing From ISS	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Watchpoints Do Not Work on Certain Variables	53	✓	✓	✓	✓	✓	✓
Navigating Projects								
May 2007	C/C++ Search Returns Nothing	54	—	—	Fixed	✓	✓	—
May 2007	Nios II IDE Import System Library Dialog Box Does Not Close After Import	54	Fixed	✓	✓	✓	✓	—
Dec 2006	Nios II IDE Freezes While Displaying the Splash Screen	54	✓	✓	✓	✓	✓	✓
Dec 2006	Internal Error When Double-Clicking on a Large Objdump File	55	✓	✓	✓	✓	✓	✓

Table 3. Nios II EDS Errata

Added or Updated	Issue	Page	Affected Version					
			8.0	7.2 SP1	7.2	7.1 SP1	7.1	7.0
Jun 2006	C/C++ Scanner Does Not Support Certain C/C++ Constructs	55	✓	✓	✓	✓	✓	✓
Jun 2006	Nios II IDE Does Not Recognize When an Object File is Built Outside the IDE	56	—	—	—	—	Fixed	✓
Nios II Studio								
Jul 2008	No System ID Validation Performed	56	✓	—	—	—	—	—
Jul 2008	Spurious Messages When Importing Pre-8.0 Projects	56	✓	—	—	—	—	—
Jul 2008	public.mk Error or System Failure After Hardware Modification	57	✓	—	—	—	—	—
Nios II Software Build Tools								
Jul 2008	BSP Editor Incorrectly Allows Mixed Drivers	58	✓	—	—	—	—	—
Jul 2008	nios2-bsp --help Displays Incorrect List of BSP Types	58	✓	—	—	—	—	—
Jul 2008	nios2-bsp-create-settings Fails to List BSP Types	59	✓	—	—	—	—	—
Jul 2008	Error Creating BSP for Pre-8.0 System	59	✓	—	—	—	—	—
Jul 2008	Makefile Error with Absolute Source Path	60	✓	—	—	—	—	—
Dec 2007	Space in --set Option Value Causes Error	60	—	Fixed	✓	—	—	—
Oct 2007	Software Build Tools Fail if Nios II EDS is Installed in a Path Containing Spaces	61	✓	✓	✓	—	—	—
Oct 2007	JTAG Instance Conflict for Multiprocessor Systems Imported Into IDE	61	Fixed	✓	✓	—	—	—
Oct 2007	system.h File Does Not Reflect Pre-7.1 Custom Instruction Name Changes	62	Fixed	✓	✓	—	—	—
May 2007	“make download-elf” Fails for Stratix II 2S60 RoHS Design Examples	62	—	—	Fixed	✓	✓	—
May 2007	User-Managed BSP Settings Not Supported in Nios II IDE	42	✓	✓	✓	✓	✓	—
May 2007	Improper Quote Format in system.h	63	—	—	Fixed	✓	✓	—
May 2007	Upper-Case File Extensions Not Supported	64	✓	✓	✓	✓	✓	—
Peripherals								
Jul 2008	DMA Transfers Truncated or Stalled	*(1)*	✓	✓	✓	✓	✓	✓
Jul 2008	Unaligned Transfers of Small Payloads Fail on SG-DMA	64	✓	—	—	—	—	—

Table 3. Nios II EDS Errata

Added or Updated	Issue	Page	Affected Version					
			8.0	7.2 SP1	7.2	7.1 SP1	7.1	7.0
Jul 2008	IDT71V416 SRAM Component HardCopy Incompatibility	65	✓	—	—	—	—	—
Jul 2008	Software Build Tools Fail With UART Driver	65	✓	—	—	—	—	—
Dec 2007	Flash Memory (CFI) Component Has Incorrect Default Timing Parameters	66	Fixed	✓	—	—	—	—
Oct 2007	Software Build Errors Due to _classic Suffix	66	—	—	Fixed	✓	✓	—
Oct 2007	Software Build Tools Select Inappropriate Memories	67	—	—	Fixed	✓	✓	—
May 2007	DMA Controller Always Busy in Burst Mode	67	✓	✓	✓	✓	✓	—
May 2007	Non System Wide Reset Can Cause Improper Initialization of Mailbox Core	68	✓	✓	✓	✓	✓	—
May 2007	Triple Speed Ethernet Designs Require a Manual Reset on Subsequent Software Downloads	68	—	—	Fixed	✓	✓	—
Jun 2006	Using the DEV_CLRn Signal Can Cause JTAG UART Instability After Device-Wide Reset	*(1)*	✓	✓	✓	✓	✓	✓
Target Software								
Jul 2008	Degraded TCP/IP Performance with NicheStack TCP/IP stack and TSE MAC	*(1)*	✓	✓	✓	—	—	—
Jul 2008	Slow TSE MAC Performance with NicheStack TCP/IP Stack	*(1)*	✓	—	—	—	—	—
Jul 2008	Default IP Address Assigned Incorrectly with NicheStack TCP/IP Stack	*(1)*	✓	—	—	—	—	—
Oct 2007	System Clock Rollover	69	—	—	Fixed	✓	✓	✓
Dec 2006	Compiling an Existing MicroC/OS-II Application Generates Errors	69	—	—	—	—	—	—
Dec 2006	Lightweight IP (lwIP) Failure When Using DHCP	*(1)*	✓	✓	✓	✓	✓	✓
Dec 2006	stdio Does Not Work with MicroC/OS-II and Small C Library	70	✓	✓	✓	✓	✓	✓
Jun 2006	cout From MicroC-OS/II Task Does Not Send Data to stdout	70	✓	✓	✓	✓	✓	✓
Jun 2006	Creating New Custom HAL Components in the Nios II IDE	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	malloc(), realloc() Failures With MicroC/OS-II	70	✓	✓	✓	✓	✓	✓
Jun 2006	Problems Using HAL Drivers With Toshiba Flash	*(1)*	✓	✓	✓	✓	✓	✓

Table 3. Nios II EDS Errata

Added or Updated	Issue	Page	Affected Version					
			8.0	7.2 SP1	7.2	7.1 SP1	7.1	7.0
Toolchain (gcc, gdb, etc.)								
Jul 2008	sof2flash Names Output Files Incorrectly	71	✓	—	—	—	—	—
Dec 2006	GNU Assembler Does Not Accept the --defsym Flag	71	✓	✓	✓	✓	✓	✓
Jun 2006	Breakpoints in C++ Constructors Fail to Halt the Processor	*(1)*	✓	✓	✓	✓	✓	✓

Notes to Table 3:

(1) This issue is documented in Altera's online Knowledge Database. To see the solution, click on the name of the issue, or search the Knowledge Database at **www.altera.com/support/kdb/kdb-index.jsp**.

(2) For further information, refer to the *Nios II C2H Compiler User Guide*.

C2H Compiler Errata

This section describes in detail the issues related to the Nios II C2H Compiler.

C2H Cache Coherency Issue with Global Variables

Cache coherency issues might be encountered when the following conditions are all true:

- A C2H-accelerated function writes to a global or static variable
- The Nios II processor is configured with a data cache
- The referenced global variable is located in cacheable memory

In this situation, the data in the global variable can be corrupted.

This issue can only arise if the accelerated function writes to the global or static variable. If the function merely reads from the variable, cache coherency is maintained correctly.

Workaround

Add a data cache flush operation in the accelerator's wrapper function. The wrapper can be found in **alt_c2h_<*function_name*>.c**. In this file, find the following comment:

```
/* Write 1 to address 0 starts the accelerator */
```

Directly after this comment, add the following function call:

```
alt_dcache_flush_all();
```

This function call ensures that the data cache is flushed immediately before the accelerator is started.

Flushing the cache entails a performance penalty.

Alternatively, allocate your global and static variables in non-cacheable regions of memory. Placing these variables in non-cacheable regions eliminates any need to perform a data cache flush.

For information about allocating variables to non-cacheable regions, refer to the *Cache and Tightly-Coupled Memory* chapter of the *Nios II Software Developer's Handbook*.

Solution status

Not fixed.

Functions Declared Without a Return Type Are Not Supported

The C2H compiler does not support functions without an explicitly declared return type.

Workaround

If you are using the implicit `int` return type, declare the return type explicitly. If your function has no return value, declare it as `void`.

Solution status

Not fixed.

Pre-7.1 Systems Are Not Supported

Starting in v. 7.1, the C2H Compiler does not support SOPC Builder systems created in SOPC Builder 7.0 or earlier (systems based on a **class.ptf** file).

Workaround

Before using the C2H Compiler, you must upgrade your SOPC Builder system. Open the system in SOPC Builder v. 7.1 or later. SOPC Builder prompts you to convert the system, creating a ***_hw.tcl** file compatible with the C2H Compiler.

Solution Status

Not fixed.

--src-dir Software Build Tools Argument Does Not Work With C2H

When using the C2H Compiler with Nios II Software Build Tools, an error occurs when using the **nios2-app-generate-makefile** argument `--src-dir` to specify a directory of source files. An example of the error is:

```
   multiple definition of 'my_accelerated_function'
```

Workaround

Specify source files individually using the `--src-files` argument.

Solution Status

Not fixed.

Accelerator Generation Failure If Tools Are Installed in Path With Spaces

If the path to your installation of the Altera Design Suite contains spaces, the C2H Compiler fails to generate the accelerators.

Workaround

Reinstall the tools to a path containing no spaces.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

The C2H Compiler Regenerates an Accelerator Unnecessarily

The C2H Compiler might regenerate an accelerator even when the accelerated function is unchanged. This problem can result from changes to files included by the C file containing the accelerated function. The C2H Compiler fails to check that the generated hardware description language (HDL) matches the previously generated HDL causing the system to be regenerated.

Workaround

To avoid this issue move the information in the include file specific to the hardware accelerator into a separate include file. This workaround prevents regeneration of the system when the HDL is unchanged.

Solution Status

Not fixed.

Error: c2h_accelerator_base_addresses.h: No such file or directory

When a C2H accelerator is compiled for the first time, the following compile-time error can result if the **Analyze all accelerators** option is selected:

```
c2h_accelerator_base_addresses.h: No such file or directory.
```

Workaround

Click **Build software and generate SOPC Builder system** and build once before building with the **Analyze all accelerators** option.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Java Heap Space Exception if Quartus II Compilation is Enabled

You might receive the following error during Quartus II compilation of your design containing a C2H accelerator if you selected the **Build software, generate SOPC Builder system, and run Quartus II compilation** in the C2H Compiler settings window.

```
    Exception in thread "main"
    java.lang.OutOfMemoryError: Java heap space

    make: *** [c2h_hdl-t] Error 1
```

Workaround

Select the **Build software and generate SOPC builder system** option in the C2H Compiler settings window, and then manually launch the Quartus II software to compile the design.

Solution Status

Not fixed.

Pointer Dereferences to Volatile Types

The C2H Compiler treats pointer dereferences to a volatile type as if they alias all other pointer dereferences. Pointers that are restrict-qualified are treated the same way.

The two loops in Example 1 cannot be scheduled concurrently because the `volatile` qualification overrides the `__restrict__` pragma.

Example 1. Non-Concurrent Loops

```
volatile int * __restrict__ fifo_rd = FIFO_RD_BASE;
volatile int * __restrict__ fifo_wr = FIFO_WR_BASE;
for ()
{
 *fifo_wr = ....;
}
for ()
{
 ... = *fifo_rd;
}
```

Workaround

Divide the function into multiple IRQ-enabled accelerators that are launched concurrently from the processor, and use FIFOs to communicate between them.

Solution Status

Not fixed.

C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File

When accelerating a function in a file, the C2H Compiler cannot link subfunctions that are defined in a different file.

Workaround

Include all subfunctions called by the accelerated function within the same source code file.

Solution Status

Not fixed.

Array Elements of structs Do Not Copy Correctly

C2H accelerators do not correctly copy array elements of structs.

For example:

```
typedef struct my_struct {
  int a;
  int b;
  int buf[BUF_SIZE];
}MY_STRUCT;
MY_STRUCT struct_a = {1, 2, {3, 3, 3, 3}};
MY_STRUCT struct_b = {9, 8, {7, 7, 7, 7}};
struct_a = struct_b;
```

In this example, the `a` and `b` elements of the structure copy correctly, but the `buf` element does not. After this assignment, `struct_a` equals `{9, 8, {3, 3, 3, 3}}`.

Workaround

Copy the array elements explicitly, as follows:

```
{
  int i=0;
  do
    {
      struct_a.buf[i] = struct_b.buf[i];
      i++;
    } while (i<LENGTH_OF_BUF_ELEMENT)
}
```

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Build Clean Causes Build Failure

Performing a clean build on a Nios II IDE project that contains a hardware accelerator can cause the next build to fail in the IDE, because the clean build erroneously deletes a file required by the C2H Compiler.

Workaround

Do not perform a clean build on projects that use hardware accelerators. If you have already performed a clean build, recompile with option **Build software, generate SOPC Builder system, and run Quartus II compilation** to regenerate the necessary files.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Changing Build Configurations Produces Unexpected Results

The C2H Compiler does not support multiple build configurations (for example Release or Debug) in the Nios II IDE. After creating one or more accelerators in a particular configuration, the C2H Compiler produces undefined results if you switch to a different build configurations and create more accelerators.

Workaround

For a specific SOPC Builder system and Nios II IDE project, specify C2H accelerators in only one build configuration. You can use multiple build configurations, as long as only one configuration specifies C2H Compiler settings.

Solution Status

Not fixed.

Comma Operators Are Not Supported

The C2H Compiler does not support comma operators, such as in Example 2.

Example 2. Comma Operators

```
for(i = 0, j = 3; i < 10; i++, j++)
{
  /* statements */
}
```

Workaround

You can code the same functionality in separate statements, as shown in Example 3.

Example 3. Simulated Comma Operators

```
j = 3;
for(i = 0; i < 10; i++)
{
  /* statements */
  j++;
}
```

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in release 7.1.

Hardware Accelerators Remain After Deleting the Software Project

If a system contains C2H accelerators, deleting the software project that defines the accelerators does not remove the accelerators from the hardware system, and the accelerator logic remains in the SOPC Builder system.

Workaround

To remove an accelerator from a system, delete the accelerator from the C2H view in the Nios II IDE first, and then recompile the software project. The C2H Compiler then removes the accelerator from the SOPC Builder system. Once the compilation is complete then the software application can be deleted from the workspace.

Solution Status

Not fixed.

Incorrect Results From Logical or Conditional Operation With Side-Effects

The C2H Compiler always evaluates both operands of logical (`&&`, `||`) and conditional (`?:`) operators. This is different from expected American National Standards Institute (ANSI) C behavior, where operands are evaluated left-to-right, and unnecessary operands are skipped.

For example, in the expression `(i-- && j--)`, if the value of `i` is zero, ANSI C does not evaluate the right-hand-side (RHS) expression, and `j` is not decremented. By contrast, the C2H Compiler evaluates both sides, decrementing `j`.

The following expressions are other examples that might be affected by this issue:

```
if (i -- || j++)

a = ((cond == 1)?  i++ : j++);
```

Workaround

Use logical and conditional operations whose operators have no side effects. Operations with side effects include pre- and post- increment and decrement operations (`++`, `--`), memory operations (`*`, `[]`, `.`, `->`), and function calls.

Solution Status

Not fixed.

Launch SOPC Builder Button in C2H View

When the Nios II IDE workspace contains multiple projects with multiple system libraries, the incorrect SOPC Builder system might open when you click **Launch SOPC Builder** in the C2H view.

Workaround

Launch SOPC Builder from the Quartus II software. Alternatively, keep only one system library project open at a time while using the C2H Compiler.

Solution Status

Not fixed.

Development Board Errata

This section describes in detail the Nios II EDS issues related to Nios development boards.

Intermittent Failures While Accessing CompactFlash Card

The Nios II Development Kit version 5.0 and later includes a CompactFlash controller peripheral suitable for interfacing to CompactFlash cards in True IDE mode on Nios development boards. In order for True IDE mode to operate, CompactFlash cards require that the ATASEL_N input be driven to ground during power-up.

The CompactFlash controller peripheral includes a configurable power register used to power-cycle CompactFlash cards in Nios II software through a metal oxide semiconductor field-effect transistor (MOSFET) on the Nios development boards. However, in certain development boards, power to the CompactFlash card does not turn off completely during this power cycle operation. Because of this condition, the CompactFlash might not sample the ATASEL_N pin during the power-cycle operation after FPGA configuration when this pin is driven to ground. Instead, the CompactFlash card might sample the ATASEL_N pin when power is first applied to the development board, when I/O is not yet driven by the FPGA (before FPGA configuration).

Workaround

If you encounter errors with CompactFlash when using the Nios development boards, try one of the following solutions:

- Use a different CompactFlash card. Certain cards are more susceptible to the power-cycling issue than others.
- Modify the Nios development board. This is recommended if you are familiar and comfortable with board-level modifications. Disconnect pin 9 (ATASEL_N) on the CompactFlash socket on your Nios development board and tie this pin to ground.

☞ The CompactFlash socket uses a staggered numbering on the pins (starting from pin 1: 1, 26, 2, 27, ...); refer to the CompactFlash Association specification for right-angle surface-mount connectors for exact specifications on this connector. This modification permanently enables True-IDE mode operation.

Solution Status

Not fixed.

Documentation Errata

This section describes in detail the Nios II EDS documentation issues.

Missing Source Control Information

The Quartus II Handbook refers to the *Using the Nios II Software Build Tools* chapter of the *Nios II Software Developer's Handbook* for information about putting Nios II projects in source control. This information does not appear in the referenced chapter.

Workaround

The missing information follows.

Archiving Nios II IDE Software Projects

This section helps you identify the files you must include when archiving a Nios II IDE software project. With this information, you can archive a Nios® II application project and its associated Nios II system library project.

You might want to archive your projects for one of the following reasons:

- To place them under source control
- To create backups
- To bundle the projects for transfer to another location

This section covers the following information:

- How to find and identify the files that you must include in an archived Nios II IDE software project.
- Which files must have write permission to allow the software projects to be built.

Table 4 and Table 5 list the files required by Nios II IDE software projects. This is the minimum set of files needed to completely rebuild a software project, including the executable and linking file (**.elf**).

Archive your Nios II IDE software projects together with the SOPC Builder system on which they are based. You cannot rebuild a Nios II IDE software project without its associated SOPC Builder system.

The files listed in Table 4 are located in the Nios II application project directory.

Table 4. Files Required for a Nios II Application Project

File Description	File Name	Write Permission Required?
All source files	for example: **app.c**, **header.h**, **assembly.s**, **lookuptable.dat**	No
Eclipse project file	**.project**	No
C/C++ Development Toolkit project file	**.cdtproject**	Yes
C/C++ Development Toolkit option file	**.cdtbuild**	No
Software configuration file	**application.stf**	No

The files listed in Table 5 are located in the Nios II system library project directory.

Table 5. Files Required for a Nios II System Library Project

File description	File name	Write permission required?
Eclipse project file	**.project**	Yes
C/C++ Development Toolkit project file	**.cdtproject**	Yes
C/C++ Development Toolkit option file	**.cdtbuild**	No
System software configuration file	**system.stf**	Yes

☞ You must have write permission for certain files. The tools write to these files as part of the build process. If the files are not writable, the toolchain fails.

Many source control tools mark local files read-only by default. In this case, you must override this behavior. You do not have to check the files out of source control unless you are modifying the Nios II software project.

Solution status

Not fixed.

Nios II IDE Online Help Expand Buttons Do Not Work

The expand text arrow and the **Show All** option in the online help do not work in Internet Explorer.

Workaround

In Internet Explorer, carry out the following actions:

1. Click the **Refresh/Show Current Topic** icon (yellow arrows icon at the top of the search pane). The table of contents appears with the selected topic highlighted.

2. Click the highlighted topic to refresh the browser frame. Expanded text works.

Alternatively, use Firefox 2.0.0.6.

Solution Status

Not fixed.

New Control Register Names Not Listed

New control register names, added to the assembler in v. 7.2, are not documented in the *Nios II Processor Reference Handbook*.

Workaround

The new control register names are shown in Table 6.

Table 6. New Control Registers

Control Register	Name
8	`pteaddr`
9	`tlbacc`
10	`tlbmisc`

Solution Status

Fixed in v. 8.0.

Arria GX Support Incorrectly Listed in Documentation HTML Page

In the Nios II 7.1 Documentation Launchpad, the **Example Designs** page incorrectly states that a fast design exists targeting the Arria GX family. Although v. 7.1 and later of the Nios II processor supports the Arria GX family, it does not include a fast design for Arria GX devices.

Workaround

Use an existing fast design targeting a Stratix II device and change the device settings.

For information about device settings, refer to the "Device Page" topic in the Quartus II software online help.

Solution Status

Fixed in v. 7.2.

Nios II IDE Help Topic Does Not Work With Linux Mozilla

The expandable Help text for the **Nios II C/C++ Application** (About Nios II IDE Projects) does not function correctly with Mozilla 1.4.3 under Linux.

Workaround

The missing information is as follows:

> A Nios II C/C++ application project contains a C/C++ program, usually including a project's `main()` function. Building a Nios II C/C++ application project results in an executable linkable file (**.elf**) that you can run on target hardware, the Nios II ISS, and the ModelSim® hardware simulator. A Nios II C/C++ application project depends on a single system library project and might reference functions in a Nios II C/C++ library project.
>
> A Nios II C/C++ application project is a Nios II IDE managed make project. The Nios II IDE creates the necessary makefiles and manages the project for you.

Refer to **www.altera.com/literature/ug/ug_nios2_ide_help.pdf** for complete text of the Nios II help system.

Solution Status

Not fixed.

Nios II Processor Reference Handbook Refers Erroneously to the Data Bus

In the *Instantiating the Nios II Processor in SOPC Builder* chapter of the *Nios II Processor Reference Handbook*, under "Caches and Tightly Coupled Memories Tab", the following text appears under "Instruction Settings":

"Usually you enable bursts on the processor's data bus when processor data is stored in DRAM, and disable bursts when processor data is stored in SRAM." This sentence should read "Usually you enable bursts on the processor's instruction master port when instructions are stored in DRAM, and disable bursts when instructions are stored in SRAM."

Solution Status

Fixed in v. 7.1.

Hardware Example Design Errata

This section describes in detail the Nios II EDS hardware example design issues.

Incorrect PLL Phase Shift in Stratix II and Cyclone II Designs

In several example designs targeting the Nios II Cyclone II 2C35 and Nios II Stratix II 2S60 RoHS development boards, the synchronous static random access memory (SSRAM) and legacy dual data rate synchronous dynamic random access memory (DDR SDRAM) clocks are implemented with an incorrect phase-locked loop (PLL) phase shift.

This issue affects the following example designs:

- 2s60 RoHS TSE SG-DMA
- 2s60 RoHS Standard
- 2s60 RoHS Full Featured
- 2c35 TSE SG-DMA
- 2c35 Standard
- 2c35 Full Featured

This issue also affects any custom 2c35 or 2s60 RoHS design using SSRAM or legacy DDR SDRAM.

The incorrect phase shift might cause timing violations when reading from or writing to SSRAM or legacy DDR SDRAM.

Workaround

To correct the PLL settings, perform the following steps:

1. Open the standard design in SOPC Builder.

2. Double-click the component instance named **pll** to launch the PLL MegaWizard interface.

3. Click **Launch Altera's ALTPLL MegaWizard** to launch the MegaWizard interface.

4. Select the **Output Clocks** page and then the clock page used to generate the SSRAM clock. Depending on the particular design, this might be **clk c1**, **clk c2**, or another clock.

5. Ensure that **Clock phase shift** is –3.38 ns.

6. Click the clock page used to generate the write clock for the legacy DDR SDRAM, if any. Depending on the particular design, this might be **clk c1**, **clk c2**, or another clock.

7. Ensure that **Clock phase shift** is 270 degrees.

8. Click **Finish** to save changes and exit the PLL MegaWizard interface.

9. Click **Finish** to save settings to the PLL instance in SOPC Builder.

10. Regenerate the system in SOPC Builder and recompile in the Quartus II software. Refer to the **readme.txt** file in the design example folder for information on compiling designs containing the DDR controller.

For further information refer to **ssram_interface_readme.html**, located in the <*Quartus II installation directory*>**/sopc_builder/documents** folder. This document discusses the SSRAM timing analysis methodology in detail. Additionally, *AN 411: Understanding PLL Timing for Stratix II Devices* discusses clock phase shift calculations and assignments for PLLs in Stratix II devices.

Solution Status

Not fixed.

Software Example Design Errata

This section describes in detail the Nios II EDS software example design issues.

Hardware Tutorial Software Example Hangs on Some Boards

The **count binary** application used in the tutorial might hang when run on some designs that do not contain a liquid crystal display (LCD) peripheral.

Workaround

Modify line 18 of the **count_binary.h** file, as follows:

```
    # define LCD_PRINTF(lcd, args...) /* Do Nothing */
```

Rebuild and run the software again.

Solution Status

Not fixed.

Dhrystone BSP Example Fails When Run on the Fast Hardware Example

You might see the following output when running the Dhrystone BSP example in the Nios II IDE after creating and building it via command line.

```
  Int_Comp:          36608
        should be:   18
  Str_Comp:          ·"
nios2-terminal: exiting due to ^D on remote
```

This error is caused by settings in the **create-this-bsp** script that cause the application to overrun the amount of memory available in the Fast design.

Workaround

To run the Dhrystone example successfully with the Software Build Tools, edit the **create-this-bsp** script located in the *<hardware example design>***/software_examples/bsp/hal_dhrystone** folder. Carry out the following steps:

1. If the BSP already exists, remove the **settings.bsp** file and run `make clean` in the BSP folder.

2. Change the following line:

```
--set hal.enable_lightweight_device_driver_api true \
```

   to:

```
--set hal.enable_reduced_device_drivers true \
```

3. Run the **create-this-bsp** script. (Alternatively, run the **create-this-app** script, found in the Dhrystone application folder.)

The BSP is regenerated with the new setting.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Software Template Error With Non-Ethernet Hardware Design

You might get an internal error if your hardware design does not contain an Ethernet MAC, and you try to create a software example from one of the following templates:

- host_filesystem
- simple_socket_server
- web_server
- zip_filesystem

These templates check for components in the system library and incorrectly test for the existence of a MAC.

Workaround

Create the system library project for your hardware design before creating your application project from one of the templates. In the system library properties, turn on the Host File System feature by carrying out the following steps:

1. Right-click the system library project and select **Properties**.

2. Click the **Software Components** button, located in the bottom left corner of the **System Library Properties** page.

3. Select **Altera Host Based File System.**

4. Turn on **Add this software component**.

5. Accept the default mount point shown in the **Mount-point** box.

When creating your application project, select this system library.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Networking Examples

If you are running a networking example design, you might be asked for a nine-digit number. You are directed to find this number on a sticker on your Nios development board, identified by the prefix **ASJ**. Not all Nios development boards have this sticker.

Workaround

If your Nios development board does not have a sticker with the letters **ASJ** followed by a nine-digit number, enter a unique nine-digit number when prompted. To avoid network address conflicts, ensure that this number is unique to each Nios board connected to your network.

Solution Status

Not fixed.

Flash Programmer Errata

This section describes in detail the Nios II EDS issues related to the flash programmer.

elf2flash Warning With Projects That Share a .ptf File

You might have difficulty burning flash with more than one flash programmer configuration. This issue can arise when switching between projects that share an SOPC Builder configuration file (**.ptf**). A typical example is a multi-processor project.

When you change flash configurations, and try to burn flash, you might see a message similar to the following warning:

```
(WARNING) elf2flash: Supplied reset address: 0x0
differs from the elf file reset vector: 0x500000.
```

This happens because the contents of the custom offset fields are improperly carried over from one flash programmer configuration to the next.

Workaround

Upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Delay When Creating New Flash Programmer Configuration

You might experience a delay of several seconds when creating a new Flash Programmer configuration in the Nios II IDE.

Workaround

There is no workaround.

Solution Status

Not fixed.

elf2flash File Size Limit

The **elf2flash** utility supports **.elf** files up to approximately 24 MBytes in size. The **elf2flash** utility might fail on files larger than 24 MBytes, with the error message `java.lang.OutOfMemoryError`.

Workaround

Lower the number of symbols in your **.elf** file by turning off debug symbols.

Alternatively, specify less initialized data in the application.

Solution Status

Not fixed.

"No such file or directory" Error for a Project Stored in a Path Containing Spaces

You might receive this spurious error message when programming flash for a project stored in a path containing spaces. The flash programmer does not correctly handle spaces in the directory path. However, this error is benign, because flash programming completes successfully.

Workaround

None required.

Solution Status

Not fixed.

Flash Programmer Generates Error with Override File

Using the Nios II Flash Programmer with an override file stored in <*Nios II EDS path*>**/bin** might generate an error:

```
4 [main] nios2-flash-programmer 4440 _cygtls::handle_exceptions: Error while dumping state
(probably corrupted stack)  Segmentation fault (core dumped)
```

Workaround

Move the override file from the <*Nios II EDS path*>`/bin` directory to the working directory of your Nios II IDE project. Then, specify the override file to **nios2-flash-programmer** using the `--override=` parameter. In Nios II IDE, you can do this in the **Additional nios2-flash-programmer arguments** field in the **Flash Programmer** dialog box. At the command line, the override file can be specified by adding the `--override=` argument to the command line, as follows:

```
nios2-flash-programmer --base-0x0 --override=my_fp_override.txt standard.flash
```

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Spurious Flash Verify Failure With nios2-flash-programmer

Using the **nios2-flash-programmer** command line utility to verify flash contents using the `--verify` argument might result in a verify failure even though flash contents are correct. The failure message is similar to the following message:

```
Verifying 00000000 ( 0%)Failed to verify at around
00000000  Verify failed
```

Workaround

To work around this issue, avoid using the `--verify` argument with the **nios2-flash-programmer** command-line utility. A verification of flash contents can be done by reading the flash contents into a file using the `--read` argument, then comparing the file to the input file used to program flash.

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Solution Status

Not fixed.

Hardware Simulation Errata

This section describes in detail the Nios II EDS issues related to hardware simulation.

bash: vsim: command not found

You might get this error when you try to launch ModelSim from the Nios II command shell. This can happen if you have a separate Cygwin installation on your machine. Windows cannot find the path to the ModelSim executable.

Workaround

Add the path to your ModelSim installation to your Path variable.

Alternatively, upgrade to v. 8.0 or later.

Solution Status

Fixed in v. 8.0.

Simulation Fails Due to Incorrect Port Properties

An issue between SOPC Builder and the Nios II IDE causes custom instruction ports to be assigned as input, std_logic, and with width 1. This causes your system to fail to generate or simulate in ModelSim.

Workaround

After adding your custom instruction, close SOPC Builder and re-launch it.

Alternatively, upgrade to v. 8.0 or later.

Solution Status

Fixed in v. 8.0.

Error: "UNC paths are not supported. Defaulting to Windows directory."

If you launch ModelSim from a working directory that is mapped via a universal naming convention (UNC) path (a path that starts with `//` instead of drive letter), you receive this error message in SOPC Builder. This error occurs because ModelSim is calling a command shell, which does not support UNC paths.

Workaround

Map the UNC path to a drive letter and use the drive letter to reference the working directory in the launching shell.

Solution Status

Not fixed.

Simulation Failure if Reset Address is Set to EPCS

Running ModelSim RTL simulation of a Nios II system fails if the reset address of the Nios II processor is set to an EPCS Serial Flash Controller because there is no simulation model for it.

Workaround

To simulate your system, temporarily set the Reset Address of the Nios II processor to the memory in which your application code resides (for example, SDRAM), then regenerate the system in SOPC Builder and run RTL simulation again. Before booting the Nios II processor from EPCS flash on your target board, change the Nios II Reset Address back to the EPCS Controller peripheral and regenerate the system in SOPC Builder and recompile in the Quartus II software to produce an updated SRAM object file (**.sof**) with the Nios II processor booting from EPCS flash.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Uninitialized .bss Variables in Simulation

If your program reads the value of an uninitialized `.bss` variable during HDL simulation, and the BSP (system library) is compiled with the **ModelSim only, no hardware support** property enabled in Nios II IDE, a warning appears about unfiltered data being `'x'`. This warning appears because when this property is enabled, the code that clears the `.bss` memory region is omitted to speed up HDL simulation so this memory region is uninitialized. The `.bss` region contains global and static local variables that are not initialized by the application so they default to a value of zero. When the Nios II processor reads uninitialized variables, it displays a warning and converts any of the bits of the uninitialized data to zero which correctly mimics the effect of the missing `.bss` clearing code. The HAL code that executes before and after `main()` might use `.bss` variables, so these warnings might appear even if your application does not use the `.bss` section.

Solution Status

Not fixed.

Host Platform Errata

This section describes in detail the Nios II EDS issues related to the host development platform.

Windows/Cygwin: Nios II Processor Generation Failure

You might see the following error message when you generate your Nios II system in SOPC Builder:

```
Error: Generator program for module 'cpu_fpoint'
did NOT run successfully
```

This error might occur on a Windows-based system when all of the following conditions are true:

- You have installed a version of Cygwin other than the one distributed with the Quartus II software.
- You launch the Quartus II software and SOPC Builder from a Nios II SDK Shell.
- You enable the floating point custom instruction (FPCI) in the Nios II processor core.

Workaround

Launch the Quartus II software and SOPC Builder from the Windows Start menu, and regenerate your system.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Windows Vista: Limited Support in Nios II EDS

The Quartus II software introduces Windows Vista (32-bit and 64-bit) support in version 7.2. However, the Nios II Embedded Design Suite supports only the following platforms:

- Windows 2000
- Windows XP
- Windows XP (64)
- Windows Vista Business Edition (32-bit)
- SUSE 9 (32-bit)
- SUSE 9 (64-bit)
- RedHAT Linux v3.0 (32-bit)
- RedHAT Linux v3.0 (64-bit)
- RedHAT Linux v4.0 (32-bit)
- RedHAT Linux v4.0 (64-bit)

Workaround

There is no workaround.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Linux: Nios II IDE Fails to Find File

The Nios II IDE fails to open project files on Linux and reports that the **GCC** compiler is not found.

Workaround

To avoid this issue, make a link to **nios2-elf-gcc** by performing the following steps:

1. Set `QUARTUS_ROOTDIR` to the path of the Quartus II installation.

2. Run **/altera/72/nios2eds/sdk_shell**

3. Make a link to **nios2-elf-gcc** by executing the following command:

```
ln -s $SOPC_KIT_NIOS2/bin/nios2-gnutools/\
   H-i686-pc-linux-gnu/bin/nios2-elf-gcc /tmp/gcc
```

4. Add this link to the path as follows:

```
export PATH=$PATH:/tmp
```

Alternatively, upgrade to v. 8.0 or later.

Solution Status

Fixed in v. 8.0

Windows XP: Internal Error When Repeatedly Stepping Into With Debugger

If you repeatedly use the **Step Into** command while debugging on Windows XP, you might receive the following error:

```
Retrieving Children: An internal error occurred
during: Retrieving Children
```

Workaround

End your debug session, then re-open it and resume debugging. Limit your use of the **Step Into** command.

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Linux: Process Fails To Terminate While Debugging With the Nios II ISS Target

If you try to interrupt or terminate a debug session targeting the Nios II ISS, you might see an error message **Interrupt Failed or Terminate Failed**. This means that the **nios2-iss** process failed to terminate. The debug session appears to have terminated in the IDE, but the **nios2-iss** process still continues running.

Workaround

Open a command shell and kill the **nios2-iss** process as follows:

1. Type:

```
jobs
```

   to get a list of process IDs.

2. Type:

```
kill -9<nios2-iss process ID>
```

to terminate the process.

Solution Status

Not fixed.

Nios II IDE Errata: Building Projects

This section describes in detail the issues related to building projects in the Nios II IDE.

Nios II IDE Hangs With UNC Project Path

In the **New Project** dialog box, if you turn on **Specify Location** and specify the path in UNC form, the IDE might hang.

Workaround

Map the UNC path into a remote drive which looks like a Windows drive to the Nios II IDE.

Solution Status

Not fixed.

Build Failure with Nios II Advanced Exceptions, MMU, and MPU

Projects created in the Nios II IDE for an SOPC Builder system containing a Nios II processor configured with advanced exceptions, the memory management unit (MMU), or the MPU fail to build, and generate the following error messages:

```
ERROR - Classic build flow for Nios II system library does not support the Nios II \
    advanced exceptions.
Use the non-classic Nios II Board Support Package instead.
<timestamp> - (SEVERE) generate: java.lang.IllegalStateException: \
    java.lang.IllegalStateException: \
    com.altera.ingenuous.GTFElement.GTFElementErrorException: \
    <error> element in GTF script
make[1]: *** [system_description/../obj/system.h-t] Error 1
make: *** [system_project] Error 2
```

Workaround

Create the software project with the Nios II software build tools command-line development flow. Alternatively, use the Nios II Studio.

Solution status

Not fixed.

First Build in the Nios II IDE Takes Several Minutes

After you install the Nios II EDS, on a 32-bit Windows Vista system, building your first Nios II IDE project might take several minutes. This is due to the time required to index the **altera.components** project.

Workaround

Upgrade to the Nios II EDS v. 7.2 SP1 or later.

Solution Status

Fixed in version 7.2 SP1.

Nios II IDE Command-Line Tools Hang

On Windows systems, the Nios II IDE command-line tools are sometimes unable to locate the project workspace automatically. When this happens, the tools might hang.

This issue potentially affects the following tools:

- **nios2-create-system-library**
- **nios2-create-application-project**
- **nios2-build-project**
- **nios2-import-project**
- **nios2-delete-project**

Workaround

Explicitly supply the Nios II IDE command-line tools with a workspace location, by means of the `-data` command-line argument. The syntax of the `-data` argument is as follows:

`-data` <*path to workspace*>

The path to the workspace must be absolute, and must not contain whitespace. Aside from these restrictions, any valid file system path can be used for the workspace. The workspace shown in Example 4 is the default workspace used by the Nios II IDE in version 7.2.

Example 4. Providing a Workspace Location

```
nios2-create-project -data c:/altera/72/nios2eds/\
bin/eclipse/nios2-ide-workspace-7.2 <other arguments>
```

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Read-Only Zip File System Base Address Fixed at 0

The symbol ALTERA_RO_ZIPFS_BASE must be set to the base address of the selected flash memory. In designs created in the Nios II IDE, ALTERA_RO_ZIPFS_BASE is hard-coded to 0.

Workaround

Develop your software project using the Nios II software build tools.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Build Command Not Functional for BSPs Created Using the Software Build Flow

The build option in the Nios II IDE menu does not rebuild BSPs imported into the IDE.

Workaround

To build the BSP, build the associated application project.

Solution Status

Not fixed.

Linker Errors with Dual-Port Memories

If your instruction master and data master ports are connected to the same dual-port memory and the ports have different addresses, your code fails to run or you experience a linker error. The Nios II IDE does not warn you of the addressing violation.

Workaround

Assign the same address to both ports of the dual-port memory.

Solution Status

Not fixed.

User-Managed BSP Settings Not Supported in Nios II IDE

For projects created using the new software build flow and imported into the Nios II IDE, the IDE configuration settings have no effect.

For example, objdump, compiler, and linker settings made in the IDE are ignored. This behavior occurs because Nios II software build tools projects are not IDE-managed projects. In addition, the make-related preferences do not pertain to imported Nios II software build tools projects. The IDE ignores these options during the build process.

Workaround

Make these settings in the project's makefile.

Solution Status

Not fixed.

Error Message: ‘Internal error occurred during “Launching”’

If you installed a Beta release of Nios II version 7.1 prior to installing the full release, you might encounter the error **An internal error occurred during "Launching"**, when building, running or debugging a software application project in Nios II IDE.

Workaround

Close Nios II IDE, open a Nios II command shell, and type `nios2-ide-clean`. The Nios II IDE launches again, and the error does not recur. Launching Nios II IDE with the `-clean` option does not affect any IDE or project settings you have made.

Altera strongly recommends that you uninstall Beta versions of the 7.1 tools before installing the released version of the Nios II EDS v. 7.1. Also, remove or rename the existing installation directories.

Solution Status

Fixed in v. 7.2.

URL Project Location Causes Project Creation Error

When you try to create a new project in an existing workspace, you might see an error dialog box saying:

```
Project cannot be created. Reason: Internal Error
```

This error might occur if the path to any project in the workspace is a Uniform Resource Locator (URL) location, for example **file:/F:/Design**. To view the path, right-click on the project and select **Properties**.

Workaround

Import your existing application and system library projects into a new workspace.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Incorrect Stack Checking When Interrupt Vector Custom Instruction Is Used

If you enable run-time stack checking in systems that include the interrupt vector custom instruction, the Nios II IDE might erroneously report a stack overflow or might fail to report an actual stack overflow.

Workaround

Do not enable the run-time stack checking and interrupt vector custom instruction features in the same application.

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Unable to Save Files Containing Copyright Symbol

On computers using a character encoding other than Cp1252, the copyright symbol included in the **board_diag.c** file is not recognized. Because of this, the file can not be saved using the Nios II IDE.

Workaround

Change the default text file encoding setting in the Nios II IDE to Cp1252 by performing the following steps:

1. In the Nios II IDE, select **Preferences** from the **Window** menu.
2. In the **General** settings, select **Workspace.**
3. Select **Default (Cp 1252)** in the **Text File Encoding** section and click **OK**.

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Compilation Error with Separate Exception Stack Option

Choosing the **Use a separate exception stack option** might cause the compilation error "`UNDEFINED VARIABLE %STACK_POINTER%`" when building a project. This error occurs if the exception stack is larger than the memory available for it.

Workaround

On the system library properties page for the project, turn off the separate exception stack or reduce the **Maximum exception stack size** setting.

Solution Status

Not fixed.

Incorrect Stack and Heap Free Space Report

The makefile reports an incorrect number of bytes free for the stack and heap, if the heap and stack are in different memory regions.

Workaround

No workaround available.

Solution Status

Not fixed.

Nios II IDE Reports Problems Without Displaying Error in Console

When building a project, the Nios II IDE reports problems, but the build output in the console does not contain any errors.

The Nios II IDE incorrectly reports some linker warnings as errors, with a dialog box saying **Errors exist in a required project**. The Dhrystone software example exhibits this behavior, and recompiling the project again makes the issue go away.

Workaround

If the Console output does not contain errors, then the project built correctly. On subsequent builds, the linker step is skipped and the errors do not appear.

Solution Status

Not fixed.

Nios II IDE Errata: Debugging Projects

This section describes in detail the issues related to debugging projects in the Nios II IDE.

Error Message when Downloading .elf File

If the **Build Automatically** option is turned on in the IDE, you might get the following error when downloading an executable and linking formal file (**.elf**):

```
"2 [main] sh 5736 fork: child 3892 - died waiting; \
    for longjmp before initialization, errno 11"
```

The **Build Automatically** option is off by default, as recommended by the Eclipse help system.

Workaround

Turn off the **Build Automatically** option and download the **.elf** file again.

Solution status

Not fixed.

Variable Casting Unsupported in ISS

Casting variables in the Variables view in the Debug perspective when using the Nios II ISS might cause an exception dialog box to open.

This exception might occur if you select **Debug as ISS** in the Nios II IDE and try to cast variables via the Variables view.

Workaround

There is no workaround.

Solution Status

Not fixed.

Trace Debug Does Not Support Instruction-Related Exceptions

The new instruction-related exception handling introduced in version 8.0 is not supported by the Nios II trace tools.

Workaround

There is no workaround.

Solution Status

Not fixed.

Error Accessing MMU and MPU Registers in the Debugger

You might get the following error if you try to read or write an MMU or MPU register via the Nios II Debugger:

```
nios2-gdb-server-wrapped: nios2debug.cpp:538: virtual ADI_ERROR \
    NIOS2DEBUG::leave_debug_mode(): Assertion 'm_state == STATE_DEBUG' failed.
```

Workaround

To read an MPU region:

1. Set region `INDEX` in the `MPUBASE` register.
2. Exit and re-enter debug mode, that is, single step.
3. Set the `RD` bit in the `MPUACC` register.
4. Exit and re-enter debug mode, that is, single step.
5. Read back `MPUBASE` for pertinent information.
6. Read back `MPUACC` for pertinent information.

To set an MPU region with consecutive value changes, only the last one, when leaving debug mode, is committed.

Solution status

Not fixed.

Trace Debug Does Not Support the JMPI Instruction

The new JMPI instruction introduced in version 7.2 is not supported by the Nios II trace tools.

Workaround

There is no workaround.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Cannot Locate Source Code in Driver Files Shared by Multiple Projects

If you hit a breakpoint in a driver file, and that driver file is shared with another project that is closed, the Nios II IDE might indicate that it cannot locate the source code.

Workaround

Open the closed system library project and resume debugging.

Solution Status

Not fixed.

Console Window Is Not Updated After ISS Error

After performing a **Run as ISS**, if you receive an ISS error in the console window, the console is not updated subsequently.

Workaround

Close the console window after receiving an ISS error. A new console window opens when a new message is available.

Solution Status

Not fixed.

Extra Arguments in Debugger for Multiple Launch Configurations

If you have multiple debug launch configurations, you might have issues with the **Additional nios2-download arguments** field. If you delete the contents of the **Additional nios2-download arguments** field in one debug launch configuration, view a second launch configuration, with a nonempty string, in the **Additional nios2-download arguments** text box, and return to the first launch configuration, the value from the second configuration appear in the first configuration.

Workaround

To remove all arguments, use a single space rather than deleting the entire contents of the field.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

GDB Error 256 When Launching Debugger

You might see an `Error 256` message when launching the debugger if you have environment variables that are longer than 1200 characters.

Workaround

Ensure that you do not have any environment variable that exceed 1200 characters.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

ISS Fails on Designs Containing TSE MAC or SG-DMA Components

You receive an Internal Error when attempting to perform an ISS simulation of designs containing the Altera TSE MAC or SG-DMA components because the Nios II ISS does not support these components.

Workaround

Remove the TSE MAC and SG-DMA components from your system and perform ISS simulation on the simplified system. You can also simulate the design in ModelSim or test it on hardware.

☞ Before removing the TSE MAC and SG-DMA components, make a copy of the unmodified system to ensure that you can return to the original configuration.

Solution Status

Not fixed.

Memory Window Sets Control Register Values Incorrectly

The memory window might incorrectly set values in memory-mapped control registers. For example, writing 0x1234 to a byte addressed register results in the value 0x3434 being stored in the register. The memory window shows this incorrect value.

Workaround

Use the GNU debugger (GDB) console window in the IDE, instead of the memory window, to write to the registers. For example, type:

```
set {int} <register address>=0x1234↵
```

in the GDB console window.

☞ You must refresh the memory window in order for it to correctly display the target value.

Solution Status

Not fixed.

Nios II IDE Debugger Cannot Display Driver Source Code for Custom Component

When debugging a custom component in the Nios II IDE, the debugger might not find the software driver files for that component.

Workaround

You must add a linked folder to an IDE project so that the IDE can find source code files associated with the component. The locations which you must add appear in a generated file called **install2.ptf** in the directory containing the **install.ptf** for the hardware design. Examine this file to find the directory path which you must add. For example, a custom component called **altera_avalon_pwm** is stored in **/data/pwm_component_test**. An excerpt from **install2.ptf** follows:

```
PACKAGE install2
{
   COMPONENT altera_avalon_pwm
   {
      VERSION 2.0
      {
         local = "/data/pwm_component_test";
      }
   }
}
```

To add the path **/data/pwm_component_test** to the IDE, carry out the following steps:

1. Right-click the system library project for your design.

2. Point to **New > Other > General > Folder**, and click **Next**.

3. Specify the parent project (for example **syslib**) in the **Enter or select the parent folder** field.

4. Click **Advanced**, turn on **Link to folder in the file system**, and browse to or type in the path to the component.

5. Click **Finish** to link the folder.

☞ Do not link a custom component to more than one system library in the IDE. The breakpoint manager sometimes fails to handle the aliases, which results in breakpoints being hit even when removed from the breakpoint manager.

Alternatively, upgrade to v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

“Additional nios2-terminal Arguments” Field Is Not Functional

On the **Target Connection** tab of the **Run** or **Debug** launch configuration dialog box, the **Additional nios2-terminal arguments** field is not functional. Entering commands in these fields has no effect.

The **Additional nios2-download arguments** and **Additional nios2-flash-programmer arguments** fields are functional.

Workaround

Upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

MicroC/OS-II “Source Not Found” Error Occurs at Breakpoint

You might receive a 'Source not found' error message if you step into MicroC/OS-II code or break on a program entry point while debugging your application.

Workaround

To eliminate this error, you must specify the proper source path mapping in the Nios II IDE by performing the following steps.

1. On the Windows menu, click **Preferences**. The **Preferences** dialog box appears.

2. Expand the **C/C++** group, then expand the **Debug** group, then click **Common Source Lookup Path**.

3. Click **Add...**. The **Add Source** dialog box appears.

4. Select **Path Mapping**, then click **OK**.

5. In the **Preferences** dialog box, select **Path Mapping: New Mapping** and click **Edit**. The **Path Mappings** dialog box appears.

6. Enter `cygdrive` in the **Name** field.

7. Click **Add...**. A dialog box appears, prompting you to specify the path mapping.

8. Enter `/cygdrive/c` in the **Compilation path** field and `c:/` in the **Local file system path** field.

9. Click **OK** in all open dialog boxes to return to the IDE workbench.

This process sets up the source path mapping for the entire workspace, which is recommended. You can also create a similar mapping for individual projects by clicking **Edit Source Lookup Path** that appears below the **Source not found** error message and following steps 4 through 8.

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Programs That Interact With a Terminal Console on Windows Do Not Work

Programs with this behavior work in v. 6.0 and earlier, but do not work in Nios II IDE v. 6.1 and later.

The Eclipse platform in v. 6.1 and later of the IDE (on Windows only) sends the string `\r\n` instead of just `\n` when running or debugging using the Terminal. This behavior can break existing software designs, and it is inconsistent with **nios2-terminal**, which still just sends `\n`.

Workaround

Change the software running on the Nios II processor to parse for `\r\n` as well as `\n`.

Solution Status

Not fixed.

“Run as ModelSim” in the Nios II IDE Fails

The **Run as ModelSim** command might fail on launch configurations created in v. 7.0 or earlier of the IDE. This problem does not occur for new launch configurations.

Workaround

Select a location for the ModelSim tool from the launch configuration dialog box. You can use the **Browse** button next to the **ModelSim path** group, or type in a path to the ModelSim directory (for example **c:/altera/71/modelsim_ae/win32aloem**).

Solution Status

Not fixed.

The Restart Command on the Run Menu Does Not Work

The **Restart** command on the **Run** menu does not work.

Workaround

Stop the program, then debug it again. If the debugger is hung in an endless loop, use the following **bash** alias to break the target, then stop it:

```
alias break="kill -2 \’ps -a | grep nios2-elf-gdb | cut -f6 -d' '\’"
```

Solution Status

Not fixed.

Watchpoints Do Not Work on Certain Variables

Watchpoints do not work on a variable whose size is not 32 bits.

Workaround

Change the types of global and static local variables to `int`, `long`, or `unsigned long` before setting watchpoints on them.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Nios II IDE Errata: Navigating Projects

This section describes in detail the issues related to navigating projects in the Nios II IDE.

C/C++ Search Returns Nothing

The C/C++ Search in the Nios II IDE does not return results because a search index is not created at start up.

Workaround

Manually invoke indexing for projects you need to search. Refer to the "Indexing and Searching Project Content" IDE online help topic for instructions.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Nios II IDE Import System Library Dialog Box Does Not Close After Import

When importing a system library project, the Nios II IDE wizard does not close after clicking the **Finish** button. The wizard imports the project successfully after clicking **Finish** once. Clicking **Finish** again results in an error message stating that the project is already created.

Workaround

Click **Cancel** in the **Import Wizard** dialog box after clicking **Finish**.

Alternatively, upgrade to v. 8.0 or later.

Solution Status

Fixed in v. 8.0.

Nios II IDE Freezes While Displaying the Splash Screen

After clicking **Switch Workspace** on the File menu on a Windows machine, a Nios II IDE splash screen appears. Unfortunately, this splash screen obscures a dialog box which asks you to specify the new workspace. This makes the IDE appear to freeze.

Workaround

Press Alt-Tab to switch applications. Two relevant application icons appear: an Eclipse icon associated with the splash screen and a Nios II IDE icon associated with the **Workspace** dialog box. Select the Nios II icon to bring the dialog box to the foreground.

Solution Status

Not fixed.

Internal Error When Double-Clicking on a Large Objdump File

On Windows when opening a large **objdump** file in the Nios II IDE, you might get the following error message:

```
Unable to create this part due to an internal error.
Reason for the failure: Editor could not be initialized.
```

Workaround

Adjust the Windows launch arguments for the Nios II IDE editor. Perform the following steps:

1. On the Windows Start menu, browse to the **Nios II EDS** program icon, right-click it, then click **Properties**. The **Windows Properties** dialog box appears.

2. In the **Target** field, append “`vmargs -Xmx1024m`” to the end of the path to the Nios II IDE executable. For example:

```
C:\altera\72\nios2eds\bin\eclipse\nios2-ide.exe -vmargs -Xmx1024m
```

Solution Status

Not fixed.

C/C++ Scanner Does Not Support Certain C/C++ Constructs

The C/C++ scanner performs C/C++ Search, navigation, open declaration, and parts of content assist. Due to limitations of the C/C++ scanner, these features do not work with C code constructs not supported by the C++ language. An example is functions that take a function pointer as an argument.

Workaround

If the C/C++ Search fails, use the File Search facility.

Solution Status

Not fixed.

Nios II IDE Does Not Recognize When an Object File is Built Outside the IDE

If you use a build process external to the Nios II IDE, the IDE might not recognize when you have generated or re-generated the **.elf** file for your project. In this case, the IDE debugger might report an error that the **.elf** file does not exist.

Workaround

Right-click on the project name and click **Refresh** to allow the IDE to recognize the **.elf** file.

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Nios II Studio Errata

No System ID Validation Performed

The **Validate Nios II system ID before software download** option does not perform System ID validation on **.elf** files built outside Nios II Studio unless the System ID is specified.

Workaround

Specify the `--sidp` and `--id` arguments in the **Tcl** extra arguments field of the **Main** tab of your launch configuration.

Solution status

Not fixed.

Spurious Messages When Importing Pre-8.0 Projects

Nios II Studio gives the following message when you import a project from a pre-8.0 version.

```
The project nature will be imported unchanged. If you wish to change the project \
    to a command-line project, remove the .project and .cdtproject files before \
    finishing this import.
```

The message is incorrect. The project is actually upgraded to a Nios II Studio application or BSP. You do not need to remove the specified files.

The following errors might also appear and can be ignored.

```
Unable to invoke: nios2-app-update-makefile --app-dir <app_dir> \
    --add-src-rdir .settings obj
SEVERE: nios2-app-update-makefile failed.
SEVERE: UserErrorException: The following option has the incorrect number of arguments \
    "--add-src-rdir".
```

Workaround

No action needed.

Solution status

Not fixed.

public.mk Error or System Failure After Hardware Modification

If you modify your SOPC Builder system and do not regenerate your BSP in the Nios II Studio, you might receive the following error:

```
make: *** [public.mk] Error 1
```

or see system failures. System failures occur because there is a system ID mismatch that is not detected by the tools.

Workaround

Always regenerate your BSP project if you modify your SOPC Buider system.

Solution status

Not fixed.

Nios II Software Build Tools Errata

This section describes in detail the issues in the Nios II software build tools.

BSP Editor Incorrectly Allows Mixed Drivers

If multiple drivers are found for a single peripheral type, and you have multiple instances of the peripheral type in your SOPC Builder system, the Nios II BSP editor allows you to manually select a different driver for each instance of the peripheral or processor. For example, if you have two altera_avalon_timer devices, you can select two different timer drivers.

This behavior is not fully supported. You must choose the same driver for both instances of the peripheral type.

You can make driver assignments in one of the following ways:

- In the Nios II BSP editor, within the **Drivers** tab. The **Driver name** associated with each peripheral can be edited, if multiple driver types are found that support the peripheral. You must make the same driver type selection for each instance of a peripheral type in the list of peripherals as defined in the **Module Name** column.
- With the Nios II software build tools in command line mode (**nios2-bsp**, **nios2-bsp-create-settings**, **nios2-bsp-update-settings**): the **set_driver** command can be invoked (either from a Tcl script or with the `--cmd` option). The **set_driver** command accepts a peripheral module instance name and driver name as arguments. If you are using the command line development flow and wish to use this command, you must invoke it once for each instance of a peripheral type in your system, similar to the BSP editor flow described above.

Workaround

There is no workaround.

Solution status

Not fixed.

nios2-bsp --help Displays Incorrect List of BSP Types

In version 8.0, the Nios II software build tools support additional HAL-based operating systems, when they are added as Tcl-scripted software packages. However, the **nios2-bsp** tool help message indicates that there are only two BSP types, HAL and UCOSII. If other HAL-based operating system components are added they do not appear in the **nios2-bsp** help message.

Workaround

See "nios2-bsp-create-settings Fails to List BSP Types" to get a correct, dynamic list of available BSP types.

Solution status

Not fixed.

nios2-bsp-create-settings Fails to List BSP Types

The help message for **nios2-bsp-create-settings** incorrectly states that using the `--type` option with no value creates a list of available BSP types.

Workaround

To get a list of available BSP types, execute the following command:

```
nios2-bsp-create-settings --type x --settings x --sopc=<existing sopcinfo path>↵
```

This returns an error message that lists available BSP types:

```
SEVERE: Available BSP type values for the --type argument are: [<list of BSP types>]
```

Solution status

Not fixed.

Error Creating BSP for Pre-8.0 System

You receive the following error message if you try to create a BSP with the v8.0 software build tools using a hardware design from v7.2 or prior.

```
SEVERE: ErrorLogException: Attempt to load "NiosII_stratixII_2s60_standard.sopc" \
    failed. The --sopc option no longer accepts *.sopc files from SOPC Builder. \
    Re-generate your system in SOPC Builder (version 8.0 or later) to create a \
    *.sopcinfo file and use that instead.
SEVERE: nios2-bsp-update-settings failed.
nios2-bsp: nios2-bsp-update-settings failed
```

Workaround

You must regenerate your v7.2 or older design in SOPC Builder and recompile in the Quartus II software before creating a BSP using the software build tools v8.0.

Solution status

Not fixed.

Makefile Error with Absolute Source Path

You might receive a "`Makefile:: *** target pattern contains no '%'`" error message from the Software build tools. This error message can appear if you are using Windows, you specify an absolute source path with the `--src-dir` option, and your application directory, specified with the `--app-dir` option, is the same as your source path.

Workaround

Use a relative path to the source directory, or organize your files so that your application directory is different from your source directory.

Solution status

Not fixed.

Space in --set Option Value Causes Error

When using **nios2-bsp-create-settings** or **nios2-bsp-update-settings**, or the **nios2-bsp** script, you might receive the following error message if you include spaces in the `--set` option value:

```
SEVERE: nios2-bsp-create-settings failed.
SEVERE: CommandLineErrorException: The following
option has the incorrect number of arguments
"--set".
```

Workaround

Enclose the setting value in double quotes. For example:

```
--set hal.make.BSP_CFLAGS_USER_FLAGS=\
   "-O1 -ffunction-sections -fverbose-asm \
   -fno-inline"
```

Alternatively, upgrade to the Nios II EDS v. 7.2 SP1 or later.

Solution Status

Fixed in version 7.2 SP1.

Software Build Tools Fail if Nios II EDS is Installed in a Path Containing Spaces

The Nios II Command Line Software Build Tools fail if you install the Nios II EDS in a path containing spaces.

Workaround

Re-install the Nios II EDS into a path that does not contain spaces.

Solution Status

Not fixed.

JTAG Instance Conflict for Multiprocessor Systems Imported Into IDE

When importing a multiprocessor project created using the Software Build Tools into the Nios II IDE, the instance IDs for the Nios II processor might be incorrect, causing JTAG errors during debugging.

Workaround

In the Nios II IDE **Debug** launch configuration, inspect each Nios II processor's instance number in the **Additional nios2-download arguments** field. Manually change the instance ID of each processor to the correct value. The correct instance ID is located in the <*SOPC Builder system name*>**.jdi** file. Search for the processor name. The instance ID is located in the "`node hpath`" line for that processor instance.

☞ The Nios II IDE allows you to set the instance ID only for the currently selected processor. To correct the instance ID on another processor, first select it.

Alternatively, upgrade to v. 8.0 or later.

Solution Status

Fixed in v. 8.0.

system.h File Does Not Reflect Pre-7.1 Custom Instruction Name Changes

If you instantiate a **class.ptf**-based custom instruction (created in SOPC Builder v. 7.0 or earlier) or upgrade a design containing a **class.ptf**-based custom instruction, subsequent custom instruction name changes are lost when you close and re-launch SOPC Editor. The **system.h** file generated by the Software Build Tools does not reflect the name change.

Workaround

Manually change the name in the **system.h** file.

Alternatively, upgrade to v. 8.0 or later.

Solution Status

Fixed in v. 8.0.

"make download-elf" Fails for Stratix II 2S60 RoHS Design Examples

You might see error messages when running the 2S60 RoHS example applications with the Software Build Tools. These messages appear due to an incorrect system ID value in the BSP makefile. The error messages are similar to the following:

```
Reading System ID at address 0x021208B8:
    ID value does not match: read 0x072DA598; expected 0x7D1BD35D
    Timestamp value was not verified: value was not specified
The software you are downloading may not run on the system which is
currently configured into the device.  Please download the correct SOF
or recompile
```

Workaround

Remove the system ID check from the application makefile. To remove the system ID check, carry out the following steps:

1. Open the application **Makefile,** located in the application project directory.

2. Search for "`download-elf`" to find the section that needs modification.

3. Remove the `$(SOPC_SYSID_FLAG)` argument from the `$(DOWNLOAD)` command in the following section:

```
.PHONY : download-elf
download-elf : $(ELF)
      @echo Info: Downloading $(ELF)
      $(DOWNLOAD) --go --cpu_name=$(CPU_NAME) $(SOPC_SYSID_FLAG)
$(WRITE_GMON_OPTION) $(ELF)
```

The resulting section looks like:

```
.PHONY : download-elf
download-elf : $(ELF)
      @echo Info: Downloading $(ELF)
      $(DOWNLOAD) --go --cpu_name=$(CPU_NAME) $(WRITE_GMON_OPTION) $(ELF
```

Rebuild the application.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

User-Managed BSP Settings Not Supported in Nios II IDE

This issue is described on page 42.

Improper Quote Format in system.h

The Nios II software build tools tcl command

```
add_sw_setting character system_h_define character_system_h \
    character_system_h a "character_system_h"
```

generates a **system.h** file with contents in double quotes (that is, `"a"`) instead of the proper single quotes (that is, `'a'`).

Workaround

Manually edit the **system.h** file to fix the appropriate line by changing double quotes to single quotes.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Upper-Case File Extensions Not Supported

The Nios II software build tools for applications and libraries (**nios2-app-generate-makefile** and **nios2-lib-generate-makefile** commands) do not support source files with certain upper-case extensions. If a file with an upper-case extension is included, the make command stops with no descriptive warning.

Only Nios II assembly language files built by the C preprocessor can have upper-case file extension (**.S**). All C language files must have the extension **.c** or **.h**. C++ language source files must have the extension **.cpp**, **.cxx**, **.cc,** or **.h**.

Workaround

Rename all C language files with the extension **.c** or **.h**. Rename all C++ language files with the extension **.cpp**, **.cxx**, **.cc,** or **.h**.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Peripheral Errata

This section describes in detail the Nios II EDS issues related to peripheral components.

Unaligned Transfers of Small Payloads Fail on SG-DMA

The Scatter Gather DMA SOPC Builder peripheral does not correctly handle unaligned transfers with small payloads. A payload length smaller than the data width causes erroneous data transfers.

Workaround

Avoid using DMA devices to transfer small payloads.

If absolutely necessary, for a 32-bit SGDMA, a minimum length of 4 bytes guarantees that data is transferred correctly.

Solution Status

Not fixed.

IDT71V416 SRAM Component HardCopy Incompatibility

The IDT71V416 SRAM SOPC Builder component, when generated in a clock domain running at 50 MHz or less, contains logic that triggers the following Design Assistant warning, and thus is not compatible with HardCopy devices.

```
Warning: (Medium) Rule C106: Clock signal source should not drive registers that are \
    triggered by different clock edges. Found 1 node(s) related to this rule.
```

This warning appears because a negedge-clocked register is created for the SRAM write signal, in order to reduce latency at lower clock speeds.

Workaround

There is no workaround.

Solution status

Not fixed.

Software Build Tools Fail With UART Driver

If you use the software build tools on a design containing the SOPC Builder UART peripheral, and if driver ioctl() support for the altera_avalon_uart component is enabled, then the BSP build process might fail. The **system.h** generated by the software build tools contains an invalid type for the UART parameter.

Workaround

To correct **system.h**, carry out the following steps:

1. Manually edit the **system.h** file by changing the following line:

   ```
   #define UART1_PARITY "N"
   ```

   to:

   ```
   #define UART1_PARITY 'N'
   ```

   ☞ Change double quotes to single quotes.

2. Build the BSP.

Solution status

This issue will be fixed in a future release of the Nios II EDS.

Flash Memory (CFI) Component Has Incorrect Default Timing Parameters

The Intel P30 presets for the Flash Memory (CFI) component in SOPC Builder contain slightly incorrect timing parameters. There is a small chance that these timing parameters might cause the Nios II Embedded Evaluation Kit (NEEK) on-board flash memory to function improperly with some designs.

Workaround

Instead of using the Intel 128P30 or 256P30 presets for the Flash Memory (CFI) component, use the options shown in Table 7.

Table 7. CFI Component Options for NEEK

Option Name	Option Value	
	128P30	256P30
Presets	Custom	Custom
Address Width (bits)	23	24
Data Width (bits)	16	16
Setup	25	25
Wait	100	100
Hold	20	20
Units	ns	ns

Alternatively, upgrade to v. 8.0 or later.

Solution Status

Fixed in v. 8.0.

Software Build Errors Due to _classic Suffix

This issue arises when you integrate a device driver for “classic” SOPC Builder IP, that is, IP described by a **class.ptf** file, rather than the newer **_hw.tcl** script. In this situation, the Nios II Software Build Tools append “_classic” to the class name that you specify for your custom component.

The tools make these modifications, for example, to macros in **system.h** and to header file names in **system.h** and **alt_sys_init.c**. The modified names cause software build errors.

Workaround

Upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Software Build Tools Select Inappropriate Memories

When selecting default memory devices for memory regions, the Nios II software build tools sometimes select a memory that is not mastered by an appropriate master port on the Nios II IDE.

Workaround:

Upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

DMA Controller Always Busy in Burst Mode

The DMA controller component (**altera_avalon_dma**), when enabled for burst transactions, does not perform transfers at widths other than its full data width. The DMA controller is always busy.

Workaround

When bursting is enabled, the DMA controller must be programmed to perform transactions at its full data width.

Solution Status

Not fixed.

Non System Wide Reset Can Cause Improper Initialization of Mailbox Core

The **altera_avalon_mailbox** peripheral might not be initialized properly when a soft (non-system-wide) reset occurs. In this condition, mailbox contents (read and write pointers) are not reinitialized and might show potentially stale data.

Workaround

Ensure that a system-wide reset event occurs by asserting the `reset_n` input to the SOPC Builder system containing the mailbox. This resets all peripherals and Nios II processors in the system.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Triple Speed Ethernet Designs Require a Manual Reset on Subsequent Software Downloads

You might experience issues in designs containing the Altera Triple Speed Ethernet MegaCore function. These issues might prevent the TSE MAC from operating properly after repeated downloads of the Nios II processor's **.elf** file. When this bug occurs, even though the Nios II processor executes the correct initialization sequence to reset the Triple Speed Ethernet component, a complete reset never occurs. The result is that the TSE MAC is unable to transmit or receive Ethernet data.

Workaround

Before downloading the **.elf** file to the Nios II processor, push the board's reset button. This action resets the Altera Triple Speed Ethernet MAC into a known-good state.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Target Software Errata

This section describes in detail the Nios II EDS issues related to target software packages.

System Clock Rollover

The `gettimeofday()` function in the Nios II HAL can experience clock rollover. After a few thousand system clock ticks, the count of microseconds since system startup overflows, resulting in an incorrect function return value.

Workaround

Upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Compiling an Existing MicroC/OS-II Application Generates Errors

The version of MicroC/OS-II is upgraded as of the v. 6.1 release, which can cause this issue. After importing a Nios II IDE project that uses MicroC/OS-II created with version 6.0 SP1 or earlier, building the project in the Nios II IDE generates errors, such as

```
#error "OS_CFG.H, Missing OS_FLAGS_NBITS: Determine
#bits used for event flags, MUST be either 8, 16 or 32'
```

Workaround

Resetting the MicroC/OS-II software package settings to their defaults fixes this issue. To reset the component settings, carry out the following steps:

1. Right-click on the application project in the Nios II C/C++ view.
2. Click **System Library Properties**.
3. Click **RTOS Options...** to display the **MicroC/OS-II RTOS Options** dialog box.
4. Expand the **MicroC/OS-II** group in the upper-left-hand corner of the dialog box.
5. Examine each page of options, and make note of your settings.
6. Click **Restore Defaults.**
7. Re-examine each page of options, and re-enter any settings that changed.

8. Click **OK** to dismiss the **MicroC/OS-II RTOS Options** dialog box.

9. Click **OK** again to dismiss the Properties dialog box.

Solution Status

This issue only affects upgrades from v. 6.0 SP1 or earlier.

stdio Does Not Work with MicroC/OS-II and Small C Library

`stdin`, `stdout`, and `stderr` do not work in MicroC/OS-II applications built with the Small C library option.

Workaround

Disable the small C library option.

Solution Status

Not fixed.

cout From MicroC-OS/II Task Does Not Send Data to stdout

If neither `printf()` or `cout` is used from `main()` before tasks are started, `cout` does not work from a task.

Workaround

Add the following C++ code to the beginning of `main()`:

```
std::ios_base::sync_with_stdio(false);
```

Solution Status

Not fixed.

malloc(), realloc() Failures With MicroC/OS-II

When you use the MicroC/OS-II RTOS, calls to `malloc()` and `realloc()` might fail if successive calls to `malloc()`or `realloc()` within a MicroC/OS-II task occur after changing the task priority of the task in which a memory block is originally allocated.

Workaround

Use one of the following workarounds:

- Allocate and/or reallocate memory blocks outside of MicroC/OS-II tasks, before task switching starts. This makes it possible to change thread priorities at runtime.
- Allocate fixed areas of memory using arrays (rather than using `malloc()`) before task switching starts. This makes it possible to change thread priorities at runtime.
- Allocate memory using `malloc()` or `realloc()` from a MicroC/OS-II task. You can change task priorities at runtime, but only for tasks that have not used `malloc()` or `realloc()`.

Solution Status

Not fixed.

Toolchain Errata

This section describes in detail the Nios II EDS issues related to the GNU compiler toolchain (gcc, gdb, etc.).

sof2flash Names Output Files Incorrectly

If **sof2flash** is called with the `--save` option and a relative path is specified for the output file (that is, including "./" or "../"), the raw binary file that is saved is incorrectly named **.rbf** instead of **<flash filename>.rbf**.

Workaround

Provide a full path to the output file(s), or change to the output directory before running the **sof2flash** command so the output file specification is `--output=`*`<filename>`*`.flash`.

Solution Status

Not fixed.

GNU Assembler Does Not Accept the --defsym Flag

According the GNU documentation, you can set an assembler definition by using the `--defsym` flag, but it does not work in the following form: `--defsym MY_VAR=1`

Workaround

There is no workaround.

Solution Status

Not fixed.

How to Contact Altera

For the most up-to-date information about Altera® products, refer to the following table.

Contact (1)	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Product literature	Website	www.altera.com/literature
Altera literature services	Email	literature@altera.com
Non-technical support (General)	Email	nacomp@altera.com
(Software Licensing)	Email	authorization@altera.com

Note:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

The following table shows the typographic conventions that this document uses.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Command names, dialog box titles, checkbox options, and dialog box options are shown in bold, initial capital letters. Example: **Save As** dialog box.
bold type	External timing parameters, directory names, project names, disk drive names, file names, file name extensions, and software utility names are shown in bold type. Examples: $\mathbf{f_{MAX}}$, **\qdesigns** directory, **d:** drive, **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Document titles are shown in italic type with initial capital letters. Example: *AN 75: High-Speed Board Design*.
Italic type	Internal timing parameters and variables are shown in italic type. Examples: t_{PIA}, $n + 1$. Variable names are enclosed in angle brackets (< >) and shown in italic type. Example: *<file name>*, *<project name>***.pof** file.
Initial Capital Letters	Keyboard keys and menu names are shown with initial capital letters. Examples: Delete key, the Options menu.
“Subheading Title”	References to sections within a document and titles of on-line help topics are shown in quotation marks. Example: “Typographic Conventions.”

Visual Cue	Meaning
`Courier type`	Signal and port names are shown in lowercase Courier type. Examples: `data1`, `tdi`, `input`. Active-low signals are denoted by suffix `n`, for example, `resetn`. Anything that must be typed exactly as it appears is shown in Courier type. For example: `c:\qdesigns\tutorial\chiptrip.gdf`. Also, sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword `SUBDESIGN`), as well as logic function names (for example, `TRI`) are shown in Courier.
1., 2., 3., and a., b., c., etc.	Numbered steps are used in a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ● •	Bullets are used in a list of items when the sequence of the items is not important.
✓	The checkmark indicates a procedure that consists of one step only.
☞	The hand points to information that requires special attention.
⚠ CAUTION	A caution calls attention to a condition or possible situation that can damage or destroy the product or the user's work.
⚠ WARNING	A warning calls attention to a condition or possible situation that can cause injury to the user.
↵	The angled arrow indicates you should press the Enter key.
👣	The feet direct you to more information on a particular topic.

Nios II Embedded Design Suite Release Notes and Errata

RN-EDS-1.1

About These Release Notes

These release notes cover versions 6.1 through 7.2 SP1 of the Altera® Nios® II Embedded Design Suite (EDS). These release notes describe the new features and errata for the Nios II EDS.

☞ From v7.2 onwards, the contents of the Nios II release notes and errata sheet that Altera formerly published with each software release are superseded by two documents:

- *Nios II Embedded Design Suite Release Notes and Errata* (this document)
- The "Nios II Processor" chapter of the *MegaCore® IP Library Release Notes and Errata*.

Errata are functional defects or errors, which might cause the product to deviate from published specifications. Documentation issues include errors, unclear descriptions, or omissions from current published specifications or product documents.

For the most up-to-date errata for this release, refer to the latest version of the *Nios II Embedded Design Suite Release Notes and Errata* on the Altera website.

New Features

Table 1 shows the new features history for the Nios II EDS, including the Nios II C2H Compiler.

For more information on the new features, refer to the *Nios II Processor Reference Handbook*, the *Nios II Software Developer's Handbook*, and the *Nios II C2H Compiler User Guide*. For information about changes to the Nios II processor core, refer to the *MegaCore IP Library Release Notes and Errata*.

Table 1. Nios II Embedded Design Suite New Features History

Version	Date	Description
7.2 SP1	December 2007	● Maintenance release
7.2	October 2007	● Nios II software build tools: Library support. ● Nios II software build tools: Support for pre-compiled libraries (.a files) as device drivers. ● Nios II software build tools: Recompiles **newlib** when new compiler flags introduced. ● Nios II software build tools: command line options to specify component search paths ● Nios II GNU compiler: targets floating point hardware when performing conversion (to and from integer) and comparison operations. ● Nios II Integrated Development Environment (IDE): Lauterbach instruction set simulator (ISS) debug option. ● Nios II IDE: **altera.components** project shortens project creation time. ● NicheStack TCP/IP Stack – Nios II Edition: Simplified licensing mechanism. ● NicheStack TCP/IP Stack – Nios II Edition: Increased DHCP timeout to 130 seconds. ● Triple Speed Ethernet MegaCore function hardware design example: Includes tightly-coupled memories; other changes. Refer to the Nios II Documentation Launchpad for details. ● Nios II processor and ECPS Flash Controller: Cyclone® III boot-loader support. ● Stratix® III and Cyclone III Fast hardware example design removed. Designs targeting those device families are available with Stratix III and Cyclone III development kits. ● Legacy SDK utilities **nios2-build** and **nios2-run** removed.
7.1	May 2007	● Nios II Software Build Tools design flow ● Arria™ GX device family support ● Design examples including new SOPC Builder components ● Altera Triple Speed Ethernet (TSE) Media Access Controller (MAC) MegaCore ● Altera Scatter-Gather Direct Memory Access (SG-DMA) Controller
7.0	March 2007	● Cyclone III device family support ● Instruction master burst support
6.1	December 2006	● NicheStack TCP/IP Stack - Nios II Edition ● Interrupt vector custom instruction ● Nios II integrated development environment (IDE) based on the latest version of the Eclipse platform ● Memory footprint reduction options

Errata Summary

Table 2 summarizes the issues that affect the Nios II Embedded Design Suite versions 6.1 through 7.2 SP1. For a detailed description of each errata issue, click on the issue name.

Table 2. Nios II EDS Errata

Added or Updated	Issue	Page	Affected Version					
			7.2 SP1	7.2	7.1 SP1	7.1	7.0	6.1
C2H Compiler								
Oct 2007	Pre-7.1 Systems Are Not Supported	10	✓	✓	—	—	—	—
Oct 2007	--src-dir Software Build Tools Argument Does Not Work With C2H	10	✓	✓	—	—	—	—
May 2007	Accelerator Generation Failure If Tools Are Installed in Path With Spaces	11	✓	✓	✓	✓	—	—
May 2007	C2H Accelerators Time Out Immediately When Untethered	*(1)*	✓	✓	✓	✓	—	—
May 2007	The C2H Compiler Regenerates an Accelerator Unnecessarily	11	✓	✓	✓	✓	—	—
May 2007	Error: c2h_accelerator_base_addresses.h: No such file or directory	12	✓	✓	✓	✓	—	—
May 2007	Java Heap Space Exception if Quartus II Compilation is Enabled	12	✓	✓	✓	✓	—	—
May 2007	Pointer Dereferences to Volatile Types	12	✓	✓	✓	✓	—	—
Dec 2006	C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File	13	✓	✓	✓	✓	✓	✓
Dec 2006	C2H Compiler Out-of-Memory Error	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Array Elements of structs Do Not Copy Correctly	13	✓	✓	✓	✓	✓	✓
Jun 2006	Build Clean Causes Build Failure	14	✓	✓	✓	✓	✓	✓
Jun 2006	Changing Build Configurations Produces Unexpected Results	15	✓	✓	✓	✓	✓	✓
Jun 2006	Closed System Library While Working With the C2H Compiler	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Comma Operators Are Not Supported	15	—	—	—	Fixed	✓	✓
Jun 2006	Hardware Accelerators Remain After Deleting the Software Project *(2)*	16	✓	✓	✓	✓	✓	✓
Jun 2006	Incorrect Results From Logical or Conditional Operation With Side-Effects	16	✓	✓	✓	✓	✓	✓
Jun 2006	Launch SOPC Builder Button in C2H View	17	✓	✓	✓	✓	✓	✓

Table 2. Nios II EDS Errata

Added or Updated	Issue	Page	Affected Version					
			7.2 SP1	7.2	7.1 SP1	7.1	7.0	6.1
Jun 2006	Multiple Clock Domains Cause Hardware Accelerator to Fail	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Simulating on ISS Not Supported	*(1)*	✓	✓	✓	✓	✓	✓
Development Boards								
Jun 2006	Intermittent Failures While Accessing CompactFlash Card	17	✓	✓	✓	✓	✓	✓
Devices								
Jun 2006	Stratix II EP2S60 ES Devices Cannot Use MRAM Byte Enables	*(1)*	✓	✓	✓	✓	✓	✓
Documentation Issues								
Dec 2007	Incomplete newlib Documentation	*(1)*	✓	✓	—	—	—	—
Oct 2007	Nios II IDE Online Help Expand Buttons Do Not Work	18	✓	✓	—	—	—	—
Oct 2007	New Control Register Names Not Listed	18	✓	✓	—	—	—	—
May 2007	Arria GX Support Incorrectly Listed in Documentation HTML Page	19	—	Fixed	✓	✓	—	—
May 2007	Nios II IDE Help Topic Does Not Work With Linux Mozilla	19	✓	✓	✓	✓	—	—
Mar 2007	GNU Documentation is Missing	*(1)*	—	—	—	Fixed	✓	—
Mar 2007	Nios II Processor Reference Handbook Refers Erroneously to the Data Bus	20	—	—	—	Fixed	✓	—
Download Cables & Debug Hardware								
Jun 2006	Communication Errors During Run/Debug Sessions Using Older Download Cables	*(1)*	✓	✓	✓	✓	✓	✓
Hardware Example Designs								
Mar 2007	Incorrect PLL Phase Shift in Stratix II and Cyclone II Designs	20	✓	✓	✓	✓	✓	—
Software Example Designs								
Oct 2007	Hardware Tutorial Software Example Hangs on Some Boards	22	✓	✓	—	—	—	—
May 2007	Dhrystone BSP Example Fails When Run on the Fast Hardware Example	22	—	Fixed	✓	✓	—	—
May 2007	Software Template Error With Non-Ethernet Hardware Design	23	—	Fixed	✓	✓	—	—

Table 2. Nios II EDS Errata

Added or Updated	Issue	Page	Affected Version					
			7.2 SP1	7.2	7.1 SP1	7.1	7.0	6.1
Jun 2006	Networking Examples	24	✓	✓	✓	✓	✓	✓
Jun 2006	RAM Test Failure When Running Memory Test Software Template on the ISS	*(1)*	✓	✓	✓	✓	✓	✓
Flash Programmer								
Oct 2007	elf2flash Warning With Projects That Share a .ptf File	25	—	Fixed	✓	✓	✓	✓
May 2007	Delay When Creating New Flash Programmer Configuration	25	✓	✓	✓	✓	—	—
Dec 2006	Communication Issue With Multiprocessor and Multi-JTAG UART Designs	43	—	—	—	—	Fixed	✓
Dec 2006	elf2flash File Size Limit	26	✓	✓	✓	✓	✓	✓
Dec 2006	"No such file or directory" Error for a Project Stored in a Path Containing Spaces	26	✓	✓	✓	✓	✓	✓
Dec 2006	Flash Programmer Generates Error with Override File	26	—	—	—	Fixed	✓	✓
Dec 2006	Spurious Flash Verify Failure With nios2-flash-programmer	27	—	—	—	Fixed	✓	✓
Dec 2006	Flash Programmer Error with Multiprocessor System	*(1)*	✓	✓	✓	✓	✓	✓
Hardware Simulation								
Dec 2007	Nios II Processor Fails in Simulation	*(1)*	Fixed	✓	✓	—	—	—
Oct 2007	bash: vsim: command not found	28	✓	✓	—	—	—	—
Oct 2007	Simulation Fails Due to Incorrect Port Properties	28	✓	✓	✓	✓	—	—
Dec 2006	Error: "UNC paths are not supported. Defaulting to Windows directory."	28	✓	✓	✓	✓	✓	✓
Jun 2006	ModelSim® Fails to Load Large Memory Models	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Simulation Failure if Reset Address is Set to EPCS	29	—	Fixed	✓	✓	✓	✓
Jun 2006	Uninitialized .bss Variables in Simulation	29	✓	✓	✓	✓	✓	✓
Host Platform								
Dec 2007	Windows/Cygwin: Nios II Processor Generation Failure	30	✓	✓	—	—	—	—
Oct 2007	Windows Vista: Limited Support in Nios II EDS	30	✓	✓	—	—	—	—

Table 2. Nios II EDS Errata

Added or Updated	Issue	Page	Affected Version					
			7.2 SP1	7.2	7.1 SP1	7.1	7.0	6.1
Oct 2007	Linux: Nios II IDE Fails to Find File	31	✓	✓	—	—	—	—
Dec 2006	Windows: Command-Line Utilities Do Not Display Any Output	31	—	—	—	—	Fixed	✓
Dec 2006	Windows XP: Internal Error When Repeatedly Stepping Into With Debugger	32	—	—	—	Fixed	✓	✓
Jun 2006	Linux: Process Fails To Terminate While Debugging With the Nios II ISS Target	32	✓	✓	✓	✓	✓	✓
Jun 2006	Linux: The Quartus® II Stand-Alone Programmer is Not Supported	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Windows: Frisk Antivirus Software Causes SOPC Builder and Nios II Command Shell to Be Unresponsive	*(1)*	✓	✓	✓	✓	✓	✓
Nios II IDE								
Building Projects								
Dec 2007	First Build in the Nios II IDE Takes Several Minutes	33	Fixed	✓	—	—	—	—
Dec 2007	Nios II IDE Command-Line Tools Hang	33	✓	✓	✓	✓	✓	✓
Oct 2007	Read-Only Zip File System Base Address Fixed at 0	34	—	Fixed	✓	✓	—	—
May 2007	Build Command Not Functional for BSPs Created Using the Software Build Flow	34	✓	✓	✓	✓	—	—
May 2007	Linker Errors with Dual-Port Memories	35	✓	✓	✓	✓	—	—
May 2007	User-Managed BSP Settings Not Supported in Nios II IDE	35	✓	✓	✓	✓	—	—
May 2007	Error Message: 'Internal error occurred during "Launching"'	36	—	—	✓	✓	—	—
May 2007	URL Project Location Causes Project Creation Error	36	✓	✓	✓	✓	—	—
Mar 2007	Incorrect Stack Checking When Interrupt Vector Custom Instruction Is Used	37	—	—	—	Fixed	✓	—
Mar 2007	Unable to Save Files Containing Copyright Symbol	37	—	—	—	Fixed	✓	—
Dec 2006	Compilation Error with Separate Exception Stack Option	38	✓	✓	✓	✓	✓	✓
Dec 2006	Illegal Project Location Error When Creating New Project	*(1)*	✓	✓	✓	✓	✓	✓

Table 2. Nios II EDS Errata

Added or Updated	Issue	Page	Affected Version					
			7.2 SP1	7.2	7.1 SP1	7.1	7.0	6.1
Dec 2006	Incorrect Address Assignment for Dual-Port Memory Mastered by Two Different Master Ports	*(1)*	✓	✓	✓	✓	✓	✓
Dec 2006	Incorrect Stack and Heap Free Space Report	38	✓	✓	✓	✓	✓	✓
Dec 2006	Nios II IDE Reports Problems Without Displaying Error in Console	38	✓	✓	✓	✓	✓	✓
Jun 2006	Build Errors After Changing Component Names in SOPC Builder	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Nios II IDE Unnecessarily Updates the SOPC Builder System File (.ptf)	*(1)*	✓	✓	✓	✓	✓	✓
	Debugging Projects							
Oct 2007	Trace Debug Does Not Support the JMPI Instruction	39	✓	✓	—	—	—	—
May 2007	Cannot Locate Source Code in Driver Files Shared by Multiple Projects	39	✓	✓	✓	✓	—	—
May 2007	Console Window Is Not Updated After ISS Error	39	✓	✓	✓	✓	—	—
May 2007	Extra Arguments in Debugger for Multiple Launch Configurations	40	—	Fixed	✓	✓	—	—
May 2007	GDB Error 256 When Launching Debugger	40	—	Fixed	✓	✓	—	—
May 2007	ISS Fails on Designs Containing TSE MAC or SG-DMA Components	41	✓	✓	✓	✓	—	—
May 2007	Memory Window Sets Control Register Values Incorrectly	41	✓	✓	✓	✓	—	—
May 2007	Previous 7.1 Beta Install Might Cause Internal Error When Launching FS2 in IDE	*(1)*	✓	✓	✓	✓	—	—
May 2007	Nios II IDE Debugger Cannot Display Driver Source Code for Custom Component	41	✓	✓	✓	✓	—	—
Dec 2006	“Additional nios2-terminal Arguments” Field Is Not Functional	43	—	—	—	Fixed	✓	✓
Dec 2006	Communication Issue With Multiprocessor and Multi-JTAG UART Designs	43	—	—	—	—	Fixed	✓
Dec 2006	MicroC/OS-II “Source Not Found” Error Occurs at Breakpoint	44	—	—	—	Fixed	✓	✓
Dec 2006	Programs That Interact With a Terminal Console on Windows Do Not Work	45	✓	✓	✓	✓	✓	✓
Dec 2006	“Run as ModelSim” in the Nios II IDE Fails	45	✓	✓	✓	✓	—	—

Table 2. Nios II EDS Errata

Added or Updated	Issue	Page	Affected Version					
			7.2 SP1	7.2	7.1 SP1	7.1	7.0	6.1
Dec 2006	The Restart Command on the Run Menu Does Not Work	45	✓	✓	✓	✓	✓	✓
Jun 2006	Breakpoints on Adjacent Lines of Assembly Fail to Halt the Processor	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Cannot Use Watchpoints in the Nios II IDE When the FS2 Console Is Open	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Debugger Cannot Step Into __sflags(), and Continues Execution Instead	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Incorrect Breakpoint Filtering on Threads	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Missing Address and Data in the Trace View	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	nios2-gdb-server Fails to Terminate After Setting a Watch Point	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	“Step failed. Target is not responding (timed out)” Error Message	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Uninitialized Memory Error When Executing From ISS	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	Watchpoints Do Not Work on Certain Variables	46	✓	✓	✓	✓	✓	✓
	Navigating Projects							
May 2007	C/C++ Search Returns Nothing	46	—	Fixed	✓	✓	—	—
May 2007	Nios II IDE Import System Library Dialog Box Does Not Close After Import	46	✓	✓	✓	✓	—	—
Dec 2006	Nios II IDE Freezes While Displaying the Splash Screen	47	✓	✓	✓	✓	✓	✓
Dec 2006	Internal Error When Double-Clicking on a Large Objdump File	47	✓	✓	✓	✓	✓	✓
Jun 2006	C/C++ Scanner Does Not Support Certain C/C++ Constructs	48	✓	✓	✓	✓	✓	✓
Jun 2006	Nios II IDE Does Not Recognize When an Object File is Built Outside the IDE	48	—	—	—	Fixed	✓	✓
	Nios II Software Build Tools							
Dec 2007	Space in --set Option Value Causes Error	49	Fixed	✓	—	—	—	—
Oct 2007	Software Build Tools Fail if Nios II EDS is Installed in a Path Containing Spaces	49	✓	✓	—	—	—	—
Oct 2007	JTAG Instance Conflict for Multiprocessor Systems Imported Into IDE	49	✓	✓	—	—	—	—

Table 2. Nios II EDS Errata

Added or Updated	Issue	Page	Affected Version					
			7.2 SP1	7.2	7.1 SP1	7.1	7.0	6.1
Oct 2007	system.h File Does Not Reflect Pre-7.1 Custom Instruction Name Changes	50	✓	✓	—	—	—	—
May 2007	“make download-elf” Fails for Stratix II 2S60 RoHS Design Examples	50	—	Fixed	✓	✓	—	—
May 2007	User-Managed BSP Settings Not Supported in Nios II IDE	35	✓	✓	✓	✓	—	—
May 2007	Improper Quote Format in system.h	51	—	Fixed	✓	✓	—	—
May 2007	Upper-Case File Extensions Not Supported	52	✓	✓	✓	✓	—	—
Peripherals								
Dec 2007	Flash Memory (CFI) Component Has Incorrect Default Timing Parameters	53	✓	—	—	—	—	—
Oct 2007	Software Build Errors Due to _classic Suffix	53	—	Fixed	✓	✓	—	—
Oct 2007	Software Build Tools Select Inappropriate Memories	54	—	Fixed	✓	✓	—	—
May 2007	DMA Controller Always Busy in Burst Mode	54	✓	✓	✓	✓	—	—
May 2007	Non System Wide Reset Can Cause Improper Initialization of Mailbox Core	54	✓	✓	✓	✓	—	—
May 2007	Triple Speed Ethernet Designs Require a Manual Reset on Subsequent Software Downloads	55	—	Fixed	✓	✓	—	—
Jun 2006	Using the DEV_CLRn Signal Can Cause JTAG UART Instability After Device-Wide Reset	*(1)*	✓	✓	✓	✓	✓	✓
Target Software								
Oct 2007	System Clock Rollover	55	—	Fixed	✓	✓	✓	✓
Dec 2006	Compiling an Existing MicroC/OS-II Application Generates Errors	56	—	—	—	—	—	✓
Dec 2006	Lightweight IP (lwIP) Failure When Using DHCP	*(1)*	✓	✓	✓	✓	✓	✓
Dec 2006	stdio Does Not Work with MicroC/OS-II and Small C Library	57	✓	✓	✓	✓	✓	✓
Jun 2006	cout From MicroC-OS/II Task Does Not Send Data to stdout	57	✓	✓	✓	✓	✓	✓
Jun 2006	Creating New Custom HAL Components in the Nios II IDE	*(1)*	✓	✓	✓	✓	✓	✓
Jun 2006	malloc(), realloc() Failures With MicroC/OS-II	57	✓	✓	✓	✓	✓	✓

Table 2. Nios II EDS Errata

Added or Updated	Issue	Page	Affected Version					
			7.2 SP1	7.2	7.1 SP1	7.1	7.0	6.1
Jun 2006	Problems Using HAL Drivers With Toshiba Flash	*(1)*	✓	✓	✓	✓	✓	✓
Toolchain (gcc, gdb, etc.)								
Dec 2006	GNU Assembler Does Not Accept the --defsym Flag	58	✓	✓	✓	✓	✓	✓
Jun 2006	Breakpoints in C++ Constructors Fail to Halt the Processor	*(1)*	✓	✓	✓	✓	✓	✓

Notes to Table 2:

(1) This issue is documented in Altera's online Knowledge Database. To see the solution, click on the name of the issue, or search the Knowledge Database at **www.altera.com/support/kdb/kdb-index.jsp**.

(2) For further information, refer to the *Nios II C2H Compiler User Guide*.

C2H Compiler Errata

This section describes in detail the issues related to the Nios II C2H Compiler.

Pre-7.1 Systems Are Not Supported

Starting in v. 7.1, the C2H Compiler does not support SOPC Builder systems created in SOPC Builder 7.0 or earlier (systems based on a **class.ptf** file).

Workaround

Before using the C2H Compiler, you must upgrade your SOPC Builder system. Open the system in SOPC Builder v. 7.1 or later. SOPC Builder will prompt you to convert the system, creating a ***_hw.tcl** file compatible with the C2H Compiler.

Solution Status

Not fixed.

--src-dir Software Build Tools Argument Does Not Work With C2H

When using the C2H Compiler with Nios II Software Build Tools, an error occurs when using the **nios2-app-generate-makefile** argument `--src-dir` to specify a directory of source files. An example of the error is:

```
multiple definition of 'my_accelerated_function'
```

Workaround

Specify source files individually using the `--src-files` argument.

Solution Status

Not fixed.

Accelerator Generation Failure If Tools Are Installed in Path With Spaces

If the path to your installation of the Altera Design Suite contains spaces, the C2H Compiler fails to generate the accelerators.

Workaround

Reinstall the tools to a path containing no spaces.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

The C2H Compiler Regenerates an Accelerator Unnecessarily

The C2H Compiler might regenerate an accelerator even when no logic changes. This problem can result from changes to files included by the C file containing the accelerated function. The C2H Compiler fails to check that the generated hardware description language (HDL) matches the previously generated HDL causing the system to be regenerated.

Workaround

To avoid this issue move the information in the include file specific to the hardware accelerator into a separate include file. This workaround prevents regeneration of the system when the HDL is unchanged.

Solution Status

Not fixed.

Error: c2h_accelerator_base_addresses.h: No such file or directory

When a C2H accelerator is compiled for the first time, the following compile-time error can result if the **Analyze all accelerators** option is selected:

```
c2h_accelerator_base_addresses.h: No such file or directory.
```

Workaround

Click **Build software and generate SOPC Builder system** and build once before building with the **Analyze all accelerators** option.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Java Heap Space Exception if Quartus II Compilation is Enabled

You might receive the following error during Quartus II compilation of your design containing a C2H accelerator if you selected the **Build software, generate SOPC Builder system, and run Quartus II compilation** in the C2H Compiler settings window.

```
    Exception in thread "main"
    java.lang.OutOfMemoryError: Java heap space

    make: *** [c2h_hdl-t] Error 1
```

Workaround

Select the **Build software and generate SOPC builder system** option in the C2H Compiler settings window and launch the Quartus II compilation manually afterward.

Solution Status

Not fixed.

Pointer Dereferences to Volatile Types

The C2H Compiler treats pointer dereferences to a volatile type as if they alias all other pointer dereferences. Pointers that are restrict-qualified are treated the same way.

For example, in the following cases, the two loops cannot be scheduled concurrently because the volatile qualification requires that a dependency exist between the two dereference might cause incorrect behavior.

```
volatile int * restrict fifo_rd = FIFO_RD_BASE;
volatile int * restrict fifo_wr = FIFO_WR_BASE;
for ()
{
 *fifo_wr = ....;
}
for ()
{
 ... = *fifo_rd;
}
```

Workaround

Divide the function into multiple IRQ-enabled accelerators that are launched concurrently from the processor, and use FIFOs to communicate between them.

Solution Status

Not fixed.

C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File

When accelerating a function in a file, the C2H Compiler cannot link subfunctions that are defined in a different file.

Workaround

Include all subfunctions called by the accelerated function within the same source code file.

Solution Status

Not fixed.

Array Elements of structs Do Not Copy Correctly

C2H accelerators do not correctly copy array elements of structs.

For example:

```
typedef struct my_struct {
  int a;
  int b;
  int buf[BUF_SIZE];
}MY_STRUCT;
MY_STRUCT struct_a = {1, 2, {3, 3, 3, 3}};
MY_STRUCT struct_b = {9, 8, {7, 7, 7, 7}};
struct_a = struct_b;
```

In this example, the `a` and `b` elements of the structure copy correctly, but the `buf` element does not. After this assignment, `struct_a` equals `{9, 8, {3, 3, 3, 3}}`.

Workaround

Copy the array elements explicitly, as follows:

```
{
  int i=0;
  do
    {
      struct_a.buf[i] = struct_b.buf[i];
      i++;
    } while (i<LENGTH_OF_BUF_ELEMENT)
}
```

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Build Clean Causes Build Failure

Performing a clean build on a Nios II IDE project that contains a hardware accelerator can cause the next build to fail in the IDE, because the clean build erroneously deletes a file required by the C2H Compiler.

Workaround

Do not perform a clean build on projects that use hardware accelerators. If you have already performed a clean build, recompile with option **Build software, generate SOPC Builder system, and run Quartus II compilation** to regenerate the necessary files.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Changing Build Configurations Produces Unexpected Results

The C2H Compiler does not support multiple build configurations (e.g. Release or Debug) in the Nios II IDE. After creating one or more accelerators in a particular configuration, the C2H Compiler produces undefined results if you switch to a different build configurations and create more accelerators.

Workaround

For a specific SOPC Builder system and Nios II IDE project, specify C2H accelerators in only one build configuration. You can use multiple build configurations, as long as only one configuration specifies C2H Compiler settings.

Solution Status

Not fixed.

Comma Operators Are Not Supported

The C2H Compiler does not support comma operators, such as the following example:

```
for(i = 0, j = 3; i < 10; i++, j++)
{
  /* statements */
}
```

Workaround

You can manually duplicate the same functionality, such as:

```
j = 3;
for(i = 0; i < 10; i++)
{
  /* statements */
  j++;
}
```

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in release 7.1.

Hardware Accelerators Remain After Deleting the Software Project

If a system contains C2H accelerators, deleting the software project that defines the accelerators does not remove the accelerators from the hardware system, and the accelerator logic remains in the SOPC Builder system.

Workaround

To remove an accelerator from a system, delete the accelerator from the C2H view in the Nios II IDE first, and then recompile the software project. The C2H Compiler then removes the accelerator from the SOPC Builder system. Once the compilation is complete then the software application can be deleted from the workspace.

Solution Status

Not fixed.

Incorrect Results From Logical or Conditional Operation With Side-Effects

The C2H Compiler always evaluates both operands of logical (`&&`, `||`) and conditional (`?:`) operators. This is different from expected American National Standards Institute (ANSI) C behavior, where operands are evaluated left-to-right, and unnecessary operands are skipped.

For example, in the expression `(i-- && j--)`, if the value of `i` is zero, ANSI C does not evaluate the right-hand-side (RHS) expression, and `j` is not decremented. By contrast, the C2H Compiler evaluates both sides, decrementing `j`.

The following expressions are other examples that might be affected by this issue:

```
(i-- || j--)

(cond ? i-- : j--)
```

Workaround

Use logical and conditional operations whose operators have no side effects. Operations with side effects include pre- and post- increment and decrement operations (`++`, `--`), memory operations (`*`, `[]`, `.`, `->`), and function calls.

Solution Status

Not fixed.

Launch SOPC Builder Button in C2H View

When the Nios II IDE workspace contains multiple projects with multiple system libraries, the incorrect SOPC Builder system might open when you click **Launch SOPC Builder** in the C2H view.

Workaround

Keep only one system library project open at a time while using the C2H Compiler.

Solution Status

Not fixed.

Development Board Errata

This section describes in detail the Nios II EDS issues related to Nios development boards.

Intermittent Failures While Accessing CompactFlash Card

The Nios II Development Kit version 5.0 and later includes a CompactFlash controller peripheral suitable for interfacing to CompactFlash cards in True IDE mode on Nios development boards. In order for True IDE mode to operate, CompactFlash cards require that the ATASEL_N input be driven to ground during power-up.

The CompactFlash controller peripheral includes a configurable power register used to power-cycle CompactFlash cards in Nios II software through a metal oxide semiconductor field-effect transistor (MOSFET) on the Nios development boards. However, in certain development boards, power to the CompactFlash card does not turn off completely during this power cycle operation. Because of this condition, the CompactFlash might not sample the ATASEL_N pin during the power-cycle operation after FPGA configuration when this pin is driven to ground. Instead, the CompactFlash card might sample the ATASEL_N pin when power is first applied to the development board, when I/O is not yet driven by the FPGA (before FPGA configuration).

Workaround

If you encounter errors with CompactFlash when using the Nios development boards, try one of the following:

- Use a different CompactFlash card. Certain cards are more susceptible to the power-cycling issue than others.
- Modify the Nios development board. This is recommended if you are familiar and comfortable with board-level modifications. Disconnect pin 9 (ATASEL_N) on the CompactFlash socket on your Nios development board and tie this pin to ground.

☞ The CompactFlash socket uses a staggered numbering on the pins (starting from pin 1: 1, 26, 2, 27, ...); refer to the CompactFlash Association specification for right-angle surface-mount connectors for exact specifications on this connector. This modification permanently enables True-IDE mode operation.

Solution Status

Not fixed.

Documentation Errata

This section describes in detail the Nios II EDS documentation issues.

Nios II IDE Online Help Expand Buttons Do Not Work

The expand text arrow and the **Show All** option in the online help do not work in Internet Explorer.

Workaround

In Internet Explorer, carry out the following actions:

1. Click the **Refresh/Show Current Topic** icon (yellow arrows icon at the top of the search pane). The table of contents appears with the selected topic highlighted.

2. Click the highlighted topic to refresh the browser frame. Expanded text now works.

Alternatively, use Firefox 2.0.0.6.

Solution Status

Not fixed.

New Control Register Names Not Listed

New control register names, added to the assembler in v. 7.2, are not documented in the *Nios II Processor Reference Handbook*.

Workaround

The new control register names are as follows:

Control Register	Name
8	`pteaddr`
9	`tlbacc`
10	`tlbmisc`
11	`fstatus`

Solution Status

Not fixed.

Arria GX Support Incorrectly Listed in Documentation HTML Page

In the Nios II 7.1 Documentation Launchpad, the **Example Designs** page incorrectly states that a fast design exists targeting the Arria GX family. Although v. 7.1 and later of the Nios II processor supports the Arria GX family, it does not include a fast design for Arria GX devices.

Workaround

Use an existing fast design targeting a Stratix II device and change the device settings.

For information about device settings, refer to the "Device Page" topic in the Quartus II software online help.

Solution Status

Fixed in v. 7.2.

Nios II IDE Help Topic Does Not Work With Linux Mozilla

The expandable Help text for the **Nios II C/C++ Application** (About Nios II IDE Projects) does not function correctly with Mozilla 1.4.3 under Linux.

Workaround

The missing information is as follows:

A Nios II C/C++ application project contains a C/C++ program, usually including a project's `main()` function. Building a Nios II C/C++ application project results in an executable linkable file (**.elf**) that you can run on target hardware, the Nios II instruction set simulator (ISS), and the ModelSim hardware simulator. A Nios II C/C++ application project depends on a single system library project and might reference functions in a Nios II C/C++ library project.

A Nios II C/C++ application project is a Nios II IDE managed make project. The Nios II IDE creates the necessary makefiles and manages the project for you.

Refer to **www.altera.com/literature/ug/ug_nios2_ide_help.pdf** for complete text of the Nios II help system.

Solution Status

Not fixed.

Nios II Processor Reference Handbook Refers Erroneously to the Data Bus

In the *Implementing the Nios II Processor in SOPC Builder* chapter of the *Nios II Processor Reference Handbook*, under “Caches & Tightly Coupled Memories Tab”, the following text appears under “Instruction Settings”: “Usually you enable bursts on the processor's data bus when processor data is stored in DRAM, and disable bursts when processor data is stored in SRAM.” This sentence should read “Usually you enable bursts on the processor's instruction master port when instructions are stored in DRAM, and disable bursts when instructions are stored in SRAM.”

Solution Status

Fixed in v. 7.1.

Hardware Example Design Errata

This section describes in detail the Nios II EDS hardware example design issues.

Incorrect PLL Phase Shift in Stratix II and Cyclone II Designs

In several example designs targeting the Nios II Cyclone II 2C35 and Nios II Stratix II 2S60 RoHS development boards, the synchronous static random access memory (SSRAM) and legacy dual data rate synchronous dynamic random access memory (DDR SDRAM) clocks are implemented with an incorrect phase-locked loop (PLL) phase shift.

This issue affects the following example designs:

- 2s60 RoHS TSE SG-DMA
- 2s60 RoHS Standard
- 2s60 RoHS Full Featured
- 2c35 TSE SG-DMA
- 2c35 Standard
- 2c35 Full Featured

This issue also affects any custom 2c35 or 2s60 RoHS design using SSRAM or legacy DDR SDRAM.

The incorrect phase shift might cause timing violations when reading from or writing to SSRAM or legacy DDR SDRAM.

Workaround

To correct the PLL settings, perform the following steps:

1. Open the standard design in SOPC Builder.

2. Double-click the component instance named **pll** to launch the PLL MegaWizard interface.

3. Click **Launch Altera's ALTPLL MegaWizard** to launch the MegaWizard interface.

4. Select the **Output Clocks** page and then the clock page used to generate the SSRAM clock. Depending on the particular design, this might be **clk c1**, **clk c2**, or another clock.

5. Ensure that **Clock phase shift** is –3.38 ns.

6. Click the clock page used to generate the write clock for the legacy DDR SDRAM, if any. Depending on the particular design, this might be **clk c1**, **clk c2**, or another clock.

7. Ensure that **Clock phase shift** is 270 degrees.

8. Click **Finish** to save changes and exit the PLL MegaWizard interface.

9. Click **Finish** to save settings to the PLL instance in SOPC Builder.

10. Regenerate the system in SOPC Builder and recompile in the Quartus II software. Refer to the **readme.txt** file in the design example folder for information on compiling designs containing the DDR controller.

For further information refer to **ssram_interface_readme.html**, located in the <*Quartus II installation directory*>**/sopc_builder/documents** folder. This document discusses the SSRAM timing analysis methodology in detail. Additionally, *AN 411: Understanding PLL Timing for Stratix II Devices* discusses clock phase shift calculations and assignments for PLLs in Stratix II devices.

Solution Status

Not fixed.

Software Example Design Errata

This section describes in detail the Nios II EDS software example design issues.

Hardware Tutorial Software Example Hangs on Some Boards

The **count binary** application used in the tutorial might hang when run on some designs that do not contain a liquid crystal display (LCD) peripheral.

Workaround

Modify line 18 of the **count_binary.h** file, as follows:

```
# define LCD_PRINTF(lcd, args...) /* Do Nothing */
```

Rebuild and run the software again.

Solution Status

Not fixed.

Dhrystone BSP Example Fails When Run on the Fast Hardware Example

You might see the following output when running the Dhrystone board support package (BSP) example in the Nios II IDE after creating and building it via command line.

```
Int_Comp:            36608
      should be:     18
```

```
  Str_Comp:          ·"
nios2-terminal: exiting due to ^D on remote
```

This error is caused by settings in the **create-this-bsp** script that cause the application to overrun the amount of memory available in the Fast design.

Workaround

To run the Dhrystone example successfully with the Software Build Tools, edit the **create-this-bsp** script located in the *<hardware example design>***/software_examples/bsp/hal_dhrystone** folder. Carry out the following steps:

1. If the BSP already exists, remove the **settings.bsp** file and run `make clean` in the BSP folder.

2. Change the following line:

```
--set hal.enable_lightweight_device_driver_api true \
```

to:

```
--set hal.enable_reduced_device_drivers true \
```

3. Run the **create-this-bsp** script. (Alternatively, run the **create-this-app** script, found in the Dhrystone application folder.)

The BSP is regenerated with the new setting.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Software Template Error With Non-Ethernet Hardware Design

You might get an internal error if your hardware design does not contain an Ethernet MAC, and you try to create a software example from one of the following templates:

- host_filesystem
- simple_socket_server
- web_server
- zip_filesystem

These templates check for components in the system library and incorrectly test for the existence of a MAC.

Workaround

Create the system library project for your hardware design before creating your application project from one of the templates. In the system library properties, turn on the Host File System feature by carrying out the following steps:

1. Right-click the system library project and select **Properties**.
2. Click the **Software Components** button, located in the bottom left corner of the **System Library Properties** page.
3. Select **Altera Host Based File System.**
4. Turn on **Add this software component**.
5. Accept the default mount point shown in the **Mount-point** box.

When creating your application project, select this system library.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Networking Examples

If you are running a networking example design, you might be asked for a nine-digit number. You are directed to find this number on a sticker on your Nios development board, identified by the prefix **ASJ**. Not all Nios development boards have this sticker.

Workaround

If your Nios development board does not have a sticker with the letters **ASJ** followed by a nine-digit number, enter a unique nine-digit number when prompted. To avoid network address conflicts, ensure that this number is unique to each Nios board connected to your network.

Solution Status

Not fixed.

Flash Programmer Errata

This section describes in detail the Nios II EDS issues related to the flash programmer.

elf2flash Warning With Projects That Share a .ptf File

You might have difficulty burning flash with more than one flash programmer configuration. This issue can arise when switching between projects that share an SOPC Builder configuration file (**.ptf**). A typical example is a multi-processor project.

When you change flash configurations, and try to burn flash, you might see a warning similar to the following:

```
(WARNING) elf2flash: Supplied reset address: 0x0
differs from the elf file reset vector: 0x500000.
```

This happens because the contents of the custom offset fields are improperly carried over from one flash programmer configuration to the next.

Workaround

Upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Delay When Creating New Flash Programmer Configuration

You might experience a delay of several seconds when creating a new Flash Programmer configuration in the Nios II IDE.

Workaround

There is no workaround available at this time.

Solution Status

Not fixed.

Communication Issue With Multiprocessor or Multi-JTAG UART Designs

See page 43.

elf2flash File Size Limit

The **elf2flash** utility supports ELF files up to approximately 24 MBytes in size. The **elf2flash** utility might fail on files larger than 24 MBytes, with the error message `java.lang.OutOfMemoryError`.

Workaround

Lower the number of symbols in your ELF file by turning off debug symbols.

Alternatively, specify less initialized data in the application.

Solution Status

Not fixed.

"No such file or directory" Error for a Project Stored in a Path Containing Spaces

You might receive this spurious error message when programming flash for a project stored in a path containing spaces. The flash programmer does not correctly handle spaces in the directory path. However, this error is benign, because flash programming completes successfully.

Workaround

None required.

Solution Status

Not fixed.

Flash Programmer Generates Error with Override File

Using the Nios II Flash Programmer with an override file stored in <*Nios II EDS path*>**/bin** might generate an error:

```
4 [main] nios2-flash-programmer 4440 _cygtls::handle_exceptions: Error while dumping state
(probably corrupted stack)  Segmentation fault (core dumped)
```

Workaround

Move the override file from the <*Nios II EDS path*>/bin directory to the working directory of your Nios II IDE project. Then, specify the override file to **nios2-flash-programmer** using the `--override=` parameter. In Nios II IDE, you can do this in the **Additional nios2-flash-programmer**

arguments field in the **Flash Programmer** dialog box. At the command line, the override file can be specified by adding the `--override=` argument to the command line, as follows:

```
nios2-flash-programmer --base-0x0 --override=my_fp_override.txt standard.flash
```

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Spurious Flash Verify Failure With nios2-flash-programmer

Using the **nios2-flash-programmer** command line utility to verify flash contents using the `--verify` argument might result in a verify failure even though flash contents are correct. The failure message is similar to the following:

```
Verifying 00000000 ( 0%)Failed to verify at around
00000000  Verify failed
```

Workaround

To work around this issue, avoid using the `--verify` argument with the **nios2-flash-programmer** command-line utility. A verification of flash contents can be done by reading the flash contents into a file using the `--read` argument, then comparing the file to the input file used to program flash.

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Solution Status

Not fixed.

Hardware Simulation Errata

This section describes in detail the Nios II EDS issues related to hardware simulation.

bash: vsim: command not found

You might get this error when you try to launch ModelSim from the Nios II command shell. This can happen if you have a separate Cygwin installation on your machine. Windows cannot find the path to the ModelSim executable.

Workaround

Add the path to your ModelSim installation to your Path variable.

Solution Status

Not fixed.

Simulation Fails Due to Incorrect Port Properties

An issue between SOPC Builder and the Nios II IDE causes custom instruction ports to be assigned as input, std_logic, and with width 1. This causes your system to fail to generate or simulate in ModelSim.

Workaround

After adding your custom instruction, close SOPC Builder and re-launch it.

Solution Status

Not fixed.

Error: "UNC paths are not supported. Defaulting to Windows directory."

If you launch ModelSim from a working directory that is mapped via a universal naming convention (UNC) path (a path that starts with // instead of drive letter), you receive this error message in SOPC Builder. This error occurs because ModelSim is calling a command shell, which does not support UNC paths.

Workaround

Map the UNC path to a drive letter and use the drive letter to reference the working directory in the launching shell.

Solution Status

Not fixed.

Simulation Failure if Reset Address is Set to EPCS

Running ModelSim RTL simulation of a Nios II system fails if the reset address of the Nios II processor is set to an EPCS Serial Flash Controller because there is no simulation model for it.

Workaround

To simulate your system, temporarily set the Reset Address of the Nios II processor to the memory in which your application code resides (for example, SDRAM), then regenerate the system in SOPC Builder and run RTL simulation again. Before booting the Nios II processor from EPCS flash on your target board, change the Nios II Reset Address back to the EPCS Controller peripheral and regenerate the system in SOPC Builder and recompile in the Quartus II software to produce an updated SRAM object file (**.sof**) with the Nios II processor booting from EPCS flash.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Uninitialized .bss Variables in Simulation

If your program reads the value of an uninitialized `.bss` variable during HDL simulation, and the BSP (system library) is compiled with the **ModelSim only, no hardware support** property enabled in Nios II IDE, a warning appears about unfiltered data being `'x'`. This warning appears because when this property is enabled, the code that clears the `.bss` memory region is omitted to speed up HDL simulation so this memory region is uninitialized. The `.bss` region contains global and static local variables that are not initialized by the application so they default to a value of zero. When the Nios II processor reads uninitialized variables, it displays a warning and converts any of the bits of the uninitialized data to zero which correctly mimics the effect of the missing `.bss` clearing code. The hardware abstraction layer (HAL) code that executes before and after `main()` might use `.bss` variables, so these warnings might appear even if your application does not use the `.bss` section.

Solution Status

Not fixed.

Host Platform Errata

This section describes in detail the Nios II EDS issues related to the host development platform.

Windows/Cygwin: Nios II Processor Generation Failure

You might see the following error message when you generate your Nios II system in SOPC Builder:

```
Error: Generator program for module 'cpu_fpoint'
did NOT run successfully
```

This error might occur on a Windows-based system when all of the following conditions are true:

- You have installed a version of Cygwin other than the one distributed with Quartus II.
- You launch Quartus II and SOPC Builder from a Nios II SDK Shell.
- You enable the FPCI in the Nios II processor.

Workaround

Launch Quartus II and SOPC Builder from the Windows Start menu, and regenerate your system.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Windows Vista: Limited Support in Nios II EDS

The Quartus II software introduces Windows Vista (32-bit and 64-bit) support in version 7.2. However, the Nios II Embedded Design Suite supports only the following platforms:

- Windows 2000
- Windows XP
- Windows XP (64)
- Windows Vista Business Edition (32-bit)
- SUSE 9 (32-bit)
- SUSE 9 (64-bit)
- RedHAT Linux v3.0 (32-bit)
- RedHAT Linux v3.0 (64-bit)
- RedHAT Linux v4.0 (32-bit)
- RedHAT Linux v4.0 (64-bit)

Workaround

There is no workaround at this time.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Linux: Nios II IDE Fails to Find File

The Nios II IDE fails to open project files on Linux and reports that the **GCC** compiler is not found.

Workaround

To avoid this issue, make a link to **nios2-elf-gcc** by doing the following:

1. Set `QUARTUS_ROOTDIR` to the path of the Quartus II installation.

2. Run **/altera/72/nios2eds/sdk_shell**

3. Make a link to **nios2-elf-gcc** by executing the following command:

```
ln -s $SOPC_KIT_NIOS2/bin/nios2-gnutools/\
   H-i686-pc-linux-gnu/bin/nios2-elf-gcc /tmp/gcc
```

4. Add this link to the path as follows:

```
export PATH=$PATH:/tmp
```

Solution Status

Not fixed.

Windows: Command-Line Utilities Do Not Display Any Output

Command-line utilities for creating and manipulating Nios II IDE projects on Windows might not display any output, even though the command completed successfully.

The following commands function correctly but do not produce output to `stdout`:

- **nios2-build-project**
- **nios2-create-application-project**
- **nios2-create-system-library**
- **nios2-delete-project**

- **nios2-import-project**

Workaround

Pipe the command output through **tee**. For example:

```
nios2-build-project --help | tee foo.txt
```

Alternatively, upgrade to the Nios II EDS v. 7.0 or later.

Solution Status

Fixed in v. 7.0.

Windows XP: Internal Error When Repeatedly Stepping Into With Debugger

If you repeatedly use the **Step Into** command while debugging on Windows XP, you might receive the following error:

```
Retrieving Children: An internal error occurred
during: Retrieving Children
```

Workaround

End your debug session, then re-open it and resume debugging. Limit your use of the **Step Into** command.

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Linux: Process Fails To Terminate While Debugging With the Nios II ISS Target

If you try to interrupt or terminate a debug session targeting the Nios II instruction set simulator (ISS), you might see an error message **Interrupt Failed or Terminate Failed**. This means that the **nios2-iss** process failed to terminate. The debug session appears to have terminated in the IDE, but the **nios2-iss** process still continues running.

Workaround

Open a command shell and kill the **nios2-iss** process as follows:

1. Type:

```
jobs
```

   to get a list of process IDs.

2. Type:

```
kill -9<nios2-iss process ID>
```

   to terminate the process.

Solution Status

Not fixed.

Nios II IDE Errata: Building Projects

This section describes in detail the issues related to building projects in the Nios II IDE.

First Build in the Nios II IDE Takes Several Minutes

After you install the Nios II EDS, on a 32-bit Windows Vista system, building your first Nios II IDE project might take several minutes. This is due to the time required to index the **altera.components** project.

Workaround

Upgrade to the Nios II EDS v. 7.2 SP1 or later.

Solution Status

Fixed in version 7.2 SP1.

Nios II IDE Command-Line Tools Hang

On Windows systems, the Nios II IDE command-line tools are sometimes unable to locate the project workspace automatically. When this happens, the tools might hang.

This issue potentially affects the following tools:

- nios2-create-system-library
- nios2-create-application-project
- nios2-build-project
- nios2-import-project
- nios2-delete-project

Workaround

Explicitly supply the Nios II IDE command-line tools with a workspace location, by means of the `-data` command-line argument. The syntax of the `-data` argument is as follows:

```
-data <path to workspace>
```

Example:

```
nios2-create-project -data c:/altera/72/nios2eds/\
    bin/eclipse/nios2-ide-workspace-7.2 \
    <other arguments>
```

The path to the workspace must not contain spaces. Aside from this restriction, any valid disk location can be used for the workspace. The workspace shown above is the default workspace used by the Nios II IDE in version 7.2.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Read-Only Zip File System Base Address Fixed at 0

The symbol `ALTERA_RO_ZIPFS_BASE` should be set to the base address of the selected flash memory. In designs created in the Nios II IDE, `ALTERA_RO_ZIPFS_BASE` is hard-coded to 0.

Workaround

Develop your software project using the Nios II software build tools.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Build Command Not Functional for BSPs Created Using the Software Build Flow

The build option in the Nios II IDE menu does not rebuild BSPs imported into the IDE.

Workaround

To build the BSP, build the associated application project.

Solution Status

Not fixed.

Linker Errors with Dual-Port Memories

If your instruction master and data master ports are connected to the same dual-port memory and the ports have different addresses, your code fails to run or you experience a linker error. The Nios II IDE does not warn you of the addressing violation.

Workaround

Assign the same address to both ports of the dual-port memory.

Solution Status

Not fixed.

User-Managed BSP Settings Not Supported in Nios II IDE

For projects created using the new software build flow and imported into the Nios II IDE, the IDE configuration settings have no effect.

For example, objdump, compiler, and linker settings made in the IDE are ignored. This behavior occurs because Nios II software build tools projects are not IDE-managed projects. In addition, the make-related preferences do not pertain to imported Nios II software build tools projects. The IDE ignores these options during the build process.

Workaround

Make these settings in the project's makefile.

Solution Status

Not fixed.

Error Message: ‘Internal error occurred during “Launching”’

If you installed a Beta release of Nios II version 7.1 prior to installing the full release, you might encounter the error **An internal error occurred during "Launching"**, when building, running or debugging a software application project in Nios II IDE.

Workaround

Close Nios II IDE, open a Nios II command shell, and type `nios2-ide-clean`. The Nios II IDE launches again, and the error does not recur. Launching Nios II IDE with the `-clean` option does not affect any IDE or project settings you have made.

Altera strongly recommends that you uninstall Beta versions of the 7.1 tools before installing the released version of the Nios II EDS v. 7.1. Also, remove or rename the existing installation directories.

Solution Status

Fixed in v. 7.2.

URL Project Location Causes Project Creation Error

When you try to create a new project in an existing workspace, you might see an error dialog box saying:

```
Project cannot be created. Reason: Internal Error
```

This error might occur if the path to any project in the workspace is a Uniform Resource Locator (URL) location, for example **file:/F:/Design**. To view the path, right-click on the project and select **Properties**.

Workaround

Import your existing application and system library projects into a new workspace.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Incorrect Stack Checking When Interrupt Vector Custom Instruction Is Used

If you enable run-time stack checking in systems that include the interrupt vector custom instruction, the Nios II IDE might erroneously report a stack overflow or might fail to report an actual stack overflow.

Workaround

Do not enable the run-time stack checking and interrupt vector custom instruction features in the same application.

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Unable to Save Files Containing Copyright Symbol

On computers using a character encoding other than Cp1252, the copyright symbol included in the **board_diag.c** file is not recognized. Because of this, the file can not be saved using the Nios II IDE.

Workaround

Change the default text file encoding setting in the Nios II IDE to Cp1252 by performing the following steps:

1. In the Nios II IDE, select **Preferences** from the **Window** menu.

2. In the **General** settings, select **Workspace.**

3. Select **Default (Cp 1252)** in the **Text File Encoding** section and click **OK**.

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Compilation Error with Separate Exception Stack Option

Choosing the **Use a separate exception stack option** might cause the compilation error "UNDEFINED VARIABLE %STACK_POINTER%" when building a project. This error occurs if the exception stack is larger than the memory available for it.

Workaround

On the system library properties page for the project, turn off the separate exception stack or reduce the **Maximum exception stack size** setting.

Solution Status

Not fixed.

Incorrect Stack and Heap Free Space Report

The makefile reports an incorrect number of bytes free for the stack and heap, if the heap and stack are in different memory regions.

Workaround

No workaround available.

Solution Status

Not fixed.

Nios II IDE Reports Problems Without Displaying Error in Console

When building a project, the Nios II IDE reports problems, but the build output in the console does not contain any errors.

The Nios II IDE incorrectly reports some linker warnings as errors, with a dialog box saying **Errors exist in a required project**. The Dhrystone software example exhibits this behavior, and recompiling the project again makes the issue go away.

Workaround

If the Console output does not contain errors, then the project built correctly. On subsequent builds, the linker step is skipped and the errors do not appear.

Solution Status

Not fixed.

Nios II IDE Errata: Debugging Projects

This section describes in detail the issues related to debugging projects in the Nios II IDE.

Trace Debug Does Not Support the JMPI Instruction

The new JMPI instruction introduced in version 7.2 is not supported by the Nios II trace tools.

Workaround

There is no workaround at this time.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Cannot Locate Source Code in Driver Files Shared by Multiple Projects

If you hit a breakpoint in a driver file, and that driver file is shared with another project that is closed, the Nios II IDE might indicate that it cannot locate the source code.

Workaround

Open the closed system library project and resume debugging.

Solution Status

Not fixed.

Console Window Is Not Updated After ISS Error

After performing a **Run as ISS**, if you receive an ISS error in the console window, the console is not updated subsequently.

Workaround

Close the console window after receiving an ISS error. A new console window opens when a new message is available.

Solution Status

Not fixed.

Extra Arguments in Debugger for Multiple Launch Configurations

If you have multiple debug launch configurations, you might have issues with the **Additional nios2-download arguments** field. If you delete the contents of the **Additional nios2-download arguments** field in one debug launch configuration, view a second launch configuration, with a nonempty string, in the **Additional nios2-download arguments** text box, and return to the first launch configuration, the value from the second configuration appear in the first configuration.

Workaround

To remove all arguments, use a single space rather than deleting the entire contents of the field.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

GDB Error 256 When Launching Debugger

You might see an `Error 256` message when launching the debugger if you have environment variables that are longer than 1200 characters.

Workaround

Ensure that you do not have any environment variable that exceed 1200 characters.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

ISS Fails on Designs Containing TSE MAC or SG-DMA Components

You receive an Internal Error when attempting to perform an ISS simulation of designs containing the Altera Triple Speed Ethernet (TSE) MAC or scatter-gather direct memory access (SG-DMA) components because the Nios II ISS does not support these components.

Workaround

Remove the TSE MAC and SG-DMA components from your system and perform ISS simulation on the simplified system. You can also simulate the design in ModelSim® or test it on hardware.

Solution Status

Not fixed.

Memory Window Sets Control Register Values Incorrectly

The memory window might incorrectly set values in memory-mapped control registers. For example, writing 0x1234 to a byte addressed register results in the value 0x3434 being stored in the register. The memory window shows this incorrect value.

Workaround

Use the GNU debugger (GDB) console window in the IDE, instead of the memory window, to write to the registers. For example, type:

```
set {int} <register address>=0x1234↵
```

in the GDB console window.

☞ You must refresh the memory window in order for it to correctly display the target value.

Solution Status

Not fixed.

Nios II IDE Debugger Cannot Display Driver Source Code for Custom Component

When debugging a custom component in the Nios II IDE, the debugger might not find the software driver files for that component.

Workaround

You must add a linked folder to an IDE project so that the IDE can find source code files associated with the component. The locations which you must add appear in a generated file called **install2.ptf** in the directory containing the **install.ptf** for the hardware design. Examine this file to find the directory path which you must add. For example, a custom component called **altera_avalon_pwm** is stored in **/data/pwm_component_test**. An excerpt from **install2.ptf** follows:

```
PACKAGE install2
{
   COMPONENT altera_avalon_pwm
   {
      VERSION 2.0
      {
         local = "/data/pwm_component_test";
      }
   }
}
```

To add the path **/data/pwm_component_test** to the IDE, carry out the following steps:

1. Right-click the system library project for your design.

2. Point to **New** > **Other** > **General** > **Folder**, and click **Next**.

3. Specify the parent project (e.g. **syslib**) in the **Enter or select the parent folder** field.

4. Click **Advanced**, turn on **Link to folder in the file system**, and browse to or type in the path to the component.

5. Click **Finish** to link the folder.

☞ Do not link a custom component to more than one system library in the IDE. The breakpoint manager sometimes fails to handle the aliases, which results in breakpoints being hit even when removed from the breakpoint manager.

Solution Status

Not fixed.

“Additional nios2-terminal Arguments” Field Is Not Functional

On the **Target Connection** tab of the **Run** or **Debug** launch configuration dialog box, the **Additional nios2-terminal arguments** field is not functional. Entering commands in these fields has no effect.

The **Additional nios2-download arguments** and **Additional nios2-flash-programmer arguments** fields are functional.

Workaround

Upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Communication Issue With Multiprocessor and Multi-JTAG UART Designs

Designs that contain multiple processors or multiple joint test action group universal asynchronous receiver-transmitters (JTAG UARTs) might not be able to communicate with the target when running, debugging, or programming flash memory. Designs which have only a single processor and JTAG UART are not affected by this issue.

Workaround

You need to explicitly specify the JTAG instance number for the target. Perform the following steps:

1. Open the **Run** or **Debug** launch configuration dialog box and click the Main tab.

2. In the **Target Hardware** section, enter `--instance=` in the **Additional nios2-download arguments** field to supply the JTAG instance information to the debugger.

For designs with a single processor and JTAG UART, the default instance number N is zero and does not need to be specified as an extra argument.

Alternatively, upgrade to the Nios II EDS v. 7.0 or later.

Solution Status

Fixed in v. 7.0.

MicroC/OS-II “Source Not Found” Error Occurs at Breakpoint

You might receive a 'Source not found' error message if you step into MicroC/OS-II code or break on a program entry point while debugging your application.

Workaround

To eliminate this error, you must specify the proper source path mapping in the Nios II IDE by performing the following steps.

1. On the Windows menu, click **Preferences**. The **Preferences** dialog box appears.

2. Expand the **C/C++** group, then expand the **Debug** group, then click **Common Source Lookup Path**.

3. Click **Add...**. The **Add Source** dialog box appears.

4. Select **Path Mapping**, then click **OK**.

5. In the **Preferences** dialog box, select **Path Mapping: New Mapping** and click **Edit**. The **Path Mappings** dialog box appears.

6. Enter `cygdrive` in the **Name** field.

7. Click **Add...**. A dialog box appears, prompting you to specify the path mapping.

8. Enter `/cygdrive/c` in the **Compilation path** field and `c:/` in the **Local file system path** field.

9. Click **OK** in all open dialog boxes to return to the IDE workbench.

This process sets up the source path mapping for the entire workspace, which is recommended. You can also create a similar mapping for individual projects by clicking **Edit Source Lookup Path** that appears below the **Source not found** error message and following steps 4 through 8.

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Programs That Interact With a Terminal Console on Windows Do Not Work

Programs with this behavior work in v. 6.0 and earlier, but do not work in Nios II IDE v. 6.1 and later.

The Eclipse platform in v. 6.1 and later of the IDE (on Windows only) sends the string `\r\n` instead of just `\n` when running or debugging using the Terminal. This behavior can break existing software designs, and it is inconsistent with **nios2-terminal**, which still just sends `\n`.

Workaround

Change the software running on the Nios II processor to parse for `\r\n` as well as `\n`.

Solution Status

Not fixed.

“Run as ModelSim” in the Nios II IDE Fails

The **Run as ModelSim** command might fail on launch configurations created in v. 7.0 or earlier of the IDE. This problem does not occur for new launch configurations.

Workaround

Select a location for the ModelSim tool from the launch configuration dialog box. You can use the **Browse** button next to the **ModelSim path** group, or type in a path to the ModelSim directory (e.g. **c:/altera/71/modelsim_ae/win32aloem**).

Solution Status

Not fixed.

The Restart Command on the Run Menu Does Not Work

The **Restart** command on the **Run** menu does not work.

Workaround

Stop the program, then debug it again. If the debugger is hung in an endless loop, use the following **bash** alias to break the target, then stop it:

```
alias break="kill -2 \’ps -a | grep nios2-elf-gdb | cut -f6 -d' '\’"
```

Solution Status

Not fixed.

Watchpoints Do Not Work on Certain Variables

Watchpoints do not work on a variable whose size is not 32 bits.

Workaround

Change the types of global and static local variables to `int`, `long`, or `unsigned long` before setting watchpoints on them.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Nios II IDE Errata: Navigating Projects

This section describes in detail the issues related to navigating projects in the Nios II IDE.

C/C++ Search Returns Nothing

The C/C++ Search in the Nios II IDE does not return results because a search index is not created at start up.

Workaround

Manually invoke indexing for projects you need to search. Refer to the "Indexing and Searching Project Content" IDE online help topic for instructions.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Nios II IDE Import System Library Dialog Box Does Not Close After Import

When importing a system library project, the Nios II IDE wizard does not close after clicking the **Finish** button. The wizard imports the project successfully after clicking **Finish** once. Clicking **Finish** again results in an error message stating that the project is already created.

Workaround

Click **Cancel** in the **Import Wizard** dialog box after clicking **Finish**.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Nios II IDE Freezes While Displaying the Splash Screen

After clicking **Switch Workspace** on the File menu on a Windows machine, a Nios II IDE splash screen appears. Unfortunately, this splash screen obscures a dialog box which asks you to specify the new workspace. This makes the IDE appear to freeze.

Workaround

Press Alt-Tab to switch applications. Two relevant application icons appear: an Eclipse icon associated with the splash screen and a Nios II IDE icon associated with the **Workspace** dialog box. Select the Nios II icon to bring the dialog box to the foreground.

Solution Status

Not fixed.

Internal Error When Double-Clicking on a Large Objdump File

On Windows when opening a large **objdump** file in the Nios II IDE, you might get the following error message:

```
Unable to create this part due to an internal error.
Reason for the failure: Editor could not be initialized.
```

Workaround

Adjust the Windows launch arguments for the Nios II IDE editor. Perform the following steps:

1. On the Windows Start menu, browse to the **Nios II EDS** program icon, right-click it, then click **Properties**. The **Windows Properties** dialog box appears.

2. In the **Target** field, append "`vmargs -Xmx1024m`" to the end of the path to the Nios II IDE executable. For example:

```
C:\altera\72\nios2eds\bin\eclipse\nios2-ide.exe -vmargs -Xmx1024m
```

Solution Status

Not fixed.

C/C++ Scanner Does Not Support Certain C/C++ Constructs

The C/C++ scanner performs C/C++ Search, navigation, open declaration, and parts of content assist. Due to limitations of the C/C++ scanner, these features do not work with C code constructs not supported by the C++ language. An example is functions that take a function pointer as an argument.

Workaround

If the C/C++ Search fails, use the File Search facility.

Solution Status

Not fixed.

Nios II IDE Does Not Recognize When an Object File is Built Outside the IDE

If you use a build process external to the Nios II IDE, the IDE might not recognize when you have generated or re-generated the ELF file for your project. In this case, the IDE debugger might report an error that the ELF file does not exist.

Workaround

Right-click on the project name and click **Refresh** to allow the IDE to recognize the ELF file.

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Nios II Software Build Tools Errata

This section describes in detail the issues in the Nios II software build tools.

Space in --set Option Value Causes Error

When using **nios2-bsp-create-settings** or **nios2-bsp-update-settings**, or the **nios2-bsp** script, you might receive the following error message if you include spaces in the `--set` option value:

```
SEVERE: nios2-bsp-create-settings failed.
SEVERE: CommandLineErrorException: The following
option has the incorrect number of arguments
"--set".
```

Workaround

Enclose the setting value in double quotes. For example:

```
--set hal.make.BSP_CFLAGS_USER_FLAGS=\
   "-O1 -ffunction-sections -fverbose-asm \
   -fno-inline"
```

Alternatively, upgrade to the Nios II EDS v. 7.2 SP1 or later.

Solution Status

Fixed in version 7.2 SP1.

Software Build Tools Fail if Nios II EDS is Installed in a Path Containing Spaces

The Nios II Command Line Software Build Tools fail if you install the Nios II EDS in a path containing spaces.

Workaround

Re-install the Nios II EDS into a path that does not contain spaces.

Solution Status

Not fixed.

JTAG Instance Conflict for Multiprocessor Systems Imported Into IDE

When importing a multiprocessor project created using the Software Build Tools into the Nios II IDE, the instance IDs for the Nios II processor might be incorrect, causing JTAG errors during debugging.

Workaround

In the Nios II IDE **Debug** launch configuration, inspect each Nios II processor's instance number in the **Additional nios2-download arguments** field. Manually change the instance ID of each processor to the correct value. The correct instance ID is located in the <*SOPC Builder system name*>**.jdi** file. Search for the processor name. The instance ID is located in the "`node hpath`" line for that processor instance.

☞ The Nios II IDE allows you to set the instance ID only for the currently selected processor. To correct the instance ID on another processor, first select it.

Solution Status

Not fixed.

system.h File Does Not Reflect Pre-7.1 Custom Instruction Name Changes

If you instantiate a **class.ptf**-based custom instruction (created in SOPC Builder v. 7.0 or earlier) or upgrade a design containing a **class.ptf**-based custom instruction, subsequent custom instruction name changes are lost when you close and re-launch SOPC Editor. The **system.h** file generated by the Software Build Tools does not reflect the name change.

Workaround

Manually change the name in the **system.h** file.

Solution Status

Not fixed.

"make download-elf" Fails for Stratix II 2S60 RoHS Design Examples

You might see error messages when running the 2S60 RoHS example applications with the Software Build Tools. These messages appear due to an incorrect system ID value in the BSP makefile. The error messages are similar to the following:

```
Reading System ID at address 0x021208B8:
    ID value does not match: read 0x072DA598; expected 0x7D1BD35D
    Timestamp value was not verified: value was not specified
The software you are downloading may not run on the system which is
currently configured into the device.  Please download the correct SOF
```

```
or recompile
```

Workaround

Remove the system ID check from the application makefile. To remove the system ID check, carry out the following steps:

1. Open the application **Makefile,** located in the application project directory.

2. Search for "`download-elf`" to find the section that needs modification.

3. Remove the `$(SOPC_SYSID_FLAG)` argument from the `$(DOWNLOAD)` command in the following section:

```
.PHONY : download-elf
download-elf : $(ELF)
      @echo Info: Downloading $(ELF)
      $(DOWNLOAD) --go --cpu_name=$(CPU_NAME) $(SOPC_SYSID_FLAG)
$(WRITE_GMON_OPTION) $(ELF)
```

The resulting section looks like:

```
.PHONY : download-elf
download-elf : $(ELF)
      @echo Info: Downloading $(ELF)
      $(DOWNLOAD) --go --cpu_name=$(CPU_NAME) $(WRITE_GMON_OPTION) $(ELF
```

Rebuild the application.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

User-Managed BSP Settings Not Supported in Nios II IDE

See page 35.

Improper Quote Format in system.h

The Nios II software build tools tcl command

```
add_sw_setting character system_h_define character_system_h \
    character_system_h a "character_system_h"
```

generates a **system.h** file with contents in double quotes (i.e. `"a"`) instead of the proper single quotes (i.e. `'a'`).

Workaround

Manually edit the **system.h** file to fix the appropriate line by changing double quotes to single quotes.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Upper-Case File Extensions Not Supported

The Nios II software build tools for applications and libraries (**nios2-app-generate-makefile** and **nios2-lib-generate-makefile** commands) do not support source files with certain upper-case extensions. If a file with an upper-case extension is included, the make command stops with no descriptive warning.

Only Nios II assembly language files built by the C preprocessor can have upper-case file extension (**.S**). All C language files must have the extension **.c** or **.h**. C++ language source files must have the extension **.cpp**, **.cxx**, **.cc,** or **.h**.

Workaround

Rename all C language files with the extension **.c** or **.h**. Rename all C++ language files with the extension **.cpp**, **.cxx**, **.cc,** or **.h**.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Peripheral Errata

This section describes in detail the Nios II EDS issues related to peripheral components.

Flash Memory (CFI) Component Has Incorrect Default Timing Parameters

The Intel P30 presets for the **Flash Memory (CFI)** component in SOPC Builder contain slightly incorrect timing parameters. There is a small chance that these timing parameters might cause the Nios II Embedded Evaluation Kit (NEEK) on-board flash memory to function improperly with some designs.

Workaround

Instead of using the Intel 128P30 or 256P30 presets for the **Flash Memory (CFI)** component, use the options shown in Table 3.

Table 3. CFI Component Options for NEEK

Option Name	Option Value	
	128P30	256P30
Presets	Custom	Custom
Address Width (bits)	23	24
Data Width (bits)	16	16
Setup	25	25
Wait	100	100
Hold	20	20
Units	ns	ns

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Software Build Errors Due to _classic Suffix

This issue arises when you integrate a device driver for “classic” SOPC Builder IP, that is, IP described by a **class.ptf** file, rather than the newer **_hw.tcl** script. In this situation, the Nios II Software Build Tools append “_classic” to the class name that you specify for your custom component. The tools make these modifications, for example, to macros in **system.h** and to header file names in **system.h** and **alt_sys_init.c**. The modified names cause software build errors.

Workaround

Upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Software Build Tools Select Inappropriate Memories

When selecting default memory devices for memory regions, the Nios II software build tools sometimes select a memory that is not mastered by an appropriate master port on the Nios II IDE.

Workaround:

Upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

DMA Controller Always Busy in Burst Mode

The DMA controller component (**altera_avalon_dma**), when enabled for burst transactions, does not perform transfers at widths other than its full data width. The DMA controller is always busy.

Workaround

When bursting is enabled, the DMA controller must be programmed to perform transactions at its full data width.

Solution Status

Not fixed.

Non System Wide Reset Can Cause Improper Initialization of Mailbox Core

The **altera_avalon_mailbox** peripheral might not be initialized properly when a soft (non-system-wide) reset occurs. Examples: the `resetrequest` signal resets a specific Nios II processor that masters the mailbox; a processor mastering the mailbox restarts execution from the start address. In this condition, mailbox contents (read and write pointers) are not reinitialized and might show potentially stale data.

Workaround

Ensure that a system-wide reset event occurs by asserting the `reset_n` input to the SOPC Builder system containing the mailbox. This resets all peripherals and Nios II processors in the system. Alternatively, ensure that any messages posted to the mailbox are cleared before issuing a soft reset.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Triple Speed Ethernet Designs Require a Manual Reset on Subsequent Software Downloads

You might experience issues in designs containing the Altera Triple Speed Ethernet MegaCore function. These issues might prevent the TSE MAC from operating properly after repeated downloads of the Nios II processor's ELF file. When this bug occurs, even though the Nios II processor executes the correct initialization sequence to reset the Triple Speed Ethernet component, a complete reset never occurs. The result is that the TSE MAC is unable to transmit or receive Ethernet data.

Workaround

Before downloading the ELF file to the Nios II processor, push the board's reset button. This action resets the Altera Triple Speed Ethernet MAC into a known-good state.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Target Software Errata

This section describes in detail the Nios II EDS issues related to target software components.

System Clock Rollover

The `gettimeofday()` function in the Nios II HAL can experience clock rollover. After a few thousand system clock ticks, the count of microseconds since system startup overflows, resulting in an incorrect function return value.

Workaround

Upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Compiling an Existing MicroC/OS-II Application Generates Errors

The version of MicroC/OS-II was upgraded in the v. 6.1 release, which can cause this issue. After importing a Nios II IDE project that uses MicroC/OS-II created with version 6.0 SP1 or earlier, building the project in the Nios II IDE generates errors, such as

```
#error "OS_CFG.H, Missing OS_FLAGS_NBITS: Determine
#bits used for event flags, MUST be either 8, 16 or 32'
```

Workaround

Resetting the MicroC/OS-II software component settings to their defaults fixes this issue. To reset the component settings, carry out the following steps:

1. Right-click on the application project in the Nios II C/C++ view.

2. Click **System Library Properties**.

3. Click **RTOS Options...** to display the **MicroC/OS-II RTOS Options** dialog box.

4. Expand the **MicroC/OS-II** group in the upper-left-hand corner of the dialog box.

5. Examine each page of options, and make note of your settings.

6. Click **Restore Defaults.**

7. Re-examine each page of options, and re-enter any settings that changed.

8. Click **OK** to dismiss the **MicroC/OS-II RTOS Options** dialog box.

9. Click **OK** again to dismiss the Properties dialog box.

Solution Status

This issue only affects upgrades from v. 6.0 SP1 or earlier.

stdio Does Not Work with MicroC/OS-II and Small C Library

`stdin`, `stdout`, and `stderr` do not work in MicroC/OS-II applications built with the Small C library option.

Workaround

Disable the small C library option.

Solution Status

Not fixed.

cout From MicroC-OS/II Task Does Not Send Data to stdout

If neither `printf()` or `cout` is used from `main()` before tasks are started, `cout` does not work from a task.

Workaround

Add the following C++ code to the beginning of `main()`:

```
std::ios_base::sync_with_stdio(false);
```

Solution Status

Not fixed.

malloc(), realloc() Failures With MicroC/OS-II

When you use the MicroC/OS-II real-time operating system (RTOS), calls to `malloc()` and `realloc()` might fail if successive calls to `malloc()` or `realloc()` within a MicroC/OS-II task occur after changing the task priority of the task in which a memory block is originally allocated.

Workaround

Use one of the following workarounds:

- Allocate and/or reallocate memory blocks outside of MicroC/OS-II tasks, before task switching starts. This makes it possible to change thread priorities at runtime.

- Allocate fixed areas of memory using arrays (rather than using `malloc()`) before task switching starts. This makes it possible to change thread priorities at runtime.
- Allocate memory using `malloc()` or `realloc()` from a MicroC/OS-II task. You can change task priorities at runtime, but only for tasks that have not used `malloc()` or `realloc()`.

Solution Status

Not fixed.

Toolchain Errata

This section describes in detail the Nios II EDS issues related to the GNU compiler toolchain (gcc, gdb, etc.).

GNU Assembler Does Not Accept the --defsym Flag

According the GNU documentation, you can set an assembler definition by using the `--defsym` flag, but it does not work in the following form: `--defsym MY_VAR=1`

Workaround

There is no workaround available at this time.

Solution Status

Not fixed.

How to Contact Altera

For the most up-to-date information about Altera® products, refer to the following table.

Contact (1)	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Product literature	Website	www.altera.com/literature
Altera literature services	Email	literature@altera.com
Non-technical support (General)	Email	nacomp@altera.com
(Software Licensing)	Email	authorization@altera.com

Note:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

The following table shows the typographic conventions that this document uses.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Command names, dialog box titles, checkbox options, and dialog box options are shown in bold, initial capital letters. Example: **Save As** dialog box.
bold type	External timing parameters, directory names, project names, disk drive names, file names, file name extensions, and software utility names are shown in bold type. Examples: $\mathbf{f_{MAX}}$, **\qdesigns** directory, **d:** drive, **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Document titles are shown in italic type with initial capital letters. Example: *AN 75: High-Speed Board Design*.
Italic type	Internal timing parameters and variables are shown in italic type. Examples: t_{PIA}, $n + 1$. Variable names are enclosed in angle brackets (< >) and shown in italic type. Example: *<file name>*, *<project name>***.pof** file.
Initial Capital Letters	Keyboard keys and menu names are shown with initial capital letters. Examples: Delete key, the Options menu.
"Subheading Title"	References to sections within a document and titles of on-line help topics are shown in quotation marks. Example: "Typographic Conventions."

Visual Cue	Meaning
`Courier type`	Signal and port names are shown in lowercase Courier type. Examples: `data1`, `tdi`, `input`. Active-low signals are denoted by suffix `n`, e.g., `resetn`. Anything that must be typed exactly as it appears is shown in Courier type. For example: `c:\qdesigns\tutorial\chiptrip.gdf`. Also, sections of an actual file, such as a Report File, references to parts of files (e.g., the AHDL keyword `SUBDESIGN`), as well as logic function names (e.g., `TRI`) are shown in Courier.
1., 2., 3., and a., b., c., etc.	Numbered steps are used in a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ● •	Bullets are used in a list of items when the sequence of the items is not important.
✓	The checkmark indicates a procedure that consists of one step only.
☞	The hand points to information that requires special attention.
⚠ CAUTION	A caution calls attention to a condition or possible situation that can damage or destroy the product or the user's work.
⚡ WARNING	A warning calls attention to a condition or possible situation that can cause injury to the user.
↵	The angled arrow indicates you should press the Enter key.
👣	The feet direct you to more information on a particular topic.

Nios II Embedded Design Suite Release Notes and Errata

RN-EDS-1.0

About These Release Notes

These release notes cover versions 6.1, 7.0, 7.1, and 7.2 of the Altera® Nios® II Embedded Design Suite (EDS). These release notes describe the new features and errata for the Nios II EDS.

☞ From v7.2 onwards, the contents of the Nios II release notes and errata sheet that Altera formerly published with each software release are superseded by two documents:

- *Nios II Embedded Design Suite Release Notes and Errata* (this document)
- The *Nios II Processor* chapter of the *MegaCore® IP Library Release Notes and Errata*.

Errata are functional defects or errors, which might cause the product to deviate from published specifications. Documentation issues include errors, unclear descriptions, or omissions from current published specifications or product documents.

For the most up-to-date errata for this release, refer to the latest version of the *Nios II Embedded Design Suite Release Notes and Errata* on the Altera website.

New Features

Table 21–1 shows the new features history for the Nios II EDS, including the Nios II C2H Compiler.

For more information on the new features, refer to the *Nios II Processor Reference Handbook*, the *Nios II Software Developer's Handbook*, and the *Nios II C2H Compiler User Guide*. For information about changes to the Nios II processor core, refer to the *MegaCore IP Library Release Notes and Errata*.

Table 21–1. Nios II Embedded Design Suite New Features History

Version	Date	Description
7.2	October 2007	● Nios II software build tools: Library support. ● Nios II software build tools: Support for pre-compiled libraries (.a files) as device drivers. ● Nios II software build tools: Recompiles **newlib** when new compiler flags introduced. ● Nios II software build tools: command line options to specify component search paths ● Nios II GNU compiler: targets floating point hardware when performing conversion (to and from integer) and comparison operations. ● Nios II Integrated Development Environment (IDE): Lauterbach instruction set simulator (ISS) debug option. ● Nios II IDE: **altera.components** project shortens project creation time. ● NicheStack TCP/IP Stack – Nios II Edition: Simplified licensing mechanism. ● NicheStack TCP/IP Stack – Nios II Edition: Increased DHCP timeout to 130 seconds. ● Triple Speed Ethernet MegaCore function hardware design example: Includes tightly-coupled memories; other changes. Refer to the Nios II Documentation Launchpad for details. ● Nios II processor and ECPS Flash Controller: Cyclone® III boot-loader support. ● Stratix® III and Cyclone III Fast hardware example design removed. Designs targeting those device families are available with Stratix III and Cyclone III development kits. ● Legacy SDK utilities **nios2-build** and **nios2-run** removed.
7.1	May 2007	● Nios II Software Build Tools design flow ● Arria™ GX device family support ● Design examples including new SOPC Builder components ● Altera Triple Speed Ethernet (TSE) Media Access Controller (MAC) MegaCore ● Altera Scatter-Gather Direct Memory Access (SG-DMA) Controller
7.0	March 2007	● Cyclone III device family support ● Instruction master burst support
6.1	December 2006	● NicheStack TCP/IP Stack - Nios II Edition ● Interrupt vector custom instruction ● Nios II integrated development environment (IDE) based on the latest version of the Eclipse platform ● Memory footprint reduction options

Errata Summary

Table 21–2 summarizes the issues that affect the Nios II Embedded Design Suite version 7.2, 7.1, 7.0 and 6.1. For a detailed description of each errata issue, click on the issue name.

Table 21–2. Nios II EDS Errata

Added or Updated	Issue	Page	Affected Version			
			7.2	7.1	7.0	6.1
	C2H Compiler					
May 2007	Accelerator Generation Failure If Tools Are Installed in Path With Spaces	21–9	✓	✓	—	—
Jun 2006	Array Elements of structs Do Not Copy Correctly	21–9	✓	✓	✓	✓
Jun 2006	Build Clean Causes Build Failure	21–10	✓	✓	✓	✓
May 2007	C2H Accelerators Time Out Immediately When Untethered	*(1)*	✓	✓	—	—
Dec 2006	C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File	21–10	✓	✓	✓	✓
Dec 2006	C2H Compiler Out-of-Memory Error	*(1)*	✓	✓	✓	✓
May 2007	The C2H Compiler Regenerates an Accelerator Unnecessarily	21–10	✓	✓	—	—
Jun 2006	Changing Build Configurations Produces Unexpected Results	21–11	✓	✓	✓	✓
Jun 2006	Closed System Library While Working With the C2H Compiler	*(1)*	✓	✓	✓	✓
Jun 2006	Comma Operators Are Not Supported	21–11	—	Fixed	✓	✓
May 2007	Error: c2h_accelerator_base_addresses.h: No such file or directory	21–12	✓	✓	—	—
Jun 2006	Hardware Accelerators Remain After Deleting the Software Project *(2)*	21–12	✓	✓	✓	✓
Jun 2006	Incorrect Results From Logical or Conditional Operation With Side-Effects	21–13	✓	✓	✓	✓
May 2007	Java Heap Space Exception if Quartus II Compilation is Enabled	21–13	✓	✓	—	—
Jun 2006	Launch SOPC Builder Button in C2H View	21–14	✓	✓	✓	✓
Jun 2006	Multiple Clock Domains Cause Hardware Accelerator to Fail	*(1)*	✓	✓	✓	✓
May 2007	Pointer Dereferences to Volatile Types	21–14	✓	✓	—	—
Jun 2006	Simulating on ISS Not Supported	*(1)*	✓	✓	✓	✓
Oct 2007	Pre-7.1 Systems Are Not Supported	21–15	✓	—	—	—
Oct 2007	--src-dir Software Build Tools Argument Does Not Work With C2H	21–15	✓	—	—	—

Table 21–2. Nios II EDS Errata

Added or Updated	Issue	Page	Affected Version			
			7.2	7.1	7.0	6.1
Development Boards						
Jun 2006	Intermittent Failures While Accessing CompactFlash Card	21–16	✓	✓	✓	✓
Devices						
Jun 2006	Stratix II EP2S60 ES Devices Cannot Use MRAM Byte Enables	*(1)*	✓	✓	✓	✓
Documentation Issues						
May 2007	Arria GX Support Incorrectly Listed in Documentation HTML Page	21–17	Fixed	✓	—	—
Dec 2006	GNU Assembler Does Not Accept the --defsym Flag	21–17	✓	✓	✓	✓
Mar 2007	GNU Documentation is Missing	*(1)*	—	Fixed	✓	—
May 2007	Nios II IDE Help Topic Does Not Work With Linux Mozilla	21–17	✓	✓	—	—
Mar 2007	Nios II Processor Reference Handbook Refers Erroneously to the Data Bus	21–18	—	Fixed	✓	—
Oct 2007	Nios II IDE Online Help Expand Buttons Do Not Work	21–18	✓	—	—	—
Oct 2007	New Control Register Names Not Listed	21–19	✓	—	—	—
Download Cables & Debug Hardware						
Jun 2006	Communication Errors During Run/Debug Sessions Using Older Download Cables	*(1)*	✓	✓	✓	✓
Hardware Example Designs						
Mar 2007	Incorrect PLL Phase Shift in Stratix II and Cyclone II Designs	21–19	✓	✓	✓	—
Software Example Designs						
May 2007	Dhrystone BSP Example Fails When Run on the Fast Hardware Example	21–21	Fixed	✓	—	—
Jun 2006	Networking Examples	21–24	✓	✓	✓	✓
Jun 2006	RAM Test Failure When Running Memory Test Software Template on the ISS	*(1)*	✓	✓	✓	✓
May 2007	Software Template Error With Non-Ethernet Hardware Design	21–22	Fixed	✓	—	—
Oct 2007	Hardware Tutorial Software Example Hangs on Some Boards	21–23	✓	—	—	—
Flash Programmer						
Dec 2006	Communication Issue With Multiprocessor or Multi-JTAG UART Designs	21–23	—	—	Fixed	✓

Table 21–2. Nios II EDS Errata

Added or Updated	Issue	Page	Affected Version			
			7.2	7.1	7.0	6.1
May 2007	Delay When Creating New Flash Programmer Configuration	21–23	✓	✓	—	—
Dec 2006	elf2flash File Size Limit	21–23	✓	✓	✓	✓
Dec 2006	“No such file or directory” Error for a Project Stored in a Path Containing Spaces	21–24	✓	✓	✓	✓
Dec 2006	Flash Programmer Generates Error with Override File	21–25	—	Fixed	✓	✓
Dec 2006	Spurious Flash Verify Failure With nios2-flash-programmer	21–25	—	Fixed	✓	✓
Dec 2006	Flash Programmer Error with Multiprocessor System	*(1)*	✓	✓	✓	✓
Oct 2007	elf2flash Warning With Projects That Share a .ptf File	21–26	Fixed	✓	✓	✓
Hardware Simulation						
Dec 2006	Error: “UNC paths are not supported. Defaulting to Windows directory.”	21–26	✓	✓	✓	✓
Jun 2006	*ModelSim® Fails to Load Large Memory Models*	*(1)*	✓	✓	✓	✓
Jun 2006	Simulation Failure if Reset Address is Set to EPCS	21–27	Fixed	✓	✓	✓
Jun 2006	Uninitialized .bss Variables in Simulation	21–28	✓	✓	✓	✓
Oct 2007	bash: vsim: command not found	21–27	✓	—	—	—
Oct 2007	Simulation Fails Due to Incorrect Port Properties	21–28	✓	✓	—	—
Host Platform						
Dec 2006	Command-Line Utilities Do Not Display Any Output	21–28	—	—	Fixed	✓
Jun 2006	Linux: Process Fails To Terminate While Debugging With the Nios II ISS Target	21–29	✓	✓	✓	✓
Jun 2006	Linux: The Quartus® II Stand-Alone Programmer is Not Supported	*(1)*	✓	✓	✓	✓
Dec 2006	Windows XP: Internal Error When Repeatedly Stepping Into With Debugger	21–30	—	Fixed	✓	✓
Jun 2006	Windows: Frisk Antivirus Software Causes SOPC Builder and Nios II Command Shell to Be Unresponsive	*(1)*	✓	✓	✓	✓
Oct 2007	Nios II EDS Does Not Support Windows Vista	21–30	✓	—	—	—
Oct 2007	Nios II IDE Fails to Find File on Linux	21–30	✓	—	—	—
Nios II IDE						
Building Projects						
Jun 2006	Build Errors After Changing Component Names in SOPC Builder	*(1)*	✓	✓	✓	✓
May 2007	Build Command Not Functional for BSPs Created Using the Software Build Flow	21–31	✓	✓	—	—

Table 21–2. Nios II EDS Errata

Added or Updated	Issue	Page	Affected Version			
			7.2	7.1	7.0	6.1
Dec 2006	Compilation Error with Separate Exception Stack Option	21–31	✓	✓	✓	✓
Dec 2006	Illegal Project Location Error When Creating New Project	*(1)*	✓	✓	✓	✓
May 2007	Linker Errors with Dual-Port Memories	21–32	✓	✓	—	—
Dec 2006	Incorrect Address Assignment for Dual-Port Memory Mastered by Two Different Master Ports	*(1)*	✓	✓	✓	✓
Dec 2006	Incorrect Stack and Heap Free Space Report	21–32	✓	✓	✓	✓
Mar 2007	Incorrect Stack Checking When Interrupt Vector Custom Instruction Is Used	21–32	—	Fixed	✓	—
Jun 2006	Nios II IDE Unnecessarily Updates the SOPC Builder System File (.ptf)	*(1)*	✓	✓	✓	✓
May 2007	User-Managed BSP Settings Not Supported in Nios II IDE	21–33	✓	✓	—	—
May 2007	Error Message: ‘Internal error occurred during “Launching”’	21–33	—	✓	—	—
Mar 2007	Unable to Save Files Containing Copyright Symbol	21–34	—	Fixed	✓	—
May 2007	URL Project Location Causes Project Creation Error	21–34	✓	✓	—	—
Dec 2006	Nios II IDE Reports Problems Without Displaying Error in Console	21–35	✓	✓	✓	✓
Oct 2007	Read-Only Zip File System Base Address Fixed at 0	21–35	Fixed	✓	—	—
	Debugging Projects					
Dec 2006	“Additional nios2-terminal Arguments” Field Is Not Functional	21–35	—	Fixed	✓	✓
Jun 2006	Breakpoints on Adjacent Lines of Assembly Fail to Halt the Processor	*(1)*	✓	✓	✓	✓
May 2007	Cannot Locate Source Code in Driver Files Shared by Multiple Projects	21–36	✓	✓	—	—
Jun 2006	Cannot Use Watchpoints in the Nios II IDE When the FS2 Console Is Open	*(1)*	✓	✓	✓	✓
Dec 2006	Communication Issue With Multiprocessor and Multi-JTAG UART Designs	21–36	—	—	Fixed	✓
May 2007	Console Window Is Not Updated After ISS Error	21–37	✓	✓	—	—
Jun 2006	Debugger Cannot Step Into __sflags(), and Continues Execution Instead	*(1)*	✓	✓	✓	✓
May 2007	Extra Arguments in Debugger for Multiple Launch Configurations	21–37	Fixed	✓	—	—
May 2007	GDB Error 256 When Launching Debugger	21–38	Fixed	✓	—	—
Jun 2006	Incorrect Breakpoint Filtering on Threads	*(1)*	✓	✓	✓	✓

Table 21–2. Nios II EDS Errata

Added or Updated	Issue	Page	Affected Version			
			7.2	7.1	7.0	6.1
May 2007	ISS Fails on Designs Containing TSE MAC or SG-DMA Components	21–38	✓	✓	—	—
May 2007	Memory Window Sets Control Register Values Incorrectly	21–38	✓	✓	—	—
Dec 2006	MicroC/OS-II “Source Not Found” Error Occurs at Breakpoint	21–39	—	Fixed	✓	✓
Jun 2006	Missing Address and Data in the Trace View	*(1)*	✓	✓	✓	✓
Jun 2006	nios2-gdb-server Fails to Terminate After Setting a Watch Point	*(1)*	✓	✓	✓	✓
May 2007	Previous 7.1 Beta Install Might Cause Internal Error When Launching FS2 in IDE	*(1)*	✓	✓	—	—
Dec 2006	Programs That Interact With a Terminal Console on Windows Do Not Work	21–40	✓	✓	✓	✓
Dec 2006	“Run as ModelSim” in the Nios II IDE Fails	21–40	✓	✓	—	—
Jun 2006	“Step failed. Target is not responding (timed out)” Error Message	*(1)*	✓	✓	✓	✓
Dec 2006	The Restart Command on the Run Menu Does Not Work	21–41	✓	✓	✓	✓
May 2007	Nios II IDE Debugger Cannot Display Driver Source Code for Custom Component	21–41	✓	✓	—	—
Jun 2006	Uninitialized Memory Error When Executing From ISS	*(1)*	✓	✓	✓	✓
Jun 2006	Watchpoints Do Not Work on Certain Variables	21–42	✓	✓	✓	✓
Oct 2007	Trace Debug Does Not Support the JMPI Instruction	21–42	✓	—	—	—
	Navigating Projects					
Dec 2006	Nios II IDE Freezes While Displaying the Splash Screen	21–43	✓	✓	✓	✓
Jun 2006	C/C++ Scanner Does Not Support Certain C/C++ Constructs	21–43	✓	✓	✓	✓
May 2007	C/C++ Search Returns Nothing	21–43	Fixed	✓	—	—
Dec 2006	Internal Error When Double-Clicking on a Large Objdump File	21–44	✓	✓	✓	✓
Jun 2006	Nios II IDE Does Not Recognize When an Object File is Built Outside the IDE	21–44	—	Fixed	✓	✓
May 2007	Nios II IDE Import System Library Dialog Box Does Not Close After Import	21–45	✓	✓	—	—
	Nios II Software Build Tools					
May 2007	“make download-elf” Fails for Stratix II 2S60 RoHS Design Examples	21–45	Fixed	✓	—	—
May 2007	User-Managed BSP Settings Not Supported in Nios II IDE	21–46	✓	✓	—	—

Table 21–2. Nios II EDS Errata

Added or Updated	Issue	Page	Affected Version			
			7.2	7.1	7.0	6.1
May 2007	Improper Quote Format in system.h	21–46	Fixed	✓	—	—
May 2007	Upper-Case File Extensions Not Supported	21–47	✓	✓	—	—
Oct 2007	Software Build Tools Fail if Nios II EDS is Installed in a Path Containing Spaces	21–47	✓	—	—	—
Oct 2007	JTAG Instance Clash for Multiprocessor Systems Imported Into IDE	21–48	✓	—	—	—
Oct 2007	system.h File Does Not Reflect Pre-7.1 Custom Instruction Name Changes	21–48	✓	—	—	—
Peripherals						
May 2007	DMA Controller Always Busy in Burst Mode	21–49	✓	✓	—	—
May 2007	Non System Wide Reset Can Cause Improper Initialization of Mailbox Core	21–49	✓	✓	—	—
May 2007	Triple Speed Ethernet Designs Require a Manual Reset on Subsequent Software Downloads	21–49	Fixed	✓	—	—
Jun 2006	Using the DEV_CLRn Signal Can Cause JTAG UART Instability After Device-Wide Reset	*(1)*	✓	✓	✓	✓
Oct 2007	Software Build Errors Due to _classic Suffix	21–50	Fixed	✓	—	—
Oct 2007	Software Build Tools Select Inappropriate Memories	21–50	Fixed	✓	—	—
Target Software						
Dec 2006	Compiling an Existing MicroC/OS-II Application Generates Errors	21–51	—	—	—	✓
Jun 2006	cout From MicroC-OS/II Task Does Not Send Data to stdout	21–52	✓	✓	✓	✓
Jun 2006	Creating New Custom HAL Components in the Nios II IDE	*(1)*	✓	✓	✓	✓
Dec 2006	Lightweight IP (lwIP) Failure When Using DHCP	*(1)*	✓	✓	✓	✓
Jun 2006	malloc(), realloc() Failures With MicroC/OS-II	21–52	✓	✓	✓	✓
Jun 2006	Problems Using HAL Drivers With Toshiba Flash	*(1)*	✓	✓	✓	✓
Dec 2006	stdio Does Not Work with MicroC/OS-II and Small C Library	21–52	✓	✓	✓	✓
Oct 2007	System Clock Rollover	21–53	Fixed	✓	✓	✓
Toolchain (gcc, gdb, etc.)						
Jun 2006	Breakpoints in C++ Constructors Fail to Halt the Processor	*(1)*	✓	✓	✓	✓

Notes to Table 21–2:

(1) This issue has been moved from the Errata to Altera's online Knowledge Database. To see the solution, click on the name of the issue, or search the Knowledge Database at **www.altera.com/support/kdb/kdb-index.jsp**.

(2) For further information, refer to the *Nios II C2H Compiler User Guide*.

C2H Compiler Errata

This section describes in detail the issues related to the Nios II C2H Compiler.

Accelerator Generation Failure If Tools Are Installed in Path With Spaces

If the path to your installation of the Altera Design Suite contains spaces, the C2H Compiler fails to generate the accelerators.

Workaround

Reinstall the tools to a path containing no spaces.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Array Elements of structs Do Not Copy Correctly

C2H accelerators do not correctly copy array elements of structs.

For example:

```
typedef struct my_struct {
  int a;
  int b;
  int buf[BUF_SIZE];
}MY_STRUCT;
MY_STRUCT struct_a = {1, 2, {3, 3, 3, 3}};
MY_STRUCT struct_b = {9, 8, {7, 7, 7, 7}};
struct_a = struct_b;
```

In this example, the `a` and `b` elements of the structure copy correctly, but the `buf` element does not. After this assignment, `struct_a` equals `{9, 8, {3, 3, 3, 3}}`.

Workaround

Copy the array elements explicitly, as follows:

```
{
  int i=0;
  do
    {
      struct_a.buf[i] = struct_b.buf[i];
      i++;
    } while (i<LENGTH_OF_BUF_ELEMENT)
}
```

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Build Clean Causes Build Failure

Performing a clean build on a Nios II IDE project that contains a hardware accelerator can cause the next build to fail in the IDE, because the clean build erroneously deletes a file required by the C2H Compiler.

Workaround

Do not perform a clean build on projects that use hardware accelerators. If you have already performed a clean build, recompile with option **Build software, generate SOPC Builder system, and run Quartus II compilation** to regenerate the necessary files.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File

When accelerating a function in a file, the C2H Compiler cannot link subfunctions that are defined in a different file.

Workaround

Include all subfunctions called by the accelerated function within the same source code file.

Solution Status

Not fixed.

The C2H Compiler Regenerates an Accelerator Unnecessarily

The C2H Compiler might regenerate an accelerator even when no logic changes. This problem can result from changes to files included by the C file containing the accelerated function. The C2H Compiler fails to check that the generated hardware description language (HDL) matches the previously generated HDL causing the system to be regenerated.

Workaround

To avoid this issue move the information in the include file specific to the hardware accelerator into a separate include file. This workaround prevents regeneration of the system when the HDL is unchanged.

Solution Status

Not fixed.

Changing Build Configurations Produces Unexpected Results

The C2H Compiler does not support multiple build configurations (e.g. Release or Debug) in the Nios II IDE. After creating one or more accelerators in a particular configuration, the C2H Compiler produces undefined results if you switch to a different build configurations and create more accelerators.

Workaround

For a specific SOPC Builder system and Nios II IDE project, specify C2H accelerators in only one build configuration. You can use multiple build configurations, as long as only one configuration specifies C2H Compiler settings.

Solution Status

Not fixed.

Comma Operators Are Not Supported

The C2H Compiler does not support comma operators, such as the following example:

```
for(i = 0, j = 3; i < 10; i++, j++)
{
  /* statements */
}
```

Workaround

You can manually duplicate the same functionality, such as:

```
j = 3;
for(i = 0; i < 10; i++)
{
  /* statements */
  j++;
}
```

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in release 7.1.

Error: c2h_accelerator_base_addresses.h: No such file or directory

When a C2H accelerator is compiled for the first time, the following compile-time error can result if the **Analyze all accelerators** option is selected:

```
c2h_accelerator_base_addresses.h: No such file or directory.
```

Workaround

Click **Build software and generate SOPC Builder system** and build once before building with the **Analyze all accelerators** option.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Hardware Accelerators Remain After Deleting the Software Project

If a system contains C2H accelerators, deleting the software project that defines the accelerators does not remove the accelerators from the hardware system, and the accelerator logic remains in the SOPC Builder system.

Workaround

To remove an accelerator from a system, delete the accelerator from the C2H view in the Nios II IDE first, and then recompile the software project. The C2H Compiler then removes the accelerator from the SOPC Builder system. Once the compilation is complete then the software application can be deleted from the workspace.

Solution Status

Not fixed.

Incorrect Results From Logical or Conditional Operation With Side-Effects

The C2H Compiler always evaluates both operands of logical (`&&`, `||`) and conditional (`?:`) operators. This is different from expected American National Standards Institute (ANSI) C behavior, where operands are evaluated left-to-right, and unnecessary operands are skipped.

For example, in the expression `(i-- && j--)`, if the value of `i` is zero, ANSI C does not evaluate the right-hand-side (RHS) expression, and `j` is not decremented. By contrast, the C2H Compiler evaluates both sides, decrementing `j`.

The following expressions are other examples that might be affected by this issue:

```
(i-- || j--)

(cond ? i-- : j--)
```

Workaround

Use logical and conditional operations whose operators have no side effects. Operations with side effects include pre- and post- increment and decrement operations (`++`, `--`), memory operations (`*`, `[]`, `.`, `->`), and function calls.

Solution Status

Not fixed.

Java Heap Space Exception if Quartus II Compilation is Enabled

You might receive the following error during Quartus II compilation of your design containing a C2H accelerator if you selected the **Build software, generate SOPC Builder system, and run Quartus II compilation** in the C2H Compiler settings window.

```
Exception in thread "main"
java.lang.OutOfMemoryError: Java heap space

make: *** [c2h_hdl-t] Error 1
```

Workaround

Select the **Build software and generate SOPC builder system** option in the C2H Compiler settings window and launch the Quartus II compilation manually afterward.

Solution Status

Not fixed.

Launch SOPC Builder Button in C2H View

When the Nios II IDE workspace contains multiple projects with multiple system libraries, the incorrect SOPC Builder system might open when you click **Launch SOPC Builder** in the C2H view.

Workaround

Keep only one system library project open at a time while using the C2H Compiler.

Solution Status

Not fixed.

Pointer Dereferences to Volatile Types

The C2H Compiler treats pointer dereferences to a volatile type as if they alias all other pointer dereferences. Pointers that are restrict-qualified are treated the same way.

For example, in the following cases, the two loops cannot be scheduled concurrently because the volatile qualification requires that a dependency exist between the two dereference might cause incorrect behavior.

```
volatile int * restrict fifo_rd = FIFO_RD_BASE;
volatile int * restrict fifo_wr = FIFO_WR_BASE;
for ()
{
 *fifo_wr = ....;
}
for ()
{
 ... = *fifo_rd;
}
```

Workaround

Divide the function into multiple IRQ-enabled accelerators that are launched concurrently from the processor, and use FIFOs to communicate between them.

Solution Status

Not fixed.

Pre-7.1 Systems Are Not Supported

Starting in v. 7.1, the C2H Compiler does not support SOPC Builder systems created in SOPC Builder 7.0 or earlier (systems based on a **class.ptf** file).

Workaround

Before using the C2H Compiler, you must upgrade your SOPC Builder system. Open the system in SOPC Builder v. 7.1 or later. SOPC Builder will prompt you to convert the system, creating a ***_hw.tcl** file compatible with the C2H Compiler.

Solution Status

Not fixed.

--src-dir Software Build Tools Argument Does Not Work With C2H

When using the C2H Compiler with Nios II Software Build Tools, an error occurs when using the **nios2-app-generate-makefile** argument `--src-dir` to specify a directory of source files. An example of the error is:

```
multiple definition of 'my_accelerated_function'
```

Workaround

Specify source files individually using the `--src-files` argument.

Solution Status

Not fixed.

Development Boards Errata

This section describes in detail the Nios II EDS issues related to Nios development boards.

Intermittent Failures While Accessing CompactFlash Card

The Nios II Development Kit version 5.0 and later includes a CompactFlash controller peripheral suitable for interfacing to CompactFlash cards in True IDE mode on Nios development boards. In order for True IDE mode to operate, CompactFlash cards require that the ATASEL_N input be driven to ground during power-up.

The CompactFlash controller peripheral includes a configurable power register used to power-cycle CompactFlash cards in Nios II software through a metal oxide semiconductor field-effect transistor (MOSFET) on the Nios development boards. However, in certain development boards, power to the CompactFlash card does not turn off completely during this power cycle operation. Because of this condition, the CompactFlash might not sample the ATASEL_N pin during the power-cycle operation after FPGA configuration when this pin is driven to ground. Instead, the CompactFlash card might sample the ATASEL_N pin when power is first applied to the development board, when I/O is not yet driven by the FPGA (before FPGA configuration).

Workaround

If you encounter errors with CompactFlash when using the Nios development boards, try one of the following:

- Use a different CompactFlash card. Certain cards are more susceptible to the power-cycling issue than others.
- Modify the Nios development board. This is recommended if you are familiar and comfortable with board-level modifications. Disconnect pin 9 (ATASEL_N) on the CompactFlash socket on your Nios development board and tie this pin to ground.

☞ The CompactFlash socket uses a staggered numbering on the pins (starting from pin 1: 1, 26, 2, 27, ...); refer to the CompactFlash Association specification for right-angle surface-mount connectors for exact specifications on this connector. This modification permanently enables True-IDE mode operation.

Solution Status

Not fixed.

Documentation Errata

This section describes in detail the Nios II EDS documentation issues.

Arria GX Support Incorrectly Listed in Documentation HTML Page

In the Nios II 7.1 Documentation Launchpad, the **Example Designs** page incorrectly states that a fast design exists targeting the Arria GX family. Although v. 7.1 and later of the Nios II processor supports the Arria GX family, it does not include a fast design for Arria GX devices.

Workaround

Use an existing fast design targeting a Stratix II device and change the device settings.

For information about device settings, refer to the "Device Page" topic in the Quartus II software online help.

Solution Status

Fixed in v. 7.2.

GNU Assembler Does Not Accept the --defsym Flag

According the GNU documentation, you can set an assembler definition by using the `--defsym` flag, but it does not work in the following form:

```
--defsym MY_VAR=1
```

Solution Status

Not fixed.

Nios II IDE Help Topic Does Not Work With Linux Mozilla

The expandable Help text for the **Nios II C/C++ Application** (About Nios II IDE Projects) does not function correctly with Mozilla 1.4.3 under Linux.

Workaround

The missing information is as follows:

> A Nios II C/C++ application project contains a C/C++ program, usually including a project's `main()` function. Building a Nios II C/C++ application project results in an executable linkable file (**.elf**) that you can run on target hardware, the Nios II instruction set

simulator (ISS), and the ModelSim hardware simulator. A Nios II C/C++ application project depends on a single system library project and might reference functions in a Nios II C/C++ library project.

A Nios II C/C++ application project is a Nios II IDE managed make project. The Nios II IDE creates the necessary makefiles and manages the project for you.

Refer to **www.altera.com/literature/ug/ug_nios2_ide_help.pdf** for complete text of the Nios II help system.

Solution Status

Not fixed.

Nios II Processor Reference Handbook Refers Erroneously to the Data Bus

In the *Implementing the Nios II Processor in SOPC Builder* chapter of the *Nios II Processor Reference Handbook*, under “Caches & Tightly Coupled Memories Tab”, the following text appears under “Instruction Settings”: “Usually you enable bursts on the processor's data bus when processor data is stored in DRAM, and disable bursts when processor data is stored in SRAM.” This sentence should read “Usually you enable bursts on the processor's instruction master port when instructions are stored in DRAM, and disable bursts when instructions are stored in SRAM.”

Solution Status

Fixed in v. 7.1.

Nios II IDE Online Help Expand Buttons Do Not Work

The expand text arrow and the **Show All** option in the online help do not work in Internet Explorer.

Workaround

In Internet Explorer, carry out the following actions:

1. Click the **Refresh/Show Current Topic** icon (yellow arrows icon at the top of the search pane). The table of contents appears with the selected topic highlighted.

2. Click the highlighted topic to refresh the browser frame. Expanded text now works.

Alternatively, use Firefox 2.0.0.6.

Solution Status

Not fixed.

New Control Register Names Not Listed

New control register names, added to the assembler in v. 7.2, are not documented in the *Nios II Processor Reference Handbook*.

Workaround

The new control register names are as follows:

Control Register	Name
8	`pteaddr`
9	`tlbacc`
10	`tlbmisc`
11	`fstatus`

Solution Status

Not fixed.

Hardware Example Design Errata

This section describes in detail the Nios II EDS hardware example design issues.

Incorrect PLL Phase Shift in Stratix II and Cyclone II Designs

In several example designs targeting the Nios II Cyclone II 2C35 and Nios II Stratix II 2S60 RoHS development boards, the synchronous static random access memory (SSRAM) and legacy dual data rate synchronous dynamic random access memory (DDR SDRAM) clocks are implemented with an incorrect phase-locked loop (PLL) phase shift.

This issue affects the following example designs:

- 2s60 RoHS TSE SG-DMA
- 2s60 RoHS Standard
- 2s60 RoHS Full Featured
- 2c35 TSE SG-DMA
- 2c35 Standard
- 2c35 Full Featured

This issue also affects any custom 2c35 or 2s60 RoHS design using SSRAM or legacy DDR SDRAM.

The incorrect phase shift might cause timing violations when reading from or writing to SSRAM or legacy DDR SDRAM.

Workaround

To correct the PLL settings, perform the following steps:

1. Open the standard design in SOPC Builder.

2. Double-click the component instance named **pll** to launch the PLL MegaWizard interface.

3. Click **Launch Altera's ALTPLL MegaWizard** to launch the MegaWizard interface.

4. Select the **Output Clocks** page and then the clock page used to generate the SSRAM clock. Depending on the particular design, this might be **clk c1**, **clk c2**, or another clock.

5. Ensure that **Clock phase shift** is –3.38 ns.

6. Click the clock page used to generate the write clock for the legacy DDR SDRAM, if any. Depending on the particular design, this might be **clk c1**, **clk c2**, or another clock.

7. Ensure that **Clock phase shift** is 270 degrees.

8. Click **Finish** to save changes and exit the PLL MegaWizard interface.

9. Click **Finish** to save settings to the PLL instance in SOPC Builder.

10. Regenerate the system in SOPC Builder and recompile in the Quartus II software. Refer to the **readme.txt** file in the design example folder for information on compiling designs containing the DDR controller.

For further information refer to **ssram_interface_readme.html**, located in the <*Quartus II installation directory*>**/sopc_builder/documents** folder. This document discusses the SSRAM timing analysis methodology in detail. Additionally, *AN 411: Understanding PLL Timing for Stratix II Devices* discusses clock phase shift calculations and assignments for PLLs in Stratix II devices.

Solution Status

Not fixed.

Software Example Design Errata

This section describes in detail the Nios II EDS software example design issues.

Dhrystone BSP Example Fails When Run on the Fast Hardware Example

You might see the following output when running the Dhrystone board support package (BSP) example in the Nios II IDE after creating and building it via command line.

```
  Int_Comp:           36608
        should be:    18
  Str_Comp:           ·"
nios2-terminal: exiting due to ^D on remote
```

This error is caused by settings in the **create-this-bsp** script that cause the application to overrun the amount of memory available in the Fast design.

Workaround

To run the Dhrystone example successfully with the Software Build Tools, edit the **create-this-bsp** script located in the *<hardware example design>*/**software_examples/bsp/hal_dhrystone** folder. Carry out the following steps:

1. If the BSP already exists, remove the **settings.bsp** file and run `make clean` in the BSP folder.

2. Change the following line:

```
--set hal.enable_lightweight_device_driver_api true \
```

    to:

```
--set hal.enable_reduced_device_drivers true \
```

3. Run the **create-this-bsp** script. (Alternatively, run the **create-this-app** script, found in the Dhrystone application folder.)

The BSP is regenerated with the new setting.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Software Template Error With Non-Ethernet Hardware Design

You might get an internal error if your hardware design does not contain an Ethernet MAC, and you try to create a software example from one of the following templates:

- host_filesystem
- simple_socket_server
- web_server
- zip_filesystem

These templates check for components in the system library and incorrectly test for the existence of a MAC.

Workaround

Create the system library project for your hardware design before creating your application project from one of the templates. In the system library properties, turn on the Host File System feature by carrying out the following steps:

1. Right-click the system library project and select **Properties**.

2. Click the **Software Components** button, located in the bottom left corner of the **System Library Properties** page.

3. Select **Altera Host Based File System.**

4. Turn on **Add this software component**.

5. Accept the default mount point shown in the **Mount-point** box.

When creating your application project, select this system library.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Hardware Tutorial Software Example Hangs on Some Boards

The **count binary** application used in the tutorial might hang when run on some designs that do not contain a liquid crystal display (LCD) peripheral.

Workaround

Modify line 18 of the **count_binary.h** file, as follows:

```
    # define LCD_PRINTF(lcd, args...) /* Do Nothing */
```

Rebuild and run the software again.

Solution Status

Not fixed.

Flash Programmer Errata

This section describes in detail the Nios II EDS issues related to the flash programmer.

Communication Issue With Multiprocessor or Multi-JTAG UART Designs

See page 21–36.

Delay When Creating New Flash Programmer Configuration

You might experience a delay of several seconds when creating a new Flash Programmer configuration in the Nios II IDE.

Workaround

There is no workaround available at this time.

Solution Status

Not fixed.

elf2flash File Size Limit

The **elf2flash** utility supports ELF files up to approximately 24 MBytes in size. The **elf2flash** utility might fail on files larger than 24 MBytes, with the error message `java.lang.OutOfMemoryError`.

Workaround

Lower the number of symbols in your ELF file by turning off debug symbols.

Alternatively, specify less initialized data in the application.

Solution Status

Not fixed.

Networking Examples

If you are running a networking example design, you might be asked for a nine-digit number. You are directed to find this number on a sticker on your Nios development board, identified by the prefix **ASJ**. Not all Nios development boards have this sticker.

Workaround

If your Nios development board does not have a sticker with the letters **ASJ** followed by a nine-digit number, enter a unique nine-digit number when prompted. To avoid network address conflicts, ensure that this number is unique to each Nios board connected to your network.

Solution Status

Not fixed.

"No such file or directory" Error for a Project Stored in a Path Containing Spaces

You might receive this spurious error message when programming flash for a project stored in a path containing spaces. The flash programmer does not correctly handle spaces in the directory path. However, this error is benign, because flash programming completes successfully.

Workaround

None required.

Solution Status

Not fixed.

Flash Programmer Generates Error with Override File

Using the Nios II Flash Programmer with an override file stored in <*Nios II EDS path*>**/bin** might generate an error:

```
4 [main] nios2-flash-programmer 4440 _cygtls::handle_exceptions: Error while dumping state
(probably corrupted stack)  Segmentation fault (core dumped)
```

Workaround

Move the override file from the <*Nios II EDS path*>/bin directory to the working directory of your Nios II IDE project. Then, specify the override file to **nios2-flash-programmer** using the `--override=` parameter. In Nios II IDE, you can do this in the **Additional nios2-flash-programmer arguments** field in the **Flash Programmer** dialog box. At the command line, the override file can be specified by adding the `--override=` argument to the command line, as follows:

```
nios2-flash-programmer --base-0x0 --override=my_fp_override.txt standard.flash
```

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Spurious Flash Verify Failure With nios2-flash-programmer

Using the **nios2-flash-programmer** command line utility to verify flash contents using the `--verify` argument might result in a verify failure even though flash contents are correct. The failure message is similar to the following:

```
Verifying 00000000 ( 0%)Failed to verify at around
00000000  Verify failed
```

Workaround

To work around this issue, avoid using the `--verify` argument with the **nios2-flash-programmer** command-line utility. A verification of flash contents can be done by reading the flash contents into a file using the `--read` argument, then comparing the file to the input file used to program flash.

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Solution Status

Not fixed.

elf2flash Warning With Projects That Share a .ptf File

You might have difficulty burning flash with more than one flash programmer configuration. This issue can arise when switching between projects that share an SOPC Builder configuration file (**.ptf**). A typical example is a multi-processor project.

When you change flash configurations, and try to burn flash, you might see a warning similar to the following:

```
(WARNING) elf2flash: Supplied reset address: 0x0
differs from the elf file reset vector: 0x500000.
```

This happens because the contents of the custom offset fields are improperly carried over from one flash programmer configuration to the next.

Workaround

Upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Hardware Simulation Errata

This section describes in detail the Nios II EDS issues related to hardware simulation.

Error: "UNC paths are not supported. Defaulting to Windows directory."

If you launch ModelSim from a working directory that is mapped via a universal naming convention (UNC) path (a path that starts with // instead of drive letter), you receive this error message in SOPC Builder. This error occurs because ModelSim is calling a command shell, which does not support UNC paths.

Workaround

Map the UNC path to a drive letter and use the drive letter to reference the working directory in the launching shell.

Solution Status

Not fixed.

Simulation Failure if Reset Address is Set to EPCS

Running ModelSim RTL simulation of a Nios II system fails if the reset address of the Nios II processor is set to an EPCS Serial Flash Controller because there is no simulation model for it.

Workaround

To simulate your system, temporarily set the Reset Address of the Nios II processor to the memory in which your application code resides (for example, SDRAM), then regenerate the system in SOPC Builder and run RTL simulation again. Before booting the Nios II processor from EPCS flash on your target board, change the Nios II Reset Address back to the EPCS Controller peripheral and regenerate the system in SOPC Builder and recompile in the Quartus II software to produce an updated SRAM object file (**.sof**) with the Nios II processor booting from EPCS flash.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

bash: vsim: command not found

You might get this error when you try to launch ModelSim from the Nios II command shell. This can happen if you have a separate Cygwin installation on your machine. Windows cannot find the path to the ModelSim executable.

Workaround

Add the path to your ModelSim installation to your Path variable.

Solution Status

Not fixed.

Simulation Fails Due to Incorrect Port Properties

An issue between SOPC Builder and the Nios II IDE causes custom instruction ports to be assigned as input, std_logic, and with width 1. This causes your system to fail to generate or simulate in ModelSim.

Workaround

After adding your custom instruction, close SOPC Builder and re-launch it.

Solution Status

Not fixed.

Host Platform Errata

This section describes in detail the Nios II EDS issues related to the host development platform.

Uninitialized .bss Variables in Simulation

If your program reads the value of an uninitialized `.bss` variable during HDL simulation, and the BSP (system library) is compiled with the **ModelSim only, no hardware support** property enabled in Nios II IDE, a warning appears about unfiltered data being `'x'`. This warning appears because when this property is enabled, the code that clears the `.bss` memory region is omitted to speed up HDL simulation so this memory region is uninitialized. The `.bss` region contains global and static local variables that are not initialized by the application so they default to a value of zero. When the Nios II processor reads uninitialized variables, it displays a warning and converts any of the bits of the uninitialized data to zero which correctly mimics the effect of the missing `.bss` clearing code. The hardware abstraction layer (HAL) code that executes before and after `main()` might use `.bss` variables, so these warnings might appear even if your application does not use the `.bss` section.

Solution Status

Not fixed.

Command-Line Utilities Do Not Display Any Output

Command-line utilities for creating and manipulating Nios II IDE projects on Windows might not display any output, even though the command completed successfully.

The following commands function correctly but do not produce output to `stdout`:

- **nios2-build-project**
- **nios2-create-application-project**
- **nios2-create-system-library**
- **nios2-delete-project**
- **nios2-import-project**

Workaround

Pipe the command output through **tee**. For example:

```
nios2-build-project --help | tee foo.txt
```

Alternatively, upgrade to the Nios II EDS v. 7.0 or later.

Solution Status

Fixed in v. 7.0.

Linux: Process Fails To Terminate While Debugging With the Nios II ISS Target

If you try to interrupt or terminate a debug session targeting the Nios II instruction set simulator (ISS), you might see an error message **Interrupt Failed or Terminate Failed**. This means that the **nios2-iss** process failed to terminate. The debug session appears to have terminated in the IDE, but the **nios2-iss** process still continues running.

Workaround

Open a command shell and kill the **nios2-iss** process as follows:

1. Type:

```
jobs
```

   to get a list of process IDs.

2. Type:

```
kill -9<nios2-iss process ID>
```

   to terminate the process.

Solution Status

Not fixed.

Windows XP: Internal Error When Repeatedly Stepping Into With Debugger

If you repeatedly use the **Step Into** command while debugging on Windows XP, you might receive the following error:

```
Retrieving Children: An internal error occurred
during: Retrieving Children
```

Workaround

End your debug session, then re-open it and resume debugging. Limit your use of the **Step Into** command.

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Nios II EDS Does Not Support Windows Vista

The Quartus II software introduces Windows Vista (32-bit and 64-bit) support in version 7.2. The SOPC Builder tool and Nios II Embedded Design Suite will support Windows Vista in version 7.2 Service Pack 1.

Workaround

There is no workaround at this time.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Nios II IDE Fails to Find File on Linux

The Nios II IDE fails to open project files on Linux and reports that the **GCC** compiler is not found.

Workaround

To avoid this issue, make a link to **nios2-elf-gcc** by doing the following:

1. Set QUARTUS_ROOTDIR to the path of the Quartus II installation.
2. Run **/altera/72/nios2eds/sdk_shell**
3. Make a link to **nios2-elf-gcc** by executing the following command:

```
ln -s $SOPC_KIT_NIOS2/bin/nios2-gnutools/\
   H-i686-pc-linux-gnu/bin/nios2-elf-gcc /tmp/gcc
```

4. Add this link to the path as follows:

```
export PATH=$PATH:/tmp
```

Solution Status

Not fixed.

Nios II IDE Errata: Building Projects

This section describes in detail the issues related to building projects in the Nios II IDE.

Build Command Not Functional for BSPs Created Using the Software Build Flow

The build option in the Nios II IDE menu does not rebuild BSPs imported into the IDE.

Workaround

To build the BSP, build the associated application project.

Solution Status

Not fixed.

Compilation Error with Separate Exception Stack Option

Choosing the **Use a separate exception stack option** might cause the compilation error "UNDEFINED VARIABLE %STACK_POINTER%" when building a project. This error occurs if the exception stack is larger than the memory available for it.

Workaround

On the system library properties page for the project, turn off the separate exception stack or reduce the **Maximum exception stack size** setting.

Solution Status

Not fixed.

Linker Errors with Dual-Port Memories

If your instruction master and data master ports are connected to the same dual-port memory and the ports have different addresses, your code fails to run or you experience a linker error. The Nios II IDE does not warn you of the addressing violation.

Workaround

Assign the same address to both ports of the dual-port memory.

Solution Status

Not fixed.

Incorrect Stack and Heap Free Space Report

The makefile reports an incorrect number of bytes free for the stack and heap, if the heap and stack are in different memory regions.

Workaround

No workaround available.

Solution Status

Not fixed.

Incorrect Stack Checking When Interrupt Vector Custom Instruction Is Used

If you enable run-time stack checking in systems that include the interrupt vector custom instruction, the Nios II IDE might erroneously report a stack overflow or might fail to report an actual stack overflow.

Workaround

Do not enable the run-time stack checking and interrupt vector custom instruction features in the same application.

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

User-Managed BSP Settings Not Supported in Nios II IDE

For projects created using the new software build flow and imported into the Nios II IDE, the IDE configuration settings have no effect.

For example, objdump, compiler, and linker settings made in the IDE are ignored. This behavior occurs because Nios II software build tools projects are not IDE-managed projects. In addition, the make-related preferences do not pertain to imported Nios II software build tools projects. The IDE ignores these options during the build process.

Workaround

Make these settings in the project's makefile.

Solution Status

Not fixed.

Error Message: 'Internal error occurred during "Launching"'

If you installed a Beta release of Nios II version 7.1 prior to installing the full release, you might encounter the error **An internal error occurred during "Launching"**, when building, running or debugging a software application project in Nios II IDE.

Workaround

Close Nios II IDE, open a Nios II command shell, and type `nios2-ide-clean`. The Nios II IDE launches again, and the error does not recur. Launching Nios II IDE with the `-clean` option does not affect any IDE or project settings you have made.

Altera strongly recommends that you uninstall Beta versions of the 7.1 tools before installing the released version of the Nios II EDS v. 7.1. Also, remove or rename the existing installation directories.

Solution Status

Fixed in v. 7.2.

Unable to Save Files Containing Copyright Symbol

On computers using a character encoding other than Cp1252, the copyright symbol included in the **board_diag.c** file is not recognized. Because of this, the file can not be saved using the Nios II IDE.

Workaround

Change the default text file encoding setting in the Nios II IDE to Cp1252 by performing the following steps:

1. In the Nios II IDE, select **Preferences** from the **Window** menu.
2. In the **General** settings, select **Workspace.**
3. Select **Default (Cp 1252)** in the **Text File Encoding** section and click **OK**.

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

URL Project Location Causes Project Creation Error

When you try to create a new project in an existing workspace, you might see an error dialog box saying:

```
Project cannot be created. Reason: Internal Error
```

This error might occur if the path to any project in the workspace is a Uniform Resource Locator (URL) location, for example **file:/F:/Design**. To view the path, right-click on the project and select **Properties**.

Workaround

Import your existing application and system library projects into a new workspace.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Nios II IDE Reports Problems Without Displaying Error in Console

When building a project, the Nios II IDE reports problems, but the build output in the console does not contain any errors.

The Nios II IDE incorrectly reports some linker warnings as errors, with a dialog box saying **Errors exist in a required project**. The Dhrystone software example exhibits this behavior, and recompiling the project again makes the issue go away.

Workaround

If the Console output does not contain errors, then the project built correctly. On subsequent builds, the linker step is skipped and the errors do not appear.

Solution Status

Not fixed.

Read-Only Zip File System Base Address Fixed at 0

The symbol `ALTERA_RO_ZIPFS_BASE` should be set to the base address of the selected flash memory. In designs created in the Nios II IDE, `ALTERA_RO_ZIPFS_BASE` is hard-coded to 0.

Workaround

Develop your software project using the Nios II software build tools.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Nios II IDE Errata: Debugging Projects

This section describes in detail the issues related to debugging projects in the Nios II IDE.

“Additional nios2-terminal Arguments” Field Is Not Functional

On the **Target Connection** tab of the **Run** or **Debug** launch configuration dialog box, the **Additional nios2-terminal arguments** field is not functional. Entering commands in these fields has no effect.

The **Additional nios2-download arguments** and **Additional nios2-flash-programmer arguments** fields are functional.

Workaround

Upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Cannot Locate Source Code in Driver Files Shared by Multiple Projects

If you hit a breakpoint in a driver file, and that driver file is shared with another project that is closed, the Nios II IDE might indicate that it cannot locate the source code.

Workaround

Open the closed system library project and resume debugging.

Solution Status

Not fixed.

Communication Issue With Multiprocessor and Multi-JTAG UART Designs

Designs that contain multiple processors or multiple joint test action group universal asynchronous receiver-transmitters (JTAG UARTs) might not be able to communicate with the target when running, debugging, or programming flash memory. Designs which have only a single processor and JTAG UART are not affected by this issue.

Workaround

You need to explicitly specify the JTAG instance number for the target. Perform the following steps:

1. Open the **Run** or **Debug** launch configuration dialog box and click the Main tab.

2. In the **Target Hardware** section, enter `--instance=` in the **Additional nios2-download arguments** field to supply the JTAG instance information to the debugger.

For designs with a single processor and JTAG UART, the default instance number N is zero and does not need to be specified as an extra argument.

Alternatively, upgrade to the Nios II EDS v. 7.0 or later.

Solution Status

Fixed in v. 7.0.

Console Window Is Not Updated After ISS Error

After performing a **Run as ISS**, if you receive an ISS error in the console window, the console is not updated subsequently.

Workaround

Close the console window after receiving an ISS error. A new console window opens when a new message is available.

Solution Status

Not fixed.

Extra Arguments in Debugger for Multiple Launch Configurations

If you have multiple debug launch configurations, you might have issues with the **Additional nios2-download arguments** field. If you delete the contents of the **Additional nios2-download arguments** field in one debug launch configuration, view a second launch configuration, with a nonempty string, in the **Additional nios2-download arguments** text box, and return to the first launch configuration, the value from the second configuration appear in the first configuration.

Workaround

To remove all arguments, use a single space rather than deleting the entire contents of the field.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

GDB Error 256 When Launching Debugger

You might see an `Error 256` message when launching the debugger if you have environment variables that are longer than 1200 characters.

Workaround

Ensure that you do not have any environment variable that exceed 1200 characters.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

ISS Fails on Designs Containing TSE MAC or SG-DMA Components

You receive an Internal Error when attempting to perform an ISS simulation of designs containing the Altera Triple Speed Ethernet (TSE) MAC or scatter-gather direct memory access (SG-DMA) components because the Nios II ISS does not support these components.

Workaround

Remove the TSE MAC and SG-DMA components from your system and perform ISS simulation on the simplified system. You can also simulate the design in ModelSim® or test it on hardware.

Solution Status

Not fixed.

Memory Window Sets Control Register Values Incorrectly

The memory window might incorrectly set values in memory-mapped control registers. For example, writing 0x1234 to a byte addressed register results in the value 0x3434 being stored in the register. The memory window shows this incorrect value.

Workaround

Use the GNU debugger (GDB) console window in the IDE, instead of the memory window, to write to the registers. For example, type:

```
set {int} <register address>=0x1234↵
```

in the GDB console window.

☞ You must refresh the memory window in order for it to correctly display the target value.

Solution Status

Not fixed.

MicroC/OS-II “Source Not Found” Error Occurs at Breakpoint

You might receive a 'Source not found' error message if you step into MicroC/OS-II code or break on a program entry point while debugging your application.

Workaround

To eliminate this error, you must specify the proper source path mapping in the Nios II IDE by performing the following steps.

1. On the Windows menu, click **Preferences**. The **Preferences** dialog box appears.

2. Expand the **C/C++** group, then expand the **Debug** group, then click **Common Source Lookup Path**.

3. Click **Add...**. The **Add Source** dialog box appears.

4. Select **Path Mapping**, then click **OK**.

5. In the **Preferences** dialog box, select **Path Mapping: New Mapping** and click **Edit**. The **Path Mappings** dialog box appears.

6. Enter `cygdrive` in the **Name** field.

7. Click **Add...**. A dialog box appears, prompting you to specify the path mapping.

8. Enter `/cygdrive/c` in the **Compilation path** field and `c:/` in the **Local file system path** field.

9. Click **OK** in all open dialog boxes to return to the IDE workbench.

This process sets up the source path mapping for the entire workspace, which is recommended. You can also create a similar mapping for individual projects by clicking **Edit Source Lookup Path** that appears below the **Source not found** error message and following steps 4 through 8.

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Programs That Interact With a Terminal Console on Windows Do Not Work

Programs with this behavior work in v. 6.0 and earlier, but do not work in Nios II IDE v. 6.1 and later.

The Eclipse platform in v. 6.1 and later of the IDE (on Windows only) sends the string `\r\n` instead of just `\n` when running or debugging using the Terminal. This behavior can break existing software designs, and it is inconsistent with **nios2-terminal**, which still just sends `\n`.

Workaround

Change the software running on the Nios II processor to parse for `\r\n` as well as `\n`.

Solution Status

Not fixed.

"Run as ModelSim" in the Nios II IDE Fails

The **Run as ModelSim** command might fail on launch configurations created in v. 7.0 or earlier of the IDE. This problem does not occur for new launch configurations.

Workaround

Select a location for the ModelSim tool from the launch configuration dialog box. You can use the **Browse** button next to the **ModelSim path** group, or type in a path to the ModelSim directory (e.g. **c:/altera/71/modelsim_ae/win32aloem**).

Solution Status

Not fixed.

The Restart Command on the Run Menu Does Not Work

The **Restart** command on the **Run** menu does not work.

Workaround

Stop the program, then debug it again. If the debugger is hung in an endless loop, use the following **bash** alias to break the target, then stop it:

```
alias break="kill -2 \’ps -a | grep nios2-elf-gdb | cut -f6 -d' '\’"
```

Solution Status

Not fixed.

Nios II IDE Debugger Cannot Display Driver Source Code for Custom Component

When debugging a custom component in the Nios II IDE, the debugger might not find the software driver files for that component.

Workaround

You must add a linked folder to an IDE project so that the IDE can find source code files associated with the component. The locations which you must add appear in a generated file called **install2.ptf** in the directory containing the **install.ptf** for the hardware design. Examine this file to find the directory path which you must add. For example, a custom component called **altera_avalon_pwm** is stored in **/data/pwm_component_test**. An excerpt from **install2.ptf** follows:

```
PACKAGE install2
{
   COMPONENT altera_avalon_pwm
   {
      VERSION 2.0
      {
         local = "/data/pwm_component_test";
      }
   }
}
```

To add the path **/data/pwm_component_test** to the IDE, carry out the following steps:

1. Right-click the system library project for your design.

2. Point to **New > Other > General > Folder**, and click **Next**.

3. Specify the parent project (e.g. **syslib**) in the **Enter or select the parent folder** field.

4. Click **Advanced**, turn on **Link to folder in the file system**, and browse to or type in the path to the component.

5. Click **Finish** to link the folder.

☞ Do not link a custom component to more than one system library in the IDE. The breakpoint manager sometimes fails to handle the aliases, which results in breakpoints being hit even when removed from the breakpoint manager.

Solution Status

Not fixed.

Watchpoints Do Not Work on Certain Variables

Watchpoints do not work on a variable whose size is not 32 bits.

Workaround

Change the types of global and static local variables to `int`, `long`, or `unsigned long` before setting watchpoints on them.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Trace Debug Does Not Support the JMPI Instruction

The new JMPI instruction introduced in version 7.2 is not supported by the Nios II trace tools.

Workaround

There is no workaround at this time.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Nios II IDE Errata: Navigating Projects

This section describes in detail the issues related to navigating projects in the Nios II IDE.

Nios II IDE Freezes While Displaying the Splash Screen

After clicking **Switch Workspace** on the File menu on a Windows machine, a Nios II IDE splash screen appears. Unfortunately, this splash screen obscures a dialog box which asks you to specify the new workspace. This makes the IDE appear to freeze.

Workaround

Press Alt-Tab to switch applications. Two relevant application icons appear: an Eclipse icon associated with the splash screen and a Nios II IDE icon associated with the **Workspace** dialog box. Select the Nios II icon to bring the dialog box to the foreground.

Solution Status

Not fixed.

C/C++ Scanner Does Not Support Certain C/C++ Constructs

The C/C++ scanner performs C/C++ Search, navigation, open declaration, and parts of content assist. Due to limitations of the C/C++ scanner, these features do not work with C code constructs not supported by the C++ language. An example is functions that take a function pointer as an argument.

Workaround

If the C/C++ Search fails, use the File Search facility.

Solution Status

Not fixed.

C/C++ Search Returns Nothing

The C/C++ Search in the Nios II IDE does not return results because a search index is not created at start up.

Workaround

Manually invoke indexing for projects you need to search. Refer to the “Indexing and Searching Project Content” IDE online help topic for instructions.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Internal Error When Double-Clicking on a Large Objdump File

On Windows when opening a large **objdump** file in the Nios II IDE, you might get the following error message:

```
Unable to create this part due to an internal error.
Reason for the failure: Editor could not be initialized.
```

Workaround

Adjust the Windows launch arguments for the Nios II IDE editor. Perform the following steps:

1. On the Windows Start menu, browse to the **Nios II EDS** program icon, right-click it, then click **Properties**. The **Windows Properties** dialog box appears.

2. In the **Target** field, append “`vmargs -Xmx1024m`” to the end of the path to the Nios II IDE executable. For example:

```
C:\altera\72\nios2eds\bin\eclipse\nios2-ide.exe -vmargs -Xmx1024m
```

Solution Status

Not fixed.

Nios II IDE Does Not Recognize When an Object File is Built Outside the IDE

If you use a build process external to the Nios II IDE, the IDE might not recognize when you have generated or re-generated the ELF file for your project. In this case, the IDE debugger might report an error that the ELF file does not exist.

Workaround

Right-click on the project name and click **Refresh** to allow the IDE to recognize the ELF file.

Alternatively, upgrade to the Nios II EDS v. 7.1 or later.

Solution Status

Fixed in v. 7.1.

Nios II IDE Import System Library Dialog Box Does Not Close After Import

When importing a system library project, the Nios II IDE wizard does not close after clicking the **Finish** button. The wizard imports the project successfully after clicking **Finish** once. Clicking **Finish** again results in an error message stating that the project is already created.

Workaround

Click **Cancel** in the **Import Wizard** dialog box after clicking **Finish**.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Nios II Software Build Tools Errata

This section describes in detail the issues in the Nios II software build tools.

"make download-elf" Fails for Stratix II 2S60 RoHS Design Examples

You might see error messages when running the 2S60 RoHS example applications with the Software Build Tools. These messages appear due to an incorrect system ID value in the BSP makefile. The error messages are similar to the following:

```
Reading System ID at address 0x021208B8:
    ID value does not match: read 0x072DA598; expected 0x7D1BD35D
    Timestamp value was not verified: value was not specified
The software you are downloading may not run on the system which is
currently configured into the device.  Please download the correct SOF
or recompile
```

Workaround

Remove the system ID check from the application makefile. To remove the system ID check, carry out the following steps:

1. Open the application **Makefile,** located in the application project directory.

2. Search for "`download-elf`" to find the section that needs modification.

3. Remove the `$(SOPC_SYSID_FLAG)` argument from the `$(DOWNLOAD)` command in the following section:

```
.PHONY : download-elf
download-elf : $(ELF)
      @echo Info: Downloading $(ELF)
      $(DOWNLOAD) --go --cpu_name=$(CPU_NAME) $(SOPC_SYSID_FLAG)
$(WRITE_GMON_OPTION) $(ELF)
```

The resulting section looks like:

```
.PHONY : download-elf
download-elf : $(ELF)
      @echo Info: Downloading $(ELF)
      $(DOWNLOAD) --go --cpu_name=$(CPU_NAME) $(WRITE_GMON_OPTION) $(ELF
```

Rebuild the application.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

User-Managed BSP Settings Not Supported in Nios II IDE

See page 21–33.

Improper Quote Format in system.h

The Nios II software build tools tcl command

```
add_sw_setting character system_h_define character_system_h \
    character_system_h a "character_system_h"
```

generates a **system.h** file with contents in double quotes (i.e. `"a"`) instead of the proper single quotes (i.e. `'a'`).

Workaround

Manually edit the **system.h** file to fix the appropriate line by changing double quotes to single quotes.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Upper-Case File Extensions Not Supported

The Nios II software build tools for applications and libraries (**nios2-app-generate-makefile** and **nios2-lib-generate-makefile** commands) do not support source files with certain upper-case extensions. If a file with an upper-case extension is included, the make command stops with no descriptive warning.

Only Nios II assembly language files built by the C preprocessor can have upper-case file extension (**.S**). All C language files must have the extension **.c** or **.h**. C++ language source files must have the extension **.cpp**, **.cxx**, **.cc,** or **.h**.

Workaround

Rename all C language files with the extension **.c** or **.h**. Rename all C++ language files with the extension **.cpp**, **.cxx**, **.cc,** or **.h**.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Software Build Tools Fail if Nios II EDS is Installed in a Path Containing Spaces

The Nios II Command Line Software Build Tools fail if you install the Nios II EDS in a path containing spaces.

Workaround

Re-install the Nios II EDS into a path that does not contain spaces.

Solution Status

Not fixed.

JTAG Instance Clash for Multiprocessor Systems Imported Into IDE

When importing a multiprocessor project created using the Software Build Tools into the Nios II IDE, the instance IDs for the Nios II processor might be incorrect, causing JTAG errors during debugging.

Workaround

In the Nios II IDE **Debug** launch configuration, inspect each Nios II processor's instance number in the **Additional nios2-download arguments** field. Manually change the instance ID of each processor to the correct value. The correct instance ID is located in the <*SOPC Builder system name*>**.jdi** file. Search for the processor name. The instance ID is located in the "`node hpath`" line for that processor instance.

☞ The Nios II IDE allows you to set the instance ID only for the currently selected processor. To correct the instance ID on another processor, first select it.

Solution Status

Not fixed.

system.h File Does Not Reflect Pre-7.1 Custom Instruction Name Changes

If you instantiate a **class.ptf**-based custom instruction (created in SOPC Builder v. 7.0 or earlier) or upgrade a design containing a **class.ptf**-based custom instruction, subsequent custom instruction name changes are lost when you close and re-launch SOPC Editor. The **system.h** file generated by the Software Build Tools does not reflect the name change.

Workaround

Manually change the name in the **system.h** file.

Solution Status

Not fixed.

Peripheral Errata

This section describes in detail the Nios II EDS issues related to peripheral components.

DMA Controller Always Busy in Burst Mode

The DMA controller component (**altera_avalon_dma**), when enabled for burst transactions, does not perform transfers at widths other than its full data width. The DMA controller is always busy.

Workaround

When bursting is enabled, the DMA controller must be programmed to perform transactions at its full data width.

Solution Status

Not fixed.

Non System Wide Reset Can Cause Improper Initialization of Mailbox Core

The **altera_avalon_mailbox** peripheral may not be initialized properly when a soft (non-system-wide) reset occurs. Examples: the `resetrequest` signal resets a specific Nios II processor that masters the mailbox; a processor mastering the mailbox restarts execution from the start address. In this condition, mailbox contents (read and write pointers) are not reinitialized and may show potentially stale data.

Workaround

Ensure that a system-wide reset event occurs by asserting the `reset_n` input to the SOPC Builder system containing the mailbox. This resets all peripherals and Nios II processors in the system. Alternatively, ensure that any messages posted to the mailbox are cleared before issuing a soft reset.

Solution Status

This issue will be fixed in a future release of the Nios II EDS.

Triple Speed Ethernet Designs Require a Manual Reset on Subsequent Software Downloads

You might experience issues in designs containing the Altera Triple Speed Ethernet MegaCore function. These issues might prevent the TSE MAC from operating properly after repeated downloads of the Nios II processor's ELF file. When this bug occurs, even though the Nios II processor executes the correct initialization sequence to reset the Triple Speed Ethernet component, a complete reset never occurs. The result is that the TSE MAC is unable to transmit or receive Ethernet data.

Workaround

Before downloading the ELF file to the Nios II processor, push the board's reset button. This action resets the Altera Triple Speed Ethernet MAC into a known-good state.

Alternatively, upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Software Build Errors Due to _classic Suffix

This issue arises when you integrate a device driver for “classic” SOPC Builder IP, that is, IP described by a **class.ptf** file, rather than the newer **_hw.tcl** script. In this situation, the Nios II Software Build Tools append “_classic” to the class name that you specify for your custom component. The tools make these modifications, for example, to macros in **system.h** and to header file names in **system.h** and **alt_sys_init.c**. The modified names cause software build errors.

Workaround

Upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Software Build Tools Select Inappropriate Memories

When selecting default memory devices for memory regions, the Nios II software build tools sometimes select a memory that is not mastered by an appropriate master port on the Nios II IDE.

Workaround:

Upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

Target Software Errata

This section describes in detail the Nios II EDS issues related to target software components.

Compiling an Existing MicroC/OS-II Application Generates Errors

The version of MicroC/OS-II was upgraded in the v. 6.1 release, which can cause this issue. After importing a Nios II IDE project that uses MicroC/OS-II created with version 6.0 SP1 or earlier, building the project in the Nios II IDE generates errors, such as

```
#error "OS_CFG.H, Missing OS_FLAGS_NBITS: Determine
#bits used for event flags, MUST be either 8, 16 or 32'
```

Workaround

Resetting the MicroC/OS-II software component settings to their defaults fixes this issue. To reset the component settings, carry out the following steps:

1. Right-click on the application project in the Nios II C/C++ view.
2. Click **System Library Properties**.
3. Click **RTOS Options...** to display the **MicroC/OS-II RTOS Options** dialog box.
4. Expand the **MicroC/OS-II** group in the upper-left-hand corner of the dialog box.
5. Examine each page of options, and make note of your settings.
6. Click **Restore Defaults.**
7. Re-examine each page of options, and re-enter any settings that changed.
8. Click **OK** to dismiss the **MicroC/OS-II RTOS Options** dialog box.
9. Click **OK** again to dismiss the Properties dialog box.

Solution Status

This issue only affects upgrades from v. 6.0 SP1 or earlier.

cout From MicroC-OS/II Task Does Not Send Data to stdout

If neither `printf()` or `cout` is used from `main()` before tasks are started, `cout` does not work from a task.

Workaround

Add the following C++ code to the beginning of `main()`:

```
std::ios_base::sync_with_stdio(false);
```

Solution Status

Not fixed.

malloc(), realloc() Failures With MicroC/OS-II

When you use the MicroC/OS-II real-time operating system (RTOS), calls to `malloc()` and `realloc()` might fail if successive calls to `malloc()`or `realloc()` within a MicroC/OS-II task occur after changing the task priority of the task in which a memory block is originally allocated.

Workaround

Use one of the following workarounds:

- Allocate and/or reallocate memory blocks outside of MicroC/OS-II tasks, before task switching starts. This makes it possible to change thread priorities at runtime.
- Allocate fixed areas of memory using arrays (rather than using `malloc()`) before task switching starts. This makes it possible to change thread priorities at runtime.
- Allocate memory using `malloc()` or `realloc()` from a MicroC/OS-II task. You can change task priorities at runtime, but only for tasks that have not used `malloc()` or `realloc()`.

Solution Status

Not fixed.

stdio Does Not Work with MicroC/OS-II and Small C Library

`stdin`, `stdout`, and `stderr` do not work in MicroC/OS-II applications built with the Small C library option.

Workaround

Disable the small C library option.

Solution Status

Not fixed.

System Clock Rollover

The `gettimeofday()` function in the Nios II HAL can experience clock rollover. After a few thousand system clock ticks, the count of microseconds since system startup overflows, resulting in an incorrect function return value.

Workaround

Upgrade to the Nios II EDS v. 7.2 or later.

Solution Status

Fixed in v. 7.2.

How to Contact Altera

For the most up-to-date information about Altera® products, refer to the following table.

Contact (1)	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Product literature	Website	www.altera.com/literature
Altera literature services	Email	literature@altera.com
Non-technical support (General)	Email	nacomp@altera.com
(Software Licensing)	Email	authorization@altera.com

Note:
(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

The following table shows the typographic conventions that this document uses.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Command names, dialog box titles, checkbox options, and dialog box options are shown in bold, initial capital letters. Example: **Save As** dialog box.
bold type	External timing parameters, directory names, project names, disk drive names, file names, file name extensions, and software utility names are shown in bold type. Examples: **f_{MAX}**, **\qdesigns** directory, **d:** drive, **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Document titles are shown in italic type with initial capital letters. Example: *AN 75: High-Speed Board Design*.
Italic type	Internal timing parameters and variables are shown in italic type. Examples: t_{PIA}, $n + 1$. Variable names are enclosed in angle brackets (< >) and shown in italic type. Example: *<file name>*, *<project name>*.**pof** file.
Initial Capital Letters	Keyboard keys and menu names are shown with initial capital letters. Examples: Delete key, the Options menu.
“Subheading Title”	References to sections within a document and titles of on-line help topics are shown in quotation marks. Example: “Typographic Conventions.”

Typographic Conventions

Visual Cue	Meaning
`Courier type`	Signal and port names are shown in lowercase Courier type. Examples: `data1`, `tdi`, `input`. Active-low signals are denoted by suffix `n`, e.g., `resetn`. Anything that must be typed exactly as it appears is shown in Courier type. For example: `c:\qdesigns\tutorial\chiptrip.gdf`. Also, sections of an actual file, such as a Report File, references to parts of files (e.g., the AHDL keyword `SUBDESIGN`), as well as logic function names (e.g., `TRI`) are shown in Courier.
1., 2., 3., and a., b., c., etc.	Numbered steps are used in a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ● •	Bullets are used in a list of items when the sequence of the items is not important.
✓	The checkmark indicates a procedure that consists of one step only.
☞	The hand points to information that requires special attention.
⚠ CAUTION	A caution calls attention to a condition or possible situation that can damage or destroy the product or the user's work.
⚡ WARNING	A warning calls attention to a condition or possible situation that can cause injury to the user.
↵	The angled arrow indicates you should press the Enter key.
👣	The feet direct you to more information on a particular topic.

Nios II Embedded Design Suite 7.1 Release Notes

May 2007, Version 7.1 **Release Notes**

This document contains release notes for the Nios® II Embedded Design Suite (EDS) version 7.1.

Table of Contents:

New Features and Enhancements

The Nios II EDS version 7.1 addresses issues found in previous releases, and adds the following primary features:

- Nios II Software Build Tools design flow
- Arria™ GX device family support
- Design examples including new SOPC Builder components
  - Triple Speed Ethernet Media Access Controller (MAC) MegaCore
  - Scatter-gather Direct Memory Access (DMA) Controller

The following sections provide a detailed list of all product updates, including known issues fixed in this release.

Device and Host Support

This release supports the following Altera® FPGA families:

- Arria GX
- Cyclone™ III
- Stratix® III
- Stratix II and Stratix II GX
- Stratix and Stratix GX
- Cyclone II
- Cyclone

The host system requirements for the Nios II EDS are based on the requirements for the Altera Complete Design Suite. Refer to *Quartus® II Installation and Licensing for PCs* or *Quartus II Installation and Licensing for UNIX and Linux Workstations* manual.

Installation and Licensing Instructions

For installation instructions, refer to *Quartus II Installation and Licensing for PCs* or *Quartus II Installation and Licensing for UNIX and Linux Workstations* on the Altera website at **www.altera.com/literature/lit-qts.jsp**.

Using Previously Installed Versions of the Nios II EDS

SOPC Builder and the Nios II Integrated Development Environment (IDE) refer to the most recently installed version of components (such as the Nios II processor and peripherals) and their software drivers. To revert to a prior released version of the Nios II development tools, you can reinstall the previous version of tools or modify the following environment variables:

- SOPC_BUILDER_PATH — Ensure that SOPC_BUILDER_PATH points to the installation directory of the desired Nios II version and no other Nios II versions. SOPC_BUILDER_PATH must also contain the paths of other IP, separated by +. For example:

```
SOPC_BUILDER_PATH=C:\altera\71\ip\sopc_builder_ip+C:\altera\71\ip\nios2_ip+
C:\altera\71\ip\triple_speed_ethernet\lib\sopc_builder+C:\altera\71\ip\ddr_ddr2_sdram\lib\sopc_builder
```

- SOPC_BUILDER_PATH_<*version*> — Ensure that SOPC_BUILDER_PATH_<*version*> points to the installation directory of the corresponding Nios II version. This environment variable is necessary only for versions 6.0 and 5.1. Exclude the period in <*version*>. For example:

```
SOPC_BUILDER_PATH_60=c:\altera\kits\nios2_60
```

- SOPC_KIT_NIOS2 — Ensure SOPC_KIT_NIOS2 points to the installation directory of the desired Nios II version and no other Nios II versions. For example:

```
SOPC_KIT_NIOS2=C:\altera\71\nios2eds
```

After modifying SOPC_BUILDER_PATH, SOPC_BUILDER_PATH_<*version*>, and SOPC_KIT_NIOS2, launch the corresponding version of the Quartus II software to ensure that the QUARTUS_ROOTDIR environment variable is updated. Exit the Quartus II software immediately. After you complete this step, the next time you launch any part of the Altera Complete Design Suite, the version you select runs.

For Windows, there is a batch script called **Altera_Tool_Switch.bat**, along with a special version of **setenv.exe** which can update system variables. **Altera_Tool_Switch.bat** automates the process of updating the system variables to switch versions of Altera software. To get these tools, download **ftp://ftp.altera.com/outgoing/download/kdb/Altera_Tool_Switch.zip**.

☞ Altera strongly recommends that you uninstall any Beta versions of the Nios II EDS v7.1 before installing the release version.

Installation Path Length

During installation of the Nios II EDS on Windows, you might receive a **Feature Transfer Error** message indicating that the filename or extension is too long. This error might occur if your install path is too long. Use an install location with a shorter path, for example, **C:\altera\**.

Nios II Processor Cores

This section describes changes to the Nios II processor cores.

There are no major changes to Nios II processor cores in this release.

SOPC Builder

This section describes changes to SOPC Builder which affect Nios II designers.

For complete revision history of SOPC Builder and the Quartus II software, refer to the release notes for the Quartus II software version 7.1. The *Quartus II Handbook, Volume 4: SOPC Builder* contains complete documentation for SOPC Builder.

The Quartus II version 7.1 release notes include information on SOPC Builder changes and the new embedded peripherals provided with SOPC Builder in this release.

When you open an SOPC builder system created in a previous version of the tools, a dialog box appears. For information on upgrading SOPC Builder systems to version 7.1, select the **More Information** option.

Nios II IDE

This section describes changes to the Nios II integrated development environment (IDE).

There are no major changes to the Nios II IDE in this release.

Nios II Software Build Tools

The Nios II Software Build Tools are introduced in the Nios II EDS version 7.1. The Nios II software build tools are command line utilities and scripts that provide similar functionality to the New Project wizard and the System Library properties page in the Nios II IDE. The software build tools also provide scripting capabilities.

Refer to the *Nios II Software Build Tools* chapter in the *Nios II Software Developer's Handbook* for details.

Nios II C-to-Hardware Acceleration (C2H) Compiler

This section describes changes to the Nios II C2H Compiler.

Integration with Nios II Software Build Tools

The C2H Compiler is supported by the Nios II software build tools.

Interrupt Generation

You can compile a top-level function to an accelerator that runs in interrupt mode.

For details on these features, refer to the *Nios II C2H Compiler User Guide*.

Flash Programmer

This section describes changes to the flash programmer in the Nios II IDE.

There are no major changes to the Flash Programmer in this release.

Target Software

This section describes changes to Altera-provided target software which runs on the Nios II processor, such as the hardware abstraction layer (HAL).

There are no major changes to the target software in this release.

Example Designs

Hardware Example Designs

HDL Top Level Design Files

The hardware examples in version 7.1 contain Verilog HDL or VHDL top level files. Block Design Files (**.bdf**), formerly included as schematic top level files, are no longer present. The system HDL file generated by SOPC Builder is the top level file for most of the example projects, demonstrating how SOPC Builder can simplify your system design process. This change removes the chance of introducing incorrect pin connections in the top level schematic file. For instructions on creating BDF top level files for the examples, refer to the **readme.txt** file included with any hardware example design.

Nios Development Board, Stratix ES Edition Example Designs Removed

The hardware examples for the Nios II 1S10ES development boards are removed in the Nios II EDS v7.1. These boards are obsolete.

TSE SG-DMA

The new Triple Speed Ethernet Scatter-Gather DMA (TSE SG-DMA) hardware design example is included for the Nios Development Board, Stratix II RoHS-compliant edition and Cyclone II edition. This design demonstrates the use of the Altera TSE MAC and SG-DMA components for Ethernet applications. For details on the design examples, refer to the Nios II EDS v7.1 Documentation Launch Pad. To view the Documentation Launch Pad, in the Windows Start menu point to **Programs**, **Altera**, **Nios II EDS 7.1**, and click on **Nios II 7.1 Documentation**.

Fast

Fast hardware example designs are included targeting Cyclone III and Stratix III devices. Refer to the Nios II EDS v7.1 Documentation Launch Pad for details on the design examples. To view the Documentation Launch Pad, in the Windows Start menu point to **Programs** > > > **Altera > > > Nios II EDS 7.1**, and click on **Nios II 7.1 Documentation**.

Small

A Joint Test Action Group (JTAG) UART is now included in the **small** hardware example designs to enable communication with the host machine.

Software Example Designs

Web Server

The Web Server software example now demonstrates control of board components over Ethernet.

Simple Socket Server and Web Server Default IP Address

The default IP addresses in these examples are updated to allow for Dynamic Host Configuration Protocol (DHCP) request packets to penetrate secure routers. The address was changed from **192.168.X.X** to **0.0.0.0**.

Errata Fixed in This Release

This section lists those errata items from previous releases that are fixed in version 7.1.

Fixed in Version 7.1: Nios II/e Core Burst Support

The Nios II wizard incorrectly allowed you to select burst support for the instruction master port in the Nios II/e core. Instruction master burst support in Nios II requires an instruction cache. The Nios II/e core does not include an instruction cache; therefore burst support can not be enabled.

Fixed in Version 7.1: Interrupt Vector Custom Instruction in Multiprocessor Systems

In designs containing multiple Nios II processors, enabling the interrupt vector custom instruction in more than one of the processors caused the following errors during Quartus II compilation:

```
Error (10430): VHDL Primary Unit Declaration error at
interruptvector_cpu_1.vhd(26): primary unit "interrupt_vector" already exists in
library "work"
Error (10523): Ignored construct interruptvector_cpu_1 at
interruptvector_cpu_1.vhd(64) due to previous errors
Error: Node instance "the_interruptvector_cpu_1" instantiates undefined entity
"interruptvector_cpu_1"
```

Fixed in Version 7.1: "nios_cpu: Unknown Break Location" Error Message

If you modified the Nios II core in an existing system by disabling the JTAG debug module in the Nios II configuration wizard, you might have received the error message **nios_cpu: Unknown Break Location nios_cpu/jtag_debug_module** during SOPC Builder generation.

Fixed in Version 7.1: Windows XP: Internal Error When Repeatedly Using Step Into

If you repeatedly used the **Step Into** command while debugging on Windows XP, you might have received the internal error, **Retrieving Children: An internal error occurred during: Retrieving Children**.

Fixed in Version 7.1: Unable to Save Files Containing Copyright Symbol

On computers using character encoding other than Windows-1252, the copyright symbol included in the **board_diag.c** file was not recognized. Because of this character, the file could not be saved using the Nios II IDE.

Fixed in Version 7.1: MicroC/OS-II "Source not found" Error When Debugging

You might have received a **Source not found** error message if you stepped into MicroC/OS-II code or broke on a program entry point while debugging your application.

Fixed in Version 7.1: Incorrect Stack Checking with Interrupt Vector Custom Instruction

If you enabled run-time stack checking in systems that included the interrupt vector custom instruction, the Nios II IDE sometimes erroneously reported a stack overflow or sometimes failed to report an actual stack overflow.

Fixed in Version 7.1: Nios II IDE Did Not Recognize Executable File Built Outside IDE

If you used a build process external to the Nios II IDE, the IDE sometimes did not recognize when you generated or regenerated the executable linkable file (**.elf**) for your project. In this case, the IDE debugger reported that the executable file did not exist.

Fixed in Version 7.1: Comma Operators Not Supported

The C2H Compiler did not support comma operators, such as in the following example:

```
for(i = 0, j = 3; i < 10; i++, j++)
{
  /* statements */
}
```

Fixed in Version 7.1: Failure Verifying Flash With the nios2-flash-programmer Command

Using the **nios2-flash-programmer** command line utility to verify flash contents using the `--verify` argument sometimes resulted in a verify failure, even when flash contents were correct. The failure message was similar to the following:

```
Verifying 00000000 ( 0%)Failed to verify at around 00000000  Verify failed
```

Fixed in Version 7.1: Using Nios II Flash Programmer With Override File in <*Nios II EDS path*>/bin Generated Error

An error resembling the following sometimes appeared:

```
4 [main] nios2-flash-programmer 4440 _cygtls::handle_exceptions: Error while
dumping state (probably corrupted stack)  Segmentation fault (core dumped)
```

Fixed in Version 7.1: Nios II Processor Reference Handbook Referred Erroneously to the Data Bus

In the *Implementing the Nios II Processor in SOPC Builder* chapter of the *Nios II Processor Reference Handbook*, under *Caches and Tightly Coupled Memories Tab*, the following text appeared under *Instruction Settings*: "Usually you enable bursts on the processor's data bus when processor data is stored in DRAM, and disable bursts when processor data is stored in SRAM." This sentence now reads "Usually you enable bursts on the processor's instruction master when instructions are stored in DRAM, and disable bursts when instructions are stored in SRAM."

Revision History

Table 1 shows the revision history for the Nios II Embedded Design Suite v7.1 Release Notes.

Table 1. Nios II Embedded Design Suite v7.1 Errata Sheet Revision History

Version	Date	Summary
1.0	May 2007	First release
1.1	May 2007	• Remove mention of Web Server remote configuration • Remove mention of incorrect SSRAM PLL phase shift in Stratix 2S60 RoHS and Cyclone II 2C35 designs

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com
Applications Hotline:
(800) 800-EPLD
Literature Services:
literature@altera.com

Nios II Embedded Design Suite 7.1 Errata Sheet

May 2007 | Errata Sheet

This document addresses known errata and documentation issues for the Nios® II Embedded Design Suite (EDS) version 7.1. Errata are functional defects or errors, which might cause the product to deviate from published specifications. Documentation issues include errors, unclear descriptions, or omissions from current published specifications or product documents. Errata items discovered after the release of Nios II EDS version 7.1 are marked with the date the items were added to this document.

Table of Contents:

For the most up-to-date errata for this release, refer to the errata sheet on the Altera® website: **http://www.altera.com/literature/es/es_nios2eds_71.pdf**

Nios II Processor Core

This section lists all issues related to the Nios II processor cores.

Level 4 Debug in Stratix III Designs

You will get the following error during Quartus® II compilation if you are using Level 4 debug in the Nios II core and are targeting a Stratix® III device. Level 4 debug is not currently supported in designs targeting Stratix III devices.

```
Error: MGL_INTERNAL_ERROR: Port object altpll|clk of width  6 is being
assigned the port altpll|stratixii_pll inst pll1|clk of width 10 which is
illegal, as port widths dont match nor are multiples…
```

Workaround: There is no workaround available at this time. This issue will be fixed in a future version of the Nios II core.

VHDL Sensitivity List Warnings During Quartus II Synthesis

You might receive warnings similar to the following in the Quartus II software when compiling a VHDL Nios II system containing the JTAG debug module.

```
Warning (10492): VHDL Process Statement warning at
cpuname_jtag_debug_module.vhd(254): signal "usr1" is read inside the Process
Statement but isn't in the Process Statement's sensitivity list
Warning (10492): VHDL Process Statement warning at
cpuname_jtag_debug_module.vhd(254): signal "ena" is read inside the Process
Statement but isn't in the Process Statement's sensitivity list
```

These warnings are benign and can be ignored. The warnings are a result of the `ena` and `usr1` signals not being included in the debug module's sensitivity list.

Double-Precision Floating-Point Operations With Floating-Point Custom Instructions

Calls to double-precision floating-point functions in **math.h** return less-precise results on Nios II processors using the floating-point custom instruction. Floating-point constants are forced to single-precision when the floating-point custom instruction is present, which affects the constants for the double-precision floating-point functions in libm.

Workaround: There is no workaround available at this time.

Peripherals

This section lists all issues related to the Altera embedded peripherals included in the Quartus II software.

TSE MAC Designs Require a Manual Reset on Subsequent Software Downloads

You may experience issues in designs containing the Altera Triple Speed Ethernet (TSE MAC) MAC which may prevent the TSE MAC from operating properly after repeated downloads of the Nios II processor's executable linkable file (**.elf**). When this bug occurs, even though the Nios II processor executes the correct initialization sequence to reset the TSE MAC component, a complete reset never occurs. The result is that the TSE MAC is unable to transmit or receive Ethernet data.

Workaround: Before downloading the software image file to the Nios II processor, push the board's reset button. This action will reset the Altera Triple Speed Ethernet MAC into a known-good state.

Non System Wide Reset Can Cause Improper Initialization of Mailbox Core

The **altera_avalon_mailbox** peripheral may not be initialized properly when a soft (non-system-wide) reset occurs. Examples: the **resetrequest** signal resets a specific Nios II processor that masters the mailbox; a processor mastering the mailbox restarts execution from the start address. In this condition, mailbox contents (read and write pointers) will not be reinitialized and may show potentially stale data.

Workaround: Ensure that a system-wide reset event occurs by asserting the **reset_n** input to the SOPC Builder system containing the mailbox. This will reset all peripherals and Nios II processors in the system. Alternatively, ensure that any messages posted to the mailbox are cleared before issuing a soft reset.

DMA Controller Always Busy in Burst Mode If Transfers Less Than Full Width

The DMA controller component (**altera_avalon_dma**), when enabled for burst transactions, will not do transfers at widths other than its full data width. The DMA controller is always busy.

Workaround: When bursting is enabled, the DMA controller must be programmed to do transactions at its full data width.

Using the DEV_CLRn Signal Can Cause JTAG UART Instability After Device-Wide Reset

If the **DEV_CLRn** pin on the FPGA input has been assigned (in Quartus II software) to generate a device-wide reset, and the FPGA is reset while the JTAG UART is active, then the JTAG UART might become unstable.

Workaround: Do not use the **DEV_CLRn** function in designs with the JTAG UART. Turn off the **Enable device wide reset (DEV_CLRn)** setting in Quartus II software.

Host Platform

This section lists all issues related specifically to the host platform.

Linux: Process Fails To Terminate While Debugging With the Nios II ISS Target

If you try to interrupt or terminate a debug session targeting the Nios II instruction set simulator (ISS), you might see an error message **Interrupt Failed or Terminate Failed**. This means that the **nios2-iss** process failed to terminate. The debug session appears to have terminated in the IDE, but the **nios2-iss** process still continues running.

Workaround: Open a command shell and kill the **nios2-iss** process as follows:

1. Type:

```
jobs
```

to get a list of process IDs.

2. Type

   `kill -9` *<nios2-iss process ID>*

   to terminate the process.

Linux: The Quartus II Stand-Alone Programmer is Not Supported

There is no Quartus II stand-alone programmer for Linux. As a result, in the Nios II IDE the Quartus II Programmer command on the Tools menu has no effect. When you attempt to download software to a board without the expected hardware image, the IDE does not launch the programmer.

Workaround: Launch the Quartus II software to run the Quartus II Programmer.

Windows: Frisk Antivirus Software Causes SOPC Builder and Nios II Command Shell to Be Unresponsive

The SOPC Builder and Nios II Command Shell might become unresponsive if run while the Frisk antivirus software is running.

Workaround: Turn off the Dynamic Virus Checking feature of the Frisk software before running SOPC Builder or the Nios II Command Shell.

Device

This section lists any device-related issues.

Stratix II EP2S60 ES Devices Cannot Use MRAM Byte Enables

Early shipments of the Nios II Development Kit, Stratix II Edition include an EP2S60 engineering sample (ES) device. Stratix II EP2S60 ES devices have a silicon problem that prevents the use of byte enables on MRAM blocks.

Workaround: Refer to the *Stratix II FPGA Family Errata Sheet* for details.

Nios II IDE

This section lists all issues relating to the Nios II IDE.

Building Projects

Build Option Not Functional for BSPs Created Using the Software Build Flow

The build option in the Nios II IDE menu does not rebuild BSPs imported into the IDE.

Workaround: To build the BSP, build the associated application project.

Previous 7.1 Beta Install Might Generate Message 'Internal error occurred during "Launching"'

If you installed a Beta release of Nios II version 7.1 prior to installing the full release, you might encounter the error **An internal error occurred during "Launching"**, when building, running or debugging a software application project in Nios II IDE.

Workaround: Close Nios II IDE, open a Nios II command shell, and type `nios2-ide -clean`. The Nios II IDE launches again, and the error does not recur. Launching Nios II IDE with the `-clean` option does not affect any IDE or project settings you have made.

Altera strongly recommends that you uninstall Beta versions of the 7.1 tools before installing the released version of the Nios II EDS v7.1. Also, remove or rename the existing installation directories.

Different Address Mapping for Master Ports Connected to Dual-Port Memories Can Cause Linker Errors

If your instruction master and data master ports are connected to the same dual-port memory and the ports have different addresses, your code fails to run or you experience a linker error. The Nios II IDE does not warn you of the addressing violation.

Workaround: Assign the same address to both ports of the dual-port memory.

Nios II IDE Does Not Support Settings for Projects Imported From the Nios II Software Build Tools Design Flow

For projects created using the new software build flow and imported into the Nios II IDE, the IDE configuration settings have no effect. For example, objdump, compiler, and linker settings made in the IDE are ignored. This behavior occurs because Nios II software build tools projects are not IDE-managed projects. In addition, the make-related options (at **Window** > > > **Preferences** > > > **Nios II** and **Window** > > > **Preferences** > > > **Nios II** > > > **New projects)** do not pertain to imported Nios II software build tools projects. The IDE ignores these options during the build process.

Workaround: Make these settings in the project's makefile.

Using a URL Location for Projects Might Cause an Error Creating Project

You might see an error dialog box saying **Project cannot be created. Reason: Internal Error** when you try to create a new project while pointing to an existing workspace. This error might occur if the path to any project in the workspace is a URL location, for example **file:/F:/Design**. To view the path, right-click on the project and select Properties.

Workaround: Import your existing application and system library projects into a new workspace.

Conflicting File Names Can Create Illegal Project Location Error When Creating New Project

You might receive an **Illegal project location** error message in the IDE if you use the default project name when creating a project in a new workspace. If the project name exists in another workspace, the IDE might not account for that in the new workspace.

Workaround: Change the project name to a name other then the default.

Choosing Separate Exception Stack Option Might Cause Compilation error "UNDEFINED VARIABLE %STACK_POINTER%" When Building a Project

This error occurs if the **Use a separate exception stack option** is turned on, and the exception stack is larger than the memory available for it.

Workaround: On the system library properties page for the project, turn off the separate exception stack or reduce the **Maximum exception stack size** setting.

When Building a Project, the Nios II IDE Reports Problems, But the Build Output in the Console Does Not Contain Any Errors

The Nios II IDE incorrectly reports some linker warnings as errors, with a dialog box saying **Errors exist in a required project**. The Dhrystone software example exhibits this behavior, and recompiling the project again makes the issue go away.

Workaround: If the Console output does not contain errors, then the project actually built fine. On subsequent builds, the linker step is skipped and the errors do not appear, because the project built without error previously.

Makefile reports incorrect number of bytes free for the stack and heap, if the heap is in a different memory region than the stack

Workaround: Do not trust the heap and stack memory report from the makefile if you have placed the heap and stack in different memory regions.

Nios II IDE Unnecessarily Updates the SOPC Builder System File (.ptf)

The Nios II IDE opens the SOPC Builder system file (**.ptf**) by invoking SOPC Builder during certain operations, which might cause SOPC Builder to change the date stamp of the file even though the system is not modified. This behavior might cause problems if you are using a version control system.

Workaround: If you are not using the Nios II C2H Compiler, you can change the PTF file properties to read-only to prevent the IDE from changing the file.

Incorrect Address Assignment for Dual-Port Memory Mastered by Two Different Master Ports

If you have a dual-port memory in your Nios II system, and only the second slave port is mastered by the processor, you might see an overlapping section error during the linking stage of building your software.

Workaround: In SOPC Builder, ensure that the first slave port of the dual-port memory is mastered by the Nios II processor. The processor does not have to master the second port.

Build Errors After Changing Component Names in SOPC Builder

If you rename components in the SOPC Builder system and then regenerate the SOPC Builder system, Nios II IDE system library projects based on that system have build errors.

Workaround: Delete the system library project from the workspace without deleting the contents from the file system, and then import the project again, selecting the regenerated SOPC Builder system. Alternatively, after regenerating the SOPC Builder system, create a new system library project for the SOPC Builder system.

Debugging Projects

GDB Error 256 When Launching Debugger

You might see an `Error 256` message when launching the debugger if you have environment variables that are longer than 1200 characters.

Workaround: Ensure that you do not have any environment variable that exceed 1200 characters.

Previous 7.1 Beta Install Might Cause Internal Error When Launching FS2 in IDE

If you have a Beta version of Nios II EDS 7.1 installed on your machine along with the release version of 7.1, you might experience an FS2 launch error. If you have selected **Use FS2 console window for trace and watchpoint support** in the Nios II IDE debugger settings window, you might receive an internal error while launching.

Workaround: Uninstall any Beta version of the Nios II EDS version 7.1. When installing a released version of the software, uninstall any Beta version first, and remove or rename the Beta installation directories before installing the released version.

Extra Arguments in Debugger for Multiple Launch Configurations

If you have multiple debug launch configuration, you might have issues with the **Additional nios2-download arguments** field. If you delete the contents of the **Additional nios2-download arguments** field in one debugger launch configuration, view a second launch configuration, with a nonempty string, in the **Additional nios2-download arguments** text box, and return to the first launch configuration, the value from the second configuration appear in the first configuration.

Workaround: To remove all arguments, use a single space rather than deleting the entire contents of the field.

Console Window Is Not Updated After ISS Error

After performing a **Run as ISS**, if you receive an ISS error in the console window, the console is not updated subsequently.

Workaround: Close the console window after receiving an ISS error. A new console window opens when a new message is available.

Cannot Locate Source Code in Driver Files Shared by Multiple Projects

If you hit a breakpoint in a driver file, and that driver file is shared with another project that is closed, the Nios II IDE might indicate that it cannot locate the source code.

Workaround: Open the closed system library project and resume debugging.

Memory Window Sets Control Register Values Incorrectly

The memory window might incorrectly set values in memory-mapped control registers. For example, writing 0x1234 to a byte addressed register results in the value 0x3434 being stored in the register. The memory window will show this incorrect value.

Workaround: Use the GDB console window in the IDE, instead of the memory window, to write to the registers. For example, type `set {int} <register address>=0x1234` in the GDB console window.

☞ You must refresh the memory window in order for it to correctly display the target value.

ISS Fails on Designs Containing TSE MAC or SG-DMA Components

You will receive an Internal Error when attempting to perform an ISS simulation of designs containing the TSE MAC or SG-SMA components because the Nios II ISS does not support these components.

Workaround: Remove the TSE MAC and SG-DMA components from your system and perform ISS simulation on the simplified system. You can also simulate the design in ModelSim® or test it on hardware.

"Run as ModelSim" in the Nios II IDE Fails

The **Run as ModelSim** command might fail on Run configurations created in prior versions of the IDE. This problem will not occur for new Run configurations.

Workaround: select a location for the ModelSim tool from the launch configuration dialog box. You can use the Browse button next to the **ModelSim path** group, or type in a path to the ModelSim directory (e.g. **c:/altera/71/modelsim_ae/win32aloem**).

Unable to Display Driver Source Code When Debugging a Custom Component in the Nios II IDE

When debugging a custom component in the Nios II IDE, the debugger might not find the software driver files for that component.

Workaround: You must add a linked folder to an IDE project so that the IDE can find source code files associated with the component. The locations which you must add appear in a generated file called **install2.ptf** in the directory containing the **install.ptf** for the hardware design. Examine this file to find the directory path which you must add. For example, a custom component called **altera_avalon_pwm** is stored in **/data/pwm_component_test**. An excerpt from **install2.ptf** follows:

```
PACKAGE install2
{
   COMPONENT altera_avalon_pwm
   {
      VERSION 2.0
      {
         local = "/data/pwm_component_test";
      }
   }
}
```

To add the path **/data/pwm_component_test** to the IDE, pick the system library for your design, and right-click the system library project, point to **New** > > > **Other** > > > **General** > > > **Folder**, and click **Next**.

Specify the parent project (e.g. **syslib**) in the **Enter or select the parent folder** field.

Click **Advanced**, turn on **Link to folder in the file system**, and browse to or type in the path to the component.

Click **Finish** to link the folder.

Note: Do not link a custom component to more than one system library in the IDE. The breakpoint manager sometimes fails to handle the aliases, which results in breakpoints being hit even when removed from the breakpoint manager.

The Restart Command on the Run Menu Does Not Work

Workaround: Stop the program, then debug it again. If the debugger is hung in an endless loop, use the following bash alias to break the target, then stop it:

```
alias break="kill -2 \`ps -a | grep nios2-elf-gdb | cut -f6 -d' '\`"
```

Programs That Interact With a Terminal Console on Windows Do Not Work

Programs with this behavior work in v6.0 and earlier, but do not work in Nios II IDE v6.1 and later.

The Eclipse platform in v6.1 and later of the IDE (on Windows only) sends the string `\r\n` instead of just `\n` when running or debugging using the Terminal. This behavior can break existing software designs, and it is inconsistent with nios2-terminal, which still just sends `\n`.

Workaround: Change the software running on the Nios II processor to parse for \r as well as \n.

"Step failed. Target is not responding (timed out)" Error Message

The Nios II IDE debugger might hang and report this message if your code contains large arrays declared as local variables on the stack.

Workaround: Place the array and any other large buffers on the heap rather than on the stack.

Incorrect Breakpoint Filtering on Threads

If you enable breakpoint filtering for a thread and later turn off filtering for the thread, the debugger might incorrectly continue to filter the thread.

Workaround: There is no workaround available at this time.

Uninitialized Memory Error When Executing From ISS

In some cases the ISS does not ignore uninitialized memory reads, even when **Uninitialized memory reads** is set to **Ignore** on the ISS Settings tab of the run configuration.

Workaround: There is no known workaround.

nios2-gdb-server Fails to Terminate After Setting a Watch Point

You might be unable to terminate **nios2-gdb-server** after setting a watchpoint in the Nios II IDE debugger and resuming execution past the end of main. You will see an error **Terminate failed**. You will not be able to start the debugger again; you will see a message reading **Another application is using the target processor...** in the Console view.

Workaround: Terminate the **nios2-gdb-server** process manually using the Windows Task Manager.

Watchpoints Do Not Work on a Variable Whose Size Is Not 32 Bits

Workaround: Change the types of global and static local variables to `int`, `long`, or `unsigned long` before setting watchpoints on them.

Debugger Cannot Step Into __sflags(), and Continues Execution Instead

The Nios II IDE debugger is unable to step into some low-level C library functions, such as `__sflags()` (`__sflags()` is called from `_fopen_r()`, called from `fopen()`). If you try to step into such a function, the debugger proceeds as if you had selected **Resume execution**.

Workaround:

- Step over such functions.
- If execution continues after trying to step in, click **Suspend** on the Run menu.

Missing Address and Data in the Trace View

If the trace options **Include load addresses**, **Include store addresses** or **Include data values** are enabled during debug, the load and store address and data will not appear at the first breakpoint in the debugging session. They will appear at subsequent breakpoints.

Workaround: To see load or store addresses and data in the instruction trace prior to `main()`, turn on **Break at alt_main()** located on the **Debugger** tab for your debug configuration.

Cannot Use Watchpoints in the Nios II IDE When the FS2 Console Is Open

Hardware watchpoints do not work in the Nios II IDE when the **Use FS2 console window for trace and watchpoint support** setting is turned on in the **Debugger** tab of the Debug configuration. You will see an error message **The execution of program is suspended because of error.** with details indicating that hardware watchpoints could not be inserted and deleted.

Workaround: If the FS2 console is open, you must use FS2 to control watchpoints. For details, see the FS2 documentation located in <*Nios II EDS install path*>**\nios2eds\bin\fs2\doc**.

Breakpoints on Adjacent Lines of Assembly Fail to Halt the Processor

Setting breakpoints on adjacent lines of assembly code might cause the Nios II processor to stop responding to the debugger. This issue does not affect Nios II cores that do not have hardware breakpoints enabled in the JTAG debug module.

Workaround: When debugging in mixed mode or disassembly view, separate breakpoints by at least one assembly instruction.

Navigating Projects

Nios II IDE Import System Library Dialog Box Does Not Close After Import

When importing a system library project, the Nios II IDE wizard does not close after clicking the **Finish** button. The wizard imports the project successfully after clicking **Finish** once. Clicking **Finish** again results in an error message stating that the project is already created.

Workaround: Click **Cancel** in the **Import Wizard** dialog box after clicking **Finish**.

C/C++ Search Returns Nothing

The C/C++ Search in the Nios II IDE does not return results because a search index is not created at start up.

Workaround: Manually evoke indexing for projects you need to search. Refer to the *Indexing and Searching Project Content* IDE online help topic for instructions.

Internal Error When Double-Clicking on a Large Objdump File

On a Windows PC when opening a large **objdump** file in the Nios II IDE, you might get the following error message: **Unable to create this part due to an internal error. Reason for the failure: Editor could not be initialized.**

Workaround: Adjust the Windows launch arguments for the Nios II IDE editor. Perform the following steps:

1. On the Windows Start menu, browse to the **Nios II EDS 7.1** program icon, right click it, then clicki **Properties**. The **Windows Properties** dialog box appears.

2. In the **Target** field, append `vmargs -Xmx1024m` to the end of the path to the Nios II IDE executable. For example:

```
C:\altera\70\nios2eds\bin\eclipse\nios2-ide.exe -vmargs -Xmx1024m
```

After Switching Workspaces, the Nios II IDE Appears to Freeze While Displaying the Splash Screen

After clicking **Switch Workspace** on the File menu on a Windows machine, a Nios II IDE splash screen appears. Unfortunately, this splash screen obscures a dialog box which asks you to specify the new workspace.

Workaround: Press Alt-Tab to switch applications. Two relevant application icons appear: an Eclipse icon associated with the splash screen and a Nios II IDE icon associated with the **Workspace** dialog box. Select the Nios II icon to bring the dialog box to the foreground.

C/C++ Scanner Does Not Understand Certain C/C++ Constructs

The C/C++ scanner performs C/C++ Search, navigation, open declaration, and parts of content assist. Due to limitations of the C/C++ scanner, these features do not work with the following code constructs:

- Kernighan & Ritchie-style C
- Functions that take a function pointer as an argument

Workaround: If the C/C++ Search fails, use the File Search facility.

Nios II Software Build Tools

This section lists any issues related to the Nios II software build tools. These are new tools introduced in Nios II EDS version 7.1.

Nios II IDE import settings for Nios II software build tools projects

For projects created using the Nios II software build tools and imported into Nios II IDE projects, the IDE configuration settings are invalid. For example, objdump, compiler, and linker settings made in the IDE have no effect. This behavior occurs because Nios II software build tools projects are not IDE-managed projects. In addition, the make-related options on **Window > Preferences > Nios II and Window > Preferences > Nios II > New project**s do not pertain to imported Nios II software build tools projects, and the IDE ignores them during the build process.

Workaround: Specify these settings in the Nios II software build tools project's makefile.

The add_sw_setting Character Tcl Command Produces Improper Quote Format in system.h

The Nios II software build tools tcl command

```
add_sw_setting character system_h_define character_system_h character_system_h a "character_system_h"
```

generates a **system.h** file with contents in double quotes i.e.: `"a"`, instead of the proper single quotes, i.e.: `'a'`.

Workaround: Manually edit the **system.h** file to fix the appropriate line by changing double quotes to single quotes.

Upper-Case File Extensions Not Supported

The Nios II software build tools for applications and libraries (nios2-app-generate-makefile and nios2-lib-generate-makefile commands) do not support source files with certain upper-case extensions. If a file with an upper-case extension is included, the make command stops with no descriptive warning.

Only Nios II assembly language files built by the C preprocessor can have upper-case file extension (**.S**). All C language files must have the extension **.c** or **.h**. C++ language source files must have the extension **.cpp**, **.cxx**, **.cc,** or **.h**.

Workaround: Rename all C language files with the extension **.c** or **.h**. Rename all C++ language files with the extension **.cpp**, **.cxx**, **.cc,** or **.h**.

"make download-elf" Fails for Stratix II 2S60 RoHS Design Examples

You might see error messages when running the 2S60 RoHS example applications with the Software Build Tools. These messages appear due to an incorrect system ID value in the BSP makefile. The error messages are similar to the following:

```
Reading System ID at address 0x021208B8:
    ID value does not match: read 0x072DA598; expected 0x7D1BD35D
    Timestamp value was not verified: value was not specified
The software you are downloading may not run on the system which is currently
configured into the device.  Please download the correct SOF or recompile
```

Workaround: Remove the system ID check from the application makefile. To remove the system ID check, carry out the following steps:

1. Open the application **Makefile,** located in the application project directory.

2. Search for `download-elf` to find the section that needs modification.

3. Remove the `$(SOPC_SYSID_FLAG)` argument from the `$(DOWNLOAD)` command in the following section:

```
.PHONY : download-elf
download-elf : $(ELF)
      @echo Info: Downloading $(ELF)
      $(DOWNLOAD) --go --cpu_name=$(CPU_NAME) $(SOPC_SYSID_FLAG) $(WRITE_GMON_OPTION) $(ELF)
```

The resulting section looks like:

```
.PHONY : download-elf
download-elf : $(ELF)
      @echo Info: Downloading $(ELF)
      $(DOWNLOAD) --go --cpu_name=$(CPU_NAME) $(WRITE_GMON_OPTION) $(ELF
```

Rebuild the application.

C2H Compiler

This section lists issues related to the Nios II C-to-Hardware Acceleration (C2H) Compiler.

C2H Regenerates Hardware Accelerator Even Though No Logic Changed

This problem can result from changes to files included by the C file containing the accelerated function. C2H fails to check that the HDL generated matches the previously generated HDL causing the system to be regenerated.

Workaround: To avoid this issue move the information in the include file specific to the hardware accelerator into a separate include file. This workaround prevents regeneration of the system when the HDL is unchanged.

Error: c2h_accelerator_base_addresses.h: No Such File Or Directory

When a C2H accelerator is compiled for the first time, the following compile-time error can result if the **Analyze all accelerators** option is selected: **c2h_accelerator_base_addresses.h: No such file or directory**.

Workaround: Click **Build software and generate SOPC Builder system** and build once before building with the **Analyze all accelerators** option.

Pointer Dereferences to Volatiles

The C2H compiler treats pointer dereferences to a volatile type as if they alias all other pointer dereferences. Pointers that are restrict-qualified are treated the same way. For example, in the cases below, the two loops cannot be scheduled concurrently because the volatile qualification requires that a dependency exist between the two dereference might cause incorrect behavior.

```
volatile int * restrict fifo_rd = FIFO_RD_BASE;
volatile int * restrict fifo_wr = FIFO_WR_BASE;
for ()
{
      *fifo_wr = ....;
}
for ()
{
      ... = *fifo_rd;
}
```

Workaround: Divide the function into multiple IRQ-enabled accelerators that are launched concurrently from the CPU, and use FIFOs to communicate between them.

Java Heap Space Exception if Quartus II Compilation is Enabled

You might receive the following error during Quartus II compilation of your design containing a C2H accelerator if you selected the **Build software, generate SOPC Builder system, and run Quartus II compilation** in the C2H Compiler settings window.

```
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space

make: *** [c2h_hdl-t] Error 1
```

Workaround: Select the **Build software and generate SOPC builder system** option in the C2H Compiler settings window and launch the Quartus II compilation manually afterward.

C2H Accelerators Time Out Immediately When Untethered

If you do not have a C2H Compiler license and close the OpenCore Plus status box while running a design containing a C2H accelerator, the design times out immediately rather than allowing the one hour grace period.

Workaround: Remain tethered during evaluation and do not close the OpenCore Plus status box. Alternatively, you can purchase a C2H Compiler license.

Accelerator Generation Failure If Tools Are Installed in Path With Spaces

If the path to your installation of the Altera Design Suite contains spaces, the C2H Compiler fails to generate the accelerators.

Workaround: Reinstall the tools to a path containing no spaces.

C2H Compiler Out-of-Memory Error

The C2H Compiler might generate an out-of-memory error at compile time for code that initializes a large local array.

Workaround: Initialize the memory with the Nios II processor and access it through pointers in the accelerator.

C2H Compiler Does Not Accelerate Subfunctions Located in a Separate File

When accelerating a function in a file, the C2H Compiler cannot link subfunctions that are defined in a different file.

Workaround: Include all subfunctions called by the accelerated function within the same source code file.

Incorrect Results From Logical or Conditional Operation With Side-Effects

The C2H Compiler always evaluates both operands of logical (&&, ||) and conditional (?:) operators. This is different from expected ANSI C behavior, for which operands are evaluated left-to-right, and unnecessary operands are skipped. For example, in the expression `(i-- && j--)`, if the value of `i` is zero, the right-hand-side (RHS) expression should not evaluate (i.e., `j` should not be decremented). However, this C2H Compiler erroneously evaluates both sides unconditionally, causing `j` to be decremented. The following example expressions could suffer from the same issue: `(i-- || j--)`, `(cond ? i-- : j--)`

Workaround: Use logical and conditional operations whose operators have no side effects. Side effects include pre-/post-fix increment operations (`++`, `--`), memory operations (`*`, `[]`, `.`, `->`), and function calls.

Closed System Library While Working With the C2H Compiler

The C2H Compiler requires the system library to obtain important details about the system, and cannot function if the system library is closed.

Workaround: Ensure that the system library project in the Nios II IDE is open prior to building an application project that contains a hardware accelerator.

Launch SOPC Builder Button in C2H View

When the Nios II IDE workspace contains multiple projects with multiple system libraries, the incorrect SOPC Builder system might open when you click the Launch SOPC Builder in the C2H view.

Workaround: Keep only one system library project open at a time while using the C2H Compiler.

Build Clean Causes Build Failure

Performing a clean build on a Nios II IDE project that contains a hardware accelerator can cause the next build to fail in the IDE, because the clean build erroneously deletes a file required by the C2H Compiler.

Workaround: Do not perform a clean build on projects that use hardware accelerators. If you have already performed a clean build, recompile with option **Build software, generate SOPC Builder system, and run Quartus II compilation** to regenerate the necessary files.

Multiple Clock Domains Causes Hardware Accelerator to Fail

If a hardware accelerator and the components connected to its master ports are in different clock domains, the accelerator might behave incorrectly.

Workaround: Assign a single clock to a hardware accelerator and all the slave ports it connects to. It is acceptable for the system to contain multiple clock domains.

Array Elements of structs Do Not Copy Correctly

C2H hardware accelerators do not correctly copy array elements of structs. For example:

```
typedef struct my_struct {
  int a;
  int b;
  int buf[BUF_SIZE];
}MY_STRUCT;

MY_STRUCT struct_a = {1, 2, {3, 3, 3, 3}};
MY_STRUCT struct_b = {9, 8, {7, 7, 7, 7}};

struct_a = struct_b;
```

In this example, the `a` and `b` elements of the struct copy correctly, but the `buf` element does not. After this assignment, `struct_a` equals `{9, 8, {3, 3, 3, 3}}`.

Workaround: Copy the array elements explicitly, as follows:

```
{
  int i=0;
  do
    {
      struct_a.buf[i] = struct_b.buf[i];
      i++;
    } while (i<LENGTH_OF_BUF_ELEMENT)
}
```

Simulating on ISS Not Supported

The Nios II instruction set simulator (ISS) does not support custom SOPC Builder components, and therefore cannot simulate systems that use hardware accelerators. You might get the following internal error if attempting to simulate using the ISS:

```
Internal Error (unhandled exception) in file cosim_main.cpp
```

Workaround: Run the program on a hardware system that includes the hardware accelerator.

Hardware Accelerators Remain After Deleting the Software Project

If a system contains C2H hardware accelerators, deleting the software project that defines the accelerators does not remove the accelerators from the hardware system, and the accelerator logic remains in the SOPC Builder system.

Workaround: To remove an accelerator from a system, delete the accelerator from the C2H view in the Nios II IDE first, and then recompile the software project. The C2H Compiler then removes the accelerator from the SOPC Builder system. Once the compilation is complete then the software application can be deleted from the workspace.

Changing Build Configurations Produces Unexpected Results

The C2H Compiler does not support multiple build configurations (e.g. Release or Debug) in the Nios II IDE. After creating one or more accelerators in a particular configuration, the C2H Compiler produces undefined results if you switch to a different build configurations and create more accelerators.

Workaround: For a specific SOPC Builder system and Nios II IDE project, specify C2H accelerators in only one build configuration. You can use multiple build configurations, as long as only one configuration specifies C2H Compiler settings.

Flash Programmer

This section lists any issues relating to the Nios II IDE flash programmer.

Delay When Creating New Flash Programmer Configuration

You might experience a delay of several seconds when creating a new Flash Programmer configuration in the IDE.

Workaround: There is no workaround available at this time.

"No such file or directory" Error for a Project Stored in a Path Containing Spaces

You might receive this spurious error message when programming flash for a project stored in a path containing spaces. The flash programmer does not correctly handle spaces in the directory path. However, this error is benign, because flash programming completes successfully.

Workaround: None required.

elf2flash File Size Limit

The elf2flash utility supports **.elf** files up to approximately 24 MBytes in size. The elf2flash utility might fail with error **java.lang.OutOfMemoryError** on files larger than 24 MBytes.

Workaround: You can either lower the number of symbols in your elf file by turning off debug symbols, or specify less initialized data in the application.

When Used With a Multiprocessor Nios II System, the Nios II Flash Programmer Might Produce an Error

The Nios II Flash Programmer might produce the following error:

```
There are two or more Nios II processors available which match the values
specified. Please use the configuration dialog to pick one, or specify the --
device and/or --instance parameters on the command line.
```

Workaround: In the Nios II IDE on the **Flash Programmer** dialog box, make sure you specify an appropriate `--instance=`<*correct instance value*> argument in **Additional nios2-flash-programmer arguments**. If the Quartus II project has been recompiled since the flash configuration's creation, click **Load JDI File**. If using command line mode, add `--instance=`<*correct instance value*> to the command line. There are two ways to find the correct value of the instance ID for a processor. The easiest is to use the Nios II IDE to create a sample flash programmer script. Alternatively, open <Quartus II project name>**.jdi**, in the Quartus II project directory. Locate the Nios II processor node by finding a value of `hpath` containing <*processor module name*>=. The instance ID is specified as `instance_id`.

For further details, refer to the *Nios II Flash Programmer User Guide*.

Download Cables & Debug Hardware

This section lists any issues related to download cables and other debug hardware.

Communication Errors During Run/Debug Sessions Using Older Download Cables

Debugging with the following Altera download cables might fail, due to electrical noise-related JTAG communication failures: USB-Blaster™ Rev A, ByteBlaster™, ByteBlasterMV™, ByteBlaster II, and MasterBlaster™ cables.

Currently, the only fully supported cable for downloading, debugging, or communicating with Nios II systems is the USB-Blaster Rev B cable or later. Revision B cables are clearly labeled as Revision B. (Revision A cables have no revision label.)

Workaround: Use a USB-Blaster Rev B cable. Older cables can be used, but they might encounter JTAG communication failures.

Development Boards

This section lists any issues related to Altera development boards.

Intermittent Failures While Accessing CompactFlash Card

The Nios II Development Kit version 5.0 and higher includes a CompactFlash controller peripheral suitable for interfacing to CompactFlash cards in True IDE mode on Nios development boards. In order for True IDE mode to operate, CompactFlash cards require that the ATASEL_N input be driven to ground during power-up.

The CompactFlash controller peripheral includes a configurable power register used to power-cycle CompactFlash cards in Nios II software through a MOSFET on the Nios development boards. However, in certain development boards, power to the CompactFlash card does not turn off completely during this power cycle operation. Because of this condition, the CompactFlash might not sample the ATASEL_N pin during the power-cycle operation after FPGA configuration when this pin is driven to ground. Instead, the CompactFlash card might sample the ATASEL_N pin when power is first applied to the development board, when I/O are not yet driven by the FPGA (before FPGA configuration).

Workaround: If you encounter errors with CompactFlash when using the Nios development boards, try one of the following:

- Use a different CompactFlash card. Certain cards are more susceptible to the power-cycling issue than others.

- Modify the Nios development board. This is recommended for users who are familiar and comfortable with board-level modifications. Disconnect pin 9 (ATASEL_N) on the CompactFlash socket on your Nios development board and tie this pin to ground.

☞ The CompactFlash socket uses a staggered numbering on the pins (starting from pin 1: 1, 26, 2, 27, ...); refer to the CompactFlash Association specification for right-angle surface-mount connectors for exact specifications on this connector. This modification permanently enables True-IDE mode operation.

Toolchain (gcc, gdb, etc.)

This section lists any issues related to the Nios II compiler toolchain, such as gcc and gdb.

Breakpoints in C++ Constructors Fail to Halt the Processor

Breakpoints set in a C++ constructor might not halt the processor due to a widespread GNU GCC, GDB issue. This is not a Nios II IDE-specific issue.

Workaround: You can work around this issue by moving all of your constructor source code into another class method, called init. Then invoke this method from within the constructor.

Target Software

This section lists any issues related to software or drivers that target the Nios II processor.

Lightweight IP (lwIP) Failure When Using DHCP

The Lightweight IP stack might experience intermittent failures when using DHCP to acquire an IP address. When the failure occurs, the following is printed to stdout:

```
Assertion "dhcp_create_request: dhcp->p_out == NULL" failed at line 1283 in
/cygdrive/c/altera/61b169/nios2eds/components/altera_lwip/UCOSII/src/downloads/lwi
p-1.1.0/src/core/dhcp.c
```

Workaround: If possible, Altera recommends switching to the NicheStack TCP/IP Stack - Nios II Edition introduced in Nios II 6.1. If not, the following workarounds can be employed.

- Power-cycle the target board and retry DHCP negotiation, which usually results in correct acquisition of an address from DHCP, assuming that the DHCP server is able to allocate IP addresses.
- Use a static IP address and disable DHCP.

stdin, stdout, and stderr Do Not Work in MicroC/OS-II Applications When Using Small C Library Option

MicroC/OS-II applications using `stdin`, `stdout`, and `stderr` fail to operate when using the small C library.

Workaround: Disable the small C library option.

malloc(), realloc() Failures With MicroC/OS-II

When using the MicroC/OS-II RTOS, calls to `malloc()` and `realloc()` might fail if successive calls to `malloc()`or `realloc()` within a MicroC/OS-II task occur after changing the task priority of the task in which a memory block is originally allocated.

Workarounds:

- Allocate and/or reallocate memory blocks outside of MicroC/OS-II tasks, before task switching starts. This makes it possible to change thread priorities at runtime.
- Allocate fixed areas of memory using arrays (rather than using `malloc()`) before task switching starts. This makes it possible to change thread priorities at runtime.
- Allocate memory using `malloc()` or `realloc()` from a MicroC/OS-II task. You can change task priorities at runtime, but only for tasks that have not used `malloc()` or `realloc()`.

cout From MicroC-OS/II Task Will Not Send Data to stdout

If neither `printf()` or `cout` is used from `main()` before tasks are started, `cout` does not work from a task.

Workaround: Add the following C++ code to the beginning of `main()`:

```
std::ios_base::sync_with_stdio(false);
```

Problems Using HAL Drivers With Toshiba Flash

The HAL CFI Flash driver might not work for Toshiba flash memory that claims to be CFI compliant.

Workaround: In the **altera_avalon_cfi_flash_table.c** file, change the `#define READ_ARRAY_MODE` from `(alt_u8)0xFF` to `(alt_u8)0xF0` and rebuild the project.

Creating New Custom HAL Components in the Nios II IDE

When you first create a component's **inc** directory or HAL header file, you might first need to perform a clean build (i.e., rebuild) of existing system library projects for the new files to be detected.

Legacy SDK

Support for the Legacy SDK mode was removed in version 6.0 of the Nios II Embedded Design Suite.

SOPC Builder and Quartus II Software

This section lists any issues related to the Quartus II software or SOPC Builder that specifically affect Nios II designers.

For further information on the Quartus II software, refer to the latest Quartus II release notes on the Altera web site:

http://www.altera.com/literature/lit-qts.jsp

Example Designs

This section lists issues related to the example designs included with the Nios II Embedded Design Suite.

Hardware Designs

Incorrect SSRAM & SDRAM PLL Phase Shift in Stratix II 2s60 RoHS Standard Design

In the standard design targeting the Nios II Stratix II 2S60 (RoHS) development boards, the PLL phase shift used to implement the SSRAM and DDR SDRAM clocks is incorrect. This problem might cause timing violations when reading from or writing to SSRAM or DDR.

If you wish to create a custom design that uses SSRAM on Nios II Stratix II 2S60 (RoHS) edition boards, Altera recommends that you set the PLL phase shift for generating the SSRAM clock to –3.38 ns in Normal PLL mode. If you wish to create a custom design that uses the legacy DDR SDRAM on Nios II Stratix II 2S60 (RoHS) edition boards, Altera recommends that you set the PLL phase shift for generating the DDR write clock to 270 degrees in Normal PLL mode.

To change the PLL settings, perform the following steps:

1. Open the standard design in SOPC Builder.
2. Double click the component instance named **pll** to launch the PLL MegaWizard interface.
3. Click **Launch Altera's ALTPLL MegaWizard** to launch the MegaWizard Plug-In Manager.
4. Select the **Output Clocks** page and then the **clk c1** page, used to generate the SSRAM clock.
5. Ensure that **Clock phase shift** is –3.38 ns.
6. Click the **clk c2** page, used to generate the DDR write clock.
7. Ensure that **Clock phase shift** is 270 degrees.
8. Click **Finish** to save changes and exit the PLL MegaWizard interface.
9. Click **Finish** to save settings to the PLL instance in SOPC Builder.
10. Regenerate the system in SOPC Builder and recompile in the Quartus II software. Refer to the **readme.txt** file in the design example folder for information on compiling designs containing the DDR controller.

For further information refer to **ssram_interface_readme.html**, located in the **<Quartus II installation directory>/sopc_builder/documents** folder. This document discusses the SSRAM timing analysis methodology in detail. Additionally, *AN 411: Understanding PLL Timing for Stratix II Devices* discusses clock phase shift calculations and assignments for PLLs in Stratix II devices.

Incorrect SSRAM PLL Phase Shift in Stratix II 2s60 RoHS and Cyclone II 2C35 TSE SG-DMA Designs

In the TSE SG-DMA design targeting the Nios II Cyclone™ II 2C35 and Nios II Stratix II 2S60 RoHS development boards, the PLL phase shift used to implement the SSRAM clock is incorrect. This problem might cause timing violations when reading from or writing to SSRAM if you increase the system clock speed above that of the example designs (85MHz), while using two clock cycles of read latency in the SSRAM interface.

While most Altera example designs targeting these boards run at 85Mhz, the TSE SG-DMA design targeting the Stratix II 2S60 RoHS board runs at 100MHz. Although no SSRAM failures have been observed with this board and example design in their current configurations, you must regenerate the design for reliable SSRAM operation.

☞ No action is required if you wish to continue using the example designs at their initial clock speed of 85MHz. However, if you wish to create a custom design that uses SSRAM and runs above 85Mhz, or if you wish to increase the clock speed of an Altera example design above 85MHz, Altera recommends that you adjust the PLL phase shift for generating the SSRAM clock. On Nios II Cyclone II 2C35 and Nios II Stratix II 2S60 RoHS edition boards, set the phase shift to –3.38 ns, in Normal PLL mode.

To change the PLL setting, perform the following steps:

1. Open the example design in SOPC Builder
2. Double click the component instance named **sys_pll** and launch the PLL MegaWizard interface.
3. Click **Launch Altera's ALTPLL MegaWizard** to launch the MegaWizard Plug-In Manager.
4. Select the **Output Clocks** page and then the **clk c2** page, used to generate the SSRAM clock.
5. Change **Clock phase shift** from 4.8 ns to –3.38 ns.
6. Click **Finish** to save changes and exit the PLL MegaWizard interface.
7. Click **Finish** to save settings to the PLL instance in SOPC Builder.
8. Regenerate the system in SOPC Builder and recompile in Quartus II.

For further information, refer to **ssram_interface_readme.html**, located in the <*Quartus II installation directory*>**/sopc_builder/documents** folder. This document discusses the SSRAM timing analysis methodology in detail.

Additionally, *AN 411: Understanding PLL Timing for Stratix II Devices* discusses clock phase shift calculations and assignments.

Software Designs

Dhrystone BSP Example Fails When Run on the Fast Hardware Example

You might see the following output when running the Dhrystone BSP example in the IDE after creating and building it via command line.

```
  Int_Comp:          36608
        should be:   18
  Str_Comp:          á"
nios2-terminal: exiting due to ^D on remote
```

This error is caused by settings in the **create-this-bsp** script that cause the application to overrun the amount of memory available in the Fast design.

Workaround: To run the Dhrystone example successfully with the Software Build Tools, edit the **create-this-bsp** script located in the <*hardware example design*>**/software_examples/bsp/hal_dhrystone** folder. Carry out the following steps:

1. If the BSP already exists, remove the **settings.bsp** file and run `make clean` in the BSP folder.
2. Change the following line:

   ```
   --set hal.enable_lightweight_device_driver_api true \
   ```

   to:

   ```
   --set hal.enable_reduced_device_drivers true \
   ```

3. Run the **create-this-bsp** script. (Alternatively, run the **create-this-app** script, found in the Dhrystone application folder.)

The BSP is regenerated with the new setting.

Software Template Error With Non-Ethernet Hardware Design

You might get an Internal Error if your hardware design does not contain an Ethernet MAC, and you try to create a software example from one of the following templates:

- host_filesystem
- simple_socket_server
- web_server
- zip_filesystem

These templates check for components in the system library and incorrectly test for the existence of a MAC.

Workaround: Create the system library project for your hardware design before creating your application project from one of the templates. In the system library properties, turn on the Host File System feature by carrying out the following steps:

1. Right-click the system library project and select **Properties**.
2. Click the **Software Components** button, located in the bottom left corner of the **System Library Properties** page.
3. Select **Altera Host Based File System.**
4. Turn on **Add this software component**.
5. Accept the default mount point shown in the **Mount-point** box.

When creating your application project, select this system library.

RAM Test Failure When Running Memory Test Software Template on the ISS

An issue in the instruction set simulator (ISS) model of the JTAG UART can cause a console communication error during the RAM test when running the Memory Test software template on the ISS.

Workaround: There is no workaround available at this time.

Networking Examples

If you are running a networking example design and you are asked for a nine-digit number after the letters **ASJ**, and your Nios II development board does not have a sticker with a nine-digit number after the letters **ASJ**, please enter a unique nine-digit number when prompted. Ensure that this number is unique to each Nios board connected to your network to avoid network address conflicts.

Hardware Simulation

This section lists issues related to simulating Nios II processor systems on an RTL simulator, such as the ModelSim simulator.

Error: "UNC paths are not supported. Defaulting to Windows directory."

If you launch ModelSim from a working directory that is mapped via a UNC path (a path that starts with `//` instead of drive letter), you receive this error message in SOPC Builder. This error occurs because ModelSim is calling a command shell, which does not support UNC paths.

Workaround: Map the UNC path to a drive letter and use the drive letter to reference the working directory in the launching shell.

Simulation Failure if Reset Address is Set to EPCS

Running ModelSim RTL simulation of a Nios II system fails if the reset address of the Nios II processor is set to an EPCS Serial Flash Controller because there is no simulation model for it.

Workaround: To simulate your system, temporarily set the Reset Address of the Nios II CPU to the memory in which your application code resides (for example, SDRAM), then regenerate the system in SOPC Builder and run RTL simulation again. Before booting the Nios II CPU from EPCS flash on your target board, change the Nios II Reset Address back to the EPCS Controller peripheral and regenerate the system in SOPC Builder and recompile in the Quartus II software to produce an updated FPGA configuration file with the Nios II CPU booting from EPCS flash.

Uninitialized BSS Variables in Simulation

If your program reads the value of an uninitialized BSS variable during HDL simulation, and the BSP (system library) is compiled with the **ModelSim only, no hardware support** property enabled in Nios II IDE, a warning appears about unfiltered data being `'x'`. This warning appears because when this property is enabled, the code that clears the BSS memory region is omitted to speed up HDL simulation so this memory region is uninitialized. The BSS region contains global and static local variables that are not initialized by the application so they default to a value of zero. When the Nios II CPU reads uninitialized variables, it displays a warning and converts any of the bits of the uninitialized data to zero which correctly mimics the effect of the missing BSS clearing code. The

HAL code that executes before and after `main()` might use BSS variables, so these warnings might appear even if your application does not use the BSS.

ModelSim Fails to Load Large Memory Models

The ModelSim software might fail to load simulation models for memory arrays larger than 128M bytes, halfwords or words in size. If the sum of the following parameters is greater than 27, the ModelSim software fails to load:

- Address bits (i.e. 14)
- Column bits (i.e. 11)
- log2(banks) (number of banks is usually 4, so this term is usually 2)
- log2(chip selects) (number of chip selects is usually 1, so this term is usually 0)

Workaround: If you do not need to simulate the entire memory space, simulate using a smaller SDRAM than the SDRAM implemented in hardware.

Documentation Issues

This section lists errors, unclear descriptions, or omissions from current published specifications or product documents.

Hardware Development Tutorial Example Does Not Run on Nios Cyclone 1C20 Board

The **count binary** application used in the tutorial hangs if run on the 1C20 development board.

Workaround: Modify the **count_binary.c** file by making lines 330 and 331 conditional, as follows:

```
#ifdef LCD_DISPLAY_NAME
    lcd = LCD_OPEN();
    lcd_init( lcd );
#endif
```

Rebuild and run the software again.

Nios II IDE Help Topic - About Nios II IDE Projects on Linux Mozilla

The expandable Help text for the **Nios II C/C++ Application** does not function correctly on Mozilla 1.4.3 under Linux.

Workaround: The missing information is as follows:

> A Nios II C/C++ application project contains a C/C++ program, usually including a project's `main()` function. Building a Nios II C/C++ application project results in an executable file (**.elf**) that you can run on target hardware, the Nios II instruction set simulator (ISS), and the ModelSim hardware simulator. A Nios II C/C++ application project depends on a single system library project and might reference functions in a Nios II C/C++ library project.
>
> A Nios II C/C++ application project is a Nios II IDE managed make project. The Nios II IDE creates the necessary makefiles and manages the project for you.

Refer to **www.altera.com/literature/ug/ug_nios2_ide_help.pdf** for complete text of the Nios II help system.

Arria GX Support Incorrectly Listed in Documentation HTML Page

In the Nios II 7.1 Documentation Launchpad, the **Example Designs** page incorrectly states that a fast design exists targeting the Arria GX family. Although Nios II release 7.1 supports the Arria GX family, it does not include a fast design for Arria GX devices.

Workaround: Use an existing fast design targeting a Stratix II device and change the device settings. For information about device settings, refer to the *Device Page* topic in the Quartus II software online help.

GNU Assembler Does Not Accept the --defsym Flag

According the GNU documentation, you can set an assembler definition by using the `--defsym` flag, but it does not work in the following form: `--defsym MY_VAR=1`

Contact Information

For more information, contact Altera's mySupport website at **www.altera.com/mysupport**. Click **Create New Service Request**, and click the **Product Related Request** form.

Revision History

Table 1 shows the revision history for the Nios II Embedded Design Suite v7.1 Errata Sheet.

Table 1. Nios II Embedded Design Suite v7.1 Errata Sheet Revision History

Version	Date	Summary
1.0	May 2007	First release
1.1	May 2007	• Build Option Not Functional for BSPs Created Using the Software Build Flow • GDB Error 256 When Launching Debugger • "make download-elf" fails for Stratix II 2S60 RoHS Design Examples • Incorrect SSRAM & SDRAM PLL Phase Shift in Stratix II 2s60 RoHS Standard Design • Incorrect SSRAM PLL Phase Shift in Stratix II 2s60 RoHS and Cyclone II 2C35 TSE SG-DMA Designs • Dhrystone BSP Example Fails When Run on the Fast Hardware Example • Internal Error Using Software Templates With Hardware Designs That Do Not Contain an Ethernet MAC • Hardware Development Tutorial Example Does Not Run on Nios Cyclone 1C20 Board • Nios II IDE Help Topic - About Nios II IDE Projects on Linux Mozilla • Arria GX Support Incorrectly Listed in Documentation HTML Page

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com
Applications Hotline:
(800) 800-EPLD
Literature Services:
literature@altera.com

Nios II Embedded Design Suite 7.0 Release Notes

March 2007, Version 7.0 Release Notes

This document lists the release notes for the Nios® II Embedded Design Suite (EDS) version 7.0.

Table of Contents:

New Features & Enhancements

The Nios II EDS version 7.0 addresses issues found in previous releases, and adds the following primary features:

- Cyclone III device family support
- Instruction master burst support

The sections below provide a detailed list of all product updates, including known issues fixed in this release.

Device & Host Support

This release supports the following Altera® FPGA families:

- Cyclone™ III
- Stratix® III
- Stratix II and Stratix II GX
- Stratix and Stratix GX
- Cyclone II
- Cyclone

The host system requirements for the Nios II EDS are based on the requirements for the Altera Complete Design Suite. Refer to *Quartus II Installation & Licensing for PCs* or *Quartus II Installation & Licensing for UNIX and Linux Workstations* manual.

Installation and Licensing Instructions

For installation instructions, refer to *Quartus II Installation & Licensing for PCs* or *Quartus II Installation & Licensing for UNIX & Linux Workstations* on the Altera website at www.altera.com/literature/lit-qts.jsp.

Using Previously Installed Versions of the Nios II EDS

SOPC Builder and the Nios II IDE refer to the most recently installed version of components (such as the Nios II processor and peripherals) and their software drivers. To revert to a prior version of the Nios II development tools, you can reinstall the previous version of tools or modify the following environment variables.

- `SOPC_BUILDER_PATH` - Ensure that `SOPC_BUILDER_PATH` points to the installation directory of the desired Nios II version and no other Nios II versions.
- `SOPC_KIT_NIOS2` - Ensure `SOPC_KIT_NIOS2` points to the installation directory of the desired Nios II version and no other Nios II versions.

If you have multiple versions of the Quartus II software installed, launch the supported version of Quartus II to ensure that the `QUARTUS_ROOTDIR` environment variable is updated.

Nios II Processor Cores

This section describes changes to the Nios II processor cores.

Instruction master burst support

The Nios II instruction master now optionally supports Avalon burst transfers for cores that include an instruction cache. You can enable this option in the Nios II wizard. Refer to the *Nios II Processor Reference Handbook* for details.

SOPC Builder

This section describes changes to SOPC Builder which affect Nios II designers. For complete revision history of SOPC Builder and the Quartus II software, refer to the release notes for the Quartus II software version 7.0. The *Quartus II Handbook, Volume 4: SOPC Builder* contains complete documentation for SOPC Builder.

There are no major changes to SOPC Builder that affect the Nios II processor this release.

Nios II IDE

This section describes changes to the Nios II integrated development environment (IDE).

There are no major changes to the Nios II IDE in this release.

Nios II C-to-Hardware Acceleration (C2H) Compiler

The Nios II C-to-Hardware Acceleration (C2H) Compiler was introduced in the Nios II EDS version 6.0. The C2H Compiler is a tool that allows you to create custom hardware accelerators directly from ANSI C source code, which can often improve the execution performance by an order of magnitude. Documentation on the C2H Compiler is available on the Nios II literature page, http://www.altera.com/literature/lit-nio2.jsp.

This section describes changes to the C2H Compiler.

There are no major changes to the C2H Compiler in this release.

Flash Programmer

This section describes changes to the flash programmer in the Nios II IDE.

There are no major changes to the Flash Programmer in this release.

Target Software

This section describes changes to Altera-provided target software which runs on the Nios II processor, such as the hardware abstraction layer (HAL) system library.

There are no major changes to the Nios II target software in this release.

Example Designs

Hardware Example Designs

There are no major changes to hardware example designs in this release.

Software Example Designs

There are no major changes to software example designs in this release.

Errata Fixed in this Release

This section lists errata items from previous releases which are fixed in version 7.0.

Command-line utilities don't display any output

Command-line utilities for creating and manipulating Nios II IDE projects on Windows might not display any output, even though the command completed successfully. The following commands function correctly but do not produce output to stdout:

- nios2-build-project
- nios2-create-application-poject
- nios2-create-system-library
- nios2-delete-project
- nios2-import-project

Communication issue with multi-CPU or multi-JTAG UART designs

Designs that contain multiple CPUs or multiple JTAG UARTs might not be able to communicate with the target when running, debugging, or programming flash memory. Designs which have only a single CPU and JTAG UART are not affected by this issue.

'Additional nios2-terminal arguments' field is not functional

On the **Target Connection** tab of the Run or Debug launch configuration dialog boxes, the **Additional nios2-terminal arguments** field is not functional. Entering commands in these fields will have no effect.

Accelerator returns -1 if C/C++ project doesn't rebuild before running

A call to an accelerated function might return the invalid result -1 if you do not build your C application project before running it on hardware. The Nios II IDE does not detect that a new FPGA configuration file (**.sof**) needs to be downloaded if the system library is not rebuilt, which might result in attempting to execute an accelerated function on hardware that does not contain the hardware accelerator.

GNU documentation is missing

The GNU documentation was erroneously omitted from the Nios II EDS v6.1 installation.

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com
Applications Hotline:
(800) 800-EPLD
Literature Services:
literature@altera.com

Nios II Embedded Design Suite 7.0 Errata Sheet

March 2007 Errata Sheet

This document addresses known errata and documentation issues for the Nios® II Embedded Design Suite (EDS) version 7.0. Errata are functional defects or errors, which might cause the product to deviate from published specifications. Documentation issues include errors, unclear descriptions, or omissions from current published specifications or product documents. Errata items discovered after the release of Nios II EDS version 7.0 are marked with the date the items were added to this document.

Table of Contents:

For the most up-to-date errata for this release, refer to the errata sheet on the Altera® website: **http://www.altera.com/literature/es/es_nios2eds_70.pdf**

Nios II Processor Core

This section lists any issues related to the Nios II processor cores.

Nios II/e core burst support

The Nios II wizard incorrectly allows you to select burst support for the instruction master in the Nios II/e core. Instruction master burst support in Nios II requires an instruction cache. The Nios II/e core does not include an instruction cache, therefore, burst support can not be enabled.

Workaround: Do not select burst support for Nios II/e cores.

Interrupt vector custom instruction in multi-processor systems

In designs containing multiple Nios II processors, enabling the interrupt vector custom instruction in more than one of the processors causes the following errors during Quartus® II compilation:

```
Error (10430): VHDL Primary Unit Declaration error at
interruptvector_cpu_1.vhd(26): primary unit "interrupt_vector" already exists in
library "work"
Error (10523): Ignored construct interruptvector_cpu_1 at
interruptvector_cpu_1.vhd(64) due to previous errors
Error: Node instance "the_interruptvector_cpu_1" instantiates undefined entity
"interruptvector_cpu_1"
```

Workaround: Apply the patch provided at *http://www.altera.com/support/kdb/solutions/rd02052007_136.html*, regenerate your SOPC Builder system, and recompile in Quartus II.

"nios_cpu: Unknown Break Location nios_cpu/jtag_debug_module" error message

If you modify the Nios II core in an existing system by disabling the JTAG debug module in the Nios II configuration wizard, you might receive the error message "**nios_cpu: Unknown Break Location nios_cpu/jtag_debug_module**" during SOPC Builder generation.

Workaround: Delete the Nios II core from the system and re-add it, ensuring that the JTAG debug core is disabled in the wizard.

VHDL sensitivity list warnings during Quartus II synthesis

You might receive warnings similar to the following in the Quartus II software when compiling a VHDL Nios II system containing the JTAG debug module.

```
Warning (10492): VHDL Process Statement warning at
cpuname_jtag_debug_module.vhd(254): signal "usr1" is read inside the Process
Statement but isn't in the Process Statement's sensitivity list
Warning (10492): VHDL Process Statement warning at
cpuname_jtag_debug_module.vhd(254): signal "ena" is read inside the Process
Statement but isn't in the Process Statement's sensitivity list
```

These warnings are benign and can be ignored. The warnings are a result of the `ena` and `usr1` signals not being included in the debug module's sensitivity list.

Double-precision Floating-Point operations with floating-point custom instructions

Calls to double-precision floating-point functions in math.h return less-precise results on Nios II processors using the floating-point custom instruction. Floating-point constants are forced to single-precision when the floating-point custom instruction is present, which affects the constants for the double-precision floating-point functions in libm.

Peripherals

This section lists any issues related to the Altera embedded peripherals included in the Quartus II software.

JTAG UART is unstable after device-wide reset

If the DEV_CLRn pin on the FPGA input has been assigned (in Quartus® II software) to generate a device-wide reset, and the FPGA is reset while the JTAG UART is active, then the JTAG UART might become unstable.

Workaround: Do not use the DEV_CLRn function on the FPGA. Turn off the Enable device wide reset (DEV_CLRn) setting in Quartus II software.

Host Platform

This section lists any issues related specifically to the host platform.

Windows XP: Internal error when repeatedly stepping into with debugger

If you repeatedly use the Step Into command while debugging on Windows XP, you might receive the internal error, "Retrieving Children: An internal error occurred during: Retrieving Children".

Workaround: End your debug session, then re-open it and resume debugging. Reduce the frequency of using the step into command.

Linux: Debugging with the Nios II ISS target can cause a process leak on Linux

If you try to interrupt or terminate a debug session targeting the Nios II instruction set simulator (ISS), you might see an error message "Interrupt Failed or Terminate Failed". This means that the nios2-iss process failed to terminate. The debug session appears to have terminated in the IDE, but the nios2-iss process still remains alive.

Workaround: Open a command shell and kill the nios2-iss process.

Linux: The Quartus II stand-alone programmer is not supported on Linux

There is no Quartus II stand-alone programmer for Linux. As a result, in the Nios II IDE the Quartus II Programmer command on the Tools menu has no effect. The IDE does not automatically launch the programmer when you attempt to download software to a board that does not match the expected hardware image.

Workaround: Launch the Quartus II software to access the Quartus II Programmer.

Windows: Frisk antivirus software causes SOPC Builder and Nios II Command Shell to be unresponsive

The SOPC Builder and Nios II Command Shell might become unresponsive if run while the Frisk antivirus software is running.

Workaround: Turn off the Dynamic Virus Checking feature of the Frisk software before running SOPC Builder or the Nios II Command Shell.

Device

This section lists any device-related issues.

Stratix® II EP2S60 ES devices cannot use MRAM byte enables

Early shipments of the Nios II Development Kit, Stratix II Edition include an EP2S60 engineering sample (ES) device. Stratix II EP2S60 ES devices have a silicon problem that prevents the use of byte enables on MRAM blocks.

Workaround: Refer to the Stratix II FPGA Family Errata Sheet for details.

Nios II IDE

This section lists any issues relating to the Nios II IDE.

Building Projects

Incorrect stack checking when interrupt vector custom instruction is used

If you enable run-time stack checking in systems that include the interrupt vector custom instruction, the Nios II IDE might erroneously report a stack overflow or might fail to report an actual stack overflow.

Workaround: Do not enable the run-time stack checking and interrupt vector custom instruction features in the same application.

Unable to save files containing copyright symbol

On computers using character encoding other than Cp1252, the copyright symbol included in the board_diag.c file is not recognized. Because of this, the file can not be saved using the Nios II IDE.

Workaround: Change the default text file encoding setting in the Nios II IDE to Cp1252 by performing the following steps:

- In the Nios II IDE, select Preferences from the Window menu
- In the General settings, select Workspace
- Select Default (Cp 1252) in the Text File Encoding section and click OK.

Illegal project location error when creating new project

You might receive an "Illegal project location" error message in the IDE if you use the default project name when creating a project in a new workspace. If the project name exists in another workspace, the IDE might not account for that in the new workspace.

Workaround: Change the project name to a name other then the default.

Compilation error "UNDEFINED VARIABLE %STACK_POINTER%" when building a project

This error occurs if the **Use a separate exception stack option** is turned on, and the exception stack is larger than the memory available for it.

Workaround: On the system library properties page for the project, turn off the separate exception stack or reduce the **Maximum exception stack size** setting.

When building a project, the Nios II IDE reports problems, but the build output in the Console doesn't contain any errors

Some linker warnings are incorrectly reported as errors in the Nios II IDE. The Dhrystone software example exhibits this behavior, and recompiling the project again makes the issue go away.

Workaround: If the Console output doesn't contain errors, then the project actually built fine. On subsequent builds, the linker step is skipped and the errors will not appear, because the project built without error previously.

Makefile reports incorrect number of bytes free for the stack and heap, if the heap is in a different memory region than the stack.

Workaround: Don't trust the heap and stack memory report from the makefile if you have placed the heap and stack in different memory regions.

Nios II IDE unnecessarily updates the SOPC Builder system file (.ptf)

The Nios II IDE opens the SOPC Builder system file (**.ptf**) by invoking SOPC Builder during certain operations, which might cause SOPC Builder to change the date stamp of the file even though the system was not modified. This might cause problems if you are using a version control system.

Workaround: If you are not using the Nios II C2H Compiler, you can change the PTF file properties to read-only to prevent the IDE from changing the file.

Incorrect address assignment for dual-port memory mastered by two different masters

If you have a dual-port memory in your Nios II system, and only the second slave port is mastered by the CPU, you might see an overlapping section error during the linking stage of building your software.

Workaround: In SOPC Builder, ensure that the first slave port of the dual-port memory is mastered by the Nios II CPU. The second port does not have to be mastered by the Nios II CPU.

Build errors after changing component names in SOPC Builder

If you rename components in the SOPC Builder system and then regenerate the SOPC Builder system, Nios II IDE system library projects based on that system will have build errors.

Workaround: After regenerating the SOPC Builder system, create a new system library project for the SOPC Builder system. Alternately, you can delete the system library project from the workspace without deleting the contents from the file system, and then re-import the project, selecting the appropriate SOPC Builder system.

Debugging Projects

MicroC/OS-II "Source not found" error occurs when debugging, after the Nios II IDE stops at a breakpoint.

You might receive a 'Source not found' error message if you step into MicroC/OS-II code or break on a program entry point while debugging your application.

Workaround: To eliminate this error, you must specify the proper source path mapping in the Nios II IDE by performing the following steps.

1. On the Windows menu, click **Preferences**. The Preferences dialog box appears.
2. Expand the **C/C++** group, then expand the **Debug** group, then click **Common Source Lookup Path**.
3. Click **Add...** .The Add Source dialog box appears.
4. Select **Path Mapping**, then click **OK**.
5. In the Preferences dialog box, select **Path Mapping: New Mapping** and click **Edit**. The Path Mappings dialog box appears.
6. Enter `cygdrive` in the **Name** field.
7. Click **Add...** . A dialog box appears, prompting you to specify the path mapping.
8. Enter `/cygdrive/c` in the **Compilation path** field and `c:/` in the **Local file system path** field.
9. Click **OK** in all open dialog boxes to return to the IDE workbench.

This process sets up the source path mapping for the entire workspace, which is recommended. You can also create a similar mapping for individual projects by clicking **Edit Source Lookup Path** that appears below the **Source not found** error message and following steps 4 through 8.

The Restart command on the Run menu does not work

Workaround: Stop the program, then debug it again. If the debugger is hung in an endless loop, use the following bash alias to break the target, then stop it:

```
alias break="kill -2 \`ps -a | grep nios2-elf-gdb | cut -f6 -d' '\`"
```

Programs that interact with a terminal Console on Windows don't work in the Nios II IDE v6.1 and later, but used to work in v6.0 or earlier

The Eclipse platform in v6.1 and later of the IDE (on Windows only) sends the string '\r\n' instead of just '\n' when running or debugging using the Terminal. This behavior can break existing software designs, and it is inconsistent with nios2-terminal, which still just sends '\n'.

Workaround: Change the software running on the Nios II processor to parse for \r as well as \n.

"Step failed. Target is not responding (timed out)" error message

The Nios II IDE debugger might hang and report this message if your code contains large arrays declared as local variables on the stack.

Workaround: Place the array and any other large buffers on the heap rather than on the stack.

Incorrect breakpoint filtering on threads

If you enable breakpoint filtering for a thread and later turn off filtering for the thread, the debugger might incorrectly continue to filter the thread.

Uninitialized Memory Error when executing from ISS

Under some cases the ISS does not ignore uninitialized memory reads, even when Uninitialized memory reads is set to Ignore on the ISS Settings tab of the run configuration. There is no known workaround.

Nios2-gdb-server fails to terminate after setting a watch point

You might be unable to terminate nios2-gdb-server after setting a watchpoint in the Nios II IDE debugger and resuming execution past the end of main. You will see an error "Terminate failed". You will not be able to start the debugger again; you will see a message reading "Another application is using the target processor..." in the Console view.

Workaround: Terminate the nios2-gdb-server.exe process manually using the Windows Task Manager.

Watchpoints do not work when set on variables whose size are not 32-bits

Workaround: Change the type of global and static local variables to int, long, or unsigned long before setting watchpoints on them.

Debugger cannot step into __sflags, and continues execution instead

The Nios II IDE debugger is unable to step into some low-level C library functions, such as `__sflags()` which is called from `_fopen_r()`. (`_fopen_r()` is called from `fopen()`.) If you try to step into such a function, execution will proceed as if you had indicated the debugger should resume execution.

Workaround: Step over such functions. Or, if execution continues after trying to step in, click **Suspend** on the Run menu.

Missing traced load/store instruction and data in the Trace view

If the trace options **Include load addresses**, **Include store addresses** or **Include data values** are enabled during debug, the load and store address and data will not appear at the first breakpoint after starting debugging. They will appear at successive breakpoints.

Workaround: To see load or store addresses and data in the instruction trace prior to main, turn on **Break at alt_main()** located on the **Debugger** tab for your debug configuration.

Cannot use watchpoints in the Nios II IDE when the FS2 console is open

Watchpoints do not work in the Nios II IDE when the **Use FS2 console window for trace and watchpoint support** setting is turned on in the **Debugger** tab of the Debug configuration. You will see an error message "The execution of program is suspended because of error." with details indicating that hardware watchpoints could not be inserted and deleted.

Workaround: If the FS2 console is open, you must use it to control watchpoints. For details, see the FS2 documentation.

Breakpoints on adjacent lines of assembly fail to halt the processor

Setting breakpoints on adjacent lines of assembly code might cause the Nios II processor to stop responding to the debugger.

Workaround: When debugging in mixed mode or disassembly view, separate breakpoints by at least one assembly instruction. This issue does not affect Nios II cores that do not have hardware breakpoints enabled in the JTAG debug module.

Navigating Projects

Internal Error when double-clicking on objdump file

On a Windows PC when opening a large objdump file in the Nios II IDE, you might get the following error message: "Unable to create this part due to an internal error. Reason for the failure: Editor could not be initialized.".

Workaround: Adjust the Windows launch arguments for the Nios II IDE editor. Perform the following steps:

10. On the Windows Start menu, browse to the **Nios II EDS 7.0** program icon, right click it, then click **Properties**. A Windows Properties dialog box appears.

11. In the **Target** field, add `vmargs -Xmx1024m` to the end of the path to the Nios II IDE executable. For example:

```
C:\altera\70\nios2eds\bin\eclipse\nios2-ide.exe -vmargs -Xmx1024m
```

Nios II IDE doesn't recognize when an elf file is built outside the IDE

If you use a build process external to the Nios II IDE, the IDE might not recognize when you have generated or re-generated the elf file for your project. In this case, the IDE debugger might report an error that the elf file doesn't exist.

Workaround: Right click on the project name and click **Refresh** to allow the IDE to recognize the elf file.

After switching workspaces, the IDE appears to freeze while displaying the Nios II IDE splash screen

After clicking **Switch Workspace** on the File menu on a Windows machine, a Nios II IDE splash screen appears. Unfortunately, this splash screen obscures a dialog box which asks you to specify the new workspace.

Workaround: Press Alt-Tab to switch applications. You will see two relevant application icons: an Eclipse icon associated with the splash screen and a Nios II IDE icon associated with the workspace dialog box. Select the Nios II icon to bring the dialog box to the foreground.

C/C++ Scanner does not understand certain C/C++ constructs

The C/C++ scanner is used for C/C++ Search, navigation, open declaration and parts of content assist. Due to limitations of the C/C++ Scanner, these features will not work with the following code constructs:

- Kernighan & Ritchie-style C
- Functions that take a function-pointer as an argument

Workaround: If the C/C++ Search fails, use the File Search facility.

C2H Compiler

This section lists any issues related to the Nios II C-to-Hardware Acceleration (C2H) Compiler.

C2H Compiler out-of-memory error

The C2H Compiler might generate an out-of-memory error at compile time for code that initializes a large local array.

Workaround: Initialize the memory with the Nios II processor and access it through pointers in the accelerator.

C2H Compiler doesn't accelerate subfunctions located in a separate file

When accelerating a function in a file, the C2H Compiler cannot link subfunctions that are defined in a different file.

Workaround: Include all subfunctions called by the accelerated function within the same source code file.

Incorrect results from logical or conditional operation with side-effects

The C2H Compiler always evaluates both operands of logical (&&, ||) and conditional (?:) operators. This is different from expected ANSI C behavior, for which operands are evaluated left-to-right, and unnecessary operands are skipped. For example, in the expression `(i-- && j--)`, if the value of `i` is zero, the right-hand-side (RHS) expression should not evaluate (i.e., `j` should not be decremented). However, this C2H Compiler erroneously evaluates both sides unconditionally, causing `j` to be decremented. The following example expressions could suffer from the same issue: `(i-- || j--)`, `(cond ? i-- : j--)`

Workaround: Use logical and conditional operations whose operators have no side effects. Side effects include pre-/post-fix increment operations (`++`, `--`), memory operations (`*`, `[]`, `.`, `->`), and function calls.

Closed system library while working with the C2H Compiler

The C2H Compiler requires the system library to obtain important details about the system, and cannot function if the system library is closed.

Workaround: Ensure that the system library project in the Nios II IDE is open prior to building an application project that contains a hardware accelerator.

Launch SOPC Builder button in C2H view

When the Nios II IDE workspace contains multiple projects with multiple system libraries, the incorrect SOPC Builder system might open when you click the Launch SOPC Builder in the C2H view.

Workaround: Keep only one system library project open at a time while using the C2H Compiler.

Build clean causes build failure

Performing a clean build on a Nios II IDE project that contains a hardware accelerator can cause the next build to fail in the IDE, because the clean build erroneously deletes a file required by the C2H Compiler.

Workaround: Do not perform a clean build on projects that use hardware accelerators. If you have already performed a clean build, recompile with option **Build software, generate SOPC Builder system, and run Quartus II compilation** to regenerate the necessary files.

Multiple clock domains causes hardware accelerator to fail

If a hardware accelerator and the components connected to its master ports are in different clock domains, the accelerator might behave incorrectly.

Workaround: Assign a single clock to a hardware accelerator and all the slave ports it connects to. It is acceptable for the system to contain multiple clock domains.

Comma operators are not supported

The C2H Compiler does not support comma operators, such as the following example:

```
for(i = 0, j = 3; i < 10; i++, j++)
{
  /* statements */
}
```

Workaround: You can manually duplicate the same functionality, such as:

```
j = 3;
for(i = 0; i < 10; i++)
{
  /* statements */
  j++;
}
```

Array elements of structs do not copy correctly

C2H hardware accelerators do not correctly copy array elements of structs. For example:

```
typedef struct my_struct {
  int a;
  int b;
  int buf[BUF_SIZE];
}MY_STRUCT;

MY_STRUCT struct_a = {1, 2, {3, 3, 3, 3}};
MY_STRUCT struct_b = {9, 8, {7, 7, 7, 7}};

struct_a = struct_b;
```

In this example, the `a` and `b` elements of the struct will copy correctly, but the `buf` element will not. After this assignment, struct_a will equal `{9, 8, {3, 3, 3, 3}}`.

Workaround: Copy the array elements explicitly, as follows:

```
{
  int i=0;
  do
    {
      struct_a.buf[i] = struct_b.buf[i];
      i++;
    } while (i<LENGTH_OF_BUF_ELEMENT)
}
```

Simulating on ISS is not supported

The Nios II instruction set simulator (ISS) does not support custom SOPC Builder components, and therefore cannot simulate systems that use hardware accelerators. You might get the following internal error if attempting to simulate using the ISS:

```
Internal Error (unhandled exception) in file cosim_main.cpp
```

Workaround: Run the program on a hardware system that includes the hardware accelerator.

Hardware accelerators remain after deleting the software project

If a system contains C2H hardware accelerators, deleting the software project that defines the accelerators does not remove the accelerators from the hardware system, and the accelerator logic remains in the SOPC Builder system.

Workaround: To remove an accelerator from a system, delete the accelerator from the C2H view in the Nios II IDE first, and then recompile the software project. The C2H Compiler then removes the accelerator from the

SOPC Builder system. Once the compilation is complete then the software application can be deleted from the workspace.

Changing build configurations produces unexpected results

The C2H Compiler does not support multiple build configurations (e.g. Release or Debug) in the Nios II IDE. After creating one or more accelerators in a particular configuration, the C2H Compiler will produce undefined results if you switch to a different build configurations and create more accelerators.

Workaround: For a specific SOPC Builder system and Nios II IDE project, specify C2H accelerators in only one build configuration. Note that you can still use multiple build configurations, as long as only one configuration specifies C2H Compiler settings.

Flash Programmer

This section lists any issues relating to the Nios II IDE flash programmer.

'No such file or directory' error

You might get this error when programming flash for a project stored in a path containing spaces. The flash programmer does not correctly handle spaces in the directory path. However, this error is benign, because flash programming will complete successfully.

Workaround: None required.

elf2flash elf size limit

The elf2flash utility supports **.elf** files up to approximately 24 MBytes in size. The elf2flash utility might fail with error "java.lang.OutOfMemoryError" on files larger than 24 MBytes.

Workaround: You can either lower the number of symbols in your elf file by turning off debug symbols, or specify less initialized data in the application.

Verifying flash with the nios2-flash-programmer command fails

Using the **nios2-flash-programmer** command line utility to verify flash contents using the `--verify` argument might result in a verify failure even though flash contents are correct. The failure message will be similar to the following:

```
Verifying 00000000 ( 0%)Failed to verify at around 00000000  Verify failed
```

Workaround: To work around this issue, avoid using the `--verify` argument with the **nios2-flash-programmer** command-line utility. A verification of flash contents can be done by reading the flash contents into a file using the `--read` argument, then comparing the file to the input file used to program flash.

Using the Nios II Flash Programmer with an override file stored in <Nios II EDS path>/bin might generate an error:

```
4 [main] nios2-flash-programmer 4440 _cygtls::handle_exceptions: Error while
dumping state (probably corrupted stack)  Segmentation fault (core dumped)
```

Workaround: To workaround this issue, move the override file from the <Nios II EDS path>/bin directory to the working directory of your Nios II IDE project. Then, specify the override file to **nios2-flash-programmer** using the `--override=` parameter. In Nios II IDE, you can do this in the **Additional nios2-flash-programmer arguments** field in the Flash Programmer dialog box. In command-line mode, the override file can be specified by adding the `--override=` argument to the command line, as follows:

```
nios2-flash-programmer --base-0x0 --override=my_fp_override.txt standard.flash
```

When used with a multiprocessor Nios II system, the Nios II Flash Programmer might produce an error:

```
There are two or more Nios II processors available which match the values
specified. Please use the configuration dialog to pick one, or specify the --
device and/or --instance parameters on the command line.
```

Workaround: In the Nios II IDE on the Flash Programmer dialog box, specify an appropriate `--instance=`<*correct instance value*> argument in **Additional nios2-flash-programmer arguments**. If using command line mode, add `--instance=`<*correct instance value*> to the command line. The correct instance value can be found in the project's system library project in the file **generated.sh**.

Download Cables & Debug Hardware

This section lists any issues related to download cables and other debug hardware.

Communication errors during run/debug sessions using older download cables

Debugging with the following Altera download cables might fail, due to electrical noise-related JTAG communication failures: USB-Blaster™ Rev A, ByteBlaster™, ByteBlasterMV™, ByteBlaster II, and MasterBlaster™ cables.

Currently, the only fully supported cable for downloading, debugging, or communicating with Nios II systems is the USB-Blaster Rev B cable or later. Revision B cables are clearly labeled as Revision B. (Revision A cables have no revision label.)

Workaround: Use a USB-Blaster Rev B cable. Older cables can be used, but they might encounter JTAG communication failures.

Development Boards

This section lists any issues related to Altera development boards.

Intermittent failures while accessing CompactFlash card

The Nios II Development Kit version 5.0 and higher includes a CompactFlash controller peripheral suitable for interfacing to CompactFlash cards in True IDE mode on Nios development boards. In order for True IDE mode to operate, CompactFlash cards require that the ATASEL_N input be driven to ground during power-up.

The CompactFlash controller peripheral includes a configurable power register used to power-cycle CompactFlash cards in Nios II software through a MOSFET on the Nios development boards. However, in certain development boards, power to the CompactFlash card will not turn off completely during this power cycle operation. Because of this, the CompactFlash might not sample the ATASEL_N pin during the power-cycle operation after FPGA configuration when this pin is driven to ground. Instead, the CompactFlash card might sample the ATASEL_N pin when power is first applied to the development board, when I/O are not yet driven by the FPGA (before FPGA configuration).

Workaround: If you encounter errors with CompactFlash when using the Nios development boards, try one of the following:

- Use a different CompactFlash card. Certain cards are more susceptible to the power-cycling issue than others.
- Modify the Nios development board. This is recommended for users who are familiar and comfortable with board-level modifications. Disconnect pin 9 (ATASEL_N) on the CompactFlash socket on your Nios development board and tie this pin to ground. Note that the CompactFlash socket uses a staggered numbering on the pins (starting from pin 1: 1, 26, 2, 27, ...); refer to the CompactFlash Association specification for right-angle surface-mount connectors for exact specifications on this connector. This modification will permanently enable True-IDE mode operation.

Toolchain (gcc, gdb, etc.)

This section lists any issues related to the Nios II compiler toolchain, such as gcc and gdb.

Breakpoints in C++ constructors fail to halt the processor

Breakpoints set in a C++ constructor might not halt the processor due to a widespread GNU GCC, GDB issue. This is not a Nios II IDE-specific issue.

Workaround: You can work around this issue by moving all of your constructor source code into another class method, called init. Then invoke this method from within the constructor.

Target Software

This section lists any issues related to software or drivers that target the Nios II processor.

Lightweight IP (lwIP) failure when using DHCP

The Lightweight IP stack might experience intermittent failures when using DHCP to acquire an IP address. When the failure occurs, the following is printed to stdout:

```
Assertion "dhcp_create_request: dhcp->p_out == NULL" failed at line 1283 in
/cygdrive/c/altera/61b169/nios2eds/components/altera_lwip/UCOSII/src/downloads/lwi
p-1.1.0/src/core/dhcp.c
```

Workaround: If possible, Altera recommends switching to the NicheStack TCP/IP Stack - Nios II Edition introduced in Nios II 6.1. If not, the following workarounds can be employed.

- Power-cycle the target board and re-attempt DHCP negotiation, which usually results in correct acquisition of an address from DHCP, assuming that the DHCP server is able to allocate IP addresses.
- Use a static IP address and disable DHCP.

stdin, stdout, and stderr don't work in MicroC/OS-II applications when using small C library option

MicroC/OS-II applications using `stdin`, `stdout`, and `stderr` fail to operate when using the small C library.

Workaround: Disable the small C library option.

malloc(), realloc() failures with MicroC/OS-II

When using the MicroC/OS-II RTOS, calls to `malloc()` and `realloc()` might fail if successive calls to `malloc()` or `realloc()` within a MicroC/OS-II task occur after changing the task priority of the task in which a memory block was originally allocated.

Workarounds:

- Allocate and/or reallocate memory blocks outside of MicroC/OS-II tasks, before task switching starts. Changing thread priorities at runtime is now possible.
- Allocate fixed areas of memory using arrays (rather than using `malloc()`) before task switching starts. Changing thread priorities at runtime is now possible.
- Allocate memory using `malloc()` or `realloc()` from a MicroC/OS-II task. You can change task priorities at runtime, but only for tasks that have not used `malloc()` or `realloc()`.

cout from MicroC-OS/II task will not send data to STDOUT

If neither `printf()` or `cout` is used from `main()` before tasks are started, `cout` will not work from a task.

Workaround: Add the following C++ code to the beginning of `main()`:

```
std::ios_base::sync_with_stdio(false);
```

Problems using HAL drivers with Toshiba Flash

The HAL CFI Flash driver might not work for Toshiba flash memory that claims to be CFI compliant.

Workaround: In the **altera_avalon_cfi_flash_table.c** file, change the `#define READ_ARRAY_MODE` from `(alt_u8)0xFF` to `(alt_u8)0xF0` and rebuild the project.

Creating new custom HAL components

When you first create a component's inc directory or HAL header file, you might first need to perform a clean build (i.e., rebuild) of existing system library projects for the new files to be detected.

Legacy SDK

Support for the Legacy SDK mode was removed in version 6.0 of the Nios II Embedded Design Suite.

SOPC Builder and Quartus II Software

This section lists any issues related to the Quartus II software or SOPC Builder that specifically affect Nios II designers.

For further information on the Quartus II software, refer to the latest Quartus II release notes on the Altera web site:

http://www.altera.com/literature/lit-qts.jsp

Example Designs

This section lists any issues related to the example designs included with the Nios II Embedded Design Suite.

Hardware Designs

Incorrect SSRAM PLL phase shift in Stratix II 2s60 RoHS and Cyclone™ II 2C35 standard and full_featured designs

The PLL phase shift used to implement the clock to SSRAM in the standard and full_featured designs targeting the Nios II Cyclone II 2C35 and Nios II Stratix II 2S60 RoHS development boards is incorrect, and may cause timing violations when reading from or writing to SSRAM if you increase the system clock speed above that of the example designs (85MHz), while using two clock-cycles of read latency in the SSRAM interface.

While most Altera example designs targeting these boards run at 85Mhz, the "standard" design targeting the Stratix II 2S60 RoHS board runs at 100MHz. Although no SSRAM failures have been observed with this board/example design in its current configuration, the design should be re-generated for reliable SSRAM operation.

Note that no action is required if you wish to continue using the example designs at their initial clock speed of 85MHz. However, if you wish to create your own custom design that uses SSRAM and runs above 85Mhz, or if you wish to increase clock speed of an Altera example design above 85MHz, Altera recommends that the PLL phase shift for generating the SSRAM clock on Nios II Cyclone II 2C35 and Nios II Stratix II 2S60 RoHS edition boards be set to -3.38ns, in "normal" PLL mode. Altera's example designs targeting these boards currently have a phase shift of -4.8ns, with the PLL in "normal" mode.

To change the PLL setting, perform the following steps:

12. Open the desired example design in SOPC Builder
13. Double click the "pll" instance existing in the design and launch the PLL MegaWizard
14. In the PLL MegaWizard, click the "Output Clocks" tab and then clock "c2", used to generate the SSRAM clock.

15. Change the PLL phase shift from -4.8ns to -3.38ns

16. Click Finish to save changes and exit the PLL MegaWizard

17. Click Finish to save settings to the PLL instance in SOPC Builder

18. Re-generate the system in SOPC Builder and re-compile in Quartus II.

For further reference, please refer to the SSRAM Interface readme document, located in the <Quartus II installation directory>/sopc_builder/documents folder. This document discusses the SSRAM timing analysis methodology in detail. Additionally, *AN 411: Understanding PLL Timing for Stratix II Devices* discusses clock phase shift calculations and assignments.

Software Designs

RAM test failure when running Memory Test software template on the ISS

An issue in the instruction set simulator (ISS) model of the JTAG UART can cause a console communication error during the RAM test when running the Memory Test software template on the ISS.

Networking Examples

If you are running a networking example design and you are asked for a 9-digit number after the letters 'ASJ', and your Nios II development board does not have a sticker with a 9-digit number after the letters 'ASJ', please enter a unique 9-digit number when prompted. Ensure that this number is unique to each Nios board connected to your network to avoid network address conflicts.

Hardware Simulation

This section lists issues related to simulating Nios II processor systems on an RTL simulator, such as the ModelSim® simulator.

Error: "UNC paths are not supported. Defaulting to Windows directory."

If you launch ModelSim from a working directory that is mapped via a UNC path (a path that starts with `//` instead of drive letter), you will get this error message in SOPC Builder. This error occurs because ModelSim is calling a cmd shell, which does not support UNC paths.

Workaround: Map the UNC path to a drive letter and use the drive letter to reference the working directory in the launching shell.

Simulation failure if reset address is set to EPCS

Running ModelSim RTL simulation of a Nios II system fails if the reset address of the Nios II processor is set to an EPCS Serial Flash Controller.

Workaround: To simulate your system, temporarily set the Reset Address of the Nios II CPU to the memory that your application code will reside (for example, SDRAM), then re-generate the system in SOPC Builder and run RTL simulation again. Before booting the Nios II CPU from EPCS flash on your target board, change the Nios II

Reset Address back to the EPCS Controller peripheral and re-generate the system in SOPC Builder and re-compile in the Quartus II software to produce an updated FPGA configuration file with the Nios II CPU booting from EPCS flash.

Uninitialized BSS variables in simulation

If your program reads the value of an uninitialized BSS variable during HDL simulation when the HAL system library has been compiled with the **ModelSim only, no hardware support** property enabled in Nios II IDE, a warning will be produced about unfiltered data being ‘x’. This occurs because when this property is enabled, the code that clears the BSS memory region is omitted to speed up HDL simulation so this memory region is uninitialized. The BSS region contains global and static local variables that are not initialized by the application so they default to a value of zero. When the Nios II CPU reads uninitialized variables, it displays a warning and converts any of the bits of the uninitialized data to zero which correctly mimics the effect of the missing BSS clearing code. The HAL code that executes before and after main() may use BSS variables so these warnings might be generated even if your application doesn’t use the BSS.

ModelSim fails to load large memory models

The ModelSim tool might fail to load simulation models for memory arrays larger than 128M bytes, halfwords or words in size. If the sum of the following parameters is greater than 27, the ModelSim tool will fail to load:

- address_bits (i.e. 14)
- column bits (i.e. 11)
- log2(number of banks) (number of banks is usually 4, so this term is usually 2)
- log2(chipselects) (number of chipselects is usually 1, so this term is usually 0)

Workaround: Simulate using a smaller SDRAM than the SDRAM implemented in hardware. This is possible if the entire memory space doesn’t need to be simulated.

Documentation Issues

This section lists errors, unclear descriptions, or omissions from current published specifications or product documents.

GNU assembler does not accept the --defsym flag

According the GNU documentation, you can set an assembler definition by using the `--defsym` flag, but it does not work in the following form: `--defsym MY_VAR=1`

Nios II Processor Reference Handbook refers erroneously to the data bus

In the *Implementing the Nios II Processor in SOPC Builder* chapter of the *Nios II Processor Reference Handbook*, under *Caches & Tightly Coupled Memories Tab*, the following text appears under *Instruction Settings*: "Usually you enable bursts on the processor's data bus when processor data is stored in DRAM, and disable bursts when processor data is stored in SRAM." This sentence should read "Usually you enable bursts on the processor's instruction master when instructions are stored in DRAM, and disable bursts when instructions are stored in SRAM."

Contact Information

For more information, contact Altera's mySupport website at **www.altera.com/mysupport**. Click **Create New Service Request**, and choose the **Product Related Request** form.

Revision History

Table 1 shows the revision history for the Nios II Embedded Design Suite v7.0 Errata Sheet.

Table 1. Nios II Embedded Design Suite v7.0 Errata Sheet Revision History

Version	Date	Errata Summary
1.0	March 2007	First release
1.1	March 2007	■ Interrupt vector custom instruction in multi-processor systems ■ Incorrect stack checking when interrupt vector custom instruction is used ■ Nios II/e core burst support ■ Duplicate errata combined: ■ MicroC/OS-II source code not found while debugging ■ "Source not found" error occurs when debugging, after the Nios II IDE stops at a breakpoint.

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com
Applications Hotline:
(800) 800-EPLD
Literature Services:
literature@altera.com

Nios II Embedded Design Suite 6.1 Release Notes

December 2006, Version 6.1 Release Notes

This document lists the release notes for the Nios® II Embedded Design Suite (EDS) version 6.1.

Table of Contents:

New Features & Enhancements

The Nios II EDS version 6.1 addresses issues found in previous releases, and adds the following primary features:

- NicheStack TCP/IP Stack - Nios II Edition
- Interrupt vector custom instruction
- Nios II integrated development environment (IDE) based on the latest version of the Eclipse platform
- Memory footprint reduction options

The sections below provide a detailed list of all product updates, including known issues fixed in this release.

Device & Host Support

This release supports the following Altera® FPGA families:

- Stratix® III
- Stratix II and Stratix II GX
- Stratix and Stratix GX
- Cyclone™ II
- Cyclone

The host system requirements for the Nios II EDS are based on the requirements for the Altera Complete Design Suite. Refer to *Quartus II Installation & Licensing for PCs* or *Quartus II Installation & Licensing for UNIX and Linux Workstations* manual.

Installation and Licensing Instructions

For installation instructions, refer to *Quartus II Installation & Licensing for PCs* or *Quartus II Installation & Licensing for UNIX & Linux Workstations* on the Altera website at www.altera.com/literature/lit-qts.jsp.

Using Previously Installed Versions of the Nios II EDS

SOPC Builder and the Nios II IDE refer to the most recently installed version of components (such as the Nios II processor and peripherals) and their software drivers. To revert to a prior version of the Nios II development tools, you can reinstall the previous version of tools or modify the following environment variables.

- `SOPC_BUILDER_PATH` - Ensure that `SOPC_BUILDER_PATH` points to the installation directory of the desired Nios II version and no other Nios II versions.
- `SOPC_BUILDER_PATH_`<*version*> - Ensure that `SOPC_BUILDER_PATH_`<*version*> points to the installation directory of the corresponding Nios II version.
- `SOPC_KIT_NIOS2` - Ensure `SOPC_KIT_NIOS2` points to the installation directory of the desired Nios II version and no other Nios II versions.

If you have multiple versions of the Quartus II software installed, launch the supported version of Quartus II to ensure that the `QUARTUS_ROOTDIR` environment variable is updated.

Nios II Processor Cores

This section describes changes to the Nios II processor cores.

Interrupt vector custom instruction

This new custom instruction introduced in Nios II version 6.1 provides lower latency when dispatching interrupts. When enabled, it is automatically supported by the hardware abstraction layer (HAL).

Floating point custom instruction

The floating point custom instruction is enhanced in version 6.1. The changes include:

- Improved divide performance (Fmax)
    - Up to 2x f_{MAX} increase depending on design and target device
    - Increased memory utilization is a result of this improvement. Designs from version 6.0 will use more embedded memory blocks when compiled in version 6.1. The divider uses approximately 160,000 memory bits and 4,000 logic elements in Cyclone, Cyclone II, and Stratix devices and approximately 2,000 ALUTS in Stratix II devices.
- Increased latency
    - The latency in clock cycles for various operations is as follows. All operations require one initial clock cycle to register the inputs.
        - Divide: 32 (26 in version 6.0)
        - Multiply: 6 (unchanged from version 6.0)
        - Add / Sub: 8 (unchanged from version 6.0)
- No denormal outputs
    - All denormal inputs and output are treated as zero in version 6.1 floating point custom instruction.

SOPC Builder

This section describes changes to SOPC Builder which affect Nios II designers. For complete revision history of SOPC Builder and the Quartus II software, refer to the release notes for the Quartus II software version 6.1. The *Quartus II Handbook, Volume 4: SOPC Builder* contains complete documentation for SOPC Builder.

There are no major changes to SOPC Builder that affect the Nios II processor this release.

Nios II IDE

This section describes changes to the Nios II integrated development environment (IDE).

Updated to latest version of Eclipse and CDT

The Nios II IDE was updated to use Eclipse version 3.2 and CDT version 3.1. This update provides faster startup time, improved debugger responsiveness, and faster memory view scrolling.

New location for projects

By default, in version 6.1 the Nios II IDE starts in a clean workspace and does not prompt for a workspace at startup. To convert your existing projects to 6.1 projects, you can either individually import your projects into the new workspace, or point the IDE at your previous workspace to convert all your projects at one time. Refer to the Upgrading Projects for the Nios II IDE v6.1 cheat sheet available in the IDE. You can also enable the **Prompt for workspace on startup** option. To find this option, click **Preferences** on the Window menu, then expand the **General** group and click the **Startup and Shutdown** page.

Nios II C-to-Hardware Acceleration (C2H) Compiler

The Nios II C-to-Hardware Acceleration (C2H) Compiler was introduced in the Nios II EDS version 6.0. The C2H Compiler is a tool that allows you to create custom hardware accelerators directly from ANSI C source code, which can often improve the execution performance by an order of magnitude. Documentation on the C2H Compiler is available on the Nios II literature page, http://www.altera.com/literature/lit-nio2.jsp.

This section describes changes to the C2H Compiler.

Pipelined function calls

A function being accelerated by the C2H Compiler can contain sub-function calls. For each sub-function called within a C2H-accelerated function, the C2H Compiler generates an additional hardware accelerator which is then invoked by the original C2H accelerator generated for the top-level function. In version 6.1, a sub-function call from within a C2H-accelerated function can be pipelined with the calling function, based on normal data dependencies, and therefore may not stall the calling function's pipeline.

Multiplier sharing

The C2H Compiler will try to conserve FPGA resources by sharing hardware multipliers, barrel shifters, and dividers whenever possible. For example, when a loop or function in the accelerated C code contains more than one multiply operation, those multiply operations may be scheduled sequentially, using the same multiplier resource. This serialization of multiply operations occurs when, and only when, the extra multiply operations can be scheduled during cycles when the shared multiplier resource is otherwise idle. Stated another way, separate multiply operations will be scheduled sequentially using the same multiplier resource until it increases the CPLI of the loop or function. At that point, additional multiplier resources will be instantiated to perform the additional multiplies in parallel. Divider and barrel shifter sharing works in the same way.

Flash Programmer

This section describes changes to the flash programmer in the Nios II IDE.

Flash programmer user interface updates

The flash programmer in the Nios II IDE was updated to allow you to customize the location to program FPGA configuration data and other data files into flash memory.

Target Software

This section describes changes to Altera-provided target software which runs on the Nios II processor, such as the hardware abstraction layer (HAL) system library.

NicheStack TCP/IP Stack - Nios II Edition

The NicheStack TCP/IP Stack - Nios II Edition is introduced in the Nios II EDS version 6.1. This TCP/IP stack supersedes the Lightweight IP (lwIP) stack that is also shipped with the Nios II EDS. The lwIP stack will be deprecated, and is not recommended for new designs.

Memory footprint reduction options

Several options are included in the Nios II IDE version 6.1 to allow you to reduce your memory footprint. These options include reducing stdio services, removing C++ support, and indicating that `main()` does not exist. Details on these options are available in the Nios II IDE on-line help.

MicroC/OS-II updated to version 2.83

The update to version 2.83 of MicroC/OS-II provides new configuration options in the IDE.

Avalon slave ports not mastered by CPU are not in system.h

In pre-6.1 versions of the Nios II EDS, all slave ports on a module that was mastered by the CPU appeared in the **system.h** file. In version 6.1, **system.h** will only contain definitions for slave ports that are mastered by the CPU in the system. For example, if a Nios II processor masters one port, S1, of a dual-port memory, only the S1 port will be defined in **system.h**; the S2 port will not be defined.

Example Designs

Hardware Example Designs

Stratix III Fast Design

A Nios II fast design targeting a Stratix III device is available in version 6.1, in the <*Nios II EDS install path*>**/examples/** directory.

Stratix II EP2S60 RoHS Fast Design

The start of on-chip memory was moved from 0x10000 to 0x0, and all the other addresses were shifted accordingly. Also, the on-chip memory latency was changed from 1 cycle to 2 cycles.

Cyclone II EP2C35 Designs

All the EP2C35 designs were updated to set all SSRAM pins to have drive strengths of 12mA (originally 24mA). This was done to reduce the noise which affected stable PLL operation.

Cyclone EP1C12 Evaluation Kit Designs

The hardware designs for the Cyclone EP1C12 evaluation board have been removed from the Nios II EDS.

Software Example Designs

Simple Socket Server

The Simple Socket Server software example now uses the NicheStack TCP/IP Stack in place of the lwIP stack.

Web Server

The Web Server software example now uses the NicheStack TCP/IP Stack in place of the lwIP stack. Web page control of various components on the board is also available.

Hello World Small

The Hello World Small example was updated to use the memory footprint reduction options. The current memory footprint is approximately 330 bytes.

Errata Fixed in this Release

This section lists errata items from previous releases which are fixed in version 6.1.

Nios II IDE does not prompt for workspace

The Nios II IDE might not prompt for the workspace at startup.

Resource(s) out of sync with the file system when searching for files in the workspace

When searching through files in the IDE workspace, you might get an error message saying that one or more resources are out of sync with the file system.

Nios II IDE stops responding after you double-click a .o file.

Double-clicking a **.o** file in the Nios II IDE on Windows might cause the IDE to stop responding. This issue occurs only on systems with a separate installation of Cygwin in addition to Cygwin installed with the Nios II development tools.

Cannot connect Nios II tightly-coupled instruction and data masters to the same dual-port memory

SOPC Builder does not generate an error if you connect Nios II tightly-coupled instruction and data masters to both ports on a dual-port on-chip memory. However, this configuration is not supported in hardware.

Nios II IDE help system fails to display content

To display the help content, your computer must able to recognize itself on the network. Incorrect proxy settings can cause the help content to fail to display.

Linux: Out of memory error in IDE

On Linux, you might receive an out of memory error in the Nios II IDE when it is launched from SOPC Builder

Linux: F1 Help in the Nios II IDE does not function on Linux

Selecting F1 in the IDE on Linux does not display help contents.

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com
Applications Hotline:
(800) 800-EPLD
Literature Services:
literature@altera.com

Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

Nios II Embedded Design Suite 6.1 Errata Sheet

December 2006 Errata Sheet

This document addresses known errata and documentation issues for the Nios® II Embedded Design Suite (EDS) version 6.1. Errata are functional defects or errors, which might cause the product to deviate from published specifications. Documentation issues include errors, unclear descriptions, or omissions from current published specifications or product documents. Errata items discovered after the release of Nios II EDS version 6.1 are marked with the date the items were added to this document.

Table of Contents:

For the most up-to-date errata for this release, refer to the errata sheet on the Altera website: **http://www.altera.com/literature/es/es_nios2eds_61.pdf**

Nios II Processor Core

This section lists any issues related to the Nios II processor cores.

"nios_cpu: Unknown Break Location nios_cpu/jtag_debug_module" error message

If you modify the Nios II core in an existing system by disabling the JTAG debug module in the Nios II configuration wizard, you might receive the error message "**nios_cpu: Unknown Break Location nios_cpu/jtag_debug_module**" during SOPC Builder generation.

Workaround: Delete the Nios II core from the system and re-add it, ensuring that the JTAG debug core is disabled in the wizard.

VHDL sensitivity list warnings during Quartus II synthesis

You may receive warnings similar to the following in the Quartus II software when compiling a VHDL Nios II system containing the JTAG debug module.

```
Warning (10492): VHDL Process Statement warning at
cpuname_jtag_debug_module.vhd(254): signal "usr1" is read inside the Process
Statement but isn't in the Process Statement's sensitivity list
Warning (10492): VHDL Process Statement warning at
cpuname_jtag_debug_module.vhd(254): signal "ena" is read inside the Process
Statement but isn't in the Process Statement's sensitivity list
```

These warnings are benign and can be ignored. The warnings are a result of the `ena` and `usr1` signals not being included in the debug module's sensitivity list.

Double-precision Floating-Point operations with floating-point custom instructions

Calls to double-precision floating-point functions in math.h will return less-precise results on Nios II processors using the floating-point custom instruction. Floating-point constants are forced to single-precision when the floating-point custom instructions is present, which affects the constants for the double-precision floating-point functions in libm.

Peripherals

This section lists any issues related to the Altera embedded peripherals included in the Quartus II software.

JTAG UART is unstable after device-wide reset

If the DEV_CLRn pin on the FPGA input has been assigned (in Quartus® II software) to generate a device-wide reset, and the FPGA is reset while the JTAG UART is active, then the JTAG UART might become unstable.

Workaround: Do not use the DEV_CLRn function on the FPGA. Turn off the Enable device wide reset (DEV_CLRn) setting in Quartus II software.

Host Platform

This section lists any issues related specifically to the host platform.

Windows XP: Internal error when repeatedly stepping into with debugger [December 2006]

If you repeatedly use the Step Into command while debugging on Windows XP, you might receive the internal error, "Retrieving Children: An internal error occurred during: Retrieving Children".

Workaround: End your debug session, then re-open it and resume debugging. Reduce the frequency of using the step into command.

Linux: Debugging with the Nios II ISS target can cause a process leak on Linux

If you try to interrupt or terminate a debug session targeting the Nios II instruction set simulator (ISS), you might see an error message "Interrupt Failed or Terminate Failed". This means that the nios2-iss process failed to terminate. The debug session appears to have terminated in the IDE, but the nios2-iss process still remains alive.

Workaround: Open a command shell and kill the nios2-iss process.

Linux: The Quartus II stand-alone programmer is not supported on Linux

There is no Quartus II stand-alone programmer for Linux. As a result, in the Nios II IDE the Quartus II Programmer command on the Tools menu has no effect. The IDE does not automatically launch the programmer when you attempt to download software to a board that does not match the expected hardware image.

Workaround: Launch the Quartus II software to access the Quartus II Programmer.

Windows: Frisk antivirus software causes SOPC Builder and Nios II Command Shell to be unresponsive

The SOPC Builder and Nios II Command Shell might become unresponsive if run while the Frisk antivirus software is running.

Workaround: Turn off the Dynamic Virus Checking feature of the Frisk software before running SOPC Builder or the Nios II Command Shell.

Command-line utilities don't display any output

Command-line utilities for creating and manipulating Nios II IDE projects on Windows might not display any output, even though the command completed successfully. The following commands function correctly but do not produce output to stdout:

- nios2-build-project
- nios2-create-application-poject
- nios2-create-system-library

- nios2-delete-project
- nios2-import-project

Workaround : Pipe the command output through **tee**. For example:

```
nios2-build-project --help | tee foo.txt
```

Device

This section lists any device-related issues.

Stratix II EP2S60 ES devices cannot use MRAM byte enables

Early shipments of the Nios II Development Kit, Stratix II Edition include an EP2S60 engineering sample (ES) device. Stratix II EP2S60 ES devices have a silicon problem that prevents the use of byte enables on MRAM blocks.

Workaround: Refer to the Stratix II FPGA Family Errata Sheet for details.

Nios II IDE

This section lists any issues relating to the Nios II IDE.

Building Projects

Illegal project location error when creating new project

You might receive an "Illegal project location" error message in the IDE if you use the default project name when creating a project in a new workspace. If the project name exists in another workspace, the IDE might not account for that in the new workspace.

Workaround: Change the project name to a name other then the default.

Compilation error "UNDEFINED VARIABLE %STACK_POINTER%" when building a project

This error occurs if the **Use a separate exception stack option** is turned on, and the exception stack is larger than the memory available for it.

Workaround: On the system library properties page for the project, turn off the separate exception stack or reduce the **Maximum exception stack size** setting.

When building a project, the Nios II IDE reports problems, but the build output in the Console doesn't contain any errors

Some linker warnings are incorrectly reported as errors in the Nios II IDE. The Dhrystone software example exhibits this behavior, and recompiling the project again makes the issue go away.

Workaround: If the Console output doesn't contain errors, then the project actually built fine. On subsequent builds, the linker step is skipped and the errors will not appear, because the project built without error previously.

Makefile reports incorrect number of bytes free for the stack and heap, if the heap is in a different memory region than the stack.

Workaround: Don't trust the heap and stack memory report from the makefile if you have placed the heap and stack in different memory regions.

Nios II IDE unnecessarily updates the SOPC Builder system file (.ptf)

The Nios II IDE opens the SOPC Builder system file (**.ptf**) by invoking SOPC Builder during certain operations, which might cause SOPC Builder to change the date stamp of the file even though the system was not modified. This might cause problems if you are using a version control system.

Workaround: If you are not using the Nios II C2H Compiler, you can change the PTF file properties to read-only to prevent the IDE from changing the file.

Incorrect address assignment for dual-port memory mastered by two different masters

If you have a dual-port memory in your Nios II system, and only the second slave port is mastered by the CPU, you might see an overlapping section error during the linking stage of building your software.

Workaround: In SOPC Builder, ensure that the first slave port of the dual-port memory is mastered by the Nios II CPU. The second port does not have to be mastered by the Nios II CPU.

Build errors after changing component names in SOPC Builder

If you rename components in the SOPC Builder system and then regenerate the SOPC Builder system, Nios II IDE system library projects based on that system will have build errors.

Workaround: After regenerating the SOPC Builder system, create a new system library project for the SOPC Builder system. Alternately, you can delete the system library project from the workspace without deleting the contents from the file system, and then re-import the project, selecting the appropriate SOPC Builder system.

Debugging Projects

MicroC/OS-II source code not found while debugging [December 2006]

You may receive a 'Source not found' error message if you step into MicroC/OS-II code while debugging your application.

Workaround: To eliminate this error, you must specify the proper source path mapping in the Nios II IDE by performing the following steps.

1. On the Windows menu, click **Preferences**. The Preferences dialog box appears.
2. Expand the **C/C++** group, then expand the **Debug** group, then click **Common Source Lookup Path**.
3. Click **Add...** .The Add Source dialog box appears.

4. Select **Path Mapping**, then click **OK**.

5. In the Preferences dialog box, select **Path Mapping: New Mapping** and click **Edit**. The Path Mappings dialog box appears.

6. Enter `cygdrive` in the **Name** field.

7. Click **Add...** . A dialog box appears, prompting you to specify the path mapping.

8. Enter `/cygdrive/c` in the **Compilation path** field and `c:/` in the **Local file system path** field.

9. Click **OK** in all open dialog boxes to return to the IDE workbench.

This process sets up the source path mapping for the entire workspace, which is recommended. You can also create a similar mapping for individual projects by clicking **Edit Source Lookup Path** that appears below the **Source not found** error message and following steps 4 through 8 above.

Communication issue with multi-CPU or multi-JTAG UART designs [December 2006]

Designs that contain multiple CPUs or multiple JTAG UARTs might not be able to communicate with the target when running, debugging, or programming flash memory. Designs which have only a single CPU and JTAG UART are not affected by this issue.

Workaround: You need to explicitly specify the JTAG instance number for the target. Perform the following steps:

1. Open the Run/Debug launch configuration dialog box and click the Main tab.

2. In the **Target Hardware** section, enter `--instance=`*N* in the **Additional nios2-download arguments** field to supply the JTAG instance information to the debugger.

For designs with a single CPU and JTAG UART, the default instance number N is zero and does not need to be specified as an extra argument.

The Restart command on the Run menu does not work [December 2006]

Workaround. Stop the program, then debug it again. If the debugger is hung in an endless loop, use the following bash alias to break the target, then stop it:

```
alias break="kill -2 \`ps -a | grep nios2-elf-gdb | cut -f6 -d' '\`"
```

'Additional nios2-terminal arguments' field is not functional [December 2006]

On the **Target Connection** tab of the Run or Debug launch configuration dialog boxes, the **Additional nios2-terminal arguments** field is not functional. Entering commands in these fields will have no effect.

The **Additional nios2-download arguments** and **Additional nios2-flash-programmer arguments** fields are functional.

Workaround: There is no workaround at this time.

"Source not found" error occurs when debugging, after the Nios II IDE stops at a breakpoint.

This issue might occur on Windows computers when the breakpoint is within a library function not contained in the Nios II C/C++ Application project.

Workaround: Open the Preferences dialog box, and navigate to the **C/C++**, **Debug**, **Common Source Lookup Path** page. In the **Common Source Lookup Path** list, add a mapping from "\cygdrive\c" to "c:\".

Programs that interact with a terminal Console on Windows don't work in the Nios II IDE v6.1, but used to work in v6.0 or earlier

The Eclipse platform in v6.1 of the IDE (on Windows only) sends the string '\r\n' instead of just '\n' when running or debugging using the Terminal. This behavior can break existing software designs, and it is inconsistent with nios2-terminal, which still just sends '\n'.

Workaround: Change the software running on the Nios II processor to parse for \r as well as \n.

"Step failed. Target is not responding (timed out)" error message

The Nios II IDE debugger might hang and report the above message if your code contains large arrays declared as local variables on the stack.

Workaround: Place the array and any other large buffers on the heap rather than on the stack.

Incorrect breakpoint filtering on threads

If you enable breakpoint filtering for a thread and later turn off filtering for the thread, the debugger might incorrectly continue to filter the thread.

Uninitialized Memory Error when executing from ISS

Under some cases the ISS does not ignore uninitialized memory reads, even when Unitialized memory reads is set to Ignore on the ISS Settings tab of the run configuration. There is no known workaround.

Nios2-gdb-server fails to terminate after setting a watch point

You might be unable to terminate nios2-gdb-server after setting a watchpoint in the Nios II IDE debugger and resuming execution past the end of main. You will see an error "Terminate failed". You will not be able to start the debugger again; you will see a message reading "Another application is using the target processor..." in the Console view.

Workaround: Terminate the nios2-gdb-server.exe process manually using the Windows Task Manager.

Watchpoints do not work when set on variables whose size are not 32-bits

Workaround: Change the type of global and static local variables to int, long, or unsigned long before setting watchpoints on them.

Debugger cannot step into __sflags, and continues execution instead

The Nios II IDE debugger is unable to step into some low-level C library functions, such as `__sflags()` which is called from `_fopen_r()`. (`_fopen_r()` is called from `fopen()`.) If you try to step into such a function, execution will proceed as if you had indicated the debugger should resume execution.

Workaround: Step over such functions. Or, if execution continues after trying to step in, click **Suspend** on the Run menu.

Missing traced load/store instruction and data in the Trace view

If the trace options **Include load addresses**, **Include store addresses** or **Include data values** are enabled during debug, the load and store address and data will not appear at the first breakpoint after starting debugging. They will appear at successive breakpoints.

Workaround: To see load or store addresses and data in the instruction trace prior to main, turn on **Break at alt_main()** located on the **Debugger** tab for your debug configuration.

Cannot use watchpoints in the Nios II IDE when the FS2 console is open

Watchpoints do not work in the Nios II IDE when the **Use FS2 console window for trace and watchpoint support** setting is turned on in the **Debugger** tab of the Debug configuration. You will see an error message "The execution of program is suspended because of error." with details indicating that hardware watchpoints could not be inserted and deleted.

Workaround: If the FS2 console is open, you must use it to control watchpoints. For details, see the FS2 documentation.

Breakpoints on adjacent lines of assembly fail to halt the processor

Setting breakpoints on adjacent lines of assembly code might cause the Nios II processor to stop responding to the debugger.

Workaround: When debugging in mixed mode or disassembly view, separate breakpoints by at least one assembly instruction. This issue does not affect Nios II cores that do not have hardware breakpoints enabled in the JTAG debug module.

Navigating Projects

Internal Error when double-clicking on objdump file [December 2006]

On a Windows PC when opening a large objdump file in the Nios II IDE, you might get the following error message: "Unable to create this part due to an internal error. Reason for the failure: Editor could not be initialized.".

Workaround: Adjust the Windows launch arguments for the Nios II IDE editor. Perform the following steps:

1. On the Windows Start menu, browse to the **Nios II EDS 6.1** program icon, right click it, then click **Properties**. A Windows Properties dialog box appears.

2. In the **Target** field, add `vmargs -Xmx1024m` to the end of the path to the Nios II IDE executable. For example:

```
C:\altera\61\nios2eds\bin\eclipse\nios2-ide.exe -vmargs -Xmx1024m
```

Nios II IDE doesn't recognize when an elf file is built outside the IDE

If you use a build process external to the Nios II IDE, the IDE might not recognize when you have generated or re-generated the elf file for your project. In this case, the IDE debugger might report an error that the elf file doesn't exist.

Workaround: Right click on the project name and click **Refresh** to allow the IDE to recognize the elf file.

After switching workspaces, the IDE appears to freeze while displaying the Nios II IDE splash screen

After clicking **Switch Workspace** on the File menu on a Windows machine, a Nios II IDE splash screen appears. Unfortunately, this splash screen obscures a dialog box which asks you to specify the new workspace.

Workaround: Press Alt-Tab to switch applications. You will see two relevant application icons: an Eclipse icon associated with the splash screen and a Nios II IDE icon associated with the workspace dialog box. Select the Nios II icon to bring the dialog box to the foreground.

C/C++ Scanner does not understand certain C/C++ constructs

The C/C++ scanner is used for C/C++ Search, navigation, open declaration and parts of content assist. Due to limitations of the C/C++ Scanner, these features will not work with the following code constructs:

- Kernighan & Ritchie-style C
- Functions that take a function-pointer as an argument

Workaround: If the C/C++ Search fails, use the File Search facility.

C2H Compiler

This section lists any issues related to the Nios II C-to-Hardware Acceleration (C2H) Compiler.

C2H Compiler out-of-memory error

The C2H Compiler might generate an out-of-memory error at compile time for code that initializes a large local array.

Workaround: Initialize the memory with the Nios II processor and access it through pointers in the accelerator.

C2H Compiler doesn't accelerate subfunctions located in a separate file

When accelerating a function in a file, the C2H Compiler cannot link subfunctions that are defined in a different file.

Workaround: Include all subfunctions called by the accelerated function within the same source code file.

Accelerator returns -1 if C/C++ project doesn't rebuild before running

A call to an accelerated function might return the invalid result -1 if you do not build your C application project before running it on hardware. The Nios II IDE does not detect that a new FPGA configuration file (**.sof**) needs to be downloaded if the system library is not rebuilt, which might result in attempting to execute an accelerated function on hardware that does not contain the hardware accelerator.

Workaround: Build your C/C++ application project before running it on hardware.

Incorrect results from logical or conditional operation with side-effects

The C2H Compiler always evaluates both operands of logical (&&, ||) and conditional (?:) operators. This is different from expected ANSI C behavior, for which operands are evaluated left-to-right, and unnecessary operands are skipped. For example, in the expression `(i-- && j--)`, if the value of `i` is zero, the right-hand-side (RHS) expression should not evaluate (i.e., `j` should not be decremented). However, this C2H Compiler erroneously evaluates both sides unconditionally, causing `j` to be decremented. The following example expressions could suffer from the same issue: `(i-- || j--)`, `(cond ? i-- : j--)`

Workaround: Use logical and conditional operations whose operators have no side effects. Side effects include pre-/post-fix increment operations (`++`, `--`), memory operations (`*`, `[]`, `.`, `->`), and function calls.

Closed system library while working with the C2H Compiler

The C2H Compiler requires the system library to obtain important details about the system, and cannot function if the system library is closed.

Workaround: Ensure that the system library project in the Nios II IDE is open prior to building an application project that contains a hardware accelerator.

Launch SOPC Builder button in C2H view

When the Nios II IDE workspace contains multiple projects with multiple system libraries, the incorrect SOPC Builder system might open when you click the Launch SOPC Builder in the C2H view.

Workaround: Keep only one system library project open at a time while using the C2H Compiler.

Build clean causes build failure

Performing a clean build on a Nios II IDE project that contains a hardware accelerator can cause the next build to fail in the IDE, because the clean build erroneously deletes a file required by the C2H Compiler.

Workaround: Do not perform a clean build on projects that use hardware accelerators. If you have already performed a clean build, recompile with option **Build software, generate SOPC Builder system, and run Quartus II compilation** to regenerate the necessary files.

Multiple clock domains causes hardware accelerator to fail

If a hardware accelerator and the components connected to its master ports are in different clock domains, the accelerator might behave incorrectly.

Workaround: Assign a single clock to a hardware accelerator and all the slave ports it connects to. It is acceptable for the system to contain multiple clock domains.

Comma operators are not supported

The C2H Compiler does not support comma operators, such as the following example:

```
for(i = 0, j = 3; i < 10; i++, j++)
{
  /* statements */
}
```

Workaround: You can manually duplicate the same functionality, such as:

```
j = 3;
for(i = 0; i < 10; i++)
{
  /* statements */
  j++;
}
```

Array elements of structs do not copy correctly

C2H hardware accelerators do not correctly copy array elements of structs. For example:

```
typedef struct my_struct {
  int a;
  int b;
  int buf[BUF_SIZE];
}MY_STRUCT;

MY_STRUCT struct_a = {1, 2, {3, 3, 3, 3}};
MY_STRUCT struct_b = {9, 8, {7, 7, 7, 7}};

struct_a = struct_b;
```

In this example, the `a` and `b` elements of the struct will copy correctly, but the `buf` element will not. After this assignment, struct_a will equal `{9, 8, {3, 3, 3, 3}}`.

Workaround: Copy the array elements explicitly, as follows:

```
{
  int i=0;
  do
    {
      struct_a.buf[i] = struct_b.buf[i];
      i++;
    } while (i<LENGTH_OF_BUF_ELEMENT)
}
```

Simulating on ISS is not supported

The Nios II instruction set simulator (ISS) does not support custom SOPC Builder components, and therefore cannot simulate systems that use hardware accelerators. You might get the following internal error if attempting to simulate using the ISS:

```
Internal Error (unhandled exception) in file cosim_main.cpp
```

Workaround: Run the program on a hardware system that includes the hardware accelerator.

Hardware accelerators remain after deleting the software project

If a system contains C2H hardware accelerators, deleting the software project that defines the accelerators does not remove the accelerators from the hardware system, and the accelerator logic remains in the SOPC Builder system.

Workaround: To remove an accelerator from a system, delete the accelerator from the C2H view in the Nios II IDE first, and then recompile the software project. The C2H Compiler then removes the accelerator from the SOPC Builder system. Once the compilation is complete then the software application can be deleted from the workspace.

Changing build configurations produces unexpected results

The C2H Compiler does not support multiple build configurations (e.g. Release or Debug) in the Nios II IDE. After creating one or more accelerators in a particular configuration, the C2H Compiler will produce undefined results if you switch to a different build configurations and create more accelerators.

Workaround: For a specific SOPC Builder system and Nios II IDE project, specify C2H accelerators in only one build configuration. Note that you can still use multiple build configurations, as long as only one configuration specifies C2H Compiler settings.

Flash Programmer

This section lists any issues relating to the Nios II IDE flash programmer.

Communication issue with multi-CPU or multi-JTAG UART designs [December 2006]

See the issue with the same name under section Nios II IDE / Debugging Projects.

'No such file or directory' error [December 2006]

You might get this error when programming flash for a project stored in a path containing spaces. The flash programmer does not correctly handle spaces in the directory path. However, this error is benign, because flash programming will complete successfully.

Workaround: None required.

elf2flash elf size limit

The elf2flash utility supports **.elf** files up to approximately 24 MBytes in size. The elf2flash utility might fail with error "java.lang.OutOfMemoryError" on files larger than 24 MBytes.

Workaround: You can either lower the number of symbols in your elf file by turning off debug symbols, or specify less initialized data in the application.

Verifying flash with the nios2-flash-programmer command fails

Using the **nios2-flash-programmer** command line utility to verify flash contents using the `--verify` argument might result in a verify failure even though flash contents are correct. The failure message will be similar to the following:

```
Verifying 00000000 ( 0%)Failed to verify at around 00000000  Verify failed
```

Workaround: To work around this issue, avoid using the `--verify` argument with the **nios2-flash-programmer** command-line utility. A verification of flash contents can be done by reading the flash contents into a file using the `--read` argument, then comparing the file to the input file used to program flash.

Using the Nios II Flash Programmer with an override file stored in <Nios II EDS path>/bin might generate an error:

```
4 [main] nios2-flash-programmer 4440 _cygtls::handle_exceptions: Error while
dumping state (probably corrupted stack)  Segmentation fault (core dumped)
```

Workaround: To workaround this issue, move the override file from the <Nios II EDS path>/bin directory to the working directory of your Nios II IDE project. Then, specify the override file to **nios2-flash-programmer** using the `--override=` parameter. In Nios II IDE, you can do this in the **Additional nios2-flash-programmer arguments** field in the Flash Programmer dialog box. In command-line mode, the override file can be specified by adding the `--override=` argument to the command line, as follows:

```
nios2-flash-programmer --base-0x0 --override=my_fp_override.txt standard.flash
```

When used with a multiprocessor Nios II system, the Nios II Flash Programmer might produce an error:

```
There are two or more Nios II processors available which match the values
specified. Please use the configuration dialog to pick one, or specify the --
device and/or --instance parameters on the command line.
```

Workaround: In the Nios II IDE on the Flash Programmer dialog box, specify an appropriate `--instance=`<*correct instance value*> argument in **Additional nios2-flash-programmer arguments**. If using command line mode, add `--instance=`<*correct instance value*> to the command line. The correct instance value can be found in the project's system library project in the file **generated.sh**.

Download Cables & Debug Hardware

This section lists any issues related to download cables and other debug hardware.

Communication errors during run/debug sessions using older download cables

Debugging with the following Altera download cables might fail, due to electrical noise-related JTAG communication failures: USB-Blaster™ Rev A, ByteBlaster™, ByteBlasterMV™, ByteBlaster II, and MasterBlaster™ cables.

Currently, the only fully supported cable for downloading, debugging, or communicating with Nios II systems is the USB-Blaster Rev B cable or later. Revision B cables are clearly labeled as Revision B. (Revision A cables have no revision label.)

Workaround: Use a USB-Blaster Rev B cable. Older cables can be used, but they might encounter JTAG communication failures.

Development Boards

This section lists any issues related to Altera development boards.

Intermittent failures while accessing CompactFlash card

The Nios II Development Kit version 5.0 and higher includes a CompactFlash controller peripheral suitable for interfacing to CompactFlash cards in True IDE mode on Nios development boards. In order for True IDE mode to operate, CompactFlash cards require that the ATASEL_N input be driven to ground during power-up.

The CompactFlash controller peripheral includes a configurable power register used to power-cycle CompactFlash cards in Nios II software through a MOSFET on the Nios development boards. However, in certain development boards, power to the CompactFlash card will not turn off completely during this power cycle operation. Because of this, the CompactFlash might not sample the ATASEL_N pin during the power-cycle operation after FPGA configuration when this pin is driven to ground. Instead, the CompactFlash card might sample the ATASEL_N pin when power is first applied to the development board, when I/O are not yet driven by the FPGA (before FPGA configuration).

Workaround: If you encounter errors with CompactFlash when using the Nios development boards, try one of the following:

- Use a different CompactFlash card. Certain cards are more susceptible to the power-cycling issue than others.
- Modify the Nios development board. This is recommended for users who are familiar and comfortable with board-level modifications. Disconnect pin 9 (ATASEL_N) on the CompactFlash socket on your Nios development board and tie this pin to ground. Note that the CompactFlash socket uses a staggered numbering on the pins (starting from pin 1: 1, 26, 2, 27, ...); refer to the CompactFlash Association specification for right-angle surface-mount connectors for exact specifications on this connector. This modification will permanently enable True-IDE mode operation.

Toolchain (gcc, gdb, etc.)

This section lists any issues related to the Nios II compiler toolchain, such as gcc and gdb.

Breakpoints in C++ constructors fail to halt the processor

Breakpoints set in a C++ constructor might not halt the processor due to a widespread GNU GCC, GDB issue. This is not a Nios II IDE-specific issue.

Workaround: You can work around this issue by moving all of your constructor source code into another class method, called init. Then invoke this method from within the constructor.

Target Software

This section lists any issues related to software or drivers that target the Nios II processor.

Lightweight IP (lwIP) failure when using DHCP [December 2006]

The Lightweight IP stack may experience intermittent failures when using DHCP to acquire an IP address. When the failure occurs, the following is printed to stdout:

```
Assertion "dhcp_create_request: dhcp->p_out == NULL" failed at line 1283 in
/cygdrive/c/altera/61b169/nios2eds/components/altera_lwip/UCOSII/src/downloads/lwi
p-1.1.0/src/core/dhcp.c
```

Workaround: If possible, Altera recommends switching to the NicheStack TCP/IP Stack - Nios II Edition introduced in Nios II 6.1. If not, the following workarounds can be employed.

- Power-cycle the target board and re-attempt DHCP negotiation, which usually results in correct acquisition of an address from DHCP, assuming that the DHCP server is able to allocate IP addresses.
- Use a static IP address and disable DHCP.

Compiling an existing MicroC/OS-II application generates errors in v6.1

The version of MicroC/OS-II was upgraded in the v6.1 release, which can cause this issue. After importing a Nios II IDE project that uses MicroC/OS-II created with version 6.0SP1 or earlier, building the project in the Nios II IDE generates errors, such as

```
#error "OS_CFG.H, Missing OS_FLAGS_NBITS: Determine #bits used for event flags,
MUST be either 8, 16 or 32'
```

Workaround: Resetting the MicroC/OS-II software component settings to their defaults fixes this issue. Right-click on the application project in the Nios II C/C++ view, and click **System Library Properties**. Click **RTOS Options...** to display the MicroC/OS-II RTOS Options dialog box. Expand the **MicroC/OS-II** group in the upper-left-hand corner of the dialog box. Make note of your settings on each page of options. Click **Restore Defaults**, and then go back to each page and re-enter settings that changed. Click **OK** to dismiss the MicroC/OS-II RTOS Options dialog box, and click **OK** again to dismiss the Properties dialog box.

stdin, stdout, and stderr don't work in MicroC/OS-II applications when using small C library option

MicroC/OS-II applications using `stdin`, `stdout`, and `stderr` fail to operate when using the small C library.

Workaround: Disable the small C library option.

malloc(), realloc() failures with MicroC/OS-II

When using the MicroC/OS-II RTOS, calls to `malloc()` and `realloc()` might fail if successive calls to `malloc()` or `realloc()` within a MicroC/OS-II task occur after changing the task priority of the task in which a memory block was originally allocated.

Workarounds:

- Allocate and/or reallocate memory blocks outside of MicroC/OS-II tasks, before task switching starts. Changing thread priorities at runtime is now possible.
- Allocate fixed areas of memory using arrays (rather than using `malloc()`) before task switching starts. Changing thread priorities at runtime is now possible.
- Allocate memory using `malloc()` or `realloc()` from a MicroC/OS-II task. You may change task priorities at runtime, but only for tasks that have not used `malloc()` or `realloc()`.

cout from MicroC-OS/II task will not send data to STDOUT

If neither `printf()` or `cout` is used from `main()` before tasks are started, `cout` will not work from a task.

Workaround: Add the following C++ code to the beginning of `main()`:

```
std::ios_base::sync_with_stdio(false);
```

Problems using HAL drivers with Toshiba Flash

The HAL CFI Flash driver might not work for Toshiba flash memory that claims to be CFI compliant.

Workaround: In the **altera_avalon_cfi_flash_table.c** file, change the `#define READ_ARRAY_MODE` from `(alt_u8)0xFF` to `(alt_u8)0xF0` and rebuild the project.

Creating new custom HAL components

When you first create a component's inc directory or HAL header file, you might first need to perform a clean build (i.e., rebuild) of existing system library projects for the new files to be detected.

Legacy SDK

Support for the Legacy SDK mode was removed in version 6.0 of the Nios II Embedded Design Suite.

SOPC Builder and Quartus II Software

This section lists any issues related to the Quartus II software or SOPC Builder that specifically affect Nios II designers.

For further information on the Quartus II software, refer to the latest Quartus II release notes on the Altera web site:

http://www.altera.com/literature/lit-qts.jsp

Example Designs

This section lists any issues related to the example designs included with the Nios II Embedded Design Suite.

Hardware Designs

There are no known issues at this time.

Software Designs

RAM test failure when running Memory Test software template on the ISS

An issue in the instruction set simulator (ISS) model of the JTAG UART can cause a console communication error during the RAM test when running the Memory Test software template on the ISS.

Networking Examples

If you are running a networking example design and you are asked for a 9-digit number after the letters 'ASJ', and your Nios II development board does not have a sticker with a 9-digit number after the letters 'ASJ', please enter a unique 9-digit number when prompted. Ensure that this number is unique to each Nios board connected to your network to avoid network address conflicts.

Hardware Simulation

This section lists issues related to simulating Nios II processor systems on an RTL simulator, such as the ModelSim® simulator.

Error: "UNC paths are not supported. Defaulting to Windows directory." [December 2006]

If you launch ModelSim from a working directory that is mapped via a UNC path (a path that starts with `//` instead of drive letter), you will get this error message in SOPC Builder. This error occurs because ModelSim is calling a cmd shell, which does not support UNC paths.

Workaround: Map the UNC path to a drive letter and use the drive letter to reference the working directory in the launching shell.

Simulation failure if reset address is set to EPCS

Running ModelSim RTL simulation of a Nios II system fails if the reset address of the Nios II processor is set to an EPCS Serial Flash Controller.

Workaround: To simulate your system, temporarily set the Reset Address of the Nios II CPU to the memory that your application code will reside (for example, SDRAM), then re-generate the system in SOPC Builder and run

RTL simulation again. Before booting the Nios II CPU from EPCS flash on your target board, change the Nios II Reset Address back to the EPCS Controller peripheral and re-generate the system in SOPC Builder and re-compile in the Quartus II software to produce an updated FPGA configuration file with the Nios II CPU booting from EPCS flash.

Uninitialized BSS variables in simulation

If your program reads the value of an uninitialized BSS variable during HDL simulation when the HAL system library has been compiled with the **ModelSim only, no hardware support** property enabled in Nios II IDE, a warning will be produced about unfiltered data being ‘x’. This occurs because when this property is enabled, the code that clears the BSS memory region is omitted to speed up HDL simulation so this memory region is uninitialized. The BSS region contains global and static local variables that are not initialized by the application so they default to a value of zero. When the Nios II CPU reads uninitialized variables, it displays a warning and converts any of the bits of the uninitialized data to zero which correctly mimics the effect of the missing BSS clearing code. The HAL code that executes before and after main() may use BSS variables so these warnings might be generated even if your application doesn’t use the BSS.

ModelSim fails to load large memory models

The ModelSim tool might fail to load simulation models for memory arrays larger than 128M bytes, halfwords or words in size. If the sum of the following parameters is greater than 27, the ModelSim tool will fail to load:

- address_bits (i.e. 14)
- column bits (i.e. 11)
- log2(number of banks) (number of banks is usually 4, so this term is usually 2)
- log2(chipselects) (number of chipselects is usually 1, so this term is usually 0)

Workaround: Simulate using a smaller SDRAM than the SDRAM implemented in hardware. This is possible if the entire memory space doesn’t need to be simulated.

Documentation Issues

This section lists errors, unclear descriptions, or omissions from current published specifications or product documents.

GNU documentation is missing [December 2006]

The GNU documentation was erroneously omitted from the Nios II EDS v6.1 installation.

Workaround: The GNU tool documentation for the Nios II EDS version 6.1 is available at http://www.altera.com/support/kdb/solutions/rd12192006_126.html. Download the zip file and unzip it into your Nios II EDS directory, i.e. C:\altera\61\nios2eds.

GNU assembler does not accept the --defsym flag

According the GNU documentation, you can set an assembler definition by using the `--defsym` flag, but it does not work in the following form: `--defsym MY_VAR=1`

Contact Information

For more information, contact Altera's mySupport website at **www.altera.com/mysupport**. Click **Create New Service Request**, and choose the **Product Related Request** form.

Revision History

Table 1 shows the revision history for the Nios II Embedded Design Suite v6.1 Errata Sheet.

Table 1. Nios II Embedded Design Suite v6.1 Errata Sheet Revision History

Version	Date	Errata Summary
1.0	December 2006	First release
1.1	December 2006	Added the following issues: • Windows XP: Internal error when repeatedly stepping into with debugger [December 2006] • MicroC/OS-II source code not found while debugging [December 2006] • Communication issue with multi-CPU or multi-JTAG UART designs [December 2006] • The Restart command on the Run menu does not work [December 2006] • 'Additional nios2-terminal arguments' field is not functional [December 2006] • Internal Error when double-clicking on objdump file [December 2006] • 'No such file or directory' error [December 2006] • Lightweight IP (lwIP) failure when using DHCP [December 2006] • Error: "UNC paths are not supported. Defaulting to Windows directory." [December 2006] • GNU documentation is missing [December 2006]

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com
Applications Hotline:
(800) 800-EPLD
Literature Services:
literature@altera.com

Nios II Embedded Design Suite 6.0 Service Pack 1 Release Notes

June 2006, Version 6.0 SP1 Release Notes

This document lists the release notes for the Nios® II Embedded Design Suite (EDS) version 6.0 Service Pack 1 (6.0 SP1).

Table of Contents:

New Features & Enhancements

The Nios II EDS version 6.0 SP1 addresses issues found in previous releases, and adds the following primary feature:

- Example designs for the RoHS-compliant Nios II Development Board, Stratix II Edition.

The sections below provide a detailed list of all product updates.

Device & Host Support

This release supports the following Altera® FPGA families:

- Stratix® II
- Stratix
- Cyclone™ II
- Cyclone

This release supports the following host environments:

- Quartus® II software version 6.0 SP1
- Windows XP Professional, Windows 2000, 32-bit Linux 8.0, and Enterprise 3 (64-bit not supported)
- ModelSim® versions supported on Windows: 6.0e OEM, 6.1d OEM, 6.1b SE, 6.1c SE, 6.1d SE
- ModelSim versions supported on Linux: 6.0e OEM, 6.1d OEM

Installation and Licensing Instructions

This section describes how to install the service pack. There are two modes of delivery for the service pack:

- A service pack download, which must be installed on top of the Nios II EDS 6.0, is available on the Altera website.
- A CD-ROM, which contains the complete installation of the Nios II EDS 6.0 SP1, is included in Nios II development kits.

Installing the Nios II Embedded Design Suite Version 6.0 SP1 on Windows

To install the service pack on a Windows computer, you must have administrative privileges.

Installing the Service Pack Download

To install the service pack on a Windows computer using the installation file downloaded from www.altera.com, perform the following steps. You must first have the Quartus II software version 6.0 SP1 and Nios II EDS v6.0 installed.

1. Exit the Quartus II software and Nios II IDE before continuing.

2. Download the service pack installation file to your hard drive.

3. Run the executable service pack file from your hard drive.

4. Follow the on-screen instructions to install the service pack.

Installing from CD-ROM

To install the Nios II software on a Windows computer using the Nios II Embedded Design Suite CD-ROM, perform the following steps:

1. Insert the compact disc into your CD-ROM drive. The Nios II setup program appears automatically. If the Nios II setup program does not start automatically, browse to the CD-ROM drive in the Windows Explorer, and run the program **launcher.exe** at the top level of the CD.

2. Click **Install Nios II EDS**. The Nios II EDS installer starts and guides you through the installation process.

Installing the Nios II Embedded Design Suite version 6.0 SP1 on Linux

To install the service pack on a Linux workstation, you must have administrative privileges.

Before installing the Nios development tools on Linux, ensure that the shell has the `DISPLAY` environment variable pointing to a valid X server. Otherwise, the installer generates the following error:

```
Updating SOPC Builder components... Exception in thread "main"
java.lang.InternalError: Can't connect to X11 window server using ':0.0' as the value
of the DISPLAY variable.
at sun.awt.X11GraphicsEnvironment.initDisplay(Native Method)
at sun.awt.X11GraphicsEnvironment.<clinit>(Unknown Source)
at java.lang.Class.forName0(Native Method)
at java.lang.Class.forName(Unknown Source)
at java.awt.GraphicsEnvironment.getLocalGraphicsEnvironment(Unknown Source)
at java.awt.Font.initializeFont(Unknown Source)
at java.awt.Font.<init>(Unknown Source)
at sopc_wizard.sopc_ui.<clinit>(sopc_ui.java:31)
```

Installing the Service Pack Download

To install the service pack on a Linux workstation using the installation file downloaded from www.altera.com, perform the following steps. You must first have the Quartus II software v6.0 SP 1 and Nios II EDS v6.0 installed.

1. Exit the Quartus II software and Nios II IDE before continuing.

2. Download the platform-specific service pack installation file into a temporary directory.

3. Change directory into the temporary directory.

4. Type the following commands at a command shell:

```
tar -xf niosii_60sp1_linux.tar
cd niosii_60sp1_linux
./install
```

The installation script guides you through the installation process.

Installing from CD-ROM

To install the Nios II EDS on a Linux workstation using the Nios II Embedded Design Suite CD-ROM, perform the following steps:

1. Insert the compact disc into your CD-ROM drive. If the computer has problems reading the CD-ROM, you might have to mount the CD-ROM manually, by typing the following at a command shell:

   ```
   mount /mnt/cdrom
   ```

2. Type the following commands at a command shell:

   ```
   cd /mnt/cdrom
   ./install
   ```

The Nios II EDS installer starts and guides you through the installation process.

Installing the USB-Blaster Download Cable on Linux

To use the USB-Blaster download cable on Linux systems, you need to set up the permissions by adding the following lines to **/etc/hotplug/usb.usermap**. You need to do this before plugging in your USB-Blaster.

```
#
# Altera USB-Blaster
#
usbblaster 0x03 0x09fb 0x6001 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
usbblaster 0x03 0x09fb 0x6002 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
```

Then add the following script as **/etc/hotplug/usb/usbblaster**.

```
#!/bin/sh
# USB-Blaster hotplug script.
# Allow any user to access the cable
chmod 666 $DEVICE
```

This script sets up your USB-Blaster permissions so that any user can access it. Type `chmod +x /etc/hotplug/usb/usbblaster` to make the script executable.

Using Previously Installed Versions of the Nios II EDS

SOPC Builder and the Nios II IDE refer to the most recently installed version of components (such as the Nios II processor and peripherals) and their software drivers. To revert to a prior version of the Nios II development tools, you can reinstall the previous version of tools or modify the following environment variables.

- `SOPC_BUILDER_PATH` - Ensure that `SOPC_BUILDER_PATH` points to the installation directory of the desired Nios II version and no other Nios II versions.
- `SOPC_BUILDER_PATH_`<*version*> - Ensure that `SOPC_BUILDER_PATH_`<*version*> points to the installation directory of the corresponding Nios II version.
- `SOPC_KIT_NIOS2` - Ensure `SOPC_KIT_NIOS2` points to the installation directory of the desired Nios II version and no other Nios II versions.

If you have multiple versions of the Quartus II software installed, launch the supported version of Quartus II to ensure that the `QUARTUS_ROOTDIR` environment variable is updated.

Licensing

You can create, compile and generate time-limited FPGA programming files for Nios II hardware systems without obtaining a license file. To generate non-time-limited FPGA programming files and flash programming files, you must obtain a license for the Nios II processor core and the Quartus II software. See the getting started material included with the Nios II Embedded Design Suite. You do not need a license if you will only develop software using the Nios II IDE.

Nios II Processor Cores

This section describes changes to the Nios II processor cores.

There are no updates.

SOPC Builder

For complete revision history of SOPC Builder and the Quartus II software, refer to the release notes for the Quartus II software version 6.0 SP1. The *Quartus II Handbook, Volume 4: SOPC Builder* contains complete documentation for SOPC Builder.

Nios II IDE

This section describes changes to the Nios II integrated development environment (IDE).

Dual-ported memories

In version 6.0, dual-ported memories were treated as if they had only one slave port. The '_S1' and '_S2' portions of generated labels were dropped. This impacted the generated file, **system.h.** In systems containing multiple Nios II CPUs where the CPUs are connected to separate ports of the dual-port memory, **system.h** reported the S1 base address for both CPUs. This issue has been addressed in this service pack so that different base addresses can be used for the CPUs connected to the dual-port memory ports.

Note that if you have a dual-port memory in your SOPC Builder design where only the second slave is mastered by the Nios II CPU, you will see an overlapping section error when linking an ELF. If you ensure that the first slave is the one mastered by the CPU (the second does not have to be) then the linker will work properly.

Nios II C-to-Hardware Acceleration (C2H) Compiler

Pipelined memory accesses

In version 6.0, pipelined memory accesses might stall the pipeline by one cycle causing performance to degrade. This service pack changes the FIFO implementation of the C2H Compiler to correct the problem.

Using the C2H compiler on software projects from previous versions

This service pack fixes an issue with the C2H Compiler hanging during compilation of a project that compiled successfully in a previous (pre-6.0) version.

Incorrect results when using the C2H Compiler with VHDL designs

This service pack fixes an issue with the instantiation of lpm_clshift which caused incorrect results when using the C2H Compiler with VHDL designs.

Signed division by a power of two

In version 6.0, if code performing signed division by a power of two was accelerated, incorrect results were generated if the numerator was negative and a non-power of two. This is due to the arithmetic shift right operation being used for division which can lead to incorrect rounding. For example assume that `data_in` is -15. The following code will return the value of -2 instead of the correct result of -1:

```
int foo (int data_in)  {
  return (data_in / 8);
}
```

This issue is fixed in this service pack by using a full division circuit. It is recommended to use unsigned numerators if possible, because the full division circuit is only used for signed division by powers of two. If this cannot be avoided, a simple workaround is to convert signed values to unsigned first, perform the division by a power of two, then restore the quotient back to a negative value if required.

The following code will return -1 if `data_in` is -15. The optimizations below will avoid the creation of a full division circuit since the numerator is an unsigned data type:

```
int foo (int data_in)  {
  unsigned int temp_data_in;

  temp_data_in = (data_in<0) ? (0-data_in) : data_in;
  return ((data_in<0) ? (0-(temp_data_in/8)) : (temp_data_in/8));
}
```

Signed modulo by a power of two

In version 6.0, if code performing signed modulo by a power of two was accelerated, incorrect results were generated if the numerator was negative. This is due to the result not being sign extended. For example assume that `data_in` is -15. The following code will return the value of 1 instead of the correct result of -7:

```
int foo (int data_in)  {
  return (data_in % 8);
}
```

This issue is fixed in this service pack by using a full division circuit. It is recommended to use unsigned numerators if possible, because the full division circuit is only used for signed modulo by powers of two. If this cannot be avoided a simple workaround is convert signed values to unsigned first, perform the modulo by a power of two, then restore the result back to a negative value if required. The following is an example of a signed modulo operation of `data_in % 8`:

```
int foo (int data_in)  {
  unsigned int temp_data_in;
  temp_data_in = (data_in<0) ? (0-data_in) : data_in;
  return ((data_in<0) ? (temp_data_in % 8) : (temp_data_in % 8));
}
```

Return value cache coherency

In version 6.0, if code with a return value is accelerated, it might return incorrect results under certain circumstances. This problem is due to the C wrapper file not using cache-bypassing reads for the Nios II processor to read the return value from the hardware accelerator. This service pack fixes this issue.

Flash Programmer

This section describes changes to the flash programmer in the Nios II IDE.

Corrected benign errors when booting from Flash with Stratix II

The following benign errors are no longer displayed by the Flash Programmer when targeting a Stratix II device and booting from flash.

```
(SEVERE) elf2flash: Read error: File not found: c:/alt
era/kits/nios2_60/components/altera_nios2
(SEVERE) elf2flash: Error reading boot copier
(SEVERE) elf2flash: Error generating Flash file, exiting
```

Target Software

This section describes changes to Altera-provided target software which runs on the Nios II processor, such as the hardware abstraction layer (HAL) system library.

There are no updates.

Example Designs

Hardware Example Designs

Examples added for lead-free Nios II Development Board, Stratix II Edition

Example hardware designs for the RoHS-compliant Nios Development Board, Stratix II Edition are included in the *<Nios II EDS install path>*/**examples/<***verilog or vhdl***>/niosII_stratixII_2s60_rohs directory**.

LAN component updated in example designs for the Nios Development Board, Cyclone II Edition

The example designs were updated to use the latest LAN91C111 Ethernet component. This new component reflects the following updated timing parameters:

```
Read_Wait_States = "20ns";
Write_Wait_States = "20ns";
Setup_Time = "20ns";
```

```
Hold_Time = "20ns";
```

Previous values:

```
Read_Wait_States = "175ns";
Write_Wait_States = "175ns";
Setup_Time = "10ns";
Hold_Time = "5ns";
```

The examples for other development boards already contain the correct, updated component.

Software Example Designs

There are no updates.

Nios II Embedded Design Suite 6.0 Release Notes

May 2006, Version 6.0 | Release Notes

This document lists the release notes for the Nios® II Embedded Design Suite (EDS) version 6.0.

Table of Contents:

New Features & Enhancements

The Nios II EDS version 6.0 addresses issues found in previous releases, and adds the following primary features:

- Nios II C-to-Hardware Acceleration (C2H) Compiler
- Floating-point custom instructions
- `cpu_resetrequest` and `cpu_resettaken` signals for individually resetting a Nios II processor core

The sections below provide a detailed list of all product updates.

Device & Host Support

This release supports the following Altera® FPGA families:

- Stratix® II
- Stratix
- Cyclone™ II
- Cyclone

This release supports the following host environments:

- Quartus® II software version 6.0
- Windows XP Professional, Windows 2000, 32-bit Linux 8.0, and Enterprise 3 (64-bit not supported)
- ModelSim® versions supported on Windows: 6.0e OEM, 6.1d OEM, 6.1b SE, 6.1c SE, 6.1d SE
- ModelSim versions supported on Linux: 6.0e OEM, 6.1d OEM

Installation and Licensing Instructions

This section describes how to install the Nios II EDS.

To install version 6.0 of the Nios II EDS, you must first install the Quartus II software version 6.0. You must have administrative privileges to install the Nios II EDS. See the *Quartus II Installation & Licensing Guide* for Quartus II system requirements and installation procedures.

If you have previous versions of the Nios II EDS installed on your system, Altera recommends that you uninstall the previous version(s) before installing version 6.0. If you need to retain a previous version on your system, install the Nios II EDS v6.0 to a different location.

Installing the Nios II Embedded Design Suite Version 6.0 on Windows

To install the Nios II software on a Windows computer using the Nios II Embedded Design Suite CD-ROM, perform the following steps:

1. Insert the Compact disc into your CD-ROM drive. The Nios II setup program appears automatically. If the Nios II setup program does not start automatically, browse to the CD-ROM drive in the Windows Explorer, and run the program **launcher.exe**. at the top level of the CD.

2. Click **Install Nios II EDS**. The Nios II EDS installer starts and guides you through the installation process.

Installing the Nios II Embedded Design Suite version 6.0 on Linux

To install the Nios II EDS on a Linux workstation using the Nios II Embedded Design Suite CD-ROM, perform the following steps:

1. Insert the Compact Disk into your CD-ROM drive. If the computer has problems reading the CD-ROM, you might have to mount the CD-ROM manually, by typing the following at a command shell:

   ```
   mount /mnt/cdrom
   ```

2. Type the following commands at a command shell:

   ```
   cd /mnt/cdrom
   ./install
   ```

   The Nios II EDS installer starts and guides you through the installation process.

Installing the USB-Blaster Download Cable on Linux

To use the USB-Blaster download cable on Linux systems, you need to set up the permissions by adding the following lines to **/etc/hotplug/usb.usermap**. You need to do this before plugging in your USB-Blaster.

```
#
# Altera USB-Blaster
#
usbblaster 0x03 0x09fb 0x6001 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
usbblaster 0x03 0x09fb 0x6002 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0
```

Then add the following script as **/etc/hotplug/usb/usbblaster**.

```
#!/bin/sh
# USB-Blaster hotplug script.
# Allow any user to access the cable
chmod 666 $DEVICE
```

This script sets up your USB-Blaster permissions so that any user can access it. Type `chmod +x /etc/hotplug/usb/usbblaster` to make the script executable.

Using Previously Installed Versions of the Nios II EDS

SOPC Builder and the Nios II IDE refer to the most recently installed version of components (such as the Nios II processor and peripherals) and their software drivers. To revert to a prior version of the Nios II development tools, you can reinstall the previous version of tools or modify the following environment variables.

- `SOPC_BUILDER_PATH` - Ensure that `SOPC_BUILDER_PATH` points to the installation directory of the desired Nios II version and no other Nios II versions.
- `SOPC_BUILDER_PATH_`<*version*> - Ensure that `SOPC_BUILDER_PATH_`<*version*> points to the installation directory of the corresponding Nios II version.
- `SOPC_KIT_NIOS2` - Ensure `SOPC_KIT_NIOS2` points to the installation directory of the desired Nios II version and no other Nios II versions.

If you have multiple versions of the Quartus II software installed, launch the supported version of Quartus II to ensure that the `QUARTUS_ROOTDIR` environment variable is updated.

Licensing

You can create, compile and generate time-limited FPGA programming files for Nios II hardware systems without obtaining a license file. To generate non-time-limited FPGA programming files and flash programming files, you must obtain a license for the Nios II processor core and the Quartus II software. See the getting started material included with the Nios II Embedded Design Suite. You do not need a license if you will only develop software using the Nios II IDE.

Nios II Processor Cores

This section describes changes to the Nios II processor cores.

Floating-point custom instructions

The Nios II core now offers a set of optional predefined custom instructions that implement floating-point single-precision (32-bit) addition, subtraction, multiplication and division. For more information, see the chapters *Processor Architecture* and *Implementing the Nios II Processor in SOPC Builder* in the *Nios II Processor Reference Handbook*.

cpu_resetrequest and cpu_resettaken signals

Optional signals to reset the Nios II CPU without affecting other components in the SOPC Builder system has been added to the Nios II processor cores. For more information, see chapters *Processor Architecture* and *Implementing the Nios II Processor in SOPC Builder* in the *Nios II Processor Reference Handbook.*

flushi & initi instructions

The cycle count for `flushi` and `initi` instructions changed from 1 to 4 cycles for the Nios II/s and Nios II/f cores.

SOPC Builder

This section describes changes to SOPC Builder which affect Nios II designers. For complete revision history of SOPC Builder and the Quartus II software, refer to the release notes for the Quartus II software version 6.0. The *Quartus II Handbook, Volume 4: SOPC Builder* contains complete documentation for SOPC Builder.

Wider data width support

SOPC Builder now supports data bus widths of up to 1024 bits.

Master/slave connections retained while using Use option

Version 6.0 fixes an issue that caused the master-slave connections to be lost when the **Use** button is turned off in the SOPC Builder table of active components. SOPC Builder now remembers the connections when the **Use** button is turned on again.

Nios II IDE

This section describes changes to the Nios II integrated development environment (IDE).

Nios II C-to-Hardware Acceleration (C2H) Compiler support

Refer to the C-to-Hardware Acceleration Compiler section.

Dual-ported memories

In the Nios II IDE, dual-ported memories are now treated as if they have only one slave port. The '_S1' and '_S2' portions of generated labels have been dropped. This impacts both the generated files **system.h** and the linker command file. The software example Tightly Coupled Memory was impacted by this change. Any customized linker command files, and any source files which reference a dual-ported memory base address by its generated labels must be updated. For example, in the case of the Tightly Copuled Memory software design, **tcm.c** was modified to replace the symbol `TIGHTLY_COUPLED_DATA_MEMORY_S1_BASE` with the new label style, `TIGHTLY_COUPLED_DATA_MEMORY_BASE`.

Nios II C-to-Hardware Acceleration (C2H) Compiler

The Nios II C-to-Hardware Acceleration (C2H) Compiler is introduced in the Nios II EDS version 6.0. The C2H Compiler is a tool that allows you to create custom hardware accelerators directly from ANSI C source code, which can often improve the execution performance by an order of magnitude. Contact your Altera representative to enable the OpenCore Plus evaluation for the C2H Compiler or purchase a license. Documentation on the C2H Compiler is available on the Nios II literature page, **http://www.altera.com/literature/lit-nio2.jsp**.

Flash Programmer

This section describes changes to the flash programmer in the Nios II IDE.

There are no updates.

Target Software

This section describes changes to Altera-provided target software which runs on the Nios II processor, such as the hardware abstraction layer (HAL) system library.

Boot from EPCS with Stratix II support

The EPCS serial configuration controller now fully supports Nios II systems in the Stratix II device family. You can boot your Nios II software from the flash memory in the EPCS device by assigning the processor reset address to the EPCS device.

Legacy SDK not supported

Legacy SDK support has been removed from Nios II version 6.0. The option to enable Legacy SDK is no longer available in SOPC Builder.

Example Designs

Hardware Example Designs

Fast design on-chip memory

The **fast** examples designs for all development kits were changed to use a read latency of 2 instead of 1 for the on-chip memory.

Software Example Designs

There are no updates.

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com
Applications Hotline:
(800) 800-EPLD
Literature Services:
literature@altera.com

nsai
I.S. EN ISO 9001

Nios II Embedded Design Suite 6.0 Errata Sheet

June 2006 Errata Sheet

This document addresses known errata and documentation issues for the Nios® II Embedded Design Suite (EDS) version 6.0. Errata are functional defects or errors, which might cause the product to deviate from published specifications. Documentation issues include errors, unclear descriptions, or omissions from current published specifications or product documents. Errata items discovered after the release of Nios II EDS version 6.0 are marked with the date the items were added to this document.

Table of Contents:

For the most up-to-date errata for this release, refer to the errata sheet on the Altera website: **http://www.altera.com/literature/es/es_nios2eds_60.pdf**

Nios II Processor Core

This section lists any issues related to the Nios II processor cores.

"nios_cpu: Unknown Break Location nios_cpu/jtag_debug_module" error message [June 2006]

If you modify the Nios II core in an existing system by disabling the JTAG debug module in the Nios II configuration wizard, you might receive the error message "**nios_cpu: Unknown Break Location nios_cpu/jtag_debug_module**" during SOPC Builder generation.

Workaround: Delete the Nios II core from the system and re-add it, ensuring that the JTAG debug core is disabled in the wizard.

VHDL sensitivity list warnings during Quartus II synthesis [June 2006]

You may receive warnings similar to the following in the Quartus II software when compiling a VHDL Nios II system containing the JTAG debug module.

```
Warning (10492): VHDL Process Statement warning at
cpuname_jtag_debug_module.vhd(254): signal "usr1" is read inside the Process
Statement but isn't in the Process Statement's sensitivity list
Warning (10492): VHDL Process Statement warning at
cpuname_jtag_debug_module.vhd(254): signal "ena" is read inside the Process
Statement but isn't in the Process Statement's sensitivity list
```

These warnings are benign and can be ignored. The warnings are a result of the `ena` and `usr1` signals not being included in the debug module's sensitivity list.

Double-precision Floating-Point operations with floating-point custom instructions

Calls to double-precision floating-point functions in math.h will return less-precise results on Nios II processors using the floating-point custom instruction. Floating-point constants are forced to single-precision when the floating-point custom instructions is present, which affects the constants for the double-precision floating-point functions in libm.

Peripherals

This section lists any issues related to the Altera embedded peripherals included in the Quartus II software.

JTAG UART is unstable after device-wide reset

If the DEV_CLRn pin on the FPGA input has been assigned (in Quartus® II software) to generate a device-wide reset, and the FPGA is reset while the JTAG UART is active, then the JTAG UART might become unstable.

Workaround: Do not use the DEV_CLRn function on the FPGA. Turn off the Enable device wide reset (DEV_CLRn) setting in Quartus II software.

Host Platform

This section lists any issues related specifically to the host platform.

Linux: Out of memory error in IDE

On Linux, you might receive an out of memory error in the Nios II IDE when it is launched from SOPC Builder.

Workaround: Run the IDE directly from the command-line.

Linux: F1 Help in the Nios II IDE does not function on Linux

There is currently no workaround. This will be addressed in a future version.

Linux: Debugging with the Nios II ISS target can cause a process leak on Linux

If you try to interrupt or terminate a debug session targeting the Nios II instruction set simulator (ISS), you might see an error message "Interrupt Failed or Terminate Failed". This means that the nios2-iss process failed to terminate. The debug session appears to have terminated in the IDE, but the nios2-iss process still remains alive.

Workaround: Open a command shell and kill the nios2-iss process.

Linux: The Quartus II stand-alone programmer is not supported on Linux

There is no Quartus II stand-alone programmer for Linux. As a result, in the Nios II IDE the Quartus II Programmer command on the Tools menu has no effect. The IDE does not automatically launch the programmer when you attempt to download software to a board that does not match the expected hardware image.

Workaround: Launch the Quartus II software to access the Quartus II Programmer.

Windows: Frisk antivirus software causes SOPC Builder and Nios II Command Shell to be unresponsive

The SOPC Builder and Nios II Command Shell might become unresponsive if run while the Frisk antivirus software is running.

Workaround: Turn off the Dynamic Virus Checking feature of the Frisk software before running SOPC Builder or the Nios II Command Shell.

Device

This section lists any device-related issues.

Stratix II EP2S60 ES devices cannot use MRAM byte enables

Early shipments of the Nios II Development Kit, Stratix II Edition include an EP2S60 engineering sample (ES) device. Stratix II EP2S60 ES devices have a silicon problem that prevents the use of byte enables on MRAM blocks.

Workaround: Refer to the Stratix II FPGA Family Errata Sheet for details.

Nios II IDE

This section lists any issues relating to the Nios II IDE.

Building Projects

Nios II IDE unnecessarily updates the SOPC Builder system file (.ptf) [June 2006]

The Nios II IDE opens the SOPC Builder system file (**.ptf**) by invoking SOPC Builder during certain operations, which might cause SOPC Builder to change the date stamp of the file even though the system was not modified. This might cause problems if you are using a version control system.

Workaround: If you are not using the Nios II C2H Compiler, you can change the PTF file properties to read-only to prevent the IDE from changing the file.

"FATAL: can't create obj/<object filename>.o: Permission denied" error message [June 2006]

You might see the error message "FATAL: can't create obj/<object filename>.o: Permission denied" in the Nios II IDE if you attempt to rebuild a system library project while the C/C++ indexer is active.

Workaround: Ensure that the indexer is not active when you initiate a rebuild.

Incorrect address assignment for dual-port memory mastered by two different masters [June 2006]

If you have a dual-port memory in your Nios II system, an only the second slave port is mastered by the CPU, you might see an overlapping section error during the linking stage of building your software.

Workaround: In SOPC Builder, ensure that the first slave port of the dual-port memory is mastered by the Nios II CPU. The second port does not have to be mastered by the Nios II CPU.

Building Nios II Advanced project using Build Project invokes 'make clean all' by default

Nios II IDE Advanced projects created with earlier (pre-6.0) versions of the Nios II IDE call 'make clean all' when the project is built, rather than 'make all'. Also, on the Project menu, clicking Clean for these projects has no effect.

Workaround: To change the behavior of the IDE so that the project invokes make all by default for a build and make clean by default for a clean build, do the following:

1. Right click on the project and click **Properties**.

2. Click the **C/C++ Make Project** page.

3. In the **Workbench Build Behavior** section, change the value for **Rebuild (Full Build)** to **all**.

4. Turn on **Clean** and type `clean` in the corresponding text box.

Build errors after changing component names in SOPC Builder

If you rename components in the SOPC Builder system and then regenerate the SOPC Builder system, Nios II IDE system library projects based on that system will have build errors.

Workaround: After regenerating the SOPC Builder system, create a new system library project for the SOPC Builder system. Alternately, you can delete the system library project from the workspace without deleting the contents from the file system, and then re-import the project, selecting the appropriate SOPC Builder system.

Debugging Projects

"Step failed. Target is not responding (timed out)" error message [June 2006]

The Nios II IDE debugger might hang and report the above message if your code contains large arrays declared as local variables on the stack.

Workaround: Place the array and any other large buffers on the heap rather than on the stack.

Incorrect breakpoint filtering on threads

If you enable breakpoint filtering for a thread and later turn off filtering for the thread, the debugger might incorrectly continue to filter the thread.

Uninitialized Memory Error when executing from ISS

Under some cases the ISS does not ignore uninitialized memory reads, even when Unitialized memory reads is set to Ignore on the ISS Settings tab of the run configuration.

Nios2-gdb-server fails to terminate after setting a watch point

You might be unable to terminate nios2-gdb-server after setting a watchpoint in the Nios II IDE debugger and resuming execution past the end of main. You will see an error "Terminate failed". You will not be able to start the debugger again; you will see a message reading "Another application is using the target processor..." in the Console view.

Workaround: Terminate the nios2-gdb-server.exe process manually using the Windows Task Manager.

Watchpoints do not work when set on variables whose size are not 32-bits

Workaround: Change the type of global and static local variables to int, long, or unsigned long before setting watchpoints on them.

Debugger cannot step into __sflags, and continues execution instead

The Nios II IDE debugger is unable to step into some low-level C library functions, such as `__sflags()` which is called from `_fopen_r()`. (`_fopen_r()` is called from `fopen()`.) If you try to step into such a function, execution will proceed as if you had indicated the debugger should resume execution.

Workaround: Step over such functions. Or, if execution continues after trying to step in, click **Suspend** on the Run menu.

Missing traced load/store instruction and data in the Trace view

If the trace options **Include load addresses**, **Include store addresses** or **Include data values** are enabled during debug, the load and store address and data will not appear at the first breakpoint after starting debugging. They will appear at successive breakpoints.

Workaround: To see load or store addresses and data in the instruction trace prior to main, turn on **Break at alt_main()** located on the **Debugger** tab for your debug configuration.

Cannot use watchpoints in the Nios II IDE when the FS2 console is open

Watchpoints do not work in the Nios II IDE when the **Use FS2 console window for trace and watchpoint support** setting is turned on in the **Debugger** tab of the Debug configuration. You will see an error message "The execution of program is suspended because of error." with details indicating that hardware watchpoints could not be inserted and deleted.

Workaround: If the FS2 console is open, you must use it to control watchpoints. For details, see the FS2 documentation.

Breakpoints on adjacent lines of assembly fail to halt the processor

Setting breakpoints on adjacent lines of assembly code might cause the Nios II processor to stop responding to the debugger.

Workaround: When debugging in mixed mode or disassembly view, separate breakpoints by at least one assembly instruction. This issue does not affect Nios II cores that do not have hardware breakpoints enabled in the JTAG debug module.

Navigating Projects

Nios II IDE might not prompt for workspace at startup

The Nios II IDE might not prompt for the workspace at startup.

Workaround: Specify the workspace at the command line using the -data argument. For example:

```
nios2-ide -data <path to workspace>
```

If the workspace does not exist, it will be created.

*Nios II IDE stops responding after you double-click a **.o** file.*

Double-clicking a **.o** file in the Nios II IDE on Windows might cause the IDE to stop responding. This issue occurs only on systems with a separate installation of Cygwin in addition to Cygwin installed with the Nios II development tools.

Workaround: Ensure that only the **cygwin1.dll** for the Nios II tools is in the PATH when launching Nios II IDE.

Error importing a software project created in Nios II IDE version 5.1 into Nios II IDE version 6.0

If you create a software project in Nios II IDE version 5.1 and then install Nios II version 6.0 and try to import that same project into your workspace, you might get an error "Problem deleting folder Debug".

Workaround: If you get this error, close the Nios II IDE and any other application which has files open from the project in question, then manually delete the Debug and/or Release directories from the file system. Re-import the project into the Nios II IDE version 6.0

Resource(s) out of sync with the file system when searching for files in the workspace

When searching through files in the IDE workspace, you might get an error message saying that one or more resources are out of sync with the file system.

Workaround: Right click in the Navigator view and click **Refresh**, and then perform the search again.

C/C++ Scanner does not understand certain C/C++ constructs

The C/C++ scanner is used for C/C++ Search, navigation, open declaration and parts of content assist. Due to limitations of the C/C++ Scanner, these features will not work with the following code constructs:

- Kernighan & Ritchie-style C
- Functions that take a function-pointer as an argument

Workaround: If the C/C++ Search fails, use the File Search facility.

C2H Compiler

This section lists any issues related to the Nios II C-to-Hardware Acceleration (C2H) Compiler.

Accelerator returns -1 if C/C++ project doesn't rebuild before running

A call to an accelerated function might return the invalid result -1 if you do not build your C application project before running it on hardware. The Nios II IDE does not detect that a new FPGA configuration file (**.sof**) needs to be downloaded if the system library is not rebuilt, which might result in attempting to execute an accelerated function on hardware that does not contain the hardware accelerator.

Workaround: Build your C/C++ application project before running it on hardware.

Incorrect results from logical or conditional operation with side-effects

The C2H Compiler always evaluates both operands of logical (&&, ||) and conditional (?:) operators. This is different from expected ANSI C behavior, for which operands are evaluated left-to-right, and unnecessary operands are skipped. For example, in the expression `(i-- && j--)`, if the value of `i` is zero, the right-hand-side (RHS) expression should not evaluate (i.e., `j` should not be decremented). However, this C2H Compiler erroneously evaluates both sides unconditionally, causing `j` to be decremented. The following example expressions could suffer from the same issue: `(i-- || j--)`, `(cond ? i-- : j--)`

Workaround: Use logical and conditional operations whose operators have no side effects. Side effects include pre-/post-fix increment operations (`++`, `--`), memory operations (`*`, `[]`, `.`, `->`), and function calls.

Closed system library while working with the C2H Compiler

The C2H Compiler requires the system library to obtain important details about the system, and cannot function if the system library is closed.

Workaround: Ensure that the system library project in the Nios II IDE is open prior to building an application project that contains a hardware accelerator.

Launch SOPC Builder button in C2H view

When the Nios II IDE workspace contains multiple projects with multiple system libraries, the incorrect SOPC Builder system might open when you click the Launch SOPC Builder in the C2H view.

Workaround: Keep only one system library project open at a time while using the C2H Compiler.

Build clean causes build failure

Performing a clean build on a Nios II IDE project that contains a hardware accelerator can cause the next build to fail in the IDE, because the clean build erroneously deletes a file required by the C2H Compiler.

Workaround: Do not perform a clean build on projects that use hardware accelerators. If you have already performed a clean build, recompile with option **Build software, generate SOPC Builder system, and run Quartus II compilation** to regenerate the necessary files.

Multiple clock domains causes hardware accelerator to fail

If a hardware accelerator and the components connected to its master ports are in different clock domains, the accelerator might behave incorrectly.

Workaround: Assign a single clock to a hardware accelerator and all the slave ports it connects to. It is acceptable for the system to contain multiple clock domains.

Comma operators are not supported

The C2H Compiler does not support comma operators, such as the following example:

```
for(i = 0, j = 3; i < 10; i++, j++)
{
  /* statements */
}
```

Workaround: You can manually duplicate the same functionality, such as:

```
  j = 3;
  for(i = 0; i < 10; i++)
  {
    /* statements */
    j++;
  }
```

Array elements of structs do not copy correctly

C2H hardware accelerators do not correctly copy array elements of structs. For example:

```
  typedef struct my_struct {
    int a;
    int b;
    int buf[BUF_SIZE];
  }MY_STRUCT;

  MY_STRUCT struct_a = {1, 2, {3, 3, 3, 3}};
  MY_STRUCT struct_b = {9, 8, {7, 7, 7, 7}};

  struct_a = struct_b;
```

In this example, the `a` and `b` elements of the struct will copy correctly, but the `buf` element will not. After this assignment, struct_a will equal `{9, 8, {3, 3, 3, 3}}`.

Workaround: Copy the array elements explicitly, as follows:

```
  {
    int i=0;
    do
      {
        struct_a.buf[i] = struct_b.buf[i];
        i++;
      } while (i<LENGTH_OF_BUF_ELEMENT)
  }
```

Simulating on ISS is not supported

The Nios II instruction set simulator (ISS) does not support custom SOPC Builder components, and therefore cannot simulate systems that use hardware accelerators. You might get the following internal error if attempting to simulate using the ISS:

```
  Internal Error (unhandled exception) in file cosim_main.cpp
```

Workaround: Run the program on a hardware system that includes the hardware accelerator.

Hardware accelerators remain after deleting the software project

If a system contains C2H hardware accelerators, deleting the software project that defines the accelerators does not remove the accelerators from the hardware system, and the accelerator logic remains in the SOPC Builder system.

Workaround: To remove an accelerator from a system, delete the accelerator from the C2H view in the Nios II IDE first, and then recompile the software project. The C2H Compiler then removes the accelerator from the

SOPC Builder system. Once the compilation is complete then the software application can be deleted from the workspace.

Changing build configurations produces unexpected results

The C2H Compiler does not support multiple build configurations (e.g. Release or Debug) in the Nios II IDE. After creating one or more accelerators in a particular configuration, the C2H Compiler will produce undefined results if you switch to a different build configurations and create more accelerators.

Workaround: For a specific SOPC Builder system and Nios II IDE project, specify C2H accelerators in only one build configuration. Note that you can still use multiple build configurations, as long as only one configuration specifies C2H Compiler settings.

Flash Programmer

This section lists any issues relating to the Nios II IDE.

elf2flash elf size limit

The elf2flash utility supports **.elf** files up to approximately 24 MBytes in size. The elf2flash utility might fail with error "java.lang.OutOfMemoryError" on files larger than 24 MBytes.

Workaround: You can either lower the number of symbols in your elf file by turning off debug symbols, or specify less initialized data in the application.

Download Cables & Debug Hardware

This section lists any issues related to download cables and other debug hardware.

Communication errors during run/debug sessions using older download cables

Debugging with the following Altera download cables might fail, due to electrical noise-related JTAG communication failures: USB-Blaster™ Rev A, ByteBlaster™, ByteBlasterMV™, ByteBlaster II, and MasterBlaster™ cables.

Currently, the only fully supported cable for downloading, debugging, or communicating with Nios II systems is the USB-Blaster Rev B cable or later. Revision B cables are clearly labeled as Revision B. (Revision A cables have no revision label.)

Workaround: Use a USB-Blaster Rev B cable. Older cables can be used, but they might encounter JTAG communication failures.

Development Boards

This section lists any issues related to Altera development boards.

Intermittent failures while accessing CompactFlash card

The Nios II Development Kit version 5.0 and higher includes a CompactFlash controller peripheral suitable for interfacing to CompactFlash cards in True IDE mode on Nios development boards. In order for True IDE mode to operate, CompactFlash cards require that the ATASEL_N input be driven to ground during power-up.

The CompactFlash controller peripheral includes a configurable power register used to power-cycle CompactFlash cards in Nios II software through a MOSFET on the Nios development boards. However, in certain development boards, power to the CompactFlash card will not turn off completely during this power cycle operation. Because of this, the CompactFlash might not sample the ATASEL_N pin during the power-cycle operation after FPGA configuration when this pin is driven to ground. Instead, the CompactFlash card might sample the ATASEL_N pin when power is first applied to the development board, when I/O are not yet driven by the FPGA (before FPGA configuration).

Workaround: If you encounter errors with CompactFlash when using the Nios development boards, try one of the following:

- Use a different CompactFlash card. Certain cards are more susceptible to the power-cycling issue than others.
- Modify the Nios development board. This is recommended for users who are familiar and comfortable with board-level modifications. Disconnect pin 9 (ATASEL_N) on the CompactFlash socket on your Nios development board and tie this pin to ground. Note that the CompactFlash socket uses a staggered numbering on the pins (starting from pin 1: 1, 26, 2, 27, ...); refer to the CompactFlash Association specification for right-angle surface-mount connectors for exact specifications on this connector. This modification will permanently enable True-IDE mode operation.

Toolchain (gcc, gdb, etc.)

This section lists any issues related to the Nios II compiler toolchain, such as gcc and gdb.

Breakpoints in C++ constructors fail to halt the processor

Breakpoints set in a C++ constructor might not halt the processor due to a widespread GNU GCC, GDB issue. This is not a Nios II IDE-specific issue.

Workaround: You can work around this issue by moving all of your constructor source code into another class method, called init. Then invoke this method from within the constructor.

Target Software

This section lists any issues related to software or drivers that target the Nios II processor.

malloc(), realloc() failures with MicroC/OS-II

When using the MicroC/OS-II RTOS, calls to `malloc()` and `realloc()` might fail if successive calls to `malloc()` or `realloc()` within a MicroC/OS-II task occur after changing the task priority of the task in which a memory block was originally allocated.

Workarounds:

- Allocate and/or reallocate memory blocks outside of MicroC/OS-II tasks, before task switching starts. Changing thread priorities at runtime is now possible.
- Allocate fixed areas of memory using arrays (rather than using `malloc()`) before task switching starts. Changing thread priorities at runtime is now possible.
- Allocate memory using `malloc()` or `realloc()` from a MicroC/OS-II task. You may change task priorities at runtime, but only for tasks that have not used `malloc()` or `realloc()`.

cout from MicroC-OS/II task will not send data to STDOUT

If neither `printf()` or `cout` is used from `main()` before tasks are started, `cout` will not work from a task.

Workaround: Add the following C++ code to the beginning of `main()`:

```
std::ios_base::sync_with_stdio(false);
```

Problems using HAL drivers with Toshiba Flash

The HAL CFI Flash driver might not work for Toshiba flash memory that claims to be CFI compliant.

Workaround: In the **altera_avalon_cfi_flash_table.c** file, change the `#define READ_ARRAY_MODE` from `(alt_u8)0xFF` to `(alt_u8)0xF0` and rebuild the project.

Creating new custom HAL components

When you first create a component's inc directory or HAL header file, you might first need to perform a clean build (i.e., rebuild) of existing system library projects for the new files to be detected.

Legacy SDK

Support for the Legacy SDK mode is removed in version 6.0 of the Nios II Embedded Design Suite.

SOPC Builder and Quartus II Software

This section lists any issues related to the Quartus II software or SOPC Builder that specifically affect Nios II designers.

For further information on the Quartus II software, refer to the latest Quartus II release notes on the Altera web site:

http://www.altera.com/literature/lit-qts.jsp

Cannot connect Nios II tightly-coupled instruction and data masters to the same dual-port memory

SOPC Builder does not generate an error if you connect Nios II tightly-coupled instruction and data masters to both ports on a dual-port on-chip memory. However, this configuration is not supported in hardware.

Workaround: Do not connect Nios II tightly-coupled masters to both ports of a dual-port memory.

Example Designs

This section lists any issues related to the example designs included with the Nios II Embedded Design Suite.

Hardware Designs

There are no known issues at this time.

Software Designs

RAM test failure when running Memory Test software template on the ISS

An issue in the instruction set simulator (ISS) model of the JTAG UART can cause a console communication error during the RAM test when running the Memory Test software template on the ISS.

Networking Examples

If you are running a networking example design and you are asked for a 9-digit number after the letters 'ASJ', and your Nios II development board does not have a sticker with a 9-digit number after the letters 'ASJ', please enter a unique 9-digit number when prompted. Ensure that this number is unique to each Nios board connected to your network to avoid network address conflicts.

Hardware Simulation

This section lists issues related to simulating Nios II processor systems on an RTL simulator, such as the ModelSim® simulator.

Simulation failure if reset address is set to EPCS

Running ModelSim RTL simulation of a Nios II system fails if the reset address of the Nios II processor is set to an EPCS Serial Flash Controller.

Workaround: To simulate your system, temporarily set the Reset Address of the Nios II CPU to the memory that your application code will reside (for example, SDRAM), then re-generate the system in SOPC Builder and run RTL simulation again. Before booting the Nios II CPU from EPCS flash on your target board, change the Nios II Reset Address back to the EPCS Controller peripheral and re-generate the system in SOPC Builder and re-compile in the Quartus II software to produce an updated FPGA configuration file with the Nios II CPU booting from EPCS flash.

Uninitialized BSS variables in simulation

If your program reads the value of an uninitialized BSS variable during HDL simulation when the HAL system library has been compiled with the **ModelSim only, no hardware support** property enabled in Nios II IDE, a warning will be produced about unfiltered data being ‘x’. This occurs because when this property is enabled, the code that clears the BSS memory region is omitted to speed up HDL simulation so this memory region is uninitialized. The BSS region contains global and static local variables that are not initialized by the application so they default to a value of zero. When the Nios II CPU reads uninitialized variables, it displays a warning and converts any of the bits of the uninitialized data to zero which correctly mimics the effect of the missing BSS clearing code. The HAL code that executes before and after main() may use BSS variables so these warnings might be generated even if your application doesn't use the BSS.

ModelSim fails to load large memory models

The ModelSim tool might fail to load simulation models for memory arrays larger than 128M bytes, halfwords or words in size. If the sum of the following parameters is greater than 27, the ModelSim tool will fail to load:

- address_bits (i.e. 14)
- column bits (i.e. 11)
- log2(number of banks) (number of banks is usually 4, so this term is usually 2)
- log2(chipselects) (number of chipselects is usually 1, so this term is usually 0)

Workaround: Simulate using a smaller SDRAM than the SDRAM implemented in hardware. This is possible if the entire memory space doesn't need to be simulated.

Documentation Issues

This section lists errors, unclear descriptions, or omissions from current published specifications or product documents.

EPCS Controller memory usage

The chapter *EPCS Device Controller Core with Avalon Interface* in the *Quartus II Handbook volume 5: Embedded Peripherals* utilizes 1Kbyte on-chip memory for its boot-loader program. However, when targetting Stratix II devices, the on-chip memory required for the boot-loader increases and requires two M4K blocks.

Error in UART Core with Avalon Interface Chapter

The chapter *UART Core with Avalon Interface* in the *Quartus II Handbook volume 5: Embedded Peripherals* incorrectly states: "When parity is Even, the parity bit is 1 if the character has an even number of 1 bits; otherwise the parity bit is 0. Similarly, when parity is Odd, the parity bit is 1 if the character has an odd number of 1 bits."

This should read: "When parity is Even, the parity bit is 1 if the number of 1's in the character plus the parity bit is even; otherwise the parity bit is 0. Or in other words, the Parity bit is set to '0' when there is an even number of '1' bits in the character. Similarly, when parity is Odd, the parity bit is 1 if the number of 1's in the character plus the parity bit has an odd number of 1 bits. Or in other words, the Parity bit is set to '0' when there is an odd number of '1' bits in the character."

Nios II IDE help system fails to display content

To display the help content, your computer must able to recognize itself on the network. Incorrect proxy settings can cause the help content to fail to display.

Workaround: Specify valid browser proxy settings.

byteenables in Avalon Interface Specification

The *Avalon Interface Specification* incorrectly describes the behavior for `byteenable` during burst transfers. The *Avalon Interface Specification States*: "The master port can assert or deassert the byteenable lines as necessary for each individual transfer within the burst." During Avalon master transfers, the master port must assert all byteenable for all transfers in the burst.

Workaround: Assert all `byteenable` lines during master burst transfers.

burstcount in Avalon Interface Specification

The *Avalon Interface Specification* incorrectly describes the behavior for `address` and `burstcount` during burst transfers. The *Avalon Interface Specification* states: "The start of a write burst is similar to the start of a fundamental master write transfer. The master port asserts `address`, `writedata`, `write`, and `byteenable` (if present) in addition to `burstcount`. ... This is the only time that the Avalon switch fabric captures `burstcount` and `address`; the master port can deassert them through the remainder of the burst." However, during Avalon master transfers, the master port must assert constant values on `address` and `burstcount` for the duration of the burst.

Workaround: During Avalon master transfers, assert constant values on `address` and `burstcount` for the duration of the burst.

Contact Information

For more information, contact Altera's mySupport website at **www.altera.com/mysupport**. Click **Create New Service Request**, and choose the **Product Related Request** form.

Revision History

Table 1 shows the revision history for the Nios II Embedded Design Suite v6.0 Errata Sheet.

Table 1. Nios II Embedded Design Suite v6.0 Errata Sheet Revision History

Version	Date	Errata Summary
1.0	May 2006	First release
1.1	June 2006	Added new items: • "nios_cpu: Unknown Break Location nios_cpu/jtag_debug_module" error message • VHDL sensitivity list warnings during Quartus II synthesis • Nios II IDE unnecessarily updates the SOPC Builder system file (.ptf) • "FATAL: can't create obj/<object filename>.o: Permission denied" error message • Incorrect address assignment for dual-port memory mastered by two different masters • "Step failed. Target is not responding (timed out)" error message

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com
Applications Hotline:
(800) 800-EPLD
Literature Services:
literature@altera.com