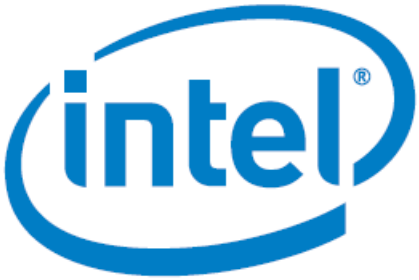


Nios II Ethernet Simple Socket Server on MAX10 FPGA DECA Kit from Arrow

Date: 5-Feb-2020

Revision: 1.0

Table of Contents

Overview

The following external parts are needed to demonstrate the design example;

- MAX10 10M50 FPGA DECA Development kit from Arrow Electronics
- Mini USB cable for programming the device
- Ethernet cable

Theory of Operation

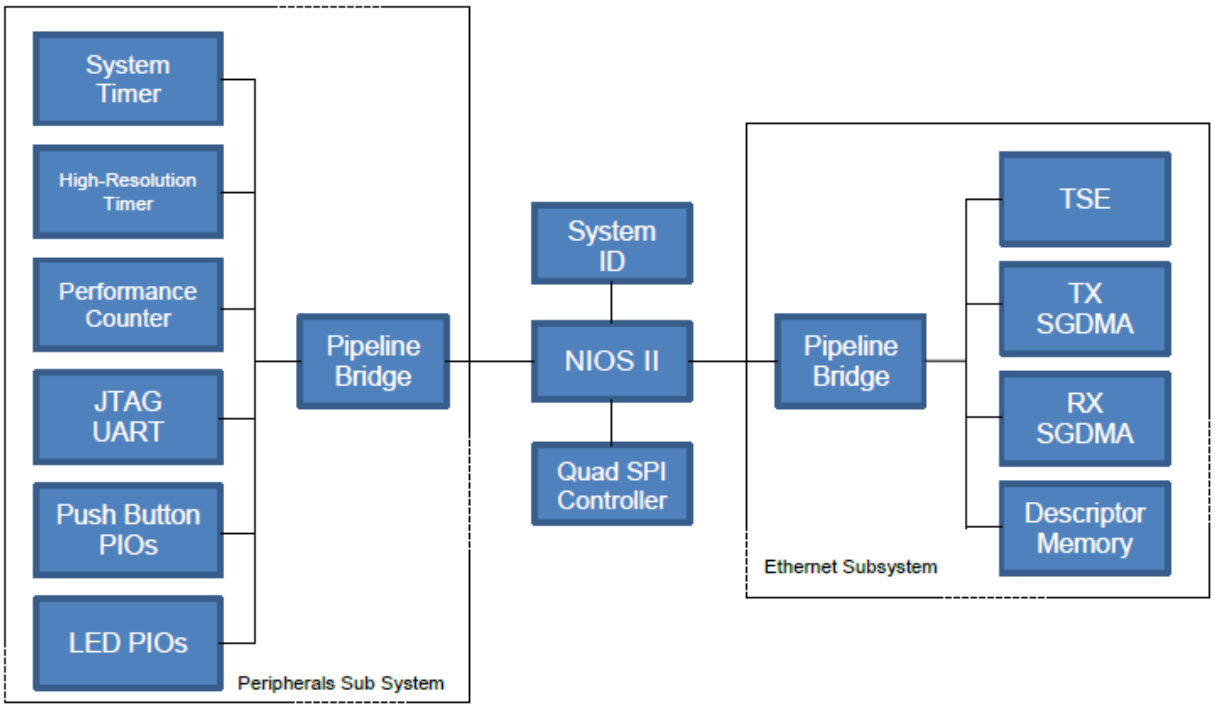


Figure 1: Ethernet Reference Design Block Diagram

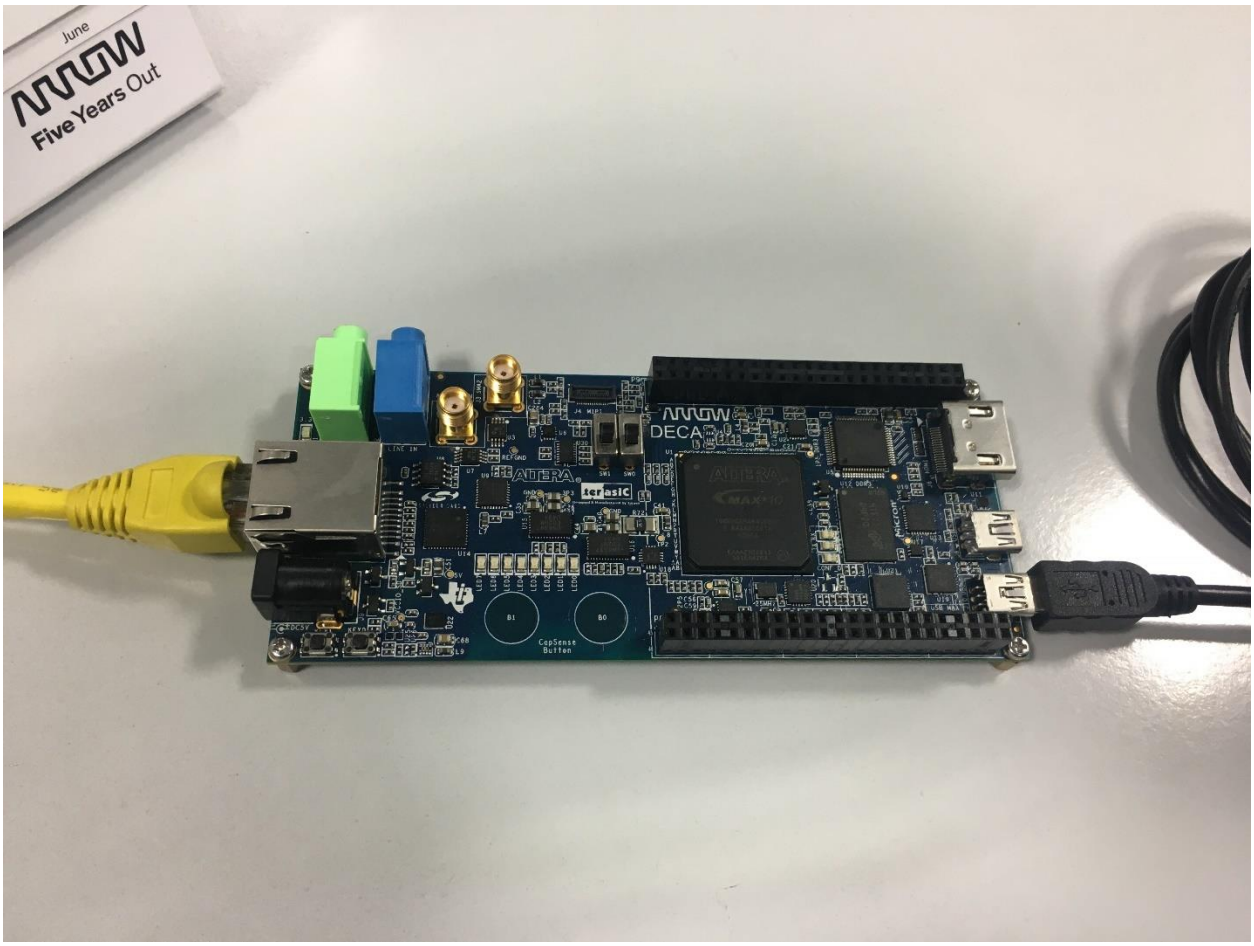
The Simple Socket Server design uses a modified version of Altera's Nios® II Ethernet Standard hardware design. It provides a mix of peripherals and memories similar to a typical Nios II processor system. This design interfaces with each hardware component on the Altera® development board, such as SDRAM, LEDs, push buttons, and an Ethernet physical interface or media access control (PHY/MAC). This Qsys-based hierarchical design has a top-level system and two subsystems, namely: peripherals subsystem and Ethernet subsystem as shown in Figure 1.

For more details on the Nios II Simple Socket Server implementation, please refer to page 1-16 on the following documentation.

Simple Demo Setup

1. Connect an Ethernet cable to its respective connector.

2. Power up the DECA kit using a mini USB to USB cable supplied along with the kit. The kit powers up with supply and JTAG via the same connector J10 which is near to the BBB connector. Your set up should look like below:

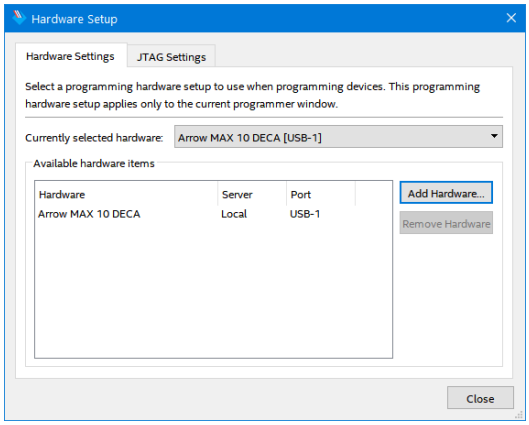


3. The design package includes a pre-compiled version of the design that downloads to the MAX 10 embedded flash. This version of the configuration includes the combined FPGA configuration and NIOS executable. Follow these instructions to load the design onto your kit:

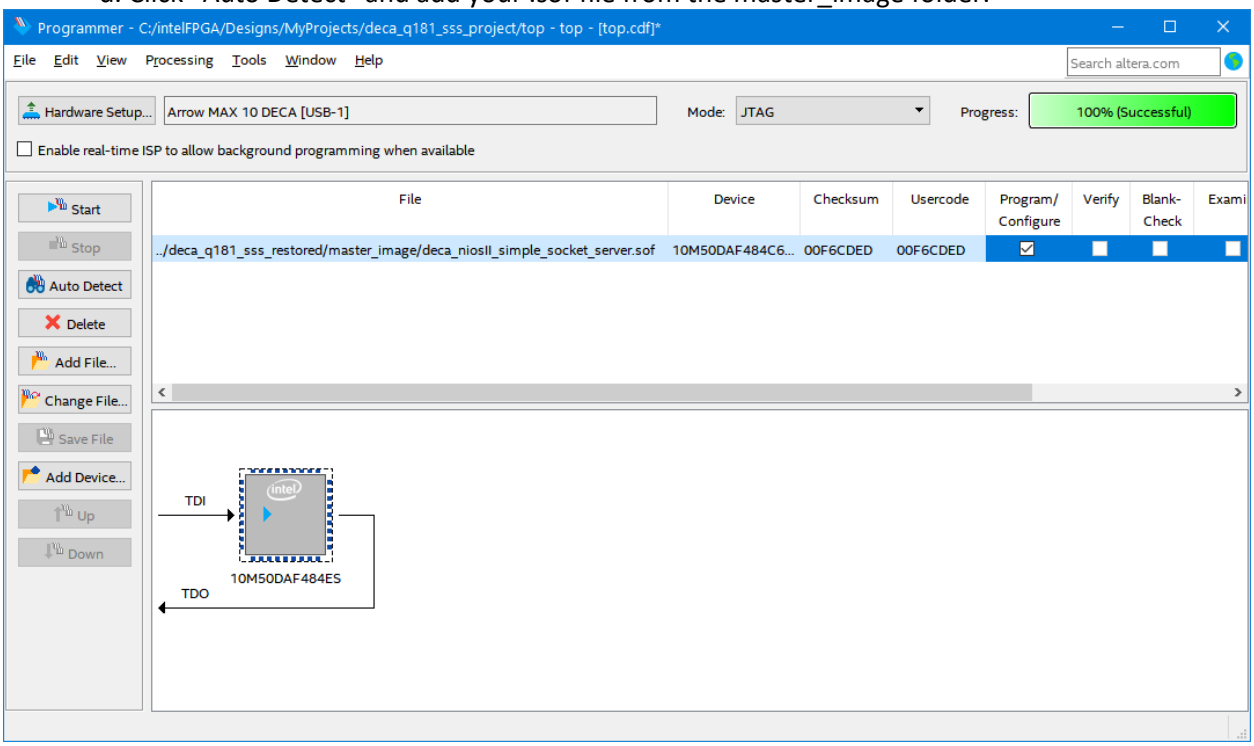
a. Make sure the board is powered on and that the USB is connected on J10

b. Startup Quartus. Then start the programmer from Tools → Programmer.

c. Click Hardware Setup and it should show as below



d. Click “Auto Detect” and add your .sof file from the master_image folder.



e. Fill in the checkbox for “Program” and select “Start” to begin downloading the design to the board – your dev kit will start running the design when the process is successful.

f. You should see LED4 blinking – this indicates that the board is reading to be remotely controlled.

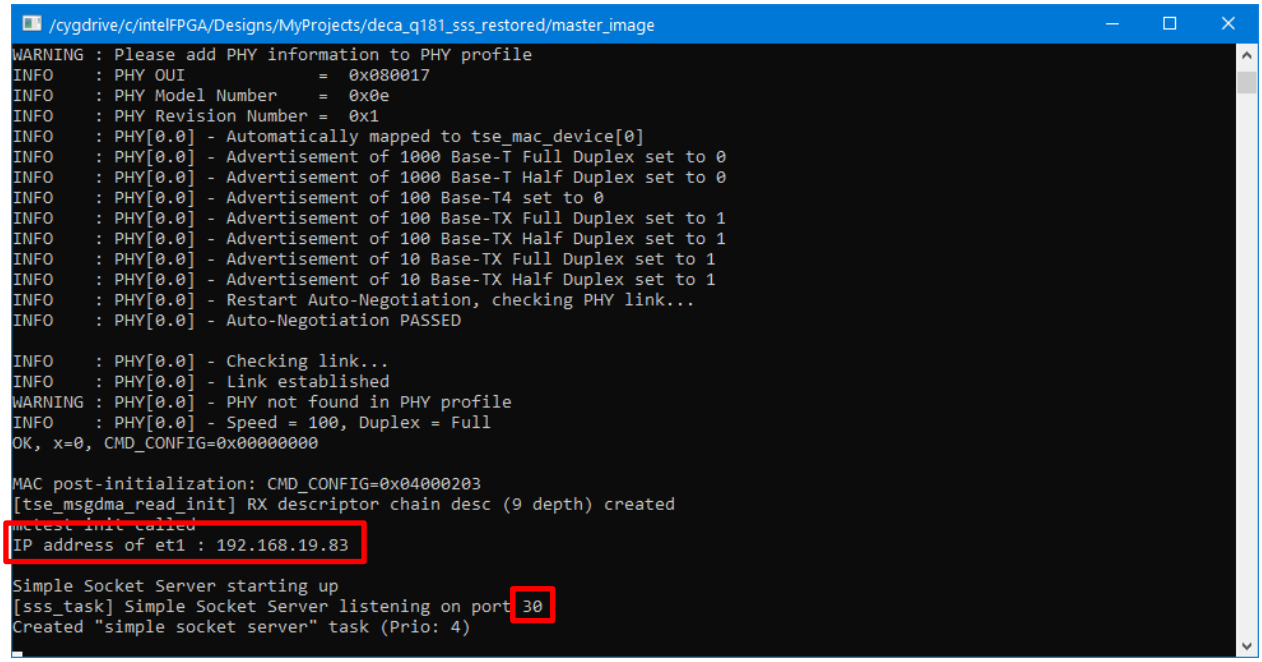
4. Open the Nios II terminal by going to Start → Altera → Nios II EDS → Nios II Command Shell

5. Enter the following command:

“nios2-download -g -r <file path>/master_image/deca_niosII_simple_socket_server.elf && nios2-terminal"

6. If you get a message prompting you to enter a 9 digit number. Enter a potentially correct MAC address like 255255248. This happens because some boards do not have a MAC address dedicated in the CFI Flash. The user has to enter the address if this is the case.

7. This will download the software onto the Nios II processor and run the software. The IP Address, Subnet Mash, and Gateway information will be set up and displayed. After it loads, the console should indicate that the Simple Socket Server is “listening on port <port number> as shown below.

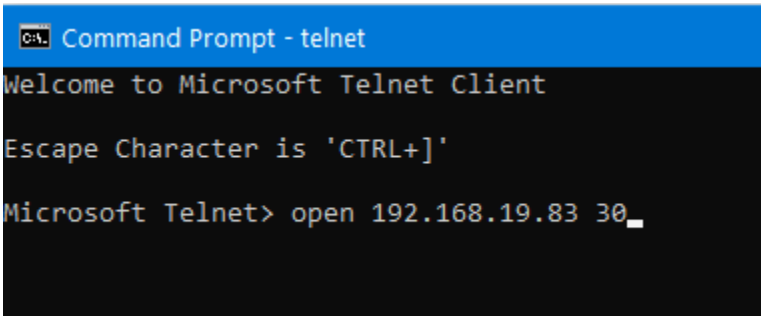


You may confirm the board for pinging by testing in a new command prompt

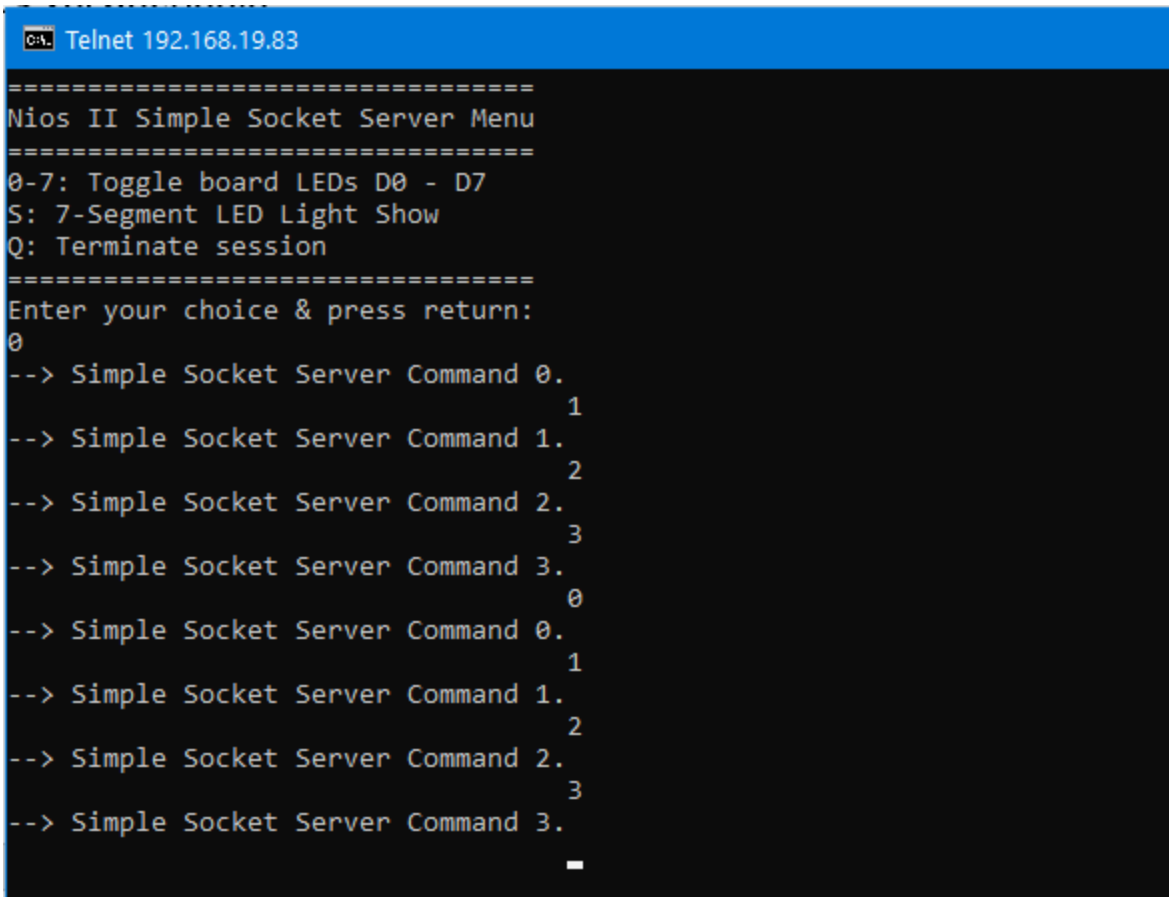
- Ping <your respective IP set by DHCP server or static IP if any>

8. Open up telnet client in the computer's command window by typing "telnet" and hitting enter.
9. Once you've entered the telnet client, open up the port by entering the command:

> *open <IP address shown on the Nios terminal> <port number>*



10. Once successfully connected, your screen should show the menu for controlling the board remotely. Entering a number between 0-3 will toggle LEDs 0-3 respectively.



How to compile the hardware

Follow the steps on the Design Store web page to extract and install the deca_q181_sss platform file. The following steps describe how to setup a project in Quartus II software in order to program the MAX10 FPGA DECA kit device with the Simple Socket Server design.

Note – extract platform

i) Launch Quartus II software and open the project top.qpf using File->Open Project.

ii) Compile the Project.

iii) Launch the Quartus II programmer.

iv) Download the .sof file output_files/top.sof and program the device using the programmer as previously described.

How to compile the software

The following steps describe how to use Nios II Software Build Tools for Eclipse software to perform the following tasks;

i) Create a software project (BSP and application) from a .sopcinfo file

ii) Add source code to the project

iii) Download and run source code on the target processor (NIOS II)

Note: Extract the m10sss18q1_deca_additionalfiles.zip to project directory after .qar extraction. It contains the following:

- Master_image Folder: having tested ready-to-run .sof & .elf to execute on the board connected. This is needs a 100Mbps Full duplex setting on the PC side for design to work. Auto-negotiation or any other Speed & Duplex settings on PC side will not work as the DECA on board Phy is 10/100Mbps only & design was defaulted to forcefully-negotiate to 100Mbps only.
- Software Folder: having the nios II software & bsp folders which can be imported to any Nios II SBT workspace.

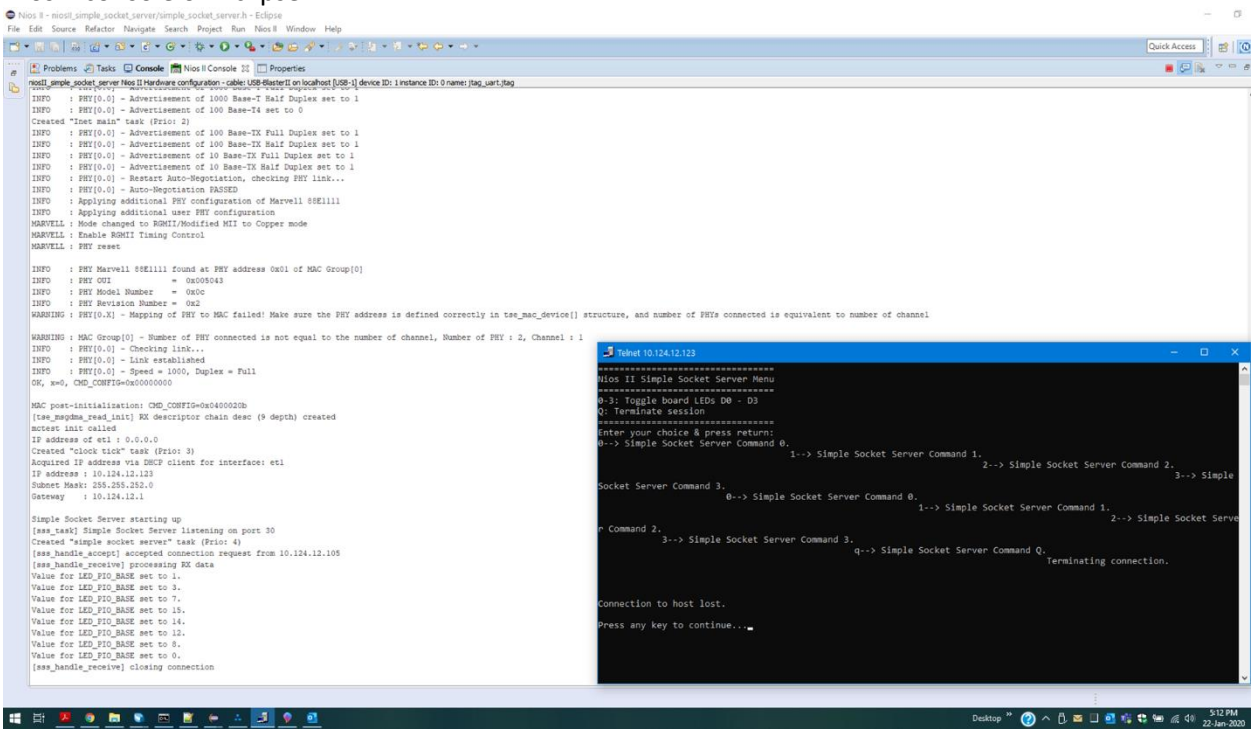
1. Open Quartus II on a windows platform
2. Launch Nios II Software Build Tools for Eclipse from the Tools menu
3. Specify the workspace directory for the project and click ok.
4. A blank workspace window pops up, right click anywhere in the Project Explorer and select the following to create a new software project; New->Nios II Application and BSP (board support package) from Template.
5. In the window that appears, browse to the location of the .sopcinfo file in the project folder and add it to the project. Provide a name for the software project i.e. deca_niosII_simple_socket_server and select "Simple Socket Server" for your template. Click Finish.
6. A BSP is created and located at deca_niosII_simple_socket_server and an application is created and located at deca_niosII_simple_socket_server_bsp.
7. The application contains source code from the template design "Simple Socket Server" that needs to be replaced with the source code located in the software/src folder. Select all the .c and .h files under the main project explorer tab. Right click and delete.
8. Right click the project name and left click the import command. We will be importing source code that has been modified for this dev kit.
9. Navigate to General → File System
    a. Browse to the location of src, click on the box next to it, add it and click Finish.
10. Import the contents of the src folder as shown below

Your project directory on the left should look similar (mem_init and the .elf will be created later)

11. Compile the project by selecting Build All from the Projects Menu or alternatively Ctrl+B.
12. Program the FPGA device by launching Quartus II programmer from the menu and loading the file output_files/top.sof.
13. Load the executable to the processor by right clicking on the Application project and selecting Run-As Nios II Hardware.
14. In the Run Configurations page under the Target connections tab. Check the options; Ignore mismatched system ID and Ignore mismatched system timestamp. If you don't see the connection specified, click Refresh Connections. Click Apply followed by Run.
15. If you get the following window, enter a 9 digit number MAC address that makes sense for the board. For example, if you type 123456789 it won't work as that is not a potential MAC address. But if you type in 255255248, it will work as that can be a potential MAC address.

This happens because on some boards there is no MAC address in the CFI flash dedicated section. In such cases, the software is designed to wait for the user to manually type in the MAC address.

16. Now your hardware image is downloaded to the FPGA through the Quartus programmer and the NIOS code has also been downloaded through Eclipse and will automatically start running. Refer to the Nios II console on Eclipse:



17. You can proceed to the demonstration as discussed in the earlier steps. The method described works well if you are modifying your code. You can maintain the same FPGA download and iterate through code revisions.

Additional Customization to play with:

<u>If DHCP feature is enabled:</u>

Application side – BSP folder “system.h” should have “#define DHCP_CLIENT” (in line 198) **uncommented** and IP addresses in simple_socket_server.h as **0.0.0.0**

Network side – Check if the DHCP server which assigns the IP address to the board is not having MAC filtering feature due to which it will fail to assign the IP inspite of DHCP enabled in your office network.

<u>If Static IP needed (no DHCP or i.e. one to one communication of board to PC):</u>

Application side – BSP folder “system.h” should have “#define DHCP_CLIENT” (in line 198) **commented** and IP addresses in simple_socket_server.h as anything like 192.168.1.10 or equivalent

Network side – Check if the Autonegotiation is getting passed but if it is not happening the board defaults to 100 Mbps full duplex, hence set your PC also to 100Mbps Full duplex mode and ping to the static IP of board it should work and subsequently the application should execute the same way.

Below is the log for static IP board to PC direct communication and execution of application, you can see the application is written to report the Static IP it is set for unlike it doesn't report for DHCP application execution as the highlighted place below. DHCP IP gets assigned and reported later in the application before SSS is started.

InterNiche Portable TCP/IP, v3.1

Copyright 1996-2008 by InterNiche Technologies. All rights reserved.
altera_eth_tse_init 0
prep_tse_mac 0
Your Ethernet MAC address is 00:07:ed:ff:e2:ac
Static IP Address is 192.168.19.83
prepped 1 interface, initializing...
tse_mac_init 0
List of PHY profiles supported (Total profiles = 5)...
Profile No. 0 :
PHY Name : Marvell 88E1111
PHY OUI : 0x005043
PHY Model Num. : 0x0c
PHY Rev. Num. : 0x02
Status Register : 0x11
Speed Bit : 14
Duplex Bit : 13
Link Bit : 10

Profile No. 1 :
PHY Name : Marvell Quad PHY 88E1145
PHY OUI : 0x005043
PHY Model Num. : 0x0d
PHY Rev. Num. : 0x02
Status Register : 0x11
Speed Bit : 14
Duplex Bit : 13
Link Bit : 10

Profile No. 2 :
PHY Name : National DP83865
PHY OUI : 0x080017
PHY Model Num. : 0x07
PHY Rev. Num. : 0x0a
Status Register : 0x11
Speed Bit : 3
Duplex Bit : 1
Link Bit : 2

Profile No. 3 :
PHY Name : National DP83848C
PHY OUI : 0x080017
PHY Model Num. : 0x09
PHY Rev. Num. : 0x00
Status Register : 0x00
Speed Bit : 0

Duplex Bit : 0
Link Bit : 0

Profile No. 4 :
PHY Name : Intel PEF7071
PHY OUI : 0x355969
PHY Model Num. : 0x00
PHY Rev. Num. : 0x01
Status Register : 0x00
Speed Bit : 0
Duplex Bit : 0
Link Bit : 0

INFO : TSE MAC 0 found at address 0x18003800
INFO : Multi Channel = No
INFO : MDIO Shared = No
INFO : MAC Type = 10/100/1000 Ethernet MAC
INFO : MAC Address = 0x18003800
INFO : MAC Device = tse_mac_device[0]
INFO : PHY Marvell 88E1111 found at PHY address 0x00 of MAC Group[0]
INFO : PHY OUI = 0x005043
INFO : PHY Model Number = 0x0c
INFO : PHY Revision Number = 0x2
INFO : PHY[0.0] - Automatically mapped to tse_mac_device[0]
INFO : PHY[0.0] - Advertisement of 1000 Base-T Full Duplex set to 1
INFO : PHY[0.0] - Advertisement of 1000 Base-T Half Duplex set to 1
INFO : PHY[0.0] - Advertisement of 100 Base-T4 set to 0
Created "Inet main" task (Prio: 2)
INFO : PHY[0.0] - Advertisement of 100 Base-TX Full Duplex set to 1
INFO : PHY[0.0] - Advertisement of 100 Base-TX Half Duplex set to 1
INFO : PHY[0.0] - Advertisement of 10 Base-TX Full Duplex set to 1
INFO : PHY[0.0] - Advertisement of 10 Base-TX Half Duplex set to 1
INFO : PHY[0.0] - Restart Auto-Negotiation, checking PHY link...
INFO : PHY[0.0] - Auto-Negotiation PASSED
INFO : Applying additional PHY configuration of Marvell 88E1111
INFO : Applying additional user PHY configuration
MARVELL : Mode changed to RGMII/Modified MII to Copper mode
MARVELL : Enable RGMII Timing Control
MARVELL : PHY reset

INFO : PHY Marvell 88E1111 found at PHY address 0x01 of MAC Group[0]
INFO : PHY OUI = 0x005043
INFO : PHY Model Number = 0x0c
INFO : PHY Revision Number = 0x2
WARNING : PHY[0.X] - Mapping of PHY to MAC failed! Make sure the PHY address is defined correctly in tse_mac_device[] structure, and number of PHYs connected is equivalent to number of channel

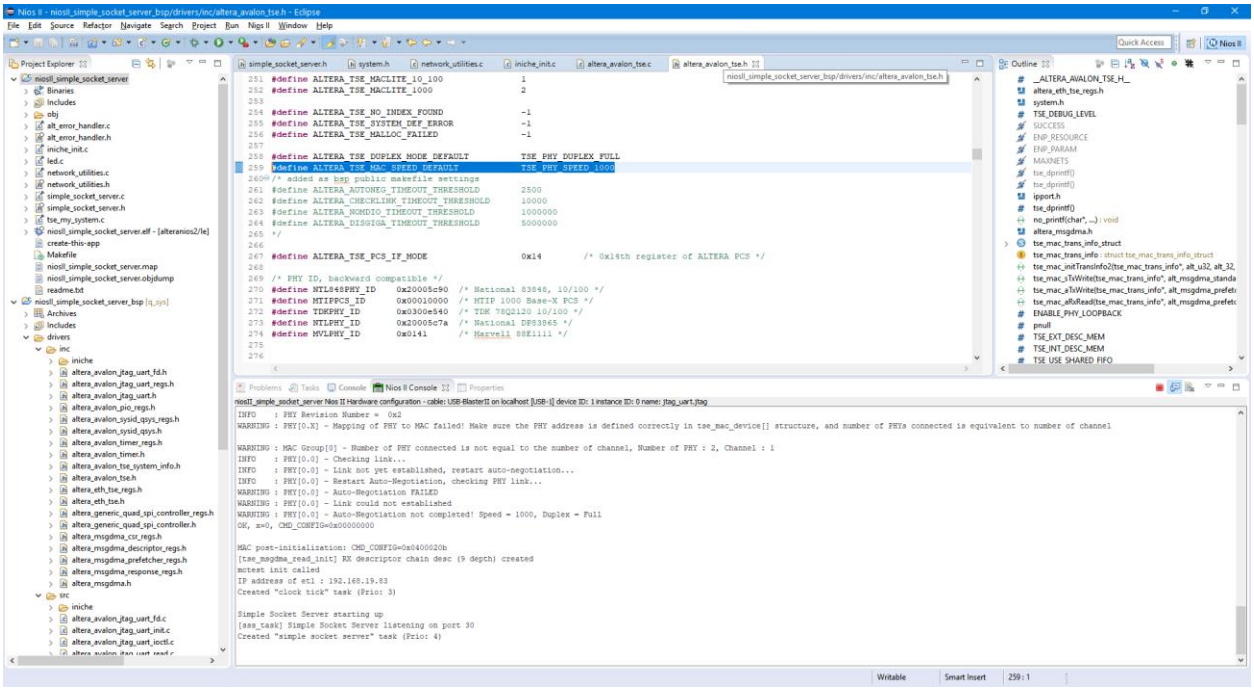
WARNING : MAC Group[0] - Number of PHY connected is not equal to the number of channel, Number of PHY : 2, Channel : 1
INFO : PHY[0.0] - Checking link...
INFO : PHY[0.0] - Link not yet established, restart auto-negotiation...
INFO : PHY[0.0] - Restart Auto-Negotiation, checking PHY link...
INFO : PHY[0.0] - Auto-Negotiation PASSED

INFO : PHY[0.0] - Link established
INFO : PHY[0.0] - Speed = 100, Duplex = Full
OK, x=39, CMD_CONFIG=0x00000000

MAC post-initialization: CMD_CONFIG=0x04000203
[tse_msgdma_read_init] RX descriptor chain desc (9 depth) created
mctest init called
IP address of et1 : 192.168.19.83
Created "clock tick" task (Prio: 3)

Simple Socket Server starting up
[sss_task] Simple Socket Server listening on port 30
Created "simple socket server" task (Prio: 4)
[sss_handle_accept] accepted connection request from 192.168.19.82
[sss_handle_receive] processing RX data
Value for LED_PIO_BASE set to 1.
Value for LED_PIO_BASE set to 3.
Value for LED_PIO_BASE set to 7.
Value for LED_PIO_BASE set to 15.
Value for LED_PIO_BASE set to 11.
Value for LED_PIO_BASE set to 9.
Value for LED_PIO_BASE set to 8.
Value for LED_PIO_BASE set to 0.
Value for LED_PIO_BASE set to 8.
Value for LED_PIO_BASE set to 12.
Value for LED_PIO_BASE set to 4.
Value for LED_PIO_BASE set to 6.
Value for LED_PIO_BASE set to 7.
[sss_handle_receive] closing connection

Static IP with 1Gbps link can be also established - by forcing the application to default itself to 1000Mbps as in #define statement below (originally it was set for 100Mbps – hence we had to force PC Network side to 100Mbps only).
This is done in the altera_avalaon_tse.h file as highlighted and also remember to set Speed & Duplex parameter on PC network side forced to 1Gbps Full duplex mode (not auto-negotiated or any other).



Author	Comments	Revision Date
Kasturi Rangan	• These are the working demo in Quartus Prime Std 18.1	2/5/2020