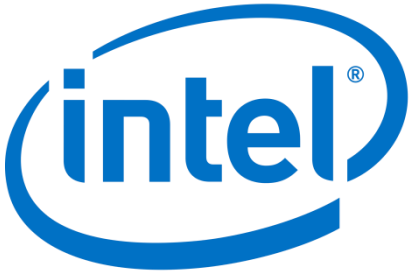

Remote System Upgrade with System Console for Cyclone 10 LP

Author: Tzy Way Ooi

Date: Aug 2018

Revision: 1.0

Contents

1 Introduction

This design example demonstrates the basic remote configuration features with EPCQ for Cyclone 10 LP device. A GUI interface is created with system console and allow user to upload new FPGA design. At the same time, you can also read remote update status, enable/disable watchdog timer and trigger reconfiguration from factory image to application image through the GUI interface in system console

2 Prerequisite

You need to have knowledge in instantiating and developing a system with Platform designer and familiar with the IPs used in this design example. Intel PSG recommends that you go through the online tutorials and training materials provided on Intel PSG's website before using this design example.

Related information:

- Altera Remote Update IP User Guide
- ASMI Parallel II Intel FPGA IP Core User Guide
- Quad-Serial Configuration (EPCQ) Devices Datasheet

3 Hardware and Software Requirements

3.1 Hardware Requirements

This design requires the following hardware:

- Cyclone 10 LP Development Kit
- Mini USB cable for programming the device

3.2 Software Requirements

This design requires the following software:

- Quartus Prime 18.0 Std
- Platform designer
- Any binary/hexadecimal file editor

4 Reference Design Installation Files

File or Directory Name	Description
master_image/output_file.jic	Quartus Prime programming file for EPCQ flash that consist of factory image, application 1 and application 2 image
factory_image/factory_hw.sof	Factory image that are used to generate programming file for EPCQ flash
application_image/application1/app1.sof application_image/application2/app2.sof	Application 1&2 image that are used to generate programming file for EPCQ flash
application_image/application1/app1.pof application_image/application2/app2.pof	Application 1&2 image that are used to generate raw programming file (RPD)

rpd/app1.rpd rpd/app2.rpd rpd/app_crc_error.rpd	Quartus Prime Raw Programming data files for application images that are used to upload design through system console

5 Block Diagram

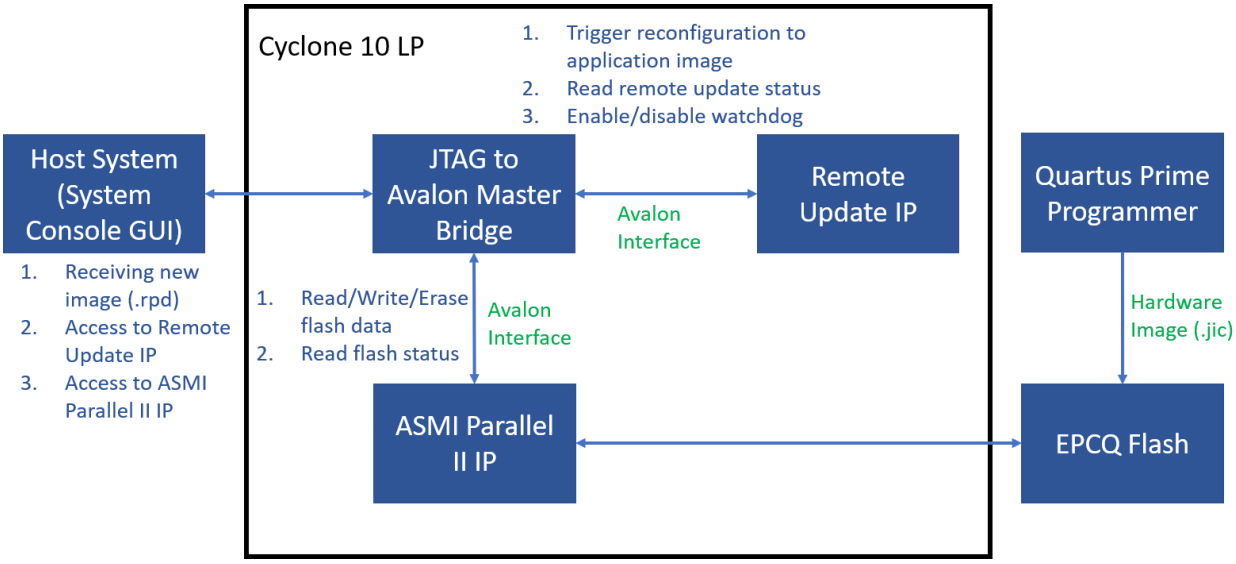

Figure 1: Design example block diagram

Figure 1 shows the various IP block within this remote system upgrade design example. You may see the detail connection between each IP blocks with the Qsys file included in this design example.

6 Generate Programming Files

This section will demonstrate on how to generate the programming files manually.

6.1 Generate JIC file

Perform the following steps to generate the JTAG indirect Configuration file (.jic) for the EPCQ flash

1. Open the **Convert Programming Files…** from **File** menu in Quartus Prime software.
2. Under **Output programming file** section, set the following items:
   a) **Programming file type:** JTAG Indirect Configuration File (.jic)
   b) **Configuration device:** EPCQ64 (select according to your EPCQ type)
   c) **Mode:** Active Serial
   d) **File name:** You may select your preferred path for the output file (.jic).
3. Check the checkbox for **Create config data RPD (Generate output_file_auto.rpd)**
4. Under **Input files to convert** section, perform the following steps:
   a. Select **Flash Loader**, then **Add Device** to select **10CL025Y** device in **Cyclone 10 LP** device family.
   b. For Factory image,
      i. Select **Page_0** of **SOF Data** and click **Properties** to open a popup window

ii. Under the **Address mode for selected pages** drop down menu, change **Auto** to **Start**
iii. Set **Start address** to **0x0** and click **OK** to close the popup window
iv. Click **Add File…** and open **factory_image/factory.sof**

c. For Application 1 image,
i. Click **Add Sof Page**
ii. Select **Page_1** of **SOF Data** and click **Properties** to open a popup window
iii. Under the **Address mode for selected pages** drop down menu, change **Auto** to **Start**
iv. Set **Start address** to **0x00400000** and click **OK** to close the popup window
v. Click **Add File…** and open **application_image/application_1/app1.sof**

d. For Application 2 image,
i. Click **Add Sof Page**
ii. Select **Page_2** of **SOF Data** and click **Properties** to open a popup window
iii. Under the **Address mode for selected pages** drop down menu, change **Auto** to **Start**
iv. Set **Start address** to **0x00600000** and click “OK
v. Click **Add File…** and open **application_image/application_2/app2.sof**

5. Click **Generate** to generate the JIC programming file

Note: You may choose compression on each SOF file during JIC file generation to compress each image. This is done by select the added SOF file and click **properties**. After that, check the **Compression** checkbox and click **OK**

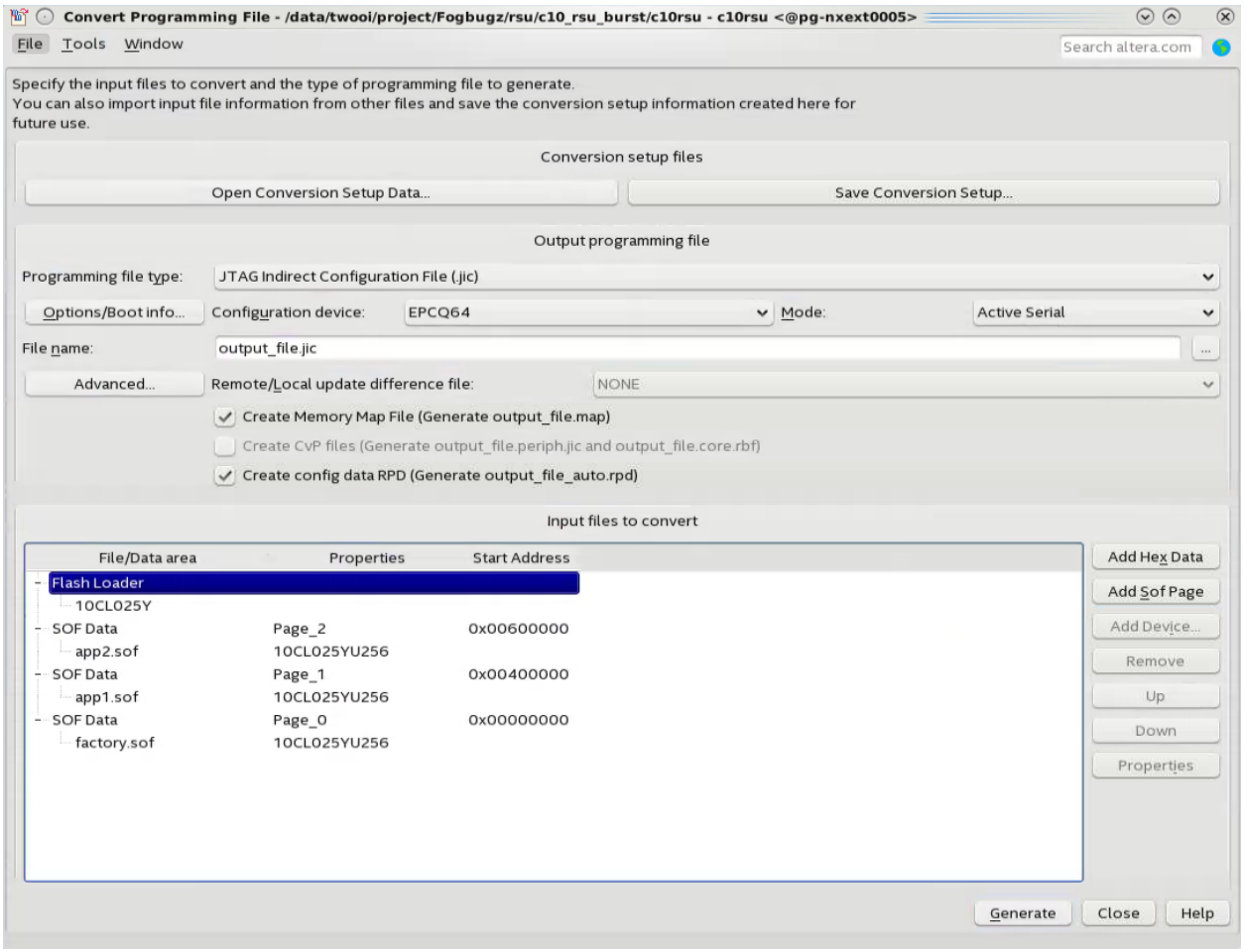

Figure 2: Convert Programming File window for JIC generation

6.2 Generate RPD file

Perform the following steps to generate the Raw Programming Data (RPD) for the new design to update the EPCQ flash

1. Obtain the POF file generated from the application design
2. Open the **Convert Programming Files…** from **File** menu in Quartus Prime software.
3. Under **Output programming file** section, set the following items:
    a. Programming file type: Raw Programming Data File (.rpd)
    b. File name: You may select your preferred path for the output file (.rpd)
4. Under **Input files to convert** section, perform the following steps:
    a. Select **Page_0** of **POF Data**
    b. Click **Add File…** and open **application_image/application_1/app1.pof**
5. Click **Generate** to generate the RPD programming file
6. Repeat the same steps 1-5 for application 2 image

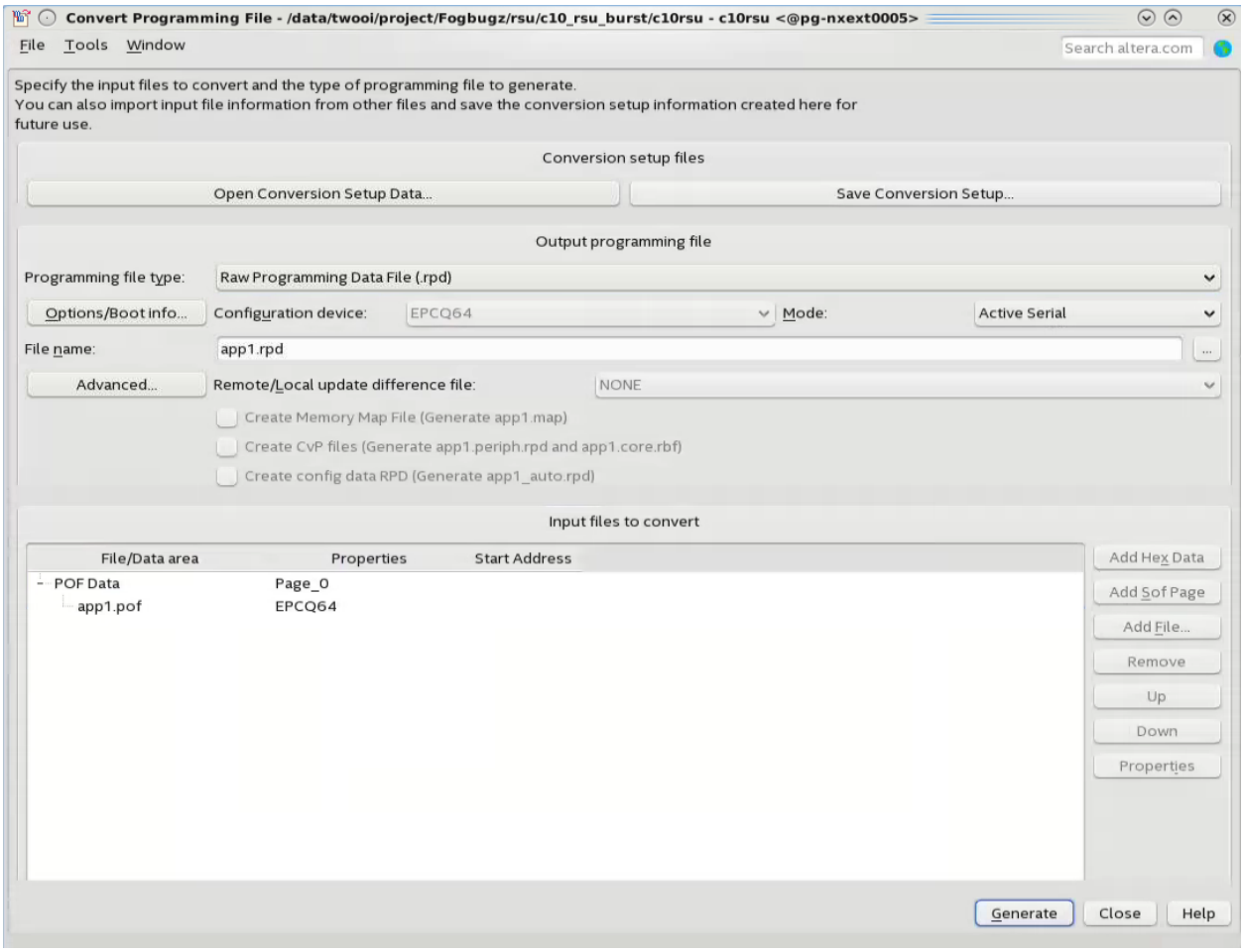

Figure 3: Convert Programming file window for RPD generation

Once you have generated the application 1 & 2 image RPD file, please take note that the generated RPD file size is having the full size of the EPCQ flash devices selected in your system. Hence, we will use binary/hexadecimal file editor to open each application RPD file and created the new RPD file with just data needed is included. The steps below are based on the “HxD-Hexeditor” where you can download it here.

1. Open the application 1 RPD file with HxD-Hexaeditor
2. Select the block data by going to **Edit-> Select block…** in Hxd-Hexeditor
3. For Application 1 image,
    a. Set **Start-offset** to **0** and set “End-offset” to **3FFFF**, then click **OK**

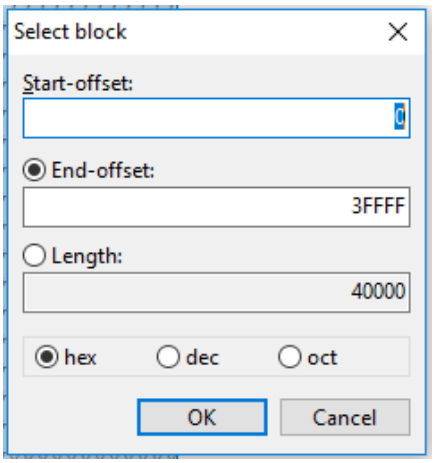

Figure 4: Select block window

b. The selected block will be highlighted and right-click on the selected block and click **Copy**
c. Create a new file by going to **File-> New**
d. Paste the copied content into the new file by going to **Edit -> Paste insert**
e. Save the new file in RPD format by going to **File -> Save as…**

4. Repeat the steps 1-3 for the RPD files for Application 2 image

Note: You may observe the RPD file data in HxD-Hexeditor to determine the End-offset for the RPD file. In the example above, you may observe that the data is 0xFF for the rest of the address after End-offset which mean those location do not have the valid data

7 Program the JIC file to the EPCQ flash

Perform the following steps to program the JIC file to the EPCQ flash

1. Open the Quartus Programmer software
2. Click **Auto Detect** and select **10CL025Y**
3. Right-click on the selected device **10CL025Y** and select **Change File**
4. Open the programming file **master_image/output_file.jic**
5. Check the **Program/Configure** checkbox for **10CL025Y** and **EPCQ64** devices
6. Click **Start** to start programming

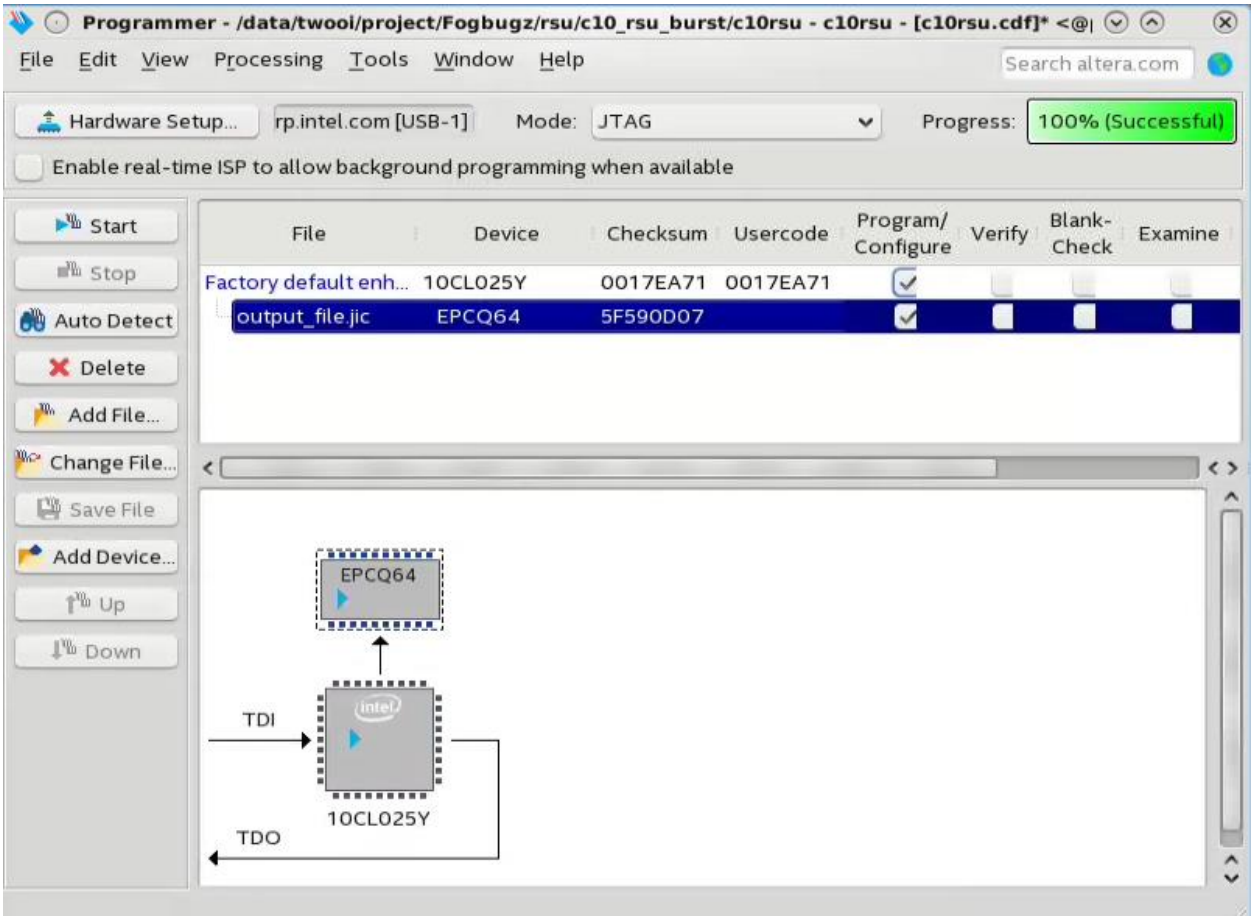

Figure 5: Quartus Programmer Window

8 Run the design

Once you have programmed the JIC file into the EPCQ flash, you may power cycle the board and the design will be loaded from flash and FPGA is configured. You will see LED0 is blinking which mean the factory image is loaded to the Cyclone 10 LP FPGA device

1. After power cycle the board, open system console by clicking **Tools**, select **System Debugging Tools** and select **System Console**
2. A TCL script is provided in the design example called **rsu_c10lp.tcl**. Launch the TCL script by using the command **source rsu_c10lp.tcl** in system console
3. You will able to see the GUI in the system console where you can access the Remote System Update IP and ASMI Parallel II IP

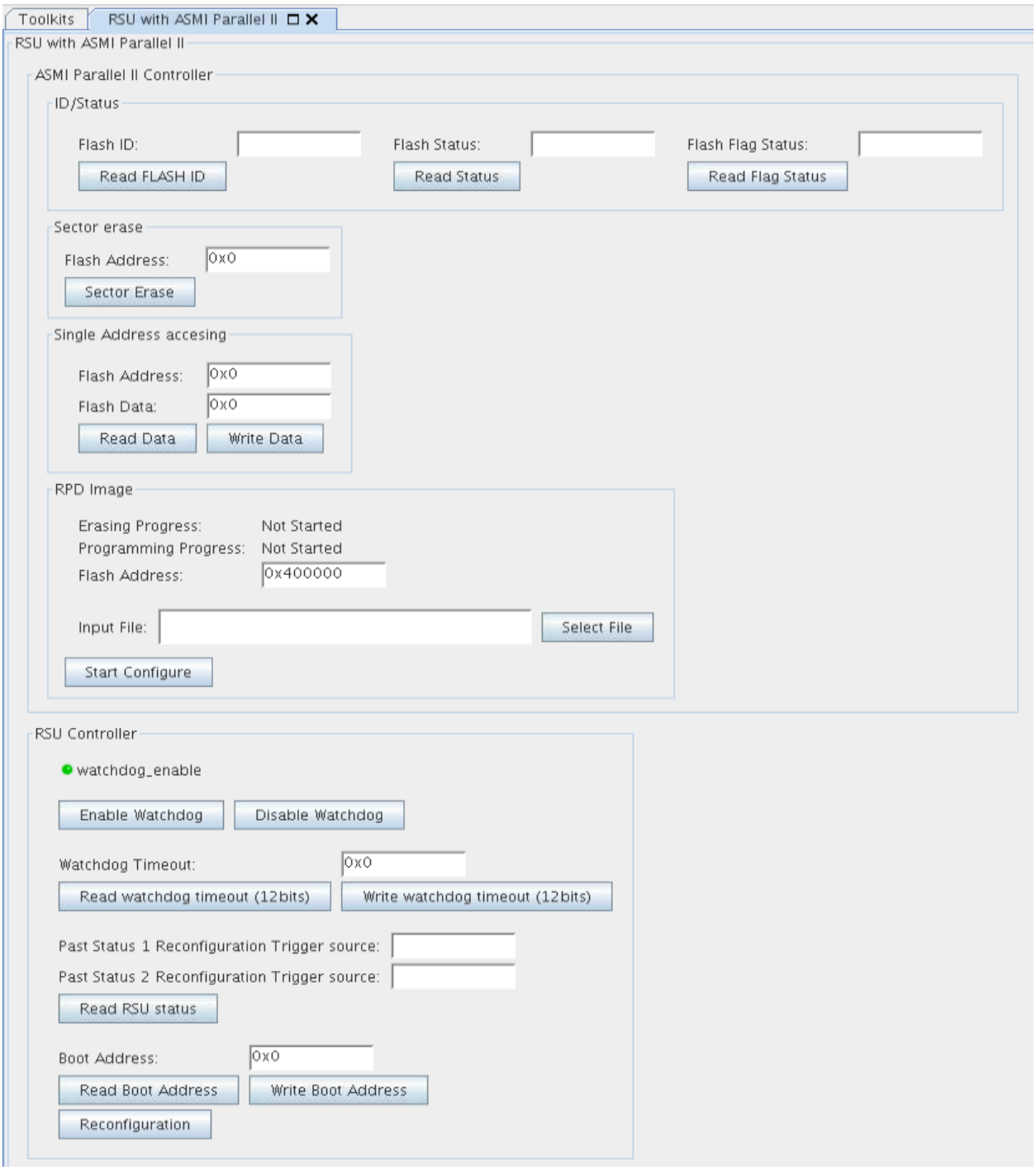

Figure 6: GUI Interface for accessing RSU and ASMI II IP

8.1 Perform remote system update without updating the EPCQ flash with RPD file

The application 1 & 2 image already programmed into the EPCQ flash with JIC programming. You may perform remote system upgrade to the system after power up the board by performing the following steps:
Under RSU controller section:

1. Disable the watchdog by selecting the **Disable Watchdog** button. Note: The Application image 1 & 2 did not reset the RSU watchdog timer and hence disable the watchdog timer to prevent watchdog timeout error
2. Write the boot address to the boot address column and select **Write Boot Address** button
    a. For Application image 1, the boot address is **0x400000**
    b. For Application image 2, the boot address is **0x600000**
3. You may read back the boot address written by selecting the **Read Boot Address** button to ensure the application boot address is written correctly
4. Trigger the reconfiguration by selecting the **Reconfiguration** button
    a. For Application image 1, you will see two LEDs which are LED1 and LED2 is blinking
    b. For Application image 2, you will see three LEDs which are LED1, LED2 and LED3 is blinking

Note: Once you have triggered reconfiguration to application configuration, you would need to go back to factory configuration to trigger reconfiguration to another application configuration

8.2 Perform remote system update with EPCQ flash is updated with new RPD file

The application 1 and 2 RPD file provided in this design example is the same as the design that has been programmed into the EPCQ flash with JIC file. To demonstrate the EPCQ flash can be updated with the RPD file section provided by the GUI in system console, we will program application 2 to the address 0x400000 while application 1 to the address 0x600000.
You may update the RPD file with the GUI provided in the system console. Note: Supported in Linux only
Under RPD image section:

1. Make sure that the board is in factory image configuration
2. Launch the TCL script with the command **source rsu_c10lp.tcl** in system console
3. In the RPD image section, click on **select file** to select the RPD File from
    a. For Application image 1 RPD file, it is in **rpd/app1.rpd**
    b. For Application image 2 RPD file, it is in **rpd/app2.rpd**
4. Write the flash address which is the address of RPD file will be written to flash in the **Flash Address** column
    a. For Application image 1 RPD file, the flash address is **0x600000**
    b. For Application image 2 RPD file, the flash address is **0x400000**
5. Click **Start Configure** and you will see the programming progress and **programming completed** indicates that the RPD file is written to the flash successfully

Once you have updated the flash with the RPD file, you can try trigger the reconfiguration.
Under RSU controller section:

1. Disable the watchdog by selecting the **Disable Watchdog** button. Note: The Application image 1 & 2 did not reset the RSU watch dog timer and hence disable the watchdog timer to prevent watchdog timeout error
2. Write the boot address to the boot address column and select **Write Boot Address** button
    a. For Application image 1, the boot address is **0x600000**
    b. For Application image 2, the boot address is **0x400000**
3. You may read back the boot address written by selecting the **Read Boot Address** button to ensure the application boot address is written correctly
4. Trigger the reconfiguration by selecting the **Reconfiguration** button
    a. For Application image 1, you will see two LEDs which are LED1 and LED2 is blinking

b. For Application image 2, you will see three LEDs which are LED1, LED2 and LED3 is blinking

5. Now, you will see that the address 0x400000 is having application 2 where LED1, LED2 and LED3 is blinking. On the other hand, the address 0x600000 is having application 1 where LED1 and LED2 is blinking

Note: Once you have triggered reconfiguration to application configuration, you would need to go back to factory configuration to trigger reconfiguration to another application configuration

8.3 Reconfiguration Trigger Status

The GUI interface provided contain the Past 1 Reconfiguration Trigger source and Past 2 Reconfiguration Trigger source. This tell you what caused the FPGA to configure the bitstream that is currently running. In Cyclone 10 LP, you can check back the two-past reconfiguration trigger source.
The table below show the representation for each bit. You may refer to Altera Remote Update IP Core User Guide for more details

Register bits	Description
Bit 4 – nconfig_source	External configuration reset (nConfig) assertion
Bit 3 – crcerror_source	CRC error during application configuration
Bit 2 – nstatus_source	nSTATUS asserted by an external device as the result of an error
Bit 1 – wdtimer_source	User watchdog timer timeout
Bit 0 – runconfig_source	Configuration reset triggered from logic array

In the section below, we are going to create some errors for the application image and read back the status of reconfiguration trigger source. Whenever each signal is triggered, the FPGA will back to Factory configuration.

8.3.1 Bit 4 – nconfig_source triggered

If you would like to see external configuration reset assertion, perform the following steps:

1. Make sure the FPGA is in Factory configuration
2. Disable watchdog by select the **Disable Watchdog** button in the GUI interface
3. Write boot address **0x400000** or **0x600000** to the boot address column and select **Write boot address** button to boot to either application 1 image or application 2 image.
4. Trigger reconfiguration by selecting "**Reconfiguration"** button
5. Close the system console
6. Now the FPGA is in Application configuration. Open Quartus programmer by clicking "**Tools**" and select **Programmer**
    a. Open programming file **factory_image/factory.sof** and select **Program/Configure**
    b. Click **Start** to program the SOF file into FPGA
7. Reopen the system console to launch the GUI interfaces
8. Click on the **Read RSU Status** and you will see the value **0x00000010** is in the past status reconfiguration trigger source column

8.3.2 Bit 3 – crcerror_source triggered

If you would like to see CRC error during application configuration, perform the following steps:

1. Make sure the FPGA is in Factory configuration
2. Disable watchdog by select the **Disable Watchdog** button in the GUI interface
3. Write the error RPD file into the flash with the GUI interfaces provided in system console
    a. Select the "**select file**" in RPD image section and choose the error RPD file in **rpd/app_crc_error.rpd**
    b. Write the flash address **0x500000** to the Flash Address column and click **Start Configure**
4. Once the error RPD file is written to the flash, write boot address **0x500000** to the boot address column and select **Write boot address** button
   Note: The address 0x500000 is having the CRC error application image
5. Trigger reconfiguration by selecting **Reconfiguration** button.
   Note: The FPGA is reconfigured with Factory image due to error in application image
6. Close the system console and reopen system console to launch the GUI interfaces
7. Click on the **Read RSU Status** and you will see the value **0x00000008** is in the past status reconfiguration trigger source column

8.3.3 Bit 2 – nstatus source triggered

If you would like to see nSTATUS asserted by an external device as the result of an error, perform the following steps:

1. Make sure the FPGA is in Factory configuration
2. Disable watchdog by select the **Disable Watchdog** button in the GUI interface
3. Write boot address **0x700000** to the boot address column and select **Write boot address** button
   Note: The address 0x700000 is empty (no application image)
4. Trigger reconfiguration by selecting **Reconfiguration** button.
   Note: The FPGA is reconfigured with Factory image due to error
5. Close the system console and reopen system console to launch the GUI interfaces
6. Click on the **Read RSU Status** and you will see the value **0x00000004** is in the past status reconfiguration trigger source column

8.3.4 Bit 1 – wdtimer_source triggered

If you would like to see user watchdog timer timeout error, perform the following steps:

1. Make sure the FPGA is in Factory configuration
2. Enable watchdog by selecting the **Enable Watchdog** button in the GUI interface
3. You may set the 12-upper bit watchdog timeout value through the GUI interface. We will not set any number in this case as we would like the timeout error happened as soon as possible
4. Write boot address **0x400000** or **0x600000** to the boot address column and select **Write boot address** button to boot to either application 1 image or application 2 image.
5. Trigger reconfiguration by selecting **Reconfiguration** button
   Note: The FPGA is reconfigured with Factory image due to watchdog timeout because the application image did not reset the watchdog timer
6. Close the system console and reopen system console to launch the GUI interfaces
7. Click on the **Read RSU Status** and you will see the value **0x2** is in the past status reconfiguration trigger source column

8.3.5 Bit 0 – runconfig_source triggered

If you would like to see configuration reset triggered from logic array, perform the following steps:

1. Make sure the FPGA is in Factory configuration
2. Disable watchdog by select the **Disable Watchdog** button in the GUI interface
3. Write boot address **0x400000** or **0x600000** to the boot address column and select **Write boot address** button to boot to either application 1 image or application 2 image.
4. Trigger reconfiguration by selecting **Reconfiguration** button

5. In either application 1 configuration or application 2 configuration, you may trigger reconfiguration by
    a. Close the GUI interfaces and reopen the GUI interfaces again in system console
    b. Trigger reconfiguration by selecting **Reconfiguration** button

    Note: When you trigger reconfiguration in application configuration, the FPGA is going to be reconfigured with the factory image
6. Close the system console and reopen system console to launch the GUI interfaces
7. Click on the **Read RSU Status** and you will see the value **0x00000001** is in the past status reconfiguration trigger source column

9 ASMI Parallel II IP access

In this design example, ASMI parallel II is used to write the RPD file to the EPCQ flash devices. The GUI interfaces in system console contain some of the functions where you can perform read/write/erase operation to the flash device

To perform read/write/erase operation, you may perform the following steps:

1. Open the GUI interfaces in system console by launching the TCL script with command **source rsu_c10lp.tcl** in system console
2. You may read the read the flash ID by clicking on the **Read FLASH ID** button
3. You may read the status register by clicking on the **Read Status** button
4. You may read the flag status register by clicking on the **Read Flag Status** button
5. To erase the flash, you can write each sector address to the **Flash Address** in the section Sector erase and select the **Sector Erase** button
6. To write data to the flash, you can write the address in **Flash Address** and data in **Flash Data** in the section Single Address accessing. Then, select the **Write Data** button
7. To read data from the flash, you can write the address in **Flash Address** in the section Single Address accessing. Then, select the **Read Data** button

9.1 Signal Tap for ASMI parallel II READ/Write Operation

This section is to show the Avalon Memory Map signal behavior when perform Write/Read operation to the ASMI parallel II

To execute the read/write operation to the control status register, you may follow the steps below for the AVMM signal behavior if you are creating your own control block to communicate with ASMI parallel II IP

1. Assert **avl_csr_write** or **avl_csr_read** signal while the **avl_csr_waitrequest** signal is low

    Note: If the **avl_csr_waitrequest** signal is high, the **avl_csr_write** or **avl_csr_read** signal must keep high until the **avl_csr_waitrequest** signal goes low
2. For read operation, set the address value on **avl_csr_address** bus

    For write operation, set the value data on the **avl_csr_writedata** bus together with the address on **avl_csr_address**
3. If it is read transaction, wait until the **avl_csr_readdatavalid** signal is asserted high to retrieve the data

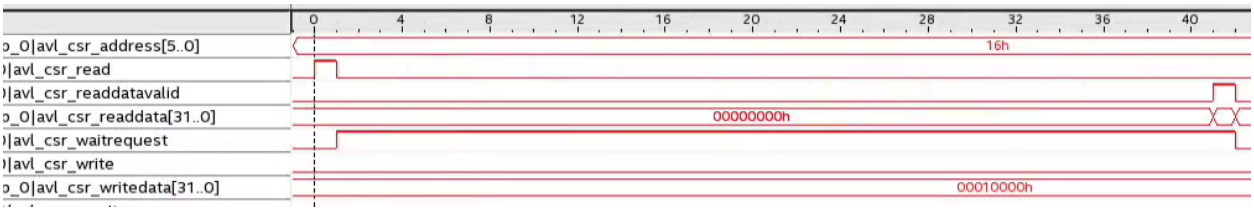

Figure 7: Perform read operation to CSR register for flash ID

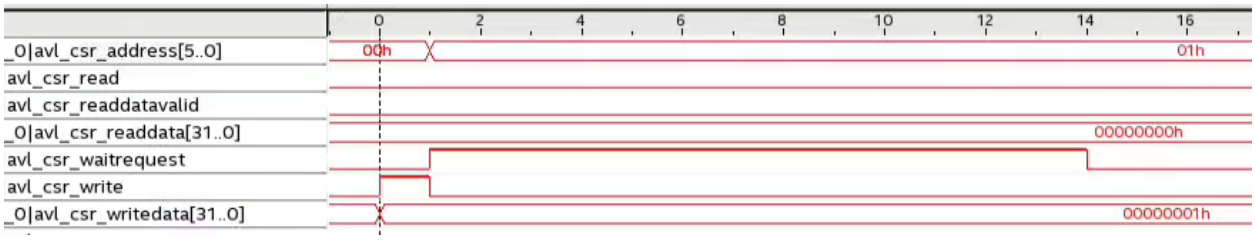

Figure 8: Perform write operation to CSR register for write enable

To execute the read/write operation to the memory interface, you may follow the steps below for the AVMM signal behavior

1. Assert the **avl_mem_write** or **avl_mem_read** signal while the **avl_mem_waitrequest** signal is low
2. For read operation, set the address value on **avl_mem_address** bus
   For write operation, set the value data on **avl_mem_writedata** bus together with the address on **avl_mem_address**
3. If it is read transaction, wait until the **avl_mem_readdatavalid** signal is asserted high to retrieve the data

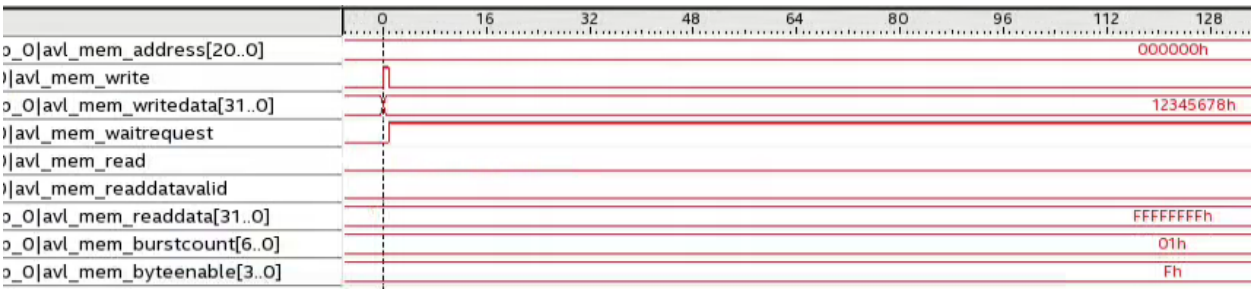

Figure 9: Write operation to memory interface

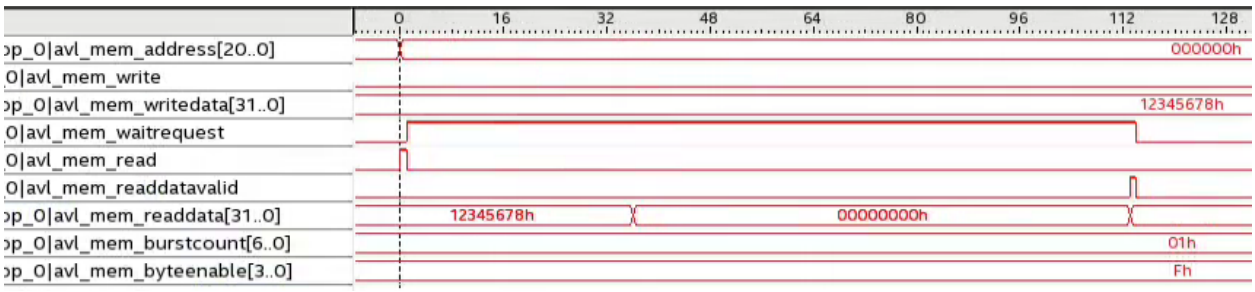

Figure 10: Read operation to memory interface

10 Revision History

Revision	Description
1.0	Remote system update for Cyclone 10 LP