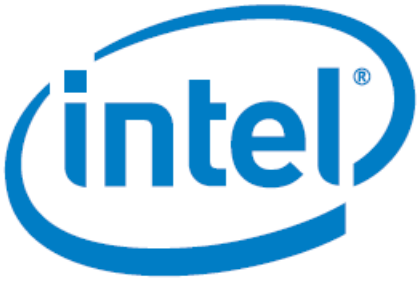


Cyclone 10 LP Remote System Upgrade Design Example User Guide

Date: 8/11/2017

Revision: 1.0

Table of Contents

Introduction

This design example demonstrates the ability of Cyclone 10 LP device booting between 2 configuration images by initiating Quartus build-in IP: Altera Remote Update IP. In the first chapter of this User Guide, the design example instructions will walk you through each of the steps to generate this design and eventually loading the 2 configuration images (1 factory image and 1 application image) on your FPGA.

Requirements

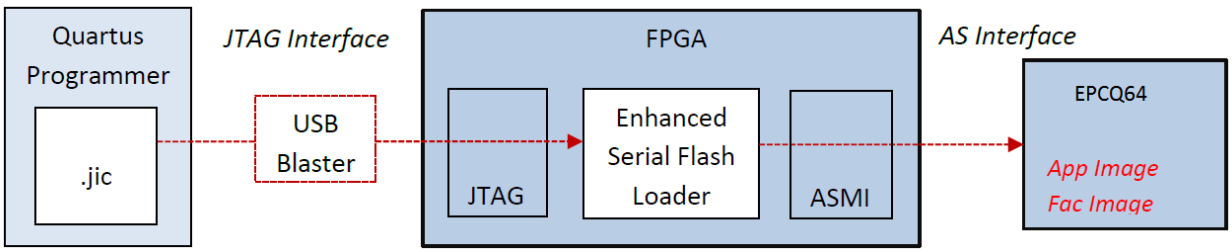
You will need the following hardware and software to implement this design example:

Quartus Version	Quartus® 17.0 Build 595 Standard Edition
IPs	Altera Remote Update, ALTPLL
Hardware	Cyclone 10 LP FPGA Evaluation Kit
Device number	10CL025YU256I7G
Configuration Device	EPCQ64
Configuration Scheme (Mode)	Active Serial

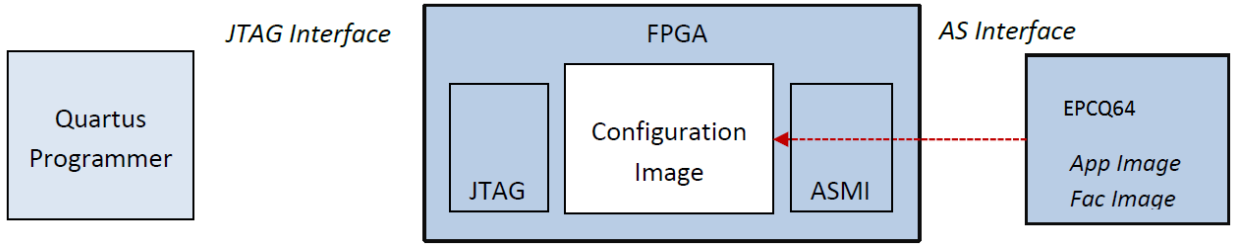
NOTE: A complete RSU design should include at a minimum 2 project files: 1 factory image and 1 application image. These two projects should be compiled separately to generate their own .sof file. However in this particular design example, for ease of use, we are using one project and two top level source files (*factory_image.v* and *application_image.v*) which we manually switch to the top entity. For your own design, it is recommended to always have two separate design projects for each image.

Theory of Operation

Step 1: Program Your Serial Configuration Device (EPCQ64) with the FPGA Configuration Image



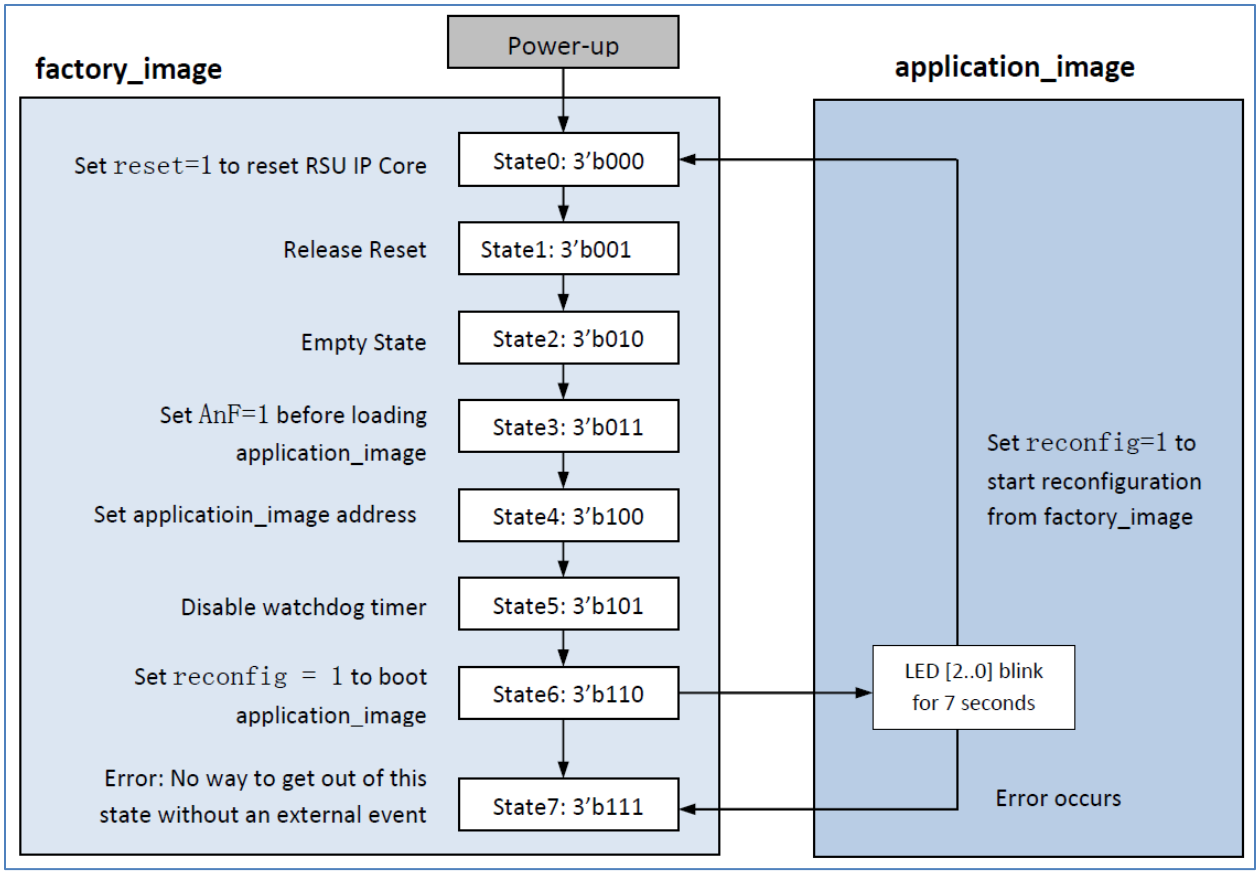
Step 2: Configure Your FPGA from Your Serial Configuration Device (EPCQ64)



The two images are first generated as SRAM Object Files (.sof) files after compilation and then combined into 1 single Jtag Indirect File (.jic). This .jic file is then loaded to an external flash (EPCQ64 in this case) through the Enhanced Serial Flash Loader (SFL) generated by Quartus Programmer as shown in the above block diagram. In this example, half of the memory resource is divided into half and each store one image. The factory image is Image 0 and the application image is Image 1. On the next power cycle after the 2 images are loaded into flash, the Cyclone 10 LP FPGA will start being configured by images loaded from the EPCQ64.

In the factory image, you will see incremental counter at LED 2:0 and LED3 show heartbeat. There is a 7-step state machine: State 0 (Reset the remote system update circuitry) -> State 1 (Release the reset) -> State 2 (Empty State) -> State 3 (Set AnF=1) -> State 4 (Set start address for application image) -> State 5 (Disable the watchdog timer) -> State 6 (Boot the application Image) -> State 7 (Hold if error occurs). LED [2…0] is set to be a counter indicating which state the factory image jumps into. LED[3] is set to be a much faster blinking light as a “heartbeat”. Once the factory image enters State 6, *reconfig* is set to 1’b1 which triggers the state machine enters the application image.

In the application image, LED [2…0] will keep blinking simultaneously for 7 seconds and then jump back to State 0 in the factory image. The only condition entering State 7 is when an error occurs. Once it enters State 7, there is no way out of this state without an external even although the “heartbeat” LED [3] still blinks indicating the factory image is still working.



How to Setup the Hardware for Link Test

Follow these steps to setup the hardware to run the reference design:

1. Connect the USB cable coming together with the Cyclone 10 LP FPGA Evaluation Kit between the PC and the kit as shown in Figure 1
2. The USB cable will supply the power as well as establish the JTAG connection with the kit
3. To enable RSU feature, make sure SW1.4 set to 1 (move to left side from the view of Figure 1). For more details about default switching settings, please refer to Cyclone 10 LP FPGA Evaluation Kit user guide
4. The hardware system is now ready for programming.

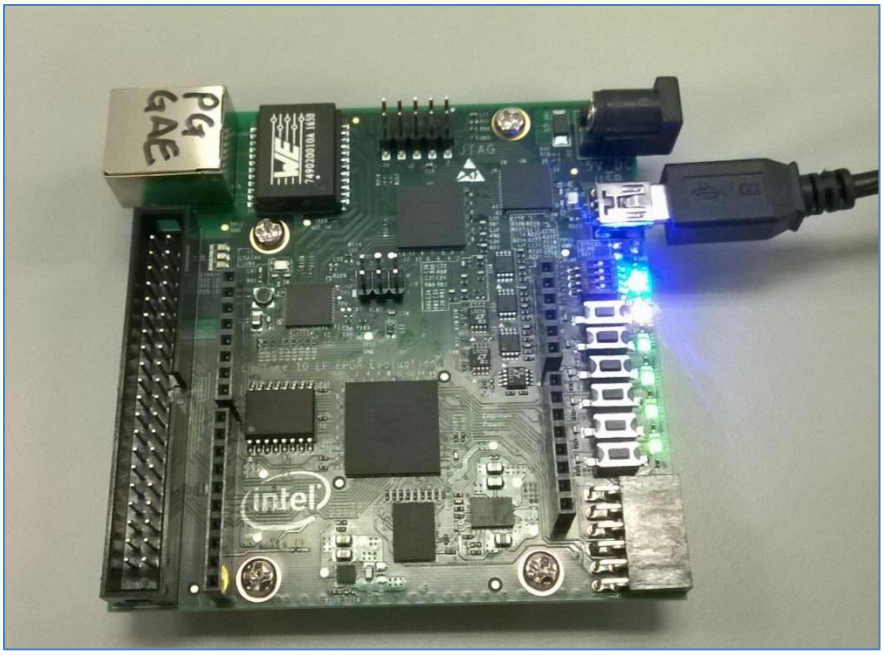


Figure 1. Cyclone 10 LP FPGA Evaluation Kit

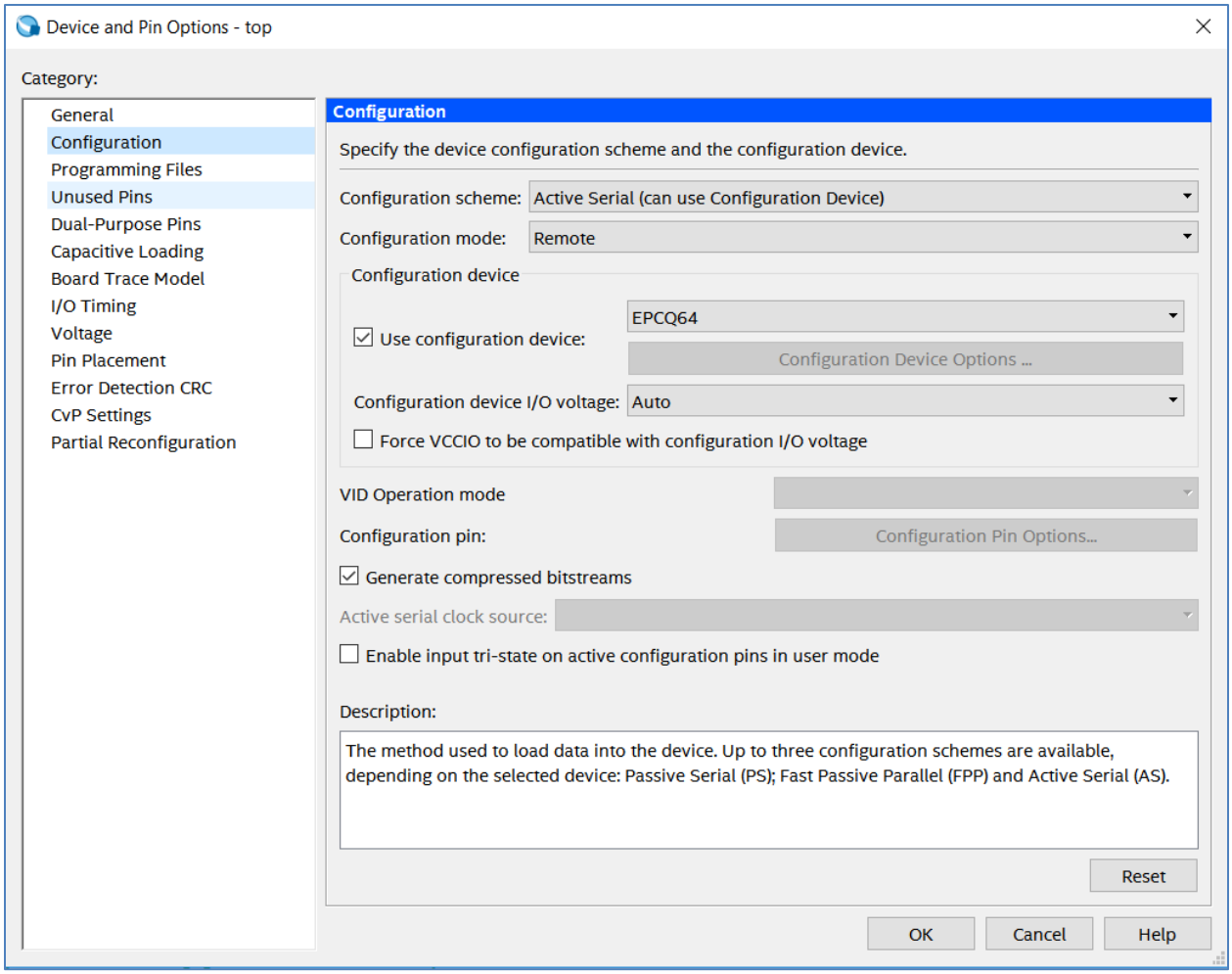
Step.1 Download .par file from design store

Download the .par file from design store and save them to your local <destination> folder. (Note: Please rename your local folder <destination> of your own)

Step.2 Generate design project and compile the Factory Image

Open C10LP_RSU.par in Quartus

I. Follow the guideline on Design Store to load your design as a template.
II. You can find a more detailed guideline here.



Configuration settings in Device and Pin Options

In the upper left Project navigator window, double click the Compilation Hierarchy title “Cyclone 10 LP: 10CL025YU256I7G” and In the Device panel, click Device and Pin Options.

- In the Device and Pin Options panel, select the correct settings.

Configuration Scheme – Active Serial

Configuration Mode – Remote

Configuration Device – EPCQ64

Compile and rename the SRAM Object File

a. Generate design project and compile factory image.

- Replace the Top-level Entity with factory_image.v
  - ✓ In Project navigator window, under “Files”, right click “source/factory_image.v” and and set as Top-Level Entity

- Click “Start Compilation”.
- After the design is compiled successfully, go to <destination>/output_files, there should be a SRAM Object File top.sof generated in the folder. Change the name to factory_image.sof.

b. Generate design project and compile Application Image
   - Replace the Top-level Entity with application_image.v
     - ✓ In Project navigator window, under “Files”, double click “Files”.
     - ✓ In the Settings window, delete factory_image.v and add application_image.v (<destination>/source/application_image.v).
     - ✓ After application_image.v is added in Project Navigator, right click “source/ application_image.v” and set as Top-Level Entity.
   - Compile and rename the SRAM Object File to application_image.sof
     - ✓ Click “Start Compilation”
     - ✓ After the design is compiled successfully, go to <destination>/output_files, change the name of top.sof to application_image.sof

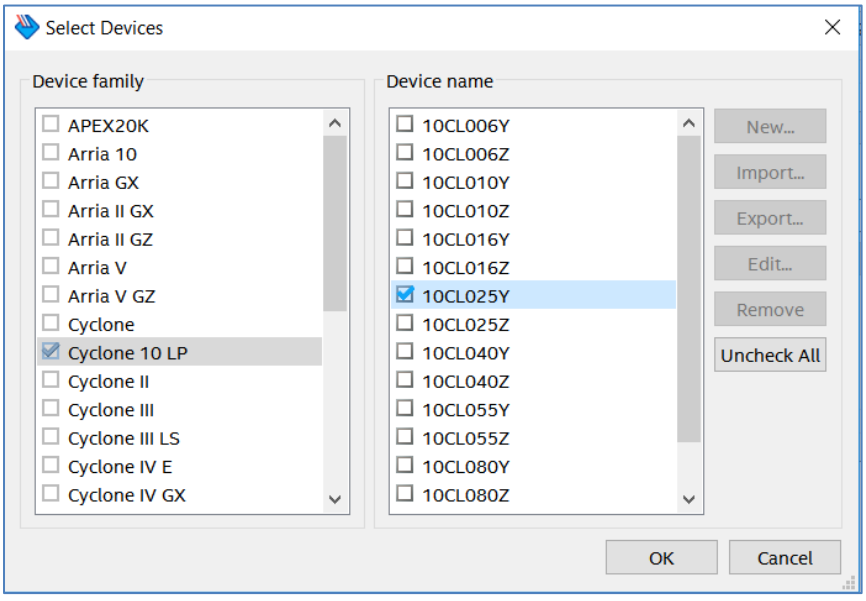
Step.3 Creating JTAG Indirect Configuration File (.jic)

a. Open Convert Programming File GUI
   - In Quartus, click File > Convert Programming Files.

b. Select Programming Type
   - Select “JTAG Indirect Configuration File (.jic)” as programming file type.

c. Select Configuration Device and Mode
   - Select “EPCQ64” as Configuration Device and select “Active Serial” mode.
   - Locate your .jic file under <destination> folder and rename it if necessary.

d. Add Flash Loader
   - Under Flash Loader section, select “Flash Loader” and click “Add device” on the right.
   - In this example, we chose “10CL025Y”.



e. Add Page 0: factory_image.sof
- Select “SOF Data (Page_0)” and click “Add File”.
- Locate to <destination> folder where restores factory_image.sof generated in Step.2.
- There should be “factory_image.sof (10CL025YU256)” added under “SOF Data (Page_0)”.
- Select “SOF Data (Page_0)” and click “Properties”.
- In the new window, select “Block” under Address mode for selected pages.
- Enter “0x00000000” as start address (corresponding to the start address of sector 0).
- Enter “0x003FFFFF” as end address (corresponding to the end address of sector 63).
- Click OK to close “SOF Data Properties” window.
- Select “factory_image.sof (10CL025YU256)” and click “Properties”.
- Check “Compression” box on the top left corner and click OK.

f. Add Page 1: application_image.sof
- Click “Add Sof Page” to create a new page.
- Select “SOF Data (Page_1)” and click “Add File”.
- Locate to <destination> folder where restores application_image.sof generated in Step.3.
- There should be “application_image.sof (10CL025YU256)” added under “SOF Data (Page_0)”.
- Select “SOF Data (Page_1)” and click “Properties”.
- In the new window, select “Block” under Address mode for selected pages.
- Enter “0x00400000” as start address (corresponding to the start address of sector 64).
- Enter “0x007FFFFF” as end address (corresponding to the end address of sector 127).
- Click OK to close “SOF Data Properties” window.
- Select “application_image.sof (10CL025YU256)” and click “Properties”.
- Check “Compression” box on the top left corner and click OK.

g. Generate .jic file and check Memory Map File (.map)
- Click “Generate” close the Converting Programming Files GUI window.
- Now in your <destination> folder, you can find your .map file along with your .jic file.
- Open the .map file and it should look like this:

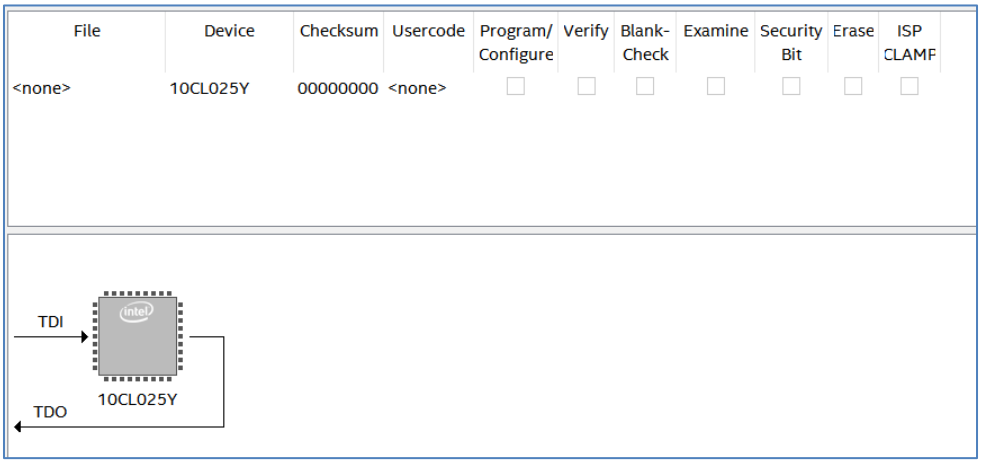
```
BLOCK        START ADDRESS        END ADDRESS

Page_0       0x00000000       0x003FFFFF (0x000328C8)
Page_1       0x00400000       0x007FFFFF (0x00432485)


Configuration device: 10CL025Y
Configuration mode: Active Serial
Quad-Serial configuration device dummy clock cycle: 8
```

TIPS: You can always open .map file to check whether your programming files are converted correctly.

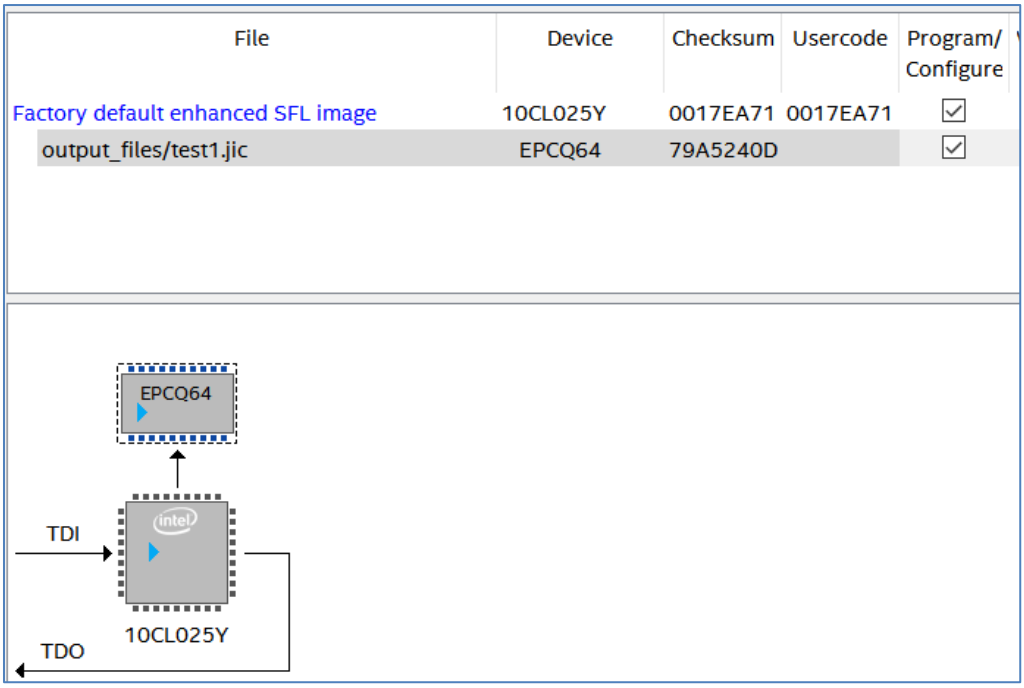
Step.4 Program FPGA

a. Auto Detect JTAG Chain

- In Quartus, click Tools > Programmer.
- Click “Hardware Setup” on the top left corner and select “USB-BlasterII[USB-1]” and close
- Click “Auto Detect” and choose the corresponding device number. In this example, please select “10CL025Y”. A JTAG chain should be generated successfully shown as below.



- Select “ <none> 10CL025Y 00000000 <none>” and click “Change File”
- Locate the .jic file you generated in Step.4.
- Then you can see it changed to configuration as below. Tick the Program/Configure box.



- Click Start to program.

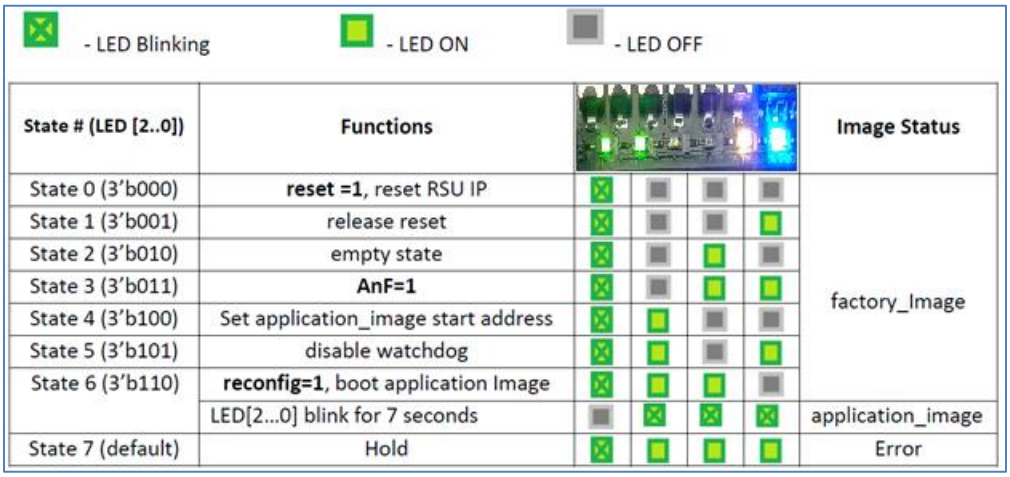
NOTE: the “Progress” Bar on the top right indicates the percentage of your configuration status. If you are careful enough, you will notice the bar will go from 0% to 100% very quickly in about 5 seconds and then goes back to 0% and start a much slower 2nd stage of process which takes around 60 seconds. The first stage indicates the Enhanced Serial Flash Loader (SFL) inside your targeted FPGA (In this example: Cyclone 10 LP) is loaded successfully. The 2nd slower stage you observe indicates the .jic file loading from the Quartus Programmer to your Flash Device (In this example: EPCQ64) through the Enhanced SFL which was just loaded successfully in the 1st Stage.

- After progress bar shows “100% (Successfully)”, power cycle the board.

- After a power cycle, the factory image in the EPCQ64 will be loaded in FPGA.

b. LED Observations
   By powering up the board, you should able be see the 4 LEDs blinking:
   - LED[2…0] are counters indicating the current stage.
   - LED[3] is the “heartbeat” which will blink until state machine enters application image.
   - When only LED[3] is blinking and LED [2…0] are all off, this means it enters Error Stage.

NOTE: The state machine will start from factory image then enter application image. After it enters the application image, LED[2…0] will blink simultaneously for around 7 seconds indicating the application image is loaded successfully after which it will revert to the factory image, starting the cycle all over again. A successful LED observation will be jumping back and forth between factory image and application image. Please see the diagram below.



State # (LED [2..0])	Functions	Image Status
State 0 (3’b000)	**reset =1**, reset RSU IP	factory_Image
State 1 (3’b001)	release reset	
State 2 (3’b010)	empty state	
State 3 (3’b011)	**AnF=1**	
State 4 (3’b100)	Set application_image start address	
State 5 (3’b101)	disable watchdog	
State 6 (3’b110)	**reconfig=1**, boot application Image	
	LED[2…0] blink for 7 seconds	application_image
State 7 (default)	Hold	Error

References

Quad-Serial Configuration (EPCQ) Device Datasheet

https://www.altera.com/en_US/pdfs/literature/hb/cfg/cfg_cf52012.pdf

Altera Remote Update IP Core User Guide

https://www.altera.com/en_US/pdfs/literature/ug/ug_altremote.pdf

Revision History

Date	Version	Changes
August 11, 2017	1.0	Initial Release