Remote System Upgrade Design Example

User Guide

Version	Date	Author	Comments
v1.1	4/26/2016	Mark Zhong	All V Series Devices
V1.2	5/22/16	Mark Zhong	Update Block Diagram, State Machine, Add “Device and Pin Options” step after Step2/b
V1.3	6/13	Mark Zhong	Change Chpter2/1 and 2/a: delete updates IP. change from .qar to .par file

Table of Contents

Introduction

Design Example Instructions

Appendix

Introduction

This design example demonstrates the ability of V series device booting between 2 configuration images by initiating Quartus build-in IP: ALTREMOTE_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.

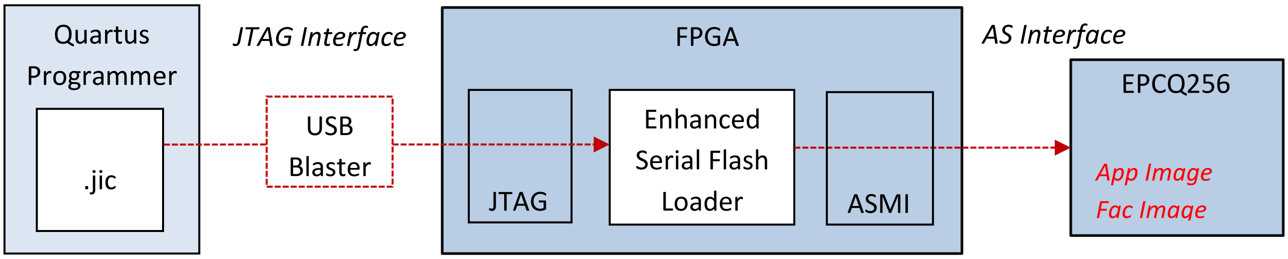
Prerequisites

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

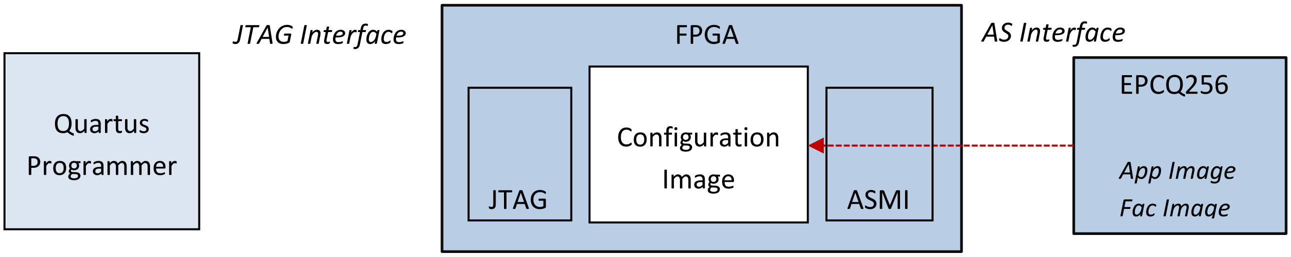
Quartus Version	15.1.0 Build 185 Standard Edition
IPs	ALTREMOTE_UPDATE, Altera PLL
Hardware	Cyclone V SoC Development Kit
Device Number	5CSXFC6D6F31C8ES
Configuration Device	EPCQ256
Configuration Scheme (Mode)	Active Serial (x4)
Accessories	19 V power cord, USB Blaster II

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.

Step 1: Program Your Serial Configuration Device (EPCQ256) with the FPGA Configuration Image



Step 2: Configure Your FPGA from Your Serial Configuration Device (EPCQ256)

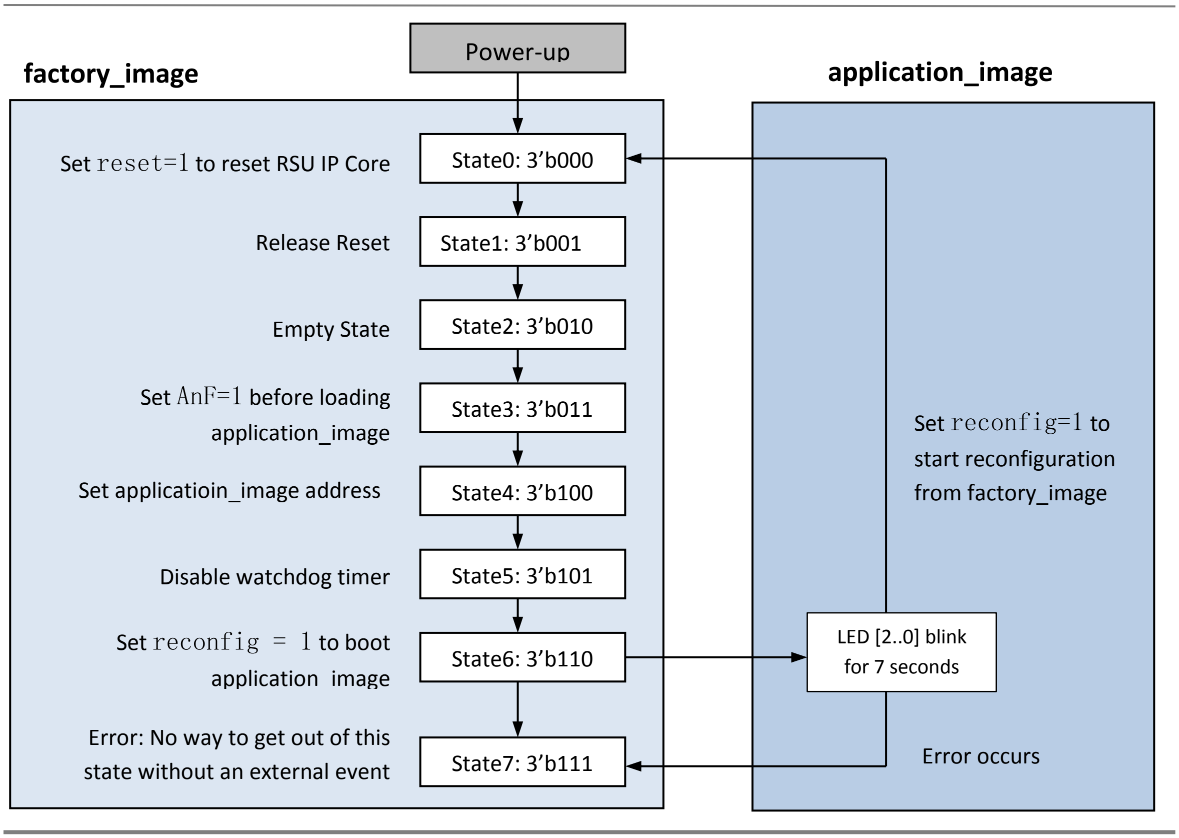


Theory of Operation

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 (EPCQ256 in this case) through the Enhanced Serial Flash Loader (SFL) generated by Quartus Programmer as shown in the above block diagram. In this particular example, half of the memory resource is divided into half and each stores 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 V FPGA will start being configured by images loaded from the EPCQ256.

In the factory image, 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'b0 which triggers the state machine enters the application image.

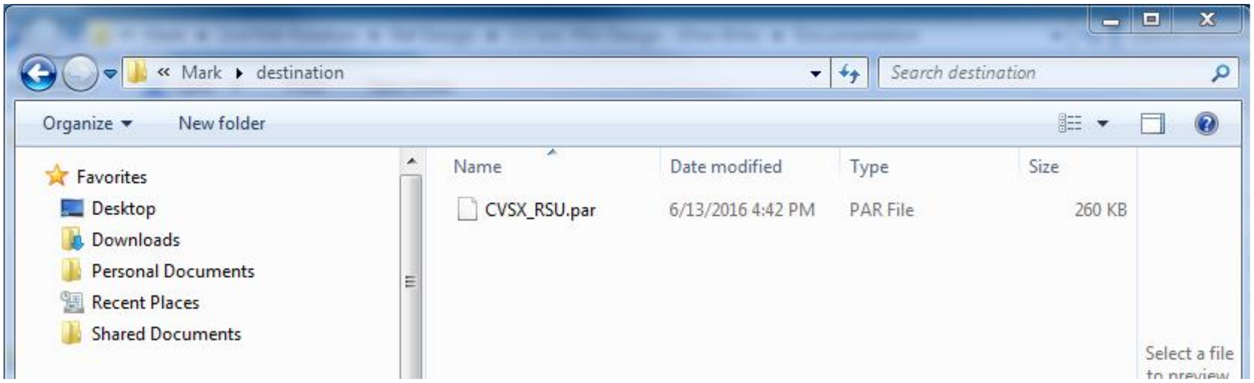
In the application image, LED [2…0] will keep blinking simultaneously for 7 seconds and then jump back to State 0 in the factory image. The only condition entering State 7 is when an error occurs. Once it enters State 7, there is no way out of this state without an external even although the "heartbeat" LED [3] still blinks indicating the factory image is still working. If you want to migrate this design to other V series devices, please make sure device number and pin assignments are modified correctly.


Power-up
factory_image
application_image
Set reset=1 to reset RSU IP Core
State0: 3'b000
Release Reset
State1: 3'b001
Empty State
State2: 3'b010
Set AnF=1 before loading application_image
State3: 3'b011
Set applicatioin_image address
State4: 3'b100
Disable watchdog timer
State5: 3'b101
Set reconfig = 1 to boot application image
State6: 3'b110
Error: No way to get out of this state without an external event
State7: 3'b111
Set reconfig=1 to start reconfiguration from factory_image
LED [2..0] blink for 7 seconds
Error occurs

Design Example Instructions

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)

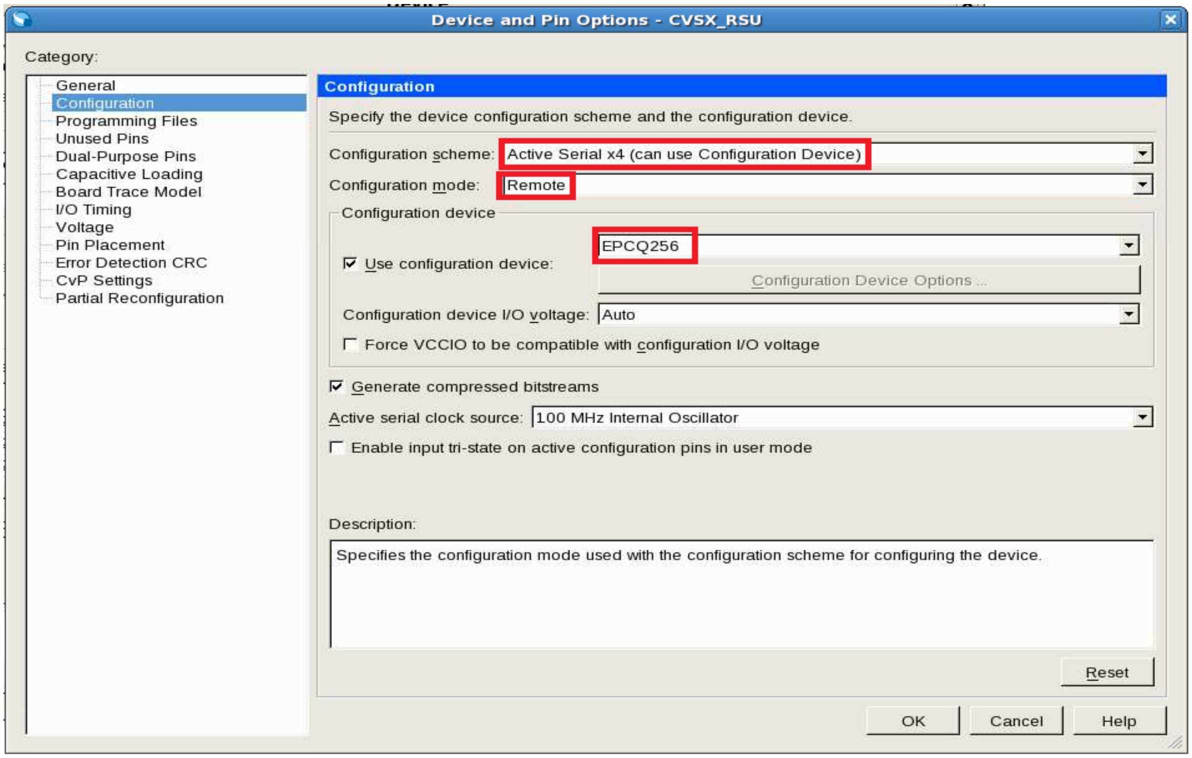


Name	Date modified	Type	Size
CVSX_RSU.par	6/13/2016 4:42 PM	PAR File	260 KB

Step.2 Generate design project and compile the Factory Image

a. Open CVSX_RSU .par in Quartus

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



Configuration

Specify the device configuration scheme and the configuration device.

Configuration scheme: Active Serial x4 (can use Configuration Device)

Configuration mode: Remote

Configuration device

☑ Use configuration device: EPCQ256

Configuration device I/O voltage: Auto

☐ Force VCCIO to be compatible with configuration I/O voltage

☑ Generate compressed bitstreams

Active serial clock source: 100 MHz Internal Oscillator

☐ Enable input tri-state on active configuration pins in user mode

Description:

Specifies the configuration mode used with the configuration scheme for configuring the device.

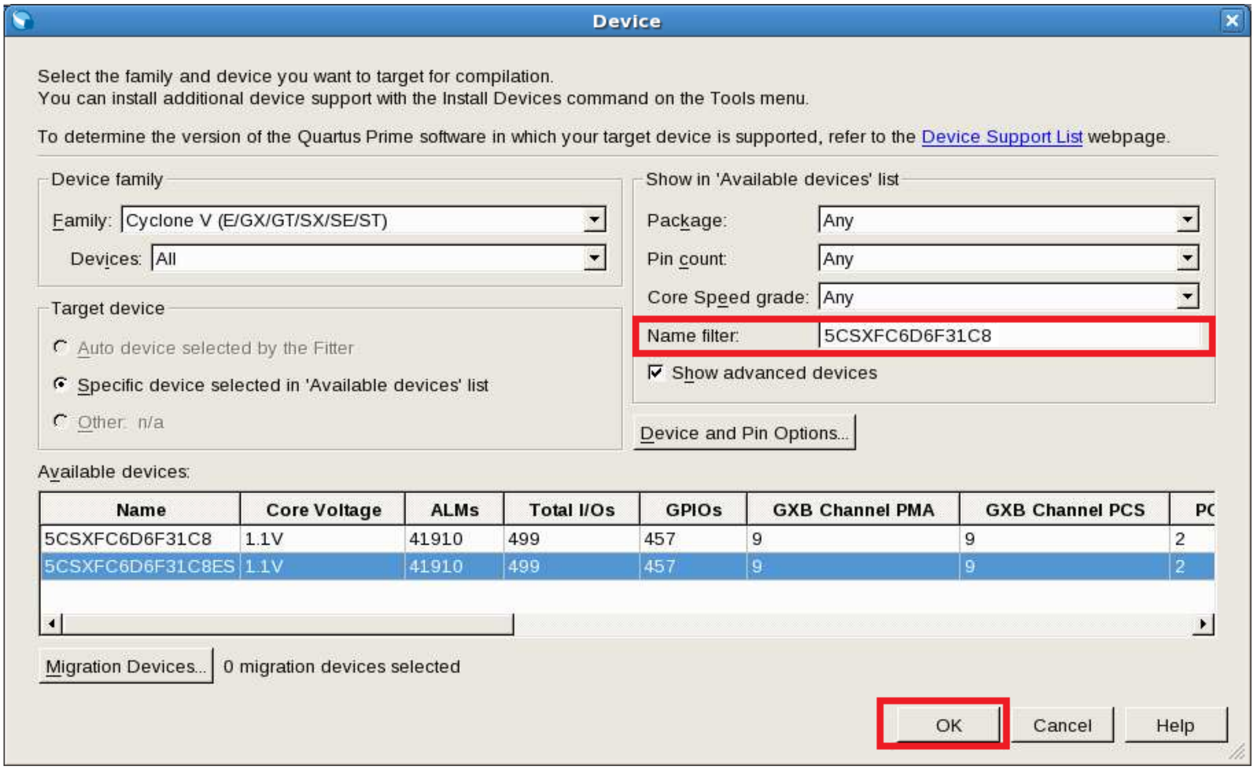
b. Configuration settings in Device and Pin Options

In the upper left Project navigator window, double click the Compilation Hierarchy title “Cyclone V: 5CSXFC6D6F31C8ES” and In the Device panel, click Device and Pin Options.

- In the Device and Pin Options panel, select the correct settings for Configuration Scheme, Configuration Mode and Configuration Device.

c. Change to your own V Series Device

Go back to the Device Panel, change to your device by searching or directly inputting the device number in “Name filter” shown as below.



Device

Select the family and device you want to target for compilation.
You can install additional device support with the Install Devices command on the Tools menu.

To determine the version of the Quartus Prime software in which your target device is supported, refer to the Device Support List webpage.

Device family

Family: Cyclone V (E/GX/GT/SX/SE/ST)

Devices: All

Show in 'Available devices' list

Package: Any

Pin count: Any

Core Speed grade: Any

Name filter: 5CSXFC6D6F31C8

☑ Show advanced devices

Target device

○ Auto device selected by the Fitter

● Specific device selected in 'Available devices' list

○ Other: n/a

Device and Pin Options...

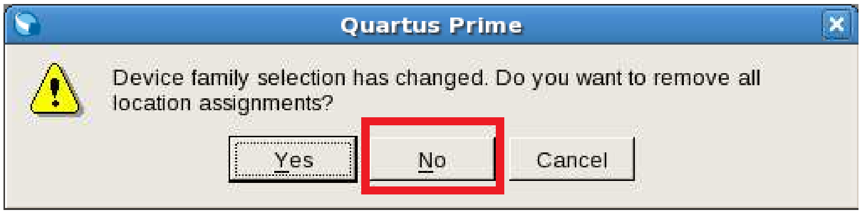
Available devices:

Name	Core Voltage	ALMs	Total I/Os	GPIOs	GXB Channel PMA	GXB Channel PCS	PC
5CSXFC6D6F31C8	1.1V	41910	499	457	9	9	2
5CSXFC6D6F31C8ES	1.1V	41910	499	457	9	9	2

Migration Devices... 0 migration devices selected

<span style="color:red">**Note:**</span> For this user guide, a Cyclone V SoC (5CSXFC6D6F31C8ES) Development Kit is being used as an example. Please make sure you change to your own device number if you are using a different one.

Note: When you try to change the device number, Quartus will ask you whether to remove all pin assignments. Please click NO in the following window. All the pin assignments will be kept at this point and should be modified in next step.

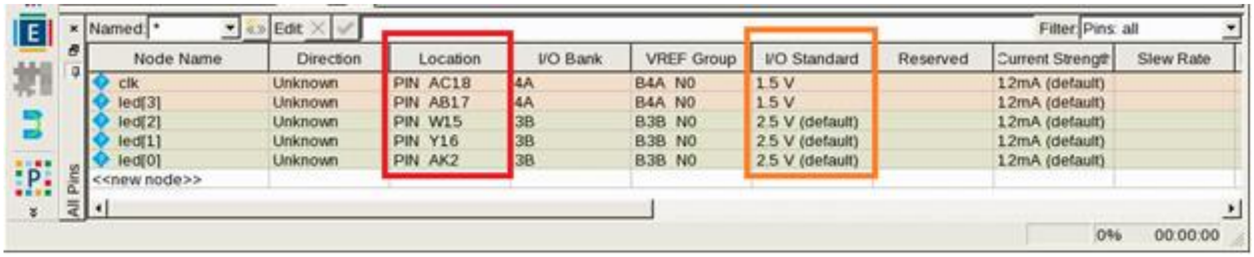


Quartus Prime

Device family selection has changed. Do you want to remove all location assignments?

d. Change Pin Assignments

- In Quartus, click *Assignments > Pin Planner*
- Change "Location" and "I/O Standard"

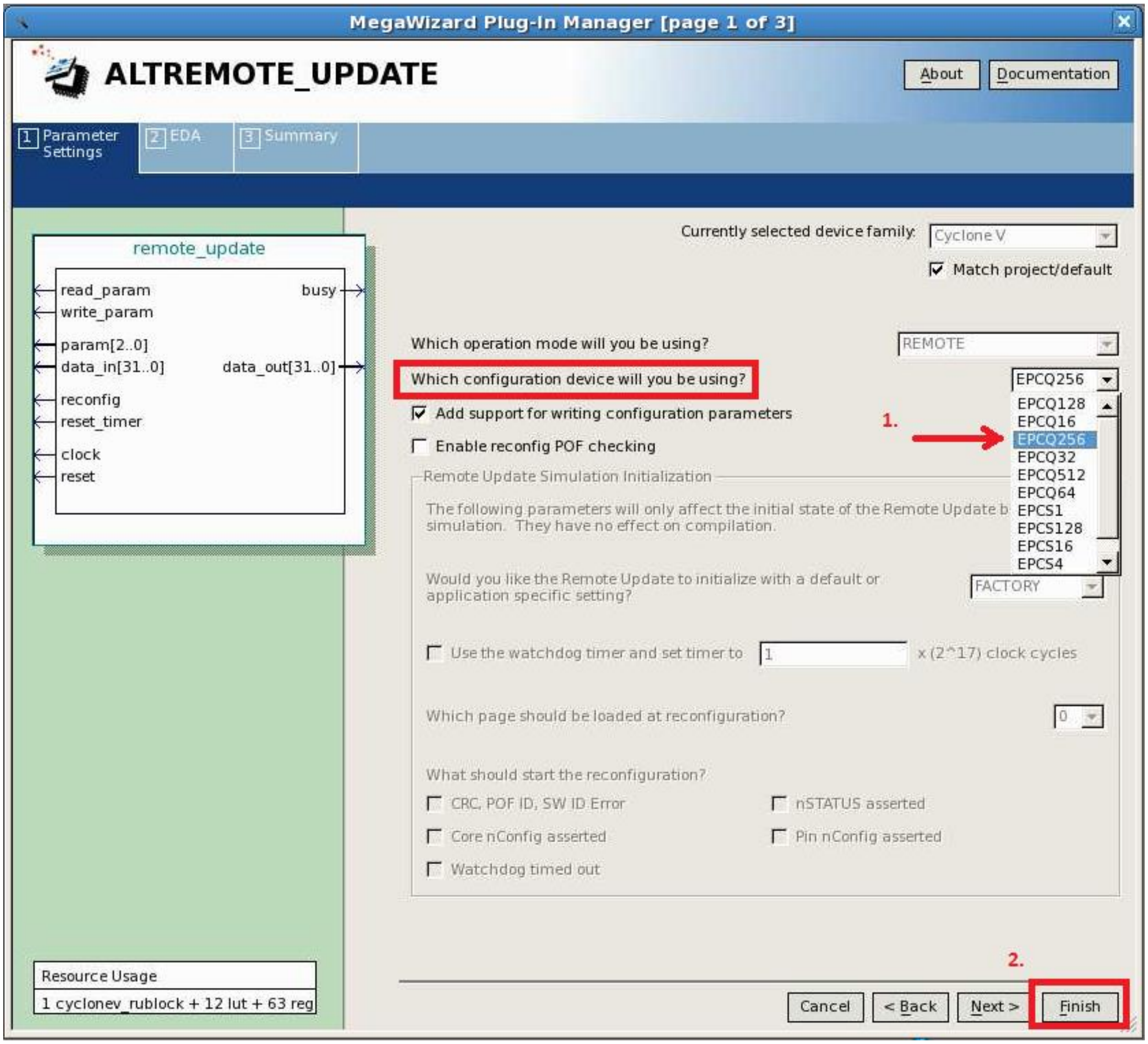


Named: * Edit Filter: Pins: all

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Current Strength	Slew Rate
clk	Unknown	PIN_AC18	4A	B4A_N0	1.5 V		12mA (default)	
led[3]	Unknown	PIN_AB17	4A	B4A_N0	1.5 V		12mA (default)	
led[2]	Unknown	PIN_W15	3B	B3B_N0	2.5 V (default)		12mA (default)	
led[1]	Unknown	PIN_Y16	3B	B3B_N0	2.5 V (default)		12mA (default)	
led[0]	Unknown	PIN_AK2	3B	B3B_N0	2.5 V (default)		12mA (default)	
<<new node>>								

NOTE: Please change both "Location" and "I/O standard" of these 5 pins referring to the "Pin Assignments" Chapter in your device's Board/Device User Guide.

e. Regenerate RSU IP to match your Configuration Device



MegaWizard Plug-In Manager [page 1 of 3]

ALTREMOTE_UPDATE

Currently selected device family: Cyclone V

☑ Match project/default

Which operation mode will you be using? REMOTE

Which configuration device will you be using? EPCQ256

☑ Add support for writing configuration parameters

☐ Enable reconfig POF checking

Remote Update Simulation Initialization

The following parameters will only affect the initial state of the Remote Update b simulation. They have no effect on compilation.

Would you like the Remote Update to initialize with a default or application specific setting? FACTORY

☐ Use the watchdog timer and set timer to 1 x (2^17) clock cycles

Which page should be loaded at reconfiguration? 0

What should start the reconfiguration?

☐ CRC, POF ID, SW ID Error

☐ nSTATUS asserted

☐ Core nConfig asserted

☐ Pin nConfig asserted

☐ Watchdog timed out

Resource Usage

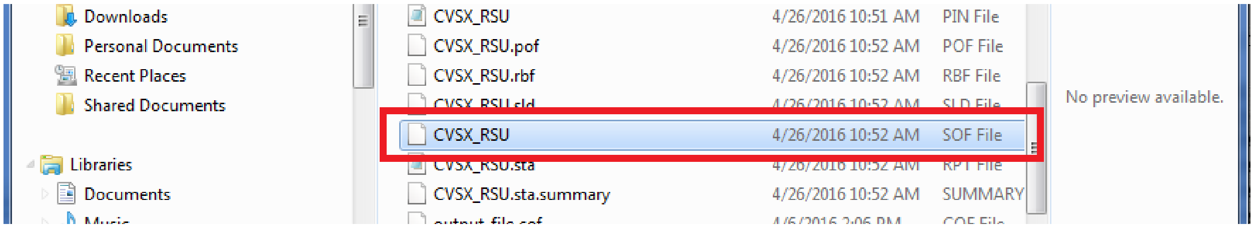
1 cyclonev_rublock + 12 lut + 63 reg

- Find out what configuration device you are using by referring to your Board User Manual.
- In Project navigator window, under “IP Components”, double click “ALTREMOTE_UPDATE”.
- In the MegaWizard Plug-In Manager “Parameter Settings” pane, select the correct configuration device. In this example, **EPCQ256** is selected.
- Click Finish to all the rest of settings to complete.

NOTE: In this example, Cyclone V SoC Development Kit uses EPCQ256 as its configuration Device. Please refer to your specific board’s reference manual to make sure which configuration flash is on your board.

f. Compile and rename the SRAM Object File to factory_image.sof

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



CVSX_RSU	4/26/2016 10:51 AM	PIN File
CVSX_RSU.pof	4/26/2016 10:52 AM	POF File
CVSX_RSU.rbf	4/26/2016 10:52 AM	RBF File
CVSX_RSU.sld	4/26/2016 10:52 AM	SLD File
CVSX_RSU	4/26/2016 10:52 AM	SOF File
CVSX_RSU.sta	4/26/2016 10:52 AM	RPT File
CVSX_RSU.sta.summary	4/26/2016 10:52 AM	SUMMARY
output_file.cof	4/6/2016 2:06 PM	COF File

Step.3 Generate design project and compile Application Image

a. 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.

b. Compile and rename the SRAM Object File to application_image.sof

- Click “Start Compilation”
- After the design is compiled successfully, go to *<destination>/ CVSX_RSU_restored/ output_files*, change the name of CVSX_RSU.sof to **application_image.sof**

Step.4 Creating JTAG Indirect Configuration File (.jic)

a. Open Convert Programming File GUI

- In Quartus, click *File > Convert Programming Files*.

b. Select Programming Type

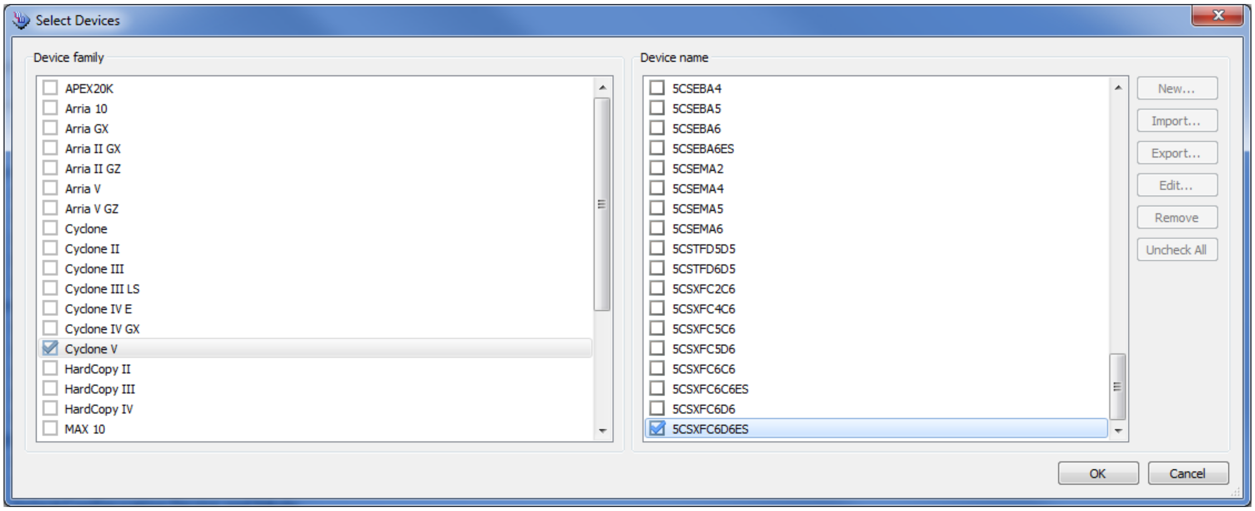
- Select “JTAG Indirect Configuration File (.jic)” as programming file type.

c. Select Configuration Device and Mode

- Select “EPCQ256” as Configuration Device and select “Active Serial x4” 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 “5CSXFC6D6ES”. Please select your device accordingly.



Select Devices

Device family

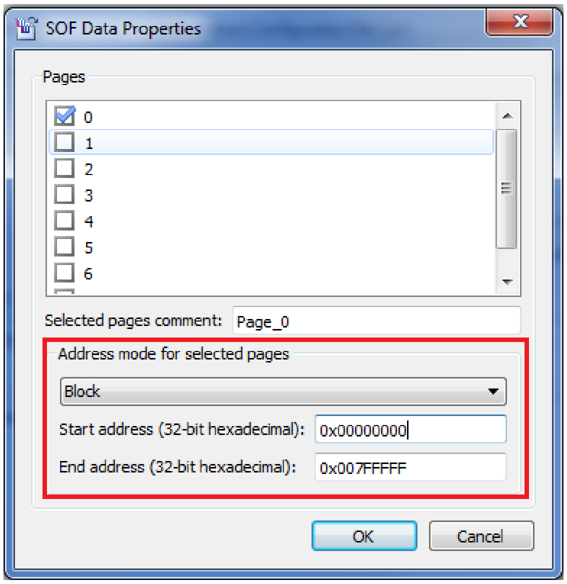
☐ APEX20K
☐ Arria 10
☐ Arria GX
☐ Arria II GX
☐ Arria II GZ
☐ Arria V
☐ Arria V GZ
☐ Cyclone
☐ Cyclone II
☐ Cyclone III
☐ Cyclone III LS
☐ Cyclone IV E
☐ Cyclone IV GX
☑ Cyclone V
☐ HardCopy II
☐ HardCopy III
☐ HardCopy IV
☐ MAX 10

Device name

☐ 5CSEBA4
☐ 5CSEBA5
☐ 5CSEBA6
☐ 5CSEBA6ES
☐ 5CSEMA2
☐ 5CSEMA4
☐ 5CSEMA5
☐ 5CSEMA6
☐ 5CSTFD5D5
☐ 5CSTFD6D5
☐ 5CSXFC2C6
☐ 5CSXFC4C6
☐ 5CSXFC5C6
☐ 5CSXFC5D6
☐ 5CSXFC6C6
☐ 5CSXFC6C6ES
☐ 5CSXFC6D6
☑ 5CSXFC6D6ES

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 (5CSXFC6D6F31)” 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 “0x007FFFFF” as end address (corresponding to the end address of sector 127).
- Click OK to close “SOF Data Properties” window.
- Select “factory_image.sof (5CSXFC6D6F31)” and click “Properties”.
- Check “Compression” box on the top left corner and click OK.



SOF Data Properties

Pages

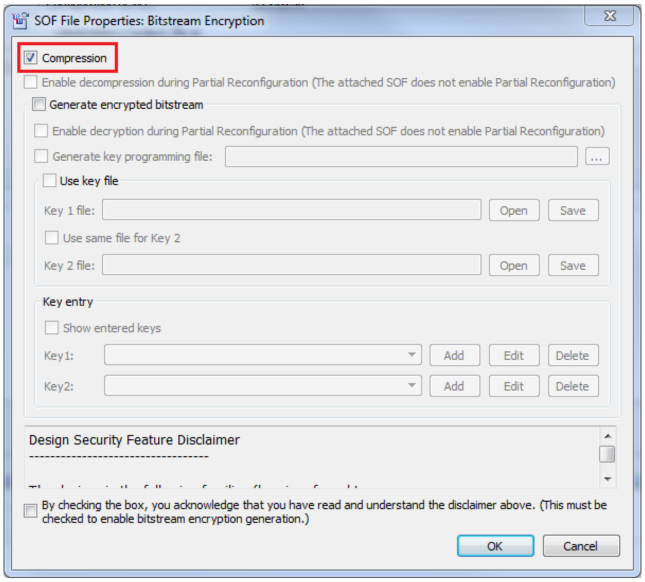
- ☑ 0
- □ 1
- □ 2
- □ 3
- □ 4
- □ 5
- □ 6

Selected pages comment: Page_0

Address mode for selected pages: Block

Start address (32-bit hexadecimal): 0x00000000

End address (32-bit hexadecimal): 0x007FFFFF



SOF File Properties: Bitstream Encryption

- ☑ Compression
- □ Enable decompression during Partial Reconfiguration (The attached SOF does not enable Partial Reconfiguration)
- □ Generate encrypted bitstream
- □ Enable decryption during Partial Reconfiguration (The attached SOF does not enable Partial Reconfiguration)
- □ Generate key programming file:
- □ Use key file

Key 1 file:

□ Use same file for Key 2

Key 2 file:

Key entry

□ Show entered keys

Key1:

Key2:

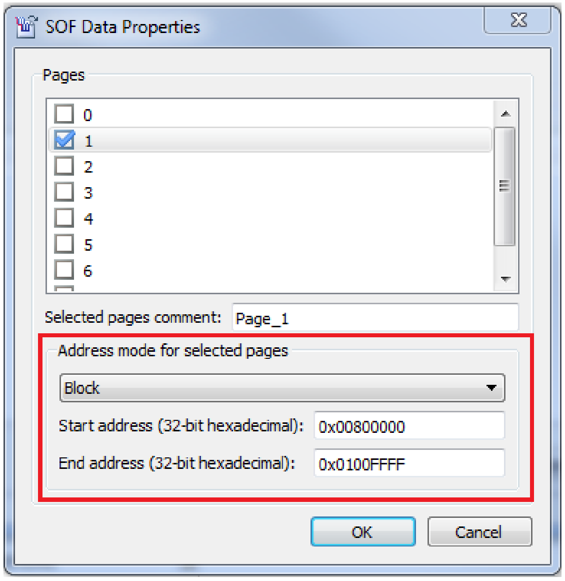
Design Security Feature Disclaimer

□ By checking the box, you acknowledge that you have read and understand the disclaimer above. (This must be checked to enable bitstream encryption generation.)

NOTE: Please see Appendix I - Memory assignment guideline for configuration flash device.

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 (5CSXFC6D6F31)" 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 "0x00800000" as start address (corresponding to the start address of sector 128).
- Enter "0x0100FFFF" as end address (corresponding to the end address of sector 255).
- Click OK to close "SOF Data Properties" window.
- Select "application_image.sof (5CSXFC6D6F31)" and click "Properties".
- Check "Compression" box on the top left corner and click OK.


SOF Data Properties

Pages

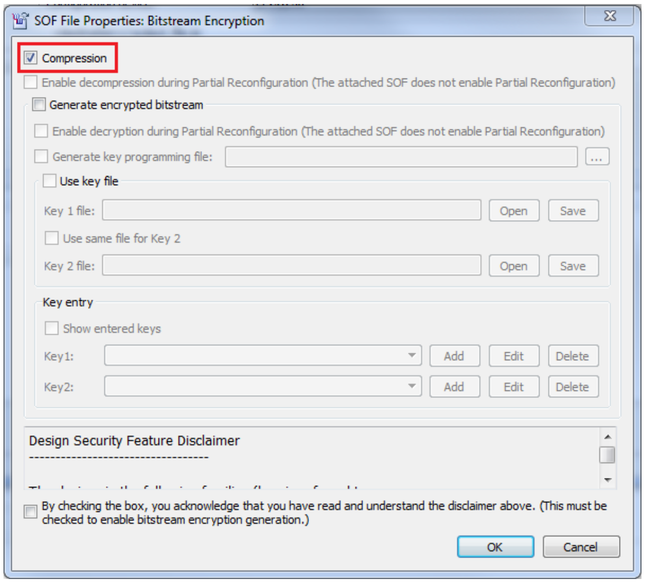
☐ 0
☑ 1
☐ 2
☐ 3
☐ 4
☐ 5
☐ 6

Selected pages comment: Page_1

Address mode for selected pages: Block

Start address (32-bit hexadecimal): 0x00800000

End address (32-bit hexadecimal): 0x0100FFFF


SOF File Properties: Bitstream Encryption

☑ Compression
☐ Enable decompression during Partial Reconfiguration (The attached SOF does not enable Partial Reconfiguration)
☐ Generate encrypted bitstream
☐ Enable decryption during Partial Reconfiguration (The attached SOF does not enable Partial Reconfiguration)
☐ Generate key programming file:
☐ Use key file
Key 1 file:
☐ Use same file for Key 2
Key 2 file:
Key entry
☐ Show entered keys
Key1:
Key2:

Design Security Feature Disclaimer

☐ By checking the box, you acknowledge that you have read and understand the disclaimer above. (This must be checked to enable bitstream encryption generation.)

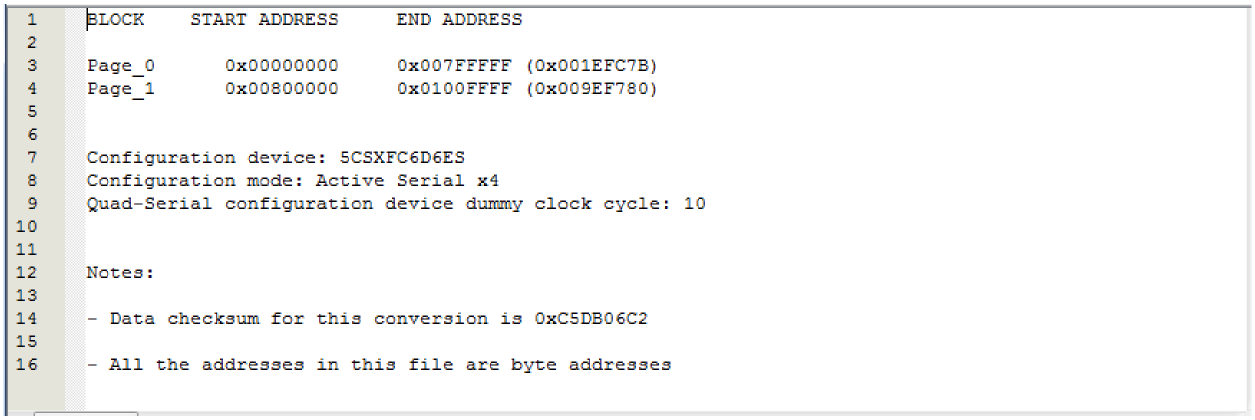
g. Save Conversion Setup File (.cof)

- Check the “Create Memory file” box.
- Click “Save Conversion Setup File” on the upper right.
- A .map file and a .cof file will be generated.

TIPS: Next time when you open this GUI, just click “Open Conversion Setup Data” and select the .cof file you save last time, all the converting data restore in this .cof file will be loaded for convenience.

h. 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:

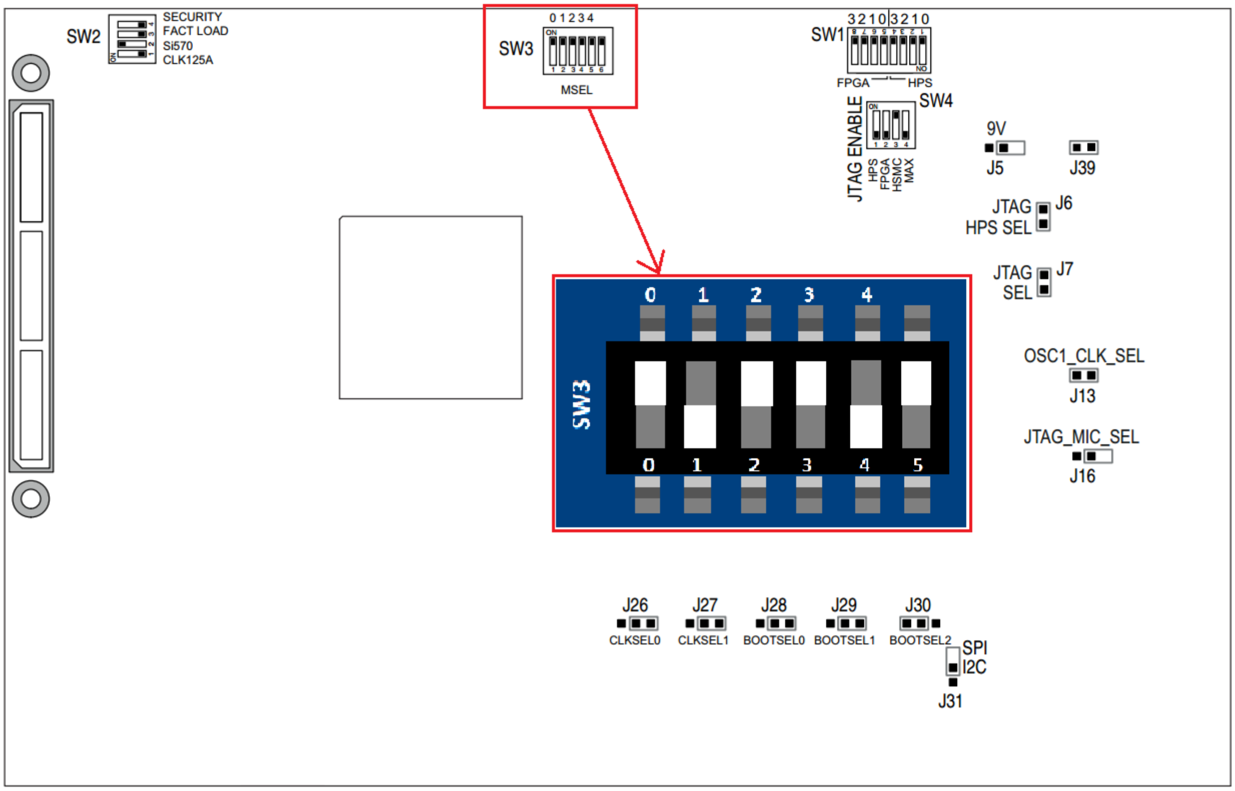

```
BLOCK     START ADDRESS      END ADDRESS

Page_0       0x00000000      0x007FFFFF  (0x001EFC7B)
Page_1       0x00800000      0x0100FFFF  (0x009EF780)


Configuration device: 5CSXFC6D6ES
Configuration mode: Active Serial x4
Quad-Serial configuration device dummy clock cycle: 10


Notes:

- Data checksum for this conversion is 0xC5DB06C2

- All the addresses in this file are byte addresses
```

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

Step.5 Program FPGA

a. MSEL Pins

- For Cyclone V SoC (5CSXFC6D6F31C8ES) Development Kit, the MSEL pin is SW5. For details please refer to its Reference Manual. Toggle the MSEL pins and make the correct selection to enable Active Serial (AS) configuration scheme as shown below.

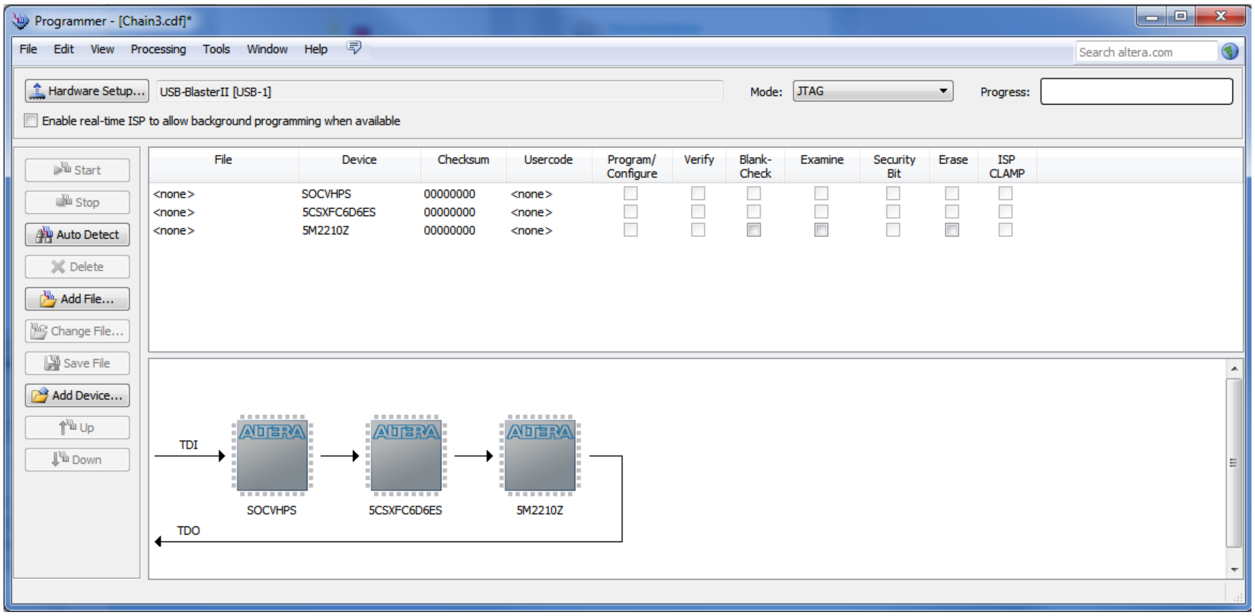


b. Board Connections and Power Up

- Connect the power supply to the board.
- Connect USB Blaster II from board to the PC where your local Quartus Software is installed.

c. 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 “5CSXFC6D6ES”. A JTAG chain should be generated successfully shown as below.



USB-BlasterII [USB-1]

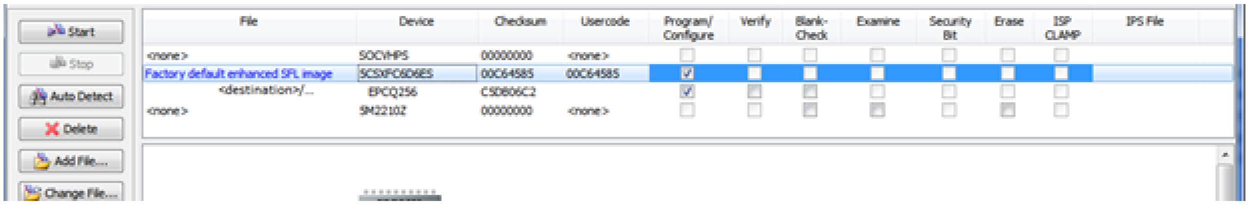
Mode: JTAG

Progress:

☐ Enable real-time ISP to allow background programming when available

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
<none>	SOCVHPS	00000000	<none>	☐	☐	☐	☐	☐	☐	☐
<none>	5CSXFC6D6ES	00000000	<none>	☐	☐	☐	☐	☐	☐	☐
<none>	5M2210Z	00000000	<none>	☐	☐	☐	☐	☐	☐	☐

- Select " <none> 5CSXFC6D6ES 00000000 <none>" and click "Change File"
- Locate the .jic file you generated in Step.4.
- Click box under "Program/Configure" of the .jic file you just added (" <destination> EPCQ256 C5DB6C2"). "<none>" should automatically be changed to "Factory default enhanced SFL image" under "File" of your Cyclone device.



File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP	IPS File
<none>	SOCVHPS	00000000	<none>	☐	☐	☐	☐	☐	☐	☐	
Factory default enhanced SFL image	5CSXFC6D6ES	00C64585	00C64585	☑	☐	☐	☐	☐	☐	☐	
<destination>/...	EPCQ256	C5DB06C2		☑	☐	☐	☐	☐	☐	☐	
<none>	5M2210Z	00000000	<none>	☐	☐	☐	☐	☐	☐	☐	

- Click "Start".

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 V SX) 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: EPCQ256) 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 EPCQ256 will be loaded in FPGA.

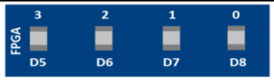
d. LED Observations

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.

☒ - LED Blinking 🟩 - LED ON ■ - LED OFF

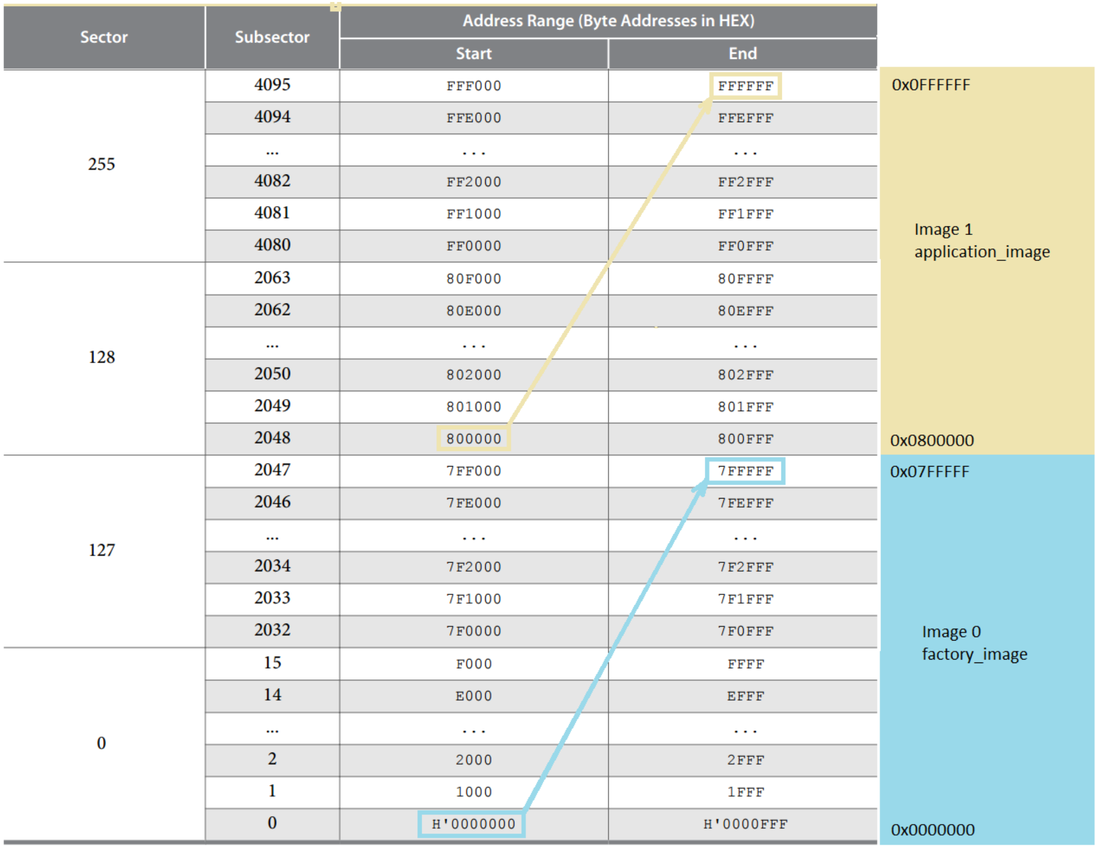
State # (LED [2..0])	**Functions**					**Image Status**
State 0 (3’b000)	**reset =1**, reset RSU IP	☒	■	■	■	factory_Image
State 1 (3’b001)	release reset	☒	■	■	🟩	
State 2 (3’b010)	empty state	☒	■	🟩	■	
State 3 (3’b011)	**AnF=1**	☒	■	🟩	🟩	
State 4 (3’b100)	Set application_image start address	☒	🟩	■	■	
State 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

Appendix

I. Memory assignment guideline for configuration flash device

Step 1. Identify the configuration flash device and refer to datasheet.

- In this design example, the configuration device is EPCQ256 and you can find the datasheet along with the development kit package.
- In the “Memory Array Organization” chapter, find the address range for EPCQ256 (Page-9).



Sector	Subsector	Address Range (Byte Addresses in HEX)	
		Start	End
255	4095	FFF000	FFFFFF
	4094	FFE000	FFEFFF
	...	...	...
	4082	FF2000	FF2FFF
	4081	FF1000	FF1FFF
	4080	FF0000	FF0FFF
128	2063	80F000	80FFFF
	2062	80E000	80EFFF
	...	...	...
	2050	802000	802FFF
	2049	801000	801FFF
	2048	800000	800FFF
127	2047	7FF000	7FFFFF
	2046	7FE000	7FEFFF
	...	...	...
	2034	7F2000	7F2FFF
	2033	7F1000	7F1FFF
	2032	7F0000	7F0FFF
0	15	F000	FFFF
	14	E000	EFFF
	...	...	...
	2	2000	2FFF
	1	1000	1FFF
	0	H'0000000	H'0000FFF

Step 2. Estimate the size of configuration file and allocate memory

- There are 512 sectors in EPCQ256 and each sector has 16 subsectors. The start address of the flash is H’0000000 and the end address of entire flash is H’1FFFFFF.
- In this design example, we only use half of the flash memory (Sector 0 ~ Sector 255). And we further break down these 256 sectors in half and each stores 1 image.
- Set image 0 starts from sector 0 (start address is H’0000000) and ends at sector 127 (end address is H’07FFFFF). For image 1, it should start from sector 128 (start address is H’0800000) and ends at sector 255 (end address is H’0FFFFFF).

II. Debugging checklist and common issues

FPGA Configuration Troubleshooter lists all the common issues during configuration and their debugging solutions.