



# TABLE OF CONTENTS

|                  |  |           |
|------------------|--|-----------|
| <b>Chapter 1</b> | <b>Introduction</b>  | <b>3</b>  |
| 1.1              | Package Contents   | 3         |
| 1.2              | TSoM Design Package  | 4         |
| 1.3              | Getting Help   | 4         |
| <b>Chapter 2</b> | <b>Board Specification</b>                                 | <b>5</b>  |
| <b>Chapter 2</b> |  | <b>5</b>  |
| 2.1              | Layout and Components                                      | 5         |
| 2.2              | Block Diagram  | 6         |
| <b>Chapter 3</b> | <b>System Interface</b>                                    | <b>9</b>  |
| <b>Chapter 3</b> |  | <b>9</b>  |
| 3.1              | Settings of FPGA Configuration Mode                        | 9         |
| 3.2              | Configuration of Cyclone V SoC FPGA on TSoM evaluation kit | 11        |
| 3.3              | Board Status Elements                                      | 18        |
| 3.4              | Board Reset Elements                                       | 18        |
| <b>Chapter 4</b> | <b>HPS Fabric Components</b>                               | <b>20</b> |
| <b>Chapter 4</b> |  | <b>20</b> |
| 4.1              | Ethernet RJ45  | 20        |
| 4.2              | USB OTG Connector  | 21        |
| 4.3              | UART to USB  | 22        |
| 4.4              | Micro SD Card  | 23        |
| 4.5              | QSPI Flash (DNI)   | 23        |
| 4.6              | Raspberry Pi 2x13 GPIO                                     | 24        |
| <b>Chapter 5</b> | <b>FPGA Fabric Components</b>                              | <b>28</b> |
| <b>Chapter 5</b> |  | <b>28</b> |
| 5.1              | User Push-buttons, Switches and LEDs                       | 28        |

|   |           |
|---|-----------|
| 5.2 TMD Header .....                              | 30        |
| 5.3 A/D Converter and Analog Input .....          | 31        |
| 5.4 HDMI TX Interface .....                       | 33        |
| <b>Chapter 6 Examples For FPGA .....</b>          | <b>35</b> |
| <b>Chapter 6 .....</b>                            | <b>35</b> |
| 6.1 HDMI TX.....                                  | 35        |
| 6.2 DDR3_RTL.....                                 | 38        |
| <b>Chapter 7 Programming the EPCS Device.....</b> | <b>41</b> |
| <b>Chapter 8 eMMC Programming .....</b>           | <b>53</b> |
| <i>Appendix</i> 61                                |           |
| <b>Chapter 7 .....</b>                            | <b>61</b> |
| <b>Chapter 8 .....</b>                            | <b>61</b> |
| <b>Chapter 9 .....</b>                            | <b>61</b> |
| 9.1 Revision History .....                        | 61        |

# Chapter 1

## Introduction

TSoM (Terasic SoC System on Module) Evaluation Kit is composed of TSoM module and Based Board(TSoM-BB). TSoM module is small, integrated single-board computers with Cyclone® V SoC at the core. The SoC SoM includes DDR3 memory, flash memory, power management, common interface controllers. Users can install the TSoM based board on it through the 260-pin edge connector, combined to a complete FPGA evaluation kit.

TSOM-BB is a based board developed based on TSoM card. The main purpose is providing expansion interface, power and JTAG configuration function for TSoM, it connects the FPGA and HPS fabric I/O of TSoM to many application interfaces, such as USB, Ethernet, HDMI, Micro SD card and so on, it expanded the FPGA I/O which are connected to 260-pin edge connector to a variety of applications. The TSoM based board also provides USB Blaster II circuit, users can configure and debug the FPGA on TSoM through the JTAG interface. TSoM based board provides power source to TSoM through the 260-pin edge connector. With the TSoM based board, users can use TSoM Evaluation Kit to develop various FPGA projects.

### 1.1 Package Contents

Figure 1-1 shows a photograph of TSoM Evaluation Kit.

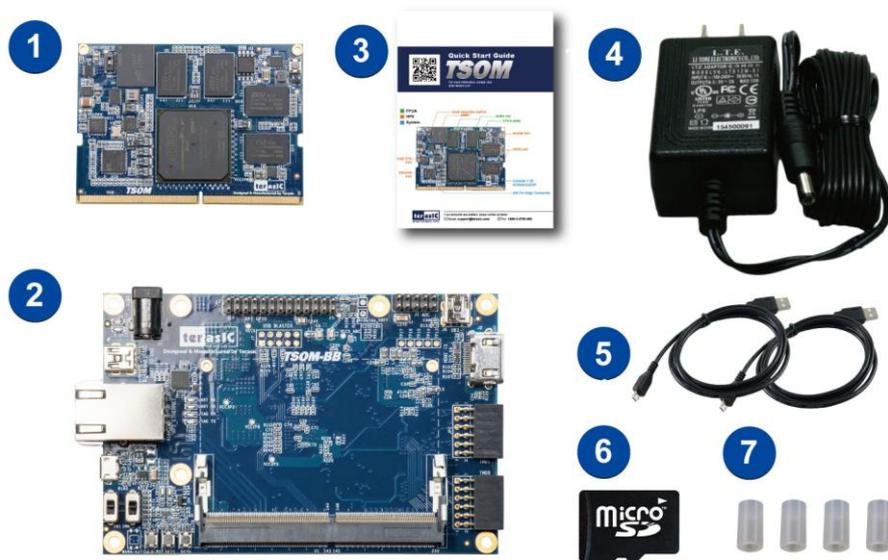


Figure 1-1 TSoM Evaluation Kit Package Contents

The TSoM Evaluation Kit package includes:

1. TSoM Module (installed)

2. TSOM Based board (TSOM-BB)
3. Quick Start Guide
4. Power DC Adapter (5V)
5. Type A to Mini-B USB Cable x2
6. MicroSD card (installed)
7. Four Silicon Footstands

## **1.2 TSoM Design Package**

The TSoM Evaluation Kit design package contains all the documents and supporting materials associated with TSoM Evaluation Kit, including the user manual, reference designs, and device datasheets.

Users can download this design package from the link: <http://TSoM.terasic.com/cd>.

## **1.3 Getting Help**

Here are the addresses where you can get help if you encounter any problems:

Terasic Technologies

9F., No.176, Sec.2, Gongdao 5th Rd, East Dist, Hsinchu City, 30070. Taiwan

Email: [support@terasic.com](mailto:support@terasic.com)

Tel.: +886-3-575-0880

Website: [TSoM.terasic.com](http://TSoM.terasic.com)

# Chapter 2

## *Board Specification*

This chapter provides an introduction to the features and design characteristics of the board.

### 2.1 Layout and Components

Figure 2-1 and Figure 2-2 shows a photograph of the TSoM Evaluation Kit (TSoM module board and TSOM Based Board). It depicts the layout of the board and indicates the location of the connectors and key components.

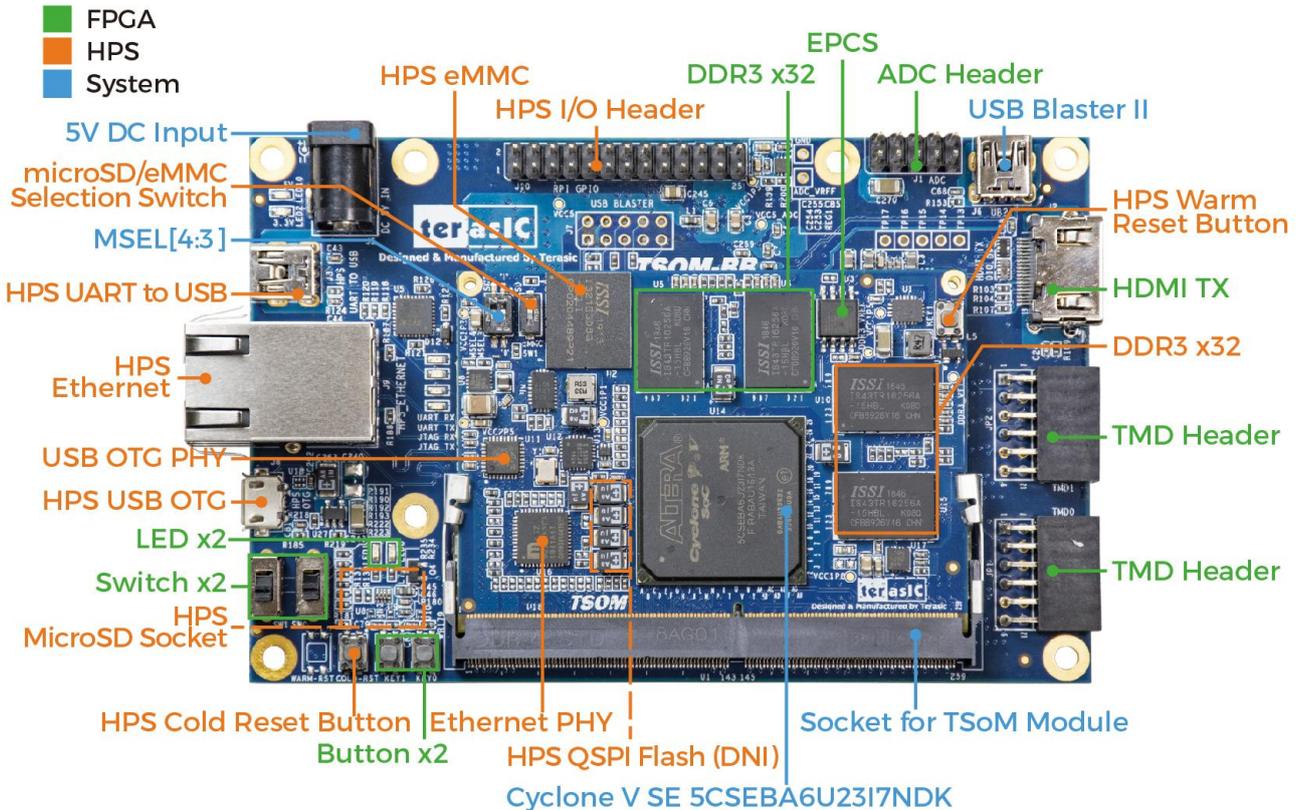


Figure 2-1 TSoM Evaluation Kit (Top View)

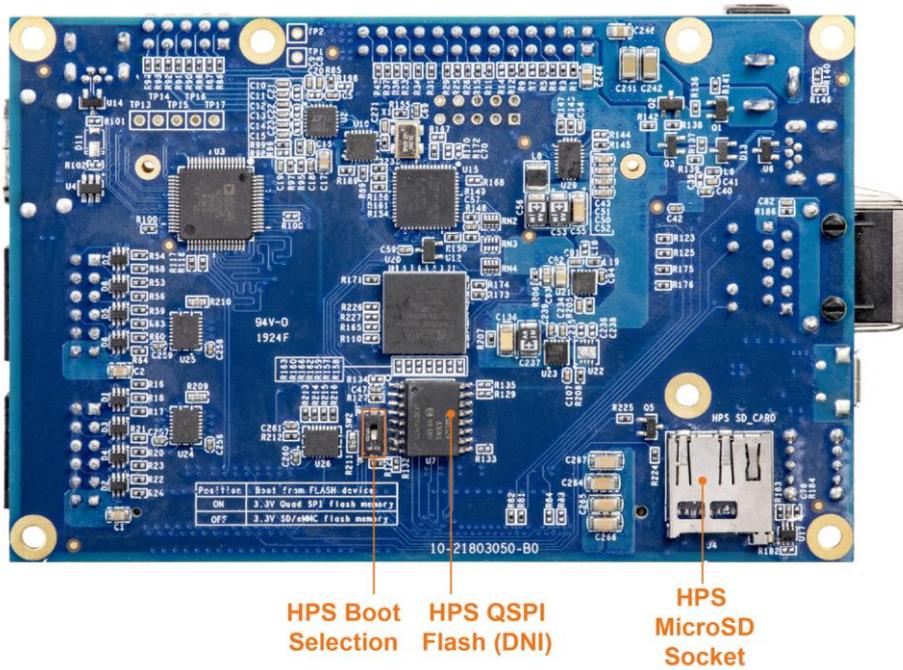


Figure 2-2 TSoM Evaluation Kit (Back View)

## 2.2 Block Diagram

Figure 2-3 is the block diagram of the TSoM Evaluation Kit and Figure 2-4 shows the block diagram of the TSoM Module board

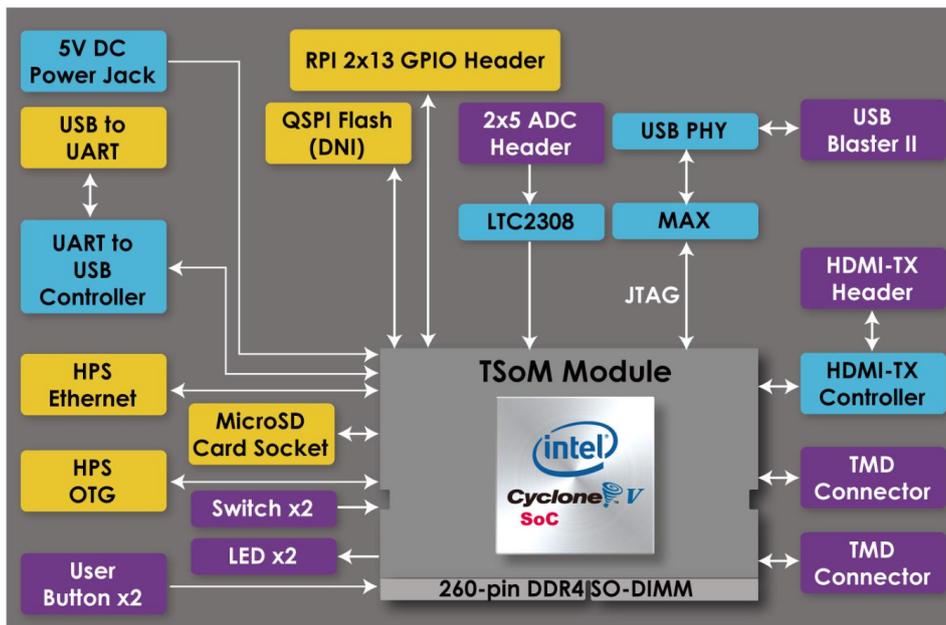
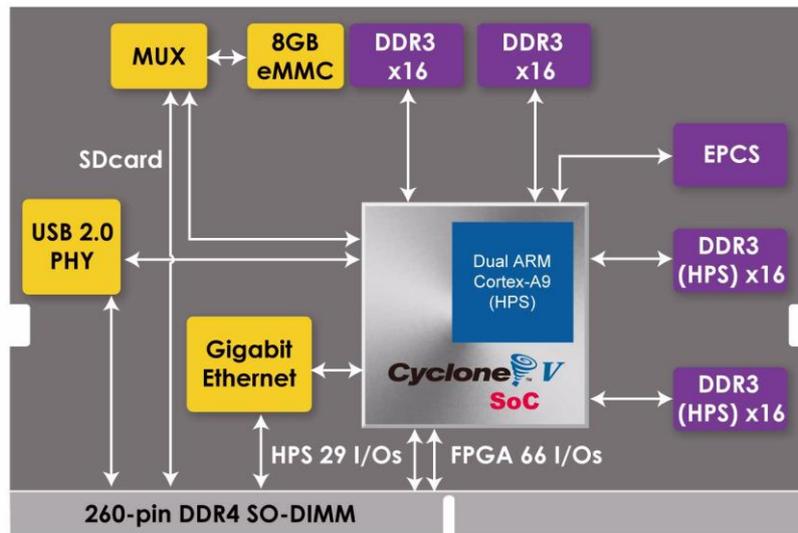


Figure 2-3 Block diagram of the TsoM evaluation kit



**Figure 2-4 Block diagram of the TSoM Module**

Detailed information about TSoM module and TSOM based board are listed below.

■ **TSoM Module**

- FPGA: Cyclone V SE 5CSEBA6U23I7NDK (110K LEs)
- Interface: DDR4 Edge (include 3.3V power source)
- Dimension: 50 mm x 70 mm
- **FPGA Fabric side:**
  - DDR3 SDRAM 1GB, 32bit – 303MHz (Soft IP)
  - LVDS Transmitter x15 & LVDS Receiver x17 pairs & GPIO x3 (Total GPIO x67)
  - EPCS64
- **HPS Fabric side:**
  - Boot Selection DIP Switch: boot from eMMC or MicroSD Card
  - DDR3 SDRAM 1GB, 32bit - 400MHz
  - USB 2.0 PHY
  - Gigabit Ethernet PHY
  - 3.3V GPIO x25 (Can/UART/SPI/I2C/Trace Buses)
  - 1.5V GPI x4 (Input pins)
  - eMMC 8GB

■ **TSOM based board**

- **System:**
  - Power source: 5V DC
  - on-board USB Blaster II
  - DDR4 socket for TSOM installation
- **FPGA Fabric:**
  - LED x 2, Key x 2, Switch x 2
  - HDMI TX v1.4, 1080P Full-HD, Compatible with HDCP v1.4
  - TMD Header x2 (support 16 GPIO)
  - ADC, 8-channel, 12-bit, 500Ksps
- **HPS Fabric:**
  - MicroSD Socket
  - Optional 512Mb QSPI FLASH (Need to rework Boot Selection resistor on Module)
  - USB to UART (USB OTG PHY, Mini-B Connector)
  - Ethernet (RJ45 Connector)
  - USB OTG (Micro-AB Connector)
  - 2x13 GPIO (include I2C/UART/SPI), Compatible with Raspberry Pi Expansion IO

# Chapter 3

## System Interface

This chapter provides an instruction for the system interface of the TSoM evaluation kit.

### 3.1 Settings of FPGA Configuration Mode

Since the factory setting of MSEL[4:0] on the TSoM module is FPPx32 mode(See **Figure 3-1**). It means that when the TSoM evaluation kit is power on, the FPGA is configured from the HPS fabric. When the software on the HPS is running, the FPGA can be configured via HPS.

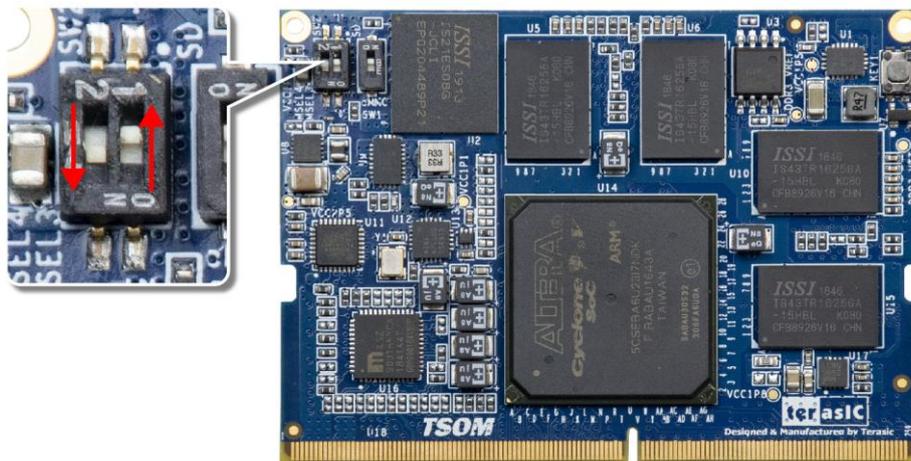
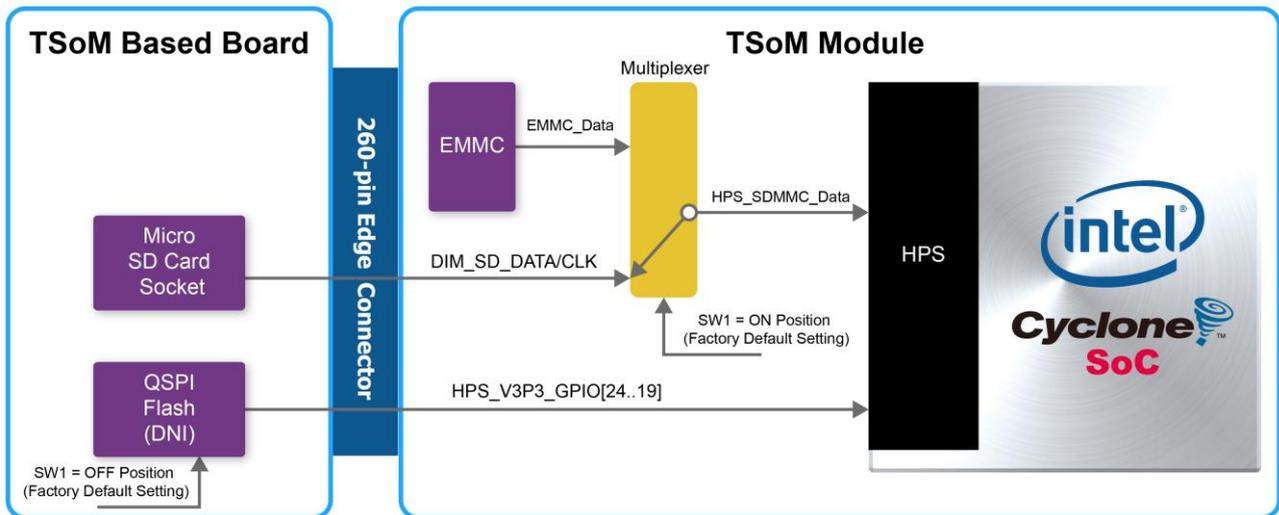


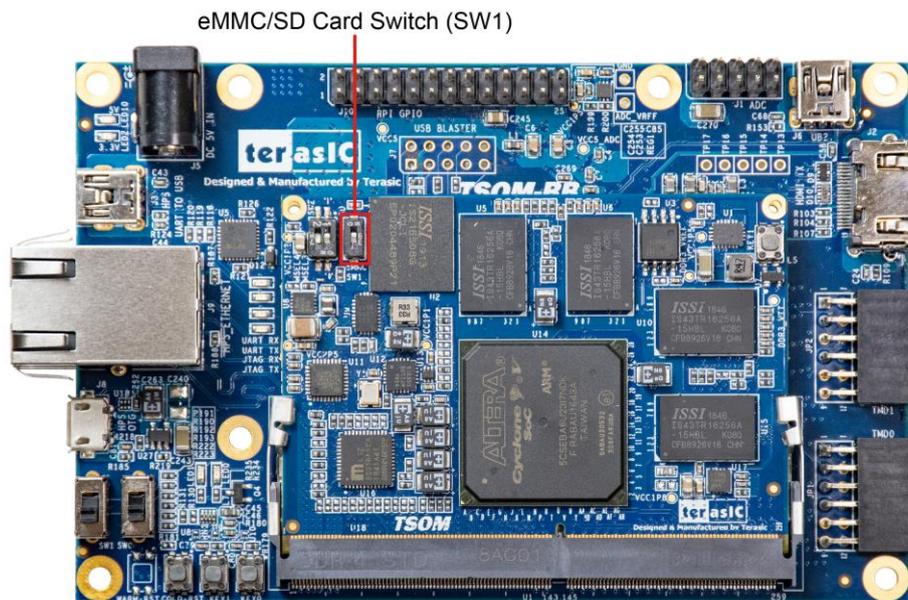
Figure 3-1 TsoM module is setting to FPPx32 mode

As shown in the **Figure 3-2**, there are 3 storage devices can be supported on TSoM evaluation kit to boot HPS: SD Card, QSPI flash and eMMC flash. The eMMC flash is embedded in the TSoM module. The SD Card and QSPI Flash are on the TSoM based board.



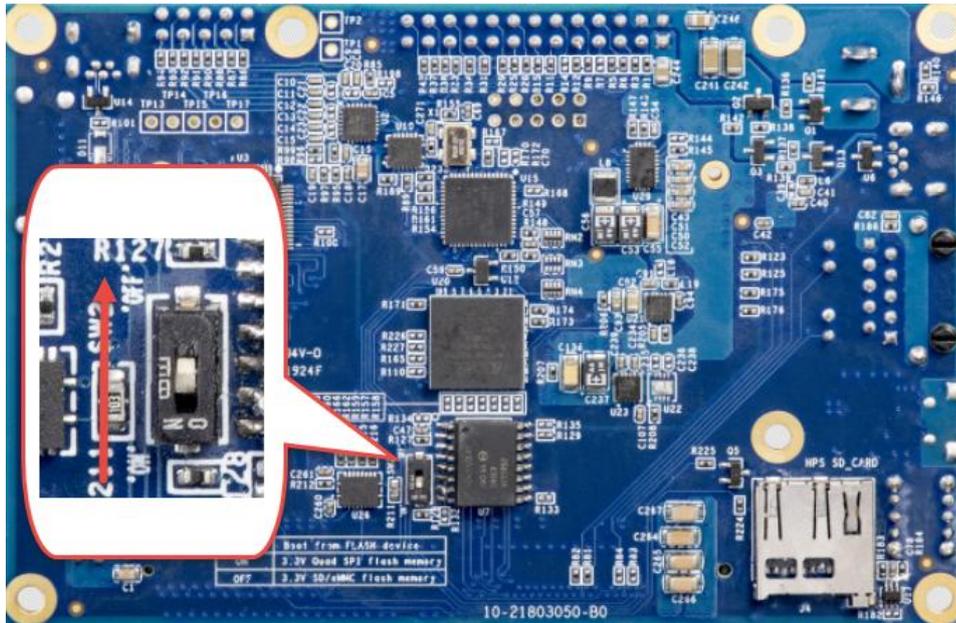
**Figure 3-2 Boot device for HPS on the TsoM evaluation kit**

As described in section 3.1 of the TSoM module manual, the SD card shares the same HPS data bus as the eMMC flash. Therefore, only one of these two storage devices can be used to boot HPS at the same time. Users can switch between the two boot devices through the switch SW1 on the TSoM module board (See **Figure 3-3**). The factory default boot device is SD card interface (SW1 is set to "ON" position).



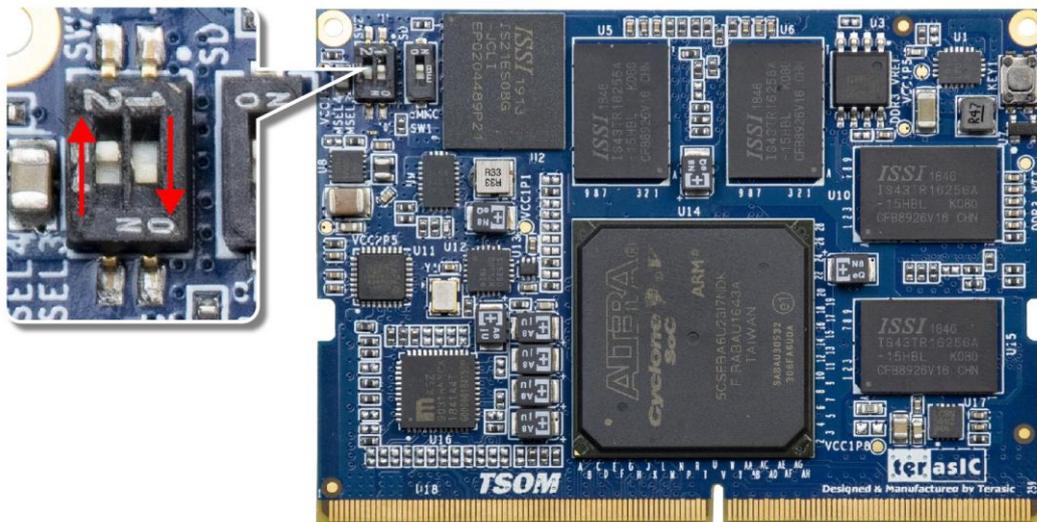
**Figure 3-3 eMMC/SD Card select switch**

The QSPI Flash is the option function and don't install on the board . Users can apply it on their own if needed. After the user installs the QSPI flash, user can adjust the SW2 on the back of the TSoM evaluation kit (See **Figure 3-4**). Switch it to "ON" position that will change the HPS boot source to QSPI flash.



**Figure 3-4 QSPI flash/SD Card select switch(Setting to SD Card )**

Also, if the user wants the FPGA is configured by EPCS on TSoM evaluation kit when power on. Then refer to section 3.1 of the TSoM module manual for EPCS part. Modify the SW2 on the TSoM module to AS mode (See [Figure 3-5](#)).



**Figure 3-5 SW2 Setting Resistors (AS Mode)**

## 3.2 Configuration of Cyclone V SoC FPGA on TSoM evaluation kit

There are two types of programming method supported by TSoM evaluation kit:

1. **JTAG programming:** It is named after the IEEE standards Joint Test Action Group. The

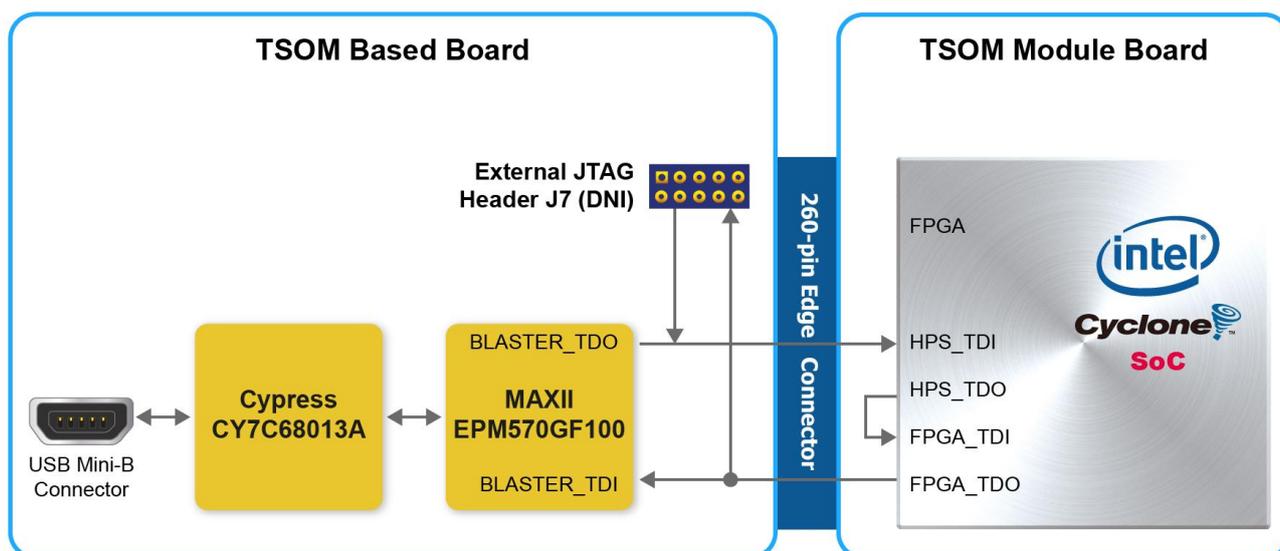
configuration bit stream is downloaded directly into the Cyclone V SoC FPGA. The FPGA will retain its current status as long as the power keeps applying to the board; the configuration information will be lost when the power is off.

2. **AS programming:** The other programming method is Active Serial configuration. The configuration bit stream is downloaded into the serial configuration device (EPCS64), which provides non-volatile storage for the bit stream. The information is retained within EPCS64 even if the TSoM evaluation kit board is turned off. When the board is powered on, the configuration data in the EPCS64 device is automatically loaded into the Cyclone V SoC FPGA.

## ■ JTAG Chain on TSoM Evaluation Kit

The FPGA device can be configured through JTAG interface on TSoM evaluation kit board, but the JTAG chain must form a closed loop, which allows Quartus II programmer to detect FPGA device. **Figure 3-6** illustrates the JTAG chain on TSoM evaluation kit board.

In addition, the TSoM evaluation kit has one external JTAG Header (J7) reserved for users to connect to JTAG chain of the TSoM evaluation kit via external blaster. The J7 header is not installed, so users need to solder a 2.54mm 2 x 5 male pin header if it is necessary.

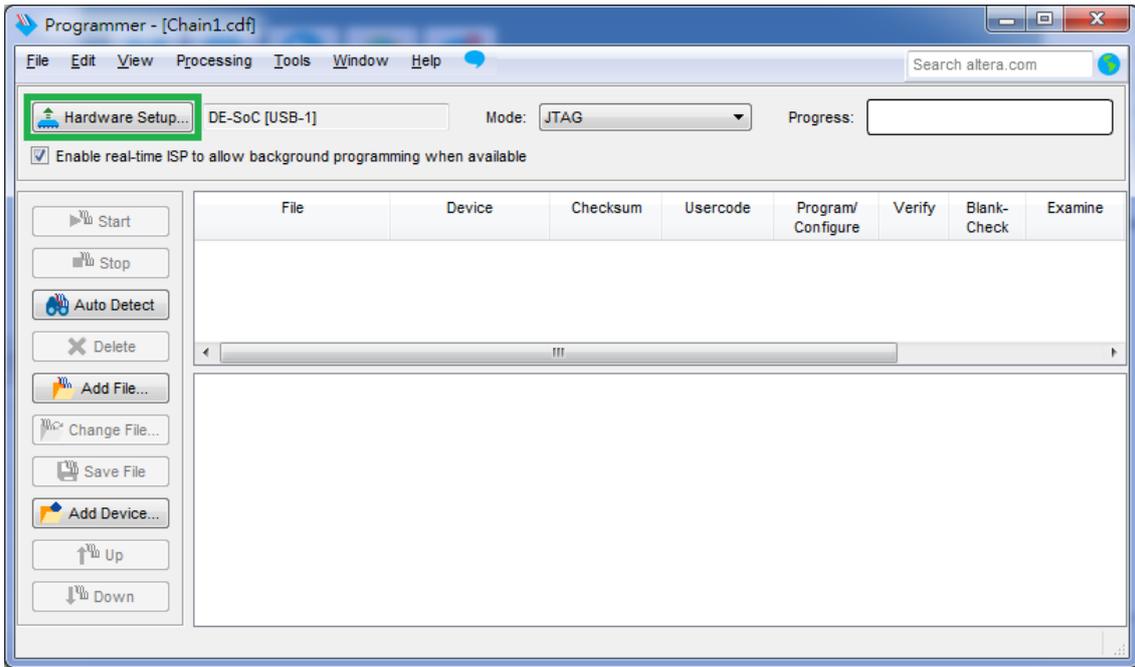


**Figure 3-6 Path of the JTAG chain**

## ■ Configure the FPGA in JTAG Mode

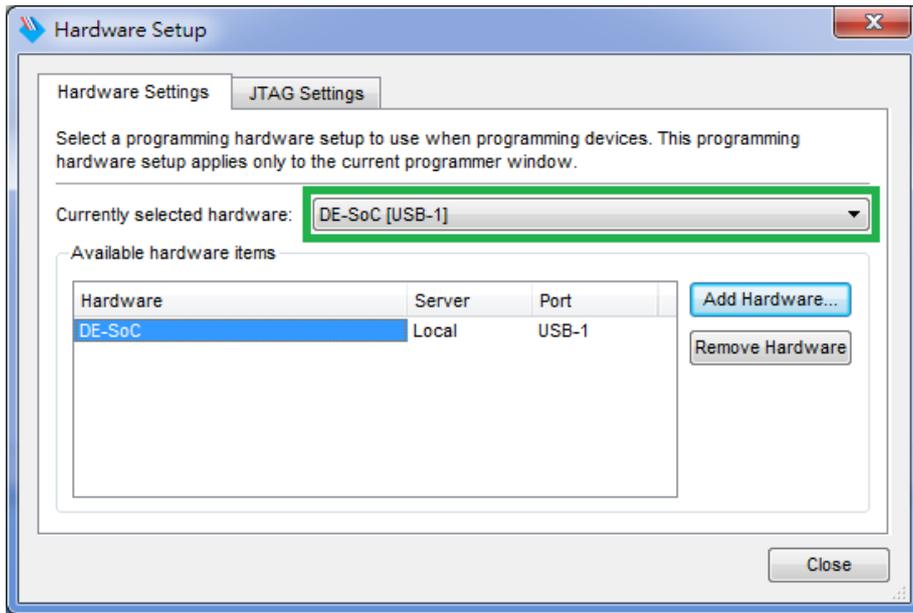
There are two devices (FPGA and HPS) on the JTAG chain. The following shows how the FPGA is programmed in JTAG mode step by step.

Open the Quartus II programmer, please Choose **Tools > Programmer**. The Programmer window opens. Please click “**Hardware Setup**”, as circled in **Figure 3-7**.



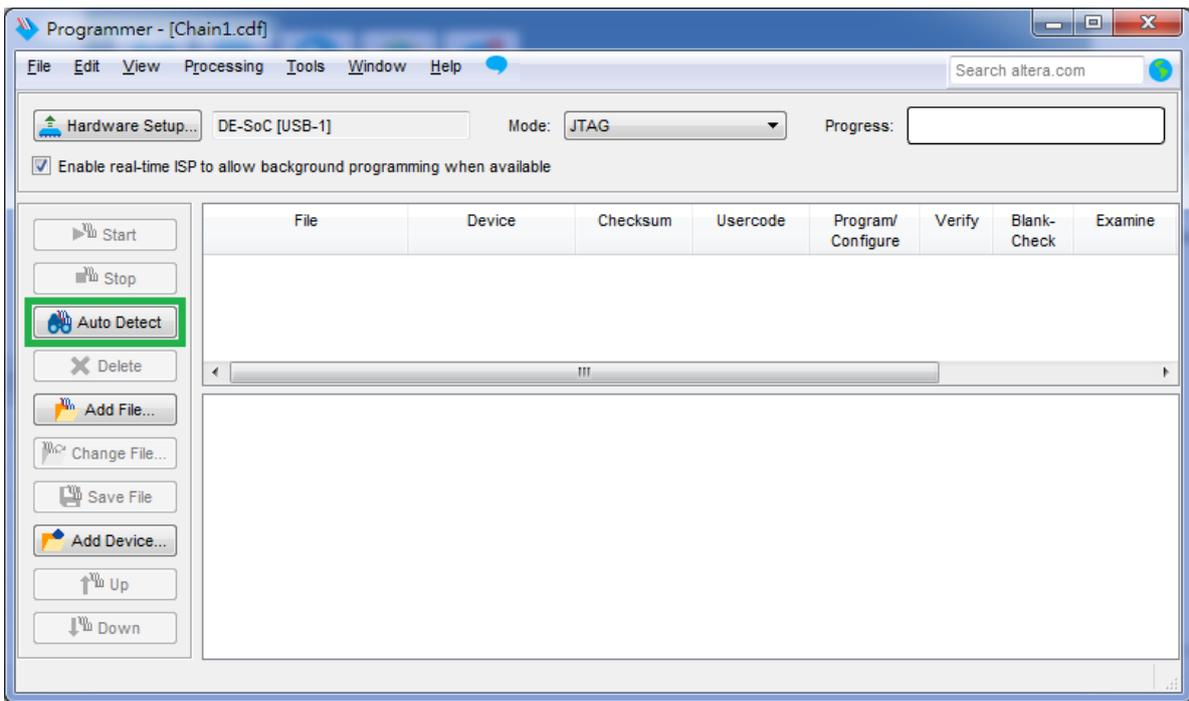
**Figure 3-7 Programmer Window**

If it is not already turned on, turn on the DE-SoC [USB-1] option under currently selected hardware and click “Close” to close the window. See [Figure 3-8](#).



**Figure 3-8 Hardware Setting**

Return to the Quartus II programmer and click “Auto Detect”, as circled in [Figure 3-9](#).



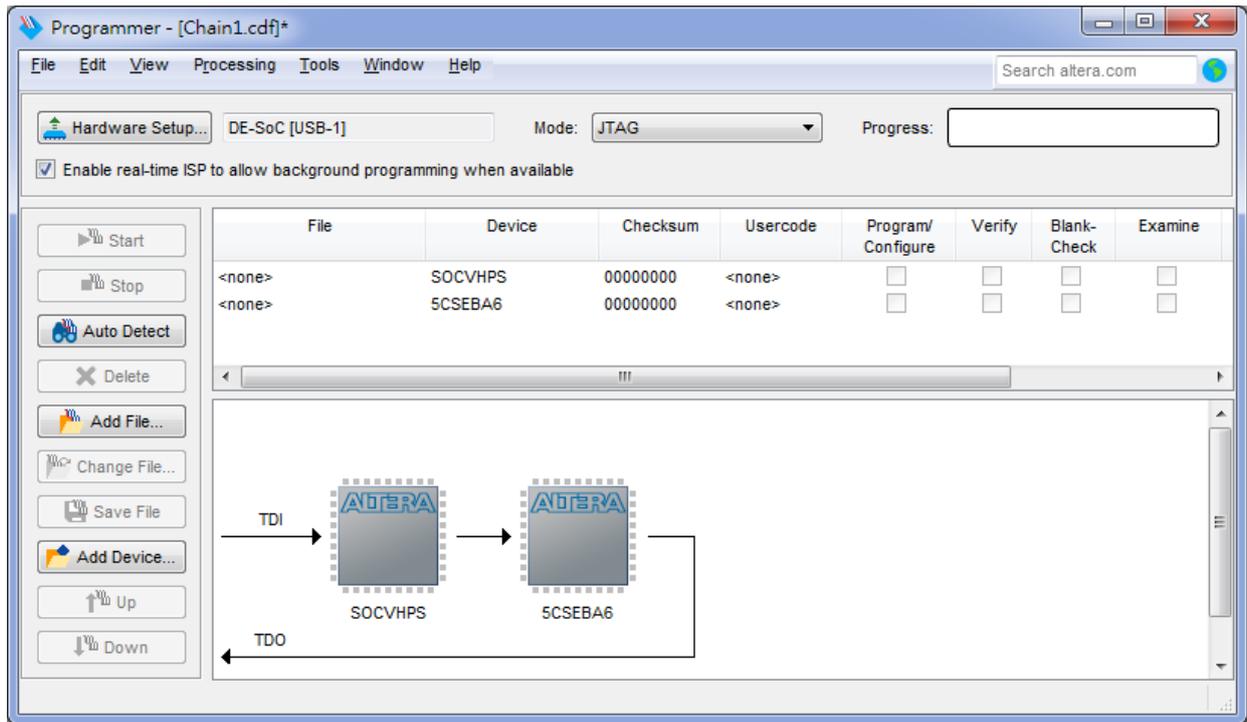
**Figure 3-9 Detect FPGA device in JTAG mode**

If the device is detected, the window of the selection device is opened, Please select detected device associated with the board and click “OK” to close the window, as circled in **Figure 3-10**.



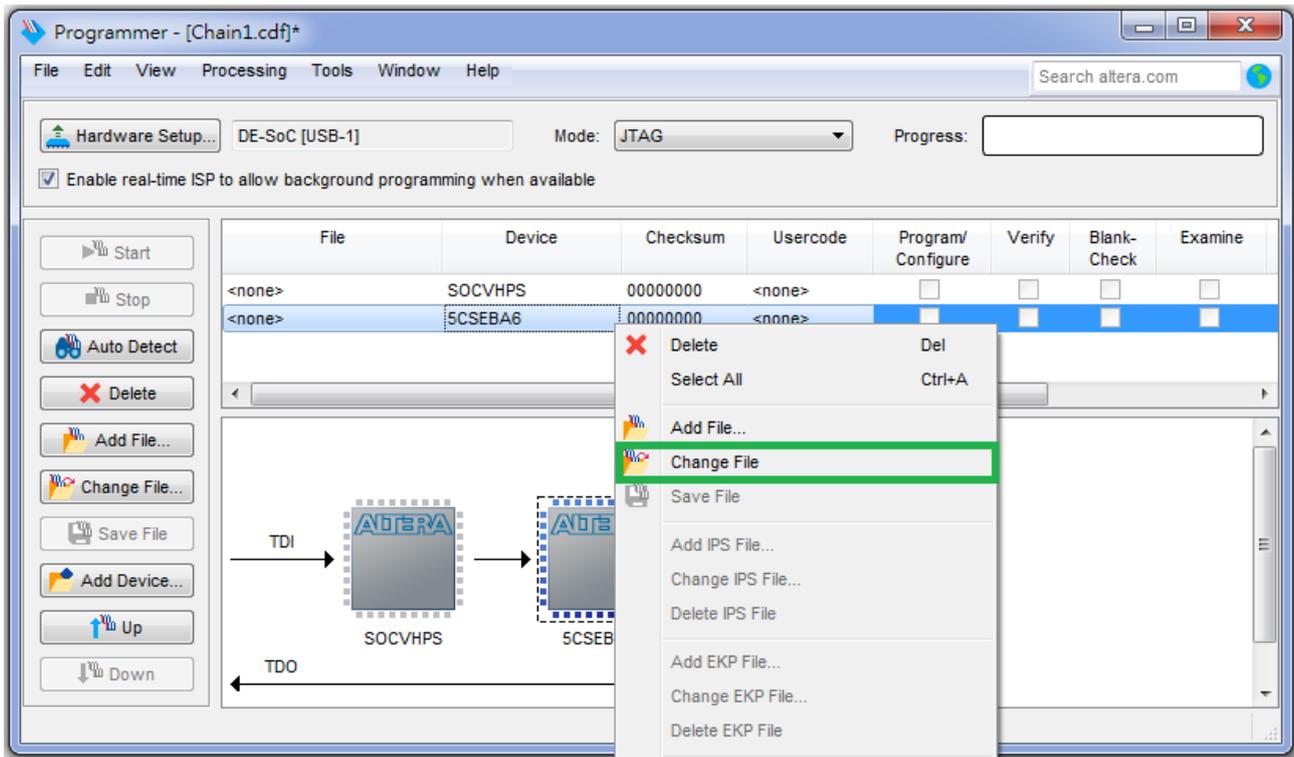
**Figure 3-10 Select 5CSEBA6 device**

Both FPGA and HPS are detected, as shown in **Figure 3-11**.



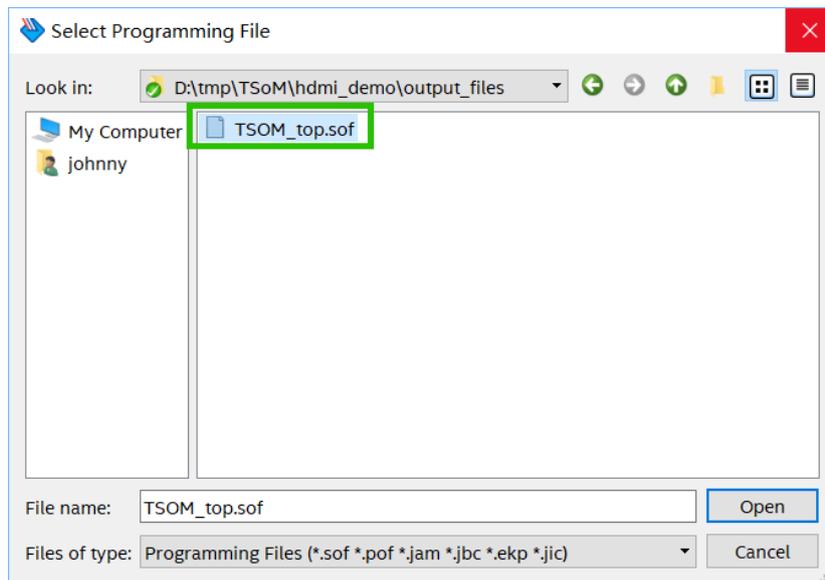
**Figure 3-11 FPGA and HPS detected in Quartus programmer**

Right click on the FPGA device and open the .sof file to be programmed, as highlighted in **Figure 3-12**.



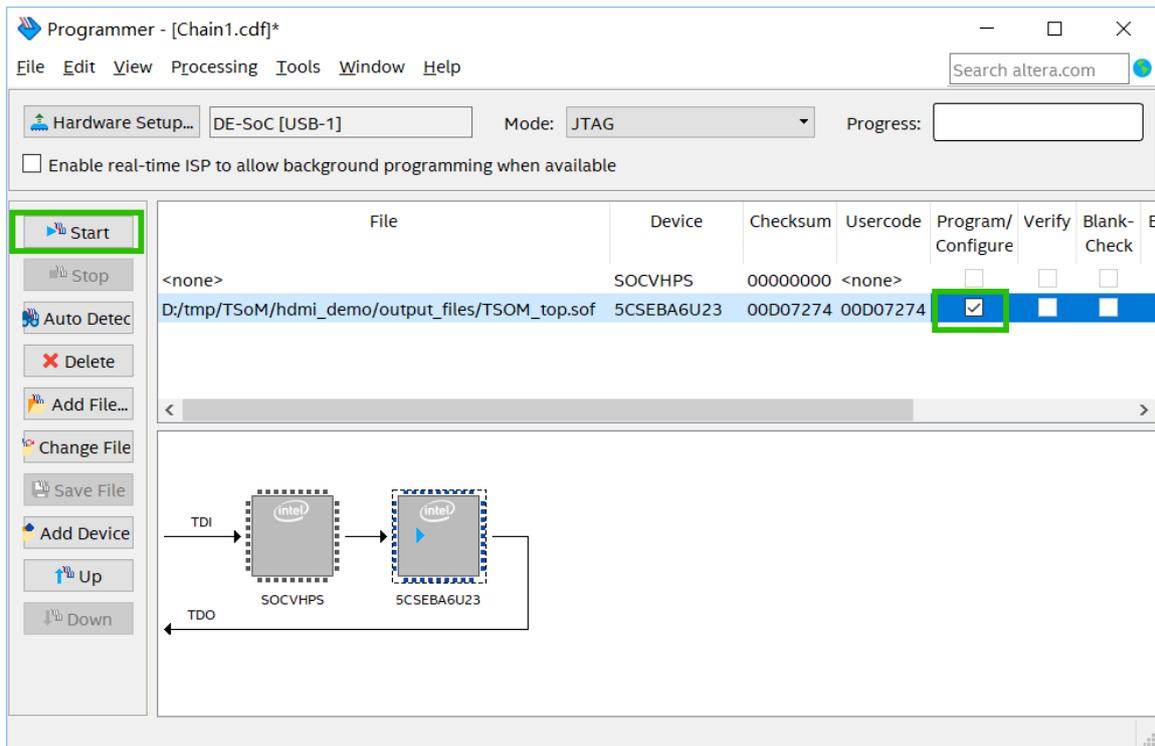
**Figure 3-12** Open the .sof file to be programmed into the FPGA device

Select the .sof file to be programmed, as shown in **Figure 3-13**.



**Figure 3-13** Select the .sof file to be programmed into the FPGA device

Click “**Program/Configure**” check box and then click “**Start**” button to download the .sof file into the FPGA device, as shown in **Figure 3-14**.

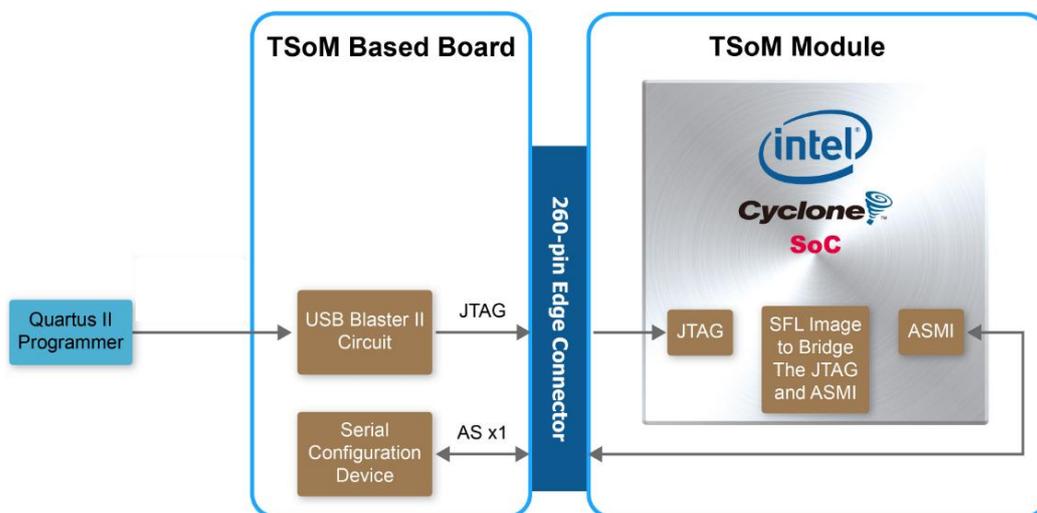


**Figure 3-14 Program .sof file into the FPGA device**

### ■ Configure the FPGA in AS Mode

The TSoM evaluation kit board uses a serial configuration device (EPCS64) to store configuration data for the Cyclone V SoC FPGA. This configuration data is automatically loaded from the serial configuration device chip into the FPGA when the board is powered up.

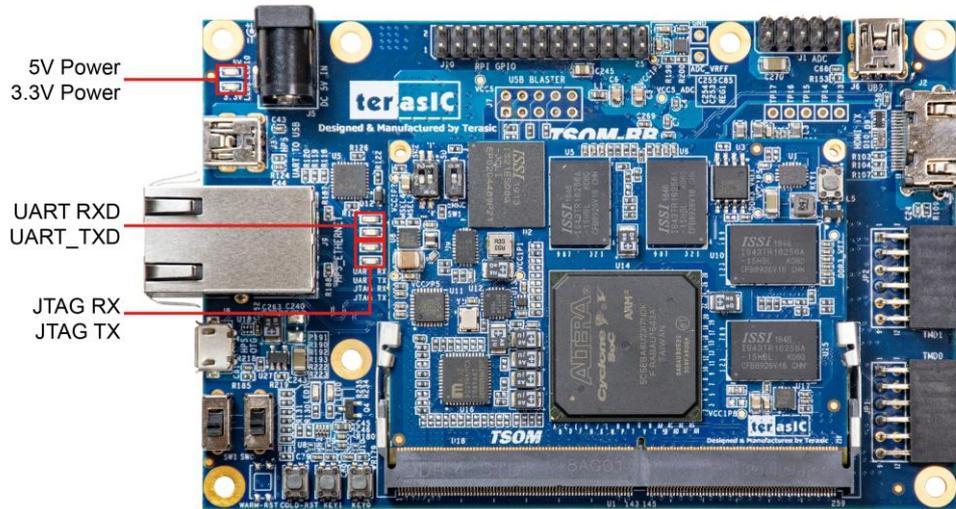
Users need to use Serial Flash Loader (SFL) to program the serial configuration device via JTAG interface. The FPGA-based SFL is a soft intellectual property (IP) core within the FPGA that bridges the JTAG and Flash interfaces. The SFL Megafunction is available in Quartus II. **Figure 3-15** shows the programming method when adopting SFL solution.



**Figure 3-15 Programming a serial configuration device with SFL solution**

### 3.3 Board Status Elements

In addition to the 9 LEDs that FPGA/HPS device can control, there are 6 indicators which can indicate the board status (See [Figure 3-16](#)), please refer the details in [Table 3-1](#).



**Figure 3-16 LED Indicators on TSoM evaluation kit**

**Table 3-1 LED Indicators**

| Board Reference | LED Name    | Description  |
|-----------------|-------------|--|
| LED2            | 3.3-V Power | Illuminate when 3.3V power is active.                        |
| LED10           | 5-V Power   | Illuminate when 5V power is active.                          |
| LED4            | JTAG_TX     | Illuminate when data is transferred from JTAG to USB Host.   |
| LED5            | JTAG_RX     | Illuminate when data is transferred from USB Host to JTAG.   |
| TXD1            | UART TXD    | Illuminate when data is transferred from FT232R to USB Host. |
| RXD1            | UART RXD    | Illuminate when data is transferred from USB Host to FT232R. |

### 3.4 Board Reset Elements

There are two HPS reset buttons on TSoM evaluation kit, HPS (cold) reset and HPS warm reset, as shown in [Figure 3-17](#). [Table 3-2](#) describes the purpose of these two HPS reset buttons. [Figure 3-18](#) is the reset tree for TSoM evaluation kit.

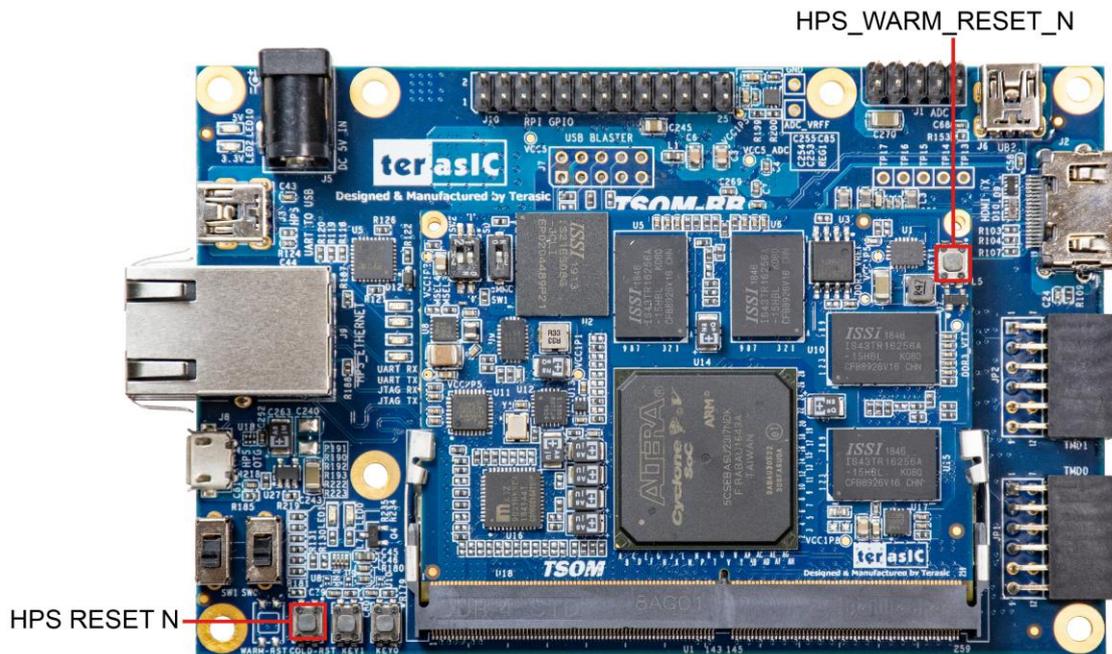


Figure 3-17 HPS cold and warm reset buttons on TSoM evaluation kit

Table 3-2 Description of Two HPS Reset Buttons on TSoM evaluation kit

| Board Reference           | Signal Name    | Description  |
|---------------------------|----------------|--|
| Button1(TSoM Based board) | HPS_RESET_N    | Cold reset to the HPS, Ethernet PHY and USB host device. Active low input which resets all HPS logics that can be reset. |
| KEY1 (TSoM Module board)  | HPS_WARM_RST_N | Warm reset to the HPS block. Active low input affects the system reset domain for debug purpose.                         |

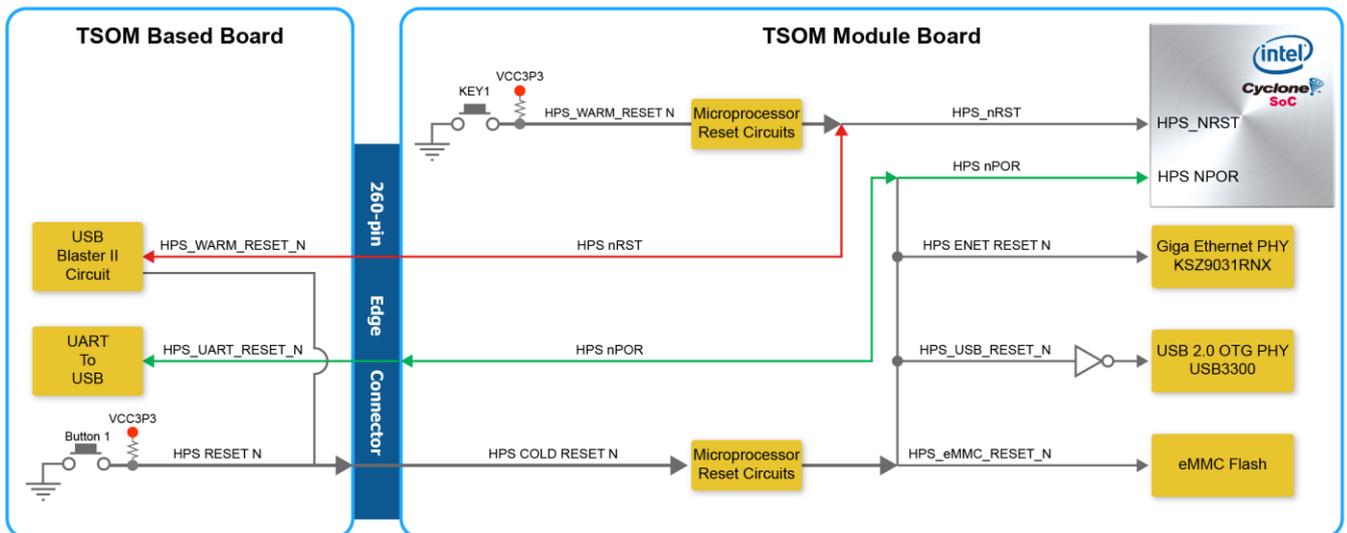


Figure 3-18 HPS reset tree on TSoM evaluation kit board

# Chapter 4

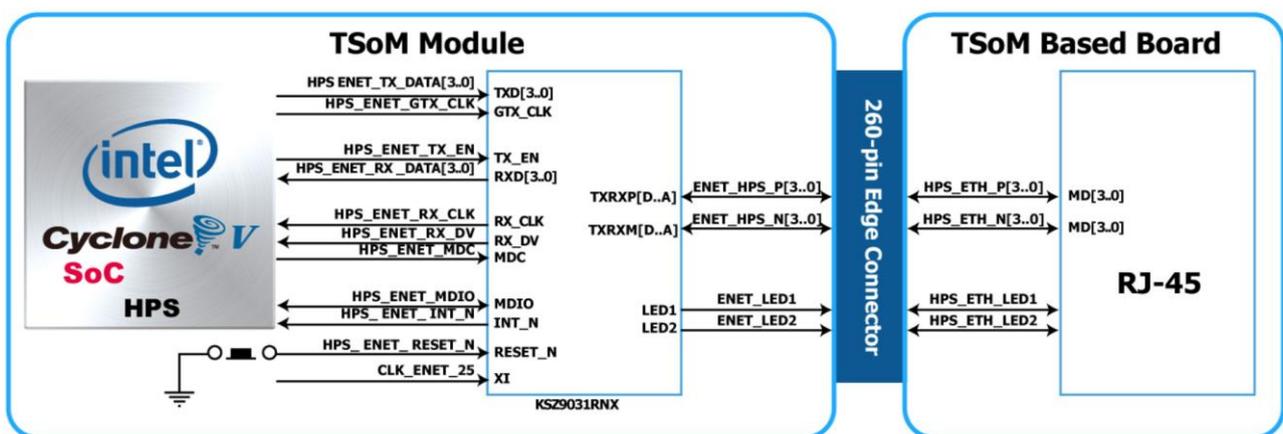
## *HPS Fabric Components*

This chapter describes the interfaces connected to TSoM Based Board. Users can access these interfaces via the HPS processor on the TSoM module.

### 4.1 Ethernet RJ45

The board supports a RJ-45 connector, which is provided by the TSoM Based Board. **Figure 4-1** shows the connections between the TSoM module, 260 Pin edge connector and RJ-45 connector.

The pin assignment of Ethernet PHY and HPS is listed in **Table 4-1**.



**Figure 4-1** The connections between the TSoM module, 260 Pin edge connector and RJ-45 connector

**Table 4-1** The pin assignment of Ethernet PHY and HPS on the TSoM module

| Signal Name         | FPGA Pin No. | Description                     | I/O Standard |
|---------------------|--------------|---------------------------------|--------------|
| HPS_ENET_TX_EN      | A12          | GMII and MII transmit enable    | 3.3V         |
| HPS_ENET_TX_DATA[0] | A16          | MII transmit data[0]            | 3.3V         |
| HPS_ENET_TX_DATA[1] | J14          | MII transmit data[1]            | 3.3V         |
| HPS_ENET_TX_DATA[2] | A15          | MII transmit data[2]            | 3.3V         |
| HPS_ENET_TX_DATA[3] | D17          | MII transmit data[3]            | 3.3V         |
| HPS_ENET_RX_DV      | J13          | GMII and MII receive data valid | 3.3V         |
| HPS_ENET_RX_DATA[0] | A14          | GMII and MII receive data[0]    | 3.3V         |
| HPS_ENET_RX_DATA[1] | A11          | GMII and MII receive data[1]    | 3.3V         |
| HPS_ENET_RX_DATA[2] | C15          | GMII and MII receive data[2]    | 3.3V         |
| HPS_ENET_RX_DATA[3] | A9           | GMII and MII receive data[3]    | 3.3V         |
| HPS_ENET_RX_CLK     | J12          | GMII and MII receive clock      | 3.3V         |
| HPS_ENET_MDIO       | E16          | Management Data                 | 3.3V         |

|                  |     |                             |            |      |
|------------------|-----|-----------------------------|------------|------|
| HPS_ENET_MDC     | A13 | Management Reference        | Data Clock | 3.3V |
| HPS_ENET_INT_N   | B14 | Interrupt Open Drain Output |            | 3.3V |
| HPS_ENET_GTX_CLK | J15 | GMIIT Transmit Clock        |            | 3.3V |

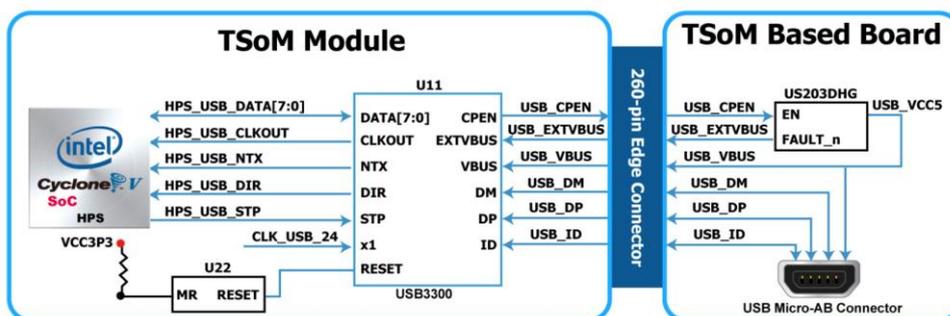
There are two LEDs, green LED (LEDG) and yellow LED (LEDY), which represent the status of Ethernet PHY (KSZ9031RN). The LED control signals are connected to the LEDs on the RJ-45 connector. The state and definition of LEDG and LEDY are listed in **Table 4-2**. For instance, the connection from board to Gigabit Ethernet is established once the LEDG lights on.

**Table 4-2 State and Definition of LED Mode Pins**

| LED (State) |        | LED (Definition) |          | Link /Activity                |
|-------------|--------|------------------|----------|-------------------------------|
| LEDG        | LEDY   | LEDG             | LEDY     |                               |
| H           | H      | OFF              | OFF      | Link off                      |
| L           | H      | ON               | OFF      | 1000 Link / No Activity       |
| Toggle      | H      | Blinking         | OFF      | 1000 Link / Activity (RX, TX) |
| H           | L      | OFF              | ON       | 100 Link / No Activity        |
| H           | Toggle | OFF              | Blinking | 100 Link / Activity (RX, TX)  |
| L           | L      | ON               | ON       | 10 Link/ No Activity          |
| Toggle      | Toggle | Blinking         | Blinking | Link / Activity (RX, TX)      |

## 4.2 USB OTG Connector

The board has an USB interfaces (using the SMSC USB3300 controller), which is provided by TSoM Based Board. A SMSC USB3300 device in a 32-pin QFN package device is used to interface to a single Type AB Micro-USB connector. This device supports UTMI+ Low Pin Interface (ULPI) to communicate to USB 2.0 controller in HPS. As defined by OTG mode, the PHY can operate in Host or Device modes. When operating in Host mode, the interface will supply the power to the device through the Micro-USB interface. **Figure 4-2** shows the connections between the TSoM module, 260 Pin edge connector and RJ-45 connector. **Table 4-3** lists the pin assignment of USB OTG PHY to HPS.



**Figure 4-2 The connections between the TSoM module, 260 Pin edge connector and USB OTG interface**

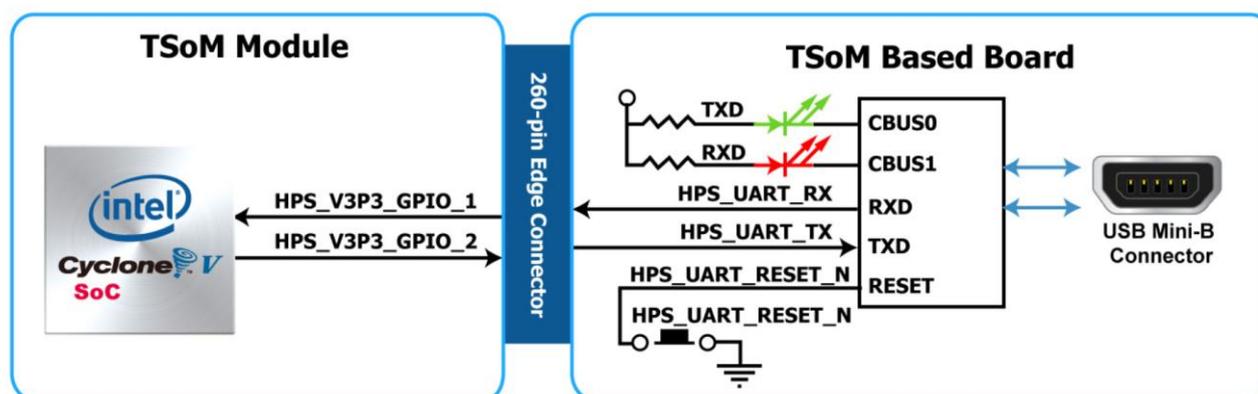
**Table 4-3 The pin assignment of USB OTG PHY to HPS on the TsoM module**

| Signal Name     | HPS Pin No. | Description                 | I/O Standard |
|-----------------|-------------|-----------------------------|--------------|
| HPS_USB_CLKOUT  | G4          | Reference Clock Output      | 3.3V         |
| HPS_USB_DATA[1] | C10         | HPS_USB_DATA[1]             | 3.3V         |
| HPS_USB_DATA[1] | F5          | HPS_USB_DATA[1]             | 3.3V         |
| HPS_USB_DATA[2] | C9          | HPS_USB_DATA[2]             | 3.3V         |
| HPS_USB_DATA[3] | C4          | HPS_USB_DATA[3]             | 3.3V         |
| HPS_USB_DATA[4] | C8          | HPS_USB_DATA[4]             | 3.3V         |
| HPS_USB_DATA[5] | D4          | HPS_USB_DATA[5]             | 3.3V         |
| HPS_USB_DATA[6] | C7          | HPS_USB_DATA[6]             | 3.3V         |
| HPS_USB_DATA[7] | F4          | HPS_USB_DATA[7]             | 3.3V         |
| HPS_USB_DIR     | E5          | Direction of the Data Bus   | 3.3V         |
| HPS_USB_NXT     | D5          | Throttle the Data           | 3.3V         |
| HPS_USB_STP     | C5          | Stop Data Stream on the Bus | 3.3V         |

### 4.3 UART to USB

The board has one UART interface connected for communication with the HPS on the TsoM module. This interface doesn't support HW flow control signals. The physical interface is implemented by UART-USB onboard bridge from a FT232R chip to the host with an USB Mini-B connector. More information about the chip is available on the manufacturer's website, or in the directory \Datasheets\UART\_TO\_USB of TSOM-BB system CD. **Figure 4-3** shows the connections between the HPS, 260-pin FT232R chip, and the USB Mini-B connector.

**Table 4-4** lists the pin assignment of UART interface connected to the HPS of TsoM module.



**Figure 4-3 Connections between the HPS and FT232R Chip**

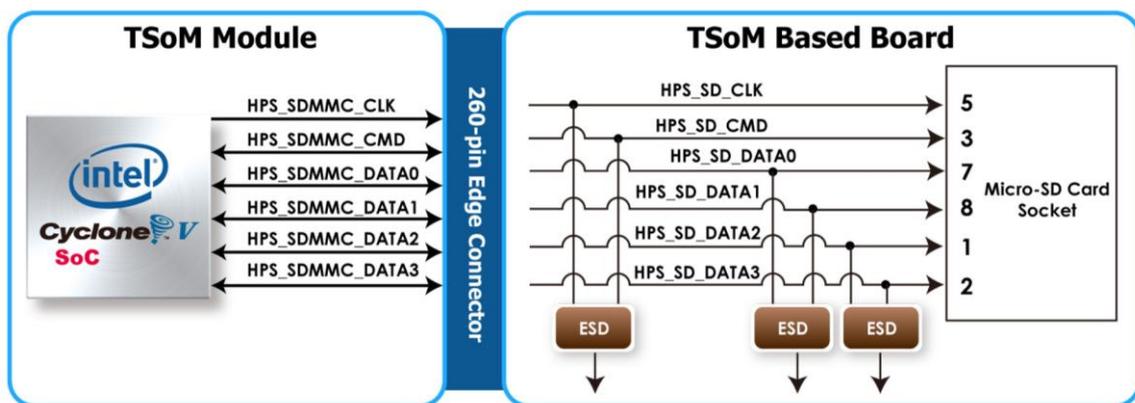
**Table 4-4 Pin Assignment of UART Interface**

| Signal Name | 260-pin Edge Connector Pin No. | FPGA Pin No. | Description          | I/O Standard |
|-------------|--------------------------------|--------------|----------------------|--------------|
| HPS_UART_RX | PIN_63                         | PIN_A22      | HPS UART Receiver    | 3.3V         |
| HPS_UART_TX | PIN_65                         | PIN_B21      | HPS UART Transmitter | 3.3V         |

## 4.4 Micro SD Card

The board supports Micro SD card interface with x4 data lines. It serves not only an external storage for the HPS on the TsoM module, but also an alternative boot option for the board. **Figure 4-4** shows signals connected between the HPS, 260 Pin edge connector and Micro SD card socket.

**Table 4-5** lists the pin assignment of Micro SD card socket to the HPS.



**Figure 4-4 Connections between the HPS, 260 Pin edge connector and and SD card socket**

**Table 4-5 Pin Assignment of Micro SD Card Socket**

| Signal Name    | 260-pin Edge Connector Pin No. | FPGA Pin No | Description         | I/O Standard |
|----------------|--------------------------------|-------------|---------------------|--------------|
| HPS_SD_CLK     | PIN_113                        | PIN_B8      | HPS SD Clock        | 3.3V         |
| HPS_SD_CMD     | PIN_111                        | PIN_D14     | HPS SD Command Line | 3.3V         |
| HPS_SD_DATA[0] | PIN_115                        | PIN_C13     | HPS SD Data[0]      | 3.3V         |
| HPS_SD_DATA[1] | PIN_117                        | PIN_B6      | HPS SD Data[1]      | 3.3V         |
| HPS_SD_DATA[2] | PIN_119                        | PIN_B11     | HPS SD Data[2]      | 3.3V         |
| HPS_SD_DATA[3] | PIN_121                        | PIN_B9      | HPS SD Data[3]      | 3.3V         |

## 4.5 QSPI Flash (DNI)

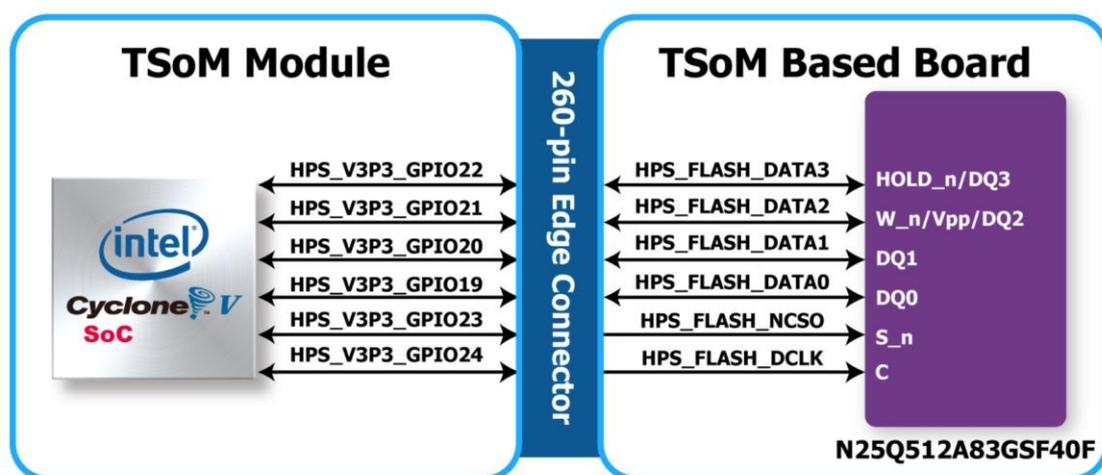
The board supports a 512M-bit serial NOR flash device (which is not installed (DNI)) for

non-volatile storage of HPS boot code, user data and program. The device is connected to HPS dedicated interface. It may contain secondary boot code.

This device has a 4-bit data interface and uses 3.3V CMOS signaling standard. Connections between Cyclone V SoC FPGA and Flash are shown in **Figure 4-5**.

To program the QSPI flash, the HPS Flash Programmer is provided both as part of the Intel Quartus Prime suite and as part of the free Intel Quartus Prime Programmer. The HPS Flash Programmer sends file contents over an Intel download cable, such as the USB Blaster II, to the HPS, and instructs the HPS to write the data to the flash memory.

Below summarizes the pins on the flash device. Signal names are from the device datasheet and directions are relative to the Cyclone V SoC FPGA.



**Figure 4-5** Connections between the HPS, 260 Pin edge connector and QSPI Flash

**Table 4-6** Pin Assignment of QSPI Flash

| Signal Name     | 260-pin Edge Connector Pin No. | FPGA Pin No | Description           | I/O Standard |
|-----------------|--------------------------------|-------------|-----------------------|--------------|
| HPS_FLASH_DATA0 | PIN_99                         | PIN_A8      | HPS FLASH Data0       | 3.3V         |
| HPS_FLASH_DATA1 | PIN_101                        | PIN_H16     | HPS FLASH Data1       | 3.3V         |
| HPS_FLASH_DATA2 | PIN_103                        | PIN_A7      | HPS FLASH Data2       | 3.3V         |
| HPS_FLASH_DATA3 | PIN_105                        | PIN_J16     | HPS FLASH Data3       | 3.3V         |
| HPS_FLASH_DCLK  | PIN_109                        | PIN_C14     | HPS FLASH Data Clock  | 3.3V         |
| HPS_FLASH_NCSO  | PIN_107                        | PIN_A6      | HPS FLASH Chip Enable | 3.3V         |

## 4.6 Raspberry Pi 2x13 GPIO

The board provides one Raspberry Pi 2x13 expansion header, which has 17 pins connected directly to the Cyclone V SoC HPS. It also comes with It also comes with two DC +5V (VCC5), two DC +3.3V (VCC3P3), and five GND pins. **Figure 4-6** and **Figure 4-7** shows I/O distribution of the

Raspberry Pi header. In addition to being used as the HPS GPIO, user can also use the corresponding peripheral controller in the HPS such as the SPI and UART interfaces. [Table 4-7](#) shows all the pin assignments of the Raspberry Pi 2x13 GPIO header.

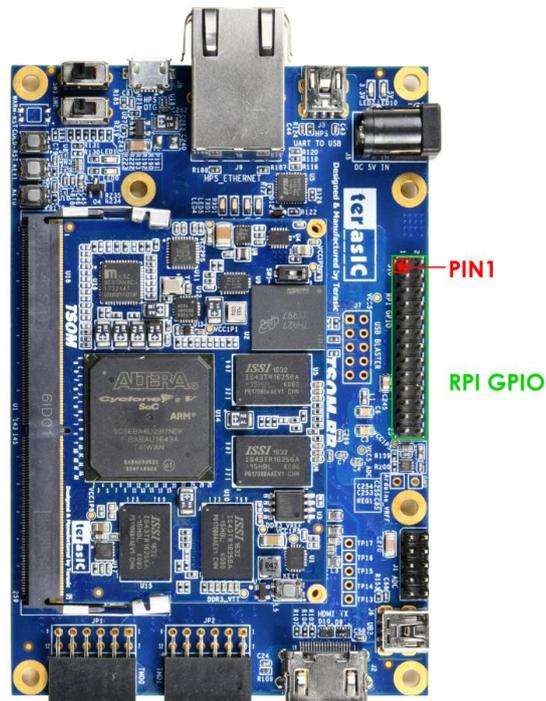


Figure 4-6 Raspberry Pi 2x13 expansion header



|            |        |         |                       |      |
|------------|--------|---------|-----------------------|------|
| RPI_GPIO17 | PIN_61 | PIN_C21 | RPI GPIO Connection 0 | 3.3V |
| RPI_GPIO18 | PIN_69 | PIN_K18 | RPI GPIO Connection 0 | 3.3V |
| RPI_GPIO22 | PIN_73 | PIN_J18 | RPI GPIO Connection 0 | 3.3V |
| RPI_GPIO23 | PIN_95 | PIN_B19 | RPI GPIO Connection 0 | 3.3V |
| RPI_GPIO24 | PIN_97 | PIN_C16 | RPI GPIO Connection 0 | 3.3V |
| RPI_GPIO25 | PIN_89 | PIN_H17 | RPI GPIO Connection 0 | 3.3V |
| RPI_GPIO27 | PIN_71 | PIN_A20 | RPI GPIO Connection 0 | 3.3V |

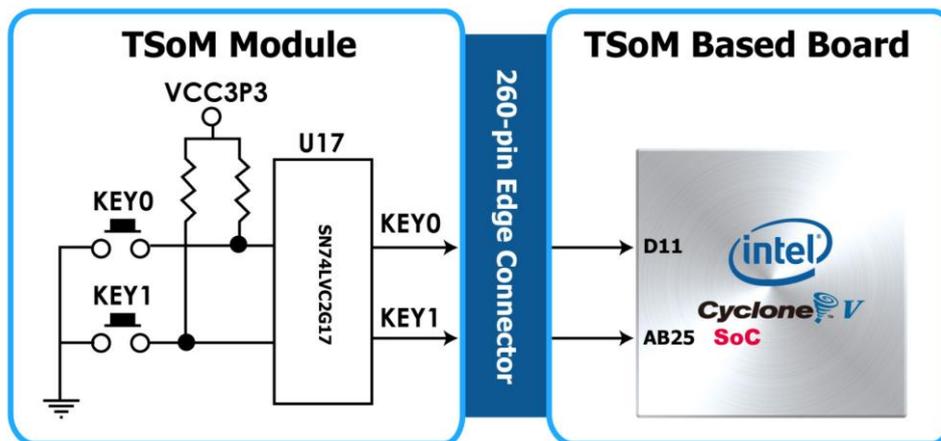
# Chapter 5

## FPGA Fabric Components

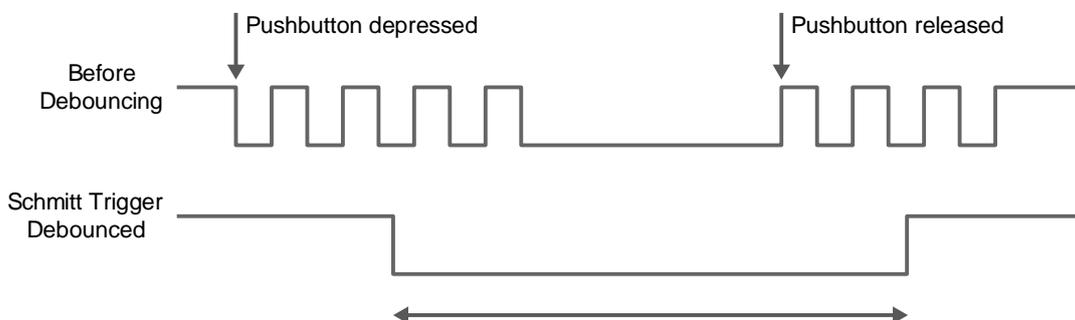
This section describes the interfaces connected to the FPGA. Users can control or monitor different interfaces with user logic from the FPGA.

### 5.1 User Push-buttons, Switches and LEDs

The TSoM Based Board has two push-buttons connected to the FPGA, as shown in **Figure 5-1**. Schmitt trigger circuit is implemented and act as switch debounce in **Figure 5-2** for the push-buttons connected. The two push-buttons named KEY0 and KEY1 coming out of the Schmitt trigger device are connected directly to the Cyclone V SoC FPGA. The push-button generates a low logic level or high logic level when it is pressed or not, respectively. Since the push-buttons are debounced, they can be used as clock or reset inputs in a circuit.



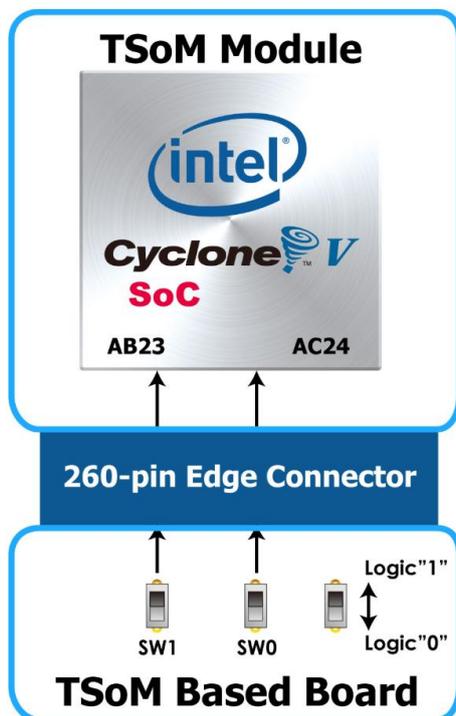
**Figure 5-1** Connections between the push-buttons and the Cyclone V SoC FPGA



**Figure 5-2** Switch debouncing

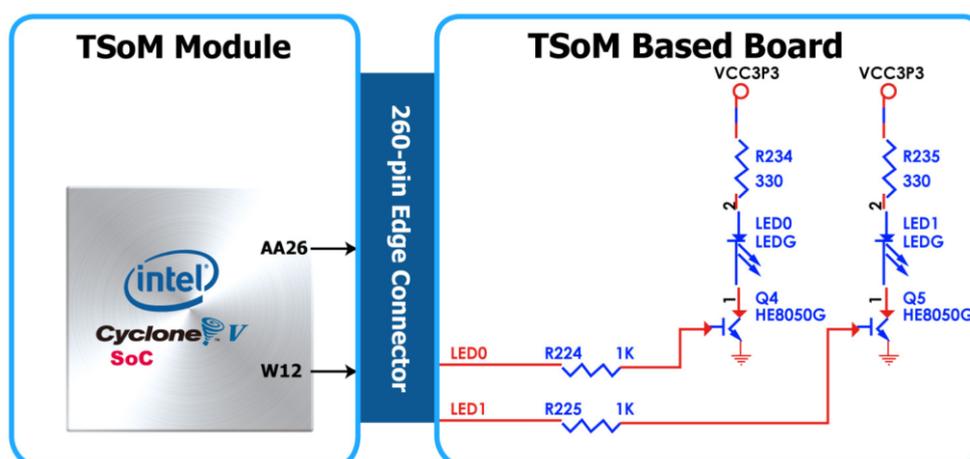
There are two slide switches connected to the FPGA, as shown in **Figure 5-3**. These switches are not debounced and to be used as level-sensitive data inputs to a circuit. Each switch is

connected directly and individually to the FPGA. When the switch is set to the DOWN position (towards the edge of the board), it generates a low logic level to the FPGA. When the switch is set to the UP position, a high logic level is generated to the FPGA.



**Figure 5-3 Connections between the slide switches and the Cyclone V SoC FPGA**

There are also two user-controllable LEDs connected to the FPGA. Each LED is driven directly and individually by the Cyclone V SoC FPGA; driving its associated pin to a high logic level or low level to turn the LED on or off, respectively. **Figure 5-4** shows the connections between LEDs and Cyclone V SoC FPGA. **Table 5-1**, **Table 5-2** and **Table 5-3** list the pin assignment of user push-buttons, switches, and LEDs.



**Figure 5-4 Connections between the LEDs and the Cyclone V SoC FPGA**

**Table 5-1 Pin Assignment of Slide Switches**

| Signal Name | 260-pin Edge Connector Pin No. | FPGA Pin No. | Description     | I/O Standard |
|-------------|--------------------------------|--------------|-----------------|--------------|
| SW0         | PIN_112                        | PIN_AC24     | Slide Switch[0] | 3.3V         |
| SW1         | PIN_114                        | PIN_AB23     | Slide Switch[1] | 3.3V         |

**Table 5-2 Pin Assignment of Push-buttons**

| Signal Name | 260-pin Edge Connector Pin No. | FPGA Pin No. | Description    | I/O Standard |
|-------------|--------------------------------|--------------|----------------|--------------|
| KEY0        | PIN_37                         | PIN_D11      | Push-button[0] | 3.3V         |
| KEY1        | PIN_39                         | PIN_AB25     | Push-button[1] | 3.3V         |

**Table 5-3 Pin Assignment of LEDs**

| Signal Name | 260-pin Edge Connector Pin No. | FPGA Pin No. | Description | I/O Standard |
|-------------|--------------------------------|--------------|-------------|--------------|
| LED0        | PIN_205                        | PIN_AA26     | LED [0]     | 3.3V         |
| LED1        | PIN_206                        | PIN_W12      | LED [1]     | 3.3V         |

## 5.2 TMD Header

The board has two TMD connectors, each connector has 8 user pins connected to the 260-pin Edge Connector (to Cyclone V SoC FPGA). It also comes with DC +3.3V (VCC3P3), and two GND pins. **Figure 5-5** shows the I/O distribution of the TMD connector. The maximum power consumption allowed for a daughter card connected to one or two TMD ports is shown in **Table 5-4** and **Table 5-5**, shows all the pin assignments of the TMD connectors.





**Figure 5-5 TMD Pin Arrangement**

**Table 5-4 Voltage and Max. Current Limit of TMD Connector(s)**

| Supplied Voltage | Max. Current Limit |
|------------------|--------------------|
| 3.3V             | 1.5A               |

**Table 5-5 all Pin Assignment of Expansion Headers**

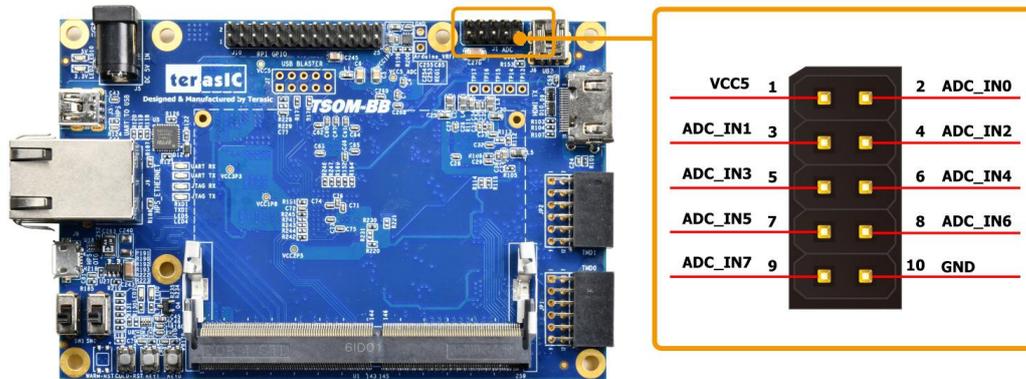
| Signal Name | 260-pin Edge Connector Pin No. | FPGA Pin No. | Description | I/O Standard |
|-------------|--------------------------------|--------------|-------------|--------------|
| TMD0_IO0    | PIN_239                        | PIN_AH6      | TMD0 IO[0]  | 3.3V         |
| TMD0_IO1    | PIN_241                        | PIN_AH5      | TMD0 IO[1]  | 3.3V         |
| TMD0_IO2    | PIN_245                        | PIN_AF7      | TMD0 IO[2]  | 3.3V         |
| TMD0_IO3    | PIN_247                        | PIN_AH2      | TMD0 IO[3]  | 3.3V         |
| TMD0_IO4    | PIN_251                        | PIN_AE8      | TMD0 IO[4]  | 3.3V         |
| TMD0_IO5    | PIN_253                        | PIN_AF9      | TMD0 IO[5]  | 3.3V         |
| TMD0_IO6    | PIN_257                        | PIN_AG5      | TMD0 IO[6]  | 3.3V         |
| TMD0_IO7    | PIN_259                        | PIN_AH4      | TMD0 IO[7]  | 3.3V         |
| TMD1_IO0    | PIN_240                        | PIN_AF11     | TMD1 IO[0]  | 3.3V         |
| TMD1_IO1    | PIN_242                        | PIN_AF10     | TMD1 IO[1]  | 3.3V         |
| TMD1_IO2    | PIN_246                        | PIN_AE12     | TMD1 IO[2]  | 3.3V         |
| TMD1_IO3    | PIN_248                        | PIN_AD12     | TMD1 IO[3]  | 3.3V         |
| TMD1_IO4    | PIN_252                        | PIN_AD10     | TMD1 IO[4]  | 3.3V         |
| TMD1_IO5    | PIN_254                        | PIN_AE9      | TMD1 IO[5]  | 3.3V         |
| TMD1_IO6    | PIN_258                        | PIN_V11      | TMD1 IO[6]  | 3.3V         |
| TMD1_IO7    | PIN_260                        | PIN_W11      | TMD1 IO[7]  | 3.3V         |

## 5.3 A/D Converter and Analog Input

The TSoM Based Board has an analog-to-digital converter (LTC2308).

The LTC2308 is a low noise, 500ksps, 8-channel, 12-bit ADC with a SPI/MICROWIRE compatible serial interface. This ADC includes an internal reference and a fully differential sample-and-hold circuit to reduce common mode noise. The internal conversion clock allows the external serial output data clock (SCK) to operate at any frequency up to 40MHz.

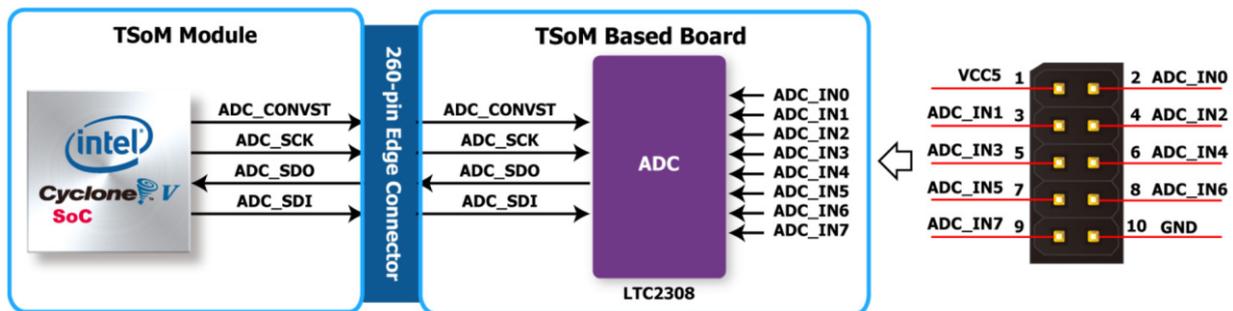
It can be configured to accept eight input signals at inputs ADC\_IN0 through ADC\_IN7. These eight input signals are connected to a 2x5 header, as shown in **Figure 5-6**.



**Figure 5-6 Signals of the 2x5 Header**

These Analog inputs are shared with the Arduino's analog input pin (ADC\_IN0 ~ ADC\_IN5), **Figure 5-7** shows the connections between the FPGA, 2x5 header, Arduino Analog input, and the A/D converter.

More information about the A/D converter chip is available in its datasheet. It can be found on manufacturer's website or in the directory \Datasheet\ADC of TSoM Based Board system CD.



**Figure 5-7 Connections between the FPGA, 2x5 header, and the A/D converter**

**Table 5-6 Pin Assignment of ADC**

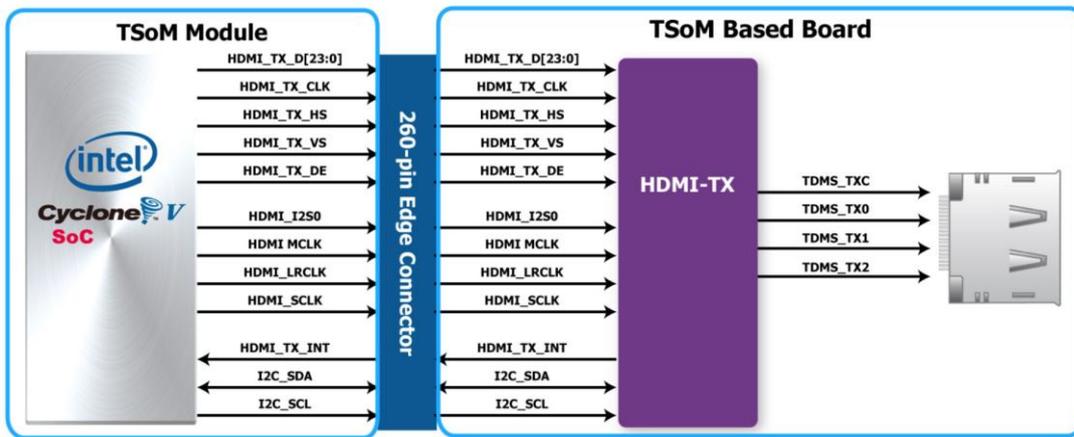
| Signal Name | 260-pin Edge Connector Pin No. | FPGA Pin No. | Description                     | I/O Standard |
|-------------|--------------------------------|--------------|---------------------------------|--------------|
| ADC_CONVST  | PIN_84                         | PIN_Y19      | Conversion Start                | 3.3V         |
| ADC_SCK     | PIN_82                         | PIN_AA20     | Serial Data Clock               | 3.3V         |
| ADC_SDI     | PIN_96                         | PIN_AA13     | Serial Data Input (FPGA to ADC) | 3.3V         |

|         |        |         |                                  |      |
|---------|--------|---------|----------------------------------|------|
| ADC_SDO | PIN_94 | PIN_Y13 | Serial Data Out<br>(ADC to FPGA) | 3.3V |
|---------|--------|---------|----------------------------------|------|

## 5.4 HDMI TX Interface

The development board provides High Performance HDMI Transmitter via the Analog Devices ADV7513 which incorporates HDMI v1.4 features, including 3D video support, and 165 MHz supports all video formats up to 1080p and UXGA. The ADV7513 is controlled via a serial I2C bus interface, which is connected to pins on the Cyclone V SoC FPGA. A schematic diagram of the audio circuitry is shown in **Figure 5-8**. Detailed information on using the ADV7513 HDMI TX is available on the manufacturer’s website, or under the Datasheets\HDMI folder on the Kit System CD.

**Table 5-7** lists the HDMI Interface pin assignments and signal names relative to the Cyclone V SoC device.



**Figure 5-8** Connections between the Cyclone V SoC FPGA and HDMI Transmitter Chip

**Table 5-7** HDMI Pin Assignments, Schematic Signal Names, and Functions

| Signal Name | 260-pin Edge Connector Pin No. | FPGA Pin No. | Description    | I/O Standard |
|-------------|--------------------------------|--------------|----------------|--------------|
| HDMI_TX_D0  | PIN_221                        | PIN_AC4      | Video Data bus | 3.3-V        |
| HDMI_TX_D1  | PIN_223                        | PIN_AD4      | Video Data bus | 3.3-V        |
| HDMI_TX_D2  | PIN_227                        | PIN_AD5      | Video Data bus | 3.3-V        |
| HDMI_TX_D3  | PIN_229                        | PIN_AE6      | Video Data bus | 3.3-V        |
| HDMI_TX_D4  | PIN_233                        | PIN_AF5      | Video Data bus | 3.3-V        |
| HDMI_TX_D5  | PIN_235                        | PIN_AF6      | Video Data bus | 3.3-V        |
| HDMI_TX_D6  | PIN_185                        | PIN_AH3      | Video Data bus | 3.3-V        |
| HDMI_TX_D7  | PIN_187                        | PIN_AH2      | Video Data bus | 3.3-V        |
| HDMI_TX_D8  | PIN_191                        | PIN_AE4      | Video Data bus | 3.3-V        |
| HDMI_TX_D9  | PIN_193                        | PIN_AF4      | Video Data bus | 3.3-V        |
| HDMI_TX_D10 | PIN_197                        | PIN_AE7      | Video Data bus | 3.3-V        |

|             |         |          |   |       |
|-------------|---------|----------|---|-------|
| HDMI_TX_D11 | PIN_199 | PIN_AF8  | Video Data bus                              | 3.3-V |
| HDMI_TX_D12 | PIN_222 | PIN_U10  | Video Data bus                              | 3.3-V |
| HDMI_TX_D13 | PIN_224 | PIN_V10  | Video Data bus                              | 3.3-V |
| HDMI_TX_D14 | PIN_228 | PIN_Y11  | Video Data bus                              | 3.3-V |
| HDMI_TX_D15 | PIN_230 | PIN_AA11 | Video Data bus                              | 3.3-V |
| HDMI_TX_D16 | PIN_234 | PIN_AD11 | Video Data bus                              | 3.3-V |
| HDMI_TX_D17 | PIN_236 | PIN_AE11 | Video Data bus                              | 3.3-V |
| HDMI_TX_D18 | PIN_186 | PIN_D12  | Video Data bus                              | 3.3-V |
| HDMI_TX_D19 | PIN_188 | PIN_C12  | Video Data bus                              | 3.3-V |
| HDMI_TX_D20 | PIN_192 | PIN_T13  | Video Data bus                              | 3.3-V |
| HDMI_TX_D21 | PIN_194 | PIN_T12  | Video Data bus                              | 3.3-V |
| HDMI_TX_D22 | PIN_198 | PIN_T11  | Video Data bus                              | 3.3-V |
| HDMI_TX_D23 | PIN_200 | PIN_U11  | Video Data bus                              | 3.3-V |
| HDMI_TX_CLK | PIN_203 | PIN_AB26 | Video Clock                                 | 3.3-V |
| HDMI_TX_DE  | PIN_204 | PIN_V12  | Data Enable Signal<br>for Digital Video.    | 3.3-V |
| HDMI_TX_HS  | PIN_215 | PIN_Y5   | Horizontal<br>Synchronization               | 3.3-V |
| HDMI_TX_VS  | PIN_217 | PIN_Y4   | Vertical<br>Synchronization                 | 3.3-V |
| HDMI_TX_INT | PIN_181 | PIN_D8   | Interrupt Signal                            | 3.3-V |
| HDMI_I2S0   | PIN_211 | PIN_AB4  | I2S Channel 0<br>Audio Data Input           | 3.3-V |
| HDMI_MCLK   | PIN_209 | PIN_AA4  | Audio Reference<br>Clock Input              | 3.3-V |
| HDMI_LRCLK  | PIN_212 | PIN_Y8   | Audio Left/Right<br>Channel Signal<br>Input | 3.3-V |
| HDMI_SCLK   | PIN_210 | PIN_W8   | I2S Audio Clock<br>Input                    | 3.3-V |
| I2C_SCL     | PIN_218 | PIN_T8   | I2C Clock                                   | 3.3-V |
| I2C_SDA     | PIN_216 | PIN_U9   | I2C Data                                    | 3.3-V |

# Chapter 6

## *Examples For FPGA*

This chapter provides examples of advanced designs implemented by RTL or Qsys on the TSoM board. These reference designs cover the features of peripherals connected to the FPGA, such as A/D Converter. All the associated files can be found in the directory \Demonstrations\FPGA of TSoM System CD.

### ■ Installation of Demonstrations

Install the demonstrations on your computer:

Copy the folder Demonstrations to a local directory of your choice. It is important to make sure the path to your local directory contains NO space. Otherwise it will lead to error in Nios II.

Note Quartus Prime v17.1 or later is required for all TSoM demonstrations to support Cyclone V SoC device.

## 6.1 HDMI TX

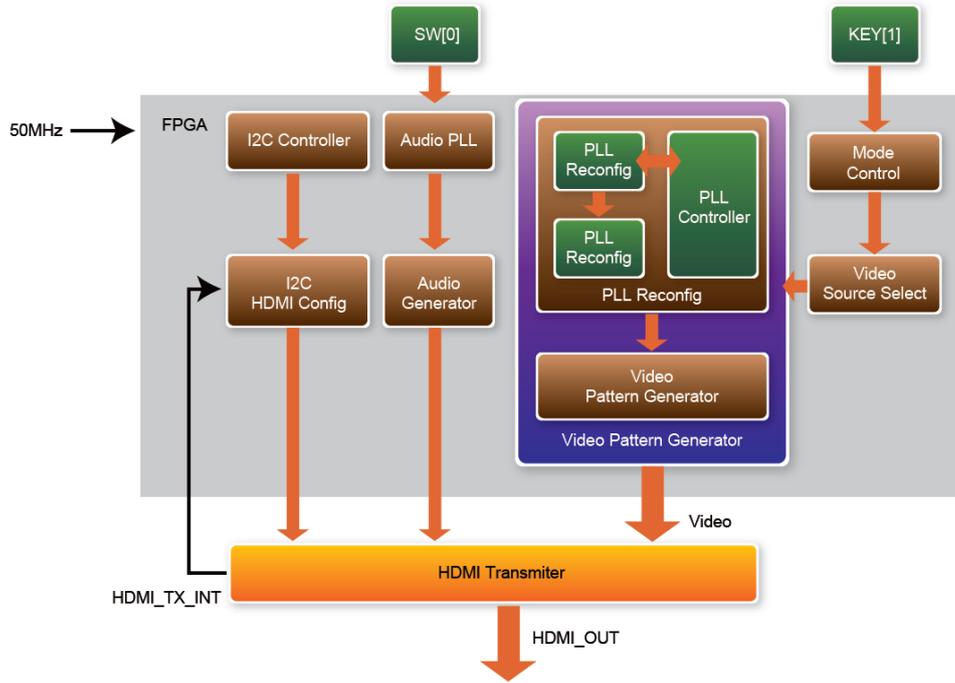
This section gives instructions to program the HDMI transmitter to generate video pattern and audio source. The entire reference is composed into three parts: video design, audio design, and I2C design. A set of built-in video patterns and audio serial data will be sent to the HDMI transmitter to drive the HDMI display with speaker. Users can hear the beeping sound from the speaker when SW0 is set to 1 on the TSoM board. The resolution can be switched by pressing KEY1.

### ■ System Block Diagram

**Figure 6-1** shows the system block diagram of this reference design. The HDMI Transmitter is configured via I2C interface by I2C Controller and I2C HDMI Config. It is necessary to configure the HDMI transmitter according to the desired settings.

An interrupt mechanism called Hot Plug Detect (HPD) is implemented in I2C HDMI config. to re-configure HDMI transmitter when HPD interrupt occurs.

The Video Patter Generator was designed to send video patter to HDMI transmitter. Its resolution can be changed by pressing KEY1. There must be a PLL Reconfig. To change the PLL dynamically to support different resolutions. The Audio PLL and Audio Generator were designed to send audio pattern to HDMI transmitter. The audio is transmitted via I2S interface in this demo. Users can choose to enable audio generator or not through SW0.



**Figure 6-1 Block Diagram of the HDMI TX Demonstration**

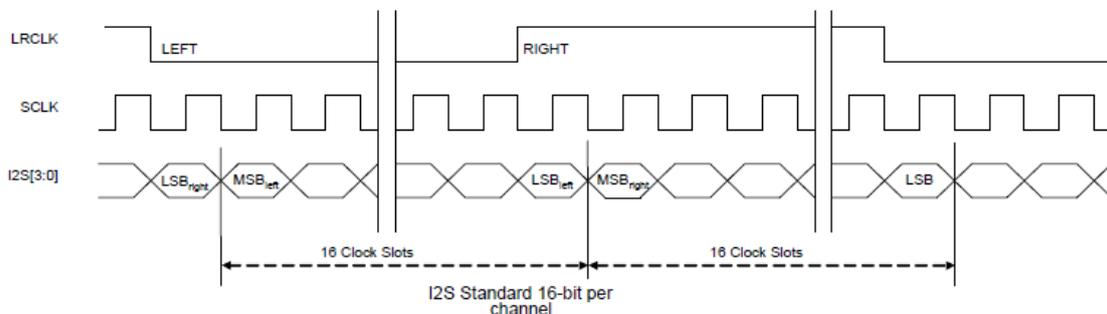
### ■ Register of HDMI Transmitter (ADV7513)

Users can save lots of developing time by paying attention to the settings of video format and audio frequency in register at address 0x15 and the format of register at address 0xAF prior to the development of HDMI transmitter. This demo uses 48KHz sampling rate and the video format is 24-bit RGB 4:4:4. For more details, please refer to the document ADV7513\_Programming\_Guide\_R0.pdf.

### ■ Audio Generator

The ADV7513 can accommodate 2 to 8 channels of I2S audio at up to a 192 KHz sampling rate. The ADV7513 supports I2S standard, left-justified serial audio, and right-justified serial audio.

**Figure 6-2** shows the left-justified serial audio with I2S standard audio of 16-bit per channel.



**Figure 6-2 I2S standard audio with 16-bit per channel**

If users want to modify the frequency of audio output, the register value at register 0x15 has to be modified according to the document ADV7513\_Programming\_Guide\_R0.pdf. The I2S standard

uses MSB to LSB serial way of transmitting. This demo uses sinusoid signal to synthesis sound from a reference of look up table created by calculated sinusoid wave data.

## ■ Video Pattern Generator

The module "Video Pattern Generator" copes with generating video patterns to be presented on the LCD monitor. The pattern is composed in the way of 24-bit RGB 4:4:4 (RGB888 per color pixel without sub-sampling) color encoding, which corresponds to the parallel encoding format defined in **Table 6-1** of the "ADV7513 Hardware User's Guide," as shown below.

**Table 6-1 Display modes of the HDMI TX demonstration**

| Pixel Data [23:0] |    |    |    |    |    |        |    |   |   |   |   |        |   |   |   |   |   |
|-------------------|----|----|----|----|----|--------|----|---|---|---|---|--------|---|---|---|---|---|
| 17                | 16 | 15 | 14 | 13 | 12 | 11     | 10 | 9 | 8 | 7 | 6 | 5      | 4 | 3 | 2 | 1 | 0 |
| R[7:0]            |    |    |    |    |    | G[7:0] |    |   |   |   |   | B[7:0] |   |   |   |   |   |

A set of display modes is implemented for presenting the generated video patterns. The module "Video Source Selector" controls the selection of current video timing among built-in display modes listed in **Table 6-2**. The module "Mode Control" allows users to switch current display mode alternatively via KEY1.

**Table 6-2 Built-in display modes for the HDMI TX demonstration**

| Pattern ID | Video Format  | PCLK (MHZ) |
|------------|---------------|------------|
| 0          | 640x480@60P   | 25         |
| 1          | 720x480@60P   | 27         |
| 2          | 1024x768@60P  | 65         |
| 3          | 1280x1024@60P | 108        |
| 4          | 1920x1080@60P | 148.5      |

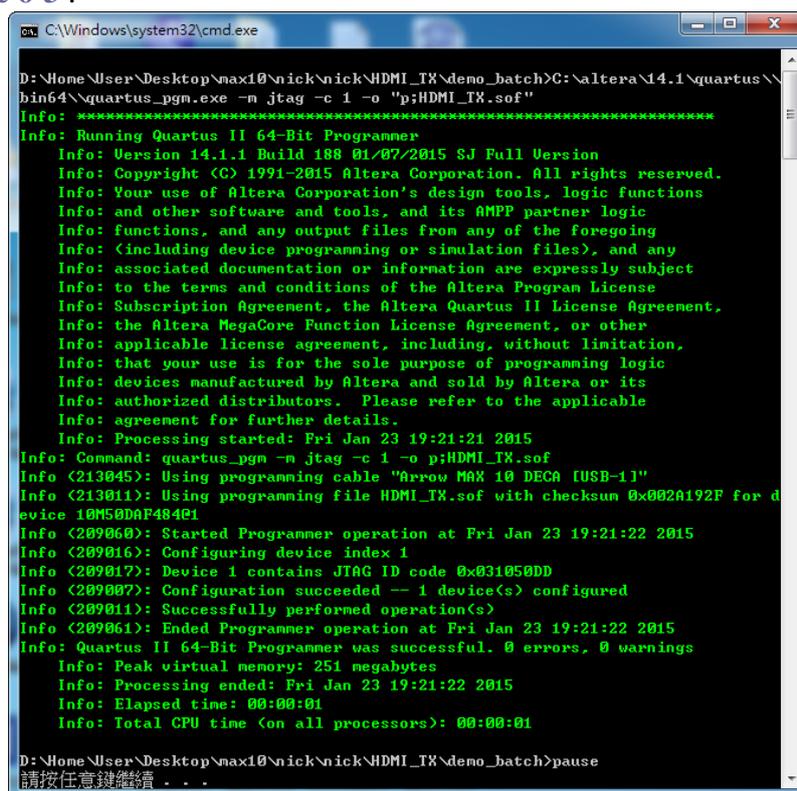
## ■ Demonstration File Locations

- Hardware project directory: \HDMI\_TX
- Bitstream used: DE10\_Nano\_HDMI\_TX.sof
- Demo batch file : \HDMI\_TX\demo\_batch\test.bat

## ■ Demonstration Setup and Instructions

- Please make sure both Quartus Prime and USB-Blaster II driver are installed on the host PC.
- Connect the TSoM board to the LCD monitor through a HDMI cable.
- Power on the TSoM board
- Launch the "test.bat" batch file from the "\HDMI\_TX\demo\_batch" folder. After the

programming and configuration are successful, the screen should look like the one shown in Figure 6-3.



```
C:\Windows\system32\cmd.exe
D:\None\User\Desktop\max10\nick\nick\HDMI_TX\demo_batch>C:\altera\14.1\quartus\
bin64\quartus_pgm.exe -m jtag -c 1 -o "p:HDMI_TX.sof"
Info: *****
Info: Running Quartus II 64-Bit Programmer
Info: Version 14.1.1 Build 188 01/07/2015 6J Full Version
Info: Copyright (C) 1991-2015 Altera Corporation. All rights reserved.
Info: Your use of Altera Corporation's design tools, logic functions
Info: and other software and tools, and its AMPP partner logic
Info: functions, and any output files from any of the foregoing
Info: (including device programming or simulation files), and any
Info: associated documentation or information are expressly subject
Info: to the terms and conditions of the Altera Program License
Info: Subscription Agreement, the Altera Quartus II License Agreement,
Info: the Altera MegaCore Function License Agreement, or other
Info: applicable license agreement, including, without limitation,
Info: that your use is for the sole purpose of programming logic
Info: devices manufactured by Altera and sold by Altera or its
Info: authorized distributors. Please refer to the applicable
Info: agreement for further details.
Info: Processing started: Fri Jan 23 19:21:21 2015
Info: Command: quartus_pgm -m jtag -c 1 -o p:HDMI_TX.sof
Info (213045): Using programming cable "Arrow MAX 10 DECA [USB-1]"
Info (213011): Using programming file HDMI_TX.sof with checksum 0x002A192F for d
evice 10M50DAF48401
Info (209060): Started Programmer operation at Fri Jan 23 19:21:22 2015
Info (209016): Configuring device index 1
Info (209017): Device 1 contains JTAG ID code 0x031050DD
Info (209007): Configuration succeeded -- 1 device(s) configured
Info (209011): Successfully performed operation(s)
Info (209061): Ended Programmer operation at Fri Jan 23 19:21:22 2015
Info: Quartus II 64-Bit Programmer was successful. 0 errors, 0 warnings
Info: Peak virtual memory: 251 megabytes
Info: Processing ended: Fri Jan 23 19:21:22 2015
Info: Elapsed time: 00:00:01
Info: Total CPU time (on all processors): 00:00:01
D:\None\User\Desktop\max10\nick\nick\HDMI_TX\demo_batch>pause
請按任意鍵繼續 . . .
```

Figure 6-3 Launch the HDMI TX demonstration from the "demo\_batch" folder

Wait for a few seconds for the LCD monitor to be powered up. There will be a pre-defined video pattern shown on the monitor, as shown in Figure 6-4. The SW0 is used to enable/disable the sound output on the TSoM board. When you switch the SW0 button to an upper position, you will hear a "beep" sound from the speaker of the HDMI display.

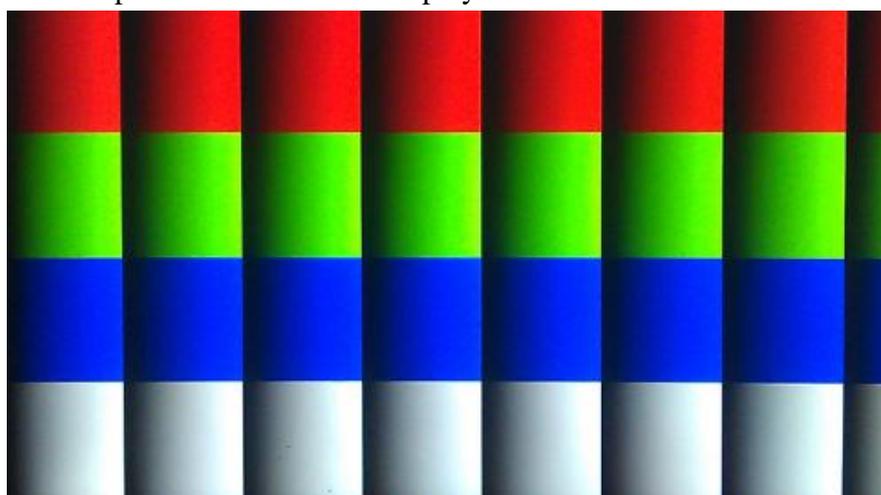


Figure 6-4 The video pattern in the HDMI TX demonstration

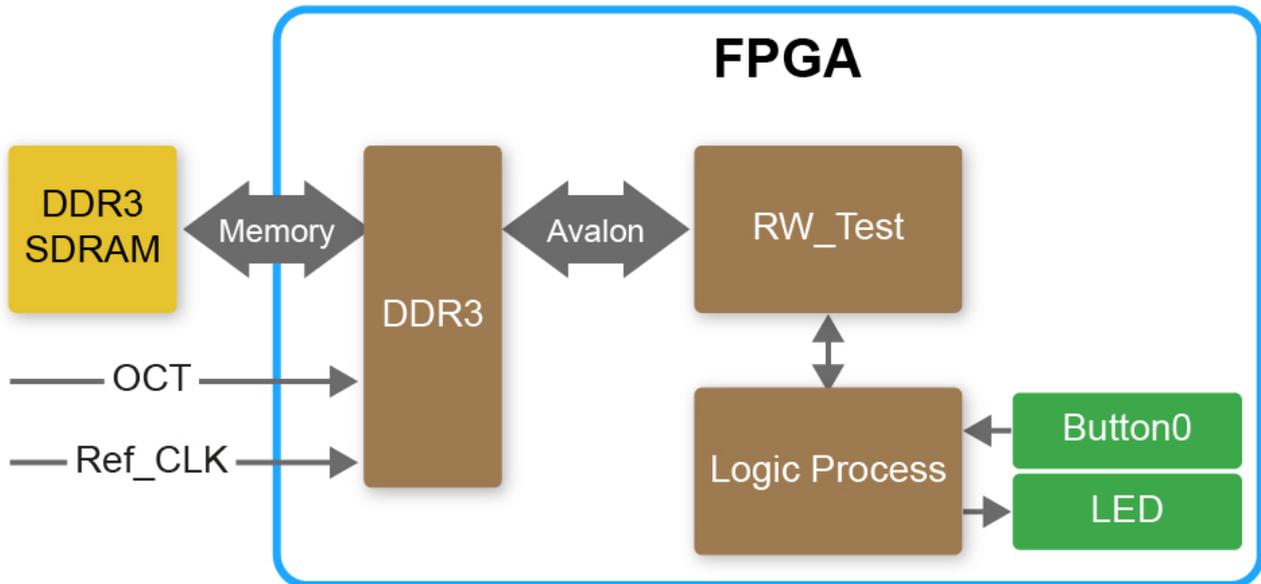
## 6.2 DDR3\_RTL

This demonstration performs a memory test function for DDR3 on the TSoM evaluation kit. The

memory size of the DDR3 used in this test is 1 GB.

## ■ System Block Diagram

**Figure 6-5** shows the function block diagram of this demonstration. The DDR3 controller uses 50 MHz as a reference clock. It generates one 300MHz clock as memory clock from the FPGA to the memory.



**Figure 6-5 Block Diagram of the DDR3\_RTL Demonstration**

## ■ Intel DDR3 SDRAM Controller with UniPHY

To use intel DDR3 controller, please perform the three major steps below:

1. Create correct pin assignments for DDR3.
2. Setup correct parameters in the dialog of DDR3 controller.

## ■ Design Tools

- Quartus Prime v17.1
- Demonstration Source Code:
  - ◆ Project Directory: Demonstration\FPGA\DDR3
  - ◆ Bit Stream: DDR3\output\_files\TSOM\_top.sof

### ■ Demonstration Batch File:

Demo Batch File Folder: DDR3\demo\_batch

The demo batch file includes following files:

- ◆ Batch File: test.bat
- ◆ FPGA Configuration File: TSOM\_top.sof

## ■ Demonstration Setup and Instructions by SOF File

Please follow below instructions to configure FPGA with .sof file through the usb-blaster JTAG interface to performance the demonstration.

- Make sure both Quartus Prime and USB-Blaster II driver are installed on the host PC.
- Connect TSoM evaluation kit to the host PC via USB cable. Install the USB-Blaster II driver if necessary.
- Power on the TSoM evaluation kit.
- Execute the demo batch file “test.bat” under the batch file folder \DDR3\demo\_batch.
- Press KEY0 on TSoM evaluation kit to start the verification process. After approximately 1 seconds, LED0 stay on to indicate the DDR3 have passed the test. **Table 6-3** lists the LED indicators.
- If LED1 is not blinking, it means the 50MHz clock source is not working.
- If LED0 start blinking upon releasing KEY0, it indicates local\_cal\_success of the corresponding DDR3 fails.
- Press KEY0 again to regenerate the test control signals for a repeat test.

**Table 6-3 LED Indicators**

| NAME | Description        |
|------|--------------------|
| LED0 | DDR3 test result   |
| LED1 | 50MHz clock source |

# Chapter 7

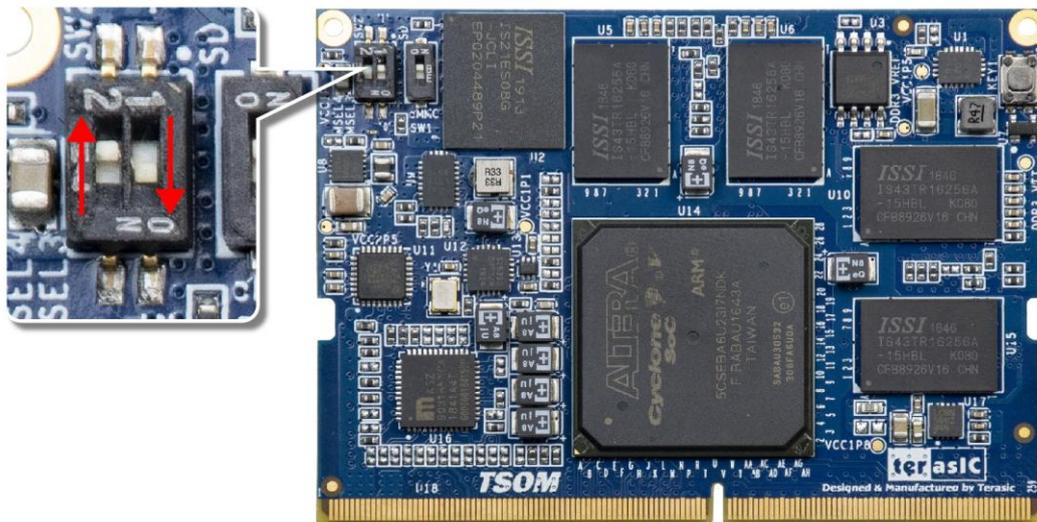
## Programming the EPCS Device

This chapter describes how to program the serial configuration (EPCS) device with Serial Flash Loader (SFL) function via the JTAG interface. Users can program EPCS devices with a JTAG indirect configuration (.jic) file, which is converted from a user-specified SRAM object file (.sof) in Quartus. The .sof file is generated after the project compilation is successful. The steps of converting .sof to .jic in Quartus II are listed below.

### ■ Before Programming Begins

The FPGA should be set to AS x1 mode i.e. MSEL[4..0] = "10010" to use the EPCS as a FPGA configuration device. Note that the factory default mode is FPPx32 mode(MSEL[4:0]= "01010").

User can set the SW2 of the TSoM module to change the MSEL[4:0] to ASx1 mode as shown in **Figure 7-1**.

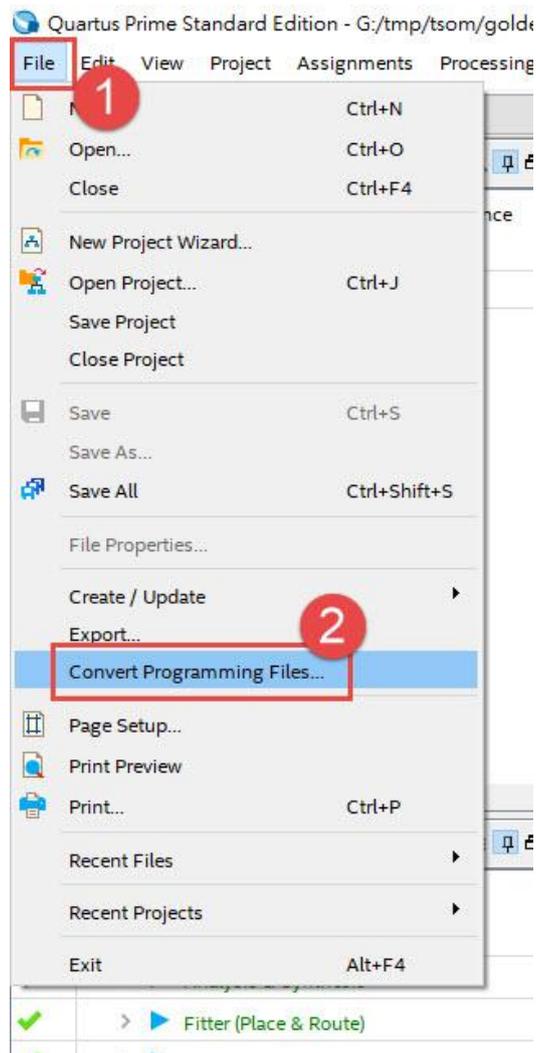


**Figure 7-1 SW2 Setting Resistors (AS Mode)**

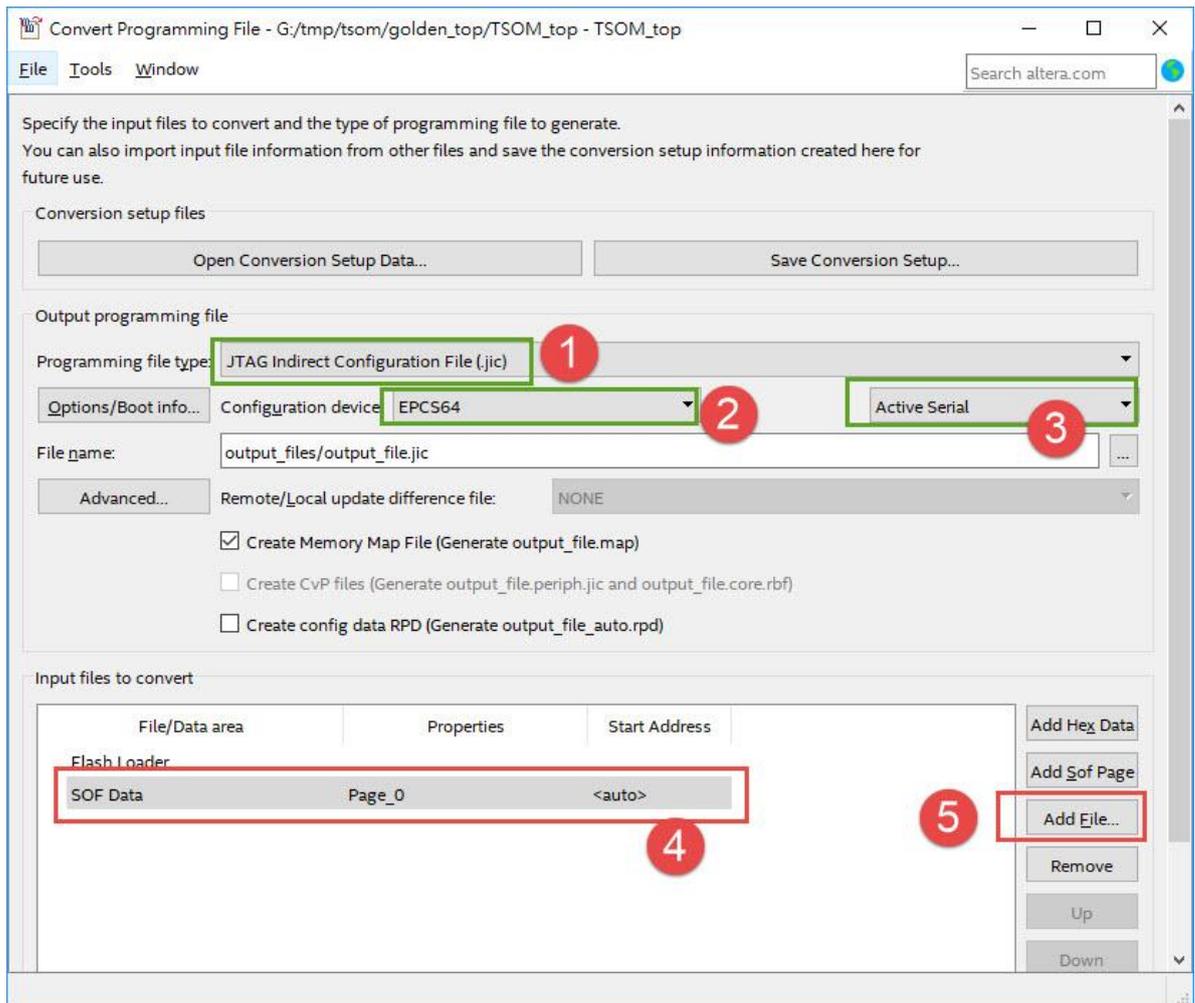
## ■ Convert .SOF File to .JIC File

To programme the design file into the EPCS device, a .jic file is required. Here's how to convert the .sof file generated by Quartus compile to .jic.

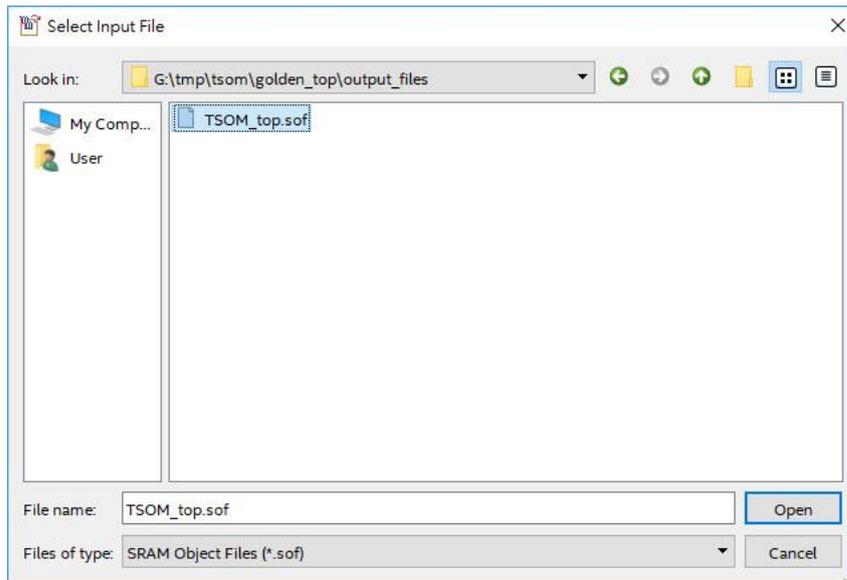
1. Choose Convert Programming Files from the File menu of Quartus II.



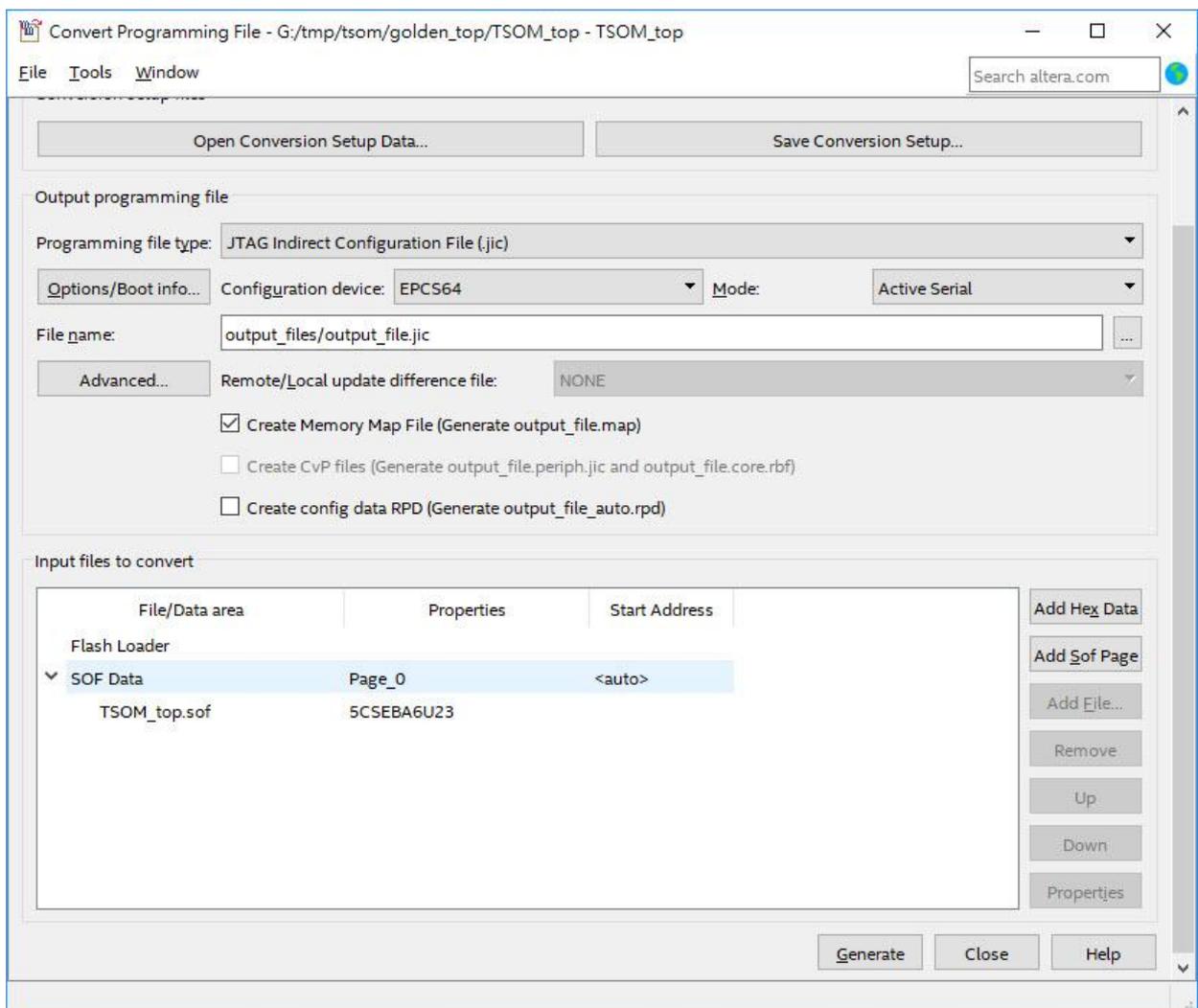
2. Select JTAG Indirect Configuration File (.jic) from the Programming file type field in the dialog of Convert Programming Files.



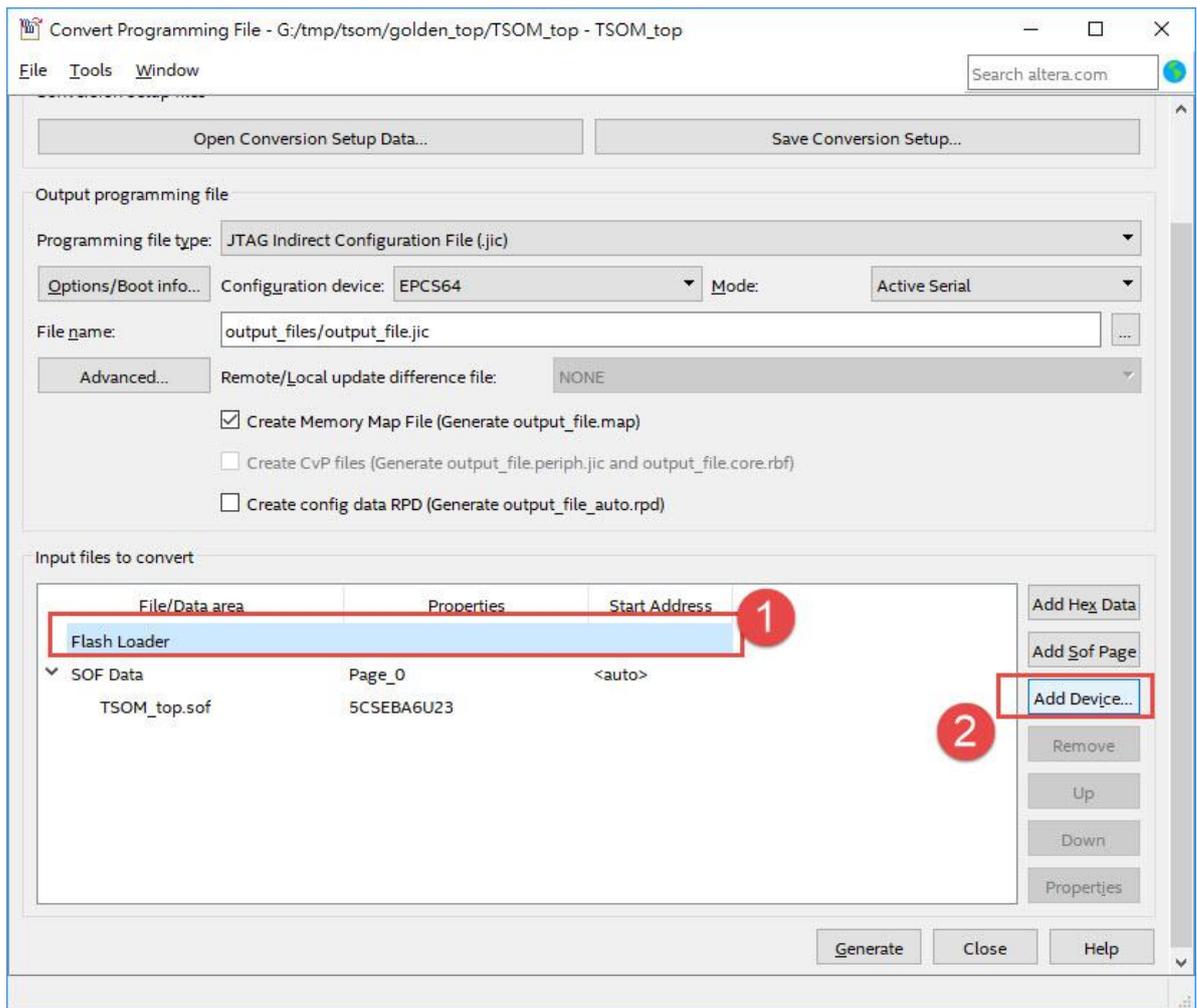
3. Choose EPCS64 from the Configuration device field.
4. Choose Active Serial from the Mode field.
5. Browse to the target directory from the File name field and specify the name of output file.
6. Click on the SOF data in the section of Input files to convert.
7. Click Add File.
8. Select the .sof to be converted to a .jic file from the Open File dialog.



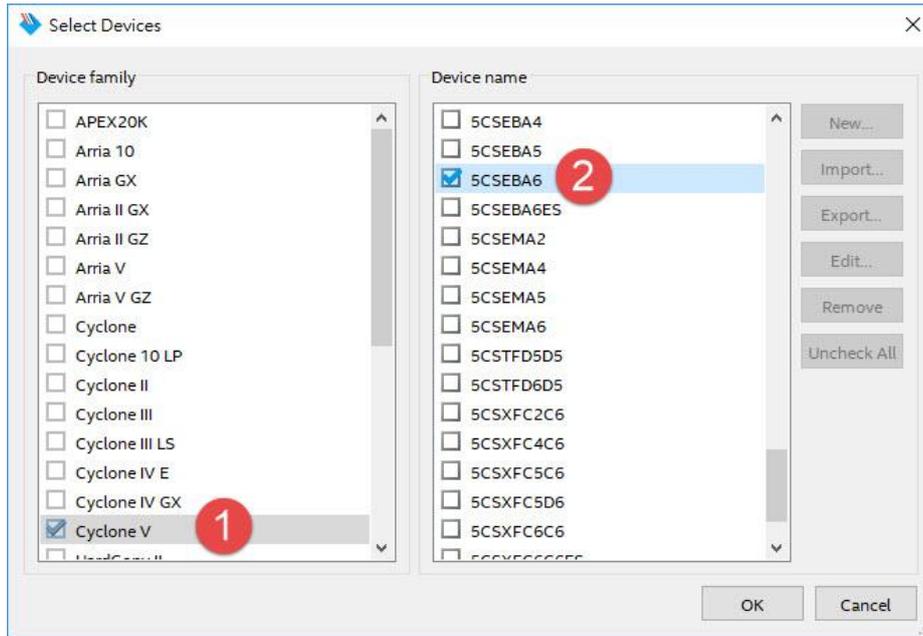
9. Click Open and the Convert Programming Files page will appear.



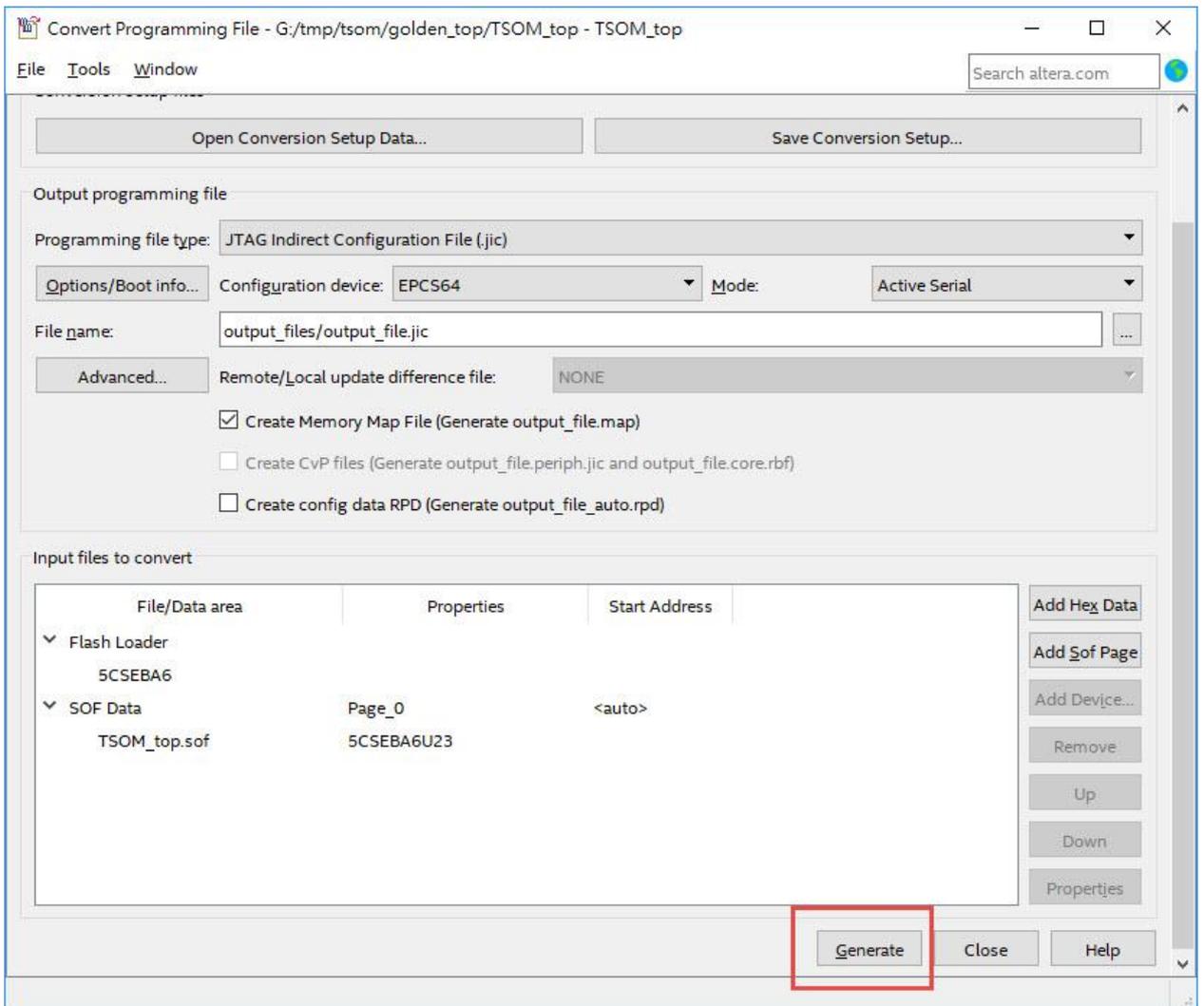
10. Click on the Flash Loader and click Add Device.



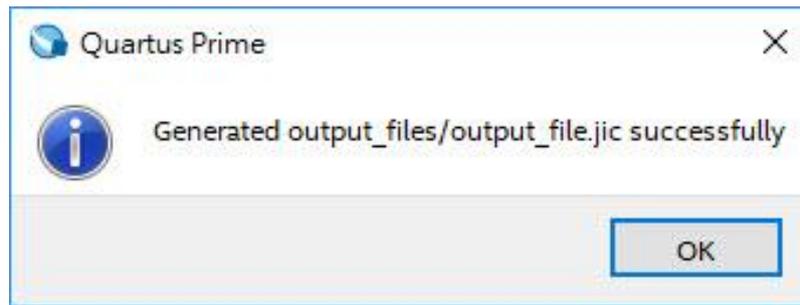
11. The Select Devices page will appear, please select the targeted FPGA to be programmed into the EPCS.



12. Click OK and the Convert Programming Files page will appear.

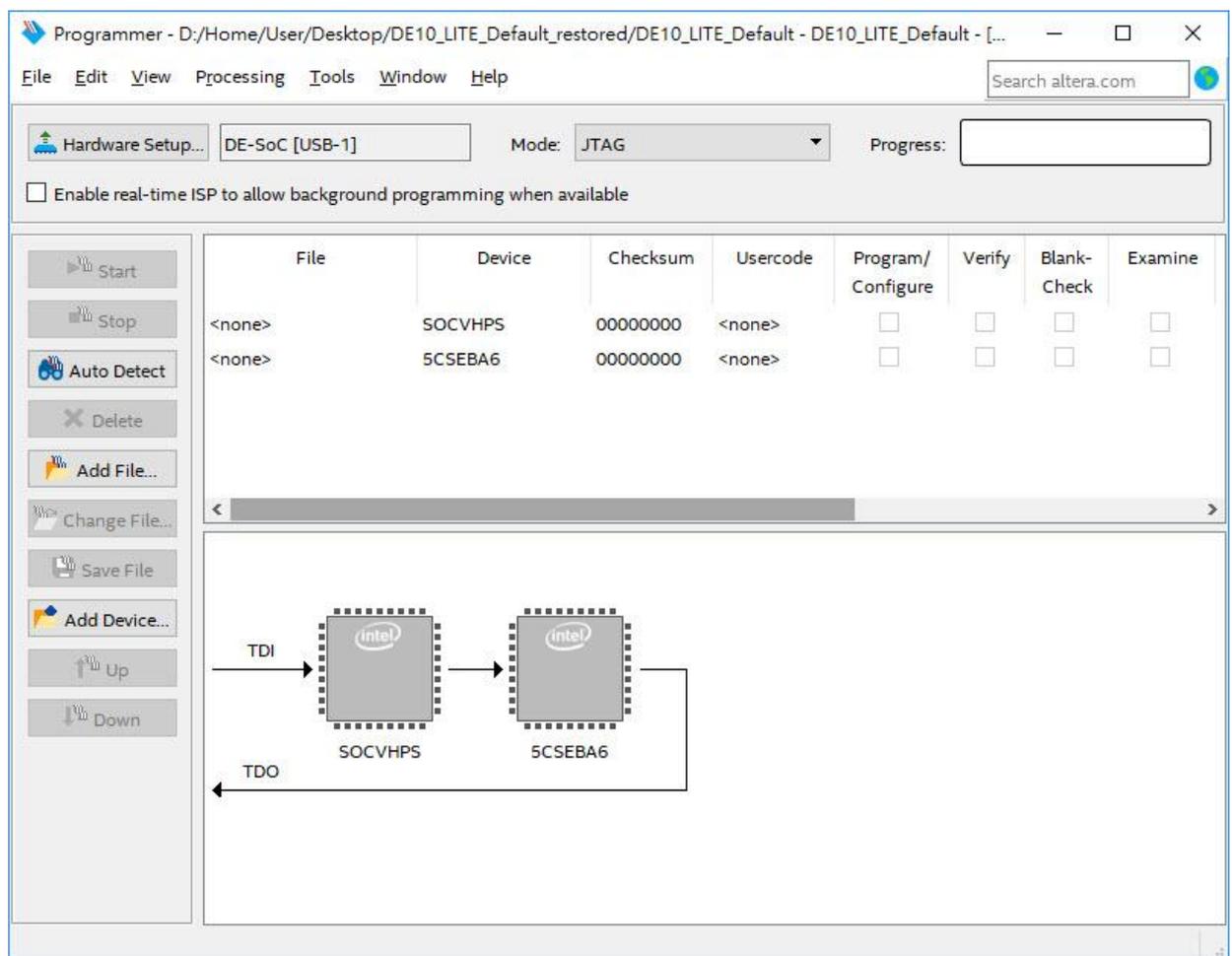


13. Click Generate and wait the .jic file is generated.

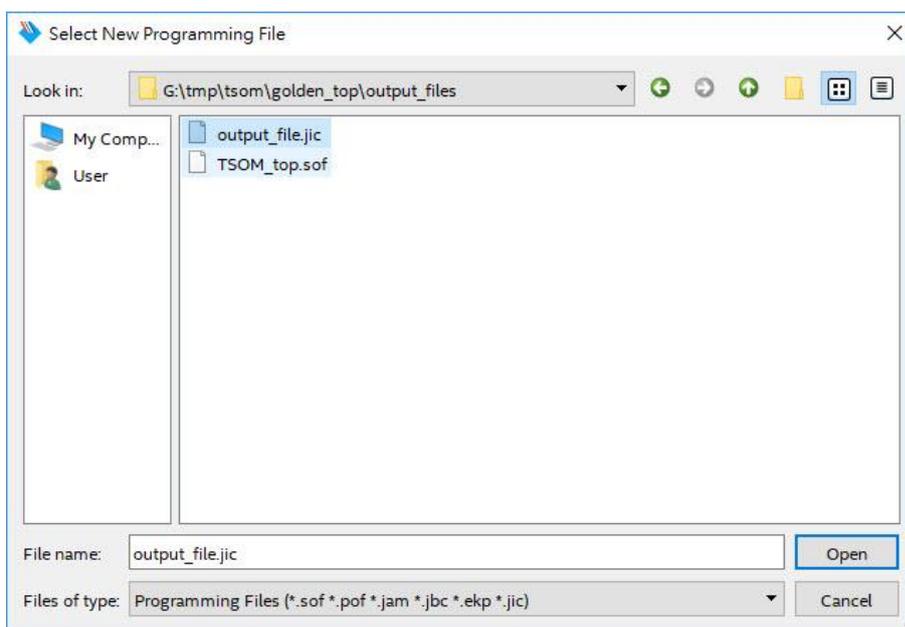
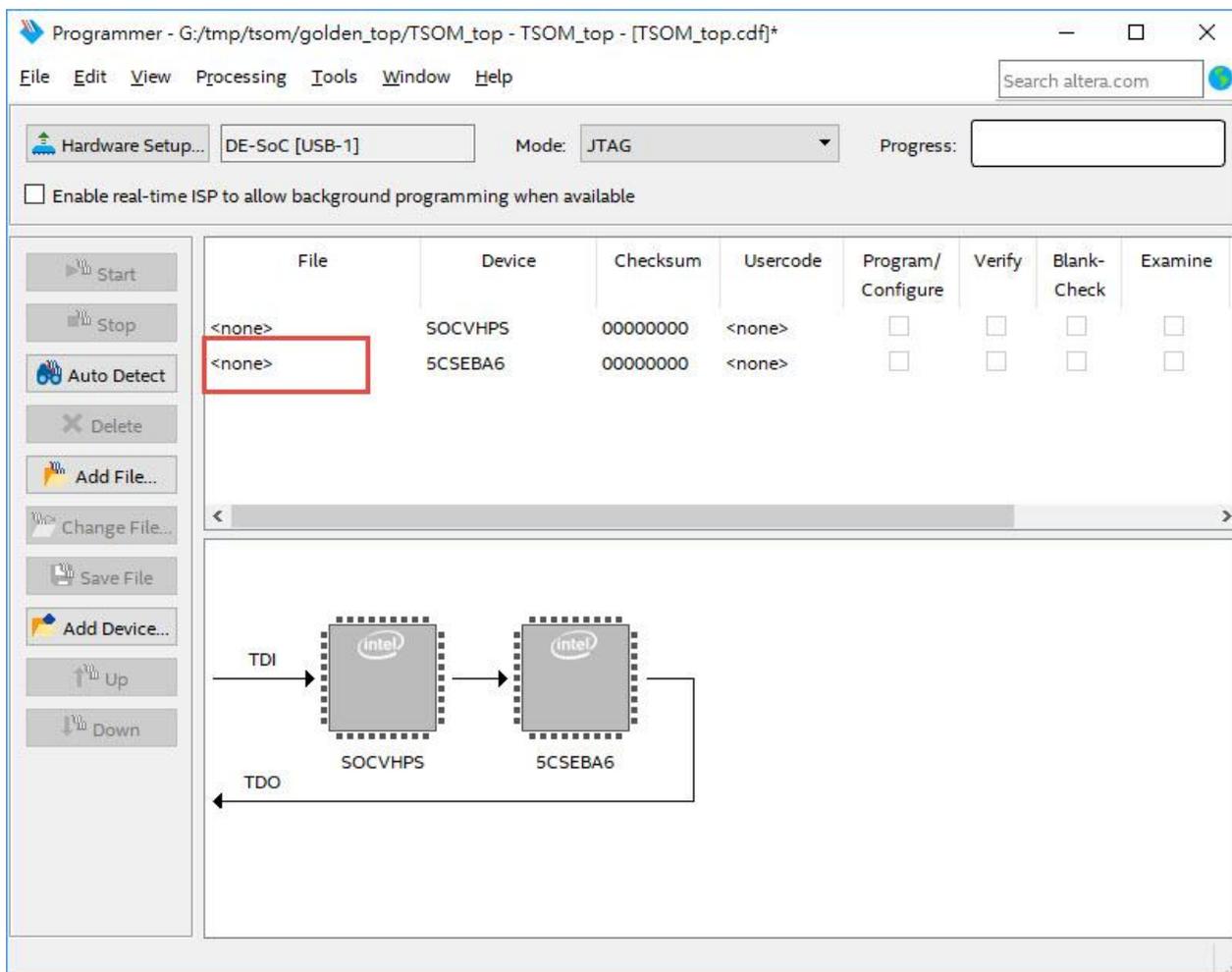


## ■ Write JIC File into the EPCS Device

1. Make sure the MSEL[4..0] is set to "10010" in AS mode.
2. Choose Programmer from the Tools menu and the Chain.cdf window will appear.
3. Click Auto Detect and then select the correct device (5CSEBA6). Both FPGA device and HPS should be detected

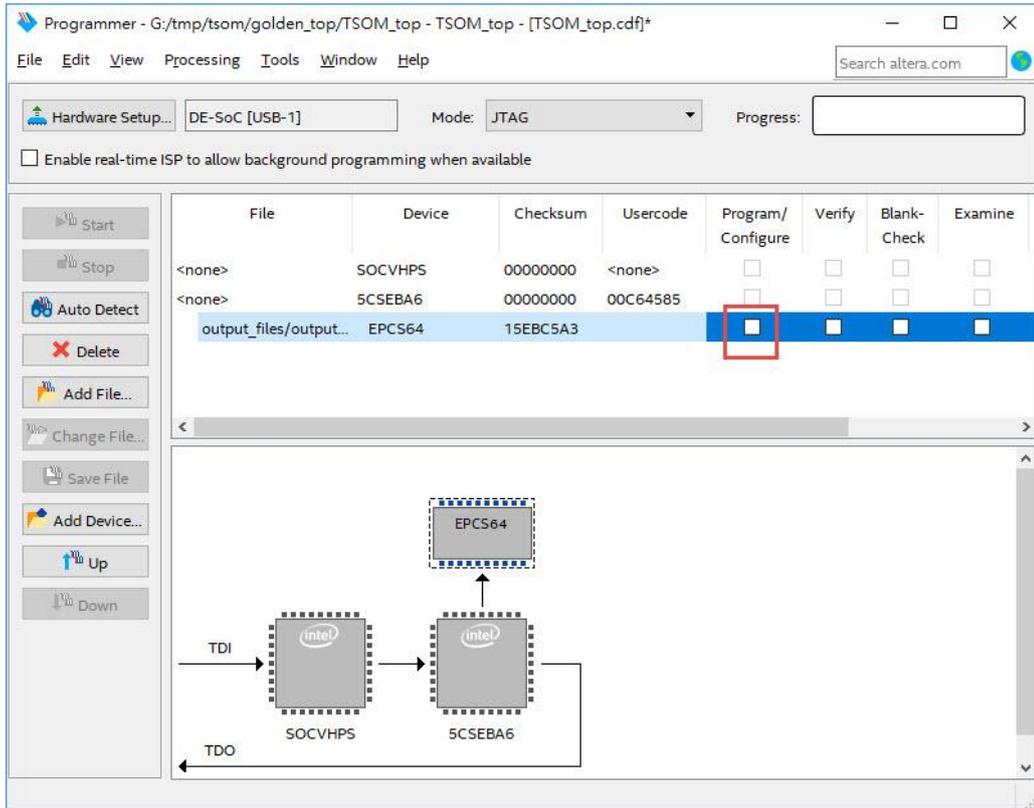


- Double click the red rectangle region shown in below and the "Select New Programming File" page will appear. Select the .jic file to be programmed.

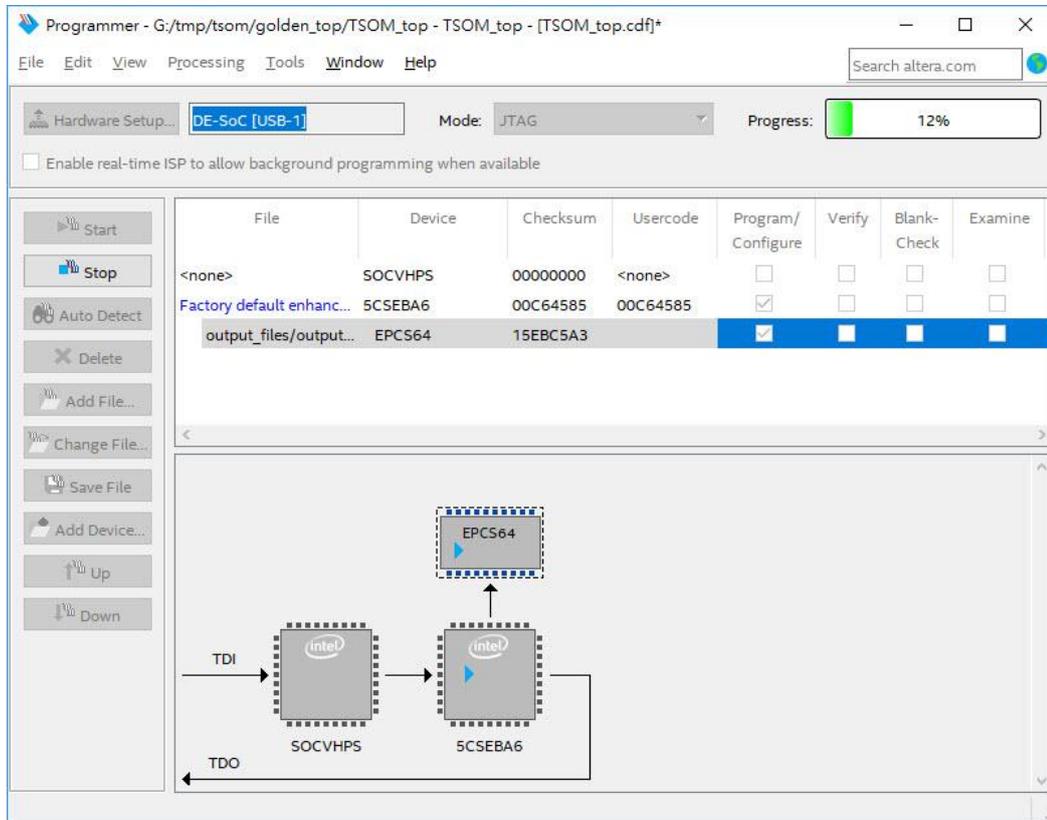


- Program the EPCS device by clicking the corresponding Program/Configure box. A factory

default SFL image will be loaded.

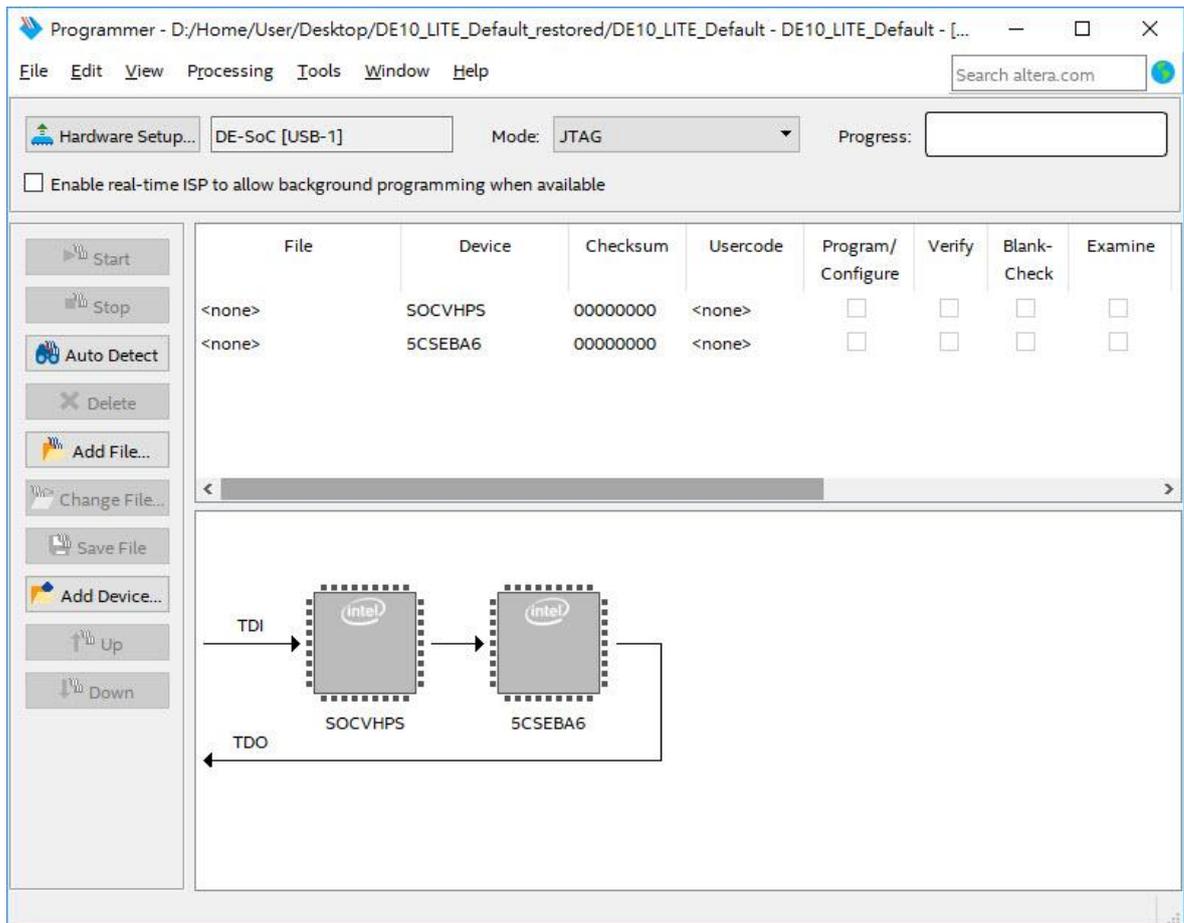


6. Click Start to program the EPCS device.

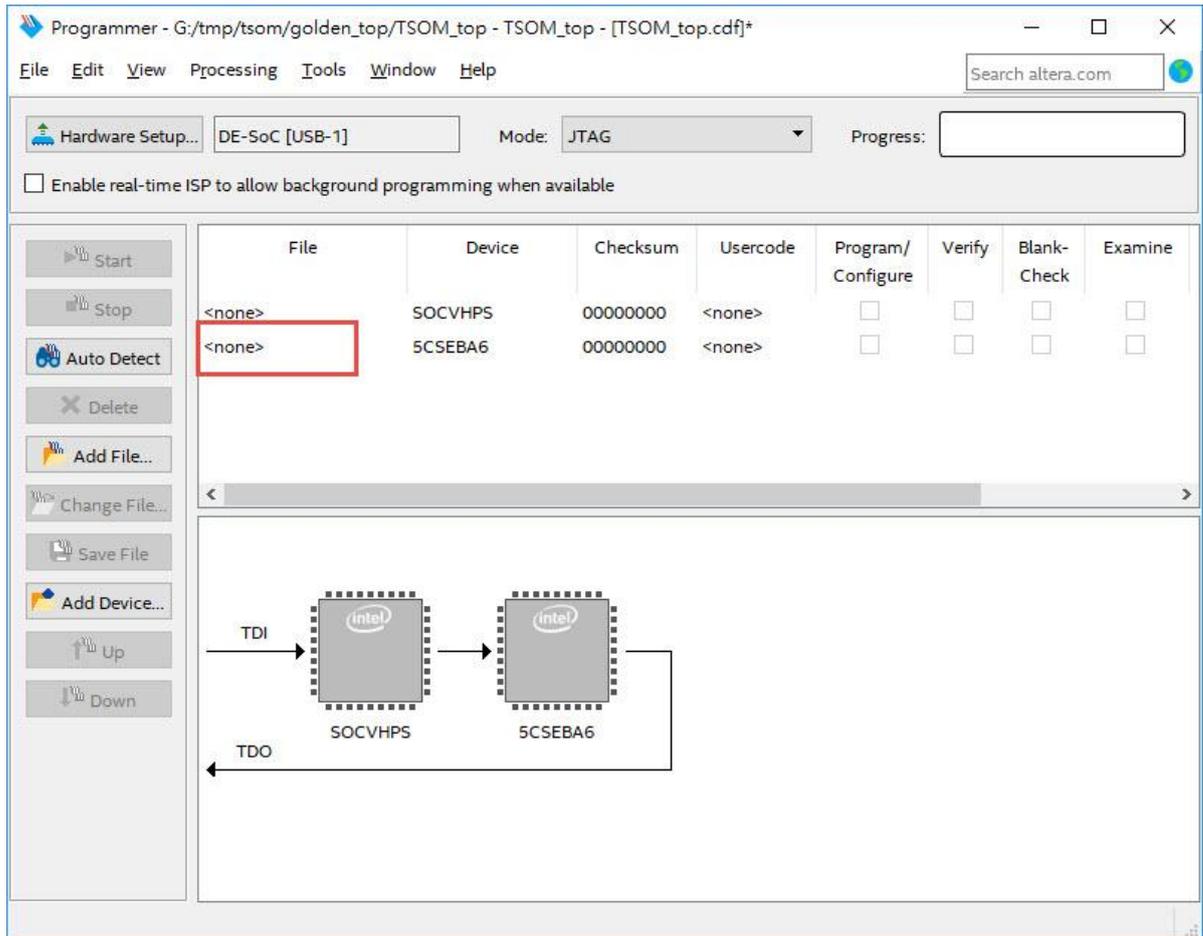


## ■ Erase the EPCS Device

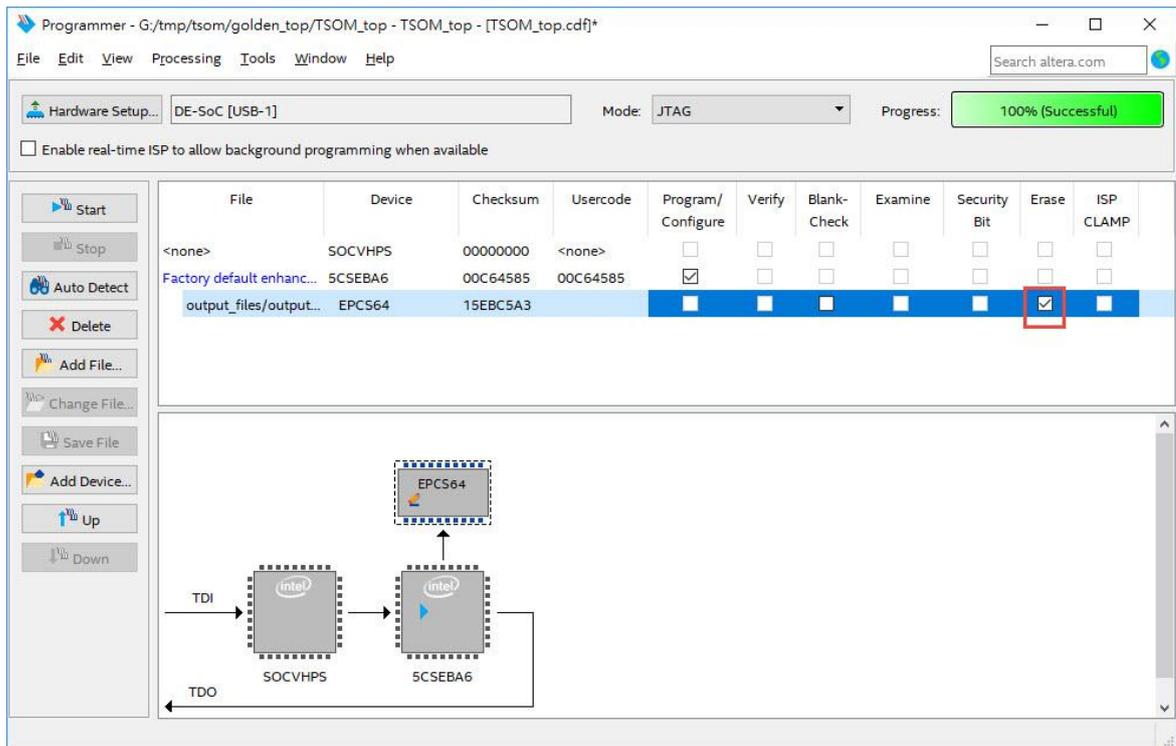
1. Erase the EPCS Device
2. Make sure the MSEL[4..0] is set to “10010” in AS mode.
3. Choose Programmer from the Tools menu and the Chain.cdf window will appear.
4. Click Auto Detect, and then select correct device, both FPGA device and HPS will detected.



5. Double click the red rectangle region shown in below, and the Select New Programming File page will appear. Select the correct .jic file.



6. Erase the EPCS device by clicking the corresponding Erase box. A factory default SFL image will be loaded.



7. Click Start to erase the EPCS device.

# Chapter 8

## *eMMC Programming*

This guide will show you how to program the Linux image into the eMMC device on the TSoM module and explain how to update device tree and zImage in the eMMC device.

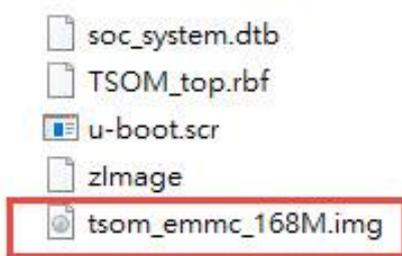
### ■ Re-flash linux image to emmc memory

1. Prepare a SD card reader and a Micro SD card (minimum 4GB capacity,for Linux boot).
2. Download the image file form the link below and use the tool such as "Win32 Disk Imager" to write the linux image file(tsom\_linux\_console.zip) into the Mircro SD card.

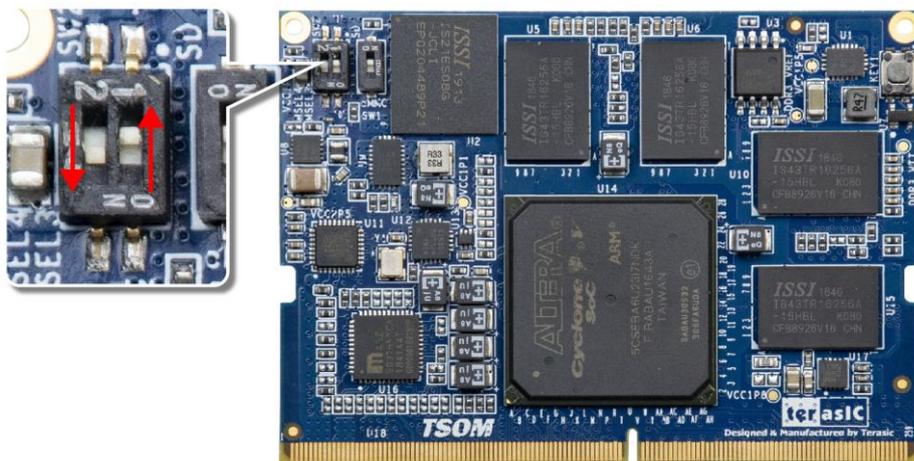
[http://mail.terasic.com.tw/~johnny/release/tsom/tsom\\_linux\\_console.zip](http://mail.terasic.com.tw/~johnny/release/tsom/tsom_linux_console.zip)

3. Download the eMMC image file from the link below and extract it, then copy the image file(tsom\_emmc\_168M.img) to the fat partition of sdcard

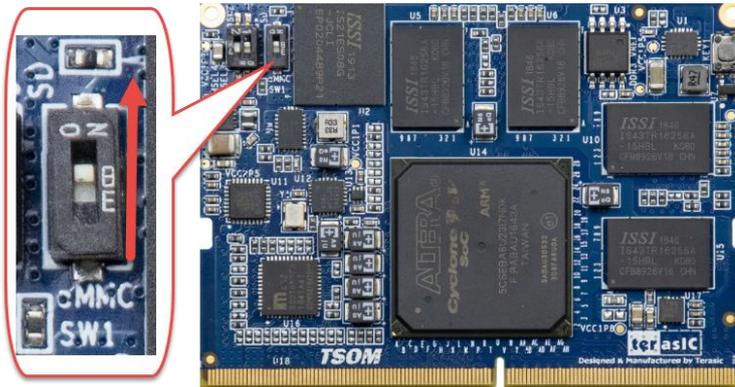
[http://mail.terasic.com.tw/~johnny/release/tsom/tsom\\_emmc\\_168M.img.zip](http://mail.terasic.com.tw/~johnny/release/tsom/tsom_emmc_168M.img.zip)



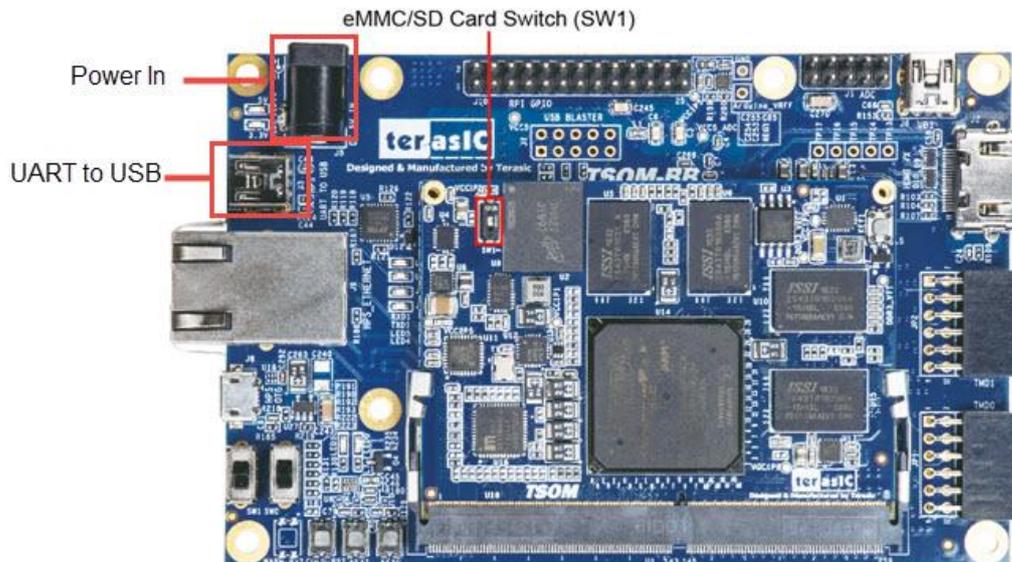
4. Insert the Micro SD card into the TSoM evaluation kit.
5. Make sure SW2 is set to FPPx32 mode(See the picture below), so the FPGA can be configured by HPS. The factory default mode is the FPPx32 mode.



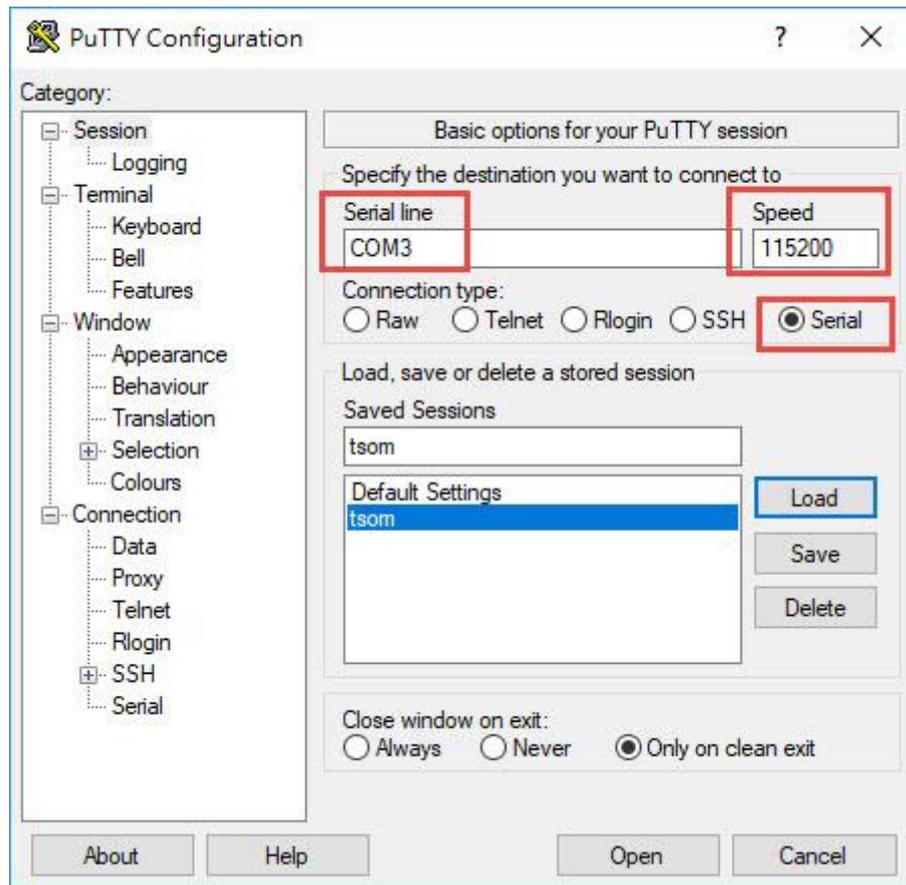
6. Make sure that SW1 on the TSoM is switch to the "ON" position, which represents the HPS boot from the SD Card.



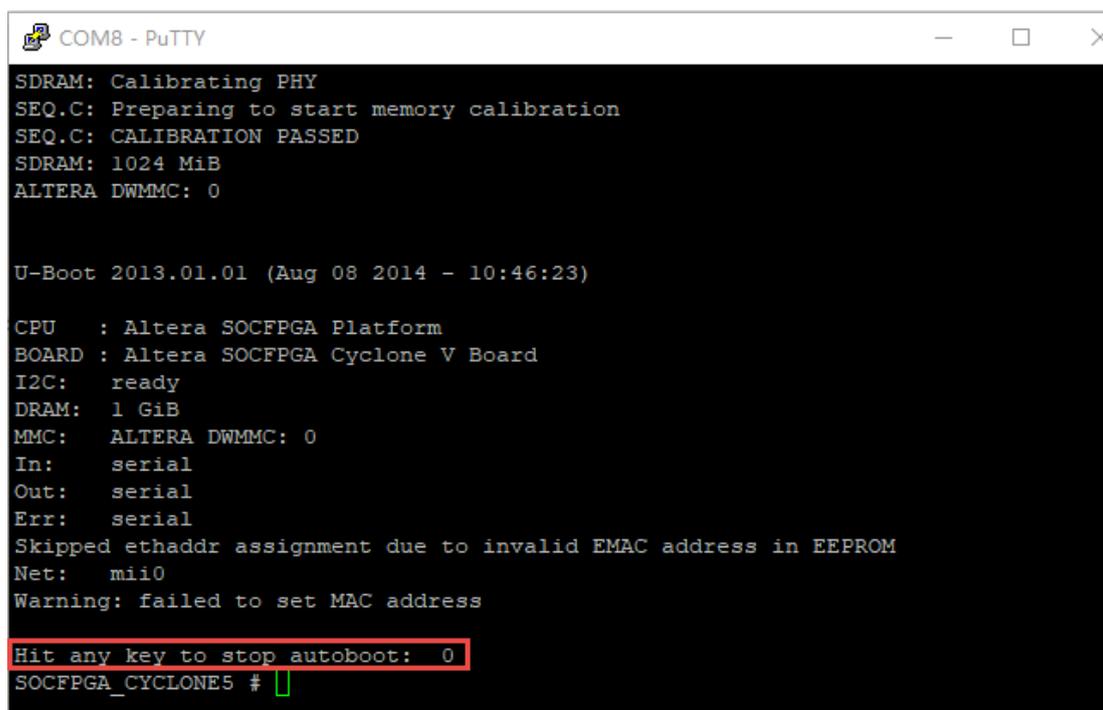
7. Connect the PC and TSoM evaluation kit via mini USB cable and connect the Power adapter to power on the board.



8. Use the terminal emulation program such as Putty on the operating system to connect to Linux running on TSoM



9. Press the CPU KEY1(WARM\_RST) button on the TSoM module and wait for the "Hit any key to stop autoboot :5" to appear in the Putty window. Press any key to enter uboot.



10. Enter the command **"fatls mmc 0:1"** to view the contents of the sdcard fat partition.

```
COM8 - PuTTY
BOARD : Altera SOCFPGA Cyclone V Board
I2C:   ready
DRAM:  1 GiB
MMC:   ALTERA DWMMC: 0
In:    serial
Out:   serial
Err:   serial
Skipped ethaddr assignment due to invalid EMAC address in EEPROM
Net:   mii0
Warning: failed to set MAC address

Hit any key to stop autoboot: 0
SOCFPGA_CYCLONE5 # fatls mmc 0:1
169984000  tsom_emmc_168m.img
17433     socfpga.dtb
3834632   zimage
          system volume information/
          qspi/
          24  readme.txt

4 file(s), 2 dir(s)

SOCFPGA_CYCLONE5 #
```

11. Enter the command **"fatload mmc 0:1 \$loadaddr tsom\_emmc\_168M.img"** to load the .img file into the DDR3.

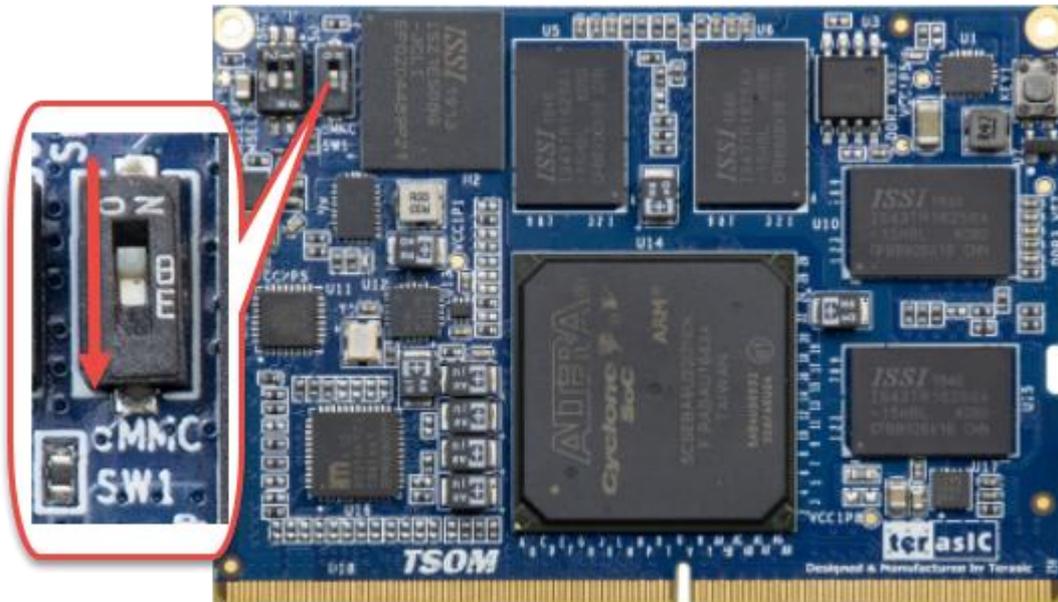
```
COM8 - PuTTY
MMC:   ALTERA DWMMC: 0
In:    serial
Out:   serial
Err:   serial
Skipped ethaddr assignment due to invalid EMAC address in EEPROM
Net:   mii0
Warning: failed to set MAC address

Hit any key to stop autoboot: 0
SOCFPGA_CYCLONE5 # fatls mmc 0:1
169984000  tsom_emmc_168m.img
17433     socfpga.dtb
3834632   zimage
          system volume information/
          qspi/
          24  readme.txt

4 file(s), 2 dir(s)

SOCFPGA_CYCLONE5 # fatload mmc 0:1 $loadaddr tsom_emmc_168M.img
reading tsom_emmc_168M.img
169984000 bytes read in 15740 ms (10.3 MiB/s)
SOCFPGA_CYCLONE5 #
```

12. Switch SW1 to "OFF" position to switch to eMMC boot mode.



13. Enter the command "**mmc rescan**" to rescan the eMMC device.

```

COM8 - PuTTY
In: serial
Out: serial
Err: serial
Skipped ethaddr assignment due to invalid EMAC address in EEPROM
Net: mii0
Warning: failed to set MAC address

Hit any key to stop autoboot: 0
SOCFPGA_CYCLONE5 # fatls mmc 0:1
169984000  tsom_emmc_l68m.img
17433     socfpga.dtb
3834632  zimage
         system volume information/
         qspi/
24       readme.txt

4 file(s), 2 dir(s)

SOCFPGA_CYCLONE5 # fatload mmc 0:1 $loadaddr tsom_emmc_l68M.img
reading tsom_emmc_l68M.img
169984000 bytes read in 15740 ms (10.3 MiB/s)
SOCFPGA_CYCLONE5 # mmc rescan
SOCFPGA_CYCLONE5 #
  
```

14. Enter the command "**mmc write \$loadaddr 0 0x510e0**" to program image file to the eMMC device.

```
COM8 - PuTTY
Skipped ethaddr assignment due to invalid EMAC address in EEPROM
Net: mii0
Warning: failed to set MAC address

Hit any key to stop autoboot: 0
SOCFPGA_CYCLONE5 # fatls mmc 0:1
 169984000  tsom_emmc_168m.img
  17433   socfpga.dtb
 3834632   zimage
          system volume information/
          qspi/
    24    readme.txt

4 file(s), 2 dir(s)

SOCFPGA_CYCLONE5 # fatload mmc 0:1 $loadaddr tsom_emmc_168M.img
reading tsom_emmc_168M.img
169984000 bytes read in 15740 ms (10.3 MiB/s)
SOCFPGA_CYCLONE5 # mmc rescan
SOCFPGA_CYCLONE5 # mmc write $loadaddr 0 0x510e0

MMC write: dev # 0, block # 0, count 332000 ... 332000 blocks write: OK
SOCFPGA_CYCLONE5 #
```

15. Note: The number "0x510e0" represents the image size, the unit is "block", each block has 512 bytes size. The size of the image file here is 169984000 bytes , i.e.  $169984000/512=332000$  (block) . 332000 converted to hexadecimal is 0x510e0.
16. Press the COLD\_RST button of the TSoM evaluation kit to reboot the board into linux. When the "login in" message appears that shows the programming eMMC success.

```
COM8 - PuTTY
Skipped ethaddr assignment due to invalid EMAC address in EEPROM
Net: mii0
Warning: failed to set MAC address

Hit any key to stop autoboot: 0
SOCFPGA_CYCLONE5 # fatls mmc 0:1
 169984000  tsom_emmc_168m.img
  17433   socfpga.dtb
 3834632   zimage
          system volume information/
          qspi/
    24    readme.txt

4 file(s), 2 dir(s)

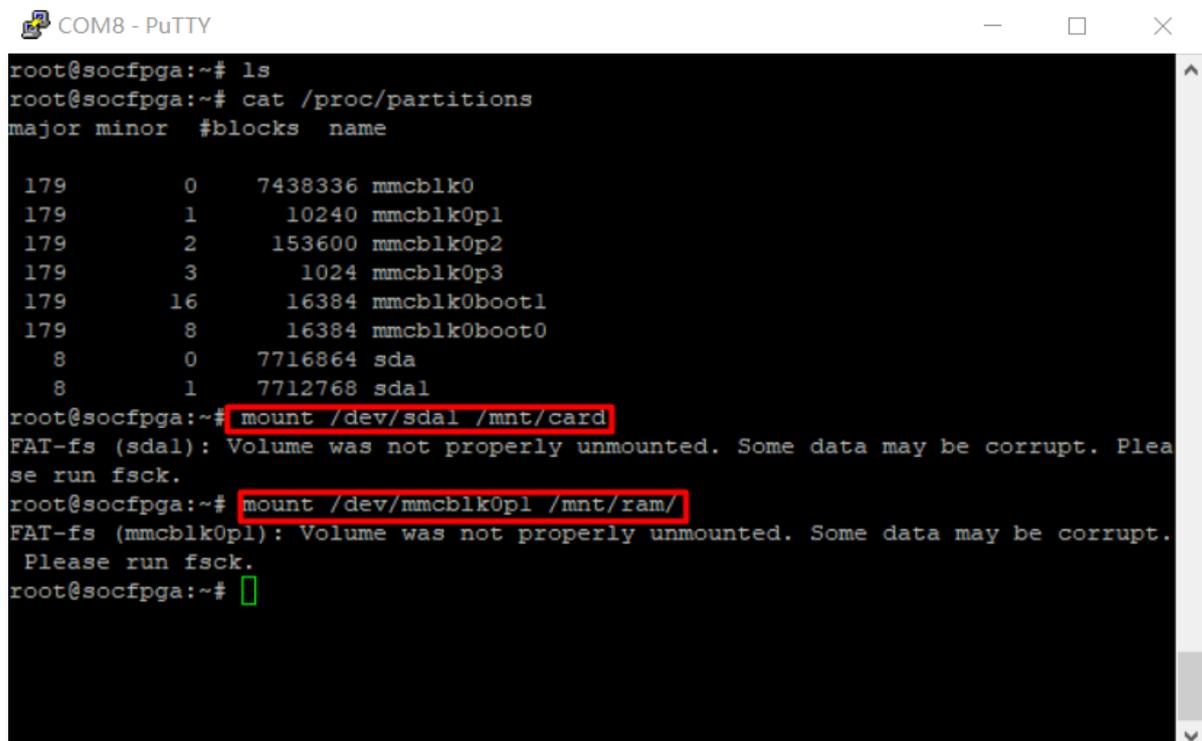
SOCFPGA_CYCLONE5 # fatload mmc 0:1 $loadaddr tsom_emmc_168M.img
reading tsom_emmc_168M.img
169984000 bytes read in 15740 ms (10.3 MiB/s)
SOCFPGA_CYCLONE5 # mmc rescan
SOCFPGA_CYCLONE5 # mmc write $loadaddr 0 0x510e0

MMC write: dev # 0, block # 0, count 332000 ... 332000 blocks write: OK
SOCFPGA_CYCLONE5 #
```

## ■ Update Device Tree and zImage in the eMMC

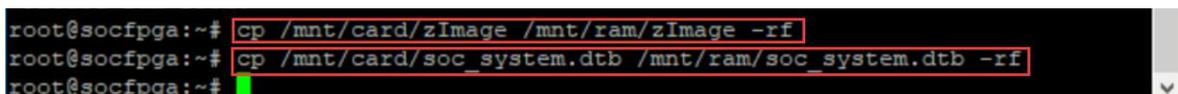
1. Before you update files to the eMMC, you should make sure the TSoM board can boot from eMMC correctly.
2. Copy your new device tree file(.dtb) and zImage files to USB flash drive.
3. Insert the USB flash drive to TSoM Board with USB OTG cable, then boot linux from eMMC
4. After the linux is boot, enter "root" to login the system, then, mount the USB flash drive and eMMC memory's partition one to linux. Commands: "**mount /dev/sda1 /mnt/card**" and "**mount /dev/mmcblk0p1 /mnt/ram/**".

(sda is USB flash drive, mmcblk0p1 is emmc memory's partition one; ignore the information when you do mount)



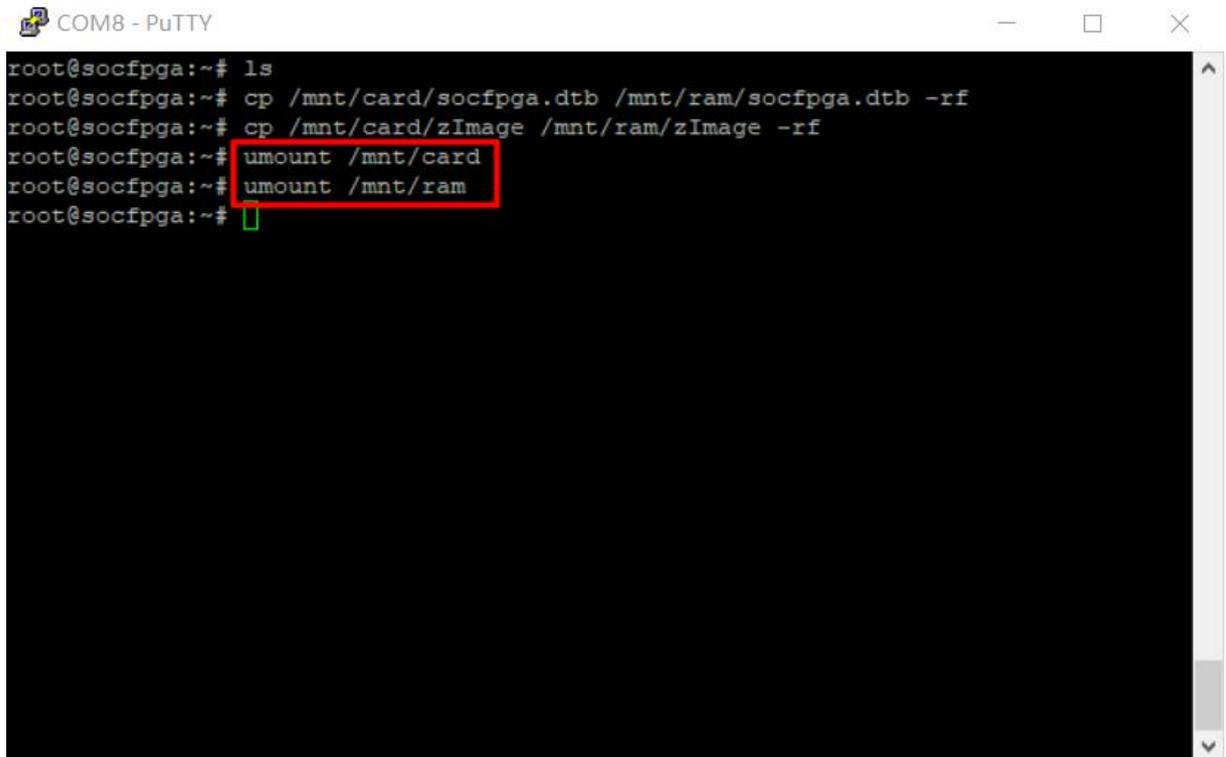
```
COM8 - PuTTY
root@socfpga:~# ls
root@socfpga:~# cat /proc/partitions
major minor #blocks name
179      0   7438336 mmcblk0
179      1    10240 mmcblk0p1
179      2   153600 mmcblk0p2
179      3    1024   mmcblk0p3
179     16    16384 mmcblk0boot1
179      8    16384 mmcblk0boot0
  8      0   7716864 sda
  8      1   7712768 sda1
root@socfpga:~# mount /dev/sda1 /mnt/card
FAT-fs (sda1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.
root@socfpga:~# mount /dev/mmcblk0p1 /mnt/ram/
FAT-fs (mmcblk0p1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.
root@socfpga:~#
```

5. Copy your new files from the USB flash drive to the eMMC memory with commands "**cp /mnt/card/zImage /mnt/ram/zImage -rf**" and "**cp /mnt/card/soc\_system.dtb /mnt/ram/soc\_system.dtb -rf**".



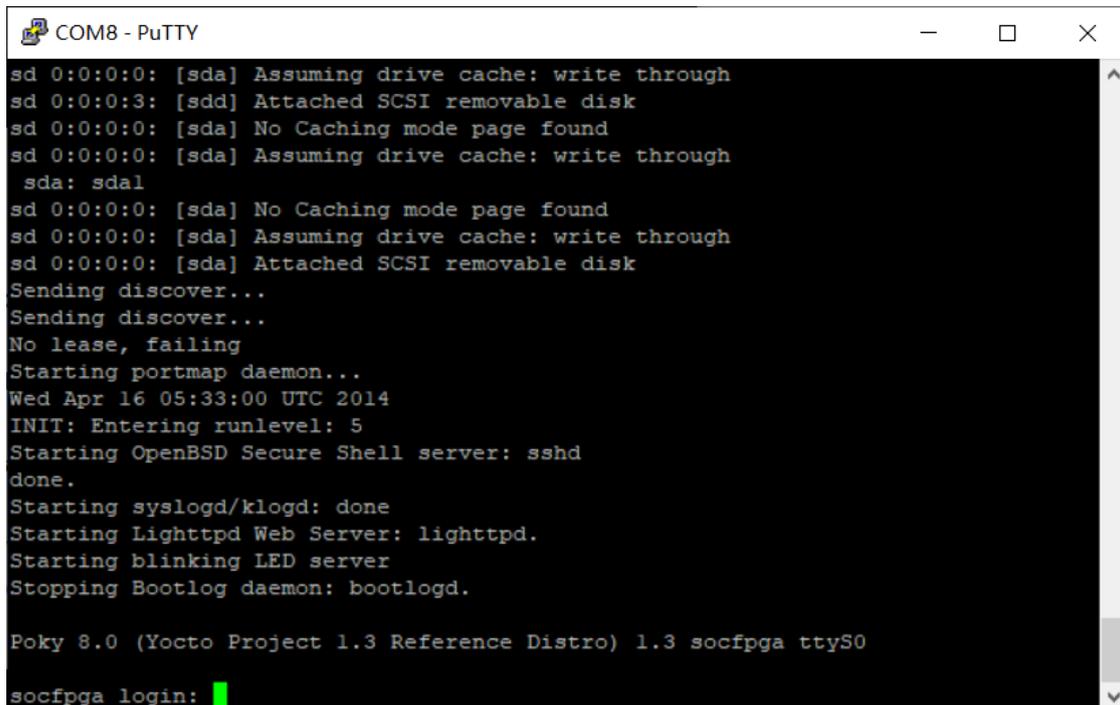
```
root@socfpga:~# cp /mnt/card/zImage /mnt/ram/zImage -rf
root@socfpga:~# cp /mnt/card/soc_system.dtb /mnt/ram/soc_system.dtb -rf
root@socfpga:~#
```

6. Umount USB flash drive(command:"umount /mnt/ram" ) and eMMC memory's partition one(command:"umount/mnt/car").



```
COM8 - PuTTY
root@socfpga:~# ls
root@socfpga:~# cp /mnt/card/socfpga.dtb /mnt/ram/socfpga.dtb -rf
root@socfpga:~# cp /mnt/card/zImage /mnt/ram/zImage -rf
root@socfpga:~# umount /mnt/card
root@socfpga:~# umount /mnt/ram
root@socfpga:~#
```

7. Press the COLD\_RST button to reboot TSoM board, then it will boot linux with your new updated files.



```
COM8 - PuTTY
sd 0:0:0:0: [sda] Assuming drive cache: write through
sd 0:0:0:3: [sdd] Attached SCSI removable disk
sd 0:0:0:0: [sda] No Caching mode page found
sd 0:0:0:0: [sda] Assuming drive cache: write through
sda: sda1
sd 0:0:0:0: [sda] No Caching mode page found
sd 0:0:0:0: [sda] Assuming drive cache: write through
sd 0:0:0:0: [sda] Attached SCSI removable disk
Sending discover...
Sending discover...
No lease, failing
Starting portmap daemon...
Wed Apr 16 05:33:00 UTC 2014
INIT: Entering runlevel: 5
Starting OpenBSD Secure Shell server: sshd
done.
Starting syslogd/klogd: done
Starting Lighttpd Web Server: lighttpd.
Starting blinking LED server
Stopping Bootlog daemon: bootlogd.

Poky 8.0 (Yocto Project 1.3 Reference Distro) 1.3 socfpga ttyS0
socfpga login: █
```

## 9.1 Revision History

| Version | Change Log                                       |
|---------|--|
| V1.0    | Initial Version (Preliminary)                    |
| V1.1    | Modify ch3 , ch7 and ch8 according rev B version |
| V1.2    | Modify chapter 8                                 |

Copyright © Terasic Inc. All rights reserved.