Unit / Module Description:	Capture Demo For Intech
Unit / Module Number:	EVP6472-SMT939
Document Issue Number	1.1
Issue Data:	1th March 2012
Original Author:	C Hong

EVP6472 Intech Demo

Abstract

Capture demonstration application based on EVP6472

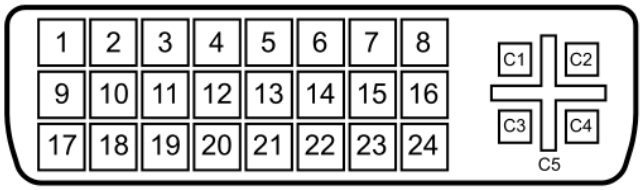
Sundance Multiprocessor Technology Ltd, Chiltern House,
Waterside, Chesham, Bucks. HP5 1PS.

Table of Contents

1. DVI Introduction

1.1 What is DVI interface

A Digital Video Interface is composed by 24 differential signals, as described in Table.1.



Pin 1 TMDS data 2− Digital red− (link 1)	Pin 16 Hot plug detect
Pin 2 TMDS data 2+ Digital red+ (link 1)	Pin 17 TMDS data 0− Digital blue− (link 1) and digital sync
Pin 3 TMDS data 2/4 shield	Pin 18 TMDS data 0+ Digital blue+ (link 1) and digital sync
Pin 4 TMDS data 4− Digital green− (link 2)	Pin 19 TMDS data 0/5 shield
Pin 5 TMDS data 4+ Digital green+ (link 2)	Pin 20 TMDS data 5− Digital red− (link 2)
Pin 6 DDC clock	Pin 21 TMDS data 5+ Digital red+ (link 2)
Pin 7 DDC data	Pin 22 TMDS clock shield
Pin 8 Analog vertical sync	Pin 23 TMDS clock+ Digital clock+ (links 1 and 2)
Pin 9 TMDS data 1− Digital green− (link 1)	Pin 24 TMDS clock− Digital clock− (links 1 and 2)
Pin 10 TMDS data 1+ Digital green+ (link 1)	C1 Analog red
Pin 11 TMDS data 1/3 shield	C2 Analog green
Pin 12 TMDS data 3- Digital blue− (link 2)	C3 Analog blue
Pin 13 TMDS data 3+ Digital blue+ (link 2)	C4 Analog horizontal sync
Pin 14 +5 V Power for monitor when in standby	C5 Analog ground Return for R, G, and B signals
Pin 15 Ground Return for pin 14 and analog sync	

Table.1 standard DVI pin out

1.2 What is our interface

To simplify the serialization complexity, the SMT939 (Fig.1) provides us with an easier user interface at the SLB (Sundance Local Bus) connector.

Table.2 shows the signal description at the SLB interface, these are the signals we need to access.

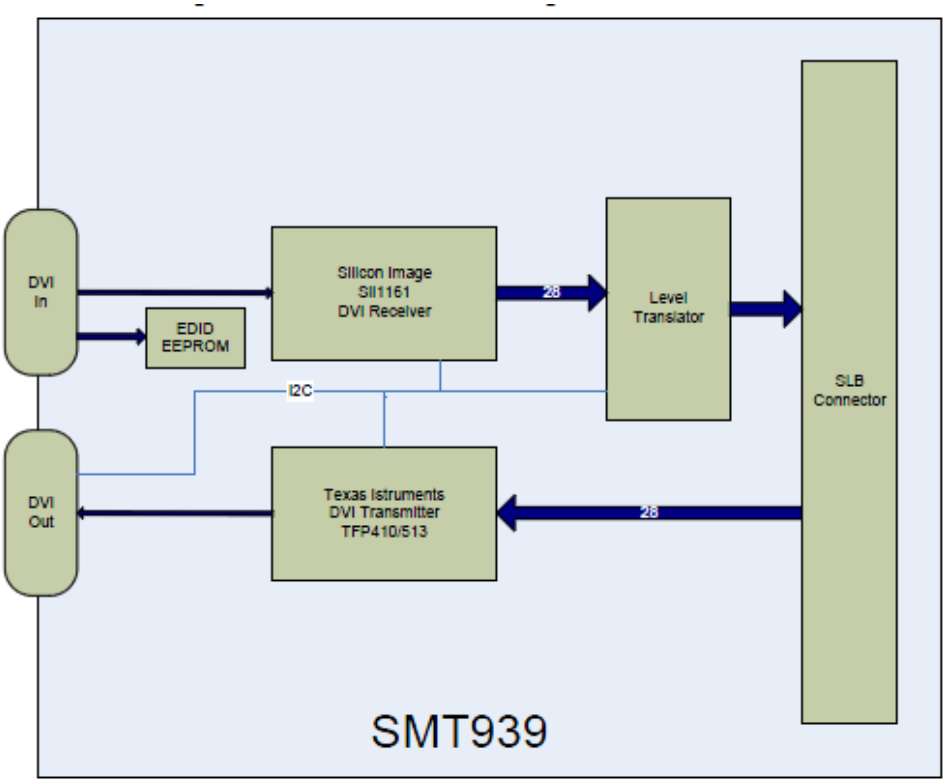


Fig1. SMT939 Block Diagram

Signal name	Type	Width	Signal description
CLK Input	I	1	DVI clock input
Red Input	I	8	8-bit red input
Green Input	I	8	8-bit green input
Blue Input	I	8	8-bit blue input
HS Input	I	1	Horizontal synchronization input
VS Input	I	1	Vertical synchronization input
DE Input	I	1	Data Enable input
CLK Output	O	1	DVI clock output
Red Output	O	8	8-bit red output
Green Output	O	8	8-bit green output
Blue Output	O	8	8-bit blue output
HS Output	O	1	Horizontal synchronization output
VS Output	O	1	Vertical synchronization output
DE Output	O	1	Data Enable output

Table.2. SLB interface for DVI signals

1.3 How signals work

Fig.2 shows how these signals are used. HS signal gives a pulse every time one row is scanned, whereas VS pulses every one frame. DE implies that it is scanning in the display window, where the 24-bit RGB data is presented.

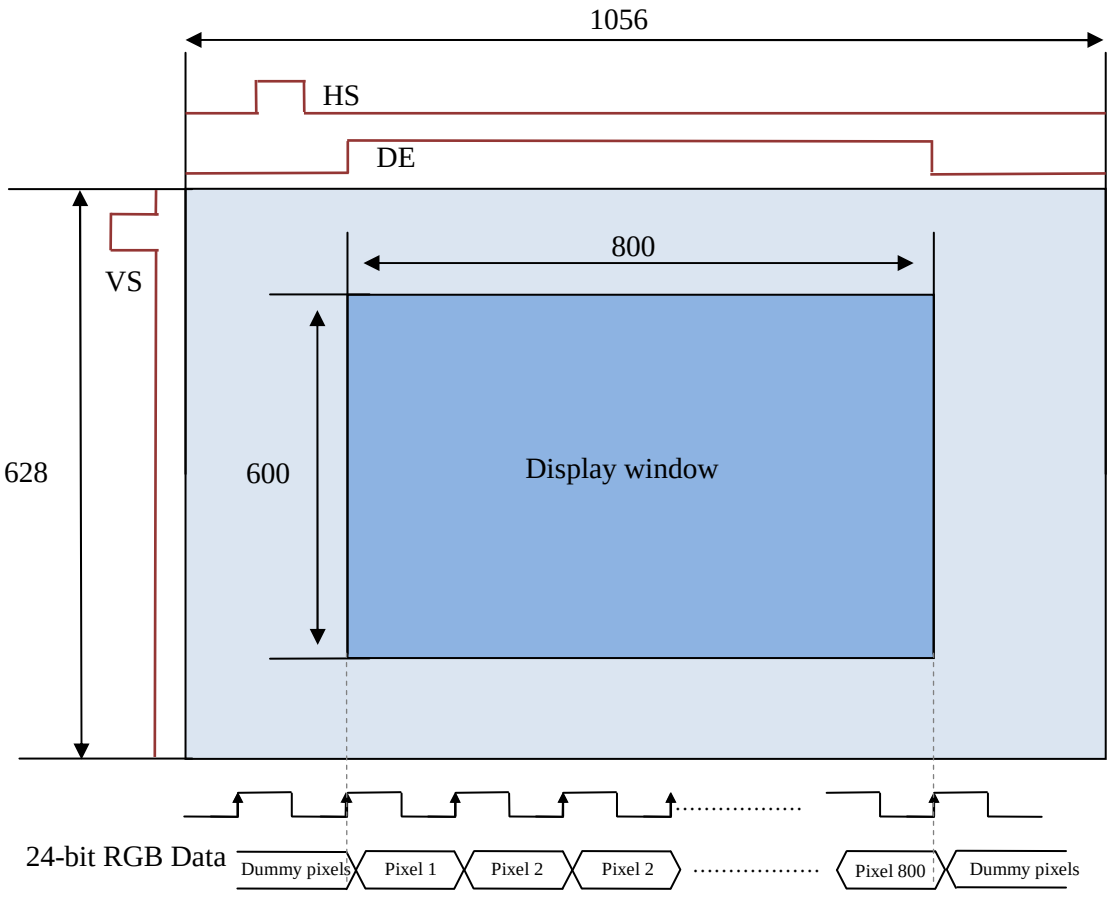


Fig.2 HS, VS, DE and Data signals

2. System Architecture

2.1 DVI signals Datapath

Fig.3 shows the datapath of the DVI signals.

The PC and the monitor are synchronized by directly linking three control signals (HS, VS, DS) from the PC to the monitor.

Only the RGB pixel data is transferred and processed by the DSP.

The input frame buffer is synchronized with the PC by keeping updating the pixels from the PC.

The monitor is synchronized with the output frame buffer by keeping fetching the pixels from the output frame buffer.

The data is transferred between FPGA and DSP using through Ethernet port and RGMII interface.

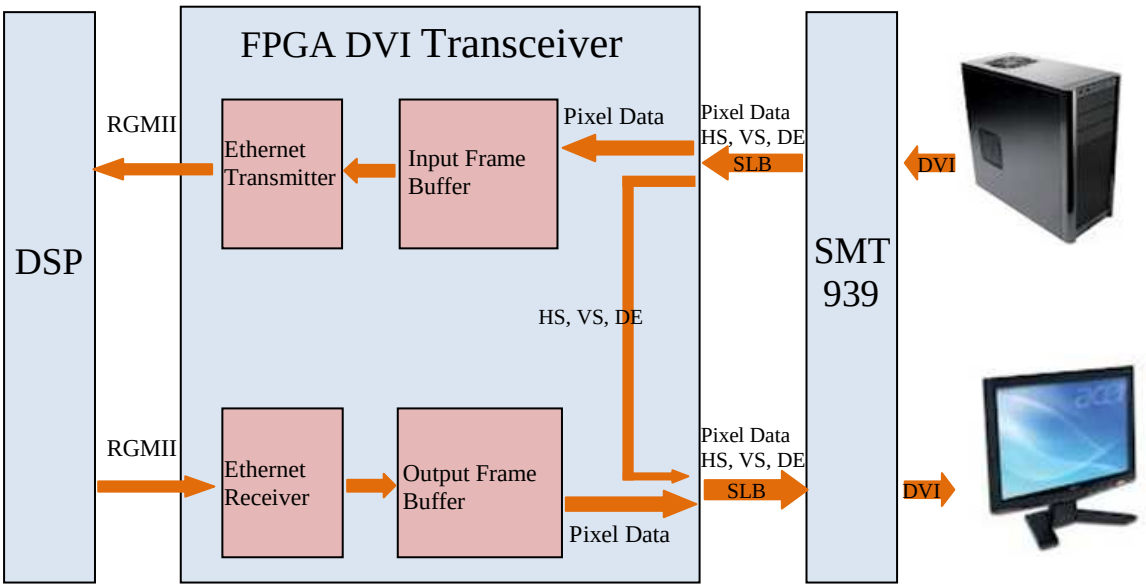


Fig.3 DVI signals datapath

2.2 System Architecture

Fig.4 gives the system architecture. The communication between board components is controlled by MicroBlaze, which is an FPGA on-chip softcore processor.

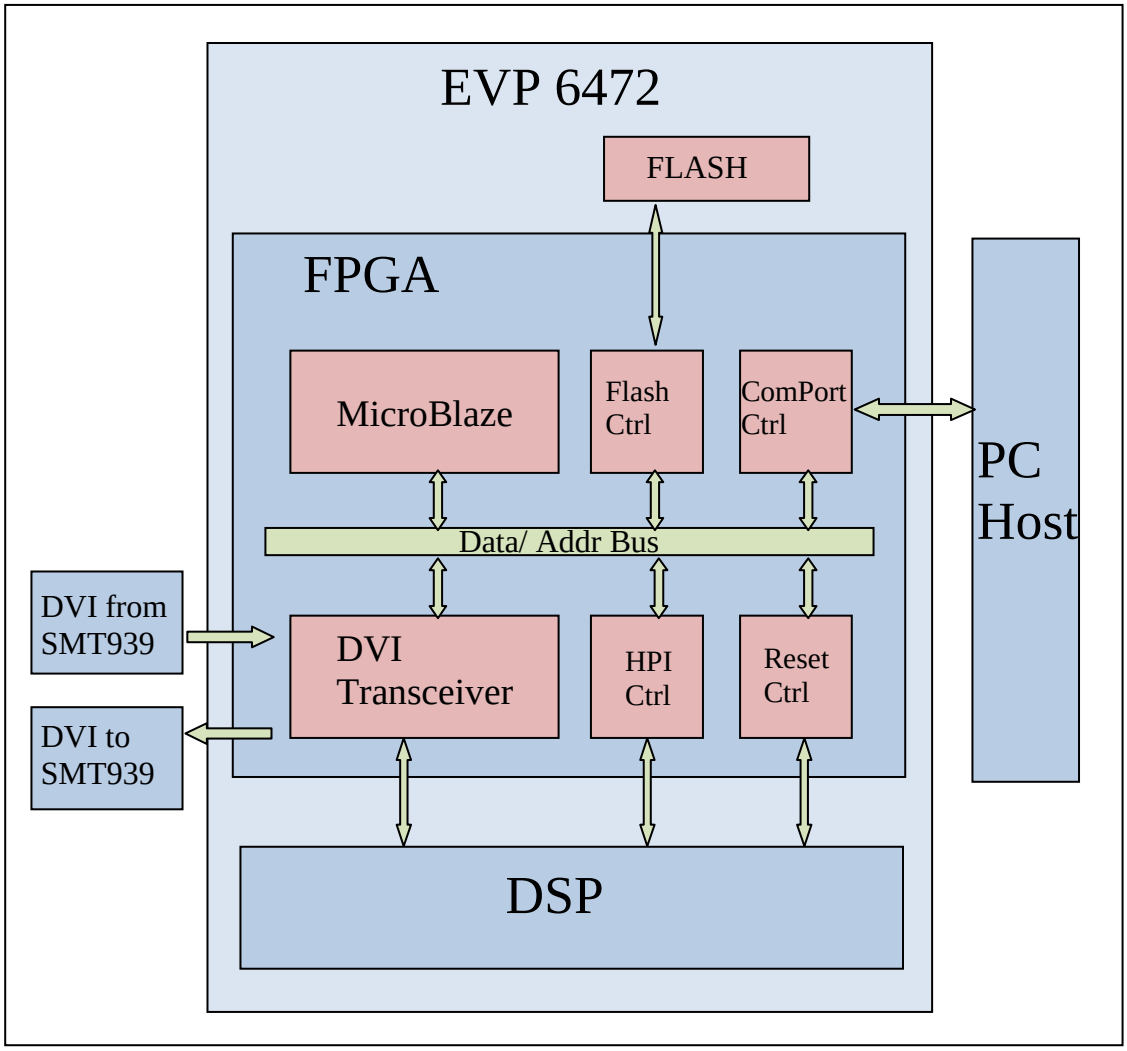


Fig.4 System Architecture

2.3 Peripheral Description

- MicroBlaze

  The FPGA on-chip microprocessor provides us a software platform to control underneath peripherals.

- DVI Transceiver

  The Transceiver passes the DVI signals from SMT939, to the DSP through Ethernet and loops them back to DVI outputs.

- HPI controller

  The HPI controller is used to configure the DSP, including programming code and

initialization settings.

- Reset Controller

  The reset controller is used to release DSP from reset.

- Comport Controller

  The Comport controller communicates with the host using the USB before boot up. This is used to load the DSP code from the host to the Flash memory.

- Flash Controller

  The Flash controller loads the DSP code from the Flash memory to the DSP when booting up.

3. How it works

After device configured:

1. After boot, the FPGA is automatically configured by the bitstream in the flash memory.
2. MicroBlaze reads the DSP code from Flash memory.
3. MicroBlaze writes the DSP code to the DSP through HPI interface.
4. MicroBlaze sets up the DSP configuration through HPI interface.
5. MicroBlaze resets and starts the DSP.
6. MicroBlaze sets up the configuration for the DVI transceiver.
7. MicroBlaze starts DVI transceiver.
8. DVI transceiver synchronizes the input frame buffer with the DVI input data.
9. MicroBlaze triggers DVI transceiver to send one frame's pixels to the DSP.
10. DSP receives one frame's pixels.
11. DSP processes one frame's pixels.
12. DSP sends one frame's pixel back to DVI transceiver.
13. DVI transceiver updates the output frame buffer with received pixel data.
14. DSP sets its interrupt signal after finishing one loop.
15. MicroBlaze receives the interrupt and starts another transmission.

Pre-requirement:

SMT6002 software

SMT111 driver

Smt372t.bin DSP application.

Boot_download.bit FPGA bitstream.

4. How to run the demonstration

Every time the EVP6472 boots up, it is self configured by loading the application code (bitstream for the FPGA and binary code for the DSP) from the flash memory. Therefore we need to program the flash for the very first time.

After flash programmed, the EVP6472 automatically starts working every time it boots up.

4.1 First time flash programming/recovery

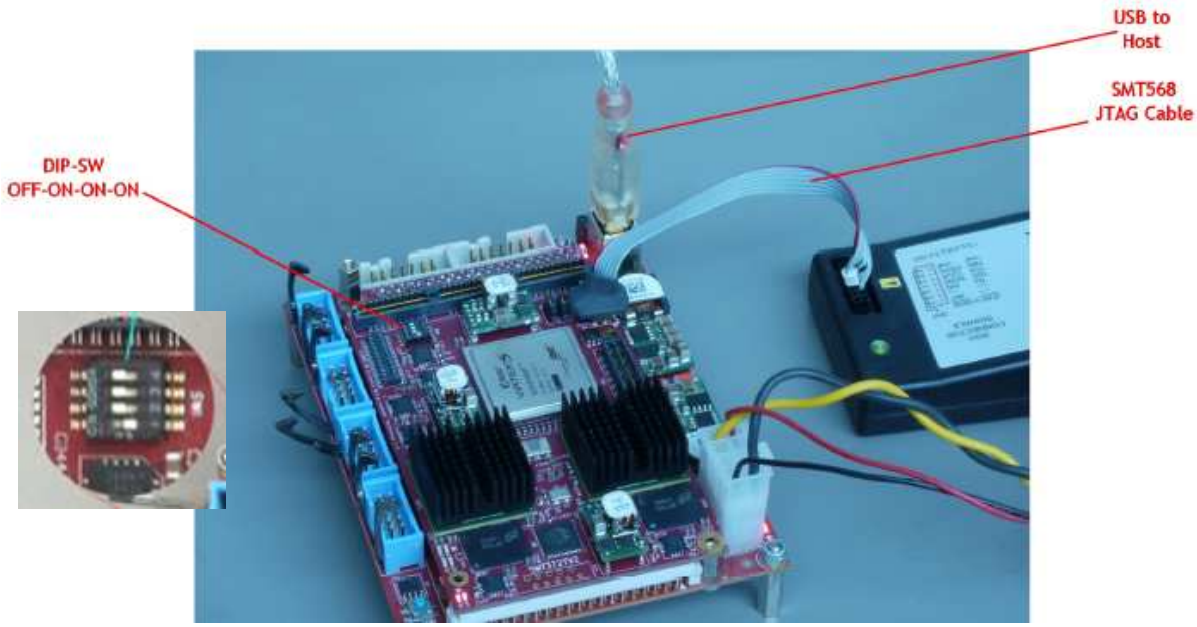
To program the flash, we use JTAG to load the bitstream to the FPGA, which communicates with the host through the USB.

The bitstream used for programming flash is the same one used for our normal application. It performs one of the actions depending on the first bit of the DIP switch (1 for programming flash and 0 for normal application). To programme the flash, steps are followed.

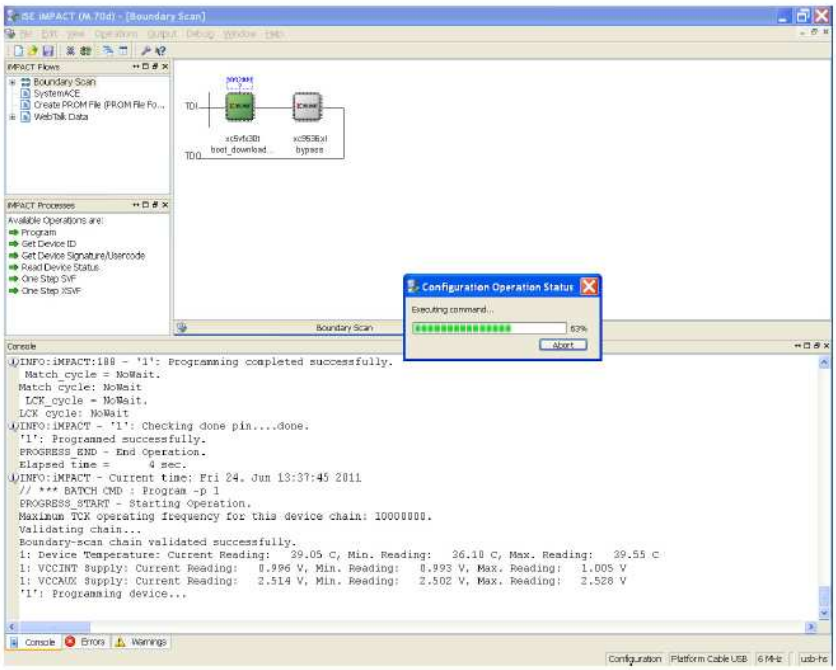
Ensure that the DIP-SW is set as shown above (position 1 OFF, the others ON).

Connect the USB cable from the SMT111 to the Host PC.

Connect the Xilinx programming pod to the SMT372T using an SMT568 JTAG cable.

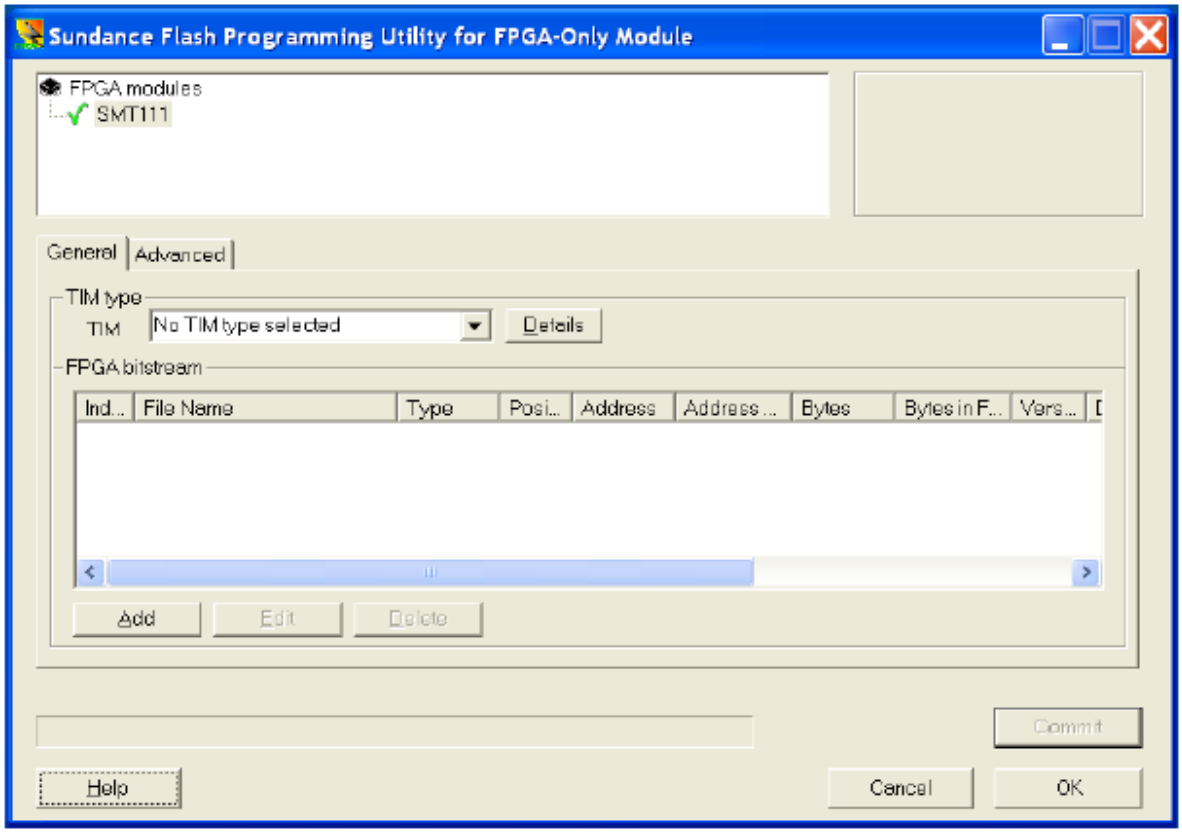


Run Xilinx impact programming tool and select boot_download.bit as the configuration file for the XC5VFX30T of the SMT372T.

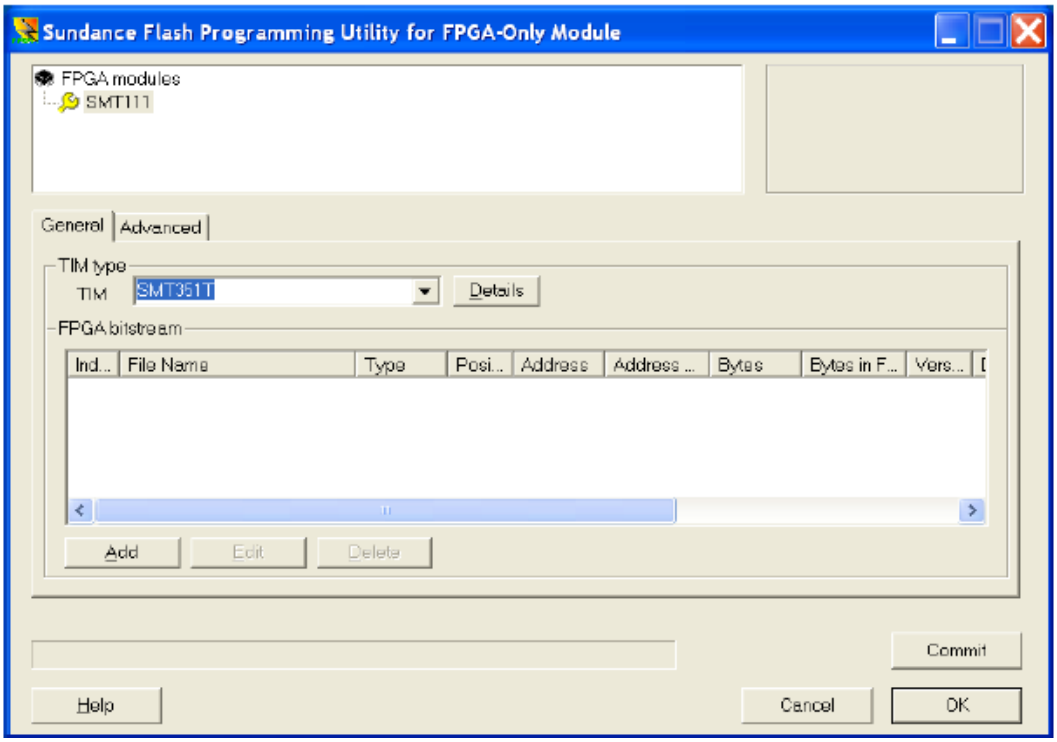


Configure the FPGA. When configuration is complete press the reset button on the SMT111. Do NOT power off the EVP system.

Run the SMT6002 application.



Select the TIM type as SMT351T.

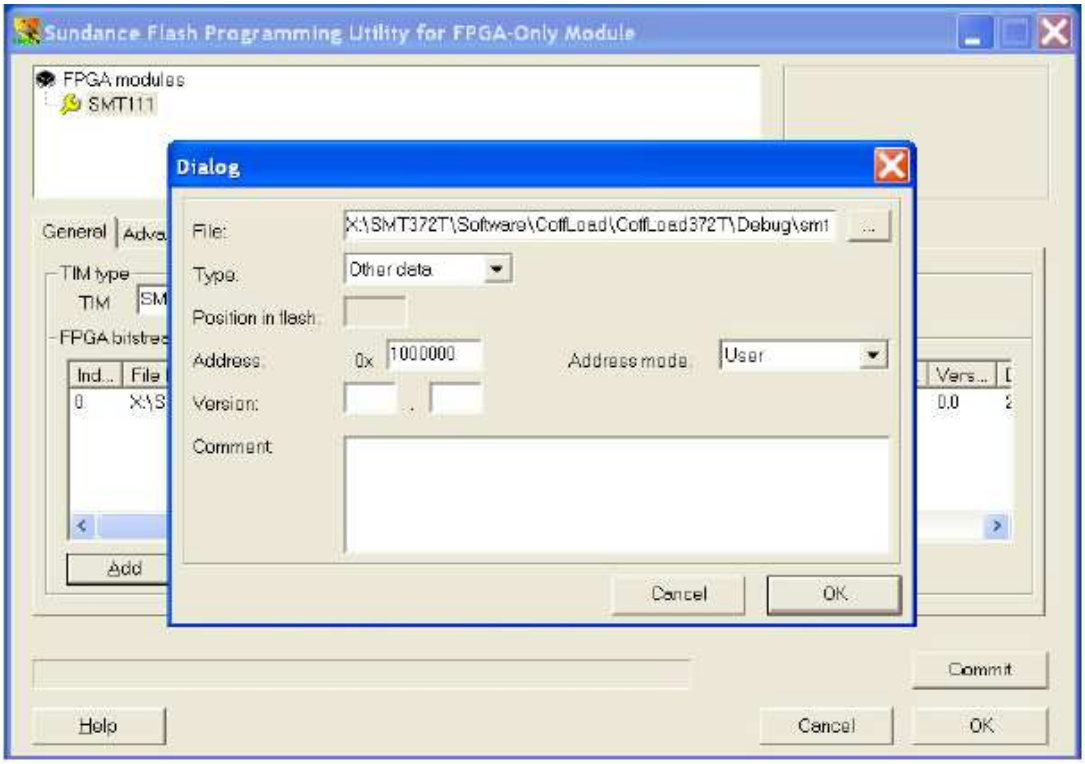


Add the boot_download.bit bitstream:

Type: Bitstream; address 0x0; address mode: basic.

Then add the smt372t.bin DSP application:

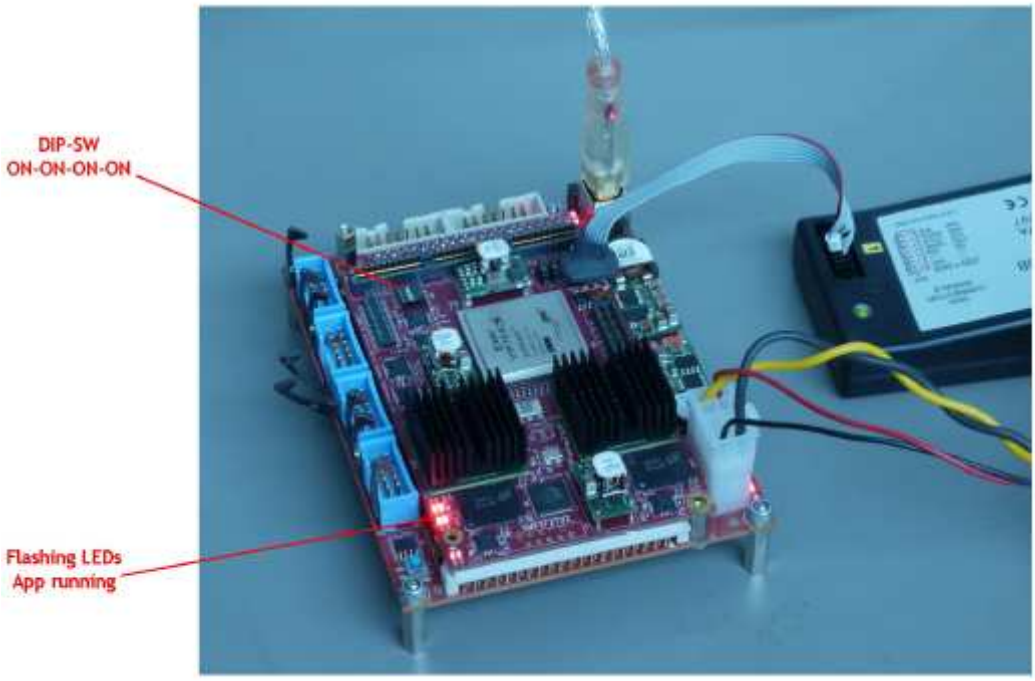
Type: other data; address 0x1000000; address mode: user.



Click OK, then Commit. When programming has finished close SMT6002.

4.2 Run the application

Set the DIP-SW as shown below (“1111” for normal application). Reset the EVP6472. Note the LEDs indicated flash for a couple of seconds.



Power off the board, plug in the SMT939, and connect two DVI ports with one input and one output.

Set the DVI input to 800x600 resolution (this application only supports 800×600 resolution, user can modify the code to get a higher resolution)

Power on the board, press the reset button to reset the DSP.

Picture is displayed on the monitor.