



Configurable Space Microsystems Innovations & Applications Center

Tutorial 13

Image Generation, ISE 10.1 on the Digilent Spartan-3E board

Introduction

This lab will be an introduction on how to establish an interconnection between a LCD screen and FPGA kit through VGA port to display stored images. The board will be a Digilent Spartan-3E starter kit.

Objective

The objective is to implement a VGA controller in a FPGA to generate images on VGA monitors. As an application to show this implementation a very simple encryption algorithm is written. The aim is not to implement complex encryption algorithms or to employ high end animations but to show the user a method to utilize VGA display ports in order to output desired video on to it.

Prerequisites

- Basic knowledge about digital design and FPGAs.
- Acquaintance with Xilinx ISE 10.1 tool.

Application

This document is used by students, who are learning FPGA design and verification.

Encryption Algorithm Overview

The encryption algorithm implemented in such a way that the colors of the original image change depending on the three bit code entered by the user.

A predefined three bit code is set in the design. Every time the user fails to enter the right three bit code using the sliding switches provided on the board (SW0, SW1 & SW2) the colors of the encrypted image will change. The truth table in Table 1 depicts this function.

Fixed code – only one bit (pw)	User input (U)	Input pixel (P)	Output pixel (O)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Table 1: Encryption Truth Table

The original image appears on the left side of the screen and the encrypted image appears on the right side of the screen as shown in Figure .



Figure 1: Displayed Image

Video Graphics Array

The term Video Graphics Array (VGA) refers either to an analog computer display standard, the 15-pin D-subminiature VGA connector, or the 640×480 resolution itself. VGA is referred to as an "array" instead of an "adapter" because it was implemented from the beginning as a single chip, replacing the Motorola 6845 and dozens of discrete logic chips covering a full-length ISA board that the MDA, CGA, and EGA used. This also allowed it to be placed directly on a PC's motherboard with a minimum difficulty (it only required video memory, timing crystals and an external RAMDAC).

1. The VGA specifications are as follows:

- 256 KB Video RAM
- 16-color and 256-color modes
- 262144-value color palette (six bits each for red, green, and blue)
- Selectable 25.2 MHz or 28.3 MHz master clock
- Maximum of 720 horizontal pixels
- Maximum of 480 lines
- Refresh rates at up to 70 Hz
- Vertical Blanking interrupt (Not all cards support this.)
- Planar mode: up to 16 colors (4 bit planes)
- Packed-pixel mode: 256 colors (Mode 13h)
- Hardware smooth scrolling support
- Some "Raster Ops" support
- Barrel shifter
- Split screen support
- 0.7 V peak-to-peak
- 75 ohm impedance (9.3mA - 6.5mW)

The VGA supports both All Points Addressable graphics modes, and alphanumeric text modes. Standard graphics modes are:

- ◆ 640×480 in 16 colors
- ◆ 640×350 in 16 colors
- ◆ 320×200 in 16 colors
- ◆ 320×200 in 256 colors

As well as the standard modes, VGA can be configured to emulate many of the modes of its predecessors (EGA, CGA, and MDA).

2. DB-15 Connector

DB-15 connector (female) appears as shown in Figure 2.



Figure 2: VGA port (on the video generator)

The pin numbering on the connector is as follows:

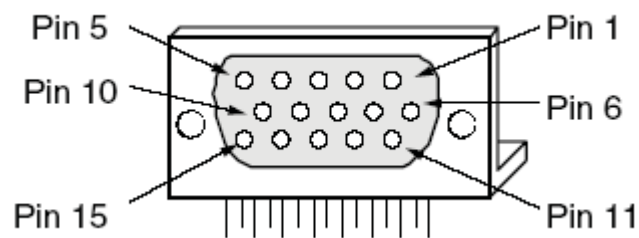


Figure 3: DB15 VGA Connector (front view)

Pin Description of VGA connector:

PIN	SIGNAL NAME	DESCRIPTION
1	RED	Red video signal
2	GREEN	Green video signal
3	BLUE	Blue video signal
4	MONID(0)	Monitor ID signal 0
5	GND DDC	Return
6, 7, 8	AGND_VID	Analog video ground
9	+5V_IO 5 V	Power for I/O device
10	GND	HSYNC and VSYNC ground
11	VGA_ID	VGA ID signal
12	MONID(2)	Monitor ID signal 2
13	HSYNC	Horizontal synchronization signal
14	VSYNC	Vertical synchronization signal
15	MONID(1)	Monitor ID signal 1

Table 2: Pin Description

3. VGA Display Port on the Spartan-3E Kit

The Spartan-3E starter kit board has a VGA display port via a DB15 connector. This port can be directly connected to most PC monitors or flat-panel LCDs using a standard monitor cable.

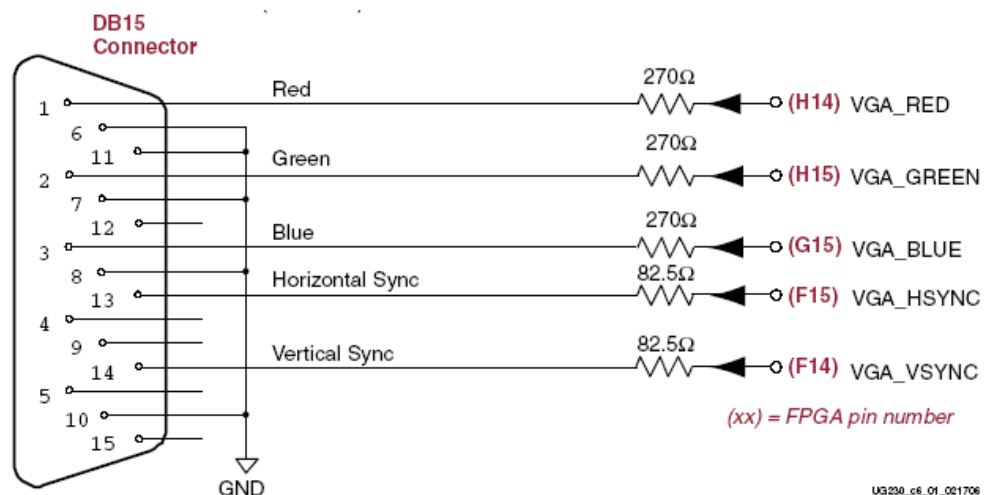


Figure 4: VGA Connections from Spartan-3E Starter Kit Board

The Spartan-3E FPGA directly drives the five VGA signals via resistors. Each color line has a series resistor, with one bit each for VGA_RED, VGA_GREEN and VGA_BLUE. The series resistor, in combination with the 75 Ω termination built into the VGA cable, ensures that the color signals remain in the VGA-specified 0V to 0.7V range. The VGA_HSYNC and VGA_VSYNC signals using LVTTTL or LVCMOS33 I/O standard drive levels. Drive the VGA_RED, VGA_GREEN and VGA_BLUE signals high or low to generate the eight colors shown in Table 3.

VGA_RED	VGA_GREEN	VGA_BLUE	Resulting Color
0	0	0	Black
0	0	1	Blue
0	1	0	Green
0	1	1	Cyan
1	0	0	Red
1	0	1	Magenta
1	1	0	Yellow
1	1	1	White

Table 3: 3-Bit Display Color Code

4. Signal Timing for a 60Hz, 640X480 VGA Display

CRT-based VGA displays use amplitude-modulated, moving electron beams (or cathode rays) to display information on a phosphor-coated screen. LCDs use an array of switches that can impose a voltage across a small amount of liquid crystal, thereby changing light permittivity through the crystal on a pixel-by-pixel basis. Although the following description is limited to CRT displays, LCDs have evolved to use the same signal timings as CRT displays. Consequently, the following discussion pertains to both CRTs and LCDs.

Within a CRT display, current waveforms pass through the coils to produce magnetic fields that deflect electron beams to transverse the display surface in a *raster* pattern, horizontally from left to right and vertically from top to bottom. As shown in Figure 5, information is only displayed when the beam is moving in the *forward* direction—left to right and top to bottom—and not during the time the beam returns back to the left or top edge of the display. Much of the potential display time is therefore lost in *blanking* periods when the beam is reset and stabilized to begin a new horizontal or vertical display pass.

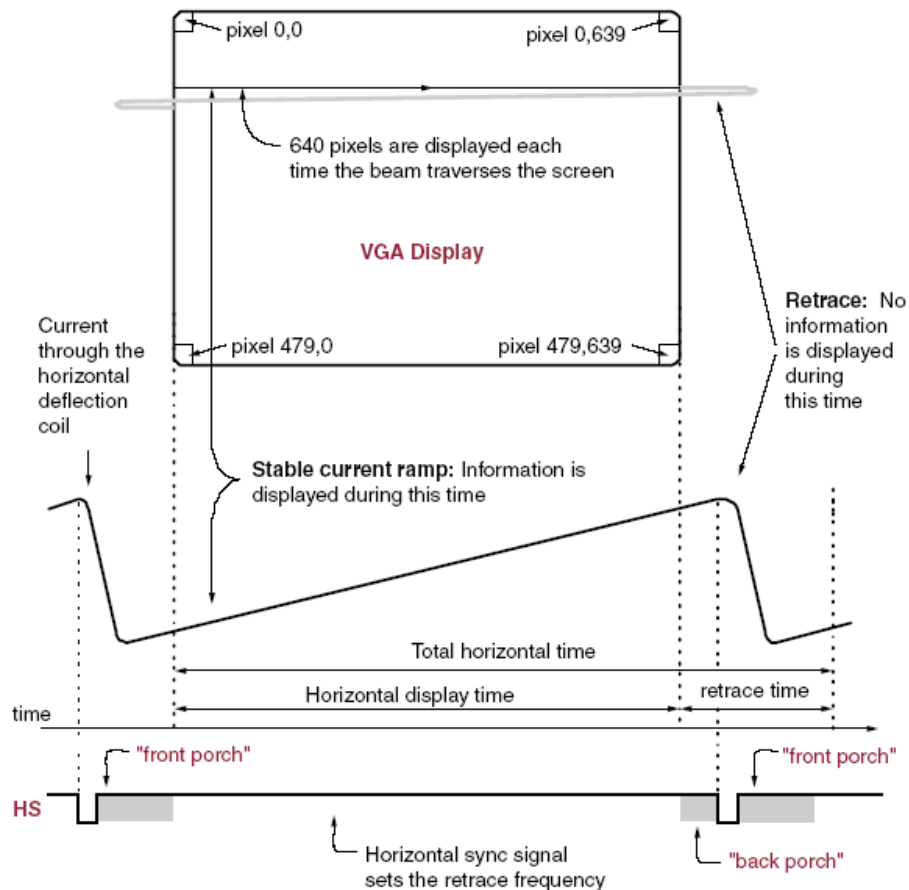


Figure 5: CRT Display Timing Example

The display resolution defines the size of the beams, the frequency at which the beam traces across the display and the frequency at which the electron beam is modulated.

Modern VGA displays support multiple display resolutions and the VGA controller dictates the resolution by producing timing signals to control the raster patterns. The controller produces TTL-level synchronizing pulses that set the frequency at which current flows through the deflection coils, and it ensures that pixel or video data is applied to the electron guns at the correct time.

Video data typically comes from a video refresh memory with one or more bytes assigned to each pixel location. The Spartan-3E Starter Kit board uses three bits per pixel, producing one of the eight possible colors shown in Table 3. The controller indexes into the video data buffer as the beams move across the display. The controller then retrieves and applies video data to the display at precisely the time the electron beam is moving across a given pixel.

As shown in Figure 5, the VGA controller generates the horizontal sync (HS) and vertical sync (VS) timing signals and coordinates the delivery of video data on each pixel clock.

The pixel clock defines the time available to display one pixel of information. The VS signal defines the *refresh* frequency of the display or the frequency at which all information on the display is redrawn. The minimum refresh frequency is a function of the display's phosphor and electron beam intensity, with practical refresh frequencies in the 60 Hz to 120 Hz range. The number of horizontal lines displayed at a given refresh frequency defines the horizontal *retrace* frequency.

5. VGA Signal Timing

The signal timings in Table 4 are derived for a 640-pixel by 480-row display using a 25 MHz pixel clock and 60 Hz ± 1 refresh. Figure 6 shows the relation between each of the timing symbols. The timing for the sync pulse width (TPW) and front and back porch intervals (TFP and TBP) are based on observations from various VGA displays. The front and back porch intervals are the pre- and post-sync pulse times. Information cannot be displayed during these times.

Symbol	Parameter	Vertical Sync			Horizontal Sync	
		Time	Clocks	Lines	Time	Clocks
T_S	Sync pulse time	16.7 ms	416,800	521	32 μ s	800
T_{DISP}	Display time	15.36 ms	384,000	480	25.6 μ s	640
T_{PW}	Pulse width	64 μ s	1,600	2	3.84 μ s	96
T_{FP}	Front porch	320 μ s	8,000	10	640 ns	16
T_{BP}	Back porch	928 μ s	23,200	29	1.92 μ s	48

Table 4: 640X480 Mode VGA Timing

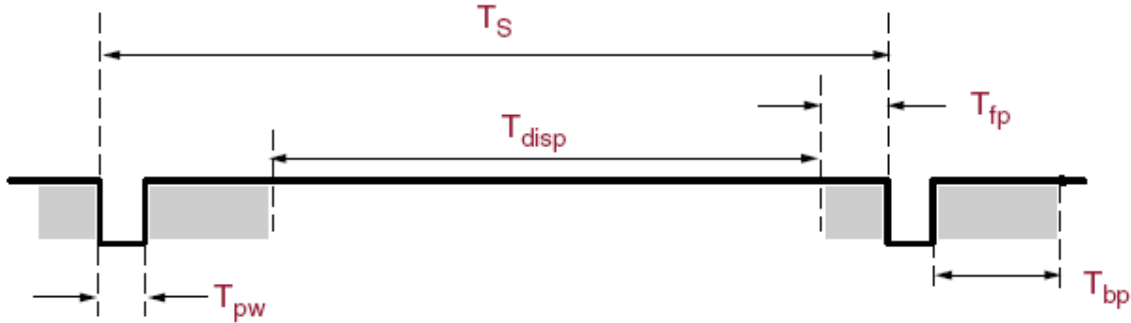


Figure 6: VGA Control Timing

Generally, a counter clocked by the pixel clock controls the horizontal timing. Decoded counter values generate the HS signal. This counter tracks the current pixel display location on a given row.

A separate counter tracks the vertical timing. The vertical-sync counter increments with each HS pulse and decoded values generate the VS signal. This counter tracks the current display row. These two continuously running counters form the address into a video display buffer. For example, the on-board DDR SDRAM provides an ideal display buffer.

No time relationship is specified between the onset of the HS pulse and the onset of the VS pulse. Consequently, the counters can be arranged to easily form video RAM addresses, or to minimize decoding logic for sync pulse generation.

Present Design Description

The design objective is to display desired images on to a VGA monitor. A simple encryption algorithm as explained earlier is used as an application to implement this design.

This design lets the user operate at 25 MHz frequency and 640X480 resolution. The design uses the 50 MHz system clock present on the Spartan-3E board.

To use the design for other frequencies and resolutions the horizontal constants and vertical constants need to be modified as shown below.

- **Horizontal constants:**

1. 800X600 and 40 MHz
Hactive <= "001100011111"
HFP <= "001101000111"
Hsynch <= "001111000111"
Hin <= "001110000111"
HBP <= "010000011111"
2. 1024x768 and 65MHz
Hactive <= "001111111111"
HFP <= "010000010111"
Hsynch <= "010010011111"
Hin <= "010001011011"
HBP <= "010100111111"
3. 1280x1024 @ 110MHz
Hactive <= "010011111111"
HFP <= "010100110011"
Hsynch <= "010110101011"
Hin <= "010101101111"
HBP <= "011010101011"
4. 1600x1200 @ 160MHz
Hactive <= "011000111111"
HFP <= "011001110111"
Hsynch <= "011100110111"
Hin <= "011011010111"
HBP <= "100001011111"

- **Vertical constants:**

1. 800X600 and 40 MHz
Vactive <= "01001010111"
VFP <= "01001011000"
Vsynch <= "01001011100"
VBP <= "01001110011"
2. 1024x768 and 65MHz
Vactive <= "01011111111"
VFP <= "01100000010"
Vsynch <= "01100001000"
VBP <= "01100100101"
3. 1280x1024 @ 110MHz
Vactive <= "01111111111"
VFP <= "10000000010"
Vsynch <= "10000000111"
VBP <= "10000110001"
4. 1600x1200 @ 160MHz
Vactive <= "10010101111"
VFP <= "10010110000"
Vsynch <= "10010110011"
VBP <= "10011011011"

IO Ports:

- Inputs:

Port	Width	Description
clk	1	Clock signal (system clock – 50 MHz)
reset	1	Reset (sliding switch – SW3)
U	3	User input password (sliding switches – SW0, SW1 & SW2)

Table 5: Input Signals

- Outputs:

Port	Width	Description
hsync	1	Horizontal synchronism of the VGA
vsync	1	Vertical synchronism of the VGA
VGA_out_red	4	Red signal to the FPGA
VGA_out_green	4	Green signal to the FPGA
VGA_out_blue	4	Blue signal to the FPGA

Table 6: Output Signals

Memory File Creation:

Memory block carrying the pixel data of the intended image need to be created. This memory will be placed in the internal memory of the FPGA.

- a. Click on **Create New Source**.
- b. Highlight on **IP (Coregen & Architecture Wizard)** in the left and call the source **mymemory** (Figure 7). Click on **Next**.

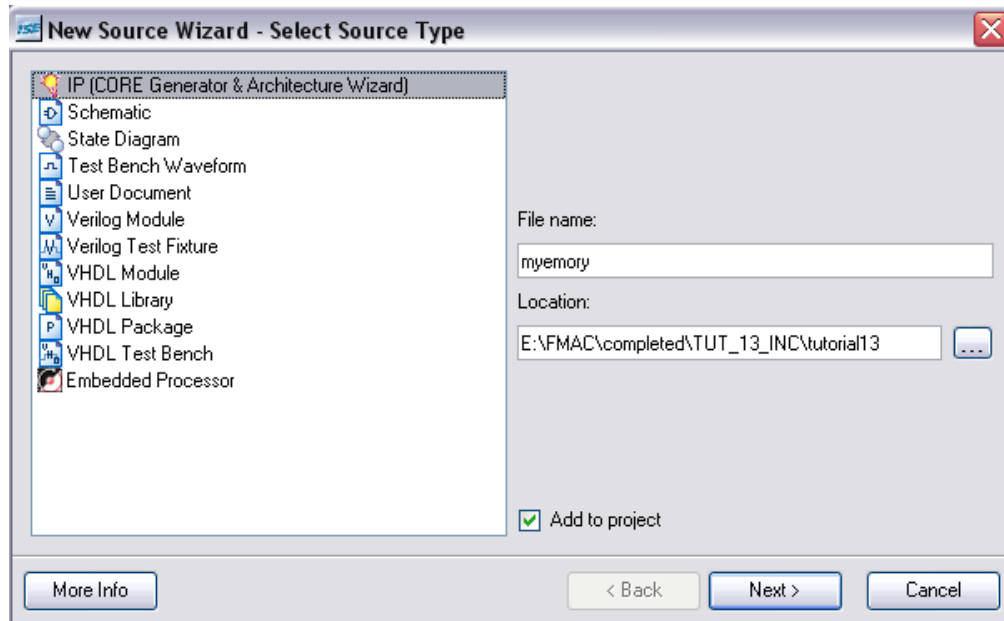


Figure 7: New Source Wizard

- c. Select **Block Memory Generator v2.8** in the submenu **RAMs & ROMs** under the **Memories & Storage Elements** option (Figure 8). Select **Next**, then **Finish**.

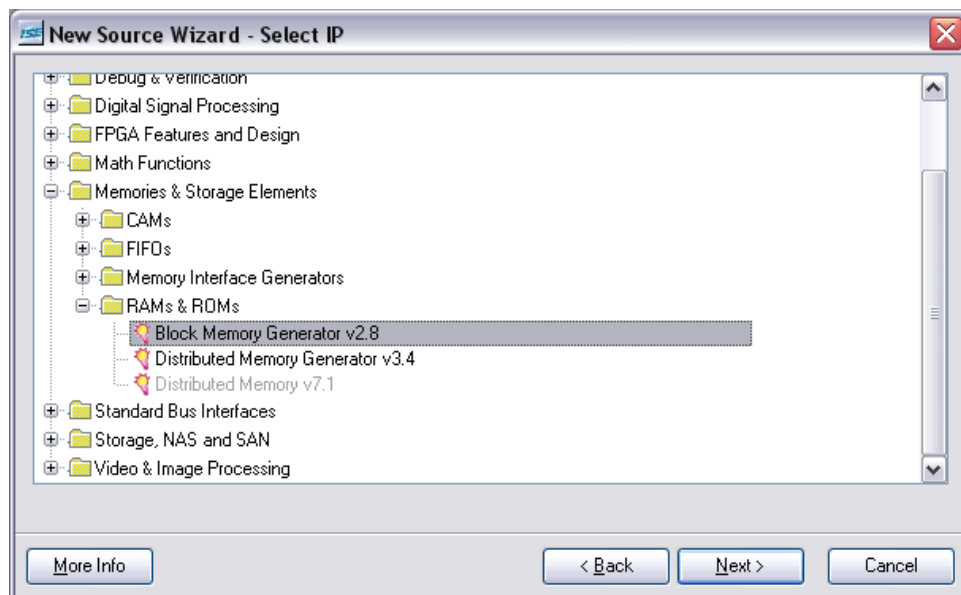


Figure 8: IP Selection

d. In the next window select the characteristics of the memory (Figure 9). Since no data will be stored in the memory, choose the option **Single Port Rom** under **Memory Type**. Then click **Next**. The image that is used in the present design has 224 rows and 224 columns (224x224 image) and only 1 bit per color is used. The VGA controller uses three main colors (Red, Green and Blue), the **width** of the memory is set to **3** and the **depth** is set to $224 \times 224 = 50176$ (50176 possible locations). Click **Next** one time until **Page 3 of 4** (Figure 10) appears.

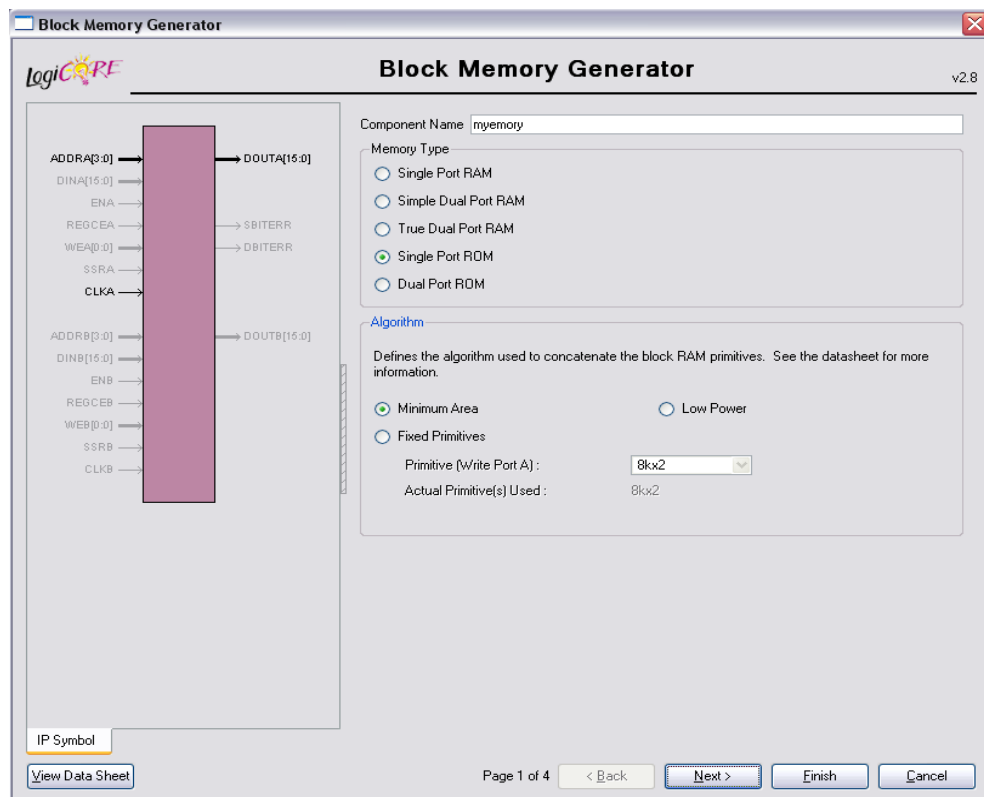


Figure 9: Single Port Block Memory

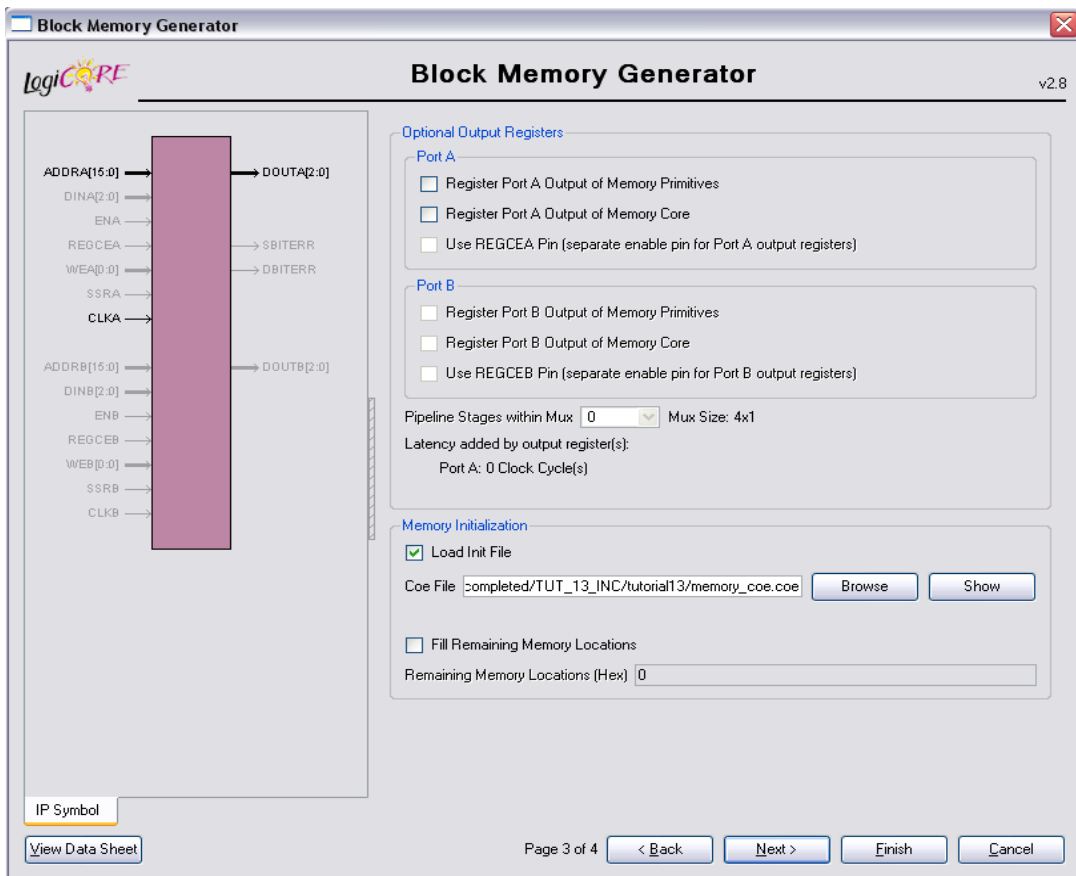


Figure 10: Single Port Block Memory. Page 3 of 4

e. In this window, load the memory with the values of the image (bart_simpson.bmp --- provided with the project). The file **memory_coe.coe** (created using the matlab file generate_simpson.m --- provided with the project) has all the values stored in the format that Xilinx uses for its memories. Check the option **Load Init File** and then click on **Load File**. Look for the file **memory_coe.coe** and open it.

f. Click on **Finish** to create the block. This process could take some minutes. You will know that Xilinx have finished with the block generation when you have the source added to your project (Figure 11).

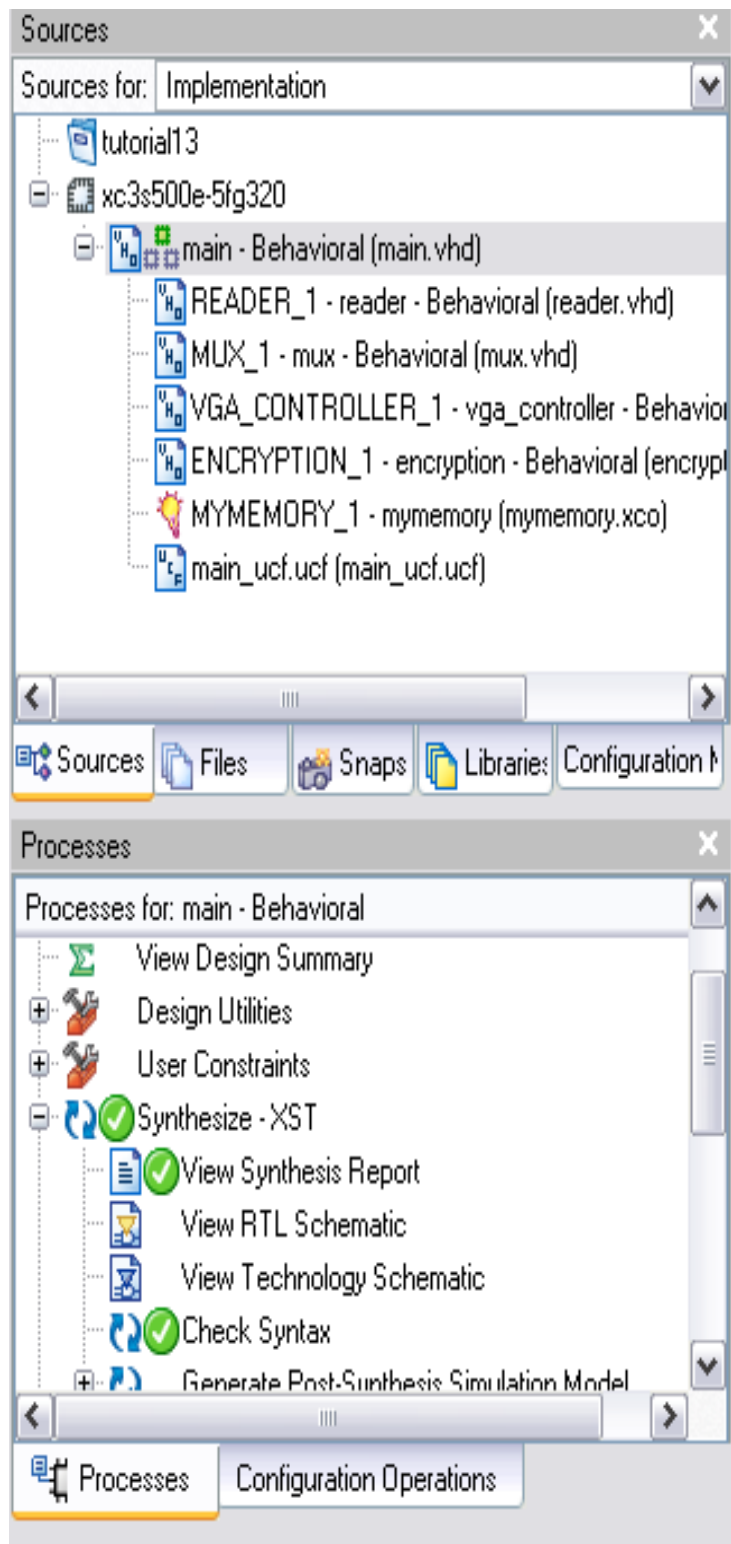


Figure 11: Generation of the Memory

Initial Set-up and Output

Connect the Digilent Spartan-3E kit to a laptop/PC as shown below.



Figure 12: USB Connection

Then connect the monitor to VGA connector (the left most connector along the top of the board) available on the FPGA board and then switch on the monitor. Now switch on the board and configure the device using the Xilinx ISE 10.1 tool.

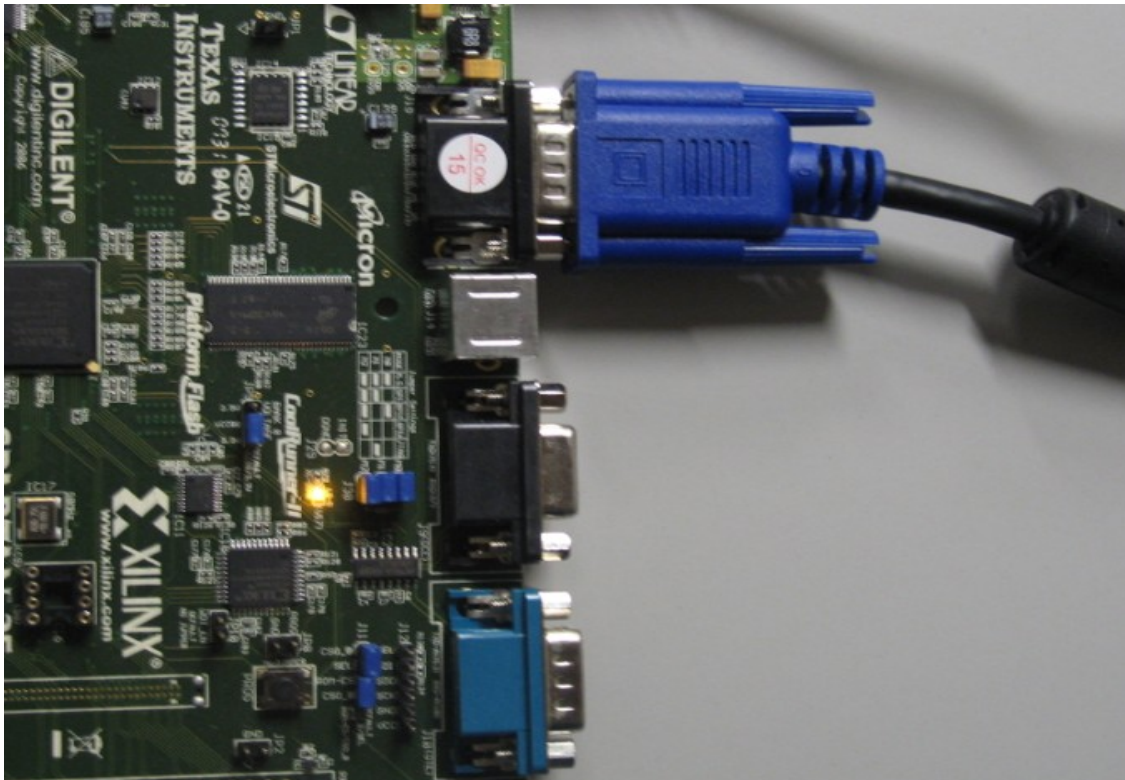


Figure 13: VGA Connector

Now following image can be observed on the monitor after setting the user code to “010” (SW0 = Low, SW1 = High, SW2 = Low and SW3 (reset) = Low).

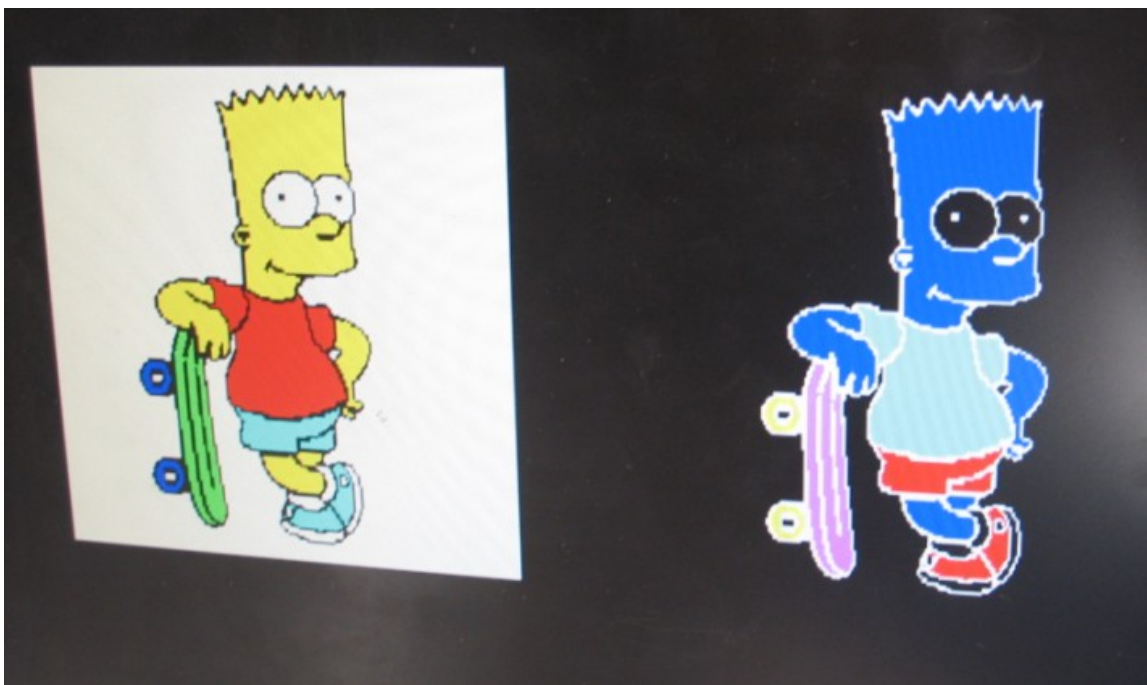


Figure 14: Output

Exercise

1. Try modify the present design to make the image move in desired direction (upwards, downwards, to the right or left) using external inputs (may be through push buttons or sliding switches).

Note: Make sure the push buttons are de-bounced in case push buttons are used to control the directions.

Solution is provided.

2. Now try to stream a small video in similar method.

Note: Need to modify the Matlab file appropriately to obtain the pixel information and the reader.vhd to the used video specifications. Also need to get a video which uses only the 8 colors (which can be represented by Spartan-3E board) in order to get a clean output.

-Author Bio

Name: Vallabh Srikanth Devarapalli
Graduate student at UNM in ECE department.
E-mail: vsdevara@unm.edu.

-Updated By

Name: Brian Zufelt
Undergraduate student at UNM in ECE department