COL215 – Digital Logic and System Design
Department of Computer Science & Engineering, IIT Delhi
Semester I, 2025-26
Setup Instructions for VGA display controller on Basys3 board

1 Storing the image on basys 3 memory

Use the following test bench to simulate the memory IP block in vivado. The test bench is an illustration of how to import the generated memory block into your verilog design file.

```
`timescale 1ns / 1ps

module tb_ROM_block;
// Inputs
reg clock = 0;
reg [13:0] rdaddress = 14'd0;

// Outputs
wire [7:0] data;

// Instantiate the Unit Under Test (UUT)
blk_mem_gen_0 uut (
    .clka(clock),
    .addra(rdaddress),
    .douta(data)
);

// Clock generation
localparam clock_period = 10;
always begin
    #(clock_period/2) clock = ~clock;
end

// Stimulus process
integer i;
initial begin
    for (i = 0; i < 16384; i = i + 1) begin
        rdaddress = i;
        #20;
    end
    $stop;
end
endmodule
```

2 Overview of image gradient operation

Grayscale 8-bit image is a 2-D matrix storing pixel values between 0-255, as shown in 1. Size of the image will be 256x256. Image should be stored in 1-D array format in block RAM, that from address 0 (0000_{16}) in row major format.

Objective is to perform image operation to extract the vertical edges in the given image. To get the edges in the image, gradient of the image need to be calculated. A 3x1 filter (shown in red in Figure 1) is used to calculate the gradient of the image.

2.1 Steps to calculate the gradient

- Filter stride along the row of the image to calculate the resultant pixels in the output image.
- Output pixel value is summation of element-wise multiplication between filter value and input image pixel as shown in given equation. Input image pixel at location (i,j) is denoted via $I(i,j)$, output pixel as $O(i,j)$.

$$O(i,j) = 1 * I(i,j-1) - 2 * I(i,j) + 1 * I(i,j+1)$$

When column index is $j < 0$ or $j > 255$, then assume the pixel value as 0, this will be applicable while calculating the border pixel value in the output image. Few illustrations:

- Case I:

$$O(0,0) = 1 * I(0,-1) - 2 * I(0,0) + 1 * I(0,1)$$
$$O(0,0) = 0 - 20 + 10 = -10$$

.

- Case II:

$$O(0,1) = 1 * I(0,0) - 2 * I(0,1) + 1 * I(0,2)$$
$$O(0,0) = 10 - 20 + 10 = 0$$

- Case III:

$$O(0,2) = 1 * I(0,1) - 2 * I(0,2) + 1 * I(0,3)$$
$$O(0,0) = 10 - 20 + 220 = 210$$

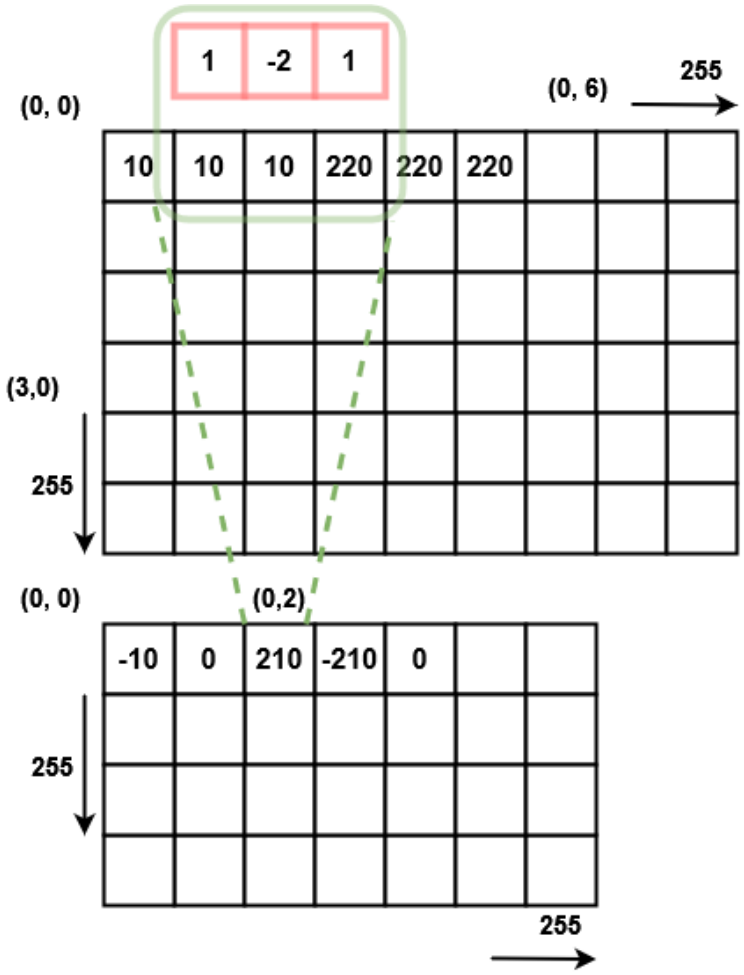


Figure 1: Illustration to perform 1D operation on grayscale image

In the final output image, negative pixel values shall be clamped to zero and any pixel value greater than 255 clamped to 255. This is done to ensure the output image is stored in 8-bit unsigned format.

2.2 Image gradient verilog module

Task is to design a module which will read image from the block RAM and calculate the image gradient as describe in section 2. The resultant image should be stored into new block RAM and displayed on the VGA monitor. All the components are describe in the Figure 2.

To calculate the output pixel, three input pixel values are required. But in one clock cycle you can read only one address from block RAM. Naive way is to read each input pixel in 3 clock cycle and calculate the output. Other way is to use temporary small storage element (Registers) to store all three pixel values and thus avoid extra clock cycles, this is shown in figure 2 (highlighted in red box). **It is not compulsory to follow the above method, you are free to design your own module.**

1. a Read-Only Memory (ROM) - You will be using this memory to store the input image.
2. Registers - You will be using these to store temporary results across clock cycles.
3. a Read-Write Memory (RWM, more popularly known as RAM or Random-Access Memory) - You will be using this memory to store output image.
4. Adder - to perform summation of three resultant pixel values.
5. VGA display controller: To drive the VGA port on basys 3 board.

2.2.1 Memories

The design requires memories to store the input and the output image. We will store the input image in a read-only memory (ROM) and store the output image in random-access memory (RAM). The memory will be used to store the following:

- Input Image: 256x256 image with each pixel of 8 bits or 1B. For storing the input we need 65536 words, 8 bits wide. These will be stored at address 0 (0000_{16}) onwards.
- Output Image: 256x256 image with each pixel of 8 bits or 1B. For storing the output we need 65536 words, 8 bits wide. These will be stored at address 0 (0000_{16}) onwards.

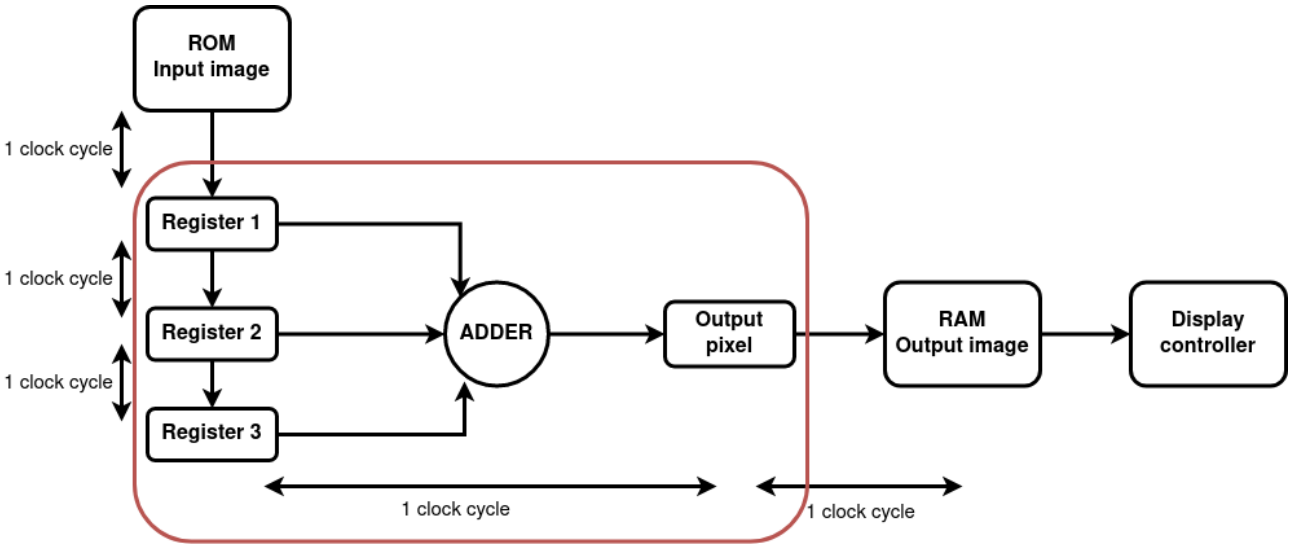


Figure 2: Block diagram component-wise

3 Setting up VGA display on Basys 3 board

In the current assignment, you will have to display the output image on a monitor. You will be designing display controller module in verilog, to drive externally connected VGA display. VGA is acronym for video graphics

array, connector has 15 pins, three lines for RGB, 2 for sync signals and rest are ground pins to regulate the current. Refer to Basys3 reference manual, section 7.1 for pin configuration (Basys3 manual).

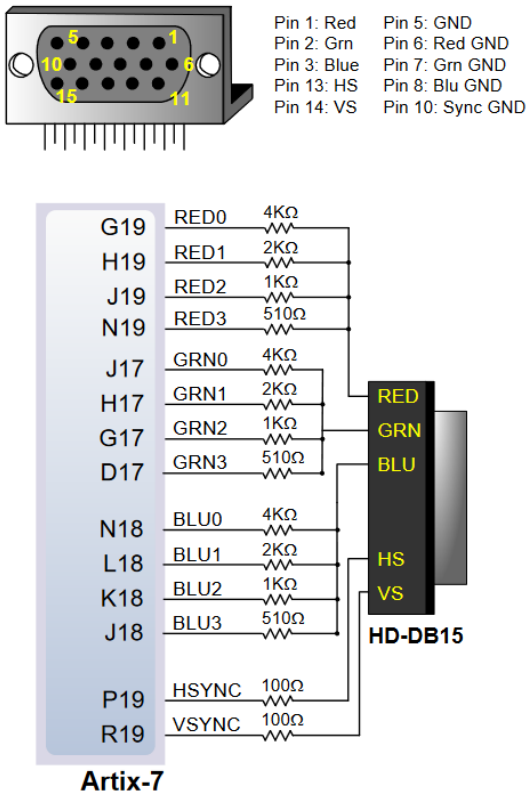
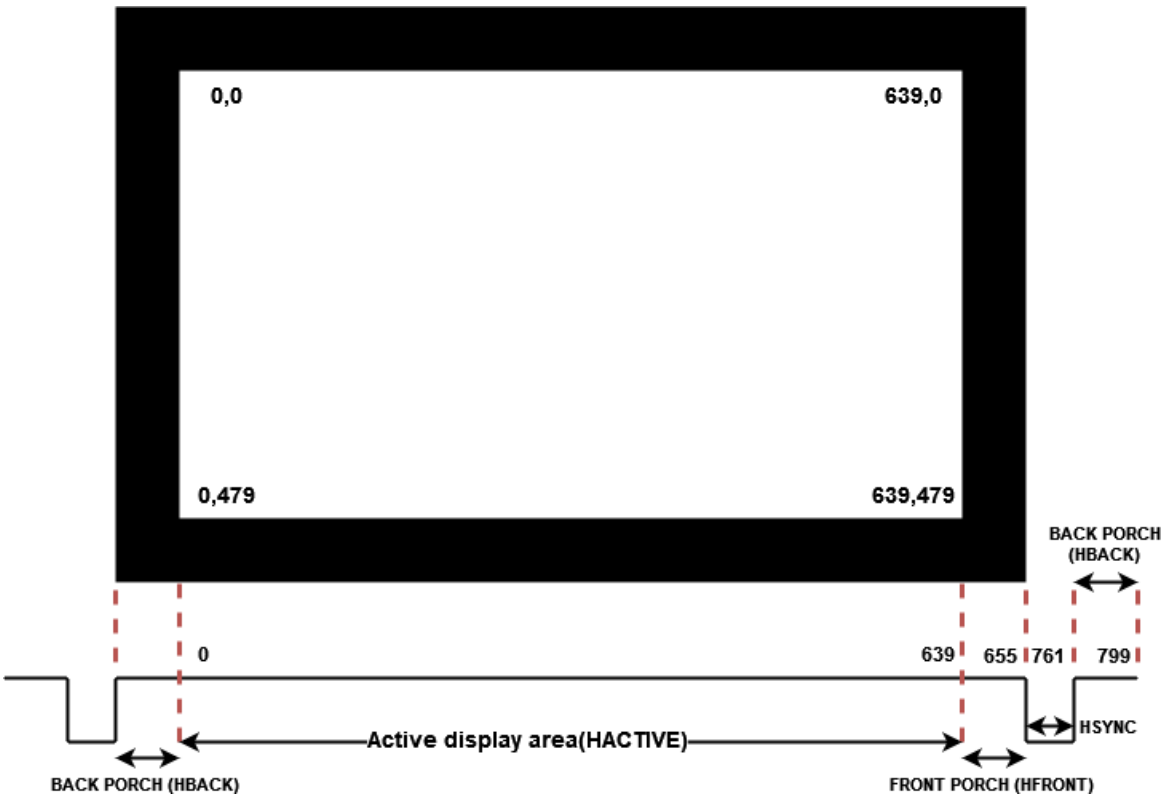
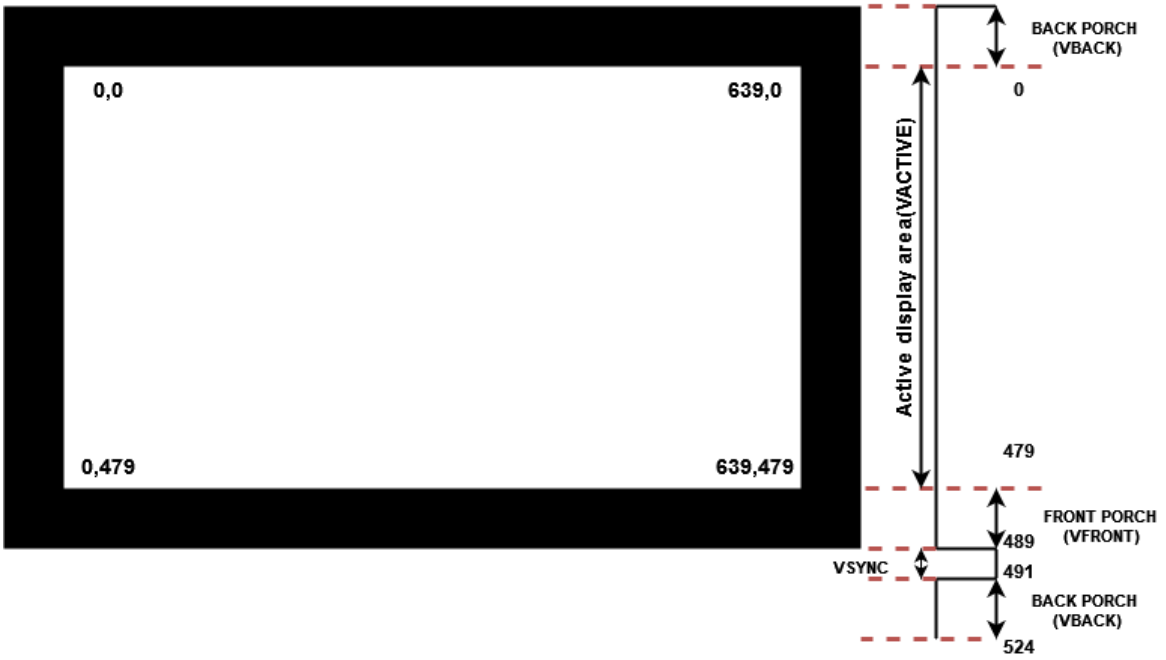


Figure 3: Pin connector for VGA basys 3 board

Each display has certain timings, as shown in the Figure 4. Timings are separated into horizontal and vertical segments. It consists of active video line, front and back porch and sync signals. To understand how the working of old CRT display, refer to CRT link. Even though the devices have been upgraded, the timing specification remains the same. Video timing has an active area where the actual video/image data is seen and blank timings in which internal circuit (electron beam in CRT) needs to reset.



(a) Horizontal pixels



(b) Vertical lines

Figure 4: Display timings diagram

There are various resolution and timing supported via display, but for now only one display timing **(640*480@60Hz)** has to be implemented which is defined in the Table 1.

Table 1: Timing specification for 640x480@60Hz

Parameter	Value
Pixel clock	25 MHz
HSYNC	96 pixels
HBACK	48 pixels
HFRONT	16 pixels
HACTIVE	640 pixels
VSYNC	2 lines
VBACK	33 lines
VFRONT	10 lines
VACTIVE	480 lines

Total number of pixels in one horizontal line (active+blanking) will be :

$$HTOT = HACTIVE + HBACK + HFRONT + HSYNC$$

$$800 = 640 + 48 + 16 + 96$$

Total number of lines in one frame will be :

$$VTOT = VACTIVE + VBACK + VFRONT + VSYNC$$

$$525 = 480 + 33 + 10 + 2$$

Total number of pixels in one frame are, **420000** which has to be displayed in fixed amount of time that is **16.66 ms** (60 Hz or 60 frames per seconds). Thus, time period for the clock for one pixel will be (25.2 MHz):

$$420000 * 60 = 25200000$$

Display controller need input clock of 25 MHz and generate corresponding HSYNC and VSYNC signal. To accomplish this, three module need to be made: (i) clock divider, (ii) Horizontal pixel counter and (iii) Vertical line counter. A block diagram of the modules with input and output signals is given in the Figure 5.

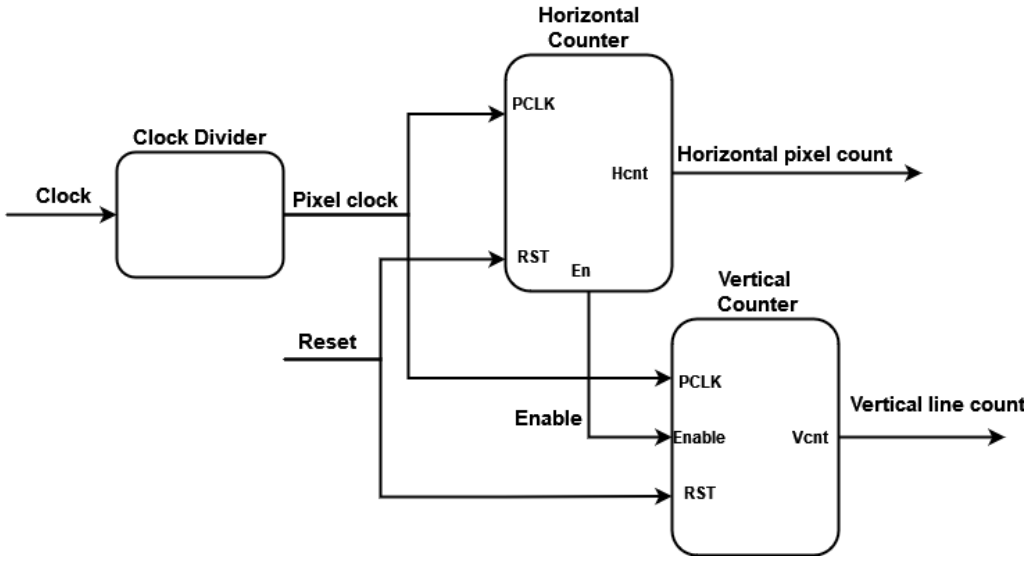


Figure 5: Module diagram for display controller

3.1 Clock divider

The clock divider will generate the 25 MHz pixel clock using the onboard board clock of 100 MHz. Essentially, board clock need to be divided to get required pixel clock.

3.2 Horizontal pixel counter

The horizontal pixel counter will keep track of pixels in the line, input will be pixel clock and reset signal. Output will be pixel count (Hcnt) and line complete signal (indicating end of the line, which will serve as enable signal to the vertical counter). Counter will start from 0 and increment till HTOT-1 then reset the count to 0.

3.3 Vertical line counter

The vertical line counter will keep track of lines in a frame, input will be pixel clock, reset and enable signal, Output will be line count (Vcnt). Counter will start from 0 and increment till VTOT-1 then reset the count to 0.

3.4 Display controller logic

Based on the **Hcnt**, **HSYNC** signal will be driven **HIGH** for the pixel count **0** to **HACTIVE** + **HFRONT** and changed to **LOW** for duration **HYSNC**. Similarly, based on **Vcnt**, **VSYNC** will be driven **HIGH** from line **0** to **VACTIVE** + **VFRONT** and **LOW** for **VSYNC** duration. The signal transition is shown in the Figure .

Additionally, image data need to be sent during HACTIVE and VACTIVE period. Image pixel location will be calculated based on Hcnt and Vcnt signal from the counter. For example, if the image size is 256x256 then it can be displayed at any location in the frame 640x480 as shown in the Figure 6.

The given image is an 8-bit grayscale format, meaning the image contains 256 levels of grayscale colours. First, to display grayscale image using VGA RGB format, each colour line of RGB should be set to the same value that is R=G=B. But basys3 has 4-bit width per colour line, so image 8-bit value should be truncated to 4-bit value via discarding lower 4 bits (i.e. using 4 MSB bits). This will reduce the 256 colour level to 16 colour level. For example, if pixel at location (0,0) has 8-bit value **10110001** then video data in RGB format will be **1011 1011 1011** (12-bit RGB).

Above logic need to be written in main display controller module which will output HSYNC, VSYNC signals, video data signals.

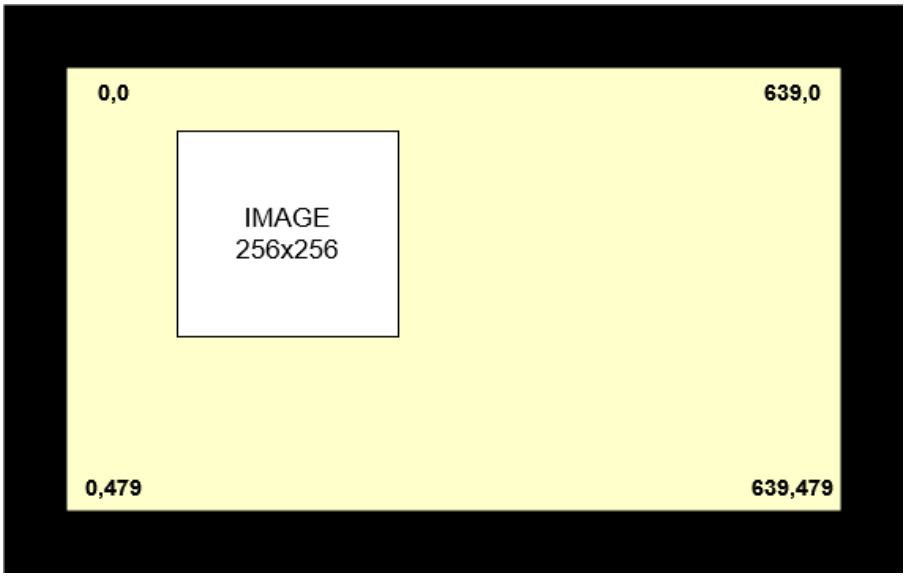


Figure 6: Positioning of image wrt display area