ECE 272 Lab #6

Video Graphics Array

Hyunjae Kim

08/11/2021

Matthew Shuman

1. Introduction

In this lab section, the general purpose is to make Video Graphics Array (VGA) with using HDLs and logic gates. To design VGA, students have to understand about the concepts of H synchronize and V synchronize, and the correlation between those two synchronizes.

After understanding the concepts and correlation of H synchronize and V synchronize, students have to design the block which outputs H synchronize signal with using counter, comparator, and synchronizer. The design of H synchronize will be used for designing synchro-counter. This synchro-counter outputs H synchronize signal and V synchronize signal. Moreover, depending on H synchronize signal and V synchronize signal, synchro-counter will output H display and V display. These will be designed by using the figures below. H display and V display has to be 1 in the interval C.

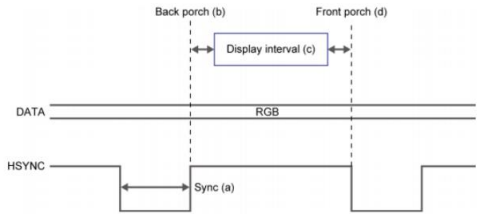


Figure 6.3 Horizontal Timing Specifications

Table 3-9 VGA Horizontal Timing Specification

VGA mode		*Horizontal Timing Spec*				
Configuration	*Resolution(HxV)*	*a(pixel clock cycle)*	*b(pixel clock cycle)*	*c(pixel clock cycle)*	*d(pixel clock cycle)*	*Pixel clock(MHz)*
VGA(60Hz)	640x480	96	48	640	16	25

Table 3-10 VGA Vertical Timing Specification

VGA mode		*Vertical Timing Spec*				
Configuration	*Resolution(HxV)*	*a(lines)*	*b(lines)*	*c(lines)*	*d(lines)*	*Pixel clock(MHz)*
VGA(60Hz)	640x480	2	33	480	10	25

Figure 6.4 Hsync and Vsync Tables

<Figure 1: Hsync and Vsync Tables>

After designing the synchro-counter, the synchro-counter will be used as a part of RGB controller (or VGA interface) with using clock modulator, decoder, and multiplexer. VGA interface has 2-bit of Red, Green, and Blue inputs, and these 2-bit inputs will be decoded and connected to multiplexer. Multiplexer will output 4-bit output of Red, Green, and Blue. The intersection of H display and V display determines the output of multiplexer, 0 or decoded 2-bit input of Red, Green, and Blue. VGA interface will be designed by using the figure below.

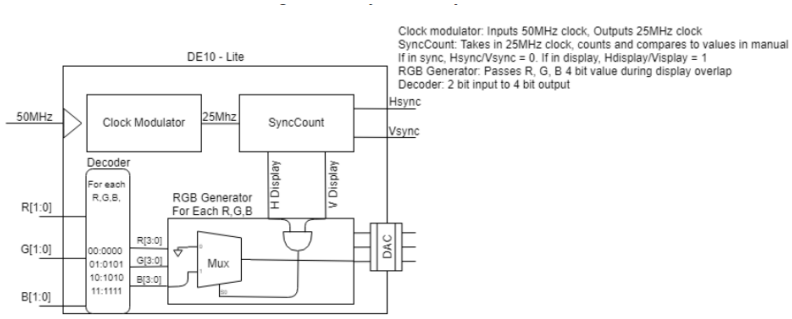


Figure 6.5 Block Diagram

<Figure 2: Provided Block Diagram of VGA interface>

When the design of VGA interface is completed, the design will be verified both hardware, DE10-Lite connected to VGA screen, and Modelsim software simulation. If both of them works as expectation, then the design is successful.

2. Design

To implement the design, DE10-Lite board and VGA screen are required. However, the situation is not sufficient for hardware verification. This means that there is no VGA screen. Thus, although there is no VGA screen in the experiment, there will be pin assignment in the design.

1) Counters
   a) Counter
      When the clock is rising edge, the counter adds 1 to the output q. Unless the reset is turned on, the counter will add 1 to the output q until it reaches $2^N - 1$. When the reset is turned on, the counter resets the output q to be 0. (Appendix A)

   b) CounterNeg
      Vsync has 2 cycle of Hsync in the interval A of Vsync. To make this happen, the counter has to start counting when it is fall edge of the clock. Thus, when the clock is falling edge, the counter adds 1 to the output q. Unless the reset is turned on, the counter will add 1 to the output q until it reaches $2^N - 1$. When the reset is turned on, the counter resets the output q to be 0. (Appendix B)

2) Comparators
   a) Comparator
      The comparator compares the input value and the parameter M. The output logic is 1 when the N-bit input value is greater than the parameter M, and if the N-bit input value is equal or less than the parameter M, then the output logic is 0. (Appendix C)

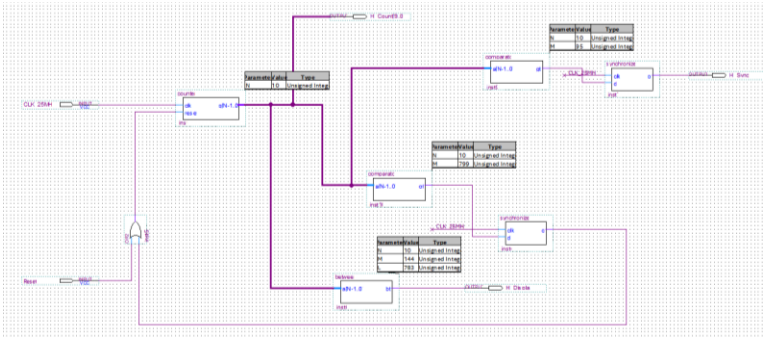
   b) Between
      The Between determines the N-bit input value whether it is between parameter M and parameter L. If the N-bit input value is in the interval [M,L], then the output logic is 1, and if the N-bit input value is not in the interval [M,L], then the output logic is 0. (Appendix D)

3) Synchronizer
   The synchronizer synchronizes the input value to input clock. To be specific, the input of synchronizer acts as clock input. Moreover, if the input has irregular matching point with the clock input, then it is complicated to perform the expected result. The synchronizer synchronizes the input which acts as the clock to the clock input in order to perform the expected result. (Appendix E)

4) Decoder
   The decoder the decodes 2-bit inputs and then outputs 4-bit value. If the input is 00, then the output is 00000, and if the input is 01, then the output is 0101. If the input is 10, then the output is 1010, and if the input is 11, then the output is 1111. This decoder is connected to each 2-bit input: Red, Green, and Blue. (Appendix F)

5) Multiplexer
   Multiplexer chooses the output among two 1-bit D values depending on S input. If S input is 0, then the output will be 4-bit input a, and if S input is 1, then the output will be 4-bit input b. (Appendix G)

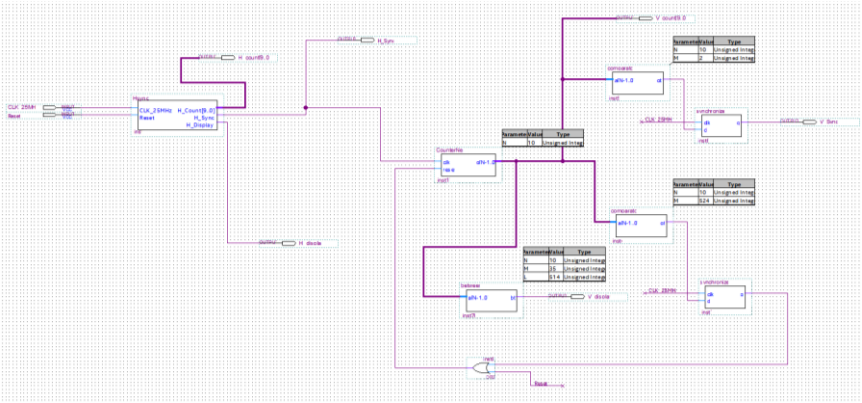
6) H-sync



<Figure 3: Design of H-sync on Quartus Software>

Hsync creates H synchronize signal which is LOW for 96 cycles and HIGH for 704 cycles. It uses 25MHz clock input and the counter counts until 10bits value, and then the upper comparator compares the input value, and then outputs 1 when the input value is greater than 95. Moreover, the lower comparator outputs 1 when the input value is greater than 799. Both synchronizer blocks synchronize the input value to 25 MHz clock. The between block checks the input values is in the interval [M(=144), L(=783)] for H display.

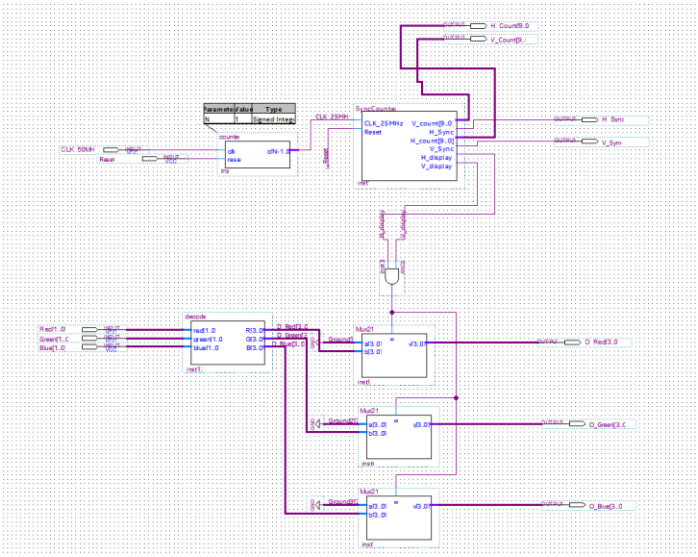
7) Synchro-Counter



<Figure 4: Design of Synchro-Counter on Quartus Software>

The output of Hsync will act as clock for the CounterNeg block. The output of CounterNeg is connected to two comparators. The first comparator outputs 1 when the input value is greater than 2 because of the feature of Vsync. The second comparator outputs 1 when the input value is greater than 524 to reset the CounterNeg block. Both synchronizer blocks synchronize the input value to 25MHz clock. The between block checks the input value is in the interval [M(=35), L(=514)] for V display.

8) VGA Interface(RGB controller)

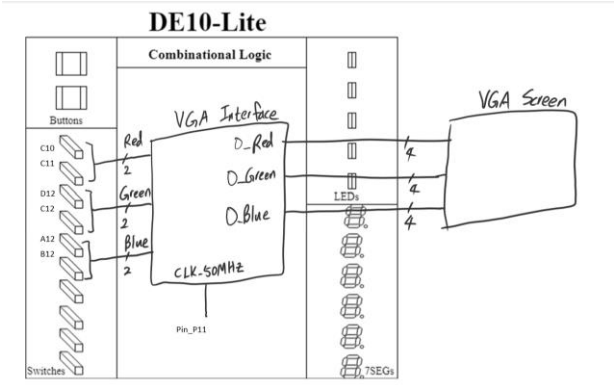


<Figure 5: Design of VGA interface on Quartus Software>

VGA interface or RGB controller controls the outputs Red, Green, and Blue on the VGA screen. First of all, by using 1-bit counter, 50MHz clock will be divided to 25MHz clock, and then 25MHz clock will be connected to Synchro-Counter. Then Synchro-counter will output Hsyn, Vsyn, H display, and V display. Each 2-bit input of RGB is passes decoder, and then the outputs will be 4-bit decoded values. Finally, Because of the feature of multiplexer, the intersection(or product) of V display and H display will determine the output of the multiplexer among two inputs of multiplexer: ground or the outputs of decoder.

9) Block Design and Pin Assignment

- Block Design



<Figure 6: Block Design of VGA interface>

The block design above shows how VGA interface works. The VGA interface gets each 2-bit input of RGB, and then prints out the output of RGB to VGA screen.

- Pin Assignment

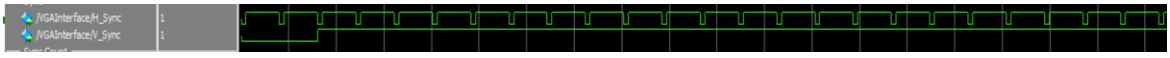
Interface Name	Interface Purpose	Input or Output	Number of Pins	Board Label	FPGA Pin
Red	Gate input	Input	2	Switch0,1	C10, C11
Green	Gate input	Input	2	Switch 2,3	D12, C12
Blue	Gate input	Input	2	Switch 4,5	A12, B12
CLK_50MHz	Gate input	Input	1	50MHz Clock	P11
O_Red	Gate output	Output	4	VGA_R [0~3]	AA1, V1, Y2, Y1
O_Green	Gate output	Output	4	VGA_G [0~3]	W1, T2, R2, R1
O_Blue	Gate output	Output	4	VGA_B [0~3]	P1, T1, P4, N2

3. Results

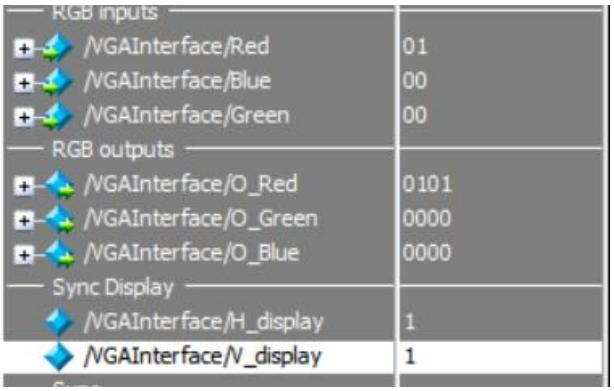
1) Results

RGB inputs	H display	V display	Expected RGB	Actual RGB
R[01],G[00],B[00]	0	1	R[0000] G[0000] B[0000]	R[0000] G[0000] B[0000]
R[00],G[01],B[00]	1	1	R[0000] G[0101] B[0000]	R[0000] G[0101] B[0000]
R[00],G[00],B[01]	1	0	R[0000] G[0000] B[0000]	R[0000] G[0000] B[0000]
R[10],G[00],B[10]	0	0	R[0000] G[0000] B[0000]	R[0000] G[0000] B[0000]
R[11],G[11],B[11]	1	1	R[1111] G[1111] B[1111]	R[1111] G[1111] B[1111]
R[11],G[11],B[11]	0	0	R[0000] G[0000] B[0000]	R[0000] G[0000] B[0000]
R[00]G[00]B[00]	1	1	R[0000] G[0000] B[0000]	R[0000] G[0000] B[0000]

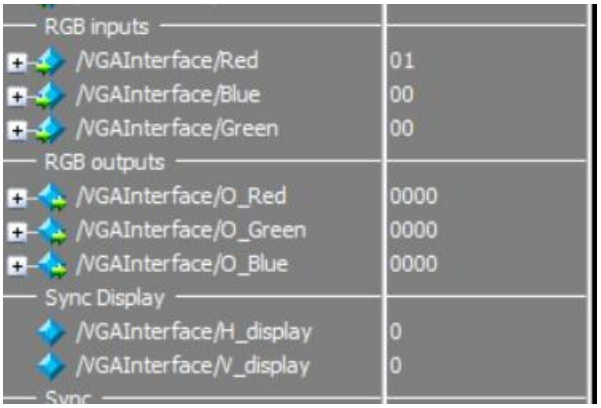
2) Modelsim Simulation



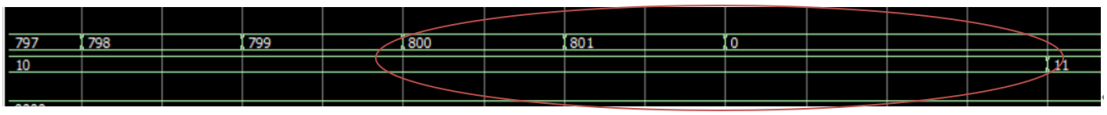
<Figure 7: H sync and V sync>



<Figure 8: RGB inputs and RGB outputs when H display and V display are 1.>



<Figure 9: RGB inputs and RGB outputs when H display and V display are 0>



<Figure 10: The counting of H sync and V sync>

Most of the parts are working as my expectation. However, there are some parts which does not match to my expectation which is marked in Figure 10. The circled part in Figure 10 works similar to the simulation of the clock in the previous lab section. Hence, I assume that this is caused by the delay of Vsync output. The reason is that Hsync signal passes 10-bit counter, and then its output is Vsync so that it will cause some delay, and this delay lead to this event.

Therefore, as considering for this reasoning, the lab simulation mostly worked as expectation, and is successful

4. Experiment Notes

In this lab section, there were some parts went well and some parts did not go well. First of all, I was struggling to design Vsync which repeats Hsync. I thought if Hsync is an integer value, then it can be implemented simple, but Hsync is not an integer value in the real world. So, I asked to professor about the parts which I stuck and my thoughts. Then professor gave me hints that the design of Vsync is similar to the design of clock. Thus, I can design the Vsync.

After I designed the Vsync, I assembled the parts I designed for VGA interface. Then I run the simulation of VGA interface, but there were some problems. The problems were that my RGB outputs are only indicating X values. So, I asked this problem to professor. Then professor gave me help to find the points caused the problems. They were caused by multiplexers which were not working on the simulation. Therefore, I redesigned my multiplexer, and then the simulation worked fine.

All of the parts went well after I solve the problems mentioned above.

5. Study Questions

1. How many pixels are displayed in a given frame of this VGA screen?

   In our design, there are 800 cycles horizontally, and 525 cycles vertically. Thus, there are 420000 pixels in VGA screen.

2. What is the frame rate of this video driver?

   The frame rate is the frequency in which a consecutive series of frames or images can appear on display panel [1]. The clock input is 25MHz, and the screen has 800 cycles horizontally and 525 cycles vertically. So, there are 420000 pixels. Then the frame rate is 25MHz/420000pixels. Therefore, the frame rate is 59.52Hz which is approximately 60Hz.

3. How many possible colors can be displayed by this video driver?

   The possible outputs of RGB are 0000, 0101, 1010, and 1111 for the video driver which we implemented in this lab section. So, there are 4 possible outputs for each Red, Green and Blue. Thus, the number of possible colors can be displayed by this video driver is 64, 4x4x4.

6. Appendix

A) Counter

```
module counter # (parameter N = 8)
                (input logic clk,
                 input logic reset,
                 output logic [N-1: 0] q);

 always_ff@(posedge clk, posedge reset)
   if(reset) q <= 0;
   else      q <= q+1;
endmodule
```

B) CounterNeg

```
module CounterNeg # (parameter N = 8)
                (input logic clk,
                 input logic reset,
                 output logic [N-1: 0] q);

 always_ff@(negedge clk, posedge reset)
   if(reset) q <= 0;
   else      q <= q+1;
endmodule
```

C) Comparator

```
module comparator # (parameter N = 8, M = 60)
                    (input logic [N-1 : 0] a,
                     output logic gt);

   assign gt = (a > M);

endmodule
```

D) Between

```
module between # (parameter N = 8, M = 60, L = 100)
                  (input logic [N-1 : 0] a,
                   output logic bt);

   assign bt = ((M <= a) & (a <= L));

endmodule
```

E) Synchronizer

```
module synchronizer(input logic clk,
                    input logic d,
                    output logic q);


    logic nl;

    always_ff@(posedge clk)
       begin
          nl <= d; //nonblocking
          q <= nl; //nonblocking
       end
 endmodule

```

F) Decoder

```
module decoder (input logic [1: 0] red, green, blue,
                output logic [3 : 0] R, G, B);

    logic o = 1;
    logic z = 0;

    assign R[3] = red[o];
    assign R[2] = red[z];
    assign R[1] = red[o];
    assign R[0] = red[z];

    assign G[3] = green[o];
    assign G[2] = green[z];
    assign G[1] = green[o];
    assign G[0] = green[z];

    assign B[3] = blue[o];
    assign B[2] = blue[z];
    assign B[1] = blue[o];
    assign B[0] = blue[z];

endmodule
```

G) Mux 2:1

```
module Mux21 (input logic [3 : 0] a, b,
              input logic s,
              output logic [3 : 0] y);


assign y = s ? b : a;

endmodule

```

<Citation>

[1] https://www.hp.com/us-en/shop/tech-takes/what-is-frame-rate