EE314 DIGITAL CIRCUITS LABORATORY

TERM PROJECT: BUFFER AND VGA DESIGN

Alper Arpaçay – 2303972 Syed Muhammad Farrukh Aijaz- 2417152

ABSTRACT

In this project, we are required to design a VGA screen using Verilog HDL with all stages required to follow certain performance specifications given to us regarding the overall Buffer latency and reliability and the reference VGA display. The design involves a VGA display comprising of four Buffers as well as the relevant packet information which read the serial data according to input, either latency or reliability are taken as a reference while reading occurs at a specified frequency.

Index Terms— Buffers, MSB, LSB, FPG

I. INTRODUCTON

Video Graphics Array (**VGA**) is a video display controller and accompanying de facto graphics standard, first introduced with the IBM PS/2 line of computers in 1987, which became ubiquitous in the PC industry within three years. The term can now refer to the computer display standard, the 15-pin D-sub miniature VGA connector, or the 640×480 resolution characteristic of the VGA hardware [1].

As for the design of the entire system comprising of the VGA and the Buffers, a FPGA is required along with a monitor having a display of 640 x 480 @ 60 Hz. The clocked frequency of the FPGA board is 25M Hz. Hence a frequency divider was used to map to a proper frequency for the board and for the display.

II. TOP-LEVEL DESIGN

i. State Diagram Representation of the Module

The overall top-level state-diagram design can be seen below in figure 1.

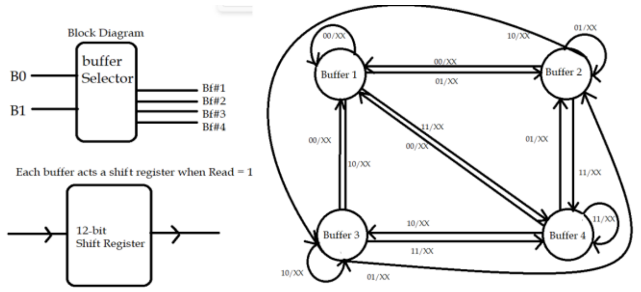


Figure 1: Overall state diagram for planned system.

We begin our design discussion with the reference block and states. Each state in the diagram can have $2^n - 1$ possible state transitions as indicated in the diagram. In our case we have 4 states hence 7 different transitions can occur for reach Buffer depending on the 4-bit serial data. A possible transition can occur for e.g., Serial data of 0011 is written, the MSB's here are 00 which indicate the decimal equivalent of 0 corresponding to Buffer 1 hence Buffer 1 is selected as the starting state. The two LSB's are 11 which corresponds to the decimal equivalent of 3 are stored as a packet inside the corresponding Buffer. This process is repeated until no more data is sent and no more state transitions occur.

ii. Writing the Data

Since data is manually entered through either switch in the FPGA or the buttons, the nature of the data is considered to be serial data [2]. In order to ensure that the serial data is fetched accurately a serial to parallel data conversion is used, the functional simulation for the serial to parallel conversion is given in fig.2 in simulation results. By grouping the serial data into 4 bits of parallel data and extracting the two most significant bits we can identify the type of Buffer to be selected and the data to be written in the Buffer suing the two least significant bits.

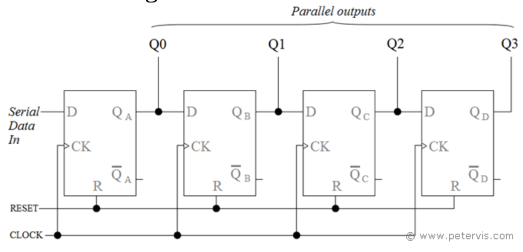


Figure 2: Serial to Parallel Data

Also, there is a pack counter that counts the number of packets present at any time inside each of the buffer, as

well as a count that counts the number of packets received in total in that relevant buffer.
As soon as the pack count is greater than 6 implying the number of packets inside the buffer exceeds 6 the pack count is decremented by 1. Simultaneously another counter named as drop is increased by 1 in order to record the number of drops in each buffer..

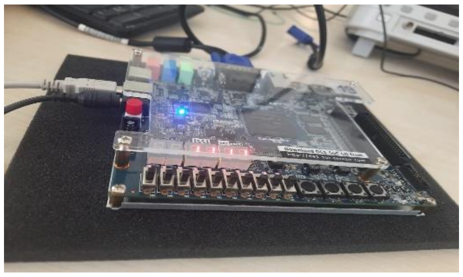


Figure 3: FPGA board with Data Input

iii. 1. Reading the Data

After the writing part is completed, we are tasked to design a reader operating at a frequency of $\frac{1}{3}\ Hz$ implying that after every 3 seconds data is to be read from the relevant Buffer depending on the latency or the reliability whichever satisfies the condition algorithms that are coded into the FPGA board. However, the task of operation was the frequency adjustment since the FPGA board operates at a clocked frequency of 25M Hz, a frequency divider was used to calibrate the reading frequency to 1/3 Hz by first reducing the frequency to from 25M Hz to 1 Hz and then adjusting it to our needs in the VGA display and reading part of the system. The simulation results are shown in the end.

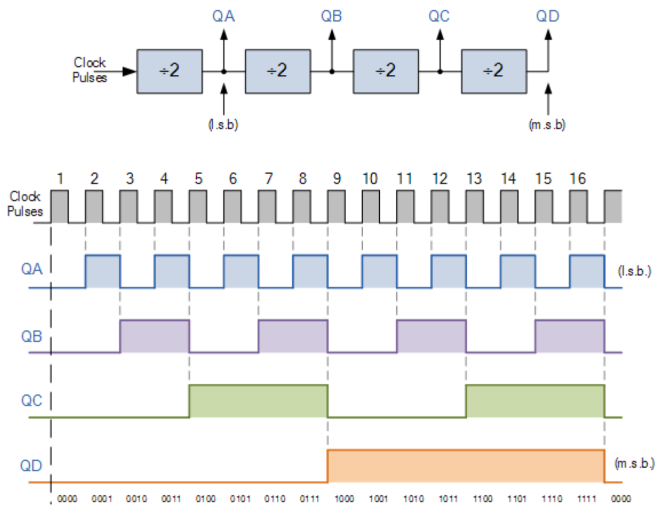


Figure 4: Frequency Divider Analogy

iii. 2. Optimizing the Read Data (Latency vs. Reliability)

An exquisite task was to optimize the reading part of the system where two important parameters were to be considered Latency and Reliability. These two factors played quite a roll each having their own preference order for the Buffers as mentioned below:

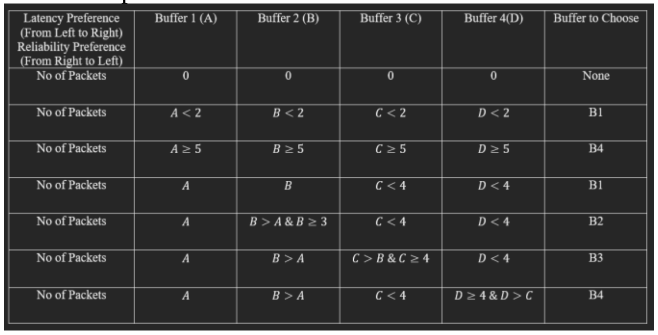
For Latency

$$Buffer\ 1 > Buffer\ 2 > Buffer\ 3 > Buffer\ 4$$

For Reliability

$$Buffer\ 4 > Buffer\ 3 > Buffer\ 2 > Buffer\ 1$$

Hence an algorithm had to be made for prioritizing either of the two depending on the number of packets in each Buffer at any present moment. A table is provided which governs the underlying concept of the algorithm including the boundary conditions for the packets and the priority of which Buffer to be read. The variables A, B, C and D here represents the number of packets in each Buffer.



Latency Preference (From Left to Right) Reliability Preference (From Right to Left)	Buffer 1 (A)	Buffer 2 (B)	Buffer 3 (C)	Buffer 4(D)	Buffer to Choose
No of Packets	0	0	0	0	None
No of Packets	$A < 2$	$B < 2$	$C < 2$	$D < 2$	B1
No of Packets	$A \geq 5$	$B \geq 5$	$C \geq 5$	$D \geq 5$	B4
No of Packets	A	B	$C < 4$	$D < 4$	B1
No of Packets	A	$B > A \,\&\, B \geq 3$	$C < 4$	$D < 4$	B2
No of Packets	A	$B > A$	$C > B \,\&\, C \geq 4$	$D < 4$	B3
No of Packets	A	$B > A$	$C < 4$	$D \geq 4 \,\&\, D > C$	B4

Figure 5: Buffer Selection Algorithm Analogy

An example condition can be explained as follows: Assume we have 3 packets in Buffer 1 and 2 packets in Buffers 2, 3, 4 respectively. Then the Buffer to be chosen for reading to should be Buffer 1 according to the 5th row in the table above. Relevant examples can be thought of for selecting each buffer depending on the number of packets in each Buffer.

iv. VGA Design

VGA stands for Video graphics array and it's one of the most diffuse standard for video transmission; it roots its definition from the way old catodic tubes work: the image is constructed one lines at times, starting from the top and each line is displayed from the left to the right (by the way this is the reason for the Y axes orientation in graphics programming.
Now we can talk about its implementation with a hardware description language: we were tasked to use Verilog HDL;

In the following code, using the pixel clock, we increment the counters connected to the horizontal and vertical signals; when they reach their maximum values they are reset.

No pixel data is created in this module, here we are interested only to the sync signals generation. To generate a RGB signal we need to couple this module with one that derive the pixel colour values from the counters: in the following module we use the last four most significant bits of the x coordinate so to obtain a change of colour every 8x10 pixels for each of the buffers whenever a data packet is written into the buffer.

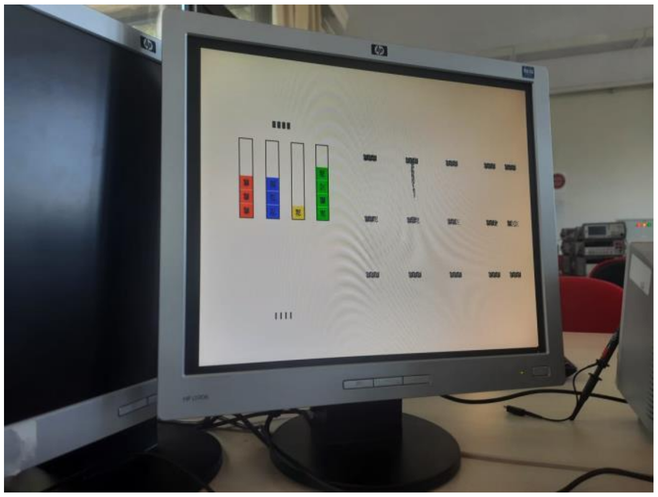


Figure 6: VGA design

III. SIMULATION RESULTS

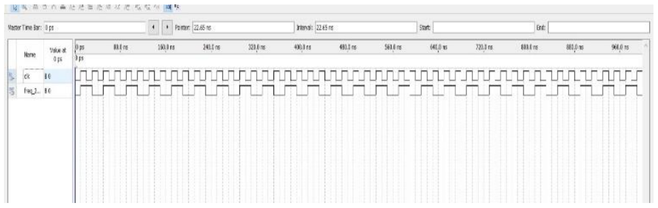


Figure 7: Frequency Clock generator for 25M Hz

A clock signal is generated at the start of writing the data into the FPGA board, however for the frequency of operation clocked at 25M Hz the above simulation for the code was used to generate the adequate frequency.

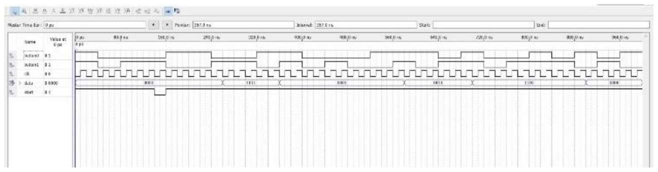
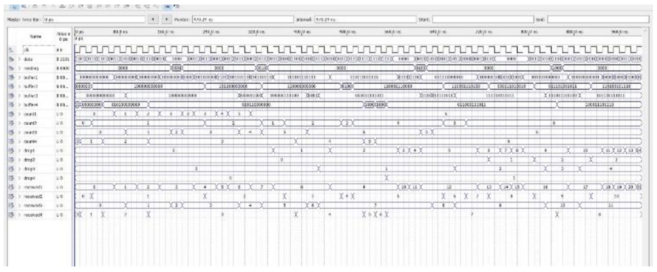


Figure 8: Serial to Parallel Data Conversion

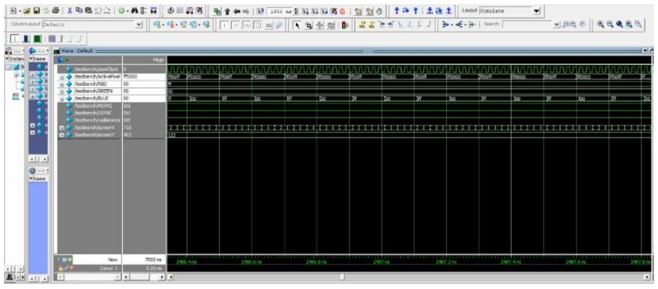
As bits were sent serially arbitrarily, they were grouped in a 4 bit binary number hence a serial to parallel data converted was needed as shown in the above simulation.



Data was read as explained above and the number of transmitted, received and drop packets were calculated and shown in the functional simulation above.

After data was read as shown in the previous image reading was carried out at a time period of 3 sec which can be observed from the functional simulation above.

Lastly since the VGA was not working as quite as expected the RTL simulation was done in order to test the code for the VGA design and see the Hexadecimal value of the RGB colours [3]



IV. CONCLUSION

To sum it all up, one can say we were tasked to design an HUD display layout which is quite a popular system used in gaming purposes mostly or widely spread in first person shooter games which require the latency and the packets dropped in order to determine the effective speed of the game itself while connected to the servers for the game via internet.

V. REFERENCES

[1] Petzold, Charles (July 1987). "Triple standard: three new video modes from IBM". PC Magazine. Ziff Davis. Retrieved 2020-04-13.

[2] https://www.versitron.com/blog/a-comparison-of-parallel-data-transmission-and-serial-data-transmission#:~:text=Serial%20Transmission%3A%20In%20this%20type,placed%20parallel%20to%20each%20other.

[3] https://www.rapidtables.com/web/color/RGB_Color.html