This report is prepared for the final project of the EE240 Digital Design Course.

**FPGA BASED DIGITAL CLOCK, CHRONOMETER AND TIMER**

**Eyüpcan Yılmaz**

**Emre Çifçi**

**Project Advisor: Prof. Şenol Mutlu**

**01.07.2021**

TABLE OF CONTENTS

**1)INTRODUCTION2**

**2)PROBLEM STATEMENT2**

**3)RELATED BACKGROUND2**

3.1)Video Graphics Array(VGA)2

3.2) Seven Segment Display3

**4)DESIGN4**

4.1)Whole Design with Inputs and Outputs4

4.2) Definitions of Components5

4.3) Connections of Components6

4.3.1) Clock Generator Logic6

4.3.2) VGA Driver of the Circuit6

**5)RESULTS8**

**6)CONCLUSION8**

**7)REFERENCES8**

APPENDIX9

# **1-Introduction**

Field Gate Programmable Arrays (FPGAs), nowadays, are important alternatives to microcontrollers. They are used for describing hardware, instead of programming a microcontroller. Because they construct hardware and do not have a sequential order, they provide more speed operations. Therefore, they are widely used in DSP applications like video and sound processing and they have a wide usage area in military systems.

# **2-PROBLEM STATEMENT**

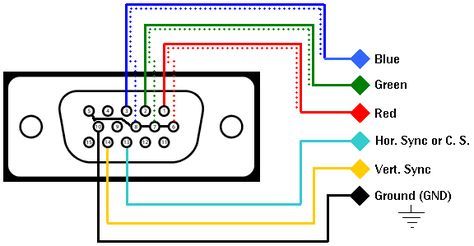
In this project, our main purpose was to obtain a digital clock with three functions using VHDL code and an FPGA board on the VGA driver. These three functions are classical digital clock (clock), chronometer(stopwatch), and timer. All these three modes can be reset, set, stopped, continued, and for the digital clock and the timer, hour digits and minute digits can be changed. Additionally, the color of digits can be changed using the color change button. To obtain such a multi-functional circuit, lots of components and signals were needed. More of them will be mentioned in detail in the design part of this project report.

# **3- RELATED BACKGROUND**

3.1 – Video Graphics Array (VGA)

Video Graphics Array (VGA) refers specifically to the display hardware that is originated with the 1987 IBM PS/2 and its VGA graphics systems, the 15-pin connector has become more common on PCs, as well as many monitors, projectors, and televisions, and so on.

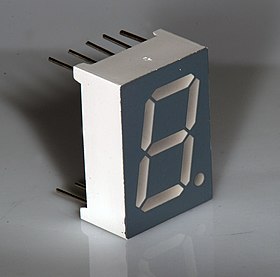
VGA provides 640x480 resolution color display screens with a refresh rate of 60Hz and 16 colors displayed at a time. VGA utilizes analog signals. This results in lower resolutions and lower quality displays on screens. Although it is the most common system, it has also disadvantages.

All VGA connectors carry analog red, green, blue color signals and horizontal and vertical synchronization signals.

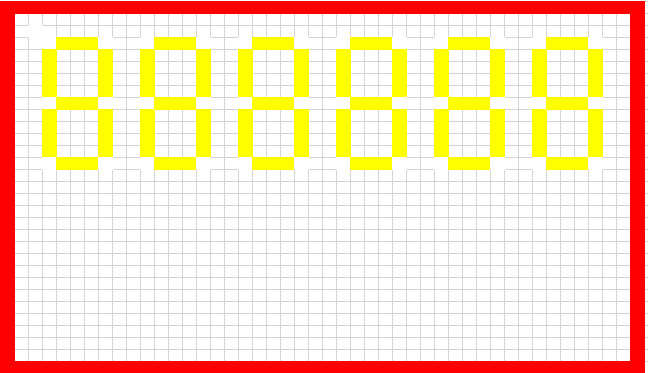
*Figure 1. VGA Connector Pinout*

3.2 Seven Segment Displaying

Seven segment display is a separate electronic component that is used frequently on digital circuits such as calculators, clocks, and so on. It has seven segments whose pins are separate and by using these pins, users can display decimal numbers on it. In this project, we imitated the seven-segment display method to show numbers on the monitor.

 In figure 3, pixel blocks are showed. Each square is set as 16 pixels and the total area occupied by digits is the up-half of the screen. There are two components to accomplish this displaying in the circuit. First is the segment creator. Segment creators match the binary data input with the corresponding segments which need to be turned on. The second component is the VGA driver, which creates these segments on the monitor by using some algorithm. These are mentioned in detail in the design part.

*Figure 2. A typical seven-segment display*

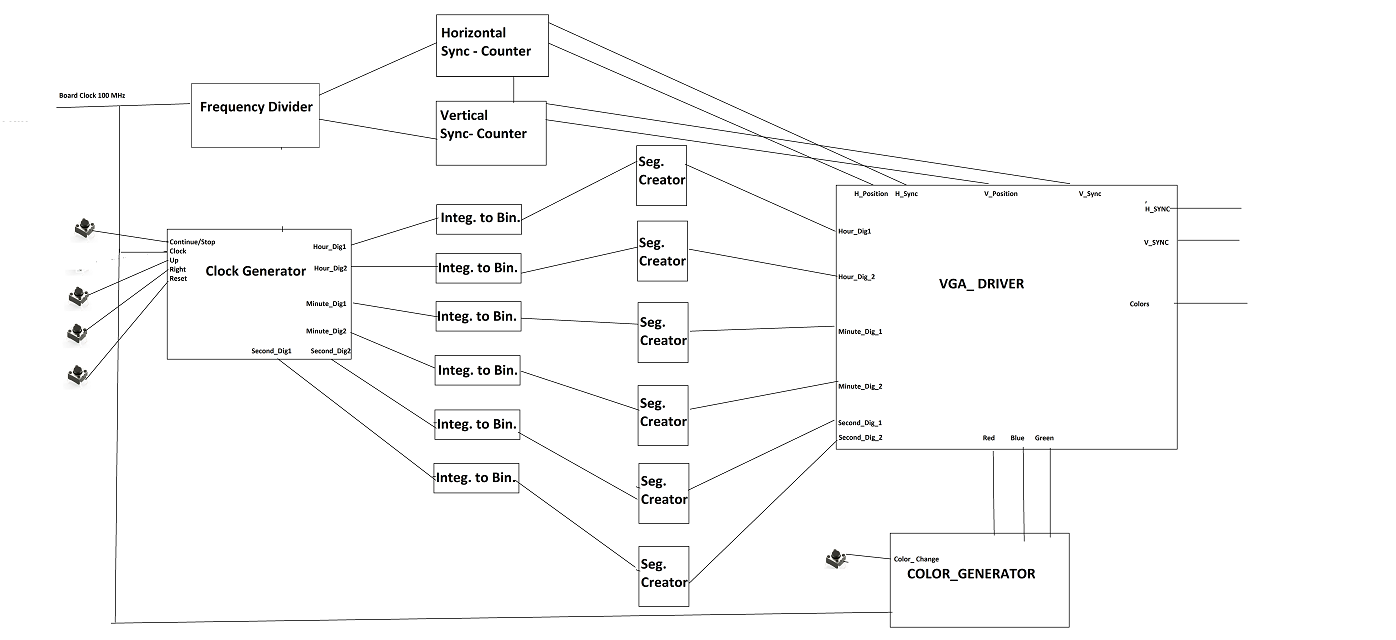


*Figure 3. Excel Simulation of Location of the Digits in the Screen*

# **4-DESIGN**

4.1) Whole Design with All Inputs and Outputs

During the design, separate entity creating is used to simplify the debugging. This allows us to test every part by using simple test benches. The whole circuit configuration is given above. (Fig. 4)



*Fig.4 Whole Circuit Design*

In this design we have the following inputs and outputs:

Inputs: clock, reset\_all (which resets horizontal and vertical synchronization), color\_change (which changes the color of digits of the clock), mode\_change (which initially starts from 00-clock then 01-stopwatch and 10-timer and goes on as we push this button), continue\_stop (which enables us to stop or continue the counting or the countdown), reset (which makes the code go back to mode 00-clock mode and makes all corresponding digits equal to 0), right (which helps us to choose which digit we want to increment-this is only applicable for an hour and minutes’ digits) and up (which increments the value of the corresponding digit with some condition (just like you should not increment the second digit of the hour 23 and make it 24 since it is something incompatible.)) and some IO expansions.

Outputs: HSYNC, VSYNC (horizontal and vertical synchronization signals for 25MHz which will be connected to the VGA monitor and colors (which is coherent with the synchronization signals).

In this part, all components will be explained one by one and then the main circuit will be presented.

4.2 Definition of Components:

* IO Expansion Component: This component is not included in the original circuit, therefore it cannot be shown on the circuit diagram. It makes it possible to control FPGA online because of the pandemic.
* clock generator: This is the component that holds the main logic behind this VHDL design. Clock generator takes clock, reset, mode\_change, continue\_stop, right, up as inputs and take out 6 digits as integers. (hour digit 1, hour digit 2, minute digit 1, minute digit 2, second digit 1, and second digit 2). To do this, first, it detects if any button is pushed or not. It holds mentioned inputs’ old values in different d flip flops and checks them for each clock cycle in its “change” process. Then, if needed, it uses the go\_right process to go right as the button is pushed holding the current position information. But, since it does not so much make sense to go second digits and change them, only hours and minutes can be changed in this design. After that (it is not after since it is a behavioral code and they all are processes), the main\_logic process is begun. In this process, according to the mode (clock, stopwatch, or timer), if continue\_stop is off (using control signal in the code) it makes us allow to change desired digits using right and up buttons, else it counts up or counts down starting from seconds just like in any other clock. Then, at the end of this part, it chooses corresponding signals for the output digits according to the mode.
* frequency\_divider: This is the component that gives us a 25MHz clock signal from the clock input signal which is 100MHz.
* horizontal\_sync\_counter: This is the component that gives coherent horizontal position and synchronization signal using clock\_divided which is the output of the frequency\_divider component.
* vertical\_sync\_counter: This is the component that gives coherent vertical position and synchronization signal using clock\_divided signal and horizontal position signal which is the one that vertical\_sync\_counter is dependent to.
* integertobinary: This is the component that enables us to convert integer values up to 9 to binary numbers with four digits. This is one of the important components since we need each digits’ binary representation in the VGA displayer just like in the 7-segment converter.
* color\_generator: This is the component that enables us to change color arbitrarily.
* segment\_creator: This is the component that converts binary representation of the digits to their 7-segment representation.
* vga\_displayer: This the component which takes out coherent color vectors to the synchronization signals and the desired locations in Fig3. Our VGA monitor is 480\*640 or (16\*30)\*(16\*40). Then if we divide the upper part of the monitor by 16\*16 squares and find out the distances between digits at which we want to locate our clock digits, we can easily display the desired clock.

4.3 Connections of the Components:

frequency\_divider takes clock and takes out clock\_divided, this clock\_divided is taken by vertical and horizontal synchronization counters. The horizontal counter’s output horizontal position is connected to both the VGA displayer and the vertical counter. Clock generator takes clock and all previously mentioned inputs and comes out with 6 integer values. For each one, the first integertobinary component is connected then segment\_creator is connected. Outputs of the segment creators are connected to the VGA displayer component and the resulting output is colors and synchronization signals.

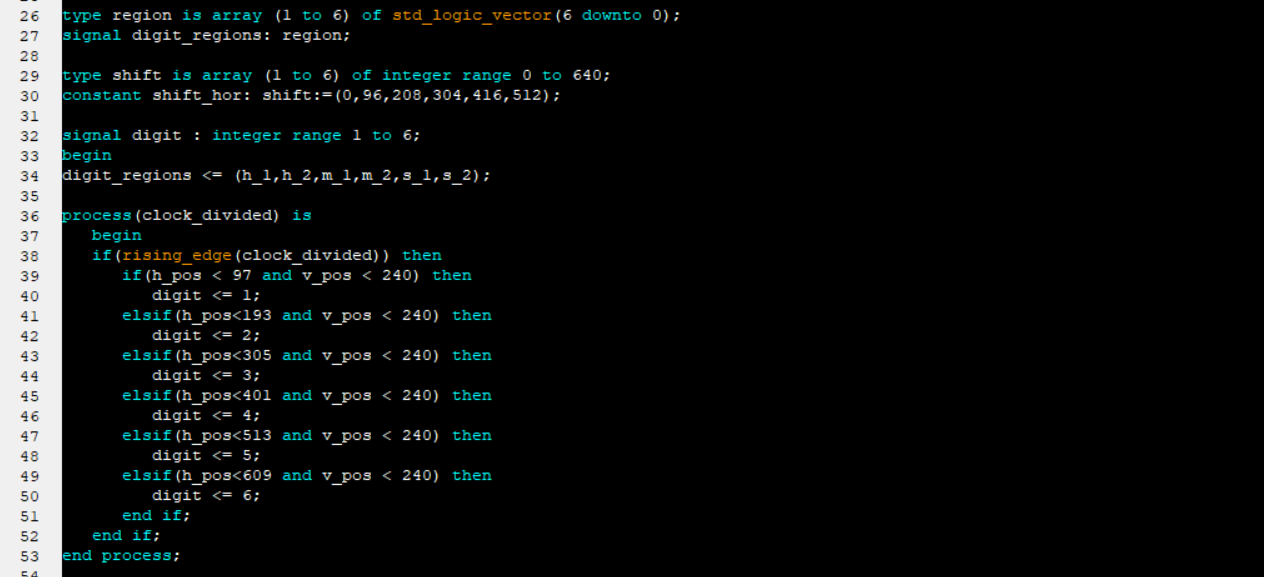
4.3.1 Clock Generator Logic

In this part, there are 3 modes: “00”, “01”, “10”. “00” corresponds to clock mode, “01” corresponds to stopwatch mode and “10” corresponds to timer mode. You can change the mode by using the mode\_change button on the FPGA board. In clock and timer mode, you should set the initial values for hour, minute, and so on. You can accomplish this by using the up and right buttons on the board. The right button is changing the selected digit of the system to change. You can increase the selected bit by using the up button. If you reach the rightmost digit while using the right button you turn the leftmost digit if you push the right button again. If you reach the highest possible value for a digit such as 9 for ones digit of the minute, you turn into 0 if you push up again. Therefore, we can accomplish the functions of the clock without using more buttons. If you power on the timer or clock after setting initials, you should use the continue\_use button on the board. It decides when is setting and when is processing. You can use this button also to stop the stopwatch. If you want to reset the whole system, you can use the reset button on the board. All modes are set according to logic rules. For example, you cannot get an hour value bigger than 20 or a minute value bigger than 59. In sum, it is considered to be a fully-functional clock/timer/stopwatch circuit.

4.3.2 VGA Driver of The Circuit:

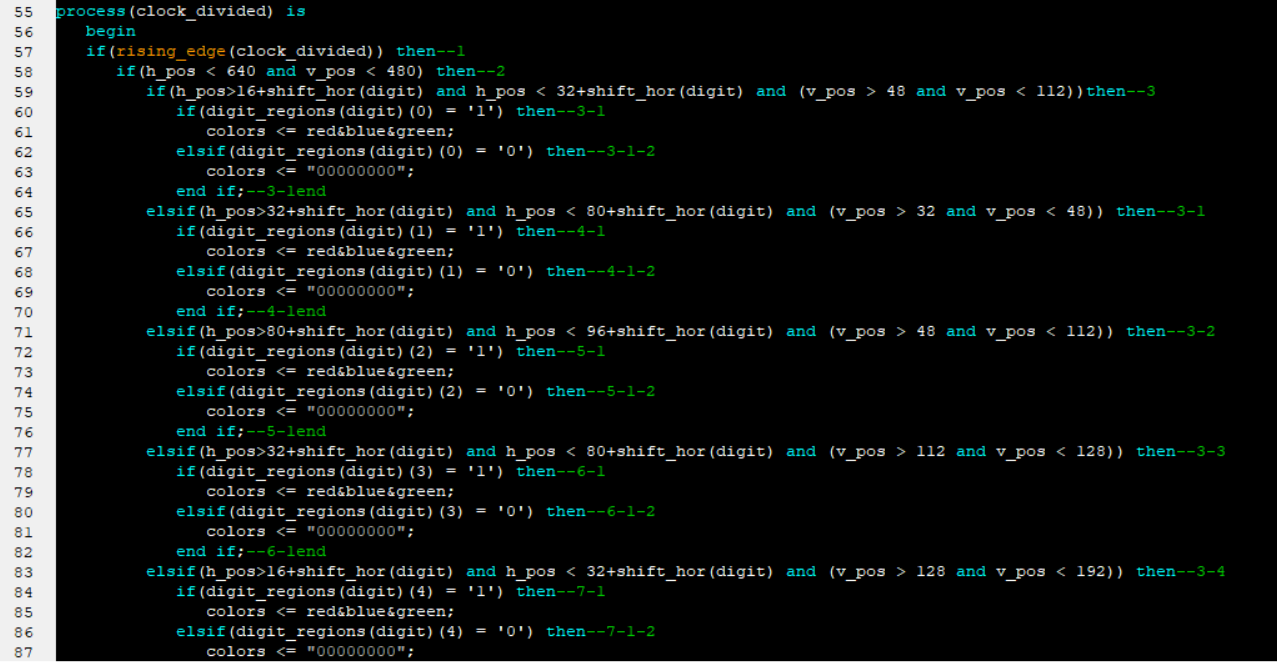
In this circuit, a VGA monitor is used to show the result of the system. Therefore, it is mandatory to create a VGA driver for this circuit. It is accomplished by dividing the screen area into regions to create a 6-digit 7-segment system. This block takes 7-bit region info about 6 digits from segment creators and an 8-bit color signal(red, blue, green) for each clock. Color generator is a basic state machine that provides different color combinations when color\_change button is pushed.

This block is classic VGA driver code written in VHDL that applies R, G, B values when the desired pixel block is reached. It uses VGA timing rules,25 MHz clock in the process. To display digits correctly, information that comes from segment creators written into an array called digit\_regions, and the corresponding shifting values per digit is written into an array shift\_horizontal.



*Figure 5. Matching Digit Numbers with Horizontal and Vertical Positions*

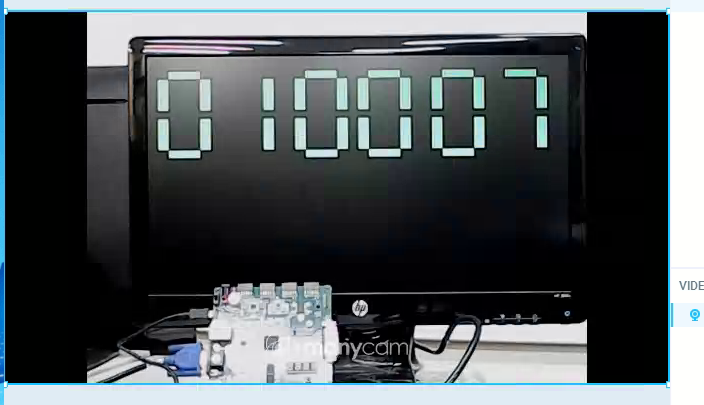
By using figure 3, it can be decided which digit the horizontal and vertical positions belong to (this procedure can be seen in Figure 5).



*Figure 6. Checking the Segments Locations and Whether These Segments Should be Active*

As seen in figure 6, by using shift\_hor array, we try to decide which segment the positions correspond to and whether these segments should be activated or not by using the digit value and digit regions that is array stores the information about the segments. If this checking is true, we determine if the region information of that digit (which is 1 in this case) is '1', then we send the color information of the digit that is obtained from the color generator component to R, G, B signals; if not, however, we send '0' for R, G, B signals, indicating that region to be black colored.

# **5-RESULTS**

As can be seen in the Figure 7 that, main target is achieved,

*Figure 7. Final Result*

# **6-CONCLUSION**

Our main goal is to achieve a digital clock implementation. The majority of our purpose was realized. In this report, the digital clock is capable of displaying hours, minutes, and seconds with timer and stopwatch functions. Because of using more than one clock signal, there are some problematic issues We faced some synchronization problems and we tried to overcome them. We could overcome some of them and we could not the rest. We try to overcome these problems in the future. In the future, this system would be expanded with additional features like an alarm clock, storage of laps in stopwatch, and date month year calendar.

# **7.References**

<https://www.xilinx.com/>

<https://wikipedia.org>

Wayne Wolf, “FPGA – Based System Design ”, Pearson Education, 2005

**APPENDIX**

Read Me

Prepare your FPGA board, use buttons

15<-reset, 14<-reset\_all, 13<-up, 12<-right, 11<-continue\_stop, 10<-mode\_change, 9<-color\_change

Users’ Manual

First of all, push 9 to change color and see the template of the circuit with all digits equal to zero. After that, you can change its color using the same button. Use each button twice to avoid any error that might come from initial conditions. (This is recommended but there might be no problem if you do not use all of them twice.) Then you are ready to use this multifunctional clock. Please be patient and wait at least >=1 second for your next button command.