

FPGA Based Digital Logic Circuits Operation for Beginners

San San Naing, Ni Ni San Hlaing, Cho Thet Nwe

Department of Electronic Engineering, Technological University, Kyaukse, Myanmar

How to cite this paper: San San Naing | Ni Ni San Hlaing | Cho Thet Nwe "FPGA Based Digital Logic Circuits Operation for Beginners"

Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-3 | Issue-5, August 2019, pp.495-501, <https://doi.org/10.31142/ijtsrd26372>



ABSTRACT

This paper presents the operations of digital circuits based on FPGA. The long term of FPGA is field programmable gate array. FPGA is an integrated circuit designed to be configured by a customer or a designer after manufacturing – hence "field-programmable". The operations of logic circuits such as logic gates, flip-flop and 7 segment are tested using quartus II software and DE2-115 and DE1 FPGA development kits in this paper. Particularly, there are three main portions such as implementation of schematic diagram, designing of the vhdl program, the connection of the control panel and displaying the result of logic circuits on FPGA kit. The operations of combinational circuits are tested by designing the VHDL programs. And then the operations of sequential circuits are observed and displayed the results of them by illustrating the schematic diagrams.

Copyright © 2019 by author(s) and International Journal of Trend in Scientific Research and Development Journal. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0) (<http://creativecommons.org/licenses/by/4.0>)



KEYWORDS: FPGA, VHDL, Combinational Circuits, Sequential Circuits, Quartus II, Schematic Diagram

I. INTRODUCTION

Logic circuits are implemented electronically, using transistors on an integrated circuit chip. Commonly available chips that use modern technology may contain hundreds of millions of transistors, as in the case of computer processors. The basic building blocks for such circuits are easy to understand, but there is nothing simple about a circuit that contains hundreds of millions of transistors [2].

II. LITERATURE OF DIGITAL CIRCUITS

Nowadays, digital technology becomes popular and very useful application more and more. Digital electronics, digital technology or digital (electronic) circuits are electronics that operate on digital signals. In contrast, analog circuits manipulate analog signals whose performance is more subject to manufacturing tolerance, signal attenuation and noise. Digital techniques are helpful because it is a lot easier to get an electronic device to switch into one of a number of known states than to accurately reproduce a continuous range of values. Digital electronic circuits are usually made from large assemblies of logic gates (often printed on integrated circuits (IC)), simple electronic representations of Boolean logic functions[4].

Logic circuits are used to build computer hardware, as well as many other types of products. All such products are broadly classified as digital hardware. The digital circuits such as combinational circuits, sequential circuits are utilized to implement those digital hardware devices [2]. Therefore this paper presents the operations of those circuits by using different designing ways.

A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a customer or a designer after manufacturing – hence "field-programmable". The FPGA configuration is generally specified using a hardware description language (HDL), similar to that used for an application-specific integrated circuit (ASIC) (circuit diagrams were previously used to specify the configuration, as they were for ASICs, but this is increasingly rare).

FPGAs contain an array of programmable logic blocks, and a hierarchy of reconfigurable interconnects that allow the blocks to be "wired together", like many logic gates that can be inter-wired in different configurations. Logic blocks can be configured to perform complex combinational functions, or merely simple logic gates like AND and XOR. In most FPGAs, logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory [3]. FPGA can be used in digital signal processing, bioinformatics, device controllers, software-defined radio, random logic, ASIC prototyping, medical imaging, computer hardware emulation, integrating multiple SPLDs, voice recognition, cryptography, filtering and communication encoding and many more [1].

VHDL

Gate	Input		Output
NOT	0		1
	1		0
AND	0	0	0
	0	1	0
	1	0	0
	1	1	1
NAND	0	0	1
	0	1	1
	1	0	1
	1	1	0
OR	0	0	0
	0	1	1
	1	0	1
	1	1	1
NOR	0	0	1
	0	1	0
	1	0	0
	1	1	0
XOR	0	0	0
	0	1	1
	1	0	1
	1	1	0
XNOR	0	0	1
	0	1	0
	1	0	0
	1	1	1

A logic circuit is a circuit that executes a processing or controlling function in a computer. This circuit implements logical operations on information to process it. Logic circuits utilize two values for a physical quantity, like voltage, to denote the Boolean values true and false or 1 and 0 respectively. There are two main types of digital logic circuits. They are combinational logic circuits and sequential logic circuits. Combinational circuitry performs like a simple function. The output is based on the present values of the input. Combinational circuitry is theoretically built from basic logic gates: AND gates, OR gates, XOR gates and inverters. The outputs of gates in combinational circuitry are never sent back directly to earlier inputs[5]. Combinational circuits are also time independent. Along with the absence of concepts like past inputs, combinational circuits also do not require any clocks. Sequential circuits are a collection of memory elements. These memory elements are flip-flops. These circuits are capable of remembering data. Therefore, a sequential circuit's output depends on current input, as well as past input. Moreover, since flip-flops are present the output of a sequential circuit also depends on the clock input.

The operations of the logic gates can be seen in the table.

III. METHODOLOGY

The operations of logic gates, combinational circuits and sequential circuits are tested by designing the VHDL program and analyzed the results on FPGA development kits (DE1, DE2). A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a customer or a designer after manufacturing- hence the term 'field-programmable'. It is a type of device that is widely used in electronic circuits. FPGAs are semiconductor devices which contain programmable logic blocks and interconnection circuits. It can be programmed or reprogrammed to the required functionality after manufacturing. The specific application of an FPGA includes digital signal processing, bioinformatics, device controllers, software-defined radio, random logic, ASIC prototyping, medical imaging, computer hardware emulation, integrating multiple SPLDs, voice recognition, cryptography, filtering, and communication encoding and many more.

3.1. FLOW CHART OF THE DIGITAL LOGIC CIRCUIT DESIGN

There are many steps in designing the digital logic circuits. Firstly, the designer must consider the design concept of the desired circuit and partitions of the circuit. And then, the interconnection of the partition blocks must be considered by the designer. Then, the designer can make the simulation in order to test the function of the complete circuit. Only if the function of the circuit is correct, the physical mapping can be done. After completing this step, the timing simulation must be performed and the designer must check the timing condition is correct or not. Only when the timing of the circuit is correct, the hardware implementation can be made. The flow chart of the digital logic circuit design is illustrated in Figure 1. This paper describes that the functions of the digital logic circuits are simulated and checks the simulation results which are correct or not.

3.2. VHDL CODE OF THE DIGITAL CIRCUITS

The operations of logic gates, combinational circuits and sequential circuits are tested with quartus II software and FPGA development kit. The operations of logic gate and combinational circuit are simulated by designing the VHDL programs. The sequential circuit is tested by designing the schematic diagram on the quartus II software and check the operation of the circuit on FPGA development kit. The program code of logic circuit can be written with different levels such as gate level, dataflow level, and behavioral level. The VHDL code for AND logic gate can be written as:

```

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY And_Gate IS
PORT( A : IN STD_LOGIC;
      B : IN STD_LOGIC;
      Z : OUT STD_LOGIC);
END ENTITY And_Gate;
ARCHITECTURE Dataflow OF And_Gate IS
BEGIN
Z <= A AND B;
END ARCHITECTURE Dataflow;

```

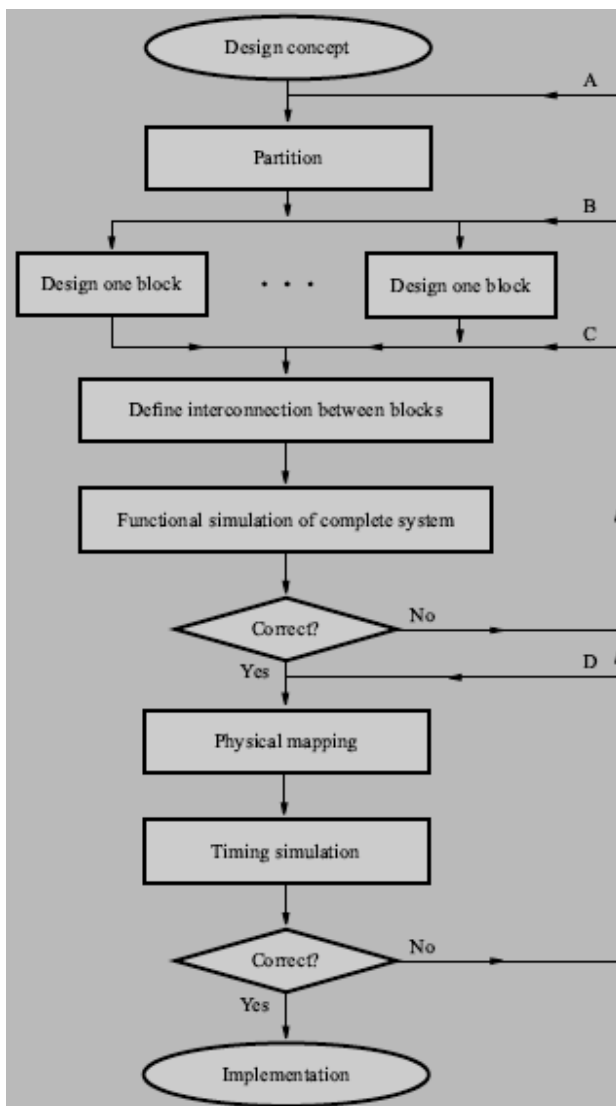


Fig1. Design Flow Chart of the Digital Logic Circuit

The program of other logic gates can be written in this form. The 7segment can be programmed in two ways in order to display the output not only the LED outputs but also the 7segment display output. The program code for 7segment for describing the output as 7segment display out is designed as follow:

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity bcd_7seg is
Port (B0,B1,B2,B3: in STD_LOGIC;
A,B,C,D,E,F,G : out STD_LOGIC);
end bcd_7seg;
architecture Behavioral of bcd_7seg is
begin
A <= B0 OR B2 OR (B1 AND B3) OR (NOT B1 AND NOT B3);
B <= (NOT B1) OR (NOT B2 AND NOT B3) OR (B2 AND B3);
C <= B1 OR NOT B2 OR B3;
D <= (NOT B1 AND NOT B3) OR (B2 AND NOT B3) OR (B1
AND NOT B2 AND B3)
OR (NOT B1 AND B2) OR B0;
E <= (NOT B1 AND NOT B3) OR (B2 AND NOT B3);
F <= B0 OR (NOT B2 AND NOT B3) OR (B1 AND NOT B2) OR
(B1 AND NOT B3);
G <= B0 OR (B1 AND NOT B2) OR ( NOT B1 AND B2) OR (B2
AND NOT B3);
end Behavioral;
    
```

IV. RESULTS OF THE PERFORMANCE OF Digital Circuits USING QUARTUS II AND FPGA Kits

This paper presents the operations of digital circuits as the 7segment display results and LED results. Moreover, these results are analyzed with the timing diagrams. Firstly, the operation of logic gate want to be presented with the LED output of the FPGA development kid.

Secondly, the BCD_to_7Segment combinational circuit is tested by using DE1 development and educational board. There are four inputs which are represented as binary value '0000' to '1001' and 10 outputs which are represented as decimal value 0 to 9. Four inputs are applied by SW3, SW2, SW1, SW0 of DE1. Decimal value 10 outputs are displayed on 7Segment of DE1.

The results of AND logic gate are described as follow:

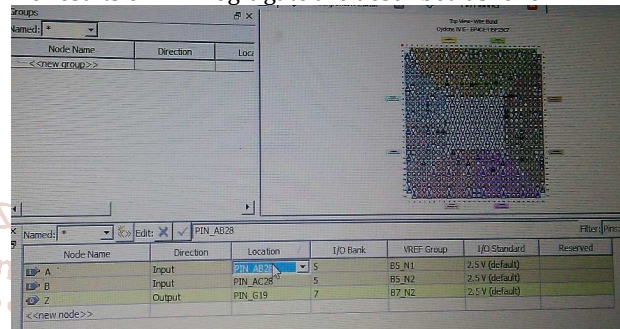


Fig2. Pin Assignment for Inputs (A,B) and Output (z)

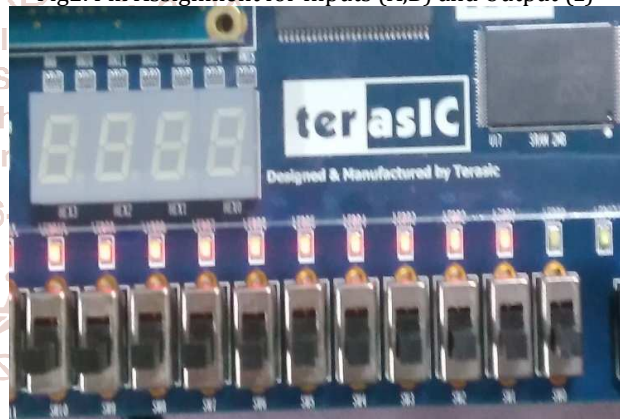


Fig3. Output LED is OFF when Inputs are '00' condition

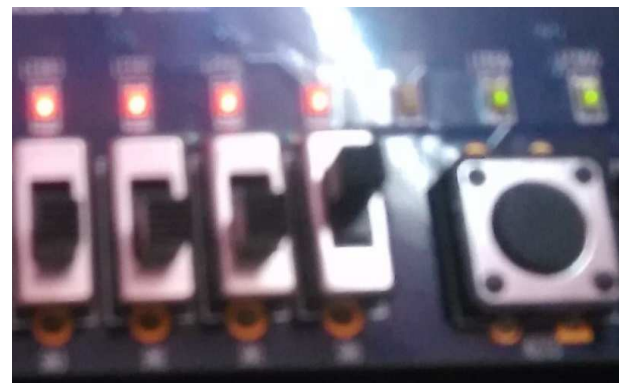


Fig4. Output LED is OFF when Inputs are '01' and '10' condition



Fig5. Output LED is OFF when Inputs are '11' condition

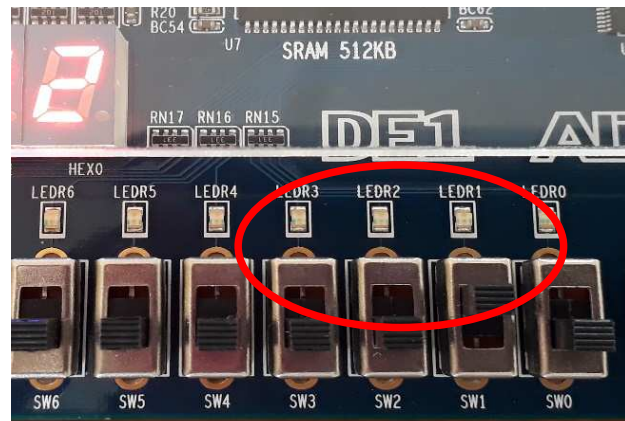


Fig9. Decimal Value 2 on 7Segment for Binary value '0010'

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard
BCDn[3]	Input	PIN_V12	7	B7_N1	3.3-VLV...default
BCDn[2]	Input	PIN_M22	6	B6_N0	3.3-VLV...default
BCDn[1]	Input	PIN_L21	5	B5_N1	3.3-VLV...default
BCDn[0]	Input	PIN_L22	5	B5_N1	3.3-VLV...default
Seven_Segment[6]	Output	PIN_J2	2	B2_N1	3.3-VLV...default
Seven_Segment[5]	Output	PIN_J1	2	B2_N1	3.3-VLV...default
Seven_Segment[4]	Output	PIN_H2	2	B2_N1	3.3-VLV...default
Seven_Segment[3]	Output	PIN_H1	2	B2_N1	3.3-VLV...default
Seven_Segment[2]	Output	PIN_F2	2	B2_N1	3.3-VLV...default
Seven_Segment[1]	Output	PIN_F1	2	B2_N1	3.3-VLV...default
Seven_Segment[0]	Output	PIN_E2	2	B2_N1	3.3-VLV...default

Fig6. Inputs and Output Pin Assignment Description

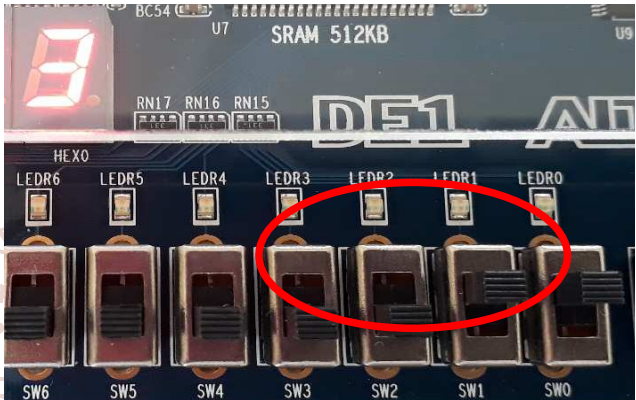


Fig10. Decimal Value 3 on 7Segment for Binary value '0011'

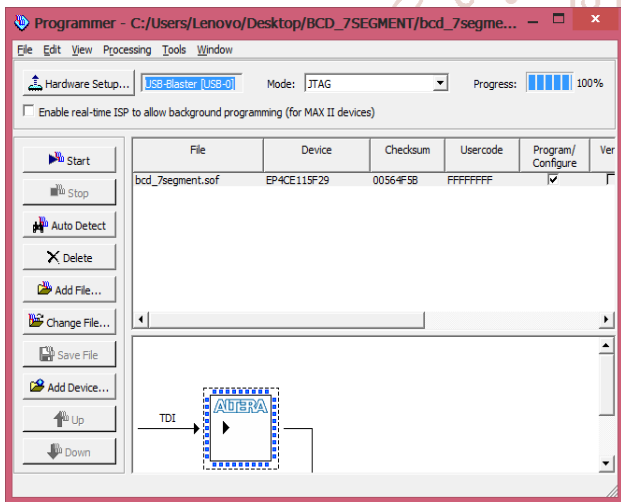


Fig7. BCD_TO_7Segment Program Uploading



Fig11. Decimal Value 4 on 7Segment for Binary value '0100'

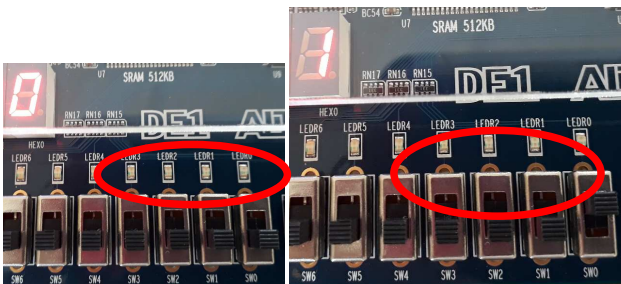


Fig8. Decimal Value 0,1 on 7Segment for Binary value '0000' and '0001'

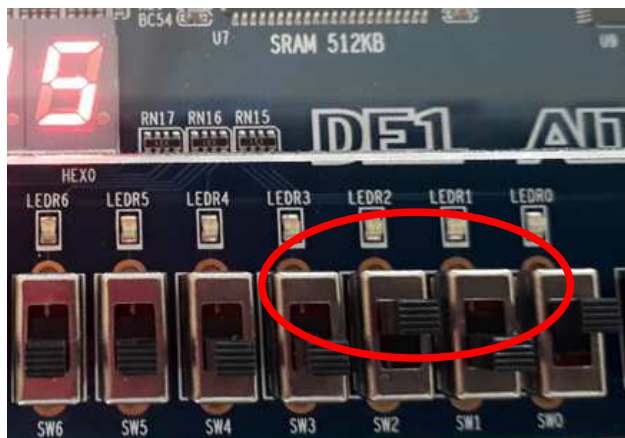


Fig12. Decimal Value 5 on 7Segment for Binary value '0101'

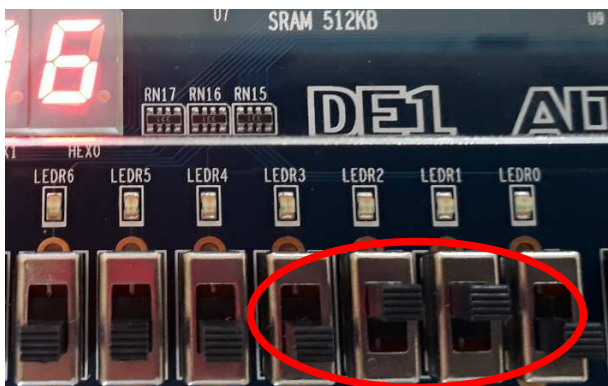


Fig13. Decimal Value 6 on 7Segment for Binary value '0110'



Fig14. Decimal Value 7 on 7Segment for Binary value '0111'



Fig15. Decimal Value 8 on 7Segment for Binary value '1000'

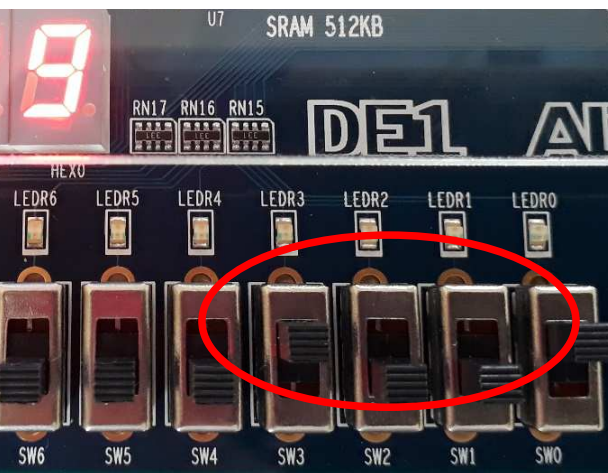


Fig16. Decimal Value 9 on 7Segment for Binary Value '1001'

The digital circuit can be tested by designing not only the VHDL code but also schematic diagram. The operations of master-slave flip-flop and 3-bit up-counter are tested by designing the schematic diagram. Firstly, the schematic diagram of master-slave is designed by using quartus II software. After designing the circuit, it must be compiled to verify that it is correct. And then the pin assignment is done for the input and output ports. Then it is uploaded to DE-115 FPGA Board in order to test the operation of the circuit. The schematic design of the master-slave flip-flop is shown in Figure.

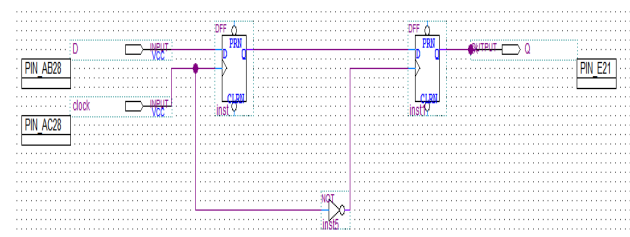


Fig17. Schematic Diagram of Master-Slave Flip-Flop

The inputs and output pin assignment are described in Figure.

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard
clock	Input	P2N_A020	5	B5_N0	2.5 V (default)
D	Input	P2N_A020	5	B5_N0	2.5 V (default)
Q	Output	P2N_E21	7	B7_N0	2.5 V (default)
<<new node>>					

Fig18. Pin Assignment of The Input and Output Ports

The uploading of the schematic diagram is illustrated in Figure. In this step, it is important to check that the hardware setup is in USB-Blaster (USB-0) and JTAG Mode. If it is not like that, it can be changed by selecting the hardware setup button. After doing this process, masterslave.sof must be selected and then Start button must be clicked in order to upload this file to FPGA DE2-115 board. It needs to wait the progress is 100%. After completing the 100% progress, it can be tested the operation of the circuit on FPGA board by applying the various input conditions.

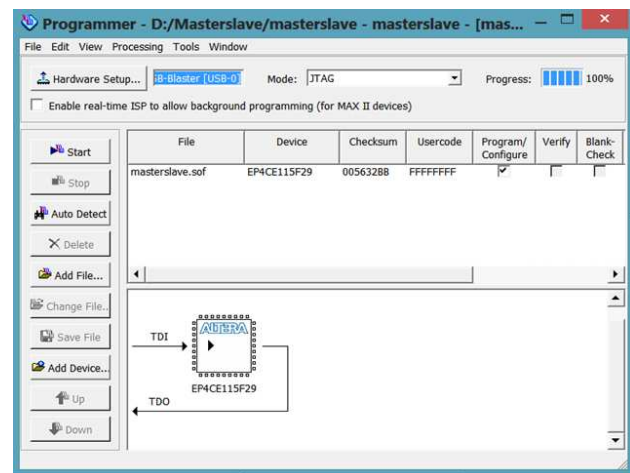


Fig19. Uploading Description of Master-Slave Flip-Flop

The operation results of master-slave flip-flop are shown in Figure. The output of the circuit is assigned with LEDG0. The Clock input is assigned with SW1 and the data input is assigned with SW0 on FPGA board.

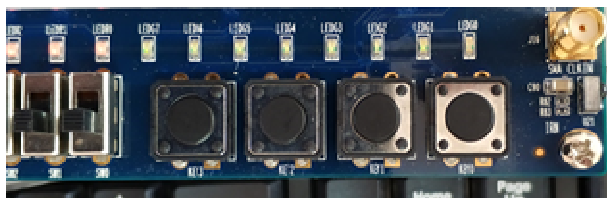


Fig20. Condition of Clock=0 and Input D=0



Fig21. Condition of Clock=1 to 0 and Input D=1



Fig22. Condition of Clock=1 to 0 and Input D=0



Fig23. Condition of Clock=1 and Input D=1

The operation results can be checked with the timing diagram. It is shown in Figure.

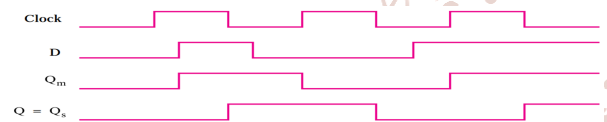


Fig24. Timing Diagram of The Master-Slave Flip-Flop

The 3-bit down-counter is implemented with three T Flip-Flops. The T input is always applied by logic '1'. The three outputs are assigned with Q0,Q1 and Q2. The schematic design of the 3-bit down-counter is shown in Figure.

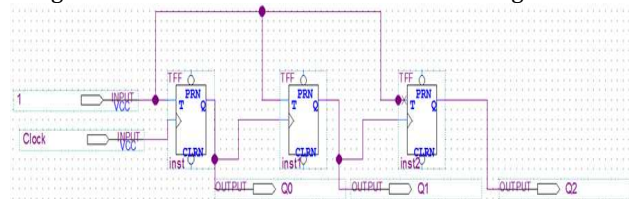


Fig25. Schematic Diagram of The 3-bit Down-Counter

The pin assignment and uploading processes are made like master-slave flip-flop.

The outputs of this circuit are set to '0' value at the initial state. The input T is always set to ON state. The clock input is set to '1010101.....'. The output Q0 toggles its previous value whenever it is arrived at the positive edge of the clock. The output Q1 toggles its previous value at every positive edge of Q1. The output Q2 toggles its previous value at every

positive edge of Q1. The circuit count decimal value 7 (it is the same as $Q_2Q_1Q_0 = 111$) at the first positive edge of the clock pulses. At this stage, the output Q0 arrives at the positive edge of the clock. So, the output Q0 toggle its previous value '0'. Therefore, its vale is logic value '1'. The output Q1 arrives at the positive edge of the Q1 at the same time. So, it toggles its previous value. The value of output Q1 is also logic '1' at this stage. Similarly, the value of output Q2 is logic '1' because the output Q2 toggles its previous value at the positive edge of the Q1.

The inputs T is assigned with SW1 and the clock input is assigned with w0. The outputs Q2, Q1 and Q0 are assigned with LEDG 2, LEDG1 and LEDG0. The operation results are described in Fig.



Fig26. T input '1' and Clock input '1' (At The 1stPositive edge of Clock)

This figure shows the circuit counts the decimal value 7 at the first positive edge of the clock pulse. In this state, the output Q0 toggle its previous value '0' and so its value is logic '1'. Similarly, the output Q1 toggles its previous value '0' and so its value is logic '1'. In the same way, the output Q2 toggles its remain value to logic '1'. Therefore, the outputs are $Q_2Q_1Q_0 = 111$.



Fig27. T input '1' and Clock input '1' (At The 2ndPositive edge of Clock)

This figure shows the circuit counts the decimal value 7 at the second positive edge of the clock pulse. In this state, the output Q0 toggle its previous value '1' and so its value is logic '0'. Similarly, the output Q1 doesn't toggle its previous value '1' and so its value is logic '1'. In the same way, the output Q2 doesn't toggle its remain value to logic '1'. Therefore, the outputs are $Q_2Q_1Q_0 = 110$.



Fig28. T input '1' and Clock input '1' (At The 3rdPositive edge of Clock)

This figure shows the circuit counts the decimal value 7 at the third positive edge of the clock pulse. In this state, the output Q0 toggle its previous value '0' and so its value is

logic '1'. Similarly, the output Q1 toggle its previous value '1' and so its value is logic '0'. In the same way, the output Q2 doesn't toggle its remain value to logic '1'. Therefore, the outputs are Q2Q1Q0 = 101.



Fig29. T input '1' and Clock input '1' (At The 4thPositive edge of Clock)

This figure shows the circuit counts the decimal value 7 at the fourth positive edge of the clock pulse. In this state, the output Q0 toggles its previous value '1' and so its value is logic '0'. Similarly, the output Q1 doesn't toggle its previous value '0' and so its value remains at logic '0'. In the same way, the output Q2 doesn't toggle its remain value to logic '1'. Therefore, the outputs are Q2Q1Q0 = 100.



Fig30. T input '1' and Clock input '1' (At The 5thPositive edge of Clock)

This figure shows the circuit counts the decimal value 7 at the fifth positive edge of the clock pulse. In this state, the output Q0 toggle its previous value '0' and so its value is logic '1'. Similarly, the output Q1 toggles its previous value '0' to logic '1'. In the same way, the output Q2 toggles its remain logic value '1' to logic '0'. Therefore, the outputs are Q2Q1Q0 = 011.



Fig31. T input '1' and Clock input '1' (At The 6thPositive edge of Clock)

This figure shows the circuit counts the decimal value 7 at the sixth positive edge of the clock pulse. In this state, the output Q0 toggle its previous value '1' and so its value is logic '0'. Similarly, the output Q1 remains its previous value '1' in this state. In the same way, the output Q2 remains logic '0'. Therefore, the outputs are Q2Q1Q0 = 010.



Fig32. T input '1' and Clock input '1' (At The 7thPositive edge of Clock)

This figure shows the circuit counts the decimal value 7 at the seventh positive edge of the clock pulse. In this state, the output Q0 toggle its previous value '0' to logic '1'. Similarly, the output Q1 toggles its previous value '1' to logic '0'. In the same way, the output Q2 remains logic value '0'. Therefore, the outputs are Q2Q1Q0 = 001.



Fig33. T input '1' and Clock input '1' (At The 8thPositive edge of Clock)

This figure shows the circuit counts the decimal value 7 at the eighth positive edge of the clock pulse. In this state, the output Q0 toggle its previous value '1' and so its value is logic '0'. Similarly, the output Q1 remains its previous value '0' in this state. In the same way, the output Q2 remains logic '0'. Therefore, the outputs are Q2Q1Q0 = 000.

After counting the decimal value 7 to 0, this circuit will count continuously like previous ways (7 to 0) till the inputs are applied.

V. CONCLUSION

This paper presents the logic circuits operation and the usage of VHDL language. And also the FPGA development kits are applied to analyze the digital logic circuit operations. There are three main parts in this paper. They are logic gate operation, combinational circuit operation and sequential circuit operation. The operations of digital logic circuits are tested by designing the VHDL program in quartus II software and display the operated results on FPGA development kit. The authors analyze the results are correct or not by applying the logic circuit datasheets. All performance results are correct exactly according to their datasheets. Therefore, the VHDL language and FPGA technology can be applied to many larger research applications.

VI. ACKNOWLEDGEMENTS

The authors would like to thank to all her teachers, parents and colleagues who support the help in their job. They also want to express special thanks to all reviewers who review this paper to be a good paper.

VII. REFERENCES

- [1] https://en.wikipedia.org/wiki/Field_programmablegate_array
- [2] Stephen Brown and Zvonko Vranesic, "Fundamental of Digital Logic with VHDL Design", Third Edition.
- [3] "FPGA Architecture for the Challenge". toronto.edu.
- [4] <https://en.m.wikipedia.org>
- [5] <https://www.technocompterscience.com>