**Abstract**

The goals of this experiment are to get familiar with Quartus and FPGA programming, using our knowledge in digital components to implement useful systems.

In this experiment we built a Simple Security System, which consisted of the following components: a priority encoder, memory constructed from 2X1 muxs and d flip-flops, a comparator, 7-segment display driver, enable ports, and input AND gates.

**Table of Contents**

1. Introduction ……………………………………………………………………………………………………………3
2. Theory …………………………………………………………………………………………………………………….4
   1. 4X2 Encoder …………………………………………………………………………………………………4
   2. 7 –Segment Driver ……………………………………………………………………………………….5
   3. The Memory System …………………………………………………………………………………….6
   4. The Comparator …………………………………………………………………………………………...8
3. System …………………………………………………………………………………………………………………….9
4. Conclusion ……………………………………………………………………………………………………………..10
5. References ……………………………………………………………………………………………………………..11

**1.0 Introduction**

The aim of this experiment was to use two tools in constructing useful circuits. The first was Quartus program, which we had used before in previous experiments. The second was FPGA, short for Field Programmable Gate Array which is an integrated circuit specified using hardware description language, meaning that we download our designed circuits using Quartus onto the FPGA to see our circuit in action.

We will talk about the components that made up our security system by including the code for each.

1. **Theory**

2.1 4X2 Encoder

The first component is a 4X2 Encoder, or priority encoder to be more specific. The priority encoders output corresponds to the currently active input which has the highest priority. So when an input with a higher priority is present, all other inputs with a lower priority will be ignored. (1)

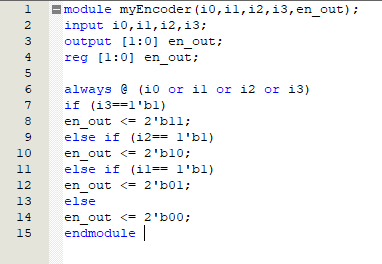
This component takes the input (passcode) from the user

Figure (1)

2.2 7- Segment Driver

The second component is the 7 segment driver. The 7-segment display consists of seven LEDs (hence its name) arranged in a rectangular fashion. Each of the seven LEDs is called a segment because when illuminated the segment forms part of a numerical digit to be displayed. This stores the correct input (passcode) of the system.

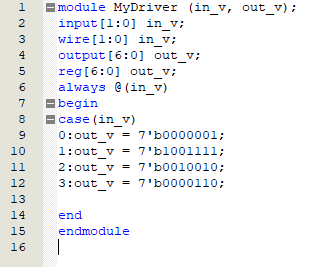


Figure (2)

2.3 The Memory System

Our memory system was constructed using 7 2X1 muxs and 7 D flip-flops. A D flip-flop is a combinational logic circuit designed to switch one of several input lines through to a single common output line. (2) The D-type flip flop are constructed from a gated SR flip-flop with an inverter added between the S and the R inputs to allow for a single D (Data) input. The “D flip flop” will store and output whatever logic level is applied to its data terminal so long as the clock input is high . (3)

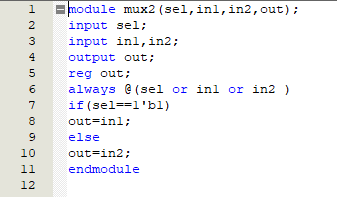


Figure (3)

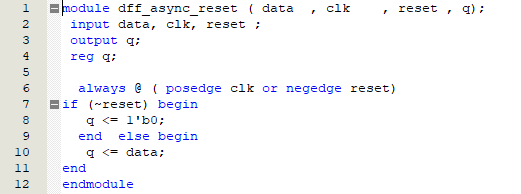


Figure (4)

After successfully compiling the two codes above, blocks were made out of each to form the block diagram of the memory system, as shown in figure (5) in the next page.

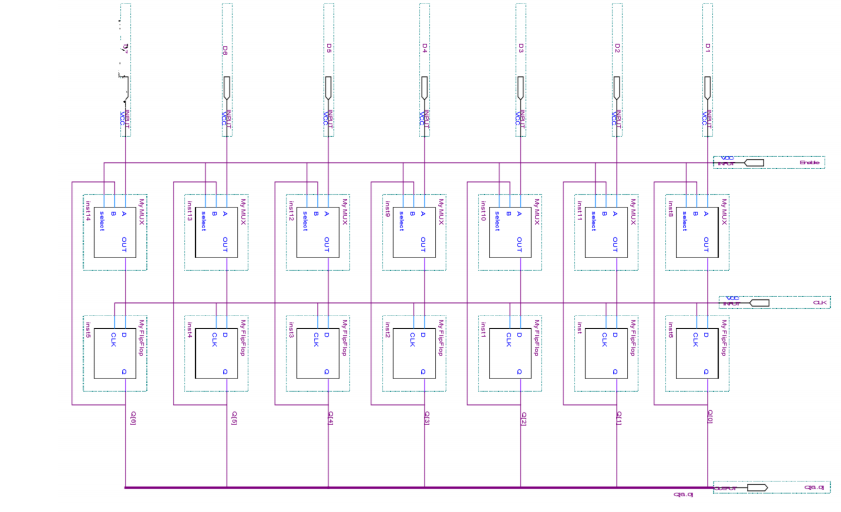


Figure (5)

2.4 The Comparator

The comparator compares the input the user has entered with the information stored in the seven segment. We have two comparators, each connected with a 7-segment, and they work in an alternating manner, which is possible due to the connection of clock inputs to them.

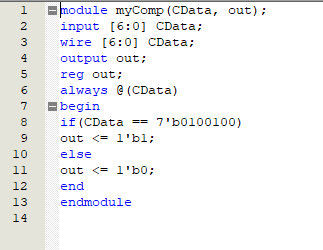


Figure (6)

1. **The system**

After successfully compiling the codes in the previous section, blocks were made out of them, and were assembled as the following figure shows, in addition to adding the input pins, output pins, one AND gate and two invertors.

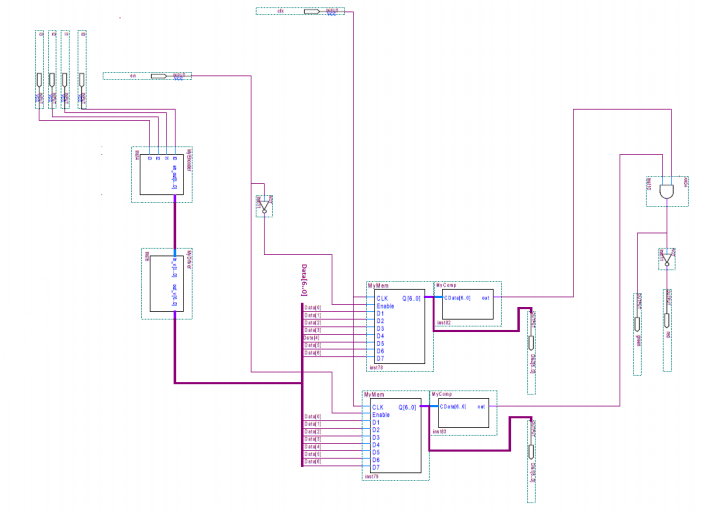


Figure (7)

1. **Conclusion**

This experiment enriched our knowledge in Quartus software and FPGA, and building useful systems out of the components we learnt about previously. We also noticed that it is much easier to work with software programs, especially when it comes to debugging the system or changing different parameters within it.

5.0 References

1. <https://www.electronics-tutorials.ws/combination/comb_4.html>
2. <https://www.electronics-tutorials.ws/combination/comb_2.html>
3. <https://www.electronics-tutorials.ws/sequential/seq_4.html>