ENCS2110

DIGITAL ELECTRONICS AND COMPUTER ORGANIZATION LABORATORY

Experiment 8: Introduction to QuartusII Software

Name: Sara Totah

Student ID: 1181779

Instructor: Khader Mohammed

T.A: Enas Jawabreh

Date: 05/05/2021

Table of Contents

Table of figures

Abstract

The aim of this experiment is to get more experience in using QuartusII and the Verilog HDL language, also, getting to know more about coding some digital circuits such as 4-bit full adder, 4-bit comparator, etc.

Theory

A. QuartusII Software

Is a software that provide user with ability to design and simulate different programmable chip designs.

B. Verilog HDL

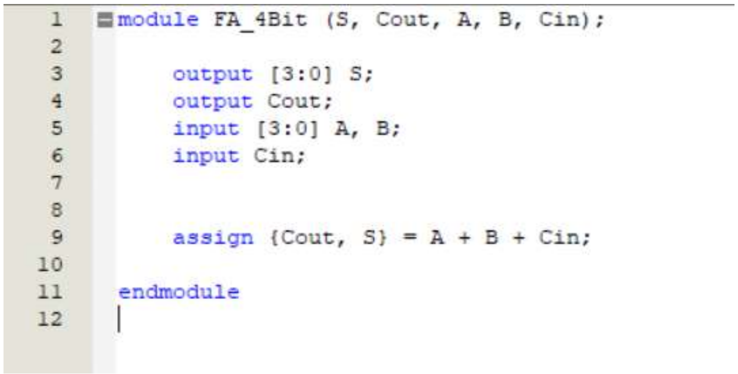
Hardware description language, it describes the hardware of digital systems in a textual form, it also can represent logic diagrams, expressions and complex circuits.

C. Logic Simulation and Synthesis

Logic simulation mainly produce timing diagrams that predicts how the hardware will behave before it is fabricated and allows the detection of functional errors in a design before physically implementing the circuit.

Procedure and Discussion

A. 4-bit Full adder



```
module FA_4Bit (S, Cout, A, B, Cin);

    output [3:0] S;
    output Cout;
    input [3:0] A, B;
    input Cin;


    assign {Cout, S} = A + B + Cin;

endmodule

```

Figure 1: 4-bit Full Adder HDL Verilog code

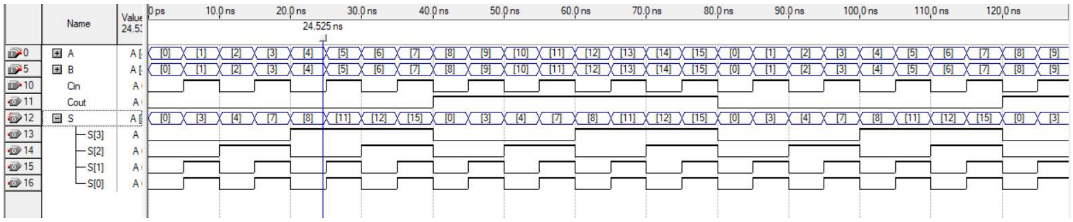
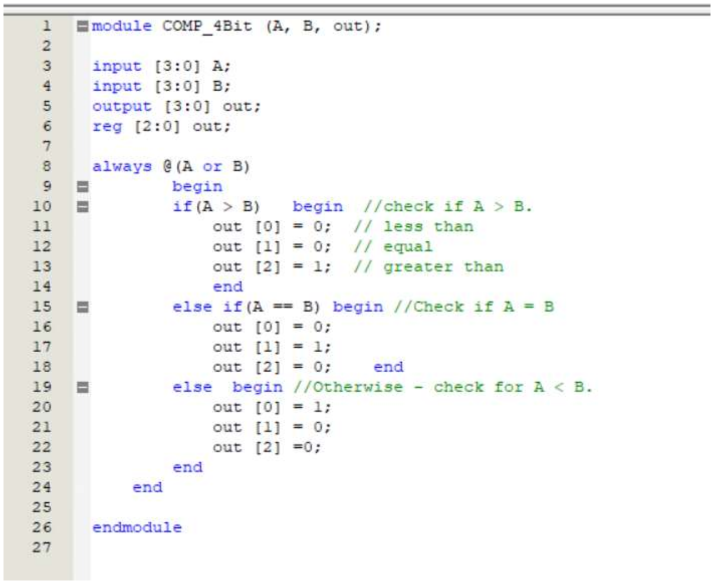


Figure 2: 4-bit Full Adder waveform

B. 4-bit Comparator



```
module COMP_4Bit (A, B, out);

input [3:0] A;
input [3:0] B;
output [3:0] out;
reg [2:0] out;

always @(A or B)
        begin
        if(A > B)   begin  //check if A > B.
            out [0] = 0;  // less than
            out [1] = 0;  // equal
            out [2] = 1;  // greater than
            end
        else if(A == B) begin //Check if A = B
            out [0] = 0;
            out [1] = 1;
            out [2] = 0;    end
        else  begin //Otherwise - check for A < B.
            out [0] = 1;
            out [1] = 0;
            out [2] =0;
        end
    end

endmodule

```

Figure 3: 4-bit Comparator HDL Verilog code

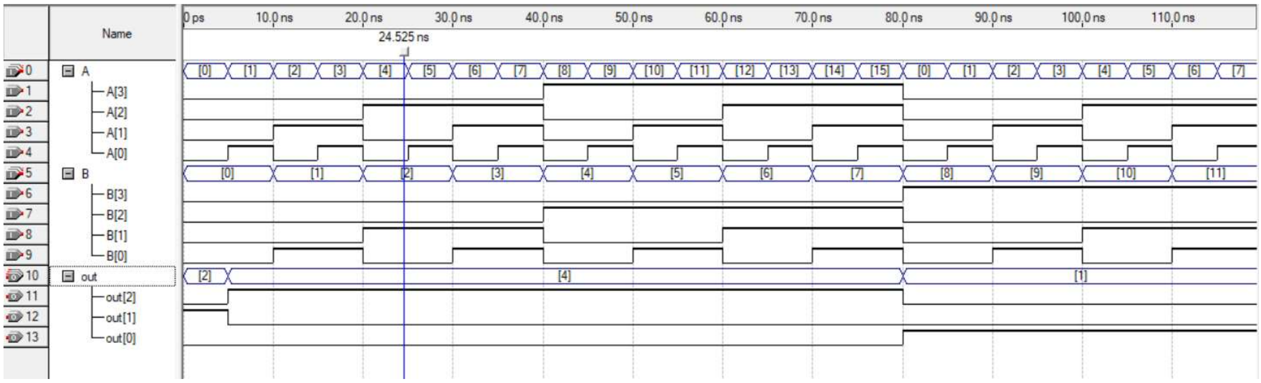
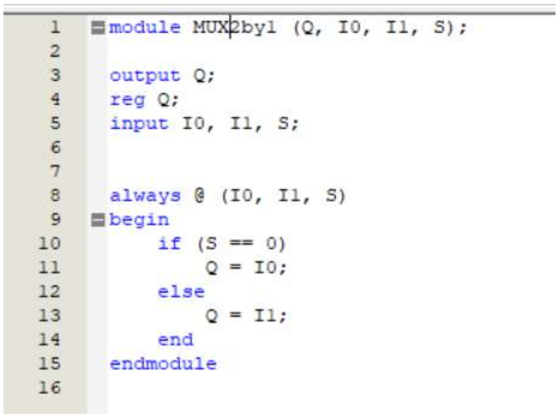


Figure 4: 4-bit Comparator waveform

C. 2 x 1 Multiplexer



```
module MUX2by1 (Q, I0, I1, S);

output Q;
reg Q;
input I0, I1, S;


always @ (I0, I1, S)
begin
    if (S == 0)
        Q = I0;
    else
        Q = I1;
    end
endmodule

```

Figure 5: 2x1 multiplexer HDL Verilog code

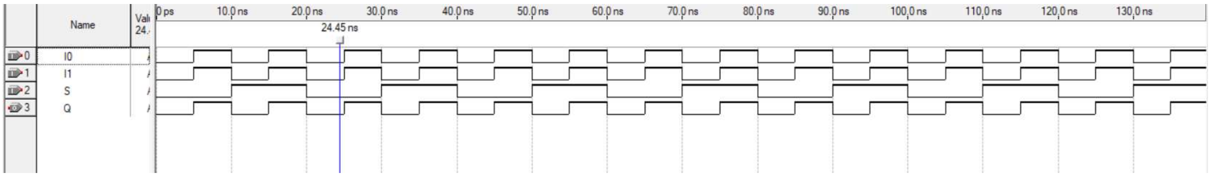
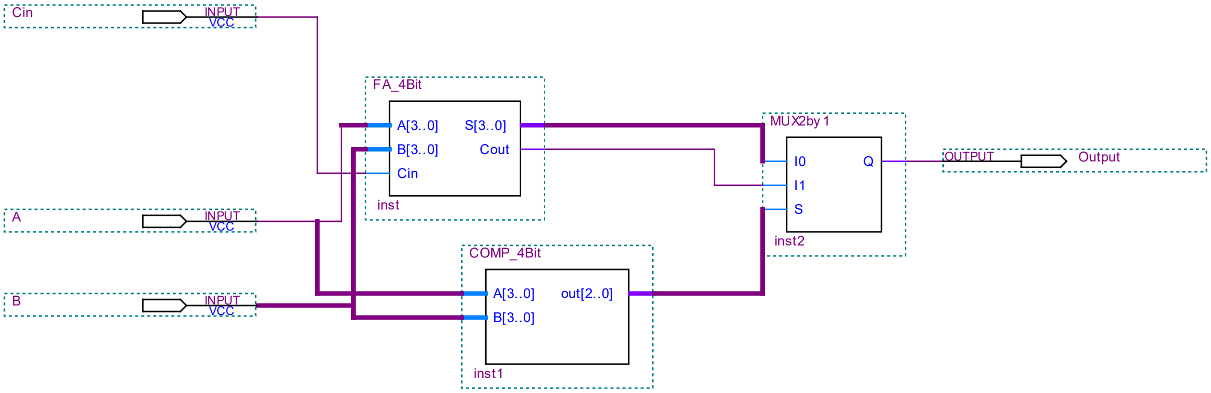


Figure 6: 2x1 multiplexer waveform

Final Block Diagram



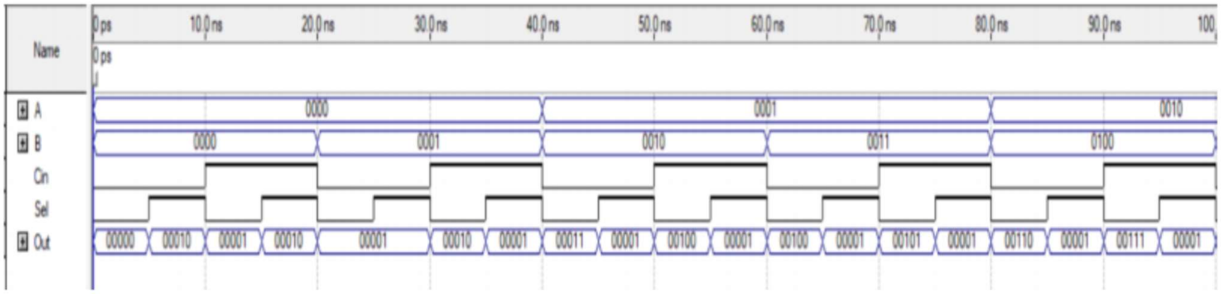


Figure 5: Waveform of the Block Diagram System

All the three circuits mentioned before, were connected together to mage the block diagram above, and it represents the adder-comparator system depending on the selection bit added to the circuit, if it was 0 then the addition of the inputs A and B will be the output, but if the selection bit was 1 then the result will be comparison between A and B.

Conclusion

In this experiment every circuit mentioned before was tested and simulated, all of them worked successfully and gave results as expected, so in conclusion this experiment helped us to build a digital system using different and separated modules.