DIGITAL ELECTRONICS AND COMPUTER ORGANIZATION LABORATORY

Dr. Anjad Badran

Experiment No. 2 - Comparators, Adders and Subtractors

Section 11

Islam Jihad Joma 1191375

T. Enas Jawabrah

21-3-2021

Abstract

In this experiment we have to learn how to use Quartus program, how to use HDL on Quartus and build 4- bits adder and 4 bits coparator and 2x1 multiplixer, we have to design them using Quartus by making codes, we have to simulate them by simulation butto.

Table of content:

Introduction (Theory)

We have to create this figure

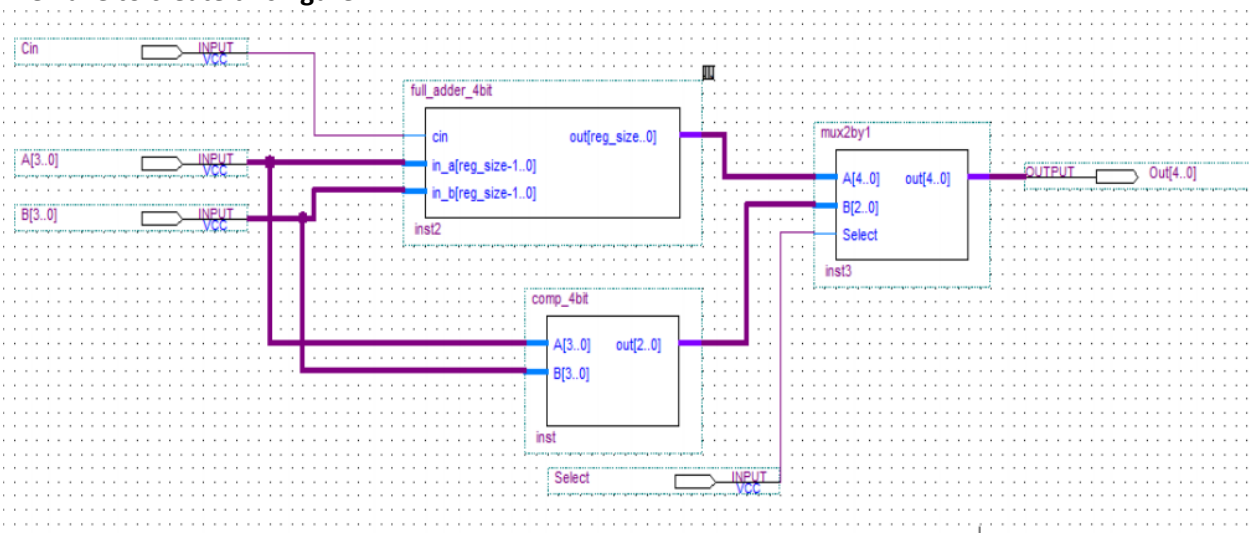


Figure8.2

As a code using Quartus HDL language that helps us to create and then semulate it

In Quartus there is 3 types in writing the code;

Gate-Level Modeling : using instantiation of primitive gate and user defined modules .

Data-Flow Modeling : using continues assignment statements with keyword assign

Behavioral Modeling : using procedural assignment statements with keyword always .

We need to create 1- bit adder so we can create the 4- bits adder as we need too for 4-bits comparator that compare between 4 digits 2 numbers, and we need a multiplixer as to contain the two blockes to work together. We need to semulate it and save the semulation too, and other coding stuff.

Procedure (Discussion & Results)

Part 1 : 4-Bit Full Adder using HDL and Quartus II

We need first to create 1- bit adder which is made of XOR gates and or gate

```
1  module FA (X,Y,Cin,sum,Cout);
2      input X,Y,Cin;
3      output sum,Cout;
4      assign sum = X^Y^Cin;
5      assign Cout = Cin&(X^Y)|(X&Y);
6  endmodule
```

And it accepts 3 inputs X<Y and the Cin from th previus one, and give us output of sum and the carry (Cout)

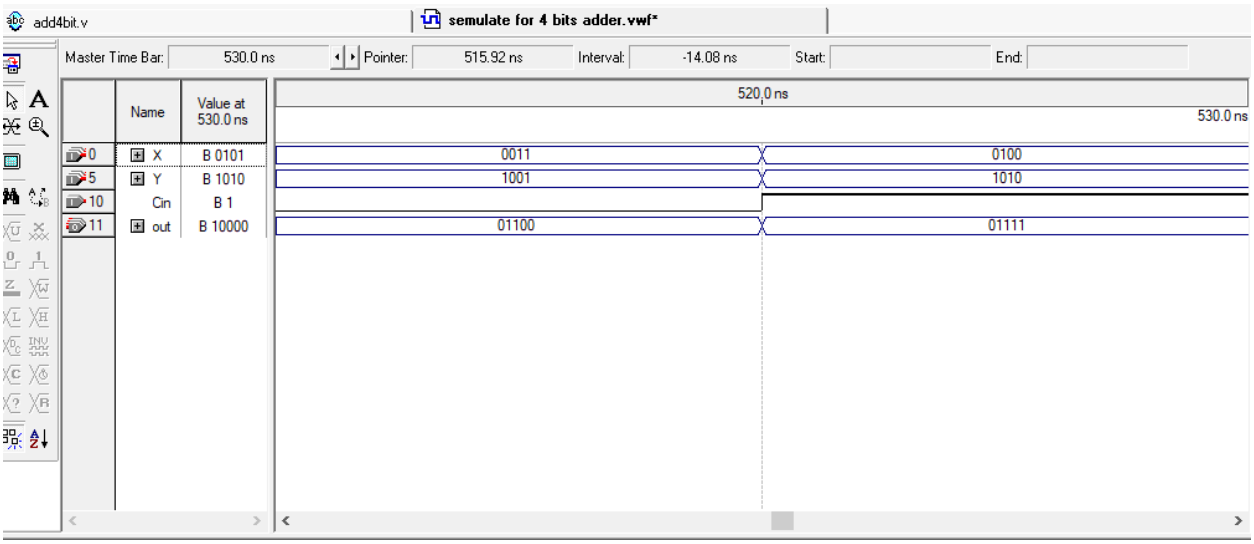
Then I called the function 4 times to create 4-bits adder and coonected them with each other(the output Cout from the previus function is a Cin for thenext function and so on..)

And I got this code

```
1  module add4bit (Cin,X,Y,out);
2      input [3:0] X;
3      input [3:0] Y;
4      input Cin;
5      output [4:0]out;
6      wire [2:0]w;
7      FA FA0(X[0],Y[0],Cin,out[0],w[0]);
8      FA FA1(X[1],Y[1],w[0],out[1],w[1]);
9      FA FA2(X[2],Y[2],w[1],out[2],w[2]);
10     FA FA3(X[3],Y[3],w[2],out[3],out[4]);
11
12 endmodule
13
```

I mareged the Cout and the sum and make them one number made of 5 bits as the teacher asked.

Then I semulated it and got a true results and I'll share a picture for a part of the semulation

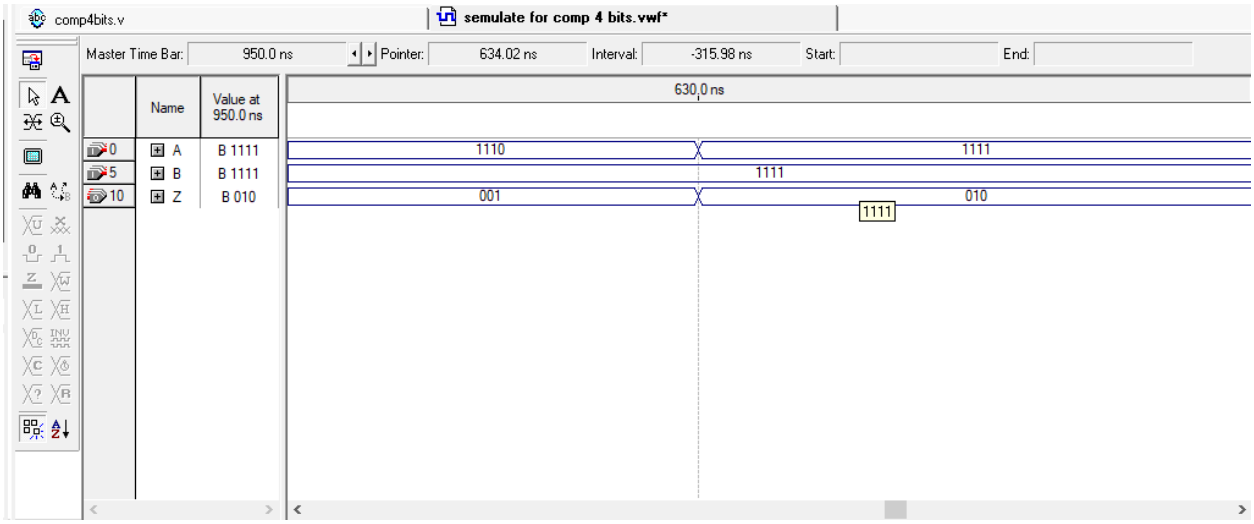

add4bit.v
semulate for 4 bits adder.vwf*
Master Time Bar: 530.0 ns
Pointer: 515.92 ns
Interval: -14.08 ns
Start:
End:
Name
Value at 530.0 ns
520.0 ns
530.0 ns
0 X B 0101
5 Y B 1010
10 Cin B 1
11 out B 10000
0011
0100
1001
1010
01100
01111

- Part 2 : 4-Bit Comparator using Quartus II

To create the comparator of 4 bits firsts it's made of inputs(X,Y) and the output as 3 bits, if the middle number is 1 and the rest is zeros then that's means the 2 numbers are equals(010) but if the outbut was like (100) that's means the first number is bigger, and if (001) means the second number is bigger. Here is the code:

```
1
2  module comp4bits(A,B,Z);
3  input [3:0]A;
4  input [3:0]B;
5  output [2:0]Z;
6
7  assign Z[2]= (A>B)? 1:0;
8  assign Z[1]= (A==B)? 1:0;
9  assign Z[0]= (A<B)? 1:0;
10
11 endmodule
```

And here is the semulation:



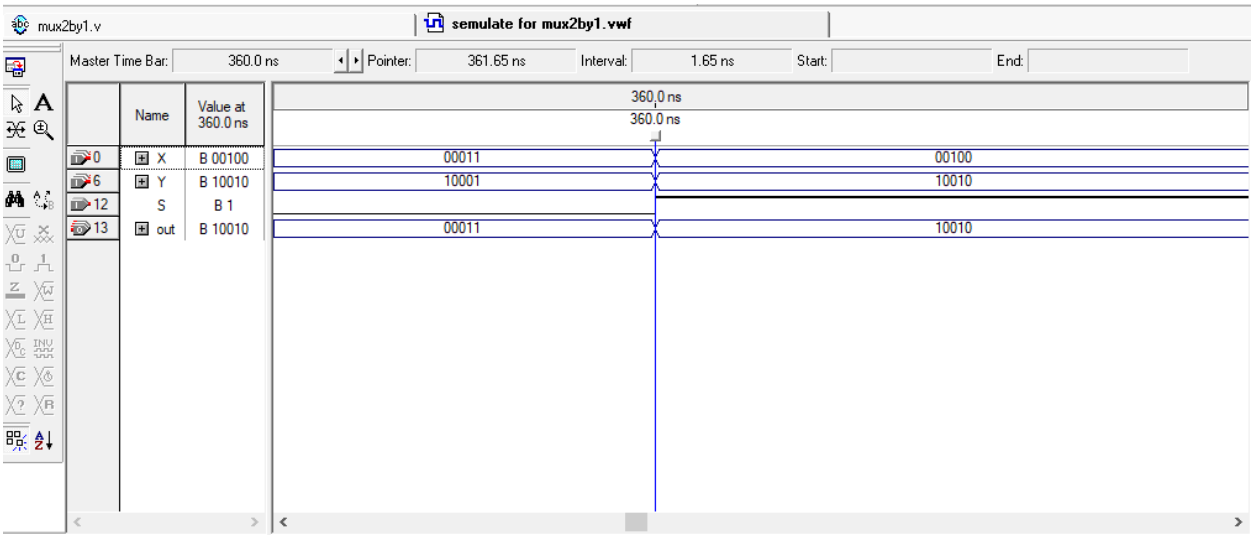
And all the results were true

- Part 3 : 2X1 MUX using Quartus II

We have to made this part to connect the 2 parts together as it's needed so I wrote the code as this:

```
1  module mux2by1 (X,Y,S,out);
2      input [4:0]Y;
3      input [4:0]X;
4      input S;
5      output [4:0]out;
6      assign out = (S==0)? X : Y;
7  endmodule
8
9
```

And the semulation was like this



and the results were true for all tries

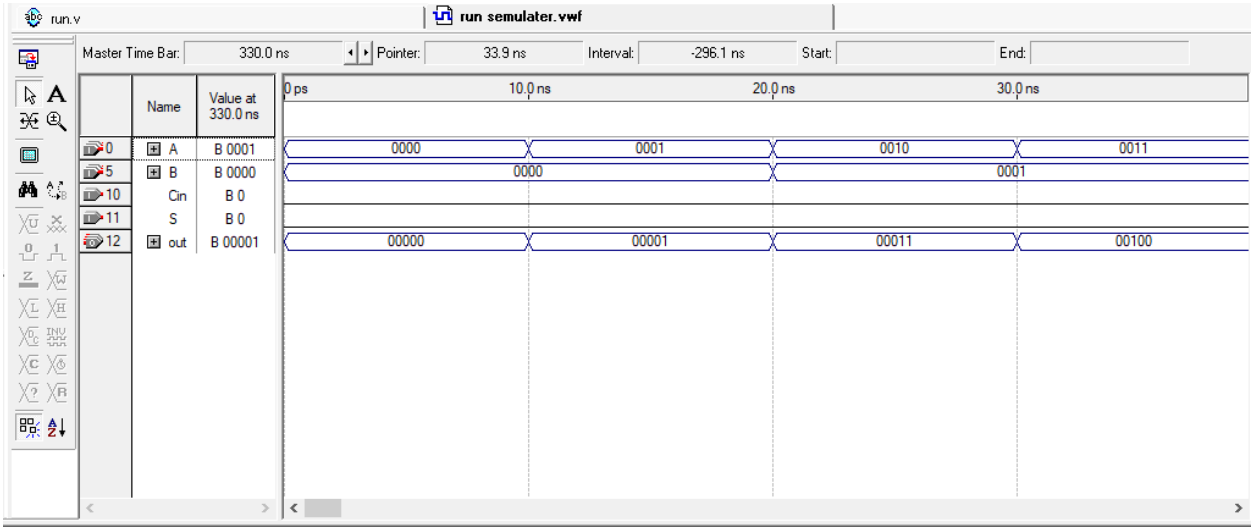
- Part 4 : run all together using Quartus II

Here I called all the 3 functions and make the outputs for some of them an inputs for others to run the program once using it

Here is the code

```
1   module run(A,B,Cin,S,out);
2     input [3:0]A;
3     input [3:0]B;
4     input S,Cin;
5     output [4:0]out;
6     wire [4:0]w1,w0;
7
8     add4bit v1(Cin,A,B,w0);
9
10    comp4bits v2(A,B,w1);
11
12    mux2by1 v3(w0,w1,S,out);
13
14    endmodule
```

And the semulation was as following:

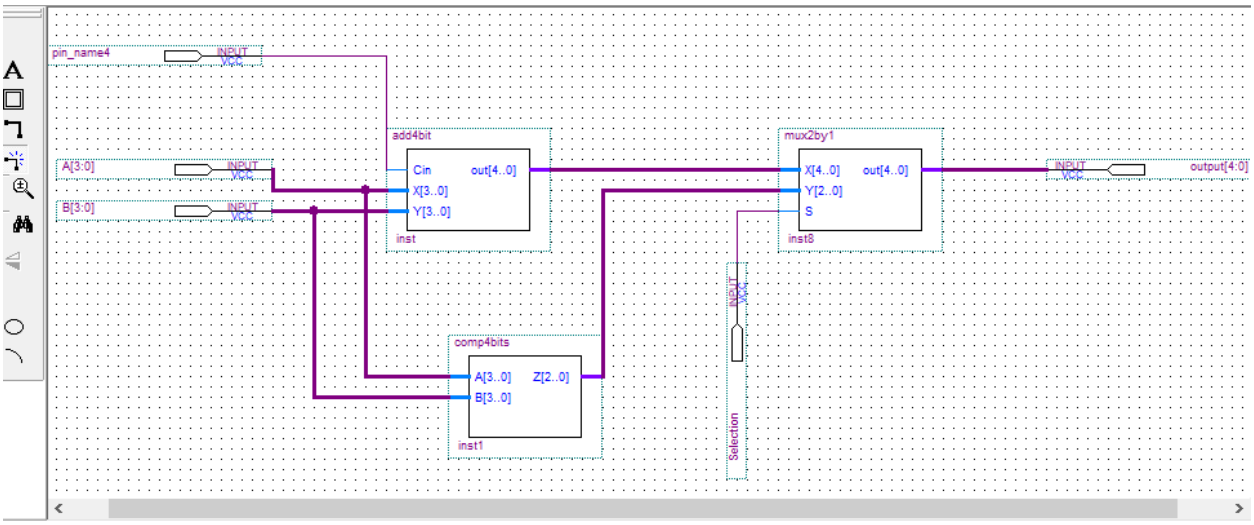


And all the semulation result were true and correct.

Conclusion

Intthis expirement I became famelier with Quartus and HDL built the 4-bits adder and 4-bits comparator and and the multiplixer and how to connect them to gother. And I have verified the correctness of my work by using the semulation button for every single code. Now I can create more and more codes with this program.

And here is a diagram for all the work made by me shows all the blocks and how they got connected togother:



References

In this expirement I didn’t get help from outside the ALL Experimets PDF, only my own work.

Appendix

I didn’t get help from outside the ALL Experimets PDF, only my own work.