
---

**Faculty of Engineering and Technology**

**Electrical and Computer Engineering Department**

**ENCG2340-Digital Systems**

**-----Verilog HDL Project Report-----**

---

**Prepared by:**

**Student's Name:** Dunia Jaser

**Student's ID**: 1201345

**Instructor**: Dr. Hanna Bullata

**Section: 4**

---

**The aim from this project is to build a BCD adder—subtractor using Verilog HDL (Hardware Description Language)**

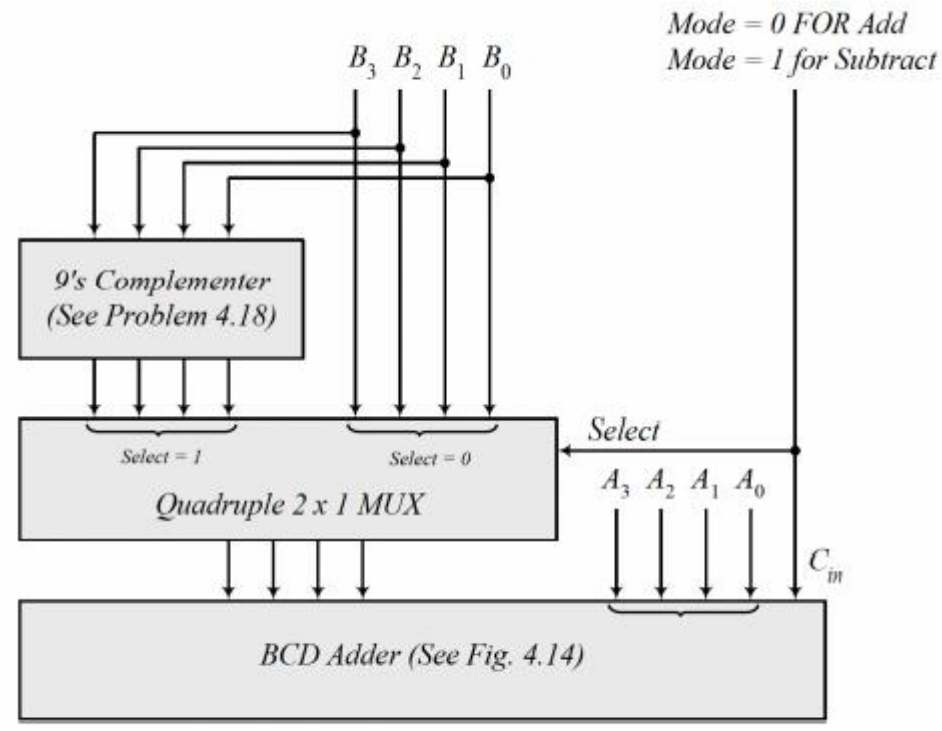



Figure 1 BCD adder-subtractor circuit

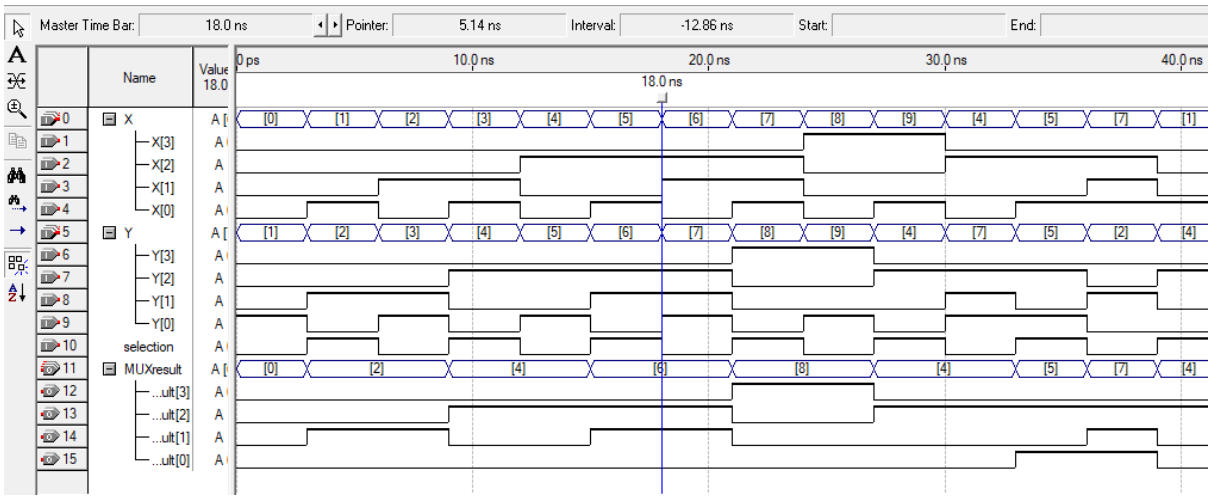
- ***First module: 2-to-1 Multiplexer.***

Design a combinational circuit that describes the quadruple 2X1 multiplexer using a **behavioral model**. Develop and simulate a **behavioral model** (don't use the data flow model for this component).

1. Verilog code:

```
1   // Dunia Jaser 1201345
2   module MUX_2_x_1(MUXresult,X,Y,selection);
3   input [3:0] X;//the 4 bit bianry number.
4   input [3:0] Y;// the 9's complement of the binary number.
5   input selection;
6
7   output [3:0] MUXresult;
8   reg [3:0]  MUXresult;
9
10  always @(X or Y or selection)
11  if (selection == 1)
12      MUXresult = Y;//Subtract case
13  else
14      MUXresult = X;//Add case
15  endmodule
```

2. Simulation:




In this component, when the selection is zero then we will take the number in X, and when the selection is 1 then we will take the number in Y. Look at the vector waveform.

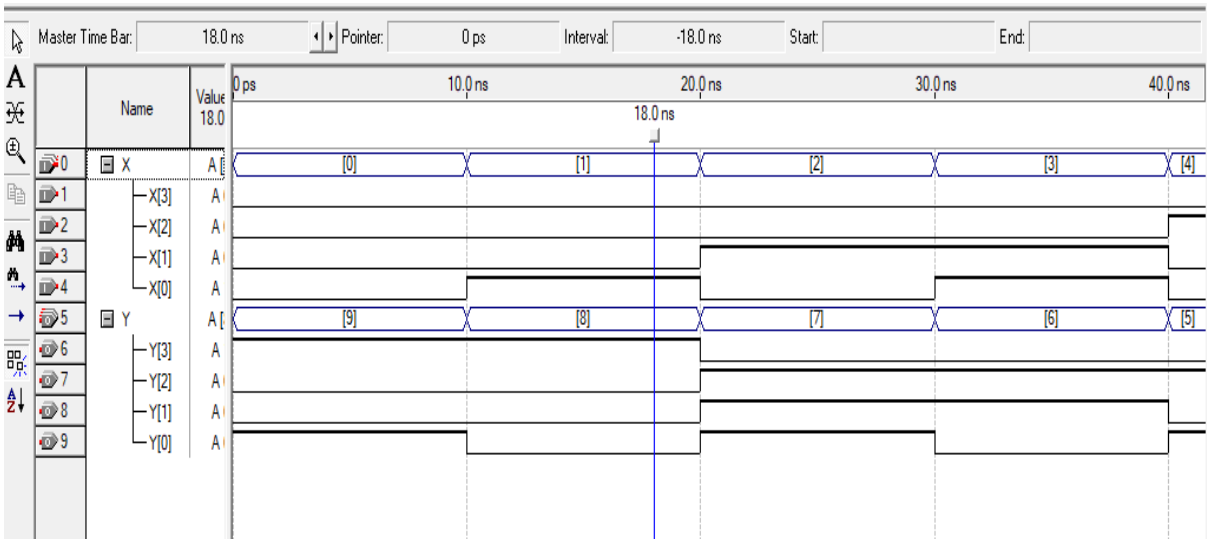
- ***Second module: 9's Complement***

Design a combinational circuit that generates the 9's complement of a BCD digit. Develop and simulate a **data flow model** (don't use the behavioral model for this component) of a circuit that generates the 9's complement.

**1.** Verilog code:

```
1   // Dunia Jaser 1201345
2   module nineComplements(Y,X);
3   input [3:0] X;
4   output [3:0] Y;
5
6
7   assign Y[0] = !X[0];
8   assign Y[1] = X[1];
9   assign Y[2]= X[2]^X[1];
10  assign Y[3]= (!X[3]) && (!X[2]) && (!X[1]) ;
11
12  endmodule
```

**2.** Simulation:




In this component, we will find the 9's complement of the number in X. For example, if the number 0 in X then the 9's complement is 9 in Y. Look at the vector waveform.

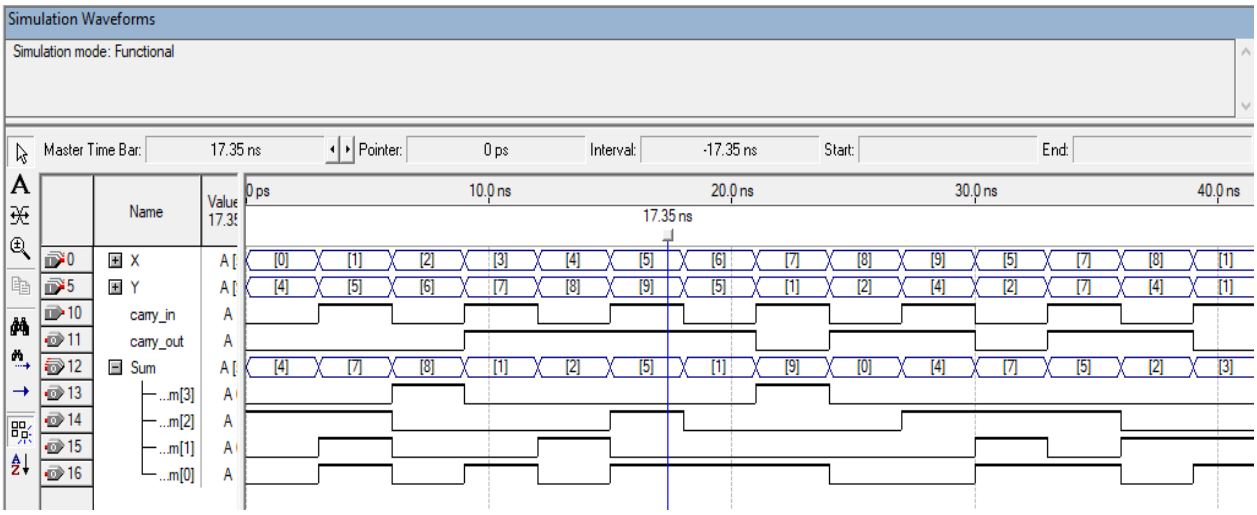
- ***Third module: BCD-Adder.***

Design a combinational circuit for BCD adder. See Figure 4.14 of the textbook for the whole design of the BCD adder. Develop and simulate a **data flow model** (don't use the behavioral model for this component) for the BCD adder.

## 1. Verilog code:

```
1  // Dunia jaser 1201345
2  module BCD_Adder(Sum,carry_out,carry_in,X,Y);
3   input carry_in;
4   input [3:0] X;
5   input [3:0] Y;
6
7   wire [3:0] Temporary;
8   wire [3:0] add_6_or_0;
9   wire temp_carry;
10
11  output [3:0] Sum;
12  output carry_out;
13
14  assign {temp_carry,Temporary}=X+Y+carry_in;
15  // to check whether the output is greater than 6
16  assign  add_6_or_0[0] = 0;
17  assign  add_6_or_0[1] = temp_carry || (Temporary[3]&&Temporary[2]) || (Temporary[1]&&Temporary[3]);
18  assign  add_6_or_0[2] = temp_carry || (Temporary[3]&&Temporary[2]) || (Temporary[1]&&Temporary[3]);
19  assign  add_6_or_0[3]=0;
20
21  assign {carry_out,Sum}= Temporary + add_6_or_0;
22  endmodule
```

## 2. Simulation:




Look at the waveform, in this component, we will find out the sum of X, Y, and carry_in. For example, when the number in X is 1, the number in Y is 5, and the number in carry_in is 1, then the result is 1+5+1 = 7.

- ***Final module: BCD-Adder-Subtractor.***

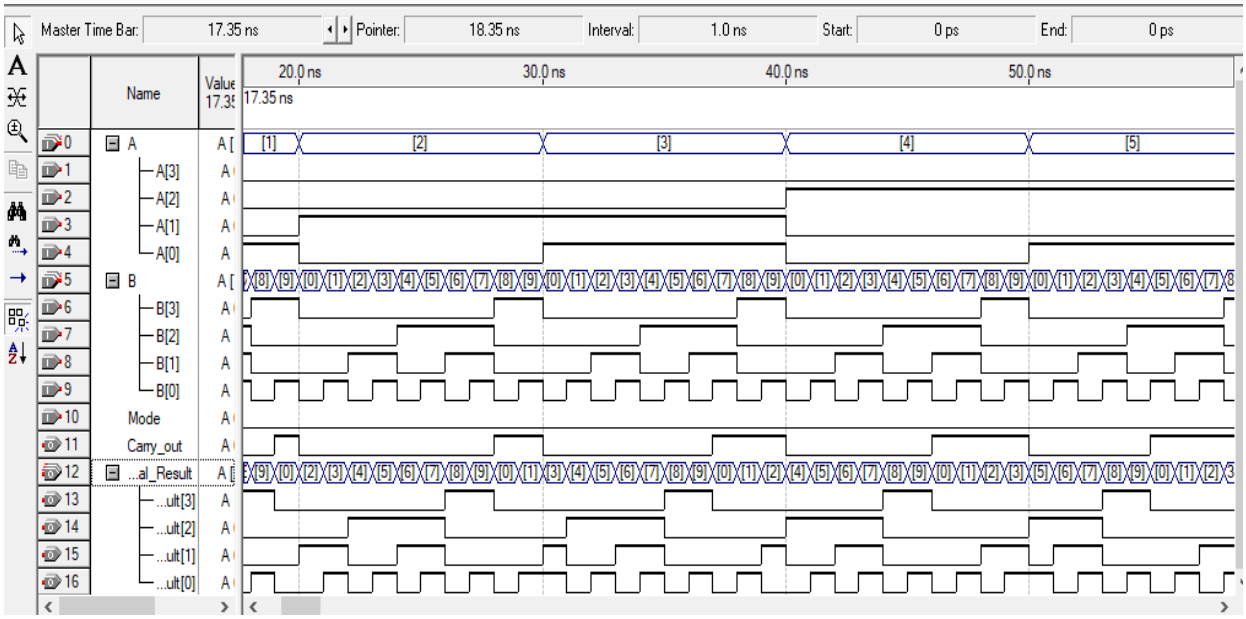
Integrate the whole components of the system by developing and simulating a **structural model** of Figure 1. In your design, make sure that (1) the BCD adder should be described as a **data flow model** in a separate module, (2) the 9's complementor should be described as **data flow model** in a separate module, and (3) the quadruple 2X1 multiplexer be described as a **behavioral model** in a separate module. Finally, all the system components are to be instantiated in a top-level module using a **structural model**.

## 1. Verilog code:

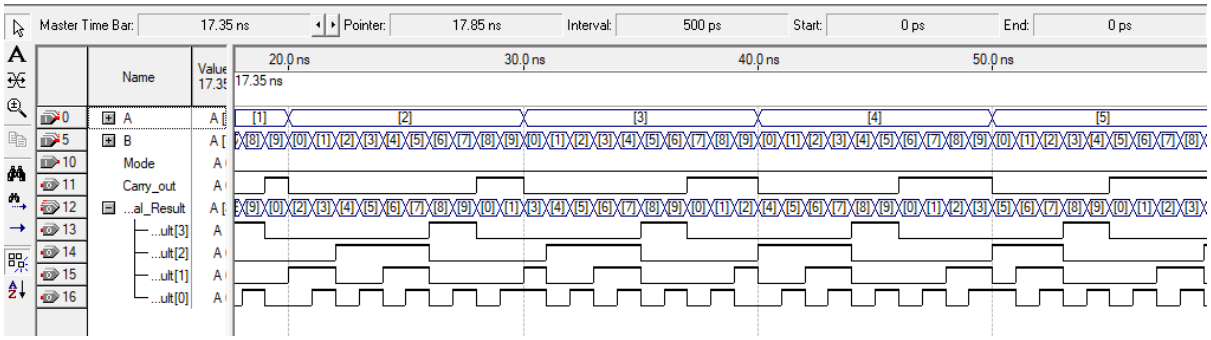
```
1   // Dunia Jaser 1201345
2   module BCD_Adder_Subtractor(A,B,Mode,Final_Result,Carry_out);
3   input [3:0] A;
4   input [3:0] B;
5   input Mode;
6
7   output Carry_out;
8   output [3:0] Final_Result;
9
10  wire [3: 0] nines_Comp;
11  wire [3: 0] mux_out;
12
13  nineComplements G0(nines_Comp,B);
14
15  MUX_2_x_1 G1(mux_out,B,nines_Comp,Mode);
16
17  BCD_Adder G2(Final_Result,Carry_out,Mode,A,mux_out);
18
19
20  endmodule
```

## 2. Simulation:

**Summation Case:**



Another clearer one:

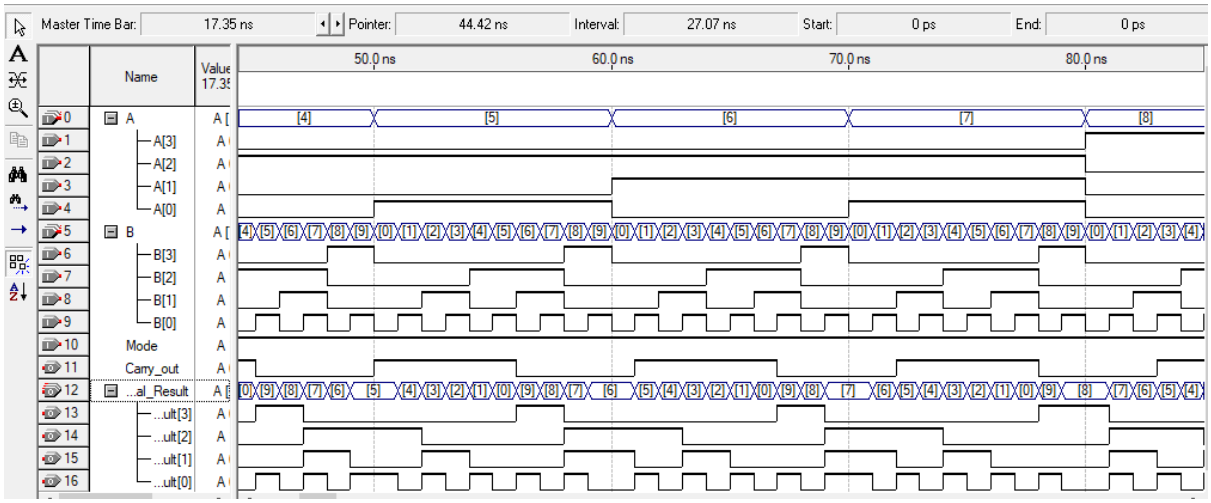



In this case, Mode is 0 (Summation), we will count the sum of A and B. For example, when the number in A is 3 and the number in B is 5, then the result will be 8. There is more examples in the vector waveform above.

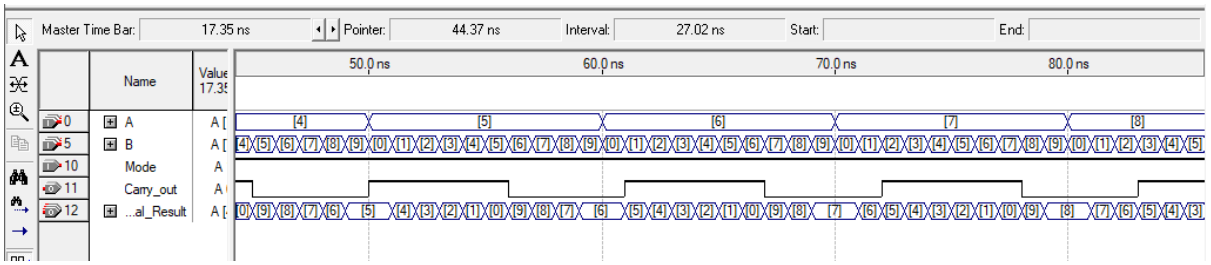
---

**Subtraction Case:**

Note: Carry_out = 1 in dicates a positive result, Carry_out = 0 indicates a negative result in this case.




Another clearer one:




In this case, the Mode is 1 (Subtraction), we will find out the result of A subtract B. For example, there are 3 cases:

1. (A = B) the answer will be 0.
2. (A>B), the answer is (9's complement of B + Mode + A).
3. (A<B), the answer is (9's complement of B + Mode + A).

There's a lot of examples in the simulation above.

## —— BLOCK DIAGRAM TO THE WHOLE CIRCUIT ——