Department of Electrical & Computer Engineering

ENCS2340 - Digital Systems

ALU Verilog Project

Prepared by: Mohammad Milhem

Student ID: 1211053

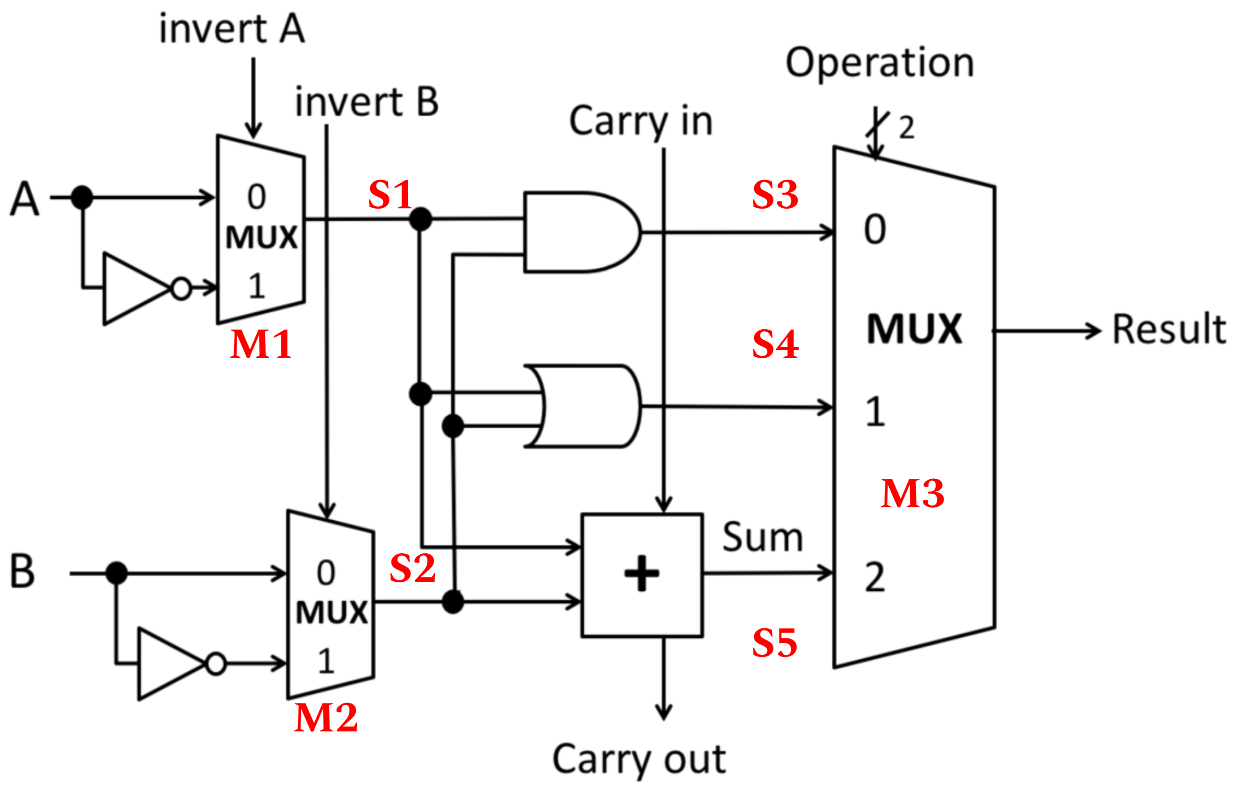
Instructor: Ayman Hroub

Date: August 18, 2022

Table of Contents

Introduction

In my project, I need to implement a 1-bit ALU circuit shown in the Figure below. For this, I will build the system components (Muxs and adder) separately using Verilog HDL. Then, I will integrate the different system components to build the whole system. Each of the system components, as well as, the entire system will be verified using simulation. Also, we will drive the circuit's functionality (using Truth table) when changing the control inputs (invert A, invert B, Carry in, and Operation).



** Adding the red tags to make working with wires easier.

Components

Full Adder

The Full Adder circuit will take 3 inputs A, B, and Cin, and it calculates the summation of input bits and set it as Sum, which is the first output, then it will set the reminder in Cout.

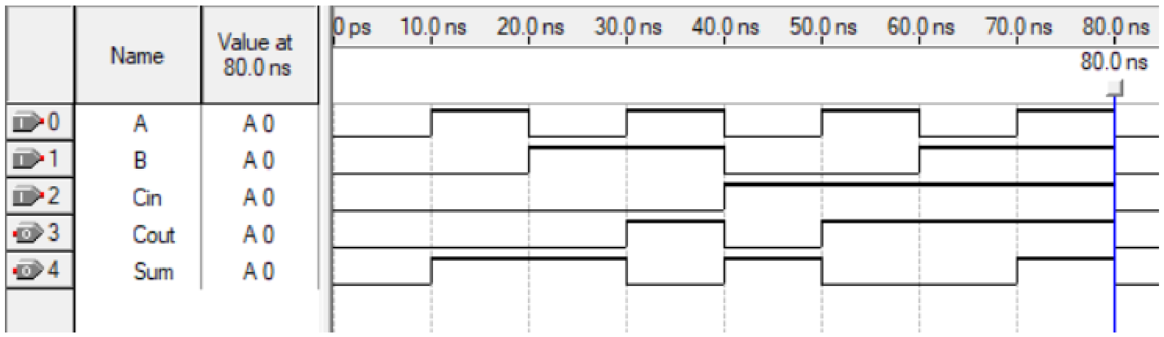
Code:

```
module Full_Adder (A,B,Cin, Sum ,Cout);

input A,B,Cin;
output Sum, Cout;
reg Sum, Cout;

always @ (A,B,Cin)
begin

{Cout , Sum} = A + B + Cin;


end
endmodule
```

Implementation of FA circuit

Simulation

The timing diagram of entering every combination of the input (Testing).



Truth Table:

The truth table of the full adder circuit using all the combinations of the inputs will be shown in the next figure.

A	B	Cin	Sum	Cout
0	0	0	0	0
1	0	0	1	0
0	1	0	1	0
1	1	0	0	1
0	0	1	1	0
1	0	1	0	1
0	1	1	0	1
1	1	1	1	1

The corresponding data for the timing diagram and the truth table are compatible which verifies the design of the circuit.

2-to-1 bit Multiplexer

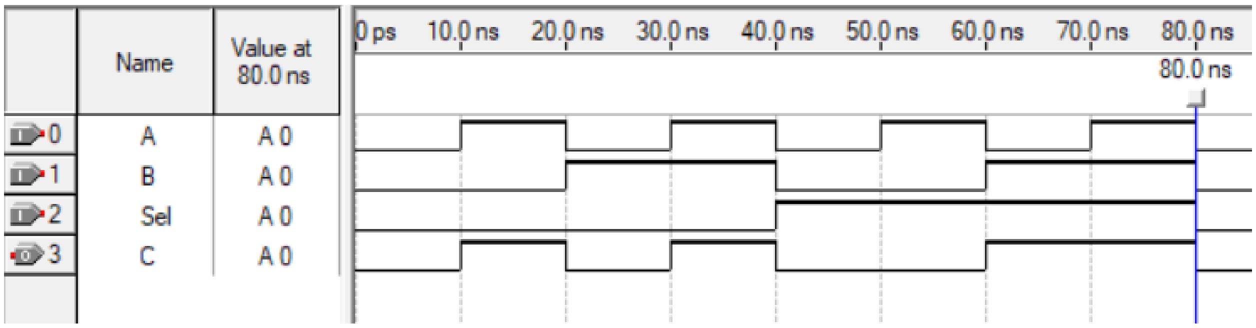
The 2-to-1 bit Multiplexer circuit will take 3 inputs A, B, and Selection bit, and based on the value of the selection bit A or B will be passed as the only output C.

Code:

```
module Mux2(A,B,Sel,C);

input A,B,Sel;
output reg C;

always @ (A,B,Sel)
begin

if (Sel == 0) C = A;
else C = B;

end
endmodule
```

Simulation:

The timing diagram of entering every combination of the input (Testing).



Truth Table:

The truth table of the 2-to-1 bit Multiplexer circuit using all the combinations of the inputs will be shown in the next figure.

A	B	Sel	C
0	0	0	0
1	0	0	1
0	1	0	0
1	1	0	1
0	0	1	0
1	0	1	0
0	1	1	1
1	1	1	1

The corresponding data for the timing diagram and the truth table are compatible which verifies the design of the circuit.

3-to-1 bit Multiplexer

The 3-to-1 bit Multiplexer circuit will take 4 inputs bits A, B, C, and Selection vector of bits of length 2, and based on the value of the selection vector of bits A or B or C will be passed as the only output Z.

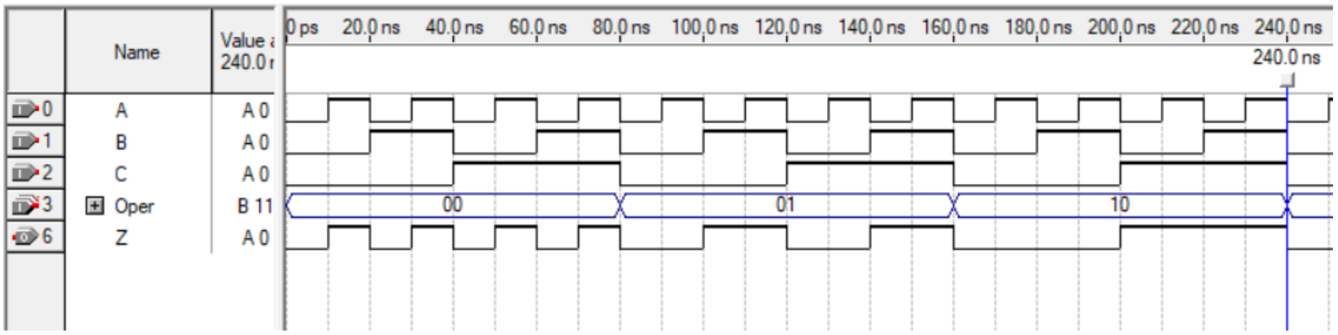
Code:

```
module Mux3(input A,B,C, input [1:0] Oper, output reg Z);

always @(A,B,C, Oper) begin

if (Oper == 2'b00) Z = A;
else if (Oper == 2'b01) Z = B;
else if (Oper == 2'b10) Z = C;

end
endmodule
```

Simulation:

The timing diagram of entering every combination of the input (Testing).

Note** the state of 2’b11 or 3 will be ignored as we only need 3 states.



Truth Table:

The truth table of the 3-to-1 bit Multiplexer circuit using all the combinations of the inputs.

Note** the state of 2'b11 or 3 will be ignored as we only need 3 states.

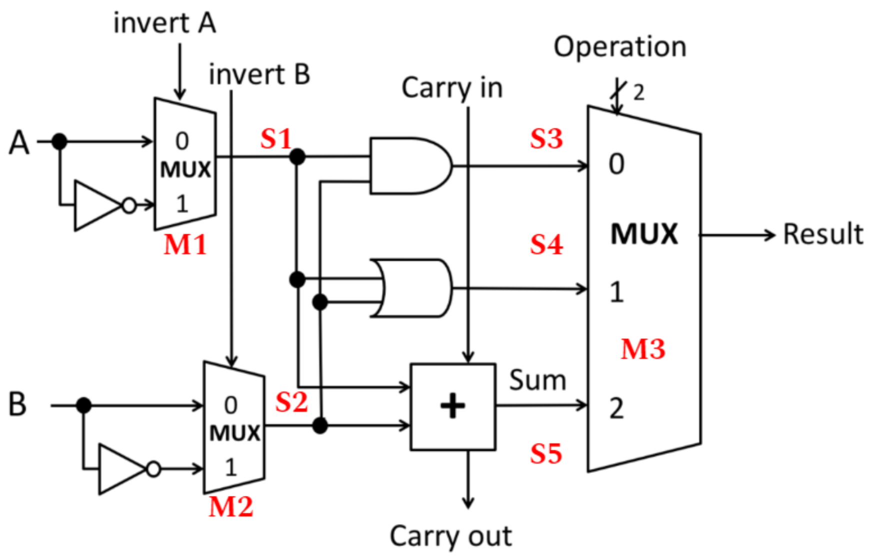
A	B	C	Oper	Z
0	0	0	00	0
1	0	0	00	1
0	1	0	00	0
1	1	0	00	1
0	0	1	00	0
1	0	1	00	1
0	1	1	00	0
1	1	1	00	1
0	0	0	01	0
1	0	0	01	0
0	1	0	01	1
1	1	0	01	1
0	0	1	01	0
1	0	1	01	0
0	1	1	01	1
1	1	1	01	1
0	0	0	10	0
1	0	0	10	0
0	1	0	10	0
1	1	0	10	0
0	0	1	10	1
1	0	1	10	1
0	1	1	10	1
1	1	1	10	1

1-bit ALU

After building each of the components separately using Verilog HDL, the module of the whole system will be as shown in the next figure.

Code:

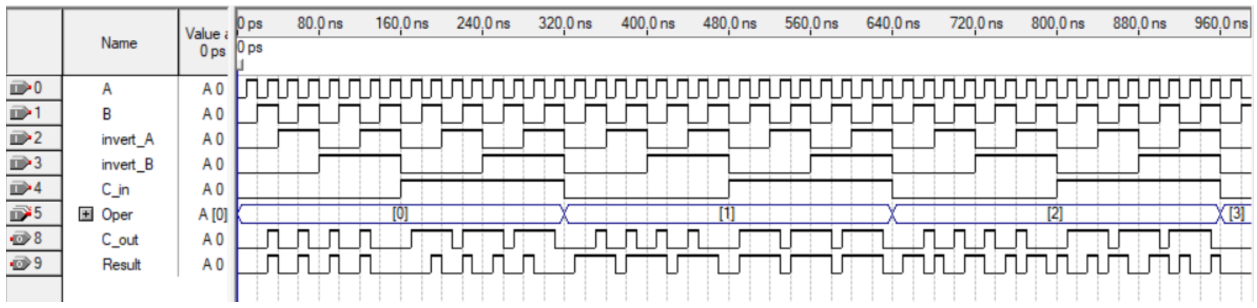
```
module ALU1 (A,B, invert_A , invert_B ,C_in, Oper, Result, C_out);
input A,B,invert_A , invert_B, C_in;
input [1:0] Oper;
output Result, C_out;

wire S1, S2, S3, S4, S5;
Mux2(A,(~A),invert_A ,S1);
Mux2(B,(~B),invert_B ,S2);
and(S3, S1, S2);
or(S4, S1, S2);
Full_Adder(S1 , S2, C_in, S5, C_out);
Mux3(S3, S4, S5, Oper, Result);

endmodule
```



Simulation:

The timing diagram of entering every combination of the input (Testing).

Note** the state of 2’b11 or 3 will be ignored as we only need 3 states.



**count every 10ns.

Truth Table:

The truth table of the 1-bit ALU circuit using all the combinations of the inputs.

The truth table is long due to the 7-bits input, there should be 2^7= 128 different combinations, however, because there is one state we don’t use which is the operation value 3, so we can reduce the size of the truth table to 96 combinations shown by the following table.

A	B	invert A	invert B	Carry in	Operation	Carry out	Result
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
1	1	0	0	0	0	1	1
0	0	1	0	0	0	0	0
1	0	1	0	0	0	0	0
0	1	1	0	0	0	1	1
1	1	1	0	0	0	0	0
0	0	0	1	0	0	0	0
1	0	0	1	0	0	1	1
0	1	0	1	0	0	0	0
1	1	0	1	0	0	0	0
0	0	1	1	0	0	1	1
1	0	1	1	0	0	0	0
0	1	1	1	0	0	0	0

1	1	1	1	0	0	0	0
0	0	0	0	1	0	0	0
1	0	0	0	1	0	1	0
0	1	0	0	1	0	1	0
1	1	0	0	1	0	1	1
0	0	1	0	1	0	1	0
1	0	1	0	1	0	0	0
0	1	1	0	1	0	1	1
1	1	1	0	1	0	1	0
0	0	0	1	1	0	1	0
1	0	0	1	1	0	1	1
0	1	0	1	1	0	0	0
1	1	0	1	1	0	1	0
0	0	1	1	1	0	1	1
1	0	1	1	1	0	1	0
0	1	1	1	1	0	1	0
1	1	1	1	1	0	0	0
0	0	0	0	0	1	0	0
1	0	0	0	0	1	0	1
0	1	0	0	0	1	0	1
1	1	0	0	0	1	1	1
0	0	1	0	0	1	0	1
1	0	1	0	0	1	0	0
0	1	1	0	0	1	1	1
1	1	1	0	0	1	0	1
0	0	0	1	0	1	0	1
1	0	0	1	0	1	1	1
0	1	0	1	0	1	0	0
1	1	0	1	0	1	0	1
0	0	1	1	0	1	1	1
1	0	1	1	0	1	0	1
0	1	1	1	0	1	0	1
1	1	1	1	0	1	0	0
0	0	0	0	1	1	0	0
1	0	0	0	1	1	1	1
0	1	0	0	1	1	1	1
1	1	0	0	1	1	1	1

0	0	1	0	1	1	1	1
1	0	1	0	1	1	0	0
0	1	1	0	1	1	1	1
1	1	1	0	1	1	1	1
0	0	0	1	1	1	1	1
1	0	0	1	1	1	1	1
0	1	0	1	1	1	0	0
1	1	0	1	1	1	1	1
0	0	1	1	1	1	1	1
1	0	1	1	1	1	1	1
0	1	1	1	1	1	1	1
1	1	1	1	1	1	0	0
0	0	0	0	0	2	0	0
1	0	0	0	0	2	0	1
0	1	0	0	0	2	0	1
1	1	0	0	0	2	1	0
0	0	1	0	0	2	0	1
1	0	1	0	0	2	0	0
0	1	1	0	0	2	1	0
1	1	1	0	0	2	0	1
0	0	0	1	0	2	0	1
1	0	0	1	0	2	1	0
0	1	0	1	0	2	0	0
1	1	0	1	0	2	0	1
0	0	1	1	0	2	1	0
1	0	1	1	0	2	0	1
0	1	1	1	0	2	0	1
1	1	1	1	0	2	0	0
0	0	0	0	1	2	0	1
1	0	0	0	1	2	1	0
0	1	0	0	1	2	1	0
1	1	0	0	1	2	1	1
0	0	1	0	1	2	1	0
1	0	1	0	1	2	0	1
0	1	1	0	1	2	1	1
1	1	1	0	1	2	1	0
0	0	0	1	1	2	1	0

1	0	0	1	1	2	1	1
0	1	0	1	1	2	0	1
1	1	0	1	1	2	1	0
0	0	1	1	1	2	1	1
1	0	1	1	1	2	1	0
0	1	1	1	1	2	1	0
1	1	1	1	1	2	0	1

The corresponding data for the timing diagram and the truth table are compatible which verifies the design of the circuit.

THE END