Digital Electronics & Microprocessors Laboratory (EC2P006)

EXPERIMENT-3

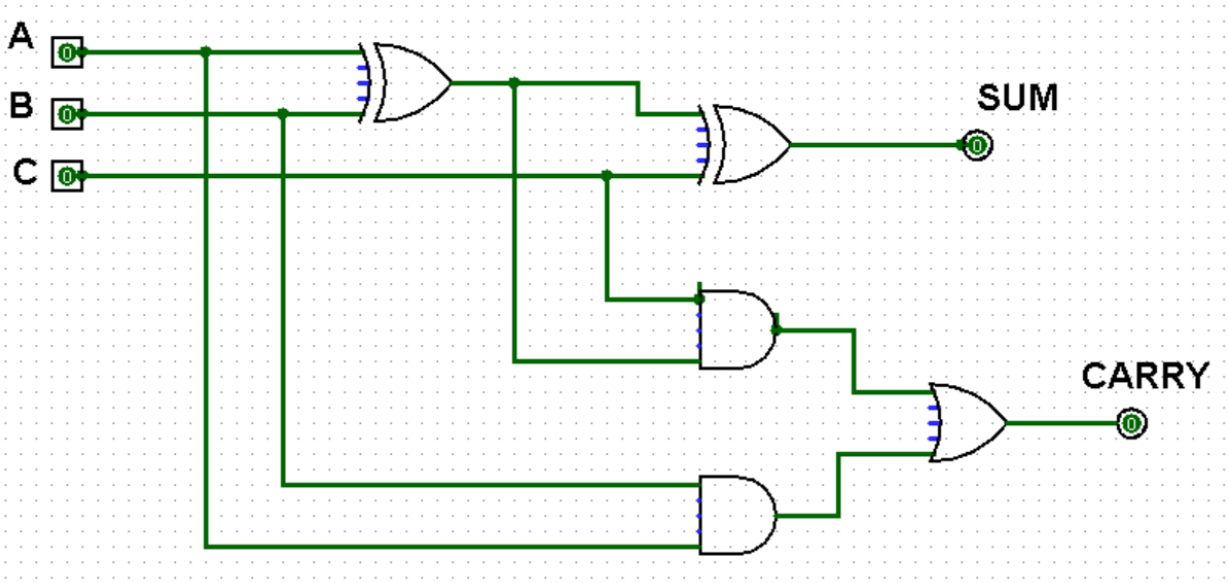
Shorya Sharma
19EE01017

PART 1

Aim:
Simulation and Implementation of a Full Adder.

Theory:
Full Adder is the adder which adds three inputs and produces two outputs. The first two inputs are A and B and the third input is an input carry as C-IN. The output carry is designated as C-OUT and the normal output is designated as S which is SUM.

Circuit Diagram:



Truth Table:

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Discussion

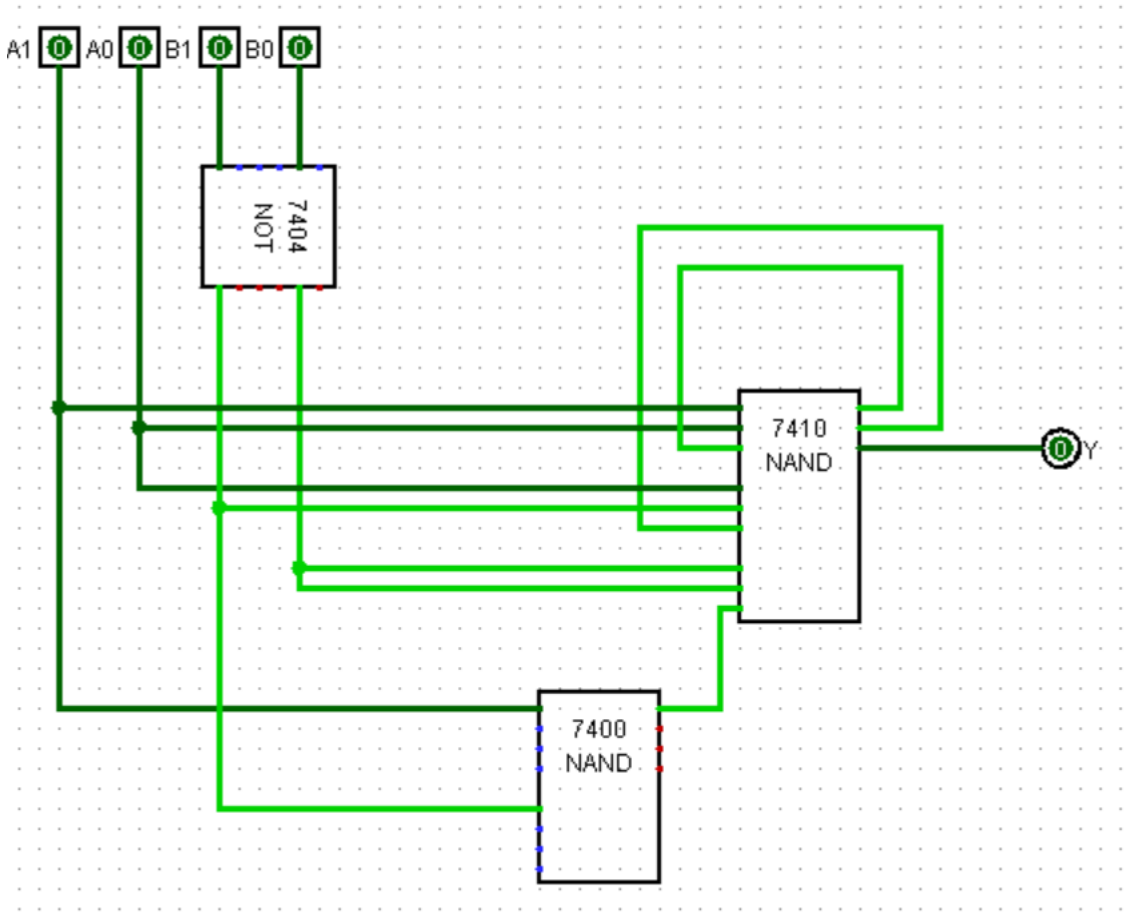
Thus, we implemented and simulated the Full Adder using a 3 input XOR gate (For the S part),3 2 input AND gates and 1 3 input OR gate (For the Cout part); thereby obtaining the desired Truth Table.

PART 2

Aim:

To design a 2-bit adder using IC Models 7404,7400 and 7410.

Circuit Diagram:



Expression obtained at Y:

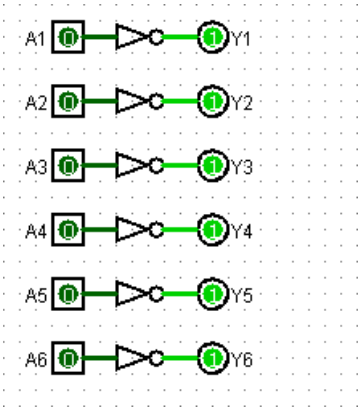
```
A0 ~B1 ~B0 + A1 ~B1 + A1 A0 ~B0
```

Truth Table:

A1	A0	B1	B0	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

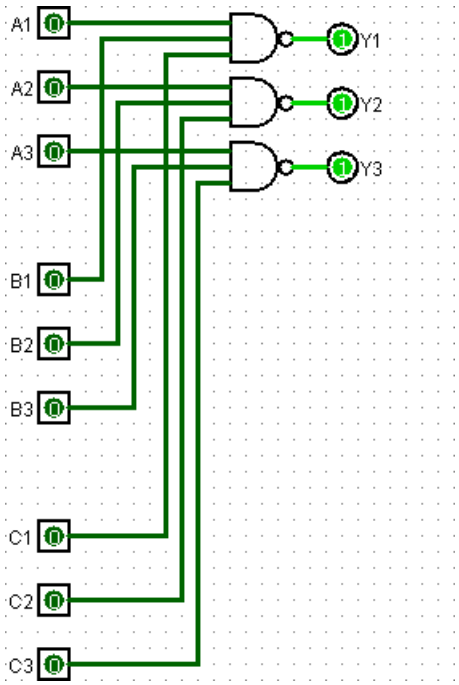
Discussion

The IC Model 7404 consists of 6 input terminals and 6 output terminals corresponding to 6 NOT gates; thereby performing the NOT operation on an output terminal corresponding to its input terminal.

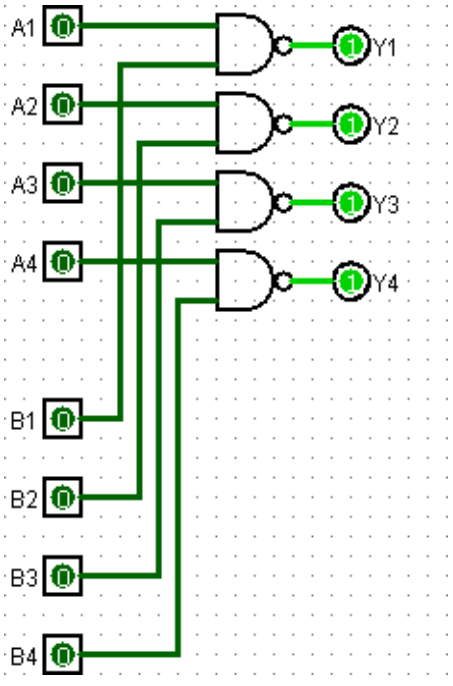


IC 7404

On the other hand 7400 NAND Gate IC Model, consists of two sets of input(A and B consisting of 4 inputs each) and a single set of output terminal(4 net) consisting of an equivalent number of NAND Gate setup in each region. Accordingly, corresponding to the right pin number, they give the desired outputs. The 7410 NAND Gate IC too works in a similar manner; with 3 3 input NAND gates corresponding to 9 inputs and 3 outputs.



IC 7410



IC 7400

PART 3

Aim:
Full Adder Implementation using 8:1 Multiplexor.

Theory:

Multiplexer is a combinational circuit that has maximum of 2^n data inputs, 'n' selection lines and single output line. One of these data inputs will be connected to the output based on the values of selection lines.

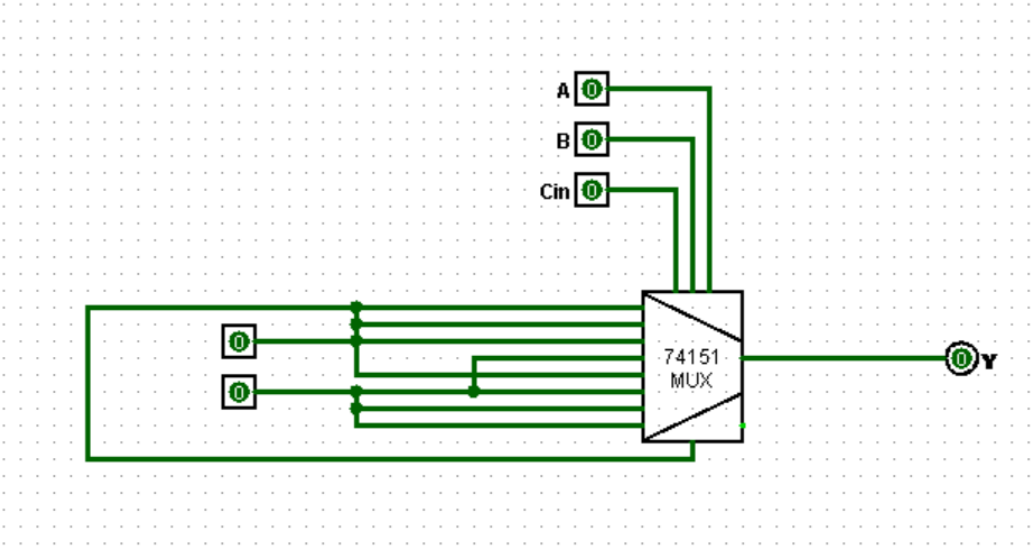
Since there are 'n' selection lines, there will be 2^n possible combinations of zeros and ones. So, each combination will select only one data input. Multiplexer is also called as Mux. the overall combination of two 4x1 Multiplexers and one 2x1 Multiplexer performs as one 8x1 Multiplexer.

An 8:1 multiplexor is literally a "truth table in the form of a circuit". It has eight data inputs (D0..D7), and one output Y. Based on the value of the select inputs (S2S1S0),any one of the data lines get connected to the output.

Truth Table Desired:

A	B	C_{in}	C_{out}
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Circuit Diagram:

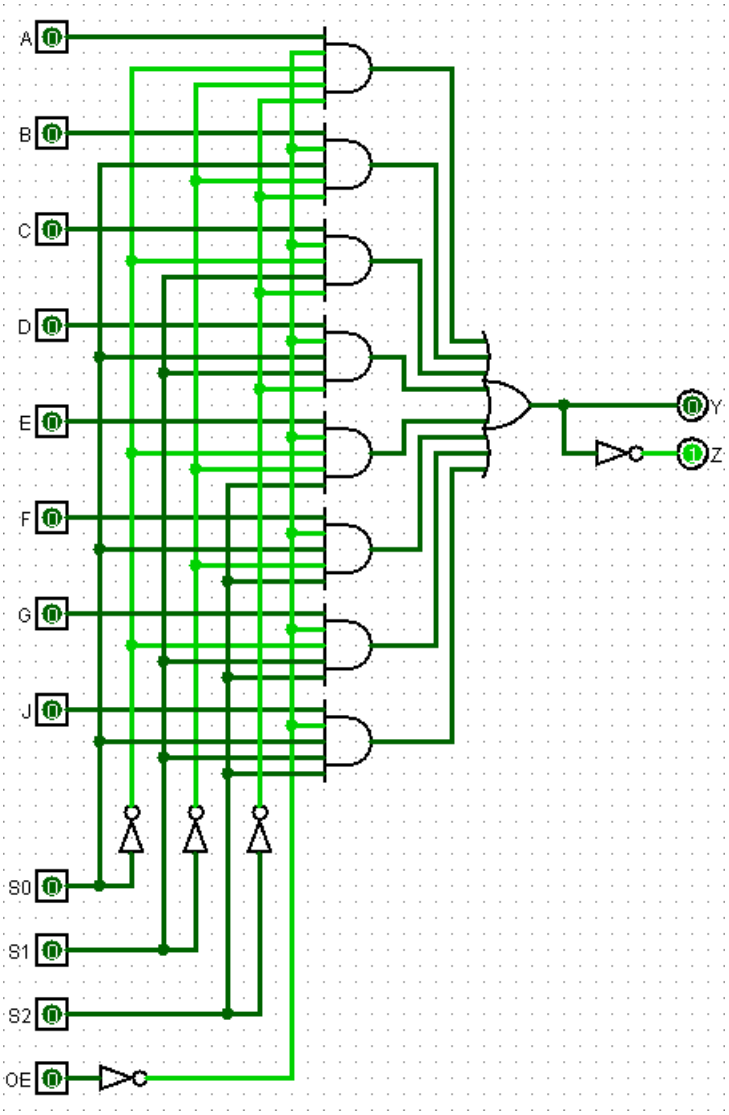


Truth Table obtained:

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

Discussion

The 74151 Model Multiplexer has the interior design as follows:



With 3 inputs,S0,S1 and S2,the multiplexer takes a net input of 2^3=8 possible signal combinations to give the desired output. Such a 3 variable function can be easily implemented using a 8:1 multiplexer by connecting 3 inputs to the select lines S0,S1 and S2;and connecting the rest of the input pins to 0 or 1 as per the requirement of the desired output. So basically, it separates a set of digital inputs referred to the select lines.

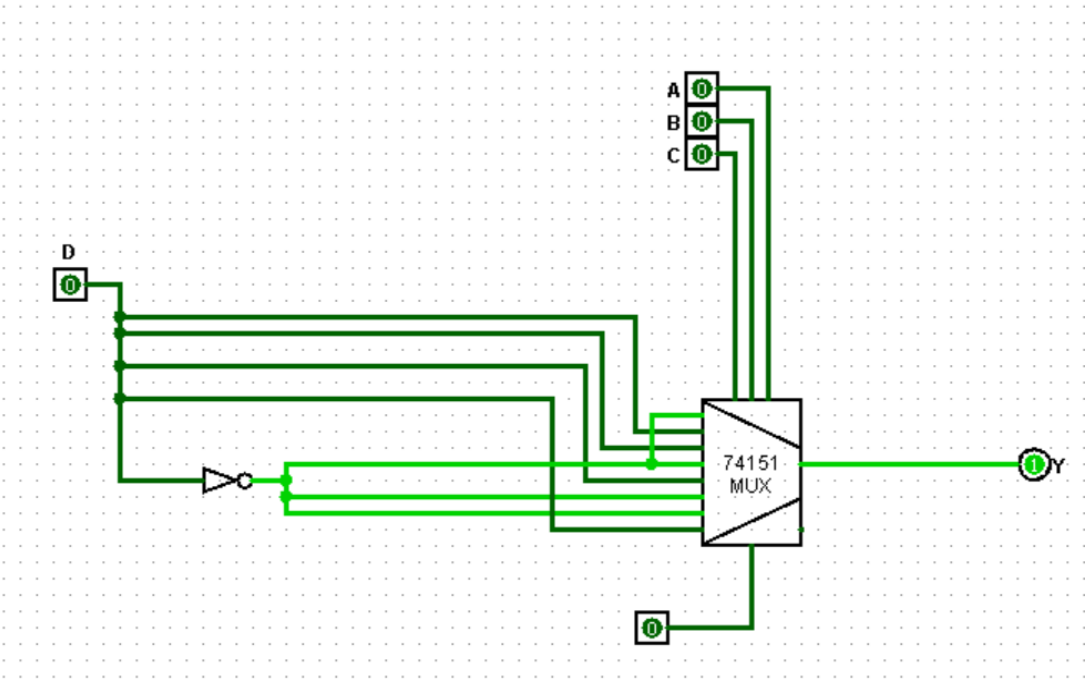
From the given set of outputs as per the inputs; it is clearly visible that Cout is: Cout=A + (B & Cin) under the given set of instructions and setup of circuit.

PART 4

Aim:

To design a 4 variable Logic function using an 8:1 multiplexor.

Screenshot of Circuit:



Expected Truth Table:

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Prepared Truth Table:

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Discussion

In this case we used 8:1 for four bits using complementary and standard connections of D instead of using the 16:1 connection; that allowed us to relate our output to more seemingly one of the input.

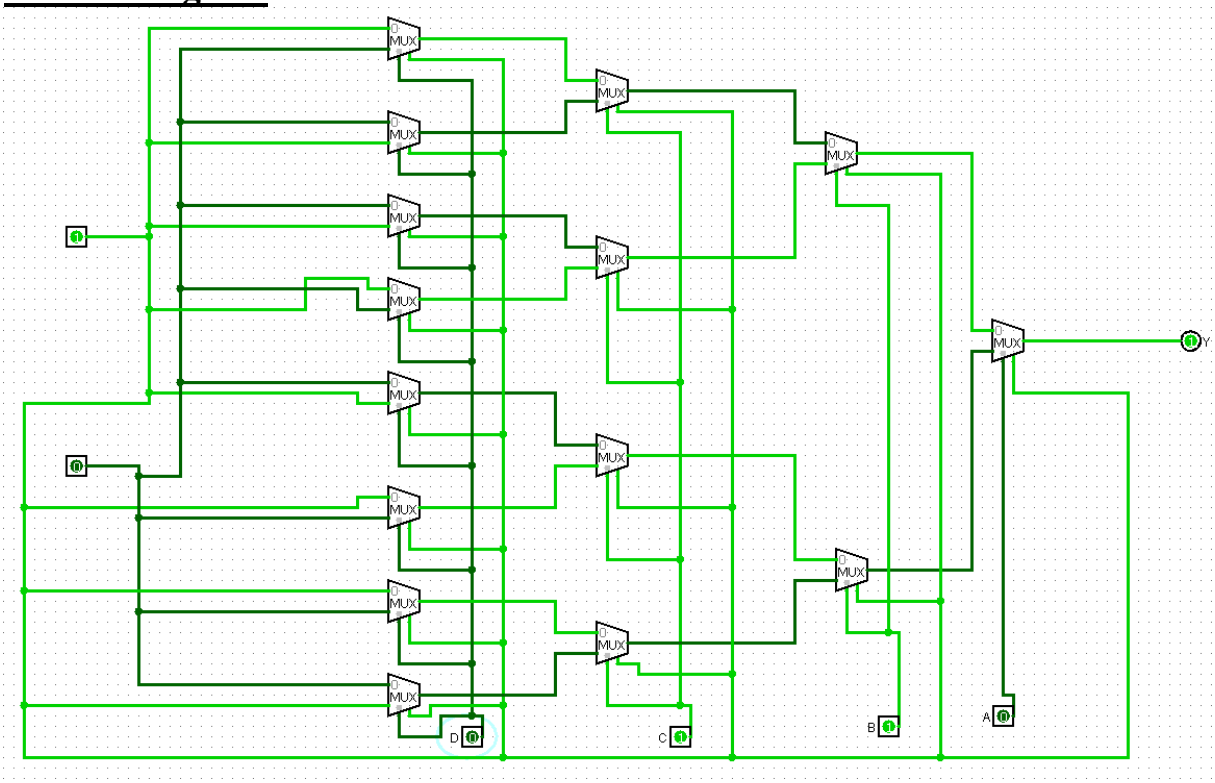
To simplify in other words; **only the input of D** seems to have an effect on Y where if we sequentially start from the first **pair** of Y values as D0,D1 and so on; we get

D0=~D,D1=D,D2=D,D3=~D,D4=D,D5=~D,D6=~D,D7=D;

HOMEWORK

Aim:
Implementing 4 variable function using 2:1 multiplexor.

Circuit Diagram:



Truth Table:

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Discussion

Basically, using the 2:1 multiplexor, what is done here basically the input terminals are reversed for every consecutive multiplexor in a pair thereby making the output entirely dependent on D similar to what we observed in case 3 where every pair output is entire dependently standardly or inversely on the value of D.

Conclusion

So finally, we implemented all variety types of adder operations using both basic gates and ICs; did poly variable functions using variety type of multiplexers and did some basic operations too. It was a wonderful experience simulating and logging the desired truth tables; studying through some basic IC models like 7400,7402,7410 that are used to perform basic operation like NOT, NAND etc.

----------X---------