

RV COLLEGE OF ENGINEERING®
(Autonomous Institution affiliated to VTU, Belagavi)
Department of Electronics and Instrumentation Engineering



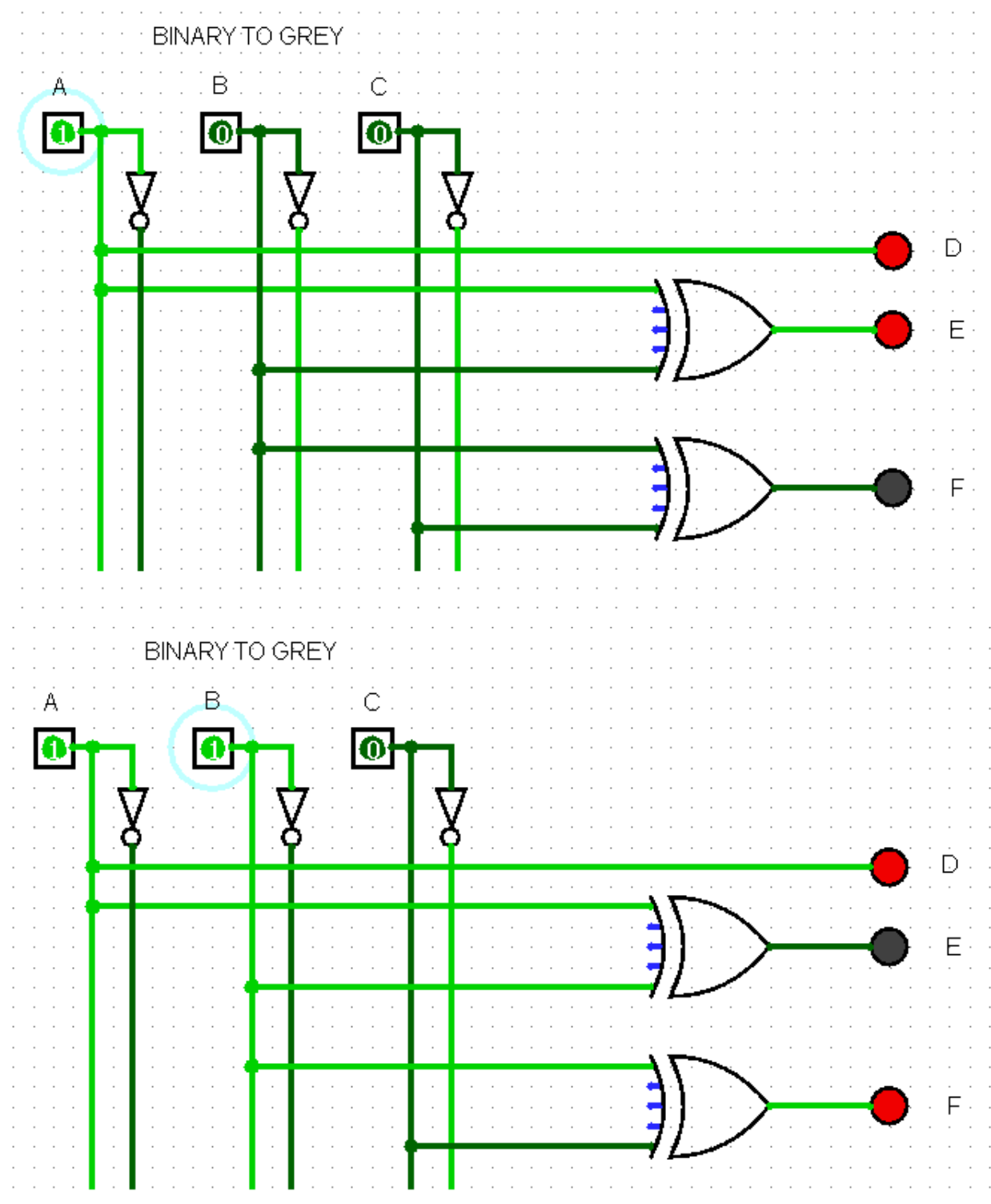
ADDC LOGISIM SIMULATION LAB
III SEMESTER
(2020-2021)

BACHELOR OF ENGINEERING
In
Electronics and Instrumentation Engineering
Submitted by

FARHAN AKHTER
1RV19EI017

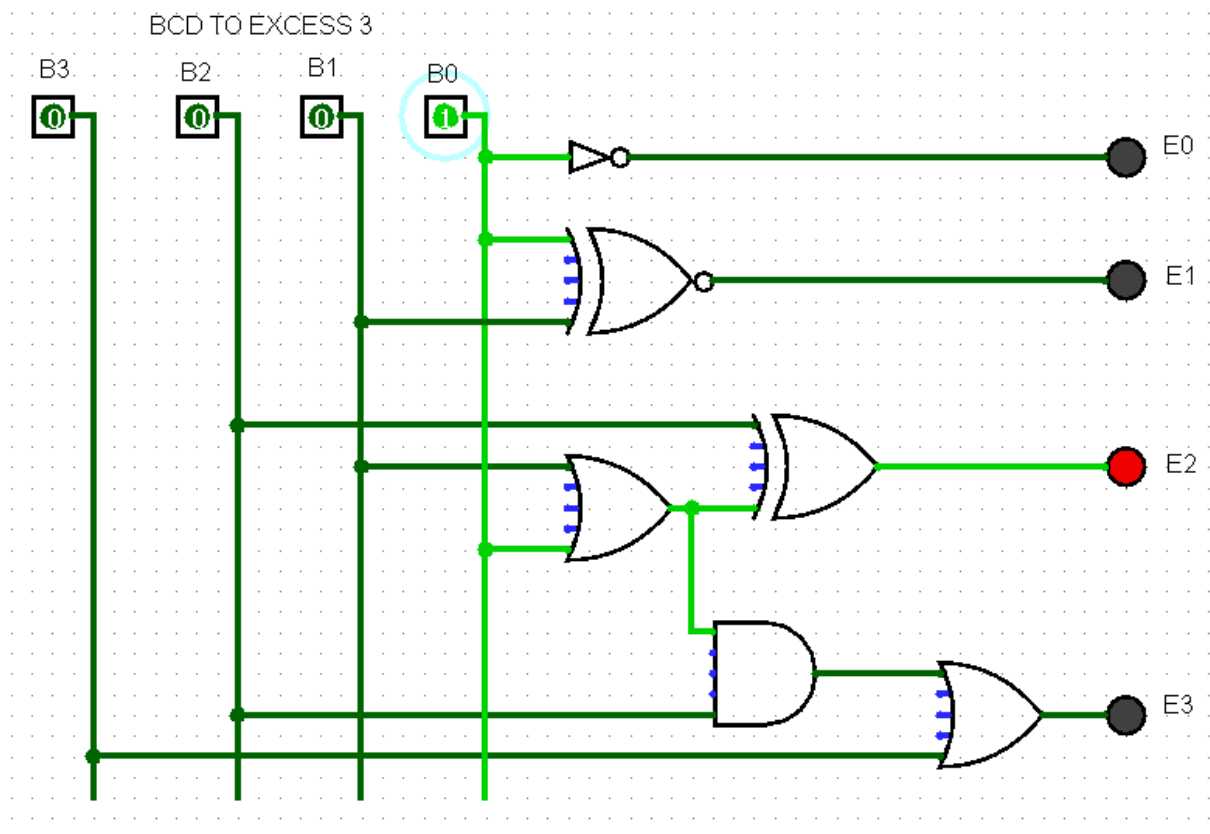
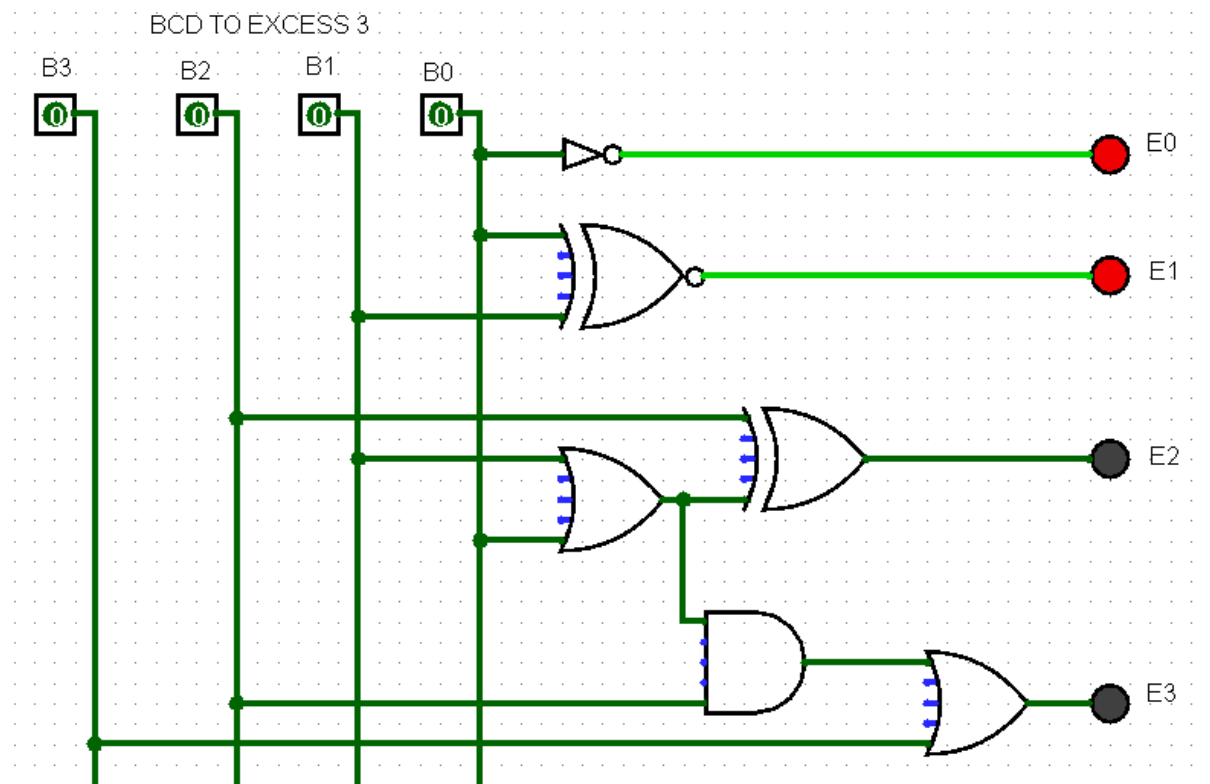
Aim: To Design Code converters a) Binary to gray b) BCD to Excess 3

1.



Aim: To Design Code converters a) Binary to gray b) BCD to Excess 3

2.



Aim: To Design Code converters a) Binary to gray b) BCD to Excess 3

Observation:

Exercise 1:

1) Binary to Gray:

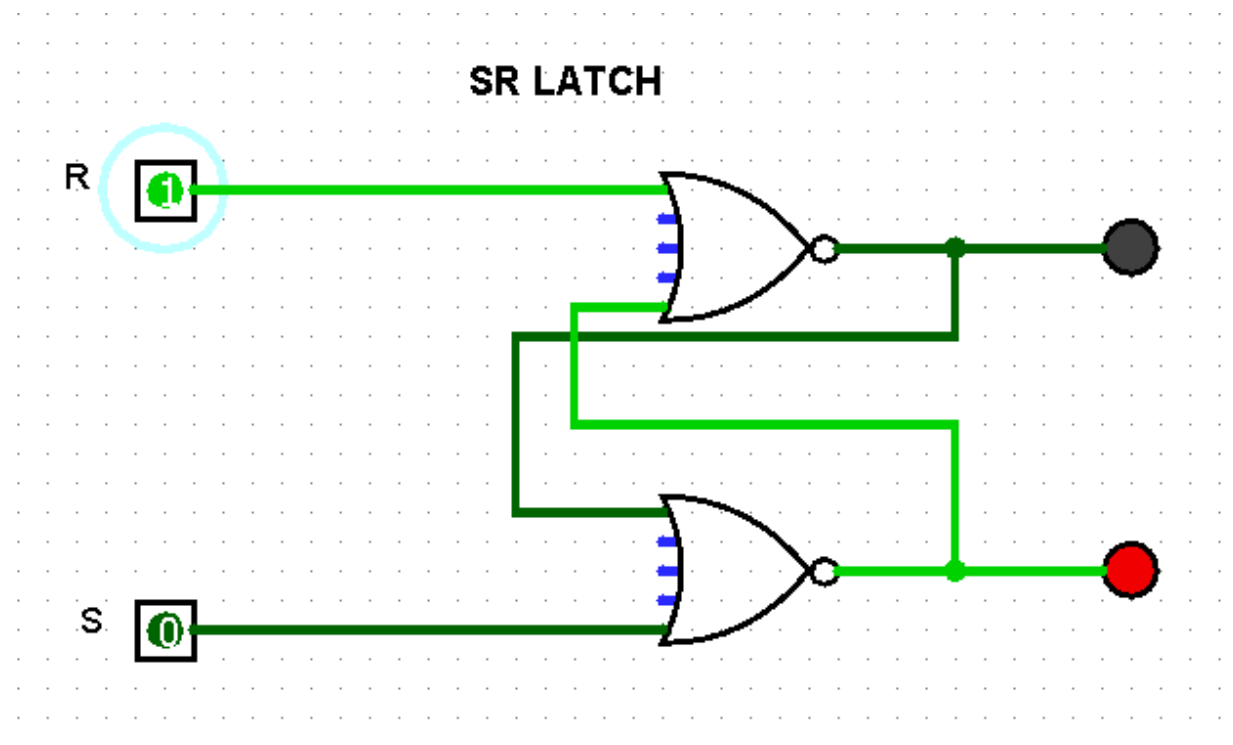
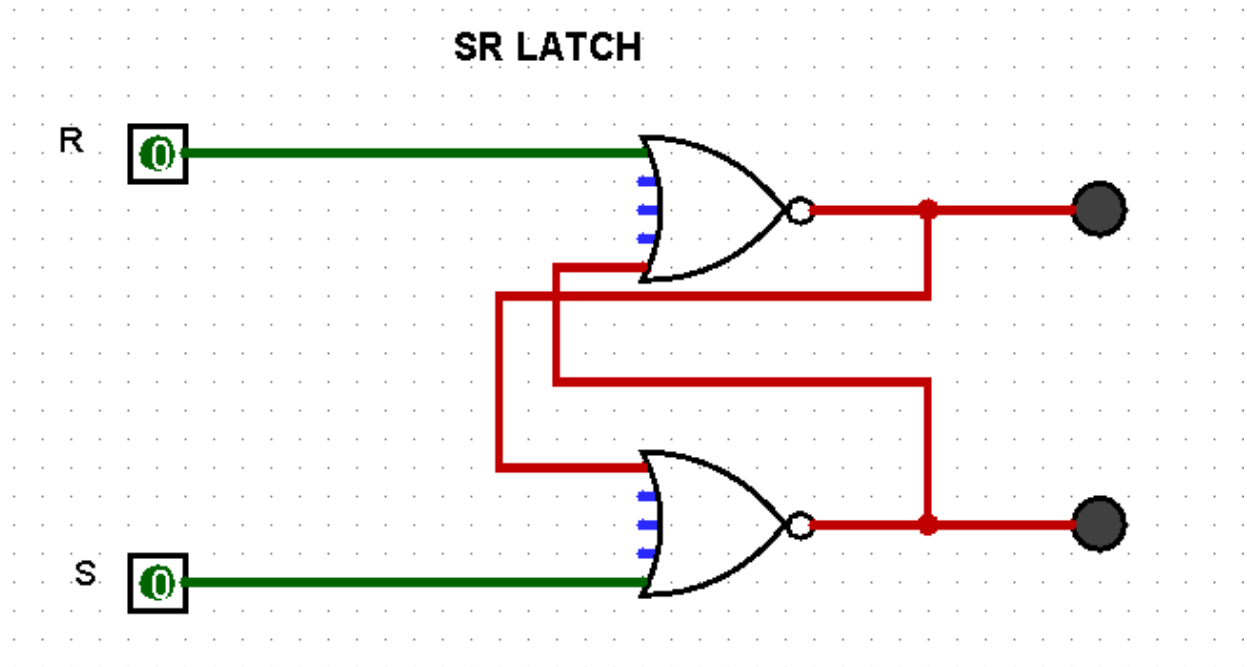
<u>Decimal</u>	<u>Binary</u>	<u>Gray</u>
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100

② Binary to Excess 3

<u>Binary</u>	<u>Excess 3</u>
0000	0011
0001	0100
0010	0101
0011	0110
0100	0111
0101	1000
0110	1001
0111	1010
1000	1011
1001	1100

Aim: To construct various latches and flip flops and simulate them

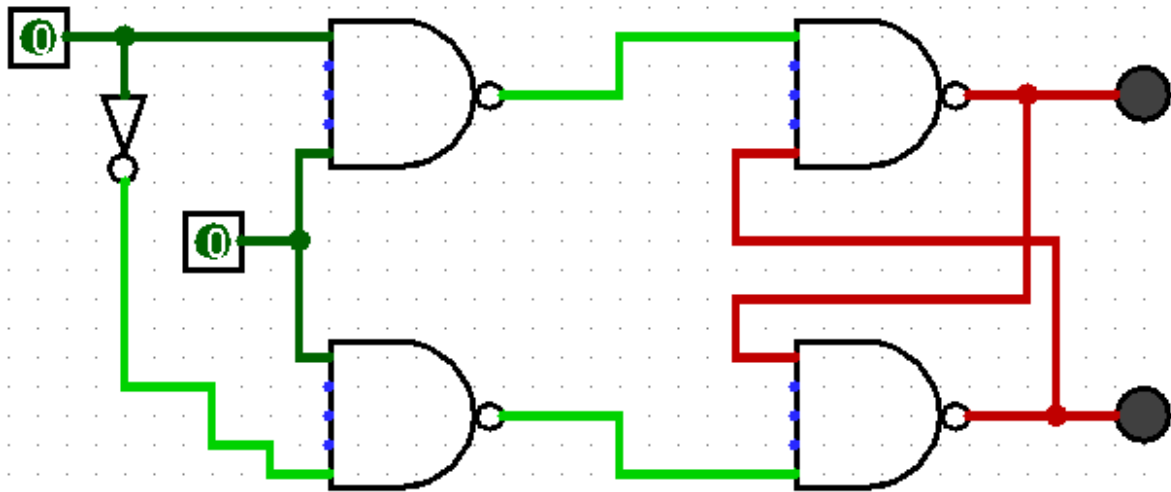
1.



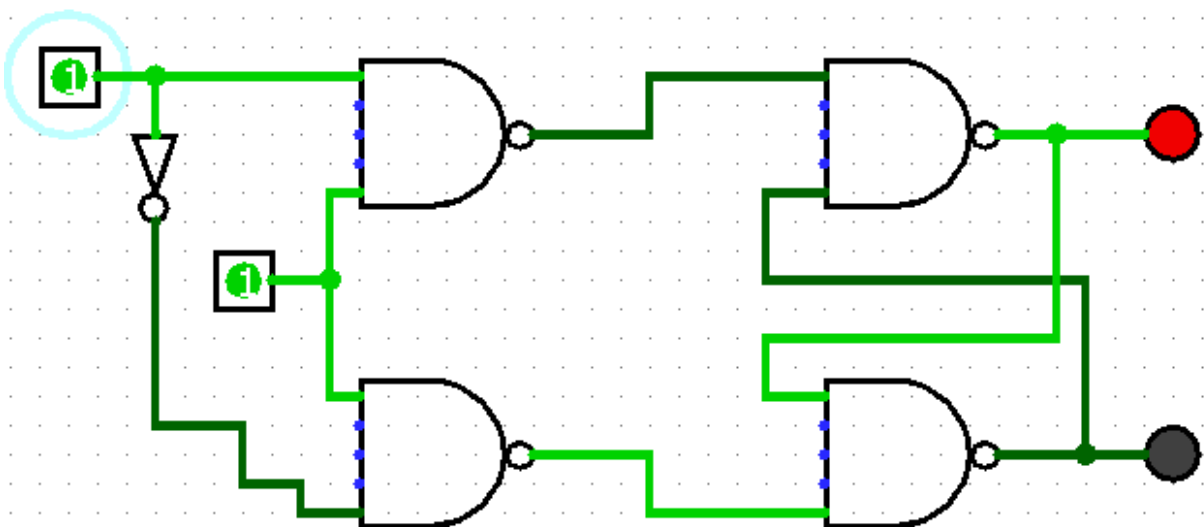
Aim: To construct various latches and flip flops and simulate them

2.

GATED D LATCH

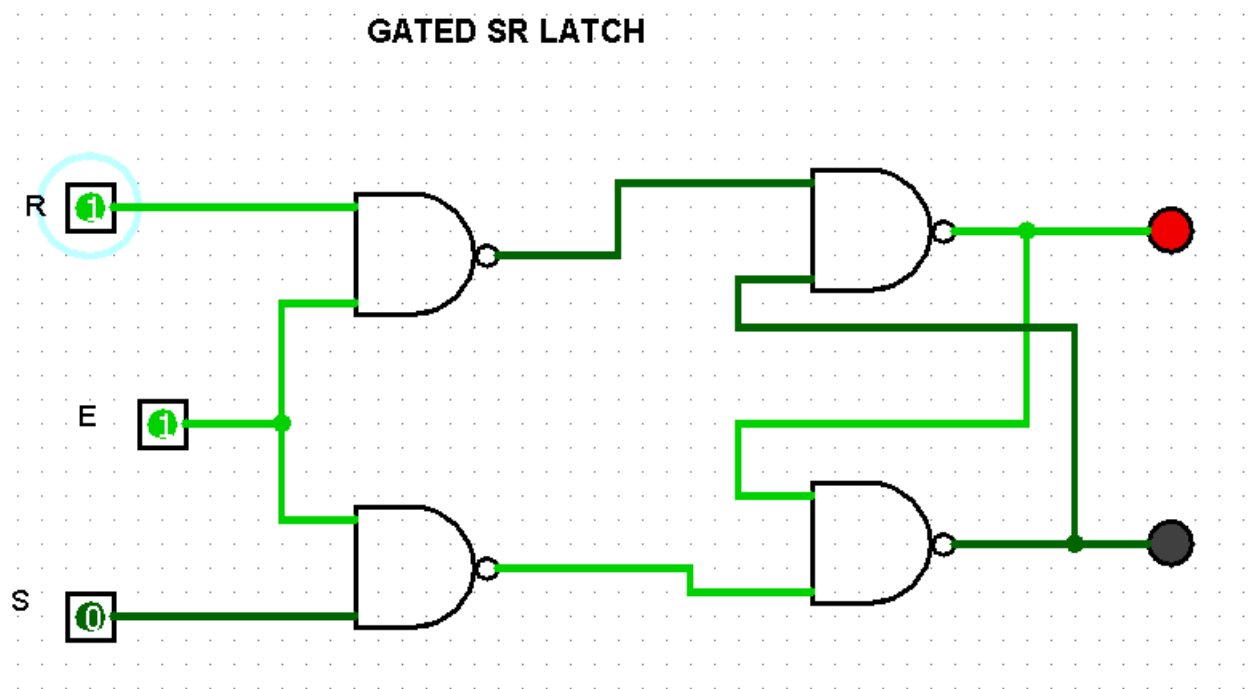
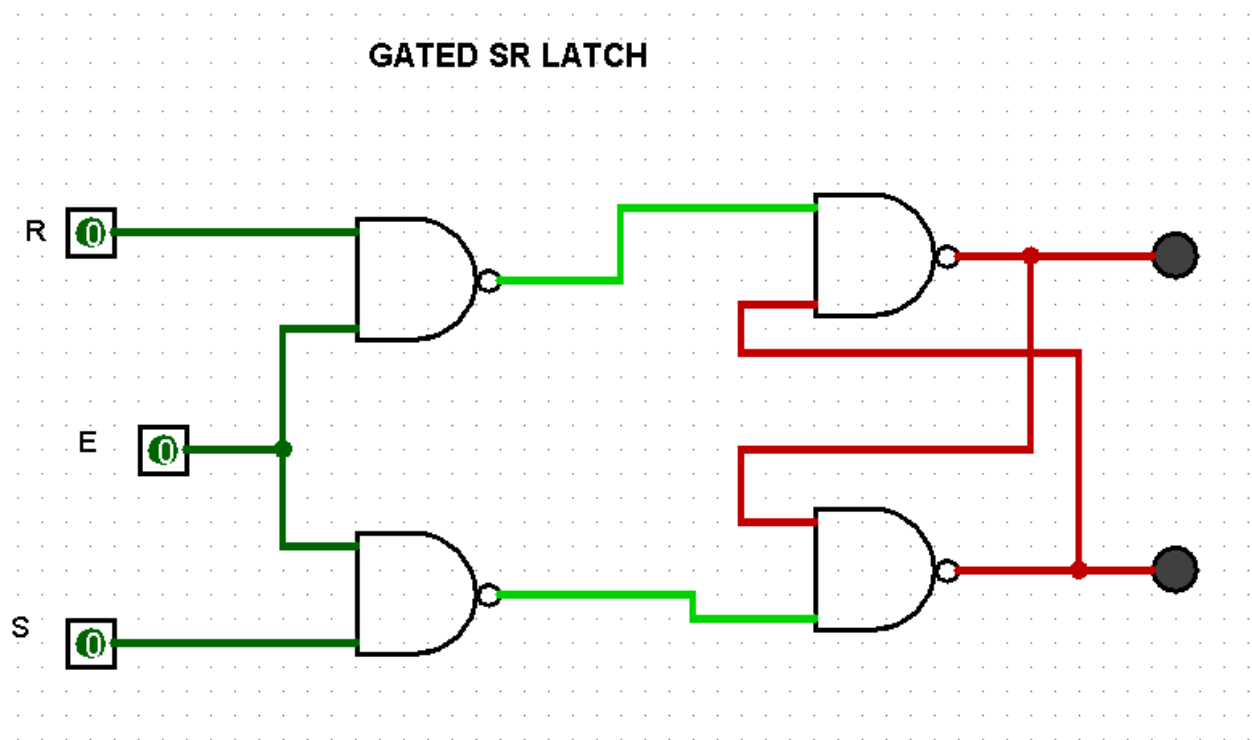


GATED D LATCH



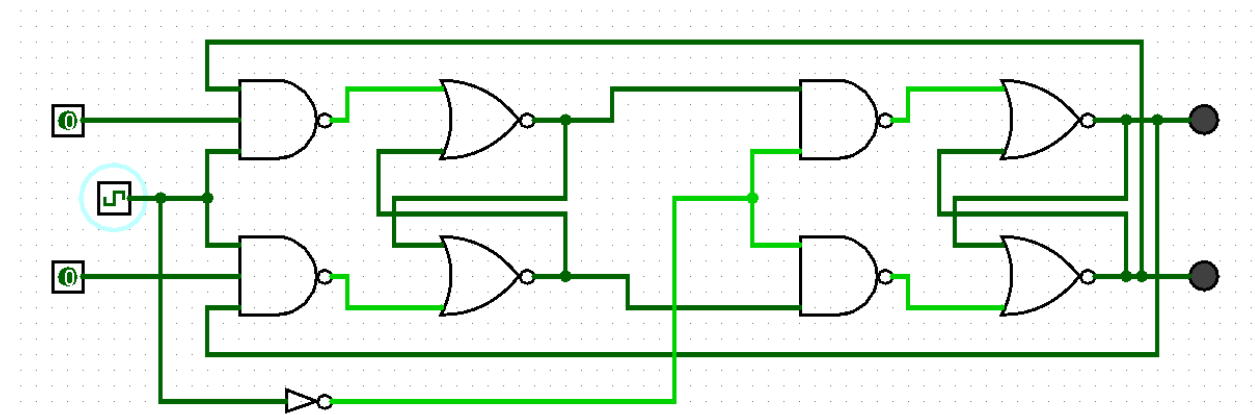
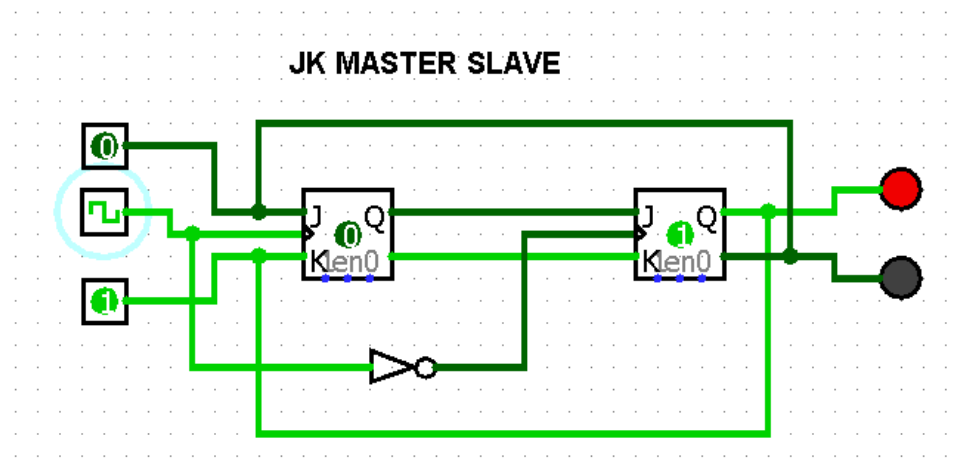
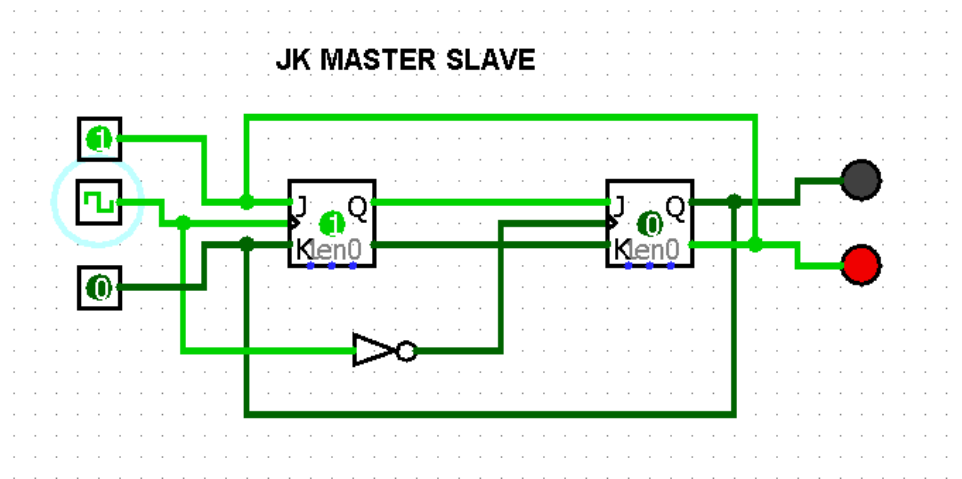
Aim: To construct various latches and flip flops and simulate them

3.



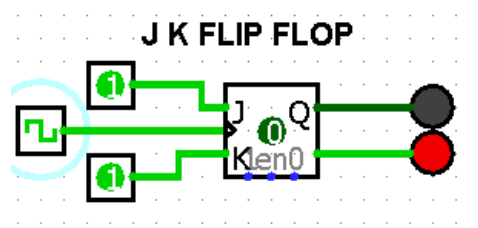
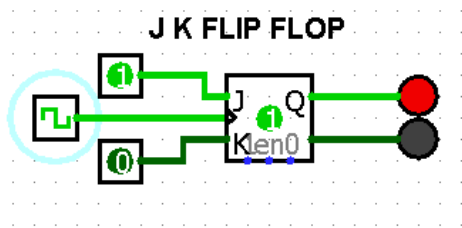
Aim: To construct various latches and flip flops and simulate them

4.

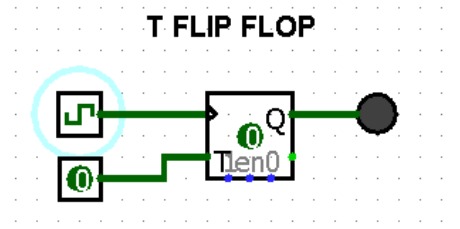
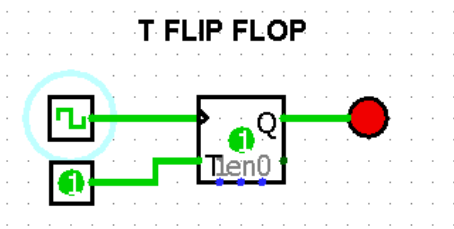


Aim: To construct various latches and flip flops and simulate them

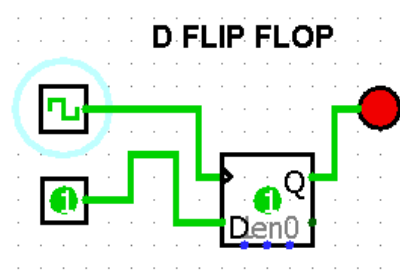
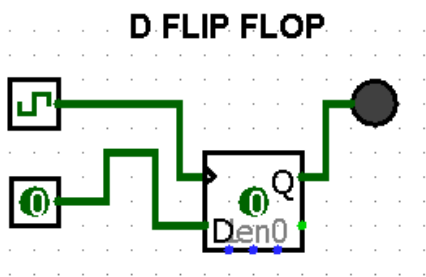
5.



6.



7.



Aim: To construct various latches and flip flops and simulate them

Observation:

Observation :

Truth Table :

S	R	Q	\bar{Q}
0	0	Invalid	
0	1	0	1
1	0	1	0
1	1	0	0

SR Latch

clk	D	Q _{n+1}
0	x	Q _n
1	0	0
1	1	1

D flip flop

clk	J	K	Q _{n+1}
0	x	x	Q _n
1	0	0	Q _n
1	0	1	0
1	1	0	1
1	1	1	\bar{Q}_n

JK ff

T	clk	Q _{n+1}
0	0	Q _n
0	1	Q _n
1	1	\bar{Q}_n

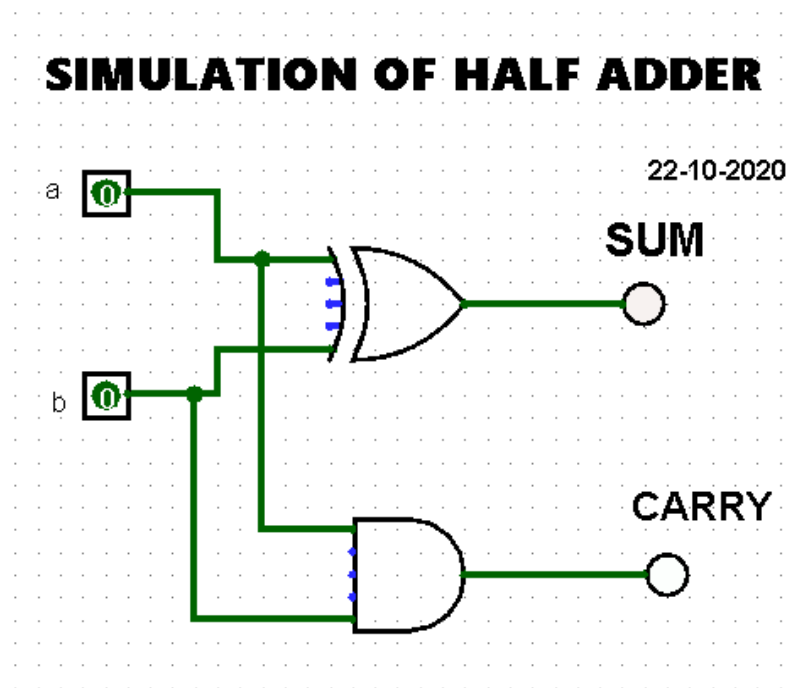
T ff

clk	S	R	Q ⁺	\bar{Q}^+
0	x	x	Q	\bar{Q}
1	0	0	Q	\bar{Q}
1	0	1	0	1
1	1	0	1	0
1	1	1	Invalid	

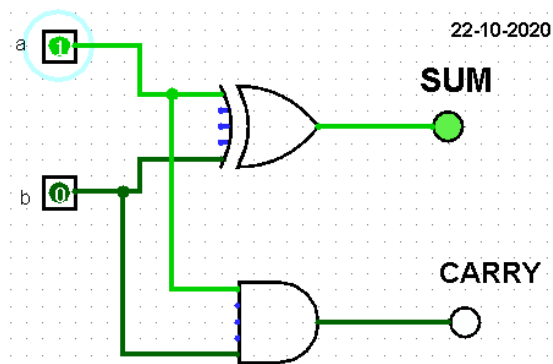
SR FlipFlop

Aim: To construct and simulate FA/FS &HA/HS

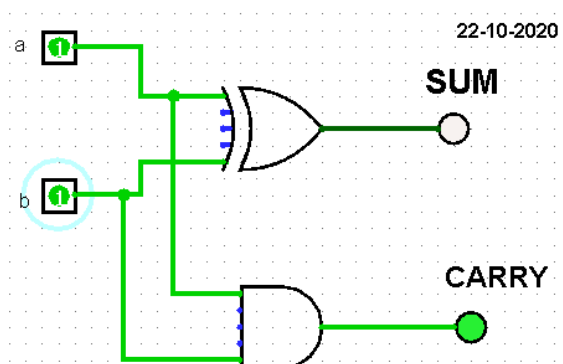
1)



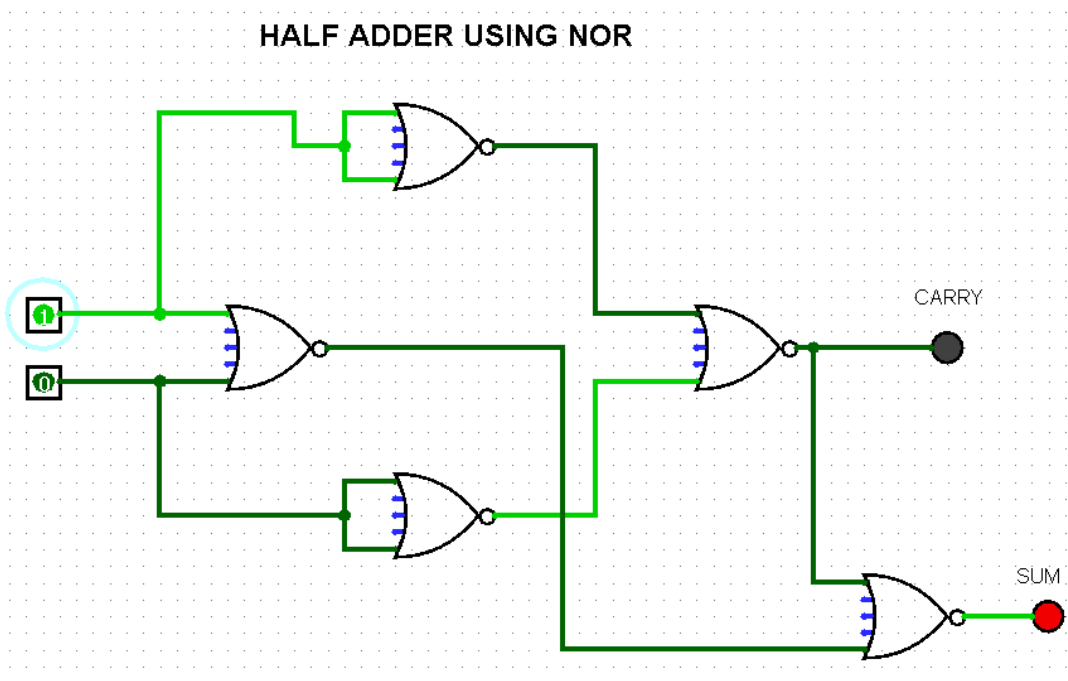
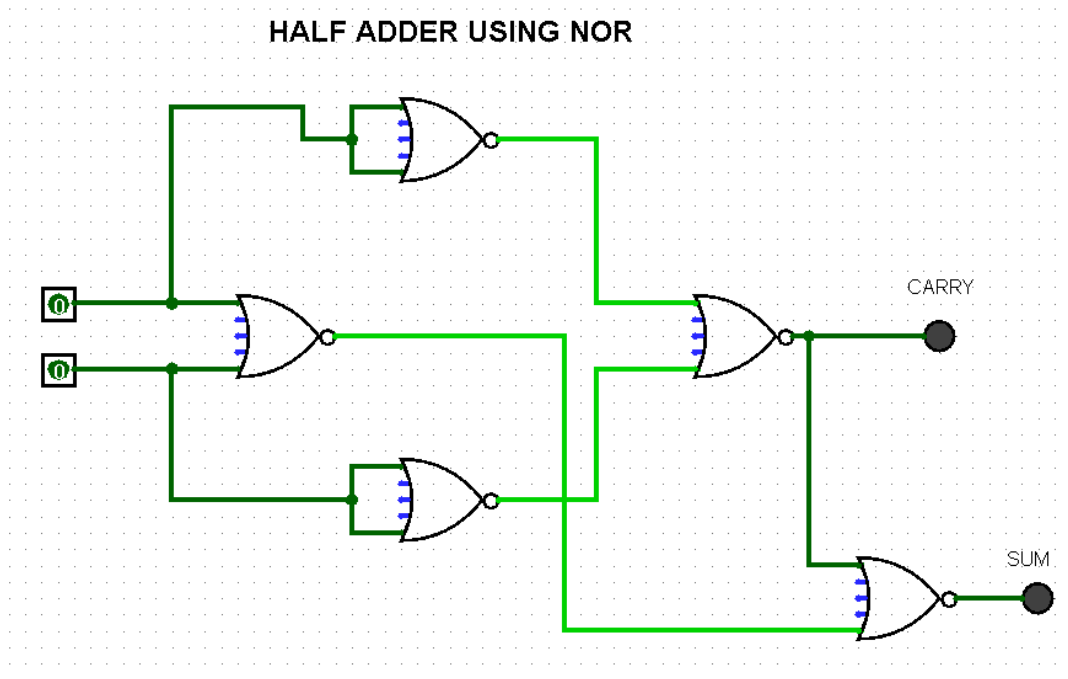
SIMULATION OF HALF ADDER



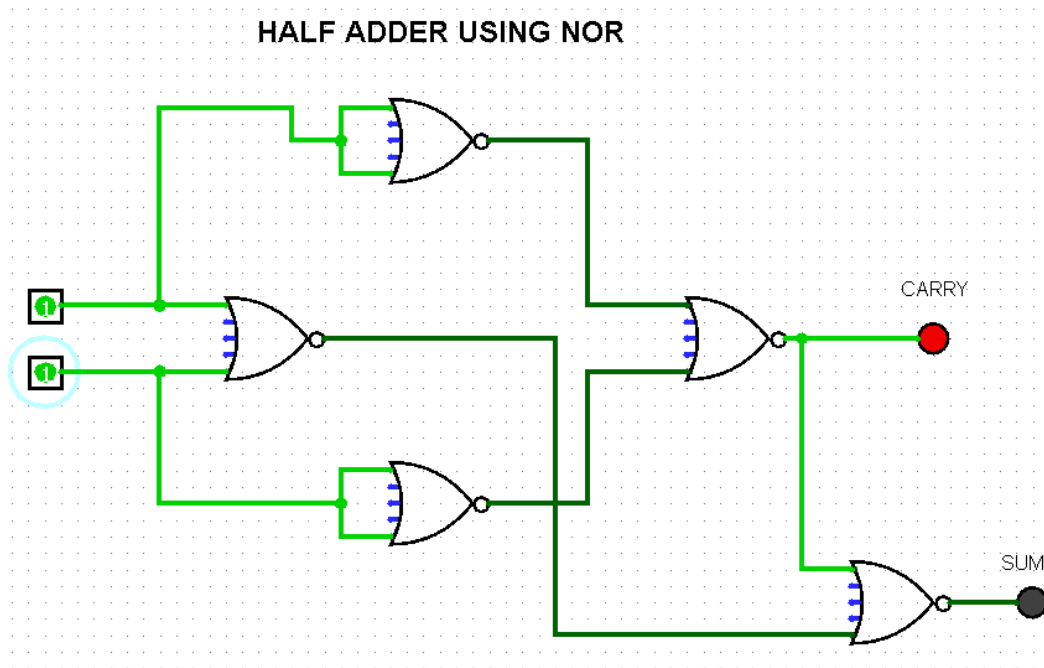
SIMULATION OF HALF ADDER



Aim: To construct and simulate FA/FS &HA/HS

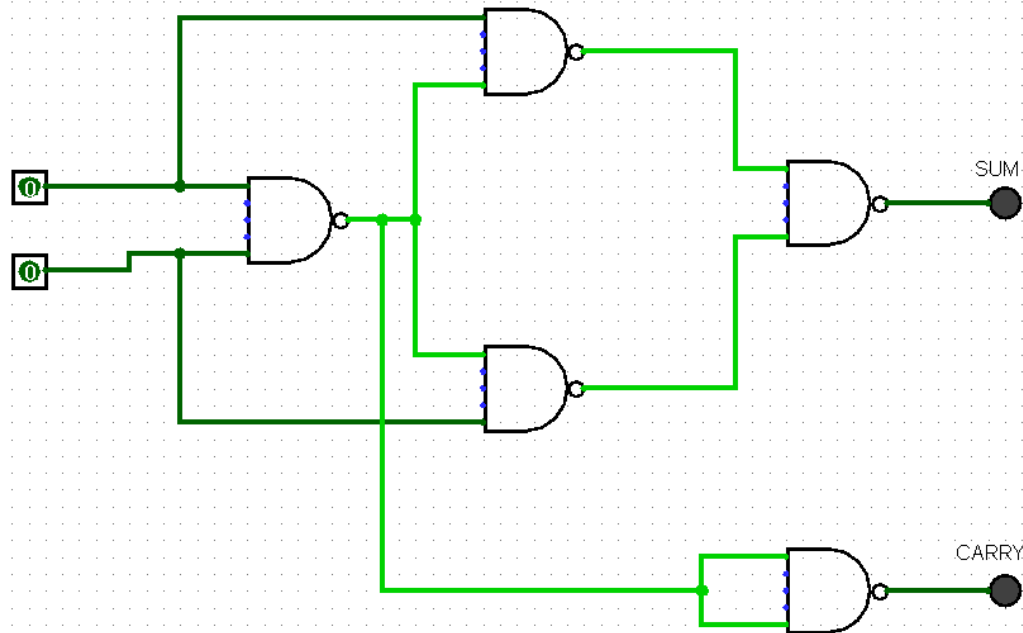


Aim: To construct and simulate FA/FS &HA/HS

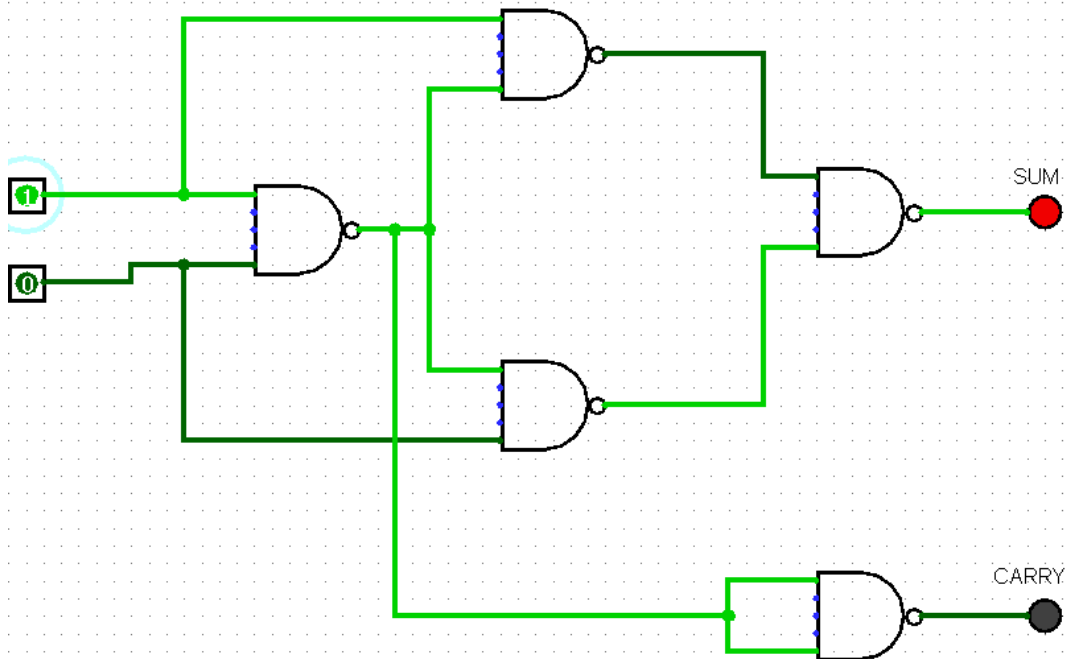


Aim: To construct and simulate FA/FS &HA/HS

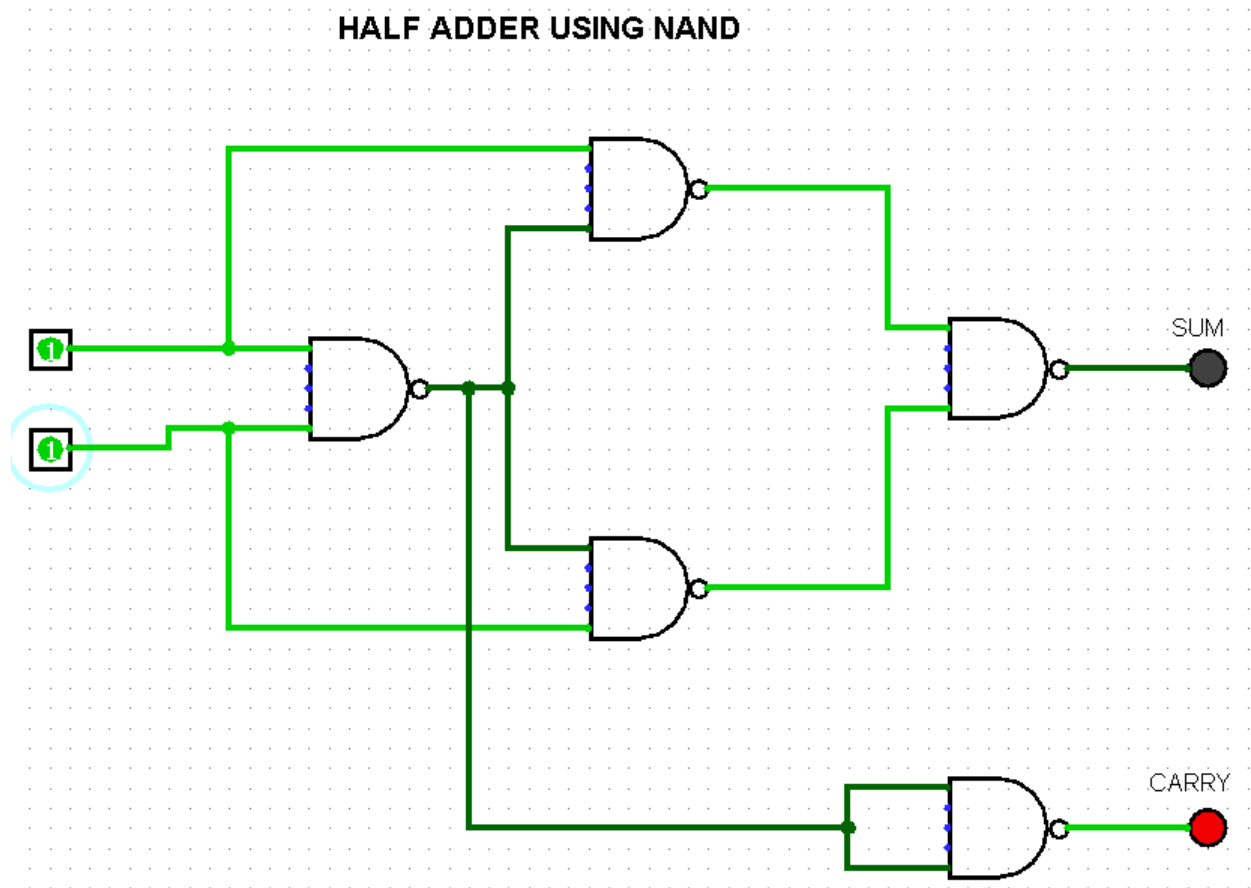
HALF ADDER USING NAND



HALF ADDER USING NAND

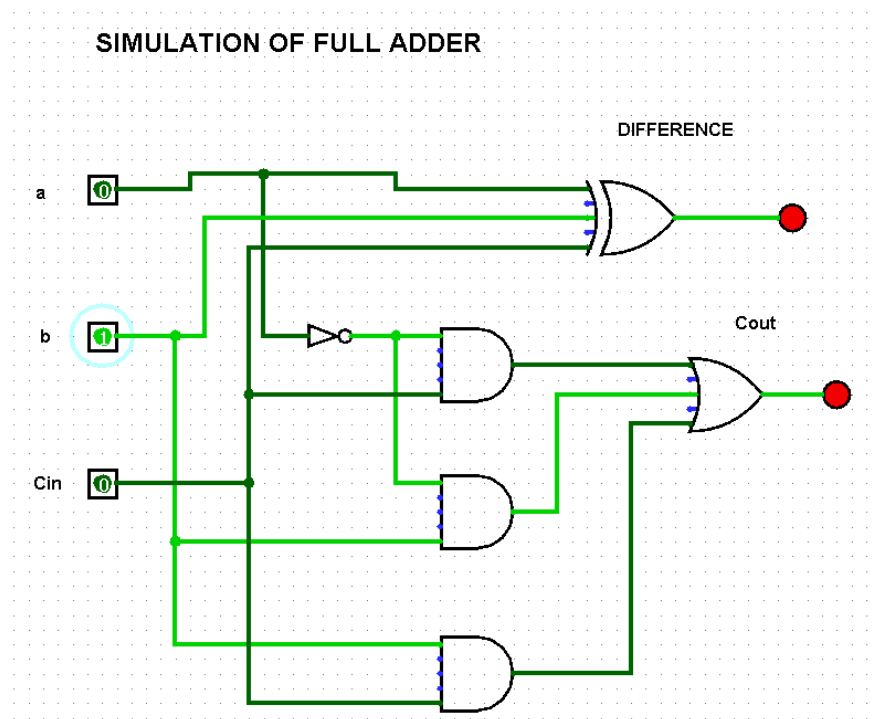
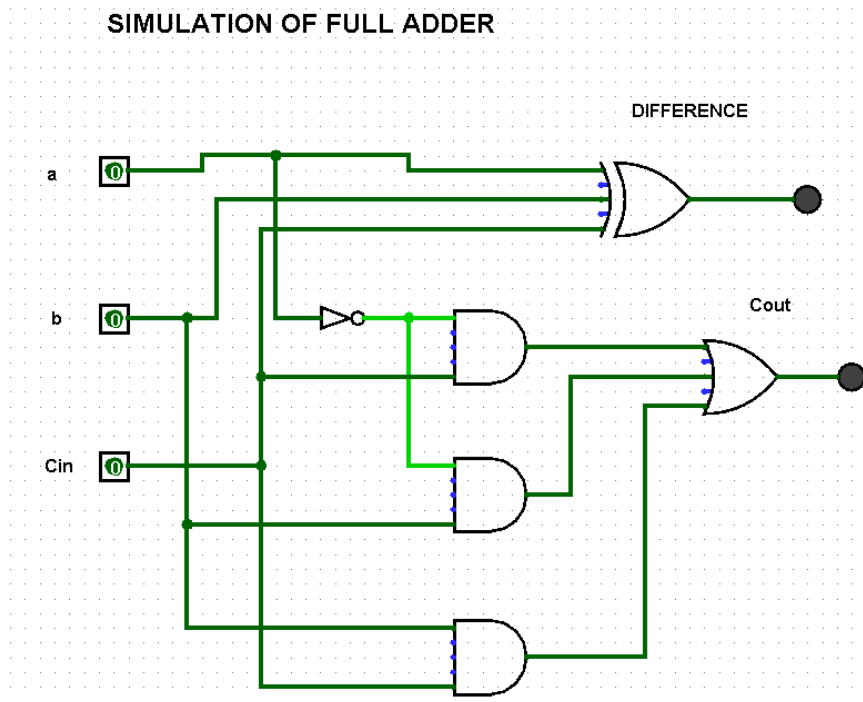


Aim: To construct and simulate FA/FS &HA/HS

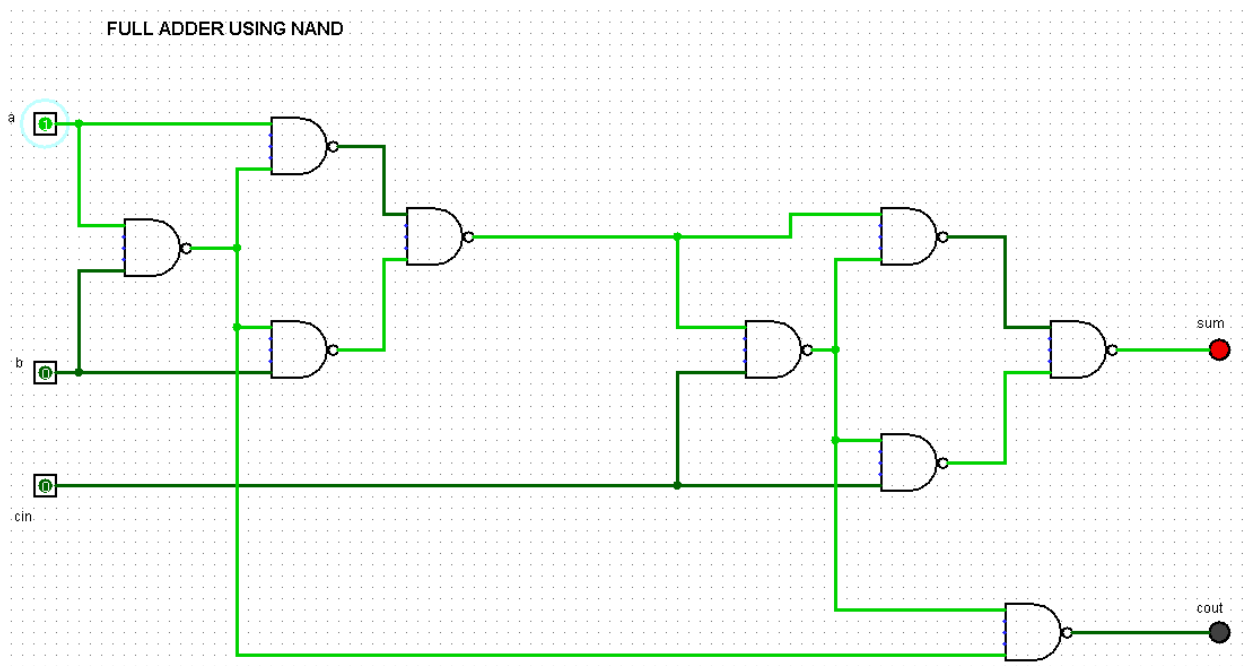
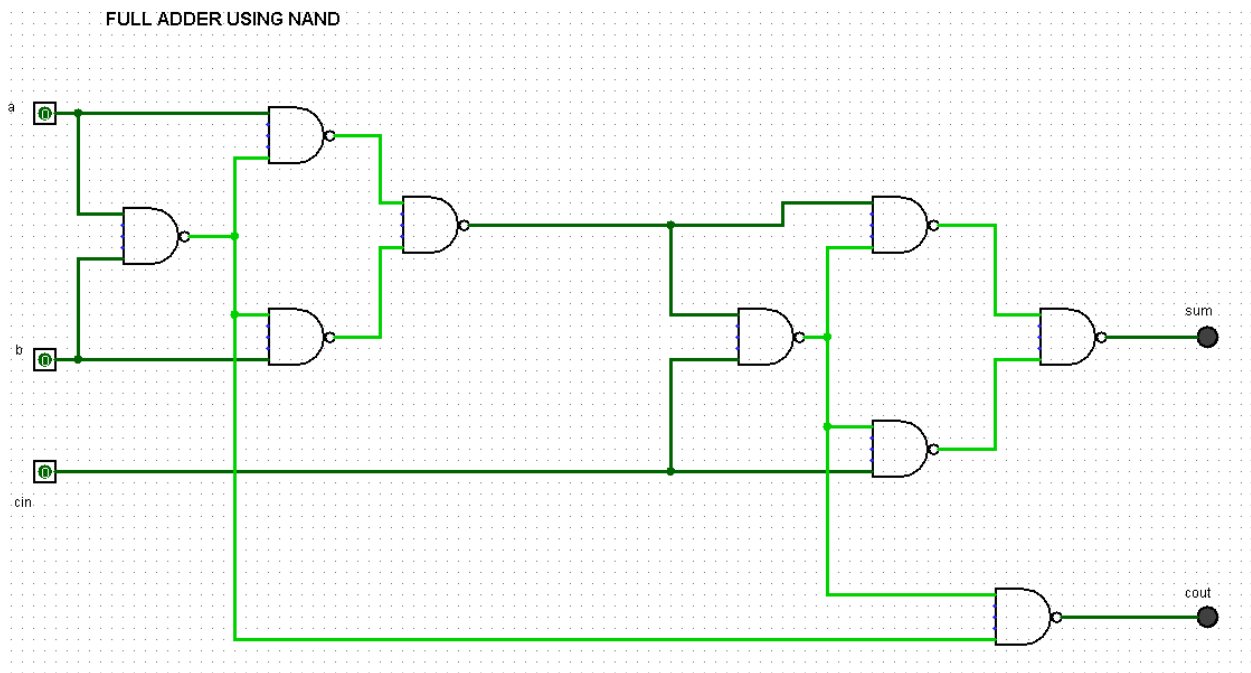


Aim: To construct and simulate FA/FS &HA/HS

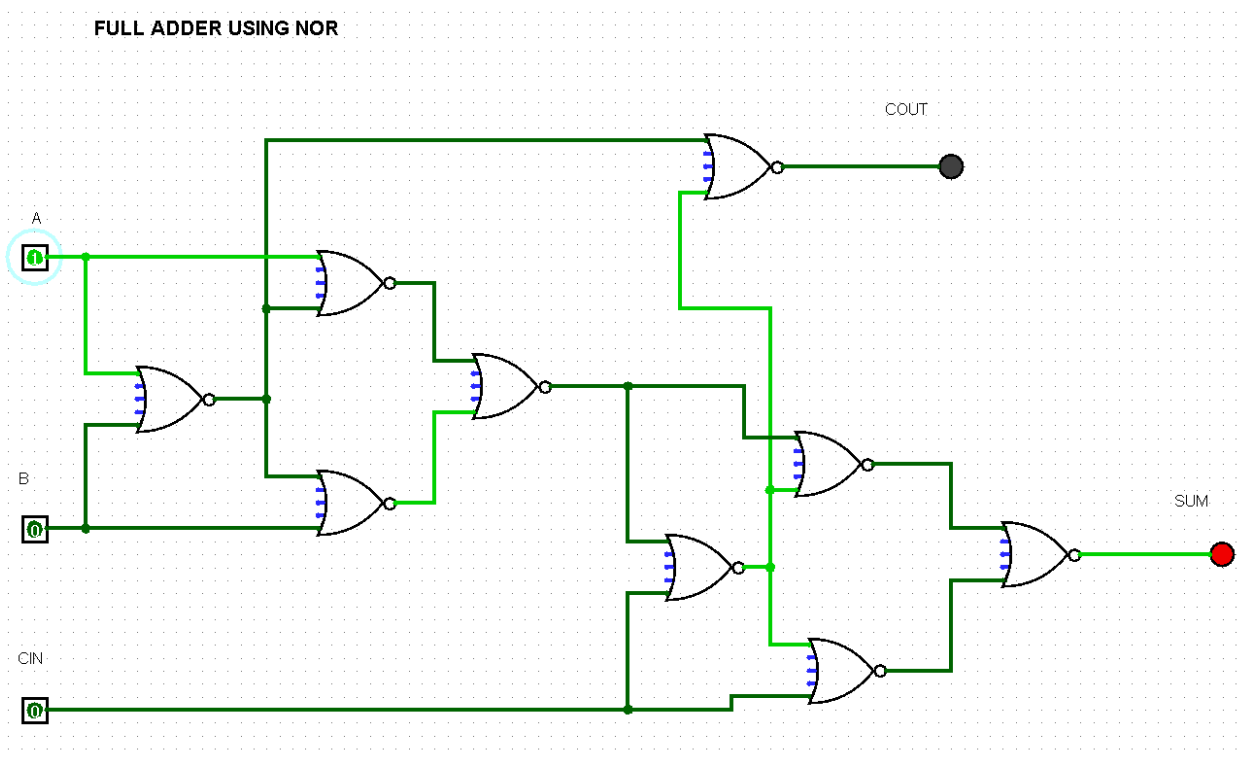
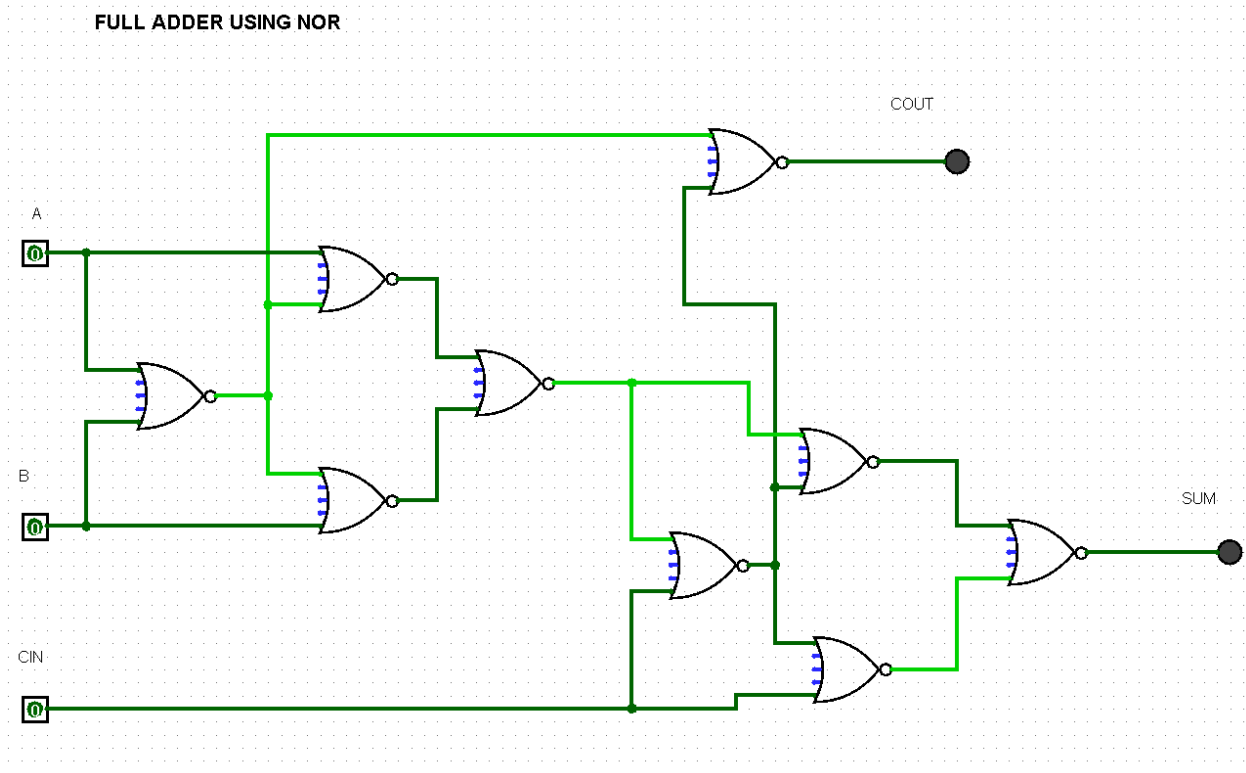
2)



Aim: To construct and simulate FA/FS &HA/HS

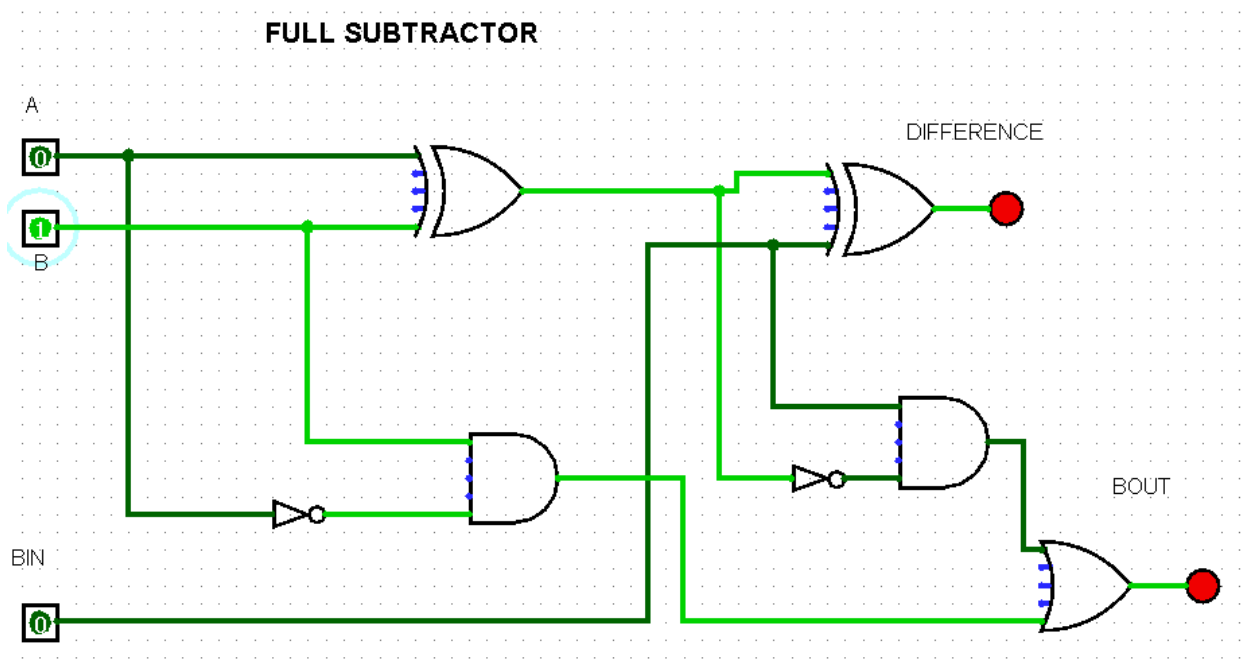
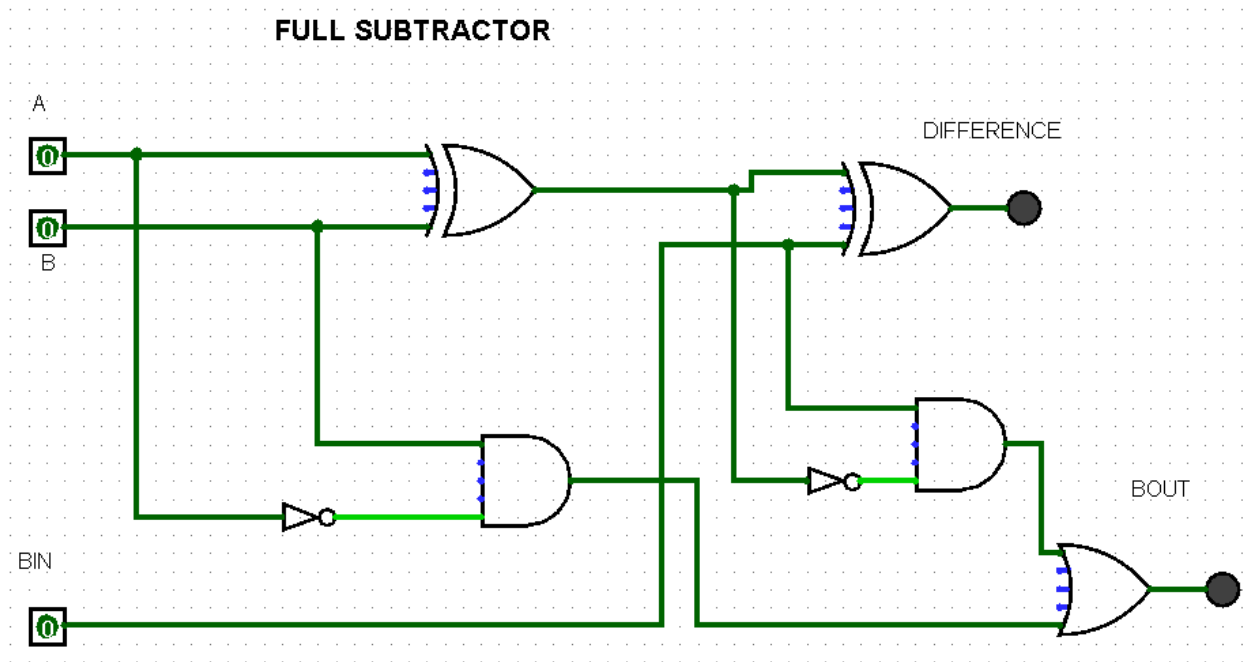


Aim: To construct and simulate FA/FS & HA/HS

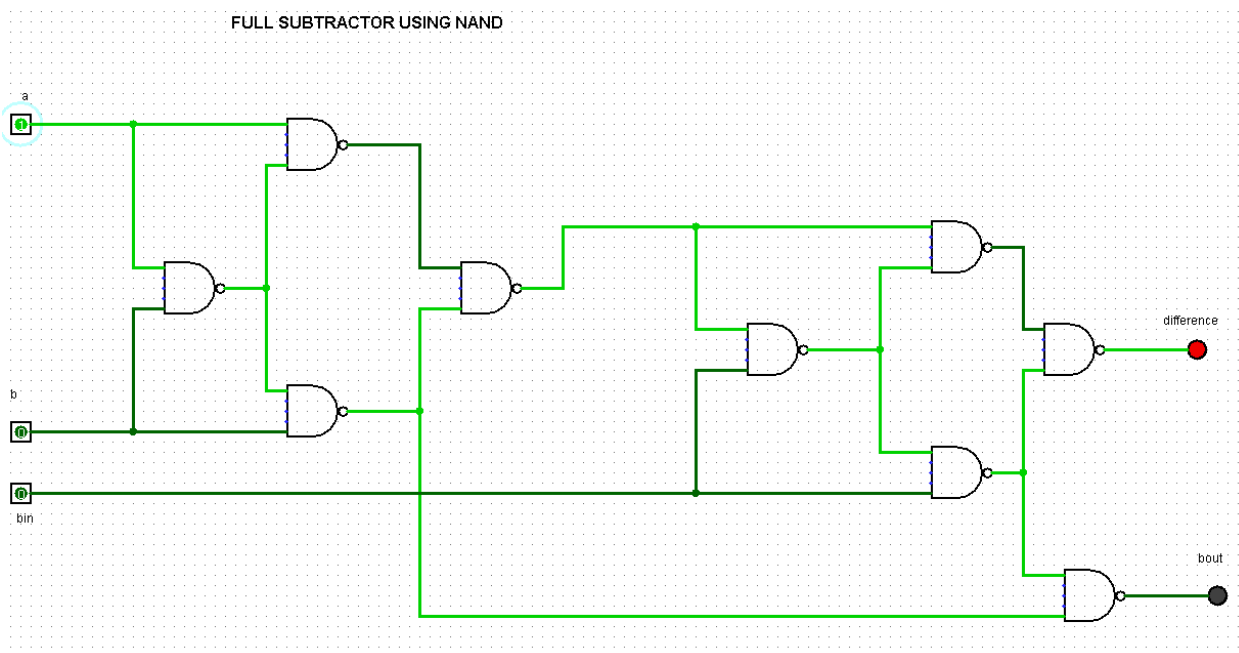
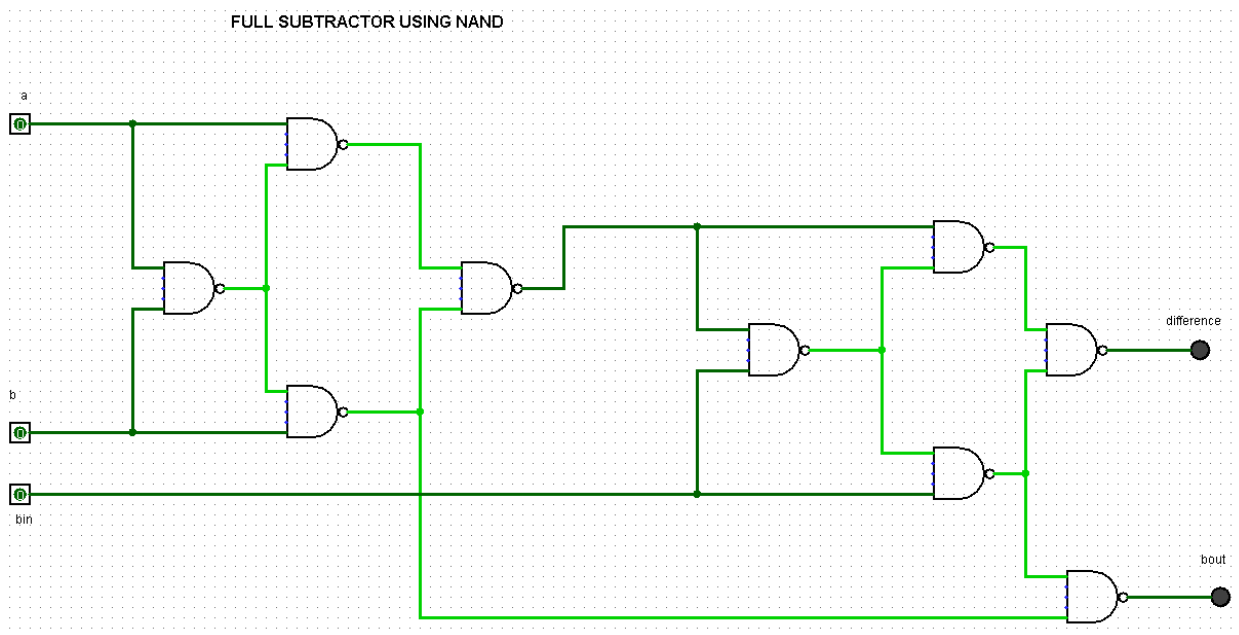


Aim: To construct and simulate FA/FS &HA/HS

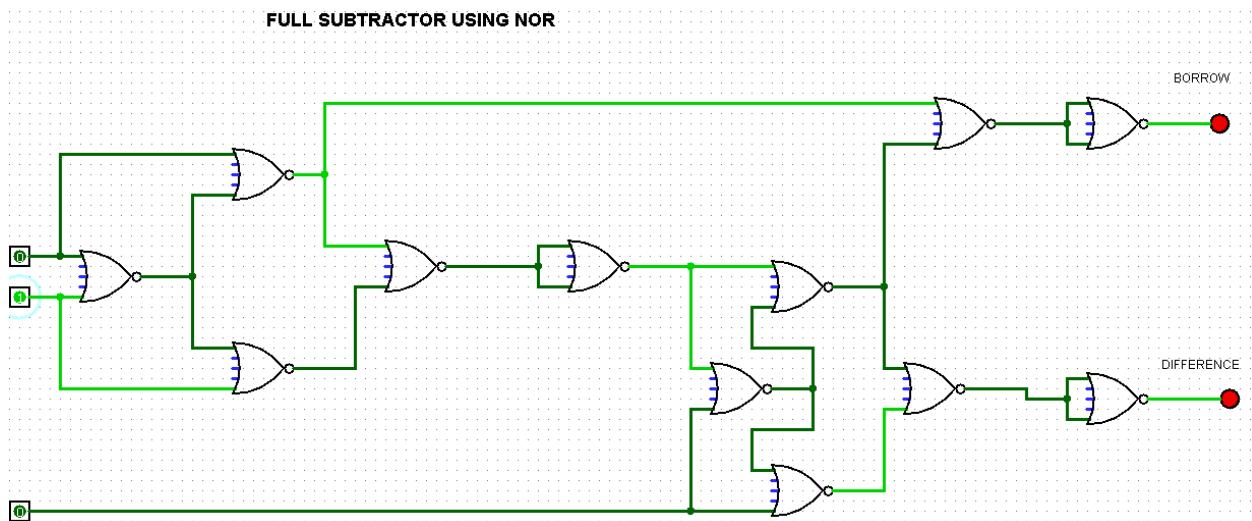
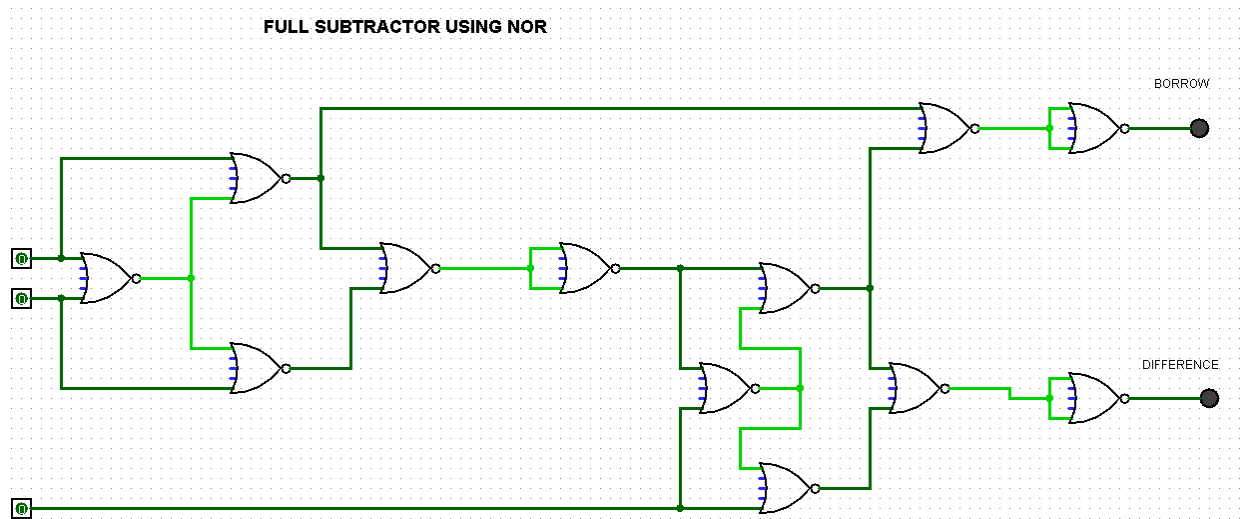
3)



Aim: To construct and simulate FA/FS &HA/HS

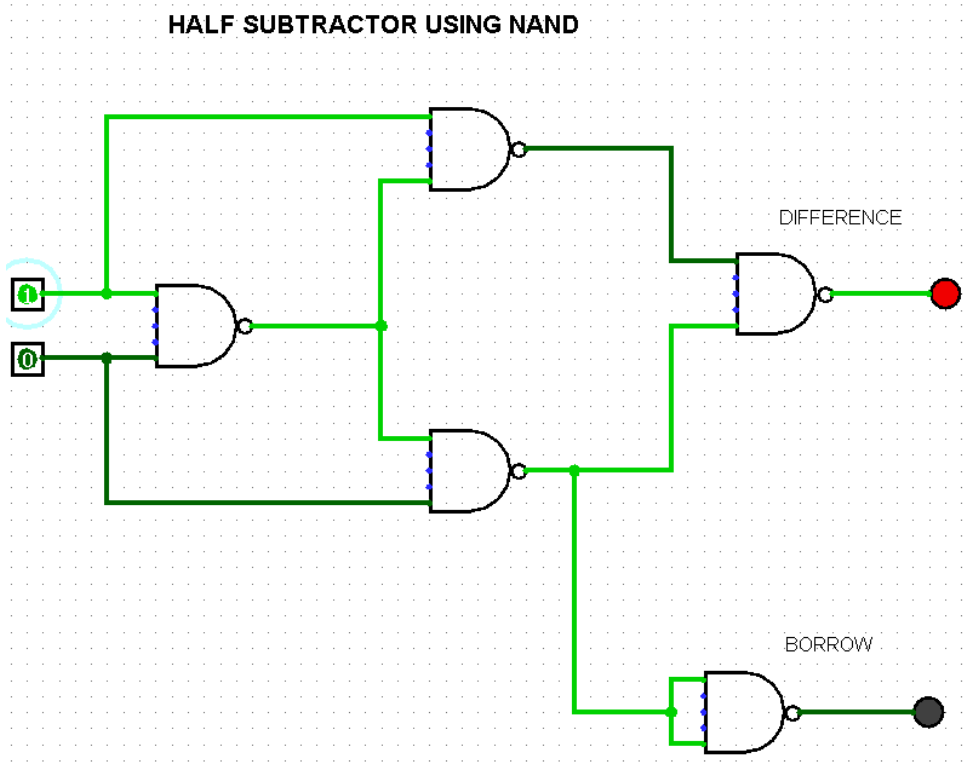
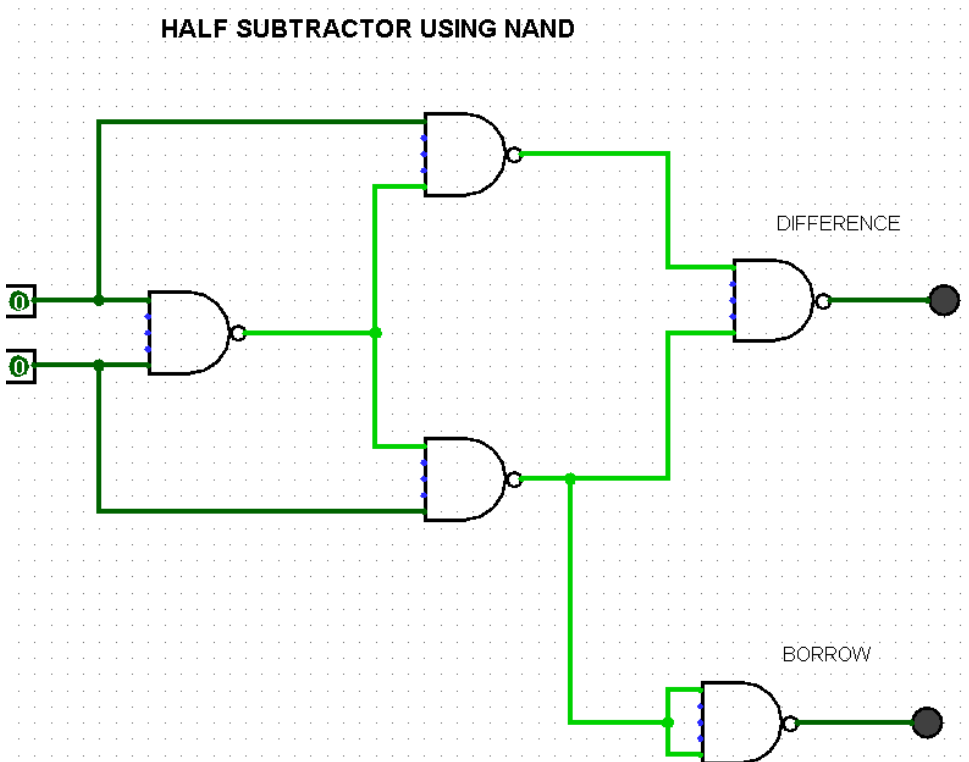


Aim: To construct and simulate FA/FS &HA/HS



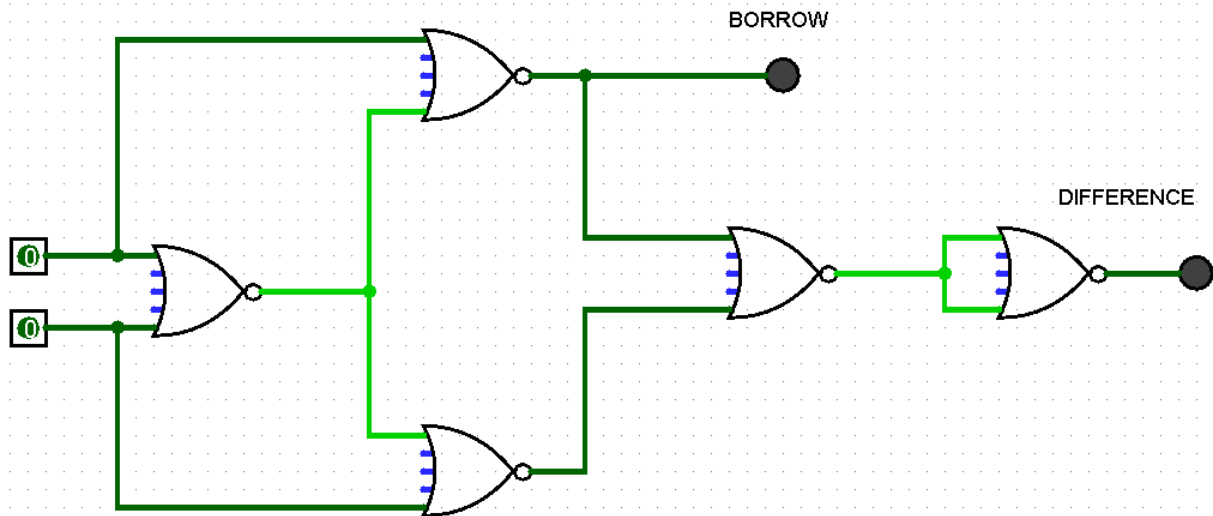
Aim: To construct and simulate FA/FS &HA/HS

4)

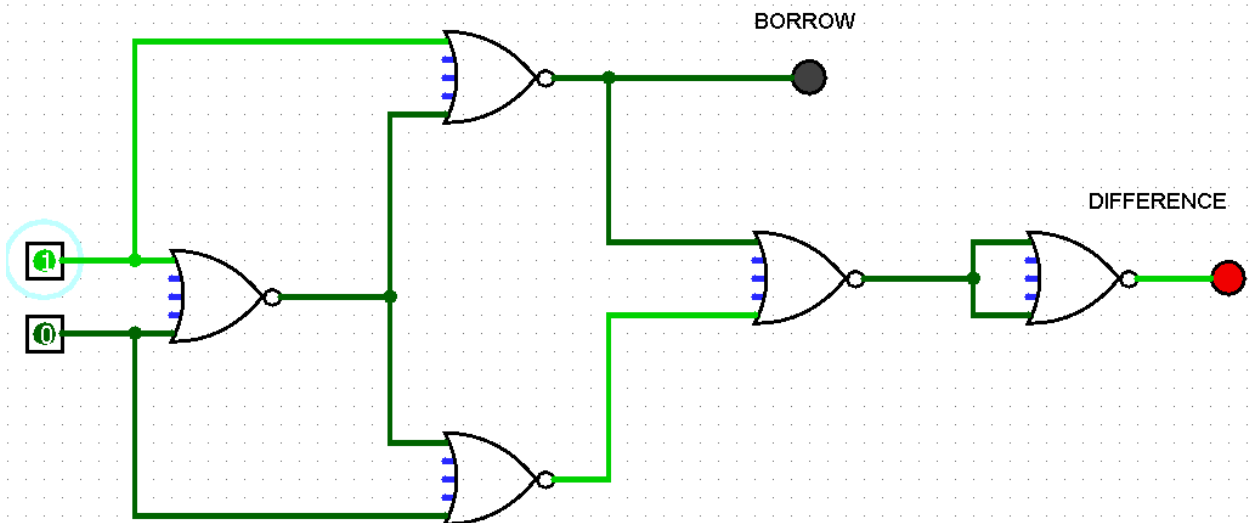


Aim: To construct and simulate FA/FS & HA/HS

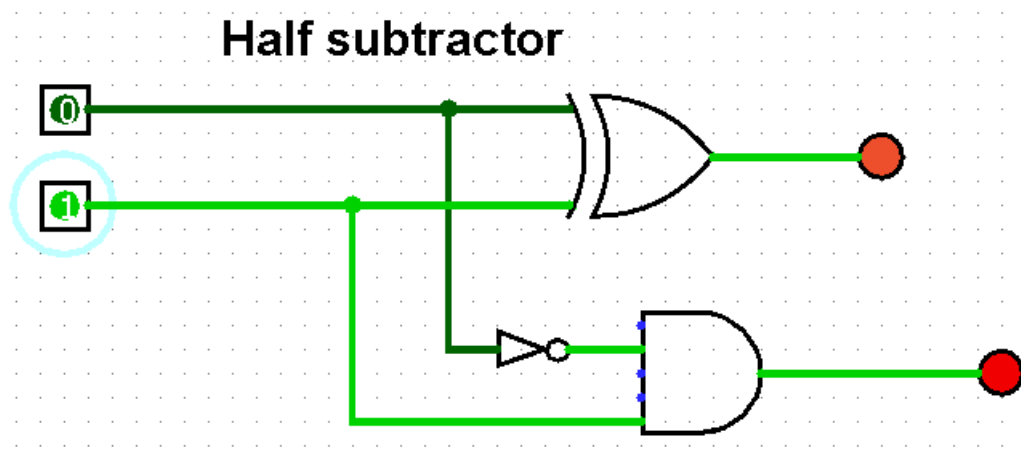
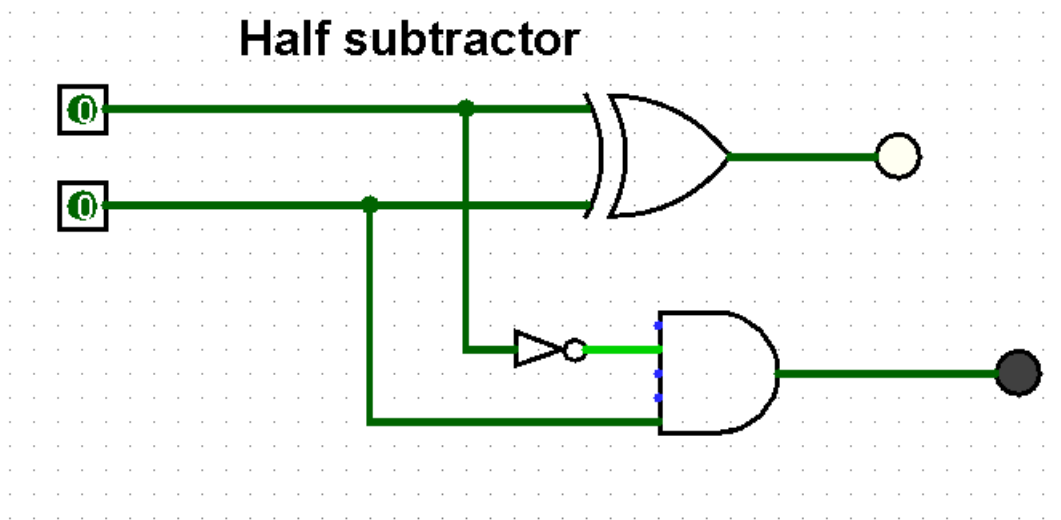
HALF SUBTRACTOR USING NOR



HALF SUBTRACTOR USING NOR



Aim: To construct and simulate FA/FS &HA/HS



Aim: To construct and simulate FA/FS & HA/HS

Observation:

Observations:

① Half Adder:

<u>Inputs</u>		<u>Outputs</u>	
<u>A</u>	<u>B</u>	<u>S</u>	<u>C</u>
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$\text{Sum} = \bar{A}B + A\bar{B} = A \oplus B$$

$$\text{Carry} = AB$$

② Full Adder:

<u>Inputs</u>			<u>Outputs</u>	
<u>A</u>	<u>B</u>	<u>C_{in}</u>	<u>S</u>	<u>C</u>
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

$$\text{Sum} = A \oplus B \oplus C_{in}$$

$$\text{Carry} = AB + B C_{in} + A C_{in}$$

③ Half Subtractor:

<u>Inputs</u>		<u>Outputs</u>	
<u>A</u>	<u>B</u>	<u>D</u>	<u>B_b</u>
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$\text{Difference} = A \oplus B$$

$$\text{Borrow} = \bar{A}B$$

④ Full Subtractor:

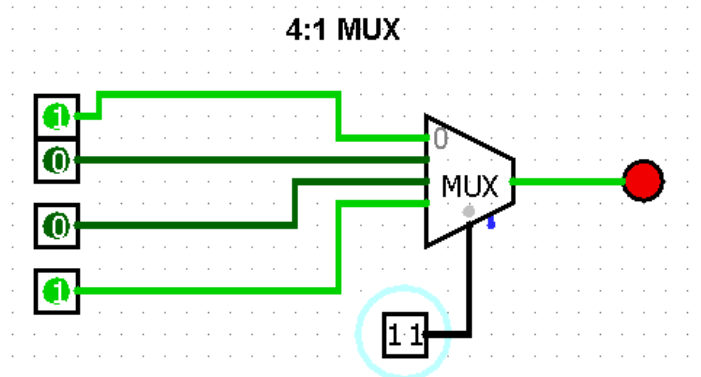
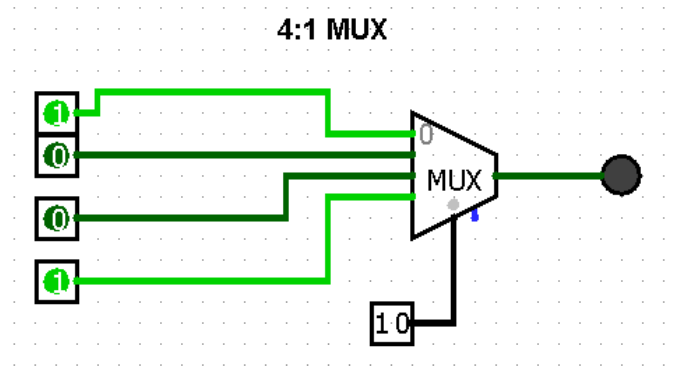
<u>Inputs</u>			<u>Outputs</u>	
<u>A</u>	<u>B</u>	<u>B_{in}</u>	<u>D</u>	<u>B_o</u>
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$D_o = A \oplus B \oplus B_{in}$$

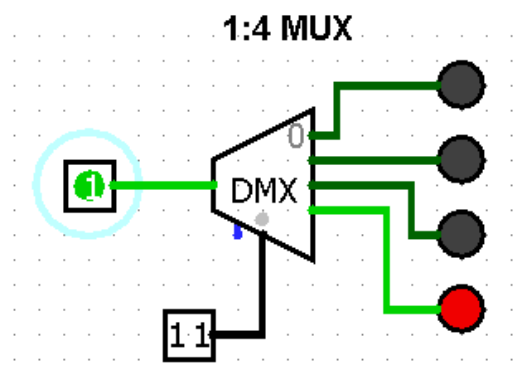
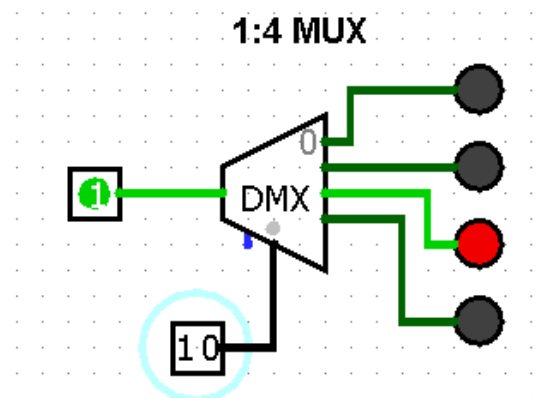
$$B_o = \bar{A}\bar{B}B_{in} + A\bar{B}B_{in} + \bar{A}BB_{in} + ABB_{in}$$

Aim: To simulate various MUX/DEMUX configs

1.



2.



Aim: To simulate various MUX/DEMUX configs

Observation :

4:1 MUX :

<u>INPUT</u>		<u>Data INPUT</u>				<u>Output</u>
<u>S₁</u>	<u>S₀</u>	<u>I₃</u>	<u>I₂</u>	<u>I₁</u>	<u>I₀</u>	<u>Y</u>
0	0	x	x	x	I ₀	I ₀
0	1	x	x	I ₁	x	I ₁
1	0	x	I ₂	x	x	I ₂
1	1	I ₃	x	x	x	I ₃

$$Y = \bar{S}_1 \bar{S}_0 I_0 + \bar{S}_1 S_0 I_1 + S_1 \bar{S}_0 I_2 + S_1 S_0 I_3$$

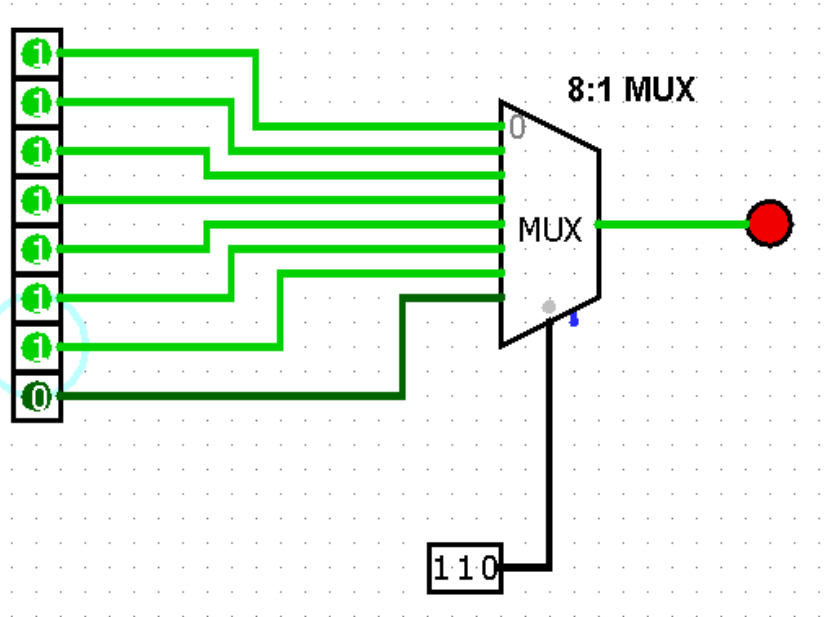
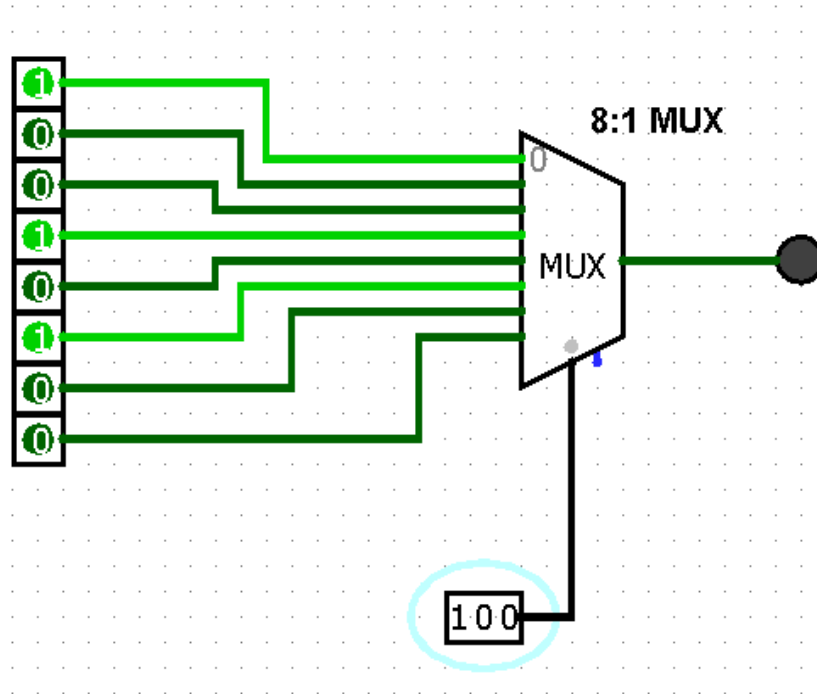
1:4 Demux :

<u>Data Input</u>	<u>Select Input</u>		<u>Output</u>			
<u>I</u>	<u>S₁</u>	<u>S₀</u>	<u>Y₃</u>	<u>Y₂</u>	<u>Y₁</u>	<u>Y₀</u>
I	0	0	0	0	0	I
I	0	1	0	0	I	0
I	1	0	0	I	0	0
I	1	1	I	0	0	0

$$Y_0 = \bar{S}_1 \bar{S}_0 I \quad Y_1 = \bar{S}_1 S_0 I \quad Y_2 = S_1 \bar{S}_0 I \quad Y_3 = S_1 S_0 I$$

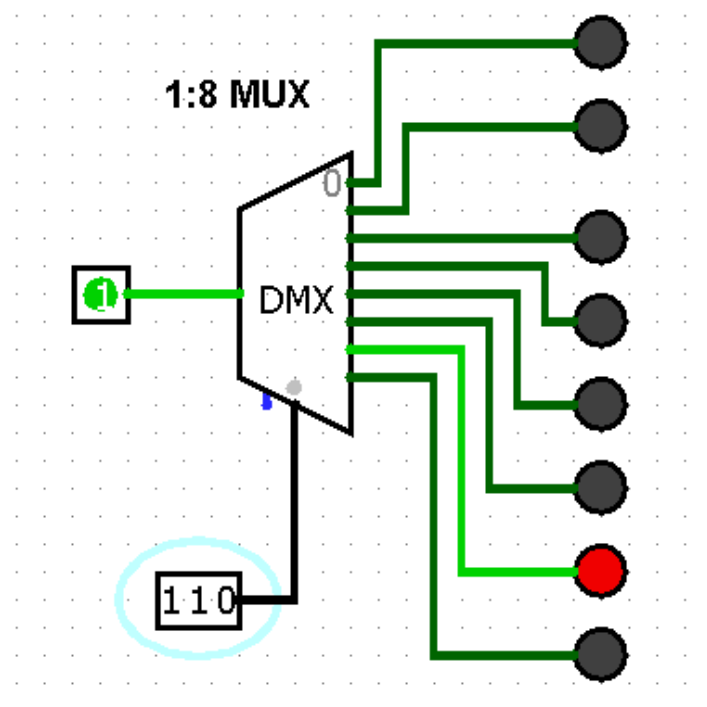
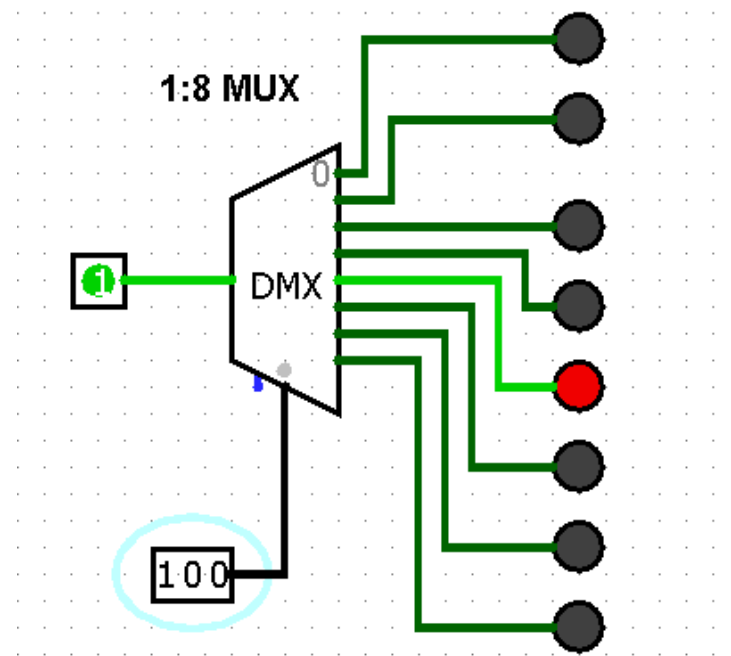
Aim: To simulate various MUX/DEMUX configs

3.



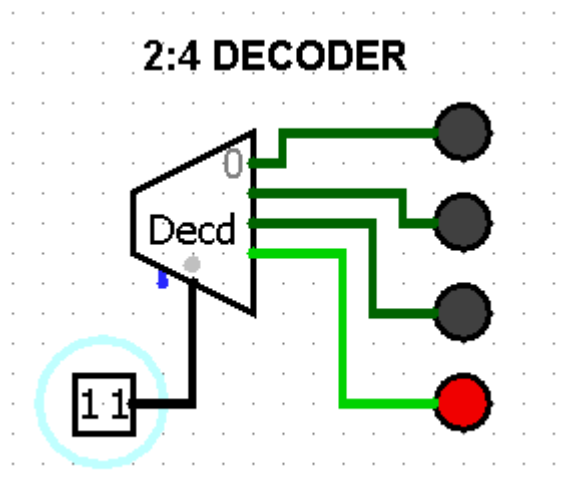
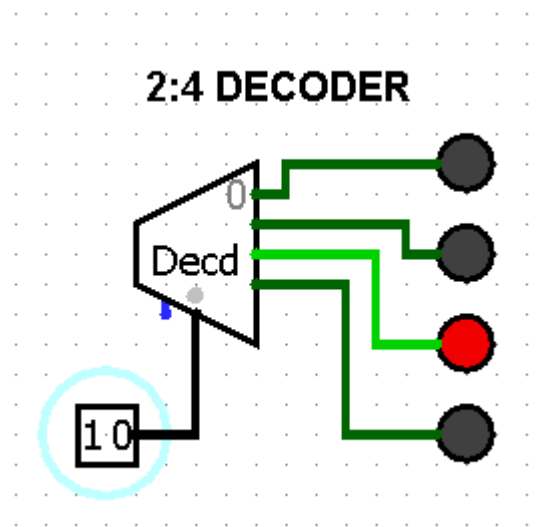
Aim: To simulate various MUX/DEMUX configs

4.



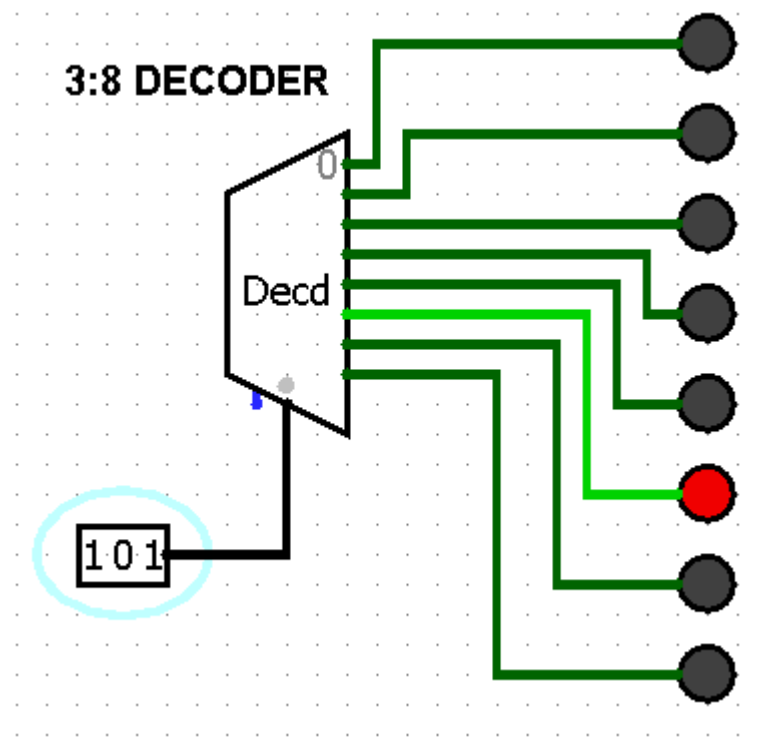
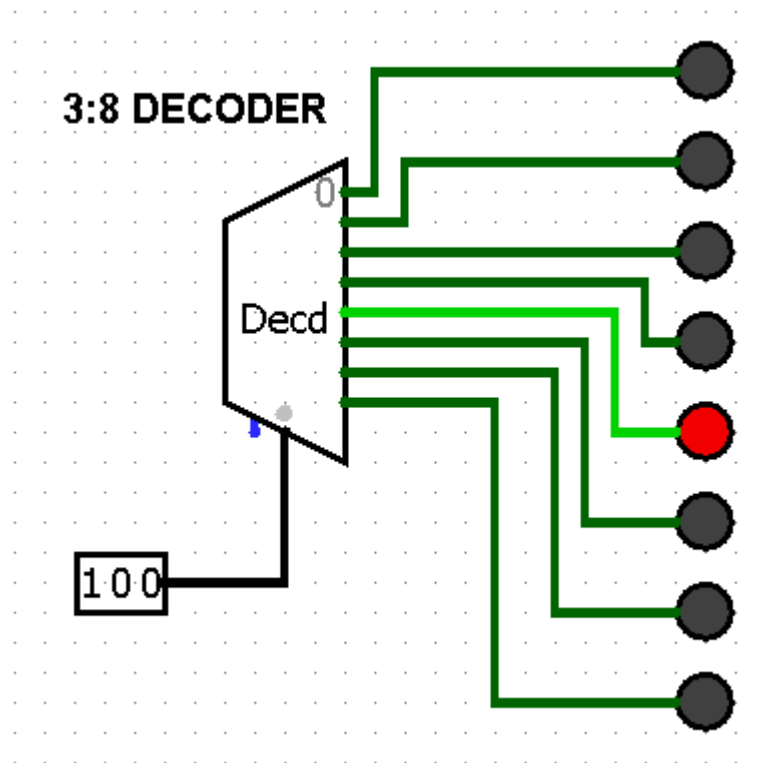
Aim: To simulate various MUX/DEMUX configs

5.



Aim: To simulate various MUX/DEMUX configs

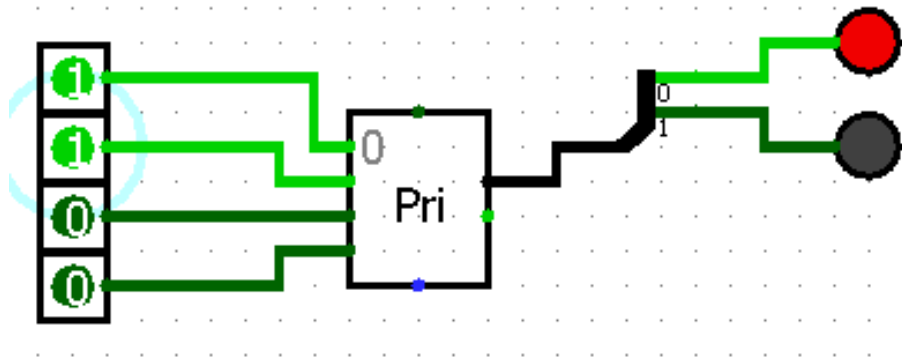
6.



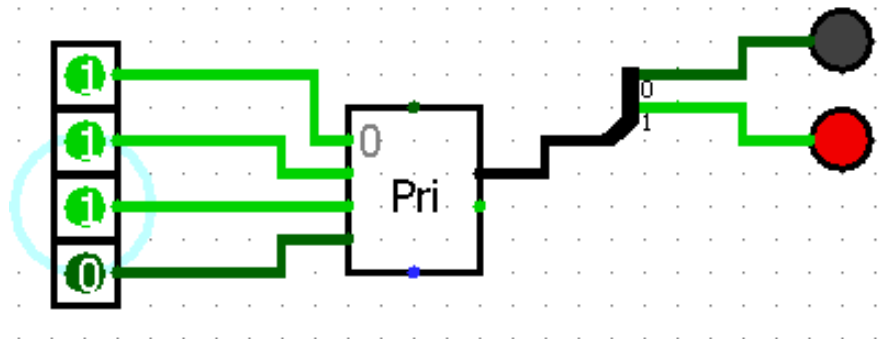
Aim: To simulate various MUX/DEMUX configs

7.

4:2 PRIORITY ENCODER

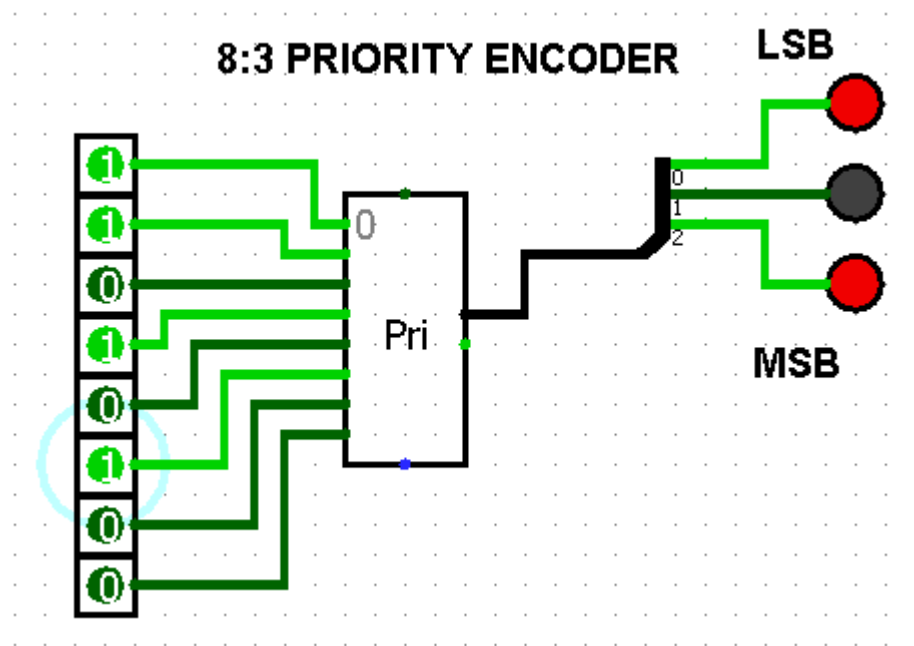
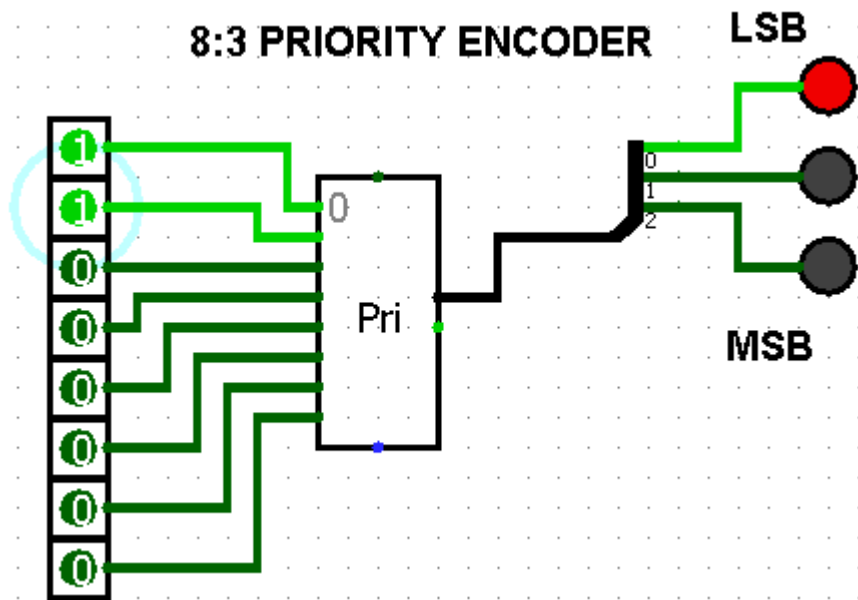


4:2 PRIORITY ENCODER



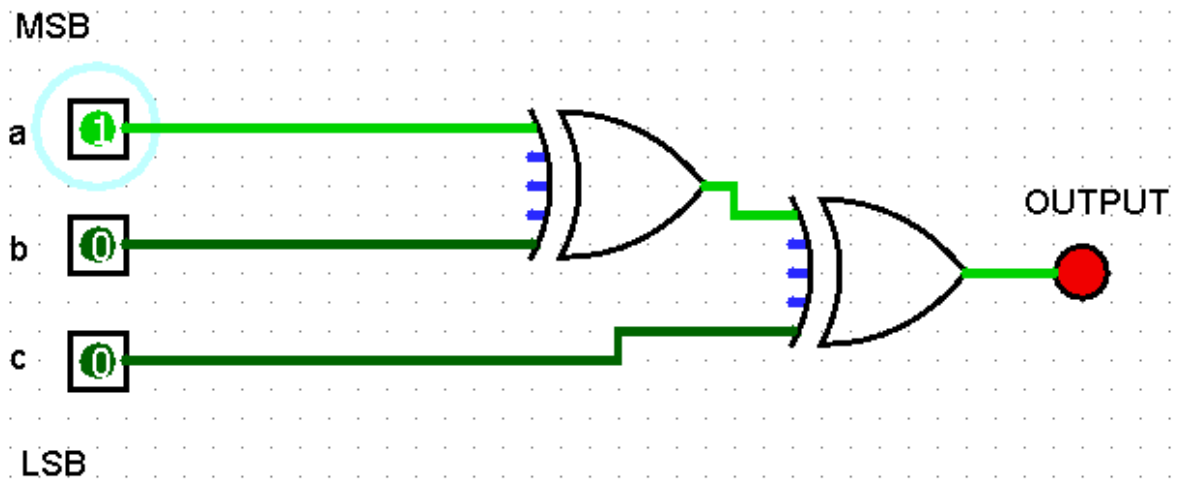
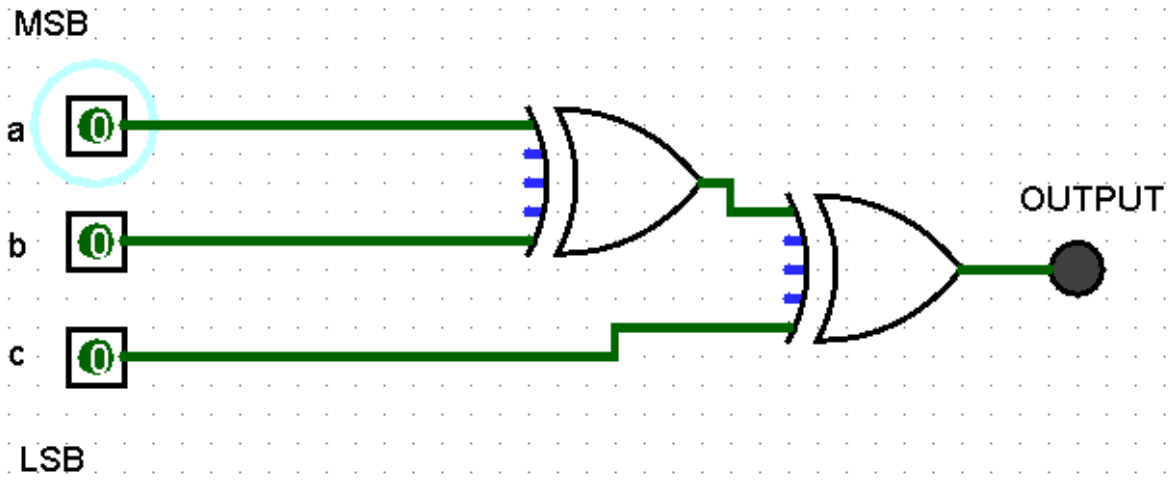
Aim: To simulate various MUX/DEMUX configs

8.

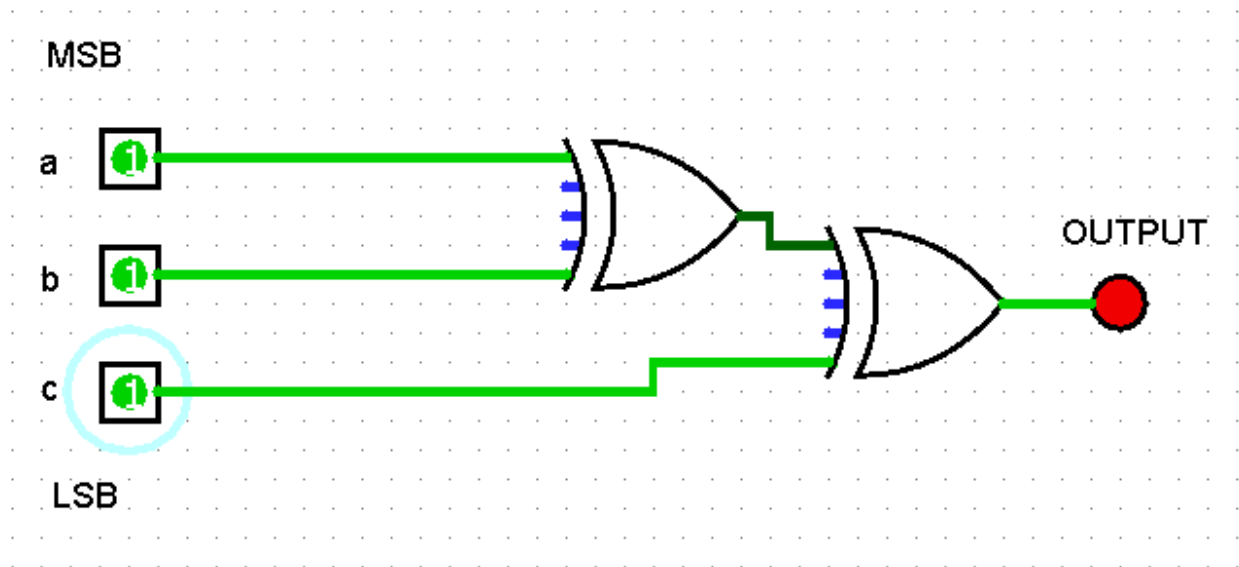


1.

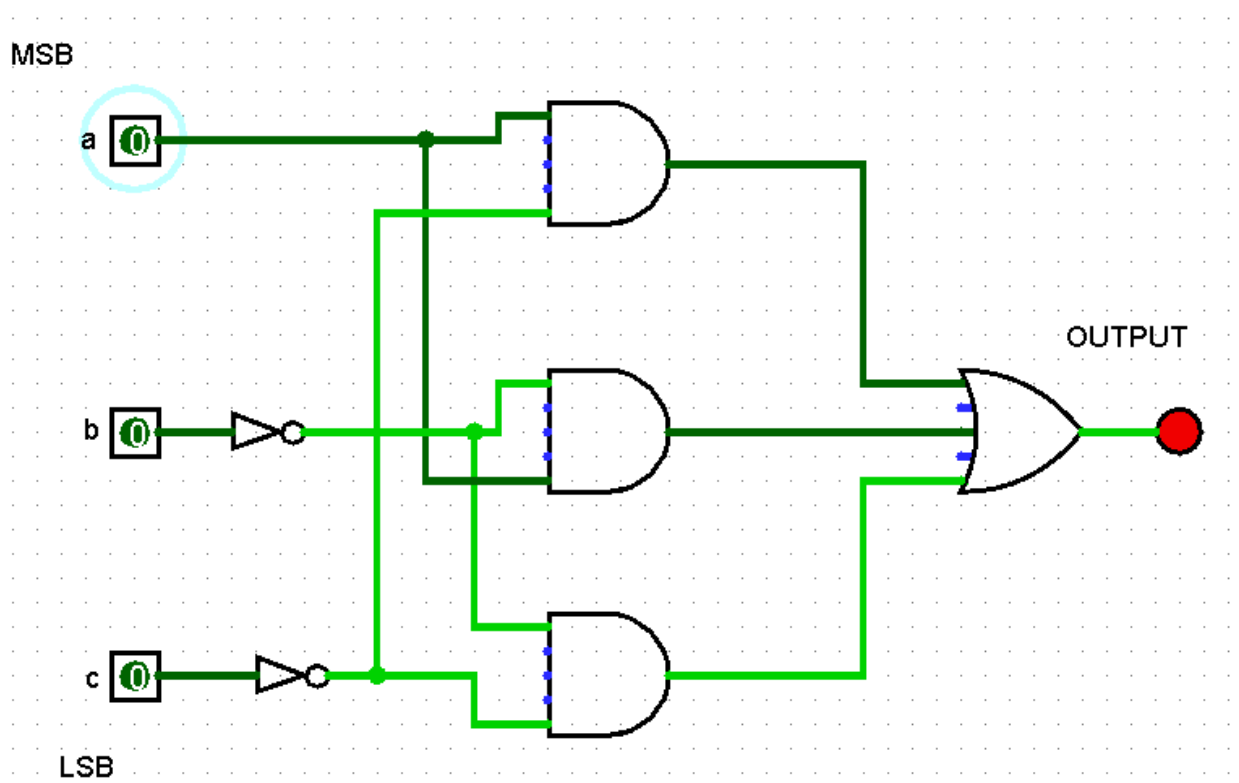
REALIZATION OF BOOLEAN EXPRESSION



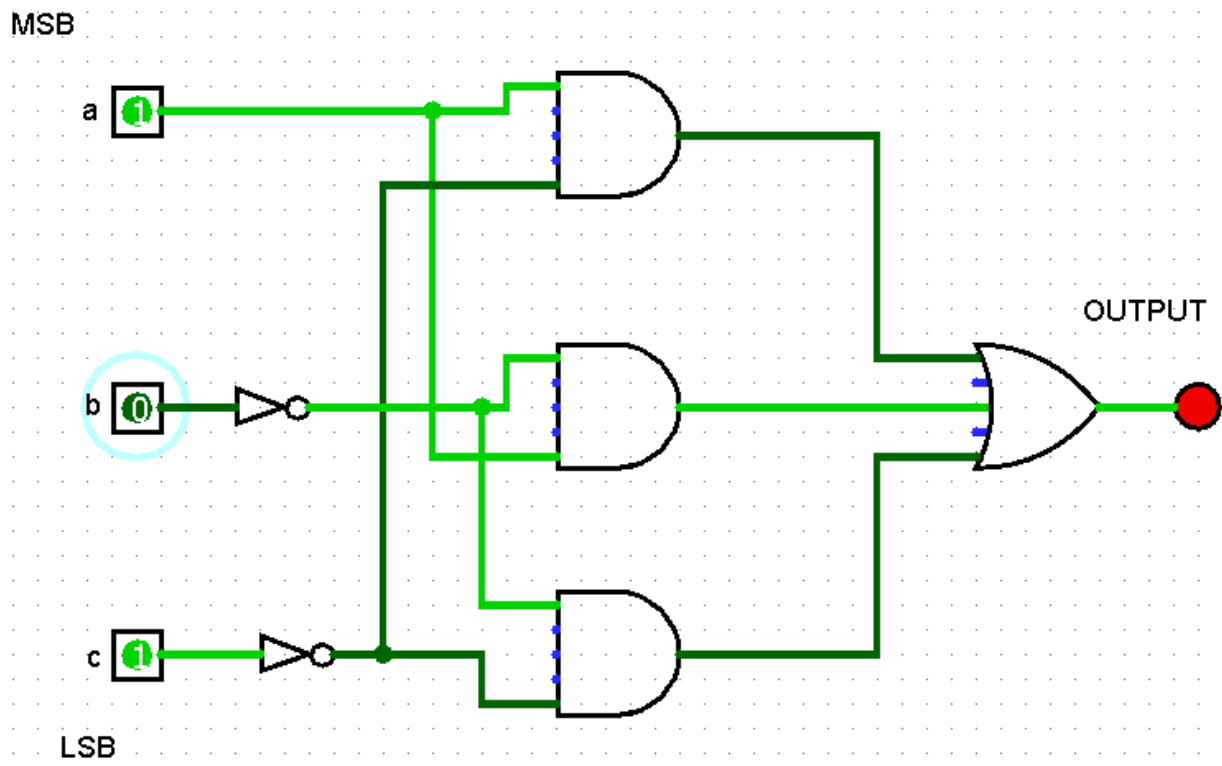
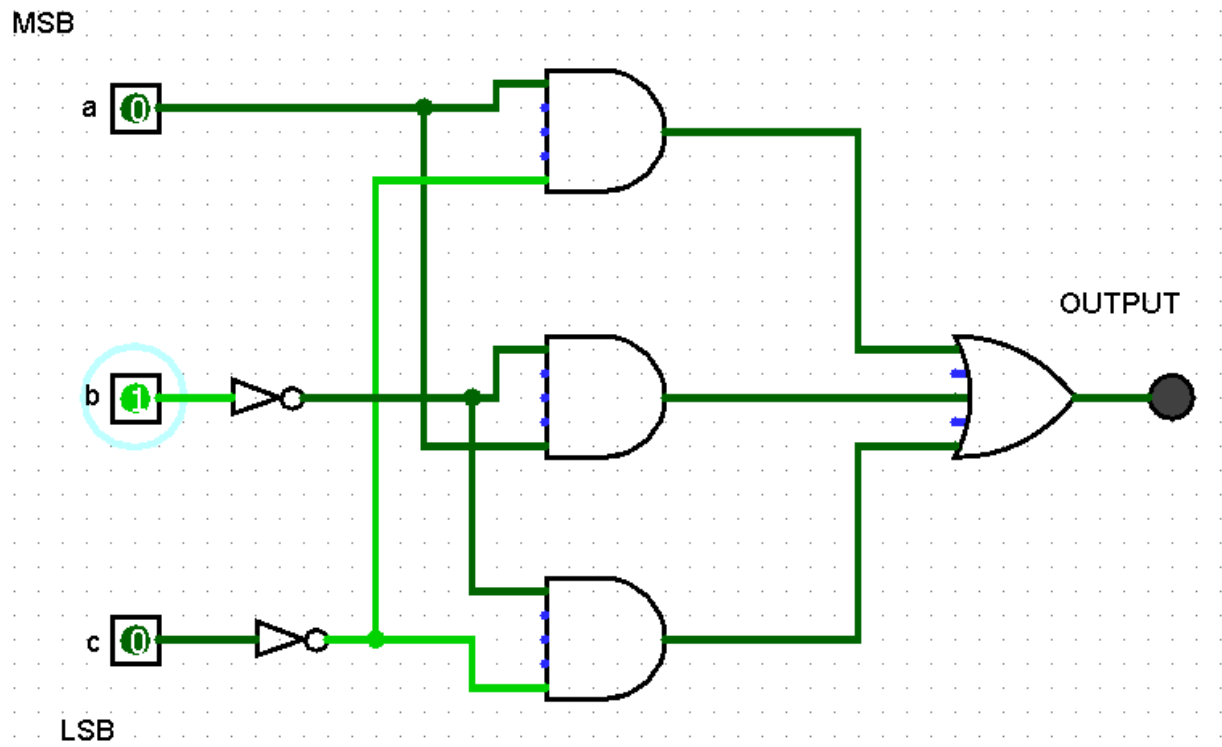
Aim: Realization of Boolean Expression/ MUX & Demux using Gates



2.

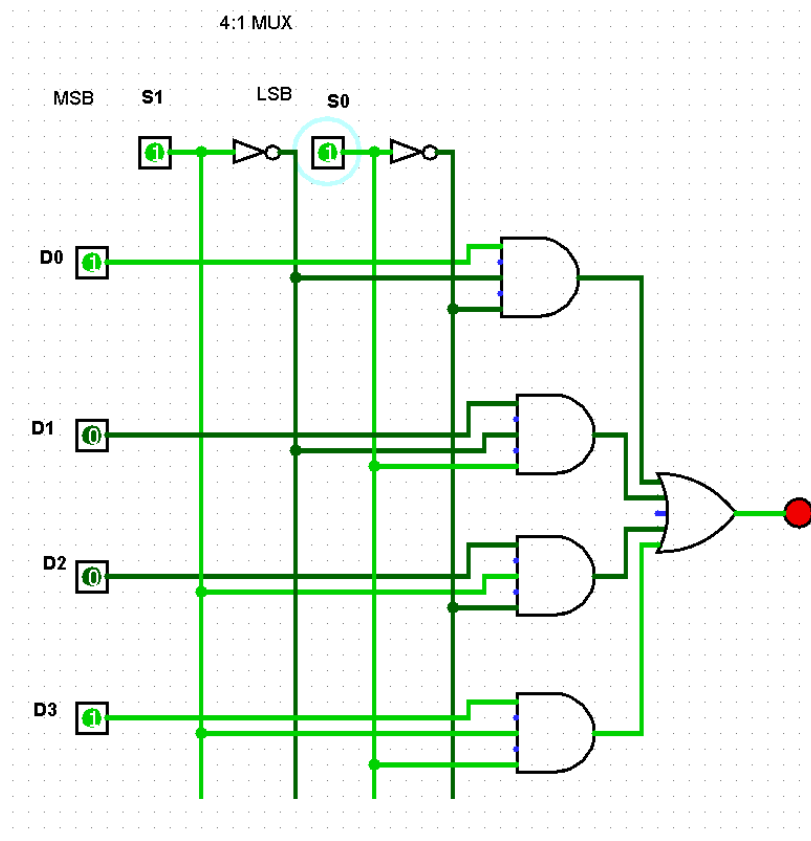
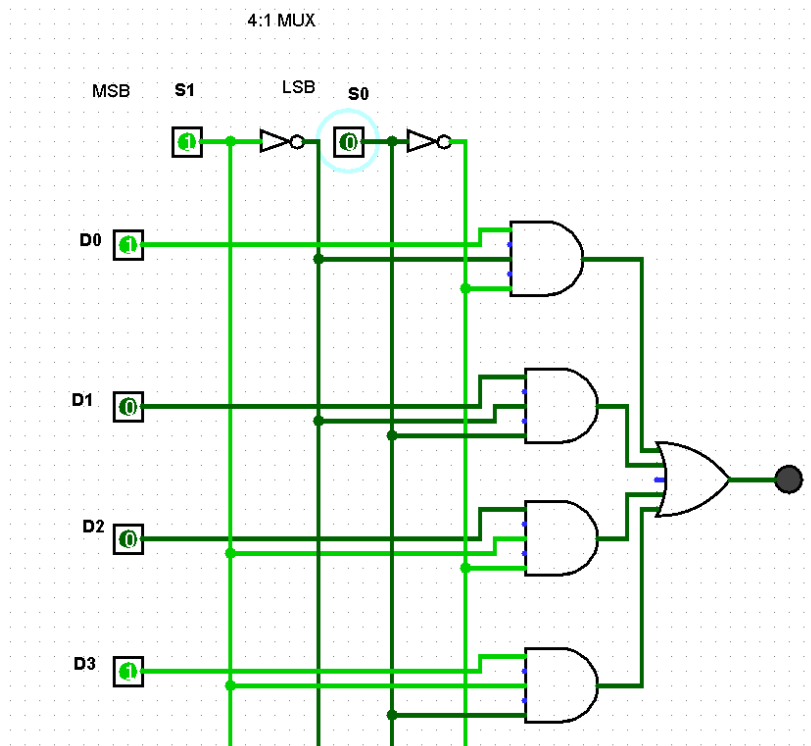


Aim: Realization of Boolean Expression/ MUX & Demux using Gates



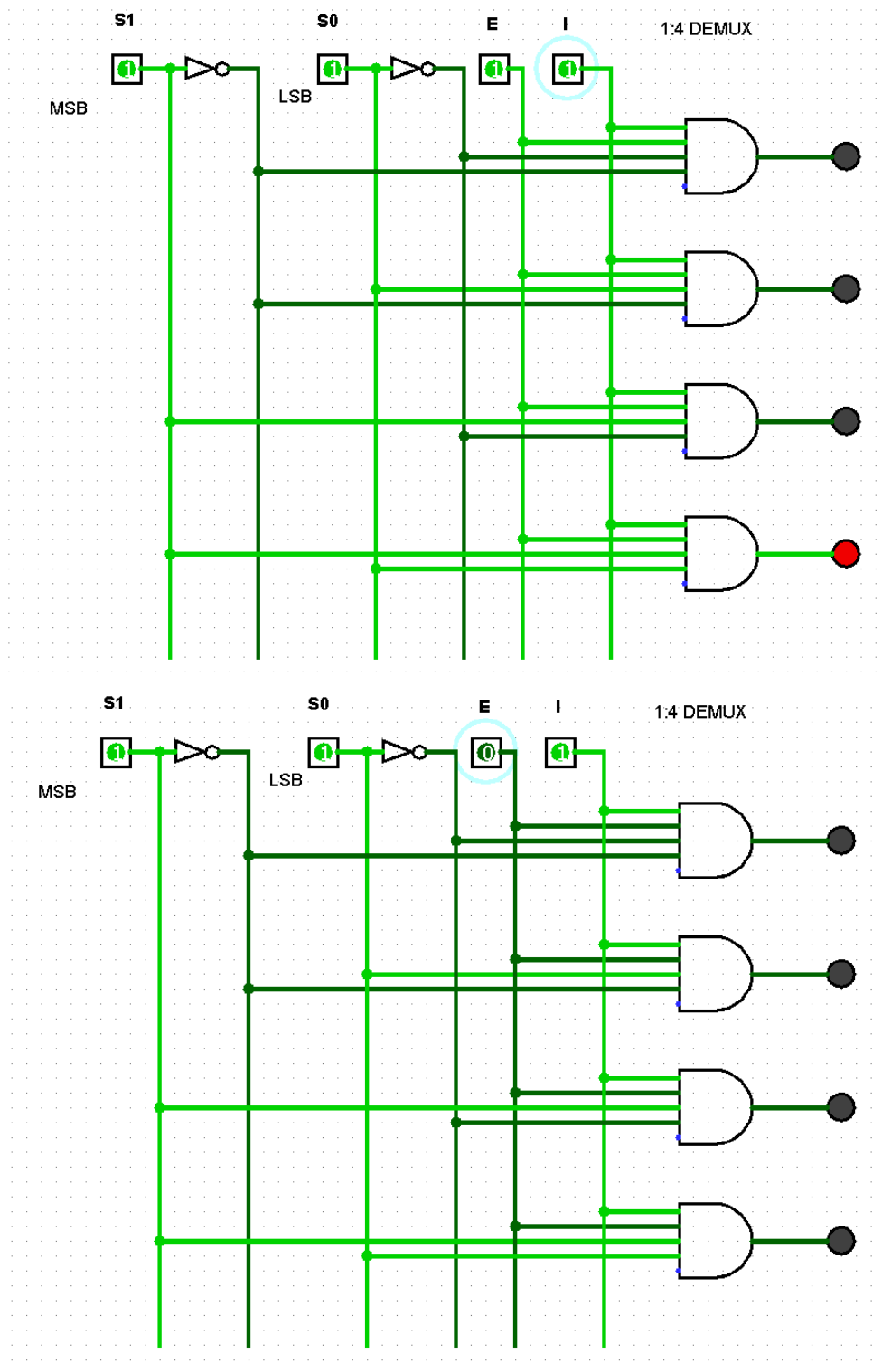
Aim: Realization of Boolean Expression/ MUX & Demux using Gates

3.



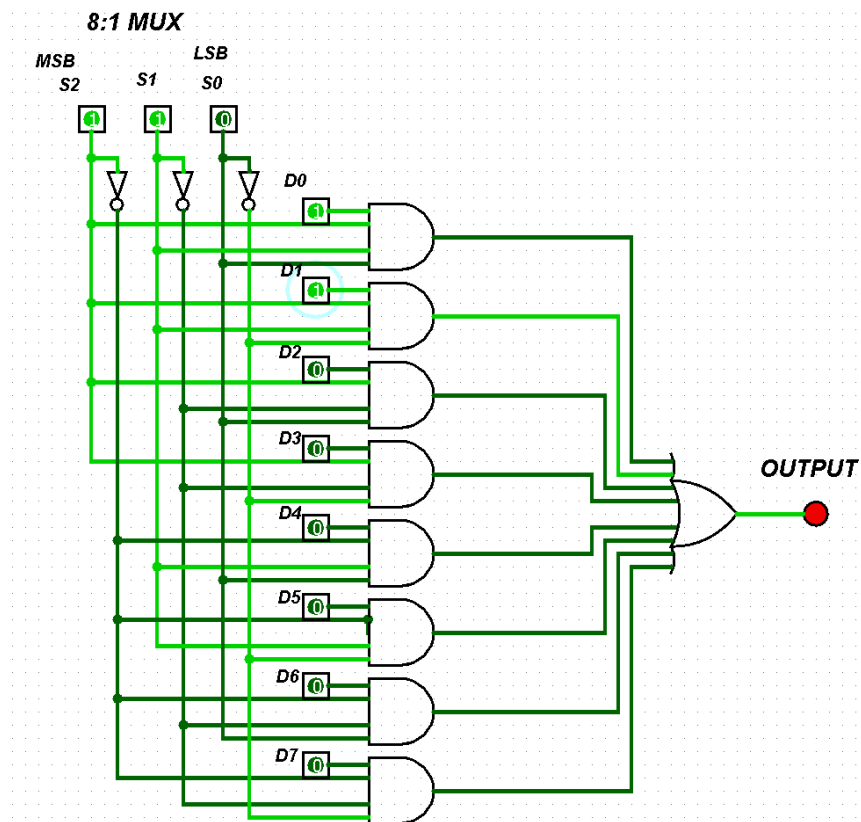
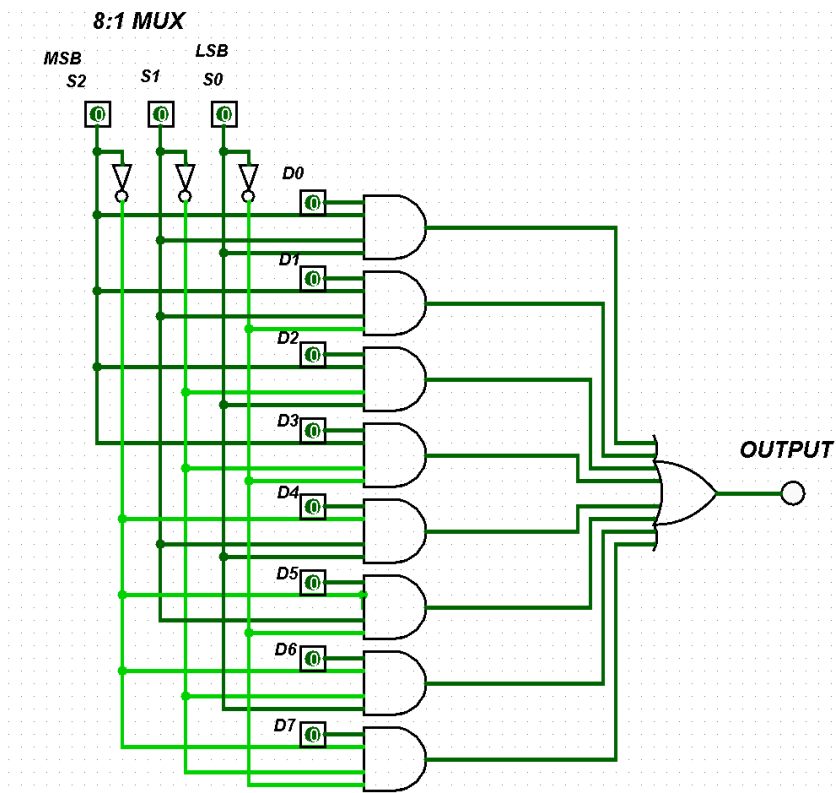
Aim: Realization of Boolean Expression/ MUX & Demux using Gates

4.

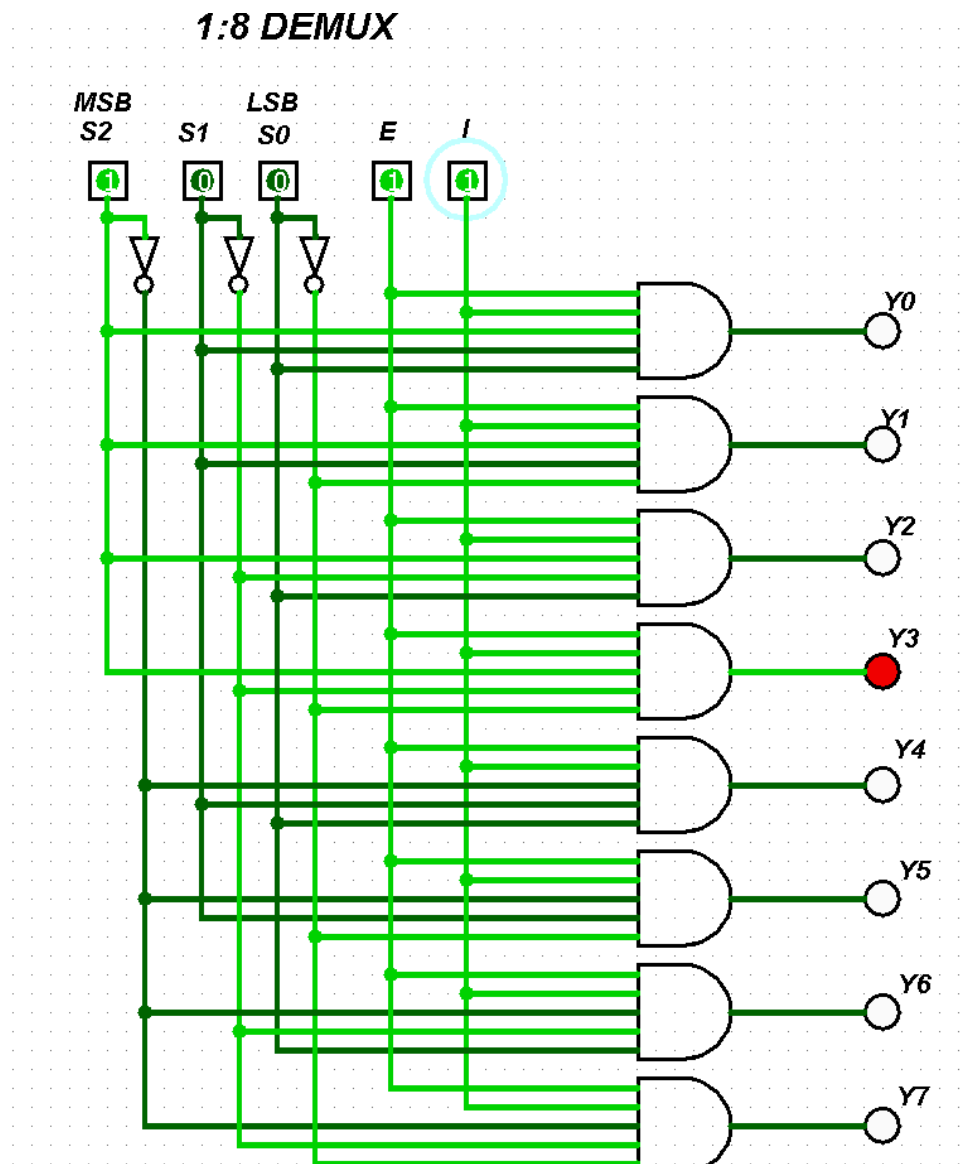


Aim: Realization of Boolean Expression/ MUX & Demux using Gates

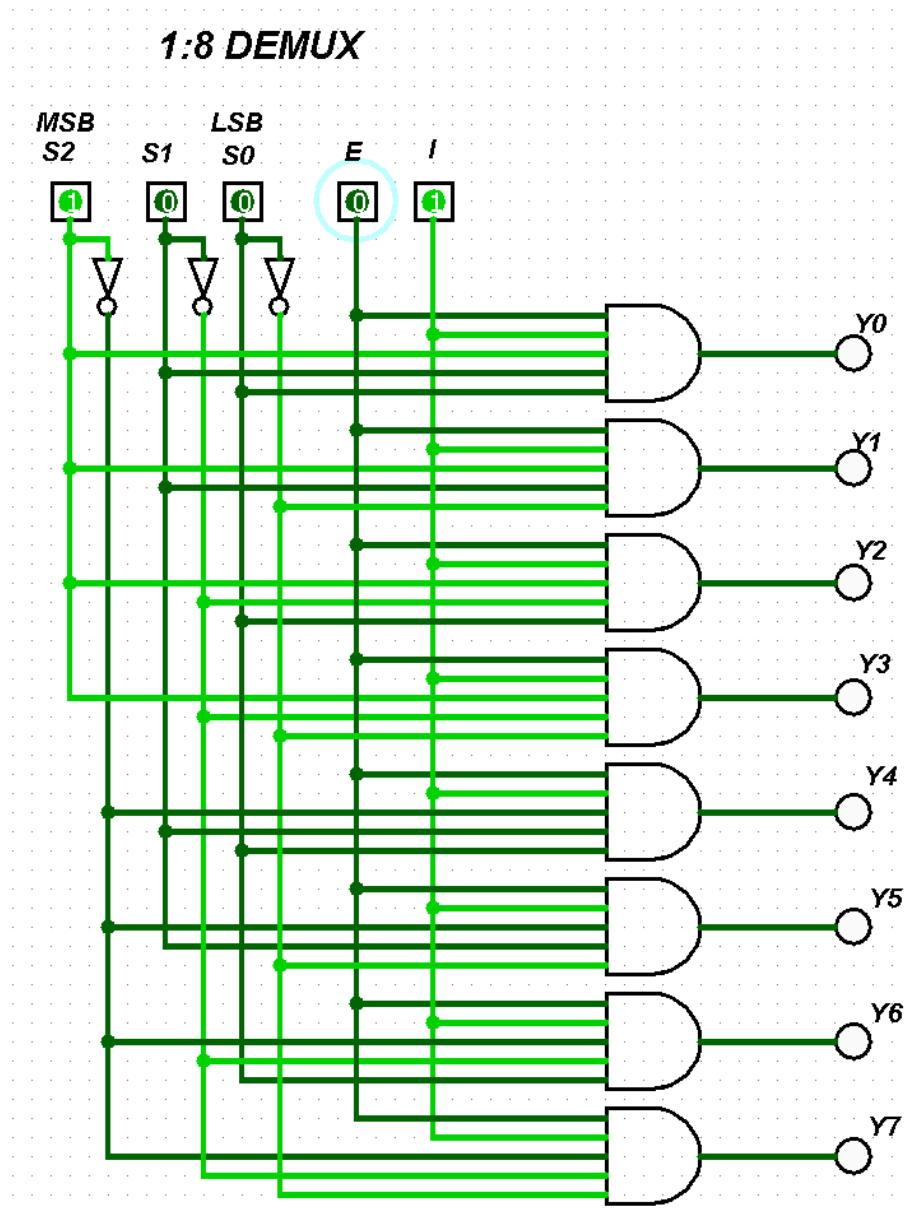
5.



6.



Aim: Realization of Boolean Expression/ MUX & Demux using Gates



1.

The timing diagram shows the signals for two D flip-flops and an AND gate over time. The signals are:

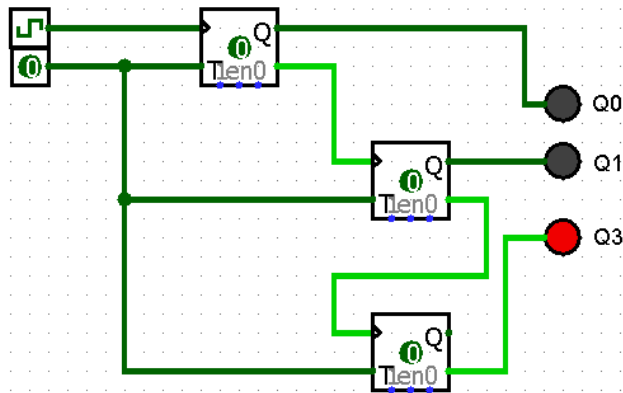
- Input 1 (Top):** A square wave signal.
- Input 2 (Middle):** A square wave signal.
- AND Gate Output (Bottom):** The output of an AND gate, which is high only when both input signals are high.
- Flip-Flop 1 (Top):** A D flip-flop with output Q. The input D is connected to the AND gate output. The output Q is connected to the input D of the second flip-flop.
- Flip-Flop 2 (Bottom):** A D flip-flop with output Q. The input D is connected to the input signal. The output Q is connected to the input D of the first flip-flop.

The diagram illustrates the timing relationship between the input signals, the AND gate output, and the outputs of the two flip-flops. The flip-flop outputs are shown as square waves that change state at the rising edges of the clock signal (the AND gate output).

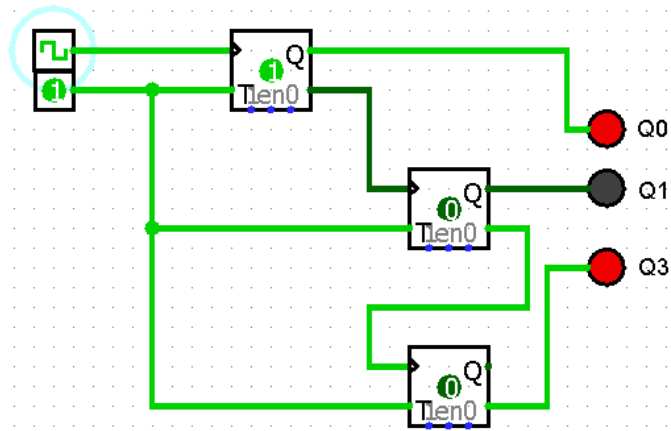
Aim : To design and simulate various counters

2.

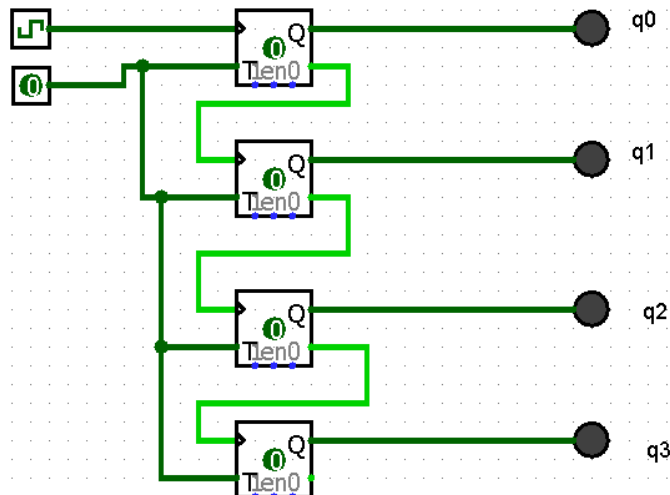
TWO BIT ASYNCHRONOUS COUNTER



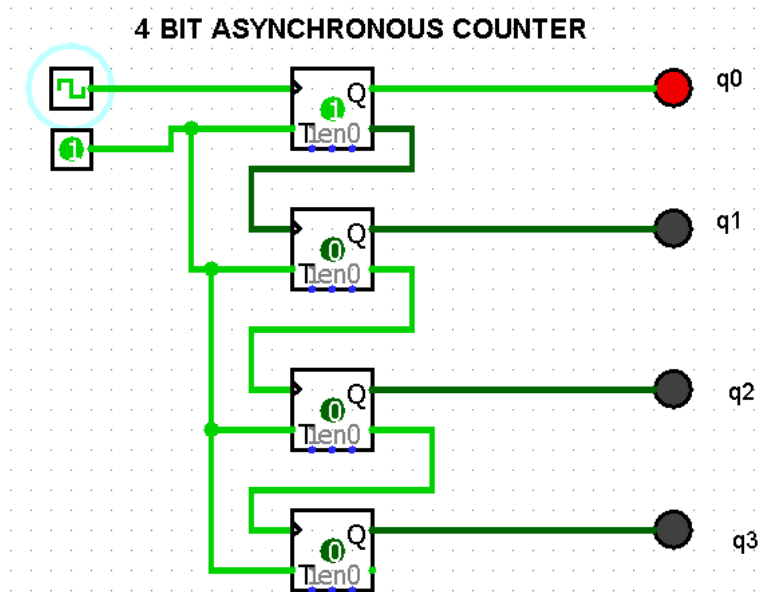
TWO BIT ASYNCHRONOUS COUNTER



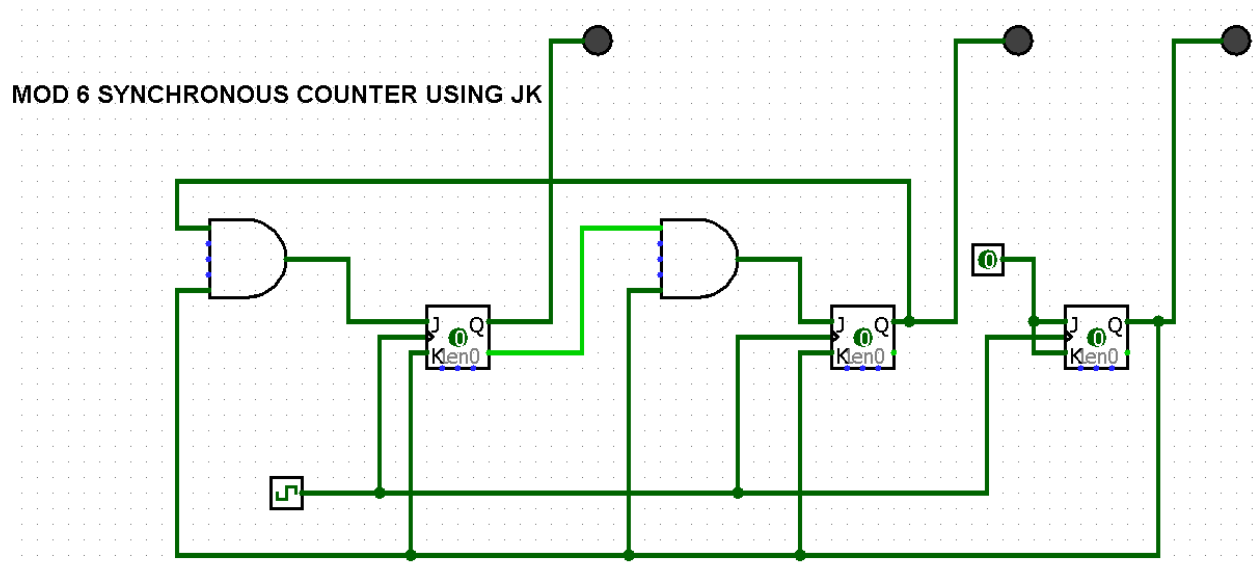
4 BIT ASYNCHRONOUS COUNTER



Aim : To design and simulate various counters

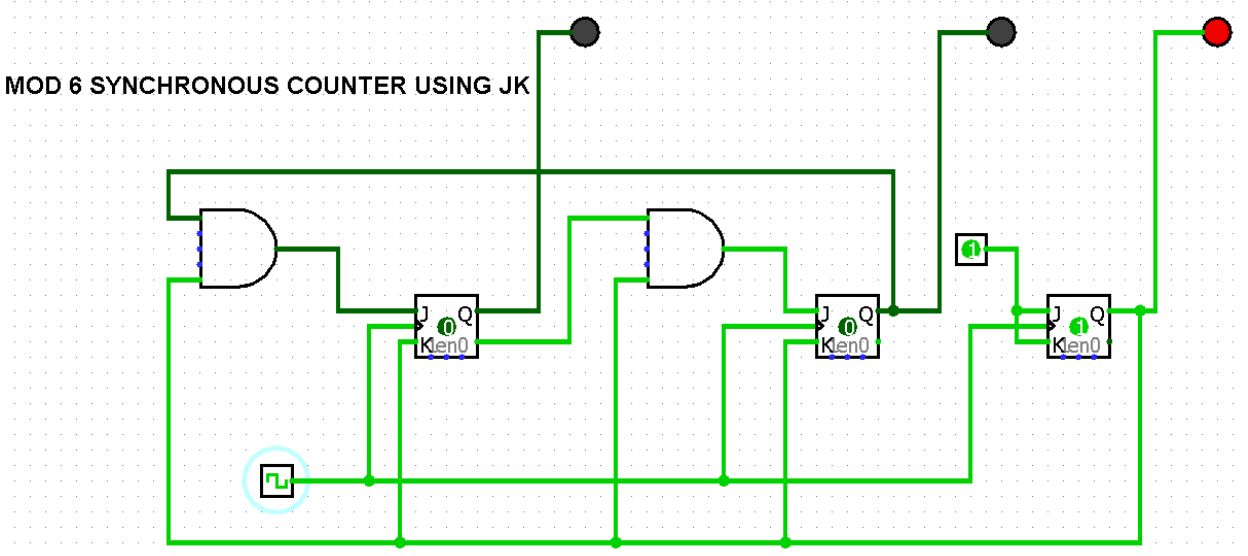


3.



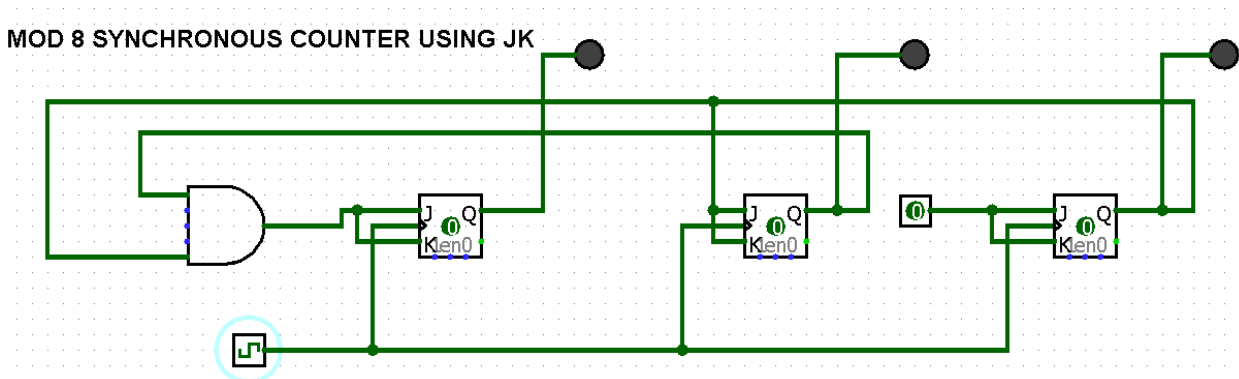
Aim : To design and simulate various counters

MOD 6 SYNCHRONOUS COUNTER USING JK

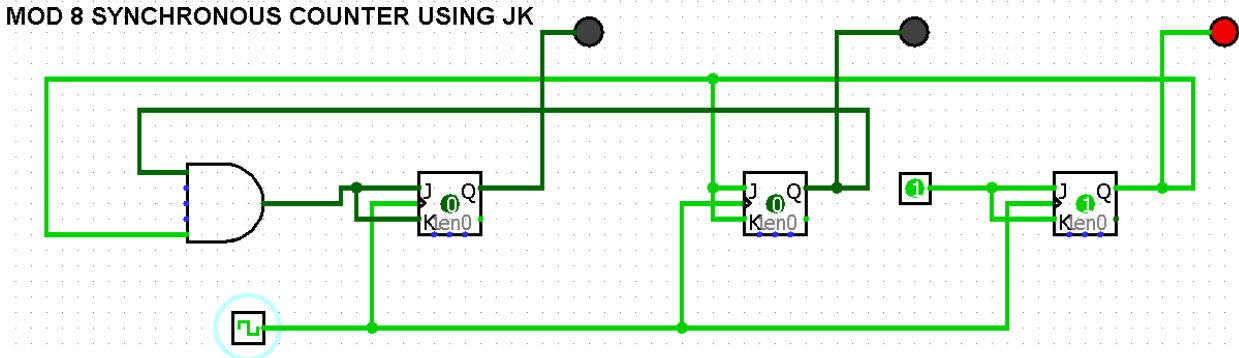


4.

MOD 8 SYNCHRONOUS COUNTER USING JK

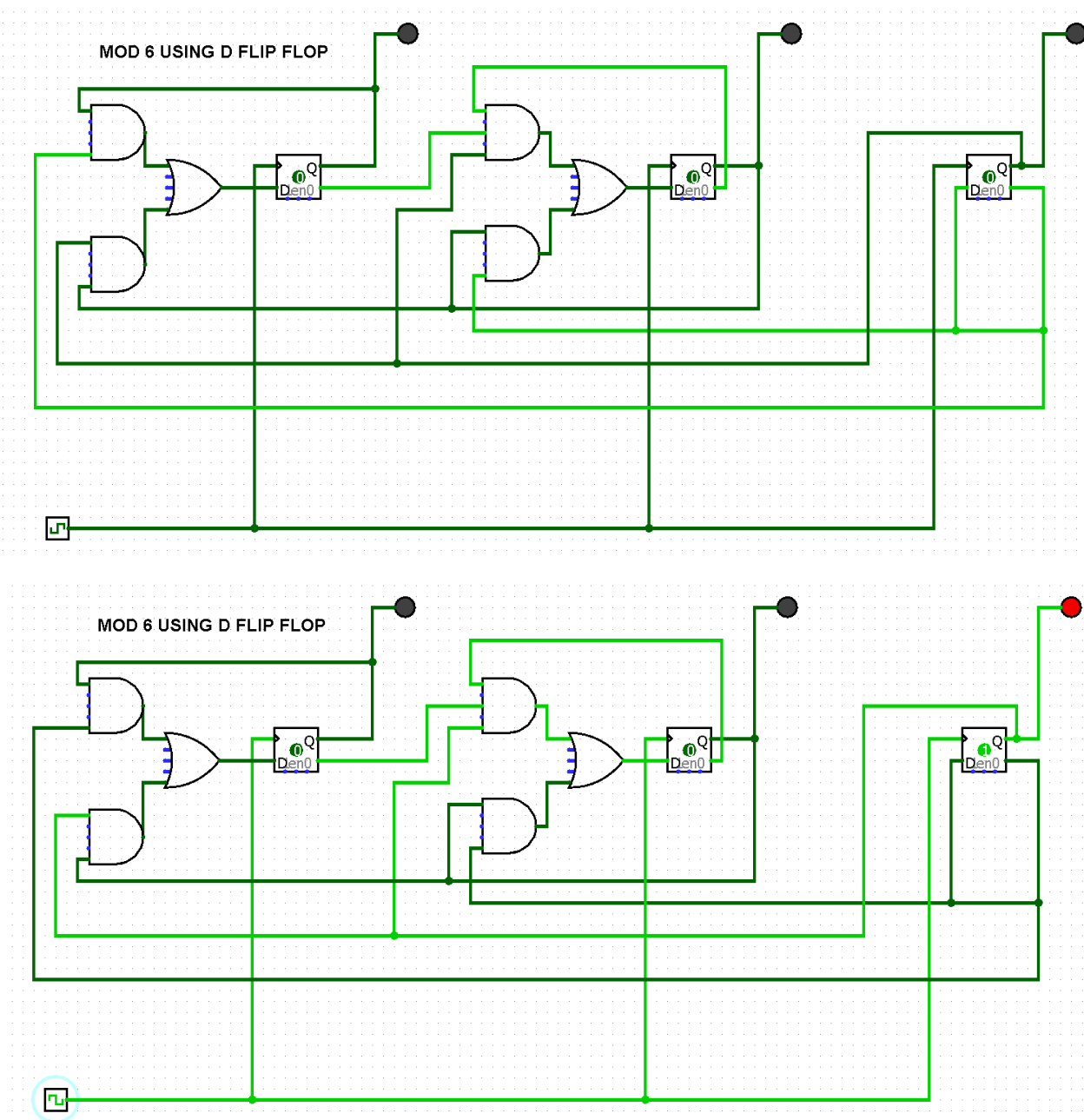


MOD 8 SYNCHRONOUS COUNTER USING JK



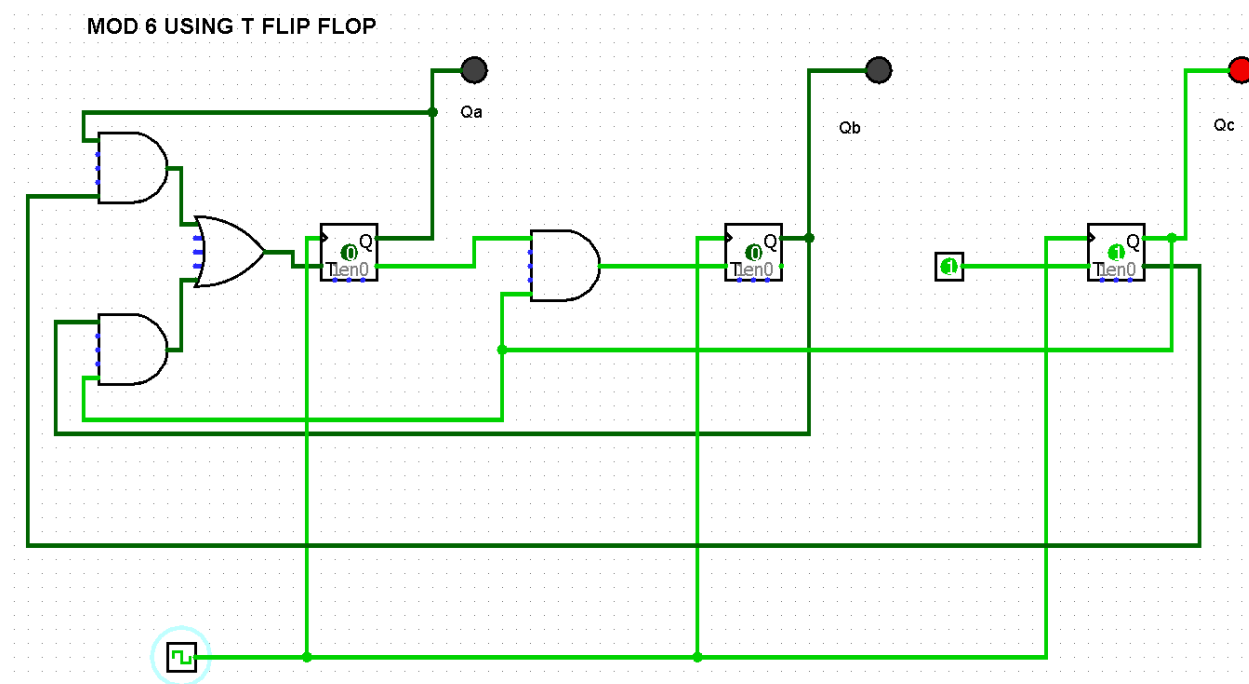
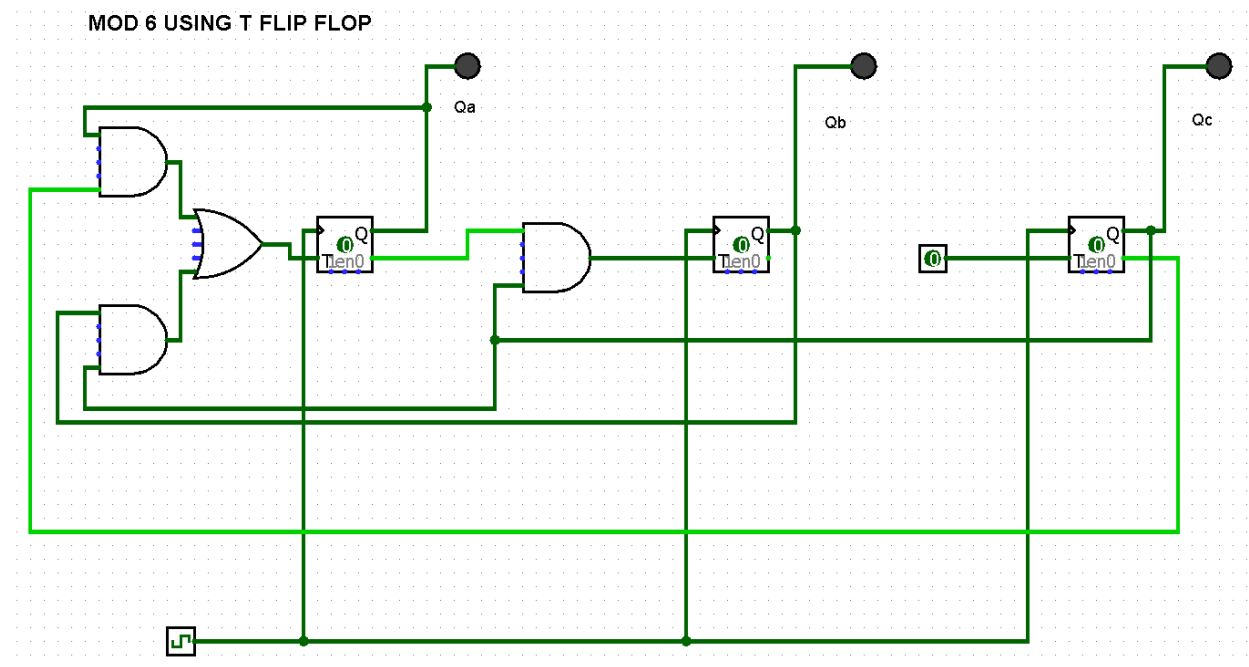
Aim : To design and simulate various counters

5.



Aim : To design and simulate various counters

6.



Aim : To design and simulate various counters

7.

