CS225/226

MINI PROJECT PART 1

LAB EXPERIMENT

Gaurav Kumar
1701CS21

PROJECT AIM:

Aim of this project is to perform these 2 tasks in Logisim :

1. Designing and simulating an ALU which performs 4 operations in Logisim software

2. Designing and simulating a sequence recognizer using FSM in Logisim software

PROBLEM 1 :

The main objective of this task is to design an ALU and simulate it in Logisim that takes in two 4 bits of input A and B and performs the multiple operations on them bit by bit and output the result, based on the operation selected.

Operations to be performed are :

1. A bitwise nand B
2. A bitwise nor B
3. A bitwise xnor B
4. A bitwise + B

PROBLEM 2 :

A sequential circuit has one input and one output. When the input sequence "1 1 0" occurs, the output should become "1" and it should remain "1" until the sequence "1 1 0" occurs again in which case output return to "0". The output remain "0" until the sequence "1 1 0" occurs a third time and so on.

COMPONENTS USED FOR PROBLEM 1 :

1. For bitwise AND: 2 4-bit inputs, 1 4-bit output, 3 splitters, 4 AND gates

2. For bitwise OR: 2 4-bit inputs, 1 4-bit output, 3 splitters, 4 OR gates

3. For bitwise XOR: 2 4-bit inputs, 1 4-bit output, 3 splitters, 4 XOR gates

4. For bitwise FULL ADDER: 3 1-bit inputs, 2 1-bit outputs, 2 XOR gates, 2 AND gates, 1 OR gate

5. For bitwise ADDER: 2 4-bit inputs, 1 1-bit input, 1 4-bit output, 11-bit output, 4 1-bit FULL ADDERs, 3 splitters, 4 XOR gates

6. For Main ALU: 2 4-bit inputs A and B, 1 2-bit input select line for mux, 1 4-bit output, 1 GND input, 3 NOT gates, AND gates, OR gate, XOR gate, 1 4-bit ADDER, 1 4:1 MULTIPLEXER with hexdisplay

COMPONENTS USED FOR PROBLEM 2 :

1. 3 D flip flops with clock, enable and clear

2. 1 1-bit input

3. 3 1-bit output for present/next state with splitter and hex display

4. 1 1-bit output for present output

5. 4 not gates

6. 10 2-input and gates, 3 3-input and gates, 1 4-input and gate

7. 1 2-input or gate, 3 4-input or gate

ANALYSIS PROBLEM 1 :

1. ALU is designed so as to perform 4 operations NAND, NOR, XNOR, ADD two 4-bit numbers A and B and output the result

2. 4 to 1 mux is used to select the operation that we need to do using 2 bit select line input

3. Separate AND, OR, XOR module has been designed that uses a splitter to split the 4 bit input into individual 1 bit inputs and then operates on individual inputs to produce the desired result

4. Designed AND, OR, XOR are then used with a not gate to get the desired output on the selected operation through mux

5. 4 FULL ADDERS are used to add two 4-bit numbers A and B to get their addition and separate module has been created named ADD

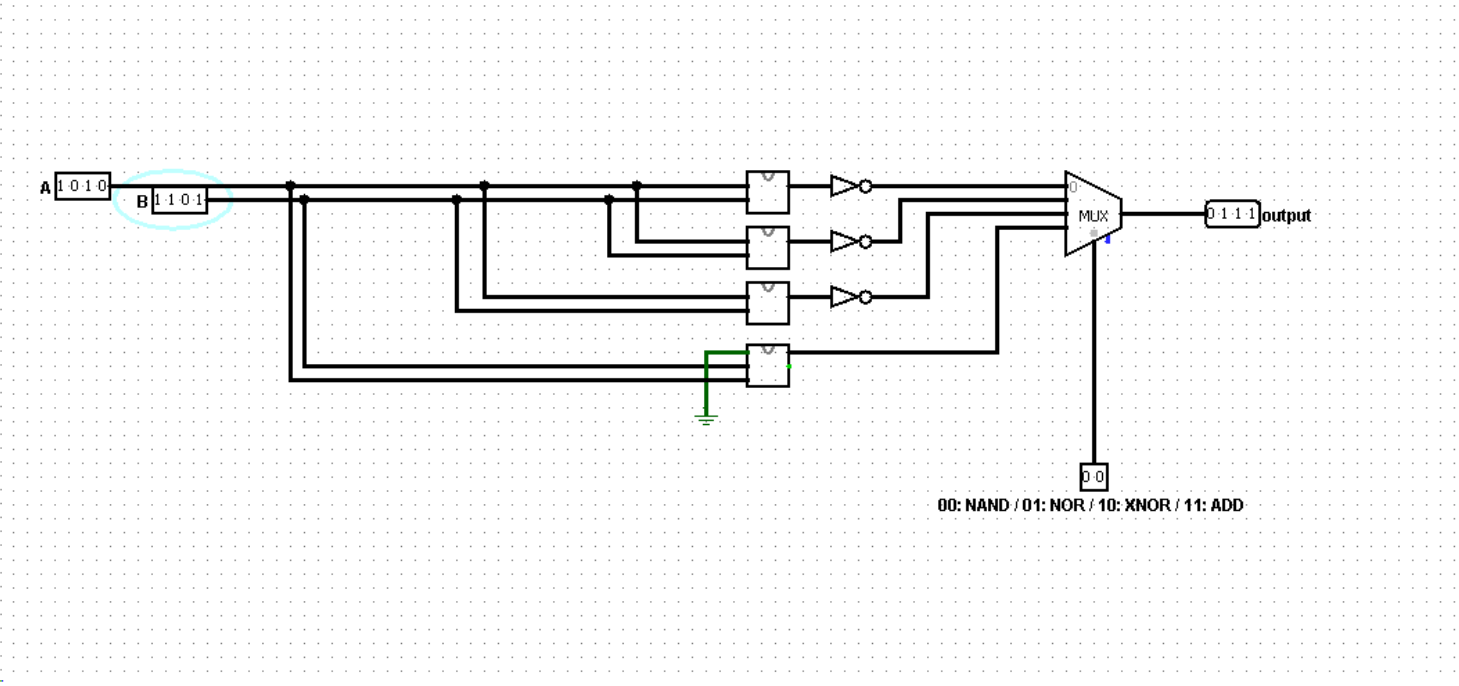
RESULT AND DEMO PROBLEM 1 :

1. **Nand**

   A = 1010

   B = 1101

   Output = 0111

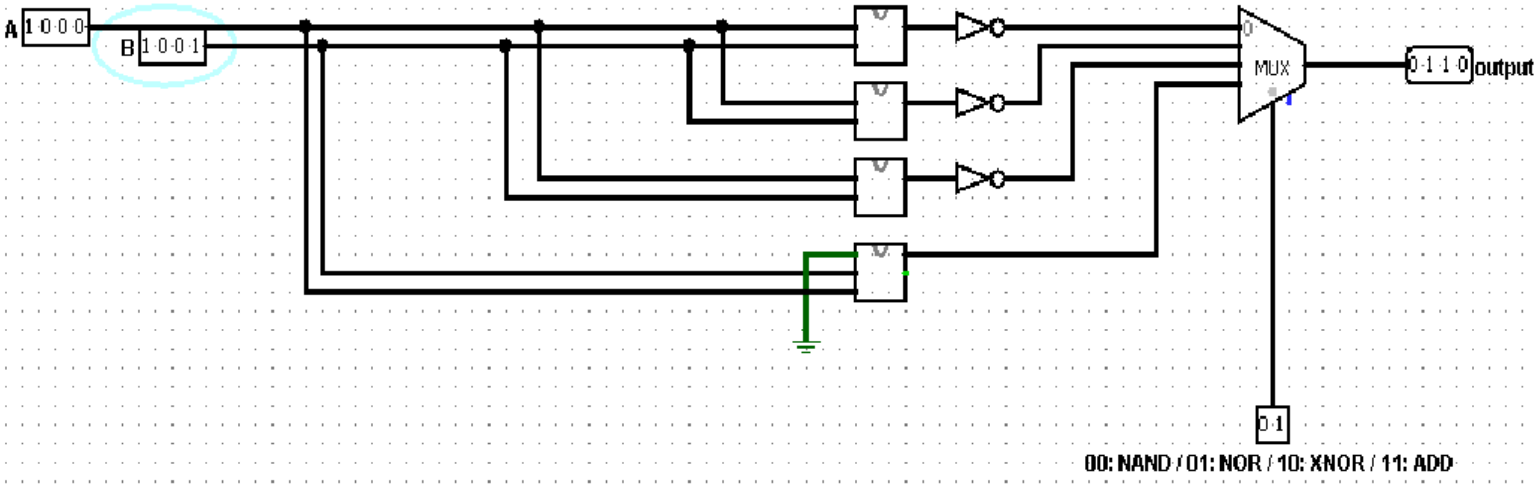


2. **Nor**

A = 0010

B = 1011

Output = 0111

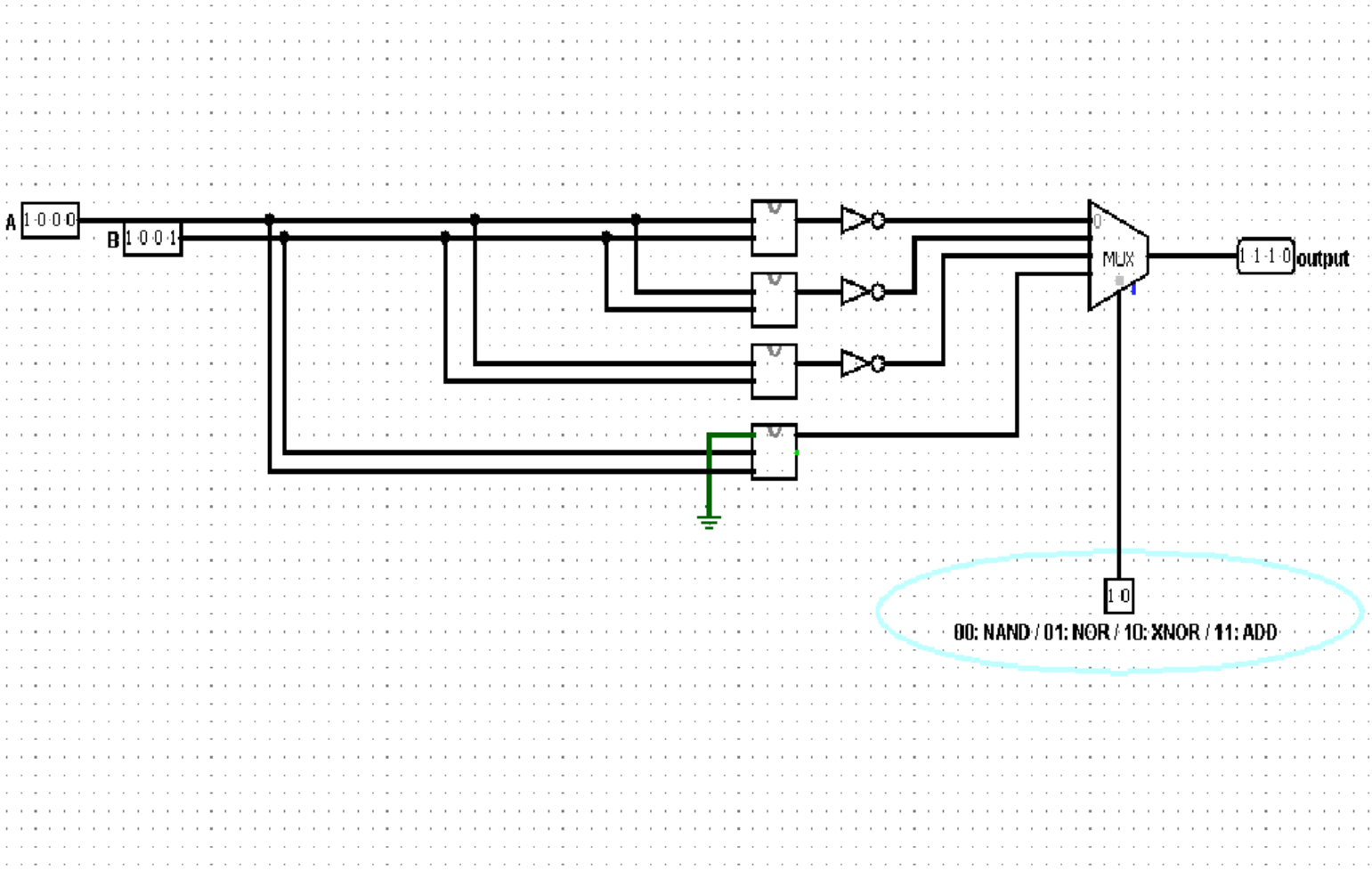


4. **Xnor**

   A = 1000

   B = 1001

   Output = 1110

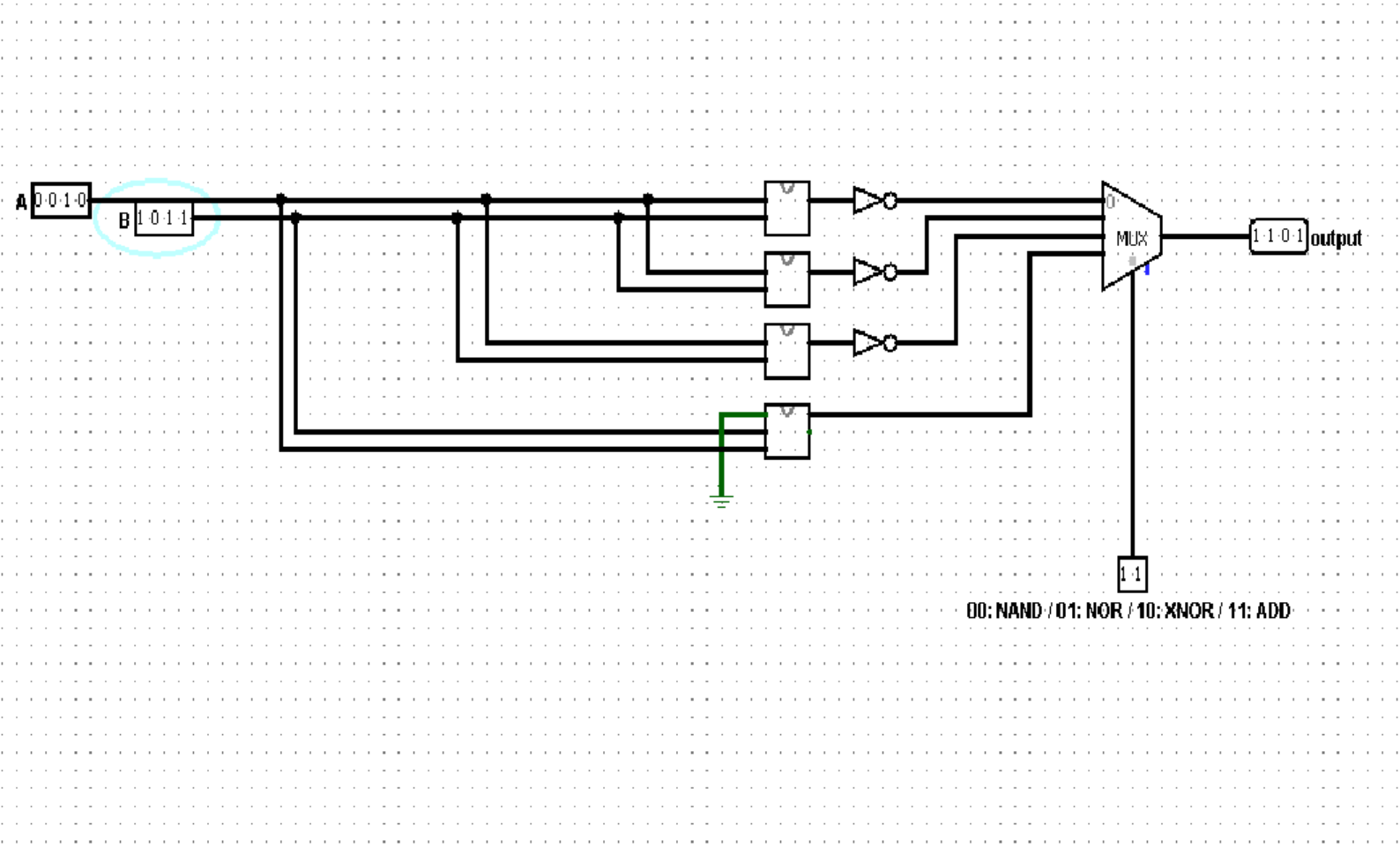


4. **Add**

   A = 0010

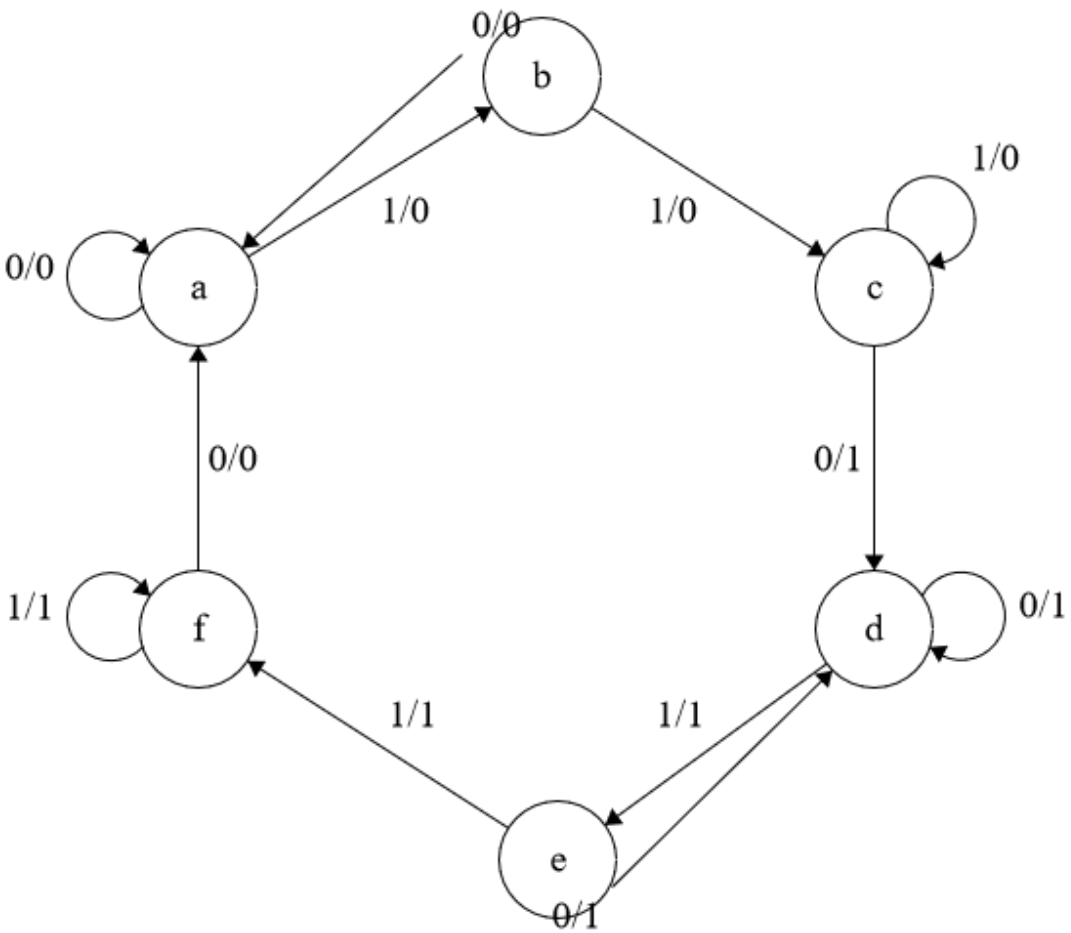
   B = 1011

   Output = 1101



ANALYSIS PROBLEM 2 :

FSM for the given problem is :



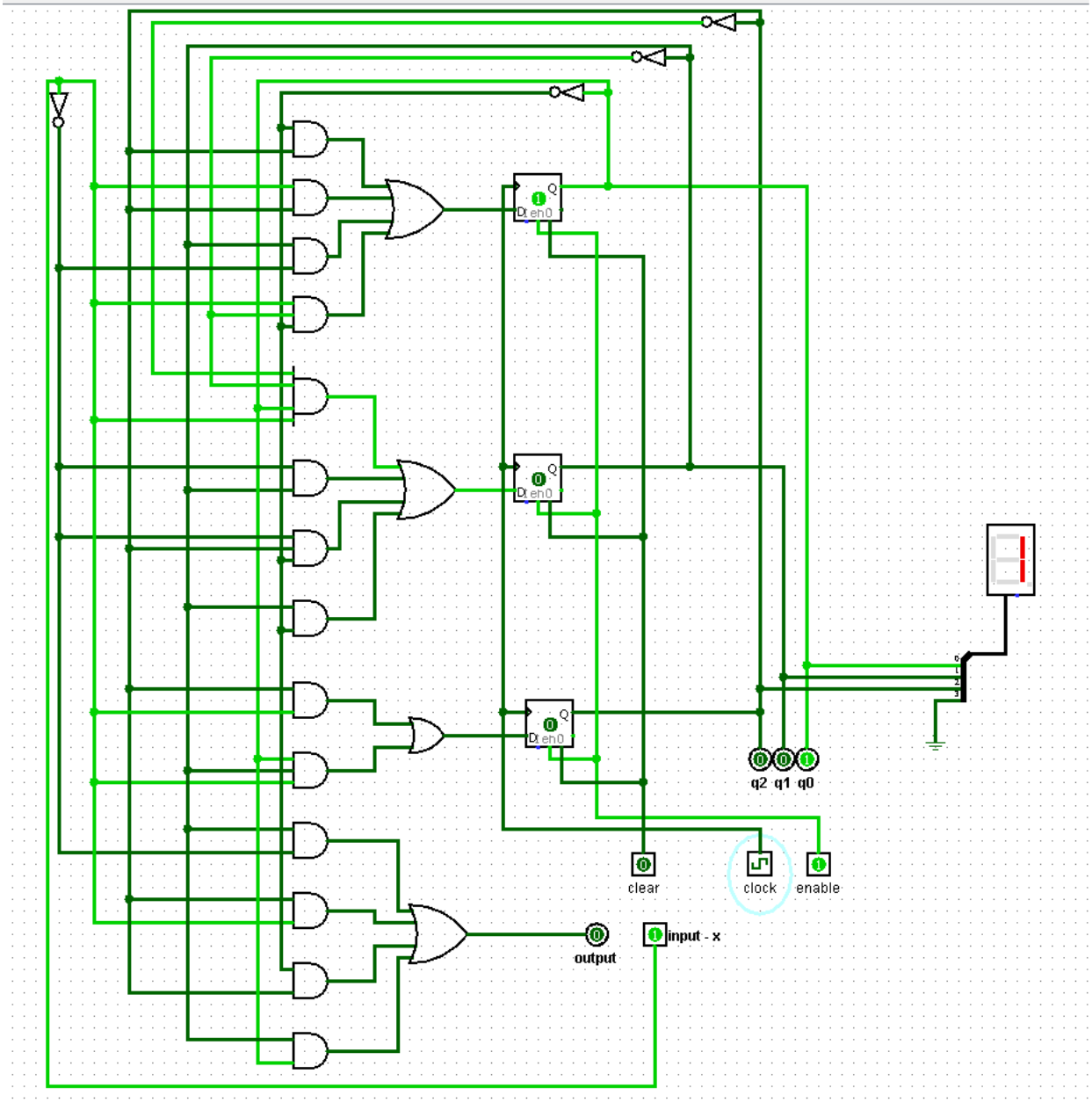
State Encoding Table and Truth Table:

State Name	State Encoding
a	000
b	001
c	010
d	011
e	100
f	101

Present State			Input	Next State			D flipflop inputs			Output
P2	P1	P0	X	Q2	Q1	Q0	D2	D1	D0	
0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0
0	0	1	1	0	1	0	0	1	0	0
0	1	0	0	0	1	1	0	1	1	1
0	1	0	1	0	1	0	0	1	0	0
0	1	1	0	0	1	1	0	1	1	1
0	1	1	1	1	0	0	1	0	0	1
1	0	0	0	0	1	1	0	1	1	1
1	0	0	1	1	0	1	1	0	1	1
1	0	1	0	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1	0	1	1

RESULT AND DEMO PROBLEM 2 :

Present state = 000
Input = 1
Next state = 001
Output = 0

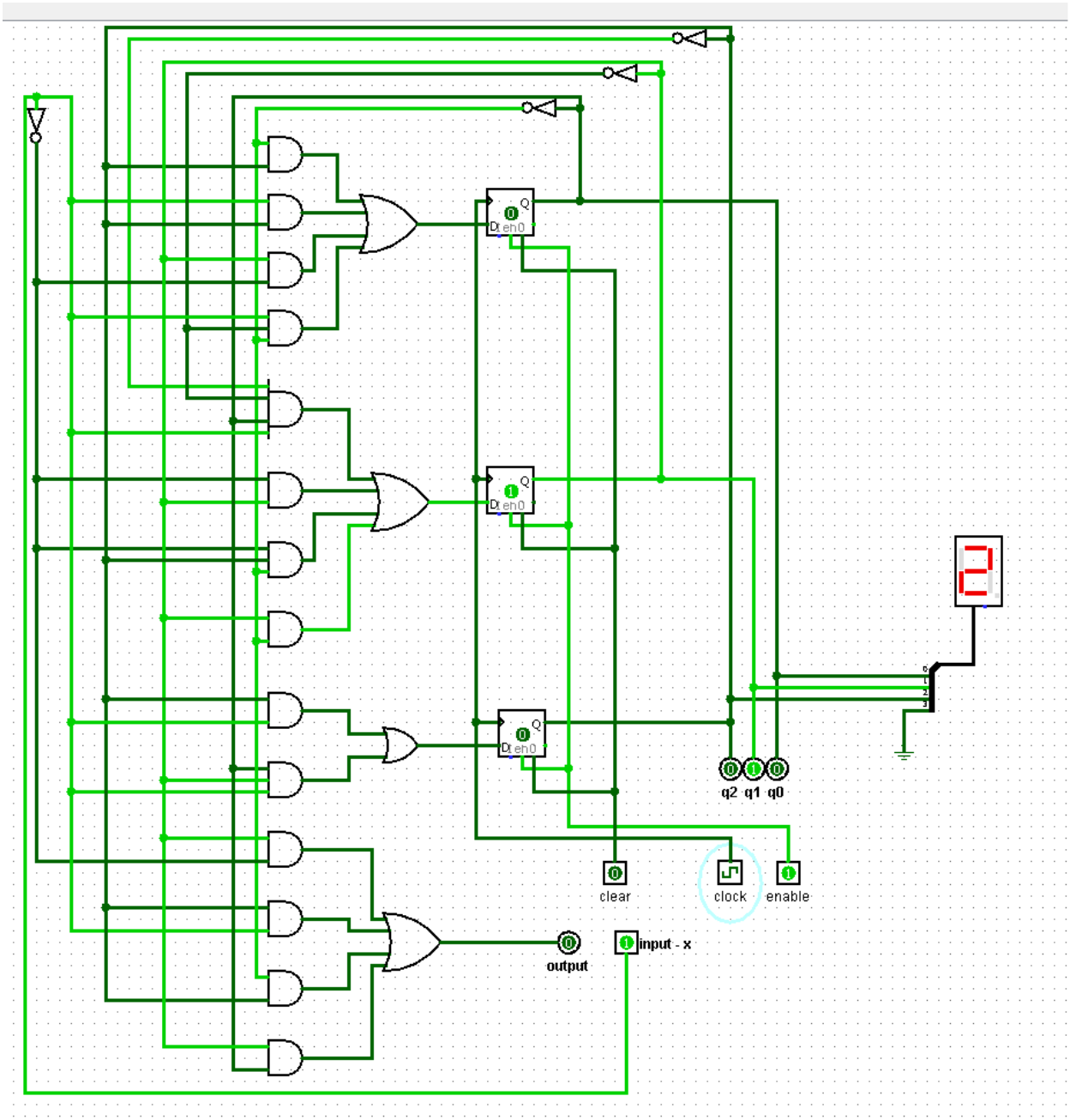


Present state = 000

Input = 1

Next state = 001

Output = 0



Present state = 010

Input = 0

Next state = 011

Output = 1