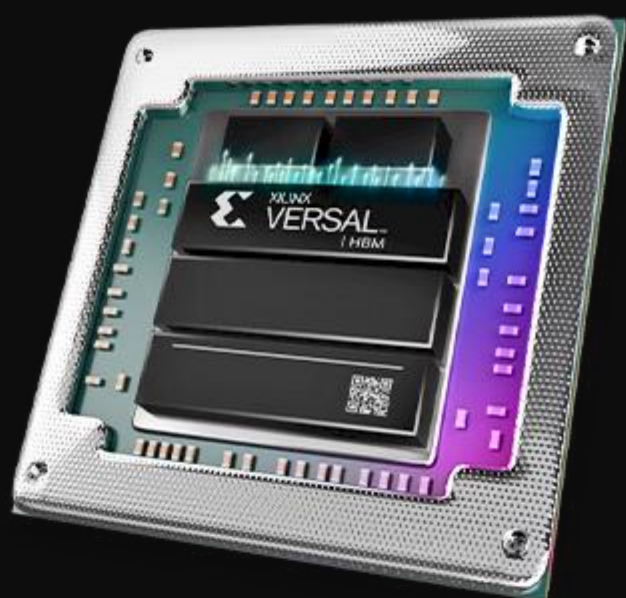
EC204

Digital System Design Lab

Lab – 5

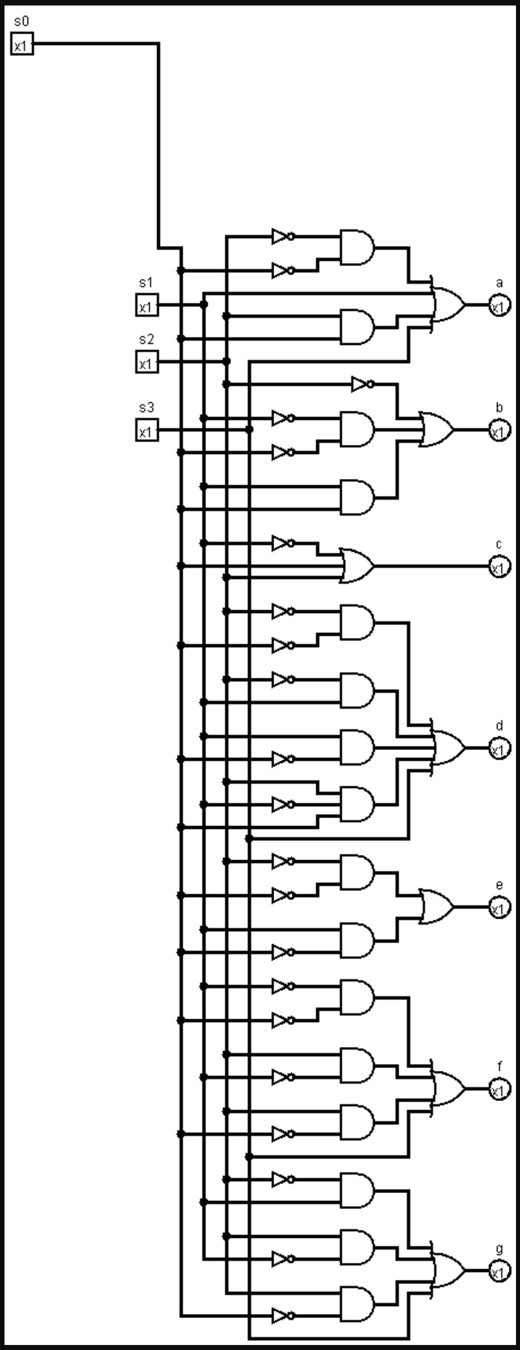


Utkarsh R Mahajan

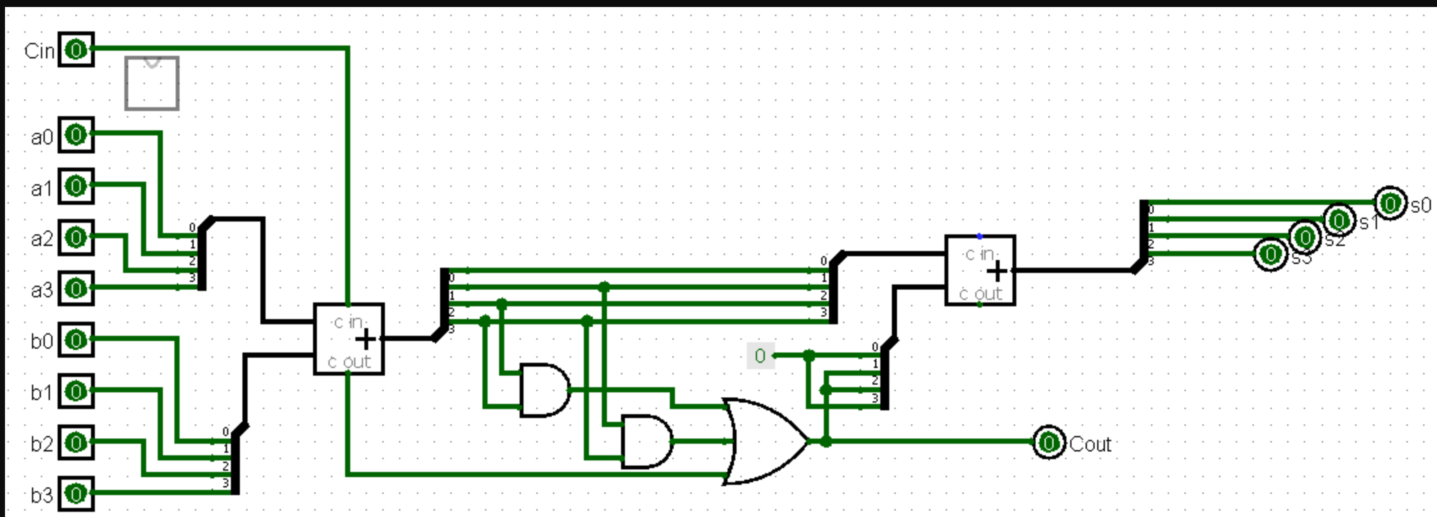
201EC164

1] Design a circuit that will perform addition of two single digit BCD numbers and display the output on seven segment display. Use the Adder available in Arithmetic library in logisim.

`->` BCD to 7segment Decoder for outputting on display.



Single digit BCD adder Circuit: 1st 4 bit adder adds 2 4 bit inputs a and b then checks for invalid bcd digits and then 2nd 4 bit adder adds 6(in binary) to correct it if a invalid bcd digit is found.



Calculated truth table, (0 indicates LSB and 3 indicates MSB in naming):

Combinational Analysis

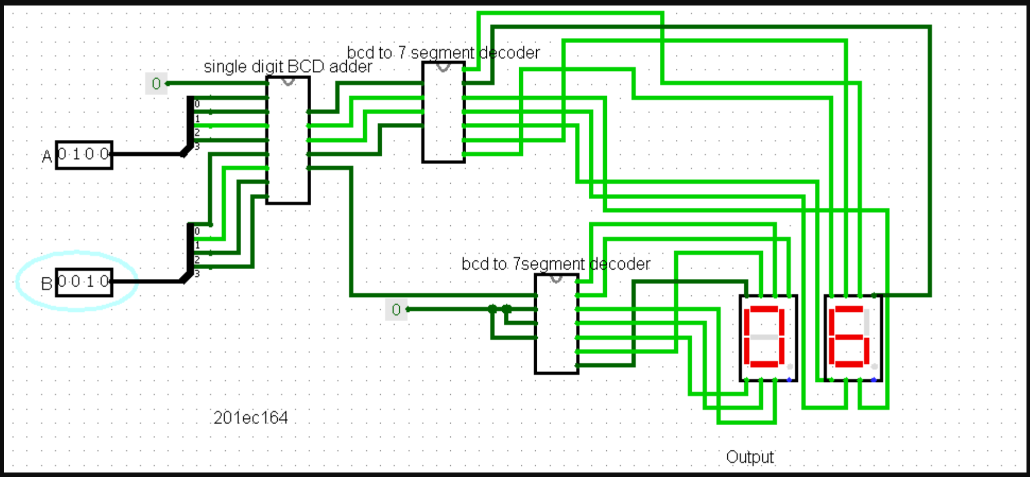
File Edit Project Simulate Window Help

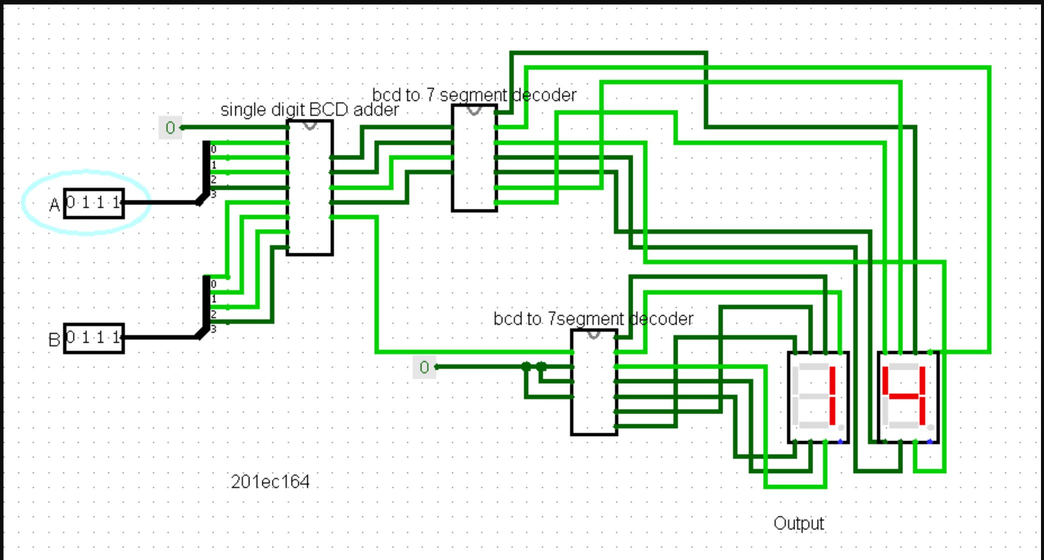
Inputs Outputs Table Expression Minimized

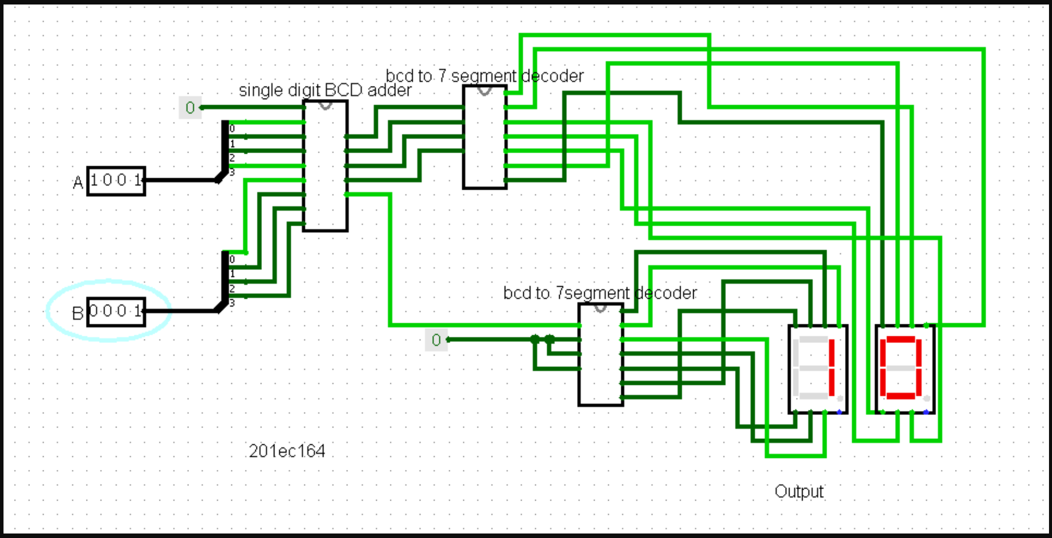
Cin	a0	a1	a2	a3	b0	b1	b2	b3	s0	s1	s2	s3	Cout
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	1	1	0	1	0	0	1
0	0	0	0	0	0	1	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	1	0	0	0	0	1
0	0	0	0	0	0	1	1	0	0	1	1	0	0
0	0	0	0	0	0	1	1	1	0	0	1	0	1
0	0	0	0	0	1	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	1	1	0	0	1	0
0	0	0	0	0	1	0	1	0	1	0	1	0	0
0	0	0	0	0	1	0	1	1	1	1	0	0	1
0	0	0	0	0	1	1	0	0	1	1	0	0	0

Build Circuit

Displaying results on 7 segment display using the previously created decoder and adder:



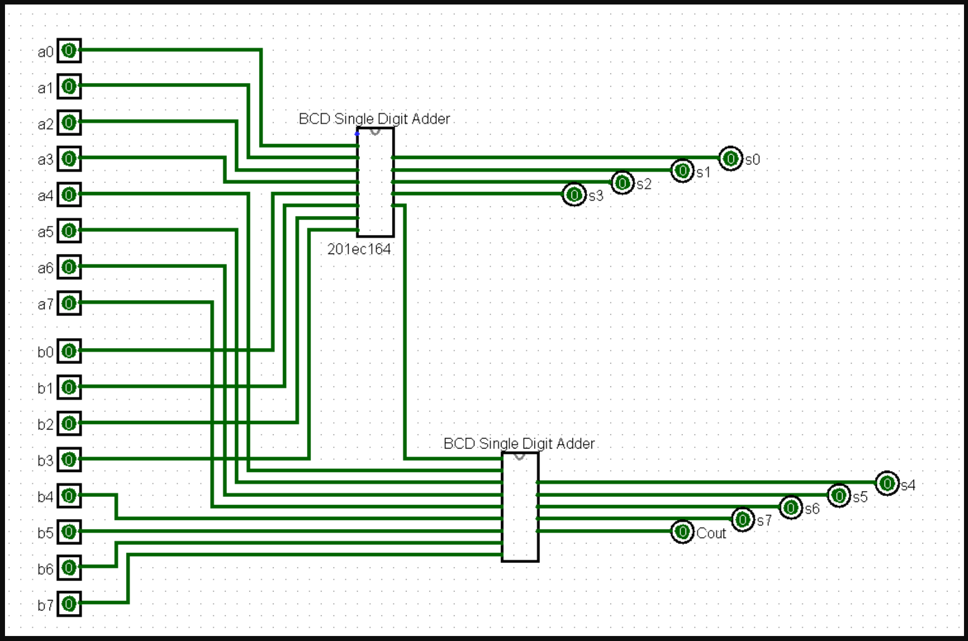




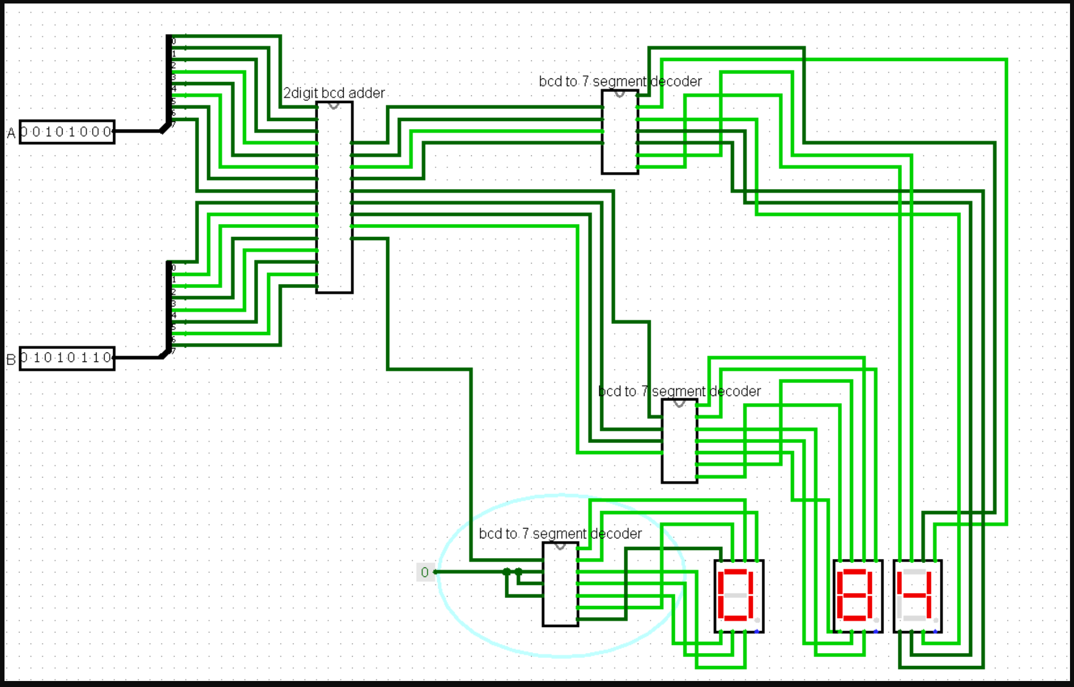
2]Extend it to addition of two 2 digit BCD numbers. Input in the range 00 to 99.

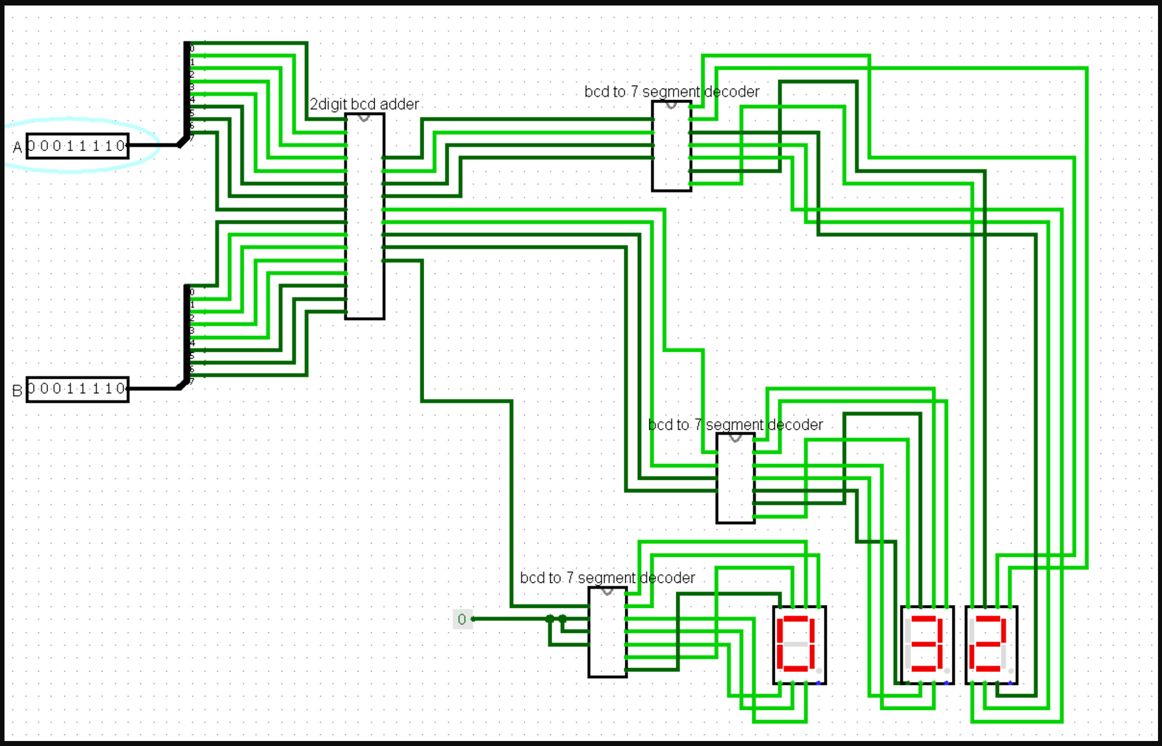
- > Using bcd to 7 segment decoder and bcd single digit adder from previous question's solution. We use 2 single digit bcd adders, connect 4bit lsb's to first 1st adder and msb's to 2nd adder and Cout of first adder as Cin for 2nd adder.

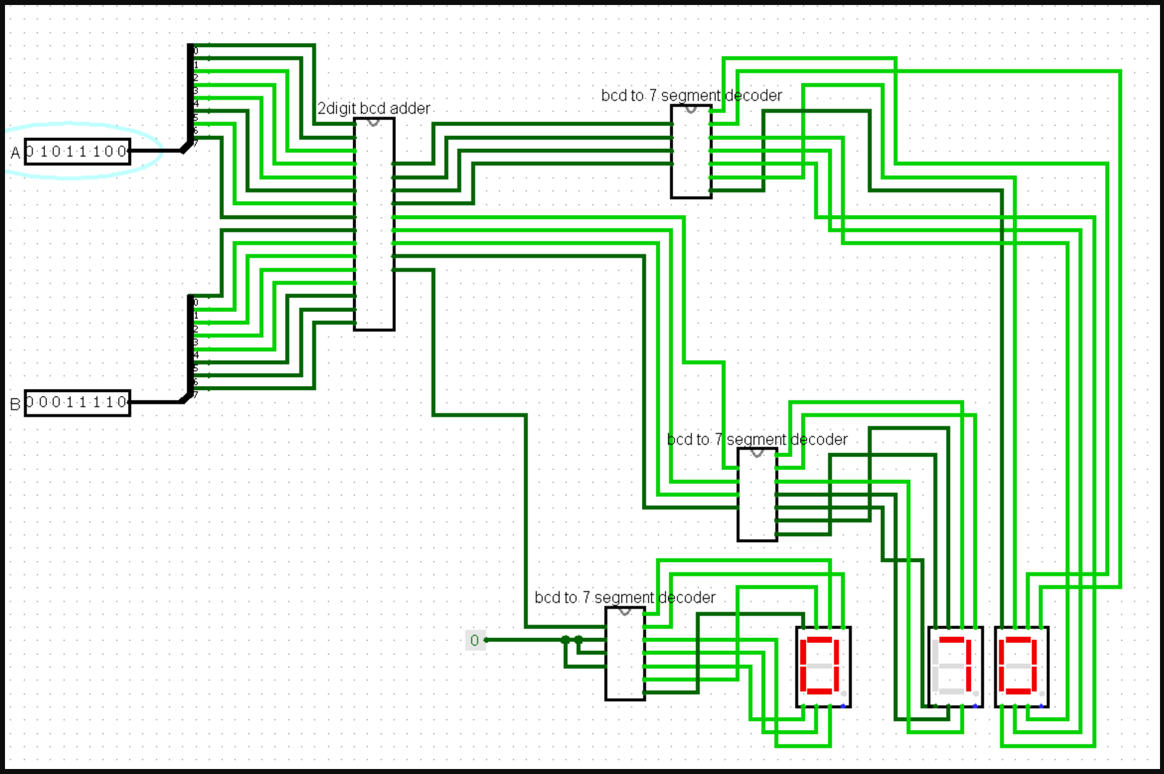
We can get the following circuit.



Displaying results on 7 segment display using the previously created decoder and adder:




2digit bcd adder
bcd to 7 segment decoder
bcd to 7 segment decoder
bcd to 7 segment decoder
A
0 0 0 1 1 1 1 0
B
0 0 0 1 1 1 1 0
0

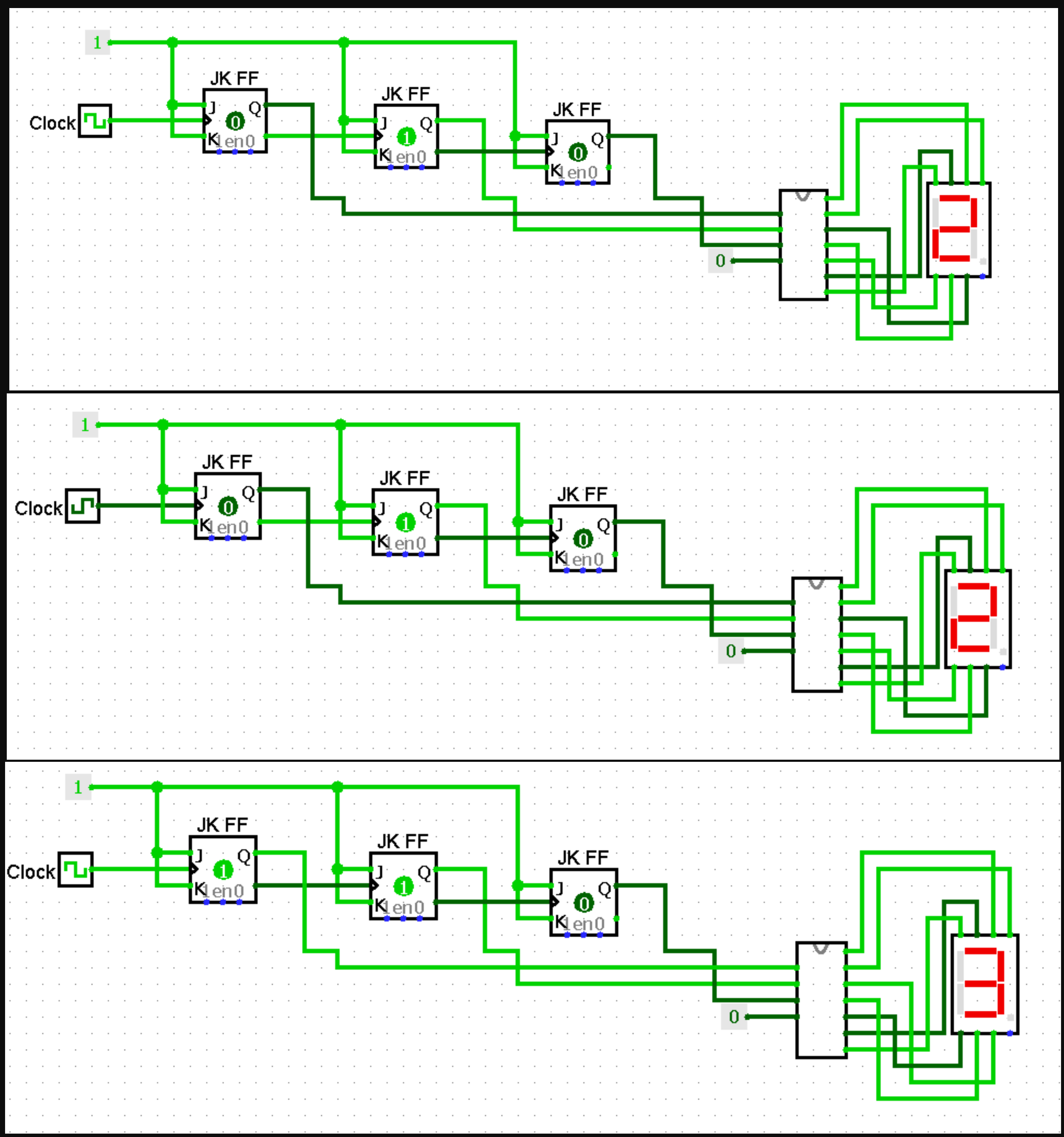

2digit bcd adder
bcd to 7 segment decoder
bcd to 7 segment decoder
bcd to 7 segment decoder
A
0 1 0 1 1 1 0 0
B
0 0 0 1 1 1 1 0
0

3](a) Design a modulo 8 ripple up counter using JKFF

- > since we need to design a mod 8 counter, we will need 3 JK flip flops (minimum n which satisfies 8 <= 2^n).

As for ripple counter we connect the input clock to the 1st JK flipflop whose output Q' are connected to input clock of next flipflop until the end. J and K will be connected to a constant positive 1 source so that the output value changes every trigger. Q of each Flipflop will give us the desired counter output.

We Will get the following circuit in Logisim:

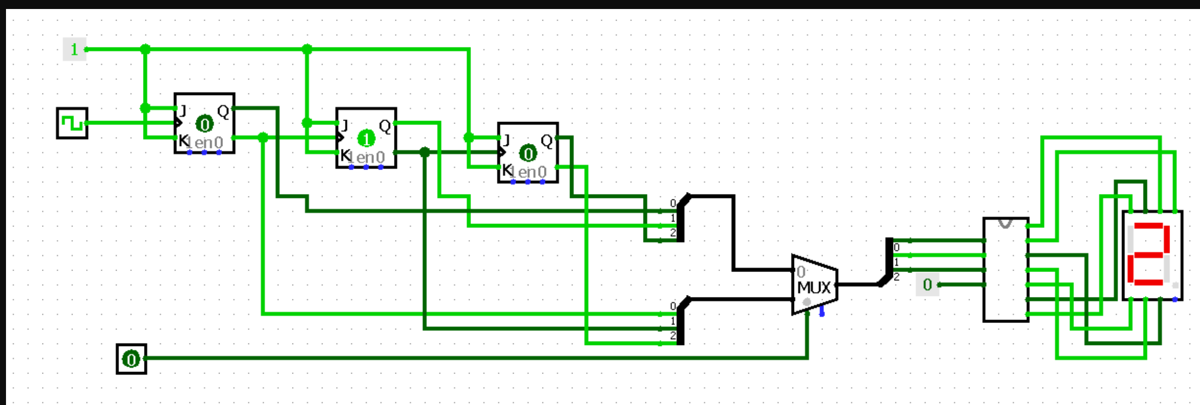


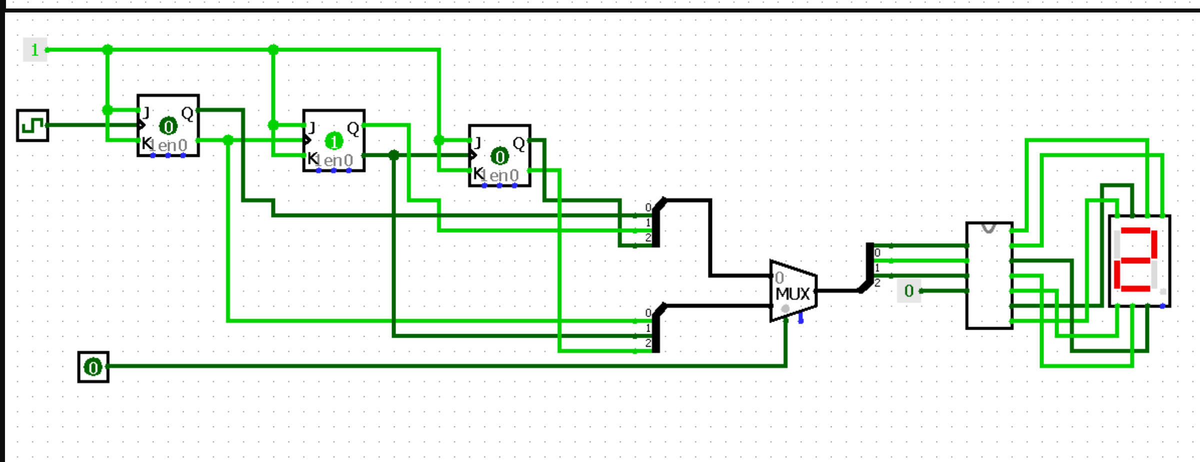
3](b) modify it to a modulo-8 ripple up-down counter

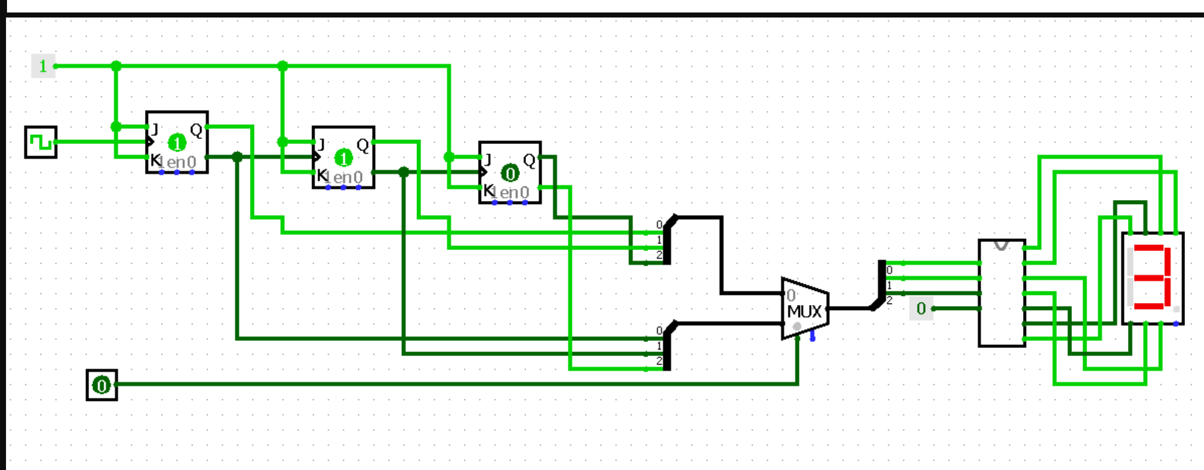
- > For a down counter, we just need to show Q’ outputs of each flip flop instead of Q. It will start with highest msb and decrement.

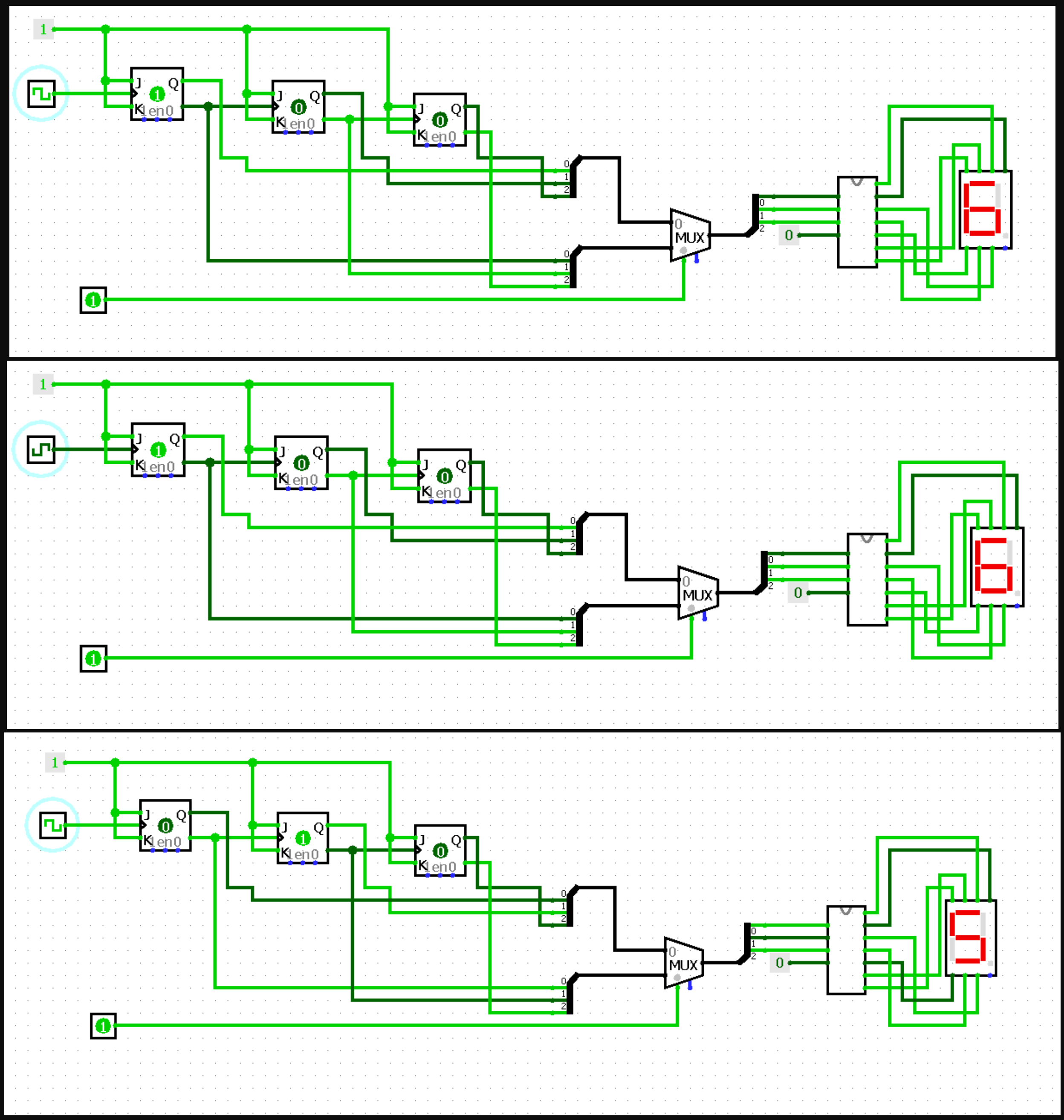
for converting it into a up-down counter, we just need to add a 2:1 mux for which when s0 will be low, the up counter should be shown by giving the output Q of each flip flops while when s0 will be high we can give output of Q’ giving up down counter.

Circuit in Logisim:









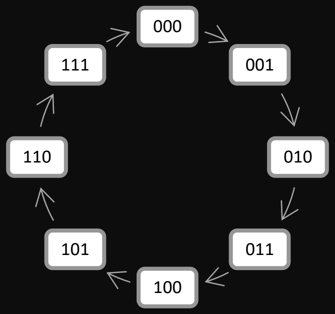


Table:

S_0 (mux)	Operation
0	Count Up
1	Count Down

Table for Up counter:

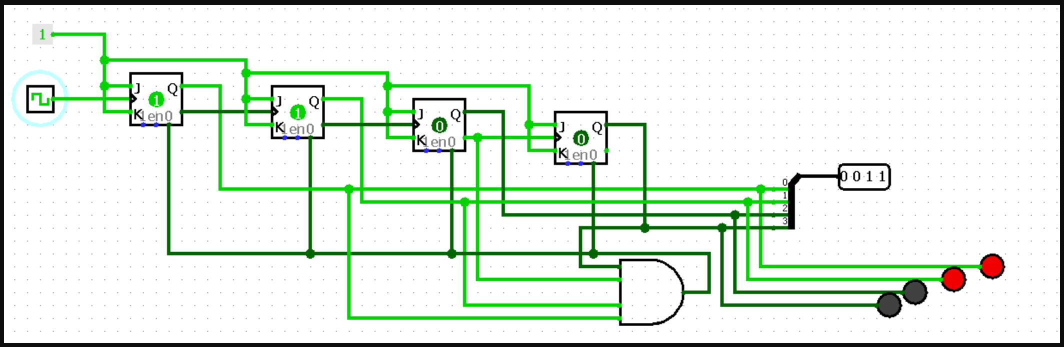
Qn	Qn+1
0 0 0	0 0 1
0 0 1	0 1 0
0 1 0	0 1 1
0 1 1	1 0 0
1 0 0	1 0 1
1 0 1	1 1 0
1 1 0	1 1 1
1 1 1	0 0 0

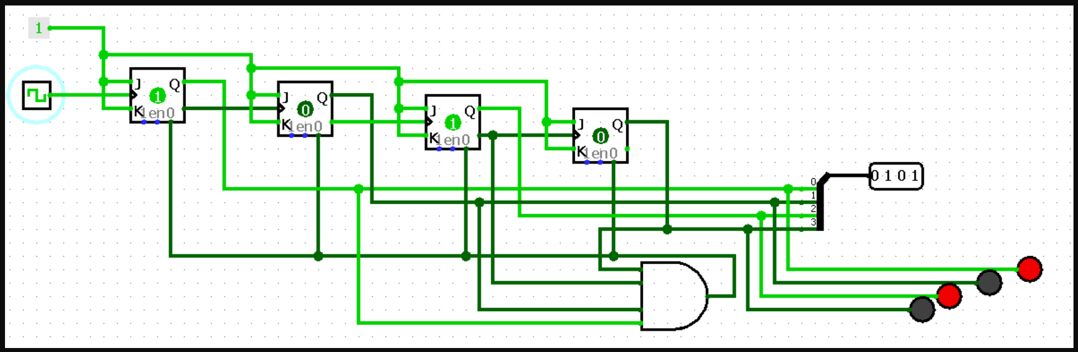
4] Design a modulo-11 ripple counter using JKFF

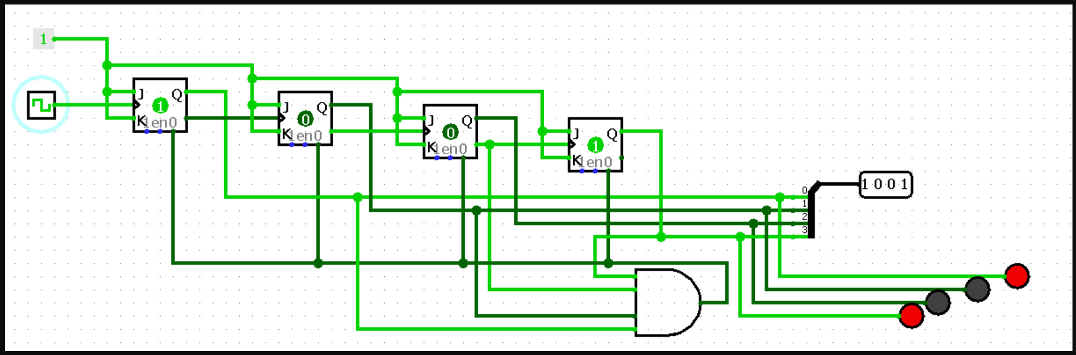
- > similar to the 3rd question, using 4 JK flip flops (minimum n which satisfies 11 <= 2^n). but to make it a mod 11 counter, we use a AND gate which will take inputs Q of flip flop checking for the condition when the output will of the flip flops will be 11 in binary. And use that condition to reset the flip flops to 0.

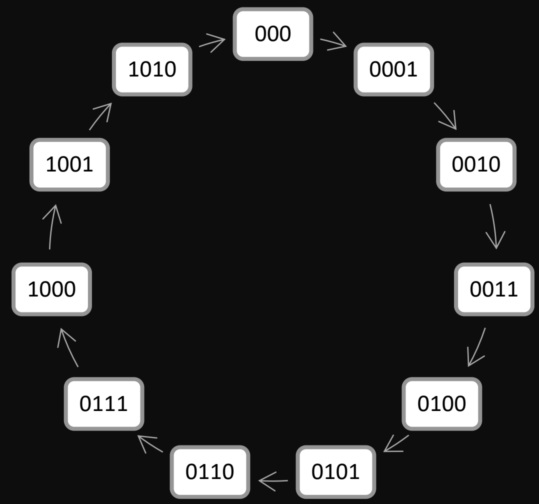
Qn	Qn+1
0 0 0 0	0 0 0 1
0 0 0 1	0 0 1 0
0 0 1 0	0 0 1 1
0 0 1 1	0 1 0 0
0 1 0 0	0 1 0 1
0 1 0 1	0 1 1 0
0 1 1 0	0 1 1 1
0 1 1 1	0 0 0 0
1 0 0 0	1 0 0 1
1 0 0 1	1 0 1 0
1 0 1 0	0 0 0 0

We get the following Results with circuit:


1
J
Q
K
en0
0 0 1 1


1
J
Q
K
en0
0 1 0 1


1
J
Q
K
en0
1 0 0 1

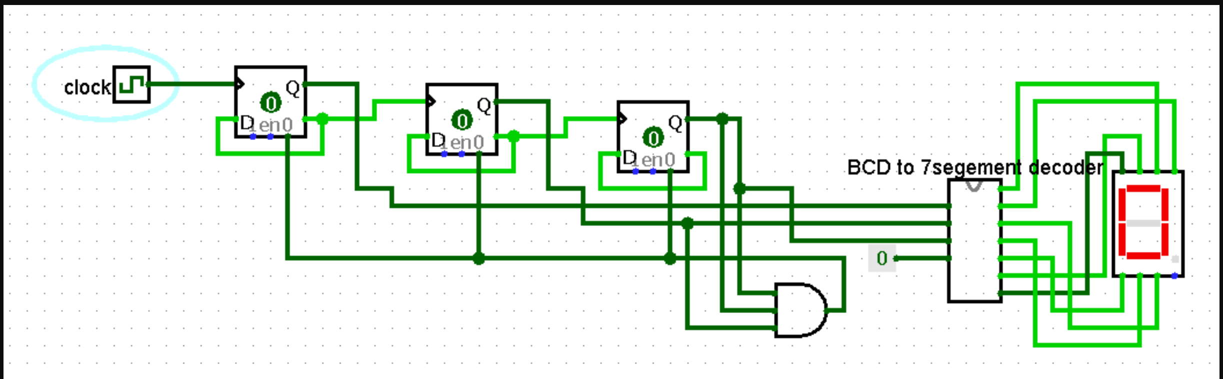

000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010

5]Design a modulo-6 ripple counter using DFF and display the output using 7 segment LED

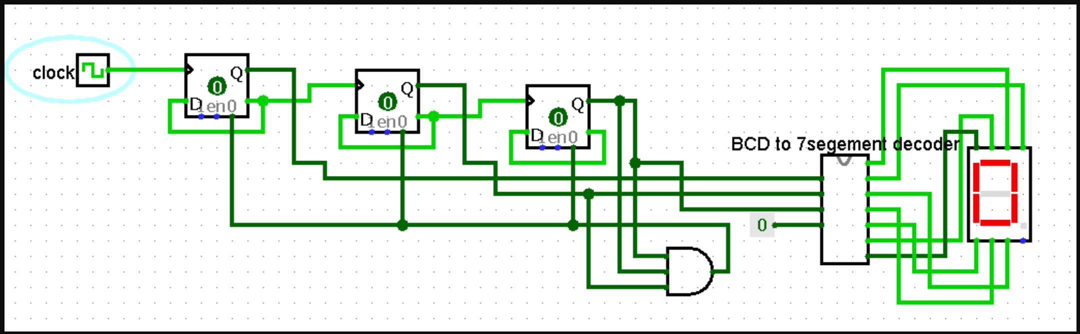
-> we can create a mod 6 ripple counter using 3 D Flip flops. Connecting output Q’ of each flip flop to its input D and connecting input Clock to the first D flip flop. And connecting its outputs Q’ to input clock of consecutive flip flops.

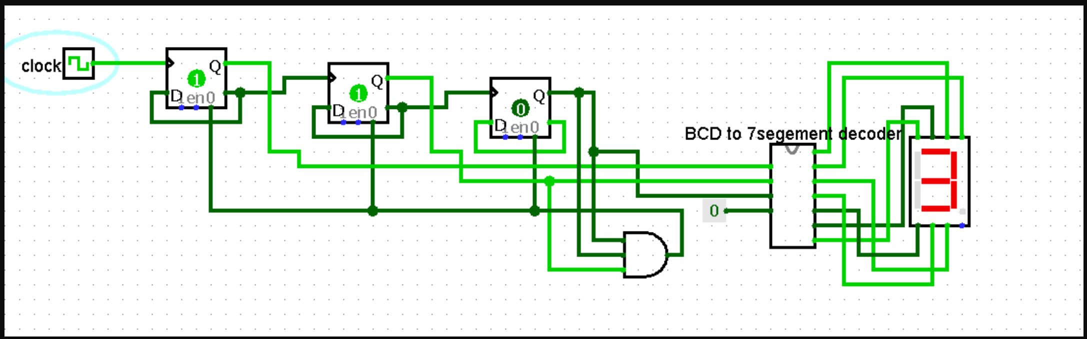
The output will be generated at outputs Q of each D flip flops. These outputs are tested for the value equaling 110(6 in binary) and then resetting all d flip flops by enabling the reset on each flip flops for that value so it outputs back 000.

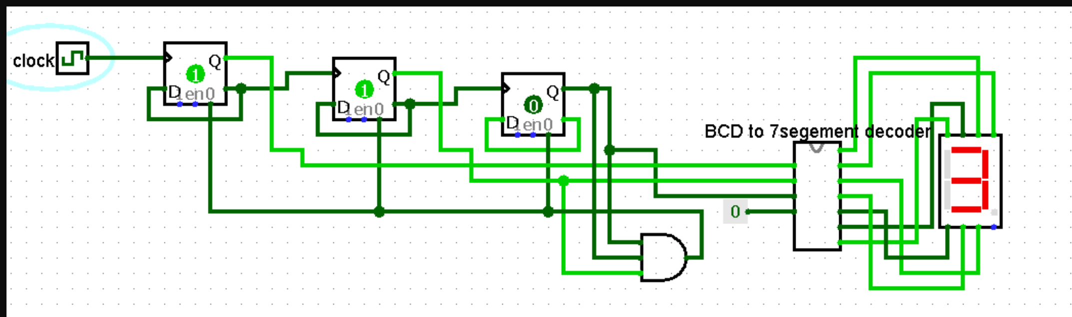
We will obtain the following circuit which is connected to BCD 7 segment decoder and outputted using 7 segment display.



Results:







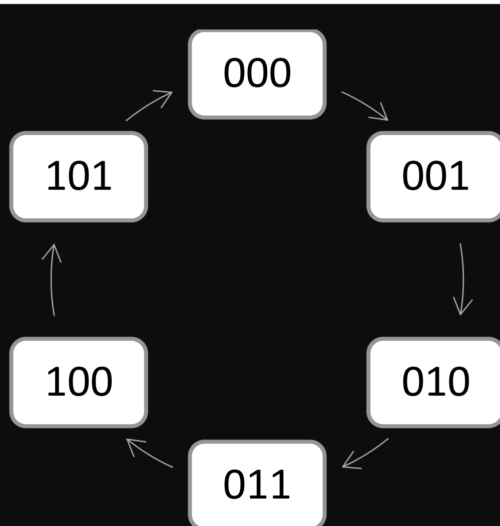


Table:

Qn	Qn+1
0 0 0	0 0 1
0 0 1	0 1 0
0 1 0	0 1 1
0 1 1	1 0 0
1 0 0	1 0 1
1 0 1	1 1 0

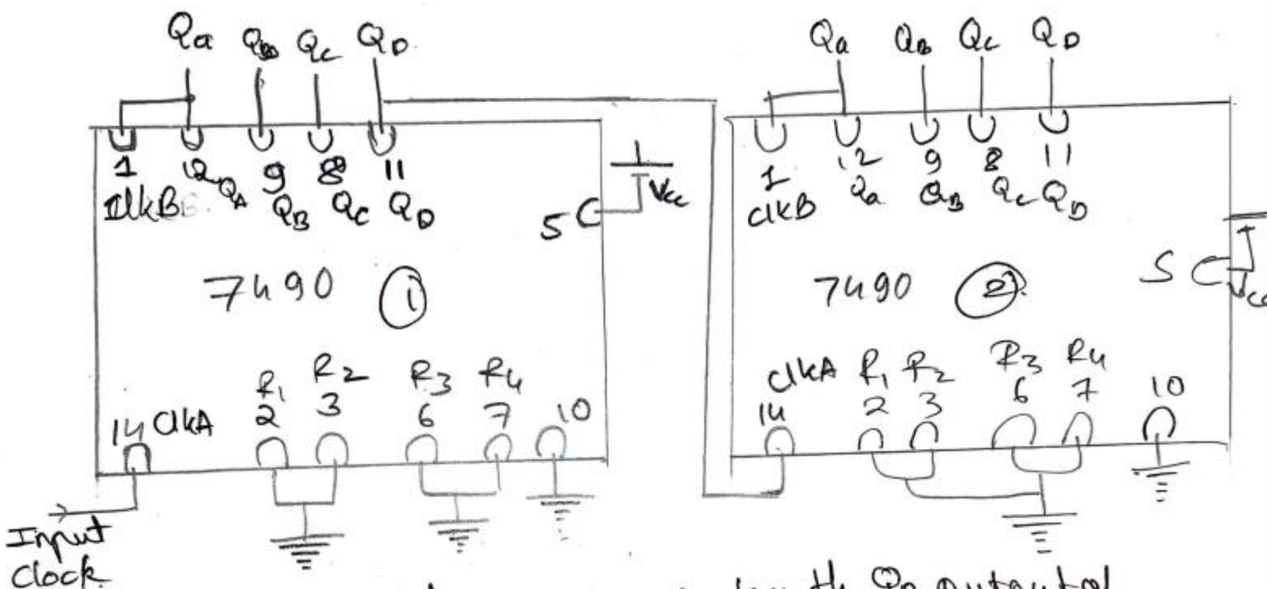
6] Draw the circuit diagram to implement a counter that can count from 00-99 using Decade counters 7490.

- > The Output will be:

Qa Qb Qc Qd (of 1st counter/ units place) Qa Qb Qc Qd (of 2nd counter/ ten's place)

(6) 201EE164

Considering Decade Counter 7490 as negative edge triggered, we can connect 2 7490 IC's together to form a mod 100 counter (0-99).



Input to the 2nd Counter is given by the QD output of 1st Counter which activates it at negative edge.

1st ~~Counter~~ output Qa, Qb, Qc Qd gives the unit's place while the output of 2nd counter gives Ten's place