

Problem Set #3, EE part

Issue date: Nov. 22, 2020; Deadline: 23:59, Nov. 29, 2020

Student Name: _____ Student No.: _____

1. FSM I

- Implement the traffic light example finite state machine, which was introduced in lecture, with Multisim (use the digital clock component in Multisim to generate clock ticks). Change the output encoding into one-hot encoding as follows, and use the red, green, yellow indicating lights to show the result. (20')

Output	Encoding L2:0
green	001
yellow	010
red	100

1. Input definition:

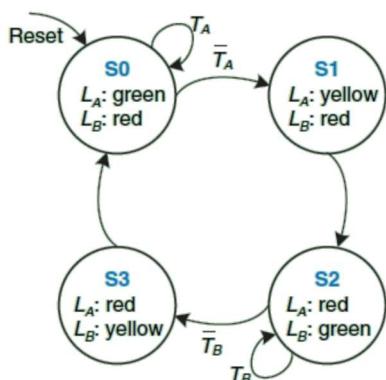
T_A, T_B (1 busy, 0 empty)

2. State definition:

State	LA:LB	Encoding S1:0
S0	green:red	00
S1	yellow:red	01
S2	red:green	10
S3	red:yellow	11

3. State transition:

Current State		Inputs		Next State	
S1	S0	TA	TB	S1'	S0'
0	0	0	X	0	1
0	0	1	X	0	0
0	1	X	X	1	0
1	0	X	0	1	1
1	0	X	1	1	0
1	1	X	X	0	0



4. State-transition equations:

$$S'_0 = \overline{S_0} \overline{S_1} \overline{T_A} + S_1 \overline{S_0} \overline{T_B}$$

$$S'_1 = S_0 \overline{S_1} + S_1 \overline{S_0}$$

5. One-hot output encoding:

S1:0	L_AR:L_AG:L_AY
00	010
01	001
10	100
11	100

$$L_{AR} = S_1$$

$$L_{AG} = \overline{S_0} \overline{S_1}$$

$$L_{AY} = S_0 \overline{S_1}$$

S1:0	L_BR:L_BG:L_BY
00	100
01	100
10	010
11	001

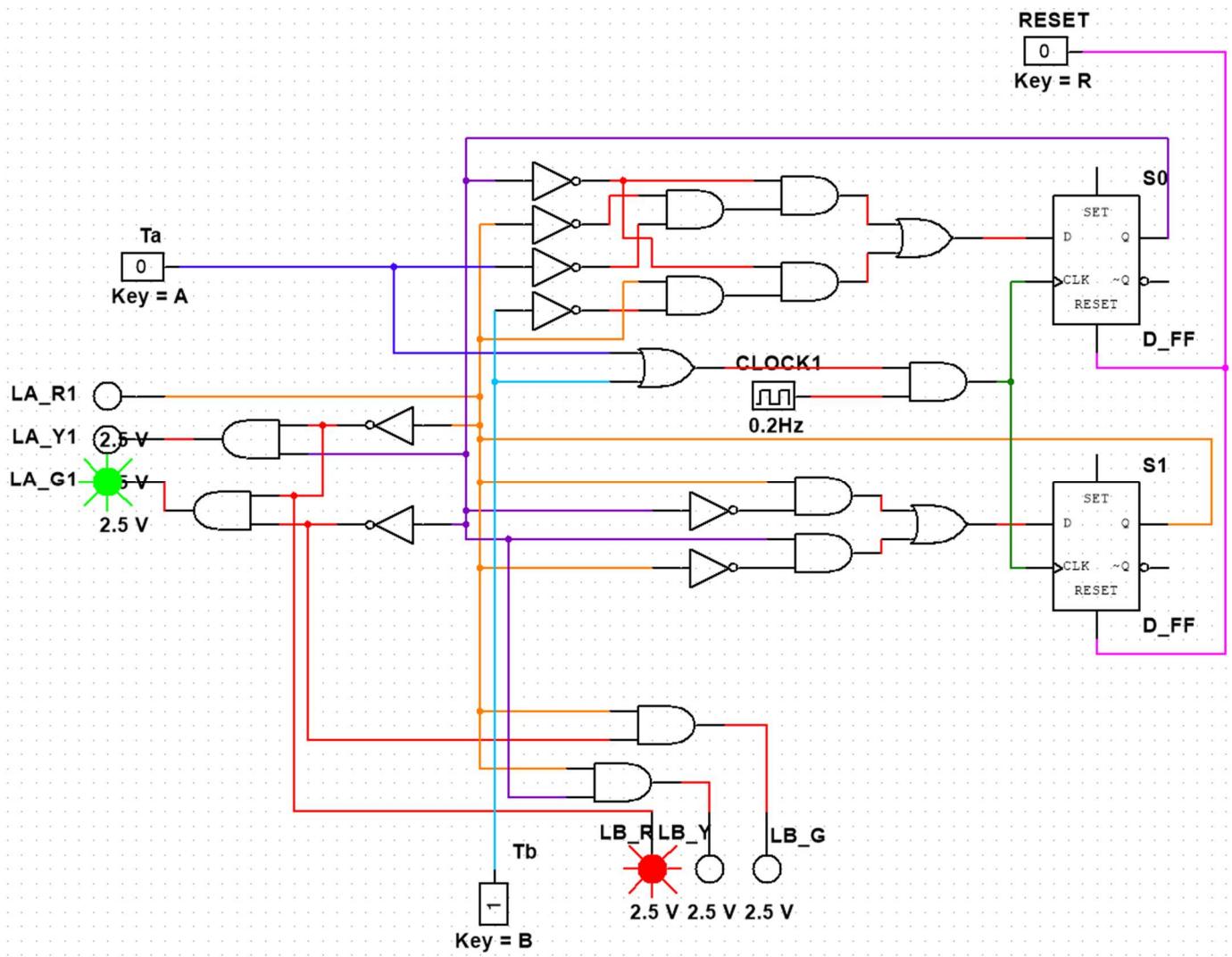
$$L_{BR} = \overline{S_1}$$

$$L_{BG} = \overline{S_0} S_1$$

$$L_{BY} = \overline{S_0} \overline{S_1}$$

6. Design circuit:

(To avoid the mess caused when input TA=0, TB=0, add an OR gate to these two and an AND gate to CLOCK so that when TA=0, TB=0, there would not be any mess.)



- In the real scenario, the lights might be controlled according to time rather than sensors. Design the time-based traffic light, for example: red 27 sec, yellow 3 sec, green 30 sec. (20')

1. Input definition:(based on time)

Take **60 seconds as a cycle**, we can design (actually a one-hot encoding):

T1: 0s-27s 1 other 0

T2: 27s-30s 1 other 0

T3: 30s-57s 1 other 0

T4: 57s-60s 1 other 0

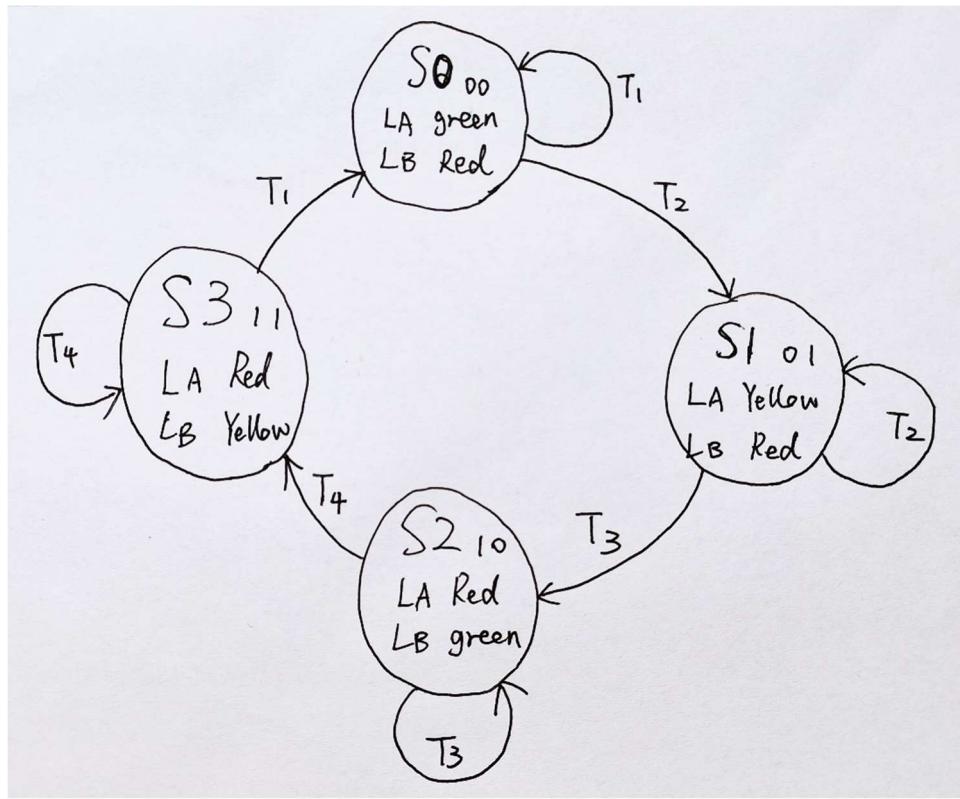
Input\Time	0s~27s	27s~30s	30s~57s	57s~60s
T1	1	0	0	0
T2	0	1	0	0
T3	0	0	1	0
T4	0	0	0	1

2. State definition:

State	LA:LB	Encoding S1:0
S0	green:red	00
S1	yellow:red	01
S2	red:green	10
S3	red:yellow	11

3. State transition:

Current State		Inputs					Next State	
S1	S0	T1	T2	T3	T4	S1'	S0'	
0	0	1	X	X	X	0	0	
0	0	X	1	X	X	0	1	
0	1	X	1	X	X	0	1	
0	1	X	X	1	X	1	0	
1	0	X	X	1	X	1	0	
1	0	X	X	X	1	1	1	
1	1	X	X	X	1	1	1	
1	1	1	X	X	X	0	0	



4. State-transition equations:

$$\begin{aligned} S'_1 &= S_0 \bar{S}_1 T_3 + \bar{S}_0 S_1 T_3 + \bar{S}_0 S_1 T_4 + S_0 S_1 T_4 = S_0 \bar{S}_1 T_3 + \bar{S}_0 S_1 T_3 + S_1 T_4 \\ S'_0 &= \bar{S}_0 \bar{S}_1 T_2 + S_0 \bar{S}_1 T_2 + S_0 \bar{S}_1 T_4 + S_0 S_1 T_4 = \bar{S}_1 T_2 + S_1 T_4 \end{aligned}$$

5. One-hot output encoding (the same as above 1.1):

S1:0	L_AR:L_AG:L_AY
00	010
01	001
10	100
11	100

$$L_{AR} = S_1$$

$$L_{AG} = \bar{S}_0 \bar{S}_1$$

$$L_{AY} = S_0 \bar{S}_1$$

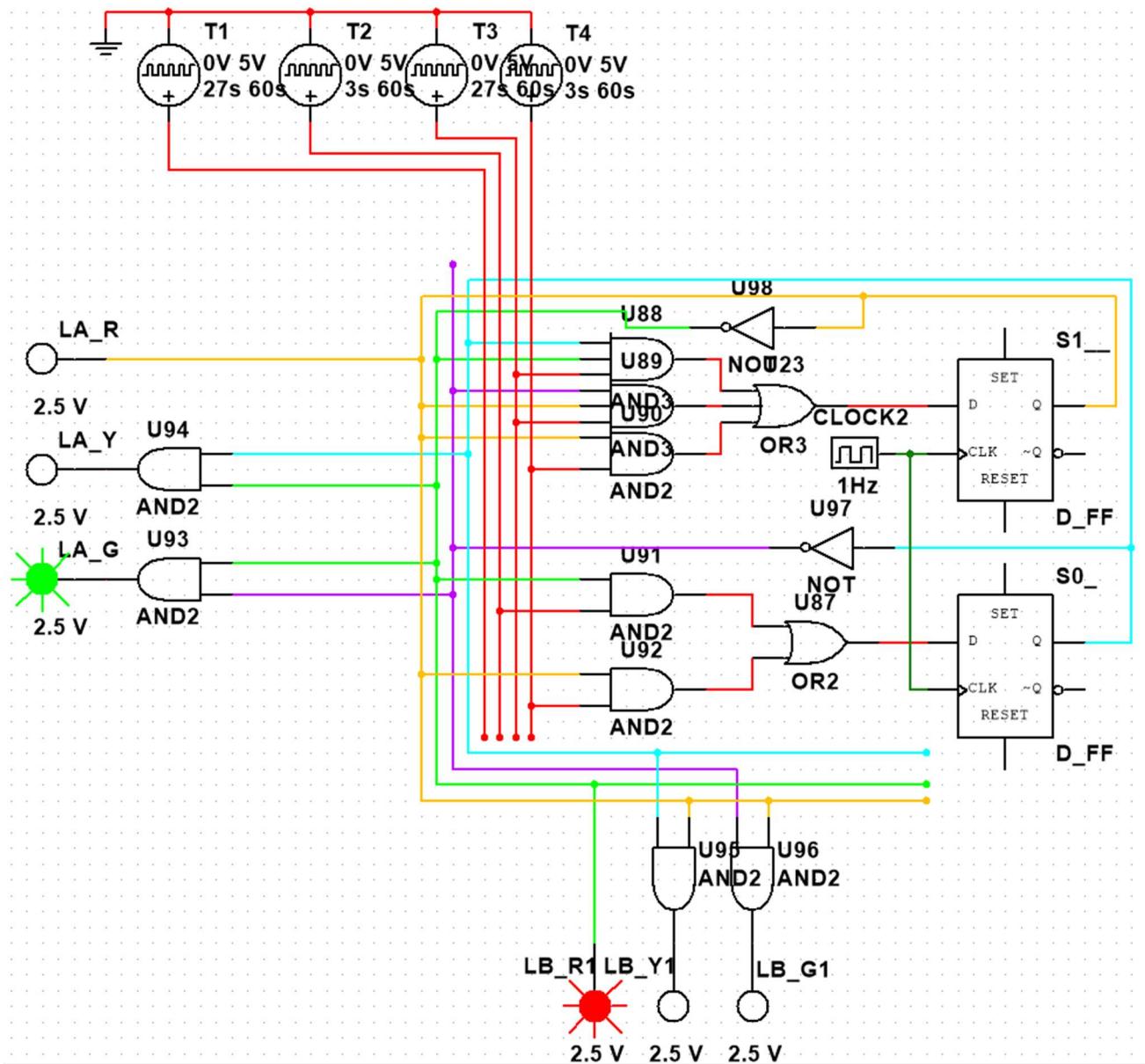
S1:0	L_BR:L_BG:L_BY
00	100
01	100
10	010
11	001

$$L_{BR} = \bar{S}_1$$

$$L_{BG} = \bar{S}_0 S_1$$

$$L_{BY} = \bar{S}_0 \bar{S}_1$$

6. Design circuit:



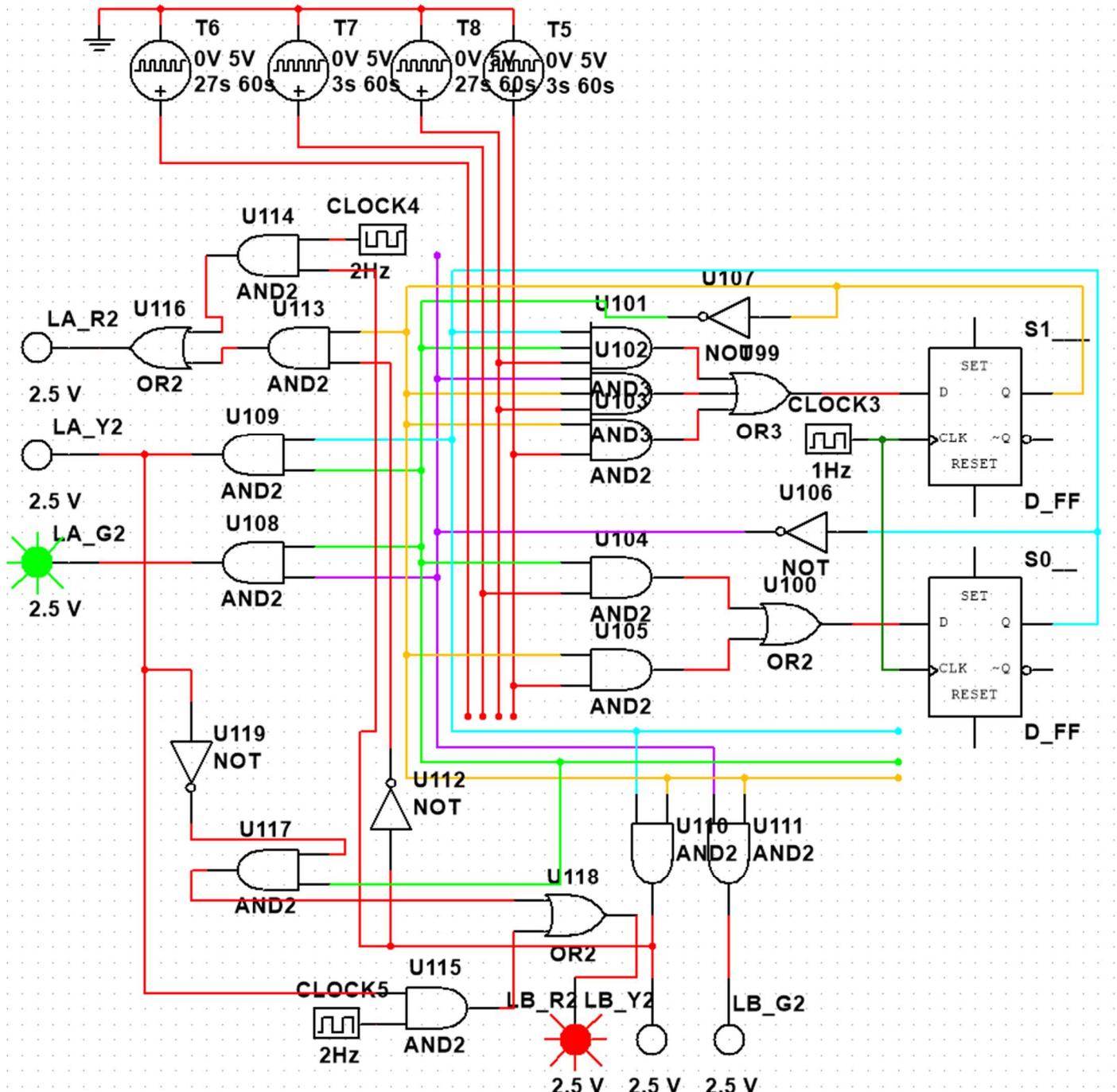
- The red light twinkles during the last 3 seconds. Can you design such a twinkling design? (bonus +15')
Notice that when the red light twinkles, the yellow light of the other group is on.

Let CLK be the clock signal for twinkling light. Then

$$L_{AR} = S_1 \overline{L_{BY}} + CLK \cdot L_{BY}$$

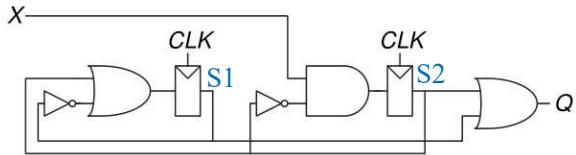
$$L_{BR} = \overline{S_1} \overline{L_{AY}} + CLK \cdot L_{AY}$$

Then we can design circuit according to this.



2. FSM II

Analyze the FSM shown as follows. Write the state transition and output tables and sketch the state transition diagram. Describe in words what the FSM does. (20')



Let the output of the first (left) D-FF be S_1 , the second (right) one be S_2 .

Define states:

State	$S_1:S_2$
S_0	00
S_1	01
S_2	10
S_3	11

Then according to the circuit, we have the state transition equations:

$$Q = S_1 + S_2$$

$$S'_2 = \overline{S}_2 X$$

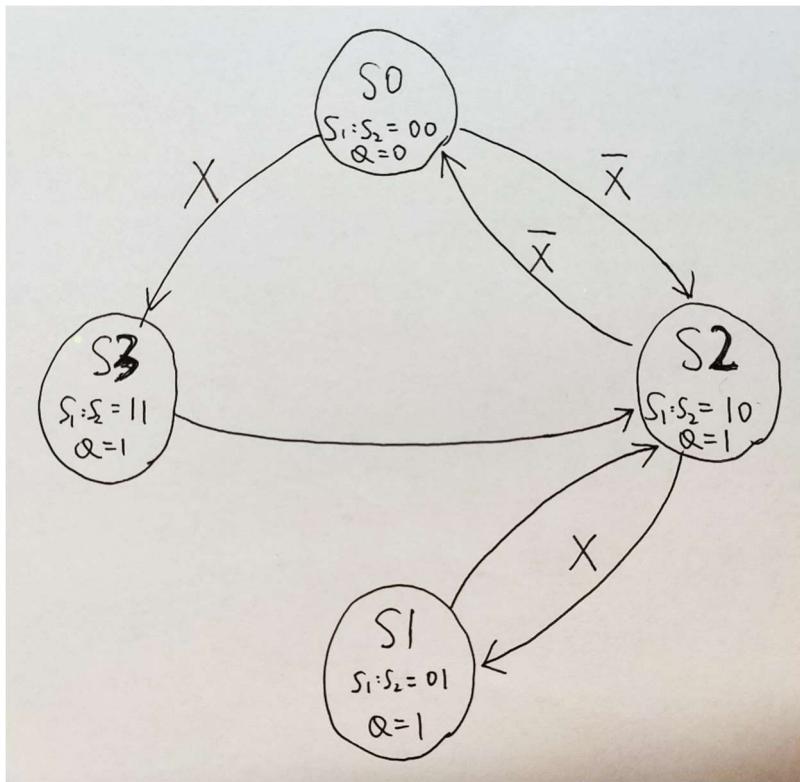
$$S'_1 = \overline{S}_1 + S_2$$

$$Q' = S'_1 + S'_2 = \overline{S}_1 + S_2 + \overline{S}_2 X$$

Then write the state transition table:

State	Q	S_1	S_2	X	S_1'	S_2'	Q'
S_0	0	0	0	0	1	0	1
S_0	0	0	0	1	1	1	1
S_1	1	0	1	X	1	0	1
S_2	1	1	0	0	0	0	0
S_2	1	1	0	1	0	1	1
S_3	1	1	1	X	1	0	1

Then we can draw the state transition diagram.



The FSM changes its state according to its current state and the input X.

When in state S0:

If the input X is 1, it will change into state S3 on the next rising edge of the clock.

If the input X is 0, it will change into state S2 on the next rising edge of the clock.

When in state S1:

It will always change into state S2 on the next rising edge of the clock.

When in state S2:

If the input X is 1, it will change into state S1 on the next rising edge of the clock.

If the input X is 0, it will change into state S0 on the next rising edge of the clock.

When in state S3:

It will always change into state S2 on the next rising edge of the clock.

3. 2 bits easy ALU (25')

Implement a 4 bits adder using the basic logic gate (NOT, AND, OR, NOR, NAND, etc.) with multisim. The ALU pins include: four input bits X₁, X₀ and Y₁, Y₀; two output bits Z₁, Z₀; one carry bit C; one control bit Ctrl

- Each ALU implements one arithmetic and one logic functions, according to the last digit of your student number

Last digit	Function
0, 1	Addition (<u>ctrl=0</u>) & AND (<u>ctrl=1</u>)
2, 3	Addition (<u>ctrl=0</u>) & OR (<u>ctrl=1</u>)
4, 5	Addition (<u>ctrl=0</u>) & XOR (<u>ctrl=1</u>)
6, 7	Addition (<u>ctrl=0</u>) & NAND (<u>ctrl=1</u>)
8, 9	Addition (<u>ctrl=0</u>) & NOR (<u>ctrl=1</u>)

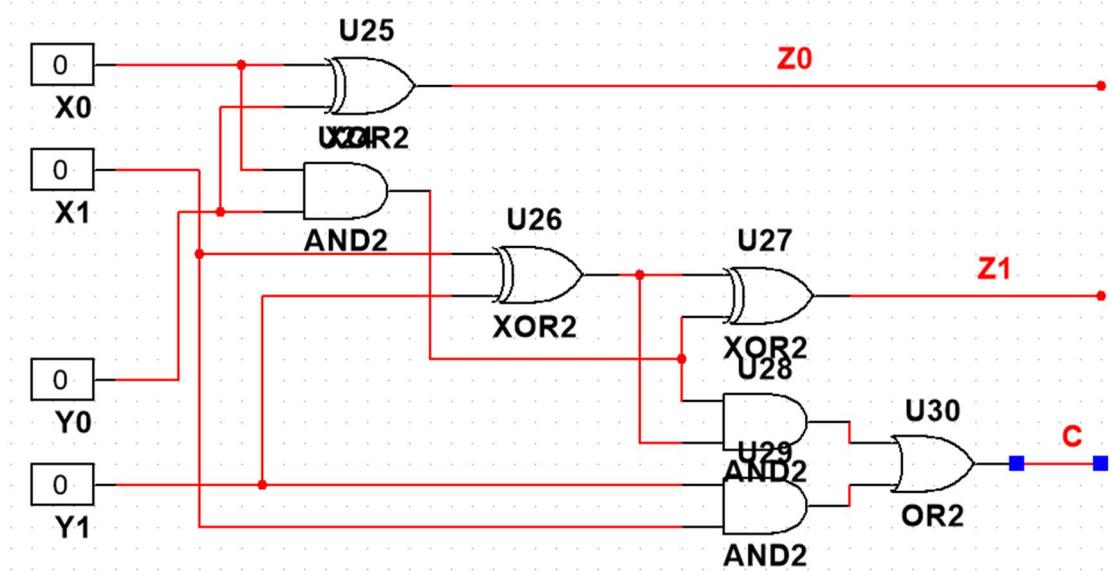
STUDENT NUMBER: *****4 -> XOR gate

First, consider the addition operation.

$$\begin{array}{r}
 & X_1 & X_0 \\
 & + & \\
 Y_1 & & Y_0 \\
 \hline
 C_2 & X_1 + Y_1 + C_1 & X_0 \oplus Y_0
 \end{array}$$

$\therefore Z_0 = X_0 \oplus Y_0$
 $Z_1 = X_1 \oplus Y_1 \oplus X_0 Y_0$
 $C = X_1 Y_1 + (X_1 \oplus Y_1) X_0 Y_0$

Now build the addition part of the circuit according to these equations.



Then consider about when the control bit $\text{Ctrl}=1$ (mark as **logic function** below).

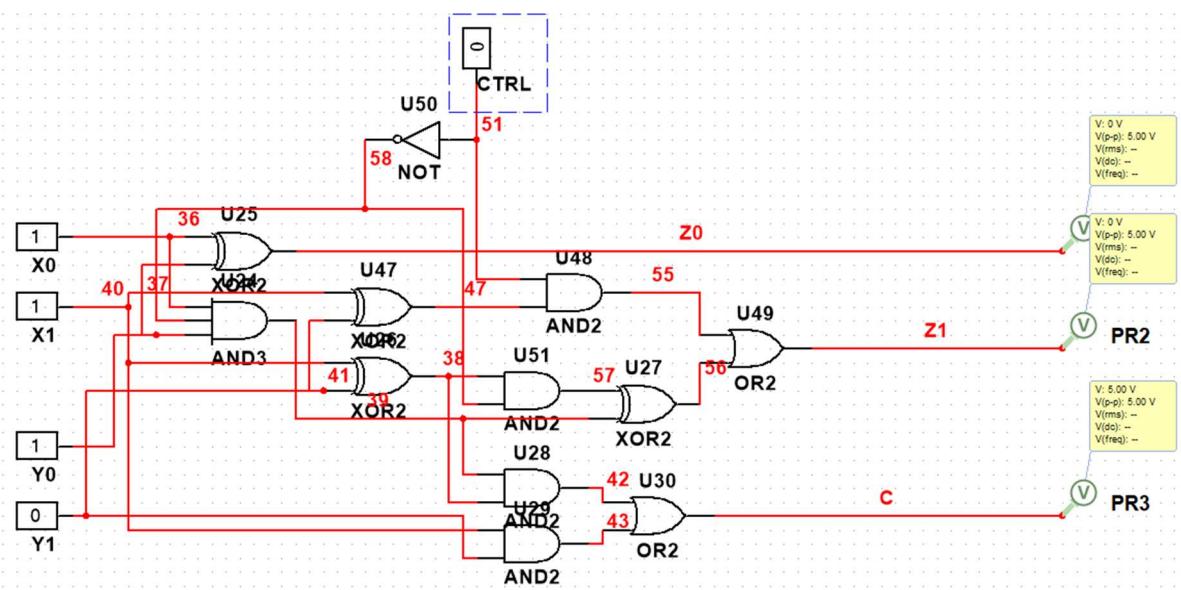
It disables the addition part of the circuit and replace it with logic function (XOR).

Then we can consider disabling the addition part of the circuit by adding AND gates (in the circuit **U24** and **U51**).

Add extra XOR gates (in the circuit **U47**) to realize the logic functions.

To make sure that the logic function part won't affect the addition function part, add AND gates (in the circuit **U48**) to disable logic function parts.

The final circuit is as below.



4. Decoder (15')

- Based on the 4-bit ALU designed in problem 3, design a 7-segment display to show the output of its addition function, the maximum result is $3+3=6$.

To make the logic simpler, choose the **Common cathode** display.

Mark the carry bit C as B_2 , then the result number is $B_2B_1B_0$ (or $B_0+2*B_1+4*B_2$).

The matching of LED and number is:

0	abcdef
1	bc
2	abdeg
3	abedg
4	bcfg
5	acdfg
6	acdefg

Then we have logic equations:

0 1 0 1
 2 3 5 6 0
 $a = \overline{B_0}B_1\overline{B_2} + B_0\overline{B_1}\overline{B_2} + \overline{B_0}\overline{B_1}B_2 + \overline{B_0}B_1B_2 + \overline{B_0}\overline{B_1}\overline{B_2}$
 $b = \overline{B_0}\overline{B_1}\overline{B_2} + \overline{B_0}B_1\overline{B_2} + B_0\overline{B_1}\overline{B_2} + \overline{B_0}\overline{B_1}B_2 + \overline{B_0}B_1B_2$
 $c = \overline{B_0}\overline{B_1}\overline{B_2} + B_0\overline{B_1}\overline{B_2} + \overline{B_0}\overline{B_1}B_2 + \overline{B_0}B_1\overline{B_2} + \overline{B_0}\overline{B_1}\overline{B_2} + B_0\overline{B_1}\overline{B_2}$
 $d = \overline{B_0}\overline{B_1}\overline{B_2} + B_0\overline{B_1}\overline{B_2} + B_0\overline{B_1}B_2 + \overline{B_0}B_1\overline{B_2} + \overline{B_0}\overline{B_1}\overline{B_2}$
 $e = \overline{B_0}\overline{B_1}\overline{B_2} + \overline{B_0}B_1B_2 + \overline{B_0}B_1\overline{B_2}$
 $f = \overline{B_0}B_1B_2 + B_0\overline{B_1}B_2 + \overline{B_0}B_1B_2 + \overline{B_0}B_1\overline{B_2}$
 $g = \overline{B_0}\overline{B_1}\overline{B_2} + B_0\overline{B_1}\overline{B_2} + \overline{B_0}\overline{B_1}B_2 + B_0\overline{B_1}B_2 + \overline{B_0}B_1\overline{B_2}$

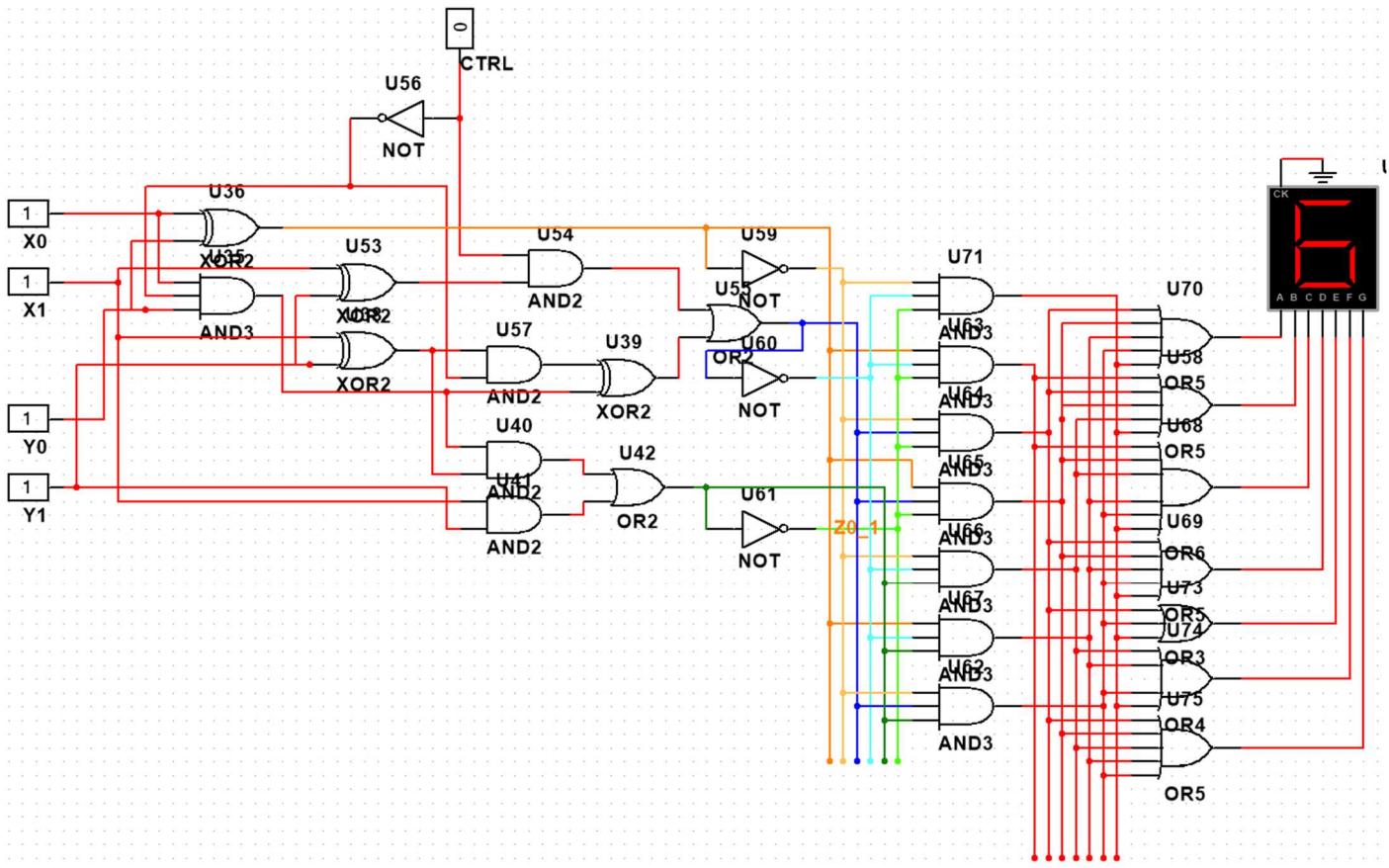
Then design circuit according to these equations.

The first column of AND gates indicate whether the input matches the number it represents.

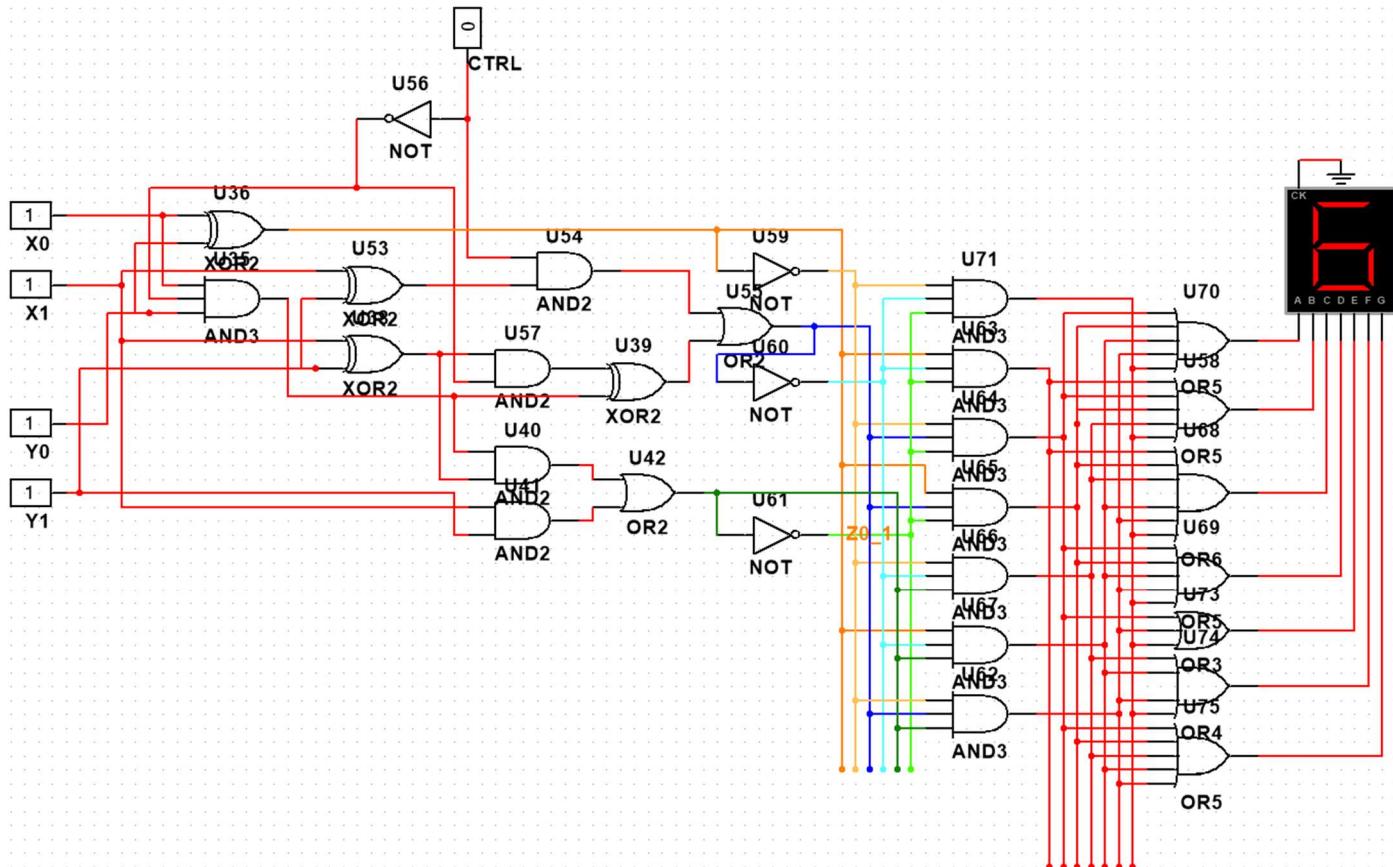
From above to below, they represent 0,1,2,3,4,5,6.

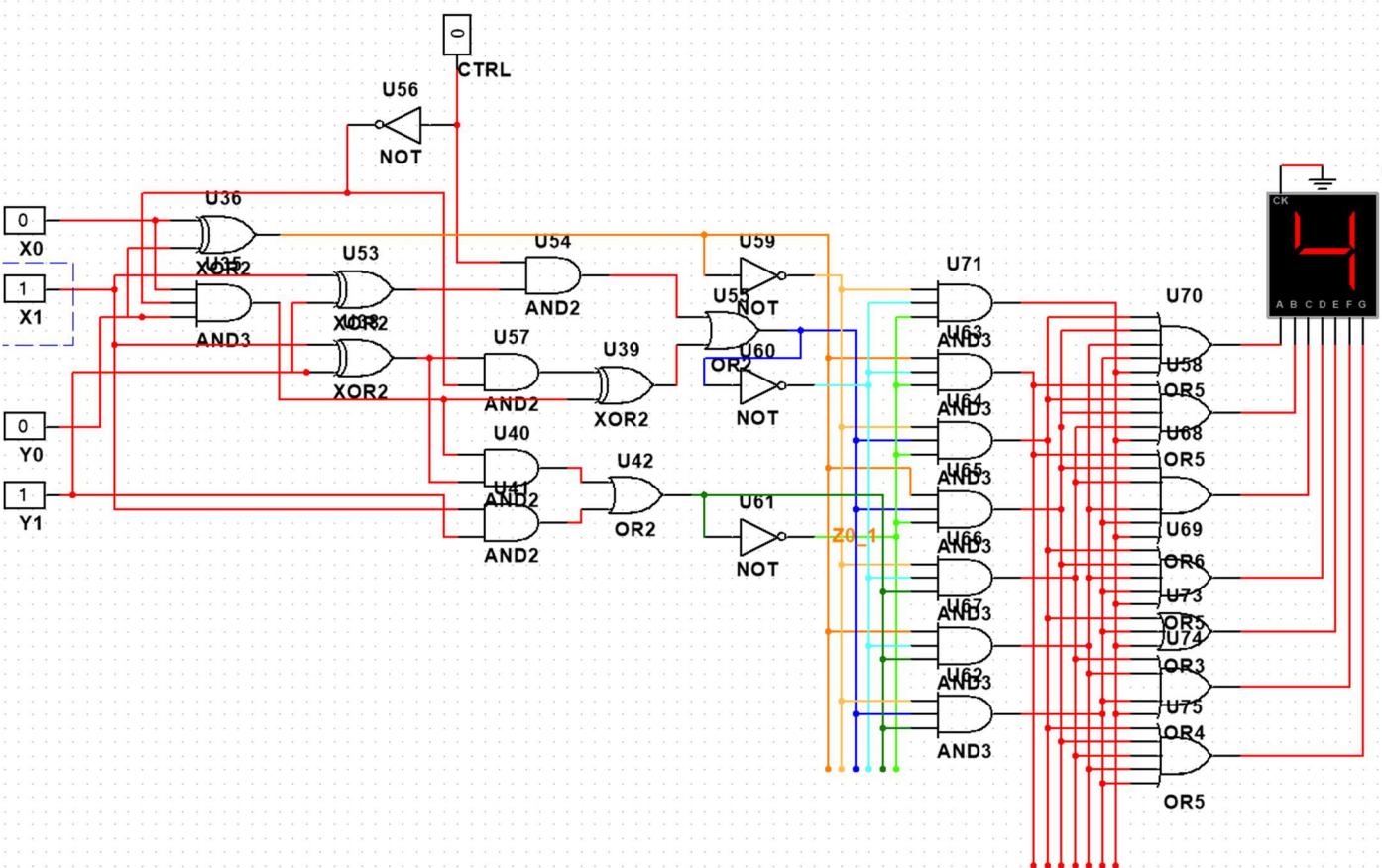
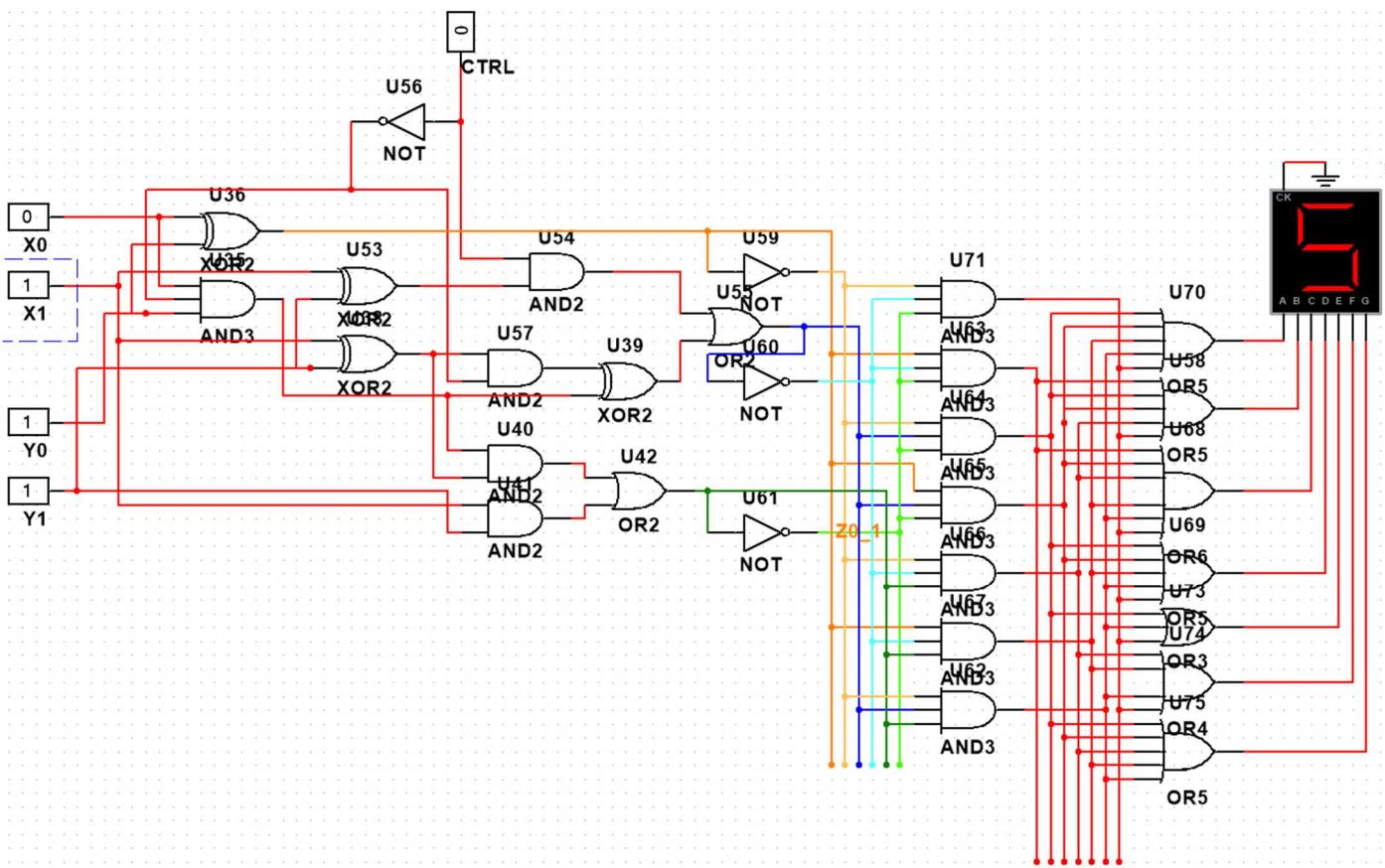
The second column of OR gates control whether the corresponding LED of the display is on.

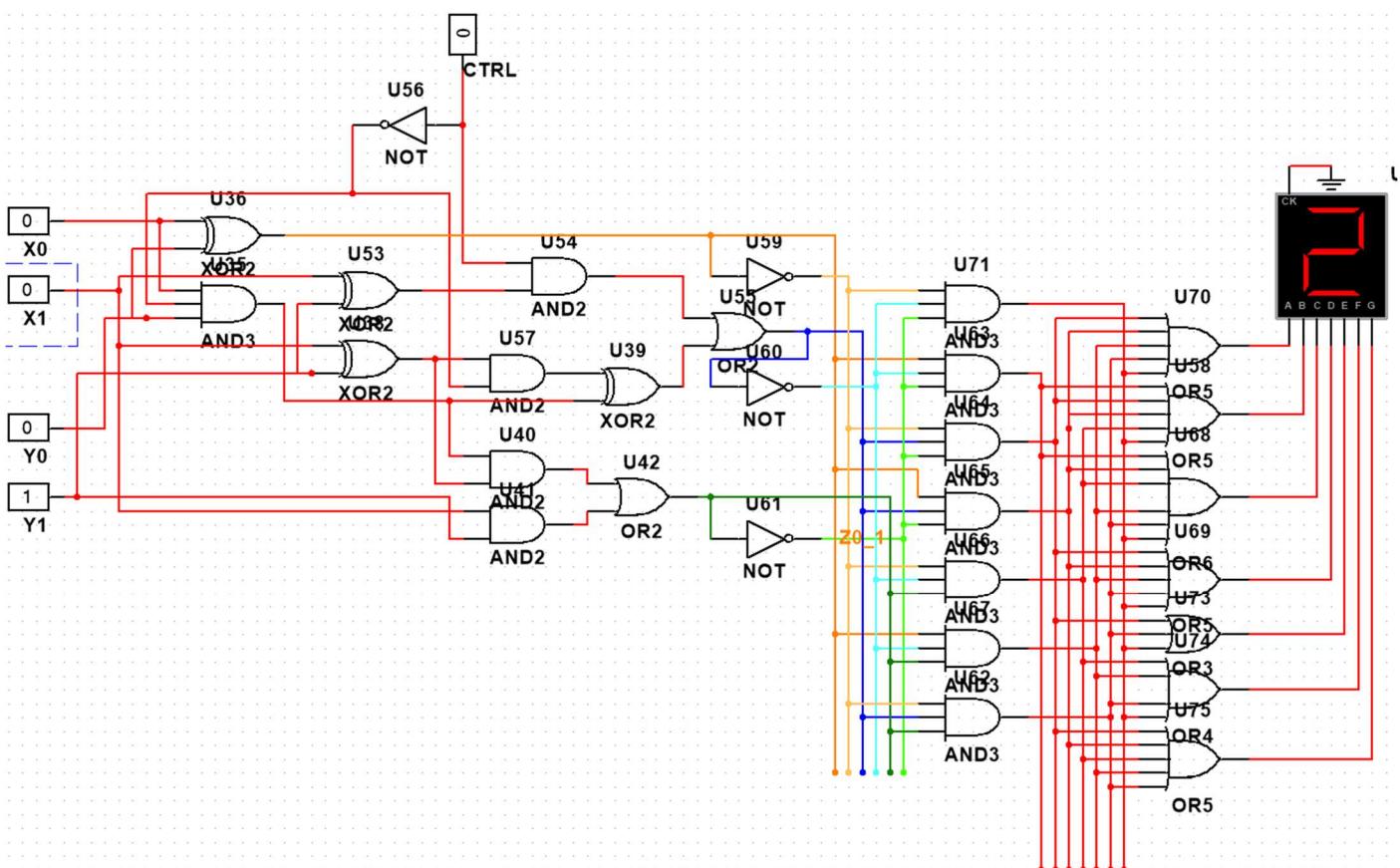
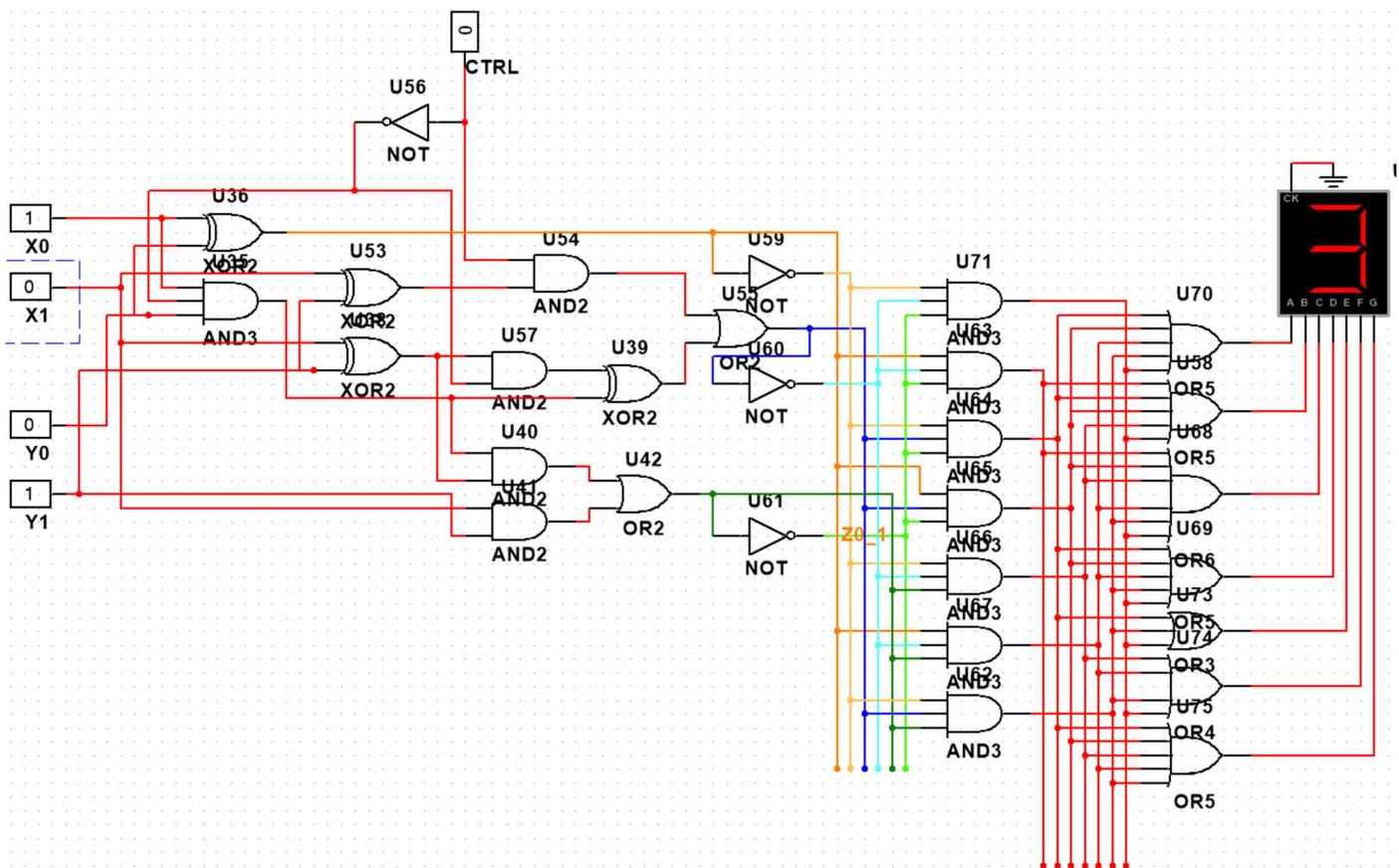
From above to below, they represent LED a,b,c,d,e,f,g.

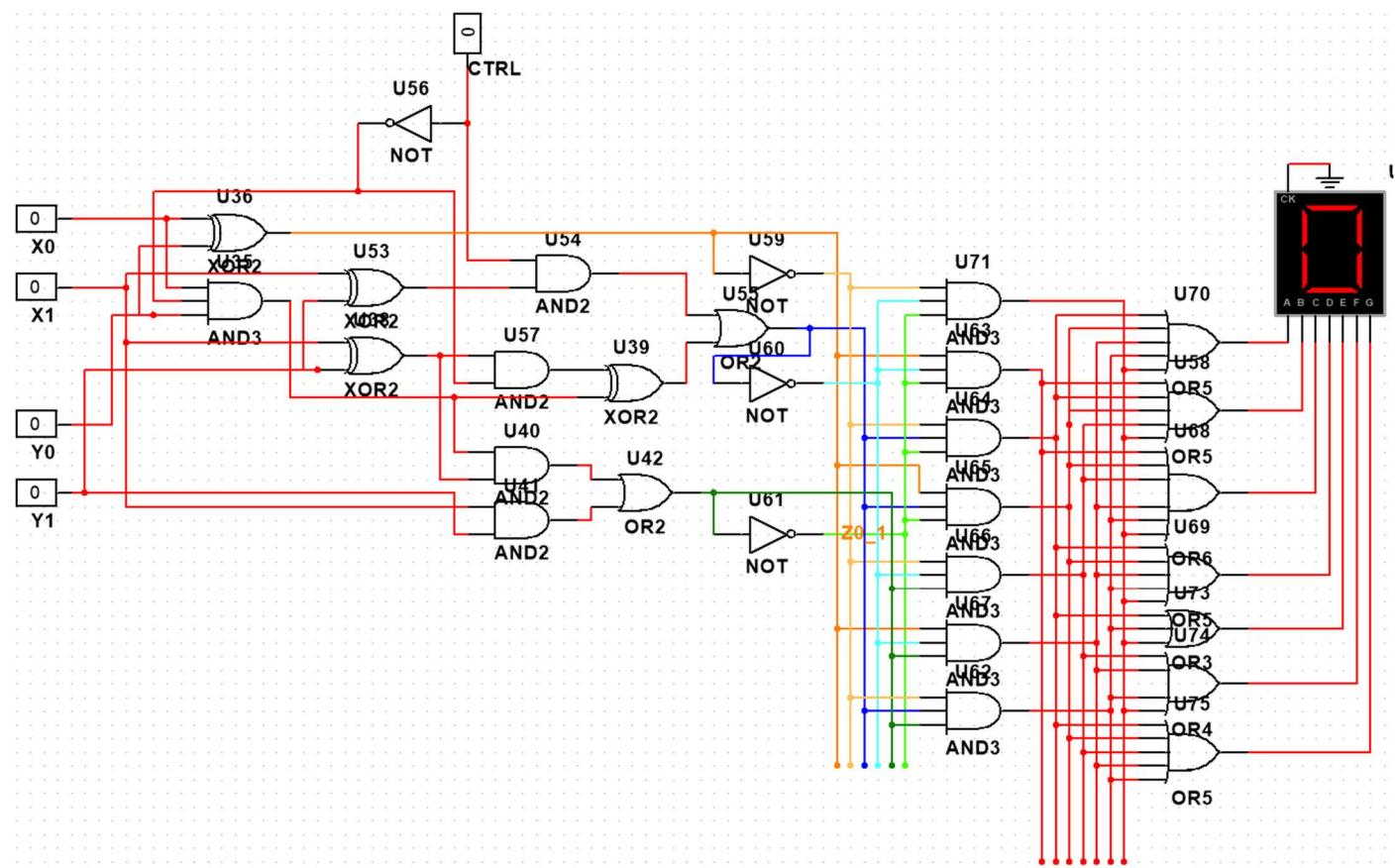
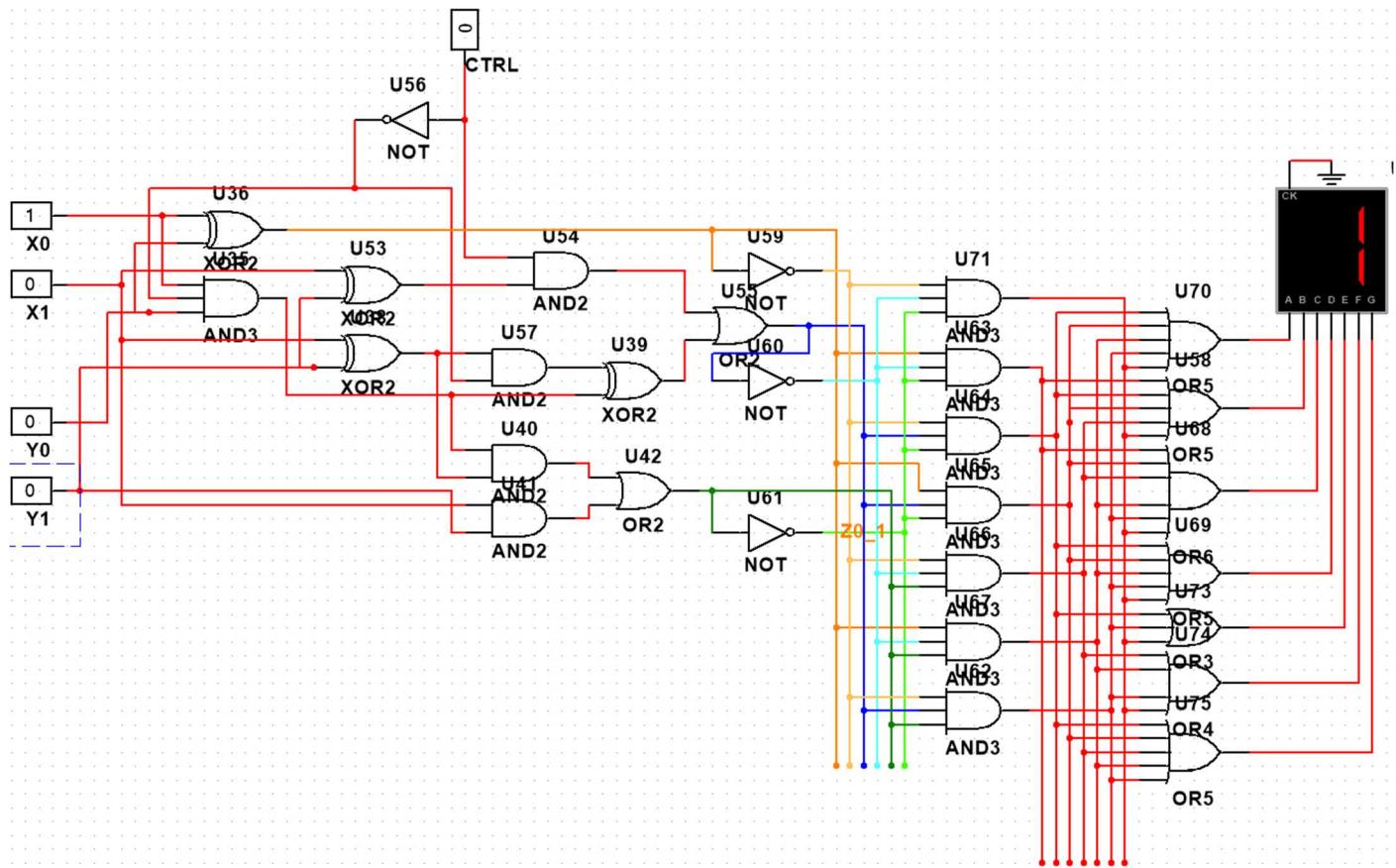


- Show the design schematic and some representative results.









- * Please submit the softcopy of your solutions to the problems on gradescope.
- * All flow charts and codes should be enclosed in your solutions.
- * Discussion on methodology is allowed, yet, the assignment should be done individually. Plagiarism, once found, grades zero for the whole homework assignment!!