

# EE\_HW3

● Graded

## Student

苏慧哲

## Total Points

115 / 115 pts

## Question 1

FSM I

55 / 55 pts

1.1

(no title)

20 / 20 pts

✓ - 0 pts Correct

- 10 pts no output truth table or expression for the one-hot coding

- 5 pts something wrong

- 20 pts wrong

1.2

(no title)

Resolved 20 / 20 pts

✓ - 0 pts Correct

- 5 pts give further explanation of your circuit

- 20 pts wrong

- 10 pts circuit wrong

C Regrade Request

Submitted on: Dec 21

解释在page 4, 每个状态对应的表达式在page 5, 还有哪里不清楚我可以继续解释。

抱歉, 可能以为网页加载问题没有看到, 已重批

Reviewed on: Dec 21

1.3

(no title)

15 / 15 pts

✓ - 0 pts Correct

- 10 pts No design procedure.

- 2.5 pts The result is only valid for the first 60 seconds

- 15 pts No answer

- 15 pts wrong answer

- 2.5 pts Only one red light flash

### Question 2

FSM II

20 / 20 pts

- 0 pts Correct

- 5 pts wrong state transition table
- 5 pts wrong output table
- 5 pts wrong state transition diagram
- 5 pts wrong description
- 2 pts Undefined state

### Question 3

2 bits easy ALU

25 / 25 pts

- 0 pts Correct

- 5 pts lack of truth table or wrong
- 5 pts lack of expressions or wrong
- 5 pts The schematic is wrong.
- 10 pts lack of desgin process
- 12.5 pts lack of logic part or wrong
- 12.5 pts lack of addition part or wrong
- 25 pts wrong

### Question 4

Decoder

15 / 15 pts

- 0 pts Correct

- 15 pts wrong
- 10 pts lack of desgin process
- 5 pts lack of expressions or wrong
- 5 pts lack of schematic or wrong.
- 5 pts lack of truth table or wrong

Question assigned to the following page: [1.1](#)

## Problem Set #3, EE part

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

Student Name: \_\_\_\_\_ 苏慧哲 \_\_\_\_\_ Student No.: \_\_\_\_\_ 2020533009 \_\_\_\_\_

### 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

Since the state transitions doesn't change, we only modify the outputs.

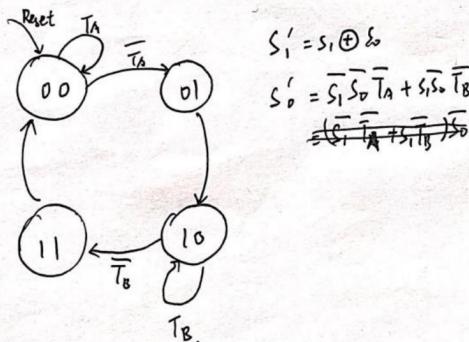
$S_1$	$S_0$	$L_{A2}$	$L_{A1}$	$L_{A0}$	$L_{B2}$	$L_{B1}$	$L_{B0}$
0	0	0	0	1	1	0	0
0	1	0	1	0	1	0	0
1	0	1	0	0	0	0	1
1	1	1	0	0	0	1	0

$$\begin{aligned}L_{A2} &= S_1 \\L_{A1} &= \bar{S}_1 S_0 \\L_{A0} &= \bar{S}_1 \bar{S}_0 \\L_{B2} &= \bar{S}_1 \\L_{B1} &= S_1 S_0 \\L_{B0} &= S_1 \bar{S}_0\end{aligned}$$

Output	L2:0
green	001
yellow	010
red	100

$$\begin{aligned}P &= L_{A2} + L_{B2} \\P &= \bar{S}_1 + \bar{S}_1 \\P &= \bar{S}_1 \\Q &= L_{A1} + L_{B1} \\Q &= S_1 S_0 + S_1 S_0 \\Q &= S_1 S_0\end{aligned}$$

Other logic remains same to those on ppt.

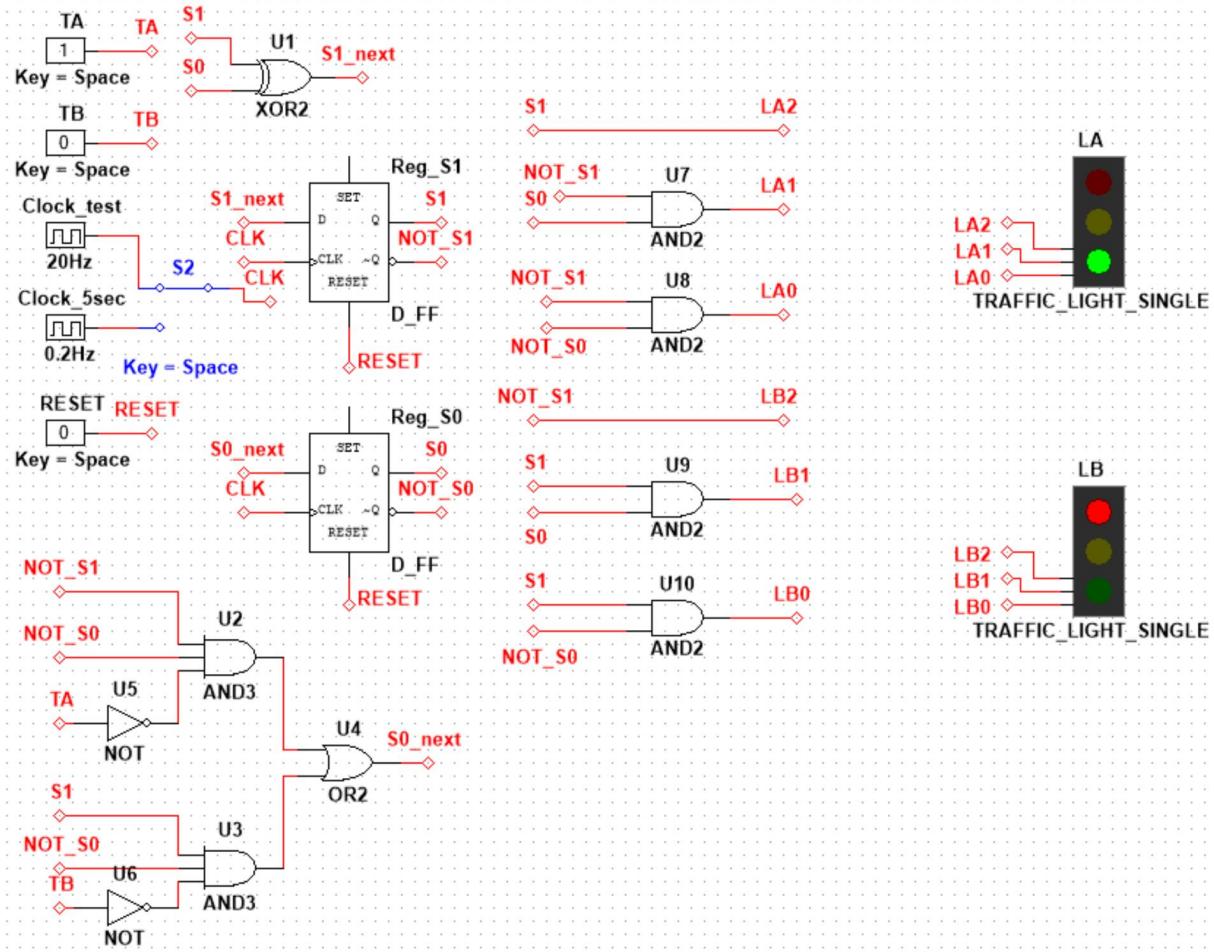


$$\begin{aligned}S'_1 &= S_1 + S_0 \\S'_0 &= \bar{S}_1 \bar{S}_0 T_A + S_1 \bar{S}_0 \bar{T}_B \\&= (\bar{S}_1 \bar{T}_A + S_1 \bar{T}_B) \bar{S}_0\end{aligned}$$

In order to test it easily, we set the clock value to 20 Hz, If you want to test it on 5-second clock, switch the S2 to change the clock value to 0.2 Hz.

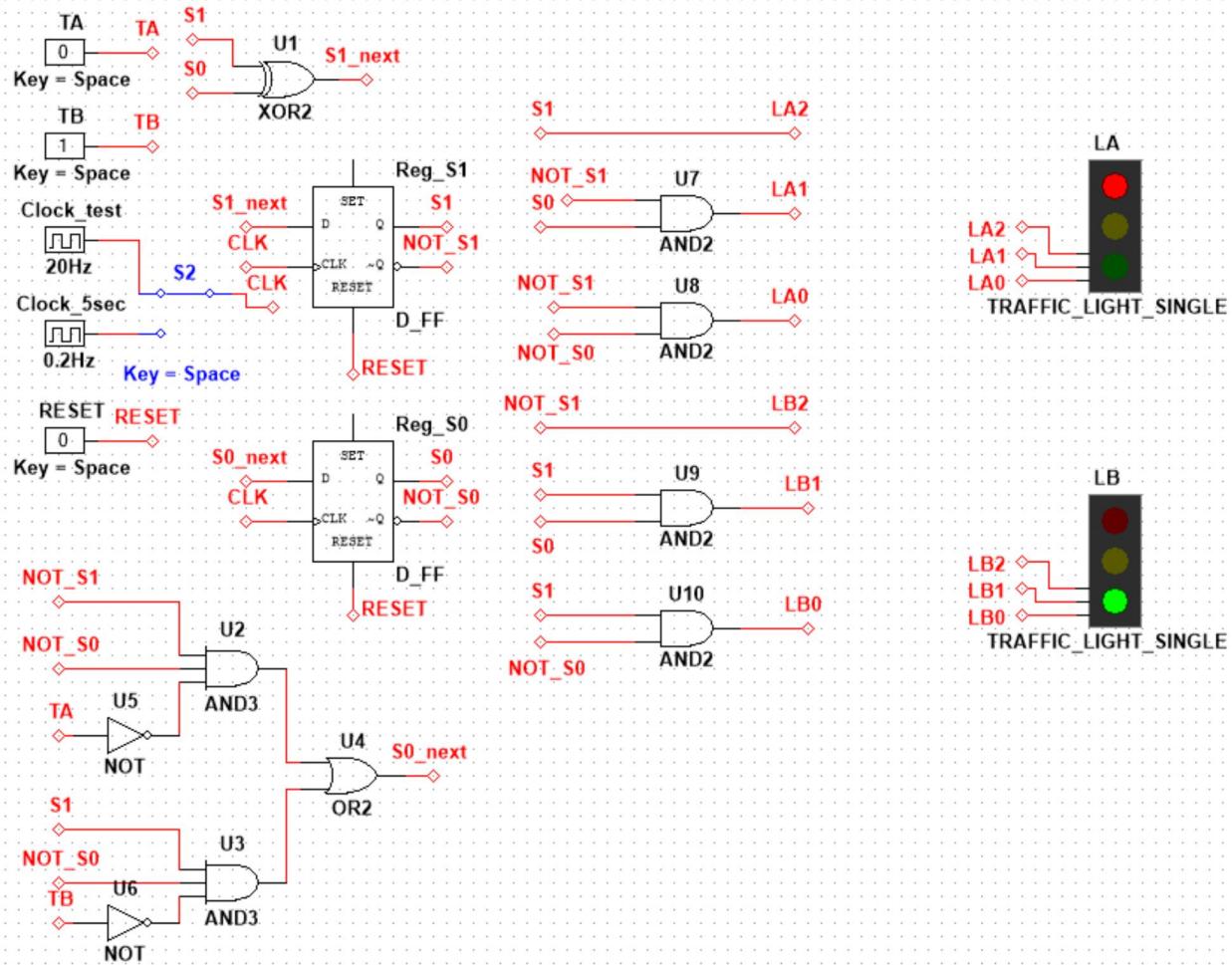


Question assigned to the following page: [1.1](#)





Questions assigned to the following page: [1.1](#) and [1.2](#)



The Circuit is given the name of question1\_1

- 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')

We set traffic light at

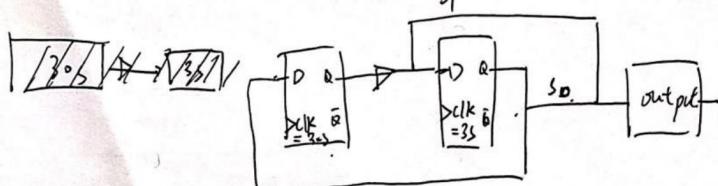
red 30 sec, green 27 sec, yellow 3 sec.



Question assigned to the following page: [1.2](#)

We need a timer to control the time of transformation.

Consider to connect two D flip-flop together, because D flip-flop will only change at the rising edge of the clk. we can hold some state for ~~some time~~ ~~some time~~ for some time by ~~some time~~ modifying the clk frequency.

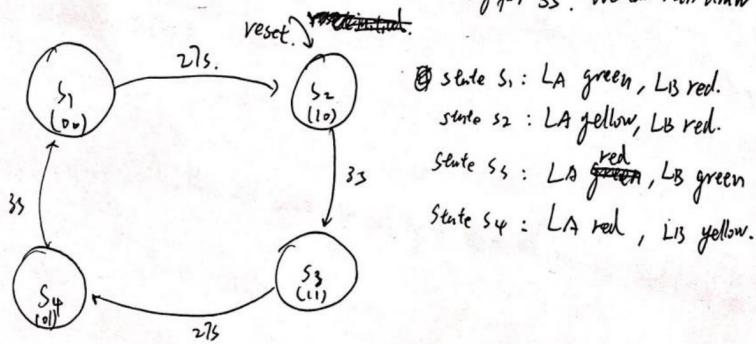


the beginning state would be  $S_1=1, S_2=0,$

and the condition that  $S_1=S_0$  will stay for 27s, the condition that  $S_1 \neq S_0$  will stay for 3s.

we watch the condition  $S_1 \neq S_0$ , with the condition that yellow light lights up.

We can then draw a FSM as follows



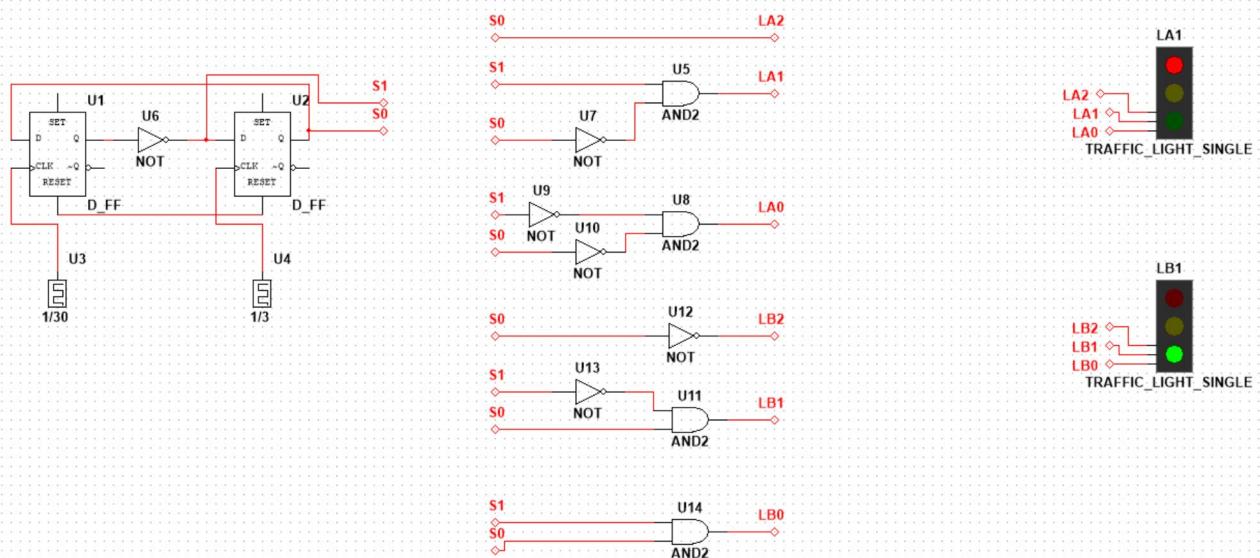


Questions assigned to the following page: [1.2](#) and [1.3](#)

$S_1$	$S_0$	$LA_2$	$LA_1$	$LA_0$	$LB_2$	$LB_1$	$LB_0$
0	0	0	0	1	0	0	0
1	0	0	1	0	0	1	0
1	1	1	0	0	0	0	1
0	1	1	0	0	0	1	0

$$LA_2 = S_0, \quad LA_1 = S_1 \bar{S}_0, \quad LA_0 = \bar{S}_1 \bar{S}_0$$

$$LB_2 = \bar{S}_0, \quad LB_1 = \bar{S}_1 S_0, \quad LB_0 = \cancel{\bar{S}_0} S_1 S_0.$$

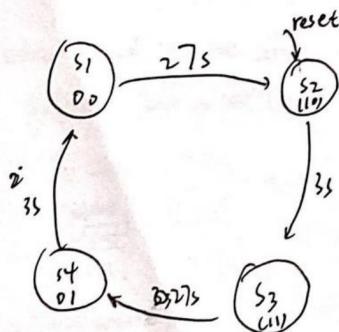


The Circuit is given the name of question1\_2

- The red light twinkles during the last 3 seconds. Can you design such a twinkling design? (bonus +15')



Question assigned to the following page: [1.3](#)



State  $S_1$ : LA green, LB red.

State  $S_2$ : LA yellow, LB red blink,

State  $S_3$ : LA red, LB green

State  $S_4$ : LA red blink, LB yellow.

output	Enuring $L_{3:0}$
green	0001
yellow	0010
red	0100
red blink	1000

$S_1$	$S_0$	$L_{A3}$	$L_{A2}$	$L_{A1}$	$L_{A0}$	$L_{B3}$	$L_{B2}$	$L_{B1}$	$L_{B0}$
0	0	0	0	0	1	0	1	0	0
1	0	0	0	1	0	1	0	0	0
1	1	0	1	0	0	0	0	0	1
0	1	1	0	0	0	0	0	1	0

$$L_{A3} = \overline{S_1} S_0 \quad L_{A2} = S_1 S_0 \quad L_{A1} = S_1 \overline{S_0} \quad L_{A0} = \overline{S_1} \overline{S_0}$$

$$L_{B3} = S_1 \overline{S_0} \quad L_{B2} = \overline{S_1} \overline{S_0} \quad L_{B1} = S_1 S_0 \quad L_{B0} = S_1 S_0$$

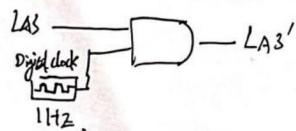
$$\therefore L_{A3} = L_{B1} = S_1 S_0 \quad L_{A2} = L_{B0} = S_1 S_0 \quad L_{A1} = L_{B3} = S_1 \overline{S_0} \quad L_{A0} = L_{B2} = \overline{S_1} \overline{S_0}$$



Questions assigned to the following page: [2](#) and [1.3](#)

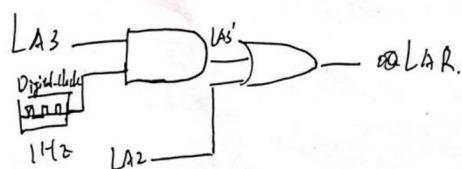
~~LA3 and LA2 both controls the red light~~

Consider how to use LA3 to make red light blink.

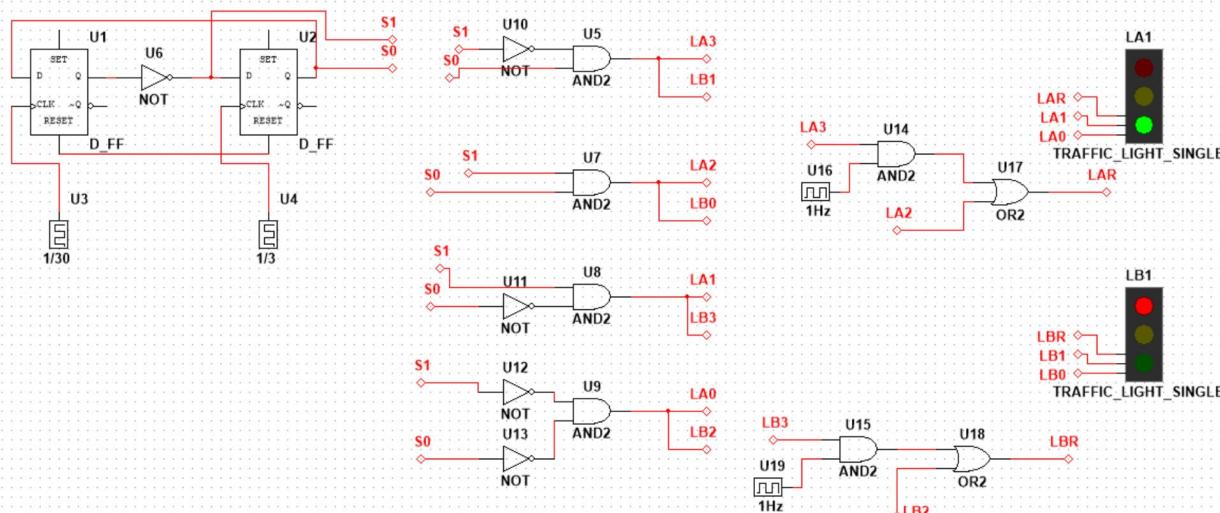


LA3' would follow the digital clock to change its value, when both LA3 and digital clock is 1, LA3' is 0.

This will make LA3' 'blink' every one second.



When lighter LA2 or LA3' is 1, LAR is 1. This will make the red light stay in the first 27s and blink in the last 3s. The same with LBR.



The Circuit is given the name of question1\_3

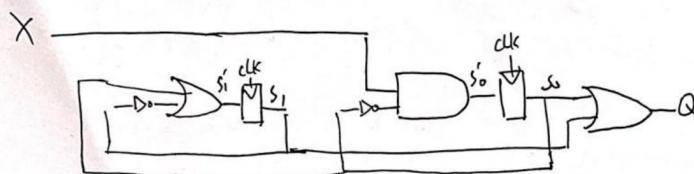
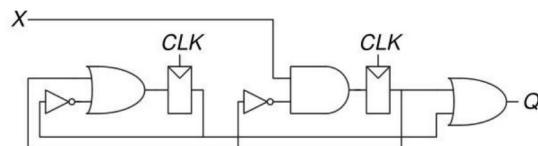
## 2. FSM II

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')



Question assigned to the following page: [2](#)



$$Q = S_0 + S_1$$

$$S'_0 = X \overline{S_0}$$

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

State table

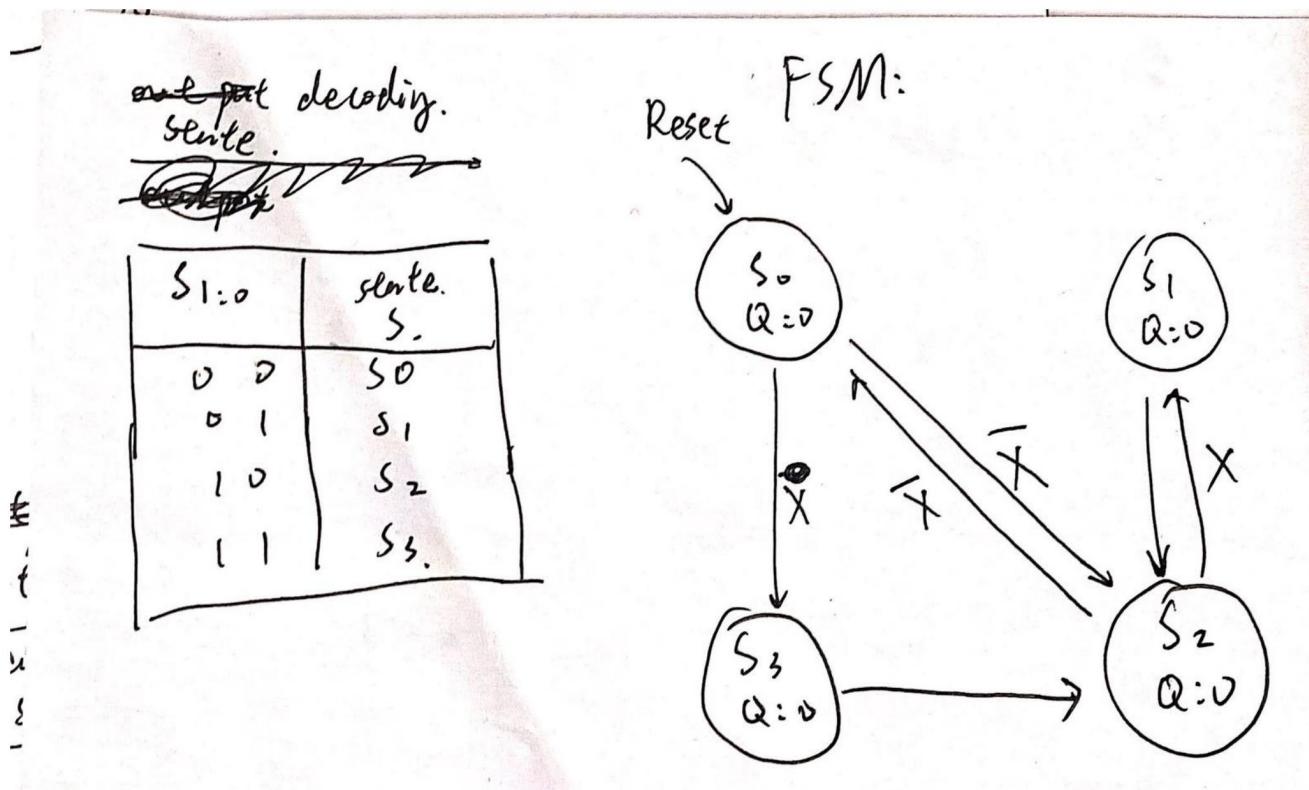
Current state		inputs	next states	
$S_1$	$S_0$	X	$S'_1$	$S'_0$
0	0	0	1	0
0	0	1	1	01
1	0	0	0	0
1	0	1	0	1
1	1	0	1	0
1	1	1	1	0
0	1	0	1	0
0	1	1	1	0

~~Output table~~

current state		<del>outputs</del>
$S_1$	$S_0$	Q
0	0	0
0	1	1
1	0	1
1	1	1



Questions assigned to the following page: [2](#) and [3](#)



It starts(resets) at S0. When input X is 0, it transforms to S2. Else if X is 1, then it transforms to S3.

When at S2, if X is 1, it transforms to S1, and then falls back to S2, regardless of X. Else if X is 0, then it transforms to S0.

When at S3, it transforms to S2, regardless of X.

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

My student number is :2020533009



Question assigned to the following page: [3](#)

half adder..

INPUT		OUTPUT	
A	B	Carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$\Rightarrow \text{XOR}$

$\Rightarrow$

$\Rightarrow \text{ADD}$



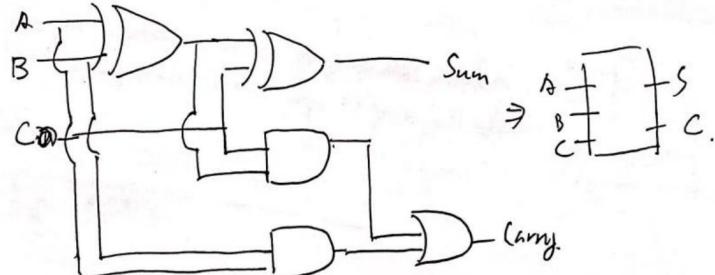
$$\Rightarrow A + B = S$$

Sum

Carry.

full adder

INPUT			OUTPUT	
A	B	C	Carry	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



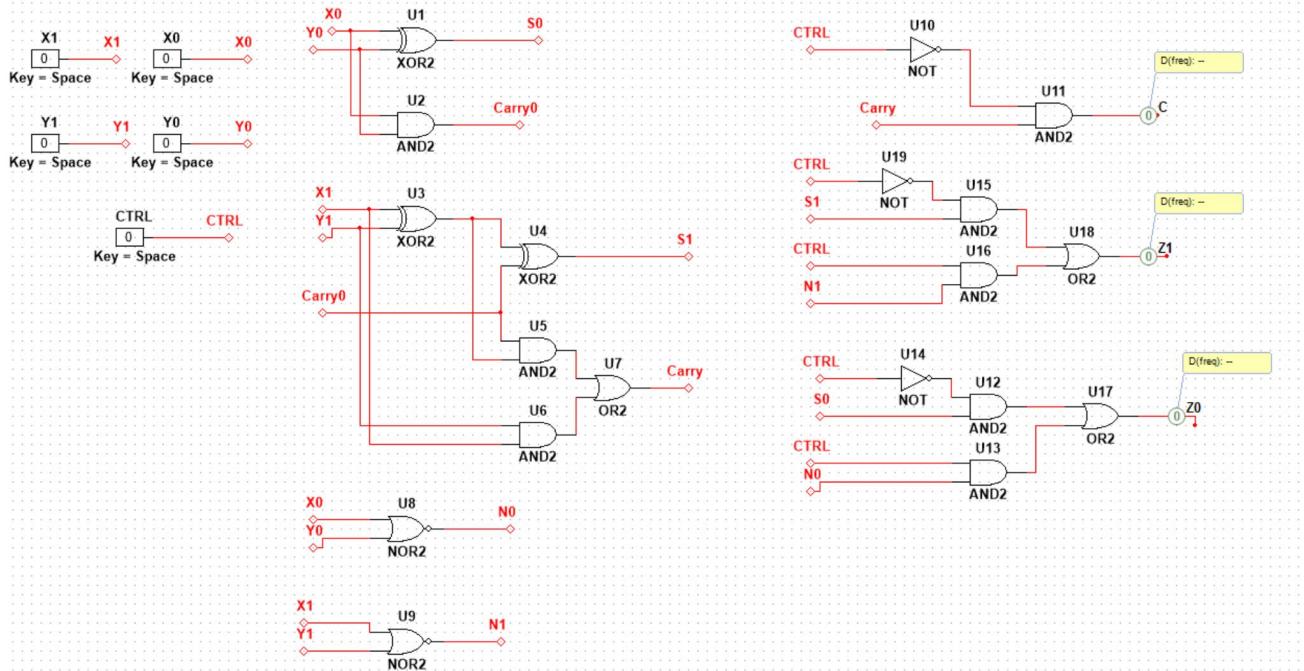
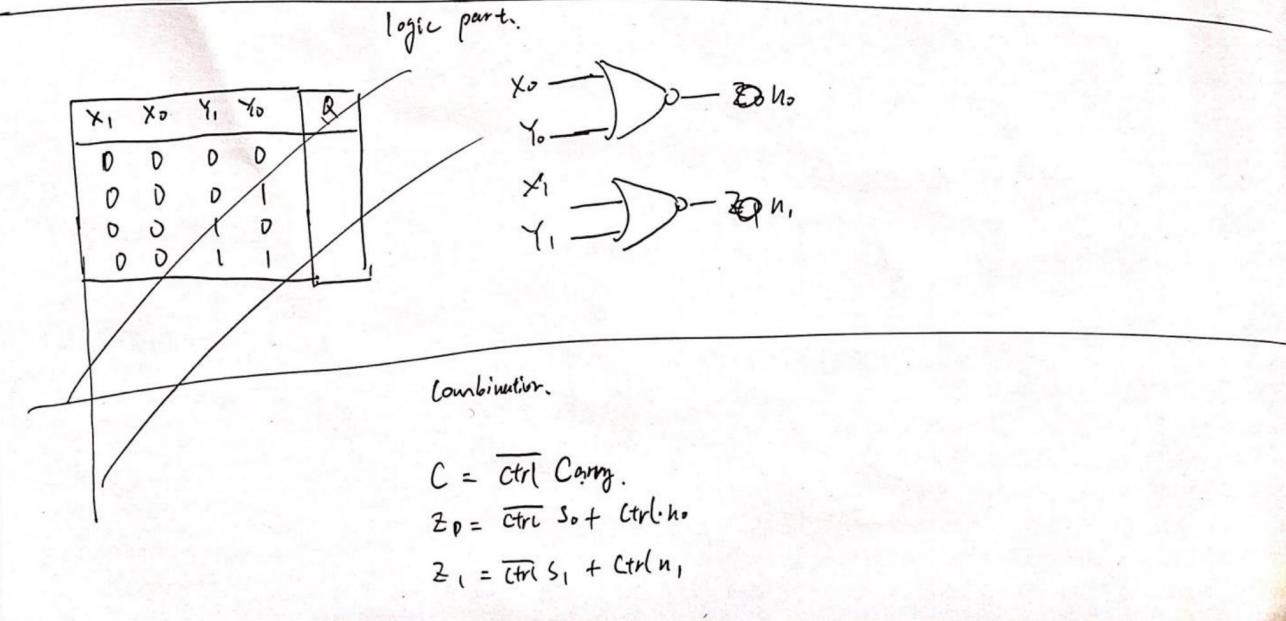
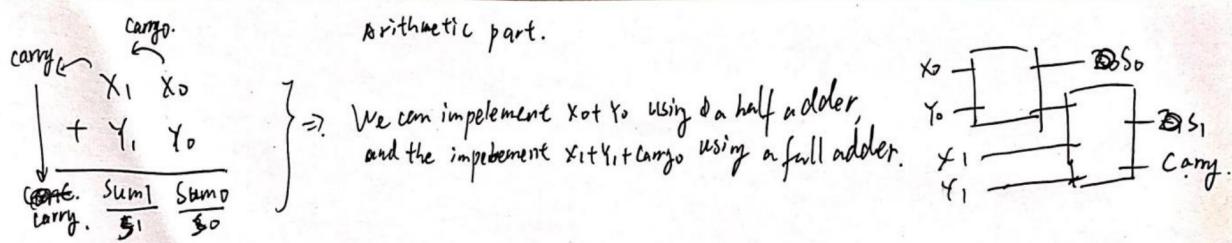
$$\Rightarrow A + B + C = S$$

Sum

Carry.



Questions assigned to the following page: [4](#) and [3](#)



The Circuit is given the name of question2\_1

4. Decoder (15')



Question assigned to the following page: [4](#)

- 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  $3+3=6$ .

$$\begin{aligned} D_0 &= z_0 \\ D_1 &= z_1 \\ D_2 &= c. \end{aligned}$$

$D_{2:0}$	$S_a$	$S_b$	$S_c$	$S_d$	$S_e$	$S_f$	$S_g$
000	1	1	1	1	1	1	0
001	0	1	1	0	0	0	0
010	1	1	0	1	1	0	1
011	1	1	1	1	0	0	1
100	0	1	1	0	0	1	1
101	1	0	1	1	0	1	1
110	1	0	1	1	1	1	1
111	1	1	1	1	0	0	0

K-map of  $S_c$ .

$D_2:1$	00	01	11	10
$D_0$	00	01	11	10
0	1	0	1	1
1	1	1	1	1

$$S_c = \overline{D}_1 + D_0 + D_2.$$

K-map of  $S_a$ .

$D_2:1$	00	01	11	10
$D_0$	00	01	11	10
0	1	1	1	0
1	0	1	1	1

$$\begin{aligned} S_a &= \cancel{\overline{D}_0 \overline{D}_1 \overline{D}_2} + \cancel{D_1} + \cancel{\overline{D}_0 \overline{D}_2} + \cancel{D_2} \\ S_a &= \overline{D}_0 \overline{D}_2 + D_1 + D_0 P_2. \end{aligned}$$

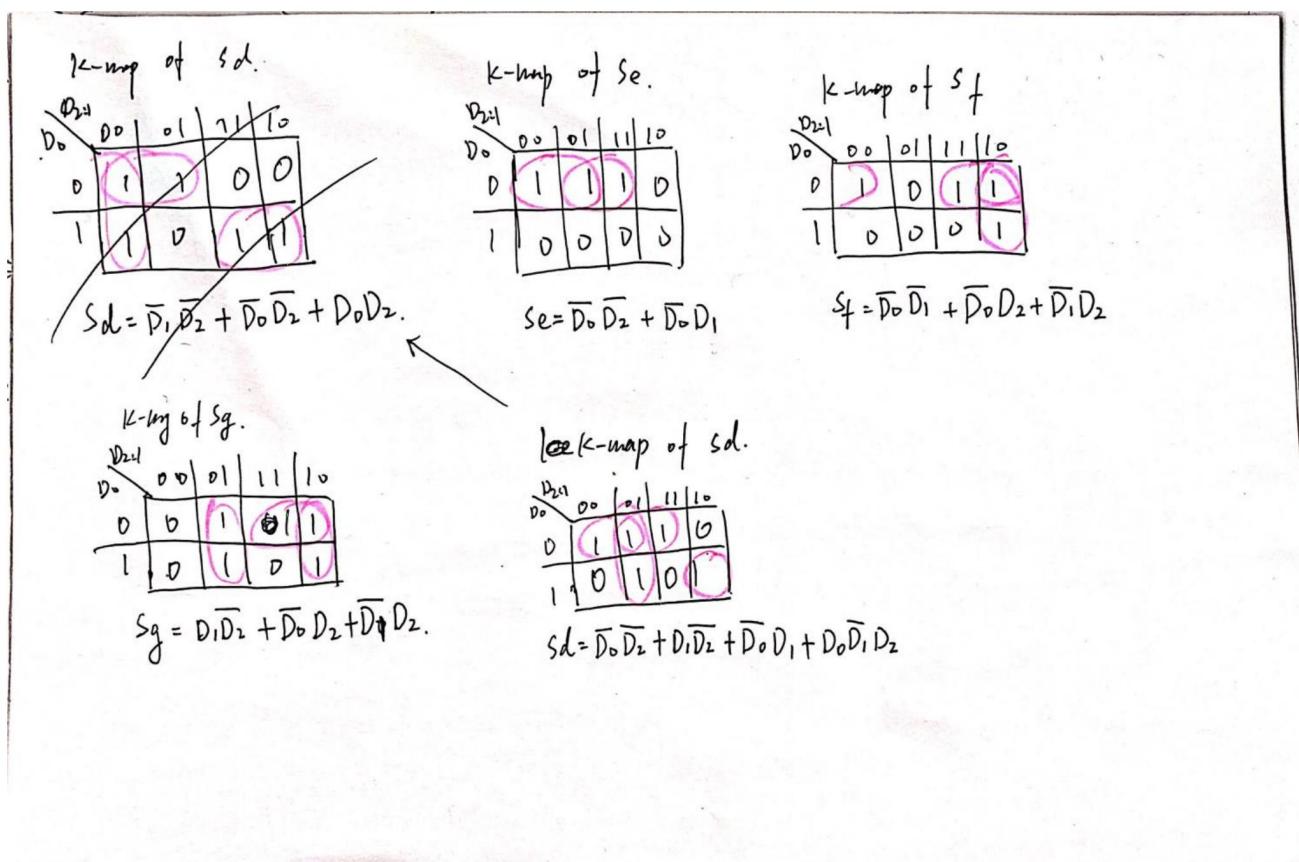
K-map of  $S_b$ .

$D_2:1$	00	01	11	10
$D_0$	00	01	11	10
0	1	1	0	1
1	1	0	1	0

$$S_b = \overline{D}_2 + \overline{D}_0 \overline{D}_1 + D_0 D_1$$



Question assigned to the following page: [4](#)

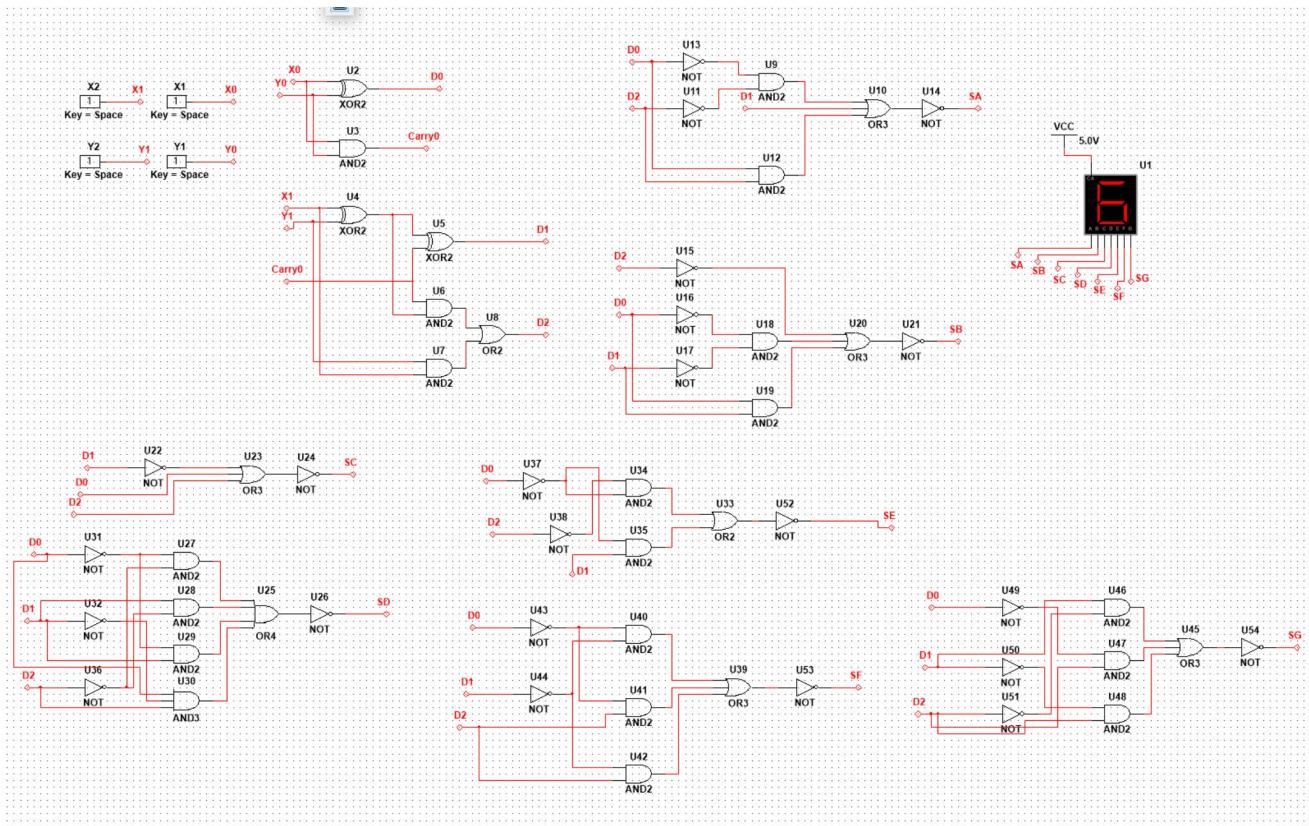


- Show the design schematic and some representative results.

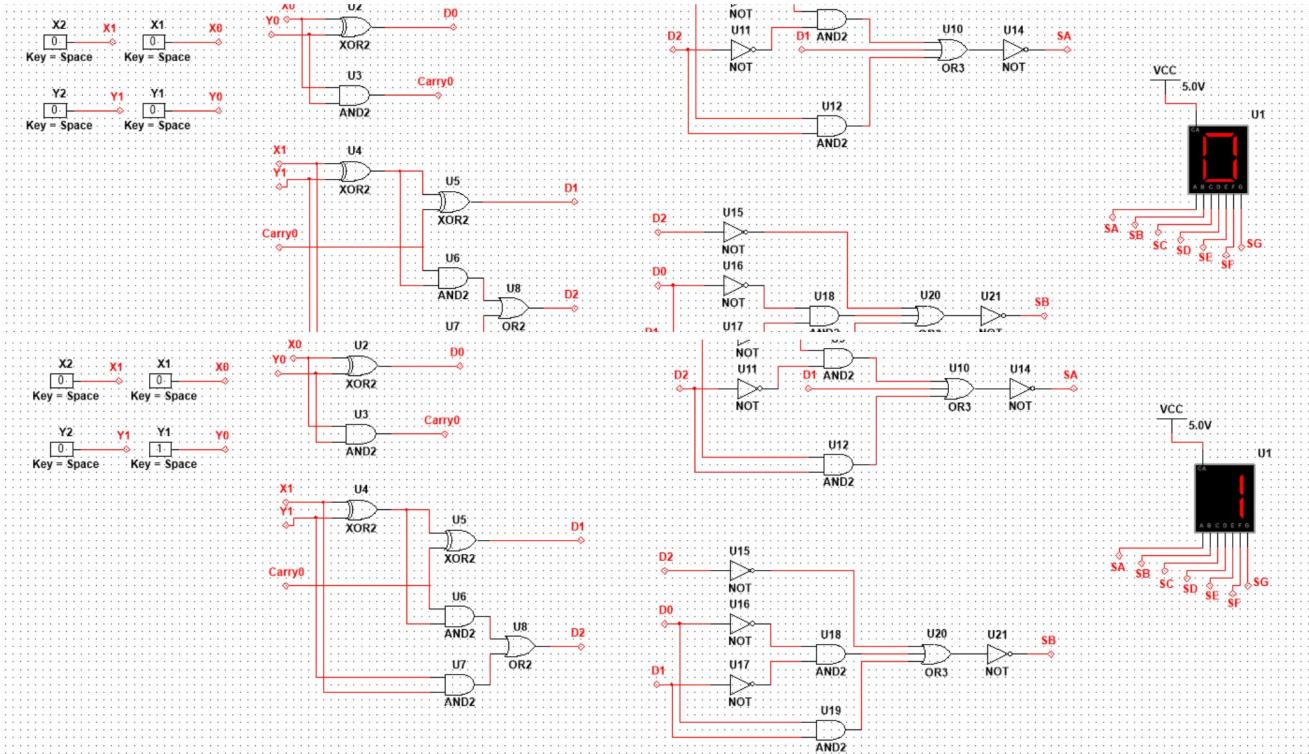
Schematic overall view:



Question assigned to the following page: [4](#)

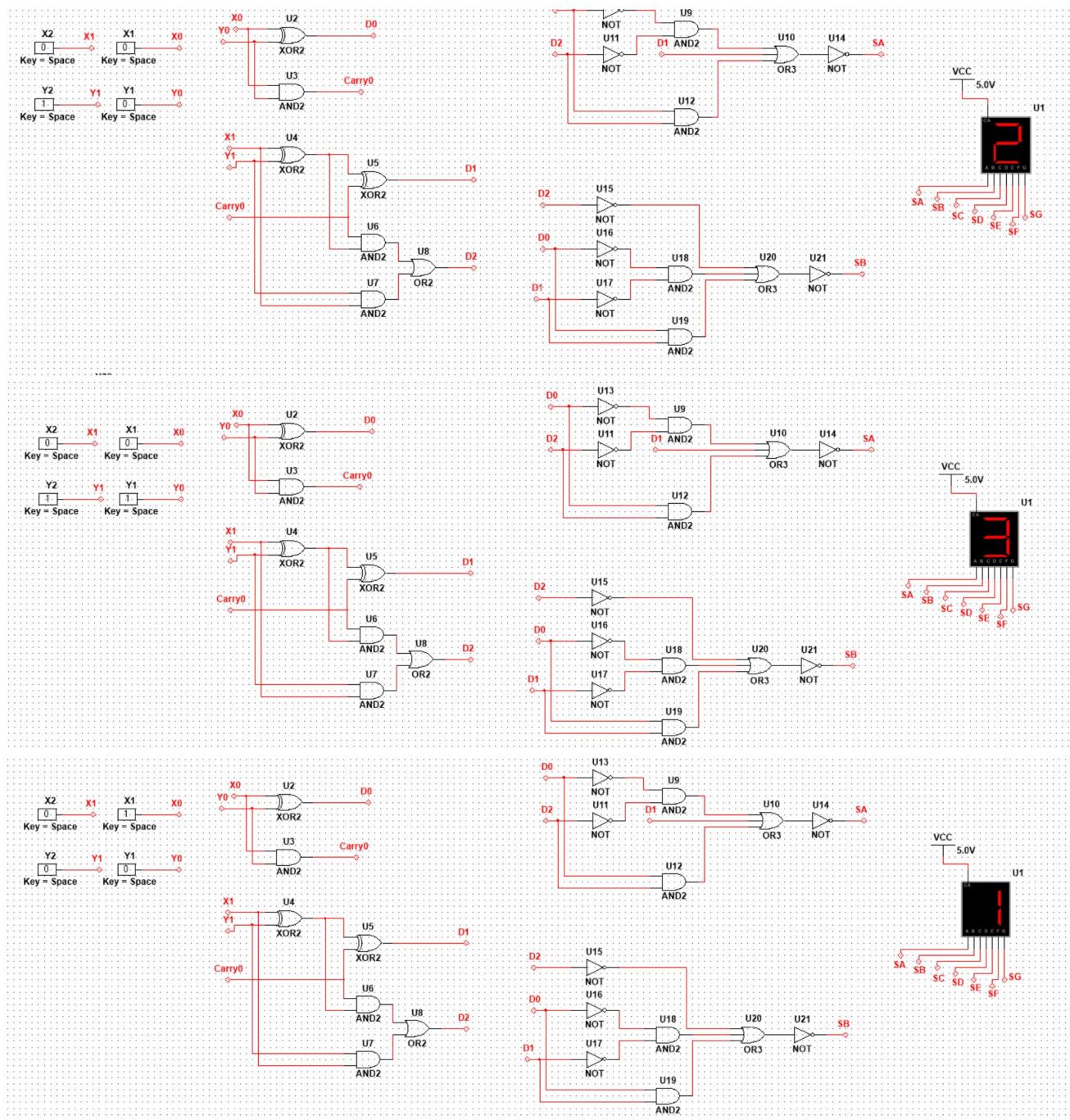


representative results:



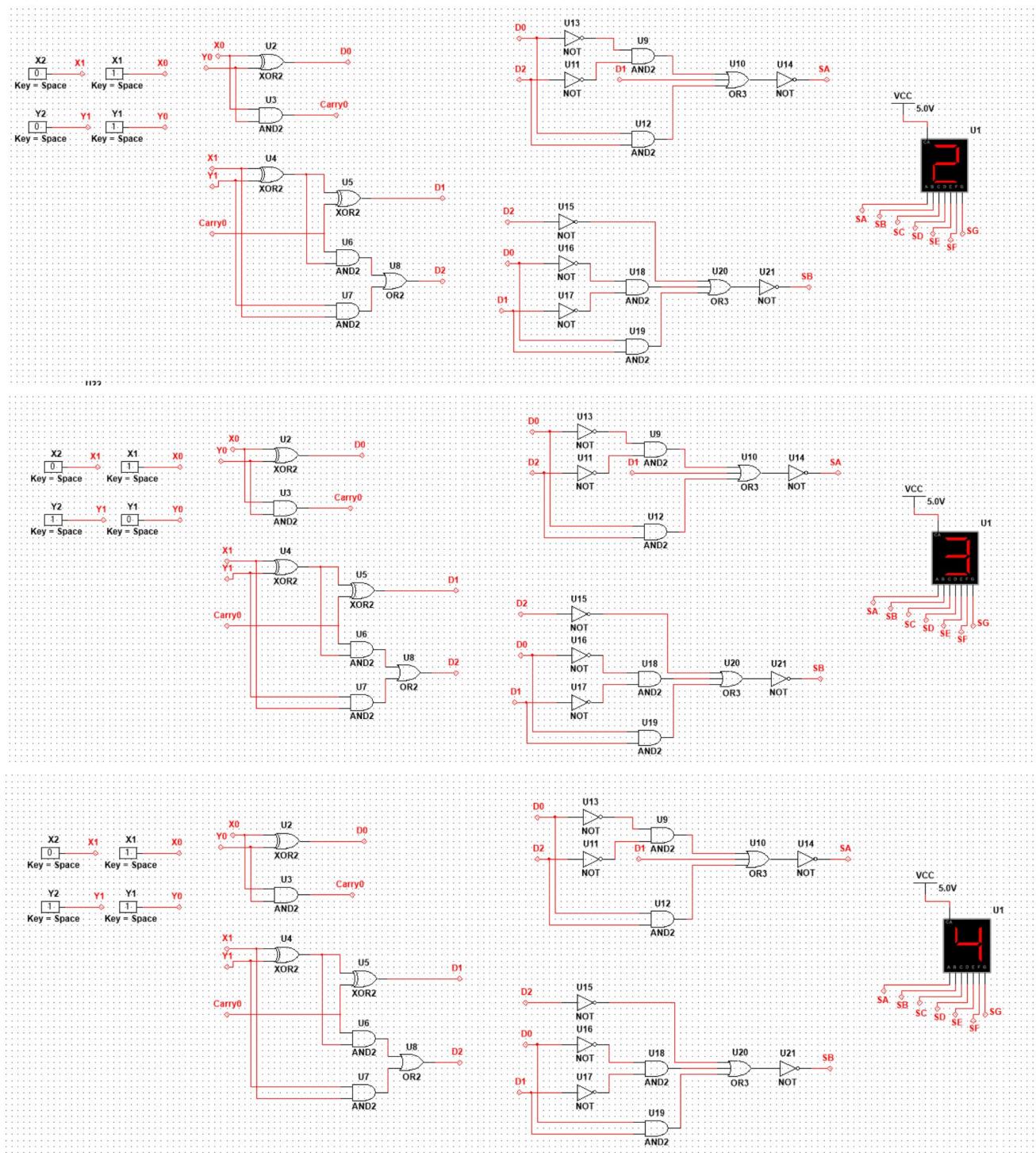


Question assigned to the following page: [4](#)



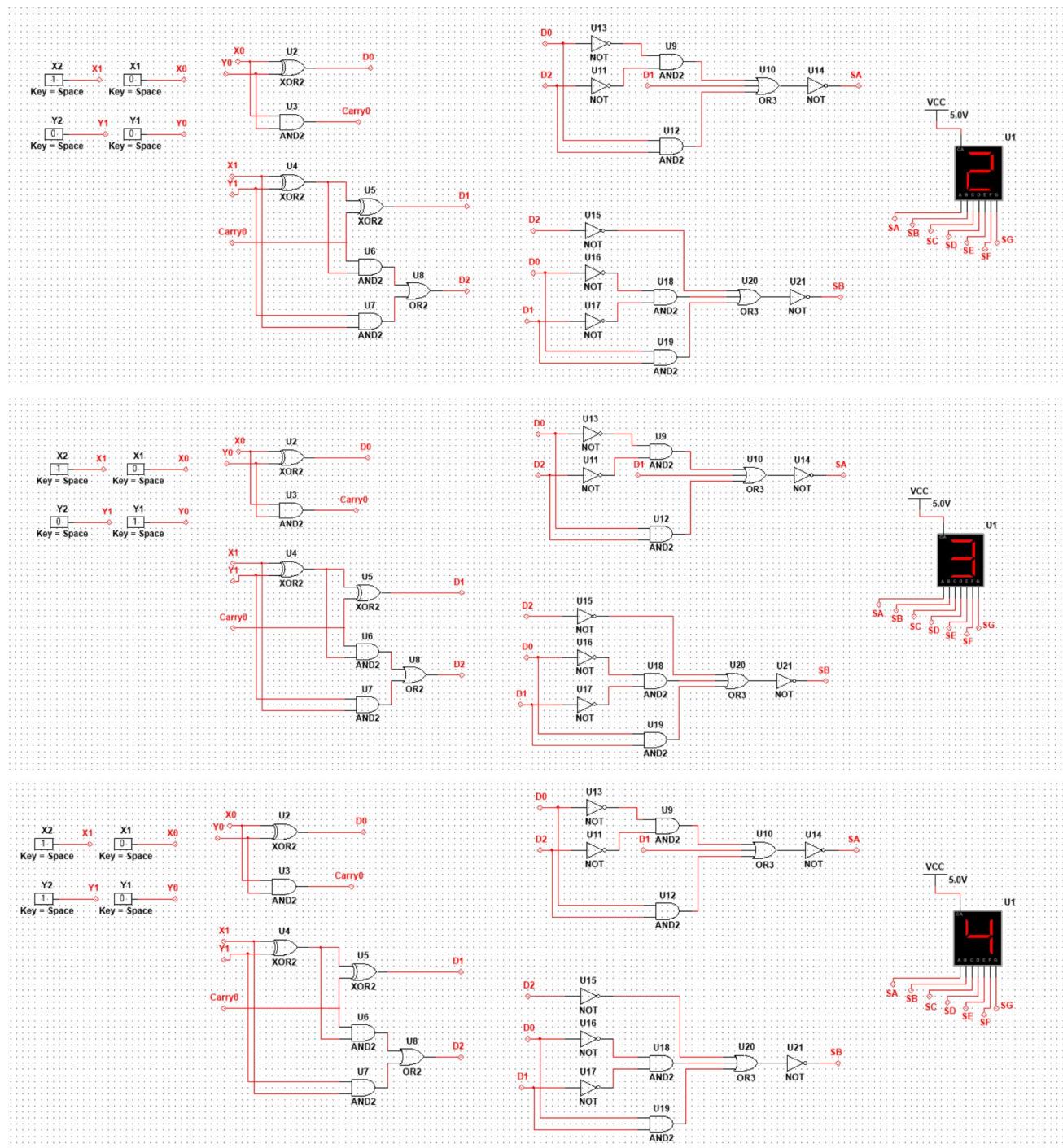


Question assigned to the following page: [4](#)



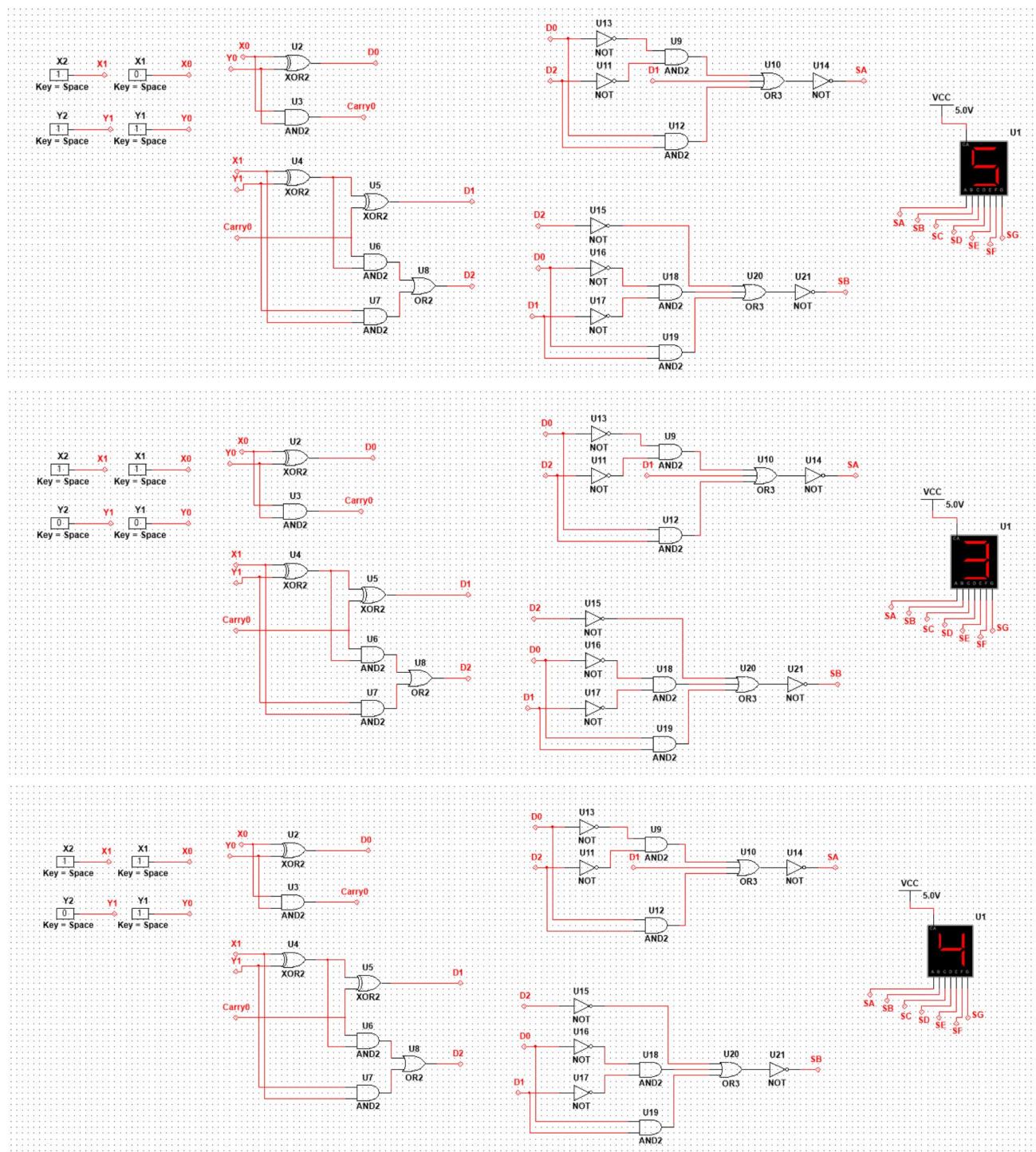


Question assigned to the following page: [4](#)



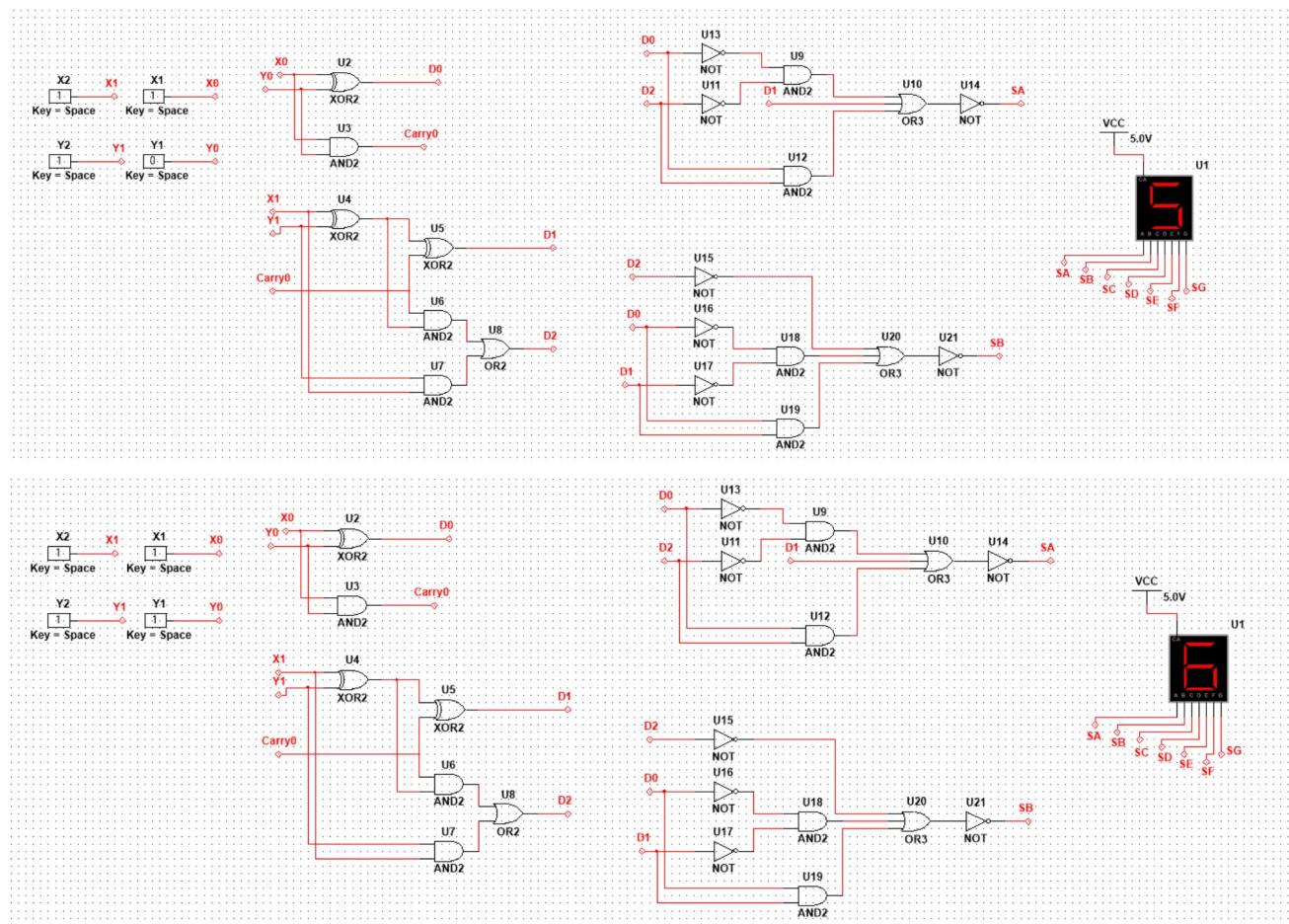


Question assigned to the following page: [4](#)





Question assigned to the following page: [4](#)



The Circuit is given the name of question4\_1

\* 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!!

