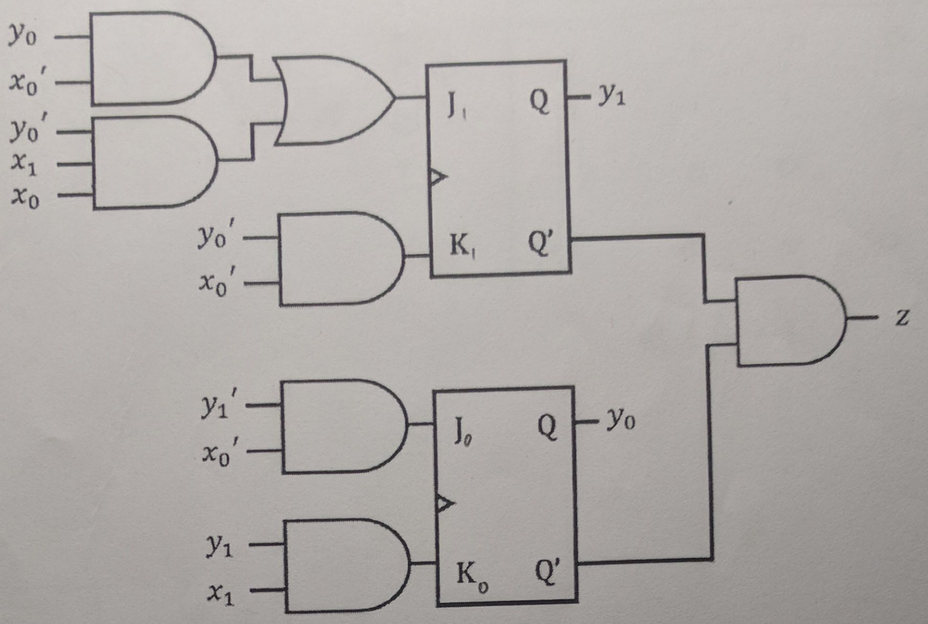
Problem 1 (20 points)

Obtain a high-level description (state transition table) of the network shown in the figure below. The system has two inputs (x_1 and x_0) and a single output z.



$J_1 = y_0 x_0' + y_0' x_1 x_0$

$K_1 = y_0' x_0'$

For a JK flip-flop, $Q(t+1) = Q(t)K'(t) + Q'(t)J(t) \rightarrow Q_1(t+1) = Q_1(t)K_1'(t) + Q_1'(t)J_1(t$

$Q_1(t+1) = Q_1(t)(y_0'x_0')' + Q_1'(t)(y_0x_0' + y_0'x_1x_0)$

$= Q_1(t)(y_0 + x_0) + Q_1'(t)(y_0x_0' + y_0'x_1x_0)$ Let $q_1 = Q_1(t)$, $q_0 = Q_0(t)$

$= q_1q_0 + q_1x_0 + q_1'q_0x_0' + q_1'q_0'x_1x_0$

K-map for $Q_1(t+1)$

x_1

0	0	1	0
1	1	0	0
1	1	1	1
0	0	1	1

q_0

q_1

x_0

$\longrightarrow$

	x_1x_0			
q_1q_0	00	01	11	10
00	0	0	1	0
01	1	1	0	0
11	1	1	1	1
10	0	0	1	1

Table 1

$Q_1(t+1)$

$J_0 = y_1'x_0'$

$y_1'y_0 + y_0x_1' + y_0'y_1'x_0'$

$K_0 = y_1x_1$

$Q_0(t+1) = Q_0(t)(y_1x_1)' + Q_0'(t)(y_1'x_0')$

$= q_0q_1' + q_0x_1' + q_0'q_1'x_0'$

Problem 2 (20 points)

Using 1 T flip-flop, 1 SR flip-flop, and at most 8 JK flip-flops, design a system as described below. Use any gates to implement your combinational logic.

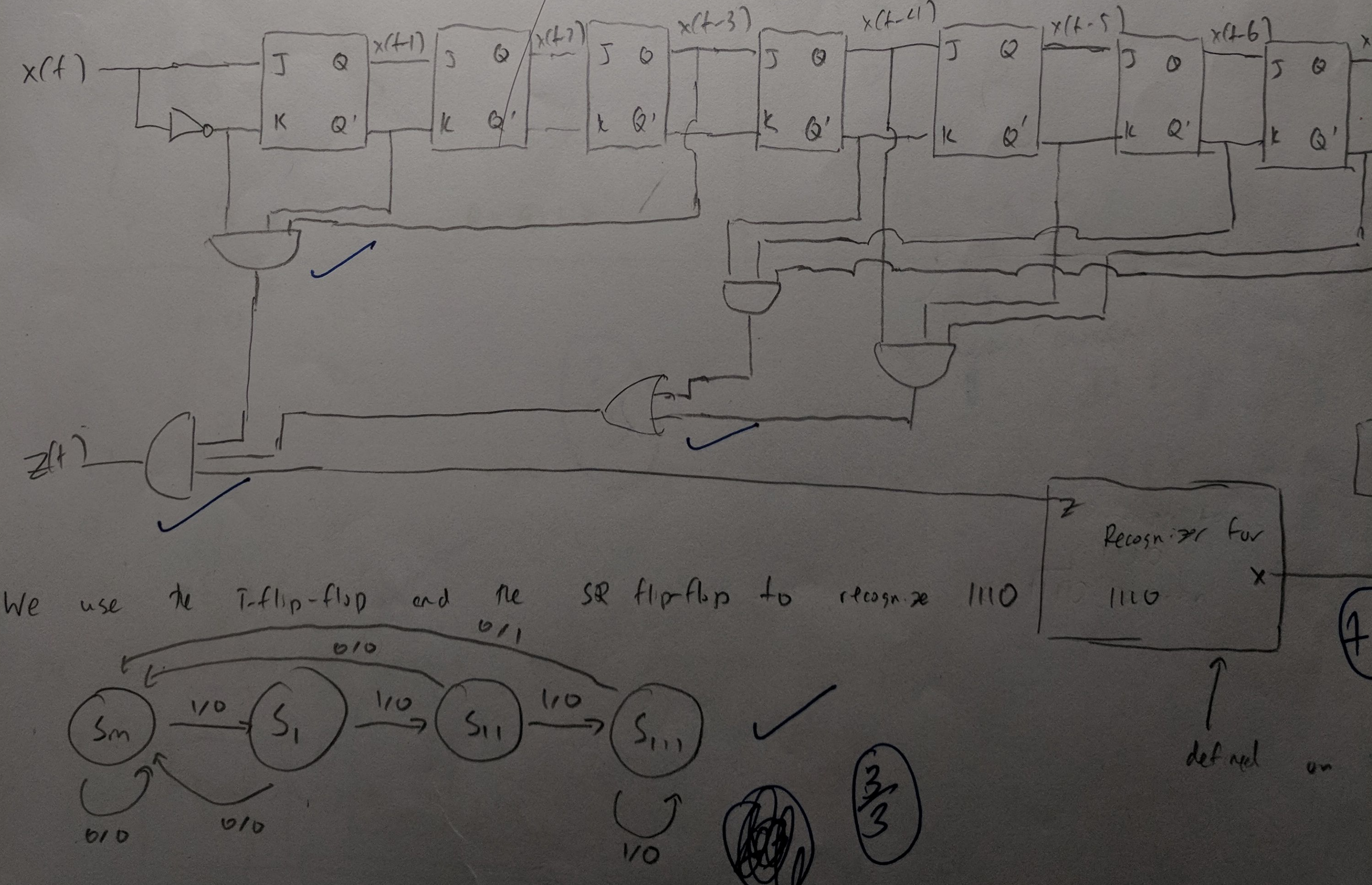
Input: $x(t) \in \{0,1\}$
Output: $z(t) \in \{0,1\}$
Function: $z(t) = \begin{cases} 1 & \text{, if } x\text{(t-11, t-8)=1110, } x\text{(t-7, t-4)=10-0 or 0-01, } x\text{(t-3, t)=1-00} \\ 0 & \text{, otherwise} \end{cases}$

For example, for the given input sequence x(t-11, t) = 111010001100, the output z(t) = 1. For the input sequence x(t-11, t) = 111000101000, the output z(t) = 0.

Note that we can create a D flip-flop from a JK flip flop by letting

J = x(t), K = x'(t).

The patterns at x(t-7, t-4) and x(t-3, t) can be recognized using combinat logic and D-flip flops.



We use the T-flip-flop and the SR flip-flop to recognize 1110

Let Sm be represented by 00
S1 by 01
S11 by 11

12

Problem 3 (20 points)

Design a system that generates a pattern of either "UCLA" or "USC" based on an input signal.

As soon as the input bit turns 1, the pattern generator starts to output "UCLA."
If the input bit turns 0 (*0 is less than 1*), the pattern generator starts to output "USC."

Some examples:

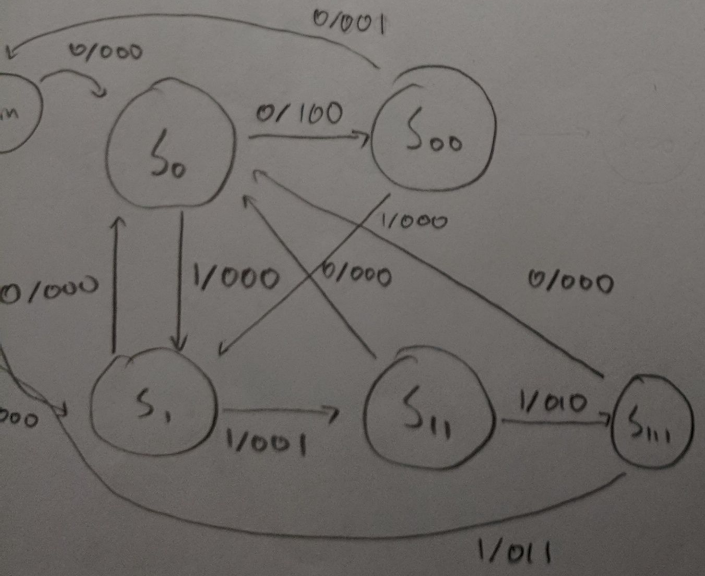
1. If the input sequence of x(t-13, t) = 11110000001111, the output sequence of **z**(t-13, t) = UCLAUSCUSCUCLA.
2. If the input sequence of x(t-9, t) = 1111001111, the output sequence of **z**(t-9, t) = UCLAUSUCLA.

Use the least number of bits for character encoding, and the minimal number of flip-flops and 4:1 multiplexers. (2 select inputs)

et $z_2 z_1 z_0$ be the output bits, with

U = 000
C = 001
✓ L = 010
A = 011
S = 100

We have the following state transition diagram:



Let S_{in} = 000
S_0 - 001
S_{00} = 010
S_1 = 011
S_{11} = 100

PS			NS, $z_2 z_1 z_0$	
q_2	q_1	q_0	x=0	x=1
0	0	0	001,000	011,000
0	0	1	010,100	011,000
0	1	0	000,001	011,000
0	1	1	001,000	100,001
1	0	0	001,000	101,010
1	0	1	001,000	

Problem 4 (20 points)

Design a pattern recognizer for 00-01 using only flip-flops as defined below and NOR gates.

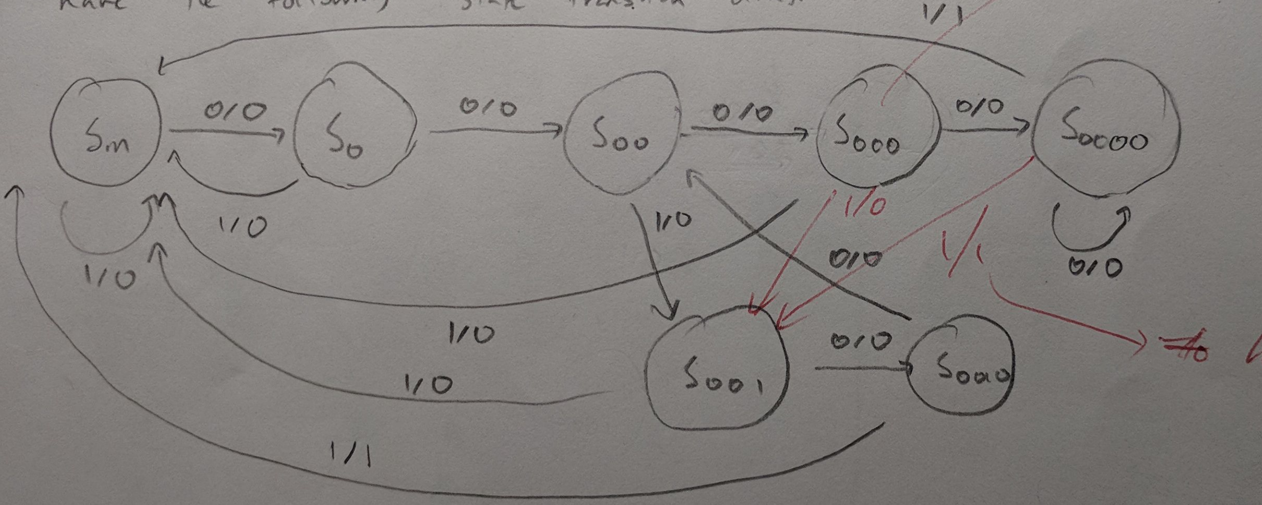
- Input: $x(t) \in \{0,1\}$
- Output: $z(t) \in \{0,1\}$
- Function: $z(t) = \begin{cases} 1 & \text{, if } x(\text{t-4, t}) = \text{00-01} \\ 0 & \text{, otherwise} \end{cases}$

3 inputs to flipflop

A	B	C	Q_{next}
0	0	0	0
0	0	1	1
0	1	0	Q
0	1	1	Q'
1	0	0	0
1	0	1	1
1	1	0	Q'
1	1	1	Q

000101

have the following state transition diagram



-2pts

to handle overlap pattern 0000101

S_{in} = 000

S_0 = 001

S_{00} = 010

S_{000} = 011

S_{001} = 100

S_{0010} = 101

S_{0000} = 110

State transition table

PS	Input	
$q_2 q_1 q_0$	x=0	x=1
000	001,0	000,0
001	010,0	000,0
010	011,0	100,0
011	110,0	000,0
100	101,0	000,0
101	010,0	000,1
110	110,0	000,1

Problem 5 (20 points)

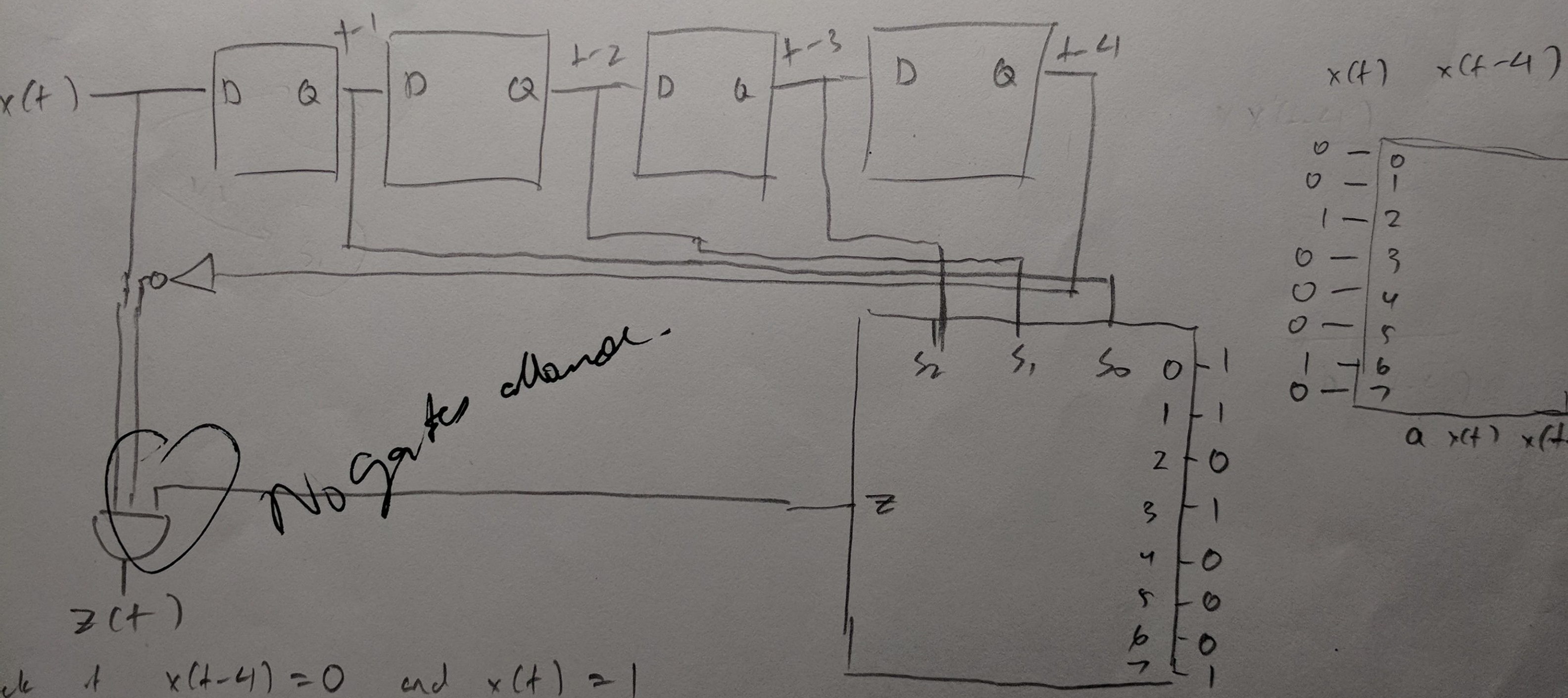
Design a system that detects the pattern of consecutive 0s followed by consecutive 1s in the most recent five bits from the input stream. In other words, the input pattern must have 0 in the start, 1 in the end, and change from 0 to 1 only once.

For example, if the input sequence x(t-10, t) = 01001000100, the output z(t) should be 0 because x(t-4, t) = 00100 does not have 1 in the end. The past output z(t-1) is also 0 because x(t-5, t-1) = 00010 does not have 1 in the end.

However, if the input sequence x(t-10, t) = 01111100001, the output z(t) should be 1 because x(t-4, t) = 00001 consists of four consecutive 0s and consecutive 1s – the last bit 1. You may assume that flip-flops are initialized to 0, so the output z(t-10, t) = 01111000001.

(3 select inputs)

- Use only flip-flops and 8:1 multiplexers. No other gates are allowed.



ck if x(t-4) = 0 and x(t) = 1

Then check if middle is 000, 001, 011, or 111

S= x(t-3)'x(t-2)'x(t-1)' + x(t-3)'x(t-2)'x(t-1) + x(t-3)'x(t-

+ x(t-3)x(

output 1 if both of these conditions are true