Name Elan [illegible] Markowitz

First Last

Student ID # 904485758

Section Dis 2B

University of California Los Angeles

Computer Science Department

CSM51A Midterm 2

Fall Quarter 2017

20th November 2017

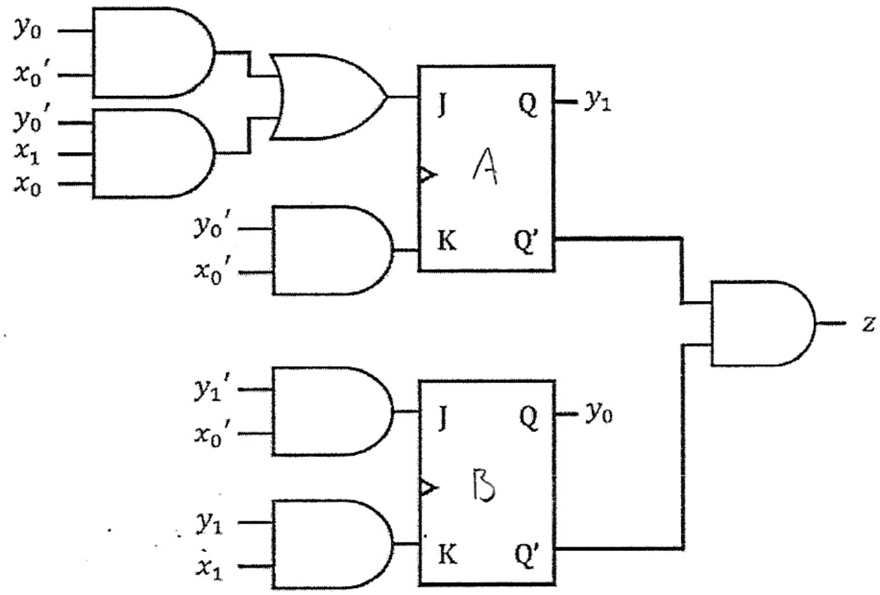
This is a closed book exam. Absolutely nothing is permitted except pen, pencil and eraser to write your solutions. Any academic dishonesty will be prosecuted to the full extent permissible by university regulations.

Time allowed 100 minutes

Problem (Possible points)	Points
1 (20)	20
2 (20)	2
3 (20)	20
4 (20)	20
5 (20)	0
Total (100)	62

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.



$J_A = x_0' y_0 + x_0 x_1 y_0'$

$K_A = x_0' y_0'$

$J_B = y_1' x_0'$

$K_B = y_1 x_1$

$y_1 = Q_A$

$y_0 = Q_B$

$z = Q_B' Q_A'$

Q_A	Q_B	x_1	x_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1

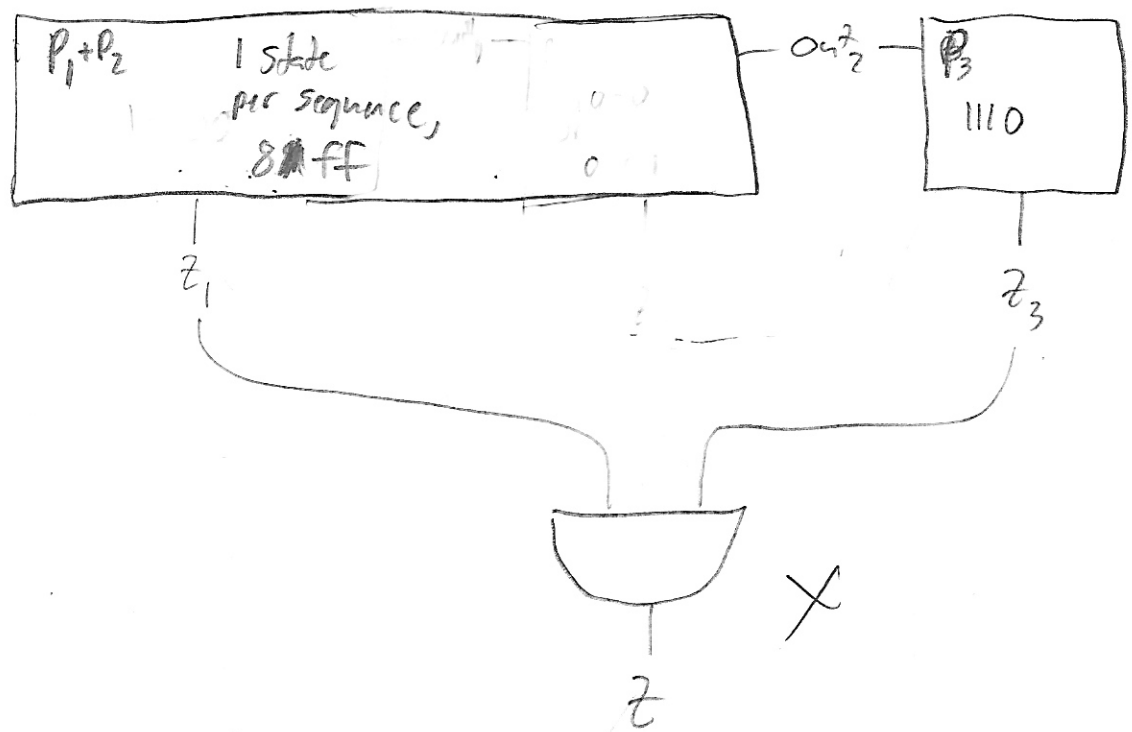
Q_A	Q_B	X_1	X_0	J_A	K_A	J_B	K_B	$Q_A(t+1)$	$Q_B(t+1)$	Z
0	0	0	0	0	1	1	0	0	1	1
0	0	0	1	0	0	0	0	0	0	1
0	0	1	0	0	1	1	0	0	1	1
0	0	1	1	1	0	0	0	1	0	1
0	1	0	0	1	0	1	0	1	1	0
0	1	0	1	0	0	0	0	0	1	0
0	1	1	0	1	0	1	0	1	1	0
0	1	1	1	0	0	0	0	0	1	0
1	0	0	0	0	1	0	0	0	0	0
1	0	0	1	0	0	0	0	1	0	0
1	0	1	0	0	1	0	1	0	0	0
1	0	1	1	1	0	0	1	1	0	0
1	1	0	0	1	0	0	0	1	1	0
1	1	0	1	0	0	0	0	1	1	0
1	1	1	0	1	0	0	1	1	0	0
1	1	1	1	0	0	0	1	1	0	0

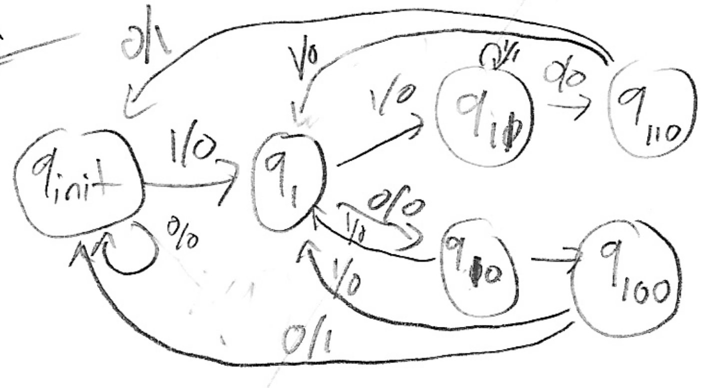
Problem 2 (20 points)

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

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.





Problem 2 (Extra sheet)

8 ff for newest 8 bits

1 state per sequence

(p3) last pattern recognizer uses 2 ff
to remember 4 states (

X

(2)

20

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 $\mathbf{z}(t-13, t)$ = UCLAUSCUSCUCLA.
2. If the input sequence of $x(t-9, t)$ = 1111001111, the output sequence of $\mathbf{z}(t-9, t)$ = UCLAUSUCLA.

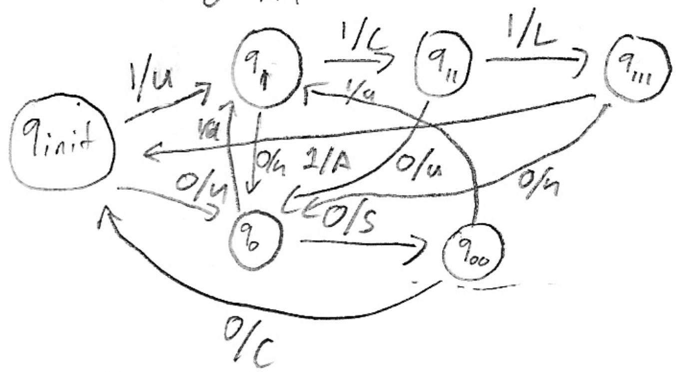
Use the least number of bits for character encoding, and the minimal number of flip-flops and 4:1 multiplexers.

encoding

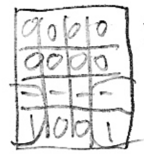
Letter	representation $z_2 z_1 z_0$
U	0 0 0
C	0 0 1
L	0 1 0
A	0 1 1
S	1 0 0
C	1 0 1

✓

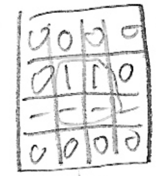
State diagram



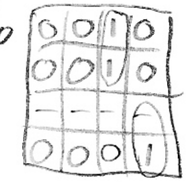
	$Q_A Q_B Q_C$
q_{init}	000
q_1	0 0 1
q_{11}	0 1 0
q_{111}	0 1 1
q_0	1 0 0
q_{00}	1 0 1



Z_2 $= X'Q_C$



Z_1 $= XQ_B$

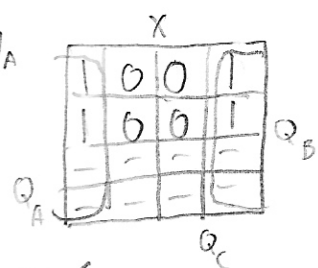


Z_0 $= XQ_A'Q_C + X'Q_AQ_C$

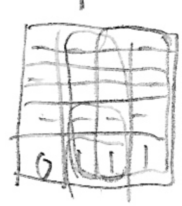
Problem 3 (Extra sheet)

State table

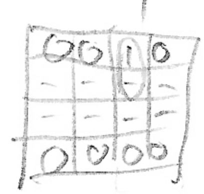
	$Q_A Q_B Q_C$	X	$Q_A^+ Q_B^+ Q_C^+$ (NS)	$Z_2Z_1Z_0$	J_A K_A	J_B K_B	J_C K_C
q_{init}	0 0 0	0	1 0 0	0 0 0	1 –	0 –	0 –
	0 0 0	1	0 0 1	0 0 0	0 –	0 –	1 –
q_1	0 0 1	0	1 0 0	0 0 0	1 –	0 –	– 1
	0 0 1	1	0 1 0	0 0 1	0 –	1 –	– 1
q_{11}	0 1 0	0	1 0 0	0 0 0	1 –	– 1	0 –
	0 1 0	1	0 1 1	0 1 0	0 –	– 0	1 –
q_{111}	0 1 1	0	1 0 0	0 0 0	1 –	– 1	– 1
	0 1 1	1	0 0 0	0 1 1	0 –	– 1	– 1
q_0	1 0 0	0	1 0 1	1 0 0	– 0	0 –	1 –
	1 0 0	1	0 0 1	0 0 0	– 1	0 –	1 –
q_{00}	1 0 1	0	0 0 0	1 0 1	– 1	0 –	– 1
	1 0 1	1	0 0 1	0 0 0	– 1	0 –	– 0
–	1 1 0	0	—	—	—		
	1 1 0	1	—	—	—		
	1 1 1	0	—	—	—		
	1 1 1	1	—				



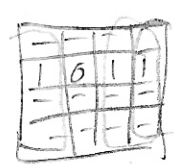
$J_A = X'$



$K_A = X + Q_C$

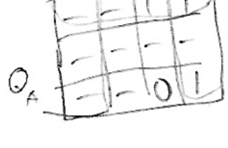


$J_B = XQ_CQ_A'$

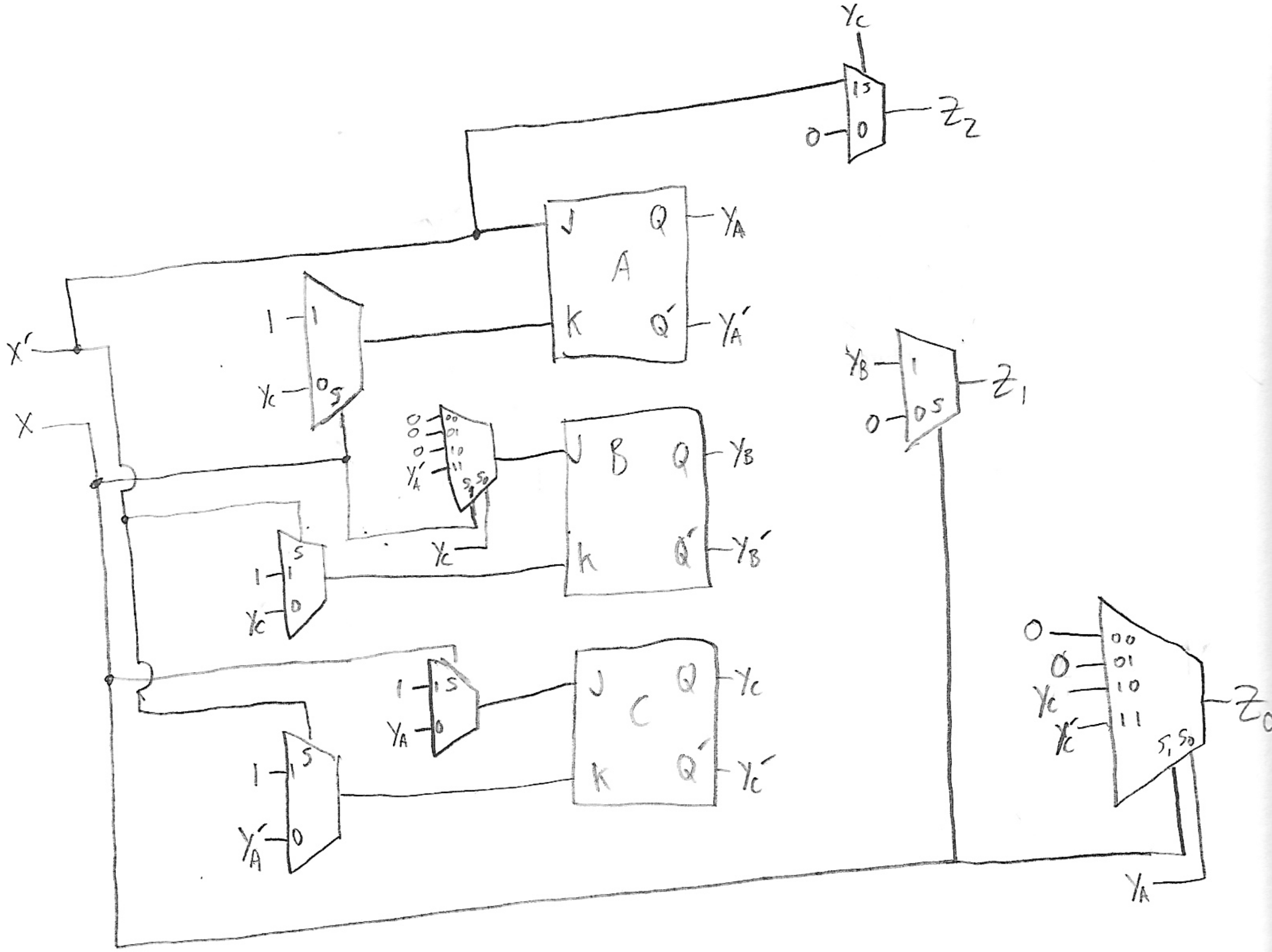


$K_B = Q_C + X'$

$J_C = X + Q_A$



$K_C = Q_A' + X'$



Yc
Z2
J Q YA
A
K Q' YA'
X'
X
Yc
YB
Z1
J B Q YB
K Q' YB'
J C Q Yc
K Q' Yc'
YA
YA'
Z0
YA

Problem 4 (20 points)

NOR NOR ⇒ POS

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

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

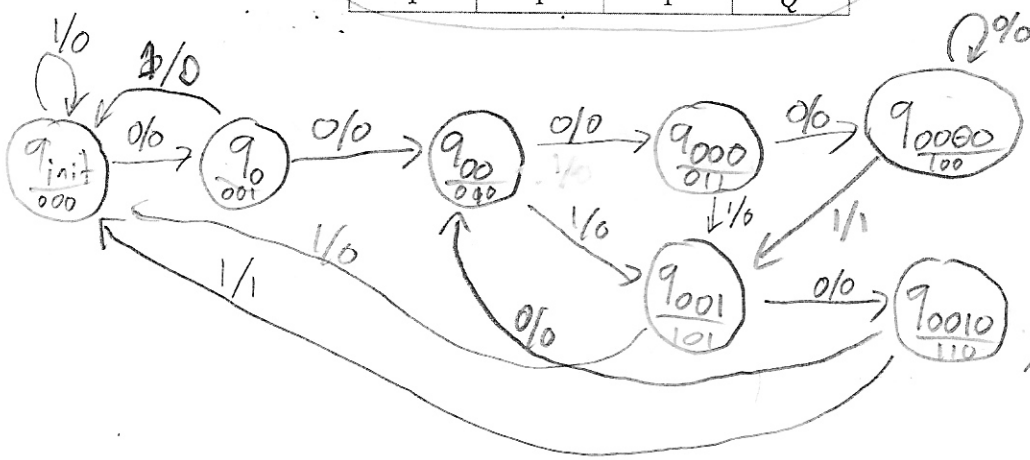
A	B	C	Q_{next}
0	0	0	0
0	0	1	1
0	1	0	Q
0	1	1	Q'
1	0	0	0
1	0	1	1
1	1	0	Q'
1	1	1	Q

0→0
000
100
010
111

A = 1
0→0
0→1
1→0
1→1

So B can be either 1 or –
B is just 1

C will be less complicated, Alternatively using 4 Dffs is the easiest option.

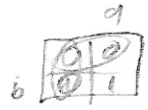


+/2

	$Q_A\ Q_B\ Q_C$
q_{init}	0 0 0
q_0	0 0 1
q_{00}	0 1 0
q_{000}	0 1 1
q_{0000}	1 0 0
q_{001}	1 0 1
q_{0010}	1 1 0
—	1 1 1

ABC flip flop excitation

	NS 0	NS 1
PS 0	000, 100, 010, 111	001, 101, 011, 110
PS 1	000, 100, 011, 110	001, 101, 010, 111



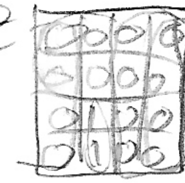
$((b') + (a'))'$ 

$ab = (a' + b')'$

Problem 4 (Extra sheet)

State table

Q_A	Q_B	Q_C	X	Q_A^+	Q_B^+	Q_C^+ (NS)	Z	A_A	B_A	C_A	A_B	B_B	C_B	A_C	B_C	C_C
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	1	1	0	0	0	0	0	0	1	0	0	1
0	1	0	1	1	0	1	0	0	0	1	0	0	0	0	0	1
0	1	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0
0	1	1	1	1	0	1	0	0	0	1	0	0	0	0	0	1
1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0
1	0	0	1	1	0	1	1	0	0	1	0	0	0	0	0	1
1	0	1	0	1	1	0	0	0	0	1	0	0	1	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0
1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
1	1	1	0	–	–	–	–	–	–	–						
1	1	1	1	–	–	–	–	–	–	–						

Z 

$x Q_A Q_C'$

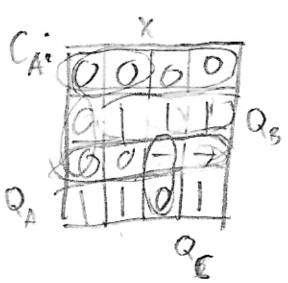
$(x' + Q_A' + Q_C)'$

I didn't intend to make such a difficult question.

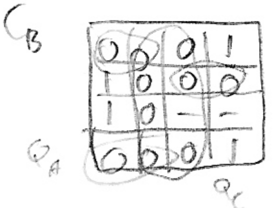
ok. +2

$A_A = A_B = A_C = B_A = B_B = B_C = 0$

NOR-NOR => POS

C_A: 

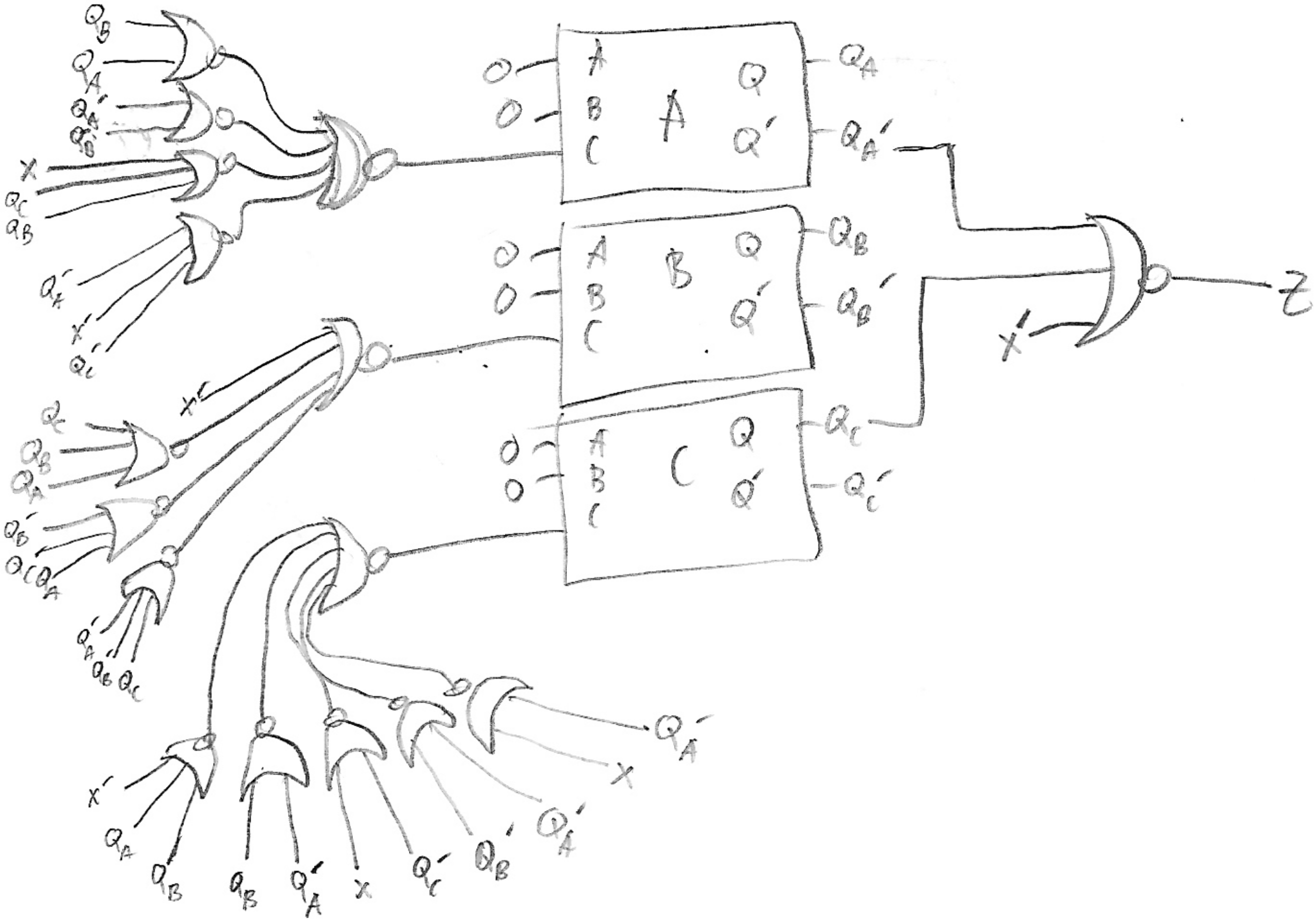
$C_A = ((Q_B' + Q_A')(Q_B' + Q_A')(Q_C' + X' + Q_B)(Q_A + X + Q_C))'$

C_B 

$C_B = ((X')(Q_C' + Q_B' + Q_A')(Q_B + Q_C + Q_A')(Q_A + Q_B + Q_C'))$

C_C 

$((X + Q_A' + Q_B')(Q_B' + Q_A)(X' + Q_C)(Q_B + Q_A)(X' + Q_A))'$


A
B
C
Q
Q'
Q_A
Q_A'
Q_B
Q_B'
Q_C
Q_C'
X
X'
Z

Problem 5 (20 points)

Design a system that detects the pattern of consecutive 0s followed by consecutive 1s in the most recent <u>five</u> 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 $z(t-10, t) = \underline{0111}1000001$.

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