NAME: LillAnne
ID: ____________

University of Victoria
Department of Computer Science
CSC 355 Digital Logic and Computer Design
ASSIGNMENT 3

SHOULD BE DUE: Thursday November 12 AT BEGINNING OF CLASS
BUT WILL BE ACCEPTED UNTIL: Thursday December 3 AT BEGINNING OF CLASS

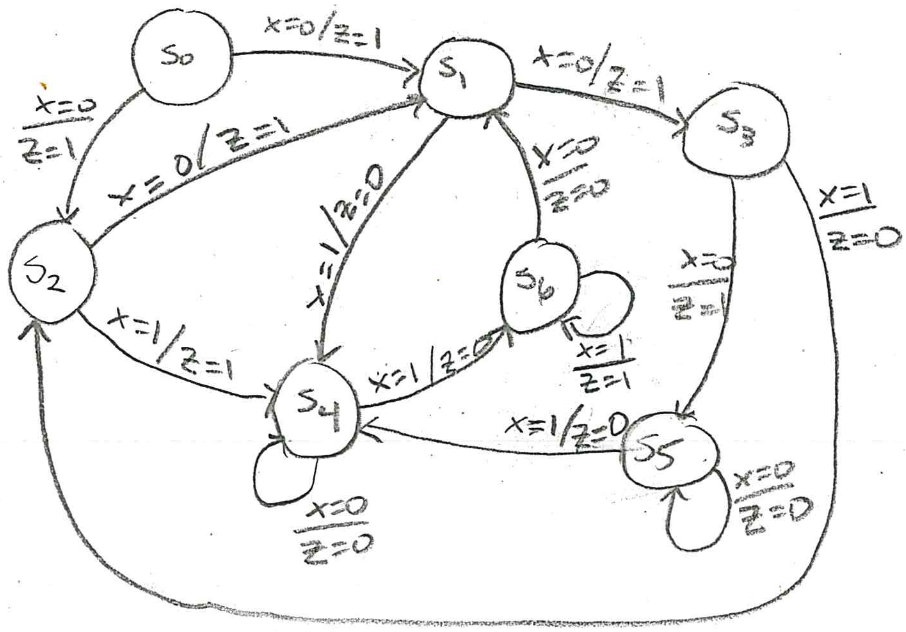
Neatness Counts!

1. Given the next state and output table shown below, give an implementation of the clocked sequential machine using 3 RS flip-flops (labeled A, B and C) plus any other gates you require, based on the state assignment S_0 = 001, S_1= 000, S_2 = 100, S_3 = 110, S_4 = 010, S_5 = 111, S_6 = 011. Make the best use of don't cares. Show:
   a) A state diagram.
   b) The next state maps for each flip-flop and the output map
   c) The maps for each of the six flip flop input equations that are required.
   d) The minimized logic functions derived from above.

	Next state		Output	
state	x = 0	x = 1	x = 0	x = 1
S_0 001	S_1=000	S_2=100	1	1
S_1 000	S_3= 110	S_4=010	1	0
S_2 100	S_1= 000	S_4=010	1	1
S_3 110	S_5=111	S_2=100	1	0
S_4 010	S_4=010	S_6=011	0	0
S_5 111	S_5=111	S_4=010	0	0
S_6 011	S_1=000	S_6=011	0	1
S_7=101	xxx	xxx	x	x

← state assignments included here

State Diagram



Next State Maps

BC \ XA	00	01	11	10
00	1	0	0	0
01	0	x	x	1
11	0	1	0	0
10	0	1	1	0

A^+

BC \ XA	00	01	11	10
00	1	0	1	1
01	0	x	x	0
11	0	1	1	1
10	1	1	0	1

B^+

BC \ XA	00	01	11	10
00	0	0	0	0
01	0	x	x	0
11	0	1	0	1
10	0	1	0	1

C^+

Maps for inputs to flip flops [6]

BC \ XA	00	01	11	10
00	1			
01		x	x	1
11		x		
10		x	x	

Sa

BC \ XA	00	01	11	10
00		1	1	x
01	x	x	x	
11	x		1	x
10	x			x

could circle this 1 with the x above. <alternate answer>

Ra

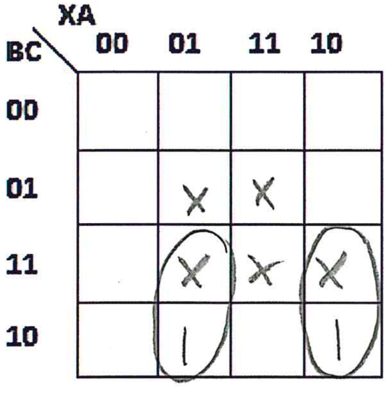
could circle this 1 with the x below <alternate answer>

BC \ XA	00	01	11	10
00	1		1	1
01		x	x	
11		x	x	x
10	x	x		x

Sb

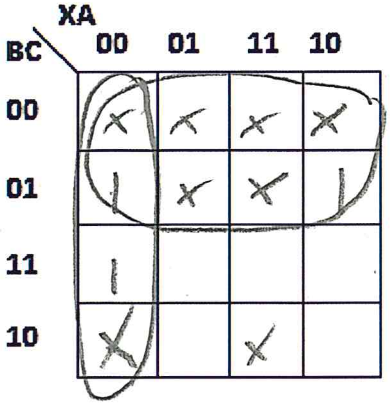
BC \ XA	00	01	11	10
00		x		
01	x	x	x	x
11	1			
10		1		

Rb



BC \ XA	00	01	11	10
00				
01		X	X	
11		X	X	X
10		1		1

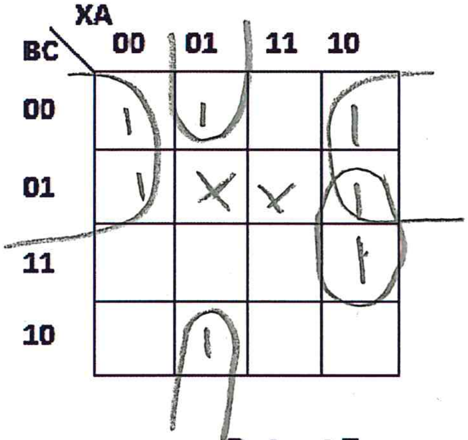
Sc



BC \ XA	00	01	11	10
00	X	X	X	X
01	1	X	X	1
11	1			
10	X		X	

Sc

Output Z



BC \ XA	00	01	11	10
00	1	1		1
01	1	X	X	1
11				1
10		1		

Output Z

alternate answer
$R_a = A\bar{B} + XAC$

Input/output functions

$S_a = \bar{A}\bar{B}\bar{C}\bar{X} + \bar{B}CX$ $\quad R_a = A\bar{B} + XBC$

alternate answer
$S_b = \bar{A}\bar{C} + XA\bar{B}$ →

$S_b = \bar{A}\bar{C} + \bar{B}\bar{C}X$ $\quad R_b = \bar{A}C\bar{X} + A\bar{B}\bar{C}X$

$S_c = A\bar{B}\bar{X} + \bar{A}B\bar{X}$ $\quad R_c = \bar{B} + \bar{A}\bar{X}$

$Z = \bar{A}\bar{B} + A\bar{B}\bar{X} + \bar{A}CX$

2. Design a finite state machine as a clocked Mealy sequential network with one input **X** and one output **Z**. The machine is a recognizer that has an output trigger sequence and a reset output to zero sequence. If the input sequence '0 1 1 0' occurs then the output changes to '1' coincident with the last bit of the sequence. The output remains at '1' until the reset sequence '0 1 0' is received on the input and, in this case, the output changes to '0' coincident with the last bit of the sequence. Initially, output **Z** is '0'.

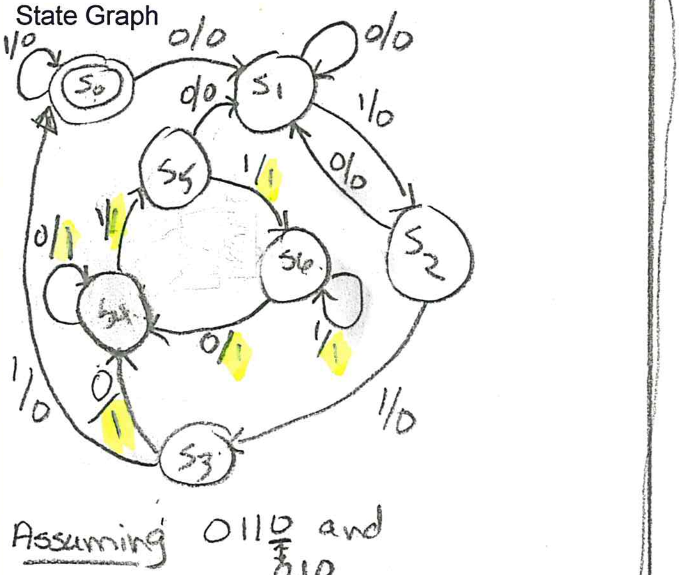
There are 2 options, depending on whether or not you assume the 0 that ends 011$\underline{0}$ can overlap w the 0 that starts $\underline{0}$10.

For example,

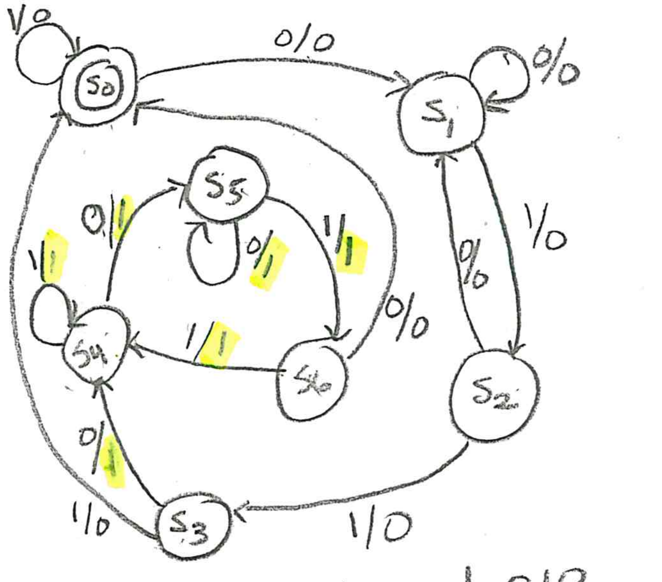
```
X = 0 0 1 0 0 0 1 1 0 0 1 0 1 0 1 1 0 0 0 0 0 1 0 1
Z = 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 1 1 1 0 0
```

For the design, provide a state graph, the corresponding state table and the output table. Number the states **S₀**, **S₁**, ... Maximum marks will be given for a correct solution with the minimum number of states. **Do not** continue the design beyond this point. (In particular, there is no need to implement using flip-flops.

State Graph



Assuming 011$\underline{0}$ and $\underline{0}$10 can overlap



Assumming 011$\underline{0}$ and $\underline{0}$10 cannot overlap

State Table and Output Table

Present	Next		Output	
	x=0	x=1	x=0	x=1
S_0	S_1	S_0	0	0
S_1	S_1	S_2	0	0
S_2	S_1	S_3	0	0
S_3	S_4	S_0	1	0
S_4	S_4	S_5	1	1
S_5	S_1	S_6	0	1
S_6	S_4	S_6	1	1
S_7	S_x	S_x	x	x

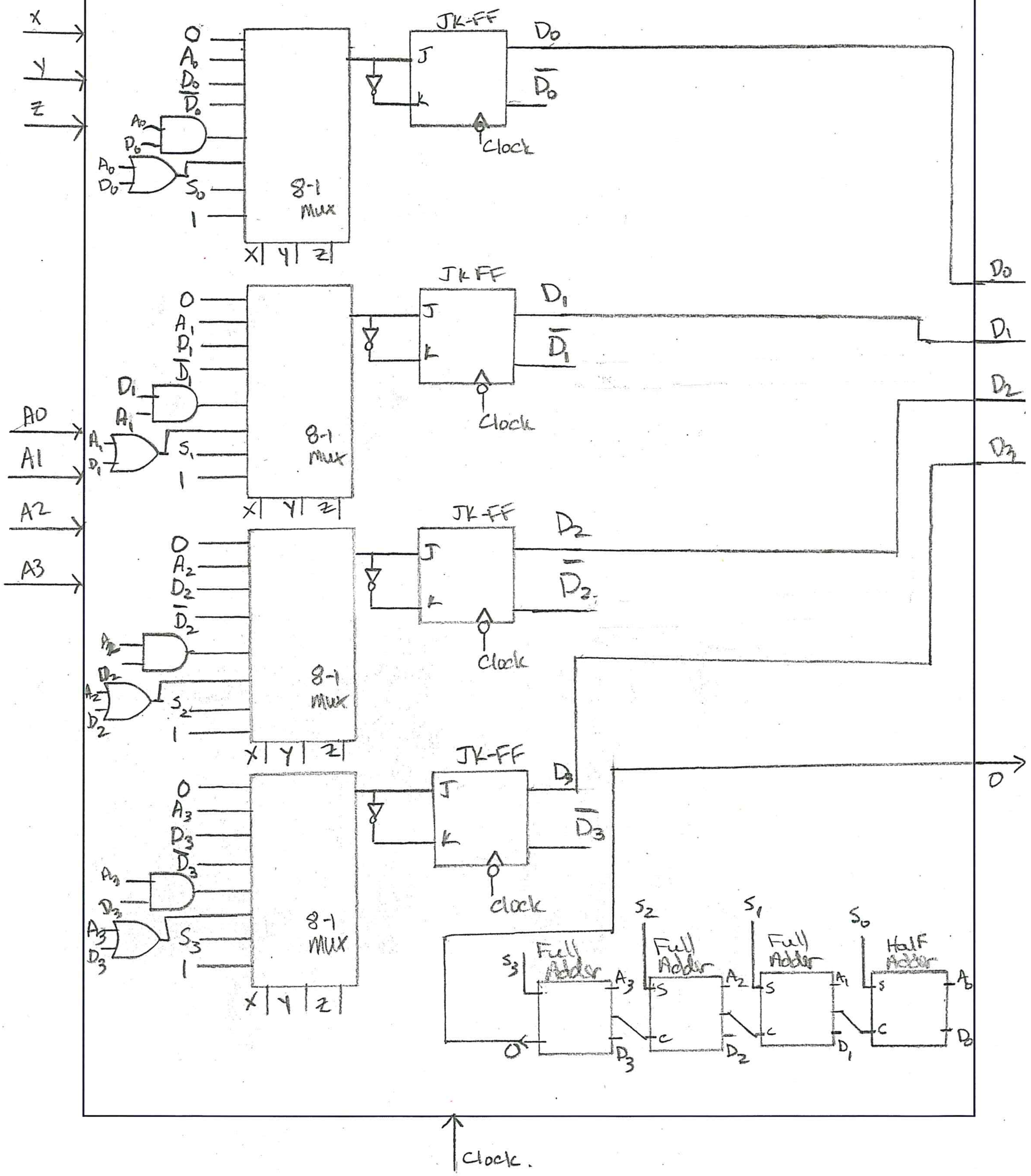
Present	Next		Output	
	x=0	x=1	x=0	x=1
S_0	S_1	S_0	0	0
S_1	S_1	S_2	0	0
S_2	S_1	S_3	0	0
S_3	S_4	S_0	1	0
S_4	S_5	S_4	1	1
S_5	S_5	S_6	1	1
S_6	S_0	S_4	0	1
S_7	S_x	S_x	x	x

3. Design a four-bit ALU/Register with the functional behaviour described in the table. It has three control inputs **X**, **Y** and **Z** and one 4 bit data input **A(3), A(2), A(1), A(0)** It has one 4 bit data output **D(3), D(2), D(1), D(0)** and one output **O** to indicate arithmetic overflow. Use master slave trailing edge J-K flip-flops for your design.

X	**Y**	**Z**	*Function*
0	0	0	Reset Register to (0000)
0	0	1	Parallel Load from A
0	1	0	Maintain Stored Value
0	1	1	Complement Stored Value
1	0	0	Bit-Wise **AND** of A and D
1	0	1	Bit-Wise **OR** of A and D
1	1	0	**ADD** (A + D)
1	1	1	Set Register to (1111)

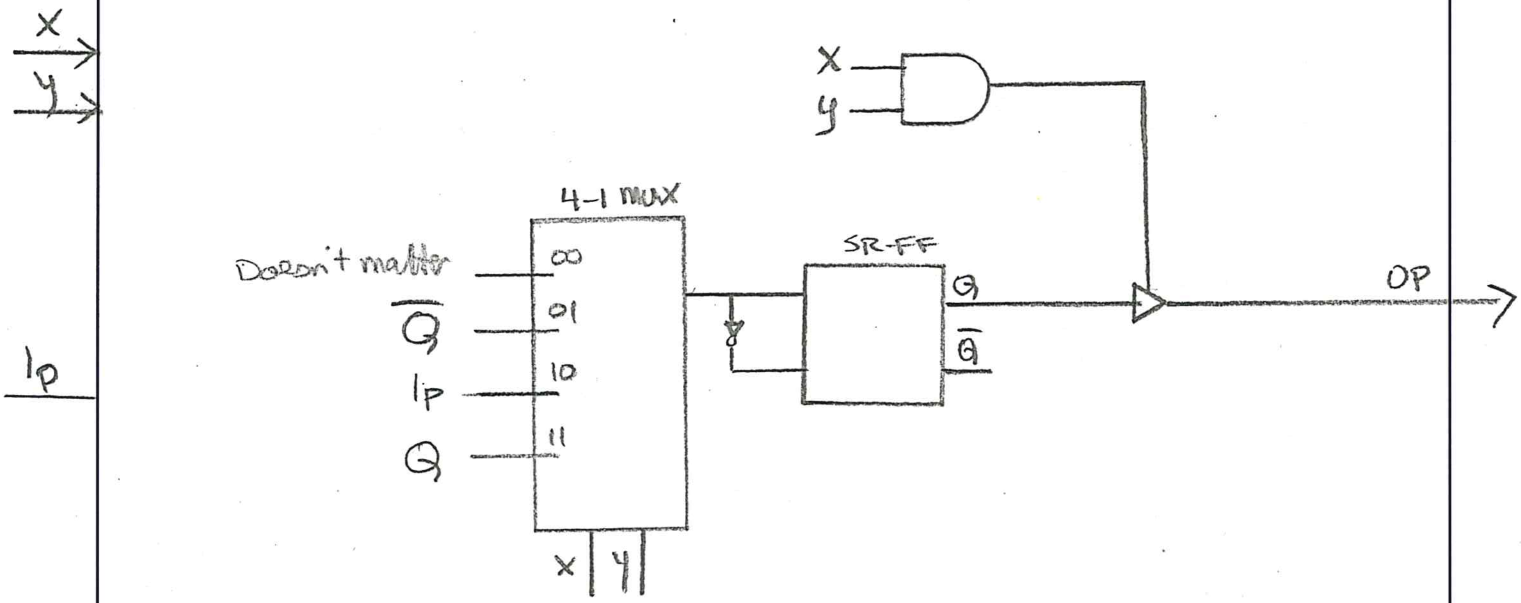
Derive a schematic to implement the ALU/Register using J-K, Flip-Flops, Multiplexers, Full-Adders, Inverters, and any other 2-input gates you need. Keep the design as simple as possible.

<see next page>


X
Y
Z
A0
A1
A2
A3
O
A_0
D_0
$\overline{D_0}$
A_0
D_0
A_0
D_0
S_0
1
8-1 mux
X Y Z
JK-FF
J
K
Clock
D_0
$\overline{D_0}$
O
A_1
D_1
$\overline{D_1}$
D_1
A_1
A_1
D_1
S_1
1
8-1 mux
X Y Z
JK FF
J
K
Clock
D_1
$\overline{D_1}$
O
A_2
D_2
$\overline{D_2}$
A_2
D_2
A_2
D_2
S_2
1
8-1 mux
X Y Z
JK-FF
J
K
Clock
D_2
$\overline{D_2}$
O
A_3
D_3
$\overline{D_3}$
A_3
D_3
A_3
D_3
S_3
1
8-1 mux
X Y Z
JK-FF
J
K
clock
D_3
$\overline{D_3}$
D_0
D_1
D_2
D_3
O
S_3
Full Adder
A_3
D_3
O
S_2
Full Adder
S
C
A_2
D_2
S_1
Full Adder
S
C
A_1
D_1
S_0
Half Adder
S
C
A_0
D_0
Clock.

4. Design a static RAM memory cell, using a set-reset flip flop for the internal storage and any other devices you require, with the following features. There are two control lines, **X** and **Y**, an input line **Ip**, and an output line **Op** (there is no clock). The values of **X**, **Y** and **Ip** require the following actions to be taken:

   - **X** = 0, **Y** = 0 : the memory cell is not selected (flip-flop contents unchanged, **Op** is to be in the high impedance state **Z**)
   - **X** = 1, **Y** = 1 : read the memory cell (output **Op** is equal to contents of flip-flop)
   - **X** = 1, **Y** = 0 : write to the memory cell (flip-flop contents to be equal to value on **Ip, Op** is to be in the high impedance state **Z**)
   - **X** = 0, **Y** = 1 : toggle the memory cell (invert the flip-flop contents, **Op** is to be in the high impedance state **Z**).



5. A gated latch (G-L FF) behaves as follows:
   - If G = 0, the flip-flop does not change state.
   - If G = 1, the next state of the flip-flop is equal to the value of L, where G and L are the two inputs to the flip flop.

   Derive the characteristic (next- state) equation for the flip-flop.

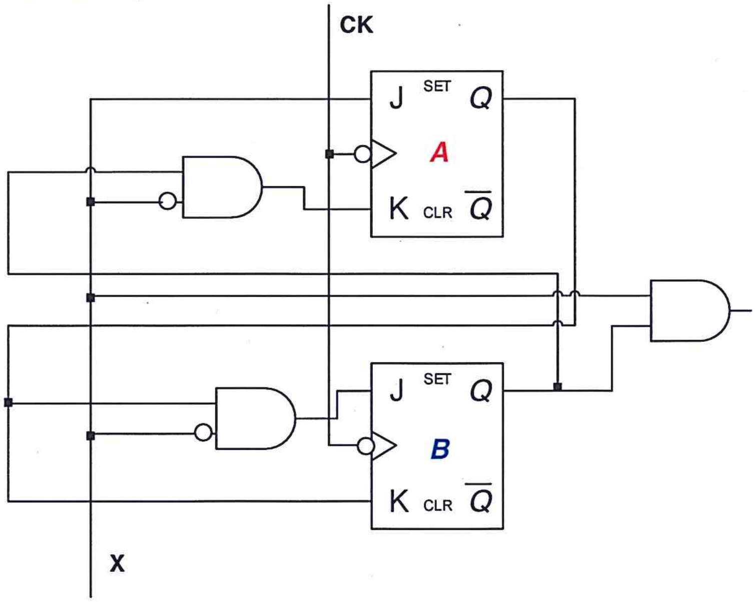
State table

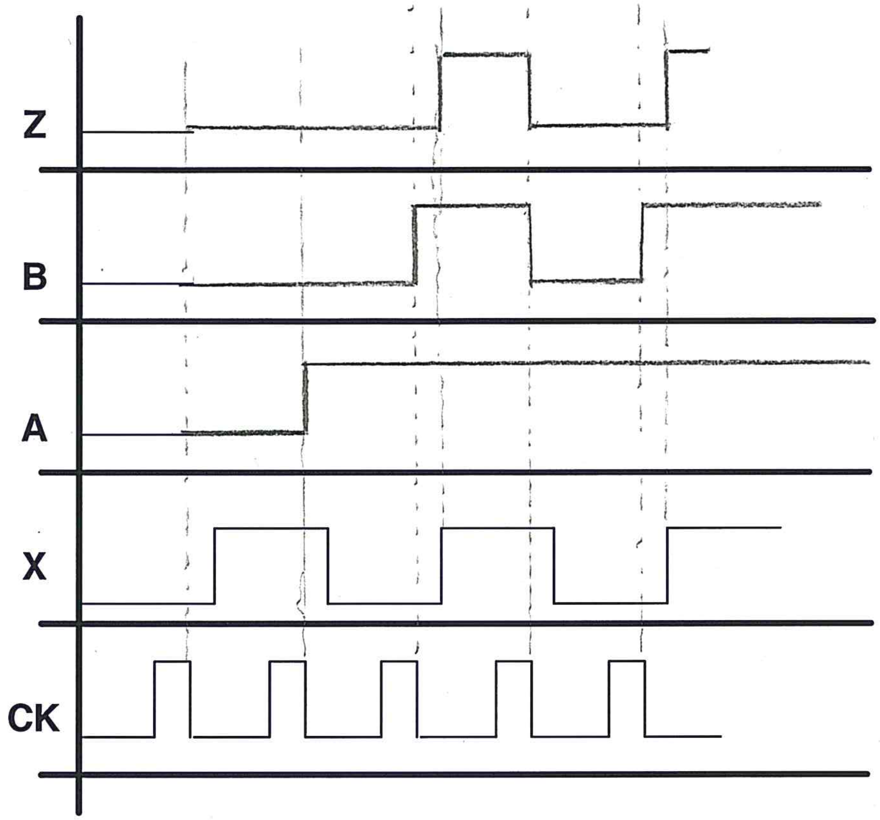
G	L	Q	Q^+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

G \ LQ	00	01	11	10
0	0	1	1	0
1	0	0	1	1

$Q^+ = \bar{G}Q + GL.$

6. Complete the timing diagram for the sequential circuit shown.





7. 4TH EDITION: Page 442 #8-7. 5th EDITION: Page 432 #7-7 **Please inform if 5th edition was used.**

Removed

□4th Edition
□5th Edition

8. 4th EDITION: Pag 442#8.8. 5th EDITION: Page 432 #7-8

a) $\frac{2M}{128K} = \frac{2^{21}}{2^{17}} = 2^4$ BUT the 2M is 8-bit bytes, while the 128K is 2 byte words.

$\therefore$ #chips needed $= 2^4/2 = 2^3 = 8$ chips.

b) $2M = 2^{21}$, So 21 lines are needed to address each byte. (See below), 3 Lines are addressing each byte, so the remaining 18 connect to all chips

c) 8 chips $= 2^3$ chips $\Rightarrow$ 3 chip select lines.