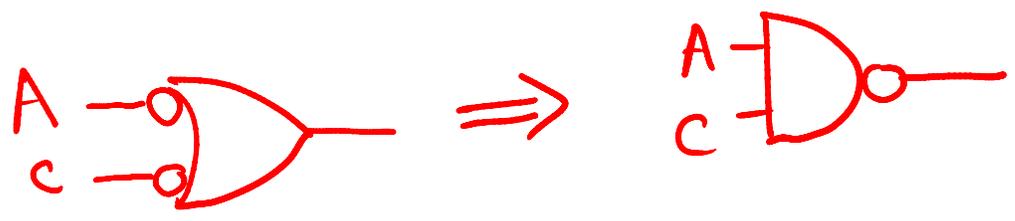
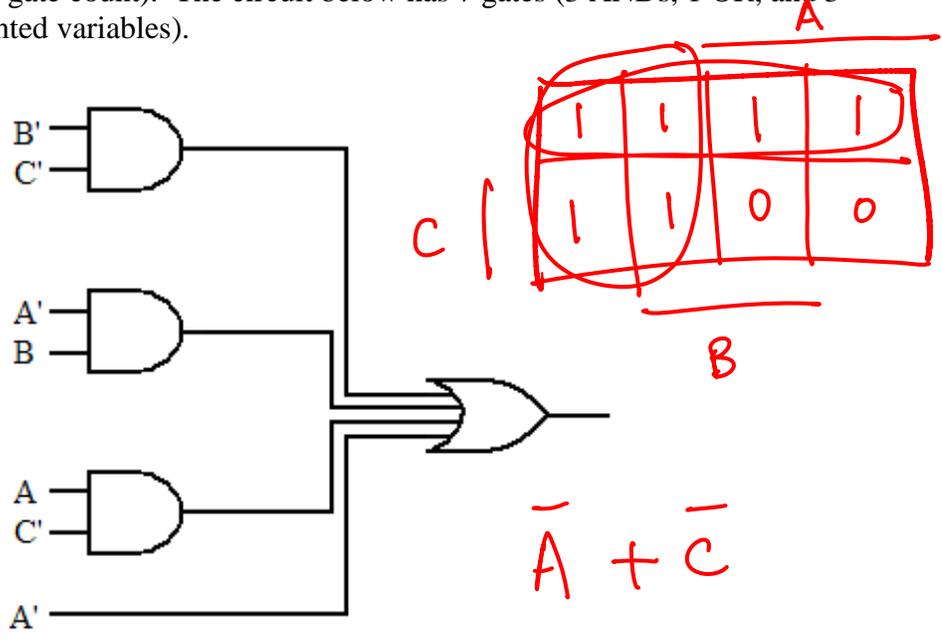
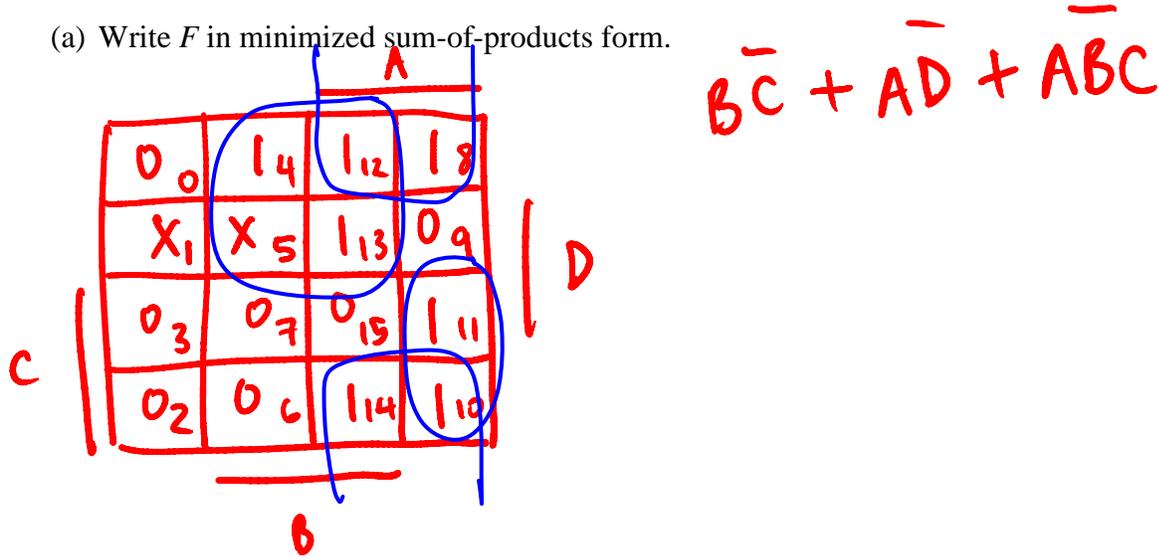


1. (15 points) Draw the smallest possible equivalent circuit to the circuit below. You may use any type of standard gate: AND, OR, NAND, NOR, NOT, XOR, etc... You may NOT assume that you have a variable's complement available as input (i.e., inverting a variable adds one to the total gate count). The circuit below has 7 gates (3 ANDs, 1 OR, and 3 different complemented variables).

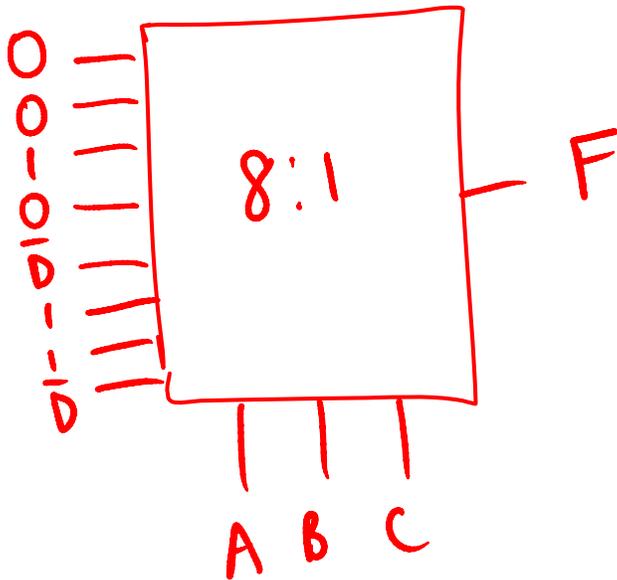


2. (20 points) Let $F(A, B, C, D) = \sum m(4, 8, 10, 11, 12, 13, 14) + d(1, 5)$.

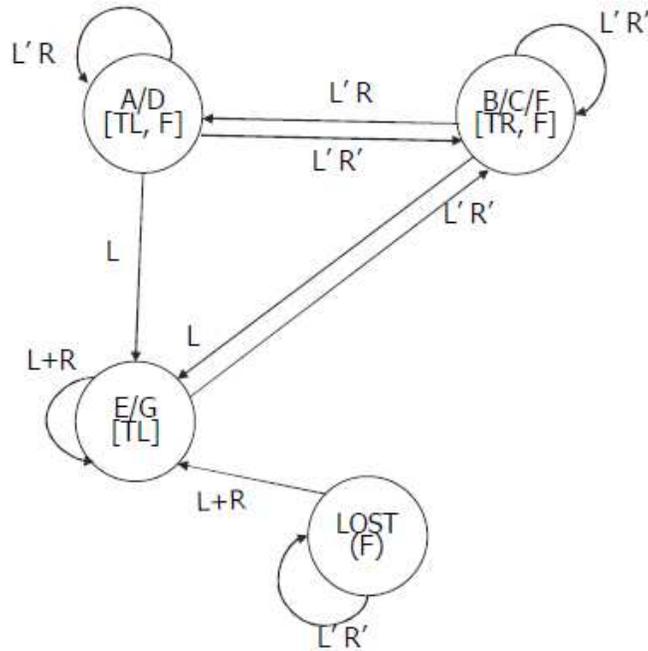
(a) Write F in minimized sum-of-products form.



(b) Implement the minimized sum-of-products form of F using an 8:1 multiplexer.



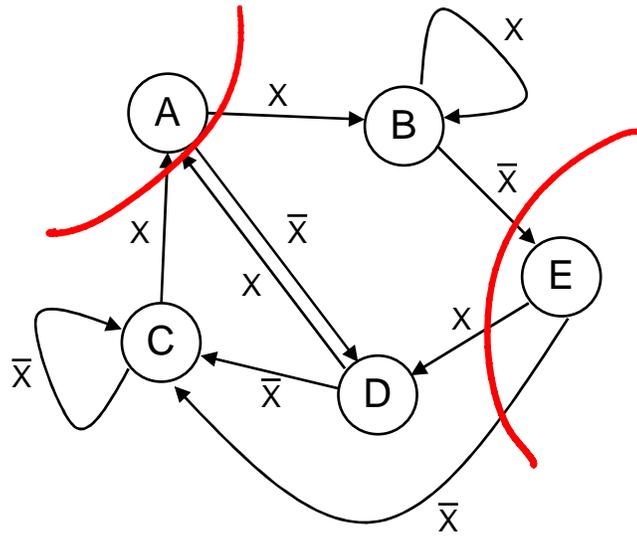
3. (15 points) Below are the FSM and state transition table from the robot ant shown in lecture.



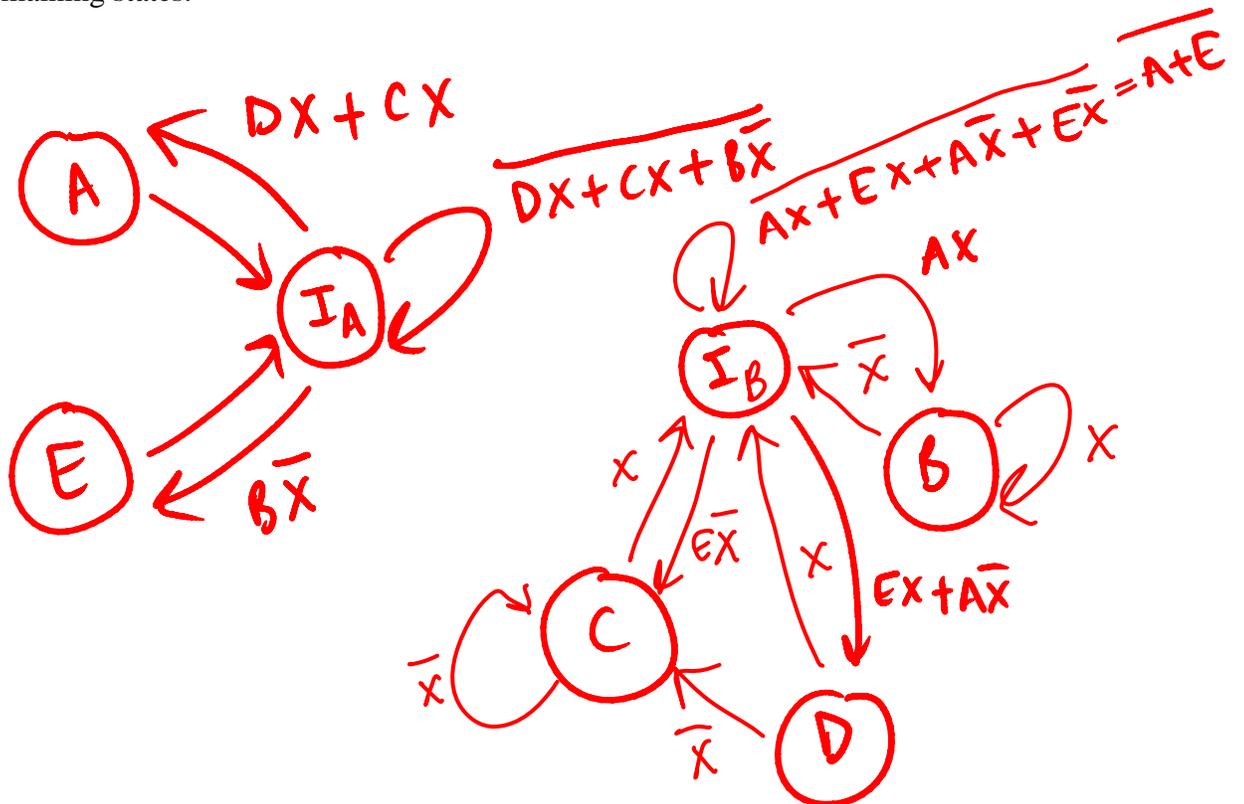
state	L	R	next state	outputs	state		next state		outputs			
					X	Y	X+	Y+	F	TR	TL	
LOST	0	0	LOST	F	0	0	0	0	0	1	0	0
LOST	X	1	E	F	0	0	X	1	0	1	0	0
LOST	1	0	E	F	0	0	1	0	0	1	0	0
E	0	0	B	TL	0	1	0	0	1	1	0	1
E	X	1	E	TL	0	1	X	1	0	1	0	1
E	1	0	E	TL	0	1	1	0	0	1	0	1
A	0	0	B	TL, F	1	0	0	0	1	1	0	1
A	0	1	A	TL, F	1	0	0	1	1	0	0	1
A	1	X	E	TL, F	1	0	1	X	0	1	0	1
B	0	0	B	TR, F	1	1	0	0	1	1	1	0
B	0	1	A	TR, F	1	1	0	1	1	0	1	0
B	1	X	E	TR, F	1	1	1	X	0	1	1	0

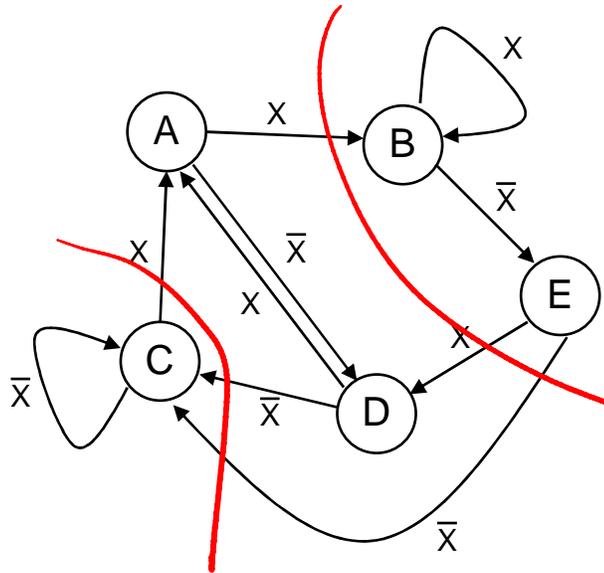
Problem continues on the next page...

4. (30 points) Given the following FSM:

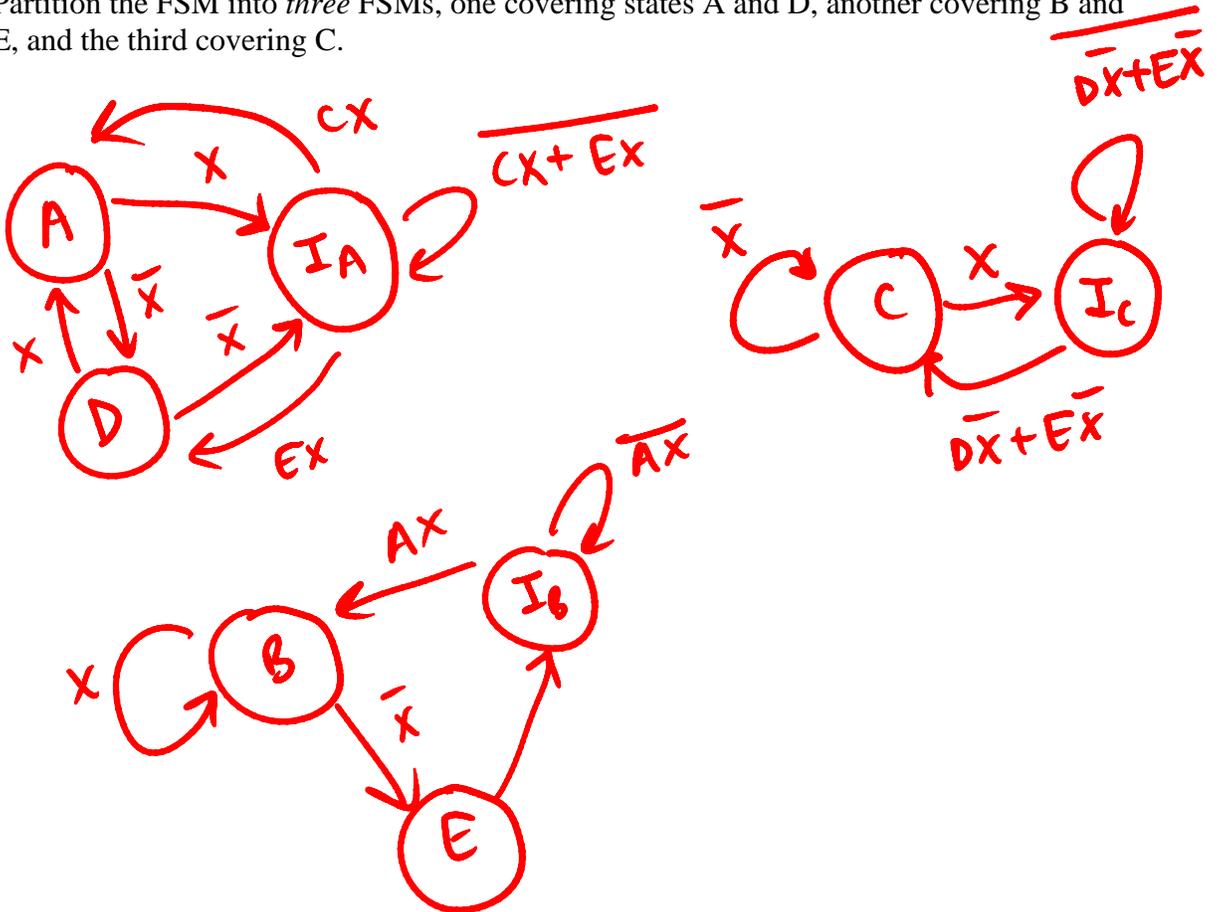


(a) Partition the FSM into two FSMs, one covering states A and E, and the other covering the remaining states.

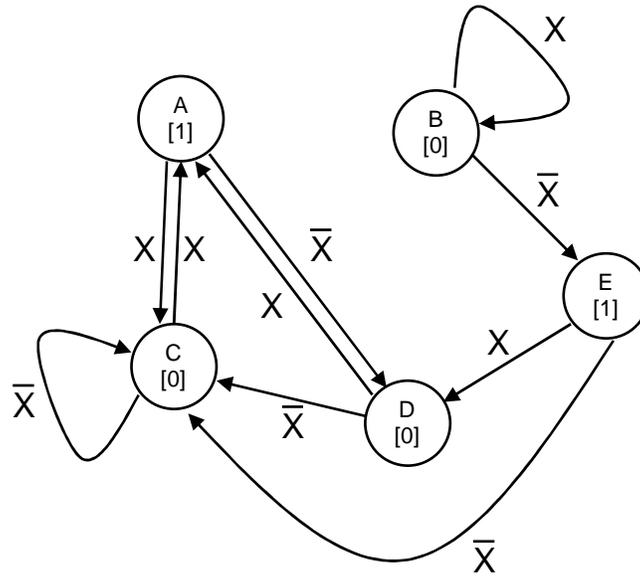




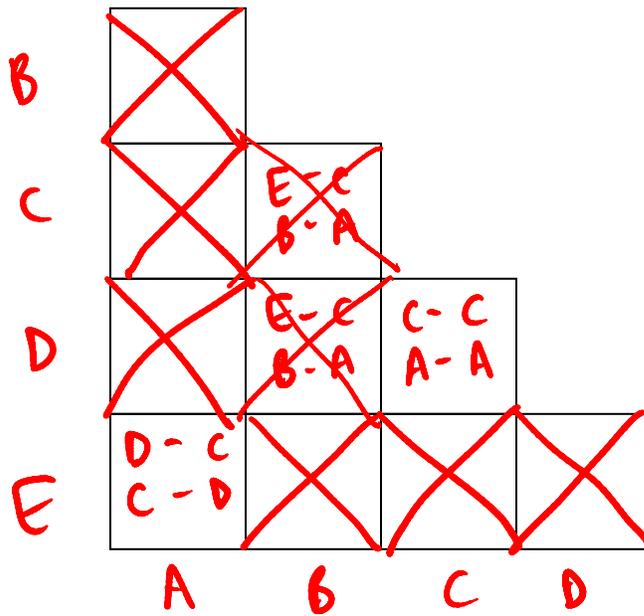
(b) Partition the FSM into *three* FSMs, one covering states A and D, another covering B and E, and the third covering C.



5. (30 points) Given the following FSM:



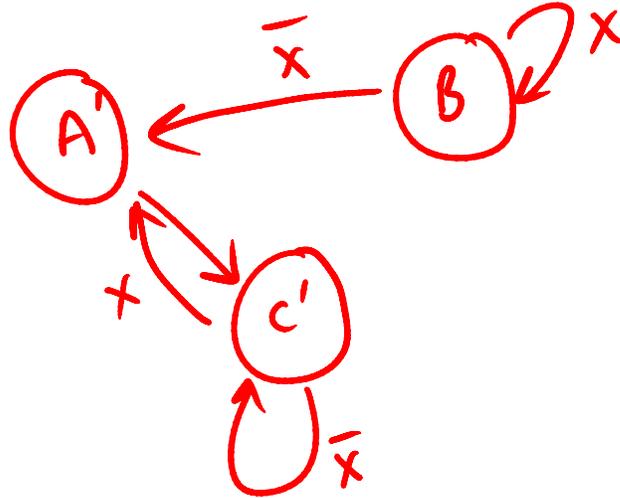
(a) Fill in the following implication chart, listing the order in which you cross off cells in part (b).



(b) List the state pairs (e.g., B-E) in the order in which you crossed them off in (a).

A-B B-E B-C
 A-C C-E B-D
 A-D D-E

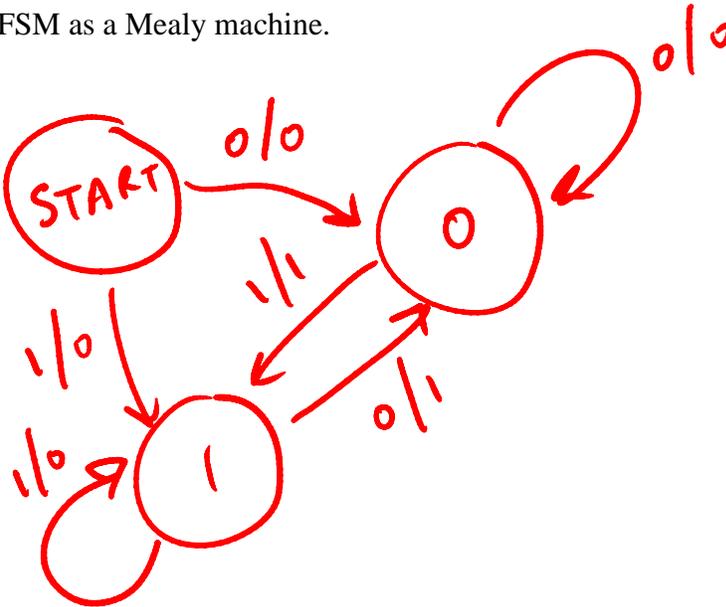
(c) Draw the minimized state diagram.



6. (60 points)

A certain device receives a bit on every clock cycle. If the bit received matches the previous bit, the machine outputs a 0, otherwise it outputs a 1. If there was no previous bit (i.e. the first bit ever received), the device outputs a 0.

(a) Draw the FSM as a Mealy machine.



(b) Using a binary (sequential) encoding, draw the state encoding table and the state transition table.

STATE	ENC	STATE	X	NEXT STATE	OUTPUT
START	00	00	0	01	0
			1	10	0
0	01	01	0	01	0
			1	10	1
1	10	10	0	01	1
			1	10	0
		11	X	XX	X

(c) Write the minimized next-state and output logic.

$$Q_1^+ = X$$

$$Q_0^+ = \bar{X}$$

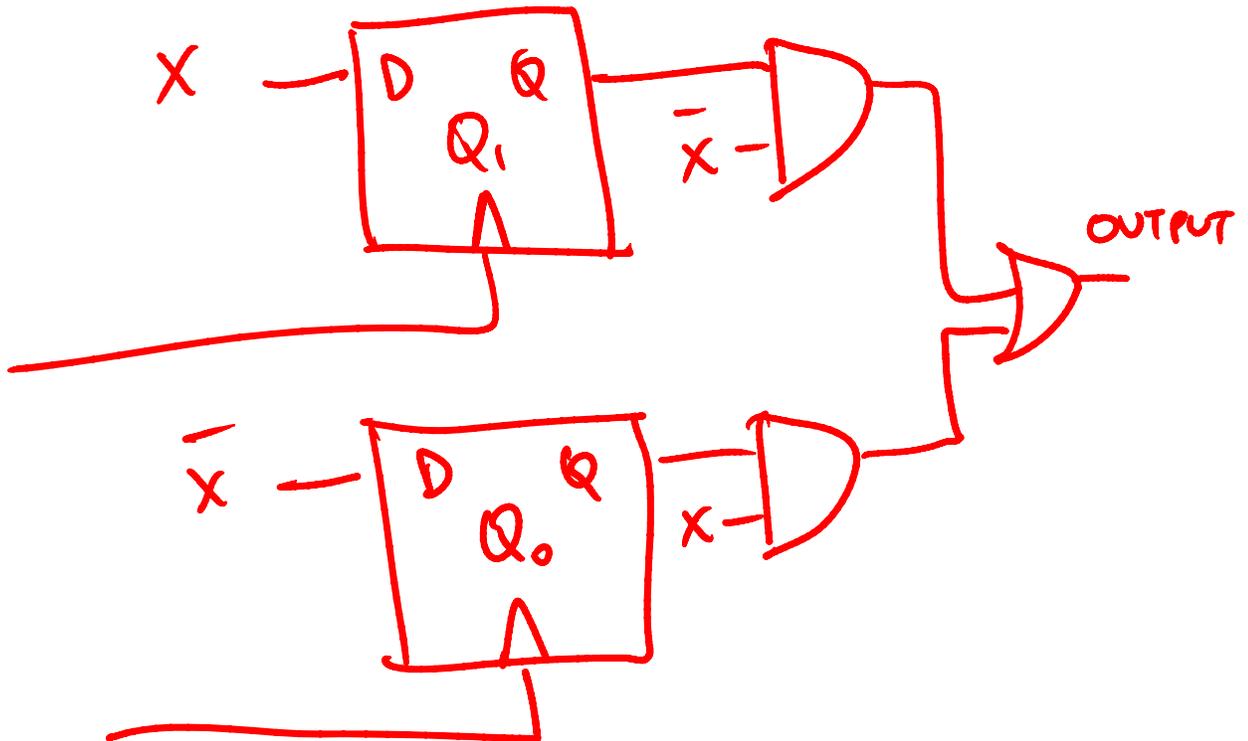
$$\text{OUTPUT} = Q_1 \bar{X} + Q_0 X$$

	Q_1			
Q_1^+	0	0	X	0
X	1	1	X	1

	Q_0	Q_1
Q_0^+	1	1
X	0	0

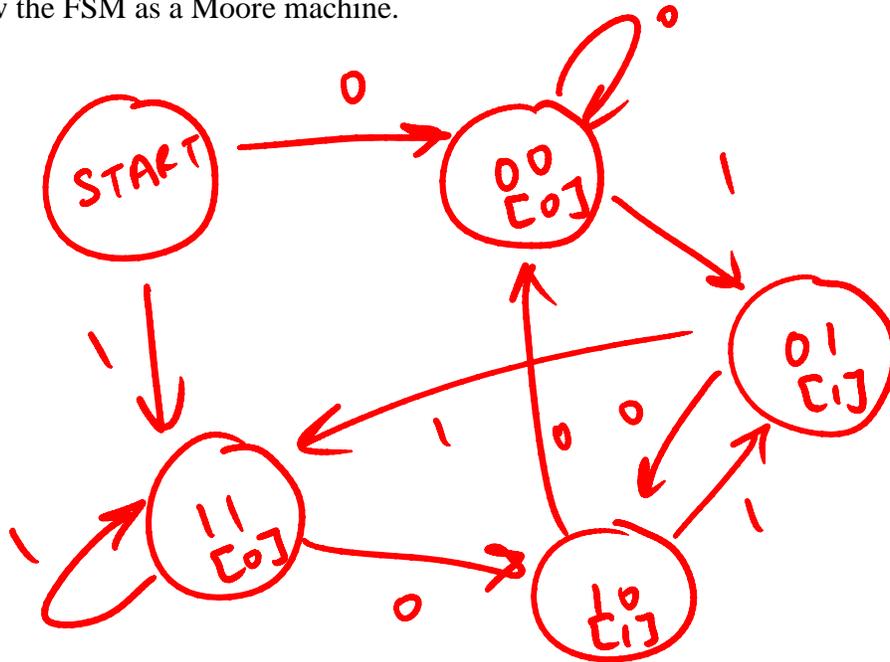
	Q_0	Q_1
OUTPUT	0	0
X	0	0

(d) Implement the circuit including the output.



A certain device receives a bit on every clock cycle. If the bit received matches the previous bit, the machine outputs a 0, otherwise it outputs a 1. If there was no previous bit (i.e. the first bit ever received), the device outputs a 0.

(e) Draw the FSM as a Moore machine.



(f) Using an “almost” one-hot encoding (where one state is encoded as all 0’s), draw the state encoding table.

STATE	ENC
START	0000
00	0001
01	0010
10	0100
11	1000

(g) Write the minimized next-state and output logic.

$$Q_0 = \overline{Q_0} \overline{Q_1} \overline{Q_2} \overline{Q_3} \overline{X} + Q_0 \overline{X} + Q_2 \overline{X}$$

$$Q_1 = Q_0 X + Q_2 X$$

$$Q_2 = Q_1 \overline{X} + Q_3 \overline{X}$$

$$Q_3 = \overline{Q_0} \overline{Q_1} \overline{Q_2} \overline{Q_3} X + Q_1 X + Q_3 X$$

$$\text{OUTPUT} = Q_1 + Q_2$$

(h) Implement the circuit including the output.

PROBLEM REMOVED

7. (30 points)



To send a text message on a cell phone without a keyboard, some characters require a certain key to be pressed multiple times in quick succession. For example, pressing “2” once enters an “A,” pressing it twice in a row enters a “B,” three times is a “C,” four is a “2,” and then the cycle repeats (five times is an “A,” six times is a “B,” etc...).

For this problem, you will create an FSM for a similar device, but to greatly simplify things, we will assume there are only 2 buttons—B2 and B3—and these buttons only have 2 outputs each: B2 generates the characters “A” and “2”; B3 generates the characters “D” and “3.”

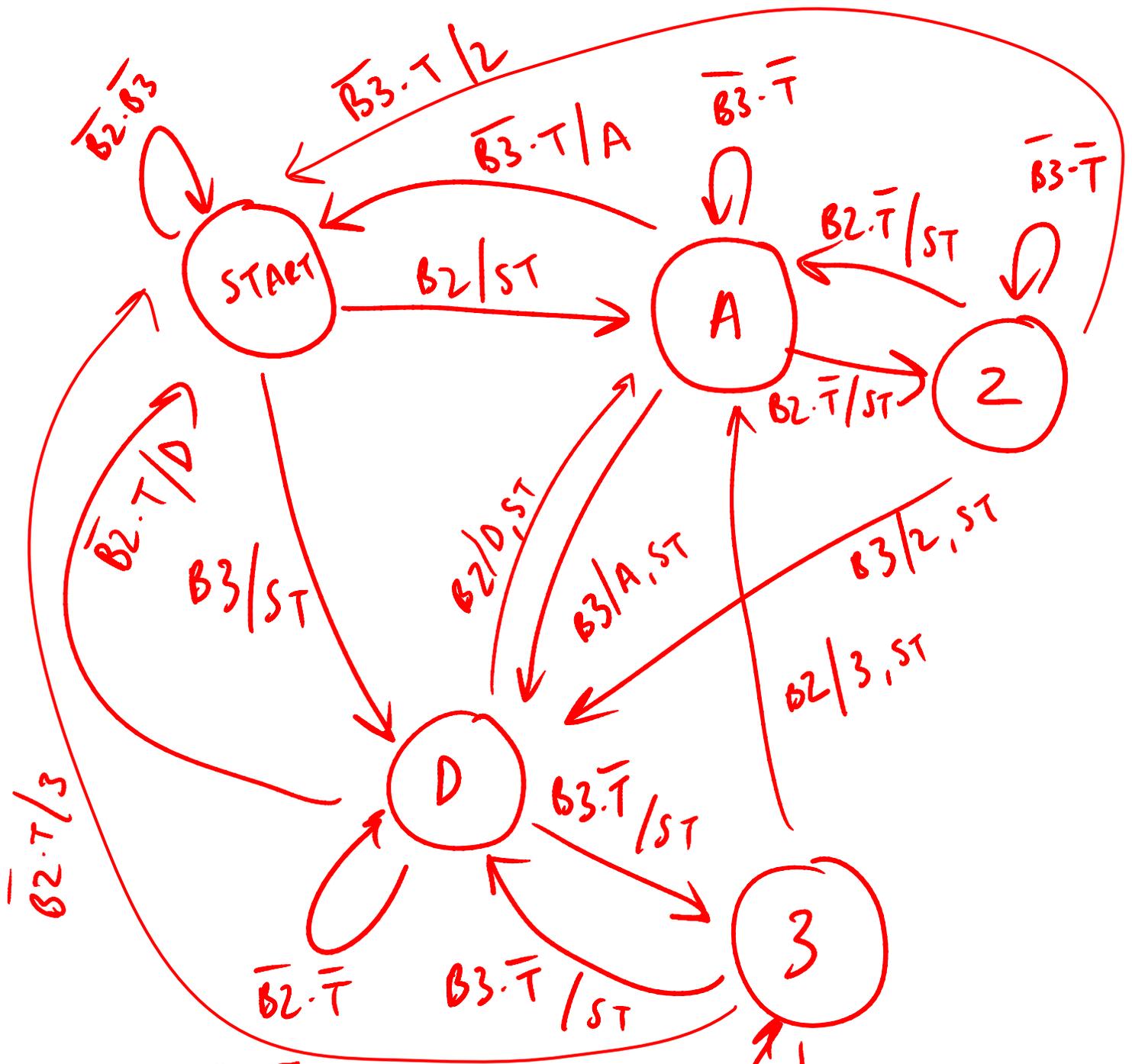
You will need a way to distinguish between single presses and multiple quick button presses. For example, a single press for “A” followed by a double press to enter “2” should not be misinterpreted as a triple press for “A” (from “A” to “2” to “A”). The former will output “A2”, while the latter will only output “A”.

An output is generated from a succession of key presses if a certain amount of time elapses or a different key is pressed. For example, pressing B2, B2, B3 in quick succession will output “2D”; pressing B2, B2, B3 in slow succession (i.e. pausing in between button presses) will output “AAD”.

Draw a Mealy machine that controls this device on the next page. Assume that B2 and B3 are never simultaneously active. Make sure every possible input transition is handled in every state.

DRAW YOUR ANSWER ON THE NEXT PAGE

Have a great summer!



B2	B3	T
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

[THIS SPACE INTENTIONALLY LEFT BLANK]