
Note: *Where appropriate in all of the questions below, you may assume that the system variables are available in both complemented and uncomplemented form.*

---

1. **[26 marks total.]**

   (a) **[4 marks.]** Consider the two circuits shown below in Fig. 1. Use algebraic transformations to prove or disprove that the two circuits given above implement the same function (do not use a truth table or Karnaugh map).

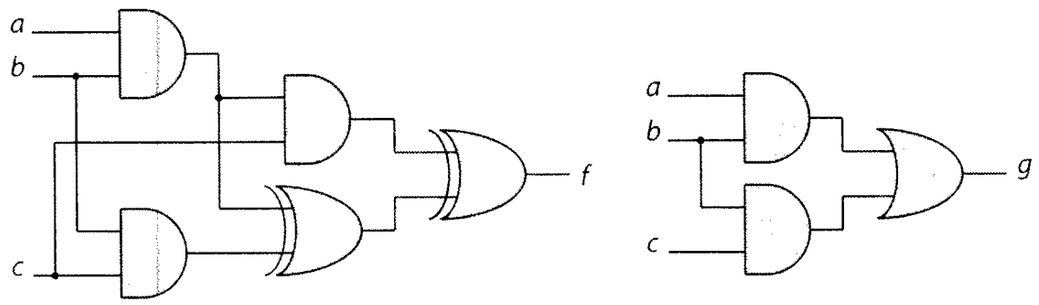


Fig. 1. Prove or disprove circuit equivalence

$f = abc \oplus (ab \oplus bc) =$

$= abc \oplus (a \oplus c)b =$

$= (\underbrace{ac \oplus a \oplus c}_{a \vee c})b = (a \vee c)b$

Since

$a \vee c = \overline{\bar{a}\bar{c}} = (a \oplus 1)(c \oplus 1) \oplus 1 =$

$= ac \oplus a \oplus c \oplus 1 \oplus 1 = ac \oplus a \oplus c$

The circuits are equivalent

$g = ab \vee bc =$

$= (a \vee c)b$

   (b) **[6 marks.]** Consider the function $f(a, b, c, d)$ implemented as follows:

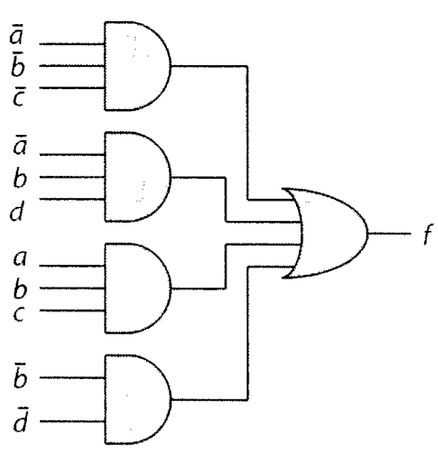


Fig. 2. Circuit for Question 1, parts (b)-(e)

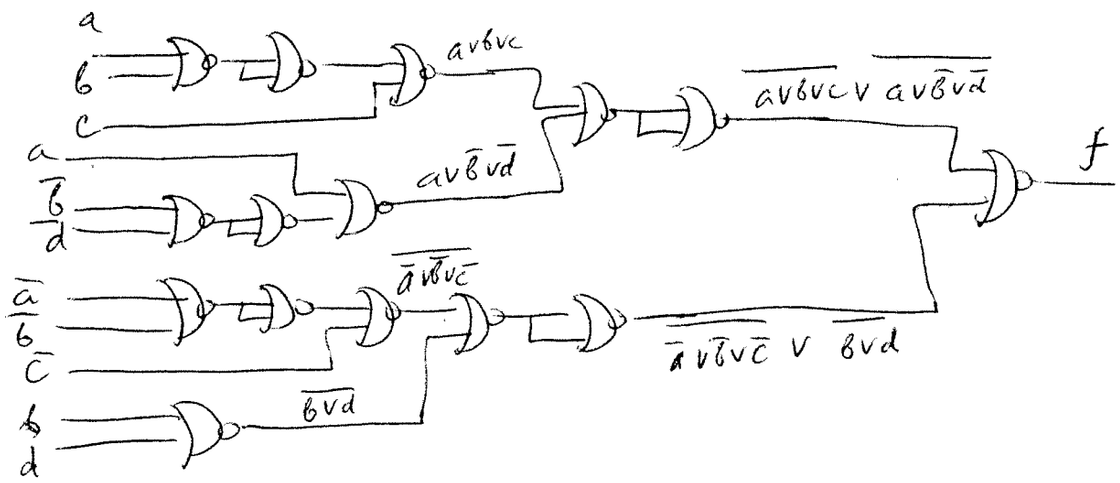
Part (b) continued ...

(Continued from p. 3.) Re-design the circuit in Fig. 2 using two-input NOR gates only (inverters are not available).

$$f = \bar{a}\bar{b}\bar{c} \vee \bar{a}bd \vee abc \vee \bar{b}\bar{d} = \overline{(a \vee b \vee c)} \vee \overline{(a \vee \bar{b} \vee d)} \vee$$

$$\vee \overline{(\bar{a} \vee \bar{b} \vee \bar{c})} \vee \overline{(b \vee d)} =$$

$$= \overline{\overline{a \vee b \vee c} \vee a \vee \bar{b} \vee d} \quad \overline{\bar{a} \vee b \vee \bar{c} \vee \bar{b} \vee d}$$



(c) **[6 marks.]** Re-design the circuit in Fig. 2 using two-input AND and two-input EXOR (XOR) gates only (inverters are not available).

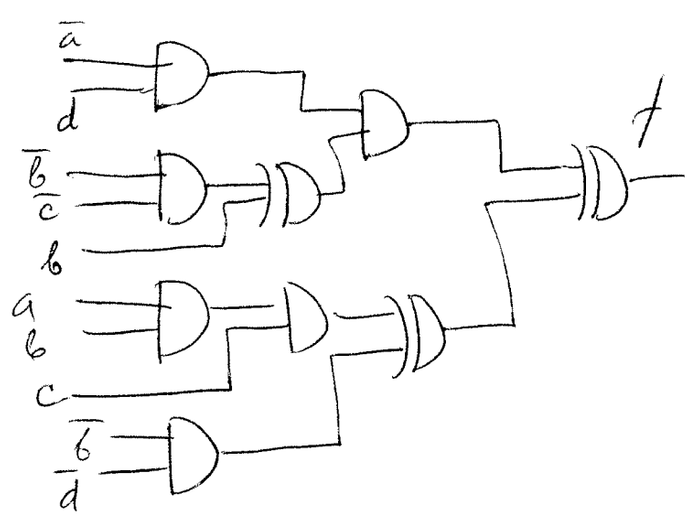
$$f = \bar{a}\bar{b}\bar{c} \vee \bar{a}bd \vee abc \vee \bar{b}\bar{d} = (\bar{a}\bar{b}\bar{c} \oplus \bar{a}bd \oplus \underbrace{\bar{a}\bar{b}\bar{c} \cdot \bar{a}bd}_{0}) \vee$$

$$\vee (abc \oplus \bar{b}\bar{d} \oplus \underbrace{abc\,\bar{b}\bar{d}}_{0}) = (\bar{a}\bar{b}\bar{c} \oplus \bar{a}bd) \vee (abc \oplus \bar{b}\bar{d}) =$$

$$= \bar{a}\bar{b}\bar{c} \oplus \bar{a}bd \oplus abc \oplus \bar{b}\bar{d} \oplus (\bar{a}\bar{b}\bar{c} \oplus \bar{a}bd)(abc \oplus \bar{b}\bar{d}) =$$

$$= \bar{a}\bar{b}\bar{c} \oplus \bar{a}bd \oplus abc \oplus \bar{b}\bar{d} \oplus \bar{a}\bar{b}\bar{c}\bar{d} =$$

$$= \bar{a}d(\bar{b}\bar{c} \oplus b) \oplus abc \oplus \bar{b}\bar{d}$$



(d) [**6 marks.**] Re-design the circuit in Fig. 2 using only 2-to-1 multiplexers. Use *at most seven* such multiplexers and no other logic gates. (You may use Shannon expansion in algebraic form or in truth-table form.)

$f = \bar{a}\bar{b}\bar{c} \vee \bar{a}bd \vee abc \vee \bar{b}\bar{d} =$

$= \bar{a} f_{a=0} \vee a f_{a=1} = \bar{a}(\underbrace{\bar{b}\bar{c} \vee bd \vee \bar{b}\bar{d}}_{f_{a=0}}) \vee a(\underbrace{bc \vee b\bar{d}}_{f_{a=1}})$

$f_{a=0} = \bar{b} f_{\substack{a=0\\b=0}} \vee b f_{\substack{a=0\\b=1}} = \bar{b}(\bar{c} \vee \bar{d}) \vee b(d)$

$f_{a=1} = \bar{b} f_{\substack{a=1\\b=0}} \vee b f_{\substack{a=1\\b=1}} = \bar{b}(\bar{d}) \vee b(c)$

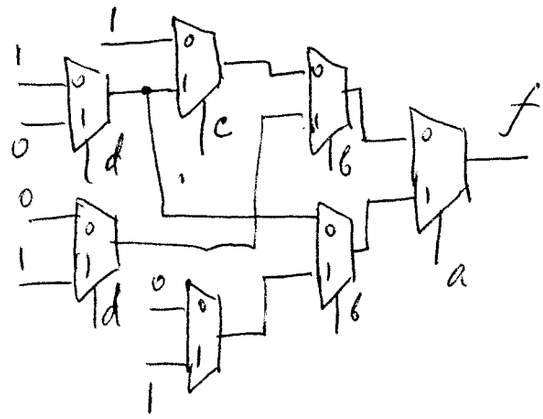
$f_{\substack{a=0\\b=0}} = \bar{c} f_{\substack{a=0\\b=0\\c=0}} \vee c f_{\substack{a=0\\b=0\\c=1}} = \bar{c}(1) \vee c(\bar{d})$

$f_{\substack{a=0\\b=1}} = \bar{d}(0) \vee d(1)$

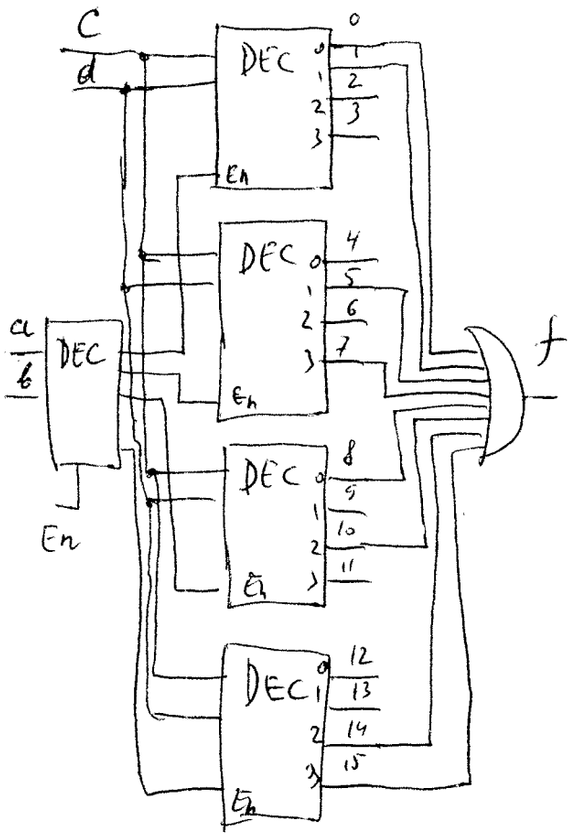
$f_{\substack{a=1\\b=0}} = \bar{d}(1) \vee d(0)$

$f_{\substack{a=1\\b=1}} = \bar{c}(0) \vee c(1)$

$f_{\substack{a=0\\b=0\\c=1}} = \bar{d}(1) \vee d(0)$



(e) [**4 marks.**] Re-design the circuit in Fig. 2 using only 2-to-4 decoders (with enable inputs). Use any additional logic gates that you may need (with any number of inputs).



we need to restore the standard (canonical) form for F:

$f = \bar{a}\bar{b}\bar{c} \vee \bar{a}bd \vee abc \vee \bar{b}\bar{d} =$

$= \bar{a}\bar{b}\bar{c}(\bar{d} \vee d) \vee \bar{a}b(\bar{c} \vee c)d \vee$

$\vee abc(\bar{d} \vee d) \vee (\bar{a} \vee a)\bar{b}(\bar{c} \vee c)\bar{d} =$

$= \bar{a}\bar{b}\bar{c}\bar{d} \vee \bar{a}\bar{b}\bar{c}d \vee \bar{a}b\bar{c}d \vee$

$\vee \bar{a}bcd \vee abc\bar{d} \vee abcd \vee \bar{a}\bar{b}\bar{c}\bar{d} \vee$

$\vee \bar{a}\bar{b}\bar{c}d \vee a\bar{b}\bar{c}\bar{d} \vee a\bar{b}\bar{c}d =$

$= \bar{a}\bar{b}\bar{c}\bar{d} \vee \bar{a}\bar{b}\bar{c}d \vee \bar{a}b\bar{c}d \vee \bar{a}bcd \vee$

$\vee abc\bar{d} \vee abcd \vee a\bar{b}\bar{c}\bar{d} \vee a\bar{b}\bar{c}d$

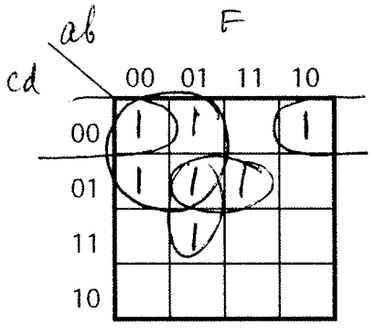
$= \Sigma(0,1,5,7,14,15,8,10) =$

$= \Sigma(0,1,5,7,8,10,14,15)$

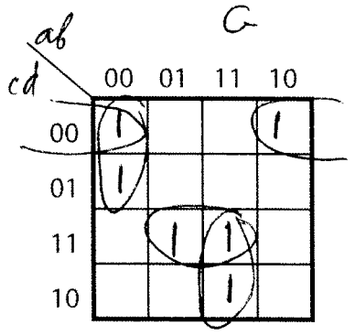
2. **[30 marks total]** Consider the two functions:

$$F(a,b,c,d) = \bar{a}\bar{b}\bar{c} \vee b\bar{c}d \vee \bar{b}\bar{c}\bar{d} \vee \bar{a}bd \vee \bar{a}b\bar{c}$$
$$G(a,b,c,d) = \bar{a}\bar{b}\bar{c} \vee abc \vee \bar{b}\bar{c}\bar{d} \vee bcd$$

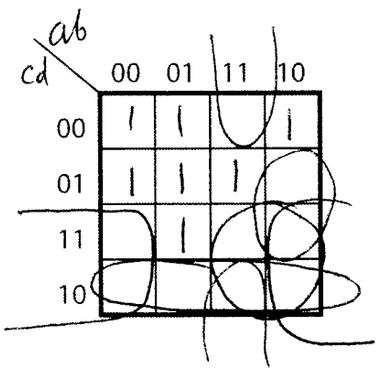
(a) **[10 marks]** Use the Karnaugh map templates below to determine minimal SOP and POS expressions for each function. If there are multiple solutions for any of these functions, give just one solution for each.



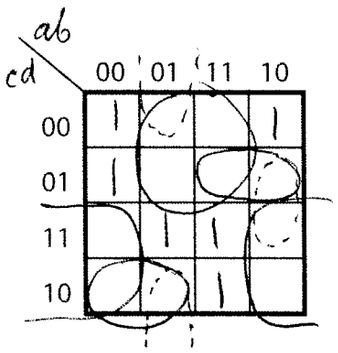
$F_{SOP} = \bar{a}\bar{c} \vee \bar{b}\bar{c}\bar{d} \vee b\bar{c}d \vee \bar{a}bd$



$G_{SOP} = \bar{a}\bar{b}\bar{c} \vee b\bar{c}\bar{d} \vee bcd \vee abc$



$F_{POS} = (\bar{a} \vee \bar{c})(b \vee \bar{c})(\bar{c} \vee d) \wedge (\bar{a} \vee \bar{b} \vee d)(\bar{a} \vee b \vee \bar{d})$



$G_{POS} = (\bar{b} \vee c)(b \vee \bar{c}) \wedge (a \vee \bar{c} \vee d)(\bar{a} \vee c \vee \bar{d})$

or

$G_{POS} = (\bar{b} \vee c)(b \vee \bar{c}) \wedge (a \vee \bar{b} \vee d)(\bar{a} \vee b \vee \bar{d})$

or

$G_{POS} = (\bar{b} \vee c)(b \vee \bar{c})(a \vee \bar{c} \vee d)(\bar{a} \vee b \vee \bar{d})$

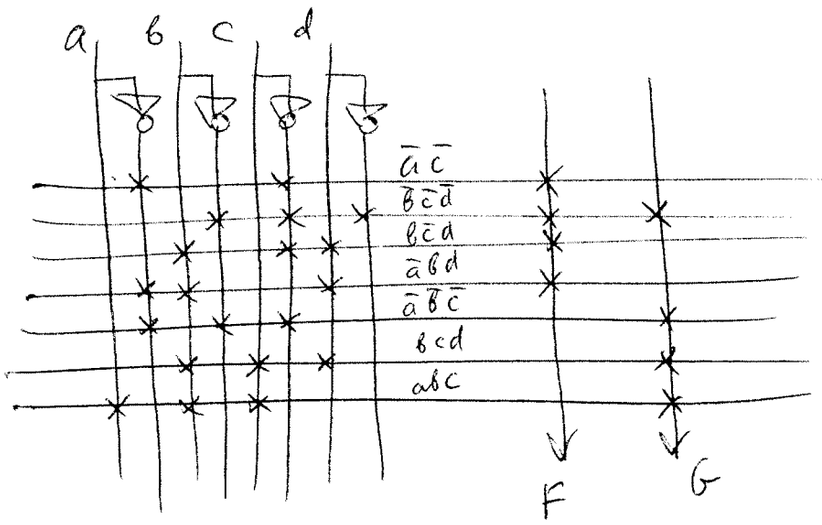
or

$G_{POS} = (\bar{b} \vee c)(b \vee \bar{c})(a \vee b \vee d)(\bar{a} \vee c \vee \bar{d})$

(b) [**4 marks.**] Implement both functions F and G using a PLA. For full marks, try to utilize shared product terms to best simplify the PLA implementation.

Let us use : $F = \bar{a}\bar{c} \vee \bar{b}\bar{c}\bar{d} \vee b\bar{c}d \vee \bar{a}bd$

$G = \bar{a}b\bar{c} \vee \bar{b}c\bar{d} \vee bcd \vee abc$



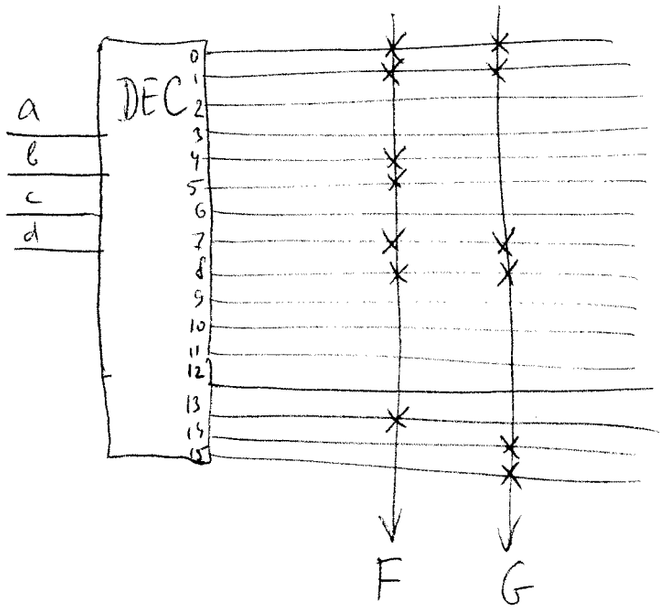
(c) [**4 marks.**] Implement both functions F and G using a ROM.

FOR ROM, we need a standard (canonical) form

FROM THE K-map,

$F_{SOP} = \sum m(0, 1, 4, 5, 7, 8, 13)$

$G_{SOP} = \sum m(0, 1, 7, 8, 14, 15)$



(d) [**4 marks.**] Derive the standard (canonical) form for the function F using algebraic manipulations. Convert your answer to shorthand SOP notation (i.e., $\sum m(...)$) and POS notation (i.e., $\prod M(...)$).

$F = \bar{a}\bar{b}\bar{c} \vee b\bar{c}d \vee b\bar{c}\bar{d} \vee \bar{a}bd \vee \bar{a}b\bar{c} =$

$= \bar{a}b\bar{c}(\bar{d} \vee d) \vee (\bar{a} \vee a)b\bar{c}d \vee (\bar{a} \vee a)\bar{b}\bar{c}\bar{d} \vee \bar{a}b(\bar{c} \vee c)d \vee$

$\vee \bar{a}b\bar{c}(\bar{d} \vee d) = \bar{a}bc\bar{d} \vee \bar{a}bcd \vee \bar{a}b\bar{c}d \vee abcd \vee \bar{a}\bar{b}\bar{c}\bar{d} \vee$

$\vee ab\bar{c}\bar{d} \vee \bar{a}b\bar{c}d \vee \bar{a}bcd \vee \bar{a}b\bar{c}\bar{d} \vee \bar{a}\bar{b}\bar{c}d =$

$= \bar{a}\bar{b}\bar{c}\bar{d} \vee \bar{a}\bar{b}\bar{c}d \vee \bar{a}b\bar{c}d \vee ab\bar{c}d \vee ab\bar{c}\bar{d} \vee \bar{a}bcd \vee \bar{a}b\bar{c}\bar{d}$

$F_{SOP} = \sum m(0,1,5,13,8,7,4) = \sum m(0,1,4,5,7,8,13)$

$F_{POS} = \prod M(2,3,6,9,10,11,12,14,15)$

(e) [**8 marks.**] Use the Quine-McCluskey algorithm to minimize the function F; use the standard SOP form you determined in part (d) as the starting point. Compare the results against the minimization of the SOP forms for F you performed in part (a) of this problem.

Implicant table

# of 1's	Minterms	Cubes		Cubes	
0	0000	0,1	000-	0,1,4,5	0-0- *
1	0001	0,4	0-00		
4	0100	0,8	-000 *		
8	1000	1,5	0-01		
5	0101	4,5	010-		
13	1101	5,13	-101 *		
7	0111	5,7	01-1 *		

Cubes * are prime implicants

Prime implicant chart

Prime Implicants		Minterms 0	1	4	8	5	13	7
0,1,4,5	0-0-	X	X	X		X		
0,8	-000	X			X			
5,13	-101					X	X	
5,7	01-1					X		X

Rows 0-0-, -000, -101, 01-1 are all have a single X covered, so they all are essential implicant

So $F_{SOP} = \bar{a}\bar{c} \vee b\bar{c}d \vee b\bar{c}d' \vee \bar{a}bd$ is minimal

3. [**20 marks total.**] Consider the circuit shown below in Fig. 3.

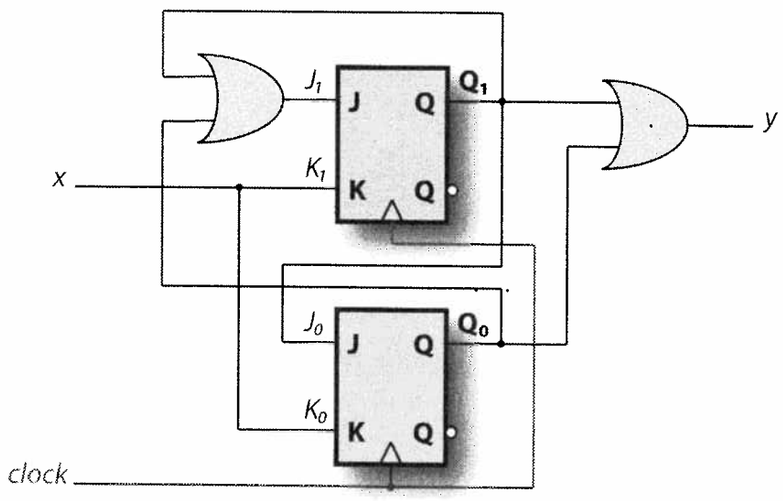


Fig. 3. Analyze and re-design this circuit to use TFFs.

(a) [**4 marks.**] Derive the excitation equations, transition equations, and the output equations.

$J_1 = Q_1 \vee Q_0$

$K_1 = X$

$J_0 = Q_1$

$K_0 = X$

$Q_1^+ = J_1\bar{Q}_1 \vee Q_1\bar{K}_1 = (Q_1 \vee Q_0)\bar{Q}_1 \vee Q_1\bar{X} = Q_0\bar{Q}_1 \vee Q_1\bar{X}$

$Q_0^+ = J_0\bar{Q}_0 \vee Q_0\bar{K}_0 = Q_1\bar{Q}_0 \vee Q_0\bar{X}$

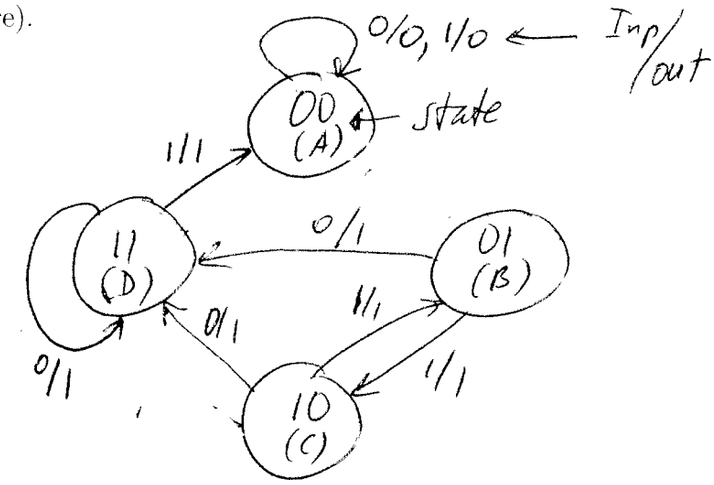
$Y = Q_1 \vee Q_0$

(b) [**4 marks.**] Create the state table that defines the current-states, input x, next-states, JKFF inputs (i.e., the excitation inputs), and the output y.

Q_1	Q_0	X	Q_1^+	Q_0^+	J_1	K_1	J_0	K_0	Y
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	1	0
0	1	0	1	1	1	0	0	0	1
0	1	1	1	0	1	1	0	1	1
1	0	0	1	1	1	0	1	0	1
1	0	1	0	1	1	1	1	1	1
1	1	0	1	1	1	0	1	0	1
1	1	1	0	0	1	1	1	1	1

(c) **[4 marks.]** Create the state diagram and specify what type of state machine is this (Mealy or Moore).

Let STATES be encoded as follows:
A = 00
B = 01
C = 10
D = 11



(d) **[8 marks.]** Re-design the circuit using T flip-flops. Draw the resulting circuit.

Q_1	Q_0	X	Q_1^+	Q_0^+	T_1	T_0	Y
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	1	1	1	0	1
0	1	1	1	0	1	1	1
1	0	0	1	1	0	1	1
1	0	1	0	1	1	1	1
1	1	0	1	1	0	0	1
1	1	1	0	0	1	1	1

We use $T_0 = Q_0 \oplus Q_0^+$ and $T_1 = Q_1 \oplus Q_1^+$ to fill in the T_1, T_0 columns in the table

Next, we minimize the excitation functions T_0, T_1:

T_1:

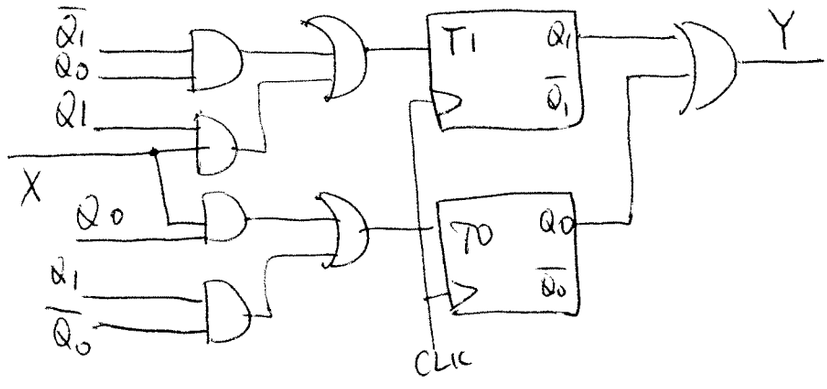
X \ Q_1Q_0	00	01	11	10
0	0	1	0	0
1	0	1	1	1

$T_1 = \bar{Q}_1 Q_0 \vee Q_1 X$

T_0:

X \ Q_1Q_0	00	01	11	10
0	0	0	0	1
1	0	1	1	1

$T_0 = Q_0 X \vee Q_1 \bar{Q}_0$

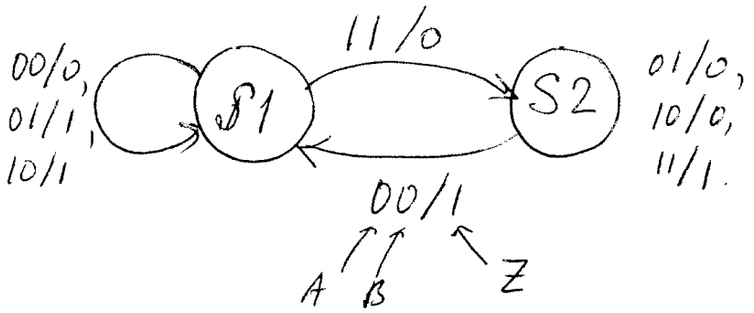


4. [**24 marks total.**] Design a sequential binary adder. Two binary sequences of an arbitrary length, corresponding to the two operands to be added, are applied to the inputs A and B, where the least-significant bit (LSB) of each sequence arrives first. The binary sum of the two numbers is produced as a corresponding output time sequence on the output Z, where the LSB is delivered first.

   (a) [**6 marks.**] Derive the state diagram for a Mealy-type system. Use whatever you determine in this step for parts (b) and (c). Note: You can do this with as few as two states!

   (If you are unable to come up with a suitable solution for this part, give a state diagram for any reasonable alternative Mealy-type system with two states and two inputs to complete parts (b) and (c).)

S1 = no carry was generated from the previous addition
S2 = a carry was generated from the previous addition



   (b) [**4 marks.**] Assign the necessary number of bits to represent the states of the system. Create the state table that defines the two input variables A, B, the current-states, next-states, and the output Z. Use don't-cares where appropriate.

Since there are two states, we need one bit, Q to represent the state. Let S1 ⇒ Q=0, S2 ⇒ Q=1

STATE TABLE

Curr. State	Inp.		Next state	OUTPUT
Q	A	B	Q^+	Z
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

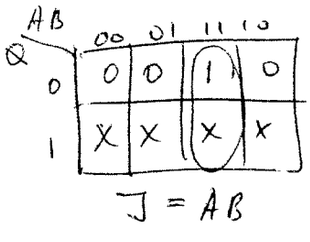
(c) **[6 marks.]** Derive the output equation and the excitation equations based on JK flip-flops. *(It is not necessary to draw the circuit).*

STATE TABLE

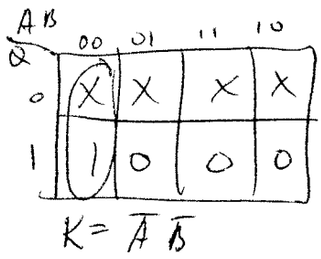
Q	A	B	Q^+	J	K	Y
0	0	0	0	0	X	0
0	0	1	0	0	X	1
0	1	0	0	0	X	1
0	1	1	1	1	X	0
1	0	0	0	X	1	1
1	0	1	1	X	0	0
1	1	0	1	X	0	0
1	1	1	1	X	0	1

Characteristic table is used to generate JK in the state table

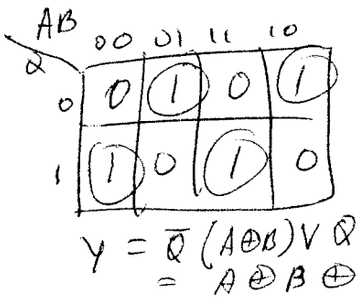
Q	Q^+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0



$J = A\bar{B}$



$K = \bar{A}\bar{B}$



$Y = \bar{Q}(A \oplus B) \vee Q(\overline{A \oplus B}) = A \oplus B \oplus Q$

(d) **[8 marks.]** Derive the state diagram for a Moore-type system. Note: You need at least four states! *(Be as messy as you like. Do not design the circuit.)*

S1 ⇒ the sum bit is 0 and no carry was generated from previous addition

S2 ⇒ the sum bit is 0 and a carry was generated

S3 ⇒ The sum bit is 1 and no carry was generated

S4 ⇒ The sum bit is 1 and a carry was generated