ECE124
Digital Circuits and Systems

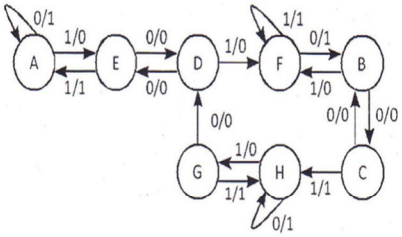
Prof. C. Gebotys

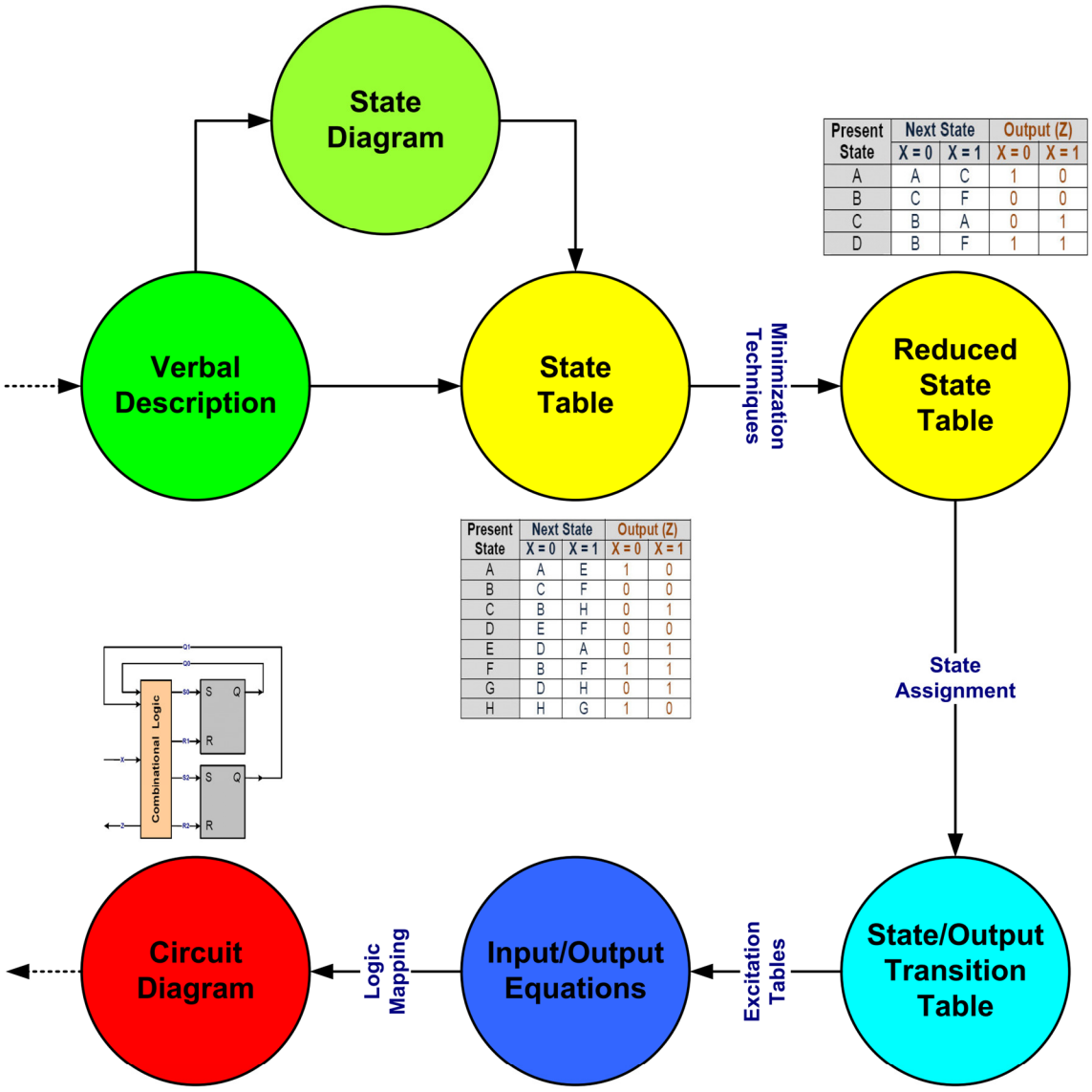
Final Exams Review

Spring 2011

Should you have any questions on this review, please contact Arash. [aTabibiazar@uwaterloo.ca]

Sequential Circuits Design Steps





Present State	Next State X = 0	Next State X = 1	Output (Z) X = 0	Output (Z) X = 1
A	A	C	1	0
B	C	F	0	0
C	B	A	0	1
D	B	F	1	1

Present State	Next State X = 0	Next State X = 1	Output (Z) X = 0	Output (Z) X = 1
A	A	E	1	0
B	C	F	0	0
C	B	H	0	1
D	E	F	0	0
E	D	A	0	1
F	B	F	1	1
G	D	H	0	1
H	H	G	1	0

Present State	Next State X = 0	Next State X = 1	Output (Z) X = 0	Output (Z) X = 1
10	10	01	1	0
00	01	11	0	0
01	00	10	0	1
11	00	11	1	1

	00	01	11	10
0	1	0	0	1
1	1	1	0	0

$Z = X\bar{Q}_0 + XQ_1$

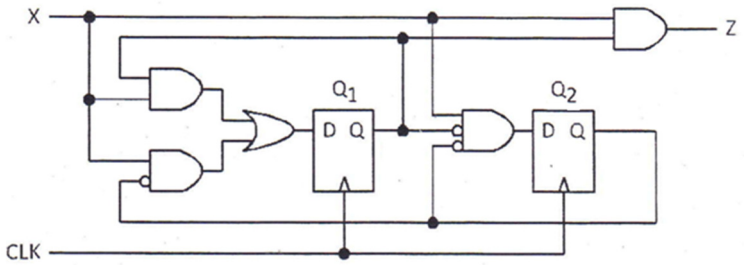
	00	01	11	10
0	1	0	0	0
1	1	0	1	1

$D_0 = \bar{Q}_1 Q_0 + XQ_1$

ECE124 Digital Circuits and Systems, Final Review, Spring 2011

[Q1] For the following clocked sequential circuit with one input (X) and one output (Z):

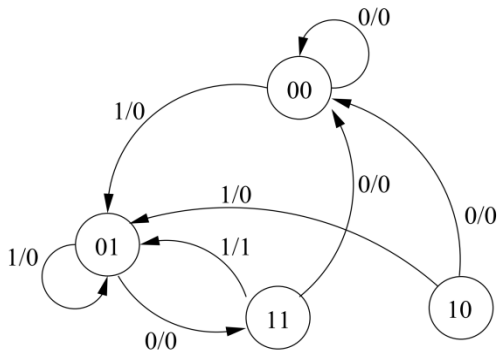
1. Drive a state table and draw a state diagram for the circuit.
2. Redesign this circuit by replacing the Q_1 flip-flop (i.e. the D flip-flop holding Q_1 state) with a JK flip-flop, and the Q_2 flip-flop with a T flip-flop. Only show the excitation equations (or state equations) for J_1, K_1, and T_2.



[Q2] Draw the state diagram for the table below that describes a finite-state machine which has one input x and one output z.

Present State	Next State		Output (z)	
	x = 0	x = 1	x = 0	x = 1
A	A	E	1	0
B	C	F	0	0
C	B	H	0	1
D	E	F	0	0
E	D	A	0	1
F	B	F	1	1
G	D	H	0	1
H	H	G	1	0

[Q3] Consider the following state diagram for a synchronous circuit with one input X and one output Z. Analyze this state diagram and draw its circuit implementation using JK flip-flop (state Q0) and T flip-flop (state Q1) and MUX-4x1 for Z.



[Q4] Draw a circuit diagram for non-overlapped '101' detector with "D" flip-flops as a Mealy and Moore machine.

[Q5] Given a 32x8 ROM chip with an enable input, show the block level required connections to construct a 128x8 ROM with above ROM chips and a decoder. How many data and address lines these ROMs have?

[Q6] Implement the circuit defined by equation $F(a,b,c,d) = \sum (0,5,6,7,11)$ using:

1. 4-to-1 multiplexers and logic gates.
2. 2-to-4 decoders with non-inverted outputs and logic gates.

[Q7] Use a 3-bit binary counter with active-high load (L) and Increment (I) control inputs (load has higher priority than increment) and implement a circuit (draw) to generate and repeat the following sequence at the output of the counter. Initial counter value is “000”.

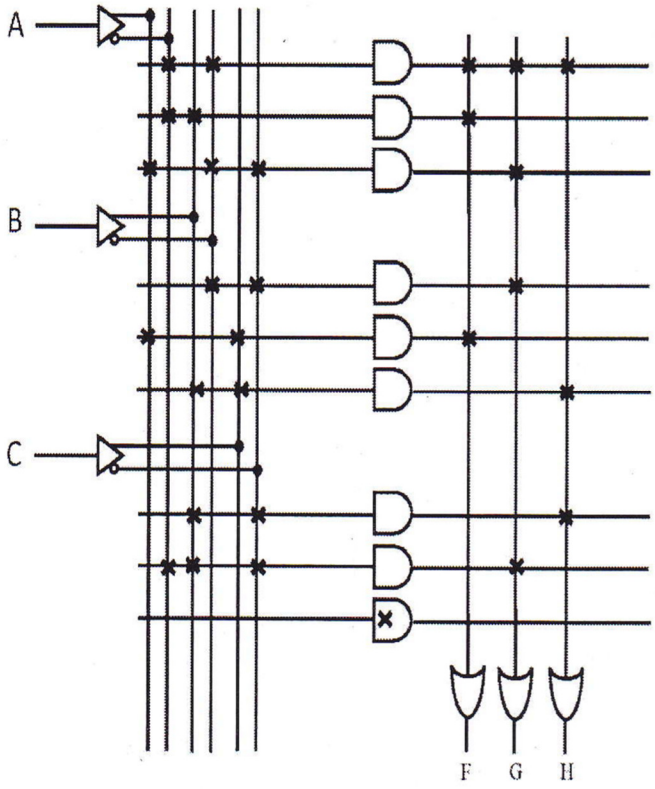
$$\cdots \rightarrow 000 \rightarrow 001 \rightarrow 010 \rightarrow 101 \rightarrow 110 \rightarrow 111 \rightarrow 000 \rightarrow \cdots$$

[Q8] Design a digital circuit that takes two 4-bit numbers A and B as input and generates output Z as follows:

- If A and B are odd numbers then Z=A-B
- If A and B are even numbers then Z=B-A
- If A is an even number and B is an odd number then Z=A+B
- If A is an odd number and B is an even number then Z=A-B-1

Assume that you have access to as many as you need of AND, OR, INV, XOR gates and only one FULL-ADDER, DECODER and MULTIPLEXER of any size.

[Q9] For the following Programmable Logic Array (PLA), find the function expressions for all outputs and draw the Karnaugh-Map for function "F".



[Q10] What are three different ways of representing a signed number? Assume 7 bit numbers and represent (-15) in each of them, then find (B-A) and (A-C) for A = 1101010, B = 0110101 and C = 0010101 in all forms.

[Q11] Find:

a) The 7’s complement of base-8 number “45201”
b) Multiplication of base-12 numbers “541” and “3”
c) Base-10 unsigned number “214.45” to its base-2 representation
d) Base-6 number “513” to its base-10 and then base-5 representations
e) Hexadecimal number “AF6” to its base-2 and base-8 representations

[Q12] For the following asynchronous sequential state table, find all possible critical/non-critical races and cycles for states "**a**" and "**c**".

	Present State	Next State			
		00	01	11	10
a	**00**	**00**	11	01	11
b	**01**	11	**01**	**01**	10
c	**11**	10	**11**	01	10
d	**10**	11	**10**	01	**10**

[Q13] A crypto module which transforms a secret key bit stream, K, into two other bit streams, X and Y has to be designed. This module must be designed as an asynchronous sequential Mealy state machine. It works as follows: if the secret key bit stream contains '011', then it's replaced with '10(-1)'. This transformation reduces the number of '1' bits in the key (which has significant impact on subsequent processing times in elliptic curve algorithms). However since we cannot represent 0, 1 and -1 with a one bit output, we use two output signals, X and Y. Whenever a m-bit sequence of 1's is detected (where m>1), X is set to 1 for the first K='1' in the sequence and when a K='0' is detected after the m^{th} '1' bit both X and Y are set to '1'. Construct a sequential state table for this module.

```
K:0001000110111100011010 00
X:0001000101100010010110 00
Y:0000000001000010000100 00
```

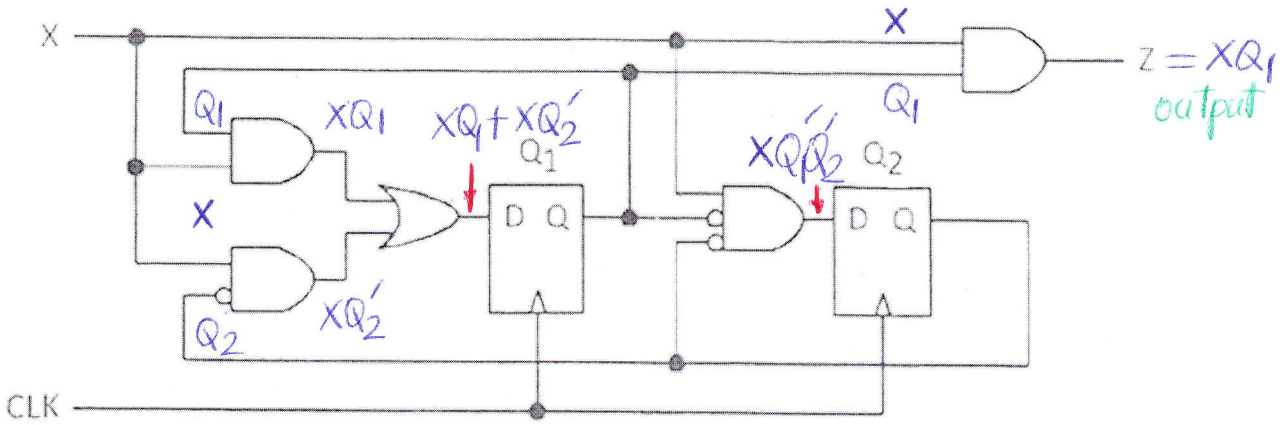
[Q14] For below sequential state table, perform a race-free state assignment and complete the entire state table:

Present State	Next State				Output			
	00	01	11	10	00	01	11	10
a	**a**	**a**	b	d	0	1	-	-
b	a	**b**	**b**	c	-	0	1	-
c	a	-	d	**c**	-	-	-	0
d	a	a	**d**	**d**	-	-	1	0

ECE124 Digital Circuits and Systems, Final Review, Spring 2011

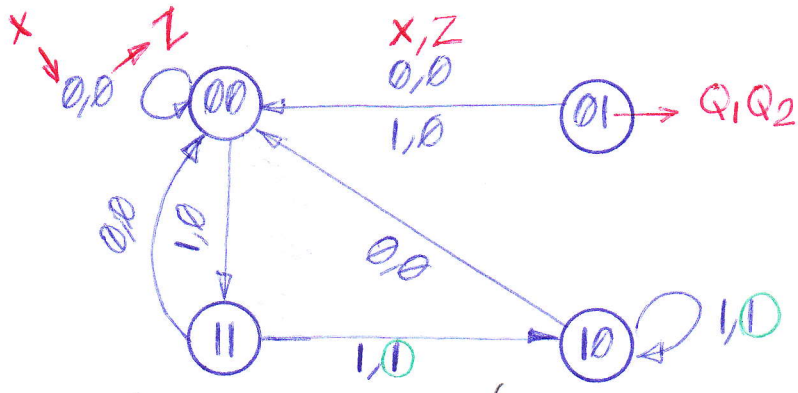
[Q1] For the following clocked sequential circuit with one input (X) and one output (Z):

1. Drive a state table and draw a state diagram for the circuit.
2. Redesign this circuit by replacing the Q_1 flip-flop (i.e. the D flip-flop holding Q_1 state) with a JK flip-flop, and the Q_2 flip-flop with a T flip-flop. Only show the excitation equations (or state equations) for J_1, K_1, and T_2.

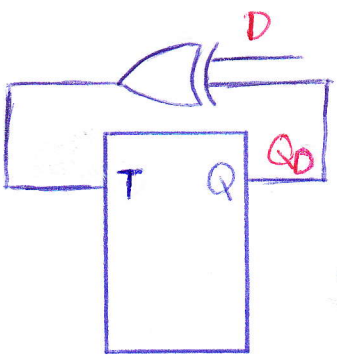


$Q_1^+ = XQ_1 + XQ_2'$ $\quad Q_2^+ = XQ_1'Q_2'$ ← states

PS Q_1Q_2	IN X	NS $Q_1^+Q_2^+$	OUT Z
00	0	00	0
00	1	11	0
01	0	00	0
01	1	00	0
10	0	00	0
10	1	10	1
11	0	00	0
11	1	10	1

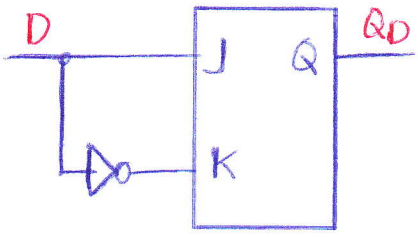


$$\begin{cases} J_1 = K_1' = D_1 = XQ_1 + XQ_2' \\ T_2 = D_2 \oplus Q_2 = XQ_1'Q_2' \oplus Q_2 \end{cases}$$



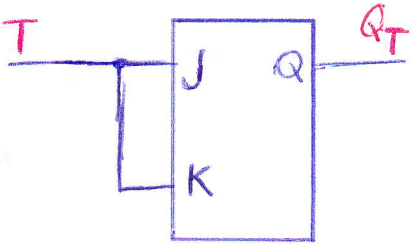
$Q^+ = D$

D-FF



$Q^+ = JQ' + K'Q$

JK-FF

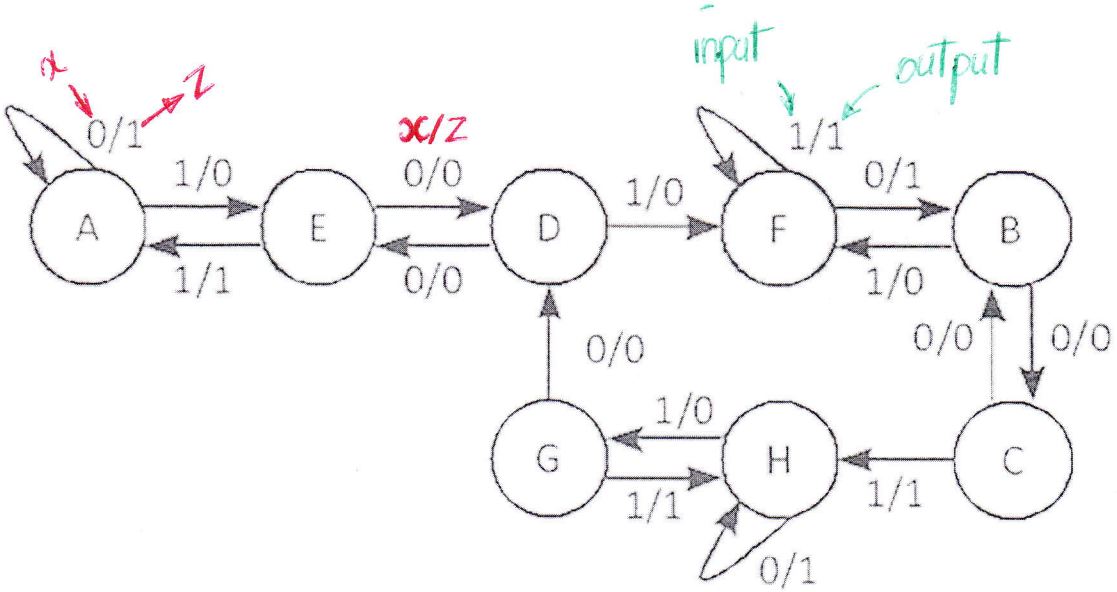


$Q^+ = T \oplus Q$

T-FF

[Q2] Draw the state diagram for the table below that describes a finite-state machine which has one input x and one output z.

Present State	Next State		Output (z)	
	x = 0	x = 1	x = 0	x = 1
A	A	E	1	0
B	C	F	0	0
C	B	H	0	1
D	E	F	0	0
E	D	A	0	1
F	B	F	1	1
G	D	H	0	1
H	H	G	1	0

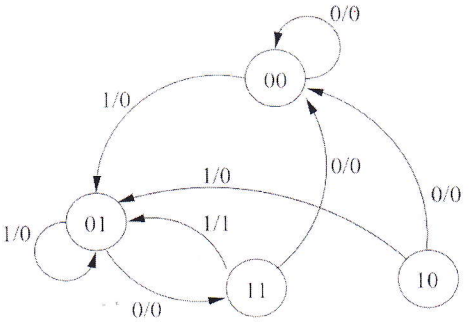


Mealy Diagram

$z = f(cs, x)$

ECE124 Digital Circuits and Systems, Final Review, Spring 2011

[Q3] Consider the following state diagram for a circuit with one input X and one output Z. Analyze this state diagram and draw its circuit implementation using JK flip-flop (state Q0) and T flip-flop (state Q1) and MUX-4x1 for Z.



D	T	Q_D^+	Q_T^+
0	0	0	Q_T
1	1	1	Q_T'

D&T FF characteristic tables

JK	SR	Q_{JK}^+	Q_{SR}^+
00	00	Q_{JK}	Q_{SR}
01	01	0	0
10	10	1	1
11	11	Q_{JK}'	—

JK&SR FF characteristic tables

PS q_1q_0	IN X	NS q_1q_0	OUT Z	JK-FF J_0	JK-FF K_0	T-FF T_1
00	0	00	0	0	X	0
00	1	01	0	1	X	0
01	0	11	0	X	0	1
01	1	01	0	X	0	0
10	0	00	0	0	X	1
10	1	01	0	1	X	1
11	0	00	0	X	1	1
11	1	01	1	X	0	1

	00	01	11	10
0	0	X	X	0
1	1	X	X	1

J0 = X

	00	01	11	10
0	X	0	1	X
1	X	0	0	X

K0 = q_1X'

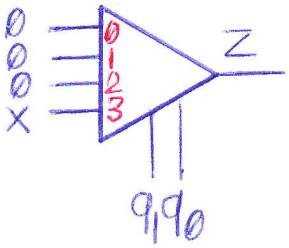
	00	01	11	10
0	0	1	1	1
1	0	0	1	1

T1 = $q_1 + q_0X'$

Flip-Flops excitation tables

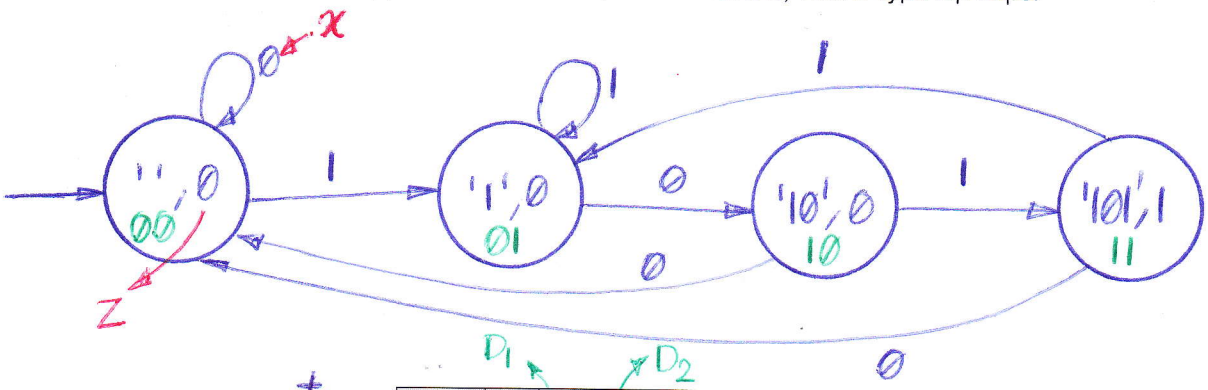
	00	01	11	10
0	0	0	0	0
1	0	0	1	0
F(x)	0	0	x	0

Z = X



Q	Q^+	JK	SR	D	T
0	0	0X	0X	0	0
0	1	1X	10	1	1
1	0	X1	01	0	1
1	1	X0	X0	1	0

[Q4] Draw a logic diagram for non-overlapped '101' detector (Moore machine) with D-type flip-flops.



Moore: $OUT = f(NS)$ ($NS \leftarrow CS^+$)

Mealy: $OUT = f(CS, IN)$

PS q_1q_2	IN x	NS q_1q_2	OUT z
00	0	00	0
00	1	01	0
01	0	10	0
01	1	01	0
10	0	00	0
10	1	11	1
11	0	00	0
11	1	01	0

D_1 D_2

$z = \begin{cases} q_1q_2 & \text{Moore ①} \\ q_1q_2'x & \text{Mealy ②} \end{cases}$

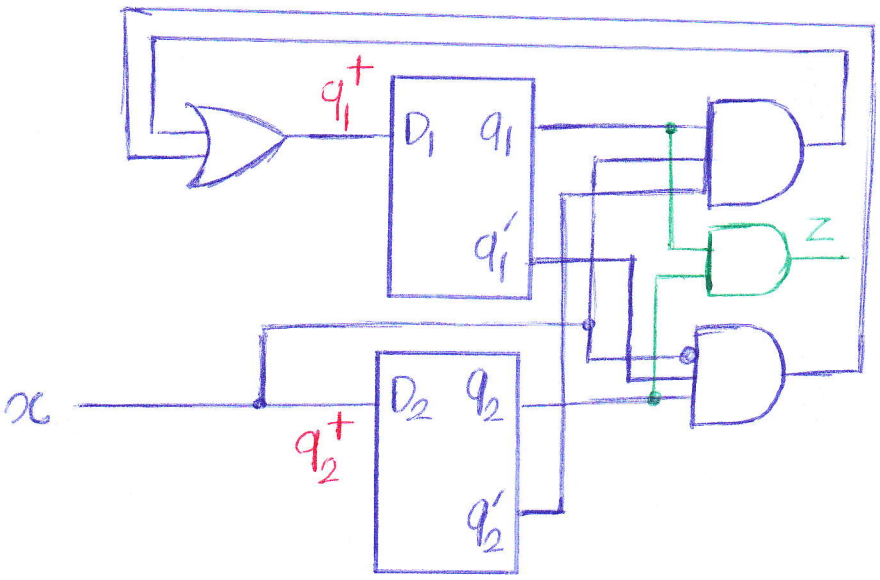
q_1

x \ q_1q_2	00	01	11	10
0	0	(1)	0	0
1	0	0	0	(1)

q_2

x \ q_1q_2	00	01	11	10
0	0	0	0	0
1	1	1	1	1

$D_1 = q_1^+ = x\bar{q}_1\bar{q}_2 + xq_1q_2'$ $\quad D_2 = q_2^+ = x$



example (overlap)

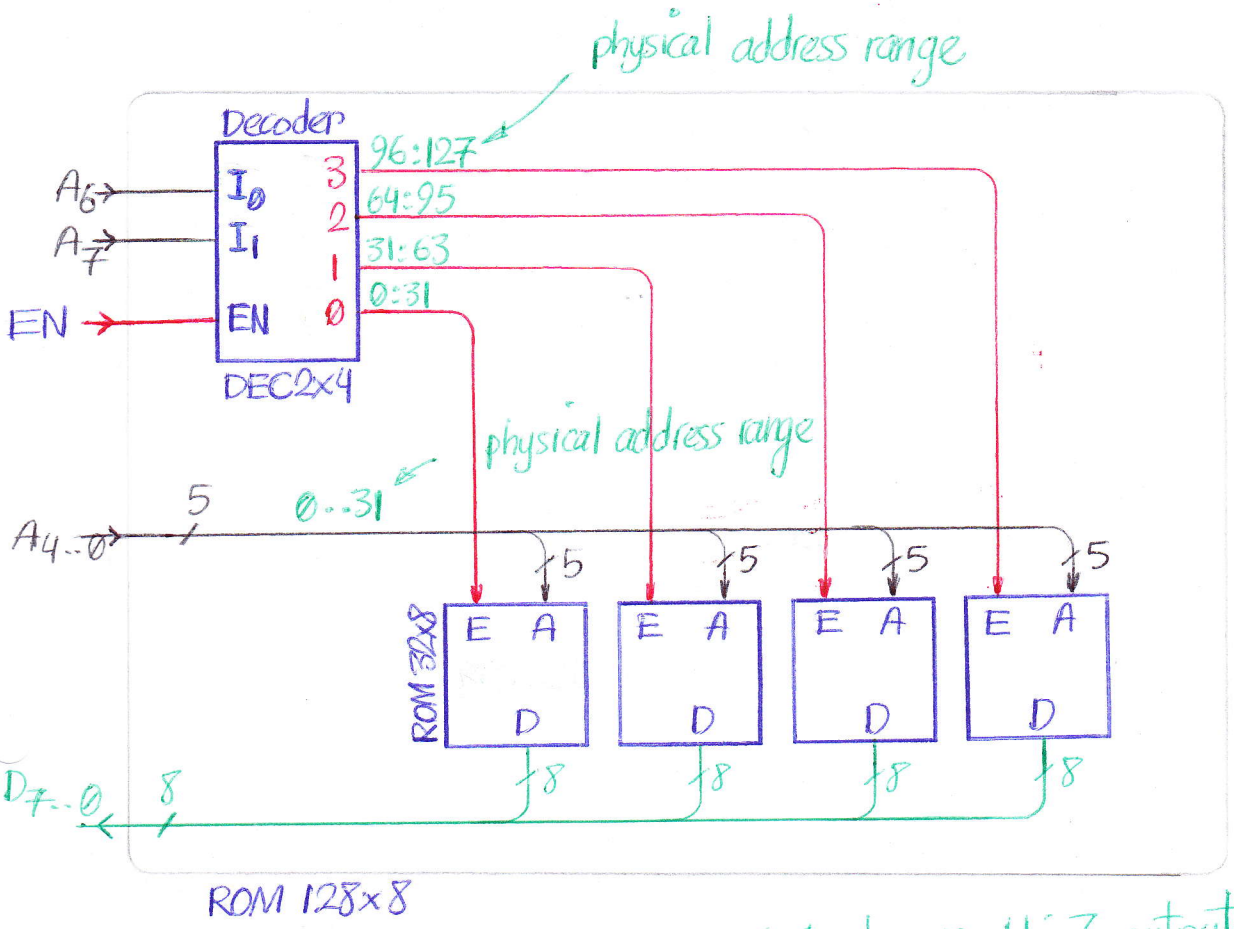
$x = 00100101011010$

$z_1 = 00000000100001000$

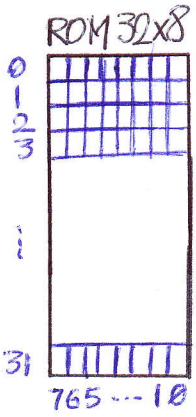
$z_2 = 00000001000010000$

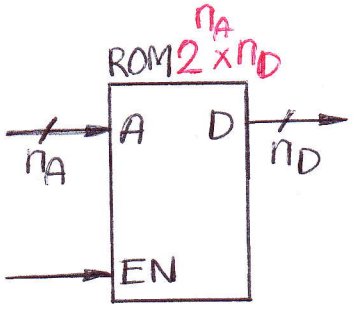
$t_0 t_1 t_2$

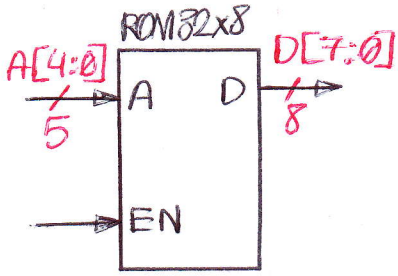
[Q5] Given a 32x8 ROM chip with an enable input, show the block level required connections to construct a 128x8 ROM with ROM chips and a decoder. How many data and address lines these ROMs have?



ROMs have Hi-Z outputs







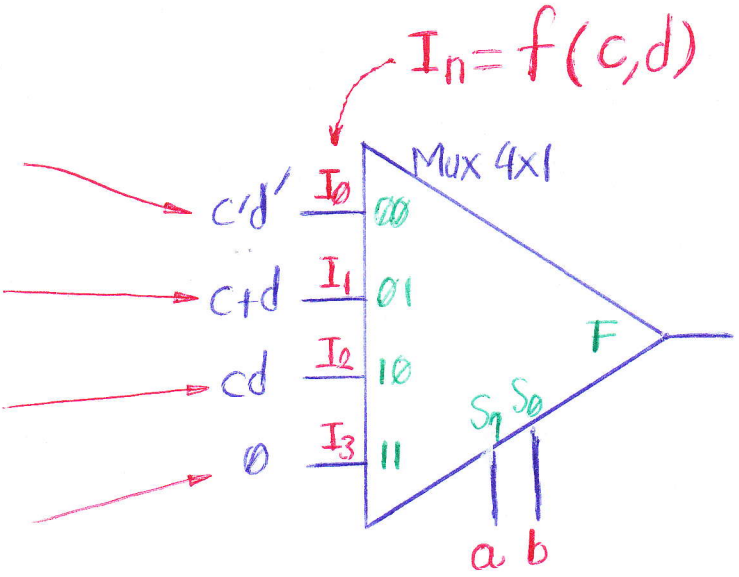
ROM-Size = 2^(#of address bits) x (# of data bits)

$= 2^{n_A} \times n_D$ bits

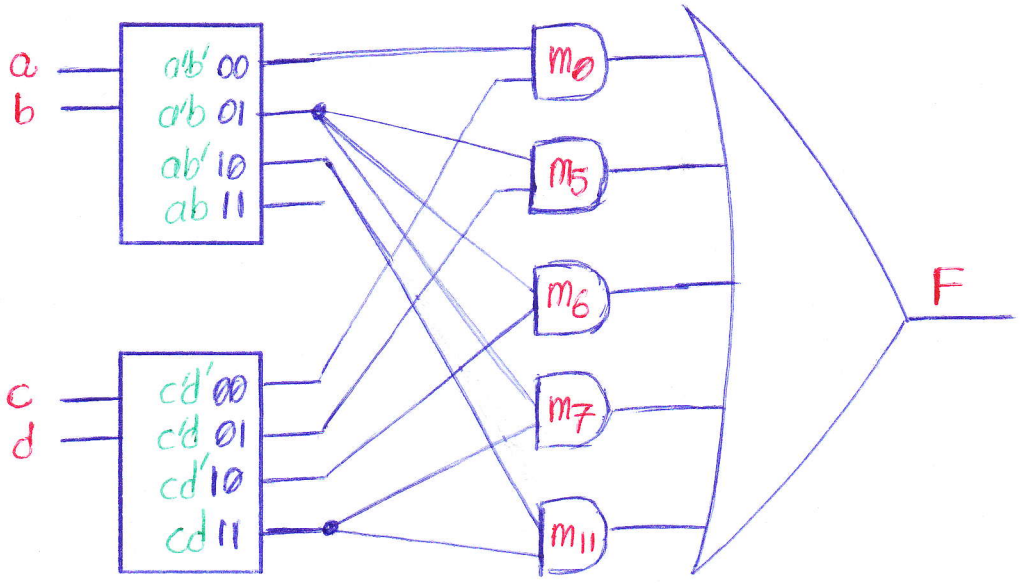
[Q6] Implement the circuit defined by equation F(a,b,c,d) = ∑ (0,5,6,7,11) using:

1. 4-to-1 multiplexers and logic gates.
2. 2-to-4 decoders with non-inverted outputs and logic gates.

m	a	b	c	d	F
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0



$$F(a,b,c,d) = \underset{m_0}{a'b'c'd'} + \underset{m_5}{a'bc'd} + \underset{m_6}{a'bcd'} + \underset{m_7}{a'bcd} + \underset{m_{11}}{ab'cd}$$



[Q7] Use a 3-bit binary counter with active-high load (L) and Increment (I) control inputs (load has higher priority than increment) and implement a circuit (draw) to generate and repeat the following sequence at the output of the counter. Initial counter value is "000".

$q_2^+ q_1^+ q_0^+$

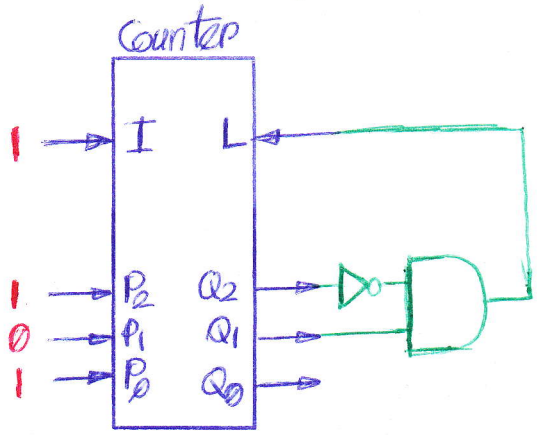
$\cdots \rightarrow 000 \rightarrow 001 \rightarrow 010 \rightarrow 101 \rightarrow 110 \rightarrow 111 \rightarrow 000 \rightarrow \cdots$

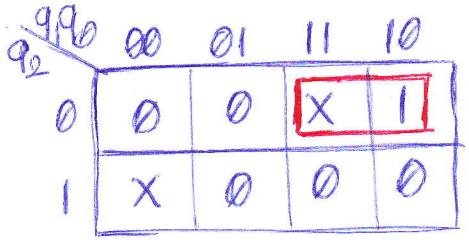
$q_2 q_1 q_0$

characteristic table →

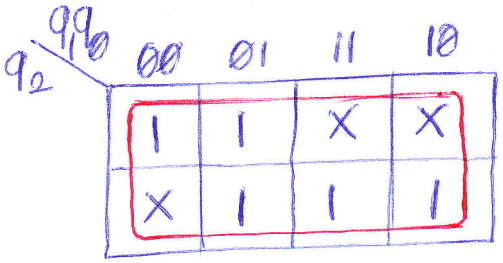
L	I	$Q^+[2:0]$
0	0	Q
0	1	Q+1
1	X	P

PS $q_2 q_1 q_0$	NS $q_2^+ q_1^+ q_0^+$	L	I	$P_2 P_1 P_0$
000	001	0	1	XXX
001	010	0	1	XXX
010	101	1	X	101
101	110	0	1	XXX
110	111	0	1	XXX
111	000	0	1	XXX
unused states { 011	XXX	X	X	XXX
100	XXX	X	X	XXX





$L = q_2' q_1$



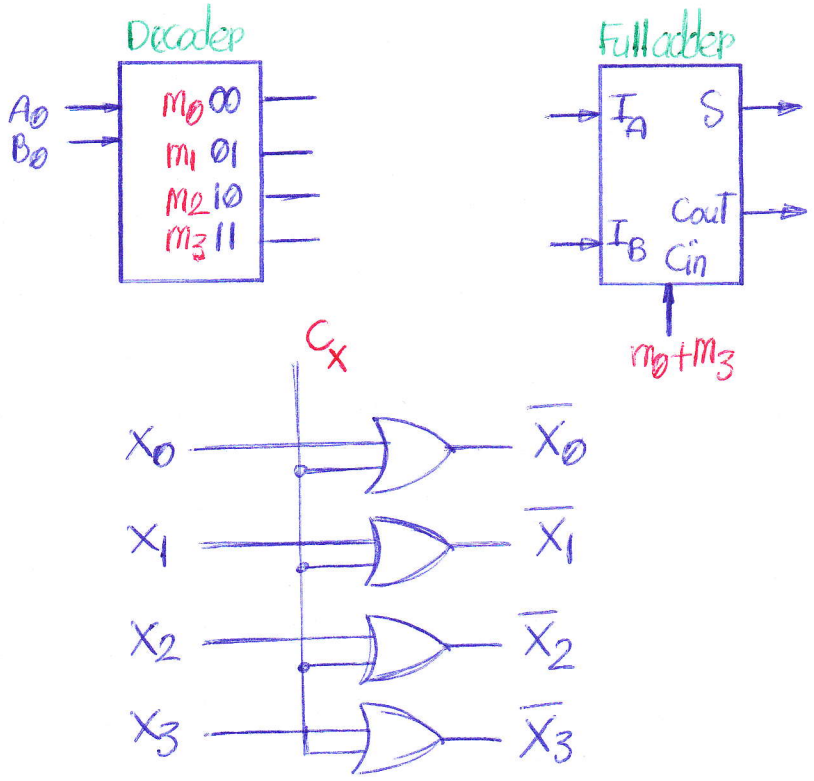
$I = 1$

[Q8] Design a digital circuit that takes two 4-bit numbers A and B as input and generates output Z as follows:

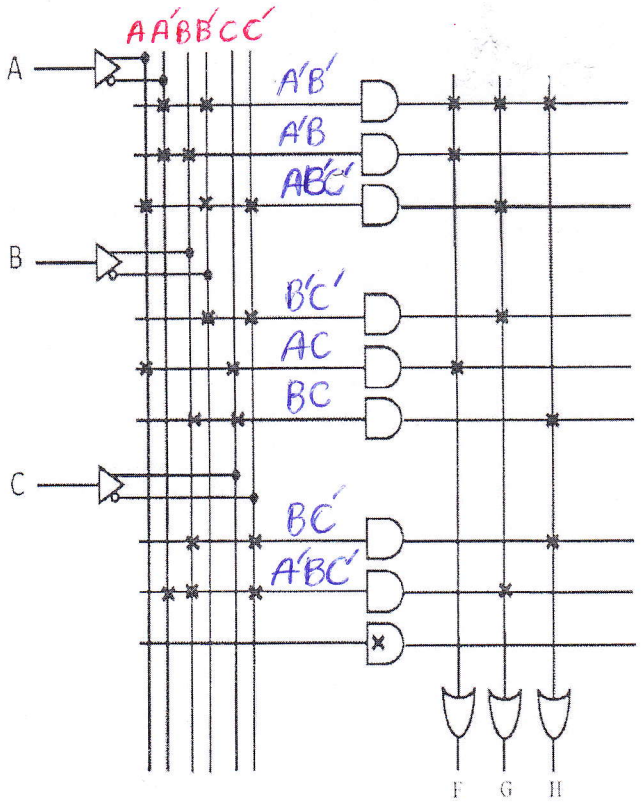
- If A and B are odd numbers then Z=A-B
- If A and B are even numbers then Z=B-A
- If A is an even number and B is an odd number then Z=A+B
- If A is an odd number and B is an even number then Z=A-B-1

Assume that you have access to as many as you need of AND, OR, INV, XOR gates and FULL-ADDER, DECODER and MULTIPLEXER of any size.

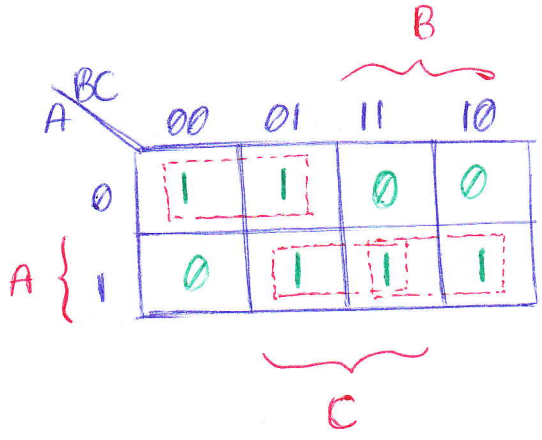
odd even A_0B_0		function Z	Full adder inputs I_A I_B C_{in}	Control Commands
m_0	00	B-A	~A B 1	$C_{in} = m_0 + m_3$
m_1	01	A+B	A B 0	$C_A = m_0$
m_2	10	A-B-1	A ~B 0	$C_B = m_2 + m_3$ } Complement circuit input
m_3	11	A-B	A ~B 1	



[Q9] For the following Programmable Logic Array (PLA), find the function expressions for all outputs and draw the Karnaugh-Maps for function "F".



$$\begin{cases} F = A'B' + A'B + AC \\ G = A'B' + AB'C' + B'C' + A'BC' \\ H = A'B' + BC + BC' \end{cases}$$



[Q10] What are three different ways of representing a signed number? Assume 7 bit numbers and represent (-15) in each of them, then find (B-A) and (A-C) for A = 1101010, B = 0110101 and C = 0010101 in all forms.

	−15
Signed & Magnitude	1001111
Signed & 1's Complement	1110000
Signed & 2's Complement	1110001

	Magnitude	1's Complement	2's Complement
B+	0110101	0110101	0110101
−A	0101010	0010101	0010110
	Sign (1)0011111	1001010	1001011
		overflow	overflow

A+	1101010	1101010	1101010
−C	1010101	1101010	1101011
	(1)(0)111110	(1)1010100	(1)1010101
	carry, sign	add end around carry	discard end carry
		1	
		10101101	

[Q11] Find:

a) The 7's complement of base-8 number "45201"
b) Multiplication of base-12 numbers "541" and "3"
c) Base-10 unsigned number "214.45" to its base-2 representation
d) Base-6 number "513" to its base-10 and then base-5 representations
e) Hexadecimal number "AF6" to its base-2 and base-8 representations

a) $-\begin{array}{r}77777\\45201\end{array}$
$\overline{32576}$

b) $\begin{array}{r}{\scriptstyle 1}\\541\\\times\ 3\end{array}$
$\overline{1403}$

c) 214.45

$214 \div 2 = 107 + 0$ (lsb)
$107 \div 2 = 53 + 1$
$53 \div 2 = 26 + 1$
$26 \div 2 = 13 + 0$
$13 \div 2 = 6 + 1$
$6 \div 2 = 3 + 0$
$3 \div 2 = 1 + 1$
$1 \div 2 = 0 + 1$ (msb)

$0.45 \times 2 = 0.9 + 0$ (msb)
$0.9 \times 2 = 0.8 + 1$
$0.8 \times 2 = 0.6 + 1$
$0.6 \times 2 = 0.2 + 1$
$0.2 \times 2 = 0.4 + 0$
$0.4 \times 2 = 0.8 + 0$ (repeats from $\times 2$)

$(214.45)_{10} = (11010110.0\overline{111}00)_2$

d) $(513)_6 = 3\times6^0 + 1\times6^1 + 5\times6^2$
$= 189$
$189 \div 5 = 37 + 4$ (lsb)
$37 \div 5 = 7 + 2$
$7 \div 5 = 1 + 2$
$1 \div 5 = 0 + 1$ (msb)
$= (1224)_5$

e) AF6 = 1010 1111 0110
5 3 6 6
$= (5366)_8$

[Q12] For the following asynchronous sequential state table, find all possible critical/non-critical races and cycles for states "**a**" and "**c**".

	Present State	Next State			
		00	01	11	10
a	00	**00**	11	01	11
b	01	11	**01**	**01**	10
c	11	10	**11**	01	10
d	10	11	**10**	01	**10**

Fundamental Mode
- input changes when circuit is stable
- only one input change at a time

transient state

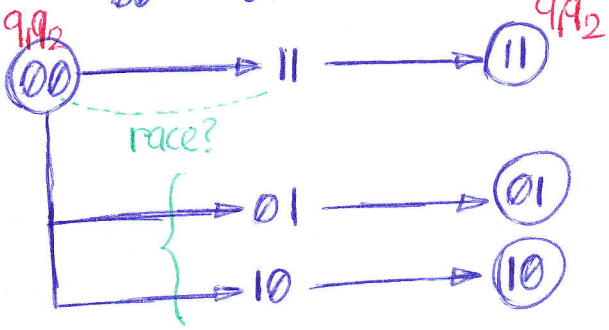
stable state

x_1x_2 $x_1^+x_2$

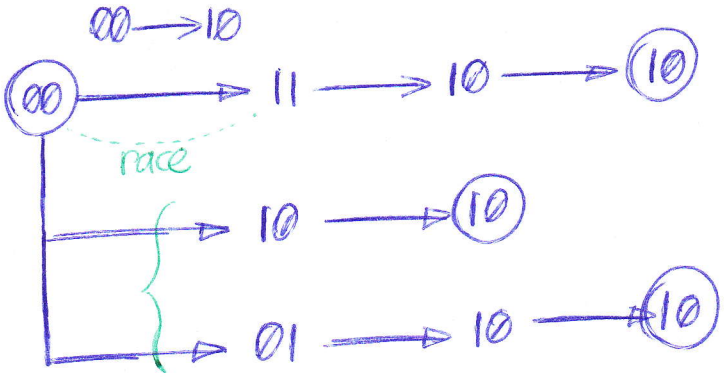
00 → 01

q_1q_2 $q_1^+q_2$

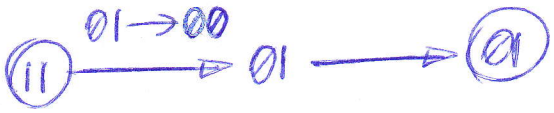
a_1



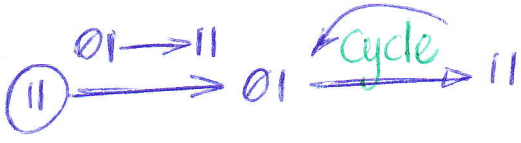
a_2



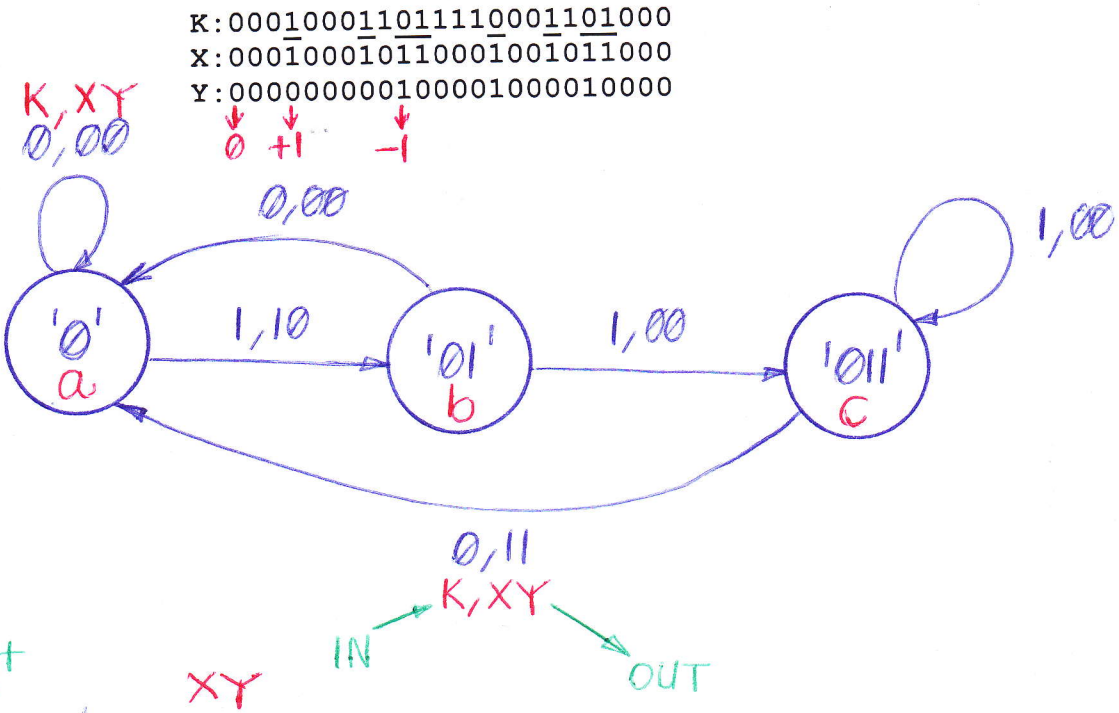
c_1



c_2



[Q13] A crypto module which transforms a secret key bit stream, K, into two other bit streams, X and Y has to be designed. This module must be designed as a synchronous sequential Mealy state machine. It works as follows: if the secret key bit stream contains '011', then it's replaced with '10(-1)'. This transformation reduces the number of '1' bits in the key (which has significant impact on subsequent processing times in elliptic curve algorithms). However since we cannot represent 0, 1 and -1 with a one bit output, we use two output signals, X and Y. Whenever a m-bit sequence of 1's is detected (where m>1), X is set to 1 for the first K='1' in the sequence and when a K='0' is detected after the m^{th} '1' bit both X and Y are set to '1'. Construct a flow table for this module.

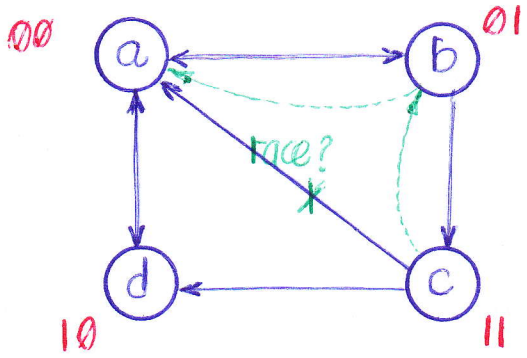


K →	Q^+		XY	
Q	0	1	0	1
a	(a)	b	00	10
b	a	c	00	00
c	(c)	a	00	11

[Q14] For below Mealy flow table, perform a race-free state assignment and complete the entire state table:

Present State	Next State				Output			
	00	01	11	10	00	01	11	10
a	a1	a2	b	d	0	1	1	0
b	a	b3	b4	c	0	0	1	0/1
c	b~~a~~	-	d	c5	0	-	0/1	0
d	a	a	d6	d7	0	1	1	0

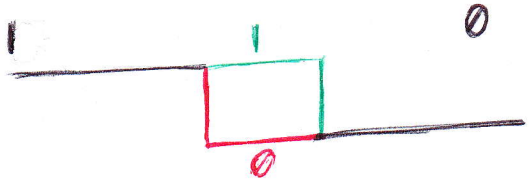
avoid transiet output pulse
- 1: to prevent static-1 hazard
- 0: to prevent static-0 hazar
- 0/1: no hazard



race free
- race condition for ⓒ → ⓐ transition!?
- break the edge to transit c → b → a

4b: ⓑ 11→10 c → ⓒ

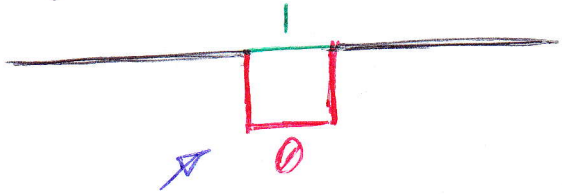
output: 1 1 0 0



we have transiet anyway
Can be set 0 or 1

2b: ⓐ 01→11 b → ⓑ

output: 1 1 1 0



to avoid transiet output
we assign 1