MODULE-5
APPLICATIONS OF DIGITAL CIRCUITS

Design of sequence detector:

The specified i/p sequence can be detected by using a sequential machine called as sequence detector. This is circuit o/p goes high when a prescribed i/p sequence occurs.

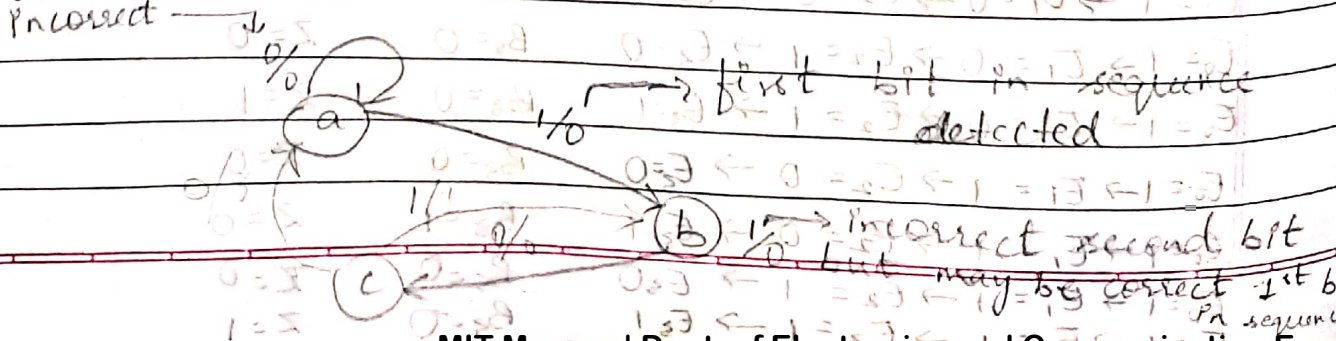
A typical i/p sequence & the corresponding o/p sequence for a described i/p sequence 101 is given as below.

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

Ex: Design a mealy type detector to detect serial type i/p of 101.

→ In a mealy type of design the no: of states in the state diagram are equal to the no: of i/p's in the desired i/p sequence.

In this example we have 3 bits in the sequence so we have 3 states in the state diagram. Once the no: of states are known, we have to draw the direction lines with states of i/p's & o/p's. Now let us start drawing the direction lines assuming state A as an initial state



State A : It checks for 1, when i/p is 1 we have detected the first bit in the sequence hence we have to go to the next state to detect the next bit in the sequence.

When i/p is zero we have to remain in the state A because bit zero is not the first bit in the sequence. In both the cases o/p is zero, since we have not yet detected all the bits in the sequence.

State 'B' : when i/p is zero, we have detected the second bit in the sequence & hence we have to go to the next state to detect the next bit in the sequence.

When i/p is 1, we have to remain in state 'B' because ~~we~~ 1 which we have detected may start the sequence or restart the sequence when o/p is still in zero.

State C: As explained for state a & state b, if the desired bit is detected we have to go to the next state otherwise we have to go to the previous state from where ~~the desired sequence~~ we can continue the desired sequence.

When complete sequence is detected, we have to make the o/p high i.e, x = 1 & we have to go to the initial state. In this case it is B'.

From the above state diagram we can determine state table as shown below.

Present state	Next state		Output	
	$x=0$	$x=1$	$x=0$	$x=1$
a	a	b	0	0
b	c	b	0	0
c	a	b	0	1

Since there are 3 states we need two flip flops & assigning state $A=00$, assigning state $B=01$, assigning state $C=10$, then we can determine the table as shown below

Present state		Next state				Output	
		$x=0$		$x=1$		z	
A	B	A^+	B^+	A^+	B^+	$x=0$	$x=1$
0	0	0	0	0	1	0	0
0	1	1	0	0	1	0	0
1	0	0	0	0	1	0	1

K-map simplification for A^+, B^+ & z.

for A^+:

AB \ x	0	1
00	0	0
01	1	0
11	X	X
10	0	0

$A^+ = B\bar{x}$

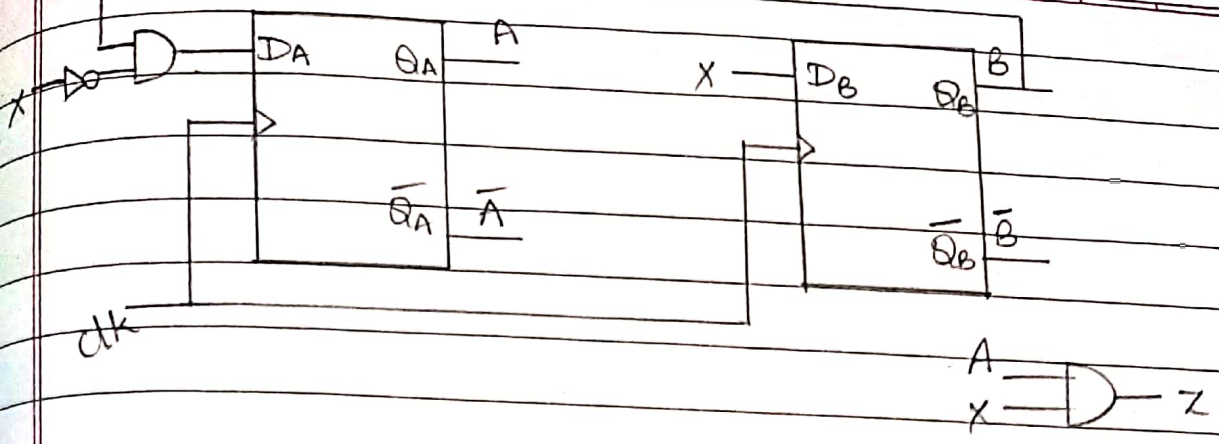
for B^+

AB \ x	0	1
00	0	1
01	0	1
11	X	X
10	0	1

$B^+ = x$

AB \ x	0	1
00	0	0
01	0	0
11	X	X
10	0	1

$z = Ax$



Construction of state graphs for the sequence detectors

2) Design a moore type sequence detector to detect a sequence serial i/p of 101.

→ In moore type design the o/p dependance only on F/F states. ∴ In the state diagram of moore machine the o/p is return with the state or without state instead of with the transition b/w the state.

Let us start, with state 'a' as initial state when i/p is '1'. we have detected the first bit in the sequence. Hence we have to the next state to detect the next sequence

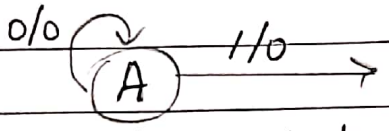
* When i/p is '0' we have to remain in same state "a" giving o/p '0'.

* State 'b' when i/p is "0" we have to go to next state i.e, state "c" because second bit is detected in the sequence

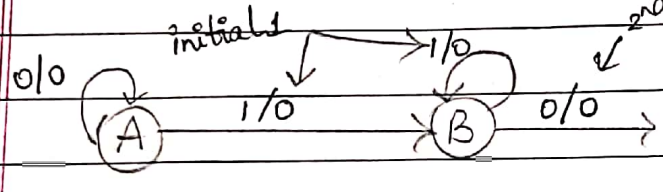
* When i/p is "1" we have to remain in state 'b' or 1 which we have detected may start the sequence & o/p is still 0.

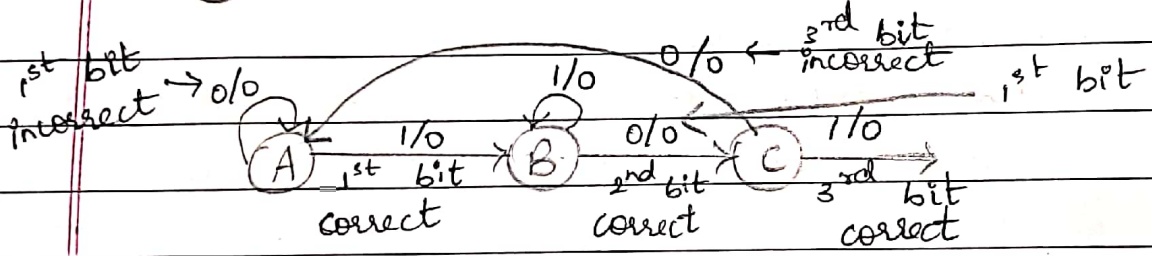
Construct a mealy state diagram that detects the sequence 10110

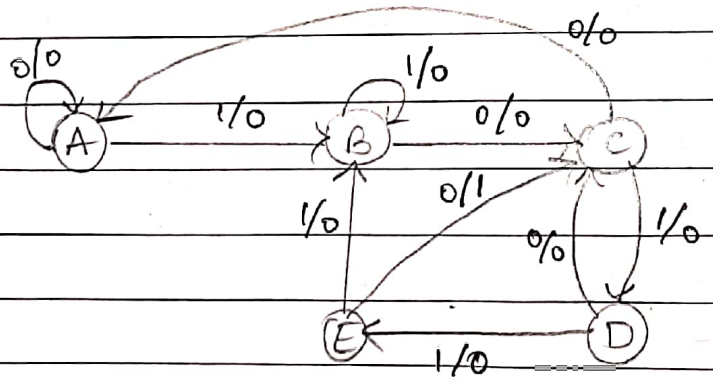
for eg: X = 10110110110
Y = 00001001001



Starting state







Construction of state graph for the sequence detectors:

Guide lines

DATE: PAGE:

(1) first construct some sample input & output sequences to make sure that you have understood the problem statement.

(2) Determine under what conditions if any, the circuit should reset to its initial state.

(3) If only one or two sequences lead to a non-zero output, then a good way to start is to construct a partial state graph for those sequences.

(4) Another way to get started is to determine what sequences or groups must be remembered by the circuit & setup states accordingly.

(5) Each time you add an arrow to the state graph, determine whether it can go to one of the previously defined states or whether a new state must be added.

(6) Check your graph to make sure that there is one & only 1 path leaving each state for each combination of values of the input variables

(7) When your graph is complete test it by applying input sequence wrt first step & make sure the output sequences are correct.

X = 0 0 1 0 1 0 0 1 0 0 0 1 0 0 1 1 0
↑ ↑ ↑ ↑ ↑ ↑
a b c d e f

Z = 0 0 0 1 0 1 0 1 1 0 0 0 1 0 1 0 0

In this example we will derive a state graph for a sequential circuit wrt the given i/p & o/p sequence.

At point 'a' the i/p sequence ends in 010. i.e., one of the sequences for which we are looking so the o/p $Z = 1$.

Note that overlapping sequences are allowed cos the problem statement doesnot say anything (no condition above reseting the circuit when a one o/p occurs.

At point 'c' the i/p sequence ends in 1001 & also we have o/p at d, e & f.

We will start construction of the state graph by working with two sequences which leads to output 010 & 1001.

Let us start with a reset state 'S_0' which corresponds to having received no inputs.

With this let us start with 'S_0' giving sequence zero to the state 'S_1'.

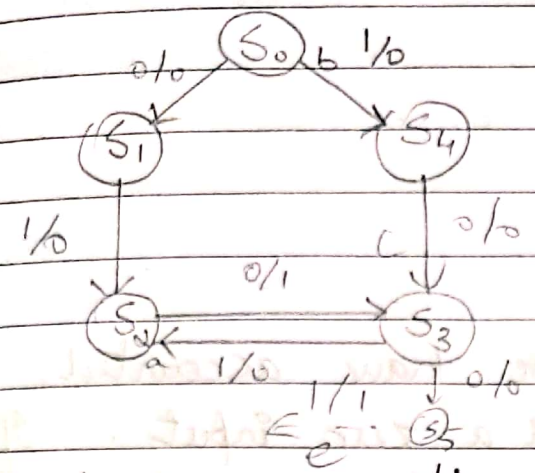
State	Sequence received
S_0	reset
S_1	0
S_2	01

In this graph S_1 corresponds to having received a sequence ending in zero. S_2 has received a sequence ending in 01, & S_3 has then if we get another '0' is S_2 we goto S_3 to get o/p 1. This is the correct sequence & again the sequence ends at 010.

DATE: PAGE:

Next we will construct the part of the state graph corresponding to the sequence.

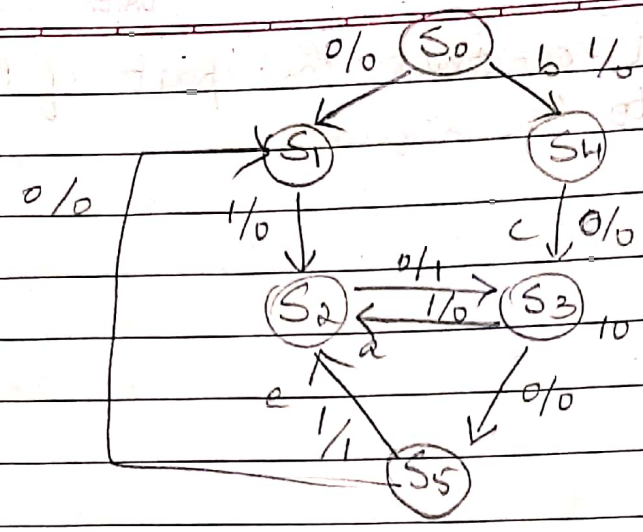
State	Sequence received
S_0	reset
S_1	0
S_2	01
S_3	10



Again we will start with 'S0' state (reset) The i/p received for S_4 is bit 1 from 'S' that creates a new state S_4. At S_4 o/p zero i.e., input to next state then from S_4 then I have to make transition from

After coming from S_4 to S_3 [C], next is I have to detect next bit i.e., zero. It is better to create a new state S_5 from S_3 & the state corresponding to sequence is 100 with this we have created a new state S_5.

If we get a one in i/p S_5 that completes the sequence 1001 & gives o/p 1 as indicated by the arrow again we ask the question whether we can go back or



In state S_1 we have accounted for a one input but not for a zero input. If we are in S_1 & if we get a zero input to which state shld we go?

If a zero i/p occurs in S_1 then the sequence ending in S_1 is 01. ∴ S_1 shld remain same state coz 00 is not part of either of the i/p sequence for which we are looking. ∴ We can ignore zero & stay in S_1 with arrow 'f'.

No matter how many '0' occurs, we still in the state S_1.

S_2 we taken care of '0' i/p but not the 1 i/p so if 1 is received sequence ends in 11, coz 11 is not part of either of 010 or 1001 sequence.

If we donot need a state which corresponds to a sequence 11, we can't stay in S_2 coz S_2 corresponds to a sequence ending in 01 ∴ We have to go to S_4 which corresponds to a sequence received ending in '1' → g.

S_3 has is already having

S_3 is already having

Next we have to examine S_4. If '1' is received in S_4, the i/p sequence ends in 11. i.e., extra 1. Eq give arrow 'h' coz 11 is not a part of required sequence for which we are looking.

If we get zero, the sequence ends in 000, this 000 is not contained in either of the two sequences, So we have to go back to S_1 coz S_1 corresponds to having received a sequence ending in two or more zeroes.

Design example of a sequential circuit wrt to state graph:-

We will now design a sequential ckt to convert a BCD to excess 3 code with the help of (or) by constructing

X				Z			
input (BCD)				Output (excess-3)			
t_3	t_2	t_1	t_0	t_3	t_2	t_1	t_0
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

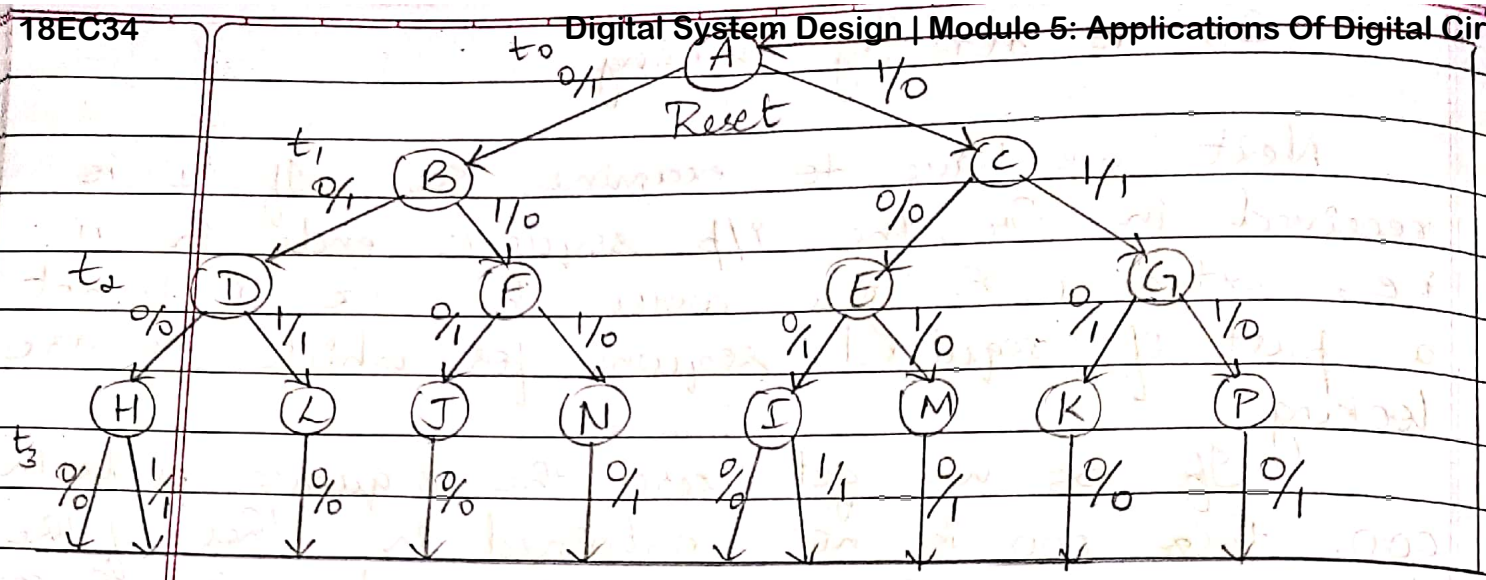


fig: State graph for code converter.

Once the state graph is drawn next we have to construct the state table

Time	i/p sequence received (LSB first)	Present state	Next state X=0	Next state X=1	Present o/p (z) X=0	Present o/p (z) X=1
t_0	reset	A	B	C	1	0
t_1	0	B	D	F	1	0
	1	C	E	G	0	0
t_2	0 0	D	H	L	0	1
	0 1	E	I	M	1	0
	1 0	F	J	N	1	0
	1 1	G	K	P	1	0
t_3	0 0 0	H	A	A	0	1
	0 0 1	I	A	A	0	1
	0 1 0	J	A	-	0	-
	0 1 1	K	A	-	0	-
	1 0 0	L	A	-	0	-
	1 0 1	M	A	-	1	-
	1 1 0	N	A	-	1	-
	1 1 1	P	A	-	1	-

After constructing the table next we have to reduce the table using row matching. When matching rows contains dashes [don't care], the dash will match with any state or with output value. By seeing the table we observe

that $H \equiv I \equiv J \equiv K \equiv L$, $M \equiv N \equiv P$ with this the table reduces to 7 rows & coz we also know that $F \equiv E \equiv G$.

Time	Present state	Next state X=0	Next state X=1	Present Output(z) X=0	Present Output(z) X=1
t_0	A	B	C	1	0
t_1	B	D	E	1	0
	C	E	E	0	1
t_2	D	H	H	0	1
	E	H	M	1	0
t_3	H	A	A	0	1
	M	A	–	1	–

After eliminating I, J, K, L, N & P, we find that $E \equiv F \equiv G$ & the table reduces to 7 rows. Now after reducing the state table we have totally 7 states namely A, B, C, D, E, H, M. wrt a 2X4 k-map we have to assign.

wrt to the guidelines the 7 states must be arranged wrt the adjacent assignments as shown below.

Q_2Q_3 \ Q_1	0	1
00	A	B
01	–	C
11	H	D
10	M	E

We have to construct the transition table by considering the assignment map.

		$Q_1^+Q_2^+Q_3^+$		Z	
	$Q_1Q_2Q_3$	x=0	x=1	x=0	x=1
A	0 0 0	100	101	1	0
B	1 0 0	111	110	1	0
C	1 0 1	110	110	0	1
D	1 1 1	011	011	0	1
E	1 1 0	011	010	1	0
H	0 1 1	000	000	0	1
M	0 1 0	000	xxx	1	x
–	0 0 1	xxx	xxx	x	x

Next we have to construct the k-map by using D-flipflop. coz wkt in D-flipflop o/p follows the i/p, D_1 D_2 D_3 is equivalent $Q_1^+Q_2^+Q_3^+$ respectively

Q_2Q_3 \ Q_1x	00	01	11	10
00	1	1	1	1
01	x	1	1	x
11	0	0	0	0
10	0	0	0	x

$D_1 = Q_1^+ = Q_2'$

Q_2Q_3 \ Q_1x	00	01	11	10
00	0	1	1	0
01	x	1	1	x
11	0	1	1	0
10	0	1	1	x

$D_2 = Q_2^+ = Q_1$

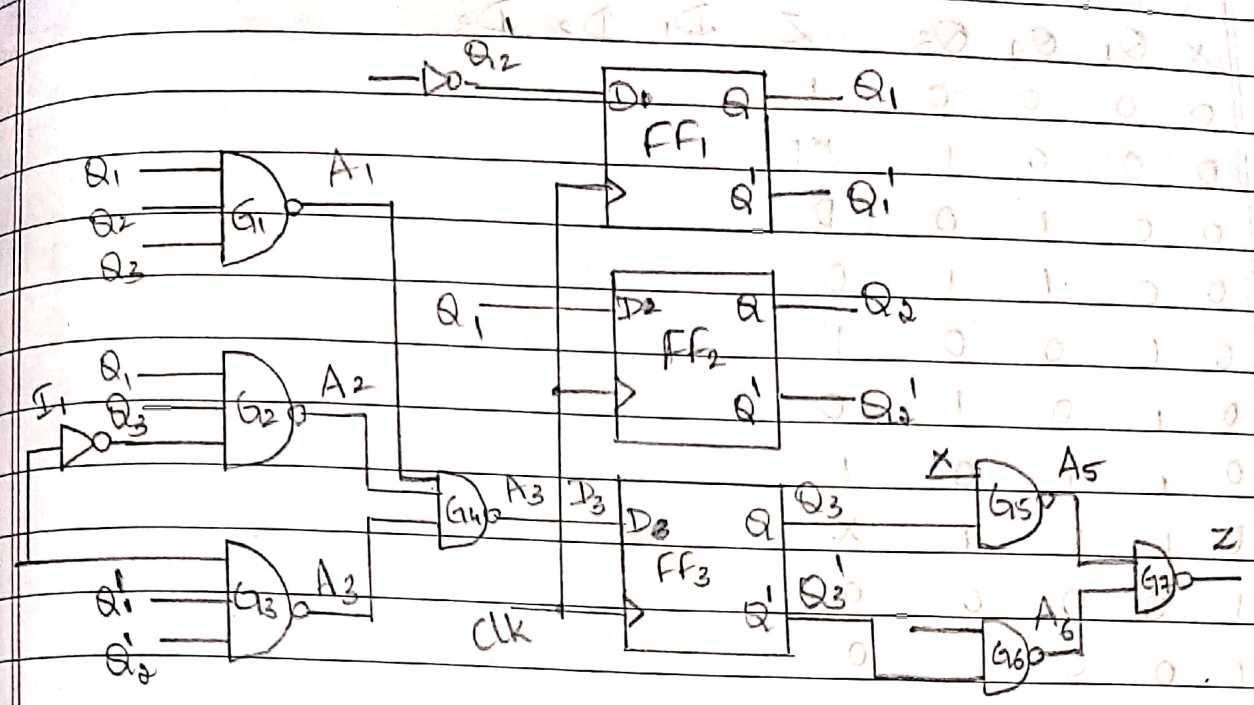
Q_2Q_3 \ xQ_1	00	01	11	10
00	0	1	0	1
01	x	0	0	x
11	0	1	1	0
10	0	1	0	x

$D_3 = Q_3^+ = Q_1Q_2Q_3 + \bar{x}Q_1\bar{Q}_3 + x\bar{Q}_1Q_2$

Q_2Q_3 \ xQ_1	00	01	11	10
00	1	1	0	0
01	x	0	1	x
11	0	0	1	1
10	1	1	0	x

~~$D_4 = Q_4^+ = \bar{x}Q_1Q_2\bar{Q}_3 + xQ_3$~~

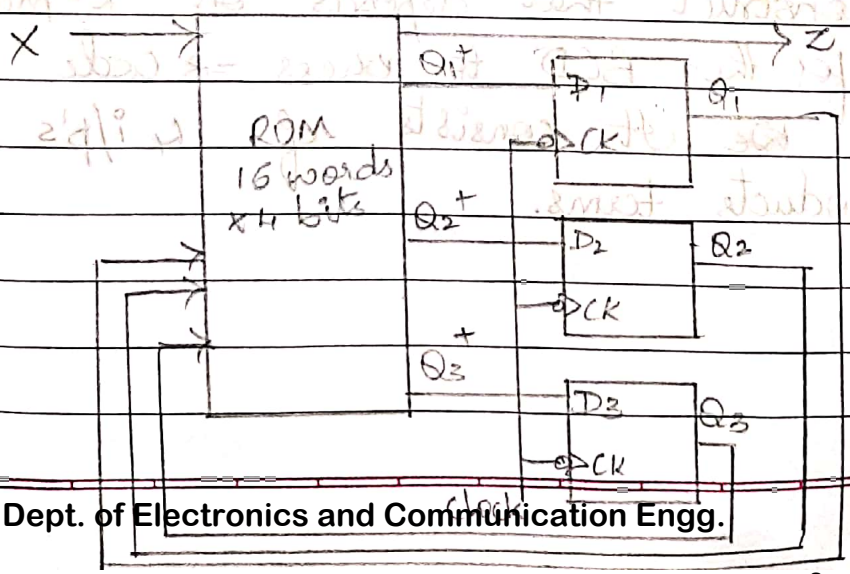
$Z = \bar{x}\bar{Q}_3 + xQ_3$



Design of sequential circuits using ROM's & PLA's:

A sequential ckt can be easily designed by using a ROM & F/F's.
ROM that can

for the sake of simplicity let us assign the previous eg of BCD to excess code inverter to implement the sequential ckt



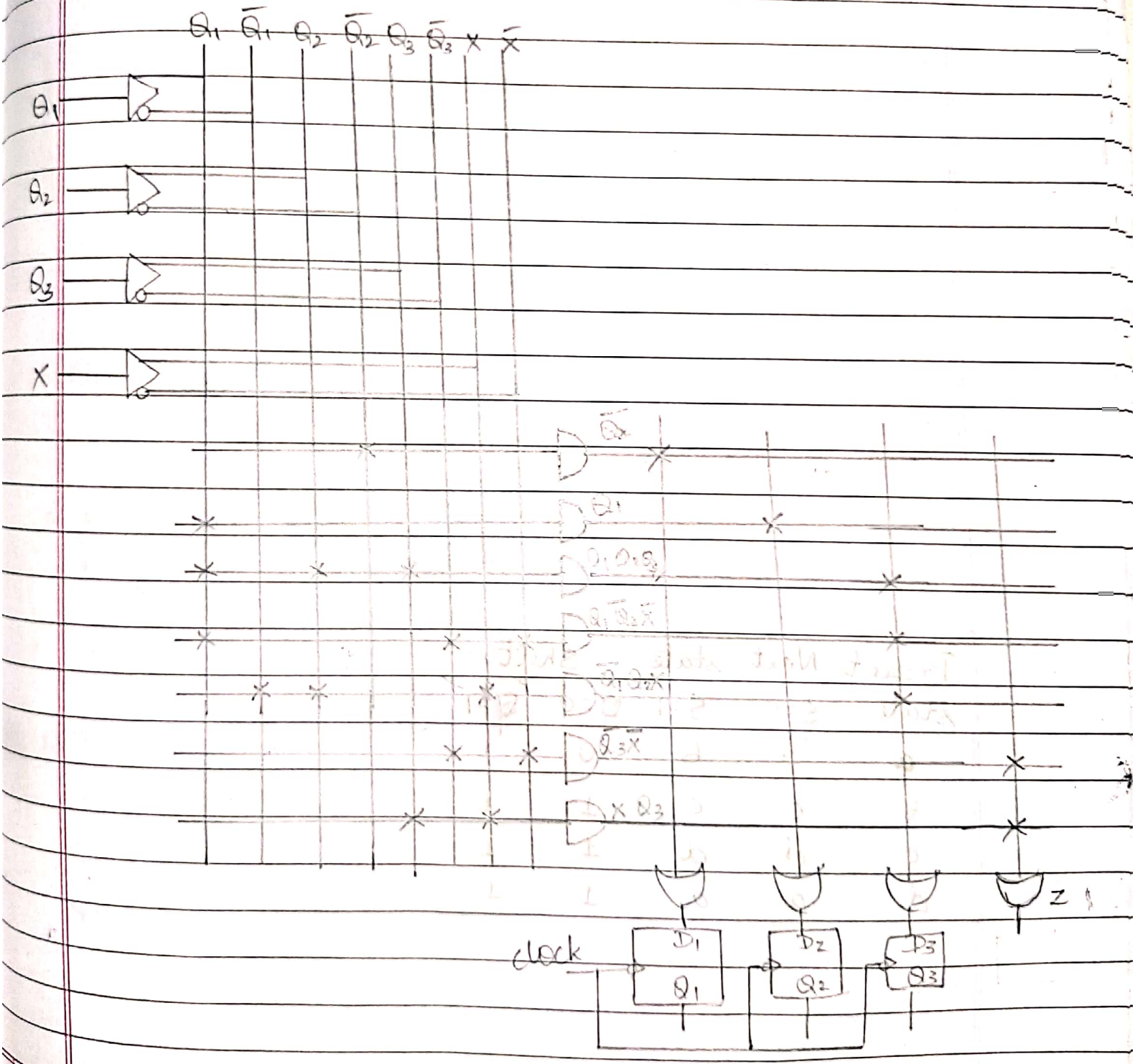
i/p				o/p			
x	Q_1	Q_2	Q_3	Z	D_1	D_2	D_3
0	0	0	0	1			
0	0	0	1	X1			
0	0	1	0	0			
0	0	1	1	0			
0	1	0	0	1			
0	1	0	1	0			
0	1	1	0	1			
0	1	1	1	X			
1	0	0	0	0			
1	0	0	1	0			
1	0	1	0	1			
1	0	1	1	1			
1	0	0	0	0			
1	1	0	1	1			
1	1	1	0	X			
1	1	1	1	X			

wrt simplification of k-map we get to know that $D_1 = Q_1^+ = \bar{Q}_2$, $D_2 = Q_2^+ = Q_1$, $D_3 = Q_3^+ = Q_1Q_2Q_3 + \bar{x}Q_1\bar{Q}_3 + x\bar{Q}_1Q_2$ & $Z = \bar{x}\bar{Q}_3 + xQ_3$.

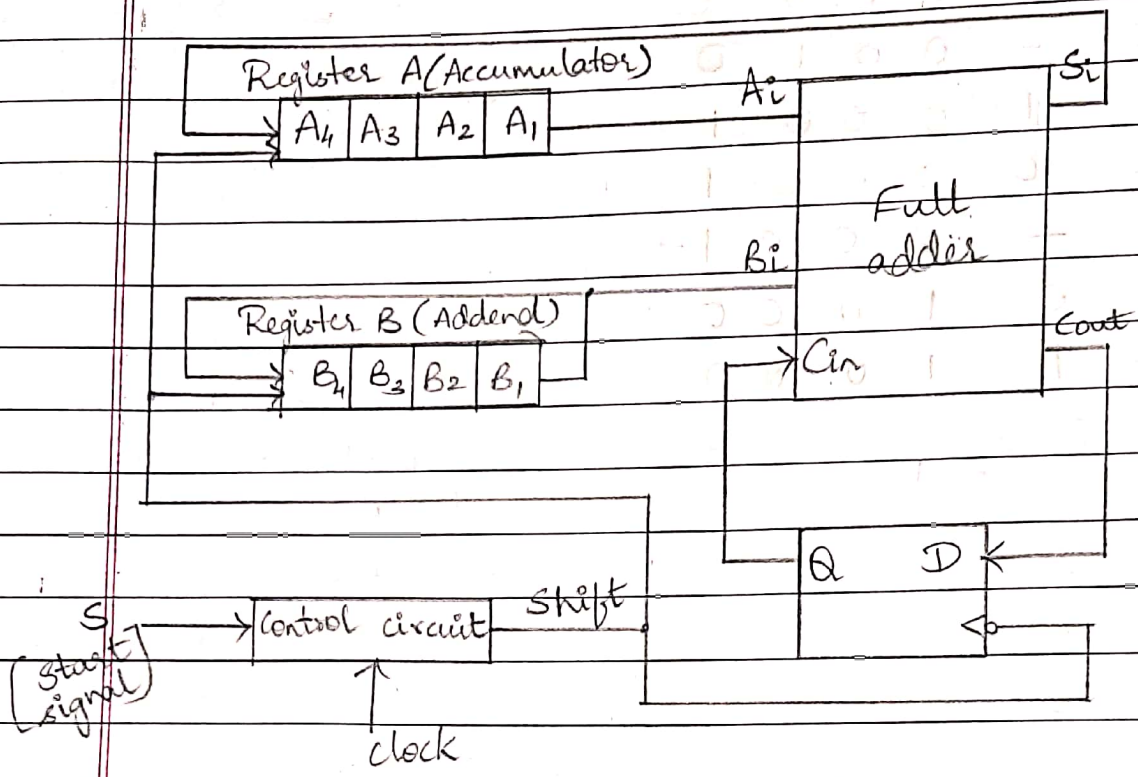
The PLA table for which now we are going to construct that depends on k-map simplification obtained for the BCD to excess -3 code for D_1, D_2, D_3 & Z. It consists of 4 i/p's, 4 o/p's & 7 products terms.

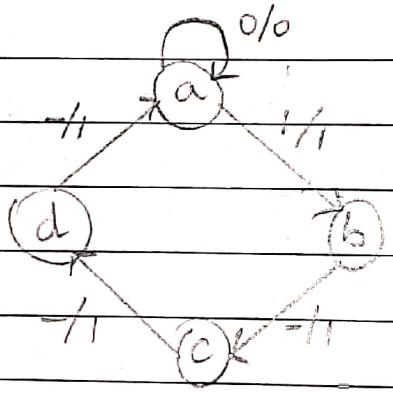
DATE: PAGE:

X	Q_1	Q_2	Q_3	z	D_1	D_2	D_3
–	–	0	–	0	1	0	0
–	1	–	–	0	0	1	0
–	1	1	1	0	0	0	1
0	1	–	0	0	0	0	1
1	0	1	–	0	0	0	1
0	–	–	0	1	0	0	0
1	–	–	1	1	0	0	0



Serial adder with accumulator:





high
– ⇒ ⊓ ⇒ 1

Present state	Next state		Shift	
	$S=0$	$S=1$	$S=0$	$S=1$
a	a	b	0	1
b	c	c	1	1
c	d	d	1	1
d	a	a	1	1

DATE: PAGE:

PS	NS x=0		NS x=1		F/F i/p's x=0				F/F i/p's x=1				o/p	
x y	x^+	y^+	x^+	y^+	J_x	K_x	J_y	K_y	J_x	K_x	J_y	K_y	x=0	x=1
0 0	0	0	0	1	0	–	0	–	0	–	1	–	0	1
0 1	1	0	1	0	1	–	–	1	1	–	–	1	1	1
1 0	1	1	1	1	–	0	1	–	–	0	1	–	1	1
1 1	0	0	0	0	–	1	–	1	–	1	–	1	1	1

K-map for J_x

S \ xy	00	01	11	10
0	0	1	–	–
1	0	1	–	–

=

K-map for K_x

S \ xy	00	01	11	10
0	x	x	1	0
1	x	x	1	0

=

K-map for J_y

	00	01	11	10
0				
1				

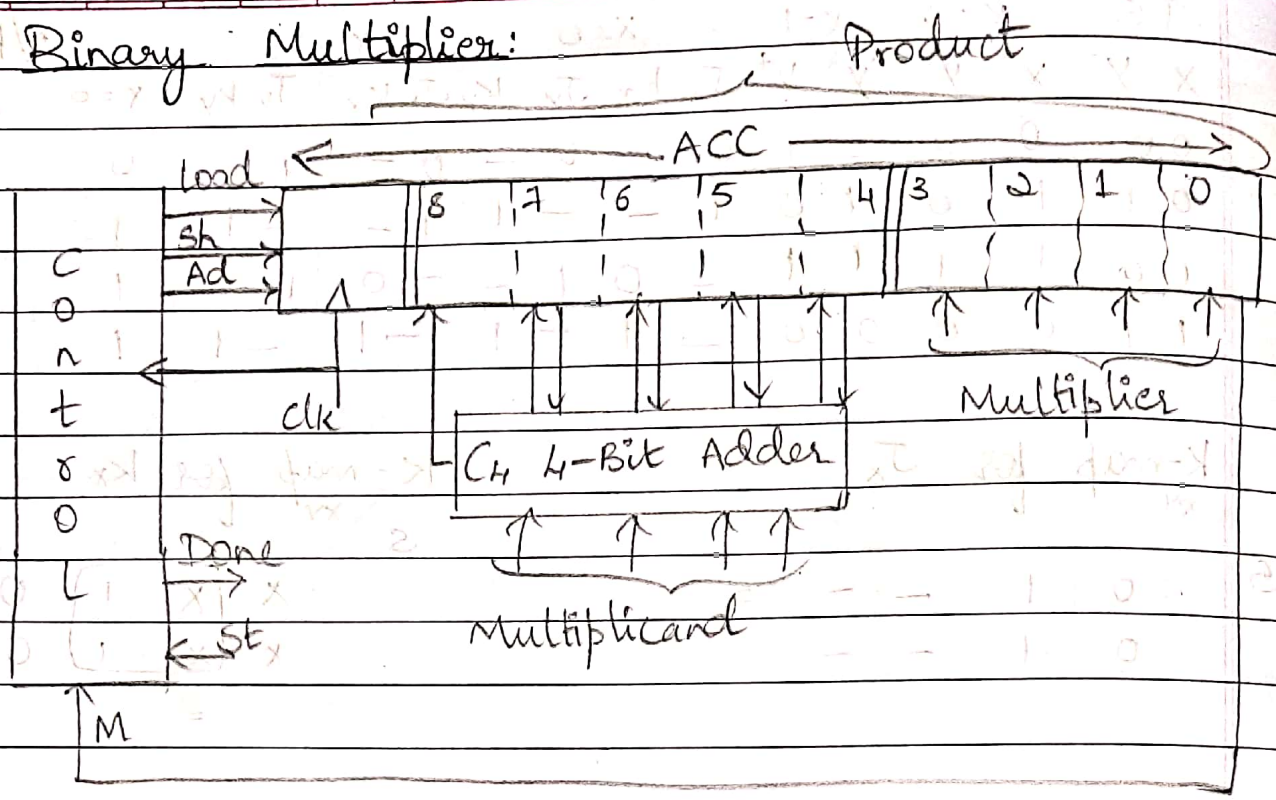
JK i/p		F/F o/p	
J_x	K_x	Q_n	Q_{n+1}
0	0	0	0
0	1	1	0
1	0	0	1
1	1	1	0

SR i/p.			
S	R	Q_n	Q_{n+1}
0	0	0	0
0	1	1	0
1	0	0	1
1	1	1	0

D i/p		
D	Q_n	Q_{n+1}
0	0	0
1	1	1

T i/p		T o/p
T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Binary Multiplier:



Block diagram of Binary Multiplier

Shift & Add

1011

1101

n x n = 2n

n
+n
n+1

0 0 0 0 0 | 1 0 1 1 ← M (11)

1 1 0 1 (13)

0 1 1 0 1 1 0 1 1

0 0 1 1 0 1 1 0 1 ← M

1 1 0 1

1 0 0 1 1 1 1 0 1

0 1 0 0 1 1 1 1 0 ← M

0 0 1 0 0 1 1 1 1 ← M

1 1 0 1

1 0 0 0 1 1 1 1 1

0 1 0 0 0 1 1 1 1 (143)

Initial contents of the product register is 0 0 0 0 0 . Add multiplicand (13) coz 1 1 0 1