

Assignment No:- 1

Title :- Binary adder & subtractor

Apparatus:-

Digital board, JIP - 4 Patch cords IC-74LS86
 IC-74LS32, IC 74LS08 (IC-74LS04 & IC-74LS00
 & Required logic gates if any.)

full adder circuit

Truth table

Input			Output	
A	B	Cin	sum	carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

K-map simplification

for sum

A	B _{cin}			
	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$$\text{Sum} = \bar{A}\bar{B}\text{cin} + \bar{A}B\bar{\text{cin}} + A\bar{B}\text{cin} + ABC\text{cin}$$

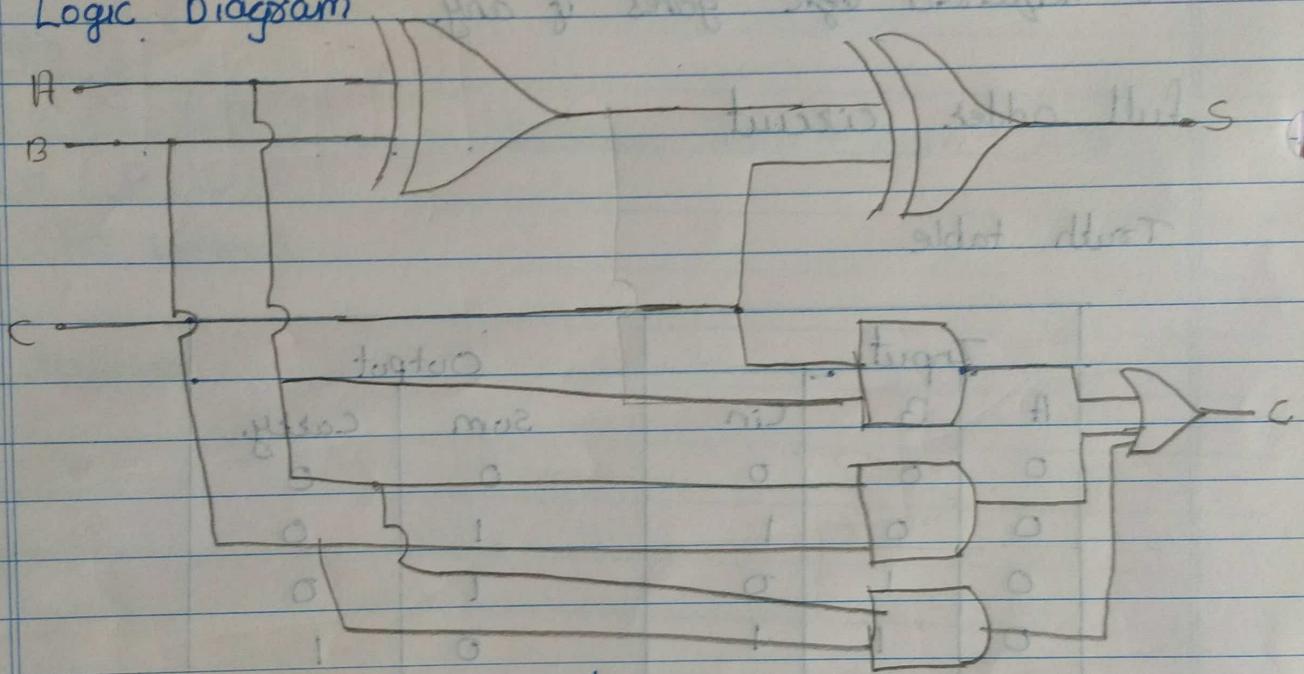
1-bit full subtractor

for carry instead of set the fourth column

A	B _{Cin}	00	01	11	10
0	0	0	1	0	0
1	0	1	1	1	1

$$\text{Carry} = AB + AC_{in} + BC_{in}$$

Logic Diagram

Full Subtractor circuit

Truth table.

Input			Output	
A	B	Bin	Diff	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	0
1	0	1	1	0
1	0	0	0	0
1	1	1	1	1

K-map simplification

A\B Bin	00	01	10	11
0	0	1	0	1
1	1	0	1	0

(for difference)

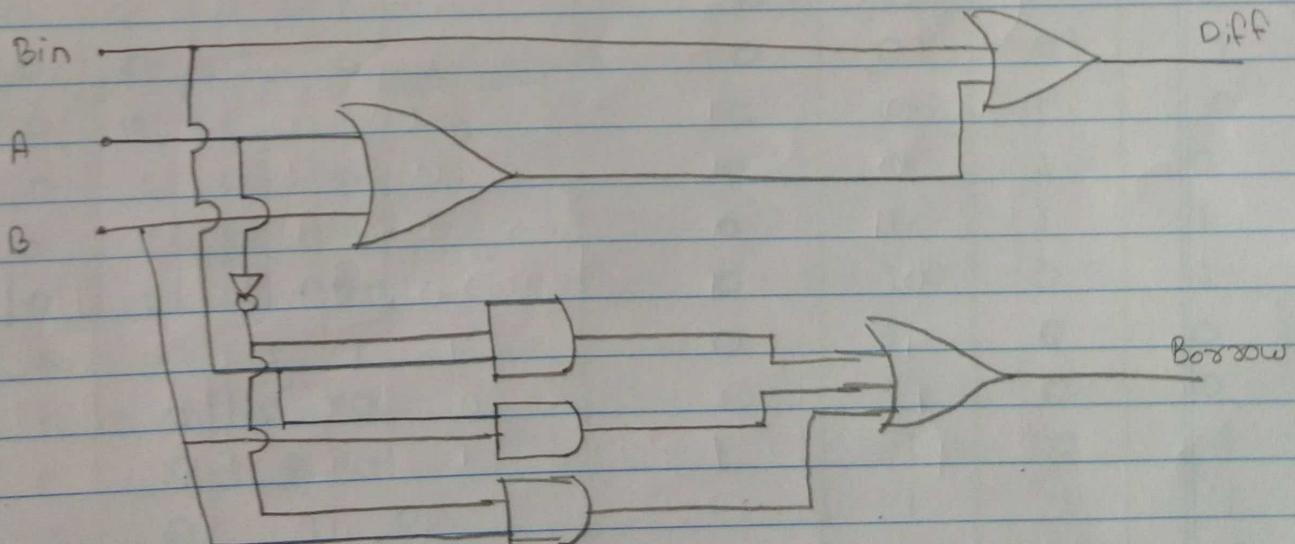
$$\text{diff} = \bar{A}\bar{B}\text{Bin} + \bar{A}B\bar{\text{Bin}} + A\bar{B}\bar{\text{Bin}} + AB\text{Bin}$$

A\B Bin	00	01	11	10
0	0	1	1	1
1	0	0	1	0

(for borrow)

$$\text{borrow} = \bar{A}\text{Bin} + B\text{Bin} + \bar{A}B$$

Logic diagram



Conclusion

understood how to write truth table, Kmap for full Subtractor 3, adder & Draw logic diagram from expression derived in Kmap simplification.

Q. i + n10 + n11 answers

b_1	n	n_10	n_11	A
0	0	0	0	0
0	1	0	1	1

(Answered 20)

margin size

Assignment No:-2

Title :-

Code Converter

Apparatus :-

Digital board, UTP - 4, Patch cords, IC-74LS86, IC-74LS32, IC-74LS08/IC-74LS04 & required logic gates if any.

1. Binary to Grey Code Converter

Truth Table:

Binary Code Input				Grey Code Output			
B ₃	B ₂	B ₁	B ₀	U ₃	U ₂	U ₁	U ₀
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	0	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	1	1
0	1	1	1	0	0	0	1
1	0	0	0	1	1	1	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	0	1	0
1	1	0	0	1	0	1	1
1	1	0	1	1	0	0	1
1	1	1	0	1	0	0	0
1	1	1	1	0	0	0	0

$$\bar{B}_3 + \bar{B}_2 \cdot \bar{B}_0 = 0$$

$$\bar{B}_3 + \bar{B}_2 \cdot \bar{B}_1 = 0$$

1. 2

SOP form will

K map simplification

for C_3

$B_3\ B_2$	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

$$\text{toget } C_3 = B_3 \oplus B_2$$

for C_2

$B_3\ B_2$	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

$$\begin{aligned} \text{toget } C_2 &= \overline{B_3} B_2 + B_3 \overline{B_2} \\ &= B_3 \oplus B_2 \end{aligned}$$

for C_1

$B_3\ B_2$	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	1	1	0	0
10	0	0	1	1

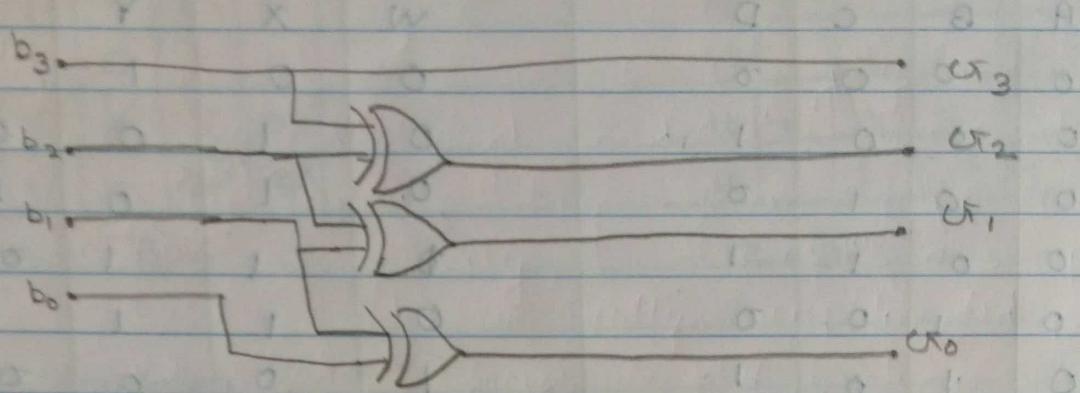
$$\begin{aligned} C_1 &= B_1 \overline{B_2} + \overline{B}_1 B_2 \\ &= B_1 \oplus B_2 \end{aligned}$$

for C_0

$B_3\ B_2$	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1

$$\begin{aligned} C_0 &= \overline{B}_1 B_0 + B_1 \overline{B}_0 \\ &= B_1 \oplus B_0 \end{aligned}$$

Logic Diagram :-



binary to gray code converted

* BCD to excess - 3

- The excess 3 binary code is unweighted self complementary BCD code.
- Self complementary property means that the 1's complement of an excess 3 number is excess 3 code of the 9's complement of the corresponding decimal number.

X	89	45
01	10 00	00
10	01 00	
00	01 10	10
X	X X	11
(X)	1	01
11		

X	89	45
01	11 10	00
00	00 00	
11	11 00	10
X	X X	11
X	1 1	01
11		

$$08 + 084 = 85 \text{ in } 100$$

$$09 + 094 = 95 \text{ in } 100$$

Kmap & Truth table

BCD(8421)

A	B	C	D		W	X	Y	Z
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
1	0	0	0		1	0	0	0
1	0	0	1		1	0	0	0
1	0	1	0		X	X	X	X
1	0	1	1		X	X	X	X
1	1	0	0		X	X	X	X
1	1	0	1		X	X	X	X
1	1	1	0		X	X	X	X
1	1	1	1		X	X	X	X

K map

		for W				
		CD	00	01	11	10
AB		00	0	0	0	0
00		0	0	0	0	0
01		0	1	1	1	1
11		X	X	X	X	X
10		1	1	X	X	X

$$W = A + BC + BD$$

		for X				
		CD	00	01	11	10
AB		00	0	1	1	1
00		0	1	1	1	1
01		1	0	0	0	0
11		X	X	X	X	X
10		0	1	X	X	X

$$X = \overline{BCD} + \overline{BD} + \overline{BC}$$

for Y

$A\bar{B}$	00	01	11	10
$\bar{A}B$	00	1	0	0
01	1	0	1	0
11	X	X	X	0X
10	1	0	X	0X

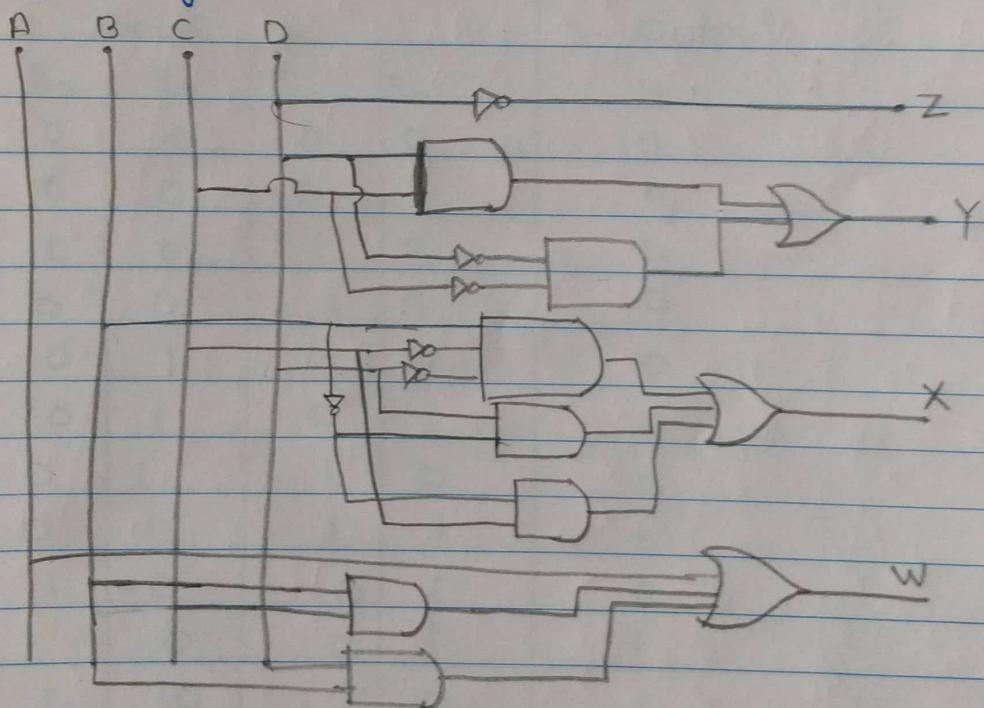
$y = \overline{CD} + CD$

for Z

$A\bar{B}$	00	01	11	10
$\bar{A}B$	00	1		1
01	1			1
11	X		X	X
10	1			X

$z = \overline{D}$

Logic Diagram



BCD To Excess 3

Conclusion :-

We understood binary to grey & BCD to excess-3 conversion.

Assignment No:- 3

Title :- BCD Adder using 7483

Apparatus : Digital board, VTP - 4 Patch cards, IC 7483
& required logic gates

★ BCD Adder

Truth Table :-

Input				Output
S ₃	S ₂	S ₁	S ₀	y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	1	1	1

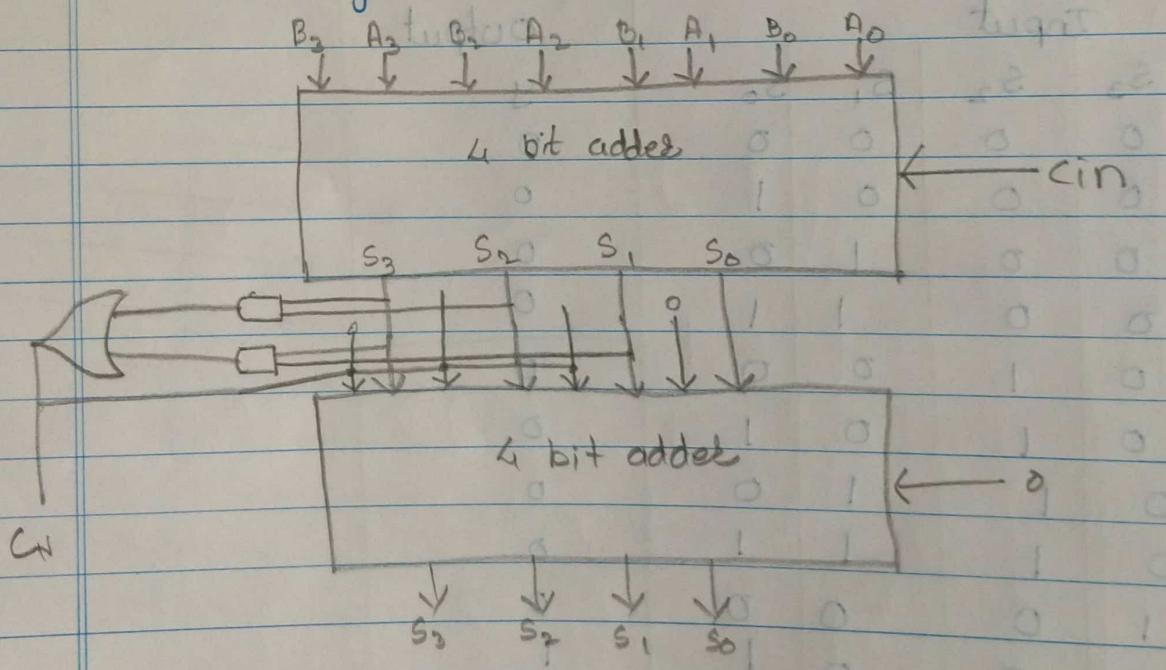
E-101 thermageddon

K map

$S_3 S_2$	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	0	0	1	1

$$y = S_3 S_2 + S_3 S_1$$

Circuit Diagram:-

Conclusion:-

We understood & implemented BCD adder using 7403.

Assignment No:- 4

Title: Multiplexers / Demultiplexers

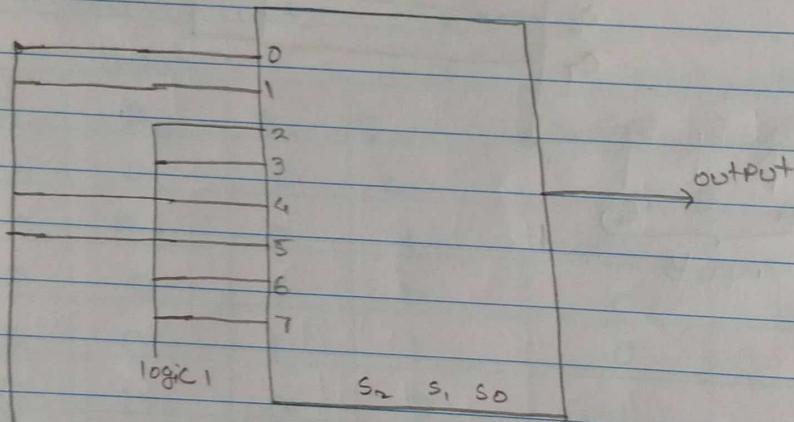
Numericals

1) Implement 8:1 mux \leftrightarrow (IC - 74153)

$$(A, B, C) = m(0, 1, 4, 5)$$

\rightarrow minterms 2, 3, 6, 4 \rightarrow logic 0

maxterms 0, 1, 4, 5 \rightarrow logic 1



2) Full adder using IC 74138

Truth Table

Inputs			Output	
A	B	Cin	sum	carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Here, sum = $\Sigma m(1, 2, 4, 7)$

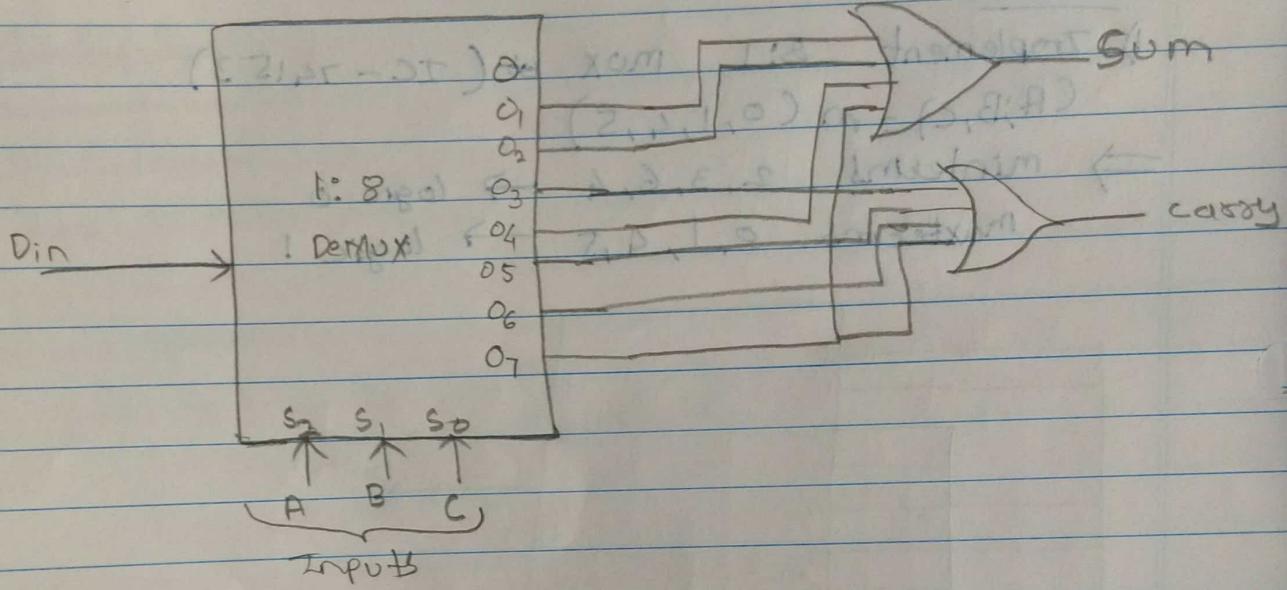
$\Sigma m(1, 2, 4, 7)$

output 0, 0, 0, 0, 0, 0, 0, 1
Should be connected to OR.

Here carry = $\Sigma m(3, 5, 6, 7)$

0, 0, 0, 0, 0, 0, 0, 1
Should be connected to OR gate.

Implementation:



Conclusion

Conclusion: Studied MUX, DEMUX & their functions.
also implemented in applications.

	bright	out	right	left	up	down
0	0	0	0	0	0	0
0	1	1	0	0	0	0
0	1	0	0	1	0	0
1	0	1	1	1	0	0
0	1	1	0	0	1	0
1	0	1	0	1	1	1
1	0	0	1	1	1	1
1	1	1	1	1	1	1

Assignment NO:-5

Title:- Digital magnitude Comparator circuit

Apparatus:- Digital board, UP-4 Patch Cords, IC-74LS86
IC-74LS32, IC-74LS08 / 74LS04 / 74LS85.

* 2 Bit Comparator :-

Truth table

Inputs				(A, A ₁) (B, B ₁) (Outputs)		
A ₁	A ₀	B ₁	B ₀	A > B	A = B	A < B
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

K map \Rightarrow - for $A > B \Rightarrow$

A ₁ A ₀	00	01	11	10
B ₁ B ₀	00	0	1	0
	01	0	0	1
	11	0	0	0
	10	0	0	1

$$A > B = A_1 A_0 \bar{B}_0 + A_1 \bar{A}_0 \bar{B}_0 \bar{B}_1 + \bar{B}_1 A_1$$

Page No. _____
Date: / /for $A = B$

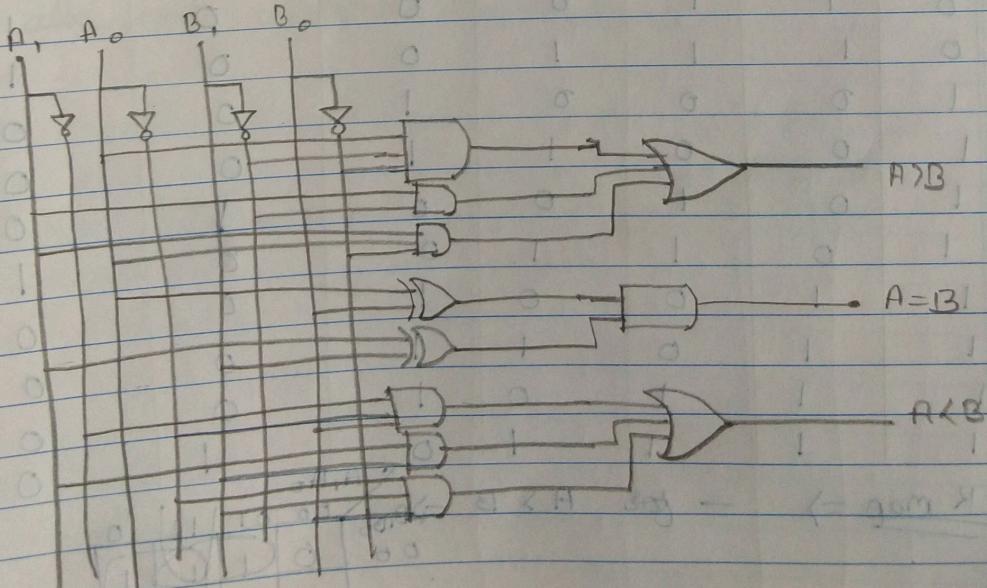
$B_1 B_0$	00	01	11	10
$A_1 A_0$	00	01	11	10
00	00	00	00	00
01	00	00	00	00
11	00	00	00	00
10	00	00	00	01

$$\begin{aligned}
 (A=B) &= \bar{A}_1 \bar{A}_0 \bar{B}_1 \bar{B}_0 + \bar{A}_1 \bar{A}_0 \bar{B}_1 B_0 + A_1 \bar{A}_0 B_1 B_0 + A_1 \bar{A}_0 B_1 \bar{B}_0 \\
 &= \bar{A}_1 \bar{B}_1 (\bar{A}_0 \bar{B}_0 + A_0 B_0) + A_1 B_1 (A_0 B_0 + \bar{A}_0 \bar{B}_0) \\
 &= (A_0 \oplus B_0) (A_1 \oplus B_1)
 \end{aligned}$$

- for $A < B$

$B_1 B_0$	00	01	11	10
$A_1 A_0$	00	01	11	10
00	00	00	00	00
01	00	00	00	00
11	00	01	01	00
10	01	01	00	01

$$A < B = \bar{A}_1 \bar{A}_0 B_0 + \bar{A}_1 B_1 + \bar{A}_0 B_1 B_0$$



Conclusion:-

We understood working of 2 bit Comparator & all concepts related to it.

Assignment NO:- 6

Title: parity generator checker using X-OR

Apparatus:-

Digital Board CTP-4, Patch cords, IC-74LS86,
8 required logic gate.

* Parity Generator checker

Truth table of Even parity generator

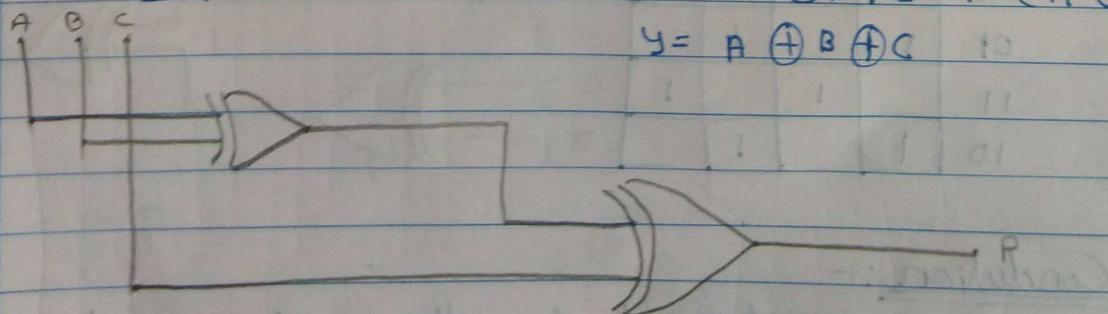
3 bit message			Even parity generator	
A	B	C	y	
0	0	0	01	
0	0	1	10	
0	1	0	11	
0	1	1	00	
1	0	0	11	
1	0	1	00	
1	1	0	01	
1	1	1	10	

K-map

A \ C	00	01	11	10
0	1			1
1	1	1		

$$\begin{aligned}
 Y &= \bar{A}\bar{B}C + \bar{A}B\bar{C} + ABC + A\bar{B}\bar{C} \\
 &= (\bar{A}\bar{B} + AB)C + (\bar{A}B + A\bar{B})\bar{C} \\
 &= (A \oplus B)C + (A \oplus B)\bar{C}
 \end{aligned}$$

$$Y = A \oplus B \oplus C$$



Truth Table

Even parity checked

4 bit received message

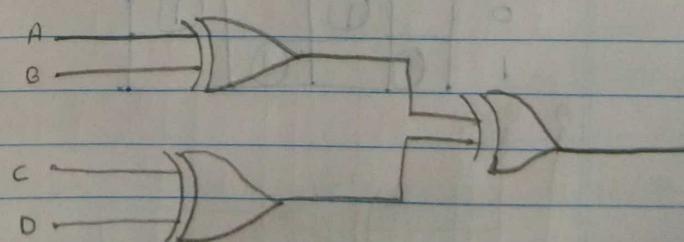
parity check error C_P

A	B	C	D	C_P
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

K-map

	CD	AB	00	01	11	10
0	0	0	0	0	1	1
0	1	0	1	1	1	0
1	0	0	1	1	0	0
1	1	0	0	0	0	1

logic/circuit



Conclusion:-

We learned how the parity is checked using logic gates.

Assignment No:-7

Title :- flip flop conversion

i) JK to D

ii) JK to T

Apparatus :- Digital board, CTP & Patch cards, IC-74LS86, 74LS32, 74LS08, 74LS04 & 74LS76.

* JK to D flip flop

Truth Table of D & FF

D	Q _{n+1}
0	0
1	1

Excitation Table

Q _n	Q _{n+1}	J	K
0	0	0	X
0	1	X	1
1	0	X	1
1	1	X	0

Boolean expression

Q _n	D	Q _{n+1}	J	K
0	0	0	0	X
0	1	1	1	X
1	0	0	X	1
1	1	1	X	0

- for J

Q _n	0	1
0	0	1
1	X	X

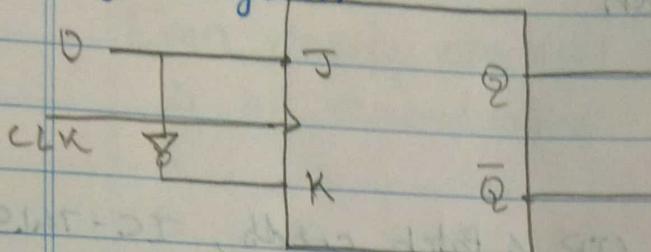
$$J = D$$

- for K

Q _n	0	1
0	X	X
1	X	0

$$K = \overline{D}$$

Logic Diagram



* JK to T flip flop

Truth Table for T FF

T	Q	Q_{n+1}
0	0	0
1	0	1
0	1	1
1	1	0

Excitation Table

Q_n	Q_{n+1}	J	K
0	0	0	X
1	1	X	0
0	1	1	X
1	0	X	1

Characteristic Table

T	Q_n	Q_{n+1}	J	K
0	0	0	0	X
0	1	1	X	0
1	0	1	1	X
1	1	0	X	1

- for J

+ for Q_n

T	0	1
0	0	X
1	(1)	X

$J = T$

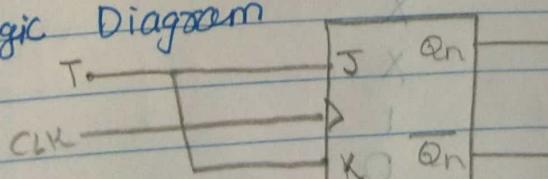
- for K

+ for Q_n

T	0	1
0	X	0
1	(X)	1

$K = T$

Logic Diagram



Conclusion

We understood & converted one FF to another.

Assignment NO:-8

Title :- 2 bit & 3 bit ripple counter using master slave JK FF

Problem Statement :- Design a 2 bit & 3 bit counter using MS JK FF

Theory

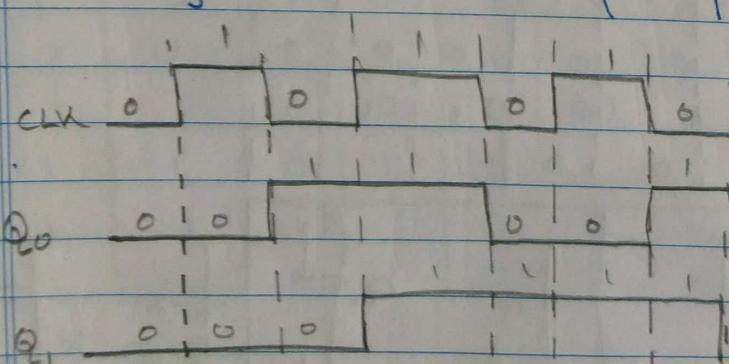
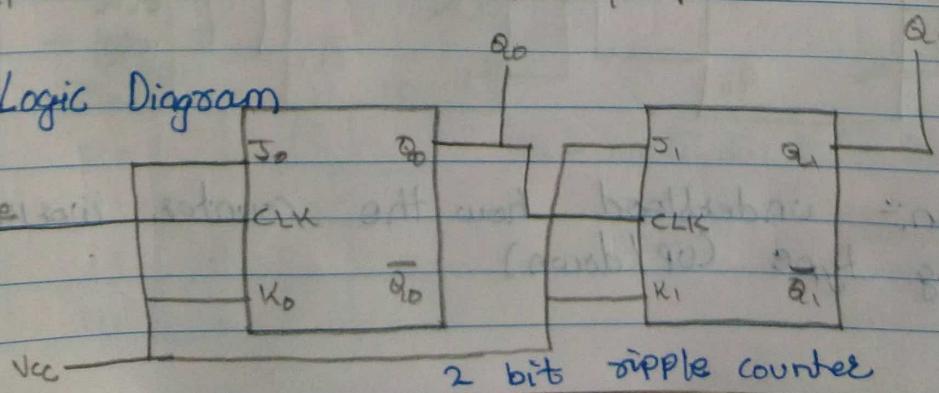
* 2 bit ripple counter

It is a binary counter which is a 2 MOD 4 counter which counts upto 2 bit state values i.e $2^2 = 4$ values. The FF having similar condition for toggling T & JK are used to construct the ripple counter.

Truth Table

Counter state

	Q_1	Q_0
0	0	0
1	0	1
2	1	0
3	1	1

Logic Diagram

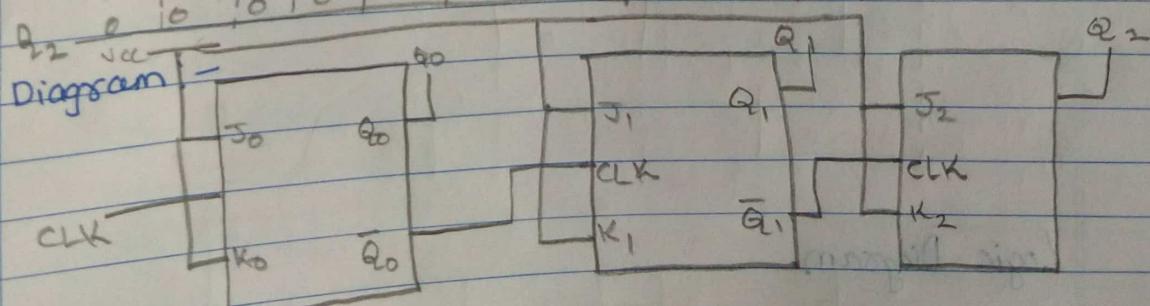
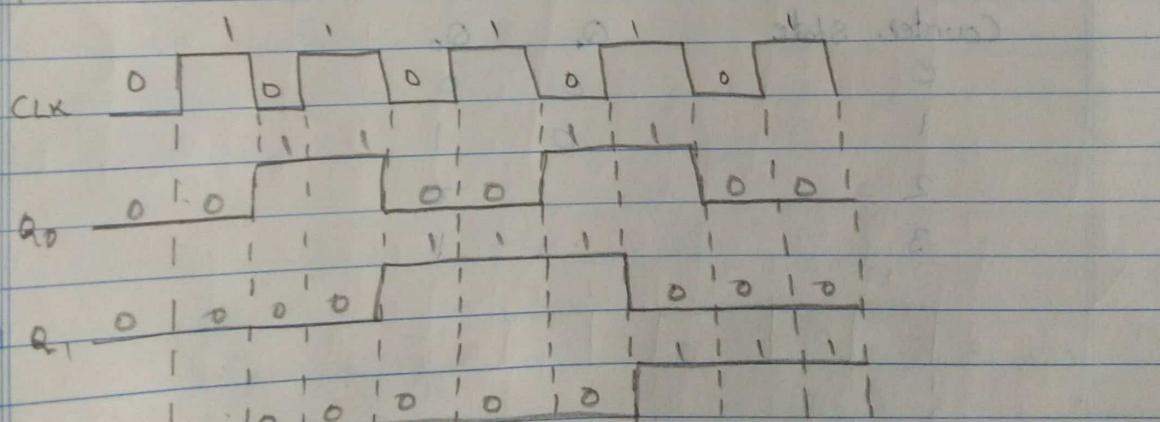
* 3 bit ripple counter

It counts upto state value i.e. $2^3 = 8$ values.

FF having similar conclusion for toggling like T & JK
are used to construct it.

Truth Table

Counter State	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1



Conclusion:- understood how the counter works & its
counting type (up/down)

Assignment No:- 9

Title:- Synchronous 3 bit updown counter.

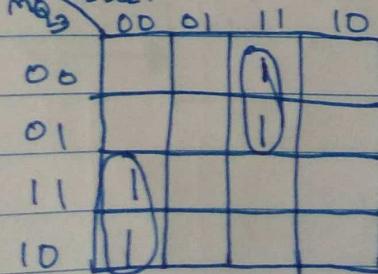
Apparatus:- Digital Board, GSP-4 Patch cord, T FF, IC 74LS 04, 74LS 86, 74LS 32

* Synchronous 3 bit updown counter

- Truth Table

M	Q_3	Q_2	Q_1	\bar{Q}_3	\bar{Q}_2	\bar{Q}_1	T_3	T_2	T_1
0	0	0	0	01	01	1	0	0	
0	0	0	1	01	1	0	0	1	1
0	0	1	0	01	0	1	0	0	1
0	0	1	1	01	0	0	1	1	1
0	1	0	0	0	01	1	0	0	1
0	1	0	1	0	1	0	0	1	1
0	1	1	0	0	0	1	0	0	1
0	1	1	1	0	0	0	1	1	1
1	0	0	0	01	01	1	1	1	1
1	0	0	1	01	1	0	0	0	1
1	0	1	0	01	0	1	0	1	1
1	0	1	1	1	0	0	0	0	1
1	1	0	0	0	01	1	1	1	1
1	1	0	1	0	1	0	0	0	1
1	1	1	0	0	0	1	0	1	1
1	1	1	1	0	0	0	0	0	1

K map

→ for T_3 m₃ = $\bar{Q}_2 Q_1$ 

$$T_3 = m \bar{Q}_2 \bar{Q}_1 + \bar{m} Q_2 Q_1$$

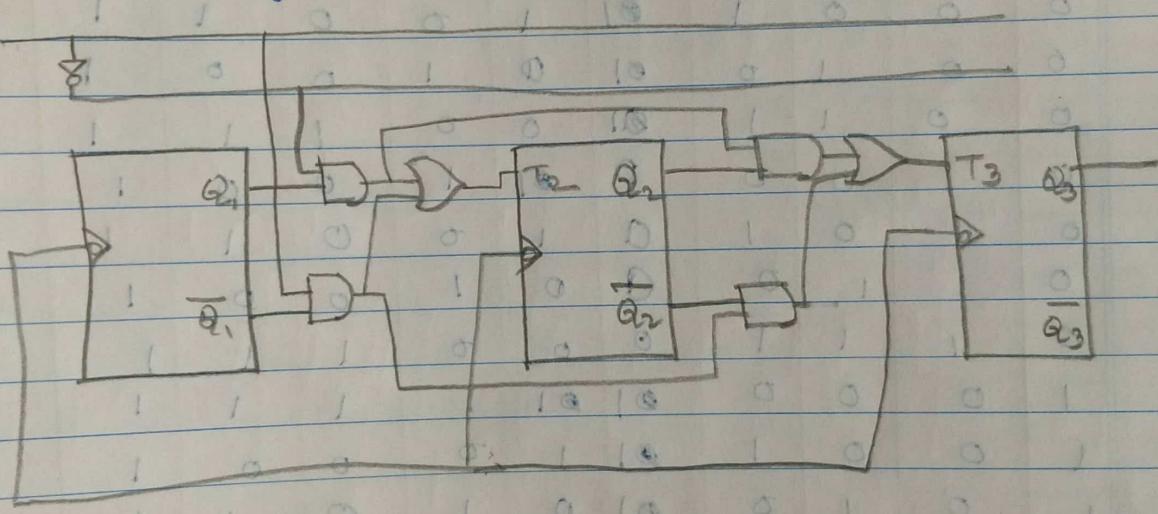
Q: on terminal

for $T_2 = \bar{Q}_2 Q_1 + Q_2 \bar{Q}_1$ enabling time is between T_1 and $\frac{T_1}{2}$

T_2	$Q_2 Q_1$	00	01	11	10	00	01	11	10
00	00	1	1	0	0	1	1	1	1
01	01	0	1	1	0	1	1	1	1
11	11	1	0	0	1	1	1	1	1
10	10	1	0	0	1	1	1	1	1

$T_2 = \bar{Q}_2 Q_1 + Q_2 \bar{Q}_1$ enabling time is between T_1 and $\frac{T_1}{2}$

Circuit Diagram



3 bit up/down counter (synchronous)

Conclusion :-

we understood working of 3 bit up/down counter

Assignment No:- 10

Title :- Mod N Counter

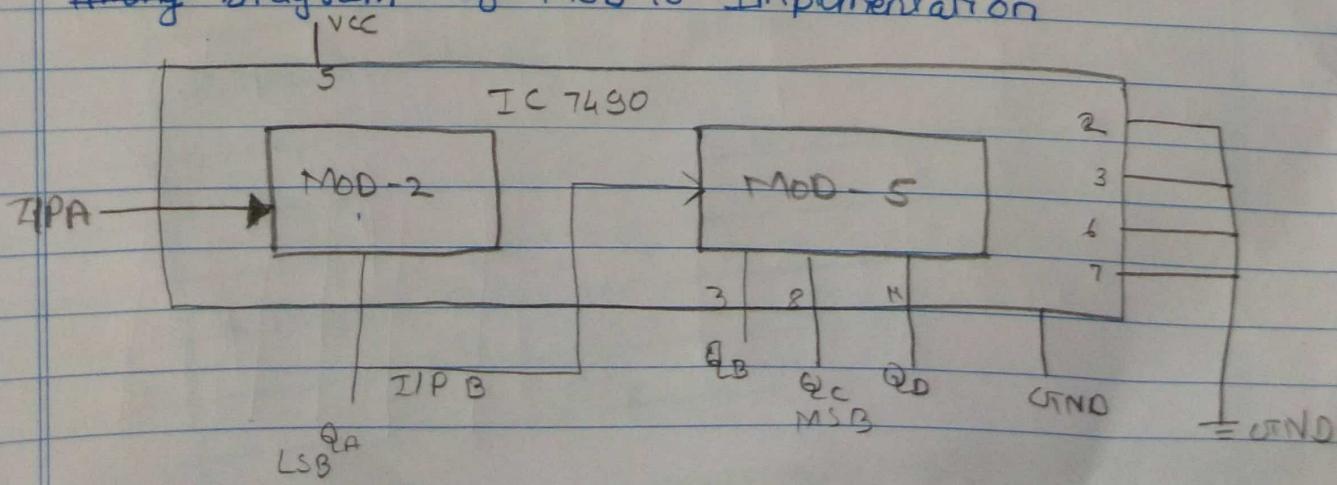
Apparatus : Digital Trainer Kit, 7490 (Decade counter),
Patch cord.

Theory :- IC - 7490 (Decade counter) is having various name like mod-n, decade, BCD. It is a ~~ECL~~ TTL MSI decade counter. It contains 4 ms FF & some gates to provide a divide by 2 counter & 3 stage binary counter which gives 135.

Truth Table

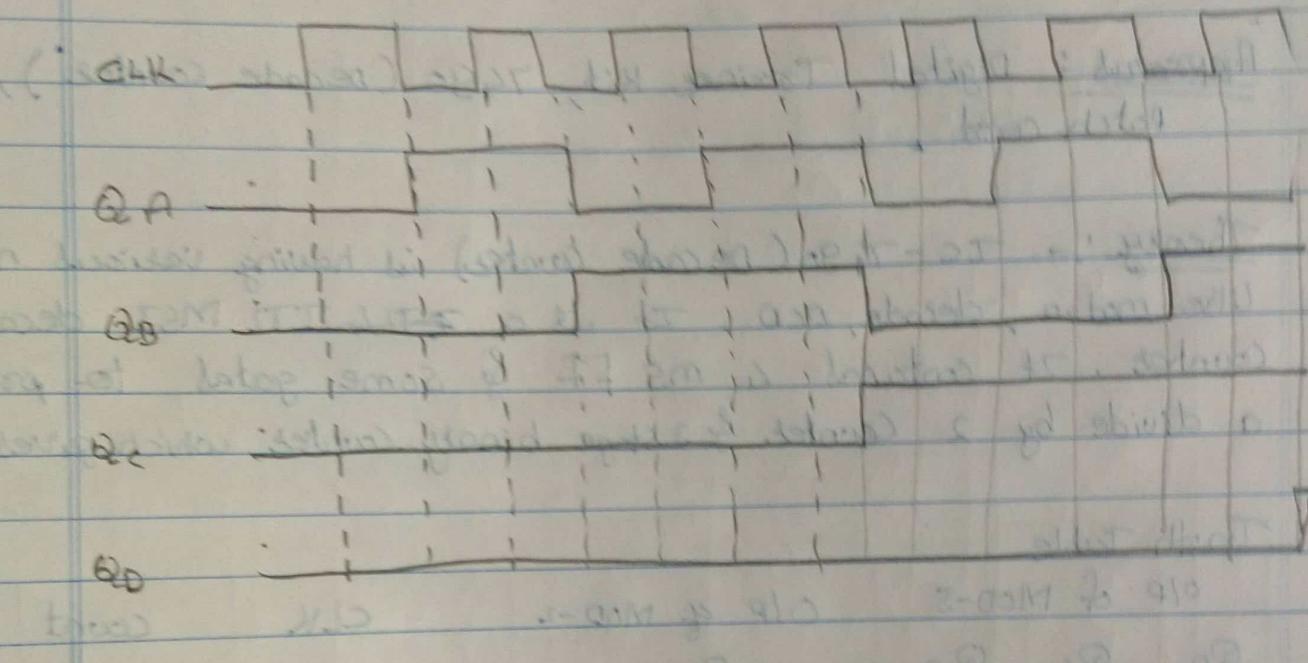
O/P of MOD-5			O/P of MOD-2	CLK	count
Q _D	Q _C	Q _B	Q _A		
0	0	0	0	6	0
0	0	0	1	1	1
0	0	1	0	2	2
0	0	1	1	3	3
0	1	0	0	4	4
0	1	0	1	5	5
0	1	1	0	6	6
0	1	1	1	7	7
1	0	0	0	8	8
1	0	0	1	9	9

Timing Diagram of MOD 10 Implementation



n=1024 Counter

Timing Diagram of MOD 1000 N-bit Counter

Conclusion

we understood how MOD-N counter works.

x	0	1	0	0
y	1	1	0	0
z	0	0	1	0
p	1	0	1	0
q	0	1	1	0
r	1	1	1	0
s	0	0	0	1
t	1	0	0	1

not standard to do in appropriate point

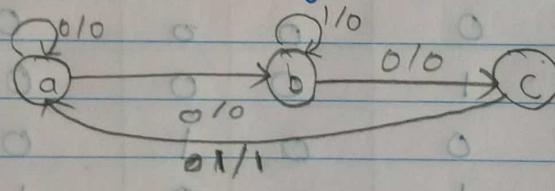
Assignment NO:-11

Title: sequence generator & checker for sequence 101.
 (Sequence Detector)

Theory:- A sequence detector is a sequential state machine that takes an input string of bits & generates an output whenever the target sequence has been detected on the present state & external input (X). Hence in diagram the output is written outside the states along with inputs.

* Design 101 sequence detector

Step 1: Develop the state diagram



Step 2: Code Assignment

X Rule 1: State having the same next state for given input condition have adjacent assignment.

(S) Rule 2: State that are next state to single state must be given adjacent assignment.

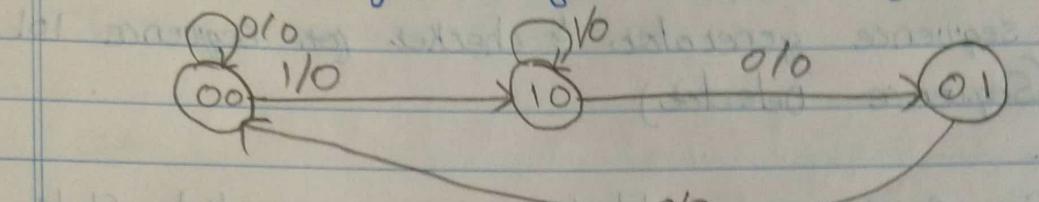
Rule 3: is given preference over Rule 2.

Previous state	States	Next state
a, c	a	a, b
b, a	b	b, c
b	c	a

K-map:

	x	0	1	
y	0	a b		
	1	c		

The state diagram after code assignment 1 is:



Step 3: Make present state/next state table, we will use diff. for design purpose.

Present state	I/P	Next state	FF Excitation	O/P		
X	Y	X'	Y'	Dx	Dy	Z
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	1
1	0	0	1	0	1	0
1	0	1	0	1	0	0
1	1	0	X	X	X	X

Step 4: Draw k-map for Dy, Dx & output Z

	xy	00	01	11	10
i	00	01	11	10	
0	0	0	X	0	
1	D	0	X	1	

	xy	00	01	11	10
i	00	01	11	10	
0	0	0	X	0	
1	0	0	X	0	

$$D_x = \bar{y} \bar{y}$$

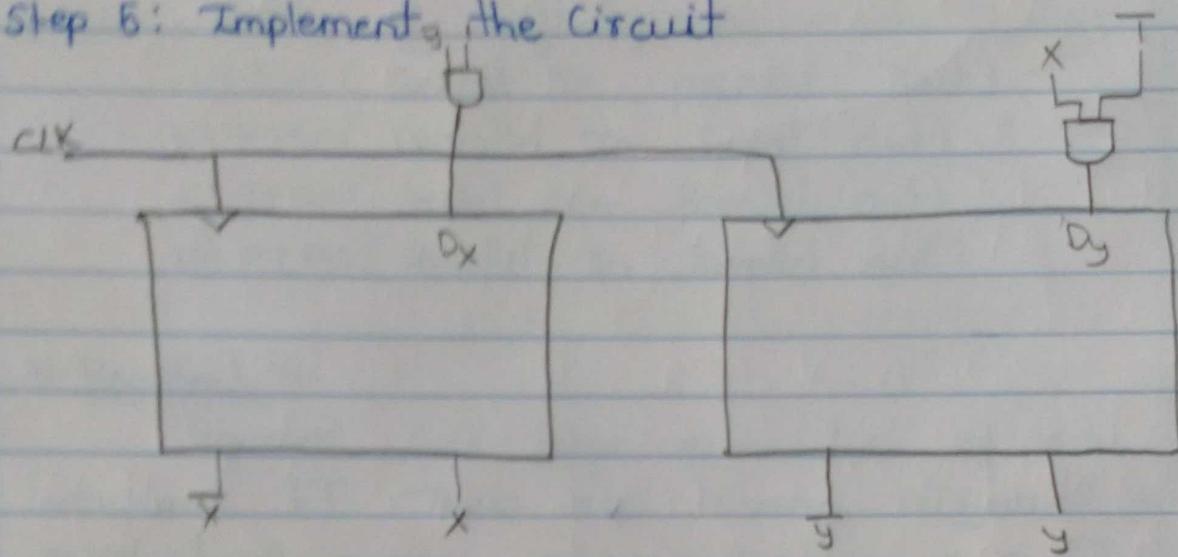
data2

$$D_y = \bar{x} \bar{y}$$

	xy	00	01	11	10
i	00	01	11	10	
0	0	0	X	0	
1	0	1	X	0	

$$Z = Y \cdot I$$

Step 5: Implementing the circuit



Conclusion: we learnt in detail about sequence detectors.

Assignment No:-12

Title : study of Shift registers

- i) SISO (Serial In parallel out)
- ii) PISO (Parallel In serial out)
- iii) SIPO (Serial In Parallel out)
- iv) PIPO (Parallel In Parallel out)

* Registers :-

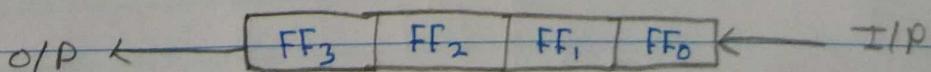
Registers hold large quantities of data than individual FF. There are many different kind of registers. A register is an extension of a FF that can store multiple bits. Registers are commonly used as temporary storage in a processor. There a binary data in a register can be moved in within the register from one of the other or outside it with application of clock pulse, such register are called shift register. These are used for data storage, transfer & contain A & L operations.

modes are :

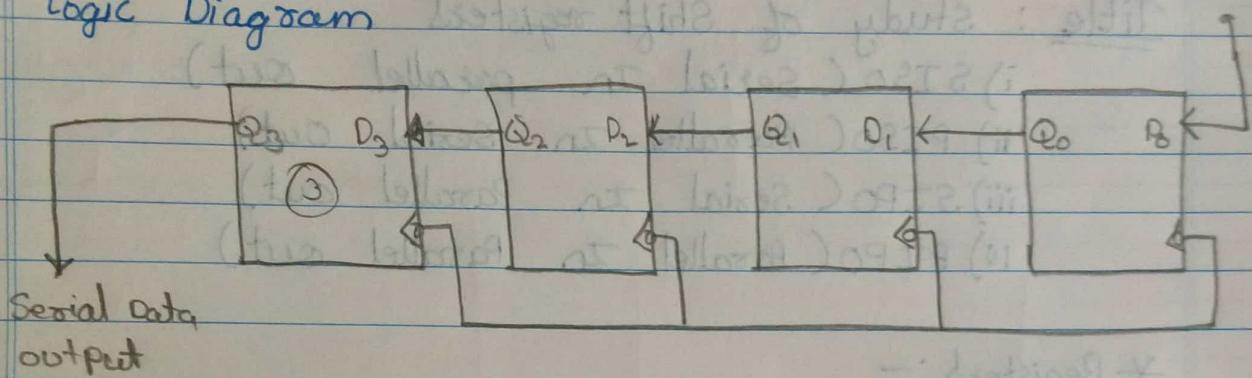
- 1) serial in parallel out.
- 2) parallel in serial out.
- 3) parallel in parallel out.
- 4) serial in parallel out.

* SISO (Shift left Mode)

bits shift to left by 1 position at every clock pulse.

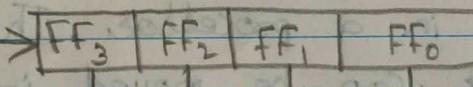
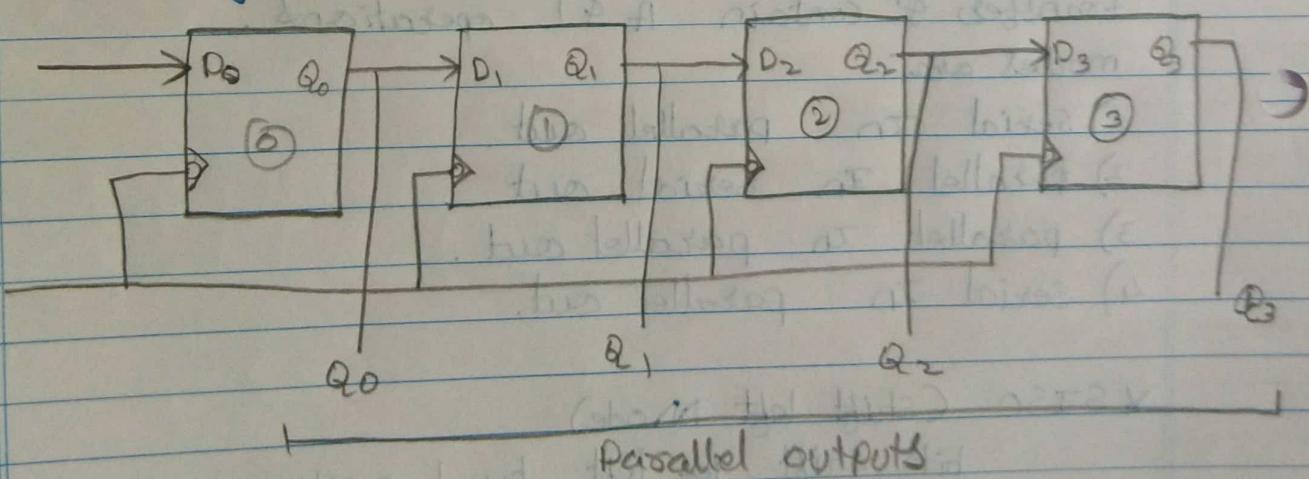


SI-PO transmitter

Logic Diagram*Serial In parallel out (SIPO)

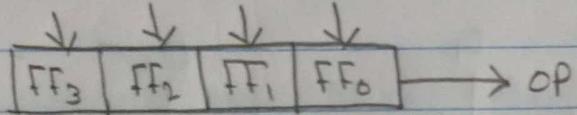
— Data is entered serially & taken out in parallel

as shown below

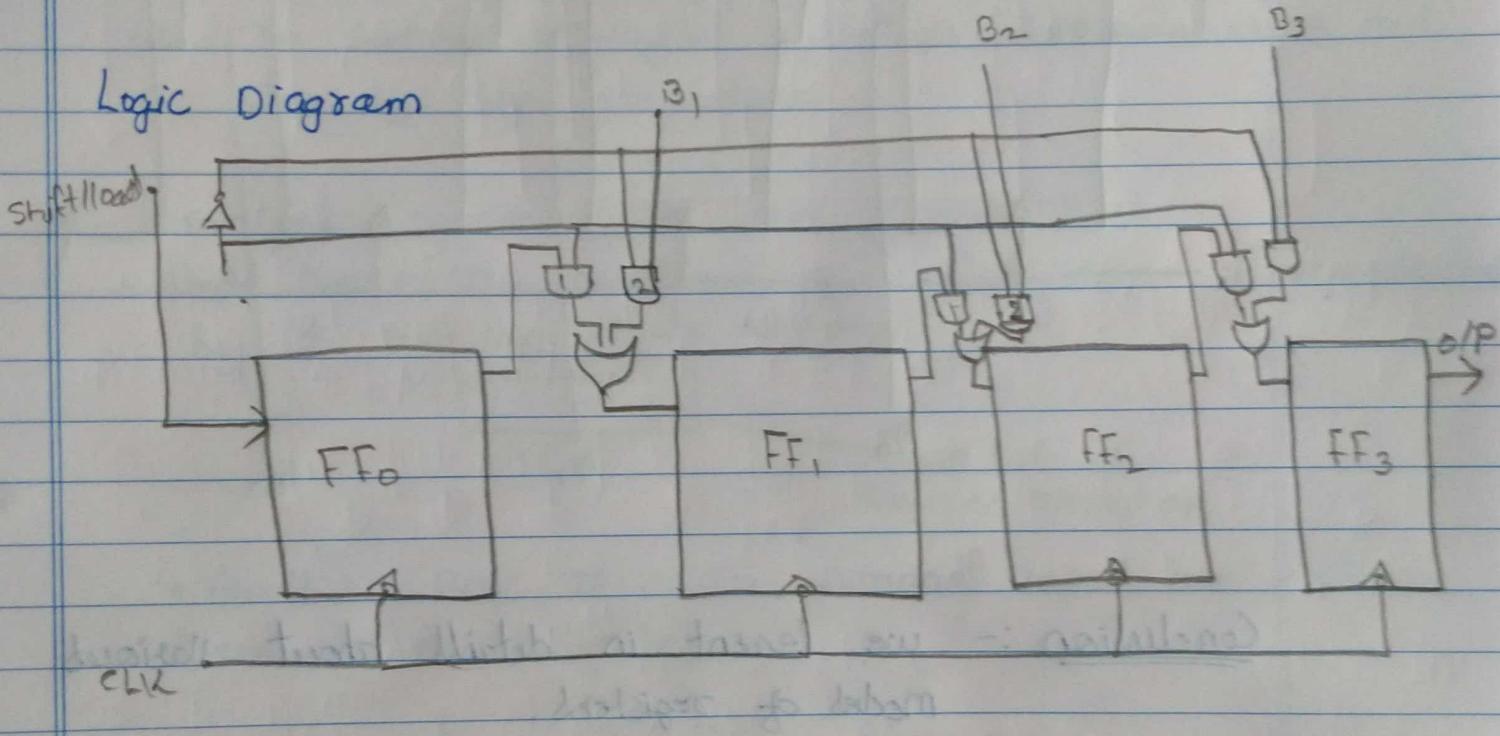
Logic Diagram

* Parallel In Serial Out (PISO)

- bits are entered in parallel & sent out serially.

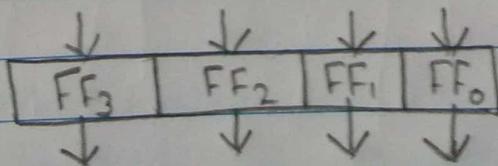


Logic Diagram

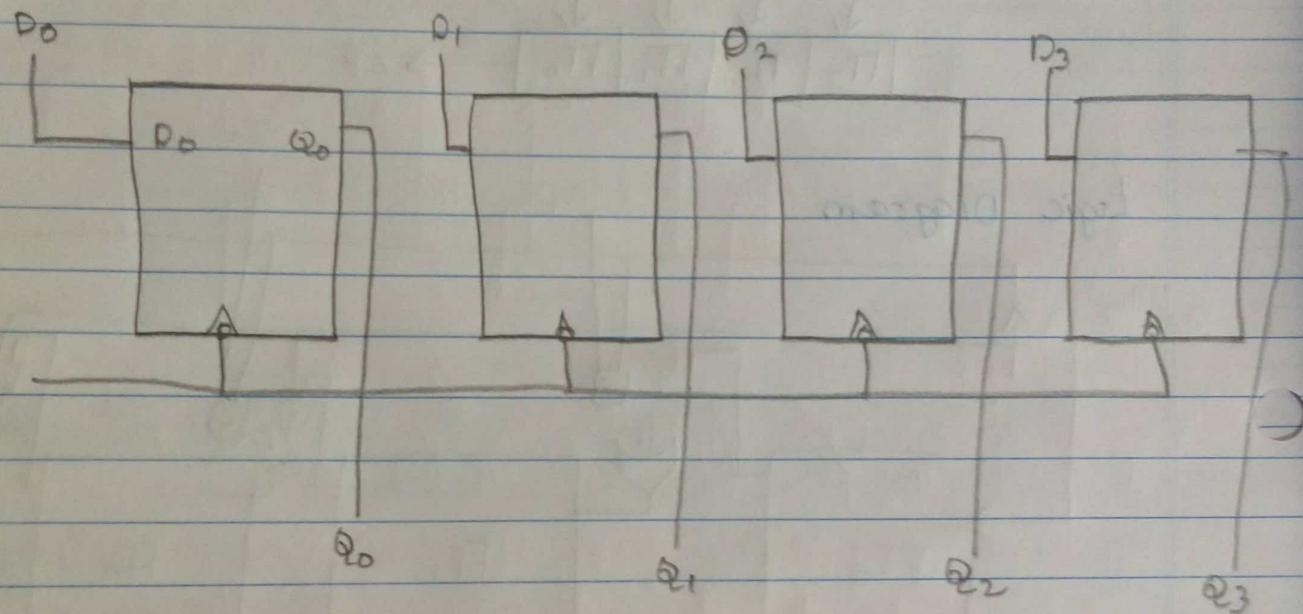


* Parallel In Parallel out (PIPO)

- All inputs are loaded simultaneously & output simultaneously. as shown below.



Logic Diagram



Conclusion :- we learnt in details about various modes of registers.

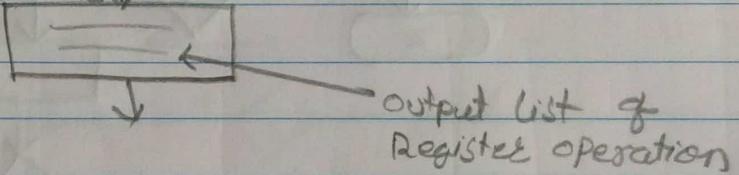
Assignment No:- 13

Title :- ASM (Algorithmic state Machine)Theory :-

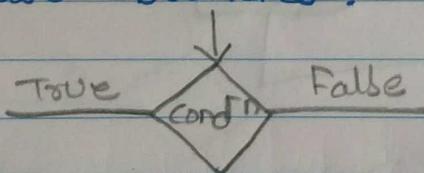
ASM stands for 'Algorithmic State Machine'. It is used to control a digital system which carries out a step by step procedure.

★ Principal Component of an ASM chart:

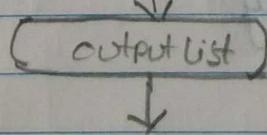
- State Box :- The state box of the system is represented by it. State Name \downarrow State assignment



- Question Box: It is a diamond shaped box with true false branches.



- Conditional output box

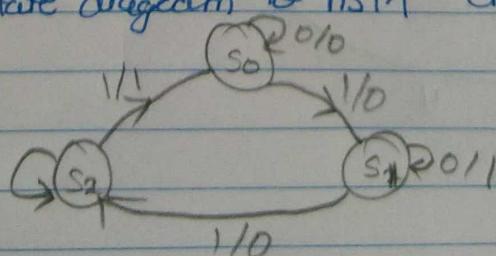


★ Mealy Machine:-

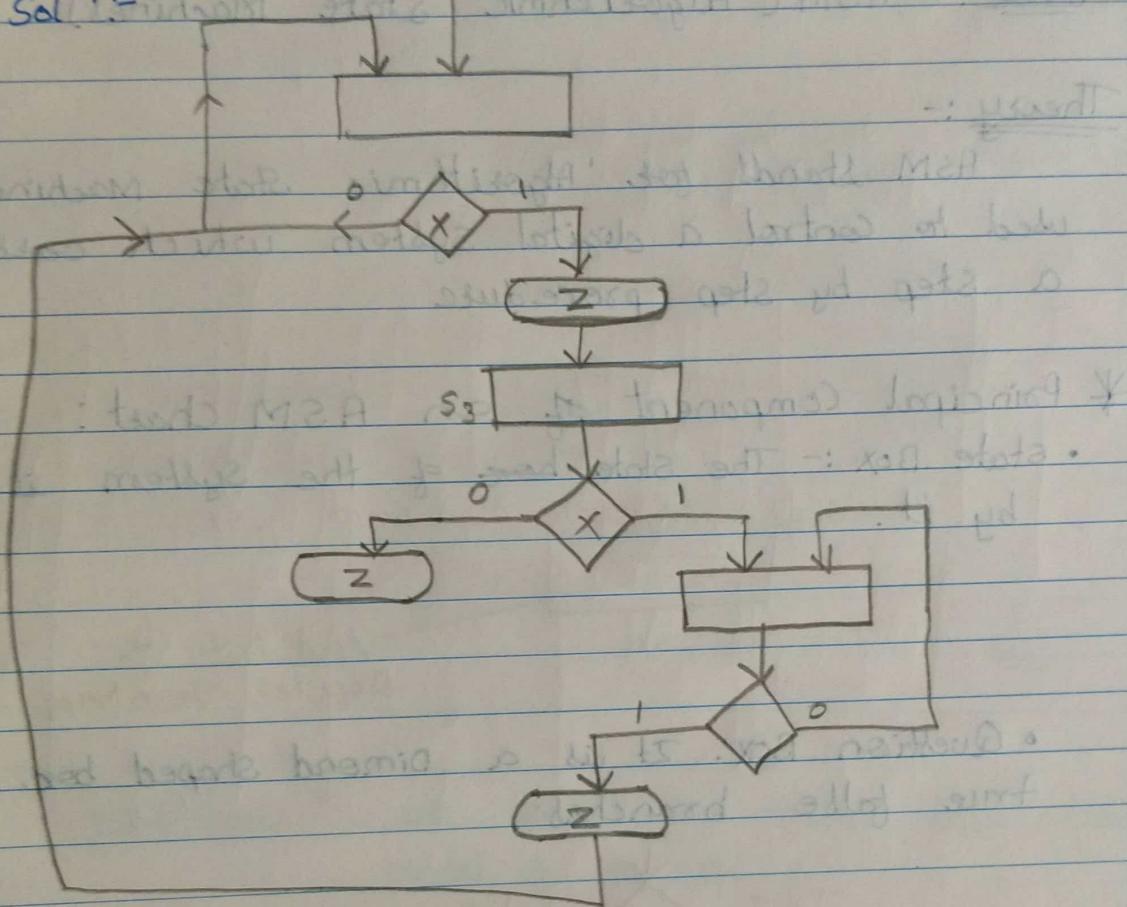
output is a function of both present state & input.

Example:-

Convert state diagram to ASM chart



Soln: random state switching M2A - ZHT

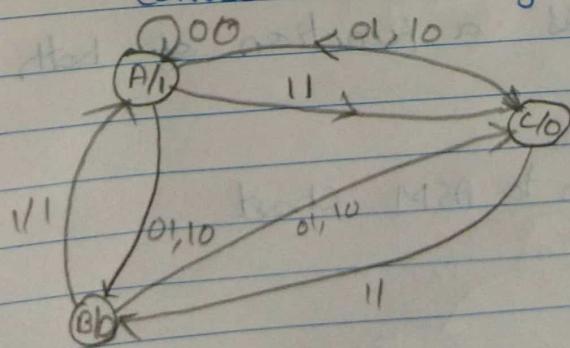


* Moore Machine:

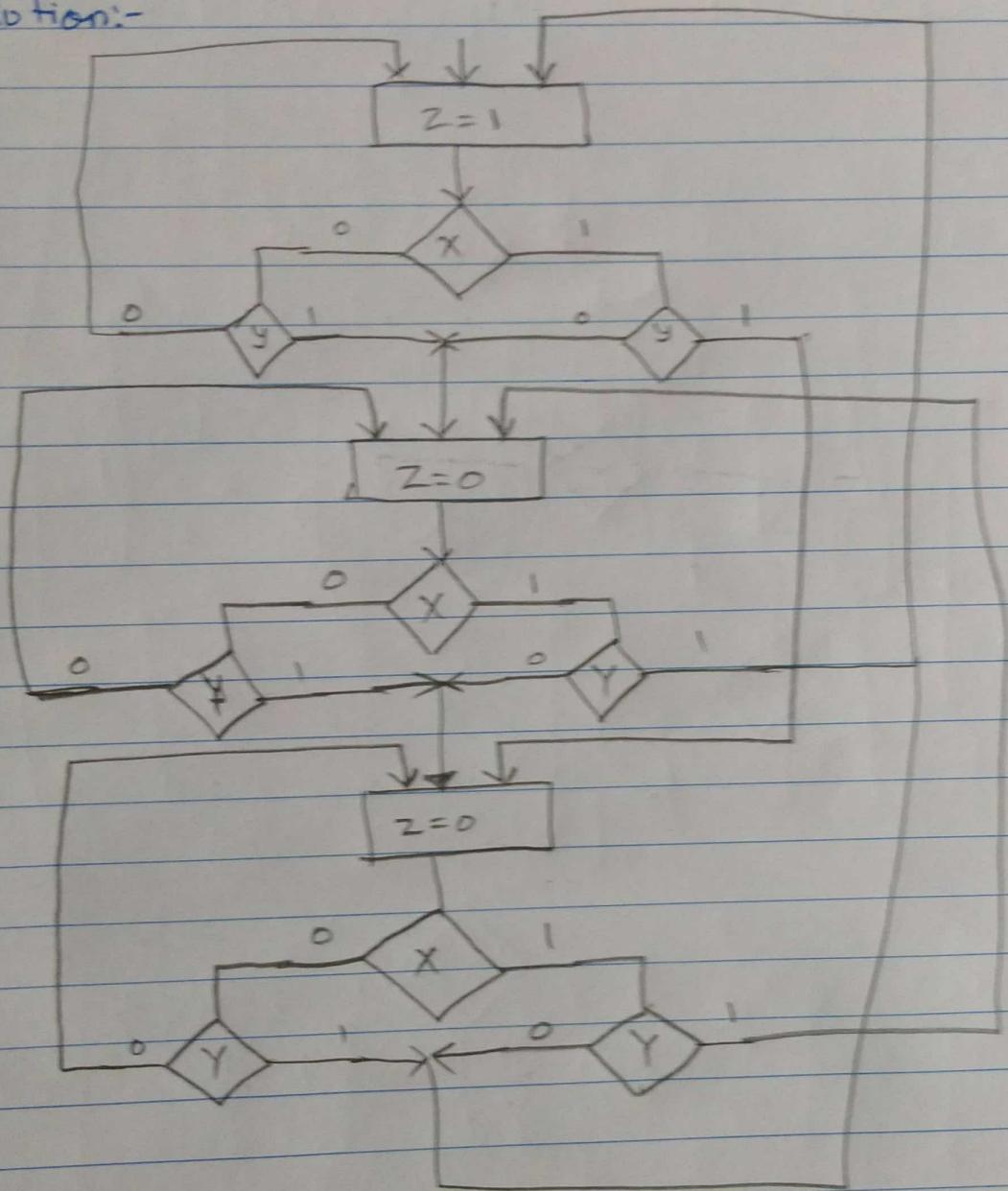
In this case, output is a function of the present state only

Example:

convert state diagram to ASM chart



Solution:-



Conclusion :- we understood about mealy & moore ASM charts.