	b)	A synchronous sequential machine has a single control input x and two outputs A & B. On consecutive falling edges of the clock, the code on A & B changes from 00 to 01 to 11 to 10 and repeats itself when $x = 1$. If at any time $x = 0$, it holds to the present state. Draw the state diagram and design the circuit using T flip flops.	L4	CO5	5 M
		OR			
11		Design a Mealy machine that produces output $Z = 1$ whenever the sequence 1101 is detected in the input stream of bits. Overlapping of the sequence is permitted. i. Write the state diagram. ii. Develop the state table. iii. Show the complete circuit realization using T flip-flops.	L5	CO5	10 M

PVP 23

Code: 23EC3302

II B.Tech - I Semester – Regular / Supplementary Examinations
NOVEMBER 2025

SWITCHING THEORY AND LOGIC DESIGN
(ELECTRONICS & COMMUNICATION ENGINEERING)

Duration: 3 hours Max. Marks: 70

Note: 1. This question paper contains two Parts A and B.
2. Part-A contains 10 short answer questions. Each Question carries 2 Marks.
3. Part-B contains 5 essay questions with an internal choice from each unit. Each Question carries 10 marks.
4. All parts of Question paper must be answered in one place.

BL – Blooms Level CO – Course Outcome

PART – A

		BL	CO
1.a)	Define Excess-3 code with an example.	L1	CO1
1.b)	What is Hamming distance?	L2	CO1
1.c)	Write the minterms of a 3-variable Boolean function.	L2	CO2
1.d)	What is the difference between canonical and standard form?	L2	CO2
1.e)	State the difference between encoder and decoder.	L1	CO4
1.f)	Write the truth table of a 2×1 multiplexer.	L2	CO4
1.g)	Write excitation table of JK flip-flop.	L1	CO4
1.h)	List the applications of shift registers.	L2	CO4
1.i)	Define Moore model.	L2	CO4
1.j)	What is state reduction in sequential circuits?	L2	CO4

PART – B

			BL	CO	Max. Marks
		UNIT-I			
2	a)	Convert $(259)_{10}$ to binary, BCD, Excess-3 and Gray code.	L3	CO1	5 M
	b)	Simplify F(A,B,C)=Σm(1,3,5,7) algebraically.	L3	CO3	5 M
		OR			
3	a)	Derive a Hamming code for 4-bit data 1011.	L3	CO1	5 M
	b)	State & prove DeMorgan's Theorems with truth tables.	L2	CO1	5 M
		UNIT-II			
4	a)	Minimize F(A,B,C,D)=Σ(0,1,2,5,7,8,9,10,14) using K-map.	L3	CO3	5 M
	b)	Implement the above minimized function using NAND-NAND gates.	L4	CO2	5 M
		OR			
5	a)	Simplify a 5-variable function F(A,B,C,D,E)=Σ(1,3,7,9,11,15,19,23,27,31) using K-map.	L3	CO3	5 M
	b)	Discuss the role of don't-care conditions in K-map with example.	L2	CO3	5 M
		UNIT-III			
6	a)	Design a 4-bit binary parallel adder.	L4	CO5	5 M
	b)	Implement F(A,B,C)=Σ(1,2,4,6) using 4x1 MUX.	L4	CO4	5 M

		OR			
7	a)	Design a full subtractor using two half subtractors.	L4	CO4	5 M
	b)	Realize the function F(A,B,C)=Σ(0,2,5,7) using decoder.	L4	CO4	5 M
		UNIT-IV			
8	a)	Convert JK flip-flop into D flip-flop.	L3	CO4	5 M
	b)	Design a 3-bit ripple counter using T- flip-flops.	L5	CO5	5 M
		OR			
9	a)	Design a synchronous MOD-6 counter using JK- flip-flops.	L3	CO4	5 M
	b)	Explain the operation of 4-bit ring counter.	L2	CO4	5 M
		UNIT-V			
10	a)	Convert the following Moore machine to mealy machine.	L3	CO5	5 M

PS	NS		O/P
	x=0	x=1	Z
A	D	B	1
B	A	E	0
C	A	E	1
D	C	A	0
E	F	D	0
F	F	D	1

Switching Theory and Logic Design

Scheme of Evaluation

November 2025

1) a) Definition — 1M
Example — 1M

1) b) Definition — 2M

1) c) Minterms — 2M

1) d) Difference — 2M (Canonical - 1M, Standard form - 1M)

1) e) Encoder — 1M
Decoder — 1M

1) f) Truth table — 2M

1) g) Excitation table — 2M

1) h) Any two applications — 2M

1) i) Definition of Moore Model — 2M

1) j) Purpose of State Reduction — 2M

2) a) Binary Representation — 2M
BCD code — 1M
Excess-3 code — 1M
Gray code — 1M

2 b) ~~K-Map & Plotting~~ Minterms of given function in a K-Map — 2M
~~Formulas~~ Simplification — 2M
Simplified expression — 1M

3) a) Formulas for parity bits P_1, P_2, P_4 — 3M
Determination of Hamming code — 2M

b) Statement — 1M
Truth table of I law — 2M
Truth table of II law — 2M

4) a) K-Map & plotting 1's in a K-Map – 2M
Combining adjacent 1's in a Map – 1M
simplified expression – 2M

b) NAND-NAND implementation – 5M

5) a) 5-variable K-Map – 2M
Plotting 1's in Map – 1M
Combining adjacent 1's – 1M
simplified expression – 1M

b) Role of Don't cares – 2M
example – 3M

6) a) Logic of A+B – 2M
Design – 2M
Logic diagram – 1M

b) Implementation Table of MUX – 2M
Determination of I_0, I_1, I_2, I_3 – 2M
Logic diagram – 1M

7) a) Truth table of Full Subtractor – 2M
expressions for D & B outputs – 2M
Logic diagram – 1M

b) Size of the decoder – 2M
Logic diagram – 3M

8) a) Conversion table – 2M
simplified expressions – 2M
Logic diagram – 1M

b) Counter states – 2M
Design – 3M

9) a) Counter states – 2M
Excitation table – 1M
Simplified expressions for FF inputs – 1M
Logic diagram – 1M

b) Logic diagram of Ring Counter – 3M
Counter states – 1M
Explanation – 1M

10) a) Mealy Conversion table – 2M
State Reduction – 2M
State table of Mealy Model – 1M

b) State diagram – 1M
State table – 1M
Excitation table – 2M
Simplified expressions & Logic diagram – 1M

11) State diagram – 3M
State table – 2M
Excitation table – 2M
Simplified expressions for FF I/p's – 2M
Logic diagram – 1M

——— X ———

Switching Theory and Logic Design
Scheme of Evaluation
A.Y: 2025-26

Part-A

1) a) Excess-3 Code is obtained by adding '3' to BCD code word

Example:

Decimal Digit	BCD Code	Excess-3 code
3	0011	0110
9	1001	1100

— 2M

1) b) Hamming distance is defined as the number of bit changes between any two code words.

For ex: Hamming distance of 1000 and 1111 is '3'.

Similarly Hamming distance between 1100 and 0100 is '1'.

— 2M

1) c) Minterms of a three variable Boolean function are $x'y'z'$, $x'y'z$, $x'yz'$, $x'yz$, $xy'z'$, $xy'z$, xyz' and xyz.

— 2M

1) d) A function is said to be in Canonical SOP form iff all the product terms are minterms.

Whereas the standard form of SOP is the function is expressed as sum of product terms in which each product term may not be minterm.

— 2M

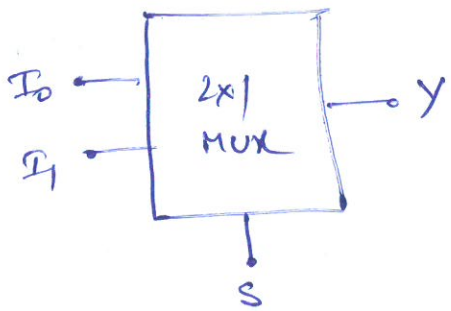
Similarly, a function is said to be in Canonical POS form iff all the sum terms are maxterms.

Otherwise standard form of POS means, the function is expressed as sum of product terms in which each product term may not be a sumterm. — (2M)

1) e) Decoder is a combinational logic circuit that converts binary information from 'n' input lines to maximum of '2^n' output lines.

Encoder ~~prof~~ performs the reverse operation of decoder. It has 2^n ip lines and 'n' output lines. — (2M)

1) f) Truth table of 2x1 Multiplexer.



Block diagram of 2x1 Mux

S	Y
0	I0
1	I1

Truth table of 2x1 Mux.

— (2M)

1) g) Excitation table of JK flip flop:-

PS	NS	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

— (2M)

1) h) Applications of Shift Registers

* Delay line
* Serial to Parallel Converter
* Parallel to Serial Converter
* Ring Counters

— (2M)

1) i) In Moore Model, next state is the function of present state and ~~outputs~~ inputs. where as output is the function of present state only.

Next State = f { Present state, inputs }

o/p = f { Present State }

— (2M)

1) j) State Reduction in sequential circuits is used to reduce the number of states. Reducing the number of state in a sequential circuit reduces the number of flip-flops required in a circuit. Therefore the complexity of the system gets reduced by using state reduction techniques.

— (2M)

Part-B

2) a) $(259)_{10} = (?)_2$

2	259	
2	129	— 1
2	64	— 1
2	32	— 0
2	16	— 0
2	8	— 0
2	4	— 0
2	2	— 0
2	1	— 0
	0	— 1

$\therefore (259)_{10} = (100000011)_2$

— (2M)

BCD code for Decimal 259 is

0010 0101 1001 — (1M)

Excess-3 code for Decimal 259 is

0101 1000 1100 — (1M)

Gray code for Decimal 259 is

$(10000011)_2 = \underline{110000010}$ — (1M)

2) b) $f(A,B,C) = \Sigma m(1,3,5,7)$

$f = A'B'C + A'BC + AB'C + ABC$ — (2M)

$= B'C(A'+A) + BC(A'+A)$

$\because A + A' = 1$

$= B'C(1) + BC(1)$

$\therefore A \cdot 1 = A$

$= B'C + BC$

$= C[B'+B]$

$\because B + B' = 1$

$= C[1]$

$= C$ — (2M)

$\therefore$ Simplified expression of f is

$\boxed{f(A,B,C) = C}$ — (1M)

3) a) Hamming code for 4-bit Binary data 1011 is

Bit Positions:	1	2	3	4	5	6	7
Data :	P_1	P_2	1	P_4	0	1	1

* Parity Bits

P_1 = Even parity of bits (3, 5, 7)

P_2 = Even parity of bits (3, 6, 7)

P_4 = Even parity of bits (5, 6, 7) — (3M)

$\therefore P_1$ = Even parity or XOR of (1, 0, 1) = 0

P_2 = Even parity or XOR of (1, 1, 1) = 1

P_4 = Even parity or XOR of (0, 1, 1) = 0

$\therefore$ 7-bit Hamming code for the data 1011 is

0 1 1 0 0 1 1 — (2M)

3) b) DeMorgan's theorems :-

i) $(x+y)' = x' \cdot y'$

ii) $(x \cdot y)' = x' + y'$

DeMorgan's first law states that, complement of sum of two or more variables is equivalent to the product of their individual complements.

DeMorgan's second law states that, complement of product of two or more variables is equivalent to sum of their individual complements. — (1M)

I law :- $(x+y)' = x' \cdot y'$

Truth table

x	y	x'	y'	$(x+y)'$	$x' \cdot y'$
0	0	1	1	1	1
0	1	1	0	0	0
1	0	0	1	0	0
1	1	0	0	0	0

$\therefore (x+y)' = x' \cdot y'$ — (2M)

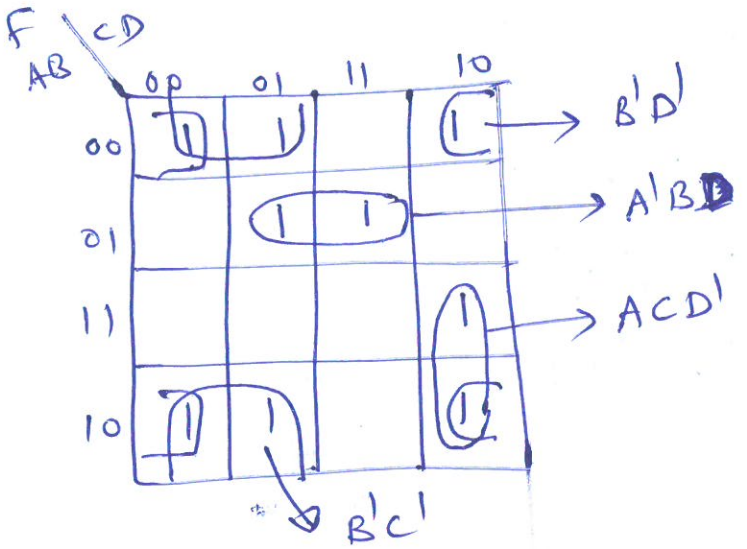
II law: $(x \cdot y)' = x' + y'$

Truth table:

x	y	x'	y'	$(x \cdot y)'$	$x' + y'$
0	0	1	1	1	1
0	1	1	0	1	1
1	0	0	1	1	1
1	1	0	0	0	0

$\therefore (x \cdot y)' = x' + y'$ — (2M)

4) a) $f(A, B, C, D) = \Sigma m(0, 1, 2, 5, 7, 8, 9, 10, 14)$



— (3M)

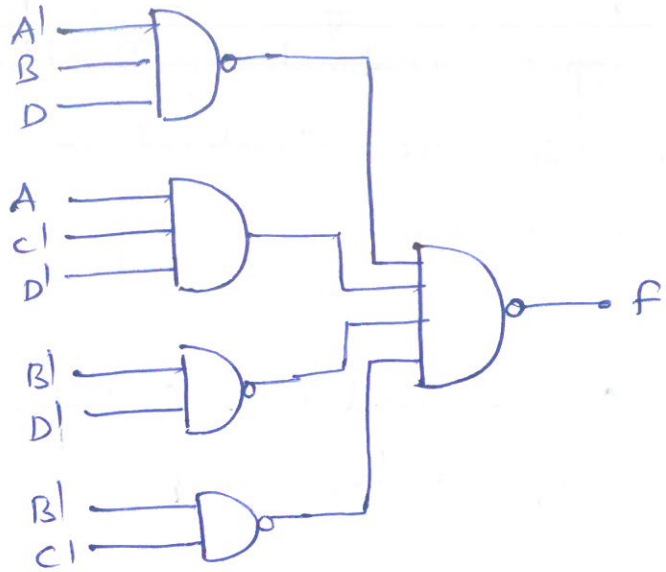
$\therefore$ Simplified expression of F using K-Map is

$$f(A, B, C, D) = A'BD + ACD' + B'D' + B'C'$$ — (2) M

4) b) Implementation of the simplified expression of 'F' in 4) a) is

$$f(A, B, C, D) = A'BD + AC'D' + B'D + B'C'$$

NAND-NAND implementation of F, assuming that complements are available is



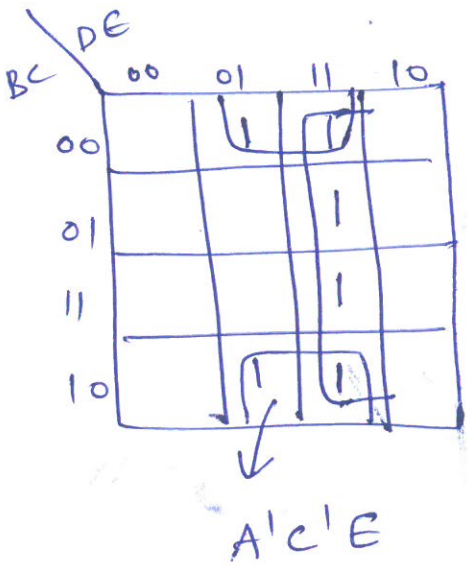
— (5M)

NAND-NAND implementation of simplified form of F.

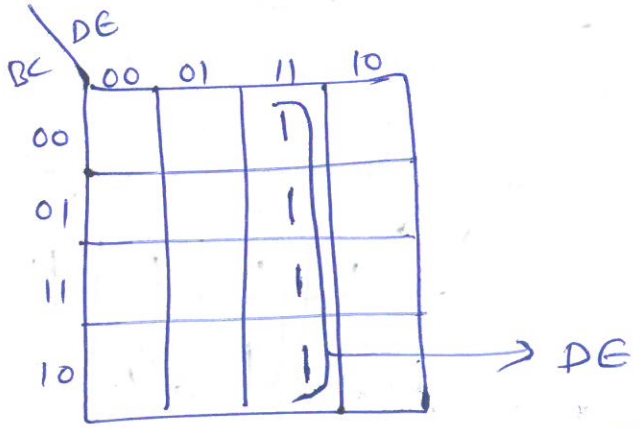
5) a) $F(A, B, C, D, E) = \Sigma(1, 3, 7, 9, 11, 15, 19, 23, 27, 31)$

F

A = 0



A = 1



— (4M)

∴ Simplified form of given 5-variable function is

$$F = A'C'E + DE$$

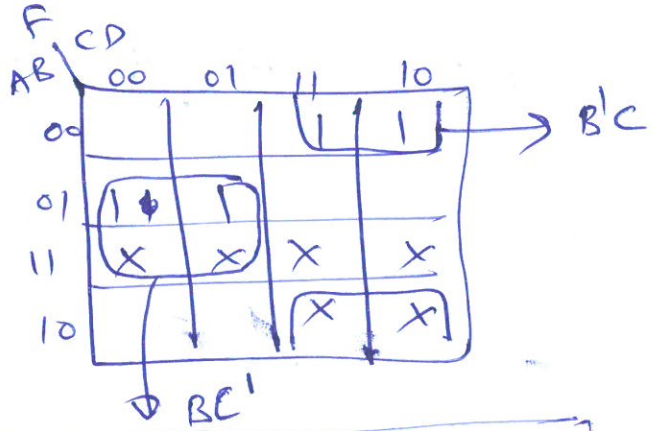
— (1M)

5) b) Sometimes the functions may have unspecified outputs for some input combinations. So the unspecified minterms of the function are considered as Don't care conditions. Don't care conditions are represented with a letter 'd' or 'x'.

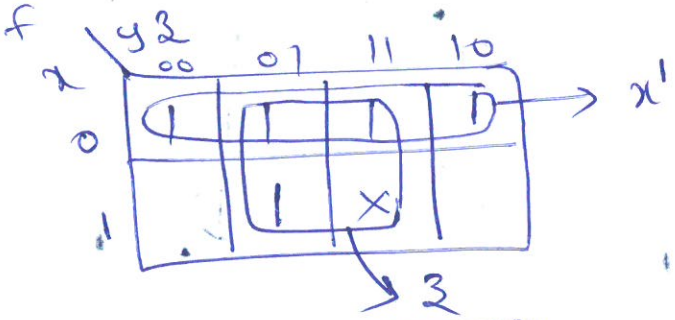
Don't care conditions can be used in a K-Map to provide further simplification of the Boolean function.

(2M)

For eg:



$\therefore f(A,B,C,D) = B'C + BC'$



$\therefore f(x,y,z) = x' + z$

(3M)

6) a) **4-bit Parallel Binary Adder :-**

Parallel Binary adder is a combinational logic circuit that performs the arithmetic sum of two binary numbers in a parallel fashion.

Let

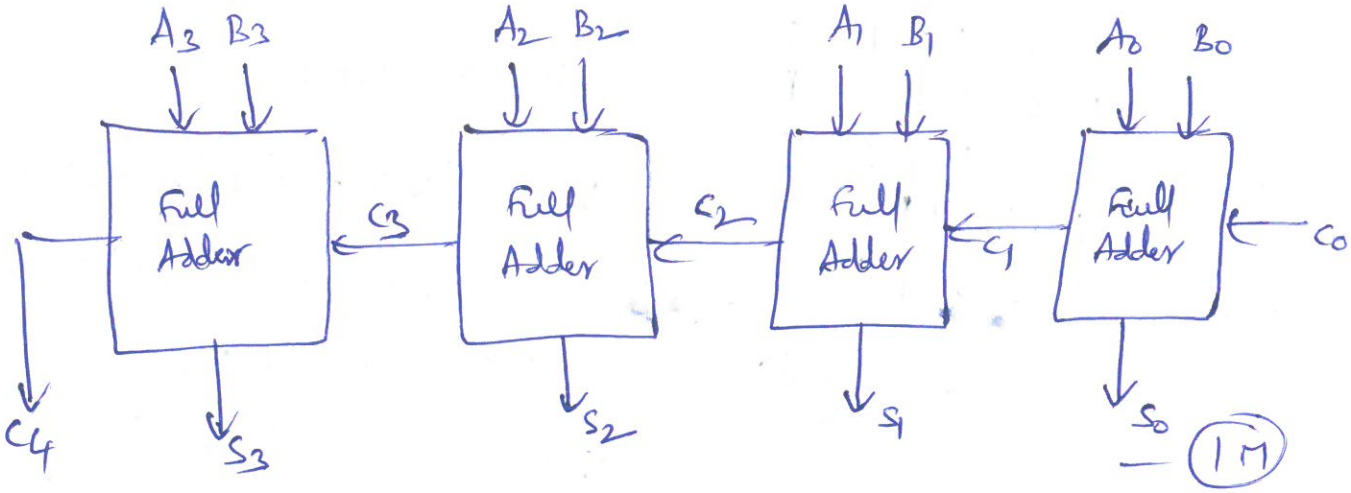
$A \rightarrow A_3\ A_2\ A_1\ A_0$

$B \rightarrow B_3\ B_2\ B_1\ B_0$ — (2M)

Sum of two 4-bit numbers is

		C_4	C_3	C_2	C_1 C_0
A →			A_3	A_2	A_1 A_0
B →	+		B_3	B_2	B_1 B_0
Sum →		C_4	S_3	S_2	S_1 S_0

— (2M)



— (1M)

Logic diagram of 4-bit Parallel Binary Adder.

6) b) Given $f(A, B, C) = \Sigma(1, 2, 4, 6)$

Implementation of f using 4x1 Mux:-

Let A be the data ip of MUX, B & C are select I/P's of Mux.

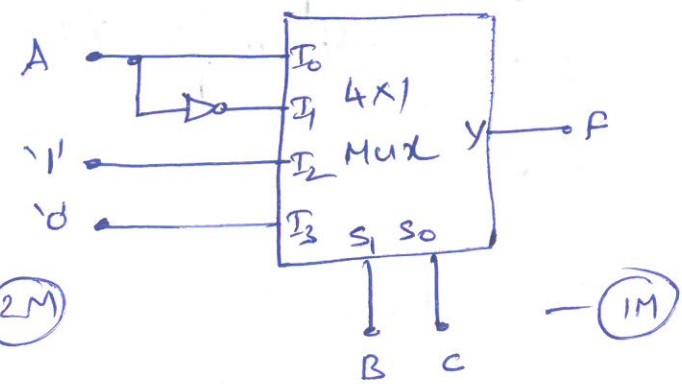
Implementation table:

	I_0	I_1	I_2	I_3
A'	0	(1)	(2)	3
A	(4)	5	(6)	7
	A	A'	1	0

— (2M)

$\therefore$ $I_0 = A$; $I_1 = A'$

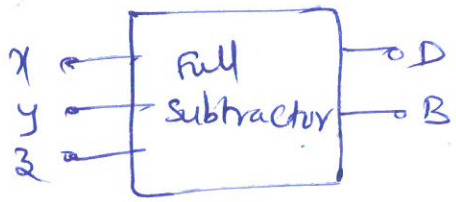
$I_2 = 1$; $I_3 = 0$ — (2M)



— (1M)

Implementation of given function 'f' using 4x1 Mux.

7) a) Full subtractor using half subtractors.



Truth table :-

x	y	z	B	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

— (2M)

$D = x'y'z + x'yz' + xy'z' + xyz$

$= z(x'y' + xy) + z'(x'y + xy')$

$= z(x \oplus y)' + z'(x \oplus y)$

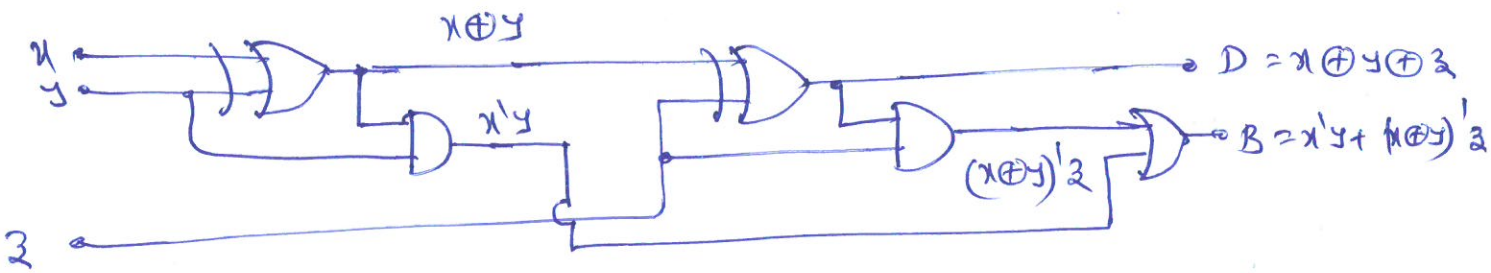
$\boxed{D = x \oplus y \oplus z}$

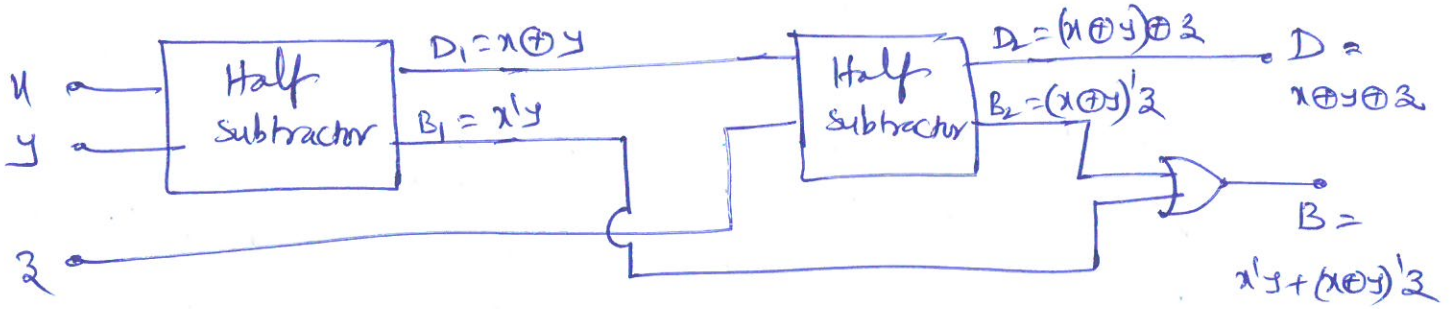
$B = x'y'z + x'yz' + x'yz + xyz$

$= z(x'y' + xy) + x'y(z' + z)$

$= z(x \oplus y)' + x'y$

$\boxed{\therefore B = (x \oplus y)'z + x'y}$ — (2M)





— (1M)

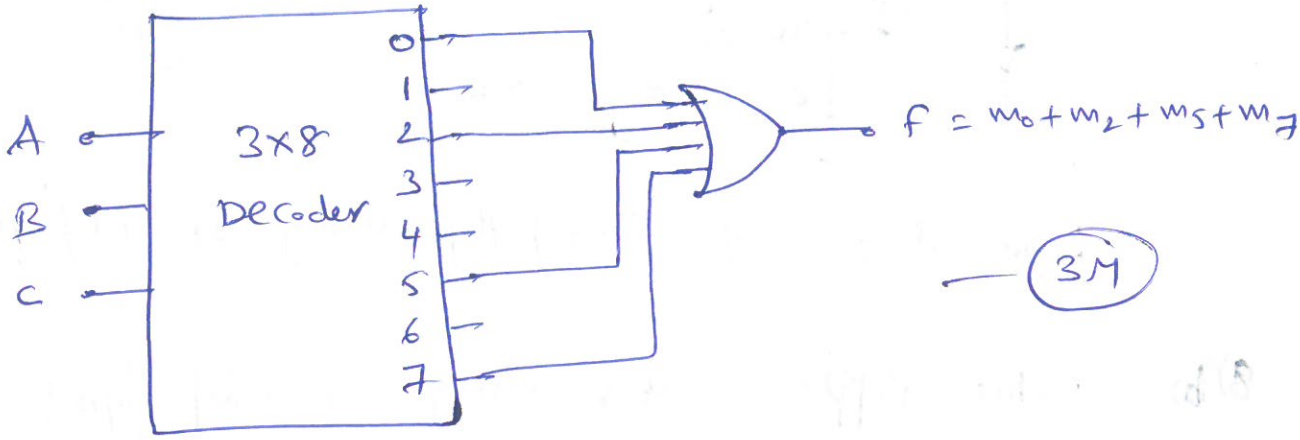
Logic Diagram of Full Subtractor using two Half Subtractors & OR gate.

7) b) $f(A,B,C) = \Sigma(0,2,5,7)$

Implementation of given function using Decoder :-

Size of the decoder required is 3×8. — (2M)

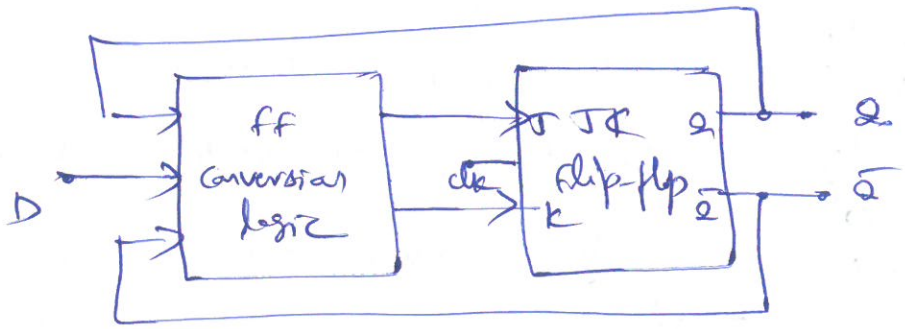
$f = m_0 + m_2 + m_5 + m_7$



— (3M)

Implementation of given function using Decoder

8) a) Conversion of JK flip-flop to D flip-flop



Truth Table of Conversion Logic :-

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

— (2M)

Excitation table of JK FF :-

PS	NS	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

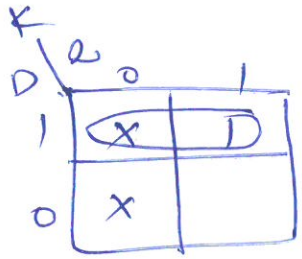
J

D \ Q_n	0	1
0		X
1	1	X

J = D

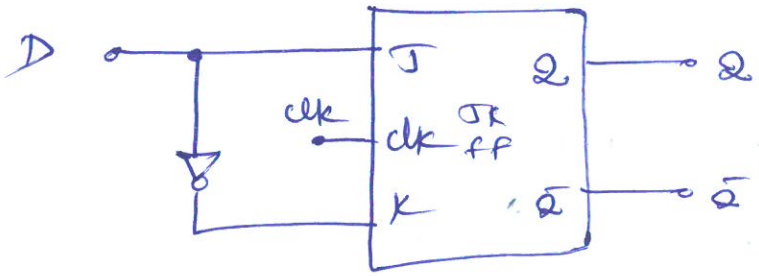
K

D \ Q_n	0	1
0	X	
1		X



$\therefore K = D$

— (2M)



— (1M)

Logic diagram of D flip-flop using JK flip-flop.

8) b) 3-bit Ripple Counter using T flip-flops :-

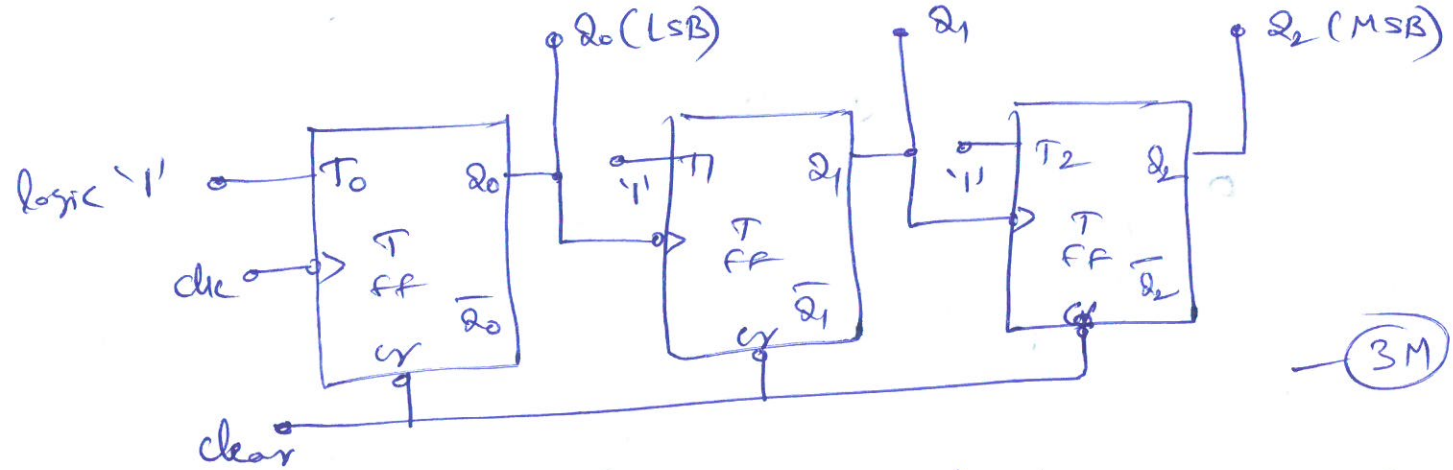
Q_2	Q_1	Q_0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Q_1 changes its state, whenever Q_0 changes from 1 to 0.

Q_2 changes its state, whenever Q_1 changes from 1 to 0.

Q_0 changes its state whenever clk changes from 1 to 0.

— (2M)



— (3M)

Logic diagram of 3-bit Ripple (UP) Counter using T flip-flops.

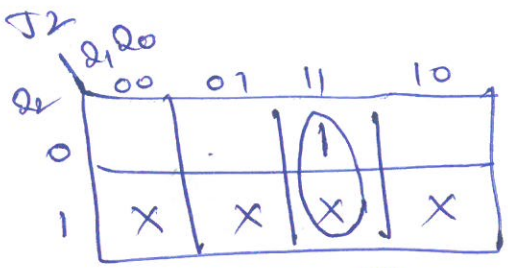
9) a) Synchronous Mod-6 Counter using JK flipflops.

No. of flipflops required = 3

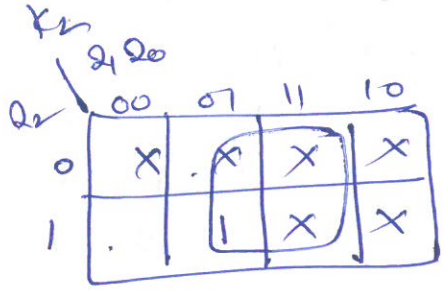
PS $Q_2Q_1Q_0$	NS $Q_2Q_1Q_0$	Flipflop Inputs J_2 K_2	J_1 K_1	J_0 K_0
0 0 0	0 0 1	0 X	0 X	1 X
0 0 1	0 1 0	0 X	1 X	X 1
0 1 0	0 1 1	0 X	X 0	1 X
0 1 1	1 0 0	1 X	X 1	X 1
1 0 0	1 0 1	X 0	0 X	1 X
1 0 1	0 0 0	X 1	0 X	X 1

3M

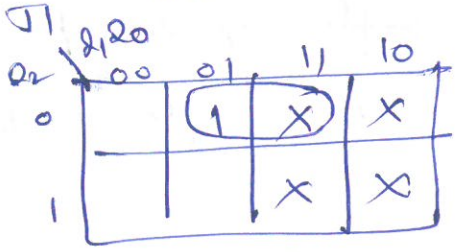
110 & 111 are unused ip combinations.



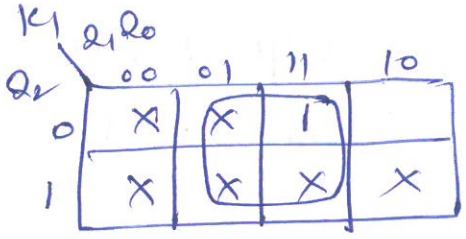
$J_2 = Q_1Q_0$



$K_2 = Q_0$



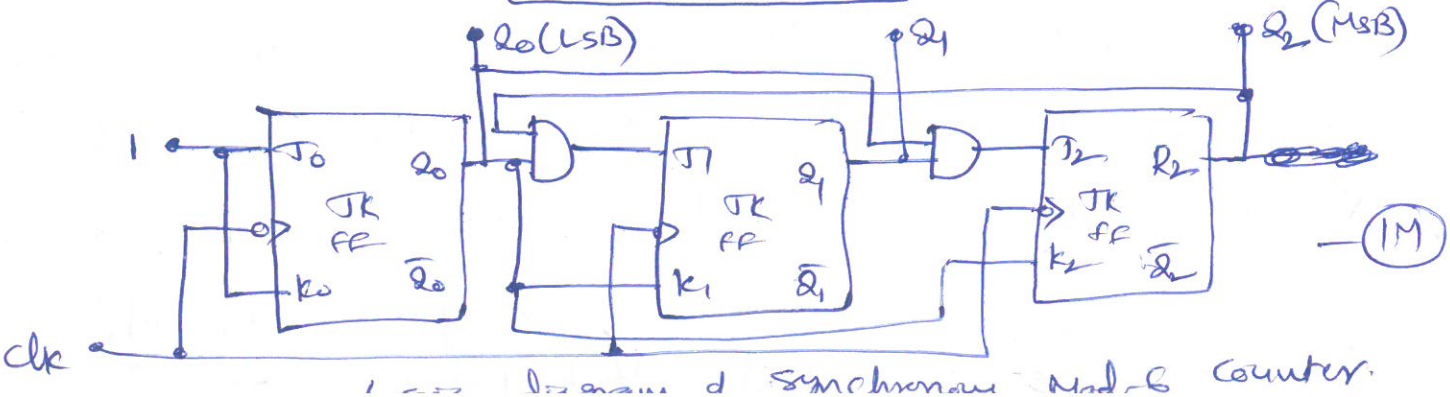
$J_1 = Q_2Q_0$



$K_1 = Q_0$

1M

$J_0 = 1$; $K_0 = 1$

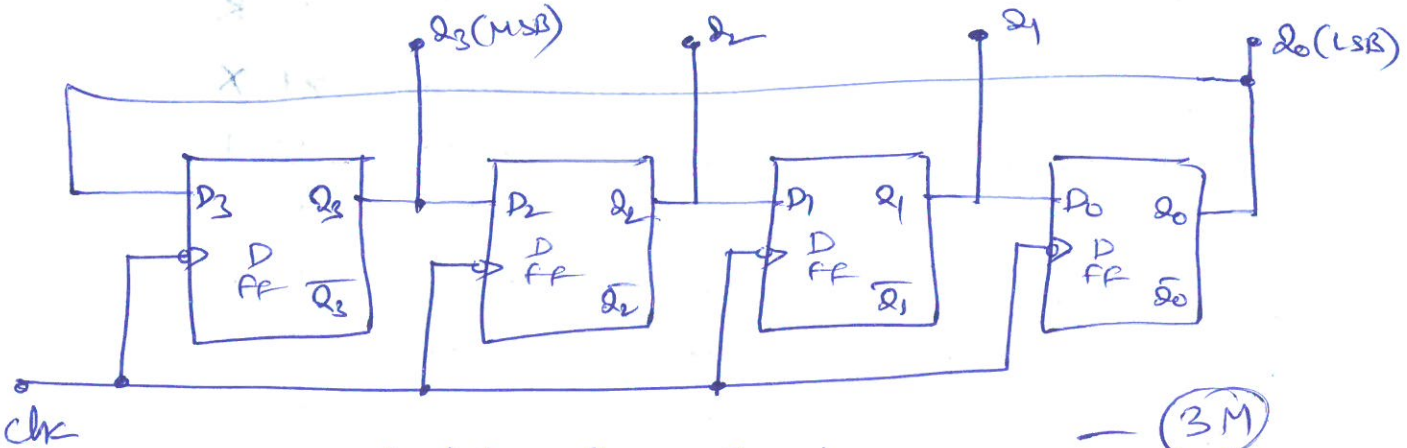


1M

Diagram of Synchronous Mod-6 Counter.

9) b) 4-bit Ring Counter :-

4-bit Ring Counter is obtained by connecting the output of the last flip-flop as the input of first flip-flop in a serial-in serial-out Shift Register.



— (3M)

4-bit Ring Counter.

Q_3	Q_2	Q_1	Q_0
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

— (1M)

Counter states in the 4-bit Ring Counter are shown above.

Since it is a shift Register it is ~~capto~~ capable of shifting the data towards right direction after each clock pulse.

Initially the counter is loaded with 1000 using preset and clear inputs. Then after the application of clock pulse, the '1' will be circulate among the four flip-flops. Since this counter has 4 distinct counter states, it is called as Mod-4 Counter. The o/p's of Ring counter can be used in stepper Motor Applications because it has sequential clock pulse — (1M)

10). a) (5)

PS	NS		O/P
	x=0	x=1	z
A	D	B	1
B	A	E	0
C	A	E	1
D	C	A	0
E	f	D	0
f	f	D	1

Mealy Model :-

PS	NS		O/P (z)	
	x=0	x=1	x=0	x=1
A	D	B	0	0
B	A	E	1	0
~~C~~	~~A~~	~~E~~	~~1~~	~~0~~
D	~~C~~ B	A	1	1
E	~~f~~ E	D	1	0
~~f~~	~~f~~	D	~~1~~	~~0~~

— (3M)

States B & C are equivalent states. Therefore 'C' can be replaced with state B.

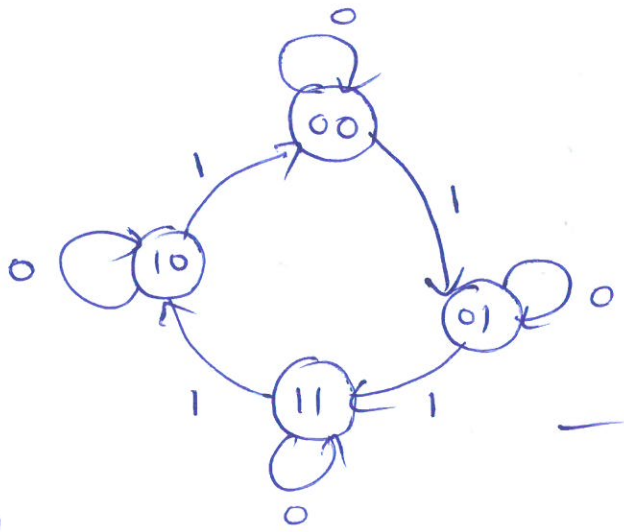
Similarly states E & f are equivalent, state 'f' can be replaced with E.

∴ Reduced state table for Mealy Model is

PS	NS		O/P (z)	
	x=0	x=1	x=0	x=1
A	D	B	0	0
B	A	E	1	0
D	B	A	1	1
E	E	D	1	0

— (2M)

10) b) State diagram :—



— (1M)

State Table :—

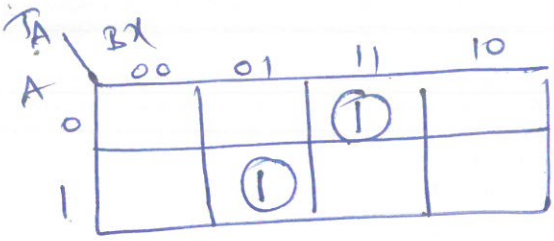
PS	NS	
A B	x=0	x=1
	A B	A B
0 0	0 0	0 1
0 1	0 1	1 1
1 0	1 0	0 0
1 1	1 1	1 0

— (1M)

Design using T flip-flops

PS	I/p	NS	FF Inputs	
A B	x	A B	T_A	T_B
0 0	0	0 0	0	0
0 0	1	0 1	0	1
0 1	0	0 1	0	0
0 1	1	1 1	1	0
1 0	0	1 0	0	0
1 0	1	0 0	1	0
1 1	0	1 1	0	0
1 1	1	1 0	0	1

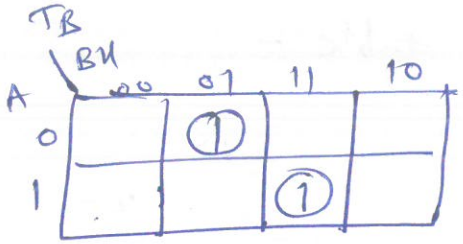
— (2M)



$T_A = AB'x + A'Bx$

$= x(A'B + AB')$

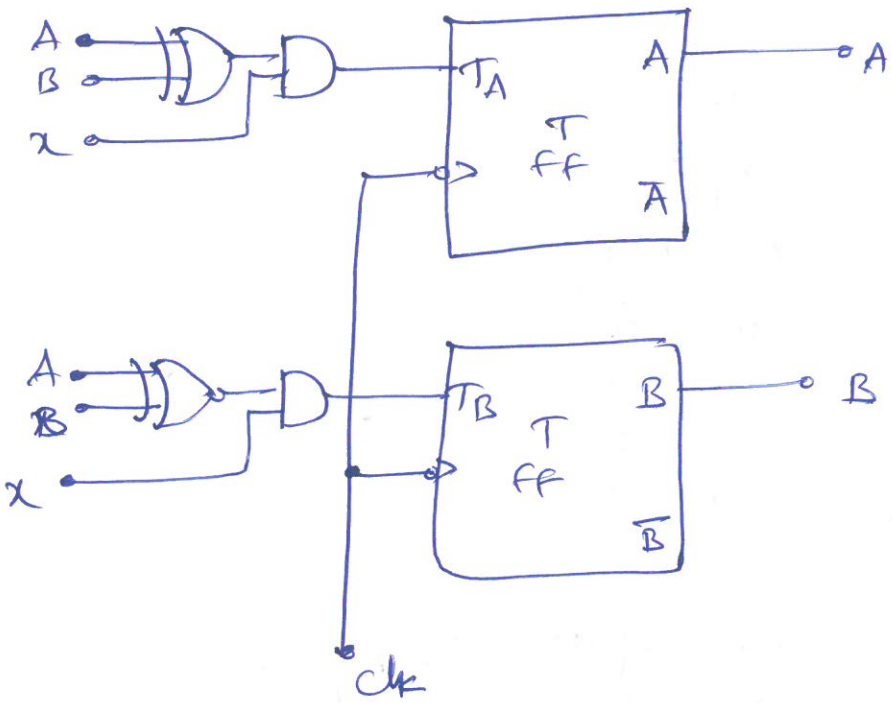
$T_A = x(A \oplus B)$



$T_B = A'B'x + ABx$

$T_B = x(AB + A'B')$

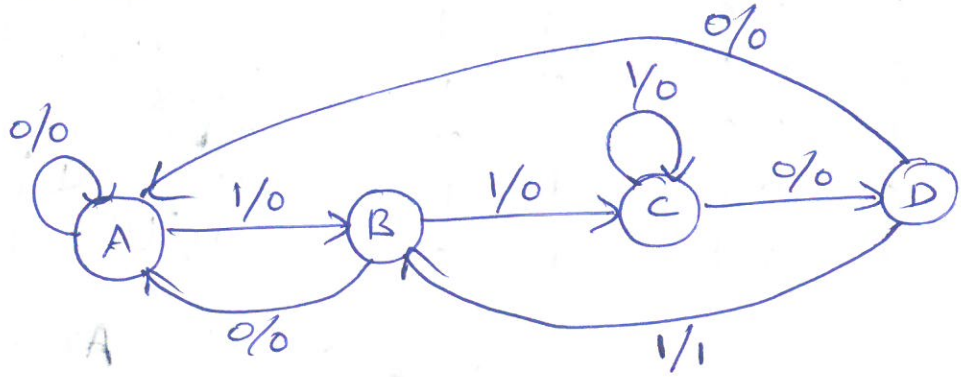
$T_B = (A \oplus B)'x$



Logic diagram using T flip-flops

— (1M)

11) Sequence Detector "1101"



— (3M)

State diagram of sequence Detector "1101"

State table :-

PS	NS		O/P	
	x=0	x=1	x=0	x=1
A	A	B	0	0
B	A	C	0	0
C	D	C	0	0
D	A	B	0	1

State Assignment :

A → 00 ; B → 01 ; C → 10 ; D → 11

PS	NS		O/P	
	x=0	x=1	x=0	x=1
00	00	01	0	0
01	00	10	0	0
10	11	10	0	0
11	00	01	0	1

— (2M)

PS AB	I/P x	NS AB	ff Inputs T_A	T_B
00	0	00	0	0
00	1	01	0	1
01	0	00	0	1
01	1	10	1	1
10	0	11	0	1
10	1	10	0	0
11	0	00	1	1
11	1	01	1	0

— (2M)

T_A

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

$T_A = AB + BX$

$T_A = B(A + X)$

T_B

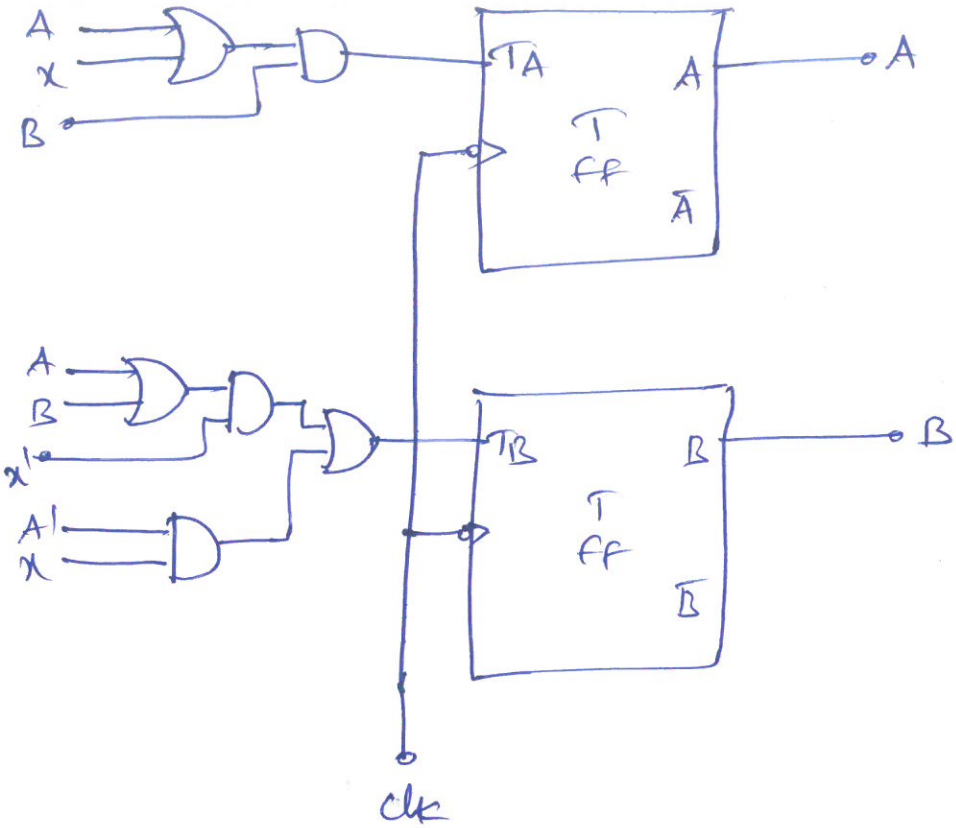
A \ BX	00	01	11	10
0		1	1	1
1	1			1

$T_B = Ax' + A'x + Bx'$

— (2M)

$\therefore T_A = (A+x)B$

$T_B = x'(A+B) + A'x$



1M

Logic diagram of sequence detector "1101" using T flip-flops.