Hall Ticket Number:

18EC305

II/IV B.Tech (Regular) DEGREE EXAMINATION

November, 2019

First Semester

Electronics and Communication Engineering

Engineering Mathematics

Time: Three Hours

Maximum : 50 Marks

Answer Question No.1 compulsorily. (1X10 = 10 Marks)

Answer ONE question from each unit. (4X10=40 Marks)

1. Answer all questions (1X10=10 Marks)

a) What is the 1's compliment of 1's compliment of the given binary number 10110011

b) What is the number of bits used in ASCII and EBCDIC codes ?

c) Design AND gate using NOR gate?

d) Convert A+B'C' to minterms.

e) Which logic device is called as data distributor ?

f) How many number of 2X1 Mux are required to construct 16X1 Mux ?

g) What is race around condition ?

h) Write the number of unused states in 4-bit johnson counter.

i) What are the applications of registers ?

j) Draw the symbols of p-MOS and n-MOS .

UNIT – I

2.a Determine the possible bases in each of the following operation which are correct in atleast one number system. 4M

(i)1234+5432=6666 (ii) $\frac{41}{3} = 13$ (iii) $\frac{302}{20} = 12.1$ (iv) $\sqrt{41} = 5$

2.b (i) Device a single error correcting code for a 11-bit group 01101110101 6M

(ii)Test the following Hamming code sequence for 11-bit message and correct it if necessary (101001011101011)

(OR)

3.a What is Overflow? Explain methods to detect overflow with example. 5M

3.b Obtain canonical SOP and POS of the following functions 5M

i)Y(A,B,C,D) =(AB+C)(B+C'D)

ii)Y(A,B,C,D) = AB+ACD

UNIT – II

4.a Minimize the following function using K-map 6M

F(A,B,C,D,E) = ∑ m (8,9,10,11,13,15,16,18,21,24,25,26,27,30,31)

4.b Explain wired OR & wired AND gates. 4M

(OR)

5.a Using the tabular method , obtain the minimal expression for 6M

F = ∑ m(6,7,8,9) +d(10,11,12,13,14,15)

5.b What are the steps involved in the design procedure of combinational circuits? 4M

UNIT – III

6.a Implement the following logic function with 8 to 1 multiplexer F(A,B,C,D) = ∑ m(0,3,6,8,11,13,15) 5M

6.b Design 3-bit comparator and explain its operation. 5M

(OR)

7.a What is priority encoder ? Explain 4 x 2 priority encoder. 5M

7.b Convert SR flip-flop to JK flip-flop. 5M

UNIT – IV

8.a Design a mod-10 ripple counter using JK flip-flop 5M

8.b Draw and explain Bi-directional shift register. 5M

(OR)

9.a Implement the following Boolean functions using PAL with four inputs and 3-wide AND-OR structure. Also write the PAL programming table. 6M

$F_1(A,B,C,D) = \sum m(2,12,13)$

$F_2(A,B,C,D) = \sum m(7,8,9,10,11,12,13,14,15)$

$F_3(A,B,C,D) = \sum m(0,2,3,4,5,6,7,8,10,11,15)$

$F_4(A,B,C,D) = \sum m(1,2,8,12,13)$

9.b Draw the symbols of CMOS NAND & CMOS NOR gates. 4M

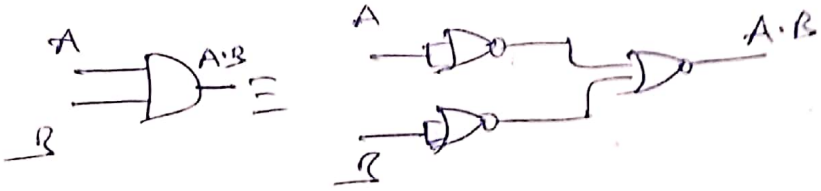
DE Scheme (18EC305)

1.a) 10110011

b) ASCII – 7 bit Code
EBCDIC – 8 bit Code

c) AND gate $A \cdot B$

$\overline{AB} = \bar{A} + \bar{B}$



d) $A + \bar{B}\bar{C}$

$[A(B+\bar{B})(C+\bar{C})] + [(A+\bar{A})\bar{B}\bar{C}]$

$[(AB+A\bar{B})(C+\bar{C})] + [A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C}]$

$[ABC + AB\bar{C} + A\bar{B}C + A\bar{B}\bar{C} + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C}]$

e) data distributer is nothing but De-multiplexer

f)

$\frac{16}{2} = 8$ (Stage-1)

$\frac{8}{2} = 4$ (Stage-2)

$\frac{4}{2} = 2$ (Stage-3)

$\frac{2}{2} = 1$ (Stage-4)

15

15 number of 2×1 are required for constructing one 16×1 Mux.

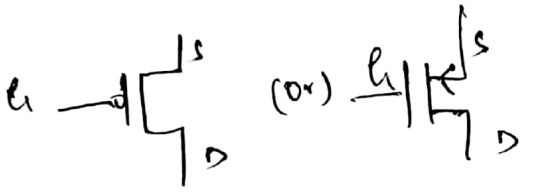
g) Race around condition:- The o/p of J-K F/F goes on complementing for every T_{pd} when J & K i/p's are equal to logic-1 and (T_{pd}) Propagation delay of F/F is less than pulse width of clock.

h) for 4 bit Johnson Counter unused states are

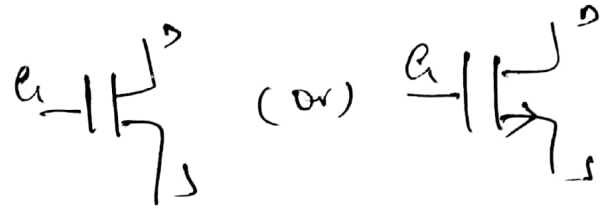
$$2^4 - (2\times4) = 16 - 8$$
$$= 8$$

i) Temporary storage of data and for shifting operations. SIPO is used for serial to parallel conversion, PISO is used for parallel to serial conversion

j) PISO



SISO



UNIT-I

2a) (i) $1234 + 5432 = 6666$

Valid in any number system with base ≥ 7

(ii) $\frac{41}{5} = 13$

$$\frac{4b+1}{3} = b+3$$

$$4b+1 = 3b+9$$

$$\boxed{b=8}$$

(iii) $\frac{302}{20} = 12.1$

$$\frac{3b^2+2}{2b} \approx b + 2 + \frac{1}{b}$$

$b = 4$

(iv) $\sqrt{41} = 5$

$$\sqrt{4b+1} = 5$$

$$4b+1 = 25$$

$b = 6$

2b)

11 bit group the bit Pattern is

P_1	P_2	D_3	P_4	D_5	D_6	D_7	P_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}
[ ]	[ ]	0	[ ]	1	1	0	[ ]	1	1	1	0	1	0	1

P_1 (1,3,5,7,9,11,13,15) must have even Parity. So $P_1 = 1$

P_2 (2,3,6,7,10,11,14,15) must have even Parity $P_2 = 0$

P_4 (4,5,6,7,12,13,14,15) must have even Parity $P_4 = 0$

P_8 (8,9,10,11,12,13,14,15) must have even Parity $P_8 = 1$

final code is 1000110111101O1

(b) given hamming code sequence 1010010111010 11

C_1 (1,3,5,7,9,11,13,15) → XOR → no error $C_1 = 0$

P_3 (3, 5, 6, 7, 10, 11, 14, 15) XOR → error $C_3 = 1$

P_2 (2, 3, 6, 7, 10, 11, 14, 15) XOR → no error $C_2 = 0$

P_1 (1, 3, 5, 7, 9, 11, 13, 15) XOR → no error $C_1 = 0$

$C_4C_3C_2C_1 = 0010$

error in 2nd bit

11100101110011 is correct code

3a) when two signed numbers are added and Result exceeds the given bit range then overflow occurs.

Two methods are there to detect overflow

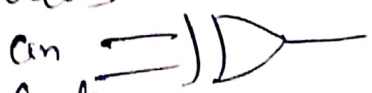
(i) By using sign bit :- if x and y are sign bit of given number and z is the resultant sign bit & condition for overflow is

x	y	z
1	1	0
0	0	1

$\bar{x}\bar{y}z + xy\bar{z}$

(ii) By using Carry method :-

Let Cin is carry into MSB and Cout is carry from MSB, if both are exclusive then overflow occurs



0 → No overflow.

1 → overflow.

3b) $Y(A, B, C, D) = (AB + \bar{C})(\bar{B} + \bar{C}D)$

$= AB + AB\bar{C}D + \bar{B}\bar{C}$ *

$= AB(C + \bar{C}) + AB\bar{C}D + (A + A)\bar{B}\bar{C}$

$AB(C + \bar{C})$	$AB\bar{C}D$	$(A + A)\bar{B}\bar{C}$
1100 ✓	1101 ✓	0110 ✓
1101 ✓		0111 ✓
1110		1110
1111		1111

$Y(A,B,C,D) = \Sigma m(6,7,12,13,14,15)$

↳ SOP

$= \Pi M(0,1,2,3,4,5,8,9,10,11)$

↳ POS

ii) $Y(A,B,C,D) = AB + ACD$

$= AB(C+\bar{C})(D+\bar{D}) + A(B+\bar{B})CD$

$= (ABC + AB\bar{C})(D+\bar{D}) + ABCD + A\bar{B}CD$

$= ABCD + AB\bar{C}D + ABC\bar{D} + AB\bar{C}\bar{D} + ABCD + A\bar{B}CD$

1111 1101 1110 1100 1111 1011

$Y(A,B,C,D) = \Sigma m(11,12,13,14,15)$

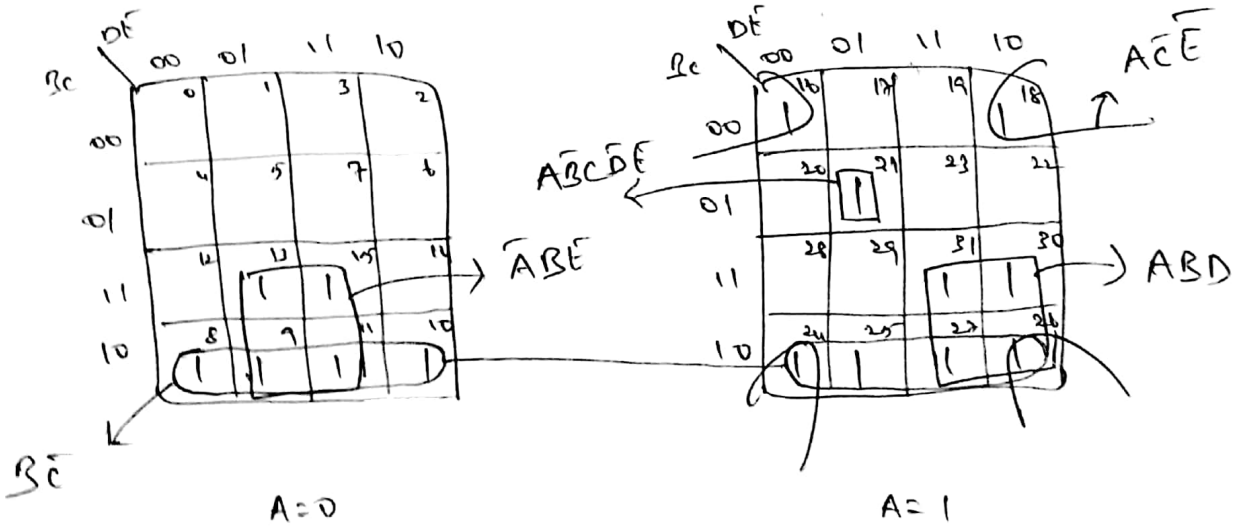
↳ SOP

$Y(A,B,C,D) = \Pi M(0,1,2,3,4,5,6,7,8,9,10)$

↳ POS

UNIT - II

4a)

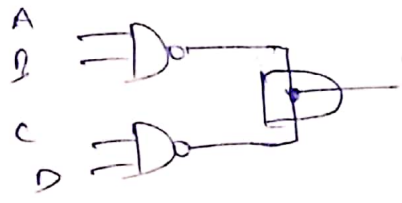


minimize exp = $B\bar{C} + \bar{A}B\bar{E} + ABD + A\bar{C}\bar{E} + AB\bar{C}\bar{D}E$

4b)

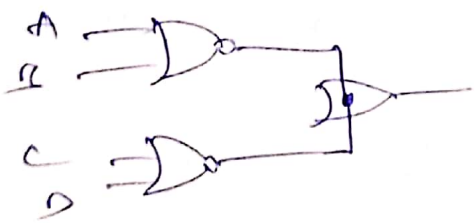
Some NAND or NOR gates allow the possibility of wire connection between the outputs of two gates to provide a specific logic function. This type of logic is called wired logic.

* Open collector TTL NAND gates, when tied together perform the wired-AND logic.



wired AND gate is not a physical gate but only a symbol to designate the function obtained from indicated wired connection.

* NOR output of ECL gates when tied together to perform a wired-OR function.



wired logic does not produce a physical second-level gate since it is just a wire connection.

5a) $F = \Sigma m(6,7,8,9) + d(10,11,12,13,14,15)$.

Tabular Method.

Index	minterms	variables ABCD.
Index 1	8	1000 ✓
Ind. 2	6	0110 ✓
	9	1001 ✓
	10	1010 ✓
	12	1100 ✓
3	7	0111 ✓
	11	1011 ✓
	13	1101 ✓
	14	1110 ✓
4	15	1111 ✓

Step 1 ✓

Index	Pairs	variables ABCD
1	8,9	100_ ✓
	8,10	10_0 ✓
	8,12	1_00 ✓
2	6,7	011_
	6,14	_110 ✓
	9,11	10_1 ✓
	9,13	1_01 ✓
	10,11	101_ ✓
	10,14	1_10 ✓
	12,13	110_ ✓
	12,14	11_0 ✓
3		

Index	Pairs	variable. ABCD	
1	7, 15	-111	✓
	11, 15	1-11	✓
	13, 15	11-1	✓
	14, 15	111-	✓

Step-3

Index	quads	vaiable	
1	8, 9, 10, 11	10--	✓
	8, 9, 12, 13	1-0-	✓
	8, 10, 12, 14	1--0	✓
2	6, 7, 14, 15	-11-	✓
	9, 11, 13, 15	1--1	✓
	10, 11, 14, 15	1-1-	✓
	12, 13, 14, 15	11--	✓

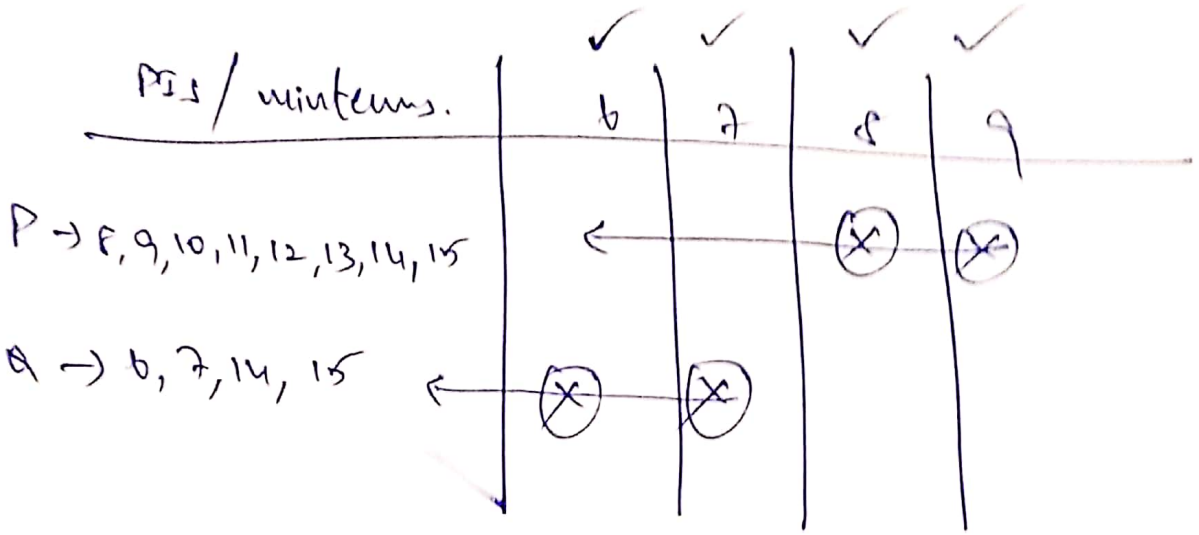
Step-4

Index	octets	variable	
1	8, 9, 10, 11, 12, 13, 14, 15	1---	P

P → 8, 9, 10, 11, 12, 13, 14, 15

Q → 6, 7, 14, 15

Prime implicant table



minimal expression is P + Q

⇒ A + BC

5b)

Steps for design Procedure of comb. ckts.

(1) The Problem is stated.

(2) The number of available input variables and required output variables is determined.

(3) The input and output variables are assigned letter symbols.

(4) The truth table that defines the required relationships between i/p and o/p is derived.

(5) The simplified Boolean function for each output is obtained

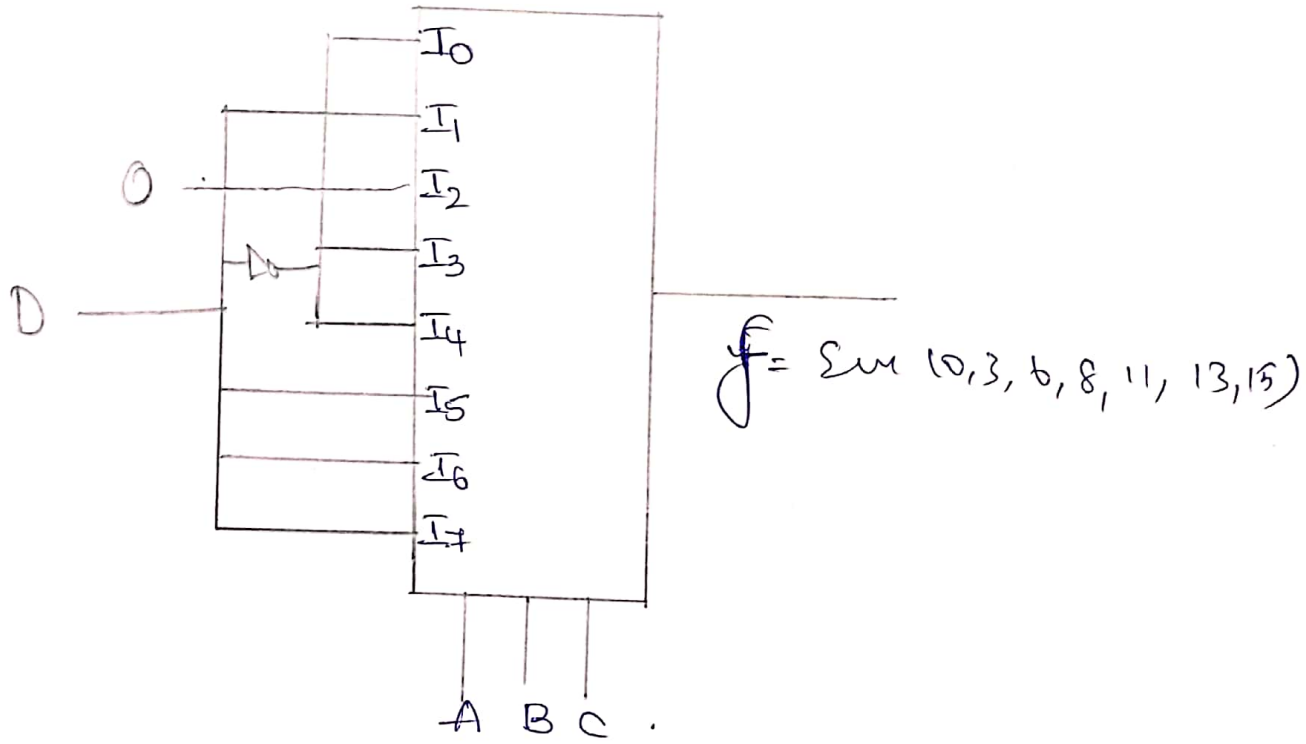
(6) logic diagram is drawn.

6. a). $F(A, B, C, D) = \Sigma m(0, 3, 6, 8, 11, 13, 15)$.

Implementation table:

Consider A, B, C as selection lines.

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
$\bar{D}$ ⇒ 0	(0)	2	4	(6)	(8)	10	12	14
D ⇒ 1	1	(3)	5	7	9	(11)	(13)	(15)
	$\bar{D}$	D	0	$\bar{D}$	$\bar{D}$	D	D	D

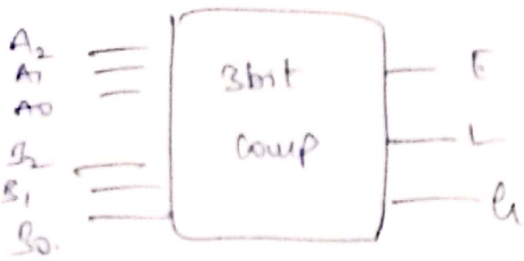


6 b). 3-bit Comparator

* Comparator ckt is used to compare magnitude of two binary numbers.

* The 3-bit A is compared with 3-bit B.

* output consists of $A>B$, $A<B$, $A=B$ cases.

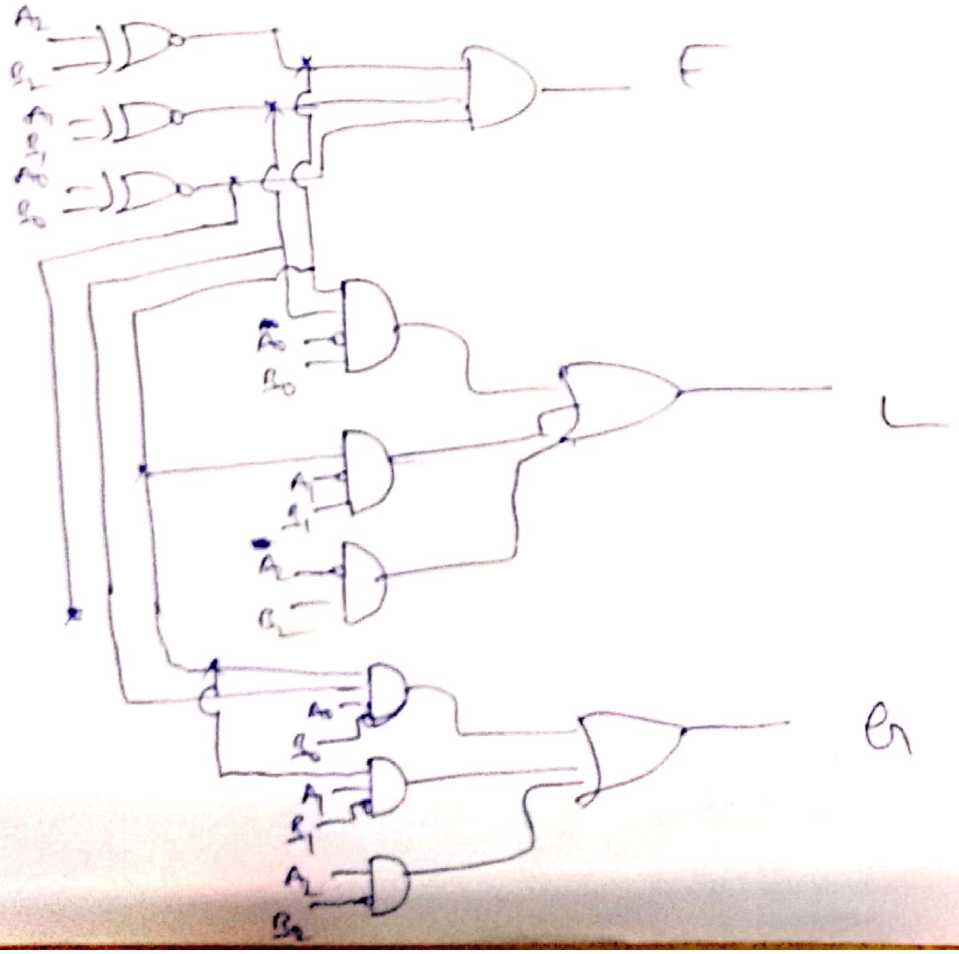


* Combinational circuits have regularity, direct expression, can be written as follows.

$$E = (A_2 \odot B_2) \cdot (A_1 \odot B_1) \cdot (A_0 \odot B_0)$$

$$L = \bar{A}_2 B_2 + (A_2 \odot B_2)\bar{A}_1 B_1 + (A_2 \odot B_2)(A_1 \odot B_1)\bar{A}_0 B_0$$

$$G = A_2 \bar{B}_2 + (A_2 \odot B_2) A_1 \bar{B}_1 + (A_2 \odot B_2)(A_1 \odot B_1) A_0 \bar{B}_0$$



7. a) (7)

Priority Encoder

A priority Encoder is a logic circuit that responds to just one input in accordance with some priority system, among all those that may be simultaneously high. The most common priority system is based on the relative magnitudes of the inputs.

4 X 2 priority Encoder :

A 4 to 2 priority encoder takes 4-inputs & produces 2-outputs. If all the inputs are 0, there is no valid input and V is equal to 0.

According to the truth table, the higher the subscript number, the higher the priority of the input.

From truth table.

$$A = D_3 + \bar{D}_3 D_2$$

$$= D_3 + D_2.$$

$$B = D_3 + \bar{D}_3 \bar{D}_2 D_1$$

$$= D_3 + \bar{D}_2 D_1$$

$$V = D_3 + D_2 + D_1 + D_0.$$

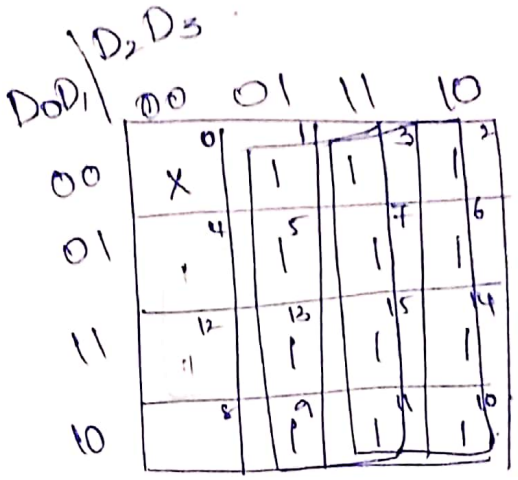
The condition for output is an OR function of all the input variables.

The minterms for the two functions A & B are.

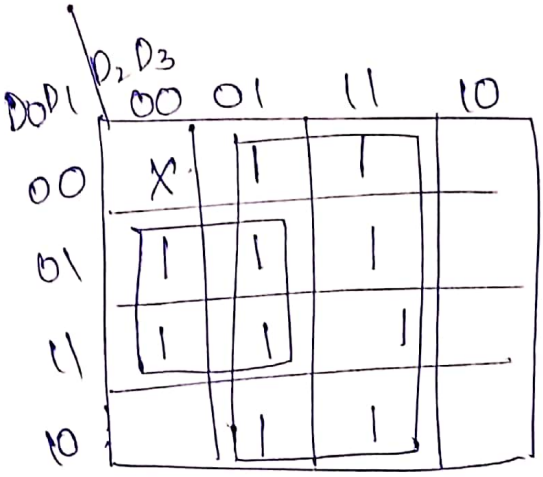
$$A = \Sigma m(1, 2, 3, 5, 6, 7, 9, 10, 11, 13, 14, 15)$$

$$B = \Sigma m(1, 3, 4, 5, 7, 9, 11, 12, 13, 15)$$

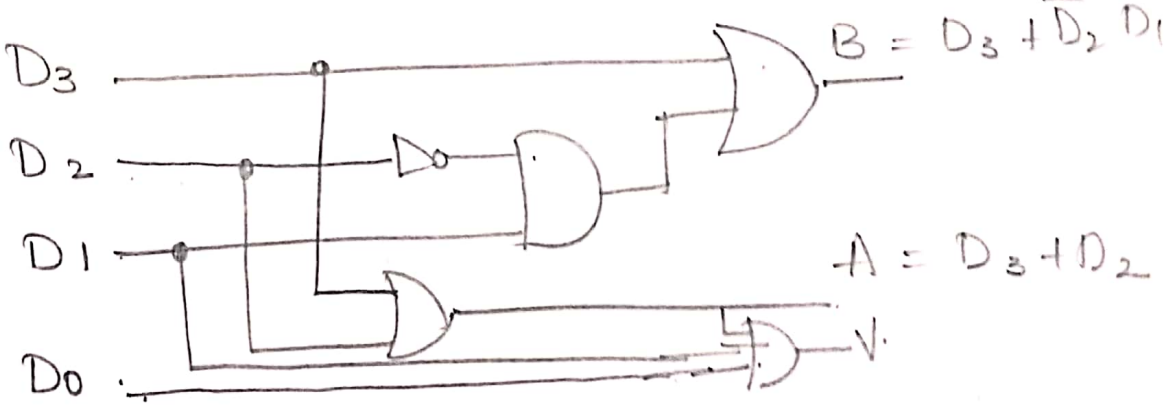
D_3	D_2	D_1	D_0	A	B
0	0	0	0	X	X
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	0	1
0	1	0	0	0	1
0	1	0	1	1	1
0	1	1	0	1	0
0	1	1	1	1	1
1	0	0	0	0	1
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	0	1
1	1	0	1	1	1
1	1	1	0	1	0
1	1	1	1	1	1



$A = D_3 + D_2$



$B = D_3 + D_1\overline{D_2}$



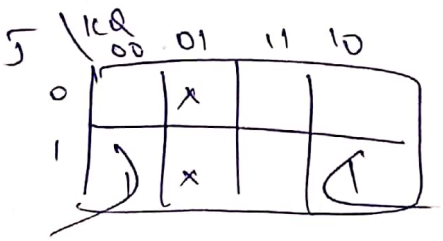
4. b). **SR flip-flop to JK flipflop:**

SR → available f/f hence excitation table

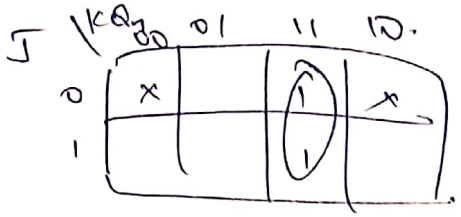
JK → Required f/f hence characteristic table

J	K	Q	Q(t+1)	S	R
0	0	0	0	0	x
0	0	1	1	x	0
0	1	0	0	0	x
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	x	0
1	1	0	1	1	0
1	1	1	0	0	1

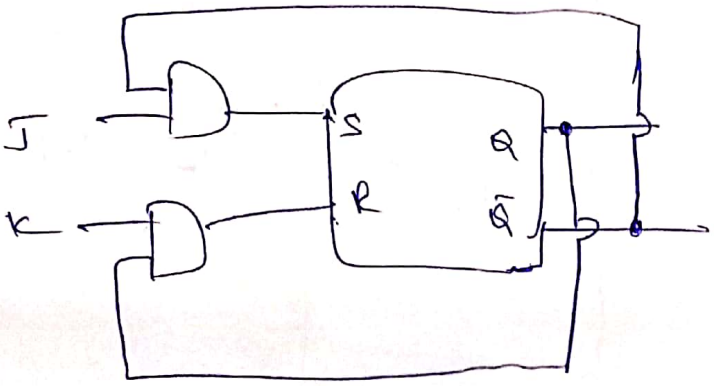
Equations for S, R using K-map.



$S = J\bar{Q}_n$



$R = KQ_n$



8. a). MOD-10 ripple counter using JK-flipflop:

give no. of states N: 10

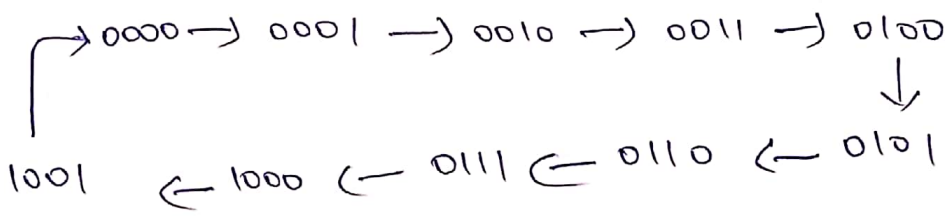
no. of flip-flops required n: ?

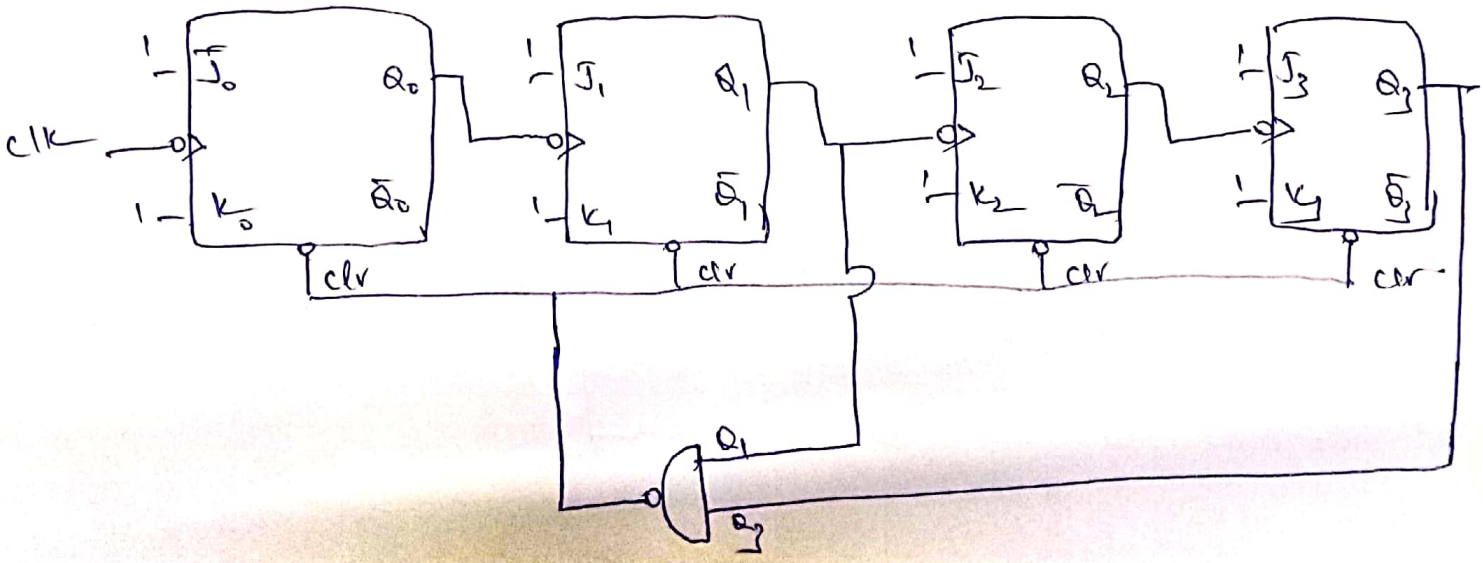
$2^n \geq N$

$2^n \geq 10$

$n = 4$

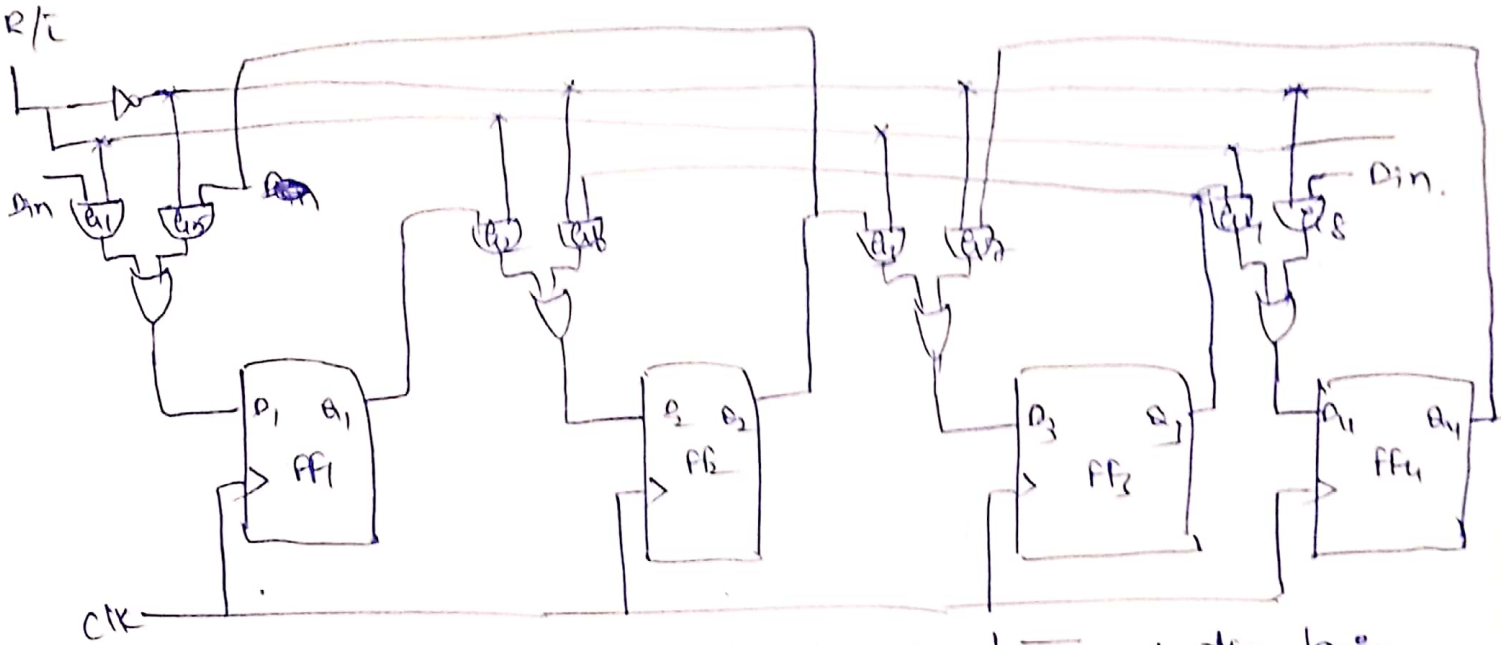
* In ripple counter External clock is connected to LSB flip-flop, and clock for remaining F/F will be taken from previous f/f output.
* All FF's must operate in toggle mode.
* At 10th clock pulse nand gate ~~becomes~~ output becomes low and it clears all flf outputs.





8. b. Bi-directional Shift Register:

A bidirectional shift register is one in which the data bits can be shifted from left to right & from right to left.



Right / $\overline{\text{Left}}$ is mode signal. When Right / $\overline{\text{Left}}$ = 1, the logic circuit works as a shift-right register. When Right / $\overline{\text{Left}}$ = 0 it works as a shift-left register. The bidirectional operation is achieved by using the mode signal & two AND gates & one OR gate for each stage.

A HIGH on the Right / $\overline{\text{Left}}$ control input enables the AND gates G_1, G_2, G_3 & G_4 and disables the AND gates G_5, G_6, G_7 & G_8 and the state of Q output of each FF is passed through the gate to the D input of the following FF. When a clock pulse occcurs, the data bits are then effectively shifted one place to the right.

A LOW on the Right / $\overline{\text{Left}}$ control input enables the AND gates G_5, G_6, G_7 & G_8 and disables the AND gates G_1, G_2, G_3 & G_4 & the Q output of each flipflop is passed to the D input of the preceeding FF / when a clock pulse

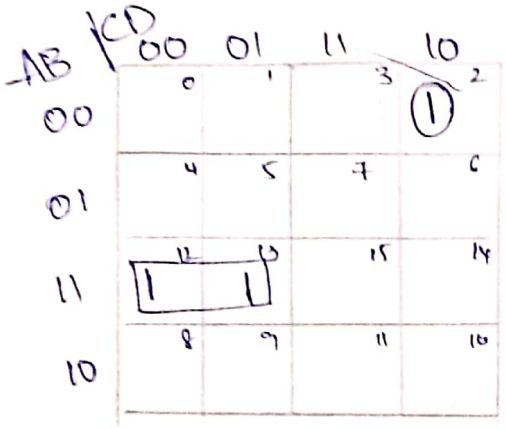
occurs, the data bits are then effectively shifted one place to the left. Hence, the circuit works as a bidirectional shift register.

9. a). $F_1(A, B, C, D) = \Sigma(2, 12, 13)$

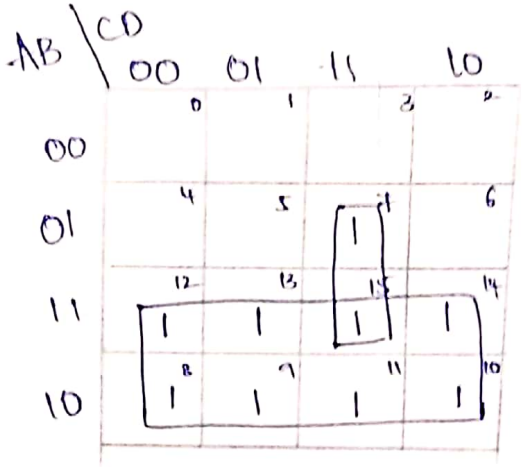
$F_2(A, B, C, D) = \Sigma m(1, 8, 9, 10, 11, 12, 13, 14, 15)$

$F_3(A, B, C, D) = \Sigma m(0, 2, 3, 4, 5, 6, 7, 8, 11, 15)$

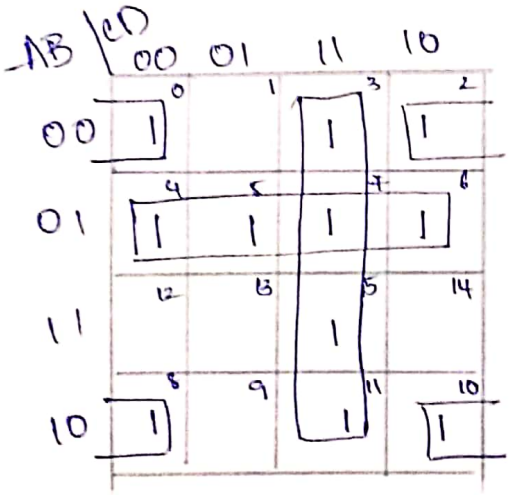
$F_4(A, B, C, D) = \Sigma m(1, 2, 8, 12, 13)$



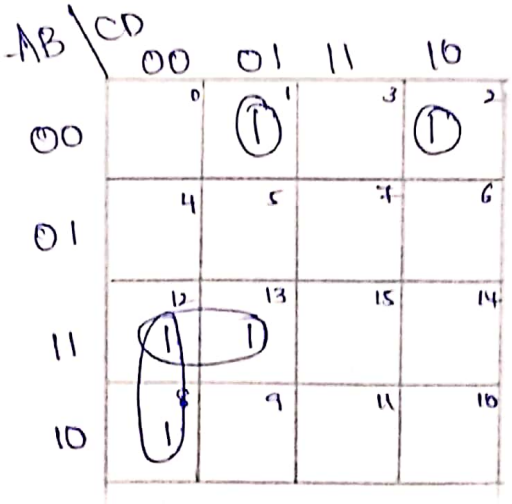
$F_1 = AB\bar{C} + \bar{A}\bar{B}C\bar{D}$



$F_2 = A + BCD$

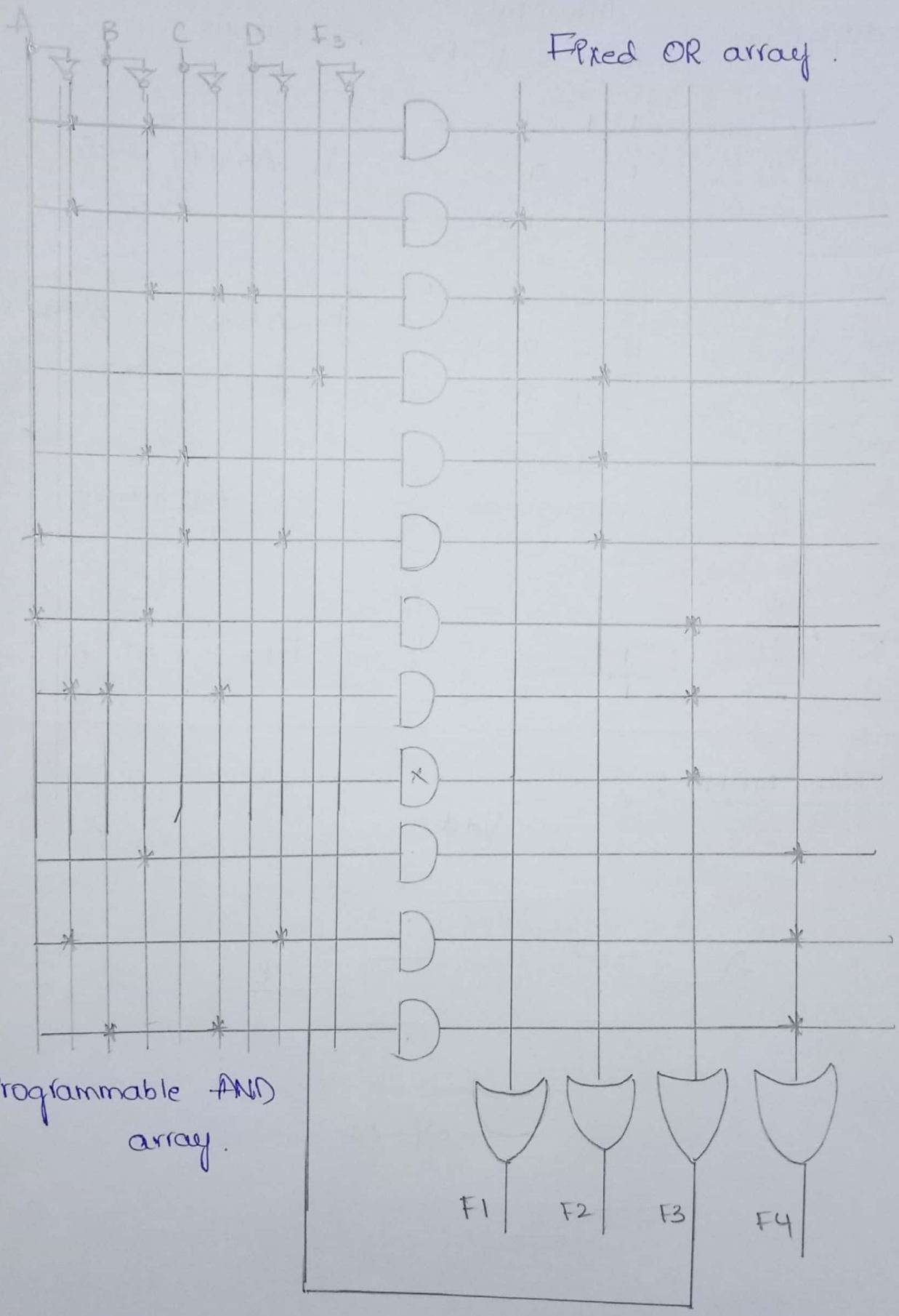


$F_3 = CD + \bar{A}B + \bar{B}\bar{D}$



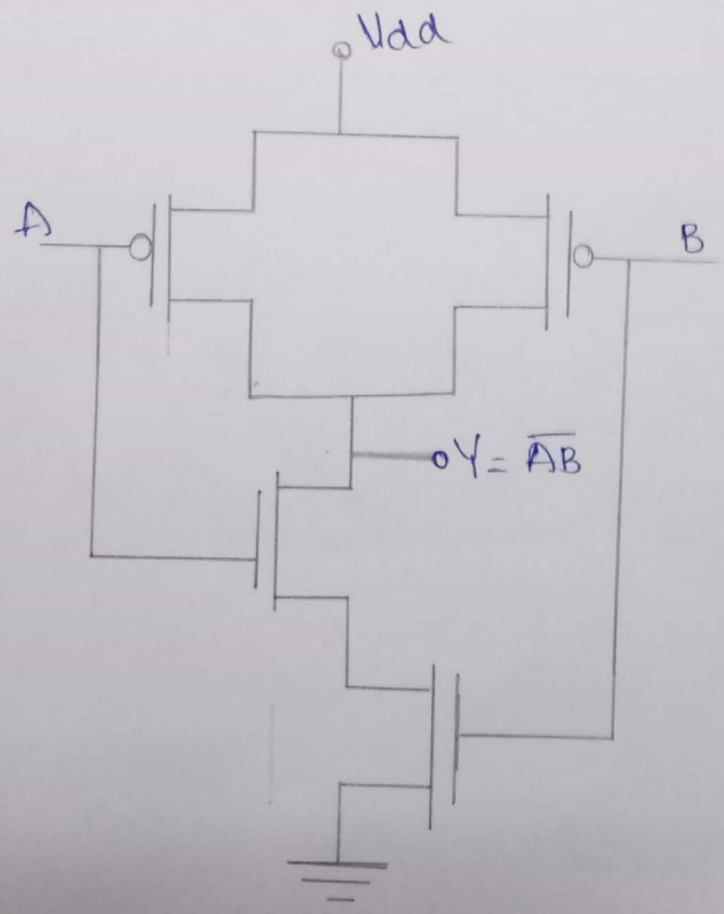
$F_4 = \bar{A}B\bar{C}D + \bar{A}BC\bar{D} + A\bar{C}\bar{D} + AB\bar{C}$

$F_4 = F_1 + A\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D.$



Product term	AND Inputs A	B	C	D	F3	Outputs
1	1	1	–	–	–	$F_1 = AB + A\bar{C} + BC\bar{D}$
2	1	–	0	–	–	
3	–	1	1	0	–	
4	–	–	–	–	1	$F_2 = \bar{A}B + A\bar{B}C + \bar{A}\bar{C}D + B\bar{C}$
5	0	–	0	1	–	$= F_3 + B\bar{C} + \bar{A}\bar{C}D$
6	–	1	0	–	–	
7	0	1	–	–	–	$F_3 = \bar{A}B + A\bar{B}C$
8	1	0	1	–	–	
9	–	–	–	–	–	
10	–	1	–	–	–	$F_4 = B + AD + AC$
11	1	–	–	1	–	
12	1	–	1	–	–	

9.b). CMOS NAND :



CMOS NOR :