$f(C,D,A,B) = CD + DA$

$= D\cdot(C+A)$

Minimal SOP

2.2

yz \ wx	00	01	11	10
00	0	1	1	0
01	X	0	0	1
11	X	0	0	1
10	0	1	1	X

$f(w,x,y,z) = x\bar{z} + \bar{x}z$

$= x \oplus z$

Imp Note → Technique use for considering min term is important. Always remember this.

2.3

xy \ zw	00	01	11	10
00	X	1	0	1
01	0	1	X	0
11	1	X	X	
10	X	0	0	X

$f(z,w,x,y) = \bar{w}\bar{y} + \bar{z}w\bar{x} + \bar{z}xy$

No. of literals = 8

Q which function doesn't implements the K-map?

xy \ wz	00	01	11	10
00	0	X	0	0
01	0	X	1	1
11	1	1	1	1
10	0	X	0	0

a) $(w+x)y$ b) $xy + yw$

c) $(w+x)(\bar{x}+y)(\bar{w}+y)$ d) None

SOP = $wy + xy = y(w+x)$

POS = $(w+y)\cdot(x+y)$

Ans → d) None

To verify option c) do each POS term contain two literals so we will cover all zeros with cube size 4, and from given terms in POS we can cover all the 0's.

Finding minimal expression.

CD \ AB	00	01	11	10
00	1			1
01	1	1	1	1
11		1	1	
10	1			1

(d) (c) (b) (a)

We have four prime implicants out of which only two are essential and two are redundant.

→ We are going to use prime implicant charts.

	0	1	2	5	7	8	9	10	13	15
a	1		1			1		1		
b				1	1				1	1
c		1		1			1		1	
d	1	1				1	1			

$f(ABCD) = \underline{a + b + c/d}$

Since, c and d both are covering same no. of cubes so we can choose anyone otherwise we will choose the term covering larger no. of cells.

Which of the following can be prime implicant of f(A, B, C)

① ab + c → Needs 2 subcubes

	00	01	11	10
0			1	
1	1	1	1	1

② bc + a → can't possible to denote

	00	01	11	10
0				
1		1	1	

③ ac + b → As above option b also can't display alone

④ ab → requires 1 subcube.

→ for a prime implicant complete option should lie in a single largest subcube

So, ④ ab is the answer.

Hint → In previous question it is necessary complete option i.e. (ab + c) lies in single subcube. We are not considering ab and c are different.

What min. terms have to be added to make the following functions a self dual?

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

pair the combination of all possible min. term.

(0,15) (1,14) (2,13) (3,12)
(4,11) (5,10) (6,9) (7,8)

→ Self dual function doesn't contain mutually exclusive terms.

After expansion we get

$\Rightarrow A'BC(D+D') + AC(B+B')D + (A+A')BC(C+C')D$

$\Rightarrow A'BCD + A'BCD' + ABCD + AB'CD + AB\cancel{C}D + ABC'D + A'\cancel{B}CD + A'BC'D$

$\Rightarrow \underset{(7)}{A'BCD} + \underset{(6)}{A'BCD'} + \underset{(15)}{ABCD} + \underset{(11)}{AB'CD} + \underset{(5)}{A'BC'D} + \underset{(13)}{ABC'D}$

→ Verify that expression doesn't contain mutually exclusive terms.

→ Add the terms which are not belongs to any pair.

To make the function self dual we add the two numbers as -

(1, 3) (14, 3) (1, 12)

(14, 12).

We can choose any one of the order pair from above 4.

Combining functions having dont cares

①

$f_1(a,b,c) = \Sigma(0,2,4) + \Sigma_\phi(3,5,7)$

$f_2(a,b,c) = \Sigma(2,3) + \Sigma_\phi(1,6,7)$

How many functions does $f_1 \cdot f_2$ and $f_1 + f_2$ represents.

Solⁿ

	f_1	f_2	$f_1 \cdot f_2$	$f_1 + f_2$
0	1	0	0	1
1	0	ϕ	0	ϕ
2	1	1	1	1
3	ϕ	1	ϕ	1
4	1	0	0	1
5	ϕ	0	0	ϕ
6	0	ϕ	0	ϕ
7	ϕ	ϕ	ϕ	ϕ

→ ϕ can hold any value from {0, 1} but any value of ϕ if 'AND' with '0' gives '0' and it 'OR' with '1' gives '1' as output.

$f_1 \cdot f_2$ have 2 don't cares that can have any value {0, 1}

So, $2^2 = 4$ functions.

Similarly $f_1 + f_2$ will have $2^4 = 16$ functions.

→ Similarly we can extend this concept upto any no. of variables.

Prime implicants and don't cares

Q The Number of prime implicants, essential prime implicants and redundant prime implicants for the function

$f(A,B,C) = \Sigma(2,5,6,7)$ are ———.

Solⁿ

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

No. of prime implicants = 3

No. of essential prime implicants = 2

No. of redundant ——— = 1

Q A function $f(A,B,C) = \Sigma(3,5,6)$ minimize to $(A + BC)$ then what are the don't cares.

Solⁿ

C \ AB	00	01	11	10
0			1	ϕ
1		1	ϕ	1

$\phi = (4, 7)$

A

BC

Dontcare should be at 4, 7.

Q In a K-map it was found out that essential PI are covering all terms except 2 min terms. Those 2 min terms are in term covered by 3-non essential prime implicants each.

What is the no. of minimal SOP expressions.

Sol^n

$f = EPI + \underbrace{\_\_\_}_{3} + \underbrace{\_\_\_}_{3}$

3×3

⇒ Total 9 functions are possible.

Excellent Question

Q In a prime implicant chart representing a boolean expression f(w,x,y,z) columns represent min terms and rows represent prime implicants, Identify P, Q, R, S and a, b.

	0	4	5	7	8	a	b
P	✓	✓					
Q	✓				✓		
R		✓	✓				
S			✓	✓		✓	✓

Sol^n [solve it self by looking the solution otherwise watch video

yz \ wx	00	01	11	10
00	1	1		1
01		1	1	
11		1	1	
10				

P → w'y'z'

Q → x'y'z'

R → w'xy'

S → xz

for S if we try to add some more 1 in k-map so that we can form a biggest cube. So that existence of P and R should maintain and also we can find out a, b.

So,

$\boxed{\begin{array}{c} a = 13 \\ b = 15 \end{array}}$

we don't include a = 1 b = 3 b'coz a, b are greater than 8 in table.

Variable Entered Map (VEM)

$f(A,B,C) = \Sigma(1,2,3,5,6)$

A	B	C	f	
0	0	0	0	→ As C
0	0	1	1	
0	1	0	1	→ all 1's
0	1	1	1	
1	0	0	0	→ As C'
1	0	1	1	
1	1	0	1	→ As C'
1	1	1	0	

A B	f
0 0	C
0 1	1
1 0	C
1 1	C'

Behaviour of f on value of AB.

In (f) section we can choose B instead of 1 but we try to make it simple so choose 1.

B \ A	0	1
0	C	C
1	1	C'

→ This way we can reduce the k-map in less no. of variables.

Minimization of VEM

Ex:

C \ AB	00	01	11	10
0	D	1	D'	D'
1	D	1	0	ϕ

Consider D and D' as the separate variables.

Step 1:

C \ AB	00	01	11	10
0	0	1	0	0
1	0	1	0	ϕ

→ A'B (SOP)

→ Replace D and D' by 0 and find product term.

Step 2 → Replace 1 → ϕ and D by 1 in original k-map, and D' → 0.

C \ AB	00	01	11	10
0	1	ϕ	0	0
1	1	ϕ	0	ϕ

→ A' [A'D]

Step-3 Replace 1 → ϕ, D → 0 and D' → 1 in original k-map

C \ AB	00	01	11	10
0	0	ϕ	1	1
1	0	ϕ	0	ϕ

→ AC' [AC'D']

Step 4: Solution can be written as

$$\boxed{A'B + A'D + AC'D'}$$

Procedure:

① Set all the variables in the cell as '0' and obtain SOP expression.

② a) Make one variable in the cell as '1' and obtain SOP by making earlier min terms (1's) as don't cares.

b) Multiply the above SOP with concern variables.

③ Repeat the step 2 untill all the variables in the cells are covered.

④ SOP of VEM is obtained by (ORing) the previous SOP expressions.

Ex → $f(A,B,C) = \Sigma(3,5,6,7)$ is realized by following VEM, then find P, Q, R, S.

b \ a	0	1
0	P	S
1	R	Q

a) P = 0, R = S = C, Q = 1

b) P = Q = 0, R = C, S = 1

c) P = 0, R = Q = C, S = 1

d) None.

Soln

A	B	C	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

⇒ (VEM)

A	B	f
0	0	0
0	1	C
1	0	C
1	1	1

P = 0, Q = 1, R = C, S = C] a)

B \ A	0	1
0	0 (P)	C (S)
1	C (R)	1 (Q)

(K-Map)

Question was about finding value of P, Q, R, S but we are extending it to find minimal expression.

b \ a	0	1
0	0	c
1	c	1

Step 1 :

b \ a	0	1
0	0	0
1	0	1

→ ab

Step 2 :

b \ a	0	1
0	0	1
1	1	ϕ

$(a+b)c$

Step-3 : $ab + ac + bc$ is the minimal expression.

→ It is also a self dual function.

finding the free variables

How many variables are free in the expression denoted by the following K-map?

yz \ wx	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	1	0	0	1
10	0	1	1	0

Sol^n

$$f = x\bar{z} + \bar{x}z = x \oplus z$$

→ It depends on only two variables x & z.

→ w, y are free variables.

SOP = $x\bar{z} + z\bar{x}$

POS = $(x+z)\cdot(\bar{x}+\bar{z})$

If SOP depends on two variable POS will also depends on two variables.

Relation between POS and SOP in case of don't care

1.

yz \ wx	00	01	11	10
00	0	ϕ	ϕ	0
01	0	0	1	1
11	1	1	1	1
10	0	0	0	0

→ wz + → yz

→ If a K-map have n-cells with don't care then total 2^n combinations of functions are possible.

→ We will include don't care if and only if it will provide us the subcube of largest size otherwise not.

→ $f = (wz + yz)$ from above K-map.

→ If we include ϕ (don't care) we consider them as '1' bcoz we want to include it in sub cubes otherwise we assume it as '0'. [In SOP]

→ In above K-map, we have no need to include ϕ so we assume them as '0'

	01	11
00	0	0

Consider the table given below

w x y z	f	f_1	f_2	f_3	f_4
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	1	1	1	1	1
4	ϕ	0	0	1	1
5	0	0	0	0	0
6	0	0	0	0	0
7	1	1	1	1	1
8	0	0	0	0	0
9	1	1	1	1	1
10	0	0	0	0	0
11	1	1	1	1	1
12	ϕ	0	1	0	1
13	1	1	1	1	1
14	1	1	1	1	1
15	1	1	1	1	1

we can repace ϕ ϕ as

0 0, 0 1, 1 0, 1 1.

As we find the minimal exp. as $wz + yz$ so we are not considering this don't care terms means assuming both as 0. so we are following the specific function f_1.

$\Rightarrow f_1$ is special case of f.

Similarly if we find POS by considering both ϕ as 0 then we get,

$POS = z \cdot (w+y)$

So, by considering f_1 we get both minimal POS and SOP.

Ex: 2 Let $f(w,x,y,z)$ is a function containing $(1,2,3,5,13)$ as min. terms and $(6,7,8,9,11,15)$ as don't care and let $f(w,x,y,z)$ be minimized SOP and $g(w,x,y,z)$ is minimized POS. Are f and g identical.

Sol^n

yz \ wx	00	01	11	10
00	0	0	0	ϕ
01	1	1	1	ϕ
11	1	ϕ	ϕ	ϕ
10	1	ϕ	0	0

→ for SOP

$SOP = f = z + \bar{w}y$

$f = \Sigma(1,2,3,5,6,7,9,11,13,15)$

yz \ wx	00	01	11	10
00	0	0	0	ϕ
01	1	1	1	ϕ
11	1	ϕ	ϕ	ϕ
10	1	ϕ	0	0

→ for POS

$g = \Sigma(1,2,3,5,13, 9)$

Here, 2^6 combinations of functions are possible due to don't care. So we can choose any 1 from 64 for minimal SOP and minimal POS.

→ And for 'f' and 'g' be identical only one function amoging 64 must satisfy both minimal POS and SOP.

→ In case of POS, include ϕ don't care assumes as '0' and not includes assumes as '1'.

→ Although 'f' have only one way to get minimal SOP but 'g' have many way to get mininmal POS. So both are not identical.

Comparing independent variables in minimal SOP and POS

Q. Minimal expression represented by map is free from

1) 1-variable
2) 2 variables
3) 3-variables
4) dependent on all. ✓

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

$\underline{SOP} = C + \bar{A}D + \bar{A}\bar{B}$

$\underline{POS} = (B\bar{C}\bar{D} + A\bar{C})'$

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

So, Both POS and SOP dependent on all four variables.

Note: imp. If no don't care present in K-map then no. of dependent variable should be same in both both POS and SOP.

Number of irredundant and minimal expression

Question.

① No. of prime implicants?
② No. of essential prime implecants?
③ No. of redendant prime implicants?
④ Minimal SOP?
⑤ How many minimal?

CD \ AB	00	01	11	10
00	1	1		
01		1	1	
11			1	1
10				1

P, Q, R, S, T, V

Sol^n →

6 prime implicant.
2 essential prime implicant.
4 redundant prime implicant.

Prime implicant chart (Table 1)

	0 ✓	4 ✓	5	10 ✓	11 ✓	13	15
P	✓	✓					
Q		✓	✓				
R			✓			✓	
S						✓	✓
T					✓		✓
V				✓	✓		

$P = A'C'D'$, $Q = A'BC'$
$R = BC'D$, $S = ABD$
$T = ACD$, $V = AB'C$

Essential PI = $P + V$

$P \to (0,4)$, $V \to (10,11)$

So, we need to cover 5, 13, 15 using remaining 4 product terms.

Reduced prime implicant chart

	5	13	15
Q	✓		
R	✓	✓	
S		✓	✓
T			✓

(Table 2)

→ Q is dominated by R and T is dominated by S
It means whatever Q, T Includes already includes in R and S but whatever R and S required Q and T are not able to provide.

So, we can remove Q and T bcoz all are of same size.
$Q \to R$, $T \to S$

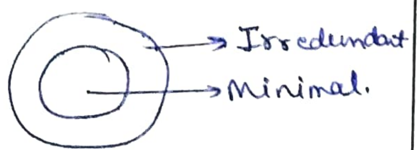
	5	13	15
R	✓	✓	
S		✓	✓

Table 3

So, minimal expression is $\underline{P+V+R+S}$

No. of minimal SOP possible:

Every minimal expression is irredundant but vice-versa is not always true.



from Table ② we can say that to cover

5 we can use Q or R
13 ———— R or S
15 ———— S or T

$\Rightarrow (Q+R)\cdot(R+S)\cdot(S+T)$

$\Rightarrow QRS + QRT + QSS,$
$QST + RRS + RRT$
$+ RSS + RST$

$\Rightarrow QRS + QRT + QS + QST$
$+ RS + RT + \cancel{RS} + RST$

$\Rightarrow QS(1+R) + RS(1+T)$
$+ RT(1+Q) + QST$

$\Rightarrow QS + RS + RT + QST$

$\Rightarrow \underline{QS + RS + RT}$

So, to cover (5, 13, 15) we can use combinations like (Q, S), (R, S), (R, T)

So, no. of minimal expression which are irredundant = 3 as,

$$\left.\begin{array}{l} P+V+Q+S \\ P+V+R+S \\ P+V+R+T \end{array}\right] \underline{3 \text{ expression}}$$

→ So finding no. of minimal exp. is a lengthy process.

Don't care never included in prime implicant chart

$f(v, w, x, y, z) =$
$\Sigma(13, 15, 17, 18, 19, 20, 21, 23, 25, 27, 29, 31) + \Sigma\phi(1, 2, 12, 24)$

$A = vz$ $\quad C = vwx'y'$ $\quad E = vw'xy$ $\quad G = w'x'yz$
$B = wxz$ $\quad D = vw'xy'$ $\quad F =$ [illegible]

	13	15	17	18	19	20	21	23	25	27	29	31
A			✓		✓		✓	✓	✓	✓	✓	✓
B	✓	✓									✓	✓
C									✓			
D						✓	✓					
E				✓	✓							
F	✓											
G				✓								
H			✓									

→ Don't care use to find min terms as prime implicants and prime implicant chart doesn't contain don't care.

function involving functions

Ex 1 : Consider the 3-variable functions

$f_1(A,B,C) = \Sigma(0,7) + \Sigma\phi(1,2,5)$

$f_2(A,B,C) = \Sigma(0,1,3) + \Sigma\phi(4,7)$

find $f_1 \cdot f_2$ and $f_1 + f_2$ when minimized.

A B C	f_1	f_2	f_1+f_2	$f_1 \cdot f_2$
0	1	1	1	1
1	ϕ	1	1	ϕ
2	ϕ	0	ϕ	0
3	0	1	1	0
4	0	ϕ	ϕ	0
5	ϕ	0	ϕ	0
6	0	0	0	0
7	1	ϕ	1	ϕ

→

OR	AND
$1 + \phi = 1$	$1 \cdot \phi = \phi$
$0 + \phi = \phi$	$0 \cdot \phi = 0$

C \ AB	00	01	11	10
0	1			
1	ϕ		ϕ	

→ $f_1 \cdot f_2$

$f_1 \cdot f_2 = A'B'$

C \ AB	00	01	11	10
0	1	ϕ		ϕ
1	1	1	1	ϕ

→ $f_1 + f_2$

$f_1 + f_2 = C + A'$

→ ~~Tricks~~

$f_1 \cdot f_2 = \Sigma(0) + \Sigma\phi(\quad)$

Note → Do these kind of questions using method only. No need to apply any tricks.

Ex : 2 Consider a new boolean operation '\$' defined as $A \$ B = A' + B$ then find $f_1 \$ f_2$.

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

(f_1)

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

(f_2)

A B C	f_1	f_2	$f_1 \$ f_2$
0	0	1	1
1	0	0	1
2	1	0	0
3	1	1	1
4	1	1	1
5	0	1	1
6	0	0	1
7	1	1	1

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

$f_1 \$ f_2 = C + A + \overline{B}$

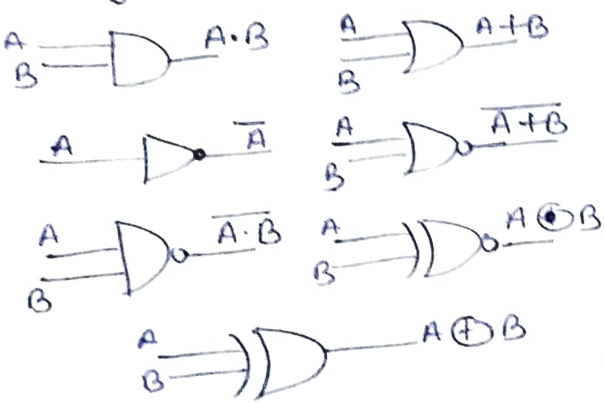
→ All three (A, B', C) are essential prime implicants.

Design and Synthesis of Combinational Circuits

Introduction to Logic Design

The main application of switching theory is in design of digital circuits. It is called as logic design.

→ These circuits are designed using basic elements called gates.



→ In DLD we consider only digital inputs.

→ During design we try to reduce the cost and propagation time (P_t).

→ We can reduce (P_t) by decreasing levels.

→ Gates uses in numbers –

① < 10 → Small Scale Integration

② 10 - 100 → Middle ——

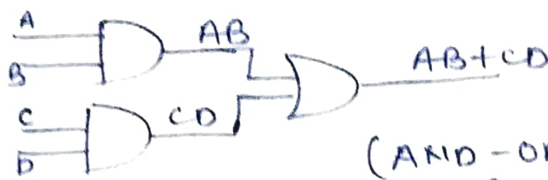
③ 100 - 1000 → Large ——

④ > 1000 → VLSI

(AND-OR), (OR-AND) realisation.

$f(A,B,C,D) = AB + CD$

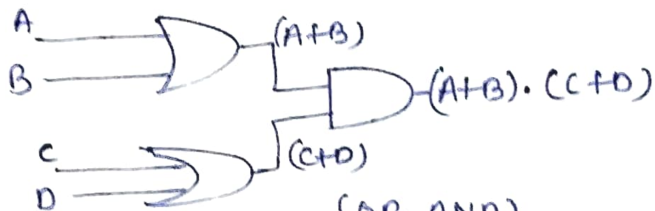
↳ SOP



(AND-OR) Realisation

POS

$f(A,B,C,D) = (A+B)(C+D)$



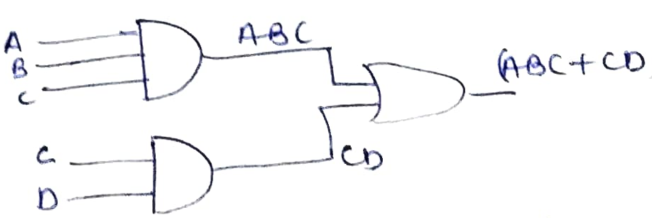
(OR-AND) Realisation

→ Any function can be represent in these two ways.

→ We can represent a function $f'(A,B,C,D)$ in two ways, such that

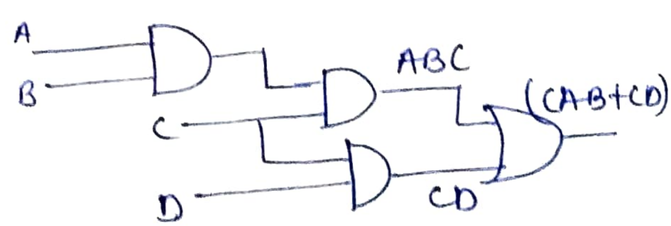
$f'(A,B,C,D) = ABC + CD$

①



→ 2 levels ckt

②



→ 3 level ckt

→ 2 level ckt is more costly but time is less and 3-level ckt is less costly but propagation time is high.

fan-In = 2 fan-In = 3

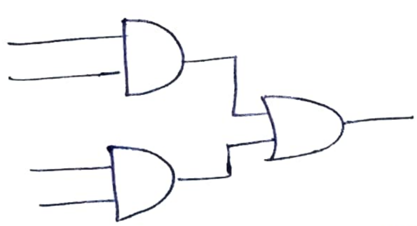
No. of level ↑ cost ↓ pt ↑

→ If don't specified anything Consider fan-in = 2.

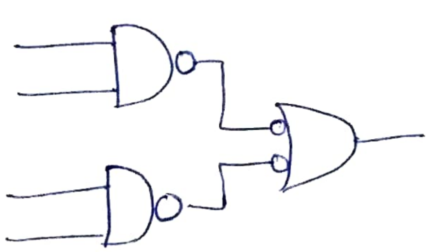
Q Why NAND and NOR are said to be universal gates?

Ans → Every function can be either in SOP or POS.

we represent SOP as (AND-OR) realisation and its logic ckt diagram is

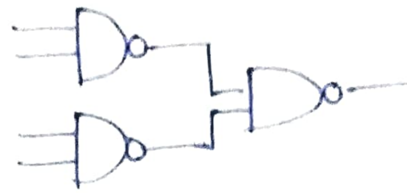


If we introduce complement (bubble) in fig as—



Both figures are same bcoz we are complement the digit at the o/p end of AND and at the i/p end of OR.

→ This diagram can be further reduce to,



So, (AND-OR) realisation also called as (NAND-NAND) realisation.

So, we can represent any SOP combination using NAND gate.

Similarly we can represent any POS combination / function using NOR gate.

→ So, to represent any fn we need only NAND and NOR Gate.

NAND → {AND, NOT} : S_1

NOR → {OR, NOT} : S_2

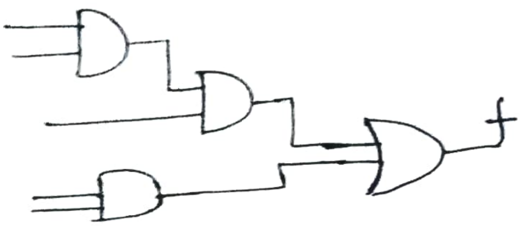
And we already prove earlier that both S_1 and S_2 are functionally complete.

→ (OR-AND) also called as (NOR-NOR) realisation.

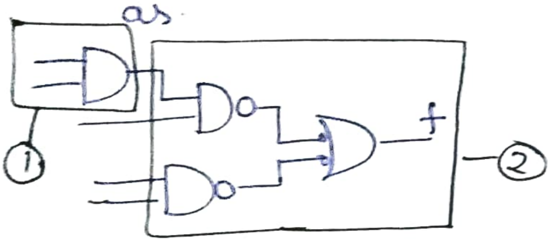
→ Realisation can be said as implementation.

→ Hence, we need to manufacture only NAND and NOR gate Instead of other gates.

Identify min no. of two i/p NAND gates required to represent the following



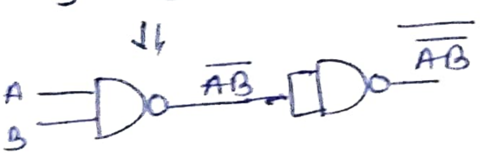
Sol^n Do some changes in (AND-OR) realisation as



So, ② part can implement using 3 NAND gate, and for ①

$(\overline{\overline{AB}}) = AB$

⇓

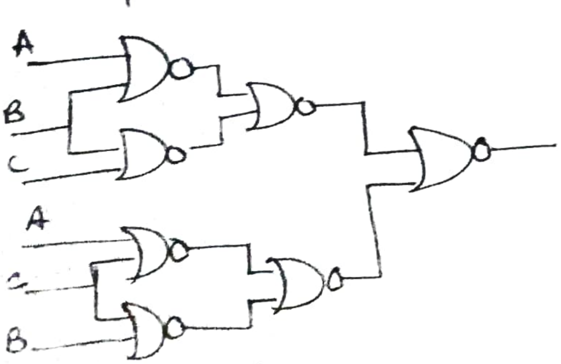


we need 2 NAND gates

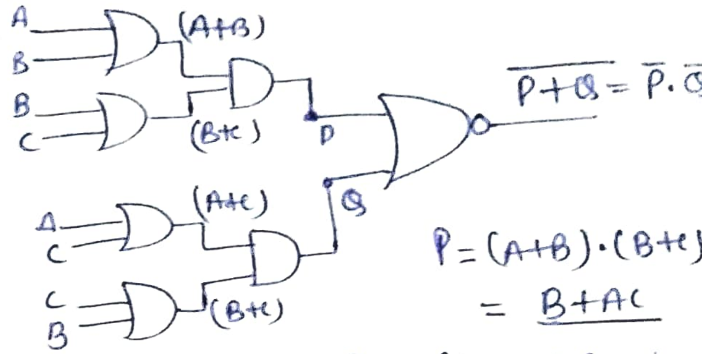
→ So we need total $(3+2) = 5$ NAND gates.

(NOR-NOR) Example

what does following f^n represents



Sol^n (NOR-NOR) is same as (OR-AND) realisation.



$P = (A+B)\cdot(B+C)$
$= \underline{B+AC}$

$Q = (A+C)(B+C)$
$= \underline{C+AB}$

$\bar{P}\cdot\bar{Q} = (\overline{B+AC})\cdot(\overline{C+AB})$

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

$= \cancel{(\bar{B}\bar{A}+\bar{B}\bar{C})\cdot(\bar{C}\bar{A}+\bar{C}\bar{B})}$

$\Rightarrow B\bar{A}C + \cancel{B\bar{A}C\bar{B}} + \cancel{B\bar{C}C\bar{A}} + \cancel{B\bar{C}C\bar{B}}$ $[C\bar{C}=0]$

wrong

$\Rightarrow (\bar{A}\bar{C}+\bar{C})(\bar{A}\bar{B}+\bar{B})$

$\Rightarrow$ $\bar{C}\bar{B}$ is the answer.

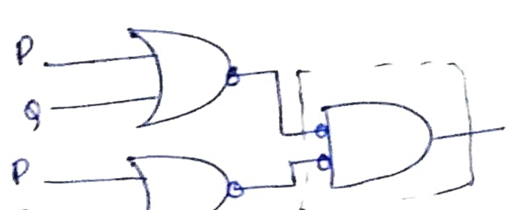
Q find the no. of 2 i/p 'NOR' gates require to represent

$F(P,Q,R) = P+QR$

Sol^n (NOR-NOR) realisation (OR-AND)

So, find POS form of $P+QR$ is

$(P+Q)\cdot(P+R)$



So, 3 NOR Gates are required.

Q Min. no of NOR gates required to implement

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

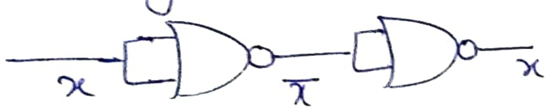
$$\Rightarrow x + x\bar{y}(1+z)$$

$$\Rightarrow x + x\bar{y} = x$$

So, here no 'NOR' gates are required to implement.

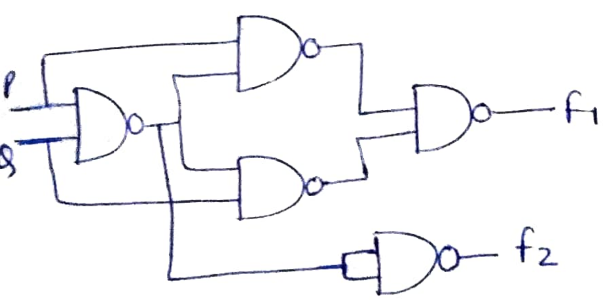
→ Convert the expression into minimal form first.

→ We can implement using following ckt.



But we have no need to this bcoz we require min. no. of NOR gates and its '0' bcoz we can simple give x as input and x as output on no gate ckt.

Half Adder.



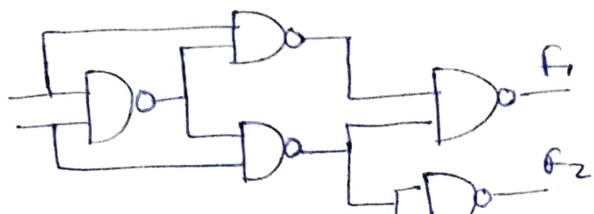
$f_1 = P \oplus Q$ (Sum)

$f_2 = PQ$ (Carry)

P	Q	f1	f2
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Half adder adds 2 bits and full adder adds 3 bits.

Half Subtractor



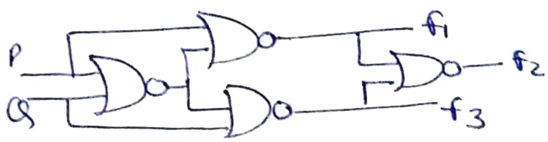
$F_1 = P \oplus Q$ (Subtract)

$F_2 = \bar{P}Q$ (Borrow)

P	Q	F1	F2
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

∘) 0 − 1 = 1 with borrow bcoz we can't 1 from 0 without borrow.

Comparator



$f_1 = \bar{P}Q$ $(P<Q)$

$f_2 = \bar{P}\bar{Q} + PQ$ $(P=Q)$

$f_3 = P\bar{Q}$ $(P>Q)$

P	Q	f2	f1	f3
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	1	1	0	0

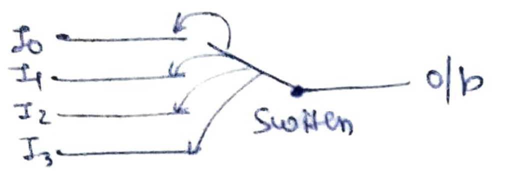
Multiplexer (MUX)

→ A MUX is an electronic circuit that can connect one out of 'n' inputs to output.

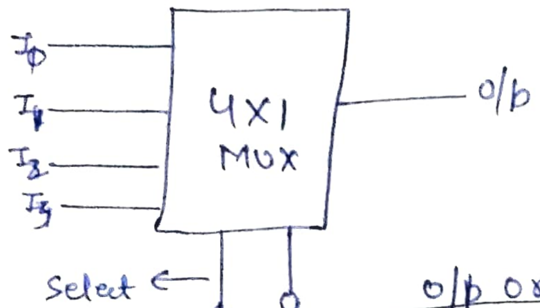
→ MUX is a kind of switch.

→ It can't change the logical level of the input, it only provides the connectn between I/p and o/p.

→ It is functionally complete, i.e all boolean functions can be realised using only multiplexer without any other gates.



You can connect switch with anyone of the i/p.



Characteristic table

S_1	S_0	o/p or Input selected
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

→ No. of select lines equals to 'n' if there are 2^n inputs.

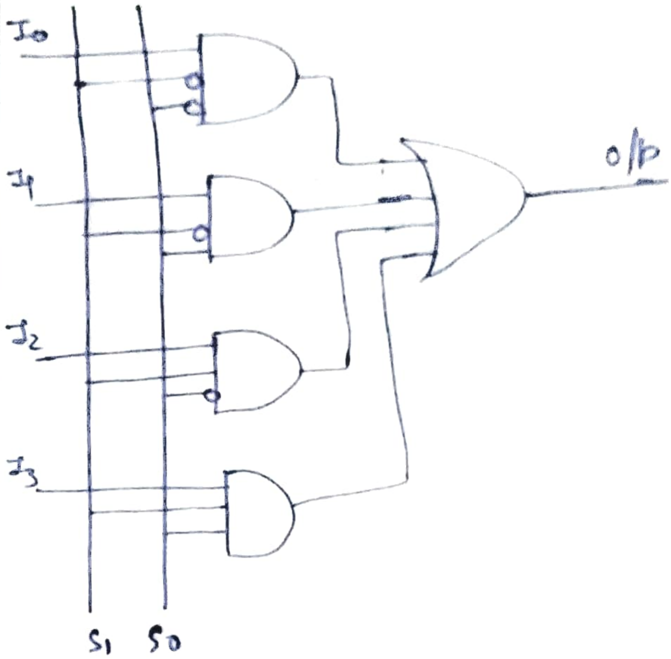
→ Characteristic equation

say o/p = f then,

$$f = \bar{S_1}\bar{S_0}I_0 + \bar{S_1}S_0I_1 + S_1\bar{S_0}I_2 + S_1S_0I_3$$

→ In ($2^n \times 1$) MUX inputs = 2^n and select lines = n

AND-OR realisation for characteristic equation.

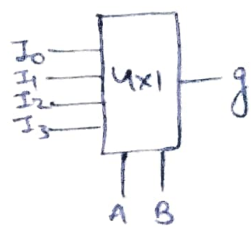


Note → MUX is functionally complete.

Implementation of function using MUX

Ex-1:

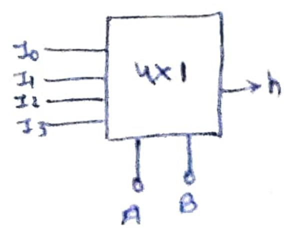
A	B	g
0	0	1
0	1	0
1	0	0
1	1	1



$$g(A,B) = \underbrace{\bar{A}\bar{B}(1)}_{I_0} + \underbrace{\bar{A}B(0)}_{I_1} + \underbrace{A\bar{B}(0)}_{I_2} + \underbrace{AB(1)}_{I_3}$$

Ex-2

A	B	h
0	0	1
0	1	0
1	0	1
1	1	0



If position A and B get interchange then input will get changed.

→ If you have ($2^n \times 1$) MUX then you can implement any function of n-variables.

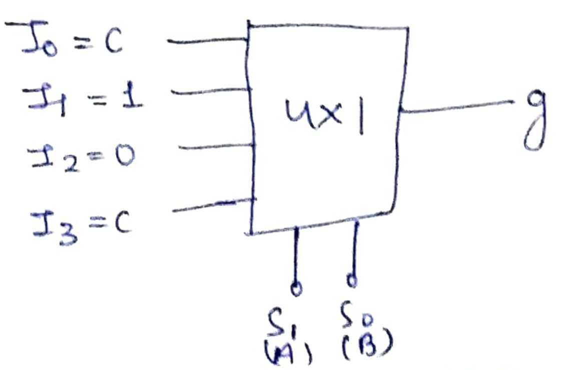
Can we implement (8×1) MUX using (4×1) MUX.

say any function.

A	B	C	g
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

$$g = \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + ABC$$

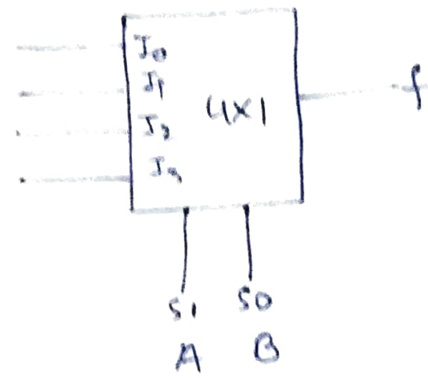
$$\Rightarrow \bar{A}\bar{B}\underbrace{(C)}_{I_0} + \bar{A}B\underbrace{(\bar{C}+C)}_{I_1} + A\bar{B}\underbrace{(0)}_{I_2} + AB\underbrace{(C)}_{I_3}$$



→ Using ($2^m \times 1$) MUX we can implement ($2^n \times 1$) MUX such that $m < n$ but you need more number of gates.

Ex-3

$$f(A,B,C) = \Sigma(1,2,4,6,7)$$

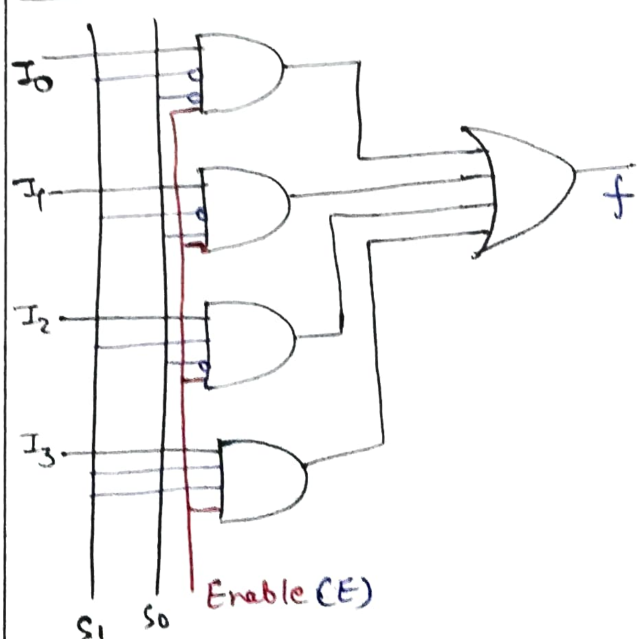


$$f = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + AB\bar{C} + ABC$$

$$\Rightarrow \underbrace{\bar{A}\bar{B}C}_{I_0} + \bar{A}B\underbrace{\bar{C}}_{I_1} + A\bar{B}\underbrace{\bar{C}}_{I_2} + AB\underbrace{(C+\bar{C})}_{I_3}$$

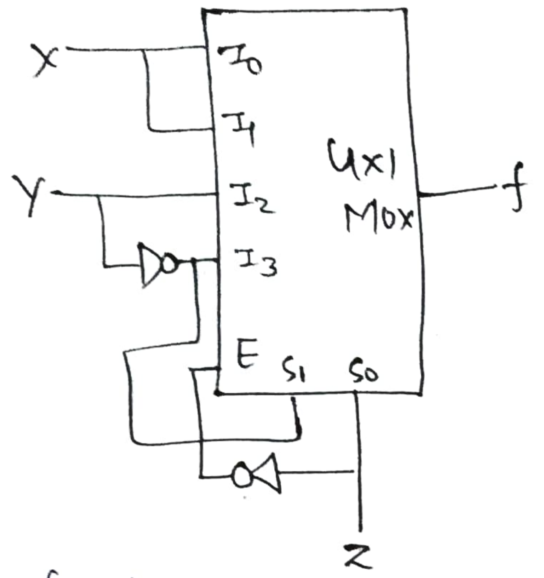
$I_0 = C$, $I_1 = \bar{C} = I_2$, $I_3 = 1$

MUX with enable Input



$$f = E(I_0\bar{S_1}\bar{S_0} + I_1\bar{S_1}S_0 + I_2 S_1\bar{S_0} + I_3 S_1 S_0)$$

Q Minimize the function represented by following MUX.



$f = E(I_0\bar{S_1}\bar{S_0} + I_1\bar{S_1}S_0$

$+ I_2 S_1\bar{S_0} + I_3 S_1 S_0)$

$\Rightarrow f = (XY\bar{Z} + XYZ$

$+ \cancel{Y\bar{Y}\bar{Z}}^{\,0} + \bar{Y}Z]\bar{Z}$

$\Rightarrow f = \bar{Z}(XY + \bar{Y}Z)$

$\Rightarrow \boxed{f = XY\bar{Z}}$

imp

Relation between select lines and input of MUX

$f(A,B,C) = \Sigma(2,3,5,6,7)$

a) $S_1 = B$, $S_0 = C$

b) $S_1 = C$, $S_0 = B$

Use 4x1 MUX only.

a) $f = \bar{A}B\bar{C} + \bar{A}BC +$

$A\bar{B}C + AB\bar{C} +$

ABC

$\Rightarrow \underbrace{(\bar{A}+A)}_{I_2} B\bar{C} + \underbrace{(\bar{A}+A)}_{I_3} BC$

$+ \underbrace{A}_{I_1}\bar{B}C + \underbrace{0}_{I_0}\bar{B}\bar{C}$

$I_0 = 0$, $I_1 = A$, $I_2 = I_3 = 1$

b) $S_1 = C$, $S_0 = B$

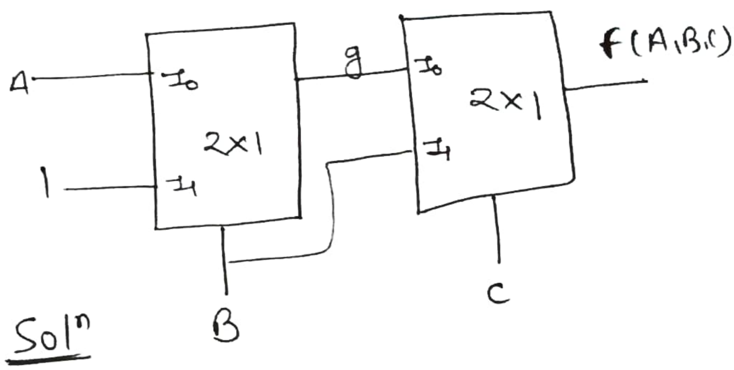
$I_0 = 0$, $I_1 = 1$, $I_2 = A$,

$I_3 = 1$

→ Input gets change according to select lines

Cascading MUX

Ex-1 What is the function in canonical SOP?



Soln

$g = A\bar{B} + B$

$f = A\bar{B}\bar{C} + B\bar{C} + BC$

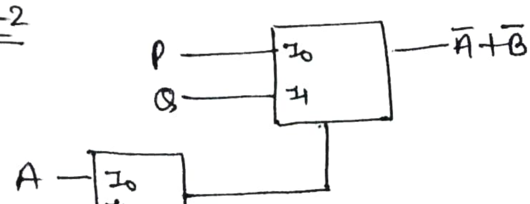
$\Rightarrow A\bar{B}\bar{C} + B\bar{C}(A+\bar{A}) + BC(A+\bar{A})$

$\Rightarrow A\bar{B}\bar{C} + AB\bar{C}\ \bar{A}B\bar{C} + ABC +$

$\bar{A}BC$

$f = \Sigma(4,6,2,7,3)$

Ex-2



find P, Q, R?

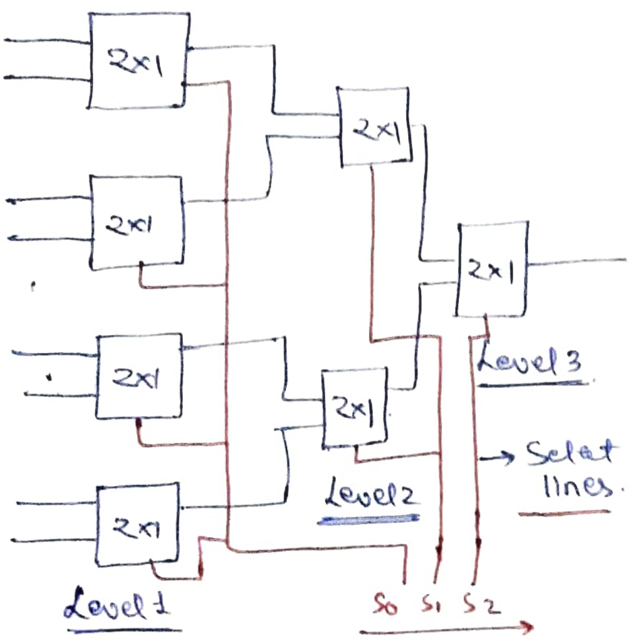
Sol^n $(\overline{RA+RB})P + (RA+RB)Q$

$\Rightarrow (R'+A')\cdot(R'+B')P + RAQ + RBQ$

$\Rightarrow RB'P + R'A'P + A'B'P + R'AQ$
$+ RBQ$

⇒ We didn't give the option here but option should be given in the question. It is better to put the option is the last expression and verify

Expansions of Multiplexers



→ To construct a 8x1 MUX using 2x1 we need 7 MUX and 3 levels.

→ To develop a Mx1 MUX using Nx1 MUX we need,

① no. of levels (k) $\geq \log_N M$

or

Remember $\boxed{k \geq \dfrac{\log_2 M}{\log_2 N}}$

→ In case of decimal values use ceil / floor function.

$k = \lceil \log_N M \rceil$
↳ In case of atleast

$k = \lfloor \log_N M \rfloor$ incase of atmost

Imp
② Total No. of devices require

$$D = \sum_{k=1}^{\lceil \log_2 M \rceil} (M/N^k)$$

Imp
③ At each level no. of devices require –

$$\left[\text{at level } k = \frac{M}{N^k} \right]$$

→ Try to derive each formula by self. So that you can remember the logic behind this.

Ex → Implement 32x1 using 4x1.

No. of level $= \lceil \log_4 32 \rceil = \lceil 2.5 \rceil$
$= 3.$

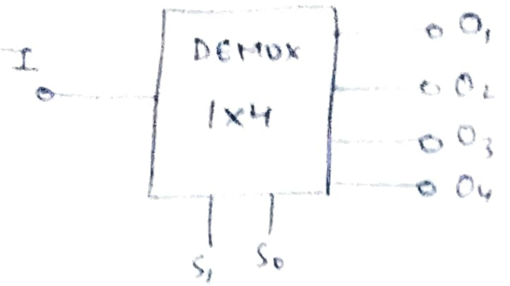
Device at levels

1 → $M/N = 8$
2 → $M/N^2 = 2$
3 → $M/N^3 = 1.$

No. of device require
$= 8+2+1$
$= 11$

Imp
Note → Order of giving values to select lines is also important.

Introduction to DEMUX



→ It performs opposite operation of MUX.

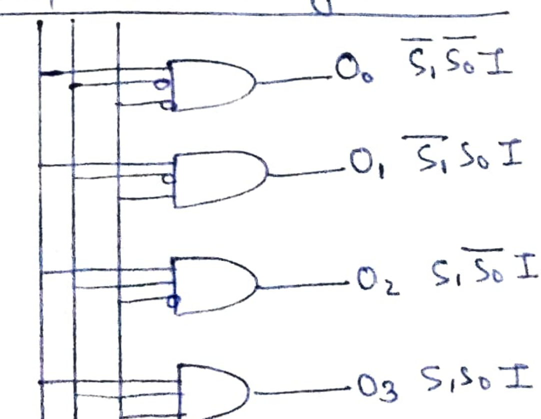
→ It has one input and 2^n outputs where 'n' is select lines.

→ It is derived from MUX by joining all the inputs together and removing 'OR' gate.

→ It is mainly used in the construction of switches.

→ DEMUX is used as receiving end and MUX is used at transmitting end.

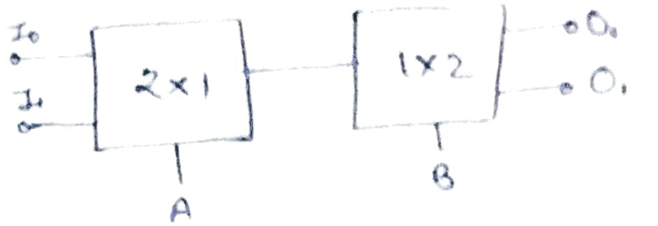
Implementation of DEMUX



S_1	S_0	O_0	O_1	O_2	O_3
0	0	I	0	0	0
0	1	0	I	0	0
1	0	0	0	I	0
1	1	0	0	0	I

Characteristic equation represents output in terms of inputs.

Ex → Switch using 2x1 MUX and 1X2 DEMUX



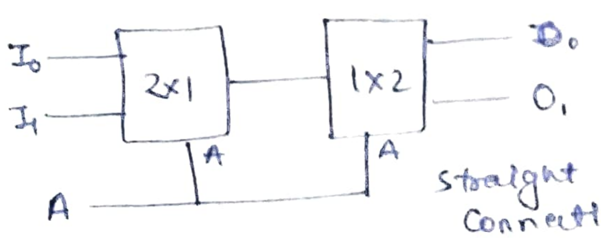
A	B	O_0	O_1
0	0	I_0	X
0	1	X	I_0
1	0	I_1	X
1	1	X	I_1

(SC) Straight Connection

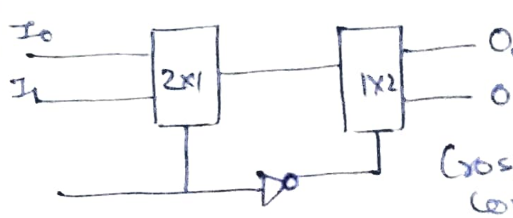
Cross Connection (CC)

SC: I_0 on O_0 or I_1 on O_1

CC: I_0 on O_1 or I_1 on O_0

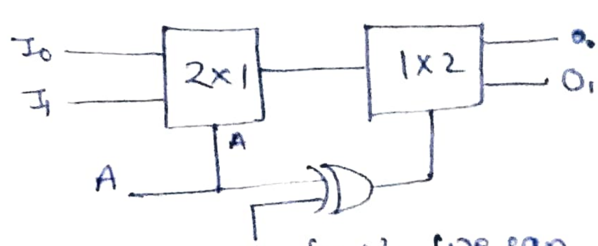


Straight Connection



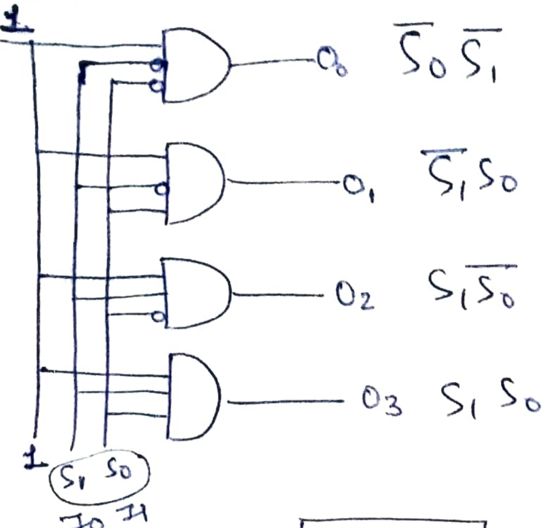
Cross Connection

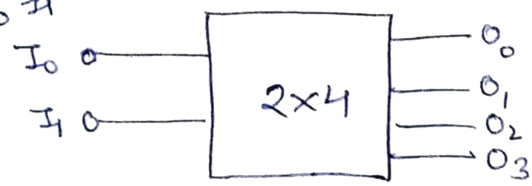
$A \oplus 0 = A$, $A \oplus 1 = \bar{A}$



$M = \{0,1\}$ We can implement both SC or CC

Introduction to Decoder





S_1	S_0	O_0	O_1	O_2	O_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

→ A decoder is a device gives 2^n outputs for n inputs. ($n \times 2^n$)

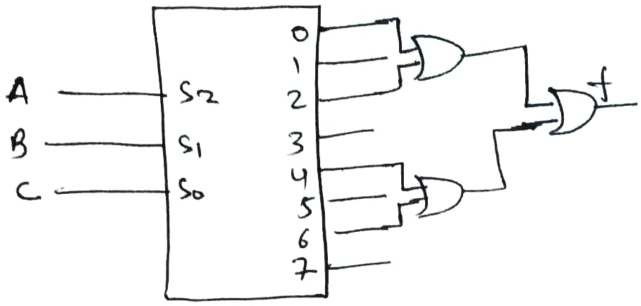
→ A demux can be converted to decoder by always setting $I=1$ and making select lines at input.

→ Since a decoder provides all the minterms, we could implement any function in canonical SOP using "OR" gate.

→ If all the 'AND' gates are replaced with 'NAND' gates, the decoder becomes active low.

→ In active low decoder we can implement any function in canonical SOP using NAND Gate.

Ex → 1



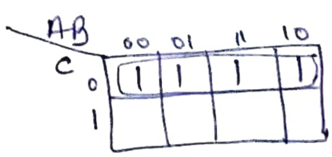
f is free from

a) 1 variable   b) two variable

c) 3 variable   d) None

Sol^n

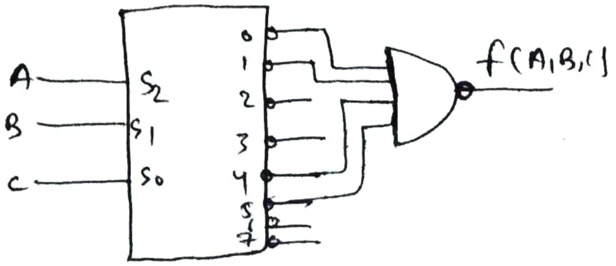
$f = \Sigma(0,2,4,6)$



$f = \bar{C}$ independent of 2 variables.

→ A function with n-variables can be independent of all the n-variables if and only if all the cells of K-map is filled with 1 or all the cells are empty in case of SOP.

Ex-2



f(A,B,C) is free from

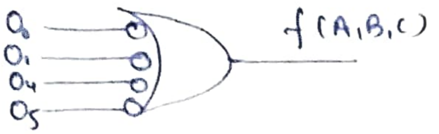
a) 1 variable b) 2 variable

c) 3 variable d) 4 variable

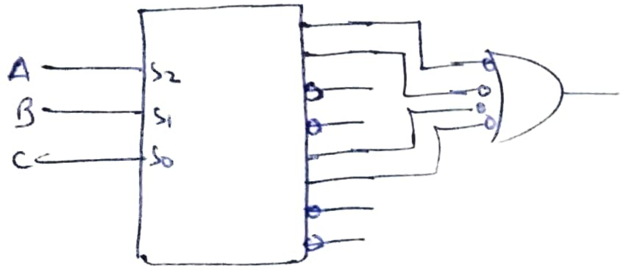
Sol^n Given decoder is active low.

Say at O_0 input = $\bar{a}\bar{b}\bar{c}$ and O/P = $(a+b+c)$

→ Replace NAND with complemented OR Gate as



If you use complemented 'OR' gate then changes in fig should be like,



$f = \Sigma(0,1,4,5)$

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

$f = \bar{B}$ independent of 2-variables (A, C).

Converting one code to another using decoder.

Convert (8 4 2 1) Code to (2 4 2 1) Code.

goto Note ;-

	8 A	4 B	2 C	1 D	2 W	4 X	2 Y	1 Z
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	0
3	0	0	1	1	0	0	1	1
4	0	1	0	0	0	1	0	0
5	0	1	0	1	1	0	1	1
6	0	1	1	0	1	1	0	0
7	0	1	1	1	1	1	0	1
8	1	0	0	0	1	1	1	0
9	1	0	0	1	1	1	1	1

Note → Just rember the method of conversion, from here and after this read about the (8421) and (2421) code from book.

Sol^n

$W = \Sigma(5,6,7,8,9) + \Sigma\phi(10-15)$

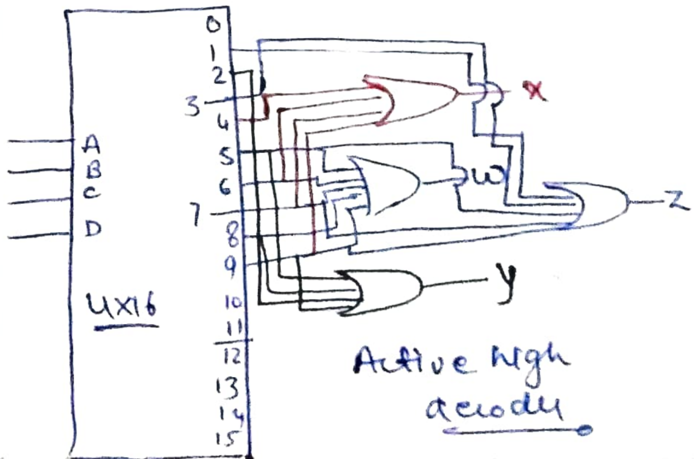
$X = \Sigma(4,6,7,8,9) + \Sigma\phi(10,11,12,13,14,15)$

$Y = \Sigma(2,3,5,8,9) + \Sigma\phi(10,11,12,13,14,15)$

$Z = \Sigma(1,3,5,7,9) + \Sigma\phi(10,11,12,13,14,15)$

→ Don't cares are same for all.

for 9 i/p we need 4 variable so, we need 4x16 decoder.



Active high decoder.

ROM using decoder

→ ROM can be used to realize combinational functions by storing appropriate values at appropriate locations.

→ Every ROM is expressed in terms of ROM matrix and decoder.

→ ROM matrix contains set of links and connections. The lines entering the matrix and leaving it are called connection and the interaction of rows and columns are called as links.

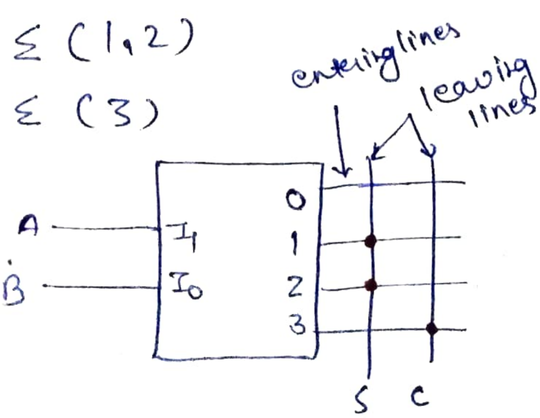
→ Decoder is a 'AND' realisation and we can implement a function using 'OR' Gate to perform (AND-OR) realisation.

Ex → Half Adder using connection and links

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$S = \Sigma(1,2)$

$C = \Sigma(3)$



→ 6 lines are connections

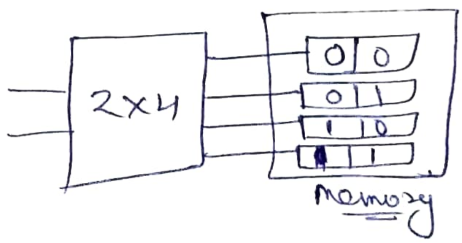
→ 3 dots are links.

→ Complete combination called as ROM.

→ In ROM all the links are pre programmed if ROM get burns or destroyed all links get break down and it will not again fix down. So, ROM is not re-programmable

→ ROM contain two things
1. Decoder
2. ROM matrix.

→ ROM is a m/m bcoz it needs to remember about all the established links.



→ Horizontal lines chooses the address and vertical line selects the particular bit.

Use of decoder in implementing functions

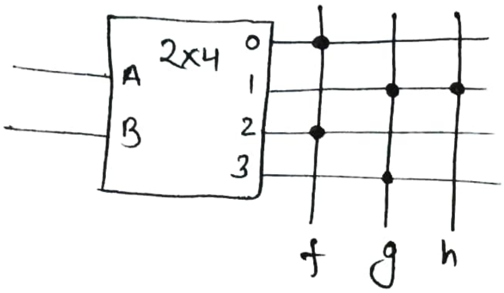
→ Each vertical line represent a function.

To implement a function with n-variable we need of decode of size $\boxed{(n \times 2^n)}$

If no. of function to implement is m then the total no. of connections (lines) required = $\boxed{2^n + m}$ and

no. of links = $\boxed{2^n \times m}$

Ex :



$f(A,B) = \Sigma(0,2)$

$g(A,B) = \Sigma(1,3)$

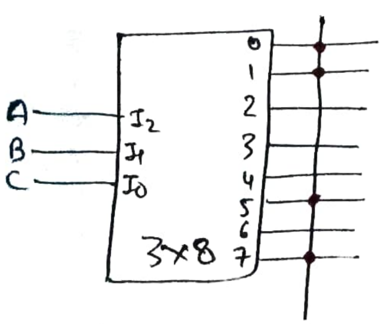
$h(A,B) = \Sigma(1)$

- No. of connection = 7 = 4+3
- —— links = 4x3 = 12

If M function in N variable are need to implement then no. of links or size of ROM matrix is equals to $(2^N \times M)$

Implementing function using Decoder + MUX

Ex-1 How to decrease no. of connections in ROM matrix?

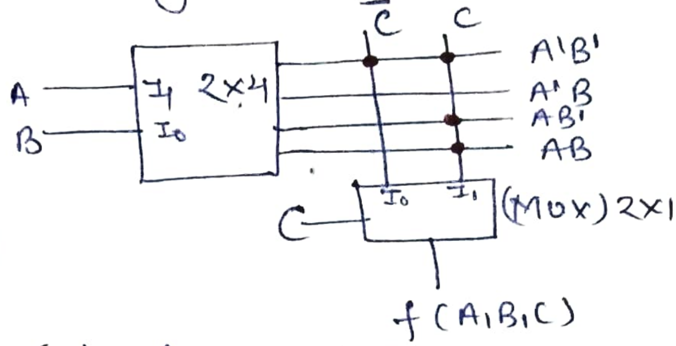


Here, total 9 connections and 8 links are present.

$f(A,B,C) = \Sigma(0,1,5,7)$

$\rightarrow f = A'B'C' + A'B'C + AB'C + ABC$

Lets implement the same fn using 2x4 decoder

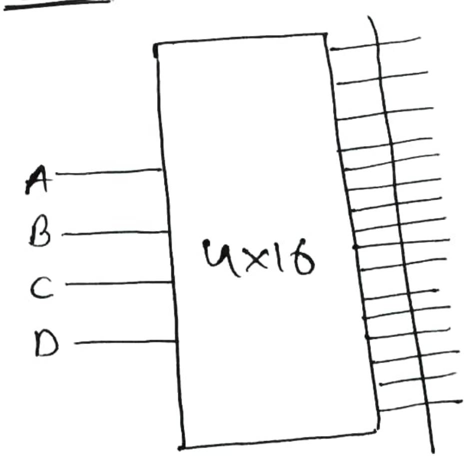


So, in above circuit we can implement the f(A,B,C).

Here we need 6 connections and 8 links.

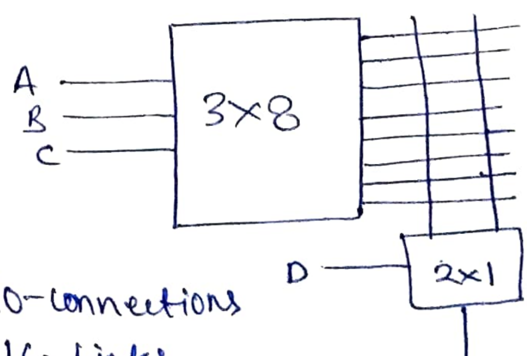
Imp → So, we can say that no. of connection can reduce but not links.

Ex-2



17 Connections and 16 Links.

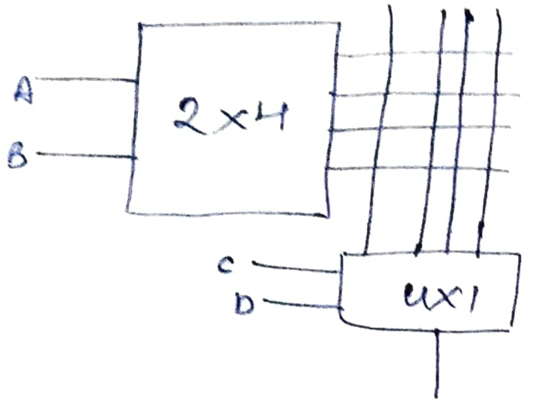
① Combination.



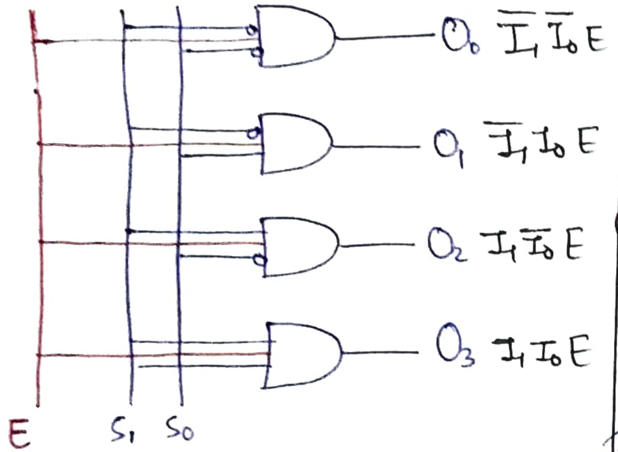
10-connections
16-Links

② Connection

8 connection
16 Links.



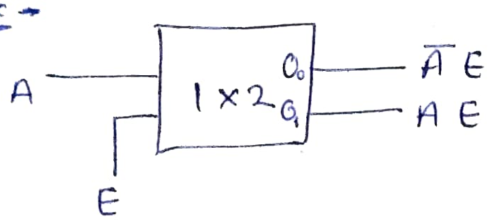
Decoder with enable input



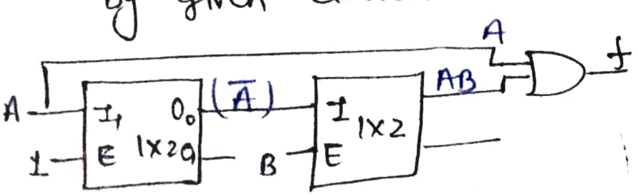
Instead of giving E = 1 we can make it enable and give control to user.

Using enable signal we can expand the decoder.

Ex →



Q find the function represented by given circuit.

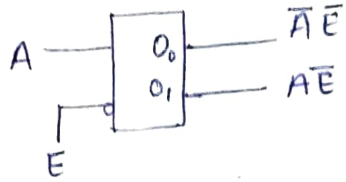


Solution

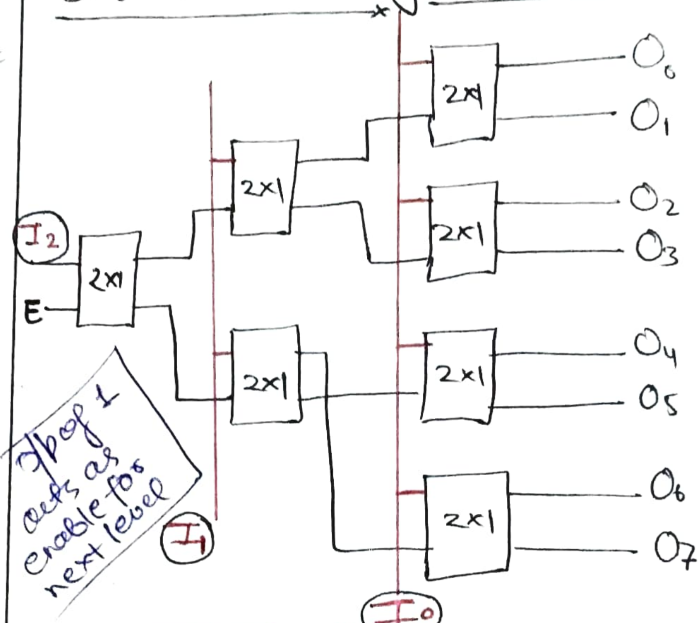
$f = AB \cdot A$
$= A(1+B)$
$= A$

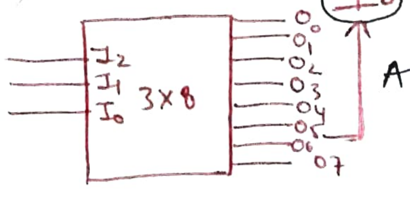
Enable को Direct multiply करते हैं

→ Enable can be active high or active low.



3x8 Decoder using 1x2 Decoder





Above diagram.

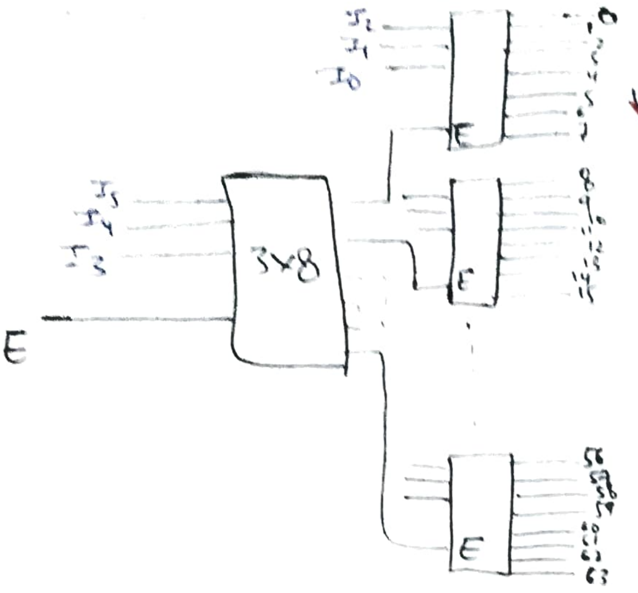
I_2	I_1	I_0	O/P
0	0	0	O_0
0	0	1	O_1
⋮	⋮	⋮	⋮
1	1	1	O_7

] function table.

Note: Must do it by self.

Construct a 2x4 decoder using 1X2 and construct its function table.

6x64 Decoder using 3x8



8 Decoders

Expansion of decoder in general

We want to construct $m \times 2^m$ decoder using $n \times 2^n$ decoder or $(\log m \times m)$ decoder using $(\log n \times n)$ decoder. then,

no. of levels required (k) is;

$$\boxed{k \geq \log_n m}$$

No. of device required (D) is;

$$\boxed{D = \sum_{k=1}^{\log_n m} \left(\frac{m}{n^k}\right)}$$

Ex → Construct a 6x64 decoder using 4x16.

No. of levels $= \left\lceil \frac{\log 64}{\log 16} \right\rceil = \lceil 3/2 \rceil = 2$

No. of decoders $= \frac{64}{16} = 8$ level 1

At level 2 $= \left\lceil \frac{8}{16} \right\rceil = 1$

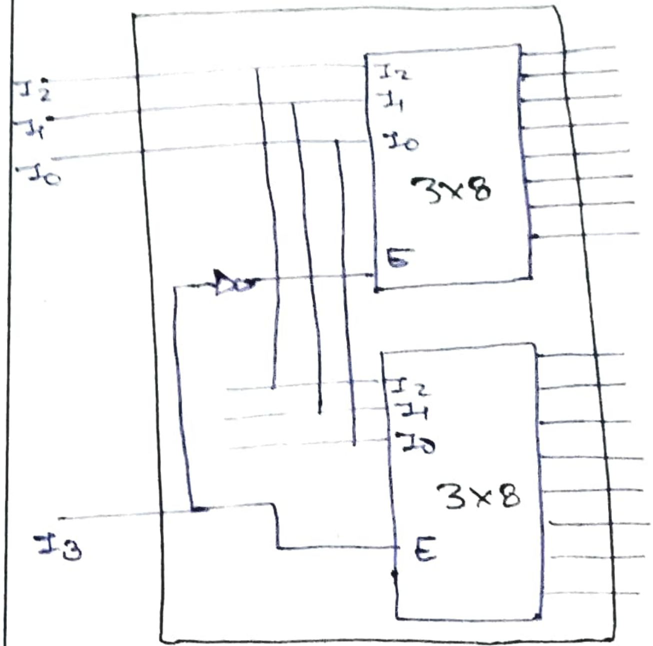
Total 9 decoders required

Assignment: To construct a 7x128 decoder using 3x8 decoders

(Wrong)

Expansion of Decoder in another way:

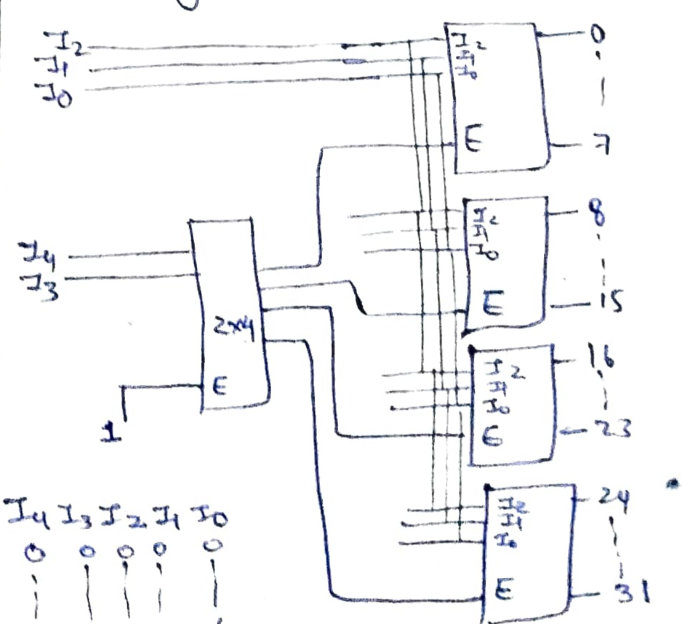
Construction of 4x16 using two 3x8 decoders.



4x16 decoder

I_3	I_2	I_1	I_0	O/P
0	0	0	0	O_0
0	0	0	1	O_1
⋮	⋮	⋮	⋮	⋮
1	1	1	1	O_{15}

Construct a 5x32 decoder using 3x8 decoders and 2x4



I_4	I_3	I_2	I_1	I_0
0	0	0	0	0
⋮	⋮	⋮	⋮	⋮
1	1	1	1	1