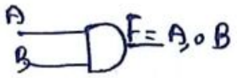
Dodla

Digital Logic Design.

— logic gates —.

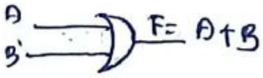
Main

AND



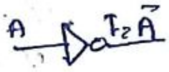
A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

OR



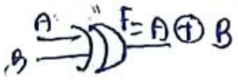
A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

Not



A	F
0	1
1	0

XOR



A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

Derived

- NAND
- NOR
- X-NOR

Logic Circuit

main difference of combinational Circuit & Sequential Circuit is,
Combinational - output only depent on present input
Sequiential - output ~~only~~ depend on Previous & current input

Combinational

* use ~~all~~ logic gates
* combinational circuit dependence on input & output variable.

Sequential

* The memory elements are circuit capable of storing binary information
* The binary information store In these memory element at any given time define the State of the Sequential circuit at that time
* The external output of a Sequential circuit depends both on the present input and the previous output state.

Steps of Design a combinational circuit.

01. truth table
02. Minimization.
03. Logic Diagram

Boolean Algebra

axioms.

$0+0=0$

$0\cdot0=0$

$1+1=1$

$1\cdot1=1$

$0\cdot1=1\cdot0=0$

$1+0=0+1=1$

$x=0;\ \bar{x}=1$

Single variable theorom.

$x\cdot0=0$

$x+1=1$

$x\cdot1=x$

$x+0=x$.

$x\cdot x=x$

$x+x=x$

$x\cdot\bar{x}=0$

$x+\bar{x}=1$

$\bar{\bar{x}}=x$

commalative low.

$x\cdot y=y\cdot x$

$x+y=y+x$.

associative low.

$x\cdot(y\cdot z)=(x\cdot y)z$

$x+(y+z)=(x+y)+z$.

Distributive low:

$x\cdot(y+z)=xy+xz$

$x+y\cdot z=(x+y)\cdot(x+z)$

Absorption.

$x+x\cdot y=x$.

De Morgan's thearom

$\overline{x \cdot y} = \bar{x} + \bar{y}$

$\overline{x + y} = \bar{x} \cdot \bar{y}$

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

$x \cdot (\bar{x} + y) = x \cdot y.$

Proof

x	y	$\bar{x}$	$\bar{y}$	$x \cdot y$	$\overline{x \cdot y}$	$x+y$	$\overline{x+y}$	$\bar{x}+\bar{y}$	$\bar{x} \cdot \bar{y}$
0	0	1	1	0	1	0	1	1	1
0	1	1	0	0	1	1	0	1	0
1	0	0	1	0	1	1	0	1	0
1	1	0	0	1	0.	1	0.	0	0.

$\therefore \overline{x \cdot y} = \bar{x} + \bar{y}$

$\overline{x + y} = \bar{x} \cdot \bar{y} \quad /_2.$

Consensus.

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

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

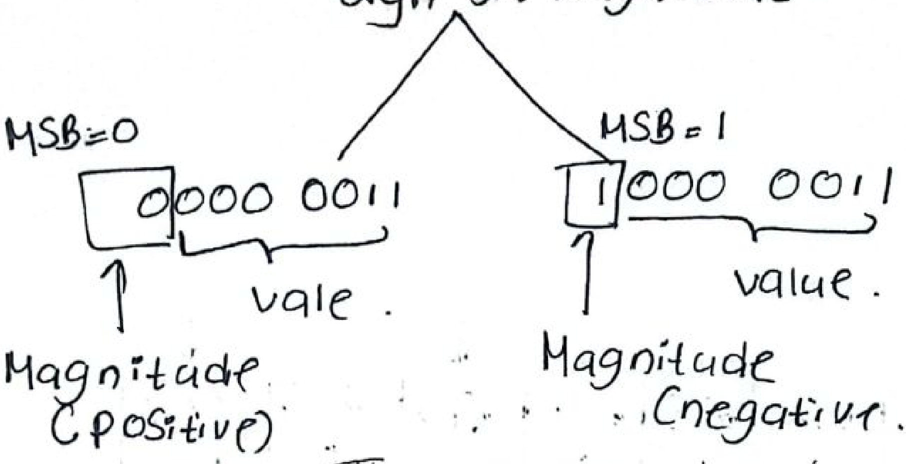
Sum of product → $x \cdot y + y \cdot z + \ldots$

Product of Sum → $(x+y) \cdot (y+z) \cdot \ldots$

Number System..

01. Binary number System...
02. Decimal number System
03. Octal number System
04. Hexa-Decimal number system.

Sign bit Magnitude:



01. The way of Showing negtive . numbers

1's complement

ex: +9 → 0000 1001.

1's complement → 1111 0110.

∴ -9 → 1111 0110.

2's complement

2's complement = 1's complement + 1.

ex +9 → 0000 1001

1's c → 1111 0110.

+ 1

2's c → 1111 0111.

∴ -9 → 1111 0111

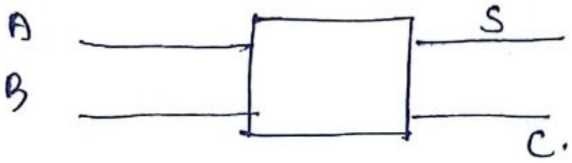
Half Adder ·

*Two inputs → A, B

* Two Output

Sum = $A \circ \bar{B} + \bar{A} B$ (XOR)

Carry . = $A \circ B$.

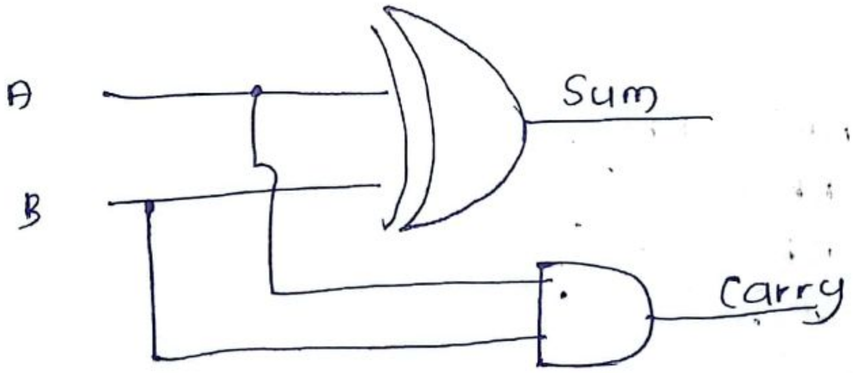


truth table .

A	B	Sum	Carray
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$A \circ B$.

$A \circ \bar{B} + \bar{A} \circ B$
(XOR)



Full Adder -

* Three inputs
* Two 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

using SOP ~~B~~ fa

Sum

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

$= \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC.$

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

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

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

Carry

$\bar{A}BC + A\bar{B}C + AB\bar{C} + ABC.$

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

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

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

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

$= \bar{A}BC + AB + AC.$

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

$=$ ~~AB + A AB + AC + AC~~. $AB + BC + AC$

using K-map

A\BC	00	01	11	10
0	0	1	0	1
1	1	0	1	0.

no groups

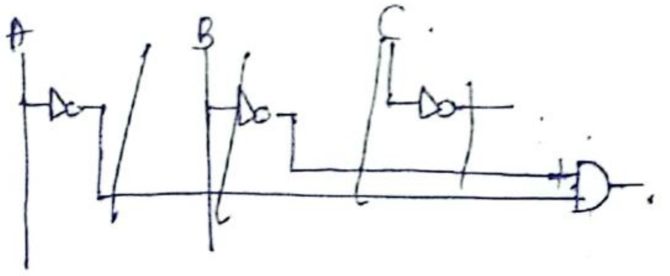
∴ $\text{Sum} = \bar{A}\bar{B}C + AB\bar{C} + A\bar{B}\bar{C} + ABC.$

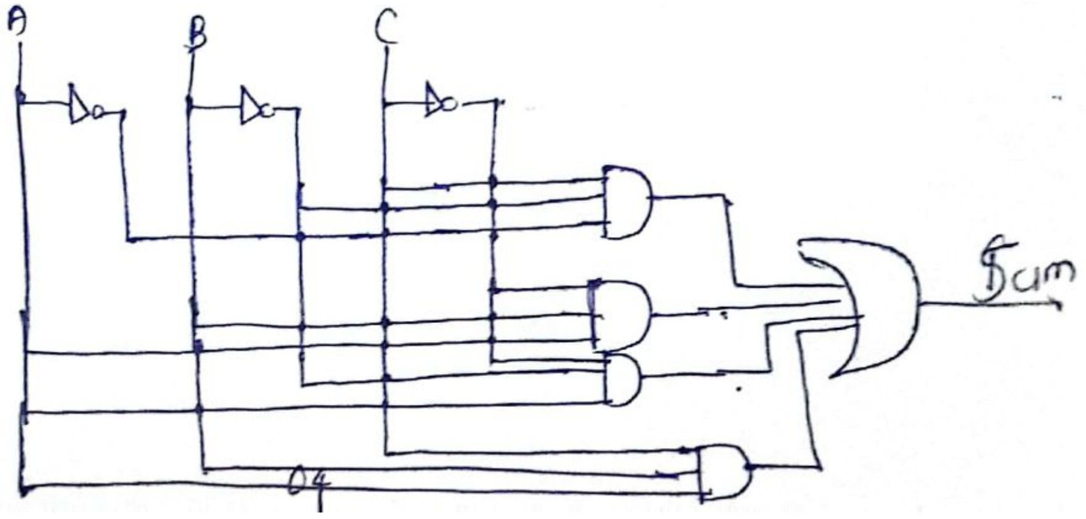
$= C(\bar{A}\bar{B} + AB) + \bar{C}(A \oplus B).$

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

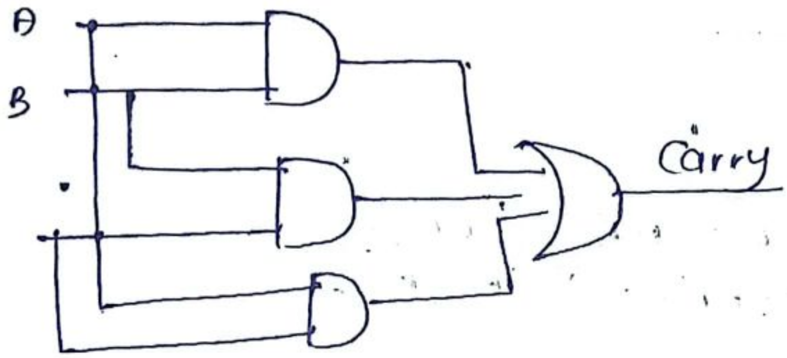
$\text{Carry} = AB + AC + BC$

Logic Gate for Sum.



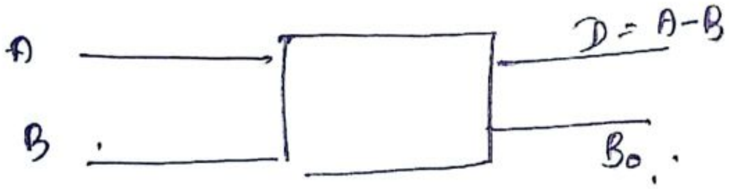


Logic gate for carry



Half Substractor

* use to Substract one binary digit from another
* two output — Different
  — Borrow.



A	B	D	B
0	0	0	0.
0	1	1	1
1	0	1	0.
1	1	0	0.

full Substractor

A	B	Bin	D	Bo
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Control Inventor

Multiplexer

It is a combinational circuit that select binary information from one to mary input line and direct it to output line.

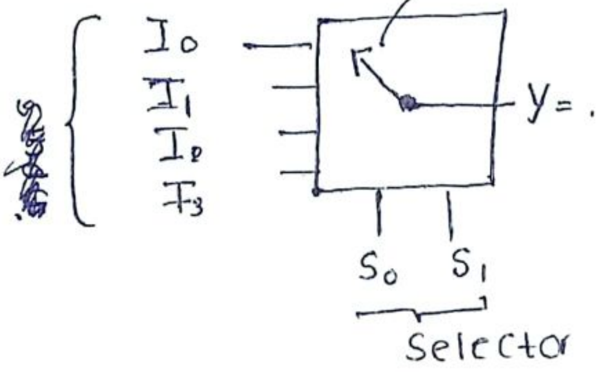
In multiplexer always have one output.

Advantageous

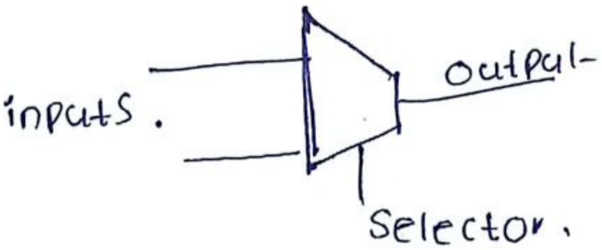
- Reduce number of wires.
- Reduce circuit complexity & cost.
- Implementation of various circuit using multiplexer.

Symbol

can move $I_0 \rightarrow I_3$ for Y.



I	S_1	S_2	Y
I_0	0	0	$Y = I_0$.
I_1	0	1	$Y = I_1$
I_2	1	0	$Y = I_2$
I_3.	1	1	$Y = I_3$.



number of Selector variable → m,

$$m = \log_2 n .$$

$$n = 2^m$$

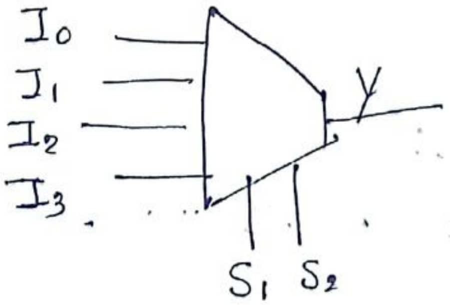
n = number of input varia-ble.

ex : numper of input → 4,

$$m = \log_2 4$$

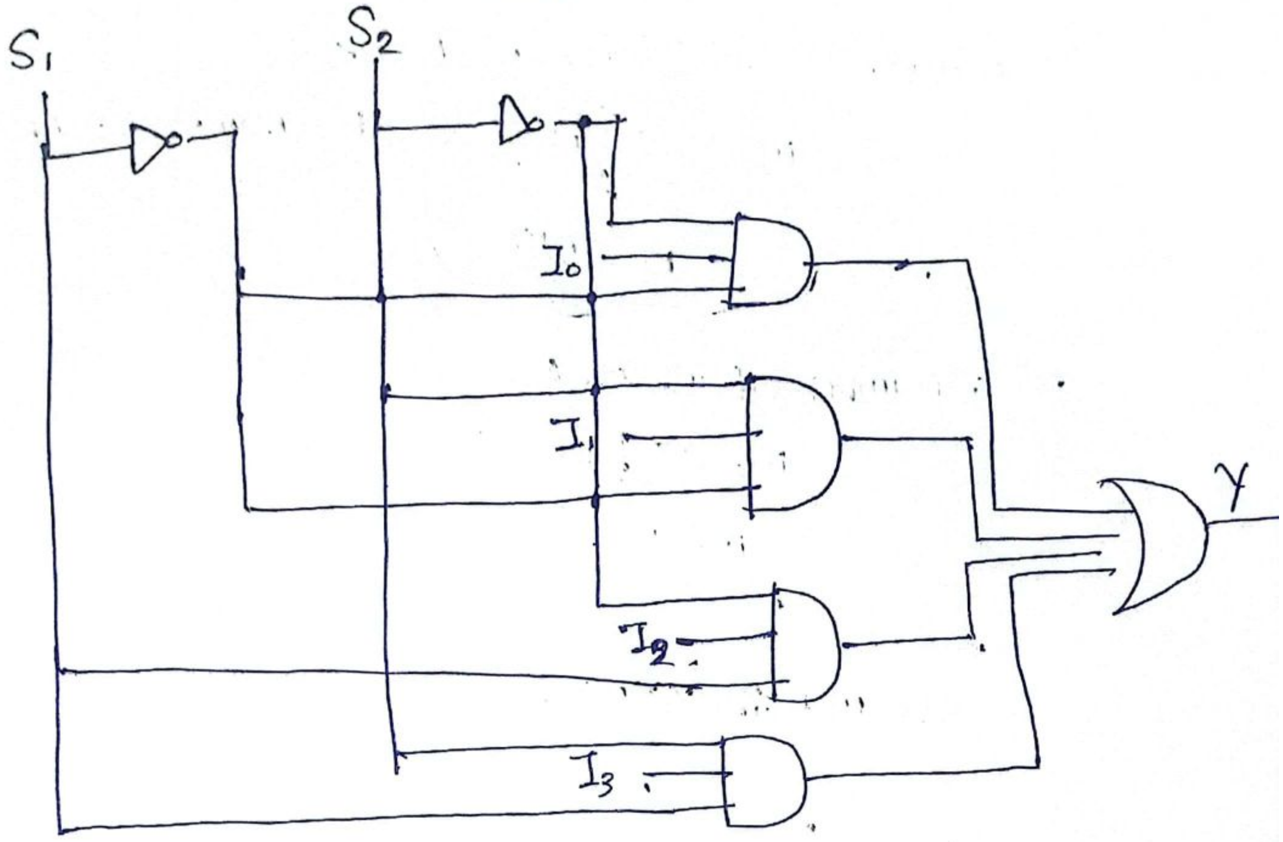
$$m = 2 .$$

4x1 multiplexer .



S_1	S_2	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

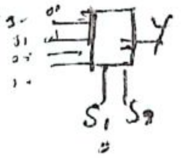
$$y = \bar{S_1}\bar{S_2}I_0 + \bar{S_1}S_2I_1 + S_1\bar{S_2}I_3 + S_1S_2I_3 .$$



Combinational Circuit

4x1 multiplexer, usin 2x1 multiplexer

truth table for 4x1.



S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

4x1..

n= input,= 4

$\frac{4}{2}$ (because we convert 2x1 multiplexer.

$\frac{4}{2} = 2$, (devide until get 01)

$+$ ③ · (2+1).

$\frac{2}{2} = 1$

this is number of 2x1 multiplexer we get

find m,

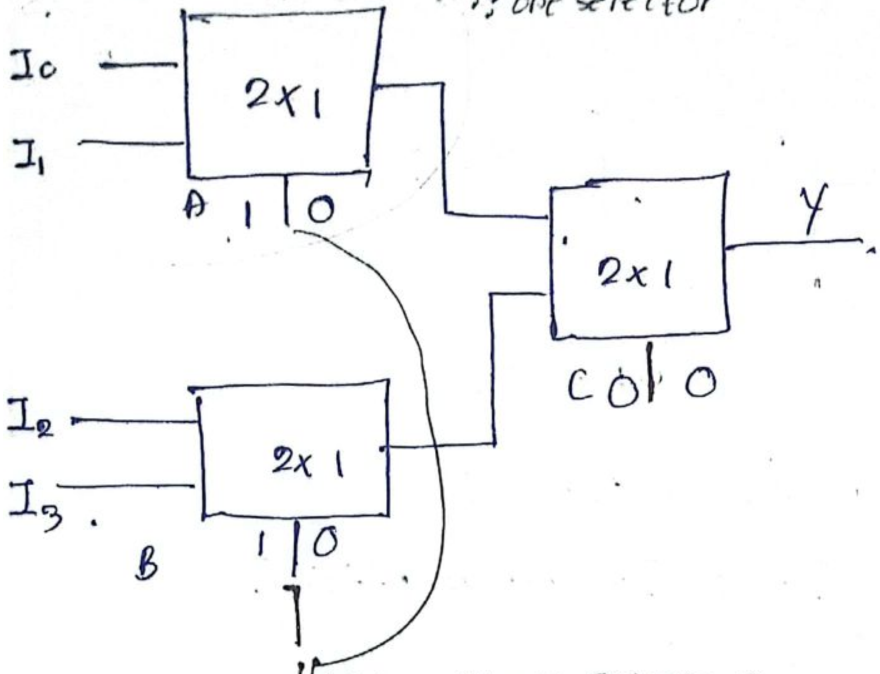
$m = \log_2 4$.

$\boxed{m = 2}$

here $n=2$

$m = \log_2 2$

$= 1$ ∴ one selector



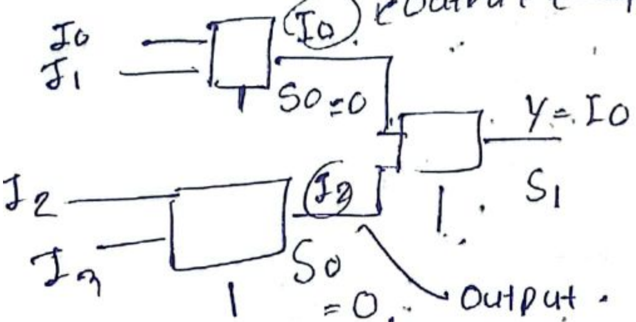
this two place selectors are same.

how to get selector value.

it get by 4x1 multiplexer's truth table,

think, If we wants y = I0, we can (0,0).

output for out I0 So Should be 0 (truth table).



output. because in truth table.

$I_2 \Rightarrow S_0 = 0$,

then, from I0 and I2,

we want to out I0,

according to truth table,.

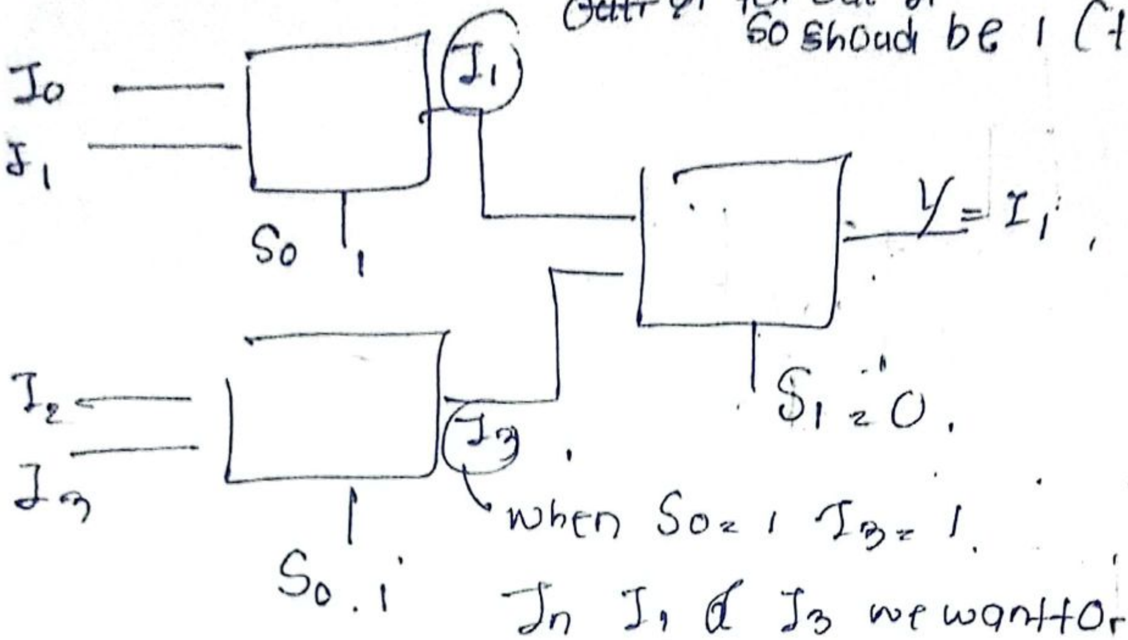
S_1	S_0	
0	0	I0
1	0	I2,

So we keep $S_1=0$.

(or).

think if we want ~~ou~~ want to set output $Y = I_1$.

~~Outr of~~ for out I_1 So should be 1 (truth table).



when $S_0 = 1$ $I_3 = 1$.

In I_1 & I_3 we want for I_1

∴ $\boxed{S_1 = 0.}$

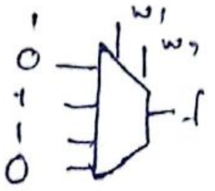
Combinational Circuit building blocks

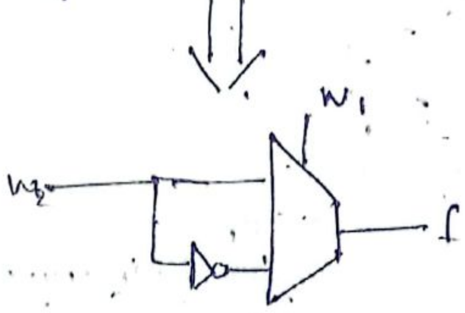
Example 01

w_1	w_2	f	
0	0	0	} w_2
0	1	1	
1	0	1	} $\overline{w_2}$
1	1	0	

⇒

w_1	f
0	w_2
1	$\overline{w_2}$





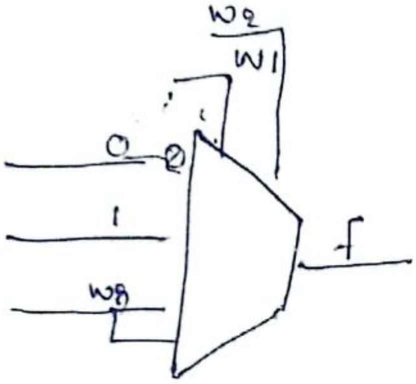
Example 02

8x1 ⇒ 4x1

w1	w2	w3	f	
0	0	0	0	0
0	0	1	0	
0	1	0	0	w3
0	1	1	1	
1	0	0	0	w3
1	0	1	1	
1	1	0	1	1
1	1	1	1	

⇒

w1	w2	f
0	0	0
0	1	w3
1	0	w3
1	1	1

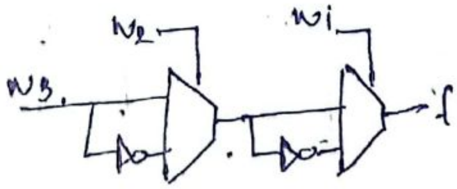


example 03.

8x1 ⟶ 2x1

w1	w2	w3	f	
0	0	0	0	$w_1 \oplus w_3$
0	0	1	1	
0	1	0	1	
0	1	1	0	
1	0	0	1	$\overline{w_2 \oplus w_3}$
1	0	1	0	
1	1	0	0	
1	1	1	1	

⇒



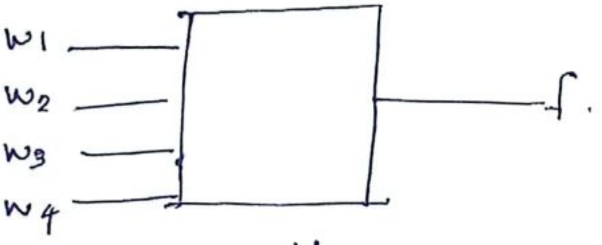
Shannon's Expansion Theorom.

Shannon's Expansion theorom express a given Logic function in tearm of select input and data input required for logic synthesis using a multiplexer.

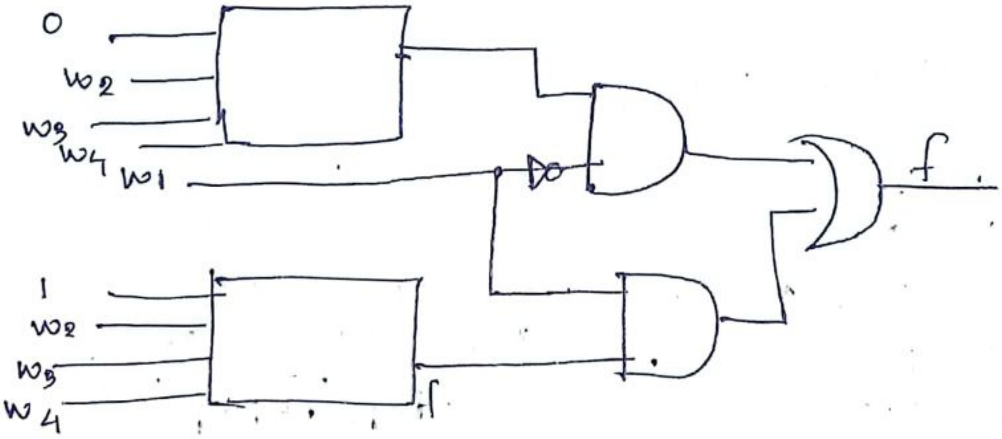
Selector variable

$f(w_1, w_2 -- w_n) = \bar{w_1} \cdot f(0, w_2, w_3 -- w_n) + w_1(1, w_2, -- w_n)$

ex:-



⇓



ex:- emplement $f = \bar{w_1}\bar{w_3} + w_1 w_2 + w_1 w_3$ using 2 to 1 multiplexer.

according to shannon's expression

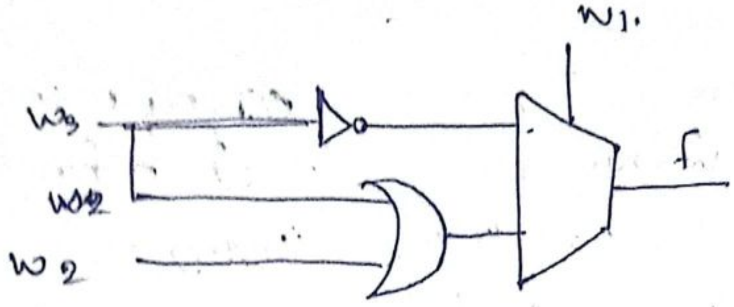
$f = \bar{w_1} f(0, w_2, w_3) + w_1 f(1, w_2 w_3)$

now create function,

$f = \bar{w_1}(1 \cdot \bar{w_3} + 0 \cdot w_2 + 0 \cdot w_3) + w_1(0 \cdot \bar{w_3} + 1 \cdot w_2 + 1 \cdot w_3)$

$= \bar{w_1}(\bar{w_3}) + w_1(w_2 + w_3).$

⇓



example 02

$f(w_1, w_2, w_3) = w_1w_2 + ~~w_1w_3~~ + w_2w_3$.

~~$= f\ \bar{w}_1 f(0, w_2, w_3) + w_1 f(1, w_2, w_3)$~~

~~$= \bar{w}_1(0 \times w_3 + 1 \cdot w_2$~~

~~$= \bar{w}_1(0 \times w_2 + 0 \cdot w_3 + 1 \times w_2 \cdot w_3) + w_1(1 \times w_2 + 1 \times w_3$~~

~~$f = \bar{w}_1w_3 + w_2w_3$ decomposition using~~ w_1 gives,

~~$= \bar{w}_1 f$~~

$f = w_1w_2 + w_1w_3 + \underline{w_2w_3}$. → *off fanction [illegible] $w_1, \bar{w}_1$ [illegible]

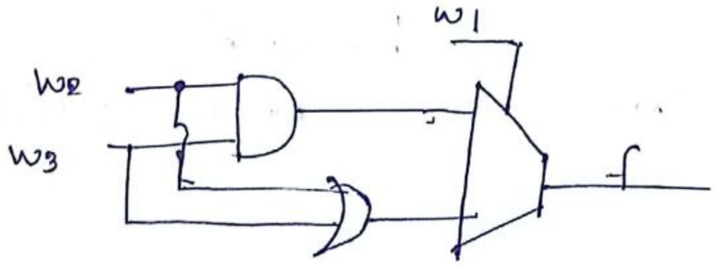
$= \bar{w}_1 f(0, w_2, w_3) + w_1(1, w_2, w_3)$.

$= \bar{w}_1(0 \times w_2 + 0 \times w_3 + w_2w_3) + w_1(1 \times w_2 + 1 \times w_3 + w_2w_3)$

$= \bar{w}_1(w_2w_3) + w_1(w_2 + w_3 + w_2w_3)$

$= \bar{w}_1(w_2w_3) + w_1(~~w_2w_3~~ \cdot w_2(1 + w_3) + w_3)$

$= \bar{w}_1(w_2w_3) + w_1(w_2 + w_3)$



Let simplyfy this only using multiplexer, let $g = w_2 w_3$

$= 0 + w_2 w_3$

$= \overline{w_2} \times 0 + w_3 w_2$

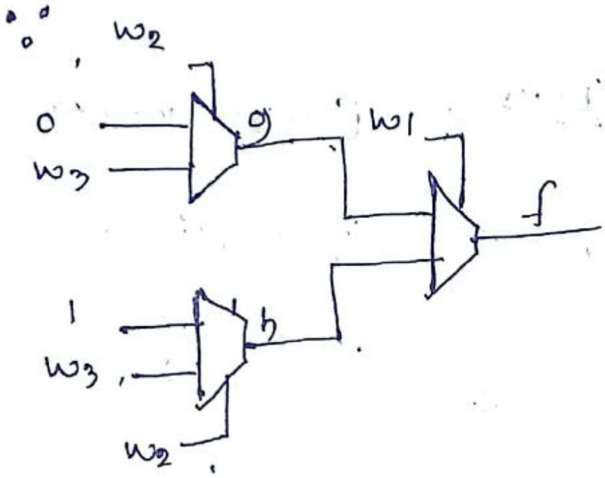
$= \overline{w_2}(0) + w_2(w_3)$

here selector is w_2, $/_2$.

$h = w_2 + w_3$

$= w_2 \times 1 + \overline{w_2}(w_3)$

expand both g & h function. using w_2 —Shannons Expansison Th



example

$f = \overline{w_1} w_3 + w_2 \overline{w_3}$ decomposition using w_1 gives.

$= \overline{w_1}(w_3 + w_2\overline{w_3}) + w_1(w_2\overline{w_3})$

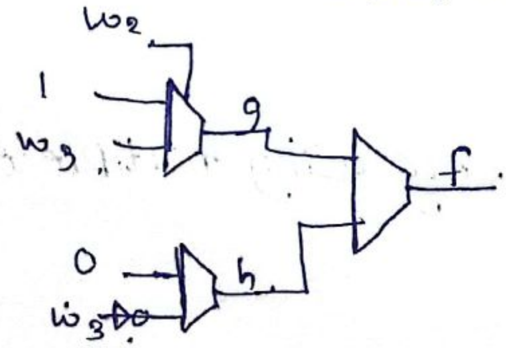
$= \overline{w_1}\underbrace{(w_2+w_3)}_{g} + w_1\underbrace{(w_2\overline{w_3})}_{h}$

$g = w_2 + w_3$

$= w_2 \times 1 + \overline{w_2}(w_3)$.

$h = w_2\overline{w_3}$

$= w_2\overline{w_3} + \overline{w_2}(0)$



(2x1 multiplexer).

if we want to . Show f usin 4x1 multiplexer, (w1,w

$f = \overline{w_1}w_3 + w_2\overline{w_3}$

$= \overline{w_1}\,\overline{w_2}(w_2\overline{w_3}) + \overline{w_1}w_2(w_2\overline{w_3})$
$+ w_1\overline{w_2}$

Changes of w1,w2
$\overline{w_1}\cdot\overline{w_2}$
$\overline{w_1}\,w_2$
$w_1\overline{w_2}$
$w_1 w_2$

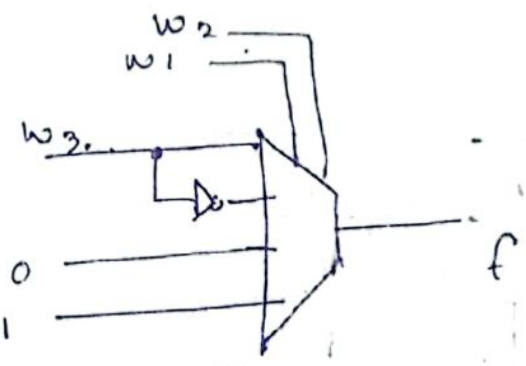
$f = \overline{w_1}w_3 + w_2\overline{w_3}$

~~$\overline{w_1}\,\overline{w_2}$ f(w2~~ $f = \overline{w_1}\,\overline{w_2} = f(w_1 = 0, w_2 = 0)$
$\overline{w_1}\,w_2 = f(w_1 = 0, w_2 = 1)$
$w_1\overline{w_2} = f(w_1 = 1, w_2 = 0)$
$w_1 w_2 = f(w_1 = 1, w_2 = 1)$

00 ඉඳලා
11 දක්වා
ගන්න

$= \overline{w_1}\,\overline{w_2}(w_3) + \overline{w_1}w_2(w_3 + \overline{w_3}) + w_1\overline{w_2}(0+0)$
$+ w_1w_2(\overline{w_3})$

$= \overline{w_1}\,\overline{w_2}(w_3) + \overline{w_1}w_2(1) + w_1\overline{w_2}(0) + w_1w_2(\overline{w_3}$



(this is not match with ser)

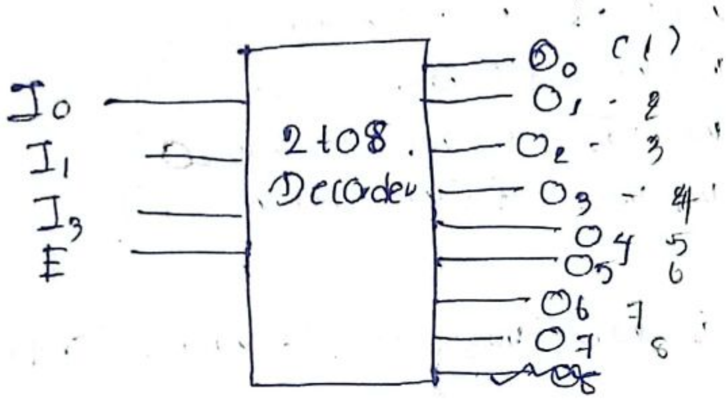
Example · 2 to 4 decoder.

Decoders

* ~~Decord~~ Decoder Circuit. are used to decode encoded information.

* Decoders are also combinational Circuit

* n input → 2^n out put.

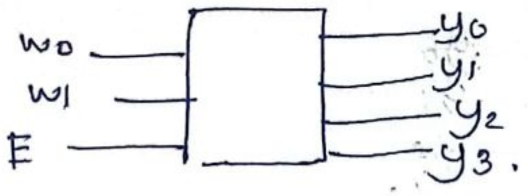


E call Enable pin, it is not relavent.
There are three input (I0, I1, I2.) lets make.
truth table.

← input →   ← output →

I0	I1	I0	O7	O6	O5	O4	O3	O2	O1	O0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

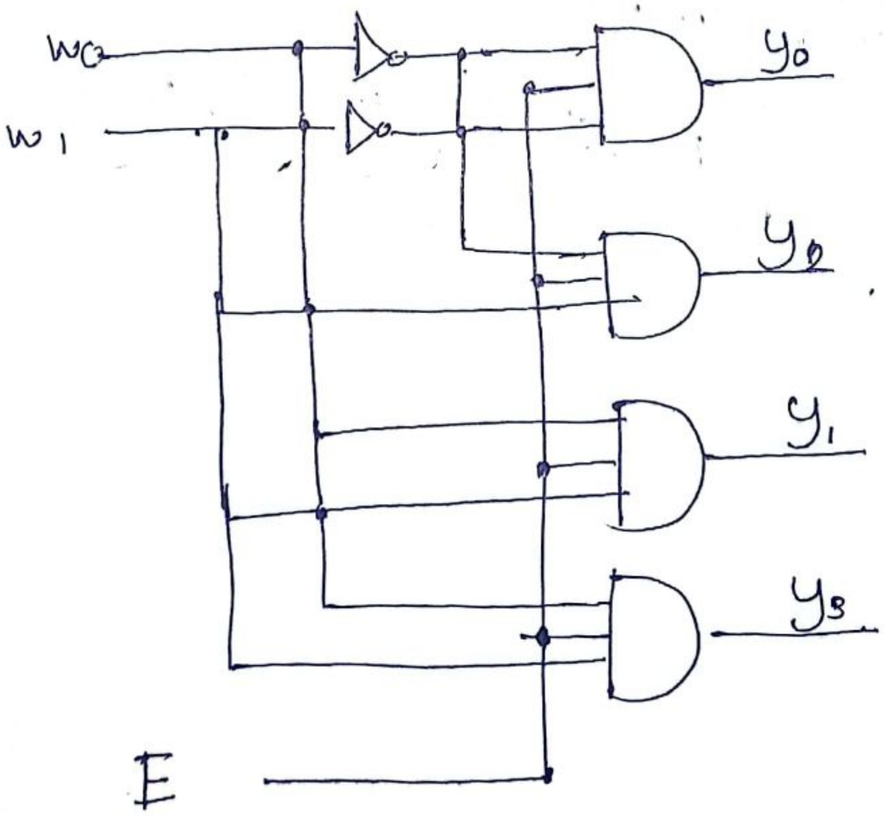
Example . 2 to 4 decoder.



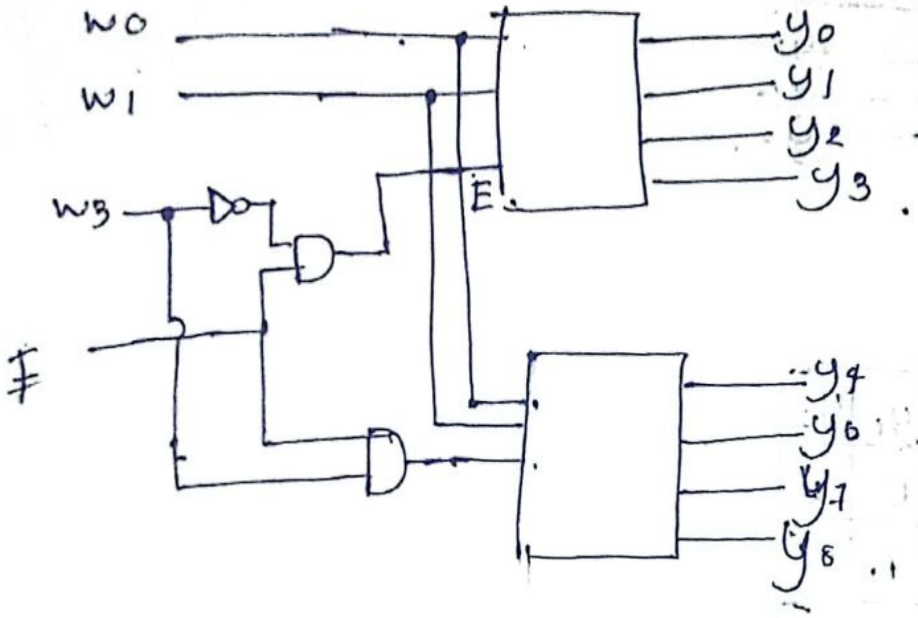
truth table .

E	w1	w0	y3	y2	y1	y0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0
0	X	X	0	0	0	0

wc
w1.
w2
w:



Example. 3 to 8 decorder ⇒ 2 to 4 decoder



Demultiplexer

One input, Many output

$n \rightarrow$ number of output line
$m \rightarrow$ selector variable

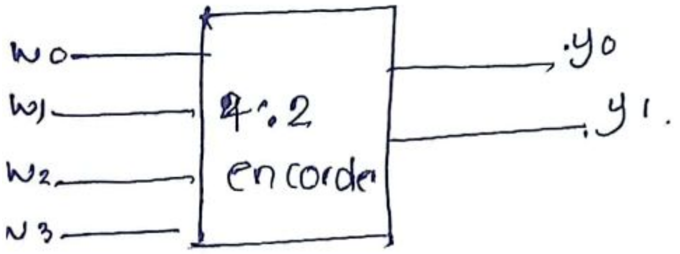
$\boxed{n = 2^m}$ $m = \log_2 n$

Encoders

2^n input → n output

2^n binary encoder

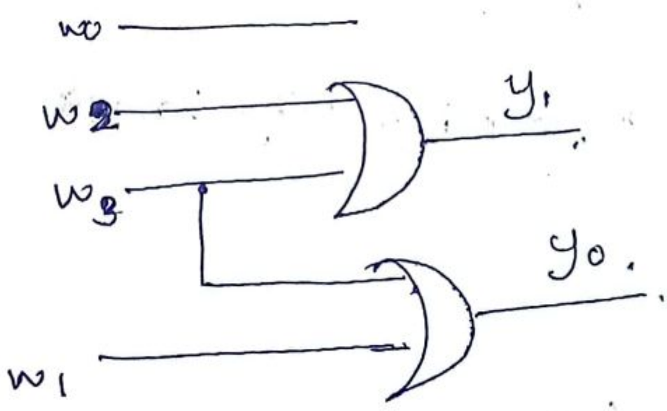
. 4 to 2 binary encorder.



truth table .

w3	w2	w1	w0	y1	y0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

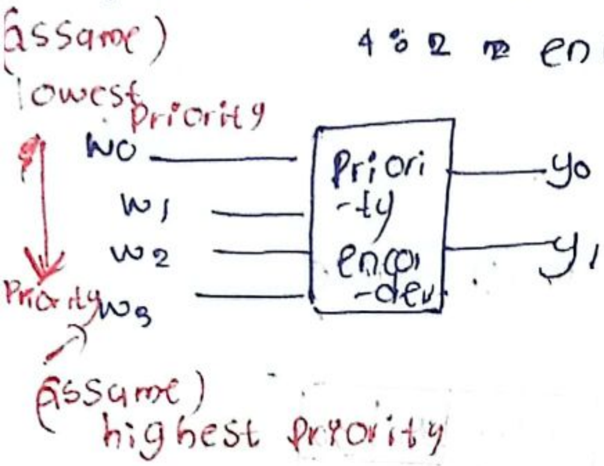
here, $y_1 = w_3 + w_2$.
$y_0 = w_3 + w_1$.



~~PP~~ Priority Encoders.

* each input has priority level associated with it.

4 : 2 encoder.



truth table.

w_3	w_2	w_1	w_0	y_1	y_0
0	0	0	0	X	X
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1

← when all the input are zero output is dont care

← when w0 is 1, y_1 & y_2 → 0

← when w_1 is 1 then w0 dont care (becau Priority of w1 > w0)

← when w_2 is 1 then w_1, w_2 are dont care, becouse Priority of w_2 > w_1, w_0

lets make equation for y_0 and y_1 using kanagh map

for y_0

$$\begin{array}{c|cccc} w_3w_2 \backslash w_1w_0 & 00 & 01 & 11 & 10 \\ \hline 00 & X & 0 & 1 & 1 \\ 01 & 0 & 0 & 0 & 0 \\ 11 & X & X & X & X \\ 10 & 1 & 1 & 1 & 1 \end{array}$$

← when w_1 is 1, y_0 is 1

when w_2 is 1, y_0 is 0

← there for this row is contridiction. ∴ we put X mark

when w_3 is 1, y_0 is 1.

$$y_0 = w_3 + \overline{w_2}\, w_1$$

for y_1

w_3w_2 \ w_1w_0	00	01	11	10
00	X	0	0	0
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$y_1 = w_2 + w_3$

~~∴ the output of Priority encorder~~

~~$= y_1 + y_0$~~

~~$= w_2 + w_3 + w_3 + \overline{w_2} w_1$~~

~~$= w_2 + w_3 + \overline{w_2} w_1$~~

~~$= w_1 + w_2 + w_3$~~

let the output of Priority encorder is Z.

$Z = w_0 + w_1 + w_2 + w_3$ (one of the input is 1 Z goes to 1).

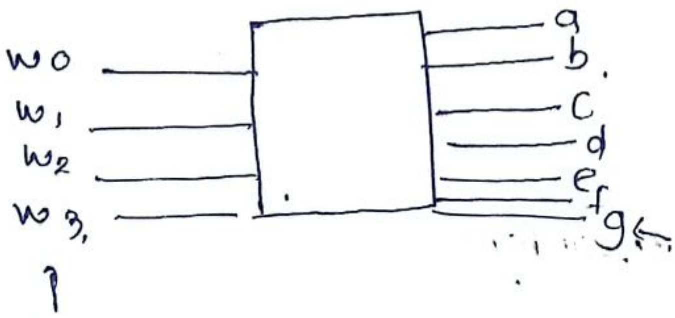
the output,

$Z = \overline{w_3}\,\overline{w_2}\,\overline{w_1}\,w_0 + \overline{w_3}\,\overline{w_2}\,w_1 + \overline{w_3}\,w_2 + w_3$

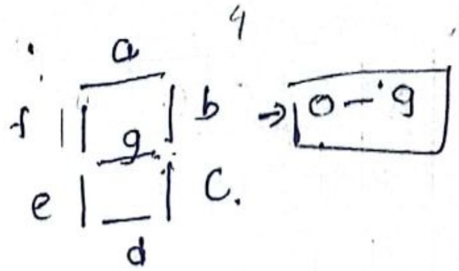
code converter.

binary to BCD.

ex:- 7 Segment display



a-g outputs.



4input because 9 has for digit

truth table.

8 4 2 1

w3	w2	w1	w0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1

E for represent 0

E for represt 1