Wednesday

DATE 25-01-17

PAGE NO. 1

Computer Architecture-

It is a concern with the structure & behaviour of the computer which includes informations, formats, instruction sets & technique of addressing memory. It is concern with the specification of the various function module such as processors & memory.

Computer Organisation-

It is concern with all the hardware components & the way they are connected together to form a computer system. The task of organization is to identify that all parts are working properly.

operations

27-01-2017
Friday

Computer Design-

It is concern with the hardware of the computer. It is concern with determination of what hardware should be used & how the path should be connected.

Logic Gates-

Binary information is represented in digital computers by physical quantities called signals.

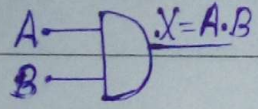
All operators within a computer carried out by needs of combination of

Teacher's Signature

Signals passing through standard block of built-in circuits that are known as logic gates.

AND Gate–

A	B	X
1	1	1
1	0	0
0	1	0
0	0	0



When all the I/Ps are 1 the O/P is 1. otherwise 0 #.

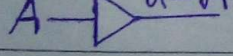
OR Gate–

A	B	X
1	1	1
1	0	1
0	1	1
0	0	0

If all I/Ps are 0 then O/P is 0 otherwise 1.

NOT Gate

A	X
1	0
0	1



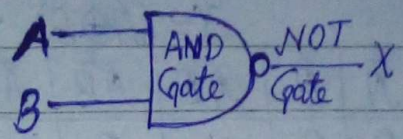
If I/P is 0 then O/P is 1 and vice-versa.

Teacher's Signature

Saturday
DATE 28/01/17
PAGE NO.

NAND Gate (NOT of AND Gate) –

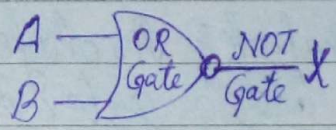
A	B	A·B	$X=\overline{A \cdot B}$
1	1	1	0
1	0	0	1
0	1	0	1
0	0	0	1



It is complement of AND Gate.

NOR Gate (NOT of OR Gate) –

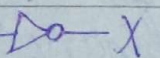
A	B	A+B	$X=\overline{A+B}$
1	1	1	0
1	0	1	0
0	1	1	0
0	0	0	1



It is complement of OR Gate.

Buffer Gate –

A	X
0	0
1	1

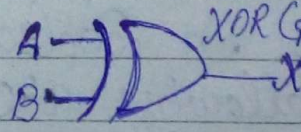


It is a storage gate used to store data.

XOR (Exclusive OR Gate) –

$$X = A \oplus B = \bar{A}B + A\bar{B}$$

A	B	$\bar{A}$	$\bar{B}$	$\bar{A} \cdot B$	$A \cdot \bar{B}$	$A \oplus B$
1	1	0	0	0	0	0
1	0	0	1	0	1	1
0	1	1	0	1	0	1
0	0	1	1	0	0	0



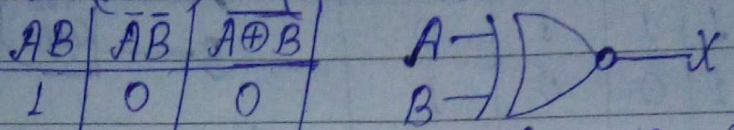
It produces O/P as 1 when both I/Ps are unequal.

Teacher's Signature

XNOR (Exclusive NOR Gate) :-

$$X = (\overline{A \oplus B}) = (\bar{A}\bar{B} + AB)$$

A	B	$\bar{A}$	$\bar{B}$	AB	$\bar{A}\bar{B}$	$\overline{A \oplus B}$
1	1	0	0	1	0	0
1	0	0	1	0	0	1
0	1	1	0	0	0	1
0	0	1	1	0	1	0



When the I/Ps signals are even or same then the O/P signal is 1 otherwise 0.

Questions-

1. What is Computer Architecture?
2. What is Logic gate? Define types of Logic Gates.
3. Draw a truth table for XOR & XNOR.

Boolean Algebra-

This is an algebra which is used to analyse & simplify the digital (logic gates) circuits. It uses only the binary numbers (i.e., 0 & 1). It is also called as binary algebra or logical algebra. Boolean Algebra was invented by George Boole in 1854.

Rules in Boolean Algebra-

Following are the important rules used in Boolean Algebra:-

(1) Variables used can have only two values binary 1 for high voltage & binary 0 for low voltage

(2) Complement of a variable is represented by (over bar). Thus, complement of a variable

Teacher's Signature

is represented as if $B=0$ then $\bar{B}=1$ as $\bar{B}=0$ then $B=1$.

(3) ORing of a variable is represented by a plus sign between variables for example of ORing is: if we have 3 variables A, B, C then ORing is represented as $A+B+C$.

(4) ANDing – Logical ANDing of the two or more variables are represented by writing (.) period or dot sign between variables. If we have three variables A, B, C then ANDing is $A.B.C$.
* Sometimes dots are omitted and it looks like ABC.

Saturday
04-02-17

There are six types of Boolean laws:

(1.) Commutative Law–

(i) $A.B = B.A$ (ii) $A+B = B+A$

(2.) Associative Law–

(i) $A.(B.C) = (A.B).C$ (ii) $A+(B+C) = (A+B)+C$

(3.) Distributive Law–

(i) $A.(B+C) = A.B + A.C$

(4.) AND Law–

(i) $A.1 = A$
(ii) $A.0 = 0$
(iii) $A.A = A$
(iv) $A.\bar{A} = 0$

(5.) OR Law–

$A+1 = 1$
$A+0 = A$
$A+A = A$
$A+\bar{A} = 1$

Teacher's Signature

(6.) Inverse Law-

(i) $\bar{\bar{A}} = A$

There are six types of Boolean laws:

(1.) Commutative Law-

Any binary operation which satisfies the following expression is referred to as Commutative Operation

(i) $A \cdot B = B \cdot A$ (ii) $A + B = B + A$

Commutative law states that changing the sequence of the variables does not have any effect on the logic circuit.

(2.) Associative Law-

This law states that the order in which the logic operations are performed is irrelevant as their effect is same.

(i) $(A \cdot B) \cdot C = A \cdot (B \cdot C)$ (ii) $(A + B) + C = A + (B + C)$.

(3.) Distributive Law-

Distributive Law states the following conditions:

(i) $A \cdot (B + C) = A \cdot B + A \cdot C$

(4.) AND Law-

This law use the AND operation therefore they are called as AND Laws.

(i) $A \cdot 1 = A$

(ii) $A \cdot 0 = 0$

(iii) $A \cdot A = A$

(iv) $A \cdot \bar{A} = 0$

Teacher's Signature

(5.) OR Law-

This law use the OR operation therefore they are called as OR Laws.

(i) $A+1=1$ (ii) $A+0=A$

(iii) $A+A=A$ (iv) $A+\bar{A}=1$

(6.) Inverse Law-

This law uses NOT operation the inversion law states the double inversion of a variable results in the original variable itself.

(i) $\bar{\bar{A}}=A$

Thursday
09.02.17

Important Boolean Function-

Boolean Algebra deals with binary variables and logic operations.

A boolean function is described by an algebraic expression called boolean variables. The constants 0 and 1 and the logic operation signals.

Consider the following example-

(P.T.O.)

Teacher's Signature

$F(A,B,C,D) = A + \overline{BC} + ADC$

Boolean Algebra -

A	B	C	D	BC	ADC	$\overline{BC}$	F
1	1	1	1	1	1	0	1
1	1	1	0	1	0	0	1
1	1	0	1	0	0	1	1
1	1	0	0	0	0	1	1
1	0	1	1	0	1	1	1
1	0	1	0	0	0	1	1
1	0	0	1	0	0	1	1
1	0	0	0	0	0	1	1
0	1	1	1	1	0	0	0
0	1	1	0	1	0	0	0
0	1	0	1	0	0	1	1
0	1	0	0	0	0	1	1
0	0	1	1	0	0	1	1
0	0	1	0	0	0	1	1
0	0	0	1	0	0	1	1
0	0	0	0	0	0	1	1

Here the left side of the equation represent the output Y so we can state $Y = A + \overline{BC} + ADC$.

Truth Table Formation - A truth table represent a table having all combinations and there are corresponding results. It is possible to connect the switching equation in a truth table. $F(A,B,C) = A + BC$

The output will be high (i.e., 1) if $A=1$ or $BC=1$.

The no. of rows in the truth table is 2^n where n is the no. of input variables.

Methods to simplify the boolean function are as follows: 1.) Karnaugh Map (K-Map).

2.) NAND Gate Method.

Teacher's Signature

Karnaugh Map (K-map)-

The boolean theorems & De-Morgan's Theorem are useful in the manipulating the logic expressions.

We can realize the logical expressions using gates.

The no. of logic gates are required for the realization of a logical expression should be reduced to a minimum possible values by K-Map.

Rules:

Rules to fill the truth table values into the map:-

1. Enter the one resulting column into its equivalent cell
2. In circle the one's in the K-map in pairs or equadator doing this thing we can circle either row wise or column wise.
3. If anyone isolated then we can also circle them.
4. For each circle write the relevant equation removing the variables with changes its value in circle.
5. When we are plotting one in the K-map then the equation will be a sum of product (SOP) and we are plotting zero in the K-map then the equation will be a product of sum (POS).

Teacher's Signature

K-Map

A	B	C	D	Y
1	1	1	1	0
1	1	1	0	1
1	1	0	1	0
1	1	0	0	1
1	0	1	1	0
1	0	1	0	1
1	0	0	1	0
1	0	0	0	1
0	1	1	1	0
0	1	1	0	1
0	1	0	1	0
0	1	0	0	1
0	0	1	1	0
0	0	1	0	1
0	0	0	1	0
0	0	0	0	1

CD \ AB	$\bar{A}\bar{B}$ 00	$\bar{A}B$ 01	AB 11	$A\bar{B}$ 10
$\bar{C}\bar{D}$ 00	1	1		1
$\bar{C}D$ 01				
CD 11				
$C\bar{D}$ 10	1	1		1

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

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

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

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

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

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

Teacher's Signature

Friday
DATE 10/02/17

SOP (Sum of Product) - Minterm = Σm
1 is main.

	A	B	C	F
0 =	0	0	0	0
1 =	0	0	1	0
2 =	0	1	0	0
3 =	0	1	1	1
4 =	1	0	0	0
5 =	1	0	1	1
6 =	1	1	0	1
7 =	1	1	1	1

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

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

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

POS (Product of Sum) - Maxterm = ΠM
0 is main

	A	B	C	F
0 =	0	0	0	0
1 =	0	0	1	0
2 =	0	1	0	0
3 =	0	1	1	1
4 =	1	0	0	0
5 =	1	0	1	1
6 =	1	1	0	1
7 =	1	1	1	1

$\Pi M = (0,1,2,4)$

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

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

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

Teacher's Signature

De-Morgan's Law -

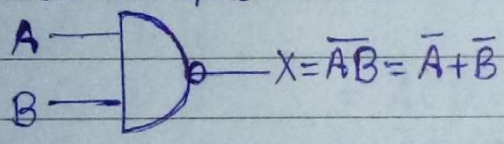
Theorem First -

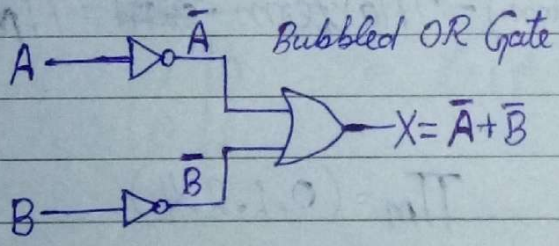
(1.) $\overline{A \cdot B} = \bar{A} + \bar{B}$

(NAND) (Bubbled OR)

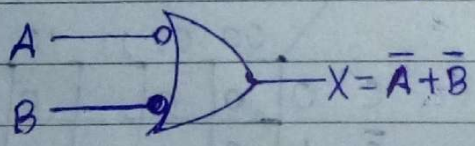
A	B	A·B	$\bar{A}$	$\bar{B}$	$\bar{A}+\bar{B}$	$\overline{A \cdot B}$
0	0	0	1	1	1	1
0	1	0	1	0	1	1
1	0	0	0	1	1	1
1	1	1	0	0	0	0

NAND Gate





Bubbled OR Gate



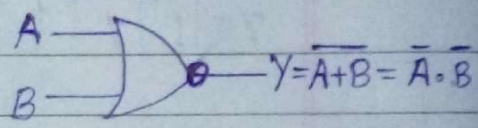
Theorem Second -

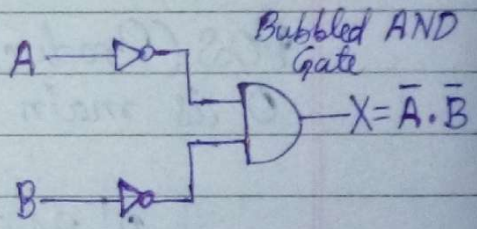
(2.) $\overline{A+B} = \bar{A} \cdot \bar{B}$

(NOR) (Bubbled AND)

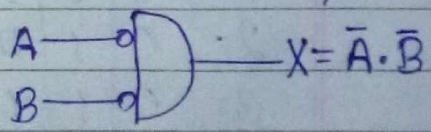
A	B	A+B	$\overline{A+B}$	$\bar{A}$	$\bar{B}$	$\bar{A} \cdot \bar{B}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

NOR Gate





Bubbled AND Gate



Teacher's Signature

Principle of Quality proved by De-Morgan's ~~law~~ First Theorem –

e.g. → (i) $A.(B+C) = A.B + A.C$

(ii) $A + (B.C) = (A+B)(A+C)$

(i)

A	B	C	(B+C)	A(B+C)	A.B	A.C	A.B + A.C
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

(ii)

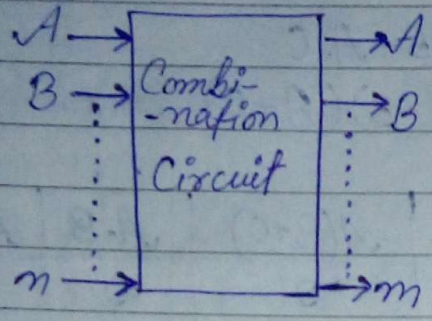
A	B	C	B.C	A+(B.C)	A+B	A+C	(A+B).(A+C)
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

* It states that where there is '+' it will be changed to '.' & vice versa.
* And if '1' is present then it will be converted into '0' & vice versa.

Teacher's Signature

SHREE
DATE 11/02/17
PAGE NO.:

Introduction to Combinational Circuit -



Combination circuit is a circuit in which we combine different gates in the circuit. For example- Encoder, Decoder, Multiplexers, Demultiplexers, etc.

Some of the following characteristics of combination circuits are -

(1.) The output of combination circuits at any instant of time depends only on the levels present at input terminal.

(2.) The combination circuit don't use any memory.

(3.) The previous state of input does not have any effect on the present state of circuit.

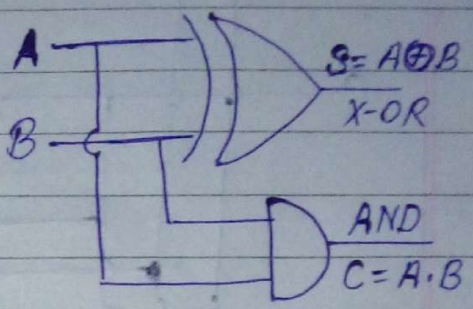
(4.) A combination circuit can have 'n' no. of inputs & 'm' no. of outputs.

Teacher's Signature

Few combinations circuits are as follows:

(1.) Half Adder –

Input		Output	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Sum = $A \oplus B$ (X-OR property)

Carry = $A \cdot B$ (AND property)

(2.) Full Adder –

Input			Output	
A	B	C_{in}	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

~~Sum = $A \oplus B$ (X-NOR property)~~

Output for Sum via K-map –

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

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

$S = A \oplus B \oplus C$

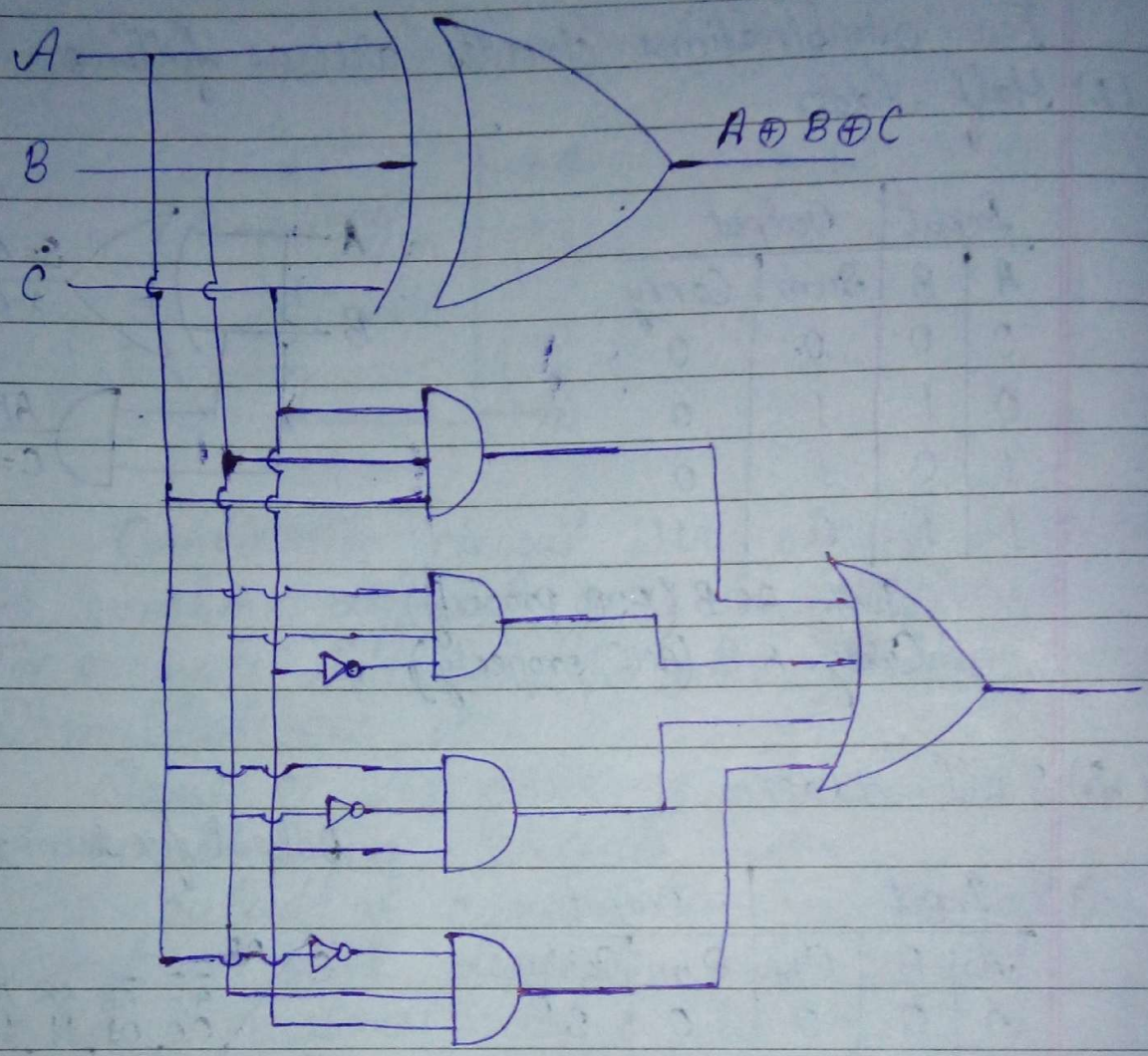
C \ AB	$\bar{A}\bar{B}$ 00	$\bar{A}B$ 01	AB 11	$A\bar{B}$ 10
$\bar{C}$ 0			1	
C 1		1	1	1

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

$= BC + AB + AC$

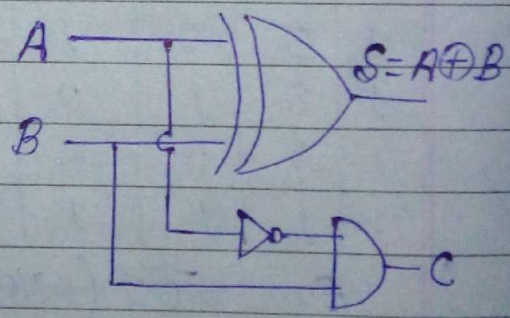
[$\because \bar{C} + C = 1$, $B + \bar{B} = 1$, $A + \bar{A} = 1$]

Teacher's Signature



Half Subtractor –

Input		Sum	Carry (C)
A	B	A-B	
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

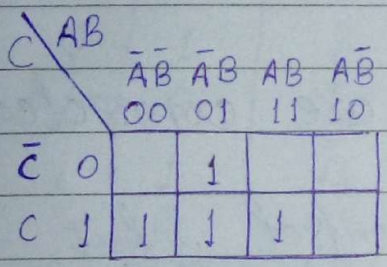


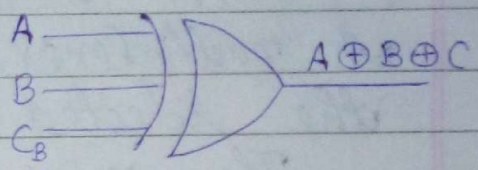
Sum = A⊕B (XOR property)

Teacher's Signature

Full Subfractor –

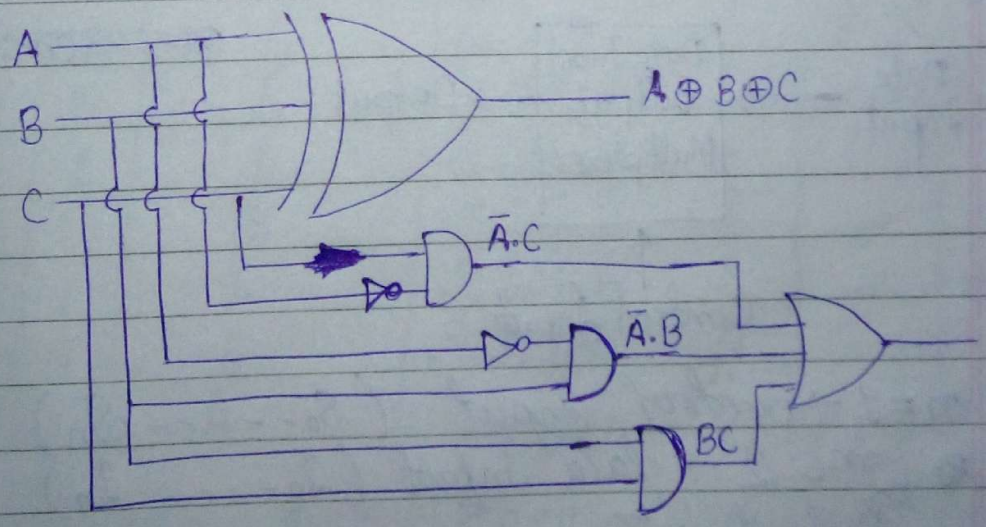
Input				Sub.	
A	B	C_B	A-B=X	X-C	Borrower
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	1	1	1
0	1	1	1	0	1
1	0	0	1	1	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	1	0	1	1





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

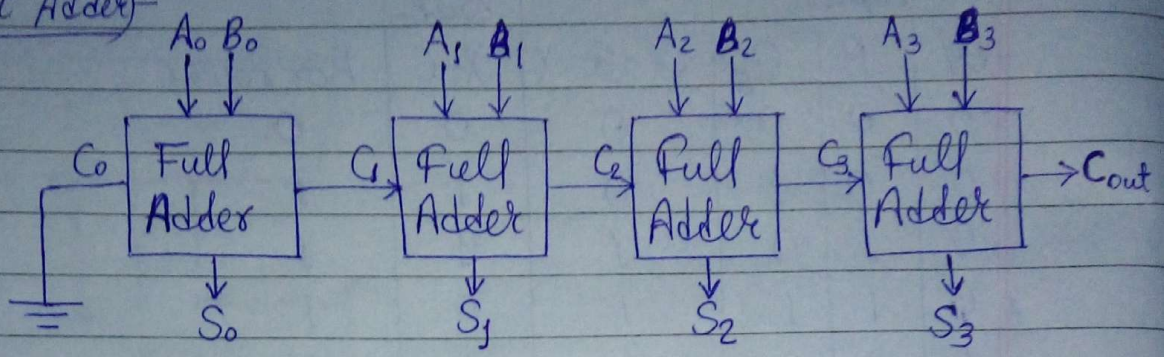
$= (\bar{A}C) + (\bar{A}B) + (BC)$



Teacher's Signature

Parallel Adder -

(4 bit Adder)



Wednesday
15-02-2017

Multiplexer -

A multiplexer has:

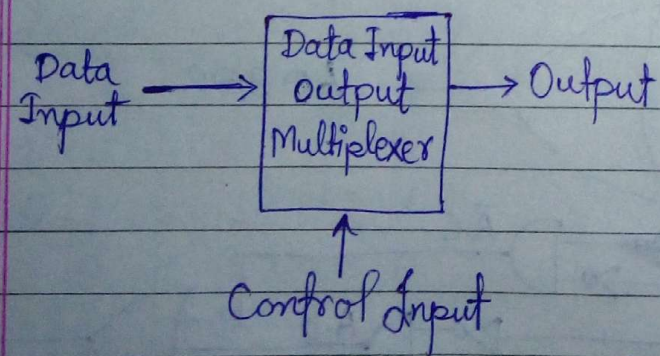
(i) 2^n data input
(ii) n control input
(iii) 1 output

A multiplexer routes are connected the selected data.

The value of the control input determine the data input that is selected.

Multiplexer called data collector.

~~$n-1 \rightarrow$ control Input~~



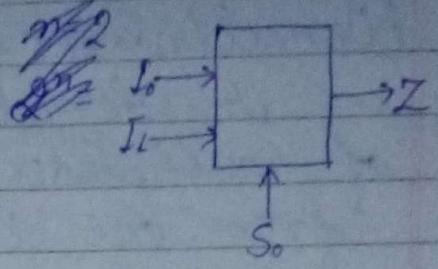
$n = 1 \rightarrow$ control Input (S_0 ----- S_n)

$2^n = 2 \rightarrow$ data input (I_0 ---- I_n)

Output $\rightarrow Z$

Teacher's Signature

S_0	Data Input	Output
0	I_0	$S_0 I_0$
1	I_1	$S_0 I_1$



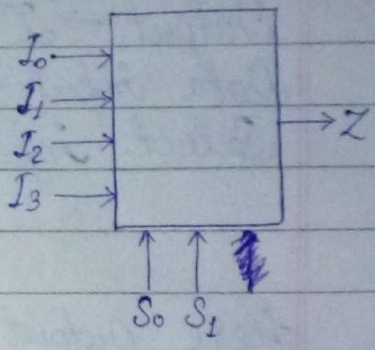
$$Z = \bar{S}_0 I_0 + S_0 I_1$$

$n = 2$

$2^n = 4$

Output = 1

4:2 Multiplexer



S_0	S_1	Data Input	Z
0	0	I_0	$\bar{S}_0 \bar{S}_1 I_0$
0	1	I_1	$\bar{S}_0 S_1 I_1$
1	0	I_2	$S_0 \bar{S}_1 I_2$
1	1	I_3	$S_0 S_1 I_3$

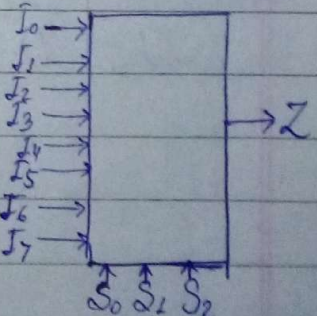
$$Z = \bar{S}_0 \bar{S}_1 I_0 + \bar{S}_0 S_1 I_1 + S_0 \bar{S}_1 I_2 + S_0 S_1 I_3$$

S_0	S_1	S_2	Data Input	Z
0	0	0	I_0	$\bar{S}_0 \bar{S}_1 \bar{S}_2 I_0$
0	0	1	I_1	$\bar{S}_0 \bar{S}_1 S_2 I_1$
0	1	0	I_2	$\bar{S}_0 S_1 \bar{S}_2 I_2$
0	1	1	I_3	$\bar{S}_0 S_1 S_2 I_3$
1	0	0	I_4	$S_0 \bar{S}_1 \bar{S}_2 I_4$
1	0	1	I_5	$S_0 \bar{S}_1 S_2 I_5$
1	1	0	I_6	$S_0 S_1 \bar{S}_2 I_6$
1	1	1	I_7	$S_0 S_1 S_2 I_7$

$n = 3$

$2^3 = 8$

Output = 1



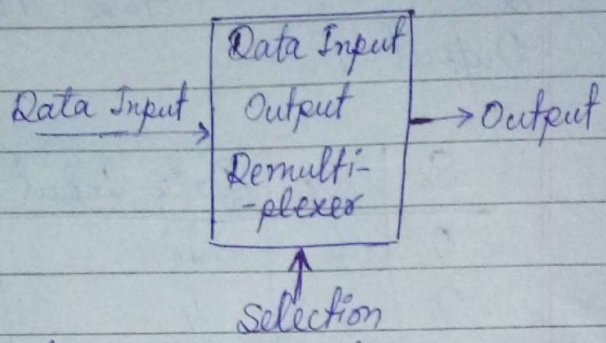
$$Z = \bar{S}_0 \bar{S}_1 \bar{S}_2 I_0 + \bar{S}_0 \bar{S}_1 S_2 I_1 + \bar{S}_0 S_1 \bar{S}_2 I_2 + \bar{S}_0 S_1 S_2 I_3 + S_0 \bar{S}_1 \bar{S}_2 I_4 + S_0 \bar{S}_1 S_2 I_5 + S_0 S_1 \bar{S}_2 I_6 + S_0 S_1 S_2 I_7$$

Teacher's Signature

Demultiplexer–

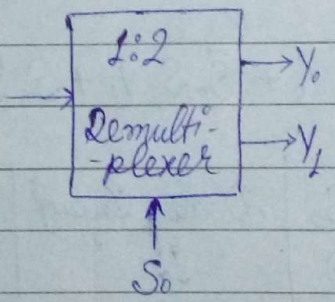
A demultiplexer is basically reverse of multiplexer function. It takes data from one line and distribute them to a given no. of output line. For this reason ~~no. of~~ demultiplexer known as data distributor.

Output → 2^n
Data input → 1
Select input → n



Here Output is 2^n and Data input is one and select input is n.

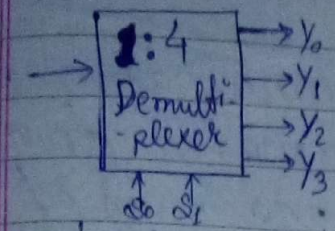
Output → 2
Data Input → 1
Select input → 1



S_0	D	Y_1	Y_0
0	D	0	D
1	D	D	0

$Y_0 = \bar{S}_0 D$
$Y_1 = S_0 D$

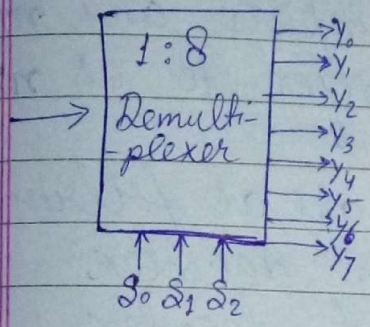
Teacher's Signature



Output → 4
Data Input → 1
Select input → 2

D	S_0	S_1	Y_3	Y_2	Y_1	Y_0
D	0	0	0	0	0	D
D	0	1	0	0	D	0
D	1	0	0	D	0	0
D	1	1	D	0	0	0

$Y_0 = \bar{S}_0 \bar{S}_1 D$
$Y_1 = \bar{S}_0 S_1 D$
$Y_2 = S_0 \bar{S}_1 D$
$Y_3 = S_0 S_1 D$



Output → 8
Data Input → 1
Select input → 3

D	S_0	S_1	S_2	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
D	0	0	0	0	0	0	0	0	0	0	D
D	0	0	1	0	0	0	0	0	0	D	0
D	0	1	0	0	0	0	0	0	D	0	0
D	0	1	1	0	0	0	0	D	0	0	0
D	1	0	0	0	0	0	D	0	0	0	0
D	1	0	1	0	0	D	0	0	0	0	0
D	1	1	0	0	D	0	0	0	0	0	0
D	1	1	1	D	0	0	0	0	0	0	0

$Y_0 = \bar{S}_0 \bar{S}_1 \bar{S}_2 D$
$Y_1 = \bar{S}_0 \bar{S}_1 S_2 D$
$Y_2 = \bar{S}_0 S_1 \bar{S}_2 D$
$Y_3 = \bar{S}_0 S_1 S_2 D$
$Y_4 = S_0 \bar{S}_1 \bar{S}_2 D$
$Y_5 = S_0 \bar{S}_1 S_2 D$
$Y_6 = S_0 S_1 \bar{S}_2 D$
$Y_7 = S_0 S_1 S_2 D$

Teacher's Signature

Thursday
DATE 16/02/17
PAGE NO.

Decoder - A decoder is a combination circuit that converts binary information from the encoded inputs to a maximum of 2^n unique output.

The decoder presented in this section are called n to m line decoders where $m <= 2^n$.

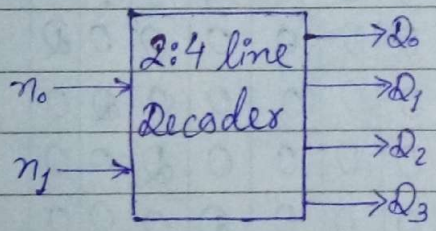
There purpose is to generate the 2^n binary combination of the n input variables.

A decoder has n inputs and m outputs and is also refer to as n×m decoder.

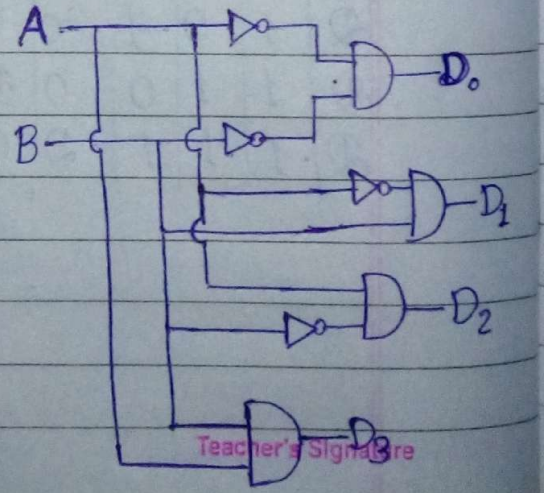
Examples of decoder are as following:-

(1.) Code Convertors - 2 to 4 line decoder.
(2.) BCD to seven segment decoder.

(1.) Example - Code Converter (2 input, 4 output)



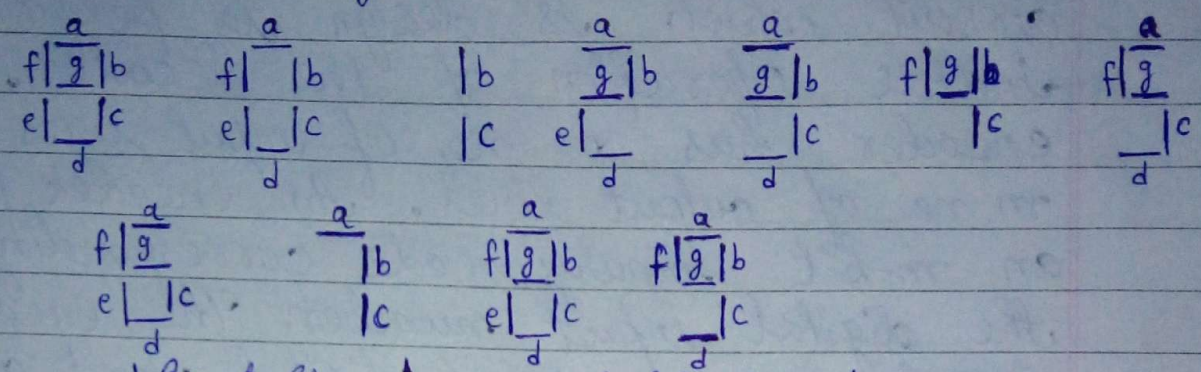
A	B	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



Teacher's Signature

Example-

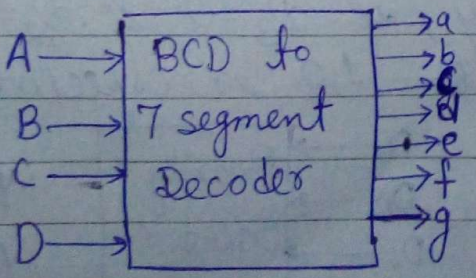
(2.) BCD to 7 segment decoder (4 inputs, 7 outputs)



Display	Input lines A	B	C	D	Output lines a	b	c	d	e	f	g	Output
0	0	0	0	1	1	1	1	1	1	1	0	
1	0	0	0	0	0	1	1	0	0	0	0	
2	0	0	1	1	1	1	0	1	1	0	1	
3	0	0	1	0	1	1	1	1	0	0	1	
4	0	1	0	1	0	1	1	0	0	1	1	
5	0	1	0	0	1	0	1	1	0	1	1	
6	0	1	1	1	1	0	1	1	1	1	1	
7	0	1	1	0	1	1	1	0	0	0	0	
8	1	0	0	1	1	1	1	1	1	1	1	
9	1	0	0	0	1	1	1	1	0	1	1	

It is a circuit used to convert the input BCD into a form suitable for the display.

It has 4 input lines A, B, C, D and 7 output lines a, b, c, d, e, f, g.



Teacher's Signature

Encoder- Encoder is a combination circuit which is design to perform the inverse operation of the decoder. An encoder has n no. of input lines and m no. of output lines. An encoder produces an m-bit binary code corresponding to the digital input number. The encoder accepts an n input digital word and convert it into an n-bit another digital word.

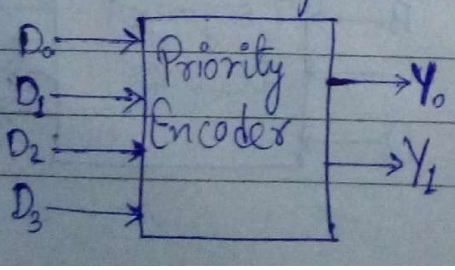
Examples of Encoder-

(1.) Priority Encoder

(2.) Decimal to BCD Encoder.

(1.) Priority Encoder- This is a special type of encoder, priority is given to the input lines. If two or more input lines are one at the same time then the input line with the highest priority will be consider. There are 4 inputs D_0, D_1, D_2 & D_3 and two outputs are Y_0 & Y_1.

Out of the four inputs D_3 has the highest priority and D_0 has the lowest priority. That means if $D_3=1$ then Y_1, Y_0 equals to 1,1 irrespective of the other input. Similarly, if $D_3=0$ & $D_2=1$ then Y_1, Y_0 equals to 1,0 irrespective of the other input.



Teacher's Signature

Highest	Input		Lowest	Output	
D_3	D_2	D_1	D_0	Y_0	Y_1
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

(2) Decimal to BCD Encoder (10 inputs, 4 outputs)

eg. →

D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	1	0	0	1

$$Y_0 = D_1 + D_3 + D_5 + D_7 + D_9$$
$$Y_1 = D_2 + D_3 + D_6 + D_7$$
$$Y_2 = D_4 + D_5 + D_6 + D_7$$
$$Y_3 = D_8 + D_9$$

Teacher's Signature