

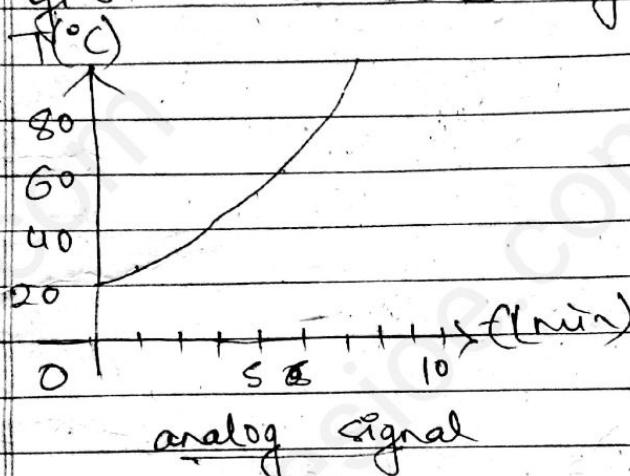
Ch - I

Date _____
Page _____

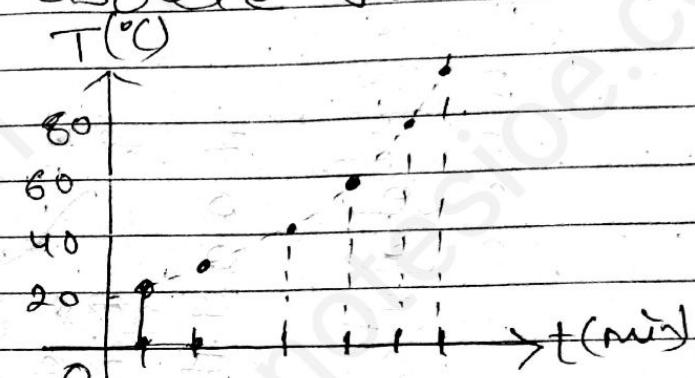
Introduction:

Analog: Analog signals are continuous where all possible values are represented.

Digital: Digital signals represent only a finite number of discrete values.



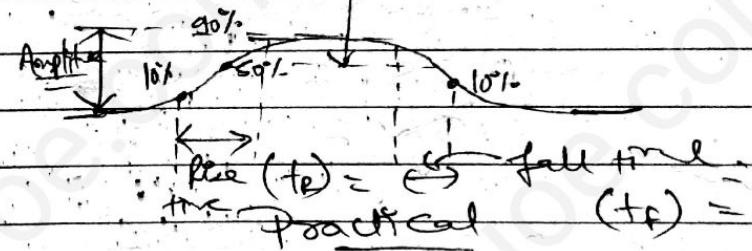
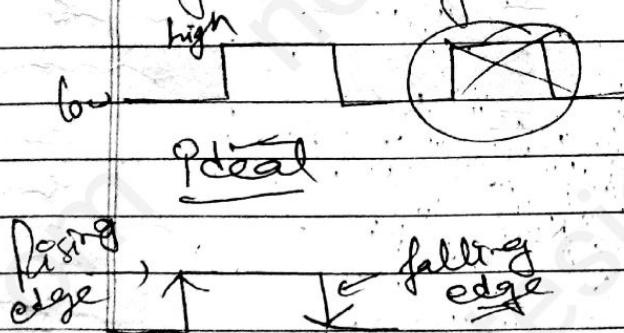
analog signal



digital signal

pulse width (t_w)

Digital waveform.



$T \rightarrow$ time period =

\rightarrow Time over which signal repeats itself.

$$F = 1/T$$

Duty cycle:

$$\rightarrow \frac{t_h}{T} \times 100\% \quad \left. \right\} \quad \frac{t_l}{T} \times 100\% \quad \begin{matrix} \text{for high} \\ \text{for low} \end{matrix}$$

Number system

Date _____
Page _____

- Decimal (base 10)
- Binary (base 2)
- Octal (base 8)
- Hexadecimal (base 16).

eg: $(25.625)_{10} \rightarrow (?)_8$

$$\begin{array}{r} 2 | 25 \\ 2 | 12 \quad I \rightarrow 1 \\ 2 | 6 \quad 0 \quad \uparrow \\ 2 | 3 \quad 0 \\ 2 | 1 \quad I \rightarrow 2 \\ 0 \quad I \rightarrow 16 \end{array}$$

$$(25)_{10} \rightarrow (11001)_2$$

$$\begin{array}{r} 0.625 \times 2 = 1.25 \quad I \\ 0.25 \times 2 = 0.5 \quad 0 \\ 0.5 \times 2 = 1 \quad I \end{array}$$

$$(0.625)_{10} = (0.101)_2$$

$$(25.625)_{10} = (11001.101)_2 \quad \#$$

BCD

Date _____
Page _____

0	0000
1	0001
9	1001
10	0001, 0000
100	0001 0000 0000

Merit \rightarrow Encoding & decoding is easy

Demerit \rightarrow occupies more space than corresponding binary

$$09 : \text{Dec} \rightarrow 1001$$

$$\text{BIN} \rightarrow 1101$$

$$\text{BCD} \rightarrow 0001 \underline{0011} =$$

Excess - 3

$$\begin{array}{r} 1 \quad 2 \\ + 3 \quad + 3 \\ \hline 4 \quad 5 \\ \downarrow \quad \downarrow \\ = (0100 \quad 0101)_{\text{Excess-3}} \end{array}$$

$$\begin{array}{r} 2 \quad 9 \\ + 3 \quad + 3 \\ \hline 5 \quad 12 \\ \downarrow \\ (0101) \cdot (1100) \end{array}$$

Gray Code

Date _____
Page _____

<u>Street</u>	<u>Grey</u>
0 0	{ 00
0 1	{ 01
1 0	{ 11
1 1	{ 10

- Unweighted & not arithmetic code.
 - Only one bit changes as it sequences from one number to next.

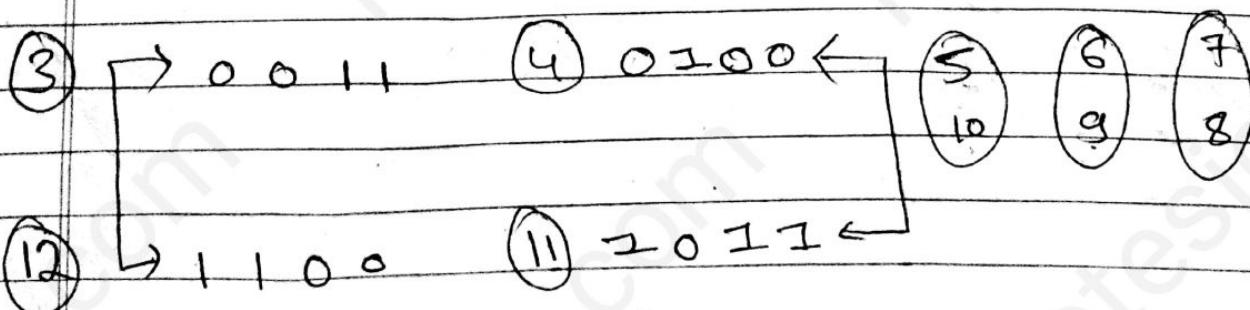
Binary to gray

$$\begin{array}{c} (\overline{1} \ 0 \ 1 \ 1 \ 0)_B \\ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ (\overline{1} \ 1 \ 1 \ 0 \ 1)_A \\ \downarrow \swarrow \swarrow \swarrow \swarrow \swarrow \\ (\overline{1} \ 0 \ 1 \ 1 \ 0)_B \end{array}$$

- \rightarrow (weighted & non-weighted) +

decimal, binary, octal, etc

Way, Excess-3



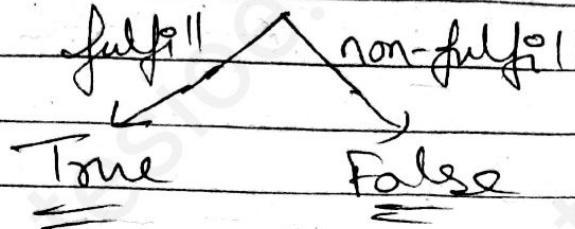
2017 I → What will BCD, Excess-3 & gray code
represent number 15? \rightarrow F-27

Ch-2

Digital logic

Date _____
Page _____

→ Logic = declarative condition

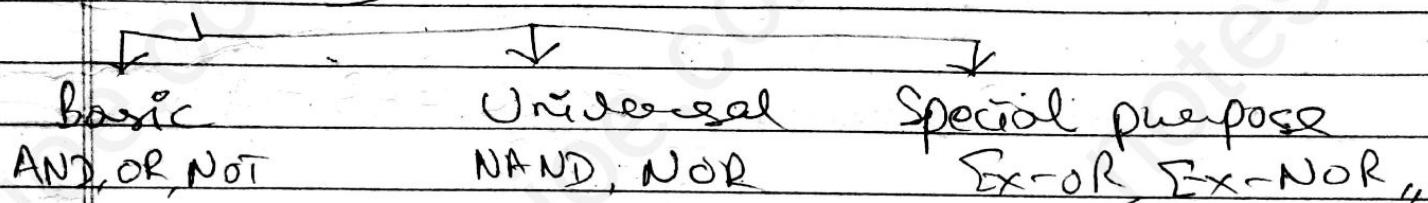


AND $\rightarrow A \cdot B$

OR $\rightarrow A + B$

NOT $\rightarrow \bar{A}$

Gates



Waveform | Timing diagram | Pulse operation.

A

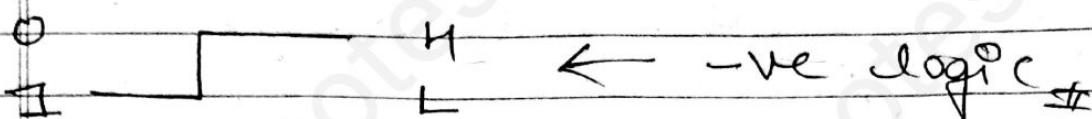
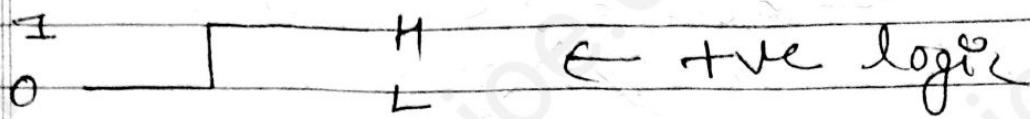
B

AND

OR

Positive & Negative logic.

Date _____
Page _____



Ds - Morgan's Law :

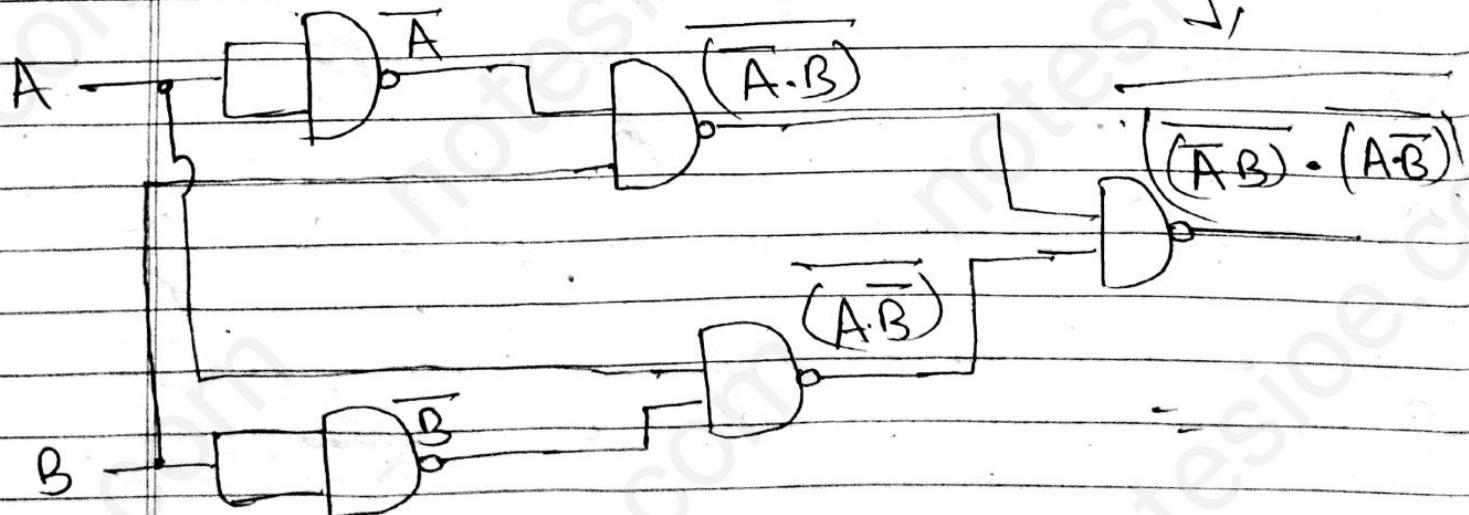
Q) Implement Ex-OR gate using NAND gates only.

$$\begin{aligned}\text{Boolean Expression of Ex-OR} &= AB' + A'B \\ &= A\bar{B} + \bar{A}B\end{aligned}$$

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

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

$$= \overline{(AB)} \cdot \overline{(A\bar{B})}$$



CH-3

(OR law, AND law, De-Morgan's laws)
 (Commutative, Associative).

Q2. Boolean Expression & Boolean function.

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

(Boolean function)



(Boolean expression.)

Boolean expression can be expressed in
 2 standard forms:

* Sum - of - products (SOP)

* Products - of - sum (POS).

I) SOP

→ When two or more product terms
 (or ANDED terms) are summed by
 boolean algebra, resulting expression is
 SOP.

$$\text{eg. } F = AB + \overline{A} \overline{B} C$$

$$F = ABC + \overline{C} \overline{D} + \overline{B} \overline{C} \overline{D} =$$

⇒ In SOP, '0' represents 'A' and 1
 represents 'A', & only 1's output
 is taken.

→ Standard SOP (Canonical SOP)

→ one where all variables in domain
 appears in each product term in
 expression.

Carey

Date _____
Page _____

→ Each individual in product term in Standard SOF form is called "minterm" & is represented by m_i .

A	B	C	Team	m_i
0	0	0	$\bar{A}\bar{B}\bar{C}$	m_0
0	0	1	$\bar{A}\bar{B}C$	m_1
⋮	⋮	⋮	⋮	⋮
1	1	1	ABC	m_7

Σm

e.g.:

91 p.			functional o/p	
A	B	C	F_1	F_2
0	0	0	1	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	0	1

$$\begin{aligned}F_1 &= \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC \\&= M_0 + M_3 + M_6 \\&= \Sigma m (0, 3, 6)\#\end{aligned}$$

$$\begin{aligned}
 F_2 &= \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC \\
 &= M_3 + M_5 + M_6 + M_7 \\
 &= \sum m(3, 5, 6, 7)
 \end{aligned}$$

Date _____
Page _____

To convert to standard SOP:

⇒ multiply each product term by $(X + \bar{X})$, where 'A' is missing term.

⇒ POS:

$$\text{eg: } (\bar{A} + B)(A + \bar{B} + \bar{C})(\bar{A} + B + C) \leftarrow \text{POS}_4$$

$$\begin{aligned}
 \text{Standard} &\rightarrow (\bar{A} + B + C)(A + \bar{B} + \bar{C})(\bar{A} + \bar{B} + C) \\
 \text{POS} &
 \end{aligned}$$

0 represents 'A'

1 represents \bar{A}

Each term in POS form is "Maxterm", & represented by 'M'.

On 3 variables:

A	B	C	Term	M_i
0	0	0	$(A + B + C)$	M_0
0	0	1	$(A + B + \bar{C})$	M_1
0	1	0	$(A + \bar{B} + C)$	M_2
.
i	j	i	$(\bar{A} + \bar{B} + \bar{C})$	M_7

eg

Date _____
Page _____

Inputs

A B C

0 0 0

0 0 1

0 1 0

0 1 1

1 0 0

1 0 1

1 1 0

1 1 1

f1

1

0

0

1

1

0

0

1

f2

0

0

0

1

1

0

1

1

$$\begin{aligned}F_1(A, B, C) &= (A + B + \bar{C})(A + \bar{B} + C)(\bar{A} + B + C)(\bar{A} + B + \bar{C}) \\&\quad (\bar{A} + \bar{B} + \bar{C}) \\&= M_1 + M_2 + M_4 + M_5 + M_7 \\&= \bar{\pi}(1, 2, 4, 5, 7)\end{aligned}$$

$$F_2(A, B, C) = \bar{\pi}(0, 1, 2, 4)$$

Converting Pos to standard Pos:

(1) Add $A \cdot \bar{A}$ to each sum term where 'A' is missing term.

(2) Apply rule $(\underline{A} + \underline{B}C) = (\underline{A} + B)(\underline{A} + C)$

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

$$\text{& } A \cdot A = I.$$

#

K-map

Date _____

Page _____

eg: Simplify & write its SOP of P.

$$F(A, B, C) = \sum m(1, 2, 3) + d(0, 5, 7)$$

~~SOP~~

A	BC	00	01	11	10	
0	X	X	X	X	X	\bar{A}
1	0	X	X	0		

$$Y = \bar{A} \#$$

$$F(A, B, C, D) = \sum m(0, 2, 5, 8, 10) + d(7, 15)$$

AB	CD	00	01	11	10	
00	X	0	0	0	0	$\bar{B}\bar{D}$
01	0	X	X	0	0	$\bar{A}BD$
11	0	0	X	0		
10	X	0	0	0	X	

$$Y = \bar{B}\bar{D} + \bar{A}BD \#$$

Simplify & write its POS expression
 $\Sigma m(1, 2, 3, 8, 9, 10, 11, 14) \oplus d(0, 4, 12)$

AB	CD	00	01	11	10
00	X	I	I	I	I
01	X	(0 0)	0X		
11	X	(0 0)	I		
10	I	I	I	I	I

$(\overline{A} + \overline{B})$

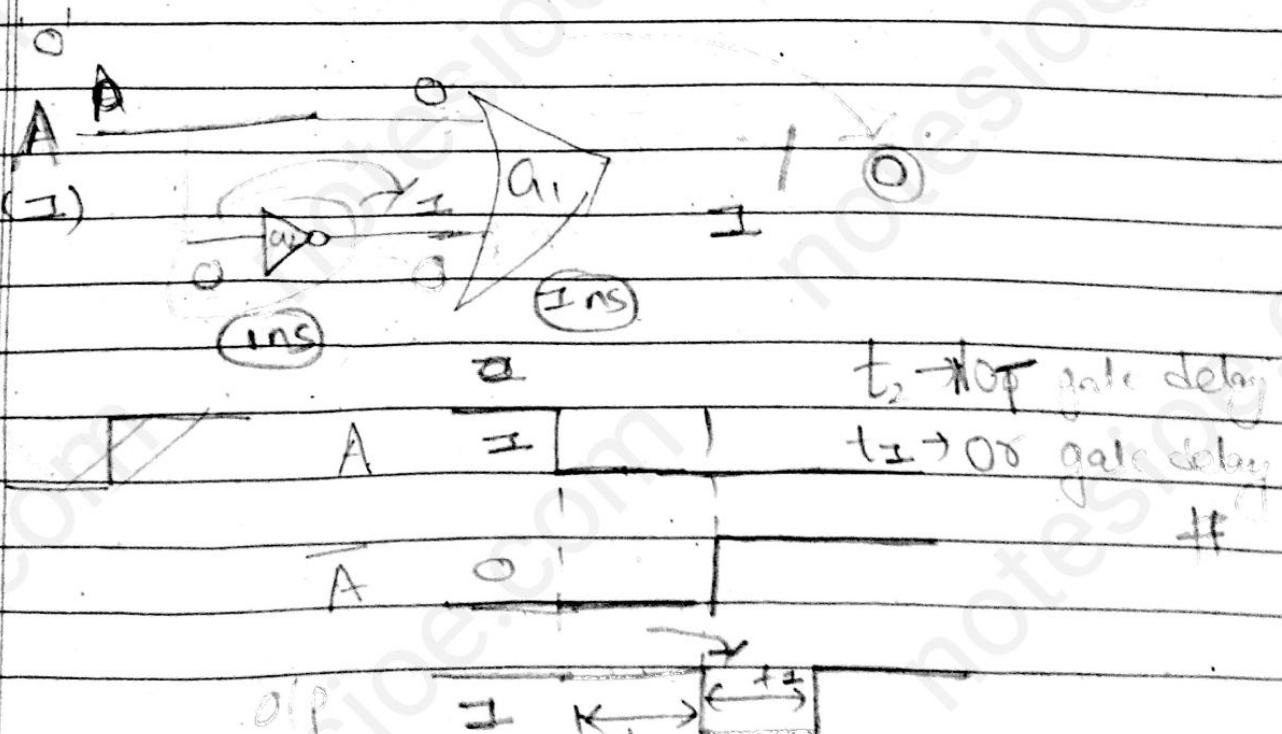
$(A + \overline{B})$

$(\overline{B} + \overline{D})$

$$Y = (\overline{A} + \overline{B})(B + \overline{D}) \#$$

Standard: ~~$(A + \overline{B} + \overline{C} + \overline{D})(A + \overline{B} + \overline{C} + \overline{S})$~~

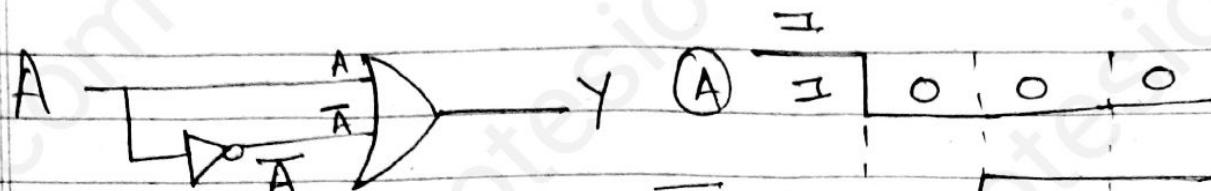
~~$(A + \overline{B} + C + \overline{D})$~~ #



(1) Static '1' Hazard

$$Y = A + \bar{A} \quad (\text{Always produces } 1 \text{ output})$$

i.e. static '1'



$\stackrel{?}{=}$

$\overline{A} \quad 0 \quad 0 \quad | \quad 1 \quad =$
 $\leftarrow t_1 \rightarrow$

$Y \quad -1 \quad 1 \quad | \quad 0 \quad =$
 $\leftarrow t_2 \rightarrow$

$t_1 = \text{NOT gate delay}$

$t_2 = \text{Or gate delay}$

(2) Static '0' Hazard

$$Y = A \cdot \bar{A}$$



$A \quad 0 \quad | \quad -1 \quad = \quad ?$

$\overline{A} \quad = \quad = \quad | \quad 0 \quad 0$

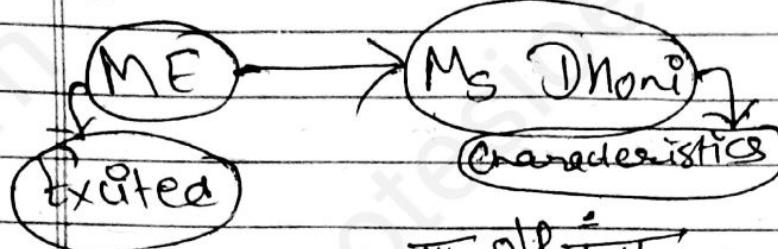
$Y \quad 0 \quad | \quad \leftarrow t_1 \rightarrow \quad \leftarrow t_2 \rightarrow$

$t_1 = \text{Not gate delay}$

$t_2 = \text{And gate delay}$

#

Convert SR to JK flip flop

Date _____
Page _____SR \rightarrow J1CgivenDesired

Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

J	K	Q_n	Q_{n+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

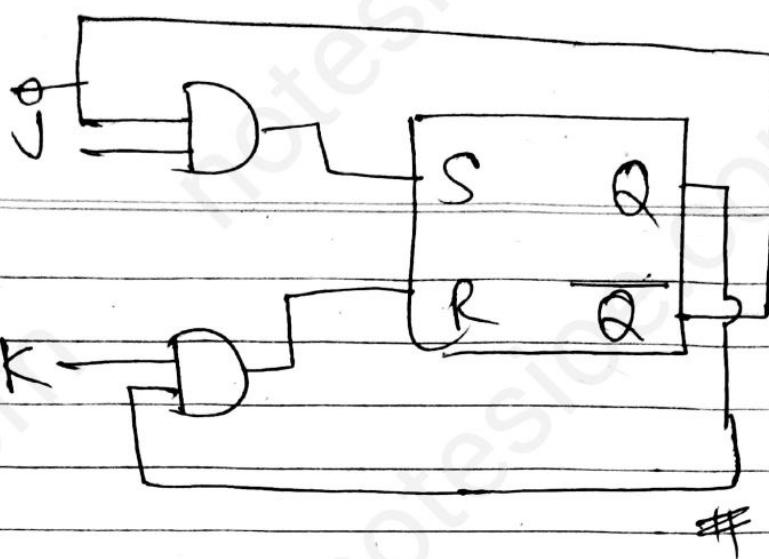
Char. table

Ex-01 table

J	K	Q_n	00	01	11	10
0	0	X	:	:	:	:
1	1	X	1			

$$S = \bar{J} \bar{Q}_n$$

R	J	Q_n	00	01	11	10	K	Q_n
0	0	X	:	1	1	X	1	
1	1							



Date _____
Page _____

$\underline{JK} \rightarrow \underline{SR}$

Given

Desired

Me

Phani

Excited

Characteristics.

Q _n	Q _{n+1}	J	K
0	0	0	x
0	1	-1	x
1	0	x	1
1	1	x	0

S	R	Q _n	Q _{n+1}	J	K
0	0	0	0	0	x
0	0	1	1	x	0
0	1	0	0	0	x
0	1	1	0	x	1
1	0	0	1	1	x
1	0	1	1	x	0
1	1	0	x	x	x
1	1	1	x	x	x

J	R	Q _n	Q _{n+1}
0	0	00	01
1	1	11	10

$J = S$

K	Q _n	Q _{n+1}
x	x	1
x	x	x

$K = R$

Q Simplify $f(A, B, C, D) = \Sigma m(1, 2, 3, 8, 9, 10, 11, 14)$
 $+ d(0, 4, 12)$ by using k-map & write
 its standard POS expression. [6]

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

$A + \bar{B}$

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

$$\text{In POS, } Y = (A + \bar{B}) \cdot (\bar{B} + \bar{D})$$

$$\{(X + C\bar{C}) \Rightarrow (X + C) \cdot (\underline{X + \bar{C}})\}$$

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

Various representation of flip flops:

for SR

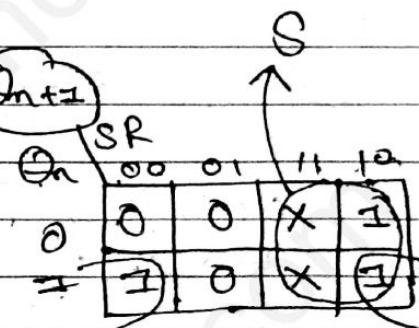
Truth table

next state

Clock	S	R	Q_{n+1}
0	X	X	$Q_n \rightarrow \text{P.S.}$
↑	0	0	Q_n (Memory)
↑	0	1	0
↑	1	0	1
↑	1	1	Invalid

Characteristics Table

clock	Q_n	S	R	Q_{n+1}
↑	0	0	0	0
↑	0	0	1	0
↑	0	1	0	1
↑	0	1	1	X
↑	1	0	0	1
↑	1	0	1	0
↑	1	1	0	1
↑	1	1	1	X



Excitation table

$$Q_{n+1} = S + Q_n \bar{R} \quad \#$$

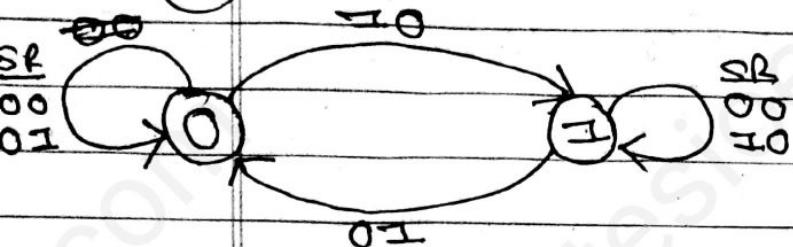
Transition	SR FF	JK FF	D FF	T FF
$Q_n \rightarrow Q_{n+1}$	S R	J K	D	T
0 0	0 X	0 X	0	0
0 1	1 0	1 X	1	1
1 0	0 1	X 1	0	1
1 1	X 0	X X	X	0

#

② FF as finite state Machine

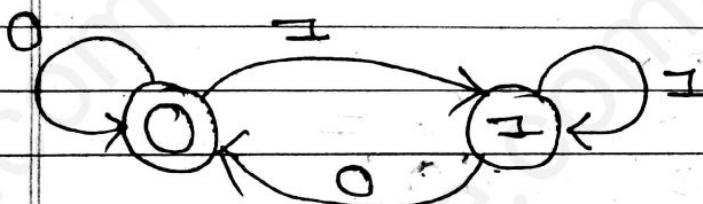
Date _____
Page _____

SR \rightarrow 00, 01, 10, 11

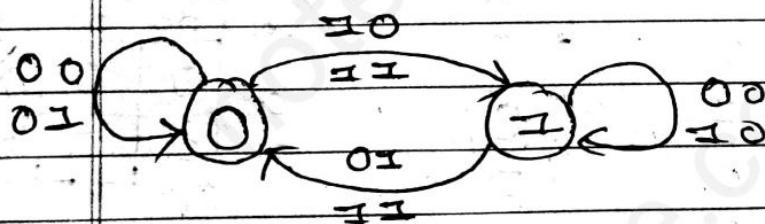


③ SR FF

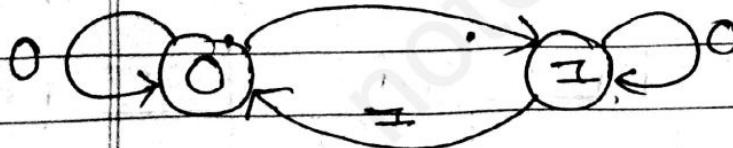
ii) D flip-flop:



iii) JK flip flop:



iv) T flip flop:



Arithmetic Circ.

Date _____
Page _____

- (1) Binary Addition
- " Subtraction
- .

Complement:

1) r's complement:

$$r\text{'s complement of } N = \underline{r^n} - N.$$

n = no. of digits

N = number

r = radix (number of digit)

e.g.

10's complement of $(\underline{305})_{10}$

$$= 10^3 - 305$$

$$= 1000 - 305 = \underline{\underline{695}}$$

10's comp. of $(\underline{0.252})_{10}$

$$= 10^0 - 0.252$$

$$= 1 - 0.252 = \underline{\underline{0.748}}$$

=

2's complement of $(\underline{\underline{10101}})_2$

$$= 2^5 - \underline{\underline{10101}}$$

$$= \underline{\underline{100000}} - \underline{\underline{10101}}$$

$$= \underline{\underline{10111}}$$

2) $(r-1)$'s complement.

$$(r-1)\text{'s complement of } N = \underline{\underline{r^m}} - \underline{\underline{r^m - N}}$$

Q

9's complement of $(25 \cdot 639)_{10}$

Date _____
Page _____

$$(25 \cdot 639)_{10} = 10^2 - 10^{-3} - 25 \cdot 639 = 73.36 \text{ #}$$

I's complement of $(101100)_2$

$$= 2^6 - 2^0 - 101100$$

$$= 111111 - 101100$$

$$= 0100 = 100 \text{ #}$$

1600000

2^s complement

Arithmetic Building Blocks

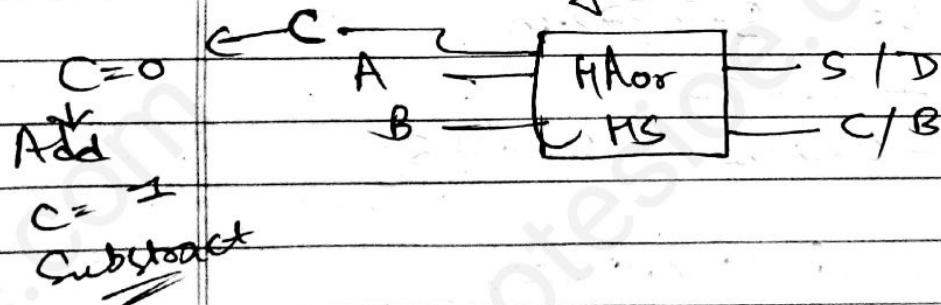
i) HA

ii) FA

iii) HS

iv) FS

Q Design half subtractor ~~and~~ and half adder
on a single circuit.



Input

C A B

Output 1

$y(\text{out})$

sum/diff

Output 2

y_{out}

carry/borrow

0 0 0

0

0

0 0 1

1

0

0 1 0

1

0

0 1 1

0

1

1 0 0

0

0

1 0 1

1

1

1 1 0

1

0

1 1 1

0

0

Row 1

	AS	00	01	11	10
C	00	01	11	10	00
=	0	1	1	0	1

Row 2

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

$$Y_{\text{out}1} = \overline{AB} + A\overline{B}$$

$$= A \oplus B$$

$$Y_{\text{out}2} = \overline{C}AB + C\overline{A}B$$

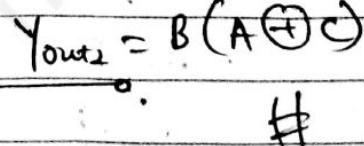
$$= B(A\overline{C} + \overline{A}C)$$

$$= B(A \oplus C)$$

A B C



$$Y_{\text{out}1} = A \oplus B$$

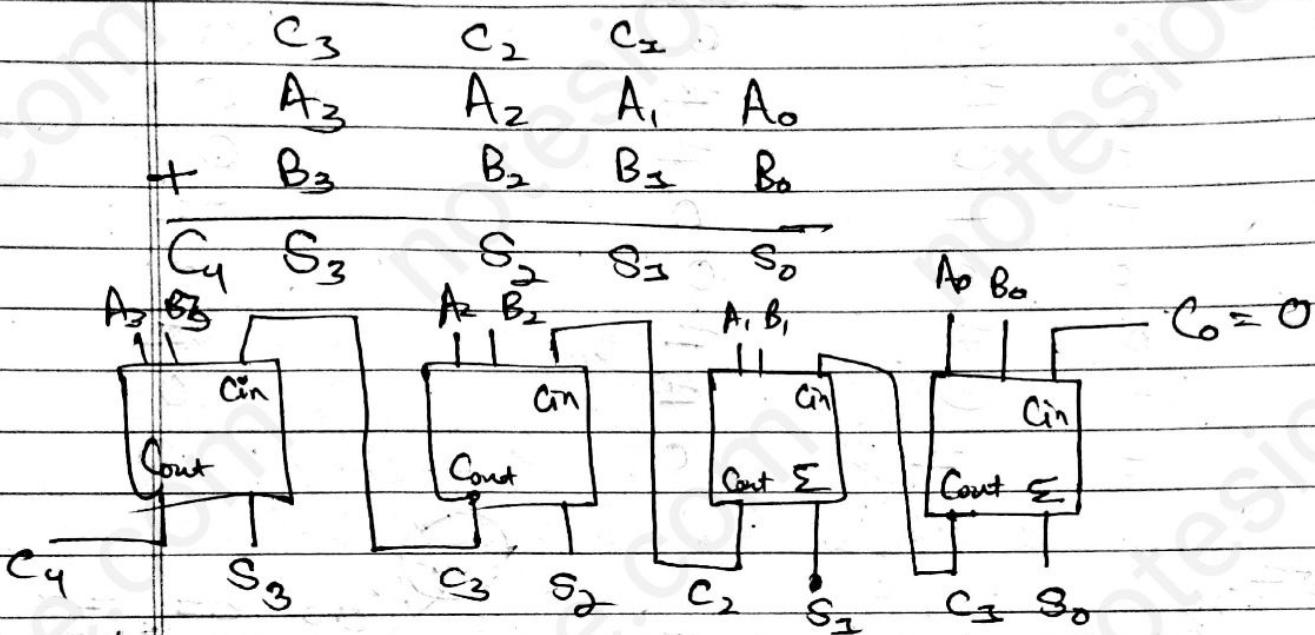


$$Y_{\text{out}2} = B(A \oplus C)$$

Binary Parallel Adder:

Date _____
Page _____

Q9 Construct a n -bit binary Parallel adder:



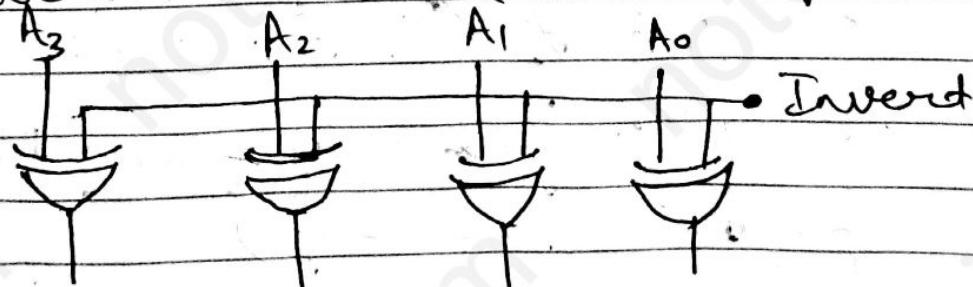
n -bit parallel adder #

Binary Adder Subtractor:

Controlled inverter:

when Invert = 0 \Rightarrow Inv. the I/p data

when Invert = 1 \Rightarrow Inv. I's complement of I/p.



Invert

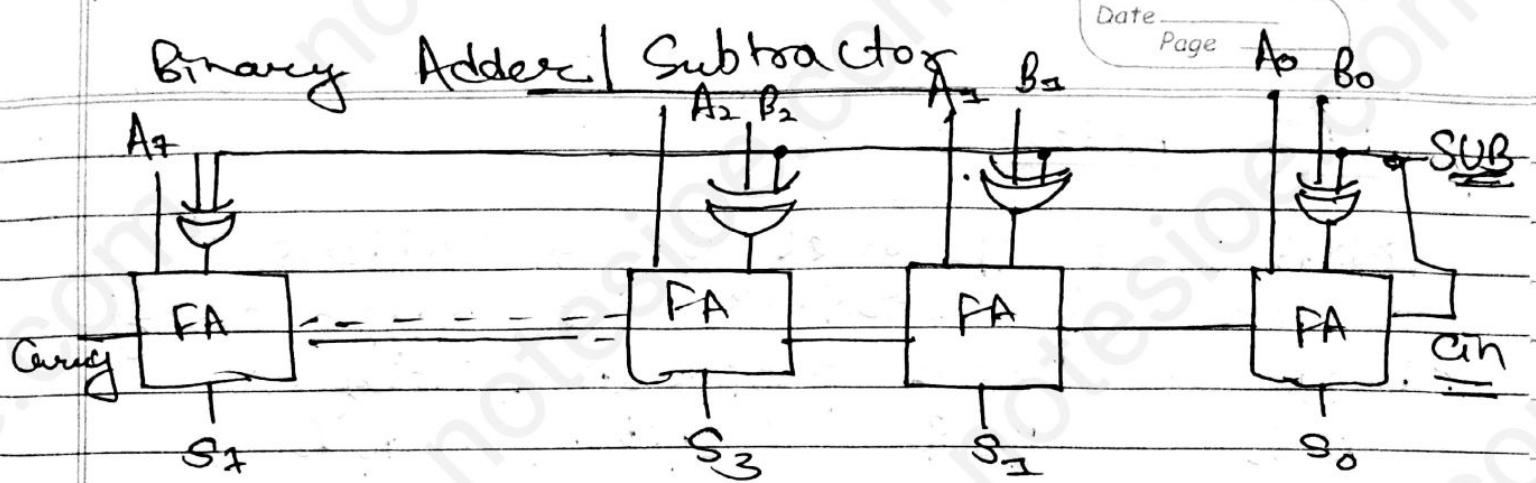


$$A \quad F \\ 0 \rightarrow 0 \quad (F=A)$$

$$1 \rightarrow 1 \quad (F=A)$$

$$2 \rightarrow 0 \quad (F=\overline{A}) \quad \#$$

Binary Adder | Subtractor



when $SUB = LOW$, Ckt performs Addition,

$$\begin{array}{ccccccccc}
 & A_7 & A_6 & A_5 & A_4 & A_3 & A_2 & A_1 & A_0 \\
 (+), B_7 & B_6 & B_5 & B_4 & B_3 & B_2 & B_1 & B_0 \\
 C_8 & S_7 & S_6 & S_5 & S_4 & S_3 & S_2 & S_1 & S_0 \\
 & & & & & & \curvearrowleft 0 \leftarrow SUB
 \end{array}$$

~~Perform Subtraction~~

$$\begin{array}{l}
 A - B \rightarrow \\
 A + (\text{2's comp. of } B) \leftarrow I \leftarrow SUB
 \end{array}$$

$$A + (I^s \text{ complement of } B + I)$$

$$I \rightarrow 0$$

$$\begin{array}{ccccccccc}
 A_7 & A_6 & A_5 & A_4 & A_3 & A_2 & A_1 & A_0 \\
 - & - & - & - & -B & -B & - & - \\
 \cancel{B_7} & \cancel{B_6} & \cancel{B_5} & \cancel{B_4} & \cancel{B_3} & \cancel{B_2} & \cancel{B_1} & \cancel{B_0} \\
 \cancel{S_7} & \cancel{S_6} & \cancel{S_5} & \cancel{S_4} & \cancel{S_3} & \cancel{S_2} & \cancel{S_1} & \cancel{S_0}
 \end{array}$$

~~#~~

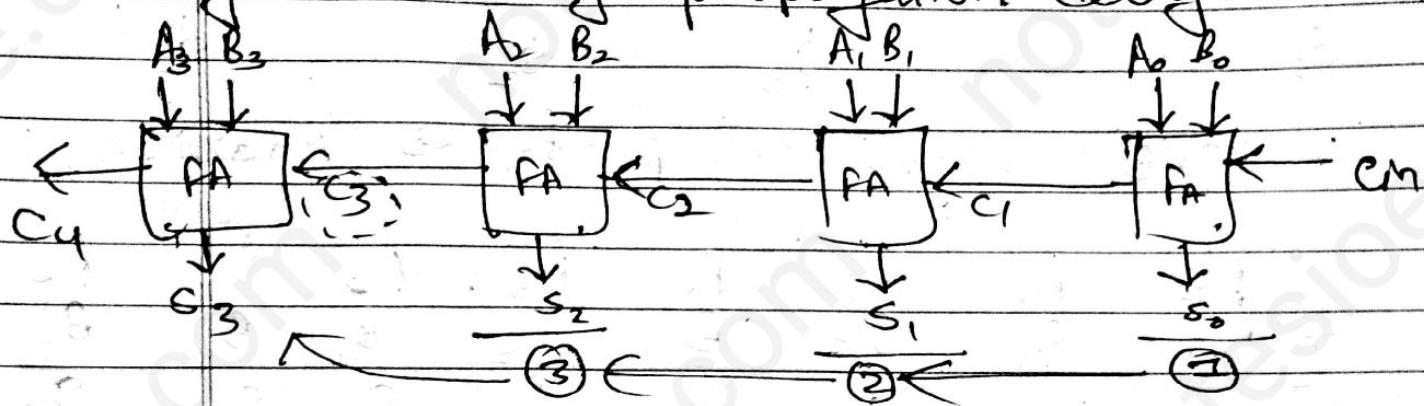
1, 2, 3, 5, 6

Date _____
Page _____

FAST Adder :

Limitation of parallel Binary Adders.

→ major limitation is it's ripple carry delay or carry propagation delay.



In Parallel adders, o/p is available only after carry is propagated through each of lower bit adders to the higher one.

~~if 1 FA \rightarrow 20 ns delay.~~

~~Total time for final o/p.
 $\rightarrow 4 \times 20 \text{ ns} = 80 \text{ ns}$~~

~~16 bit~~
 $16 \times 20 = 320 \text{ ns}$

In order to speed up Adder circuit,
fast adders are used.

Carry look ahead / PAST adder

Date _____
Page _____

→ It is based on principle of looking at lower order bits in order to generate higher order carry.

Two functions:

- i) Carry propagation (P_i)
- ii) Carry generation (G_i).

From full adder circuit we get:

$$C_{out} = AB + AC_{in} + BC_{in}$$

Generalizing for i^{th} term:

$$G = A_i B_i + A_i C_{i-1} + B_i C_{i-1} = \textcircled{A_i B_i} + \textcircled{C_{i-1}} \textcircled{(A_i + B_i)}$$

This can be written as:

$$\underline{G} = \underline{C_i} + \underline{P_i} \underline{C_{i-1}}$$

where, C_i = Carry generation = $A_i B_i$

P_i = Carry propagation = $A_i + B_i$

If $A_i B_i = 1$, then i^{th} stage will generate carry no matter previous stage generates or not.

If $A_i + B_i = 1$, then this stage will propagate carry if available from preceding stage to next stage.

Here C_i & P_i are obtained after one gate delay once A & B are placed.

$$G = A_i B_i + C_{i-1} (A_i + B_i)$$

$$C_i = C_0 + P_0 C_{i-1}$$

Date _____
Page _____

Starting from LSB:

$$C_0 = C_0 + P_0 C_{-1} \quad [C_{-1} \text{ will be '0' normally}]$$

$$C_1 = C_1 + P_1 C_0$$

$$\Rightarrow C_1 = C_1 + P_1 (C_0 + P_0 C_{-1})$$

$$= C_1 + P_1 C_0 + P_1 P_0 C_{-1}$$

$$C_2 = C_2 + P_2 (C_1 + P_1 C_0 + P_1 P_0 C_{-1})$$

$$= C_2 + P_2 C_1 + P_2 P_1 C_0 + P_2 P_1 P_0 C_{-1}$$

:

These eqn can be realized with multi-level input AND (2nd level) and OR gate (3rd level) in two levels. Max. delay = 1 + 2 = 3 gate delay for carry.

1st level (one gate delay) → to produce $C_0, P_0, C_1, P_1, \dots$

2nd level (one gate delay) → Anding them → $P_0 C_0, P_1 P_0 C_{-1}, \dots$

3rd level (one gate delay) → Oring them.

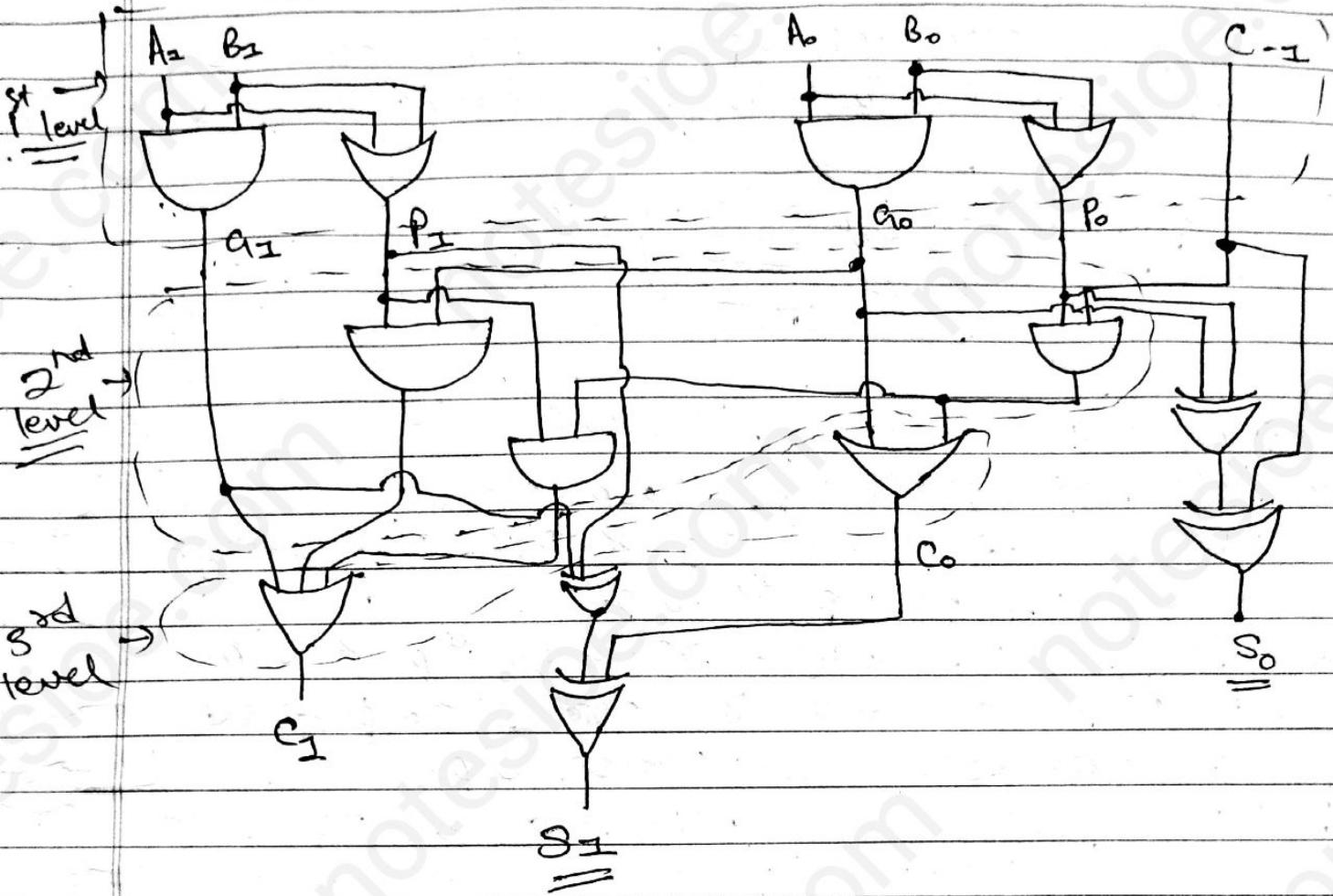
For sum = $A_i \oplus P_i \oplus C_{i-1}$, it requires 2 gate delay for two XOR gates.

∴ Total gate delay = 3 + 2 = 5.

But for ripple carry adder, it requires (2n) gate delay.

2-bit fast adder

Date _____
Page _____



#

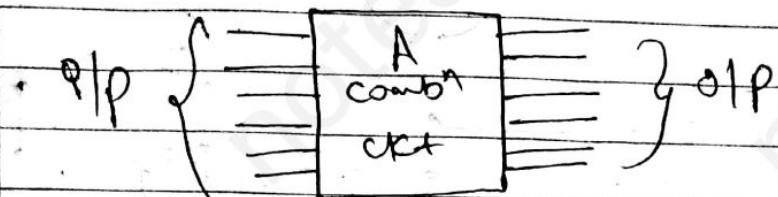
Ch-4

Combinational logic Processing Ckt

Date _____
Page _____

Combinational ckt:

→ logical ckt, the o/p of which depends upon the combination of present inputs.

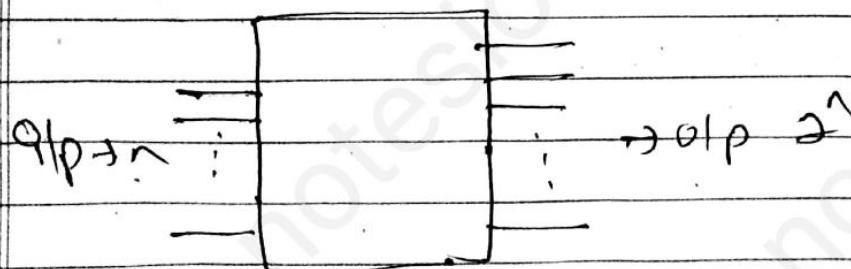


Design step:

- ① → We will be given a problem.
- ② → we determine no. of I/p & O/p and assign letter symbol to I/p & O/p.
- ③ → We write truth table relating I/p & O/p.
- ④ → We make k-map & obtain simplified Boolean expression for each O/p.
- ⑤ → Lastly we draw logical combinational ckt.

M.1

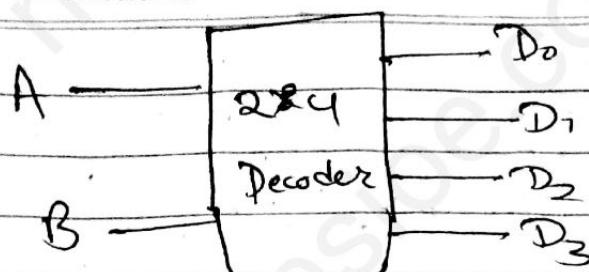
Decoder:



→ combinational ckt that convert binary information from 'n' I/p lines to max. 2^n unique lines.

2:4 Decoder

Date _____
Page _____



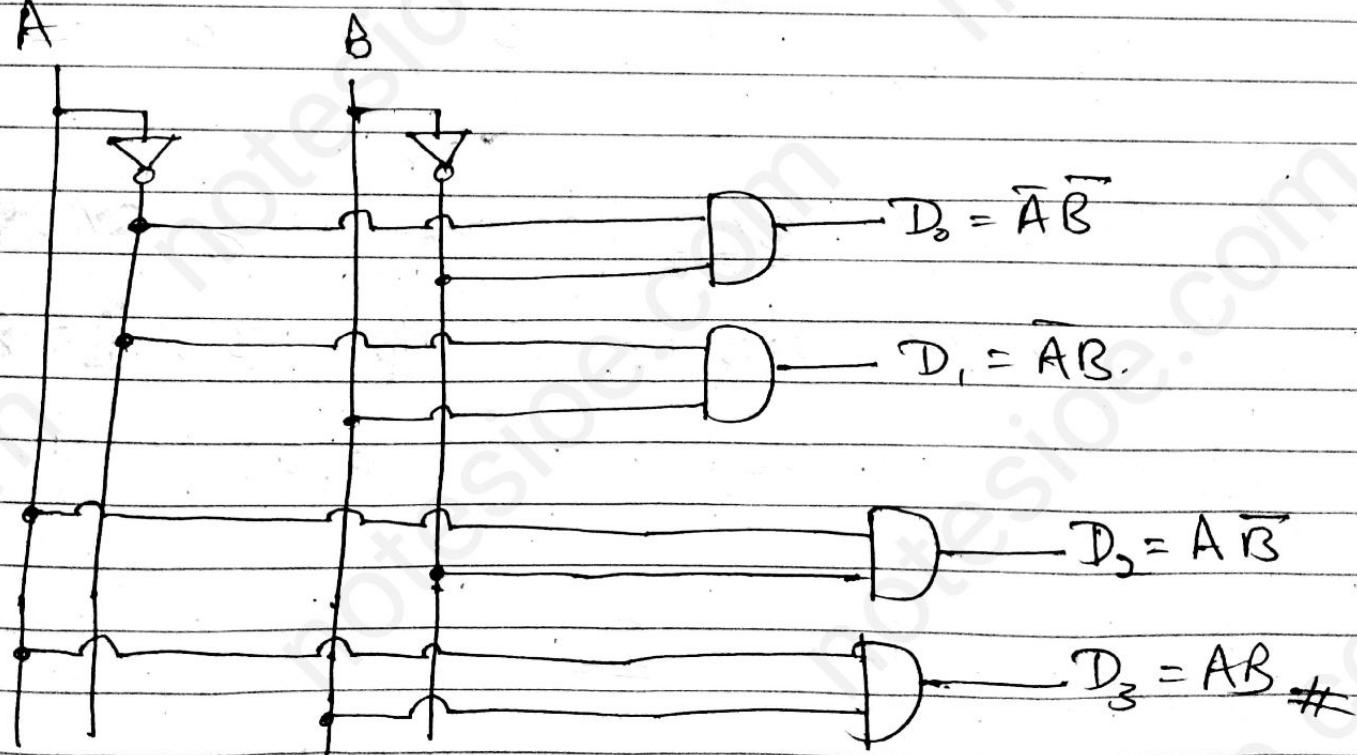
Block diagram of 2:4 decoder

Input

A	B
0	0
0	1
1	0
1	1

Output

D ₀	D ₁	D ₂	D ₃
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

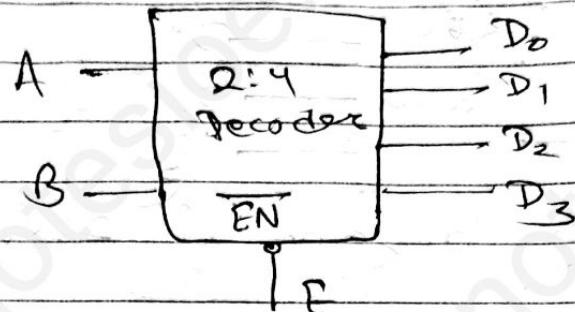


(3x8)

Date _____
Page _____

Decoder with enable:

I/P



I/P

O/P

E	A	B	D ₀	D ₁	D ₂	D ₃
1	X	X	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	1	1	0	0	0	1

Q Construct 3x8 decoder using 2x4 decoder

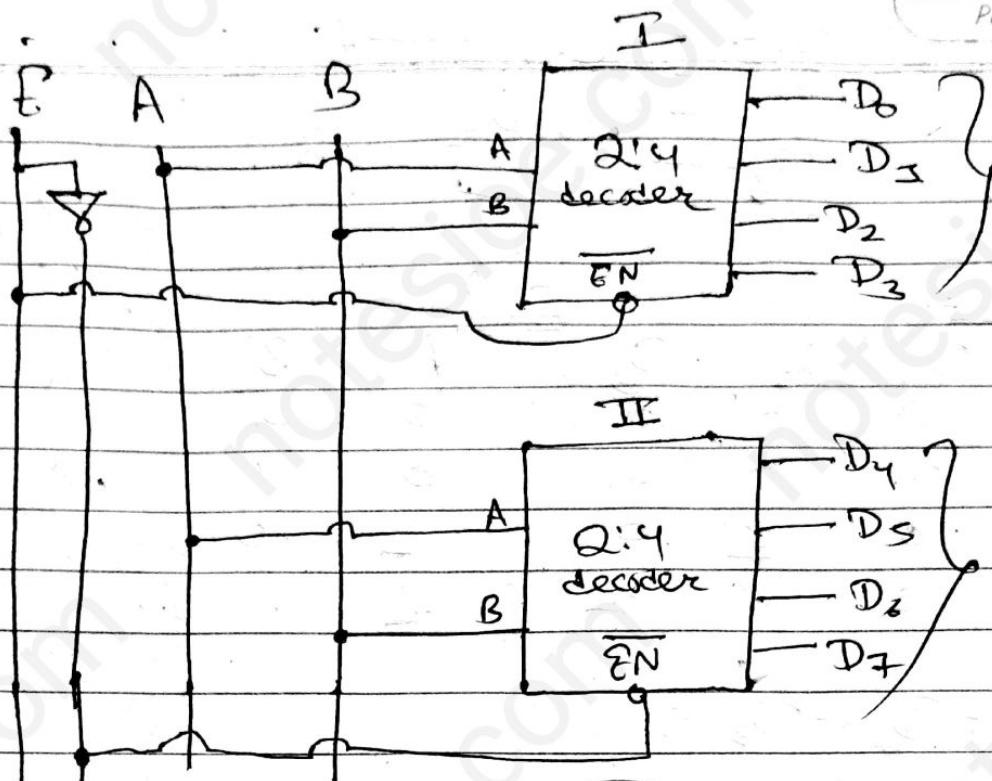
for 3x8 decoder, no. of I/P = 3, O/P = 8

for 2x4 decoder, " " = 2, " " = 4

E	A	B	<u>D</u>
0	0	0	D ₀
0	0	1	D ₁
0	1	0	D ₂
0	1	1	D ₃
1	0	0	D ₄
1	0	1	D ₅
1	1	0	D ₆

2x4 decoder, when E=0

2x4 decoder, when E=1



BCD to decimal \rightarrow

$4:10 =$

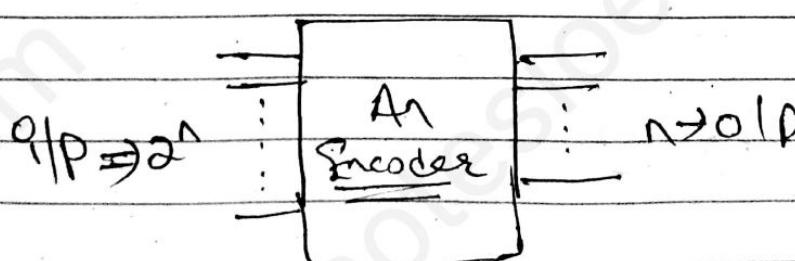
(ABC)

$(D_0 - D_9) =$

$4:16$

#

Encoder

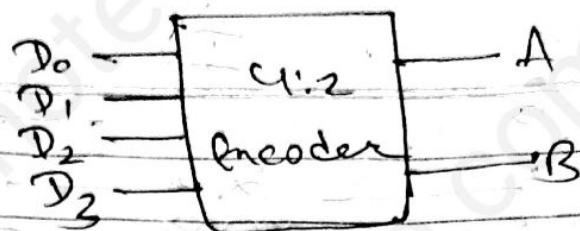


Block diagram of encoder.

$4:2$ Encoder

4:2

Date _____
Page _____

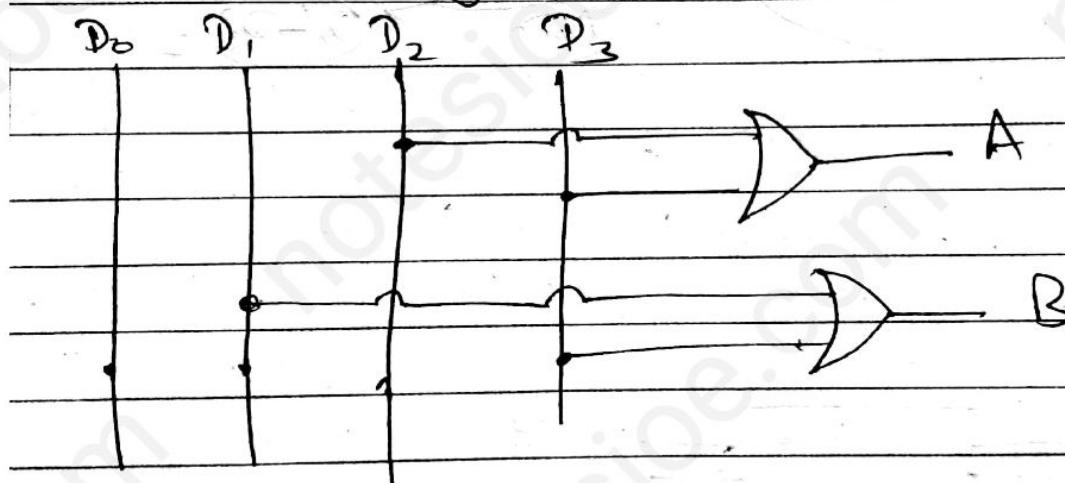


I/P O/P

	D ₀	D ₁	D ₂	D ₃	A	B
1	0	0	0	0	0	0
0	1	0	0	0	0	1
0	0	1	0	0	1	0
0	0	0	1	0	1	1

$$A = D_2 + D_3$$

$$B = D_1 + D_3$$



4x2 encoder

Q3

Doneedit

If 2 QIP are enable at same time O/P is undefined.

Priority Encoder

(4x2) =

Date _____
Page _____

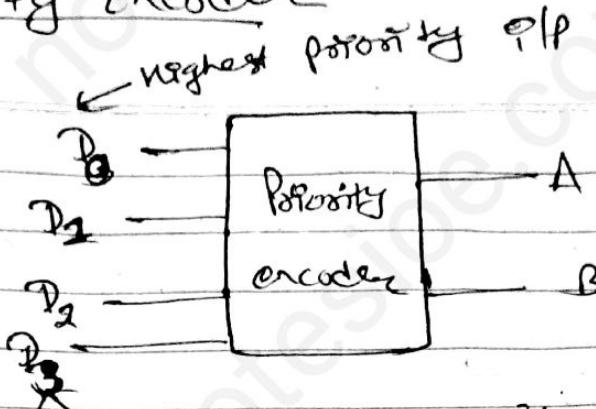


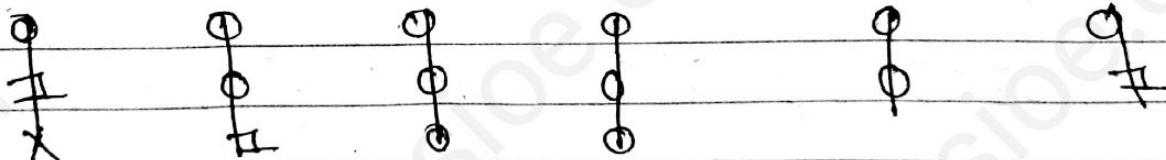
Fig: Block diagram of priority encoder.

U:2 P. encoder

QIP

QIP

D ₀	D ₁	D ₂	D ₃	A	B
0	0	0	0	X	X
1	X	X	X	0	0
0	1	X		0	1
0	0	1	X	1	0
0	0	0	1	1	1



$$A = \overline{D_0} \overline{D_3} D_2 + \overline{D_0} \overline{D_3} \overline{D_2} D_3 = \overline{D_0} \overline{D_1} (D_2 + D_3)$$

$$B = \overline{D_0} D_1 + \overline{D_0} \overline{D_2} \overline{D_3} \#$$

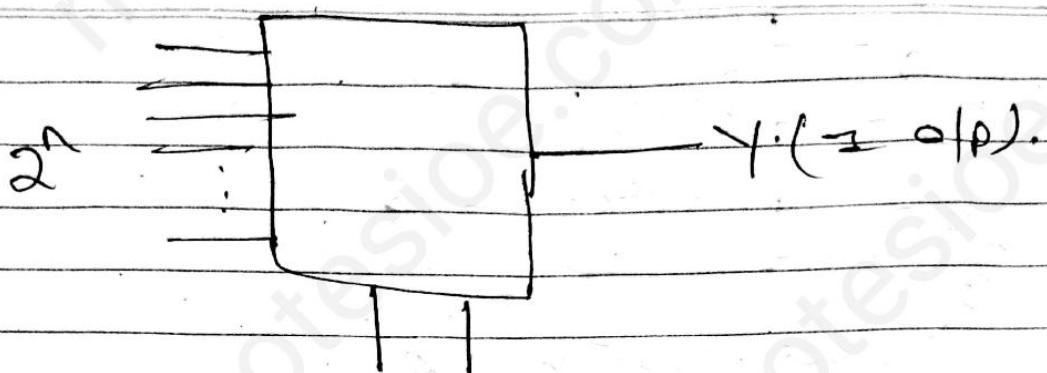
match

8:03

2

MUX

Date _____
Page _____



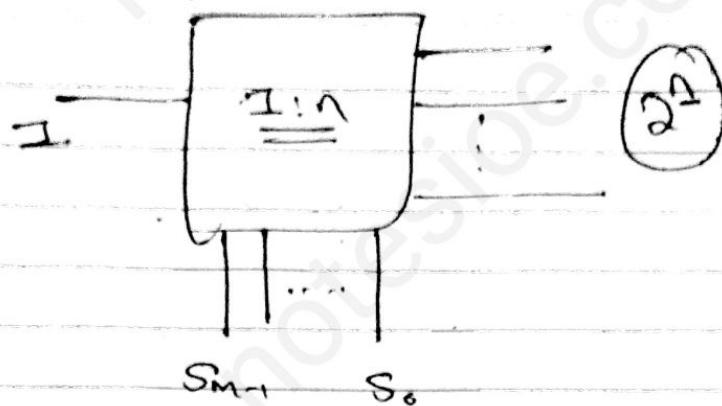
n select
lines

4: 1 (MUX)

<u>I_D</u>	<u>S₃</u>	<u>S₂</u>	<u>Y</u>
I ₀	0	0	I ₀
I ₁	0	1	I ₁
I ₂	1	0	I ₂
I ₃	1	1	I ₃

$$Y = \overline{S_3} \overline{S_2} I_0 + \overline{S_3} S_2 I_1 + S_3 \overline{S_2} I_2 + S_3 S_2 I_3$$

Dowx



(1:4)

IP

	IP	IP		011			
D	S_1	S_0	γ	γ_1	γ_2	γ_3	
D	0	0	D	0	0	0	
D	0	H	0	D	0	0	
D	H	0	0	0	D	0	
D	H	H	0	0	0	D	#