Electronics: The study of motion of electron inside a semi--conductor is known as Electronics.

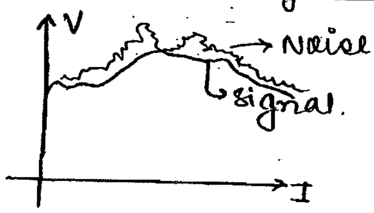
Note: there is a contsolled conductivity in the case of semi--conductor

Gate: it is an electronic switching circuit made up of semiconductor switching devices.

Logic gates: Electronic switching ckt ie gate with logical ideas implementation by Mr. boolean are logical gates.

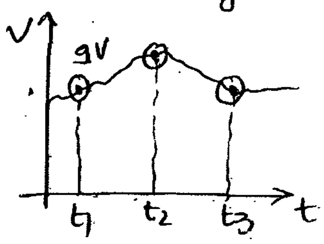
Need of digital signal:

Disadvantages of analog signal:



(a) More bandwidth required.
(b) More power consumption.
(c) it is effected by noise more. (at recieving end if we amplify the signal, noise is also amplified)
(d) no chance for encryption.

advantages of digital signal:



(a) Less bandwidth required.
(b) Less power consumption.
(c) it is effected less by noise.
(d) encryption is possible in digital signal.

Encryption is using digital data in the form of codes.

Block diagram:

Analog i/p → [ADC] → [Digital system] → [DAC] → Analog O/P

EX→ μp, ALU

main systems: μp, ALU, computers, calculators ⇒ subsystems: Registers, counters, flip flops ⇒ Building blocks: logic gates.

⇓

contains diodes, BJT, mosfet

SYLLABUS

hapter 1: Basics:
- Boolean algebra.
- logic gates.
- K-maps.
- Number system.
- codes
- Data representation.

hapter 2: combinational circuits:
- Arthematic circuits
  - → half adder, half subtractor, full adder, full subtractor.
  - → parallel binary adder, look ahead carry adder
  - → BCD adder, 2's compliment adder subtractor.
- combinational circuits
  - → Decoder, multiplexer, demux, Encoder, decoder
  - → comparator, code converter, parity generator, checker.

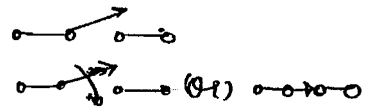
nit/chaper 3: sequential circuits:
- → Latches, flipflops, Registers, counters, state machines (mealy more, moore model).

hapter 4: logic families.
- → RTL, DTL, HTL, TTL, ECL
- → MOS logic: NMOS, PMOS, CMOS (mainly for Gate Exam)

hapter 5: ADC and DAC conversions.

Note: for 'n' switches we get 2^n possible states.

switch: → $s=0$ →
→ $s=1$ →



{ one switch has two states

S_1	
0	OFF
1	ON

⇨ two switches have four states.

Boolean logical ideas:

- These are categorized into three ways:
  - → two function that produce constant '0' or '1'. (Null, identity)
  - → four functions with unimary operations, complementary (→ NOT) and transfer (→ Buffer).
  - → ten functions with binary operations. (AND, OR, NAND, NOR, EX-OR, EX-NOR, Inhibition, Implication).

Note: for n input variables, we get (2^n) combinations and (2^{2^n}) $[2^{2n}]$ possible functions.

Ex = $n=2$ → $2^n \to 2^2 \to 4$ combinations.
→ $2^{2^n} \to 2^{2^2} \to 16$ functions / operations.

Truth tables:

		Null	AND	$x\cdot\bar{y}$ (inhibition)	x transfer Buffer	$\bar{x}\cdot y$ (inhibition)	y transfer Buffer	Ex-OR $x\oplus y$	$x+y$	$\overline{x+y}$	coincidence $x\odot y$ (EXNOR)	NOT $\bar{y}$	$x+\bar{y}$ (implication)	NOT $\bar{x}$	$\bar{x}+y$ (implication)	NAND $\overline{(x\cdot y)}$	Identity
X	Y	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

f_0 = Null operation

$f_1 = x\cdot y = x \cap y$ (AND)

$f_2 = x\cdot\bar{y} = x/y$ = (x but not y) ⇒ Inhibition.

f_3 = x transfer ⇒ Buffer

$f_4 = \bar{x}\cdot y = y/x$ = (y but not x) ⇒ inhibition.

f_5 = y transfer ⇒ Buffer.

$f_6 = x\oplus y = \bar{x}\cdot y + x\bar{y}$ ⇒ Ex-OR (for diff i/p → o/p is high)

$f_7 = x+y = x\cup y$

$f_8 = \overline{x+y} = x \downarrow y$ ⇒ NOR

$f_9 = x \odot y = \bar{x}\bar{y} + xy$ ⇒ when i/p same → o/p is high
⇒ <u>coincidence logic gate</u> (or) <u>Equivalence logic gate</u>.

$f_{10} = \bar{y}$ = complementary = <u>NOT</u> gate.

$f_{11} = x + \bar{y}$ = $x \subset y$ = If y then x ⇒ <u>implication</u>

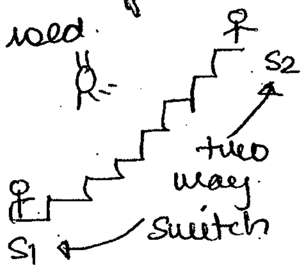
$f_{12} = \bar{x}$ = complementary = <u>NOT</u> gate.

$f_{13} = \bar{x} + y$ = $x \supset y$ = If x then y = <u>implication</u>

$f_{14} = \overline{(x \cdot y)}$ = <u>NAND</u> = $x \uparrow y$

$f_{15} = 1$ = <u>Identity</u>.

Note: for staircase and accelerator operation EX-OR logic gate is used.



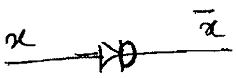
S_2	S_1	bulb $= S_2 \oplus S_1$
0	0	0
0	1	1
1	0	1
1	1	0

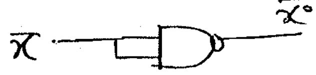
⇒ Basic logic gates: NOT, AND, OR

⇒ combinational logic gates: NAND (AND NOT) & NOR (NOT OR)

⇒ special purpose logic gates: EX-OR, EX-NOR (Arithmetic operations).

∘ NAND & NOR are said to be universal logic gates.

NOT 

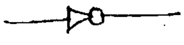
NOT from NAND ⇒ 

Demorgan's theorem.

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

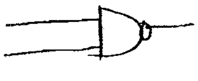
$\overline{x + y} = (\bar{x} \cdot \bar{y})$

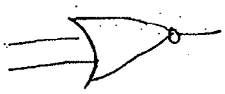
⇒ <u>symbols</u>:

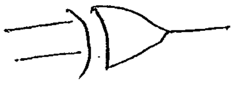
NOT 

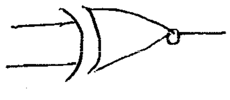
AND 

OR 

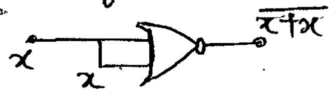
NAND 

NOR 

EX-OR 

Ex-NOR 

NOT from NOR



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

⊛ Shortcut: No of Nand and NOR gate required:

Gate	No of Nand gates	No of NOR gates
→ NOT	1	1
→ AND	2	3
→ OR	3	2
→ Ex-OR	4	5
→ Ex-NOR	5	4

NAND to NOR (4) (OR) NOR to NAND (4)

⊛ Design of logic gates using universal logic gates:

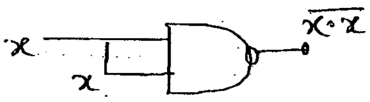
- universal logic gates are NAND and NOR.

(a) NOT from NAND:

→ output of NOT ⇒ $\bar{x}$

→ No of Nand gates req = 1

Circuit:



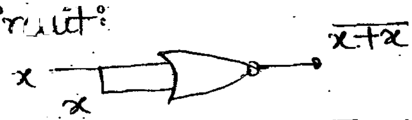
→ $\overline{x\cdot x} = \bar{x}+\bar{x} \Rightarrow \bar{x}$

(b) NOT from NOR:

→ output of NOT gate = $\bar{x}$

→ No of ~~NAND~~ NOR gates req = 1

circuit:



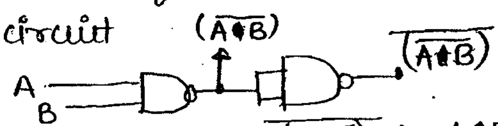
→ $\overline{x+x} = \bar{x}\cdot\bar{x} = \bar{x}$

(c) AND from NAND:

→ output of AND ⇒ $x\cdot y$

→ No of NAND req = 2

circuit

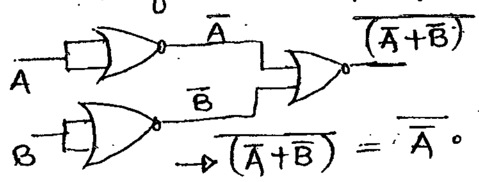


$\overline{(\overline{A\cdot B})} \Rightarrow A\cdot B$

(d) AND from NOR:

→ output of AND = $x\cdot y$

→ No of NOR req = 3

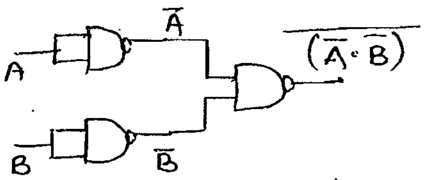


→ $\overline{(\bar{A}+\bar{B})} = \bar{\bar{A}}\cdot\bar{\bar{B}} = A\cdot B$

(e) OR from NAND:

→ output of OR ⇒ $x+y$

→ no of Nand ⇒ 3

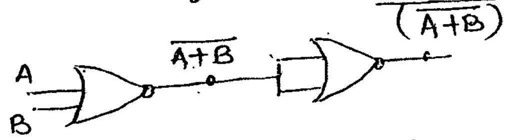


→ $\overline{(\bar{A}\cdot\bar{B})} \Rightarrow \bar{\bar{A}}+\bar{\bar{B}} = A+B$

(f) OR from NOR:

→ output of OR = $x+y$

→ No of NOR = 2

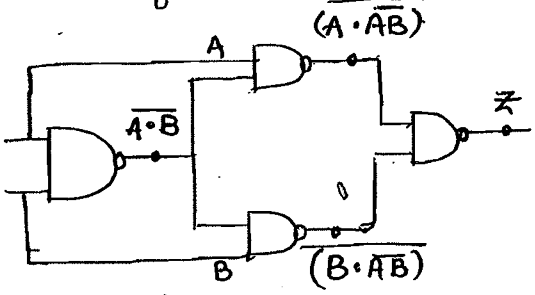


→ $\overline{(\overline{A+B})} = A+B$

g) EXOR from NAND: $(A \oplus B)$

→ output of EX-OR = $\bar{A}B + A\bar{B}$

→ no of NAND = 4



$z = \overline{\left[\overline{(A\cdot\overline{AB})}\cdot\overline{(B\cdot\overline{AB})}\right]}$

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

$z = A\cdot\overline{AB} + B\cdot\overline{AB}$

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

$z = \underbrace{A\cdot\bar{A}}_{\to 0} + A\bar{B} + B\cdot\bar{A} + \underbrace{B\bar{B}}_{\to 0}$

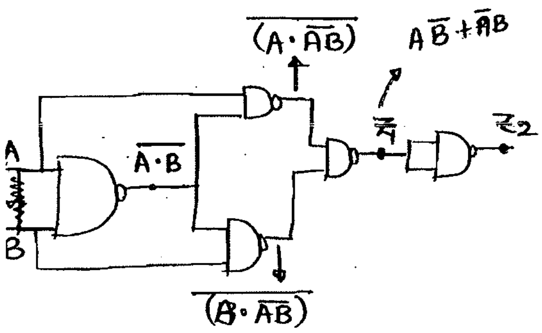
$= A\bar{B} + B\bar{A}$

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

) EX-NOR from NAND: $(A \odot B)$

→ output of EXNOR = $\bar{A}\bar{B} + AB$

→ no of nand req = 5



$z_1 = A\bar{B} + \bar{A}B$

$z_2 = \overline{z_1}$

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

$= \overline{A\bar{B}}\cdot\overline{\bar{A}B}$

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

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

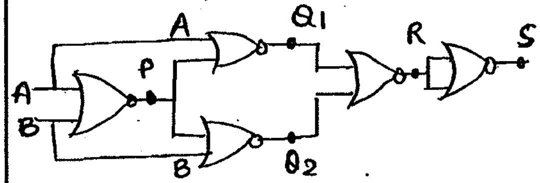
$= \underbrace{A\bar{A}}_{\to 0} + \bar{A}\bar{B} + BA + \underbrace{B\bar{B}}_{\to 0}$

$z_2 = \bar{A}\bar{B} + AB$

(h) EX-OR from NOR:

→ output of EX-OR = $\bar{A}B + A\bar{B}$

→ no of NOR = 5



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

$Q_1 = \overline{(A+P)} = \overline{(A+\overline{(A+B)})} = \bar{A}\cdot\overline{\overline{(A+B)}}$

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

$Q_2 = \overline{(B+P)} = \overline{(B+\overline{(A+B)})} = \bar{B}\cdot\overline{\overline{(A+B)}}$

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

$R = \overline{(Q_1+Q_2)} = \overline{[\bar{A}\cdot(A+B) + \bar{B}(A+B)]}$

$= \overline{[\underbrace{\bar{A}A}_{\to 0} + \bar{A}B + \bar{B}A + \underbrace{\bar{B}B}_{\to 0}]} = \overline{[\bar{A}B + \bar{B}A]}$

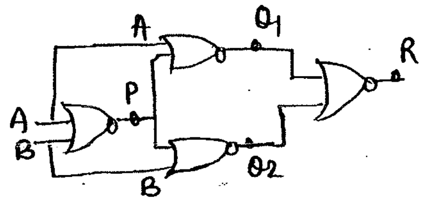
$R = \overline{(\bar{A}B + A\bar{B})}$

$S = \bar{R} = \overline{\overline{(\bar{A}B + A\bar{B})}} = \bar{A}B + A\bar{B}$

(J) EX-NOR from NOR:

→ output of EXNOR = $\bar{A}\bar{B} + AB$

→ no of NOR = 4



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

$Q_1 = \overline{(A+P)} = \overline{A+\overline{(A+B)}} = \bar{A}\cdot\overline{\overline{(A+B)}}$

$= \bar{A}\cdot(A+B) = \underbrace{A\bar{A}}_{\to 0} + \bar{A}B = \bar{A}B$

$Q_2 = \overline{(B+P)} = \overline{(B+\overline{A+B})} = \bar{B}\cdot\overline{\overline{(A+B)}}$

$= \bar{B}\cdot(A+B) = \bar{B}A + \underbrace{\bar{B}B}_{\to 0} = A\bar{B}$

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

$= \overline{\bar{A}B}\cdot\overline{A\bar{B}}$

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

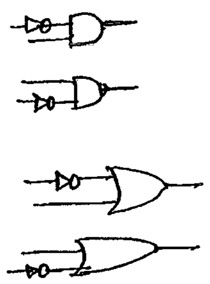
$= (A+\bar{B})\cdot(\bar{A}+B)$

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

$= \underbrace{A\bar{A}}_{\to 0} + AB + \bar{B}\bar{A} + \underbrace{\bar{B}B}_{\to 0}$

Inhibition → $\bar{x}\cdot y$, $x\cdot\bar{y}$

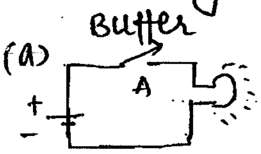
Implication → $\bar{x}+y$, $x+\bar{y}$



these are also universal logic gates.

⊛ switching circuits as logic gates:

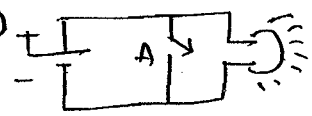
(a) Buffer



A	bulb
0	off(0)
1	ON(1)

$f = A$ (Buffer)

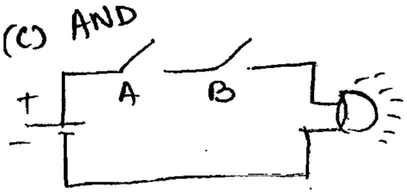
(b) NOT



A	bulb
0	ON(1)
1	Off(0)

$f = \bar{A}$ (NOT)

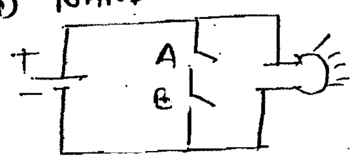
(c) AND



A	B	bulb
0	0	0
0	1	0
1	0	0
1	1	1

$f = A\cdot B$ (AND)

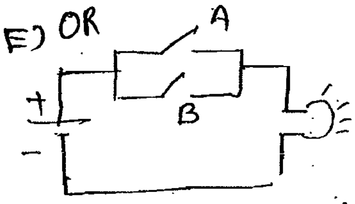
(d) NAND



A	B	bulb
0	0	1
0	1	1
1	0	1
1	1	0

$f = \overline{A\cdot B}$ = NAND.

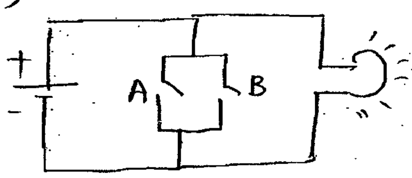
(E) OR



A	B	bulb
0	0	0
0	1	1
1	0	1
1	1	1

$f = A+B$ (OR)

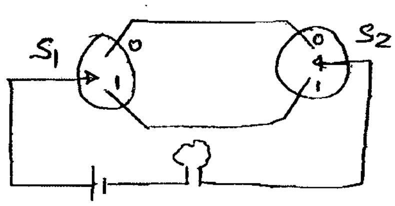
(F) NOR



A	B	bulb
0	0	1
0	1	0
1	0	0
1	1	0

$f = \overline{A+B}$ = NOR

(G)

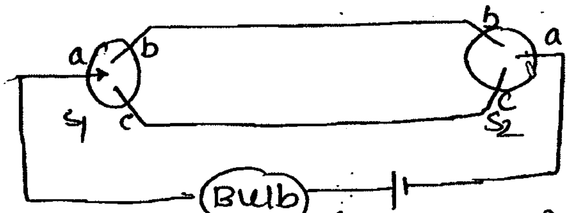


S_1	S_2	bulb
0	0	1
0	1	0
1	0	0
1	1	1

$f = S_1 \odot S_2$

⇒ workbook practice : CH2 : QT1, T2

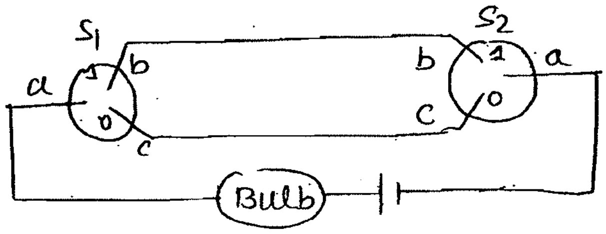
QT1 A two way switch has three terminals a, b and c. In ON position (logic value 1), a is connected to b, and in off position a is connected to c. Two of these switches are connected to bulb as shown in figure:



Which of the following expression, if true, will always result in lightning of bulb?

(a) $S_1 \cdot \overline{S_2}$ (b) $S_1 + S_2$ (c) $\overline{S_1 \oplus S_2}$ (d) $S_1 \oplus S_2$

sol^n



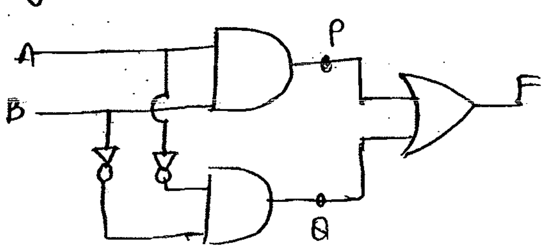
S_1	S_2	bulb
0	0	1
0	1	0
1	0	0
1	1	1

} $S_1 \odot S_2$

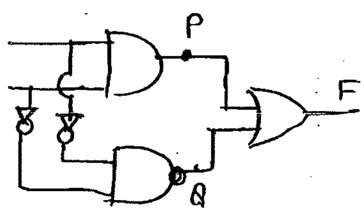
$S_1 \odot S_2 = \overline{S_1 \oplus S_2}$

T2 which one of the following represents coincidence logic gate?

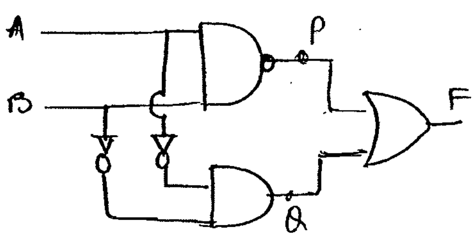
(a)



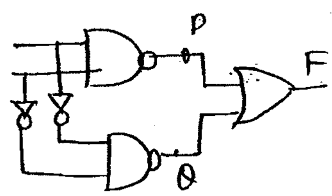
(c)



(b)



(d)

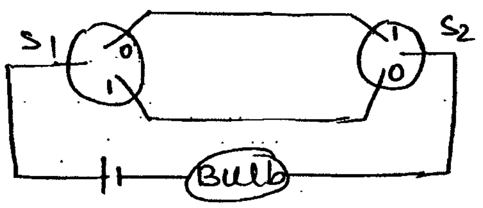


sol^n (a) truth table

A	B	P	Q	F
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

} $S_1 \odot S_2$ → coincidence logic gate

Q Ex:



S_1	S_2	bulb
0	0	0
0	1	1
1	0	1
1	1	0

} $S_1 \oplus S_2$

⊛ canonical forms:

	x	y	minterm → (product term)	f(ex)
m_0	0	0	$\bar{x}\cdot\bar{y}$	0
m_1	0	1	$\bar{x}\cdot y$	1 ✓
m_2	1	0	$x\cdot\bar{y}$	0
m_3	1	1	$x\cdot y$	1 ✓

$= f = \underset{01}{m_1} + \underset{11}{m_3}$
$= \bar{x}\cdot y + x\cdot y$
SOP
$f = \Sigma m(1,3)$

$\boxed{f = \Sigma m(1,3)}$

	x	y	maxterm → (sum term)	f(ex)
M_0	0	0	$x+y$	0 ✓
M_1	0	1	$x+\bar{y}$	1
M_2	1	0	$\bar{x}+y$	0 ✓
M_3	1	1	$\bar{x}+\bar{y}$	1

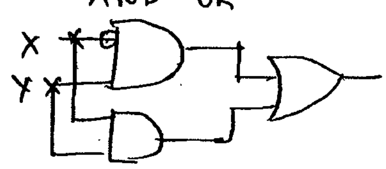
$f = M_0 + M_2$
$= (x+y)\cdot(\bar{x}+y)$
⇒ POS
$\boxed{f = \pi M(0,2)}$

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

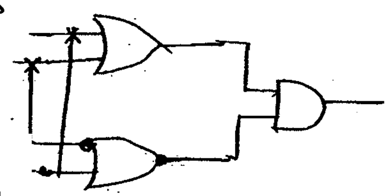
i.e. $\overline{(\text{minterm})} = (\text{maxterm})$

$\overline{m_0} = M_0$ $\overline{M_0} = m_0$
$\overline{m_1} = M_1$ & $\overline{M_1} = m_1$
$\overline{m_2} = M_2$ $\overline{M_2} = m_2$
$\overline{m_3} = M_3$ $\overline{M_3} = m_3$.

SOP → AND – OR



POS → OR – AND



$f = \Sigma m(1,3) = \pi M(\underline{0,2})$
missing no.

Q

x	y	f
0	0	0
0	1	1
1	0	1
1	1	0

SOP = ?
POS = ?

SOP(f) = $\bar{x}y + x\bar{y}$
POS(f) = $(\bar{x}+\bar{y})\cdot(x+y)$ } This is EX-OR gate
$= \bar{x}(x+y) + \bar{y}(x+y)$
$= \underset{\to 0}{x\bar{x}} + \bar{x}y + \bar{y}x + \underset{\to 0}{\bar{y}y}$
$= \bar{x}y + x\bar{y} = x \oplus y$

- In canonical SOP form, each product term must contains all variable of given expression.

Note: In canonical SOP, each product term is known as minterm.

- In canonical POS, each sum term is known as maxterm.

x	y	f
0	0	0
0	1	0
1	0	0
1	1	1

SOP = ?
POS = ?

$SOP(f) = x \cdot y$

$POS(f) = (x+y)(x+\bar{y})(\bar{x}+y)$

$= x+y\,[x(\bar{x}+y)+\bar{y}(\bar{x}+y)]$

$= (x+y)(\cancel{x\bar{x}}^{0}+xy+\bar{y}\bar{x}+\cancel{y\bar{y}}^{0}]$

$= (x+y)(xy+\bar{y}\bar{x})$

$= x(xy+\bar{y}\bar{x})+y(xy+\bar{y}\bar{x})$

$= xy+\bar{y}\cancel{\bar{x}x}^{0}+yx+\bar{y}\cancel{\bar{x}y}^{0}$

$= x \cdot y$

<u>Note</u>: In the min-term zero corresponding variable is complimented and one corresponding variable is taken without complimentation.

In the case of max-term, 0 corresponding variable is taken without complimentation & 1 corresponding variable is taken with complimentation.

$f = \bar{x}\cdot y + x\cdot y$ → canonical SOP

$$\begin{array}{cc} \bar{x}\cdot y & x\cdot y \\ \Downarrow & \Downarrow \\ 01 & 11 \\ 1 & 3 \end{array}$$

$f = \Sigma m(1,3)$ → standard form

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

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

$= y(1)$

$= y$ → minimal form

	x	y	f
0	0	0	1
1	0	1	0
2	1	0	1
3	1	1	0

find SOP & POS:
→ canonical form
→ standard form
→ minimal form

<u>SOP</u>: $f = \Sigma m(0,2)$ → standard form

$f = m_0 + m_2$

$= 00 \quad 10$

$f = \bar{x}\bar{y} + x\bar{y}$ → canonical form

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

$= \bar{y}(1)$

$f = \bar{y}$ → minimal form.

POS: $f = \pi M(1,3) \rightarrow$ ~~cano~~ standard form.

$= M_1 + M_3$

$\quad\;\; 01 \quad 11$

$f = x\bar{y} + \bar{x}\bar{y} \rightarrow$ canonical form.

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

$f = \bar{y} \rightarrow$ minimal form.

① ~~If~~ more zero's is given in output function then we go for min term with (1)

If more 1's are given in output function then we go for max term with (0).

$$f = \Sigma m(1,5) = \pi M(0,2,3,4,6,7)$$

$101 \rightarrow n=3 \rightarrow 2^3 = 8$ combinations

$$= \pi M(1,9) = \Sigma m(0,1,2,3,4,5,6,7,8,10,11,12,13,14,15)$$

$9 \rightarrow 1001 \rightarrow n=4 \rightarrow 2^4 = 16$ combinations.

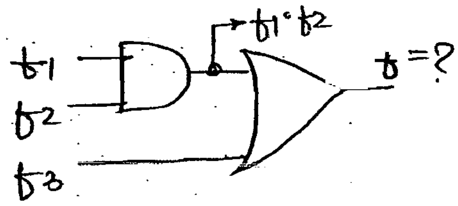
Ex: $f1 = \Sigma m(0,1,2,4)$

$f2 = \Sigma m(1,3,4,5)$

$f3 = \Sigma m(2,3,4,5,6,7)$

AND $= f1 \cdot f2 = f1 \cap f2$

OR $= A + B = A \cup B$.



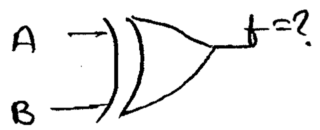
$f1 \cdot f2 = \Sigma m(1,4)$

$f3 = \Sigma m(2,3,4,5,6,7)$

$f \Rightarrow f1 \cdot f2 + f3 = \Sigma m(1,2,3,4,5,6,7)$.

Ex: $A = \Sigma m(0,1,4,5)$

$B = \Sigma m(2,3,4,5,6,7)$



$f = A \odot B$

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

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

$\neq$

$\bar{A}B \rightarrow$ inhibition $\rightarrow$ B but not A.

$\bar{A}B = \Sigma m(2,3,6,7)$

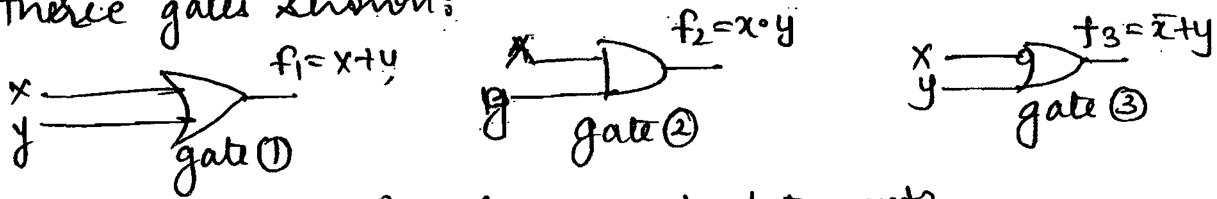
$A\bar{B} \rightarrow$ inhibition $\rightarrow$ A but not B

$A\bar{B} = \Sigma m(0,1)$

$\bar{A}B + A\bar{B} = \Sigma m(0,1,2,3,6,7)$

▷ workbook practice: CH2 : Q 21

Q21 A universal logic gate can ~~be~~ implemented any boolean function by connecting sufficient no of them appropriately.
These gates shown:

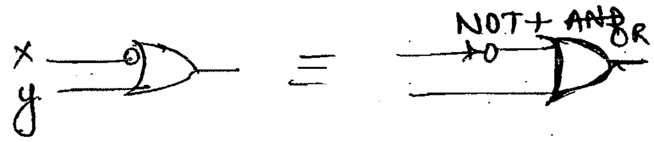


which of the following is correct statement?
(a) Gate 1 is a universal gate.
(b) Gate 2 is a universal gate.
(c) ✓ gate 3 is a universal gate.
(d) ~~gate~~ none of the gates are universal gate.

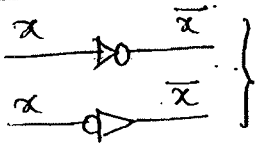
Ans: gate ①: $f_1 = x+y$ → it is OR gate
↳ not universal.

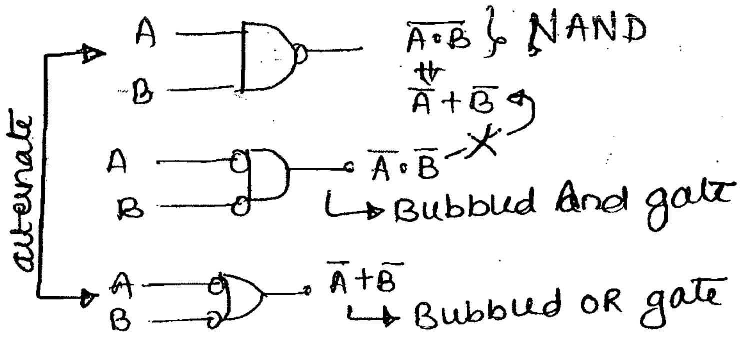
gate ②:- $f_2 = x \cdot y$ → it is AND gate
↳ not universal

gate ③:- $f_3 = \bar{x}+y$ → implication
↳ universal gate.



⑥ Alternative logic gates:



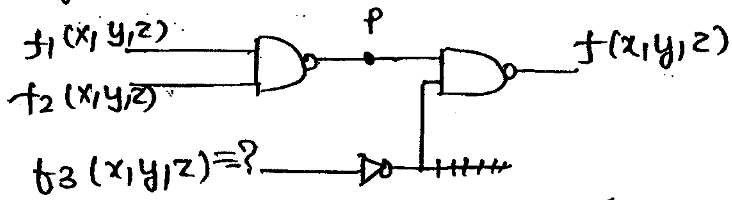


NAND $\xrightarrow{\text{alternate}}$ Bubbled OR gate
NOR $\xrightarrow{\text{alternate}}$ Bubbled And Gate
AND $\xrightarrow{\text{alternate}}$ Bubbled NOR gate
OR $\xrightarrow{\text{alternate}}$ Bubbled NAND gate

shortcut.

⇨ workbook practice : CH2 : Q3, Q1, Q12

Q3 consider the ~~following logic~~ circuit whose inputs are functions f_1, f_2 and f_3. and output is f.
given that:



$f_1(x,y,z) = \Sigma(0,1,3,5)$
$f_2(x,y,z) = \Sigma(6,5)$
$f(x,y,z) = \Sigma(1,4,5)$.

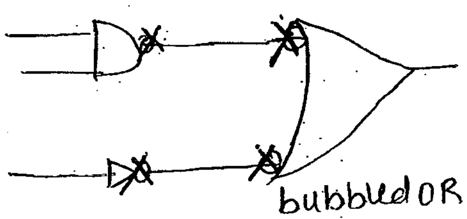
f_3 is

0, 1, 2, 3, 4, 5, 6, 7, 8

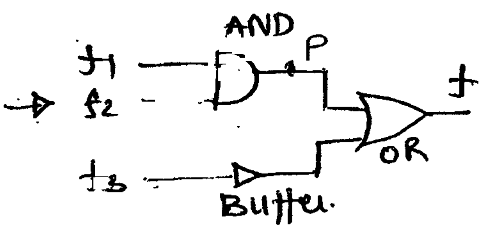
✓(a) $\Sigma(1,4,5)$ (b) $\Sigma(6,7)$
(c) $\Sigma(0,1,3,5)$ (d) None.

Sol^n

$p = \overline{f_1 \cdot f_2} = \bar{f_1} + \bar{f_2}$
$= \Sigma(2,4,6,7,8) + \Sigma(1,2,3,4,7,8)$ X
$= \Sigma(1,2,3,4,6,7,8)$



→



$p = f_1 \cdot f_2 = f_1 \cap f_2$

$p = \Sigma m(5)$

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

$f = P \cup f_3$

$\Sigma m(1,4,5) = \Sigma m(5) \cup \Sigma m(\text{——})$.

$\therefore f_3 = \Sigma m(1,4,5)$ or $\Sigma m(1,4)$.

Q1 consider the following SOP expression F;

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

the equivalent ~~sum of product~~ product of sum expression is:

✓(a) $F = (A+B+C)(\bar{A}+B+C)(\bar{A}+\bar{B}+C)$
(b) $F = (A+\bar{B}+\bar{C})(A+B+C)(\bar{A}+\bar{B}+\bar{C})$
(c) $F = (\bar{A}+B+\bar{C})(A+\bar{B}+\bar{C})(A+B+C)$
(d) $F = (\bar{A}+\bar{B}+C)(A+B+\bar{C})(A+B+C)$