classmate
Date
Page

# DIGITALS

## Boolean Expression:-

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

It mainly has literals (variable such as x, y, z), a binary operator, '=' (and, or, +)

Truth table:-

Tabular representation of behavior of a function for different set of values

No. of rows = $2^n+1$ ; No. of columns = $n+1$ (n - no. of literals)

- Types of expression:

* Normal Expression
* Canonical Expression

→ Any expression can have multiple equivalent functions
→ 1's represent min-term (product) (All literals )
→ 0's represent max-term (sum)

* $f(x,y,z) = z + \bar{y}z$ . Expressed as product terms
  ↳ SOP (Sum of product)
* $f(x,y,z) = (x+\bar{y})\cdot(x+z)$ Expressed as sum terms
  ↳ POS (product of sum)

* Min-terms is a product term which represents one's of functional value either in normal or complimentary form.

Scanned by CamScanner

* Max-terms: Sum term represents 0 which has all literals but with its complementary form

* $f_1(x,y,z) = x+\bar{y}z$ (SOP) (Disjunctive)
$f_2(x,y,z) = (x+\bar{y})\cdot(x+z)$ (POS) (Conjunctive)
$f_3(x,y,z) = \bar{x}\bar{y}z + x\bar{y}\bar{z} + x\bar{y}z + xy\bar{z} + xyz$ (SOP)
$f_4(x,y,z) = (x+y+z)\ (x+\bar{y}+z)(x+\bar{y}+\bar{z})$ (POS)

| x | y | z | f |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

Simplest form of expression in terms of cost is minimal expression
Ex: $f_1(x,y,z)$

$f_4(x,y,z)$ is max-term cannonical form (M)
$f_3(x,y,z)$ is min-term cannonical form (m)
-OR-

$f_3(x,y,z)$ is Disjunctive cannonical form
$f_4(x,y,z)$ is Conjunctive cannonical form.

Max-term cannonical POS form of expression where every sum term is max term

Min term Cannonical SOP form of expression where every product term is min term

Equivalent expression-OR- equivalent function:- When the expressions give the same output for all set of values.

$f_3(x,y,z) = \bar{x}\bar{y}z + x\bar{y}\bar{z} + x\bar{y}z + xy\bar{z} + xyz$
$= m_1 + m_4 + m_5 + m_6 + m_7$
$= \Sigma m\ (1,4,5,6,7)$

$f_4(x,y,z) = (x+y+z)\ (x+\bar{y}+z)(x+\bar{y}+\bar{z})$
$= M_0 \cdot M_2 \cdot M_3$
$= \Pi M\ (0,2,3)$

→ Min or max term cannonical expressions are not minimal expression. They are not efficient in terms of speed or number of gates used.

* All these are complete boolean function and combinational circuit (f is defined for all set of inputs - complete boolean functions)
* If for some inputs the output is not defined or definite then incomplete boolean function (It's o/p is represented as x).

K-maps (Karnaugh Maps):- Graphical representation of truth table

NOTE 1 * If there are n literals then there are $2^n$ cells

Ex:- $\bar{x}(\bar{y}+z)+\bar{z}$

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

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

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

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

Truth table:-

| x | y | z | f |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

| x \ yz | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |

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

$f = \Sigma m(0,1,2,3,4,6)$

$f = \Pi M(5,7)$

NOTE 2* K-maps is represented in the form of $m \times n$, where m and n are in the power of 2.

NOTE 3* Every cell differs by its neighbour by one bit.

Ex:- $f = \Sigma m(0,1,3,6,7,9)$

K-maps:-

| ab \ cd | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1* | 1 | 1 | 0 |
| 01 | 0 | 0 | 1 | 1* |
| 11 | 0 | 0 | 0 | 0 |
| 10 | 0 | 1* | 0 | 0 |

Essential product term

$f = \bar{a}\bar{b}\bar{c} + \bar{a}cd + \bar{a}bc + a\bar{b}\bar{c}d$

- OR -

$f = \bar{a}\bar{b}\bar{c} + \bar{b}\bar{c}d + \bar{a}cd + \bar{a}bc$

$\bar{a}(\bar{b}\bar{c}+bc) + \bar{b}\bar{c}d + \bar{a}cd$

Ex: $f = \Sigma m(0,2,5,7,8,10,14,15)$

| ab \ cd | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 0 | 0 | 1 |
| 01 | 0 | 1 | 1 | 0 |
| 11 | 0 | 0 | 1 | 1 |
| 10 | 1 | 0 | 0 | 1 |

$f = \bar{b}\bar{d} + \bar{a}bd + abc$

Ex: $\Sigma m(0,3,4,6,7)$

Minimal Sum:

| $x \backslash yz$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 |

$f = \bar{y}\bar{z} + yz + x\bar{z}$

$f = y \odot z + x\bar{z}$

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

| $a \backslash bc$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 |

$f = \bar{b}\bar{c} + bc + ab$

$f = (b+\bar{c})\cdot(a+\bar{b}+c)$

↳ minimal product

Ex: $f(w,x,y,z) = \Pi M(1,3,5,8,10,11,14)$

| $wx \backslash yz$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 0 | 0 | 1 |
| 01 | 1 | 0 | 1 | 1 |
| 11 | 1 | 1 | 1 | 0 |
| 10 | 0 | 1 | 0 | 0 |

Minimal sum:-

Minimal product:

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

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

*. Implicant is a product terms that implies the function

Implies

$f_1 \Rightarrow f_2$

$1 \Rightarrow 1$

$0 \Rightarrow 0$ } both should satisfy.

(mo) (p1)

* Subsume: $\bar{a}\bar{b}\bar{c}$, $\bar{b}\bar{c}$ ; $\bar{b}\bar{c}$ subsumes $\bar{a}\bar{b}\bar{c}$

∴ p1 subsumes mo.

A term p1 is said to subsumes p2 if all literals in p1 are in p2

* In any expression we have subsume of one term then it is not minimal

$f = \bar{b}\bar{c} + bc + ab + \bar{a}\bar{b}\bar{c}$

This f is not minimal

A expression is said to be minimal if there are no subsumes of any product term

* Implicates are sum terms which are implied by the function

All possible product terms are implicant

All possible sum terms are implicates

* An implicant is said to be prime implicant if it does not subsume any other term in same exp

* Minimal expression: SOP form which comprimises prime implicant

* An implicate is said to be prime implicate if it doesn't have subsumes. in same exp

Implicant is product term and not expression.

minterms are implicants

* $f(w,x,y,z) = \Sigma m(0,1,3,7,8,12)$

| wx \ yz | 00 | 01 | 11 | 10 * |
|---|---|---|---|---|
| 00 | 1 | 1 | 1 | 0 |
| 01 | 0 | 0 | 1* | 0 |
| 11 | *1 | 0 | 0 | 0 |
| 10 | 1 | *0 | 0 | 0 |

All product terms from K-map

$f(w,x,y,z) = \bar{x}\bar{y}\bar{z} + w\bar{y}\bar{z} + \bar{w}\bar{x}z + \bar{w}yz$
$+ \bar{w}\bar{x}\bar{y}$ // All prime implicants

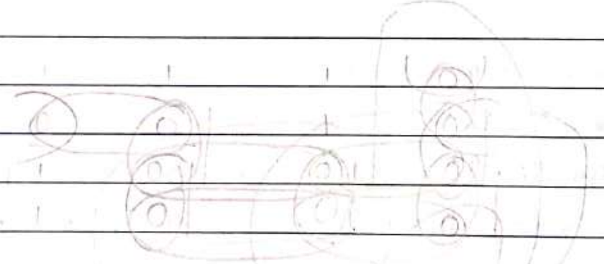
If $T_1$ subsumes $T_2$, $T_1$ is not prime.

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

Implicants: $m_0, m_1, m_3, m_7, m_8, m_{12}$.

Implicates: $M_2, M_4, M_5, M_6, M_9, M_{10}, M_{11}, M_{13}, M_{14}, M_{15}$.

13 - di
2 - quad



Prime implicate: $(w+\bar{x}+y)\,(w+\bar{x}+z)\,(\bar{x}+y+\bar{z})$
$(\bar{y}+z)\,(\bar{w}+\bar{y})\,(\bar{w}+\bar{z})$

Terms essential for all minimal expression: essential terms

Prime implicants:-
$w\bar{y}\bar{z}$, $\bar{x}\bar{y}\bar{z}$, $\bar{w}\bar{x}\bar{y}$, $\bar{w}\bar{x}z$, $\bar{w}yz$

Essential p.i:- A prime implicates associated with essential one cells

Essential prime implicates: A prime implicates associated with essential 0's cell

Essential prime implicants:- $(w\bar{x}\bar{z})$, $(\bar{w}yz)$
Essential prime implicates:- $(\bar{y}+z)$, $(\bar{w}+\bar{z})$

How many minimal sums are there:
$f = \Sigma p_1 + \Sigma p_2 + \bar{w}\bar{x}\bar{y}$

The boolean functions for which some input values the o/p is not defined (it takes 0 or 1) then it is incomplete boolean function.
The term is represented as - (don't care)
$f(w,x,y) = \Sigma m(0,1,3,4) + d_c(7)$
$= \Pi m(2,5,6) + d_c(7)$

* $f(x,y,z) = \Pi M(2,3,7)$

| $x \backslash yz$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1* | 1* | 0 | 0* |
| 1 | 1 | 1* | 0* | 1 |

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

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

Implicants: $m_0, m_1, m_4, m_5, m_6$
$\bar{x}\bar{y}\bar{z}, \bar{x}\bar{y}z, x\bar{y}z, x\bar{y}z, xy\bar{z}$

Implicates: $M_2, M_3, M_7$
$(x+\bar{y}+z), (x+\bar{y}+\bar{z}), (\bar{x}+\bar{y}+\bar{z})$.

Prime implicants: $\bar{y}$, $x\bar{z}$
Prime implicates $(\bar{y}+\bar{z})\ (x+\bar{y})$

Ex: * $f(x,y,z,w) = \Pi M(0,2,3,7,9) + dc(1,10)$
Incomplete Boolean functions,

| | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | - | 0 | 0 |
| 01 | 1 | 1* | 0* | 1* |
| 11 | 1 | 1 | 1 | 1 |
| 10 | *1 | 0 | 1* | - |

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

Implicants
$m_4, m_5, m_6, m_8, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}$

$(\bar{w}x\bar{y}\bar{z}), (\bar{w}x\bar{y}z),$
$(\bar{w}xy\bar{z}), (w\bar{x}\bar{y}\bar{z}),$
$(w\bar{x}yz), (wx\bar{y}\bar{z}),$
$(wx\bar{y}z), (wxy\bar{z}),$
$(wxyz)$.

Implicates.
$M_0, M_2, M_3, M_7, M_9$

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

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

Essential term: $x\bar{y}$, $x\bar{z}$, $wy$, $w\bar{z}$

Essential:
$(w+x)$

Ex: $f(w,x,y,z) = \Pi M(0,1,3,7,8,12) + dc(5,6,11,15)$

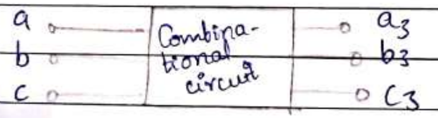
| wx \ yz | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 1* |
| 01 | 1 | - | 0 | - |
| 11 | 0 | 1* | - | 1 |
| 10 | 0 | 1* | - | 1 |

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

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

3 bit binary to excess 3
will convert 3 bit i/p to +3 excess output

| a | b | c | Output $a_3$ | $b_3$ | $c_3$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | - | | |
| 1 | 1 | 0 | - | | |
| 1 | 1 | 1 | - | | |



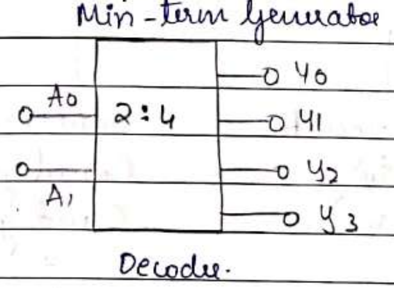

No. of gates
1-10 SSI
10-100 MSI
100-1000 LSI
>1000 VLSI

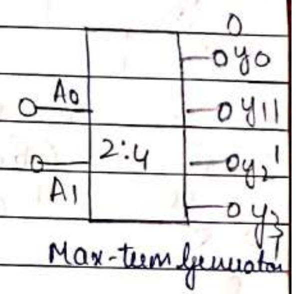
* MSI Components:-
Medium Scale Integration (10-100)

Decoder:- A combinational logic circuit that can transform from one form to another.

$n : 2^n$

n inputs $2^n$ outputs.




| A | B (A0) | y0 | y1 | y2 | y3 | En |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| x | x | 0 | 0 | 0 | 0 | 0 |

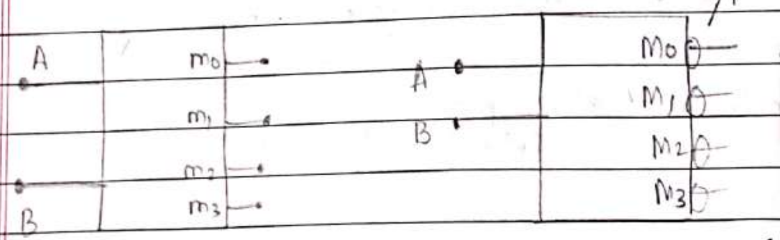
$y_0 = \bar{A_1}\bar{A_0}$ $\qquad y_0 = A_1 + A_0$

$y_1 = \bar{A_1} A_0$ $\qquad y_1 = A_1 + \bar{A_0}$

$y_2 = A \bar{A_0}$ $\qquad y_2 = \bar{A} + A_0$

$y_3 = A A_0$ $\qquad y_3 = \bar{A} + \bar{A_0}$

**Logic high / Active high IC's** :- When enable is high or logic 1
**Active low** : When enable is low or zero




nor = min
nand = max
and - max
or - min

min-term generator (nor gates) (Or gates)

max-term generator (nand gates) (and gates)

$f = \overline{\overline{m_0 + m_1}}$

$f = \overline{\overline{m_0}\,\overline{m_1}}$

$f = \overline{M_0 \cdot M_1}$

$\overline{f} = \overline{M_2 . M_3}$

$f = \overline{\overline{M_2} + \overline{M_3}}$

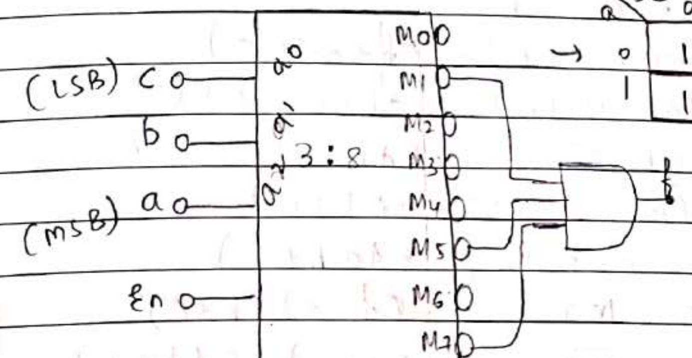
$f = \overline{m_2 + m_3}$)

NAND, AND } maxterm

NOR, OR } minterm

* $f(a,b,c) = \bar{a}b + a\bar{c} + b\bar{c}$ (min-term)
Implement using suitable $n:2^n$ decoder
(i) AND gates only (without using truth table

$m_0$ - $\bar{a}\bar{b}$
$m_1$ - $\bar{a}b$
$m_2$ - $a\bar{b}$
$m_3$ - $ab$.




| a \ bc | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |

$M_1$ $M_5$ $M_7$

(i) AND gates

$\overline{\overline{f}} = \overline{\bar{a}b + a\bar{c} + b\bar{c}}$

$\overline{a+b} = \bar{a}\cdot\bar{b}$

$\overline{f} = \overline{\bar{a}b}\;\;\overline{a\bar{c}}\;\;\overline{b\bar{c}}$   M

010   101   011

$f = \overline{(a+\bar{b})\;(\bar{a}+c)\;(b+c)} = (a+\bar{b}+c\bar{c}).(\bar{a}+b\bar{b}+c)(a\bar{a}+$

$f = M_2 . M_3\; M_4\; M_6\; M_0$

Use demorgans for NAND and NOR they have complement.

$f = \bar{a}b + a\bar{c} + b\bar{c}$
$= \bar{a}b\cdot 1 + a.1.\bar{c} + b\cdot\bar{c}\cdot 1$
$= \bar{a}b\,(c+\bar{c}) + a\,(b+\bar{b})\,\bar{c} + (a+\bar{a})\,b\bar{c}$
$= \bar{a}bc + \bar{a}b\bar{c} + ab\bar{c} + a\bar{b}\bar{c} + ab\bar{c} + \bar{a}b\bar{c}$

A+A=A

Group common

$= \bar{a}bc + \bar{a}b\bar{c} + ab\bar{c} + a\bar{b}\bar{c} + \bar{a}\bar{b}\bar{c} = \Pi M(\cdot, 5, 7)$
$= m_3 + m_2 + m_6 + m_4 + m_0 = \Sigma m(3, 2, 6, 4, 0)$

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

$f = \bar{a}b + a\bar{c} + b\bar{c}$

$f = (\bar{a}b + a\bar{c} + b^{-})\cdot(\bar{a}b + a\bar{c} + \bar{c})$

$= (\bar{a}b + a\bar{c} + b^{-})\cdot(\bar{a}b + \bar{c}(a+1))$

$= (\bar{a}b + a\bar{c} + b^{-})\cdot(\bar{a}b + \bar{c})$

$= (\overline{a\bar{c}} + \bar{a} + \bar{b})\cdot(\bar{a}b + \bar{c})$

$= (\overline{\bar{a} + \bar{c} + b^{-}})\cdot(\bar{a}b + \bar{c})$

$= M_7 \; (111) \; (\bar{a}+\bar{c})(b+\bar{c})$

$= \overline{\overline{M_7}} \; (\bar{a} + 0 + \bar{c})(0 + b + \bar{c})$

or 0 zero and a1

a missing term.

$= M_7 \; (\bar{a} + b\cdot\bar{b} + \bar{c})(a\cdot\bar{a} + b + \bar{c})$

$= M_7 \; (M_5 \cdot M_7) \; (M_1 \cdot M_5)$

$= M_1 \cdot M_5 \cdot M_7$

a-3 bit

+ The odd parity generator using decoder and nor- gates only

{ OR / nor

| a \ bc | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |

$f = (\bar{a}+b+\bar{c})\cdot(\bar{a}+\bar{b}+c)\cdot(\bar{a}+\bar{b}+\bar{c})\cdot(\bar{a}+\bar{b}+\bar{c})$

$f = M_1 \cdot M_2 \cdot M_4 \cdot M_7$

$\bar{f} = \overline{M_1 \cdot M_2 \cdot M_4 \cdot M_7} = \overline{\overline{M_1} + \overline{M_2} + \overline{M_4} + \overline{M_7}}$

$= \overline{m_1 + m_2 + m_4 + m_7}$

= 4 bit even parity generator

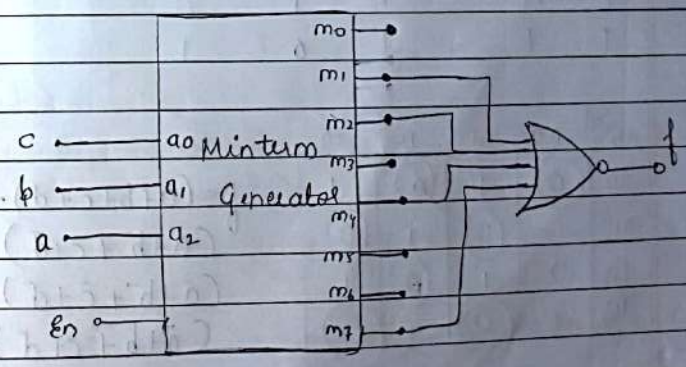
| a | b | c | o/p |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

If OR gate → write in SOP form

If AND gate → write in POS form

If NOR gate → take POS apply de-morgan's law

If NAND gate → take SOP apply de-morgan's law



classmate Date ____ Page ____

3) 4-bit even parity using NOR gate

| a | b | c | d | f |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |

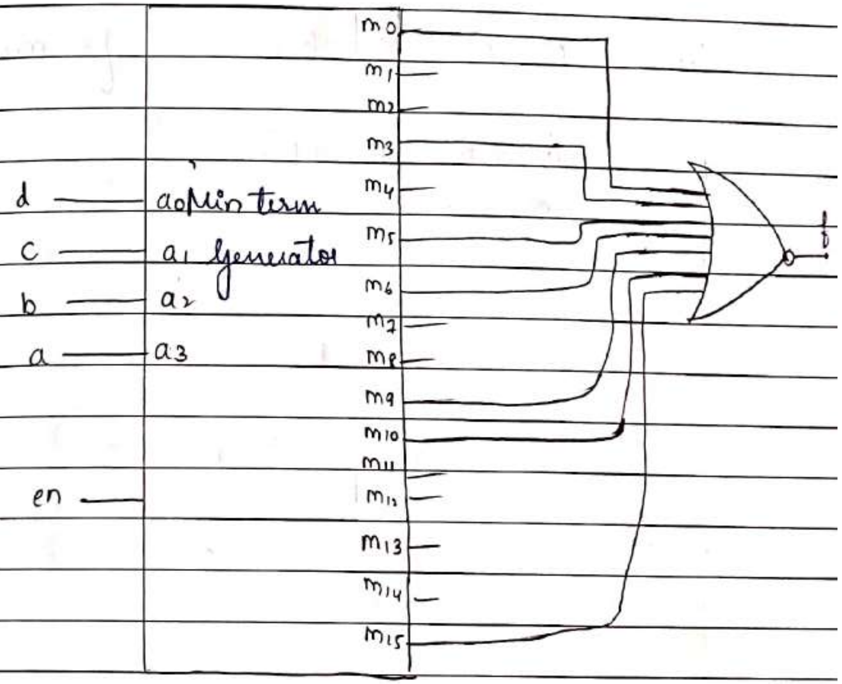
| ab \ cd | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 1 | 0 | 1 |
| 01 | 1 | 0 | 1 | 0 |
| 11 | 0 | 1 | 0 | 1 |
| 10 | 1 | 0 | 1 | 0 |

$f = (a+b+c+d)\cdot(a+b+\bar{c}+\bar{d})$
$(a+\bar{b}+c+\bar{d})\ (a+\bar{b}+\bar{c}+d)$
$(\bar{a}+\bar{b}+\bar{c}+\bar{d})$
$(\bar{a}+b+c+\bar{d}).(\bar{a}+b+\bar{c}+d)$

classmate Date ____ Page ____

$f = M_0\ M_3\ M_5\ M_6\ M_9\ M_{10}\ M_{15}$

$f = m_0 + m_3 + m_5 + m_6 + m_9 + m_{10} + m_{15}$




* $f = \bar{x}y + \bar{y}z$

Using nor and decoder

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

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

$= 011 \quad 010 \quad 101 \quad 001$

$= m_3 + m_2 + m_5 + m_1 \qquad = \Pi M(0, 1, 4, 6, 7)$

$$f = \overline{\overline{m_3 + m_2 + m_5 + m_1}}$$

$$f = \overline{\overline{m_3}\ \overline{m_2}\ \overline{m_5}\ \overline{m_1}}$$

$$f = \overline{M_3\ M_2\ M_5\ M_1}$$

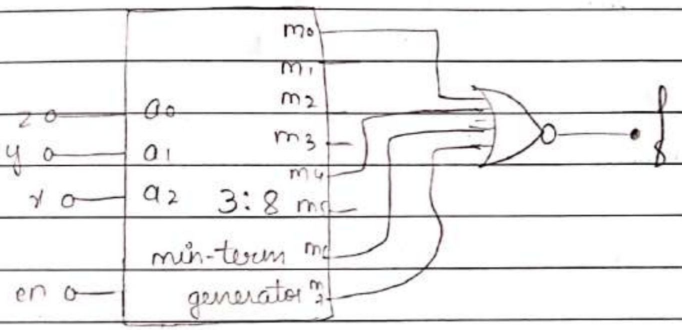
$$f = \overline{M_3} + \overline{M_2} + \overline{M_5} + \overline{M_1}$$

$$f = m_3 + m_2$$

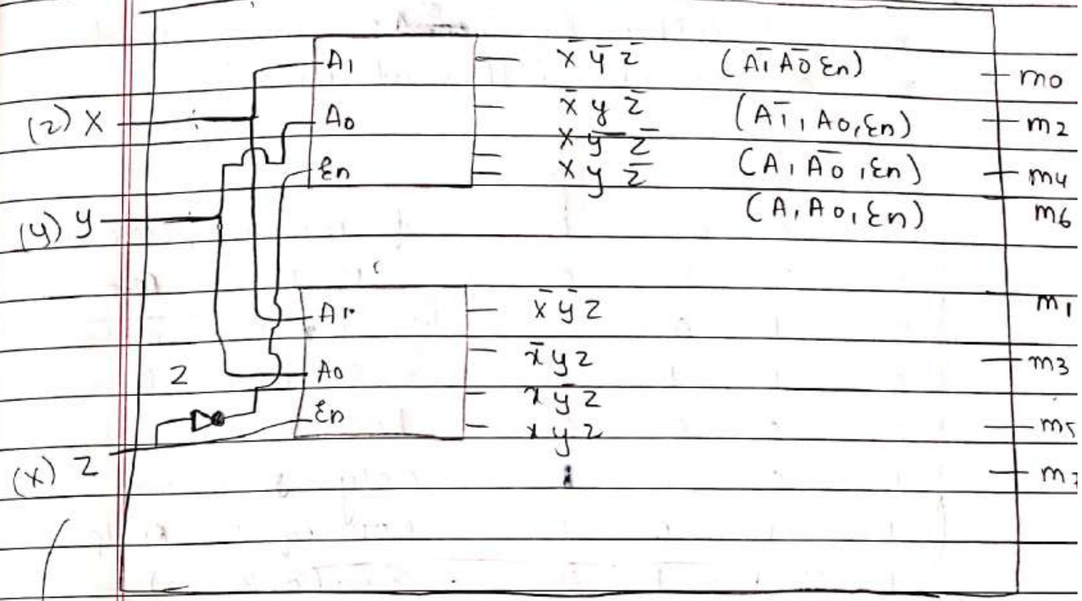
So $f = \overline{\overline{M_0 M_4 M_6 M_7}}$

$$f = \overline{\overline{M_0} + \overline{M_4} + \overline{M_6} + \overline{M_7}}$$

$$f = m_0 + m_4 + m_6 + m_7$$



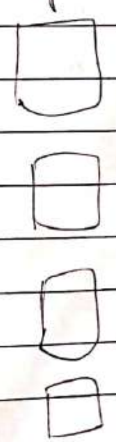

* Write the output




Change to x sequence

En - most significant

* Implement 4:16 using 2:4



Assume: 0 is not divisible then OR

* Design a 3-bit input system which checks if the no is divisible by 3 or 5. Using decoder and OR/NOR gate with least i/p's.

| x | y | z | f |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

$f = m_0 + m_3 + m_6 + m_5$

$f = M_1 M_2 M_4 M_7$

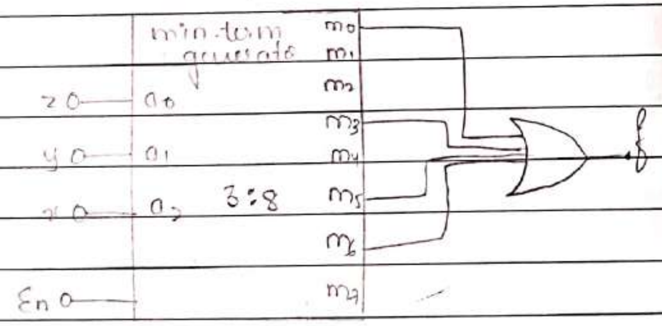
$f = \overline{m_1 + m_2 + m_4 + m_7}$

So -OR- Gate.

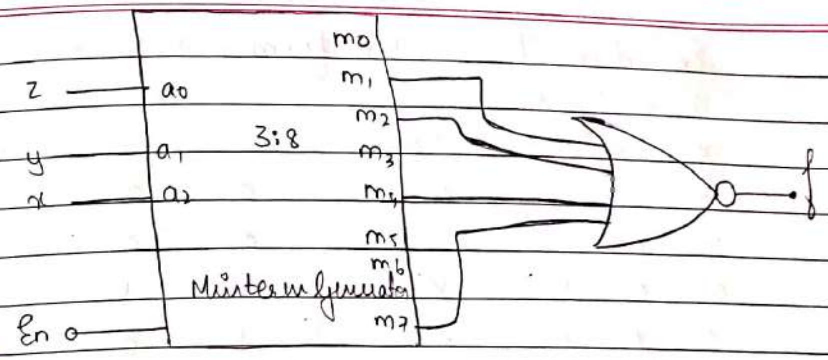
because OR requires only 4 i/ps and NOR requires 4 i/p's.

$= \Sigma m(0,3,5,6)$

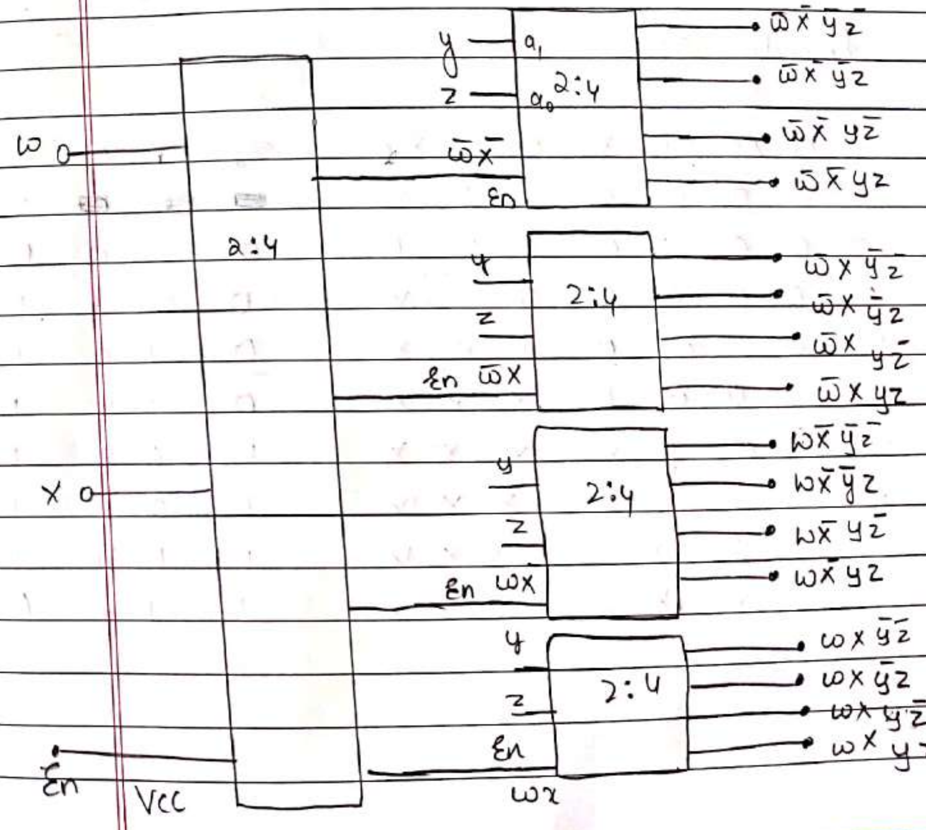
$= \Pi M(1,2,4,7)$




-OR-




* Implement 4:16 using 2:4 decoder.



Encoder: It is $2^n:n$ form

Priority Encoder:

| $X_3$ | $X_2$ | $X_1$ | $X_0$ | $Z_1$ | $Z_0$ | Valid |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | X | 0 | 1 | 1 |
| 0 | 1 | X | X | 1 | 0 | 1 |
| 1 | X | X | X | 1 | 1 | 1 |

Encoder: 8:3

| $X_7$ | $X_6$ | $X_5$ | $X_4$ | $X_3$ | $X_2$ | $X_1$ | $X_0$ | $Z_2$ | $Z_1$ | $Z_0$ | Valid |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | [redacted] | [redacted] | [redacted] | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | X | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | X | X | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | X | X | X | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | X | X | X | X | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | X | X | X | X | X | 1 | 0 | 1 | 1 |
| 0 | 1 | X | X | X | X | X | X | 1 | 1 | 0 | 1 |
| 1 | X | X | X | X | X | X | X | 1 | 1 | 1 | 1 |

* Design a comparator that compares two numbers in the range 0-3. Using max-term generator decoder and gate with least inputs.

| A | | B | | A>B | A<B | A=B |
|---|---|---|---|---|---|---|
| $A_1$ | $A_0$ | $B_1$ | $B_0$ | $f_1$ | $f_2$ | $f_3$ |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 |

$f_1 = M_0 M_1 M_2 M_3 M_5 M_6 M_7 M_{10} M_{11} M_{15}$ (10)

$f_2 = M_0 M_4 M_5 M_8 M_9 M_{10} M_{12} M_{13} M_{14} M_{15}$ (10)

$f_3 = M_1 M_2 M_3 M_4 M_6 M_7 M_8 M_9 M_{11} M_{12} M_{13} M_{14}$ (12)

$f_1 = \Sigma m (m_4, m_8, m_9, m_{12}, m_{13}, m_{14})$

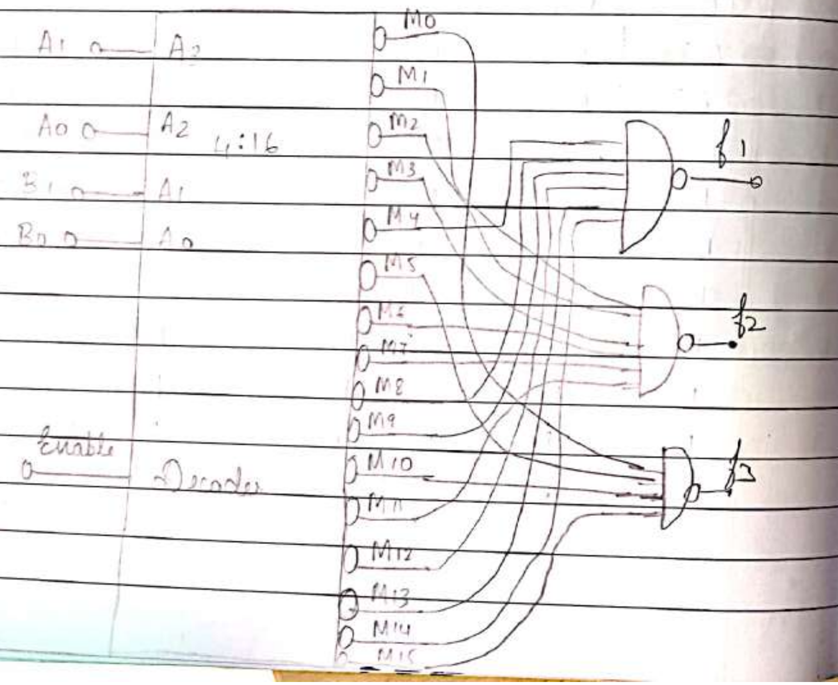
$= \Sigma m (4, 8, 9, 12, 13, 14)$ (6)

$f_1 = \overline{M_4 + M_8 + M_9 + M_{12} + M_{13} + M_{14}}$

$f_2 = \Sigma m (1, 2, 3, 6, 7, 11)$ (6)

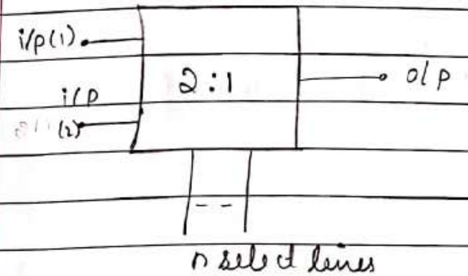
$f_2 = \overline{M_1 + M_2 + M_3 + M_6 + M_7 + M_{11}}$

$f_3 = \Sigma m (0, 5, 10, 15)$ (4) $= \overline{M_0 + M_5 + M_{10} + M_{15}}$




* Multiplexers :- It is MSI component (10-100 gates in IC)

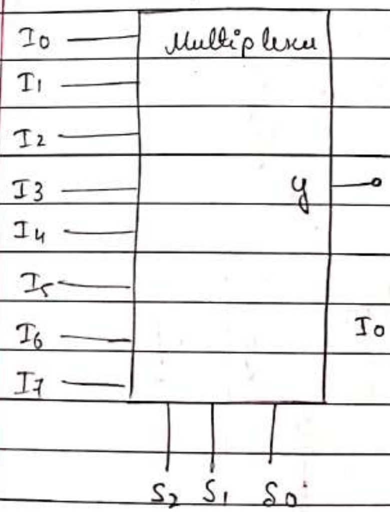
It is $2^n:1$ form





$I_0 \bar{S_1} \bar{S_0} + I_1 \bar{S_1} S_0 + I_2 S_1 \bar{S_0} + I_3 S_1 S_0$

$2^3:1$



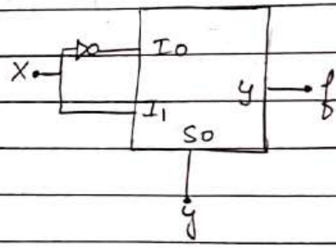

$y = I_0 \bar{S_2} \bar{S_1} \bar{S_0} + I_1 \bar{S_2} \bar{S_1} S_0 + I_2 \bar{S_2} S_1 \bar{S_0} + I_3 \bar{S_2} S_1 S_0 + I_4 S_2 \bar{S_1} \bar{S_0} + I_5 S_2 \bar{S_1} S_0 + I_6 S_2 S_1 \bar{S_0} + I_7 S_2 S_1 S_0$.

$I_0$

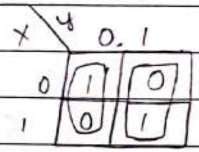
Ex:- $f(x,y) = \Sigma m(0,3)$ using 2:1

y select line:

| X | y | f |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |




$\Rightarrow$

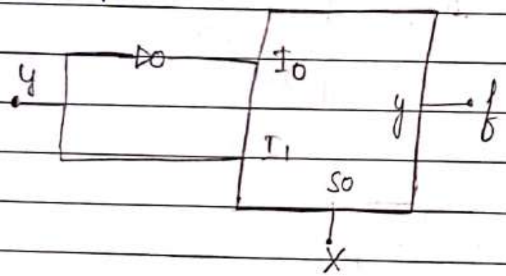


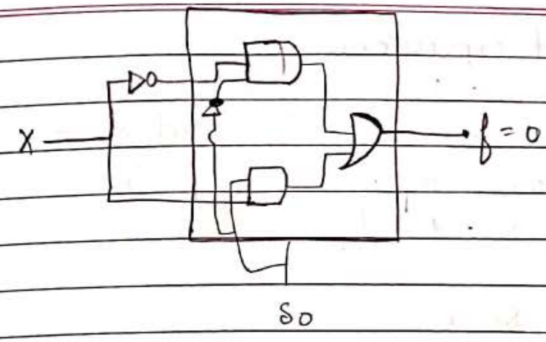

X select line

| X(select) | y | f |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

$X(0) \rightarrow \bar{y}$

$X(1) \rightarrow y$




Implement the following $f(x,y,z) = \bar{x}z + xy + \bar{y}\bar{z}$ Using 4:1 multiplexer & x,y as select lines (without using truth table)

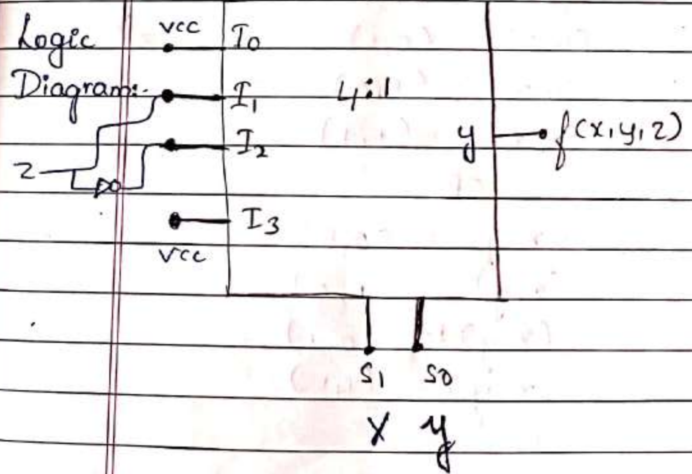
K-map:

| x \ yz | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 |

$\rightarrow I_1$ $I_0$ $I_2$ $I_3$

// We assume others are zero because its complete boolean fn

Logic Diagram:-




// Only z is i/p so z, $\bar{z}$, vcc, gro-

2 Mux input expressions.

$(x,y)$ $(0,0) = 1$
$(x,y)$ $(0,1) = z$
$(x,y)$ $(1,0) = \bar{z}$
$(x,y)$ $(1,1) = 1$

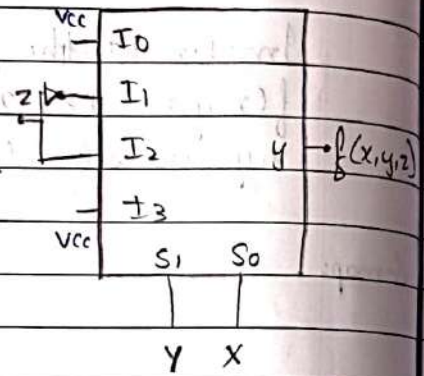
dependent on $z$

$I_i = f(z)$

② Take $S_1 = y$, $S_0 = x$

| $x$ \ $yz$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 |

$I_0$, $I_1$, $I_2$, $I_3$




$I_0 = 1$ $(y,x) = (0,0)$ $S_1 = y$
$I_1 = \bar{z}$ $(y,x) = (0,1)$ $S_0 = x$
$I_2 = z$ $(y,x) = (1,0)$
$I_3 = 1$ $(y,x) = (1,1)$

OR

$(x,y) = (0,0)$
$(x,y) = (1,0)$
$(x,y) = (0,1)$
$(x,y) = (1,1)$

* Implement the following
// Write the statement for select lines
→* Assuming $S_1 = y$
$S_0 = w$

$$f(w,x,y,z) = \Sigma m(0,3,4,7,9,11)$$

K-map:-

| $wx$ \ $yz$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 0 | 1 | 0 |
| 01 | 1 | 0 | 1 | 0 |
| 11 | 0 | 0 | 0 | 0 |
| 10 | 0 | 1 | 1 | 0 |

$I_0$, $I_1$, $I_2$, $I_3$

Input mux expressions:
$(S_1, S_0)$
$(y, w)$ $(0,0)$ $I_0 = \bar{z}$
$(y, w)$ $(0,1)$ $I_1 = \bar{x}.z$
$(y, w)$ $(1,0)$ $I_2 = z$
$(y, w)$ $(1,1)$ $I_3 = \bar{x}z$

We say $f(x,z) = I_i$

| $x$ \ $z$ | 0 | 1 |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 0 |

$I_0 = \bar{z}$

| $x$ \ $z$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 0 | 0 |

$I_1 = \bar{x}.z$

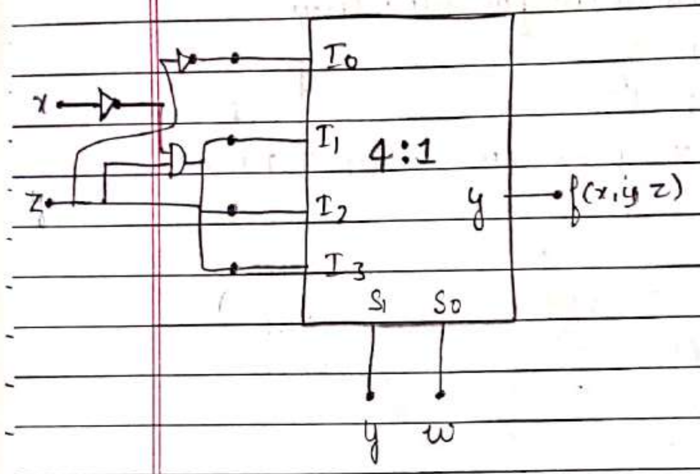
| $x$ \ $z$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 0 | 1 |

$I_2 = z$

| $x$ \ $z$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 0 | 0 |

$I_3 = \bar{x}z$

## Logic Diagram:




* 1) $S_1 = w$; $S_0 = z$.

K-map:

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

$\rightarrow I_1$, $\rightarrow I_0$, $\rightarrow I_2$, $\rightarrow I_3$

Input mux expressions:

| $(S_1, S_0)$ | | |
|---|---|---|
| $(\bar{w},\bar{z})$ | (0,0) | $I_0 = \bar{y}$ |
| $(\bar{w},z)$ | (0,1) | $I_1 = y$ |
| $(w,\bar{z})$ | (1,0) | $I_2 = gnd$ |
| $(w,z)$ | (1,1) | $I_3 = \bar{x}$ |

$I_i = f(x, y)$

K-maps for individual terms:-

$I_0$

| x \ y | 0 | 1 |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 0 |

$I_0$ $\quad I_0 = \bar{y}$

| x \ y | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 0 | 1 |

$I_1 = y$.

$I_2 =$

| x \ y | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 0 |

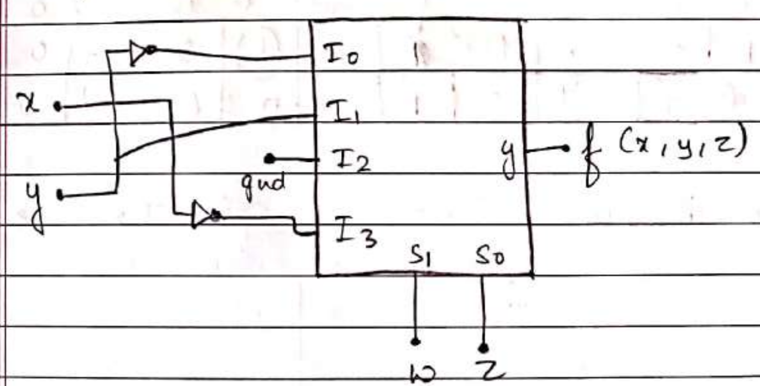
ground.

$I_3$:

| x \ y | 0 | 1 |
|---|---|---|
| 0 | 1 | 1 |
| 1 | 0 | 0 |

$= \bar{x}$

## Logic Diagram:




* $S_1 = x$ $\quad S_0 = y$

Assuming $S_1 = x$ ; $S_0 = y$

K-map:

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

$I_2$ $I_0$ $I_1$ $I_3$

4/p mux: $(S_1, S_0)$ $\qquad I_i = f(w, z)$

| | | |
|---|---|---|
| $(x, y)$ | $(0,0)$ | $I_0 =$ |
| $(x, y)$ | $(0,1)$ | $I_1 =$ |
| $(x, y)$ | $(1,0)$ | $I_2 =$ |
| $(x, y)$ | $(1,1)$ | $I_3 =$ |

$I_0$

| w \ z | 0 | 1 |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 0 | 1 |

$I_1$

| w \ z | 0 | 1 |
|---|---|---|
| 0 | 0 | [illegible] |
| 1 | 0 | 1 |

$I_2$

| w \ z | 0 | 1 |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 0 | 0 |

$I_3$

| w \ z | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 0 | 0 |

$I_0 = \bar{w}\bar{z} + wz \qquad I_1 = z \qquad \bar{w}\bar{z} = I_2 \qquad I_3 = \bar{w}z$

Implement,

$f(w, x, y, z) = \Pi M(0, 3, 6, 9, 11, 14, 15)$ using 8:1 mux with $w, y, z$ as select lines (using K-maps)

K-map:

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

$I_0$ $I_1$ $I_3$ $I_2$

$I_4$ $I_5$ $I_7$ $I_6$

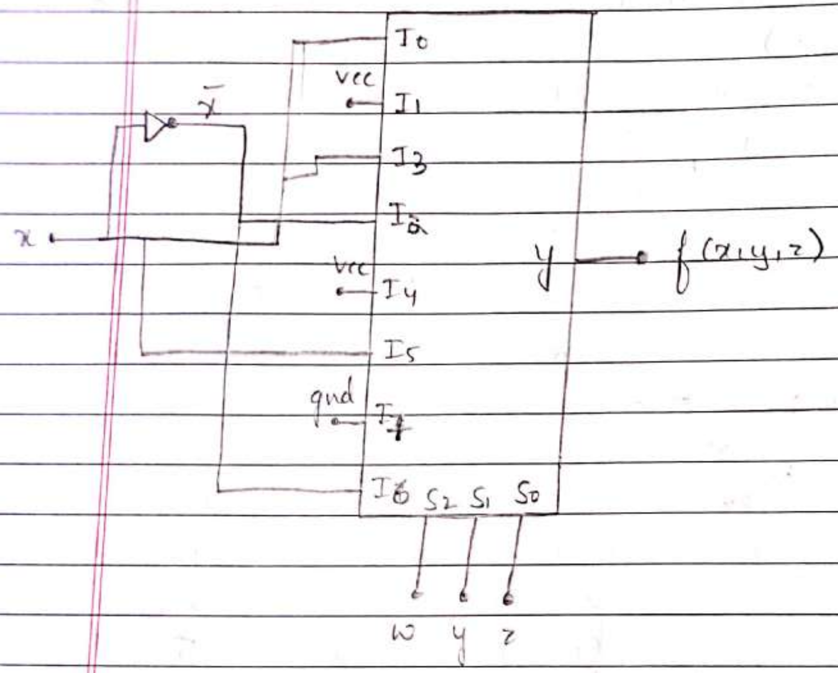
Mux i/p: Assuming $w = S_2 \quad y = S_1 \quad z = S_0$

$(S_2, S_1, S_0)$

| | | |
|---|---|---|
| $(w, y, z)$ | $(0,0,0)$ | $I_0 = x$ |
| $(w, y, z)$ | $(0,0,1)$ | $I_1 = V_{CC}$ |
| $(w, y, z)$ | $(0,1,0)$ | $I_2 = \bar{x}$ |
| $(w, y, z)$ | $(0,1,1)$ | $I_3 = \bar{x}$ |
| $(w, y, z)$ | $(1,0,0)$ | $I_4 = V_{CC}$ |
| $(w, y, z)$ | $(1,0,1)$ | $I_5 = x$ |
| $(w, y, z)$ | $(1,1,0)$ | $I_6 = gnd$ |
| $(w, y, z)$ | $(1,1,1)$ | $I_7 = \bar{x}$ |

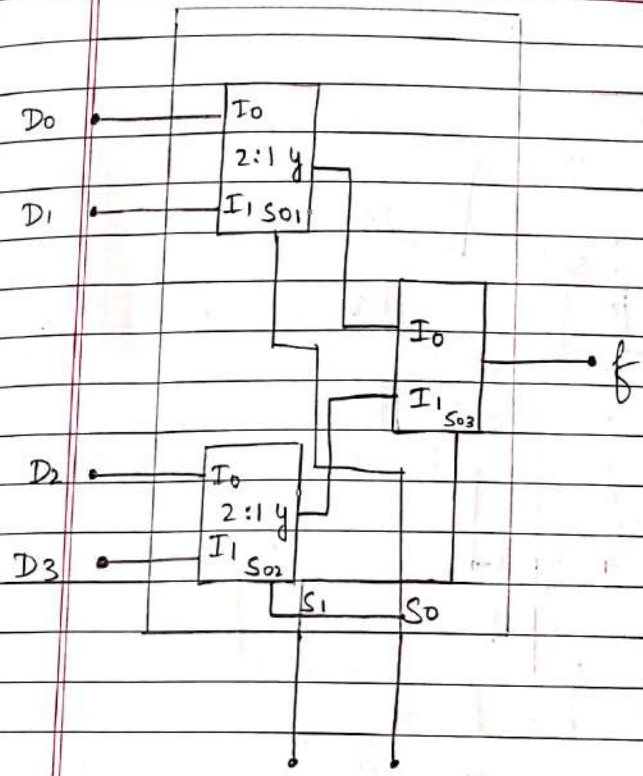
→ Here $I_i = f(x)$

Logic Diagram:-



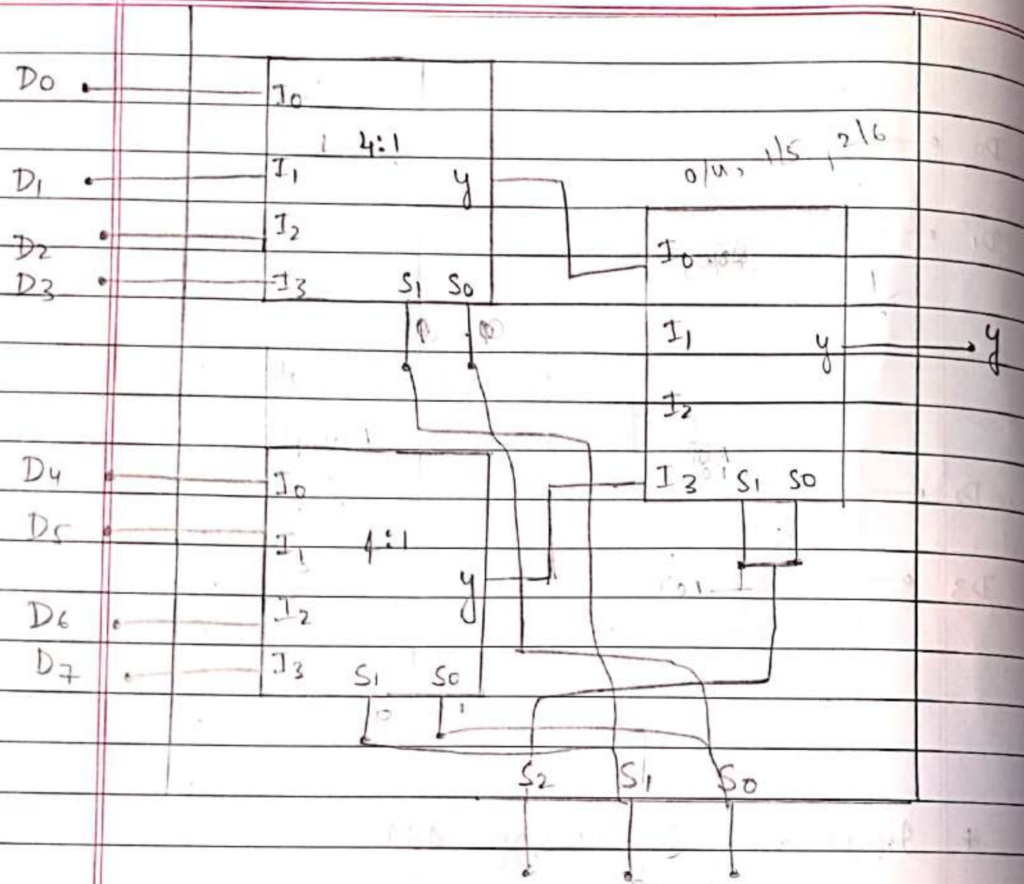

* Implement 4:1 using 2:1

| $S_1$ | $S_0$ | | $S_{o1}$ | $S_{o2}$ | $S_{o3}$ |
|---|---|---|---|---|---|
| 0 | 0 | $D_0$ | 0 | | 0 |
| 0 | 1 | $D_1$ | 1 | | 0 |
| 1 | 0 | $D_2$ | | 0 | 1 |
| 1 | 1 | $D_3$ | | 1 | 1 |




* Implement 8:1 using 4:1

## Multiplexer tree



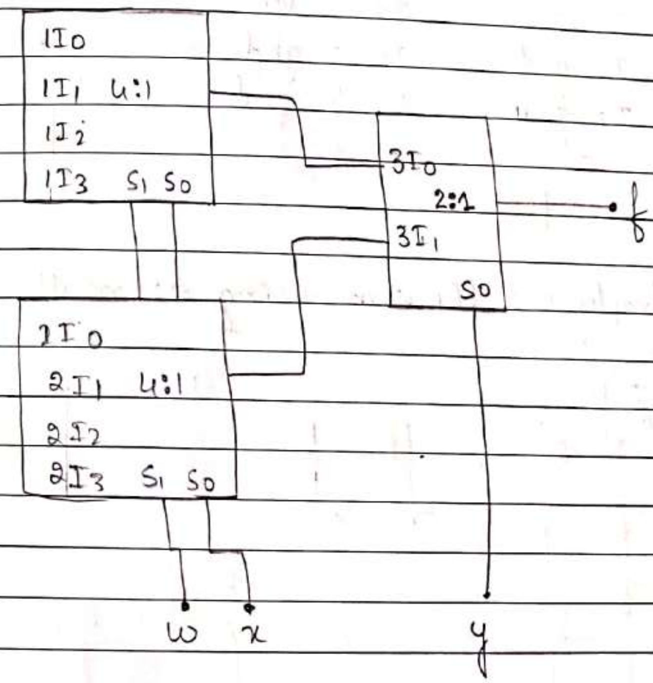

* Implement the following function using a mux tree with y as a selectline to the second level mux and w,x as select line to first level mux. Use 4:1 mux

$f(w,x,y,z) = \sum m(0,3,4,6,7)$.

K-map:

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

3I0 (columns 00, 01)    3I1 (columns 10, 11)



3I0 1I0 2I0

| w \ xz | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 |

1I1 1I2 1I3

| w \ xz | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |

2I1 2I2 2I3

$I_0 = \bar{z}$ $\quad I_4 = z$

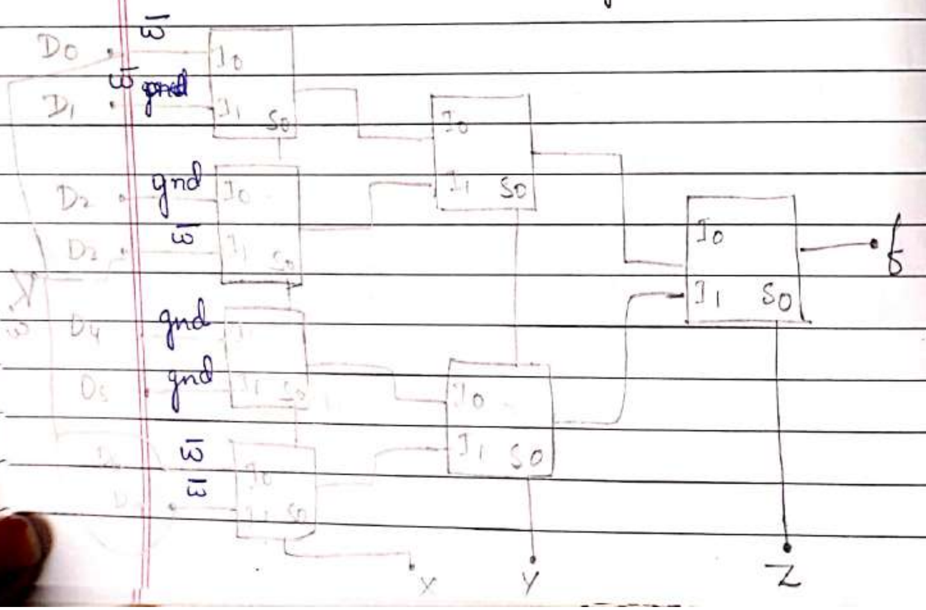
$I_1 = \bar{z}$ $\quad I_5 = vcc$

$I_2 = gnd$ $\quad I_6 = gnd$

$I_3 = gnd$ $\quad I_7 = gnd.$

$I_i = f(z)$

Implement the same using 2:1 multiplexer.




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

D0 D1 D2 D3 D4 D5 D6 D7

$f(\omega) = I_i$

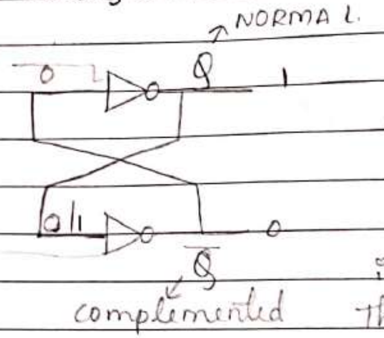
$D_0 = I_0 = \bar{\omega}$ $\quad \bar{\omega} = D_1$

$D_4 = gnd$ $\quad D_5 = gnd$

$D_2 = gnd$ $\quad D_6 = \bar{\omega}$

$D_3 = \bar{\omega}$ $\quad D_7 = \bar{\omega}$

# SEQUENTIAL CIRCUITS




Bistable element is capable of storing one bit

If Q=0 then bit stored in this element is 0. The bit stored is called as state

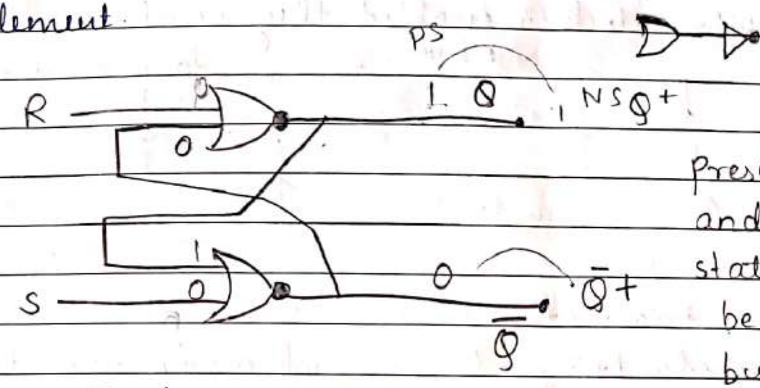
if it 0 it is reset state
if it 1 it is set state

Both of them can be one but it depends on manufacturing and diff in response time. If both react on same time then they don't become stable it is called metastable state. At some point of time one gate will respond in lesser time driving the other gate to stable state. metastable is also called forbidden state.

The normal o/p is the value stored or the state of the flipflops. The complement o/p is always the complement of normal o/p.
It is called bistable element ∴ it can take two stable states (set or reset)

If we want to store the user intended state then we need external inputs

SR-Latch
S and R can control the state of the bistable element. SR are external inputs to bistable element.




Present state and next state may be same but there is change

Q, Q̄ (present state represents previous input), Q+ and Q̄+ is next state. NS state depends on change in i/p & previous inputs

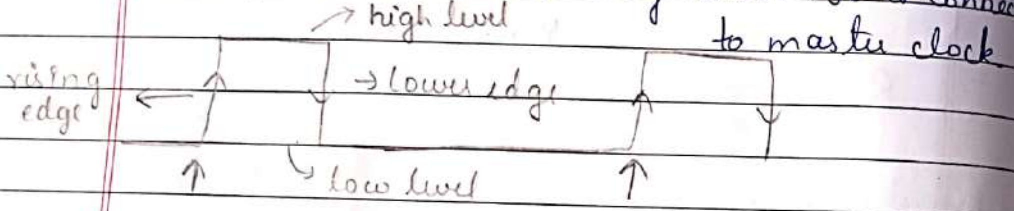
S.C. are the circuits in which o/p's are not just function of i/p's but also previous history of i/p's

Types of SC:-

i) Synchronous circuits (ASC)
ii) Asynchronous circuits $\times$ (SSC)

ASC: Change in input will immediately change the o/p

SSC: change in input will not immediately change out. They do it in discrete intervals of time. Most of hardwares are connected to single clock when clock is enabled they react. It is connected to master clock.



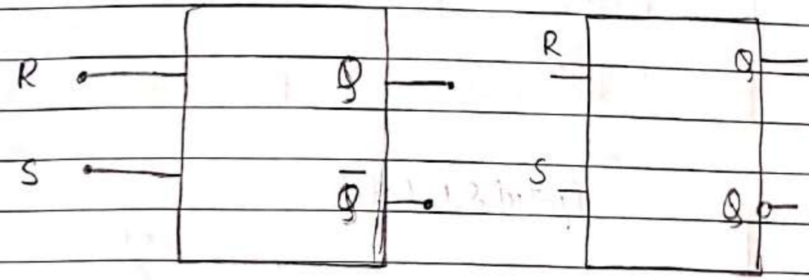

level triggered : high or low

the edge triggered : only when edge at that point it reacts

edge triggered :
at that that i/p's corresponding o/p is given

Any input at the points other than trigger point are rejected

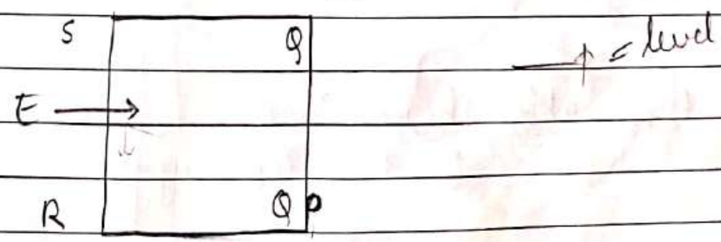
| R | S | $Q^+$ | $\bar{Q}^+$ | |
|---|---|---|---|---|
| 0 | 0 | $Q$ | $\bar{Q}$ | retain |
| 0 | 1 | 1 | 0 | set |
| 1 | 0 | 0 | 1 | reset |
| 1 | 1 | 0* | 0* | metastable |



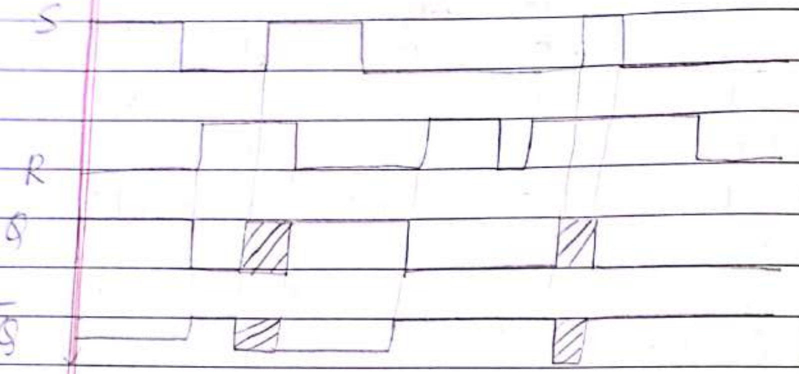

Block representation of SR-Latch.

Gated SR latch is also asynchronous SC that responds immediately on i/p change only when enabled.

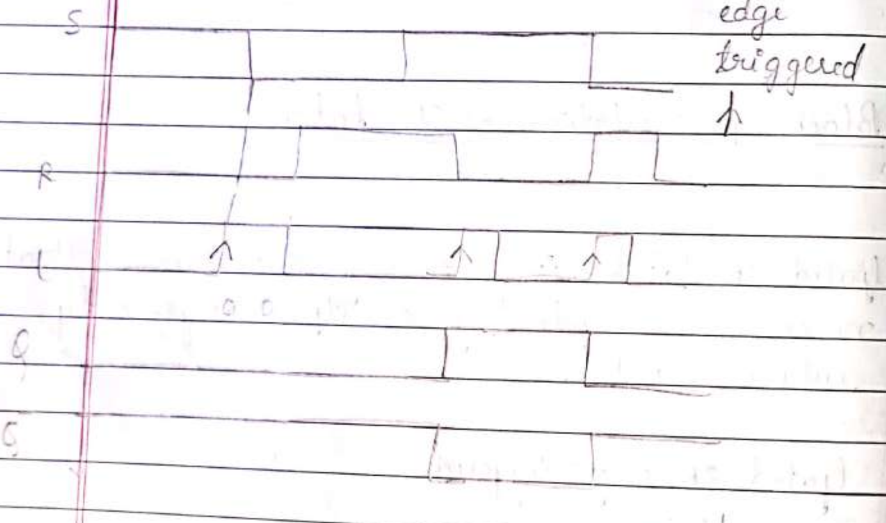
Gated SR edge triggered



## Ungated S-R Latch Behaviour

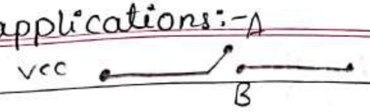


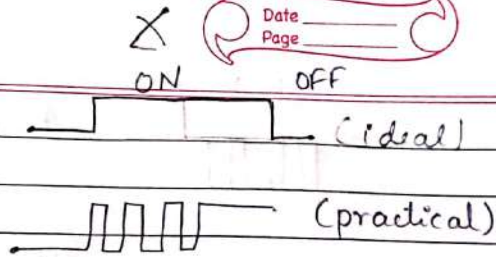

## Gated S-R latch




## applications:-

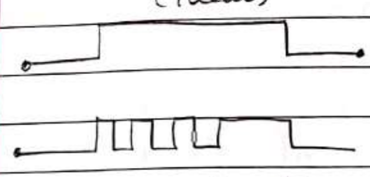
From A-B




* SR latch is used or acts as de-bouncing circuit in a switch

(Bouncing effect of switch)
↳ lead to errors or damage

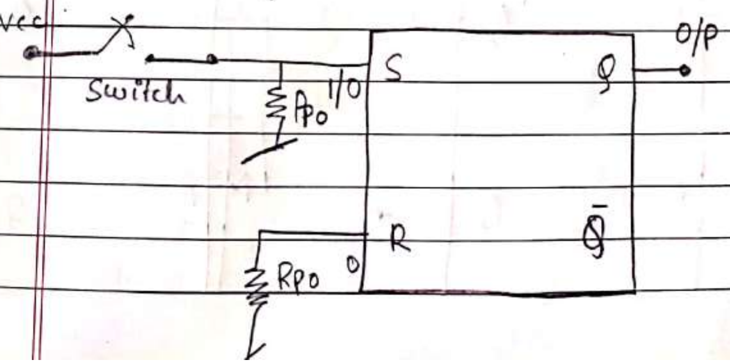



When switched is closed it doesn't reach ON state immediately as expected in the ideal case it goes through series of bounces to acheive a permanent ON state which happens in smaller amount of time.

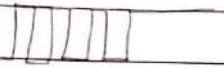
pull down ressistor = always ground

This is called bouncing effect of switch. It causes error and damages the components.

### Debouncing circuit:-

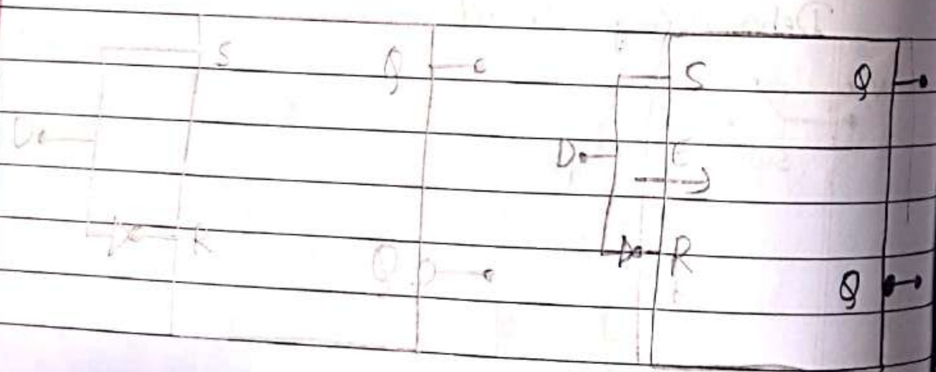



pull down resistors maintain 0 at i/p unless external high at that point is provided bouncing effect

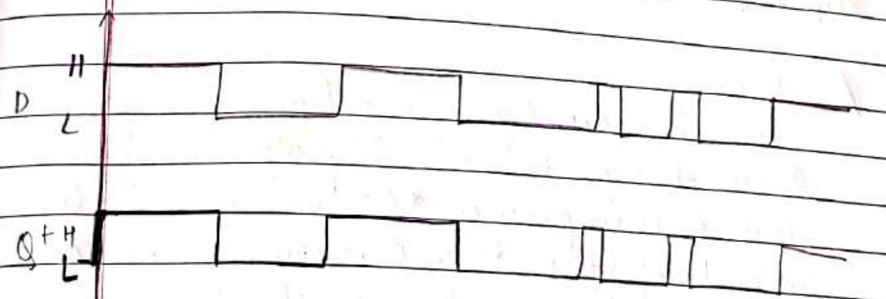



when switch is high $S=1; R=0$ so it sets than during bouncing switch is low so $S=0; R=0$ so it retain again it bounces so $S=1; R=0$ so it is set and again retains the state to keep it ↑ in reducing the bouncing effect. Then some circuits are used to reset it when during bouncing switch is low the resistors pull down the value to 0

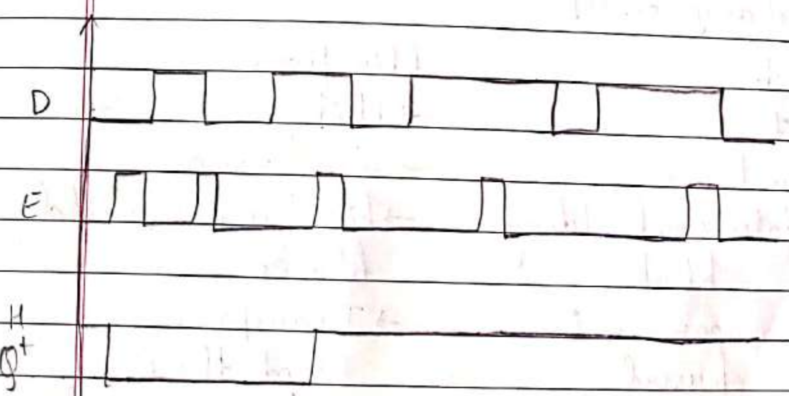
D-latch:




D-latch




Gated D-latch:



Latch can store 1bit at a time. As soon as i/p is changed o/p is changed so asynchronous sc

## * Flip-Flops:-

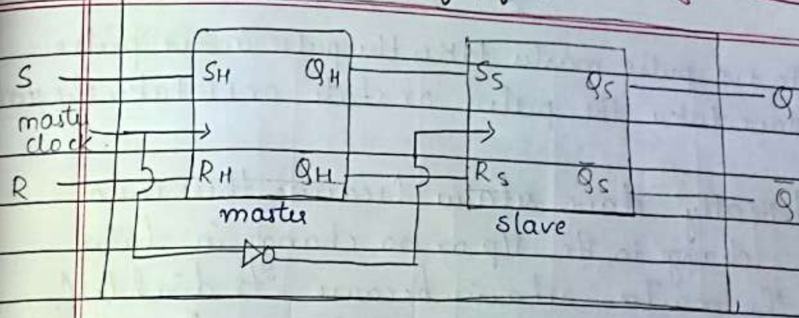
/* 1-bit
asynchronous (gates) - Transparency
As i/p changes immediately o/p changes. This is transparency */. Transparency adds to speed of latch. It is fast-speed device */

It is also sc. It can store 1-bit. The change in external i/p to get reflected o/p there is delay. But delay is expected. These are synchronous sc. It is controlled by master clock. It takes one complete clock cycle to reflect the change on o/p

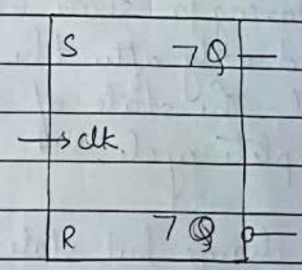
| Latch | Flip-flops |
|---|---|
| → 1bit | → 1bit |
| → Asynchronous | → Synchronous |
| → Gates are building blocks | → Latches are building blocks |
| → Transparency is observed. | → Transparency is not there |
| → High-speed devices | → Low-speed devices. An expected delay is there. |

// edge triggered

## Pulse-triggered flipflops: (Master-Slave flipflops)



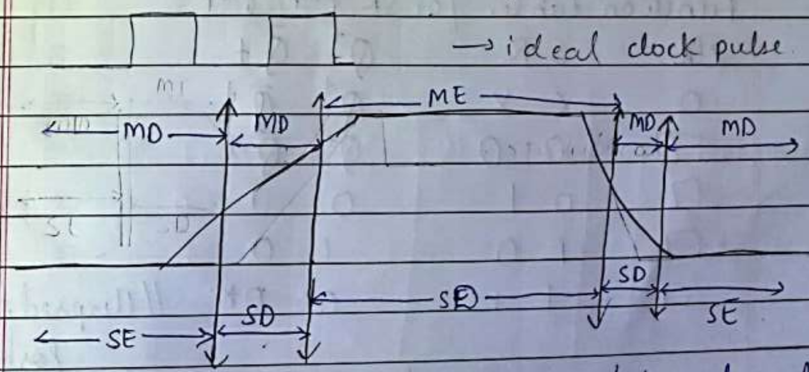

SR flipflops:-




/* clock is periodic train of pulses; enable is train of pulses but not periodic */

State of slave is state of flipflop

Clock adds a feature of being synchronous




slave = ret // } set
mas = ret // disabled
FF = ret

slave = set
mas = change
FF = retain

slave = change
mas = retain
FF =

$0 - 3$
$\overline{2 \text{ bits}}$.

In +ve pulse master takes the pulse in -ve pulse slave take the pulse or state or FF takes the state

Initially slave retains because there is no change in its i/p or no change in state. the master retains because it's disabled. Even while storing a state it takes the new i/p's and only after the complete cycle o/p is reflected. The state of FF is effected after one complete cycle.

In middle one change but state of FF is not effected

Flip-flop can store value even when external i/p change

Function table for SR flip-flop:

| clk | S | R | $Q^+$ | $\bar{Q}^+$ |
|---|---|---|---|---|
| 0 | X | X | $Q^+$ | $\bar{Q}^+$ |
| ⎍ | 0 | 0 | $Q^+$ | $\bar{Q}^+$ |
| ⎍ | 0 | 1 | 0 | 1 |
| ⎍ | 1 | 0 | 1 | 0 |
| ⎍ | 1 | 1 | 0* | 0* |

// Unpredicated or forbidden

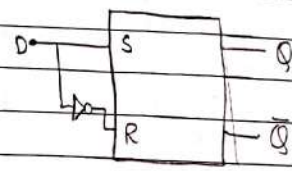
Limitation of SR flipflop which is overcome by JK flip-flop.

Flip-flops — Pulse T / level T (Master-Slave)
— Edge Tri

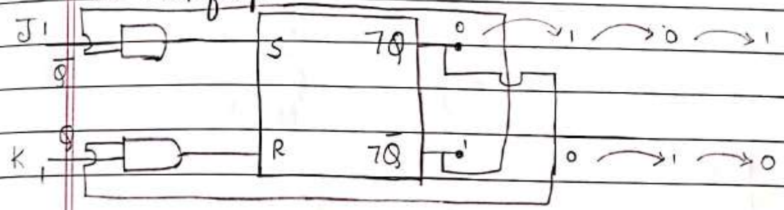
JK-Flipflop:-

D-Flipflop (SR with D-latch):-

| D | clk | $\bar{D}$ | $Q$ | $\bar{Q}$ |
|---|---|---|---|---|
| 0 | ⎍ | 1 | 0 | 1 |
| 1 | ⎍ | 0 | 1 | 0 |
| i/p | X | i/p | $Q$ | $\bar{Q}$ |

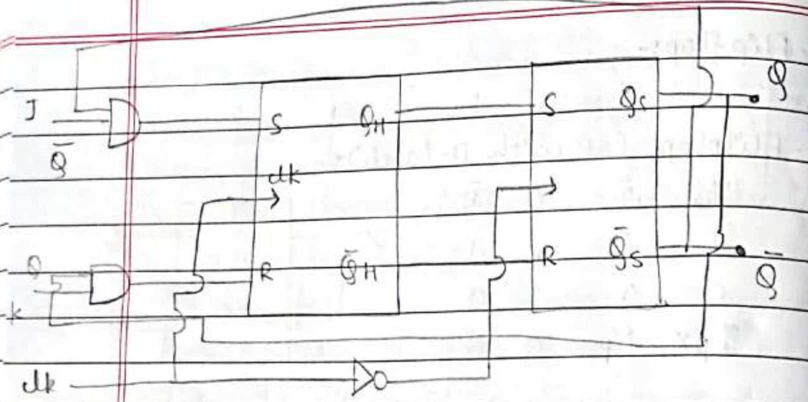


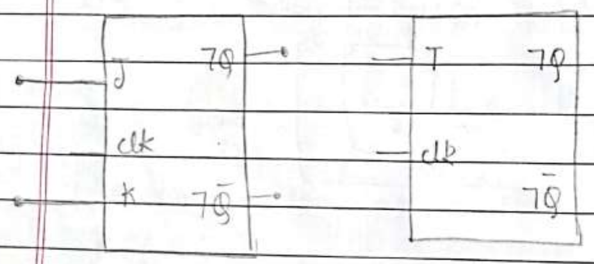

JK-Flipflop




| clk | J | K | $Q^+$ | $\bar{Q}^+$ | |
|---|---|---|---|---|---|
| ⎍ | 0 | 0 | $Q^+$ | $\bar{Q}^+$ | (retain) |
| ⎍ | 0 | 1 | 0 | 1 | (reset) |
| ⎍ | 1 | 0 | 1 | 0 | (set) |
| X | i/p | i/p | $Q^+$ | $\bar{Q}^+$ | (retain) |
| ⎍ | 1 | 1 | $\bar{Q}$ | $Q$ | (toggle) |

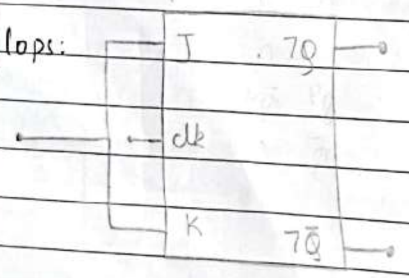
## JK flipflop:



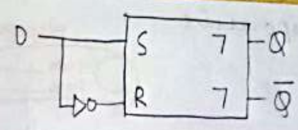




JK flipflop T-flipflop

T-flipflops:

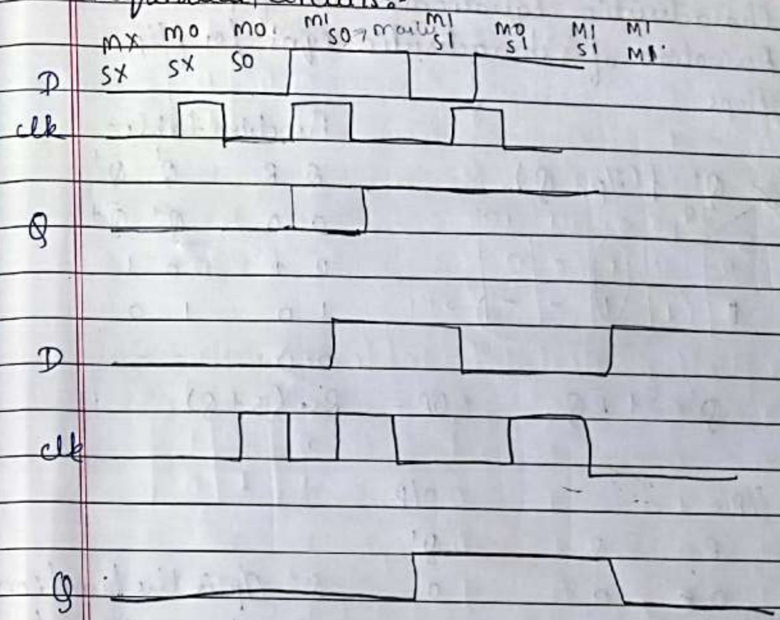



| clk | T | $Q^t$ | $\bar{Q}^+$ |
|---|---|---|---|
| $\sqcap$ | 0 | $Q$ | $\bar{Q}$ |
| $\sqcap$ | 1 | $Q$ | $Q$ |
| X | retain | | |



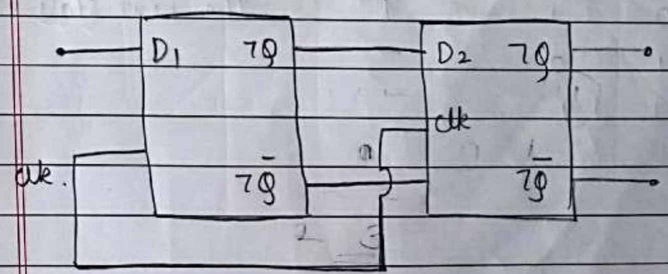

## D-flipflop:-

## Sequential Circuits:-




Serial in serial out

Serial in parallel out




shift register

Edge triggered is more responsive than pulse triggered.

## Characteristic Equations:- (SR-flipflop)

Derivation of characteristic eqn's for flip flops

$Q^+ = f(i/p, Q)$.

| S \ RQ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | - | - |

Function table:

| S | R | $Q^+$ | $\bar{Q}^+$ |
|---|---|---|---|
| 0 | 0 | $Q^+$ | $\bar{Q}^+$ |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 0 0 | | - | - |

$Q' = S + \bar{R}Q$ — OR — $\bar{R} \cdot (S + Q)$

Next state table:-

| i/p's | | | o/p |
|---|---|---|---|
| S | R | Q | $Q^+$ |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | — |
| 1 | 1 | 1 | — |

Op is the function is set of external inputs and present state. Using function table we design the next state table.

## JK-flipflop:- (fun

| J | K | $Q^+$ | $\bar{Q}^+$ | clk |
|---|---|---|---|---|
| 0 | 0 | $Q^+$ | $\bar{Q}^+$ | Ω |
| 0 | 1 | 0 | 1 | Ω |
| 1 | 0 | 1 | 0 | Ω |
| 1 | 1 | $\bar{Q}$ | $Q$ | Ω |
| i/p | i/p | $Q^+$ | $\bar{Q}^+$. | X |

Next-state table:

| i/p's | | | | |
|---|---|---|---|---|
| J | K | Q | $Q^+$ | $\bar{Q}^+$ |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | [illegible] | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | [illegible] | 1 |
| 1 | 1 | 1 | 0 | 0 |

| J \ KQ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 |

$Q^+ = \bar{K}Q + J\bar{Q}$

$Q^+ = (J + Q) \cdot (\bar{K} + \bar{Q})$

Date ______
Page ______

* D-flipflop: Next state table

| D | Q | $Q^+$ | $\bar{Q}^+$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0. |

$Q^+$

| D \ Q | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 1 | 1 |

SOP = $D = Q^+$

$Q^+ = D$

* T flipflop: Next state table

| T | Q | $Q^+$ | $\bar{Q}^+$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | [illegible] | 0 | 1 |

$Q^+$

| T \ Q | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 0 |

$Q^+ = T\bar{Q} + \bar{T}Q$

$= T \oplus Q$

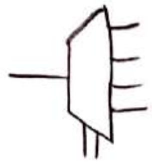
$Q^+ =$ [illegible] $(T+Q)\cdot(\bar{T}+\bar{Q})$

T flipflop:- function table

| clk | T | $Q^+$ | $\bar{Q}^+$ |
|---|---|---|---|
| ⎍ | 0 | Q | $\bar{Q}$ |
| ⎍ | 1 | $\bar{Q}$ | Q |
| X | i/p | retain | |

Date ______
Page ______

SR-flipflops:-



$1:2^n$ (demux)
n select lines

## Decimal Adder:-

BCD - binary coded decimal and not the exact decimal (its all binary)

Answer

| | (0-9) | |
|---|---|---|
| Decimal no | BCD | Binary |
| 4 | 0100 | 0100 |
| 14 | 00010100 | 1110 |

BCD 256 entries

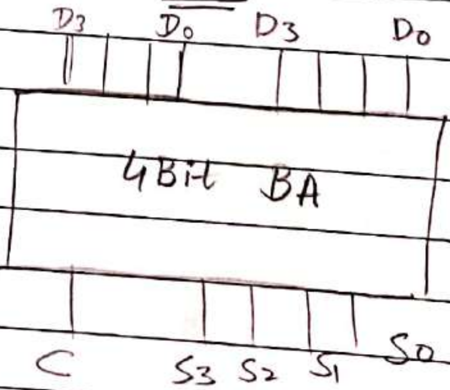
Every digit → 4 bits

A = 9 B = 9 (0-9) (0-9)

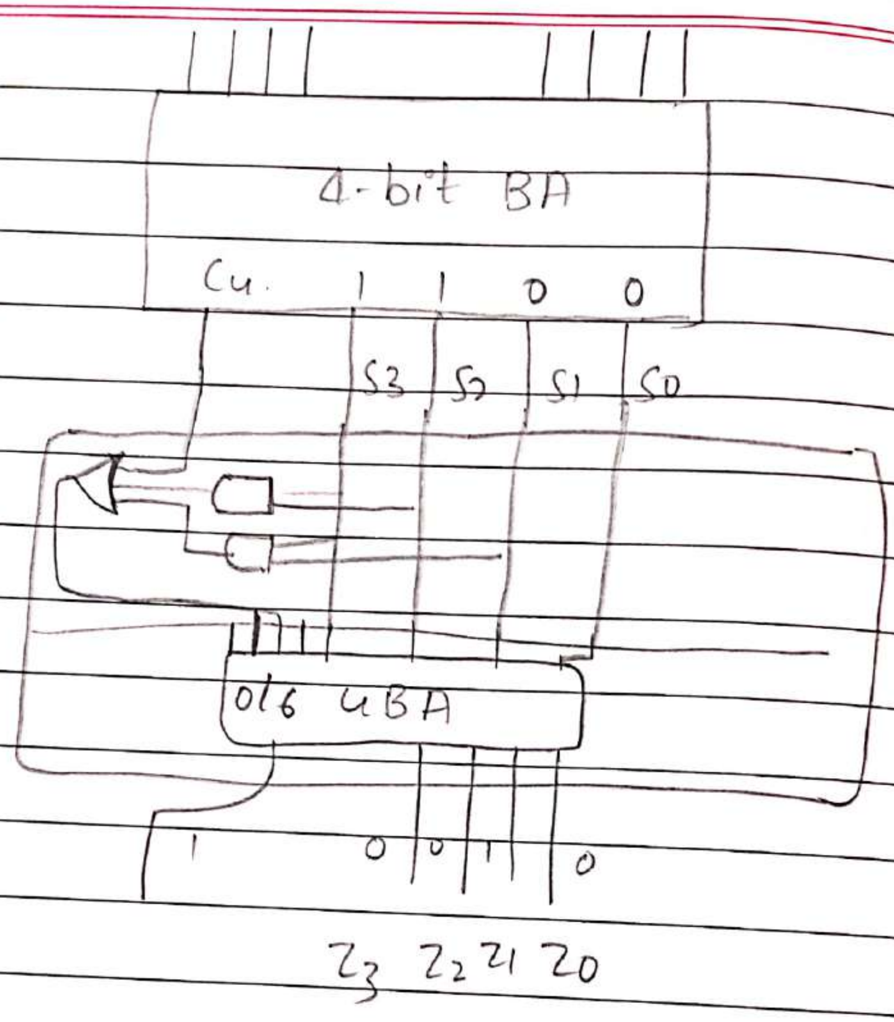
all 256 entries will be either of this 18

0001 1000 (18)

always 5 bits




| $S_3S_2$ \ $S_1S_0$ | | | | |
|---|---|---|---|---|
| | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 |
| | 1 | 1 | 1 | 1 |
| | 0 | 0 | 1 | 1 |


4-bit BA
Cu.
1 1 0 0
S3 S2 S1 S0
0/6 4BA
1 0 0 1 0
Z3 Z2 Z1 Z0