

## → Fault-

Suppose there is a fault due to which -  
Input  $y$  is stuck at 0

∴ If  $x=0, y=1$  input is given by user, then  
 $x=0, y=0$  will be the input that would be going to circuit.

- How to detect whether an "and" gate has a fault due to which  $y$  is stuck at 0.

Give Input  $x=1$  and  $y=1$ . If output is 0, <sup>then</sup> there is a fault

- To test fault - input should be as such that the outputs in case of fault and no fault are different from each other.

- OR -  $x+y$

$y$  is stuck at 0.

How to detect it

$xy = 01 \Rightarrow$  Output is 0

- XOR -  $xy \Rightarrow 00 \rightarrow 0; 01 \rightarrow 1; 10 \rightarrow 1; 11 \rightarrow 0$   
 Let  $y$  be stuck at 0.

$y=1, x=0$  → This helps to detect fault.

## Complementation fault at input -

This means whatever the user input is, the complement of that would be passed to the system.

Suppose - AND gate - complementation fault at y

Suppose  $x = 0$

and user pass  $y = 1$

then Output = 0 ; expected output = 1

User pass  $y = 0$

then output = 1 ; expected output = 0

$\therefore$  to detect complementation fault at y in AND gate,  
 $x = 1, y = \text{anything}$

XOR : complementation at y.

~~we can~~  $x \oplus y$  and  $x \oplus y'$  are <sup>always</sup> <sub>opposite</sub>

Therefore this can be detected for any example of x and y.

→ Fault Locate - Where is the fault when it is known there is faults.

## Permitted Instructions -

$ib \leftarrow \text{number}$

$s \leftarrow ib \quad t \leftarrow ib$

$ib \leftarrow s + t \quad -, *, / \text{ similar}$

$a \leftarrow ib$

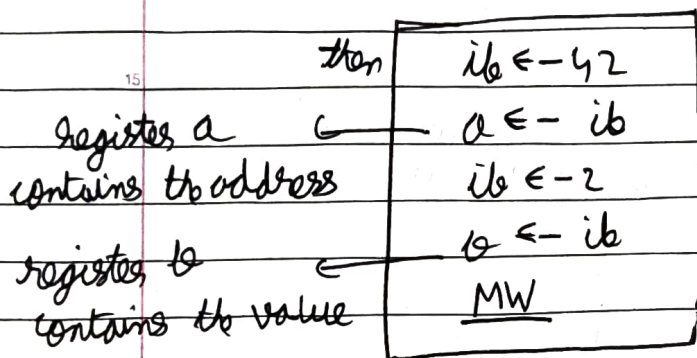
$ib \leftarrow a$

$\text{print}(ib)$

$MR \equiv \text{memory read} \Rightarrow b \leftarrow [a]$

$MW \equiv \text{memory write} \Rightarrow [a] \leftarrow b$

Suppose we want to store 2 at memory address = 42



'a' is also called MAR  $\equiv$  memory <sup>address</sup> Register  
'b' " " " MBR = " ~~Register~~ Register buffer

\* a, b are general purpose registers also but for memory read and write instructions  
a and b act as memory registers

C acts as PC

\*  $ib \leftarrow PC$  is valid



-  $+$   $\equiv$  OR ;  $\cdot$   $\equiv$  AND ;  $'$   $\equiv$  NOT

i) Boolean exp. true for  $xy=1$  only  $\equiv xy'$

ii) true for  $xy=00$  and  $xy=11 \equiv \boxed{x'y' + xy}$

$\hookrightarrow$  Sum of product form

Another Possible Ans -

Product of Sum form

$$\Rightarrow \boxed{((x+y)' \cdot (x'+y'))'}$$

Another ans  $\equiv \boxed{(x+y') \cdot (x'+y)}$

iii) SOP form; true for  $xyz = 010 ; 110$

Ans  $= \boxed{y z'}$  ( $\because x$  can be anything)

iv) true for  $xyz = 010$  ~~and~~ ;  $110$  and  $111$

$$\Rightarrow 010 + 110 + 111$$

$$\Rightarrow 010 + 110 + 110 + 111 \quad (\because a+a=a)$$

$$\Rightarrow 10 + 11$$

$$\Rightarrow yz' + xy$$

\* Prove  $\equiv \boxed{a + a'b = a + b}$

~~$a + a'b$~~   $a + (a'b) = (a+a') \cdot (a+b)$   
 $= a + b$

14

Camlin	Page
Date	/ /

if  $a$  and  $b$  are sets, then

In sets,  $a - b = a \cdot b'$   $\equiv a \text{ AND (NOT } b)$