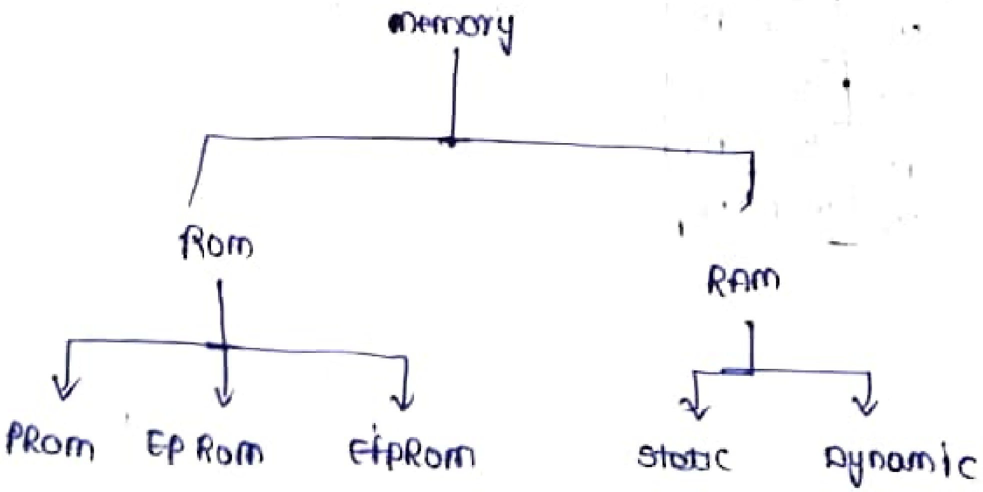
UNIT-4

Design with programmable logic devices

⊛ **PLD :-**

It is integrated circuit with programmable gates with divided in to and array in to or aray.

Device	AND--array	OR-array
PRom	fixed	Programmable
PLA	Programmable	Programmable
PAL	Programmable	fixed



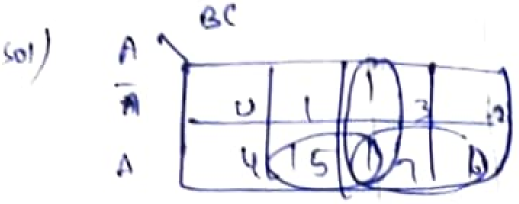
⊛ Rom is the NON Volatile memory

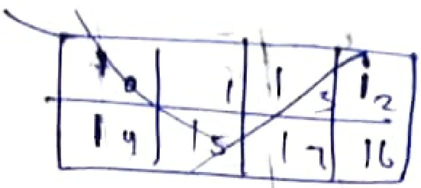
⊛ RAM is the Volatile memory

8 Implement the combination with a PLA having 3 input · 4 product terms and 2 output of the product $f(A,B,C) = \Sigma m(3,5,6,7)$

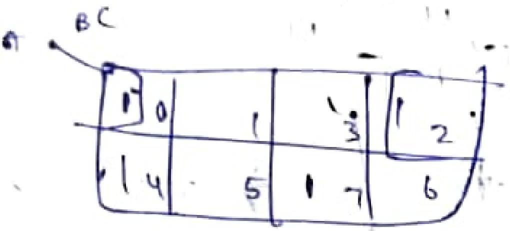
$f_2(A,B,C) = \Sigma m(0,2,4,7)$

Sol)

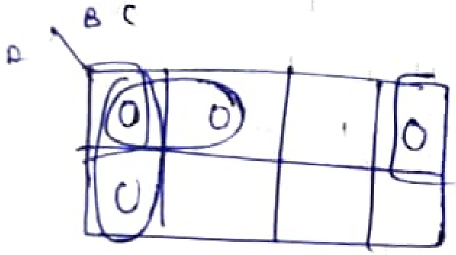


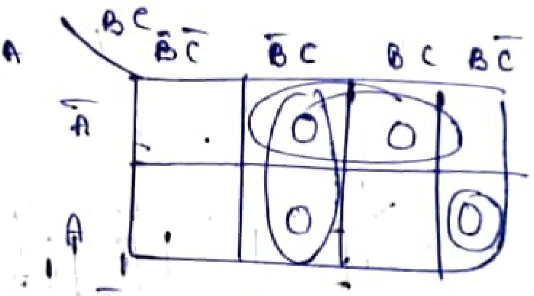


$f_1 = BC + AB + AC$



$f_2 = \overline{A}\overline{C} + ABC + \overline{B}\overline{C}$





$\overline{f_1} = (B+C)(A+B)(A+C)$

$\overline{f_2} = (\overline{\overline{A}+\overline{B}+C})(B+\overline{C})$
$(A+\overline{C})$

$f_2 = (\overline{B+C}) \cdot (\overline{A+B})(\overline{A+C})$

$f_2 = \overline{(\overline{A}+\overline{B}+C)} + \overline{(B+\overline{C})} + \overline{(A+\overline{C})}$

Programmable table:

$\overline{f_1} = \overline{B}\overline{C} + \overline{A}\overline{B} + \overline{A}\overline{C}$

Product term	A B C	$f_1(C)$	$f_2(T)$
$\overline{B}\overline{C}$	- 0 0	1	1
$\overline{A}\overline{C}$	0 - 0	1	1
$\overline{A}\overline{B}$	0 0 -	1	0
ABC	1 1 1	0	1

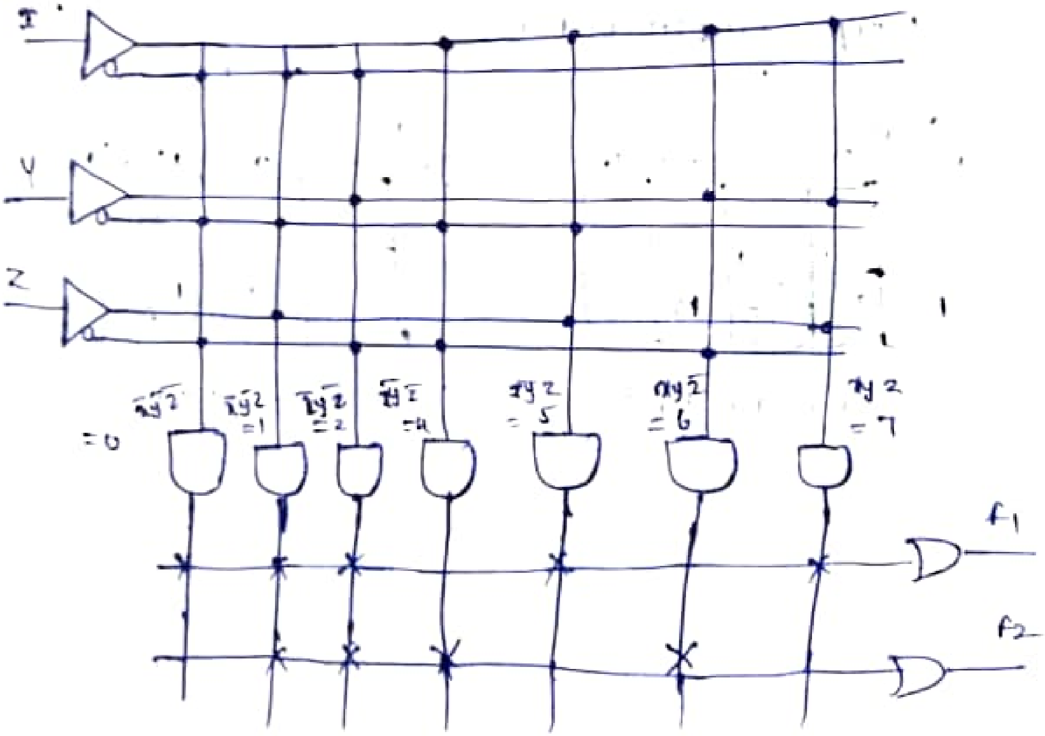
Q design a logic circuit for the following boolean expression using PROM. $f_1(x,y,z) = \Sigma_m(0,1,2,5,7)$

$f_2(x,y,z) = \Sigma m(1,2,4,6)$

Sol:]

Step 1:

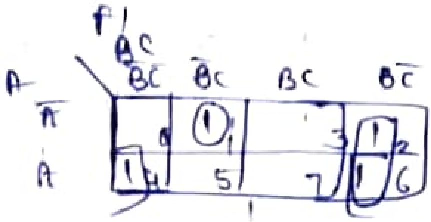
x	y	z	F1	F2
0	0	0	1	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	0	
1	0	1	1	0
1	1	0	0	1
1	1	1	1	0

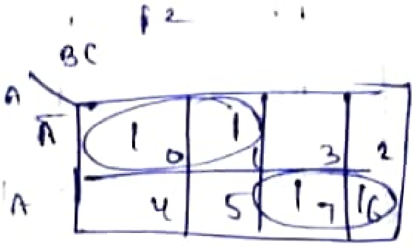


sol5) $f1(A,B,C) = \Sigma m(1,2,4,6)$

$f2(A,B,C) = \Sigma m(0,1,6,7)$

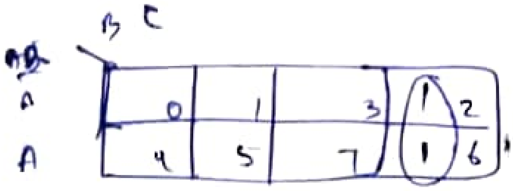
$f3(A,B,C) = \Sigma m(2,6)$





$f1 = B\bar{C} + \bar{C}A + \bar{B}C\bar{A}$

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



$f_3 = B\bar{C}$

Present ~~state~~ term	Input A	B	C	Output f1	f2	f3
$B\bar{C}$	–	1	0	1	0	1
$\bar{C}A$	1	–	0	1	0	0
$\bar{B}C\bar{A}$	0	0	1	1	0	0
$\bar{B}\bar{A}$	0	0	–	0	1	0
BA	1	1	–	0	1	0

Date
08/06/2023

* A combinational logic circuit has 4 input and two o/p's. The o/p is 1, gives I o/p when the i/p combination is greater than or equal to 1001 and the o/p 2 gives high o/p when i/p combination is < 1001. implement the crict with PLA

Sol:

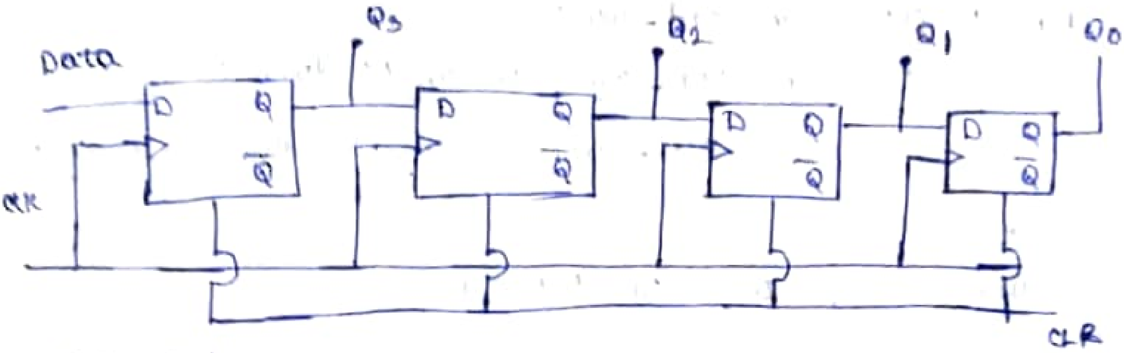
	A	B	C	D	f_1	f_2
0	0	0	0	0	0	1
1	0	0	0	1	0	1
2	0	0	1	0	0	1
3	0	0	1	1	0	1
4	0	1	0	0	0	1
5	0	1	0	1	0	1
6	0	1	1	0	0	1
7	0	1	1	1	0	1
8	1	0	0	0	0	1
9	1	0	0	1	1	0
10	1	0	1	0	1	0
11	1	0	1	1	1	0
12	1	1	0	0	1	0
13	1	1	0	1	1	0
14	1	1	1	0	1	0
15	1	1	1	1	1	0

$f_1(A,B,C,D) = \Sigma m(9,10,11,12,13,14,15) = AB + AD + AC$

$f_2(A,B,C,D) = \Sigma m(0,1,2,3,4,5,6,7,8) = \bar{A} + \bar{B}\bar{C}\bar{D}$

Date
16/08/2022

serial - in - Parallel - out - shift - Register



VHDL code:-

```
      library IEEE;
      use ieee.std_logic_1164.all;
     entity SIPO is
   Port ( clk clear : in std_logic;
            Data    : in std_logic;
             Q      : out std_logic vector (3 down to 0)
           end SIPO
     Architecture arch of SIPO is
      begin
      process (CLK)
      begin
        if clear = '1' then
            Q = "0000"
    else if (CLK 'event and clk = '1') then
        Q(3 down to 1) <= Q(2 down to 0);
             Q(0) <= Data
           end if
          end process
           end arch
```

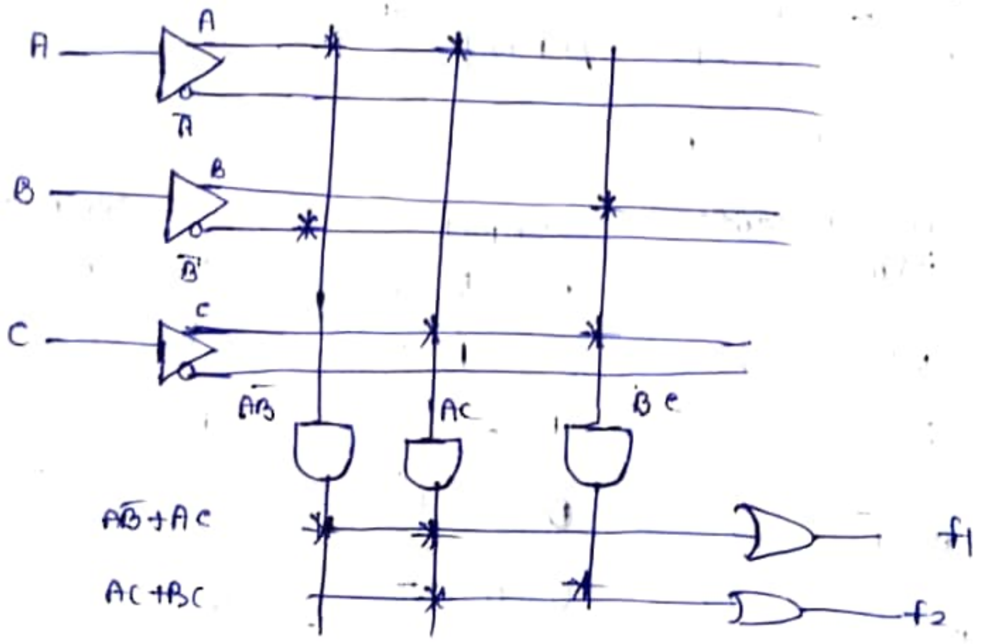
Step-3: PLA programmable table

Present term	Input A	B	C	Output f_1	f_2
$A\bar{B}$	1	0	—	1	0
AC	1	—	1	1	1
BC	—	1	1	0	1

A-1
$\bar{B}$=0

A-1
$\bar{B}$-0

Step-4: logic diagram



Implement the combinational circuit for the function using PLA $f_1(A,B,C)$ $\Sigma m(1,2,4,6)$; $f_2(A,B,C) = \Sigma m(0,1,6,7)$, $f_3(A,B,C)$ $\Sigma m(2,6)$

Date
01/06/2022

Implement the combinational circuit for the function using PLA $f_1(A,B,C) = \Sigma m(4,5,7)$

$f_2(F,B,C) = \Sigma m(3,5,7)$

sol:

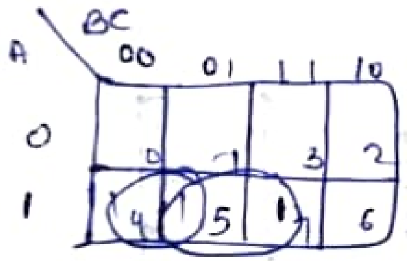
Step -1:

Input			output	
A	B	C	f_1	F_2
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	01	0
1	1	1	1	1

Step 2:

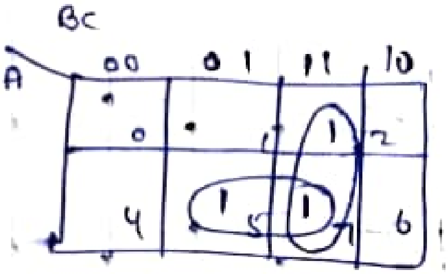
k-map simplification

f_1



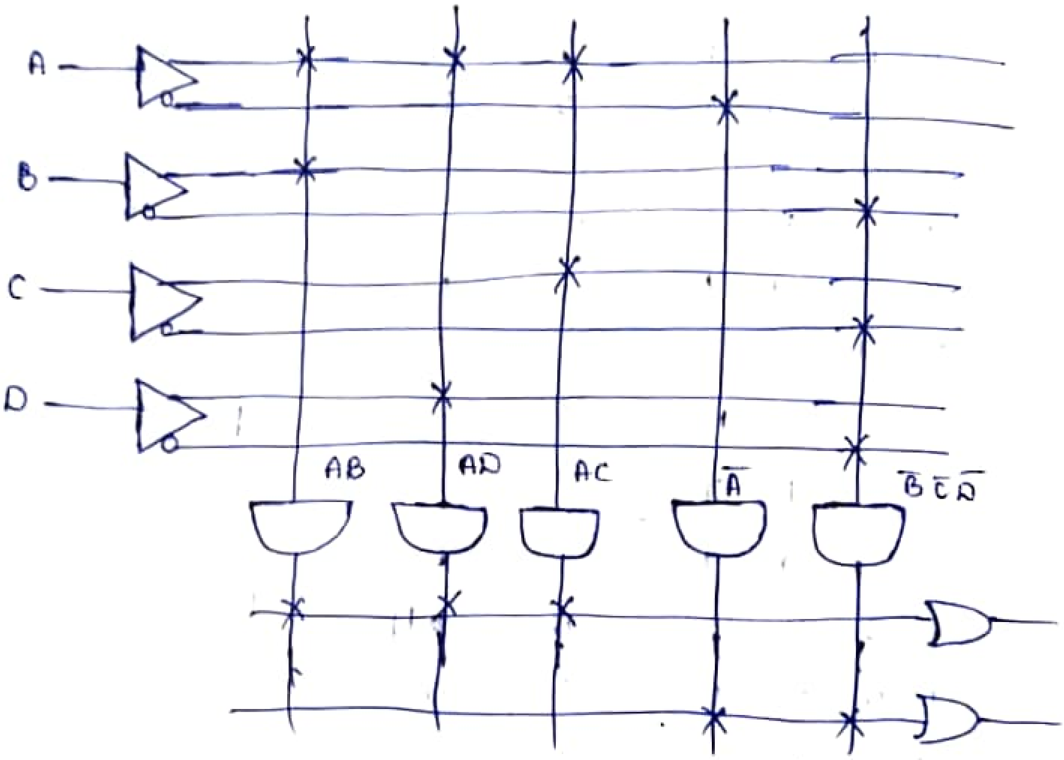
$f_1 = A\bar{B} + AC$

f_2



$f_2 = AC + BC$

Product	A	B	C	D	f1	f2
	i/P				O/P	
A B	1	1	-	-	1	0
A D	1	-	-	1	1	0
A C	1	-	1	-	1	0
$\bar{A}$	0	-	-	-	0	1
$\bar{B}\bar{C}\bar{D}$	-	0	0	0	0	1



7) Design a BCD to excess 3 code converted using PAL (Programmable array logic).

B3	B2	B1	B0	+3 ⇒	E3	E2	E1	E0
0	0	0	0		0	0	1	1
0	0	0	1		0	1	0	0
0	0	1	0		0	1	0	1
0	0	1	1		0	1	1	0
0	1	0	0		0	1	1	1
0	1	0	1		1	0	0	0
0	1	1	0		1	0	0	1
0	1	1	1		1	0	1	0
1	0	0	0		1	0	1	1
1	0	0	1		1	1	0	0

$E_3(B_3\ B_2\ B_1\ B_0) = \Sigma m(5,6,7,8,9) = B_3 + B_2B_0 + B_2B_1$

B_3B_2 \ B_1B_0	00	01	11	10
00				
01		1	1	1
11	X	X	X	X
10	1	1	X	X

$E_2(B_3\ B_2\ B_1\ B_0) = \Sigma m(1,2,3,4,9) = B_1\bar{B_2} + B_0\bar{B_2} + B_2\bar{B_0}\bar{B_1}$

$E_1(B_3\ B_2\ B_1\ B_0) = \Sigma m(0,3,4,7,8) = \bar{B_0}\bar{B_1} + B_0B_1$

$E_0(B_3\ B_2\ B_1\ B_0) = \Sigma m(0,2,4,6,8) = \bar{B_0}$

Difference b/w PROM, PLA & PAL

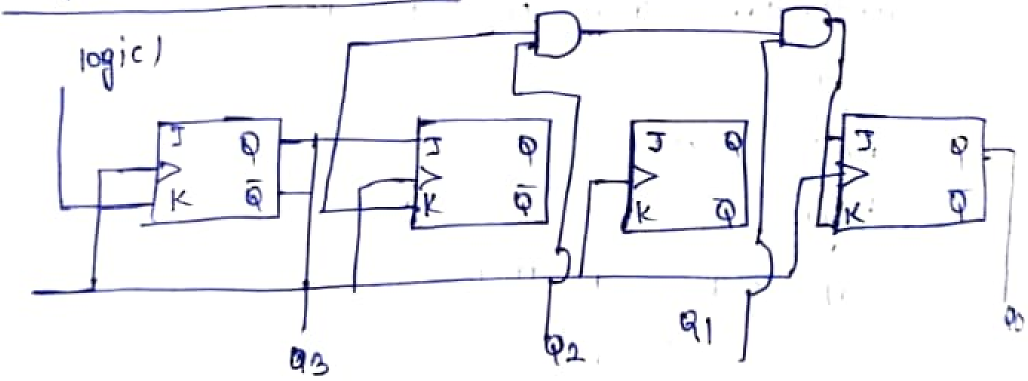
PROM	PLA	PAL
1) fixed AND array programmable OR array	1) Both AND & OR are programmable	1) Programmable AND array & fixed OR array
2) All the minterms are decoded	2) AND array can be programmed to get desired minterm →	
3) only boolean function in standard SOP form can implemented	3) Any boolean function in sop form can be implemented. →	
4) PROM very cheaper for ic's	4) It is closely than PLA & PROM	It is also cheaper

PIPO

```
entity PIPO is
Port (, CLK : In std-logic
        D : in std-logic-vector (3 down to 0);
        Q : out std-logic-vector (3 down to 0););
  end PIPO
  architecture arch of PIPO, is
  begin
 Process (CLK)
 begin
    if (CLK 'event and CLK = '1') then
        Q <= D;
    end if
    end process
    end arch
```

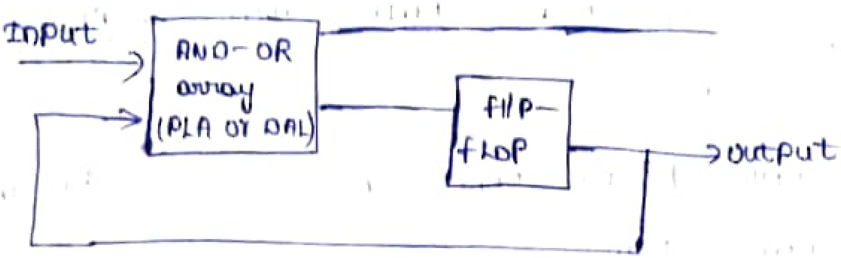
Synchronous → Common
CLK Pulse

synchronous up counter

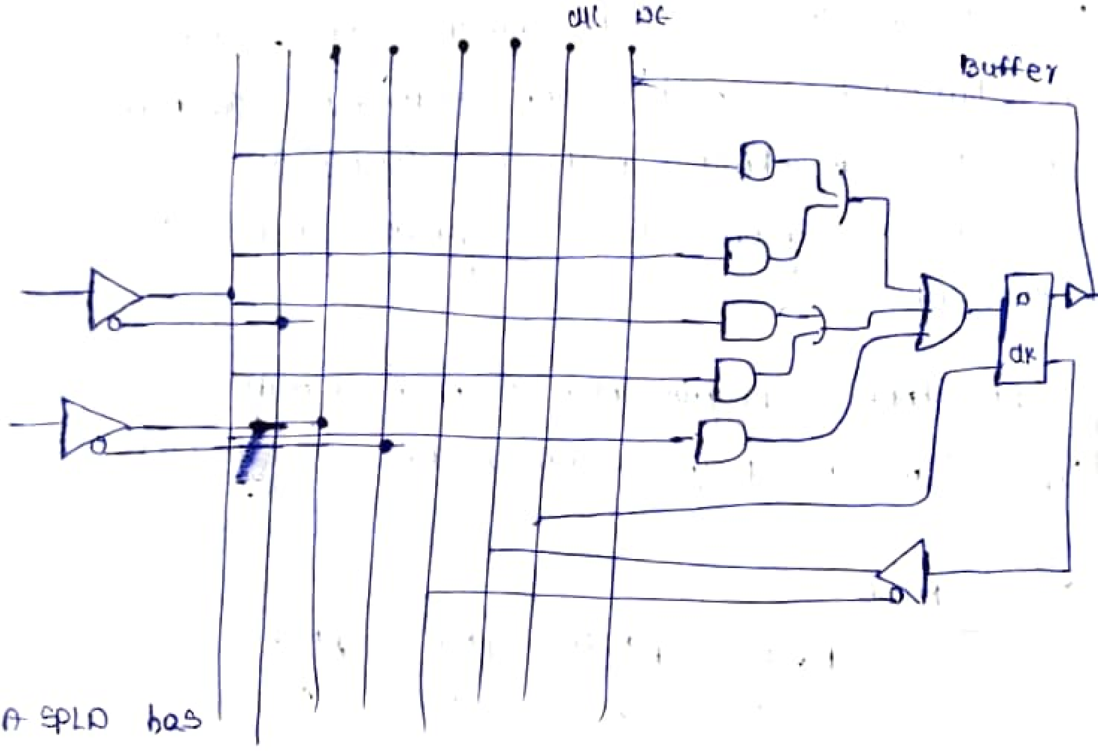


```
library ~~ieee~~
use ieee.std-logic-1164.all
  entity upcounter is
   Port (clK, RST : In std-logic
        count : in out std-logic ); .vector (3 down to
     end up counter.
```

SPLD



Basic macro cell logic:-



A SPLD has from 3 to 10 . macro cells unity on IC.

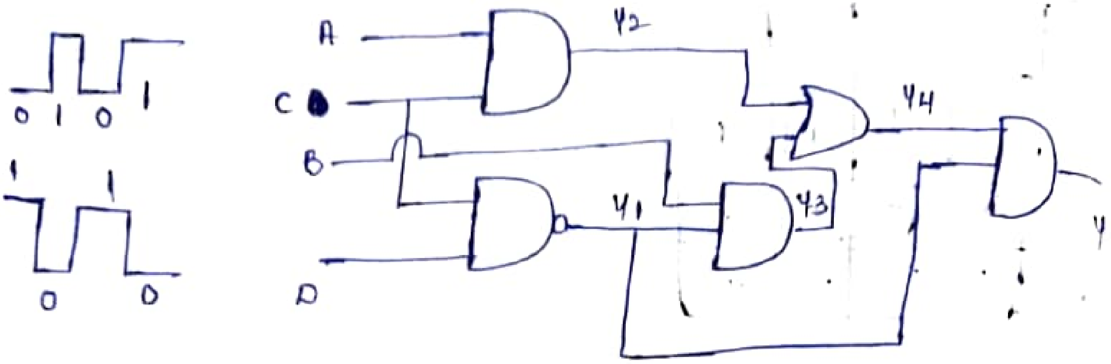
① A Dynamic hazard is defined as a transition change occur in 3 or more times. at an o/p terminal of a logic network, when the o/p is supposed to change only is during a transition b/w two . i/p states, differing in the value of 1 variable.

② essential hazard is a type of hazard that exceeds only in asynchronous sequential circuit with 2 or more feed back

Date
10/06/2022

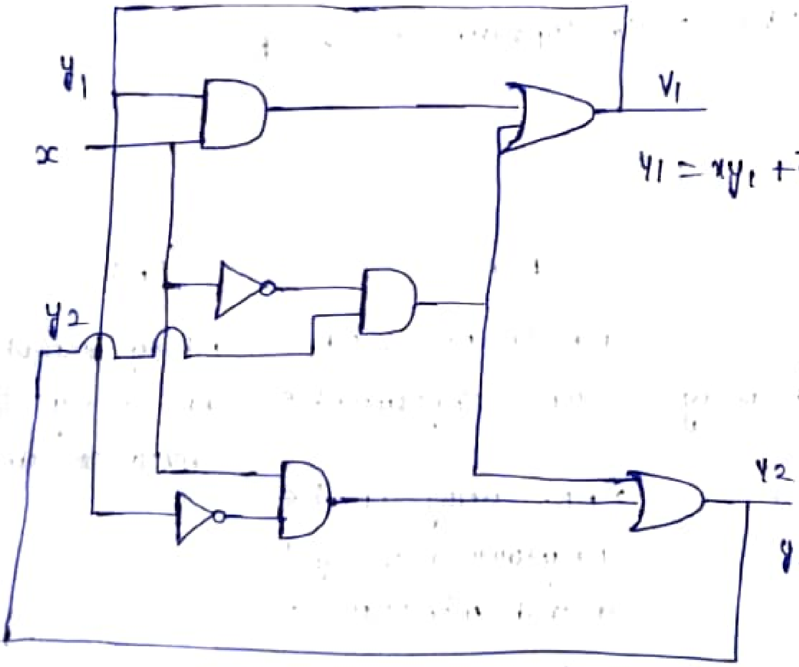
4

Dynamic hazard (combinational circuit)



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

essential hazard (asynchronous sequen



$Y_1 = xy_1 + \bar{x}y_2$

$Y_2 = x\bar{y}_1 + \bar{x}y_2$

Sequential programmable devices

i) sequential (or simple) programmable logic device

ii) complex programmable logic devices (CPLD)

iii) field - programmable logic device

® commerial CPLDS have up to 50 PLD blocks.

Date
15/06/22

UNIT-5

FPGA (field programmable gate array)

® FPGA logic block consists configurable logic blocks (CLB)

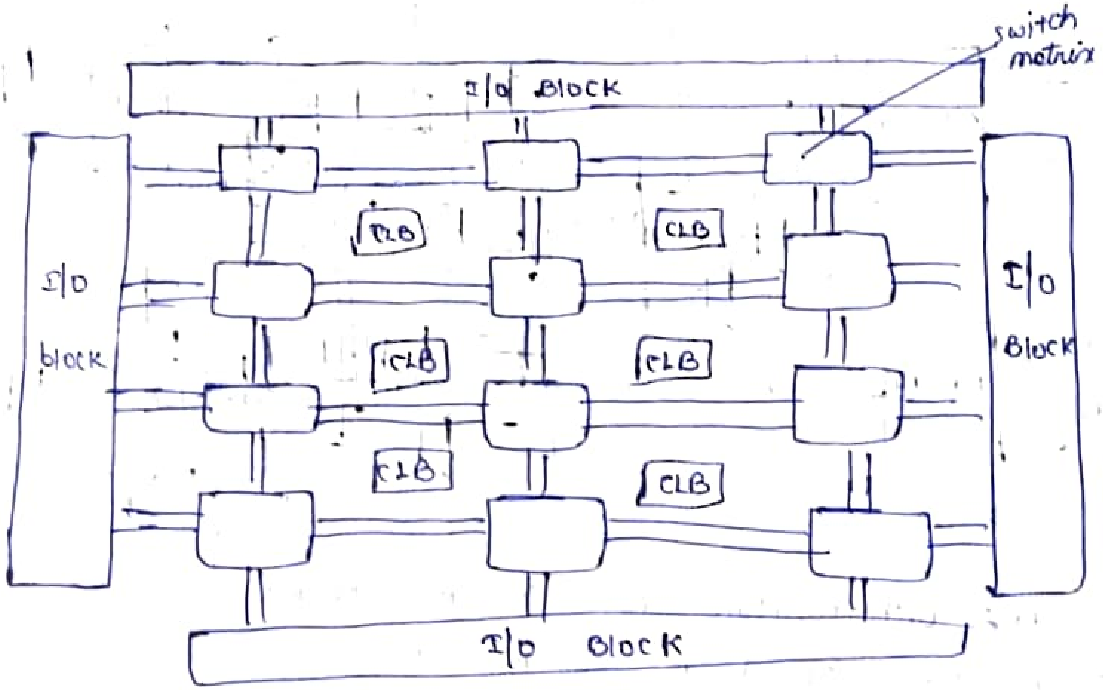
CLR consists

i) Lookup tables
ii) multiplexer
iii) gates
iv) flip flops

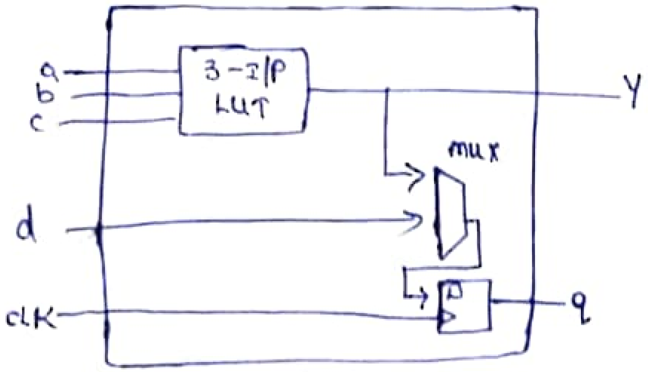
family FPGA

i) xilinix
ii) Actel
iii) Altera

Basic Architecture of xilinx spartan



Basic FPGA Architecture (logic block)



- An essential hazard is cost by a unequal dela[y] along 2 or more path that originate from the same input

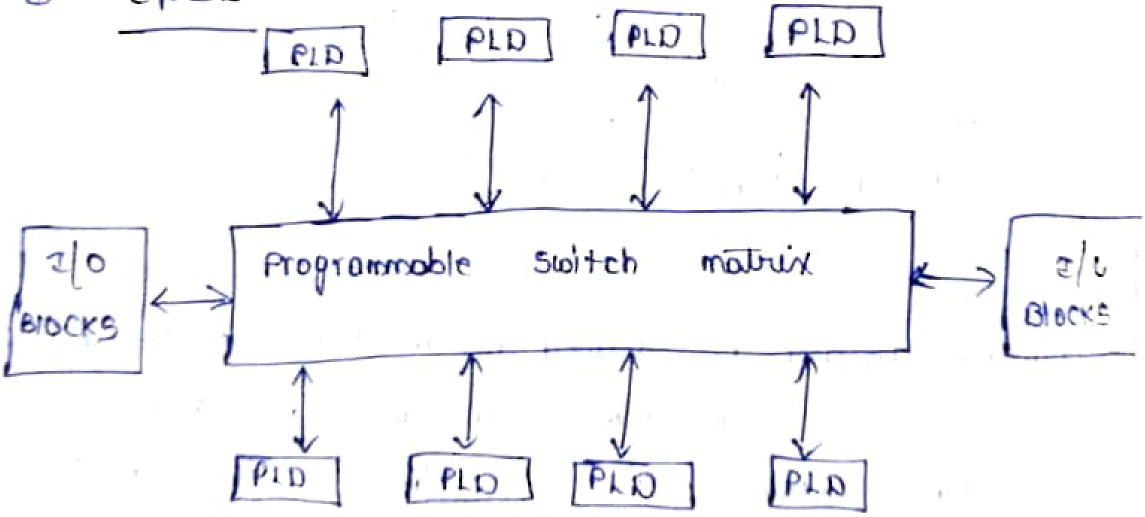
- essential hazard cannot be corrected by adding the redunant terms

- the / problem

- To avoid the essential hazard each f.b ckt must be handled with the Individual ~~do~~ delay with in the feedback path is compared with delays of other signals that originate from the input terminal.

- PAL (AND is programmable & OR is fixed)
- PLA (OR is programmable & AND is fixed)

- CPLD

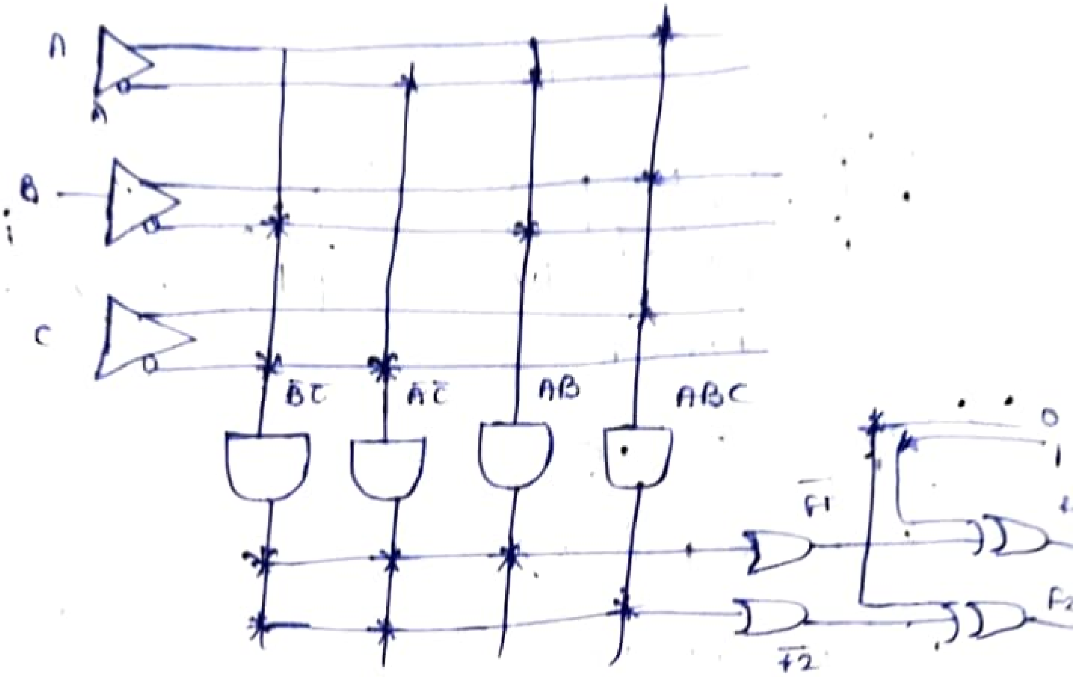


~~Program~~

- PLD can handle 10-20 logic equation

- For more complex circuits complex PLD or CPLD can be used ~~to~~ in which PLD are inter connected.

logic diagram:-



Implement the following Boolean function using PAL $Y(A,B,C,D) = \sum_m (2,3,8,9,10,12,13)$

$Z(A,B,C,D) = \sum_m (1,3,4,6,9,12,14)$

Sol:-

F1

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	1 3	1 2
$\bar{A}B$	4	5	7	6
AB	1 12	1 13	15	14
$A\bar{B}$	1 8	1 9	11	1 10

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

f.2

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	1 1	1 3	2
$\bar{A}B$	1 4	5	7	1 6
AB	1 12	13	15	1 14
$A\bar{B}$	8	1 9	11	10

=> 4 => mirror

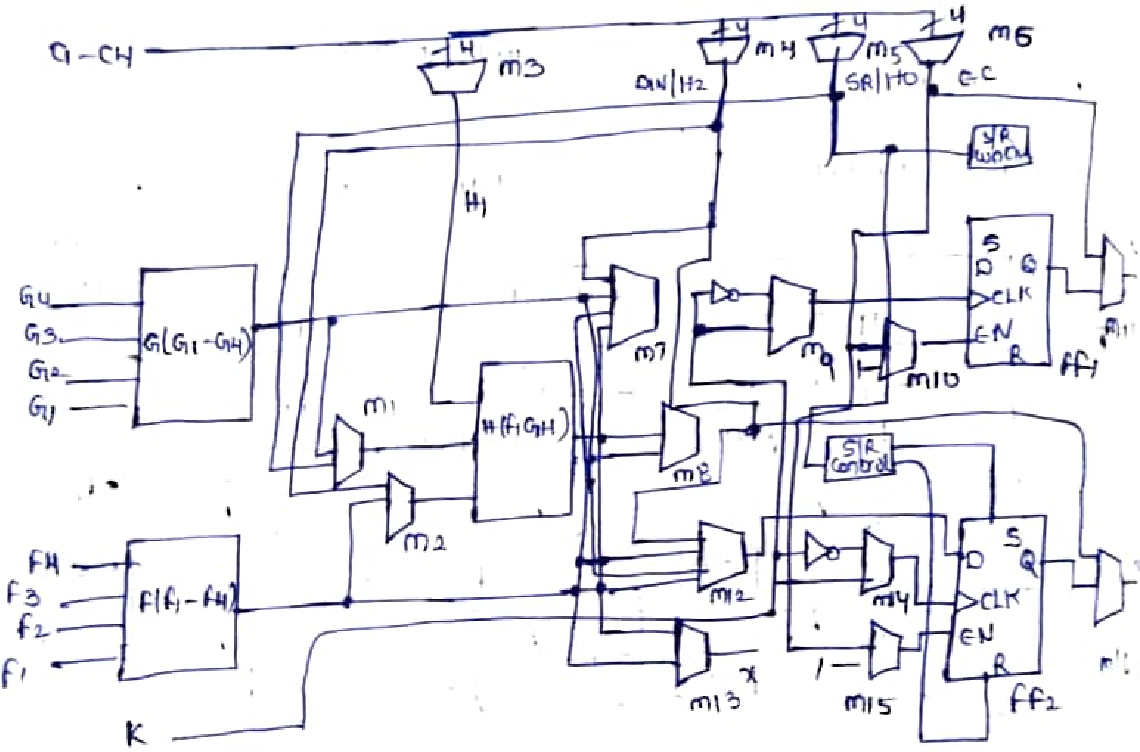
$= B\bar{D} + \bar{B}\bar{C}D + \bar{A}\bar{B}$

⊛ FPGA consists of an array of : 100 or 1000 of logic blocks surrounded by a programmable i/p & o/p blocks connected together. the term of field Programmable programming in the field.

⊛ modified the device function (user friendly)

⊛ In exponsive, easy radiation of logic network in the hard ware

Configurable logic block (CLB)



⊛ This architectuture for XC-4000 (xilinx)

⊛ the clock speed is 80 MHZ (XC-4000) speed.

⊛