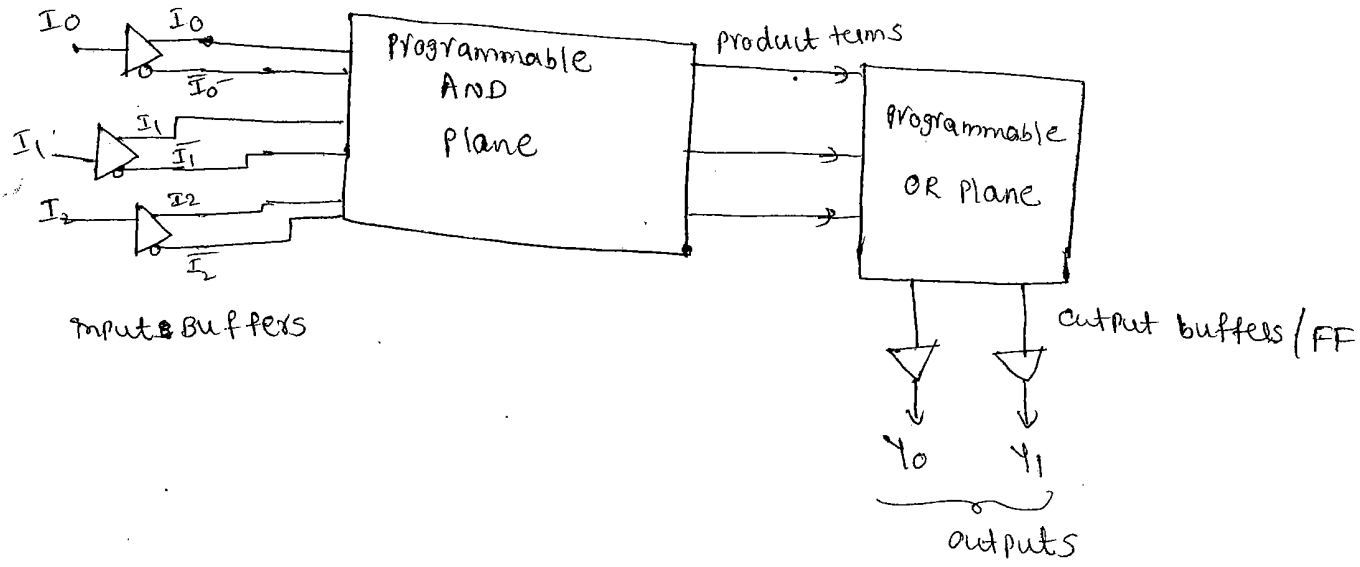
UNIT – VII

Semiconductor Integrated circuit Design

PLAs :– (Programmable Logic Array)

The PLA is one type of Programmable logic device which has a set of Programmable AND planes followed by a set of programmable OR planes and which can then be conditionally complemented to produce an output.

Block diagram of a PLA device:-

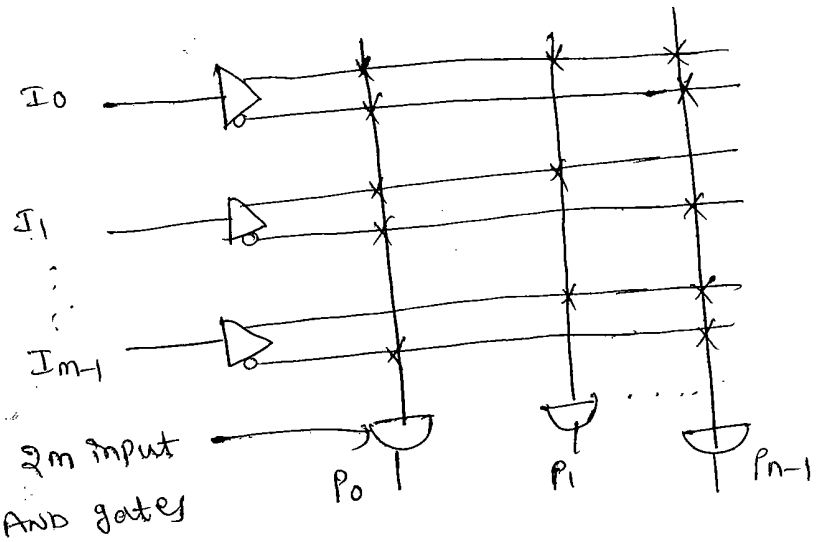


→ In the PLA approach instead of generating all the min terms a seperate logic is implemented which generates only the required product terms.

→ This saves lot of silicon area and also the common product terms are identified and only one product term is

generated for that particular term.

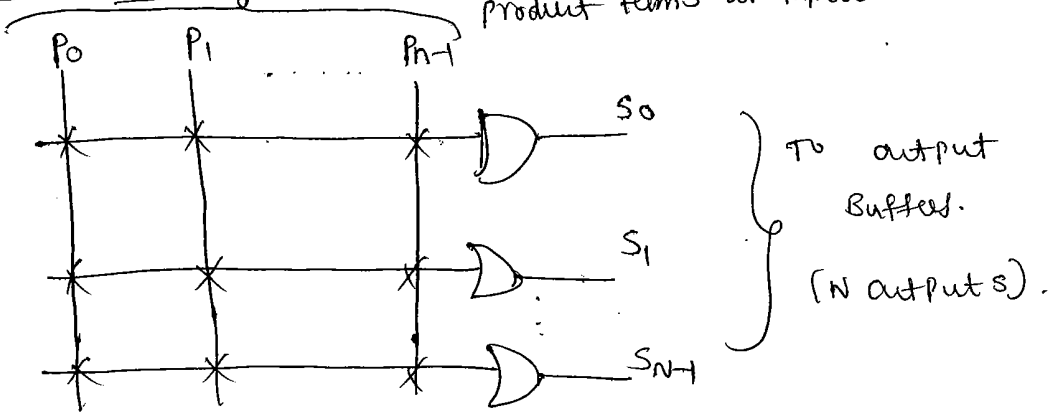
Simplified representation of AND matrix:-



where m → number of inputs.

n → number of product terms.

Simplified representation of OR matrix:-



Application of PLA:-

(1) we can implement both combinational as well as sequential circuits using PLA.

(2) For combinational circuits, the PLA devices with only

→ To implement the sequential circuits we use the PLA device with Flipflop and buffers included in the output stage.

Designing of Combinational circuits using PLA :-

Procedure:

Step1 : Prepare the truth table

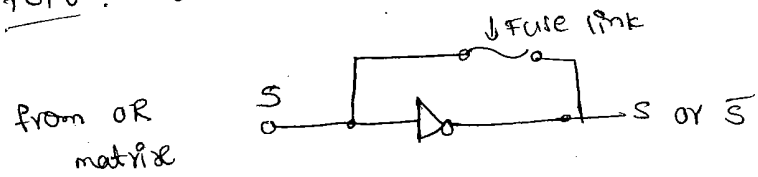
Step2 : write boolean expression in Sop form

Step3 : obtain the minimum SOP form to reduce the number of product terms to minimum.

Step4 : Decide the input connections of AND matrix for generating the required product terms.

Step5 : Then decide the input connections of OR matrix to generate the required sum terms.

Step6 : Decide the connection of invert / non-invert matrix



→ It can invert its input if active low output is required. Its input is passed without any inversion if active high output is required.

→ The output will be same as input if fuse link is closed, where as we get $\bar{S}$ at output if fuse link is open circuit.

Step7 :-

program the PLA .

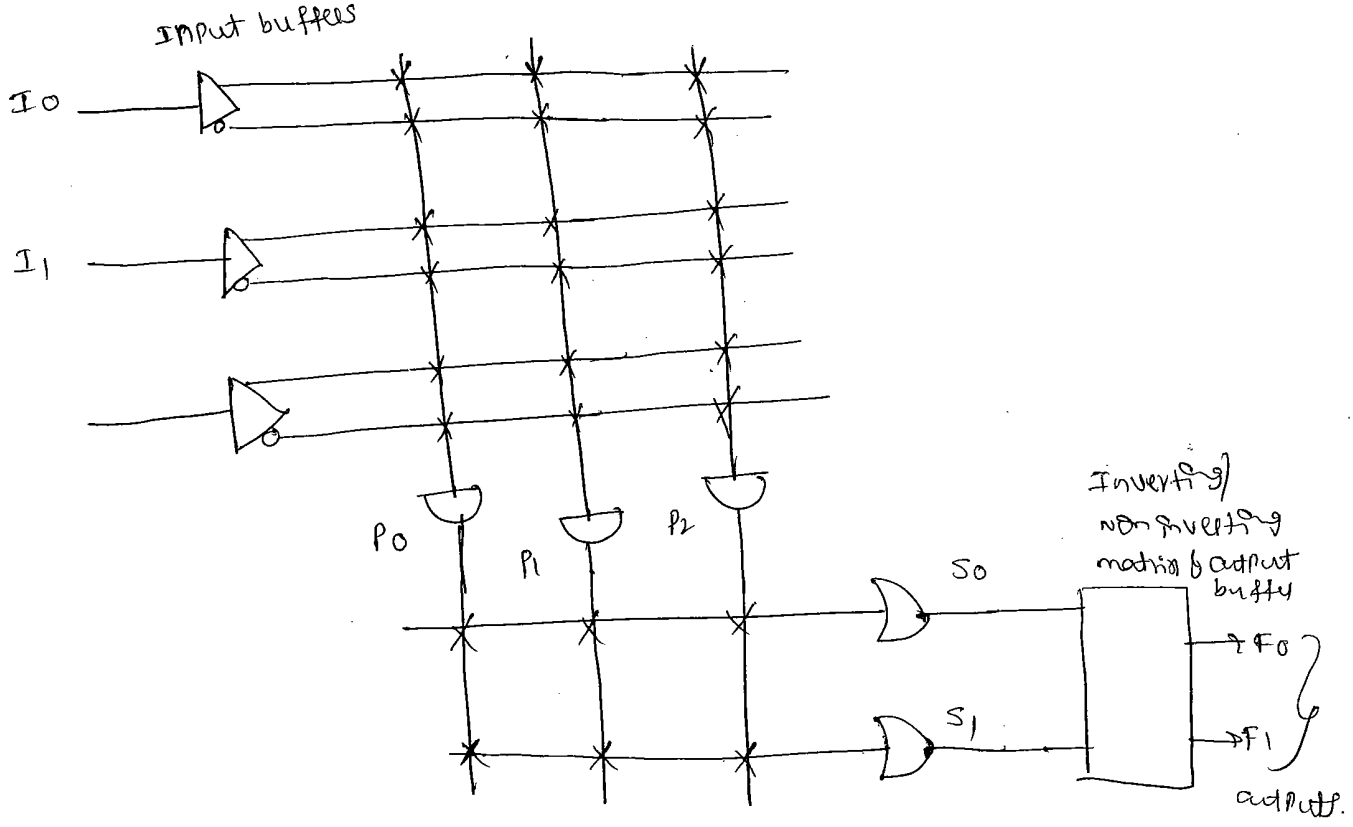
Combinational circuit for a PLA with 3 inputs, three product terms and 2 outputs;

let 3 inputs I_0, I_1, I_2.

number of outputs = number of OR gate = 2.

number of product terms = number of AND gate = 3.

→ Hence this PLA device will have 3 AND gate each of which hav 6 inputs & 2 OR gate.



Specifying size of a PLA

The size of a PLA is specified by m x p x n.

m → no. of inputs

p → No. of product terms

n → no of outputs.

(1) A combinational circuit is defined by the functions

$F_1(A,B,C) = \Sigma m(4,5,7)$

$F_2(A,B,C) = \Sigma m(3,5,7)$.

Implement this circuit with a PLA having 3 inputs, 3 product terms and 2 outputs.

$F_1(A,B,C) = \Sigma m(4,5,7) = A\bar{B}\bar{C} + A\bar{B}C + ABC$

$F_2(A,B,C) = \Sigma m(3,5,7) = \bar{A}BC + A\bar{B}C + ABC$.

minimization:

$F_1(A,B,C) = A\bar{B}\bar{C} + A\bar{B}C + ABC$

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

$= A\bar{B} + ABC$ ($\because C+\bar{C}=1$ in Boolean algebra).

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

~~$F_2 = \bar{A}BC + A\bar{B}C + ABC$~~

$= A[\bar{B}(C+1) + BC]$

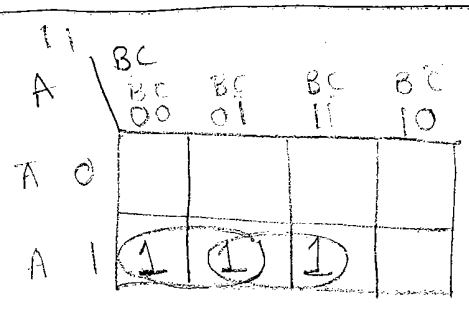
$= A[\bar{B}C + \bar{B} + BC]$

$= A[\bar{B} + C(B+\bar{B})]$

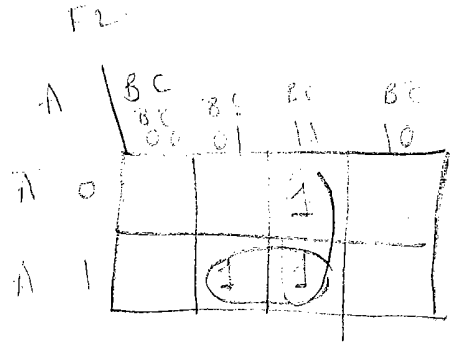
$= A[\bar{B}+C]$

$F_1(A,B,C) = A\bar{B} + AC$ ——— (1)

OR minimization using K-maps:



$F_1 = A\bar{B} + AC$.



$F_2 = BC + AC$.

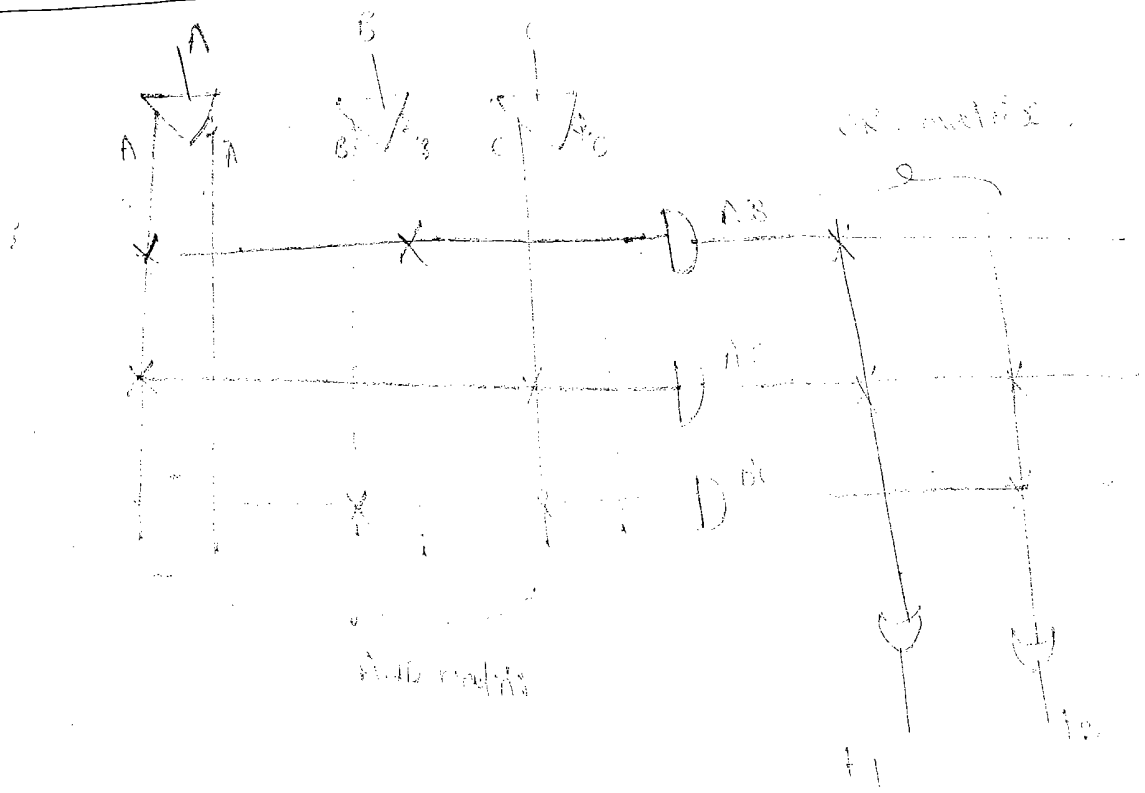
→ Decide inputs to AND matrix.

from expression of F_1 & F_2 we have A, $\bar{B}$, B, C inputs.

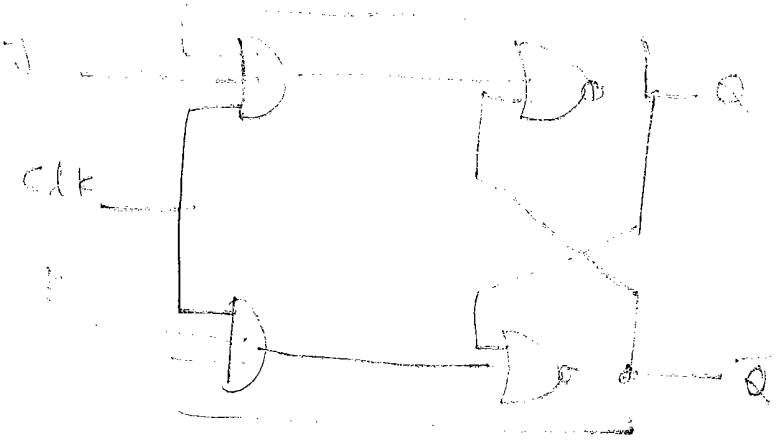
→ Decide inputs to OR matrix.

The inputs to OR matrix are $A\bar{B}$, AC, $\bar{B}C$.

PLA Implementation :—



(2) Implement JK Flip-flop circuit using PLA.



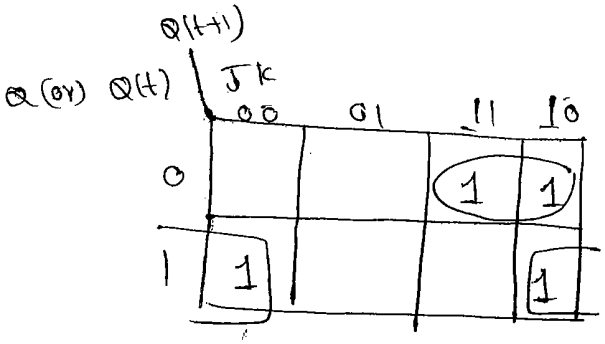
The Characteristic table or truth Table of JK-FF is,

present state Q(t)	J	k	next state Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1 ✓
0	1	1	1 ✓
1	0	0	1 ✓
1	0	1	0
1	1	0	1
1	1	1	0

Q(t) = 0 then Q(t+1) = J

Q(t) = 1 then Q(t+1) = $\overline{k}$

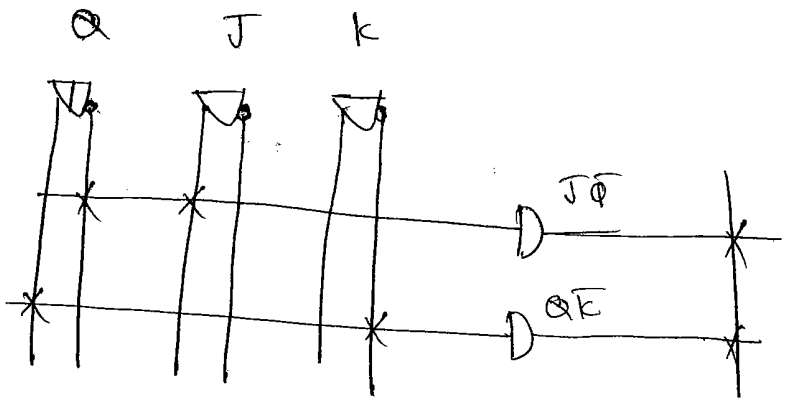
Simplify using k-map;



$$Q(t+1) = \overline{Q}J + Q\overline{k}$$

→ Inputs required for AND gates are, Q, $\overline{Q}$, J, $\overline{k}$

→ 2 AND gates required & one OR gate for 1 output.

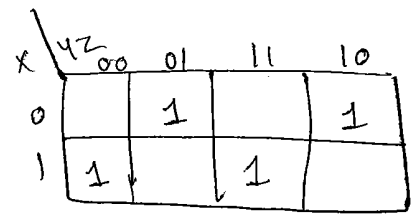


(1) Sketch a diagram for 2 input XOR using PLA.

(2) Implement full adder using PLA.

X	Y	Z	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

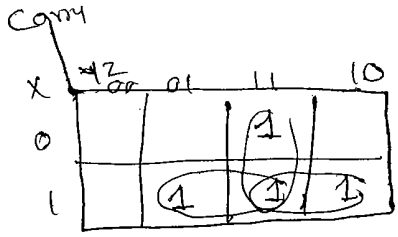
Sum



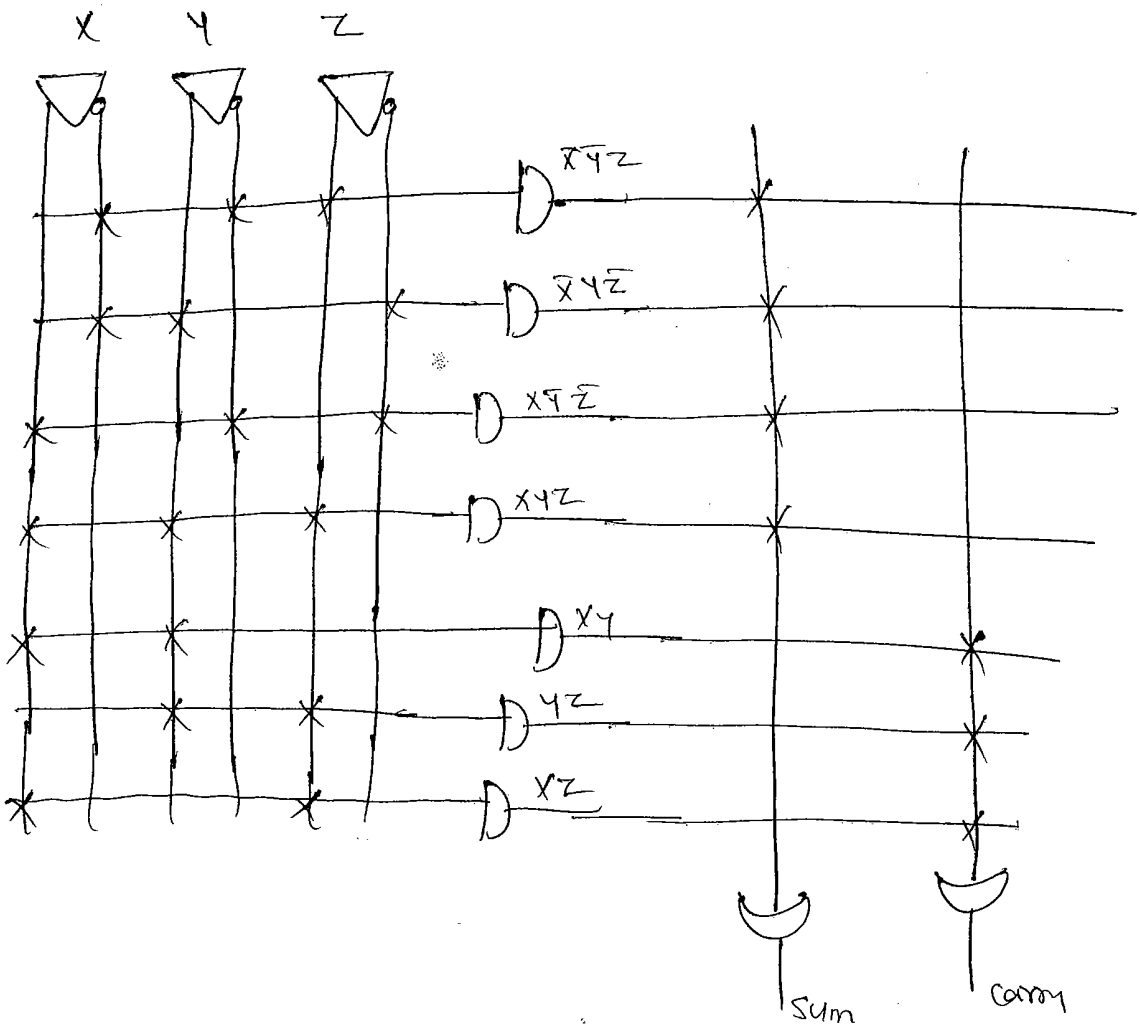
$Sum = \bar{X}\bar{Y}Z + \bar{X}Y\bar{Z} + X\bar{Y}\bar{Z} + XYZ$

Cannot simplified using k-map.

Carry



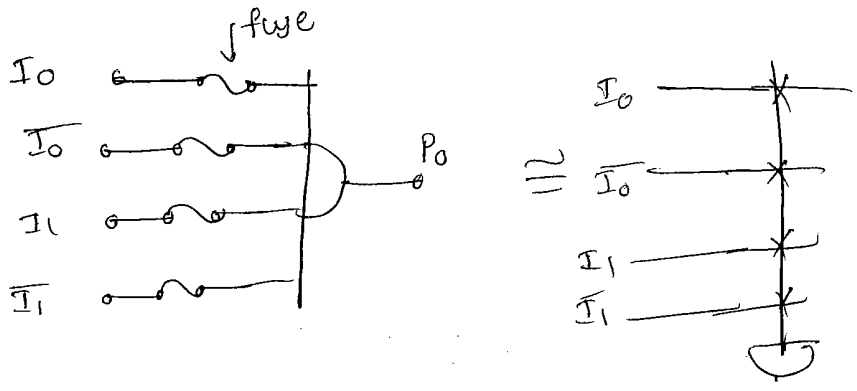
$Carry = XY + XZ + YZ$.



Programmable Array logic (PAL) :—

The PAL is a special type of PLA where the OR array is not programmable.

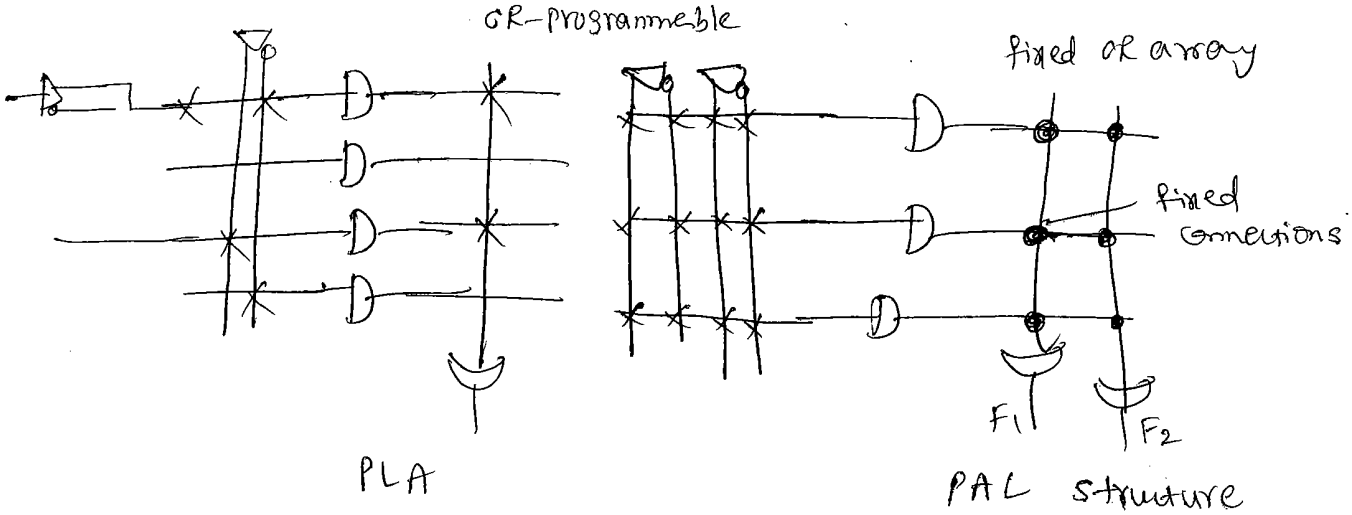
→ means In PAL AND array is programmable but OR array is fixed, whereas in PLA, both arrays are programmable.



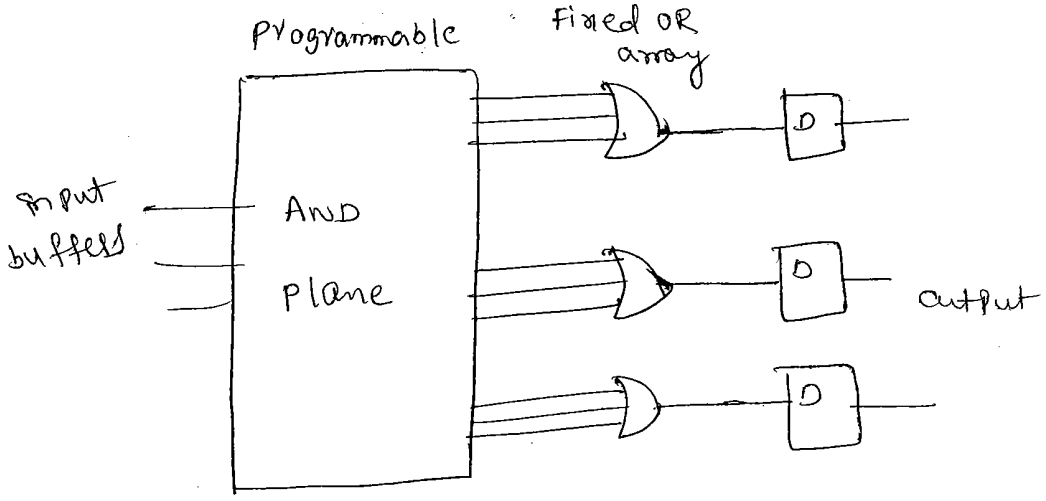
→ The advantage of PALs is the elimination of fuses in the OR array and special electronic circuits to blow these fuses.

→ since these special electronic circuits and programmable OR array occupy a very large area, the area is reduced in PAL.

→ many AND gates in first level and one OR gate of the network output



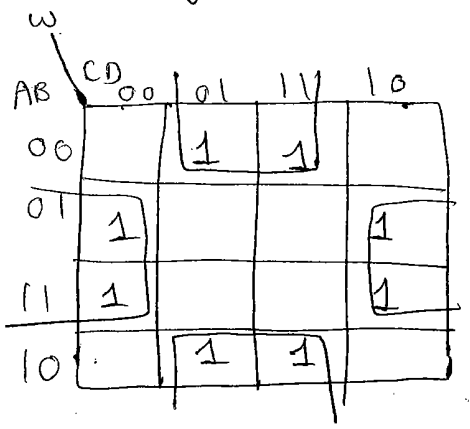
PAL structure :



(1) Implement following Boolean function using PAL.

$$W(A, B, C, D) = \Sigma m(1, 3, 4, 6, 9, 11, 12, 14)$$

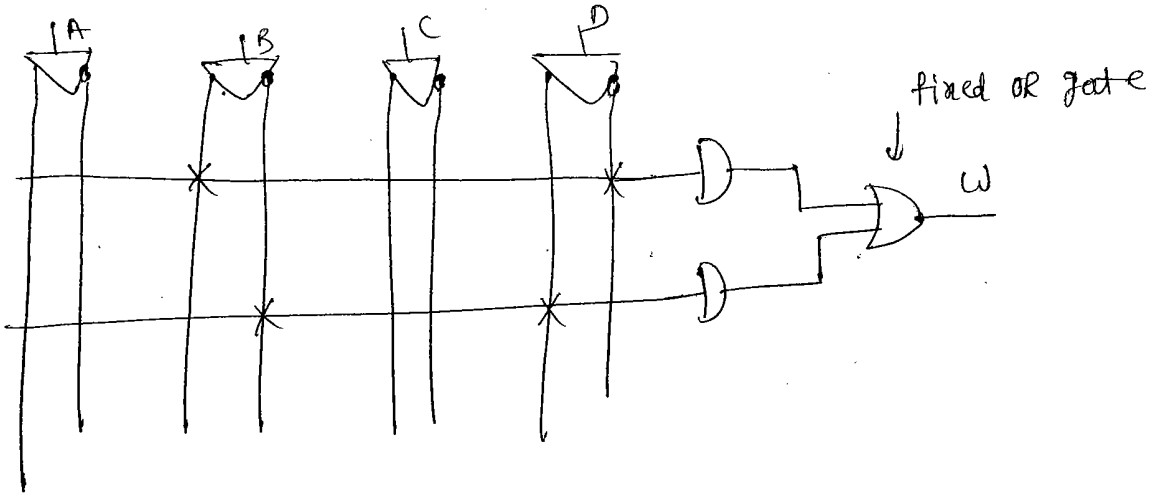
→ Simplify using K-map.



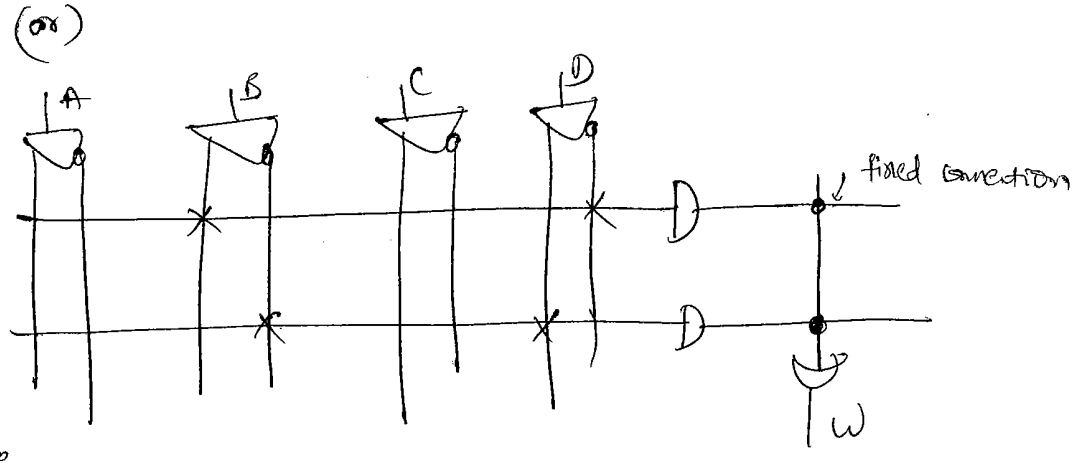
8421

1 → 0001
3 → 0011
4 → 0100
6 → 0110
9 → 1001
11 → 1011
12 → 1100
14 → 1110

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



(or)

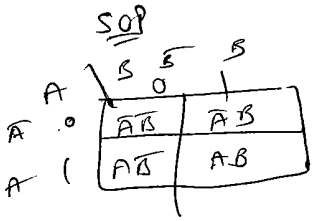


PAL 16L8

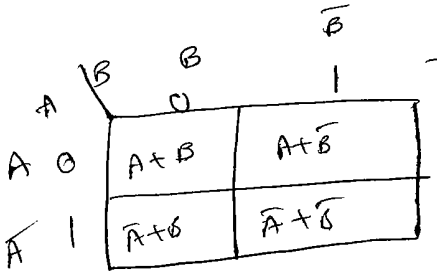
Seven AND gates to each OR gate.

Product of Sum (POS) simplification using k-map:

SOP

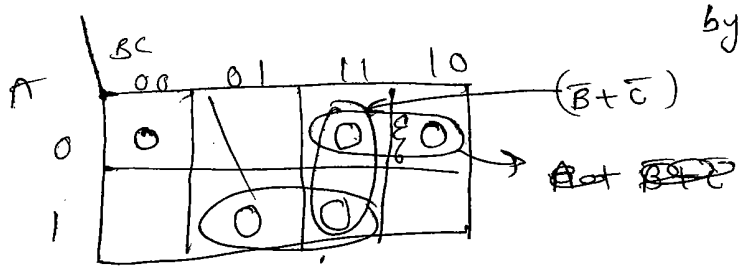


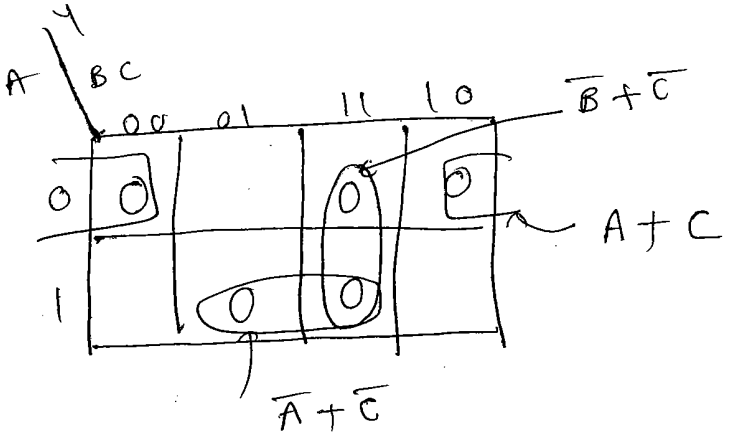
POS



Ex:-

$Y = \pi M(0, 2, 3, 5, 7)$ represents POS terms by '0'.

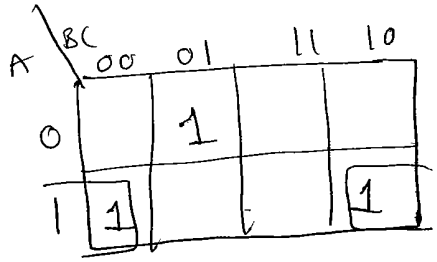




$Y = (\bar{A}+\bar{C}) \cdot (\bar{B}+\bar{C}) \cdot (A+C)$

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

(OR)



1, 4, 6.

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

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

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

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

$$= \bar{A}C\bar{B} + 0 + AB\bar{C} + A\bar{C}$$

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

Binary to Gray encoder :

B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1

$G_3 = B_3$

$G_2 = B_3 \oplus B_2$

$G_1 = B_2 \oplus B_1$

$G_0 = B_1 \oplus B_0$

$G_{13} =$

Standard cells :-

→ Standard cells are pre-defined logic elements used in the circuit.

→ The design methodology that uses standard cells is known as cell-based design methodology.

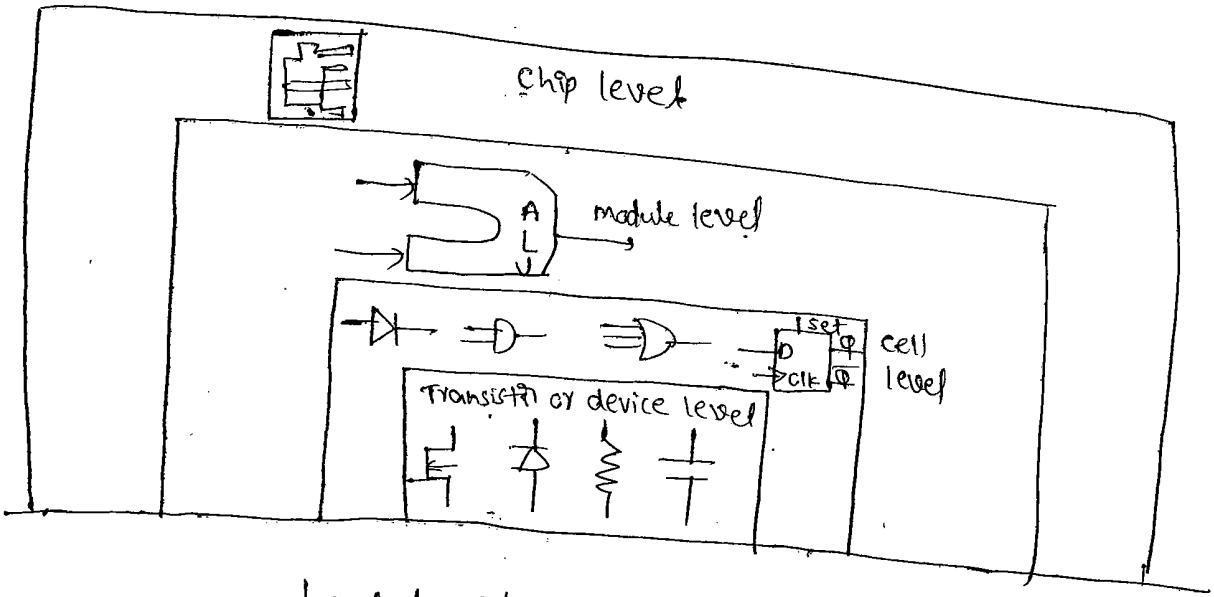
→ Hence, standard cells are the basic building blocks of cell-based IC design methodology.

→ A standard-cell library is one of the foundations upon which the VLSI design approach is built.

→ A standard cell is designed either to store information or perform a specific logic function (Inverter, AND, OR etc).

→ The type of standard cell created to store data is referred to as a sequential cell. (FF & latch).

→ Standard cells are built on transistors. They are one abstraction level higher than Transistor.



Level of abstraction.

→ Hardware block can be represented in four different abstraction levels during the chip implementation process.

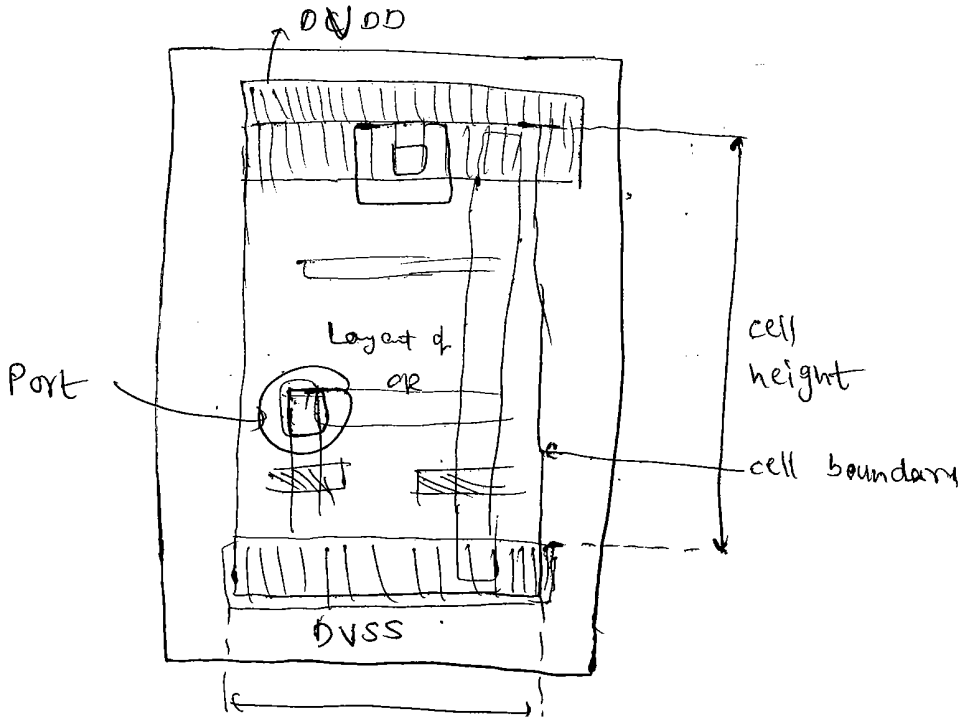
Lowest level :- The lowest level is the transistor or device level.

→ At this level, entire block is described directly by very basic building elements of transistors, diodes, capacitors and resistors.

cell level :- In this, designs are composed of standard cells.

module level :- Designs are represented by modules such as adders, multiplier, ALU & shifter.

Chip level :- The highest level is chip level, At this level, designs are partitioned into subsystems, such as DSP, micro controller, MPEG decoder, UART, USB, DMA, ADC, DAC & PLL.



→ During chip construction process, the designer's HDL code is transformed to a netlist using synthesis tool. The netlist is composed of a certain number of standard cells, each one having its specific logic function.

→ The intended system functions, described by HDL are realized by standard cells in this netlist.

→ These standard cells are placed within the chip's floorplan by special place and route tool.

Cell's physical size defined by

1) cell height
2) cell width.

→ cell boundary attribute used by place and route tool to place cells during placement stage.

→ Only metal1 layer is used inside the cell layout since higher level metals are reserved for signal routing.

→ It is important to complete each standard cell's layout with least amount of silicon area.

→ Physically, the standard cells within an ASIC library have a fixed size in one dimension (usually height) so that they can be placed and aligned along rows of the chip.

⇒ The other name of cell-based design methodology is cell-based ASIC or CBIC in simple.

Factors Influencing Low Power VLSI Design:

1. Reduce V_{DD}.
2. Power supply Reduction
3. Variation of the threshold voltage
4. Optimal Power Voltage
5. Compensating for lower speed.
6. Voltage switch
7. Reduce C
8. Partition Blocks
9. Locality of Reference
10. Clocks and control.
11. Logic design.
12. Buffer design
13. Reduce A
14. Glitch Avoidance
15. Point-to-Point Buses.

Reduce V_{DD}:-

→ Development in fabrication are already moving from the existing standard 5V towards a new level of 3.3V and experimental processes are looking at even lower voltages.

Power Supply Reduction :—

→ One of the motivation in technology development has been to increase the levels of Integration by Reducing feature sizes.

→ However as gate lengths are reduced the electric field strength increases in the gate region. This leads to realiability problems as the high electric field strengths accelerate the Conducting Es to such speeds cause substrate current and then cause latch-up problem.

→ There are 3 approaches to enabling further feature size reduction

(1) lightly doped drain (LDD) technique allows the smallest gate length.

(2) new circuit techniques which avoids high electric fields across individual transistors.

(3) Reduce the supply voltage.

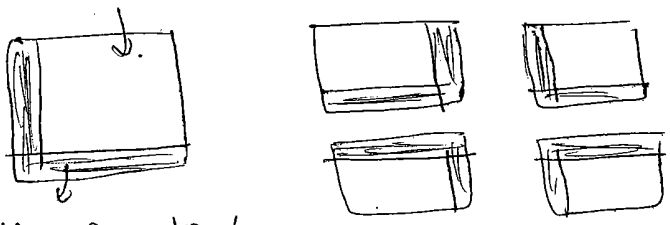
$$\text{Power delay} = \frac{P}{f_{max}} = V_{DD}^2 \, \Sigma (C_i P(r_i))$$

→ The variation in V_{DD} actually leads to a quadratic change in the power-delay product.

f_{max} → max possible frequency of a circuit represents the fastest throughput.

Partition Blocks

In general it is best to partition large blocks into smaller ones. But, the power calculation for each memory access is based upon the capacitances of the bit and word lines which run vertically & horizontally across array.



address generation & bit detectors circuit.

→ Instead of array is broken down into 4 sub-circuits (each with its own support circuitry) and only one unit is addressed with each access, then the product of activity & capacitance is reduced by a half.

Reduce C:

The best strategy is to Reduce capacitance.

Buffer design:-

One problem is the design of circuitry to drive large capacitance. The basic solution is a sequence of buffers with increasing gate widths;

Reduce A:

Reduce A, the average activity on each gate. Power is only expended when a node is switched, If switching is restricted to when Information Changes then Power is minimised.

Glitch Avoidance:-

With some digital logic, there are spurious transitions known as glitches which occurs due to partially resolved functions.