UNIT III - VLSI CIRCUIT DESIGN PROCESSES

1. Explain VLSI Design flow.

Answer:

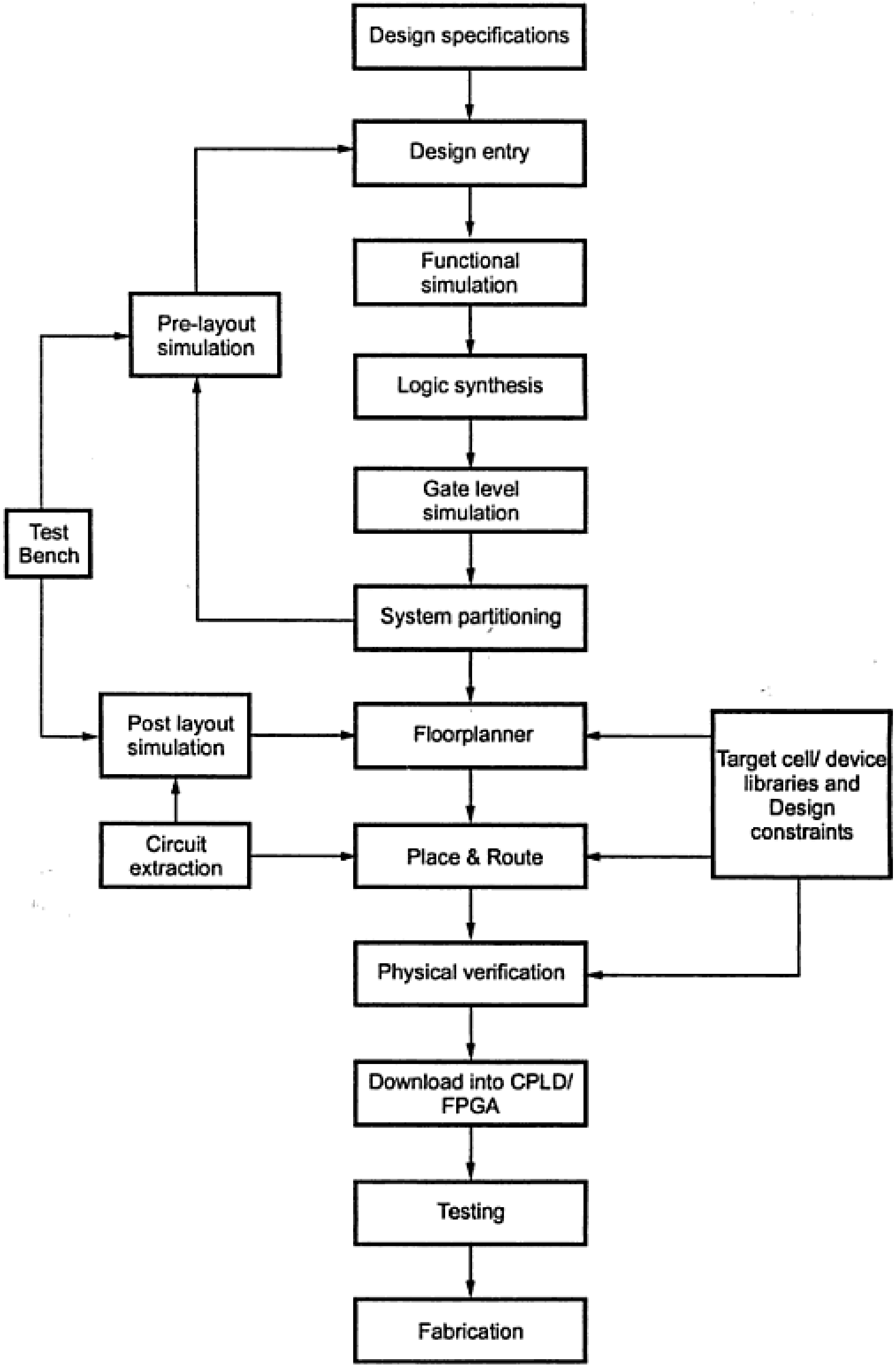


Fig. 3.1 VLSI design flow

i) Design Specifications

- Specification of a design is as a guide to choose the right technology and for knowing the needs of the vendor. Specifications allows each engineer to understand the entire design. It helps the engineer for designing correct interface with rest of the circuit or system. It reduces time required for design and also misassumptions if any.

Any specification includes following information :

1. A block diagram providing details how designed chip fit into the entire system.
2. Internal block diagram for every subsection and its function.
3. Input threshold levels of all input pins and driving capability of output pins.
4. Timing specifications like setup and hold times, propagation delays and clock-cycle time.
5. Package type required.
6. Total gate count of the system under design.
7. Total power consumption of the circuit.
8. Test procedures for different tests.
9. Total cost of the target design chip.

ii) Design Entry

- User can enter a design with a schematic editor or any other text-based software tool, either a hardware description language (VHDL or VERILOG).

Schematic Entry

- It provides a graphical interface for design entry. A design can be build by a user with individual gates or he can combine gates to create functional blocks.

HDL Entry

- This entry supports mixed level description where gate and netlist constructs both are used along with functional descriptions.

iii) Functional Simulation

- It is the process where logic in the design is checked before user implements it in a device. As the timing information is not available at this early stage of design flow, functional simulator tests the logic of design using unit delays.

iv) Logic Synthesis

- Here logic synthesis tool is used which produces Netlist (textual information) from synthesis process. Logic cells and their interconnections are described in detail in the Netlist. Netlist is an EDIF (Electronic Data Interchange Format) file. Thus during synthesis behavioural information in the HDL file is translated into a structural netlist.

v) System Partitioning

- System partitioning is the process of dividing a large and complex system into smaller modules.

vi) Prelayout Simulation

- This is required for verification of a circuit design through software programs. Here stimuli is applied to design over a specific time period and recording, analyzing the respective response from the model.

vii) Floorplanner

- The main function of floorplanner is to estimate the required chip area that will be used for each standard cell or module of the design. It is responsible for performance improvement of the design. Floorplanner is a tool that lets user generate and edit hierarchical floorplans.

viii) Place and Route

- After design mapping, flow engine places and routes the design. All logic blocks, including the Configurable Logic Blocks (CLB) and Input-Output Blocks (IOB) are assigned specific locations on the die at place stage.

- In the route stage, the logic blocks are assigned, particular interconnect elements on die.

ix) Circuit Extraction

- This process determines the resistances and capacitances of all the interconnections.

x) Post layout simulation

- After physical place and route, this simulation is carried out. While carrying out this simulation propagation delays of logic cells and interconnection delays of interconnect are taken into account. If post layout simulation results ful-fill the design specifications, designer can proceed for chip finishing part.

xi) Physical Verification

- After placement and routing and full custom editing physical verification is carried out. It is the process of interpreting the physical layout data to determine whether it conforms to the electrical design rules, physical design rules and source schematic. Design Rule Check (DRC), Electrical Rule Check (ERC), Antenna check and short circuit check are the processes which comes under physical verification.

xii) Testing

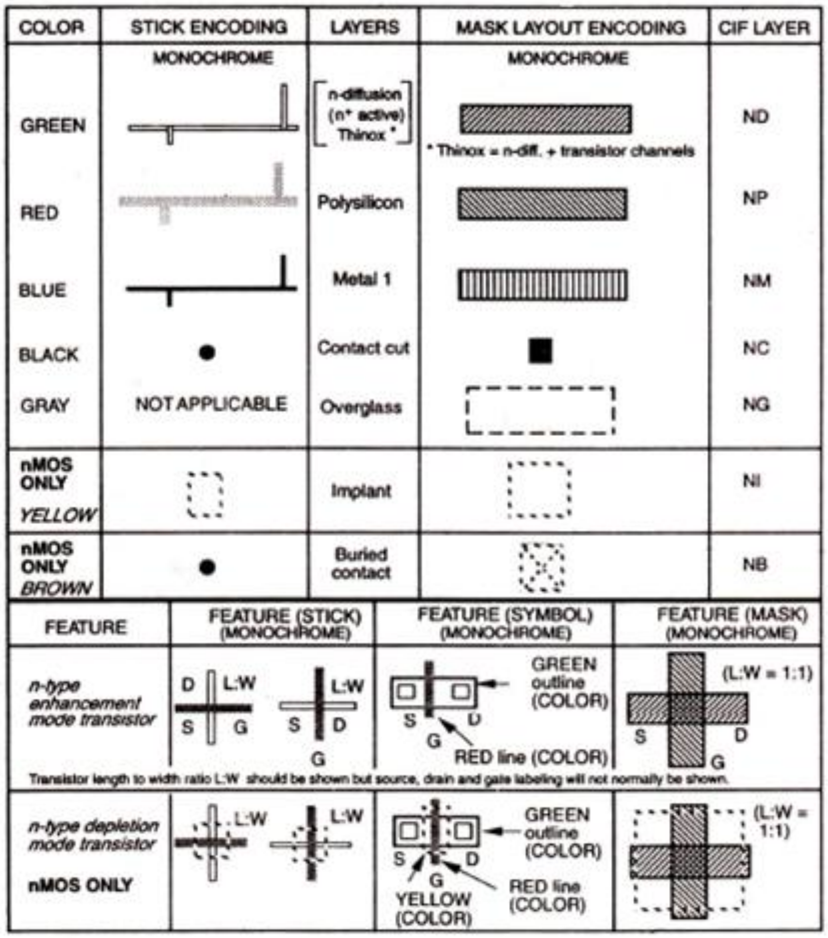
- During production of chips, it is necessary to have some sort of built-in tests for designed system which continuously tests the system over long period of time. Chip will fail because of some electrical or mechanical problems that will usually show up with such testing procedure.

xiii) Chip Fabrication

- Before submitting design for fabrication, input-output pads should be included in the design and it's connectivity should be verified.
- Then appropriate package selection for the design and selecting bonding plan for the package is required. Details of how each pad of design is connected to each pin of package is required.

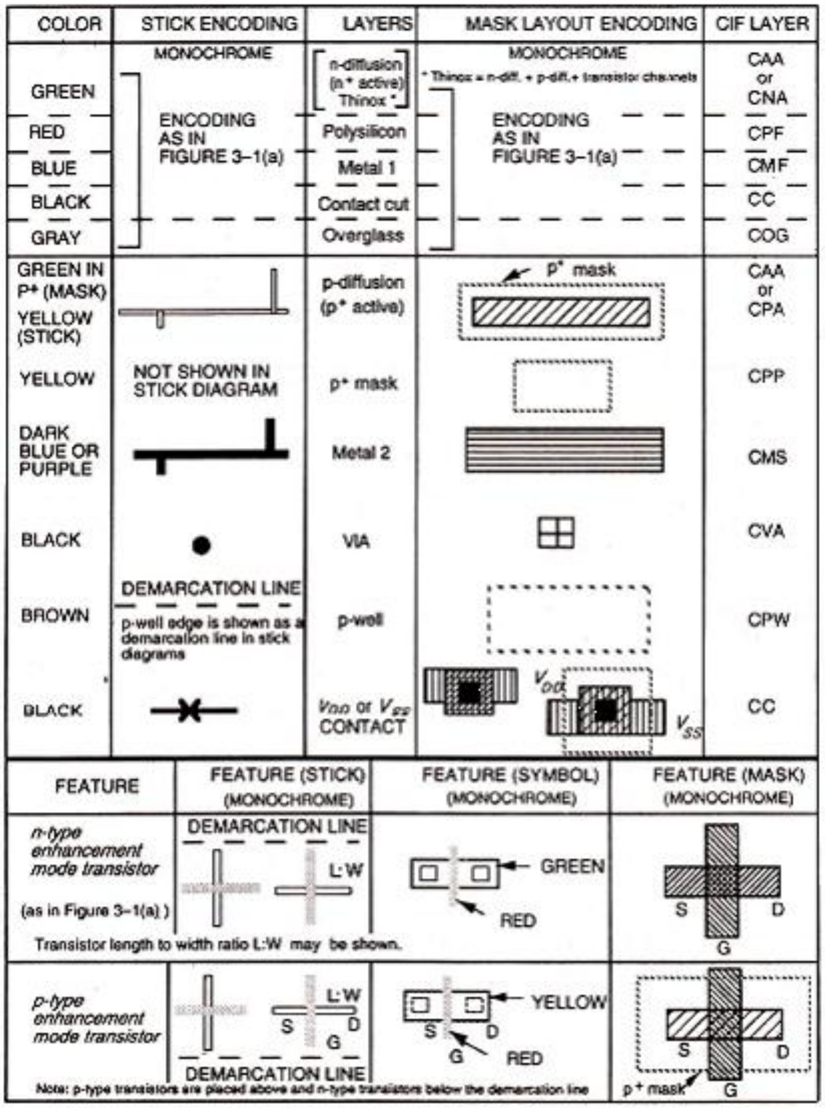
2. Draw the monochrome stick diagram encodings for nMOS and CMOS process.

Answer:



COLOR	STICK ENCODING	LAYERS	MASK LAYOUT ENCODING	CIF LAYER
	MONOCHROME		MONOCHROME	
GREEN		[n-diffusion (n⁺ active) Thinox *]	▨ * Thinox = n-diff. + transistor channels	ND
RED		Polysilicon	▨	NP
BLUE		Metal 1	▥	NM
BLACK	●	Contact cut	■	NC
GRAY	NOT APPLICABLE	Overglass		NG
nMOS ONLY *YELLOW*		Implant		NI
nMOS ONLY *BROWN*	●	Buried contact		NB

FEATURE	FEATURE (STICK) (MONOCHROME)	FEATURE (SYMBOL) (MONOCHROME)	FEATURE (MASK) (MONOCHROME)
n-type enhancement mode transistor	D L:W S G L:W S D G	GREEN outline (COLOR) S D G RED line (COLOR)	(L:W = 1:1) S D G
Transistor length to width ratio L:W should be shown but source, drain and gate labeling will not normally be shown.			
n-type depletion mode transistor nMOS ONLY	L:W L:W	GREEN outline (COLOR) S D G YELLOW (COLOR) RED line (COLOR)	(L:W = 1:1)



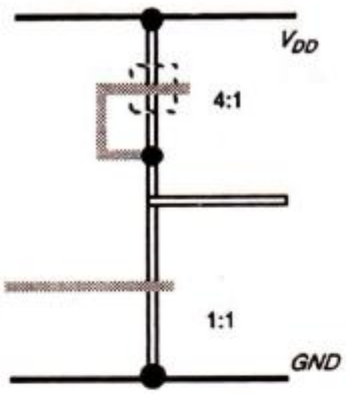
COLOR	STICK ENCODING	LAYERS	MASK LAYOUT ENCODING	CIF LAYER
GREEN	MONOCHROME ENCODING AS IN FIGURE 3–1(a)	n-diffusion (n⁺ active) Thinox *	MONOCHROME * Thinox = n-diff. + p-diff.+ transistor channels ENCODING AS IN FIGURE 3–1(a)	CAA or CNA
RED		Polysilicon		CPF
BLUE		Metal 1		CMF
BLACK		Contact cut		CC
GRAY		Overglass		COG
GREEN IN P+ (MASK) YELLOW (STICK)		p-diffusion (p⁺ active)	p⁺ mask	CAA or CPA
YELLOW	NOT SHOWN IN STICK DIAGRAM	p⁺ mask		CPP
DARK BLUE OR PURPLE		Metal 2		CMS
BLACK	●	VIA	⊞	CVA
BROWN	DEMARCATION LINE p-well edge is shown as a demarcation line in stick diagrams	p-well		CPW
BLACK		V_{DD} or V_{SS} CONTACT	V_{DD} V_{SS}	CC

FEATURE	FEATURE (STICK) (MONOCHROME)	FEATURE (SYMBOL) (MONOCHROME)	FEATURE (MASK) (MONOCHROME)
n-type enhancement mode transistor (as in Figure 3–1(a))	DEMARCATION LINE L:W	GREEN RED	S D G
Transistor length to width ratio L:W may be shown.			
p-type enhancement mode transistor	L:W S D G DEMARCATION LINE	YELLOW S D G RED	S D p⁺ mask G
Note: p-type transistors are placed above and n-type transistors below the demarcation line			

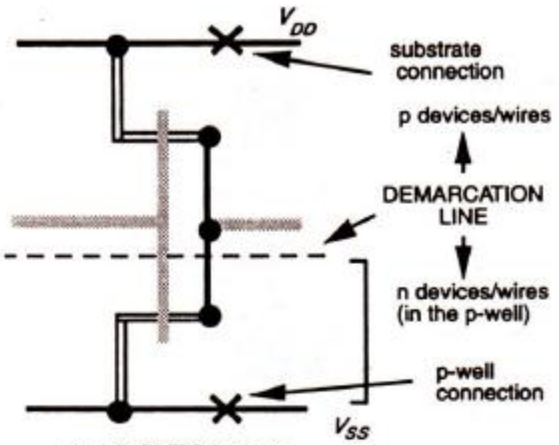
Encodings for a double metal CMOS p-well process

3. Draw the stick diagram for nMOS and CMOS inverter.

Answer:



4:1 nMOS inverter



p-well CMOS inverter

4. Explain about design rules for layout.

Answer:

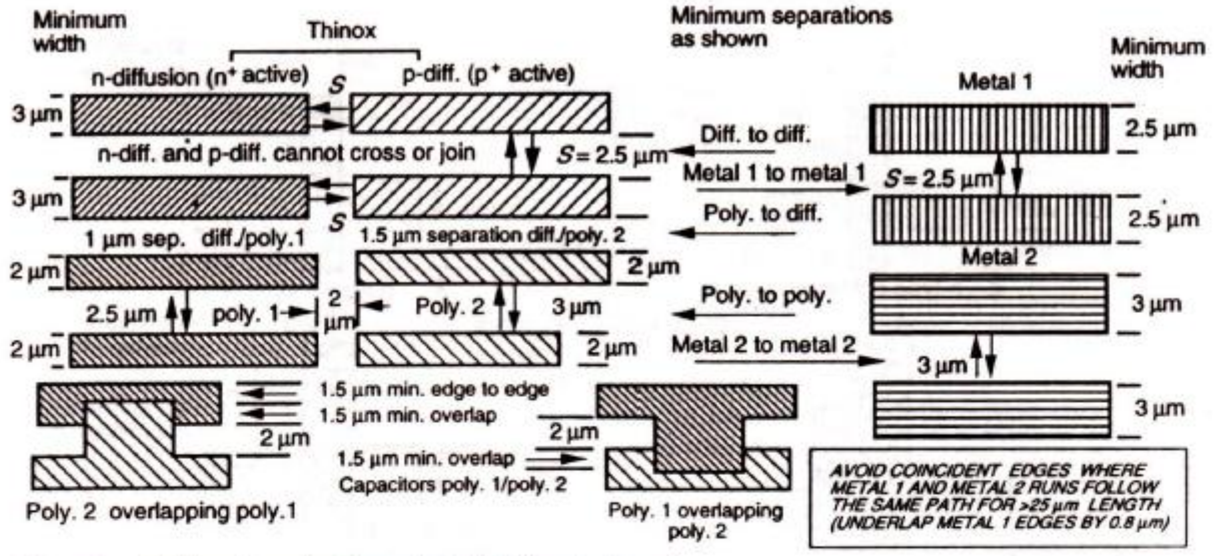
The object of a set of design rules is to allow a ready translation of circuit design concepts, usually in stick diagram or symbolic form, into actual geometry in silicon. The design rules are the effective interface between the circuit/system designer and the fabrication engineer. Clearly, both sides of the interface have a vested interest in making their own particular tasks as easy as possible and design rules usually attempt to provide a workable and reliable compromise that is friendly to both sides.

Circuit designers in general want tighter, smaller layouts for improved performance and decreased silicon area. On the other hand, the process engineer wants design rules that result in a *controllable and reproducible* process. Generally we find that there has to be a compromise for a competitive circuit to be produced at a reasonable cost.

One of the important factors associated with design rules is the achievable definition of the process line. Definition is determined by process line equipment and process design. For example, it is found that if a 10:1 wafer stepper is used instead of a 1:1 projection mask aligner, the level-to-level registration will be closer. Design rules can be affected by the maturity of the process line. For example, if the process is mature, then one can be assured of the process line capability, allowing tighter designs with fewer constraints on the designer.

5. Draw the 2µm design rules for wires, transistors and contact cuts.

Answer:



Otherwise polysilicon 2 must not be coincident with polysilicon 1

Note: Where no separation is specified, wires may overlap or cross (e.g. metal may cross any layer). For p-well CMOS, n-diff. wires can only exist inside and p-diff. wires outside the p-well. For n-well CMOS, p-diff. wires can only exist inside and n-diff. wires outside the n-well.

FIGURE 3.13(a) Design rules for wires (interconnects) (Orbit 2 µm CMOS).

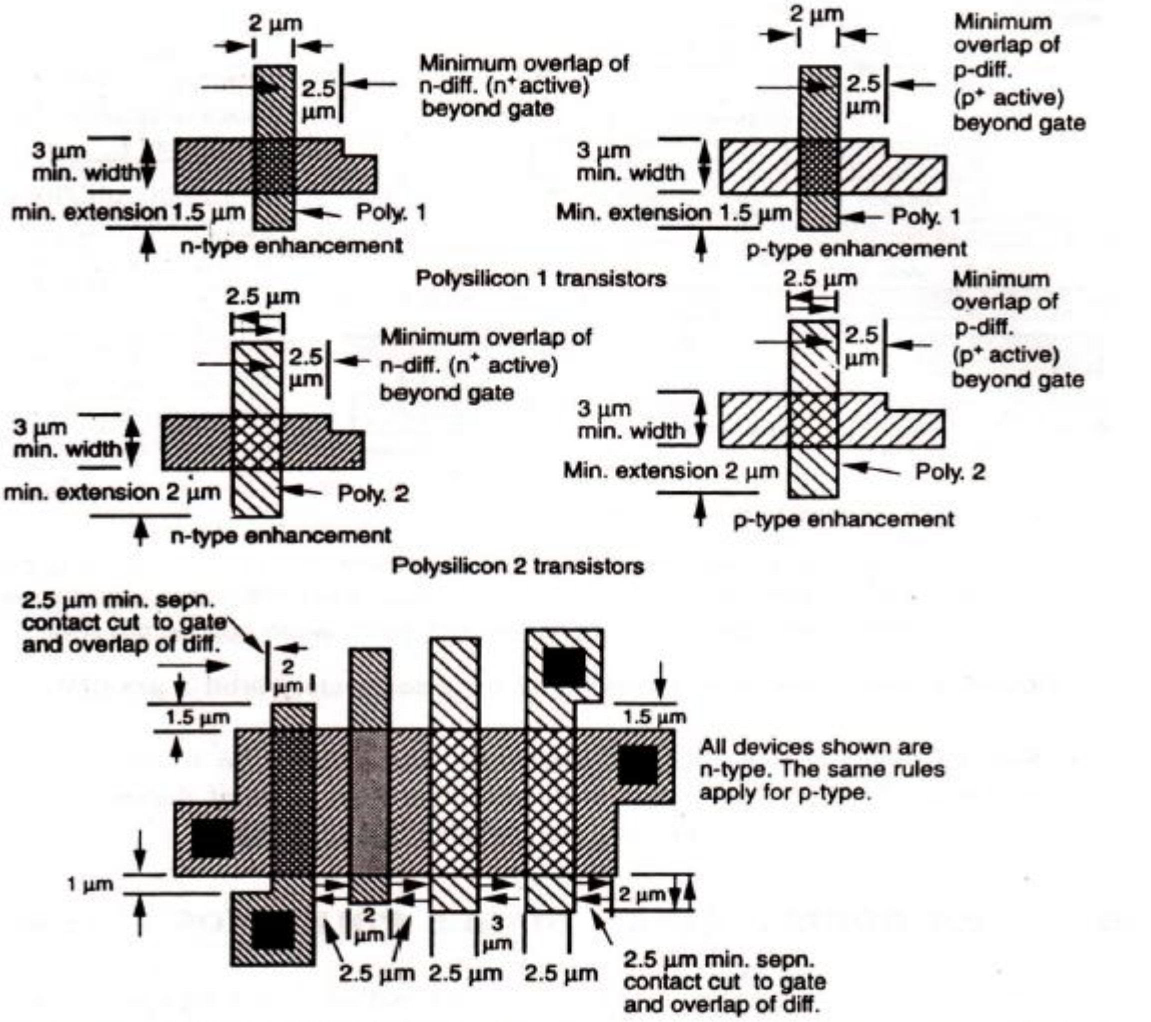

2 μm
Minimum overlap of n-diff. (n⁺ active) beyond gate
2.5 μm
3 μm min. width
min. extension 1.5 μm
Poly. 1
n-type enhancement
2 μm
Minimum overlap of p-diff. (p⁺ active) beyond gate
2.5 μm
3 μm min. width
Min. extension 1.5 μm
Poly. 1
p-type enhancement
Polysilicon 1 transistors
2.5 μm
Minimum overlap of n-diff. (n⁺ active) beyond gate
2.5 μm
3 μm min. width
min. extension 2 μm
Poly. 2
n-type enhancement
2.5 μm
Minimum overlap of p-diff. (p⁺ active) beyond gate
2.5 μm
3 μm min. width
Min. extension 2 μm
Poly. 2
p-type enhancement
Polysilicon 2 transistors
2.5 μm min. sepn. contact cut to gate and overlap of diff.
2 μm
1.5 μm
1.5 μm
All devices shown are n-type. The same rules apply for p-type.
1 μm
2 μm
2 μm
3 μm
2.5 μm 2.5 μm 2.5 μm
2.5 μm min. sepn. contact cut to gate and overlap of diff.

FIGURE 3.13(b) **Transistor related design rules (Orbit 2 μm CMOS) minimum sizes and overlaps.**

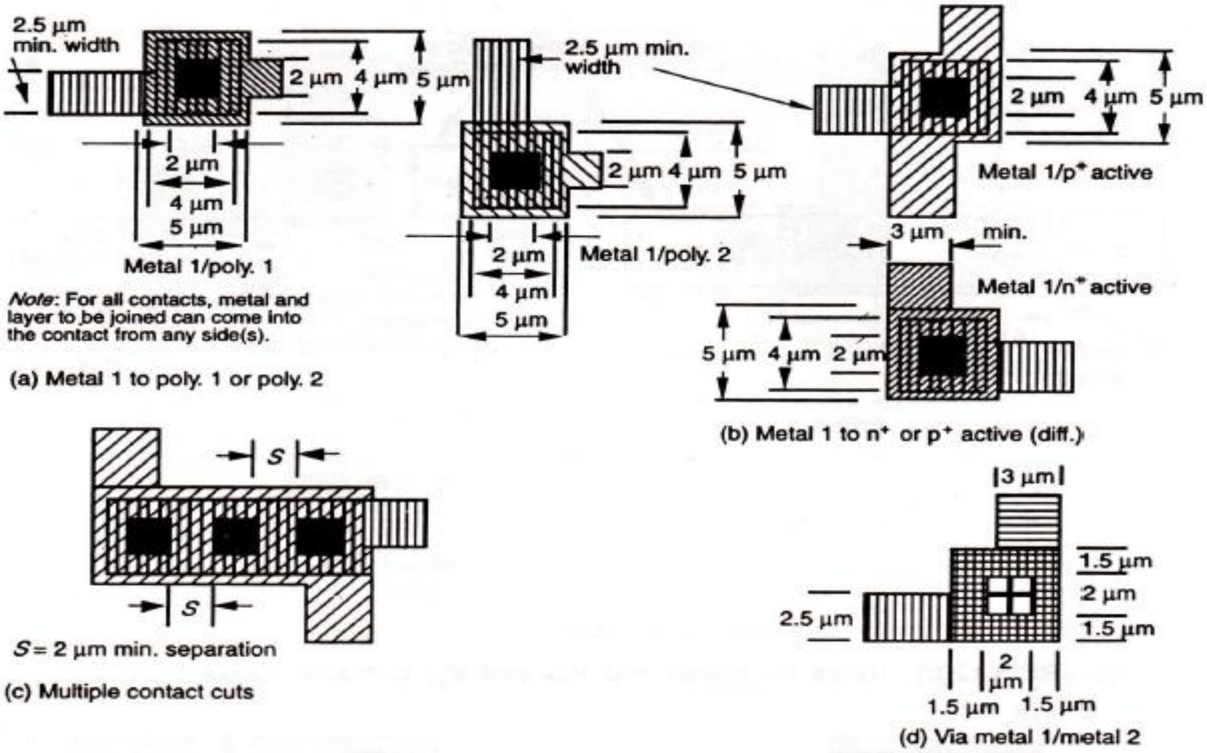


FIGURE 3.13(c) **Rules for contacts and vias (Orbit 2 μm CMOS).**

6. Draw the layout diagram for two input nMOS NOR gate.

Answer:

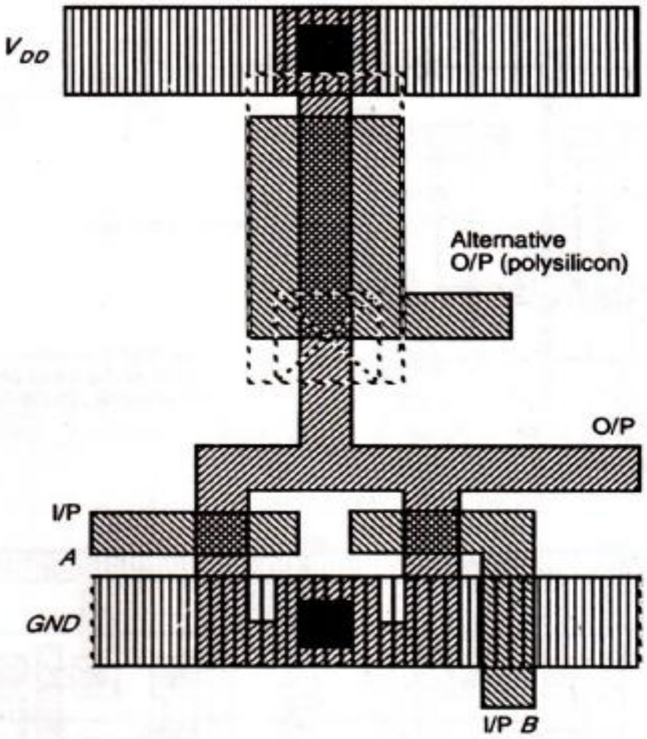


FIGURE 3.16 Two I/P nMOS *Nor* gate.

7. Explain about scaling of MOS circuits.

Answer:

VLSI fabrication technology is still in the process of evolution which is leading to smaller line widths and feature size and to higher packing density of circuitry on a chip.

The scaling down of feature size generally leads to improved performance and it is important therefore to understand the effects of scaling. There are also future limits to scaling down which may well be reached in the next decade.

Microelectronic technology may be characterized in terms of several indicators, or figures of merit. Commonly, the following are used:

- Minimum feature size
- Number of gates on one chip
- Power dissipation
- Maximum operational frequency
- Die size
- Production cost.

Many of these figures of merit can be improved by shrinking the dimensions of transistors, interconnections and the separation between features, and by adjusting the doping levels and supply voltages. Accordingly, over the past decade, much effort has been directed toward the upgrading of process technology and the resultant scaling down of devices and feature size.

The most commonly used models are the constant electric field scaling model and the constant voltage scaling model. They both present a simplified view, taking only first degree effects into consideration, but are easily understood and well suited to educational needs.

8. What are the effects of scaling of MOS circuits? Explain in detail.

Answer:

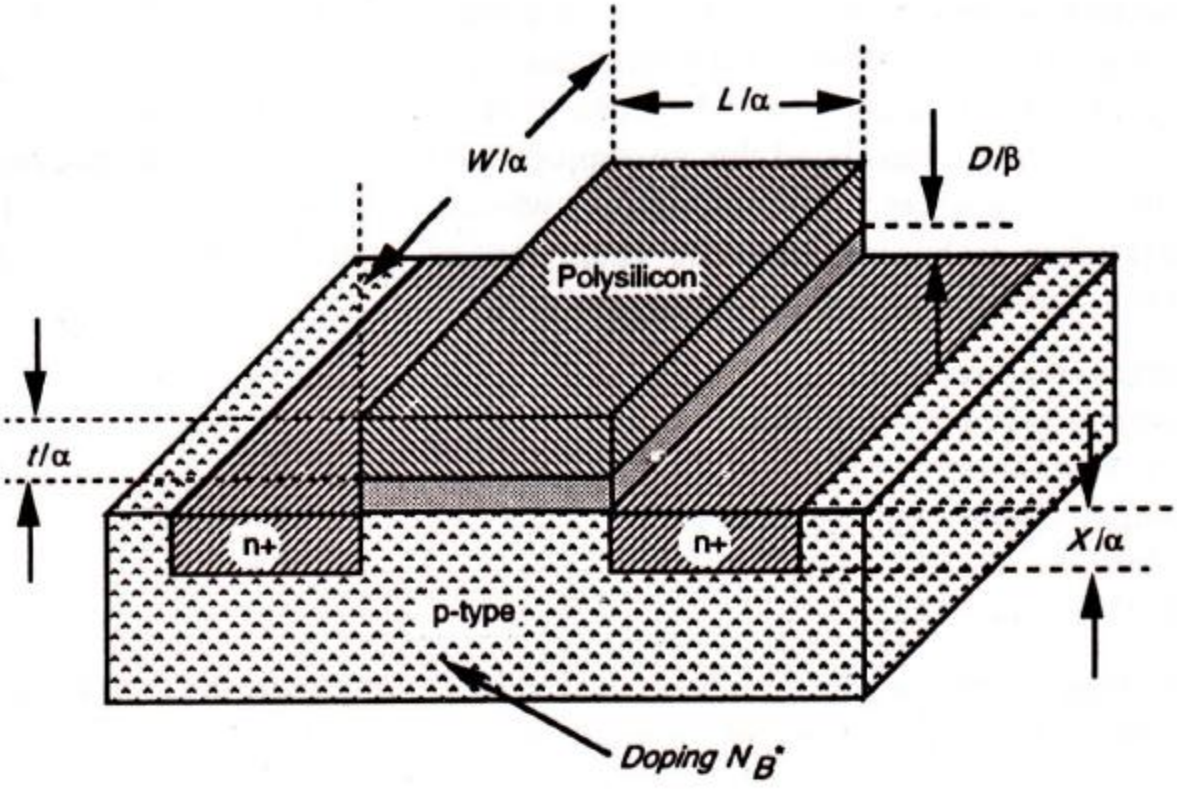


FIGURE 5.1 Scaled nMOS transistor (pMOS similar).

In order to accommodate the three models, two scaling factors—$1/\alpha$ and $1/\beta$—are used. $1/\beta$ is chosen as the scaling factor for supply voltage V_{DD} and gate oxide thickness D, and $1/\alpha$ is used for all other linear dimensions, both vertical and horizontal to the chip surface. For the constant field model and the constant voltage model, $\beta = \alpha$ and $\beta = 1$ respectively are applied.

SCALING FACTORS FOR DEVICE PARAMETERS

Gate Area A_g

$$A_g = L.W.$$

where L and W are the channel length and width respectively. Both are scaled by $1/\alpha$.

Thus A_g is scaled by $1/\alpha^2$

Gate Capacitance Per Unit Area C_0 or C_{ox}

$$C_0 = \frac{\varepsilon_{ox}}{D}$$

where ε_{ox} is the permittivity of the gate oxide (thinox) [= $\varepsilon_{ins.}\varepsilon_0$] and D is the gate oxide thickness which is scaled by $1/\beta$

Thus C_0 is scaled by $\frac{1}{1/\beta} = \beta$

Gate Capacitance C_g

$$C_g = C_0 L.W.$$

Thus C_g is scaled by $\beta \frac{1}{\alpha^2} = \frac{\beta}{\alpha^2}$

Parasitic Capacitance C_x

C_x is proportional to $\frac{A_x}{d}$

where d is the depletion width around source or drain which is scaled by l/α, and A_x is the area of the depletion region around source or drain which is scaled by $1/\alpha^2$.

Thus C_x is scaled by $\frac{1}{\alpha^2} \cdot \frac{1}{1/\alpha} = \frac{1}{\alpha}$

Carrier Density in Channel Q_{on}

$$Q_{on} = C_0 \, . \, V_{gs}$$

where Q_{on} is the average charge per unit area in the channel in the 'on' state. Note that C_0 is scaled by β and V_{gs} is scaled by l/β.

Thus Q_{on} is scaled by 1

Channel Resistance R_{on}

$$R_{on} = \frac{L}{W} \frac{1}{Q_{on}\mu}$$

where μ is the carrier mobility in the channel and is assumed constant.

Thus R_{on} is scaled by $\frac{1}{\alpha} \frac{1}{1/\alpha} 1 = 1$

Gate Delay T_d

T_d is proportional to $R_{on} . C_g$

Thus T_d is scaled by $\frac{1.\beta}{\alpha^2}\ \frac{\beta}{\alpha^2}$

Maximum Operating Frequency f_0

$$f_0 = \frac{W}{L}\ \frac{\mu C_0 V_{DD}}{C_g}$$

or, f_0 is inversely proportional to delay T_d.

Thus f_0 is scaled by $\frac{1}{\beta/\alpha^2} = \frac{\alpha^2}{\beta}$

Saturation Current I_{dss}

$$I_{dss} = \frac{C_0\mu}{2}\ \frac{W}{L}\ (V_{gs} - V_t)^2$$

noting that both V_{gs} and V_t are scaled by $1/\beta$, we have

I_{dss} is scaled by $\beta(1/\beta)^2 = 1/\beta$

Current Density J

$$J = \frac{I_{dss}}{A}$$

where A is the cross-sectional area of the channel in the 'on' state which is scaled by $1/\alpha^2$

So, J is scaled by $\frac{1/\beta}{1/\alpha^2} = \frac{\alpha^2}{\beta}$

Switching Energy Per Gate E_g

$$E_g = \frac{1C_g}{2}(V_{DD})^2$$

So, E_g is scaled by $\frac{\beta}{\alpha^2} \cdot \frac{1}{\beta^2} = \frac{1}{\alpha^2\beta}$

Power Dissipation Per Gate P_g

P_g comprises two components such that

$$P_g = P_{gs} + P_{gd}$$

where the static component

$$P_{gs} = \frac{(V_{DD})^2}{R_{on}}$$

and the dynamic component

$$P_{gd} = E_g f_0$$

It will be seen that both P_{gs} and P_{gd} are scaled by $1/\beta^2$

So, P_g is scaled by $1/\beta^2$

Power Dissipation Per Unit Area P_a

$$P_a = \frac{P_g}{A_g}$$

So, P_a is scaled by $\frac{1/\beta^2}{1/\alpha^2} = \alpha^2/\beta^2$

Power-speed Product P_T

$$P_T = P_g \,.\, T_d$$

So, P_T is scaled by $\frac{1}{\beta^2} \cdot \frac{\beta}{\alpha^2} = \frac{1}{\alpha^2\beta}$

Summary of Scaling Effects

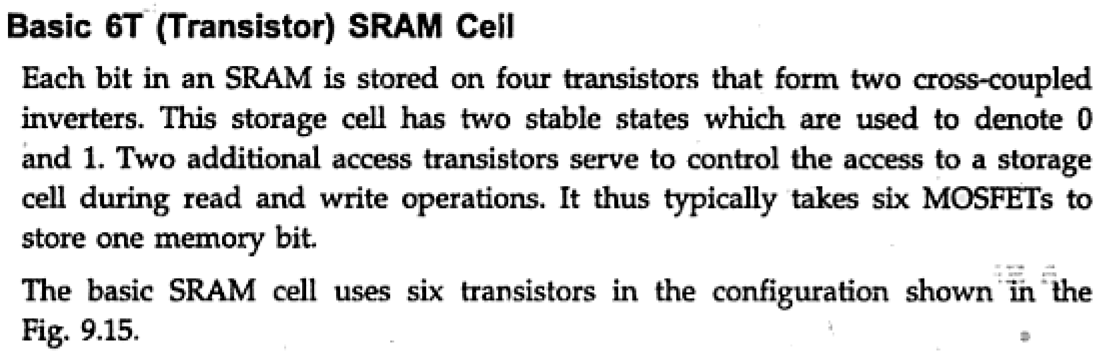
Parameters		*Combined V and D*	*Constant E*	*Constant V*
V_{DD}	Supply voltage	$1/\beta$	$1/\alpha$	1
L	Channel length	$1/\alpha$	$1/\alpha$	$1/\alpha$
W	Channel width	$1/\alpha$	$1/\alpha$	$1/\alpha$
D	Gate oxide thickness	$1/\beta$	$1/\alpha$	1
A_g	Gate area	$1/\alpha^2$	$1/\alpha^2$	$1/\alpha^2$
C_0(or C_{ox})	Gate C per unit area	β	α	1
C_g	Gate capacitance	β/α^2	$1/\alpha$	$1/\alpha^2$
C_x	Parasitic capacitance	$1/\alpha$	$1/\alpha$	$1/\alpha$
Q_{on}	Carrier density	1	1	1
R_{on}	Channel resistance	1	1	1
I_{dss}	Saturation current	$1/\beta$	$1/\alpha$	1
A_c	Conductor X-section area	$1/\alpha^2$	$1/\alpha^2$	$1/\alpha^2$
I	Current density	α^2/β	α	α^2
V_g	Logic 1 level	$1/\beta$	$1/\alpha$	1
E_g	Switching energy	$1/\alpha^2.\beta$	$1/\alpha^3$	$1/\alpha^2$
P_g	Power dispn per gate	$1/\beta^2$	$1/\alpha^2$	1
N	Gates per unit area	α^2	α^2	α^2
P_a	Power dispn per unit area	α^2/β^2	1	α^2
T_d	Gate delay	β/α^2	$1/\alpha$	$1/\alpha^2$
f_0	Max. operating frequency	α^2/β	α	α^2
P_T	Power-speed product	$1/\alpha^2.\beta$	$1/\alpha^3$	$1/\alpha^2$

Constant E:$\beta = \alpha$; Constant V:$\beta = 1$

UNIT VI - ARRAY SUBSYSTEMS

1. **Explain the operation of 6 transistor SRAM cell.**

Answer:



Basic 6T (Transistor) SRAM Cell

- Each bit in an SRAM is stored on four transistors that form two cross-coupled inverters. This storage cell has two stable states which are used to denote 0 and 1. Two additional access transistors serve to control the access to a storage cell during read and write operations. It thus typically takes six MOSFETs to store one memory bit.
- The basic SRAM cell uses six transistors in the configuration shown in the Fig. 9.15.

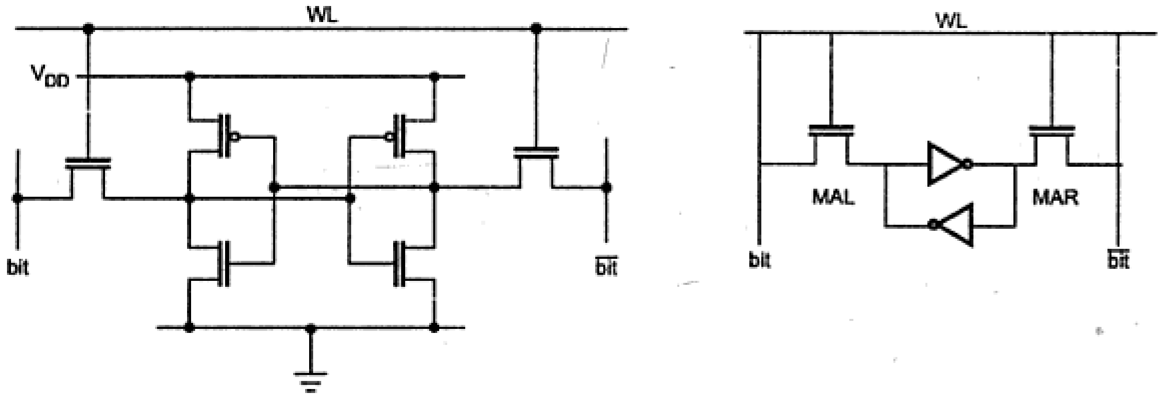
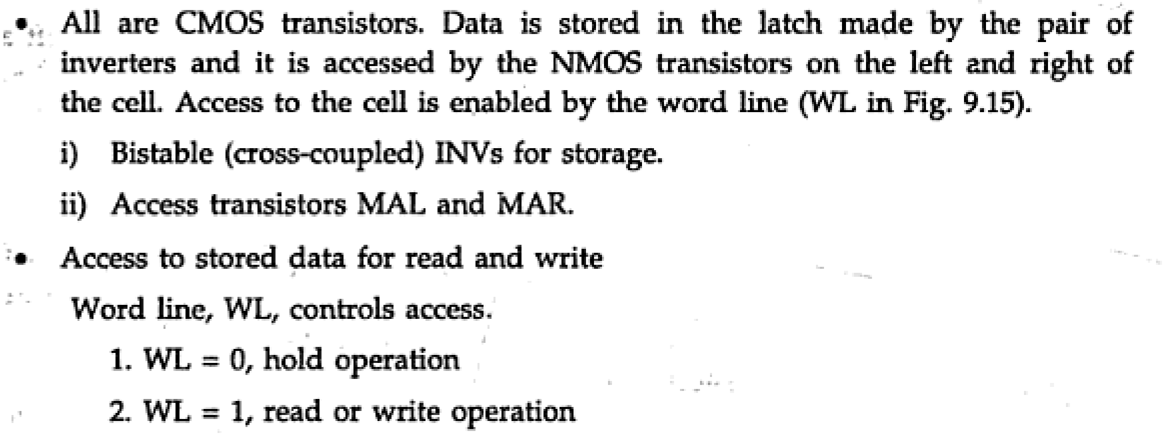


Fig. 9.15 6T SRAM cell



- All are CMOS transistors. Data is stored in the latch made by the pair of inverters and it is accessed by the NMOS transistors on the left and right of the cell. Access to the cell is enabled by the word line (WL in Fig. 9.15).
  - i) Bistable (cross-coupled) INVs for storage.
  - ii) Access transistors MAL and MAR.
- Access to stored data for read and write

  Word line, WL, controls access.

  1. WL = 0, hold operation
  2. WL = 1, read or write operation

SRAM Write Operation

- The SRAM writes into the latch through the NMOS transistors on the left and right sides of the cell. The problem is that the coupled inverters will resist state changes. Their resistance is lowered by making them small and by overpowering them with a write driver in the column amplifier.

SRAM Read Operation

- The SRAM reads the value stored in the latch using a differential amplifier. The transistors in the SRAM cell do not produce good logic levels because 1) they have small sizes (so we can write into them), 2) they have to pass through NMOS transistors which have non-ideal (resistive) behaviour when passing a "one" value and 3) they may be driving a significant capacitive load. The poor output levels that they produce must be translated into valid logic levels and a special amplifier is required to do the translation.

Read Timing

- Read timing in the SRAM is similar to the ROM. A valid address is presented and a short time later the contents of the SRAM cell will appear at the Q output. The primary difference is that the SRAM is more sensitive to the number of rows because the NMOS transistors associated with Bit and nBit are not designed to drive those signals poorly.

SRAM configuration (rows × columns)	Read timing Address to output
4 × 16	2.2 ns
4 × 32	3.2 ns
16 × 16	2.7 ns
4 × 1024	37.9 ns

Write Timing

- Writing an SRAM requires a little care. The column write driver must not drive the Bit and nBit lines before the addresses stabilize. Fig. 9.17 shows the waveform required for writing into the SRAM.

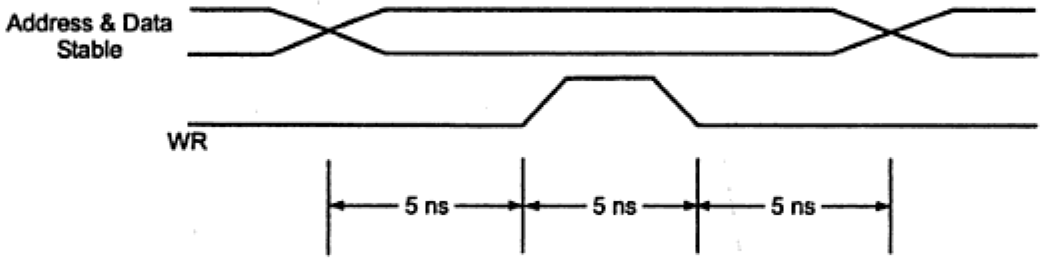


Fig. 9.17 Write timing

2. Explain the operation of DRAM cell.

Answer:

1-Transistor DRAM Cell

- Fig. 5.47 shows single transistor DRAM.

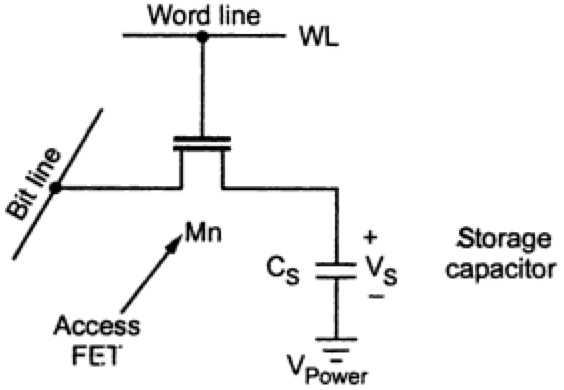


Fig. 5.47 Write timing

DRAM Operation

- RAM data is held on the storage capacitor.
  i) Temporary - due to leakage currents which drain charge.
- Charge storage
  i) If C_S is charged to V_S
  ii) $Q_S = C_S\ V_S$
- If $V_S = 0$, then $Q_S = 0$: LOGIC 0.
- If V_S = large, then $Q_S > 0$: LOGIC 1.

1. Write Operation

- Turn-on access transistor : WL = V_{DD}.
  i) Apply voltage, V_d (high or low), to bit line.
  ii) C_S is charged (or discharged).
  iii) If $V_d = 0$.
- $V_S = 0$, $Q_S = 0$, store logic 0.
  i) If $V_d = V_{DD}$.
- $V_S = V_{DD} - V_{th}$, $Q_S = C_S\ (V_{DD} = V_{th})$, logic 1.

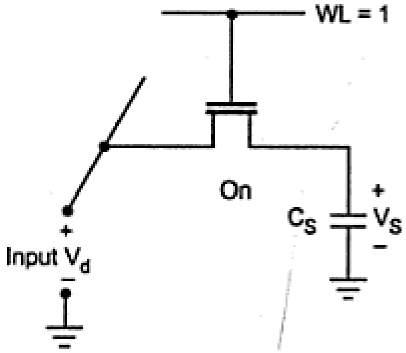


Fig. 9.19 Write operation

2. Hold Operation

- Turn-off access transistor : WL = 0.
- Charge held on C_S.
- During hold, leakage currents will slowly discharge C_S.

  i) Due to leakage in the access transistor when it is OFF.
- If I_L is known, can determine discharge time.

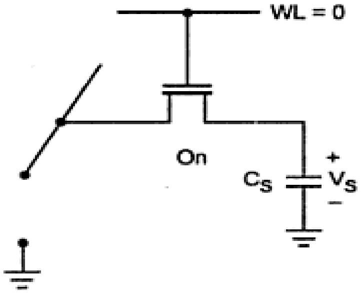


Fig. 9.20 Hold operation

3. Hold Time, t_h

i) Max time voltage on C_S is high enough to be a logic 1.

- Time to discharge from V_{max} to V_1 (in Fig. 9.20).

ii) $t_h = (C_S / I_L)\ (V_S)$, if we estimate I_L as a constant.

- Desire large hold time.
- t_h increases with larger C_S and lower I_L.
- Typical value, t_h = 0.5 sec.

iii) With I_L = 1 nA, C_S = 50 pF, and V_S = 1 V.

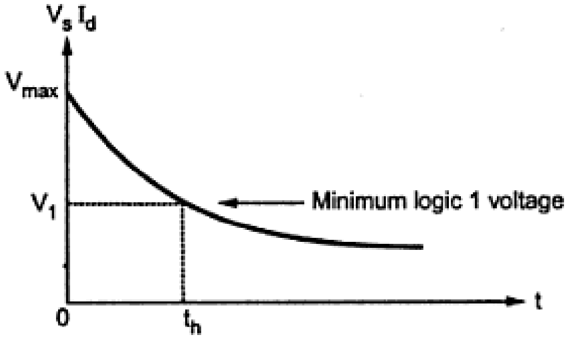


Fig. 9.21

- DRAM is "Dynamic", i.e. data is stored for only short time.

3-Transistor DRAM Cell

- A three-transistor DRAM cell is shown in Fig. 9.24.

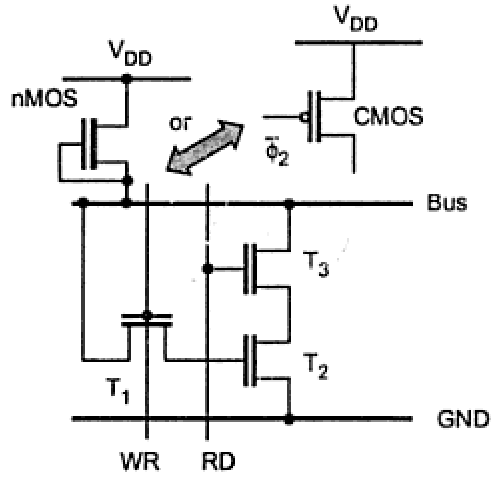


Fig. 9.24 Three-transistor dynamic memory cell

- When control line RD is LOW, then a bit may be read from bus through T_1 by making WR to HIGH. This in turns WR to LOW through T_2. The bit value is stored by gate capacitance (C_g) of T_2 while RD and WR are LOW.
- For reading stored bit RD is mode HIGH, is it pulls down bus to ground through T_3 and T_2 if a logic '1' was stored. Otherwise T_2 will not conduct and bus will remain HIGH because of pull-up arrangements.

3. Explain about SRAM & DRAM.

Answer:

SRAM

- SRAM means Static Random Access Memory. Random access means that locations in the memory can be written to or read from in any order, regardless of the memory location that was last accessed.
  1) Static : Holds data as long as power is applied.
  2) Volatile : Cannot hold data if power is removed.
- Static Random Access Memory (SRAM) is widely used in integrated circuits. All of the popular microprocessors have elaborate on-chip caching schemes for improved performance. All of these caches use SRAM.
- A SRAM cell has three different states it can be in :
  1) Standby or hold where the circuit is idle,
  2) Reading when the data has been requested and
  3) Writing when updating the data

Advantages of SRAM

- Static Random Access Memory (SRAM) offers many advantages in semiconductor integrated circuit applications. Some of the key advantages of SRAM include :

1. Compatibility with modern logic - optimized CMOS manufacturing processes.
2. High speed - fastest of all semiconductor memories.
3. Ease of use - no refresh necessary.
4. High density - high bits per area can be achieved.
5. Versatility :
   i) Embedded SRAM size can range from less than 1 kB to 10's of MB.
   ii) Basic cell structure can be modified to support multi-port access.

- Most modern integrated circuits contain atleast one SRAM array. Many VLSI chip designers will, at one time or another, work on SRAM arrays for their designs. Designing SRAM arrays, however, is quite different from designing standard digital integrated circuits.

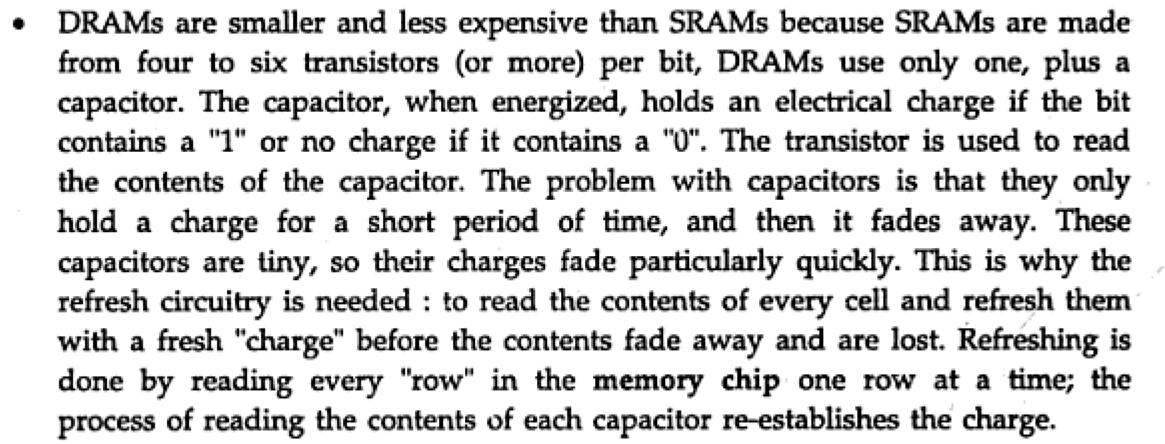
DRAM

- The term **dynamic** indicates that the memory must be constantly **refreshed** (re-energized) or it will lose its contents. **RAM** (Random Access Memory) is sometimes referred to as **DRAM** to distinguish it from **static RAM (SRAM)**. Static RAM is faster and less volatile than dynamic RAM, but it requires more power and is more expensive.

  1. **Dynamic** : Must be refreshed periodically.
  2. **Volatile** : Loses data when power is removed.

- Dynamic RAM is a type of RAM that only holds its data if it is continuously accessed by special logic called a **refresh circuit**. Many hundreds of times each second, this circuitry reads the contents of each memory cell, whether the memory cell is being used at that time by the computer or not. Due to the way in which the cells are constructed, the reading action itself refreshes the contents of the memory. If this is not done regularly, then the DRAM will lose its contents, even if it continues to have power supplied to it. This refreshing action is why the memory is called **dynamic.**

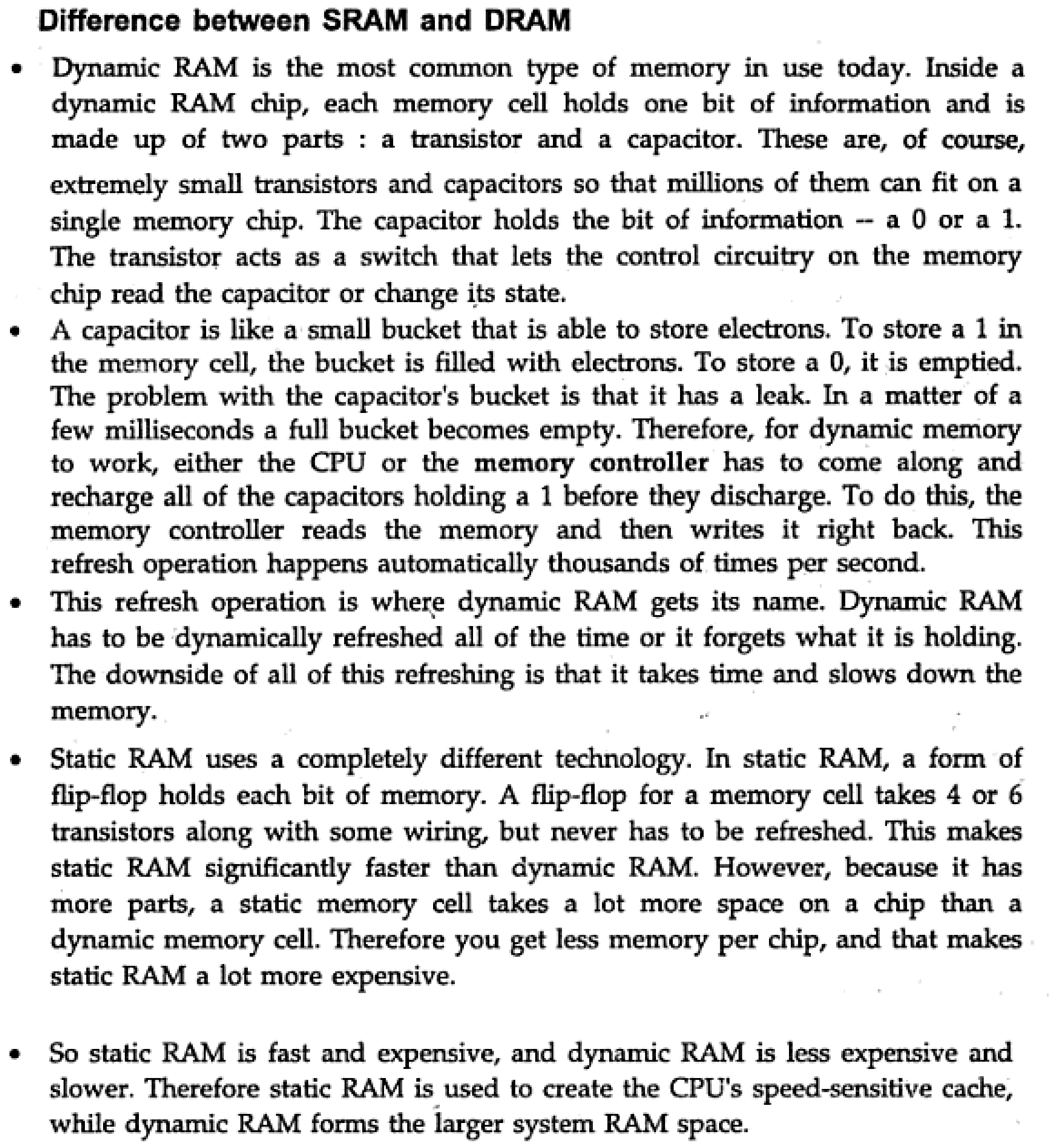
- DRAM is manufactured using a similar process to how processors are : a silicon substrate is etched with the patterns that make the transistors and capacitors (and support structures) that comprise each bit. DRAM costs much less than a processor because it is a series of simple, repeated structures, so there is not the complexity of making a single chip with several million individually-located transistors.



- DRAMs are smaller and less expensive than SRAMs because SRAMs are made from four to six transistors (or more) per bit, DRAMs use only one, plus a capacitor. The capacitor, when energized, holds an electrical charge if the bit contains a "1" or no charge if it contains a "0". The transistor is used to read the contents of the capacitor. The problem with capacitors is that they only hold a charge for a short period of time, and then it fades away. These capacitors are tiny, so their charges fade particularly quickly. This is why the refresh circuitry is needed : to read the contents of every cell and refresh them with a fresh "charge" before the contents fade away and are lost. Refreshing is done by reading every "row" in the **memory chip** one row at a time; the process of reading the contents of each capacitor re-establishes the charge.

4. Write the differences between SRAM & DRAM.

Answer;



Difference between SRAM and DRAM

- Dynamic RAM is the most common type of memory in use today. Inside a dynamic RAM chip, each memory cell holds one bit of information and is made up of two parts : a transistor and a capacitor. These are, of course, extremely small transistors and capacitors so that millions of them can fit on a single memory chip. The capacitor holds the bit of information -- a 0 or a 1. The transistor acts as a switch that lets the control circuitry on the memory chip read the capacitor or change its state.
- A capacitor is like a small bucket that is able to store electrons. To store a 1 in the memory cell, the bucket is filled with electrons. To store a 0, it is emptied. The problem with the capacitor's bucket is that it has a leak. In a matter of a few milliseconds a full bucket becomes empty. Therefore, for dynamic memory to work, either the CPU or the **memory controller** has to come along and recharge all of the capacitors holding a 1 before they discharge. To do this, the memory controller reads the memory and then writes it right back. This refresh operation happens automatically thousands of times per second.
- This refresh operation is where dynamic RAM gets its name. Dynamic RAM has to be dynamically refreshed all of the time or it forgets what it is holding. The downside of all of this refreshing is that it takes time and slows down the memory.
- Static RAM uses a completely different technology. In static RAM, a form of flip-flop holds each bit of memory. A flip-flop for a memory cell takes 4 or 6 transistors along with some wiring, but never has to be refreshed. This makes static RAM significantly faster than dynamic RAM. However, because it has more parts, a static memory cell takes a lot more space on a chip than a dynamic memory cell. Therefore you get less memory per chip, and that makes static RAM a lot more expensive.
- So static RAM is fast and expensive, and dynamic RAM is less expensive and slower. Therefore static RAM is used to create the CPU's speed-sensitive cache, while dynamic RAM forms the larger system RAM space.

- While DRAM supports access times of about 60 nanoseconds, SRAM can give access times as low as 10 nanoseconds. In addition, its cycle time is much shorter than that of DRAM because it does not need to pause between accesses. Unfortunately, it is also much more expensive to produce than DRAM. Due to its high cost, SRAM is often used only as a memory cache.
- When deciding which type of RAM to use, a system designer must consider access time and cost. SRAM devices offer extremely fast access times (approximately four times faster than DRAM) but are much more expensive to produce. Generally, SRAM is used only where access speed is extremely important. A lower cost-per-byte makes DRAM attractive whenever large amounts of RAM are required. Many embedded systems include both types : a small block of SRAM (a few kilobytes) along a critical data path and a much larger block of DRAM (perhaps even Megabytes) for everything else.

Type	Volatile	Writeable	Erase Size	Max Erase Cycles	Cost (per byte)	Speed
SRAM	Yes	Yes	Byte	Unlimited	Expensive	Fast
DRAM	Yes	Yes	Byte	Unlimited	Moderate	Moderate
Masked ROM	No	No	n/a	n/a	Inexpensive	Fast
PROM	No	Once, with a device programmer	n/a	n/a	Moderate	Fast
EPROM	No	Yes, with a device programmer	Entire chip	Limited (consult datasheet)	Moderate	Fast
EEPROM	No	Yes	Byte	Limited (consult datasheet)	Expensive	Fast to read, slow to erase/write.
Flash	No	Yes	Sector	Limited (consult datasheet)	Moderate	Fast to read, slow to erase/write.

UNIT V PROGRAMMABLE LOGIC DEVICES

1. Explain Programmable Logic Devices.

Answer:

Hardware realization of logic networks is very time-consuming and expensive. Once logic functions are realized in hardware, it is difficult to change them. In some cases, we need logic networks that are easily changeable. One such case is logic networks whose output functions need to be changed frequently, such as control logic in microprocessors, or logic networks whose outputs need to be flexible, such as additional functions in wrist watches and calculators. Another case is logic networks that need to be debugged before finalizing. Programmable logic devices (PLDs) serve this purpose.

PLD is defined as a programmable logic device or PLD is an electronic component used to build reconfigurable digital circuits. Unlike logic gate, which has a fixed function, a PLD has an undefined function at the time of manufacture. Before the PLD can be used in a circuit it must be programmed (i.e., reconfigured).

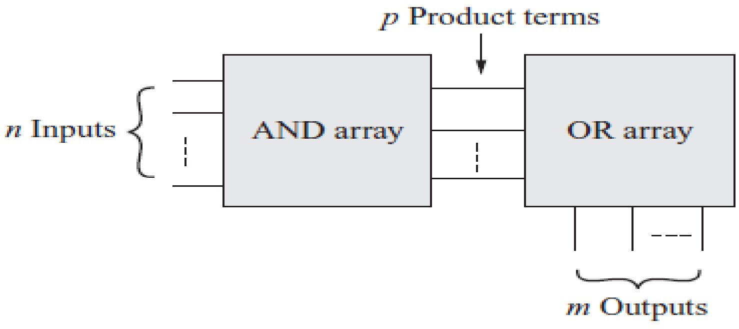
On these PLDs, all transistor circuits are laid out on IC chips prior to designers use. With PLDs, designers can realize logic networks on an IC chip, by only deriving concise logic expressions such as minimal sums or minimal products, and then making connections among pre-laid logic gates on the chip. So, designers can realize their own logic networks quickly and inexpensively using these pre-laid chips, because they need not design logic networks transistor circuits, and layout for each design problem. Thus, designers can skip substantial time of months for hardware design.

CAD programs for deriving minimal sums or minimal products are well developed, so logic functions can be realized very easily and quickly as hardware, using these CAD programs. The ease in changing logic functions without changing hardware is just like programming in software, so the hardware in this case is regarded as “programmable.” Programmable logic arrays (i.e., PLAs) and FPGAs are typical programmable logic devices.

2. Explain Programmable Logic Arrays (PLA).

Answer:

Programmable Logic Array (PLA) is an IC chip used for two-level combinational logic circuits. A PLA consists of an AND array and an OR array. Both the AND array and OR array are programmable. The architecture of PLA’s is shown in figure below.



In order to store logic expressions, connections between the MOSFET gates and vertical lines in the AND array and also connections between the MOSFET gates and horizontal lines in the OR array are set up by semiconductor manufacturers during fabrication according to customer specifications. This is known as programming PLD programmable logic devices. Since for these connections only one mask is required to program the transistors, PLAs are inexpensive when production volume is high enough to make the custom preparation cost of the connection mask negligibly small. Because of low cost and design flexibility, PLAs are extensively used in VLSI chips, such as microprocessor chips for general computation and microcontroller chips for home appliances, toys, and watches. In the PLA approach instead of generating all the minterms a separate logic is implemented which generates only the required product terms. This saves lot of silicon area. Also common product terms are identified and only one product term is generated for that particular term.

3. Explain advantages, disadvantages and applications of PLA's.

Answer:

PLA's have the following advantages over random-logic gate networks, where random-logic gate networks are those that are compactly laid out on an IC chip:

1. There is no need for the time-consuming logic design of random-logic gate networks and even more time-consuming layout.
2. Design checking is easy, and design change is also easy.
3. Layout is far simpler than that for random-logic gate networks, and thus is far less time consuming.
4. When new IC fabrication technology is introduced, we can use previous design information with ease but without change, making adoption of the new technology quick and easy.
5. Only the connection mask needs to be custom made.

PLA is a very inexpensive approach, greatly shortening design time. PLA's have the following disadvantages compared with random-logic gate networks. Random logic gate networks have higher speed than PLA's or ROM's.

1. Random-logic gate networks occupy smaller chip areas than PLA's or ROM's, although the logic design and the layout of random-logic gate networks are far more tedious and time-consuming.
2. With large production volumes, random-logic gate networks are cheaper than PLA's or ROM's.

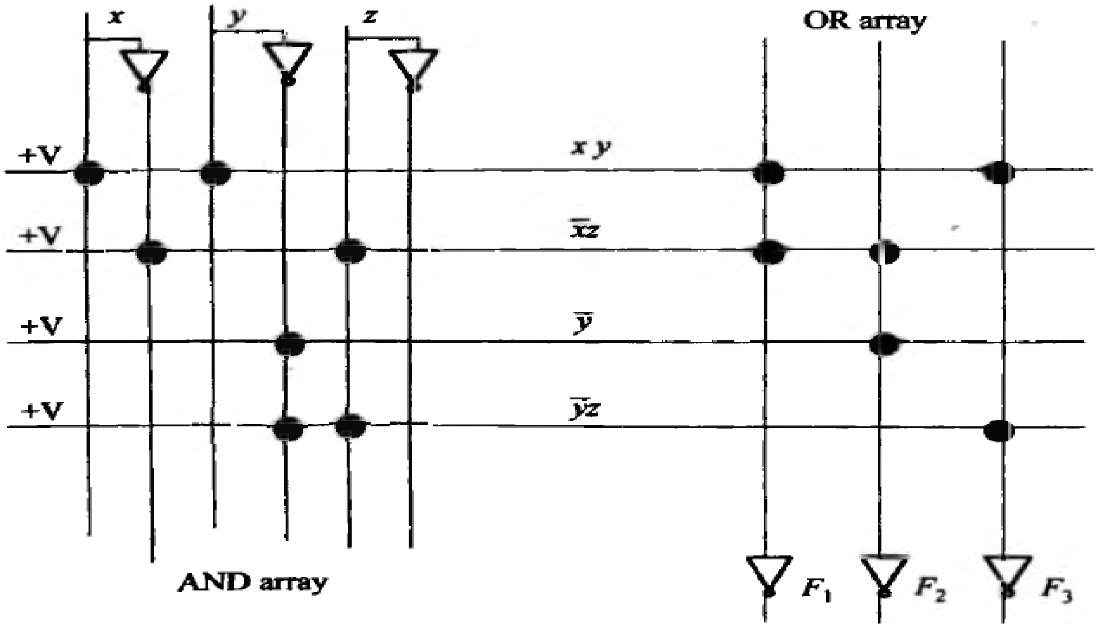
Applications of PLA's:
A microprocessor chip uses many PLA's because of easy of design change and check. PLA's are used in its control logic, which is complex and requires many changes even during its design. PLA's are used for code conversions, micro programs, address conversions, decision tables, bus priority resolvers, and memory overlay.

When a new product is to be manufactured in small volume or test marketed, PLA is used. When new product is well received in the market and does not need further changes PLA's can be replaced by random-logic gate networks for low cost for high volume production and high speed. Full custom design approach is very time consuming, taking months or years, but if PLA's are used in the control logic, a number of different custom design chips with high performance can be made quickly by changing only one connection mask for the PLAs, although these chips cannot have drastically different performance and functions.

4. Implement the following Boolean functions using PLA.

$$F_1 = xy + \bar{x}z$$
$$F_2 = \bar{y} + \bar{x}z$$
$$F_3 = xy + \bar{y}z$$

Answer:

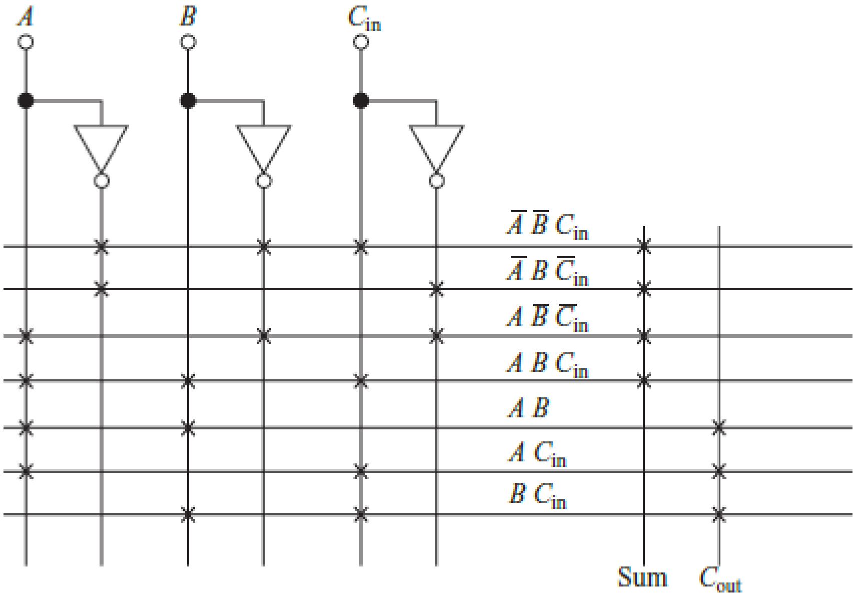


5. Implement Full Adder using PLA.

Answer:

$$\text{Sum} = \overline{A}\ \overline{B}\ C_{\text{in}} + \overline{A}\ B\ \overline{C}_{\text{in}} + A\ \overline{B}\ \overline{C}_{\text{in}} + ABC_{\text{in}}$$

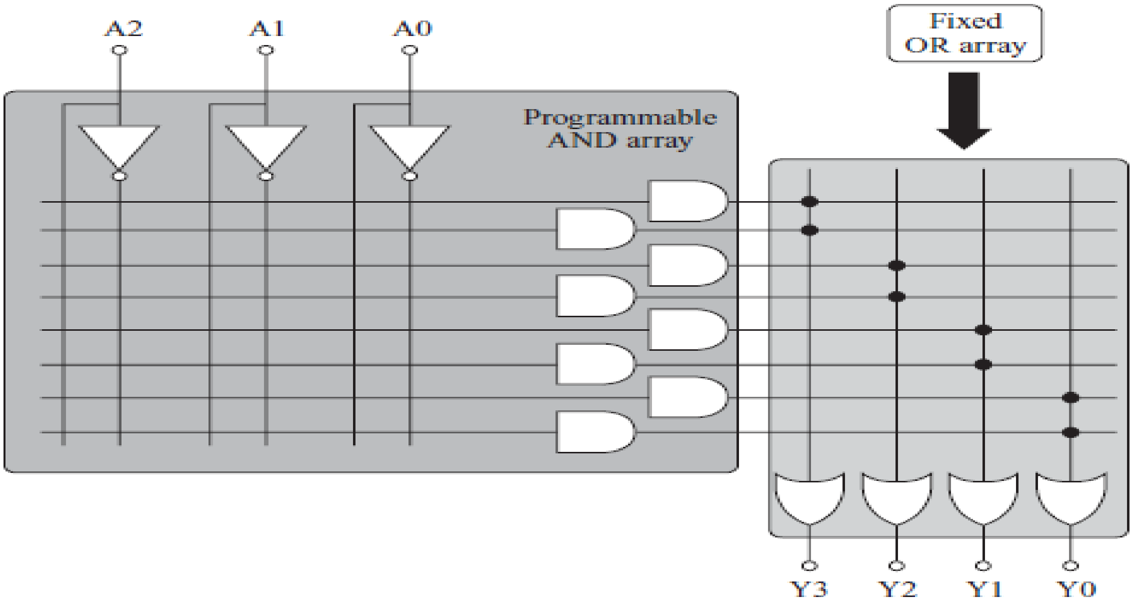
$$C_{\text{out}} = AB + AC_{\text{in}} + BC_{\text{in}}$$


A
B
C_{in}
$\overline{A}\,\overline{B}\,C_{\text{in}}$
$\overline{A}\,B\,\overline{C}_{\text{in}}$
$A\,\overline{B}\,\overline{C}_{\text{in}}$
$A\,B\,C_{\text{in}}$
$A\,B$
$A\,C_{\text{in}}$
$B\,C_{\text{in}}$
Sum
C_{out}

6. Explain Programmable Array Logic (PAL).

Answer:

A programmable array logic (PAL) is a special type of a PLA where the OR array is not programmable. In other words, in a PAL, AND array is programmable but the OR array is fixed; whereas in a PLA, both arrays are programmable. The architecture of PAL is shown in figure below.



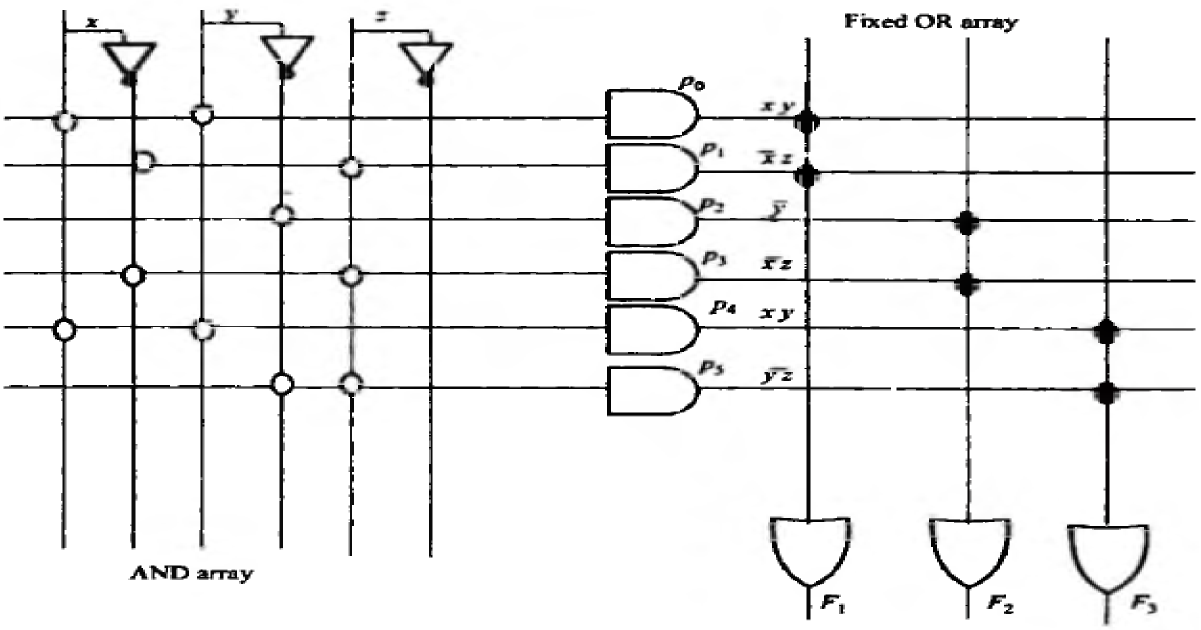
Unlike PLA, the product terms cannot be shared between the OR gates. Each function must be simplified individually to reduce the product terms to maximum two. If the SOP expression contains more than two product terms, each OR gate can be used to implement the function partially, and then summed using the additional OR gate to implement the complete function.

The advantage of PAL's 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 substantially reduced in PAL. Since single-output, two-level networks are needed most often in design practice, many two-level networks which are mutually unconnected are placed in some PAL packages.

7. Implement the following Boolean functions using PAL.

$$F_1 = xy + \bar{x}z$$
$$F_2 = \bar{y} + \bar{x}z$$
$$F_3 = xy + \bar{y}z$$

Answer:



8. **Explain design approaches in VLSI.**

Answer:

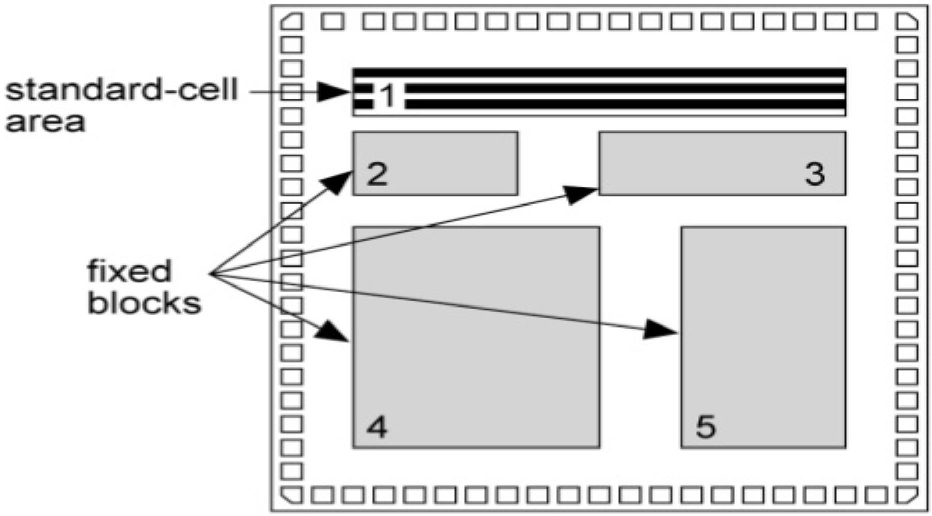
Following are the design approaches in VLSI,

1) Full Custom
2) Semi Custom
   a) Cell Based
      i. Standard Cells
      ii. Macro Cells
   b) Array Based
      i. Pre-diffused Gate Arrays
      ii. Pre-wired FPGA's

Full Custom:
In the custom design approach, each individual transistor is designed and laid out manually. The main advantage of this method is that the circuit is highly optimized for speed, area, or power. This design style is only suitable for very high performance circuitries, however, due to amount of manual work involved.

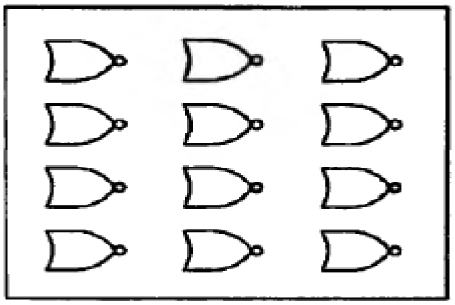
Semi Custom:
In this approach, the majority of the chip is designed using a group of predefined cells called as standard cells and rest are designed manually. The cells are predesigned, pretested and precompiled. It is up to the designer to import them into the design.

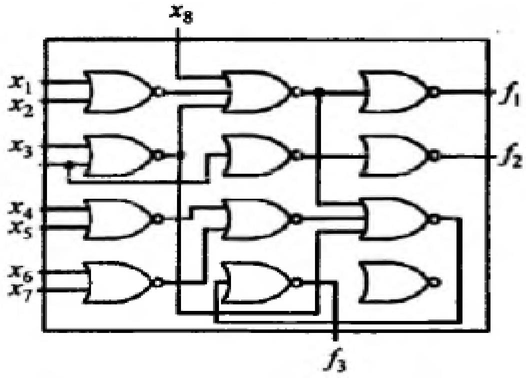
Standard Cells:
Predesigned logic cells like gates, multipliers, flip flops etc. known as standard cells are used in standard cell based design. The designer defines only the placement of the standard cells and interconnections. Standard cells can be placed anywhere on the silicon. The designer can save time and reduces the risk by making use of predesigned standard library. A standard cell requires less area for a given function as macros are very compact.



Gate Arrays:

A gate array is an IC chip on which gates are placed in matrix form without connection among the gates. By connecting gates, we can realize logic works. Connections among gates run in narrow strips of space between columns or rows of gates. These strips of space are called routing channels.



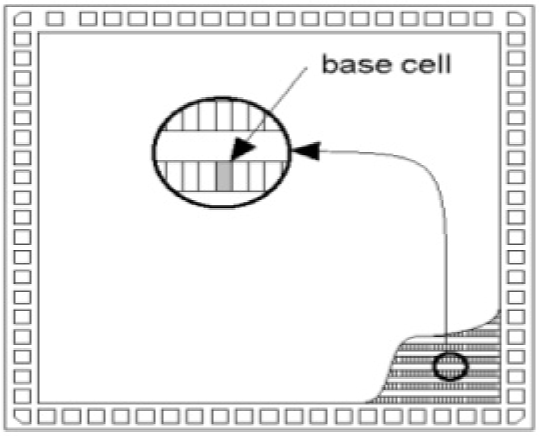


Gate arrays are of three types.

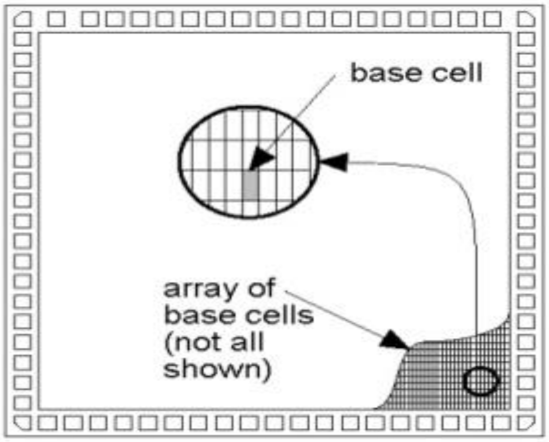
1) Channeled Gate Arrays
2) Channel Less Gate Arrays
3) Structured Gate Arrays

Channeled Gate Arrays:

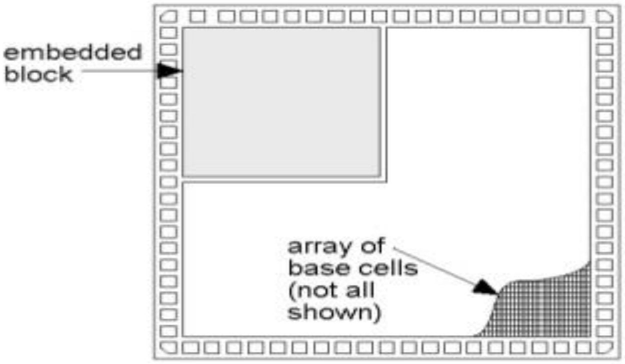
It contains multiple rows of basic cells with interconnect spaces between the rows. The space between the rows of cells is fixed.



Channel Less Gate Arrays (or) Sea of Gates:
Here there is no predefined area set aside for routing between the rows of cells. It occupies the entire core of the chip. All interconnections are passed over cells. The number of array elements is increased.



Structured Gate Arrays:
It contains custom blocks for embedded gate array functions containing different memory types and size as well as variety of embedded functions. It gives improved area efficiency.

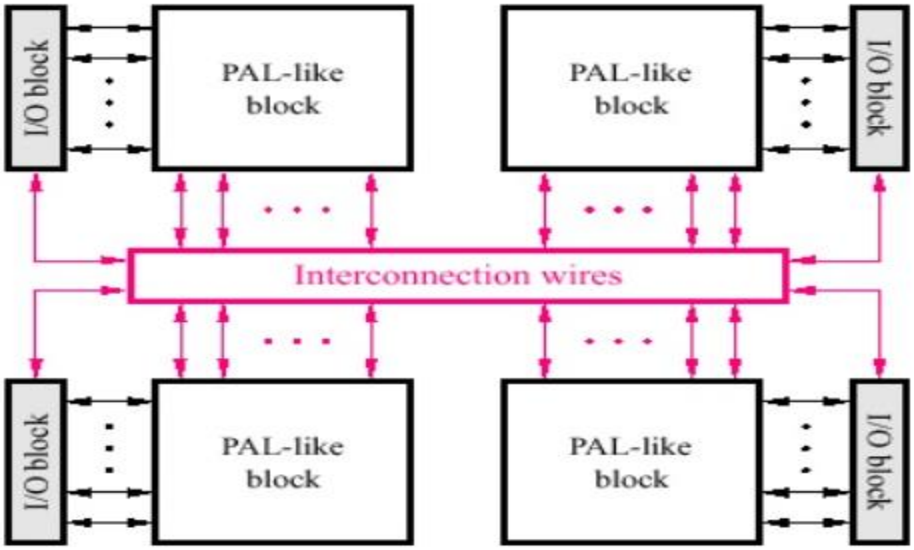


9. Explain the architecture of CPLD's.

Answer:

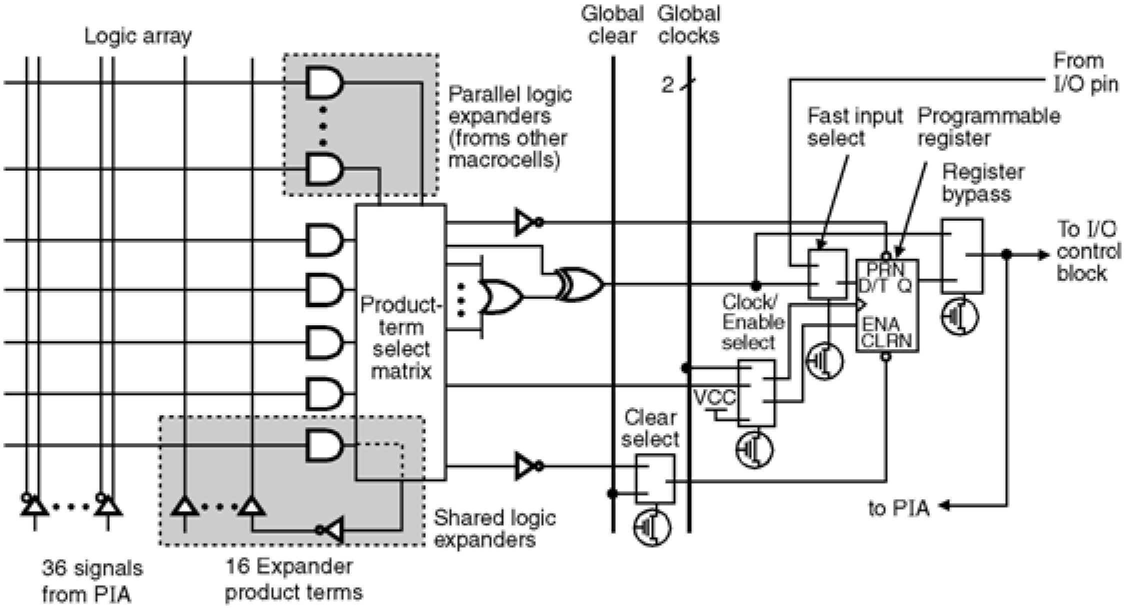
Complex Programmable Logic Devices (CPLD's) has large number of PAL's on a single chip, connected to each other through a cross point switch and can handle much more complex logic. CPLD is a programmable logic device with complexity between that of PAL's and FPGA's, and architectural features of both. The building block of a CPLD is the logical block, which contains logic block which contains logic implementing intended logic operations. It has non-volatile configuration memory and can function immediately on system start-up. CPLDs typically have the equivalent of thousands to tens of thousands of logic gates. The devices are programmed using programmable elements like EPROM cells, EEPROM cells, or Flash EPROM cells. The architecture of CPLD shown in figure below has,

1) PAL Blocks (Functional Blocks)
2) Interconnect matrix
3) Input/ Output (I/O) Blocks



Functional Block:

A typical functional and I/O block is shown in figure below. The AND plane can accept inputs from the I/O blocks, other function blocks, or feedback from the same function block. The terms are then ORed together using a fixed number of OR gates, and terms are selected via a large multiplexer. The outputs of the multiplexer can then be sent straight out of the block, or through a clocked flip-flop. This particular block includes additional logic such as a selectable exclusive OR and a master reset signal, in addition to being able to program the polarity at different stages.



Input/ Output (I/O) Block:

The I/O block is used to drive signals to the pins of the CPLD device at the appropriate voltage levels with the appropriate current. Usually, a flip-flop is included. This is done on outputs so that clocked signals can be output directly to the pins without encountering significant delay. It is done for inputs so that there is not much delay on a signal before reaching a flip-flop which would increase the device hold time requirement. Also, some small amount of logic is included in the *VO* block simply to add some more resources to the device.

Interconnect Wires:
The CPLD interconnect is a very large programmable switch matrix that allows signals from all parts of the device go to all other parts of the device. While no switch can connect all internal function blocks to all other function blocks, there is enough flexibility to allow many combinations of connections.

10. Explain advantages and applications of CPLD's.

Answer:

Advantages of CPLD's:

1) Ease of design
2) Faster time to market
3) Low development tools
4) Longer time in market through field upgrade ability
5) Increased product revenue
6) Reduced PCB area
7) Decreased component inventory
8) Lower cost

Applications of CPLD's:

1) Used in implementing random glue logic to prototyping small gate arrays.
2) Realize complex designs such as graphics controller, LAN controllers, UARTS, cache control etc.
3) Used for conversions of designs which consist of multiple SPLD's into a smaller number of CPLD's.
4) Easy to make design changes through reprogramming and reconfigure hardware without power down.

11. Explain FPGA architecture.

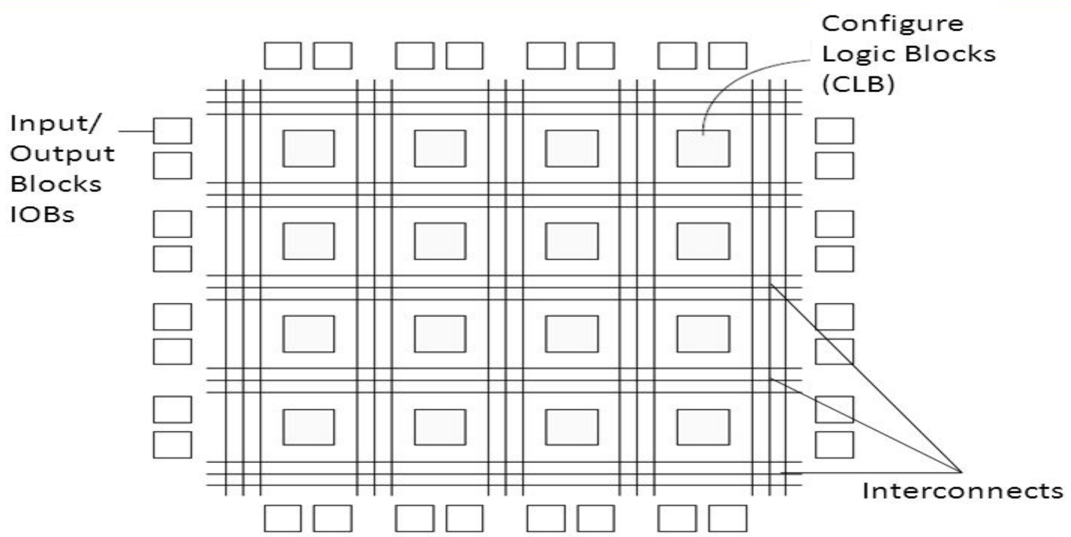
Answer:

A field-programmable gate array (FPGA) is a semiconductor device that can be configured by the customer or designer after manufacturing, hence the name field-programmable. FPGAs are programmed using a logic circuit diagram or a source code in a hardware description language (HDL) to specify how the chip will work. FPGA can be used to implement any logical function that an application-specific integrated circuit (ASIC) could perform. Unlike an Application Specific Integrated Circuit (ASIC) which can perform a single specific function for the lifetime of the chip and it can be reprogrammed to perform a different function in a matter of microseconds.

The architecture of FPGA shown in figure below contains,

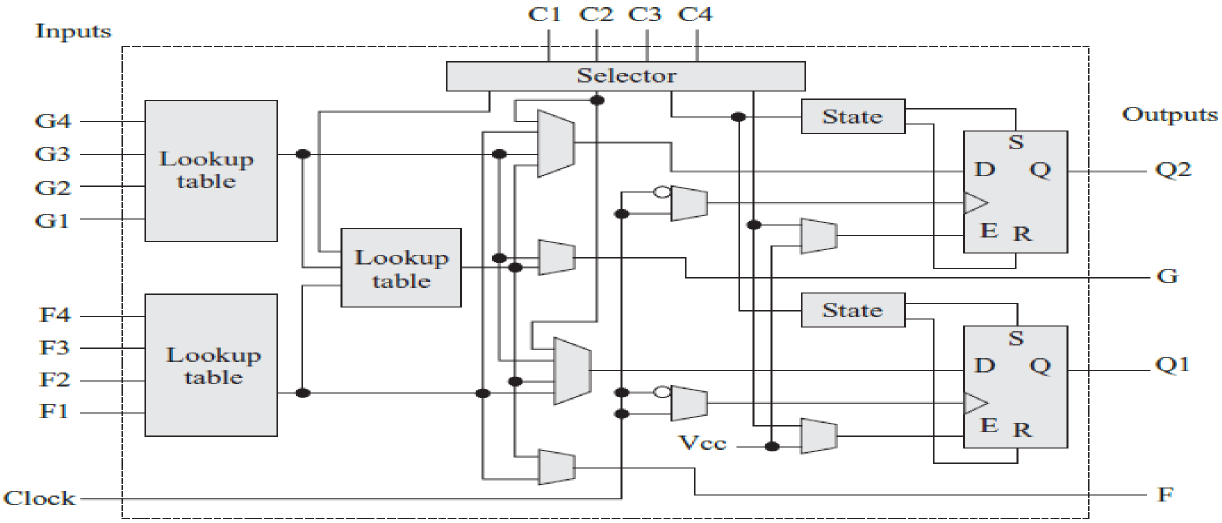
1) Rectangular array of configurable logic blocks (CLBs) capable of implementing a variety of logic functions.

2) Programmable interconnection resources or wringing tracks in simple wires to route the signals between the CLBs.
3) Switches to connect the horizontal and vertical wiring tracks.
4) Configurable I/O blocks for signal conditioning at the chip input and output pins.



Configurable Logic Blocks (CLB):

In the configurable logic block, look up table is used to implement any number of different functionality. The input lines go into the input and enable the lookup table. The output of the lookup table gives the result of the logic function that it implements. Lookup table is implemented using SRAM cells and multiplexers. A lookup table with *K* inputs corresponds to 2K X l-bit SRAM, and the user can realize any k-input logic function by programming logic function's truth table directly into the memory. Number of different possible functions for k input LUT is 2K. Advantage of such architecture is that it supports implementation of so many logic functions; however, the disadvantage is unusually large number of memory cells required to implement such a logic block in case number of inputs is large. Figure below shows the block diagram of typical CLB.



It contains two four input lookup tables fed by CLB inputs, and a third lookup table fed by the other two. This arrangement allows the CLB to implement a wide range of logic functions of up to nine inputs, two separate four input functions, or other possibilities. Each CLB also contains two flip-flops. Each CLB contains circuitry that allows it to efficiently perform arithmetic. Hence lookup table is a small black box to implement its intended function taking many inputs giving a single output. Also, users can configure the lookup tables as read/write RAM cells.

Routing Techniques:

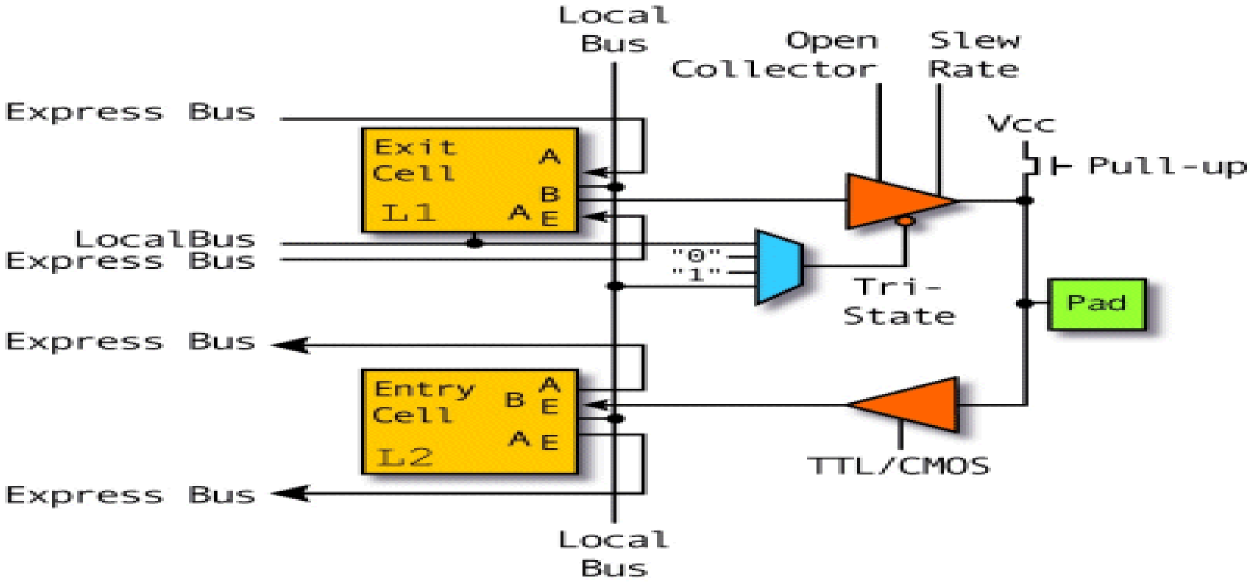
Routing architecture comprises of programmable switches and wires. Routing provides connection between I/O blocks and logic blocks, and between one CLB and another CLB. The type of routing architecture decides area consumed by routing and density of logic blocks. Routing technique used in an FPGA largely decides the amount of area used by wire segments and programmable switches as compared to area consumed by logic blocks. Connection between different CLBs is done through switch matrix. This is similar to the switching box in the telecommunications. A wire segment can be described as two end points of an interconnect with no programmable switch between them. A sequence of one or more wire segments in an FPGA can be termed as a track.

There are four types of wire segments available:

1. General purpose segments, the ones that pass through switches in the switch block.
2. Direct interconnect: ones which connect logic block pins to four surrounding connecting blocks.
3. Long line: high fan out uniform delay connections.
4. Clock lines: clock signal provider which runs all over the chip.

Configurable I/O Blocks:

A configurable I/O block, shown in figure below is used to bring signals onto the chip and send them back off again. It consists of an input buffer and an output buffer with three state and open collector output controls. Typically, there are pull up resistors on the outputs and sometimes pull down resistors. The polarity of the output can usually be programmed for active high or active low output and often the slew rate of the output can be programmed for fast or slow rise and fall times. In addition, there is often a flip-flop on outputs so that clocked signals can be output directly to the pins without encountering significant delay. It is done for inputs so that there is not much delay on a signal before reaching a flip-flop which would increase the device hold time requirement.

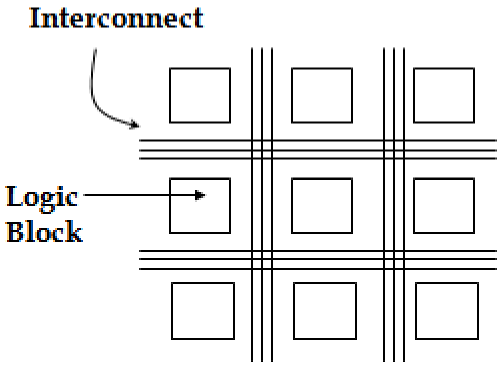


12. Explain different types of FPGA architectures.

Answer:

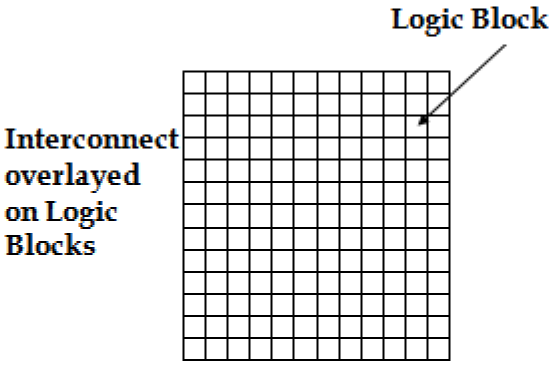
Symmetrical Array FPGA:

In this type, the structure is similar to a gate array with routing channels where each logic cell in a gate array is replaced with a logic block. It consists of a many square cluster of logic blocks surrounded by input output (I/O) blocks as shown in figure below. Each one of the CLB is able to handle a function with four Boolean variables. The interconnection resources or wiring tracks would run along the entire cluster, connecting thee logic blocks. Part of the programmable nature of the design allows users to turn on or turn off transistor based switches to connect or disconnect specific logic cells. This essentially allows the user to control which of the cells is used in creating output from the cluster in particular applications. Since the configuration of the switches is stored in Static-RAM (which is volatile), a battery backup must be used.



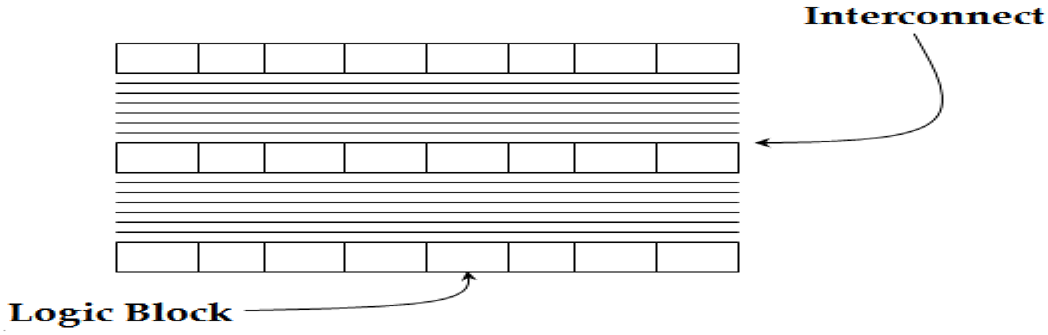
Sea of Gates:

In his design the type of interconnections between components uses overlays of the entire logic block. This allows for a much greater speed and usage of up to 40,000 logic gates, which is many, more than the maximum of 1800 used in Xilinx's early chips.

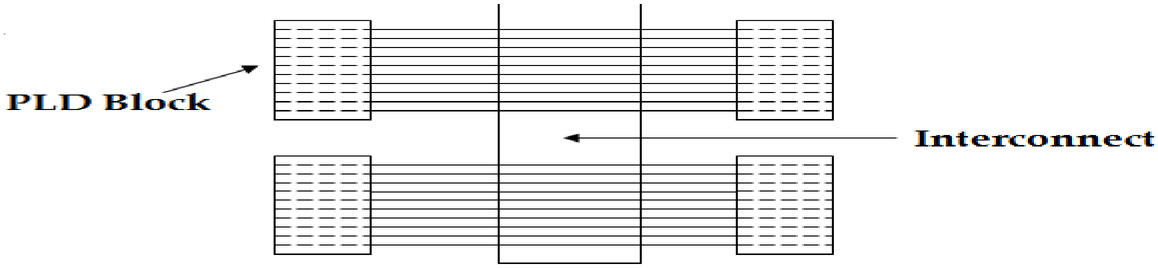


Row Based Arrays:

Row-based architecture consists of alternating rows of logic blocks and programmable interconnect tracks. Input output blocks is located in the periphery of the rows. One row may be connected to adjacent rows via vertical interconnect. Logic modules can be implemented in various combinations. Combinatorial modules contain only combinational elements where as Sequential modules contain both combinational elements along with flip-flops. These sequential modules can implement complex combinatorial-sequential functions. Routing tracks are divided into smaller segments connected by anti-fuse elements between them.



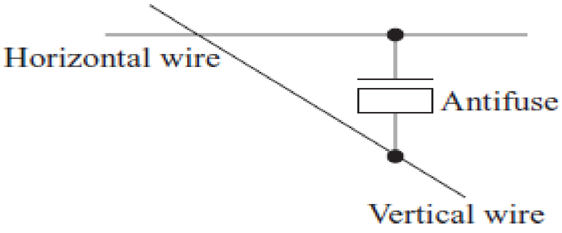
Hierarchical PLD's:
In this design less than twenty logic blocks are used, which is less than the hundreds used by Xilinx and thousands used by Actel. Despite the relatively small number of logic blocks, this design uses a comparable number of logic gates (up to 20,000). It achieves this task by creating an array of logic gates within the blocks. No external memory units are required to store information, unlike the usage of SRAM for anti-fuse technology. This allows the chips to be truly reprogrammed; however, the speed of reprogramming is nowhere near as fast as with SRAM technology.

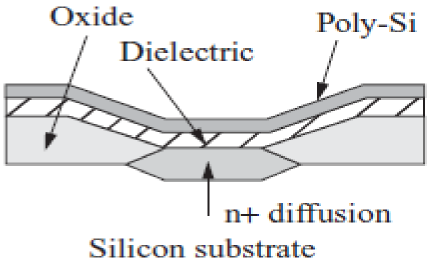


13. Explain different FPGA or CPLD programming technologies.

Answer:

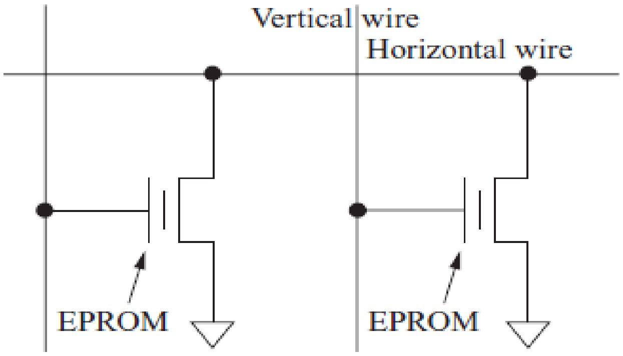
Antifuse:
The antifuse FPGAs are programmed by applying high voltage between the two terminals of the fuse to break down the dielectric material of the fuse. The antifuse switch used in FPGA is shown in figure below. Antifuse structure is normally used in an open circuit condition. However, when they are programmed, a low resistance path is established. The top and bottom layers are conducting, and the middle layer is an insulator. In normal conditions, the insulating layer isolates the top and bottom layers. But when the antifuse is programmed, a low resistance path is established through the insulator. The antifuse switches have smaller on-resistance and parasitic capacitance than pass transistors and transmission gates. Hence, it supports higher switching speed. Antifuse switches are one-time programmable, so design changes are not possible.

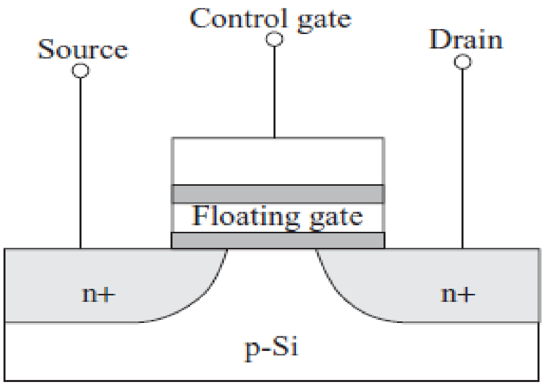




EPROM:

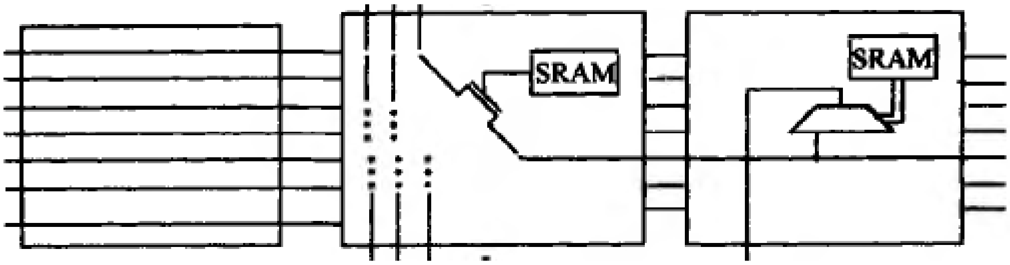
The EPROM and EEPROM technology are programmed using high voltages. The devices are reprogrammable and nonvolatile, and can be programmed while the devices are embedded in the system. The EPROM and EEPROM programming is based on the flash memory cell as shown in figure below which uses two gates, one is the control gate and another is the floating gate. Under normal mode of operation, there are no changes on the floating gate, and the transistor behaves like a normal transistor with low threshold voltage. When a high voltage is applied to the control gate, the floating gate is charged, and the threshold voltage is increased. The transistor becomes permanently OFF.

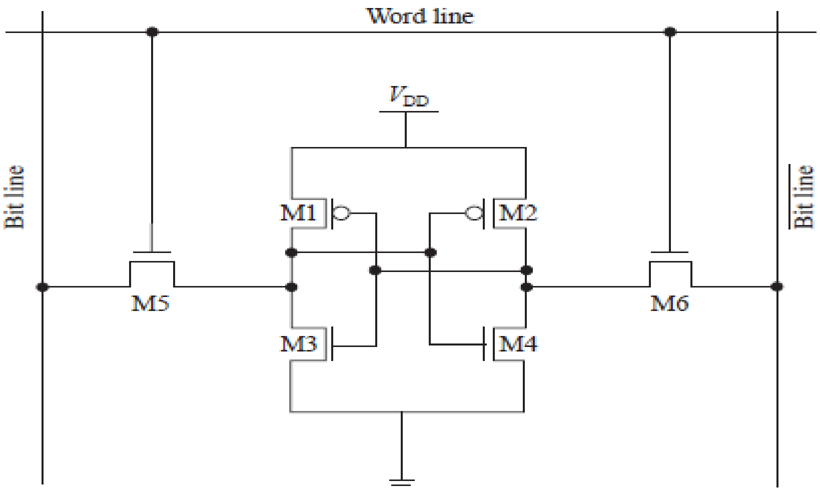




SRAM:

In the SRAM, the logic functions are based on the stored bits in the SRAM. These devices use CMOS transmission gates for switching. SRAM cells are used to control the state of pass transistors which can establish connections between horizontal and vertical wires. An SRAM memory cell consists of six transistors as shown in figure below.





14. Explain advantages and applications of FPGA's.

Answer:

Advantages of FPGA's:

1) Very fast custom logic
2) Massively parallel operation
3) Much faster than DSP engines
4) Faster than microcontrollers and microprocessors
5) More flexible than dedicated chipsets
6) More affordable and less risky than ASIC
7) Allows unlimited product differentiation
8) Reprogrammable at any time

Applications of FPGA's:

1) Applications of FPGAs include digital signal processing, software-defined radio, aerospace and defense systems, ASIC prototyping, medical imaging, computer vision, speech recognition, cryptography, bioinformatics, computer hardware emulation and a growing range of other areas.
2) FPGAs originally began as competitors to CPLDS. As their size, capabilities, and speed increased, they began to take over larger and larger functions to the state where some are now marketed as full systems on chips (SoC). Particularly with the introduction of dedicated multipliers into FPGA architectures in the late 1990s, applications which had traditionally been the sole reserve of DSPs, began to incorporate FPGAs instead.
3) FPGAs especially find applications in any area or algorithm that can make use of the massive parallelism offered by their architecture. One such area is code breaking of cryptographic algorithms.
4) FPGAs are increasingly used in conventional high performance computing applications where computational kernels such as FFT or convolution are performed on the FPGA instead of a microprocessor.
5) The inherent parallelism of the logic resources on an FPGA allows for considerable compute throughput even at a low MHz clock rates. The flexibility of the FPGA allows for even higher performance by trading off precision and range in the number format for an increased number of parallel arithmetic units. This has driven a new type of processing called reconfigurable computing, where time intensive tasks are offloaded from software to FPGAs.
6) The adoption of FPGAs in high performance computing is currently limited by the complexity of FPGA design compared to conventional software and the extremely long turnaround times of current design tools, where 4-8 hours wait is necessary after even minor changes to the source code.
7) Traditionally, FPGAs have been reserved for specific vertical applications where the volume of production is small. For these low-volume applications, the premium that companies pay in hardware costs per unit for a programmable chip is more affordable than the development resources spent on creating an ASIC for a low-volume application. Today, new cost and performance dynamics have broadened the range of vide applications.

15. Compare PLA and PAL.

Answer:

S.No.	PLA	PAL
1	Combinational logic circuits uses AND and OR planes	Combinational logic circuits uses AND and OR planes
2	AND and OR planes are programmable	Only AND plane programmable
3	Costlier than PAL	Cheaper than PLA
4	Extremely flexible	Moderately flexible
5	More no. of functions can be implemented	Less no. of functions can be implemented
6	More area	Less area
7	More delay	Less delay

16. Compare FPGA and CPLD.

Answer:

S.No.	CPLD	FPGA
1	Uses PAL like blocks	Uses Look Up Tables
2	Gate density up to 10,000 gates	Gate density up to 10,00,000 gates
3	Performance independent of routing	Performance depends on routing
4	Configuration context stored in ROM	Configuration context stored in RAM
5	Configuration context is Non-Volatile	Configuration context is Volatile
6	Suitable for low to medium density designs	Suitable for medium to high density designs
7	Simple structure	Complex structure
8	Crossbar interconnection fabric	Channel based interconnection
9	Can be reprogrammed limited no. of times	Can be reprogrammed many times
10	Coarse grained structure	Fine grained structure

17. Explain the parameters influencing Low Power Design.

Answer:

1. **Power Supply Reduction**

One of the main motivations in technology development has been to increase the levels of integration by reducing feature sizes. However, as gate lengths are reduced (without reducing voltage levels) the electric field strength increases in the gate region. This leads to reliability problems as the high electric field strengths accelerate the conducting electrons to such speeds that they cause substrate current (by dislodging holes on impact in the drain area) and actually penetrate the gate oxide. So to avoid high electric fields across transistors reduce supply voltage. Developments in fabrication are already moving from the existing standard of 5 V towards a new level of 1.8 V and experimental processes are looking at even lower voltages.

2. Variation of Threshold Voltage

The speed of a circuit is a function of (V_{gs} – V_T) where V_{gs} is the gate-source voltage and V_T is threshold voltage. Thus, it is also desirable to reduce the magnitude of the threshold voltages either to minimize the reduction in speed or to allow further reduction in V_{DD}.

3. Compensating for Lower Speed

If the target supply voltage is lowered, then the circuit speed will be reduced and the critical-path delay will be increased. To compensate for this, a designer would then have to insert or redistribute latches so that the desired frequency is again attained.

4. Voltage Swing

Reduced voltage swing reduces total power consumption. For example an internal bus architecture which is designed for operation at about 2 V with an internally generated supply for the bus itself. Modified thresholds, and special driving and sensing circuitry, allow the bus to swing less that 1 V. This not only saves power in itself but also increases the bus speed making operation at 2 V more attractive.

5. Reduce C

The second strategy is to reduce capacitance. This comes naturally with smaller feature sizes and so a circuit designer will generally wish to use the minimum geometries possible in the given technology.

6. Partition Blocks

It is best to partition large blocks into smaller ones, the product of activity and capacitance is reduced.

7. Locality of Reference

This is a design philosophy m which signals are generated and used locally in terms of their physical location on the silicon surface since the further a signal has to travel; the higher is the capacitance of that connection. With signals being processed locally, there is greater opportunity for parallel execution. With parallel execution, there is greater throughput that could be traded off for a lower supply voltage and so lower power consumption.

8. Clocks and Control

In architectures with distributed processing, the question arises as to whether there should be global control and clock signals. On the one hand, there needs to be synchronization between communicating pairs of processors; on the other hand, the global distribution network has a very large capacitance and is switched frequently. By using True Single Phase Clocking (TSPC), a design can greatly reduce the capacitance of its largest network. TSPC has already been used to implement extremely fast and power-hungry designs. The speed advantage could be traded off against power by designing for lower supply voltages.

9. Logic Design

Use logic families which feature low capacitance. One promising family is the Complementary Pass Transistor Logic (CPL). This uses networks of purely n-type pass-transistors to form logic functions (without any p-types). All signals are generated in complementary values and the outputs from the logic functions drive CMOS inverters.

10. Buffer Design

One recurrent problem is the design of circuitry to drive a relatively large capacitance (particularly external loads). The basic solution is a sequence of buffers with increasing gate widths; the design issue is what should be the size ratio (f) of each successive buffer. With speed as the main consideration, the classical value is f=e.

11. Reduced 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 minimized. This can be summarized by the phrase transition avoidance. As a first observation, this argues against the use of circuit styles which involve precharging and discharging as part of logic evaluation.

12. Glitch Avoidance

With some digital logic, there are spurious transitions (known as glitches) which occur due to partially resolved functions to a 1 as the logic is resolving before returning to a final value of 0. This wastes power. The problem is reduced in general by designing circuits so that there are equal delay paths between all of the gate inputs and the system inputs, thus equalizing arrival times of changing signals.

13. Point-to-Point Buses

Suppose there are two independent slowly-varying digital signals within a component. If these are distributed on independent data buses, then transitions only occur when information changes. If instead, the two signals are combined by a multiplexer onto a single bus for distribution, there is also likely to be a transition when the multiplexer is switched (i.e. when the control signal changes). Although point-to-point buses incur an area cost due to the extra interconnect routing, they save significant power by avoiding transitions which occur when mixing independent signals.

14. Reviewing the Algorithm

The power consumption of a complex system can be greatly influenced at the algorithmic level. Normally, component power consumption corresponds to the usual algorithmic performance criterion of speed since algorithmic speed is a function of the number of operations and this translates onto the component as the amount of switching. Thus, the programmer’s desire to reduce the number of steps in a computation will naturally reduce the power consumption of its implementation.

15. Reduce short circuit current

A designer needs to consider short circuit current in two ways: first, how to minimize what is unavoidable, second how to avoid what is unnecessary.

16. Resistive Networks

Some logic styles deliberately use resistive networks formed from transistors to establish the value of the output signal (e.g., pseudo-nMOS). These styles cannot be used for low power design. Secondly, some strategies for avoiding power loss involve generating multiple voltage levels using resistive networks either on-chip or at the system level. This static power loss must be carefully included in the evaluation of such strategies.

17. Switching Current

As the input to a CMOS inverter changes, there is a period during which both transistors are switched ON that is when the input voltage is between (V_{DD} - V_{tp}) and V_{tn}. During this period, there is short circuit current and so power dissipation. This is clearly dependent upon the rise time of the input signal. For poorly designed circuits, this power loss can be about 20% of the total power dissipation. A simple rule-of-thumb for designers is to size the transistors so that the delay in the output signal is the same as that of the input; with this strategy, the short-circuit power loss is reduced to 1- 2% of the dynamic power dissipation.

18. Glitch Propagation

The output glitch propagates on to other stages. In practice, this signal often takes the form of a slowly varying voltage which covers the centre of its range causing short circuit currents in the next gate. This is another source of power dissipation and a further reason to avoid logic glitches.

18. Explain about Standard Cells.

Answer:

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 (such as inverting, a logic AND, or a logic OR). The type of standard cell created to store data is referred to as a sequential cell. Flip-flops (FF) and latches are examples of sequential cells, which are indispensable elements of any ASIC library. The type of standard cell used to perform logic operations on the signals presented on its inputs is called combinational cell. Standard cells are built on transistors. They are one abstraction level higher than transistors.

An ASIC library or standard cell library is a group of standard cells glued together as a package. Typically, an ASIC library contains a sufficient number of combinational cells to perform any logic operation required by commonly used design styles with decent efficiency. It should also have many types of sequential cells to meet any storage requirements. A typical modem ASIC library, could have more than several hundred different standard cells. Those cells are categorized into groups by their functionality, such as INV, BUF, NAND, NOR, AND, OR, XOR, Boolean functions, flip-flop, and scan flip-flop.

UNIT V CMOS TESTING

1. Explain Need for Testing.

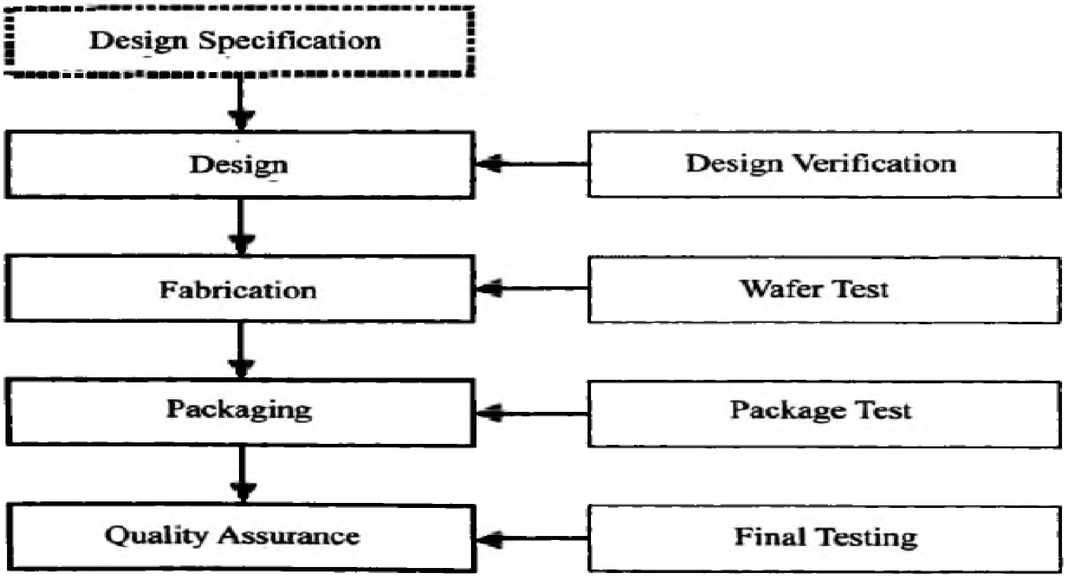
Answer:

The reduction in feature size increases the probability that a manufacturing defect in the IC will result in a faulty chip. A very small defect can easily result in a faulty transistor or interconnecting wires when the feature size is less than 100 nm. Furthermore, it takes only one faulty transistor or wire to make the entire chip fail to function properly or at the required operating frequency. Yet, defects created during the manufacturing process are unavoidable, and, as a result, some number of ICs is expected to be faulty; therefore, testing is required to guarantee fault free products, regardless of whether the product is a VLSI device or an electronic system composed of many VLSI devices. It is also necessary to test components at various stages during the manufacturing process. For example, in order to produce an electronic system, we must produce ICs, use these ICs to assemble printed circuit boards (PCBs), and then use the PCBs to assemble the system. There is general agreement with the rule of ten, which says that the cost of detecting a faulty IC increases by an order of magnitude as we move through each stage of manufacturing, from device level to board level to system level and finally to system operation in the field. Electronic testing includes IC testing, PCB testing, and system testing at the various manufacturing stages and, in some cases, during system operation. Testing is used not only to find the fault-free devices, PCBs, and systems but also to improve production yield at the various stages of manufacturing by analyzing the cause o f defects when faults are encountered. In some systems, periodic testing is performed to ensure fault-free system operation and to initiate repair procedures when faults are detected. Hence, VLSI testing is important to designers, product engineers, test engineers, managers, manufacturers, and end-users.

2. Explain testing at various stages.

Answer:

A testable circuit is defined as a circuit whose internal nodes of interest can be set to 0 or 1 and in which any change to the desired logic value at the node of interest, due to a fault, can be observed externally. VLSI development process is shown in figure below, where it can be seen that some form of testing is involved at each stage of the process. Based on a customer or project need, a ASIC device requirement is determined and formulated as a design specification. Designers are then responsible for synthesizing a circuit that satisfies the design specification and for verifying the design. Design verification is a predictive analysis that ensures that the synthesized design will perform the required functions when manufactured. When a design error is found, modifications to the design are necessary and design verification must be repeated. As a result, design verification can be considered as a form of testing. Once verified, the VLSI design then goes to fabrication. At the same time, test engineers develop a test procedure based on the design specification and fault models associated with the implementation technology. A defect is a flaw or physical imperfection that may lead to a fault. Due to unavoidable statistical flaws in the materials and masks used to fabricate ICs, it is impossible for 100% of any particular kind of IC to be defect-free. Thus, the first testing performed during the manufacturing process is to test the ICs fabricated on the wafer in order to determine which devices are defective. The chips that pass the wafer-level test are extracted and packaged. The packaged devices are retested to eliminate those devices that may have been damaged during the packaging process or put into defective packages. Additional testing is used to assure the final quality before going to market. This final testing includes measurement of such parameters as input output timing specifications, voltage and current. In addition, burn-in or stress testing is often performed where chips are subjected to high temperatures and supply voltage. The purpose of bum-in testing is to accelerate the effect of defects that could lead to, failures in the early stages of operation of the IC.



3. Explain challenges in VLSI testing.

Answer:

The physical implementation of a VLSI device is very complicated. Any small piece of dust or abnormality of geometrical shape can result in a defect. Defects are caused by process variations or random localized manufacturing imperfections. Process variations affecting transistor channel length, transistor threshold voltage, metal interconnect width and thickness, and inter metal layer dielectric thickness will impact logical and timing performance. Random localized imperfections can result in resistive bridging between metal lines, resistive opens in metal lines, improper via formation, etc. Recent advances in physics, chemistry, and materials science have allowed production of nanometre scale structures using sophisticated fabrication techniques. It is widely recognized that nanometre-scale devices will have much higher manufacturing defect rates compared to conventional complementary metal oxide semiconductor (CMOS) devices. They will have much lower current drive capabilities and will be much more sensitive to noise-induced errors such as crosstalk. They will be more susceptible to failures of transistors and wires due to soft (cosmic) errors, process variations, electromigration, and material aging. As the integration scale increases, more transistors can be fabricated on a single chip, thus reducing the cost per transistor; however, the difficulty of testing each transistor increases due to the increased complexity of the VLSI device and increased potential for defects, as well as the difficulty of detecting the faults produced by those defects.

4. Explain Test Principles.

Answer:

A fault is a representation of a defect reflecting a physical condition that causes a circuit to fail to perform in a required manner. A failure is a deviation in the performance of a circuit or system from its specified behavior and represents an reversible state of a component such that it must be repaired in order for it to provide its intended design function. A circuit error is a wrong output signal produced by a defective circuit. A circuit defect may lead to a fault, a fault can cause a circuit error, and a circuit error can result in a system failure.

Exhaustive Testing:

To test a circuit with n inputs and m outputs, a set of input patterns is applied to the circuit under test (CUT), and its responses are compared to the known good responses of a fault-free circuit. Each input pattern is called a test vector. In order to completely test a circuit, many test patterns are required; however, it is difficult to know how many test vectors are needed to guarantee a satisfactory reject rate. If the CUT is an n-input combinational logic circuit, we can apply all 2^n possible input patterns for testing stuck-at faults; this approach is called exhaustive testing. If a circuit passes exhaustive testing, we might assume that the circuit does not contain functional faults, regardless of its internal structure. Unfortunately, exhaustive testing is not practical when *n* is large. Furthermore, applying all 2^n possible input patterns to an n-input sequential logic circuit will not guarantee that all possible states have been visited. However, this method of applying all possible input test patterns to an n-input combinational logic circuit also illustrates the basic idea of functional testing.

Functional Testing:

In this testing every entry in the truth table for the combinational logic circuit is tested to determine whether it produces the correct response. In practice, functional testing is considered by many designers and test engineers to be testing the CUT as thoroughly as possible in a system-like mode of operation. In either case, one problem is the lack of a quantitative measure of the defects that will be detected by the set of functional test vectors.

Structural Testing:

The approach of structural testing is to select specific test patterns based on circuit structural information and a set of fault models. Structural testing saves time and improves test efficiency, as the total number of test patterns is decreased because the test vectors target specific faults that would result from defects in the manufactured circuit. Structural testing cannot guarantee detection of all possible manufacturing defects, as the test vectors are generated based on specific fault models; however, the use of fault models does provide a quantitative measure of the fault-detection capability of a given set of test vectors for a targeted fault model. This measure is called fault coverage.

Fault Simulation:

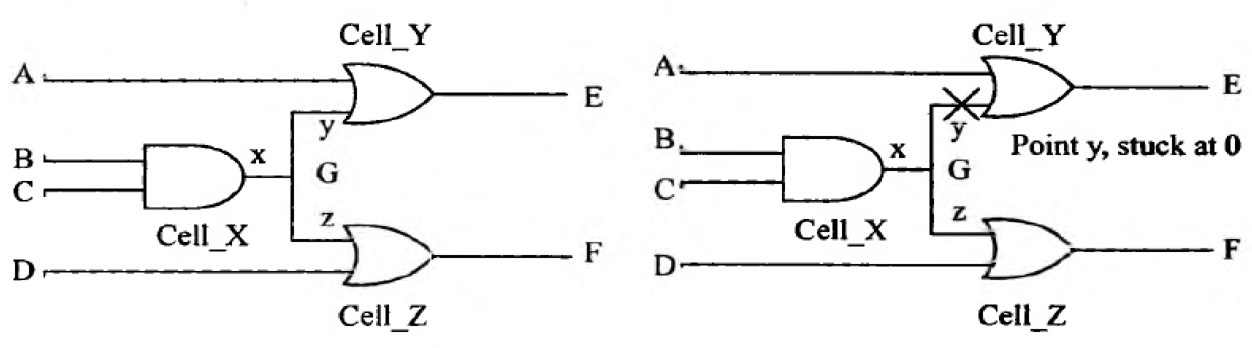
Any input pattern or sequence of input patterns that produces a different output response in a faulty circuit from that of the fault-free circuit is a test vector, or sequence of test vectors, that will detect the faults. The goal of test generation is to find an efficient set of test vectors that detects all faults considered for that circuit. Because a given set of test vectors is usually capable of detecting many faults in a circuit, fault simulation is typically used to evaluate the fault coverage obtained by that set of test vectors. As a result, fault models are needed for fault simulation as well as for test generation.

5. Explain Stuck At Fault Model.

Answer:

The most commonly used model in VLSI circuit testing is the stuck at fault (SAF) model. The SAF model assumes that any node (a net in a netlist) within a silicon chip has the potential risk of being permanently tied to power (stuck at one, SAl) or ground (stuck at zero, SAO) due to various manufacturing defects. Either SAl or SAO makes the affected node non-functional since that node cannot be switched by the circuit for logic operation any longer. Consequently, the chips that contain such nodes are regarded as bad chips and cannot be delivered to the customer. Design for test is the art of inserting some extra testing circuitry inside the chip to search for such SAF nodes.

Figure below shows an example of an SAO fault model. In this circuit, there are seven nodes (or nets); A, B, C, D, E, F, and G. An SAO fault is presented at physical location y, which belongs to node G. In other words, node G is always at ground electric potential and cannot be switched by its driver Cell X to logic “1,” regardless of Cell X’s drive strength. Thus, this circuit is not qualified for its intended design function and should be discarded.



Fault-free model

Stuck at fault model

During fault simulation or Automatic Test Pattern Generation (ATPG), test patterns are generated to stimulate the circuit and detect the effects of such SAFs. During this process, every single node in the circuit is assumed to have the potential of being stuck at either 0 or 1. The aim of a good set of test patterns is to detect all of these faulting nodes in the circuit using a minimum of resources.

6. Explain Controllability and Observability.

Answer:

The most common approach to testing a digital circuit is to toggle every node inside the circuit and observe the corresponding effect. The foundation of this approach is the SAF model. However, in practice, this is not always easily achieved. In a circuit of combinational logic, (he logic states of the internal nodes can be determined if the circuit’s inputs are all known. But for a circuit that includes sequential elements, such as flip-flops and latches, this is not true. Some of the node’s logic states depend on these sequential cell’s previous states. This leads to controllability and observability issues.

In the design for testability, for any node in a circuit, controllability is defined as the capability of a node being driven to 1 or 0 through a circuit’s inputs. If this node can be driven faithfully to 1 and 0, it is regarded as controllable. Observability is defined as the capability of the logic state of this node being observed at the circuit’s outputs. If the logic state of this node can reliably be observed, this node is regarded as observable. Whether a circuit node is stuck at 1 (or 0) is only testable if that node is both controllable and observable.

7. Explain Design for Testability (DFT).

Answer:

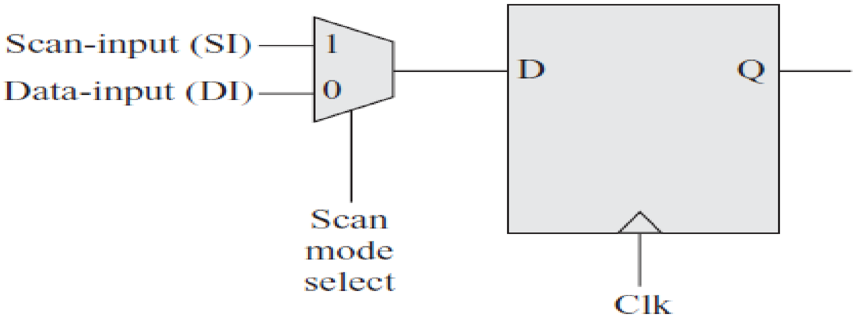
Test engineers usually have to construct test vectors after the design is completed. This invariably requires a substantial amount of time and effort that could be avoided if testing is considered early in the design flow to make the design more testable. As a result, integration of design and test, referred to as design for testability (DFT), was proposed in the 1970s. To structurally test circuits, we need to control and observe logic values of internal lines. Unfortunately, some nodes in sequential circuits can be very difficult to control and observe; for example, activity on the most significant bit of an n-bit counter can only be observed after 2^{n-1} clock cycles. Testability measures of controllability and observability were first defined in the 1970s to help find those parts of a digital circuit that will be most difficult to test and to assist in test pattern generation for fault detection. Many DFT techniques have been proposed since that time.

DFT techniques generally fall into one of the following three categories:
(1) Ad hoc DFT techniques
(2) Level-sensitive scan design (LSSD) or scan design
(3) Built-in self-test (BIST)

8. Explain Scan based Testing.

Answer:

The automatic test generation methodology is well suited for combinational circuits but not very useful for sequential circuits. Therefore, for the sequential circuits, a different technique is used, which is known as scan design. In this process, all the flip-flops are replaced by a scan flip-flop (SFF). The SFF has two modes of operation: (a) normal mode in which the flip-flop is operated in the conventional mode; and (b) scan mode in which the flip-flops are connected serially to form a large chain of shift registers throughout the entire chip. A test compiler program automatically replaces the flip flops by the SFF. This is known as scan chain insertion. By applying clock pulses, a large stream of data can be shifted in and out through the scan chain. Therefore, every sequential element can be thoroughly verified. A typical SFF is shown in figure below.



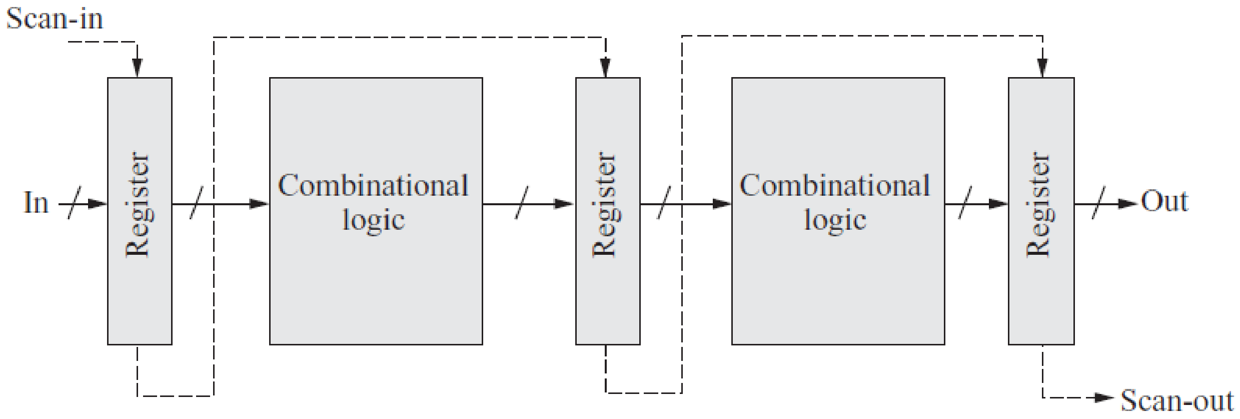
A 2:1 multiplexer is added to the input of a normal D-flip-flop to construct an SFF. When scan mode is selected to logic 1, the scan-input (SI) data goes to the D-input of SFF, and with a clock pulse, scan-input data shifts to the Q-output. The Q-output of SFF is connected to the SI input of next SFF, as well as to the input of the logic block. When scan mode is selected to logic 0, the data-input (DI) goes to the Q-output, and normal operation proceeds.

9. Explain Serial Scan and Parallel Scan Test.

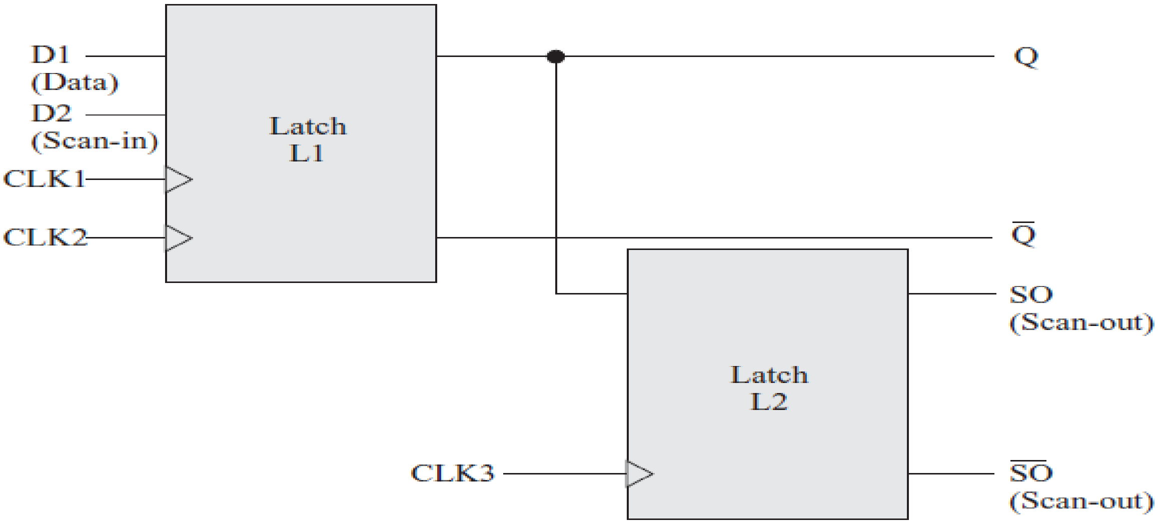
Answer:

Serial Scan Test:

Figure below shows a schematic of the serial scan test. In the scan-mode, the scan-in data input flows through the chain of registers, as illustrated by the dotted line. In the normal mode, the normal input flows through the registers and the combinational logic blocks as shown by the solid lines.



The most popular serial-scan test is called level-sensitive scan design (LSSD), which was developed by researchers from IBM in the 1970s. The LSSD is constructed using two latches, L1 and L2, as shown in figure below. The first latch called the master latch is operated using two clocks CLK1 and CLK2. The second latch called the slave is operated using a third clock CLK3. The master latch has two data inputs: D1 (data) and D2 (scan-in).



In the normal circuit operation, the signals D1, CLK1, and Q act as latch input, clock, and the output. The test clocks CLK2 and CLK3 are kept low at this mode of operation. In the scan mode, the D2 and SO signals act as scan-in and scan-out. In scan mode, the clock CLK1 is kept low, and the clocks CLK2 and CLK3 are applied by two non-overlapping two-phase clock signals.

Parallel Scan Test:
For large circuits, the size of the scan chain is too big, and there, the scan test requires a significant amount of time. To avoid this, the whole scan-chain is divided into smaller scan-blocks, and each block is scanned independently. This way it saves the overall scan test time.

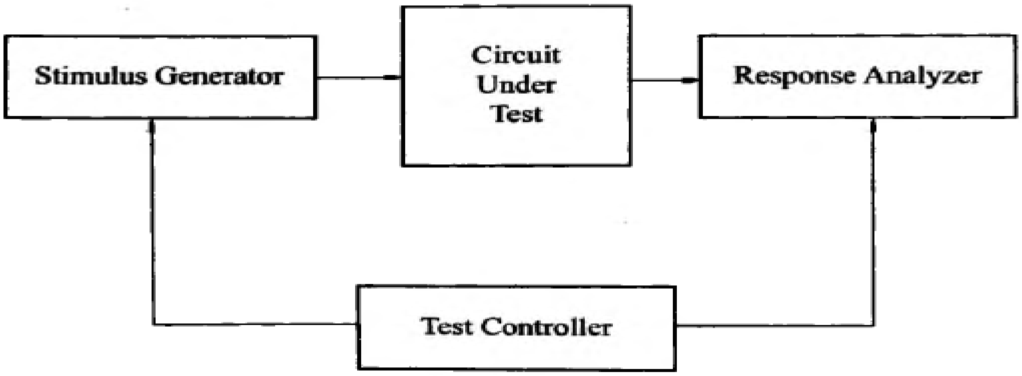
10. Explain Built in Self Test (BIST).

Answer:

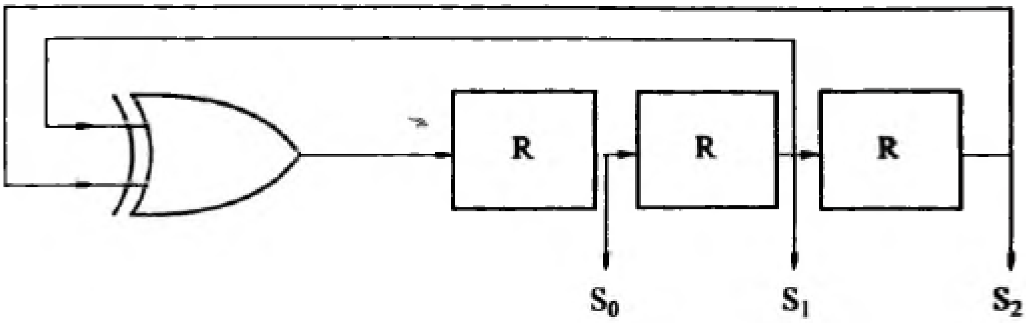
Built-in Self-Test, or BIST, is the technique of designing additional hardware and software features into integrated circuits to allow them to perform self-testing, i.e., testing of their own operation (functionally, parametrically, or both) using their own circuits, thereby reducing dependence on an external automated test equipment (ATE).

BIST is a Design-for-Testability (DFT) technique, because it makes the electrical testing of a chip easier, faster, and more efficient, aid less costly. The concept of BIST is applicable to just about any kind of circuit, so its implementation can vary as widely as the product diversity that it caters to. As an example, a common BIST approach for DRAM's includes the incorporation onto the chip of additional circuits for pattern generation, timing, mode selection, and go/no-go diagnostic tests.

The general format of a built-in self-test design is illustrated in figure below. It contains a means for supplying test patterns to the device under test and a means of comparing the device's response to a known correct sequence.



There are many ways to generate stimuli. Most widely used are the exhaustive and the random approaches. In the exhaustive approach, the test length is 2^n where n is the number of inputs to the circuit. The exhaustive nature of the test means that all detectable faults will be detected, given the space of the available input signals. An N-bit counter is a good example of an exhaustive pattern generator. For circuits with large values of N, the time to cycle through the complete input space might be prohibitive. An alternate approach is to use random testing that implies the application of a randomly chosen subset of 2^n possible input patterns. This subset should be selected so that reasonable fault coverage is obtained. An example of a pseudorandom pattern generator is the linear-feedback shift register (or LFSR), which is shown in figure below.
It consists of a serial connection of 1-bit registers. Some of the outputs are XOR'd and fed back to the input of the shift register. An N-bit LFSR cycles through 2^{n-1} states before repeating the sequence, which produces a seemingly random pattern. Initialization of the registers to a given seed value determines what will be generated, subsequently.



The response analyzer could be implemented as a comparison between the generated response and the expected response stored in an on-chip memory, but this approach represents too much area overhead to be practical. A cheaper technique is to compress the responses before comparing them. Storing the compressed response of the correct circuit requires only a minimal amount of memory, especially when the compression ratio is high. The response analyzer then consists of circuitry that dynamically compresses the output of the circuit under test and the comparator. The compressed output is often called the signature of the circuit, and the overall approach is dubbed signature analysis.

11. Define the following terms.

- **a) Yield**
- **b) Reject Rate**
- **c) Fault Coverage**
- **d) Fault Detection Efficiency**

Answer:

a) Yield

The yield of a manufacturing process is defined as the percentage of acceptable parts among all parts that are fabricated.

$$\text{Yield} = \frac{\text{Number of acceptable parts}}{\text{Total number of parts fabricated}}$$

b) Reject Rate

The ratio of field-rejected parts to all parts passing quality assurance testing is referred to as the reject rate, also called the defect level.

$$\textbf{Reject rate} = \frac{\textbf{Number of faulty parts passing final test}}{\textbf{Total number of parts passing final test}}$$

c) Fault Coverage

Fault coverage is defined as the ratio of number of faults detected in a circuit to the total number of faults present in the circuit.

$$\textbf{Fault coverage} = \frac{\textbf{Number of detected faults}}{\textbf{Total number of faults}}$$

d) Fault Detection Efficiency

$$\text{Fault detection effeciency} = \frac{\text{Number of detected faults}}{\text{Total number of faults-number of undetectable faults}}$$