Q:1 a

N-MOS depletion mode transistor:-

This transistor is normally ON, even with Vgs=0. The channel will be implanted while fabricating, hence it is normally ON. To cause the channel to cease to exist, a – ve voltage must be applied between gate and source.

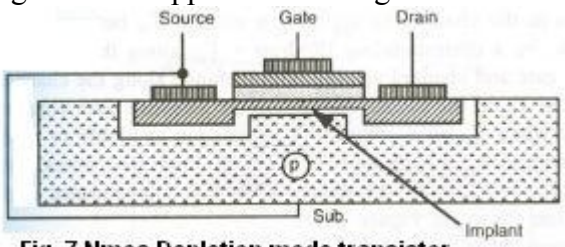
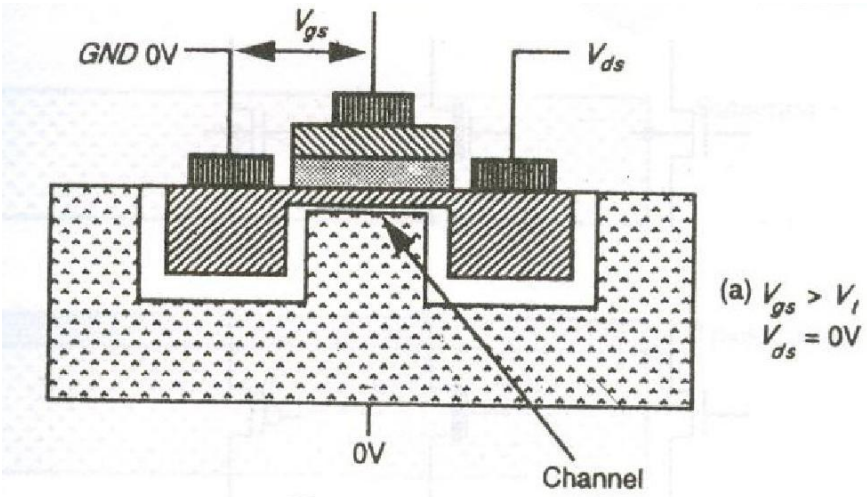
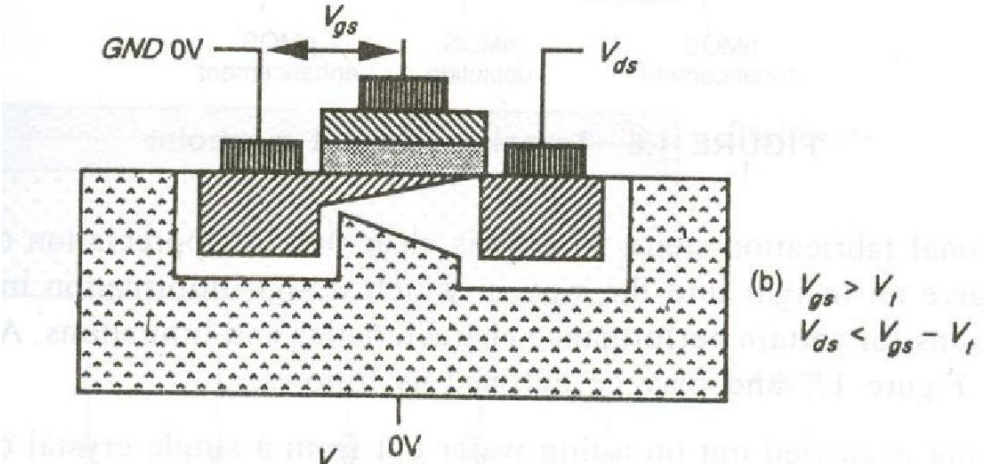


Fig. 7 Nmos Depletion mode transistor

NOTE: Mobility of electrons is 2.5 to 3 times faster than holes. Hence P-MOS devices will have more resistance compared to NMOS.

Enhancement mode Transistor action:-





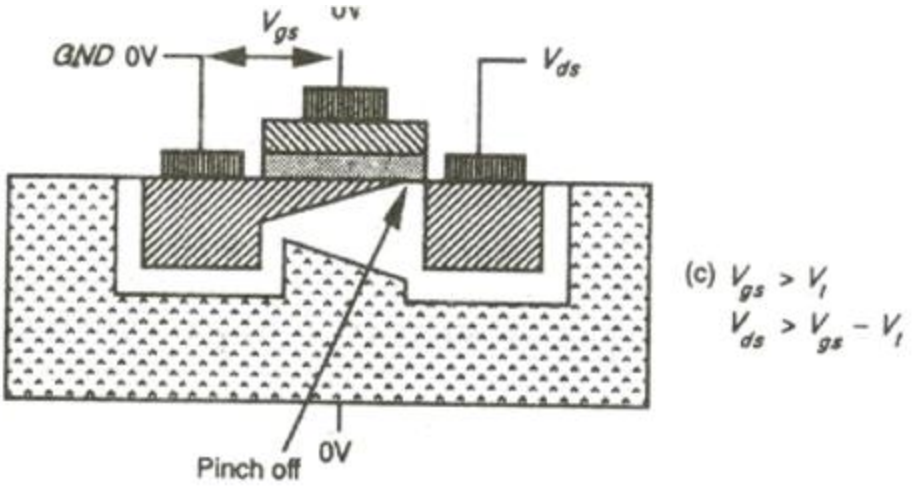


Figure 8(a)(b)(c) Enhancement mode transistor with different Vds values

To establish the channel between the source and the drain a minimum voltage (V_t) must be applied between gate and source. This minimum voltage is called as "Threshold Voltage". The complete working of enhancement mode transistor can be explained with the help of diagram a, b and c.

a) $V_{gs} > V_t$
$V_{ds} = 0$

Since $V_{gs} > V_t$ and $V_{ds} = 0$ the channel is formed but no current flows between drain and source.

b) Vgs > Vt
Vds < Vgs - Vt

This region is called the non-saturation Region or linear region where the drain current increases linearly with Vds. When Vds is increased the drain side becomes more reverse biased(hence more depletion region towards the drain end) and the channel starts to pinch. This is called as the pinch off point.

c) Vgs > Vt
Vds > Vgs - Vt

This region is called Saturation Region where the drain current remains almost constant. As the drain voltage is increased further beyond (Vgs-Vt) the pinch off point starts to move from the drain end to the source end. Even if the Vds is increased more and more, the increased voltage gets dropped in the depletion region leading to a constant current.

The typical threshold voltage for an enhancement mode transistor is given by Vt = 0.2 * Vdd.

Three MOS operating regions are :Cutoff or subthreshold region, linear region and saturation region.

The following equation describes all these three regions:

$$I_{ds} = \begin{cases} 0; & V_{gs} - V_t \leq 0 \quad cut\text{-}off \\ \beta\left[(V_{gs} - V_t)V_{ds} - \frac{V_{ds}^2}{2}\right]; & 0 < V_{ds} < V_{gs} - V_t \quad linear \\ \frac{\beta}{2}(V_{gs} - V_t)^2; & 0 < V_{gs} - V_t < V_{ds} \quad saturation \end{cases}$$

where β is MOS transistor gain and it is given by $\beta=\mu\ \varepsilon/t_{ox}(W/L)$ again 'μ' is the mobility of the charge carrier

'ε' is the permittivity of the oxide layer. 't_{ox}'is the thickness of the oxide layer.

'W' is the width of the transistor.(shown in diagram)

'L' is the channel length of the transistor.(shown in diagram)

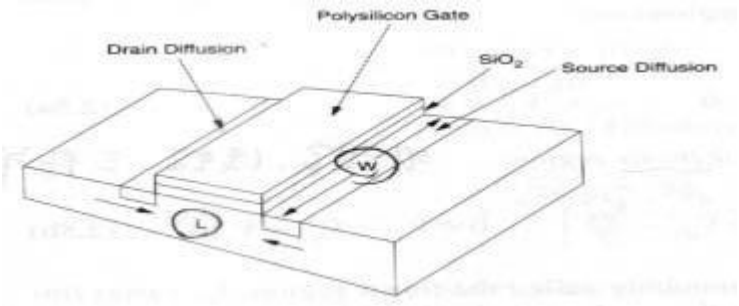


Diagram just to show the length and width of a MOSFET.

The graph of Id and Vds for a given Vgs is given below:

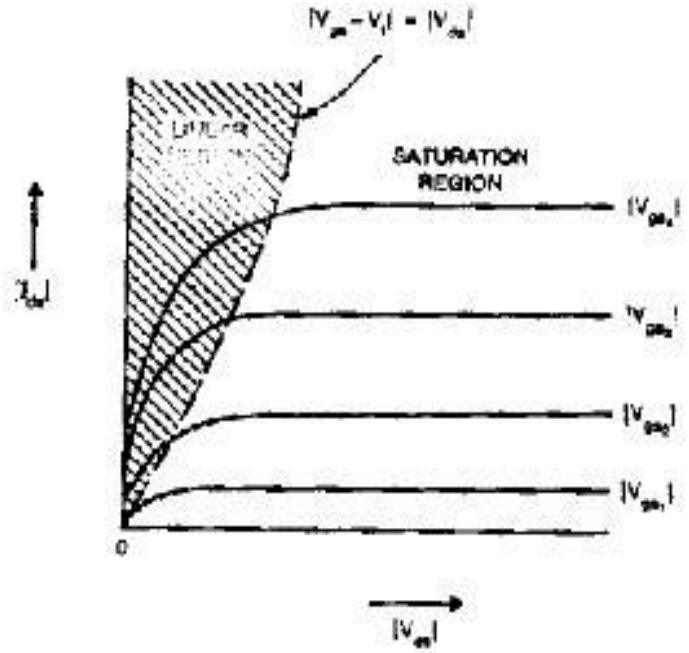


Figure 3: VI Characteristics of MOSFET

Q:1 b

CMOS INVETER CHARACTERISTICS

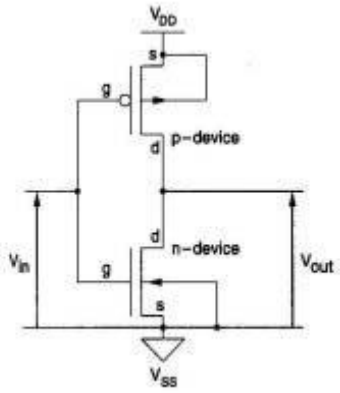


Figure 4: CMOS Inverter

CMOS inverters (Complementary NOSFET Inverters) are some of the most widely used and adaptable MOSFET inverters used in chip design. They operate with very little power loss and at relatively high speed. Furthermore, the CMOS inverter has good logic buffer characteristics, in that, its noise margins in both low and high states are large.

A CMOS inverter contains a PMOS and a NMOS transistor connected at the drain and gate terminals, a supply voltage VDD at the PMOS source terminal, and a ground connected at the NMOS source terminal, were VIN is connected to the gate terminals and VOUT is connected to the drain terminals.(given in diagram). It is important to notice that the CMOS does not contain any resistors, which makes it more power efficient that a regular resistor-MOSFET inverter. As the voltage at the input of the CMOS device varies between 0 and VDD, the state of the NMOS and PMOS varies accordingly. If we model each transistor as a simple switch activated by VIN, the inverter's operations can be seen very easily:

MOSFET	Condition on MOSFET	State of MOSFET
NMOS	Vgs<Vtn	OFF
NMOS	Vgs>Vtn	ON
PMOS	Vsg<Vtp	OFF
PMOS	Vsg>Vtp	ON

The table given, explains when the each transistor is turning on and off. When VIN is low, the NMOS is "off", while the PMOS stays "on": instantly charging VOUT to

logic high. When Vin is high, the NMOS is "on and the PMOS is "off": taking the voltage at VOUT to logic low.

Inverter DC Characteristics:

Before we study the DC characteristics of the inverter we should examine the ideal characteristics of inverter which is shown below. The characteristic shows that when input is zero output will high and vice versa.

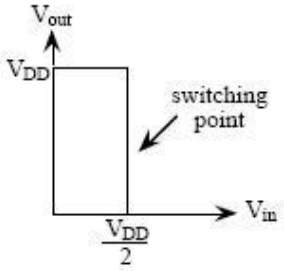


Figure 5: Ideal Characteristics of an Inverter

The actual characteristics is also given here for the reference. Here we have shown the status of both NMOS and PMOS transistor in all the regions of the characteristics.

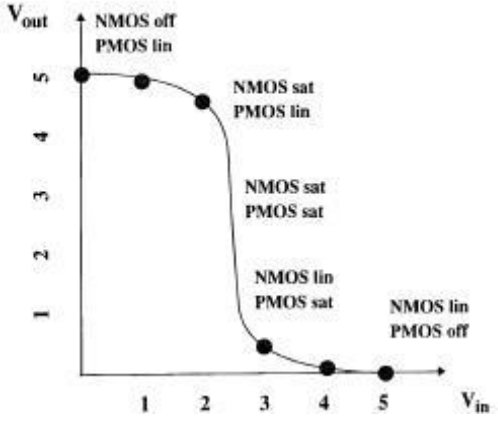


Figure 6: Actual Characteristics of an Inverter

Graphical Derivation of Inverter DC Characteristics:

The actual characteristics is drawn by plotting the values of output voltage for different values of the input voltage. We can also draw the characteristics, starting with the VI characteristics of PMOS and NMOS characteristics.

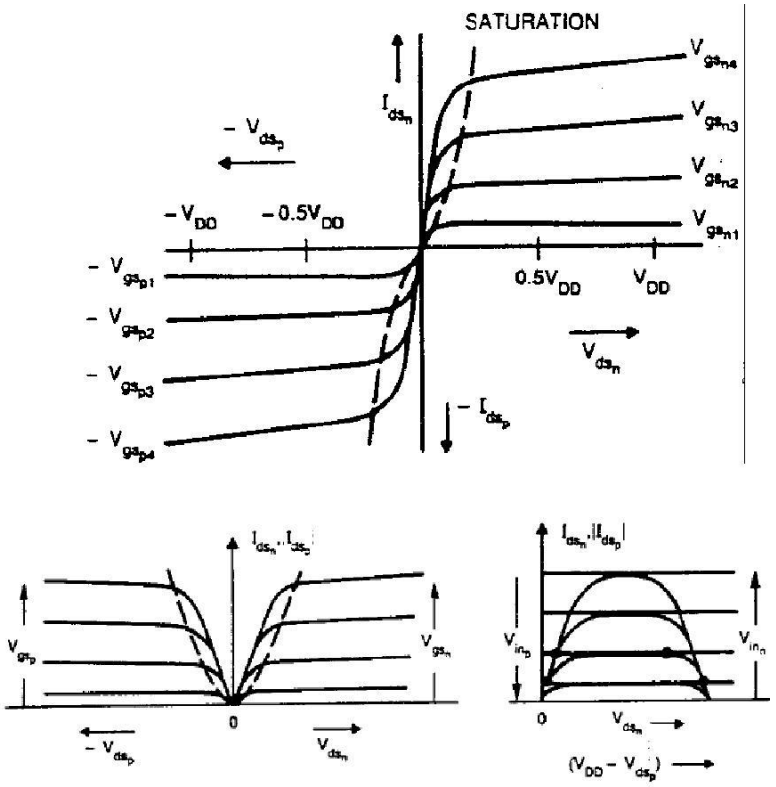


Figure 7a,b,c: Graphical Derivation of DC Characteristics

The characteristics given in figure 7a is the vi characteristics of the NMOS and PMOS characteristics (plot of Id vs. Vds). The figure 7b shows the values of drain current of PMOS transistor is taken to the positive side the current axis. This is done by taking the absolute value of the current. By superimposing both characteristics it leads to figure 7c. the actual characteristics may be now determined by the points of common Vgs intersection as shown in figure 7d.

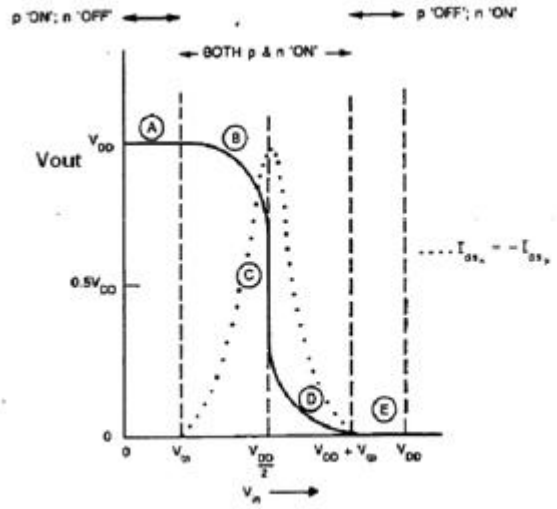


Figure 7d: CMOS Inverter DC Characteristics

Figure 7d shows five regions namely region A, B, C, D & E. also we have shown a

dotted curve which is the current that is drawn by the inverter.

Region A:

The output in this region is High because the P device is OFF and n device is ON. In region A, NMOS is cutoff region and PMOS is on, therefore output is logic high. We can analyze the inverter when it is in region B. the analysis is given below:

Region B:

The equivalent circuit of the inverter when it is region B is given below.

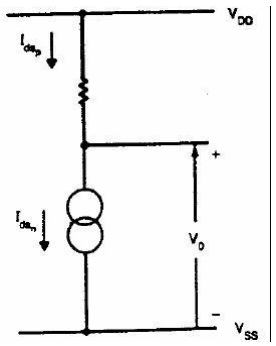


Figure 8: Equivalent circuit in Region B

In this region PMOS will be in linear region and NMOS is in saturation region.

The expression for the NMOS current is $I_{ds_n} = \beta_n \frac{[V_{in} - V_{t_n}]^2}{2}$,

The expression for the PMOS current is $I_{ds_p} = -\beta_p \left[(V_{in} - V_{DD} - V_{t_p})(V_O - V_{DD}) - \frac{1}{2}(V_O - V_{DD})^2 \right]$

The expression for the voltage Vo can be written as

$$V_O = (V_{in} - V_{t_p}) + \left[(V_{in} - V_{t_p})^2 - 2\left(V_{in} - \frac{V_{DD}}{2} - V_{t_p} \right) V_{DD} - \frac{\beta_n}{\beta_p}(V_{in} - V_{t_n})^2 \right]^{1/2}$$

Region C:

The equivalent circuit of CMOS inverter when it is in region C is given here. Both n and p transistors are in saturation region, we can equate both the currents and we can obain the expression for the mid point voltage or switching point voltage of a inverter. The corresponding equations are as follows:

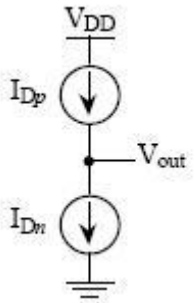


Figure 9: Equivalent circuit in Region C

The corresponding equations are as follows:

$$I_{dsp} = \frac{1}{2}\beta_p(V_{in} - V_{DD} - V_{tp})^2$$

$$I_{dsn} = \frac{1}{2}\beta_n(V_{in} - V_{tn})^2$$

By equating both the currents, we can obtain the expression for the switching point voltage as,

$$V_{in} = \frac{V_{DD} + V_{tp} + V_{tn}\sqrt{\frac{\beta_n}{\beta_p}}}{1 + \sqrt{\frac{\beta_n}{\beta_p}}}.$$

Region D: the equivalent circuit for region D is given in the figure below.

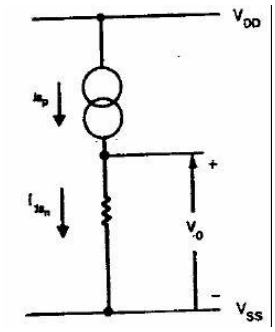


Figure 10: equivalent circuit in region D

We can apply the same analysis what we did for region B and C and we can obtain the expression for output voltage.

Region E:

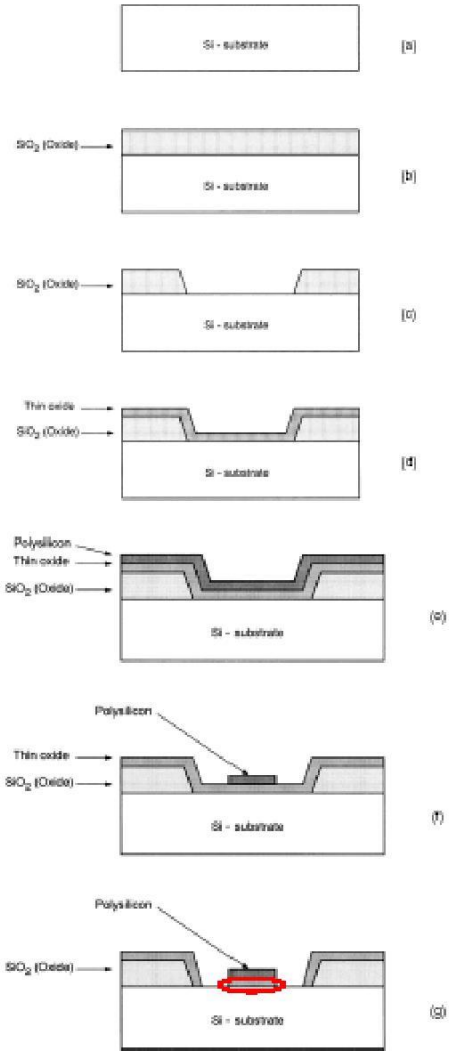
The output in this region is zero because the P device is OFF and n device is ON.

OR

Q: 2 a

NMOS Fabrication:

The process starts with the oxidation of the silicon substrate (Fig. 9(a)), in which a relatively thick silicon dioxide layer, also called field oxide, is created on the surface (Fig. 9(b)). Then, the field oxide is selectively etched to expose the silicon surface on which the MOS transistor will be created (Fig. 9(c)). Following this step, the surface is covered with a thin, high-quality oxide layer, which will eventually form the gate oxide of the MOS transistor (Fig. 9(d)). On top of the thin oxide, a layer of polysilicon (polycrystalline silicon) is deposited (Fig. 9(e)). Polysilicon is used both as gate electrode material for MOS transistors and also as an interconnect medium in silicon integrated circuits. Undoped polysilicon has relatively high resistivity. The resistivity of polysilicon can be reduced, however, by doping it with impurity atoms


Si - substrate
(a)
SiO_2 (Oxide)
Si - substrate
(b)
SiO_2 (Oxide)
Si - substrate
(c)
Thin oxide
SiO_2 (Oxide)
Si - substrate
(d)
Polysilicon
Thin oxide
SiO_2 (Oxide)
Si - substrate
(e)
Polysilicon
Thin oxide
SiO_2 (Oxide)
Si - substrate
(f)
Polysilicon
SiO_2 (Oxide)
Si - substrate
(g)

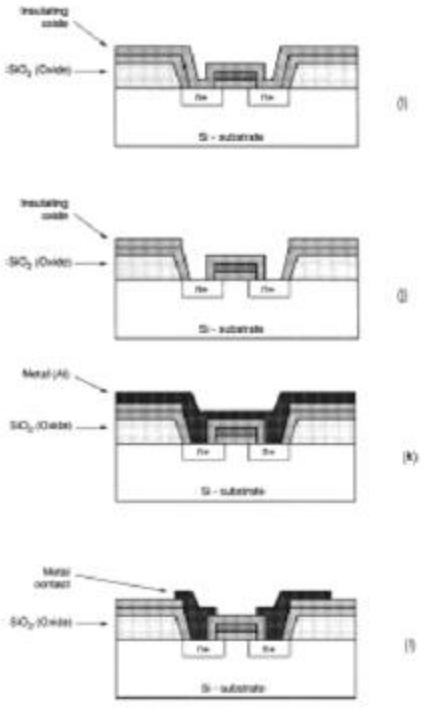


Figure 9 NMOS Fabrication process steps

After deposition, the polysilicon layer is patterned and etched to form the interconnects and the MOS transistor gates (Fig. 9(f)). The thin gate oxide not covered by polysilicon is also etched away, which exposes the bare silicon surface on which the source and drain junctions are to be formed (Fig. 9(g)). The entire silicon surface is then doped with a high concentration of impurities, either through diffusion or ion implantation (in this case with donor atoms to produce n-type doping). Figure 9(h) shows that the doping penetrates the exposed areas on the silicon surface, ultimately creating two n-type regions (source and drain junctions) in the p-type substrate. The impurity doping also penetrates the polysilicon on the surface, reducing its resistivity. Note that the polysilicon gate, which is patterned before doping actually defines the precise location of the channel region and, hence, the location of the source and the drain regions. Since this procedure allows very precise positioning of the two regions relative to the gate, it is also called the self-aligned

process. Once the source and drain regions are completed, the entire surface is again covered with an insulating layer of silicon dioxide (Fig. 9 (i)). The insulating oxide layer is then patterned in order to provide contact windows for the drain and source junctions (Fig. 9 (j)). The surface is covered with evaporated aluminum which will form the interconnects (Fig. 9 (k)). Finally, the metal layer is patterned and etched, completing the interconnection of the MOS transistors on the surface (Fig. 9 (l)). Usually, a second (and third) layer of metallic interconnect can also be added on top of this structure by creating another insulating oxide layer, cutting contact (via) holes, depositing, and patterning the metal.

Q: 2 b.

I) Channel length modulation:

The channel length of the MOSFET is changed due to the change in the drain to source voltage. This effect is called as the channel length modulation. The effective channel length & the value of the drain current considering channel length modulation into effect is given by,

Change in v_{DS} in saturation, implies no change in corresponding i_D and hence infinite resistance in saturation. This is because of the assumption that once channel is pinched off, further increase in v_{DS} have no effect on the channel's shape. But, in practice, as v_{DS} is increased, the channel pinch-off point is moved slightly away from the drain, toward the source.

Channel Length Modulation: With an increase in v_{DS}, the channel length decreases from L to L- L, but voltage drop across it remains the same, and the additional drop will appear across the depletion region between the end of the channel and the drain

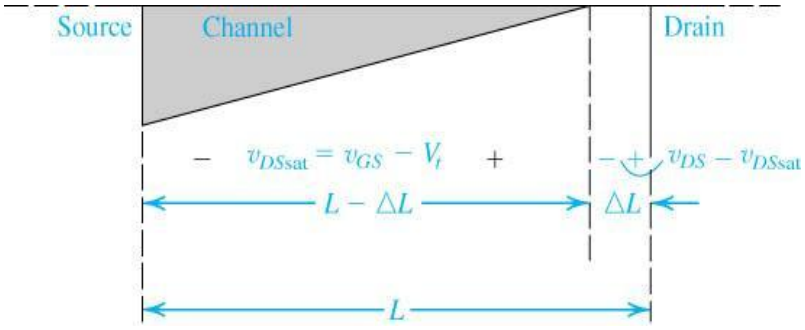


Figure: Channel length modulation

To account for the dependence of i_D on v_{DS} in saturation, replace L with L- L in the i_D Eq. λ is a process technology parameter and now the expression for i_D becomes,

$$i_D = \frac{1}{2} k_n' \frac{W}{L} (v_{GS} - V_t)^2 (1 + \lambda v_{DS})$$

$$I_{ds} = \frac{\beta}{2}\{(V_{gs} - V_t)^2(1 + \lambda V_{ds})\}$$

$$L_{eff} = L - \sqrt{2\varepsilon_o \frac{\varepsilon_{Si}}{qN}(V_{ds} - [V_{gs} - V_t])}.$$

Where λ is the channel length modulation factor.

ii) **Noise Margin:**

Noise margin is a parameter related to input output characteristics. It determines the allowable noise voltage on the input so that the output is not affected.

We will specify it in terms of two things:

LOW noise margin
HIGH noise margin

LOW noise margin: is defined as the difference in magnitude between the maximum Low output voltage of the driving gate and the maximum input Low voltage recognized by the driven gate.

$NM_L=|V_{ILmax} - V_{OLmax}|$

HIGH noise margin: is defined difference in magnitude between minimum High output voltage of the driving gate and minimum input High voltage recognized by the receiving gate.

$\mathbf{NM_H=|Voh_{min} - V_{IHmin}|}$

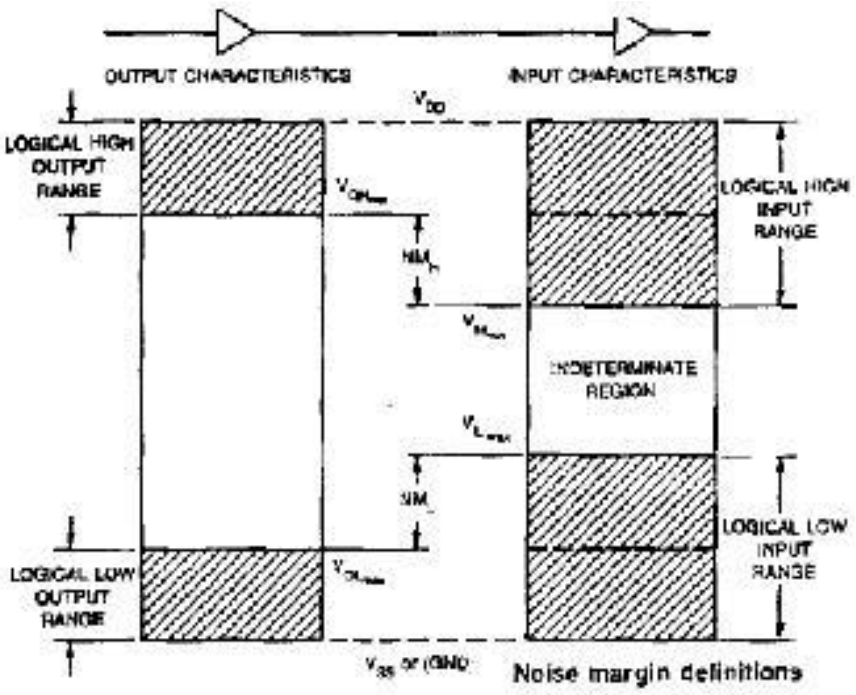


Figure 12: noise margin definitions

Figure shows how exactly we can find the noise margin for the input and output.

We can also find the noise margin of a CMOS inverter. The following figure gives the idea of calculating the noise margin.

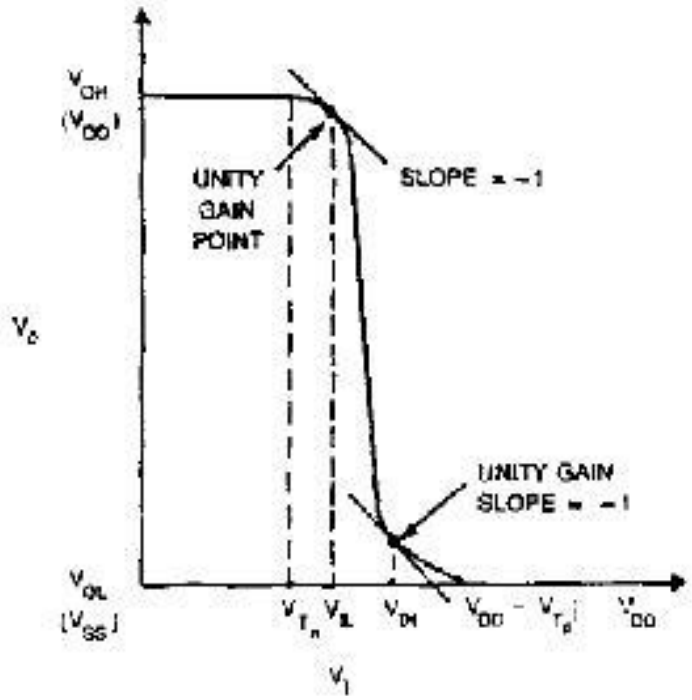


Figure 13: CMOS inverter noise margins

Q:3a

Basic Structure

A CMOS logic circuit is in effect an extension, or a generalization, of the CMOS inverter:

The CMOS logic gate consists of two networks: the pull-down network (PDN) constructed of NMOS transistors, and the pull-up network (PUN) constructed of PMOS transistors (see Fig. 8). The two networks are operated by the input variables, in a complementary fashion. Thus, for the three-input gate represented in Fig. 8, the PDN will conduct for all input combinations that require a low-output (Y= 0) and will then pull the output node down to ground, causing a zero voltage to appear at the output, V_Y= 0. Simultaneously, the PUN will be off, and no direct dc path will exist between V_{DD} and ground. On the other hand, all input combinations that call for a high output (Y=1) will cause the PUN to conduct, and the PUN will then pull the output node up to V_{DD}, establishing an output voltage $V_Y = V_{DD}$ Simultaneously, the PDN will be cut off, an again, no dc current path between V_{DD} and ground will exist in the circuit.

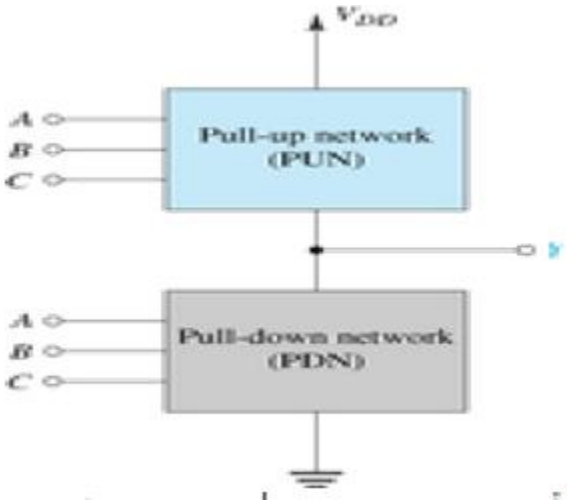


Figure 8 Representation of a three-input CMOS logic gate. The PUN comprises PMOS transistors, and the PDN comprises NMOS transistors

Since the PDN comprises NMOS transistors, and since an NMOS transistor conducts when the signal at its gate is high, the PDN is activated (i.e., conducts) when the inputs are high. In a dual manner, the PUN comprises PMOS transistors, and a PMOS transistor conducts when the input signal at its gate is low; thus the PUN is activated when the inputs are low.

The PDN and the PUN each utilizes devices in parallel to form an OR function, and devices in series to form an AND function. For the circuit in Fig. 9(a), we observe that Q_A will conduct when *A* is high(V_A=V_{DD}) and will then pull the output node down to ground (V_Y=0V, Y = 0). Similarly, Q_B conducts and pulls *Y* down when *B* is high. Thus Y will be low when A is high or B is high, which can be expressed as

$$\overline{Y} = A + B$$

or equivalently

$$Y = \overline{A + B}$$

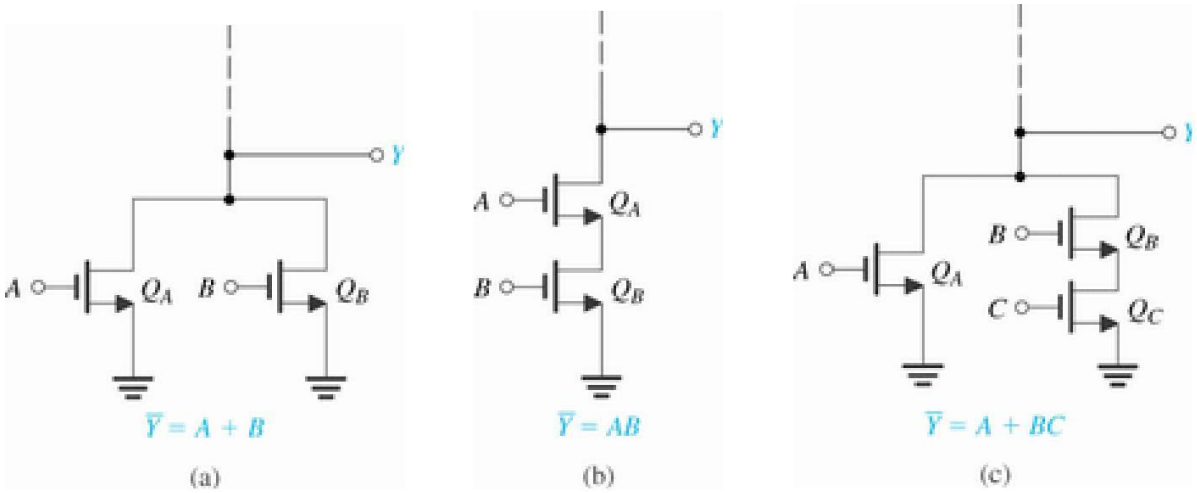


Fig : pull down networks

The PDN in Fig. 9(b) will conduct only when *A* and *B* are both high simultaneously. Thus *Y* will be low when *A* is high *and B* is high,

$$\overline{Y} = AB$$

or equivalently

$$Y = \overline{AB}$$

As a final example, the PDN in Fig. 9(c) will conduct and cause Y to be 0 when A is high *or* when *B and C* are both high, thus

$$\overline{Y} = A + BC$$

or equivalently

$$Y = \overline{A + BC}$$

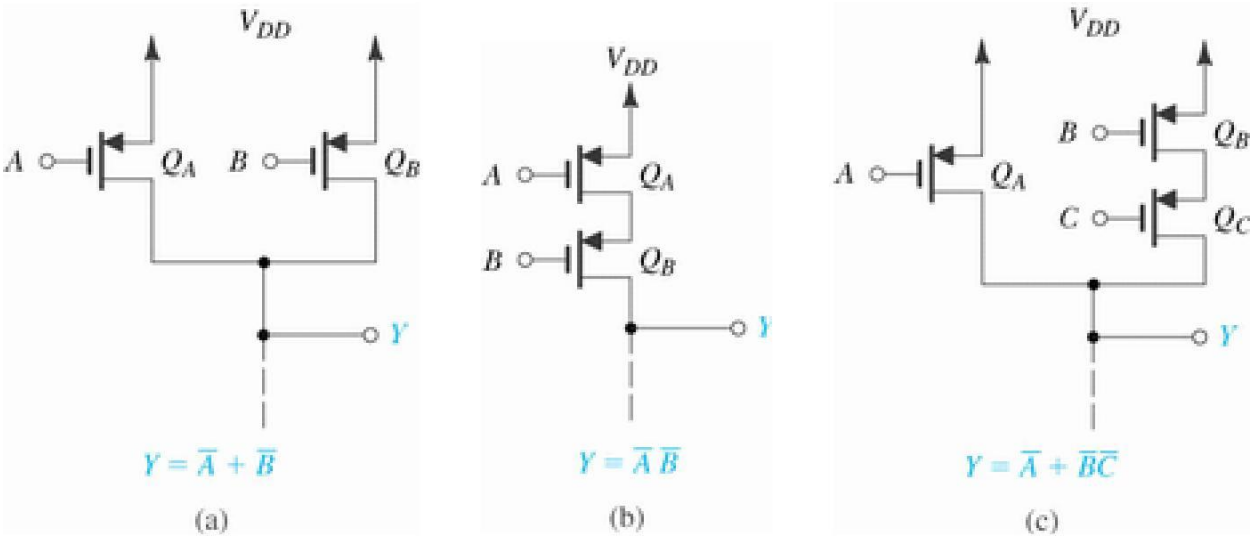


Figure: Examples of pull-up networks.

Next consider the PUN examples shown in Fig. 10. The PUN in Fig. 10(a) will conduct and pull Y up to V_{DD} (*Y* =1) when *A* is low *or B* is low, thus

$$Y = \overline{A} + \overline{B}$$

The PUN in Fig. 10(b) will conduct and produce a high output ($V_Y = V_{DD}$, Y= 1) only when *A and B* are both low, thus

$$Y = \overline{A}\,\overline{B}$$

Finally, the PUN in Fig. 10(c) will conduct and cause Y to be high (logic 1) if *A* is low *or* if *B and C* are both low, thus

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

With CMOS there are two types of diffusion: n-type is drawn in green and p-type in brown. These are on the same layers in the chip and must not meet. In fact, the method of fabrication required that they be kept relatively far apart.

Modern CMOS processes usually support more than one layer of metal. Two are common and three or more are often available.

Actually, these conventions for colors are not universal; in particular, industrial (rather than academic) systems tend to use red for diffusion and green for polysilicon. Moreover, a shortage of colored pens normally means that both types of diffusion in CMOS are colored green and the

polarity indicated by drawing a circle round p-type transistors or simply inferred from the context. Colorings for multiple layers of metal are even less standard.

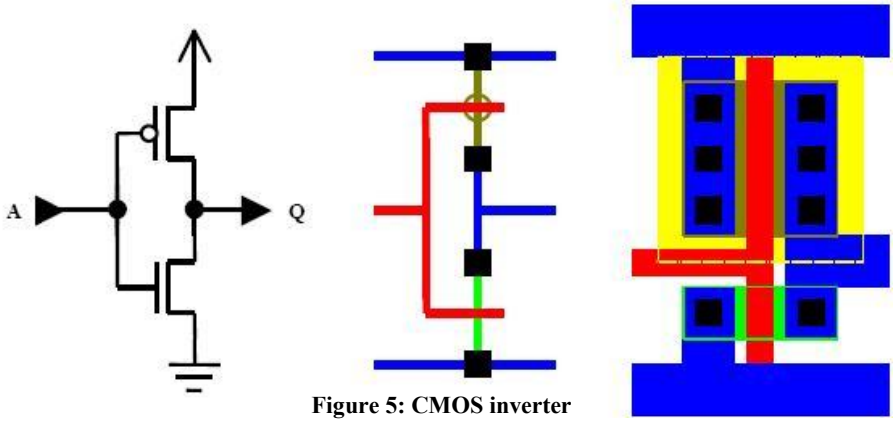
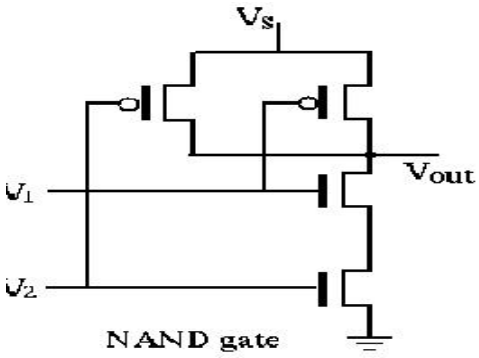


Figure 5: CMOS inverter

Figure 5 shows the schematic, stick diagram and corresponding layout of CMOS inverter
Q: 3b.



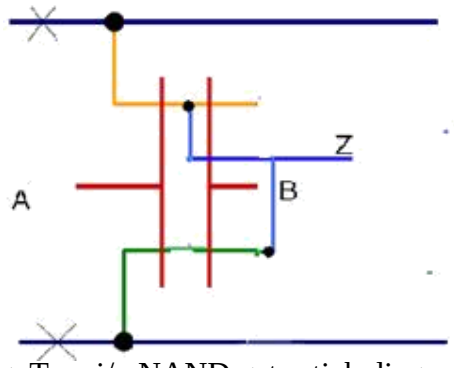
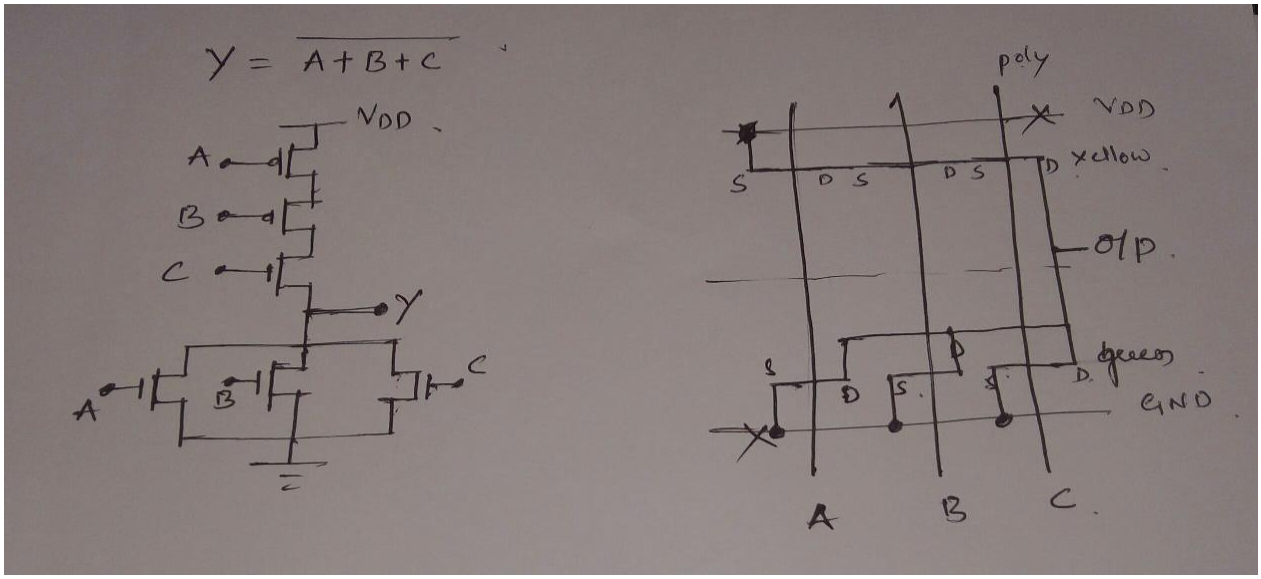


Fig. Two i/p NAND gate stick diagram



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

Q:3b.

Via is used to connect higher level metals from metal1 connection. The cross section and layout view given figure 13 explain via in a better way.

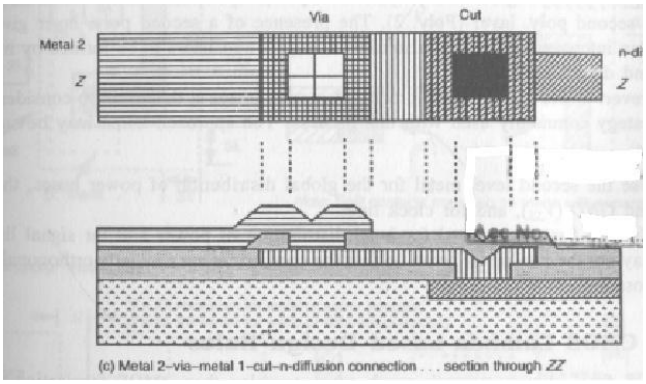


Figure : cross section showing the contact cut and via

Figure shows the design rules for contact cuts and Vias. The design rule for contact is minimum 2λx2λ and same is applicable for a Via.

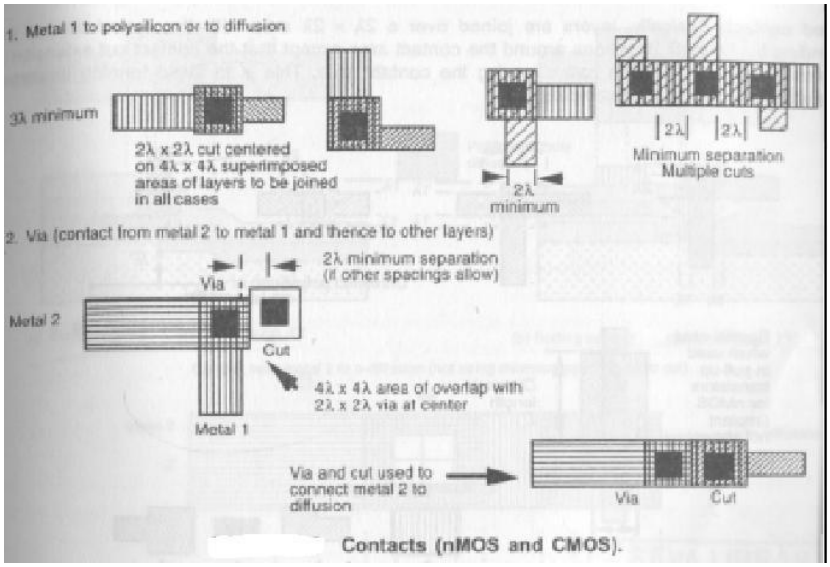


Figure Design rules for contact cuts and vias

Buried contact: The contact cut is made down each layer to be joined and it is shown in figure

Figure : Buried contact

Butting contact: The layers are butted together in such a way the two contact cuts become contiguous. We can better under the butting contact from figure.

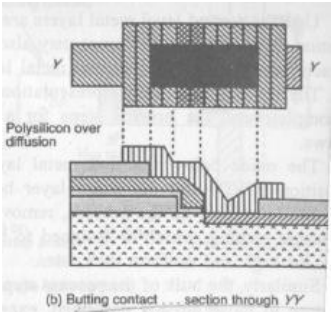
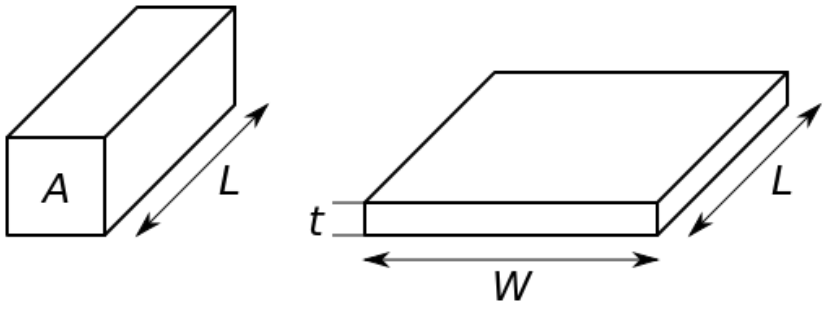


Figure : Butting contact

Q:4a

Sheet resistance heet resistance is applicable to two-dimensional systems in which thin films are considered two-dimensional entities. When the term sheet resistance is used, it is implied that the current is along the plane of the sheet, not perpendicular to it.

In a regular three-dimensional conductor, the resistance can be written as



$$R = \rho L/A = \rho L/Wt$$

where ρ is the resistivity, A is the cross-sectional area, and L is the length. The cross-sectional area can be split into the width W and the sheet thickness t .

Upon combining the resistivity with the thickness, the resistance can then be written as

$$R = \rho L/Wt = R_s L/W$$

where Rs is the sheet resistance. If the film thickness is known, the bulk resistivity (in Ω·cm) can be calculated by multiplying the sheet resistance by the film thickness in cm:

If L=W, R=Rs

The Sheet resistance Rs of nMOS is 10kohm

The Rs of pMOS is 2.5X10kohm for 5um Technology

Q:4b.

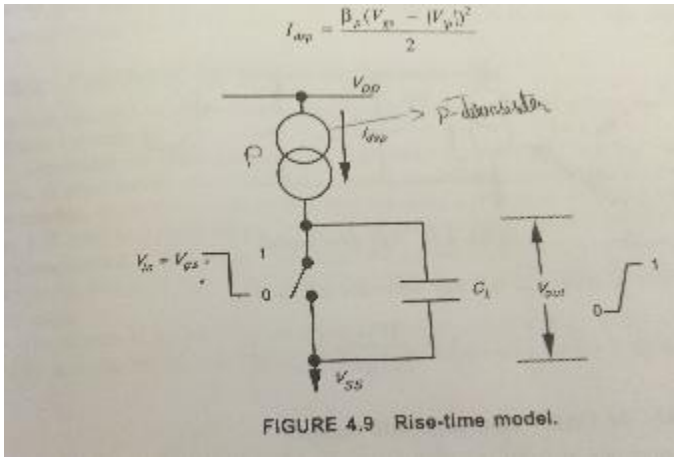


FIGURE 4.9 Rise-time model.

This current charges C_L and, since its magnitude is approximately constant, we hav

$$V_{out} = \frac{I_{dsp}t}{C_L}$$

Substituting for I_{dsp} and rearranging we have

$$t = \frac{2C_L V_{out}}{\beta_p (V_{gs} - |V_{tp}|)^2}$$

$$\tau_r \doteq \frac{3C_L}{\beta_p V_{DD}}$$

onably well with a more detailed analysis in which the chargin ectly, into two parts: (1) saturation and (2) resistive region of

timation

applied to the discharge of C_L through the n-transistor. The c en as Figure 4.10.

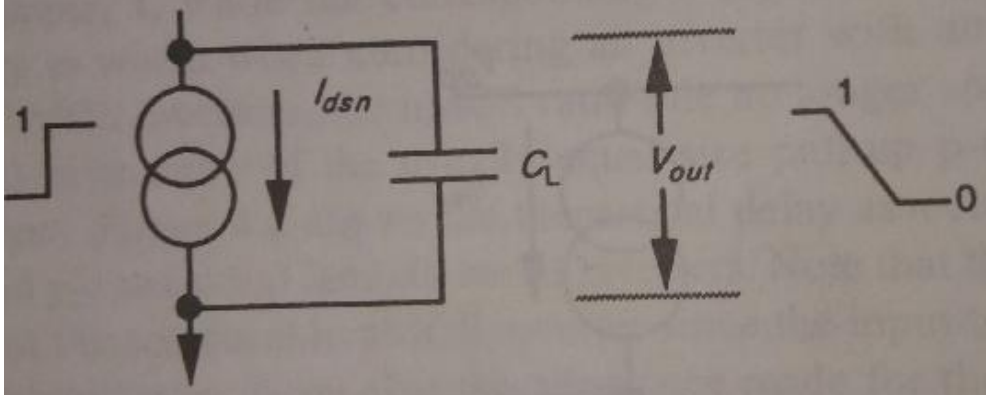
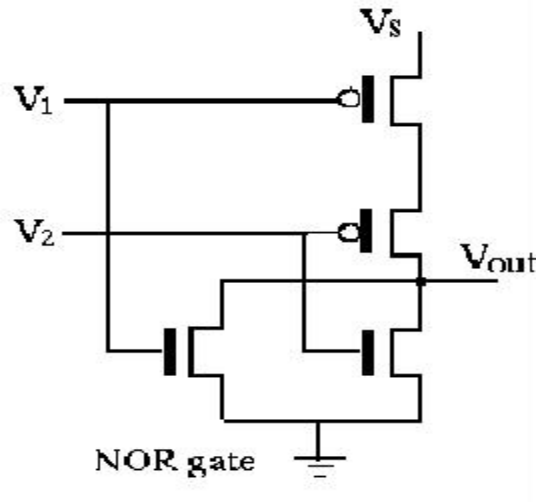


FIGURE 4.10 Fall-time model.

sumptions we may write for fall-time:

$$\tau_f \doteq \frac{3C_L}{\beta_n V_{DD}}$$

Q: 4c



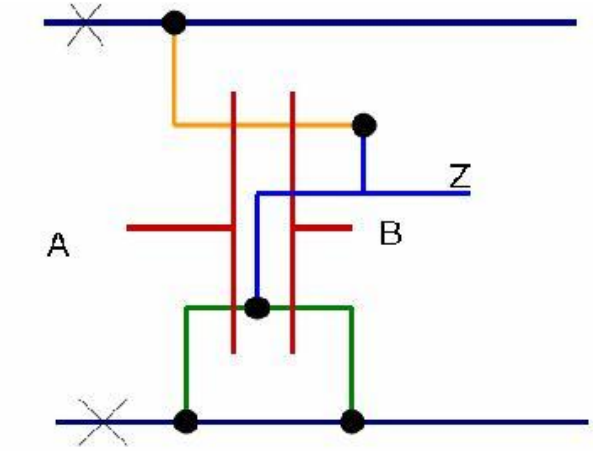
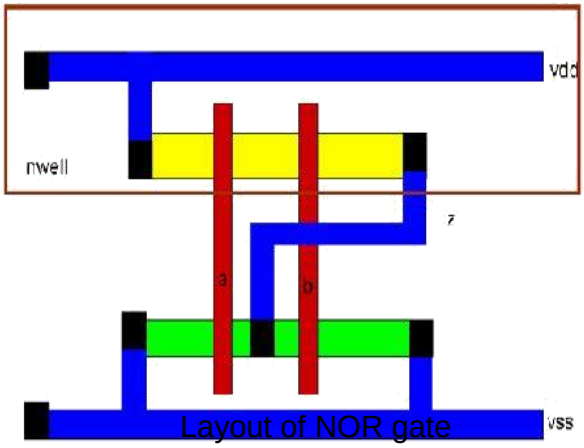


Figure 7:Stick diagram of nor gate



Layout of NOR gate

Q:5a.

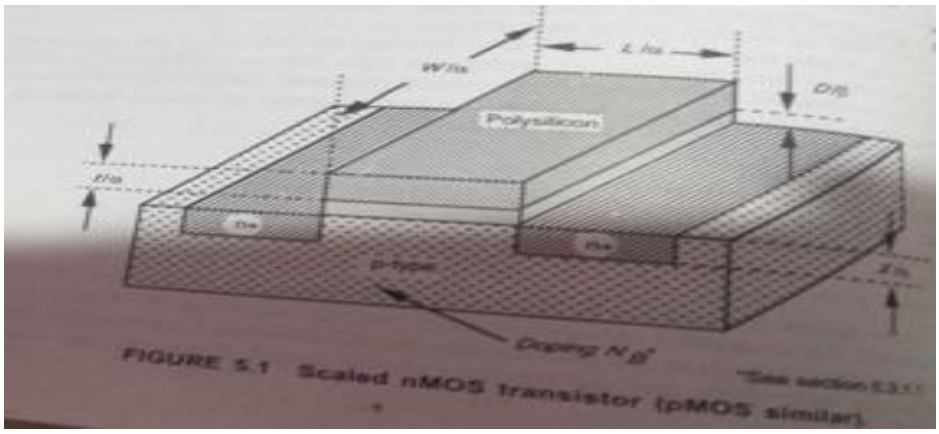
The process of reducing the horizontal and vertical dimension of the transistor is called Scaling.The device is scaled by a factor S, This S is quantity grates than 1.

There are two types of Scaling techniques

1. Constant Field Scaling
2. Constant Voltage scaling

Two scaling factors 1/α and 1/β are used, 1/β is the scaling factor for supply voltage and gate oxide thickness. 1/α scaling factor for linear dimensions both horizontal and vertical dimensions.

- For Constant Field Scaling $\beta = \alpha$
- Constant Voltage scaling $\beta = 1$



Parameters	Description	General (Combined V and Dimension)	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_o (or C_{ox})	Gate capacitance per unit area	β	α	1
C_g	Gate capacitance	β/α^2	$1/\alpha$	$1/\alpha^2$
C_x	Parsitic 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

Q:5b.

7.1.1 Some Problems

Some of the problems associated with VLSI design are:

1. How to design large complex systems in a reasonable time and with reasonable effort. This is a problem shared with other approaches to system design.
2. The nature of architectures best suited to take full advantage of VLSI and the technology.
3. The testability of large/complex systems once implemented in silicon.

Problems 1 and 3 are greatly reduced if two aspects of standard practice are accepted:

- Approach the design in a top-down manner and with adequate computer-aided tools to do the job. Partition the system sensibly, aiming for simple interconnection between subsystems and high regularity within subsystems. Generate and then verify each section of the design.
- Design testability into the system from the outset and be prepared to devote a significant proportion (e.g. up to 30%) of the total chip area to test and diagnostic facilities.

These problems are the subject of considerable research and development activity at this time.

Q:5c

Nature of the bus architecture linking the subunits is discussed below. Some of the possibilities are:

One bus architecture:

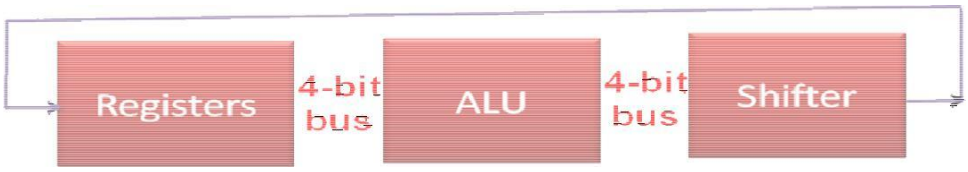


Figure : One bus architecture Sequence:

1. 1st operand from registers to ALU. Operand is stored there.
2. 2nd operand from register to ALU and added.
3. Result is passed through shifter and stored in the register

Two bus architecture:

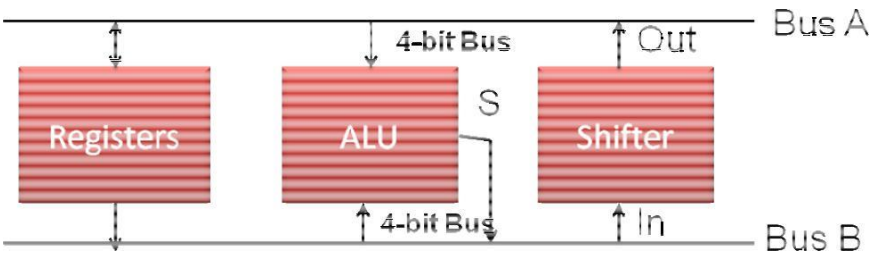


Figure : Two bus architecture Sequence:

1. Two operands (A & B) are sent from register(s) to ALU & are operated upon, result S in ALU.
2. Result is passed through the shifter & stored in registers.

Three bus architecture:

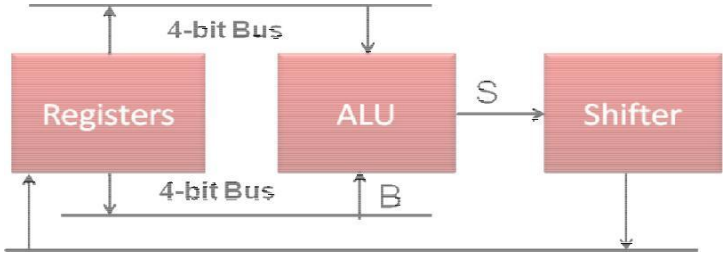


Figure : Three bus architecture Sequence:

Two operands (A & B) are sent from registers, operated upon, and shifted result (S) returned to another register, all in same clock period.

In pursuing this design exercise, it was decided to implement the structure with a 2 – bus architecture. A tentative floor plan of the proposed design which includes some form of interface to the parent system data bus is shown in figure 6.7.

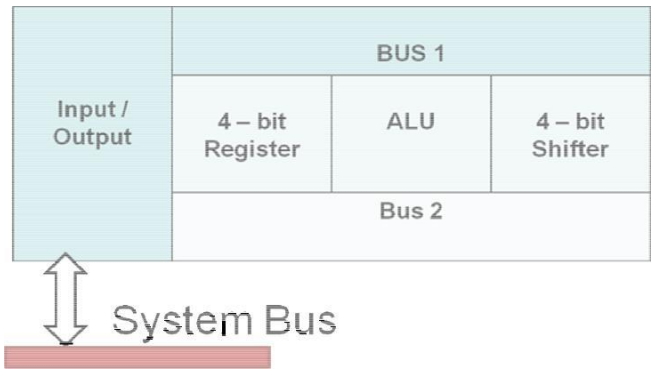


Figure 6: Tentative floor plan for 4 – bit datapath

Q:6a

Design of a 4-bit adder:

The truth table of binary adder is as shown in table 6.1

Inputs			Outputs	
A_k	B_k	C_{k-1}	S_k	C_k
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

As seen from the table any column k there will be three inputs namely A_k , B_k as present input number and C_{k-1} as the previous carry. It can also be seen that there are two outputs sum S_k and carry C_k.

From the table one form of the equation is:

Sum $S_k = H_k C_{k-1}' + H_k' C_{k-1}$

New carry $C_k = A_k B_k + H_k c_{k-1}$

Where

Half sum $H_k = A_k' B_k + A_k B_k'$

Adder element requirements

Table 6.1 reveals that the adder requirement may be stated as:

If $A_k = B_k$ then $S_k = C_{k-1}$

Else $S_k = C_{k-1}'$

And for the carry C_k

If $A_k = B_k$ then $C_k = A_k = B_k$

Else $C_k = C_{k-1}$

Thus the standard adder element for 1-bit is as shown in the figure 6.11.

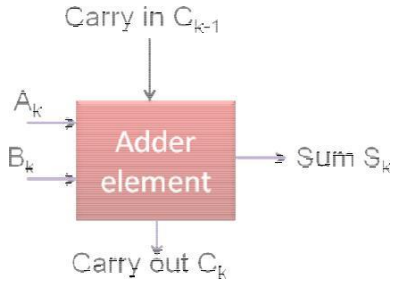


Figure : Adder element

Implementing ALU functions with an adder:

An ALU must be able to add and subtract two binary numbers, perform logical operations such as And, Or and Equality (Ex-or) functions. Subtraction can be performed by taking 2's complement of the negative number and perform the further addition. It is desirable to keep the architecture as simple as possible, and also see that the adder performs the logical operations also. Hence let us examine the possibility.

The adder equations are:

Sum $S_k = H_k C_{k-1}' + H_k' C_{k-1}$

New carry $C_k = A_k B_k + H_k C_{k-1}$

Where

Half sum $H_k = A_k' B_k + A_k B_k'$

Let us consider the sum output, if the previous carry is at logical 0, then

$S_k = H_k . 1 + H_k' . 0$

$S_k = H_k = A_k' B_k + A_k B_k'$ – An Ex-or operation Now, if C_{k-1} is logically 1, then

$S_k = H_k . 0 + H_k' . 1$

$S_k = H_k$' – An Ex-Nor operation

Next, consider the carry output of each element, first C_{k-1} is held at logical 0, then $C_k = A_kB_k + H_k \cdot 0$

$C_k = A_kB_k$ - An And operation

Now if C_{k-1} is at logical 1, then

$C_k = A_kB_k + H_k \cdot 1$

On solving $C_k = A_k + B_k$ - An Or operation

The adder element implementing both the arithmetic and logical functions can be implemented as shown in the figure 6.12.

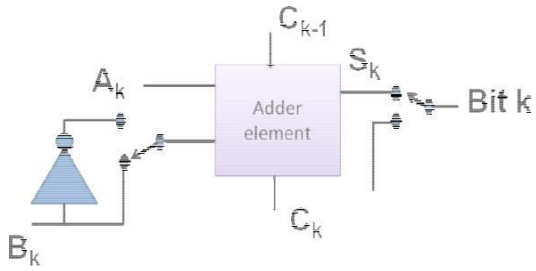


Figure: 1-bit adder block

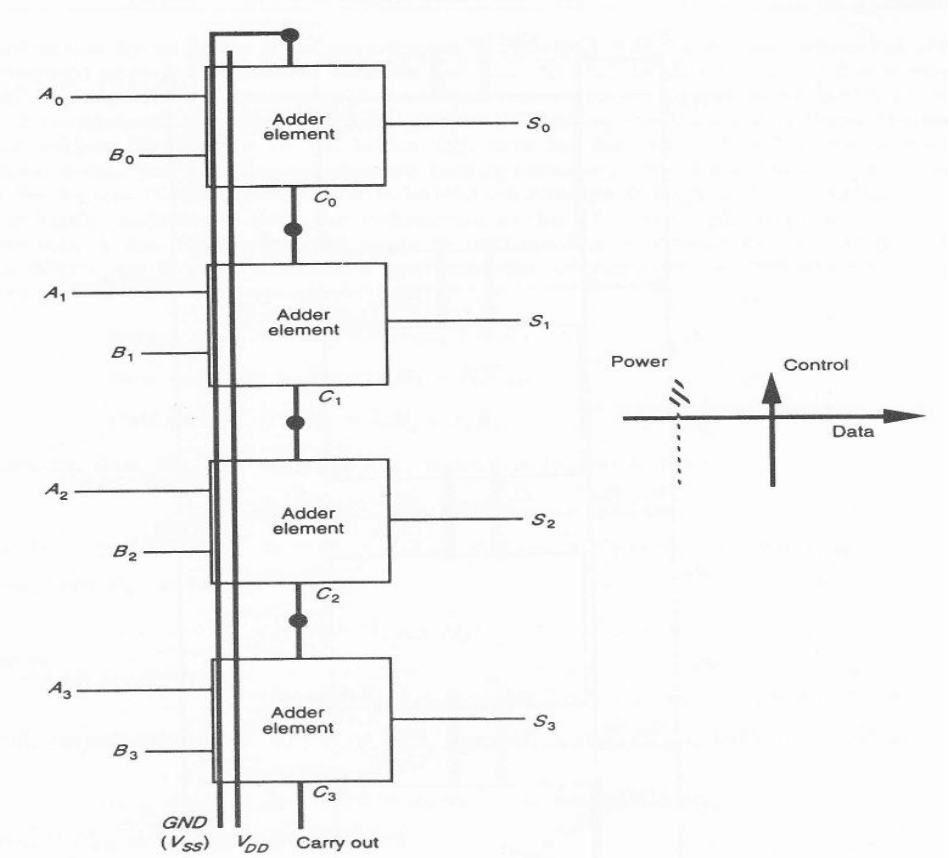


FIGURE 8.11(a) 4-bit adder.

Q:6b

Manchester carry – chain

This implementation can be very per formant (20 transistors) depending on the way the XOR function is built. The carry propagation of the carry is controlled by the output of the XOR gate. The generation of the carry is directly made by the function at the bottom. When both input signals are 1, then the inverse output carry is 0.

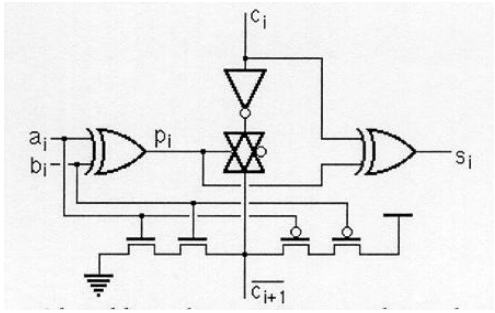


Figure-: An adder with propagation signal controlling the pass-gate

In the schematic of Figure 6.12, the carry passes through a complete transmission gate. If the carry path is precharged to VDD, the transmission gate is then reduced to a simple NMOS transistor. In the same way the PMOS transistors of the carry generation is removed. One gets a Manchester cell.

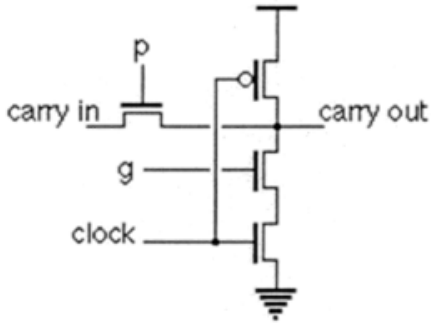


Figure-: The Manchester cell

The Manchester cell is very fast, but a large set of such cascaded cells would be slow. This is due to the distributed RC effect and the body effect making the propagation time grow with the square of the number of cells. Practically, an inverter is added every four cells, like in Figure 6

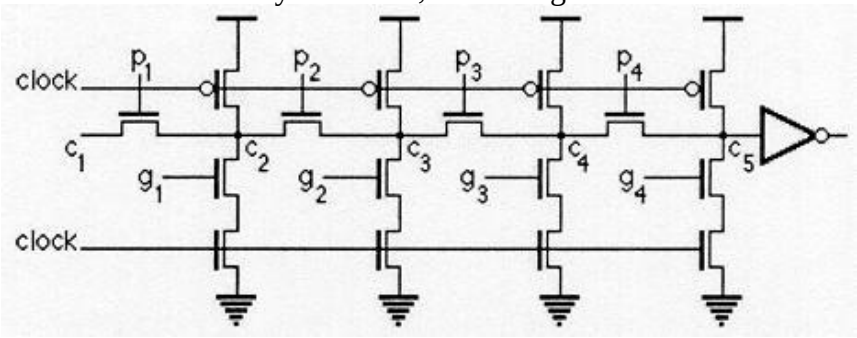


Figure-**:** The Manchester carry cell

Q:6c

The Carry-Select Adder

This type of adder is not as fast as the Carry Look Ahead (CLA) presented in a next section. However, despite its bigger amount of hardware needed, it has an interesting design concept. The Carry Select principle requires two identical parallel adders that are partitioned into four-bit groups. Each group consists of the same design as that shown on Figure 6.18. The group generates a group carry. In the carry select adder, two sums are generated simultaneously. One sum assumes that the carry in is equal to one as the other assumes that the carry in is equal to zero. So that the predicted group carry is used to select one of the two sums.

It can be seen that the group carries logic increases rapidly when more high- order groups are added to the total adder length. This complexity can be

decreased, with a subsequent increase in the delay, by partitioning a long adder into sections, with four groups per section, similar to the CLA adder.

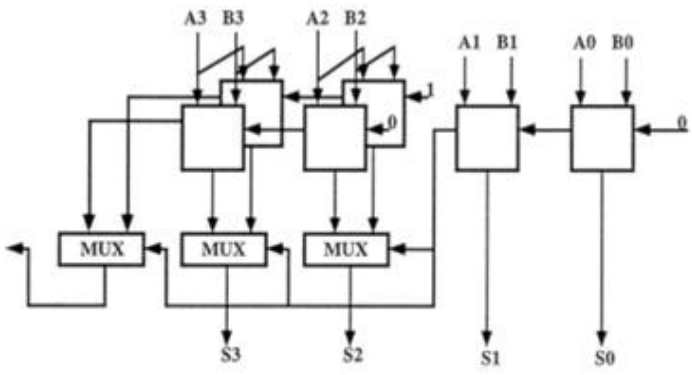


Figure-: The Carry Select adder Optimization of the carry select adder

- Computational time

$$T = K_1 n$$

- Dividing the adder into blocks with 2 parallel paths

$$T = K_1 n/2 + K_2$$

- For a n-bit adder of M-blocks and each block contains P adder cells in series $T = PK_1 + (M - 1) K_2$; $n = M.P$ minimum value for T is when $M=\sqrt{(K_1 n / K_1)}$

Q:7a

Guidelines may be set out as follows:

l. Define the requirements (properly and carefully)'
2. Partition the overall architecture into appropriate subsystems.
3. Consider communication paths carefully in order to develop sensible interrelationships between subsystems.
4. Draw a floor plan of how the system is to map onto the silicon (and alternate between 2, 3 and 4 as necessary).
5. Aim for regular structures so that design is largely a matter of replication.
6. Draw suitable (stick or symbolic) diagrams of the leaf-cells of the subsystems.

7. Convert each cell to a layout.
8. Carefully and thoroughly carry out a design rule check on each cell.
9. Simulate the performance of each cell/subsystem

The whole design process will be greatly assisted if considerable care is taken with:

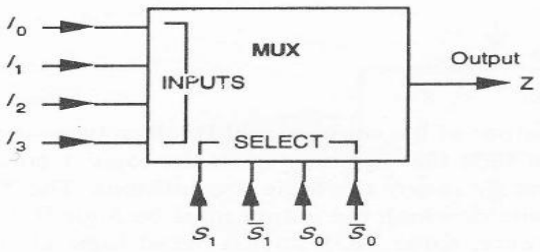
1. the partitioning of the system so that there are clean and clear subsystems with a minimum interdependence and complexity of interconnection between them.
2. The design simplification within subsystems so that architectures are adopted which allow the exploitation of a cellular design concept. This allows the system to be composed of relatively few standard cells which are replicated to form highly regular structures.

Q:7c

Multiplexers (Data Selectors)

The requirements and general arrangement of a four-way multiplexer a

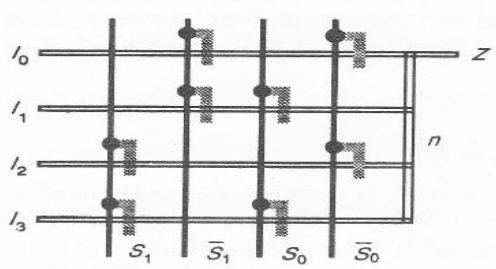
$$Z = I_0 . \overline{S}_1 . \overline{S}_0 + I_1 . \overline{S}_1 . S_0 + I_2 . S_1 . \overline{S}_0 + I_3 . S_1 . S_0$$



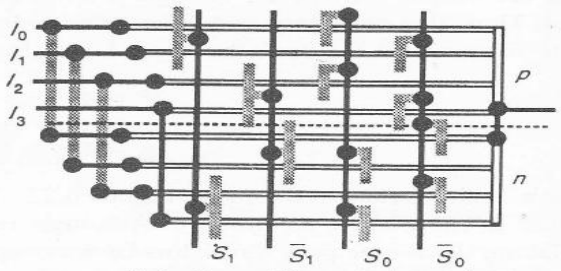
Truth table

S_1	S_0	Z
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

FIGURE 6.23 Selector logic circuit.



(a) nMOS switches



Note: V_{DD} and V_{SS} contacts not shown.

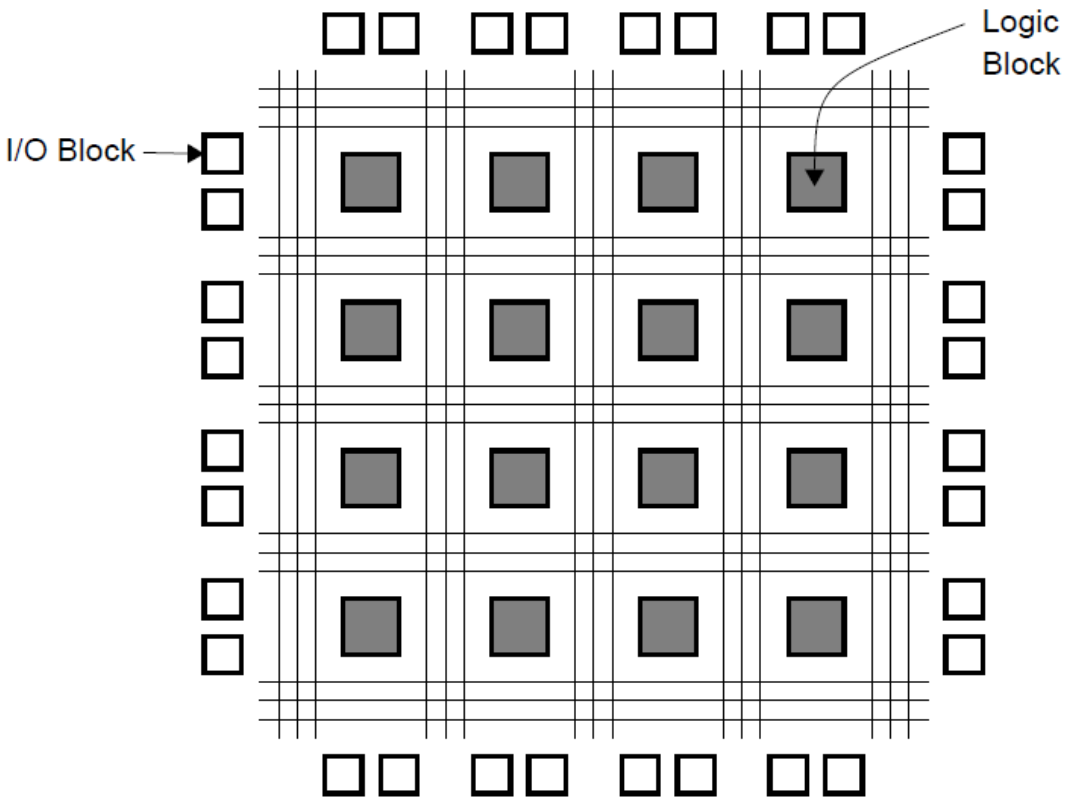
(b) Transmission gates (CMOS)

FIGURE 6.24 Switch logic implementations of a four-way multiplexer.

Q:8a

A **field-programmable gate array** (**FPGA**) is an integrated circuit designed to be configured by a customer or a designer after manufacturing – hence "field-programmable". The FPGA configuration is generally specified using a hardware description language (HDL), similar to that used for an application-specific integrated circuit (ASIC) (circuit diagrams were previously used to specify the configuration, as they were for ASICs, but this is increasingly rare).

FPGAs contain an array of programmable logic blocks, and a hierarchy of reconfigurable interconnects that allow the blocks to be "wired together", like many logic gates that can be inter-wired in different configurations. Logic blocks can be configured to perform complex combinatioal functions, or merely simple logic gates like AND and XOR. In most FPGAs, logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory



The FPGA is Field Programmable Gate Array. It is a type of device that is widely used in electronic circuits. FPGAs are semiconductor devices which contain programmable logic blocks and interconnection circuits. It can be programmed or reprogrammed to the required functionality after manufacturing.

The FPGA Architecture consists of three major components

- Programmable Logic Blocks, which implement logic functions
- Programmable Routing (interconnects), which implements functions
- I/O blocks, which are used to make off-chip connections

Programmable Logic Blocks

The programmable logic block provides basic computation and storage elements used in digital systems. A basic logic element consists of programmable combinational logic, a flip-flop, and some fast carry logic to reduce area and delay cost.

Modern FPGAs contain a heterogeneous mixture of different blocks like dedicated memory blocks, multiplexers. Configuration memory is used throughout the logic blocks to control the specific function of each element.

Programmable Routing

The programmable routing establishes a connection between logic blocks and Input/Output blocks to complete a user-defined design unit.

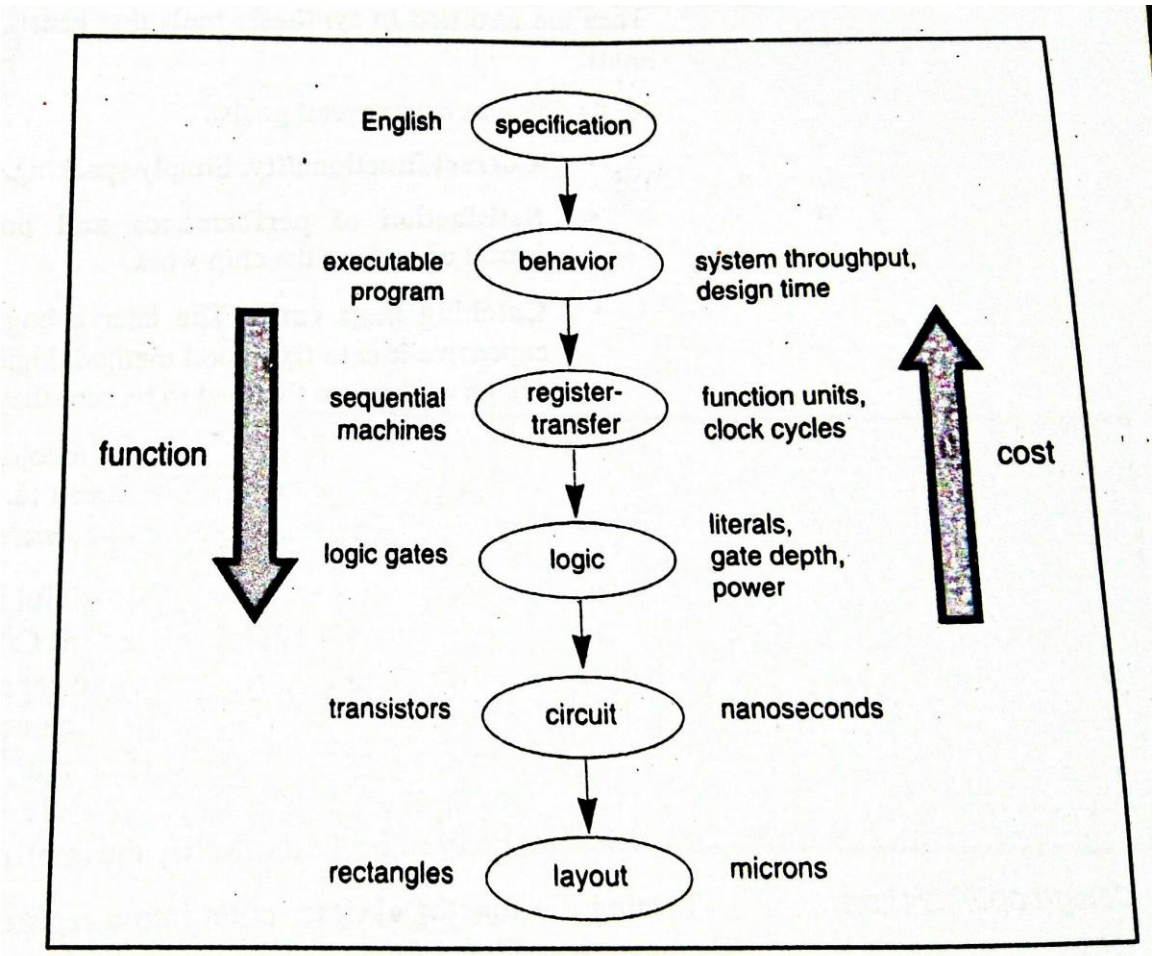
It consists of multiplexers pass transistors and tri-state buffers. Pass transistors and multiplexers are used in a logic cluster to connect the logic elements.

Programmable I/O

The programmable I/O pads are used to interface the logic blocks and routing architecture to the external components. The I/O pad and the surrounding logic circuit form as an I/O cell. These cells consume a large portion of the FPGA's area. And the design of I/O programmable blocks is complex, as there are great differences in the supply voltage and reference voltage.

The selection of standards is important in I/O architecture design. Supporting a large number of standards can increase the silicon chip area required for I/O cells. With advancement, the basic FPGA Architecture has developed through the addition of more specialized programmable function blocks.

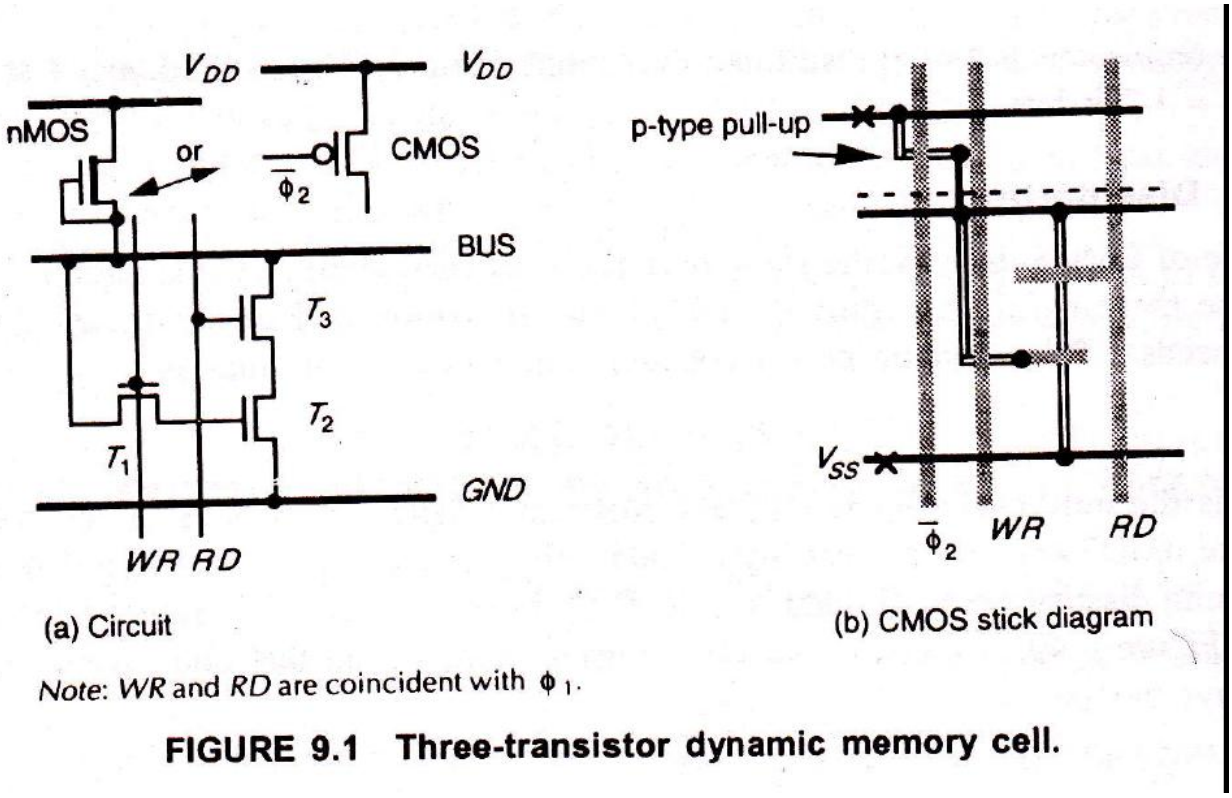
Q: 8 b


English
specification
executable program
behavior
system throughput, design time
function
sequential machines
register-transfer
function units, clock cycles
cost
logic gates
logic
literals, gate depth, power
transistors
circuit
nanoseconds
rectangles
layout
microns

Q: 9 a

3T dynamic RAM cell:

Circuit diagram



(a) Circuit

(b) CMOS stick diagram

Note: *WR* and *RD* are coincident with ϕ_1.

FIGURE 9.1 Three-transistor dynamic memory cell.

Working

- RD = low, bit read from bus through T1, WR = high, logic level on bus sent to Cg of T2, WR = low again
- Bit level is stored in Cg of T2, RD=WR=low
- Stored bit is read by RD = high, bus will be pulled to ground if a 1 was stored else 0 if T2 non-conducting, bus will remain high.

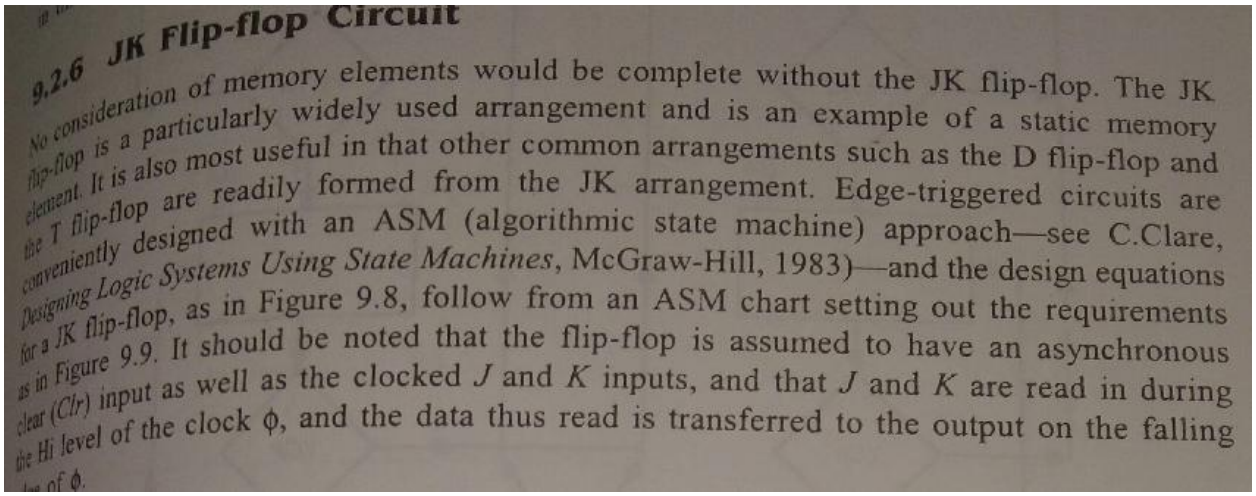
Dissipation

- Static dissipation is nil
- Depends on bus pull-up & on duration of RD signal & switching frequency

Volatility

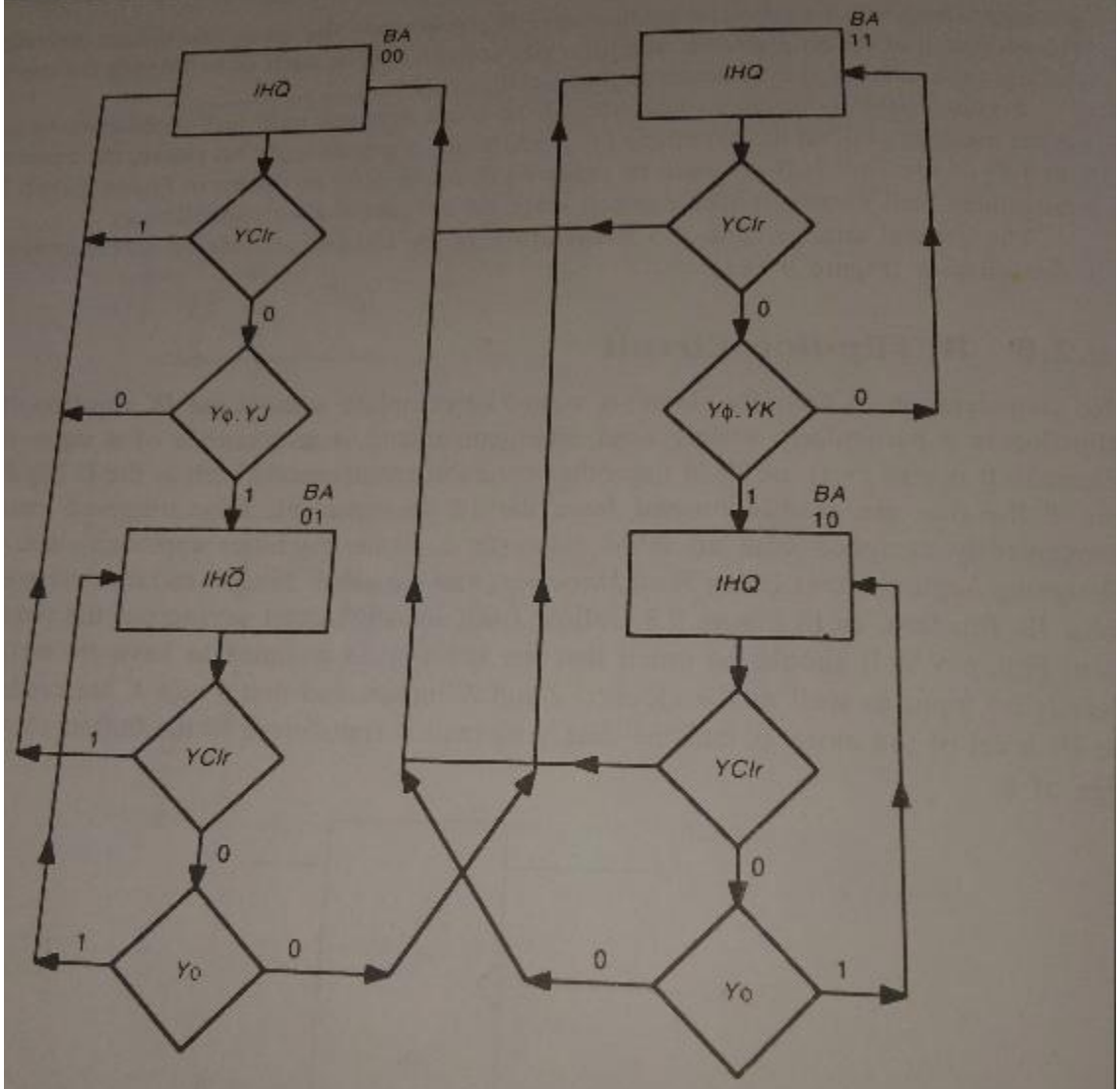
- Cell is dynamic, data will be there as long as charge remains on Cg of T2

Q:9b



9.2.6 JK Flip-flop Circuit

No consideration of memory elements would be complete without the JK flip-flop. The JK flip-flop is a particularly widely used arrangement and is an example of a static memory element. It is also most useful in that other common arrangements such as the D flip-flop and the T flip-flop are readily formed from the JK arrangement. Edge-triggered circuits are conveniently designed with an ASM (algorithmic state machine) approach—see C.Clare, *Designing Logic Systems Using State Machines*, McGraw-Hill, 1983)—and the design equations for a JK flip-flop, as in Figure 9.8, follow from an ASM chart setting out the requirements as in Figure 9.9. It should be noted that the flip-flop is assumed to have an asynchronous clear (*Clr*) input as well as the clocked *J* and *K* inputs, and that *J* and *K* are read in during the Hi level of the clock ϕ, and the data thus read is transferred to the output on the falling



Note: *IHQ* = *Q* output 'Hi' (Immediate)
YClr = Yes clear
Yφ . YJ = Yes clock and *J*, . . ., etc.

FIGURE 9.9 ASM chart for JK flip-flop.

ic gate implement

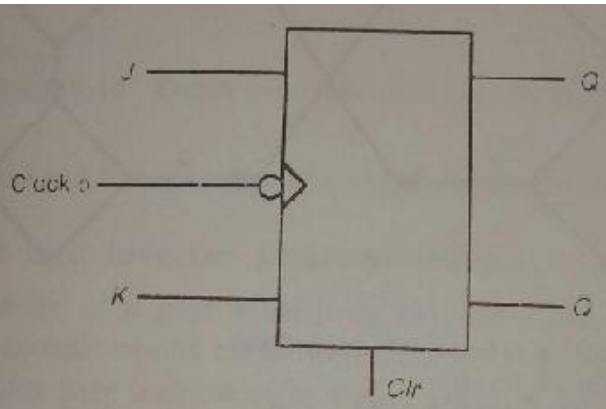


FIGURE 9.8 JK flip-flop.

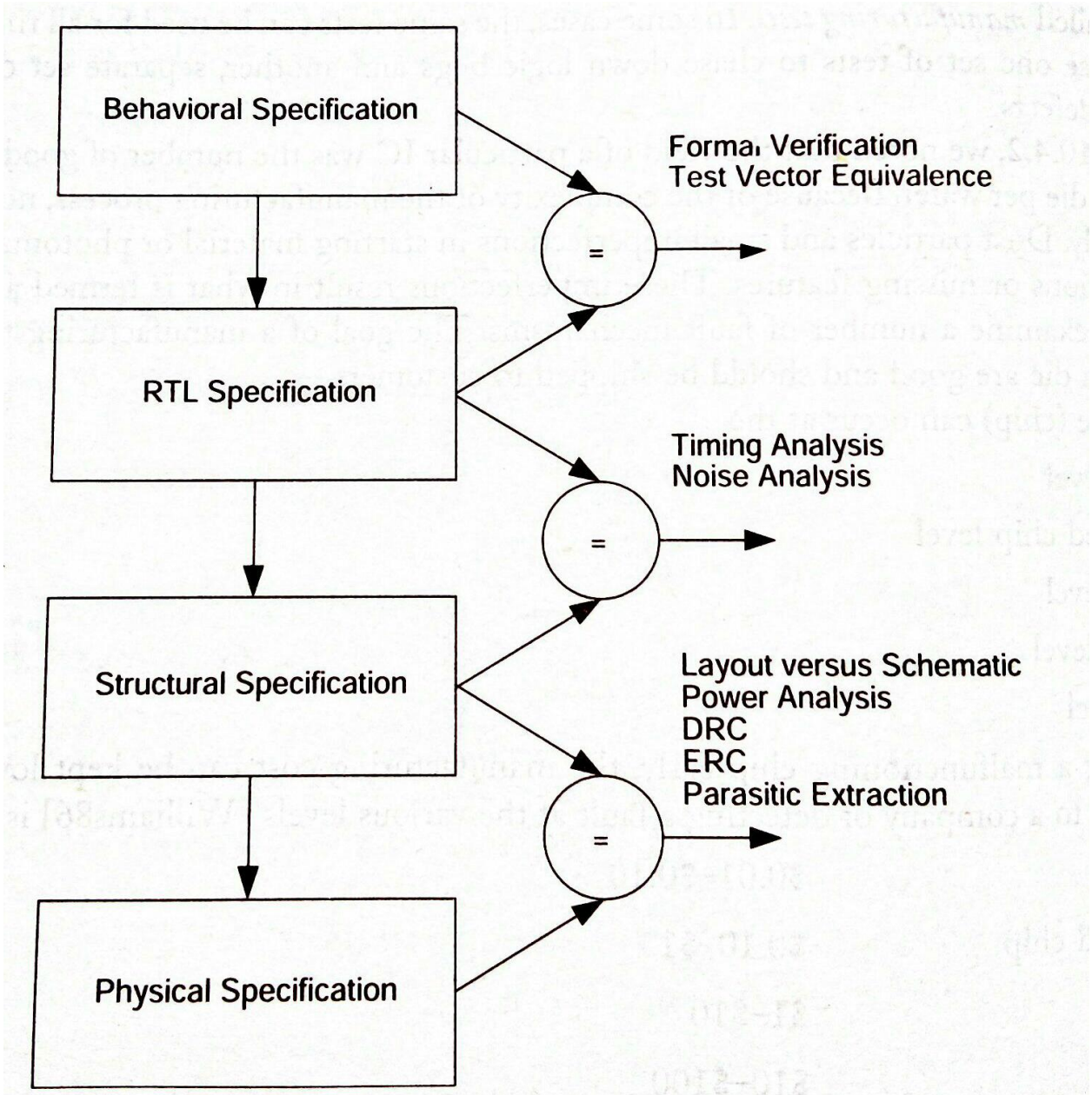
Design equations are readily derived from the ASM chart of Figure 9.9 and, making the ordary variable assignments (AB in the figure), we may express the requirements as lows:

$$A = a.(\overline{Clr}).(\bar{b} + \bar{\phi} + \bar{K}) + \bar{b}.(\overline{Clr}).J.\phi$$

$$B = (\overline{Clr}).(a.\bar{\phi} + b.\phi)$$

re output $Q = B$, and a and b are the fed back state of the secondary variables A and B spectively.

Q:10 a



Verification tests are usually the first ones a designer might construct as part of the design process. Does this adder add? Does this counter count? Does this state-machine yield the right outputs each cycle? Does this modem decode data correctly?

In Chapter 10, we noted that verification tests were required to prove that a synthesized gate description was functionally equivalent to the source RTL. Figure 12.1 shows that we may want to prove that the RTL is equivalent to the design specification at a higher behavioral or specification level of abstraction. The

Q:10 b

the words is found to be defective, the memory can be reconfigured to access the spare row instead. Laser-cut wires or electrically programmable fuses can be used for configuration. Similarly, if the memory has many banks and one or more are found to be defective, they can be disabled, possibly even under software control.

12.6.5.3 Power Elevated power can cause failure due to excess current in wires, which in turn can cause metal migration failures. In addition, high-power devices raise the die temperature, degrading device performance and, over time, causing device parameter shifts. The method of dealing with this component of manufacturability is to minimize power through design techniques described elsewhere in this text. In addition, a suitable package and heat sink should be chosen to remove excess heat.

12.6.5.4 Process Spread We have seen that process simulations can be carried out at different process corners. Monte Carlo analysis, which was introduced in Section 5.5.6, can provide better modeling for process spread and can help with centering a design within the process variations.

12.6.5.5 Yield Analysis When a chip has poor yield or will be manufactured in high volume, dice that fail manufacturing test can be taken to a laboratory for yield analysis to locate the root cause of the failure. If particular structures are determined to have caused many of the failures, the layout of the structures can be redesigned. For example, during volume production ramp-up for a major microprocessor, the silicide over long thin polysilicon lines was found to often crack and raise the wire resistance. This in turn led to slower-than-expected operation for the cracked chips. The layout was modified to widen polysilicon wires or strap them with metal wherever possible, boosting the yield at higher frequencies.

12.6.5 Design for Manufacturability

Circuits can be optimized for manufacturability to increase their yield. This can be done in a num different ways.

12.6.5.1 Physical At the physical level (i.e., mask level), the yield and hence manufacturability improved by reducing the effect of process defects. The design rules for particular processes will freq have guidelines for improving yield. The following list is representative.

- Increase the spacing between wires where possible—this reduces the chance of a defect cau short circuit.
- Increase the overlap of layers around contacts and vias—this reduces the chance that a misalig will cause an aberration in the contact structure.
- Increase the number of vias at wire intersections beyond one if possible—this reduces the cha a defect causing an open circuit.

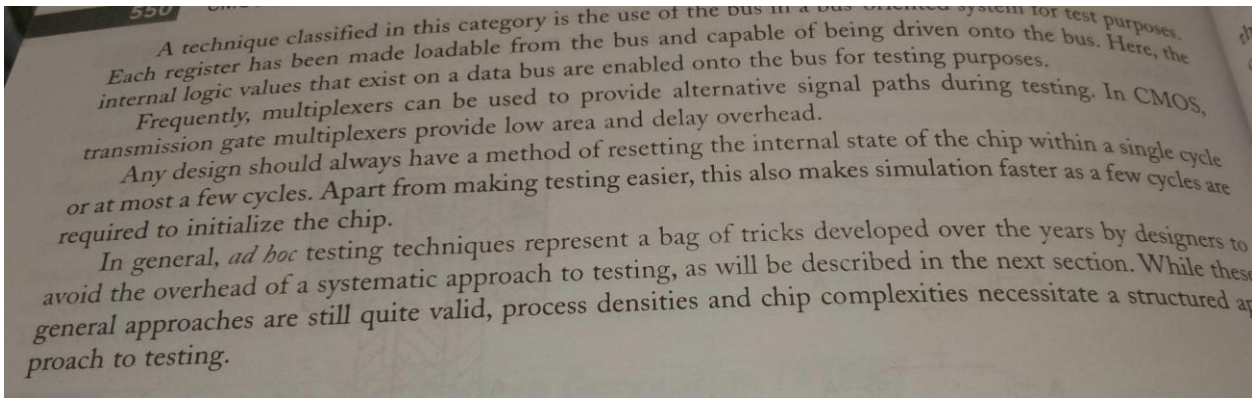
Increasingly, design tools are dealing with these kinds of optimizations automatically.

12.6.5.2 Redundancy Redundant structures can be used to compensate for defective components chip. For example, memory arrays are commonly built with extra rows. During manufacturing test, if

Q:10 c

Ad hoc test techniques, as their name suggests, are collections of ideas aimed at reducing the combinational explosion of testing. They are summarized here for historical reasons. They are only useful for small designs where scan, ATPG, and BIST are not available. A complete scan-based testing methodology is recommended for all digital circuits. Having said that, common techniques for ad hoc testing involve:

- Partitioning large sequential circuits
- Adding test points
- Adding multiplexers
- Providing for easy state reset



A technique classified in this category is the use of the bus in a bus-oriented system for test purposes. Each register has been made loadable from the bus and capable of being driven onto the bus. Here, the internal logic values that exist on a data bus are enabled onto the bus for testing purposes.

Frequently, multiplexers can be used to provide alternative signal paths during testing. In CMOS, transmission gate multiplexers provide low area and delay overhead.

Any design should always have a method of resetting the internal state of the chip within a single cycle or at most a few cycles. Apart from making testing easier, this also makes simulation faster as a few cycles are required to initialize the chip.

In general, *ad hoc* testing techniques represent a bag of tricks developed over the years by designers to avoid the overhead of a systematic approach to testing, as will be described in the next section. While these general approaches are still quite valid, process densities and chip complexities necessitate a structured approach to testing.