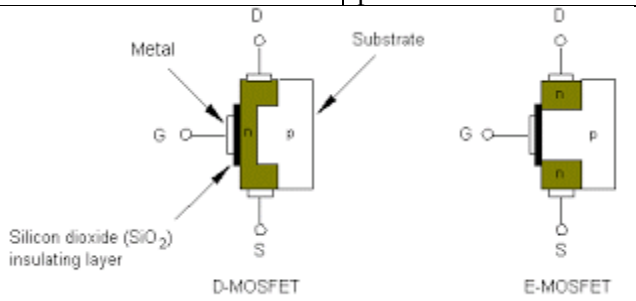
1	
a	• MOSFET is more tolerant to heat (stable to thermal changes) and can simulate a good resistor. • MOSFET's have high input impedance(goes infinity at lower frequencies) • MOSFET's are power efficient as the outputs are controlled by input voltage instead of input current. • Current Mirror Circuits made from MOSFET are far better than that with BJT. • MOSFET's have less chances of thermal runaway than BJT's.
b	The change in the threshold voltage of a MOSFET, because of the voltage difference between body and source is called body effect. The expression for the threshold voltage is given by the following expression. $V_{th} = V_{T0} + \gamma(\sqrt{\lvert 2\phi_F - V_{SB} \rvert} - \sqrt{\lvert 2\phi_F \rvert})$

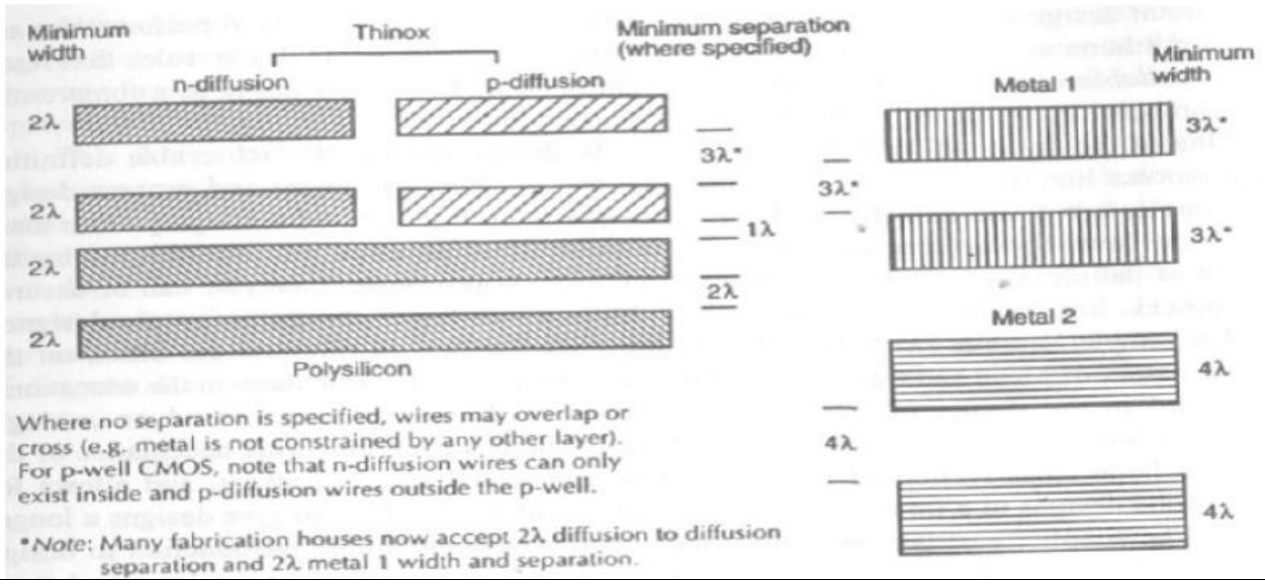
c

	D-MOSFET	E-MOSFET
1	Voltage from gate to source (Vgs) can be positive or negative.	Voltage from gate to source (Vgs) is always positive.
2		
3	The transistor requires the Gate-Source voltage, (VGS) to switch the device "OFF". The depletion mode MOSFET is equivalent to a "Normally Closed" switch.	The transistor requires a Gate-Source voltage, (VGS) to switch the device "ON". The enhancement mode MOSFET is equivalent to a "Normally Open" switch.

d

$$P_T = P_g * T_d = \frac{1}{\beta^2}\left(\frac{\beta}{\alpha^2}\right) = \frac{1}{\alpha^2\beta}$$

e



f

- **Standard Capacitance : [□Cg]**

→ Standard capacitance □Cg is defined the gate-to-channel capacitance of a MOS transistor having W=L = feature size, i.e., 'standard' or 'feature size'.

∴

$$\square C_g = \frac{\varepsilon_0 \varepsilon_{ins} A}{D}$$

$$C = K \times 10^{-4} \frac{PF}{\mu m^2}$$

g

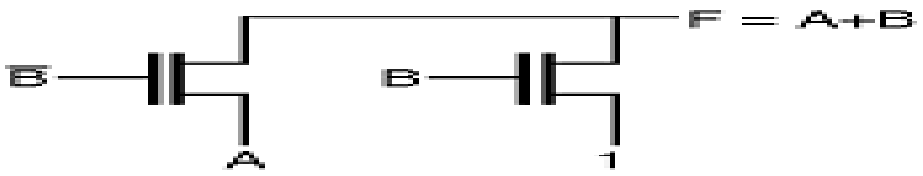
It is a CMOS-based switch, in which PMOS passes a strong 1 but poor 0, and NMOS passes strong 0 but poor 1. Both PMOS and NMOS work simultaneously.

h

- CMOS domino logic
- n-p CMOS logic

i

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1



'OR' gate using pass transistor logic

j

	FPGA	ASIC
1	Reconfigurable circuit. FPGAs can be reconfigured with a different design. They even have capability to reconfigure a part of chip while remaining areas of chip are still working! This feature is widely used in accelerated computing in data centres.	Permanent circuitry. Once the application specific circuit is taped-out into silicon, it cannot be changed. The circuit will work same for its complete operating life.
2	Design is specified generally using hardware description languages (HDL) such as VHDL or Verilog.	Same as for FPGA. Design is specified using HDL such as Verilog, VHDL etc.

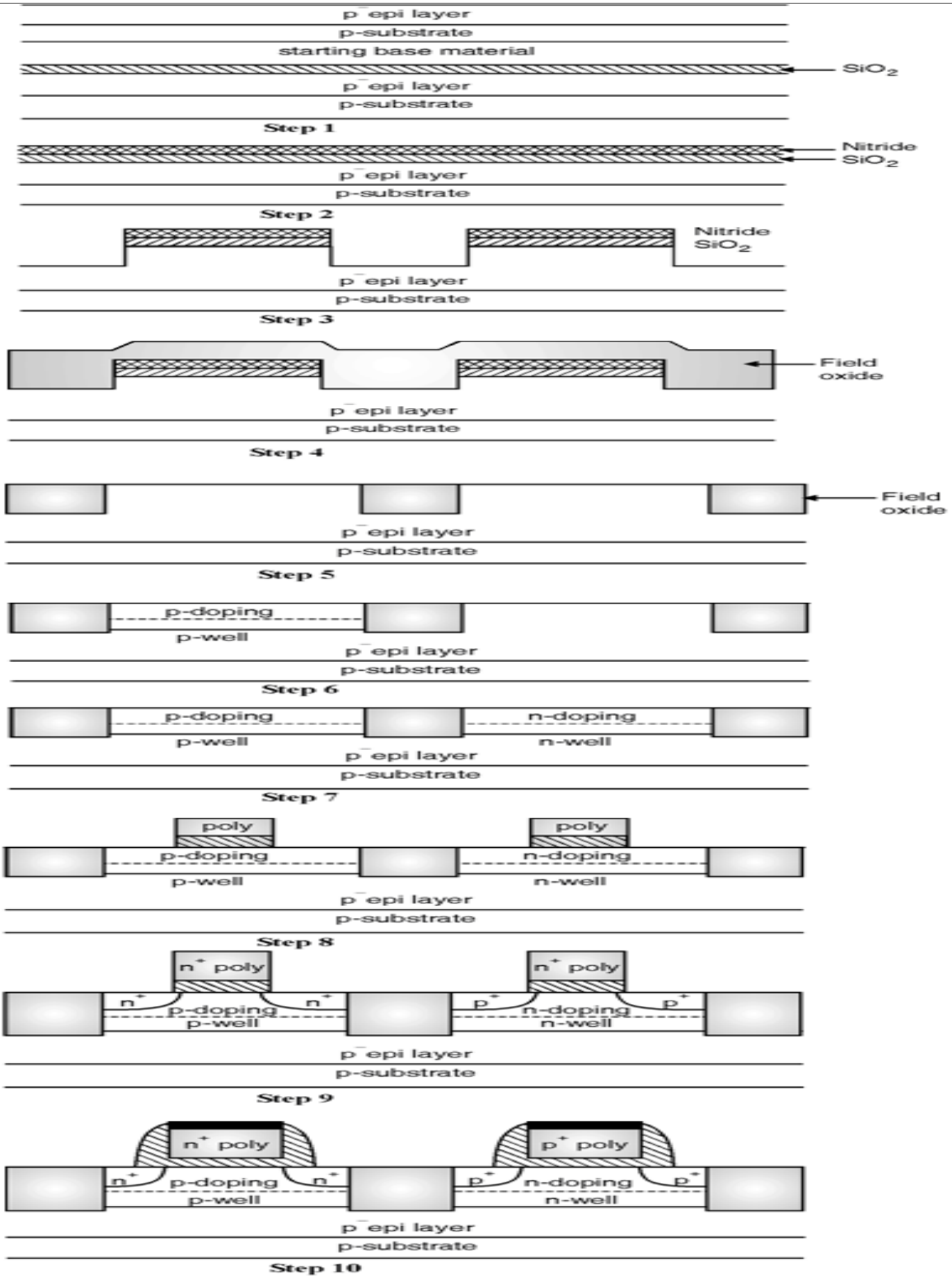
k

- Design with RTL description + logic synthesis tool ,Abstract level, Independent to fabrication technology, Reuse when fabrication technology changing
- Functional verification can be done early, Optimized to meet the desired functionality
- Analogous to computer programming, Textual description with comments

l

The conditional operator selects an expression for evaluation depending on the value of condition.
Conditional Expression? true expression : false expression

2.a	In Dual-well process both p-well and n-well for NMOS and PMOS transistors respectively are formed on the same substrate. The main advantage of this process is that the threshold voltage, body effect parameter and the transconductance can be optimized separately. The starting material for this process is p+ substrate with epitaxially grown p-layer which is also called as epilayer. The process starts with a p-substrate surfaced with a lightly doped p-epitaxial layer. **Step 1 :** A thin layer of SiO_2 is deposited which will serve as the pad oxide. **Step 2 :** A thicker sacrificial silicon nitride layer is deposited by chemical vapour deposition. **Step 3 :** A plasma etching process is used to create trenches used for insulating the devices. **Step 4 :** The trenches are filled with SiO_2 which is called as the field oxide. **Step 5 :** To provide flat surface chemical mechanical planarization is performed and also sacrificial nitride and pad oxide is removed. **Step 6 :** The p-well mask is used to expose only the p-well areas, after this implant and annealing sequence is applied to adjust the well doping. This is followed by second implant step to adjust the threshold NMOS transistor. **Step 7 :** The n-well mask is used to expose only the n-well areas, after this implant and annealing sequence is applied to adjust the well doping. This is followed by a second implant step to adjust the threshold voltage of PMOS transistor. **Step 8 :** A thin layer of gate oxide and polysilicon is chemically deposited and patterned with the help of polysilicon mask. **Step 9 :** Ion implantation to dope the source and drain regions of the PMOS (p^+) and NMOS (n^+) transistors is used this will also form n^+ polysilicon gate and p^+ polysilicon gate for NMOS and PMOS transistors respectively. **Step 10 :** Then the oxide or nitride spacers are formed by chemical vapour deposition (CVD). **Step 11 :** In this step contact or holes are etched, metal is deposited and patterned. After the deposition of last metal layer final passivation or overglass is deposited for protection.



Dual well process steps

2.b

Given,

$V_T = 400mV = 0.4V$.

Voltage applied at gate

$V_{GS} = 900mV = 0.9V$.

$I_{DS} = 1mA$.

Find the drain current for.

$V_{GS} = 1400mV$.

MOSFET is operating in saturation

$$I_{DS} = K(V_{GS} - V_T)^2$$

$$\text{or } 1 \times 10^{-3} = K(0.9 - 0.4)^2$$

$$\text{or } K = \frac{10^{-3}}{(0.5)^2} = 4 \times 10^{-3} A/V^2$$

For $V_{GS} = 1.4V$.

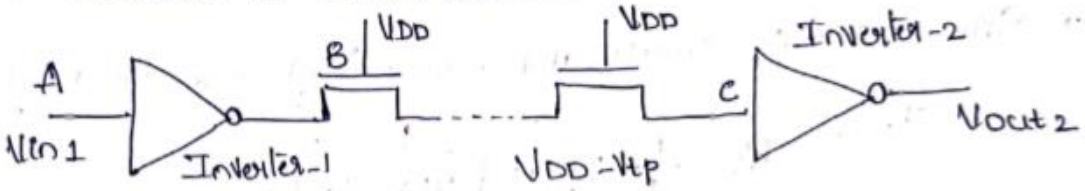
$$I_{DS} = K(V_{GS} - V_T)^2$$

$$= 4 \times 10^{-3} (1.4 - 0.4)^2$$

$$I_{DS} = 4mA.$$

3.a

⇒ The input to inverter 2 comes from the output of inverter1 but passes through one or more nMOS transistors used as switches in series (called pass transistors)



⇒ The connection of pass transistors in series will degrade the logic 1 level into inverter 2 so that the output will not be a proper 0 logic level.

⇒ The critical condition is when point A is at 0 volts and B is thus at VDD, but the voltage into inverter 2 at point C is now reduced from VDD by the threshold voltage of the series pass transistor.

⇒ with all pass transistor gates connected to VDD, there is a loss of Vtp. The input voltage to inverter 2 is

$$V_{in2} = V_{DD} - V_{tp}.$$

where V_{tp} - threshold voltage for a pass transistor.

⇒ Consider inverter-1 with input = VDD. If the input is at VDD, then the p.d transistor T_2 is conducting but with a low voltage across it, it is in resistive region. R_1.

⇒ The p.u transistor T_1 is in saturation and is represented as a current source.

For the p.d transistor

$$I_{ds} = K \frac{W_{p.d1}}{L_{p.d1}} \left[(V_{DD} - V_t) V_{ds1} - \frac{V_{ds1}^2}{2} \right]$$

$$R_1 = \frac{V_{ds1}}{I_{ds1}} = \frac{1}{K} \frac{L_{p.d1}}{W_{p.d1}} \left[\frac{1}{V_{DD} - V_t - \frac{V_{ds1}}{2}} \right]$$

⇒ Since V_{DS1} is small and hence $\frac{V_{DS1}}{2}$ can be neglected

Thus $$R_1 = \frac{1}{K} \frac{}{2p.d1} \left[\frac{1}{V_{DD} - V_t} \right]$$

⇒ The pull-up device in depletion mode with $V_{gs} = 0$

$$I_1 = I_{ds} = K \frac{W_{p.u1}}{L_{p.u1}} \frac{(-V_{td})^2}{2}$$

The output of inverter 1,

$$V_{out1} = -I_1 R_1$$

$$V_{out1} = -I_1 R_1 = \frac{2_{p.d1}}{2_{p.u1}} \left[\frac{1}{V_{DD} - V_t} \right] \frac{(-V_{td})^2}{2}$$

⇒ Consider inverter 2, when input = $V_{DD} - V_{tp}$

Similar to inverter 1

$$R_2 = \frac{1}{K} \frac{}{2p.d2} \frac{1}{[(V_{DD} - V_{tp}) - V_t]}$$

$$I_2 = K \frac{1}{2p.u2} \frac{(-V_{td})^2}{2}$$

Output voltage of inverter 2 $V_{out2} = -I_2 R_2$

$$V_{out2} = I_2 R_2 = \frac{\tau_{pd2}}{\tau_{pu2}} \left[\frac{1}{V_{DD} - V_{tp} - V_t} \right] \frac{(-V_{td})^2}{2}$$

⇒ If inverter 2 is to have the same output voltage under these conditions then $V_{out1} = V_{out2}$.

$$I_1 R_1 = I_2 R_2$$

$$\frac{\tau_{pu2}}{\tau_{pd2}} = \frac{\tau_{pu1}}{\tau_{pd1}} \frac{(V_{DD} - V_t)}{(V_{DD} - V_{tp} - V_t)}$$

⇒ Taking typical values

$$V_t = 0.2 V_{DD}$$

$$V_{tp} = 0.3 V_{DD}.$$

$$\frac{\tau_{pu2}}{\tau_{pd2}} = \frac{\tau_{pu1}}{\tau_{pd1}} \frac{0.8}{0.5}$$

Therefore $\frac{\tau_{pu2}}{\tau_{pd2}} = 2 \frac{\tau_{pu1}}{\tau_{pd1}} = \frac{8}{1}$

Therefore an inverter driven through one or more pass transistors has a $\frac{\tau_{pu}}{\tau_{pd}}$ ratio of $\frac{8}{1}$

3.b

⇒ Latch-up in CMOS Circuits:

→ A problem which is inherent in the p-well & n-well Processes is due to the relatively large no. of junctions which are formed in these structures, the consequent Presence of parasitic transistors are diodes.

Latch-up is a condition in which the parasitic components give rise to the establishment of low-resistance conducting Paths between V_{DD} & V_{ss} with disastrous results.

⇒ There are, in effect, two transistors & two resistances, which form a path between V_{DD} & V_{SS}.

⇒ If sufficient substrate current flows to generate enough voltage across R_s to turn on transistor T_1, this will then draw current through R_p and if the voltage developed is sufficient T_2 will also turn on, establishing a low-resistance path between the supply rails.

⇒ If the current gains of the two transistors are such that $\beta_1 \times \beta_2 > 1$, latch-up may occur.

⇒

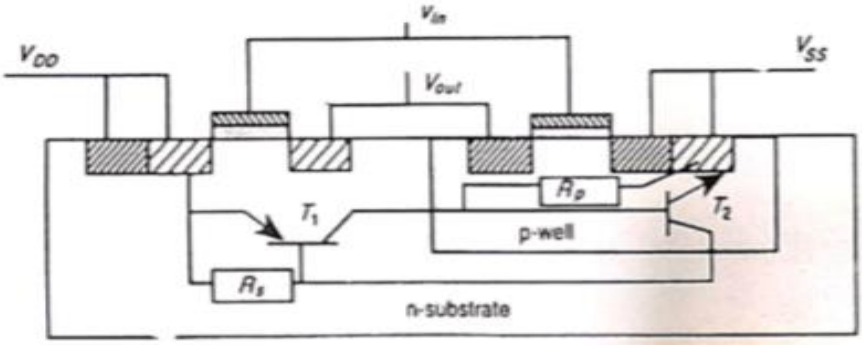


FIGURE 2.21 Latch-up effect in p-well structure.

⇒ With no injected current, the parasitic transistors will exhibit high resistance, but sufficient substrate current flow will cause switching to the low-resistance state.

⇒ Once latched-up, this condition will be maintained until the latch-up current drops below I_L.

⇒ Remedies for the latch-up problem include:

1. an increase in substrate doping levels with a consequent drop in the value of R_s.
2. Reducing R_p by control of fabrication parameters and

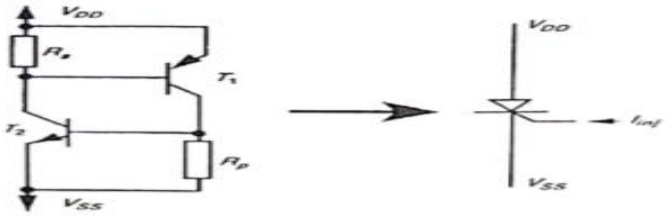


FIGURE 2.22 Latch-up circuit model.

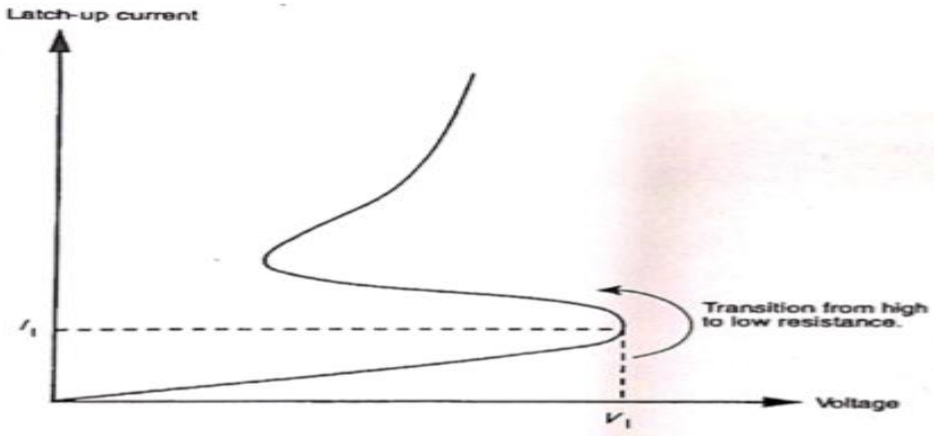


FIGURE 2.23 Latch-up current versus voltage.

by ensuring a low contact resistance to Vss. (8)

3. elaborate measures such as introduction of guard rings.

4.a

Rule 1:

When two or more 'sticks' of the same type cross or touch each other that represents electrical contact.

Rule 2:

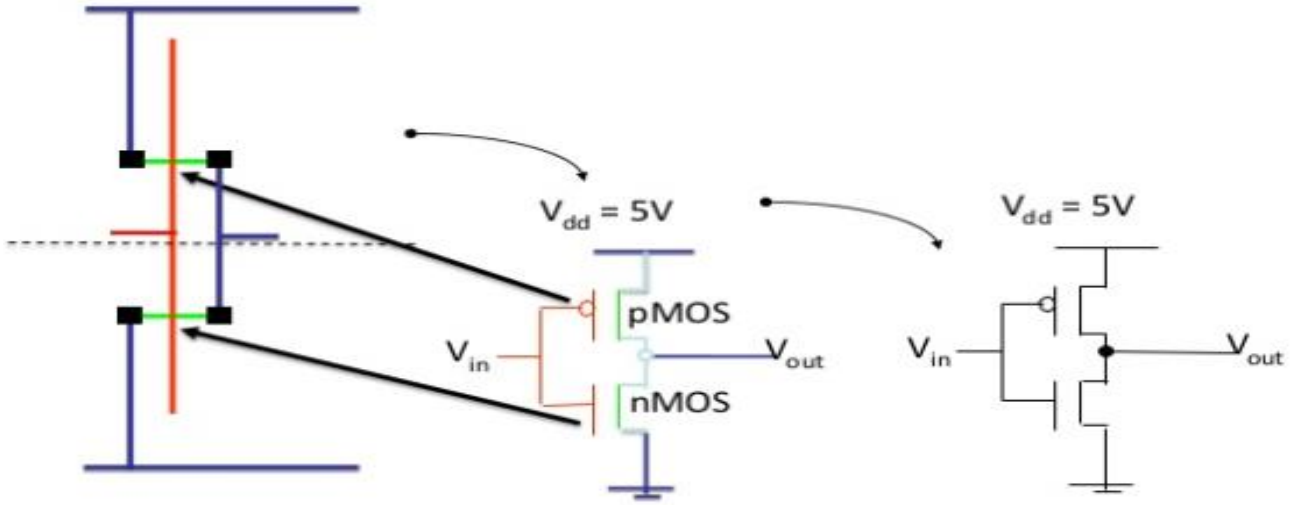
When two or more 'sticks' of different type cross or touch each other there is no electrical contact.

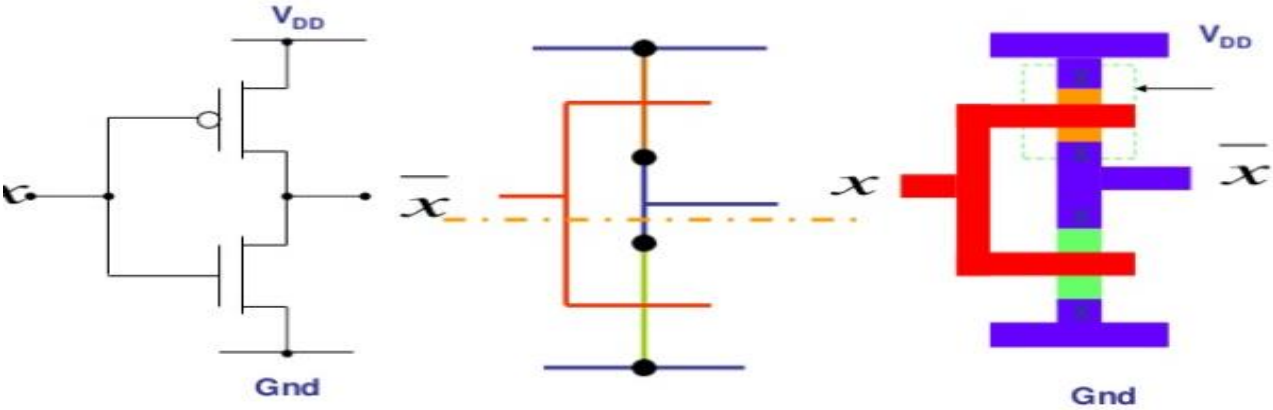
(If electrical contact is needed we have to show the connection explicitly)

Rule 3:
When a poly crosses diffusion it represents a transistor.

Note: If a contact is shown then it is ***not*** a transistor.

Rule 4:
In CMOS a demarcation line is drawn to avoid touching of p-diff with n-diff. All PMOS must lie on one side of the line and all NMOS will have to be on the other side.





4.b

Propagation Delays:

Cascaded Pass Transistor

⇒ MOS technology offers freedom in the use of pass transistors as series or parallel switches in logic array.

↳ Therefore, logic signals must pass through a number of pass transistors in series.

⇒ A chain of four such transistors is shown in figure, in which all gates have been connected to VDD. (logic 1) a signal to be propagated to the output.

⇒ The circuit may be modeled, and it is then possible to evaluate the delay through the network.

⇒ The response at node V_2 with respect to time is given by

$$e \cdot \frac{dV_2}{dt} = (I_1 - I_2) = \frac{[(V_1 - V_2) - (V_2 - V_3)]}{R}$$

As the number of sections in such a network becomes large, this expression reduces to.

$$RC \frac{dV}{dt} = \frac{d^2V}{dx^2}$$

$$I_1 = \frac{V_1 - V_2}{R}$$

$$I_2 = \frac{V_2 - V_3}{R}$$

R.

where

R = resistance per unit length.

C = capacitance per unit length.

x = distance along network from input.

↳ The time taken by the signal to propagate a distance x is known as propagation delay time, τ_p

$$\tau_p \propto x^2$$

when logic to circuits are considered

$$R_{total} = n.R.R_s$$

$$C_{total} = n.C.\Pi C_g.$$

where n - no. of stages.

R - Resistance per section w.r.to R_s.

R_s - Sheet Resistance

C - Capacitance per section w.r.t $\square C_g$

$\square C_g$ - Standard capacitance.

Total propagation delay τ_d

$$\tau_d = n^2 R_s C \cdot R_s \square C_g$$

$$\tau_d = n^2 R_s C \tau$$

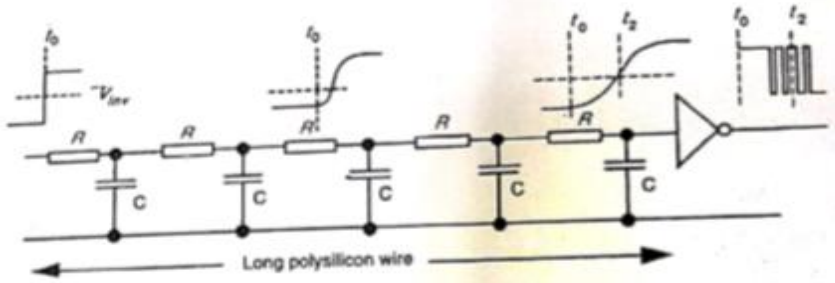
- Long polysilicon wires:

→ These also contribute series R & C as the case for cascaded pass transistors and signal propagation is slowed down.

⇒ The use of buffers is recommended for long polysilicon runs. Use of buffers is to drive two desirable effects:

1. The signal propagation is speeded up &
2. Reduction in sensitivity to noise.

⇒ The slow rise time of the signal at the input of the inverter means that the input voltage spends a relatively long time in the vicinity of V_{inv}, that small disturbance due to noise will switch the inverter state between '0' & '1'

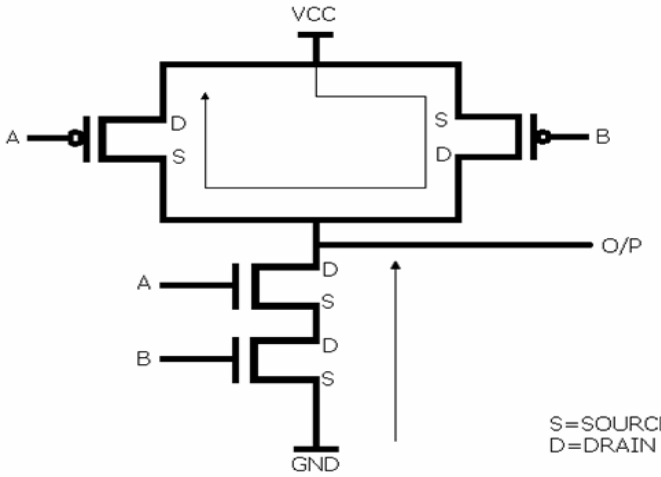


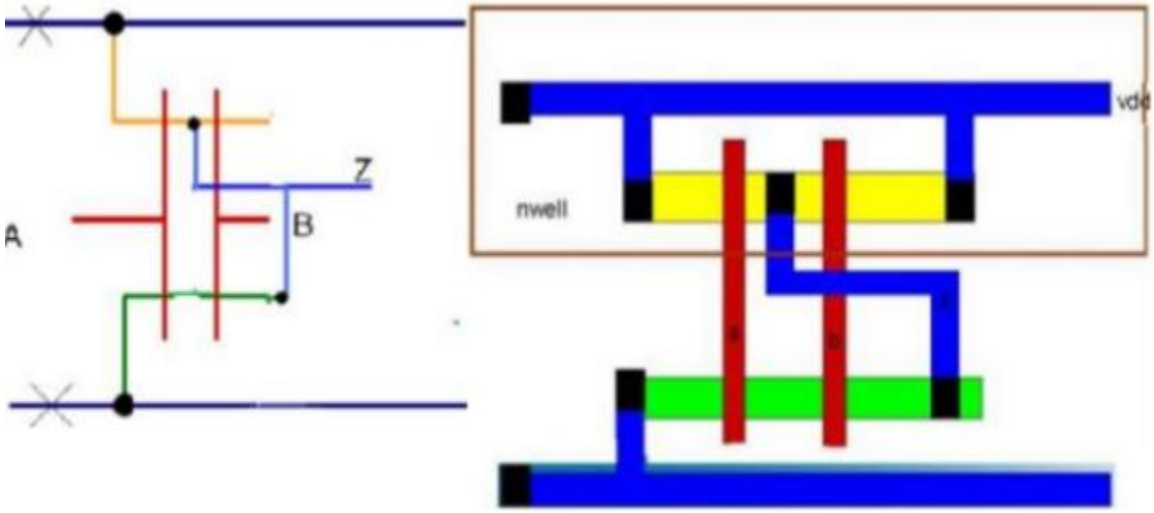
Note: V_{inv} = Inverter threshold

FIGURE 4.18 Possible effects of delays in long polysilicon wires.

⇒ It is essential that long polysilicon wires be driven by suitable buffers to guard the effects of noise and to speed-up the rise-time of propagated signal edges.

5.a





5.b

- Gate capacitance C_g $C_g = C_o * L * W$

  Thus C_g is scaled up by $\beta * 1/\alpha^2 = \beta/\alpha^2$

- Parasitic capacitance C_x

  C_x is proportional to A_x/d
  where d is the depletion width around source or drain and scaled by $1/\alpha$

  A_x is the area of the depletion region around source or drain, scaled by $(1/\alpha^2)$.
  Thus C_x is scaled up by $\{1/(1/\alpha)\} * (1/\alpha^2) = 1/\alpha$

- Carrier density in channel Q_{on}

  $Q_{on} = C_o * V_{gs}$
  where Q_{on} is the average charge per unit area in the 'on' state.
  C_o is scaled by β and V_{gs} is scaled by $1/\beta$

  Thus Q_{on} is scaled by 1

- Power dissipation per gate P_g

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

P_g comprises of two components: static component P_{gs} and dynamic component P_{gd}:

Where, the static power component is given by: $P_{gs} = \frac{V_{DD}^2}{R_{on}}$

And the dynamic component by: $P_{gd} = E_g f_o$

Since V_{DD} scales by $(1/\beta)$ and R_{on} scales by 1, P_{gs} scales by $(1/\beta^2)$.

Since Eg scales by $(1/\alpha^2\ \beta)$ and f_o by (α_2 /β), P_{gd} also scales by $(1/\beta^2)$. Therefore, P_g scales by $(1/\beta^2)$.

- Gate area A_g

$$A_g = L * W$$

Where L: Channel length and W: Channel width and both are scaled by
Thus A_g is scaled up by $1/\alpha^2$

- Channel Resistance R_{on}

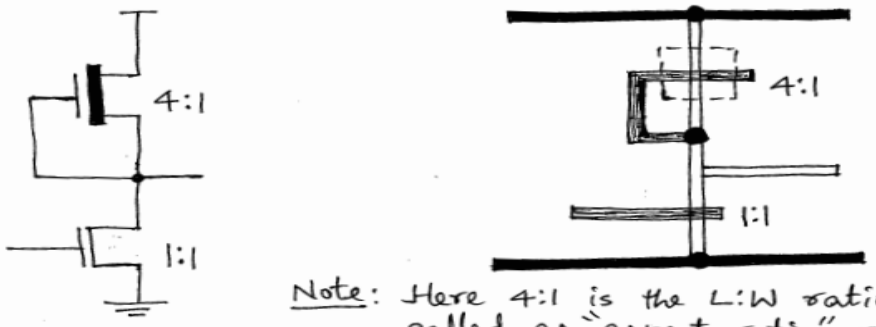
$$R_{on} = \frac{L}{W} * \frac{1}{Q_{on} * \mu}$$

Where μ = channel carrier mobility and assumed con

Thus R_{on} is scaled by 1.

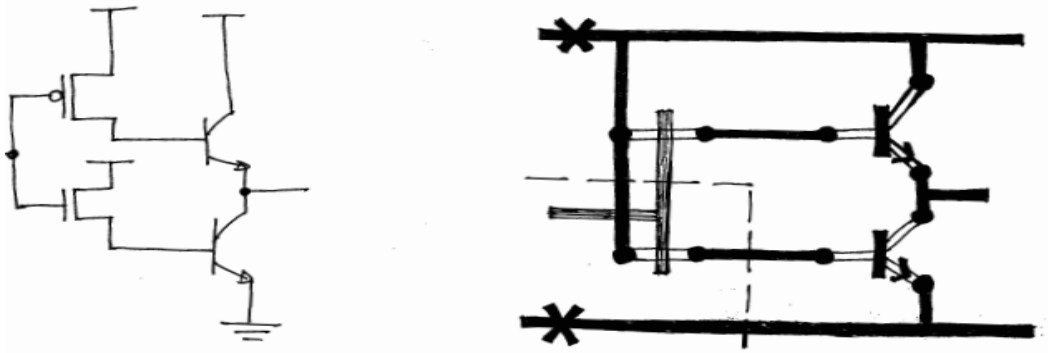
6.a

ii) 4:1 nMOS inverter:



Note: Here 4:1 is the L:W ratio, which is called as "aspect ratio", Z.

ii) n-well based BiCMOS inverter:



Two bipolar transistors (T3 and T4), one nMOS and one pMOS transistor (both enhancement-type devices, OFF at Vin=0V)
The MOS switches perform the logic function & bipolar transistors drive output loads
Vin = 0 :
T1 is off. Therefore T3 is non-conducting
T2 ON - supplies current to base of T4
T4 base voltage set to Vdd.
T4 conducts & acts as current source to charge load CL towards Vdd.
Vout rises to Vdd - Vbe (of T4)
Vin = Vdd :
T2 is off. Therefore T4 is non-conducting.
T1 is on and supplies current to the base of T3
T3 conducts & acts as a current sink to discharge load CL towards 0V.
Vout falls to 0V+ VCEsat (of T3)
Note : VCEsat (of T3) is saturation V from T3 collector to emitter
T3 & T4 present low impedances when turned on into saturation & load CL will be charged or discharged rapidly
• Output logic levels will be good & will be close to rail voltages since VCEsat is quite small & VBE » 0.7V. Therefore, inverter has high noise margins
• Inverter has high input impedance, i.e., MOS gate input
• Inverter has low output impedance
• Inverter has high drive capability but occupies a relatively small area
• However, this is not a good arrangement to implement since no discharge path exists for current from the base of either bipolar transistor when it is being turned off, i.e.,
• when Vin=Vdd, T2 is off and no conducting path to the base of T4 exists
• when Vin=0, T1 is off and no conducting path to the base of T3 exists

6.b

By means of suitable programming, the adder can be utilized to implement other functions such as subtraction, AND, OR, EX-OR and EX-NOR. Thus, the adder can be utilized to perform the arithmetic & logical functions, as explained below —

i) Addition : directly possible. $(A+B)$

ii) Subtraction : this is possible by adding the 2's complement of the subtrahend to the minuend. i.e, (A + 2's complement of B = A − B). Two's complement is possible by complementing the subtrahend and then adding '1' to it.

iii) EX-OR : We know that the sum-bit is given by, $S_k = A_k \oplus B_k \oplus C_{k-1}$. When C_{k-1} is held at '0', then

$$S_k = A_k \oplus B_k \oplus '0' = \underline{\underline{A_k \oplus B_k}}$$

iv) EX-NOR : Similarly when C_{k-1} is held at '1',

$$S_k = A_k \oplus B_k \oplus '1' = \underline{\underline{A_k \odot B_k}}$$

v) AND : We know that the carry is given

$$C_k = A_k B_k + B_k C_{k-1} + A_k C_{k-1}.$$

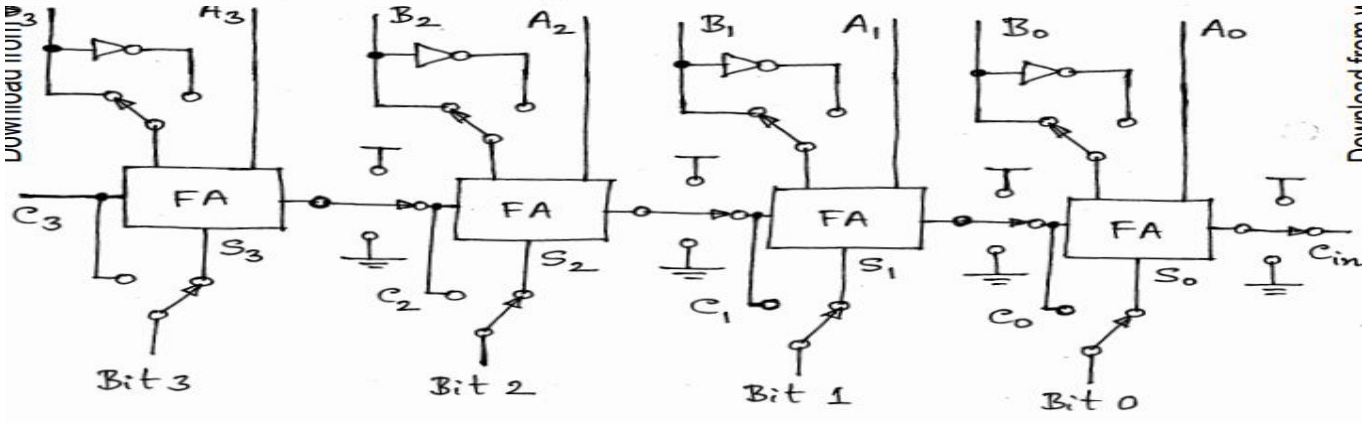
When C_{k-1} is held at '0', then

$$\underline{\underline{C_k = A_k . B_k}}$$

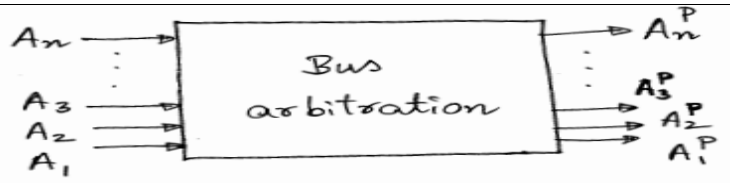
vi) OR : Similarly when C_{k-1} is held at '1',

$$C_k = A_k B_k + B_k + A_k = \underline{\underline{A_k + B_k}}.$$

As it is possible to implement all these functions using adder element, the ALU can be designed by means of the following arrangement :—



7.a

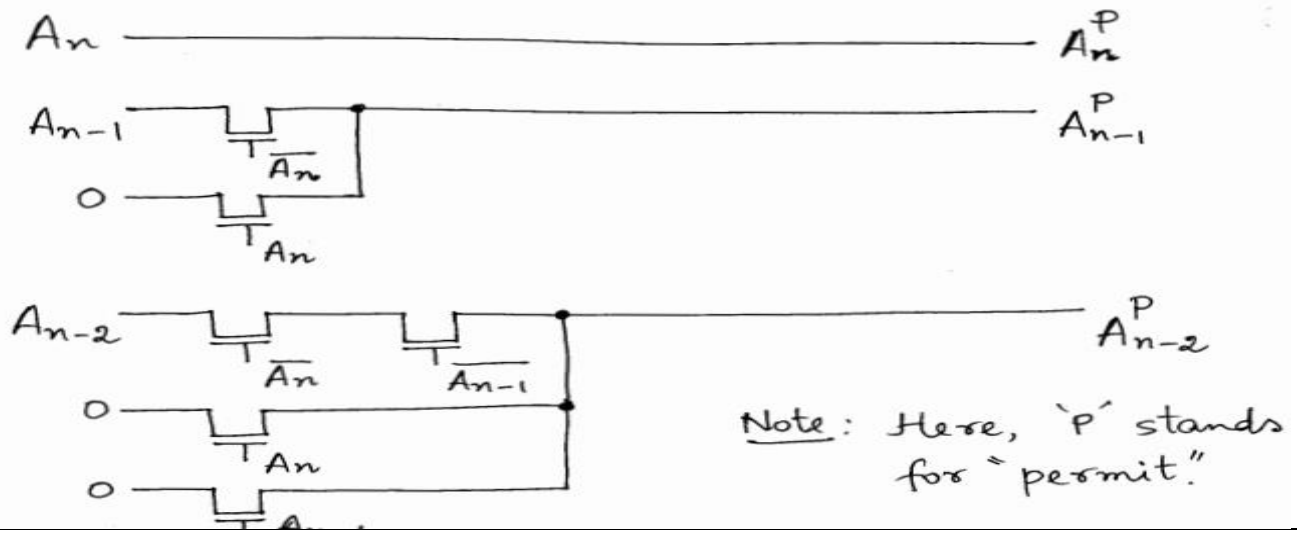


This is utilized to grant the bus to the different devices based on their priority. Here, An is the device having highest priority and A_1 is the one having least priority. Hence when An requests for the bus, it has to be granted, irrespective of the status of other devices. When A_1 requests for the bus, it will be granted only when all the other devices don't have any request for the bus. Taking the example of 4 devices, the truth-table can be written as follows :—

A_4	A_3	A_2	A_1
1	x	x	x
0	1	x	x
0	0	1	x
0	0	0	1
0	0	0	0

A_4^P	A_3^P	A_2^P	A_1^P
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
0	0	0	0

Based on the truth table, the circuits for the logics can be written as follows—

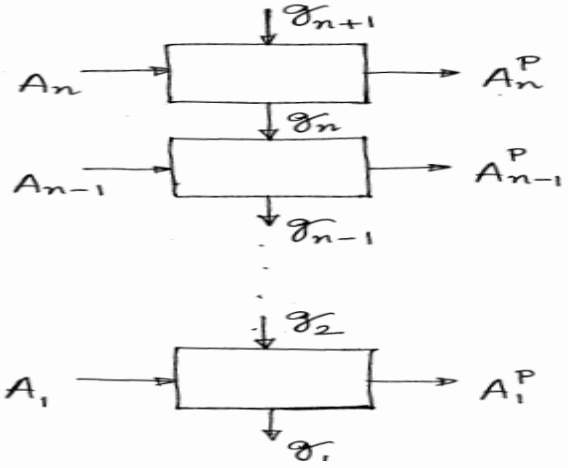


Note: Here, 'P' stands for "permit."

The circuits shown above indicate that the highest priority device will have one diffusion path, for its bus arbitration logi The next device will have two diffusion paths & two switches. Similarly, A_{n-2} will have three diffusion paths and four switche Hence, the design cannot be taken as a standard cell, & thus it is not structured

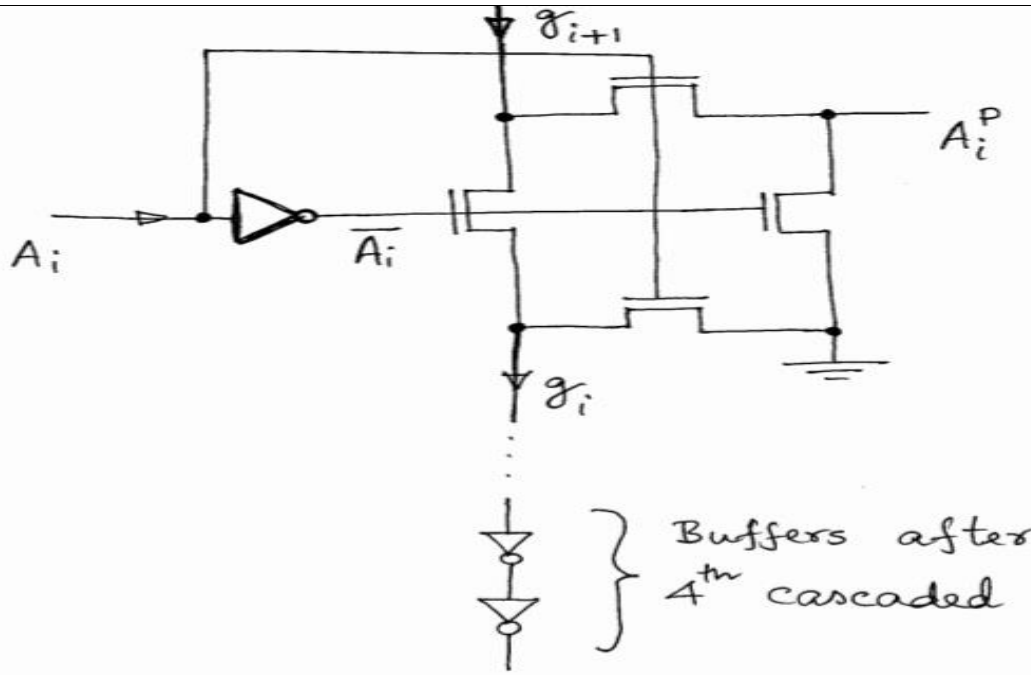
To make the design structured, a new signal called "grant" is going to be utilized as follows –

To make the design structured, a new signal called "grant" is going to be utilized, as follows –

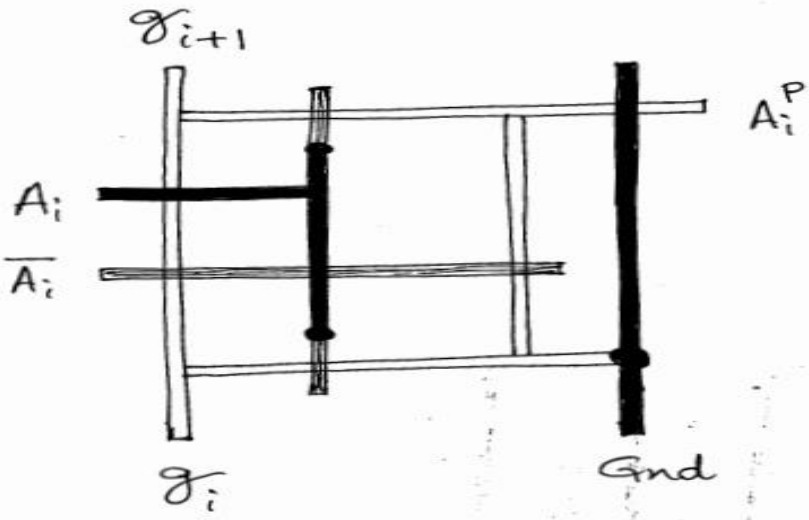


Here, g_{n+1} always remains '1', and it will be passed on to the device A_{n-1} only when A_n allows the grant signal to pass. This means that, when $A_n = 1$, A_n^P will be '1' and $g_n = 0$. When $A_n = 0$, g_n will become '1' and A_{n-1}^P can become '1' if $A_{n-1} = 1$. This process goes on until the device having least priority.

The circuit for the bus-grant logic can be written as follows –

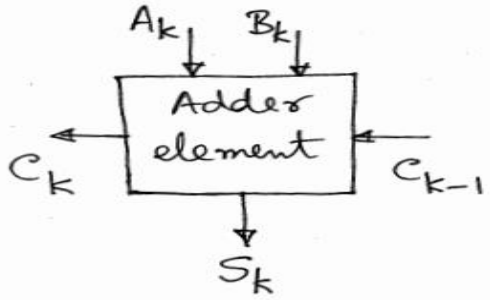


In the diagram shown above, A_i^P will be '1' only if $g_{i+1} = 1$ and $A_i = 1$. At that time, $g_i = 0$. If $A_i \neq 1$, then $g_i = g_{i+1}$ and $A_i^P = 0$. The circuit showing four transistors can be considered as a 'cell' and its stick diagram can be written as shown —



7.b

Adder element: The adder element is a Full-adder, which gets replicated for the implementation of parallel addition. The diagram & the truth-table are as shown



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

The conventional expression for sum & carry are –

$$S_k = A_k \oplus B_k \oplus C_{k-1}$$

& $$C_k = A_k B_k + B_k C_{k-1} + A_k C_{k-1}$$

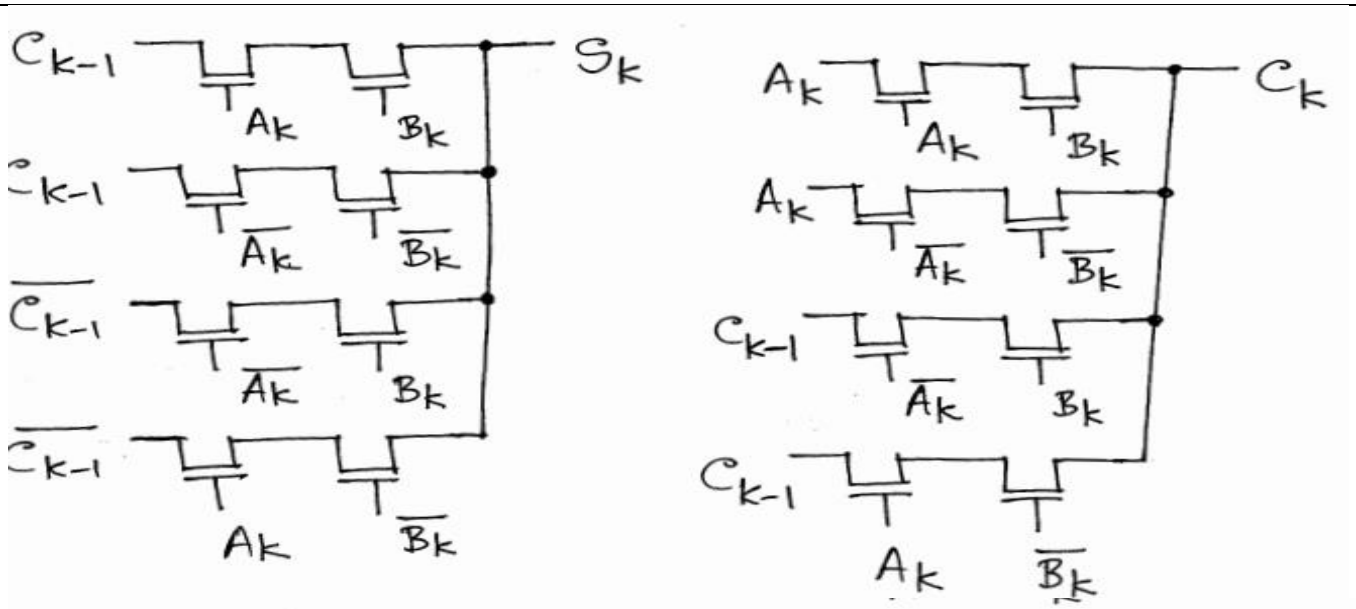
These expressions are utilized to realize the adder element using the logic gates. But here, we shall make an effort to realize the full-adder using pass-transistors. For this purpose, we shall write the expressions in a different way.

From the observation of the truth-tabl it can be deduced as follows –

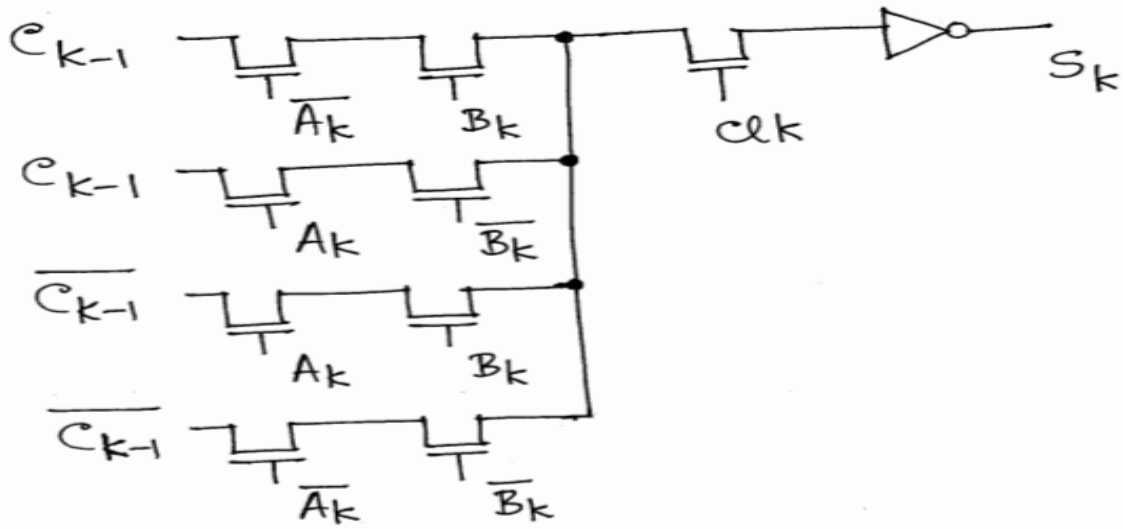
i) When $A_k = B_k$, $S_k = C_{k-1}$
else, $S_k = \overline{C_{k-1}}$

ii) When $A_k = B_k$, $C_k = A_k = B_k$
else, $C_k = C_{k-1}$

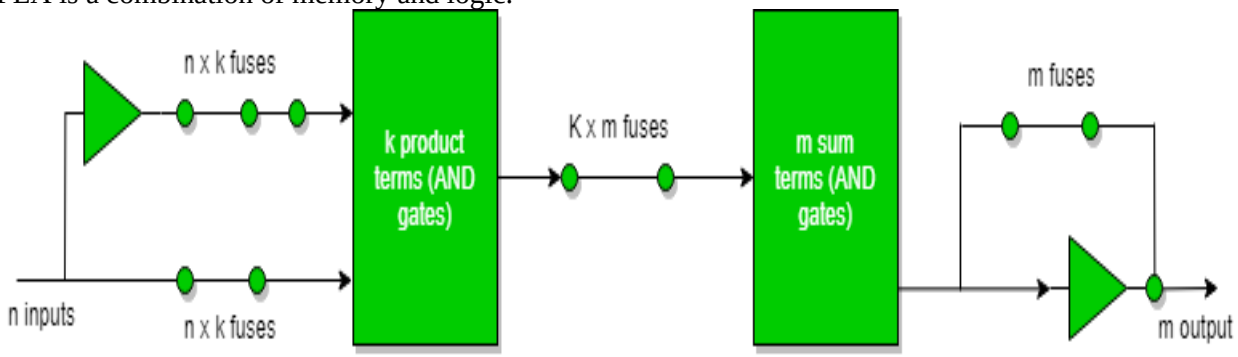
Using these expressions, the circuit for the full-adder can be written using the pass-transistors, as shown below –



But the sum-bit requires buffering as well as the clocking overhead. Hence, the ckt. for sum-bit is altered as –



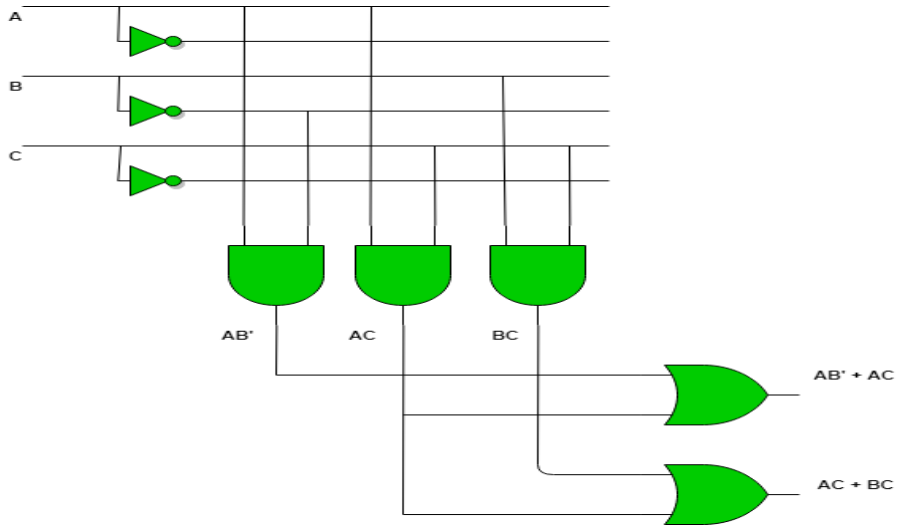
8.a Programmable Logic Array (PLA) is a fixed architecture logic device with programmable AND gates followed by programmable OR gates.PLA is basically a type of programmable logic device used to build reconfigurable digital circuit. PLDs have undefined function at the time of manufacturing but they are programmed before made into use. PLA is a combination of memory and logic.



A	B	C	F1	F2
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1

F1 = AB'C' + AB'C + ABC' + ABC
on simplifying we get : F1 = AB' + AC

F2 = A'BC + AB'C + ABC
on simplifying we get: F2 = BC + AC



PLA is used for implementation of various combinational circuits using buffer, AND gate and OR gate. In PLA, all the minterms are not realized but only required minterms are implemented. As PLA has programmable AND gate array and programmable OR gate array, it provides more flexibility but disadvantage is, it is not easy to use.

Applications:

- PLA is used to provide control over data path.
- PLA is used as a counter.
- PLA is used as decoders.
- PLA is used as a BUS interface in programmed I/O

8.b

Keywords: task, endtask

- Must be used if the procedure has
- any timing control constructs
- zero or more than one output arguments
- May be on or more input arguments
- Task declaration and invocation

Declaration syntax

task <task_name>;

<I/O declarations>

<variable and event declarations>

begin // if more than one statement needed

<statement(s)>

end // if *begin* used!

endtask

- Task declaration and invocation
- Task invocation syntax

  <task_name>;

  <task_name> (<arguments>);

- input and inout arguments are passed into the task
- output and inout arguments are passed back to the invoking statement when task is completed
- I/O declaration in modules vs. tasks
  - Both used keywords: input, output, inout
  - In modules, represent ports
    - connect to external signals
  - In tasks, represent arguments
    - pass values to and from the task

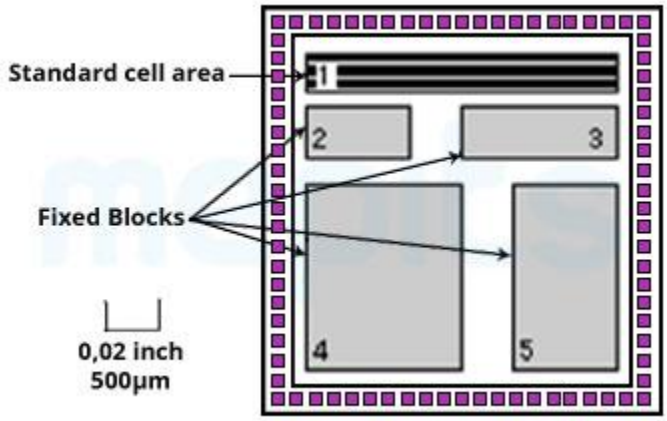
```
module operation;
parameter delay = 10;
reg [15:0] A, B;
reg [15:0] AB_AND, AB_OR, AB_XOR;
initial
    $monitor( …);

initial
begin
    …
end
always @(A or B)
begin
    bitwise_oper(AB_AND, AB_OR,        AB_XOR, A, B);
end

task bitwise_oper;
output [15:0]  ab_and, ab_or,                ab_xor;
input [15:0] a, b;
begin
     #delay ab_and = a & b;
     ab_or = a | b;
     ab_xor = a ^ b;
end
endtask
endmodule
```

9.a

- A cell-based ASIC (cell-based IC, or CBIC pronounced sea-bick) uses predesigned logic cells (AND gates, OR gates, multiplexers, and flip-flops, for example) known as standard cells. • One can apply the term CBIC to any IC that uses cells, but it is generally accepted that a cell-based ASIC or CBIC means a standard-cell based ASIC.
- The standard-cell areas (also called flexible blocks) in a CBIC are built of rows of standard cells like a wall built of bricks. The standard-cell areas may be used in combination with microcontrollers or even microprocessors, known as mega cells. Mega cells are also called mega functions, full-custom blocks, system-level macros (SLMs), fixed blocks, cores, or Functional Standard Blocks (FSBs).

	• A cell-based ASIC (CBIC) die with a single standard-cell area (a flexible block) together with four fixed blocks • The ASIC designer defines only the placement of the standard cells and the interconnect in a CBIC. However, the standard cells can be placed anywhere on the silicon; this means that all the mask layers of a CBIC are customized and are unique to a particular customer. • The advantage of CBICs is that designers save time, money, and reduce risk by using a predesigned, pretested, and pre characterized standard-cell library. • In addition each standard cell can be optimized individually. During the design of the cell library each and every transistor in every standard cell can be chosen to maximize speed or minimize area. • The disadvantages are the time or expense of designing or buying the standard-cell library and the time needed to fabricate all layers of the ASIC for each new design. 

9.b	Code for 4 bit CLA
	module CLA(a,b,ci,co,s);
	input [3:0]a,b;
	output [4:0]s;
	input ci;
	output co;
	wire [3:0]G,P,C;
	assign G = a&b;
	assign P = a^b;
	assign co=G[3]+ (P[3]&G[2]) + (P[3]&P[2]&G[1]) + (P[3]&P[2]&P[1]&G[0]) + (P[3]&P[2]&P[1]&P[0]&ci);
	assign C[3]=G[2] + (P[2]&G[1]) + (P[2]&P[1]&G[0]) + (P[2]&P[1]&P[0]&ci);
	assign C[2]=G[1] + (P[1]&G[0]) + (P[1]&P[0]&ci);
	assign C[1]=G[0] + (P[0]&ci);
	assign C[0]=ci;
	assign s = {co,P^C};
	endmodule
	Code for Test Bench
	module fi;
	reg [3:0] a;
	reg [3:0] b;
	reg ci;
	wire co;
	wire [4:0] s;
	CLA uut (.a(a),.b(b),.ci(ci),.co(co),.s(s));
	initial begin
	// Initialize Inputs
	a = 4'b1101;
	b = 4'b1011;
	ci = 0;
	end
	endmodule