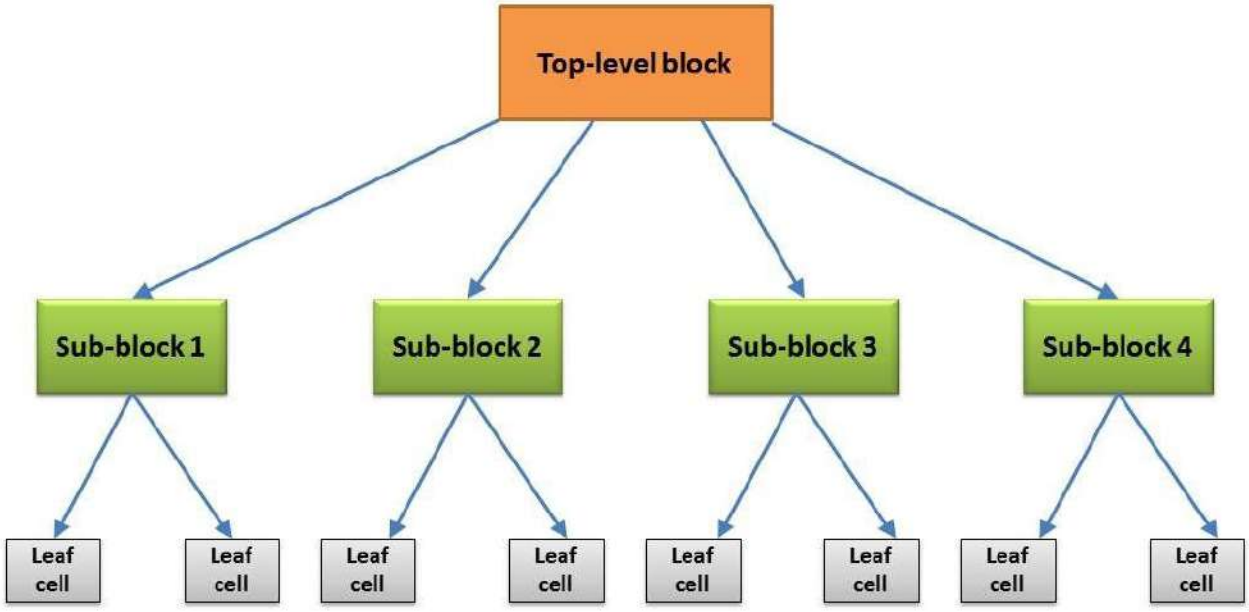
UNIT-IV

Subsystem Design and Layout-1: Switch logic, pass transistor, Gate logic, inverter, NAND gates, NOR gates, pseudo nMOS, Dynamic CMOS, Examples of structured design: Parity generator, Bus arbitration, multiplexers, logic function block, code converter.

****

Subsystem Design and Layout: A digital system contains several functional blocks, which are known as subsystems. For example, an arithmetic logic unit (ALU) contains functional blocks, or subsystems such as adders, subtractors, multipliers, dividers, comparators, and shift registers.

- For designing digital systems in MOS technology there are two ways of building the circuits. They are Switch logic(Pass transistor,TG) and Gate logic (NMOS/COMS/BiCMOS) deals with designing of subsystems, which is a small part in a larger system called *leaf cell*.
- The most basic leaf cell of any digital system is the logic gates and these are seen in different technologies like nMOS, CMOS and BiCMOS.
- While designing high regularity should be maintained. This indicates detailed designing of few leaf cells which can be replicated and interconnected to form system.



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

- Partitioning of the system so that there are clear subsystems with minimum interdependence and minimum complexity of interconnection between them.
- The design within the subsystems should be simple which helps in cellular design concept. This helps the systems in having few standard cells which are replicated to form high regular structures.

General Considerations of Subsystem Design:

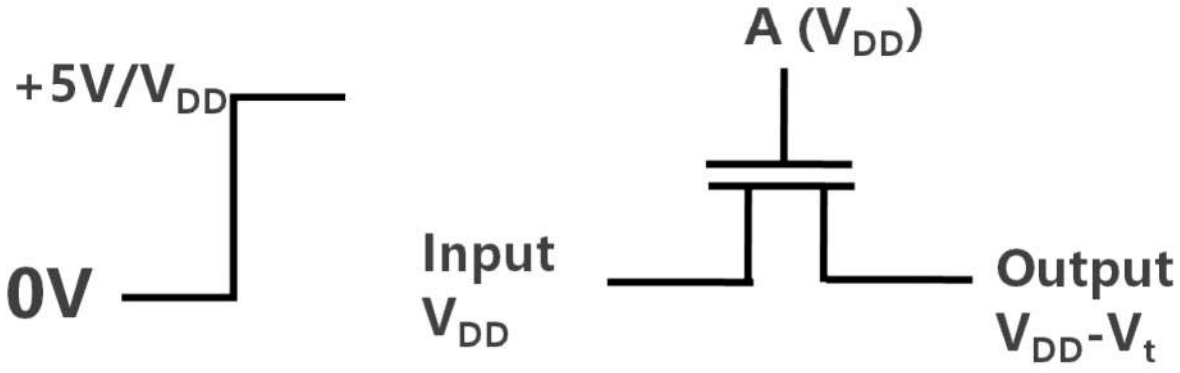
- Lower unit cost.
- Higher reliability.
- Lower power dissipation, lower weight and lower volume.
- Better performance.
- Enhanced repeatability.
- Possibility of reduced design/development periods.

SWITCH LOGIC

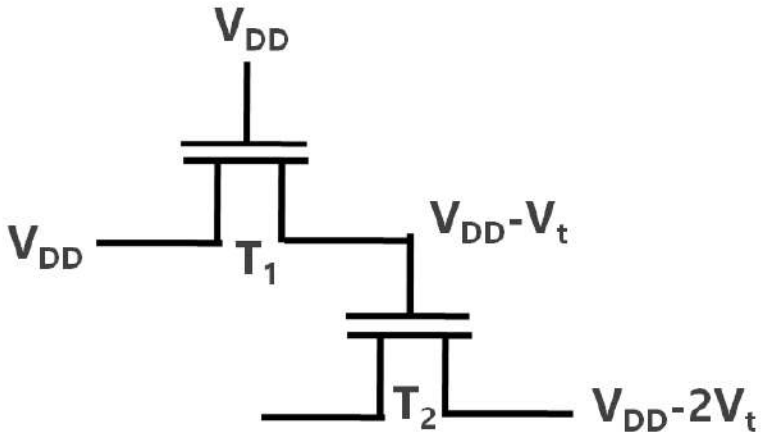
- The switch logic is built on 'pass transistors' or on 'transmission gates'. Switch logic is fast for small arrays and draws no static current from the supply rails. Hence power dissipation of such circuits is small because current flows only on switching.
- Pass transistor can be used as a switch in passing the signals. Switch logic arrangements using basic OR and AND connections along with other arrangements

SWITCH LOGIC (PASS TRANSISTOR)

- It requires only one transistor and one gate terminal.
- It transmits logic 0 well, but when V_{DD} is applied to the drain the voltage at the source is V_{DD}-V_t.



- When using nMOS switch logic no pass transistor gate input may be driven through one or more pass transistors.



- Since the signal out of pass transistor T_1 does not reach a full logic 1 by threshold voltage effects signal is degraded by below a true logic 1, this degraded voltage would not permit the output of T_2 to reach an acceptable logic 1 level.

Advantage

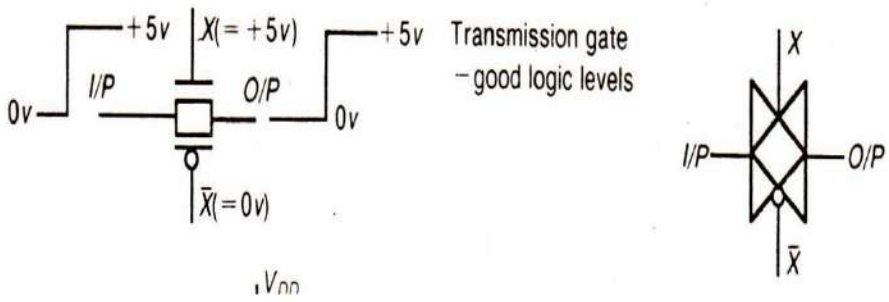
- Requires minimum geometry.
- Do not dissipate standby power, since they do not have a path from supply to ground.

Disadvantages

- Degradation in the voltage levels due to undesirable threshold voltage effects.
- Never drive a pass transistor with the output of another pass transistor

TRANSMISSION GATE

A MOS transistor switch built by connecting n-type and p-type transistor in parallel. The switch transmits logic 0 and logic 1 equally well from source to drain. So it is called transmission gate. Thus current can flow through this element in either direction.



Operation

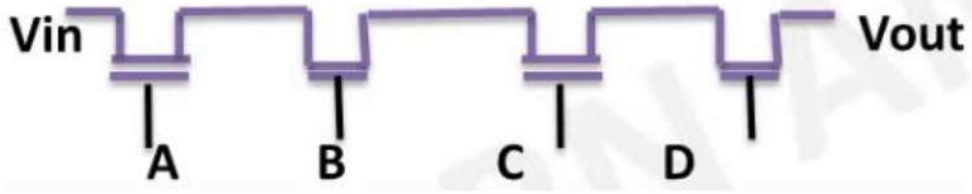
- When the gate input to the nMOS transistor is '0' and the complementary '1' is gate input to the pMOS , thus both are turned off.
- When gate input to the nMOS is '1' and its complementary '0' is the gate input to the pMOS , both are turned on and passes any signal '1' and '0' equally without any degradation.
- The use of transmission gates eliminates the undesirable threshold voltage effects

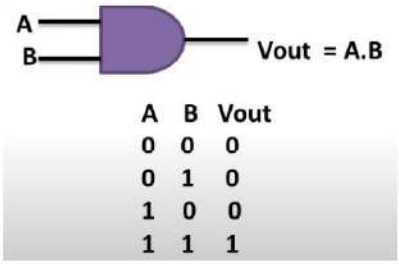
Advantages

- Transmission gates eliminate the signal degradation in the output logic levels.
- Transmission gate consists of two transistors in parallel and except near the positive and negative rails.

Disadvantages

- Transmission gate requires more area than nMOS pass circuitry.
- Transmission gate requires complemented control signals.



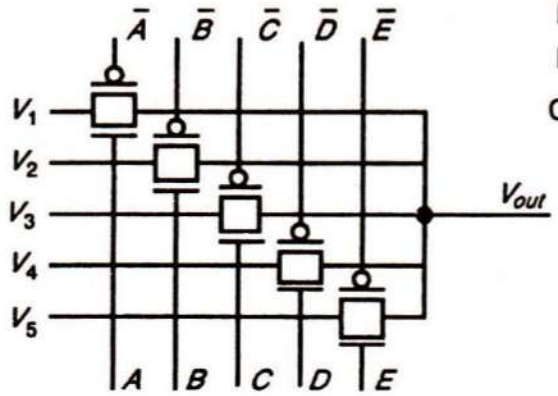


$V_{out} = V_{in}$ when $A.B.C.D = 1$

$V_{out} = V_{in}$ when $A.B.C.D.E.F.G.H = 1$

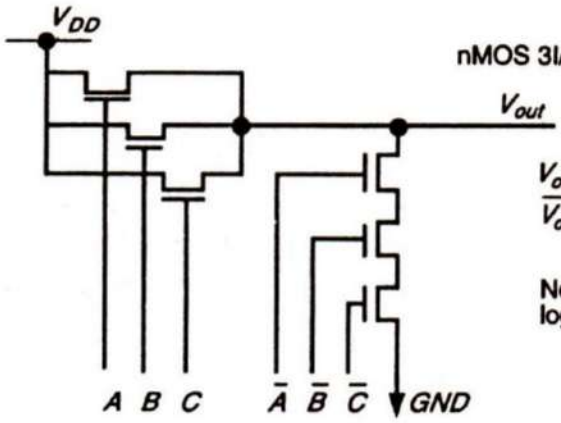
V_{out} = ? when $A.B.C.D.E.F.G.H \neq 1$

CMOS 5-way selector



$V_{out} = V_1.A + V_2.B + V_3.C + V_4.D + V_5.E$ assuming A, B, C, D and E are mutually exclusive

(V_{out} logic levels will not be degraded by V_t effects)



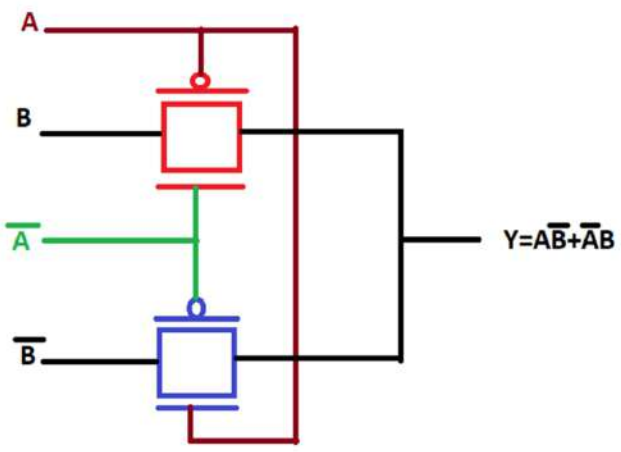
nMOS 3I/P *Or* gate

$V_{out} = A + B + C$ (degraded by V_t)

$\overline{V_{out}} = \bar{A}.\bar{B}.\bar{C}$.

Note the arrangement to satisfy both logic '1' and logic '0' states

Implement 2-input XOR gate using transmission gates.

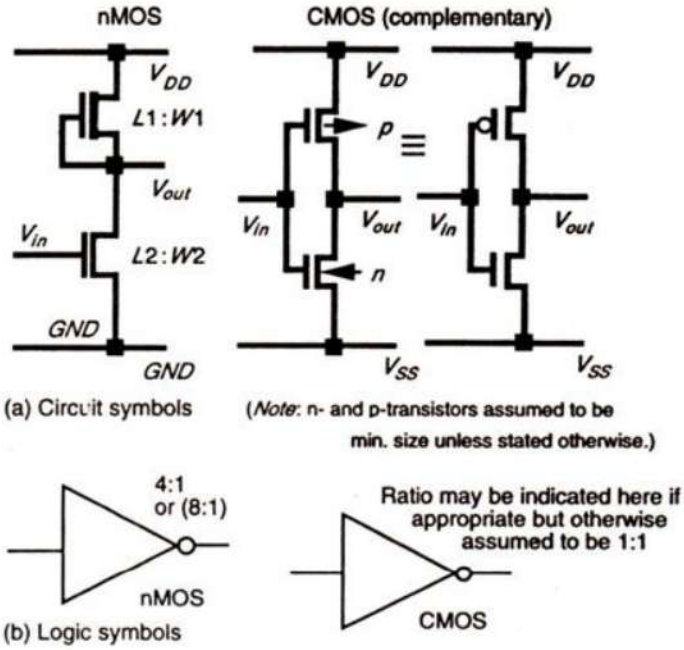


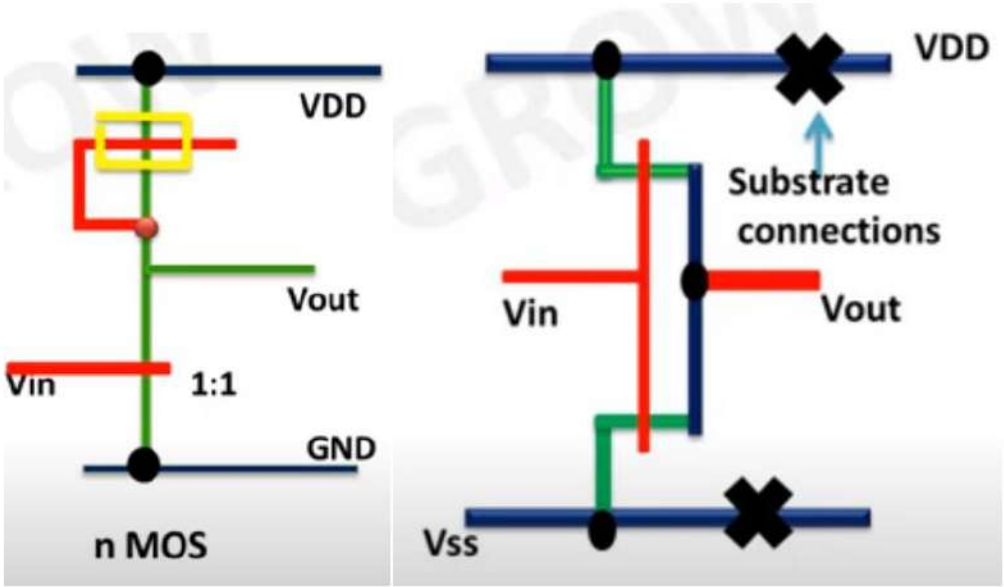
Gate Logic (Restoring Logic)

Gate logic is based on the general arrangement of inverter as it is the simplest gate. The inverter can be constructed with nMOS, CMOS or BiCMOS technology. Similar to this NAND and NOR can be constructed. Also AND and OR can be constructed with an inverter for NAND and NOR respectively.

Inverter using NMOS and CMOS

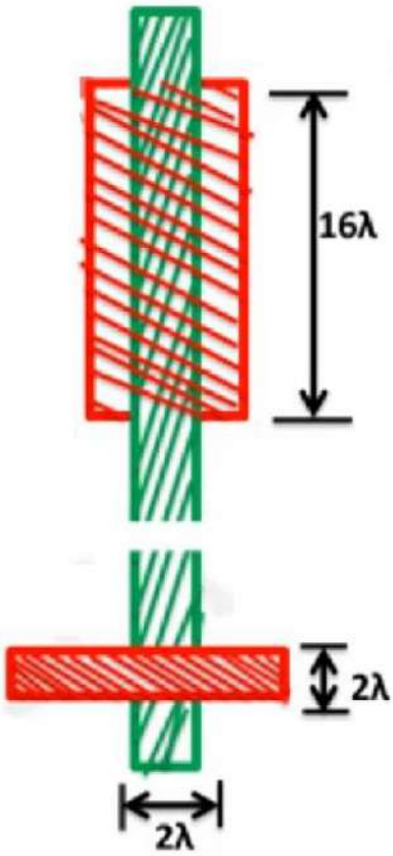
Inverters are also employed to complement and restore logic levels that have been degraded (e.g. because they have passed through pass transistors).





In achieving the desired pull-up to pull-down ratio, several possibilities emerge, two of which are illustrated in Figures for an 8:1 nMOS inverter. Note the effect that the different approaches have on **power dissipation P_d** and on **the area occupied** by the inverter. Also note the resistance and capacitance values.

The CMOS inverter carries no static current and thus has no power dissipation unless switching. The switching dissipation for fast CMOS logic circuits will be considerable.



Power Dissipation in nMOS inverter

Zp.u = Lp.u/Wp.u = 8

Rp.u = Zp.u x Rs = 80kΩ

Zp.d = Lp.d/Wp.d = 1

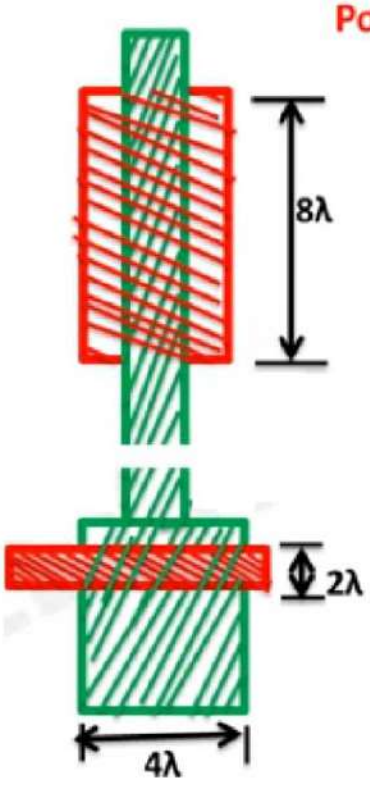
Rp.d = Zp.d x Rs = 10kΩ

on power dissipation Pd = $\frac{V^2}{Rp.u+Rp.d}$ = 0.28mW

Input capacitance = 1□Cg

8:1 nMOS inverter (minimum size p.d.)

Power Dissipation in nMOS inverter



Zp.u = Lp.u/Wp.u = 4

Rp.u = Zp.u x Rs = 40kΩ

Zp.d = Lp.d/Wp.d = 1/2

Rp.d = Zp.d x Rs = 5kΩ

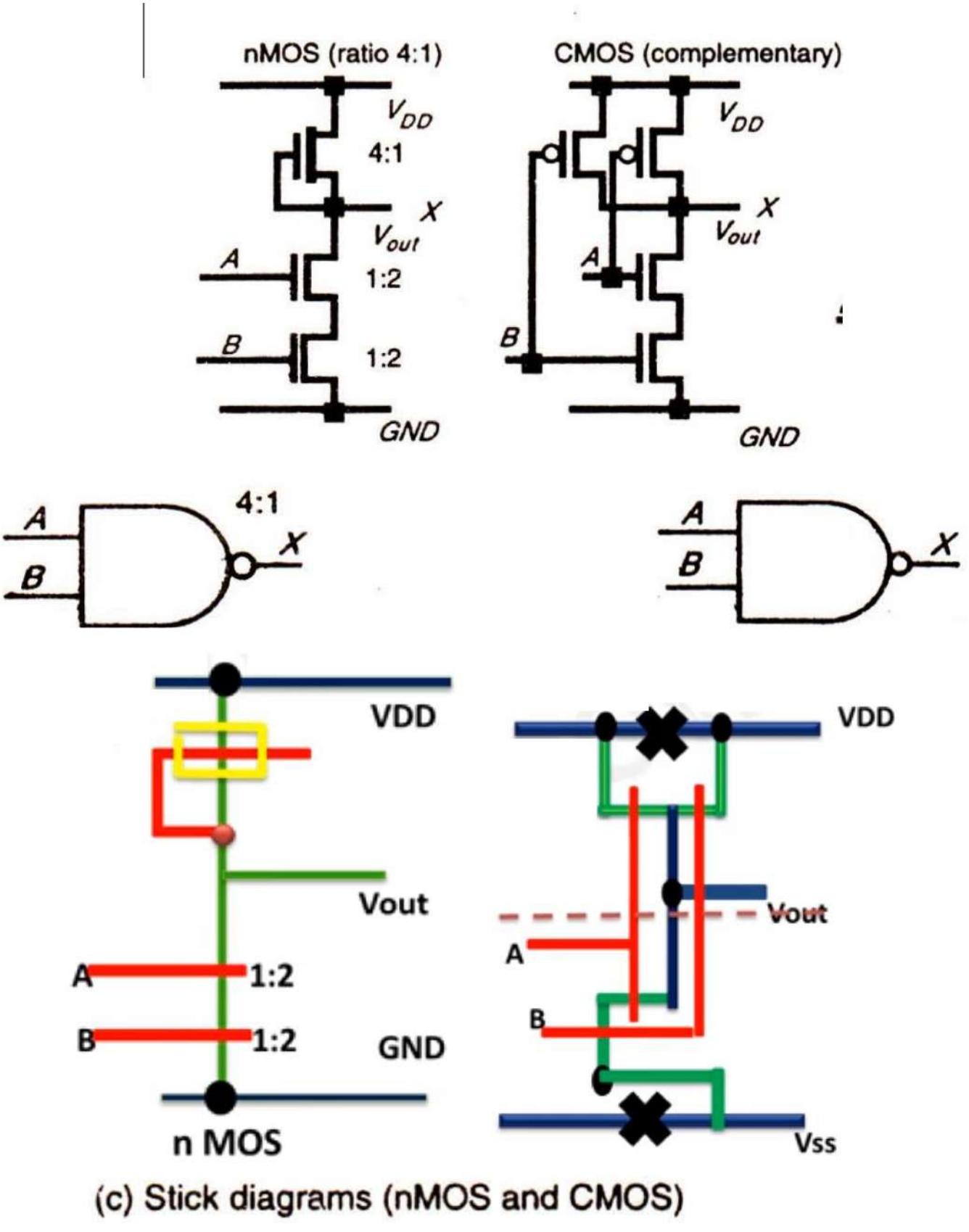
on power dissipation Pd = $\frac{V^2}{Rp.u+Rp.d}$ = 0.56mW

Input capacitance = 2□Cg

An alternative 8:1 nMOS inverter.

Two-Input nMOS and CMOS *NAND* Gates

Two-input *Nand* gate arrangements are given in Figure. The nMOS (and pseudo-nMOS) L: *W* ratios should be carefully noted since they must be chosen to achieve the desired overall *Zp.u/ Z p d* ratio (where *Zpd.* is contributed in this case by *both* input transistors in series).



(c) Stick diagrams (nMOS and CMOS)

NAND Gate two significant factors:

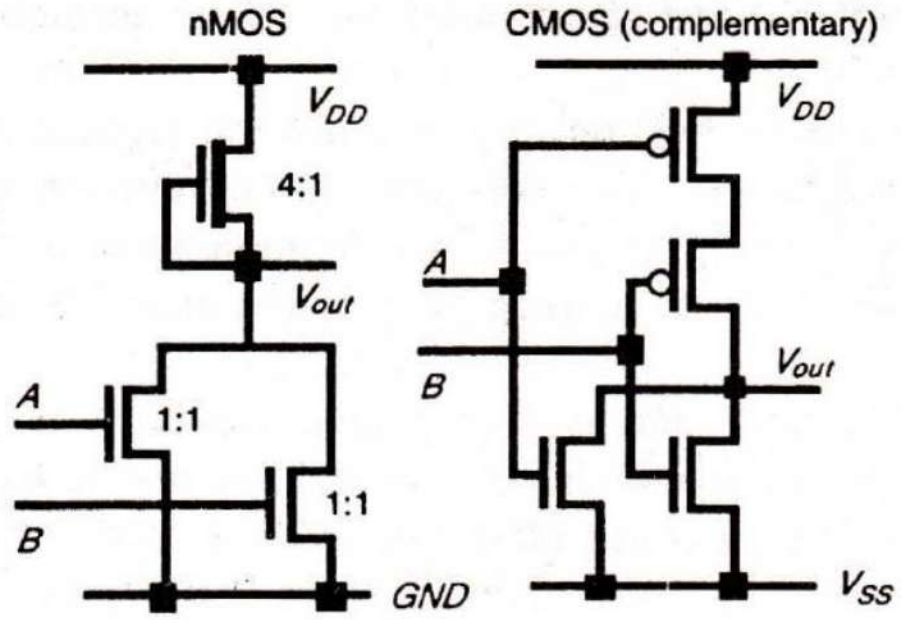
1. nMOS *Nand* gate *area requirements* are considerably greater than those of a corresponding nMOS inverter, since not only must pull-down transistors be added in series to provide the desired number of inputs, but, as inputs are added, so must there be a corresponding adjustment of the length of the pull-up transistor channel to maintain the required overall ratio .

. 2. nMOS *Nand* gate *delays* are also increased in direct proportion to the number of inputs added. If each pull-down transistor is kept to minimum size (2λ x 2 λ), then each will present 1 Cg at its input, but if there are *n* such inputs, then the length and *resistance* of the pull-up transistor must be increased by a factor of *n* to keep the correct ratio. Thus, delays associated with the nMOS *Nand* are

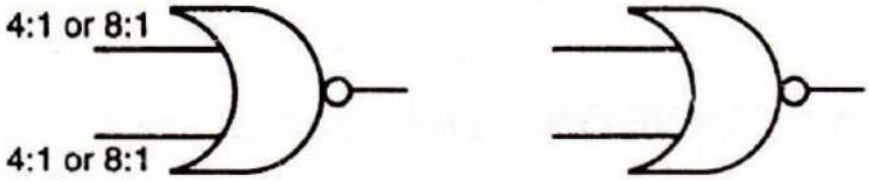
$$\tau_{Nand} = n\tau_{inv}$$

where *n* is the number of inputs and T_{inv} is the corresponding nMOS inverter delay.

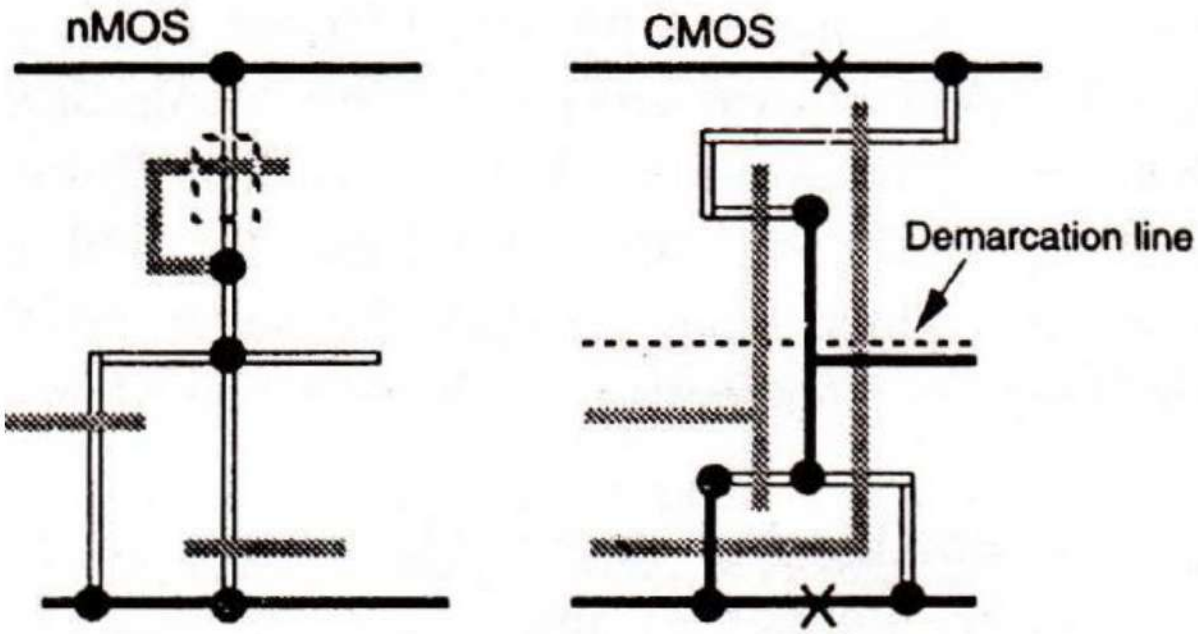
Two-Input nMOS and CMOS NOR Gates



(a) Circuit diagrams



(b) Logic symbol



The area occupied by the nMOS (or pseudo-nMOS) *Nor* gate is reasonable since the pull-up transistor dimensions are unaffected by the number of inputs accommodated. In consequence, the *Nor* gate is as fast as the corresponding inverter and is the preferred inverter-based nMOS (or pseudo-nMOS) logic gate when a choice is possible. Obviously, the ratio between *zp.u.* and *zp.d.* of any one leg must be appropriate to the source from which that input is driven for nMOS design.

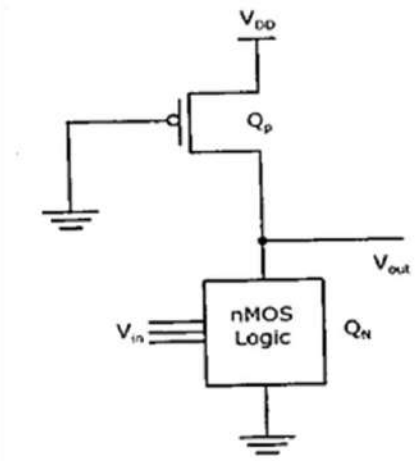
The CMOS *Nor* gate consists of a pull-up p-transistor-based structure, which implements the logic 1 conditions and a complementary n-transistor arrangement to implement the logic 0 -conditions at the output. In the case of the *Nor* gate, the p-structure consists of transistors in series, one for each input, while the *n* pull-down arrangement has as many transistors in parallel as there are inputs to the *Nor* gate. Thus, the already predominant resistance of the p-devices is aggravated in its effect by the number connected in series. Risetime and fall-time asymmetry on capacitive loads is thus increased and there will also be a shift. in the transfer *(Vin or Vo)* characteristic which will reduce noise immunity. For these reasons, CMOS (complementary logic) *Nor* gates with more than two inputs may require adjustment of the p- and/or n-transistor geometries *(L: W* ratios).

Other Forms of CMOS Logic

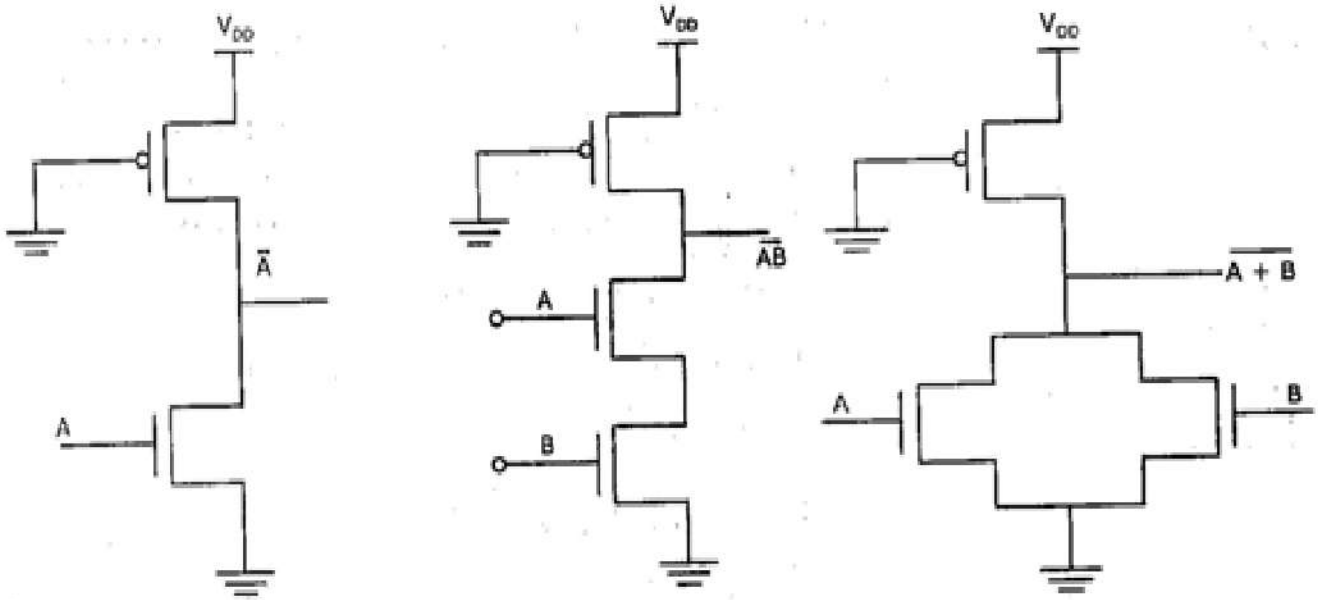
The availability of both n- and p-transistors makes it possible for the CMOS designer to explore and exploit various alternatives to inverter-based CMOS logic.

Pseudo-NMOS Logic

- In Pseudo NMOS logic, the pull up network is realized by a single PMOS transistor. The gate terminal of the PMOS transistor is connected to the ground. It remains permanently in the ON state. Depending on the input combinations, output goes low through the pull down transistor.



- Here, only the nMOS logic (Qn) is driven by the input voltage, while the gate of p-transistor(Qp) is connected to ground or substrate and Qp acts as an active load for Qn. Except for the load device, the pseudo-nMOS gate circuit is identical to the pull-down network(PDN) of the complementary CMOS gate.
- The realization of logic circuits using pseudo-nMOS logic is as shown in figure.



(a) Pseudo nMOS Inverter (b) Pseudo nMOS NAND (c) Pseudo nMOS NOR

Advantages

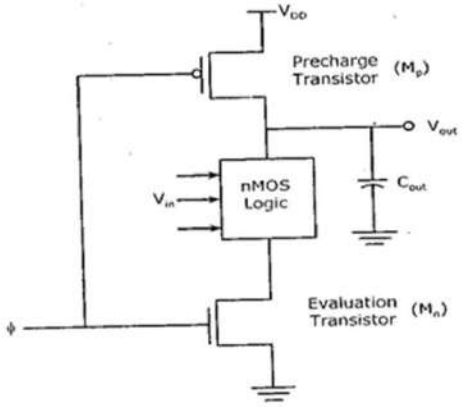
- Uses less number of transistors as compared to CMOS logic.
- Geometrical area and delay gets reduced as it requires less transistors.
- Low power dissipation.

Disadvantages

- The main drawback of using a pseudo nMOS gate instead of a CMOS gate is that the always on PMOS load conducts a steady current when the output voltage is lower than V_{DD}.
- Layout problems are critical.

DYNAMIC CMOS LOGIC

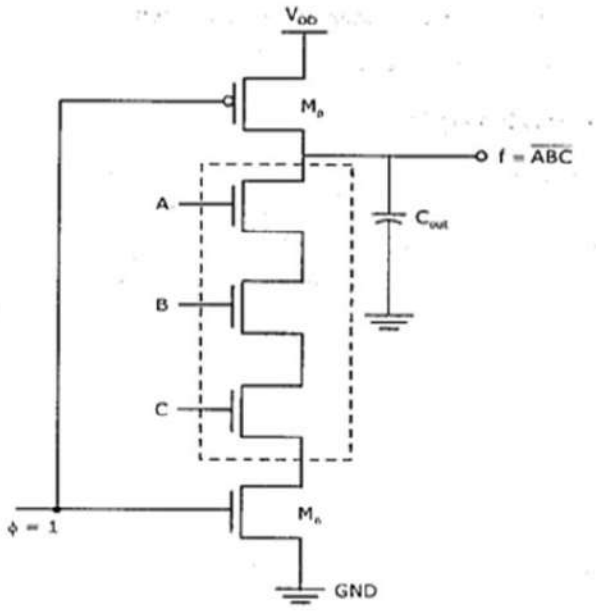
- A dynamic CMOS logic uses charge storage and clocking properties of MOS transistors to implement logic operation. Here the clock ø drives nMOS evaluation transistor and pMOS precharge transistor.



The gate (clock ø) defines two phases, evaluation and precharge phase during each clock cycle.

Working

- When clock ø = 0 the circuit is in precharge phase with the pMOS device Mp ON and the evaluation nMOS Mn OFF. This establishes a conducting path between V_{DD} and the output allowing C_{out} to charge to a voltage $V_{out} = V_{DD}$. Mp is often called the precharge FET.
- When clock ø = 1 the circuit is in evaluation phase with the pMOS device Mp OFF and the evaluation nMOS Mn ON. If the logic block acts like a closed switch the C_{out} can discharge through logic array and Mn, this gives a final result of $V_{out} = V_{DD}$
- The logic formation is formed by three series connected FETs (3-input NAND gate) is shown in figure.



Advantages

- Low power dissipation.
- Large noise margin.
- Small area due to less number of transistors.

EXAMPLES OF STRUCTURED DESIGN

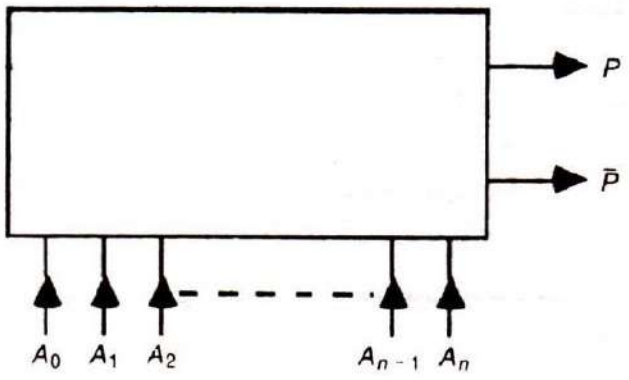
Parity Generator

The parity generating technique is one of the most widely used error detection techniques for the data transmission.

- In digital systems, when binary data is transmitted and processed, data may be subjected to *noise.*
- Hence, **parity bit** is added to the word containing data in order to make number of 1s either even or odd. During the transmission of binary data, the message containing the data bits along with parity bit is transmitted from transmitter node to receiver node.
- At the receiving end, the number of 1s in the message is counted and if it *doesn't match with the transmitted one, then it means there is an error in the data.*
- A parity generator is a combinational logic circuit that generates the parity bit in the transmitter. On the other hand, a circuit that checks the parity in the receiver is called parity checker. A combined circuit or devices of parity generators and parity checkers are commonly used in digital systems to detect the single bit errors in the transmitted data word.
- In even parity, the added parity bit will make the total number of 1s an even amount whereas in odd parity the added parity bit will make the total number of 1s odd amount.

Parity Generator: It is combinational circuit that accepts an n-1 bit stream data and generates the additional bit that is to be transmitted with the bit stream. This additional or extra bit is termed as a parity bit.

This circuit shows the parity of binary of parity numbers or words. In VLSI a circuit to be designed to indicate parity of a binary number is shown in the Fig. for an (n+1) bit input.

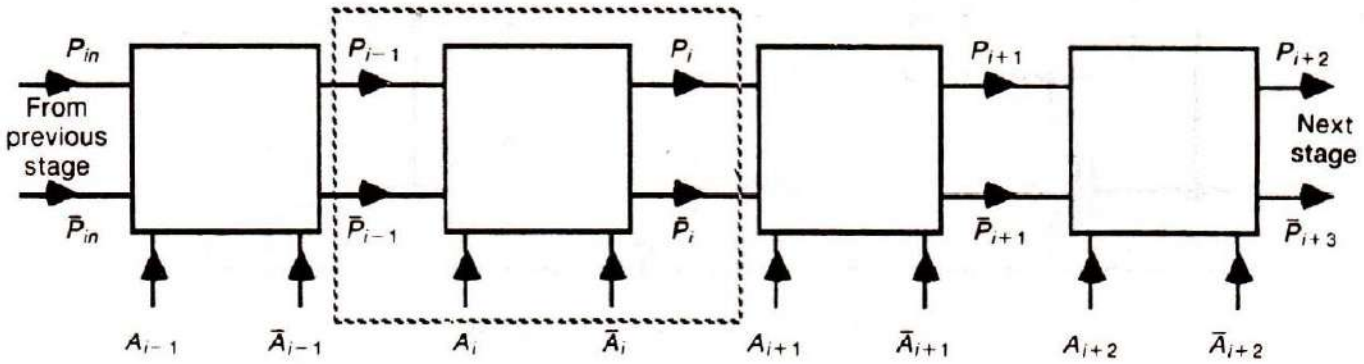


Note: $P = \begin{cases} 1 & \text{Even number of 1s at input} \\ 0 & \text{Odd number of 1s at input} \end{cases}$

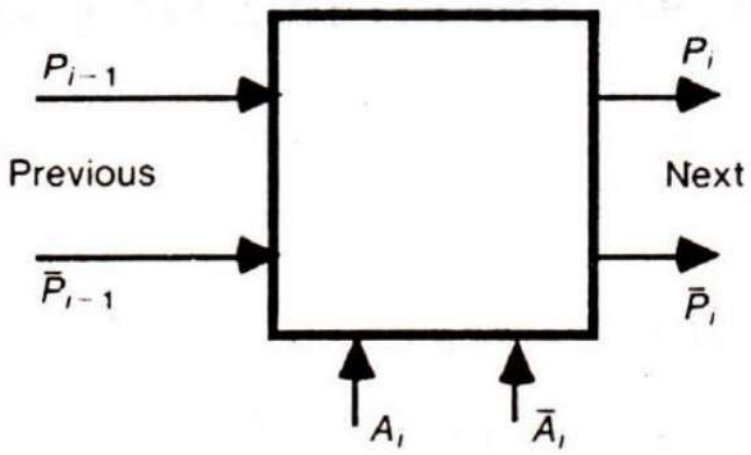
Parity generator basic block diagram.

Since the no. of bits is undefined, a general solution on cascadable bit- wise and a regular structure is shown in Fig.

- A standard or basic one-bit cell from which an n-bit parity generator may be formed. The standard/ basic cell is shown in the Fig.
- The parity information is passed from one cell to next and the parity information is modified or retained depending on the input lines Ai and A'i



Parity generator-structured design approach.



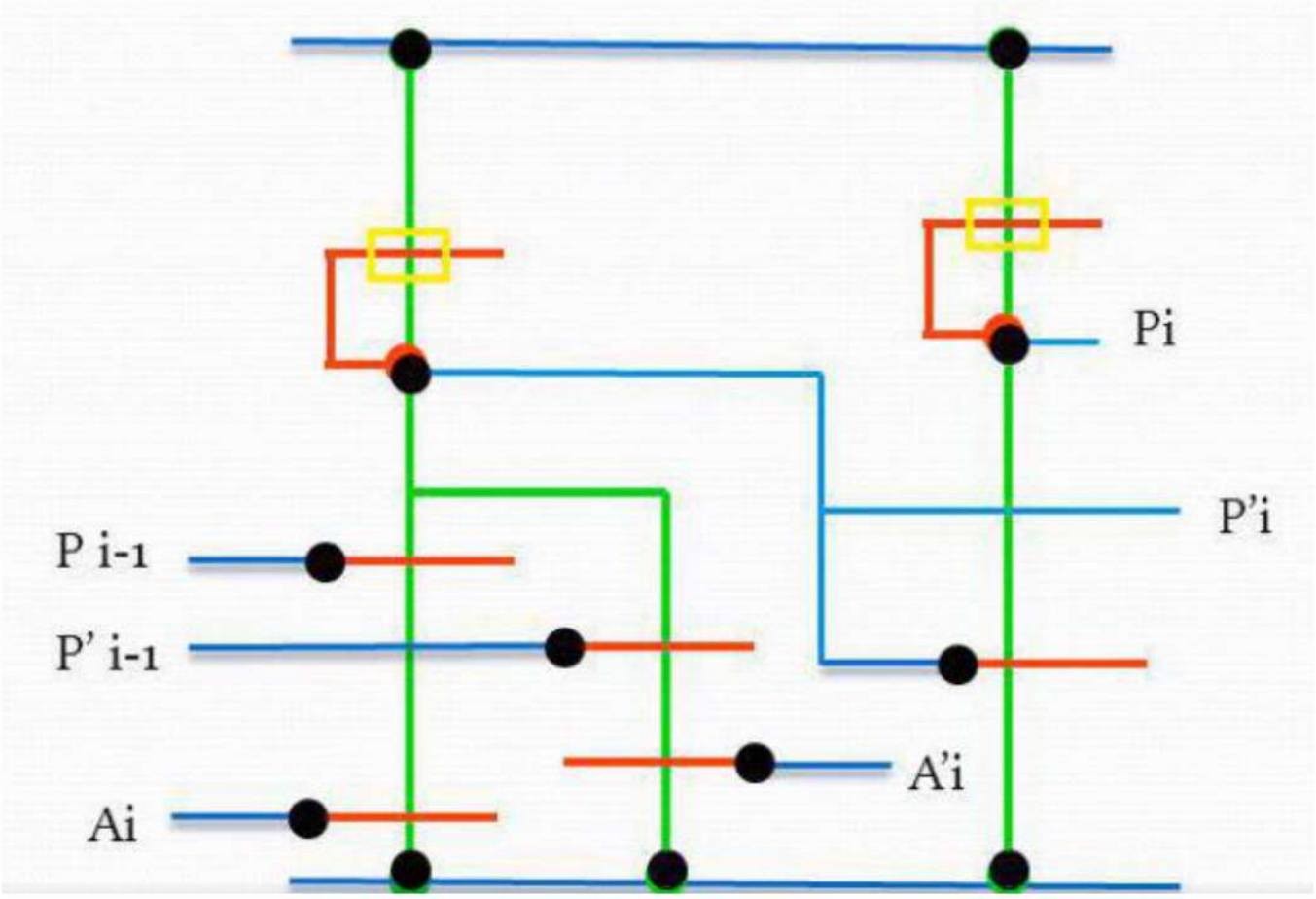
Parity generator-basic one-bit cell

- If Ai = 1, parity output Pi will change to P'i-1 (i.e., if Ai [input] =1 and Pi-1 [previous parity] =1, the output parity Pi will change to 0)
- If Ai = 0, output parity Pi will remain in the same state of Pi-1 (i.e., if Ai [input] =0 and Pi-1 [previous parity] =1, the output parity Pi will remain as 1)
- Suitable arrangement for such cell is implemented using the expression

Pi = P'i-1 Ai + Pi-1 A'i

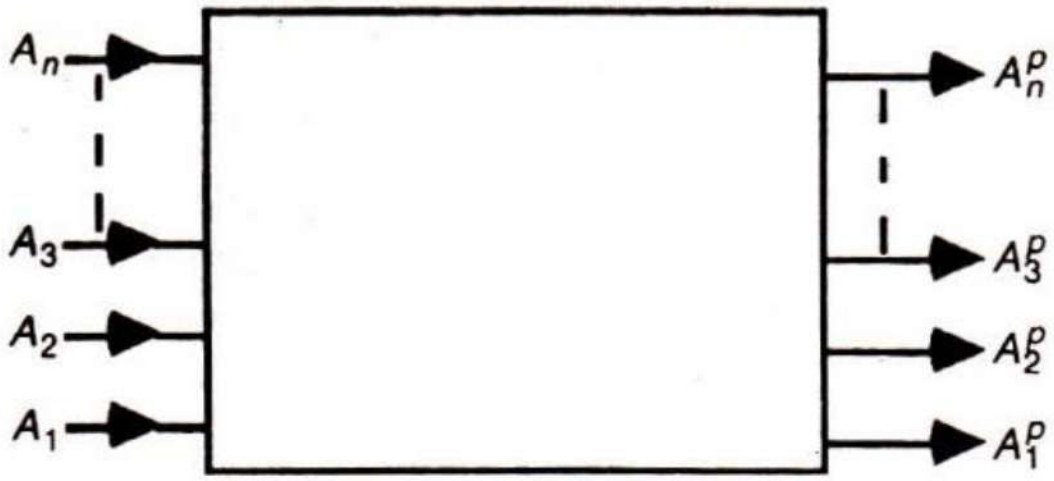
$A_i = 1$ parity is changed, $P_i = \bar{P}_{i-1}$

$A_i = 0$ parity is unchanged, $P_i = P_{i-1}$



Bus Arbitration Logic for n-line Bus

If the highest priority line An is Hi (Logic 1), then output line A_n^P will be High and all other output lines Low (Logic 0), irrespective of the state of the other input lines A1 -- -A_{n-1}. Similarly, .A^P_{n-1} will be Hi only when A_{n-1} is Hi and An is Lo; again the state of all input lines of lower priority (A 1 - - - An-2) will have no effect and all other output lines will be Lo.



This requirement can be expressed algebraically as follows:

$$A_n^P = A_n$$

$$A_{n-1}^P = \bar{A}_n \cdot A_{n-1} \qquad [\bar{A}_{n-1}^P = A_n + \bar{A}_{n-1}]$$

$$A_{n-2}^P = \bar{A}_n \cdot \bar{A}_{n-1} \cdot A_{n-2} \qquad [\bar{A}_{n-2}^P = A_n + A_{n-1} + \bar{A}_{n-2}]$$

$$\vdots \qquad\qquad \vdots$$

$$A_n^P = \bar{A}_n \cdot \bar{A}_{n-1} \cdot \bar{A}_{n-2} \quad \ldots\ldots \quad \bar{A}_3 \cdot \bar{A}_2 \cdot A_1 \quad \text{(etc.)}$$

Truth table

A_n	.....	A_3	A_2	A_1	A_n^P	.....	A_3^P	A_2^P	A_1^P
0	.....	0	0	0	0	.....	0	0	0
0		0	0	1	0	.....	0	0	1
0		0	1	X	0		0	1	0
0		1	X	X	0		1	0	0
.		.	.	.	.		.	.	.
.		.	.	.	.		.	.	.
.		.	.	.	.		.	.	.
1	.....	X	X	X	1	.....	0	0	0

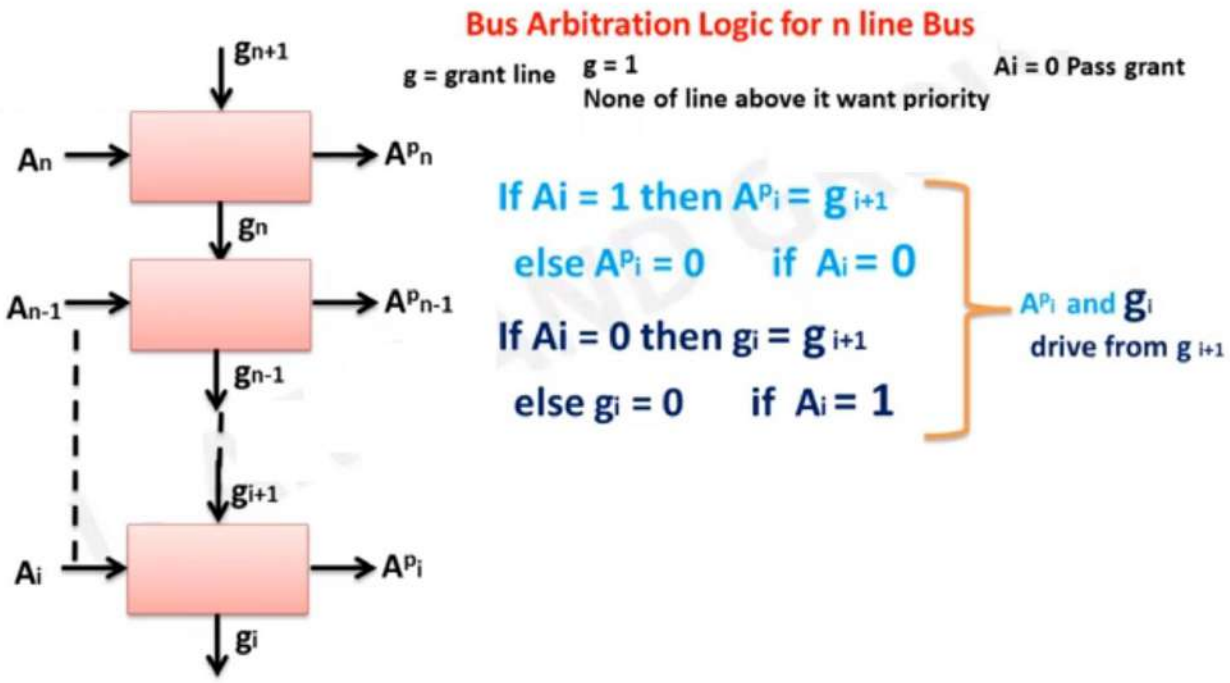
*X = Don't care

A regular structure having been arrived at, the requirements for each cell may be expressed as follows:

$$A_i^P = \begin{cases} g_{i+1} & \text{if } A_i = 1 \\ \text{or } 0 & \text{otherwise} \end{cases}$$

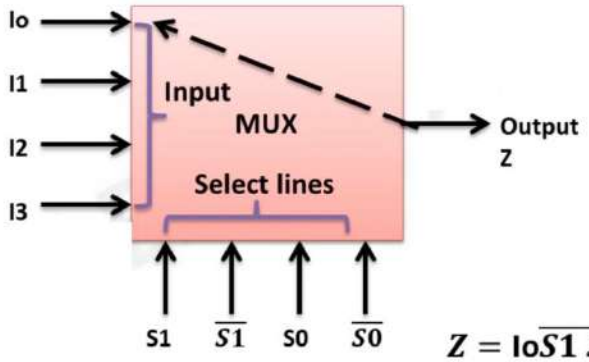
$$g_i = \begin{cases} 0 & \text{if } A_i = 1 \\ \text{or } g_{i+1} & \text{otherwise} \end{cases}$$

The art of arriving at conveniently expressed relationships which allow a structured design is one which must be cultivated and it is often helped by adopting an 'if, then, else (or otherwise)' approach. The solution to the problem under consideration can be formulated after expressing the need of each cell in words:



Multiplexer (Data Selector)

- A multiplexer or mux is a combinational circuits that selects several analog or digital input signals and forwards the selected input into a single output line.
- A multiplexer of 2^n inputs has n selected lines, are used to select which input line to send to the output.



S1	S0	Z
0	0	Io
0	1	I1
1	0	I2
1	1	I3

$$Z = Io\overline{S1}\,\overline{S0} + I1\,\overline{S1}\,S0 + I2\,S1\,\overline{S0} + I3\,S1\,S2$$

Now let's look at the logic circuitry required to perform this multiplexing operation. The data output is equal to the state of the *selected* data input. You can therefore, derive a logic expression for the output in terms of the data input and the select inputs.

The data output is equal to D_0 only if $S_1 = 0$ and $S_0 = 0$: $Y = D_0\overline{S}_1\overline{S}_0$.

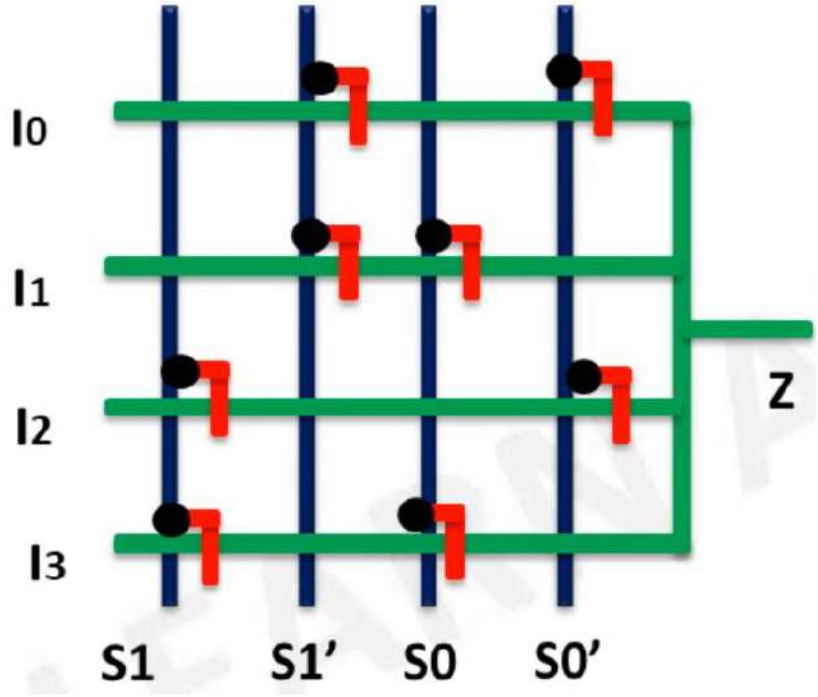
The data output is equal to D_1 only if $S_1 = 0$ and $S_0 = 1$: $Y = D_1\overline{S}_1S_0$.

The data output is equal to D_2 only if $S_1 = 1$ and $S_0 = 0$: $Y = D_2S_1\overline{S}_0$.

The data output is equal to D_3 only if $S_1 = 1$ and $S_0 = 1$: $Y = D_3S_1S_0$.

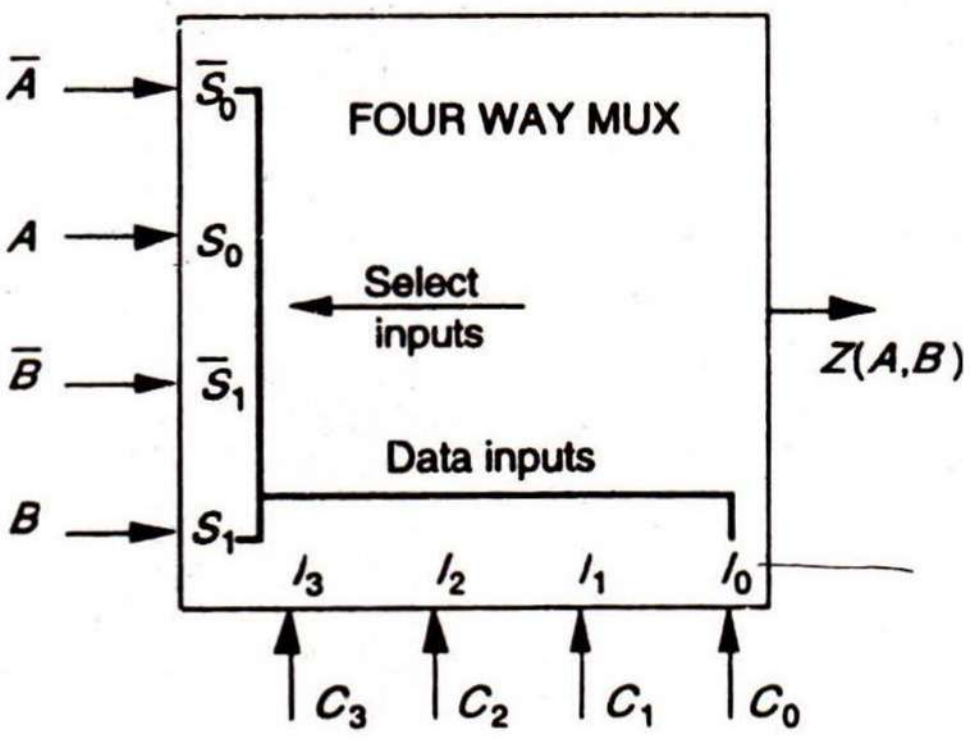
When these terms are ORed, the total expression for the data output is

$$Y = D_0\overline{S}_1\overline{S}_0 + D_1\overline{S}_1S_0 + D_2S_1\overline{S}_0 + D_3S_1S_0$$



Logic Function Block

An arrangement to generate any function of two variables (A, *B)* is readily formed from any form of four-way multiplexer. It will be seen that the required function is generated by driving the multiplexer select inputs from the required two variables *A* and *B* and by 'programming' the inputs I_0 *to* I_3 appropriately with Os and 1 s, as indicated in the figure. Larger multiplexers may be similarly employed to generate any function of up to four variables (16-way multiplexer).



INPUT PROGRAMMING				FUNCTION $Z(A,B)$	
C_3	C_2	C_1	C_0		
0	0	0	0	0	$Z=0$
0	0	0	1	$\bar{A}.\bar{B}$; $\overline{A+B}$	*Nor*
0	0	1	0	$A.\bar{B}$	
0	0	1	1	$\bar{B}$	Not *B*
0	1	0	0	$\bar{A}.B$	
0	1	0	1	$\bar{A}$	Not *A*
0	1	1	0	$A.\bar{B}+\bar{A}.B$	*Exclusive-Or*
0	1	1	1	$\bar{A}+\bar{B}$; $\overline{AB}$	*Nand*
1	0	0	0	$A.B$	*And*
1	0	0	1	$\bar{A}.\bar{B}+A.B$	Comparator
1	0	1	0	A	$O/P=A$
1	0	1	1	$A+\bar{B}$	
1	1	0	0	B	$O/P=B$
1	1	0	1	$\bar{A}+B$	
1	1	1	0	$A+B$	*Or*
1	1	1	1	1	$Z=1$

FIGURE 6.26 General logic function block (two variables).

A Four-line Gray Code to Binary Code Converter

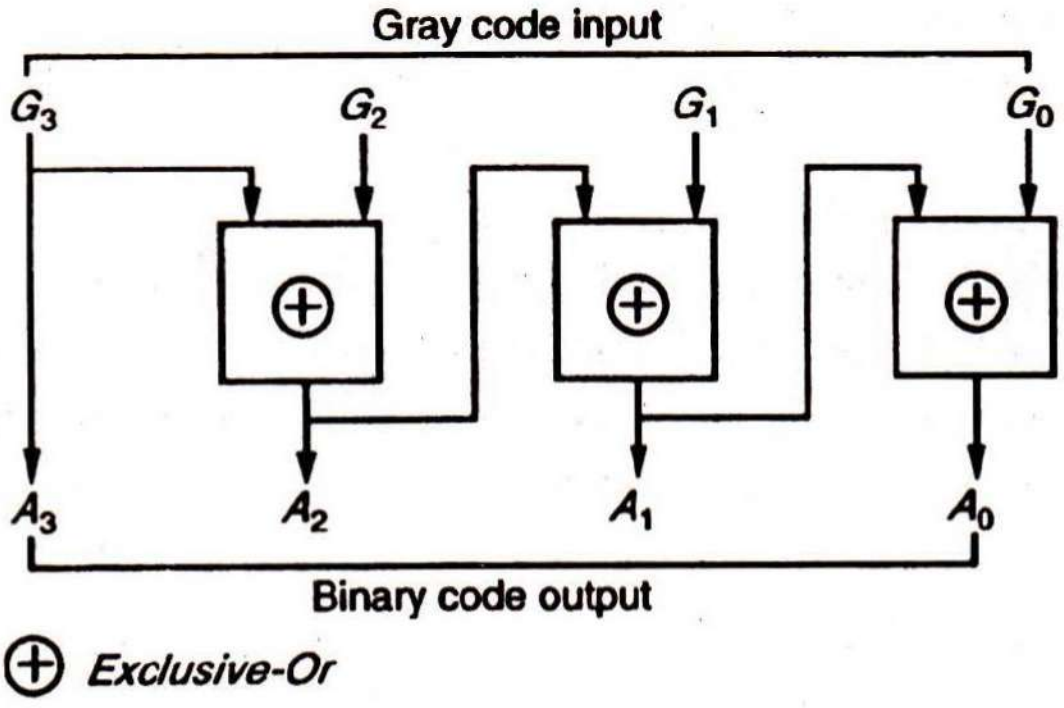
As a further exercise, which employs a very widely used logic arrangement (the *Exclusive Or* gate), consider the requirement for code conversion from Gray to binary. It will be seen that the following expressions relate the two codes:

$$\left.\begin{aligned} A_0 &= \bar{G}_0.A_1 + G_0.\bar{A}_1 \\ A_1 &= \bar{G}_1.A_2 + G_1.\bar{A}_2 \\ A_2 &= \bar{G}_2.A_3 + G_2.\bar{A}_3 \end{aligned}\right\} \quad \textit{Exclusive-Or} \text{ operations}$$

$$A_3 = G_3$$

Gray code				Binary code			
G_3	G_2	G_1	G_0	A_3	A_2	A_1	A_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	1	1	0	1	0
1	1	1	0	1	0	1	1
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	0	0	1	1	1	1	0
1	0	0	0	1	1	1	1

A suitable arrangement is set out in Figure, and the only detailed design required is that of a two input *Exclusive-Or* gate.



Ten Marks Questions

1. Draw stick diagram of inverter using NMOS & CMOS and calculate power dissipation of NMOS Inverter.
2. Draw the stick diagram of two input NMOS and CMOS NAND gates and mention the significant factors.
3. Draw the stick diagram of two input NMOS and CMOS NOR gates and mention the significant factors.
4. Write short notes on
    a) Pseudo NMOS Logic
    b) Dynamic CMOS Logic
5. Draw and explain block diagram of parity generator with stick diagram.
6. Draw and explain block diagram of multiplexer with stick diagram.
7. Explain the logic function block with truth table.
8. Explain the gray to binary code converter with truth table.