2.1 INTRODUCTION

Digital circuits used in all digital systems are fundamentally of two types, they are, combinational logic circuits and sequential logic circuits. In combinational logic circuits, the outputs are Boolean functions of single or multiple Boolean input variables that are presently applied to the system and is independent of previous output(s). In sequential circuits, the outputs(s) are not only dependent on present inputs, but also dependent on previous outputs.

In its general form, a combinational logic can be represented by a Multiple Input Single Output (MISO) system. The input variables are represented as node voltages with respect to the ground potential using the convention of positive logic a logic "1" is represented by high voltage of V_{DD} and a logic "0" is represented by a low voltage of "0". The output node is loaded with a capacitance C_L, where C_L represents the total capacitance seen at the output node due to the parasitic device capacitance and the interconnect capacitances.

The model of a such a combinational logic circuit is shown in Fig. 2.1.1.

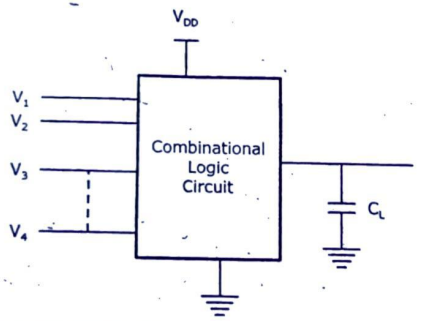


Fig. 2.1.1 Model of a Combinational Logic Circuit

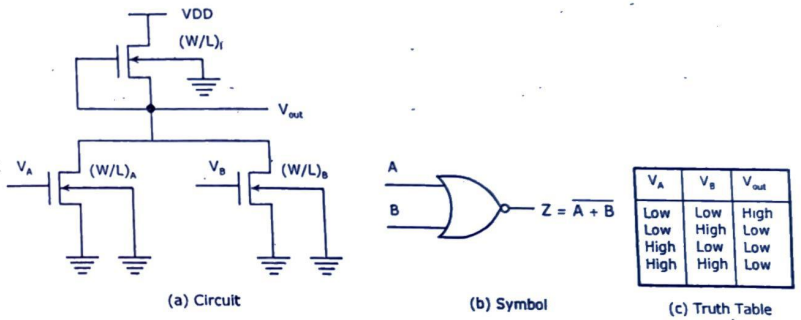
Critical voltage points on Voltage Transfer Characteristic (VTC), such V_{OL} or V_{th} are considered to be the important design parameters for combinational logic circuits. Other design parameters are the silicon area, static and dynamic power dissipation.

2.2 MOS LOGIC CIRCUITS WITH DEPLETION NMOS LOADS

Here we discuss about various CMOS combinational logic designs. We will discuss with a brief treatment of MOS logic circuit with depletion NMOS load. The NMOS depletion load circuits widely used in the many areas in digital circuit design we will examine simple circuit configurations such as to input NAND and NOR gates.

2.2.1 Two-Input NOR Gate (NOR 2 Gate)

The circuit diagram, logic symbol and the corresponding truth table of the two-input depletion-load NOR gate is shown in Fig. 2.2.1.



V_A	V_B	V_{out}
Low	Low	High
Low	High	Low
High	Low	Low
High	High	Low

Fig. 2.2.1 Two Input NOR Gate with Depletion Load

The Boolean OR operation is performed by the parallel connection of the two enhancement type NMOS, which act as driver in NOR circuit. If the logic high is given to the input of transistor A(V_A) or transistor B(V_B) then the corresponding transistor will turn on and a path between output and ground will be established. Consequently, the output voltage will become low. In this case, the circuit operates like a depletion load inverter with respect to its static behaviour. A similar result is achieved when both V_A and V_B are high in which case two parallel conducting path are created between the output terminal and the ground.

If, on the other hand, both V_A and V_B are low, both driver transistors are in cut-off. In this case, the output terminal voltage is pulled to a logic high level by the depletion type NMOS load transistor.

Note that all the substrate of three transistors are connected to ground so, in load transistor body effect will take place. We can simplify the DC analysis significantly by assuming structural similarities between the circuit and the simple NMOS depletion load inverter.

Calculation of V_{OH} : When both input voltages V_A and V_B are lower than the corresponding driver threshold voltage, the both drivers are turned off and no drain current will conduct. Hence the load transistor, which operates in linear region, should have a zero current.

i.e., $$I_{D,(l)} = \frac{K_{n,(l)}}{2}\left[2\,|\,V_{T,(l)}\,(V_{OH})\,|.(V_{DD} - V_{OH}) - (V_{DD} - V_{OH})^2\right] = 0 \quad \ldots (2.2.1)$$

Which gives, $V_{OH} = V_{DD}$

Calculation of V_{OL} : For calculating V_{OL} we have to consider three cases because the output is low when three different combinations input voltages are applied to the input of driver transistors. These combinations are,

(1) VA = V_{OH} VB = V_{OL}

(2) VA = V_{OL} VB = V_{OH}

(3) VA = V_{OH} VB = V_{OH}

For first two cases (1) and (2), the NOR circuit reduces to a NMOS depletion-load inverter. Considering that the threshold voltage of the two drivers are identical i.e., $V_{To,(A)} = V_{To,(B)} = V_{To}$, the driver-to-load ratio of the corresponding inverter can be found as,

CASE 1

$$K_R = \frac{K_{(d),A}}{K_{(l)}} = \frac{K'_{n,(d)}\left(\frac{W}{L}\right)_A}{K'_{n,(l)}\left(\frac{W}{L}\right)_l} \quad \text{... (2.2.2)}$$

CASE 2

$$K_R = \frac{K_{(d),B}}{K_{(l)}} = \frac{K'_{n,(d)}\left(\frac{W}{L}\right)_B}{K'_{n,(l)}\left(\frac{W}{L}\right)_l} \quad \text{... (2.2.3)}$$

The output low voltage V_{OL} in both cases can be found as,

$$V_{OL} = V_{OH} - V_{TO} - \sqrt{(V_{OH} - V_{TO})^2 - \left(\frac{K_{(l)}}{K_{(d)}}\right)|V_{T,(l)}(V_{OL})|^2} \quad \text{... (2.2.4)}$$

As the $\left(\frac{W}{L}\right)$ ratio for both the drivers are identical i.e., $\left(\frac{W}{L}\right)_A = \left(\frac{W}{L}\right)_B$ so the output low voltage (V_{OL}) will be same for both cases (1) and (2).

In case (3), when both the driver transistors are turned on, the saturated load current is a sum of two linear mode currents.

i.e., $$I_{D,(l)} = I_{D,(d)A} + I_{D,(d)B} \quad \text{... (2.2.5)}$$

$$\frac{K_{(l)}}{2}\left[V_{T,(l)}(V_{OL})\right]^2 = \frac{K_{(d),A}}{2}\left[2(V_A - V_{TO})V_{OL} - V_{OL}^2\right] + \frac{K(d),B}{2}\left[2(V_B - V_{TO})V_{OL} - V_{OL}^2\right] \quad \text{... (2.2.6)}$$

As $V_A = V_B = V_{OH}$, so we can derive an equivalent driver-to-load ratio for the NOR circuit as,

$$K_R = \frac{K_{(d),A} + K_{d,(B)}}{K_{(l)}} = \frac{K'_{n,(d)}\left[\left(\frac{W}{L}\right)_A + \left(\frac{W}{L}\right)_B\right]}{K'_{n,(l)}\left(\frac{W}{L}\right)_l} \quad \text{... (2.2.7)}$$

Thus we can replace two drivers with a single driver circuit with driver-to-load ratio as in Eq. (2.2.7).

Now the output low voltage is,

$$V_{OL} = V_{OH} - V_{TO} - \sqrt{(V_{OH} - V_{TO})^2 - \left(\frac{K_{(l)}}{K_{(d),A} + K_{(d),B}}\right)|V_{T,(l)}(V_{OL})|^2} \quad \text{... (2.2.8)}$$

Note that V_{OL} given by Eq. (2.2.8) is lower than that calculated by Eq. (2.2.4). So, we can conclude that the worst case condition from the static operation point of view i.e., highest possible V_{OL} value, is in cases (1) and (2).

This result also gives an idea for a simple design strategy for NOR gate. Usually, when one input is high we get a worst case for V_{OL} i.e., higher value. Hence we assume that only one input (either V_A or V_B) is logic high and determine the driver-to-load ratio using Eq. (2.2.7).

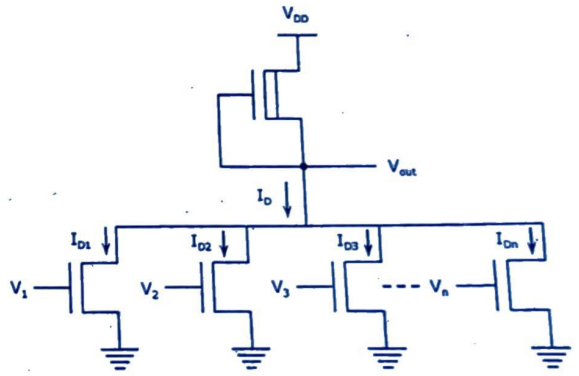
Then, $$K_{(d),A} = K_{(d),B} = K_R K_{(l)} \quad \text{... (2.2.9)}$$

This yields two identical driver transistors, which guarantee the required value of V_{OL} in the worst case. When both inputs are logic high, the output voltage is even lower than the required maximum. V_{OL}, thus the design constraint is satisfied.

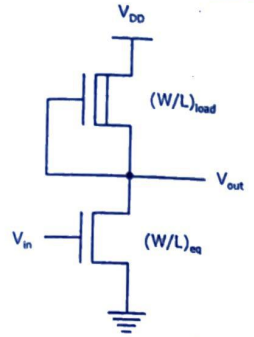
2.2.1.1 Generalized NOR Structure with Multiple Inputs

We can expand the explanation of two input NOR gate into multiple input NOR gate to generate n-input NOR gates, which consists of a n-parallel driver transistors and as well as connected with load transistor in series.

An n-input NOR with NMOS depletion load logic and equivalent circuit are shown in Fig. 2.2.2.



(a)



(b)

Fig. 2.2.2 Generalized n-input NOR Structure and its Equivalent Inverter Circuit

In Fig. 2.2.2, the combination current I_D is supplied by the driver transistors which are turned on.

The combined pull-down current is expressed as,

$$I_D = \sum_{k(on)} I_{D,k} = \begin{cases} \sum_{k(on)} \frac{\mu_n C_{ox}}{2} \left(\frac{W}{L}\right)_k \left[2(V_{GS,k} - V_{T0})V_{out} - V_{out}^2\right] \text{linear} \\ \sum_{k(on)} \frac{\mu_n C_{ox}}{2} \left(\frac{W}{L}\right)_k (V_{GS,k} - V_{T0})^2 \text{ saturation} \end{cases} \quad \text{... (2.2.10)}$$

Assuming that the input voltages of all driver transistors are identical.

i.e., $V_{GS,k} = V_{GS}$ for k = 1, 2, ..., n ... (2.2.11)

Thus, the pull-down current expression can be rewritten as,

$$I_D = \begin{cases} \frac{\mu_n C_{ox}}{2} \left(\sum_{k(on)} \left(\frac{W}{L}\right)_k\right) [2(V_{GS} - V_{T0})V_{out} - V_{out}^2] \text{ linear} \\ \frac{\mu_n C_{ox}}{2} \left(\sum_{k(on)} \left(\frac{W}{L}\right)_k\right) (V_{GS} - V_{T0})^2 \text{ saturation} \end{cases} \quad \text{... (2.2.12)}$$

For static analysis, we can reduce the equivalent circuit of n-input NOR gate as shown in Fig. 2.2.2(b).

The (W/L) ratio of the driven transistor is,

$$\left(\frac{W}{L}\right)_{eq} = \sum_{k(on)} \left(\frac{W}{L}\right)_k \quad \text{... (2.2.13)}$$

Keep in mind that the source terminals of all enhancement type NMOS driver transistors are connected to ground and the drivers do not experience any substrate-bias effect. However, the depletion-type NMOS load transistor is subjected to substrate-bias effect, as the source to substrate voltage $V_{SB} - V_{out}$.

2.2.1.2 Transient Analysis of NOR Gate

The two-input NOR gate which all the relevant parasitic device capacitance is shown in Fig. 2.2.3.

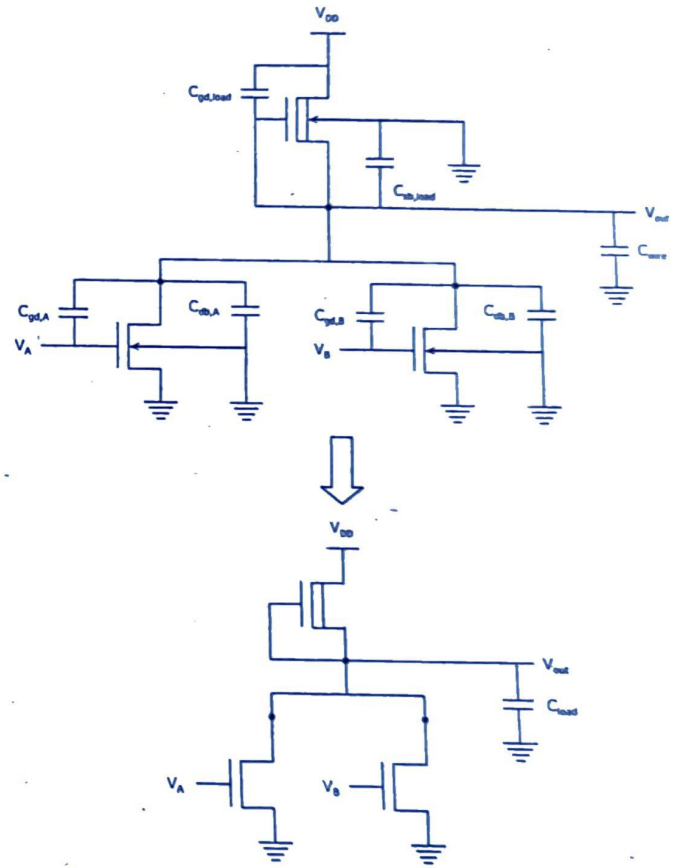


Fig. 2.2.3 Parasitic Device Capacitances in the NOR2 Gate and the Lumped Equivalent Load Capacitance

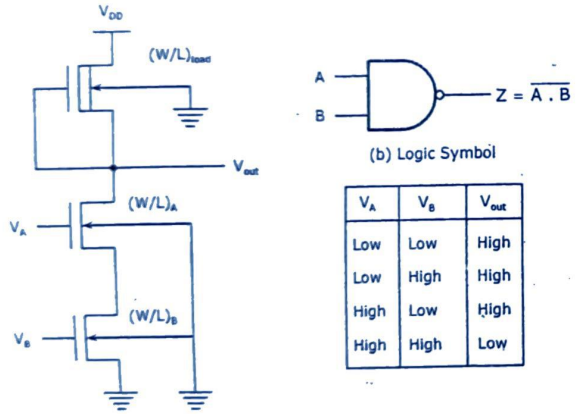
In Fig. 2.2.3, all the parasitic capacitances are combined into one lumped capacitance, connected between the output node and the ground. The value of the combined load capacitance C_{load} is,

$$C_{load} = C_{gd,A} + C_{gd,B} + C_{gd,load} + C_{db,A} + C_{db,B} + C_{sb,load} + C_{wire} \quad \ldots (2.2.14)$$

The load capacitance at the output node of the equivalent inverter corresponding to a NOR gate is always larger than the total lumped load capacitance of the actual inverter with same dimensions. Thus, the actual transient response of the NOR gate will be slower than that of the equivalent inverter.

2.2.2 Two-Input NAND Gate

The circuit diagram, logic symbol and the corresponding truth table of the two-input depletion load NAND gate is shown in Fig. 2.2.4.



(a) CMOS NAND2 Gate

(b) Logic Symbol

V_A	V_B	V_{out}
Low	Low	High
Low	High	High
High	Low	High
High	High	Low

(c) Truth Table

Fig. 2.2.4 A Two-input Depletion-load NAND Gate, its Logic Symbol and the Corresponding Truth Table

The Boolean AND operation is performed by the series connection of two enhancement type NMOS transistors. If both the input V_A and V_B are logic high then only, there is a conducting path between output terminal and ground. In this case the output will be low, which is the complement result of AND operation.

Otherwise either both or one driver transistor will be off and the output voltage will be pulled to a logic high level by the depletion type NMOS load transistor.

From Fig. 2.2.4 it is clear that except driver transistor B, all other are subjected to substrate bias effect as $V_{SB} \neq 0$ in both transistors except transistors B. It can be found that for all three combination when output voltage is logic high,

$$V_{OH} = V_{DD}$$

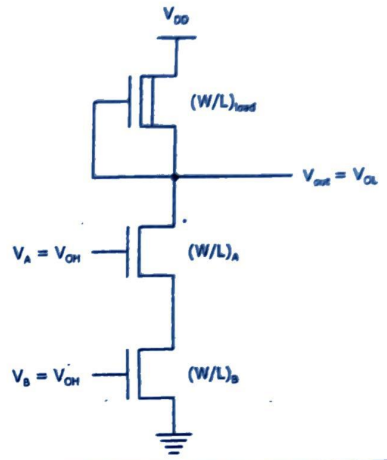


Fig. 2.2.5 The NAND2 Gate with Both of its Inputs at a Logic-high Level

For calculation of V_{OL} consider Fig. 2.2.5 when both inputs are equal to V_{OH}. It is seen that the drain currents of the all the transistors are equal.

$$I_{D,(l)} = I_{D,(A)} = I_{D,(B)} \quad \ldots (2.2.15)$$

$$\frac{K_{(l)}}{2}|V_{T,(l)}(V_{OL})|^2 = \frac{K_{d,(A)}}{2}\left[2(V_{GS,(A)} - V_{T,(A)})V_{DS,(A)} - V_{DS,(A)}^2\right]$$

$$= \frac{K_{d,(B)}}{2}\left[2(V_{GS,(B)} - V_{T,(B)})V_{DS,(B)} - V_{DS,(B)}^2\right] \quad \ldots (2.2.16)$$

Neglecting substrate bias effect for transistor A, for simplicity.

i.e., $V_{T(A)} = V_{T(B)} = V_{TO}$

Since the sourced substrate voltage of transistor A is relatively low. The drain to source voltage of both the driver transistor can be expressed as, using Eq. (2.2.16),

$$V_{DS,(A)} = V_{OH} - V_{TO} - \sqrt{(V_{OH} - V_{TO})^2 - \left(\frac{K_{(l)}}{K_{d,(A)}}\right)|V_{T,(l)}(V_{OL})|^2} \quad \ldots (2.2.17)$$

$$V_{DS,(B)} = V_{OH} - V_{TO} - \sqrt{(V_{OH} - V_{TO})^2 - \left(\frac{K_{(l)}}{K_{d,(B)}}\right)|V_{T,(l)}(V_{OL})|^2} \quad \ldots (2.2.18)$$

Note that V_{OL} is equal to the sum of the drain to source voltage of both drivers. Assume the two drivers are identical.

i.e., $K_{d,(A)} = K_{d,(B)} = K_{(d)}$

Thus,
$$V_{OL} \approx 2\left[V_{OH} - V_{T0} - \sqrt{(V_{OH} - V_{T0})^2 - \left(\frac{K_{(l)}}{K_{(d)}}\right) | V_{T,(l)}(V_{OL}) |^2}\right] \quad \ldots (2.2.19)$$

Now for two identical drivers connected in series, the drain currents can be written as,

$$I_{D,(A)} = \frac{K_{(d)}}{2}\left[2(V_{GS,(A)} - V_{T0})V_{DS,(A)} - V^2_{DS(A)}\right] \quad \ldots (2.2.20)$$

$$I_{D,(B)} = \frac{K_{(d)}}{2}\left[2(V_{GS,(B)} - V_{T0})V_{DS,(B)} - V^2_{DS(B)}\right] \quad \ldots (2.2.21)$$

As both currents are equal, then,

$$I_0 = I_{D,(A)} = I_{D,(B)} = \frac{I_{D,(A)} + I_{D,(B)}}{2} \quad \ldots (2.2.22)$$

Using $V_{GS,(A)} = V_{GS,(B)} - V_{DS,(B)}$, Eq. (2.2.22) yields.

$$I_D = \frac{K(d)}{4}\left[2(V_{GS,(B)} - V_{T0})(V_{DS,(A)} + V_{DS,(B)}) - (V_{DS,(A)} + V_{DS,(B)})^2\right] \quad \ldots (2.2.23)$$

Now let $V_{GS} = V_{GS,(B)}$ and $V_{DS} = V_{DS,(A)} + V_{DS,(B)}$ it implies that,

$$I_D = \frac{K_{(d)}}{4}\left[2(V_{GS} - V_{T0})V_{DS} - V^2_{DS}\right] \quad \ldots (2.2.24)$$

Thus from Eq. (2.2.24), we can conclude that two NMOS transistor connected in series and with the same voltage be have like one NMOS transistor with $K_{eq} = 0.5\ K_{(d)}$.

2.2.2.1 Generalized NAND Structure with Multiple Inputs

An n-input NAND with NMOS depletion load logic and it's equivalent inverter circuit are shown in Fig. 2.2.6.

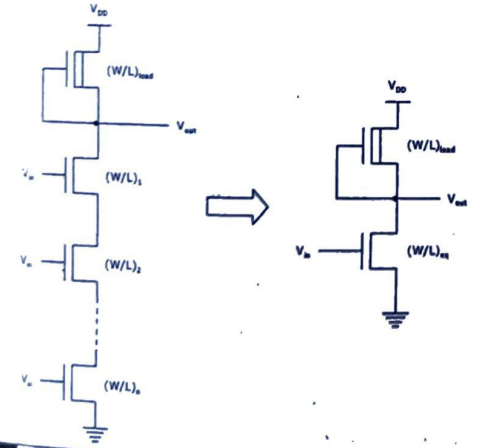


Fig. 2.2.6 The Generalized NAND Structure and its Inverter Equivalent

In Fig. 2.2.6, by neglecting the substrate bias effect and assuming that the threshold voltages of all transistors are equal to V_{T0}, the driver drain current T_0 for both linear and saturation can be expressed as,

$$I_D = \frac{\mu_n C_{ox}}{2}\left(\frac{1}{\sum_{k(on)}\frac{1}{\left(\frac{W}{L}\right)_k}}\right)\begin{cases}[2(V_{in} - V_{T0})V_{out} - V^2_{out}] \text{ linear} \\ (V_{in} - V_{T0})^2 \text{ saturation}\end{cases} \quad .. (2.2.25)$$

The multiple-input NAND gate can also be reduced to an equivalent inverter as shown in Fig. 2.2.6, the (W/L) ratio of the driver transistor is,

$$\left(\frac{W}{L}\right)_{eq} = \frac{1}{\sum_{k(on)}\frac{1}{\left(\frac{W}{L}\right)_k}} = \frac{1}{n}\left(\frac{W}{L}\right) \quad \ldots (2.2.26)$$

Thus, the series structure consisting of n driver transistors has an equivalent (W/L) ratio of $(W/L)_d$ when all inputs are logic high.

2.2.2.2 Transient Analysis of NAND Gate

The two-input NAND gate with all the relevant parasitic device capacitances is shown in Fig. 2.2.7.

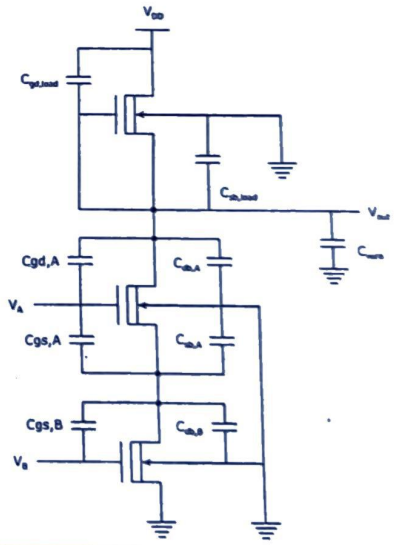


Fig. 2.2.7 Parasitic Device Capacitances in the NAND2 Gate

When $V_A = V_{OH}$ and V_B switches from V_{OH} to V_{OL}, then the combined load capacitance C_{load} is,

$$C_{load} = C_{gd,L} + C_{dg,A} + C_{gd,B} + C_{gs,A} + C_{db,A} + C_{db,B} + C_{sb,A} + C_{sb,L} + C_{wire} \quad \ldots (2.2.27)$$

Now, consider that $V_B = V_{OH}$ and V_A switches from V_{OH} to V_{OL}, then V_{out} will rise but the intend node voltage will remain low because bottom transistor is ON. As the lumped capacitance is a function of applied voltage so,

$$C_{load} = C_{gd,I} + C_{gd,A} + C_{gb,A} + C_{sb,I} + C_{wire} \quad \ldots (2.2.28)$$

Thus, it is expected that the high-to-low switching delay from signal V_B connected to the bottom transistor is larger than the high-to-low switching delay from signal V_A connected to the top transistor.

2.3 CMOS LOGIC GATES

The design analysis of CMOS logic gates can be based on the basic principles developed for the NMOS depletion-load logic circuits. To implement the CMOS logic gates using the PMOS and NMOS circuits.

Here we explain the NOR gate and NAND gate implementations using the CMOS logic gates.

2.3.1 CMOS NOR2 (Two-Input-NOR) Gate

The CMOS NOR gate and its complementary operations is shown in the Fig. 2.3.1.

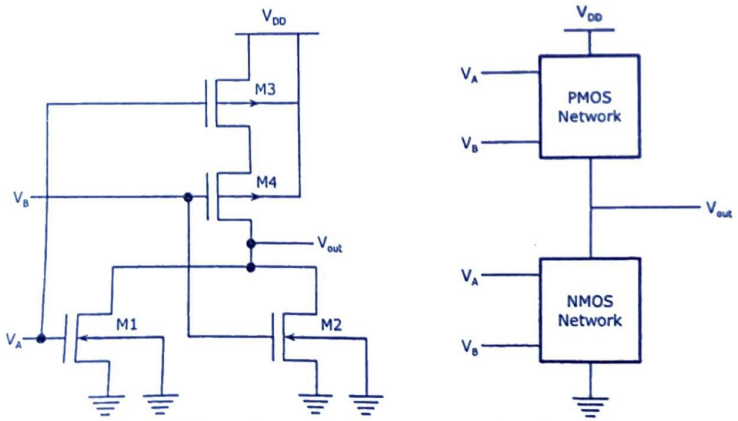


Fig. 2.3.1 A CMOS NOR2 Gate and its Complementary Operation

The circuit Fig. 2.3.1 have contain a parallel connected n-net and series. Connected complimentary p-net. The input voltages V_A and V_B applied to the gates of NMOS and PMOS transistor.

It shows the complimentary operation, when either one or both input high the p-net is cut-off. If both the input voltages are low the p-net creates a conducting path between the output node and supply voltage V_{DD}.

Thus, dual circuit structure allows that, for any given input combination, the output is connected to either to V_{DD} or to ground.

The equivalent circuit for the CMOS NOR2 gate is shown in the Fig. 2.3.2.

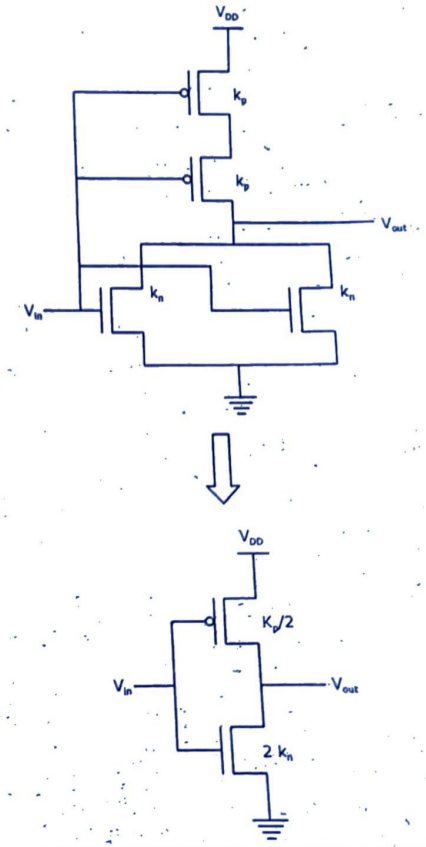


Fig. 2.3.2 A CMOS NOR2 Gate and its Inverter Equivalent

The switching threshold voltage is find out by using NOR2 equivalent-inverter approach.

The Fig. 2.3.3 shows the CMOS NOR2 gate with the parasitic device capacitance.

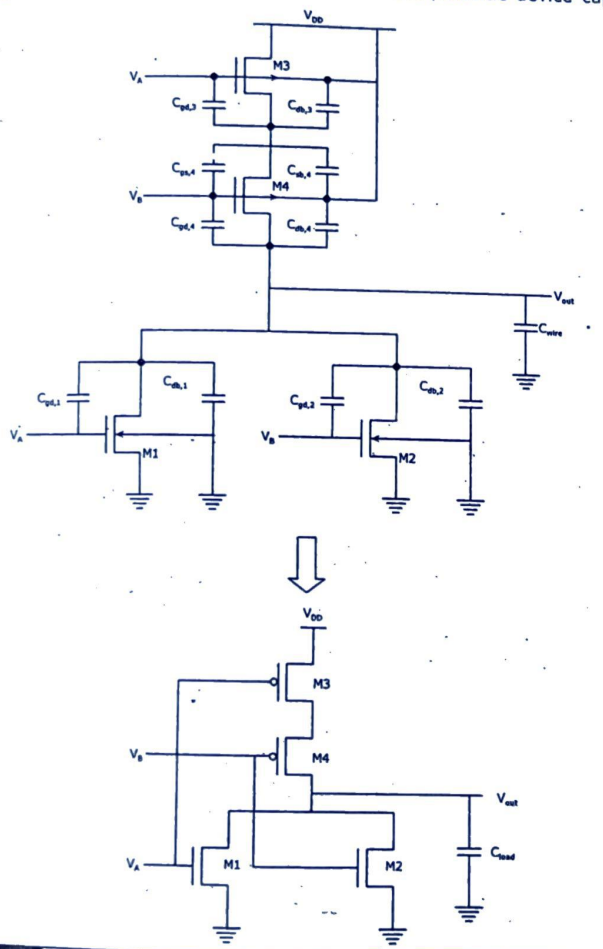


Fig. 2.3.3 Parasitic Device Capacitances of the CMOS NOR2 Circuit and the Simplified Equivalent with the Lumped Output Load Capacitance

The inverter equivalent and the corresponding lumped output load capacitance. In the worst case, the total lumped capacitance is assumed to be equal to the sum of all internal parasitic device capacitance Fig. 2.3.3.

2.3.2 CMOS NAND2 (Two-Input-NAND) Gate

The Fig. 2.3.5 shows the two-input CMOS NAND gate.

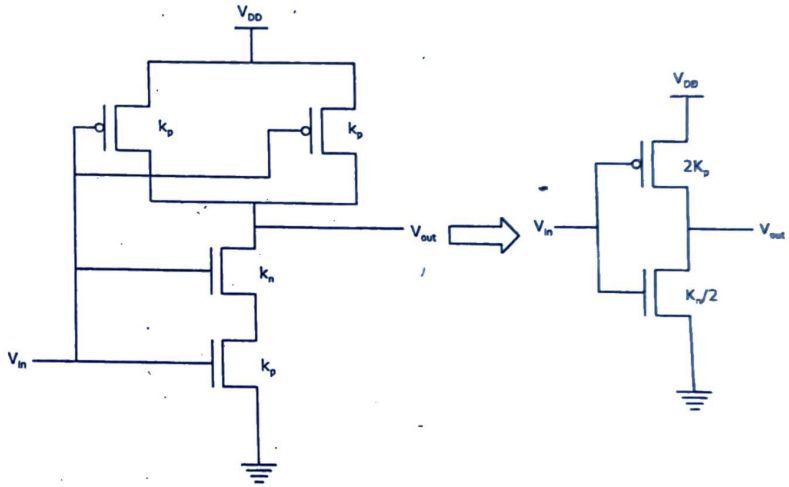


Fig. 2.3.4 A CMOS NOR2 Gate and its Inverter Equivalent

The operation is exact dual to the CMOS NOR2 gate. The n-net consisting of a two series connected NMOS transistors creates a conducting path between the output node and the ground only if both inputs are logic high.

In this case both of the parallel connected PMOS transistors in the p-net will be Off. For all other input combinations, either one or both of the PMOS transistors will be turned on when n-net cut-off, thus creating a current path between the output node and the power supply voltage.

Here we can observe the area requirements of CMOS combinational logic gates. In comparison equivalent NMOS depletion load logic, the total number transistors in CMOS gates is about twice the number of transistors in NMOS gates.

The silicon area of CMOS is not need like twice of NMOS circuit. Thus, the area disadvantage of CMOS logic may actually be smaller than the simple transistor count suggests.

2.3.3 Layout of Simple CMOS Logic Gate

Fig. 2.3.5 shows a simple layout of CMOS NOR2 gate, using the single layer metal and single layer polysilicon.

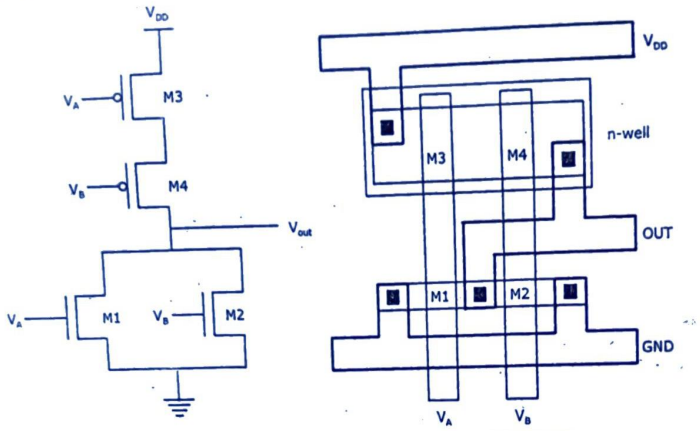


Fig. 2.3.5 Sample Layout of the CMOS NOR2 Gate

The p-type diffusion area for PMOS transistors and the n-type diffusion area for NMOS transistors are aligned to allow simple routing of the gate signals via the parallel polysilicon lines running vertically.

Fig. 2.3.6 shows a simple layer of CMOS NAND2 gate, using the single layer metal and single layer polysilicon.

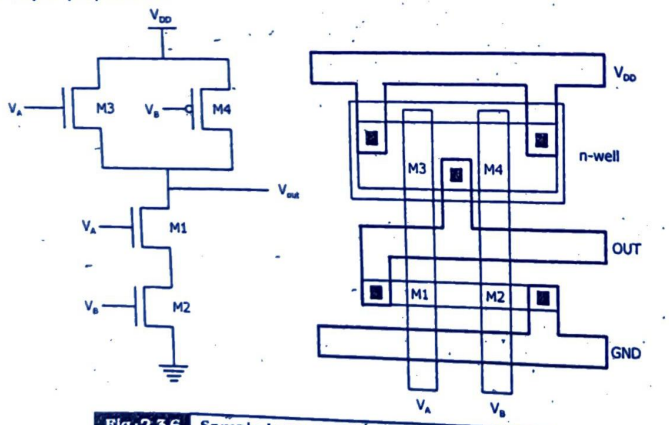


Fig. 2.3.6 Sample Layout of the CMOS NAND2 Gate

2.4 COMPLEX LOGIC CIRCUITS

The basic circuit structures and design principles developed for NOR and NAND in CMOS logic design can be easily extended to complex logic gate, to realize other complex Boolean functions of multiple input variables. One of the most attractive feature of NMOS and CMOS logic circuits is the ability to realize complex function using a small number of transistors.

Let us consider the following Boolean function as an example.

$$Z = \overline{A(D + E) + BC} \qquad \ldots (2.4.1)$$

The Eq. (2.4.1) can be realized as an NMOS circuit as shown in Fig. 2.4.1.

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

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

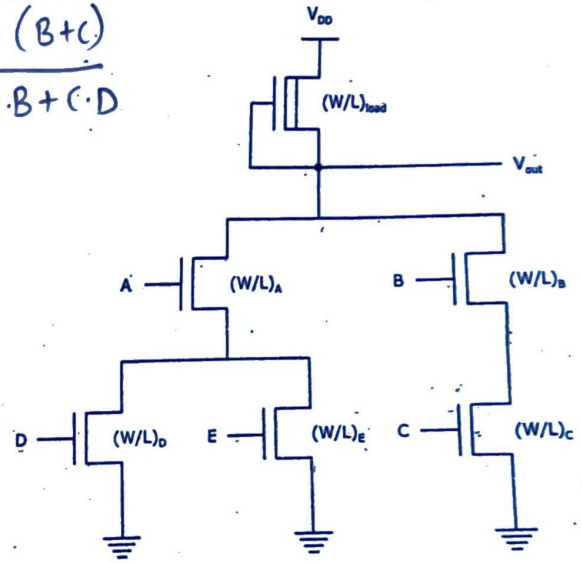


Fig. 2.4.1 NMOS Complex Logic Gate Realizing the Boolean Function given in Eq. (2.4.1)

In Fig. 2.4.1, the left hand side NMOS driver consisting of three driver transistors is used to perform the logic function A(D + E) and the right hand side branch is used to perform the function BC. By connecting these two branches in parallel and by placing a load transistor between output terminal and power supply (V_{DD}), we can obtain the complex function given in Eq. (2.4.1) easily. Note that here each input variable is assigned to only one driver.

If all the inputs are high, then the equivalent driver (W/L) ratio of the pull-down network is expressed as,

$$\left(\frac{W}{L}\right)_{eq} = \frac{1}{\frac{1}{\left(\frac{W}{L}\right)_B} + \frac{1}{\left(\frac{W}{L}\right)_C}} + \frac{1}{\frac{1}{\left(\frac{W}{L}\right)_A} + \frac{1}{\left(\frac{W}{L}\right)_D + \left(\frac{W}{L}\right)_E}} \quad \ldots (2.4.2)$$

In order to calculate the logic-low voltage V_{OL}, we have to consider the various classes. These may be listed as,

(1) A-D Class 1

(2) A-E Class 1

(3) B-C Class 1

(4) A-D-E Class 2

(5) A-D-B-C Class 3

(6) A-E-B-C Class 3

(7) A-D-E-B-C Class 4

Now assuming that all the driver transistors have the same (W/L) ratio, in the class 1 path such as A-E has the highest series resistance followed by class2, class3 and class4. So,

$$V_{OL1} > V_{OL2} > V_{OL3} > V_{OL4} \quad \ldots (2.4.3)$$

The design strategy yields the following ratios for the three worst case paths,

$$\left(\frac{W}{L}\right)_A = \left(\frac{W}{L}\right)_D = 2\left(\frac{W}{L}\right)_d$$

$$\left(\frac{W}{L}\right)_A = \left(\frac{W}{L}\right)_E = 2\left(\frac{W}{L}\right)_d$$

$$\left(\frac{W}{L}\right)_B = \left(\frac{W}{L}\right)_C = 2\left(\frac{W}{L}\right)_d$$

2.4.1 Complex CMOS Logic Gates

The combination of PMOS-network and NMOS network is CMOS logic gate here we realize the complex CMOS logic gate. This logic gate is realized with the help of complex PMOS and NMOS logic gates.

The CMOS logic gate realization of the Boolean function Z in Eq. (2.4.1) is shown in Fig. 2.4.2.

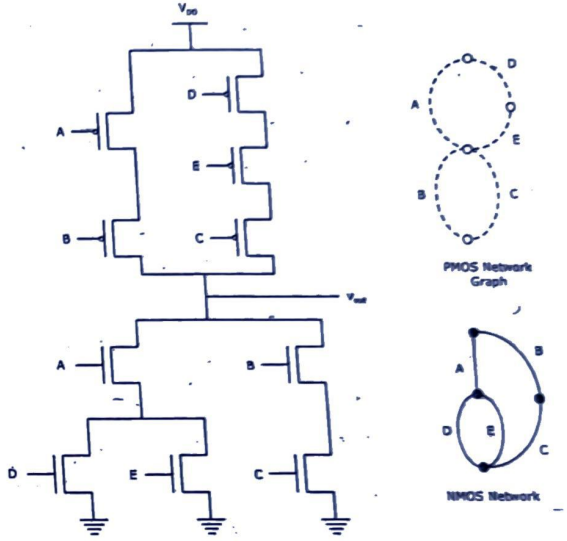


Fig. 2.4.2 A Complex CMOS Logic Gate Realizing the Boolean Function Eq. (2.3.1)

From the Fig. 2.4.2 we observe the NMOS network is connected with PMOS network. The PMOS network, on the other hand must be the dual network of the n-net. This means that the parallel connections in the NMOS network will correspond to a series connection in the PMOS network and all the series connections in the pull-down network correspond to a parallel connection in the n-net.

2.4.1.1 Layout Technique using Euler Graph Method

Euler graph method is used to determine if any complex CMOS gate can be physically laid out in an optimum fashion. The following steps explains the construction of minimum-area layout for the complex CMOS logic gate,

Step 1 : Start with either NMOS or PMOS tree (NMOS for this example) and connect lines for transistor segments, labeling devices with vertex points as circuit nodes.

Step 2 : Place a new vertex within each confined area on the pull-down graph and connect neighbouring vertices with new lines, making sure to cross each edge of the pull-down tree only once.

Step 3 : The new graph represents the pull-up tree and is the dual of the pull-down tree.

The stick diagram shown in Fig. 2.4.3 (done with arbitrary gate ordering) gives a very non-optimum layout for the CMOS gate shown in Fig. 2.4.2.

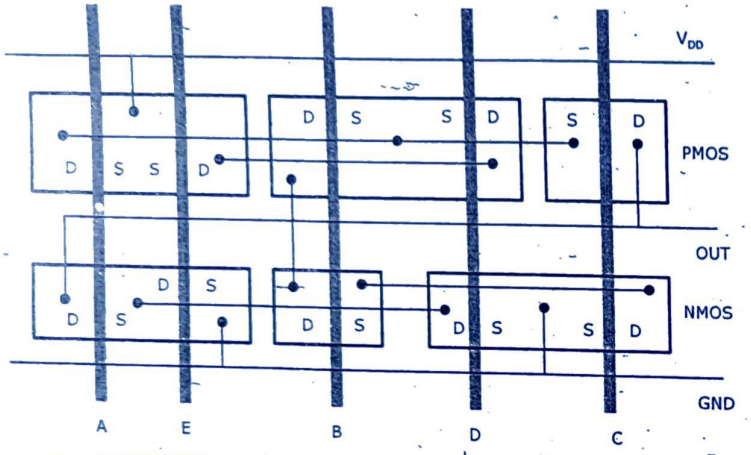


Fig. 2.4.3 Stick-Diagram Layout of the Complex CMOS Logic Gate, with an Arbitrary Ordering of the Polysilicon Gate Columns

By using the Euler path approach to re-order the polysilicon lines of the Fig. 2.4.3, we can obtain an optimum layout. Find Euler path in both the pull-down tree graph and the pull-up tree graph with identical ordering of the inputs. Euler path traverses each branch of the graph exactly one. Fig. 2.4.4 shows the construction of a common Euler path for both graphs in Fig. 2.4.2.

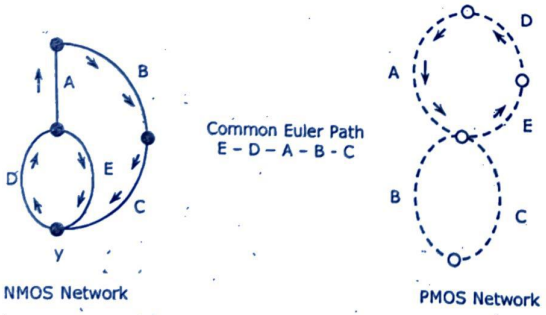


Fig. 2.4.4 Finding a Common Euler Path in Both Graphs for n-Net and p-Net Provides a Gate Ordering that Minimizes the Number of Diffusion Breaks

In Fig. 2.4.4, by reordering the input gates as E-D-A-B-C, we can obtain an optimum layout of the given CMOS gate with single actives for both the NMOS and PMOS devices. The stick diagram is shown in Fig. 2.4.5.

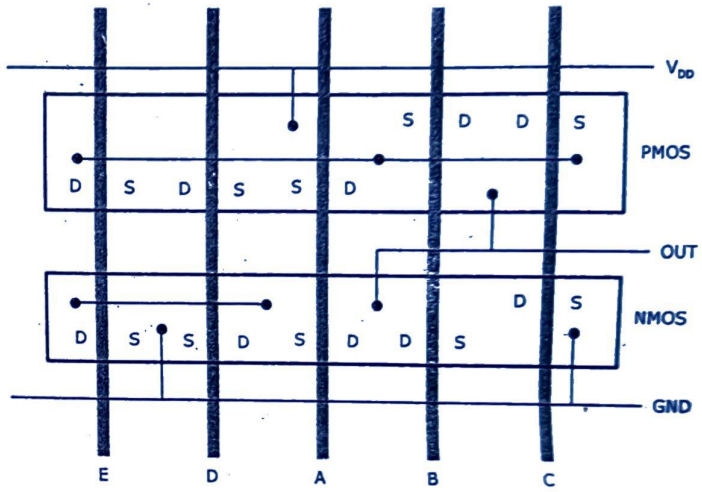


Fig. 2.4.5 Optimized Stick-diagram Layout of the Complex CMOS Logic Gate

2.4.2 AOI and OAI Gates

Generally complex CMOS networks, both the pull down and pull up networks have a less limitations. The complex CMOS logic is basically two categories subsets. They are,

(1) AND-OR-Inverter (AOI).

(2) OR-AND-Inverter (OAI).

2.4.2.1 AND-OR-INVERT (AOI) CMOS Gates

AOI complex CMOS gate can be used to directly implement a Sum-Of-Products (SOP) form of Boolean function. The pull-down N-tree can be implemented as,

(1) Product terms yield series-connected NMOS transistors.

(2) Sums are denoted by parallel-connected legs.

(3) The complete function must be an inverted representation.

NOTE : *The pull-up p-tree is derived as the dual of the N-tree.*

Example of CMOS AOI gate is shown in Fig. 2.4.6.

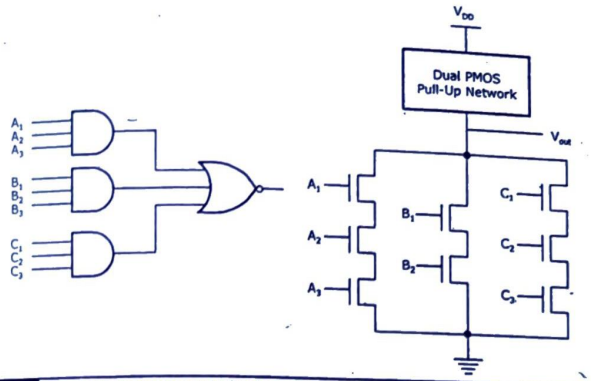


Fig. 2.4.6 An AND-OR-INVERT (AOI) Gate and the Corresponding Pull-down Net

2.4.2.2 OR-AND-INVERT (OAI) CMOS Gates

An OAI CMOS gate is similar to the AOI gate except that it is an implementation of Product-Of-Sums (POS) form of boolean function realization. The N-tree is implemented as,

(1) Each product term is a set of parallel transistors for each input in the term.

(2) All product terms (parallel groups) are put in series.

(3) The complete function is again assumed to be an inverted representation.

NOTE : ***The p-tree can be implemented as the dual of the N-tree.***

Example of CMOS OAI gate is shown in Fig. 2.4.7.

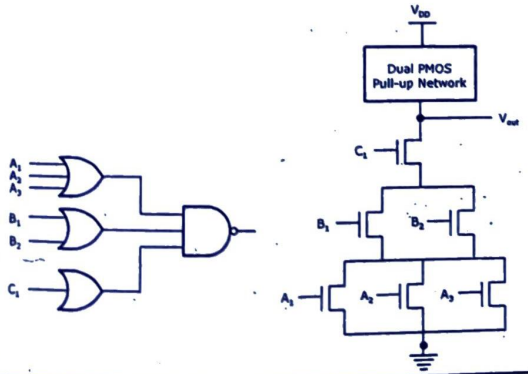


Fig. 2.4.7 An OR-AND-INVERT (OAI) Gate and the Corresponding Pull-down Net

2.4.2.3 Pseudo-NMOS Gates

The design of complex CMOS gates is complicated in AOI and OAI implementations. To reduce complexity we use the pseudo-NMOS gate for implementation of complex CMOS logic gates.

Two complementary transistor, one NMOS and one PMOS, are needed for every input of complex CMOS circuit. One possible way to reduces the number of transistors is to use the single PMOS transistor, with its gate terminal connected to ground, as the load device as shown in Fig. 2.4.8.

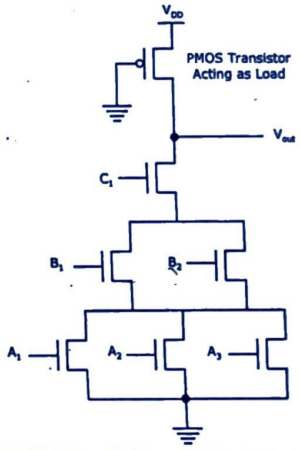


Fig. 2.4.8 The Pseudo-NMOS Implementation of the OAI Gate in Fig. 2.4.7

With this simple arrangement, the complex CMOS gate can be implemented with the less number of transistors.

The similarities of pseudo-NMOS gate to depletion-load NMOS logic gate common. The most important disadvantage of using pseudo-NMOS gate instead of full-CMOS gate is non-zero static power dissipation, the always ON-PMOS load devices conducts a steady-state current when the output voltage lower than V_{DD}.

2.5 CMOS Full-Adder Circuit

The 1-bit full adder circuit is one of the most widely used building blocks in all data processing (arithmetic) and digital signal processing architectures. For 1-bit full adder logic function sum and carry equations are,

$$\text{Sum} = A \oplus B \oplus C = ABC + A\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}\,\overline{C}B$$

$$\text{And carry} = AB + AC + BC$$

We can draw a gate level circuit for both sum and carry using the corresponding Boolean function. Also using the same Boolean expression we can design a transistor- level schematic for both. An inverter is used for both sum and carry as shown in Fig. 2.5.1.

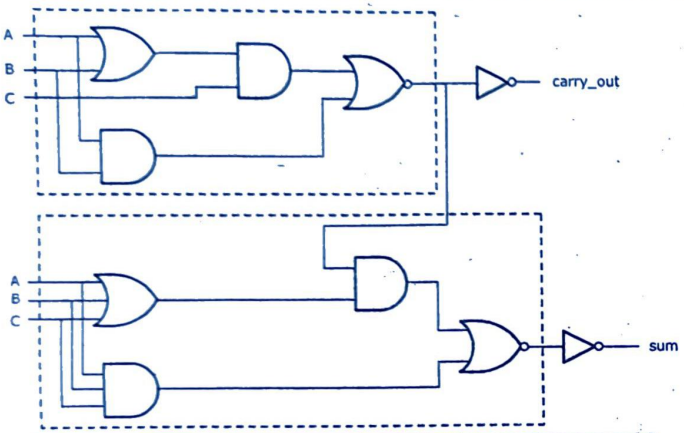


Fig. 2.5.1 Gate-Level Schematic of the One-Bit Full-Adder Circuit

Note that in Fig. 2.5.1 the carry-out internal node is used as an input to the adder complex CMOS gate. The transistor-level schematic of the 1-bit full adder circuit using Fig. 2.5.1 is shown in Fig. 2.5.2.

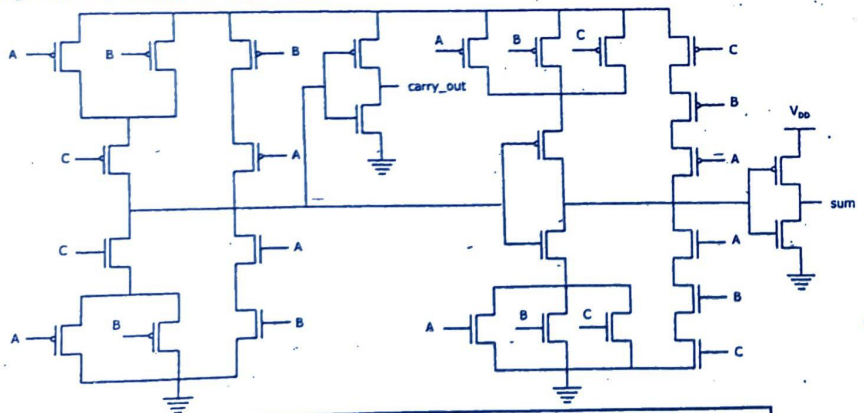


Fig. 2.5.2 Transistor-level Schematic of the One-bit Full-adder Circuit

Note that Fig. 2.5.2 contains a total of 28 transistors among which 14 are PMOS and 14 are NMOS transistors. It also contains two CMOS inverters that are used to generate the outputs.

2.6 CMOS TRANSMISSION GATES (PASS GATES)

A CMOS Transmission Gate (TG) is a simple switch circuit consisting of one NMOS and one PMOS transistors connected in parallel. The gate voltages applied to these two transistors are set to be complementary signals. Fig. 2.6.1 shows a CMOS transmission gate connected between two terminals A and B. The CMOS Transmission Gate (TG) operates as a bidirectional switch connected between terminals A and B and is controlled by control signal C.

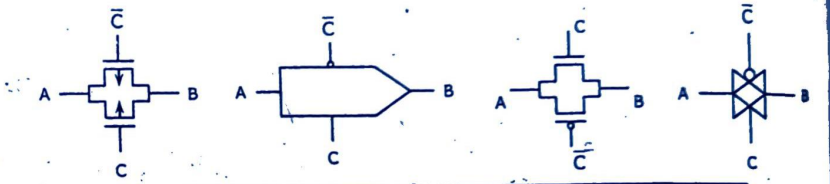


Fig. 2.6.1 Different Representations of CMOS Transmission Gate (TG)

If the control signal C is at logic high, i.e., equal to V_{DD}, then both transistors are turned on and provide a low resistance current path between nodes A and B. If on the other hand, the control signal C is low, then both transistors will be off and the path between the nodes A and B will be an open circuit. This condition is also called a high impedance state.

To perform a detailed DC analysis of CMOS transmission gate, we consider the bias condition shown in Fig. 2.6.2.

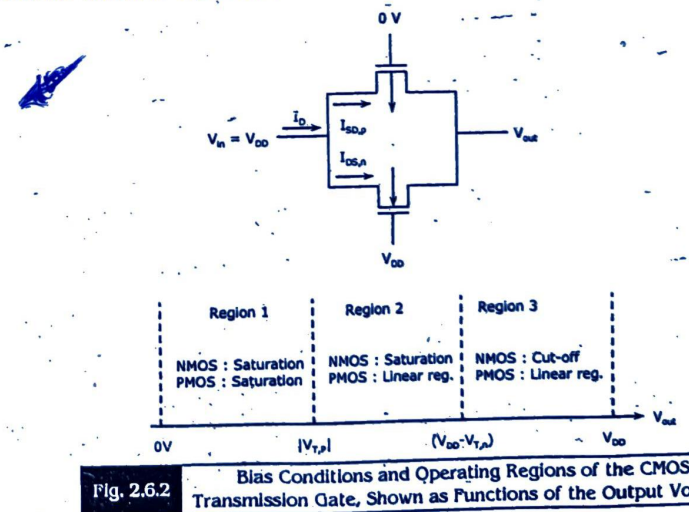


Fig. 2.6.2 Bias Conditions and Operating Regions of the CMOS Transmission Gate, Shown as Functions of the Output Voltage

The input node (A) is connected to a constant logic-high voltage $V_{in} = V_{DD}$. The control signal is also logic-high, thus ensuring both the transistors are turned on.

The drain to source and gate to source voltages of NMOS transistors are,

$$V_{DS,n} = V_{DD} - V_{out}$$

$$V_{GS,n} = V_{DD} - V_{out} \quad \ldots (2.6.1)$$

The NMOS transistor will be turned off for $V_{out} > |V_{T,p}|$ and will operate in the saturation ... (2.6.2)
node for $V_{out} > |V_{T,p}|$. The V_{DS} and V_{GS} voltages of the PMOS transistor are,

$$V_{DS,p} = V_{out} - V_{DD}$$

$$V_{GS,p} = V_{DD} \quad \ldots (2.6.3)$$

$$\ldots (2.6.4)$$

Consequently, the PMOS transistor is in saturation for $V_{out} < V_{T,p}$ and it operates in the linear region for $V_{out} > |V_{T,p}|$. Note that, unlike the NMOS transistor, PMOS transistor remains turned on, regardless of the output voltage level V_{out}.

The total current through the transmission gate is the sum of the NMOS drain current and PMOS drain current.

$$I_D = I_{DS,n} + I_{SD,p} \quad \ldots (2.6.5)$$

The equivalent resistance for NMOS and PMOS transistors are calculated as,

$$R_{eq,n} = \frac{V_{DD} - V_{out}}{I_{DS,n}} \quad \ldots (2.6.6)$$

$$R_{eq,p} = \frac{V_{DD} - V_{out}}{I_{SD,p}} \quad \ldots (2.6.7)$$

The total equivalent resistance of the CMOSTG will then be the parallel equivalent of these two resistances, $R_{eq,n}$ and $R_{eq,p}$. Now, the equivalent values for the three operating regions of the transmission gate will be calculated.

Region 1 : Here, the output voltage V_{out} is smaller than the absolute value of the PMOS transistors threshold voltage, i.e., $V_{out} < |V_{T,p}|$. According to Fig. 2.6.2, both the transistors are in saturation.

We obtain equivalent resistance of both the devices as,

$$R_{eq,n} = \frac{2(V_{DD} - V_{out})}{K_n(V_{DD} - V_{out} - V_{T,n})^2} \quad \ldots (2.6.8)$$

$$R_{eq,p} = \frac{2(V_{DD} - V_{out})}{K_n(V_{DD} - |V_{T,p}|)^2} \quad \ldots (2.6.9)$$

Region 2 : In this region, $|V_{T,p}| < V_{out} < (V_{DD} - V_{out})$. Thus the PMOS transistor now operate in the linear region, while the NMOS transistor continues to operate in saturation.

Now we can write the equivalent resistance as,

$$R_{eq,n} = \frac{2(V_{DD} - V_{out})}{K_n (V_{DD} - V_{out} - V_{T,n})^2} \quad \ldots (2.6.10)$$

$$R_{eq,p} = \frac{2(V_{DD} - V_{out})}{K_p \left[2(V_{DD} - |V_{T,p}|)(V_{DD} - V_{out}) - (V_{DD} - V_{out})^2\right]}$$

$$= \frac{2}{K_p \left[2(V_{DD} - |V_{T,p}|) - (V_{DD} - V_{out})\right]} \quad \ldots (2.6.11)$$

Region 3 : Here the output voltage is $V_{out} > (V_{DD} - V_{T,p})$. Hence, the NMOS transistor will be turned off, which results in on circuit equivalent. The PMOS transistor will continue to operate in the linear region,

$$R_{eq,p} = \frac{2}{K_p \left[2(V_{DD} - |V_{T,p}|) - (V_{DD} - V_{out})\right]} \quad \ldots (2.6.12)$$

By combining the equivalent resistance values obtained for the three operation regions, we can plot the total resistance of CMOS transmission gate as a function of the output voltage V_{out} as shown in Fig. 2.6.3.

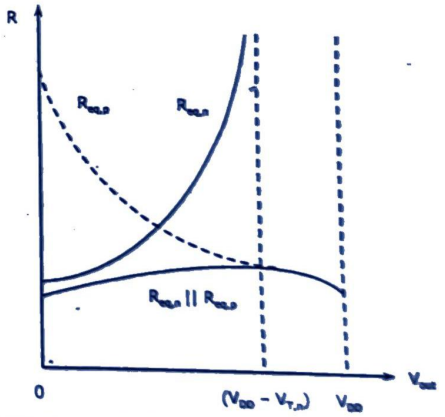


Fig. 2.6.3 Equivalent Resistance of the CMOS Transmission Gate, Plotted as a Function of the Output Voltage

From Fig. 2.6.3, it is evident that the total equivalent resistance of the CMOS TG remains relatively constant, i.e., almost independent of the output voltage, while the equivalent resistance of the NMOS and PMOS transistors are strongly dependent on the output voltage V_{out}. This property of CMOS TG is naturally describe for switching circuit application. A CMOS TG which is turned on by a logic control signal can be replaced by its simple equivalent resistance for dynamic analysis as shown in Fig. 2.6.4.

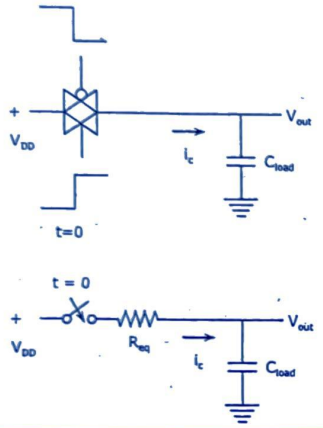


Fig. 2.6.4 Replacing the CMOS TG with its Resistor Equivalent for Transient Analysis

2.6.1 Pass-Transistor Logic

Pass-transistor logic employs CMOS TG that result in compact circuit structures which may even require a small number of transistors than their standard CMOS counterparts. The control signal and its complement must be simultaneously used for TG application.

Fig. 2.6.5 shows a 2-input multiplexer circuit using CMOS TG.

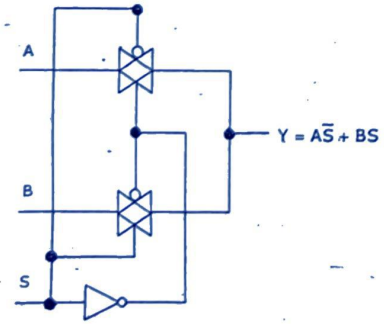


Fig. 2.6.5 CMOS TG Implementation of a 2 × 1 Multiplexer Circuit

The operation of the circuit can be explained as follows. If the control input is low, the upper TG conducts and the output Y is equal to A. On the other hand, if the control input is high, the lower TG conducts and the output Y is equal to B.

Fig. 2.6.6 shows a 2 input XOR gate designed using pass transistor logic.

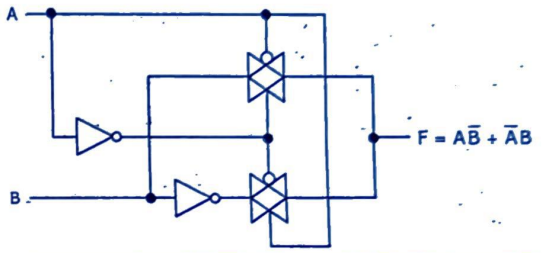


Fig. 2.6.6 Eight-transistor CMOS TG Implementation of the XOR Function

The above design employs 8 transistors. We could also have a more economical 6 transistor implementation of a 2 input XOR gate using pass transistor logic shown in Fig. 2.6.7.

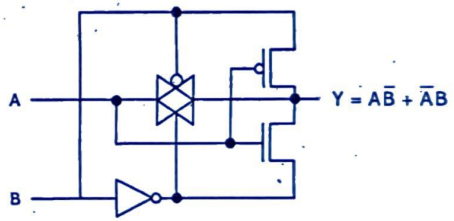


Fig. 2.6.7 Six Transistor Implementation of 2 Input XOR Gate using CMOS TG

2.6.2 Complementary Pass-Transistor Logic

The Complementary Pass-Transistor Logic (CPL) reduces the complexity of the full CMOS pass gate logic circuits. In CPL, only NMOS pass transistor network is used for logic operation. All the inputs are applied in complementary form and the circuit also produces the complementary outputs. Elimination of PMOS transistors reduces the parasitic capacitances associated with each node in the circuit, so higher speed of operation can be achieved than with CMOS TG. To eliminate threshold voltage drops, the voltage of the NMOS transistors must be reduced to 0V through threshold-adjustment implants. This reduces the overall noise immunity and problems like sub-threshold conduction in off mode occurs.

The circuit diagrams of CPL NOR2 and a CPL NAND2 are shown in Fig. 2.6.8.

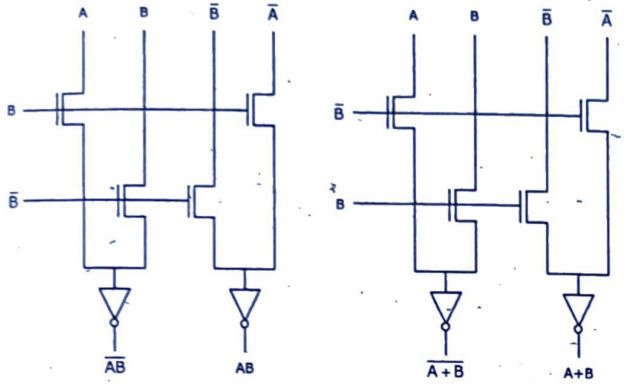


Fig. 2.6.8 Circuit Diagram of CPL

In Fig. 2.6.8, each circuit requires 8 transistors, double than that required using the conventional CMOS realizations. XOR and XNOR functions realized with CPL have a similar complexity as conventional CMOS realizations. The circuit diagram of a CPL-based XOR gate is shown in Fig. 2.6.9.

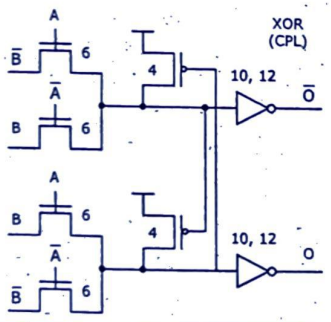


Fig. 2.6.9 Circuit Diagram of a CPL-based XOR Gate

The XOR gate shown in Fig. 2.6.9 contains a PMOS pull-up arrangement configured like a latch. If XOR is true, the upper internal node goes high to $V_{DD} - V_T$ while the lower internal node goes low to GND, thus causing the cross-coupled PMOS load devices to latch and pull the upper internal node all the way to V_{DD}. If XOR is false, the opposite will happens. The inverters provides both true and complement outputs. Cross-coupled PMOS pull-up transistors are used to speed-up the output response.

UNIT-III

SEQUENTIAL MOS LOGIC CIRCUITS

SYLLABUS [M.TECH - H&K]

Behavior of Bistable Elements, SR Latch, Clocked Latch and Flip Flop Circuits, CMOS D Latch and Edge Triggered Flipflop.

3.1 INTRODUCTION

The word 'sequential' means that things happen in a sequence, i.e., one after another. In sequential logic circuits, the clock signal determines when things will happen next. A sequential logic circuit can be implemented by adding feedback to a combinational logic circuit as shown in Fig. 3.1.1.

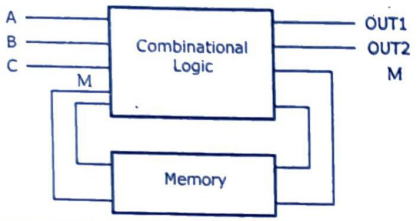


Fig. 3.1.1 Block Representation of Sequential Circuits

In sequential logic circuits, the output is determined by the current inputs as well as the previously applied input variables. The regenerative behavior is due to either a direct or indirect feedback connection between the outputs and the input.

Sequential logic circuits can be used to construct complex circuits such as counters, shift registers, latches or memories. To operate these types of circuits sequentially, we need clock pulse to change their states. Sequential logic circuits that use the clock signal for synchronization may change their state on either the rising or falling edge or both of the actual clock signals.

The classification of the logic circuits based on their temporal behavior is shown in Fig. 3.1.2.

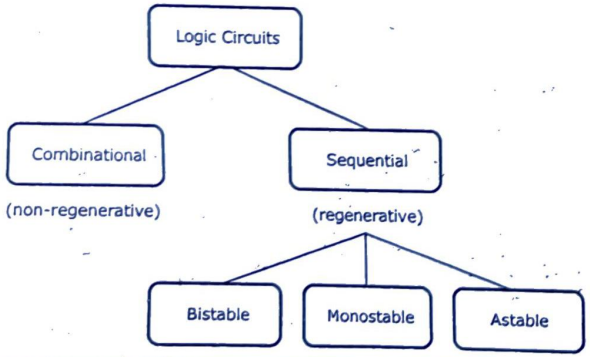


Fig. 3.1.2 Classification of Logic Circuit based on their Temporal Behavior

From Fig. 3.1.2 we can say that the critical components of sequential systems are the regenerative circuits which can be classified into three main groups,

(1) **Bistable :** It has two stable states, each of which can be attained under certain input and output conditions.

(2) **Monostable :** It has only one stable state. Even if the circuit is temporarily perturbed to the opposite state, they will return in time to their stable state.

(3) **Astable :** It has no stable state and oscillate between several states.

However, the bistable circuits are the most widely used and all the basic latches and flip-flops, register and memory elements used in digital system fall into this category. In this unit, we will discuss about the behavior of a basic bistable element followed by CMOS implementation of SR latch, JK latch, D latch and memories.

3.2 BEHAVIOR OF BISTABLE ELEMENTS

The basic bistable element is comprised of two identical inverters connected back-to-back i.e., output of one is connected to input of other and vice-versa as shown in Fig 3.2.1(a). The Voltage Transfer Characteristics (VTC) of the inverter pair bistable element are shown in Fig. 3.2.1(b).

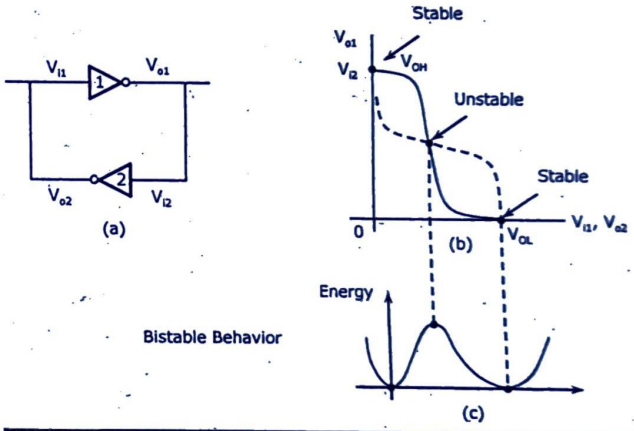
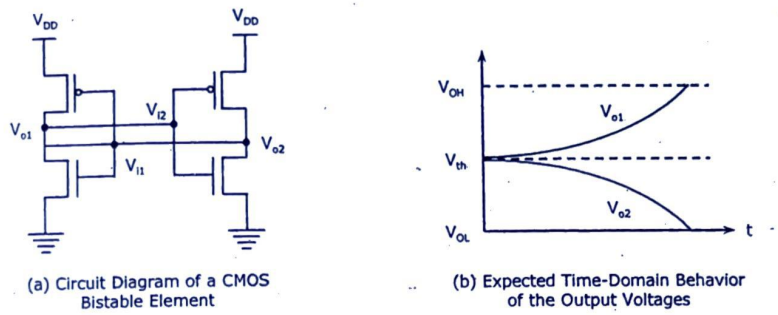


Fig. 3.2.1 Static Behavior of the Two-Inverter Basic Bistable Element

It is seen that the two VTC's intersect at three points. From the qualitative view of the potential energy levels corresponding to the three operating points as shown in Fig. 3.2.1(c) we will find that the latch has two stable states where the dc voltage transfer curves cross at the V_{OH} and V_{OL} points as shown in Fig. 3.2.1(b), but also exhibits an unstable state when the VTC's cross near their V_{th} switching points. In actual physical circuits, the latch will never stay at the unstable points, since any small electrical noise in the circuit will trigger it to one side or the other.

Fig. 3.3.2 shows the CMOS bistable element and characteristics.



(a) Circuit Diagram of a CMOS Bistable Element

(b) Expected Time-Domain Behavior of the Output Voltages

Fig. 3.2.2 CMOS Bistable Element and Characteristics

The CMOS bistable element shown in Fig. 3.2.2(a) will either be in state '0' with V_{o1} at ground and V_{o2} at V_{DD} or in state '1' with V_{o1} at V_{DD} and V_{o2} at ground. Expected time-domain behavior of the output voltage are shown in Fig. 3.2.2(b).

Analysis of the Time-Domain Behavior : Consider a bistable circuit which is operating initially at unstable operating point i.e., $V_{o1} = V_{o2} = V_{th}$, as shown in Fig. 3.2.3.

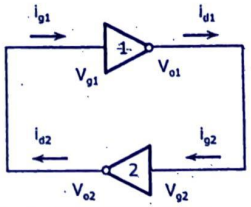


Fig. 3.2.3 Small-signal Input and Output Currents of the Inverters

For simplicity assume that the input (gate) capacitance C_g of each inverter is much larger than its output (drain) capacitance C_d. i.e., $C_g >> C_d$. Note that the drain current of each inverter is equal, consequently we can represent the drain current, as a function of gate voltage, as,

$$i_{g1} = i_{d2} = g_m V_{g2} \quad \text{... (3.2.1)}$$

$$i_{g2} = i_{d1} = g_m V_{g1}$$

Here the g_m is the small signal transcendence of the inverter. Now, the gate voltages of each inverter can be expressed interms of gate charger q_1 and q_2 as,

$$V_{g1} = \frac{q_1}{C_g}, V_{g2} = \frac{q_2}{C_g} \quad \text{... (3.2.2)}$$

Now, the small-signal gate current of each inverter can be written as,

$$i_{g1} = C_g \frac{dV_{g1}}{dt} \quad \text{... (3.2.3)}$$

$$i_{g2} = C_g \frac{dV_{g2}}{dt}$$

Combining Eqs. (3.2.1) and (3.2.3), we get,

$$g_m V_{g2} = C_g \frac{dV_{g1}}{dt} \quad \text{... (3.2.4)}$$

$$g_m V_{g1} = C_g \frac{dV_{g2}}{dt} \quad \text{... (3.2.5)}$$

Using Eqs. (3.2.2), (3.2.4) and (3.2.5), we get,

$$\frac{g_m}{C_g} q_2 = \frac{dq_1}{dt} \quad \text{... (3.2.6)}$$

$$\frac{g_m}{C_g} q_1 = \frac{dq_2}{dt} \quad \text{... (3.2.7)}$$

Combining Eqs. (3.2.6) and (3.2.7) to get a second order differential equation, we get,

$$\frac{g_m}{C_g} q_1 = \frac{C_g}{g_m} \frac{d_2 q_1}{dt^2}$$

$$\Rightarrow \quad \frac{d^2 q_1}{dt^2} = \left(\frac{g_m}{C_g}\right)^2 q_1 \quad \text{... (3.2.8)}$$

$$\Rightarrow \quad \frac{d^2 q_1}{dt^2} = \frac{1}{\tau_0^2} q_1 \quad \text{... (3.2.9)}$$

Where,

$$\tau_0 = \frac{C_g}{g_m}$$

Now the time-domain solution of Eq. (3.2.9) for q_1, is,

$$q_1(t) = \frac{q_1(0) - \tau_0 q_1'(0)}{2} e^{-t/\tau_0} + \frac{q_1(0) + \tau_0 q_1'(0)}{2} e^{t/\tau_0} \quad \text{... (3.2.10)}$$

With initial condition,

$$q_1(0) = C_g V_{g1}(0) \quad \text{... (3.2.11)}$$

Replacing the gate charge of both inverters with the corresponding output voltage variables (note $V_{g1} = V_{01}$ and $V_{g2} = V_{01}$), we get,

$$V_{02}(t) = \frac{1}{2}[V_{01}(0) - \tau_0 V'_{01}(0)]e^{-t/\tau_0} + \frac{1}{2}[V_{01}(0) + \tau_0 V'_{01}(0)]e^{t/\tau_0} \quad \ldots (3.2.12)$$

$$V_{01}(t) = \frac{1}{2}[V_{01}(0) - \tau_0 V'_{01}(0)]e^{-t/\tau} + \frac{1}{2}[V_{01}(0) + \tau_0 V'_{01}(0)]e^{t/\tau_0} \quad \ldots (3.2.13)$$

For large value of t, Eqs. (3.2.12) and (3.2.13) can be reduced to,

$$V_{01} \approx \frac{1}{2}[V_{01}(0) + \tau_0 V'_{01}(0)]e^{t/\tau_0}$$

$$V_{02} \approx \frac{1}{2}[V_{02}(0) + \tau_0 V'_{02}(0)]e^{t/\tau_0} \quad \ldots (3.2.14)$$

From the Eq. (3.2.14), it is clear that the magnitude of both output voltages increases exponentially with time. Depending upon the polarity of the initial small changes $dV_{01}(O)$ and $dV_{02}(O)$ the output voltage of both inverters will diverge from V_{th} to either V_{OL} or V_{OH}. Both the voltages diverge exponentially in opposite directions as shown in Fig. 3.2.4.

V_{01} : $V_{th} \rightarrow V_{OH}$ or V_{OL}

V_{02} : $V_{th} \rightarrow V_{OL}$ or V_{OH}

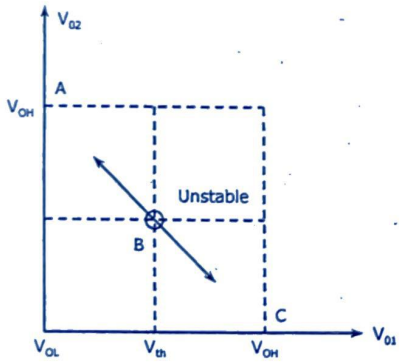


Fig. 3.2.4 Representation of Phase-Plane for the Bistable Circuit

3.3 SR LATCH CIRCUIT

The bistable element discussed in the previous section preserves its stable as long as the power supply voltage is provided, hence the circuit can perform a simple memory function if holding its state. However, there was no provision for allowing its state to be changed externally from one stable state to the other. To incorporate this facilities we have to add external trigger input.

We can add inputs in the bistable circuit by replacing the two cross-coupled inverters with two cross-coupled NOR gates or NAND gates.

This modified circuit has two inputs S and R two outputs Q and $\bar{Q}$, where $\bar{Q}$ represents the complement of output Q. Due to these two inputs S and R, the resulting latch is known as SR latch, where S and R stand for set and Reset. The stored bit is present on the output Q.

SR is the most fundamental latch when static gates are used as building blocks for any circuit design. The characteristic equation for SR latch is,

$$Q_{n+1} = S + \bar{R}Q$$

3.3.1 CMOS SR Latch : NOR Gate Version

The NOR gate implemented latch and its block diagram is shown in Fig. 3.3.1,

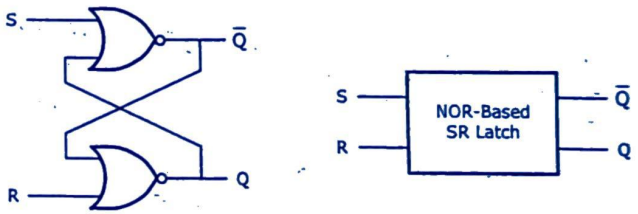


Fig. 3.3.1 Gate-Level Schematic and Block Diagram of the NOR-based

The CMOS implementation of a SR latch with S and R as the two triggering inputs is shown in Fig. 3.3.2.

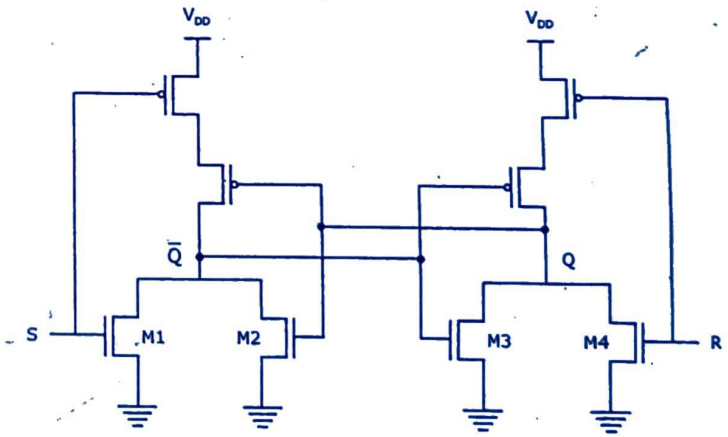


Fig. 3.3.2 CMOS SR Latch Circuit based on NOR2 Gates

Operation of CMOS NOR-Based SR Latch : In Fig. 3.3.2, if set goes high, M1 is turned on, forcing $\overline{Q}$ low which in turn pulls Q high. If reset goes high, M4 is turned on, Q is pulled low and $\overline{Q}$ is pulled high. If both set and reset are low, both M1 and M4 are off and the latch holds its existing state indefinitely. If both set and reset go high, both Q and $\overline{Q}$ are pulled low, giving an indefinite state. Therefore, R = S = 1 is not allowed.

The truth table of CMOS NOR-based SR latch is given in Table 3.3.1.

Table 3.3.1 Truth Table of the NOR-based SR Latch Circuit

S	R	Q_{n+1}	$\overline{Q_{n+1}}$	Operation
0	0	Q_n	$\overline{Q_n}$	Hold
1	0	1	0	Set
0	1	0	1	Reset
1	1	0	0	Not allowed

The operation of the CMOS SR latch shown in Fig. 3.3.2 can be examined in more detail by considering the operating modes of the four NMOS transistors M1, M2, M3 and M4. If the set input (S) is equal to V_{OH} and the reset input (R) is equal to V_{OL}, both M1 and M2 will be ON. Consequently, the voltage at $\overline{Q}$ will be $V_{OL} = 0$. At the same time, both M3 and M4 are turned OFF resulting in a logic high voltage V_{OH} at node Q. If the reset (R) input is equal to V_{OH} and the set (s) input is equal to V_{OL}, the situation will be reversed i.e., M1 and M2 will turn OFF and M3 and M4 will be turned ON.

When both the input voltages are equal to V_{OL} there are two possibilities. Depending on the previous state of the SR latch, either M2 or M3 will be ON while both of the triggering transistors M_1 and M_4 are OFF. This will generate a logic low-level of $V_{OL} = 0$ at one of the output nodes while the complementary output node is at V_{OH}. The operating modes of the transistors in NOR-based CMOS SR latch circuit is given in Table 3.3.2.

Table 3.3.2 Operation Modes of the Transistors in the NOR-based CMOS SR Latch Circuit

S	R	Q_{n+1}	$\overline{Q_{n+1}}$	Operation
V_{OH}	V_{OL}	V_{OH}	V_{OL}	M1 and M2 on, M3 and M4 off
V_{OL}	V_{OH}	V_{OL}	V_{OH}	M1 and M2 off, M3 and M4 on
V_{OL}	V_{OL}	V_{OH}	V_{OL}	M1 and M4 off, M2 on, or
V_{OL}	V_{OL}	V_{OL}	V_{OH}	M1 and M4 off, M3 on

3.3.1.1 Transient Analysis of CMOS SR Latch

• For the transient analysis, we have to consider an event which results in a state change i.e., either an initially reset latch being set by applying a set signal or an initially set latch being reset by applying the reset signal. To calculate the switching times for both output node, we will have to consider the parasitic capacitance associated with each node.

The total lumped capacitance at each output node can be expressed as,

$$C_Q = C_{gb,2} + C_{gb,5} + C_{db,3} + C_{db,4} + C_{db,7} + C_{sb,7} + C_{db,8} \quad \ldots (3.3.1)$$

$$C_{\overline{Q}} = C_{gb,3} + C_{gb,7} + C_{db,1} + C_{db,5} + C_{sb,5} + C_{db,6}$$

The CMOS SR NOR latch with lumped capacitances at both output nodes is shown in Fig. 3.3.3.

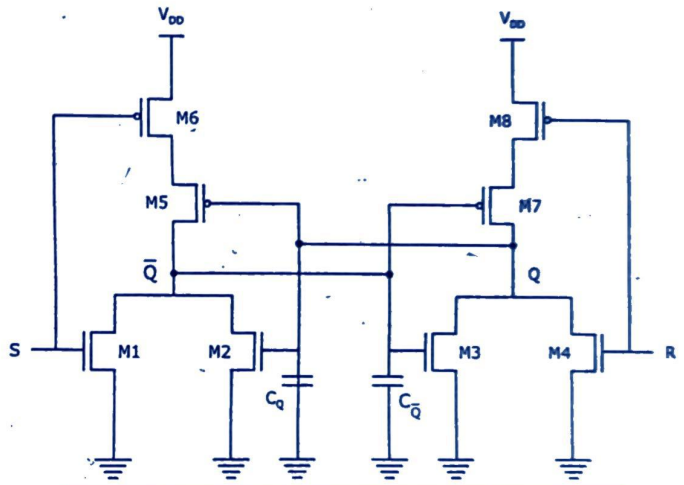


Fig. 3.3.3 Circuit Diagram of the CMOS SR Latch Showing the Lumped Load Capacitances at Both Output Nodes

In Fig. 3.3.3, assuming that the latch is initially reset and that a set operation is being performed by applying S = 1 and R = 0, the rise time associated with node Q can now be estimated as,

$$\tau_{rise,Q}(\text{SR-latch}) = \tau_{rise,Q}(\text{NOR2}) + \tau_{fall,\overline{Q}}(\text{NOR 2}) \quad \ldots (3.3.2)$$

However, this approach leads to a simple first order prediction for the time delay as opposed to the simultaneous solution of two coupled differential equations.

3.3.2 Depletion Load NMOS SR Latch : NOR Version

A depletion load version of the NOR-based SR latch is shown in Fig. 3.3.4.

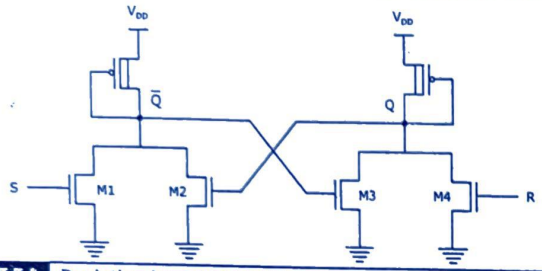
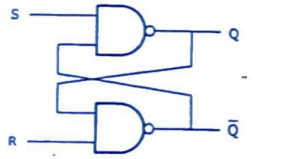


Fig. 3.3.4 Depletion-load NMOS SR Latch Circuit based in NOR2 Gates

Functionally it is the same as CMOS version. But, interms of noise margin and power dissipation CMOS is better than NMOS because they provide a full output swing between ground i.e., (OV) and V_{DD}.

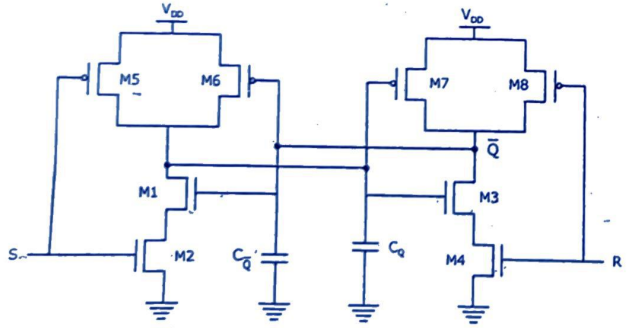
3.3.3 CMOS SR Latch : Nand Gate Version

The NAND-based SR latch contains the basic memory cell (back-to-back inverters) built into two NAND gates to allow setting the state of the latch. The gate-level symbol, block diagram and CMOS NAND-based SR latch are shown in Fig. 3.3.5.



(a) Gate Level Circuit of NAND Bases SR Latch

(b) Block Diagram



(c) Transistor Level CMOS NAND Based SR Latch

Fig. 3.3.5 CMOS SR Latch

The circuit responds to active low S and R inputs. If S goes to 0 (while R = 1), Q goes high, pulling $\bar{Q}$ low and the latch enters set state. If R goes to 0 (while = 1), $\bar{Q}$ goes high pulling Q low and the latch is Reset. Hold state requires both S and R to be high. S = R = 0 is not allowed, it would result in an indeterminate state. The truth table of NAND-based CMOS SR latch circuit is given in Table 3.3.3.

Table 3.3.3 Truth Table of NAND-based CMOS SR Latch

S	R	Q_{n+1}	$\overline{Q_{n+1}}$	Operation
0	0	1	1	Not allowed
0	1	1	0	Set
1	0	0	1	Reset
1	1	Q_n	$\bar{Q}_n$	Hold

3.3.4 Depletion Load NMOS SR Latch : NAND Version

A depletion load version of the NAND based SR latch is shown in Fig. 3.3.6.

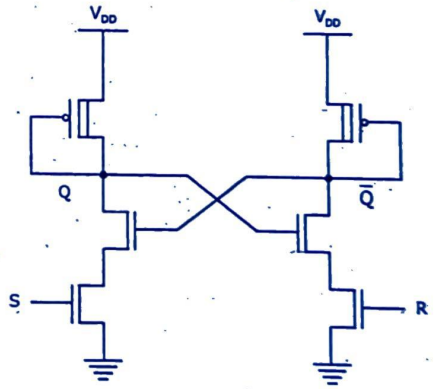


Fig. 3.3.6 Depletion-load NMOS NAND-based SR Latch Circuit

Functionally it is the same as the CMOS version. But in terms of noise margin, the CMOS NAND-based SR latch is better than NMOS because it consumes less dc power.

3.4 CLOCKED LATCH AND FLIP-FLOP CIRCUITS

The SR latch discussed previously is an example of an asynchronous sequential circuit which will respond to the change occurring in input signals at a circuit delay dependent time point during their operation. To facilitate synchronous operation, the circuit response can be controlled simply by adding a gating clock signal to the circuit so that the outputs will respond to the input level only during the active period of a clock pulse.

3.4.1 Clocked SR Latch : NOR Version

The clocked NOR-based SR latch, contains the basic memory cell built into two NOR gates to allow setting the state of the latch with a clock added as shown in Fig. 3.4.1.

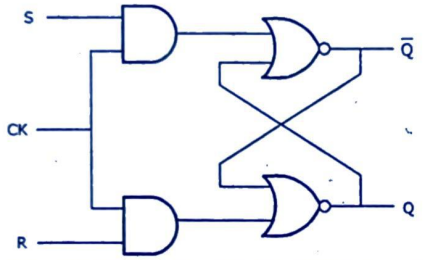


Fig. 3.4.1 Gate-level Schematic of the Clocked NOR-based SR Latch

In Fig. 3.4.1, if the Clock (CK) is equal to logic "0", the input signals have no influence upon the circuit response. The outputs of the two AND gates will remain at logic "0" which force the SR latch to hold its current state regardless of the S and R input signals. When the clock goes to level "1", the logic levels applied to the S and R inputs and permitted to reach the SR latch and possibly change its state.

The circuit is strictly level sensitive during active clock phases i.e., any change occurring in the S and R input voltage when the clock level is equal to "1" will be rejected onto the circuit outputs. Consequently, when narrow spike glitch occurring during an active clock phase can set or reset the latch if the loop delay is shorter than the pulse width. The sample input and output waveforms are shown in Fig. 3.4.2.

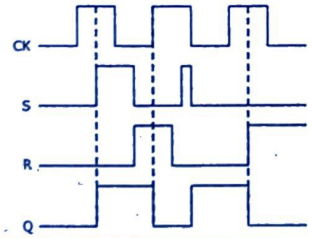


Fig. 3.4.2 Sample Input and Output Waveforms Illustrating the Operation of the Clocked NOR-based SR Latch Circuit

The sample waveforms shown in Fig. 3.4.2 shows the level-sensitive nature of clocked SR latch. Note that four times Q changes state,

(1) When S goes high during positive CK.

(2) On leading CK edge after changes in S and R during CK low time.

(3) A positive glitch in S while CK is high.

(4) When R goes high during positive CK.

The CMOS AOI implementation of clocked NOR-based SR latch for Fig. 3.4.1 is shown in Fig. 3.4.3.

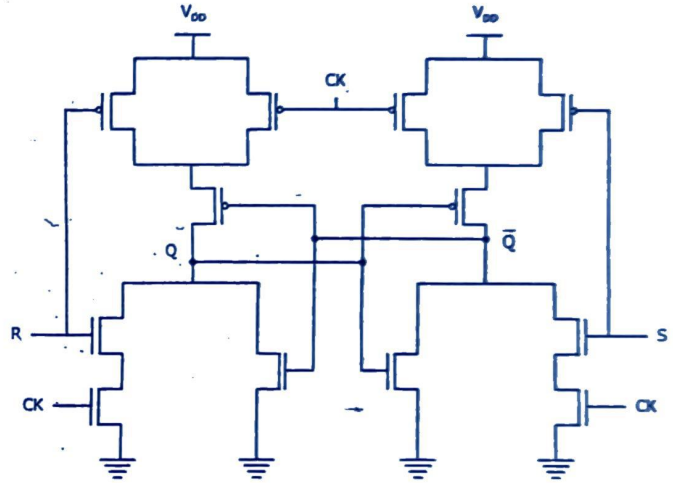


Fig. 3.4.3 AOI-based Implementation of the Clocked NOR-based SR Latch Circuit

Fig. 3.4.3 has only 12 transistors. When CK is low, two series legs in N-tree are open and two parallel transistors in P-tree are ON, thus, retaining state in the memory cell. When CK is high, the circuit becomes simply a NOR-based CMOS latch which will respond to inputs S and R.

3.4.2 Clocked SR Latch : NAND Version

We can also implement, the NAND based flip-flop as shown in Fig. 3.4.4. Note that in this circuit both the inputs S and R as well as clock signals are active low. It means when the clock is logic low, there will be an influence in the output regarding the inputs S and R. We can implement this circuit at transistor level using OAI structure.

Fig. 3.4.4 shows the NAND based SR flip-flop, with active low input.

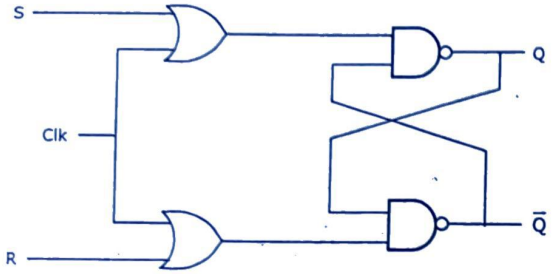
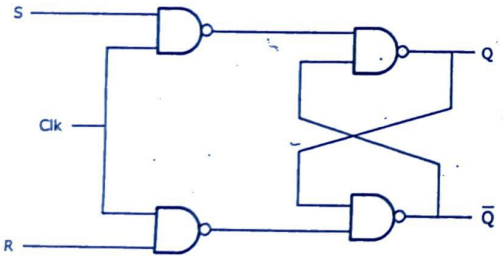
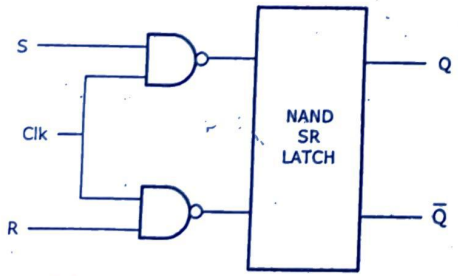


Fig. 3.4.4 NAND Based SR Flip-Flop, with Active Low Input

The another implementation of NAND based SR flip-flop is shown in Fig. 3.4.5. Here, both input signals as well as clock are actual high. When the CLK = 0 i.e., logic low the latch will preserve its previous state, on the other hand when CLK = 1, S = 1 and R = 0 the output Q will be set and when CLK = 1, S = 0 and R = 1 the latch will be reset.



(a) NAND Based SR Flip-Flop with Active High Input



(b) Block (Partial) Representation of (a)

Fig. 3.4.5 NAND-based SR Flip-flop

3.4.3 Clocked CMOS JK Latch : NAND Version

The SR latch has a problem that when both S and R are high, its state becomes indeterminate. This problem can be overcome by adding two feedback lines from the outputs to inputs, such that all states in the truth table are allowable as shown in Fig. 3.4.6. The resulting circuits are called as a JK flip-flops.

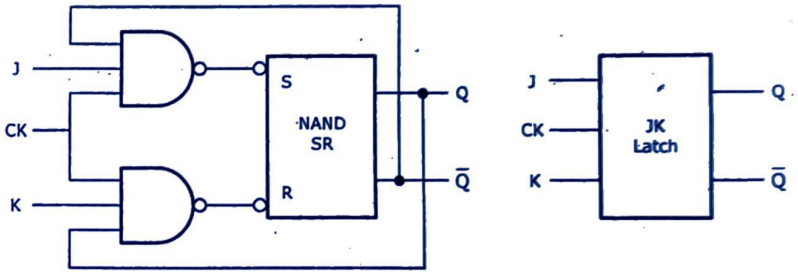


Fig. 3.4.6 Gate-level Schematic of the Clocked NAND-based JK Latch Circuit

The all NAND implementation of the JK latch with active high inputs is shown in Fig. 3.4.7.

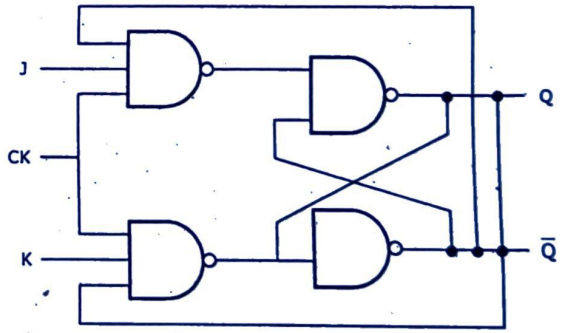


Fig. 3.4.7 All-NAND Implementation of the Clocked JK Latch Circuit

When CK = 0 or logic low it will preserve its state or we can say that JK latch will respond to CK when it is equal to logic high. The input J and K of the circuit of the circuit correspond to set and reset inputs of basic SR latch. The flip-flop is in set position when CK = 1, J = 1, K = 0 and it can be reset when CK = 1, J = 0, k = 1. If CK = 1, J = 1, K = 1 the latch will simply switches its state due to feedback.

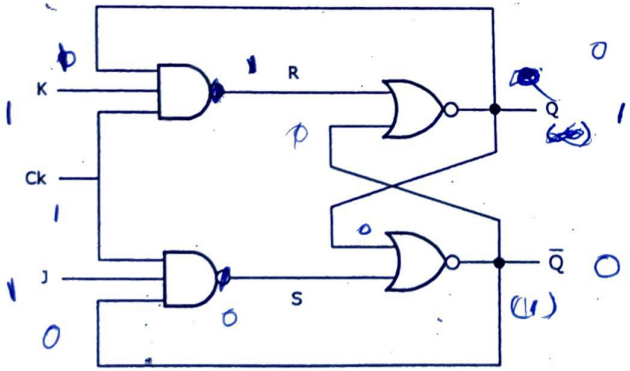
We can summarize the operation of JK flip-flop as given in Table 3.4.1.

Table 3.4.1 Detailed Truth Table of the JK Latch Circuit

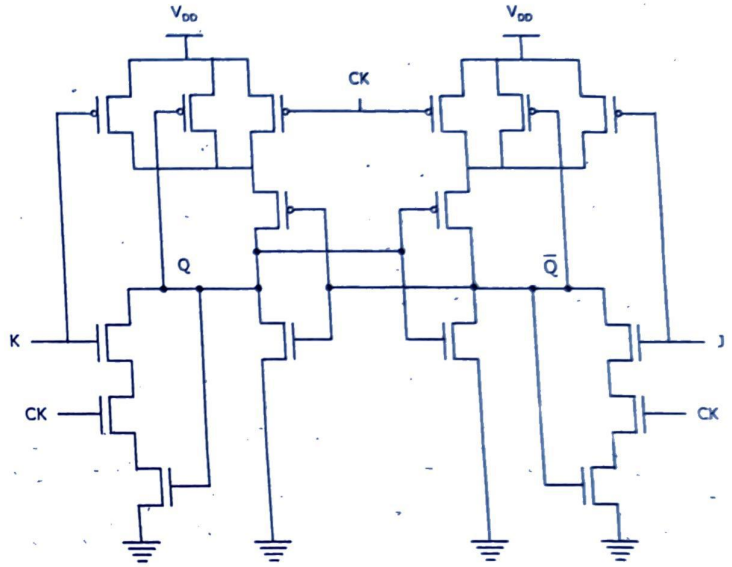
J	K	Q_n	$\overline{Q_n}$	S	R	Q_{n+1}	$\overline{Q_{n+1}}$	Operation
0.	0	0	1	1	1	0	1	Hold
		1	0	1	1	1	0	
0	1	0	1	1	1	0	1	Reset
		1	0	1	0	0	1	
1	0	0	1	0	1	1	0	Set
		1	0	1	1	1	0	
1	1	0	1	0	1	1	0	Toggle
		1	0	1	0	0	1	

3.4.4 Clocked CMOS JK Latch : NOR Version

The NOR-based implementation of clocked JK latch and its CMOS AOI realization is shown in Fig. 3.4.8.



(a) Gate-Level Schematic of the Clocked NOR-Based JK Latch Circuit



(b) CMOS AOI Realization of the JK Latch

Fig. 3.4.8 JK Latch Element

The AOI-based circuit structure results in a relatively low transistor count and more compact circuit compared to all NAND realization. If both the inputs are equal to logic "1" during active phase of the clock pulse, the output of the circuit will oscillate continuously until either the clock becomes inactive (i.e., goes to zero), or one of the input signals goes to zero. In order to prevent the latch from oscillating continuously during the clock active time, the clock width must be kept smaller than the switching delay time of the latch. Otherwise, several oscillations may occur before the clock goes low again. In practise this may be difficult to achieve.

If the clock timing constraint is satisfied, then the output of the JK latch will toggle only one for each clock pulse, if both inputs are equal to logic '1'. The circuit which is operated exclusively in this mode is called a toggle switch as shown in Fig. 3.4.9.

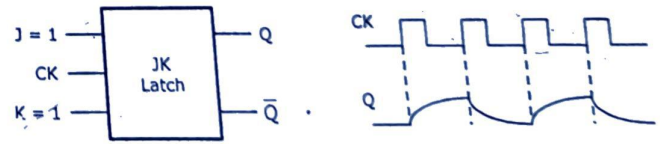


Fig. 3.4.9 Operation of the JK Latch as a Toggle Switch

3.4.5 Master-Slave Flip-Flop

Most of the timing limitations encountered in the previously examined clocked latch circuits can be prevented using two latch stages in a cascade configuration. The two cascaded stages are activated with opposite clock phases. The master slave flip-flop consisting of NAND based JK latches is shown in Fig. 3.4.10.

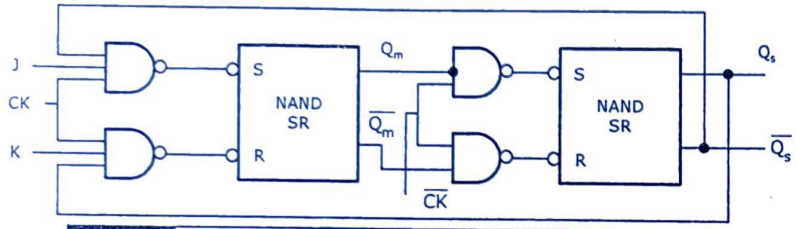


Fig. 3.4.10 Master-slave Flip-flop Consisting of NAND-based JK Latches

The input latch is called the "master" and it is activated when the clock pulse is high. During this phase, the inputs J and K allow data to be entered into the flip-flop and when the cock pulse switches to zero the master latch becomes inactive and the second-stage latch called the "slave" becomes active. The output levels of the flip-flop circuit are determined during this second phase based on the master stage outputs set in the previous phase.

Since the master outputs and slave stages are effective has decoupled from each other with the opposite clocking scheme the circuit is never transparent i.e., change reason the master-slave JK FF allows for tagging when J = K = "1" but it elements the possibility of uncontrolled oscillations since only one stage is active at any given time.

A JK master-slave FF (NOR-based version) is shown in Fig. 3.4.11.

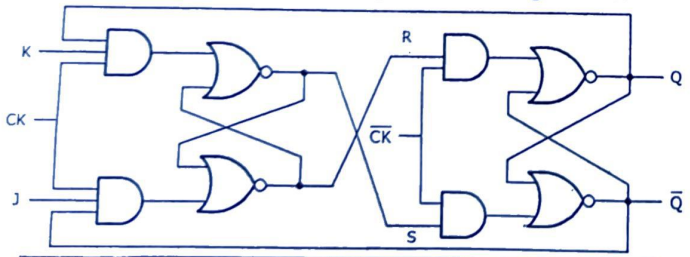


Fig. 3.4.11 NOR-based Realization of the JK Master-slave Flip-flop

The feedback paths that occur from Q and $\overline{Q}$ slave outputs to the master inputs AOI gates. The circuit does not exhibit any tendency to oscillate when J = K = 1 no matter how long the clock period. Since opposite clock phases activate the master and slave latches separately. NOR-based version uses 4 AOI CMOS gates (28 transistors). But the latches can be susceptible to "ones catching".

3.4.5.1 JK MS Flip-Flop Problem : One's Catching

Although the JK master-slave flip-flop can be considered edge-triggered in regards to a change in Q_S at the negative clock edge, it is actually level sensitive in regards to noise on J or K during the clock high interval. Note that the positive glitch in J which sets the master latch at Q_m = 1 during the clock high interval and then also reflects itself in Q_S = 1 at the negative going clock edge, called "One's catching". Same problem can occur with a glitch in K during clock high, causing a reset operation. Since the master latch actually sets and latches on the noise glitch, the error is then transmitted to the slave latch during clock. Sample input and output waveforms of MS FF circuit are shown in Fig. 3.4.12.

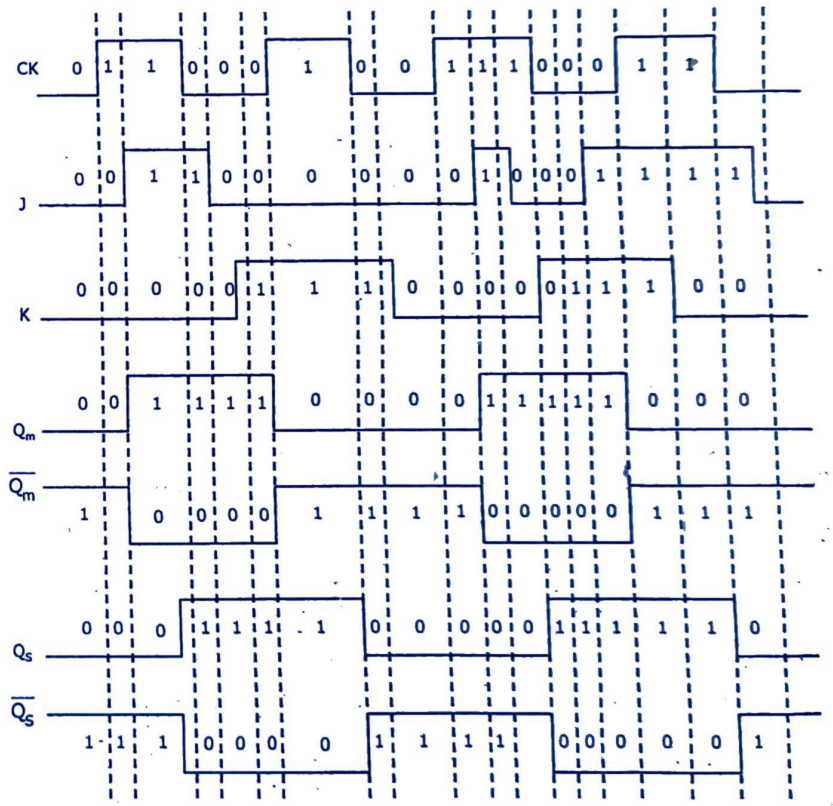


Fig. 3.4.12 Sample Input and Output Waveforms of the Master-slave Flip-flop Circuit

3.5 CMOS D-LATCH AND EDGE TRIGGERED FLIP-FLOP

In the digital integrated circuits, a large selection of CMOS-based sequential circuits have gained popular, with the wide spread use for CMOS circuit techniques, especially in VLSI design. The flip-flop and latch circuits can be implemented with CMOS gates and their design is so simple and straight forward. However, we can use AOI or OAI gates for the implementation of latch and flip-flops because simple CMOS circuits require more number of transistors.

3.5.1 CMOS D-Latch Implementation

A D-latch is implemented at the gate level by simply utilizing a NOR-based S-R latch, connecting D to input S and connecting $\overline{D}$ to input R with an inverter as shown in Fig. 3.5.1.

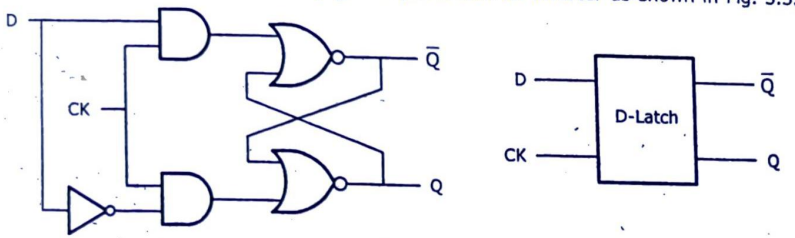


Fig. 3.5.1 Gate-level Schematic and the Block Diagram View of the D-latch

In Fig. 3.5.1, when CK goes high, D is transmitted to output (and $\overline{D}$ to $\overline{Q}$). When CK goes low, the latch remains in its previous state. Now can say that, CK is an enable input which allows data to be accepted into the D flip-flop when active.

The D-latch implementation with CMOS TG switches is shown in Fig. 3.5.2.

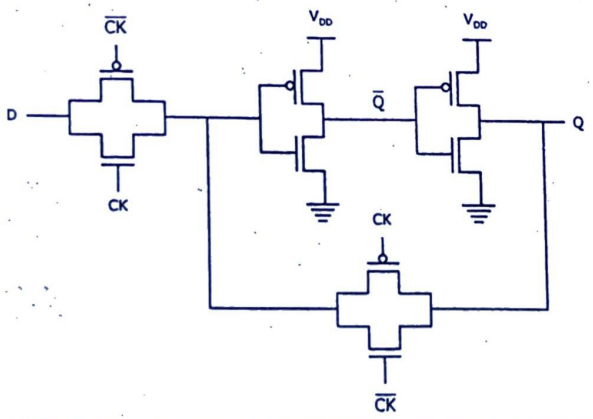


Fig. 3.5.2 CMOS Implementation of the D-latch (Version 1)

In Fig. 3.5.2, the input TG is activated with CK while the latch feedback loop TG is activated with $\overline{CK}$. Input D is accepted when CK is high. When CK goes low, the input is open-circuited and the latch is set with the prior data D.

A schematic view of the D-latch can be obtained using simple switches in place of the TG's as shown in Fig. 3.5.3.

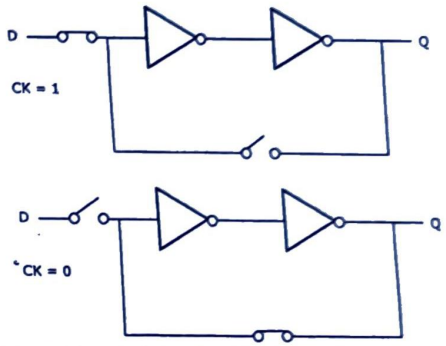


Fig. 3.5.3 Schematic View of D-latch using Inverters

When CK = 1, the input switch is closed allowing input data into the latch. When CK = 0, the input switch is opened and the feedback loop switch is closed, setting the latch.

An alternate, preferred version of the CMOS D-latch is implemented with two tri-state inverters and a normal CMOS inverter as shown in Fig. 3.5.4.

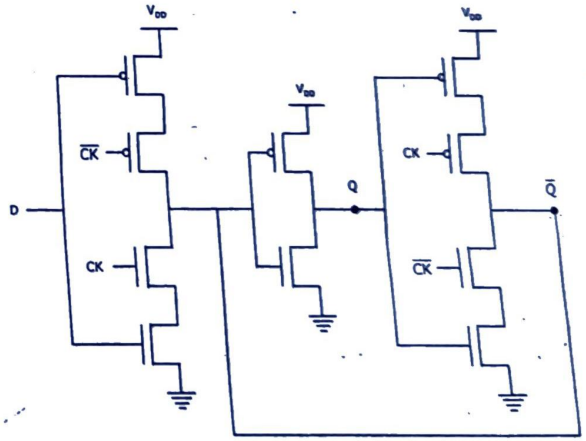


Fig. 3.5.4 CMOS Implementation of the D-Latch (Version 2)

Functionally it is similar to previous chart D-latch. When CK is high, the first tri-state inverter sends the inverted input through the second inverter, while the second tri-state is in its high z state. Output Q is following input D. When CK is low, the first tri-state goes into its high z state, while the second tri-state inverter closes the feedback loop, holding the data Q and $\overline{Q}$ in the latch.

3.5.1 CMOS D-Latch Timing

Latch Timing : For the system to work correctly, the set-up time, hold time and pulse-width must be sufficient for each bistable element,

(1) Set-up time (t_{setup}) is the minimum time that the data input of a bistable element to be held stable prior to the active clock signal.

(2) Hold time (t_{hold}) is the minimum time that the data input of the bistable element must be held stable after the active clock signal disappears.

(3) The pulse width is addition of t_{setup} and t_{hold} as shown in Fig. 3.5.5.

Timing Diagram : In order to guarantee adequate time to get correct data at the first inverter input before the input switch opens, the data must be valid for a given time (t_{setup}) prior to the CK going low. To guarantee adequate time to set the latch with correct data, the data must remain valid for a time (t_{hold}) after the CK goes low. Violations of t_{setup} and t_{hold} can cause metastability problems and chaotic transient behaviour as shown in Fig. 3.5.5.

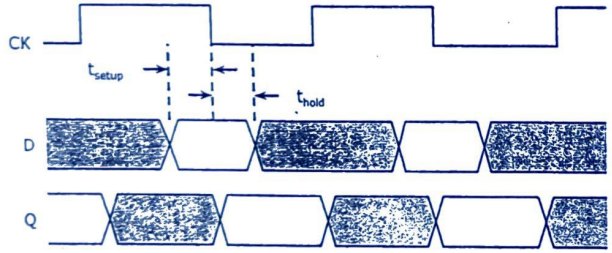


Fig. 3.5.5 CMOS D-Latch Timing

3.5.2 CMOS Edge-Triggered D-Flip-Flop

The CMOS latch is not edge-triggered storage element. Because the output changes according to the input, i.e., the latch is transparent, while the clock is high. The transparency property make the application this D-latch unsuitable for counters and some data storage implements. So, to implement these applications we introduces the edge. Triggered MS D-Flip-flops there are two types of CMOS edge-triggered elements,

(1) Negative edge-triggered MS D-Flip-Flop (DFF).

(2) Positive edge-triggered MS D-Flip-Flop (DFF).

3.5.2.1 Negative Edge-Triggered MS D Flip-Flop

Fig. 3.5.6 shows a D-flip-flop, constructed by cascading two D-latch circuits. The first stage (master) is driven by the clock-signal. However, the second stage (slave) is driven by the inverted clock signal. Thus, the master stage is positive level sensitive while the slave stage is negative level sensitive.

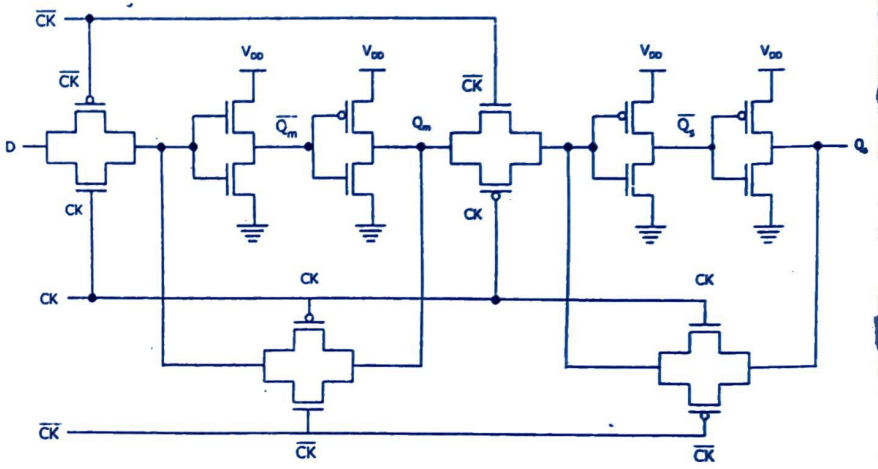


Fig. 3.5.6 CMOS Negative (Falling) Edge-triggered Master-slave D-Flip-flop (DFF)

When the clock is high the master stage follows the D-input while the slave stage holds the previous value. When the clock changes from logic "1" to logic "0", the master latch cases to sample the input and store the D-value at the time of clock transition. At the same time the slave latch locks in the master latch output and the master stage starts sampling the input again. Thus, the circuit is negative edge-triggered D flip-flop by virtue of the fact that it samples the input at the falling edge of the clock pulse.

3.5.2.2 CMOS Positive Edge-Triggered D-Flip-Flop (DFF)

Another implementation of edge-triggered D-flip-flop is shown in Fig. 3.5.7, which consists of six NAND3 gates. This D-flip-flop is positive edge-triggered as shown in waveform chart in Fig. 3.5.8. Initially all the signal values except for S are 0 i.e., (S, R, CK, D) = (1, 0, 0, 0) and the Q = 0. In the second phase, both D and R switch to 1 i.e., (S, R, CK, D) = (1, 1, 0, 1) but no change in Q occurs and the Q = 0. In the third phase, if CK = 1 i.e., (S, R, CK, D) = (1, 1, 1, 1), the output of gate 2 switches to 0, which in turn sets the output of the last stage SR latch to 1. Thus, the output of this D flip-flop switches to 1 at the positive going edge of the clock signal, CK. In the ninth phase of the waveform, Q output is not affected by negative going edge of CK or by other signal changes.

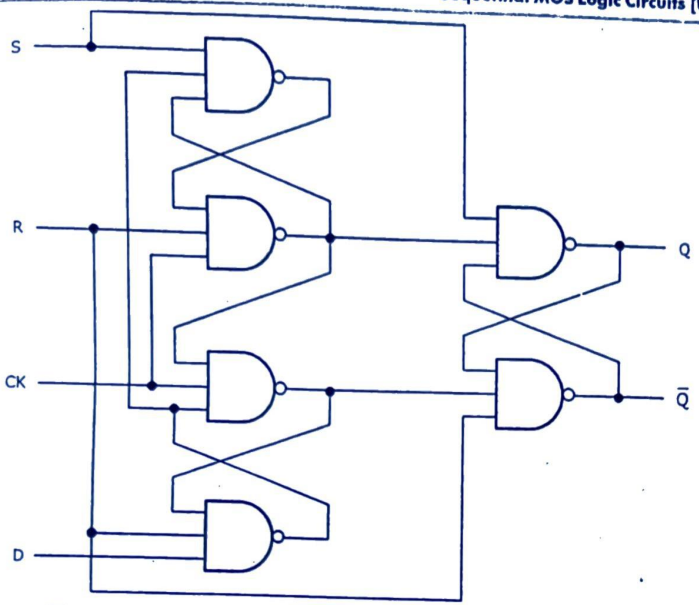


Fig. 3.5.7 NAND3-based Positive Edge-triggered D-flip-flop (DFF)

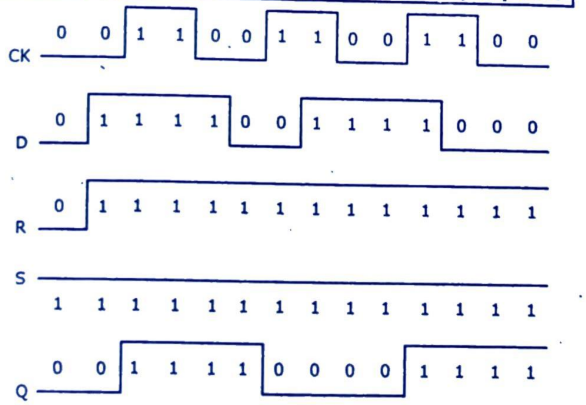


Fig. 3.5.8 Timing Diagram of the Positive Edge-triggered D-flip-flop (DFF)

❖ ❖ ❖

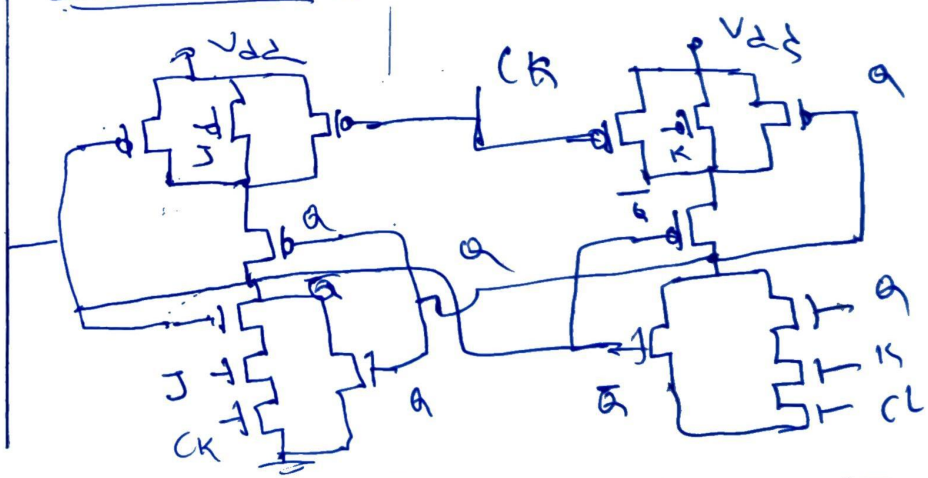
UNIT-IV

DYNAMIC LOGIC CIRCUITS

SYLLABUS [M.TECH - H&K]

Basic Principle, Voltage Bootstrapping, Synchronous Dynamic Pass Transistor Circuits, Dynamic CMOS Transmission Gate Logic, High Performance Dynamic CMOS Circuits.

JK flip flop



4.1 DYNAMIC LOGIC CIRCUITS

Digital circuits are commonly implemented using static CMOS logic design which allows versatile implementation of logic functions based on static, steady-state behavior of simple CMOS structures. The outputs are generated depending on the applied input voltage after a line delay and the output voltage level is prescribed as long as the power supply is provided. However, the static design approach may require a large number of transistors to implement a function and this may result in larger delays.

In most VLSI circuit design problems, reduction of circuit delay and silicon area is a major objective. Dynamic logic circuits, in those situations, offer several significant advantages over static CMOS logic circuits. Operation of all dynamic logic circuits depends on temporary storage of charge in parasitic node capacitances, instead of relying on steady-state circuit behavior. Since, the capacitors cannot retain charge for an indefinite period of time, hence periodic updating of internal node voltages is required. In order to control charge refreshing, the dynamic logic circuits may require periodic clock signals.

Dynamic logic techniques save area by reducing the number of transistors per gate and save power by reducing the number of gates and the static current in structures such as flip-flops and shift registers. Dynamic CMOS circuits save chip area while enhancing speed over conventional CMOS circuits, but precautions must be taken to ensure proper operation. By using the common clock signals, the system enables synchronize the operation of various circuit blocks. Capability of temporarily storing a state at a capacitive node allows implementing simple sequential circuits with memory functions.

Disadvantage of the dynamic storage is the use of small-sized, leaky capacitors for storing logic values. They must clocked at a minimum operating frequency in order to maintain their charge.

4.1.1 Dynamic D-Latch

The dynamic D-latch circuit is shown in Fig. 4.1.1.

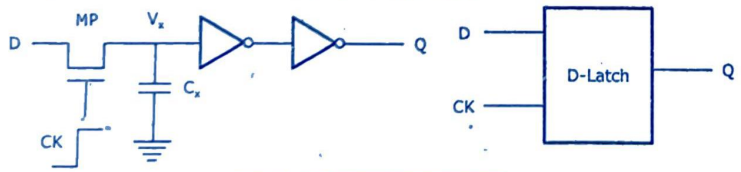


Fig. 4.1.1 Dynamic D-Latch

In Fig. 4.4.1, the parasitic input capacitance C_x plays an important role in dynamic operation of circuit. Input pass transistor is driven by external clock signal. When CK = 1, Pass Transistor (MP) turns ON, C_x charged or discharged through MP, depending on input (D) voltage level, Q assumes same logic level as input. When CK = 0, MP turns OFF, C_x is isolated from D, amount of charge stored in C_x during last cycle determines output voltage level Q.

It can easily be seen that this circuit performs the function of simple D-latch. Infact, the transistor count can be reduced by removing the last inverter stage if the latch output can be inverted.

4.2 BASIC PRINCIPLES OF PASS TRANSISTOR CIRCUITS

The fundamental building block of dynamic logic circuits, consisting of an NMOS pass transistor, MP driving the gate of another NMOS transistor is shown in Fig. 4.2.1.

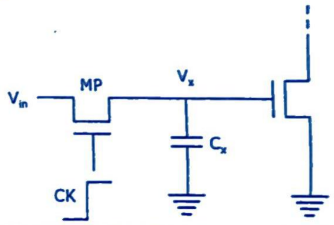


Fig. 4.2.1 NMOS Dynamic D-Latch

In Fig. 4.2.1, the pass transistor MP is driven by periodic clock signal and acts as an access switch to either charge up or down the parasitic capacitance C_x, depending on V_{in}.

When CK = 1, there is a possibility of two operations,

(1) Logic '1' transfer.

(2) Logic '0' transfer.

The output of depletion-load NMOS inverter depends on voltage V_x. MP provides only current path to the intermediate capacitive node (soft node) X. When CK = 0, the MP ceases to conduct and charge stored in the parasitic capacitance C_x continues to determine output level of the inverter.

4.2.1 Logic '1' Transfer

Let us assume that V_x = 0 V at time t = 0, when the clock signal at the gate of the pass transistor goes from 0 to V_{DD}. If V_{in} at logic 1, then $V_{in} = V_{OH} = V_{DD}$. Pass transistor MP provides the current path to the intermediate capacitive node V_x. Now for MP, $V_{DS} = V_{GS}$ which means $V_{DS} > V_{GS} - V_{T,n}$. The equivalent circuit for logic '1' transfer event is shown in Fig. 4.2.2.

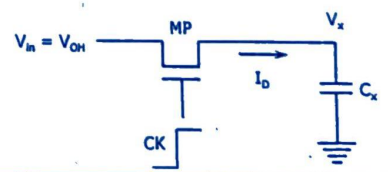


Fig. 4.2.2 Equivalent Circuit of the Logic "1" Transfer Event

As $V_{DS} > V_{GS} - V_{T,n}$, transistor MP is operating in the saturation region. Since MP is conducting in saturation and the saturation drain current starts to charge capacitor C_x, we can write,

$$C_x \frac{DV_x}{dt} = \frac{K_n}{2}(V_{DD} - V_x - V_{T,n})^2 \quad \ldots (4.2.1)$$

$$\Rightarrow \quad dt = \frac{2C_x}{K_n} \frac{dV_x}{(V_{DD} - V_x - V_{T,n})^2} \quad \ldots (4.2.2)$$

Integrating Eq. (4.2.2) on both sides with appropriate limits we get,

$$\int_0^t dt = \frac{2C_x}{K_n} \int_0^{V_x} \frac{dV_x}{(V_{DD} - V_x - V_{T,n})^2} \quad \ldots (4.2.3)$$

$$= \frac{2C_x}{K_n} \left(\frac{1}{V_{DD} - V_x - V_{T,n}} \right)_0^{V_x}$$

$$t = \frac{2C_x}{k_n} \left[\left(\frac{1}{V_{DD} - V_x - V_{T,n}} \right) - \left(\frac{1}{V_{DD} - V_{T,n}} \right) \right] \quad \ldots (4.2.4)$$

Now, Eq. (4.2.4) can be solved for $V_x(t)$ as,

$$V_x(t) = (V_{DD} - V_{T,n}) \frac{\left(\frac{k_n(V_{DD} - V_{T,n})}{2C_x} \right) t}{1 + \left(\frac{k_n(V_{DD} - V_{T,n})}{2C_x} \right) t} \quad \ldots (4.2.5)$$

Variation of node voltage V_x with respect to Eq. (4.2.5) is plotted as a function of time is shown in Fig. 4.2.3.

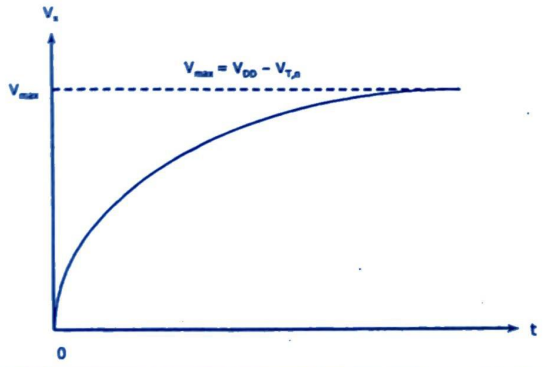


Fig. 4.2.3 Variation of V_x as a Function of Time During Logic "1" Transfer

The fact that the node voltage V_x rises from its initial value of 0 V and approaches $V_{max} = (V_{DD} - V_{T,n})$. The pass transistor MP will turn OFF when $V_x = V_{max}$, since at this point, its gate-to-source voltage (V_{GS}) is equal to its threshold voltage ($V_{T,n}$). Therefore, we can say that the voltage of node V_x cannot attain the full scale value of V_{DD} and always attains a value of $V_{DD} - V_{T,n}$ for a logic '1' transfer.

Pass Transistors in Series : The node voltages in the pass transistor chain during logic '1' transfer are shown in Fig. 4.2.4.

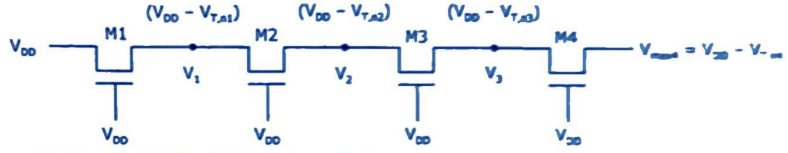


Fig. 4.2.4 Node Voltages in a Pass-transistor Chain during the Logic "1" Transfer

In Fig. 4.2.4, three pass transistor are cascaded and the output of one acts as the input of second one. If the initial input voltage is V_{DD}, then transistor M1 is in saturation and the voltage at node 1 is $V_{DD} - V_{T,n1}$. Assuming that the transistors are identical, transistor M2 operates at the saturation boundary and the voltage at node 2 is $V_{DD} - V_{T,n2}$. It can be seen that the output of the pass transistor chain is one threshold voltage lower than V_{DD} irrespective of the number of pass transistors in the chain.

Pass Transistors Driving Gate of Another Pass Transistors : Node voltages during the logic '1' transfer, when each transistor is driving another pass transistor is shown in Fig. 4.2.5.

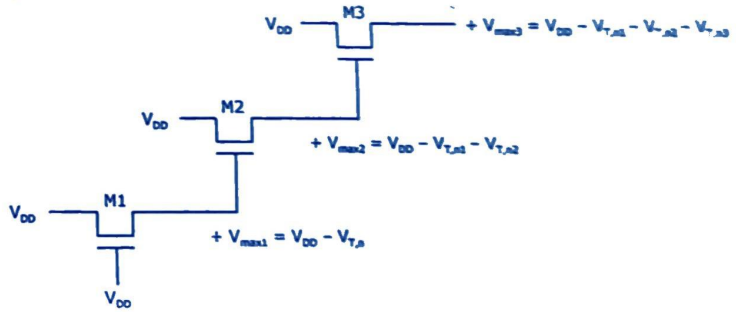


Fig. 4.2.5 Node Voltages during the Logic "1" Transfer, when each Pass Transistor is Driving another Pass Transistor

The output of pass transistor M1 can reach the limit $V_{max1} = V_{DD} - V_{T,n1}$. This voltage drives the gate of transistor M2 which also operates in the saturation region. However, the gate-to-source voltage of M2 cannot exceed $V_{DD} - V_{T,n1}$. Hence, the output voltage of transistor can reach the limit $V_{DD} - V_{T,n1} - V_{T,n1}$. Therefore, each stage causes a significant drop in the voltage level.

4.2.2 Logic "0" Transfer

Let us assume that the soft-node voltage is equal to 1 initially, i.e., V_x (t = 0) = V_{max} = (V_{DD} – $V_{T,n}$). Now a logic "0" is applied to the input at time t = 0 i.e., V_{in} = 0V and the clock signal CK at the gate of transistor MP becomes high at t = 0. Then, the pass transistor MP starts conducting. Now, the node V_x acts as the drain and V_{in} acts as the source. The equivalent circuit for logic '0' transfer is shown in Fig. 4.2.6.

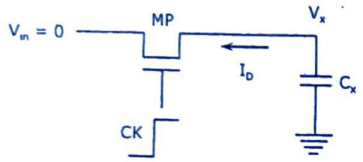


Fig. 4.2.6 Equivalent Circuit for the Logic "0" Transfer Event

In this case, we have $V_{GS} = V_{DD}$ and $V_{DS} = V_{max}$. Since, $V_{DS} < V_{GS} - V_{T,n1}$ the pass transistor MP will be operating in the linear region. The pass transistor MP now discharge the capacitor C_x as,

$$-C_x \frac{dV_x}{dt} = \frac{k_n}{2}(2(V_{DD} - V_{T,n})V_x - V_x^2) \quad \ldots (4.2.6)$$

$$\Rightarrow \quad dt = -\frac{2C_x}{k_n} \cdot \frac{dV_x}{2(V_{DD} - V_{T,n})V_x - V_x^2} \quad \ldots (4.2.7)$$

Integrating Eq. (4.2.7) with respect to t on both sides, we get,

$$\int_0^t dt = -\frac{2C_x}{k_n} \int_{V_{DD}-V_{T,n}}^{V_x} \left(\frac{\frac{1}{2(V_{DD} - V_{T,n})}}{2(V_{DD} - V_{T,n}) - V_x} + \frac{\frac{1}{2(V_{DD} - V_{T,n})}}{V_x} \right) dV_x \quad \ldots (4.2.8)$$

$$t = \frac{C_x}{k_n(V_{DD} - V_{T,n})} \ln\left(\frac{2(V_{DD} - V_{T,n}) - V_x}{V_x} \right)\Bigg|_{V_{DD} - V_{T,n}}^{V_x} \quad \ldots (4.2.9)$$

Finally, the fall-time expression for the node voltage V_x can be obtained as,

$$t = \frac{C_x}{k_n(V_{DD} - V_{T,n})} \ln\left(\frac{2(V_{DD} - V_{T,n}) - V_x}{V_x} \right) \quad \ldots (4.2.10)$$

Variation of node voltage V_x with respect to Eq. (4.2.10) is plotted as function of time is shown in Fig. 4.2.7.

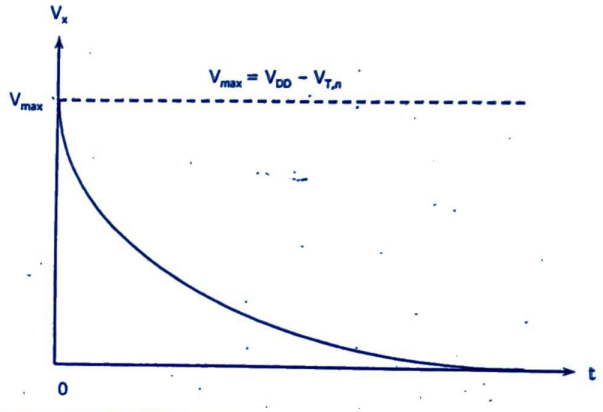


Fig. 4.2.7 Variation of V_x as a Function of Time During Logic "0"

From Fig. 4.2.7, it can be seen that the voltage drops from its logic high level of V_{max} to 0V. The fall time for the soft-node voltage V_x can be calculated from Eq. (4.2.10) as,

$$t_{90\%} = \frac{C_x}{k_n(V_{DD} - V_{T,n})} \ln\left(\frac{(2 - 0.9)(V_{DD} - V_{T,n})}{0.9(V_{DD} - V_{T,n})} \right)$$

$$= \frac{C_x}{k_n(V_{DD} - V_{T,n})} \ln\left(\frac{1.1}{0.9} \right) \quad \ldots (4.2.11)$$

$$t_{10\%} = \frac{C_x}{k_n(V_{DD} - V_{T,n})} \ln\left(\frac{1.9}{0.1} \right) \quad \ldots (4.2.12)$$

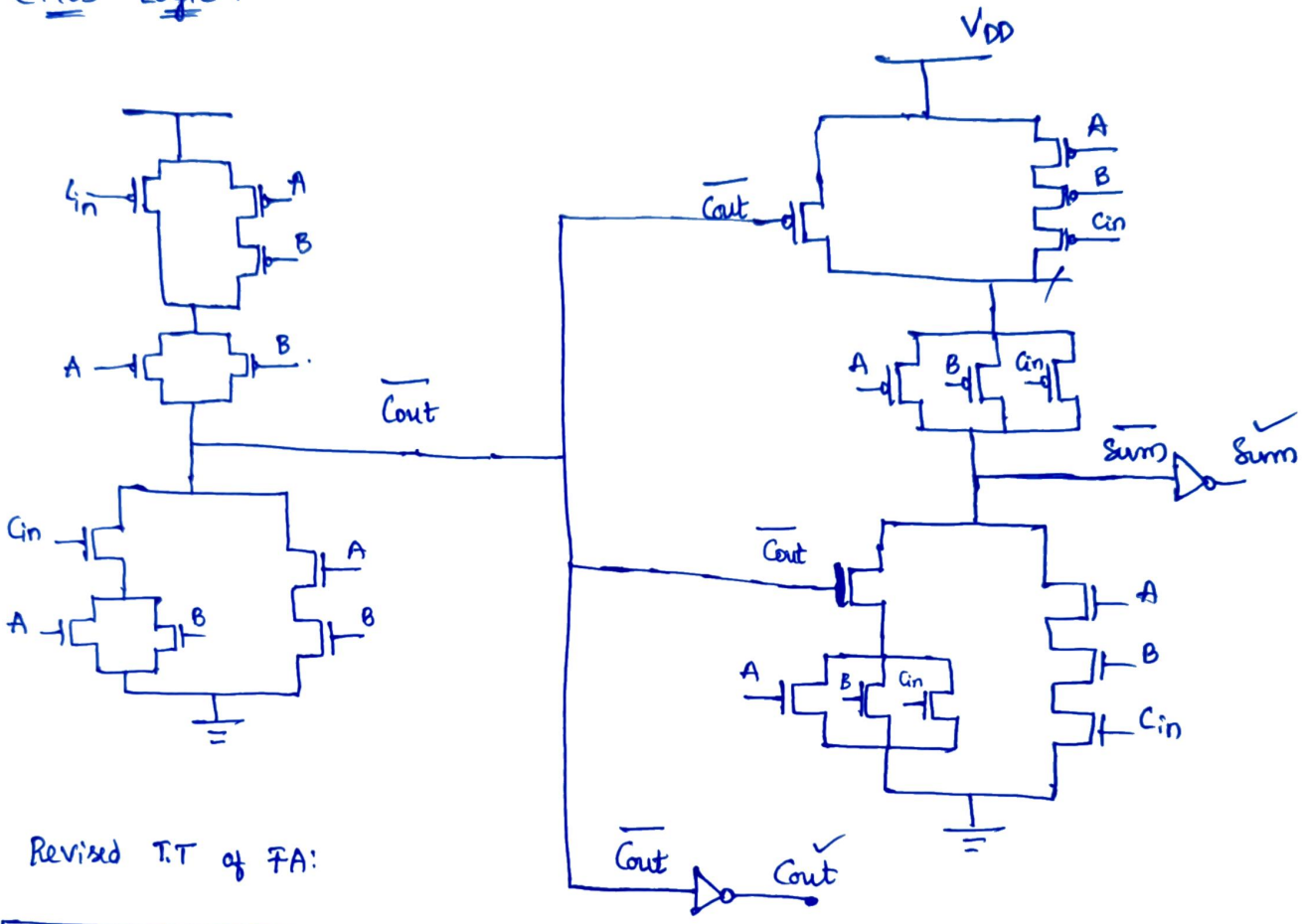
By definition, the fall of time (τ_{fall}) the soft-node voltage V_x is the difference between $t_{10\%}$ and $t_{90\%}$, which is found as,

$$\tau_{fall} = t_{10\%} - t_{90\%}$$

$$= \frac{C_x}{k_n(V_{DD} - V_{T,n})} [\ln(19) - \ln(1.22)]$$

$$\tau_{fall} = 2.74 \frac{C_x}{k_n(V_{DD} - V_{T,n})} \quad \ldots (4.2.13)$$

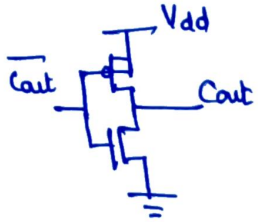
CMOS Logic :



Revised T.T of FA:

A	B	Cin	$\overline{Cout}$	Sum
0	0	0	1	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	0	1

Inverter



$Sum = C_{in}\overline{Cout} + B\overline{Cout} + A\overline{Cout} + ABC_{in}$

$Sum = \overline{Cout}(A + B + C_{in}) + ABC_{in}$

Design of a CMOS Full Adder :-

$2^n = 2^3 = 8$ bit

Truth Table

A	B	Cin	Sum	Carry
0	0	0	0	0
0	0	1	1 $\bar{A}\bar{B}C_{in}$	0
0	1	0	1 $\bar{A}B\bar{C}_{in}$	0
0	1	1	0	1 = $\bar{A}BC_{in}$
1	0	0	1 $A\bar{B}\bar{C}_{in}$	0
1	0	1	0	1 = $A\bar{B}C_{in}$
1	1	0	0	1 = $AB\bar{C}_{in}$
1	1	1	1 ABC_{in}	1 = ABC_{in}

(1) $Sum = \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + A.B.C_{in}$

$= \bar{A}(\bar{B}C_{in} + B\bar{C}_{in}) + A(\bar{B}\bar{C}_{in} + BC_{in})$

$= \bar{A}(B \oplus C_{in}) + A(B \odot C_{in})$

$= \bar{A}(B \oplus C_{in}) + A\overline{(B \oplus C_{in})}$

{ Ex-OR = $\bar{A}B + A\bar{B}$ = $A \oplus B$

Ex-NOR = $\bar{A}\bar{B} + AB$ }

* If $B \oplus C_{in} = D$

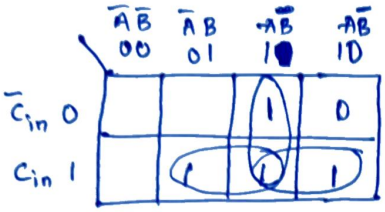
$= \bar{A}.D + A.\bar{D}$

$= A \oplus D$

$Sum = A \oplus B \oplus C_{in}$

② Carry $C_{out} = \bar{A}BC_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in}$

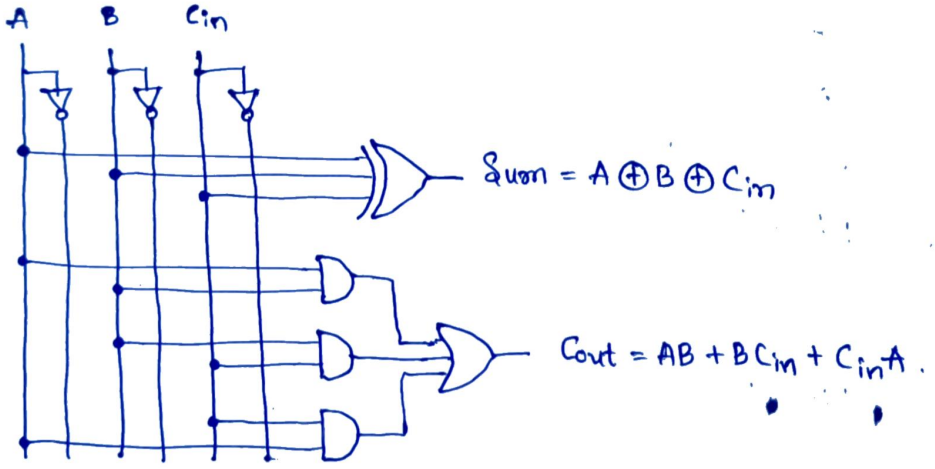
using K-map.



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

$$\boxed{C_{out} = AB + C_{in}(A+B)}$$

Gate level Design of full Adder :



CMOS Logic :

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

$C_{out} = C_{in}(A+B) + AB$ ✓

Basic Circuit