Logic Design

Dinesh Sharma
Microelectronics group
EE Department, IIT Bombay

Contents

List of Figures

Chapter 1

Transistor Models

In this booklet, we shall use simple analytical models for MOS transistors. We use a sign convention according to which, voltage and current symbols associated with the pMOS transistor (such as V_{Tp}) have positive values. Then, the n channel formulae can be used for both transistors and we shall assign signs to quantities explicitly.

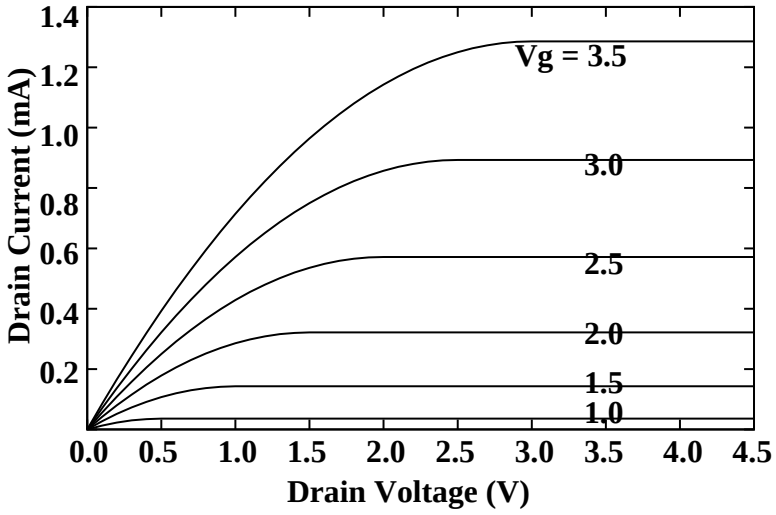


Figure 1.1: MOS characteristics according to the simple analytic model

The model we use is described by the following equations:
for $V_{gs} \leq V_T$,

$$I_{ds} = 0 \tag{1.1}$$

for $V_{gs} > V_T$ and $V_{ds} \leq V_{gs} - V_T$,

$$I_{ds} = K\left[(V_{gs} - V_T)V_{ds} - \frac{1}{2}V_{ds}^2\right] \tag{1.2}$$

and for $V_{gs} > V_T$ and $V_{ds} > V_{gs} - V_T$,

$$I_{ds} = K\frac{(V_{gs} - V_T)^2}{2} \tag{1.3}$$

The saturation region equation is somewhat oversimplified because it assumes that the current is independent of V_{ds}. In reality, the current has a weak dependence on V_{ds} in this region.

In order to model the saturation region more accurately, we adopt an "Early Voltage" like formalism.

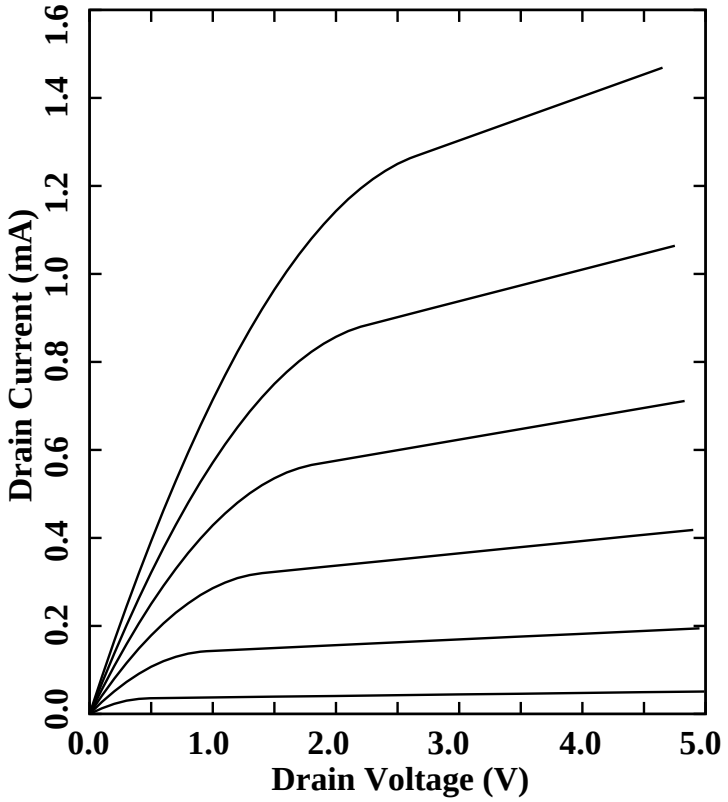


Figure 1.2: MOS characteristics with non zero conductance in saturation

It is assumed that the current increases linearly in the saturation region. All linear

characteristics in saturation can be produced backwards towards negative drain voltages and will intersect the drain voltage axis at a single point at -V_E. (This is, at best, an approximation). Because the conductance in saturation is now non zero, the onset of saturation has to be redefined, so that the current and its derivative are continuous at the boundary of linear and saturation regimes. The current equations are given by:
For $V_{gs} > V_T$ and $V_{ds} \leq V_{dss}$,

$$I_{ds} = K\left[(V_{gs} - V_T)V_{ds} - \frac{1}{2}V_{ds}^2\right] \tag{1.4}$$

and for $V_{gs} > V_T$ and $V_{ds} > V_{dss}$,

$$I_{ds} = I_{dss}\frac{V_d + V_E}{V_{dss} + V_E} \tag{1.5}$$

Where V_E is the 'Early Voltage'. Here V_{dss} and I_{dss} are saturation drain voltage and drain current respectively. Since the current values must match at either side of $V_{ds} = V_{dss}$, we must have:

$$I_{dss} \equiv K\left[(V_{gs} - V_T)V_{dss} - \frac{1}{2}V_{dss}^2\right]. \tag{1.6}$$

For the curve to be smooth and continuous at $V_d = V_{dss}$, the value of the first derivative should match on either side of V_{dss}. Therefore,

$$K(V_{gs} - V_T - V_{dss}) = \frac{I_{dss}}{V_{dss} + V_E}$$

So,

$$K(V_{gs} - V_T - V_{dss})(V_{dss} + V_E) = K\left[(V_{gs} - V_T)V_{dss} - \frac{1}{2}V_{dss}^2\right] \tag{1.7}$$

This leads to a quadratic equation in V_{dss}

$$\frac{1}{2}V_{dss}^2 + V_E V_{dss} - (V_{gs} - V_T)V_E = 0 \tag{1.8}$$

Solving this quadratic, we get

$$V_{dss} = V_E\left(\sqrt{1 + \frac{2(V_{gs} - V_T)}{V_E}} - 1\right) \tag{1.9}$$

For $V_E >> V_{gs} - V_T$ this reduces to

$$V_{dss} \simeq (V_{gs} - V_T)\left(1 - \frac{V_{gs} - V_T}{2V_E}\right) \tag{1.10}$$

Characteristics of a MOS transistor using this model are shown in fig.1.2. While accurate modeling of the output conductance is essential for linear design, the simpler model assuming constant I_d in saturation is often adequate for preliminary digital design. In any case, final designs will have to be validated with detailed simulations. In this booklet, we shall use the simple model for MOS devices to keep the algebra simple.

Chapter 2

Static CMOS Logic Design

Static logic circuits are those which can hold their output logic levels for indefinite periods as long as the inputs are unchanged. Circuits which depend on charge storage on capacitors are called dynamic circuits and will be discussed in a later chapter.

2.1 Static CMOS Design style

The most common design style in modern VLSI design is the Static CMOS logic style. In this, each logic stage contains pull up and pull down networks which are controlled by input signals. The pull up network contains p channel transistors, whereas the pull down network is made of n channel transistors. The networks are so designed that the pull up and pull down networks are never 'on' simultaneously. This ensures that there is no static power consumption.

2.2 CMOS Inverter

The simplest of such logic structures is the CMOS inverter. In fact, for any CMOS logic design, the CMOS inverter is the basic gate which is first analyzed and designed in detail. Thumb rules are then used to convert this design to other more complex logic. The basic CMOS inverter is shown in fig. 2.1. We shall develop the characteristics of CMOS logic through the inverter structure, and later discuss ways of converting this basic structure more complex logic gates.

2.2.1 Static Characteristics

The range of input voltages can be divided into several regions.

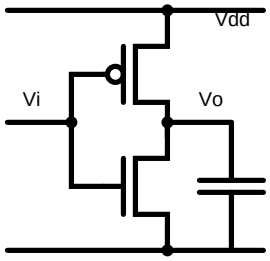


Figure 2.1: The basic CMOS inverter

nMOS 'off', pMOS 'on'

For $0 < V_i < V_{Tn}$ the n channel transistor is 'off', the p channel transistor is 'on' and the output voltage = V_{dd}. This is the normal digital operation range with input = '0' and output = '1'.

nMOS saturated, pMOS linear

In this regime, both transistors are 'on'. The input voltage V_i is $> V_{Tn}$, but is small enough so that the n channel transistor is in saturation, and the p channel transistor is in the linear regime. In static condition, the output voltage will adjust itself such that the currents through the n and p channel transistors are equal. The absolute value of gate-source voltage on the p channel transistor is V_{dd}- V_i, and therefore the "over voltage" on its gate is V_{dd}- V_i- V_{Tp}. The drain source voltage of the pMOS has an absolute value V_{dd}-V_o. Therefore,

$$I_d = K_p \left[(V_{dd} - V_i - V_{Tp})(V_{dd} - V_o) - \frac{1}{2}(V_{dd} - V_o)^2 \right] = \frac{K_n}{2}(V_i - V_{Tn})^2 \qquad (2.1)$$

Where symbols have their usual meanings.

We define $\beta \equiv \mathrm{K_n}/\mathrm{K_p}$. We make the substitution $V_{dp} \equiv V_{dd} - V_o$, where $V_{dp}$is the absolute value of the drain-source voltage for the p channel transistor. Then,

$$(V_{dd} - V_i - V_{Tp})V_{dp} - \frac{1}{2}V_{dp}^2 = \frac{\beta}{2}(V_i - V_{Tn})^2 \qquad (2.2)$$

Which gives the quadratic

$$\frac{1}{2}V_{dp}^2 - V_{dp}(V_{dd} - V_i - V_{Tp}) + \frac{\beta}{2}(V_i - V_{Tn})^2 = 0 \qquad (2.3)$$

Solutions to the quadratic are:

$$V_{dp} = (V_{dd} - V_i - V_{Tp}) \pm \sqrt{(V_{dd} - V_i - V_{Tp})^2 - \beta(V_i - V_{Tn})^2} \qquad (2.4)$$

These equations are valid only when the pMOS is in its linear regime. This requires that

$$V_{dp} \equiv V_{dd} - V_o \leq V_{dd} - V_i - V_{Tp}$$

Therefore, we must choose the negative sign. Thus

$$V_{dd} - V_o = (V_{dd} - V_i - V_{Tp}) - \sqrt{V_{dd} - V_i - V_{Tp})^2 - \beta(V_i - V_{Tn})^2} \tag{2.5}$$

Therefore,

$$V_o = V_i + V_{Tp} + \sqrt{(V_{dd} - V_i - V_{Tp})^2 - \beta(V_i - V_{Tn})^2} \tag{2.6}$$

Since V_o must be $\geq V_i + V_{Tp}$, the limit of applicability of the above result is given by

$$(V_{dd} - V_i - V_{Tp})^2 = \beta(V_i - V_{Tn})^2$$

That is, the solution for V_o is valid for

$$V_i \leq \frac{V_{dd} + \sqrt{\beta}V_{Tn} - V_{Tp}}{1 + \sqrt{\beta}} \tag{2.7}$$

In the case where we size the n and p channel transistors such that

$$K_n = K_p; \text{ so } \beta = 1$$

we have

$$V_o = (V_i + V_{Tp}) + \sqrt{(V_{dd} - V_{Tn} - V_{Tp})(V_{dd} - 2V_i + V_{Tn} - V_{Tp})} \tag{2.8}$$

with

$$V_i \leq \frac{V_{dd} + V_{Tn} - V_{Tp}}{2}$$

nMOS saturated, pMOS saturated

At the limit of applicability of eq. 2.7, when the input voltage is exactly at

$$V_i = \frac{V_{dd} + \sqrt{\beta}V_{Tn} - V_{Tp}}{1 + \sqrt{\beta}} \tag{2.9}$$

both transistors are saturated. Since the currents of both transistors are independent of their drain voltages in this condition, we do not get a unique solution for V_o by equating drain currents. The currents will be equal for all values of V_o in the range

$$V_i - V_{Tn} \leq V_o \leq V_i + V_{Tp}$$

Thus the transfer curve of an inverter shows a drop of $V_{Tn}+ V_{Tp}$ at a voltage near $V_{dd}/2$. This is actually an artifact of the simple transistor model chosen for this

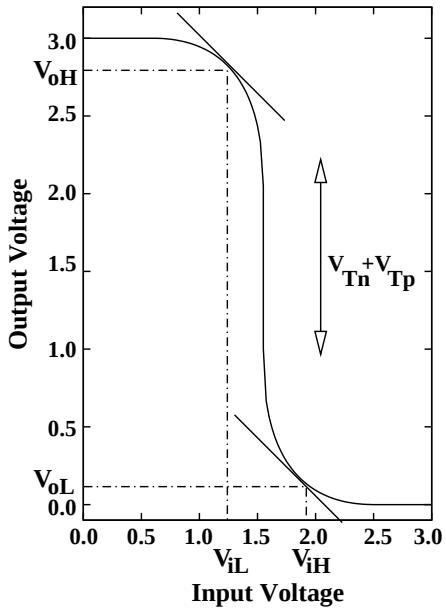


Figure 2.2: Transfer Curve of a CMOS inverter

analysis, which assumes perfect saturation of drain current. In a real case, the drain current does depend on the drain voltage (albeit weakly) in the saturation region. If the model incorporates an Early Voltage like effect, the drop near the middle of the characteristic is more gradual.

nMOS linear, pMOS saturated

At the gate voltage given by eq. 2.9, both transistors are saturated. As we increase V_i beyond this value, such that

$$\frac{V_{dd} + \sqrt{\beta} V_{Tn} - V_{Tp}}{1 + \sqrt{\beta}} < V_i < V_{dd} - V_{Tp}$$

both transistors are still 'on', but nMOS enters the linear regime while pMOS gets saturated. Equating currents in this condition,

$$I_d = \frac{K_p}{2}(V_{dd} - V_i - V_{Tp})^2 = K_n \left[(V_i - V_{Tn}) V_o - \frac{1}{2} V_o^2 \right] \tag{2.10}$$

From this, we get the quadratic equation

$$\frac{1}{2} V_o^2 - (V_i - V_{Tn}) V_o + \frac{(V_{dd} - V_i - V_{Tp})^2}{2\beta} = 0 \tag{2.11}$$

This has solutions

$$V_o = (V_i - V_{Tn}) \pm \sqrt{(V_i - V_{Tn})^2 - \frac{(V_{dd} - V_i - V_{Tp})^2}{\beta}} \tag{2.12}$$

Since the equations are valid only when the n channel transistor is in the linear regime ($V_o < V_i - V_{Tn}$), we choose the negative sign. This gives,

$$V_o = (V_i - V_{Tn}) - \sqrt{(V_i - V_{Tn})^2 - \frac{(V_{dd} - V_i - V_{Tp})^2}{\beta}} \tag{2.13}$$

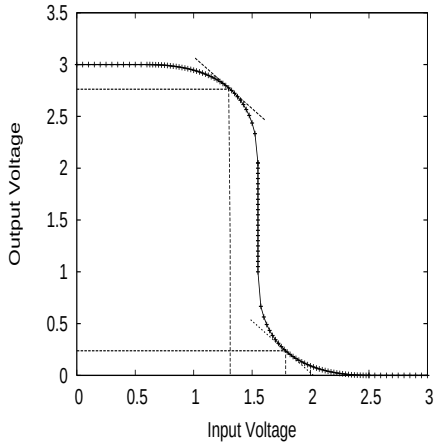
Again, in the special case where $\beta = 1$, we have

$$V_o = (V_i - V_{Tn}) - \sqrt{(V_{dd} - V_{Tn} - V_{Tp})(2V_i - V_{dd} - V_{Tn} + V_{Tp})} \tag{2.14}$$

nMOS 'on', pMOS 'off'

As we increase the input voltage beyond V_{dd}- V_{Tp}, the p channel transistor turns 'off', while the n channel conducts strongly. As a result, the output voltage falls to zero. This is the normal digital operation range with input = '1' and output = '0'.

The figure below shows the transfer curve of an inverter with V_{dd}= 3V, V_{Tn}= 0.6V and V_{Tp}= 0.5V, and $\beta = 1$.



The plot produced by SPICE for this circuit with realistic models is quite similar.

2.2.2 Noise margins

The requirement from a digital circuit is that it should distinguish logic levels, but be insensitive to the exact analog voltage at the input. This implies that

the flat portions of the transfer curve (where $\frac{\partial V_o}{\partial V_i}$ is small) are suitable for digital logic. We select two points on the transfer curve where the slope ($\frac{\partial V_o}{\partial V_i}$) is -1.0. The coordinates of these two points define the values of (V_{iL},V_{oH}) and (V_{iH},V_{oL}). Robust digital design requires that the output high level be higher than what is acceptable as a high level at the input ($V_{oH} > V_{iH}$). The difference between these two levels is the 'high' noise margin. This is the amount of noise that can ride on the worst case 'high' output and still be accepted as a 'high' at the input of the next gate. Similarly, we require $V_{oL} < V_{iL}$. The difference, $V_{iL} - V_{oL}$ is the 'low' noise margin. Obviously, it is of interest to evaluate the values of these noise margins. For the discussion which follows, we shall use the expressions derived earlier for $\beta = 1$ to keep the algebra simple.

Calculation of V_{iL} and V_{oH}

from eq. (2.8)

$$V_o = (V_i + V_{Tp}) + \sqrt{(V_{dd} - V_{Tn} - V_{Tp})(V_{dd} + V_{Tn} - V_{Tp} - 2V_i)}$$

From this, we can evaluate $\frac{\partial V_o}{\partial V_i}$ and set it = -1.

$$\frac{\partial V_o}{\partial V_i} = -1 = 1 - \sqrt{\frac{V_{dd} - V_{Tn} - V_{Tp}}{V_{dd} + V_{Tn} - V_{Tp} - 2V_i}} \tag{2.15}$$

This gives

$$V_{iL} = \frac{3V_{dd} + 5V_{Tn} - 3V_{Tp}}{8} \tag{2.16}$$

Substituting this in eq.(2.8), we get

$$V_{oH} = \frac{7V_{dd} + V_{Tn} + V_{Tp}}{8} = V_{dd} - \frac{V_{dd} - V_{Tn} - V_{Tp}}{8} \tag{2.17}$$

Calculation of V_{iH} and V_{oL}

When the input is 'high', we should use eq.(2.14).

$$V_o = (V_i - V_{Tn}) - \sqrt{(V_{dd} - V_{Tn} - V_{Tp})(2V_i - V_{dd} - V_{Tn} + V_{Tp})}$$

Differentiating with respect to V_i gives

$$\frac{\partial V_o}{\partial V_i} = -1 = 1 - \sqrt{\frac{V_{dd} - V_{Tn} - V_{Tp}}{2V_i - V_{dd} - V_{Tn} + V_{Tp}}} \tag{2.18}$$

From where, we get

$$V_{iH} = \frac{5V_{dd} + 3V_{Tn} - 5V_{Tp}}{8} \tag{2.19}$$

and

$$V_{oL} = \frac{V_{dd} - V_{Tn} - V_{Tp}}{8} \tag{2.20}$$

Calculation of Noise Margins

The high noise margin is given by

$$V_{oH} - V_{iH} = \frac{V_{dd} - V_{Tn} + 3V_{Tp}}{4} \tag{2.21}$$

Similarly, the Low noise margin is

$$V_{iL} - V_{oL} = \frac{V_{dd} + 3V_{Tn} - V_{Tp}}{4} \tag{2.22}$$

The two noise margins can be made equal by choosing equal values for V_{Tn} and V_{Tp}.

2.2.3 Dynamic Considerations

In this section, we analyze the dynamic behaviour of the inverter. For the calculation of rise and fall times, we shall assume that only one of the two transistors in the inverter is 'on'. (Notice that this is more conservative than the input high and low conditions determined by slope considerations in eq.2.19 and 2.16). We shall continue to use the simple model described at the beginning of this booklet.

Rise time

When the input is low, the n channel transistor is 'off', while the p channel transistor is 'on'. The equivalent circuit in this condition is shown in fig. 2.3. From

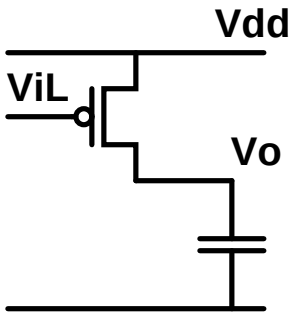


Figure 2.3: CMOS inverter with the nMOS 'off'

Kirchoff's current law at the output node,

$$I_{dp} = C\frac{dV_o}{dt}$$

so,

$$\frac{dt}{C} = \frac{dV_o}{I_{dp}}$$

This separates the variables, with the LHS independent of operating voltages and the RHS independent of time. Integrating both sides, we get

$$\frac{\tau_{rise}}{C} = \int_0^{V_{oH}} \frac{dV_o}{I_{dp}}$$

Till the output rises to $V_{iL}+ V_{Tp}$, the p channel transistor is in saturation. Since the current is constant, the integration is trivial. If $V_{oH} > V_{iL} + V_{Tp}$ (which is normally the case), the integration range can be broken into saturation and linear regimes. Thus

$$\begin{aligned}\frac{\tau_{rise}}{C} &= \int_0^{V_{iL}+V_{Tp}} \frac{dV_o}{\frac{K_p}{2}(V_{dd} - V_{iL} - V_{Tp})^2} \\ &\quad + \int_{V_{iL}+V_{Tp}}^{V_{oH}} \frac{dV_o}{K_p\left[(V_{dd} - V_{iL} - V_{Tp})(V_{dd} - V_o) - \frac{1}{2}(V_{dd} - V_o)^2\right]}\end{aligned}$$

We define $V_1 \equiv V_{dd} - V_o$ and $V_2 \equiv V_{dd} - V_{iL} - V_{Tp}$, so $dV_o = -dV_1$.
We get

$$\frac{K_p\tau_{rise}}{2C} = \frac{V_{iL} + V_{Tp}}{V_2^2} - \int_{V_2}^{V_{dd}-V_{oH}} \frac{dV_1}{2V_1V_2 - V_1^2}$$

The integral can be evaluated as

$$\begin{aligned} I &\equiv -\int_{V_2}^{V_{dd}-V_{oH}} \frac{dV_1}{2V_1V_2 - V_1^2} \\ &= \frac{1}{2V_2}\int_{V_{dd}-V_{oH}}^{V_2} \left(\frac{1}{V_1} + \frac{1}{2V_2 - V_1}\right) dV_1 \\ &= \frac{1}{2V_2}\left[\ln\frac{V_1}{2V_2 - V_1}\right]_{V_{dd}-V_{oH}}^{V_2} \\ &= \frac{1}{2V_2}\ln\frac{2V_2 - V_{dd} + V_{oH}}{V_{dd} - V_{oH}} \end{aligned}$$

Therefore,

$$\frac{K_p\tau_{rise}}{2C} = \frac{V_{iL} + V_{Tp}}{V_2^2} + \frac{1}{2V_2}\ln\frac{2V_2 - V_{dd} + V_{oH}}{V_{dd} - V_{oH}}$$

or

$$\frac{K_p \tau_{rise}}{2C} = \frac{V_{iL} + V_{Tp}}{(V_{dd} - V_{iL} - V_{Tp})^2} + \frac{1}{2(V_{dd} - V_{iL} - V_{Tp})} \ln \frac{2V_2 - V_{dd} + V_{oH}}{V_{dd} - V_{oH}}$$

Thus,

$$\begin{aligned} \tau_{rise} &= \frac{C(V_{iL} + V_{Tp})}{\frac{K_p}{2}(V_{dd} - V_{iL} - V_{Tp})^2} \\ &+ \frac{C}{K_p(V_{dd} - V_{iL} - V_{Tp})} \ln \frac{V_{dd} + V_{oH} - 2V_{iL} - 2V_{Tp}}{V_{dd} - V_{oH}} \end{aligned} \tag{2.23}$$

The first term is just the constant current charging of the load capacitor. The second term represents the charging by the pMOS in its linear range. This can be compared with resistive charging, which would have taken a charge time of

$$\tau = RC \ln \frac{V_{dd} - V_{iL} - V_{Tp}}{V_{dd} - V_{oH}}$$

to charge from $V_{iL} + V_{Tp}$ to V_{oH}.

Fall time

When the input is high, the n channel transistor is 'on' and the p channel transistor is 'off'. If the output was initially 'high', it will be discharged to ground through

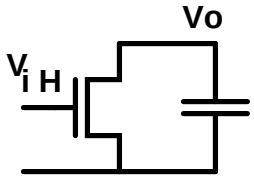


Figure 2.4: CMOS inverter with the pMOS 'off'

the nMOS. To analysis the fall time, we apply Kirchoff's current law to the output node. This gives

$$I_{dn} = -C \frac{dV_o}{dt}$$

Again, separating variables and integrating from the initial voltage ($= V_{dd}$) to some terminal voltage V_{oL} gives

$$\frac{\tau_{fall}}{C} = -\int_{V_{dd}}^{voL} \frac{dV_o}{I_{dn}}$$

The n channel transistor will be in saturation till the output voltage falls to V_i- V_{Tn}. Below this voltage, the transistor will be in its linear regime. Thus, we can divide the integration range in two parts.

$$\begin{aligned}\frac{\tau_{fall}}{C} &= -\int_{V_{dd}}^{V_i-V_{Tn}} \frac{dV_o}{I_{dn}} - \int_{V_i-V_{Tn}}^{V_{oL}} \frac{dV_o}{I_{dn}} \\ &= \int_{V_i-V_{Tn}}^{V_{dd}} \frac{dV_o}{\frac{K_n}{2}(V_i-V_{Tn})^2} \\ &\quad + \int_{V_{oL}}^{V_i-V_{Tn}} \frac{dV_o}{K_n[(V_i-V_{Tn})V_o - \frac{1}{2}V_o^2}\end{aligned}$$

Therefore

$$\begin{aligned}\frac{K_n\tau_{fall}}{2C} &= \frac{V_{dd}-V_i+V_{Tn}}{(V_i-V_{Tn})^2} + \int_{V_{oL}}^{V_i-V_{Tn}} \frac{dV_o}{2V_o(V_i-V_{Tn})-V_o^2} \\ &= \frac{V_{dd}-V_i+V_{Tn}}{(V_i-V_{Tn})^2} + \frac{1}{2(V_i-V_{Tn})}\int_{V_{oL}}^{V_i-V_{Tn}} dV_o\left(\frac{1}{V_o}+\frac{1}{2(V_i-V_{Tn})-V_o}\right)\end{aligned}$$

Which gives

$$\begin{aligned}\frac{K_n\tau_{fall}}{2C} &= \frac{V_{dd}-V_i+V_{Tn}}{(V_i-V_{Tn})^2} + \frac{1}{2(V_i-V_{Tn})}\left[\ln\frac{V_o}{2(V_i-V_{Tn})-V_o}\right]_{V_{oL}}^{V_i-V_{Tn}} \\ &= \frac{V_{dd}-V_i+V_{Tn}}{(V_i-V_{Tn})^2} + \frac{1}{2(V_i-V_{Tn})}\ln\frac{2(V_i-V_{Tn})-V_{oL}}{V_{oL}}\end{aligned}$$

and therefore

$$\tau_{fall} = \frac{C(V_{dd}-V_i+V_{Tn})}{\frac{K_n}{2}(V_i-V_{Tn})^2} + \frac{C}{K_n(V_i-V_{Tn})}\ln\frac{2(V_i-V_{Tn})-V_{oL}}{V_{oL}} \tag{2.24}$$

Again, the first term represents the time taken to discharge at constant current in the saturation regime, whereas the second term is the quasi-resistive discharge in the linear regime.

2.2.4 Trade off between power, speed and robustness

As we scale technologies, we improve speed and power consumption. However, as we can see from the expression for noise margins, (eq 2.21 and eq 2.22) the noise margin becomes worse. We can improve noise margins by choosing relatively higher threshold voltages. However, this will reduce speeds. We could also increase V_{dd}- but that would increase power dissipation. Thus we have a trade off between power, speed and noise margins.

This choice is made much more complicated by process variations, because we have to design for the worst case.

2.2.5 CMOS Inverter Design Flow

The CMOS inverter forms the basis of most static CMOS logic design. More complex logic can be designed from it by simple thumb rules. A common (though not universal) design requirement is symmetric charge and discharge behaviour and equal noise margins for high and low logic values. This requires matched values of K_n and K_p and equal values of $V_{Tn}$and V_{Tp}. For a constant load capacitance, rise and fall times depend linearly on K_n and K_p. Thus it is a straightforward calculation to determine transistor geometries if speed requirements and technological parameters are given. However, as transistor geometries are made larger, self loading can become significant. We now have to model the load capacitance as

$$C_{Load} = C_{ext} + \alpha K_n$$

where we have assumed that $\beta = K_n/K_p$ is kept constant. α is a technological constant. We use the expressions for $K\tau/C$ which depend only on voltages. Once these values are calculated, the geometry can be determined.

In the extreme case, when self capacitance dominates the load capacitance, K/C becomes constant and τ becomes geometry independent. There is no advantage in using wider transistors in this regime to increase the speed. It is better to use multi-stage logic with tapered buffers in this regime. This will be discussed in the module on Logical Effort.

2.2.6 Conversion of CMOS Inverters to other logic

Once the basic CMOS inverter is designed, other logic gates can be derived from it. The logic has to be put in a canonical form which is a sum of products with a bar (inversion) on top. For every '.' in the expression, we put the corresponding n channel transistors in series and the corresponding p channel transistors in parallel. for every '+', we put the n channel transistors in parallel and the p channel transistors in series. We scale the transistor widths up by the number of devices (n or p) put in series. The geometries are left untouched for devices put in parallel. Fig.2.5 shows the implementation of $\overline{A.B + C.(D + E)}$ in CMOS logic design style.

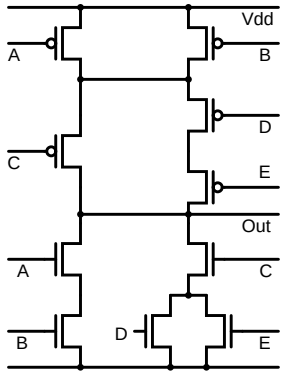


Figure 2.5: CMOS implementation of $\overline{A.B + C.(D + E)}$

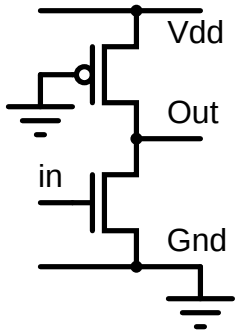
Chapter 3

Beyond Static CMOS

3.1 Pseudo nMOS Design Style

CMOS design style ensures that the logic consumes no static power. This is because the pull down and pull up networks are never 'on' simultaneously. However, this requires that signals have to be routed to the n pull down network *as well as* to the p pull up network. This means that the load presented to every driver is high. This fact is exacerbated by the fact that n and p channel transistors cannot be placed close together as these are in different wells which have to be kept well separated in order to avoid latchup.

Pseudo nMOS design style reduces dynamic power (by reducing capacitive loading) at the cost of having non-zero static power by replacing the pull up network by a single pMOS transistor with its gate terminal grounded. The pseudo nMOS inverter is shown below.



Notice that since the pMOS is not driven by signals, it is always 'on'. The effective gate voltage seen by the pMOS transistor is V_{dd}. Thus the overvoltage on the p channel gate is always V_{dd}- V_{Tp}. When the nMOS is turned 'on', a direct path between supply and ground exists and static power will be drawn.

3.1.1 Static Characteristics

As we sweep the input voltage from ground to V_{dd}, we encounter the following regimes of operation:

nMOS 'off'

This is the case when the input voltage is less than V_{Tn}. The output is 'high' and no current is drawn from the supply.

nMOS saturated, pMOS linear

As the input voltage is raised above V_{Tn}, we enter this region. The input voltage is assumed to be sufficiently low that the output voltage exceeds the saturation voltage $V_i - V_{Tn}$. Normally, this voltage will be higher than V_{Tp}, so the p channel transistor is in linear mode of operation. Equating currents through the n and p channel transistors, we get

$$K_p \left[(V_{dd} - V_{Tp})(V_{dd} - V_o) - \frac{1}{2}(V_{dd} - V_o)^2 \right] = \frac{K_n}{2}(V_i - V_{Tn})^2 \tag{3.1}$$

defining $V_1 \equiv V_{dd} - V_o$ and $V_2 \equiv V_{dd} - V_{Tp}$, we get

$$\frac{1}{2}V_1^2 - V_2 V_1 + \frac{\beta}{2}(V_i - V_{Tn})^2 = 0 \tag{3.2}$$

with solutions

$$V_1 = V_2 \pm \sqrt{V_2^2 - \beta(V_i - V_{Tn})^2}$$

substituting the values of V_1 and V_2 and choosing the sign which puts V_o in the correct range, we get

$$V_o = V_{Tp} + \sqrt{(V_{dd} - V_{Tp})^2 - \beta(V_i - V_{Tn})^2} \tag{3.3}$$

nMOS linear, pMOS linear

As the input voltage is increased, the output voltage will decrease in accordance with equation(3.3). At some point, the output voltage will fall below $V_i - V_{Tn}$. It can be shown that this will happen when

$$V_i > V_{Tn} + \frac{V_{Tp} + \sqrt{V_{Tp}^2 + (\beta + 1)V_{dd}(V_{dd} - 2V_{Tp})}}{\beta + 1}.$$

The nMOS is now in its linear mode of operation. We shall not derive the expression for the output voltage in this mode of operation in the discussion here. The solution is straightforward, though algebraically tedious.

nMOS linear, pMOS saturated

As the input voltage is raised still further, the output voltage will fall below V_{Tp}. The pMOS transistor is now in saturation regime. Equating currents, we get

$$K_n \left[(V_i - V_{Tn})V_o - \frac{1}{2}V_o^2 \right] = \frac{K_p}{2}(V_{dd} - V_{Tp})^2$$

which gives

$$\frac{1}{2}V_o^2 - (V_o - V_{Tn})V_o + \frac{(V_{dd} - V_{Tp})^2}{2\beta}$$

This can be solved to get

$$V_o = (V_i - V_{Tn}) - \sqrt{(V_i - V_{Tn})^2 - (V_{dd} - V_{Tp})^2/\beta} \tag{3.4}$$

3.1.2 Noise margins

As in the case of CMOS inverter, we find points on the transfer curve where the slope is -1.

When the input is low and output high, we should use eq(3.3). Differentiating this equation with respect to V_i and setting the slope to -1, we get

$$V_{iL} = V_{Tn} + \frac{V_{dd} - V_{Tp}}{\sqrt{\beta(\beta + 1)}} \tag{3.5}$$

and

$$V_{oH} = V_{Tp} + \sqrt{\frac{\beta}{\beta + 1}}\ (V_{dd} - V_{Tp}) \tag{3.6}$$

When the input is high and the output low, we use eq(3.4). Again, differentiating with respect to V_i and setting the slope to -1, we get

$$V_{iH} = V_{Tn} + \frac{2}{\sqrt{3\beta}}\ (V_{dd} - V_{Tp}) \tag{3.7}$$

and

$$V_{oL} = \frac{(V_{dd} - V_{Tp})}{\sqrt{3\beta}} \tag{3.8}$$

To make the output 'low' value lower than V_{Tn}, we get the condition

$$\beta > \frac{1}{3}\left(\frac{V_{dd} - V_{Tp}}{V_{Tn}}\right)^2$$

This condition on values of β places a requirement on the ratios of widths of n and p channel transistors. The logic gates work properly only when this equation is satisfied. Therefore this kind of logic is also called 'ratioed logic'. In contrast, CMOS logic is called ratioless logic because it does not place any restriction on the ratios of widths of n and p channel transistors for static operation. The noise margin for pseudo nMOS can be determined easily from the expressions for V_{iL}, V_{oL}, V_{iH}, V_{oH}.

3.1.3 Dynamic characteristics

In the sections above, we have derived the behaviour of a pseudo nMOS inverter in static conditions. In the sections below, we discuss the dynamic behaviour of this inverter.

Rise Time

When the input is low and the output rises from 'low' to 'high', the nMOS is off. The situation is identical to the charge up condition of a CMOS gate with the pMOS being biased with its gate at 0V. This gives

$$\tau_{rise} = \frac{C}{K_p(V_{dd} - V_{Tp})}\left[\frac{2V_{Tp}}{V_{dd} - V_{Tp}} + \ln\frac{V_{dd} + V_{oH} - 2V_{Tp}}{V_{dd} - V_{oH}}\right] \tag{3.9}$$

Fall Time

Analytical calculation of fall time is complicated by the fact that the pMOS load continues to dump current in the output node, even as the nMOS tries to discharge the output capacitor.

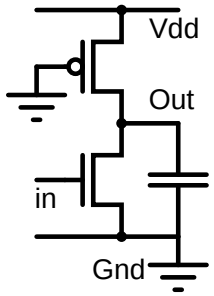


Figure 3.1: 'high' to 'low' transition on the output

Thus the nMOS should sink the discharge current as well as the drain current of the pMOS transistor. We make the simplifying assumption that the pMOS current

remains constant at its saturation value through the entire discharge process. (This will result in a slightly pessimistic value of discharge time). Then,

$$I_p = \frac{K_p}{2}(V_{dd} - V_{Tp})^2$$

. We can write the KCL equation at the output node as:

$$I_n - I_p + C\frac{dV_o}{dt} = 0$$

which gives

$$\frac{\tau_{fall}}{C} = -\int_{V_{dd}}^{V_{oL}} \frac{dV_o}{I_n - I_p}$$

We define $V_1 \equiv V_i - V_{Tn}$ and $V_2 \equiv V_{dd} - V_{Tp}$. The integration range can be divided into two regimes. nMOS is saturated when $V_1 \leq V_o < V_{dd}$ and is in linear regime when $V_{oL} < V_o < V_1$. Therefore,

$$\frac{\tau_{fall}}{C} = -\int_{V_{dd}}^{V_1} \frac{dV_o}{\frac{1}{2}K_n V_1^2 - I_p} - \int_{V_1}^{V_{oL}} \frac{dV_o}{K_n(V_1 V_o - \frac{1}{2}V_o^2) - I_p}$$

so,

$$\frac{\tau_{fall}}{C} = \frac{V_{dd} - V_1}{\frac{1}{2}K_n V_1^2 - I_p} + \int_{V_{oL}}^{V_1} \frac{dV_o}{K_n(V_1 V_o - \frac{1}{2}V_o^2) - I_p}$$

3.1.4 Pseudo nMOS design Flow

We design the basic inverter first and then map the inverter design to other logic circuits. The load device size is calculated from the rise time. From eq. 3.9 we have

$$\tau_{rise} = \frac{C}{K_p(V_{dd} - V_{Tp})}\left[\frac{2V_{Tp}}{V_{dd} - V_{Tp}} + \ln\frac{V_{dd} + V_{oH} - 2V_{Tp}}{V_{dd} - V_{oH}}\right]$$

Given a value of τ_{rise}, operating voltages and technological constants, K_p and hence, the geometry of the p channel transistor can be determined.

Geometry of the n channel transistor in the reference inverter design can be determined from static considerations. Using eq. 3.4, the output 'low' level is given by:

$$V_o = (V_i - V_{Tn}) - \sqrt{(V_i - V_{Tn})^2 - (V_{dd} - V_{Tp})^2/\beta}$$

If the desired value of the output 'low' level is given, we can calculate β. But $\beta \equiv K_n/K_p$ and K_p is already known. This evaluates K_n and hence, the geometry of the n channel transistor.

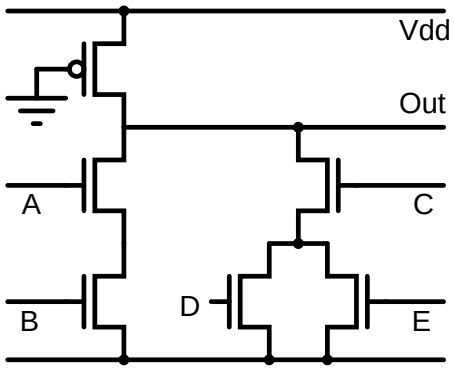


Figure 3.2: Pseudo NMOS implementation of $\overline{A.B + C.(D + E)}$

3.1.5 Conversion of pseudo nMOS Inverter to other logic

Once the basic pseudo nMOS inverter is designed, other logic gates can be derived from it. The procedure is the same as that for CMOS, except that it is applied only to nMOS transistors. The p channel transistor is kept at the same size as that for an inverter.

The logic is expressed as a sum of products with a bar (inversion) on top. For every '.' in the expression, we put the corresponding n channel transistors in series and for every '+', we put the n channel transistors in parallel. We scale the transistor widths up by the number of devices put in series. The geometries are left untouched for devices put in parallel. Fig.3.2 shows the implementation of $\overline{A.B + C.(D + E)}$ in pseudo NMOS logic design style.

3.2 Complementary Pass gate Logic

This logic family is based on multiplexer logic.

Given a boolean function F(x1, x2, ..., xn), we can express it as:

$$F(x1, x2, \ldots, xn) = xi \cdot f1 + \overline{xi} \cdot f2$$

where f1 and f2 are reduced expressions for F with xi forced to 1 and 0 respectively. Thus, F can be implemented with a multiplexer controlled by xi which selects f1 or f2 depending on xi. f1 and f2 can themselves be decomposed into simpler expressions by the same technique.

To implement a multiplexer, we need both xi and $\overline{xi}$. Therefore, this logic family needs all inputs in true as well as in complement form. In order to drive

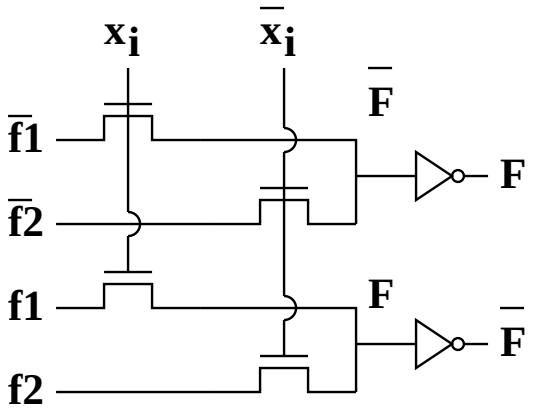


Figure 3.3: Basic Multiplexer with logic restoring inverters

other gates of the same type, it must produce the outputs also in true and complement forms. Thus each signal is carried by two wires. This logic style is called "Complementary Passgate Logic" or CPL for short.

3.2.1 Basic Multiplexer Structure

Pure passgate logic contains no 'amplifying' elements. Therefore, it has zero or negative noise margin. (Each logic stage degrades the logic level). Therefore, multiple logic stages cannot be cascaded. We shall assume that each stage includes conventional CMOS inverters to restore the logic level. Ideally, the multiplexer should be composed of complementary pass gate transistors. However, we shall use just n channel transistors as switches for simplicity.
This gives us the multiplexer structure shown in fig.3.3.

3.2.2 Logic Design using CPL

Since both true and complement outputs are generated by CPL, we do not need separate gates for AND and NAND functions. The same applies to OR-NOR, and XOR-XNOR functions.

To take an example, let us consider the XOR-XNOR functions. Because of the inverter, the multiplexer for the XOR output first calculates the XNOR function given by $A.B+\overline{A}.\overline{B}$. If we put A = 1, this reduces to B and for A = 0, it reduces to $\overline{B}$. Similarly, for the XNOR output, we generate the XOR expression = $A.\overline{B}+\overline{A}.B$ which will be inverted by the logic level restoring inverter. The expression reduces to $\overline{B}$ for A = 1 and to B for A = 0. This leads to an implementation of XOR-

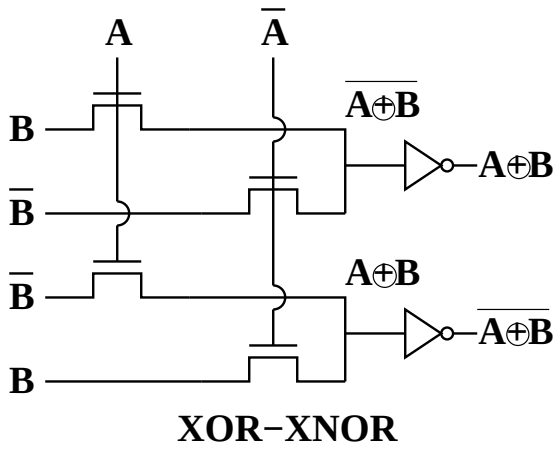


Figure 3.4: Implementation of XOR and XNOR by CPL logic.

XNOR as shown in fig.3.4

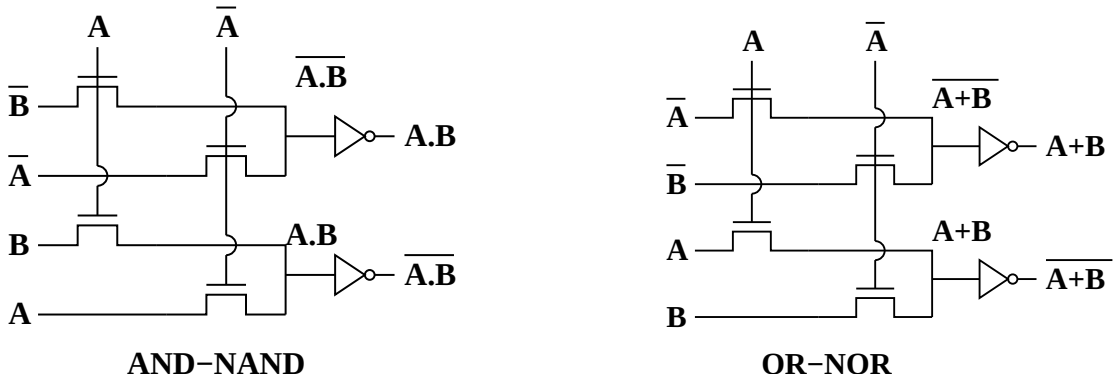


Figure 3.5: Implementation of (a) AND-NAND and (b) OR-NOR functions using complementary passgate logic.

Implementation of AND and OR functions is similar. In case of AND, the multiplexer should output $\overline{A.B}$ to be inverted by the buffer. This reduces to $\overline{B}$ when A = 1. When A = 0, it evaluates to $1 = \overline{A}$. For NAND output, the multiplexer should output A.B, which evaluates to B for A = 1 and to 0 (or A) when A = 0.

3.2.3 Buffer Leakage Current

The circuit configuration described above uses nMOS multiplexers. This limits

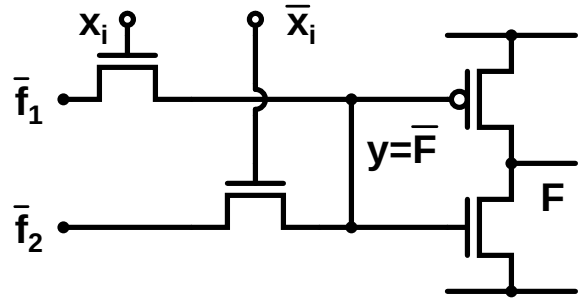


Figure 3.6: High leakage current in inverter

the 'high' output of the multiplexer (node y - which is the input for the inverter) to V_{dd} - V_{Tn}. Consequently, the pMOS transistor in the buffer inverter never quite turns off. This results in static power consumption in the inverter. This can be

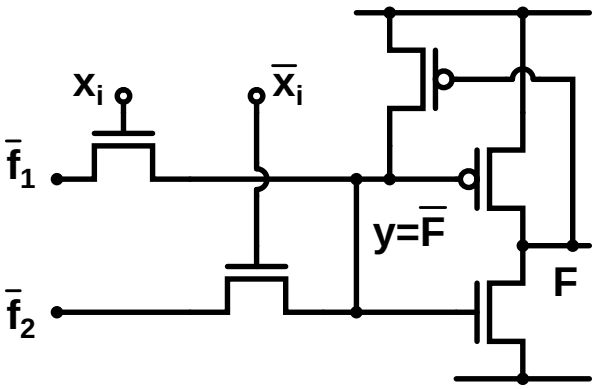


Figure 3.7: Pull up pMOS to avoid leakage in the inverter

avoided by adding a pull up pMOS as shown in fig. 3.7. When the multiplexer output (y) is 'low', the inverter output is high. The pMOS is therefore off and has no effect. When the multiplexer output goes 'high', the inverter input charges up, the output starts falling and turns the pMOS on. Now, as the multiplexer output (y) approaches V_{dd} - V_{Tn}, the nMOS switch in the multiplexer turn off. However, the pMOS pull up remains 'on' and takes the inverter input all the way to V_{dd}. This avoids leakage in the inverter.

However, this solution brings up another problem. Consider the equivalent circuit when the inverter output is 'low' and the pMOS is 'on'. Now if the multiplexer output wants to go 'low', it has to fight the pMOS pullup - which is trying to keep

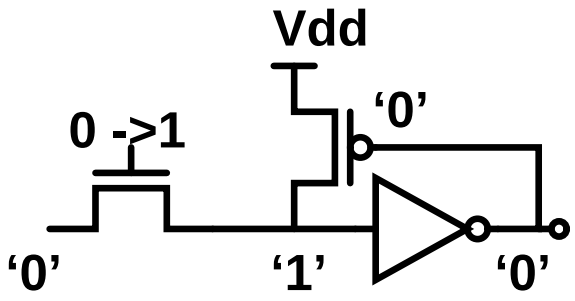


Figure 3.8: Problem with a low to high transition on the output

this node ‘high’.

In fact, the multiplexer n transistor and the pull up p transistor constitute a pseudo nMOS inverter. Therefore, the multiplexer output cannot be pulled low unless the transistor geometries are appropriately ratioed.

3.3 Cascade Voltage Switch Logic

We can understand this logic configuration as an attempt to improve pseudo-nMOS logic circuits. Consider the NOR gate shown below: Static power is consumed by

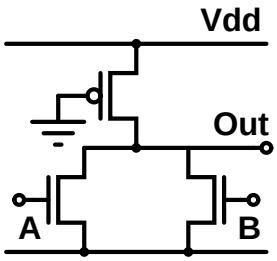


Figure 3.9: Pseudo-nMOS NOR

this NOR circuit whenever the output is ‘LOW’. This happens when A OR B is TRUE. We wish that the pMOS could be turned off for just this combination of inputs.

To turn the pMOS transistor off, we need to apply a ‘HIGH’ voltage level to its gate whenever A OR B is true. This obviously requires an OR gate. Non-inverting

gates cannot be made in a single stage. However, We can create the OR function by using a NAND of $\overline{A}$ and $\overline{B}$ as shown in figure 3.10. But then what about the

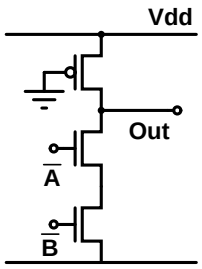


Figure 3.10: Pseudo-nMOS OR from complemented inputs

pMOS drive of this circuit?

We want to turn the pMOS of this OR circuit off when both $\overline{A}$ and $\overline{B}$ are 'HIGH'; i.e. when A = B = 0. This means we would like to turn the pMOS of this circuit off when the NOR of A and B is 'TRUE'.

But we already have this signal as the output of the first (NOR) circuit! So the two circuits can drive each other's pMOS transistors and avoid static power consumption. This kind of logic is called Cascade Voltage Switch Logic (CVSL). It

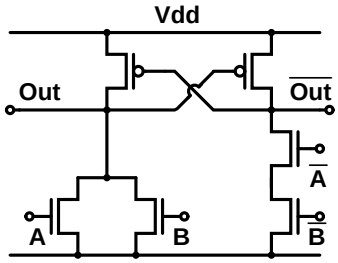


Figure 3.11: OR-NOR implementation in Cascade Voltage Switch Logic

can use any network f and its complementary network $\overline{f}$ in the two cross-coupled branches. The complementary network is constructed by changing all series connections in f to parallel and all parallel connections to series, and complementing all input signals.

CVSL shares many characteristics with static CMOS, CPL and pseudo-nMOS.

- Like CMOS static logic, there is no static power consumption.

- Like CPL, this logic requires both True and Complement signals. It also provides both True and complement outputs. (Dual Rail Logic).
- Like pseudo nMOS, the inputs present a single transistor load to the driving stage.
- The circuit is self latching. This reduces ratioing requirements.

3.4 Dynamic Logic

In this style of logic, some nodes are required to hold their logic value as a charge stored on a capacitor. These nodes are not connected to their 'drivers' permanently. The 'driver' places the logic value on them, and is then disconnected from the node. Due to leakage etc., the logic value cannot be held indefinitely. Dynamic circuits therefore require a *minimum* clock frequency to operate correctly. Use of dynamic circuits can reduce circuit complexity and power consumption substantially. When the clock is low, pMOS is on and the bottom nMOS is off. The output

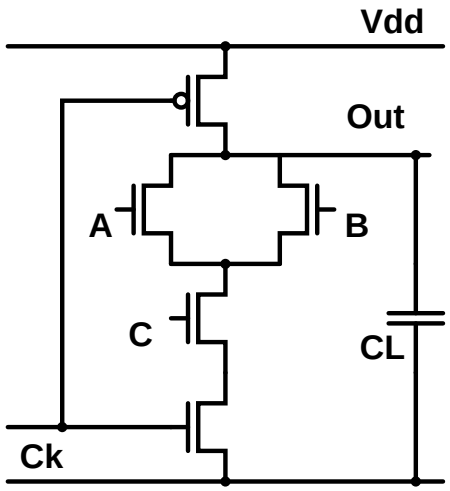
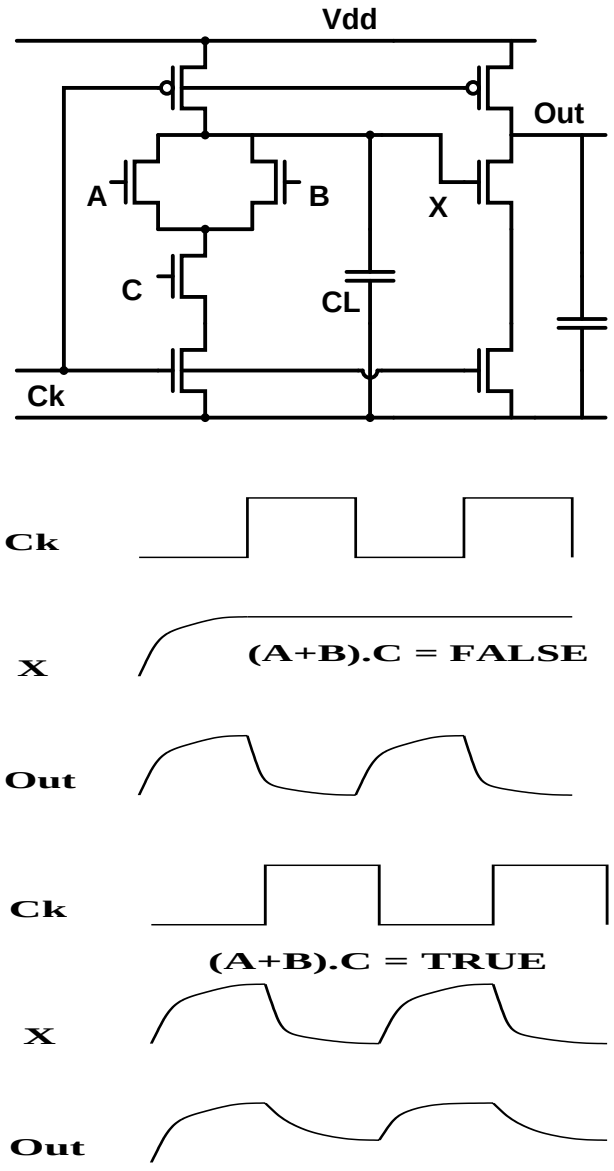


Figure 3.12: CMOS dynamic gate to implement $\overline{(A+B).C}$.

is 'pre-charged' to 1 unconditionally. When the clock goes high, the pMOS turns off and the bottom nMOS comes on. The circuit then conditionally discharges the output node, if (A+B).C is TRUE. This implements the function $\overline{(A+B).C}$.

3.4.1 Problem with Cascading CMOS dynamic logic

There is no problem when (A+B).C is false. X pre-charges to 1 and remains at 1.



When (A+B).C is TRUE, X takes some time to discharge. During this time, charge placed on the output leaks away as the input to nMOS of the inverter is not 0.

3.4.2 Four Phase Dynamic Logic

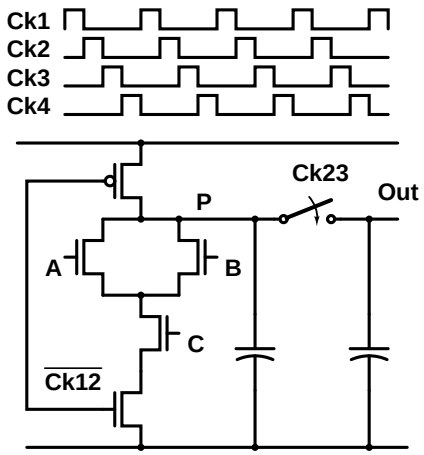


Figure 3.13: CMOS 4 phase dynamic logic

The problem can be solved by using a 4 phase clock. The idea is to sample the previous stage only after its evaluation is complete.

In phase 1, node P is pre-charged. In phase 2, P as well as the output are pre-charged. In phase 3, The gate evaluates. In phases 4 and 1, the output is isolated from the driver and remains valid. This is called a type 3 gate. It evaluates in phase 3 and is valid in phases 4 and 1. Similarly, we can have type 4, type 1 and type 2 gates. A type 3 gate can drive a type 4 or a type 1 gate. Similarly, type

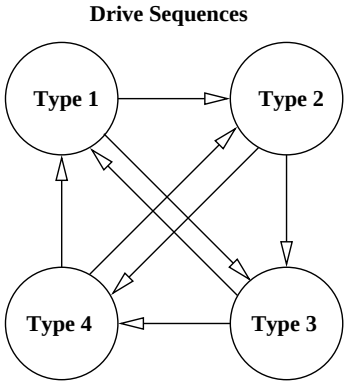


Figure 3.14: CMOS 4 phase dynamic logic drive constraints

4 will drive types 1 and 2; type 1 will drive types 2 and 3; and type 2 will drive

types 3 and 4. We can use a 2 phase clock if we stick to type 1 and type 3 gates (or type 2 and type 4 gates) as these can drive each other.

3.4.3 Domino Logic

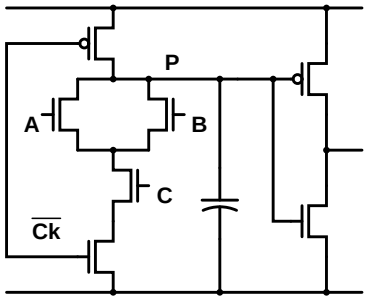


Figure 3.15: CMOS domino logic

Another way to eliminate the problem with cascading logic stages is to use a static inverter after the CMOS dynamic gate. Recall that the cascaded dynamic CMOS stage causes problems because the output is pre-charged to V_{dd}. If the final value is meant to be zero, the next stage nMOS to which the output is connected erroneously sees a one till the pre-charged output is brought down to zero. During this time, it ends up discharging its own pre-charged output, which it was not supposed to do. If an inverter is added, the output is held 'low' before logic evaluation. If the final output is zero, there is no problem anyway. If the final output is supposed be one, the next stage is erroneously held at zero for some time. However, this does not result in a false evaluation by the next stage. The only effect it can have is that the next stage starts its evaluation a little later. However, the addition of an inverter means that the logic is non-inverting. Therefore, it cannot be used to implement any arbitrary logic function.

3.4.4 Zipper logic

Instead of using an inverter, we can alternate n and p evaluation stages. The n stage is pre-charged high, but it drives a p stage. A high pre-charged stage will keep the p evaluation stage off, which will not cause any malfunction. The p stage will be pre-discharged to 'low', which is safe for driving n stages. This kind of logic is called zipper logic.

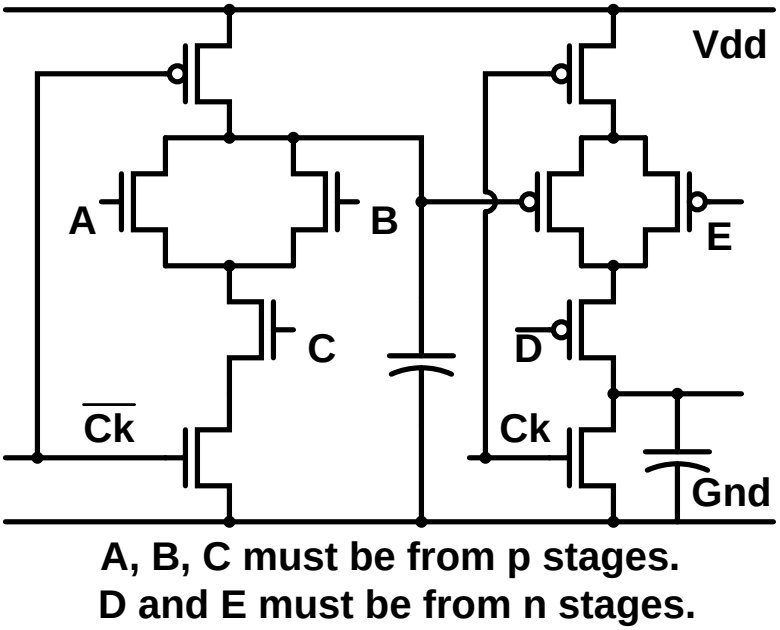


Figure 3.16: Zipper logic