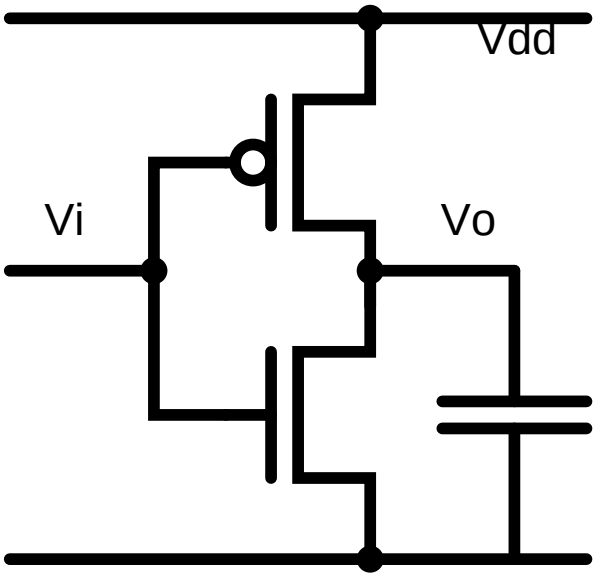
Logic Design Styles

Dinesh Sharma

Microelectronics Group, EE Department
IIT Bombay, Mumbai

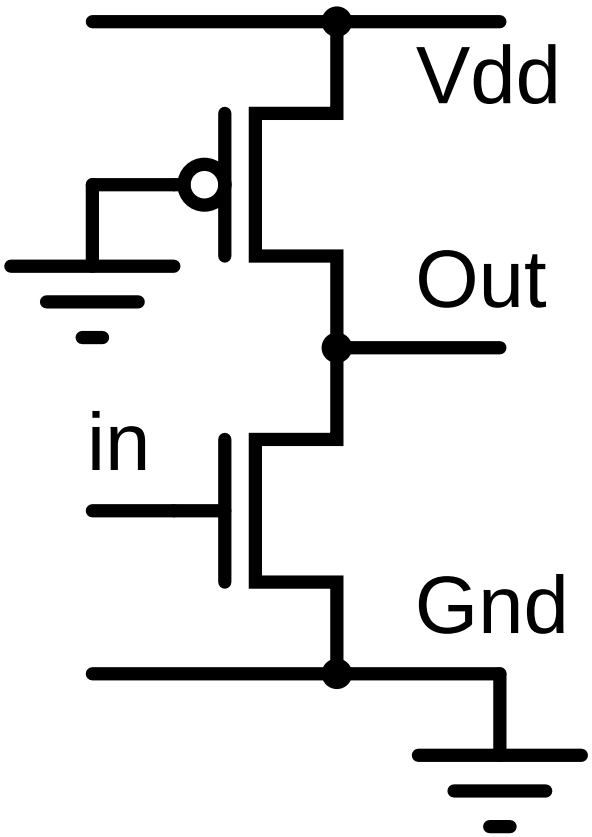
July 26, 2016

CMOS summary



- Logic consumes no static power in CMOS design style.
- However, signals have to be routed to the n pull down network *as well as* to the p pull up network.
- So the load presented to every driver is high.
- This 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



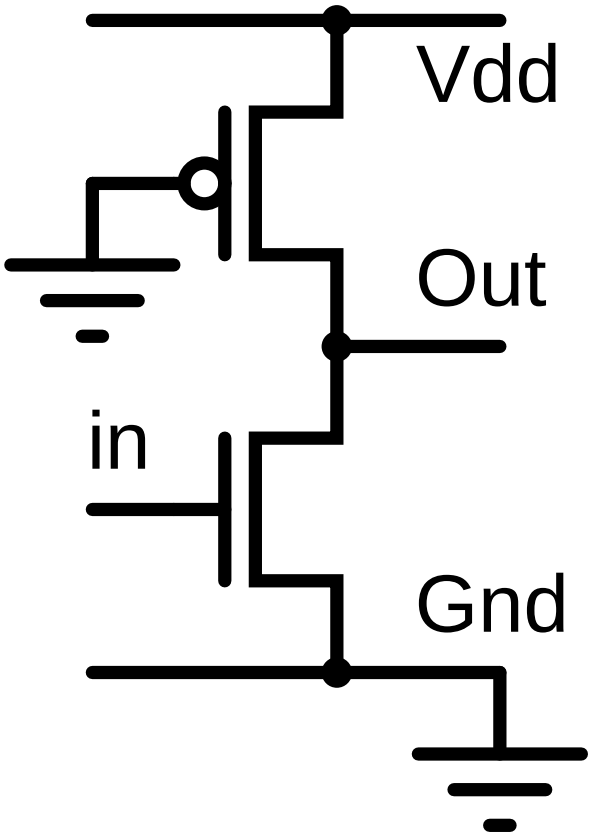
- The CMOS pull up network is replaced by a single pMOS transistor with its gate grounded.
- Since the pMOS is not driven by signals, it is always ‘on’.
- The effective gate voltage seen by the pMOS transistor is V_{dd}. Thus the over-voltage on the p channel gate is always V_{dd}- V_{Tp}.
- When the nMOS is turned ‘on’, a direct path between supply and ground exists and static power will be drawn.
- However, the dynamic power is reduced due to lower capacitive loading

Static Characteristics

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

- nMOS 'off'
- nMOS saturated, pMOS linear
- nMOS linear, pMOS linear
- nMOS linear, pMOS saturated

Low input



- When the input voltage is less than V_{Tn}. The output is 'high' and no current is drawn from the supply.
- As we raise the input just above V_{Tn}, the output starts falling.
- In this region the nMOS is saturated, while the pMOS is linear

nMOS saturated, pMOS linear

The input voltage is assumed to be sufficiently low so 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

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

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

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

nMOS saturated, pMOS linear

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

The solutions are:

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

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

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

nMOS linear, pMOS linear

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

- As the input voltage is increased, the output voltage will decrease.
- The output voltage will fall below $V_i - V_{Tn}$ 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. The derived equation does not apply beyond this input voltage.

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

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

Noise Margins

We find points on the transfer curve where the slope is -1. When the input is low and output high, we should use

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

Differentiating this equation with respect to V_i and setting the slope to -1, we get

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

and

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

When the input is high and the output low, we use

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

Differentiating with respect to V_i and setting the slope to -1, we get

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

and

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

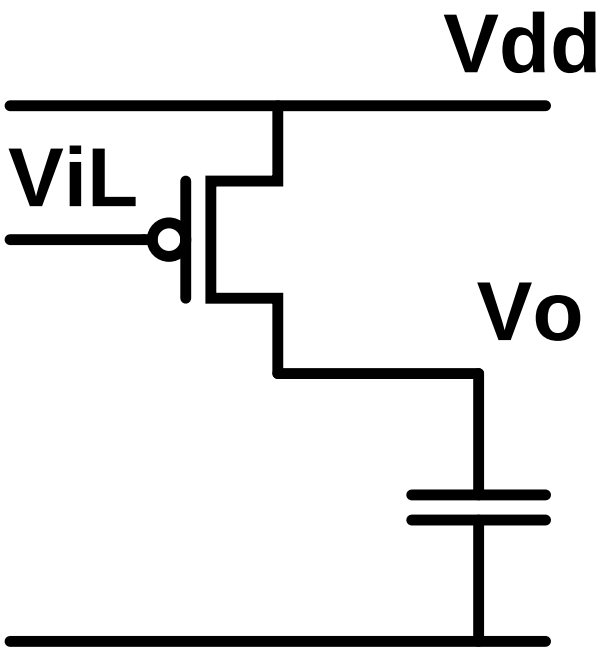
Ratioed Logic

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 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}.

Rise Time

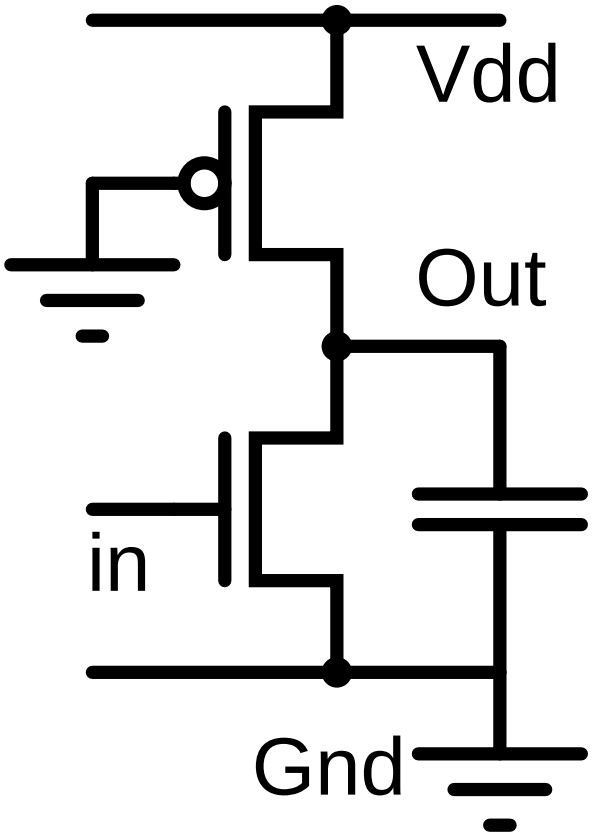


When the input is low, the nMOS is off and the output rises from 'low' to 'high'.
The situation is identical to the charge up condition of a CMOS gate with the pMOS being biased with its gate at 0V.

This gives

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

Fall Time



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.
The nMOS needs to sink the discharge current as well as the drain current of the pMOS transistor.
Simplifying assumption:
pMOS current remains constant at its saturation value through the entire discharge process.

(This will result in a slightly pessimistic value of discharge time).

Fall Time

If we assume that the pMOS current remains constant at its saturation value,

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

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

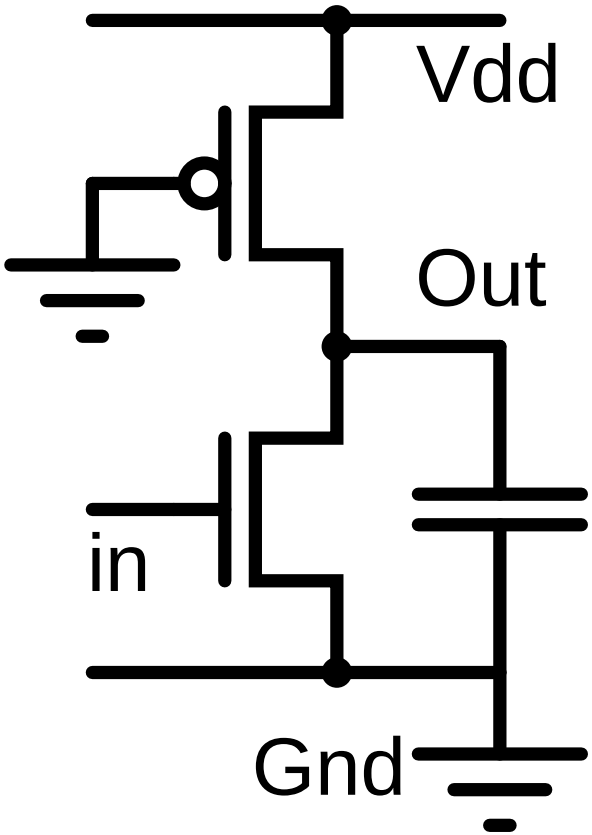
$$I_n - I_p + C\frac{dV_o}{dt} = 0$$

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 $V_2 \equiv V_{dd} - V_{Tp}$.

Fall Time



The integration range can be divided into two regimes.

- nMOS is saturated when $V_1 \leq V_o < V_{dd}$.
- It is in the linear regime when $V_{oL} < V_o < V_1$.

Fall Time

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

so,

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

Pseudo nMOS Inverter design

- We design the basic inverter and then scale device sizes based on the logic function being designed.
- The load device size is calculated from the rise time.

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

- 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.

Pseudo nMOS Inverter design

- Geometry of the n channel transistor can be determined from static considerations.

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

- We take V_{oL}= V_{Tn}, and 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.

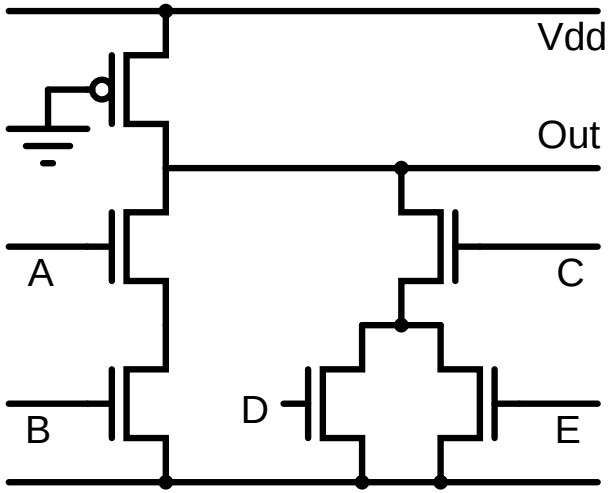
Conversion to other logic

- 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.

Conversion to other logic

- 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.
- 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.

$\overline{A.B + C.(D + E)}$ in pseudo-nMOS



- A and B are in series.
- The pair is in parallel with C which is in series with a parallel combination of D and E.

Implementation of $\overline{A.B + C.(D + E)}$ in pseudo-nMOS logic design style.