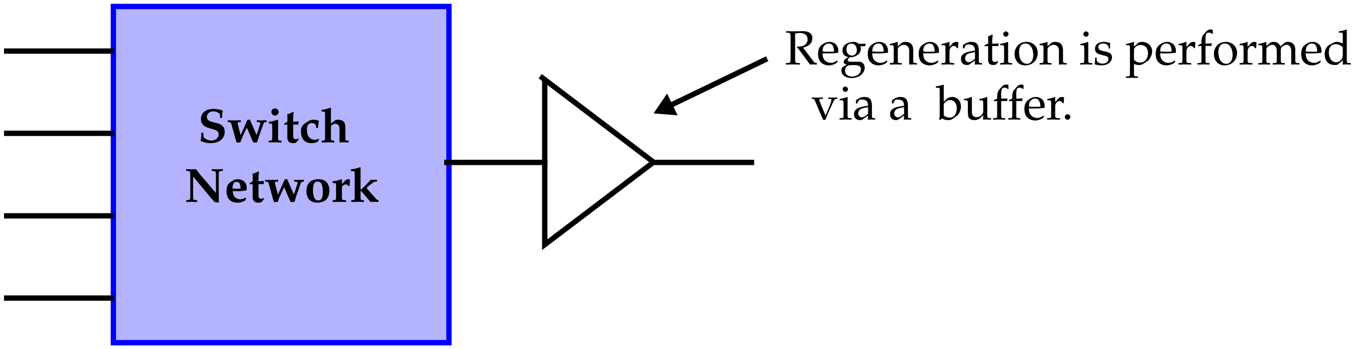
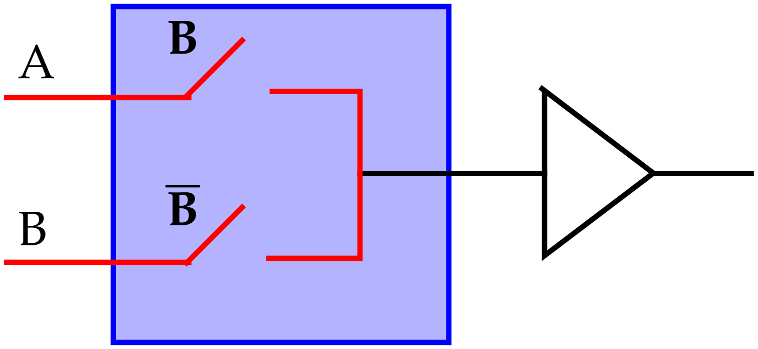
Pass Gate Logic

An alternative to implementing complex logic is to realize it using a logic network of pass transistors (switches).



Regeneration is performed via a buffer.

We have already observed a series connection of two switches implements AND while a parallel connection implements OR.



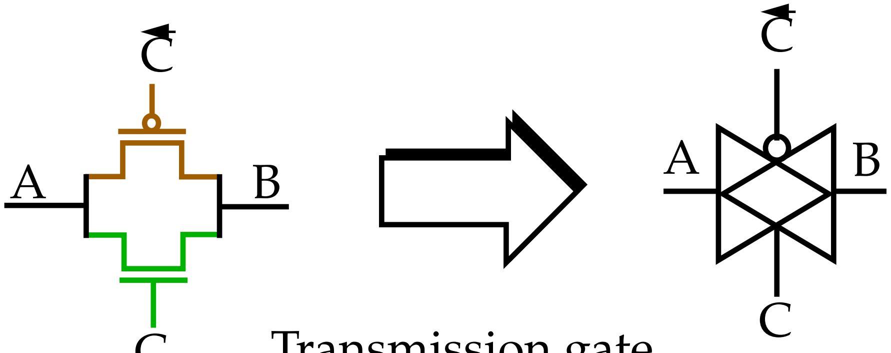
$\overline{\text{B}}$ is not redundant, it ensures a low impedance path exists when B is low.

Pass Gate Logic

Advantage: fast and simple.

Complex gates can be implemented using minimum number of transistors, which also reduces parasitics.

Static and dynamic performance depends on a switch with low parasitic resistance and capacitance.

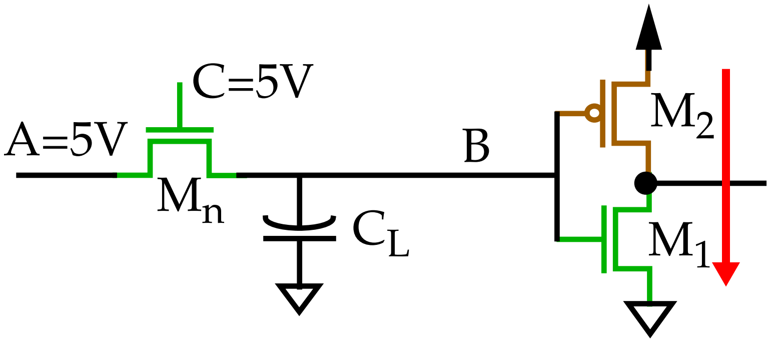


Transmission gate

Therefore, pass gate networks are often constructed from bi-directional transmission gates.

Pass Gate Logic

Both transistors are important:



Here, M_n turns off when V_B reaches (5 - V_{Tn}) or approximately 3.5V!

Note, the V_{Tn} is increased due to the **body effect**.

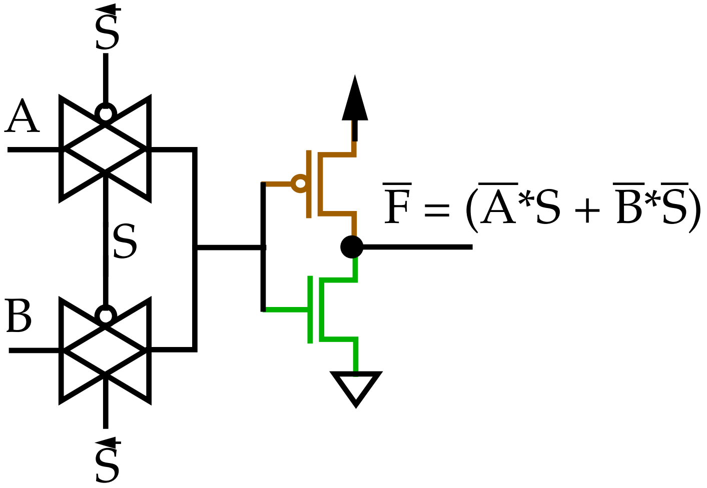
This reduces the *noise margin* and increases *static power* dissipation.

Also, the **resistance** of the switch increases dramatically when the output voltage reaches V_{in} -V_{Tn} (linear mode).

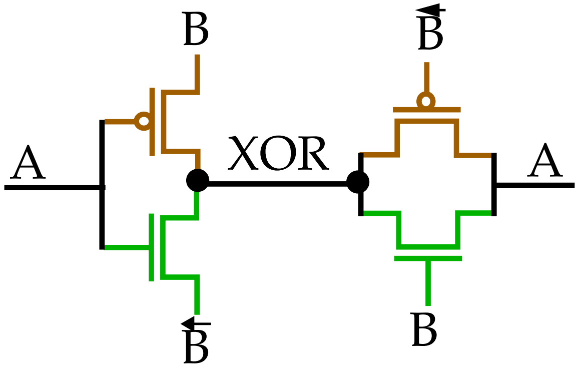
The combination of both an PMOS and NMOS avoids this problem but requires that the control and its complement be available.

Pass Gate Logic

Transmission gates can implement complex gates very efficiently


$\overline{S}$
A
S
B
$\overline{S}$
$\overline{F} = (\overline{A}*S + \overline{B}*\overline{S})$

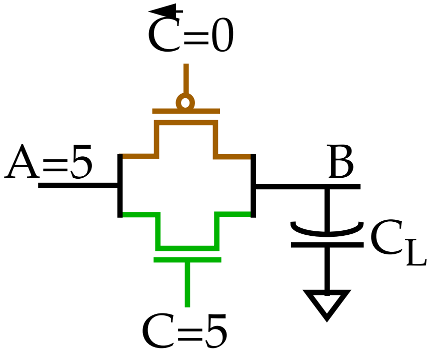
2-to-1 MUX requires 6 transistors



XOR requires 6 transistors

Design Issues:

- Resistance.



Parallel connection of resistances R_n and R_p

$$R_n = (V_{DD} - V_{out})/I_n$$

$$R_p = (V_{DD} - V_{out})/I_p$$

Currents are dependent on V_{out} and operation region

Pass Gate Logic Design Issues

- Resistance (cont).

During the *low-to-high* transition, the pass transistors pass through several operation modes.

As V_{GS} is always equal to V_{DS}, the NMOS is either in saturation or off.

The V_{GS} of the PMOS is V_{DD}, and the device changes from saturation to linear.

- $V_{out} < |V_{Tn}|$: NMOS and PMOS saturated.
- $|V_{Tp}| < V_{out} < V_{DD} - V_{Tn}$: NMOS saturated, PMOS linear.
- $V_{DD} - V_{Tn} < V_{out}$: NMOS cutoff, PMOS linear.

It is important to incorporate the *body effect* when computing I_p and I_n.

The expression for the resistance of a pass gate *without* the body effect.

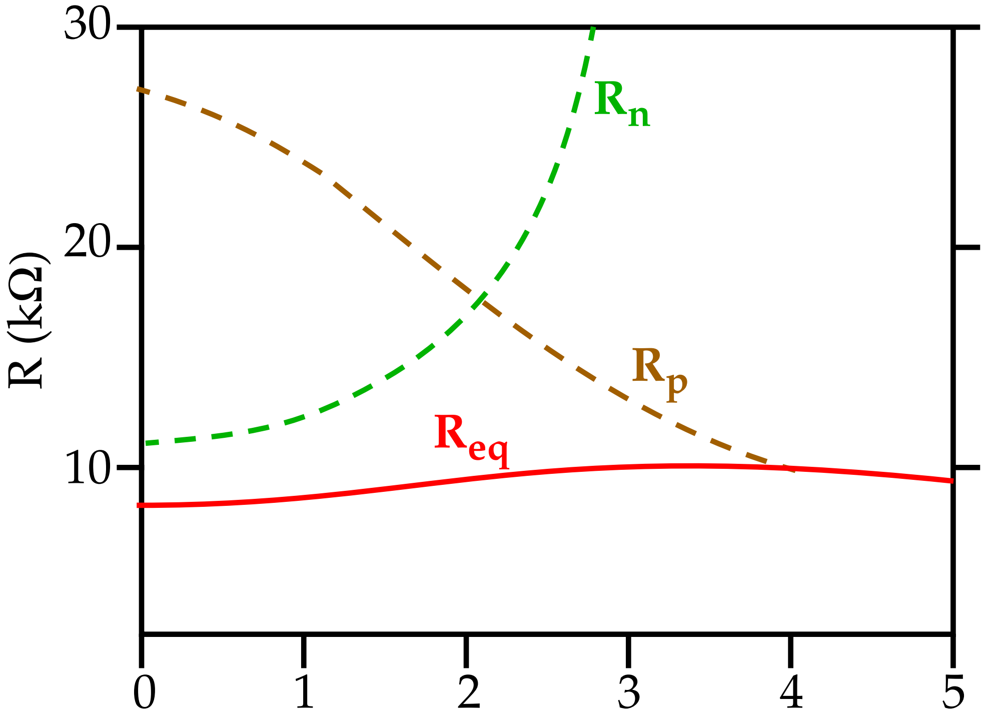
$$R_{eq} \approx \frac{1}{k_n(V_{DD} - V_{Tn}) + k_p(V_{DD} - |V_{Tp}|)}$$

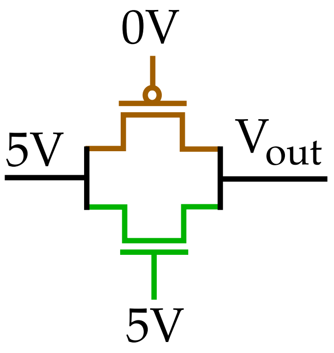
Pass Gate Logic Design Issues

- Resistance (cont).

Simulated values of :

$$R_{eq} = R_p \parallel R_n$$

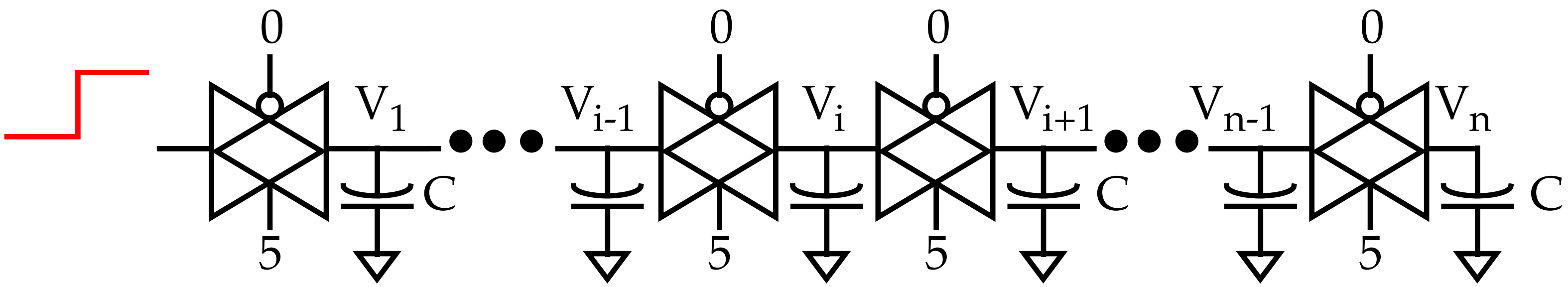




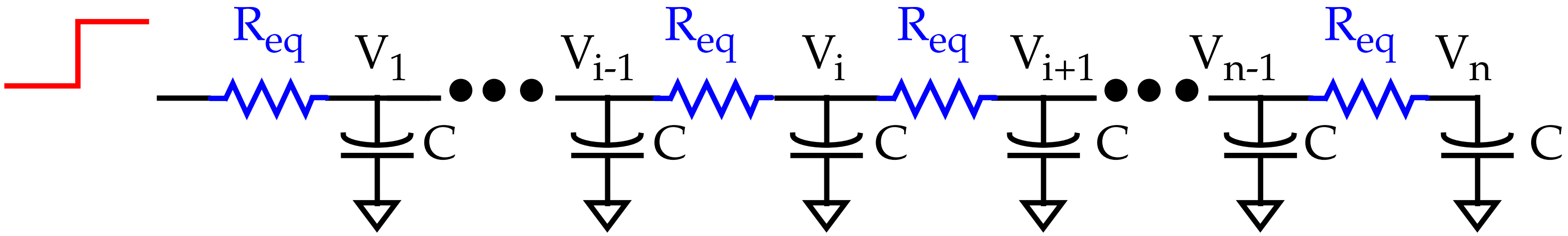
R_{eq} is relatively constant at 10 kΩ so a **constant resistance** switch model is reasonable.

Pass Gate Logic Design Issues

- Delay



In order to analyze the response, let’s replace the pass gates with R_{eq}s.



Delay is found by solving a set of differential equations of the form:

$$\frac{\partial V_i}{\partial t} = \frac{1}{R_{eq}C}(V_{i+1} + V_{i-1} - 2V_i)$$

Pass Gate Logic Design Issues

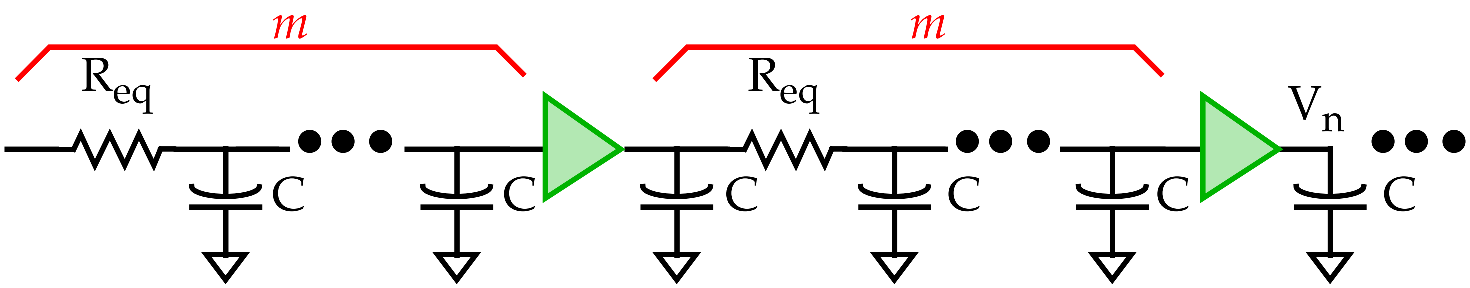
- Delay (cont).

An estimate of the dominant time constant at the output of n pass gates:

$$\tau(V_n) = \sum_{k=0}^{n} CR_{eq}k = CR_{eq}\frac{n(n+1)}{2}$$

Propagation delay is proportional to n^2!

For large n, it is better to break the chain every m switches and insert buffers:



Total delay assuming buffer delay is t_{buf} is:

$$t_p = 0.69\left[\frac{n}{m}CR_{eq}\frac{m(m+1)}{2}\right] + \left(\frac{n}{m} - 1\right)t_{buf} = 0.69\left[CR_{eq}\frac{n(m+1)}{2}\right] + \left(\frac{n}{m} - 1\right)t_{buf}$$

Pass Gate Logic Design Issues

- Delay (cont).

Here, delay exhibits only a linear dependence on the # of switches *n*.

The optimal number of switches, m_{opt}, between buffers is found:

$$\frac{\partial t_p}{\partial m} = 0 \longrightarrow m_{opt} = 1.7\sqrt{\frac{t_{pbuf}}{CR_{eq}}}$$

As t_{buf} increases, the number of switches grows.

In current technologies, m_{opt} is typically 3 or 4.

For example, assume R_{eq} = 10kΩ, C = 10fF, and t_{pbuf} = 500ps.

This yields an optimal value of *m* equal to **3.8**.

Therefore, a buffer every 4 transmission gates is suggested.

Pass Gate Logic Design Issues

- Transistor sizing

  Pass gate logic family is a member of the *ratioless* logic class.

  The dc characteristics are not affected by the sizes.

  Performance, to the first order, is **not impacted** by changing the W/L.

  Increasing the size reduces the resistance, but this is offset by the increase in diffusion capacitance.

  Therefore, minimum sized devices should ALWAYS be used, unless the chain drives a significant external load capacitance.

  In this case, ordering transistors from largest to smallest in the pass gate chain will help reduce delay.

  This is analogous to the argument given earlier for logic gate transistors close to the output.

NMOS-Only Transmission Gate

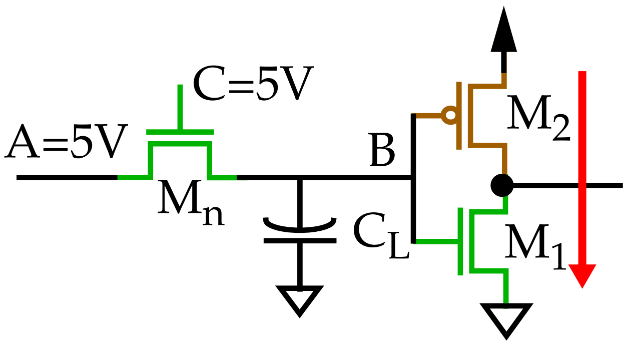
Disadvantages of pass gate:

- Requires both NMOS and PMOS, in different wells.
- Both true and complemented polarities of the control signal needed.
- Parallel connection of both transistors increases node capacitance.

Therefore, an *NMOS-only* version is advantageous.

Problems:

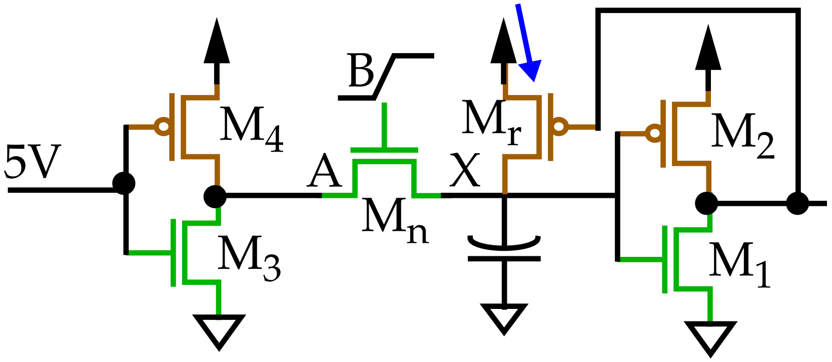
- Reduced noise margins due to the threshold voltage drop.
- Static power consumption.



NMOS-Only Transmission Gate

One solution is to add a PMOS device, called a **level restorer**.

Level restorer



The output of the inverter is "feedback" as a control signal.

It turns on when the inverter output goes low ($V_{out} < V_{DD} - |V_{tp}|$) and restores node *X* to V_{DD}.

This eliminates the static power consumed.

However, the size of the PMOS transistor is important, since a conflict is created during switching.

For example, assume node *A*=0, storage node *X*=V_{DD} and *B*=0->1.

A conducting path exists from V_{DD}-M_r-M_n-M_3-GND.

NMOS-Only Transmission Gate

Let R_r, R_n and R_3 represent the resistances of transistors M_r, M_n and M_3.

If R_r is too small, it will be *impossible* to bring node *X* below V_M.

This is called the **writability problem**, used in reference to feedback circuits.

Let's simplify the analysis of finding the switching point by grounding M_r's input (open the feedback loop).

Assume M_r is in *linear* mode, M_n is in *saturation* and V_A is close to GND.

$$I = k_3(V_{DD} - V_{Tn})V_A \qquad \text{(linear)} \qquad (1)$$

$$I = \frac{k_n}{2}(V_B - V_A - V_{Tn})^2 \qquad (\text{for } V_X = V_M) \qquad (2)$$

$$I = k_r\left[(V_{DD} - |V_{Tp}|)(V_{DD} - V_M) - \frac{(V_{DD} - V_M)^2}{2}\right] \qquad (3)$$

I is set by (3), which allows V_A to be found via (1) and then V_B as a function of the k-parameters (the objective).

NMOS-Only Transmission Gate

Let’s set the condition that $V_B < V_{DD}$ -- in other words, some value of V_B less than V_{DD} will set $V_X < V_M$ (which allows the inverter to switch).

Assume the sizes of M_3 and M_n are identical and V_{DD}=5V, $V_{Tn}=|V_{Tp}|$=0.75V and V_M=2.5V:

$$V_B = 3.87\sqrt{\frac{k_r}{k_n}} + 1.76\frac{k_r}{k_n} + 0.75 \le 5V$$

The boundry condition for this constraint to be valid is $m = k_n/k_p > 1.55$.

Smaller values do not allow the inverter to switch.

Using a value of 3 is reasonable, which amounts to making the NMOS pass gate transistor equal to PMOS restoring device.

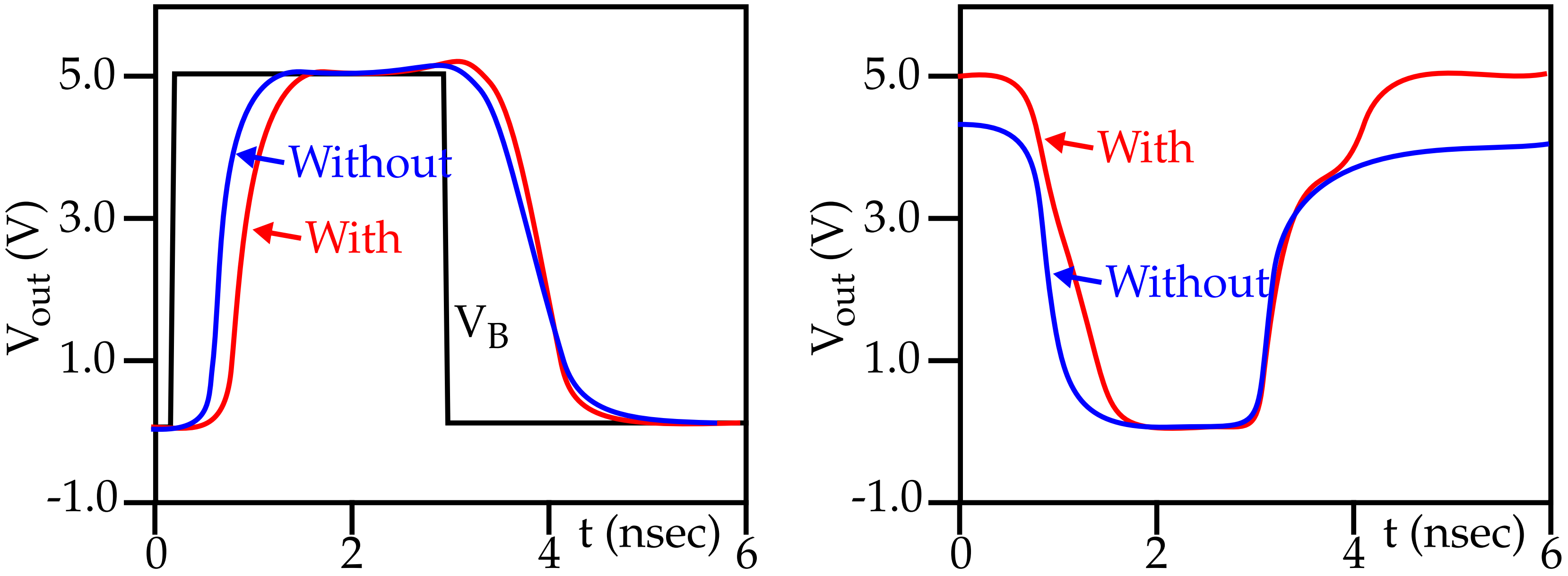
What about performance?

Adding the level restorer increases the capacitance at V_X.

Also, the rise time of the inverter is slowed due to the fight.

NMOS-Only Transmission Gate

However, the fall time is improved slightly.



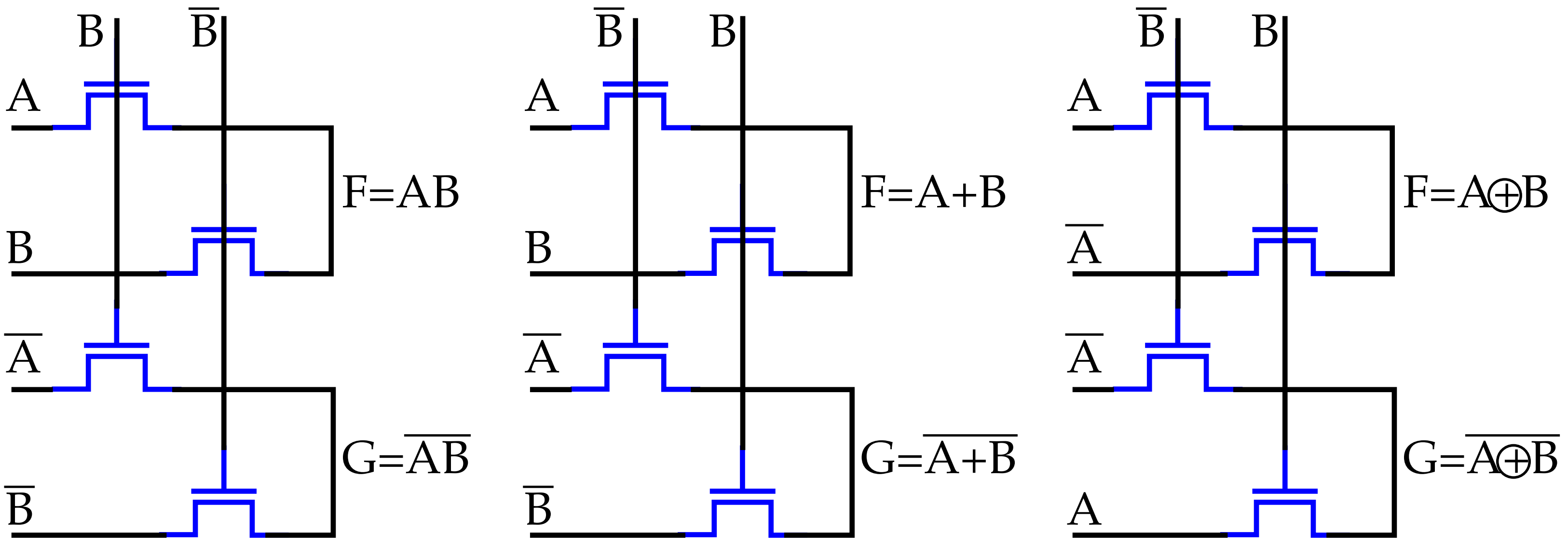
A second method of implementing NMOS-only pass gate networks is to change V_T (if your manufacturer supports it).

A zero V_T transistor for M_n (a natural device) is one possibility.

This logic style is called **Complementary Pass-Transistor Logic** (CPL).

CPL

Examples:



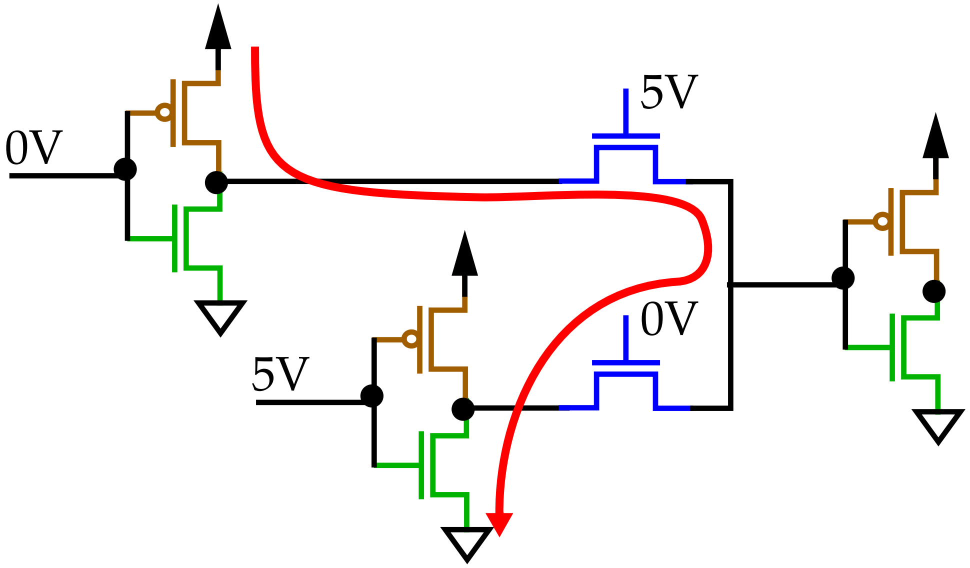
Properties:

- They are *differential* circuits.

  Eliminates inverters and allows minimal implementations, e.g., XOR.
- CPL is *static* (low impedance connection to V_{DD} and GND).
- V_T (including body effect) is reduced to below $|V_{Tp}|$, eliminating *static power* in successor gates.
- The design is *modular* -- all gates use exactly the same topology.

CPL

The main disadvantages is that turning off a zero-V_T device is hard (plus it has a reduced noise margin).



Note that a 4-input NAND requires three 2-input NANDs + buffer for **14** transistors, which is > **8** for the full complementary version!

The applicability of CPL is strongly dependent on the logic function to be implemented, e.g. 2-transistor XOR good for multipliers and adders.

CPL is extremely fast and efficient. Routing overhead is significant however.