CMOS and BiCMOS Logic

TTL logic is used widely, but has a few limitations:

- The input impedance is not infinity. This limits the fanout.
- The corresponding power consumption is somewhat high (5mW computed previously).

CMOS avoids these problems.

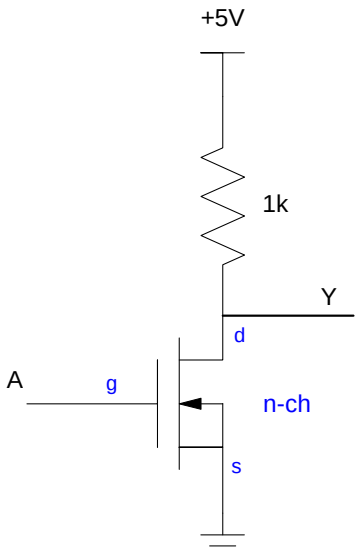
For the following circuits, assume

$$R_{ds} = 1\Omega \qquad \text{@ Vgs = 5V}$$

CMOS Inverter

Version 1: n-channel MOSFET.

- Whan VA = 0V, the MOSFET is off. Rds = infinity
- When VA = 5V, the MOSFET is on: Rds = 1 Ohm (approx)



CMOS inverter (take 1)

This results in the following table:

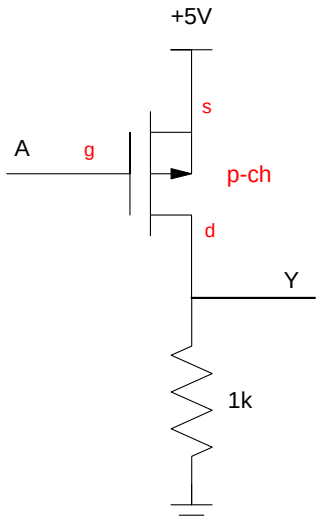
VA	Rhigh	Rlow	Y
0V	1k	infinity	+5V
5V	1k	1 Ohm	0.005V

Note that this is an inverter:

$$Y = \overline{A}$$

Version 2: p-channel MOSFET

- Whan VA = 5V,
  - Vgs = 0V
  - The MOSFET is off. Rds = infinity
- When VA = 0V,
  - Vgs = -5V
  - The MOSFET is on: Rds = 1 Ohm (approx)



CMOS Inverter (take 2)

This results in the same table:

VA	Rhigh	Rlow	Y
0V	1 Ohm	1k	4.995 V
5V	infinity	1k	0.V

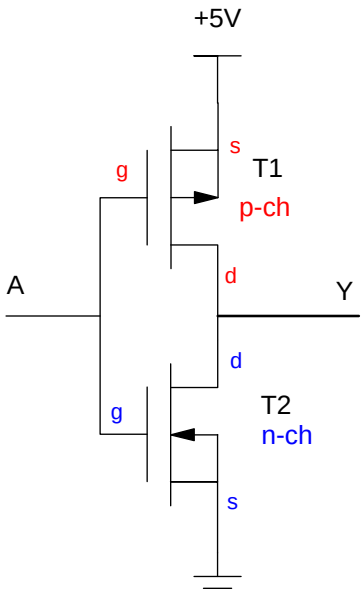
Again, note that this is an inverter

$$Y = \overline{A}$$

Version 3: Actual CMOS Inverter

The problem with the previous two designs is that when the MOSFET is turned on, you're drawing 5mA. Multiply this by 1 million gates and you have a major problem with power consumption.

To avoid this problem, combine the previous two circuits:



CMOS Inverter

This results in the same logic but with the current consumption being zero: the total resistance between power and ground is always infinite:

VA	Rhigh (T1)	Rlow (T2)	Y
0V	1 Ohm	infinity	5.00 V
5V	infinity	1 Ohm	0.00 V

Actually, there will be a slight current draw when you switch logic levels. Momentarily, the MOSFETs go through the saturated region where current is allowed to flow. This results in spikes in the current consumption: spikes at the points where you change logic levels (the clock edges typically).

CMOS NOR gate:

When implementing CMOS logic, both the high side and low side of the circuit are made up of CMOS logic. The two are related by DeMorgan's theorem:

- On the low side, when the MOSFETs are turned on, the output is pulled low (~Y)
- On the high side, when the MOSFETs are turned on, the output is pulled high (Y)
- DeMorgan's theorem allows you to convert from the low side (~Y) to high side (Y)

For the sake of simplicity, let's ignore the high-side and assume it's just a 1k resistor.

If you have a single MOSFET to ground, you wind up with an inverter:

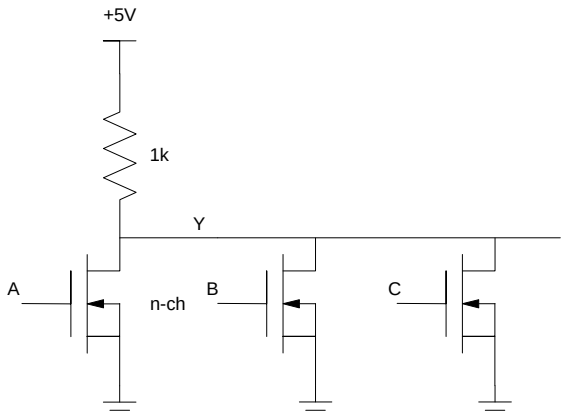
$$Y = \overline{A}$$

If you add more MOSFETs to ground, the output will be low if any of the MOSFETs are turned on.

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

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

meaning you have a NOR gate.



NOR gate with n-channel MOSFETs

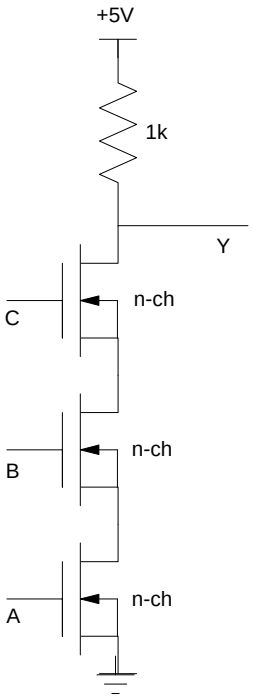
CMOS NAND gate:

In contrast, if you place the MOSFETs in series, Y will be pulled low if A and B and C are turned on:

$$\overline{Y} = ABC$$

$$Y = \overline{ABC}$$

resulting in a NAND gate:



NAND Gate with n-channel MOSFETs

CMOS Combinational Logic:

One neat thing about CMOS logic is you can implement an entire logic function with a single gate by using MOSFETs in series and parallel. For example, design a circuit to implement

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

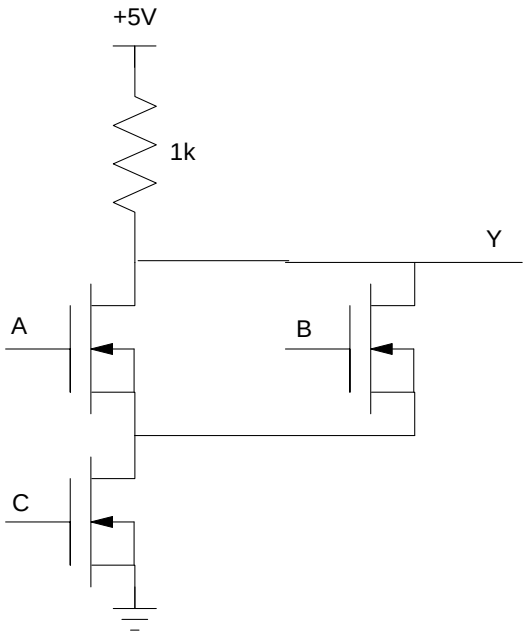
Solution: Use DeMorgan's theorem:

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

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

OR means parallel.

AND means series



$Y = \overline{A}\overline{B} + \overline{C}$ using n-channel MOSFETs

Sidelight: The high-side is the dual of the low side. From DeMorgan's theorem:

$Y = \overline{A}\overline{B} + \overline{C}$ (p-channel: high side)

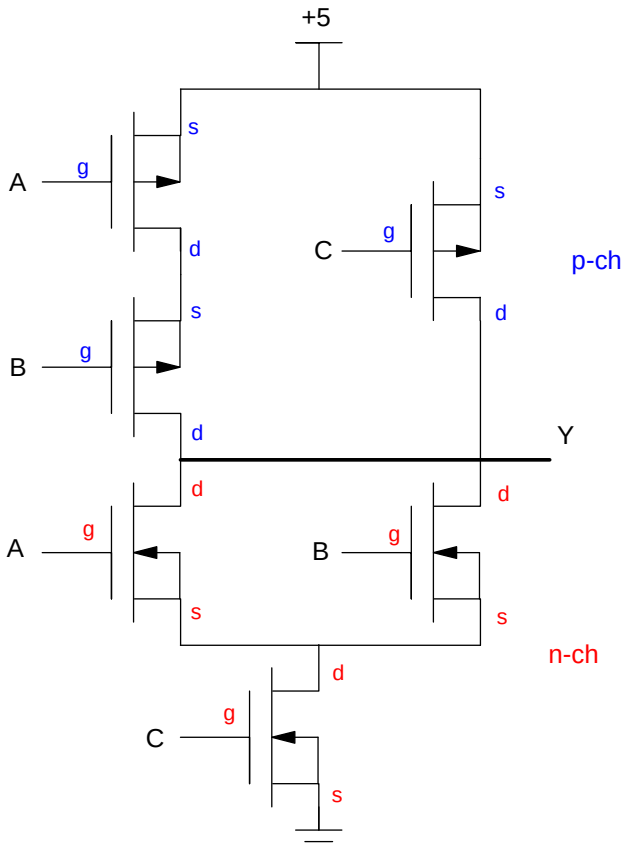
$\overline{Y} = (A + B)C$ (n-channel: low side)

Use the equation for Y to design the high-side.

Use the equation for ~Y for the low side.

This results in

- A and B are in series on the high side (AND) and parallel on the low side (OR)
- C is in parallel on the high-side (OR) and series on the low side (AND)



CMOS logic for $Y = \overline{A}\overline{B} + \overline{C}$