Georgia Tech College of Engineering
School of Electrical and Computer Engineering

ECE 2020 Lecture 4: Transistors, CMOS circuits

Instructor: Samuel Talkington

September 5, 2024

Logistics

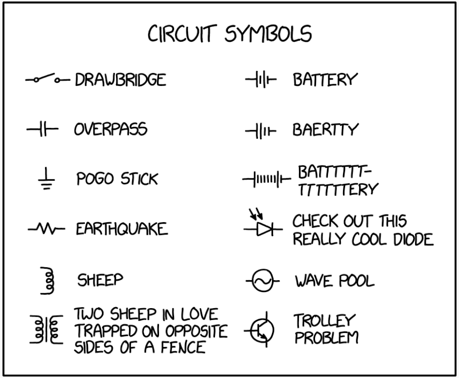
- **Adjusted Friday office hours** this week: 12:30pm-2:00pm in person.
- First homework assignment was due last night. Great work!
- Coming soon:
  - Tonight: HW 2—due September 17th **with optional 2 day extension** to Thursday, September 19th.
  - Tonight: Updated course schedule.
  - Weekend: Pre-lab worksheet—due September 12th before class.
- Schedule for the next two weeks:
  - First lab next Thursday, September 12th, in-class.
  - First exam, September 19th, in-class.

Agenda

Today's agenda

Going down one *layer of abstraction*:

- Transistors
- NMOS vs PMOS
- Inverters
- Pull-up and pull-down networks
- CMOS circuits



Source: xkcd

Recap

Let $F : \{0,1\}^3 \to \{0,1\}$ denote the logic function:

$$F(A,B,C) = \overline{\left(A \cdot B + \overline{A} \cdot B \cdot \overline{C} + A \cdot C\right)} \cdot \left(A + \overline{B \cdot C}\right)$$

Simplify this function as much as possible using Boolean Algebra theorems.

Note that

$$
\begin{aligned}
F(A,B,C) &= \overline{(A \cdot B + \overline{A} \cdot B \cdot \overline{C} + A \cdot C)} \cdot (A + \overline{B \cdot C}) \\
&\overset{(1)}{=} \left(\overline{(A \cdot B)} \cdot \overline{(\overline{A} \cdot B \cdot \overline{C})} \cdot \overline{(A \cdot C)}\right) \cdot (A + \overline{B \cdot C}) \\
&\overset{(2)}{=} ((\overline{A} + \overline{B}) \cdot (A + \overline{B} + C) \cdot (\overline{A} + \overline{C})) \cdot (A + \overline{B \cdot C}) \\
&\overset{(3)}{=} \left(\left(\underbrace{\overline{A} \cdot A}_{=0} + \overline{A} \cdot \overline{B} + \overline{A} \cdot C + \overline{B} \cdot A + \underbrace{\overline{B} \cdot \overline{B}}_{=\overline{B}} + \overline{B} \cdot C\right) \cdot \left(\overline{A} + \overline{C}\right)\right) \cdot (A + \overline{B \cdot C}) \\
&= \left(\left(\overline{A} \cdot C + \underbrace{\underbrace{\underbrace{\overline{B} \cdot A + \overline{B}}_{=\overline{B}} + \overline{B} \cdot C}_{=\overline{B}+\overline{B}\cdot C=\overline{B}} + \overline{A} \cdot \overline{B}}_{=\overline{B}+\overline{A}\cdot\overline{B}=\overline{B}}\right) \cdot \left(\overline{A} + \overline{C}\right)\right) \cdot (A + \overline{B \cdot C}),
\end{aligned}
$$

where:

Step (1) is by DeMorgan's Theorem, $\overline{X + Y + Z} = \overline{X} \cdot \overline{Y} \cdot \overline{Z}$.

Step (2) Is by DeMorgan's Theorem, $\overline{X \cdot Y \cdot Z} = \overline{X} + \overline{Y} + \overline{Z}$, applied multiple times.

Step (3) is by the distributive property, $(A + B)(C + D) = AC + AD + BC + BD$

then,

$$
\begin{aligned}
F(A,B,C) &\overset{(4)}{=} (\overline{A}\cdot C+\overline{B})\cdot(\overline{A}+\overline{C})\cdot\underbrace{(A+\overline{B\cdot C})}_{=A+\overline{B}+\overline{C}} \\
&\overset{(5)}{=} (\overline{A}\cdot C+\overline{B})\cdot\Big(\underbrace{\overline{A}\cdot A}_{=0}+\overline{A}\cdot\overline{B}+\overline{A}\cdot\overline{C}+\overline{C}\cdot A+\overline{C}\cdot\overline{B}+\underbrace{\overline{C}\cdot\overline{C}}_{=\overline{C}}\Big) \\
&= (\overline{A}\cdot C+\overline{B})\cdot\Big(\overline{A}\cdot\overline{B}+\underbrace{\overline{A}\cdot\overline{C}+\underbrace{\overline{C}\cdot A+\underbrace{\overline{C}\cdot\overline{B}+\overline{C}}_{=\overline{C}}}_{=\overline{C}\cdot A+\overline{C}=\overline{C}}}_{=\overline{A}\cdot\overline{C}+\overline{C}=\overline{C}}\Big) \\
&\overset{(6)}{=} (\overline{A}\cdot C+\overline{B})\cdot(\overline{A}\cdot\overline{B}+\overline{C})\,,
\end{aligned}
$$

where:

- Step (4) is by the absorption identity, $A + A \cdot B = A$, applied 3 times.
- Step (5) is by the distributive property and DeMorgan's Theorem.
- Step (6) is by the absorption property, applied 3 times.

Finally,

$$F(A,B,C) \overset{(7)}{=} \underbrace{\overline{A} \cdot C \cdot \overline{A} \cdot \overline{B}}_{=(\overline{A}\cdot\overline{A})\cdot C\cdot\overline{B}=\overline{A}\cdot C\cdot\overline{B}} + \underbrace{\overline{A} \cdot C \cdot \overline{C}}_{=\overline{A}\cdot(C\cdot\overline{C})=0} + \underbrace{\overline{B} \cdot \overline{A} \cdot \overline{B}}_{=(\overline{B}\cdot\overline{B})\cdot\overline{A}=\overline{B}\cdot\overline{A}} + \overline{B} \cdot \overline{C}$$

$$= \boxed{\overline{A} \cdot \overline{B} \cdot C + \overline{A} \cdot \overline{B} + \overline{B} \cdot \overline{C},}$$

where step (7) is by the distributive property.

Transistors

Motivation for transistors

Our current switching model has some flaws:

- Switching diagrams involve physically disconnecting wires
- We're assuming we have an omnipotent magical hand to control all switches

There's a better way to switch—a *transistor*:

- A switch that is *powered/electronic*
- A switch that is *controllable*—we can make *any logic*
- Very small, maybe even microscopic

History of transistors

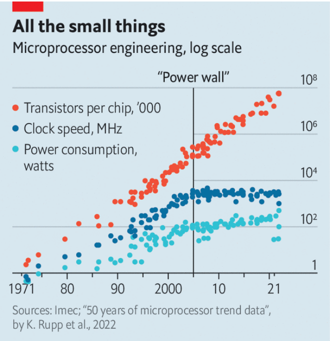
A three-way Nobel Prize!

Transistor lore

- A whole squad invented transistors at **Bell Labs**.
- It was such a massive W that 3 of them got the Nobel Prize
- It was so important that a whole new branch of Quantum Mechanics was invented (surface physics) to explain it.
- Then they got **smaller** and *smaller* and smaller...

Contemporary history of transistors

Moore's Law?



Source: The Economist

Transistors: Where are they now?

- Computer chips can have **billions** of transistors.
- Transistors/chip, speed, power used to ×2/yr: "Moore's Law".
- Nowadays, some claim *Moore's Law is Dead*.. ⋆ cough ⋆ NVIDIA..
- However, we are still fitting more chip in a smaller space. (remember old laptops?)

Is Moore's Law Dead?

Group discussion question

Do you think **Moore's Law** is dead? Why or why not?

Basic electricity

The water analogy and Ohm's law:

What is CMOS?

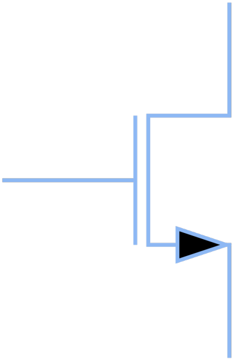
CMOS

A **complementary metal oxide semiconductor transistor** (CMOS) is a 3-terminal device that acts as a voltage-controlled resistance. It is a **powered switch** that can be controlled with a voltage.

In digital logic, a MOS transistor is operated so its resistance is always very high (and the transistor is "off") or very low (and the transistor is "on").

NMOS–Normally open switches

NMOS diagram



Digital logic

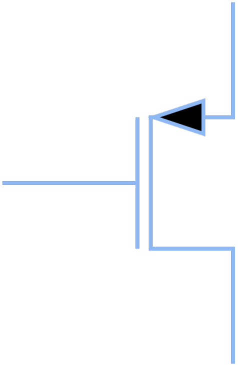
v_{gs}	switch $d \to s$	R_{ds}
high	closed	small
low	open	large

Analog logic

$$R_{ds} = \begin{cases} \text{large } \Omega & v_{gs} \leq 0 \\ \text{small } \Omega & v_{gs} > 0 \end{cases}$$

PMOS–Normally closed switches

PMOS diagram



Digital logic

v_{gs}	switch $d \rightarrow s$	R_{ds}
high	open	large
low	closed	small

Analog logic

$$R_{ds} = \begin{cases} \text{small } \Omega & v_{gs} < 0 \\ \text{large } \Omega & v_{gs} \geq 0 \end{cases}$$

Graphical analysis

NMOS (N-Type)

PMOS (P-Type)

Transistors

Transistors: They are just cool switches.

We can use NMOS and PMOS transistors to design logical operations in real life!

Question: What is switching logic for a NOT gate?

Inverters

What is an inverter

Problem: How to draw a complemented signal $Y = \overline{X}$ in a switching logic circuit?

Solution: Use transistors as your switches, and exploit the fact that they are controllable to perform inversion.

Inverter design example

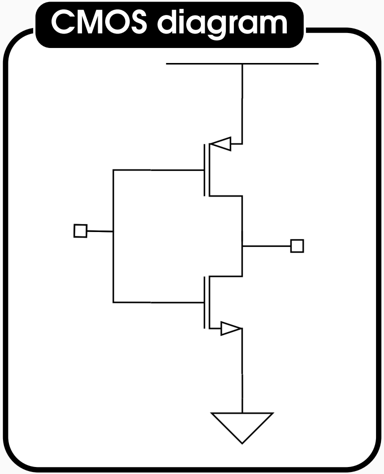
Example

Suppose you want to implement the NOT gate (aka, inverter):

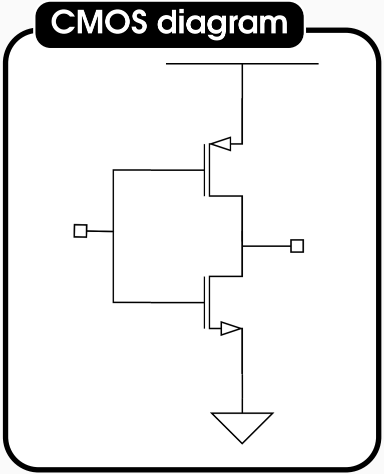
$$v_{\text{out}} = \overline{v_{\text{in}}}$$

with CMOS. How would this be done?

Inverter design example



Inverter design example



Pull up/pull down networks

CMOS logic in general

- **Key idea:** use two complementary CMOS switching networks to implement our desired **low** and **high** outputs:
- **Pull up network (high):** Connects the output to the supply (HIGH) when needed
- **Pull down network (low):** Connects the output to the ground (LOW) when needed.

Graphical explanation of pull up/pull down

Pull-up/pull-down networks

- **Key idea:** use two complementary CMOS switching networks to implement our desired **low** and **high** outputs:
- **Pull up network (high):** Connects the output to the supply (HIGH) when needed
- **Pull down network (low):** Connects the output to the ground (LOW) when needed.

Logic implementation with CMOS

Implementing a NOR gate

Suppose you want to implement the NOR gate in CMOS. How do we do this?

CMOS: NOR

Implementing a generic logic function

Implement the following logic function in CMOS:

$$F = A \cdot \overline{\left(\overline{B} \cdot C\right)}$$

CMOS: Generic logic

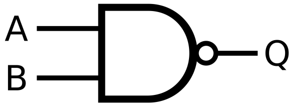
Participation puzzle: implement a CMOS NAND gate

Participation puzzle

Consider our friend, the NAND gate.

- Derive pull-up and pull-down expressions for a NAND gate.
- Construct a CMOS schematic for a NAND gate.
- Indicate the locations of the pull-up and pull-down networks.

Logic gate and truth table



The NAND gate

A	B	$Q = \overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0