University of Illinois at Urbana-Champaign
Dept. of Electrical and Computer Engineering

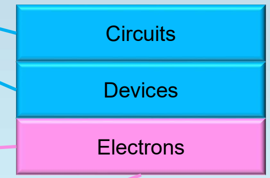
ECE 120: Introduction to Computing

Logic Gates

Today: How Can We Build Gates?

3. Functions on bits (Boolean operators, gates)

4. Implementation?



2. Representations based on bits

1. Two voltage levels → 1 bit

How can we build gates?

But First: Check Out My New Invention!

Last night I had a great idea.

I call it a **"torch."**

At night, you can **point it at things**.

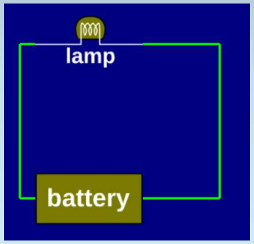
And **they will be lit up**.

Anything!

Your car or bike.

Your door lock.

A friend.



What do you think?

You Think I Should Do What?

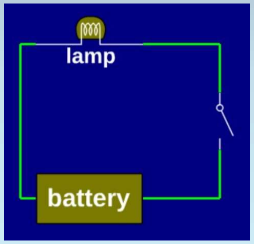
Like this?

I think **people already make those**.

The switch is **controlled by your thumb**.

They call it a **flashlight**.

I won't be able to patent it.



Don't Worry: Here's Another Idea

So, you like switches?

Let's put a bunch of switches together.

Each controlled by ~~our~~ your thumbs.

When we want **to change a bit**,

we will just **flip a switch**!

We'll call it a **hand-operated computer**!

We'll need about **2,000,000,000** switches.

What do you think?

Still Don't Like It? One Last Try…

What if we develop a **voltage-controlled switch**?

Then **one switch**

- can **control another switch,**
- which can **control a third switch,**
- **and so on**!

Instead of using **your thumbs**, we can **build circuits with 2,000,000,000 switches**!

Now THAT's a really cool idea!

Let's Take a Bragging Break

John Bardeen, 1908-1991

1947: **invented transistor** at Bell Labs with Shockley & Brattain

1951: joined **Illinois ECE faculty** (and Physics)

1956: **Nobel Prize, Physics**

1972: second **Nobel Prize, Physics**, for Bardeen-Cooper-Schrieffer **(BCS) theory of superconductivity**

Bardeen's First Ph.D. Student (1954)

Nick Holonyak, Jr., 1928-

1962: invented **visible light LED** at GE

1963: joined **Illinois ECE faculty**

(also invented laser diodes for **CDs/DVDs**, **dimmer switches**, and more)

1973: **National Academy of Engineering**

2003: **National Medal of Technology**

2008: **National Inventors Hall of Fame**

(among many other awards)

Holonyak's First(?) Ph.D. Student (1967)

Greg Stillman, 1936-1999

1975: joined **Illinois ECE faculty**

invented **avalanche photodiodes**
(for amplifying small photon sources),
among many other things

1985: **National Academy of Engineering**

1985-1987: **Founding Director of MNTL**
(the Micro- and Nano-Technology Lab)

Stillman's First Ph.D. Student (1979)

Milton Feng, 1950-

1991: joined **Illinois ECE faculty**

2003: invented **Terahertz transistors**

Jan 2004: invented **light-emitting transistor** (with Nick!)

Nov 2004: invented **transistor laser**
(also with Nick!)

2016: just retired…

But Not Just Faculty!

Jack Kilby, 1923-2005

1947: **BSEE from Illinois**

1958-59: invented **integrated circuit** at TI

(also invented the **thermal printer**
and the **handheld calculator**)

1967: **National Academy of Engineering**

2000: **Nobel Prize, Physics**

(See why we expect a lot of you?)

Digital Electronics is Based on MOSFETs

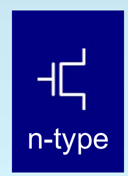
Digital electronics today uses MOSFETs.

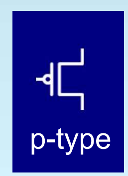
- the material: Metal-Oxide Semiconductors
- the mechanism: Field-Effect Transistors (electric field/voltage-controlled)

There are two kinds, named after the charge carrier,

- **n**(egative)**-type**, and
- **p**(ositive)**-type**,

drawn as shown here.



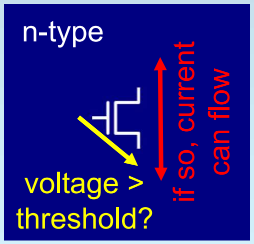


n-type is On With Positive Gate to Source/Drain Voltage

An **n-type MOSFET**

- turns **on** (switch is **closed**, allowing current to flow)
- if the **voltage from gate** (left terminal) **to other terminals exceeds a threshold**

If the voltage is smaller, the transistor is **off** (the switch is **open**).



Our Voltages Will Be Binary

We need two voltages:

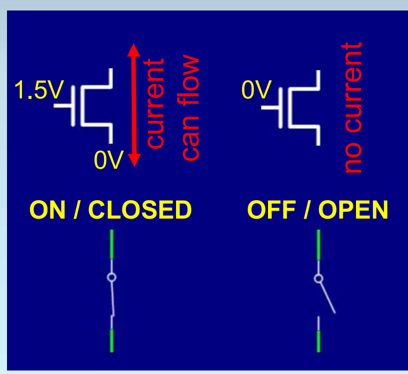
- **0V**, a ground
  (this is the binary 0 value)
- $\mathbf{V_{dd}}$, around **1.5V**, high voltage*
  (this is the binary 1 value)

*Used to be 5V, but has been decreasing for decades. The rate of decrease is now slowing down.

Use Binary Voltages to Control n-Type MOSFETs

n-type only **turns on when gate voltage is high** ($\mathbf{V_{dd}}$).

An **n-type** can pull one terminal down to **0V** with the other terminal.

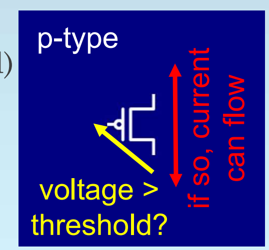


p-type is On With Negative Gate to Source/Drain Voltage

A **p-type MOSFET**

- turns **on** (switch is **closed**, allowing current to flow)
- if the **voltage from other terminals to the gate** (left terminal) **exceeds a threshold**

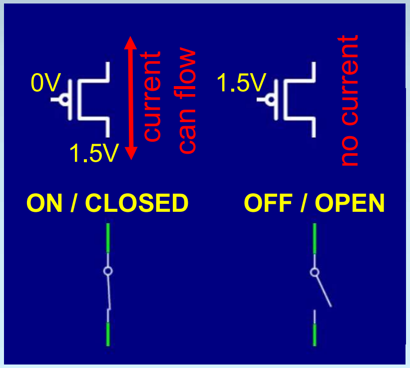
If the voltage is smaller, the transistor is **off** (the switch is **open**).



Use Binary Voltages to Control p-Type MOSFETs

p-type only **turns on when gate voltage is low** (**0V**).

A **p-type** can pull one terminal up to $\mathbf{V_{dd}}$ with the other terminal.



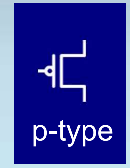
The Drawings Help You Remember How They Work

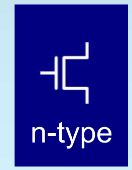
Notice the use of the inverter bubble on the **p-type**.

Use it to help you remember:

- **p-type turns on with low voltage** (**0V**, or binary **0**).
- **n-type turns on with high voltage** (**Vdd**, or binary **1**).

The names may not be so helpful (again, they refer to charge carriers).





Gates are Based on Complementary MOS (CMOS)

So how do we build gates?

Gates use complementary structures of **p-type** and **n-type MOSFETs**.

Each gate uses an equal number of each type.

For that reason, we say that

- most **digital systems are based on CMOS**,
- or Complementary MOS.

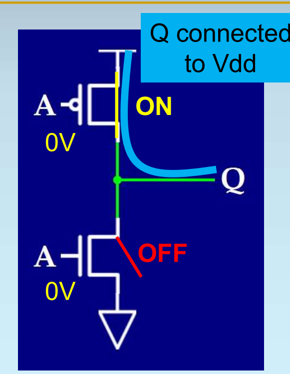
What Does This Gate Do? (when A=0V)

Here is the simplest gate.

What does it do?

Let's write a truth table!

A	Q
0V	1.5V
1.5V	



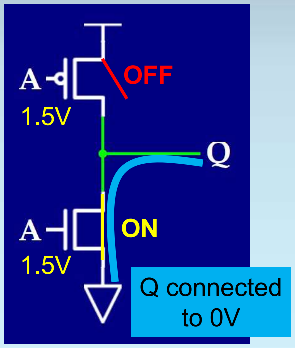
What Does This Gate Do? (when A=1.5V)

Here is the simplest gate.

What does it do?

Let's write a truth table!

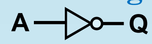
A	Q
0V	1.5V
1.5V	0V



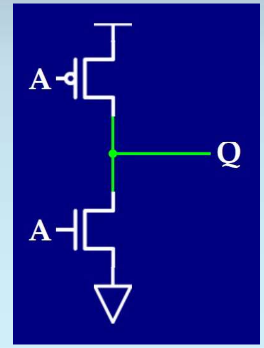
It's a NOT Gate!

Now convert the truth table from voltages to binary.

It's a NOT gate!



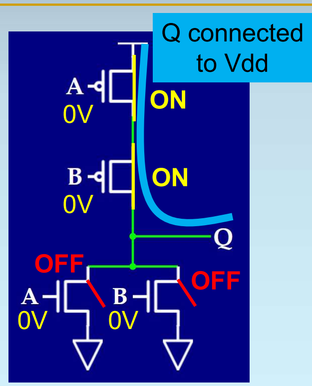
	A	Q	
(0)	0V	1.5V	(1)
(1)	1.5V	0V	(0)



Let's Analyze Another Structure

What about this structure?

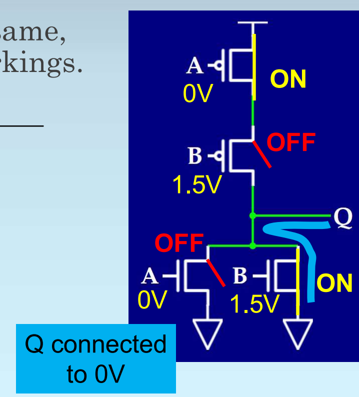
A	B	Q
0	0	1
0	1	
1	0	
1	1	



Next, Assume A = 0 and B = 1

The A value is the same, so we leave the markings.

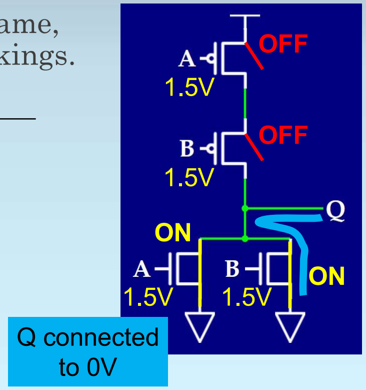
A	B	Q
0	0	1
0	1	0
1	0	
1	1	



Next, Assume A = 1 and B = 1 (BOTTOM LINE!)

The B value is the same,
so we leave the markings.

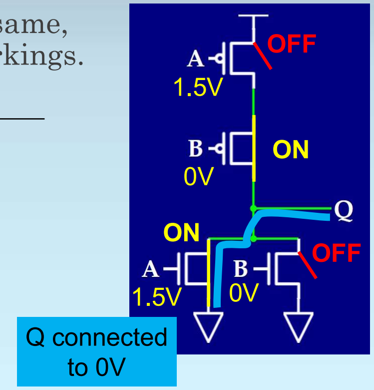
A	B	Q
0	0	1
0	1	0
1	0	
1	1	0



Finally, Assume A = 1 and B = 0

The A value is the same,
so we leave the markings.

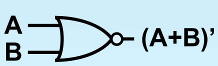
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0



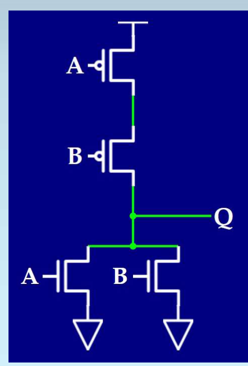
It's a NOR Gate!

We see that **Q = (A+B)'**.

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0



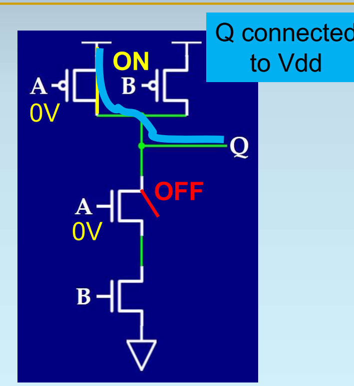
NOR gate



And Just One More to Analyze…

What if A=0?

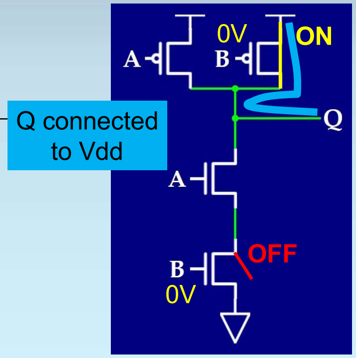
A	B	Q
0	0	1
0	1	1
1	0	
1	1	



Notice that the Circuit is Symmetric in A and B

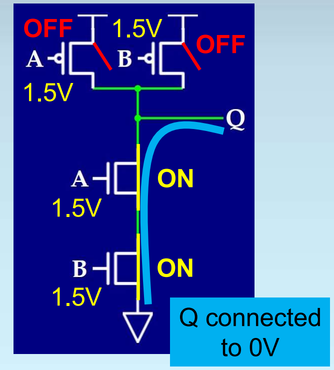
What if B=0?

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	



And if Both A = 1 and B = 1?

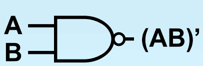
A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0



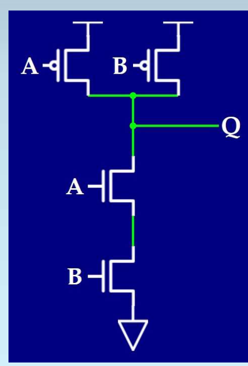
It's a NAND Gate!

We see that **Q = (AB)'**.

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0



NAND gate



Generalizing to More Inputs

Notice the common features
- p-type always connected to Vdd.
- n-type always connected to 0V.
- One side is parallel, the other is serial (they are duals* of one another).

Can you generalize NAND/NOR to more inputs?

Let's try it in the online tool…

*See Notes Section 2.2.1.

A Couple of Practical Limits

Gates scale to about 4 inputs before using more gates is a better approach.

One can easily

- design an AND or an OR gate with CMOS
- by swapping n-type with p-type,
- but MOSFETs don't work properly in those designs.
- Try it in the online tool to see what happens.
- (NAND followed by NOT is, of course, AND.)