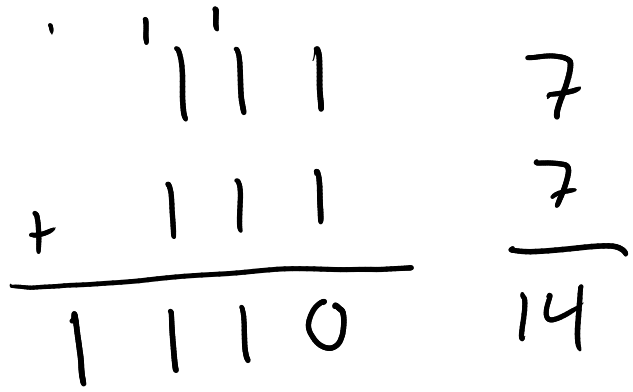
Warmup



ES 4: Boolean equations and logic gates

Steven Bell

5 September 2024

Tufts UNIVERSITY

Nervous about

- Workload / time needed for solid understanding
  "I've never taken this many credits before..."
- Background / content being new
  "This course was required for my major, but I know very little about it..."
- Building circuits
  "I've never worked with circuit hardware before..."

From the reading check

- What good are sum-of-products and product-of-sums?
- minterms, pls explain.
- How do I tell if something is combinational?
- How do I go about memorizing the difference between NOT, AND, OR, XOR, NAND, NOR, XNOR?

By the end of class today, you should be able to:

- Write truth tables for AND, OR, XOR, NOT, NAND, NOR, XNOR
- Write a boolean equation from an English description and vice-versa
- Draw a logic diagram from a boolean equation and vice-versa
- Write a truth table from a boolean equation and vice-versa

Key representations

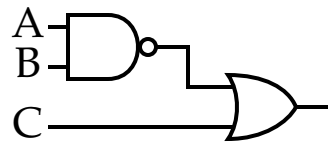
Boolean equation

$$\overline{AB} + C$$

Truth table

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1

Logic diagram



Key representations

English description

Not both apples and bananas...

Code

```
!(A&&B) || C
```

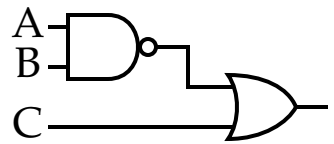
Boolean equation

$$\overline{AB} + C$$

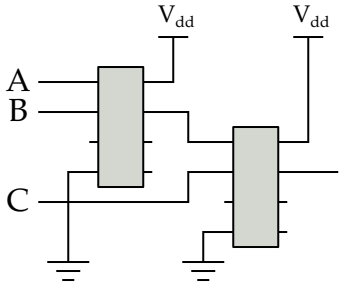
Truth table

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1

Logic diagram

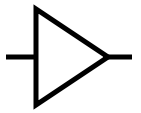
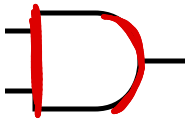
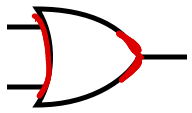
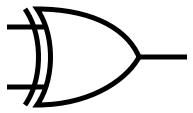
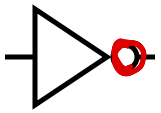
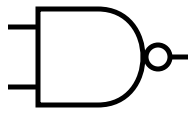
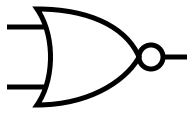
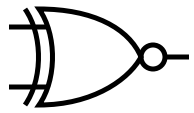


Circuit diagram



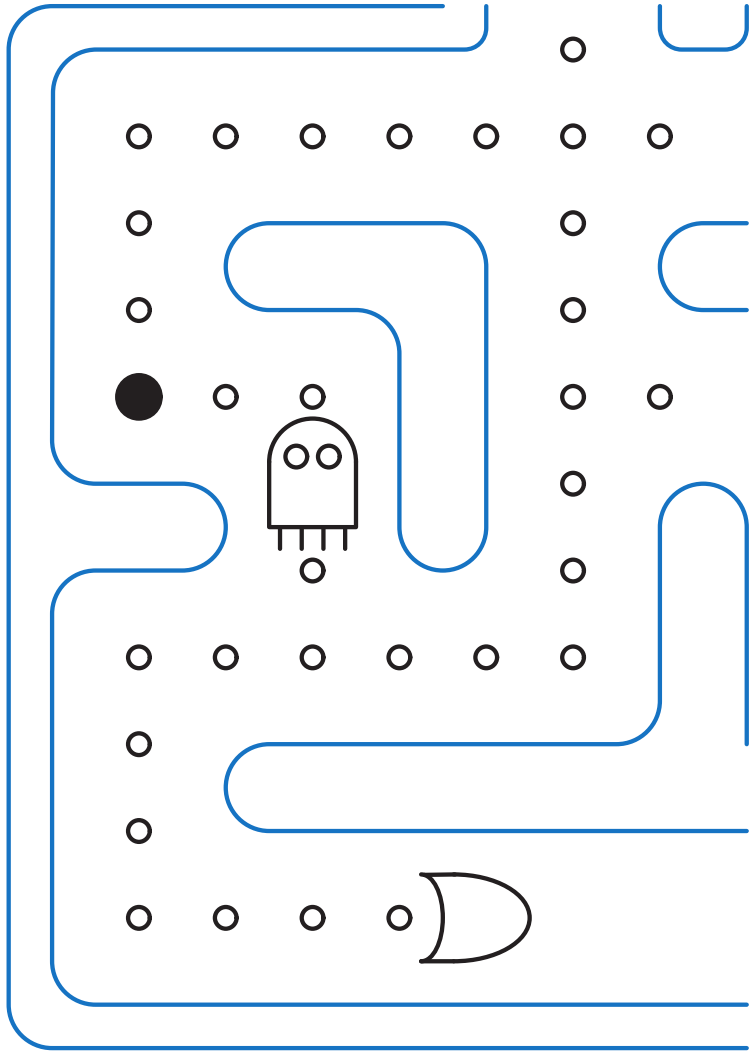
NOT

Inverted

Name	Buffer	AND	OR	XOR	Inverter	NAND	NOR	XNOR
Logic symbol								
Logic equation	A	AB	$A+B$	$A \oplus B$	$\overline{A}$	$\overline{AB}$	$\overline{A+B}$	$\overline{A \oplus B}$

Xoplus

Truth table								
	A Y 0 0 **1 1**	A B Y 0 0 0 0 1 0 1 0 0 **1 1 1**	A B Y 0 0 0 **0 1 1** **1 0 1** **1 1 1**	A B Y 0 0 0 **0 1 1** **1 0 1** 1 1 0	A Y **0 1** 1 0	A B Y **0 0 1** **0 1 1** **1 0 1** 1 1 0	A B Y **0 0 1** 0 1 0 1 0 0 1 1 0	A B Y **0 0 1** 0 1 0 1 0 0 **1 1 1**
C++ equivalent	`A`	`A&&B`	`A||B`	`A!=B`	`!A`	`!(A&&B)`	`!(A||B)`	`A==B`
74-series IC	7407	7408	7432	7486	7404	7400	7402	various



Pacman is like an OR gate...?


STAR TREK


Y
STOR TREK
A
B

Bigger equations

We'll use a **bar** to indicate inversion: $\overline{A+B}$

You might also see "¬" for inversion: $\neg(A+B)$

AND takes precedence over OR:

$$AB + C = (AB) + C$$

(Think multiplication over addition)

$A(B+C)$

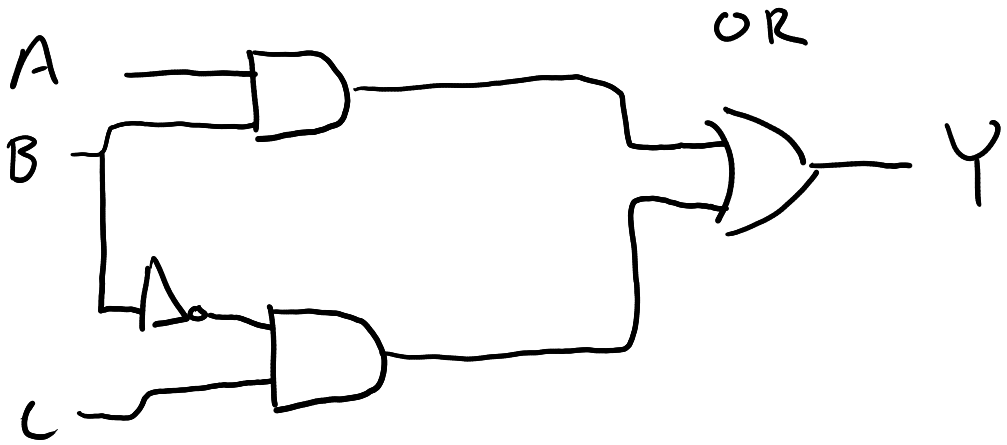
Set theory uses ∧ and ∨ for AND and OR respectively.

Drawing logic diagrams

Elements are objects

Nodes are the interconnections

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

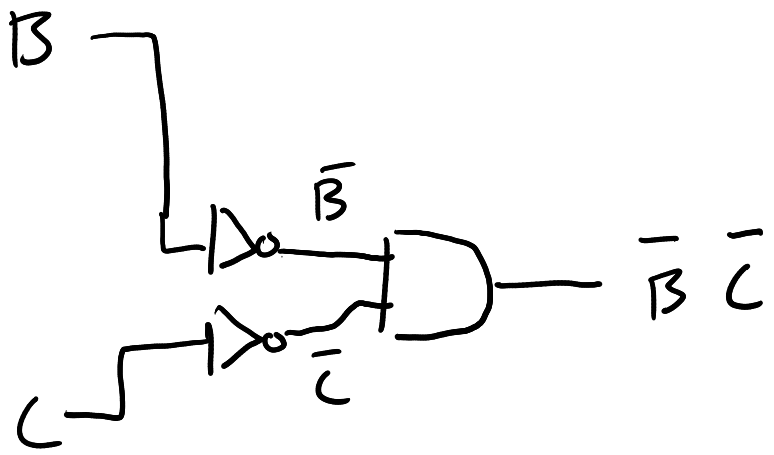


Drawing logic diagrams

Elements are objects

Nodes are the interconnections

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

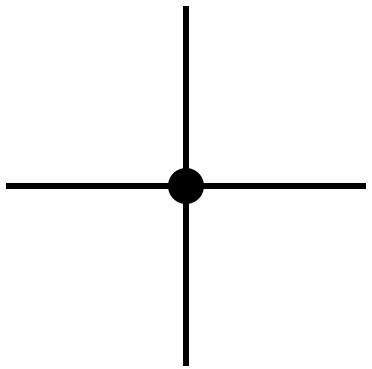


NAND

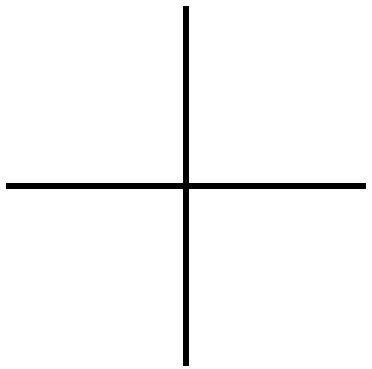
B	C	$\overline{B}$	$\overline{C}$	$\overline{B}\overline{C}$	$\overline{BC}$
0	0	1	1	1	1
0	1	1	0	0	1
1	0	0	1	0	1
1	1	0	0	0	0

Logic diagram connections

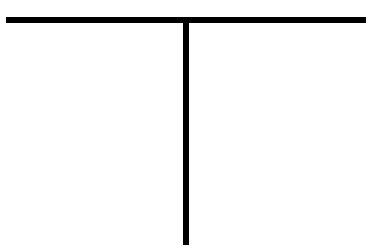
Wires are connected if there's a dot:



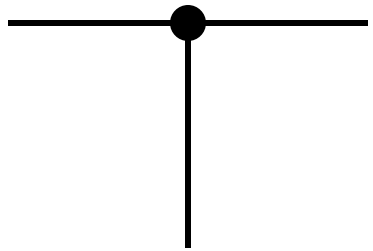
connected



not connected



connected



connected
(dot is only for clarity)

What is combinational?

A circuit is combinational if:

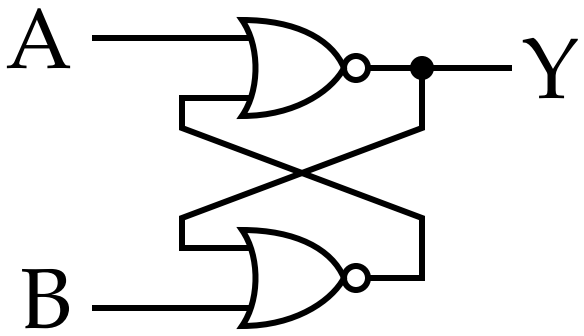
1. It is a discrete logic gate, or
2. It is composed of combinational elements such that:

    There are no cycles, and

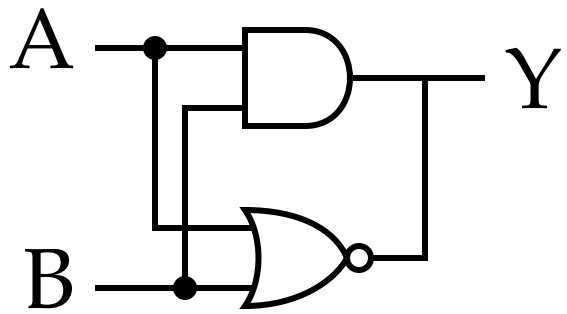
    Every node (wire) is only driven by one gate

Which of the circuits below are combinational?

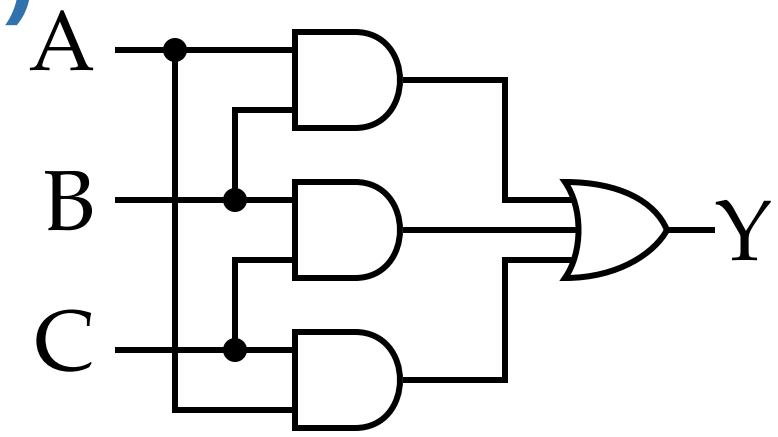
(1)



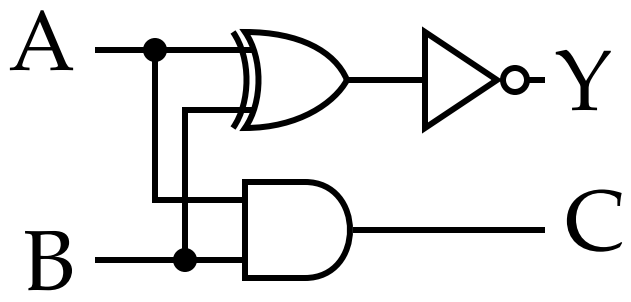
(2)



(3)



(4)



Respond at **pollev.com/stevenbell**

Who cares, anyway?

In a combinational circuit,
outputs are a function of **only the current inputs**

Feedback loops can break this!

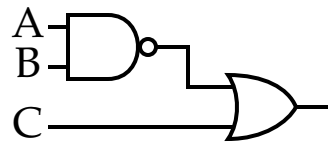
Driving one wire with multiple gates breaks the digital abstraction!

Translating representations

Boolean equation

$$\overline{AB} + C$$

Logic diagram



Truth table

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1

Boolean equation ↔ logic diagram

Replace operations with gates, following precedence

Replace gates with operations, nesting to create precedence

Translating representations

Boolean equation

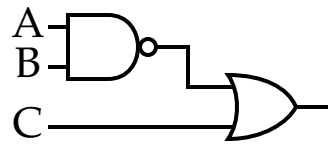
$$\overline{AB} + C$$

Evaluate

Truth table

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1

Logic diagram



Boolean equation to truth table

$$Y = AB + \overline{C}$$

A	B	C	AB	Y
0	0	0	0	1
0	0	1	0	0
0	1	0	0	1
0	1	1	0	0
1	0	0	0	1
1	0	1	0	0
1	1	0	1	1
1	1	1	1	1

Translating representations

Boolean equation

$\overline{AB} + C$

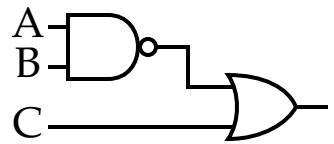
Evaluate

Standard forms

Truth table

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1

Logic diagram



Truth table to boolean equation

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

$$\bar{A}\,\bar{B}\,\bar{C} + \bar{A}\,\bar{B}\,C + A\,\bar{B}\,\bar{C} = Y$$

Truth table to boolean equation

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

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

1 1 1 1 1 1 1 1 0

Truth table to boolean equation

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

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

1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1

Translating representations

Self-explanatory (hopefully!)

Code

```
!(A&&B) || C
```

English description

Not both apples and bananas...

Boolean equation

$\overline{AB} + C$

Evaluate

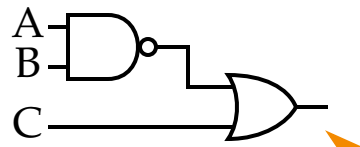
Standard forms

Truth table

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1

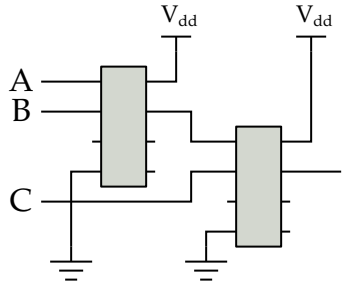
Next week!

Logic diagram



Prelab #2!

Circuit diagram



For next week

1. Read the book (2.3-2.7) and complete the pre-class reading check

    Due at **11 AM** the day of class, so I can review it

2. Welcome survey posted as soon as I finalize lab times

3. Plan to come to lab next week!

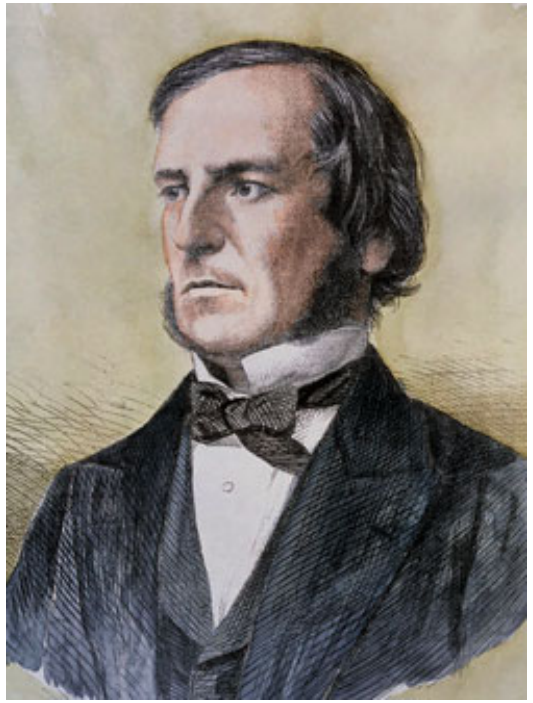
Bonus material

Backup slides, extra information, or just stuff
I had to cut out of the lecture for the sake of time.

A useful circuit

We have two 2-bit numbers. Design a circuit to calculate A > B.

Boolean or boolean?



George Boole
(1815-1864)

(Photo from Wikimedia)

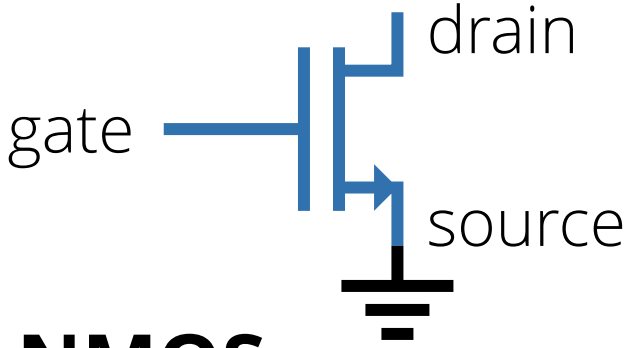
Why is NAND a big deal?

For the purposes of this course,
MOS transistors are electrically-controlled **switches**.

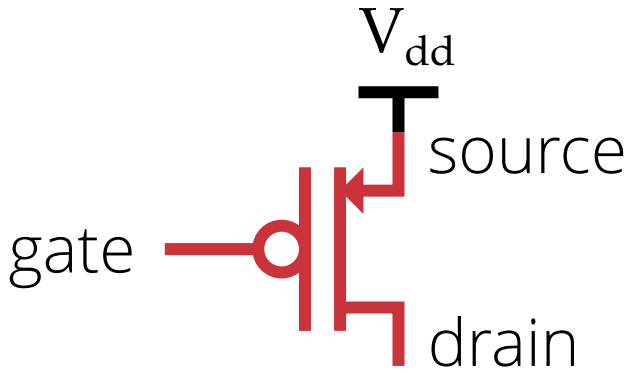
They have one terminal that controls the "switch", called the **gate**.

When the transistor is on, current can flow between the other terminals, called the **source** and **drain**.

They come in two types:

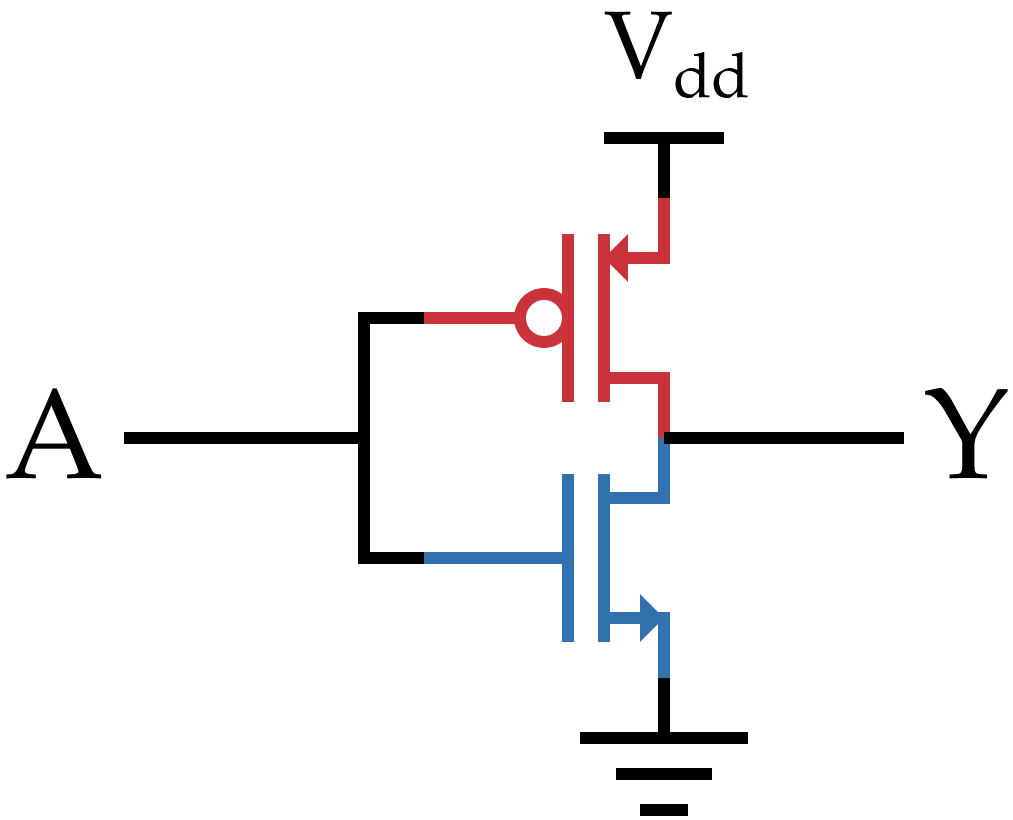


NMOS



PMOS

A simple gate



PMOS: "closed" when gate is **low**.

NMOS: "closed" when gate is **high**.

A more complex gate