ECE 550DK
Fundamentals of Computer Systems and Engineering

Fall 2023

From Transistors to Gates

Xin Li & Dawei Liu
Duke Kunshan University

Slides are derived from work by
Andrew Hilton, Tyler Bletsch and Rabih Younes (Duke)

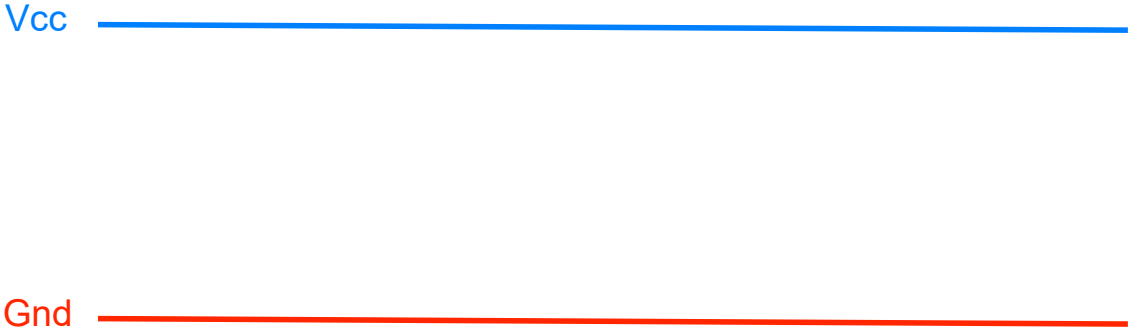
Last time….

(Almost) every class will start with the same question:

- Who can remind us what we talked about last time?
  (besides course policies)

- Abstraction
  - Interface vs Implementation

Power (Vcc) and Ground (Gnd)

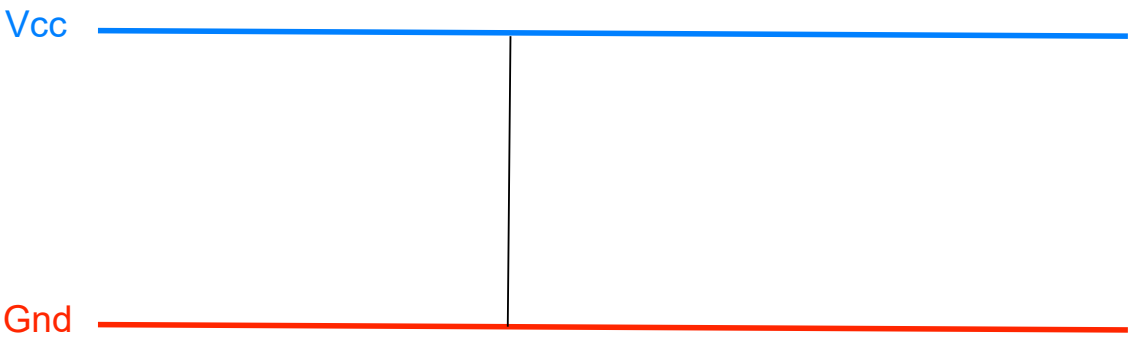


- Two supply rails:
  - Power (aka Vcc, sometimes called Vdd) , e.g., +1.0 V
    - Logically, 1
  - Ground (Gnd, or Vss), e.g., 0 V
    - Logically, 0
  - Use Vcc/Gnd because that's what Quartus uses

Wires

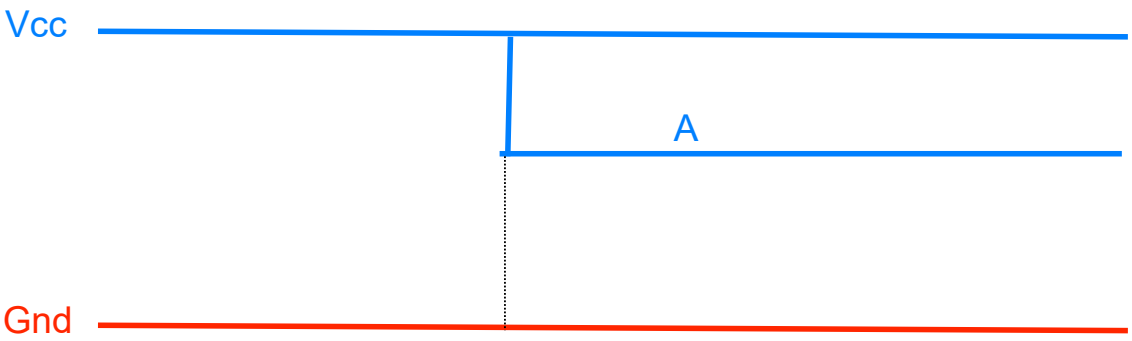
- A wire (or other conductor) causes current to flow

Short circuit



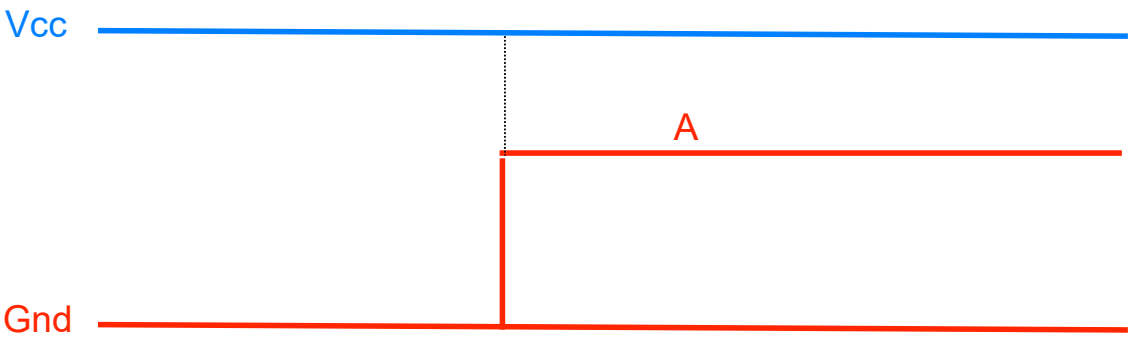
- Short circuit: direct connection from power to ground
  - Very high current
  - Generates a lot of heat
  - Destroys your chip
  - Very bad!

Switching



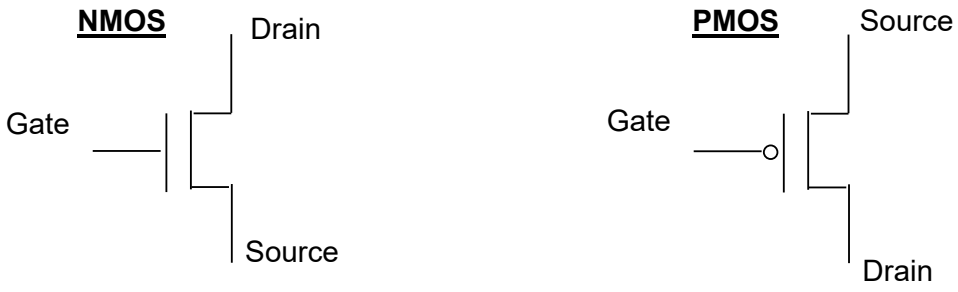
- Suppose instead we had some sort of switch
  - Here, top connection conducts, connecting A to Vcc
  - The bottom half resists insulating A from Gnd

Switching



- If we switch our connection…
  - Current flows as A changes voltage levels
  - Connection to power is closed, so no short circuit

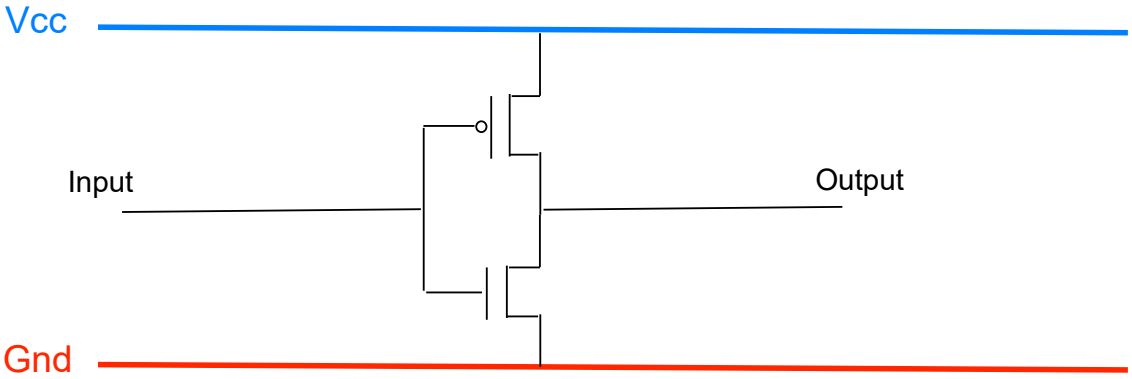
Transistors: Electrically controlled switches



- Two types:
  - NMOS (left: no circle):
    - Conducts when gate is 1, resists when gate is 0
  - PMOS (right: circle):
    - Conducts when gate is 0, resists when gate is 1

PMOS = “P-type Metal Oxide Semiconductor”, NMOS = “N-type Metal Oxide Semiconductor”

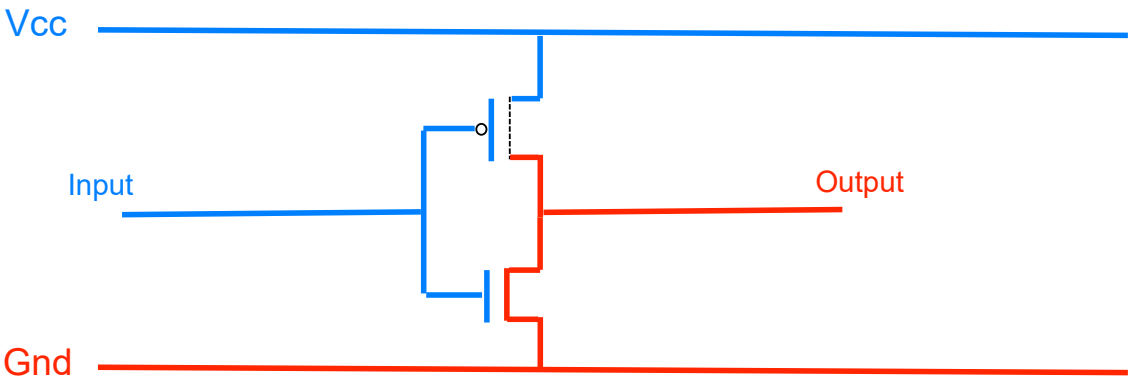
CMOS: Complementary MOS



- CMOS (most common, all we care about):
  - Put PMOS and NMOS in complementary fashion
    - Either PMOS conducts or NMOS conducts, but not both
- Form a logic gate
  - Input (s): connected to gates of transistors
  - Output: connected to drains

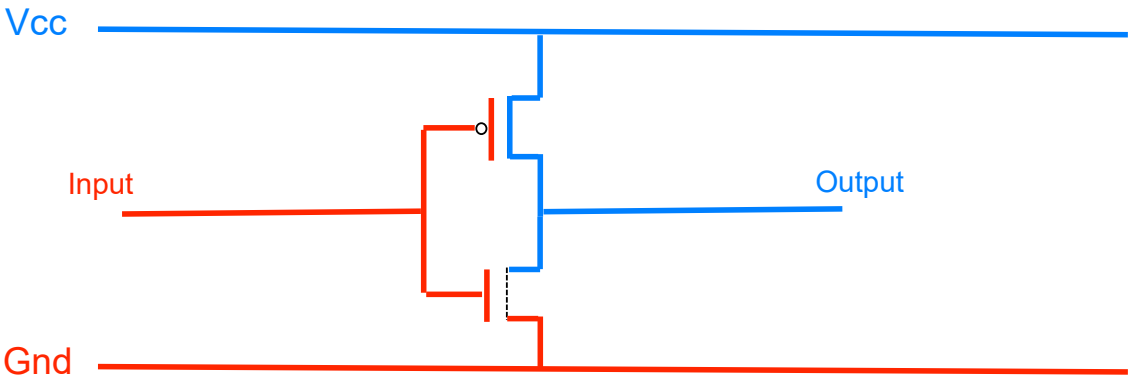
CMOS = "Complementary Metal Oxide Semiconductor"

CMOS: Complementary MOS



- Let's see how this works.
  - Suppose Input = 1 (circuitry to control input, not shown)
  - PMOS transistor resists
    - No connection between Output and Vcc
  - NMOS transistor conducts
    - Connection between Output and Gnd
  - Output is 0

CMOS: Complementary MOS

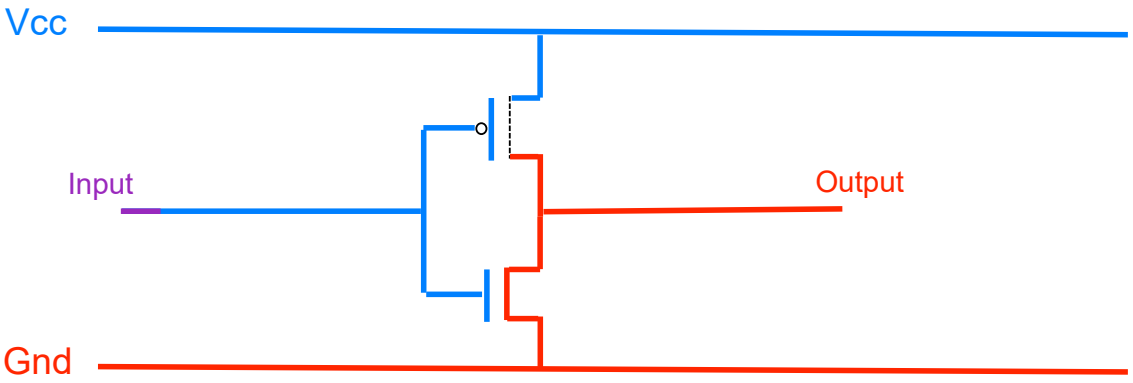


- Now suppose Input changes to 0
  - PMOS transistor conducts
    - Connection between Output and Vcc
  - NMOS transistor conducts
    - No Connection between Output and Gnd
  - Output is 1

Switching delays

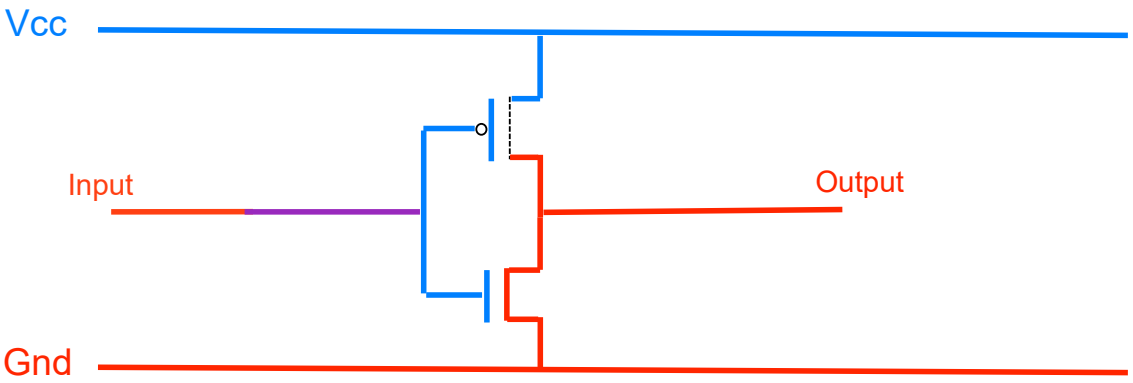
- Note: this doesn't happen instantly
  - There is some delay as these change
  - Factors that affect the delay
    - Voltage
      - Higher voltage = faster switching, but more power/energy
    - Resistance
      - Lower resistance = faster switching
    - Capacitance
      - Lower capacitance = faster switching
  - Calculating delay = hard, so we let our tools do it

CMOS: Complementary MOS



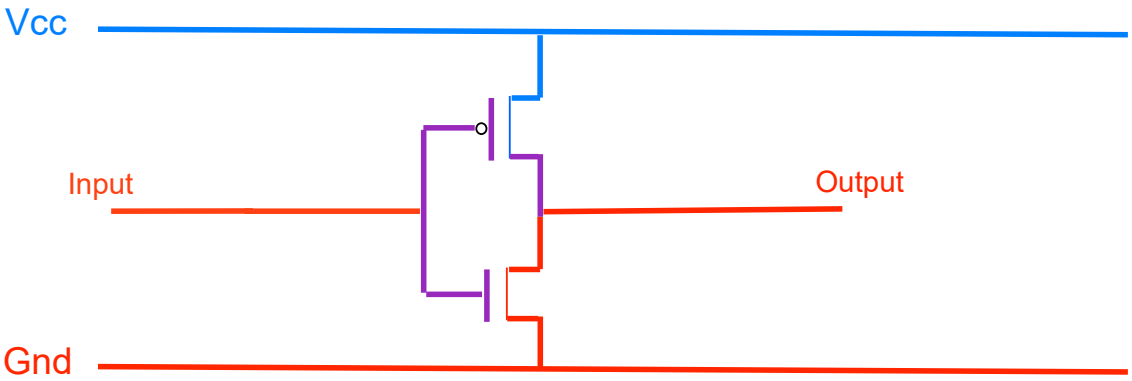
- Slightly more accurate with respect to time
  - Input starts to swing from 1 to 0 (not instant)

CMOS: Complementary MOS



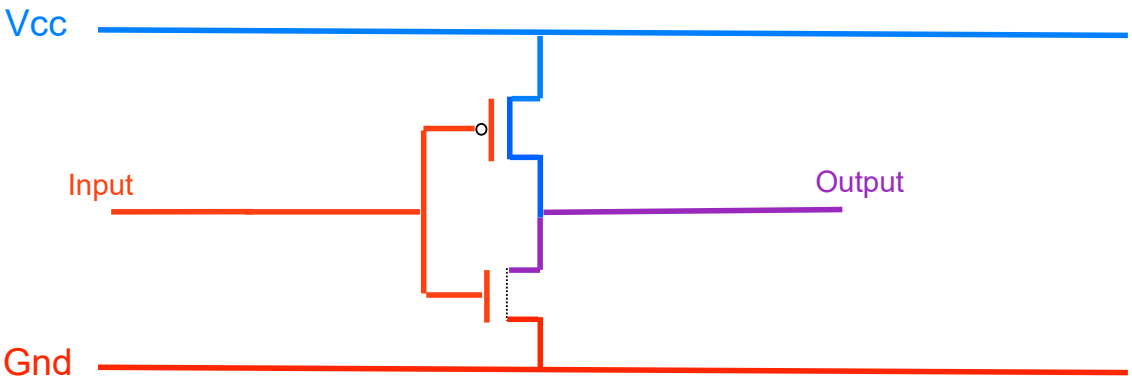
- Slightly more accurate with respect to time
  - Input starts to swing from 1 to 0 (not instant)
  - Change propagates along wires (also takes time)

CMOS: Complementary MOS



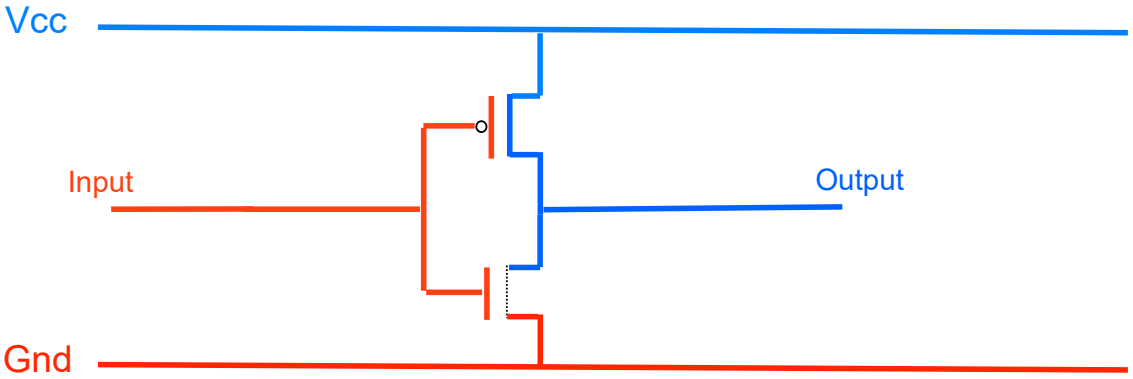
- Slightly more accurate with respect to time
  - Input starts to swing from 1 to 0 (not instant)
  - Change propagates along wires (also takes time)
  - Transistors start to switch (partially conductive)

CMOS: Complementary MOS



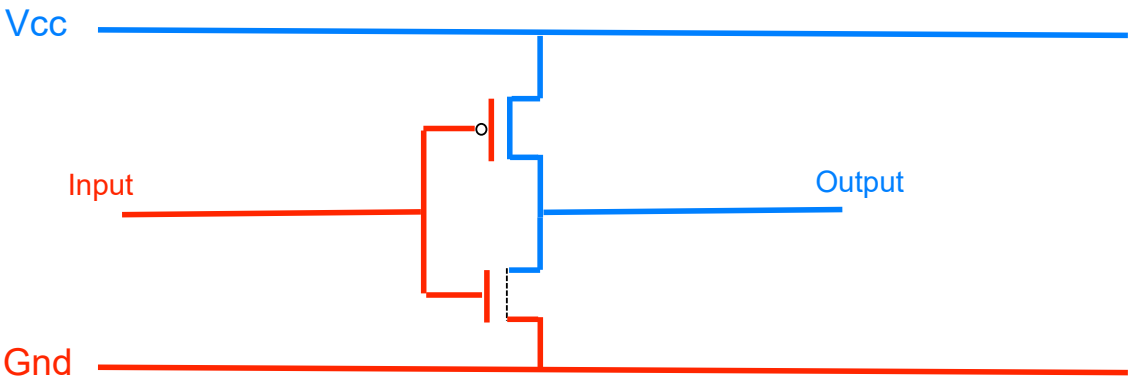
- Slightly more accurate with respect to time
  - Input starts to swing from 1 to 0 (not instant)
  - Change propagates along wires (also takes time)
  - Transistors start to switch (partially conductive)
  - Inputs reach 0 (sometime)
    - Transistors fully open/closed
    - Output may take time to transition

CMOS: Complementary MOS

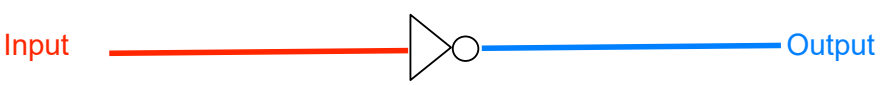


- Slightly more accurate with respect to time
  - Input starts to swing from 1 to 0 (not instant)
  - Change propagates along wires (also takes time)
  - Transistors start to switch (partially conductive)
  - Inputs reach 0 (sometime)
    - Transistors fully open/closed
    - Output may take time to transition

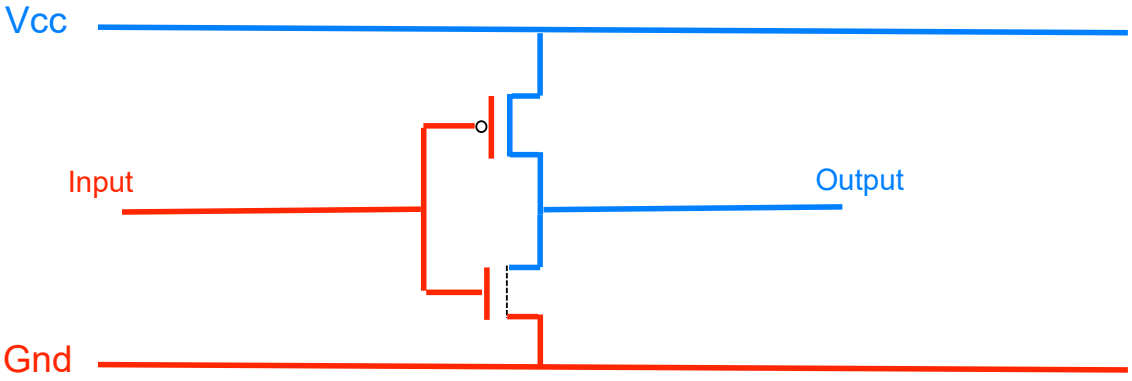
Our first logic gate: The inverter



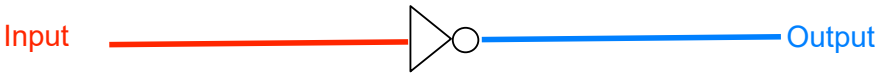
- This circuit is a logic gate: inverter or "NOT gate"
  - Gives logical negation of its input
    - Input = 0, Output = 1
    - Input = 1, Output = 0
  - Typically, just draw the gate, instead of the transistors:



Our first logic gate: The inverter



- (Small) Example of abstraction
  - Interface: "do logical negation"
  - Implementation: how to hook up the transistors



Let's build a more interesting gate

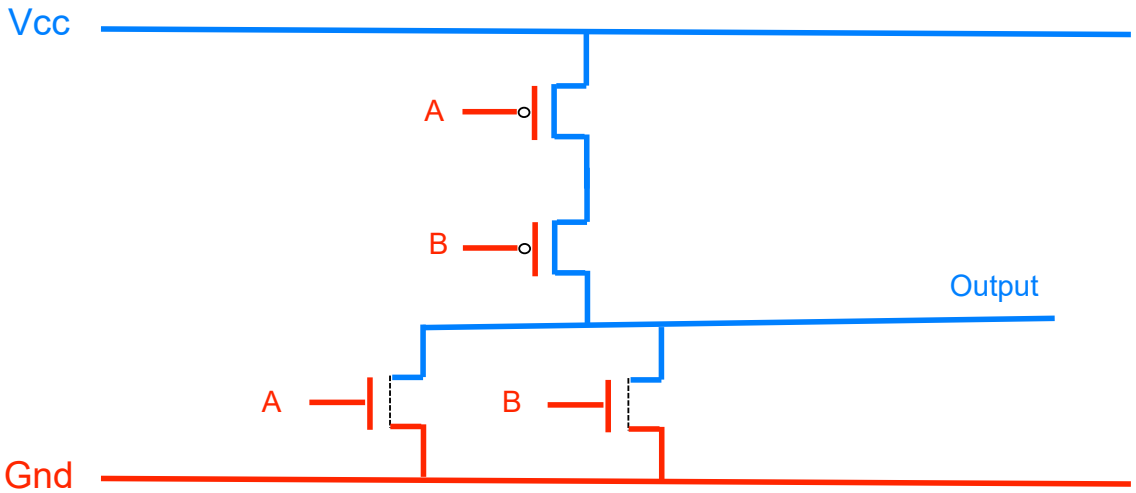
- Next, let us build a 2-input NOR gate
  - Here is a **truth table** for NOR
    - Shows output values for all possible inputs
  - Output =1 when A and B = 0
    - Connect PMOS in **series**
    - (Not A) and (Not B)
  - Output = 0 when A or B = 1
    - Connect NMOS in **parallel**
    - Not (A or B)

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

Note: two formulas are logically equivalent (DeMorgan's Laws)

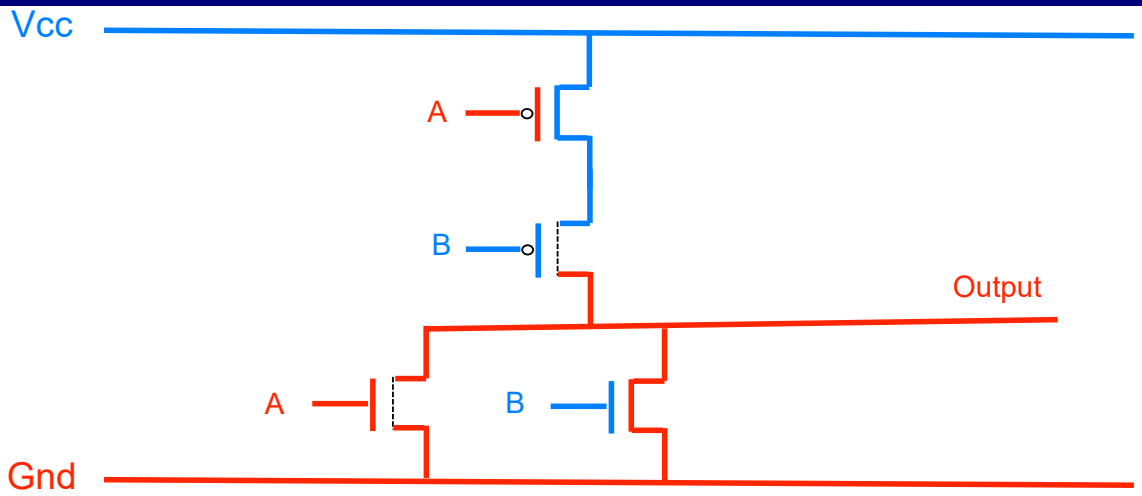
- PMOS formula has NOTs on inputs
- NMOS formula has NOT on output

The NOR gate



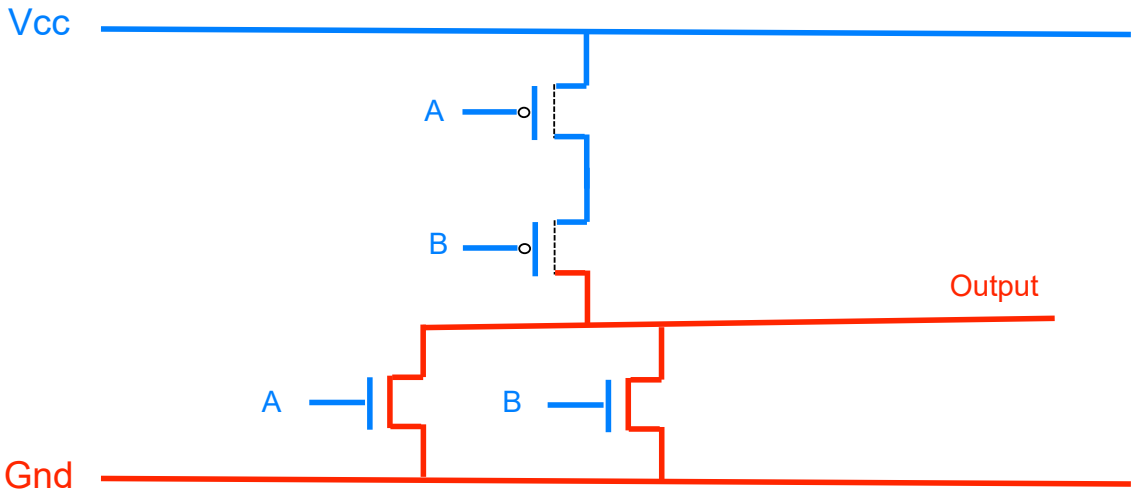
- NOR Gate
  - PMOS in series (both A and B must be 0 to get 1)
  - NMOS in parallel (either A or B at 1 results in 0)
- Side note: real chips have several layers to route wires
  - 3D drawing is hard, just label inputs

The NOR gate

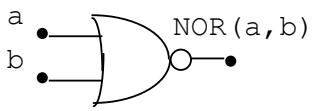


- NOR Gate
  - Same gate, just changed B's value to 1
  - Now output = 0
  - PMOS connected to B resists, blocking connection to Vcc
  - NMOS connected to B conducts, forming connection go Gnd

The NOR gate



- NOR Gate
  - Same gate, now change A to 1
  - Output stays at 0
  - Two connections to Gnd, but that's fine

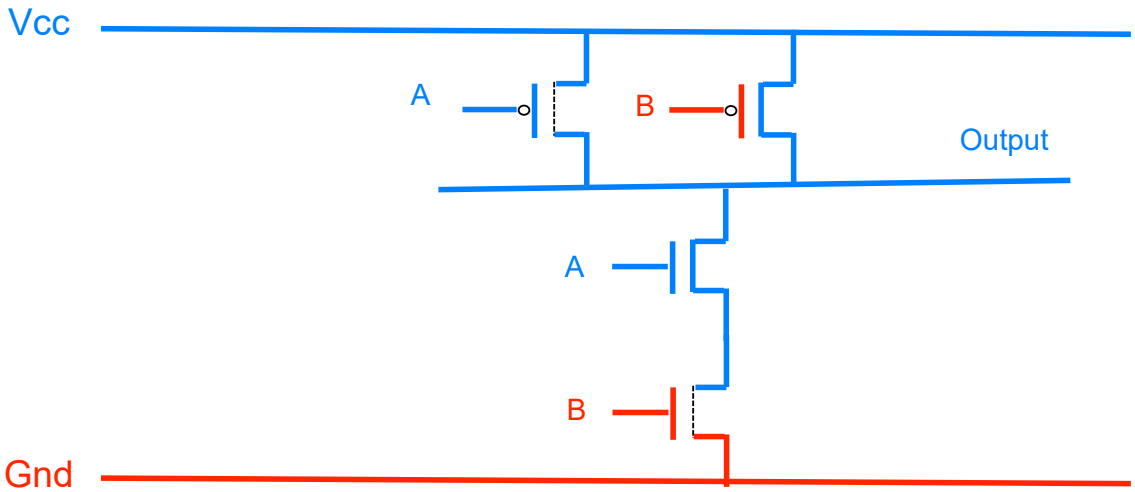


Let's build a more interesting gate

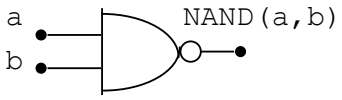
- I'll let you all try a 2-input NAND gate
  - Here is a **truth table** for NAND

A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

The NAND Gate

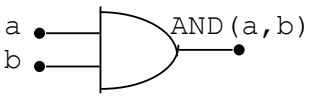


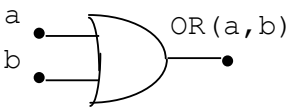
- The NAND Gate
  - PMOS in parallel (either at 0 results in a 1)
  - NMOS in series (both at 1 results in 0)

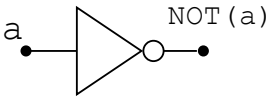


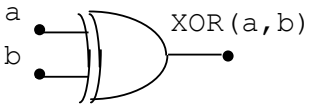
Boolean Gates

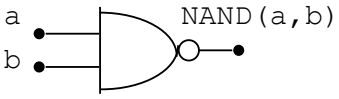
- Actually a bunch of standard logic gates:

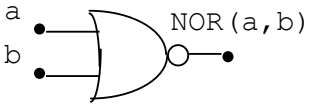


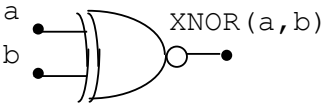








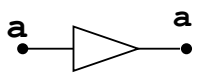




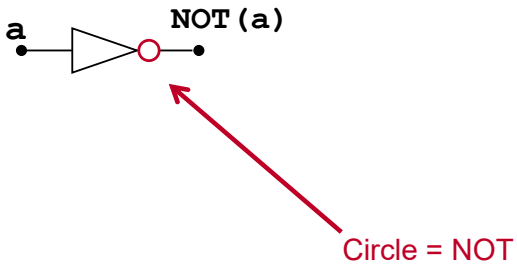
How to keep them all straight?

Guide to Remembering your Gates

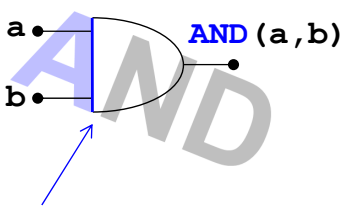
- This one looks like it just points its input where to go
  - It just produces its input as its output
  - Called a buffer



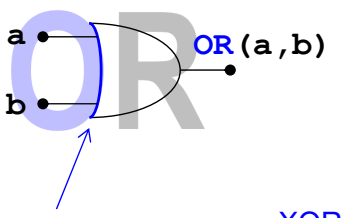
- A circle always means negate (invert)



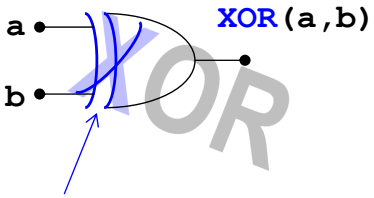
Guide to Remembering your Gates



Straight like an A

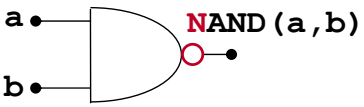


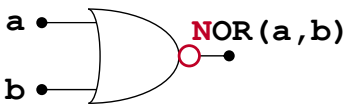
Curved, like an O

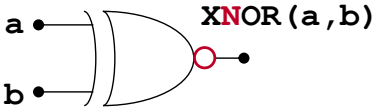


XOR looks like OR (curved line), but has two lines (like an X does)

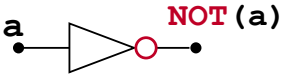
Circle means NOT







(XNOR is 1-bit "equals" by the way)



Next Time…

- Next time, we'll delve into Combinatorial Logic
  - Putting gates together to do useful things