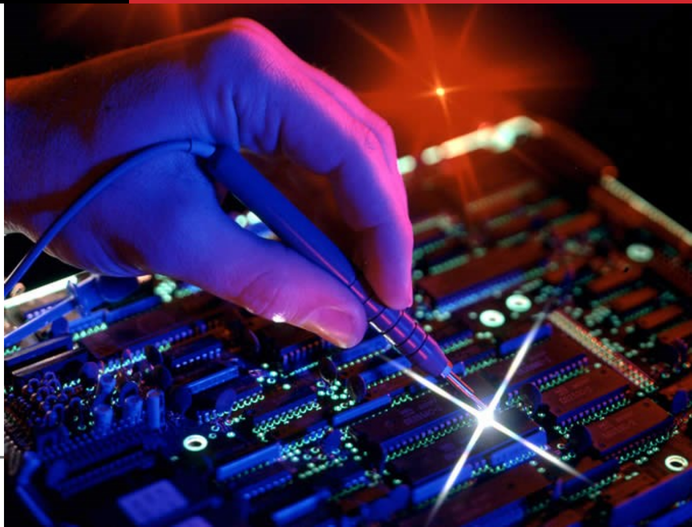
Advanced Computer Architecture

TRANSISTORS TO LOGIC GATES

Dennis A. N. Gookyi

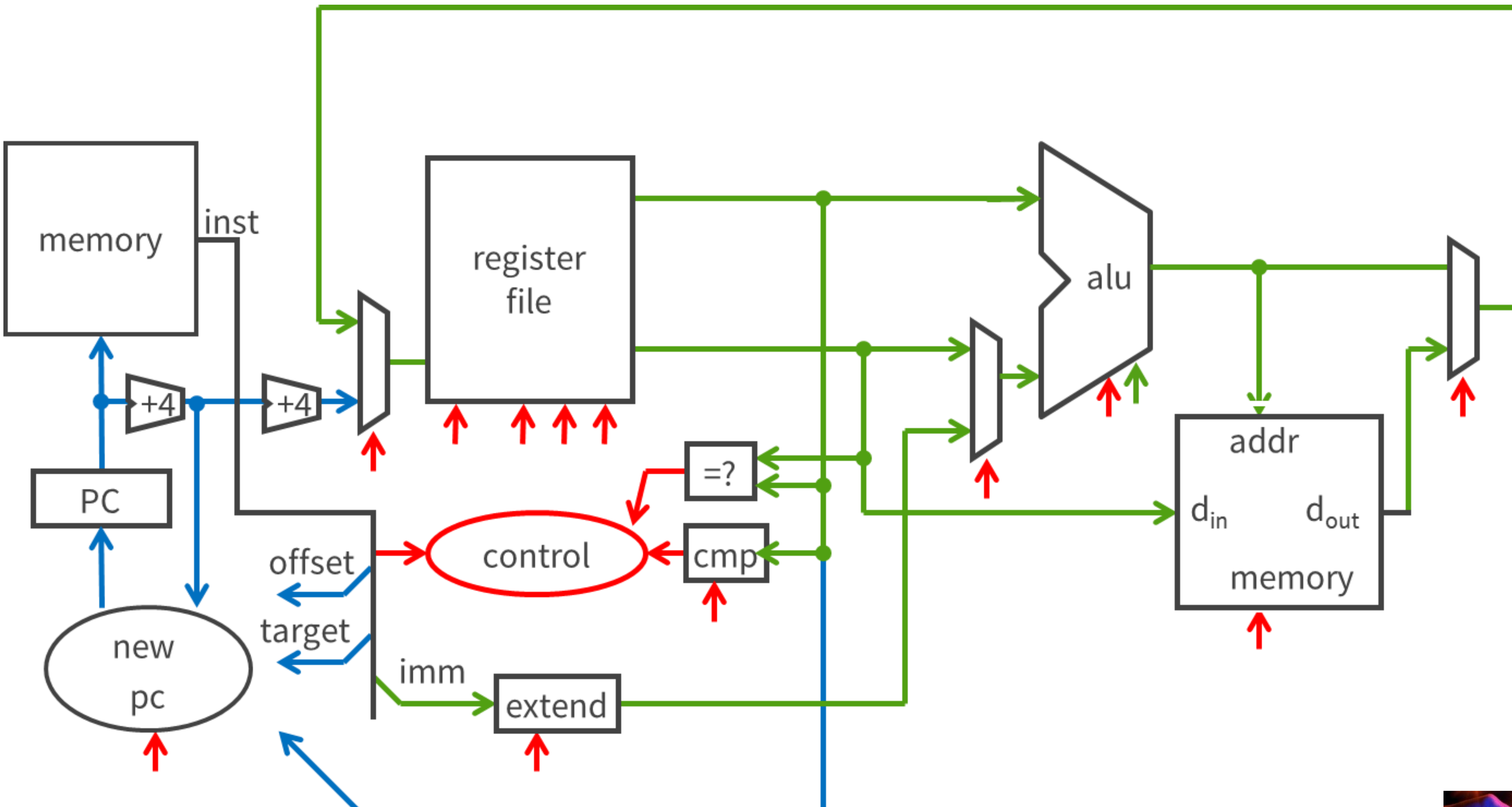


CONTENTS

- TRANSISTORS TO LOGIC GATES

BIG PICTURE: BUILDING A PROCESSOR

- Single cycle processor



WHAT IS THIS?

- How does it work?

```
#include <stdio.h>

int main() {
  printf("Hello world!\n");
  return 0;
}
```

COMPILERS AND ASSEMBLERS

- From high level language to machine language

C

```
int x = 10;
x = 2 * x + 15;
```

compiler

r0 = 0

RISC-V assembly language

```
addi r5, r0, 10
muli r5, r5, 2
addi r5, r5, 15
```

r5 = r0 + 10
r5 = r5 * 2
r5 = r5 + 15

Everything is a number!

assembler

RISC-V machine language

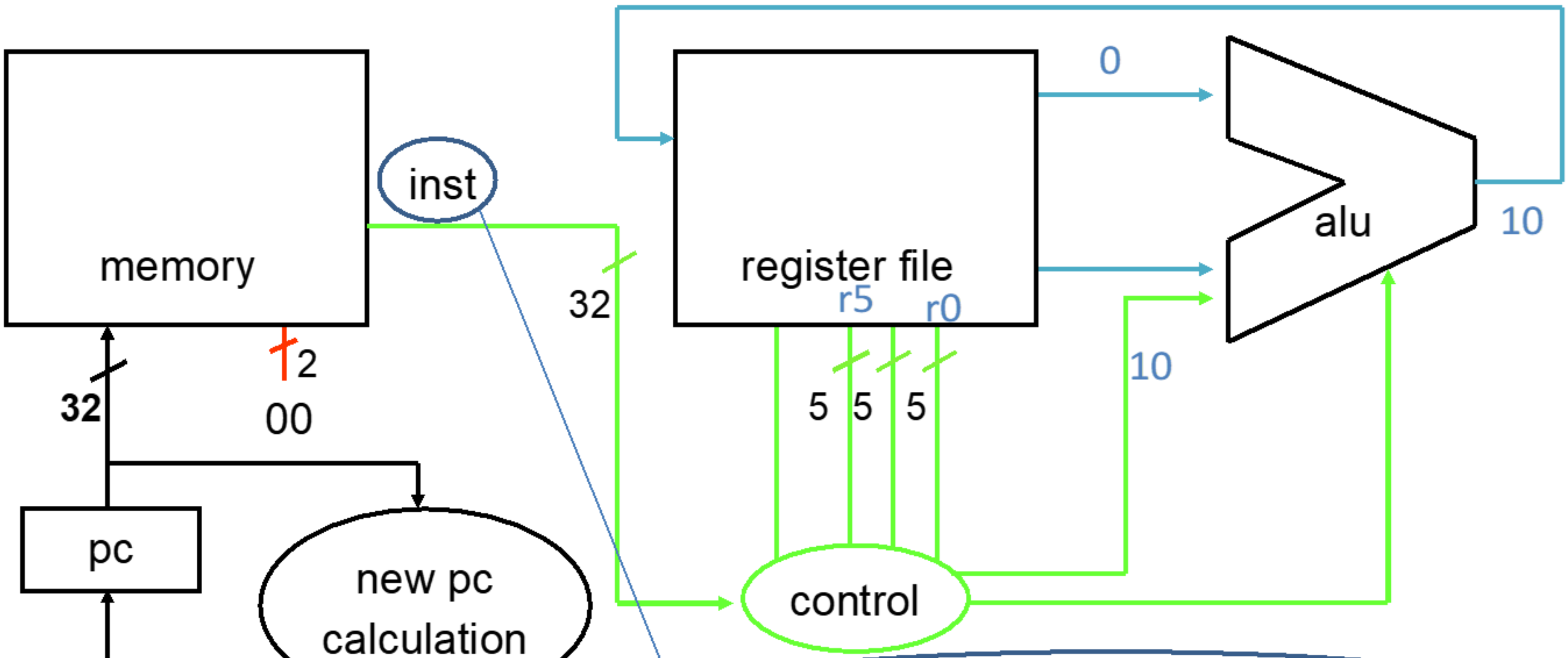
10 | r0 | | r5 | op = addi

```
000000001010 00000 000 00101 0010011
00000000000100101001001010010011
000000001111 00101 000 00101 0010011
```

15 | r5 | | r5 | op = addi

COMPILERS AND ASSEMBLERS

❖ How to design a simple process




```
00: addi    r5, r0, 10
04: muli    r5, r5, 2
08: addi    r5, r5, 15
```

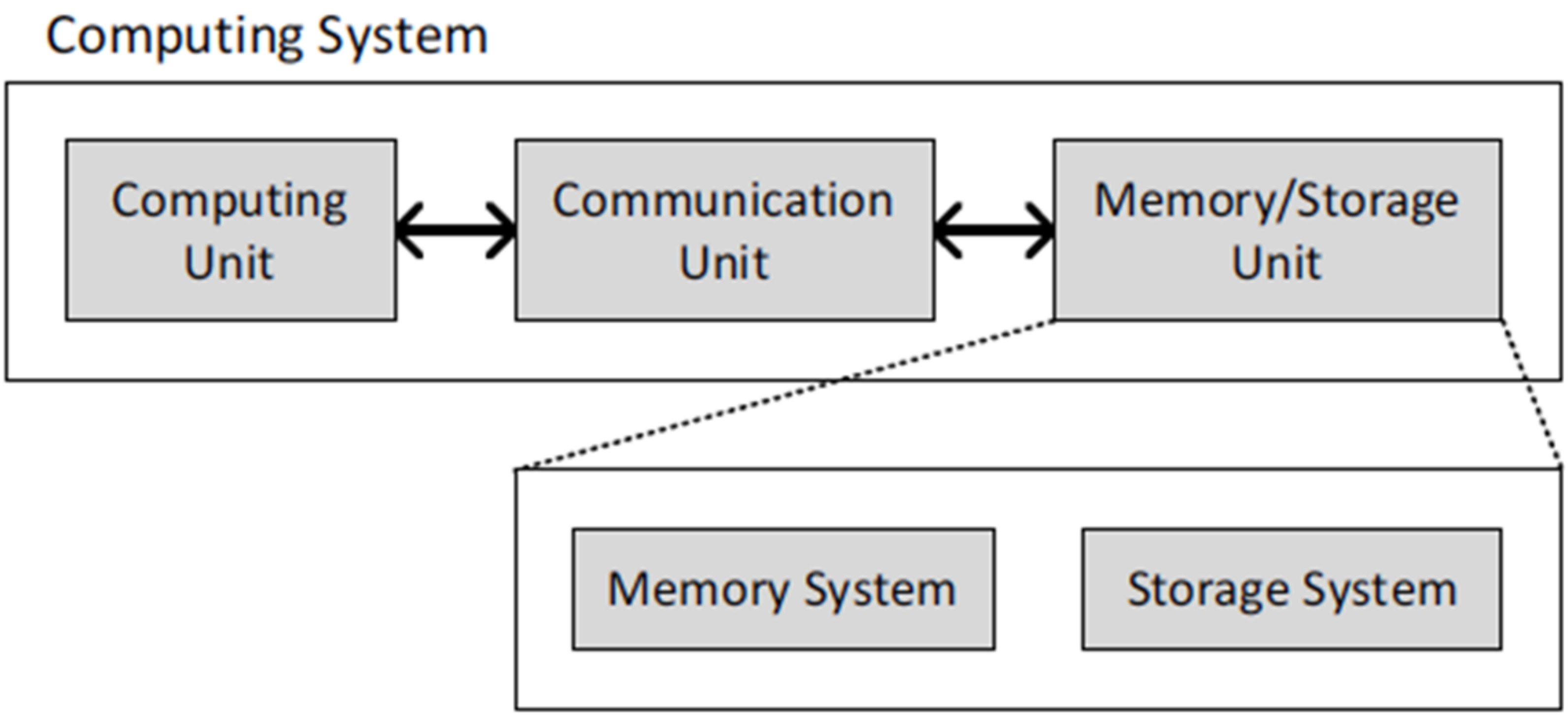
INSTRUCTION SET ARCHITECTURE (ISA)

- Abstract interface between hardware and the lowest level software

- User portion of the instruction set plus the operating system interfaces used by application programmers

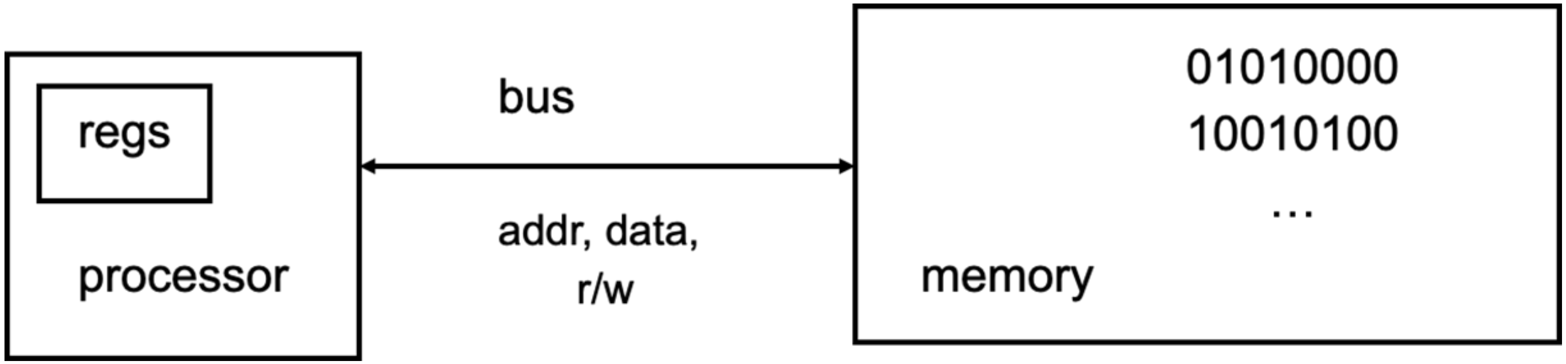
BASIC COMPUTER SYSTEM

- Three key components
  - Computation
  - Communication
  - Storage/memory



BASIC COMPUTER SYSTEM

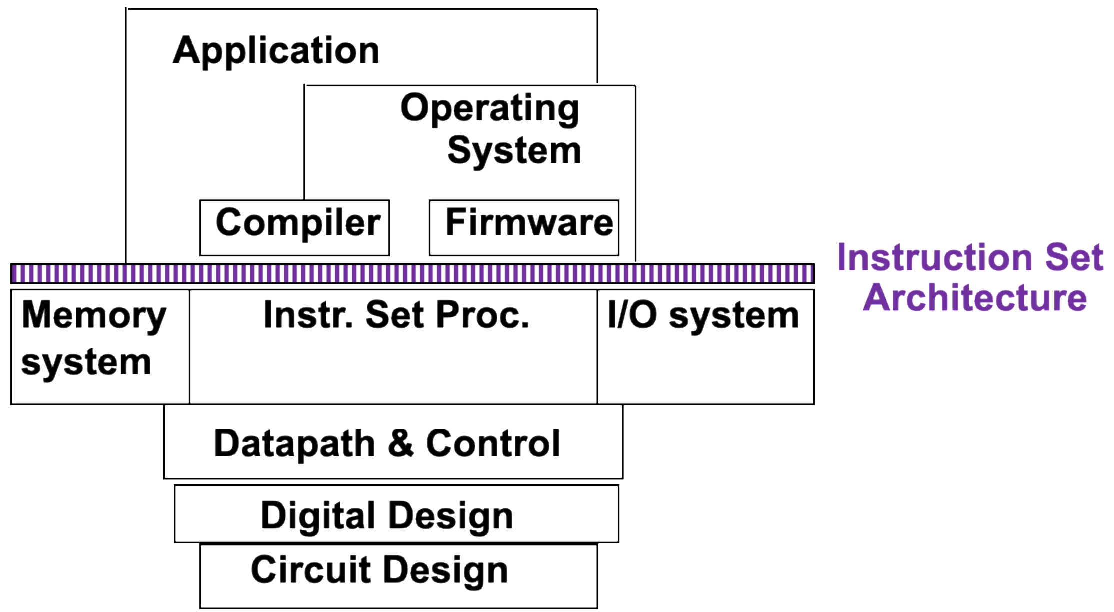
- A processor executes instructions
  - Processor has some internal state in storage elements (registers)
- A memory holds instructions and data
  - Von Neumann architecture: combined **inst** and **data**
- A bus connects the two



OVERVIEW

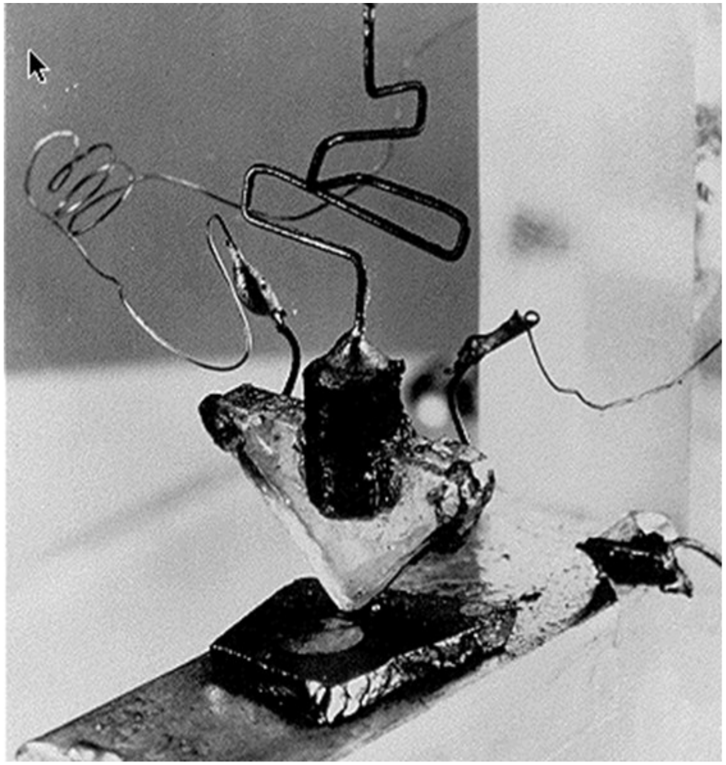
- Covered in this course

- Problem
- Algorithm
- Program/Language
- Runtime System (VM, OS, MM)
- ISA (Architecture)
- Microarchitecture
- Logic
- Devices
- Electrons



WHERE DID IT BEGIN:

- Electrical Switch
  - On/Off
  - Binary
- Transistor



The first transistor on a workbench at AT&T Bell Labs in 1947

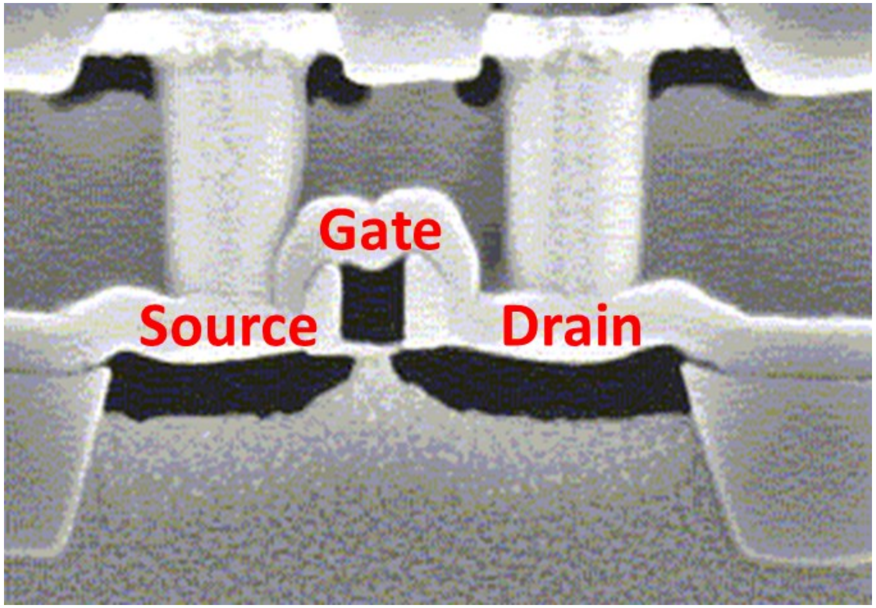
TRANSISTORS

- Computers are built from very large numbers of very small (and relatively simple) structures: transistors
  - Intel 4004, in 1971, had 2300 MOS transistors
  - Intel's Pentium IV microprocessor, 2000, was made up of more than 42 Million MOS transistors
  - Apple's M2 Max, offered for sale in 2022, is made up of more than 67 Billion MOS transistors

Problem
Algorithm
Program/Language
Runtime System (VM, OS, MM)
ISA (Architecture)
Microarchitecture
Logic
Devices
Electrons

MOS TRANSISTOR

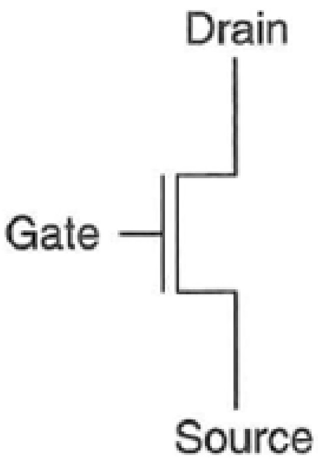
- By combining
  - Conductors (Metal)
  - Insulators (Oxide)
  - Semiconductors



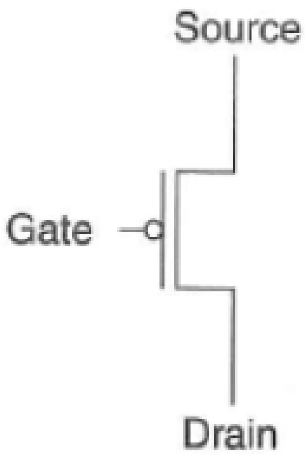
- We get a Transistor (MOS)
- Why is this useful?
  - We can combine many of these to realize simple logic gates
- The electrical properties of metal-oxide semiconductors are well beyond the scope of what we want to understand in this course
  - They are below our lowest level of abstraction

MOS TRANSISTOR

- There are two types of MOS transistors: n-type and p-type



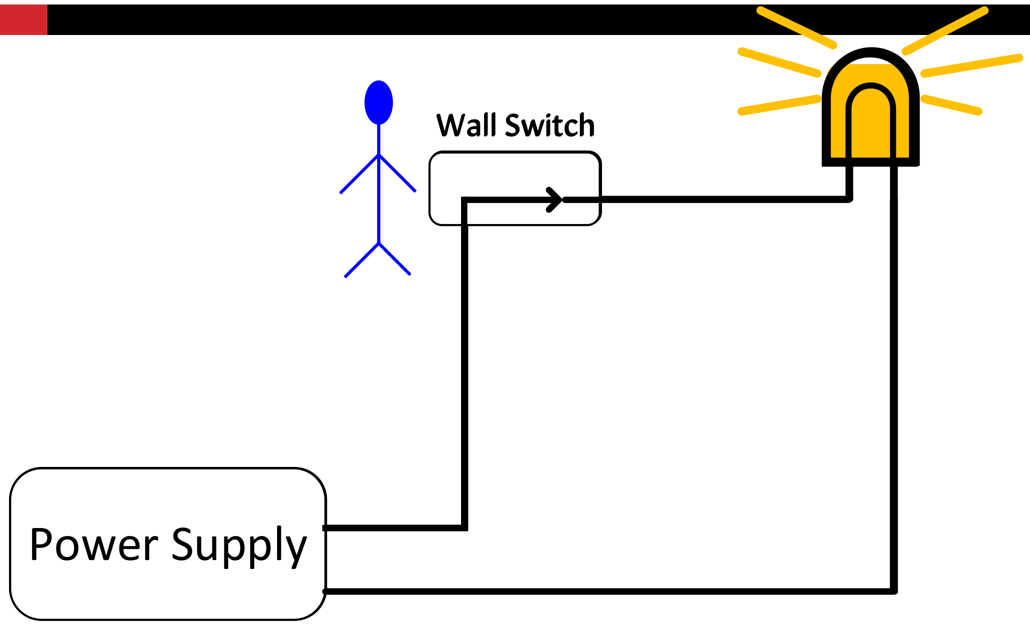
n-type



p-type

- They both operate "logically," very similar to the way wall switches work

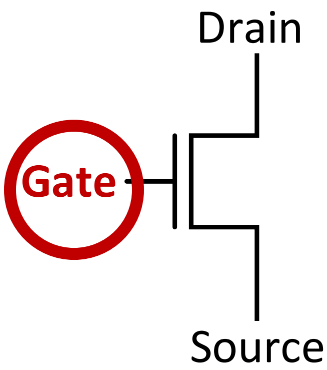
MOS TRANSISTOR



- In order for the lamp to glow, electrons must flow
- In order for electrons to flow, there must be a closed circuit from the power supply to the lamp and back to the power supply
- The lamp can be turned on and off by simply manipulating the wall switch to make or break the closed circuit

MOS TRANSISTOR

- Instead of the wall switch, we could use an n-type or a p-type MOS transistor to make or break the closed circuit



Schematic of an n-type MOS transistor

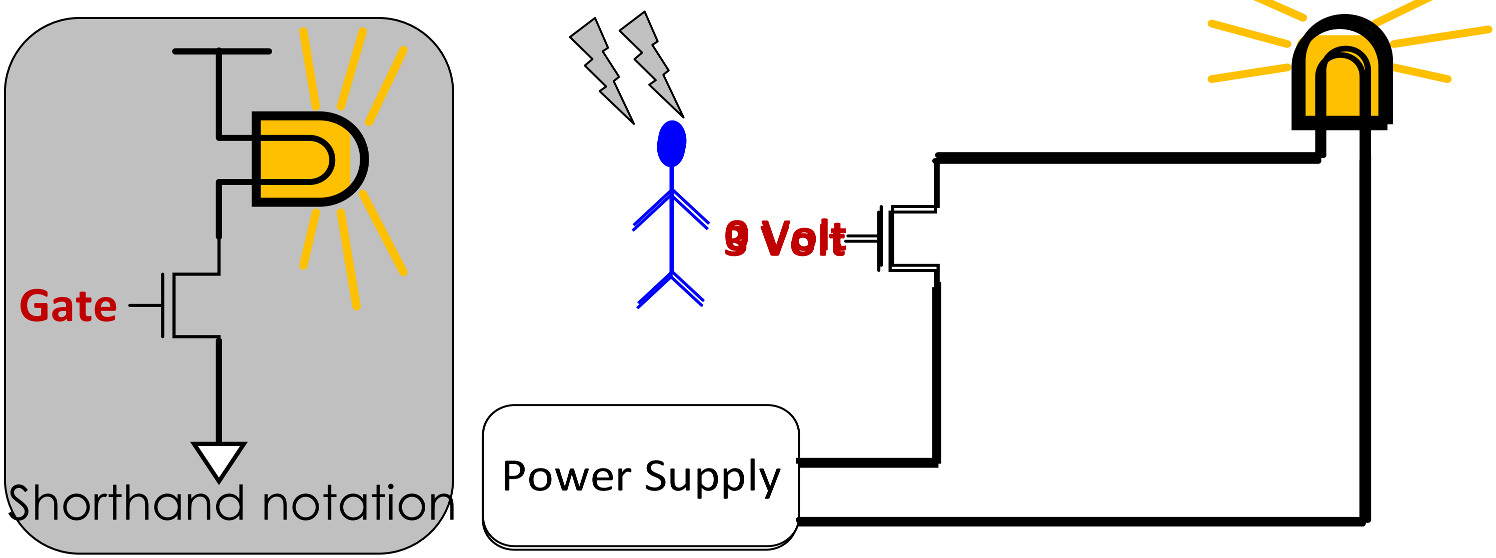
If the gate of an n-type transistor is supplied with a **high** voltage, the connection from source to drain acts like a piece of wire (i.e., the circuit is closed)

Depending on the technology, high voltage can range from 0.3V to 3

If the gate of the n-type transistor is supplied with **zero** voltage, the connection between the source and drain is broken (i.e., the circuit is open)

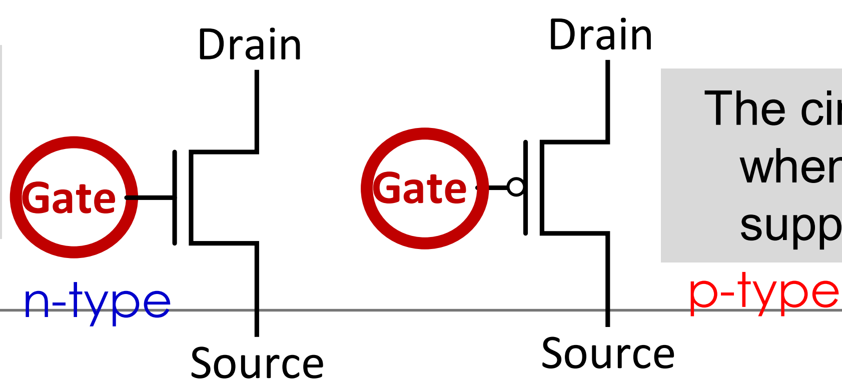
MOS TRANSISTOR

- The n-type transistor in a circuit with a battery and a bulb



- The p-type transistor works in exactly the opposite fashion from the n-type transistor

The circuit is closed when the gate is supplied with 3V



The circuit is closed when the gate is supplied with 0V

LOGIC GATES

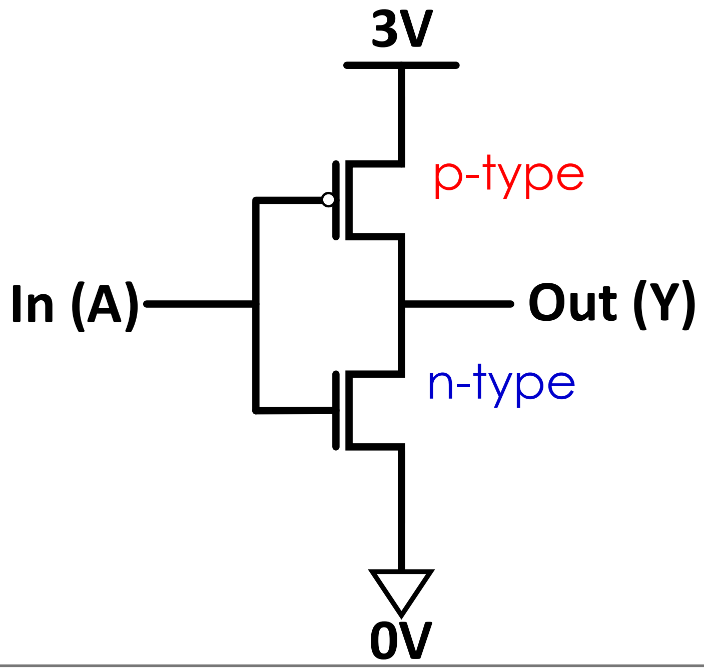
- We know how a MOS transistor works
  - How do we build logic structures out of MOS transistors?
    - We construct basic logical units out of individual MOS transistors
    - These logical units are called logic gates
    - They implement simple Boolean functions

LOGIC GATES

- Modern computers use both n-type and p-type transistors, i.e. Complementary MOS (CMOS) technology

nMOS + pMOS = CMOS

- The simplest logic structure that exists in a modern computer

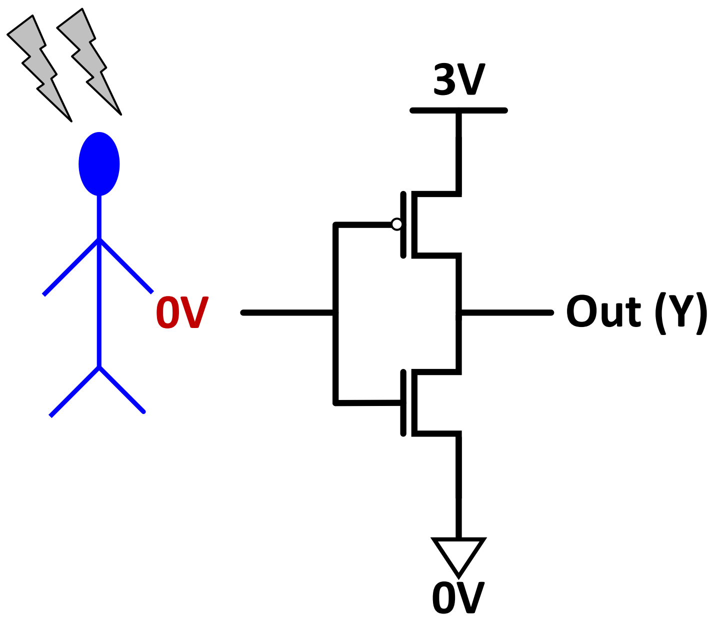


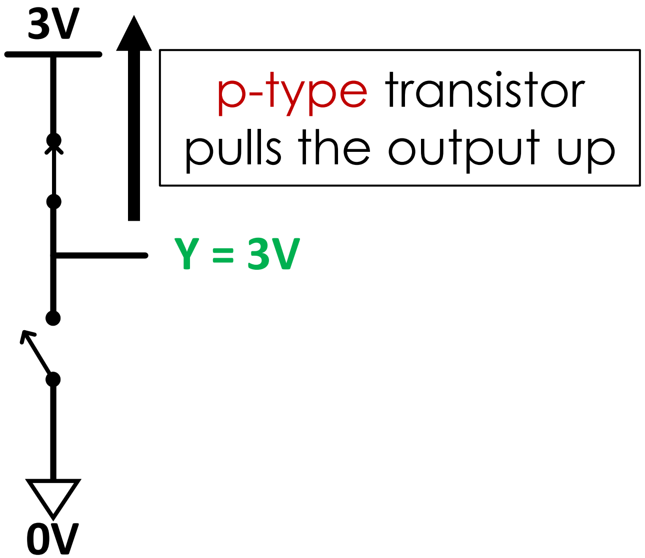
What does this circuit do?

LOGIC GATES

❖ Functionality of CMOS circuits

What happens when the input is connected to 0V?



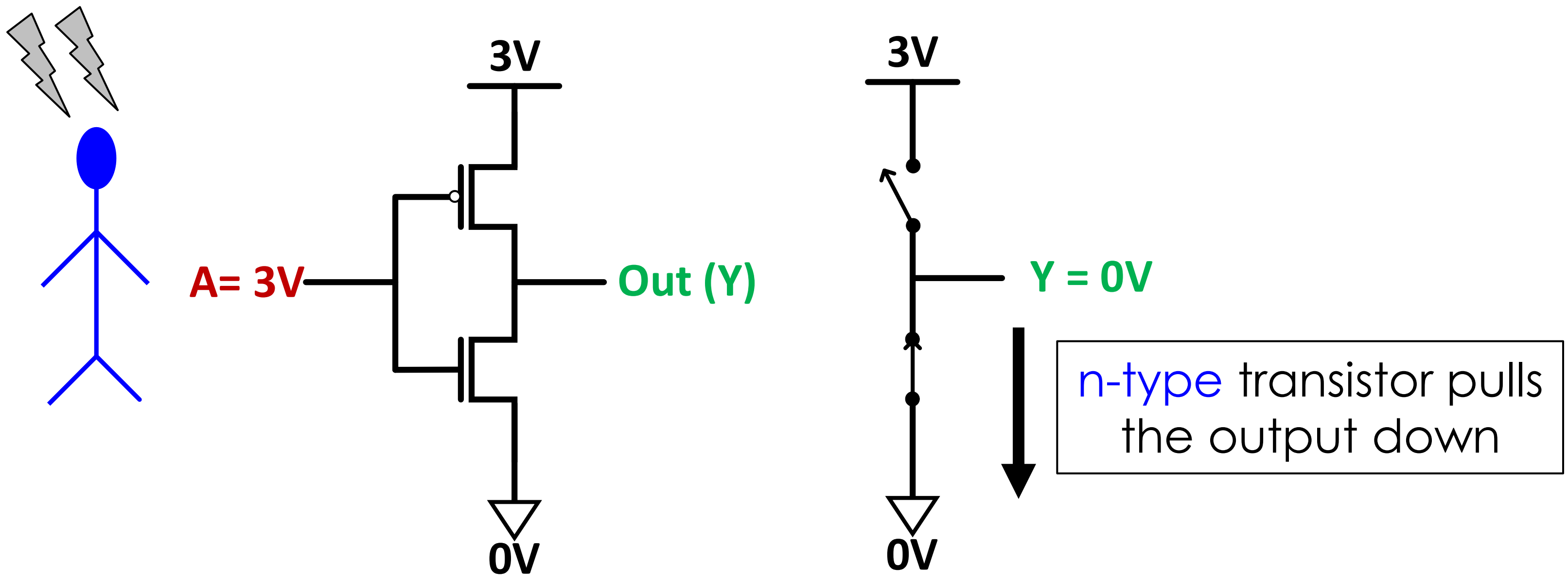


p-type transistors are good at pulling up the voltage

LOGIC GATES

- Functionality of CMOS circuits

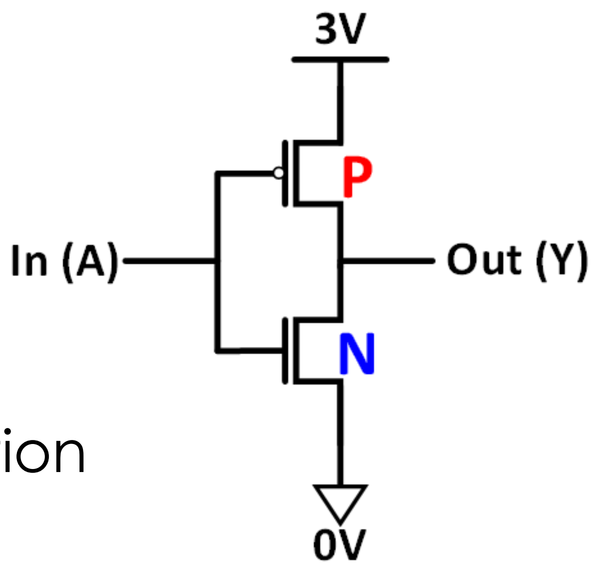
What happens when the input is connected to 3V?



n-type transistors are good at pulling dow**n** the voltage

LOGIC GATES

- Functionality of CMOS circuits
- This is actually the CMOS NOT Gate
- Why do we call it NOT?
  - If A = 0V then Y = 3V
  - If A = 3V then Y = 0V
- Digital circuit: one possible interpretation
  - Interpret 0V as logical (binary) 0 value
  - Interpret 3V as logical (binary) 1 value

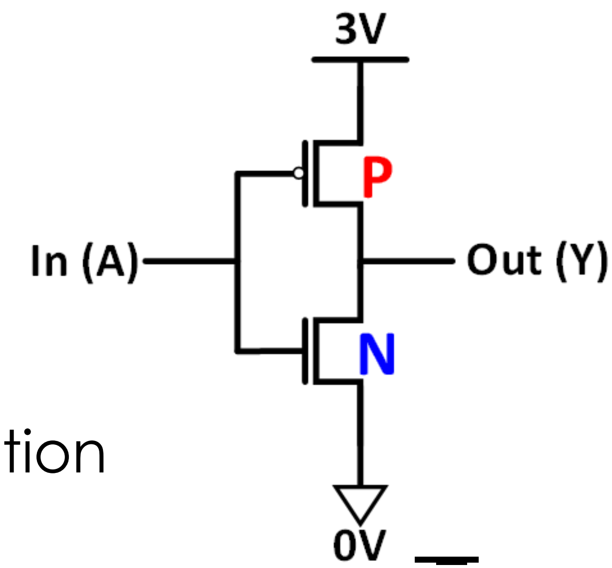


A	P	N	Y
0	ON	OFF	1
1	OFF	ON	0

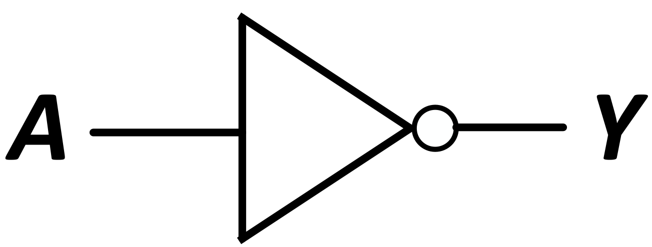
$$Y = \bar{A}$$

LOGIC GATES

- Functionality of CMOS circuits
- This is actually the CMOS NOT Gate
- Why do we call it NOT?
  - If A = 0V then Y = 3V
  - If A = 3V then Y = 0V
- Digital circuit: one possible interpretation
  - Interpret 0V as logical (binary) 0 value
  - Interpret 3V as logical (binary) 1 value



$$Y = \bar{A}$$



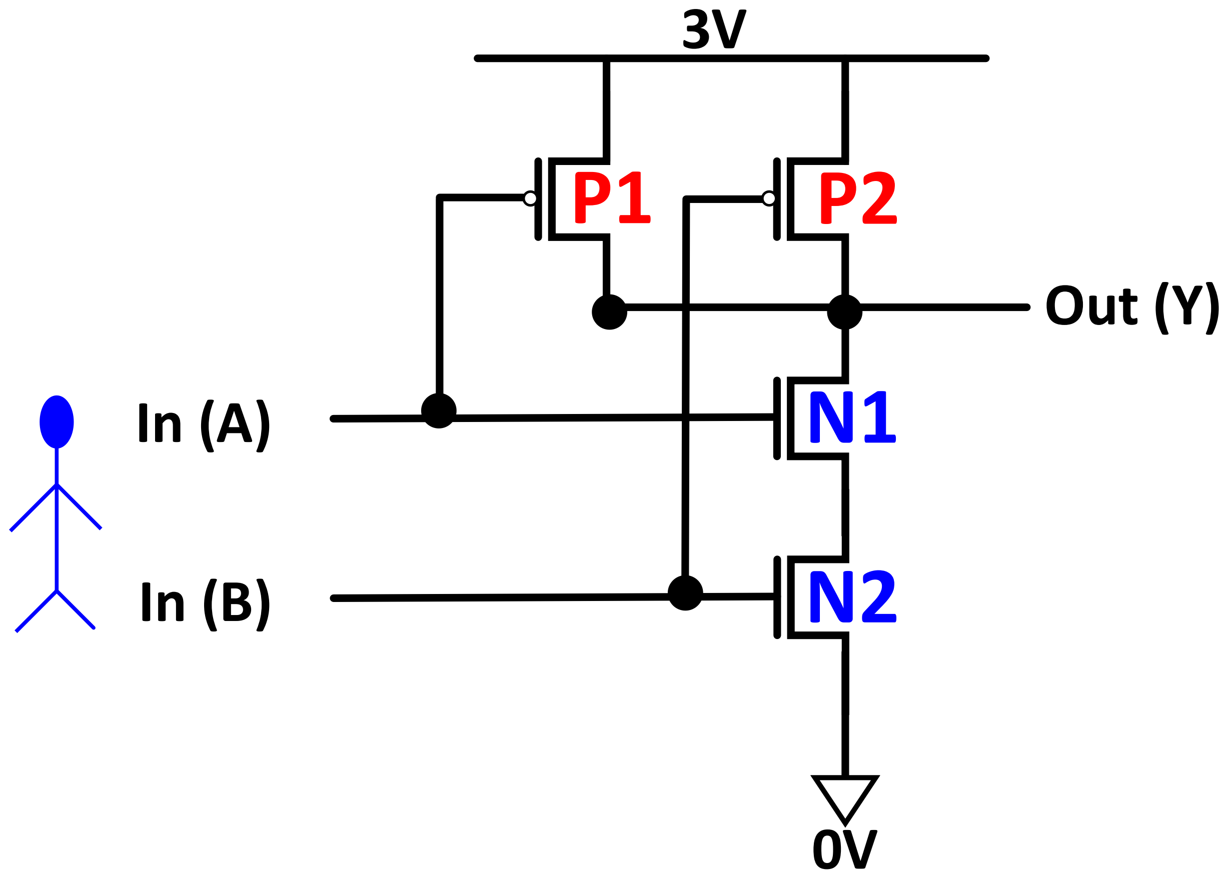
We call this a NOT gate
or an inverter
(bubble indicates inversion)

Truth table: shows what is the logical output of the circuit for each possible input

A	Y
0	1
1	0

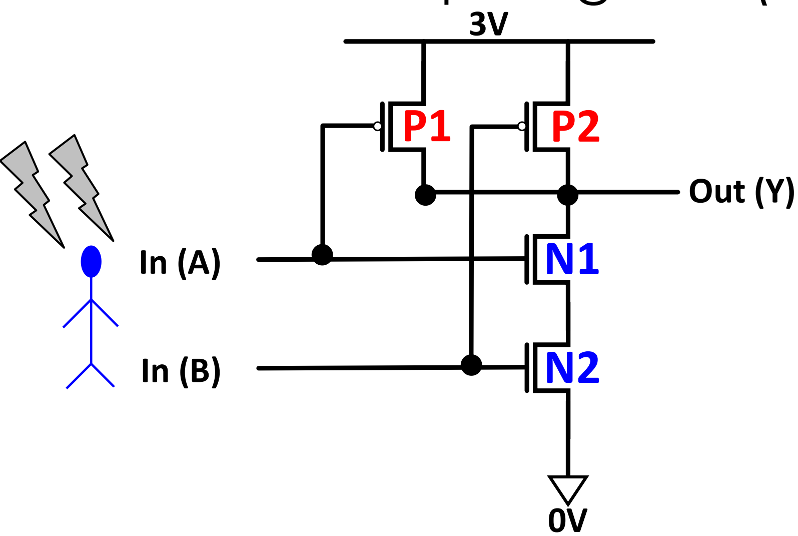
LOGIC GATES

- More complex gates



LOGIC GATES

- More complex gates (CMOS NAND gate)



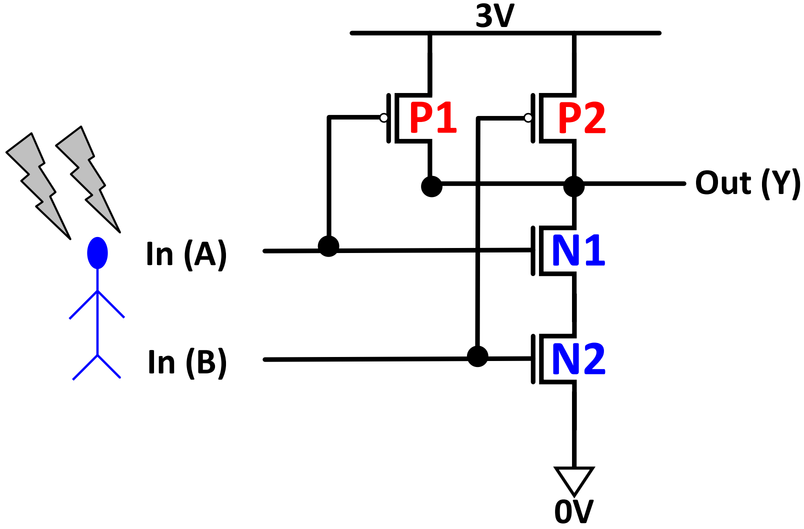
$$Y = \overline{A \cdot B} = \overline{AB}$$

A	B	P1	P2	N1	N2	Y
0	0	ON	ON	OFF	OFF	1
0	1	ON	OFF	OFF	ON	1
1	0	OFF	ON	ON	OFF	1
1	1	OFF	OFF	ON	ON	0

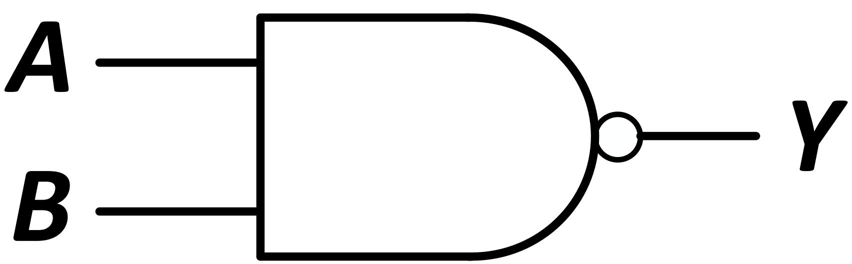
- P1 and P2 are in parallel; only one must be ON to pull up the output to 3V
- N1 and N2 are connected in series; both must be ON to pull down the output to 0V

LOGIC GATES

- More complex gates (CMOS NAND gate)



$$Y = \overline{A \cdot B} = \overline{AB}$$



A	*B*	*Y*
0	0	1
0	1	1
1	0	1
1	1	0

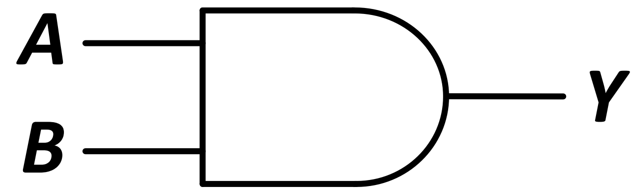
We call this a NAND gate
(bubble indicates inversion)

LOGIC GATES

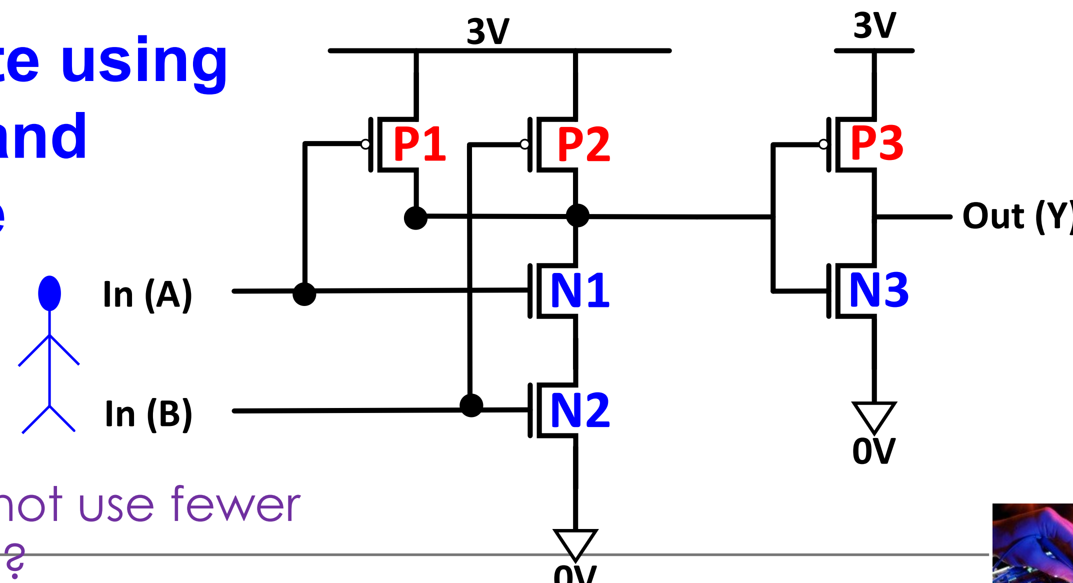
- More complex gates (CMOS AND gate)

$$Y = A \cdot B = AB$$

A	*B*	*Y*
0	0	0
0	1	0
1	0	0
1	1	1



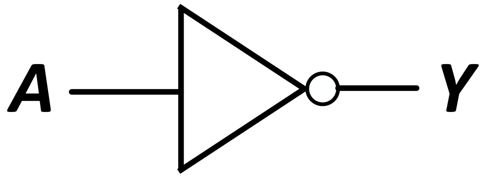
We make an AND gate using one NAND gate and one NOT gate

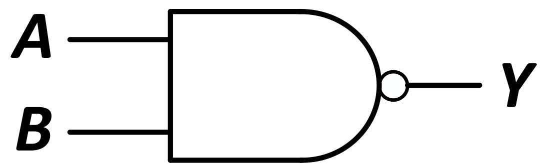


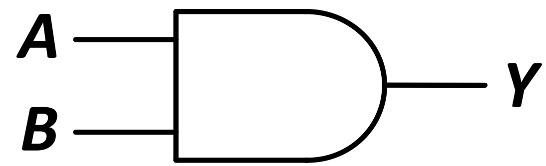
Food for thought: Can we not use fewer transistors for the AND gate?

LOGIC GATES

- CMOS NOT, NAND, AND gates



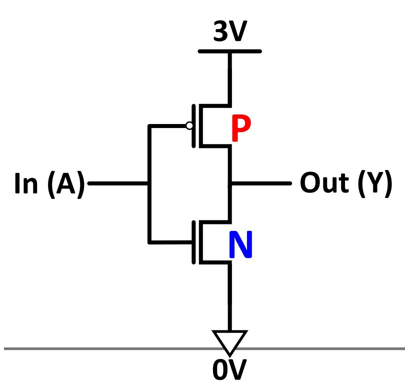


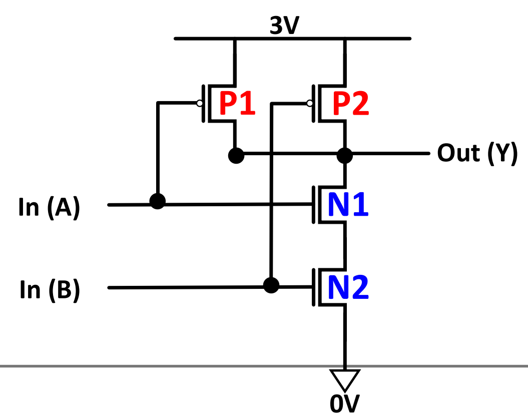


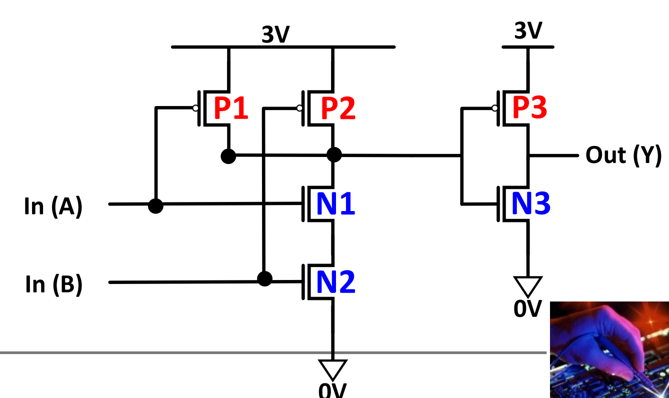
A	Y
0	1
1	0

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

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







MOORE'S LAW

- 1965
- # of transistors integrated on a die doubles every 18-24 months (i.e., grows exponentially with time)

- Amazingly visionary
  - 2300 transistors, 1 MHz clock (Intel 4004) - 1971
  - 16 Million transistors (Ultra Sparc III)
  - 42 Million transistors, 2 GHz clock (Intel Xeon) – 2001
  - 55 Million transistors, 3 GHz, 130nm technology, 250mm2 die (Intel Pentium 4) – 2004
  - 290+ Million transistors, 3 GHz (Intel Core 2 Duo) – 2007
  - 721 Million transistors, 2 GHz (Nehalem) - 2009
  - 1.4 Billion transistors, 3.4 GHz Intel Haswell (Quad core) – 2013
  - 7.2 Billion transistors, 3-3.9 GHz Intel Broadwell (22-core) – 2016

MOORE'S LAW

❖ # of transistors integrated on a die doubles every 18-24 months (i.e., grows exponentially with time)