Digital Circuits

CSCI 101

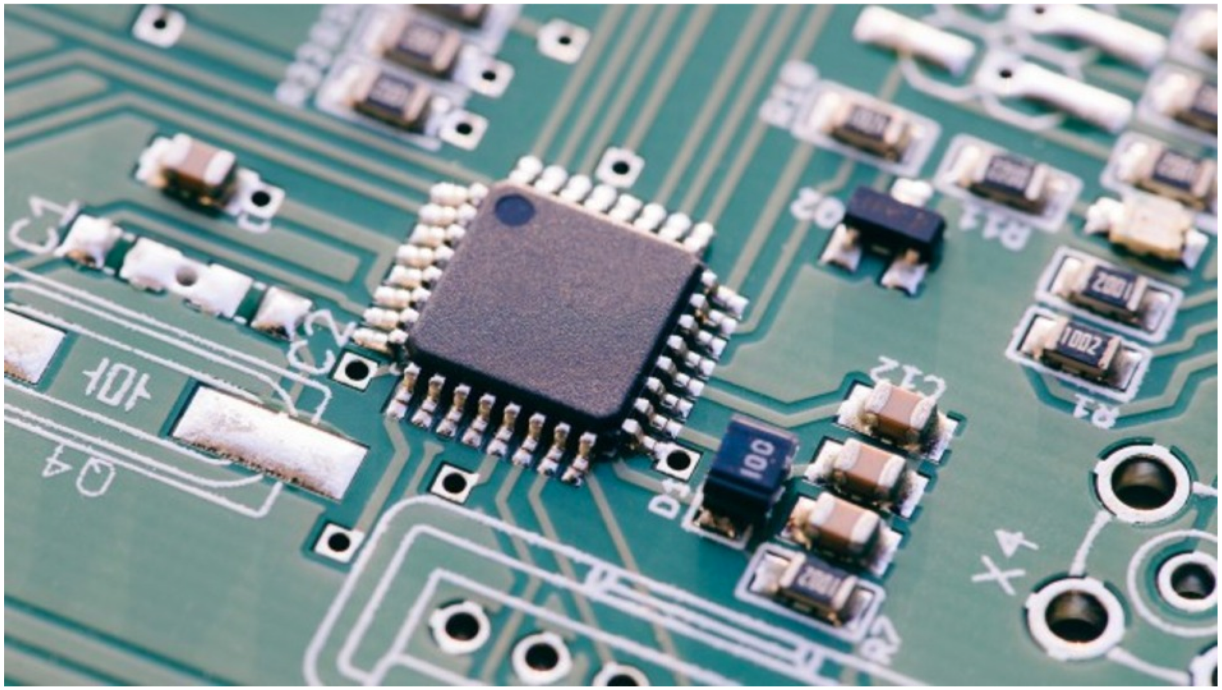
Spring 2018

Professors Briggs and Grant

Today

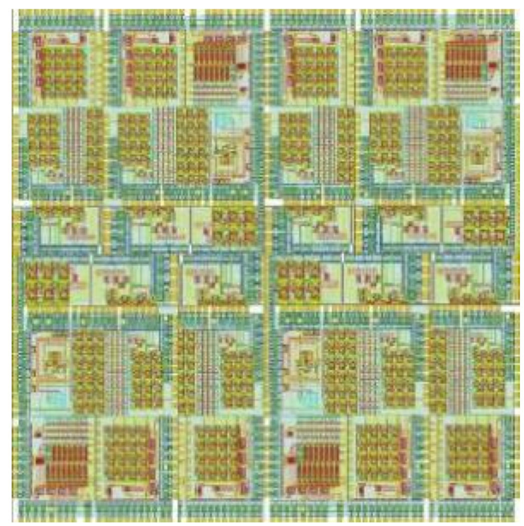
- Digital Circuits
  - Logic gates: AND, OR, NOT
  - Truth tables
  - Transistors
  - Logic circuits: XOR, adder, flip-flop

- Reading: *CS for All* section 4.3

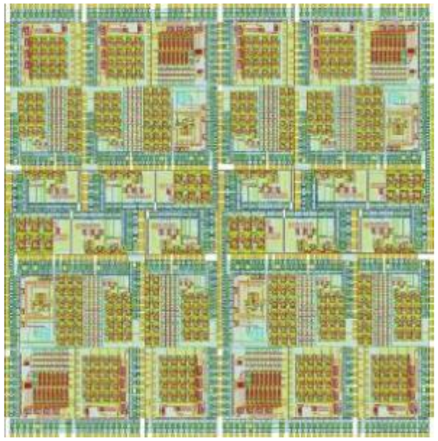


Computer **board**

Computer **chip**



Chips, Circuits, and Gates



- Chip – integrated circuit of many transistors made using aluminum or copper and imprinted on a silicon base
- Gate – a low-level construction that produces a binary output based on one or more binary inputs (e.g., AND, OR, NOT)
- Circuit – some combination of gates (made of transistors)

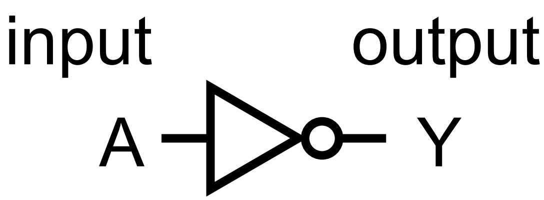
Digital Circuits

- Why binary?
- On lowest level, wires carry voltage
- 2 possible states on each wire:

0V / 5V   0 / 1   off / on   false / true

Inverter

Can switch a binary signal from 0 to 1 and vice versa



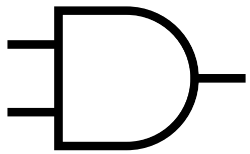
NOT

A	Y
0	1
1	0

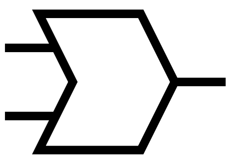
$$Y = \overline{A}$$

AND and OR gates

Combine 2 binary signals to form a single output



AND

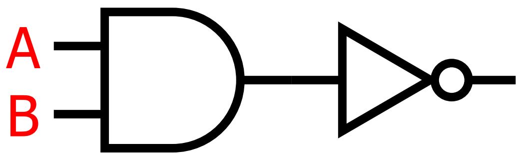


OR

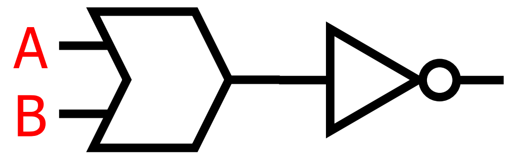
A	B	A • B	A + B
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

NAND, NOR, and XOR Gates

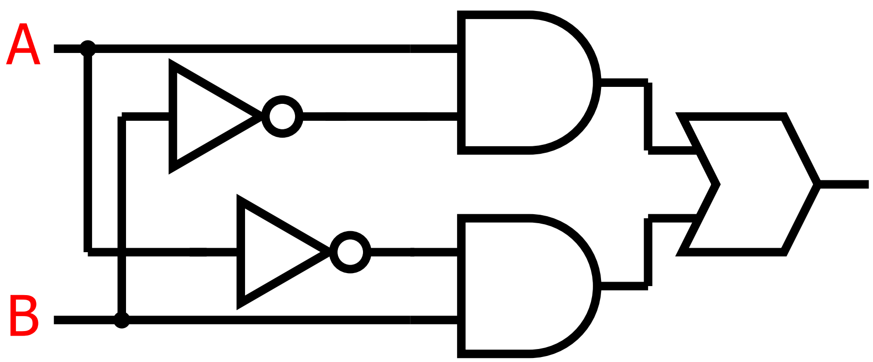
- Build a circuit for $\overline{(A \bullet B)}$



- Build a circuit for $\overline{(A + B)}$

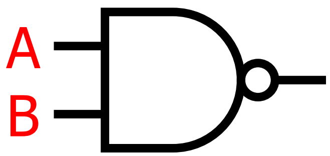


- Build a circuit for XOR: $(A \bullet \overline{B}) + (\overline{A} \bullet B)$ ← Sum of Products Equation



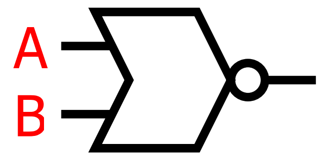
NAND, NOR, and XOR Gates

- Build a circuit for $\overline{(A \bullet B)}$



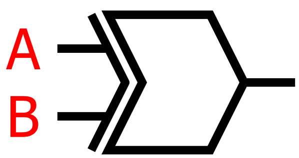
NAND

- Build a circuit for $\overline{(A + B)}$



NOR

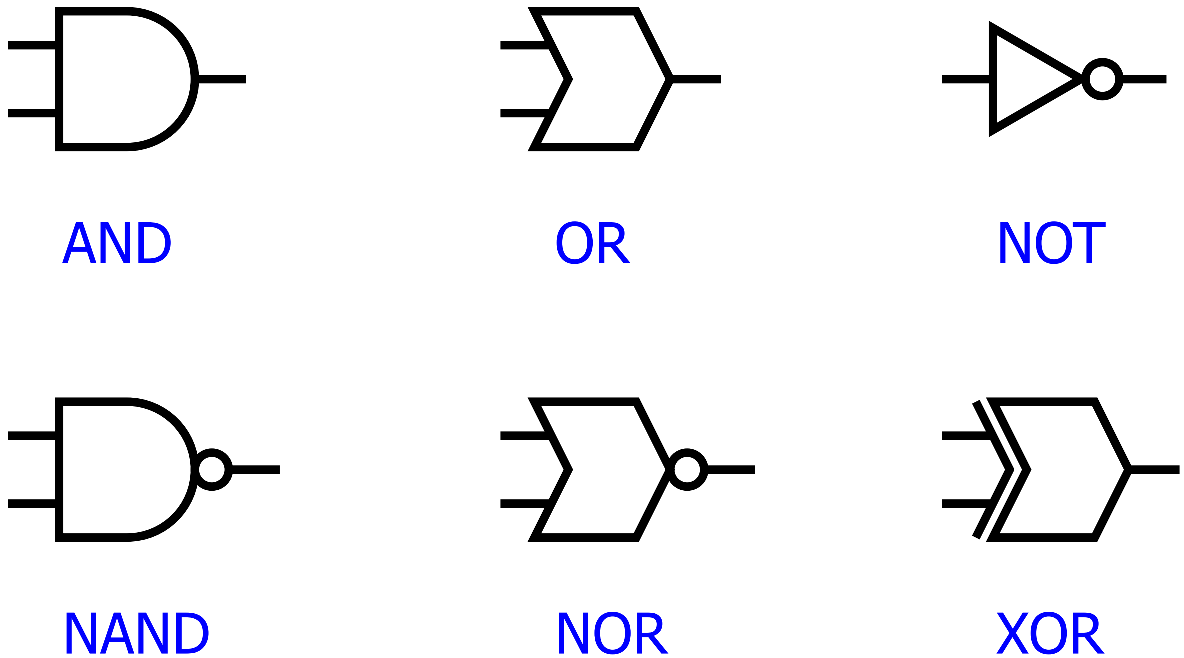
- Build a circuit for $(A \bullet \overline{B}) + (\overline{A} \bullet B)$



XOR

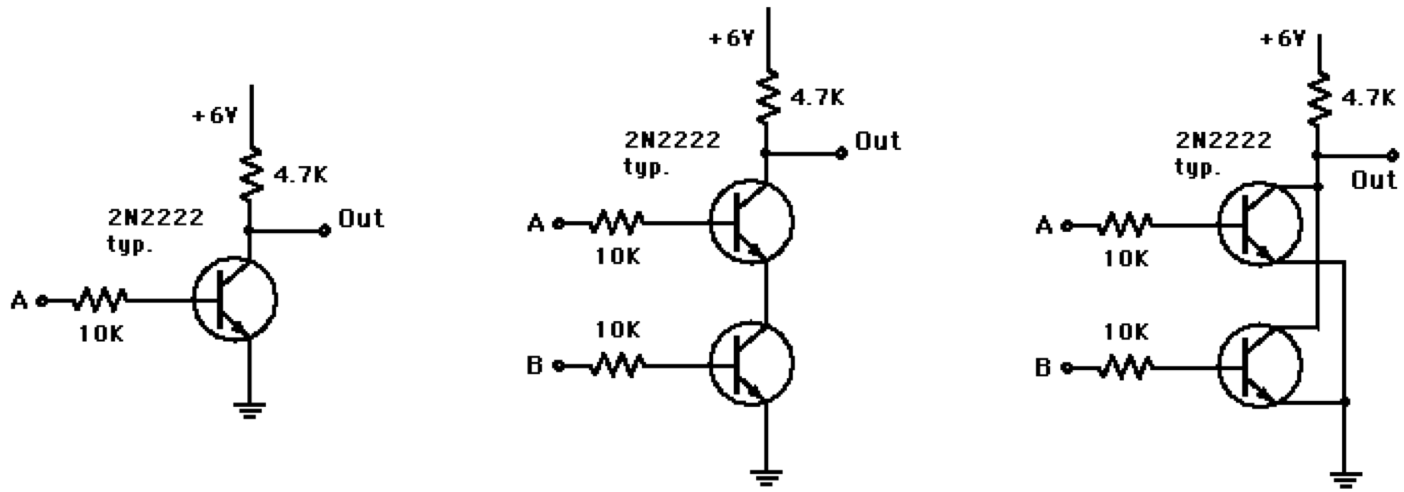
Abstraction

Summary: Basic Logic Gates



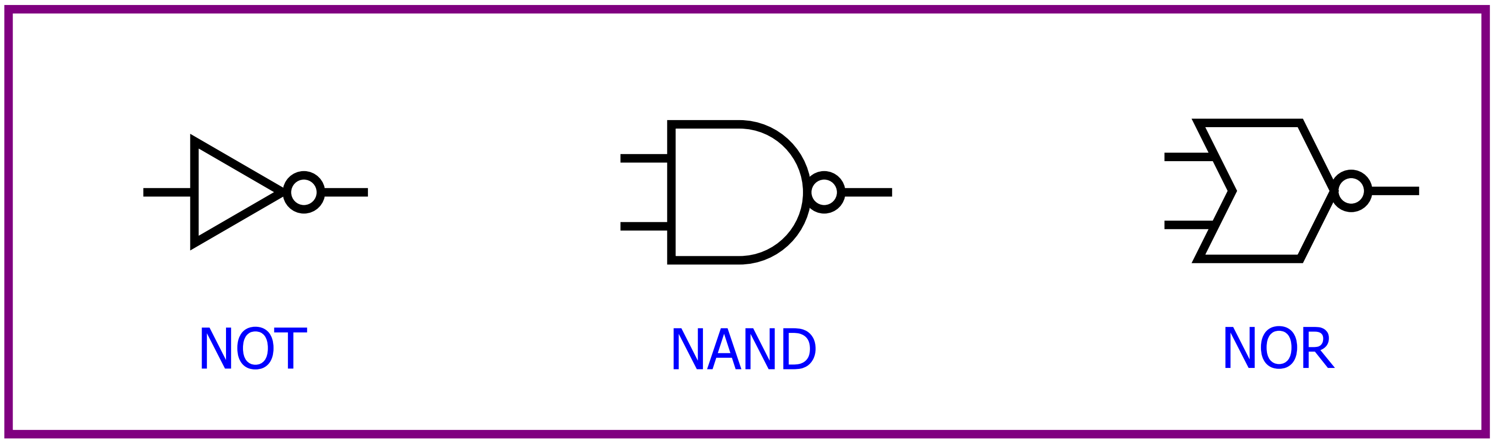
Transistors

- Transistors work like on/off switches for electricity
- Logic gates can be built with transistors



Transistors

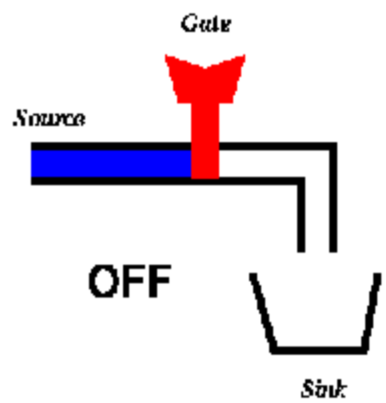
- Transistors work like on/off switches for electricity
- Logic gates can be built with transistors

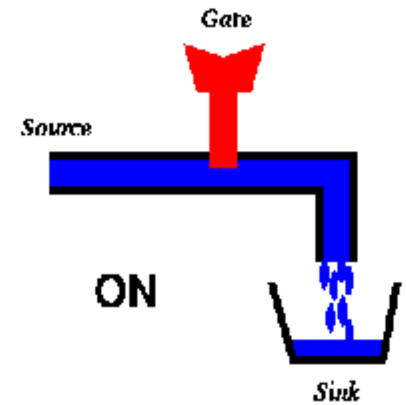


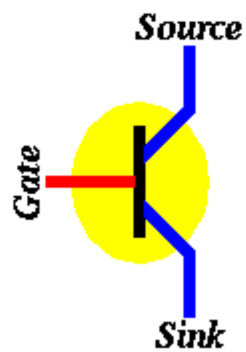
The abstraction we'll use when building circuits

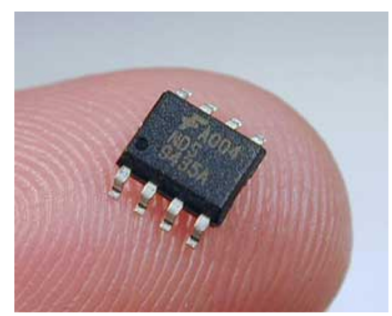
What Transistors Do

- Work like faucet
  - Constant supply of available water
  - When valve is open, water can flow through
  - Can determine if water is flowing (1) or not (0) with sensor below spout
- Transistors work with electricity instead of water and semiconductor materials rather than valves









Binary Arithmetic

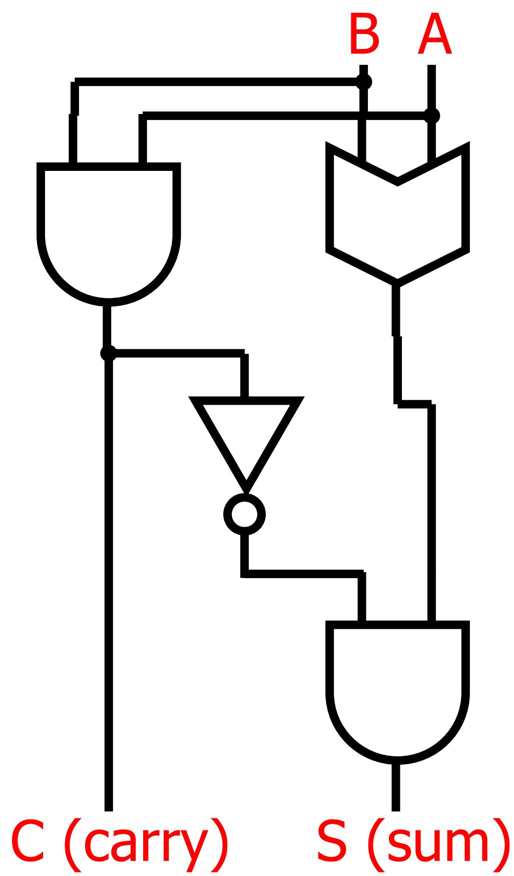
$$\begin{array}{r} 9 \\ +\ \ 3 \\ \hline 12 \end{array} \qquad \begin{array}{c} \text{----->} \\ \text{----->} \\ \end{array} \qquad \begin{array}{r} 1001_2 \\ +\ \ 11_2 \\ \hline 1100_2 \end{array}$$

$$\begin{array}{r} 12 \\ +\ \ 5 \\ \hline 17 \end{array} \qquad \begin{array}{c} \text{----->} \\ \text{----->} \\ \end{array} \qquad \begin{array}{r} 1100_2 \\ +\ 101_2 \\ \hline 10001_2 \end{array}$$

There are 10 types of people in the world: those who know binary, and those who don’t.

The Half-Adder Circuit

- How can we add two 1-bit binary numbers with gates?



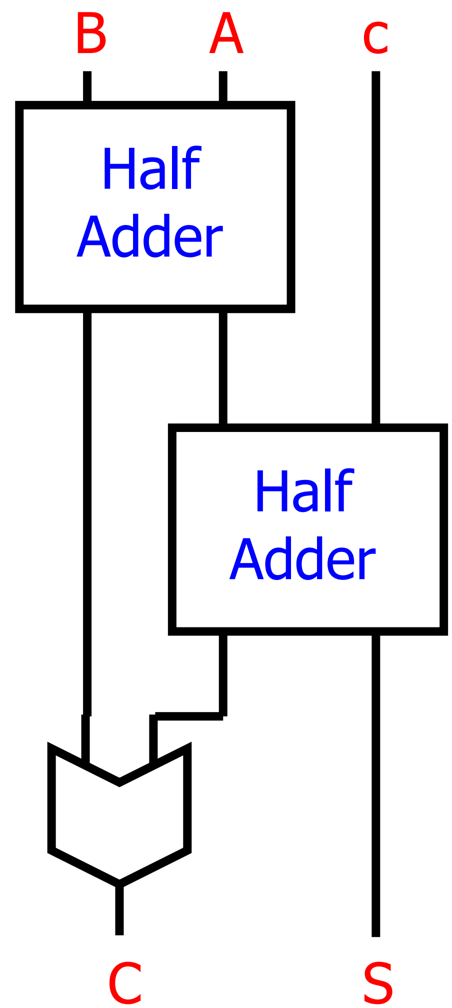
The Half-Adder Circuit

- How can we add two 1-bit binary numbers with gates?

(Another level of abstraction)

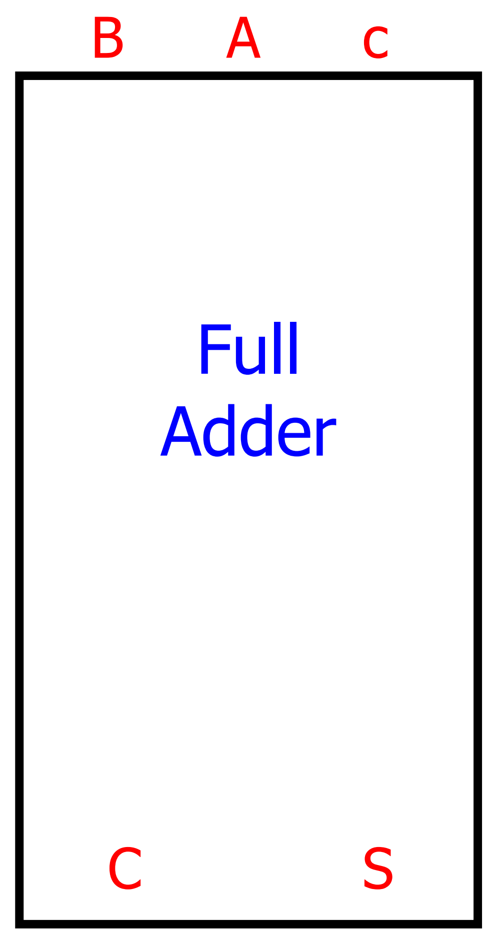
The Full-Adder Circuit

- What if there is a carry bit input from a previous addition?



The Full-Adder Circuit

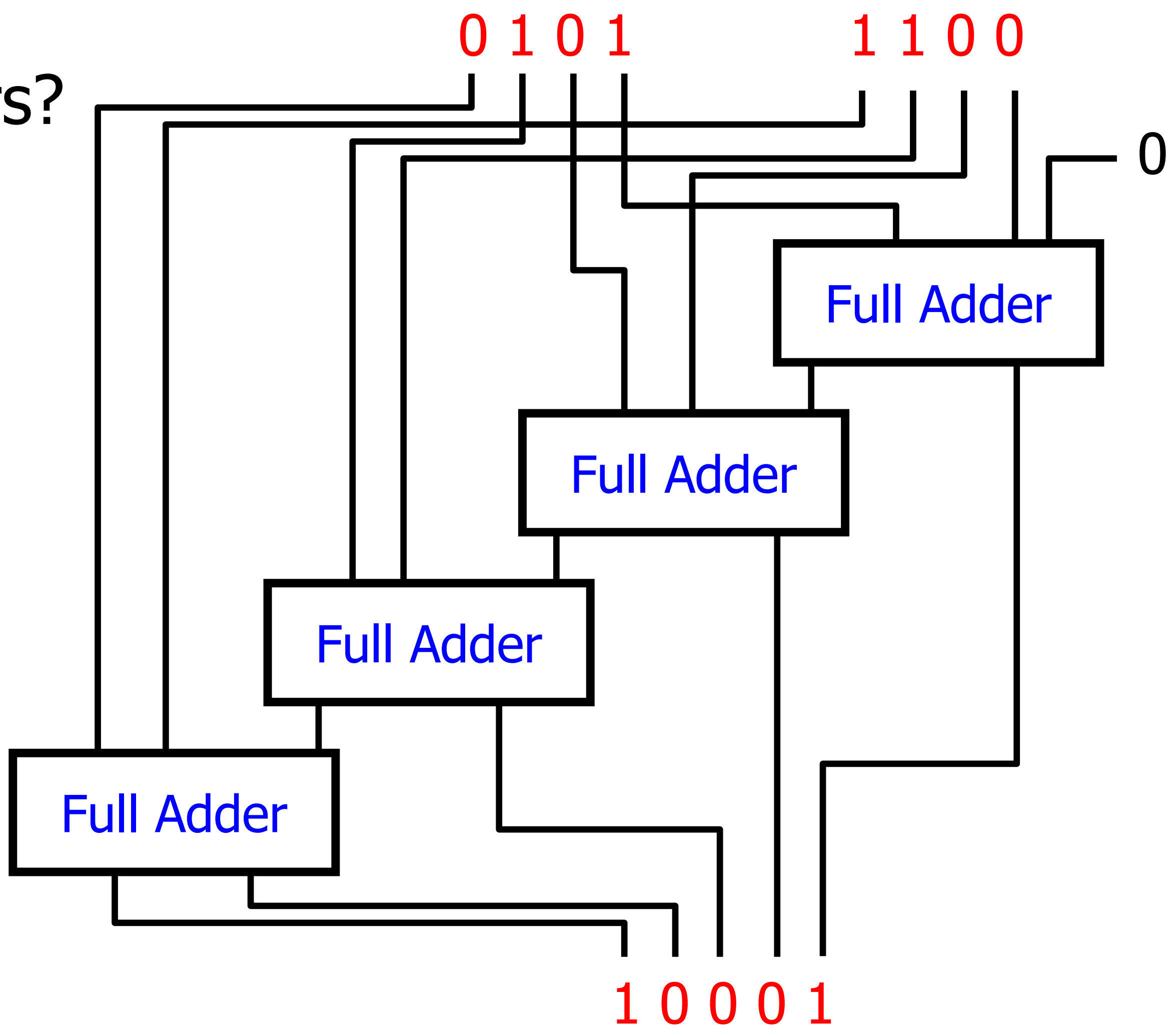
- What if there is a carry bit input from a previous addition?



(Yet another level of abstraction)

Four-Bit Adder Circuit

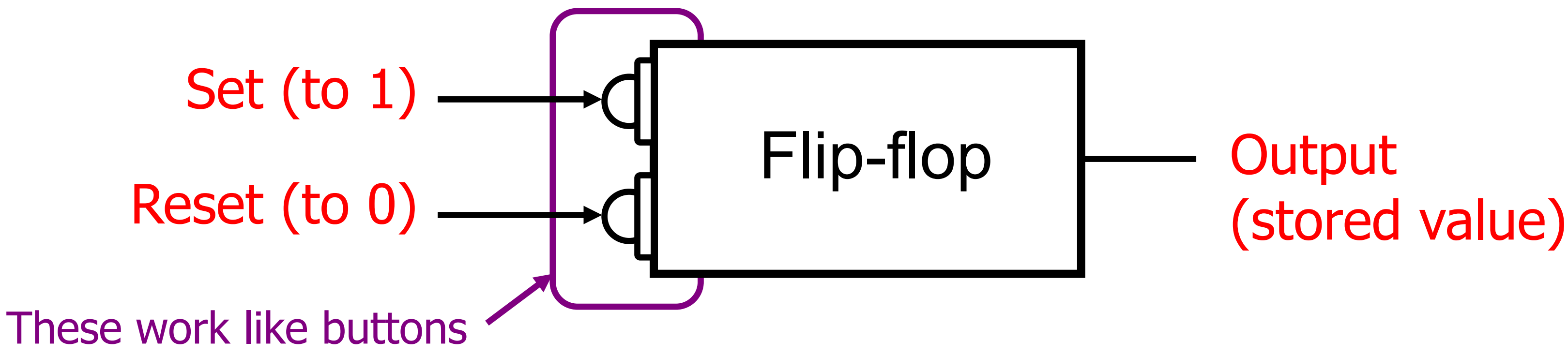
- How can we add two 4-bit numbers?



Current CPUs are 32-bit or 64-bit (can handle that many bits of data at once)

Designing Memory Circuitry

- How can we design circuitry to store values over time?



- Implementation using our basic gates: