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

\[
\begin{array}{c|c}
A & Y \\
0 & 1 \\
1 & 0 \\
\end{array}
\]

\[Y = \overline{A}\]
AND and OR gates

Combine 2 binary signals to form a single output

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>A • B</th>
<th>A + B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
NAND, NOR, and XOR Gates

- Build a circuit for \((A \cdot B)\)
- Build a circuit for \((A + B)\)
- Build a circuit for XOR: \((A \cdot \overline{B}) + (\overline{A} \cdot B)\)
NAND, NOR, and XOR Gates

- Build a circuit for \( (A \cdot B) \)
  - NAND Gate

- Build a circuit for \( (A + B) \)
  - NOR Gate

- Build a circuit for \( (A \cdot \overline{B}) + (\overline{A} \cdot B) \)
  - XOR Gate

Abstraction
Summary: Basic Logic Gates

- AND
- OR
- NOT
- NAND
- NOR
- XOR
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

\[
\begin{align*}
9 + 3 & \quad \Rightarrow \quad 1001_2 \\
12 & \quad \Rightarrow \quad + 11_2 \\
1100_2 & \\
\end{align*}
\]

\[
\begin{align*}
12 + 5 & \quad \Rightarrow \quad 1100_2 \\
17 & \quad \Rightarrow \quad + 101_2 \\
10001_2 & \\
\end{align*}
\]

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?

\[ \begin{align*}
B & \quad A \\
\downarrow & \\
\text{C (carry)} & \quad \text{S (sum)}
\end{align*} \]
The Half-Adder Circuit

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

Half Adder

(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:

```
  Set (to 1)  
  Reset (to 0)  
  Output (stored value)
```

These work like buttons