University of Illinois at Urbana-Champaign
Dept. of Electrical and Computer Engineering

ECE 120: Introduction to Computing

Representations and Bits

Represent One Type of Information with Another

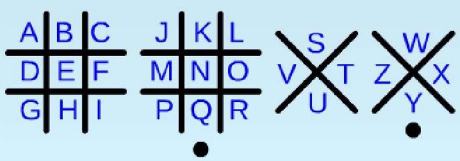
We often represent one type of information with other patterns, physical quantities, and so forth.

examples
- English letters represented by drawn patterns
- colors represented by variations in radio signal amplitude

The **mapping from one form to another** is called a **representation**.

Knowing the Representation May Help You

The code above is called a tic-tac-toe code: each letter (information) is represented by a drawing (pattern).



What Do We Need to Make a Representation Useful?

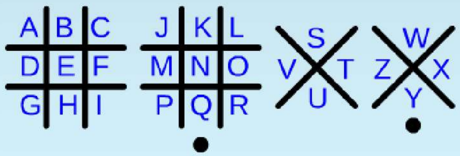
What properties are necessary for a representation to be useful?

Hints:
- Think about the tic-tac-toe code.
- Think about algorithm properties.

First Answer: Representations Must be Well-Defined

All users must **know the translation in advance**.

Our goal is communication, not obfuscation.



Some Mappings May Not be Usable by Computers

0	1	2	3	4	5	6	7	8	9
A	B	C	D	E	F	G	H	I	J
K	L	M	N	O	P	Q	R	S	T
U	V	W	X	Y	Z				

If we use 10 digits to represent 26 letters as shown above, what does “143” mean?

BED? **BOX**? **VYN**?

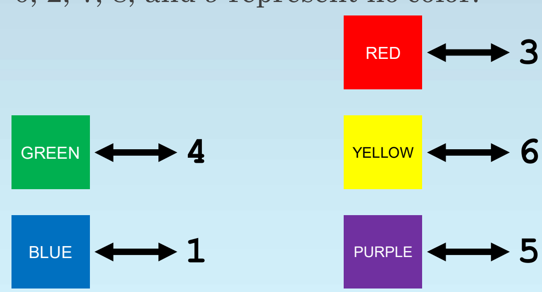
Computers are dumb—they cannot guess.

Second Answer: Representations Must be Unambiguous

Each pattern must represent **at most one thing**.

But Some Patterns May Represent Nothing

In the representation below, the digits 0, 2, 7, 8, and 9 represent no color.



Computers are Based on Electrons

In digital systems, electrons are **all we have** to represent information!

What can you ask about electrons?

How many electrons are in a certain place? (related closely to voltage)

So…
- Choose a ground: 0V by definition.
- Pick a higher voltage (called $\mathbf{V_{dd}}$).

Computer Representations are Based on BInary digiTs

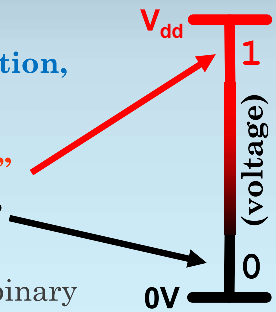
Now ask:

At a given physical location, what is the voltage?

Voltage near V_{dd} is a "1."

Voltage near 0V is a "0."

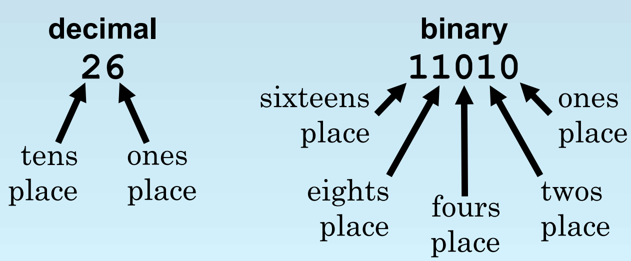
The location thus holds a binary digit, which we call a **bit**.



Physical Locations Enable Place Value

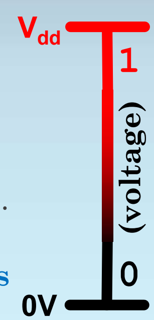
Each bit is somewhere on a computer chip.

So using positional / **place value** is natural.



Represented by What? The Answer is Always "Bits"

Remember:
- Electrons are all we have inside computers.
- No decimal, no hexadecimal, no letters, no real numbers, no colors.
- **ALL computer representations are based on bits.**



A Question for You: How Many Bits do We Need?

How many bits do we need to represent a whole number in the range…

- from **0 to 31**?
  - 32 different integers
  - so **we need 5 bits** (2^5 = 32 bit patterns)
- from **0 to 100**?
  - 101 different integers
  - so **we need 7 bits** (2^7 = 128 bit patterns)

We Need One Bit Pattern for Each Possible Thing

Trick question: How many bits do we need to represent two books?

- **The Collected Works of Shakespeare**
- **Our textbook by Patt & Patel**
  - 2 different books
  - so **we need only 1 bit!** (2^1 = 2 bit patterns)

What matters is the **number of things**, not what those things are.

How Many Bits Do We Need to Represent N Things?

Let's test your understanding (and generalize)!

How many bits do we need to represent…

- a whole number from **1000 to 1100**?
  101 different integers, so **7 bits** (2^7 = 128)
- one of **199 flavors of ice cream**?
  199 different flavors, so **8 bits** (2^8 = 256)
- **a living person**?
  7-8 billion people, so **33 bits** (2^{33} > 8 billion)
- **N things**?
  $\lceil \log_2 N \rceil$ (ceiling / integer at least as large as log base 2 of **N**)