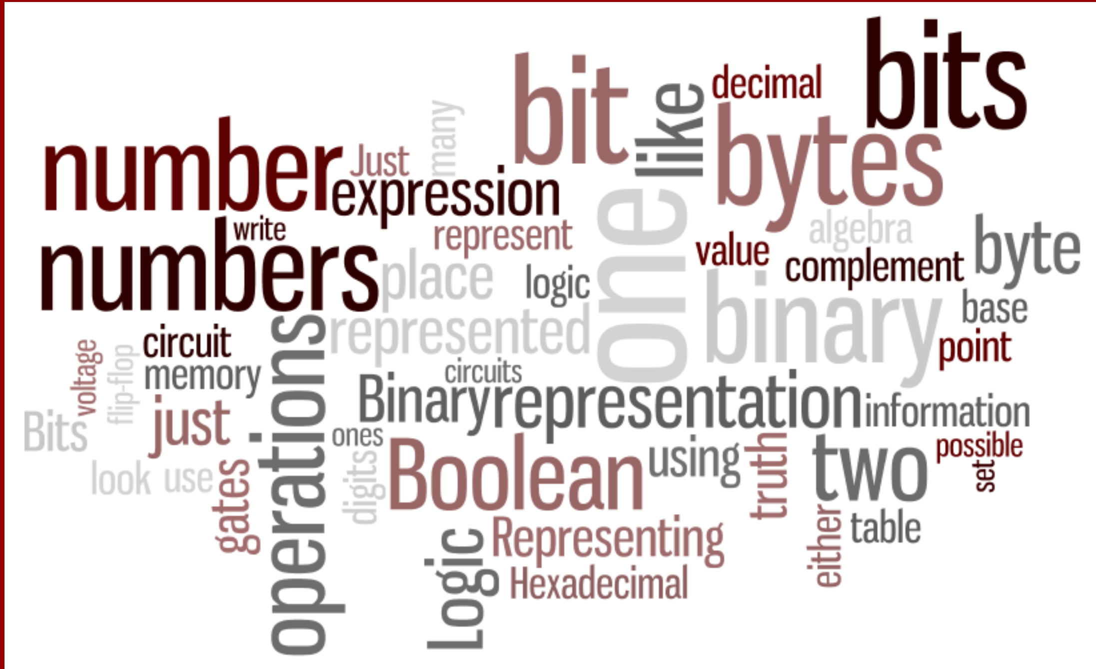
9/7/18

Journal: Explain the process of "carrying" a digit when adding. For example, when we do 26 + 99, why do we "carry the 1 to the tens place, why do we "carry" a 1 to the hundreds place?



From the Bottom Up: It's All Just Bits

CS Matters in Maryland
CS Principles

Objectives:

Content: I will be able to work with numbers from different numbering systems.
Social: I will explain my reasoning to other people.
Language: I will use my words to explain my addition a nd calculation process to others in the class.

Just Bits

- Inside the computer, all information is stored as **bits**
  - A "bit" is a single unit of information
  - Each "bit" is set to either zero or one
- How do we get complex systems like Google, Matlab, and our cell phone apps?

Abstraction!

Objectives:

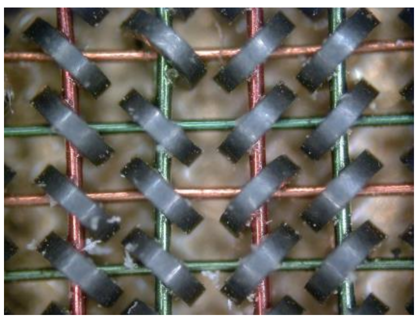
Content: I will be able to work with numbers from different numbering systems.

Social: I will explain my reasoning to other people.

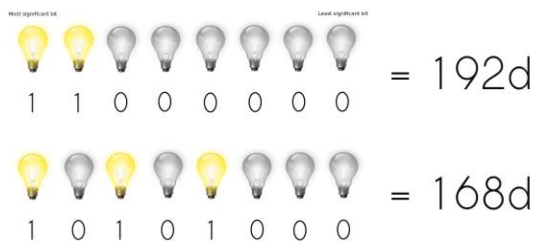
Language: I will use my words to explain my addition a nd calculation process to others in the class.

What's a Bit Between Friends?

- Bits can be represented in *many* different ways
- They are all equivalent *abstractions*

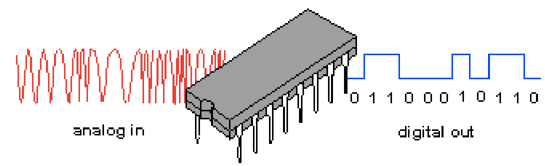


Magnetic core memory
(each core represents one bit)
wikipedia.com



Light signals
delightlylinux.files.wordpress.com

From Computer Desktop Encyclopedia
© 1998 The Computer Language Co. Inc.



Electrical voltages
babbage.cs.qc.edu

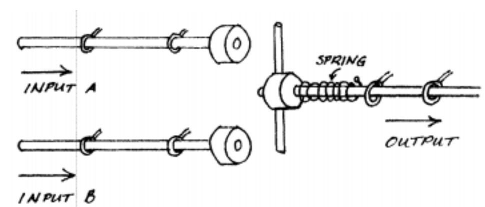


FIGURE 4
Mechanical implementation of the OR function

Mechanical devices
Hillis, *The Pattern on the Stone*

Objectives:
Content: I will be able to work with numbers from different numbering systems.
Social: I will explain my reasoning to other people.
Language: I will use my words to explain my addition a nd calculation process to others in the class.

The Jacquard Loom

Objectives:

Content: I will be able to work with numbers from different numbering systems.

Social: I will explain my reasoning to other people.

Language: I will use my words to explain my addition a nd calculation process to others in the class.

Storing Bits

- How are these "bits" stored in modern computers?
- A bit is just an electrical signal or voltage (by convention: "low voltage" = 0; "high voltage" = 1)
- A circuit called a "flip-flop" can store a single bit
  - A flip-flop can be "set" (using an electrical signal) to either 0 or 1
  - The flip-flop will hold that value until it receives a new signal telling it to change
- Bits can be operated on using ***gates*** (which "compute" a function of two or more bits)

Objectives:

Content: I will be able to work with numbers from different numbering systems.
Social: I will explain my reasoning to other people.
Language: I will use my words to explain my addition a nd calculation process to others in the class.

Orders of Magnitude

- One 0/1 (“no/yes”) “bit” is the basic unit of memory
  - Eight (2^3) bits = one byte
  - 1,024 (2^{10}) bytes = one kilobyte (1K)*
  - 1,024K = 1,048,576 (2^{20} bytes) = one megabyte (1M)
  - 1,024K (2^{30} bytes) = one gigabyte (1G)
  - 1,024 (2^{40} bytes) = one terabyte (1T)
  - 1,024 (2^{50} bytes) = one petabyte (1P)
  - ... 2^{80} bytes = one yottabyte (1Y?)

Objectives:

Content: I will be able to work with numbers from different numbering systems.
Social: I will explain my reasoning to other people.
Language: I will use my words to explain my addition a nd calculation process to others in the class.

Scaling Up Memory

- Computer ***chip***:
  - Many (millions) of circuits
  - Etched onto a silicon wafer using VLSI (Very Large-Scale Integration) technology
  - Lots of flip-flops or DRAM devices == memory chip

Objectives:

Content: I will be able to work with numbers from different numbering systems.
Social: I will explain my reasoning to other people.
Language: I will use my words to explain my addition a nd calculation process to others in the class.

Scaling Up Memory

- Each byte has an ***address*** (and we use binary numbers to represent those addresses…)
  - An address is represented using a ***word***, which is typically either;
    - 2 bytes (16 bits) -- earliest PCs
      - Only 64K combinations ⇒ memory is limited to 64K (65,535) bytes!
    - 4 bytes (32 bits) -- first Pentium chips
      - This brings us up to 4G (4,294,967,295) bytes of memory!
    - 8 bytes (64 bits) -- modern Pentium chips
      - Up to 16.8 *million terabytes* (that's 18,446,744,073,709,551,615 bytes!)

Objectives:

Content: I will be able to work with numbers from different numbering systems.

Social: I will explain my reasoning to other people.

Language: I will use my words to explain my addition a nd calculation process to others in the class.

Storing Information

- One bit can't tell you much… (just 2 possible values)
- Usually we group 8 bits together into one "byte"
- ***QUESTION: How many possible values (combinations) are there for one byte?***

$2 \cdot 2 \cdot 2 = 2^3$ 8

$2 \cdot 2 \cdot 2 \cdot 2$

2^4

$1 = 2$

$2 = 4$

$3 = 8$

$4 = 2^4 = 16$

Objectives:

Content: I will be able to work with numbers from different numbering systems.

Social: I will explain my reasoning to other people.

Language: I will use my words to explain my addition a nd calculation process to others in the class.

Storing Information

- A byte can just be thought of as an 8-digit binary (base 2) number
  - Michael Littman's octopus counting video [3 min]
- Low-order or least significant bit == ones place
  - Next bit would be “10s place” in base 10 -- ***what about base 2?***
- High-order bit or most significant bite in a byte == ***?? place***

Objectives:

Content: I will be able to work with numbers from different numbering systems.
Social: I will explain my reasoning to other people.
Language: I will use my words to explain my addition a nd calculation process to others in the class.

Conversions

- Binary ➔ decimal: multiply each digit by its place value and add the results
- Decimal ➔ binary:
  - Find the largest power of two that is less than or equal to the decimal number
  - Put a one in that place column in the binary number
    - Add spaces for the smaller binary digits
    - E.g., if the largest multiple of two that fits is 256, you would write:

      1 __ __ __ __ __ __ __ __
  - Find the next smallest multiple of two
  - Put a one in that place column (and a zero in any columns you skipped)

Objectives:

Content: I will be able to work with numbers from different numbering systems.
Social: I will explain my reasoning to other people.
Language: I will use my words to explain my addition a nd calculation process to others in the class.

Conversions

- **EXERCISE**: You try it!
  - Write 97_{10} in binary

    128

    01100001

    97
    - 64
    33

    $2^8 - 1 = 255$

  - Write 11001110_2 in decimal

    2^1 2^0

Objectives:

Content: I will be able to work with numbers from different numbering systems.

Social: I will explain my reasoning to other people.

Language: I will use my words to explain my addition a nd calculation process to others in the class.

Practice with Binary

Objectives:

Content: I will be able to work with numbers from different numbering systems.
Social: I will explain my reasoning to other people.
Language: I will use my words to explain my addition a nd calculation process to others in the class.

Homework

DECIMAL TO BINARY

41	
30	
5	
10	
99	
123	
244	
13	
78	
143	
94	
58	
190	
202	
6	

BINARY TO DECIMAL

1111	
1101	
100101	
10	
00111100	
100	
110	
11111101	
1000100	
100001	
11010	
10101011	
10011001	
1110111	
11111	

Objectives:

Content: I will be able to work with numbers from different numbering systems.
Social: I will explain my reasoning to other people.
Language: I will use my words to explain my addition a nd calculation process to others in the class.