Bits, Bytes and Integers – Part 1

15-213/15-513: Introduction to Computer Systems
2nd Lecture, May 18, 2022

Instructors:

Zack Weinberg

Waitlist questions

- All waitlist questions should go to Amy Weis
  alweis@andrew.cmu.edu

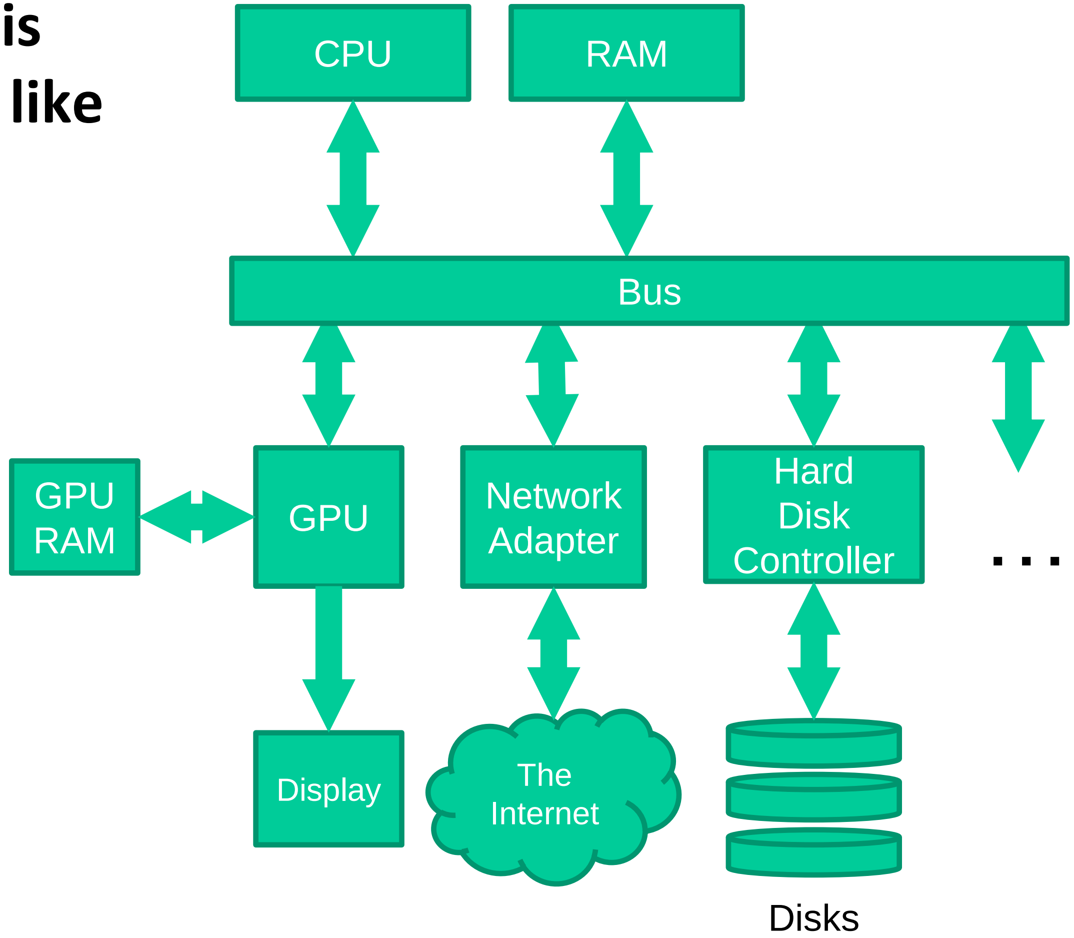
- Please don’t contact the instructors with waitlist questions.

Reminder about in-person attendance

- This room is full.
- In-person classes are primarily for undergraduate students who are enrolled in 15-***2***13.
- If I told you in email to enroll in 513 for a reason *other than* "you're a graduate student," you are also entitled to be in this room.
- If you're enrolled in 15-***5***13 because you are a graduate student, however, you may attend *only if there is space*.

Roadmap – Inside a Computer

- You may have seen this block diagram, or one like it, before.

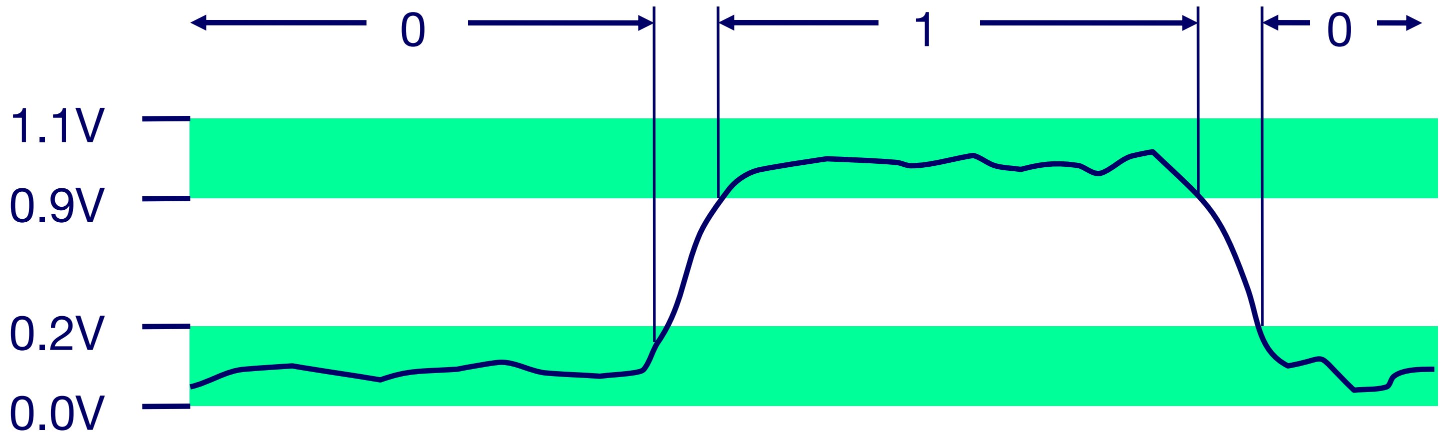


Today: Bits, Bytes, and Integers

- **Representing information as bits**
- Bit-level manipulations
- Integers
  - Representation: unsigned and signed
  - Conversion, casting
  - Expanding, truncating
  - Addition, negation, multiplication, shifting
  - Summary
- Representations in memory, pointers, strings

Everything is bits

- **Each bit is 0 or 1**
- **By encoding/interpreting sets of bits in various ways**
  - Computers determine what to do (instructions)
  - … and represent and manipulate numbers, sets, strings, etc…
- **Why bits? Electronic Implementation**
  - Easy to store with bistable elements
  - Reliably transmitted on noisy and inaccurate wires



For example, can count in binary

- **Base 2 Number Representation**
  - Represent 15213_{10} as 11101101101101_2
  - Represent 1.20_{10} as $1.0011001100110011[0011]..._2$
  - Represent 1.5213×10^4 as $1.1101101101101_2 \times 2^{13}$

Encoding Byte Values

- **Byte = 8 bits**
  - Binary 00000000_2 to 11111111_2
  - Decimal: 0_{10} to 255_{10}
  - Hexadecimal 00_{16} to FF_{16}
    - Base 16 number representation
    - Use characters ‘0’ to ‘9’ and ‘A’ to ‘F’
    - Write $FA1D37B_{16}$ in C as
      - 0xFA1D37B
      - 0xfa1d37b

Hex	Decimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

```
15213: 0011 1011 0110 1101
          3    B    6    D
```

Activity: binary, hexadecimal, twos complement

https://www.cs.cmu.edu/afs/cs/academic/class/15213-m22/www/activities/213_lecture2.pdf

Preview: Combining bytes…

C Data Type	Typical 32-bit	Typical 64-bit
`char`	1	1
`short`	2	2
`int`	4	4
`long`	4	8
`float`	8	8
`double`	8	8
pointer	4	8

Preview: … to make integers

	W			
	8	**16**	**32**	**64**
UMax	255	65,535	4,294,967,295	18,446,744,073,709,551,615
TMax	127	32,767	2,147,483,647	9,223,372,036,854,775,807
TMin	-128	-32,768	-2,147,483,648	-9,223,372,036,854,775,808

- **UMax = $2^w - 1$ where *w* is the number of bits ("word size")**
- **UMin = 0**
- **TMax = $2^{w-1} - 1$**
- **TMin = -2^{w-1}**
  - Asymmetric!
  - Because of zero