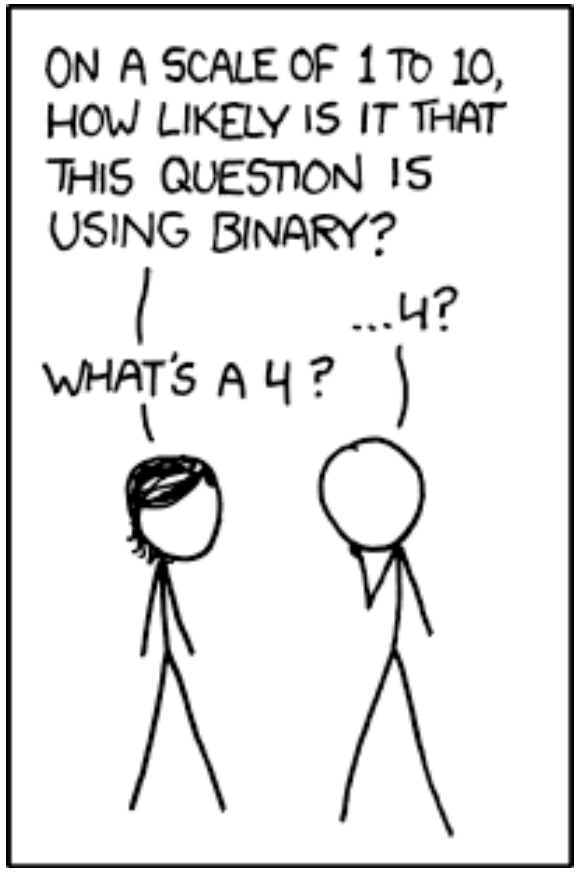
Memory, Data, & Addressing I

CSE 351 Autumn 2023

Instructor:
Justin Hsia

Teaching Assistants:

Afifah Kashif	Malak Zaki
Bhavik Soni	Naama Amiel
Cassandra Lam	Nayha Auradkar
Connie Chen	Nikolas McNamee
David Dai	Pedro Amarante
Dawit Hailu	Renee Ruan
Ellis Haker	Simran Bagaria
Eyoel Gebre	Will Robertson
Joshua Tan	



http://xkcd.com/953/

Relevant Course Information

- Upcoming deadlines
  - Pre-Course Survey and HW0 due tonight
  - HW1 due Monday (10/2)
  - Lab 0 due Monday (10/2)
    - This lab is *exploratory* and looks like a HW; the other labs will look a lot different
- Ed Discussion etiquette
  - For anything that doesn't involve sensitive information or a solution, post publicly (you can post anonymously!)
  - If you feel like you question has been sufficiently answered, make sure that a response has a checkmark

EPA

- Encourage class-wide learning!
- Effort
  - Attending support hours, completing all assignments
  - Keeping up with Ed Discussion activity
- Participation
  - Making the class more interactive by asking questions in lecture, section, support hours, and on Ed Discussion
- Altruism
  - Helping others in section, support hours, and on Ed Discussion

Memory & Data I

Lesson Summary (1/2)

- Memory is a long, *byte-addressed* array
  - Word size bounds the size of the *address space* and memory
  - Address of chunk of memory given by address of lowest byte in chunk
- Endianness determines memory storage order for multi-byte data
  - Least significant byte in lowest (little-endian) or highest (big-endian) address of memory chunk
- Programming Data
  - Variable declaration allocates space for data type size
  - Assignment results in value being put in memory location

Lesson Summary (2/2)

- Terminology:
  - byte-oriented memory, word size, address, address space
  - most-significant bit (MSB), least-significant bit (LSB), big-endian, little-endian

- Learning Objectives:
  - (Define the concept of pointers and) their significance in computer memory organization.
  - (Design code that can correctly) interpret and manipulate multi-byte data in both little-endian and big-endian byte orderings.

- What lingering questions do you have from the lesson?

Memory & Data I – Context

Modern System Details

- Current x86-64 systems use **64-bit (8-byte) words** ("64-bit machines")
  - Potential address space: $\mathbf{2^{64}}$ addresses
    2^{64} bytes ≈ **1.8 x 10^{19} bytes**
    = 18 billion billion bytes = 18 EB (exabytes)
  - Actual physical address space: **48 bits**
    - This is sufficient space for now and allows for some operating system tricks
    - Example address: `0x 7f fc 3d d5 06 94`

- There's a lot more to this story… stay tuned for virtual memory!

Discussion Question

- Discuss the following question(s) in groups of 3-4 students
  - I will call on a few groups afterwards so please be prepared to share out
  - Be respectful of others' opinions and experiences

- Over time, computers have grown in word size:

Word size	Instruction Set Architecture	First? Intel CPU	Year Introduced
8-bit	??? (Poor & Pyle)	Intel 8008	1972
16-bit	x86	Intel 8086	1978
32-bit	IA-32	Intel 386	1985
64-bit	IA-64	Itanium (Merced)	2001
64-bit	x86-64	Xeon (Nocona)	2004

  - What do you think were some of the *causes*, *advantages*, and *disadvantages* of this trend?

Memory & Data I – Practice

Group Work Time

- During this time, you are encouraged to work on the following:
  1) If desired, continue your discussion
  2) Work on the lesson problems (solutions at the end of class)
  3) Work on the homework problems

- Resources:
  - You can revisit the lesson material
  - Work together in groups and help each other out
  - Course staff will circle around to provide support

Practice Questions (1/2)

- By looking at the bits stored in memory, I can tell what a particular 4 bytes is being used to represent.

  A. True B. False

  many possible encoding schemes

- We can fetch a piece of data from memory as long as we have its address.

  A. True B. False

  need: ① address ✓
  ② data size ✗

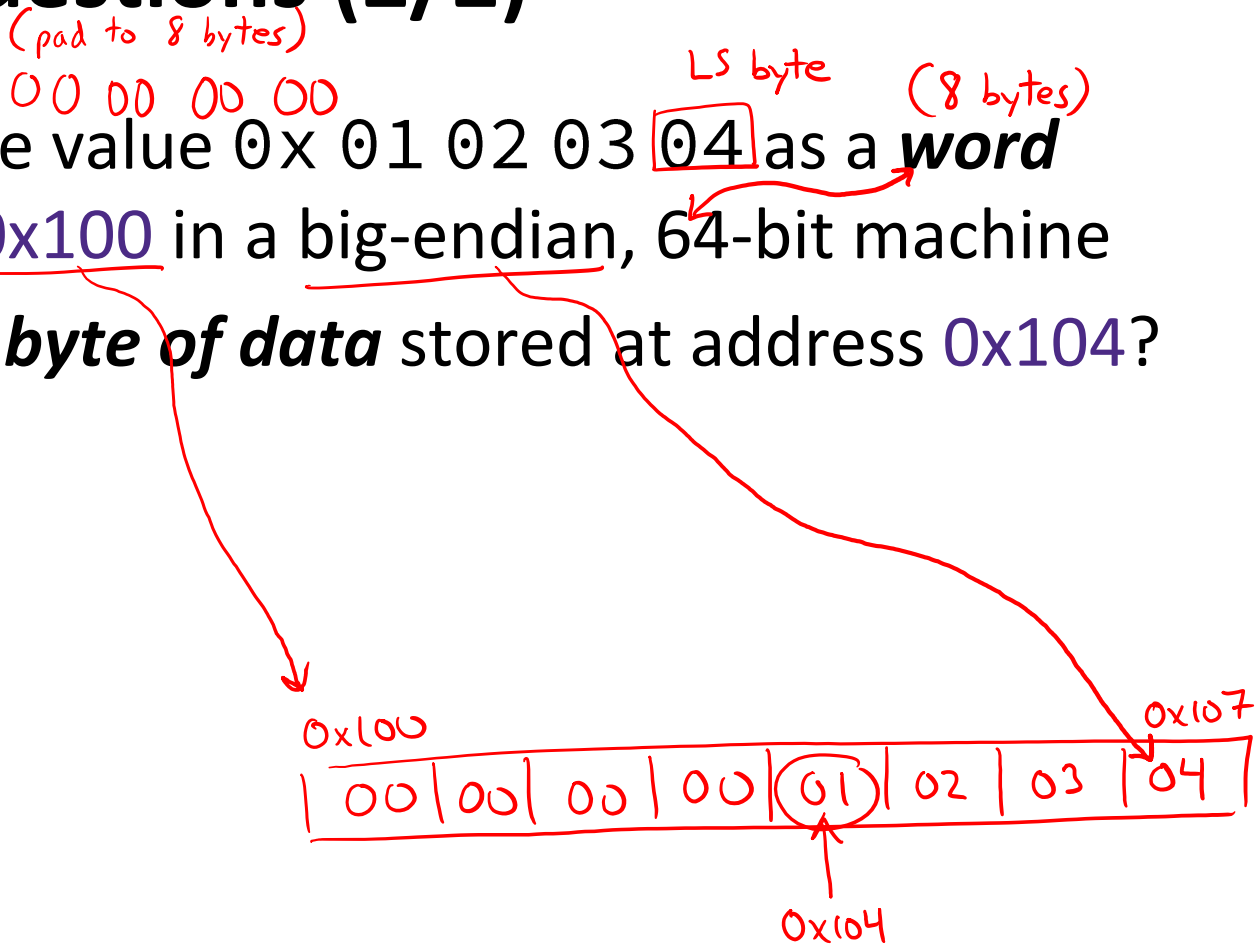
- Which of the following bytes have a most-significant bit (MSB) of 1?

  → 8 bits = 2 hex digits

  0b 0110 0011 — A. 0x63
  0b 1001 0000 — B. 0x90
  0b 1100 1010 — C. 0xCA
  0b 0000 1111 — D. ~~0xF~~ 0x0F

Practice Questions (2/2)

- We store the value 0x 01 02 03 04 as a ***word*** at address 0x100 in a big-endian, 64-bit machine
- What is the ***byte of data*** stored at address 0x104?



A. 0x04
B. 0x40
C. 0x01
D. 0x10
E. We're lost…