Computer Organization and Programming for Scientific Computing

Lecture 2

Ramani Duraiswami

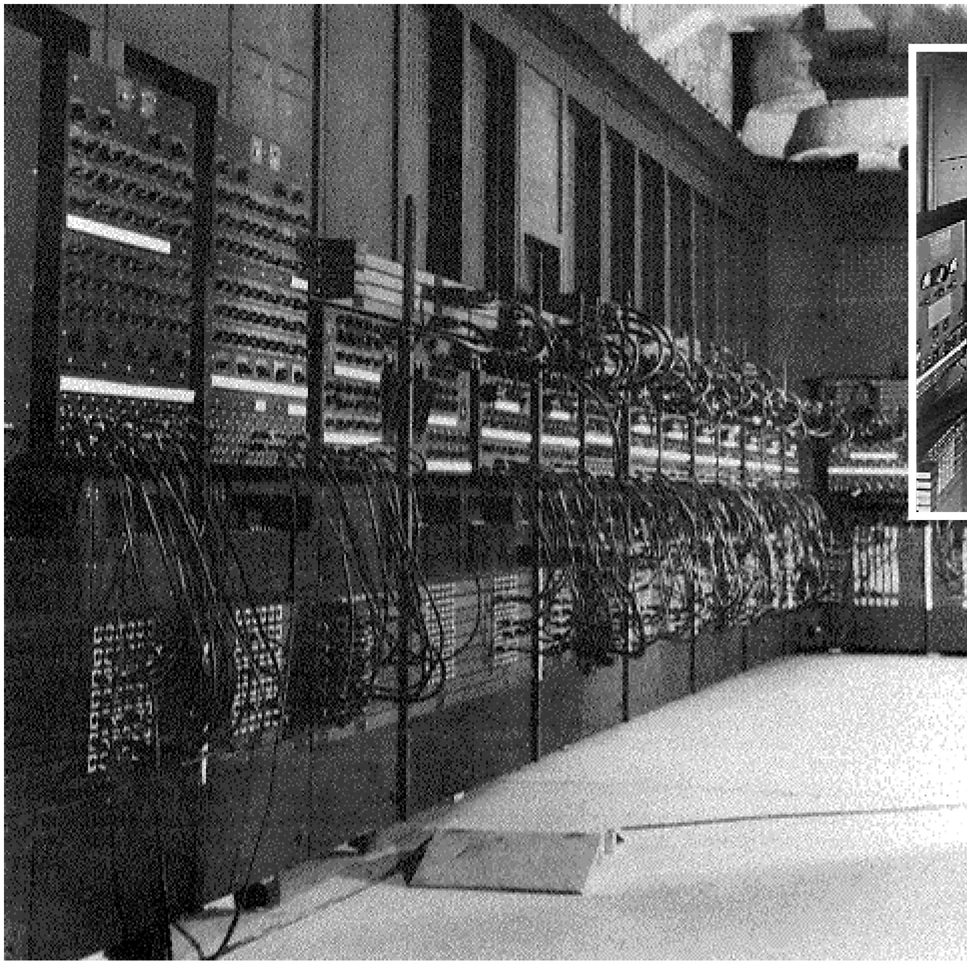
What is a computer?

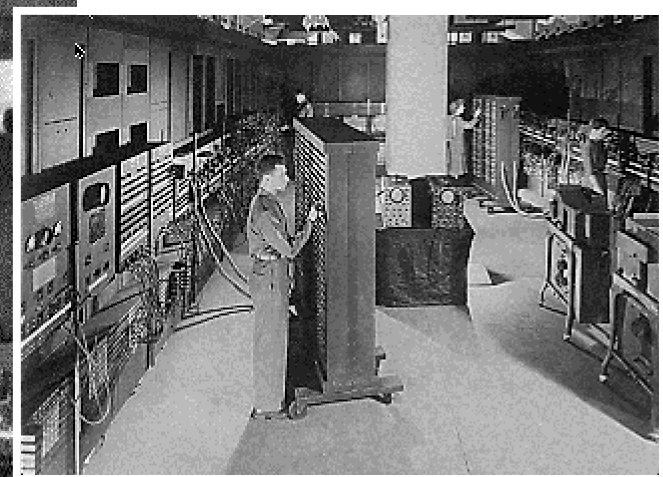
- History
  - Initially a person who computed.
  - Mechanical calculators
  - Electrical calculators
  - ENIAC
  - Von Neumann machine (Eckart, Mauchly)
- Abstraction
  - Turing machines
    - Developed to answer Hilbert's decision problem
  - Representation of functions

Earliest computers

- Little more than calculators
- Programming required actual rewiring
- Would take a long time to program a new calculation
- Programming was “physical” requiring rewiring
- ENIAC developed at the end of WW-2
  - Used decimal system!

ENIAC
Electronic Numerical Integrator and Computer





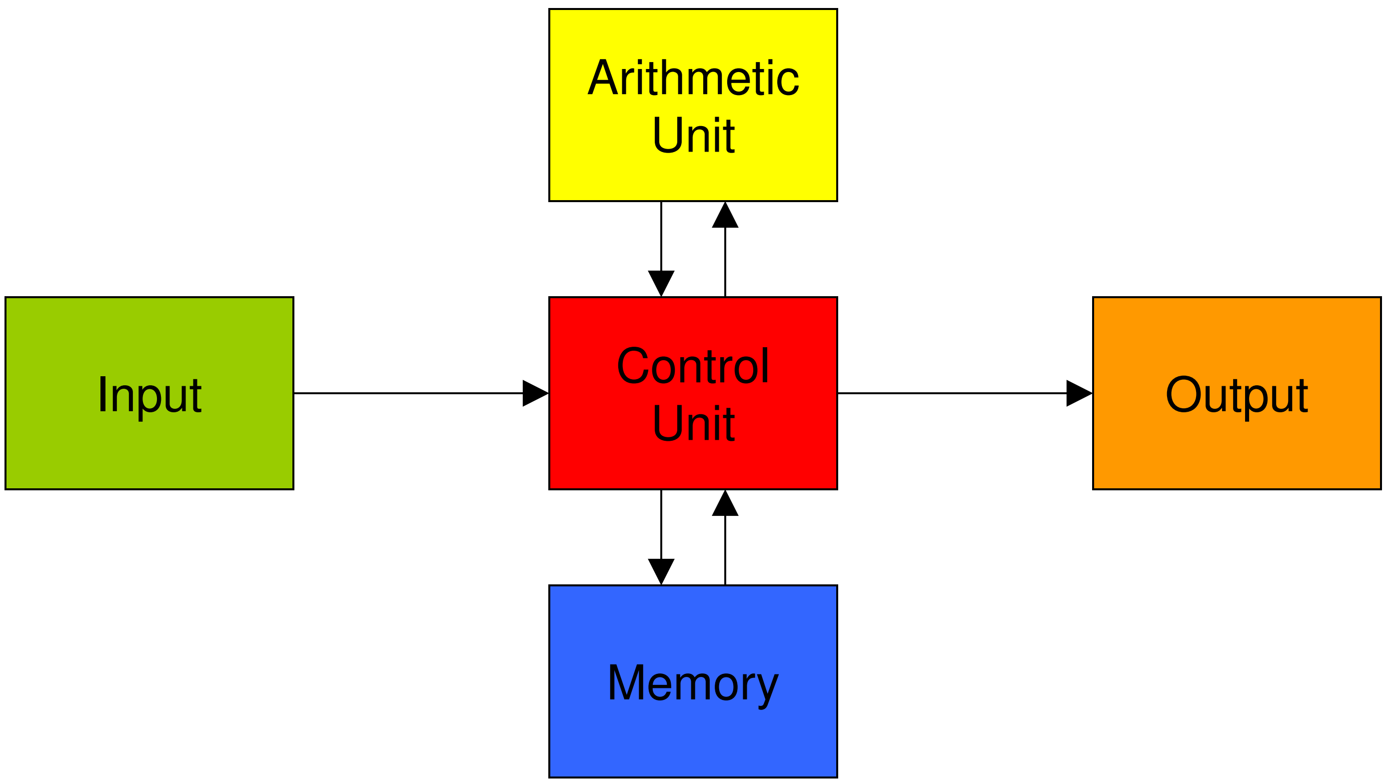
(Virginia Tech –
History of Computing)

Von-Neumann machine

- Actually based on ideas of several people
  - Turing
  - Eckert & Mauchly
  - Johann von Neumann (he wrote the summary report)
- Both instructions and data stored sequentially in binary in the memory unit
- Instructions were executed sequentially except where a conditional instruction would cause a jump to an instruction elsewhere
- Fetch-Decode-Execute
- Binary switching circuits for computation and control
- This is how all modern-day computers work
  - Called the von-Neumann machine
  - Eckert & Mauchly were furious it was not named after them
    - They claimed it was their idea first, but could not implement it during the war due to time constraints

Based on slides from SUNY, Buffalo

The stored-program concept



Simple computer model

- **CPU**
  - Job is to manipulate information
  - Information is stored in bits
- **Bits** – basic unit
  - 0/1
- Bits are too small a unit for storage, so usually stored as **words**
- Words are composed of **bytes** – 8 bits
- 32 bit word => 4 bytes
  - 2^{32} = 4;294;967;296 = 4.3×10^9 pieces of information
- Memory in the CPU is **registers** which is near and fast
- Main memory is a bit more distant and slower

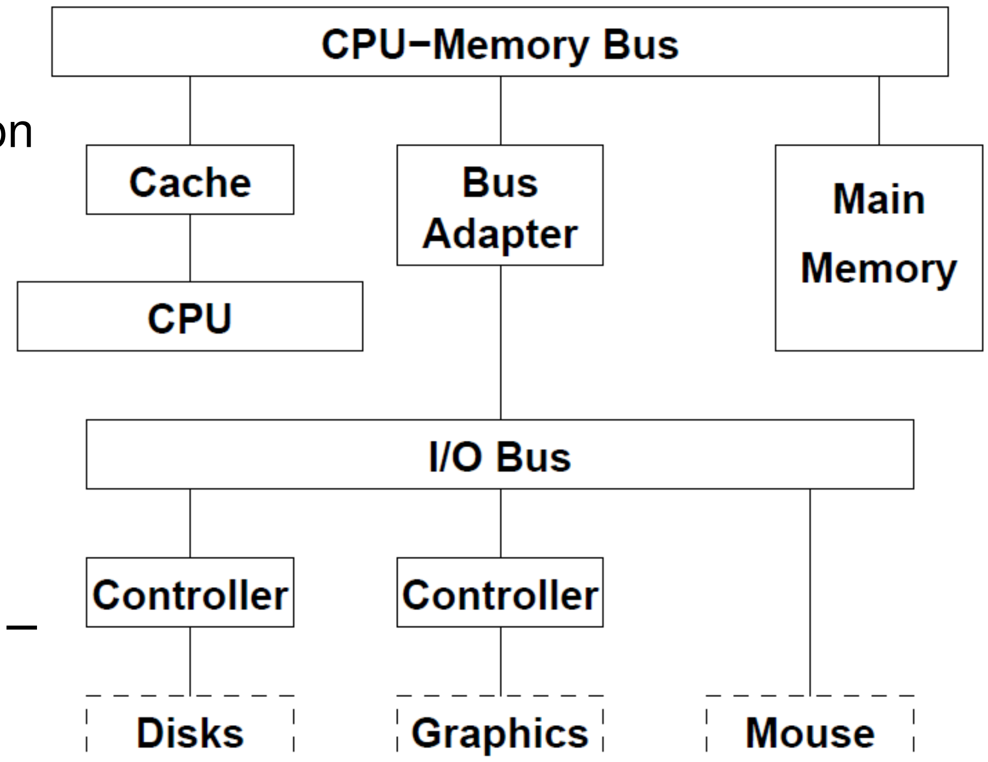


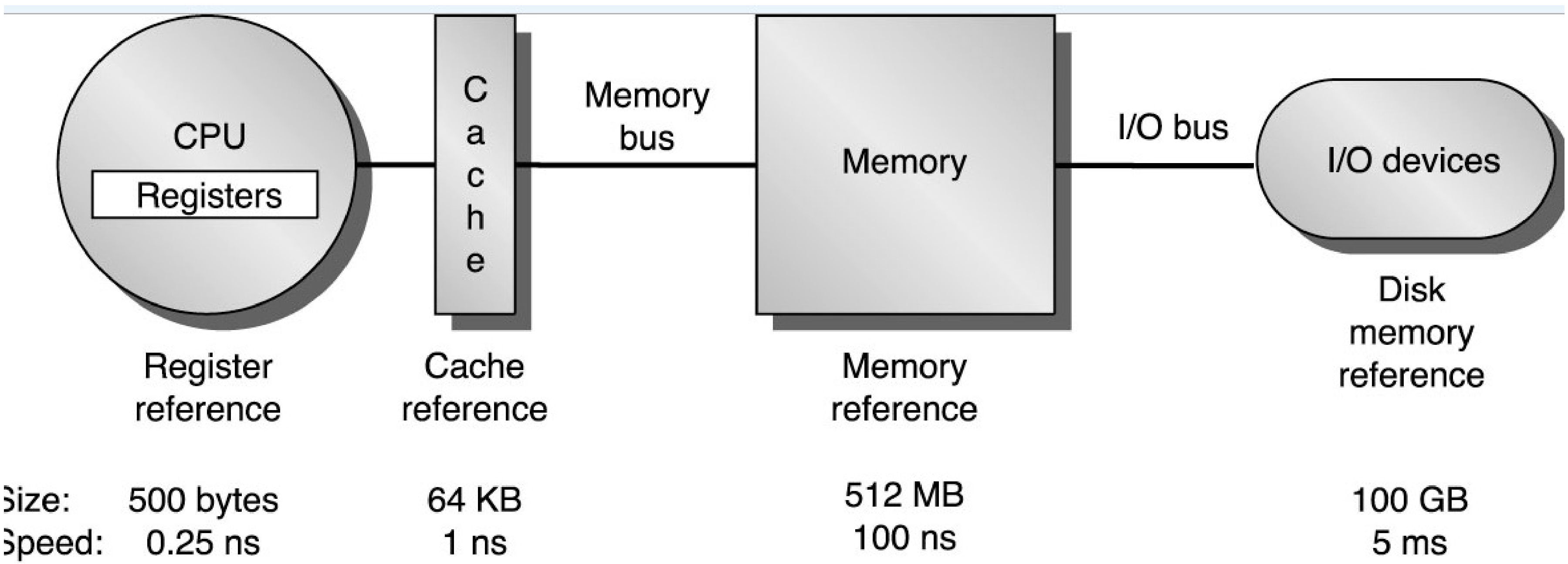
Figure 1.1: A simple computer

Simple Computer Model

- Information is stored in in registers and in main memory.
- CPU processes information by executing sequences of **instructions** that change and combine the contents of its register and of memory.
- The operation of a CPU is rigidly synchronized by a **clock**.
- Time between consecutive ticks of the clock is a **cycle**.
- The clock speed determines how fast the computer executes instructions.
- Other things being equal, faster the clock, faster the computer
  - Often other things often are not equal!

Memory System

- The memory hierarchy
  - Registers, cache, main memory, I/O devices (local disks), Networked disks, cloud …



Memory system

- Although registers are the working memory of the CPU, there are not enough of them
- Time to read to main memory is considerably less than the CPU cycle.
- To take care of this memory systems use a high-speed buffer called the **cache** where data and instructions are pre-fetched and stored
- Blocks of data are transferred from main memory to the cache as needed.
  - read and written at high speed.
- When data and instructions overflow main memory, they must be written out to a slower storage device such as a disk.
- Essentially main memory is then a cache for the disk
- Hierarchical system
- Can add more levels to the hierarchy
  - Networked storage
  - Internet storage

Memory addresses and instruction sizes

- Each location has an address
- 8 bit addresses and instructions 8088
- 16 bit addresses and instructions 8086, 80286
- 32-bit addresses and instructions, 80386
- 64 bit current Intel chips

Memory

- Cache is fast, but slower than memory
- Good ways to implement algorithms ensure that data that is necessary is available in cache for loading to registers
- 2^32 address spaces correspond to about 4 GB of space
  - Beginning to be exceeded in modern PCs
- 2^64 is much larger than many things in the world: 1.845×10^{19}, 18.45 exabytes

Bus & Peripheral devices

- Buses move bits between its parts.
  - Disk to memory
  - Memory to cache
  - Cache to registers
- Bus-width is the number of bits that can be simultaneously moved
- To transmit a 32 bit word in one step, width is 32 data lines.
- The rate at which information can be transferred over the bus is the bandwidth or throughput of the bus.
- Several devices can connect to the bus and try to send their information over it
- Bus must also provide a protocol by which two devices can talk to each other, and each device must have hardware to use this protocol.
- Peripheral devices are the way the computer communicates with the outside world.
  - mouse, printer, disk
- Complicated, high-speed peripherals like a disk generally have a controller -- a small computer in its own right that manages I/O transactions.

Processes and O/S

- A process is a program in action on the machine: basic unit of computation
- More than one process can run on a machine, but only one use the CPU at a time.
- Processes that exist at the same time are called concurrent processes.
- Operating system coordinates concurrent processes, stopping and starting them as required.
- O/S runs in a privileged mode that allows it to do things ordinary processes cannot.
- Interrupts are the way of directing the operating system's attention to conditions that require immediate attention. When an interrupt occurs, the current process is suspended and control turned over to an interrupt handler that does whatever is necessary.
  - For example, when a process makes a request for input, it generally cannot proceed until the input data arrives.
  - Rather than tie up the CPU, the operating system (which also handles I/O requests) suspends the execution of the requesting process, and starts another process.
  - after the data arrives, the operating system reactivates the original process.
- Swapping of processes is called a context shift.
- Operating system runs on the CPU as process in its own right.
- CPU runs in two modes. User programs run in user mode; the operating system runs in a privileged executive or system mode.
- Fortunately, a knowledge of these details is rarely needed in scientific computing.

Fixed point representation

- How can we represent a number in a computer's memory?
- Fixed point is an obvious way:
- Used to represent integers on computers, and real numbers on some DSPs:
- Each **word** (storage location) in a machine contains a fixed number of digits.
- Example: A machine with a 6-digit word might represent 2005 as

0	0	2	0	0	5

- This only allows us to represent integers and uses a decimal system

Binary/Decimal/Octal/Hexadecimal

- Numbers can be represented in different bases
- Usually humans use decimal
  - Perhaps because we have ten fingers
- Computer memory often has two states
  - Assigned to 0 and 1
  - Leads to a binary representation
- Octal and Hexadecimal representations arise by considering 3 or 4 memory locations together
  - Lead to 2^3 and 2^4 numbers

Binary Representation

- Most computers use **binary (base 2)** representation.

  0 1 0 1 1 0

- Each digit has a value 0 or 1.
- If the number above is binary, its value is
- $1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$. (or 22 in base 10)
- Adding numbers in binary

	0	0	0	1	1
+	0	1	0	1	0
	0	1	1	0	1

Note the "**carry**" here!

$0 + 0 = 0$
$0 + 1 = 1$
$1 + 0 = 1$
$1 + 1 = 10$ (binary) $= 10_2 = 2$
$1 + 1 + 1 = 11$ (binary)$=11_2 = 3$

Negative numbers

- One way computers represent negative numbers is using the *sign-magnitude representation:*
- **Sign magnitude**: if the first bit is zero, then the number is positive. Otherwise, it is negative.
- 0 0 0 1 1 Denotes +11.
- 1 0 0 1 1 Denotes -11.

Range of fixed point numbers

Largest 5-digit (5 bit) binary number:	0	1	1	1	1	=15
Smallest:	1	1	1	1	1	=-15
Smallest positive:	0	0	0	0	1	=1

Overflow

If we try to add these numbers:

	0	1	1	1	1	=15
+	0	1	0	0	0	= 8

we get

1	0	1	1	1	=-7.

We call this "overflow": the answer is too large to store, since it is Outside the range of this number system.

- Fixed point arithmetic:
  - Easy: always get an integer answer.
  - Either we get exactly the right answer, or we can detect
  - overflow.
  - The numbers that we can store are equally spaced.
  - Disadvantage: **very** limited range of numbers.