Overview of Computer Systems

Hui Chen [a]

[a]CUNY Brooklyn College, Brooklyn, NY, USA

August 29, 2023

Outline

Acknowledgement

The content of most slides come from the authors of the textbook:

Null, Linda, & Lobur, Julia (2018). The essentials of computer organization and architecture (5th ed.). Jones & Bartlett Learning.

Table of Contents

Lesson Objectives

- Computer organization vs. architecture.
- Units of measure common to computer systems.
- Computer as a layered system.
- Components of von Neumann architecture and the function of basic computer components.
- Cloud computing
- Parallel computer

Table of Contents

Study Computer Architecture?

Most computing programs, such as Computer Science has a Computer Organization and Architecture class.

Why do we want to study computer organization and architecture?

Computer Organization vs. Computer Architecture

Did you say “studying Computer Organizations and Computer Architecture”, but our class is called “Principles of Computer Architecture”?

Computer Organization

To computer science people:

- To understand, how hardware interacts with software.
- To become familiar with how various circuits and components fit together to create working computer systems.
- To understand control signals:
  - physical aspects of computer systems (e.g., circuit design, control signals, memory types)

Simply put, how does a computer work, more precisely, how does the computer run my program?

Computer Architecture

- Focuses on the structure and behavior of the computer system
- Refers to the logical and abstract aspects of system as seen by the programmer.
  - e.g., instruction sets, instruction formats, data types, addressing modes.

Essentially, how do I design a computer that is able to run someone's programs efficiently?

Computer Organization vs. Computer Architecture

- Distinction between computer organization and computer architecture is not clear-cut.
- Computer science and computer engineering hold differing opinions,
  - They can stand alone; are interrelated and interdependent.

Study Computer Architecture?

Why do we want to study computer organization and architecture?

Comprehension of computer organization and architecture ultimately leads to a deeper understanding of computers and computation

- the heart and soul of computer science
- Design better programs, including system software such as compilers, operating systems, and device drivers.
- Optimize program behavior.
- Evaluate (benchmark) computer system performance.
- Understand time, space, and price trade-offs

Table of Contents

Principle of Equivalence of Hardware and Software:

"Anything that can be done with software can also be done with hardware, and anything that can be done with hardware can also be done with software."

where we assume speed and cost are not a concern

Basic Components

A computer has three major components

- A processor to interpret and execute programs
- A memory to store both data and programs
- A mechanism for transferring data to and from the outside world.

Specification of an Example Computer

Let's check out a computer's specification ... (e.g., a Dell computer or ...)

What do all of these mean?

Measurements of Capacity

Prefix	Symbol	Power of 10	Power of 2
Kilo	K	10^3	2^{10}
Mega	M	10^6	2^{20}
Giga	G	10^9	2^{30}
Tera	T	10^{12}	2^{40}
Peta	P	10^{15}	2^{50}
Exa	E	10^{18}	2^{60}
Zetta	Z	10^{21}	2^{70}
Yotta	Y	10^{24}	2^{80}

Speed of Clock

Hertz = clock cycles per second (frequency)

- 1MHz = 1,000,000Hz
- Processor speeds are measured in MHz or GHz.

Unit of Storage

Byte = a unit of storage

- 1 KB = 2^{10} Bytes = 1024 Bytes $\approx 2^{10}$ Bytes = 1000 Bytes
- 1 MB = 2^{20} = 1,048,576 Bytes
- Main memory (RAM) is measured in MB
- Disk storage is measured in GB for small systems, TB for large systems

Measures of Time and Space

Prefix	Symbol	Power of 10	Power of 2
Milli	m	10^{-3}	2^{-10}
Macro	μ	10^{-6}	2^{-20}
Nano	n	10^{-9}	2^{-30}
Pico	p	10^{-12}	2^{-40}
Femto	f	10^{-15}	2^{-50}
Atto	a	10^{-18}	2^{-60}
Zepto	z	10^{-21}	2^{-70}
Yocto	y	10^{-24}	2^{-80}

Example Measures of Time and Space

- Millisecond = 1 thousandth of a second
  - Hard disk drive access times are often 10 to 20 milliseconds.
- Nanosecond = 1 billionth of a second
  - Main memory access times are often 50 to 70 nanoseconds.
- Micron (micrometer) = 1 millionth of a meter
  - Circuits on computer chips are measured in microns.

Cycle Time vs. Frequency

Cycle time is the reciprocal of clock frequency, such as a bus operating at 133MHz has a cycle time of 7.52 nanoseconds:

133,000,000 cycles/second = 7.52ns/cycle

Specification of an Example Computer

Now Let's look at the computer's specification again ... (e.g., a Dell computer or ...)

What do all of these mean?

CPU

The microprocessor is the “brain” of the system. It executes program instructions.

What is the one shown? (Does it mention clock frequency)?

Memory

Large main memory capacity means you can run larger programs with greater speed than computers having small memories.

- RAM = random access memory. Time to access contents is independent of its location (e.g., memory capacity of 32 GB DDR3 SDRAM, or double data rate type three synchronous dynamic RAM)
- Cache is a type of temporary memory that can be accessed faster than RAM.

Bus

Does it mention the speed the data can be moved from RAM to CPU or vice versa?

Cache Memory

Example: "128KB L1 cache, 2MB L2 cache" also describes a type of memory.

To provide even faster access to data, many systems contain a special memory called cache.

A system can have multiple levels of cache memory, e.g., the level 1 (L1: that is built into the microprocessor chip and helps speed up access to frequently used data.)

L1 is smaller and (probably) faster than the L2 cache that is built-in memory chips situated between the microprocessor and main memory.

Secondary Storage (Disk)

Hard disk capacity determines the amount of data and size of programs you can store. (HDD vs. SSD)

Example: A 1TB HDD at 7200 RPM (7200 RPM is the rotational speed of the disk. Generally, the faster a disk rotates, the faster it can deliver data to RAM. (There are many other factors involved.))

Example: Interface with SATA (serial advanced technology attachment) which describes how the hard disk interfaces with (or connects to) other system components.

Ports and Expansion Ports

Ports allow movement of data to and from devices external to the computer.

Example: "10 USB ports, 1 serial port." Serial ports transfer data by sending a series of electrical pulses across one or two data lines.

Expansion slots. System buses can be augmented by dedicated I/O buses. PCI, peripheral component interface, is one such bus. HDMI port (High-Definition Multimedia Interface, used to transmit audio and video).

More about Ports

Serial ports send data as a series of pulses along one or two data lines.

Parallel ports send data as a single pulse along at least eight data lines.

USB, Universal Serial Bus, is an intelligent serial interface that is self-configuring. (It supports "plug and play.")

. . .

About Display

Explore it yourself ...

Summary

Basic performance characteristics of computer systems, including:

- Processor speed,
- Memory speed,
- Memory capacity, and
- Interconnection data rates

Table of Contents

Evolution of Computing Machinery

- The evolution of computing machinery has taken place over several centuries.
- The evolution of computers is usually classified into different generations according to the technology of the era.

Generations

- Generation 0: Mechanical Calculating Machines
- Generation 1: Vacuum Tube Computers
- Generation 2: Transistorized Computers
- Generation 3: Integrated Circuit Computers
- Generation 4: VLSI Computers

Moor's Law

Moore's Law (1965)

- Gordon Moore, Intel founder
- "The density of transistors in an integrated circuit will double every year."

Contemporary version:

- "The density of silicon chips doubles every 18 months."

But this "law" cannot hold forever ...

Table of Contents

Level Hierarchy

Writing complex programs requires a “divide and conquer” approach, where each program module solves a smaller problem.

Complex computer systems employ a similar technique through a series of virtual machine layers.

Abstract virtual machines

Level 6	User	Executable programs
Level 5	High-level language	C++, Java, Python, JavaScript, ...
Level 4	Assembly Language	Assembly code
Level 3	System Software	Operating Systems, library code
Level 2	Machine	Instruction Set Architecture
Level 1	Control	Microcode or hardware
Level 0	Digital Logic	Circuits, gates, ...

Characteristics of Abstract virtual machines

Each virtual machine layer is an abstraction of the level below it.

The machines at each level execute their own particular instructions, calling upon machines at lower levels to perform tasks as required.

Computer circuits ultimately carry out the work.

Level 6: User

Program execution and user interface level.

The level with which we are most familiar.

Level 5: High-Level Language Level

The level with which we interact when we write programs in languages such as C, Pascal, Lisp, and Java.

Level 4: Assembly Language Level

Acts upon assembly language produced from Level 5, as well as instructions programmed directly at this level.

Level 3: System Software Level

- Controls executing processes on the system.
- Protects system resources.
- Assembly language instructions often pass through Level 3 without modification.

Level 2: Machine Level

- Also known as the Instruction Set Architecture (ISA) Level.
- Consists of instructions that are particular to the architecture of the machine.
- Programs written in machine language need no compilers, interpreters, or assemblers.

Level 1: Control Level

- A control unit decodes and executes instructions and moves data through the system.
- Control units can be microprogrammed or hardwired.
- A microprogram is a program written in a low-level language that is implemented by the hardware.
- Hardwired control units consist of hardware that directly executes machine instructions.

Level 0: Digital Logic Level

- This level is where we find digital circuits (the chips).
- Digital circuits consist of gates and wires.
- These components implement the mathematical logic of all other levels.

Table of Contents

Virtual Computing Platforms

Cloud computing: virtual computing platforms

- Saas (Sofware As A Service): Level 6
- Paas (Platform As A Service): Level 3 – 5
- IaaS (Infrastructure As A Service): Level 0 – 2

Table of Contents

Invention of The von Neumann Model

The invention of stored program computers has been ascribed to a mathematician, John von Neumann

Stored-program computers have become known as von Neumann Architecture systems.

Characteristics of the von NeuMann Model

The stored-program computers have the following characteristics:

- Three hardware systems:
  - A central processing unit (CPU)
  - A main memory system
  - An I/O system
- The capacity to carry out sequential instruction processing.
- A single data path between the CPU and main memory – This single path is known as the von Neumann bottleneck.

Fetch-Decode-Execute

These computers employ a fetch-decode-execute cycle to run programs as follows

1. The control unit fetches the next instruction from memory using the program counter to determine where the instruction is located.
2. The instruction is decoded into a language that the ALU can understand.
3. Any data operands required to execute the instruction are fetched from memory and placed into registers within the CPU.
4. The ALU executes the instruction and places results in registers or memory.

Modified von Neumann Architecture

Adding a System Bus.

1. Data bus moves data from RAM to CPU registers (and vice versa).
2. Address bus holds address of the data that the data bus is currently accessing.
3. Control bus carries the necessary control signals that specify how the information transfer is to take place.

Table of Contents

Non-von Neumann Models

- Conventional stored-program computers have undergone many incremental improvements over the years.
- These improvements include adding specialized buses, floating-point units, and cache memories, to name only a few.
- But enormous improvements in computational power require departure from the classic von Neumann architecture.
- Adding processors is one approach.

Parallel Processing

A common method of providing increased computational power.

New Models

DNA computers, quantum computers, and dataflow systems.

At this point, it is not yet clear whether any of these systems will provide the basis for the next generation of computers.

Parallel Processors and Parallel Computing

Multicore architectures are parallel processing machines that allow for multiple processing units (often called cores) on a single chip.

What is a core?

Each processing unit has its own ALU and set of registers, but all processors share memory and some other resources.

Multi-core Computers

- Multiple cores, or multicore, provides the potential to increase performance without increasing the clock rate.
- Example: With Dual core, caches became larger,
  - it made performance sense to create two and then three levels of cache on a chip,
  - with the first-level cache dedicated to an individual processor, and
  - levels two and three being shared by all the processors.

Multi-core Computers: Multitasking

Multiple cores does not mean it will run your programs more quickly.

- Application programs (including OS) must be written to take advantage of multiple processing units.

Multicore computers are very useful for multitasking

- Example: you may be reading email, listening to music, browsing the Web, and burning a DVD all at the same time.
- These "multiple tasks" can be assigned to different processors and carried out in parallel, provided the operating system is able to manipulate many tasks at once.

Multi-core Computers: Multithreading

Multithreading can also increase the performance of any application with inherent parallelism.

- Programs are divided into threads, which can be thought of as mini-processes.

Example: GUI applications are often multithreaded

- one thread can download text,
- while each image is controlled and downloaded by a separate thread.

If an application is multithreaded, separate threads can run in parallel on different processing units.

Multi-core Cmoputers: Programming Challenge

- If parallel machines and other non-von Neumann architectures give such huge increases in processing speed and power, why isn't everyone using them everywhere?
- The answer lies in their programmability.
  - Advances in operating systems that can utilize multiple cores have put these chips in laptops and desktops that we can buy today;
  - However, true multiprocessor programming is more complex than both uniprocessor and multicore programming and requires people to think about problems in a different way, using new algorithms and programming tools.

Table of Contents

Summary and Questions

Provide an overview of computer organizations and architecture that we shall investigate in a greater length in this semester:

- Components of computer systems
- CPU, memory, and disk
- Instruction set architecture
- von-Neumann architecture and non-von-Neumann architecture
- Units of measurements

Questions?