Computer Systems

Hardware, Software and Layers of Abstraction

Automation & Computers

• Fundamental question of computer science: What can be automated?
• Computers automate processing of information
• Computer is general-purpose machine that can solve different types of problems
  – Hardware: physical aspects of system
  – Software: set of programs that instruct hardware to perform problem-solving tasks

Computer Organization

• How components fit together to create working computer system
• Encompasses physical aspects of computer systems
• Concerned with how computer hardware works

Computer Architecture

• Structure & behavior of computer system
• Logical aspects of system implementation as seen by programmer
• Concerned with how computer is designed
• Combination of hardware components with instruction set architecture; ISA is interface between software that runs on machine & hardware that executes it

Computer Hardware

• Basic hardware components: CPU, memory, I/O devices
• Information flows from one component to another via a group of connecting wires called the bus

Terminology & Units of Measure

• Basic unit of data storage in computer is the byte
• Memory size (both primary and secondary) usually given in multiples of bytes:
  – 1 K = 1 kilobyte = 1024 bytes
  – 1 M = 1 megabyte = 1,048,576 bytes
  – 1 G = 1 gigabyte = 1,073,741,824 bytes
  – 1 T = 1 terabyte = 1,099,511,627,776 bytes
Terminology & Units of Measure

- Processor speed is usually given in terms of cycles per second (hertz) or instructions per second
  - 1 MHz = 1 megahertz, or 1000 hertz
  - 1 gigahertz = 1 million hertz
  - 1 mips = 1 million instructions per second
- Common units of time:
  - 1 ms = 1 millisecond (1/1000)
  - 1 µs = 1 microsecond (1/1,000,000)
  - 1 ns = 1 nanosecond (1/1,000,000,000)

Historical Development of Computers

- Evolution of computing hardware divided into generations
- Each generation defined by technology used to build machines

Generation 0: Mechanical Calculating Machines

- Seventeenth century: Pascal’s Pascaline and Leibniz’s Stepped Reckoner
  - Pascal’s design used in adding machines as late as 1920; capable of addition with carry and subtraction
  - Leibniz’s design more versatile; could do multiplication & division as well as addition & subtraction

Generation 0

- Nineteenth century: Charles Babbage & Ada Lovelace
  - Babbage developed calculator he called the Difference Engine for mechanizing solution of polynomial functions
  - Babbage designed (but never built) the Analytical Engine; general-purpose machine including a mechanical processor, storage, and I/O devices
  - Ada Lovelace wrote the world’s first computer program, a plan for how the engine would calculate numbers

Generation 1: Vacuum Tube Computers (1945-1953)

- Several contenders for inventor of first electronic computer:
  - Konrad Zuse: Z1 used electromechanical relays instead of Babbage’s mechanical cranks
  - John Atanasoff: ABC was first completely electronic computer, built specifically to solve systems of linear equations
  - John Mauchly and J. Presper Eckert: ENIAC was first general-purpose electronic computer
Vacuum Tubes

- Control flow of electrons in electrical systems like valves control flow of water in plumbing systems
  - Electrons flow from negatively charged cathode to positively charged anode
  - A control grid or grids within the tube can reduce or prevent electron flow
  - Vacuum tube can act as either switch or amplifier

Generation 2: Transistors (1954-1965)

- Vacuum tube technology wasn’t very dependable
- Transistor: solid-state version of vacuum tube
  - Transistors smaller, use less power, and more reliable
  - Revolutionized entire electronics industry
- Computers still large and expensive at this stage, but beginning to be mass produced; many manufacturers emerged at this point


- Integrated circuit, or microchip, allow many transistors on a single silicon chip
- Computers became faster, smaller, cheaper, more powerful
- First time-sharing and multiprogramming systems appeared at this time
- Computing became more affordable to increasingly smaller organizations

Generation 4: VLSI (1980-present)

- With third generation, multiple transistors were packed onto one chip; current generation features multiple levels of integration:
  - SSI: small-scale integration; 10-100 components per chip
  - MSI: medium-scale integration; 100-1,000
  - LSI: large-scale integration; 1,000-10,000
  - VLSI: very large-scale integration; more than 10,000 components per chip

Generation 4

- VLSI allowed Intel to create (in 1971) the world’s first microprocessor
  - 4004 was fully functional, 4-bit system that ran at 108 KHz
  - Intel also introduced RAM chip, accommodating 4K bits of memory on a single chip
- VLSI spawned development of microcomputers, increased computing power of all classes of computers

Moore’s Law

- In 1965, Intel founder Gordon Moore stated:
  “The density of transistors in an integrated circuit will double every year.”
- Current version of Moore’s Law predicts doubling of density of silicon chips every 18 months
- Moore originally thought this postulate would hold for 10 years; advances in chip manufacturing processes have allowed the law to hold for 40 years, and it is expected to last for perhaps another 10
Principle of Equivalence of Hardware & 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
- Modern computers are implementations of algorithms that execute other algorithms

Semantic Gap

- Open space between the physical components of a computer system and the high-level instructions of an application
- Semantic gap is bridged at each level of abstraction

Abstraction

- Complete definition of abstraction includes the following:
  - Suppression of detail
  - Outline structure
  - Division of responsibility
  - Subdivision of system into smaller subsystems

Abstraction & computer systems

- Can look at a computer as being a machine composed of a hierarchy of levels
  - Each level has specific function
  - Each level exists as a distinct hypothetical machine (or virtual machine)
- Each level’s virtual machine executes its own particular set of instructions, calling upon machines at lower levels to carry out tasks as necessary

Abstraction & computer systems

- Text uses the following labels to describe levels of abstraction in a computer system:
  - App7
  - HOL6
  - Asm86
  - OS4
  - ISA3
  - Mc2
  - LG1
- Each level has its own language to describe tasks performed by computer

Level App7

- The application level is composed of those programs designed to do specific kinds of tasks for end users
- An application may have some sort of programming language associated with it (macros or shortcuts, e.g.)
- Ideally, end users need not be concerned with the actions and language(s) associated with lower levels in the abstraction hierarchy
Level HOL6

- The high order language layer is the layer of abstraction at which most programmers operate
- Applications are typically written in high order languages
- High order languages are characterized by:
  - Portability across platforms
  - Relative ease of use
  - Relatively high level of abstraction, requiring translation of program code prior to execution

Level Asmb5

- The assembly language level is an intermediate step between high order language and the machine language of a particular processor
- Programs at the HOL6 level are usually compiled to level Asmb5, then translated (assembled) to machine language
- Source code can also be written in assembly language

Level OS4

- The operating system is responsible tasks related to multiprogramming, including:
  - Memory protection
  - Process synchronization
  - Device management
- Operating systems were originally developed for multuser systems, but even most single user systems utilize an operating system
- Compilers and assemblers are also considered systems software

Level ISA3

- The instruction set architecture, or machine language level, consists of the set of instructions recognized by the particular hardware platform
- Instructions at this level are directly executable without any translation

Level Mc2

- The microinstruction or control level is the level at which the computer decodes and executes instructions and moves data in and out of the processor
- The processor’s control unit interprets machine language instructions, causing required actions to take place

Level LG1

- The digital logic level consists of the physical components of the computer system, the actual electronic gates and wires
- Boolean algebra and truth tables can be used to describe the operations at this level