Generations in Computer Organization

- **Zeroth** generation – mechanical computers (1642 – 1945)
- **First** generation – vacuum tubes (1945 – 1955)
- **Second** generation – transistors (1955 – 1965)
- **Third** generation – integrated circuits (1965 – 1980)
- **Fourth** generation – VLSI (Very Large Scale Integration ) (1980 – ?)

Milestones in Computer Organization

- **1642**, Blaise Pascal, mechanical machine with gears to add and subtract
- c. **1670**, Baron Gottfried Wilhelm von Leibniz, mechanical machine with gears to add, subtract, multiply and divide
- **1834**, Charles Babbage, Analytical Engine (Ada Lovelace wrote its “assembler”)
- **1936**, Konrad Zuse, Z1, calculator made of electromagnetic relays
- 1940’s, John Atanasoff, George Stibbitz, Howard Aiken (Mark I and II); each worked independently on calculating machines with properties such as binary arithmetic and capacitors for memory
Milestones in Computer Organization

- **1943**, Alan Turing and British govt., COLOSSUS, first electronic computer
- **1946**, John Mauchley and J. Presper Eckert, ENIAC, vacuum tubes
- **1949**, Maurice Wilkes, EDSAC, first stored-program computer
- **1952**, John von Neumann, most current computers use his basic design

Milestones in Computer Organization

- **1950's**, researchers at M.I.T (Massachusetts Institute of Technology), TX-0, first computer using transistors
- **1961**, IBM, 1401, very popular small business machine

Milestones in Computer Organization

- **1963**, Burroughs, B500, first machine designed for high-level language (Algol)
- **1964**, Seymour Cray of CDC (Control Data Corp.), 6600, nearly 10 times as fast as IBM's 7094 because CPU highly parallelized
- **1965**, PDP-8, first mass-market mini-computer. Used a single bus.

Milestones in Computer Organization

- **1964**, IBM, System/360, a family of machines (low end to high end), all compatible. First computer with multiprogramming (several programs in memory at once). Used integrated circuits (dozens of transistors on one chip)
- **1970**, PDP-11, DEC, used integrated circuits.
- **1974**, Intel, 8080, first general-purpose computer on a chip
Milestones in Computer Organization

- 1974, Cray-1, first vector computer (single instructions on vectors of numbers)
- 1978, DEC, VAX, first 32-bit mini-computer
- Steve Jobs and Steve Wozniak, Apple, personal computer (PC)
- 1981, IBM, IBM PC, became most popular PC. Used MS-DOS by Microsoft as OS (operating system) and chip by Intel

- 1985, MIPS (company), MIPS, First commercial RISC machine
- 1987, Sun Microsystems, SPARC, RISC machine
- 1990, IBM, RS6000, first superscalar machine (CPU very parallelized)

Milestones in Computer Organization

- Moore’s Law: number of transistors on a chip doubles every 18 months.

Virtuous cycle:

- Competition
- New markets, companies
- Technological Advances
- Better products
- Lower prices
- New applications

Memory Hierarchy Importance

1980: no cache in µproc; 1995: 2-level cache on chip
(1989 first Intel µproc with a cache on chip)
**Definition**

The terms *processor* and *computational engine* refer broadly to any mechanism that drives computation.

**Processor Organization**

**Von Neumann Architecture**

- Characteristic of most modern processors
- Reference to mathematician John Von Neumann who was one of the computer architecture pioneers
- Central idea is *stored program*

**Three Basic Components of Von Neumann Architecture**

- Processor
- Memory
- I/O facilities

All interact to form a complete computer
**Illustration of Von Neumann Architecture**

- **Computer**
  - Processor
  - Memory
  - Input/output facilities

**Processor**

- Digital device
- Performs computation involving multiple steps
- Building blocks used to form computer system

**Hierarchical Structure and Computational Engines**

- Most computer architecture follows a hierarchical approach
- Subparts of a large, central processor are sophisticated enough to meet our definition of processor
- Some engineers use term *computational engine* for sub-piece that is less powerful than main processor

**Illustration of Processor Hierarchy**

- **CPU**
  - Trigonometry engine
  - Graphics engine
  - Other components
  - Query engine
  - Arithmetic engine
Major Components of a Conventional Processor

- Controller
- Computational engine (ALU)
- Local data storage
- Internal interconnection(s)
- External interface

Illustration of a Conventional Processor

Parts of a Conventional Processor [1]

- Controller
  - Overall responsibility for execution
  - Moves through sequence of steps
  - Coordinates other units
- Computational engine
  - Operates as directed by controller
  - Typically provides arithmetic and Boolean operations (ALU)
  - Performs one operation at a time

Parts of a Conventional Processor [2]

- Local data storage
  - Holds data values for operations
  - Must be loaded before operation can be performed
  - Typically implemented with registers
- Internal interconnections
  - Allow transfer of values among units of the processor
  - Sometimes called data path
Parts of a Conventional Processor [3]

- External interface
  - Handles communication between processor and rest of computer system
  - Provides connections to external memory as well as external I/O devices

Example Register Organizations

(a) 8088/86
(b) 80386, Pentium II
(c) MC68000
Arithmetic Logic Unit (ALU)

- Main computational engine in conventional processor
- Complex unit that can perform variety of tasks
- Typical ALU operations
  - Integer arithmetic (add, subtract, multiply, divide)
  - Shift (left, right, circular)
  - Boolean (and, or, not, exclusive or)

Processor Categories and Roles

- Many possible roles for individual processors in
  - Coprocessors
  - Microcontrollers
  - Microsequencers
  - Embedded system processors
  - General-purpose processors

Coprocessor

- Operates in conjunction with and under the control of another processor
- Usually
  - Special-purpose processor
  - Performs a single task
  - Operates at high speed
- Example: floating point accelerator

Microcontroller

- Programmable device
- Dedicated to control of a physical system
- Example: run automobile engine or grocery store door
Example Steps a Microcontroller Performs (Automatic Door)

do forever {
    wait for the sensor to be tripped;
    turn on power to the door motor;
    wait for a signal that indicates the
doors is open;
    wait for the sensor to reset;
    delay ten seconds;
    turn off power to the door motor;
}

Microsequencer

- Similar to microcontroller
- Controls coprocessors and other engines within a large processor
- Example: move operands to floating point unit; invoke an operation; move result back to memory

Embedded System Processor

- Runs sophisticated electronic device
- Usually more powerful than microcontroller
- Example: control DVD player, including commands from a remote control

General-Purpose Processor

- Most powerful type of processor
- Completely programmable
- Full functionality
- Example: CPU in a personal computer
Processor Performance

Clock and Instruction Rate

- **Clock cycle**
  - Time interval in which all basic circuits (steps) inside a processor must complete
  - Time at which gates are clocked

- **Clock rate**
  - \( 1 / \text{clock cycle} \) (GHz – billion cycles per second)

- **Instruction rate**
  - Measure of time required to *execute* instructions (MIPS – million instructions per second)
  - Varies because some instructions take more time (require more cycles) than others
  - Can typically estimate average cycles per instruction for normal job mix

Basic Performance Equation

Define:

- \( N = \) actual number of instructions executed in program
- \( S = \) average number of cycles for instructions in program
- \( R = \) clock rate
- \( T = \) program execution time

\[
T = \frac{N \times S}{R}
\]

Improve Performance

- To improve performance, must decrease \( N \) or \( S \) or increase \( R \)
- Parameters are not independent (e.g., increasing \( R \) may increase \( S \))
- \( N \) primarily controlled by compiler
- Processors with largest \( R \) may not have the best performance (due to different \( S \))
- Making logic circuits faster is a clear win, increases \( R \) while \( S \) and \( N \) remain unchanged
Benchmarks

- **SPEC rating**
  - Based upon a suite of applications
    \[ SPEC_j = \frac{\text{run time on reference computer}}{\text{run time on test computer}} \]
  - \[ SPEC \text{ rating} = \left( \prod_{i=1}^{n} SPEC_i \right)^{1/n} \]
  - http://www.spec.org

- **LINPACK** (Scientific Computing)
  - Speed in solving linear system of equation
  - http://www.top500.org/lists/2005/11/basic