Computer Organization
&
Assembly Language Programming

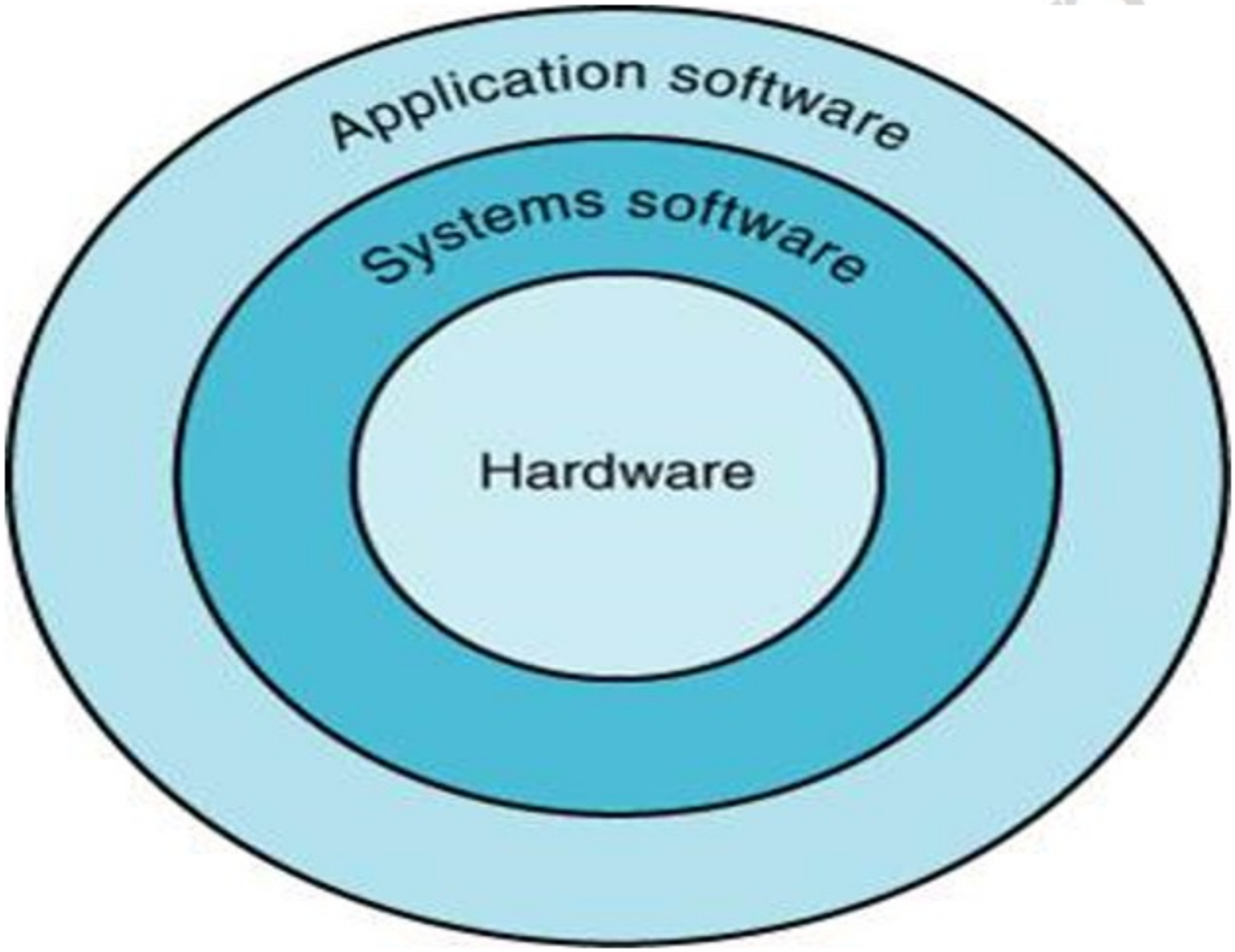
Description of basic computer

- Computers are the digital devices that performs the various computational task.
- Digital means there is the process of representing the information by the help of the certain discrete values.
- Information is represented in digital computers in terms of bits.
- By various coding techniques these groups of bits can not only represent numbers but also other discrete symbols.

Description of basic computer

- A computer system is sub-divided into two functional entities: hardware and software.
- Hardware consists of all electronics components.
- Software consists of instruction and data that the computer manipulate to perform various tasks.

Description of basic computer



Description of basic computer

- Application software is all the computer software that causes a computer to perform useful tasks beyond the running of the computer itself.
- System software is computer software designed to operate and control the computer hardware and to provide a platform for running application software.
- Computer hardware is the collection of physical elements that comprise a computer system

Computer Organization

- It refers to the operational units and their interconnections that realize the architectural specifications.
- It is concerned with the way the hardware components operate and the way they are connected together to form the computer system.
- Examples are things that are transparent to the programmer:
  - control signals.
  - interfaces between computer and peripherals.
  - the memory technology being used.

Von Neumann Architecture

- Between 1945 & 1951 John von Neumann set down the structure, layout, interaction, cooperation, realisation, implementation, functionality and activity for the whole computer as a system. The Von Neumann Architecture is characterized by: -
- **A memory, arithmetical-logical unit (ALU), control unit, input and output devices**,
- All parts of a computer are connected together by Bus,
- Memory and Devices are controlled by CPU.
- Data can pass through bus in half duplex mode to and from CPU.

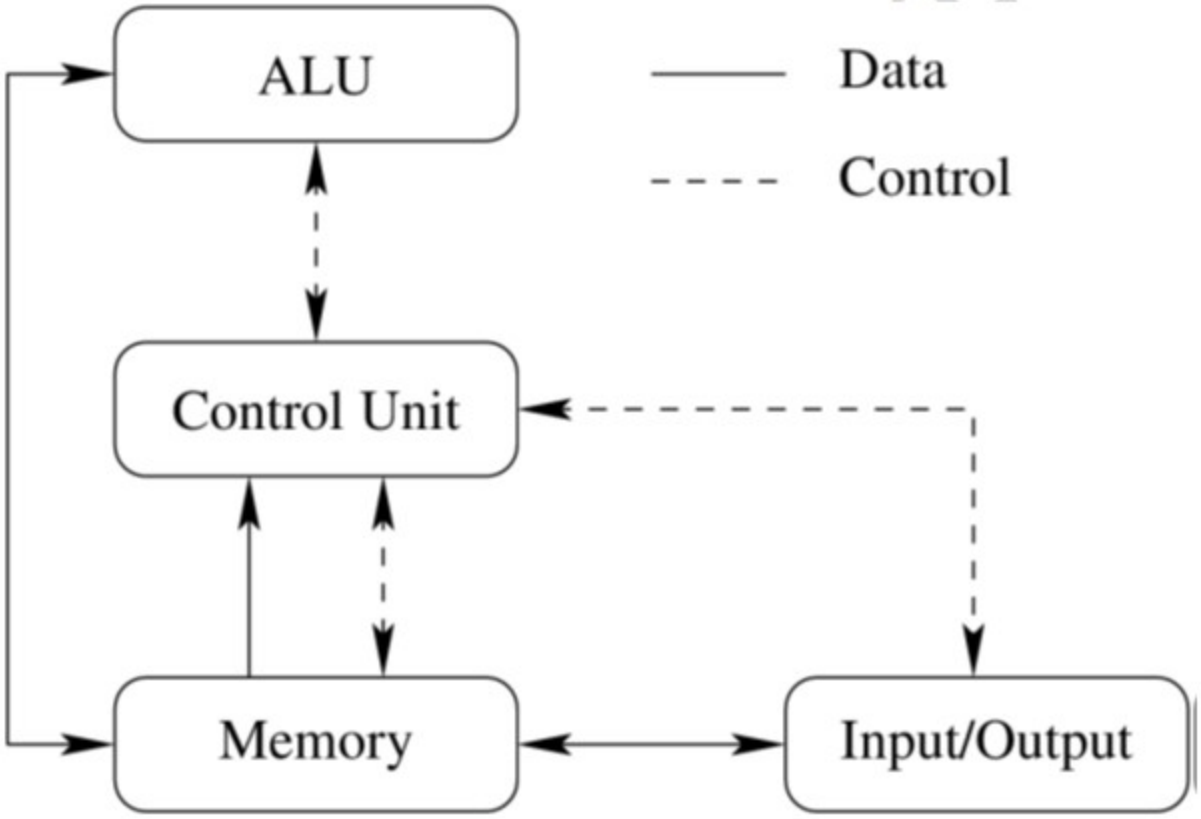
Von Neumann Architecture

- Memory holds both programs and data; this is also known as the **stored program concept**.
- Memory is addressed **linearly**; that is, there is a single sequential numeric address for each and every memory location.
- Memory is addressed by the location number without regard to the data contained within.
- Memory is split to small cells with the same size. Their ordinal numbers are called address numbers.

Von Neumann Architecture

- Program consists of a sequence of instructions. Instructions are executed in order they are stored in memory.
- Sequence of instructions can be changed only by unconditional or conditional jump instructions.
- Instructions, characters, data and numbers are represented in binary form.

Diagrammatic view of Von Neumann Architecture



Advantages of Von Neumann

- Control Unit gets data and instruction in the same way from one memory. It simplifies design and development of the Control Unit.
- Data from memory and from devices are accessed in the same way.
- Memory organization is in the hands of programmers.

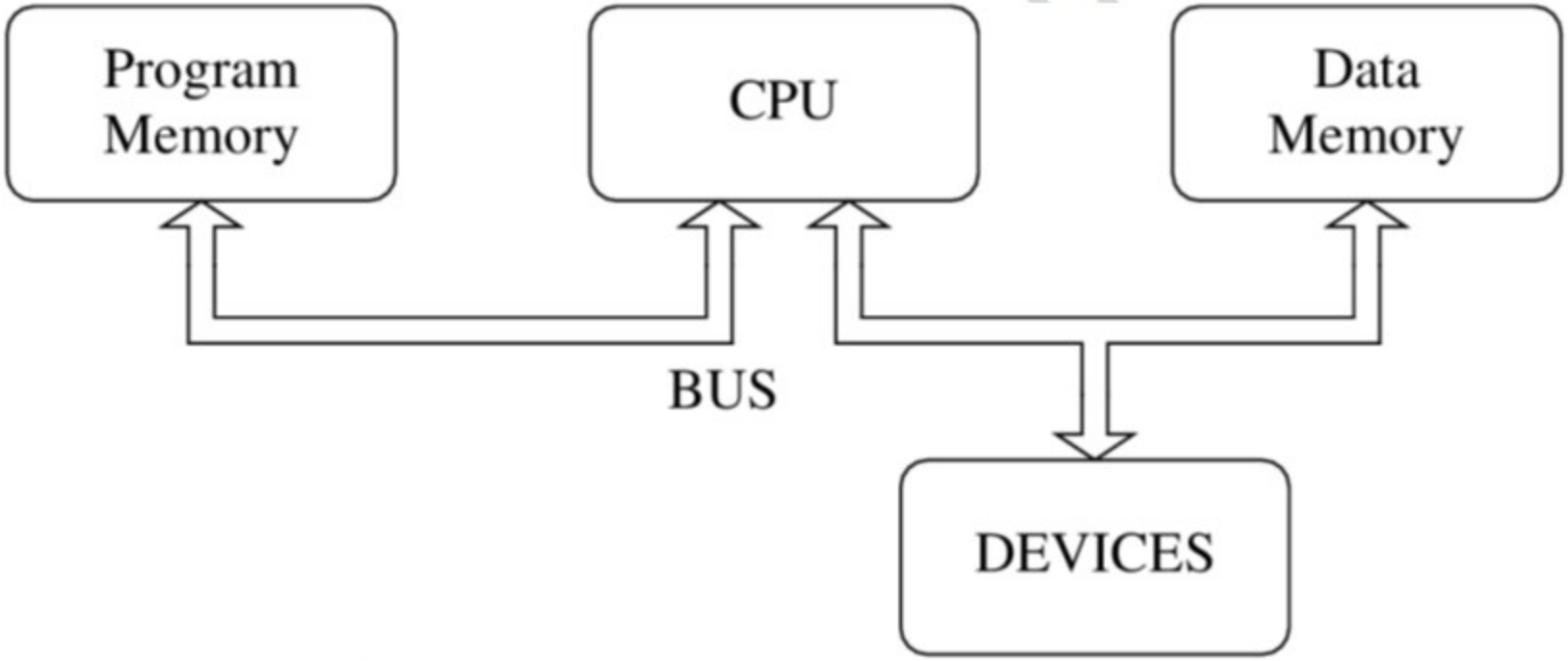
Disadvantages of Von Neumann

- Serial instruction processing does not allow parallel execution of program. Parallel executions are simulated later by the Operating system.
- One bus is a bottleneck. Only one information can be accessed at the same time.
- Instruction stored in the same memory as the data can be accidentally rewritten by an error in a program.

Harvard Architecture

- MARK II computer was finished at Harvard University in 1947. It wasn't so modern as the computer from von Neumann team. But it introduced a slightly different architecture. Memory for data was separated from the memory for instruction. This concept is known as the **Harvard architecture**.

Diagrammatic view of Harvard Architecture



Advantages of Harvard

- since it has two memories , this allows parallel access to data and instructions.
- Development of the Control Unit is expensive and needs more time
- Data and instructions are accessed the same way.
- Both memories can use different cell sizes.

Disadvantages of Harvard

- Free data memory cant be used for instruction and vice-versa.
- Production of a computer with two buses is more expensive and needs more time.

Harvard vs von Neumann

Harvard

- Two memories with two Buses allow parallel access to data access and instructions.
- Control unit for two buses is more complicated and more expensive.
- Program cant write itself.
- Both memories can use different sizes.

Von Neumann

- Content of the memory if organised and all installed memory can be used.
- One bus is simpler for the control unit design
- Computer with one bus is cheaper.
- Error in a program can rewrite instruction and crash program execution.

Harvard vs von Neumann

Harvard

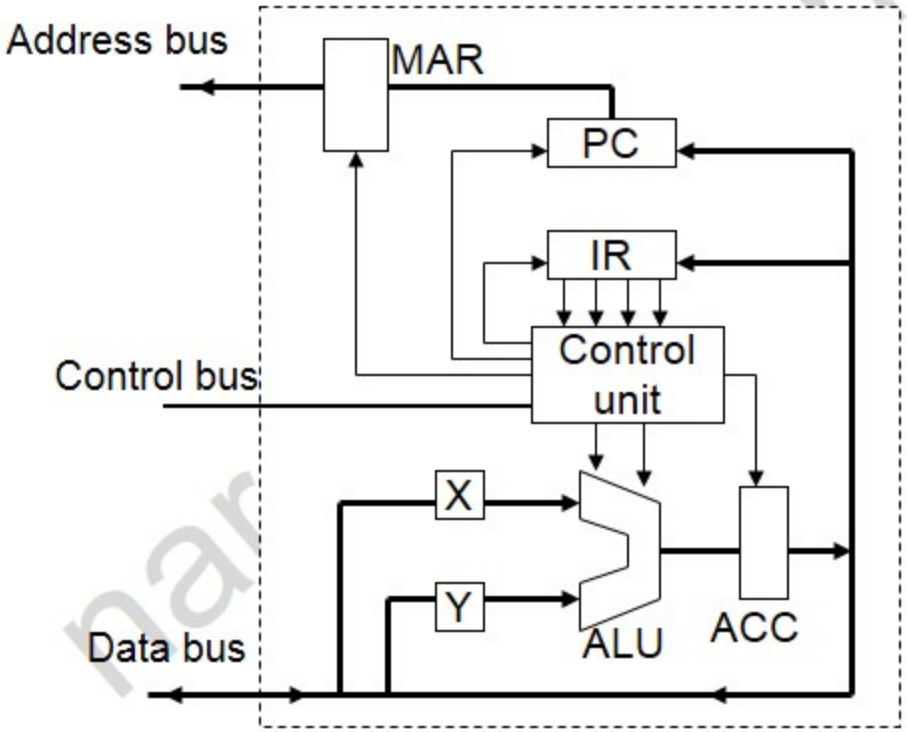
- Development of a complicated Control Unit needs more time.
- Free data memory can't be used for instruction and vice-versa.

Von Neumann

- Development of the Control Unit is cheaper and faster.
- Data and instruction is accessed in the same way.
- One Bus (for Data, instruction and devices) is a bottleneck.

What are microprocessors?

- A microprocessor is a processor (or Central Processing Unit, CPU) fabricated on a single integrated circuit.



A simple microprocessor architecture

Evolution of Computers

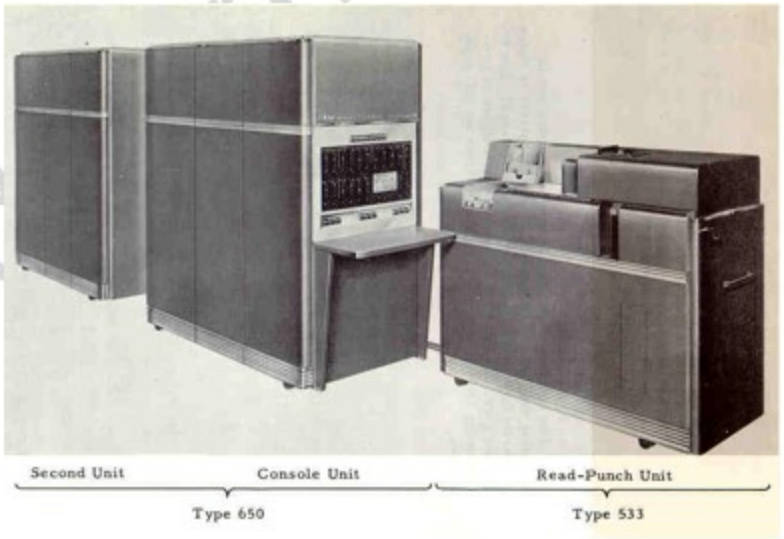
- First generation (1939-1954) - vacuum tube
- Second generation (1954-1959) - transistor
- Third generation (1959-1971) - IC
- Fourth generation (1971-present) - microprocessor

Evolution of Computers

- First generation (1939-1954) - vacuum tube

Computer-Science Center
University of Virginia

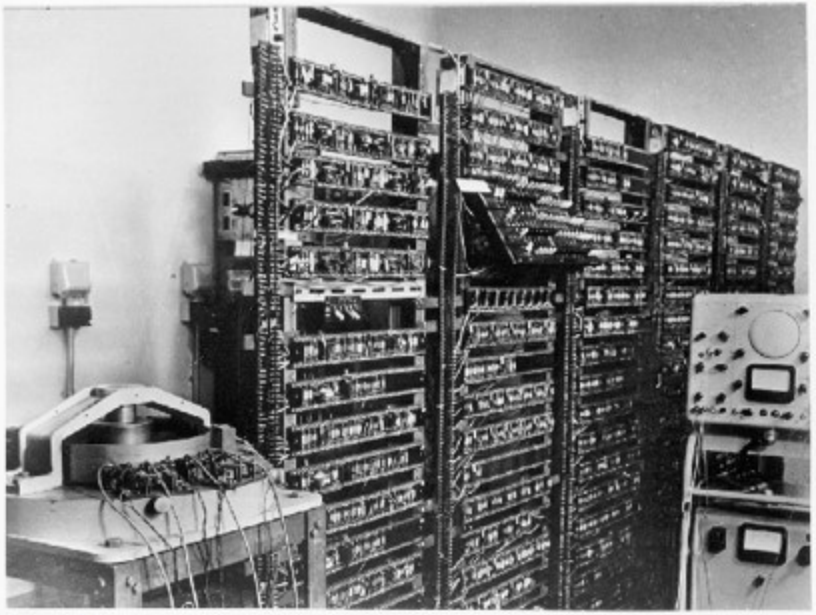
Burroughs 205 Computer
1960-1964



IBM 650, 1954

Evolution of Computers

- Second generation (1954-1959) - transistor



Manchester University Experimental Transistor Computer

Evolution of Computers

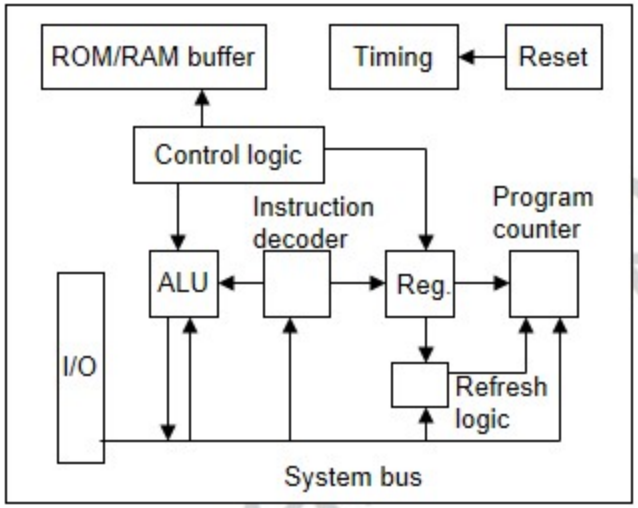
- Third generation (1959-1971) - IC

PDP-8, Digital Equipment Corporation

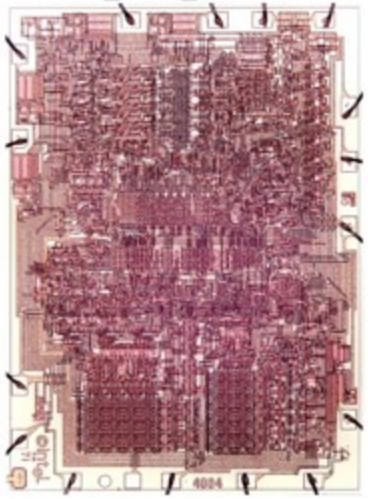
— Thanks to the use of ICs, the DEC PDP-8 is the least expensive general purpose small computer in 1960s

Evolution of Computers

- Fourth generation (1971-present) - microprocessor
  - In 1971, Intel developed 4-bit 4004 chip for calculator applications.



Block diagram of Intel 4004



4004 chip layout

http://www.intel.com

A good review article: The History of The Microprocessor, Bell Labs Technical Journal, Autumn, 1997

Types of Computer

Super computers

- Are the most powerful available. These computers are high capacity computers that run continuously and are being used by very big organizations mostly big corporations and government institutions. Users of super computers include NASA and US government, some big schools of companies.


NASA

Mainframe computers

- Are less powerful than super computers but are capable of great processing speed, multi tasking capability and high data storage. They are used by most banks to process information of depositors and millions of daily bank transactions. Insurance companies use them for their policy holders database. These computers have specialized wiring system and usually occupies a big room with temperature control.

Mid-range computers

- Are used for medium sized companies for specific purposes. They may be used for certain assembly line operations or manufacturing stages in big companies. The size of mini computers may be as a washing machine. It may be a stand alone system for specialized applications including network servers. They are also called minicomputers.

Microcomputers or desktop Computers

- Are the most common and widely used computer today. There are two types of microcomputers, they are the desktop computers and the notebook computer.
- Desktop computers are the common computer that you see in homes, schools, and in most businesses. They are small enough to be placed on a desk or table but are too big to be carried around.

- Notebook Computer – also known as laptop computers are portable, lightweight and are easy to carry around.

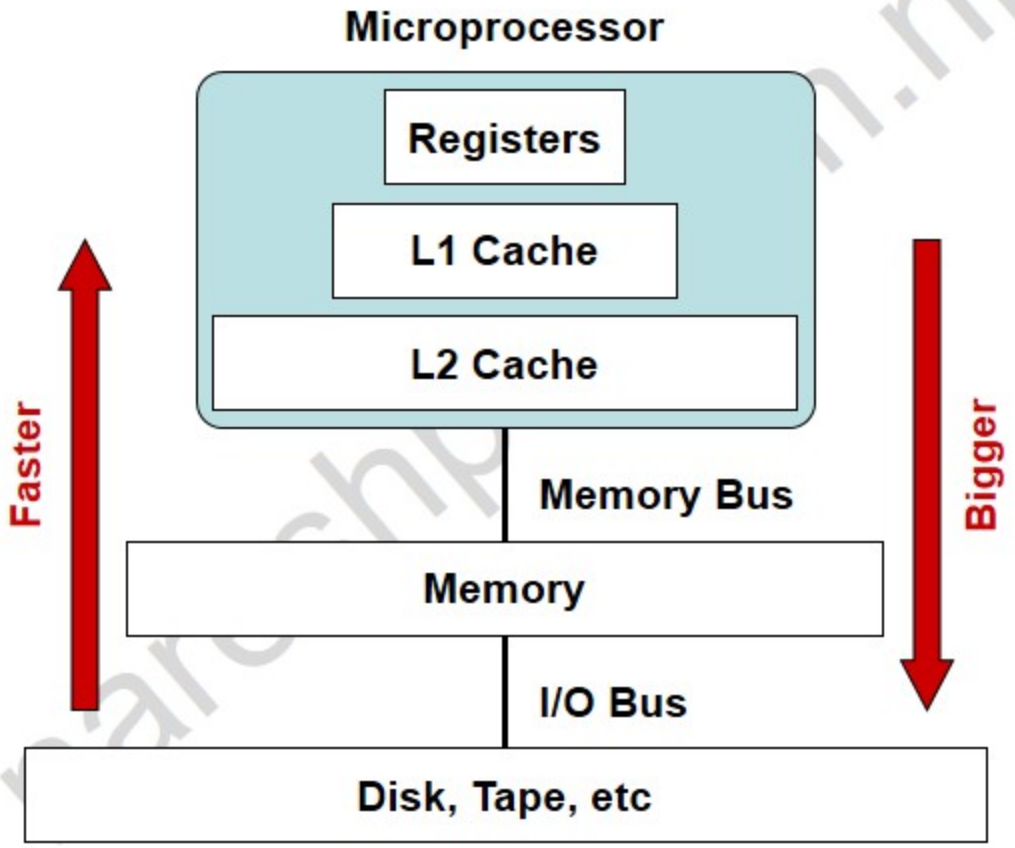
tablet personal computer

- Is a fancy notebook that has a swivel design and can accept handwriting using a digital pen. Today, laptop or notebook computers have evolved so that it comes in various sizes, shape and features.

Handheld computer

- Are the smallest computers that are designed to fit into one hand or palm that is why they are also called palm-top computers. These computers may combine pen input, personal organizer tools and communication capabilities such as telephone and internet applications.
- Personal digital assistants or PDAs are the most common palm-top or hand held computers available today. It is so because PDAs have all the features of a cell phone, organizer and some basic computers application into one. And in some cases, it even includes camera, audio and video capabilities.

Typical Memory Hierarchy



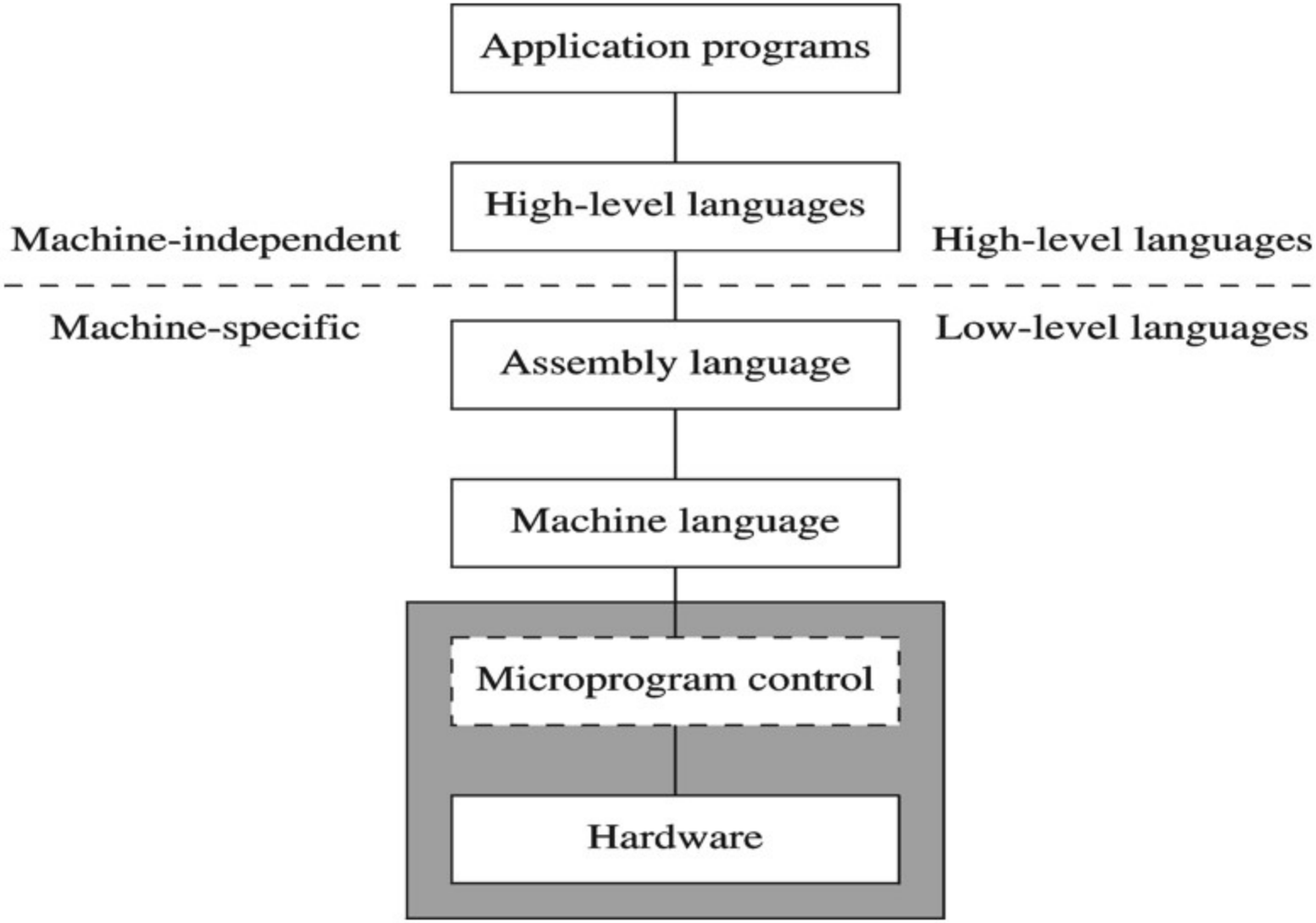
Memory

- Ordered sequence of bytes
  - The sequence number is called the memory address
- Byte addressable memory
  - Each byte has a unique address
  - Supported by almost all processors
- Physical address space
  - Determined by the address bus width
  - Pentium has a 32-bit address bus
    - Physical address space = **4GB = 2^{32} bytes**
  - Itanium with a 64-bit address bus can support
    - Up to **2^{64} bytes** of physical address space

The Need for a Memory Hierarchy

- Widening (expand) speed gap between CPU and main memory
  - Processor operation takes less than 1 ns
  - Main memory requires more than 50 ns to access
- Each instruction involves at least one memory access
  - One memory access to fetch the instruction
  - Additional memory accesses for instructions involving memory data access
- Memory bandwidth limits the instruction execution rate
- Cache memory can help bridge the CPU-memory gap
- Cache memory is small in size but fast

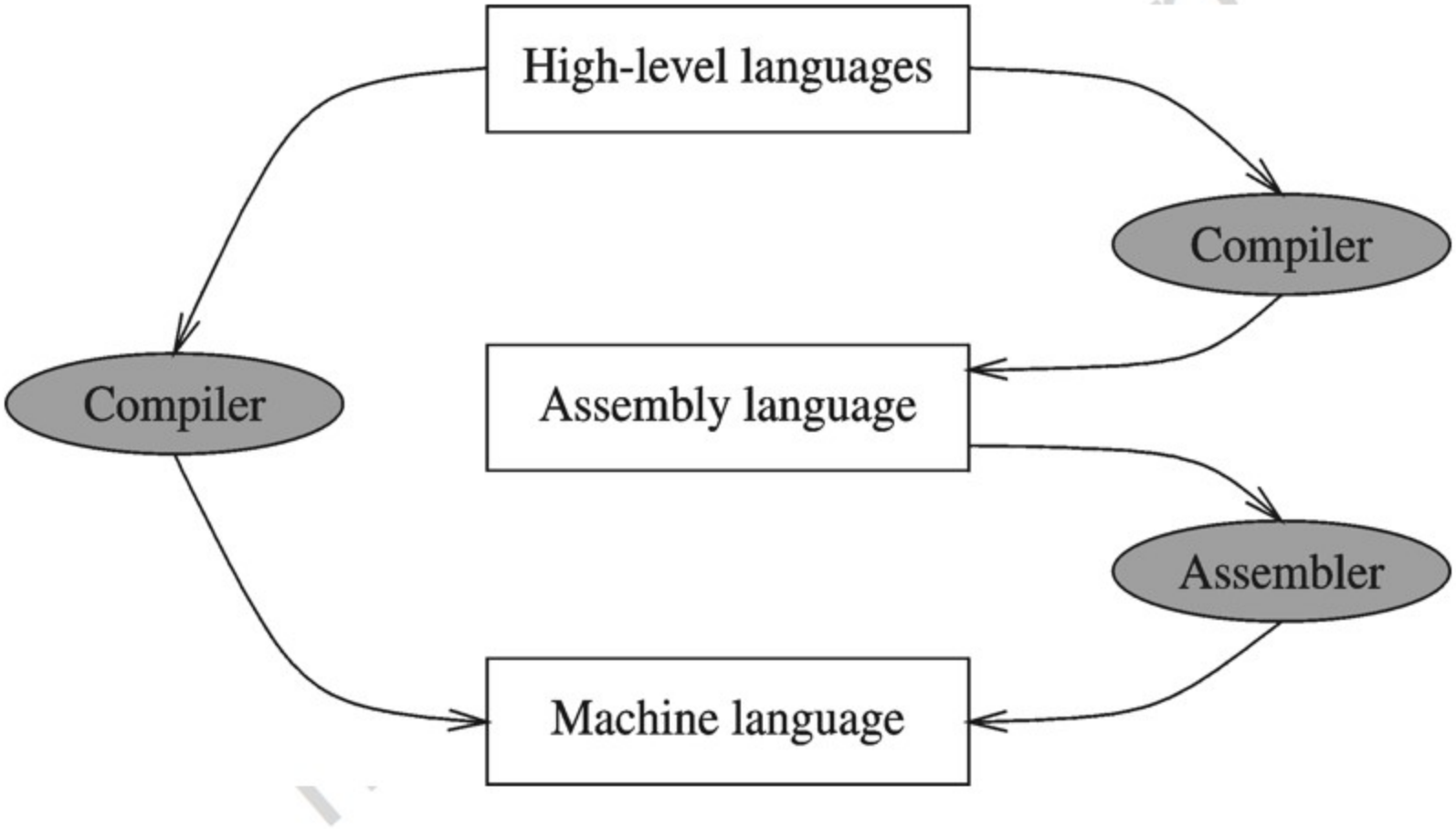
A Hierarchy of Languages



Assembly and Machine Language

- Machine language
  - Native to a processor(CPU): executed directly by hardware
  - Instructions consist of binary code: 1s and 0s
- Assembly language
  - A programming language that uses **symbolic names** to represent operations, registers and **memory locations**.
  - Slightly higher-level language
  - Readability of instructions is better than machine language
  - One-to-one correspondence with machine language instructions
- Assemblers translate (assembly) to machine code
- Compilers translate high-level programs (c++, jave) to machine code
  - Either directly, or
  - Indirectly via an assembler

Compiler and Assembler



Instructions and Machine Language

- Each command of a program is called an instruction (it instructs the computer what to do).
- Computers only deal with binary data, hence the instructions must be in binary format (0s and 1s) .
- The set of all instructions (in binary form) makes up the computer's machine language. This is also referred to as the instruction set.

Instruction Fields

- Machine language instructions usually are made up of several fields. Each field specifies different information for the computer. The major two fields are:
- Opcode field which stands for operation code and it specifies the particular operation that is to be performed.
  - Each operation has its unique opcode.
- Operands fields which specify where to get the source and destination operands for the operation specified by the opcode.
  - The source/destination of operands can be a constant, the memory or one of the general-purpose registers.

Assembly vs. Machine Code

Instruction Address	Machine Code	Assembly Instruction
0005	B8 0001	MOV AX, 1
0008	B8 0002	MOV AX, 2
000B	B8 0003	MOV AX, 3
000E	B8 0004	MOV AX, 4
0011	BB 0001	MOV BX, 1
0014	B9 0001	MOV CX, 1
0017	BA 0001	MOV DX, 1
001A	8B C3	MOV AX, BX
001C	8B C1	MOV AX, CX
001E	8B C2	MOV AX, DX
0020	83 C0 01	ADD AX, 1
0023	83 C0 02	ADD AX, 2
0026	03 C3	ADD AX, BX
0028	03 C1	ADD AX, CX
002A	03 06 0000	ADD AX, i
002E	83 E8 01	SUB AX, 1
0031	2B C3	SUB AX, BX
0033	05 1234	ADD AX, 1234h

Translating Languages

English: D is assigned the sum of A times B plus 10.

High-Level Language: D = A * B + 10

A statement in a high-level language is translated typically into several machine-level instructions

Intel Assembly Language:

```
mov   eax, A
mul   B
add   eax, 10
mov   D, eax
```

Intel Machine Language:

```
A1 00404000
F7 25 00404004
83 C0 0A
A3 00404008
```

Mapping Between Assembly Language and HLL

- Translating HLL programs to machine language programs is not a one-to-one mapping
- A HLL instruction (usually called a **statement**) will be translated to one or more machine language instructions

Mapping between some C instructions and 8086 assembly language

Instruction Class	C	Assembly Language
Data Movement	a = 5	MOV a, 5

Arithmetic/Logic	b = a + 5	MOV ax, a
		ADD ax, 5
		MOV b, ax

Control Flow	goto LBL	JMP LBL

Advantages of High-Level Languages

- Program development is faster
  - High-level statements: fewer instructions to code
- Program maintenance is easier
  - For the same above reasons
- Programs are portable
  - Contain few machine-dependent details
    - Can be used with little or no modifications on different machines
  - Compiler translates to the target machine language
  - However, Assembly language programs are not portable

Why Learn Assembly Language?

- Accessibility to system hardware
  - Assembly Language is useful for implementing system software
  - Also useful for small embedded system applications that require to access hardware directly.
- Writing assembly programs gives the computer designer the needed deep understanding of the instruction set and how to design one
- To be able to write compilers for HLLs, we need to be expert with the machine language. Assembly programming provides this experience

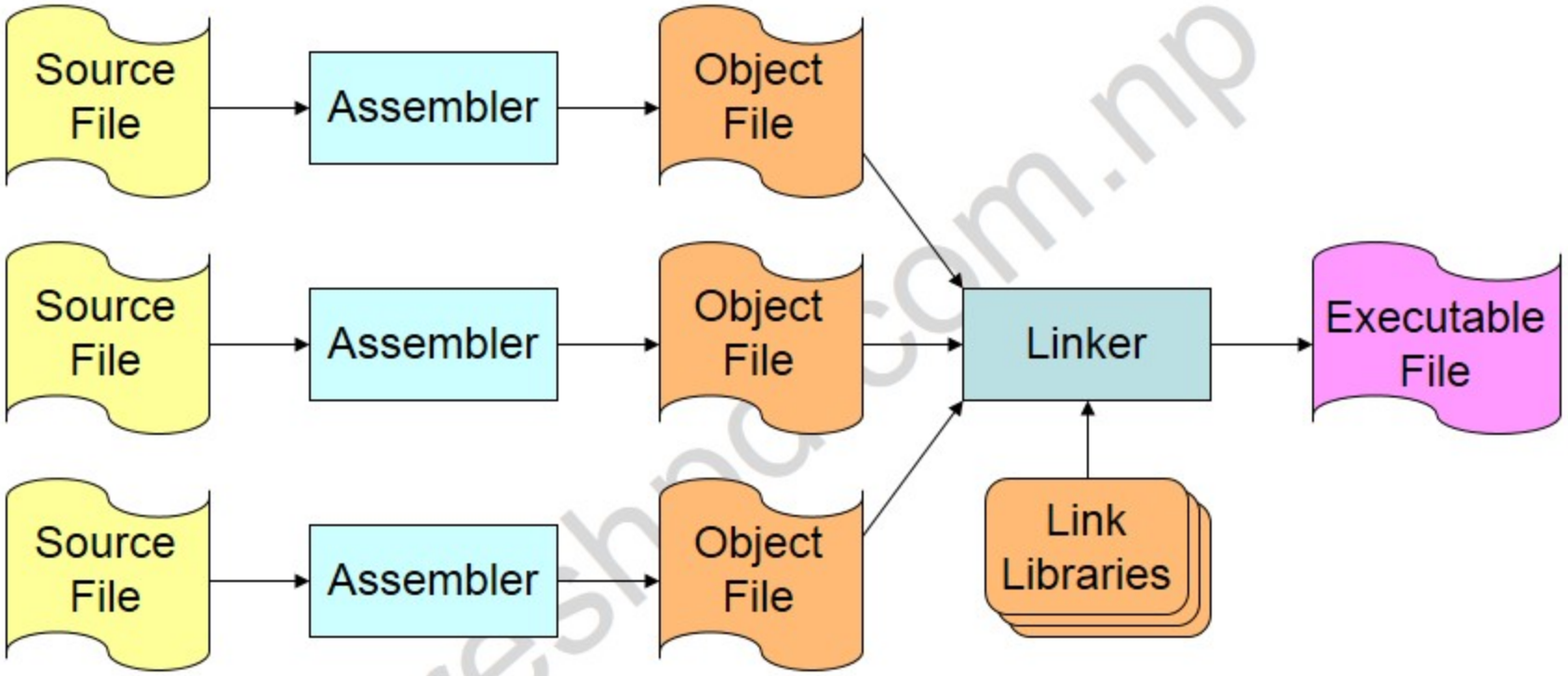
Assembler

- Software tools are needed for editing, assembling, linking, and debugging assembly language programs
- An assembler is a program that converts *source-code* programs written in assembly language into object files in machine language

Linker and Link Libraries

- You need a linker program to produce executable files
- linker program combines your program's object file created by the assembler with other object files and link libraries, and produces a single executable program

Assemble and Link Process



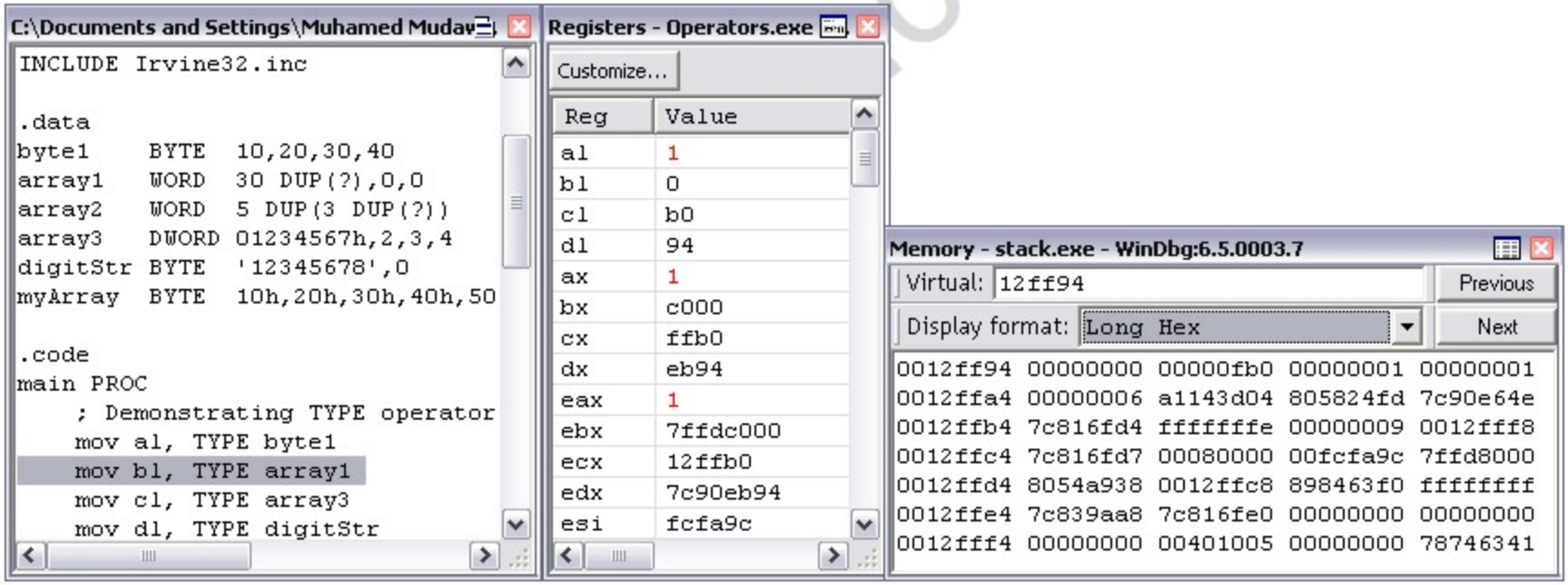
A project may consist of multiple source files

Assembler translates each source file separately into an object file

Linker links all object files together with link libraries

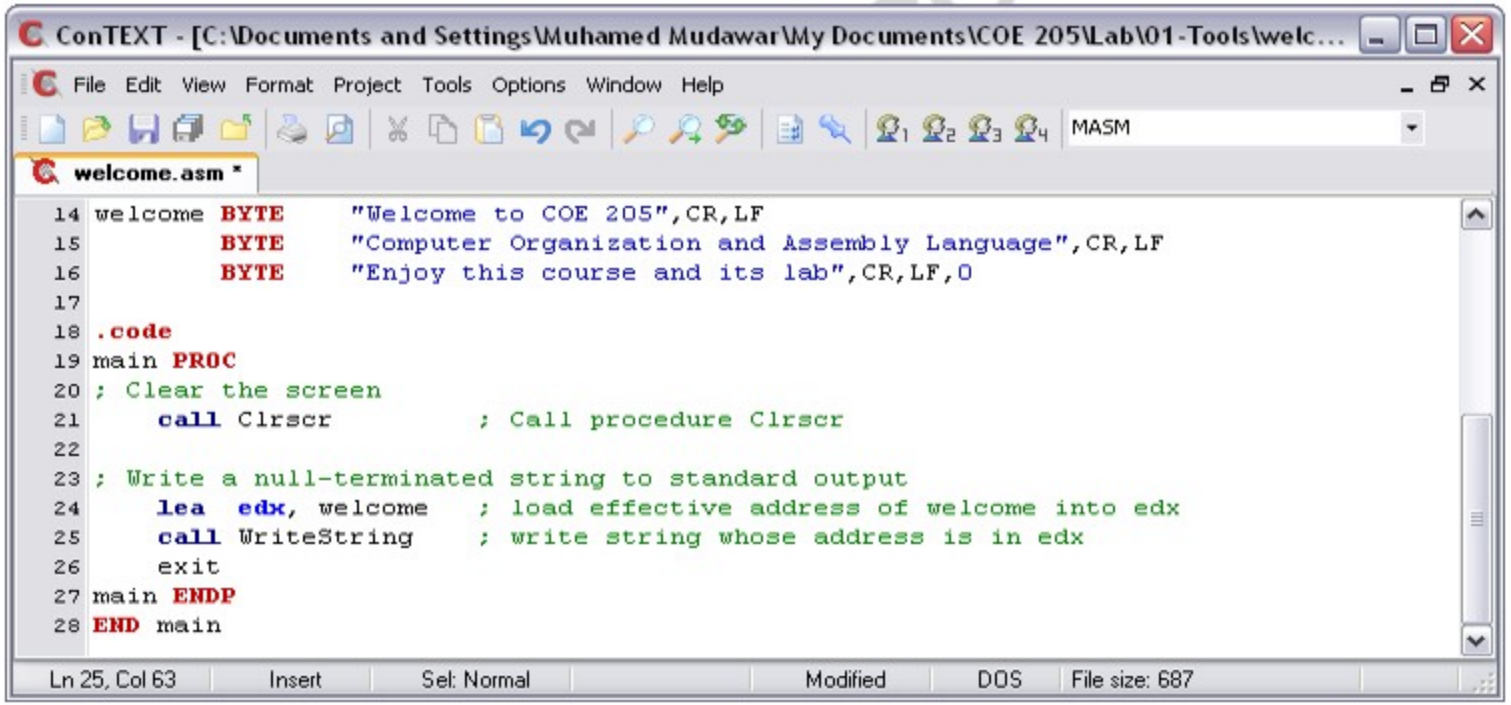
Debugger

- Debugger allows you to trace the execution of a program
- Also, allows you to view code, memory, registers, etc.
- Example: 32-bit Windows debugger

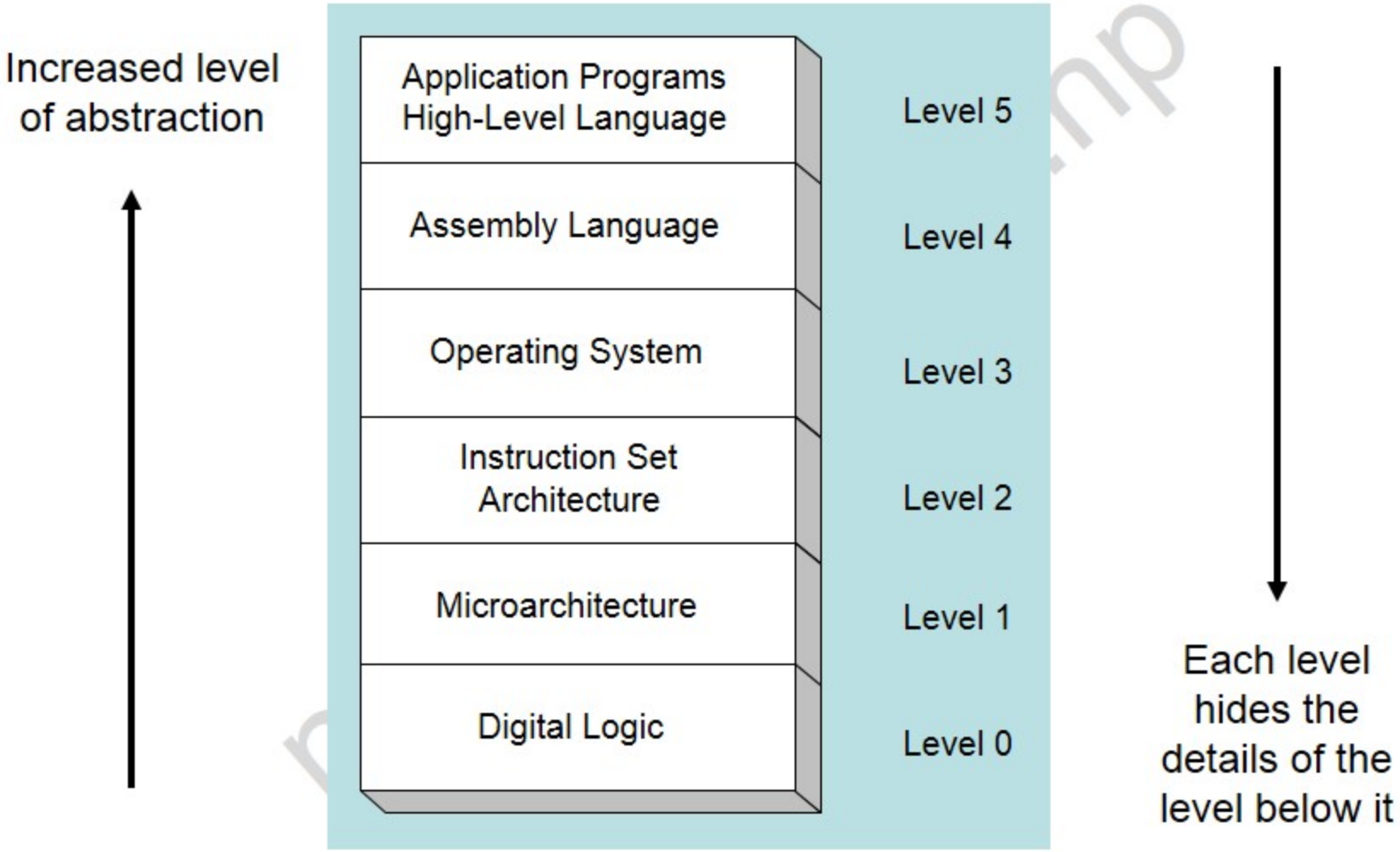


Editor

- Editor: Allows you to create assembly language source files
- Some editors provide syntax highlighting features and can be customized as a programming environment



Programmer's View of a Computer System



Programmer's View - 2

- Application Programs (Level 5)
  - Written in high-level programming languages
  - Such as Java, C++, Pascal, Visual Basic . . .
  - Programs compile into assembly language level (Level 4)
- Assembly Language (Level 4)
  - Instruction mnemonics are used
  - Have one-to-one correspondence to machine language
  - Calls functions written at the operating system level (Level 3)
  - Programs are translated into machine language (Level 2)
- Operating System (Level 3)
  - Provides services to level 4 and 5 programs
  - Translated to run at the machine instruction level (Level 2)

Programmer's View - 3

- Instruction Set Architecture (ISA) (Level 2)
  - Specifies how a processor (CPU) functions
  - Machine language is executed by Level 1 (microarchitecture)
- Microarchitecture (CPU) (Level 1)
  - Controls the execution of machine instructions (Level 2)
  - Implemented by digital logic (Level 0)
- Digital Logic (Level 0)
  - Implements the microarchitecture (CPU)
  - Uses digital logic gates
  - Logic gates are implemented using transistors

Instruction Set Architecture (ISA)

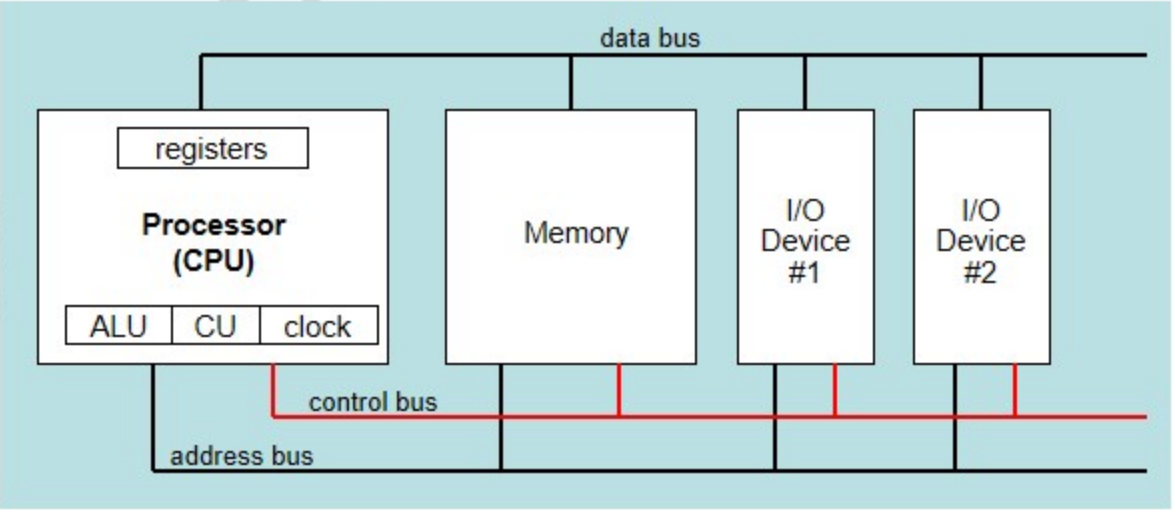
- Collection of assembly/machine instruction set of the machine
- Machine resources that can be managed with these instructions:
  - Memory
  - Programmer-accessible registers.

Next ...

- Welcome
- Assembly-, Machine-, and High-Level Languages
- Assembly Language Programming Tools
- Programmer's View of a Computer System
- Basic Computer Organization

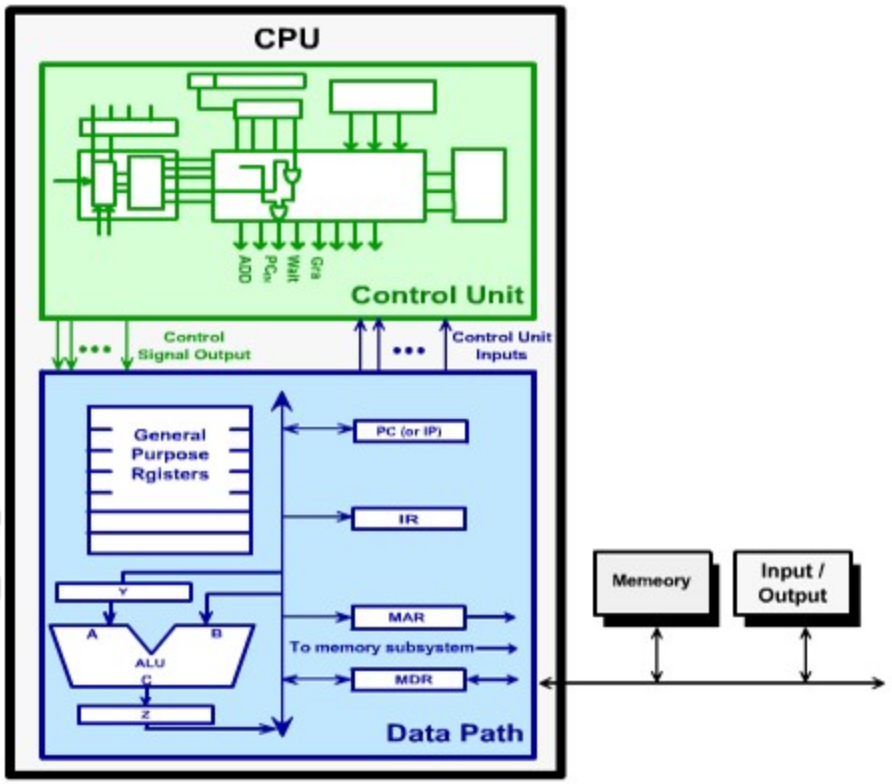
Basic Computer Organization

- Since the 1940's, computers (HW)have 3 classic components:
  - Processor, called also the CPU (Central Processing Unit)
  - Memory and Storage Devices
  - I/O Devices
- Interconnected with one or more buses
- Bus consists of
  - Data Bus
  - Address Bus
  - Control Bus

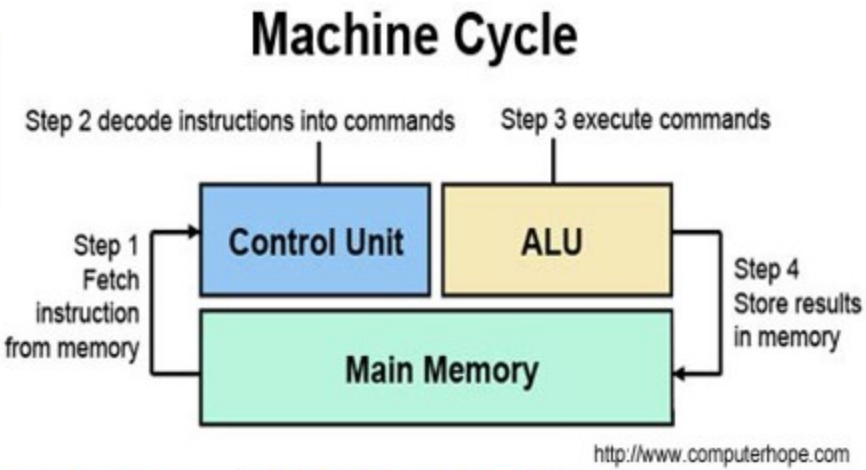


Processor (CPU)

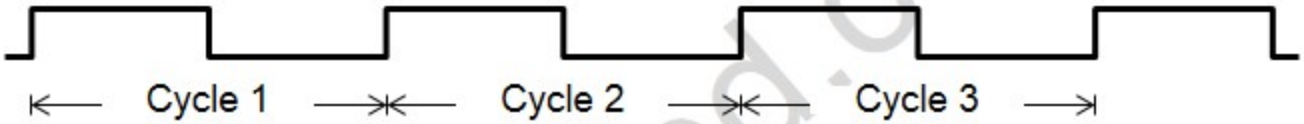
- Processor consists of
  - Datapath
    - ALU
    - Registers
  - Control unit
- ALU
  - Performs arithmetic and logic instructions
- Control unit (CU)
  - Generates the control signals required to execute instructions
- Implementation varies from one processor to another



Clock cycle



- What is machine cycle?
- Clock cycle is duration of a cycle= 1 / Clock rate

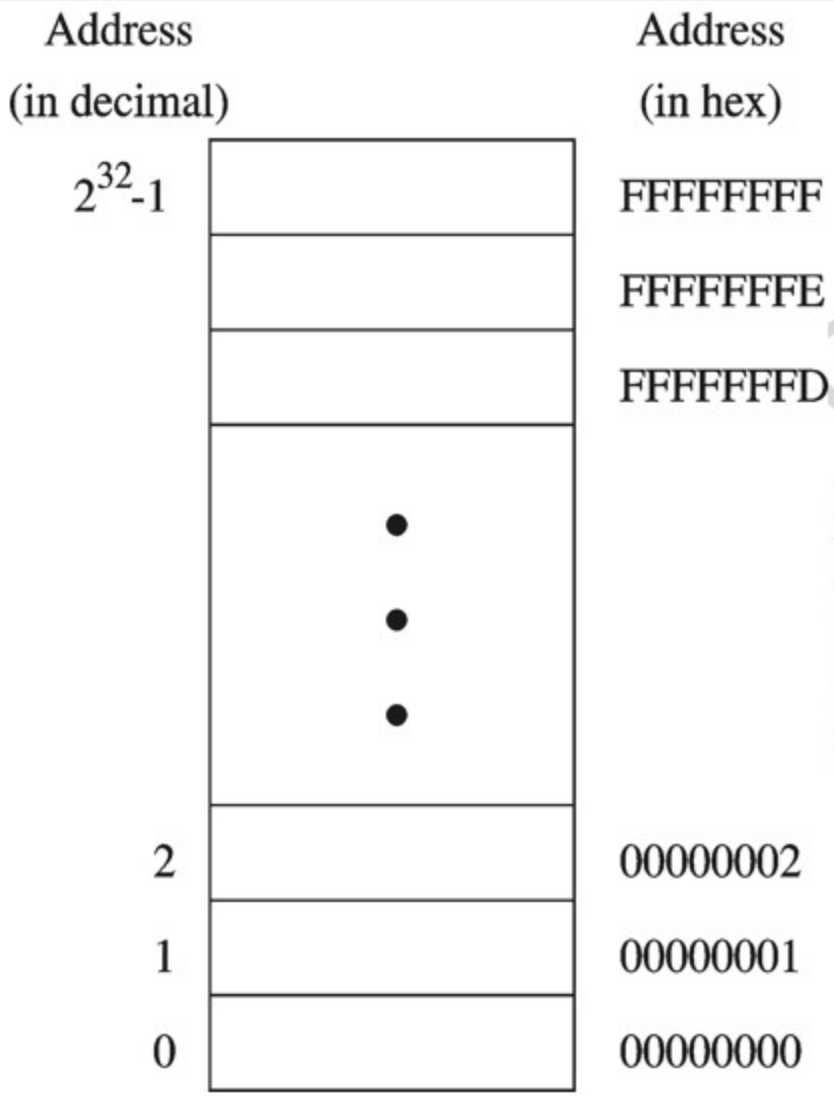


- Clock rate = Cycles per second
  - 1 Hz = 1 cycle/sec
  - 1 KHz = 10^3 cycles/sec
  - 1 MHz = 10^6 cycles/sec
  - 1 GHz = 10^9 cycles/sec
- Clock cycles measure the execution of instructions.
- Clock rate measure the CPU speed

Memory

- Ordered sequence of bytes
  - The sequence number is called the memory address
- Byte addressable memory
  - Each byte has a unique address
  - Supported by almost all processors
- Physical address space
  - Determined by the *address bus width*
  - Pentium has a 32-bit address bus
    - Physical address space = **4GB = 2^{32} bytes**
  - Itanium with a 64-bit address bus can support
    - Up to **2^{64} bytes** of physical address space

Address Space



Address Space is the set of memory locations (bytes) that can be addressed

Memory Devices

- Random-Access Memory (RAM)
  - Usually called the main memory
  - It can be read and written to
  - It does not store information permanently (Volatile , when it is powered off, the stored information are gone)
  - Information stored in it can be accessed in any order at equal time periods (hence the name random access)
  - Information is accessed by an address that specifies the exact location of the piece of information in the RAM.
  - DRAM = Dynamic RAM
    - 1-Transistor cell
    - Slow
    - Typical choice for main memory
  - SRAM: Static RAM
    - 6-Transistor cell,
    - faster
    - Typical choice for cache memory

Memory Devices

- ROM (Read-Only-Memory)
  - A read-only-memory, non-volatile i.e. stores information permanently
  - Has random access of stored information
  - Used to store the information required to startup the computer
  - Many types: ROM, EPROM, EEPROM, and FLASH

- Cache
  - A very fast type of RAM that is used to store information that is most frequently or recently used by the computer
  - Recent computers have 2-levels of cache; the first level is faster but smaller in size (usually called internal cache), and the second level is slower but larger in size (external cache).

Typical Memory Hierarchy

- Registers are at the top of the hierarchy
  - Typical size < 1 KB
  - Access time < 0.5 ns
- Level 1 Cache (8 – 64 KB)
  - Access time: 0.5 – 1 ns
- L2 Cache (512KB – 8MB)
  - Access time: 2 – 10 ns
- Main Memory (1 – 2 GB)
  - Access time: 50 – 70 ns
- Disk Storage (> 200 GB)
  - Access time: milliseconds