



CC312: Computer Organization

Dr. Ahmed Abou EL-Farag

Dr. Marwa El-Shenawy

the essentials of

Computer Organization and Architecture

Linda Null and Julia Lobur

Chapter 1

Introduction

Chapter 1 Objectives



- Know the difference between computer organization and computer architecture.
- Understand units of measure common to computer systems.
- Appreciate the evolution of computers.
- Understand the computer as a layered system.
- Be able to explain the von Neumann architecture and the function of basic computer components.

1.1 Overview



Why study computer organization and architecture?

- Design better programs, including system software, and device drivers.
- Optimize program behavior.
- Evaluate (benchmark) computer system performance.
- Understand time, space, and price tradeoffs.

1.1 Overview



- Computer organization
 - Encompasses all physical aspects of computer systems.
 - E.g., circuit design, control signals, memory types.
 - *How does a computer work?*
- Computer architecture
 - Logical aspects of system implementation as seen by the programmer.
 - E.g., instruction sets, instruction formats, data types, addressing modes.
 - *How do I design a computer?*

1.2 Computer Components



- There is no clear distinction between matters related to computer organization and matters relevant to computer architecture.
- 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.****

* Assuming speed is not a concern.

1.2 Computer Components

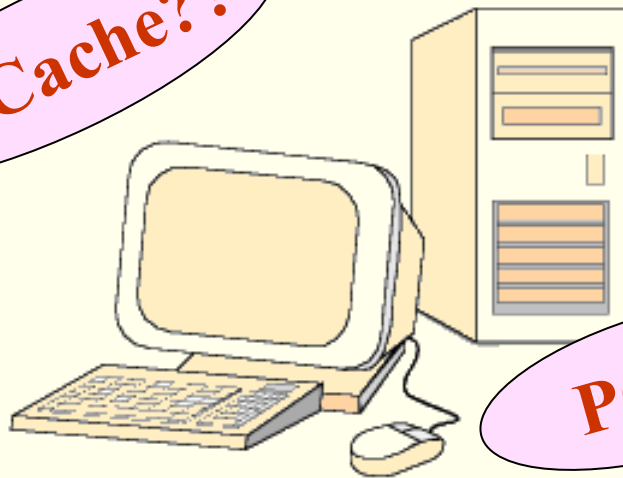


- At the most basic level, a computer is a device consisting of three pieces:
 - A processor to interpret and execute programs
 - A memory to store both data and programs
 - A mechanism for transferring data to and from the outside world.

1.3 An Example System

Consider this advertisement:

For Sale: Obsolete Computer – Cheap! Cheap! Cheap!



- Pentium III 667MHz
- 133MHz 64MB SDRAM
- 32KB L1 cache, 256KB L2
- 30GB EIDE hard drive (7200rpm)
- 48X max variable CD-ROM
- 2 USB ports, 1 serial port, 1 parallel port
- Monitor, 19", .24mm AG, 1280x1024 at 85Hz
- Intel 3D AGP graphics card
- 56K PCI voice modem
- 64-bit PCI sound card

MHz??

L1 Cache??

MB??

PCI??

USB??

What does it all mean??

1.3 An Example System



Measures of capacity and speed:

- Kilo- (K) = 1 thousand = 10^3 and 2^{10}
- Mega- (M) = 1 million = 10^6 and 2^{20}
- Giga- (G) = 1 billion = 10^9 and 2^{30}
- Tera- (T) = 1 trillion = 10^{12} and 2^{40}
- Peta- (P) = 1 quadrillion = 10^{15} and 2^{50}

Whether a metric refers to a power of ten or a power of two typically depends upon what is being measured.

1.3 An Example System



- Hertz = clock cycles per second (frequency)
 - 1MHz = 1,000,000Hz
 - Processor speeds are measured in MHz or GHz.
- Byte = a unit of storage
 - 1KB = 2^{10} = 1024 Bytes
 - 1MB = 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.

1.3 An Example System



Measures of time and space:

- Milli- (m) = 1 thousandth = 10^{-3}
- Micro- (μ) = 1 millionth = 10^{-6}
- Nano- (n) = 1 billionth = 10^{-9}
- Pico- (p) = 1 trillionth = 10^{-12}
- Femto- (f) = 1 quadrillionth = 10^{-15}

1.3 An Example System



- 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.

1.3 An Example System

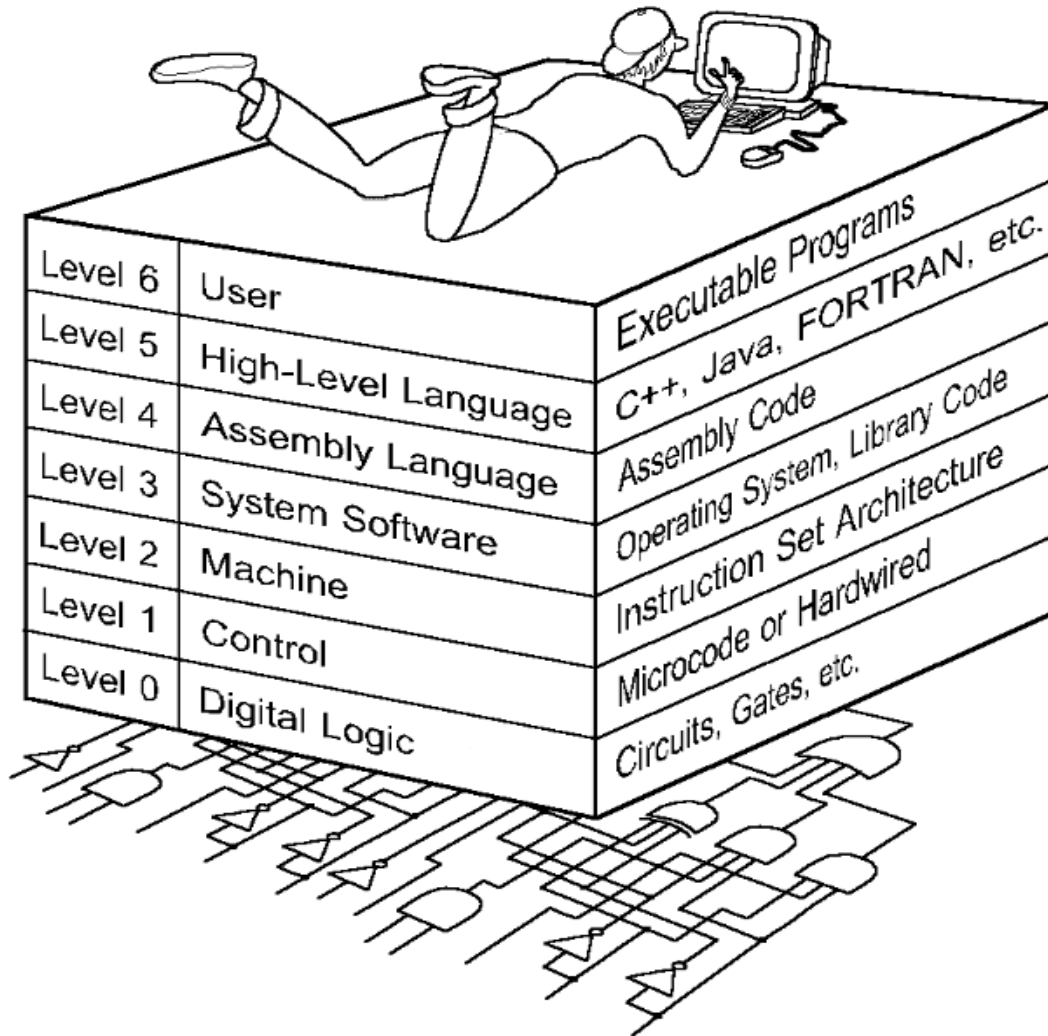


- We note that cycle time is the reciprocal of clock frequency.
- A bus operating at 133MHz has a cycle time of 7.52 nanoseconds:

$$133,000,000 \text{ cycles/second} = 7.52\text{ns/cycle}$$

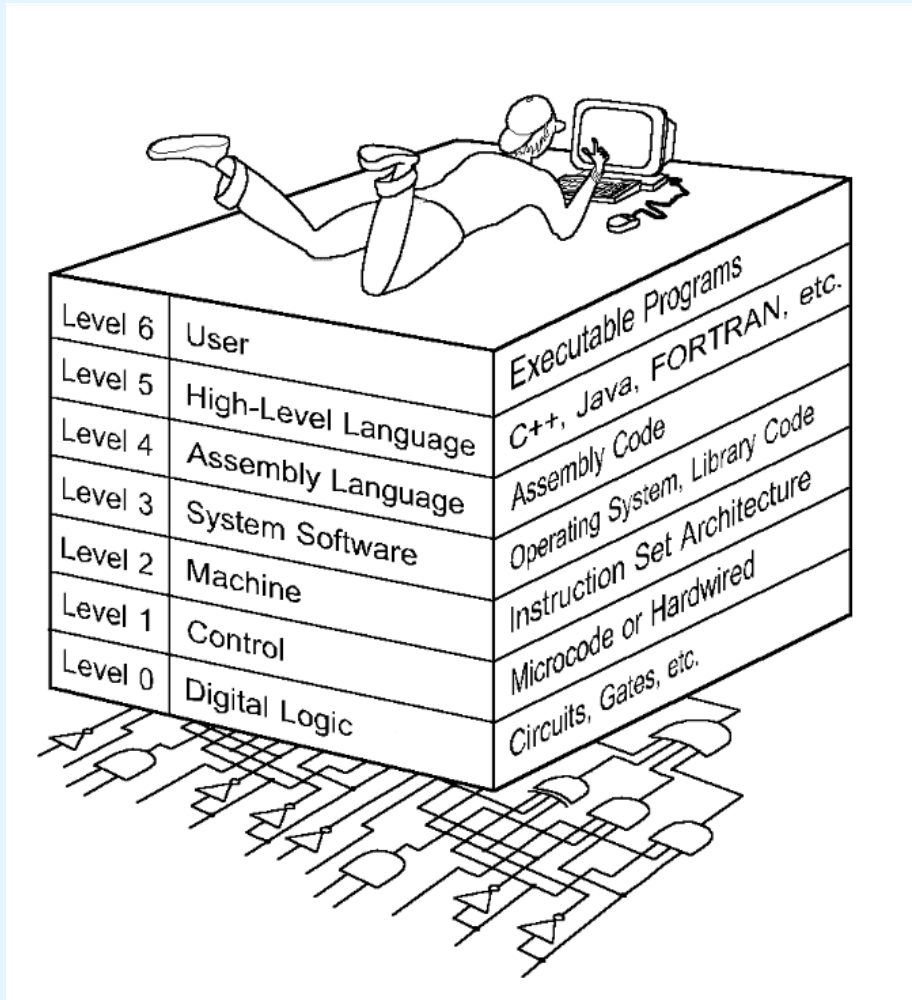
Now back to the advertisement ...

1.6 The Computer Level Hierarchy



1.6 The Computer Level Hierarchy

- 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.



1.6 The Computer Level Hierarchy



- Level 6: The User Level
 - 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.

1.6 The Computer Level Hierarchy



- **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.

1.6 The Computer Level Hierarchy



- 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.

1.6 The Computer Level Hierarchy



- 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.

1.6 The Computer Level Hierarchy



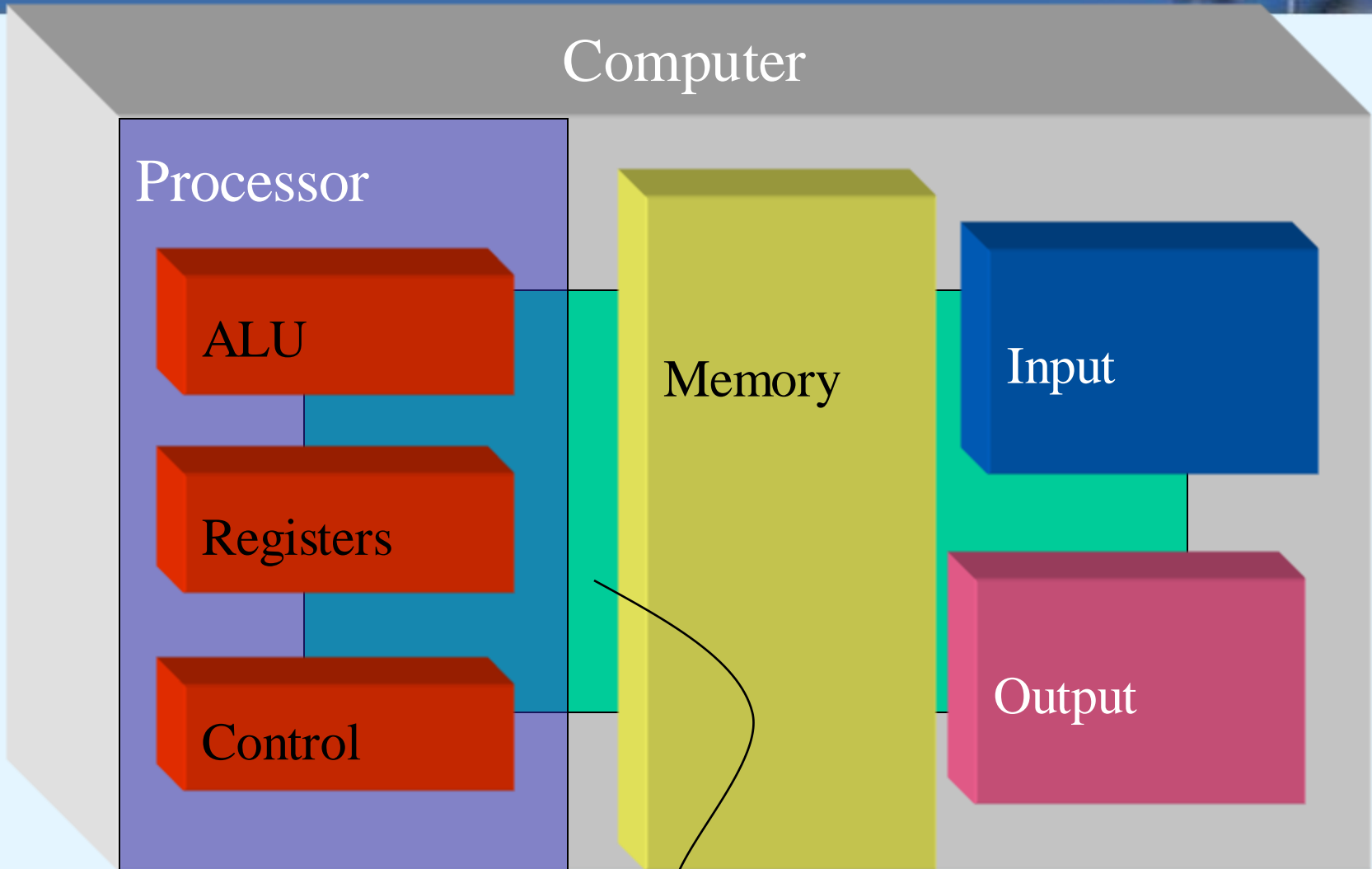
- Level 0: Digital Logic Level
 - This level is where we find digital circuits (the chips).
 - Digital circuits consist of gates and wires.
 - These components implement the mathematical logic of all other levels.

1.7 The von Neumann Model



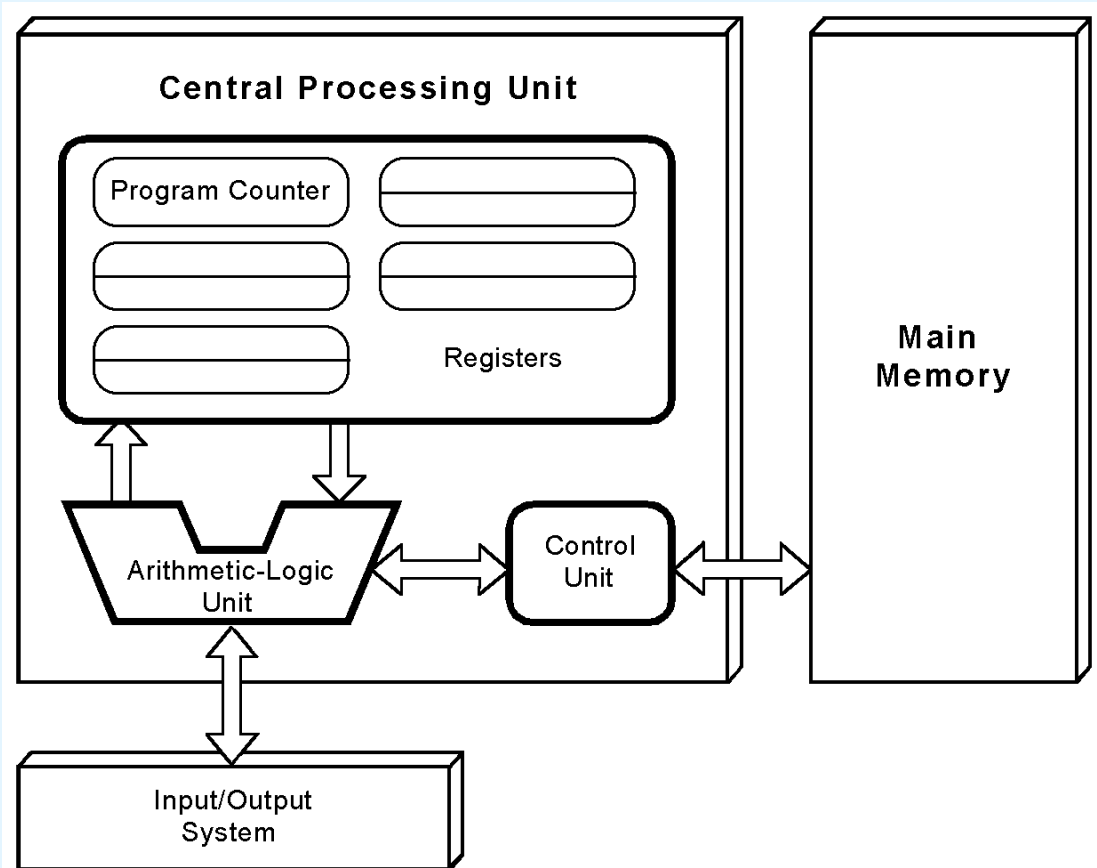
- Today's stored-program computers have the following characteristics:
 - Three hardware systems:
 - A central processing unit (CPU)
 - A main memory system
 - An I/O system
 - The capacity to carry out sequential instruction processing.
 - A single data path between the CPU and main memory.
 - This single path is known as the *von Neumann bottleneck*.

Five Classic Components



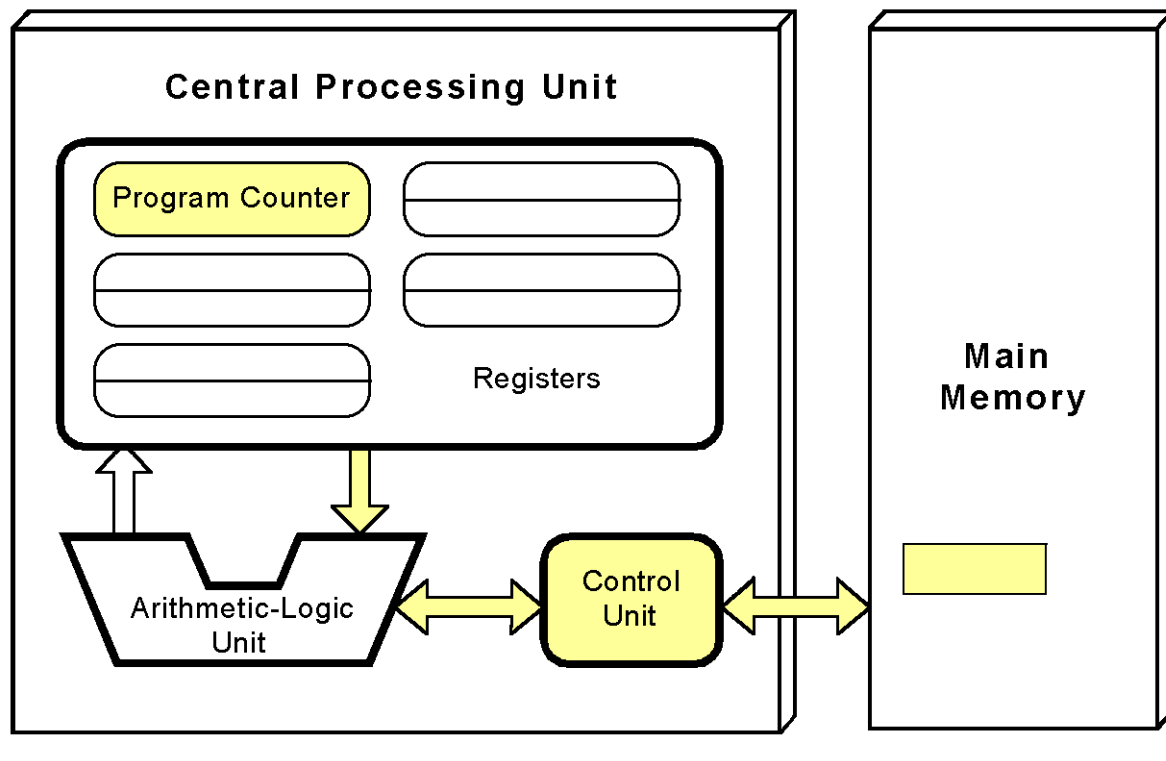
1.7 The von Neumann Model

- This is a general depiction of a von Neumann system:
- These computers employ a fetch-decode-execute cycle to run programs as follows . . .



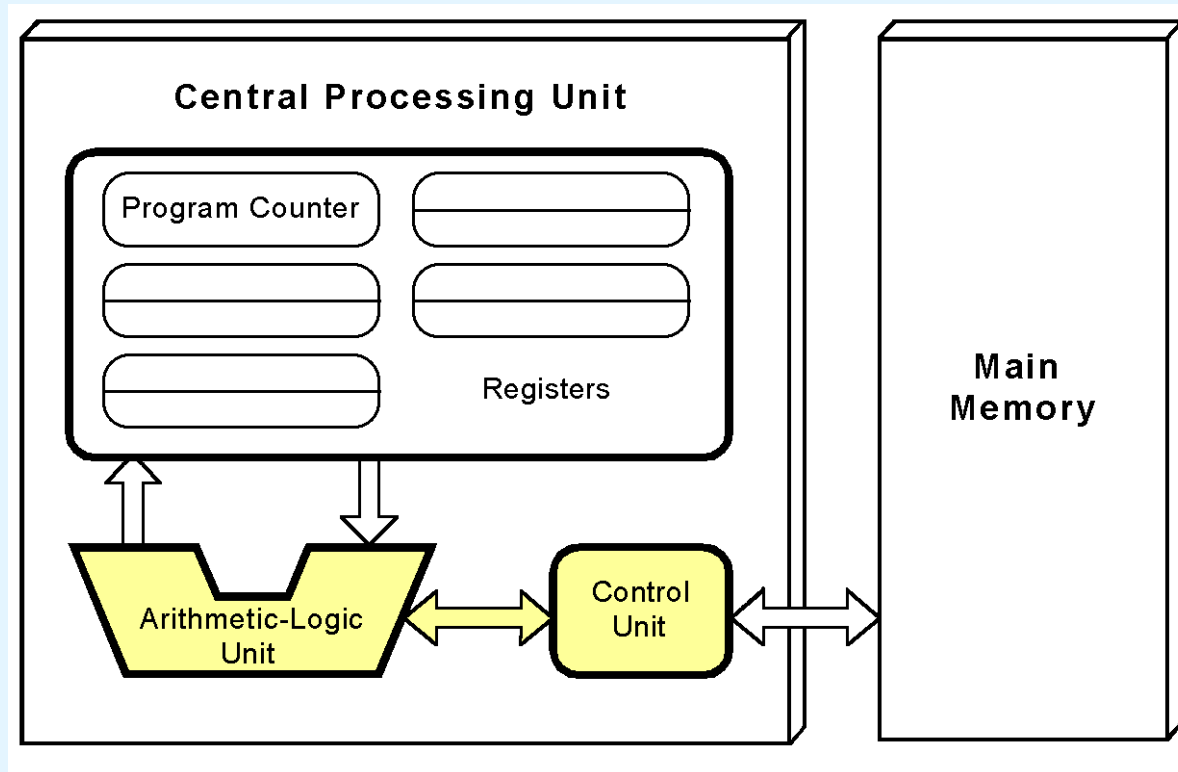
1.7 The von Neumann Model

- The control unit fetches the next instruction from memory using the program counter to determine where the instruction is located.



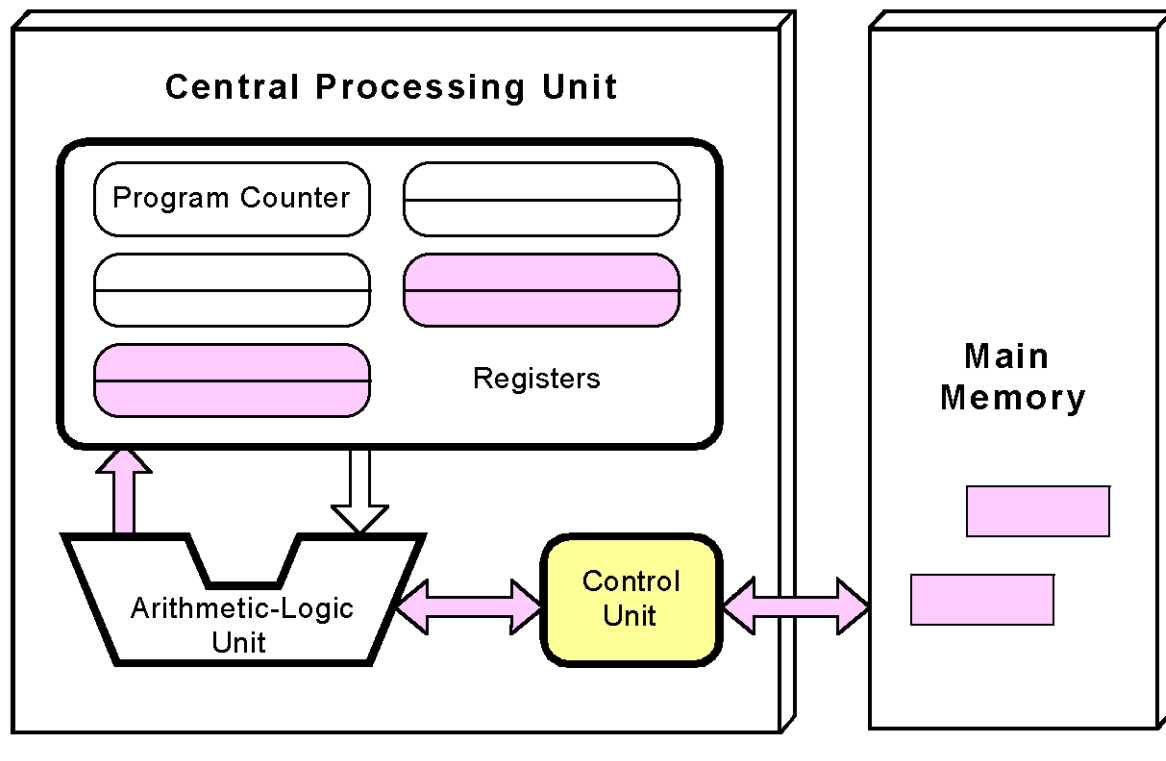
1.7 The von Neumann Model

- The instruction is decoded into a language that the ALU can understand.



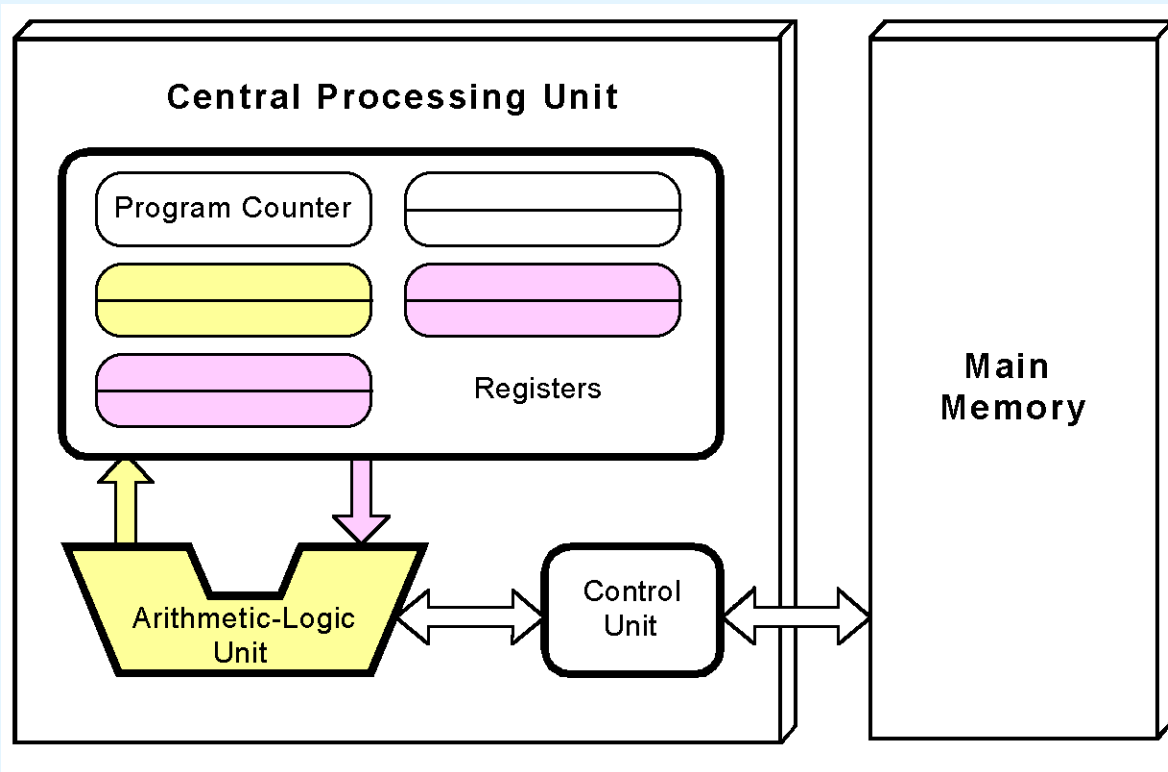
1.7 The von Neumann Model

- Any data operands required to execute the instruction are fetched from memory and placed into registers within the CPU.



1.7 The von Neumann Model

- The ALU executes the instruction and places results in registers or memory.



1.8 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.

1.8 Non-von Neumann Models



- In the late 1960s, high-performance computer systems were equipped with dual processors to increase computational throughput.
- In the 1970s supercomputer systems were introduced with 32 processors.
- Supercomputers with 1,000 processors were built in the 1980s.
- In 1999, IBM announced its Blue Gene system containing over 1 million processors.

1.8 Non-von Neumann Models



- Parallel processing is only one method of providing increased computational power.
- More radical systems have reinvented the fundamental concepts of computation.
- These advanced systems include genetic computers, quantum computers, and dataflow systems.
- At this point, it is unclear whether any of these systems will provide the basis for the next generation of computers.

Conclusion



- This chapter has given you an overview of the subject of computer architecture.
- You should now be sufficiently familiar with general system structure to guide your studies throughout the remainder of this course.
- Subsequent chapters will explore many of these topics in great detail.