Computers and Software

CMPT 145

Copyright Notice

Learning Objectives I

After studying this chapter, a student should be able to:

- Describe the different components of a computer system.
- Explain how the components of a computer system interact.
- Describe the operational behaviour of the machine cycle.
- Explain the difference between machine language and programming language.
- Explain the difference between a compiled language and an interpreted language.

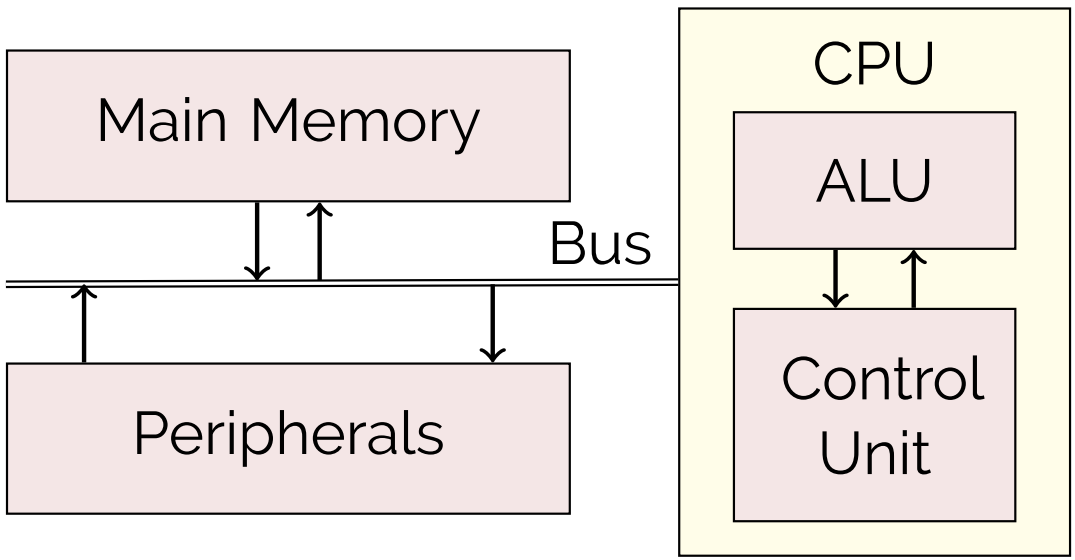
Learning Objectives II

- Explain the difference between an Interactive Development Environment, a compiler, and an interpreter.
- Describe the purpose of a computer operating system.
- Describe the distinguishing features of interactive and non-interactive applications.

Section 1

Readings Review

Computer Systems Simplified



The Machine Cycle

1. The CPU fetches an instruction from main memory.
   - The instruction pointer (IP) contains the address of the instruction.
   - The instruction is copied to the instruction register (IR).
   - The CPU updates the IP to contain the address of the next instruction.
2. The CPU decodes the instruction in the IR.
   - The CPU activates the appropriate circuits in the ALU, or sometimes, in the control unit itself.
3. The CPU calls for the execution of the circuit: electronic signals pass through the circuit.

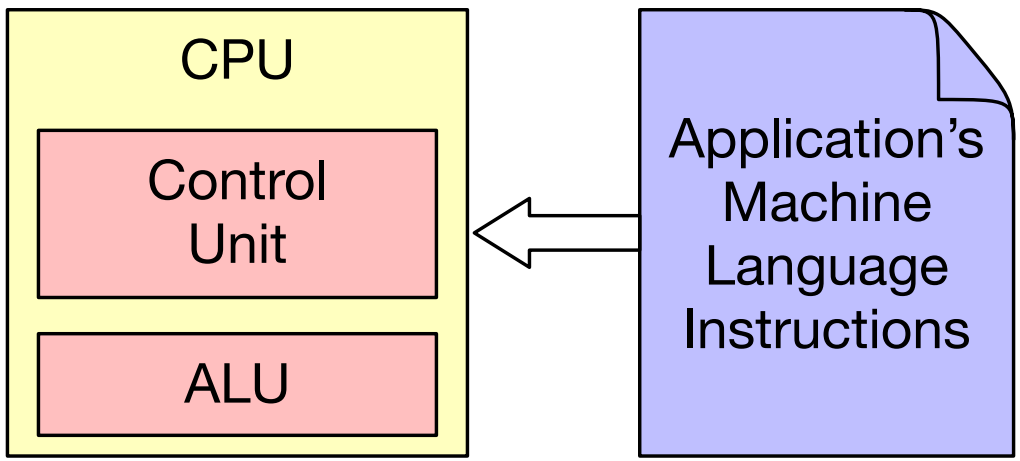
These three operations repeat.

Machine Language Simplified

Each instruction is typically very simple. Examples:

- A load instruction retrieves data from a given address in main memory, and stores it in a given register.
- An add instruction adds the data in 2 given registers together, storing the result in a third register.
- A store instruction sends data from a given register out to a given address in main memory.
- A jump instruction tells the control unit to change the value stored in the instruction pointer.

Machine Language Applications: Information



Programming Languages

- There are hundreds!
- Learning your first language is more difficult because you are also learning to program.
- You will need to learn lots more.
- Each one you learn is a little easier, because the underlying computational concepts are the same.

Compilers and Interpreters

- Programs and scripts are all text documents.
- Compilers, interpreters are machine language applications.
- A compiler:
  - Input: a program as input
  - Task: Translate program to machine language.
  - Output: A machine language application.
  - Note: The compiler does not execute the application.
- An interpreter:
  - Input: a script or program as input
  - Task: Execute the script one line at a time.
  - Output: depends on the script.
  - Note: The interpreter does not output an application.

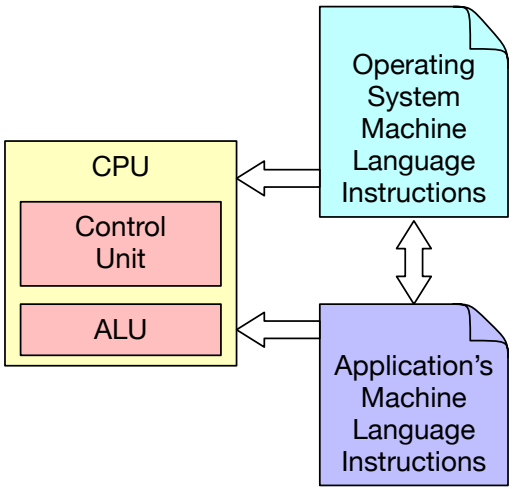
Languages again

- Python is an interpreted language; a Python script needs a Python interpreter to run.
- C/C++ is a compiled language. Once compiled, a computer can run the application directly.
- Java is compiled to byte-code for a virtual machine; a Java application requires the JVM to run.
- (Modern Python interpreters also have a virtual machine, so the issue gets a bit muddy.)

Debuggers and IDEs

- A debugger is an interpreter that allows interactive exploration of the script's variables and data, as the script is being executed interactively.
- An interactive development environment, or IDE, coordinates the interaction between applications like editors, compilers, interpreters, and debuggers.

Operating Systems



- An operating system provides software for applications to interact with shared resources:
  - Multiple applications running at the same time.
  - Peripherals like disk drives, network connections, monitors and screens, etc

Software Applications I

- Graphical User Interface (GUI) applications:
  - User input from keyboard, mouse, touch-screen, stylus, etc
  - Visual display of images, icons, text, for communication to the user
  - Examples: Web-browsers, Microsoft Word, PyCharm IDE, games
- Console applications:
  - User input from keyboard only.
  - Communication with user in text-only
  - Examples: Every program in CMPT 141; debuggers

Software Applications II

- Non-interactive applications:
  - Input usually from data files or initial values provided by the user.
  - Useful for tasks that are repeated a lot.
  - Examples: compilers, interpreters, data analytics, simulations

Section 2

Demonstrations

Computer Architecture Links

- Von Neumann architecture
- CARDIAC

Programming Languages

- Hello world in several languages:
  - C++
  - Java
  - Python
  - LaTeX

Compilers and Interpreters

- Compiling to machine code: C++
- Compiling to byte-code for a virtual machine: Java
- Interpreter: Python
- Bonus: Python byte-code
- Bonus: Typesetting: producing a document as a form of programming!

Applications

- GUI application: TexShop for LaTeX
- Interactive Console Application: Vim Text Editor
- Non-interactive Console application: LaTeX

Section 3

Review

Review of Learning Objectives

Make sure that you can do all of the following:

- Summarize the picture on Slide 6 in words. Explain how these components interact.
- Describe the machine cycle.
- Describe machine language, and why we need programming languages.
- Describe how compiled and interpreted languages differ.
- Explain the differences between IDEs and compilers.
- Describe the services provided by an operating system.