

Learning Objectives

After completing this unit, you should be able to:

- Describe fetch-decode-execute in terms of CPU, Bus, and RAM.
- Encode/decode 16-bit instructions from the Brookshear machine.
- Define the instructions LOAD, STORE, MOVE, ADD, and JUMP.
- Explain the difference (none) between code and data in memory.
- Trace the execution of machine code in registers and in memory.
- Write a simple program (6-10 instructions) in machine language.
- Perform bit masking logic operations using AND, OR, and XOR.

Textbook Sections

- 2.1 Computer Architecture
- 2.2 Machine Language
- 2.3 Program Execution
- 2.4 Arithmetic/Logic

Video Lectures

- Machine Language
- What IS a Dot Diva?
- CS is Changing Everything

Assignments

Exercise03 Hardware; Chapter 2 Problems

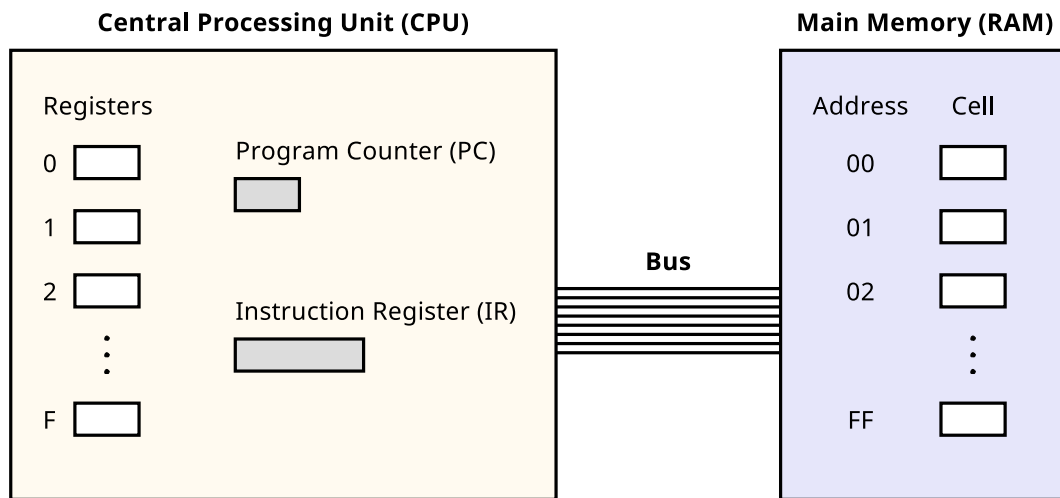
Lab03 Brookshear Machine; CPU and RAM Simulator

Exercise 3A : Hardware

Have you ever wondered what goes on inside your smartphone? Not just how the apps run, but how the hardware actually works? Last week you designed a simple circuit that adds two 4-bit numbers. With enough time and energy, you could learn to build an entire computer! But that's not the goal of this course, so we'll abstract those details. For now, let's take a look at how computers store and process information.

Model 1 Computer Architecture

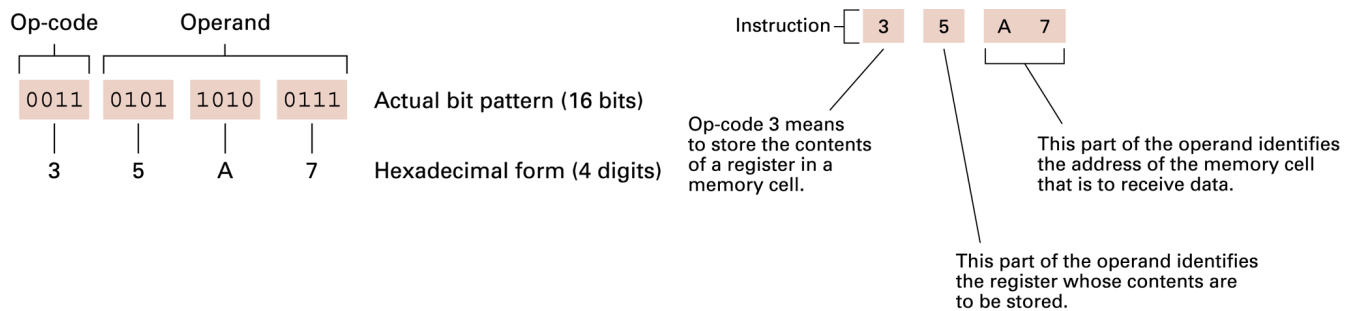
Here is the 8-bit machine described in Appendix C of Brookshear and Brylow (2015):



1. What are the three main components in the diagram? Based on its name, what do you think each hardware component does?
2. How many registers does the CPU have? How many memory cells are in RAM?
3. The CPU has circuits for adding the contents of two registers; think of registers as the "pins" in last week's lab. What is the largest number this machine can add without overflowing?

4. Common tasks the CPU performs include *loading* data from a memory cell into a register and *storing* data from a register into a memory cell. Describe what the CPU would need to do in order to add twenty numbers stored in memory.

Model 2 Machine Instructions



5. How many bits is the op-code? How many possible op-codes can the machine support?

6. The op-code for loading data from memory into a register is 1. Write an instruction (in hex) for loading data from address 3D into register 4.

7. Why is the instruction register in Model 1 twice as large as the other registers? How many memory cells are needed to store a single instruction?