CS1001

Lecture 4

Overview

- Modern Machine Architecture
- Modern Processors
- Memory, Data Storage
- Benchmarks
- Homework 1

Goals

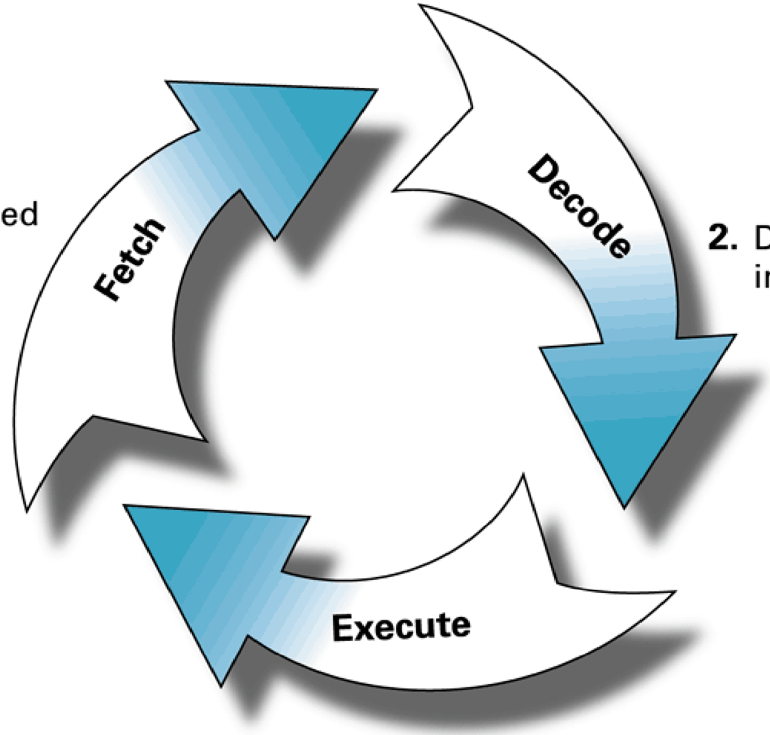
- Understand the components of modern machine architecture
- Examine some basic assembly languages
- Learn how to codify solutions using a given set of actions

Assignments

- Brookshear, Ch 2 (Read)
- Read linked documents on these slides (slides will be posted in courseworks)

Figure 2.8: The machine cycle

1. Retrieve the next instruction from memory (as indicated by the program counter) and then increment the program counter.



2. Decode the bit pattern in the instruction register.

3. Perform the action requested by the instruction in the instruction register.

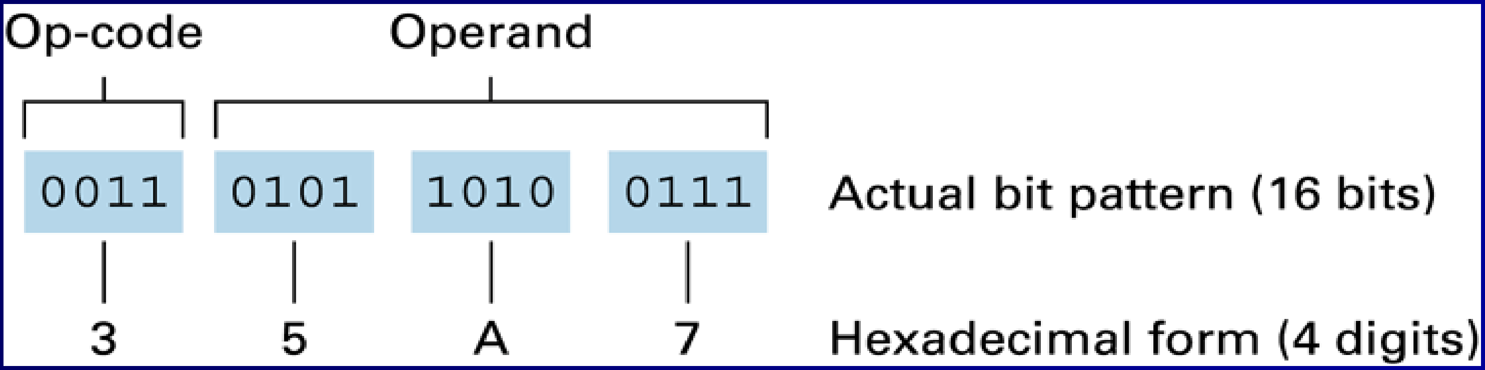
What is an “instruction”

- A sequence of on/off values that specify a command and optionally data to operate on
- Nowadays, we have 32-bit processors; 64-bit processors available. Each instruction is 32 bits on a 32 bit processor, 64 bits on a 64 bit processor

How are instructions regulated?

- All mainstream computers have a clock to control the flow of electricity (data) throughout the computer
- Each clock cycle accomplishes some unit of work, but not necessarily a full instruction

Figure 2.5: The composition of an instruction for the machine in Appendix C



Op-code

Operand

0011 0101 1010 0111 Actual bit pattern (16 bits)

3 5 A 7 Hexadecimal form (4 digits)

Figure 2.1: CPU and main memory connected via a bus

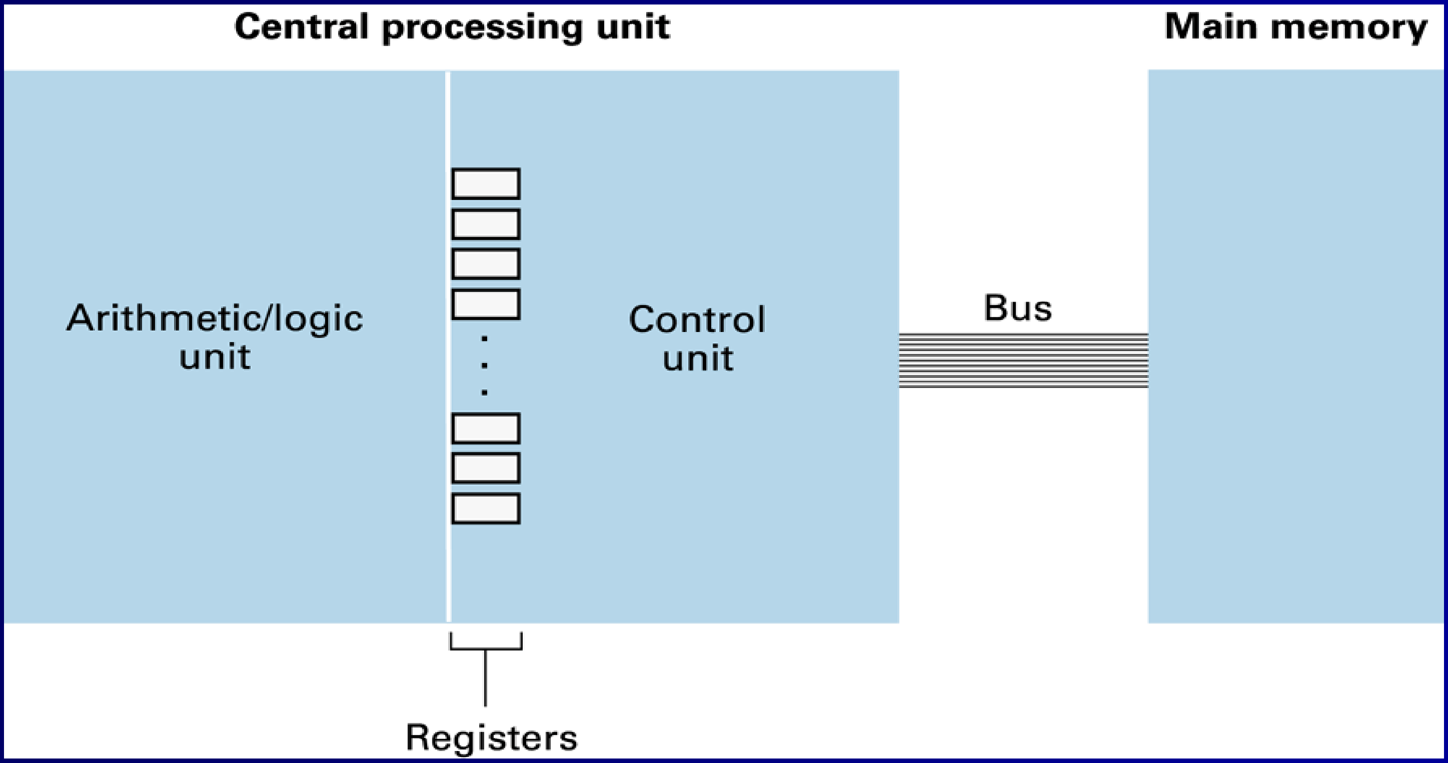


Figure 2.4: The architecture of the machine described in Appendix C

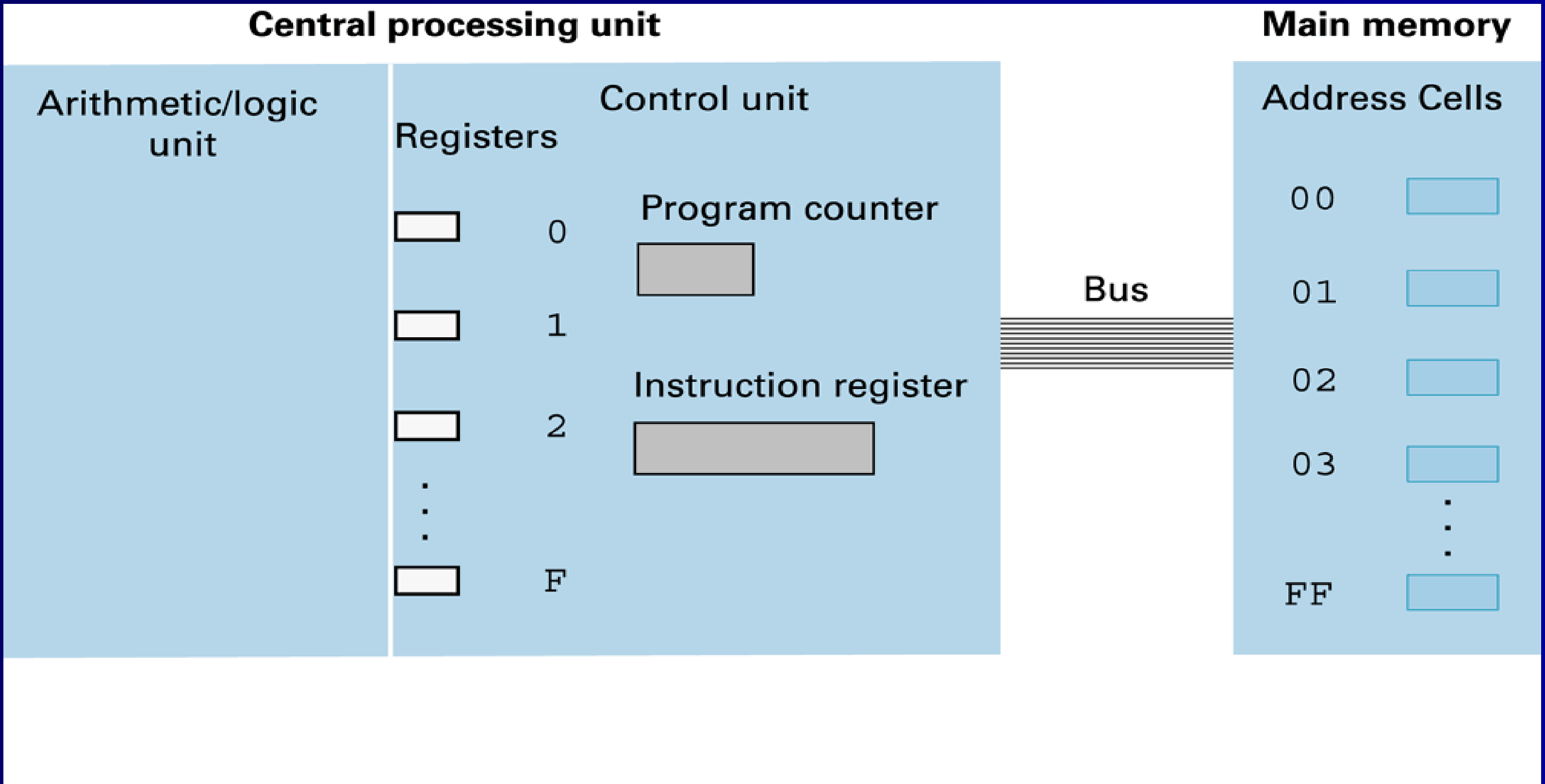
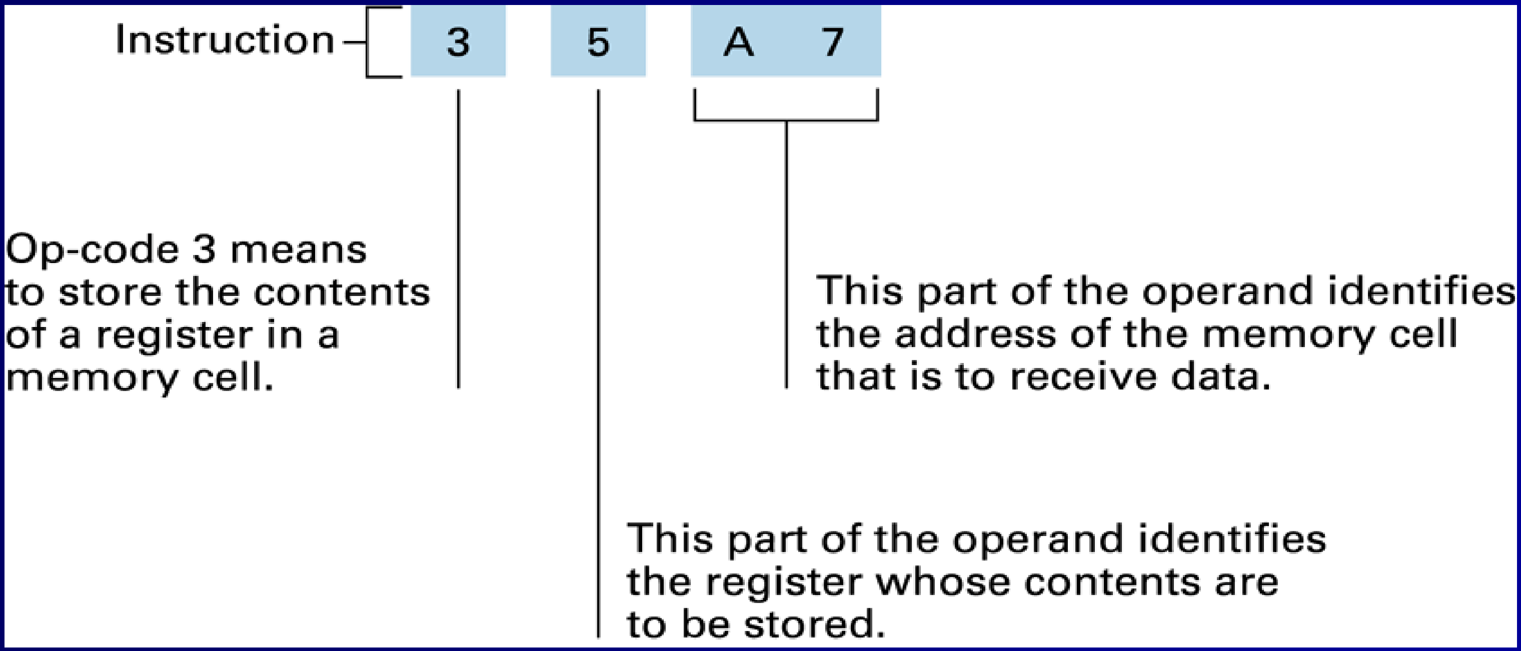


Figure 2.6: Decoding the instruction 35A7

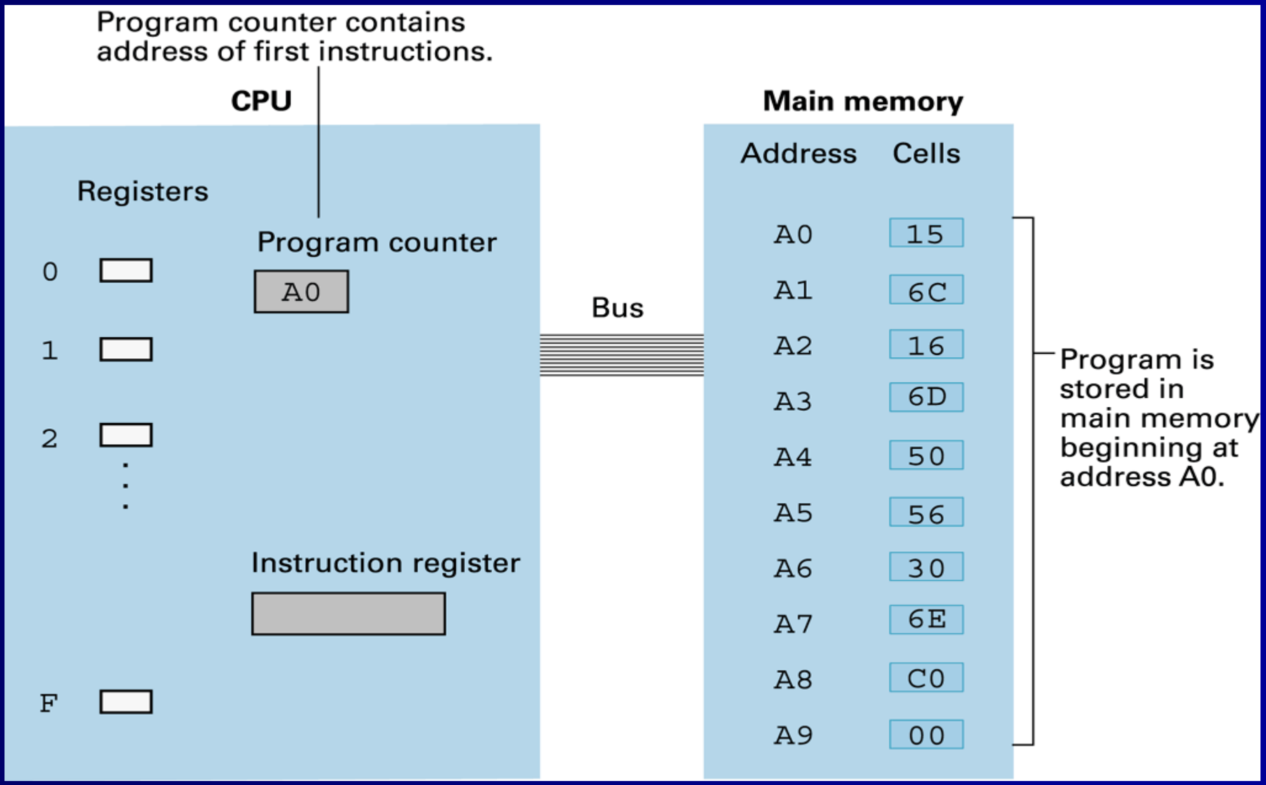


Op-code 3 means to store the contents of a register in a memory cell.

This part of the operand identifies the address of the memory cell that is to receive data.

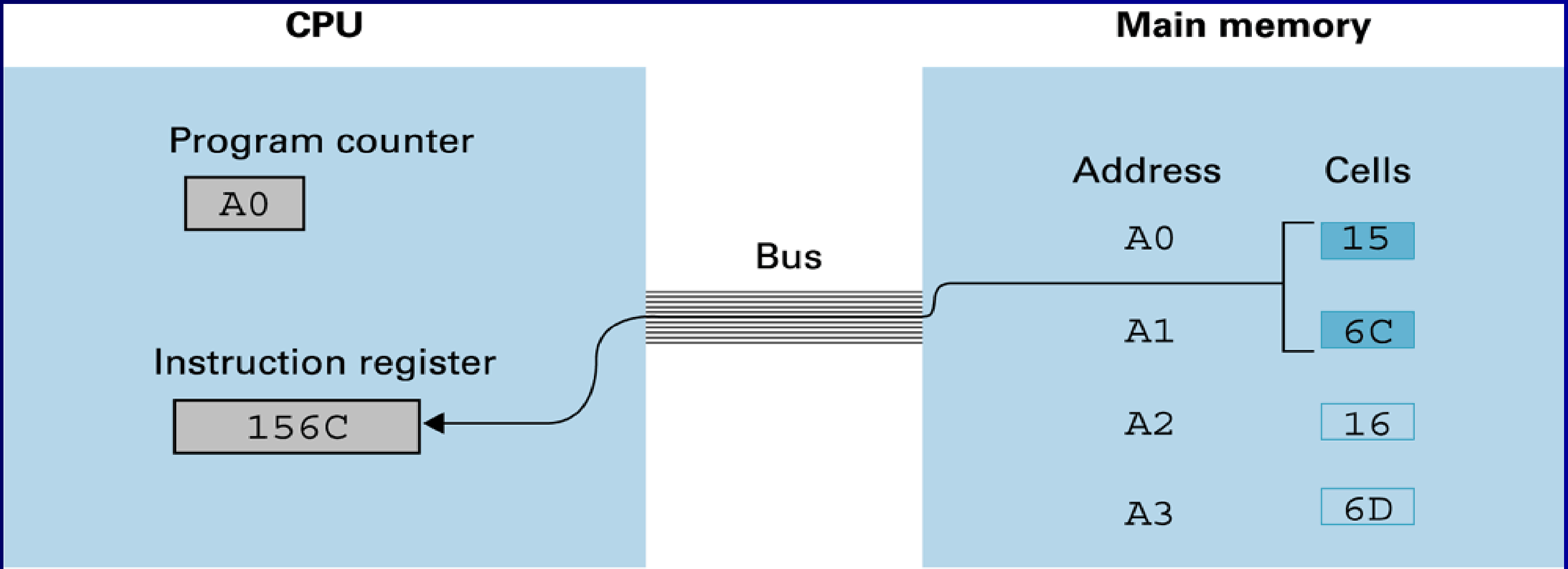
This part of the operand identifies the register whose contents are to be stored.

Figure 2.10: The program from Figure 2.7 stored in main memory ready for execution



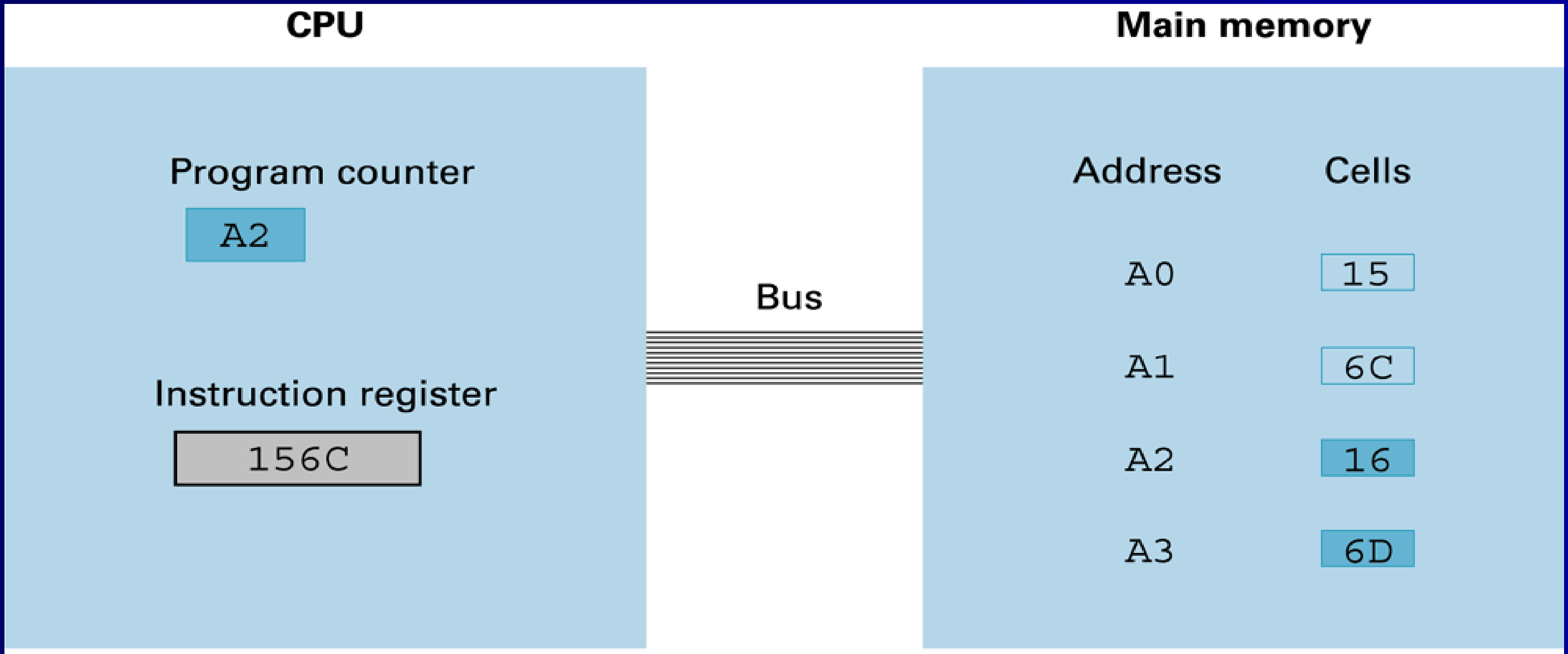
Address	Cells
A0	15
A1	6C
A2	16
A3	6D
A4	50
A5	56
A6	30
A7	6E
A8	C0
A9	00

Figure 2.11: Performing the fetch step of the machine cycle (continued)



a. At the beginning of the fetch step the instruction starting at address A0 is retrieved from memory and placed in the instruction register.

Figure 2.11: Performing the fetch step of the machine cycle



b. Then the program counter is incremented so that it points to the next instruction.

A simple instruction set

- MOVR <id>
- MOVL <id>
- MOVU <id>
- LABEL <labelid>
- GOTO <labelid>

Simulators

- Modern Architectures
  - Intel
  - Apple/Motorola Power
  - ARM
  - Simulation/Emulation

Homework

- The challenge – expressing a process using a fixed number of operations
- The problem:

Two creatures (C1 and C2) are to be parachuted onto random locations on an infinite line. When they land, their parachutes detach and remain where they are. The robots may be programmed from the following instruction set:

Go left one unit	(MOVL <C1 or C2>)
Go right one unit	(MOVR <C1 or C2>)
Label	(LABEL <labelid>)
Skip next instruction unless there is a parachute here	(SKIPPAR)
Go to label	(GOTO <labelid>)

Each instruction takes one cycle to execute.

Program the robots to collide.