Introduction to Computer Organization and Architecture

Topic 2. Instruction set architecture

Han, Nguyen Dinh (han.nguyendinh@hust.edu.vn)

Faculty of Mathematics and Informatics
Hanoi University of Science and Technology

Outline

$\boxed{1}$

Introduction

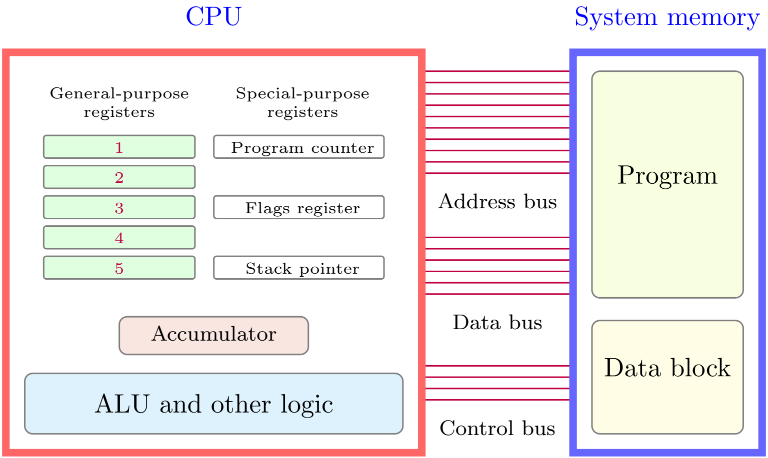
The von Neumann model of a computer

A major advance in computer design occurred in the late 1940s, when John von Neumann had the idea that a computer should be permanently hardwired with a small set of general-purpose operations

It is important to note that the general-purpose computer can solve more-specific problems as it accepts input programs, expressed in machine language, that would organize the basic hardware operations. Thus, to command a computer's hardware, we must speak its language

The von Neumann model consists of a single, shared memory for programs and data, a single bus for memory access, and a separate central processing unit that operates fetching and execution cycles seriously

The von Neumann model of a computer



The stored-program concept

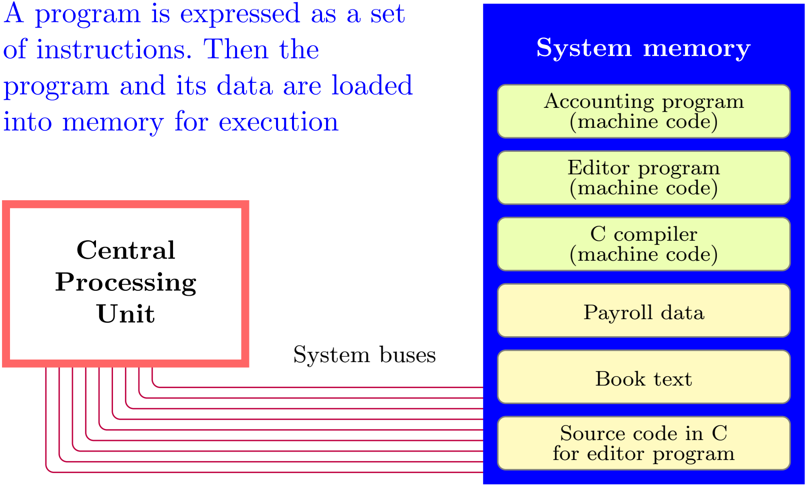
We recall that machine language is the lowest level of programming language that directly corresponds to the binary codes executed by a computer's hardware. The words of a machine's language are called *instructions*, and its vocabulary is called *instruction set*

In a von Neumann computer, a program is expressed as a set of instructions. Then the program is loaded into memory for execution

Although programs written in high-level languages became independent of particular makes and models of computers, these languages still echoed, at a higher level of abstraction, the underlying architecture of the von Neumann model of a machine

The stored-program concept

A program is expressed as a set of instructions. Then the program and its data are loaded into memory for execution



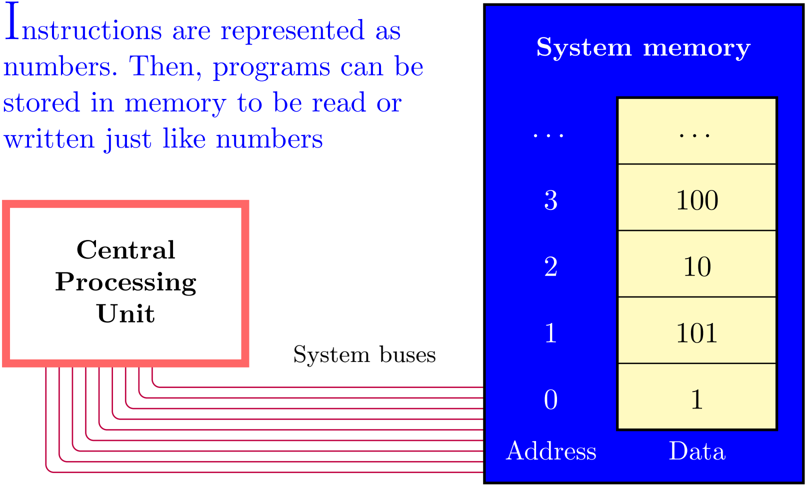
Instruction set architecture

RISC Instruction Set (desktops and servers)	RISC Instruction Set (embedded computers)	CISC Instruction Set
Sun SPARC	MIPS INC MIPS-16	PDP-11
MIPS INC MIPS-64	Mitsubishi M32R	Intel x86
IBM-Mo. PowerPC	Hitachi SuperH	AMD
HP PA-RISC	Thumb	VAX
Digital Alpha	ARM	System/360

$\boxed{2}$

Representing instructions in the computer

Instructions are represented as numbers. Then, programs can be stored in memory to be read or written just like numbers

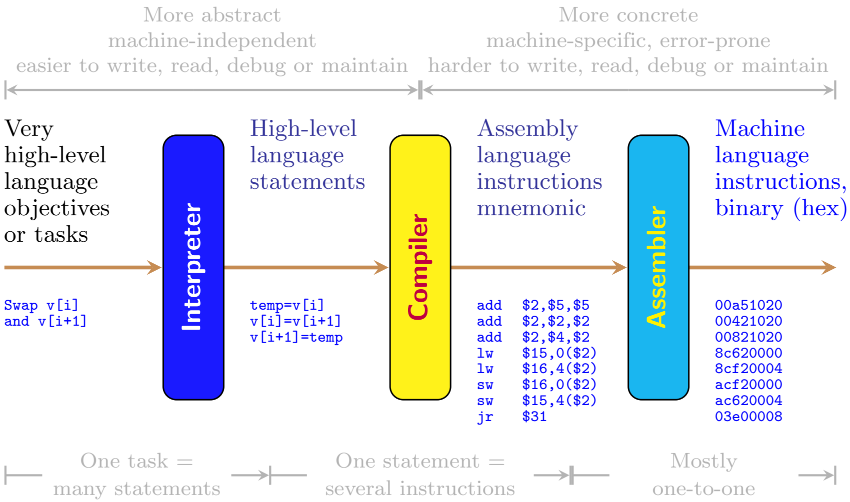


Numbers are kept in the computer hardware as a series of high and low electronic signals, and so they are considered base 2 (or *binary*) numbers

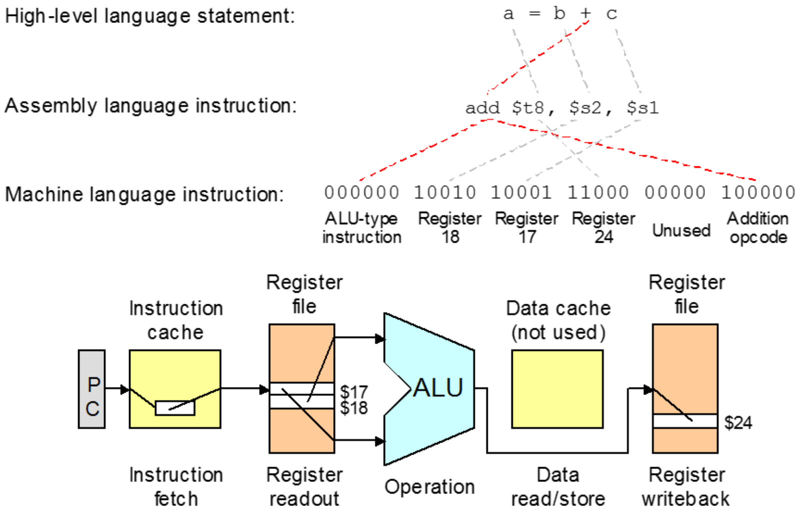
A single digit of a binary number is thus the "atom" of computing because all information is composed of binary digits or *bits*

Instructions are also kept in the computer as a series of high and low electronic signals and may be represented as numbers. We will call the numeric version of instructions *machine language* in order to distinguish it from assembly language. In addition, a sequence of such instructions will be called *machine code*

Representing instructions in the computer



Representing instructions in the computer



$\boxed{3}$

MIPS operands

$\boxed{4}$

MIPS instruction set

$\boxed{5}$

Your second assignment

Your second assignment

Students work individually to carry out the following tasks:

1. Investigate insight-fully the MIPS and RISC-V instruction set architectures
2. Write and run programs in MIPS that allow users to add, subtract, multiply and divide two integer numbers
3. Upgrade your MIPS programs so that they can demonstrate work of instructions for making decisions (i.e. statements represent if-then-else, loop, while, switch, etc.)
4. Convert/rewrite your MIPS programs to RISC-V programs
5. Report (in Word or Latex) and submit your results

NOTE ⇝ Please convert your report to PDF when you submit it

Thank you very much for your attention!