Impact of Machine Architectures

T1 Chapter 2

By,

Prof. Archana Chaudhari

Text Book: T1. T. W. Pratt, M. V. Zelkowitz, "Programming Languages Design and Implementation‖, 4th Ed, PHI, ISBN 81-203-2035-2

Impact of Machine Architectures
T1 Chapter 2

In developing a programming lang, architecture of the software influences the design of a language in two ways

1. The underling computer on which programs written in the language will execute
2. The execution model or virtual comp. that support that language on the actual hardware

- The operation of a computer
  - Computer Hardware
  - Firmware Computers
  - Translators and virtual architecture
- Virtual Computers and Binding Times

The operation of a computer
(T1 Chapter 2)

- A computer is an integrated set of algorithms and data structure capable of storing and executing program
- A computer may constructed physical device using wires, IC, circuit boards, and like it is called actual or hardware computer
- Working of any computer is based on the s/w present in it
- The s/w contains collection of program and these program created using suitable programming language.
- Six major components of Computer are:
  1. Data
  2. Primitive Operations
  3. Sequence Control
  4. Data access
  5. Storage management
  6. Operation Environment

Computer Hardware

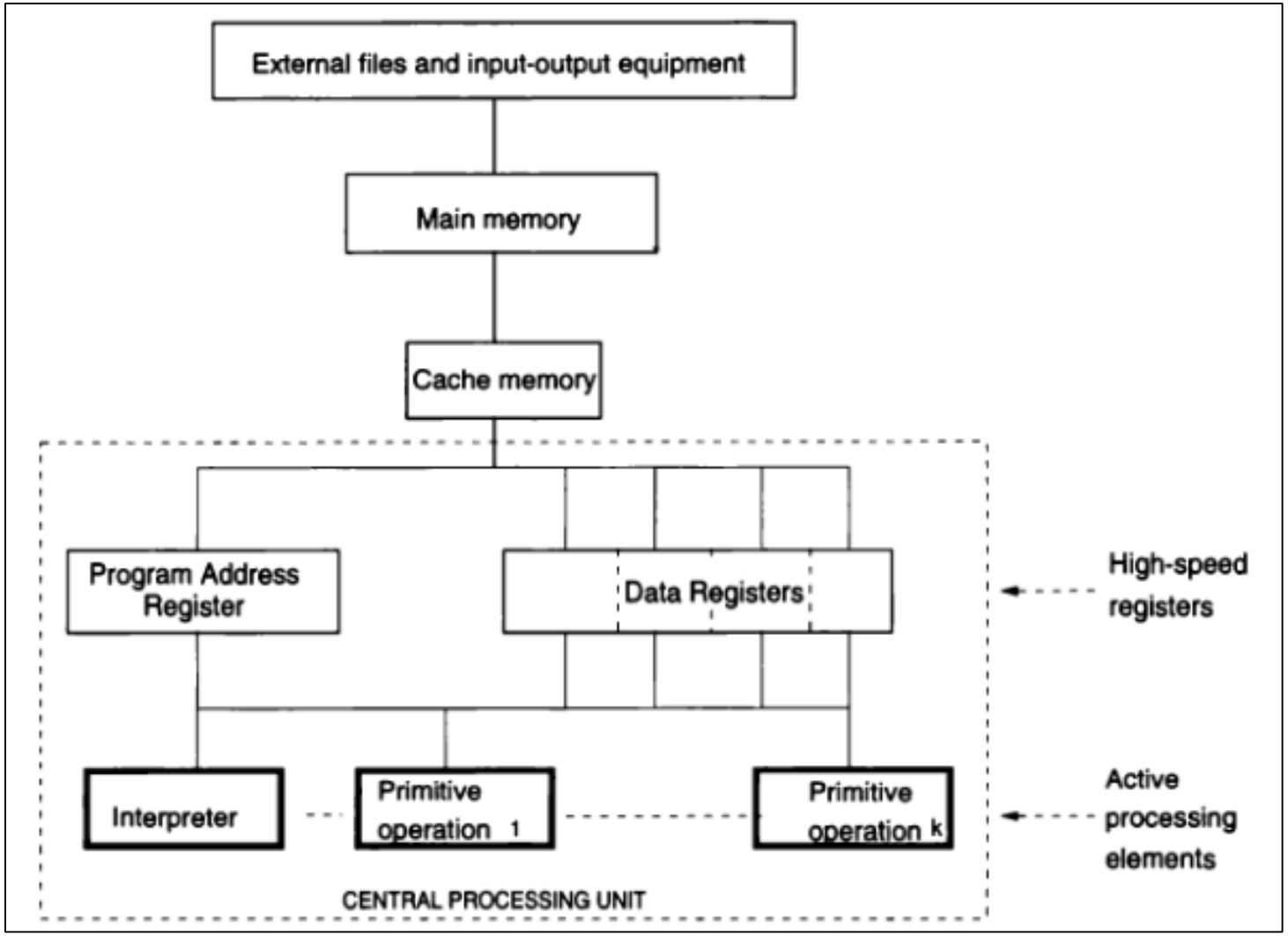


Fig 1: Organization of a conventional computer

1. Data

- A computer must provide various kinds of elementary data items and data structures to be manipulated.
- Three major data storage components: main memory, high-speed registers, and external files.
  - Main memory: is usually organized as a linear sequence of bits subdivided into fixed-length words (typically 32 or 64 bits) or 8-bit bytes (typically 4 or 8 bytes per word).
  - The high-speed registers: The contents of a register may represent data or the address in main memory containing the data or next instruction. A high-speed cache memory is often situated between main memory and the registers as a mechanism to speed up access to data from this main memory.
  - External files: stored on magnetic disk, magnetic tape, or CD-ROM, are usually subdivided into records, each of which is a sequence of bits or bytes.

2. Primitive Operations

- A computer must provide a set of primitive operations useful for manipulating the data.
- It includes:
  - primitives for arithmetic on each built-in numeric data type (e.g., real and integer addition, subtraction, multiplication, and division),
  - primitives for testing various properties of data items (e.g., test for zero, positive, and negative numbers),
  - primitives for accessing and modifying various parts of a data item (e.g., retrieve or store a character in a word and retrieve or store an operand address in an instruction),
  - primitives for controlling input-output (I/O) devices,
  - primitives for sequence control (e.g., unconditional and return jumps).

3. Sequence Control

- A computer must provide mechanisms for controlling the sequence in which the primitive operations are to be executed.
- primitive operations are allowed to modify the program address register to transfer control to another part of the program

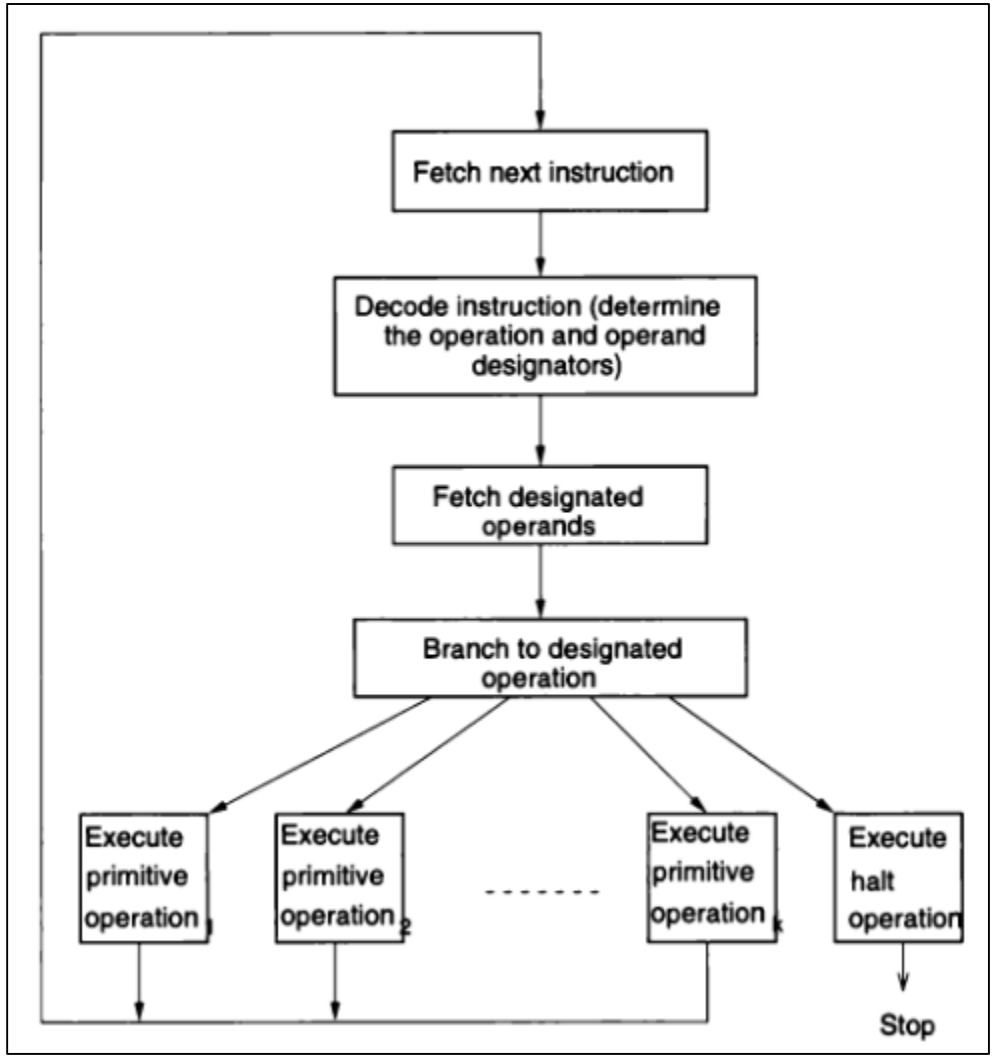


Fig 2: Program interpretation and execution

4. Data access

- A computer must provide mechanisms for controlling the data supplied to each execution of programming instruction.

5. Storage management

- A computer must provide mechanisms to control the allocation of storage for programs and data.
- To keep all resources of the computer (e.g., memory, central processor, external data devices) operating as much as possible.
- Operations within CPU happen at the nanosecond level, accessing memory occurs at the microsecond level, and external data operations occur at the millisecond level
- For speeding up the imbalance between main memory and the central processor, a cache memory is used.
- A cache memory is a small high-speed data storage that is between main memory and the central processor.

6. Operating Environment

- A computer must provide mechanisms for communication with an external environment containing programs and data to be processed.
- The operating environment of a computer ordinarily consists of a set of peripheral storage and I/O devices.

Firmware Computers

- Firmware A set of machine-language instructions implemented by programs, called microprograms, stored in programmable read-only memory in the computer (PROM).
- Firmware Advantages
  - flexibility - by replacing the PROM component we can increase the set of machine instructions. e.g. we can implement vector multiplication as a machine operation.
  - less cost of the hardware - the simpler the instructions, the easier to hardwire them.

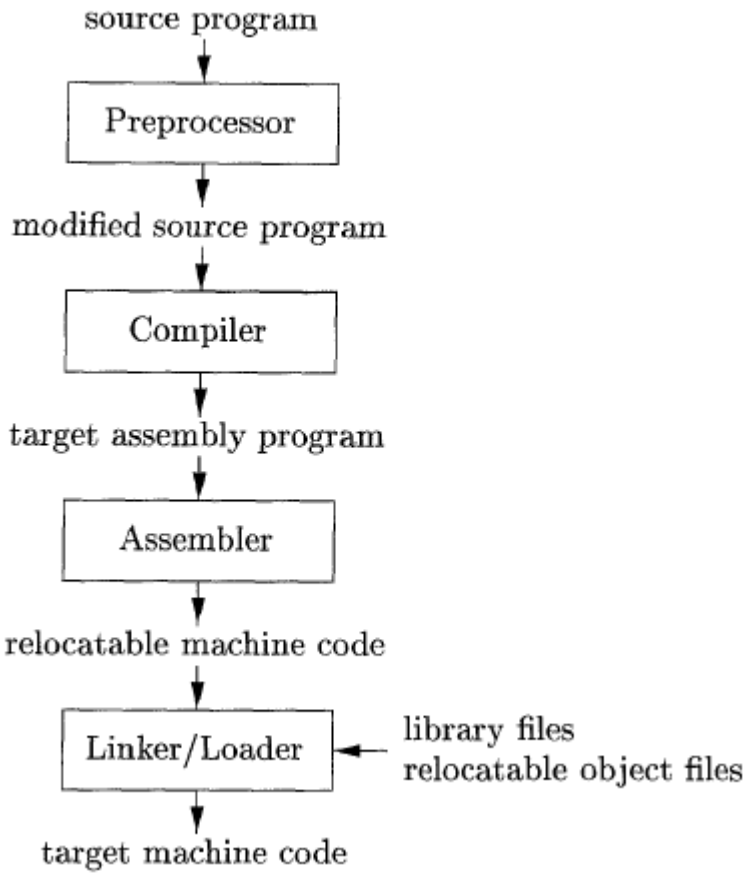
Translators and Virtual Architectures

High-level language ----------→ machine language

- Translation (Compilation)
- Interpretation

Translation (or compilation):

- Input: high-level language program
- Output: machine language code
- Types of translation (high to low-level):
    a) A pre-processor or a macroprocessor:
        #include<stdio.h>
        # define SIZE 5
    b) A compiler
    c) An assembler
    d) A loader or link editor:
- Advantages: efficient executable



Interpretation

- The program is not translated to machine language code.
- Instead, it is executed by another program.
- Example: Prolog interpreter written in C++
- Advantages of interpretation:
  - easy to be implemented
  - easy to debug
  - portability
  - flexibility
  - easy to modify the interpreter

Disadvantages: slow execution

Virtual Computers and Binding Times

- Virtual Computers and Language Implementations
- Hierarchies of Virtual Machines
- Binding and Binding Time

What is a computer ?

- A computer is a programmable machine that receives input, stores and manipulates data or information, and provides output in a useful format.
- While a computer can, in theory, be made out of almost anything, and mechanical examples of computers have existed through much of recorded human history, the first electronic computers were developed in the mid-20th century (1940–1945).

Virtual Computer

- **The Virtual** Computer **is completely reconfigurable in every respect.**
- **Computing machines based on reconfigurable logic are hyper-scalable.**

Binding

- Binding - fixing a feature to have a specific value among a set of possible values.
- E.G. - your program may be named in different ways, when you choose a particular name, then you have done a binding.

Binding time

- **Binding time** is the moment in the program's life cycle when this association occurs.
- Many properties of a **programming language** are defined during its creation.
- For instance, the meaning of key words such as while or for in C, or the size of the integer data type in Java, are properties defined at **language** design **time**.

Binding Occurs at

- language definition
- language implementation
- translation
- execution

Binding

- At language definition - available data types and language structures, e.g in C++ the assignment statement is =, while in Pascal it is :=
- At language implementation - concerns representation of numbers and arithmetic operations
- Binding At translation –
  - Chosen by the programmer - variable types and assignments
  - Chosen by the compiler - relative locations of variables and arrays
  - Chosen by the loader - absolute locations

Binding

- At execution –
  - Memory contents
    - On entry to a subprogram (copying arguments to parameter locations)
    - At arbitrary points (when executing assignment statements)
- Recursive programs
- Dynamic libraries

Importance of binding times

- If done at translation time - efficiency is gained
- If done at execution time - flexibility is gained.
- Example: Consider X = X+10

Thank You