History of Programming Languages

T1 Chapter 1

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.

General Purpose Processor Architecture

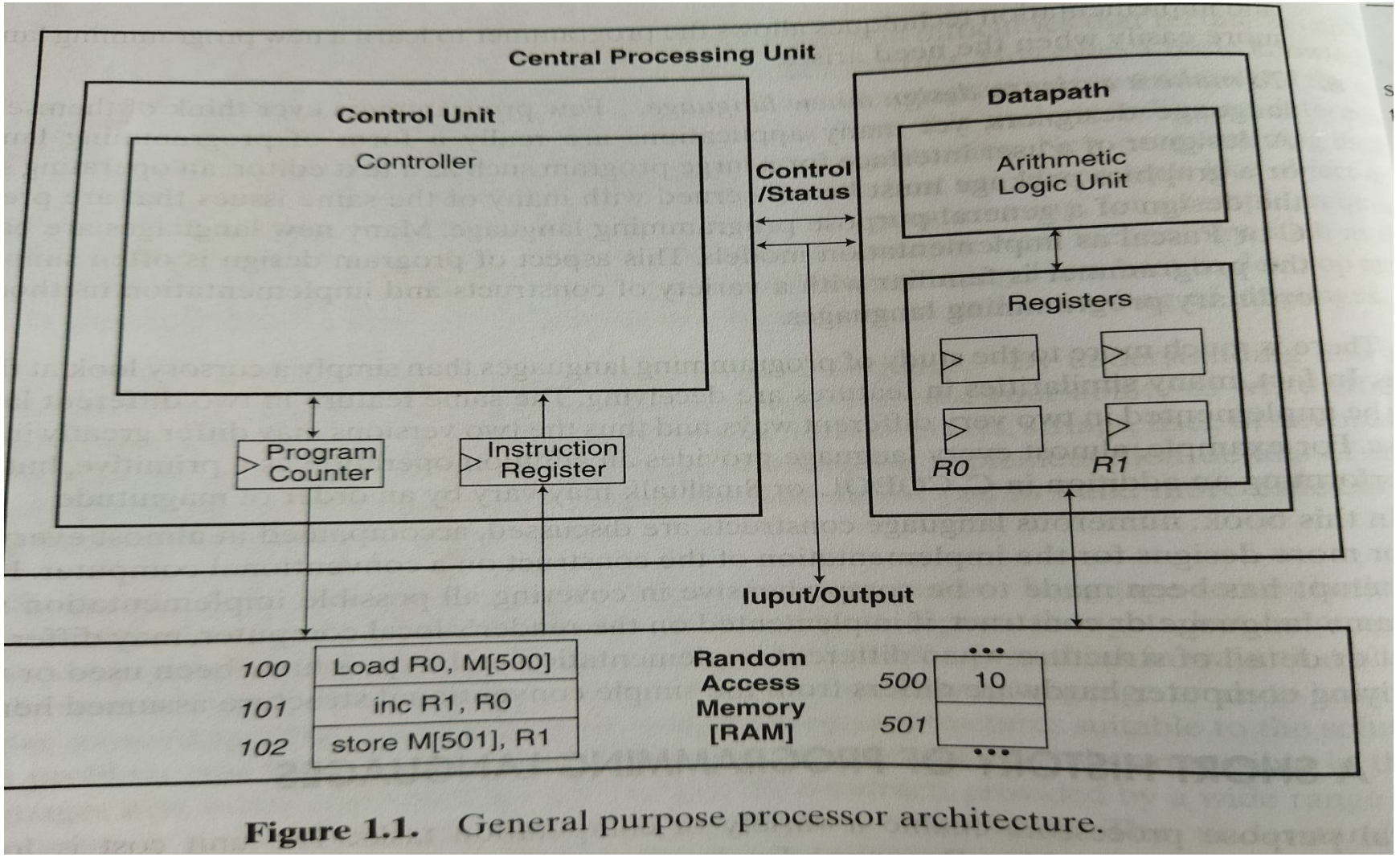


Figure 1.1. General purpose processor architecture.

General Purpose Processor

- The operations of Datapath are:
  - Load : Read Memory location into register
  - ALU Operations: Input Certain register through ALU, store back in register
  - Store: Write register to memory location
- Control Unit : Configures the datapath operations as determined by sequence desired operation stored in memory
- The sequence of instructions stored in the memory is called program

- The set of sub operations during the execution of an instruction is called instruction cycle.
- The Key architectural consideration are:
  - N-bit ALU, registers, buses, memory, data interface
  - PC size determines address space
  - Clock frequency based on the longest register-register delay and memory access time
  - Register model that specifies the number of registers in the CPU and their functionality
  - Memory hierarchy that facilitates the use of cache memory
  - Pipelining and superscalar architectures to enhance performance

Programming Domains

- Scientific applications:
  - In the early 40s computers were invented for scientific applications.
  - The applications require large number of floating point computations.
  - FORTRAN was the first language developed scientific applications.
  - ALGOL 60 was intended for the same use.

Programming Domains(Cont..)

- Business applications:
  - The first successful language for business was COBOL.
  - Produce reports, use decimal arithmetic numbers and characters.
  - The arrival of PCs started new ways for businesses to use computers.
  - Spreadsheets and database systems were developed for business.

Programming Domains(Cont..)

- **Artificial intelligence**
  - Symbolic rather than numeric computations are manipulated.
  - Symbolic computation is more suitably done with linked lists than arrays.
  - LISP was the first widely used AI programming language.

Programming Domains(Cont..)

- **Systems programming**
  – The O/S and all of the programming supports tools are collectively known as its system software.
  – System software need efficiency because of continuous use.
  – It should also have a support for interfacing with external devices to be written
  Eg. Support a programs for plug & play devices
  – C, C++ are usually preferred for system programming

Programming Domains(Cont..)

- **Scripting languages**
  - – Put a list of commands, called a script, in a file to be executed.
  - – PHP is a scripting language used on Web server systems.
  - – Its code embedded in HTML documents.
  - – The code is interpreted on the server before the document is sent to a requesting browser.

Two Levels of Instructions

- Assembly Level (Embedded Systems, drivers)
- Structured Languages Level (C, C++, Java etc)

Very low-level languages

- Those are machine languages and assembly languages, machine-dependent coding systems. They were initially fully binary, and then symbolic.
- There is one native machine language, and usually one assembly language per processor model.
- Upward compatibility .

Evolution of Major Programming Language

FORTRAN

- **Fortran**- Formula Translation
- Fortran was originally developed by a team at IBM in 1957 for scientific calculations.
- Later developments made it into a high level programming language.
- **Fortran** is a general-purpose **imperative programming language** that is especially suited to numeric & Scientific computation
- Fortran ruled this programming area for a long time and became very popular for high performance computing, because.
- It supports −
  - Numerical analysis and scientific computation
  - Structured programming
  - Array programming
  - Modular programming
  - Generic programming
  - High performance computing on supercomputers
  - Object oriented programming
  - Concurrent programming
  - Reasonable degree of portability between computer systems

FORTRAN (Cont..): Facts about Fortran

- Originally developed for scientific calculations, it had very limited support for character strings and other structures needed for general purpose programming.
- Later extensions and developments made it into a high level programming language with good degree of portability.
- Original versions, Fortran I, II and III are considered obsolete now.
- Oldest version still in use is Fortran IV, and Fortran 66.
- Most commonly used versions today are : Fortran 77, Fortran 90, and Fortran 95.
- Fortran 77 added strings as a distinct type.
- Fortran 90 added various sorts of threading, and direct array processing.

LISP

- One of the earliest programming languages.
- Based on the concept of computing by evaluating functions. Very good for symbolic computing.
- For years, the only language for Artificial Intelligence work. (Prolog is 12 years younger.)
- Two kind of Data structure: atoms and lists
- Two dialects of LISP are: Scheme, COMMON LISP.
- Scheme is well suited to educational application.
- Lisp's successors are very elegant (Miranda, ML, Haskell) but not nearly as widely used.

LISP Example

```
; Lisp Example function
; The following code defines a Lisp predicate function
; that takes two lists as arguments and returns True
; if the two lists are equal, and NIL (false) otherwise
  (DEFUN equal_lists (lis1 lis2)
   (COND
     ((ATOM lis1) (EQ lis1 lis2))
     ((ATOM lis2) NIL)
     ((equal_lists (CAR lis1) (CAR lis2))
                 (equal_lists (CDR lis1) (CDR lis2)))
     (T NIL)
   )
)
```

Algol 60

- It was the first to have block structure, recursion, and a *formal definition*.
- It is not used now, but it is the ancestor of most contemporary languages.
- As far as design goes, Algol 60 was without doubt the most important innovation in the history of programming languages

Algol 60 Example

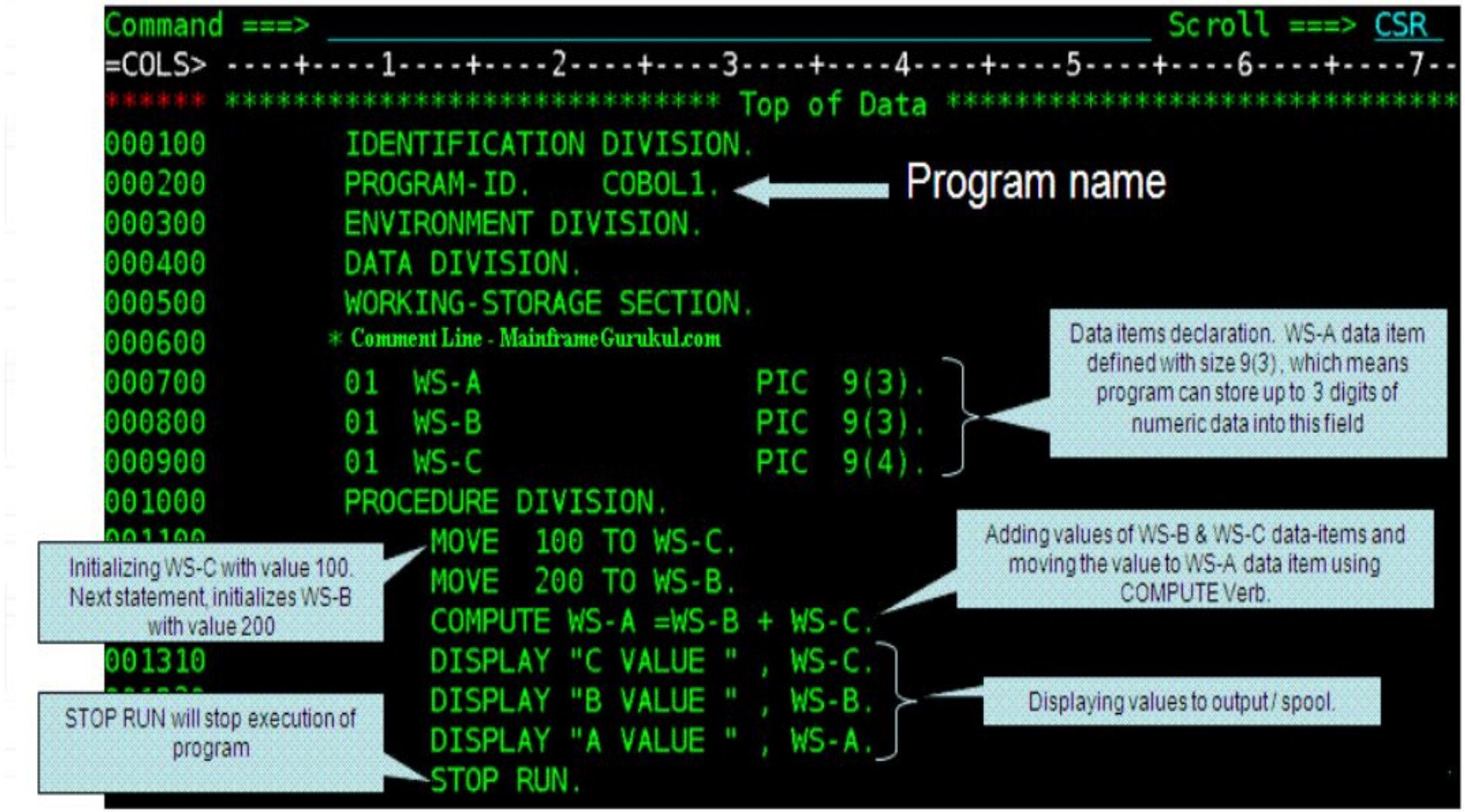
```
let n = readi
let x = vector 1::n of 0.0
for i = 1 to n do x( i ) := readr
!bubble sort starts here
for i = 1 to n - 1 do
   for j = 1 to n - i do
   if x( j ) > x( j + 1 ) do
   begin
        let temp = x( j )
        x( j ) := x( j + 1 )
        x( j + 1 ) := temp
   end
!now write out the answers
write "The sorted numbers are'n'n"
for i = 1 to n do write x( i ),"'n" ?
```

Cobol

- Business-oriented computations
  - very strict program organization
  - poor control structures
  - elaborate data structures, record type introduced for the first time.

  Used to be very popular in business and government, much less at universities.

Cobol Example



Basic

- The first in history language of personal computing.
- The *first* programming language for many programmers: designed to be easy to learn.
- Very simple, limited, though still general-purpose.
- The current popular version of BASIC is visual basic

PL/I

- A combination of features believed (at the time) best in Fortran, Algol 60, Cobol.
  - the first language designed to be completely general, good for all possible applications
  - actively promoted by IBM
  - not used much today.
- An interesting feature introduced in PL/I:
  event handling.

Simula 67

- An extension of Algol 60 designed for simulation of concurrent processes.
- Introduced the central concepts of object orientation: classes and encapsulation.
- Predecessor of Smalltalk and C++.
- Now unused.

Algol 68

- A very elegant design, unmatched till today.
- Full orthogonality.
- Extremely difficult to implement.
- A very clever formal description, unfortunately hard to understand for most potential users.
- Completely unused.

Pascal

- A conceptually simplified and cleaned-up successor of Algol 60.
- A *great* language for teaching structured programming.
- An excellent *first* language to learn: teaches good programming habits.
- Its later extensions (for example, Delphi) are full-fledged systems programming packages, as powerful as any Java kit.

Modula-2

- A better, conceptually uniform successor of Pascal.
- Mechanisms to program concurrency (many processes running in parallel).
- Not used as much as it deserves.
- Its successors, Modula-3 and Oberon, are even more conceptually appealing, practically useful—and almost not used at all. (They lost the popularity contest with C++.)

Prolog

- A very high-level programming language.
- Declarative, based on a subset of logic, with proofs interpreted as computation.
- Very powerful:
  - Non-deterministic (built-in backtracking).
  - Elaborate, flexible pattern matching.
  - Associative memory.
  - Pattern-directed procedure invocation.
- In skilled hands, it is a very strong tool.

Ada

- The result of an elaborate, multi-stage design process, and a more successful attempt at generality than PL/I.
- Completely standard: there can be no dialects (like Java, except that Microsoft...).
- [Dialect: a particular form of a language which is peculiar to a specific region or social group]
- however, two standards: Ada 83 (the original), and Ada 95.
- Ada has been designed to support concurrency in a very neat, systematic way.

Smalltalk

- It is the purest object-oriented language ever designed (till now), cleaner than Java, much cleaner than C++.
- Comes complete with a graphical interface and an integrated programming environment.
- In skilled hands, a powerful tool.

C

- The implementation language of Unix.
- A great tool for systems programming and a software development language on personal computers.
- Once fashionable, still in use, but usually superseded by C++.
- Dangerous if not used properly:
  not recommended to novice programmers.
- Powerful set of operators.

C++

- An *object-oriented* extension of the *imperative* language C.
- This is a hybrid design, with object orientation added to a completely different base language.
- Complicated syntax, difficult semantics.
- Very fashionable, very much in demand.

Java

- A neat, cleaned up, sized-down reworking of C++.
- Full object orientation (though not as consistent as Smalltalk)
- Designed for Internet programming, but general-purpose.
- It contains references but no pointer.
- It has a facility and support for applets and concurrency

Scripting languages

- Text processing:
  - Perl
  - Python
- Web programming
  - JavaScript
  - PHP

.NET Language: C#

- It is based on C++, Java and Delphi
- It includes pointers, delegates, properties, enumeration types, a limited kind of dynamic typing, and anonymous type.
- It evolving rapidly