Electrical Engineering

Computer Fundamentals

Comprehensive Theory

with Solved Examples and Practice Questions

MADE EASY Publications

MADE EASY Publications

Corporate Office: 44-A/4, Kalu Sarai (Near Hauz Khas Metro Station), New Delhi-110016
E-mail: infomep@madeeasy.in
Contact: 011-45124660, 8860378007
Visit us at: www.madeeasypublications.org

Computer Fundamentals

First Edition : 2015
Second Edition : 2016
Third Edition : 2017
Fourth Edition : 2018

Contents

Computer Fundamentals

Chapter 1
Basic Architecture 2

1.1 Computer System 2
1.2 Layers of Abstraction 3
1.3 Computer Organization and Computer Architecture 3
1.4 Evolution of Digital Computers 5
1.5 Structure and Function of a Computer System 5
1.6 Components of Computer Structure 6
1.7 Bus Structure 7
Student's Assignment 9

Chapter 2
CPU Organization 11

2.1 Introduction 11
2.2 Machine Instructions 12
2.3 Instruction Formats 13
2.4 Addressing Modes 16
2.5 Types of Machine Instructions 23
2.6 Instruction Stream Vs Data Stream 26
2.7 ALU Design 27
2.8 Bus Organizations of Datapath 30
2.9 Implementations of Datapath 31
2.10 Basics of Control Unit 31
2.11 Multi Cycle Datapath and Control 34
2.12 Control Unit 36
2.13 Microoperations and Control Signals 37
2.14 Control Unit Implementation 40
Student's Assignment 47

Chapter 3
Memory Organization 52

3.1 Introduction 52
3.2 Memory or Primary Memory (Core Memory/ Store/Storage) 53
3.3 Associative Memory 56
3.4 Address Space 56
3.5 Cache Memory 60
3.6 Direct Mapped Cache (1-way Set Associative Cache) 62
3.7 Fully Associative Cache 65
3.8 Set Associative Cache (k-way Set Associative Cache) 66
3.9 Virtual Memory 70
3.10 Page Fault 72
3.11 Page Replacement 73
3.12 Page Replacement Algorithms 74
3.13 Advantages and Disadvantages of Virtual Memory 79
Student's Assignment 83

Chapter 4
Input/Output Organization 88

4.1 Introduction 88
4.2 I/O Access Structure 88
4.3 I/O Modules 89
4.4 I/O Techniques 90
4.5 Direct Memory Access 94
4.6 Secondary Storage 98
4.7 Magnetic Memory 100
4.8 Optical Memory 102
4.9 Structure of a Disk (Disk Structure) 105
Student's Assignment 109

Chapter 5
Data Representation 112

5.1 Fixed-Point Representation 112
5.2 IEEE Floating-Point Number Representation 116
5.3 Computer Arithmetic 117
5.4 Adding 2's Complement Numbers 118
5.5 Multiplying Floating-Point Numbers 119
Student's Assignment 122

Chapter 6

Basic Concepts of Operating Systems ..125

6.1 Operating System (OS) 125
6.2 Structure of Computer System 125
6.3 Layered View of Operating System Services .. 126
6.4 History of Operating System 126
6.5 Types of Operating System 126
6.6 Functions of Operating System 129
6.7 Operating System Components 130
6.8 Process 130
6.9 Process State Models 131
6.10 Thread 136
6.11 Multi Threading 138
Student's Assignment 141

Chapter 7

File System 145

7.1 Introduction 145
7.2 Directories 146
7.3 File Management System 148
7.4 File System Organization 149
7.5 File Allocation Methods 150
Student's Assignment 157

Chapter 8

Networking Fundamentals 159

8.1 Introduction 159
8.2 Delays in Computer Networks 160
8.3 Protocol Layering 160
8.4 Circuit-Switched 164
8.5 Packet Switching 164
Student's Assignment 165

Chapter 9

Programming Methodology 168

9.1 Data Segments in Memory 168
9.2 Scope of Variable 170
9.3 C Variable 171
9.4 Address arithmetic in C 176
9.5 Value of Variable in C Language 176
9.6 Flow Control in C 177
9.7 Function 185
9.8 Recursion 193
9.9 Backtracking 195
9.10 C Scope Rules 196
9.11 Storage Class 198
9.12 Pointers 206
Student's Assignment 216

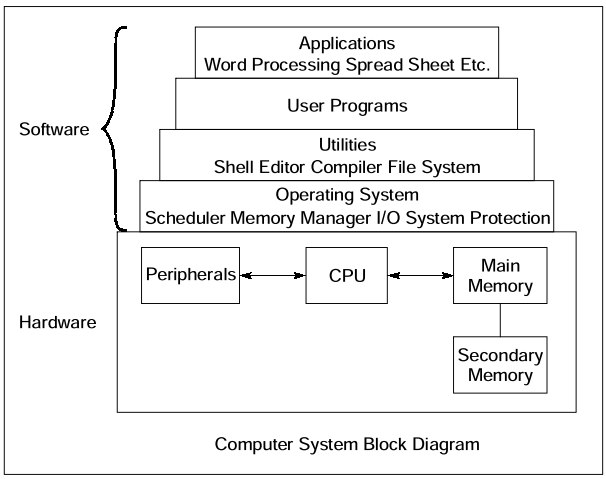
■■■■

CHAPTER 1

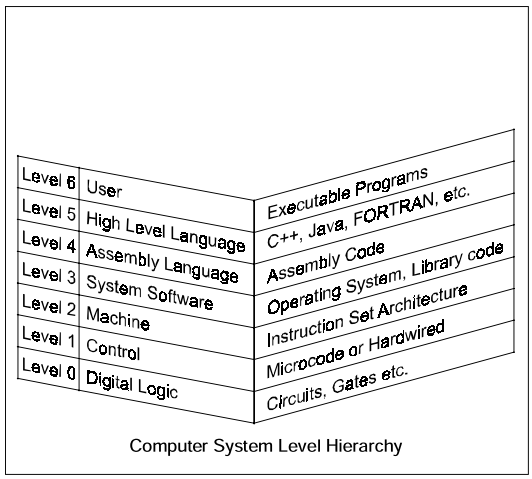
Basic Architecture

1.1 Computer System

Computer system is divided into two functional entities: Hardware and Software.



Computer System Block Diagram



Computer System Level Hierarchy

- **Hardware:** Lowest level in a computer are all the electronic circuits and physical devices from which it is built.

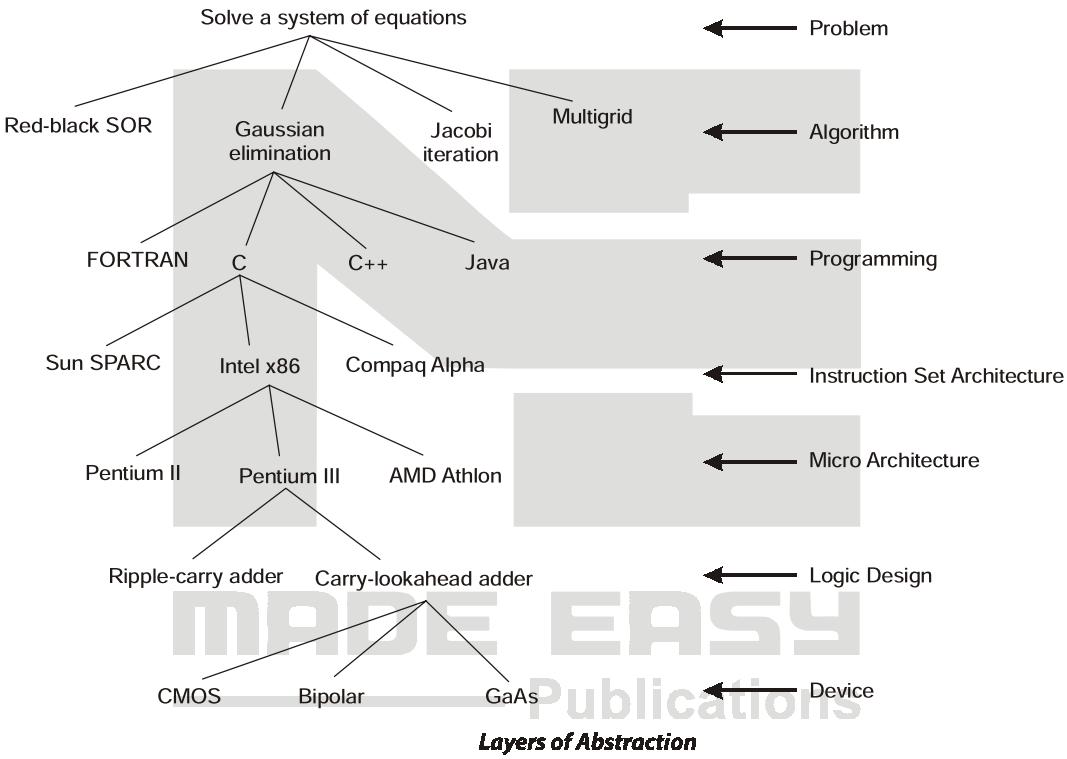
  Hardware consisting of its physical devices (CPU, memory, bus, storage devices, ...)
- **Software:** Sequences of instructions and data that make computers do useful work.

  Software, consisting of the programs it has (Operating system, applications, utilities, ...)

  Program is a sequence of instructions for a particular task.
- Operating system is set of programs included in system software package and Link between hardware and user needs.

1.2 Layers of Abstraction

- **Problem Statement:** stated using "natural language". It may be ambiguous are imprecise.
- **Algorithm:** step-by-step procedure, guaranteed to finish. It is definiteness, effective computability, and finiteness.
- **Program:** Express the algorithm using a computer language such as high-level language and low-level language.
- **Instruction Set Architecture (ISA):** It specifies the set of instructions the computer can perform using data types and addressing modes.
- **Micro-architecture:** It is detailed organization of a processor implementation.
- **Logic Circuits:** Combine basic operations to realize micro-architecture.
- **Devices:** Which is properties of materials and manufacturability.



Layers of Abstraction

1.3 Computer Organization and Computer Architecture

Computer design: The determination of how to interconnect the components and which components to used based upon some specifications.

1.3.1 Computer Architecture (CA)

- Computer architecture is the conceptual design and fundamental operational structure of a computer system. It is a functional description of requirements and design implementations for the various parts of a computer.
- It is the science and art of selecting and interconnecting hardware components to create computers that meet functional, performance and cost goals.

1.4 Evolution of Digital Computers

First generation: Vacuum tube computers (1945~1953)

- Program and data reside in the same memory (stored program concepts: John von Neumann)
- Vacuum tubes were used to implement the functions (ALU & CU design)
- Magnetic core and magnetic tape storage devices are used.
- Using electronic vacuum tubes, as the switching components.
- Assembly level language is used

Second generation: Transistorized computers (1954~1965)

- Transistor were used to design ALU & CU
- High Level Language is used (FORTRAN)
- To convert HLL to MLL compiler were used
- Separate I/O processor were developed to operate in parallel with CPU, thus improving the performance
- Invention of the transistor which was faster, smaller and required considerably less power to operate

Third generation: Integrated circuit computers (1965~1980)

- IC technology improved
- Improved IC technology helped in designing low cost, high speed processor and memory modules
- Multiprogramming, pipelining concepts were incorporated
- DOS allowed efficient and coordinate operation of computer system with multiple users
- Cache and virtual memory concepts were developed
- More than one circuit on a single silicon chip became available.

Fourth generation: Very large scale integrated (VLSI) computers (1980~2000)

- CPU termed as microprocessor
- INTEL, MOTOROLA, TEXAS,NATIONAL semiconductors started developing microprocessor
- Workstations, microprocessor (PC) & Notebook computers were developed
- Interconnection of different computer for better communication LAN, MAN and WAN
- Computational speed increased by 1000 times
- Specialized processors like Digital Signal Processor were also developed.

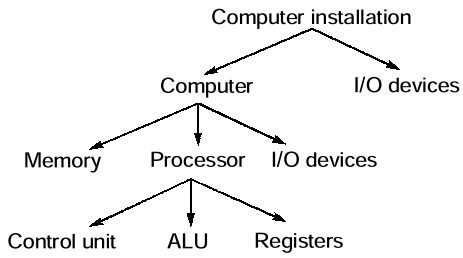
Fifth generation: System-on-chip (SOC) computers (2000~)

- E-Commerce, E- banking, home office
- ARM, AMD, INTEL, MOTOROLA
- High speed processor - GHz speed
- Because of submicron IC technology lot of added features in small size.

1.5 Structure and Function of a Computer System

The designer need only deal with a particular level of the system at a time. At each level, the system consists of a set of *components and their interrelationships*.

The behavior at each level depends only on a simplified, abstracted characterization of the system at the next lower level. At each level, the designer is concerned with structure and function. Important relationships are explained in the figure.

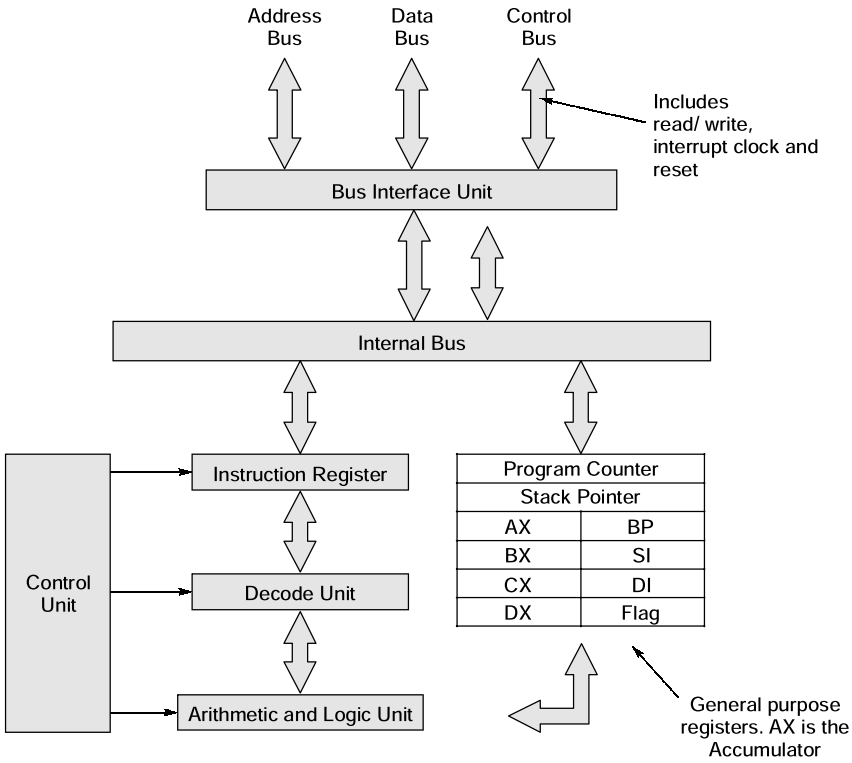


Structure is the way in which components relate to each other (shown in the following figure). Function is the operation of individual components as part of the structure. Functions are Data processing, Data storage, Data movement and Control.

1.7.2 CISC and RISC Architectures

CISC (Complex Instruction Set Computers)	RISC (Reduced Instruction Set Computers)
• Large instruction set	• Compact instruction set
• Instruction formats are of different lengths	• Instruction formats are all of the same length
• Instructions perform both elementary and complex operations	• Instructions perform elementary operations
• Control unit is micro- programmed	• Control unit is simple and hardwired
• Not pipelined or less pipelined	• Pipelined
• Single register set	• Multiple register set
• Numerous memory addressing options for operands	• Compiler and IC developed simultaneously
• Emphasis on hardware	• Emphasis on software
• Includes multi-clock complex instructions	• Single-clock, reduced instruction only
• Memory-to-memory: "LOAD" and "STORE" incorporated in instructions	• Register to register: "LOAD" and "STORE" are independent instructions
• Small code sizes, high cycles per second	• Low cycles per second, large code sizes
• Transistors used for storing complex instructions	• Spends more transistors on memory registers
Examples of CISC processors: • VAX • PDP-11 • Motorola 68000 family • Intel x86 architecture based processors.	**Examples** of RISC processors • Apple iPods (custom ARM7TDMI SoC) • Apple iPhone (Samsung ARM1176JZF) • Nintendo Game Boy Advance (ARM7) • Sony Network Walkman (Sony in-house ARM based chip)

1.7.3 General CPU Architecture (8086 Microprocessor)



1.7.4 Issues of Computer Design

- Cannot assume infinite speed and memory.
- Speed mismatch between memory and processor
- Handle bugs and errors
- Multiple processors, processes, threads
- Shared memory
- Disk access
- Better performance with reduced power

Summary

- **Computer:** A device that accepts input, processes data, stores data, and produces output, all according to a series of stored instructions.
- **Hardware:** Includes the electronic and mechanical devices that process the data; refers to the computer as well as peripheral devices.
- **Software:** A computer program that tells the computer how to perform particular tasks.
- **Computer organization:** Interconnection of hardware to form the computer system
- **Computer architecture:** the structure and behaviour of the computer perceived by the user.
- **Input:** Whatever is put into a computer system.
- **Data:** Refers to the symbols that represent facts, objects, or ideas.
- **Information:** The results of the computer storing data as bits and bytes; the words, numbers, sounds, and graphics.
- **Output:** Consists of the processing results produced by a computer.
- **Main Memory:** Area of the computer that temporarily holds data waiting to be processed, stored, or output. *Example:* Cache and Main memory
- **Secondary Storage:** Area of the computer that holds data on a permanent basis when it is not immediately needed for processing. Example: Disk, Floppy, etc.

Student's Assignments

Q.1 What does CISC and RISC means?
(a) Common instruction set controller and rare instruction set controller
(b) Complex instruction set controller and reduced instruction set controller
(c) Compiled instruction set source code and recompiled instruction source code
(d) None of the above

Q.2 A 32-bit address bus allows access to a memory of capacity
(a) 64 Mb (b) 16 Mb
(c) 1 Gb (d) 4 Gb

Q.3 The system bus is made up of
(a) data bus
(b) data bus and address bus
(c) data bus and control bus
(d) data bus, control bus and address bus

Q.4 Which of the following is not involved in a memory write operation?
(a) MAR (b) PC
(c) MDR (d) data bus

Q.5 The read/write line
(a) belongs to the data bus
(b) belongs to the control bus
(c) belongs to the address bus
(d) CPU bus

Q.6 _____ is a piece of hardware that executes a set of machine-language instructions.
(a) controller (b) bus
(c) processor (d) motherboard

Q.7 Given below are some statements associated with the registers of a CPU. Identify the false statement.
(a) The program counter holds the memory address of the instruction in execution.
(b) Only opcode is transferred to the control unit.
(c) An instruction in the instruction register consists of the opcode and the operand.
(d) The value of the program counter is incremented by 1 once its value has been read to the memory address register.

Q.8 In Flynn's classification of computers, the vector and array classes of machines belong to
(a) Single instruction/single data category
(b) Single instruction/multiple data category
(c) Multiple instruction/single data category
(d) Multiple instruction/multiple data category

Q.9 The following are four statements regarding what a CPU with only a set of 32 bit registers can perform?
1. Hold and operate on 32 bit integers
2. Hold and operate on 16 bit integers
3. Hold and operate on 64 bit floating point arithmetic
4. Hold and operate on 16 bit UNICODE characters

Which of the following is true about such a CPU?
(a) all are true (b) 1,2 and 3 only
(c) 1,2 and 4 only (d) 1,3 and 4 only

Q.10 The following are four statements about Reduced Instruction Set Computer (RISC) architectures.
1. The typical RISC machine instruction set is small, and is usually a subset of a CISC instruction set.
2. No arithmetic or logical instruction can refer to the memory directly.
3. A comparatively large number of user registers are available.
4. Instructions can be easily decoded through hard-wired control units.

Which of the above statements is true?
(a) 1 and 3 only
(b) 1,3 and 4 only
(c) 1, 2 and 3 only
(d) All of these

Q.11 The word length of a CPU is defined as
(a) the maximum addressable memory size.
(b) the width of a CPU register (integer or float point).
(c) the width of the address bus.
(d) the number of general purpose CPU registers.

Q.12 Which of the following statements is false about CISC architectures?
(a) CISC machine instructions may include complex addressing modes, which require many clock cycles to carry out.
(b) CISC control units are typically micro-programmed, allowing the instruction set to be more flexible.
(c) In the CISC instruction set, all arithmetic/logic instructions must be register based.
(d) CISC architectures may perform better in network centric applications than RISC.

Q.13 Which one is required while establishing the communication link between CPU and peripherals?
(a) Synchronization mechanism
(b) Conversion of signal values
(c) Operating modes
(d) All of the above

Student's Assignments

Answer Key

1. (d)	**2.** (d)	**3.** (d)	**4.** (b)	**5.** (b)
6. (c)	**7.** (a)	**8.** (b)	**9.** (c)	**10.** (d)
11. (b)	**12.** (c)	**13.** (d)		

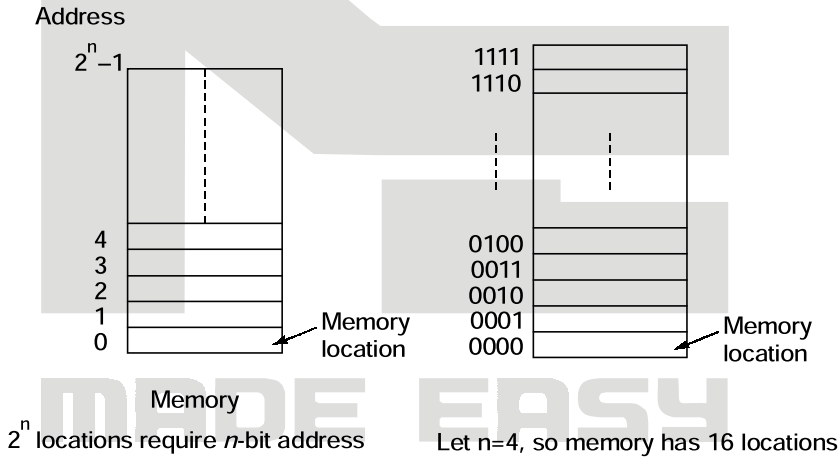
CHAPTER 2

CPU Organization

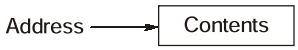
2.1 Introduction

How to address main memory?

Main memory is a set of storage locations. Each location of memory has a unique address (a binary number starting from zero). Each "addressable" location holds a fixed number of bits. Any location can be accessed at high speed in any order (random access memory).



Memory consists of addressable locations. A memory location has 2 components: Address and Contents.



Data transfer between CPU and memory involves address bus and data bus

There are two ways stored information can be organized.

1. **Little Endian (little end first):** First location can hold least significant byte.
   Example: Intel uses little endian

Example - 2.2 Write the two address instructions for the following statement:

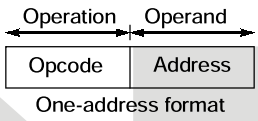
$$X = (A + B) \times (C + D)$$

Solution:

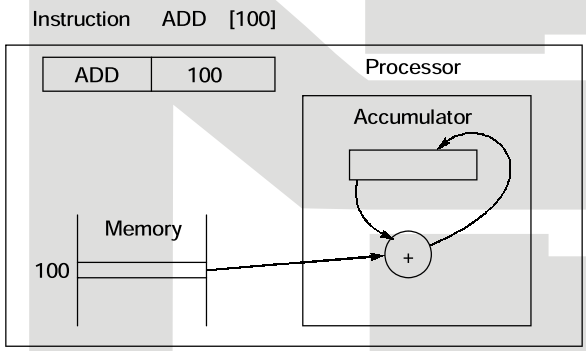
```
MOV R1,  A   ;   R1 ← M[A]
ADD R1,  B   ;   R1 ← R1 + M[B]
MOV R2,  C   ;   R2 ← M[C]
ADD R2,  D   ;   R2 ← R2 + D
MUL R1,  R2  ;   R1 ← R1 × R2
MOV X,   R1  ;   M[X] ← R1
```

2.3.3 One address Instruction

One address instruction has only two fields. One for opcode and other filed for operand.



Example:



Advantages:

- Shorter instruction
- Eliminates two memory accesses
- Faster accessing location inside processor than memory

Disadvantages:

- Only one location for one operand and result
- Still it may need one memory access

Example - 2.3 Write the one address instructions for the following statement:

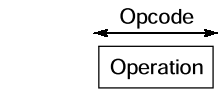
$$X = (A + B) \times (C + D)$$

Solution:

```
LOAD  A  ;   AC ← M[A]
ADD   B  ;   AC ← AC + M[B]
STORE T  ;   M[T] ← AC
LOAD  C  ;   AC ← M[C]
ADD   D  ;   AC ← AC + M[D]
MUL   T  ;   AC ← AC × M[T]
STORE X  ;   M[X] ← AC
```

2.3.4 Zero Address Instruction

Zero address instruction has only opcode filed. It has no address field.

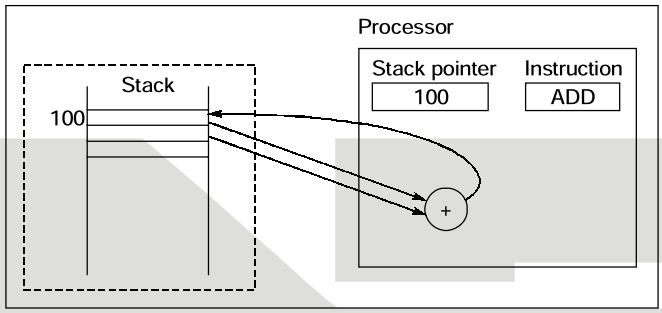


Zero Address Instruction Format

It is possible to eliminate all addresses by using specific locations. Usually the locations for the operands/result are the top two locations of a *stack* (a last-in first-out queue) in memory or implemented with registers within the processor.

Stack pointer is a register within the processor used to hold, the address of the top location. Zero-address format may be useful to compilers for producing code for arithmetic expressions (using reverse Polish notation) with the help of stack.

Example: Instruction: ADD ; it adds top two elements of the stack and stores result at top of stack



***Example* - 2.4** Write the zero address instructions for the following statement.

$$X = (A + B) \times (C + D)$$

Solution:

LOAD	A	;	AC ← M[A]
PUSH	A	;	TOS ← A
PUSH	B	;	TOS ← B
ADD		;	TOS ← (A +B)
PUSH	C	;	TOS ← C
PUSH	D	;	TOS ← D
ADD		;	TOS ← (C + D)
MUL		;	TOS ← (C + D) × (A + B)
POP	X	;	M[X] ← TOS

A stack-organized computer does not use an address field for the instructions ADD and MUL. The PUSH and POP instructions, however, need an address field to specify the operand that communicates with the stack.

2.4 Addressing Modes

The different ways in which the location of an operand is specified in an instruction are referred to as addressing modes. It is the method used to identify the location of an operand.

Addressing: "*The general subject of specifying where the operands are*" is called addressing.

- Addressing mode may present within operation field (opcode) or in a separate ***mode*** field.
- The decoding step in the instruction cycle determines the operation to be performed, the addressing mode of the instruction, and the location of the operands.

2.5 Types of Machine Instructions

Based on operation performed, machine instructions can be divided into the following:

1. Data transfer Instructions
2. Data Manipulation Instructions (Computation): Arithmetic and Logical Instructions.
3. Program Control Instructions.

2.5.1 Data Transfer Instructions

Instructions that transfers data from one location(Register/Memory) to another location(Register/Memory) without changing the data. Data transfer operations supported by many processors are Load, Store, Move, Input, Output, Push, Pop, Exchange, etc.

- **LOAD:** Data transfers from memory to register.
- **STORE:** Data transfers from register to memory.
- **MOVE:** Data transfers from register to register.
- **IN:** Transfers data from input device to register.
- **OUT:** Transfers data from register to output device.
- **PUSH:** Gets data from memory or register on to the top of stack.
- **POP:** Gets data from top of stack to memory or register.
- **XCHG:** Exchanges the data between memory and registers.

2.5.2 Data Manipulation Instructions

- **Arithmetic Instructions:** Performs an arithmetic operations such as addition, subtraction, multiplication division, Increment, Decrement, etc.
  Example: ADD, SUB, MUL, DIV, INC, DEC, etc.
- **Logical Instructions:** Performs bit-wise logical operation such as AND, OR, exclusive-OR, NOT, shift, rotate, etc.
  Example: AND, OR, NOT, XOR, SHL, SHR, ROR, ROL, etc.
- **Arithmetic and Logical Instructions:** Performs operations such as arithmetic shift left, arithmetic shift right, etc.
  Example: SAL, SAR, etc.

2.5.3 Program Control Instructions

1. Compare and Test Instructions

To compare and test status flags we need support of condition code register.

Condition code register(CCR): Condition Code Register has following flag bits:

(*i*) **5 status bits:** To indicate a situation after the arithmetic or logical operation.

Carry borrow (C)
Overflow (V)
Zero (Z)
Negative (N)
Half Carry (H)
} ⇒ Used CPU's branch instructions to make decisions

(*ii*) 2 interrupt masking bits

(*iii*) 1 stop disable bit