Computer Architecture
Instruction

Moniruzzaman

Adjunct Lecturer
North Western University, Khulna-9000

Confession

- Most of the materials have been collected from Internet.
- Images are taken from Internet.
- Various books are used to make these slides.
- Various slides are also used.
- References & credit:
  - Atanu Shome, Assistant Professor, CSE, KU.
  - Computer Organization and Design: the Hardware/Software Interface - Textbook by David A Patterson and John L. Hennessy.
  - Computer Organization and Architecture - Book by William Stallings

Instructions

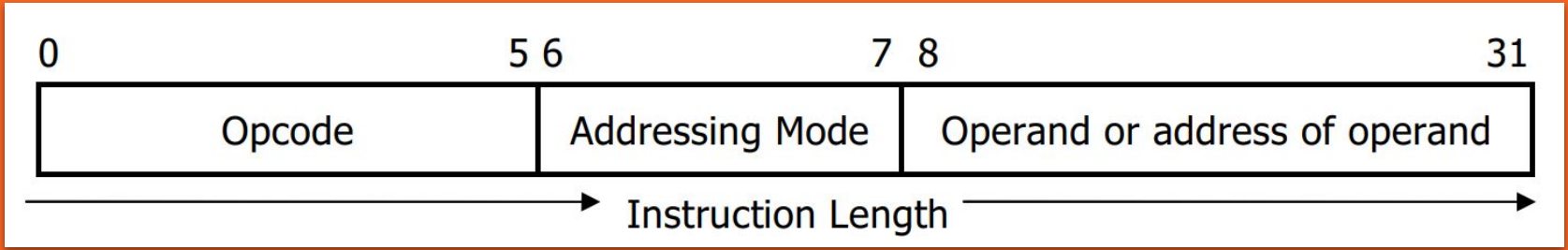
- **Opcode:** It specifies the operation (`ADD`, `SUBTRACT`, `MULTIPLY`, etc.) to be performed.

- **Operands:** An address field of operand on which data processing is to be performed.

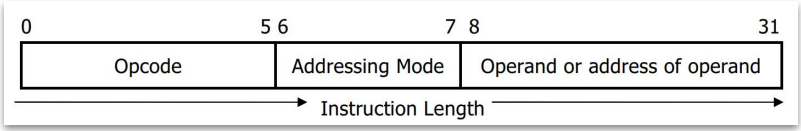
An operand can reside **in the memory** or a **processor register** or can be incorporated within the operand field of instructions as **an immediate constant**.

Therefore a mode field is needed that specifies the way the operand or its address is to be determined.

An Instruction Format of 32 bits



Instruction Length



From the figure we have the following observations :

(i) The size of the opcode is **6 bits**. So we will have 2^6 = 64 operations.

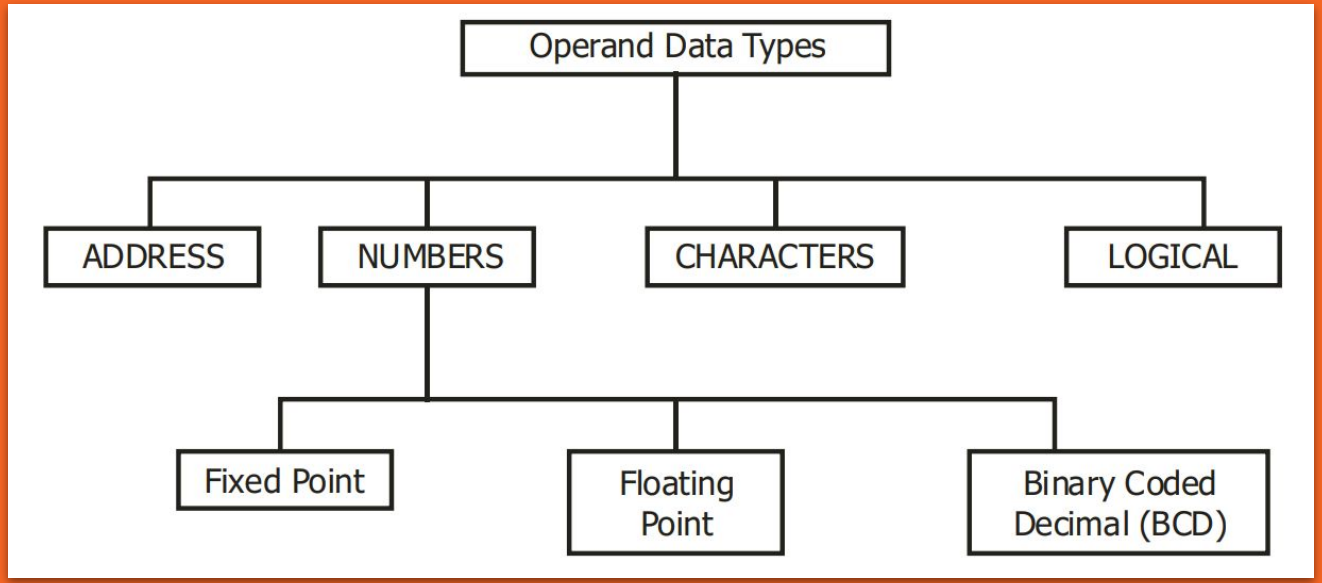
(ii) There is only one operand address machine.

(iii) There are **two bits** for addressing modes. Therefore , there are 2^2 = 4 different addressing modes possible for this machine.

(iv) The last field (8 - 31 bits = 24 bits) here is the operand or the address of operand field.

In case of immediate operand the maximum size of the unsigned operand would be 2^{24}. In case it is an address of operand in memory, then the maximum physical memory size supported by this machine is **2^{24} = 16 Mb**.

Operand Data Types



Types of Instructions

-

Number of Addresses

- 3-address/operand Instructions
- 2-address/operand Instructions
- 1-address/operand Instructions
- 0-address/operand Instructions

Number of addresses (Example)

Instruction		Comment
SUB	Y, A, B	Y ← A − B
MPY	T, D, E	T ← D × E
ADD	T, T, C	T ← T + C
DIV	Y, Y, T	Y ← Y ÷ T

(a) Three-address instructions

Instruction		Comment
MOVE	Y, A	Y ← A
SUB	Y, B	Y ← Y − B
MOVE	T, D	T ← D
MPY	T, E	T ← T × E
ADD	T, C	T ← T + C
DIV	Y, T	Y ← Y ÷ T

(b) Two-address instructions

Instruction		Comment
LOAD	D	AC ← D
MPY	E	AC ← AC × E
ADD	C	AC ← AC + C
STOR	Y	Y ← AC
LOAD	A	AC ← A
SUB	B	AC ← AC − B
DIV	Y	AC ← AC ÷ Y
STOR	Y	Y ← AC

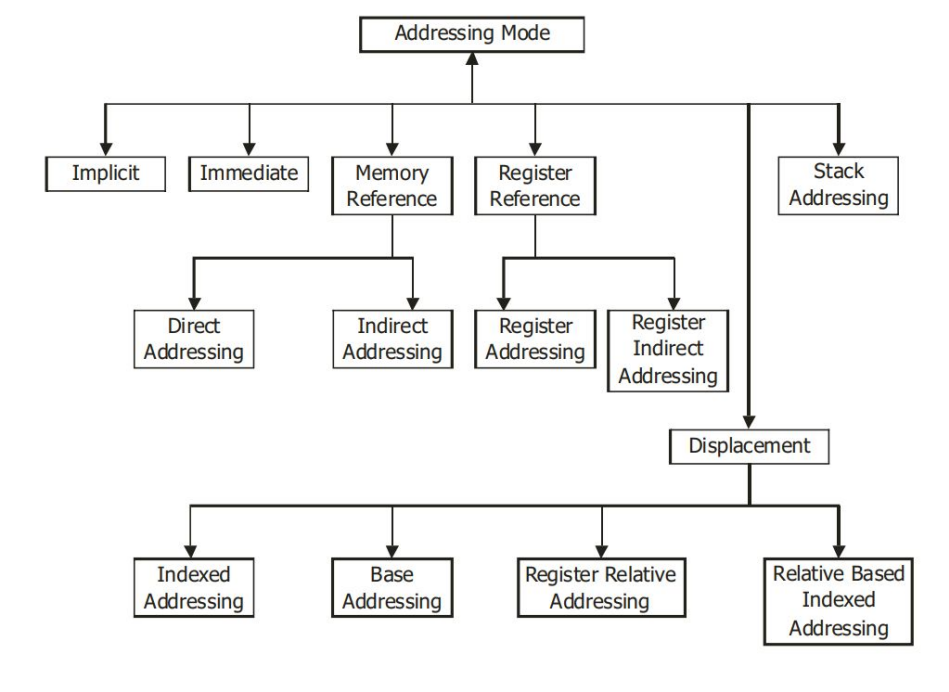
(c) One-address instructions

CMC	Complement carry
STC	Set carry
CLD	Clean direction flag

(d) Zero-address instructions

Figure 12.3 Programs to Execute $Y = \frac{A - B}{C + (D \times E)}$

Addressing Modes



Instruction Length

- Affected by and affects:
  - Memory size
  - Memory organization
  - Memory transfer length
  - Memory transfer

- Trade off between powerful instruction repertoire and saving space.

Allocation of Bits

- Number of addressing modes
- Number of operands
- Register versus memory
- Number of register sets
- Address range
- Address granularity

Reduced Instruction Set Computer (RISC)

Complex Instruction Set Computer (CISC)

Instructions: RISC has fewer instructions, each of which can perform a single operation.

Instruction format: RISC works on a fixed instruction format that keeps less than 100 instructions to be processed.

Emphasis: RISC emphasizes software.

Transistors: RISC uses less transistors for hardware space.

Cycles: RISC reduces cycles per instruction.

Instructions: CISC has a large number of complex instructions that can perform multiple internal operations.

Instruction format: CISC works in a variable instruction format.

Emphasis: CISC emphasizes hardware.

Transistors: CISC uses more transistors for hardware space.

Cycles: CISC processors use a single instruction to do all of the loading, evaluating, and storing operations.

Thank You