Computer System Architecture
COMP201Th
Unit: 3
Programming the Basic Computer

Lecture: 1

Instruction Formats

A computer performs a task based on the instruction provided. Instruction in computers comprises groups called fields. These fields contain different information as for computers everything is in 0 and 1 so each field has different significance based on which a CPU decides what to perform. The most common fields are:

- Operation field specifies the operation to be performed like addition.
- Address field which contains the location of the operand i.e. register or memory location.
- Mode field which specifies how operand is to be founded.

Instruction is of variable length depending upon the number of addresses it contains. Generally, CPU organization is of three types based on the number of address fields:

1. Single Accumulator organization
2. General Register organization
3. Stack organization

In the first organization, the operation is done involving a special register called the accumulator.

In second on multiple registers are used for the computation purpose.

In the third organization the work on stack basis operation due to which it does not contain any address field.

Based on the number of address, instructions are classified as:

1. **Zero Address Instructions:**

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. The following program shows how X = (A + B) ∗ (C + D) will be written for a stack organized computer. (TOS stands for top of stack)

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

To evaluate arithmetic expressions in a stack computer, it is necessary to convert the expression into reverse Polish notation. The name “zero-address” is given to this type of computer because of the absence of an address field in the computational instructions.

2. **One Address Instructions:** This uses an implied ACCUMULATOR register for data manipulation. One operand is in the accumulator and the other is in the register or memory location. Implied means the CPU already known that one operand is in the accumulator so there is no need to specify it. The program to evaluate X=(A+B)*(C+D) is:

LOAD	A	AC ← M [A]
ADD	B	AC ← A [C] + 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

All operations are done between the AC register and a memory operand. T is the address of a temporary memory location required for storing the intermediate result.

3. **Two Address Instructions:**

Two address instructions are the most common in commercial computers. Here again each address field can specify either a processor register or a memory word. The program to evaluate X = (A + B) * (C + D) is as follows:

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

The MOV instruction moves or transfers the operands to and from memory and processor registers. The first symbol listed in an instruction is assumed to be both a source and the destination where the result of the operation is transferred.

4. **Three Address Instructions:** This has three address field to specify a register or a memory location. Program created are much short in size but number of bits per instruction increase. These instructions make creation of program much easier but it does not mean that program will run much faster because now instruction only contain more information but each micro operation (changing content of register, loading address in address bus etc.) will be performed in one cycle only.

The program in assembly language that evaluates X=(A+B)*(C+D) is shown below:

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

It is assumed that the computer has two processor registers, R1 and R2. The symbol M[A] denotes the operand at memory address symbolized by A.