Instruction Format

A presentation by Aryan Singh, 14200121176, Department of Computer Science and Engineering

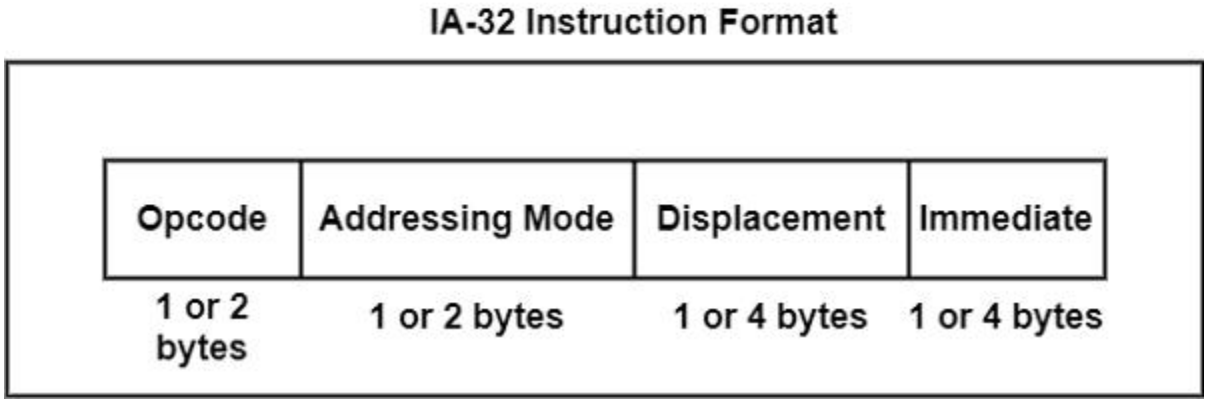
What is Instruction Format: *A brief Idea!*

Instruction includes a set of operation codes and operands that manage with the operation codes. Instruction format supports the design of bits in an instruction. It contains fields including opcode, operands, and addressing mode.

The instruction length is generally preserved in multiples of the character length, which is 8 bits. When the instruction length is permanent, several bits are assigned to opcode, operands, and addressing modes.

The function of allocating bits in the instruction can be interpreted by considering the following elements –

- Number of addressing modes
- Number of operands
- Number of CPU registers
- Number of register sets
- Number of address lines



IA-32 Instruction Format

Opcode	Addressing Mode	Displacement	Immediate
1 or 2 bytes	1 or 2 bytes	1 or 4 bytes	1 or 4 bytes

Diagram of Intel Architecture- 32 bits

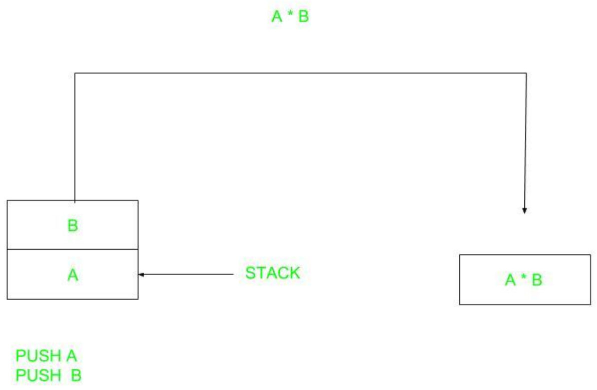
Types of Address Instruction

There are three address instructions :

1. Memory Address for the two operands and one destination need to be specified.
2. It is also called the general register organization.
3. Instruction add `R1, R2, R3`
4. Microoperation: `R1 <- R2 + R3`

Zero Address Instruction

A stack-based computer does not use the address field in the instruction. To evaluate an expression first it is converted to reverse Polish Notation i.e. Postfix Notation.



```
Expression: X = (A+B)*(C+D)

Postfixed : X = AB + CD + *

TOP means top of stack

M[X] is any memory location
```

One Address Instruction

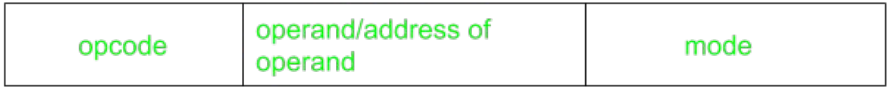
This uses an implied ACCUMULATOR register for data manipulation. One operand is in the accumulator and the other is in the register or memory location. Implied means that the CPU already knows that one operand is in the accumulator so there is no need to specify it.

Expression: X = (A+B)*(C+D)

Postfixed : X = AB + CD + *

TOP means top of stack

M[X] is any memory location



opcode	operand/address of operand	mode

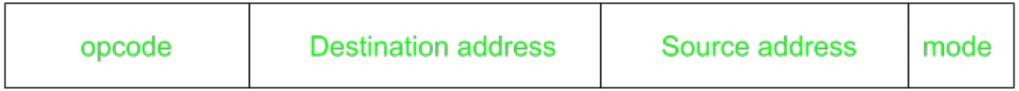
Two Address Instruction

This is common in commercial computers. Here two addresses can be specified in the instruction. Unlike earlier in one address instruction, the result was stored in the accumulator, here the result can be stored at different locations rather than just accumulators, but require more number of bit to represent address.

```
Expression: X = (A+B)*(C+D)

R1, R2 are registers

M[] is any memory location
```



opcode	Destination address	Source address	mode

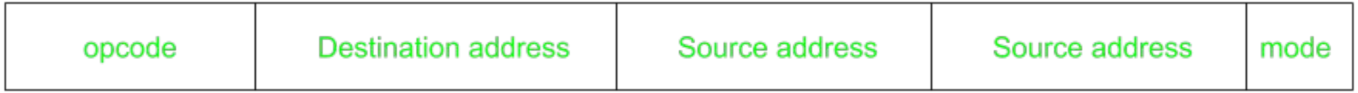
Three Address Instruction

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.

```
Expression: X = (A+B)*(C+D)

R1, R2 are registers

M[] is any memory location
```



opcode	Destination address	Source address	Source address	mode

Well known Addressing Modes: *Just the names!*

1. Implied Mode
2. Immediate Mode
3. Register Mode
4. Register Indirect Mode
5. Direct Address Mode
6. Indirect Address Mode
7. Relative Address Mode
8. Indexed Address Mode
9. Auto-Increment and Auto-Decrement

Thank you!

A presentation by Aryan Singh, 14200121176, Department of Computer Science and Engineering

A special thanks to Geeks For Geeks, and Tutorialspoint. Made using libreoffice.