MCA Part I

Paper-II

Topic: Types of Instruction

Prepared by: Dr. Kiran Pandey

Email-id: kiranpandey.nou@gmail.com

Introduction

Instructions in computer are used to translate high level language programs. There are many ways of classifying the instructions. One of them depends on the number of operands explicitly specified in the instructions. They are zero single double and three operand instructions.

Zero Operand Instructions

The instructions in which operands are not explicitly specified are known as zero-operand or address instructions. The implicit operands are assumed to be in the registers.

Examples

CMC	Complement carry
STC	Set carry
CLD	Clean direction flag

The above instruction operate on implicitly assumed flag register.

One Operand Instructions

The instructions in which a single operand is explicitly specified are known as one operand or address instructions.

Examples

INC A X	AX ← AX + 1

DEC CX	CX ← CX –1 (decrement CX content by 1)
POP BX	POP BX from the stack

The above instructions operate n the single operand specified in the instruction.

Two Operands Instruction

The instruction in which two operands are explicitly specified are known as two operands or address instructions.

Examples

MOV AX, 100	AX ← 100
ADD AX, BX	AX ← AX + BX
SUB CX, 1	CX ← CX –1

Three operands Instructions

The instructions in which three operands are explicitly specified are known as three operands or address instructions.

Examples

IMUL BX, CX, 10	BX ← CX ✶10
IMUL AX, BX, 1024	AX ← BX ✶ 1024

The three operands instructions IMUL is supported by the 80186 processors only.

The processors also supports a wide array of instructions to perform the movement of data (between memory, CPU, I/o devices), arithmetic and logical operations(addition, subtraction, multiplication, division, data, conversion, comparisons), branch (jump instructions), and control of processor operations (enable interrupt, set direction flag), string operations and protection control. The figure -3 below shows classification of instruction.

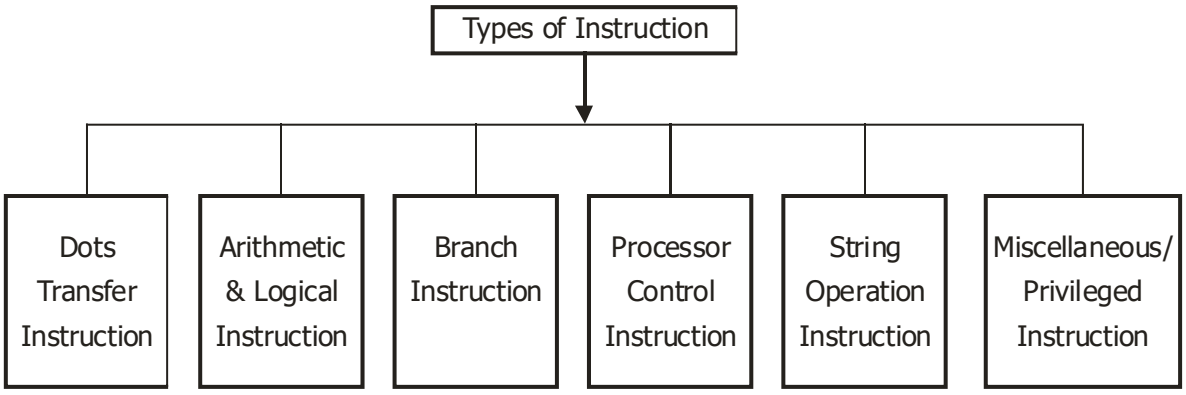

Figure 3 : Types of Instruction

Data transfer Instructions

These instructions transfer data from one location in the computer to another location without changing the data contents.

The most common transfer are between.

- Resister to register.
- Memory to register.
- Register to memory.
- Register to I/O part and vice versa.

These instruction have the following properties.

- Two operands, the source and destination
- They must be of the same data-type that is either of type byte or type word.
- Both the source and destination control refer to memory locations in the same instructions.
- The source can be register or a memory location or an immediate data.
- The destination can be register or a memory location.
- These instructions do not effect the CPU flags.

The instruction also need the mode of addressing for each operand. A table is given below, which lists some data transfer instructions with their mnemonic symbols. Different computers may use different mnemonics for the same instruction.

Operation Name	Mnemonic	Description
MOVE	MOV	Transfer data from Register/Memory/ Immediate to Register/Memory.
XCHG	XCHG	Exchange the contents of the source and destination.
PUSH	PUSH	Push register or memory to stack
POP	POP	Transfer data from top of stack to processor register
LOAD	LD	Loads the contents from memory to register
Set	SET	Specified operands replaced by 1.
Clear	CLEAR	Specified operand replaced by 0's.

Arithmetic and Logic Instructions

These instructions perform arithmetic and logical operations on data.

Arithmetic : The basic arithmetic operands are ADD, SUBTRACT, MULTIPLY and DIVIDE.

Following points are to be noted about arithmetic operations:

- Instruction generally involves two operands that is source and destination.
- They must be of same data type.
- Instructions affect the CPU flag to reflect the status of operations.

Some arithmetic operation their mnemonics and description are given below in the table.

Arithmetic operations	Mnemonics	Description
Addition	ADD	Adds two operands and stores the result.
Subtraction	SUB	Subtracts source from destination.
Decrement	DEC	Subtract one from destination.
Compare	CMP	Compare source with destination and flags are set to indicate carry.
Multiply	MUL	Multiplies source to destination.
Divide	DIV	Divide accumulator by unsigned value stored in reg/men.

Logical Instruction: AND, OR, NOT, XOR are same logical instructions that operates on binary data stored in register. Logical shift (Left shift or Right shift) is also used for transfer of bits either to the left or to the right.

Example:

(i) AND AX, FOOOH, if AX = FFOOH
then AX = FFOO & FOOO
= FOOOH

(ii) R_1 = 1101 1001
R_2 = 0010 1110
R x or R_2 = 1111 0111

Shift Operation : It is basically of three types :

(i) Logical shift : Insets zero to the end of bit position and the other bits of a word are shifted left or right respectively.

(ii) Arithmetic shift: On the arithmetic shift right, the sign bit is replicated into the bit position to its right. On an arithmetic shift left, a logical shift left is performed on all bits but the sign bit, which is retained.

(iii) Circular shift: Rotate left or rotate right. Bits are shifted out at one end of the word are not lost as in a logical shift but are calculated back into the other end.

Branch Instructions

The branch instructions transfers control from the normal sequence of instruction execution to the specified destination or target instruction. These instructions are broadly categorized as,

(i) Conditional branch and

(ii) Unconditional branch.

The conditional jump instructions transfer control to the target location if some specified condition is met or satisfied. some conditional branch jump instructions are JA, JB, JE etc.

Example:

Branch Operation	Mnemonic	Description
Jump	JMP	Unconditional transfer of control to the table.
Jump on Above	JA	Jump if (CF = 0 and ZF = 0) after unsigned math.
Looping	Loop	Loop if CX ≠ zero

Note : CF = carry flog; ZF = Zero flag and CX = Register.

Processor control instructions

The processor supports a variety of instructions modifying the CPU status flag register called as the processor control instructions. Almost all the instructions except those in the data transfer category uses the status of the flags in the flag register during the execution. Generally the flag register is set to reflect the status of the result automatically. Most of the processor control instructions affect the CPU flags.

Examples:

Control Operation	Mnemonic	Description
Clear carry	CLC	Clears the carry flag
Complement carry	CMC	Complements the carry flag
Set carry	STC	Set the carry flog to 1.

String Operations Instructions

A string is a contiguous sequence of types or words. strings can be used to hold any type of data or information that will fit into bytes. There are number of operations that can be performed with strings.

All string instructions require two operands. All instructions assume that the same source operand is in the data segment (DS) and the destination is in the extra segment (ES). In string instructions, the operands are not mentioned explicitly. Therefore, all the registers used by the instructions must be loaded prior to the execution of the instruction.

Example:

Shang Operation	Mnemonics	Description
Clear direction flag	CLD	Clear direction flag so that pointers in shing instructions are updated by incrementing.
Move string	MOVS	Transfers a byte or a word from the source string to the destination string.

The **SKIP** instruction is a zero-address instruction and skips the next instruction to be executed in sequence. In other words, it increments the value of PC by one instruction length. The **SKIP** can also be conditional. For example, the instruction **ISZ** skips the next instruction only if the result of the most recent operation is zero.

CALL and **RETN** are use for CALLing subprograms and RETurning from them. Assume that a memory stack has been built such that stack pointer points to a non-empty location stack and expand towards zero address.

Miscellaneous and Privileged Instructions: These instructions do not fit in any of the above categories. I/O instructions: start I/O, stop I/O, and test I/O. typically, I/O destination is specified as an address. Interrupt and state-swapping operations: There are two kinds of exceptions, interrupts that are generated by hardware and traps, which are generated by programs. Upon receiving interrupts, the state of current processes will be saved so that they can be restarted after the interrupts has been taken care of. Most computer instructions are divided into two categories, privileged and non-privileged. A process running in privileged mode can execute all instructions from the instruction set while a process running in user mode can only execute a sub-set of the instructions. I/O instructions are one example of privileged instruction, clock interrupts are another one.