

Directive	Purpose	Storage Space
DB	Define Byte	allocates 1 byte
DW	Define Word	allocates 2 bytes
DD	Define Doubleword	allocates 4 bytes
DQ	Define Quadword	allocates 8 bytes
DT	Define Ten Bytes	allocates 10 bytes

choice DB 'y'  
num DW 12345  
neg\_num DW -12345  
big\_num DQ 123456789  
real\_value1 DD 1.234  
real\_value2 DQ 123.456

Directive	Purpose
RESB	Reserve a Byte
RESW	Reserve a Word
RESD	Reserve a Doubleword
RESQ	Reserve a Quadword
REST	Reserve a Ten Bytes

The TIMES directive directive allows multiple multiple initializations initializations to the same value

marks TIMES 9 DW 0

The %assign directive directive can be used to define numeric constants like the EQU directive.

This directive allows redefinition.

%assign TOTAL 10

The EQU directive is used for defining constants.

The syntax of the EQU directive

CONSTANT\_NAME EQU expression

TOTAL\_STUDENTS equ 50

The %define directive allows defining defining both numeric and string constants

%define PTR [EBP+4]

This directive also allows redefinition and it is case-sensitive.

In this addressing addressing mode, a register register contains contains the operand, operand

MOV DX, AGE ; Register in first operand

MOV COUNT, CX ; Register in second operand

MOV EAX, EBX ; Both the operands are in registers

- An immediate operand has a constant value or an expression..

- When an instruction with two operands uses immediate addressing

- the first operand may be a register or memory location.

- Second operand is an immediate constant.

- The first operand defines the length of the data.

BYTE\_VALUE DB 120 ; A byte value is defined

WORD\_VALUE DW 200 ; A word value is defined

ADD BYTE\_VALUE, 49 ; An immediate operand 55 is added

MOV AX, 45H ; Immediate constant 45H is transferred to AX

Memory addressing mode

- MOV instruction that is used for moving data from one storage space to another.

- The MOV instruction takes two operands.

MOV destination, source

The value of source operand remains unchanged

Both the operands in MOV operation should be of same size

MOV register, register

MOV register, immediate

MOV memory, immediate

MOV register, memory

MOV memory, register

BYTE 1 Byte

WORD 2 Bytes

DWORD 4 Bytes

QWORD 8 Bytes

TBYTE 10 Bytes

The INC instruction is used for incrementing an operand by one

INC Instruction

The DEC instruction is used for decrementing an operand by one. It works on a single operand that can be either in a register or in memory

DEC Instruction

The ADD and SUB instructions are used for performing simple addition/subtraction operation

- Register to register

- Memory to register

- Register to memory

- Register to constant data

- Memory to constant data

- MUL (Multiply) instruction handles unsigned data

- IMUL (Integer Multiply) handles signed data.

- Both instructions affect the Carry and Overflow flag.

- DIV (Divide) instruction is used for unsigned data

- IDIV (Integer Divide) is used for signed data

DIV/IDIV Instructions

The dividend is in an accumulator.

Sr.No.	Instruction	Format
1	AND	AND operand1, operand2
2	OR	OR operand1, operand2
3	XOR	XOR operand1, operand2
4	TEST	TEST operand1, operand2
5	NOT	NOT operand1

The OR instruction is used for supporting logical expression by performing bitwise OR operation

The XOR instruction implements the bitwise XOR operation

- The NOT instruction implements the bitwise NOT operation.

- Reverses the bits in an operand.

- The operand could be either in a register or in the memory.

## Assembly Class 2

### Variables

#### Allocating Storage Space for Initialized Data

#### Allocating Storage Space for Uninitialized Data

#### Multiple initializations

### Constants

#### %assign

#### EQU

#### %define

### Addressing Modes

#### Register addressing

#### Immediate addressing

#### Memory addressing

#### The MOV Instruction

### Type Specifiers

### Assembly - Conditions

#### Unconditional jump

#### JMP instruction

Instruction	Description	Flags tested
JE/JZ	Jump Equal or Jump Zero	ZF
JNE/JNZ	Jump not Equal or Jump Not Zero	ZF
JG/JNLE	Jump Greater or Jump Not Less/Equal	OF, SF, ZF
JGE/JNL	Jump Greater/Equal or Jump Not Less	OF, SF
JL/JNGE	Jump Less or Jump Not Greater/Equal	OF, SF
JLE/JNG	Jump Less/Equal or Jump Not Greater	OF, SF, ZF

Instruction	Description	Flags tested
JE/JZ	Jump Equal or Jump Zero	ZF
JNE/JNZ	Jump not Equal or Jump Not Zero	ZF
JA/JNBE	Jump Above or Jump Not Below/Equal	CF, ZF
JAЕ/JNB	Jump Above/Equal or Jump Not Below	CF
JB/JNAE	Jump Below or Jump Not Above/Equal	CF
JBE/JNA	Jump Below/Equal or Jump Not Above	AF, CF

Instruction	Description	Flags tested
JXCZ	Jump if CX is Zero	none
JC	Jump If Carry	CF
JNC	Jump If No Carry	CF
JO	Jump If Overflow	OF
JNO	Jump If No Overflow	OF
JP/JPE	Jump Parity or Jump Parity Even	PF
JNP/JPO	Jump No Parity or Jump Parity Odd	PF
JS	Jump Sign (negative value)	SF
JNS	Jump No Sign (positive value)	SF

#### Conditional jump

#### CMP instruction

CMP destination, source

### Loops

- The JMP instruction instruction can be used for implementing implementing loops

MOV CL, 10

L1:

<LOOP-BODY>

### Logical Instructions

#### AND Instruction

#### OR Instruction

#### XOR Instruction

#### NOT Instruction