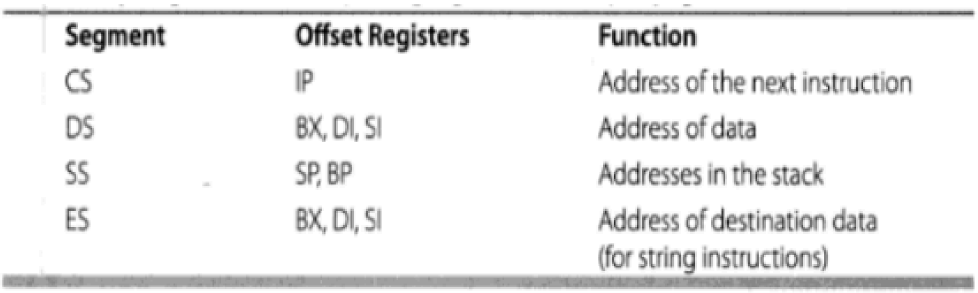
Lecture:4 - Addressing Modes

The different ways in which a microprocessor can access data from memory locations are referred to as its addressing modes.

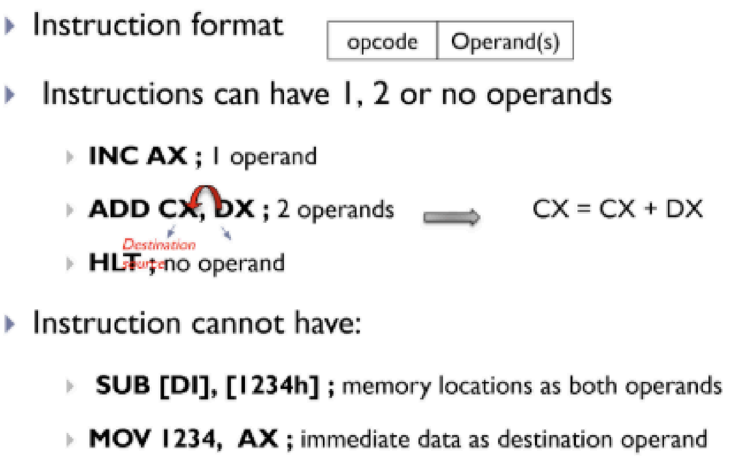
MAP EVERYTHING/CALCULATE [SEGMENT:OFFSET] OF ANY INSTRUCTION FOLLOWING THIS TABLE:



Segment	Offset Registers	Function
CS	IP	Address of the next instruction
DS	BX, DI, SI	Address of data
SS	SP, BP	Addresses in the stack
ES	BX, DI, SI	Address of destination data (for string instructions)

Addressing modes and categories are mainly of 5 types:

1) Addressing Data
2) Addressing Program codes in memory
3) Addressing Stack in memory
4) Addressing I/O
5) Implied addressing

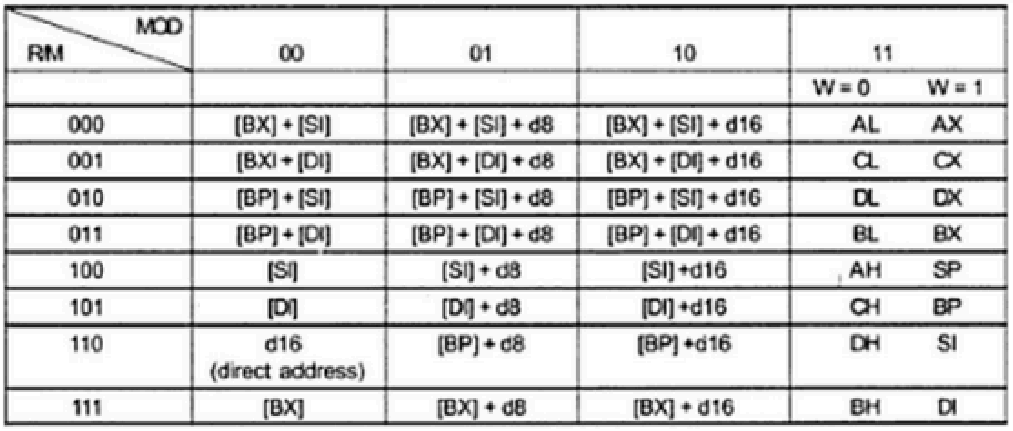


- Instruction format

opcode	Operand(s)

- Instructions can have 1, 2 or no operands
  - **INC AX ;** 1 operand
  - **ADD CX, DX ;** 2 operands ⟹ CX = CX + DX

    Destination
    source
  - **HLT ;** no operand
- Instruction cannot have:
  - **SUB [DI], [1234h] ;** memory locations as both operands
  - **MOV 1234, AX ;** immediate data as destination operand



MOD RM	00	01	10	11	
				W = 0	W = 1
000	[BX] + [SI]	[BX] + [SI] + d8	[BX] + [SI] + d16	AL	AX
001	[BXI + [DI]	[BX] + [DI] + d8	[BX] + [DI] + d16	CL	CX
010	[BP] + [SI]	[BP] + [SI] + d8	[BP] + [SI] + d16	DL	DX
011	[BP] + [DI]	[BP] + [DI] + d8	[BP] + [DI] + d16	BL	BX
100	[SI]	[SI] + d8	[SI] +d16	AH	SP
101	[DI]	[DI] + d8	[DI] +d16	CH	BP
110	d16 (direct address)	[BP] + d8	[BP] +d16	DH	SI
111	[BX]	[BX] + d8	[BX] + d16	BH	DI

1) Addressing Data

To calculate the Physical address of these instructions:

DS : BX/ DI/ SI [Following the table]

PA = DS * **10h + offset;** // here the offsets will be either **BX/DI/SI**

Addressing data again has 7 sub classifications:

1. Immediate addressing
2. Direct addressing
3. Register [direct] addressing
4. Register indirect addressing
5. Base-plus-index addressing
6. Register relative addressing
7. Base-relative-plus-index addressing

Examples of each:

1.Immediate addressing:

MOV AX, **12h**; // a direct hex value in source

2.Direct addressing:

MOV BX, [1234h]; // An offset is given in source/destination and other one is a register

3.Register [direct] addressing:

MOV AX,BX; // both are registers

4.Register indirect addressing:
MOV CX, [BX]; // Offset value is provided in a register in **BX/DI/SI** either is source or destination

#Exception:
MOV CX, [BP]; // if BP is in the offset then it will be considered as register relative addressing since a displacement of 00h is automatically added after it like [BP+00h] and to find out the Physical address of in the memory we will add the BP offset with SS segment's value instead of DS. => **SS * 10h +[BP]#**

5. Base-plus-index addressing:
MOV DX, [BX+DI]; // Base and Index **[BX/DI/SI]** are given in offset part either is source or destination

6.Register relative addressing:
MOV AX, [BX+1000h] // A displacement is added with the offset either in the source or destination

7.Base-relative-plus-index addressing:
MOV AX, [BX+DI+10h] // A base, index and displacement will be in the offset either in the source or destination

Example:

Given, BX = 0300H, SI = 0200H, DS = 1000H, ARRAY = 1000H

Instruction	Addressing mode	Source	Destination
MOV AX, BX	Register Direct Addressing	BX = 0300H	AX
MOV AH, 3AH	Immediate Addressing	3AH	AH
MOV [1234H], AX	Direct Addressing	AX	11234H
MOV [BX], CL	Register Indirect Addressing	CL	10300H
MOV [BX+SI],DL	Base-plus-index Addressing	DL	10500H

MOV CL, [BX + 4H]	Register Relative Addressing	10304H PA = 1000 * 10H + (0300 +4) = 10304H	CL
MOV ARRAY[BX +SI], DX	Base-relative-plus-index Addressing	DX	11500H DS*10H + (0300+0200+ 1000) = 11500H

2) Addressing Program Codes

Used with **JMP or **CALL** instructions usually.

Addressing codes again has 3 sub-classifications:

i) Direct
ii) Indirect
iii) Relative

To calculate the Physical address of these instructions:

CS : IP [Following the table]

PA = CS * 10h + IP;

// here the offset/ IP will vary:

- If IP is given in BX/DI/SI registers **as offset** then first we have to find out DS*10h + [BX/DI/SI] (this is the physical address where IP is located) first then add this IP with CS*10h
- If IP is given in BP register as offset the first we have to find out SS*10h + [BP] (this is the physical address where IP is located) then add this with CS*10h

[Check examples for better understanding]

1.Direct:

JMP BX; // any general purpose register can be here AX/BX/CX/DX or SP/BP/DI/SI, **here, AX/BX/CX/DX/SP/BP/DI/SI**

2. Indirect:

JMP [BX]; // **[BP],[BX],[DI],[SI]**

3. Relative:

JMP [BX + 100h]; // any relative register with displacement

Examples:

Address	10600h	10601h	20600h	200ABh	200ACh	33412h	39A87h	39A88h
Data	12h	34h	56h	87h	9Ah	B0	FEh	20h

A portion of a memory is given above, where there are memory locations and their corresponding instruction bytes or data. Here, DS = 2000h, SS = 1000h, CS = 3000h, BP = 0400h, SI = 0200h, BX = 00AB.

a) Find out what will be the physical address of the memory for **CALL [BX]** instruction. Also mention which data we will get from this location from the given table.

b) Find out what will be the physical address of the memory for **JMP[BP + SI]** instruction. Also mention which data we will get from this location from the given table.

Solution:

a)

As **Call [BX]** is an **addressing codes in program [Indirect]** type addressing mode.

The physical address will be = CS * 10h + IP

Now, the value of IP is in the given table at the physical address = (DS *10 +BX)
= (2000*10 + 00AB)
= 200ABh

Therefore, the IP is at 200ABh of the table, IP = 9A87h

So, the physical address of CALL [BX] is = CS *10h + IP
= 3000h * 10h + 9A87h
= 39A87h;

The data that we will get from this location/ after executing the instruction is = FEh.

b)

As **JMP[BP + SI]** is an **addressing codes in program [Relative]** type addressing mode.

The physical address will be = CS * 10h + IP

Now, the value of IP is in the given table at the physical address = (SS *10 +[BP+SI])
= (1000*10 + 0400 + 0200)
= 10600h

Therefore, the IP is at 200ABh of the table, IP = 3412h

So, the physical address of JMP[BP + SI] is = CS *10h + IP
= 3000h * 10h + 3412h
= 33412h;

The data that we will get from this location/ after executing the instruction is = B0h.

3) Addressing Stack

**Push/ POP/ CALL instructions are usually associated with it.

PUSH AX;

POP CX ;

CALL SUM; // SUM is a procedure name

To calculate the Physical address **of Stack for these instructions**:

SS : SP [Following the table]

PA = SS * 10h + SP;

4. Addressing Input and Output Port

*IN and OUT instructions are used to address I/O ports

i) direct addressing

```
IN AL, 05h ; Here 05h is a input port number
```

ii) Indirect addressing

```
OUT DX,  AL ; DX contains the address of I/O port
```

*Only the DX register can be used to point at an I/O port otherwise not.

5. Implied Addressing

*No explicit address is given with the instruction
*Implied within the instruction itself
Examples:

```
CLC  ; clear carry flag
HLT   ; halts the program
RET   ; return
```