8086 Microprocessor Programming & Instruction Set(Chapter 3)

Adane Tadesse

Faculty of Electrical and Computer Engineering

Computer and control Stream

Jimma University

Jimma, Oromia, Ethiopia

email: addmomcomp@gmail.com

January 30, 2020

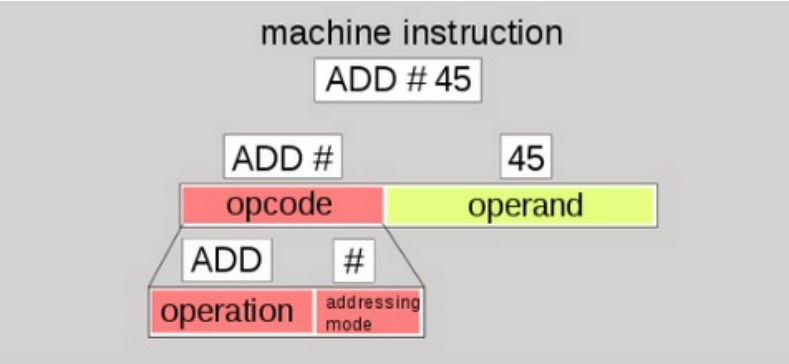
Outline

Machine Language Instruction format

- A machine language instruction format has one or more number of fields associated with it.
- The first field is called as operation code field or **opcode** field
  - Which indicates the type of the operation to be performed by the CPU
- The instruction format also contains other fields known as o**perand** fields
  - it specifies the data or the memory location of the data on which the operation will be performed. **I, R, M** types

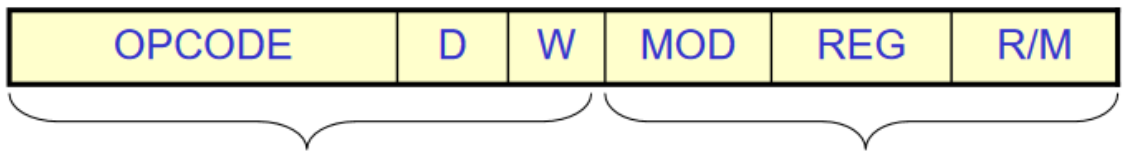
cont..

- The CPU executes the instruction using the information which reside in these fields.



- An instruction can be coded with 1 to 6 bytes
- The general instruction formats of 8086 microprocessors are described as follows:

- **One byte Instruction**:
  - This format is only one byte long and may have the implied data or register operands.
  - The least significant 3-bits of the opcode are used for specifying the register operand
  - all the 8-bits form an opcode and the operands are implied
- **Two byte Instruction**:
  - This format is 2 bytes long.
  - The first byte of the code specifies the operation code and width of the operand specified by w bit.



- **Opcode** field (6 bits) specifies the operation such as add, subtract, or move
- Register Direction Bit (**D** bit)
  - Tells the register operand in REG field in byte 2 is source or destination operand
    - 1: Data flow to the REG field from R/M
    - 0: Data flow from the REG field to the R/M
- Data Size Bit (**W** bit)
  - Specifies whether the operation will be performed on 8-bit or 16-bit data
    - 0: 8 bits
    - 1: 16 bits

- **REG** field is used to identify the register for the first operand

REG	W = 0	W = 1
000	AL	AX
001	CL	CX
010	DL	DX
011	BL	BX
100	AH	SP
101	CH	BP
110	DH	SI
111	BH	DI

- 2-bit **MOD** field and 3-bit **R/M** field together specify the second operand

CODE	EXPLANATION
00	Memory Mode, no displacement follows*
01	Memory Mode, 8-bit displacement follows
10	Memory Mode, 16-bit displacement follows
11	Register Mode (no displacement)

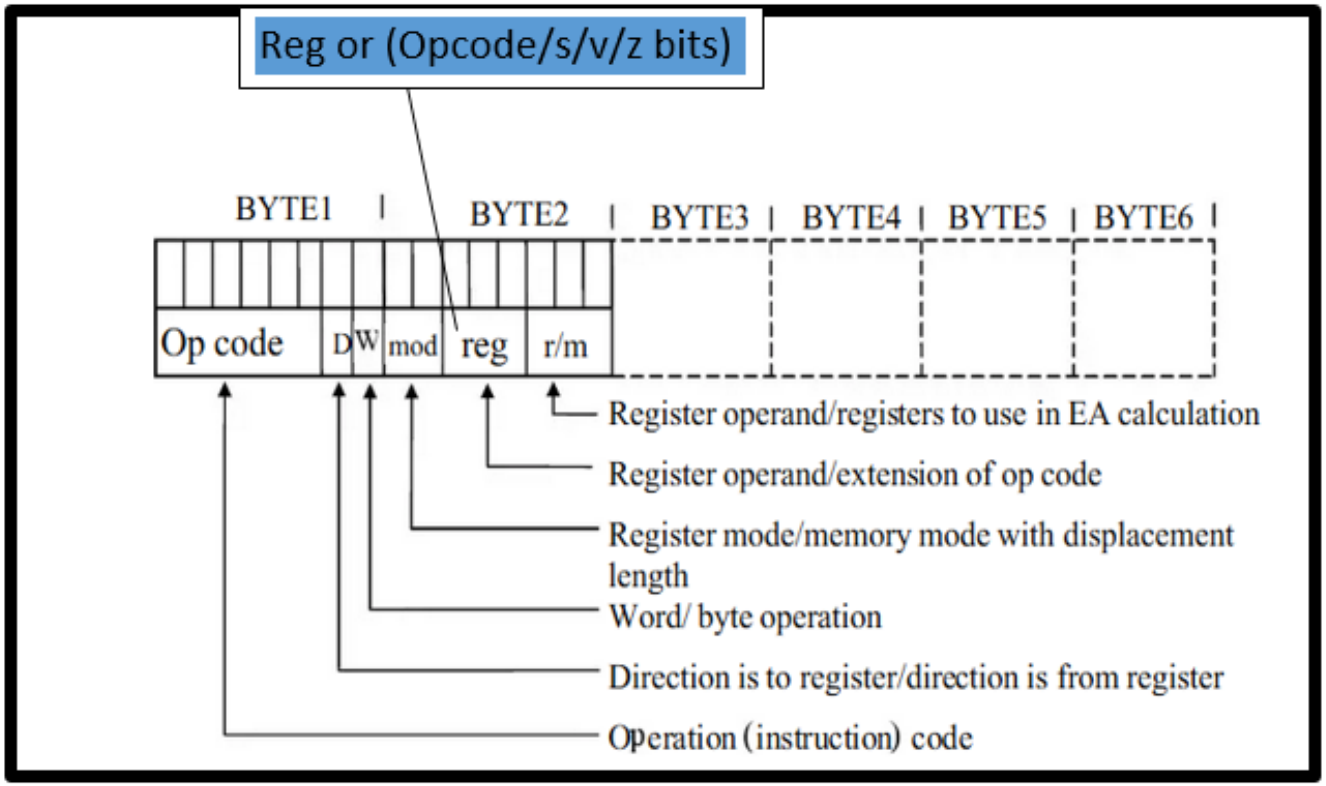
*Except when R/M = 110, then 16-bit displacement follows

(a)

MOD = 11			EFFECTIVE ADDRESS CALCULATION			
R/M	W = 0	W = 1	R/M	MOD = 00	MOD = 01	MOD = 10
000	AL	AX	000	(BX) + (SI)	(BX) + (SI) + D8	(BX) + (SI) + D16
001	CL	CX	001	(BX) + (DI)	(BX) + (DI) + D8	(BX) + (DI) + D16
010	DL	DX	010	(BP) + (SI)	(BP) + (SI) + D8	(BP) + (SI) + D16
011	BL	BX	011	(BP) + (DI)	(BP) + (DI) + D8	(BP) + (DI) + D16
100	AH	SP	100	(SI)	(SI) + D8	(SI) + D16
101	CH	BP	101	(DI)	(DI) + D8	(DI) + D16
110	DH	SI	110	DIRECT ADDRESS	(BP) + D8	(BP) + D16
111	BH	DI	111	(BX)	(BX) + D8	(BX) + D16

(b)

- General instruction format



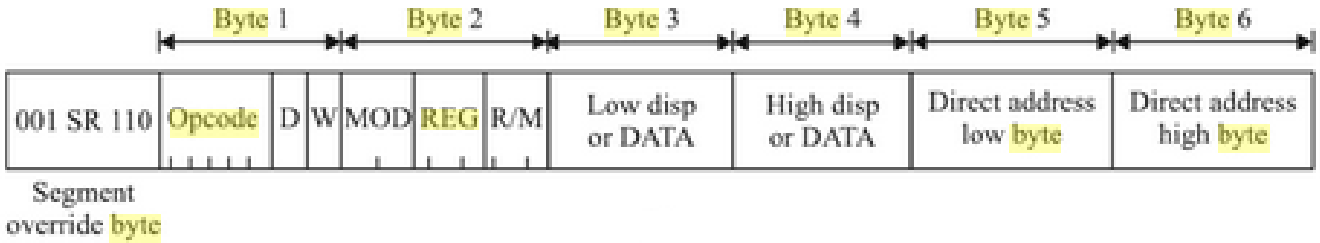
General Instruction format except Segment override prefix

- In some case in 2nd byte have MOD, **OPCODE** and R/ M and for some of the cases have MOD, **REG** and R/M.
- First, OPCODE bits in 2nd byte of instruction format. This field is 3 bit wide. Under that we have three single bit fields **S**, **V** and **z**.
- **S bit:**This is referred to as sign extension.
  - 8-bit operation with 8-bit immediate operand is indicated by W=0 and S=0
  - 16-bit operation with 16-bit immediate operand is indicated by W=1 and S=0
  - 16-bit operation with 8-bit immediate operand is indicated by W=1 and S=1

- **V bit:** Used by shift and rotate to determine single and variable bit shifts and rotate.
  - V = 0 shift/rotate count is one
  - V = 1 shift/rotate count is specified in CL register
- **Z bit:** This bit is used as a compare bit with zero flag in conditional repeat (REP) and loop instructions.
  - Z = 0 repeat/loop while zero flag is clear
  - Z = 1 repeat/loop while zero flag is set

- **segment override prefix byte (SOP byte)** to start with.

  The SOP byte is 001 **SR** 110.



- Where SR value is provided as per table shown below.

xx	Segment register
00	ES
01	CS
10	SS
11	DS

Examples:

- Construct the machine code for the following instructions:
  - (a) MOV BL,CH (code for MOV instruction is 100010)
  - (b) ADD AX,64H (code for ADD instruction is 000000)
  - (c) SUB BX,(DI) (code for SUB instruction is 001010)
  - (d) MOV 1234[SI],AX
  - (e) MOV DS:43[BP],CX

8086 Addressing Modes

- Addressing mode indicates a way of locating data or operands.
- Depending upon the data types used in the instruction and the memory addressing modes, any instruction may belong to one or more addressing modes, or some instruction may not belong to any of the addressing modes.
- it tells us what type of the operand and the way they are accessed from the memory for execution of an instruction and how to fetch particular instruction from the memory.

- According to the flow of instruction execution, the instructions may be categorized as:
  - Sequential control flow instructions: instructions which after execution, transfer control to the next instruction appearing immediately after it (in the sequence) in the program.
    - For example, the arithmetic, logical, data transfer and processor control instructions are sequential control flow instructions.
  - Control transfer instructions: transfer control to some predefined address or the address somehow specified in the instruction, after their execution.
    - For example, INT, CALL , RET and JUMP instructions fall under this category.

addressing modes for sequential control transfer instructions

- **Immediate**: immediate data is a part of instruction, and appears in the form of successive byte or bytes.
  - when the instruction is assembled the operand comes immediately after the opcode
  - This addressing mode execute quickly because of the above point.
    - A) MOV AL,23Fh B) ADD AX,0E23Fh
  - This addressing mode is load information to any register except segment and flag registers. **EA==IP**

- **Register Addressing Mode**: The data is stored in a register and it is referred using the particular register.
  - Memory is not accessed when this addressing mode is executed
  - It is relatively fast
    - MOV BX,AX
    - ADD AL,BL
  - The source must be equal or less size to the destination
- **Direct Addressing Mode**: In this mode, address of the operand is directly specified in the instruction. Here only the offset address is specified, the segment being indicated by the instruction.

- In the direct addressing mode the data is in some memory location and the address of the data in memory come immediately after the instruction.
- This address is the offset address and one can calculate the physical address by shifting left the DS register and adding it to the offset.
  - **MOV AX, [5000H] ; PA=10H*DS+5000H.**
  - Find the physical address of the memory location and its contents after the execution of the following, assuming that DS = 1512H.
    - MOV AL,99H

      MOV [3518],AL ;**PA=10H*DS+3518H=18638H which contain 99H**

- **Register indirect addressing mode** : The address of the memory location which contains data or operand is determined in an indirect way, using the offset registers.
  - The address of the memory location where the operand resides is held by a register.
  - The registers used for this purpose are SI, DI, and BX.
  - If these three registers are used as pointers, that is, if they hold the offset of the memory location.
  - They must be combined with DS or ES in order to generate the 20-bit Effective address(EA).
  - MOV AL,[BX]; **PA=10H*DS+[BX]**

Instruction Set of 8086

Dos and Donts While Using Instructions