

# Construction of 8086 Machine Code

## **Course Teacher:**

**Md. Obaidur Rahman, Ph.D.**

Professor

Department of Computer Science and Engineering (CSE)  
Dhaka University of Engineering & Technology (DUET), Gazipur.

**Course ID:** CSE - 4503

**Course Title:** Microprocessors and Assembly Language  
Department of Computer Science and Engineering (CSE),  
Islamic University of Technology (IUT), Gazipur.

# Lecture References:

---

- ▶ **Book:**

- ▶ *Microprocessors and Interfacing: Programming and Hardware, Chapter # 3, **Author:** Douglas V. Hall*

- ▶ **Lecture Materials:**

# Construction of 8086 Machine Codes

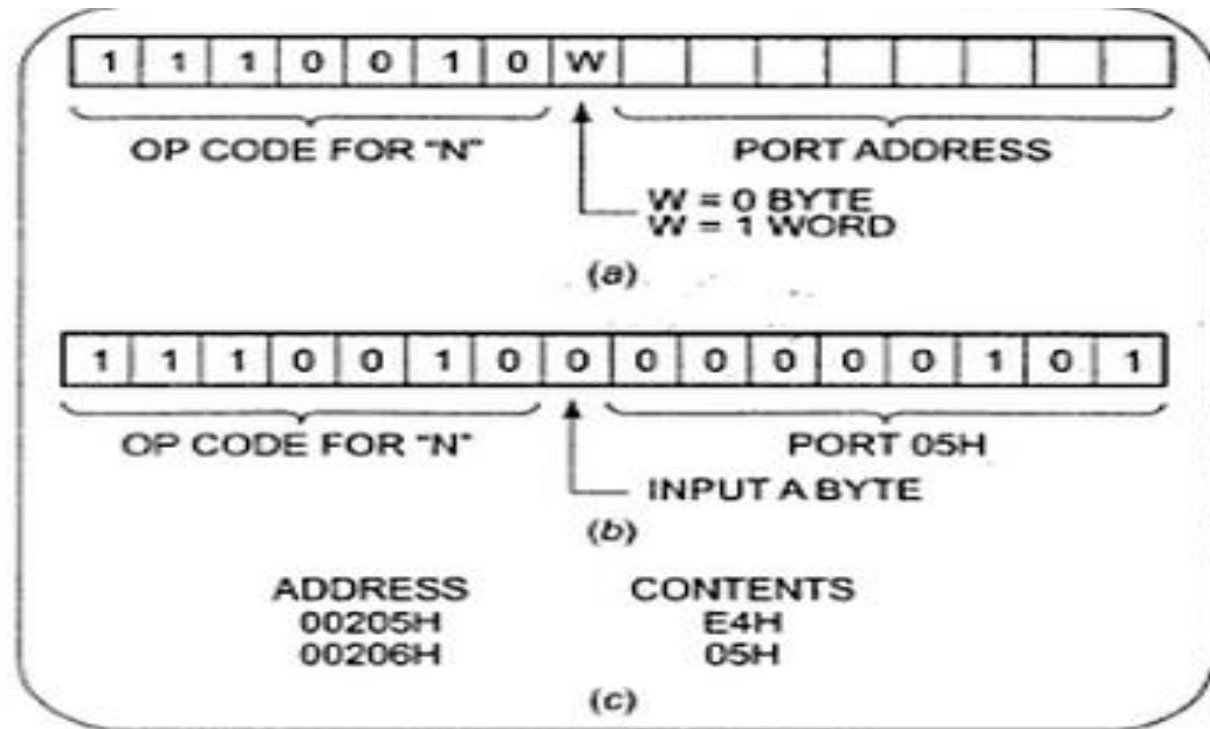
---

- ▶ 8086 has 117 instructions in its instruction set.
- ▶ Each instruction in 8086 is associated with a **binary code**.
- ▶ Most of the time this work will be done by assembler.
- ▶ The things needed to keep in mind is:
  - ▶ ***Instruction templates*** and ***coding formats***
  - ▶ **MOD** and **R/M** Bit patterns for particular instruction

# Instruction Template

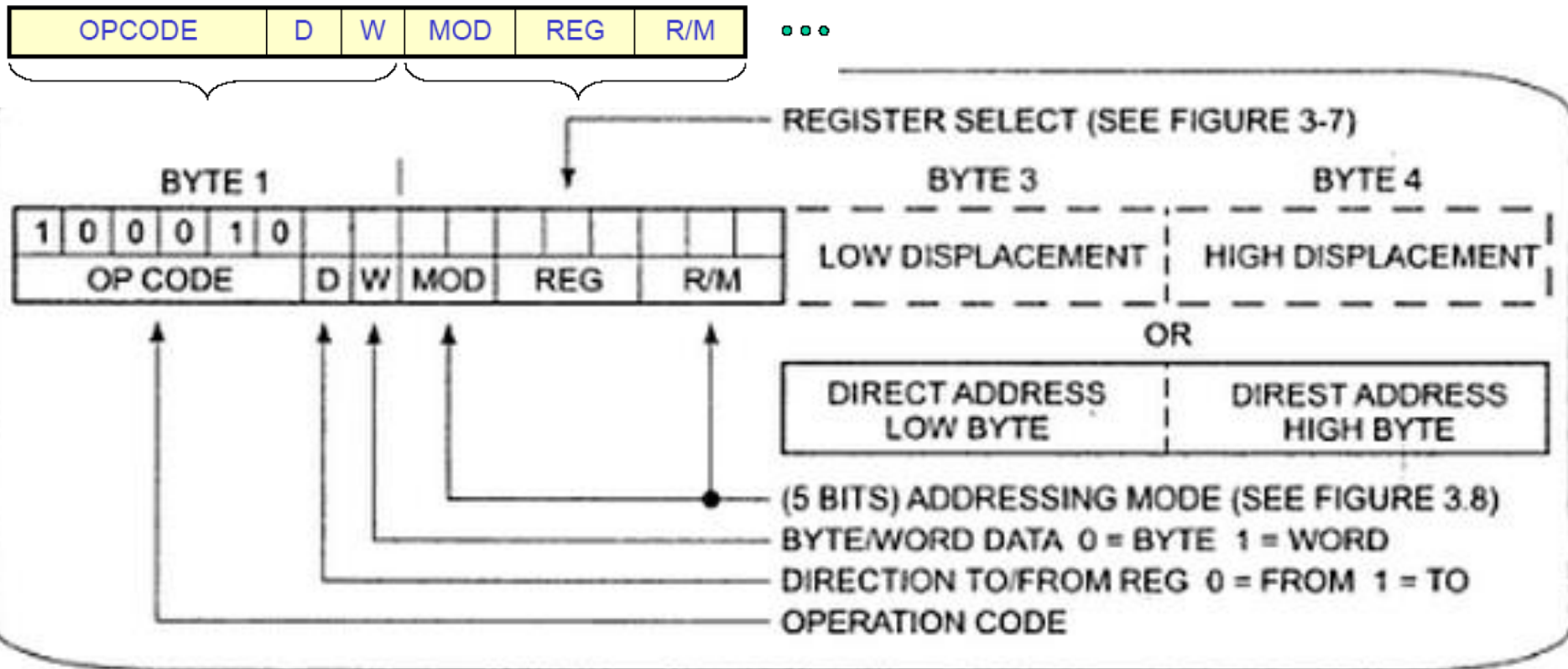
- ▶ The Intel literature shows two different formats for coding 8086 instructions.
- ▶ Instruction templates help you to code the instruction properly.

▶ **Example:**  
`IN AL, 05H`



# MOV Instruction Coding Format

- ▶ MOV data from a register to a register or from a register to a memory location or from a memory location to a register. (**Operation Code of MOV: 100010**)



# MOV Instruction Coding: **REG Field**

- ▶ REG field is used to identify the register of the one 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

# MOV Instruction Coding: MOD and R/M Field

- ▶ 2-bit Mode (MOD) and 3-bit Register/Memory (R/M) fields specify the other operand.
- ▶ Also specify the addressing mode.

RM \ MOD	MOD			
	00	01	10	11
				W = 0    W = 1
000	[BX] + [SI]	[BX] + [SI] + d8	[BX] + [SI] + d16	AL    AX
001	[BX] + [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

# MOV Instruction Coding: MOD and R/M Field

---

- ▶ If the other operand in the instruction is also one of the eight register then put in 11 for MOD bits in the instruction code.
- ▶ If the other operand is memory location, there are 24 ways of specifying how the execution unit should compute the effective address of the operand in the main memory.
- ▶ If the effective address specified in the instruction contains displacement less than 256 along with the reference to the contents of the register then put in 01 as the MOD bits.
- ▶ If the expression for the effective address contains a displacement which is too large to fit in 8 bits then out in 10 in MOD bits.



# Example 1

---

- MOV BL,AL
- Opcode for MOV = 100010
- We'll encode AL so
  - D = 0 (AL source operand)
- W bit = 0 (8-bits)
- MOD = 11 (register mode)
- REG = 000 (code for AL)
- R/M = 011

OPCODE	D	W	MOD	REG	R/M
100010	0	0	11	000	011

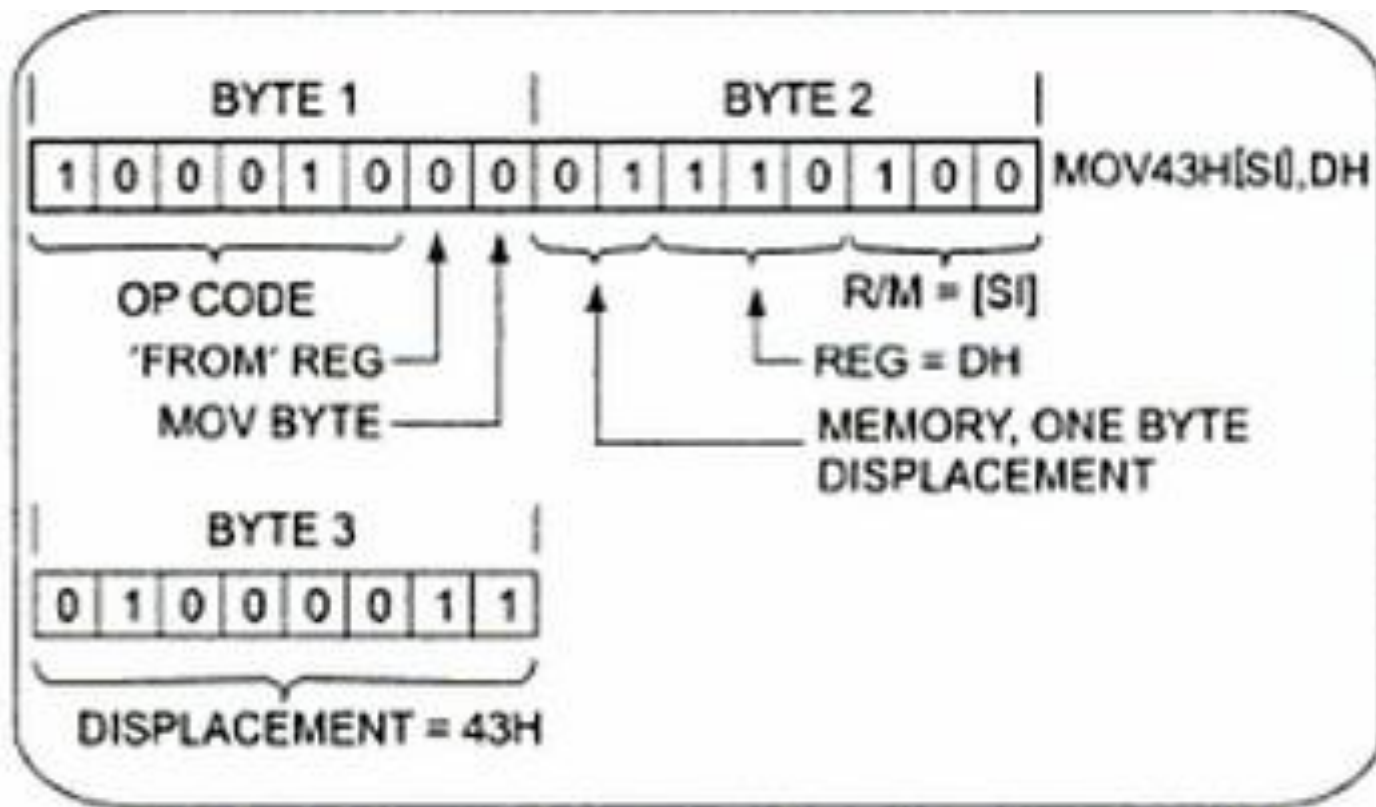
## Example 2

---

- ▶ `MOV CL, [BX]`
- ▶ `100010` – opcode for `MOV`
- ▶ D bit 1 – ‘To’ register (considering CL)
- ▶ W bit 0 – Moving 1 byte
- ▶ Memory, no displacement – 00
- ▶ REG CL – 001
- ▶ R/M – 111 for `[BX]`
- ▶ So, `10001010000001111 = 8A0Fh`

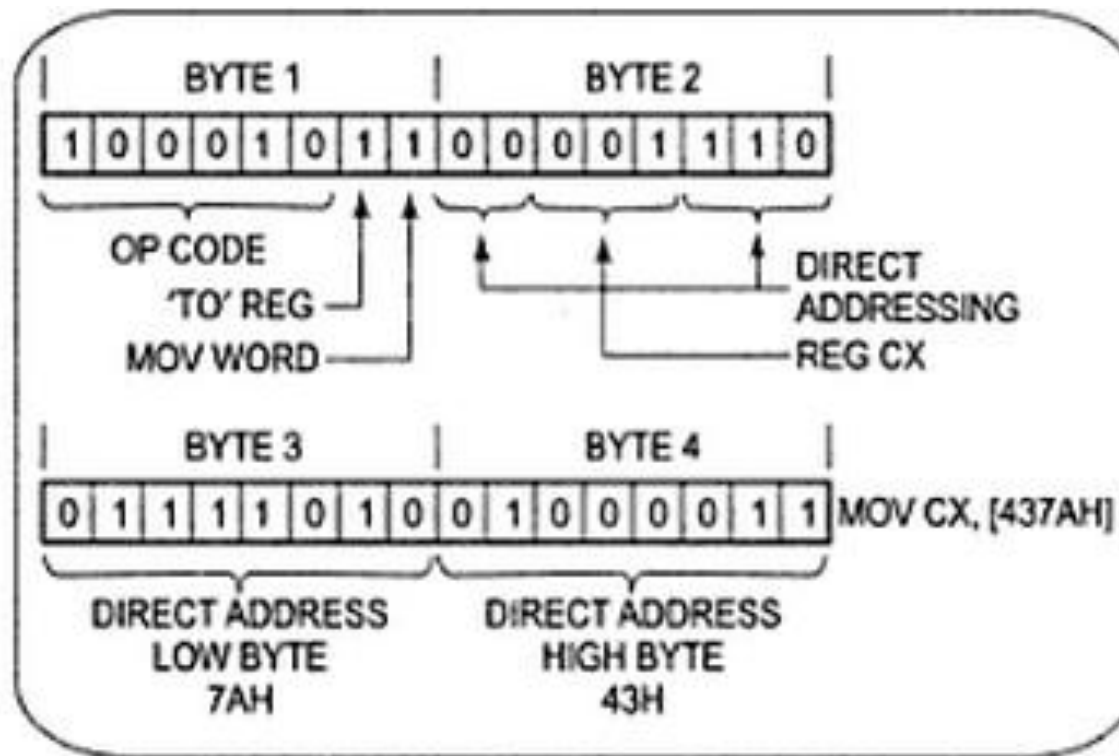
## Example 3

- **MOV 43H [SI], DH:** Copy a byte from DH register to memory location.



## Example 4

- **MOV CX, [437AH]:** Copy the contents of the two memory locations to the register CX.



# Thank You !!

---

