INSTRUCTION SET OF 8086

The 8086 instructions can be classified into following six groups.

1. Data transfer instructions
2. Arithmetic instructions
3. Logical instructions
4. String manipulating instructions
5. Control transfer instructions
6. Processor control instructions

➡ The data transfer group includes instructions for moving data between registers, register and memory, register and stack memory, and accumulator and IO device.

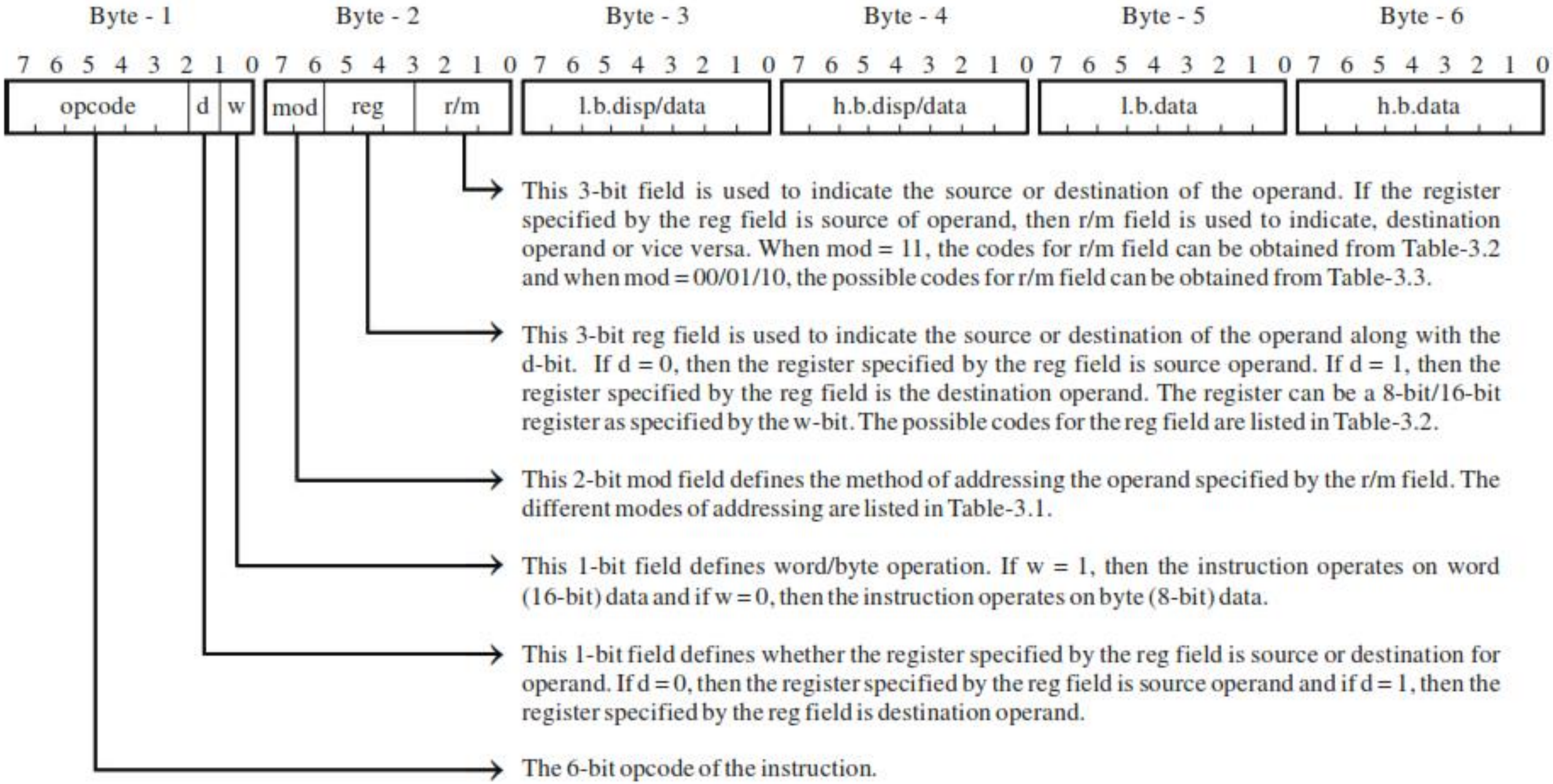
➡ The arithmetic group includes instructions for addition and subtraction of binary, BCD and ASCII data, and instructions for multiplication and division of signed and unsigned binary data.

➡ The logical group includes instructions for performing logical operations like AND, OR, Exclusive-OR, Complement, Shift, Rotate, etc. The string manipulation group includes instructions for moving string data between two memory locations and comparing string data word by word or byte by byte.

➡ The control transfer group includes instructions to call a procedure/subroutine in the main program. It also includes instructions to jump from one part of a program to another part either conditionally (after checking flags) or unconditionally (without checking flags).

➡ The processor control group includes instructions to set/clear the flags, to delay and halt the processor execution.

INSTRUCTIONS FORMAT



This 3-bit field is used to indicate the source or destination of the operand. If the register specified by the reg field is source of operand, then r/m field is used to indicate, destination operand or vice versa. When mod = 11, the codes for r/m field can be obtained from Table-3.2 and when mod = 00/01/10, the possible codes for r/m field can be obtained from Table-3.3.

This 3-bit reg field is used to indicate the source or destination of the operand along with the d-bit. If d = 0, then the register specified by the reg field is source operand. If d = 1, then the register specified by the reg field is the destination operand. The register can be a 8-bit/16-bit register as specified by the w-bit. The possible codes for the reg field are listed in Table-3.2.

This 2-bit mod field defines the method of addressing the operand specified by the r/m field. The different modes of addressing are listed in Table-3.1.

This 1-bit field defines word/byte operation. If w = 1, then the instruction operates on word (16-bit) data and if w = 0, then the instruction operates on byte (8-bit) data.

This 1-bit field defines whether the register specified by the reg field is source or destination for operand. If d = 0, then the register specified by the reg field is source operand and if d = 1, then the register specified by the reg field is destination operand.

The 6-bit opcode of the instruction.

General format of 8086 instruction.

- The size of 8086 instruction is one to six bytes.

- In general, the first byte of the instruction will have a 6-bit opcode and two special bit indicators d-bit and w-bit or (s-bit and w-bit) or (v-bit and w-bit).

w-bit : This bit appears in the format of instructions which can operate on both byte and word data.

If w = 0, then the data operated by the instruction is 8-bit/byte.

If w = 1, then the data operated by the instruction is 16-bit/word.

d-bit : This bit appears in the format of instructions which has a double operand. In double operand instructions, one of the operand should be a register specified by reg field. The d-bit is used to specify whether the register specified by reg field is source operand or destination operand.

If d = 0, then the register specified by reg field is source operand.

If d = 1, then the register specified by reg field is destination operand.

s-bit : This bit appears in the format of arithmetic instructions which operate on immediate data. If s = 1, w = 1 and immediate data is 8-bit then the immediate data is sign extended to 16-bit and used for arithmetic operation.

sw = 00 → 8-bit operation with an 8-bit immediate data.

sw = 01 → 16-bit operation with a 16-bit immediate data.

sw = 11 → 16-bit operation with a sign extended 8-bit immediate operand.

v-bit : This bit appears in the format of shift and rotate instructions.

If v = 0, then the shift/rotate operation is performed one time.

If v = 1, then the content of CL is count value for number of shift/rotate operations.

z-bit : This bit appears in the format of REP prefix for string instructions and is used for comparing with zero flag.

If z = 0, then repeat execution of string instruction until zero flag is zero.

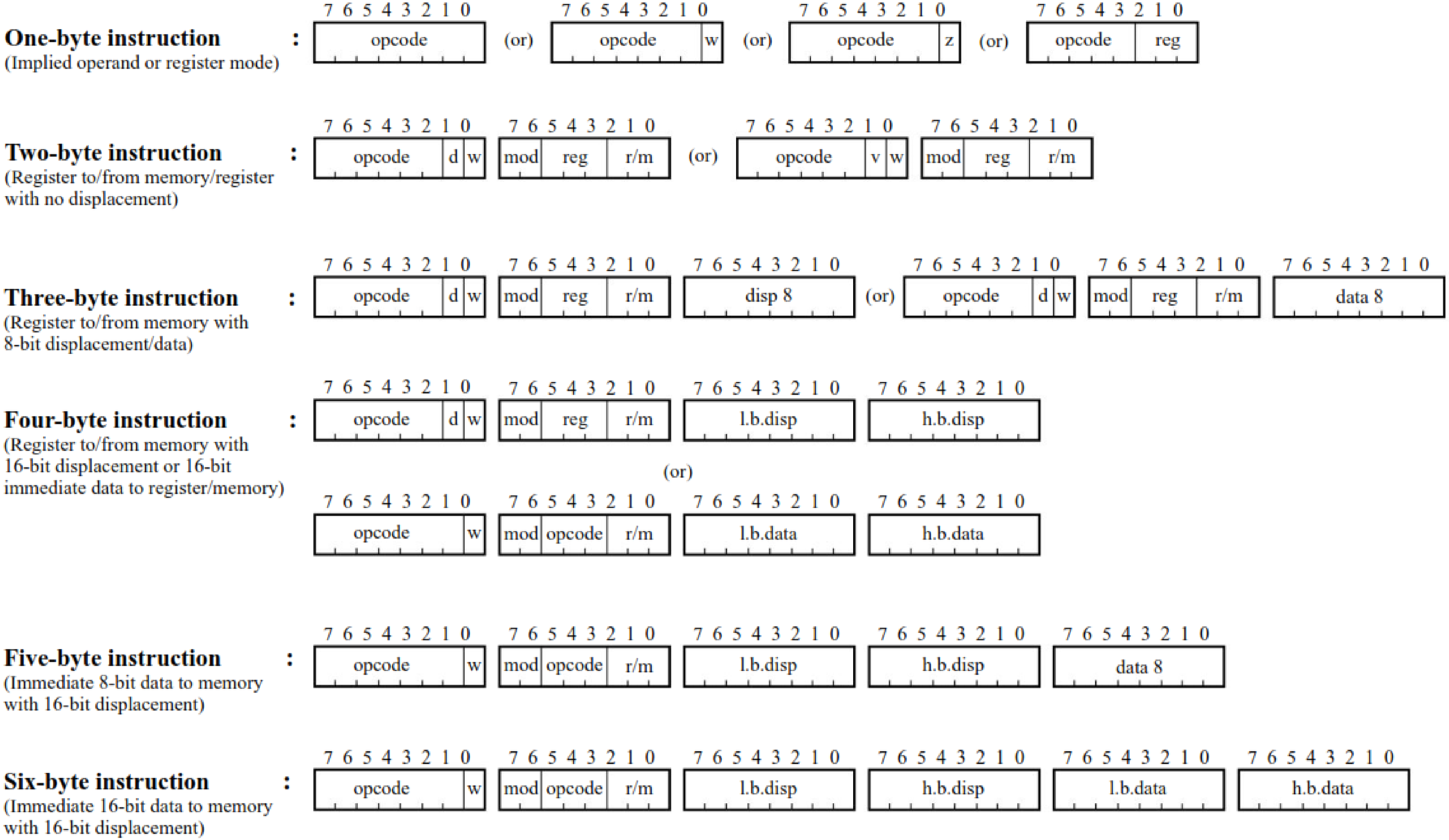
If z = 1, then repeat execution of string instruction until zero flag is one.

In multi-byte instructions, the second byte will specify the addressing mode of the operands. The second byte usually has three fields : mod, reg and r/m. The mod field is 2-bit wide and it defines the method of addressing the operand specified by r/m field. The r/m field is 3-bit wide and it is used to indicate the source or destination operand in memory/register. The reg field is 3-bit wide and it is used to indicate the source or destination operand in register. If register specified by reg field is source operand then r/m field is used to indicate destination operand or vice versa.

Code for mod field	Name of the mode
00	Memory mode with no displacement
01	Memory mode with 8-bit displacement
10	Memory mode with 16-bit displacement
11	Register mode

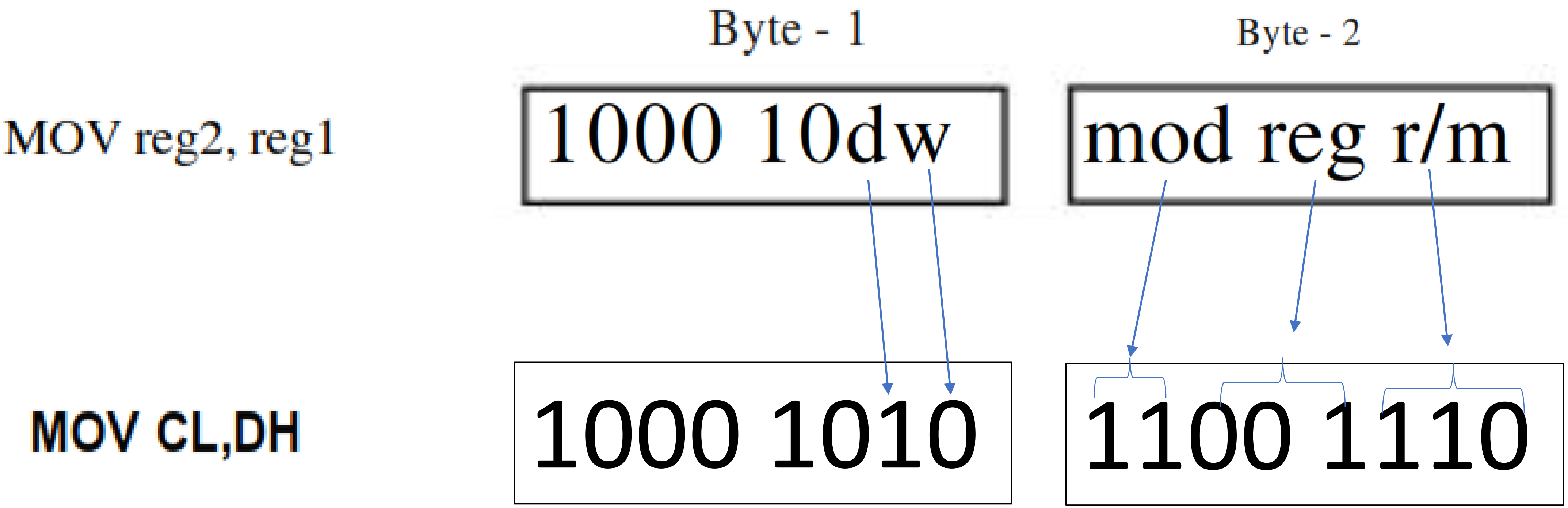
Code for reg field	Name of the register represented by the code when w = 0 or 1	
	When w = 0	When w = 1
000	AL	AX
001	CL	CX
010	DL	DX
011	BL	BX
100	AH	SP
101	CH	BP
110	DH	SI
111	BH	DI

Code for r/m field	Effective address calculation when mod = 00/01/10		
	mod = 00	mod = 01	mod = 10
000	[BX + SI]	[BX + SI + disp8]	[BX + SI + disp16]
001	[BX + DI]	[BX + DI + disp8]	[BX + DI + disp16]
010	[BP + SI]	[BP + SI + disp8]	[BP + SI + disp16]
011	[BP + DI]	[BP + DI + disp8]	[BP + DI + disp16]
100	[SI]	[SI + disp8]	[SI + disp16]
101	[DI]	[DI + disp8]	[DI + disp16]
110	[disp16]	[BP + disp8]	[BP + disp16]
111	[BX]	[BX + disp8]	[BX + disp16]


7 6 5 4 3 2 1 0
One-byte instruction :
(Implied operand or register mode)
opcode (or) opcode w (or) opcode z (or) opcode reg
Two-byte instruction :
(Register to/from memory/register with no displacement)
opcode d w | mod reg r/m (or) opcode v w | mod reg r/m
Three-byte instruction :
(Register to/from memory with 8-bit displacement/data)
opcode d w | mod reg r/m | disp 8 (or) opcode d w | mod reg r/m | data 8
Four-byte instruction :
(Register to/from memory with 16-bit displacement or 16-bit immediate data to register/memory)
opcode d w | mod reg r/m | l.b.disp | h.b.disp
(or)
opcode w | mod opcode r/m | l.b.data | h.b.data
Five-byte instruction :
(Immediate 8-bit data to memory with 16-bit displacement)
opcode w | mod opcode r/m | l.b.disp | h.b.disp | data 8
Six-byte instruction :
(Immediate 16-bit data to memory with 16-bit displacement)
opcode w | mod opcode r/m | l.b.disp | h.b.disp | l.b.data | h.b.data

Note : *reg - register ; mod - mode ; r/m - register/memory ; l.b.disp - low byte of displacement ; h.b.disp - high byte of displacement ; l.b.data - low byte of data ; h.b.data - high byte of data ; data8 - 8-bit data.*

Fig. 3.1 : Examples of 8086 instruction formats.



Byte - 1

Byte - 2

MOV reg2, reg1

1000 10dw

mod reg r/m

MOV CL,DH

1000 1010

1100 1110