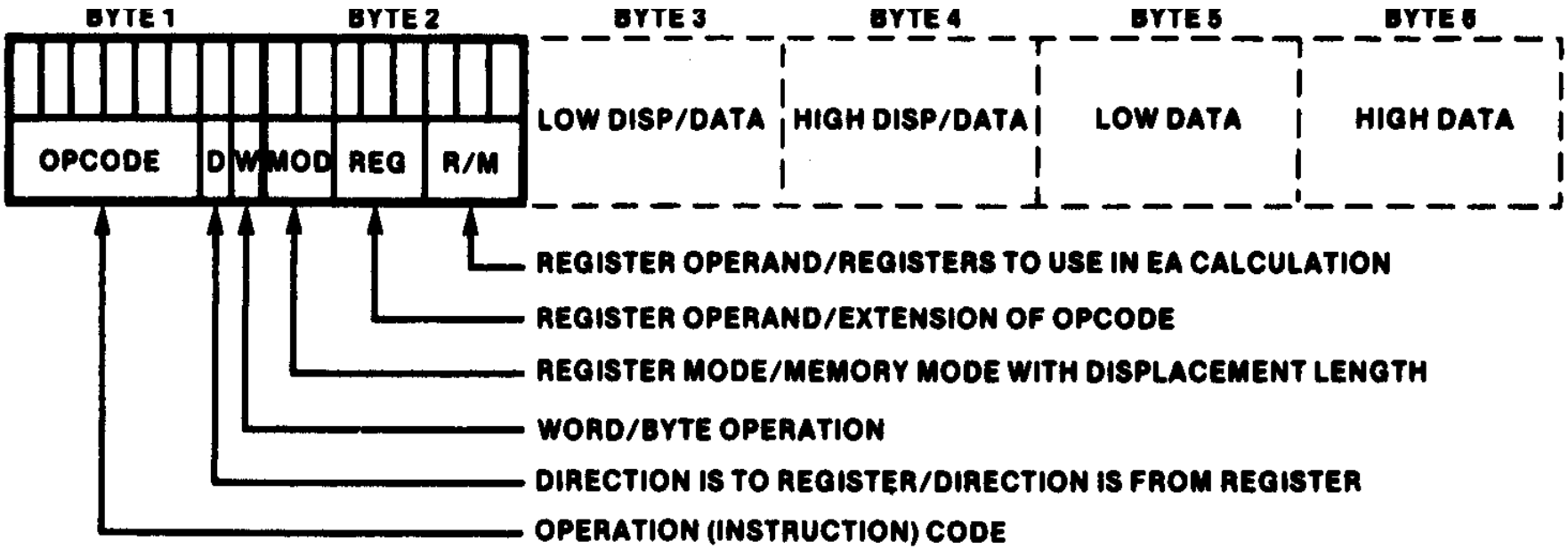
Machine Codes for 8086 Instructions

Machine Language Codes

- Each instruction is coded as one or more bytes
- The first byte is generally an OpCode
  - ➢ A numeric code representing a particular instructions
- Additional Bytes may affect the action of the instruction or provide information about the data acted upon by the instruction

Variable Format Instructions

- The meaning of bits in an instruction varies depending on the instruction
- The number of bytes in an instruction varies depending on the needs of the instruction
- General instruction format for machine code



Machine Code (OpCode)

- 8086 OpCodes are 6,7 or 8 bits
- Byte 1 specification
  - Opcode field (6-bits)
    - Specifies the operation to be performed
  - Register direction bit (D-bit)
    - 1 – the register operand is a destination operand
    - 0 – the register operand is a source operand
  - Data size bit (W-bit)
    - 1 – 16-bit data size
    - 0 – 8-bit data size

Machine Code (mod/reg/r/m)

- Byte 2 specification
  - Register (REG) field (3-bit)
    - Identifies 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

  - Mode (MOD) field (2-bits)
    - Together with R/M field to specify the second operand
  - Register/Memory (R/M) field (3-bit)

Machine Code (mod/reg/r/m)

- Byte 2 specification

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

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

Machine Code(Register/Memory)

- Interpretation depends on **mod** field contents
- **mod** =11 r/m =(same code as **reg**)
  - Second operand is the specified register
- **mod** =00 r/m= 110
  - second argument is the address specified in the next two bytes of the instruction, direct near addressing –DS: offset (Unless segment override is in effect)

Machine Code(Example)

- ***EXAMPLE:***
  - MOV BL, AL
  - Encode the above instruction in machine code
- ***Solution:***
  - OPCODE = 100010 (for MOV), D = 0 (source), W = 0 (8-bit)
  - This leads to BYTE 1 = $10001000_2 = 88_{16}$
  - In byte 2 the source operand, specified by REG, is AL
  - REG = 000, MOD = 11, R/M = 011
  - Therefore, BYTE 2 = $11000011_2 = C3_{16}$
  - MOV BL, AL = $88C3_{16}$

Machine Code(Example)

- ***EXAMPLE:***
  - ADD AX, [SI]
  - Encode the above instruction in machine code

- ***Solution:***
  - OPCODE = 000000 (for ADD), D = 1 (dest.), W = 1 (16-bit)
  - This leads to BYTE 1 = $00000011_2 = 03_{16}$
  - In byte 2 the destination operand, specified by REG, is AX
  - REG = 000, MOD = 00, R/M = 100
  - Therefore, BYTE 2 = $00000100_2 = 04_{16}$
  - ADD AX, [SI] = 0304_{16}

Machine Code(Example)

- *EXAMPLE:*
  - XOR CL, [1234H]
  - Encode the above instruction in machine code

- *Solution:*
  - OPCODE = 001100 (for XOR), D = 1 (dest.), W = 0 (8-bit)
  - This leads to BYTE 1 = $00110010_2 = 32_{16}$
  - In byte 2 the destination operand, specified by REG, is CL
  - REG = 001, MOD = 00, R/M = 110
  - Therefore, BYTE 2 = $00001110_2 = 0E_{16}$
    BYTE 3 = 34_{16} BYTE 4 = 12_{16}
  - XOR CL, [1234H] = $320E3412_{16}$

Machine Code(Example)

- *EXAMPLE:*
  - ADD [BX][DI]+1234H, AX
  - Encode the above instruction in machine code

- *Solution:*
  - OPCODE = 000000 (for ADD), D = 0 (source), W = 1 (16-bit)
  - This leads to BYTE 1 = $00000001_2 = 01_{16}$
  - In byte 2 the destination operand, specified by REG, is AX
  - REG = 000, MOD = 10, R/M = 001
  - Therefore, BYTE 2 = $10000001_2 = 81_{16}$
    BYTE 3 = 34_{16} BYTE 4 = 12_{16}
  - ADD [BX][DI]+1234H, AX = 01813412_{16}

Encoding a Complete Program in Machine Code

- Steps in encoding a complete assembly program:
  - Identify the general machine code format
  - Evaluate the bit fields
  - Express the binary-code instruction in hexadecimal form
- To execute the program, the machine code of the program must be stored in the code segment of memory.
- The first byte of the program is stored at the lowest address.

Program for moving ten consecutive byte from one location to another

- ***Solution***:

```
MOV AX, 2000H    ;LOAD AX REGISTER
MOV DS, AX       ;LOAD DATA SEGMENT ADDRESS
MOV SI, 100H     ;LOAD SOURCE BLOCK POINTER
MOV DI, 120H     ;LOAD DESTINATION BLOCK POINTER
MOV CX, 10H      ;LOAD REPEAT COUNTER
NXTPT: MOV AH, [SI]    ;MOVE SOURCE BLOCK ELEMENT TO AH
       MOV [DI], AH    ;MOVE ELEMENT FROM AH TO DEST. BLOCK
       INC SI          ;INCREMENT SOURCE BLOCK POINTER
       INC DI          ;INCREMENT DESTINATION BLOCK POINTER
       DEC CX          ;DECREMENT REPEAT COUNTER
       JNZ   NXTPT     ;JUMP TO NXTPT IF CX NOT EQUAL TO ZERO
NOP                    ;NO OPERATION
```

Encoding a Program in Machine Code

Instruction	Type of instruction	Machine code
MOV AX,2000H	Move immediate data to register	$1011100000000000000100000_2 = B80020_{16}$
MOV DS,AX	Move register to segment register	$1000111011011000_2 = 8ED8_{16}$
MOV SI,100H	Move immediate data to register	$101111100000000000000001_2 = BE0001_{16}$
MOV DI,120H	Move immediate data to register	$101111110010000000000001_2 = BF2001_{16}$
MOV CX,10H	Move immediate data to register	$101110010001000000000000_2 = B91000_{16}$
MOV AH,[SI]	Move memory data to register	$1000101000100100_2 = 8A24_{16}$
MOV [DI],AH	Move register data to memory	$1000100000100101_2 = 8825_{16}$
INC SI	Increment register	$01000110_2 = 46_{16}$
INC DI	Increment register	$01000111_2 = 47_{16}$
DEC CX	Decrement register	$01001001_2 = 49_{16}$
JNZ NXTPT	Jump on not equal to zero	$0111010111110111_2 = 75F7_{16}$
NOP	No operation	$1001000_2 = 90_{16}$

Encoding a Program in Machine Code

Memory address	Contents	Instruction
200H	B8H	MOV AX,2000H
201H	00H	
202H	20H	
203H	8EH	MOV DS,AX
204H	D8H	
205H	BEH	MOV SI,100H
206H	00H	
207H	01H	
208H	BFH	MOV DI,120H
209H	20H	
20AH	01H	
20BH	B9H	MOV CX,10H
20CH	10H	
20DH	00H	
20EH	8AH	MOV AH,[SI]
20FH	24H	
210H	88H	MOV [DI],AH
211H	25H	
212H	46H	INC SI
213H	47H	INC DI
214H	49H	DEC CX
215H	75H	JNZ $-9
216H	F7H	
217H	90H	NOP