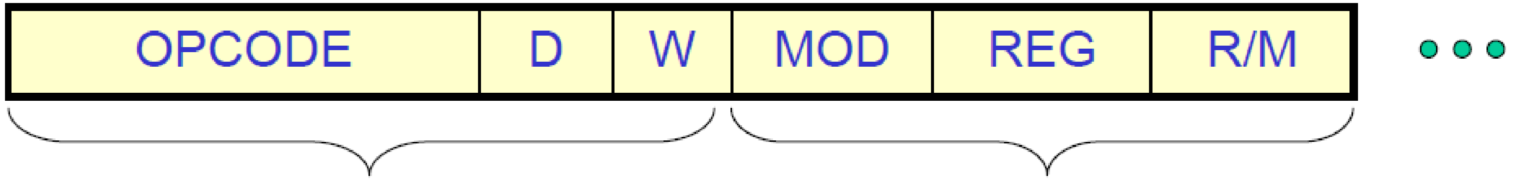
x86 Instruction Encoding Part-2

Lecture # 37

Converting Assembly Language Instructions to Machine Code



OPCODE	D	W	MOD	REG	R/M

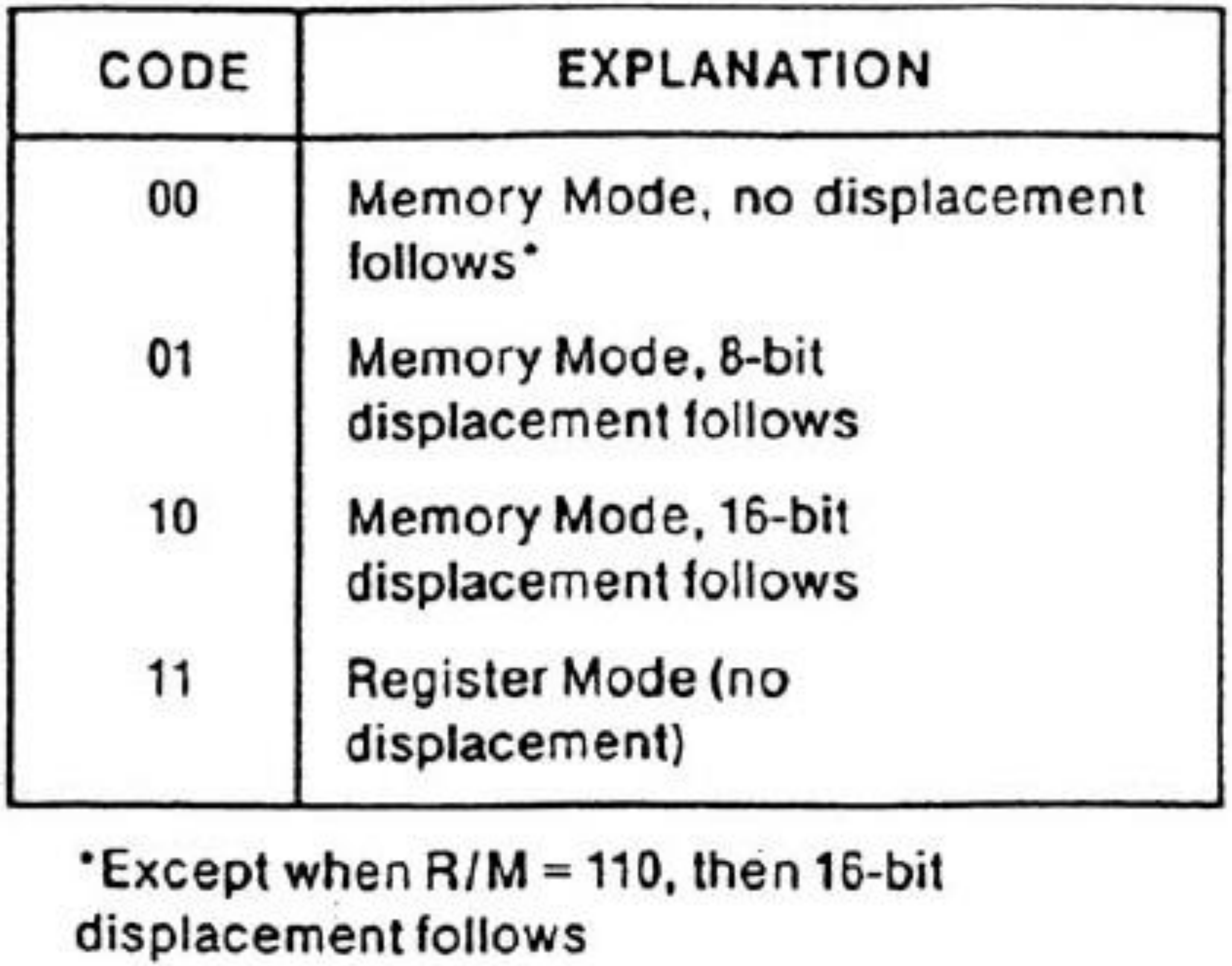
• • •

- An instruction can be coded with 1 to 6 bytes
- **Byte 1 contains three kinds of information:**
  - 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
- **Byte 2 has two fields:**
  - Mode field (MOD) – 2 bits
  - Register field (REG) - 3 bits
  - Register/memory field (R/M field) – 3 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

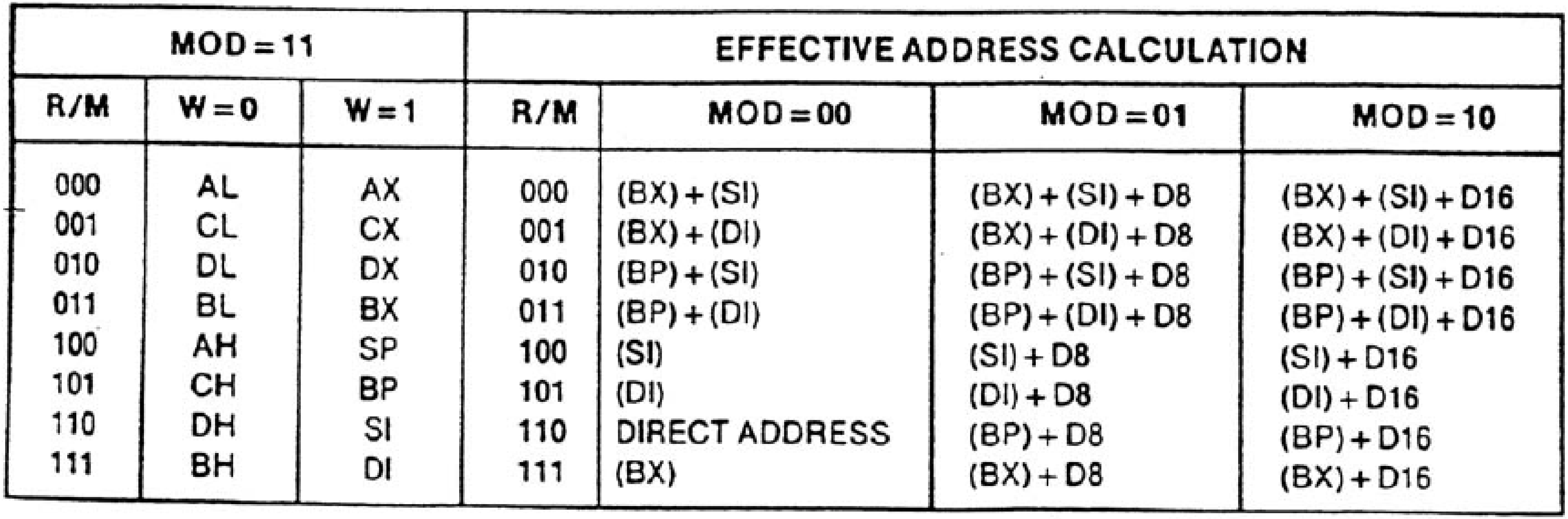
2 bit Mode Field



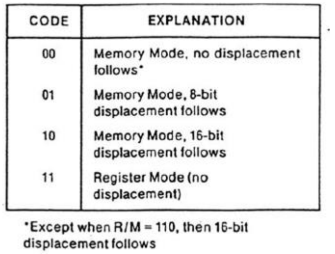
CODE	EXPLANATION
00	Memory Mode, no displacement follows*
01	Memory Mode, 8-bit displacement follows
10	Memory Mode, 16-bit displacement follows
11	Register Mode (no displacement)

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

3 bit R/M Field



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



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

2 bit mode and 3 bit R/M fields together identify the second operand.

Examples

- 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

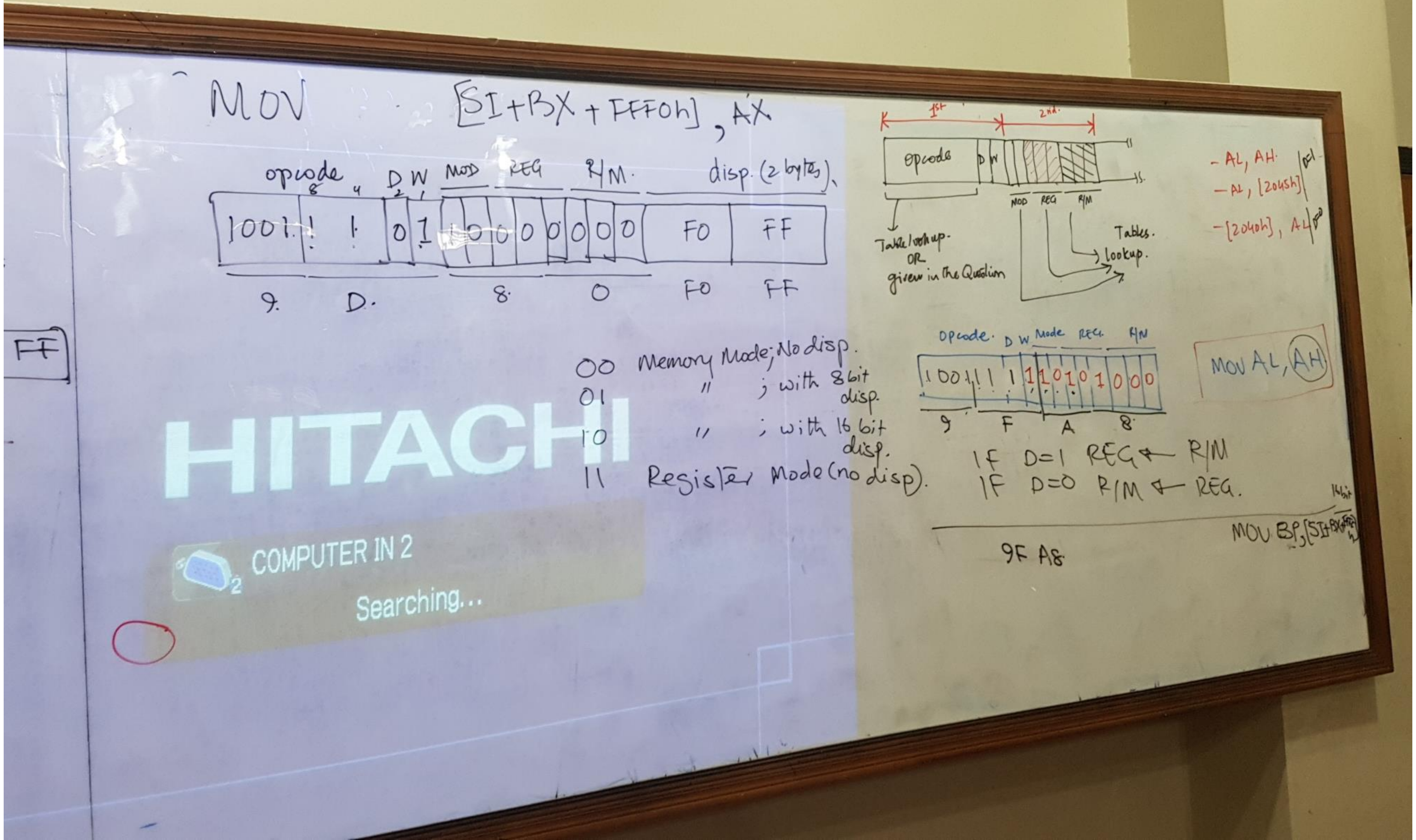
OPCODE	D	W	MOD	REG	R/M
100010	0	0	11	000	011

MOV BL,AL => 10001000 11000011 = 88 C3h

ADD AX,[SI] => 00000011 00000100 = 03 04 h

ADD [BX][DI] + 1234h, AX => 00000001 10000001 __ __ h

=> 01 81 34 12 h



MOV [SI+BX+FFF0h], AX

opcode		D	W	MOD		REG			R/M			disp. (2 bytes).	
8	4	2	1										
1001 1	1	0	1	1	0	0	0	0	0	0	0	F0	FF
9	D			8				0				F0	FF

00 Memory Mode; No disp.
01 " ; with 8 bit disp.
10 " ; with 16 bit disp.
11 Resister Mode (no disp).

1st — 2nd.

opcode | D | W | MOD | REG | R/M

Table lookup. OR given in the Question

Tables. lookup.

- AL, AH. D=1
- AL, [2045h]
- [2040h], AL D=0

opcode.				D	W	Mode		REG.			R/M		
1 0 0 1	1	1	1	1	1	1	0	1	0	1	0	0	0
9	F					A				8			

MOV AL, AH

IF D=1 REG ← R/M
IF D=0 R/M ← REG.

16bit
MOV BP, [SI+BX+FFF0h]

9F A8.

DEMOs of Semester Projects (Part # 1)

Basic Android to Arduino two way communication interfacing:

- Using MIT App Inventor 2 for Android Application
- Write Arduino code in C/C++ using Arduino IDE
- Interfacing Bluetooth models with Arduino UNO