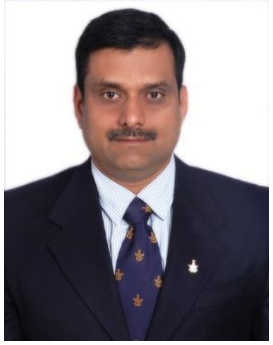
MPI Tutorial-4

8086 Assembly Language to Op-code

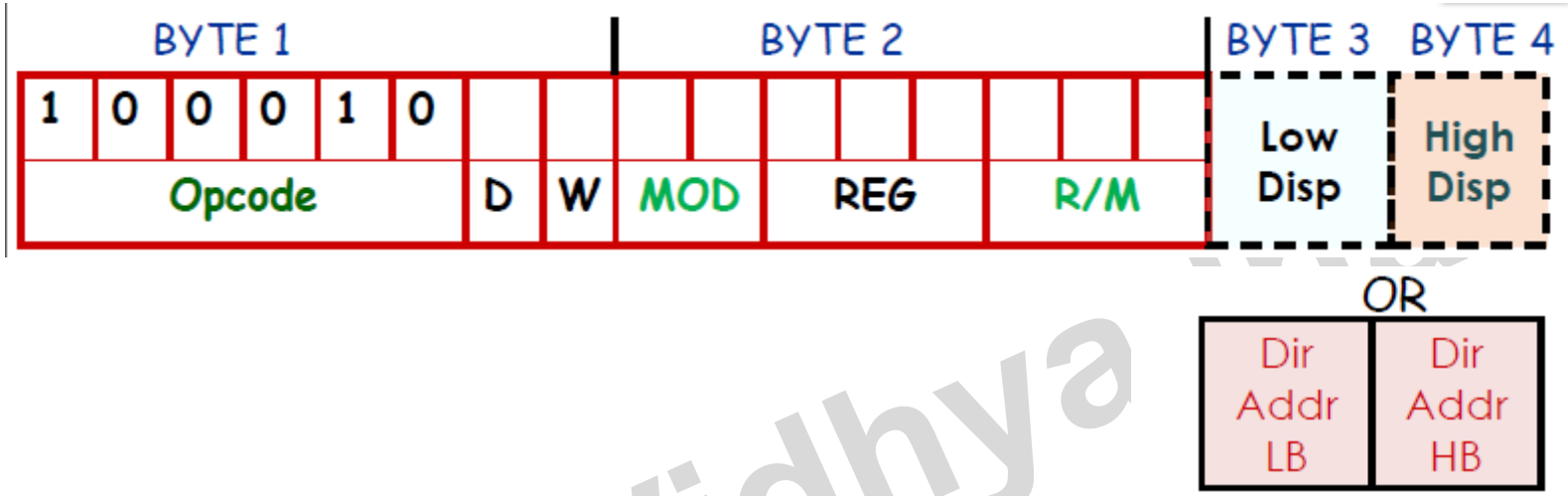
&

Data Transfer ALPs

By Dr. Sanjay Vidhyadharan



OP Code Format in 8086



BYTE 1						BYTE 2	BYTE 3
				1 BIT	3 BITS	LOW DISP.	HIGH DISP.
OPCODE				W	REG		

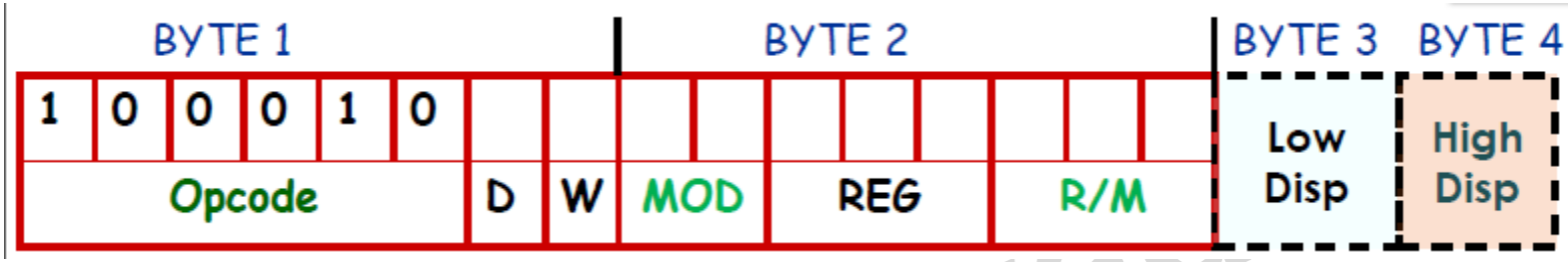
D=0 (Direction from Register)
D=1 (Direction to Register)
W=0 (Data-byte)
W=1 (Data-word)
MOD+ R/M - Addressing Modes

MOD with R/M (16 BITS)

Operands	Memory Operands			Register Operands	
	No Displacement	Displacement 8-bit	Displacement 16-bit		
MOD / R/M	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		BL	BX
100	(SI)	(SI) + D8	(SI) + D16	AH	SP
101	(DI)	(DI) + D8	(DI) + D16	CH	BP
110	D16	(BP) + D8	(BP) + D16	DH	SI
111	(BX)	(BX) + D8	(BX) + D16	BH	DI

Problem-1

Write the equivalent machine language code: **MOV CH, BL.**



MOV = Move — 76543210
Register/Memory to/from Register — 100010 dw
Immediate to Register/Memory — 1100011 w

REG	W = 0	W = 1
000	AL	AX
001	CL	CX
010	DL	DX
011	BL	BX

REG	W = 0	W = 1
100	AH	SP
101	CH	BP
110	DH	SI
111	BH	DI

Problem-2

Write the equivalent machine language code: **MOV 1234 [BP], DX**

Opcode	D	W	MOD	REG	R/M	LB displacement	HB displacement
100010	0	1	10	010	110	34H	12H

Final Ans : 89 96 34 12H.

MOV = Move 76543210
Register/Memory to/from Register 100010 dw
Immediate to Register/Memory 1100011 w

REG	W = 0	W = 1
000	AL	AX
001	CL	CX
010	DL	DX
011	BL	BX

MOD	Interpretation
00	Memory mode with no displacement follows except for 16-bit Displacement when R/M = 110
01	Memory mode with 8-bit displacement
10	Memory mode with 16-bit displacement
11	Register mode (no displacement)

Problem-3

Write the equivalent machine language code: **MOV DS: 2345 [BP], DX**

Problem-3

Write the machine language equivalent code for : **MOV DS: 2345 [BP], DX**

Solution:

Here we have to specify DX using REG field. The D bit must be o, indicating that DX is the source register. The REG field must be 010 to indicate DX register. The w bit must be 1 to indicate word operation. 2345 [BP] is specified with MOD=10 and R/M = 110 and displacement = 2345 H. Whenever BP is used to generate the Effective Address (EA), the default segment would be SS. In this example, we want the segment register to be DS, we have to provide the 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.

SR	Segment register
00	ES
01	CS
10	SS
11	DS

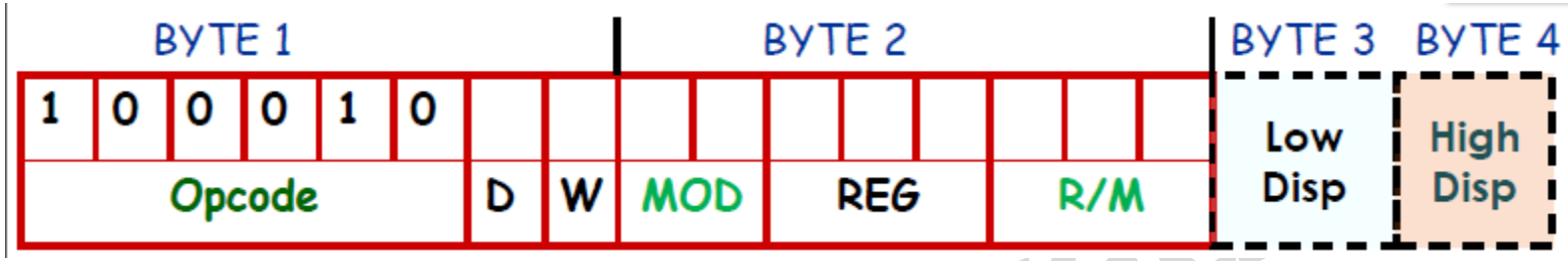
To specify DS register, the SOP byte would be 001 11 110 = 3E H. Thus the 5 byte code for this instruction would be **3E 89 96 45 23 H.**

SOP	Opcode	D	W	MOD	REG	R/M	LB disp.	HD disp.
3EH	1000 10	0	1	10	010	110	45	23

Suppose we want to code MOV SS: 2345 (BP), DX. This generates only a 4 byte code, without SOP byte as SS is already the default segment register in this case.

Problem-4

Write the equivalent machine language code: **SUB BX, [DI]**



001010 1 1 00 011 101

= 2 B 1 D (H).

SUB = Subtract

Reg/Memory and Register to Either	001010 dw
Immediate from Register/Memory	100000 sw
Immediate from Accumulator	0010110 w

REG	W = 0	W = 1
000	AL	AX
001	CL	CX
010	DL	DX
011	BL	BX

Problem-4

Write the Op-code for : **SUB BX, [DI]**

Solution:

This instruction subtracts the 16 bit content of the memory location addressed by DI and DS from BX.

The 6 bit Opcode for SUB is 001010(base-2).

D=1 so that REG field of byte 2 is the destination operand.

W=1 indicates 16 bit operation.

MOD = 00

REG = 011

R/M = 101

The machine code is 0010 1011 0001 1101

= 2 B 1 D (base-16).

Problem-5

Write a program to add two 16-bit numbers where starting address is 2000 and the numbers are at 3000 and 3002 memory address and store result into 3004 and 3006 memory address.

Problem-5

Write a program to add two 16-bit numbers where starting address is 2000 and the numbers are at 3000 and 3002 memory address and store result into 3004 and 3006 memory address.

Solution:

Memory	Mnemonics	Operands	Comment
2000	MOV	CX, 0000	[CX] <- 0000
2003	MOV	AX, [3000]	[AX] <- [3000]
2007	MOV	BX, [3002]	[BX] <- [3002]
200B	ADD	AX, BX	[AX] <- [AX] + [BX]
200D	JNC	2010	Jump if no carry
200F	INC	CX	[CX] <- [CX] + 1
2010	MOV	[3004], AX	[3004] <- [AX]
2014	MOV	[3006], CX	[3006] <- [CX]
2018	HLT		Stop

Problem-6

Write a program in 8086 microprocessor to find out the addition of two 8-bit BCD numbers, where numbers are stored from starting memory address 2000 : 500 and store the result into memory address 2000 : 600 and carry at 2000 : 601.

Problem-6

Write a program in 8086 microprocessor to find out the addition of two 8-bit BCD numbers, where numbers are stored from starting memory address 2000 : 500 and store the result into memory address 2000 : 600 and carry at 2000 : 601.

Solution:

MEMORY ADDRESS	MNEMONICS	COMMENT
400	MOV AL, [500]	AL<-[500]
404	MOV BL, [501]	BL<-[501]
408	ADD AL, BL	AL<-AL+BL
40A	DAA	DECIMAL ADJUST AL
40B	MOV [600], AL	AL->[600]
40F	MOV AL, 00	AL<-00
411	ADC AL, AL	AL<-AL+AL+cy(prev)
413	MOV [601], AL	AL->[601]
417	HLT	END

Problem-7

Write a program to add the content of memory location 2000 : 0500 with content of memory location 3000 : 0600 and store result into 5000 : 0700 memory location..

Problem-7

Write a program to add the content of memory location 2000 : 0500 with content of memory location 3000 : 0600 and store result into 5000 : 0700 memory location..

Solution:

Memory	Mnemonics	Operands	Comment
1000	MOV	CX, 2000	[CX] <- 2000
1004	MOV	DS, CX	[DS] <- [CX]
1006	MOV	AX, [500]	[AX] <- [500]
100A	MOV	CX, 3000	[CX] <- 3000
100E	MOV	DS, CX	[DS] <- [CX]
1010	ADD	AX, [600]	[AX] <- [AX] + [600]
1014	MOV	CX, 5000	[CX] <- 5000
1018	MOV	ES, CX	[ES] <- [CX]
101A	MOV	[700], AX	[700] <- [AX] RESULT
101E	HLT		Stop

Problem-8

Write a program in 8086 microprocessor to multiply two 8-bit numbers, where numbers are stored from offset 500 and store the result into offset 600.

Problem-8

Write a program in 8086 microprocessor to multiply two 8-bit numbers, where numbers are stored from offset 500 and store the result into offset 600.

Solution:

MEMORY ADDRESS	MNEMONICS	COMMENT
400	MOV SI, 500	SI=500
403	MOV DI, 600	DI=600
406	MOV AL, [SI]	AL<-[SI]
408	INC SI	SI=SI+1
409	MOV BL, [SI]	BL<-[SI]
40B	MUL BL	AX=AL*BL
40D	MOV [DI], AX	AX->[DI]
40F	HLT	END