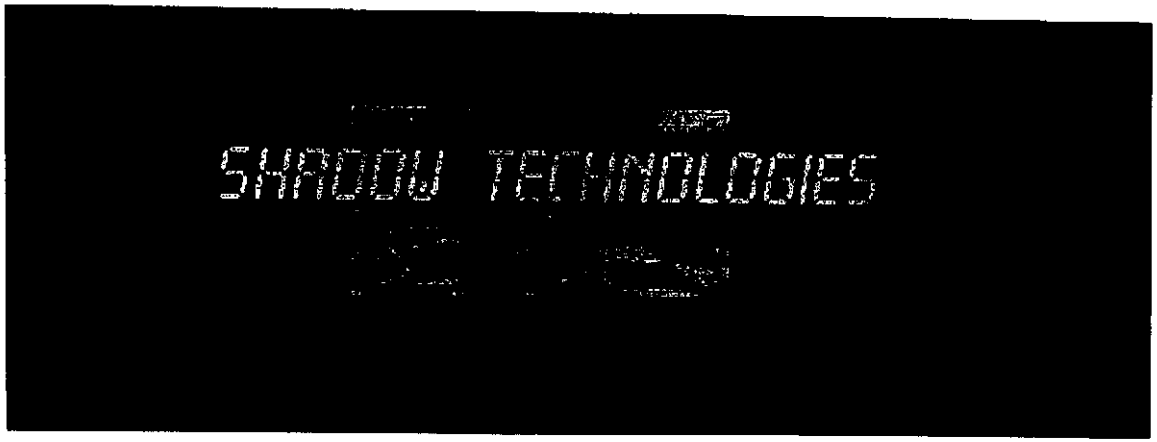
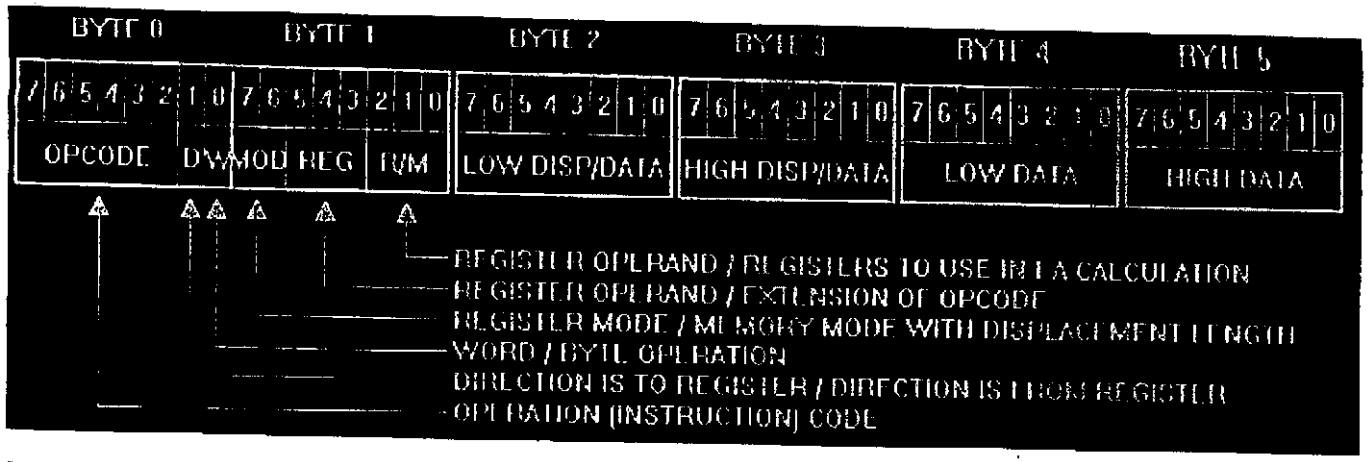


142

8086/8088 Assembly Language Programming

Assembly/Machine Language Coding

Machine Language Encoding Derivatives



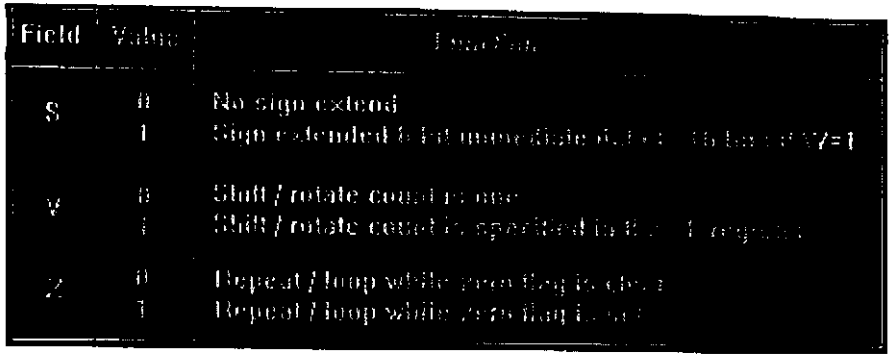
MOD=11			EFFECTIVE ADDRESS CALCULATION		
R/M	W=0	W=1	R/M	MOD=00	MOD=01	MOD=10
000	AL	AX	000	(BX) + (SI)	(BX) + (SI) + D8	(BX) + (SI) + D16
001	CL	CX	001	(BX) + (DI)	(BX) + (DI) + D8	(BX) + (DI) + D16
010	DL	DX	010	(BP) + (SI)	(BP) + (SI) + D8	(BP) + (SI) + D16
011	BL	BX	011	(BP) + (DI)	(BP) + (DI) + 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

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

MOD	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

Register	SEL
CS	00
DS	01
SS	10
ES	11



Field	Value	Function
S	0	No sign extend
	1	Sign extended 8-bit immediate data to 16 bits if W=1
V	0	Shift/rotate count is one
	1	Shift/rotate count is specified in CL register
Z	0	Repeat/loop while zero flag is clear
	1	Repeat/loop while zero flag is set

DATA TRANSFER

MOV = Move

Register/Memory to/from register	100010dw|mod reg r/m|Disp-Lo|Disp-Hi
Immediate to register/meory	1100011w|mod 000 r/m|Disp-Lo|Disp-Hi|Data|Data if w=1
Immediate to register	1011wreg|Data|Data if w=1
Memory to accumulator	1010000w|Addr-Lo|Addr-Hi
Accumulator to memory	1010001w|Addr-Lo|Addr-Hi
Register/Memory to segment register	10001110|mod 0 SR r/m|Disp-Lo|Disp-Hi
Segment Register to register/Memory	10001100|mod 0 SR r/m|Disp-Lo|Disp-Hi

PUSH = Push

Register/Memory	11111111|mod 110 r/m|Disp-Lo|Disp-Hi
Register	01010reg
Segment register	000SR110

POP = Pop

Register/Memory	11111111|mod 110 r/m|Disp-Lo|Disp-Hi
Register	01011reg
Segment register	000SR111

XCHG = Exchange

Register/memory with register	1000011w|mod reg r/m|Disp-Lo|Disp-Hi
Register with accumulator	10010reg

IN = Input from

Fixed Port	1110010w|Data-8
Variable Port	1110110w

Out = Ouput to

Fixed port	1110011w|Data-8
Variable port	1110111w

XLAT = Table lookup to al	11010111
LEA = Load EA to register	10001101|mod reg r/m|Disp-Lo|Disp-Hi
LDS = Load Pointer to DS	11000101|mod reg r/m|Disp-Lo|Disp-Hi
LES = Load Pointer to ES	11000100|mod reg r/m|Disp-Lo|Disp-Hi
LAHF = Load AH with Flags	10011111
SAHF = Store AH into Flags	10011110
PUSHF = Push flags	10011100
POPF = Pop flags	10011101

ARITHMETIC

Add = add

Reg/memory with register to either	000000dw|mod reg r/m|Disp-Lo|Disp-Hi
Immediate to register/memory	100000sw|mod 000 r/m|Disp-Lo|Disp-Hi|Data|Data if s:w=01
Immediate to accumulator	0001010w|Data|Data if w=1

ADC = Add with carry

Reg/Memory with register to either	000100dw|mod reg r/m|Disp-Lo|Disp-Hi
Immediate to register/memory	100000sw|mod 010 r/m|Disp-Lo|Disp-Hi|Data if s:w=01
Immediate to accumulator	0001010w|Data|Data if w=1

INC = Increment

Register/Memory	1111111w|mod 000 r/m|Disp-Lo|Disp-Hi
Register	01000reg|

AAA = Ascii adjust for add	00110111
DAA = Decimal adjust for add	00100111

SUB = Subtract

Reg/memory and register to either	001010dw|mod reg r/m|Disp-Lo|Disp-Hi
Immediate from register/memory	100000sw|mod 101 r/m|Disp-Lo|Disp-Hi|Data|Data if s:w=01
Immediate from accumulator	0010110w|Data|Data if w=1

SBB = Subtract with borrow

Reg/memory and register to either	000110dw|mod reg r/m|Disp-Lo|Disp-Hi
Immediate from register/memory	100000sw|mod 011 r/m|Disp-Lo|Disp-Hi
Immediate from accumulator	0001110w|Data|Data if w=1

DEC = Decrement

Register/memory	1111111w|mod 001 r/m|Disp-Lo|Disp-Hi
Register	01001reg

NEG = Change sign 1111011w|mod 0111 r/m|Disp-Lo|Disp-Hi

CMP = Compare

Register/memory and register	001110dw|mod reg r/m|Disp-Lo|Disp-Hi
Immediate with register/memory	100000sw|mod 111 r/m|Disp-Lo|Disp-Hi|Data|Data if s:w=1
Immediate with accumulator	0011110w|Data

AAS = Ascii adjust for subtract	00111111
DAS = Decimal adjust for subtract	00101111
MUL = Multiply(unsigned)	1111011w|mod 100 r/m|Disp-Lo|Disp-Hi
IMUL = Integer multiply(signed)	1111011w|mod 101 r/m|Disp-lo|Disp-Hi
AAM = Ascii adjust for multiply	11010100|00001010|Disp-Lo|Disp-Hi
DIV = Divide(unsigned)	1111011w|mod 110 r/m|Disp-Lo|Disp-Hi
IDIV = Integer Divide(Signed)	1111011w|mod 111 r/m|Disp-Lo|Disp-Hi
AAD = Ascii Adjust for divide	11010101|00001010|Disp-Lo|Disp-Hi
CBW = Convert byte to word	10011000
CWD = Convert word to doubleword	10011001

LOGIC

NOT = Invert	1111011w|mod 010 r/m|Disp-Lo|Disp-Hi
SHL/SAL = Shift logical/arithmetic left	110100vw|mod 100 r/m|Disp-Lo|Disp-Hi
SHR = Shift logical right	110100vw|mod 101 r/m|Disp-Lo|Disp-Hi
SAR = Shift arithmetic right	110100vw|mod 111 r/m|Disp-Lo|Disp-Hi

ROL	= Rotate left	110100vw|mod 000 r/m|Disp-Lo|Disp-Hi
ROR	= Rotate right	110100vw|mod 001 r/m|Disp-Lo|Disp-Hi
RCL	= Rotate through carryf left	110100vw|mod 010 r/m|Disp-Lo|Disp-Hi
RCR	= Rotate through carryf right	110100vw|mod 011 r/m|Disp-Lo|Disp-Hi

AND = Boolean AND

Reg/memory with register to either	001000dw|mod reg r/m|Disp-Lo|Disp-Hi
Immediate to register/memory	1000000w|mod 100 r/m|Disp-Lo|Disp-Hi|Data|Data if w=1
Immediate to accumulator	0010010w|Data|Data if w=1

TEST = And function to flags no result

Register/memory and register	000100dw|mod reg r/m|Disp-Lo|Disp-Hi
Immediate data and register memory	1111011w|mod 000 r/m|Disp-Lo|Disp-Hi|Data|Data if w=1
Immediate and accumulator	1010100w|Data

OR = Boolean OR

Reg/Memory and register to either	000010dw|mod reg r/m|Disp-Lo|Disp-Hi
Immediate to register memory	0011010w|mod 001 r/m|Disp-Lo|Disp-Hi
Immediate to accumulator	1010100w|Data|Data if w=1

XOR = Exclusive Boolean OR

Reg/memory and register to either	001100dw|mod reg r/m|Disp-Lo|Disp-Hi
Immediate to register/memory	0011010w|Data|Disp-Lo|Disp-Hi|Data|Data if w=1
Immediate to accumulator	0011010w|Data|Data if w=1

STRING MANIPULATION

REP = Repeat	1111001z
MOVS = Move byte/word	1010010w
CMPS = Compare byte/word	1010011w
SCAS = Scan byte/word	1010111w
LODS = Load byte/word to AL/AX	1010110w
STDS = Store byte/word from AL/AX	1010101w

CONTROL TRANSFER

CALL = Call

Direct within segment	11101000|IP-INC-LO|IP-INC-HI
Indirect within segment	11111111|mod 010 r/m|Disp-Lo|Disp-Hi
Direct intersegment	10011010|IP-Lo|IP-Hi|CS-LO|CS-HI
Indirect intersegment	11111111|mod 011 r/m|Disp-Lo|Disp-Hi

RET = Return from call

Within segment	11000011
Within segment adding immediate to SP	11000010|Data-Lo|Data-Hi
Intersegment	11001011
Intersegment adding immediate to SP	11001010|Data-Lo|Data-Hi

JE/JZ = Jump equal/zero	01110100|IP-INC8
JL/JNGE = Jump less/not greater-equ	01111100|IP-INC8
JLE/JNG = Jump less or equ/not greater	01111110|IP-INC8
JB/JNAE = Jump below/not above equal	01110010|IP-INC8
JBE/JNA = Jump below equal/not above	01110110|IP-INC8
JP/JPE = Jump parity/parity even	01111010|IP-INC8
JO = Jump on overflow	01110000|IP-INC8
JS = Jump on sign	01111000|IP-INC8
JNE/JNZ = Jump not equal/not zero	01110101|IP-INC8
JNL/JGE = Jump not less/greater or equ	01111101|IP-INC8
JNLE/JG = Jump not less equal/greater	01111111|IP-INC8
JNB/JAE = Jump not below/above or equ	01110011|IP-INC8
JNBE/JA = Jump not below or equ/above	01110111|IP-INC8
JNP/JPO = Jump not parity/parity odd	01111011|IP-INC8
JNO = Jump not overflow	01110001|IP-INC8
JNS = Jump not sign	01111001|IP-INC8
JCXZ = Jump on cx zero	11100011|IP-INC8

JMP = Unconditional Jump

Direct within segment	11101001|IP-INC-LO|IP-INC-HI
Direct within segment-short	11101011|IP-INC8
Indirect within segment	11111111|mod 100 r/m|Disp-Lo|Disp-Hi
Direct intersegment	11101010|IP-LO|IP-HI|CS-LO|CS-HI
Indirect intersegment	11111111|mod 101 r/m|Disp-Lo|Disp-Hi

LOOP = Loop CX times	11100010|IP-INC8
LOOPZ/LOOPE = Loop while zero/equ	11100001|IP-INC8
LOOPNZ/LOOPNE = Loop not zero/equ	11100000|IP-INC8

INT	= Interupt	
Type specified		11001101|Data-8
INT3		11001100
INTO	= Interupt on overflow	11001110
IRET	= Interupt return	11001111

PROCESSOR CONTROL

CLC	= Clear carry	11111000
CMC	= Complement carry	11110101
STC	= Set carry	11111001
CLD	= Clear direction	11111100
STD	= Set direction	11111101
CLI	= Clear interupt	11111010
STI	= Set interupt	11111011
HLT	= Halt	11110100
WAIT	= wait	10011011
ESC	= Escape to external device	11011xxx|mod yyy r/m|Disp-Lo|Disp-Hi
LOCK	= Buss lock prefix	11110000
SEGMENT	= Segment overide prefix	001SR110

8 of 8