Machine Language Coding

By Dan Kohn
University of Southern Mississippi
Computer Engineering Technology

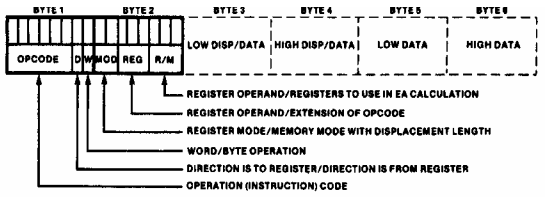
Converting ASM to Machine Code

- In general, each Machine instruction specifies:
  - What operation is to be performed
  - Operand(s)
  - Byte or Word Operation
  - Reg to Reg OR Reg to Mem
  - Addressing mode (if Mem)

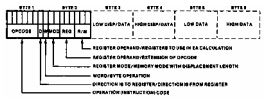
One Byte Instructions

- Generally specify a simpler operation with a register or flag bit.

General Instruction Format



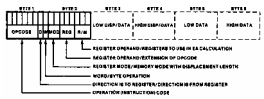
General Instruction Format



Opcode (6 bits)

Specifies the operation, such as add, subtract or move, that is to be performed.

General Instruction Format



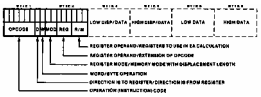
Register Direction Bit (D Bit)

Tells whether the register operand specified by REG in byte 2 is the source or destination operand.

Logic 1 indicates destination
Logic 0 indicates source.

General Instruction Format

Data Size bit (W bit)
8 or 16 bit operation

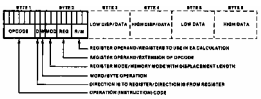


Logic 0 – 8 bit
Logic 1 – 16 bit

General Instruction Format

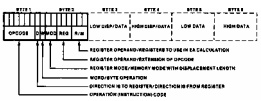
Byte 2

Mod Field (mode)



00 – Memory mode (no displacement)
01 – Memory mode (8 bit displacement follows)
10 – Memory mode (16 bit displacement follows)
11 – register mode

General Instruction Format

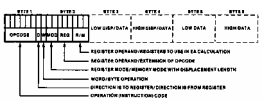


Byte 2

Reg Field
Identifies the register for 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

General Instruction Format



Byte 2

R/M Field Specifies the 2nd operand (Dependent on MOD field)

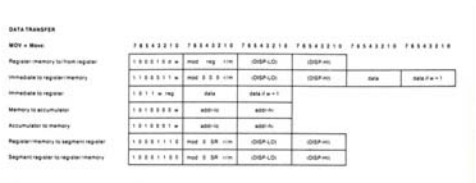
MOD = 11			EFFECTIVE ADDRESS CALCULATION			
R/M	W = 0	W = 1	R/M	MOD = 00	MOD = 01	MOD = 10
000	AL	AX	000	(BX) + (SI)	(BX) + (SI) + D8	(BX) + (SI) + D16
001	CL	CX	001	(BX) + (DI)	(BX) + (DI) + D8	(BX) + (DI) + D16
010	DL	DX	010	(BP) + (SI)	(BP) + (SI) + D8	(BP) + (SI) + D16
011	BL	BX	011	(BP) + (DI)	(BP)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

Example 4.1

- Convert the following to machine language

MOV BL, AL

1st find op code in table 3-6. In our case:



So our first bits will be 100010

Example 4.1 (cont)

2nd Bit D, determines whether the register specified by the REG part of byte 2 is a source or destination operand. We will encode AL in REG field of byte to; therefore, D is set to 0 for source operand.

Example 4.1 (cont)

3rd Bit W, this is an 8 bit operation so W will be set to 0

This makes byte one of the instruction:

$10001000_2 = 88_{16}$

Example 4.1 (cont)

4th The source operand is AL so we set the REG to 000 (as per fig 4.2)

5th Since the second operand is also a register, MOD field is set to 11 (as per fig 4-3a)

6th R/M is now set to 011 to indicate the destination is BL (as per table 4-3b).

Example 4.1 (cont)

This makes byte two of the instruction

$11000011_2 = C3_{16}$

So the machine code for MOV BL, AL is:

$88C3_{16}$

But…….

There is a second answer to the example:

If the D bit is changed to a 1 (so the REG of byte two is destination) then:

Byte 1 of the instruction is 10001010_2 or $8A_{16}$

But….. (cont)

Byte 2:

The Destination operand is BL so we set the REG to 011 (as per fig 4.2)

Since the second operand is also a register, MOD field is set to 11 (as per fig 4-3a)

R/M is now set to 000 to indicate the source is AL (as per table 4-3b).

So the second byte would be 11011000_2 or $D8_{16}$

Debug

Debug is a DOS program that is used to:

- write and debug a machine language / ASM program
- examine / modify memory
- examine / modify IO Ports
- examine / modify registers.

Read sections 4.3 – 4.10 (you will need to know this for future labs!)