

CH4 Data Transfers, Addressing, and Arithmetic

Data Transfer Instructions

Operand Types

- Immediate - a constant integer
 - value is encoded within the instruction
- Register - the name of a register
 - register name is converted to a number and encoded within the instruction
- Memory - reference to a location in memory
 - memory address is encoded within the instruction, or a register holds the address of a memory location

MOV instruction

Move from source to destination. Syntax: MOV destination, source

- No more than one memory operand permitted
- **CS, EIP, and IP** cannot be the destination
- No immediate to segment moves

```
.data
    count BYTE 100
    wVal  WORD 2
.code
    MOV  BL, count    ; 8-bit
    MOV  AX, wVal     ; 16-bit
    MOV  count, AL    ; 8-bit

; ERROR :
    MOV  AL, wVal     ; 8-bit , 16-bit
    MOV  AX, count    ; 16-bit, 8-bit
    MOV  EAX, count   ; 32-bit, 8-bit
```

```

.data
    bVal    BYTE    100
    bVal2   BYTE    ?
    wVal    WORD    2
    dVal    DWORD    5

.code
; ERROR EXAMPLE :
MOV DS ,45      ;immediate move to DS not permitted
MOV ESI, wVal   ;size mismatch 32-bit, 16-bit
MOV EIP, dVal   ;EIP cannot be the destination
MOV 25 , bVal   ;immediate value cannot be destination
MOV bVal2, bVal;memory-to-memory move not permitted

```

Zero extension

When you copy a smaller value into a larger destination the **MOVZX instruction fills (extends)** the upper half of the destination with zeros.

The destination must be a register

```

MOV    BL, 10001111b
MOVZX  AX, BL      ; AX: 00000000 10001111

MOVZX  reg32, reg/mem8
MOVZX  reg32, reg/mem16
MOVZX  reg16, reg/mem8

```

Sign extension

The **MOVSX instruction fills the upper half of the destination with a copy of the source operand's sign bit.**

```

MOV    BL, 10001111b
MOVSX  AX, BL      ; AX: 11111111 10001111

```