

Multiplication and Division Instructions

Muhammad Afzaal
m.afzaal@nu.edu.pk

Book Chapter

- “Assembly Language for x86 Processors”
- Author “Kip R. Irvine”
- 6th Edition
- Chapter 7
 - Section 7.4

Integer Multiplication

- Integer multiplication in x86 can be performed as a 32-bit, 16-bit or 8-bit operation
- The default destination operand in these instructions is the accumulator register (EAX/AX/AL)

MUL Instruction (1/2)

- Used to multiply unsigned numbers
- The multiplier and multiplicand must be the same size and product is twice their size
- MUL instruction has a single operand which is multiplier

`MUL reg/mem8 ;AX = AL * 8-bit reg/mem`

`MUL reg/mem16 ;DX:AX = AX * 16-bit reg/mem`

`MUL reg/mem32 ;EDX:EAX = EAX * 32-bit reg/mem`

MUL Instruction (2/2)

- MUL sets the CF and OF if upper half of product is non-zero
- When AL is multiplied with an 8-bit value, CF and OF are set if AH is non-zero

Multiplicand	Multiplier	Product
AL	8-bit reg/mem	AX
AX	16-bit reg/mem	DX:AX
EAX	32-bit reg/mem	EDX:EAX

IMUL Instruction

- Used to multiply signed numbers
- Preserves sign by extending highest bit of lower half of the product into the upper bits of product
- Three formats of the IMUL instruction

- One-operand format

`IMUL reg/mem ;AX = AL * reg/mem`

- Two-operand format

`IMUL reg, reg/mem ;AX = reg * reg/mem`

- Three-operand format

`IMUL reg, reg/mem, imm ;reg = reg/mem * imm`

DIV Instruction

- Used to divide unsigned integers
- Performs 8-bit, 16-bit and 32-bit integer division
- Takes only one operand which is the divisor

`DIV reg/mem8`

`DIV reg/mem16`

`DIV reg/mem32`

Dividend	Divisor	Quotient	Remainder
AX	8-bit reg/mem	AL	AH
DX:AX	16-bit reg/mem	AX	DX
EDX:EAX	32-bit reg/mem	EAX	EDX

IDIV Instruction

- Used to divide signed numbers
- Uses same operand types as `DIV` instruction
- Before executing 8-bit division, the dividend (`AX`) must be completely sign-extended using
 - `CBW` (Convert Byte to Word)
 - `CWD` (Convert Word to Double-word)
 - `CDQ` (Convert Double-word to Quad-word)
- Remainder has the same size as dividend

Sign Extension Instructions

■ CBW

- Extends the sign-bit of AL into AH

```
MOV AL, -8 ; AL=1111 1000
```

```
CBW          ; AX=1111 1111 1111 1000
```

■ CWD

- Extends the sign-bit of AX into DX

```
MOV AX, -8 ; AX=1111 1111 1111 1000
```

```
CWD          ; AX=1111 1111 1111 1000
```

```
              ; DX=1111 1111 1111 1111
```

■ CDQ

- Extends the sign-bit of EAX into EDX

Extended Addition and Subtraction

- Extended precision addition/subtraction helps add/subtract numbers having almost unlimited size
- ADC helps to add two numbers using the carry flag
- SBB helps to subtract two numbers using borrow from carry flag

ADC Instruction

- Adds both a source operand and the value of CF to a destination operand
- Instruction format and limitations are same as that of ADD instruction

```
MOV DL, 0
MOV AL, 0FFh
ADD AL, 0FFh      ; AL=FEh, CF=1
ADC DL, 0          ; DL/AL=01FEh
```

SBB Instruction

- Subtracts both a source operand and value of CF from a destination operand
- Possible operands are same as for the SUB instruction

```
MOV AH, 7
MOV AL, 1
SUB AL, 2      ; AL=FFh, CF=1
SBB AH, 0      ; AH/AL=06FFh
```