

CS 47

Machine-Level Programming: Review

Topics

- Arithmetic operations

class12.ppt

CS 47 Spring 2008

Address Computation Instruction

`leal Src, Dest`

- `Src` is address mode expression
- Set `Dest` to address denoted by expression

Uses

- Computing address **without** doing memory reference
 - E.g., translation of `p = &x[i];`
- Computing arithmetic expressions of the form $x + k*y$
 - $k = 1, 2, 4, \text{ or } 8.$

- 2 -

CS 47 Spring, 2008

Some Arithmetic Operations

| Format | Computation |
|--------|-------------|
|--------|-------------|

Two Operand Instructions

| | |
|------------------------------|-----------------------------------------------------|
| <code>addl Src, Dest</code> | $Dest = Dest + Src$ |
| <code>subl Src, Dest</code> | $Dest = Dest - Src$ |
| <code>imull Src, Dest</code> | $Dest = Dest * Src$ |
| <code>sall Src, Dest</code> | $Dest = Dest \ll Src$ Also called <code>shll</code> |
| <code>sarl Src, Dest</code> | $Dest = Dest \gg Src$ Arithmetic |
| <code>shrl Src, Dest</code> | $Dest = Dest \gg Src$ Logical |
| <code>xorl Src, Dest</code> | $Dest = Dest \wedge Src$ |
| <code>andl Src, Dest</code> | $Dest = Dest \& Src$ |
| <code>orl Src, Dest</code> | $Dest = Dest Src$ |

- 3 -

CS 47 Spring, 2008

Some Arithmetic Operations

| Format | Computation |
|--------|-------------|
|--------|-------------|

One Operand Instructions

| | |
|------------------------|--------------------|
| <code>incl Dest</code> | $Dest = Dest + 1$ |
| <code>decl Dest</code> | $Dest = Dest - 1$ |
| <code>negl Dest</code> | $Dest = - Dest$ |
| <code>notl Dest</code> | $Dest = \sim Dest$ |

- 4 -

CS 47 Spring, 2008

Examples

```
%eax = 00110101 (53)
%ecx = 00001111 (15)
%edx = 11000011 (-61) or (195)
%ebx = 00001001 (9)
```

```
addb %ecx, %eax      %eax = 01000100 (68)
subb %ecx, %edx      %edx = 10110100 (-76)
imulb %ebx, %ecx     %ecx = 10000111 (135 -> -121)
salb $3, %ecx        %ecx = 01111000 (120)
sarb $4, %edx        %edx = 11111100 (-4)
shrb $4, %edx        %edx = 00001100 (12)
```

- 5 -

CS 47 Spring, 2008

Examples

```
%eax = 00110101 (53)
%ecx = 00001111 (15)
%edx = 11000011 (-61)
%ebx = 00001001 (9)
```

```
xorb %ecx, %eax      %eax = 00111010 (58)
andb %ecx, %eax      %eax = 00000101 (5)
orb %ecx, %eax        %eax = 00111111 (63)
incb %ecx             %ecx = 00010000 (16)
dec b %edx            %edx = 11000010 (-62)
neg b %eax            %eax = 11001011 (-53)
not b %ecx            %eax = 11110000 (-16)
```

- 6 -

CS 47 Spring, 2008

HW2 Common Errors

Repeating binary expansion

$$x = 1.[110] = 1.110110110\dots$$

Notice that $1.75 < x < 2.0$ (use this to check answer)

Shift left by period (3)

$$8x = 1110.[110]$$

$$8x - x = 7x = 1110 - 1 = 1101 = 13$$

$$x = 13/7 = 1.[857142]$$

- 7 -

CS 47 Spring, 2008

HW2 Common Errors (2)

$3/5 = 11/101$ Do long division in binary

$$\begin{array}{r}
 \overline{0.1001[1001]} \\
 101 \overline{) 11.0000000} \leftarrow \text{starting value} \\
 \underline{10.1} \\
 .1000 \\
 \underline{101} \\
 110 \leftarrow \text{same as starting value} \\
 \underline{101}
 \end{array}$$

Check: $16x = 1001.[1001]$ $16x = 9 + x$ $15x = 9$
 $x = 9/15 = 3/5.$

- 8 -

CS 47 Spring, 2008