CS 47

Machine-Level Programming: Review

Topics

- **Machine Instructions**
- **Repeating Digits**

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, or 8.**

Some Arithmetic Operations

Format		Computation	
Two Operand Instructions			
`addl`	*Src,Dest*	*Dest* = *Dest* + *Src*	
`subl`	*Src,Dest*	*Dest* = *Dest* – *Src*	
`imull`	*Src,Dest*	*Dest* = *Dest* * *Src*	
`sall`	*Src,Dest*	*Dest* = *Dest* << *Src*	Also called `shll`
`sarl`	*Src,Dest*	*Dest* = *Dest* >> *Src*	Arithmetic
`shrl`	*Src,Dest*	*Dest* = *Dest* >> *Src*	Logical
`xorl`	*Src,Dest*	*Dest* = *Dest* ^ *Src*	
`andl`	*Src,Dest*	*Dest* = *Dest* & *Src*	
`orl`	*Src,Dest*	*Dest* = *Dest* | *Src*	

Some Arithmetic Operations

Format		Computation
One Operand Instructions		
`incl`	*Dest*	*Dest* = *Dest* + 1
`decl`	*Dest*	*Dest* = *Dest* – 1
`negl`	*Dest*	*Dest* = – *Dest*
`notl`	*Dest*	*Dest* = ~ *Dest*

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)
```


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)
decb %edx             %edx = 11000010   (-62)
negb %eax             %eax = 11001011   (-53)
notb %ecx             %eax = 11110000   (-16)
```

Evaluating Repeating Digits

Repeating binary expansion

x = 1.[011] = 1.011011011…

Notice that 1.25 < x < 1.5 (use this to check answer)

Shift left by period (3)

8x = 1011.[011]

8x – x = 7x = 1011 – 1 = 1010 = 10

x = 10/7 = 1.[428571]

Finding Repeating Digits

7/5 = 111/101 Do long division in binary

```
        1.0110[0110]
    ┌──────────────
101 │ 111.0000000 ← starting value
    101
    ----
     1000        ←
      101
      ---
       110
       101
       ---
        1000   ← same as previous value
```

Check: 16x = 10110.[0110] 16x = 21 + x 15x = 21

x = 21/15 = 7/5 = 1.4

HW2 Problem 4

Address	Value
0x100	0xFD
0x104	0xB8
0x108	0x24
0x10c	0x44

Register	Value
%eax	0x100
%ecx	0x4
%edx	0x3

Instruction	Destination	Value
addl %edx, 8(%eax)	0x108	0x27
leal (%eax,%edx,2), %ecx	%ecx	0x106
orl $0x120, %ecx	%ecx	0x124
negl 4(%eax)	0x104	0xFFFFFF48
shrl $2, 12(%eax)	0x10C	0x11
orl %edx, 8(%eax)	0x108	0x27