CS430 Computer Architecture

Spring 2015

Chapter 12
Types of Operations

- Data Transfer - copies data from a source operand into a destination operand
- x86 Examples (Reference: http://zeus.cs.pacificu.edu/ryand/cs320/2005/cs320.html)
  - mov ax,1 ; move 1 into ax
  - movzx ax, 10000000b ; mov 128 zero-extended into ax
  - movsx ax, 10000000b ; mov -128 sign-extended into ax
  - push ebx ; push 32-bit contents of ebx onto the stack
  - pop edx

Types of Operations

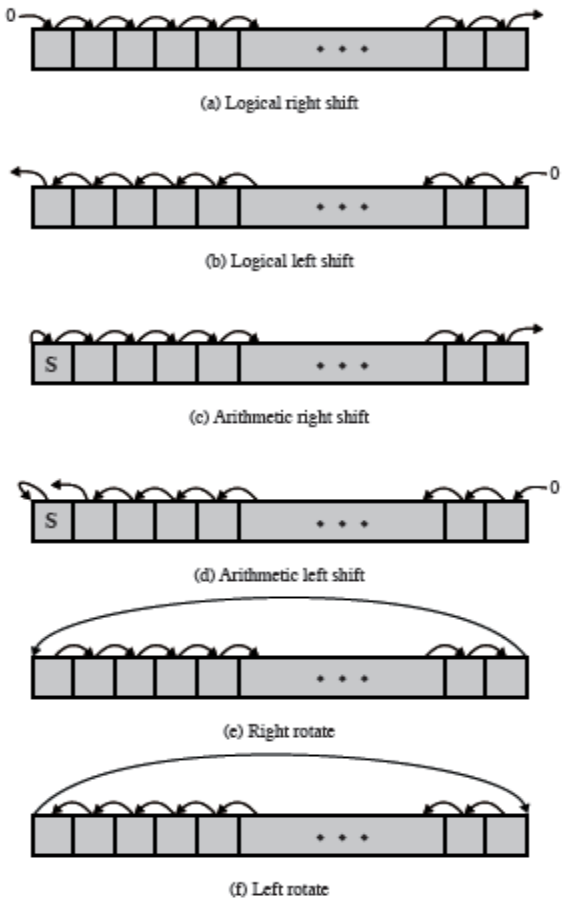
- Arithmetic - perform some arithmetic calculation and in the case where the processor has a flags register, sets the flags accordingly
- x86 Examples
  - add ax,bx ; ax<-ax+bx
  - sub ax,1 ; ax<-ax-1
  - inc cx
  - dec cx

Types of Operations

- Logical - instructions used to do some kind of bit manipulation
- x86 Examples
  - and bh,0fh
  - or ax,10h
  - xor ax,bx

Types of Operations

- More Logical
  - shr ax,1 ; (a)
  - shl ax,1 ; (b)
  - sar bh,1 ; (c)
  - sal bh,1 ; (d)
  - ror edx,1 ; (e)
  - rol edx,1 ; (f)



Types of Operations

- Transfer of Control
  - Conditional branch (conditional jump)
  - Unconditional branch
  - Subroutine call
  - Interrupt

Types of Operations

- Conditional branch - branching is conditionally based on some flags register or some status register
- x86 Example
  - jne top ; branch to top if ZF = 0
  - jb top ; unsigned … branch to top if not above or equal
    ; CF = 1
  - jl top ; signed … branch to top if not greater or equal
    ; SF <> OF

Types of Operations

- Conditional branch
- x86 Example
  - Conditional branch instructions assume a calculation occurred setting flags in the flags register BEFORE the branch occurs
  - dec ax
    jne top

Types of Operations

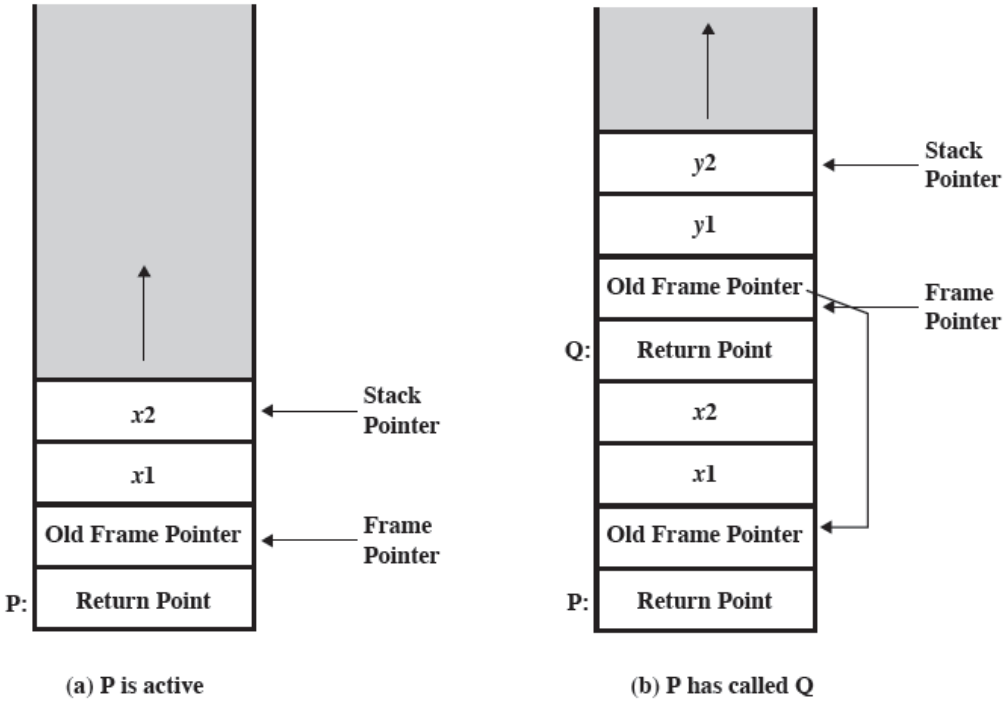
- Unconditional branch - the branch occurs regardless
- x86 Example
  - jmp top

Types of Operations

- Subroutine call
- x86 Example
  - call Foo ; Foo is an assembly language subroutine

Types of Operations

- Subroutine call - what is happening below?



Types of Operations

- Subroutine call - a typical x86 procedure might begin with the following code:

```
push ebp

mov ebp, esp

sub esp, space_for_locals
```

Types of Operations

- Interrupt
- x86 Example
  - int 0h ; transfer control to the address stored in the
    ; interrupt vector table at location 0

Types of Operations

- Interrupt - in x86 Real Mode,
  - an interrupt has an integer in the range of 0-255 called the interrupt type
  - The addresses from 00000 to 003ff are reserved for interrupt vectors
  - An interrupt vector is an address (segment & offset) of a particular interrupt service routine

Types of Operations

- Interrupt Vector Type

Interrupt Vector Type	Stored At
0	00000:00003
1	00004:00007
....	
t	4t : 4t+3

- We see that each interrupt vector requires four bytes. The first two bytes contain the offset (bytes reversed). The next two bytes contain the segment (bytes reversed).

Types of Operations

- How do interrupts work? Remember, after most instructions, the microprocessor checks for pending interrupts. If detected, the microprocessor

1) Push the flags register on the stack (why?)
2) Clear the interrupt and trap flags (why?)
3) Push CS
4) Determine the interrupt location based on the type
5) CS = second word of the interrupt vector
6) Push IP
7) IP = first word of the interrupt vector

- The instruction IRET transfers control back to the caller