Bit Shifts (Practice)

ICS312

Machine-Level and Systems Programming

Henri Casanova (henric@hawaii.edu)

(q1) Logical vs. Arithmetic Shift

- For each code fragment below give the content of the register value after the shift

```
mov al, 0F0h
sar al, 1
```

```
mov al, 0FFh
shl al, 1
```

```
mov al, 003h
sar al, 2
```

```
mov ax, 01FE0h
sar al, 1
```

```
mov ax, 08000h
sar ax, 15
```

(q1) Solutions

- For each code fragment below give the content of the register value after the shift

```
mov al, 0F0h
sar al, 1
```

F8

```
mov al, 0FFh
shl al, 1
```

FE

```
mov al, 003h
sar al, 2
```

00

```
mov ax, 01FE0h
sar al, 1
```

F0

```
mov ax, 08000h
sar ax, 15
```

FFFF

(q2) Multiplying / Dividing

- For each code fragment below, which are meant to multiply/divide values, say whether the numerical result is valid for signed and unsigned values

unsigned signed

```
mov al, 0FFh
shr al, 1   ; /2
```

```
mov al, 5
shl al, 2   ; *4
```

```
mov al, 100
sal al, 1   ; *2
```

```
mov al, 5
sar al, 10 ; /1024
```

(q2) Solutions

- For each code fragment below, which are meant to multiply/divide values, say whether the numerical result is valid for signed and unsigned values

	unsigned	signed
`mov al, 0FFh` `shr al, 1 ; /2`	valid	invalid
`mov al, 5` `shl al, 2 ; *4`	valid	valid
`mov al, 100` `sal al, 1 ; *2`	valid	invalid
`mov ax, 5` `sar al, 10 ; /1024`	valid	valid

(q3) Rotate Shifts

- For each code fragment below, give the value of the register that was shifted

```
mov al, 0FFh
ror al, 3
```

```
mov al, 001h
ror al, 1
```

```
mov al, 00Fh
rol al, 7
```

(q3) Solutions

- For each code fragment below, give the value of the register that was shifted

```
mov al, 0FFh
ror al, 3
```

FF

```
mov al, 001h
ror al, 1
```

80

```
mov al, 00Fh
rol al, 7
```

87

(q4) Mystery Program

- What does this program do?

```
.segment data
  msg db "yes!",10,0

.segment text
  mov ebx, eax
  shr eax, 24
  cmp bl, al
  jne nope
  mov eax, msg
  call print_string
nope:
```

(q4) Solutions

- What does this program do?

```
.segment data
  msg db "yes!",10,0

.segment text
  mov ebx, eax
  shr eax, 24
  cmp bl, al
  jne nope
  mov eax, msg
  call print_string
nope:
```

- It says "yes!" if the 8 most significant bits of eax are identical to its 8 least significant bits

(q5) Mystery Program

- What does this program do?

```
  mov ebx, 0
  mov cl, 32
loop_begin:
  shl ebx, 1
  shr eax, 1
  adc ebx, 0
  dec cl
  cmp cl, 0
  jg loop_begin
```

(q5) Mystery Program

- What does this program do?

```
  mov ebx, 0
  mov cl, 32
loop_begin:
  shl ebx, 1
  shr eax, 1
  adc ebx, 0
  dec cl
  cmp cl, 0
  jg loop_begin
```

- It **reverses** the bits in eax and stores them in ebx
- x86 doesn't have a hardware instruction to do this, but ARM does! (RBIT)