

Review Questions

- ❖ Assume that the register `%rax` currently holds the value `0x 01 02 03 04 05 06 07 08`
- ❖ Answer the questions about the following instruction (`<instr> <src> <dst>`):

`xorw $-1, %ax`

- Operation type:
- Operand types:
- Operation width:
- (extra) Result in `%rax`:

4

Practice Question

- ❖ Which of the following are valid implementations of `rcx = rax + rbx`?

- `addq %rax, %rcx`
- `addq %rbx, %rcx`
- `movq %rax, %rcx`
- `addq %rbx, %rcx`
- `movq $0, %rcx`
- `addq %rax, %rcx`
- `xorq %rax, %rax`
- `addq %rax, %rcx`
- `addq %rbx, %rcx`

26

Arithmetic Example

```
long simple_arith(long x, long y)
{
    long t1 = x + y;
    long t2 = t1 * 3;
    return t2;
}
```

Register	Use(s)
<code>%rdi</code>	1 st argument (x)
<code>%rsi</code>	2 nd argument (y)
<code>%rax</code>	return value

```
y += x;
y *= 3;
long r = y;
return r;
```

```
simple_arith:
    addq    %rdi, %rsi
    imulq   $3, %rsi
    movq    %rsi, %rax
    ret
```

27

Understanding `swap()`

Registers

<code>%rdi</code>	<code>0x120</code>
<code>%rsi</code>	<code>0x100</code>
<code>%rax</code>	
<code>%rdx</code>	

Memory Word Address

123	0x120
	0x118
	0x110
	0x108
456	0x100

```
swap:
    movq    (%rdi), %rax    # t0 = *xp
    movq    (%rsi), %rdx    # t1 = *yp
    movq    %rdx, (%rdi)    # *xp = t1
    movq    %rax, (%rsi)    # *yp = t0
    ret
```

30

Complete Memory Addressing Modes

❖ General:

- $D(Rb, Ri, S) \quad \text{Mem}[\text{Reg}[Rb] + \text{Reg}[Ri] * S + D]$
 - Rb: Base register (any register)
 - Ri: Index register (any register except %rsp)
 - S: Scale factor (1, 2, 4, 8) – *why these numbers?*
 - D: Constant displacement value (a.k.a. immediate)

❖ Special cases (see CSPP Figure 3.3 on p.181)

- $D(Rb, Ri) \quad \text{Mem}[\text{Reg}[Rb] + \text{Reg}[Ri] + D] \quad (S=1)$
- $(Rb, Ri, S) \quad \text{Mem}[\text{Reg}[Rb] + \text{Reg}[Ri] * S] \quad (D=0)$
- $(Rb, Ri) \quad \text{Mem}[\text{Reg}[Rb] + \text{Reg}[Ri]] \quad (S=1, D=0)$
- $(, Ri, S) \quad \text{Mem}[\text{Reg}[Ri] * S] \quad (Rb=0, D=0)$

31

Address Computation Examples

%rdx	0xf000
%rcx	0x0100

$D(Rb, Ri, S) \rightarrow$
 $\text{Mem}[\text{Reg}[Rb] + \text{Reg}[Ri] * S + D]$
 ↑ ignore the memory access for now

Expression	Address Computation	Address (8 bytes wide)
$0x8(\%rdx)$		
$(\%rdx, \%rcx)$		
$(\%rdx, \%rcx, 4)$		
$0x80(, \%rdx, 2)$		

32