ICS Homework 7

November 1, 2022

1   Data Movement

You are given the following information. A function with prototype

```
1  void decode1(long *xp, long *yp, long *zp);
```

is compiled into assembly code, yield the following:

```
1    void decode1(long *xp, long *yp, long *zp)
2    xp in %rdi, yp in %rsi, zp in %rdx
3  decode1:
4    movq   (%rdi), %r8
5    movq   (%rsi), %rcx
6    movq   (%rdx), %rax
7    movq   %r8, (%rdx)
8    movq   %rcx, (%rdi)
9    movq   %rax, (%rsi)
```

Parameters xp, yp, and zp are stored in registers $\%rdi$, $\%rsi$, and $\%rdx$, respectively.

Write C code for $decode1$ that will have an effect equivalent to the assembly code shown.

```
1  void decode1(long *xp, long *yp, long *zp) {
2          long x = *xp;
3          long y = *yp;
4          long z = *zp;
5
6          *zp = x;
7          *xp = y;
8          *yp = z;
9  }
```

2   Arithmetic and Logical Operations

Suppose a 64-bit little endian machine has the following memory and register status:

Address	Value	Register	Value
0x100	0x0000000000002019	%rax	0x2121
0x108	0xffffffffaabb8922	%rbx	0x100
0x110	0x1212121212121212	%rcx	0x2
0x118	0x1300130013001300	%rdx	0x9

Each operation take effect on the status of memory and register, please fill in the blanks in the following table:

Operation	Destination	Value
subq (%rbx),%rax	%rax	0x108
incq -8(%rax)	0x100	0x000000000000201a
decq %rdx	%rdx	0x8
imulq $4,0x100(%rdx,%rcx,4)	0x110	0x4848484848484848
shrq $4,%rax	%rax	0x10
imulq 0x10	%rax, %rdx	0x100, 0x0
notw (%rax,%rdx)	0x100	0xdfe5
andq 0x10(%rax,%rcx,4),%rax	%rax	0x100
leaq 9(%rax,%rcx,8),%rdx	%rdx	0x119

3 Conditional Code

Indicate the status (0, 1 or unchanged) of the following flags after each instruction, please write "−" if the flag doesn't change. Assume 3 in %rax and −8 in %rbx.

NOTE: Each instruction works independently and would **NOT** affect each other.

Instruction	OF	CF	ZF	SF
addq %rbx, %rax	0	0	0	1
subq %rax, %rbx	0	0	0	1
leaq (%rax, %rax, 2), %rax	-	-	-	-
xorq %rax, %rax	0	0	1	0
salq $2, %rbx	0	1	0	1
cmpq %rax, %rbx	0	0	0	1
testq %rax, %rbx	0	0	1	0