1. (20) Fill in the blanks: Consider two instructions, which we’ll call i1 and i2, with i2 immediately following i1, and with i1 not being a jump of any kind. Then just before i1 is finished executing, the ___ will contain the address of ____. Just after i1 finishes, that address will be copied to the ____ bus. Assume no caches or instruction queues.

2. Consider the following code fragment:

```assembly
... jnz aplace
movl $0x7fffffff, %eax
aplace:
    movl $0x7fffffff, %ebx
    shll $2, %eax
    sall $2, %ebx
    addl %ecx,%edx
    ______________ ohhhhnoooo ...
```

(a) (15) Suppose, both here and in subsequent parts, that the offset of the first `movl`, listed in the output of `as -a`, turns out to be 0028. At what offset will the second `movl` begin?

(b) (15) In the output of `as -a` in assembling this code, what will be the machine language code generated for that second `movl`?

(c) (20) What will be the machine language code generated for `jnz aplace`?

(d) (15) In the instruction following the second `addl`, we’d like to jump to `ohhhhnoooo` if the last instruction produced a situation in which the sum of two positive numbers came out “negative.” List all possible instructions that we could put in the blank.

(e) (15) Suppose in running this code under GDB, we issue the commands

```bash
(gdb) b aplace
(gdb) run
(gdb) p/x %eip
```

Say the output of the last command is 0x80400000. Give a numerical expression (hex numbers are OK), for the memory address of the beginning of the `.text` segment.

**Solutions:**

1. PC; i2, address

2a. The instruction will assemble to 5 bytes, so the next offset will be 0x28 + 5 = 0x2d.

2b. bbffff7f

2c. 7505

2d. js or jo

2e. 0x80400000 - (28+5)