Ta

CSE 401 midterm winter 2019

February 12, 2019

1.

Assume that $s0 to $s4 hold inteers and may be used as indices or offsets. Assume that the base addresses of arrays A and B are in $s6 and $s7. $t registers are used to hold temporary values. What does this code do?

```
add $t0, $s6, $s0 — t0 = *~~B~~ +1 = A[1]
add $t1, $s7, $s1 — t1 = *B + 1 = B[1]
lw $s0, O($t0)    — s0 = *A[1] = content of A[1]
addi $t2, $t0, 4  — t2 = A[1]+4 = A[5] → A[2]
lw $t0, O($t2)    — t0 = *A[5] = content of A[2]
add $t0, $t0, $s0   t0 = 1 + 1 = 2 = 2
sw $t2, O($t1)      t2 = 1  - integer 1
```

This code stores three ints in $s0, $t0, and $t2. It initially starts by setting the index to store at $t0 and $t1, then stores the value at A[index] in $s0. It then →

0 ✓

2.

You are tasked with designing a CPU with 40-bit wordsize. You are using a RISC instruction set, similar to MIPS, with about 200 instruction codes. How many registers do you have? Why?

$2^7 = 128$ registers 2 ✓ Explanation on back →

3.

On a 32-bit MIPS machine, you need to execute the following single-precision floating point arithmetic instructions:

Fraction 23-bits, but since 1. Fraction, can hold ·24 bit unsigned int ·2

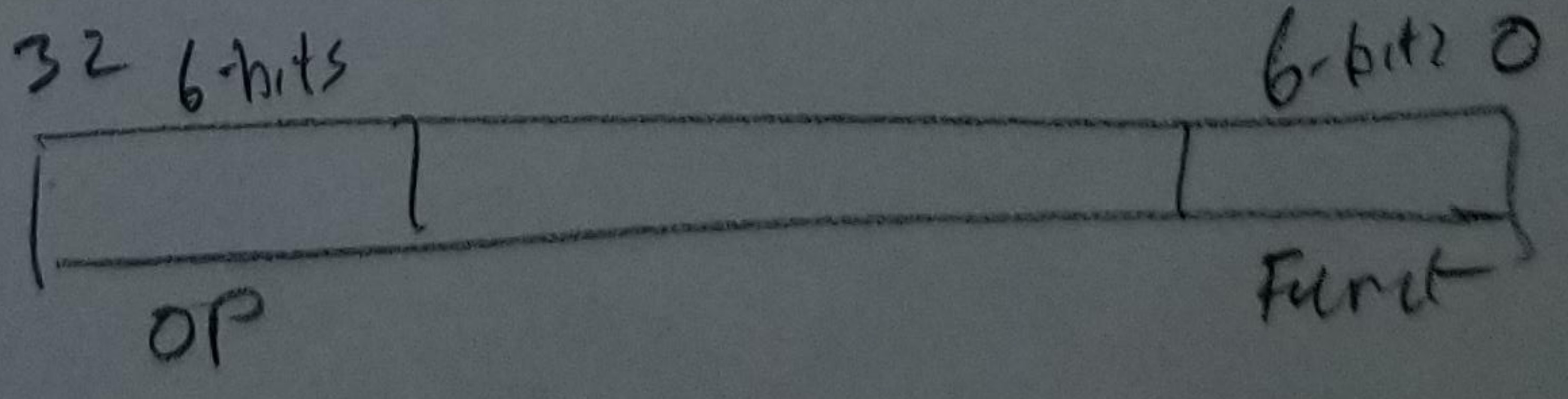
1. 3.5e25 * 3.5e25 / 12.0e20
2. 13.2e-30 * 3.5e25
3. 13.2e-20 / 3.5e25 + 5.7e8

Do any of the above operations overflow or underflow? If so, can they be rewritten so they work? Note: all the above expressions are in base 10, the notation eNNN means 10 raised to the NNN power.

Extra

In problem 1, what is the maximum number of instruction codes for this machine? Explain.

on back



0.0.0.0.0.0 $\cdot 10^{-13}$

$\cdot 10^{14}$

3.5

0.0,0,0.0,0,0,0.0,0.0,0,0.0.3.5 $\cdot 10^{12}$