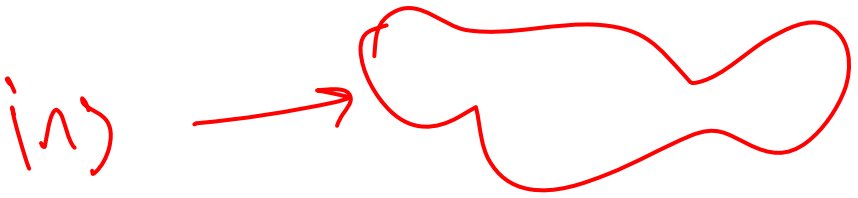
Review Problem 10

- How would the ALU be used to help with each of the following branches? The first is filled in for you:
  - beq ($rs == $rt) subtract $rt from $rs, use zero flag
  - bne ($rs != $rt) sub $rt from $rs, use !zero
  - bgez ($rs ≥ 0) $rs + $0, !negative flag
  - bgtz ($rs > 0) $rs + $0, !neg AND !zero.
  - blez ($rs ≤ 0) $rs + $0, (zero || negative)
  - bltz ($rs < 0) $rs + $0, neg flag.



Review Problem 11

- Design a 4-bit sra (shift arithmetic right) unit. Note that sra $t0, 1 = $t0/2, $t0, 2 = $t0/4, …

Review Problem 12

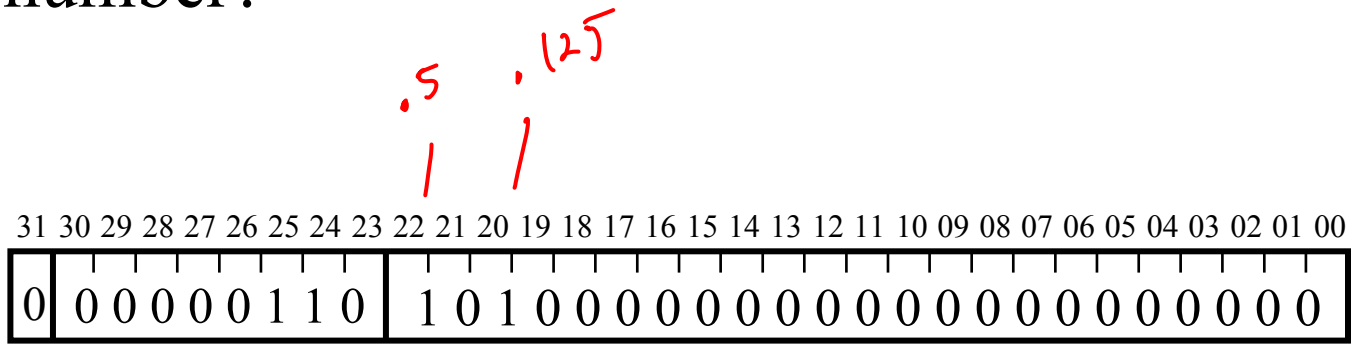
- Write MIPS assembly to compute $t1 = $t0*5 without using a multiply or divide instruction.

```
add  $t1, $t0, $t0      // t1 = t0 x 2
add  $t1, $t1, $t0      // t1 = t0 x 3
  ..
  ..
```

```
sll  $t1, $t0, 2        // t1 = t0 x 4
add  t1, t0, t1
```

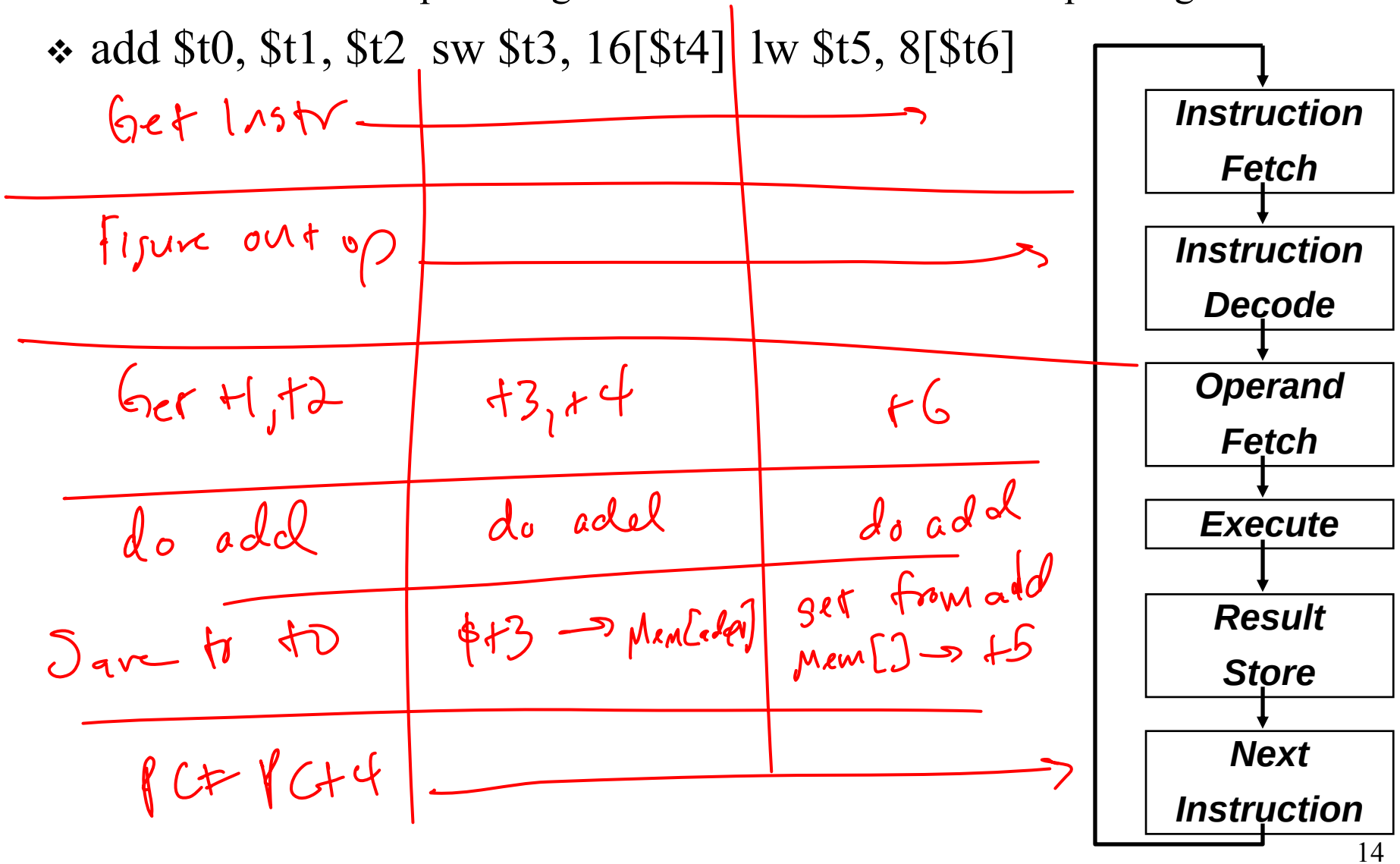
Review Problem 13

- What is the value of the following floating-point number?



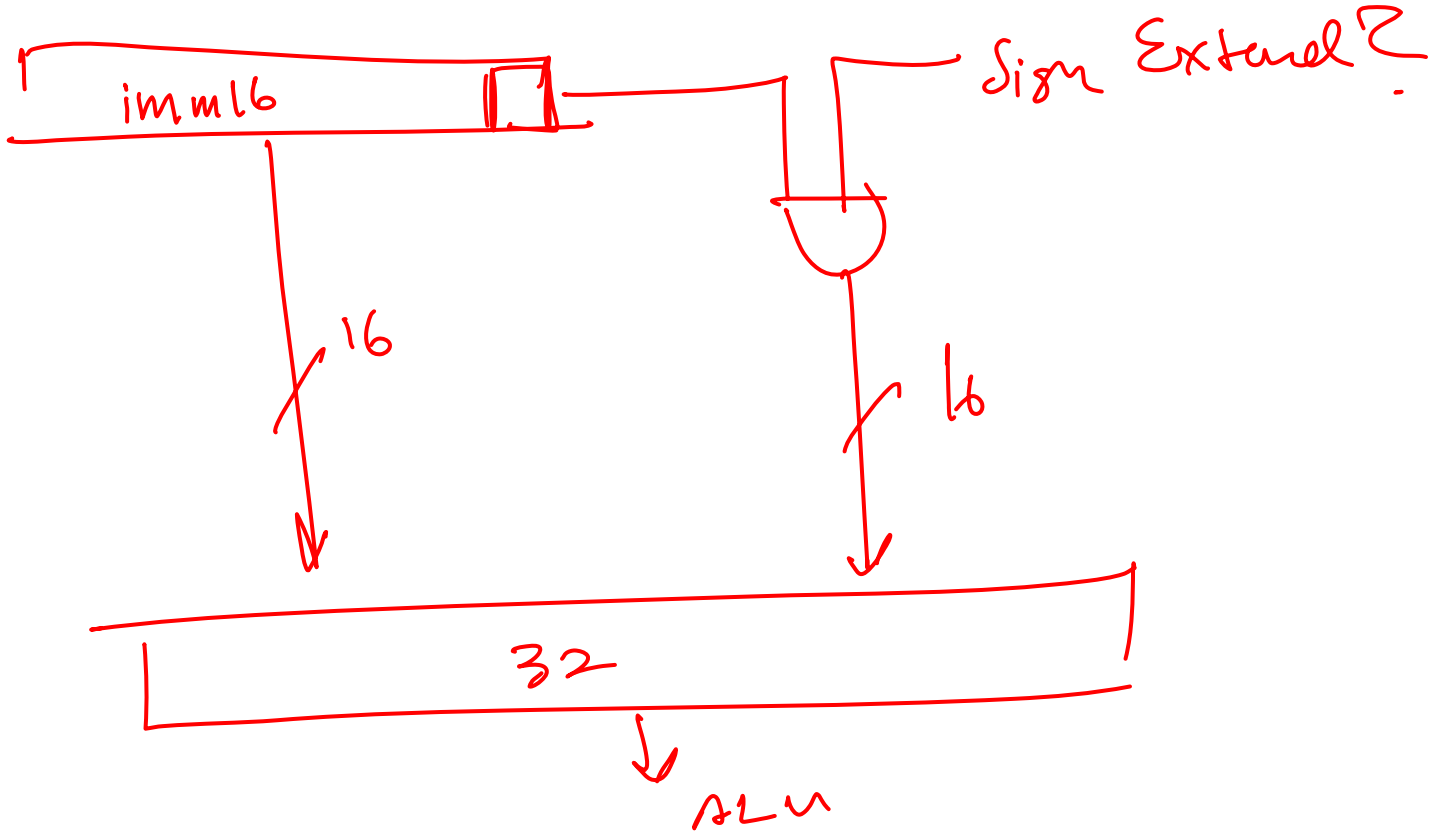
Review Problem 14

- What is done for these ops during each of the CPU's execute steps at right?
  - add $t0, $t1, $t2 sw $t3, 16[$t4] lw $t5, 8[$t6]



Review Problem 15

- Immediate vals for ADDI are sign-extended, while those for ORI are extended with zeros. Build a sign-extend unit that can handle both.



Review Problem 16

- Develop a single-cycle CPU that can do LW and SW (only). Make it as simple as possible