WANG Yuqi
[redacted]

1. Short Answer Questions

Q1. $(FD2)_{16} = (1111\ 1101\ 0010)_2$

$1111\ 11010010 - 1$

$= 1111\ 11010001$

Reverse: -000000101110

To decimal: $-(2^1 + 2^2 + 2^3 + 2^5)$

$= -46$

Q2. Sign: 1

Biased Exponent: 1000 1001 $\rightarrow 2^0 + 2^3 + 2^7 = (137)_{10}$

Significand: 111 1100 1101.0000 0000 0000

$E = 137 - 127 = 10$

∴ Shift Radix point to right by 10 bits:

$-1111\ 1100\ 110.1\ 0000\ 0000\ 0000$

$= -2022.5$

Q3.

1)
```
  0111
+ 1111
------
 ~~1~~0110
```
No overflow, because Signs are different

2)
```
  1110
+ 1000
------
 ~~1~~0110
```
Overflow.
Both operands are negative, but result is positive.

Q4. sltu $t3 $t2 $t1

Q5.
```
addi $sp, $sp, -8
sw $s1, 4($sp)
sw $s2, 0($sp)
```

2. MIPS: Translate Pseudo-instruction.

Q1.

```
sll $t0 $t1 25
srl $t1 $t1 7
or $t1 $t0 $t1
```

Q2.

```
sll $t1, $t2, 5
sub $t1, $t1, $t2
```

Q3.

```
lui $t0 0x0001
ori $t0 $0 0x0002
add $t0 $t1 $t0
lw $t4 0($t0)
```

3. Translate MIPS → C

```
if ((x>=y || z<=w) && x==z){
    if(y!=z) z=y-2
    else x=y+z
}
```

4. Understand MIPS Code

(1) continue;

(2)

$t0: stores memory address of currently visiting element of S (index iterator)
$t1: stores value of currently visiting element of S.
$v0: stores memory address of the largest element found in S so far
$v1: stores value of largest element found in S so far.

(3) move $v0, $t0:
- copy value stored in $t0 to $v0
- this essentially keep track of the address of the largest element visited.

move $v1, $t1:
- likewise, copy value stored in $t1 to $v1.
This essentially keep track of the value of the largest element visited.

(4) It checks if $t0 (address of currently visiting element) is NOT EQUAL to $a1 (address of the last element)

If TRUE: it jumps back to “loop”
If FALSE: it continues the code below.

This effectively checks whether we’ve reached the end of the array S. If not, we keep on looping. Otherwise, we end the loop.

(5) The program iterate through each element of the array S, and stores the address and value of the largest element in $v0, and $v1

(6) $v0 = 0x20060000 + 6x4
= 0x20060018
$v1 = 106