计算机组成原理

HW4

T1

(1)

$13.34375_{10} = 1101.01011_2 = 1.10101011 \times 2^3$

所以指数为 011 + 01111111 = 10000010，尾数为 10101011

浮点数表示为：`0 10000010 10101011000000000000000`

> 双精度！看题目！

(2)

9/(-4) = -2, (-9)/4 = -2, (-9)/(-4) = 2

9%(-4) = 1, (-9)%4 = -1, (-9)%(-4) = -1

(3)

$51_{10} = 00110011_2$，$5_{10} = 00000101_2$

相乘得 $11111111_2 = -1_{10}$

同理得：

100 * 8 = 01100100 * 00001000 = 00100000 = 32

51 * (-5) = 1

(-100) * 8 = -32

T2

(1)

lui x5, 0x00789;

addi x5, 0xabc;

> 0xabc 符号位拓展是负数!

h5 (2)

0xffffe297 为 auipc 指令，左移 12 位后的立即数为 -0x2000，所以 pc 变为 0x00001000， x5 变为 0x00001000

> PC 不变，3004!

h5 (3)

0x00c28067 为 jalr 指令，offset 为 12， rs1 为 x5，rd 为 x0，所以 pc 变为 0x0000100c

h5 (4)

代码如下:

```
slli x6, x6, 3
add x5, x5, x6
slli x7, x7, 2
add x5, x5, x7 #x5 = x5 + x6*8 + x7*4
lw x8, 0(x5)
```

实验题1

h5 (1)

代码如下:

```
slli x8, x6, 2
add x8, x5, x8
lw x9, 0(x8); #x9 = a[x6]
slli x10, x7, 2;
add x10, x5, x10;
lw x11, 0(x10) #x11 = a[x7]
ble x9, x11, done #if a[x6]<=a[x7], jump to done
sw x9, 0(x10) #a[x7] = x9
sw x11, 0(x8) #a[x6] = x11
done:
```

h5 (2)

两重循环，根据大小关系决定是否交换相邻元素，重复 99 次

```
BEGIN:
addi x13, x0, 0
addi x12, x0, 99
LOOP1:
beq x12, x13, END
# TODO (可自行添加标签)
addi x14, x0, 0
adddi x15, x0, 99
LOOP2:
# TODO (可自行添加标签)
add x6, x5, x14 #x6
addi x7, x6, 1 #x7
jal x1, SWAP #调用SWAP函数
addi x14, x14, 1
bne x14, x15, LOOP2
LOOP2END:
addi x13, x13, 1
jal x0, LOOP1
SWAP:
slli x8, x6, 2
add x8, x5, x8
lw x9, 0(x8); #x9 = a[x6]
slli x10, x7, 2
add x10, x5, x10
lw x11, 0(x10) #x11 = a[x7]
ble x9, x11, done #if a[x6]<=a[x7], jump to done
sw x9, 0(x10) #a[x7] = x9
sw x11, 0(x8) #a[x6] = x11
done: jalr x0, 0(x1) #返回被调用地址
END: nop
```