

## 题目 1. 计算题:

1. 将 13.34375 按照 IEEE 754 标准转换为双精度浮点数。
2. 计算  $9/(-4)$ 、 $(-9)/4$ 、 $(-9)/(-4)$ 、 $9\%(-4)$ 、 $(-9)\%4$ 、 $(-9)\%(-4)$ 。

注: 可以先根据 PPT 上的信息自己猜测, 再通过 C 代码进行验证。(事实上, 这些式子的计算结果是符合直觉的。)

3. 在 8 位 有符号数意义下, 计算  $51*5$ 、 $100*8$ 、 $51*(-5)$ 、 $(-100)*8$ , 要求用十进制写出结果 (更多拓展见思考题 1)。

$$1. \quad 13.34375_{10} = 1101.01011 = 1.10101011 \times 2^3$$

双精度数表示为  $(-1)^s \times (1 + \text{有效位数}) \times 2^{(\text{指数} - 1023)}$

因此该数按双精度应表示为: (省略号都是省略了 0)

$$\underbrace{1 \ 100 \dots 10}_{11 \text{ 位}} \underbrace{10101011 \ 000 \dots}_{52 \text{ 位}}$$

$$2. \quad \begin{array}{ll} 9 / -4 = -2 & 9 \% (-4) = 1 \\ (-9) / 4 = -2 & (-9) \% 4 = -1 \\ (-9) / (-4) = 2 & (-9) \% (-4) = -1 \end{array}$$

$$3. \quad \textcircled{1} \quad 51 \times 5 = -1 \quad (\text{产生溢出}) \quad \textcircled{2} \quad 100 \times 8 = 16 \quad (\text{产生溢出})$$

$$\begin{array}{r} 0010011 \\ \times 0000010 \\ \hline 0010011 \\ 0010011 \\ \hline 0010011 \\ \hline 11111111 \end{array}$$

$$\begin{array}{l} 01100100 \text{ 左移 3 位} \\ \Rightarrow 00100000 \end{array}$$

$$\textcircled{3} 51 \times (-5) = 1 \text{ (溢出)} \quad \textcircled{4} (-100) \times 8 = -32$$

00110011

10011100 左移3位

$\Rightarrow 11100000$

```

  x 1111011
  -----
    00110011
    0110011
    10011
    0011
    011
    11
    1
  -----
  00000001
  
```

**题目 2.** 指令题 (执行结果应写上所有影响的寄存器, 包含 pc 寄存器):

1. 将伪指令 `li x5, 0x00789abc` 翻译为两条真实指令的组合。
2. 当前 pc 在 `0x00003000` 位置, 执行指令 `0xffffe297` 的结果是什么?
3. 执行完上一问后, 下一条指令是 `0x00c28067`, 执行结果是什么?
4. 考虑数组 `int r[8][8]`, 假设 `int` 占四个字节, 数组基址在 `x5`, 请写出汇编程序, `x8 = r[x6][x7]`。

注: 不考虑越界, 其它寄存器任意使用。

1. `lui x5, 0x0078a` //由于addi指令立即数第11位为1, 会导致符号扩展为负数, 所以此处需要+1  
`addi x5, 0xabc`

2. 指令 `0xffffe297` 翻译成汇编指令为: `andip x5, 0xffffe`  
 取结果为, 将 pc 值与 `0xffffe000` 相加的结果 `0x00001000`  
 即 `x5 = 0x00001000`

3. 指令  $0x00c28067$  翻译成汇编指令为: `jalr x0, 12(x5)`  
故结果为跳转到  $x5 + 12$  的位置,  
即  $PC = 0x0000100C$

4. `addi x9, x0, 32`  
`mul x10, x6, x9`  
`addi x9, x0, 4`  
`mul x9, x7, x9`  
`add x9, x10, x9` //  $x9 = (x6) * 32 + (x7) * 4$   
`add x9, x5, x9` // 此时  $x9$  中保存了目标数据的地址  
`lw x8, 0(x9)`

**实验题 1.** 编译与执行 (本题只需要在提交的 pdf 中提供第二问的完整代码, 但建议自行验证结果。RARS 默认数据段开头为  $0x0000$ , 可直接 Execute 页面双击修改):

1. 假设 `int` 占四个字节, 数组 `int a[100]` 基址在 `x5`, 请写出代码使 `a[x6] > a[x7]` 时交换 `a[x6]` 与 `a[x7]` 的值, 否则不变。

注: 不要改变 `x5` 到 `x7` 的值, 可使用 `x8` 到 `x11`, 以方便下一问。

2. 如上问条件, 请在框架下写出汇编程序, 完成对 `a` 的冒泡排序 (从小到大)。

BEGIN:

`addi x13, x0, 0`

`addi x12, x0, 99`

LOOP1:

`beq x12, x13, END`

`# TODO (可自行添加标签)`

LOOP2:

`# TODO (可自行添加标签)`

LOOP2END:

`addi x13, x13, 1`

`jal x0, LOOP1`

SWAP:

`# TODO (利用上一问)`

END: `nop`

```
BEGIN:
    addi x13, x0, 0
    addi x12, x0, 99
```

```
LOOP1:
    beq x12, x13, END
    add x6, x0, x0
    addi x7, x6, 1
```

```
LOOP2:
    addi x8, x0, 4
    mul x9, x6, x8
    mul x10, x7, x8
    add x8, x5, x9
    add x9, x5, x10
    lw x10, 0(x8)
    lw x11, 0(x9)
    blt x11, x10, SWAP
```

#此时，x8,x9两个寄存器保存了 a[x6],a[x7]的地址，x10,x11两个寄存器保存了 a[x6],a[x7]的数值

```
RETURN: addi x6, x6, 1
    addi x7, x7, 1
    sub x11, x12, x13
    addi x11, x11, 1
    beq x7, x11, LOOP2END  #判断该轮遍历是否应该结束，从而进入下一轮
    jal x0, LOOP2
```

```
LOOP2END:
    addi x13, x13, 1
    jal x0, LOOP1
```

```
SWAP:
    sw x11, 0(x8)
    sw x10, 0(x9)
    jal x0, RETURN
```

```
END: nop
```