

Solution of Discussion03

DrinkLessMilkTea

2025 年 3 月 10 日

1 Floating Point

1.1

(a)

$$8.25 = 0b1000.01 = 0b1.00001 * 2^3$$

$$S = 0$$

$$Exp = 3 + 127 = 130 = 0b10000010$$

$$M = 0b00001$$

$$8.25 = 0b010000010000010...0 = 0x41040000$$

(b)

$$39.5625 = 0b100111.1001 = 0b1.001111001 * 2^5$$

$$S = 0$$

$$Exp = 5 + 127 = 132 = 0b10000100$$

$$M = 0b001111001$$

$$39.5625 = 0b0100001000011110010...0 = 0x423E4000$$

$$\begin{aligned}0x00000F00 &= 0000000000000000000011110...0 \\&= 00000000000000000000000011110...0\end{aligned}$$

$$Exp = 0$$

$$\begin{aligned} 0x00000F00 &= (-1)^0 \times 2^{-126} \times 0.000000000001111 \\ &= (2^{-12} + 2^{-13} + 2^{-14} + 2^{-15}) \times 2^{-126} \end{aligned}$$

$$0x00000000 = 0$$
$$0xFF94BEEF = 0b11111111001010010111110111011111$$

$$Exp = 0x FF$$

$$M \neq 0$$

(f)

$$= 0x7F800000$$

$$= 0x FF800000$$

(g) $\frac{1}{3}$ 无法被准确表达, 只能近似表达

2 More Floating Point

2.1

$2 = (-1)^0 \times 0b1.0 \times 2^1 = 0b0100000000..0$, 即 Exp 为 128, M 为 0, 则最小步长为 $2^{-23} \times 2^1 = 2^{-22}$, 所以下一个大于 2 的最小正数为 $2 + 2^{-22}$

2.2

同理, $4 = (-1)^0 \times 0b1.0 \times 2^2$, 即 Exp 为 129, M 为 0, 则最小步长为 $2^{-23} \times 2^2 = 2^{-21}$, 所以下一个大于 4 的最小正数为 $4 + 2^{-21}$

2.3

当步长大于 2^0 时, 之后的所有数都会是偶数, 因为之后的步长都为偶数, 所以最大的奇数是步长为 2^0 的最后一步

步长为 2^0 , 即 $(Exp - 127) \times 2^{-23} = 0$, 所以得 Exp 为 150, 所以最大的奇数为 $(1 + (1 - 2^{-23})) \times 2^{23} = 2^{24} - 1$, 从 2^{24} 开始之后的数都为偶数

3 RISC-V Instructions

3.1

已知寄存器 s0 保存的是数组的基地址

(a) 这条指令将基地址后 12 字节按字加载到寄存器 t0 中, 12 字节对应三个整数, 所以是把下标为 3 的数组元素 4 加载到寄存器 t0 中

(b) 这条指令将寄存器 t0 的值写入基地址后 16 字节, 即数组下标为 4 的位置, t0 的值为 4, 所以数组下标为 4 的元素更新为 4

(c) 首先是一条左移指令, 将 t0 中的数左移 2 位放入 t1 中, 此时 t0 为 4, 所以 t1 为 16

然后是一条加法指令, 将 t1 和 s0 中的值相加后放入 t2, 此时 s0 为基地址, 则 t2 保存的是基地址后 16 个字节的地址, 即下标为 4 的数组元素的地址

接着是加载指令, 将 t2 保存的地址加载到寄存器 t3 中, 此时 t3 为 4

然后又是一条加法指令, t3 寄存器自增 1, t3 寄存器为 5

然后是写入指令, 将 t3 寄存器的值写入 t2 指向的内存, 即数组下标为 4 的元素更新为 5

(d) 首先是一条加载指令, 将 s0 中的地址对应的字加载到 t0, 此时 t0 为 1

然后是一条异或指令, 将 t0 和 0xFFFF 异或后放入 t0, 即 0xFFFF 和 0x1 异或, 结果为 0FFE

最后是加法指令, t0 寄存器的值自增 1, 结果为 0xFFF

4 RISC-V Memory Access

4.1

t0: 0x00FF0004
t1: 36
t2: 0x0000000C
s0: 0xDEADB33F
s1: 0xFFFFFFF5

4.2

0xF9120504: 0xABADCAF8
0xBEEFDAB0: 0x00000000
0xABADCAFC: 0x0504DAB0
0xABADCAF8: 0xB0000400

5 Lost in Translation

5.1

```
li s0 4  
li s1 5  
li s2 6  
li s3 10
```

```
add s3 s3 s0
add s3 s3 s1
add s3 s3 s2
```

```
sw x0 0(s0)
li s1 2
sw s1 4(s0)
sw s1 8(s0)
```

```
li s0 5
li s1 10
add s2 s1 s1
bne s2 s1 ELSE
li s0 0
j EXIT
ELSE:
    sub s1 s0 1
EXIT:
```

```
li s0 0
li s1 1
li s2 30
LOOP:
    beq s0 s2 EXIT
    add s1 s1 s1
    addi s0 s0 1
    j LOOP
EXIT:
```

```
    li s1 0
LOOP:
    bge x0 s0 EXIT
    add s1 s1 s0
    sub s0 s0 1
    j LOOP
EXIT:
```