

计算机组成原理

作业三

T1

“

假设寄存器 t0 中初始状态下保存的值为 0x00002023。请回答下面的问题:

- ① 对于指令 `sub t2, t0, t1`, 导致结果溢出的 t1 的值的范围?
- ② 假设 PC (程序计数器) 当前值为 0x0D000000, 则 `jal` 指令可以到达的地址范围是多少? 如果是 `blt` 呢?

h5 (1)

t0 为正数, 所以不可能是下溢出。

若为上溢出, 则 $t1 = -(0x7fffffff - 0x00002023) - 1 = 0x80002023$

所以 t1 范围为 $[0x80000000, 0x80002023]$

h5 (2)

`jal` 指令立即数有 20 位, bit0 默认为 0, 范围为 $[0xffff0000, 0x000ffffc]$, 所以可到达 $0x0CF00000 \sim 0x0D0FFFFE$

`blt` 指令立即数有 12 位, bit0 默认为 0, 范围为 $[0xffff000, 0x00000ffe]$, 所以可到达 $0x0CFFF000 \sim 0x0D000FFE$

注: 跳转是非连续的, 按两字节对齐。上面只是表示范围, 范围内的地址不一定能直接到达。

T2

“

本题目中所述的 int 整型变量都是 32 位的。

- ① 将 $-a * 2 - (b + c) - (d + b + c) + 200$ 转换为 RV32I 指令 (a, b, c, d 均为 int 整型数值, 且已经分别保留在寄存器 t0, t1, t2, t3 中。不考虑溢出问题)。请尝试使用尽可能少的寄存器和尽可能少的指令。

- ② 将 $A[2 * j] = B[i - 8]$ 转换为 RV32I 指令，其中 A, B 为 int 整型数组。它们的基址分别保存在寄存器 a0, a1 中。i, j 均为 int 整型变量，且已保存在寄存器 t0, t1 中（不考虑溢出以及非法访问问题，所有数据都已经四字对齐）。尽可能为你的指令添加注释，使得助教能够更好地理解你的代码。

h5 (1)

```

1 add t0, t0, t0; #t0=a*2=a+a
2 add t1, t1, t2; #t1=b+c
3 add t2, t1, t3; #t2=d+b+c=d+t1
4 andi t3, t3, x0; #t3=0
5 add t3, t3, t0; #t3=-a*2
6 add t3, t3, t1; #t3=-a*2-(b+c)
7 add t3, t3, t2; #t3=-a*2-(b+c)-(d+b+c)
8 addi t3, t3, 200; #t3=-a*2-(b+c)-(d+b+c)+200

```

h5 (2)

```

1 subi t0, t0, 8; #t0=i-8
2 add t1, t1, t1; #t1=2*j
3 slli t0, t0, 2; #t0=t0*4
4 slli t1, t1, 2; #t1=t1*4
5 add a0, a0, t1; #a0=&A[2*j]
6 add a1, a1, t0; #a1=&B[i-8]
7 lw t2, 0(a1); #t2=B[i-8]
8 sw t2, 0(a0); #A[2*j]=B[i-8]

```

T3

“

我们知道 RISC-V 存储是小端序的，即低地址存储低位，高地址存储高位。阅读如下代码：

```

1 lb t1, 1(t0);
2 sw t1, 4(t0);

```

初始条件下，t0 的内容为 0x2023，地址 0x2023 的内容为 0x20881124。请问：

- ① 该代码执行后，地址 0x202A 的内容是什么？
- ② 如果 RISC-V 是大端序存储的，那么该代码执行后，地址 0x202A 的内容是什么？

h5 (1)

```
t1 = 0x00000011
```

```
M[0x2027] = 0x00000011
```

```
M[0x202A] = 0x00
```

h5 (2)

```
M[0x202A] = 0x88
```

T4

“

现在我们需要使用 RV32I 指令求解斐波那契数列的前 n 项，其中 n 为 int 整型变量，保存在内存地址 `place` 中。斐波那契数列的第一项和第二项分别保存在内存地址 `first` 和 `second` 中。请根据以上信息编写 RV32I 指令，将从第一项开始的结果依次保存在从内存地址 `save` 开始的连续内存中。

```

1  la t0, place; #t0=place
2  lw t1, 0(t0); #t1=n
3  beqz t1, done; #n==0? done: keep
4  la t2, save; #t2=save
5  la t3, first; #t3=first
6  lw t3, 0(t3); #t3=f(1)
7  sw t3, 0(t2); #save f(1)
8  subi t1, t1, 1; #n--
9  beqz t1, done; #n==1? done: keep
10 addi t2, t2, 4; #指向下个储存位置
11 la t4, second; #t4=second
12 lw t4, 0(t4); #t4=f(2)
13 sw t4, 0(t2); #save f(2)
14 subi t1, t1, 1; #n--
15 beqz t1, done; #n==2? done: keep
16 loop: add t4, t4, t3; #t4=f(n)=f(n-1)+f(n-2)
17 sub t3, t4, t3; #t3=f(n-1)
18 addi t2, t2, 4; #指向下个储存位置
19 sw t4, 0(t2); #save f(n)
20 subi t1, t1, 1; #n--
21 bgtz t1, loop; #n>0? keep: done
22 done: #结束程序

```

实验题1

已提交