

COD WEEK3

1. 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

1.1 1

对于指令 `sub t2, t0, t1`, 如果结果溢出, 说明结果超出了32位有符号数的表示范围。

t1本身范围: $t1 \in [-2^{31}, 2^{31} - 1]$

设t0为x, 则 $t2=t0-t1=x-t1$ 的范围也是: $[-2^{31}, 2^{31} - 1]$

所以不溢出的范围是 $t1 \in [x + 1 - 2^{31}, x + 2^{31}]$

所以溢出范围: $t1 \in [-2^{31}, x - 2^{31}]$

t0 的值为 0x00002023

因此, 溢出范围应该是 $[0x80000000, 0x80002023]$

1.2 2

`jal` 指令, 立即数20位, 2^{20} , 第0位默认为0, 只能跳转到偶数地址, 因此, `jal` 指令可以到达的地址范围为 $[0x0CF00000, 0x0D0FFFFE]$ 的偶数地址

`blt` 指令, 立即数12位, 同上, 就是 $[0x0CFFF000, 0x0D000FFE]$ 的偶数地址

2. 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节对齐）。

2.1 1

```
1  .text
2  #计算 -a * 2, 结果保存在寄存器 t0 中
3  slli t0, t0, 1
4  sub t0, zero, t0    #t0 = -t0
5
6  #计算 b + c, 结果保存在寄存器 t1 中
7  add t1, t1, t2
8
9  #计算 d + b + c + b + c, 结果保存在寄存器 t2 中
10 add t2, t3, t1
11 add t2, t2, t1
12
13 #计算 -a * 2 - (b + c) - (d + b + c) + 200, 结果保存在寄存器 t3 中
14 addi t3, t0, 200
15 sub t3, t3, t2
16
```

2.2 2

```
1  .text
2  #计算数组索引 i - 8 的地址, 结果保存在寄存器 t0 中
3  addi t0, t0, -8    #i-8
4  slli t0, t0, 2     #4*[i-8], int占4字节
5  add t0, a1, t0
6
7  #计算数组索引 2 * j 的地址, 结果保存在寄存器 t1 中
8  slli t1, t1, 2
9  slli t1, t1, 2
10 add t1, a0, t1
11
12 #从内存中读取 B[i - 8] 的值, 保存在寄存器 t2 中
13 lw t2, 0(t0)
```

```
14
15 #将 B[i - 8] 的值写入 A[2 * j], 保存在内存中
16 sw t2, 0(t1)
17
```

3. 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：

```
lb t1, 1(t0);
```

```
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？

3.1 1

lb 读取 0x11

sw 把 t1 存储在 4(t0)，由于是小端序存储，所以 0x202A 存储 0

$M[0x202A] = 0$

3.2 2

lb 读取 0x88 到 t1

sw 把 t1 存储在 4(t0)，由于是大端序存储，所以 0x202A 存储 0x88

$M[0x202A] = 0x88$

4. 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  .text
2  # 将n加载到寄存器t0中
3  lw t0, 0(place)
4  # 将第一项和第二项加载到寄存器t1和t2中
5  lw t1, 0(first)
6  lw t2, 0(second)
7  # 将第一项保存到内存中
8  sw t1, 0(save)
9  # 将第二项保存到内存中
10 sw t2, 4(save)
11 # 初始化计数器i为2
12 li t3, 2
13 loop:
14     # 如果i >= n, 跳出循环
15     bge t3, t0, end
16     # 计算下一项的值
17     add t4, t1, t2
18     # 保存下一项的值到内存中
19     slli t5, t3, 2          #4*i
20     add t5, t5, save
21     sw t4, 0(t5)
22     # 更新t1和t2
23     add t1, zero, t2
24     add t2, zero, t4
25     # 更新计数器i
26     addi t3, t3, 1
27     # 继续循环
28     jal zero loop
29 end:
30
```

5. 实验题1

已提交在bb系统中