

中国科学技术大学计算机学院 《计算机系统概论》实验报告



RISC-V LAB 02 : Greatest Common Divisor

姓名:王晨 学号:PE20060014

完成日期:2020 年 12 月 18 日

一、实验要求：

用RISC-V指令集实现求两个正整数的最大公因数GCD，用t1和t2保存待计算的两个正整数，求出的结果放在t1中。

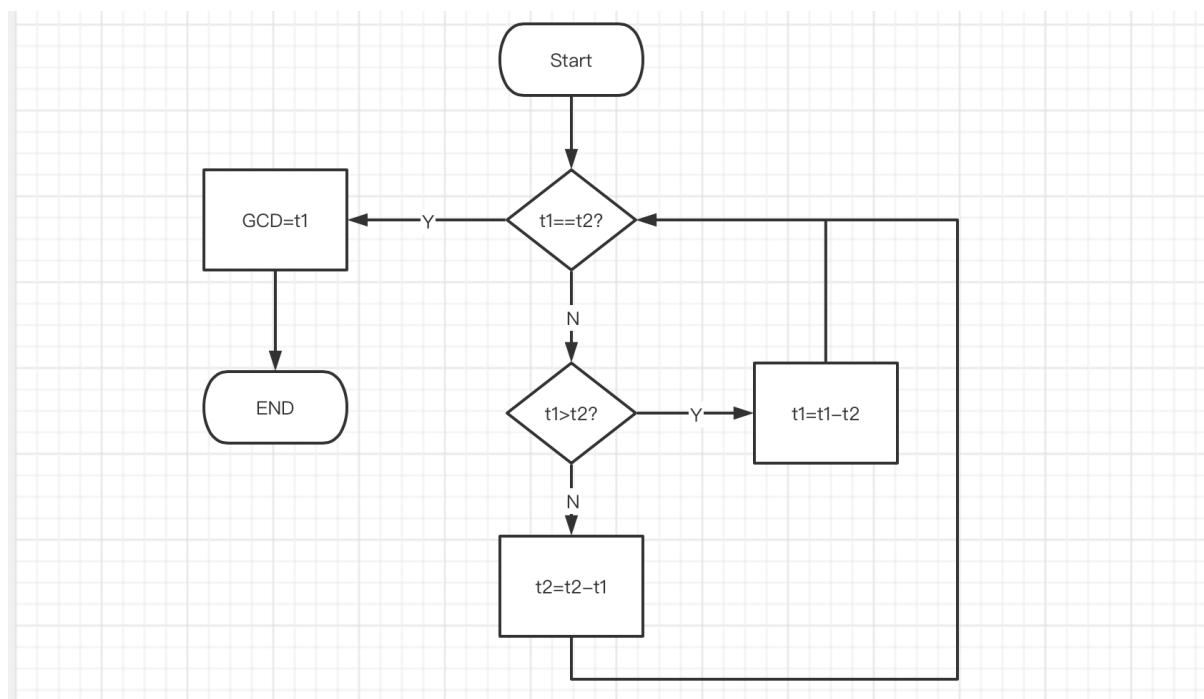
二、实验环境

Mac OS Big Sur 11.1

Rars risc-v Simulator

三、算法思路：

同实验2一样，还是使用更相减损法求GCD，算法流程图如下：



不过由于RISC-V直接提供DIV除法指令，借此可以实现除法和求余，因此本实验还可以使用辗转相除法，这样效率会更高。

四、程序代码及注释：

求GCD结果放在t1寄存器中，程序非常简单，注释已写上，就不再赘述了。

① .data

② hexNum1:

③ .word 0x00000104

- ④ hexNum2:
- ⑤ .word 0x00000068
- ⑥
- ⑦ .text
- ⑧ .global _start # Provide program starting address to linker
- ⑨ _start:
- ⑩ lw t1,hexNum1 # Load 2 hexNumber into t0 and t1
- ⑪ lw t2,hexNum2
- ⑫ CHECKEQL: beq t1,t2,DONE # return if t0 = t1
- ⑬ blt t1,t2,MINUS # t2=t2-t1 if t1 < t2
- ⑭ sub t1,t1,t2 # t1=t1-t2 if t1 > t2
- ⑮ b CHECKEQL
- ⑯ MINUS: sub t2,t2,t1
- ⑰ b CHECKEQL
- ⑱ DONE: ret # return and t1 saves the result GCD

```

1  .data
2  hexNum1:
3      .word 0x00000104
4  hexNum2:
5      .word 0x00000068
6
7  .text
8  .global _start          # Provide program starting address to linker
9  _start:
10     lw t1,hexNum1       # Load 2 hexNumber into t0 and t1
11     lw t2,hexNum2
12     CHECKEQL:          beq t1,t2,DONE # return if t0 = t1
13                        blt t1,t2,MINUS # t2=t2-t1 if t1 < t2
14                        sub t1,t1,t2    # t1=t1-t2 if t1 > t2
15                        b CHECKEQL
16     MINUS:              sub t2,t2,t1
17                        b CHECKEQL
18     DONE:               ret          # return and t1 saves the result GCD

```

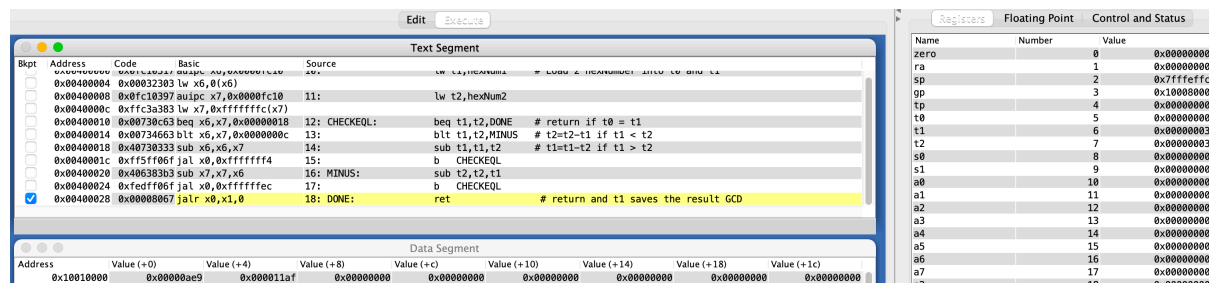
五、测试用例和结果分析：

1.t1=x104=260,t2=x68=104,从输出结果可以看到GCD=t1=0x34=52, t2=52,结果正确。

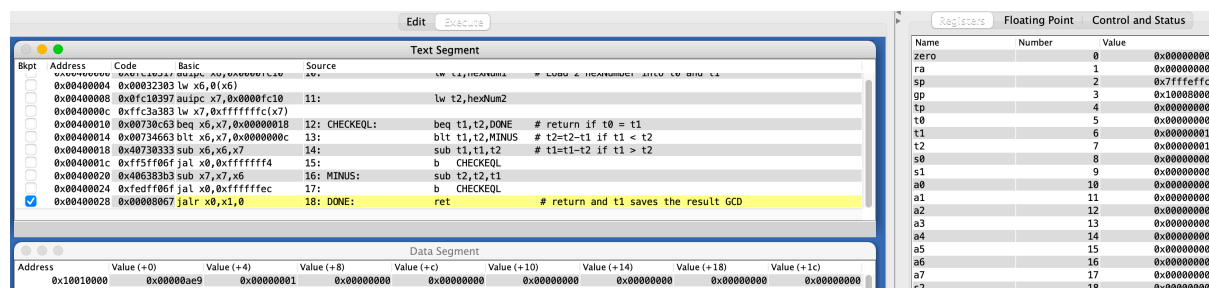
Text Segment		Registers	
Bkpt	Address Code Basic Source	Name	Value
	0x00400000 0x0fc18317 auipc x6,0x0000fc10 10: lw t1,hexNum1 # Load 2 hexNumber into t0 and t1	zero	0
	0x00400004 0x00032383 lw x6,0(x6)	ra	1
	0x00400008 0x0fc18397 auipc x7,0x0000fc10 11: lw t2,hexNum2	sp	2
	0x0040000c 0xffc3a383 lw x7,0xffffffc(x7)	gp	3
	0x00400010 0x00738c63 beq x6,x7,0x00000018 12: CHECKEQL: beq t1,t2,DONE # return if t0 = t1	tp	4
	0x00400014 0x00734663 blt x6,x7,0x0000000c 13: blt t1,t2,MINUS # t2=t2-t1 if t1 < t2	t0	5
	0x00400018 0x40738333 sub x6,x6,x7 14: sub t1,t1,t2 # t1=t1-t2 if t1 > t2	t1	6
	0x0040001c 0xf151f06f jal x0,0xfffffff4 15: b CHECKEQL	t2	7
	0x00400020 0x406383b3 sub x7,x7,x6 16: MINUS: sub t2,t2,t1	s0	8
	0x00400024 0xfedff06f jal x0,0xfffffff4 17: b CHECKEQL	s1	9
	0x00400028 0x00000000 ret 18: DONE: ret # return and t1 saves the result GCD	a0	10
		a1	11
		a2	12
		a3	13
		a4	14
		a5	15
		a6	16
		a7	17
		s2	18

Data Segment	
Address	Value (+0)
0x10010000	0x00000104
0x10010004	0x00000068

2. 随机尝试两个较大的数，例如 $t1=xae9=2793$ ， $t2=x11af=4527$ ， $GCD=t1=3$ ，结果正确。



3. $t1=$ 任意值， $t2=1$ ，输出结果 $t0=1$ ，结果正确。



六、实验心得：

本实验使用risc-v指令集实现求两个正整数的GCD，实验成功。通过risc-v丰富的Branch跳转指令，将比较和跳转用一条指令完成，因此相对LC-3指令集来说，程序大大简化，对每个分支，只需要写1~2条指令就可以完成。但是由于使用了更相减损法，若两个数字相差很大，仍然需要大量减法和比较才能得到最终结果。

因此如果追求更高的计算效率，可以采用带除2的更相减损法，或者辗转相除法，由于期末时间有限，就不再写程序比较了。