



哈爾濱工業大學 (深圳)
HARBIN INSTITUTE OF TECHNOLOGY

实验报告

开课学期: 2022 春季

课程名称: 计算机组成原理 (实验)

实验名称: 从 C 语言到机器码

实验性质: 综合设计型

实验学时: 2 地点:

学生班级: 20 级计科 8 班

学生学号: 200210231

学生姓名: 王木一

作业成绩:

实验与创新实践教育中心制

2022 年 3 月

1、实验结果截图

(需贴出执行文件运行的结果截图)

```
comp2008@comp2008:~/lab1$ riscv64-unknown-elf-gcc lab1.c -o lab1
comp2008@comp2008:~/lab1$ spike pk lab1
bbl loader
my ID:200210231
29791
```

2、汇编代码注释(只需写主程序和子程序即可)

示例: `addi sp,sp,-16`

将堆栈指针寄存器 `sp` 与立即数(-16)相加, 再存入堆栈指针寄存器 `sp`, 即 $sp = sp + (-16)$

注: 本次实验主程序为 `main`, 其中首先打印学号, 再调用 `cube()` 函数计算立方。以下注释为将.s 文件拷贝到 `rars` 软件中进行编辑而得。为计算立方, 使用两次原码一位乘法, 故汇编指令中有两次循环。

```
1      .file "lab1.c"
2      .option nopic
3      .attribute arch, "rv64i2p0_m2p0_a2p0_f2p0_d2p0_c2p0"
4      .attribute unaligned_access, 0
5      .attribute stack_align, 16
6      .text
7      .section .rodata
8      .align 3
9  .LC0:
10     .string "%d\n"
11     .text
12     .align 1
13     .globl cube
14     .type cube, @function
15 cube:
16     addi sp,sp,-48      # 设置栈指针: sp = sp + (-48), 设置栈指针sp, 将栈指针寄存器sp与立即数-48相加, 再写入sp
17     sd ra,40(sp)       # 入栈: 将返回地址ra寄存器中的值存入地址为R[sp]+40的内存中
18     sd s0,32(sp)       # 入栈: 将s0寄存器中的值存入地址为R[sp]+32的内存中
19     addi s0,sp,48      # 立即数加: s0 = sp + 48, 栈指针寄存器sp与立即数48相加, 再存入s0, 即s0的值为起始sp的值
20     li a5,31           # 加载立即数: a5 = 31, 将立即数31写入a5
21     sw a5,-20(s0)      # 存word数: 将a5寄存器的值(31)存入地址为R[s0]-20的内存中(即cube
22     li a5,31           # 加载立即数: a5 = 31, 将立即数31写入a5
23     sw a5,-24(s0)      # 存word数: 将a5寄存器的值(31)存入地址为R[s0]-24的内存中(即sqrt
24     li a5,8192         # 加载立即数: a5 = 8192, 将立即数8192写入a5
25     addi a5,a5,-256    # 立即数加: a5 = a5 + (-256), 将a5与立即数-256相加, 再写入a5
26     sw a5,-36(s0)      # 存word数: 将a5寄存器的值(8192-256)存入地址为R[s0]-36的内存中(即mul
27     li a5,7            # 加载立即数: a5 = 7, 将7写入a5
28     sw a5,-28(s0)      # 存word数: 将a5寄存器的值(7)存入地址为R[s0]-28的内存中(即循环计数i
29     j .L2             # 无条件跳转: 直接跳转到.L2标志, 未存返回地址
```

```

30 .L4:
31     lw     a4, -24(s0)          # 取word数: 将存储在地址R[s0]-24的内存中数取出, 写入a4 (取sqrt
32     sraiw  a5, a4, 31          # 立即数算术右移word: 将a4中的值算术右移31位, 结果截断为32位, 写入a5
33     srlw   a5, a5, 31          # 立即数逻辑左移word: 将a5中的值逻辑左移31位, 结果截断为32位, 写入a5
34     addw   a4, a4, a5          # word加: a4 = a4 + a5, 将a4中的值与a5中的值相加, 结果截断为32位, 写入a4
35     andi   a4, a4, 1           # 立即数与: a4 = a4 & 1, 将a4与立即数1按位与, 再写入a4
36     subw   a5, a4, a5          # word减: a5 = a4 - a5, 将a4中的值与a5中的值相加减, 结果截断为32位, 写入a5
37     sext.w a5, a5              # word数符号扩展: 将a5中的数符号扩展到64位. 写入a5
38     mv     a4, a5              # 复制数: a4 = a5, 将a5的值写入a4 (以上31-38行指令都是计算乘数的最后一位)
39     li     a5, 1               # 加载立即数: a5 = 1, 将立即数1写入a5
40     bne    a4, a5, .L3         # 分支: 若a4中的数不等于a5中的数, 就跳转到.L3
41     lw     a4, -24(s0)          # 取word数: 将存储在地址R[s0]-24的内存中数取出, 写入a4
42     lw     a5, -36(s0)          # 取word数: 将存储在地址R[s0]-36的内存中数取出, 写入a5
43     addw   a5, a4, a5          # word加: a5 = a4 + a5, 将a4中的值与a5中的值相加, 结果截断为32位, 写入a5 (原码一位乘的相加)
44     sw     a5, -24(s0)          # 存word数: 将a5寄存器的值存入地址为R[s0]-24的内存中
45 .L3:
46     lw     a5, -24(s0)          # 取word数: 将存储在地址R[s0]-24的内存中数取出, 写入a5
47     sraiw  a5, a5, 1           # 立即数算术右移word: 将a5中的值算术右移1位, 结果截断为32位, 写入a5 (原码一位乘的右移)
48     sw     a5, -24(s0)          # 存word数: 将a5寄存器的值存入地址为R[s0]-24的内存中
49     lw     a5, -28(s0)          # 取word数: 将存储在地址R[s0]-28的内存中数取出, 写入a5
50     addiw  a5, a5, -1          # 加立即数word: a5 = a5 + (-1), 将a5与立即数-1相加, 结果截断32位, 再写入a5 (for循环i的更新)
51     sw     a5, -28(s0)          # 存word数: 将a5寄存器的值存入地址为R[s0]-28的内存中
52 .L2:
53     lw     a5, -28(s0)          # 取word数: 将存储在R[s0]-28的内存中数取出, 写入a5
54     sext.w a5, a5              # word数符号扩展: 将a5中的数符号扩展到64位. 写入a5
55     bge    a5, zero, .L4       # 分支: 若a5中的数大于等于0, 就跳转到.L4
56     lw     a5, -24(s0)          # 取word数: 将存储在地址R[s0]-24的内存中数取出, 写入a5
57     sllw   a5, a5, 8           # 立即数逻辑左移word: 将a5中的值算术左移8位, 结果截断32位, 写入a5
58     sw     a5, -24(s0)          # 存word数: 将a5寄存器的值存入地址为R[s0]-24的内存中
59     li     a5, 7               # 加载立即数: a5 = 7, 将立即数7写入a5
60     sw     a5, -32(s0)          # 存word数: 将a5寄存器的值存入地址为R[s0]-32的内存中
61     j      .L5                 # 无条件跳转: 直接跳转到.L5标志, 未存返回地址
62 .L7:
63     lw     a4, -20(s0)          # 取word数: 将存储在地址R[s0]-32的内存中数取出, 写入a4 (取cube
64     sraiw  a5, a4, 31          # 立即数算术右移word: 将a4中的值算术右移31位, 结果截断为32位, 写入a5
65     srlw   a5, a5, 31          # 立即数逻辑左移word: 将a5中的值逻辑左移31位, 结果截断为32位, 写入a5
66     addw   a4, a4, a5          # word加: a4 = a4 + a5, 将a4中的值与a5中的值相加, 结果截断为32位, 写入a4
67     andi   a4, a4, 1           # 立即数与: a4 = a4 & 1, 将a4与立即数1按位与, 再写入a4
68     subw   a5, a4, a5          # word减: a5 = a4 - a5, 将a4中的值与a5中的值相加减, 结果截断为32位, 写入a5
69     sext.w a5, a5              # word数符号扩展: 将a5中的数符号扩展到64位. 写入a5
70     mv     a4, a5              # 复制数: a4 = a5, 将a5的值写入a4 (以上63-70行指令都是计算乘数的最后一位)
71     li     a5, 1               # 加载立即数: a5 = 1, 将立即数1写入a5
72     bne    a4, a5, .L6         # 分支: 若a4中的数不等于a5中的数, 就跳转到.L6
73     lw     a4, -20(s0)          # 取word数: 将存储在地址R[s0]-20的内存中数取出, 写入a4
74     lw     a5, -24(s0)          # 取word数: 将存储在地址R[s0]-24的内存中数取出, 写入a5
75     addw   a5, a4, a5          # word加: a5 = a4 + a5, 将a4中的值与a5中的值相加, 结果截断为32位, 写入a5 (原码一位乘的相加)
76     sw     a5, -20(s0)          # 存word数: 将a5寄存器的值存入地址为R[s0]-20的内存中
77 .L6:
78     lw     a5, -20(s0)          # 取word数: 将存储在地址R[s0]-20的内存中数取出, 写入a5
79     sraiw  a5, a5, 1           # 立即数算术右移word: 将a5中的值算术右移1位, 结果截断为32位, 写入a5 (原码一位乘的右移)
80     sw     a5, -20(s0)          # 存word数: 将a5寄存器的值存入地址为R[s0]-20的内存中
81     lw     a5, -32(s0)          # 取word数: 将存储在地址R[s0]-28的内存中数取出, 写入a5
82     addiw  a5, a5, -1          # 加立即数word: a5 = a5 + (-1), 将a5与立即数-1相加, 结果截断32位, 再写入a5 (for循环i的更新)
83     sw     a5, -32(s0)          # 存word数: 将a5寄存器的值存入地址为R[s0]-32的内存中
84 .L5:
85     lw     a5, -32(s0)          # 取word数: 将存储在地址R[s0]-32的内存中数取出, 写入a5
86     sext.w a5, a5              # word数符号扩展: 将a5中的数符号扩展到64位, 写入a5
87     bge    a5, zero, .L7       # 分支: 若a5中的数大于等于0, 就跳转到.L7
88     lw     a5, -20(s0)          # 取word数: 将存储在地址R[s0]-20的内存中数取出, 写入a5 (取最后的结果)
89     mv     a1, a5              # 复制数: a1 = a5, 将a5的值写入a1
90     lui    a5, %hi(.LC0)       # a5 = 0 + %hi(.LC0), 将.LC0标志高20位地址加到a5
91     addi   a0, a5, %lo(.LC0)    # a0 = a5 + %lo(.LC0), .LC0标低12位地址加到a5, 写入a0
92     call   printf              # 跳转执行printf函数, 打印立方值
93     nop                          # 无操作
94     ld     ra, 40(sp)          # 出栈: 取出原来的返回地址 (回到main) 到ra
95     ld     s0, 32(sp)          # 出栈: 取出原来的s0寄存器值到s0
96     addi   sp, sp, 48          # 恢复栈指针: sp = sp + 48, 恢复栈指针sp
97     jr     ra                  # 无条件跳转: 跳转到ra中存的地址 (回到main)
98 .size    cube, .-cube
99 .section .rodata
100 .align 3

```

```

101 .LC1:
102     .string "my ID:200210231"
103     .text
104     .align 1
105     .globl main
106     .type main, @function
107 main:
108     addi    sp,sp,-16           # 设置栈指针: sp = sp + (-16), 将栈指针寄存器sp与立即数-16相加, 再写入sp
109     sd      ra,8(sp)           # 入栈: 将返回地址, 即ra中的值存入地址为R[sp]+8的内存中
110     sd      s0,0(sp)           # 入栈: 将s0寄存器中的值存入地址为R[sp]+0的内存中
111     addi    s0,sp,16           # 立即数加: s0 = sp + 16, 栈指针寄存器sp与立即数16相加, 再写入s0, 即s0的值为起始sp的值
112     lui     a5,%hi(.LC1)       # a5 = 0 + %hi(.LC1), 将.LC1标志高20位地址加到a5
113     addi    a0,a5,%lo(.LC1)    # a0 = a5 + %lo(.LC1), .LC1标志低12位地址加到a5, 写入a0
114     call    puts               # 跳转执行puts函数, 打印学号 (200210231)
115     call    cube               # 跳转执行cube函数, 计算31的立方
116     li      a5,0               # 加载立即数: a5 = 0, 将0加载到a5寄存器
117     mv      a0,a5              # 复制数: a0 = a5, 将a5的值写入a0
118     ld      ra,8(sp)           # 出栈: 取出原来的返回地址到ra
119     ld      s0,0(sp)           # 出栈: 取出原来的s0寄存器值到s0
120     addi    sp,sp,16           # 恢复栈指针: sp = sp + 16, 恢复栈指针sp
121     jr      ra                 # 返回: 跳转到ra寄存器指示的地址, 即返回
122     .size   main, .-main
123     .ident  "GCC: (GNU) 9.2.0"

```

3、机器码注释（只需写主程序和子程序即可）

示例:

```
1141          addi    sp,sp,-16
```

1141: 二进制为 0001 0001 0100 0001

fun3: 000, imm: 110000, rd/rs1: 00010, op: 01

c.addi 指令: $sp = sp + (-16)$

注: 此机器码来自.o 文件 (未经链接)

lab1.o: file format elf64-littleriscv

Disassembly of section .text:

0000000000000000 <cube>:

0: 7179 addi sp,sp,-48

7179=011 1 00010 11110 01

op=01 func3=011 rd/rs1=00010 imm=111110 c.addi16sp : $sp = sp + (-3)*16$

2: f406 sd ra,40(sp)

f406=111 101000 00001 10

op=10 func3=111 uimm=000101 rs2=00001 c.sdsp: $MEM(sp+(10)*4) = ra$

4: f022 sd s0,32(sp)

f022=111 100000 01000 10

op=10 func3=111 uimm=000100 rs2=01000 c.sdsp: $MEM(sp+(8)*4) = ra$

6: 1800 addi s0,sp,48

1800=000 11000000 000 00

Op=00 func3=000 uzuimm=00001100 rd'=000 c.addi4spn:s0 = $sp + 12*4$

8: 47fd li a5,31

47fd=010 0 01111 11111 01

Op=01 func3=010 imm=011111 rd=01111 c.li: $a5 = 31$

a: fef42623 sw a5,-20(s0)

fef42623=1111111 01111 01000 010 01100 0100011

Opcode=0100011 func3=010 imm=111111101100 rs2=01111 rs1=01000

sw: $MEM(s0-20)=a5$

e: 47fd li a5,31

47fd=010 0 01111 11111 01

Op=01 func3=010 imm=011111 rd=01111 c.li: $a5 = 31$

10: fef42423 sw a5,-24(s0)

fef42423=1111111 01111 01000 010 01000 0100011

Opcode=0100011 func3=010 imm=111111101000 rs2=01111 rs1=01000

sw: $MEM(s0-24)=a5$

14: 6789 lui a5,0x2

6789=011 0 01111 00010 01

Op=01 func3=011 nzimm=000010 rd=01111 c.lui: a5 = 0x2000
 16: f0078793 addi a5,a5,-256 # 1f00 <main+0x1e0e>
 f0078793=111100000000 01111 000 01111 0010011
 Opcode=0010011 func3=000 func7=0000000 imm=111100000000 rs1=01111 rd=01111
 addi: a5 = a5 + (-256)
 1a: fcf42e23 sw a5,-36(s0)
 fcf42e23=1111110 01111 01000 010 11100 0100011
 Opcode=0100011 func3=010 imm=111111011100 rs2=01111 rs1=01000
 sw: MEM(s0-36)=a5
 1e: 479d li a5,7
 479d=010 0 01111 00111 01
 Op=01 func3=010 imm=000111 rd=01111 c.li: a5 = 7
 20: fef42223 sw a5,-28(s0)
 fef42223=1111111 01111 01000 010 00100 0100011
 Opcode=0100011 func3=010 imm=111111100100 rs2=01111 rs1=01000
 sw: MEM(s0-28)=a5
 24: a091 j 68 <.L2>
 a091=101 0 0 00 0 1 0 010 0 01
 op=01 func3=101 imm=00000100010 c.j: PC=PC+34*2
 0000000000000026 <.L4>:
 26: fe842703 lw a4,-24(s0)
 fe842703=111111101000 01000 010 01110 0000011
 opcode=0000011 func3=010 rs1=01000 rd=01110 imm=111111101000
 lw: a4 = MEM(s0-24)
 2a: 41f7579b sraiw a5,a4,0x1f
 41f7579b=0100000 11111 01110 101 01111 0011011
 opcode=0011011 rd=01111 func3=101 rs1=01110 shamt=11111
 sraiw:a5 = a4 >> 0x1f
 2e: 01f7d79b srliw a5,a5,0x1f
 01f7d79b=0000000 11111 01111 101 01111 0011011
 opcode=0011011 rd=01111 func3=101 rs1=01111 shamt=11111
 srliw:a5 = a5 >> 0x1f
 32: 9f3d addw a4,a4,a5
 9f3d=100 1 11 110 01 111 01
 op=01 rs2'=111 rs1'/rd'=110 func3=100 c.addw: a4 = a4 + a5
 34: 8b05 andi a4,a4,1
 8b05=100 0 10 110 00001 01
 op=01 imm=000001 rs1'/rd'=110 func3=100 c.andi: a4 = a4 & 1
 36: 40f707bb subw a5,a4,a5
 40f707bb=0100000 01111 01110 000 01111 0111011
 opcode=0111011 rd=01111 func3=000 rs1=01110 rs2=01111 func7=0100000
 subw: a5 = a4 - a5
 3a: 2781 sext.w a5,a5
 2781=001 0 01111 00000 01

```

op=01 func3=001 rd/rs1=01111                                c.sext.w: a5 = sext(a5)
3c: 873e                                mv a4,a5
873e=100 0 01110 01111 10
op=01 rs2=01111 rd=01110 func3=100                          c.mv:a4 = a5
3e: 4785                                li a5,1
4785=010 0 01111 00001 01
Op=01 func3=010 imm=000001 rd=01111                        c.li: a5 = 1
40: 00f71963                          bne a4,a5,52 <.L3>
00f71963=0000000 01111 01110 001 10010 1100011
opcode=1100011 imm=0000000010001 fun3=001 rs1=01110 rs2=01111
bne: a4!=a5? PC=PC+imm*2:PC=PC+4
44: fe842703                          lw a4,-24(s0)
fe842703=111111101000 01000 010 01110 0000011
opcode=0000011 func3=010 rs1=01000 rd=01110 imm=111111101000
lw: a4 = MEM(s0-24)
48: fdc42783                          lw a5,-36(s0)
fdc42783=111111011100 01000 010 01111 0000011
opcode=0000011 func3=010 rs1=01000 rd=01111 imm=111111011100
lw: a5 = MEM(s0-36)
4c: 9fb9                              addw a5,a5,a4
9fb9=100 1 11 111 01 110 01
op=01 rs2'=110 rs1'/rd'=111 func3=100                      c.addw: a5 = a5 + a4
4e: fef42423                          sw a5,-24(s0)
fef42423=1111111 01111 01000 010 01000 0100011
Opcode=0100011 func3=010 imm=111111101000 rs2=01111 rs1=01000
sw: MEM(s0-24)=a5
0000000000000052 <.L3>:
52: fe842783                          lw a5,-24(s0)
fe842783=111111101000 01000 010 01111 0000011
opcode=0000011 func3=010 rs1=01000 rd=01111 imm=111111101000
lw: a5 = MEM(s0-24)
56: 4017d79b                          sraiw a5,a5,0x1
4017d79b=0100000 00001 01111 101 01111 0011011
opcode=0011011 rd=01111 func3=101 rs1=01111 shamt=00001
sraiw:a5 = a5 >> 0x1
5a: fef42423                          sw a5,-24(s0)
fef42423=1111111 01111 01000 010 01000 0100011
Opcode=0100011 func3=010 imm=111111101000 rs2=01111 rs1=01000
sw: MEM(s0-24)=a5
5e: fe442783                          lw a5,-28(s0)
fe442783=111111100100 01000 010 01111 0000011
opcode=0000011 rd=01111 func3=010 rs1=01000 imm=111111100100
lw: a5 = MEM(s0-28)
62: 37fd                              addiw a5,a5,-1

```

```

37fd=001 1 01111 11111 01
op=01 func3=001 rs1/rd=01111 imm=111111          c.addiw: a5 = a5 + (-1)
64:    fef42223                sw  a5,-28(s0)
fef42223=1111111 01111 01000 010 00100 0100011
Opcode=0100011 func3=010 imm=111111100100 rs2=01111 rs1=01000
sw: MEM(s0-28)=a5
00000000000000068 <.L2>:
68:    fe442783                lw  a5,-28(s0)
fe442783=111111100100 01000 010 01111 0000011
opcode=0000011 rd=01111 func3=010 rs1=01000 imm=111111100100
lw: a5 = MEM(s0-28)
6c:    2781                    sext.w  a5,a5
2781=001 0 01111 00000 01
op=01 func3=001 rd/rs1=01111                    c.sext.w: a5 = sext(a5)
6e:    fa07dce3                bgez  a5,26 <.L4>
fa07dce3=1111101 00000 01111 101 11001 1100011
opcode=1100011 imm=1111111011100 func3=101 rs1=01111 rs2=00000
bne: a5>=0? PC=PC+imm*2:PC=PC+4
72:    fe842783                lw  a5,-24(s0)
fe842783=111111101000 01000 010 01111 0000011
opcode=0000011 func3=010 rs1=01000 rd=01111 imm=111111101000
lw: a5 = MEM(s0-24)
76:    0087979b                slliw a5,a5,0x8
0087979b=0000000 01000 01111 001 01111 0011011
opcode=0011011 rd=01111 func3=001 rs1=01111 shamt=01000
slliw:a5 = a5 << 0x8
7a:    fef42423                sw  a5,-24(s0)
fef42423=1111111 01111 01000 010 01000 0100011
Opcode=0100011 func3=010 imm=111111101000 rs2=01111 rs1=01000
sw: MEM(s0-24)=a5
7e:    479d                    li   a5,7
479d=010 0 01111 00111 01
Op=01 func3=010 imm=000111 rd=01111            c.li: a5 = 7
80:    fef42023                sw  a5,-32(s0)
fef42023=1111111 01111 01000 010 00000 0100011
Opcode=0100011 func3=010 imm=111111100000 rs2=01111 rs1=01000
sw: MEM(s0-32)=a5
84:    a091                    j    c8 <.L5>
a091=101 0 0 00 0 1 0 010 0 01
op=01 func3=101 imm=00000100010                c.j: PC=PC+34*2
00000000000000086 <.L7>:
86:    fec42703                lw  a4,-20(s0)
fec42703=111111101100 01000 010 01110 0000011
opcode=0000011 rd=01110 func3=010 rs1=01000 imm=111111101100

```



```

lw: a4 = MEM(s0-20)
8a: 41f7579b          sraiw  a5,a4,0x1f
41f7579b=0100000 11111 01110 101 01111 0011011
opcode=0011011 rd=01111 func3=101 rs1=01110 shamt=11111
sraiw:a5 = a4 >> 0x1f
8e: 01f7d79b          srliw  a5,a5,0x1f
01f7d79b=0000000 11111 01111 101 01111 0011011
opcode=0011011 rd=01111 func3=101 rs1=01111 shamt=11111
srliw:a5 = a5 >> 0x1f
92: 9f3d              addw   a4,a4,a5
9f3d=100 1 11 110 01 111 01
op=01 rs2'=111 rs1'/rd'=110 func3=100          c.addw: a4 = a4 + a5
94: 8b05              andi  a4,a4,1
8b05=100 0 10 110 00001 01
op=01 imm=000001 rs1'/rd'=110 func3=100          c.andi: a4 = a4 & 1
96: 40f707bb          subw   a5,a4,a5
40f707bb=0100000 01111 01110 000 01111 0111011
opcode=0111011 rd=01111 func3=000 rs1=01110 rs2=01111 func7=0100000
subw: a5 = a4 - a5
9a: 2781              sext.w  a5,a5
2781=001 0 01111 00000 01
op=01 func3=001 rd/rs1=01111          c.sext.w: a5 = sext(a5)
9c: 873e              mv    a4,a5
873e=100 0 01110 01111 10
op=01 rs2=01111 rd=01110 func3=100          c.mv:a4 = a5
9e: 4785              li    a5,1
4785=010 0 01111 00001 01
Op=01 func3=010 imm=000001 rd=01111          c.li: a5 = 1
a0: 00f71963          bne  a4,a5,b2 <.L6>
00f71963=0000000 01111 01110 001 10010 1100011
opcode=1100011 imm=0000000010001 fun3=001 rs1=01110 rs2=01111
bne: a4!=a5? PC=PC+imm*2:PC=PC+4
a4: fec42703          lw    a4,-20(s0)
fec42703=111111101100 01000 010 01110 0000011
opcode=0000011 rd=01110 func3=010 rs1=01000 imm=111111101100
lw: a4 = MEM(s0-20)
a8: fe842783          lw    a5,-24(s0)
fe842783=111111101000 01000 010 01111 0000011
opcode=0000011 func3=010 rs1=01000 rd=01111 imm=111111101000
lw: a5 = MEM(s0-24)
ac: 9fb9              addw   a5,a5,a4
9fb9=100 1 11 111 01 110 01
op=01 rs2'=110 rs1'/rd'=111 func3=100          c.addw: a5 = a5 + a4
ae: fef42623          sw    a5,-20(s0)

```

```

fef42623=1111111 01111 01000 010 01100 0100011
Opcode=0100011 func3=010 imm=111111101100 rs2=01111 rs1=01000
sw: MEM(s0-20)=a5
00000000000000b2 <.L6>:
    b2:    fec42783                lw    a5,-20(s0)
fec42783=111111101100 01000 010 01111 0000011
opcode=0000011 rd=01111 func3=010 rs1=01000 imm=111111101100
lw: a5 = MEM(s0-20)
    b6:    4017d79b                sraiw   a5,a5,0x1
4017d79b=0100000 00001 01111 101 01111 0011011
opcode=0011011 rd=01111 func3=101 rs1=01111 shamt=00001
sraiw:a5 = a5 >> 0x1
    ba:    fef42623                sw     a5,-20(s0)
fef42623=1111111 01111 01000 010 01100 0100011
Opcode=0100011 func3=010 imm=111111101100 rs2=01111 rs1=01000
sw: MEM(s0-20)=a5
    be:    fe042783                lw     a5,-32(s0)
fe042783=111111100000 01000 010 01111 0000011
opcode=0000011 rd=01111 func3=010 rs1=01000 imm=111111100000
lw: a5 = MEM(s0-32)
    c2:    37fd                    addiw   a5,a5,-1
37fd=001 1 01111 11111 01
op=01 func3=001 rs1/rd=01111 imm=111111          c.addiw: a5 = a5 + (-1)
    c4:    fef42023                sw     a5,-32(s0)
fef42023=1111111 01111 01000 010 00000 0100011
Opcode=0100011 func3=010 imm=111111100000 rs2=01111 rs1=01000
sw: MEM(s0-32)=a5
00000000000000c8 <.L5>:
    c8:    fe042783                lw     a5,-32(s0)
fe042783=111111100000 01000 010 01111 0000011
opcode=0000011 rd=01111 func3=010 rs1=01000 imm=111111100000
lw: a5 = MEM(s0-32)
    cc:    2781                    sext.w   a5,a5
2781=001 0 01111 00000 01
op=01 func3=001 rd/rs1=01111                    c.sext.w: a5 = sext(a5)
    ce:    fa07dce3                bgeza5,86 <.L7>
fa07dce3=1111101 00000 01111 101 11001 1100011
opcode=1100011 imm=1111111011100 func3=101 rs1=01111 rs2=00000
bne: a5>=0? PC=PC+imm*2:PC=PC+4
    d2:    fec42783                lw     a5,-20(s0)
fec42783=111111101100 01000 010 01111 0000011
opcode=0000011 rd=01111 func3=010 rs1=01000 imm=111111101100
lw: a5 = MEM(s0-20)
    d6:    85be                    mv     a1,a5

```

```

85be=100 0 01011 01111 10
op=01 rs2=01111 rd=01011 func3=100          c.mv: a1 = a5
d8: 000007b7          lui a5,0x0
000007b7=00000000000000000000 01111 0110111
opcode=0110111 rd=01111 imm=00000000000000000000
lui:a5 = 0x0
dc: 00078513          mv a0,a5
00078513=0000000 00000 01111 000 01010 0010011
opcode=0010011 rd=01010 func3=000 rs1=01111 rs2=00000 func7=0000000
mv: add a0 = a5 + 0
e0: 00000097          auipc ra,0x0
00000097=00000000000000000000 00001 0010111
opcode=0010111 rd=00001 imm=00000
auipc:ra = PC+0
e4: 000080e7          jalr ra # e0 <.L5+0x18>
000080e7=0000000000000 00001 000 00001 1100111
Opcode=1100111 rd=00001 func3=000 rs1=00001 imm=0000000000000000
jalr: ra = PC+4;PC=ra+0
e8: 0001          nop
0001=000 0 00000 00000 01          c.nop:无操作
ea: 70a2          ld ra,40(sp)
70a2=011 1 00001 01000 10
op=10 func3=011 rd=00001 uimm=000101          c.ldsp: ra = MEM(sp+10*4)
ec: 7402          ld s0,32(sp)
7402=011 1 01000 00000 10
op=10 func3=011 rd=01000 uimm=000100          c.ldsp: s0 = MEM(sp+8*4)
ee: 6145          addi sp,sp,48
6145=011 0 00010 10001 01
op=01 func3=011 rd=00010 nzimm=000011          c.addi16sp : sp = sp + (+3)*16
f0: 8082          ret
8082=100 0 00001 00000 10
op=10 func3=100 rs1=00001 imm=000000          c.jr: PC=ra+0
00000000000000f2 <main>:
f2: 1141          addi sp,sp,-16
1141=000 1 00010 10000 01
op=01 func3=000 rs1/rd=00010 nzimm=110000          c.addi: sp = sp + (-16)
f4: e406          sd ra,8(sp)
e406=111 001000 00001 10
op=10 func3=111 rs2=00001 uimm=000001          c.sdsp: MEM(sp+(2)*4) = ra
f6: e022          sd s0,0(sp)
e022=111 000000 01000 10
op=10 fun3=111 rs2=01000 uimm=000000          c.sdsp: MEM(sp+(0)*4) = s0
f8: 0800          addi s0,sp,16
0800=000 01000000 000 00

```

```

op=00 func3=000 rd'=000 nzuimm=00000100      c.addi4spn: s0 = sp + 16
fa:      000007b7                lui   a5,0x0
000007b7=00000000000000000000 01111 0110111
opcode=0110111 rd=01111 imm=000000000000000000
lui:a5 = 0x0
fe:      00078513                mv    a0,a5
00078513=00000000 00000 01111 000 01010 0010011
opcode=0010011 rd=01010 func3=000 rs1=01111 rs2=00000 func7=00000000
mv: add a0 = a5 + 0
102:     00000097                auipc  ra,0x0
00000097=00000000000000000000 00001 0010111
opcode=0010111 rd=00001 imm=00000
auipc:ra = PC+0
106:     000080e7                jalr  ra # 102 <main+0x10>
000080e7=0000000000000 00001 000 00001 1100111
Opcode=1100111 rd=00001 func3=000 rs1=00001 imm=000000000000
jalr: ra = PC+4;PC=ra+0
10a:     00000097                auipc  ra,0x0
00000097=00000000000000000000 00001 0010111
opcode=0010111 rd=00001 imm=00000
auipc:ra = PC+0
10e:     000080e7                jalr  ra # 10a <main+0x18>
000080e7=0000000000000 00001 000 00001 1100111
Opcode=1100111 rd=00001 func3=000 rs1=00001 imm=000000000000
jalr: ra = PC+4;PC=ra+0
112:     4781                    li    a5,0
4781=010 0 01111 00000 01
Op=01 func3=010 imm=00000 rd=01111              c.li: a5 = 0
114:     853e                    mv    a0,a5
853e=100 0 01010 01111 10
op=01 rs2=01111 rd=01010 func3=100              c.mv:a0 = a5
116:     60a2                    ld     ra,8(sp)
60a2=011 0 00001 01000 10
op=10 func3=011 rd=00001 uimm=000001            c.ldsp: ra = MEM(sp+2*4)
118:     6402                    ld     s0,0(sp)
6402=011 0 01000 00000 10
op=10 func3=011 rd=01000 uimm=000000            c.ldsp: s0 = MEM(sp+0*4)
11a:     0141                    addi  sp,sp,16
0141=000 0 00010 10000 01
op=01 func3=000 rs1/rd=00010 nzimm=010000      c.addi: sp = sp + (+6)
11c:     8082                    ret
8082=100 0 00001 00000 10
op=10 func3=100 rs1=00001 imm=000000            c.jr: PC=ra+0

```

