

（深圳）

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： T2615

学生班级： 6

学生学号： 200110618

学生姓名： 邓皓元

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） |
| 2、汇编代码注释（只需写主程序和子程序即可） |
| main:  addi sp,sp,-32  将堆栈指针寄存器sp与立即数(-32)相加，再存入堆栈指针寄存器sp，即sp = sp + (-32)  sd ra,24(sp)  将返回函数地址存入栈指针寄存器sp(-8)位置  sd s0,16(sp)  将帧指针地址存入栈指针寄存器sp(-16)位置  addi s0,sp,32  将帧指针与立即数(32)相加，即s0 = sp + 32，将帧指针指向栈底  li a5,18  读取立即数，将立即数18存入a5(a5 = id = 18)  sd a5,-24(s0)  读取a5的值存入帧指针(-24)位置  ld a0,-24(s0)  将帧指针(-24)位置的值存入a0(a0 = a5 = 18)  call cube  调用cube函数  mv a5,a0  将a0的值存入a5，即a5 = a0 = 18  mv a1,a5  将a5的值存入a1，即a1 = a5 = 18  lui a5,%hi(.LC0)  计算LC0的地址，a0,…a7保存函数参数地址  addi a0,a5,%lo(.LC0)  计算LC0的地址，a0,…a7保存函数参数地址  call printf  调用printf函数  li a5,0  读取立即数，将立即数0存入a5(a5 = 0)  mv a0,a5  将a5的值存入a0，即a0 = a5 = 0,恢复参数地址  ld ra,24(sp)  读取ra的值存入栈指针寄存器sp(24)的位置，恢复ra返回地址  ld s0,16(sp)  读取s0的值存入栈指针寄存器sp(16)的位置，恢复s0帧指针  addi sp,sp,32  将堆栈指针寄存器sp与立即数(32)相加，再存入堆栈指针寄存器sp，即sp = sp + 32，恢复sp栈指针寄存器  jr ra  跳转到ra地址  .size main, .-main  .align 1  .globl cube  .type cube, @function  cube:  addi sp,sp,-64  将堆栈指针寄存器sp与立即数(-64)相加，再存入堆栈指针寄存器sp，即sp = sp + (-64)  sd s0,56(sp)  将帧指针地址存入栈指针寄存器sp(-8)位置  addi s0,sp,64  将帧指针与立即数(64)相加，即s0 = sp + 64，将帧指针指向栈底  mv a5,a0  将a0的值存入a5，即a5 = a0(a5 = a)  sw a5,-52(s0)  读取a5(a)的值存入帧指针(-52)位置，代表帧指针(-52)-帧指针(-49)存入a5(int a)的值  lw a5,-52(s0)  读取帧指针(-52)-帧指针(-49)位置的值存入a5(a5 = a)  sw a5,-20(s0)  读取a5的值存入帧指针(-20)位置，代表帧指针(-20)-帧指针(-17)存入a5的值(x = a5 = a)  lw a5,-52(s0)  读取帧指针(-52)-帧指针(-49)位置的值存入a5(a5 = a)  sd a5,-32(s0)  读取a5的值存入帧指针(-32)位置，代表帧指针(-32)-帧指针(-25)存入a5的值(y = a5 = a)  sd zero,-40(s0)  读取0存入帧指针(-40)位置，代表帧指针(-40)-帧指针(-33)存入result的值(result = 0)  sw zero,-44(s0)  读取0存入帧指针(-44)位置，代表帧指针(-44)-帧指针(-41)存入i的值(i = 0)  j .L4  跳转到.L4（第一部分）  .L6:  ld a4,-32(s0)  读取帧指针(-32)-帧指针(-25)位置的值存入a4(a4 = y)  srai a5,a4,63  将a4进行立即数算数右移63位的值存入a5(y为正数则a5 = 0，y为负数则a5 = -1)  srli a5,a5,63  将a5进行立即数逻辑左移63位的值存入a5(y为正数则a5 = 0，y为负数则a5 = -2^63)  add a4,a4,a5  即a4 = a4 + a5(y为正数则a4 = a4 + a5 = y，y为负数则a4 = a4 + a5 =y - 2^63)（将a4转换为原码）  andi a4,a4,1  将a4与立即数1的与存入a4，即用0和1表示a4奇偶(a4为奇数则a4 = 1，a4为偶数则a4 = 0)  sub a5,a4,a5  将a4与a5的差存入a5，即a5 = a4 - a5(y为正数则a5 = a4 - a5 = a4，y为负数则a5 = a4 - a5 = a4 + 2^63)  (将a4转换回补码)  mv a4,a5  将a5的值存入a4，即a4 = a5  li a5,1  读取立即数，将立即数1存入a5(a5 = 1)  bne a4,a5,.L5  进行条件判断，若a4 不等于 a5(a4 ≠ 1)，则跳转到.L5  lw a5,-20(s0)  读取帧指针(-20)-帧指针(-17)位置的值存入a5(a5 = x)  ld a4,-40(s0)  读取帧指针(-40)-帧指针(-33)位置的值存入a4(a4 = result)  add a5,a4,a5  将a5和a4的和存入a5，即a5 = a4 + a5(a5 = result + x)  sd a5,-40(s0)  读取a5的值存入帧指针(-40)位置，代表帧指针(-40)-帧指针(-33)存入a5的值(result = result + x)  .L5:  lw a5,-20(s0)  读取帧指针(-20)-帧指针(-17)位置的值存入a5(a5 = x)  slliw a5,a5,1  将a5的值立即数(字)逻辑左移1位(a5 << 1)  sw a5,-20(s0)  读取a5的值存入帧指针(-20)位置，代表帧指针(-20)-帧指针(-17)存入a5的值(x = a5 = x << 1)  ld a5,-32(s0)  读取帧指针(-32)-帧指针(-25)位置的值存入a5(a5 = y)  srai a5,a5,1  将a5的值立即数算数右移1位(a5 >> 1)  sd a5,-32(s0)  读取a5的值存入帧指针(-32)位置，代表帧指针(-32)-帧指针(-25)存入a5的值(y = a5 = y >> 1)  lw a5,-44(s0)  读取帧指针(-44)-帧指针(-41)位置的值存入a5(a5 = i)  addiw a5,a5,1  将a5和立即数1的和存入a5，即a5 = a5 + 1  sw a5,-44(s0)  读取a5的值存入帧指针(-44)位置，代表帧指针(-44)-帧指针(-41)存入a5的值(i = a5 = i + 1)  .L4:（第一部分）  lw a5,-44(s0)  读取帧指针(-44)-帧指针(-41)位置的值存入a5(i)  sext.w a4,a5  读取a5的值存入a4，即a4 = a5(a4 = i)  li a5,15  读取立即数，将立即数15存入a5(a5 = 15)  ble a4,a5,.L6  进行条件判断，若a4 < a5(i < 15)，则跳转到.L6  ld a5,-40(s0)  读取帧指针(-40)-帧指针(-33)位置的值存入a5(a5 = result)  sw a5,-20(s0)  读取a5的值存入帧指针(-20)位置，代表帧指针(-20)-帧指针(-17)存入a5的值(x = a5 = result)  lw a5,-52(s0)  读取帧指针(-52)-帧指针(-49)位置的值存入a5(a5 = a)  sd a5,-32(s0)  读取a5的值存入帧指针(-32)位置，代表帧指针(-32)-帧指针(-25)存入a5的值(y = a5 = a)  sd zero,-40(s0)  读取0存入帧指针(-40)位置，代表帧指针(-40)-帧指针(-33)存入result的值(result = 0)  sw zero,-48(s0)  读取0存入帧指针(-48)位置，代表帧指针(-48)-帧指针(-45)存入i的值(i = 0)  j .L7  跳转到.L7  .L9:  ld a4,-32(s0)  读取帧指针(-32)-帧指针(-25)位置的值存入a4(a4 = y)  srai a5,a4,63  将a4进行立即数算数右移63位的值存入a5(y为正数则a5 = 0，y为负数则a5 = -1)  srli a5,a5,63  将a5进行立即数逻辑左移63位的值存入a5(y为正数则a5 = 0，y为负数则a5 = -2^63)  add a4,a4,a5  即a4 = a4 + a5(y为正数则a4 = a4 + a5 = y，y为负数则a4 = a4 + a5 =y - 2^63)（将a4转换为原码）  andi a4,a4,1  将a4与立即数1的与存入a4，即用0和1表示a4奇偶(a4为奇数则a4 = 1，a4为偶数则a4 = 0)  sub a5,a4,a5  将a4与a5的差存入a5，即a5 = a4 - a5(y为正数则a5 = a4 - a5 = a4，y为负数则a5 = a4 - a5 = a4 + 2^63)  (将a4转换回补码)  mv a4,a5  将a5的值存入a4，即a4 = a5  li a5,1  读取立即数，将立即数1存入a5(a5 = 1)  bne a4,a5,.L8  进行条件判断，若a4 不等于 a5(a4 ≠ 1)，则跳转到.L8  lw a5,-20(s0)  读取帧指针(-20)-帧指针(-17)位置的值存入a5(a5 = x)  ld a4,-40(s0)  读取帧指针(-40)-帧指针(-33)位置的值存入a4(a4 = result)  add a5,a4,a5  将a4与a5的和存入a5，即a5 = a4 + a5(a5 = x + result)  sd a5,-40(s0)  读取a5的值存入帧指针(-40)位置，代表帧指针(-40)-帧指针(-33)存入a5的值(result = result + x)  .L8:  lw a5,-20(s0)  读取帧指针(-20)-帧指针(-17)位置的值存入a5(a5 = x)  slliw a5,a5,1  将a5的值立即数(字)逻辑左移1位(a5 << 1)  sw a5,-20(s0)  读取a5的值存入帧指针(-20)位置，代表帧指针(-20)-帧指针(-17)存入a5的值(x = a5 = x << 1)  ld a5,-32(s0)  读取帧指针(-32)-帧指针(-25)位置的值存入a5(a5 = y)  srai a5,a5,1  将a5的值立即数算数右移1位(a5 >> 1)  sd a5,-32(s0)  读取a5的值存入帧指针(-32)位置，代表帧指针(-32)-帧指针(-25)存入a5的值(y = a5 = y >> 1)  lw a5,-48(s0)  读取帧指针(-48)-帧指针(-45)位置的值存入a5(a5 = i)  addiw a5,a5,1  将a5和立即数1的和存入a5，即a5 = a5 + 1  sw a5,-48(s0)  读取a5的值存入帧指针(-48)位置，代表帧指针(-48)-帧指针(-45)存入a5的值(i = a5 = i + 1)  .L7:(第二个部分)  lw a5,-48(s0)  读取帧指针(-48)-帧指针(-45)位置的值存入a5(i)  sext.w a4,a5  读取a5的值存入a4，即a4 = a5(a4 = i)  li a5,31  读取立即数，将立即数15存入a5(a5 = 31)  ble a4,a5,.L9  进行条件判断，若a4 < a5(i < 31)，则跳转到.L9  ld a5,-40(s0)  读取帧指针(-40)-帧指针(-33)位置的值存入a5(a5 = result)  sext.w a5,a5  读取a5的值存入a5，即a5 = a5(a5 = result)  mv a0,a5  读取a5的值存入a0作为返回参数地址  ld s0,56(sp)  读取s0的值存入栈指针寄存器sp(56)的位置，恢复s0帧指针  addi sp,sp,64  将堆栈指针寄存器sp与立即数(64)相加，再存入堆栈指针寄存器sp，即sp = sp + 64，恢复sp栈指针寄存器  jr ra  跳转到ra地址  .size cube, .-cube  .ident "GCC: (GNU) 9.2.0" |
| 3、机器码注释（只需写主程序和子程序即可） |
| 0000000000000000 <main>:  0: 1101 addi sp,sp,-32  1101：二进制为0001 0001 0000 0001  fun3：000，imm：100000，rd/rs1：00010，op：01  c.addi指令：sp = sp + (-32)  2: ec06 sd ra,24(sp)  ec06：二进制为1110 1100 0000 0110  fun3：111，uimm[8:3]：000011，rs2：00001，op：10  c.sdsp指令：24(sp) = ra  4: e822 sd s0,16(sp)  e822：二进制为1110 1000 0010 0010  fun3：111，uimm[8:3]：000010，rs2：01000，op：10  c.sdsp指令：16(sp) = s0  6: 1000 addi s0,sp,32  1000：二进制为0001 0000 0000 0000  fun3：000，uimm[9:2]：00001000，rd：00000，op：00  c.addi指令：s0 = sp + 32  8: 47c9 li a5,18  47e9：二进制为0100 0111 1110 1001  fun3：010，imm：010010，rd：01111，op：01  c.addiw指令：a5 = 18  a: fef43423 sd a5,-24(s0)  fef43423：二进制为1111 1110 1111 0100 0011 0100 0010 0011  fun3：011，offset：1111 1110 1000，rs2：01111，rs1:01000，op：0100011  sd指令：-24(s0) = a5  e: fe843503 ld a0,-24(s0)  fe843503：二进制为1111 1110 1000 0100 0011 0101 0000 0011  fun3：011，offset：1111 1110 1000，rd：01010，rs1:01000，op：0000011  ld指令：a0=-24(s0)  12: 00000097 auipc ra,0x0  00000097：二进制为0000 0000 0000 0000 0000 0000 1001 0111  imm[31:12]：0000 0000 0000 0000 0000，rd：00001，op：0010111  auipc指令：ra = 0x0  16: 000080e7 jalr ra # 12 <main+0x12>  000080e7：二进制为0000 0000 0000 0000 1000 0000 1110 0111  fun3：000，offset：0000 0000 0000，rs1：00001，op：1100111  jalr指令：jalr ra  1a: 87aa mv a5,a0  87aa：二进制为1000 0111 1010 1010  fun3：100，rd：01111，rs2：01010，op：10  c.mv指令：a5 = a0  1c: 85be mv a1,a5  85be：二进制为1000 0101 1011 1110  fun3：100，rd：01011，rs2：01111，op：10  c.mv指令：a1 = a5  1e: 000007b7 lui a5,0x0  000007b7：二进制为0000 0000 0000 0000 0000 0111 1011 0111  imm[31:12]：0000 0000 0000 0000 0000，rd：01111，op：0110111  lui指令：a5 = 0x0  22: 00078513 mv a0,a5  00078513：二进制为0000 0000 0000 0111 1000 0101 0001 0011  fun3：000，imm：0000 0000 0000，rs1：01111，rd：01010，op：0010011  addi指令：a0 = a5 + 0  26: 00000097 auipc ra,0x0  00000097：二进制为0000 0000 0000 0000 0000 0000 1001 0111  imm[31:12]：0000 0000 0000 0000 0000，rd：00001，op：0010111  auipc指令：ra = 0x0  2a: 000080e7 jalr ra # 26 <main+0x26>  000080e7：二进制为0000 0000 0000 0000 1000 0000 1110 0111  fun3：000，offset：0000 0000 0000，rs1：00001，op：1100111  jalr指令：jalr ra  2e: 4781 li a5,0  4781：二进制为0100 0111 1000 0001  fun3：010，imm：000000，rd：01111，op：01  c.addiw指令：a5 = 0  30: 853e mv a0,a5  853e：二进制为1000 0101 0011 1110  fun3：100，rd：01010，rs2：01111，op：10  c.mv指令：a0 = a5  32: 60e2 ld ra,24(sp)  60e2：二进制为0110 0000 1110 0010  fun3：011，uimm[8:3]：000011，rd：00001，op：10  c.ld指令：ra = 24(sp)  34: 6442 ld s0,16(sp)  6442：二进制为0110 0100 0100 0010  fun3：011，uimm[8:3]：000010，rd：01000，op：10  c.ldsp指令：s0 = 16(sp)  36: 6105 addi sp,sp,32  6105：二进制为0110 0001 0000 0101  fun3：011，imm[9:4]：000010，op：01  c.addi16sp指令：sp = sp + 32  38: 8082 ret  8082：二进制为1000 0000 1000 0010  fun3：011，rs1：00001，op：01  c.jr指令：jalr x0，0(ra)  000000000000003a <cube>:  3a: 7139 addi sp,sp,-64  7139：二进制为0111 0001 0011 1001  fun3：011，imm[9:4]：111100，op：01  c.addi16sp指令：sp = sp + (-64)  3c: fc22 sd s0,56(sp)  fc22：二进制为1111 1100 0010 0010  fun3：111，uimm[8:3]：000111，rs2：01000，op：10  c.sdsp指令：56(sp) = s0  3e: 0080 addi s0,sp,64  0080：二进制为0000 0000 1000 0000  fun3：000，uimm[9:2]：00001000，rd：000，op：00  c.addi4spn指令：s0 = sp + 64  40: 87aa mv a5,a0  87aa：二进制为1000 0111 1010 1010  fun3：100，rd：01111，rs2：01010，op：10  c.mv指令：a5 = a0  42: fcf42623 sw a5,-52(s0)  fcf42623：二进制为1111 1100 1111 0100 0010 0110 0010 0011  fun3：010，offset：1111 1100 1100，rs2：01111，rs1:01000，op：0100011  sw指令：-52(s0) = a5  46: fcc42783 lw a5,-52(s0)  fcc42783：二进制为1111 1100 1100 0100 0010 0111 1000 0011  fun3：010，offset：1111 1100 1100，rd：01111，rs1: 01000，op：0000011  lw指令：a5 = -52(s0)  4a: fef42623 sw a5,-20(s0)  fef42623：二进制为1111 1110 1111 0100 0010 0110 0010 0011  fun3：010，offset：1111 1110 1100，rs2：01111，rs1:01000，op：0100011  sw指令：-20(s0) = a5  4e: fcc42783 lw a5,-52(s0)  fcc42783：二进制为1111 1100 1100 0100 0010 0111 1000 0011  fun3：010，offset：1111 1100 1100，rd：01111，rs1: 01000，op：0000011  lw指令：a5 = -52(s0)  52: fef43023 sd a5,-32(s0)  fef43023：二进制为1111 1110 1111 0100 0011 0000 0010 0011  fun3：011，offset：1111 1110 0000，rs2：01111，rs1:01000，op：0100011  sd指令：-32(s0) = a5  56: fc043c23 sd zero,-40(s0)  fc043c23：二进制为1111 1100 0000 0100 0011 1100 0010 0011  fun3：011，offset：1111 110101000，rs2：00000，rs1:01000，op：0100011  sd指令：-40(s0) = 0  5a: fc042a23 sw zero,-44(s0)  fc042a23：二进制为1111 1100 0000 0100 0010 1010 0010 0011  fun3：010，offset：1111 1101 0100，rs2：00000，rs1:01000，op：0100011  sw指令：-44(s0) = 0  5e: a0a9 j a8 <.L4>  a0a9：二进制为1010 0000 1010 1001  fun3：101，offset[11:1]：000 0010 0101  c.j指令：j .L4  0000000000000060 <.L6>:  60: fe043703 ld a4,-32(s0)  fe043703：二进制为1111 1110 0000 0100 0011 0111 0000 0011  fun3：011，offset：1111 1110 0000，rd：01000，rs1:01110，op：0000011  ld指令：a4=-32(s0)  64: 43f75793 srai a5,a4,0x3f  43f75793：二进制为0100 0011 1111 0111 0101 0111 1001 0011  fun3：101，shamt：111111，rd：01000，rs1: 01111，op：0010011  srai指令：a5 = a4 >> 63  68: 93fd srli a5,a5,0x3f  93fd：二进制为1001 0011 1111 1101  fun3：100，uimm[5:0]：111111，rd：111，op：01  c.srli指令：a5=a5 << 63  6a: 973e add a4,a4,a5  973e：二进制为1001 0111 0011 1110  fun3：100，rd：01110，rs2：01111，op：10  c.add指令：a4 = a4 + a5  6c: 8b05 andi a4,a4,1  8b05：二进制为1000 1011 0000 0101  fun3：100，imm：000001，rd：110，rs2：01111，op：01  c.andi指令：a4 = a4 and 1  6e: 40f707b3 sub a5,a4,a5  40f707b3：二进制为0100 0000 1111 0111 0000 0111 1011 0011  fun7：0100000，fun3：000，rs2：01111，rs1：01110，rd: 01111，op：0110011  sub指令：a5 = a4 - a5  72: 873e mv a4,a5  873e：二进制为1000 0111 0011 1110  fun3：100，rd：01110，rs2：01111，op：10  c.mv指令：a4 = a5  74: 4785 li a5,1  4785：二进制为0100 0111 1000 0101  fun3：010，imm：000001，rd：01111，op：01  c.addiw指令：a5 = 1  76: 00f71963 bne a4,a5,88 <.L5>  00f71963：二进制为0000 0000 1111 0111 0001 1001 0110 0011  fun3：001，offset[12:1]：00000 0001 0010，rs2：01111，rs1：01110，op：1100011  bne指令：bne a4，a5，88(.L5)  7a: fec42783 lw a5,-20(s0)  fec42783：二进制为1111 1110 1100 0100 0010 0111 1000 0011  fun3：010，offset：1111 1110 1100，rd：01111，rs1: 01000，op：0000011  lw指令：a5 = -20(s0)  7e: fd843703 ld a4,-40(s0)  fd843703：二进制为1111 1101 1000 0100 0011 0111 0000 0011  fun3：011，offset：1111 1101 1000，rd：01110，rs1:01000，op：0000011  ld指令：a4=-40(s0)  82: 97ba add a5,a5,a4  97ba：二进制为1001 0111 1011 1010  fun3：100，rd：01111，rs2：01110，op：10  c.add指令：a5 = a5 + a4  84: fcf43c23 sd a5,-40(s0)  fcf43c23：二进制为1111 1100 1111 0100 0011 1100 0010 0011  fun3：011，offset：1111 1101 1000，rs2：01111，rs1：01000，op：0100011  sd指令：-40(s0) = a5  0000000000000088 <.L5>:  88: fec42783 lw a5,-20(s0)  fec42783：二进制为1111 1110 1100 0100 0010 0111 1000 0011  fun3：010，offset：1111 1110 1100，rd：01111，rs1: 01000，op：0000011  lw指令：a5 = -20(s0)  8c: 0017979b slliw a5,a5,0x1  0017979b：二进制为0000 0000 0001 0111 1001 0111 1000 1011  fun3：001，shamt：000001，rd：01111，rs1: 01111，op：0001011  slliw指令：a5=a5<<1  90: fef42623 sw a5,-20(s0)  fef42623：二进制为1111 1110 1111 0100 0010 0110 0010 0011  fun3：010，offset：1111 1110 1100，rs2：01111，rs1:01000，op：0100011  sw指令：-20(s0) = a5  94: fe043783 ld a5,-32(s0)  fe043783：二进制为1111 1110 0000 0100 0011 0111 1000 0011  fun3：011，offset：1111 1110 0000，rd：01000，rs1:01111，op：0000011  ld指令：a5=-32(s0)  98: 8785 srai a5,a5,0x1  8785：二进制为1000 0111 1000 0101  fun3：100，uimm：000001，rd：111，op：01  c.srai指令：a5 = a5 >> 1  9a: fef43023 sd a5,-32(s0)  fef43023：二进制为1111 1110 1111 0100 0011 0000 0010 0011  fun3：011，offset：1111 1110 0000，rs2：01111，rs1:01000，op：0100011  sd指令：-32(s0) = a5  9e: fd442783 lw a5,-44(s0)  fd442783：二进制为1111 1101 0100 0100 0010 0111 1000 0011  fun3：010，offset：1111 1101 0100，rd：01111，rs1: 01000，op：0000011  lw指令：a5 = -44(s0)  a2: 2785 addiw a5,a5,1  2785：二进制为0010 0111 1000 0101  fun3：000，imm：000001，rd：01111，op：01  c.addiw指令：a5 = a5 + 1  a4: fcf42a23 sw a5,-44(s0)  fcf42a23：二进制为1111 1100 1111 0100 0010 1010 0010 0011  fun3：010，offset：1111 1101 0100，rs2：01111，rs1:01000，op：0100011  sw指令：-44(s0)=a5  00000000000000a8 <.L4>:  a8: fd442783 lw a5,-44(s0)  fd442783：二进制为1111 1101 0100 0100 0010 0111 1000 0011  fun3：010，offset：1111 1101 0100，rd：01111，rs1: 01000，op：0000011  lw指令：a5 = -44(s0)  ac: 0007871b sext.w a4,a5  0007871b：二进制为0000 0000 0000 0111 1000 0111 0001 1011  fun3：000，imm：0000 0000 0000，rd：01110，rs1: 01111，op：0011011  addiw指令：a4 = a5 + 0  b0: 47bd li a5,15  47bd：二进制为0100 0111 1011 1101  fun3：010，imm：001111，rd：01111，op：01  c.addiw指令：a5 = 15  b2: fae7d7e3 bge a5,a4,60 <.L6>  fae7d7e3：二进制为1111 1010 1110 0111 1101 0111 1101 0011  fun3：101，offset[12:1]：1111 1101 0111，rs1：01111，rs2： 01110，op：1010011  bge指令：bge a5,a4,60 (.L6)  b6: fd843783 ld a5,-40(s0)  fd843783：二进制为1111 1101 1000 0100 0011 0111 1000 0011  fun3：011，offset：1111 1101 1000，rd：01111，rs1:01000，op：0000011  ld指令：a5=-40(s0)  ba: fef42623 sw a5,-20(s0)  fef42623：二进制为1111 1110 1111 0100 0010 0110 0010 0011  fun3：010，offset：1111 1110 1100，rs2：01111，rs1:01000，op：0100011  sw指令：-20(s0) = a5  be: fcc42783 lw a5,-52(s0)  fcc42783：二进制为1111 1100 1100 0100 0010 0111 1000 0011  fun3：010，offset：1111 1100 1100，rd：01111，rs1: 01000，op：0000011  lw指令：a5 = -52(s0)  c2: fef43023 sd a5,-32(s0)  fef43023：二进制为1111 1110 1111 0100 0011 0000 0010 0011  fun3：011，offset：1111 1110 0000，rs2：01111，rs1:01000，op：0100011  sd指令：-32(s0) = a5  c6: fc043c23 sd zero,-40(s0)  fc043c23：二进制为1111 1100 0000 0100 0011 1100 0010 0011  fun3：011，offset：1111 110101000，rs2：00000，rs1:01000，op：0100011  sd指令：-40(s0) = 0  ca: fc042823 sw zero,-48(s0)  fc042823：二进制为1111 1100 0000 0100 0010 1000 0010 0011  fun3：010，offset：1111 1101 0000，rs2：00000，rs1:01000，op：0100011  sw指令：-48(s0) = 0  ce: a0a9 j 118 <.L7>  a0a9：二进制为1010 0000 1010 1001  fun3：101，offset[11:1]：000 0010 0101  c.j指令：j .L4  00000000000000d0 <.L9>:  d0: fe043703 ld a4,-32(s0)  fe043703：二进制为1111 1110 0000 0100 0011 0111 0000 0011  fun3：011，offset：1111 1110 0000，rd：01000，rs1:01110，op：0000011  ld指令：a4=-32(s0)  d4: 43f75793 srai a5,a4,0x3f  43f75793：二进制为0100 0011 1111 0111 0101 0111 1001 0011  fun3：101，shamt：111111，rd：01000，rs1: 01111，op：0010011  srai指令：a5 = a4 >> 63  d8: 93fd srli a5,a5,0x3f  93fd：二进制为1001 0011 1111 1101  fun3：100，uimm[5:0]：111111，rd：111，op：01  c.srli指令：a5=a5 << 63  da: 973e add a4,a4,a5  973e：二进制为1001 0111 0011 1110  fun3：100，rd：01110，rs2：01111，op：10  c.add指令：a4 = a4 + a5  dc: 8b05 andi a4,a4,1  8b05：二进制为1000 1011 0000 0101  fun3：100，imm：000001，rd：110，rs2：01111，op：01  c.andi指令：a4 = a4 and 1  de: 40f707b3 sub a5,a4,a5  40f707b3：二进制为0100 0000 1111 0111 0000 0111 1011 0011  fun7：0100000，fun3：000，rs2：01111，rs1：01110，rd: 01111，op：0110011  sub指令：a5 = a4 - a5  e2: 873e mv a4,a5  873e：二进制为1000 0111 0011 1110  fun3：100，rd：01110，rs2：01111，op：10  c.mv指令：a4 = a5  e4: 4785 li a5,1  4785：二进制为0100 0111 1000 0101  fun3：010，imm：000001，rd：01111，op：01  c.addiw指令：a5 = 1  e6: 00f71963 bne a4,a5,f8 <.L8>  00f71963：二进制为0000 0000 1111 0111 0001 1001 0110 0011  fun3：001，offset[12:1]：00000 0001 0010，rs2：01111，rs1：01110，op：1100011  bne指令：bne a4，a5，f8(.L8)  ea: fec42783 lw a5,-20(s0)  fec42783：二进制为1111 1110 1100 0100 0010 0111 1000 0011  fun3：010，offset：1111 1110 1100，rd：01111，rs1: 01000，op：0000011  lw指令：a5 = -20(s0)  ee: fd843703 ld a4,-40(s0)  fd843703：二进制为1111 1101 1000 0100 0011 0111 0000 0011  fun3：011，offset：1111 1101 1000，rd：01110，rs1:01000，op：0000011  ld指令：a4=-40(s0)  f2: 97ba add a5,a5,a4  97ba：二进制为1001 0111 1011 1010  fun3：100，rd：01111，rs2：01110，op：10  c.add指令：a5 = a5 + a4  f4: fcf43c23 sd a5,-40(s0)  fcf43c23：二进制为1111 1100 1111 0100 0011 1100 0010 0011  fun3：011，offset：1111 1101 1000，rs2：01111，rs1：01000，op：0100011  sd指令：-40(s0) = a5  00000000000000f8 <.L8>:  f8: fec42783 lw a5,-20(s0)  fec42783：二进制为1111 1110 1100 0100 0010 0111 1000 0011  fun3：010，offset：1111 1110 1100，rd：01111，rs1: 01000，op：0000011  lw指令：a5 = -20(s0)  fc: 0017979b slliw a5,a5,0x1  0017979b：二进制为0000 0000 0001 0111 1001 0111 1000 1011  fun3：001，shamt：000001，rd：01111，rs1: 01111，op：0001011  slliw指令：a5=a5<<1  100: fef42623 sw a5,-20(s0)  fef42623：二进制为1111 1110 1111 0100 0010 0110 0010 0011  fun3：010，offset：1111 1110 1100，rs2：01111，rs1:01000，op：0100011  sw指令：-20(s0) = a5  104: fe043783 ld a5,-32(s0)  fe043783：二进制为1111 1110 0000 0100 0011 0111 1000 0011  fun3：011，offset：1111 1110 0000，rd：01000，rs1:01111，op：0000011  ld指令：a5=-32(s0)  108: 8785 srai a5,a5,0x1  8785：二进制为1000 0111 1000 0101  fun3：100，uimm：000001，rd：111，op：01  c.srai指令：a5 = a5 >> 1  10a: fef43023 sd a5,-32(s0)  fef43023：二进制为1111 1110 1111 0100 0011 0000 0010 0011  fun3：011，offset：1111 1110 0000，rs2：01111，rs1:01000，op：0100011  sd指令：-32(s0) = a5  10e: fd042783 lw a5,-48(s0)  fd042783：二进制为1111 1101 0000 0100 0010 0111 1000 0011  fun3：010，offset：1111 1101 0000，rd：01111，rs1: 01000，op：0000011  lw指令：a5 = -48(s0)  112: 2785 addiw a5,a5,1  2785：二进制为0010 0111 1000 0101  fun3：000，imm：000001，rd：01111，op：01  c.addiw指令：a5 = a5 + 1  114: fcf42823 sw a5,-48(s0)  fcf42823：二进制为1111 1100 1111 0100 0010 1000 0010 0011  fun3：010，offset：1111 1101 0000，rs2：01111，rs1:01000，op：0100011  sw指令：-48(s0)=a5  0000000000000118 <.L7>:  118: fd042783 lw a5,-48(s0)  fd042783：二进制为1111 1101 0000 0100 0010 0111 1000 0011  fun3：010，offset：1111 1101 0000，rd：01111，rs1: 01000，op：0000011  lw指令：a5 = -48(s0)  11c: 0007871b sext.w a4,a5  0007871b：二进制为0000 0000 0000 0111 1000 0111 0001 1011  fun3：000，imm：0000 0000 0000，rd：01110，rs1: 01111，op：0011011  addiw指令：a4 = a5 + 0  120: 47fd li a5,31  47fd：二进制为0100 0111 1111 1101  fun3：010，imm：011111，rd：01111，op：01  c.addiw指令：a5 = 31  122: fae7d7e3 bge a5,a4,d0 <.L9>  fae7d7e3：二进制为1111 1010 1110 0111 1101 0111 1101 0011  fun3：101，offset[12:1]：1111 1101 0111，rs1：01111，rs2： 01110，op：1010011  bge指令：bge a5,a4,d0 (.L9)  126: fd843783 ld a5,-40(s0)  fd843783：二进制为1111 1101 1000 0100 0011 0111 1000 0011  fun3：011，offset：1111 1101 1000，rd：01111，rs1:01000，op：0000011  ld指令：a5=-40(s0)  12a: 2781 sext.w a5,a5  2781：二进制为0010 0111 1000 0001  fun3：001，imm：000000，rd：01111，op：01  c.addiw指令：a5 = a5 + 0  12c: 853e mv a0,a5  853e：二进制为1000 0101 0011 1110  fun3：100，rd：01010，rs2：01111，op：10  c.mv指令：a0 = a5  12e: 7462 ld s0,56(sp)  7462：二进制为0111 0100 0110 0010  fun3：011，uimm[8:3]：000111，rd：01000，op：10  c.ldsp指令：s0 = 56(sp)  130: 6121 addi sp,sp,64  6121：二进制为0110 0001 0010 1001  fun3：011，imm[9:4]：000100，op：01  c.addi16sp指令：sp = sp + 64  132: 8082 ret  8082：二进制为1000 0000 1000 0010  fun3：011，rs1：00001，op：01  c.jr指令：jalr x0，0(ra)  Disassembly of section .rodata:  0000000000000000 <.LC0>:  0: 4c25 li s8,9  2: 644c ld a1,136(s0)  ...  Disassembly of section .comment:  0000000000000000 <.comment>:  0: 4700 lw s0,8(a4)  2: 203a4343 fmadd.s ft6,fs4,ft3,ft4,rmm  6: 4728 lw a0,72(a4)  8: 554e lw a0,240(sp)  a: 2029 0x2029  c: 2e39 addiw t3,t3,14  e: 2e32 fld ft8,264(sp)  10: 0030 addi a2,sp,8  Disassembly of section .riscv.attributes:  0000000000000000 <.riscv.attributes>:  0: 3441 addiw s0,s0,-16  2: 0000 unimp  4: 7200 ld s0,32(a2)  6: 7369 lui t1,0xffffa  8: 01007663 bgeu zero,a6,14 <.riscv.attributes+0x14>  c: 002a c.slli zero,0xa  e: 0000 unimp  10: 1004 addi s1,sp,32  12: 7205 lui tp,0xfffe1  14: 3676 fld fa2,376(sp)  16: 6934 ld a3,80(a0)  18: 7032 0x7032  1a: 5f30 lw a2,120(a4)  1c: 326d addiw tp,tp,-5  1e: 3070 fld fa2,224(s0)  20: 615f 7032 5f30 0x5f307032615f  26: 3266 fld ft4,120(sp)  28: 3070 fld fa2,224(s0)  2a: 645f 7032 5f30 0x5f307032645f  30: 30703263 0x30703263 |