

（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：

学生班级： 3

学生学号： 200110309

学生姓名： 段子豪

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 |
| J{Q0}@Q4_NONC_IG0JHGL2M |
| 2、汇编代码注释 |
| **main**:  **addi**    sp,sp,-32  将sp与立即数-32相加，再存入sp寄存器  **sd**  s0,24(sp)  将s0寄存器值送到sp寄存器值+24的地址处保存（双字）  **addi**    s0,sp,32  将sp与立即数32相加，再存入fp寄存器,栈帧寄存器  **li**  a5,9  把立即数9送到a5寄存器，是个伪指令好像，汇编器会改成addi a5 zero 9，初始化被乘数变量（字）  **sw**  a5,-20(s0)  把a5寄存器值送入fp寄存器值与-20相加地址处保存，即存储被乘数  **li**  a5,9  把立即数9送到a5寄存器，初始化乘数变量（字）  **sw**  a5,-24(s0)  把a5寄存器值送入fp寄存器值与-24相加地址处保存（字），即存储乘数  **sw**  zero,-28(s0)  把zero寄存器值0送入fp寄存器值与-28相加地址处保存（字），即保存中间结果，最后是积，一开始是0  **j**   .L2  跳转到.l2处  .**L4**:  **lw**  a5,-24(s0)  将fp寄存器值与-24相加地址处数据送入a5寄存器（字），取乘数  **andi**    a5,a5,1  将a5寄存器值与立即数1与，结果送入a5寄存器，计算移位后乘数最低位  **sw**  a5,-32(s0)  把a5寄存器值送入fp寄存器值与-32相加地址处保存（字），保存移位后乘数最低位  **lw**  a5,-32(s0)  将fp寄存器值与-32相加地址处数据送入a5寄存器（字），取移位后乘数最低位  **sext.w**  a5,a5  **beq** a5,zero,.L3  判断a5和zero寄存器值是否相等，相等则跳转到l3，判断移位后乘数最低位是否为0  **lw**  a4,-28(s0)  将fp寄存器值与-28相加地址处数据送入a4寄存器（字），取中间结果  **lw**  a5,-20(s0)  将fp寄存器值与-28相加地址处数据送入a4寄存器（字），即取移位后被乘数值  **addw**    a5,a4,a5  将a4寄存器值与a5寄存器值相加送到a5寄存器（字），计算新的中间结果  **sw**  a5,-28(s0)  把a5寄存器值送入fp寄存器值与-28相加地址处保存（字），即保存中间结果  .**L3**:  **lw**  a5,-20(s0)  将fp寄存器值与-20相加地址处数据送入a5寄存器（字），即取出被乘数移位结果  **slliw**   a5,a5,1  将a5寄存器值左移一位，送入a5寄存器，即被乘数移位结果左移一位  **sw**  a5,-20(s0)  把a5寄存器值送入fp寄存器值与-20相加地址处保存（字），即保存左移后的被乘数  **lw**  a5,-24(s0)  将fp寄存器值与-24相加地址处数据送入a5寄存器（字），即取出乘数移位结果  **sraiw**   a5,a5,1  将a5寄存器值右移一位，送入a5寄存器，即乘数移位结果右移一位  **sw**  a5,-24(s0)  把a5寄存器值送入fp寄存器值与-24相加地址处保存（字），即保存右移后的乘数  .**L2**:  **lw**  a5,-24(s0)  将fp寄存器值与-24相加地址处数据送入a5寄存器（字），即取乘数  **sext.w**  a5,a5  **bne** a5,zero,.L4  判断a5寄存器值乘数和0是否相等，不相等则跳转到.l4，即判断乘法是否结束，通过判断移位后的乘数是否等于0  **nop**  **空指令，什么都不干**  **nop**  **空指令，什么都不干**  **ld**  s0,24(sp)  将sp寄存器值与24相加地址处数据送入fp寄存器（双字），即恢复原来的栈帧  **addi**    sp,sp,32  将sp与立即数32相加，再存入sp寄存器，弹栈  **jr**  ra  跳转到ra寄存器的值代表的地址，即return |
| 3、机器码注释 |
| 示例：    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)  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)  ec22 sd s0,24(sp)  ec22：二进制为1110 1100 0010 0010  fun3：111，imm：0 1100 0 ，rs2：010 00，op：10  c.swsp指令：sd s0,24(sp)（ps：在riscv官方文档并没有找到16位的sd指令）  ？1000 addi s0,sp,32  1000：二进制为0001 0000 0000 0000  fun3：000，imm：1 0000 000，rd：0 00，op：00  c.addi4spn指令：addi s0,sp,32  47a5 li a5,9  47a5：二进制为0100 0111 1010 0101  li是伪指令  fef42623 sw a5,-20(s0)  fef42623：二进制为1111 1110 1111 0100 0010 0110 0010 0011  imm：1111 1110110 0 ，rs2：0 1111，rs1：0100 0，fun3：010，opcode：010 0011  s指令：sw a5,-20(s0)  47a5 li a5,9  47a5：二进制为0100 0111 1010 0101  li是伪指令  fef42423 sw a5,-24(s0)  fef42423：二进制为1111 1110 1111 0100 0010 0100 0010 0011  imm：1111 1110100 0 ，rs2：0 1111，rs1：0100 0，fun3：010，opcode：010 0011  s指令：sw a5,-24(s0)  fe042223 sw zero,-28(s0)  fe042223：二进制为1111 1110 0000 0100 0010 0010 0010 0011  imm：1111 1110010 0 ，rs2：0 0000，rs1：0100 0，fun3：010，opcode：010 0011  s指令：sw zero,-28(s0)  a82d j 101a6 <main+0x50>  a82d：二进制为1010 1000 0010 1101  j是伪指令  fe842783 lw a5,-24(s0)  fe842783：二进制为1111 1110 1000 0100 0010 0111 1000 0011  imm：1111 1110 1000 ，rs1：0100 0，rd：0111 1，fun3：010，opcode：000 0011  i指令：lw a5,-24(s0)  8b85 andi a5,a5,1  8b85：二进制为1000 1011 1000 0101  fun3：100，imm：0000 01，rd/rs1：1011 1，op：01  c.andi指令：andi a5,a5,1  fef42023 sw a5,-32(s0)  fef42023：二进制为1111 1110 1111 0100 0010 0000 0010 0011  imm：1111 1110000 0 ，rs2：0 1111，rs1：0100 0，fun3：010，opcode：010 0011  s指令：sw a5,-32(s0)  fe042783 lw a5,-32(s0)  fe042783：二进制为1111 1110 0000 0100 0010 0111 1000 0011  imm：1111 1110 1000 ，rs1：0100 0，rd：0111 1，fun3：010，opcode：000 0011  i指令：lw a5,-32(s0)  2781 sext.w a5,a5  2781：二进制为0010 0111 1000 0001  cb81 beqz a5,1018e <main+0x38>  cb81：二进制为1100 1011 1000 0001  fun3：110，offset：0 10000 00，rs1：11 1，op：01  c.beqz指令：beqz a5,1018e <main+0x38>  fe442703 lw a4,-28(s0)  fe442703：二进制为1111 1110 0100 0100 0010 0111 0000 0011  imm：1111 1110 0100 ，rs1：0100 0，rd：0111 0，fun3：010，opcode：000 0011  i指令：lw a4,-28(s0)  fec42783 lw a5,-20(s0)  fec42783：二进制为1111 1110 1100 0100 0010 0111 1000 0010 1111  imm：1111 1110 1100 ，rs1：0100 0，rd：1000 0，fun3：010，opcode：000 0011  i指令：lw a5,-20(s0)  9fb9 addw a5,a5,a4  9fb9：二进制为1001 1111 1011 1001  func4：1001，rd/rs1：11111，rs2：011 10，op：01  c.addw指令：addw a5,a5,a4  fef42223 sw a5,-28(s0)  fef42223：二进制为1111 1110 1111 0100 0010 0010 0010 0011  imm：1111 1110010 0 ，rs2：0 1111，rs1：0100 0，fun3：010，opcode：010 0011  s指令：sw a5,-28(s0)  fec42783 lw a5,-20(s0)  fec42783：二进制为1111 1110 1100 0100 0010 0111 1000 0011  imm：1111 1110 1100 ，rs1：0100 0，rd：1000 0，fun3：010，opcode：000 0011  i指令：lw a5,-20(s0)  0017979b slliw a5,a5,0x1  0017979b：二进制为0000 0000 0001 0111 1001 0111 1001 1011  imm：0000 0000 0001 ，rs1：0111 1，rd：0111 1，fun3：001，opcode：001 1011  i指令：slliw a5,a5,0x1  fef42623 sw a5,-20(s0)  fef42623：二进制为1111 1110 1111 0100 0010 0110 0010 0011  imm：1111 1110110 0 ，rs2：0 1111，rs1：0100 0，fun3：010，opcode：010 0011  s指令：sw a5,-20(s0)  fe842783 lw a5,-24(s0)  fe842783：二进制为1111 1110 1000 0100 0010 0111 1000 0011  imm：1111 1110 1000 ，rs1：0100 0，rd：0111 1，fun3：010，opcode：000 0011  i指令：lw a5,-24(s0)  4017d79b sraiw a5,a5,0x1  4017d79b：二进制为0100 0000 0001 0111 1101 0111 1001 1011  imm：0000 0000 0001 ，rs1：0111 1，rd：0111 1，fun3：101，opcode：001 1011  i指令：sraiw a5,a5,0x1  fef42423 sw a5,-24(s0)  fef42423：二进制为1111 1110 1111 0100 0010 0100 0010 0011  imm：1111 1110100 0 ，rs2：0 1111，rs1：0100 0，fun3：010，opcode：010 0011  s指令：sw a5,-24(s0)  fe842783 lw a5,-24(s0)  fe842783：二进制为1111 1110 1000 0100 0010 0111 1000 0010 1111  imm：1111 1110 1000 ，rs1：0100 0，rd：0111 1，fun3：010，opcode：000 0011  i指令：lw a5,-24(s0)  2781 sext.w a5,a5  2781：二进制为0010 0111 1000 0001  f3e9 bnez a5,1016e <main+0x18>  f3e9：二进制为1111 0011 1110 1001  fun3：111，offset：1 00110 10，rs1：11 1，op：01  c.bnez指令：beqza5,1018e <main+0x38>  0001 nop  0001：二进制为0000 0000 0000 0001  0001 nop  0001：二进制为0000 0000 0000 0001  6462 ld s0,24(sp)  6462：二进制为0110 0100 0110 0010  fun3：011，imm：0110 00，rd/rs1：0100 0，op10  c.lwsp指令：ld s0,24(sp)  6105 addi sp,sp,32  6105：二进制为0110 0001 0000 0101  fun3：011，imm：0000 01，rd/rs1：0001 0，op：10  addi16sp指令：addi sp,sp,32  8082 ret  8082：二进制为1000 0000 1000 0010  ret伪指令 |