

（深圳）

实验报告

开课学期： 2025春季

课程名称：计算机组成原理（实验）

实验名称： RISC-V汇编程序设计

实验性质： 设计型

实验学时： 4 地点： T2506

学生班级： 20233117

学生学号： 2023311709

学生姓名： 宁中昊

作业成绩：

实验与创新实践教育中心制

2025年4月

|  |
| --- |
| 1、指令解析 |
| 要求：查看**可执行程序**的反汇编文件，在算术运算、移位运算、访存、分支跳转四种指令中，每种分别选出一条机器指令，并参照示例对它们进行解析。 |
| **示例：**  指令1：   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 机器码 | 16进制 | 2进制 | | | | | c3418513 | 1100 0011 0100 0001 1000 0101 0001 0011 | | | | | opcode | 001 0011 | funct3 | 000 | funct7 | - | | rd | 0101 0（x10/a0） | rs1 | 0001 1（x3/gp） | rs2 | - | | imm | 1100 0011 0100（-972的补码） | | | | | | 指令功能 | (a0) ← (gp) - 972 | | | | |   指令1： 10098: c5018613 addi a2,gp,-944   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 机器码 | 16进制 | 2进制 | | | | | c5018613 | 1100 0101 0000 0001 1000 0110 0001 0011 | | | | | opcode | 001 0011 | funct3 | 000 | funct7 |  | | rd | 0110 0（x12/a2） | rs1 | 0001 1（x3/gp） | rs2 |  | | imm | 1100 0101 0000（-944的补码） | | | | | | 指令功能 | (a2) ←(gp) - 944 | | | | |   指令2： 1015c: 01f7d793 srli a5,a5,0x1f   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 机器码 | 16进制 | 2进制 | | | | | 01f7d793 | 0000 0001 1111 0111 1101 0111 1001 0011 | | | | | opcode | 001 0011 | funct3 | 101 | funct7 |  | | rd | 0111 1（x15/a5） | rs1 | 0111 1（x15/a5） | rs2 |  | | imm | 0000 0001 1111（0x1f） | | | | | | 指令功能 | (a5) ←(a5) >> 0x1f | | | | |   指令3： 100c4: 00012503 lw a0,0(sp)   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 机器码 | 16进制 | 2进制 | | | | | 00012503 | 0000 0000 0000 0001 0010 0101 0000 0011 | | | | | opcode | 000 0011 | funct3 | 010 | funct7 |  | | rd | 0101 0（x10/a0） | rs1 | 0001 0（x2/sp） | rs2 | 000 | | imm | 0000 0000 0000 | | | | | | 指令功能 | (a0)←(sp) | | | | |   指令4： 102a8: e9dff0ef jal ra,10144 <fadd>   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 机器码 | 16进制 | 2进制 | | | | | e9dff0ef | 1110 1001 1101 1111 1111 0000 1110 1111 | | | | | opcode | 110 1111 | funct3 |  | funct7 |  | | rd | 0000 1（x1/ra） | rs1 |  | rs2 |  | | imm | 1111 1111 1111 0100 1110 | | | | | | 指令功能 | 跳转到编号为10144的指令 | | | | | |

|  |
| --- |
| 2、汇编程序实验结果 |
| 要求：自行构造3个测试用例，对自行编写的汇编程序进行测试，并记录运行结果。 |
| **示例：**  【用例1】float1：-9.777 float2：-1.234  运行结果： |
| 【用例1】float1：-12.3 float2：-6.5  运行结果：-18.8 |
| 【用例2】float1：0.5 float2：0.5  运行结果：1.0 |
| 【用例3】float1：1000 float2：0.1  运行结果：1000.1 |

|  |
| --- |
| 3、思考与讨论 |
| （1）用自己的语言描述子程序的工作流程。  首先保护现场；  接着读取浮点数，包括提取符号位、提取阶码和提取尾数并添加隐含的1；  然后对齐尾数，并同步修改阶码；  接下来将对齐后的尾数相加；  规格化后组装结果并输出。  （2）进入子程序时为何要保护现场，以及子程序返回前为何要恢复现场？  因为在子程序运行之前主程序运行占用了寄存器，而后续子程序要使用寄存器可能会覆盖原本主程序的数据，因此先时主程序的数据入栈以保护现场；  子程序结束之后，需要将数据弹栈以恢复到原本的状态，即恢复现场。  （3）试对比分析编译器生成的汇编程序与自行编写的汇编程序各有什么异同。  同：代码逻辑实现思路相同；  异：   1. 编译器生成的在每一次使用寄存器后都访问了内存，将数据存入内存，而自行编写不会； 2. 编译器生成的汇编程序使用了大量伪代码如li、ret等，自行编写的没有使用伪代码； 3. 自行编写的汇编程序为各个步骤添加了注释，可读性更好，而编译器非必要不会添加注释。 |