**深 圳 大 学 实 验 报 告**

**课程名称：­ 计算机系统**

**实验项目名称： LC-3机器码编程试验**

**学院： 计算机与软件学院**

**专业： 计算机科学与技术**

**指导教师： 蔡晔**

**报告人： 吉虹颖 学号： 2024150269 班级： 1 2**

**实验时间： 2025年 4月 17 日 星期四**

**实验报告提交时间： 2025年 4 月 17 日 星期四**

**教务部制**

|  |
| --- |
| **实验目的：**  1. 掌握处理仿真工具LC-3软件的安装和使用方法。  2. 学会在LC-3仿真环境下编辑程序和转换为可执行目标程序的方法。  3. 学会在LC-3仿真环境下运行和调试程序的方法。 |
| **实验内容与实验要求:**  1. 安装LC-3仿真器  2. 利用LC3EDIT输入机器代码程序（0/1模式）并创建可执行目标程序。  3. 利用LC3EDIT输入机器代码程序（hex模式）并创建可执行目标程序。  4. 利用LC3EDIT输入汇编代码程序并创建可执行目标程序。  5. 利用LC-3 Simulator仿真器运用对应目标程序。  6. 学习和掌握断点，单步执行等调试方法和手段。 |
| **实验步骤**   1. 安装LC-3仿真器 2. 按照老师给的压缩包，解压后打开”LC301.exe”,然后点击”Unzip”,选择一个文件夹放入即可； 3. 在根据目录找到该文件夹，可见到”LC3Edit.exe” “Simulate.exe”等文件则说明LC—3压缩包已经安装完毕，接下来可以进行实验编译了；        1. 利用LC3EDIT输入机器代码程序（0/1模式）并创建可执行目标程序。 2. 先打开”LC3Edit.exe”,输入PPT中的二进制代码；      1. 点击右上角的”File”,点击”Save”;      1. 在文件名一栏输入”EX1.bin”,点击保存即可      1. 点击上方的”Translate”,点击”Convert Base 2”或者直接点击图标，即可在文件夹中生成一个”EX1.obj”,看到下底提示栏输出信息”Convert complete -0 error[s]”,即可      1. 在文件夹中可见到两个文件，分别为”EX1.bin”和”EX1.obj”,则说明已创建了目标程序；      1. 在“Simulate.exe”中运行程序，说明程序可运行且正常；      1. 利用LC3EDIT输入机器代码程序（hex模式）并创建可执行目标程序。 2. 先打开”LC3Edit.exe”,输入PPT中的十六进制代码；      1. 点击右上角的”File”,点击”Save”;      1. 在文件名一栏输入”EX2.hex”,点击保存即可      1. 点击上方的”Translate”,点击”Convert Base 2”或者直接点击图标，即可在文件夹中生成一个”EX2.obj”,看到下底提示栏输出信息”Convert complete -0 error[s]”,即可      1. 在文件夹中可见到两个文件，分别为”EX2.hex”和”EX2.obj”,则说明已创建了目标程序；      1. 在“Simulate.exe”中运行程序，说明程序可运行且正常；      1. 利用LC3EDIT输入汇编代码程序并创建可执行目标程序。 2. 先打开”LC3Edit.exe”,输入PPT中的汇编语言代码；      1. 点击右上角的”File”,点击”Save”      1. 在文件名一栏输入”EX3.asm”,点击保存即可      1. 点击上方的”Translate”,点击”Asmble ”或者直接点击图标，即可在文件夹中生成一个”EX3.obj”,看到下底提示栏输出信息”Convert complete -0 error[s]”,即可      1. 文件夹中可见到多个文件，与之前的两个实验不同，分别为”EX3.asm”和”EX3.obj”等,则说明已创建了目标程序；      1. 在“Simulate.exe”中运行程序，说明程序可运行且正常:      1. 利用LC-3 Simulator仿真器运用对应目标程序   Example 1: 利用LC-3 Simulator仿真器运用对应目标程序   1. 输入代码并创建文件： 2. 首先打开“LC-3Edit”中输入该问题的一段程序： 3. 点击右上角的”File”,点击”Save”：      1. 保存该文件，并将文件命名为“mutiply.bin”，单击转化为.obj文件，可在文件夹中看到两个文件： 2. 运行程序： 3. 开启模拟器“Simulator.exe”，可见到两个窗口，一个Consule(模拟输入和输出设备)和一个主窗口；左上方是寄存器状态，右上角是机器状态，下方是内存：      1. 单右上角的，在文件夹中找到 “multiply.obj”，点击“打开”： 2. 然后可以看到如下界面：红框勾起来的部分是模拟器的部分内存内容，蓝色箭头指向x3200，这是我的程序的第一条指令，也就是即将被执行的指令；      1. 设置变量的值：点击，弹出的“Set Value”窗口，在Location字段中选择R4，在Value字段中输入x5，点击Apply；然后再Location字段中选择R5，输入x3,点击OK：      1. 点击，运行程序，点击后会弹出一个窗口，点击OK就好，然后去看上方寄存器中的R2（后面的是R2的值），结果是20，但是原本的答案应该是3，所以说明结果出错了，故应该进行调试：      1. 调试程序：   调试程序有两种方法，第一种是逐行调试，第二种是利用断点循环调试，下面我对两种方式都进行了尝试，所以写下我的调试步骤：   1. 逐步调试： 2. 因为上文中已经将程序运行了一遍，程序运行正常确保了每条指令都正常执行； 3. 首先打断点，双击x3204前的灰色小圆点，则会变成一个红色大圆点，证明其为断点：      1. 双击R5，在弹窗中设置R5为x3, R4设置为x5,然后点击OK；然后点击蓝色箭头，在弹窗中设置PC的值为x3200,现在蓝色箭头指向第一行，两个寄存器化为目标值，接下来可以进行调试程序了：        1. 点击，step over,这是PC指向X3201，IR存放着第一条指令X54A0，指令为：将R2清零；再次点击，PC指向下一条指令，IR存放着第二条指令，指令为：将R2赋值为5；之后连续两次点击，可以看见R5变为x3,再变为x2, 再变为x1;此时的R5既是乘数也是计数器（每次循环结束会自减）：        1. 反复调试，并不能发现错误在何处触发分支指令的执行，每次分支指令执行，二者必选其一，此时，分支被执行，因为状态码被add指令设置，add之后结果为x2，正数，因此状态码P为1，如果状态码中Z或P为1，分支执行，因此分支被执行，此时PC指向X3201，等待下一次循环。每一次循环都没有问题，故猜测错误出现在循环位置处 2. 利用断点调试：   在分支指令处设置断点，在每一次循环迭代的结尾处都会暂停，这样可以查看寄存器状态   1. 首先打断点：点击，在弹出的窗口里，将Location选择PC，Value输入x3203，然后点击ADD，断点列表中有两条信息都与PC有关（当PC为x3024和x3203时，模拟器都会暂停），然后点击OK；      1. 设置初值：点击，将Location选择PC，Value输入x3200,将Location选择R5，Value输入x3:      1. 运行程序：点击，会弹出一个弹窗，点击OK关掉弹窗，注意观察寄存器的值；蓝色箭头和PC均指向x3203，R4未变，R5则变为x2，R2变为x5；状态码p为1，意味着继续执行程序时，分支会跳转。   点击，关闭弹出的窗口，观察寄存器，目前已经循环两次，R2变为x1，R2变为xA（十进制10），状态码p为1，因此循环将继续执行。  点击，关闭弹出的窗口，观察寄存器，此时R5变为0，R2为十进制15，因为3\*5=15，此时应该停止，但是状态码Z为1，分支指令将继续执行，多做一次，这里出了问题。     1. 修改错误：通过修改分支指令使只有当p = 1时，循环就会执行正确的次数，为验证其正确性，在”LC3Edit”修改第四行的分支指令如下：“0000 001 111111101 ;跳转到 location x3201”：     点击右上方的“File”和“Save”;然后点击，重新生成”multiply.obj”文件；然后在“Simulator.exe”中，点击，选择“multiply.obj”文件，打开它：     1. 再次点击，弹出的“Set Value”窗口，在Location字段中选择R4，在Value字段中输入x5，点击Apply；然后再Location字段中选择R5，输入x3,点击OK：      1. 点击，运行程序，点击后会弹出一个窗口，点击OK就好，然后去看上方寄存器中的R2（后面的是R2的值），发现答案为15，是正确答案，调试成功；      1. 学习和掌握断点，单步执行等调试方法和手段。   Example 2: 本程序的目的是让用户输入两个数（0到9），然后求和，然后打印（同样介于0和9）在Console窗口中。   1. 输入代码并生成创建可运行文件： 2. 首先打开“LC-3Edit”中输入该问题的一段程序：      1. 点击右上角的”File”,点击”Save”：      1. 保存该文件，并将文件命名为“example.asm”，单击转化为.obj文件，可在文件夹中看到多个文件：        1. 运行程序和调试程序： 2. 开启模拟器“Simulator.exe”，可见到两个窗口，一个Consule(模拟输入和输出设备)和一个主窗口；左上方是寄存器状态，右上角是机器状态，下方是内存； 3. 点击，在文件中找到“example.obj”,并打开它：      1. 见图：halt在x3008行，从x3009行开始，你会在每行都能看到一个ASCII码值。在x3009行，看到x54，这个是表示字母T的ASCII码。在x300A行，看到x68，代表“h”的ASCII码。整个字符串，“The sum of those two number is”存储在内存地址从x3009到x3028中，最后一个地址存储着空格；      1. 双击x3008前的小灰色圆点，变为红色的一个断点，然后点击，运行程序：      1. 在“consule”窗口中显示“input a character>”,提示用户输入字符：   它会一直等待我输入，且下方“\_\_\_\_\_instructions executed”会一直变化：       1. 我先输入4， 传值给R1，consule提示我再次输入一个数，然后输入3，观察寄存器R1 R0 和Consule的变化；x0034是4的ASCII码，x0033是3的ASCIII码，x0067是g的ASCIII码；但是3 + 4 = 7，所以此时出错了      1. 修改错误： 2. 分析： 在调试过程中发现，当通过控制台输入数字字符时，程序并未正确处理其ASCII编码与实际数值之间的转换。例如：输入字符4时，寄存器R0中存储的是其ASCII码值x34;输入字符3时，R0存储的值为x33。若直接对这两个ASCII值进行加法运算，结果x67对应的ASCII字符为g，因此程序输出g而非预期的数值结果7。 该问题的根本原因在于，数字字符0至9在ASCII编码中分别对应x30至x39，而程序未剥离高4位的x30以获取真实数值。解决方法是在运算前，减去x30（或”0”）将ASCII码转换为对应的数字值, 此时再进行加法运算（4+3=7），即可得到正确结果。   这一修正需在程序的数据处理段添加相应指令，确保输入字符转换为数值后再参与运算。   1. 打开“LC-3Edit”,输入修改后的代码：   需要添加5个指令：两个是载入两个mask，一个是将-x30加到第一个数中，一个是对另一个数做同样的操作，最后一个是在输出前将结果加上x30:       1. 点击右上方的“File”和“Save”;然后点击，重新生成”example.obj”文件；然后在“Simulator.exe”中，点击，选择“example.obj”文件，打开它：      1. 首先调试一下修改后的代码，查看一下寄存器中的数值；双击x3008前的小灰色圆点，变为红色的一个断点，然后点击，运行程序，分别输入3，4；观察到寄存器R0 R1 R2中数值为3 4 7，正确了；      1. 此时再删除断点，完整的运行程序，点击，先后输入3，4，再consule中查看结果，结果正确，说明调试修改成功。 |
| **实验结论：**  在本次实验中，我深入学习并实践了LC-3仿真器的安装、使用以及调试方法。通过对LC3Edit工具的操作，我分别尝试以二进制代码（0/1模式**）**、十六进制代码（hex模式）和汇编语言（assembly）三种方式编写程序，并成功将其编译生成可执行目标程序文件（.obj）。这些目标程序随后被加载到LC-3 Simulator中进行调试与运行。  在整个过程中，我对LC-3的程序编辑器（LC3Edit.exe）和模拟器（Simulator.exe）的配合流程有了全面的认识，特别是理解了目标程序的生成过程，及其与高级语言开发环境（如 Dev-C++、Code::Blocks）中编译器与调试器之间的关系。  在LC-3模拟器中，我能够直观地观察程序运行过程中的内存状态、寄存器内容以及机器状态寄存器等底层信息，这大大加深了我对计算机底层工作机制的理解。通过设置断点、单步执行等调试手段，我能够逐条分析指令的执行效果，并对程序进行精确的错误定位和修改。  例如，在调试某个汇编程序时，我利用寄存器与内存窗口成功定位到了数据传输中的错误指令，并通过修改指令参数解决了程序运行异常的问题。这种调试方法不仅锻炼了我的逻辑分析能力，也提升了我对汇编程序运行过程的掌控力。  总的来说，本次实验使我不仅掌握了LC-3仿真环境的基本使用方法，还对编译原理、目标代码生成及调试流程有了更直观、深入的理解。这些技能为我今后学习更复杂的系统结构和底层编程打下了坚实的基础。 |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。