计算机体系设计实践报告

2021 学年秋季学期

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **学号** | **姓名** | **专业** | **班级** | **成绩** |
| 245 | 20211060245 | 陈俊宏 | 人工智能 | 人工智能 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

指导教师：武丽雯

软件学院 计算机体系设计实践 王逍

二 O 二二年十一月

# 计算机体系设计实践

**实验目的**

实践课设计 4 个实验，基本涵盖了计算机体系结构与组成课程的重点内容，实验电路的 设计与教材保持一致，实验电路在 FPGA 实验板上实现，并使用调试软件 JULAB 完成实验 的调试与分析。

**实验要求**

实验课前复习相关原理，按实验指导书的要求认真预习。 实验时独立思考，掌握实验设备或软件的构造和操作方法，按实验指导书要求设计或验

证实验内容，测试有关数据，分析相应的问题。

实验课结束时需要整理提交 fpga 工具箱，并检查工具箱附件是否缺失。

由于 fpga 工具箱数量有限，实验以小组形式完成，2 人 1 个小组，提交 1 份实验报告。

**实验资料环境和资料**

Win10+FPGA 设计软件 Quartus II、实验调试软件 Julab、实验相关设计文件（工程模板、 电路设计源文件、虚拟构图文件）等。请找实验知道老师索取。

**实验地点**

计算机硬件实验室：软件学院大楼 1428 室

# 实验 3 指令和寻址方式

实验操作

1. 下载实验资源

解压缩指令和寻址方式，得到两个文件，*Lab\_JUC2.sof* 是用来下载到 DE2-115 的实验模 型机 JUC2 电路文件，*JUC2.scc* 是给实验调试软件用的 CPU 配置文件。

2. 实验电路下载

点击桌面上的 QuartusII Programmer 图标，打开 QuartusII 编程器，点击添加编程文件按 钮（Add File），浏览选择前面下载的实验模型机 JUC2 电路文件 *Lab\_JUC2.sof*，点击 Start 按钮，下载到 DE2-115 实验板。

3. 验证实验电路功能

打开实验调试软件 JULAB3，选择模型计算机实验类型，在“CPU 数据通路”窗口，可 以看到模型机 JUC2 的数据通路图。在“文件”菜单下，选择“打开 CPU 配置”选项，浏 览选择前面下载的 *JUC2.scc*，文件中载入的观察信号出现在调试软件的“寄存器及总线信息 窗口”。

实验的验证需要使用“主存汇编/调试窗口”为模型机主存写入机器指令，在该窗口输 入的汇编指令会自动翻译成对应的机器指令并写入主存；完成的汇编指令可以导出到文件中 保存，也可以直接导入已有的汇编文件；更多详细内容参阅实验指导书第 5 章 5.4.2 实验系 统软件中的“主存汇编/调试窗口”内容。

实验的验证需要使用“主存信息显示窗口”以十六进制形式手工输入数据到模型机主存； 点击工具栏中的主存刷新按钮，可以在该窗口查看模型机主存各单元内容；主存内容可以导 入或导出；更多详细内容参阅实验指导书第 5 章 5.4.2 实验系统软件中的“主存信息显示窗 口”内容。

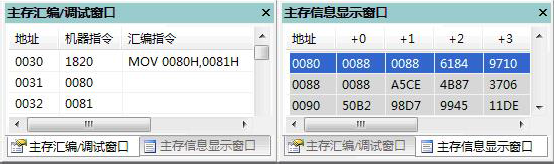
实验的验证要以机器指令单步的方式执行写入到模型机主存中的指令，点击工具栏中的 指令单步按钮执行指令，如果出错或想重新从头开始调试，工具栏中的复位按钮可以对模型 机硬件电路进行复位，更多调试方法参阅实验指导书第 5 章 5.4.2 实验系统软件中的“运行 及调试”内容。

## 实验记录

1. 基本寻址方式

将下面表格中的指令通过“主存汇编/调试窗口”输入到模型机的主存，以表格中的第 一条指令 MOV 0080H,0081H 为例，如下图左；执行前，在“主存信息显示窗口”中，为 主存的 0080H 单元手工输入数据 0088H，输入完成后以“指令单步”方式运行，“主存汇

编/调试窗口”会动态跟踪执行的机器指令，可以观察到光标移动到了下一条即将执行的指 令处。刷新主存内容，如果指令执行成功，在“主存信息显示窗口”的 0081H 单元，可以 看到指令的执行将 0080H 单元的数据 0088H 成功传送到了 0081H 单元，如下图右。



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 指令 | 执行前数据 | 执行后数据 | 结果分析 |
| ① | MOV 0080H, 0081H | (0080H)= 0088H (0081H)= 0000H | (0081H)= 0088H | 两个操作数的寻址方式都是直接寻址　该指令的功能是将　0080H　单元的内容传送到　0081H　单元 |
| ② | MOV #0080H, R0 | (R0)= 0000H | (R0)= 0080H | 源操作数的寻址方式是　立即寻址　，立即数包含在指令中，所在单元的地址是　8800H　，R0的内容即来自于该单元 |
| ③ | MOV (0080H), R1 | (0080H)= 0088H (0088H)= 0082H | (R1)= 0082H | 源操作数的寻址方式是　间接寻址　，0080H单元存放的是　有效地址　，R1寄存器内容是主存　0088H　单元的内容 |
| ④ | MOV (R1), R2 | (R1)= 0082H  (0082H)= 0000H | (R2)= 0000H | 源操作数的寻址方式是　间接寻址　，R1寄存器的内容是　有效地址　，R2寄存器的内容是主存　0082H　单元的内容 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ⑤ | MOV 8(R0), 0082H | (R0)= 0080H  (0082H)= 0000H | (0082H)= 0082H | 源操作数的寻址方式是　变址寻址　，有效地址的计算方法是　0080H+8H　，0082H单元的内容是主存　0088H　单元的内容 |

2. 移位、条件转移指令和相对寻址

将下面汇编语言程序手工翻译成机器指令，填写在横线上，在“主存信息显示窗口”， 将翻译好的机器指令，输入到模型机的主存，以“指令单步”方式运行。

1

2

3

4

5

6

7

ORG 0030H

0030: 1601 0505 ; MOV #0505，R1

0032: 6601 0001 ; AND #0001，R1

0034: 0378 0001 ; JNZ 1(PC)

0036: 0000 ;

0037: 0101 ;

HALT

ROL R1

0038: 0420 0032 ; JMP 0032H

下表已经给出了开始几条指令运行记录的内容，在后面的空白行上记录后续执行的指令

行号以及执行后的相关数据（如相关寄存器和 PSW 的变化），分析执行结果的意义（如程 序是否转移，转移的目的地址是多少），直到运行到 HALT 指令。

|  |  |  |
| --- | --- | --- |
| 指令 行号 | 指令执行后相关数据 | 结果分析 |
| 02 | (R1) = 0505H |  |
| 03 | (PSW) = 0  (R1) = 0001H | PSW 中的零标志位 Z= 0 。 |
| 04 | (PC) = 0037H | 发生 （发生／不发生）转移。相对寻址的有效地址  EA = (PC) + 偏移量，该指令计算有效地址时(PC) =  0037H，所以转移的目的地址是 0037H 。 |

3. 入栈和出栈指令

将下面汇编语言程序输入到模型机的主存，输入时注意根据指令的字长确定每条指令所 在的主存地址，以“指令单步”方式运行。观察堆栈指针 SP、堆栈存储单元以及相关寄存 器和内存单元的变化，记录在下表中，理解堆栈的用法。

。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 指令 | 执行前数据 | 执行后数据 | 结果分析 |
| ① | MOV #0041H，R0 |  | (R0)= 0041H |  |
| ② | PUSH R0 | (R0) = 0041H  (SP) = 0030H (002F) = 0000H | (SP) = 002F H (002F) = 0041H | 堆栈的第一个数据存放在主存0041H单元，其地址存放在R0  寄存器中。 |
| ③ | PUSH 0040H | (0040H) = 5555H (SP) = 002F H (002E) = 0000H | (SP) = 002E H  (002E) = 5555 H | 堆栈空间是朝着地址 减小／增 大 方向增长的，称作向上增长 |
| ④ | POP (R0) | (SP) = 002E H  (R0)= 0041H (0041H)= 0000H | (SP) = 002F H  (R0)= 0041H (0041H)= 5555H | 堆栈遵循 先进先出／后进先出 的原则。0041H 单元的内容是原 来 002E H 单元的内容。 |
|  |  |  |  |  |
| ⑤ | POP R1 | (SP) = 002F H  (R1)= 0000H | (SP) = 0030H  (R1)= 0000H | R1 的内容是原来 002F H 的内容。 |

4. 子程序调用和返回

下面的程序将 0038H 单元的内容读入寄存器 R1，调用子程序完成乘 2，返回主程序后 将结果保存到 0039H 单元。程序运行前需要先设置 0038H 单元的值，。

|  |  |  |  |
| --- | --- | --- | --- |
| 1 |  |  | ORG 0030H |
| 2 | 0030: | ; | MOV 0038H, R1 |
| 3 | 0031: |  |  |
| 4 | 0032: | ; | CALL 0040H |

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | 0033: |  |  |
| 6 | 0034: | ; | MOV R1, 0039H |
| 7 | 0035: |  |  |
| 8 | 0036: | ; | HALT |
| 9 |  |  | ORG 0040H |
| 10 | 0040: | ; | ADD R1, R1 |
| 11 | 0041: | ; | RET |

将上面的程序输入到模型机，将机器码填入横线上。单步运行，观察子程序调用和返回

前后的堆栈变化，填写下面的表格。

|  |  |  |  |
| --- | --- | --- | --- |
| 指令行号 | 执行前数据 | 执行后数据 | 结果分析 |
| 02 | (R1)= 0000H  (0038H)= 5050H | (R1)= 5050H | R1 的内容和 0038H 单元的数据一致。 |
| 04 | (SP) = 002F H (002F) = 0000H (PC) = 0040H | (SP) = 002F H (002F) = 0034H (PC) = 0041H | 执行后堆栈中存放的是返回地址，即  CALL 指令下面一条指令的地址。 |
| 11 | (SP) = 0030H  (PC) = 0036H | (SP) = 0030H  (PC) = 0036H | 执行后的 PC 内容来自于堆栈的栈顶单元， 即返回到 CALL 指令下面一条指令。 |
| 06 | (R1)= A0A0H | (0039H = A0A0 H | 0039H 单元的数据即 R1 寄存器的内容， 是 0038H 单元数据的 2倍 。 |

## 实验小结及实验分工

实验分工：单人完成

实验小结：实验要求掌握基本寻址方式；移位、条件转移指令和相对寻址；入栈和出栈指令；子程序调用和返回四个方面的应用原理，需要具备一定量的基础理论知识，实验本身来说只要细心且有耐心，不难完成。