



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
	20211120138	薛凡豪	网络空间安全		
	20211120148	罗安乔	网络空间安全		

指导教师：冯立波

软件学院 计算机体系设计实践 王逍

二〇二一年九月

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。

主存汇编/调试窗口			主存信息显示窗口				
地址	机器指令	汇编指令	地址	+0	+1	+2	+3
0030	1820	MOV 0080H,0081H	0080	0088	0088	6184	9710
0031	0080		0088	0088	A5CE	4B87	3706
0032	0081		0090	50B2	98D7	9945	11DE

	指令	执行前数据	执行后数据	结果分析
①	MOV 0080H, 0081H	(0080H)= <u>0088H</u> (0081H)= <u>0000H</u>	(0081H)= <u>0088H</u>	两个操作数的寻址方式都是 <u>直接</u> 寻址，该指令的功能是将 <u>0080H</u> 单元的内容传送到 <u>0081H</u> 单元
②	MOV #0080H, R0	(R0)= <u>0000H</u>	(R0)= <u>0080H</u>	源操作数的寻址方式是 <u>立即寻址</u> ，立即数包含在指令中，所在单元的地址是 <u>0034H</u> ，R0 的内容即来自于该单元
③	MOV (0080H), R1	(0080H)= <u>0088H</u> (0088H)= <u>0082H</u>	(R1)= <u>0082H</u>	源操作数的寻址方式是 <u>间接寻址</u> ，0080H 单元存放的是 <u>0088H</u> ，R1 寄存器的内容是主存 0088H 单元的内容
④	MOV (R1), R2	(R1)= <u>0082H</u> (<u>0082H</u>)= <u>0090H</u>	(R2)= <u>0090H</u>	源操作数的寻址方式是 <u>寄存器间接寻址</u> ，R1 寄存器的内容是 <u>有效地址</u> ，R2 寄存器的内容是主存 <u>0082H</u> 单元的内容

⑤	MOV 8(R0), 0082H	(R0)= <u>0080H</u> (0082H)= <u>0090H</u>	(0082H)= <u>0082H</u>	源操作数的寻址方式是 <u>偏移寻址</u> 有效地址的计算方法 是 (R0)+8 , 0082H 单元的内容是主存 <u>0088H</u> 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	ORG 0030H
2	<u>0030:1601, 0505</u> ; MOV #0505, R1
3	<u>0032:6601, 0001</u> ; AND #0001, R1
4	<u>0034:0378, 0001</u> ; JNZ 1(PC)
5	<u>0036:0000</u> ; HALT
6	<u>0037:0101</u> ; ROL R1
7	<u>0038:0420, 0032</u> ; 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) = <u>0505</u>	将立即数 0505 放入寄存器 R1 中
03	(PSW) = <u>0</u> (R1) = <u>0001</u>	PSW 中的零标志位 Z= <u>0</u> 。
04	(PC) = <u>0037</u>	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) = <u>0036</u> ，所以转移的目的地址是 0037_____。
05	0000	指令未运行
06	(R1)=0002	移位，原来 R1 中的 0001 移位为 0002
07	(PC)=0032	条件转移指令/发生跳转，跳到地址为 0032 中存的指令

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)= <u>0041</u>	
②	PUSH R0	(R0) = <u>0041</u> (SP) = <u>0030</u> (002F) = 0000	(SP) = <u>002F</u> (002F) = <u>0041</u>	堆栈的第一个数据存放在主存 002F 单元，其地址存放 <u>SP</u> 寄存器中。
③	PUSH 0040H	(0040H) = <u>5555</u> (SP) = <u>002F</u> (002E) = <u>0000H</u>	(SP) = <u>002EH</u> (002E) = <u>5555H</u>	堆栈空间是朝着地址 <u>减小</u> 方向增长的，称作向上增长
④	POP (R0)	(SP) = <u>002E</u> (R0)= <u>0041</u> (0041H)= <u>0000</u>	(SP) = <u>002F</u> (R0)= <u>0041</u> (0041H)= <u>5555</u>	堆栈遵循 <u>后进先出</u> 的原则。 0041H 单元的内容是原来 <u>0040H</u> 单元的内容。
⑤	POP R1	(SP) = <u>002F</u> (R1)= <u>0041</u>	(SP) = <u>0030</u> (R1)= <u>0041</u>	R1 的内容是原来 <u>R0</u> 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: <u>1801</u> ; MOV 0038H, R1
3	0031: <u>0038</u> _____
4	0032: <u>06A0</u> ; CALL 0040H
5	0033: <u>0040</u> _____
6	0034: <u>1060</u> ; MOV R1, 0039H
7	0035: <u>0039</u> _____
8	0036: <u>0000</u> ; HALT
9	ORG 0040H
10	0040: <u>2041</u> ; ADD R1, R1
11	0041: <u>0002</u> ; 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)= <u>0000</u> (0038H)= <u>1C20</u>	(R1)= <u>1C20</u>	R1 的内容和 0038H 单元的数据一致。
04	(SP) <u>0030</u> (002F)= <u>0000</u> (PC) = <u>0032</u>	(SP) = <u>002F</u> (002F) = <u>0034</u> (PC) = <u>0040</u>	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) = <u>002F</u> (PC) = <u>0041</u>	(SP) = <u>0030</u> (PC) = <u>0034</u>	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)= <u>2002</u>	(0039H) = <u>2002</u>	0039H 单元的数据即 R1 寄存器的内容，是 0038H 单元数据的 <u>2</u> 倍。

实验小结及实验分工

① 出现的问题及解决办法

1、在配置实验环境的过程中，一开始导入的 sof 文件有误，无法匹配所实验内容，导致无法进行实验，最后询问别人后，才意识到自己导入的时候，选择错了，应该要选择 JULAB/Logic/Lab_DE2-115_sof 中的 Lab_JUC2.sof 才可以成功进行实验。

2、实验过程中，一开始在进行复位、指令单步、主存刷新等步骤后，没有出现对应的实验结果，看到的全是 0，最后请教他人后，才出现来讲义上相对应的实验结果。

3、本次实验中，由于对实验原理不是很熟悉，导致在进行实验的时候浪费了很多时间。

② 收获及心得体会

1、本次实验需要进行“主存汇编”、“机器指令”、“地址”、“汇编指令”、“主存储器内容”以及“控制存储器内容”等窗口等的使用，需要对这些窗口有一定的了解。

2、通过本次实验，学习了解了复位、指令单步、主存刷新等操作，对指令和寻址方式这次实验有了更深刻的理解。

3、完成此次实验后，知道了指令的寻址方式有很多种，比如说跳跃寻址、直接寻址、间接寻址寄存器寻址等方式。

4、在本次实验中，花费了大量的时间，主要原因是对实验原理的不熟悉，以及实验操作过程中不太清楚导致，但在通过与他人交流后，完成了本次实验，希望自己可以吸取本次实验的经验，争取在下一次实验中做得更好。