

上海大学 计算机学院

《计算机组成原理实验》报告六

姓名 XXXXXXXXXX 学号 XXXXXXXXXX
时间 周二 9-11 机位 14 指导教师 刘跃军

实验名称: 机器语言程序实验

一、实验目的

- 1、编制机器语言简单程序。
- 2、成功运行机器语言程序。

二、实验原理

1、指令的形式化表示:

前几次实验课中我学习用中文自然语言表述了各种操作，中国人很容易理解它们，但印度人很难理解它们，机器也无法理解它们，而且自然语言用多了容易产生歧义，所以不适合用来表述大量的操作——写程序。另一方面，在实验三中我们使用微指令的二进制具体编码(16 进制数形式)来表示它，在实验五中用机器指令的微程序入口地址(16 进制数形式)来表示它们，这几种表示可以直接被机器“理解”，但人理解起来就非常吃力。表面上看，人理解 16 进制数形式指令很困难的原因是这些指令太相似了，其实根本原因是 16 进制数形式的指令没有对指令功能的任何反映，所以“人”用起来不方便，尤其是在使用大量指令来编制解决实际问题的程序时，这种不反映指令逻辑功能的形式更显得难以使用。如何解决这个问题呢？任何人都会想到一个简单的方法——给每条(微)指令再规定一个反映其逻辑功能的“符号”，习惯上称这套符号为“汇编指令”。

2、汇编:

由于汇编指令和 16 进制数指令是同一条指令的两种形式化符号，所以二者之间存在着一张符号对应表——翻译工具。厂家为实验箱的机器指令规定的两种形式化符号对应表见 101~102 页的表，其中“助记符”列就是“汇编指令”符号，“机器码 1”和“机器码 2”是指令的二进制数形式符号。有了这张对应表，人就可以用汇编指令符号来编写程序，以方便编程时的逻辑思维，编好程序后，只要对照这张表按顺序逐条将指令翻译成指令的二进制形式，然后将二进制数形式的程序送入计算机，计算机就能“理解”程序的逻辑操作了。这个翻译过程叫“汇编”。

显然，每个人都可以定义自己的“汇编指令”，只要你规定好符号“对应

表”就行了。所以，汇编指令“符号”和机器无关，而汇编指令的功能完全依赖于机器。

如果你高兴，完全可以编制一套中文符号的汇编指令。在这个符号系统，你可以用中文写程序。

3、模型机的指令类型简介：

助记符	机器码 1	机器码 2	注释
<u>FATCH_</u>	000000xx		实验机占用，不可修改，复位后，所有寄存器清0(R除外)，首先执行 <u>FATCH</u> 指令取指
	000001xx		未使用
	000010xx		未使用
	000011xx		未使用
ADD A,R?	000100xx		将寄存器 R? 的值加入累加器 A 中
ADD A,@R?	000101xx		将间址存储器的值加入累加器 A 中
ADD A,MM	000110xx	MM	将存储器 MM 地址的值加入累加器 A 中
ADD A,#II	000111xx	II	将立即数 II 加入累加器 A 中
ADDC A,R?	001000xx		将寄存器 R? 的值加入累加器 A 中，带进位
ADDC A,@R?	001001xx		将间址存储器的值加入累加器 A 中，带进位
ADDC A,MM	001010xx	MM	将存储器 MM 地址的值加入累加器 A 中，带进位
ADDC A,#II	001011xx	II	将立即数 II 加入累加器 A 中，带进位
SUB A,R?	001100xx		从累加器 A 中 减去寄存器 R? 的值
SUB A,K?	001100xx		从累加器 A 中 减去常数 K? 的值
SUB A,@R?	001101xx		从累加器 A 中 减去间址存储器的值
SUB A,MM	001110xx	MM	从累加器 A 中 减去存储器 MM 地址的值
SUB A,#II	001111xx	II	从累加器 A 中 减去立即数 II 加入累加器 A 中
SUBC A,R?	010000xx		从累加器 A 中 减去寄存器 R? 值，减进位
SUBC A,@R?	010001xx		从累加器 A 中 减去间址存储器的值，减进位
SUBC A,MM	010010xx	MM	从累加器 A 中 减去存储器 MM 地址的值，减进位
SUBC A,#II	010011xx	II	从累加器 A 中 减去立即数 II，减进位
AND A,R?	010100xx		累加器 A“与”寄存器 R? 的值
AND A,@R?	010101xx		累加器 A“与”间址存储器的值
AND A,MM	010110xx	MM	累加器 A“与”存储器 MM 地址的值
AND A,#II	010111xx	II	累加器 A“与”立即数 II
OR A,R?	011000xx		累加器 A“或”寄存器 R? 的值
OR A,@R?	011001xx		累加器 A“或”间址存储器的值
OR A,MM	011010xx	MM	累加器 A“或”存储器 MM 地址的值
OR A,#II	011011xx	II	累加器 A“或”立即数 II

算术运算指令

数据传输指令

MOV A,R?	011100xx		将寄存器 R? 的值送到累加器 A 中
MOV A,@R?	011101xx		将间址存储器的值送到累加器 A 中
MOV A,MM	011110xx	MM	将存储器 MM 地址的值送到累加器 A 中
MOV A,#II	011111xx	II	将立即数 II 送到累加器 A 中
MOV R?,A	100000xx		将累加器 A 的值送到寄存器 R? 中
MOV @R?,A	100001xx		将累加器 A 的值送到间址存储器中
MOV MM,A	100010xx	MM	将累加器 A 的值送到存储器 MM 地址中
MOV R?,#II	100011xx	II	将立即数 II 送到寄存器 R? 中
READ MM	100100xx	MM	从外部地址 MM 读入数据，存入累加器 A 中
WRITE MM	100101xx	MM	将累加器 A 中数据写到外部地址 MM 中
	100110xx		未使用
	100111xx		未使用
JC MM	101000xx	MM	若进位标志置 1，跳转到 MM 地址
JZ MM	101001xx	MM	若零标志位置 1，跳转到 MM 地址
	101010xx		未使用
JMP MM	101011xx	MM	跳转到 MM 地址
	101100xx		未使用
	101101xx		未使用

INT	101110xx		实验机占用，不可修改。进入中断时，实验机硬件产生_INT_指令
CALL MM	101111xx	MM	调用 MM 地址的子程序
IN	110000xx		从输入端口读入数据到累加器 A 中
OUT	110001xx		将累加器 A 中数据输出到输出端口
	110010xx		未使用
RET	110011xx		子程序返回
RR A	110100xx		累加器 A 右移
RL A	110101xx		累加器 A 左移
RRC A	110110xx		累加器 A 带进位右移
RLC A	110111xx		累加器 A 带进位左移
NOP	111000xx		空指令
CPL A	111001xx		累加器 A 取反，再存入累加器 A 中
	111010xx		未使用
RETI	111011xx		中断返回

图 1/2/3/4. 模型机的指令类型

三、实验内容

1. 实验任务一：编写并运行机器语言程序，将 R1 中的数值左移 n

次送 OUT，n 是 R2 中的数值。（指令码从 10 开始）

(1) 实验步骤

先进行对任务的分析、拆分，将其转换为汇编语言。基本的算法思想就是：在每次循环中，先检查 R2 寄存器中存储数据是否为 0，如果为 0，则直接将目前 R1 寄存器中存储的数据输出（防止 R2=0），如果不为 0，则将 R2 寄存器中存储数据自减 1，将 R1 寄存器中存储数据左移一位，如此循环往复，直到 R2 寄存器中存储数据为 0 时，跳出循环。最后只需输出 R1 寄存器中存储数据即可。具体的汇编语言代码如下所示：

表 1. 任务一汇编语言分析与机器码

汇编语言	机器码 1	机器码 2
LOOP:MOV A ,R2	72	/
ADD A ,#01	1D	01
SUB A ,#01	3D	01
JZ LOOP1	A4	1F
SUB A,#01	3D	01
MOV R2, A	82	/
MOV A, R1	71	/
RLA	D4	/
MOV R1,A	81	/
JMP LOOP	AC	10
MOV A ,R1	71	/
LOOP1. OUT	C4	/

(但是实验结束后，仔细想了想，为了使检验零标志位信息的指令生效，使 R2++ 再 -- 的两行红色代码操作可以用 SUB A ,#00 来实现，如此替换，可以在不影响结果的前提下，减少代码量，方便操作)

分析得出对应的机器码后，需要将机器码打入对应的内存单元中，按两次“TV/ME”键后，进入程序存储器，在 Adr 下打入 10，Data 中打入 72，按 NX，依次将下表中的值都打入，并检查是否输入正确。

表 2. 各内存单元对应机器码

单元	机器码	单元	机器码
10	72	19	82
11	1D	1A	71
12	01	1B	D4
13	3D	1C	81
14	01	1D	AC
15	A4	1E	10
16	1F	1F	71
17	3D	20	C4
18	01	/	/

至此，本实验任务的准备阶段与操作阶段结束，下面进入检验阶段。

再按 3 次“TV/ME”键，进入内部寄存器，在 uPC 一栏输入 10，按三下“F2/NX”键，将 R1、R2 分别赋值为 02、01，再按数次“F2/NX”键切回 uPC 界面，不断按下“STEP”按键，观察 R1、R2 寄存器上的数字变化以及 OUT 上的数字。

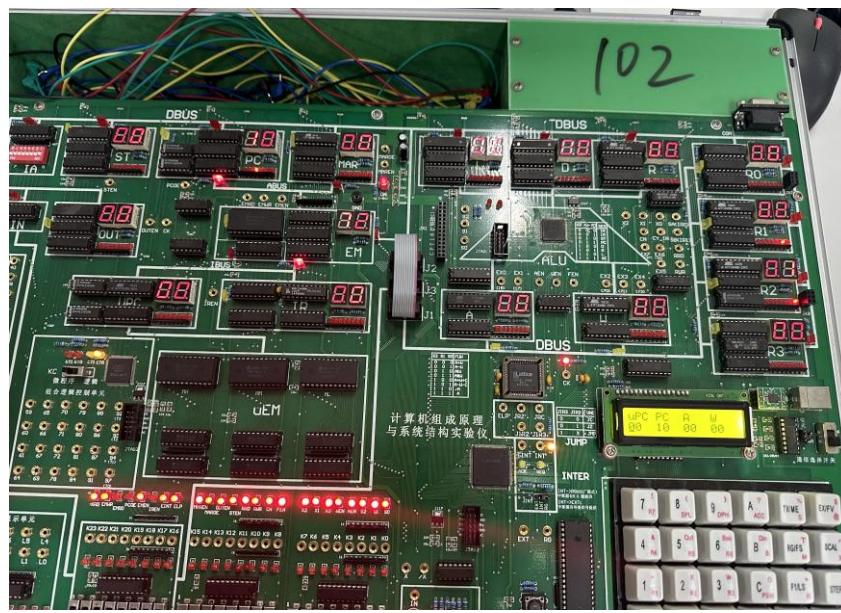


图 1. 实验任务一基础赋值、设置

(额外测试 R2=00 时, 能否正确输出结果, 所以增加一组实验, 在机器码、R1 及其他东西都不变的情况下, 将 R2 改成 00, 不断按下 STEP 按键, 观察结果)

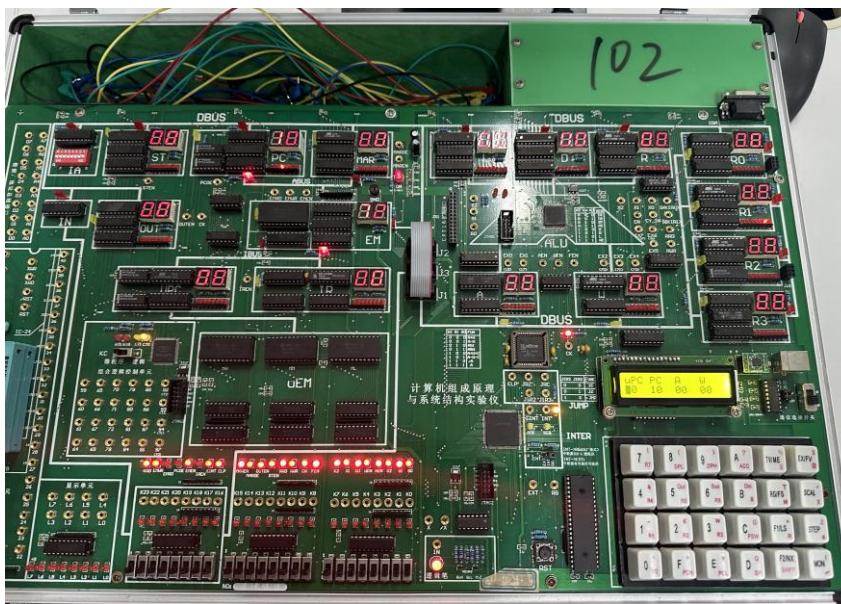


图 2. 实验任务一额外测试赋值、设置

(2) 实验现象

在按 STEP 的过程中, 观察到 R2 寄存器上的数字从 01 变成了 00, R1 寄存器上的数字从 02 变成了 04, 最终 OUT 输出结果为 04。具体见图 3。



图 3. 实验任务一测试结果

(额外测试: R2、R1 寄存器上的数字未产生变化, OUT 输出结果与 R1 相同, 为 02。具体见图 4。)

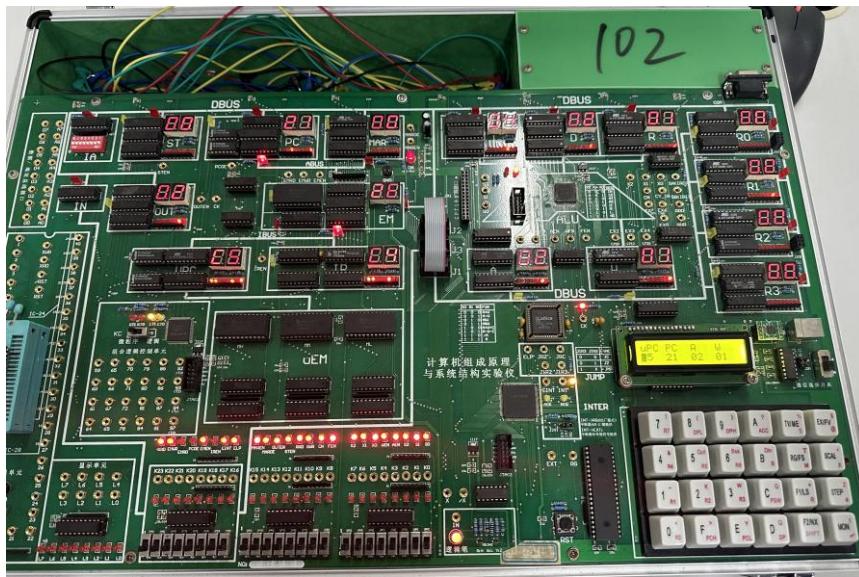


图 4. 实验任务一额外测试结果

(3) 数据记录、分析与处理

实验步骤及现象中已经分析、处理、记录过了数据, 在此不进行过多赘述。

(4) 实验结论

正确编写并运行机器语言程序, 实现了“将 R1 中的数值左移 n 次送 OUT, n 是 R2 中的数值(指令码从 10 开始)”的操作, 完成了实验任务一。

2. 实验任务二: 编写并运行机器语言程序, 完成 $R3=R0 \times R1$, 结果

送 OUT。 (指令码从 30 开始)

(1) 实验步骤

先进行对任务的分析、拆分，将其转换为汇编语言。基本的算法思想就是：在循环前，先检查 R0、R1 寄存器中存储的数据是否存在一个为 00，如果存在，直接输出 00 即可。如果没有 00，则进入下述循环：将 R3 中存储数据移到 A 寄存器中，把 R0 寄存器中数据加到 A 寄存器中，把 A 寄存器中数据移动到 R3 中（相当于 $R3 = R0$ ），把 R1 寄存器中数据移动到 A 寄存器中，将 A 寄存器中数据-1，再将 A 寄存器数据移到 R1 寄存器中（相当于 $R1 = R1 - 1$ ），检查 R1 是否为 0，如果 R1 为 0，则跳出循环，将 R3 寄存器中数据移动到 A 寄存器中，再送到 OUT 中。如果不为 0，则继续循环。

具体的汇编语言代码如下所示：

表 3. 任务二汇编语言分析与机器码

汇编语言	机器码 1	机器码 2
MOV A, R0	70	/
SUB A, #00	3C	00
JZ LOOP1	A4	48
MOV A, R1	71	/
SUB A, #00	3C	00
JZ LOOP1	A4	48
MOV A, #00	7C	00
LOOP:MOV A, R3	73	/
ADD A, R0	10	/
MOV R3, A	83	
MOV A, R1	71	/
SUB A, #01	3C	01
MOV R1, A	81	/
JZ LOOP1	A4	47
JMP LOOP	AC	3C
MOV A, R3	73	/
LOOP1:OUT	C4	/

分析得出对应的机器码后，需要将机器码打入对应的内存单元中，按两次“TV/ME”键后，进入程序存储器，在 Adr 下打入 30，Data 中打入 70，按 NX，依次将下表中的值都打入，并检查是否输入正确。

表 4. 任务二各内存单元对应机器码

单元	机器码	单元	机器码
30	70	3D	10
31	3C	3E	83
32	00	3F	71
33	A4	40	3C
34	48	41	01

35	71	42	81
36	3C	43	A4
37	00	44	47
38	A4	45	AC
39	48	46	3C
3A	7C	47	73
3B	00	48	C4
3C	73	/	/

至此，本实验任务的准备阶段与操作阶段结束，下面进入检验阶段。

再按 3 次“TV/ME”键，进入内部寄存器，在 uPC 一栏输入 30，按三下“F2/NX”键，将 R0、R1 分别赋值为 03、02，再按数次“F2/NX”键切回 uPC 界面，不断按下“STEP”按键，观察 R0、R1、R3 寄存器上的数字变化以及 OUT 上的数字。

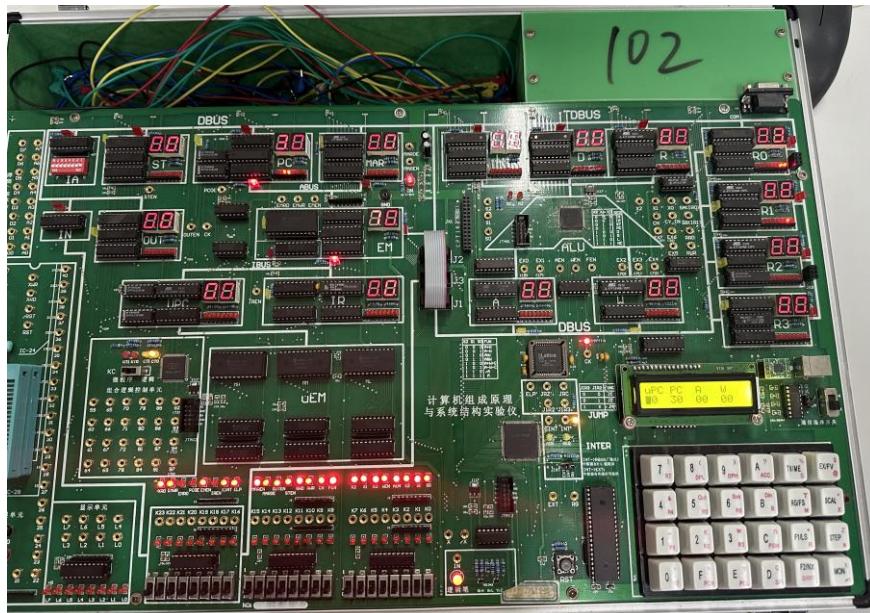


图 5. 实验任务二基础赋值、设置

(2) 实验现象

在按 STEP 的过程中，观察到 R1 寄存器上的数字从 02 变成 01，最终变成 00，R0 寄存器上的数字一直保持 03 不变，R3 寄存器中的数字从 00 变成 03，最终变成 06，最终 OUT 输出结果为 06。具体见图 6。



图 6. 实验任务二测试结果

(3) 数据记录、分析与处理

实验步骤及现象中已经分析、处理、记录过了数据，在此不进行过多赘述。

(4) 实验结论

正确编写并运行了机器语言程序，实现了“ $R3=R0 \times R1$, 结果送 OUT。”（指令码从 30 开始）的操作，完成了实验任务二。

四、建议和体会

1、用小键盘操作感觉要比连线方便不少，至少不会因为某根导线无法正常工作或者某一根导线错连而导致需要重新来过=--。

2、吸取了前一次的教训，提前预习，即在进行实验课前，需要对该节课实验的内容对应的理论知识（虽然好像与机组理论课关系不大）进行温故与复习并且对实验步骤进行大框架上的构想与思考，最好能够提前学会要用到的知识点。

3、TV/ME 真的非常非常重要！

五、思考题

Q: 建立“中文汇编指令”需要哪些条件？

A: 要先定义/翻译一个中文指令集，覆盖常见的计算机操作，如数据存取、运算、跳转、控制等。比如“移动”、“加/减”、“跳转”；设计一门与汇编语言类似的语法规则，且该语法与中文语言习惯需要相一致；还需要开发一个专门的中文汇编器，可以将中文汇编代码转化为机器语言，汇编器还需要能够解析中文指令并执行相应的操作……

(实验报告上传： 超星教学平台作业，请注意上传时间截止日期，一般为一个星期。请及时上传，晚交会影响报告成绩。)