

# 上海大学 计算机学院

## 《计算机组成原理实验》报告 6

姓名 孔馨怡 学号 22122128  
时间 周一 9-11 机位 25 指导教师 顾惠昌

---

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:

由于中文自然语言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。

附表：

助记符	机器码 1	机器码 2	注释
_FATCH_	000000xx		实验机占用，不可修改，复位后，所有寄存器清0(IR 除外)，首先执行 FATCH 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,#I I	000111xx	I I	将立即数 I I 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,#I I	001011xx	I I	将立即数 I I 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,@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,#I I	001111xx	I I	从累加器 A 中 减去立即数 I I 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?	010010xx		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,#I I	010011xx	I I	从累加器 A 中 减去立即数 I I，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,#I I	010111xx	I I	累加器 A“与”立即数 I I
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,#I I	011011x	II	累加器 A“或”立即数 I I
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,#I I	011111xx	I I	将立即数 I I 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?,#I I	100011xx	I I	将立即数 I I 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
		111100xx		未使用
		111101xx		未使用
		111110xx		未使用
		111111xx		未使用

### 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

1. 注视仪器，打开电源，手不要远离电源开关，随时准备关闭电源，注意各数码管、发光管的稳定性，静待 10 秒，确信仪器稳定、无焦糊味。

##### 2. 设计汇编语言：

根据题目要求确定算法，用汇编符号编程，然后查看指令对照表，完成源程序汇编。

指令	助记符 1	助记符 2	地址 1: 机器码 1	地址 2: 机器码 2
MOV	A	R1	10:71	
LOOP:				
RL	A		11:D4	

MOV	R1	A	12:81	
MOV	A	R2	13:72	
SUBC	A	#01H	14:4C	15:01
MOV	R2	A	16:82	
MOV	A	R1	17:71	
JZ	LOOP1		18:A4	19:1C
JZ	LOOP		1A:AC	1B:11
LOOP1:				
OUT			1C:C4	

### 3. 操作:

①把目标程序送入 EM 的 10 地址: 进入存储器模式 (按小键盘的 TV/ME 键, 液晶屏显示 “EM…”), 在 Adr 中送入 10, 在 Data 下写入 71; 按 NX 键进入 11, Data 写入 D4; 按 NX 键进入 12, Data 写入 81; 按 NX 键进入 13, Data 写入 72… (重复上述操作, 按照步骤 2 中设计的表格的将 14-1E 的数据送入)。即:

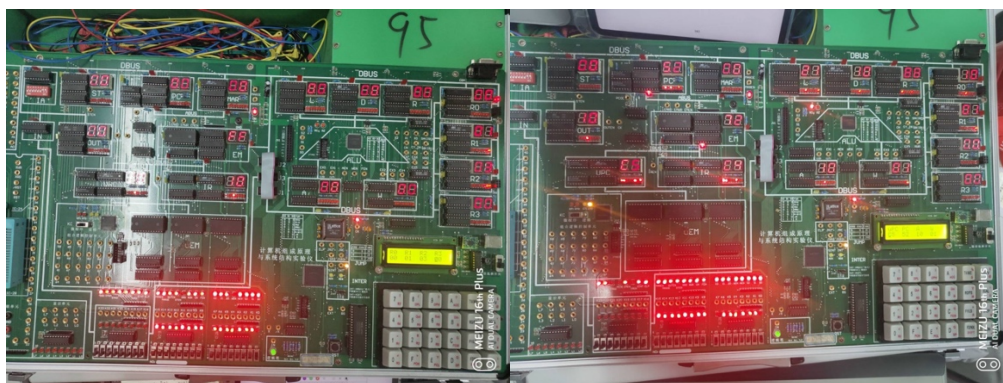
EM	Adr	Data
	10	71
	11	D4
	12	81
	13	72
	14	4C
	...	...
	1B	11
	1C	C4

②进入内部寄存器模式 (按 TV/ME 键, 切换到  $\mu$ PC 模式, 液晶屏显示 “ $\mu$ PC…”), PC 输入 10, 按 NX 键找到 R1, 键入 01, R2 键入 03。

③按 STEP 键, 观察 A 寄存器、R1 寄存器、R2 寄存器的示数变化。以及 D、OUT 示数。

### (2) 实验现象 (左图为初始状态, 右图为结果)

最终: R2 寄存器显示 00, R1 寄存器和 OUT 寄存器显示 10。



### (3) 实验结论

成功将 R1 的数值左移 R2 位后放入 OUT。

## 2. 实验任务零：（自己做的）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

#### 1. 设计汇编语言：

根据题目要求确定算法，用汇编符号编程，然后查看指令对照表，完成源程序汇编。

指令	助记符 1	助记符 2	地址 1: 机器码 1	地址 2: 机器码 2
LOOP				
MOV	A	R1	30:71	
ADD	A	#00	31:1C	
JZ	LOOP1		33:A4	34:42
MOV	R1	A	35:81	
MOV	A	R3	36:73	
ADD	A	R0	37:10	
MOV	R3	A	38:83	
MOV	A	R1	39:71	
SUB	A	#01	3A:3C	3B:01
ADD	A	R0	3C:1C	3D:00
JZ	LOOP1		3E:A4	3F:43
MOV	R1	A	40:81	
JMP	LOOP		41:AC	42:30
LOOP1				
MOV	A	R3	43:73	
OUT			44:C4	

#### 2. 操作：

①把目标程序送入 EM 的 30 地址：进入存储器模式（按小键盘的 TV/ME 键，液晶屏显示“EM…”），在 ADR 中送入 30，在 Data 下写入 71；按 NX 键进入 31，Data 写入 1C；按 NX 键进入 32，Data 写入 00；按 NX 键进入 33，Data 写入 A4…（重复上述操作，按照步骤 2 中设计的表格的将 14-1E 的数据送入）。即：

EM	Adr	Data
	30	71
	31	1C
	32	00
	33	A4
	34	42
	...	...
	43	73
	44	C4

②进入内部寄存器模式（按 TV/ME 键，切换到  $\mu$ PC 模式，液晶屏显示“ $\mu$ PC…”），PC 输入 30，按 NX 键找到 R0，键入 03，R1 键入 04。

③按 STEP 键，观察 A 寄存器、R0 寄存器、R1 寄存器的示数变化。以及 D、OUT 示数。

(2) 实验现象（忘记拍照了）

最终：R0 寄存器显示 03，R1 寄存器显示 00，A 寄存器和 OUT 寄存器显示 0C。

(3) 实验结论

成功将  $R3 = R0 \times R1$ ，结果送 OUT。

3. 实验任务二：编写并运行机器语言程序，完成  $R3 = R0/R1$ ，结果送 OUT。（指令码从 30 开始）

TIPS: R1 是 0 的时候  $OUT = FF$  ;  $R1=1$ ,  $OUT=R3$ ;  $R0/R1$  整除， $OUT=BE$ ;  
不整除， $OUT=$ 余数. 要求 R1R0 不能被改变

(1) 实验步骤

1. 设计汇编语言：

根据题目要求确定算法，用汇编符号编程，然后查看指令对照表，完成源程序汇编。

指令	助记符 1	助记符 2	地址 1：机器码 1	地址 2：机器码 2
MOV	A	R1	30:71	
SUBC	A	#01H	31:4C	32:01
JZ	ZERO		33:A4	34:52
JC	JIN		35:A0	36:4F
MOV	A	R0	37:70	
SUBC	A	R1	38:41	
MOV	A	R2	39:82	
JZ	BE		3A:A4	3B:5A
JC	YU		3C:A0	3D:54
MOV	A	R3	3E:73	
ADD	A	#01H	3F:1C	40:01
MOV	R3	A	41:83	
LOOP:				
MOV	A	R2	42:72	
SUBC	A	R1	43:41	
MOV	R2	A	44:82	
JZ	BE		45:A4	46:5A
JC	YU		47:A0	48:54
MOV	A	R3	49:73	
ADD	A	#01H	4A:1C	4B:01
MOV	R3	A	4C:83	
JMP	LOOP		4D:AC	4E:42

ZERO:				
MOV	A	#FFH	4F:7C	50:FF
OUT			51:C4	
JIN:				
MOV	A	R0	52:70	
OUT			53:C4	
YU:				
MOV	A	R2	54:72	
MOV	A	R3	55:73	
ADD	A	#01H	56:1C	57:01
MOV	R3	A	58:83	
OUT			59:C4	
BE:				
MOV	A	#BEH	5A:7C	5B:BE
OUT			5C:C4	

## 2. 操作:

①把目标程序送入 EM 的 30 地址: 进入存储器模式 (按小键盘的 TV/ME 键, 液晶屏显示 “EM…”), 在 Adr 中送入 30, 在 Data 下写入 71; 按 NX 键进入 31, Data 写入 4C; 按 NX 键进入 32, Data 写入 01; 按 NX 键进入 33, Data 写入 A4… (重复上述操作, 按照步骤 2 中设计的表格的将 14-1E 的数据送入)。即:

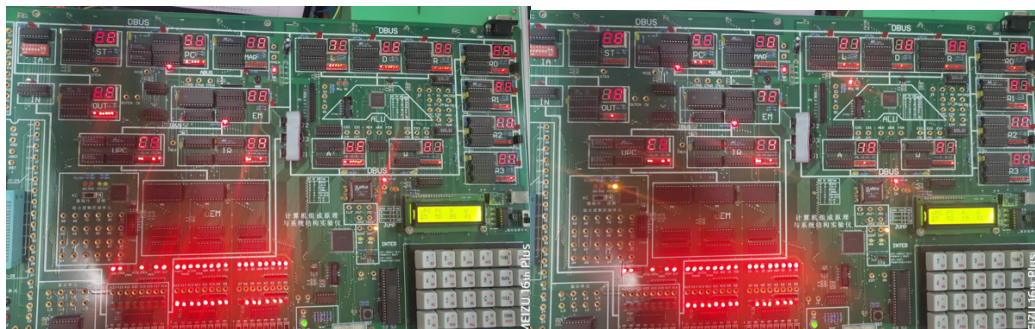
EM	Adr	Data
	30	71
	31	4C
	32	01
	33	A4
	...	...

②进入内部寄存器模式 (按 TV/ME 键, 切换到  $\mu$ PC 模式, 液晶屏显示 “ $\mu$ PC…”), PC 输入 30, 按 NX 键找到 R0 和 R1, 键入初始值。

③按 STEP 键, 观察 A 寄存器、R0 寄存器、R1 寄存器, R2, OUT 的示数变化。

## (2) 实验现象 (忘记拍照了, 只拍了 BE 的最终版和 01 的没有改错版本)

序号	R3=	R0/	R1	OUT
1	00	10	00	FF
2	10	10	01	10
3	02	04	02	BE
4	02	05	02	01



### (3) 实验结论

成功将  $R3 = R0/R1$ , 并按照不同结果送 OUT。

## 四、建议和体会

1. 由于没有学过汇编语言，需要对各指令作用及指令对照表进行学习理解。
2. 按 STEP 键的过程中，要时刻观察各个寄存器的示数变化情况，能够更加清晰地理解掌握指令语言及基层逻辑。

## 五、思考题

### 建立“中文汇编指令”需要哪些条件？

答：

1. 硬件支持：需要有一套能够识别和执行中文指令的硬件架构。
2. 汇编器的支持，及软件兼容性
3. 标准化和文档：为了使新的汇编语言能够被广泛接受和应用，需要制定相应的标准，并编写清晰完备的文档和教程，以便传播。