

上海大学 计算机学院

《计算机组成原理实验》报告 12

姓名 刘远航 学号 22121883

时间 20240501 机位 20 指导教师 刘跃军

实验名称：建立汇编指令系统

一、实验目的

1. 建立一个含中文助记符的汇编指令系统。
2. 用建立的指令系统编制一段程序，并运行之。

二、实验原理

1. 编制汇编指令

① 编制微指令对应的 24 个控制信号的电平；② 编制了 μEM 中从某地址开始的连续 4 个地址中的 24bit 值，即连续的四条微指令；③ 汇编指令是表达机器指令功能的指令助记符，二者对应关系由编制的汇编指令表确定。

按步完成这三个编制过程，就定义好一条全新的汇编指令，进一步也可以定义一个汇编指令系统——指令集。

汇编环境 CP226 考虑到了教学上定义汇编指令系统的需求，提供了完成这三个编制任务的集成环境，只要按规定的格式送入编制的符号，系统就会生成相应的汇编指令或汇编指令系统。

2. 汇编表文件

这个文件的后缀为.DAT,它是一个二维表格式文件，其每一行对应一条指令，这个表共有 3 列，如图 1。第一列是指令的汇编助记符，宽度为 20 个半角字符。第二列是指令的 16 进制编码形式（机器指令），在实验箱系统就是指令的微程序在 μEM 中的起始地址，宽度为 8 个半角字符。第 3 列是这条指令的字节数，宽度为 1 个半角字符，这是本表的重要汇编信息，也是设立本表的原因之一。

这个文件的主要作用是：当编译（汇编）源程序时，查此表把汇编指令翻译成机器指令。即这就是汇编表。构造这个表文件时也不能带标题行。

	20	8	1
ADD A, R0	10	1	
OUTA	14	1	

图 1. 汇编表文件格式

3. 微程序型指令文件

这个文件的后缀为.MIC,它也是一个二维表格式的文件，其每一行对应

一条微指令，这个表共有 11 列（字段），每一列都定义好了属性和宽度，例如：图 2 是指令集 insfile1.MIC 的格式。

这个表的主要作用是：当系统调用此文件时把其第 4 列“微程序”的内容送入其第 3 列“微地址”指定的 μ EM（微程序存储器）单元。即初始化 μ EM。

表的第一列为指令的汇编助记符,内容与表 1 的第 1 列一致。5 到 11 列是对本行微指令的说明，内容可以省略。

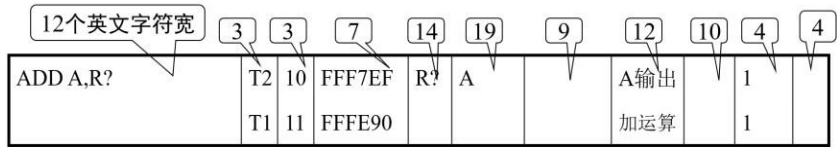


图 2. 微程序型指令文件格式

4. 指令的机器码文件

这个文件的后缀为.MAC,也是一个二维表格式文件，每一行对应一条指令，表共有 5 列，如图 3。第 1 列是汇编助记符，宽度 14，与表 1 的第 1 列一致。第 2 列是机器码 1，它是指令的微程序在 μ EM 中起始地址的二进制表示，其最后两位是对 R0~R3 的选择，所以与表 2 的第 3 列一致，宽度为 15。

第 3 列是机器码 2，是指令带的立即数或存储器地址。第 4 列是机器码 3，是指令带的第二个存储器地址，宽度 2。第 5 列是注释，宽度 100，用于对指令进行说明。实验箱默认指令系统 insfile1 没有机器码 3。

这个文件的主要作用是：解释汇编表的机器码细节，所以当编译源程序中的多字节指令时，可能要查此表。构造这个表文件时也不能带标题行。

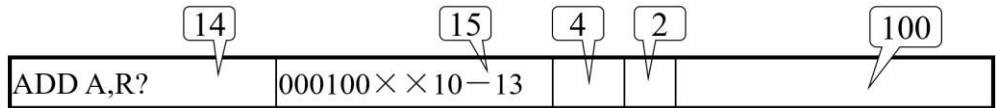


图 3. 机器码文件格式

三、实验内容

1. 实验任务一、编制一个汇编指令系统，包含下列助记符

指令助记符	指令意义描述
LD A,#*	将立即数打入累加器 A
A-W A,#*	累加器 A 减立即数
跳到 *	无条件跳转指令
OUTA	累加器 A 输出到 OUT
延时	延长显示时间

(1) 实验步骤

在 CP226 环境中，按照以下步骤进行操作，建立相关文件，为实验二做准备：

1、在系统文件中依次选择目录 C:\Program Files\CPTH+\DATA\。

2、在 DATA 目录中选择需要创建的文件 (.mic, .dat 或 .mac) 的模板(在这里我选择的是 infile2.mic, infile2.dat, infile2.mac 文件作为模板)。

3、将选中的三个文件复制到 D 盘, 以便后续修改, 防止改错之后重启电脑。

4、按照各个文件类型的格式依次为这三个类型的文件编写内容, 具体如下所示。(注意格式)

1.mac

FATCH 000000xx

LD A, #* 000001xx 将立即数放入到 A 寄存器

A-W A, #* 000010xx 累加器 A 减立即数

跳到 * 000011xx 无条件跳转指令

OUTA 000100xx 累加器 A 输出到 OUT

延时 000101xx 延长显示时间

1.mic

FATCH T0 00 CBFFFF

01 FFFFFFFF

02 FFFFFFFF

03 FFFFFFFF

LD A, #* T1 04 C7FFF7

T0 05 CBFFFF

06 FFFFFFFF

07 FFFFFFFF

A-W A, #* T2 08 C7FFE7

T1 09 CBFE91

T0 0A CBFFFF

0B FFFFFFFF

跳到 * T1 0C C6FFFF

T0 0D CBFFFF

0E FFFFFFFF

	0F FFFFFFFF
OUTA	T1 10 FFDF9F
	T0 11 CBFFFF
	12 FFFFFFFF
	13 FFFFFFFF
延时	T3 14 FFFFFFFF
	T2 15 FFFFFFFF
	T1 16 FFFFFFFF
	T0 17 CBFFFF

1.dat		
LD A,#*	04	2
A-W A,#*	08	2
跳到 *	0C	2
OUTA	10	1
延时	14	1

5、在下拉菜单中选择调入指令系统/微程序，将 1.mic 调入该程序。

(2) 实验现象

将 1.mic 调入程序后，下边的指令集已经变为，修改后的指令集，即成功完成了对原助记符的替换。

(3) 数据记录、分析与处理

助记符	机器码	指令意义描述
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
LD A, #*	000001xx	将立即数打入累加器 A
A-W A, #*	000010xx	累加器 A 减立即数
跳到	000011xx	无条件跳转指令

(4) 实验结论

成功编制一个汇编指令系统,包含相应助记符,达成实验任务一的目的。

2. 实验任务二、用所编制的指令系统,写出源程序,完成 OUT 寄存器交替显示 55, 22, 和 55-22 的值

(1) 实验步骤

- 1、在实验任务一基础上,打开实验箱电源,检验实验箱的基本功能是否正常。确认无误后,启动 CP226 软件。
- 2、检查 PC 机与实验箱的通信端口 COM4 是否连接正常。
- 3、调入 1.mic 指令系统,在 CP226 汇编语言程序集成开发环境下编写如下汇编程序:

```
LOOP: LD A, #55H
```

```
OUTA
```

```
延时
```

```
延时
```

```
延时
```

```
延时
```

```
LD A, #22H
```

```
OUTA
```

```
延时
```

```
延时
```

```
延时
```

```
延时
```

```
LD A, #55H
```

```
A-W A, #22H
```

```
OUTA
```

```
延时
```

```
延时
```

```
延时
```

```
延时
```

```
跳到 LOOP
```

END

- 4、保存文件，文件扩展名为.asm。
- 5、编译并下载源程序至实验箱，调试并运行程序，观察并记录实验结果。

(2) 实验现象

OUT 寄存器交替显示 55、22 和 33 的值。

(3) 数据记录、分析与处理

汇编程序的具体分析如下所示

LOOP: LD A,#55H	将立即数 55H 打入累加器 A
OUTA	累加器 A 输出到 OUT
延时	延长显示时间
延时	延长显示时间
延时	延长显示时间
延时	延长显示时间
LD A, #22H	将立即数 22H 打入累加器 A
OUTA	累加器 A 输出到 OUT
延时	延长显示时间
延时	延长显示时间
延时	延长显示时间
延时	延长显示时间
LD A, #55H	将立即数 55H 打入累加器 A
A-W A, #22H	累加器 A 减立即数 22H
OUTA	累加器 A 输出到 OUT
延时	延长显示时间
延时	延长显示时间
延时	延长显示时间
延时	延长显示时间

跳到 LOOP	无条件跳转程序，转跳 LOOP
END	汇编程序终止

(4) 实验结论

这次实验成功利用 CP226 软件编写汇编程序，并使用实验任务一所编制的指令系统，完成了实验任务二，实现了 OUT 寄存器交替显示 55、22 和 55-22 的值，达成了实验任务二的目的。

三、建议和体会

一定要注意（.mac，.mic，.dat）文件的**格式**要求，认真对照，以免出错导致运行出错，这次实验一开始没注意格式，一直出错，打击了自信心。

课前一定要仔细了解本节课所需要的知识，做好预习及复习，实验时一定要仔细思考，不能想当然地理解。我深入了解了汇编指令新系统，明白了助记符的真正含义，是用简洁而清晰的符号来完成指定的简单动作，深刻体会到了汇编语言与高级程序语言的关系，在今后的学习生活中我会更加清晰地明白计算机的具体工作原理。

四、思考题

问题：为什么汇编指令中可以用“中文符号”？

答：原有的英文符号仅是助记符，并不直接产生作用，其需要通过汇编指令系统转换为微指令码才能在机器中运行，而大部分中文字在计算机内都可以用 16 位 2 进制数表示。因此通过更改汇编指令系统的文件，可以更改汇编指令系统中的指令助记符，使其变为中文，实现在汇编指令中使用“中文符号”。