

上海大学 计算机学院

《计算机组成原理实验》报告十四

姓名 胡才郁 学号 20121034

时间 周四 9-11 机位 指导教师 刘学民

实验名称：课堂练习(4)

一、实验目的

1. 加强中文汇编指令系编写。
2. 加强汇编程序编写。

二、实验原理

1. 汇编表文件

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

图 1 汇编表文件格式

这个文件的后缀为 .DAT, 它是一个二维表格式文件, 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。

2. 微程序型指令文件

| | | | | | | | | | | |
|----------|----|----|--------|----|----|---|-----|----|---|---|
| 12个英文字符宽 | 3 | 3 | 7 | 14 | 19 | 9 | 12 | 10 | 4 | 4 |
| ADD A,R? | T2 | 10 | FFF7EF | R? | A | | A输出 | | 1 | |
| | T1 | 11 | FFFE90 | | | | 加运算 | | 1 | |

图 2 微程序型指令文件格式

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。

3. 指令的机器码文件

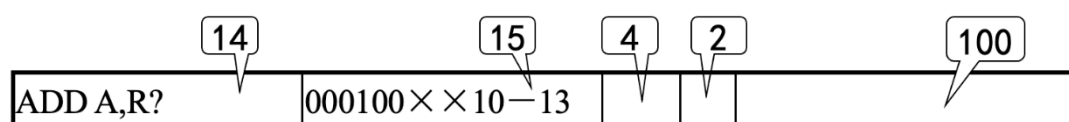


图 3 机器码文件格式

这个文件的后缀为 .MAC, 也是一个二维表格式文件, 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。

三、实验内容

1. 实验任务一:

编写一个独立的小型中文指令系统, 包括如下指令:

输入 A; 乘积 3*A; 跳到 *; 进位跳 *;
输出乘积; 赋值 A,#*。

使用一个演示程序来测试这些指令。当 3 倍的乘积无进位时, out 寄存器显示其值, 有进位时, 则显示 FF 值。

(1) 实验步骤

- ①打开集成编程环境。连接实验设备, 打开集成编程环境
- ②连接 PC 机与实验箱的通信口, 选择“COM4, 装入设计好的中文指令系统, 并打开编写好的后缀为.asm 的程序, 编译并下载源程序。
- ③运行程序, 给定不同的输入, 并观察实验箱中 OUT 寄存器的变化情况。

(2) 实验现象

当输入值较小, 此输入值乘 3 后未超过一个字节时, OUT 寄存器中显示此输入值乘 3 后的结果, 结果如下。

下面三张图分别为输入为 1、2、3 时, OUT 寄存器中的输出结果:

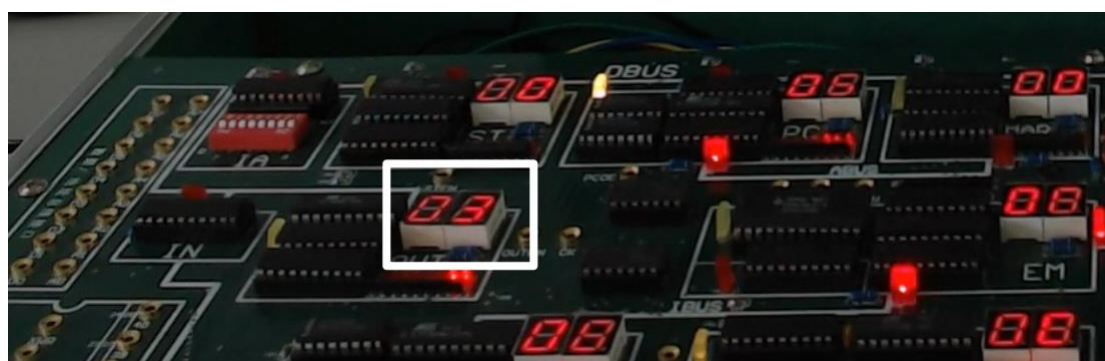


图 4 输入为 1 时 OUT 寄存器的示数

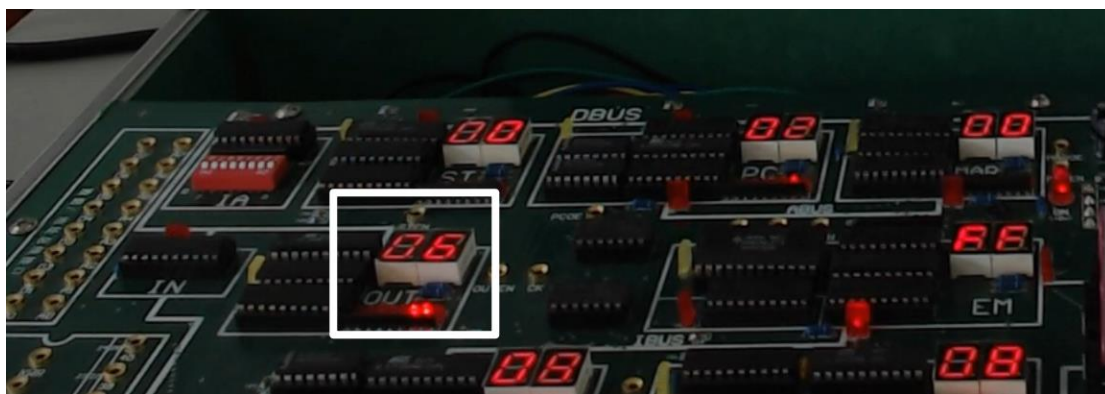


图 5 输入为 2 时 OUT 寄存器的示数



图 6 输入为 3 时 OUT 寄存器的示数

而当输入值较大时，若此输入值乘 3 后的结果超出一个字节可以表示的范围时，此时 OUT 寄存器中输出为 OUT。如下图所示：

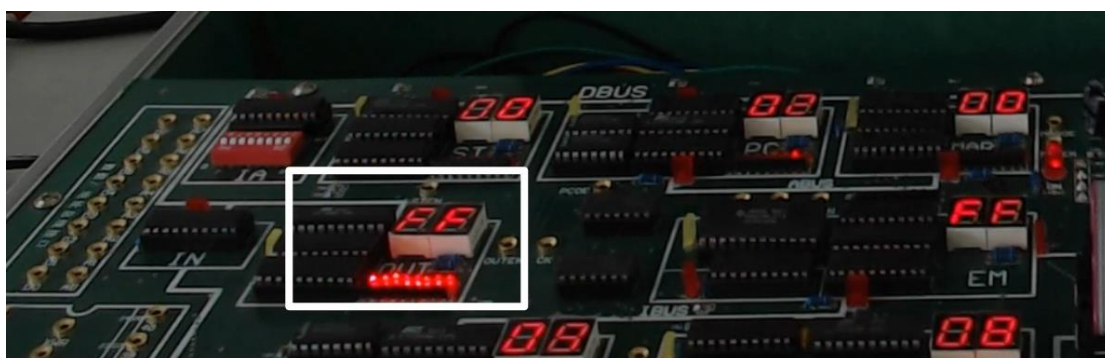


图 7 输入值过大时 OUT 寄存器的示数

(3) 数据记录、分析与处理

本实验的汇编指令系统使用中文汇编系统，此处的汇编指令可以使用中文符号，原因如下：

任何符号仅仅起到一个助记符的作用，计算机解析这一助记符，按照事先保存在计算机内的汇编表将助记符“翻译”为一条指令，一条指令又被“翻译”为不同的微指令进行执行。此处翻译成的伪指令与汇编指令的对应关系，在.MIC 文件之中定义。

所以即便是传统的英文，也需要经过解析才能被“翻译”为一条指令。中

文与英文都对应着计算机底层的内码，只是一个中文字符占两个半角，所以中文符号并不影响在内存中的存储与识别。在创建汇编系统时，在.DAT 文件中，只要你所设置的指令不超过规定的长度，中文符号和英文符号都是一样的，它们都仅仅只是一个助记符，没有实际含义，甚至可以使用其他特殊符号。既然如此，中文符号以及全世界各国语言的符号本身也都是一种符号，只要事先编写好符号与指令的对应表，都可以被计算机解析。

本实验的中文汇编指令系统的.MIC 文件如下表所示：

表 1 . MIC 文件

| | | | |
|----------|----|----|--------|
| _FATCH_ | T0 | 00 | CBFFFF |
| | | 01 | FFFFFF |
| | | 02 | FFFFFF |
| | | 03 | FFFFFF |
| LD A, #* | T1 | 04 | C7FFF7 |
| | T0 | 05 | CBFFFF |
| | | 06 | FFFFFF |
| | | 07 | FFFFFF |
| 乘积 3*A | T1 | 08 | FFFF8F |
| | T0 | 09 | FFFC7D |
| | | 0A | FFFE90 |
| | | 0B | CBFFFF |
| 跳到 * | T1 | 0C | C6FFFF |
| | T0 | 0D | CBFFFF |
| | | 0E | FFFFFF |
| | | 0F | FFFFFF |
| 进位跳 * | T1 | 10 | C6FFFF |
| | T0 | 11 | CBFFFF |
| | | 12 | FFFFFF |
| | | 13 | FFFFFF |
| OUTA | T3 | 14 | FFDF9F |
| | T2 | 15 | CBFFFF |
| | T1 | 16 | FFFFFF |
| | T0 | 17 | FFFFFF |
| 赋值 A, #* | T1 | 18 | C7FFF7 |
| | T0 | 19 | CBFFFF |

| | | | |
|--|--|----|--------|
| | | 1A | FFFFFF |
| | | 1B | FFFFFF |

其中，以指令“乘积 3*A”为例，分析对应的微指令执行过程。指令“乘积 3*A”分为 4 条微指令，除最后一条取值指令外，第一条将 A 寄存器中内容传送到 W 中，第二条将 A*2 的值传送到 A，此操作通过左移指令实现，第三条将 A 与 W 相加得到的结果放入 A 中。

表 2. 乘积 3*A 对应微指令

| | | | | |
|--------|----|----|--------|-----------|
| 乘积 3*A | T1 | 08 | FFFF8F | A → W |
| | T0 | 09 | FFFCDF | A * 2 → A |
| | | 0A | FFFE90 | A + W → A |
| | | 0B | CBFFFF | 取指 |

对于前三条微指令而言，将其后 12 位拆分转化位二进制如下表所示，其中，有 0 的位置均为控制信号有效，对应的控制信号如下表所示，因此，通过这些微指令可实现指令“乘积 3*A”的功能。

表 3. 乘积 3*A 对应微指令 0-11 位

| | | | |
|------|------|------|---------------------------|
| 1111 | 1000 | 1111 | X ₂₋₀ = 100 直通 |
| 1100 | 1101 | 0111 | X ₂₋₀ = 100 左移 |
| 1110 | 1001 | 0000 | FEN = 0 设置 CN |

微指令第 11 位到第 0 位控制信号如下表所示

表 4. 0-11 位控制信号

| | | | | | | | | | | | |
|-----|-----|----|-----|----|----|----|-----|-----|----|----|----|
| 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RRD | RWR | CN | FEN | X2 | X1 | X0 | WEN | AEN | S1 | S2 | S3 |

此实验的.asm 文件如下表所示：

表 5. .asm 文件

| | |
|-------|------------|
| LOOP: | 输入 A |
| | 乘积 3*A |
| | 进位跳 标号 1 |
| | 输出 |
| | 跳到 标号 2 |
| 标号 1: | 赋值 A, #FFH |
| 标号 2: | 跳到 LOOP |
| | END |

下面分析此程序如何实现本实验任务:

此程序为一个死循环,接收输入后,将输入值乘 3,此结果若超出一个字节,则执行“标号 1”中的复制语句,将 A 的值赋值为立即数 FFH。之后将结果在 OUT 寄存器之中输出。

(4) 实验结论

成功编制题目要求的中文汇编指令系统,并实现了在未产生进位的情况下输出 3*输入值,否则输出 FF 的结果。

四、建议和体会

诚惶诚恐中迎来了计算机组成原理实验课的尾声,对每次课堂老师在微信群中的耐心答疑充满感激。这两个学期的学习伴随着我个人对计算机组成原理了解的步步深入,一次次的实验帮助我理解理论课堂上学习的内容。

春季学期注定对我们来说是个难以忘记的学期,它有着两个学期的考试都堆在一起的紧张刺激,也有着疫情突如其来的措手不及。不知不觉,这也是我足不出校的整整第 90 天。查询了学校的健康之路我才发现,自 3 月 16 日以来,我已经做过了 40 次核酸,以及数不清多少次抗原。

由于学校三学期制度,使得我们的节奏进度较快,导致自己没有充足的时间去细细探究每一个计算机组成部件的设计思想、理念、应用的延申和拓展,让自己可以熟练掌握其应用。而疫情的限制,无法亲自动手做实验,实在是可惜。

最后,希望疫情早日结束,大家都能回到正常的生活轨迹之中。