

# 上海大学 计算机学院

## 《计算机组成原理实验》报告 6

姓名 严昕宇

学号 20121802

时间 2022. 2. 21

机位 22

指导教师 周时强

实验名称: 指令系统实验

### 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. 汇编语言

给每条(微)指令再规定一个反映其逻辑功能的“符号”。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。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。

#### 2. 机器指令查看及修改

- 1) 进入存储器模式,在 **Adr** 下送入目标单元的地址,在 **Data** 下写入指令;按 **NX** 进入下地址,在 **data** 下写入下一条指令;依次完成所有指令写入。
- 2) 注意:指令连续存放。

#### 3. 指令实验

| $\mu$ PC | PC | A  | W  |
|----------|----|----|----|
| 00       | 00 | 00 | 00 |

按 **TV** 选择到该模式,可实验存放于目标内存单元中的机器指令。  
 $\mu$ PC 对应的是该机器指令的入口地址。

**PC** 对应的是计数器,按需求初始化。

**A** 对应的是送入 **A** 寄存器中的数据, **W** 对应的是送入 **W** 寄存器中的数据,按 **NEXT** 对 **R0~R3** 寄存器的值进行设置。

当各个量输入完毕,按 **STEP** 键,送入时钟脉冲,(时需送入多个时钟脉冲以完成指令实验)。

### 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 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)

编写代码及对应机器码如下：

|               |       |
|---------------|-------|
| LOOP:MOV A,R1 | 71    |
| RL A          | D4    |
| MOV R1,A      | 81    |
| MOV A,R2      | 72    |
| SUB A,#01     | 3D 01 |
| MOV R2,A      | 82    |
| JZ LOOP1      | A4 1B |
| JMP LOOP      | AC 10 |
| MOV A,R1      | 71    |
| LOOP1:OUT     | C4    |

- ② 将机器码写入相应的内存单元：Adr 下打入 10，Data 中打入 72，按 NX，将下表中的值都打入，并检查是否输入正确。

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

注：当循环时，入口地址需对应好。

- ③ 在  $\mu$ PC 模式下验证功能。赋初值  $\mu$ PC(00) PC(10) R0(01) R1(02)。按 STEP 观察微指令执行的过程。
- ④ 记录实验过程和现象，关闭电源。

##### (2) 实验现象

给 R1、R2 赋值，R1(01) R2(02)  
左移 2 次后 OUT 输出 04H

##### (3) 实验结论

实现了实验目的 1 与 2。

成功编写并运行了能够将 R1 中的数值左移 n 次送 OUT 的机器语言程序，当 R1 寄存器初值为 01H，R2 寄存器初值为 02H 时，结果为 10H。

## 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

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)

编写代码及对应机器码如下：

|               |       |
|---------------|-------|
| 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    |

- ① 将机器码写入相应的内存单元：Adr 下打入 30，Data 中打入 70，按 NX，将下表中的值都打入，并检查是否输入正确。

| 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  |    |     |

注：当循环时，入口地址需要对应好。

- ② 在  $\mu$ PC 模式下验证功能。赋初值  $\mu$ PC(00) PC(30) R0(01) R1(02)。按 STEP 观察微指令执行的过程。
- ③ 记录实验过程和现象，关闭电源。

## (2) 实验现象

给 R1、R2 赋值，R1(01) R2(02)  
左移 3 次后 OUT 输出 02H

## (3) 实验结论

实现了实验目的 1 与 2。

成功编写并运行了能够计算  $R3=R0 \times R1$ , 结果送 OUT 的机器语言程序，当 R0 寄存器初值为 01H，R1 寄存器初值为 02H 时，结果为 02H。

# 四、建议和体会

## 体会

本次实验让我收获颇丰，体会良多。

经过本次实验，我深入掌握了利用简单的汇编语言实现一些基本操作的方法，并成功自己编制出了用于实现实验任务的汇编代码，在编制代码的过程中，我切身体会到高级程序设计语言的编译器将高级语言翻译为汇编语言的模样。同时我也对计算机如何运行我们的指令有了更深入地了解。

我还学会了如何通过查表来了解一个机器的汇编指令与机器指令的对应关系，我知道了汇编语言翻译为机器码的原理。同时通过这样的人工翻译，我对每一类型的计算机拥有自己的指令集这样的概念有了较为深入的理解。对我的底层处理能力有极大的提升。

# 五、思考题

**问题：建立“中文汇编指令”需要哪些条件？**

答：根据本实验有关内容，可以了解到，汇编语言仅仅只是一种符号和机器码之间的简单映射。所以如果要建立中文汇编指令，那么我们需要编写中文汇编指令的解析器，而这需要我们知道机器本身的指令集，换言之我们需要找到机器的机器指令表，然后将我们的中文符号与这张表上的指令进行对应，从而形成符号对应表并将其在硬件与软件上实现。