



# 计算机组成原理实验指导书

---

## Principles of Computer Organization Experiment Instruction Book

### 实验 6 复杂模型机实验

燕山大学软件工程系

# 实验 6 复杂模型机实验

## 6.1 实验目的

- (1) 在简单模型机的基础上，通过知识的综合运用，进行 5 条机器指令的微程序设计。
- (2) 进一步理解微程序控制器的工作原理，掌握指令与微指令的区别与联系。
- (3) 通过编写和调试微程序，提高研究与设计能力。

## 6.2 实验要求

- (1) 做好实验预习，读懂实验电路图，熟悉实验元器件的功能特性和使用方法。
- (2) 在实验前做好微程序设计的全部工作，实验时只进行调试与验证。
- (3) 按照实验内容与步骤的要求，独立思考，认真仔细地完成实验。

## 6.3 实验原理

本实验电路与简单模型机实验电路相同，电路如图 6.1 所示，电路详细说明请见实验 5，此处不再赘述。

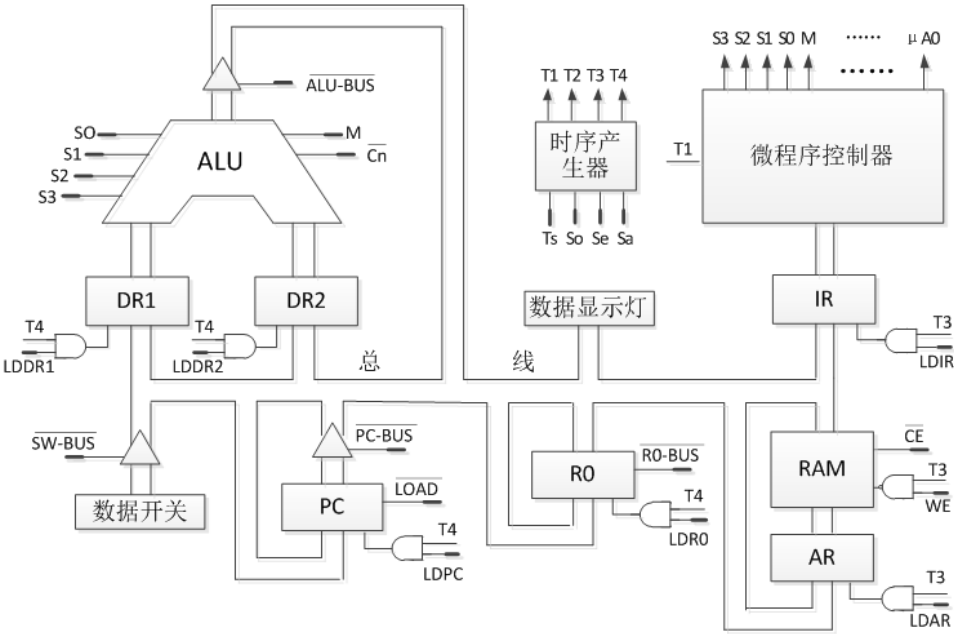


图 6.1 复杂模型机总框图

在简单模型机实验中，控制存储器里存放了 4 个微程序，对应 4 条指令，即此模型机只能运行 4 条指令。

本实验在简单模型机实验的基础上，实现一个包含 5 条指令的指令集。实验电路、指令格式都无需修改，只要修改控制存储器中的微程序，以及内存中的测试程序。

本实验用到的微指令微指令格式与简单模型机实验相同，如表 6-1 所示。

表 6-1 微指令格式

位	23	22	21	20	19	18	17	16	15	14	13	12
控制信号	S3	S2	S1	S0	M	Cn	CE	WE	LOAD	LDR0	LDDR1	LDDR2

位	11	10	9	8	7	6	5	4	3	2	1	0
控制信号	LDIR	LDPC	LDAR	ALU-B	PC-B	SW-B	R0-B	P(1)	μA3	μA2	μA1	μA0

### 6.4 实验内容与步骤

本实验的任务为：在基本模型机的基础上，将 ADD 指令修改为 SUB 指令，并增加一条 NOT 指令，NOT 指令功能为 $(NOT R0) \rightarrow R0$ 。实现表 6-2 所示的指令集。

表 6-2 机器指令格式

助记符	机器码（A 为内存地址 8bit）	长度
IN	000XXXXXX	8bit
SUB	001XXXXXX A	16bit
STA	010XXXXXX A	16bit
JMP	011XXXXXX A	16bit
NOT	100XXXXXX	8bit

实验步骤如下：

1. 根据 SUB 指令及 NOT 指令的功能要求修改微程序流程图，将下面的图 6.2 补充完整。注意安排好微指令的存储地址，请在所有方框的右上角用八进制标出地址。
2. 根据微程序流程图修改微程序代码表，将表 6-3 补充完整。
3. 修改测试程序，将表 6-4 补充完整，程序功能要求：在执行完减法后，将结果取反，然后再存入存储单元。
4. 运行虚拟实验系统，导入实验电路，打开电源。
5. 进行电路预设。将 DR1、DR2 和 AR 的  $\overline{MR}$  置 1，将计数器的  $\overline{CR}$ 、ENT、ENP 置 1，时序发生器的 Step 置 1（可在开电源之前设置）。微地址寄存器 74LS175 和指令寄存器 IR 的  $\overline{MR}$  置 1。

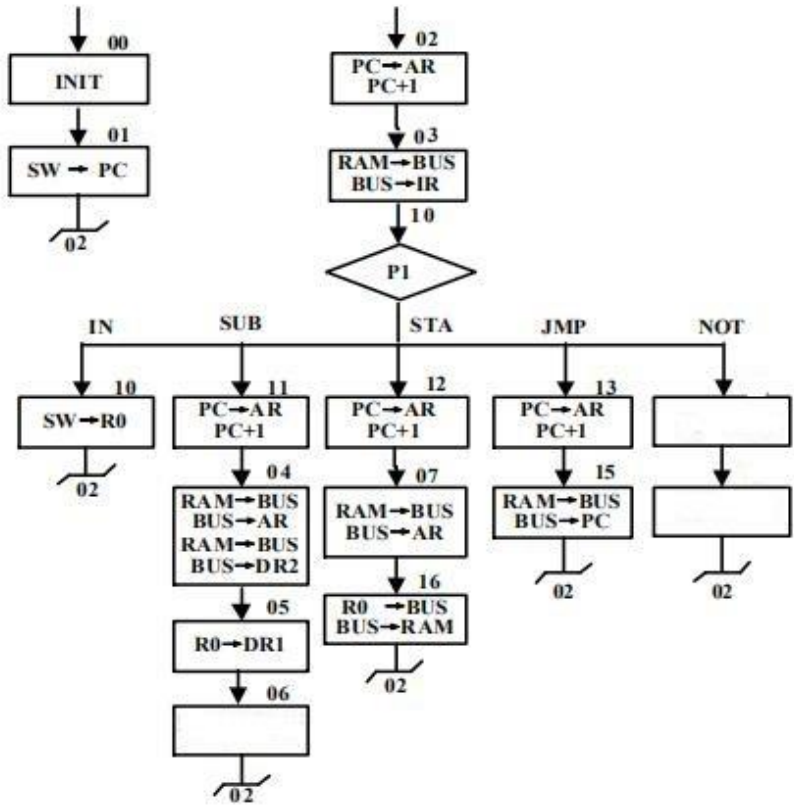


图 6.2 复杂模型机微程序流程图

表 6-3 微程序二进制代码表

位	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
地址	S3	S2	S1	S0	M	Cn	CE	WE	LOAD	LDR0	LDDR1	LDDR2	LDIR	LDPC	LDAR	ALU-B	PC-B	SW-B	R0-B	P(1)	uA3	uA2	uA1	uA0
00	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1
01	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	1	0
02	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	1	0	1	1	0	0	0	1	1
03	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	1	1	1	1	1	1	0	0	0
04	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	1	1	1	1	0	0	1	0	1
05	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	0
06																								
07	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	1	1	1	1	0	1	1	0	0
10	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0	1	1	0	1	0	0	0	1	0
11	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	1	0	1	1	0	0	1	0	0
12	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	1	0	1	1	0	0	1	1	1
13	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	1	0	1	1	0	1	1	0	1
14																								
15																								
16																								
17																								

表 6-4 RAM 中的程序和数据

地址（八进制）	内容	含义
00	00000000	IN（开关数据自定）
01		
02	00001000	10（八进制）
03		
04	01000000	STA
05	00001001	11（八进制）
06	01100000	JMP
07	00000000	跳转的目的地址
10	00001011	操作数
11		运行结果

6. 利用菜单“工具/存储器芯片设置”选项，修改控制存储器 EPROM2716 原有代码，写入新的微程序代码。
7. 利用菜单“工具/存储器芯片设置”选项，按照表 6-4 修改 RAM6116 的内容。
8. 在数据开关（SW7~SW0）上设置好程序起始地址（00000000），单步执行 3 条微指令。
9. 通过数据开关设置操作数 1 的值为 00010100。

10. 继续运行程序，观察执行过程是否与微程序流程图一致，以及最终结果是否正确。
11. 如果遇到错误，找到错误的原因，并修改至正确为止。

## 6.5 实验结果

按照实验步骤，将 1、2、3 步的结果填入相应位置。

## 6.6 思考与分析

1. 为使用微指令的 P 字段实现微程序分支，微指令存放的地址是否有限制？本实验中 NOT 指令的两条微指令是否可以存放任意地址中？
2. 微指令的编码格式有哪几种？本实验使用的是哪一种？
3. 若本实验还要再增加 2 条指令，是否需要增加或修改硬件？哪些部件需要修改？