****

**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。

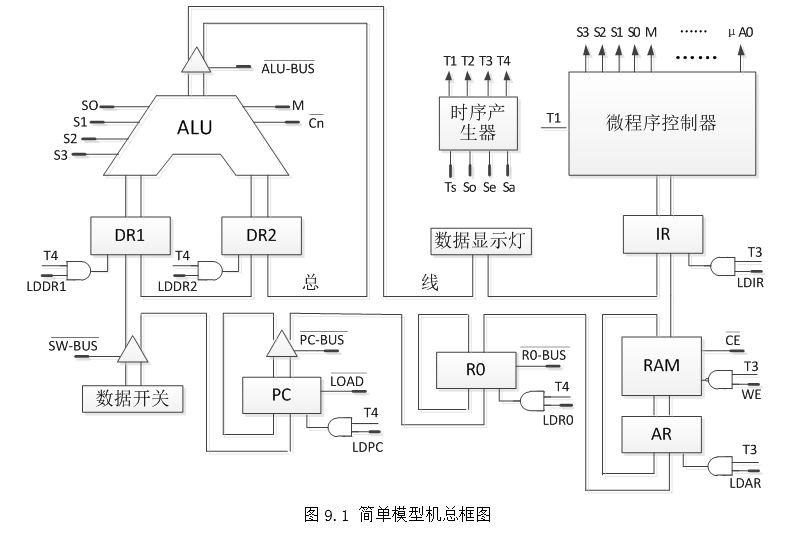


图6.1 复杂模型机总框图

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 |  |  | WE |  | LDR0 | LDDR1 | LDDR2 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 位 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 控制信号 | LDIR | LDPC | LDAR |  |  |  |  | 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)→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. 进行电路预设置。将DR1、DR2和AR的置1，将计数器的、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。

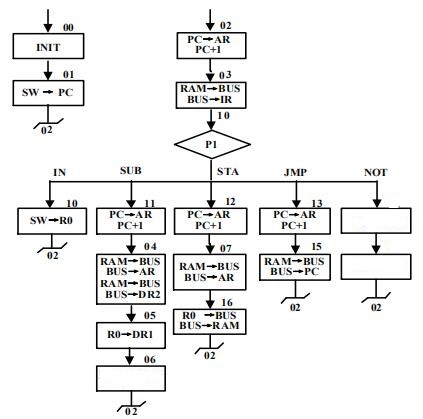


图6.2 复杂模型机微程序流程图

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) | μA3 | μA2 | μA1 | μA0 |
| 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 |

1. 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。
2. 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。
3. 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。
4. 通过数据开关设置操作数1的值为00010100.
5. 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。
6. 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到任意地址中？
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？