**西安电子科技大学**

**计算机组织与体系结构课程设计 课程实验报告**

**实验名称 基本模型机的设计与实现**

计算机科学与技术 学院 班

成 绩

姓名 学号

同作者

实验日期 年 月 日

实验地点实验批次

|  |
| --- |
| 指导教师评语：  指导教师：  年 月 日 |
| **实验报告内容基本要求及参考格式**  一、实验目的  二、实验所用仪器（或实验环境）  三、实验基本原理及步骤（或方案设计及理论计算）  四、实验数据记录（或仿真及软件设计）  五、实验结果分析及回答问题（或测试环境及测试结果） |

**计算机组织与体系结构课程设计**

**㈠ 基本**[**模型机设计与实现**](#_top)

**一**．**实验目的**

1．深入理解基本模型计算机的功能、组成知识；

2．深入学习计算机各类典型指令的执行流程；

3．学习微程序控制器的设计过程和相关技术，掌握LPM\_ROM的配置方法。

4．在掌握部件单元电路实验的基础上，进一步将单元电路组成系统，构造一台基本模型计算机。

5．定义五条机器指令，并编写相应的微程序，上机调试，掌握计算机整机概念。掌握微程序的设计方法，学会编写二进制微指令代码表。

6．通过熟悉较完整的计算机的设计，全面了解并掌握微程序控制方式计算机的设计方法。

**二．实验原理**

1．在部件实验过程中，各部件单元的控制信号是人为模拟产生的，而本实验将能在微过程控制下自动产生各部件单元控制信号，实现特定的功能。实验中，计算机数据通路的控制将由微过程控制器来完成，CPU从内存中取出一条机器指令到指令执行结束的一个指令周期，全部由微指令组成的序列来完成，即一条机器指令对应一个微程序。

2．指令格式

（1）指令格式

采用寄存器直接寻址方式，其格式如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 位 | 7654 | 32 | 10 |
| 功能 | OP-CODE | rs | rd |

其中，OP-CODE为操作码，rs为源寄存器，rd为目的寄存器，并规定：

|  |  |  |  |
| --- | --- | --- | --- |
| Rs或rd | | 选定的寄存器 | |
| 00 | | R0 | |
| 助记符 | 机器指令码 | | Addr地址码 | | 功能说明 | |
| IN  MOV addr  ADD addr  SUB addr  OUT addr | 0 0H  1 0H XX H  2 0H XX H  3 0H XX H  4 0H XX H | | | | “INPUT”中的数据→R0  R0 -> [addr]  R0 + [addr] -> R0  R0 – [addr] -> R0  [addr] → BUS | |

注：基本模型机无R1,R2寄存器，但实际电路中有R1,R2寄存器。

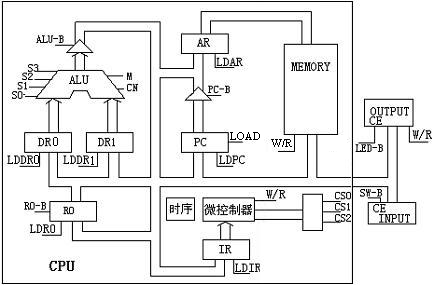


图1-1 数据通路框图

其中IN为单字长（8位二进制），其余为双字长指令，XX H 为addr对应的十六进制地址码。为了向RAM中装入程序和数据，检查写入是否正确，并能启动程序执行，还必须设计三个控制台操作微程序。

1、存储器读操作（KRD）：下载实验程序后按总清除按键（CLR）后，控制台SWA、

SWB为“0 0”时，可对RAM连续手动读出操作。

2、存储器写操作（KWE）：下载实验程序后按总清除按键（CLR）后，控制台

SWA、SWB为“0 1”时，可对RAM连续手动写操作。

1. 启动程序（RP）：下载实验程序后按总清除按键（CLR）后，控制台SWA、

SWB为“1 1”时，即可转入到微地址“01”号“取指令”微指令，启动程序运行。

根据以上要求设计数据通路框图，如图1-1所示。

|  |  |  |
| --- | --- | --- |
| SWB | SWA | 控制台指令 |
| 0  0  1 | 0  1  1 | 读内存（KRD）  写内存（KWE）  启动程序（RP） |

表1-1 24位微代码定义：

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 14 13 | 12 11 10 | 987 | 6 | 5 | 4 | 3 | 2 | 1 |
| S3 | S2 | S1 | S0 | M | Cn | WE | A9 A8 | | A | B | C | uA5 | uA4 | uA3 | uA2 | uA1 | uA0 |

**表1-2 A、B、C各字段功能说明：**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **A字段** | | | | **B字段** | | | | **C字段** | | | |
| 15 | 14 | 13 | 选择 | 12 | 11 | 10 | 选择 | 9 | 8 | 7 | 选择 |
| 0 | 0 | 0 |  | 0 | 0 | 0 |  | 0 | 0 | 0 |  |
| 0 | 0 | 1 | LDRi | 0 | 0 | 1 | RS-B | 0 | 0 | 1 | P（1） |
| 0 | 1 | 0 | LDDR1 | 0 | 1 | 0 |  | 0 | 1 | 0 |  |
| 0 | 1 | 1 | LDDR2 | 0 | 1 | 1 |  | 0 | 1 | 1 |  |
| 1 | 0 | 0 | LDIR | 1 | 0 | 0 |  | 1 | 0 | 0 | P（4） |
| 1 | 0 | 1 | LOAD | 1 | 0 | 1 | ALU-B | 1 | 0 | 1 | LDAR |
| 1 | 1 | 0 | LDAR | 1 | 1 | 0 | PC-B | 1 | 1 | 0 | LDPC |

**24位微代码中各信号的功能**

(1) uA5—uA0：微程序控制器的微地址输出信号，是下一条要执行的微指令的微地址。

(2) S3、S2、Sl、S0：由微程序控制器输出的ALU操作选择信号，以控制执行16种算术操作或16种逻辑操作中的某一种操作。

(3) M：微程序控制输出的ALU操作方式选择信号端。M＝0执行算术操作；M＝l执行逻辑操作。

(4) /Cn：微程序控制器输出的进位信号，/Cn＝0表示ALU运算时最低位有进位，/Cn＝1则表示无进位。

(5) WE：微程序控制器输出的RAM控制信号。当/CE＝0时，如WE＝0为存储器读；如WE＝1为存储器写。

(6) A9、A8——译码后产生CS0、CS1、CS2信号，分别作为SW\_B、RAM、LED的选通控制信号。

(7) A字段（15、14、13）——译码后产生与总线相连接的各单元的输入选通信号（见表6-1）。

(8) B字段（12、11、10）——译码后产生与总线相连接的各单元的输出选通信号。

(9) C字段（9、8、7） ——译码后产生分支判断测试信号P(1)~P(4)和LDPC信号。

系统涉及到的微程序流程见图1-2。当执行“取指令”微指令时，该微指令的判断测试字段为P(1)测试。由于“取指令”微指令是所有微程序都使用的公用微指令，因此P(1)的测试结果出现多路分支（见图1-2左图）。用指令寄存器的高4位（IR7-IR4）作为测试条件，出现5路分支，占用5个固定地址单元。

控制台操作为P(4)测试（见图1-2右图），它以控制台信号SWB、SWA作为测试条件，出现了3路分支，占用3个固定微地址单元。当分支微地址单元固定后，剩下的其它地方就可以一条微指令占用控制存储器的一个微地址单元，随意填写。

当全部微程序设计完毕后，应将每条微指令代码化，表1-3即为图1-2的微程序流程图按微指令格式转化而成的“二进制微代码表”。

**表1-3 二进制微代码表**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 微地址 | 微指令 | S3 | S2 | S1 | S0 | M | CN | WE | A9 | A8 | A | B | C | UA5-UA0 |
| 00 | 018110 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 000 | 000 | 100 | 010000 |
| 01 | 01ED82 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 110 | 110 | 110 | 000011 |
| 02 | 00C048 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 100 | 000 | 001 | 001000 |
| 03 | 00ED04 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 110 | 110 | 100 | 000100 |
| 04 | 038201 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 000 | 001 | 000 | 000001 |
| 05 | 00ED06 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 110 | 110 | 100 | 000110 |
| 06 | 00B007 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 011 | 000 | 000 | 000111 |
| 07 | 01A00D | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 010 | 000 | 000 | 001101 |
| 10 | 001001 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 001 | 000 | 000 | 000001 |
| 11 | 01ED83 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 110 | 110 | 110 | 000011 |
| 12 | 01ED85 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 110 | 110 | 110 | 000101 |
| 13 | 01ED8E | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 110 | 110 | 110 | 001110 |
| 14 | 01ED97 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 110 | 110 | 110 | 010111 |
| 15 | 919A01 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 001 | 101 | 000 | 000001 |
| 16 | 00ED0E | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 110 | 110 | 100 | 001110 |
| 17 | 00B015 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 011 | 000 | 000 | 010101 |
| 20 | 01ED92 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 110 | 110 | 110 | 010010 |
| 21 | 01ED94 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 110 | 110 | 110 | 010100 |
| 22 | 01A010 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 010 | 000 | 000 | 010000 |
| 23 | 018001 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 000 | 000 | 000 | 000001 |
| 24 | 062011 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 010 | 000 | 000 | 010001 |
| 25 | 01A016 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 010 | 000 | 000 | 010110 |
| 26 | 619A01 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 001 | 101 | 000 | 000001 |
| 27 | 00ED18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 110 | 110 | 100 | 011000 |
| 30 | 00A019 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 010 | 000 | 000 | 011001 |
| 31 | 070A01 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 000 | 101 | 000 | 000001 |

指令寄存器（IR）：指令寄存器用来保存当前正在执行的一条指令。当执行一条指令时，先把它从内存取到缓冲寄存器中，然后再传送至指令寄存器。指令划分为操作码和地址码段，由二进制数构成，为了执行任何给定的指令，必须对操作码进行测试“P(1)”，通过节拍脉冲T4的控制，以便识别所要求的操作。

指令译码器: 根据指令中的操作码强置微控制器单元的微地址，使下一条微指令指向相应的微程序首地址。

实验中LCD液晶显示屏可以用来显示模型机CPU中各组成单元的内容。将B100\_C.sof文件下载到实验台后，按系统复位键，LCD液晶显示屏即显示CPU中各组成单元的内容。其功能说明如下：

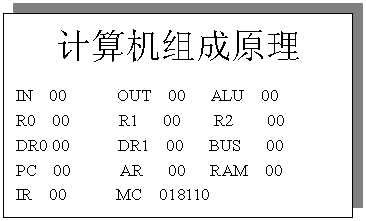
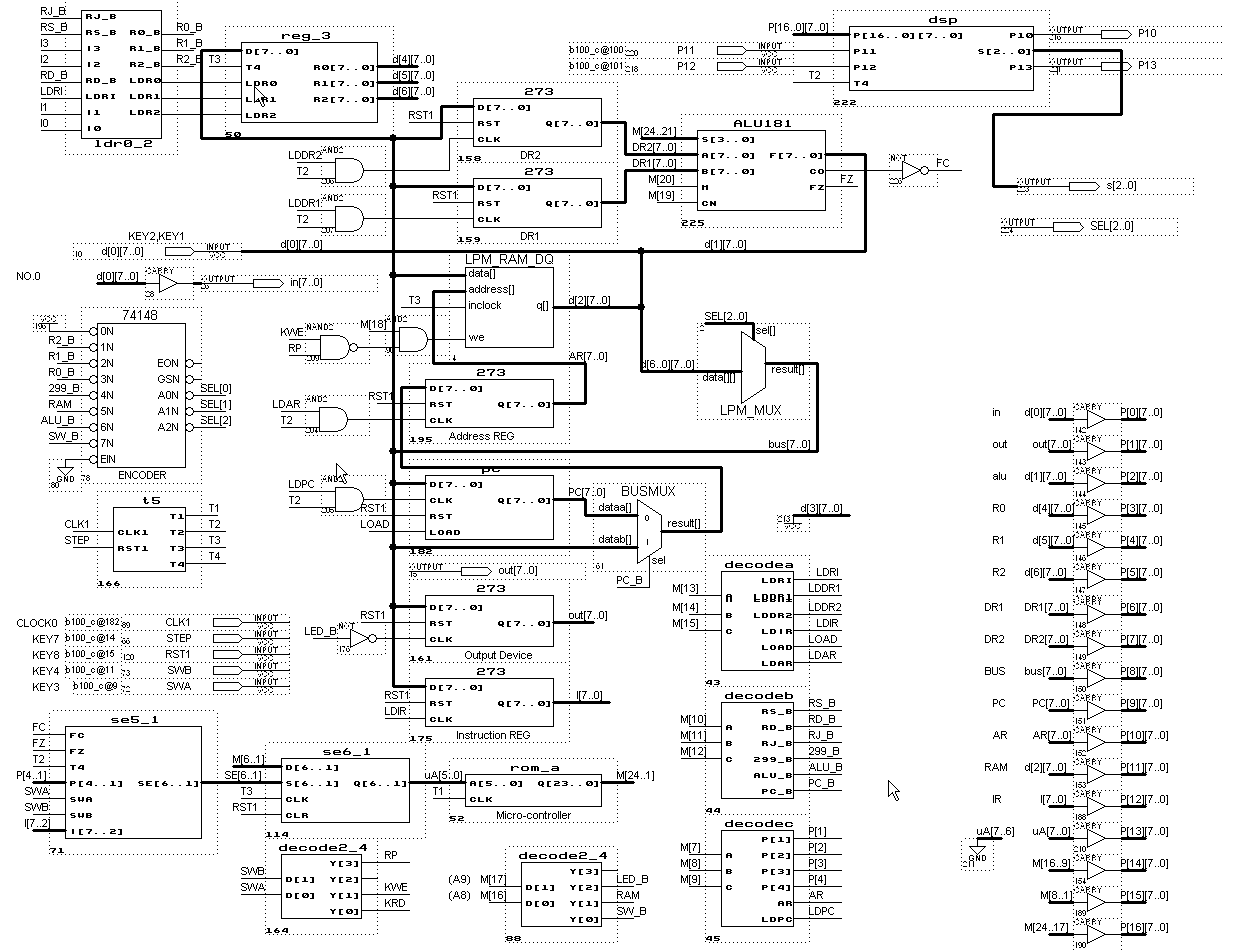


图1-2 LCD液晶显示屏

LCD液晶显示屏功能说明

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 作 用 | 名称 | 作用 |
| IN | 输入单元INPUT | DR1 | 暂存器DR1 |
| OUT | 输出单元OUTPUT | DR2 | 暂存器DR2 |
| ALU | 算术逻辑单元 | PC | 程序计数器 |
| BUS | 内部数据总线 | AR | 地址寄存器 |
| R0 | 寄存器R0 | RAM | 程序/数据存储器 |
| R1 | 寄存器R1 | IR | 指令寄存器 |
| R2 | 寄存器R2 | MC | 微程序控制器 |

 图1-3 模型计算机电路原理图

实验程序：

说明： 1、指令IN为单字节指令，指令码：00，其功能是将输入口IN的数据送到寄存 R0；

2、指令MOV [0AH]为双字节指令，指令码：100A，其功能是将R0中的数据送到RAM地址0AH中；

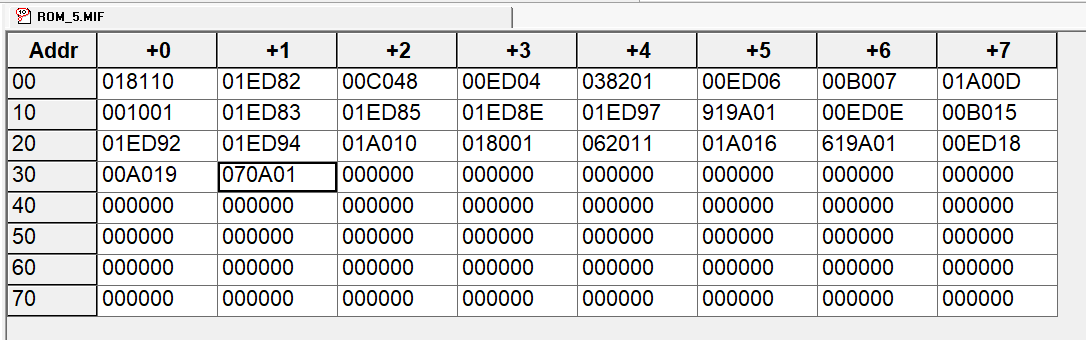
3、指令ADD [0BH]为双字节指令，指令码：200B，其功能是将R0中的数据与RAM的0BH地址单元中数据求和，存入R0中；

4、指令OUT[0BH]为双字节指令，指令码：300B，其功能是将R0中的数据与RAM的0BH地址单元中数据求差，存入R0中；

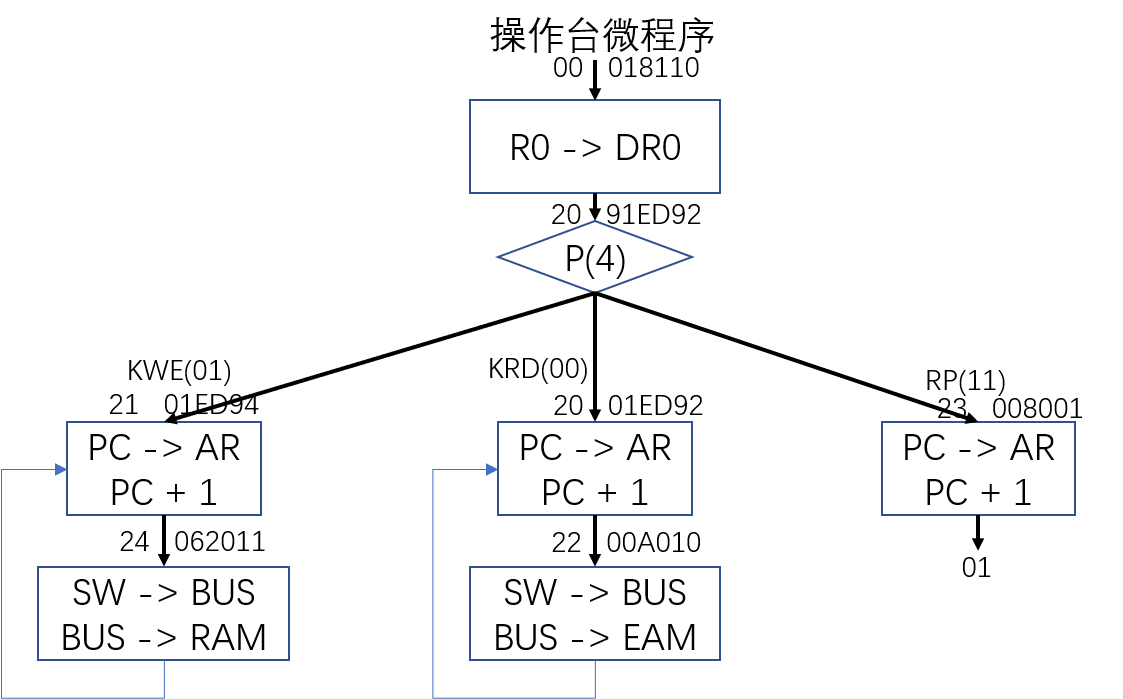
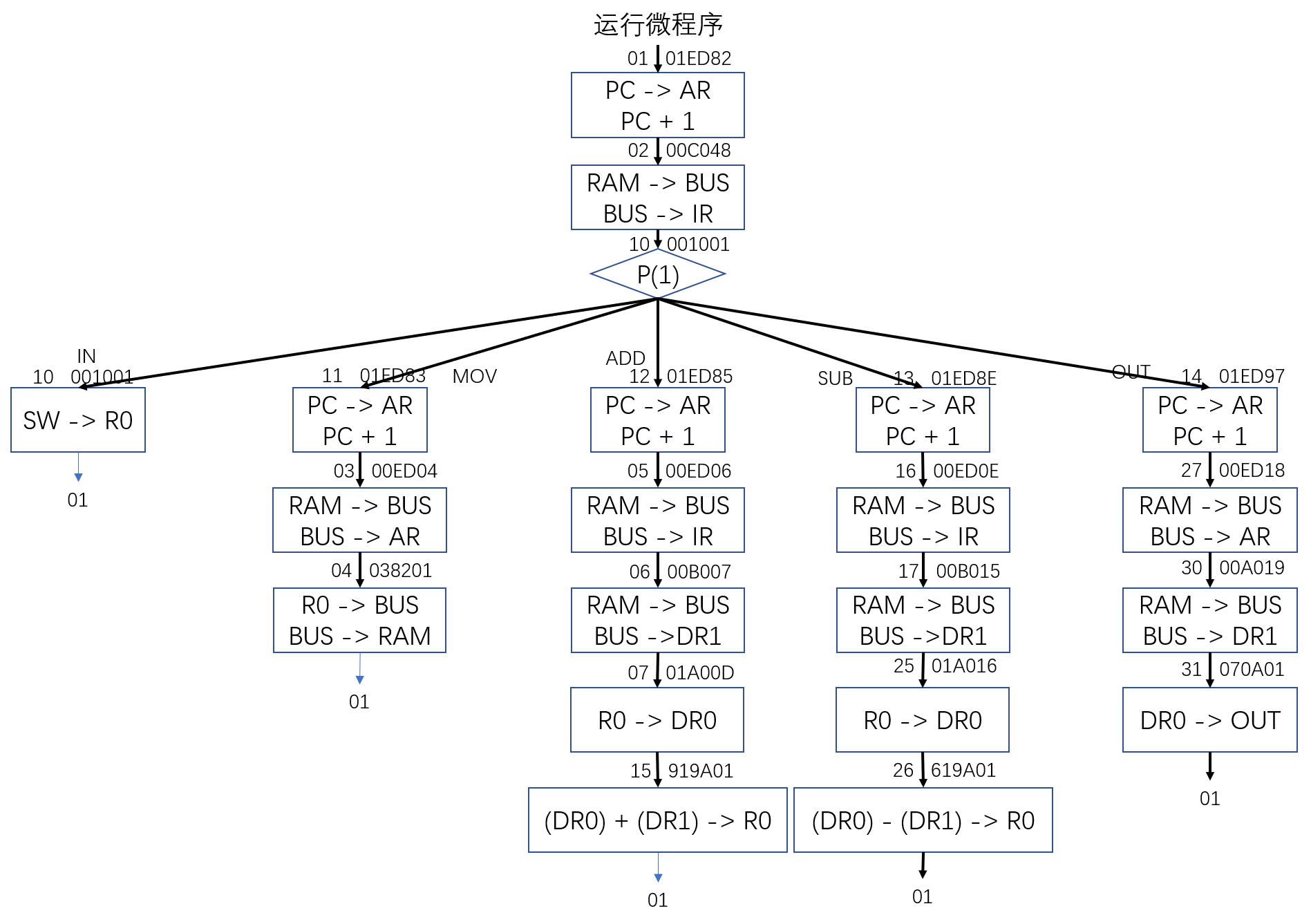
5、指令JMP [12H] 为双字节指令，指令码：4012，其功能是将RAM的12H地址单元中的数据送到OUT输出口上。

三．实验步骤

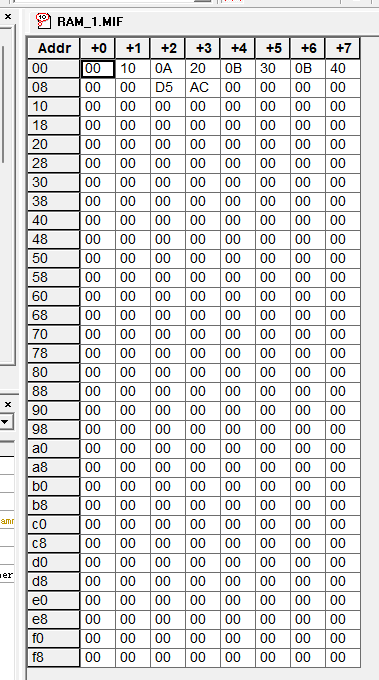
1．微程序的输入：根据表1-3所对应的二进制微代码，编辑LPM\_ROM配置文件ROM\_11.mif，并将其保存在与实验电路b100\_c.bdf工程所在的文件夹中，与实验电路b100\_c.bdf一同编译。图1-3是示例原理图。



2．输入模型机的程序



（二）配置LPM\_RAM



四．实验心得

计组课设整体复杂度高，综合性强，实验中遇到过很多困难，也加深了对计算机组成与体系结构的理解。