计算机组成原理实验报告

计算机科学与技术学院（网络空间安全学院）

**班级**： 计算机21-1 **姓名**： 梁浩铂 **成绩**：

|  |  |  |
| --- | --- | --- |
| **实验名称** | **日期** | **地点** |
| **实验六、简单模型机组成原理实验** | **2023.11.27** | **XXY-A502** |

###### 实验目的

1. 在掌握各部件功能的基础上，组成一个简单的计算机整机系统—模型机；

2. 了解微程序控制器是如何控制模型机运行的，掌握整机动态工作过程;

3. 定义五条机器指令，编写相应微程序并具体上机调试。

###### 预习要求（20分）

1. 复习指令系统和中央处理器两章的内容，熟悉图6-2中微程序流程图；

2. 预习实验内容，填写下表；

|  |  |
| --- | --- |
| 指令助记符 | 指令的第7654位值（I7I6I5I4） |
| IN AX, KIN | 0000 |
| MOV BX, 01H | 0010 |
| ADD AX, BX | 0001 |
| OUT DISP, AX | 0011 |
| JMP 00H | 0100 |

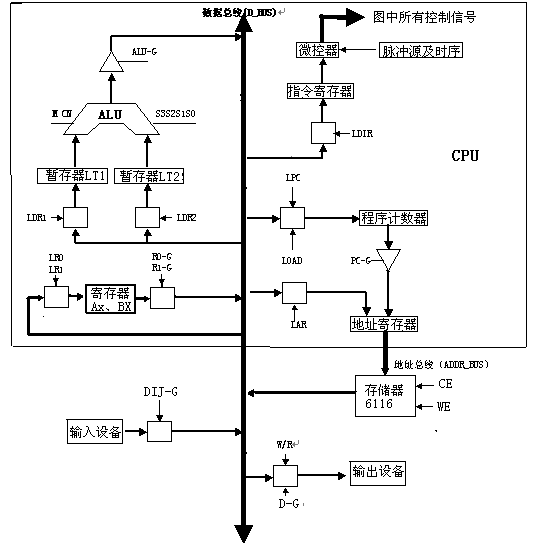
###### 实验设备：

JY系列计算机组成原理实验系统一套，排线若干。

###### 实验原理

**1）模型机结构：**

模型机结构框图见图6-1。



**图6-1 模型机结构框图**

* 运算器ALU由四片74LS181构成，暂存器1由两片74LS273构成，暂存器2由两片74LS273构成。微控器部分控存由三片28c16构成,除此之外，CPU的其它部分都由EPM570（可编程逻辑器件）集成。
* 存储器部分由两片6116构成16位存储器，地址总线只有低八位有效，因而其存储空间为00H—FFH。
* 输出设备由四个LED数码管及其译码、驱动电路构成，当D-G和W/R均为低电平时将数据总线的数据送入数码管显示。在开关方式下，输入设备由16位电平开关及两个三态缓冲芯片74LS244构成，当DIJ-G为低电平时将16位开关状态送上数据总线。在键盘方式或联机方式下，数据可由键盘或上位机输入，然后由监控程序直接送上数据总线，因而外加的数据输入电路可以不用。

**注：本系统的数据总线为16位，指令、地址和程序计数器均为8位。当数据总线上的数据打入指令寄存器、地址寄存器和程序计数器时，只有低8位有效**。

**2）工作原理：**

在微程序控制器的实现方式下，系统机器指令功能由执行一段控存中的微程序实现，一条机器指令对应一段微程序；另外，读、写机器指令也分别由相应的微程序段来完成。

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，必须设计三个控制操作微程序。

存储器读操作（MRD）：拨动清零开关CLR对地址、指令寄存器清零后，指令译码输入CA1、CA2为“00”时，按“单步”键，可对RAM连续读操作。

存储器写操作（MWE）：拨动清零开关CLR对地址、指令寄存器清零后，指令译码输入CA1、CA2为“10”时，按“单步”键，可对RAM连续写操作。

启动程序（RUN）：拨动开关CLR对地址、指令寄存器清零后，指令译码输入CA1、CA2为“11”时，按“单步”键，即可转入到第01号“取指”微指令，启动程序运行。

**注：CA1、CA2由控制总线的E4、E5给出。键盘操作方式时由监控程序直接对E4、E5赋值，无需接线。开关方式时可将E4、E5接至控制开关CA1、CA2，由开关来控制。**

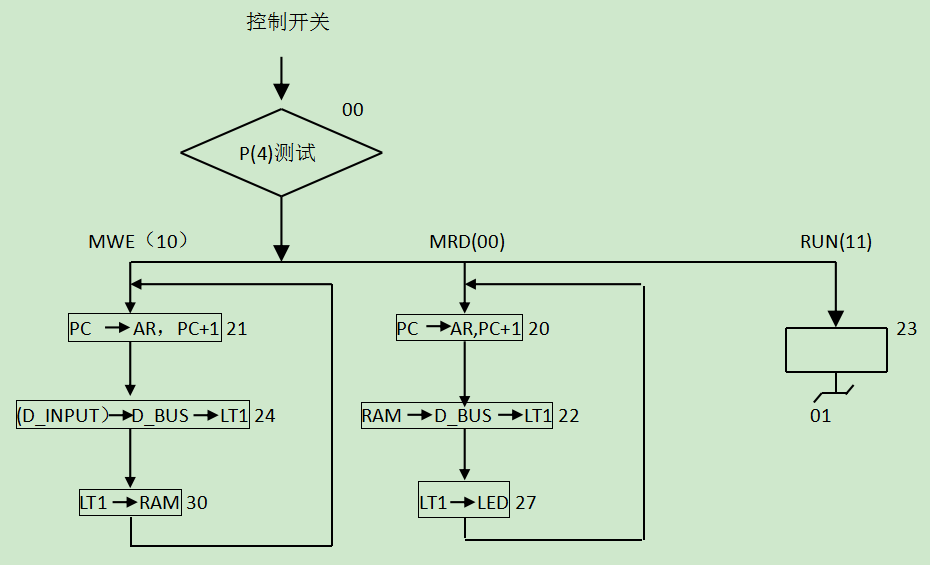
本系统设计的微指令字长24位，其控制位顺序如下：

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 14 13 | 12 11 10 | 9 8 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| S3 | S2 | S1 | S0 | M | Cn | WE | 1A | 1B | F1 | F2 | F3 | uA5 | uA4 | uA3 | uA2 | uA1 | uA0 |

Ｆ１、Ｆ２、Ｆ３三个字段的编码方案如下表，其余控制位的含义见实验五。

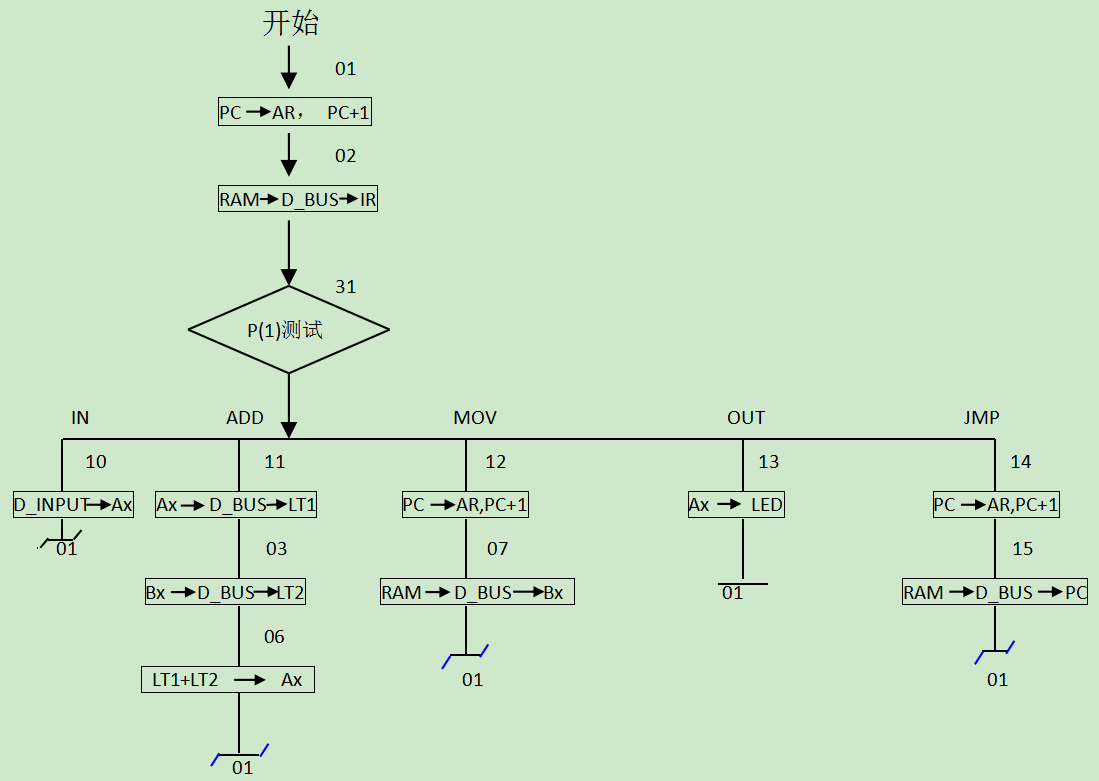
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| F1字段 | | F2字段 | | F3字段 | |
| 15 14 13 | 选择 | 12 11 10 | 选择 | 9 8 7 | 选择 |
| 0 0 0 | LDRi | 0 0 0 | RAG | 0 0 0 | P1 |
| 0 0 1 | LOAD | 0 0 1 | ALU-G | 0 0 1 | AR |
| 0 1 0 | LDR2 | 0 1 0 | RCG | 0 1 0 | P3 |
| 0 1 1 | 自定义 | 0 1 1 | 自定义 | 0 1 1 | 自定义 |
| 1 0 0 | LDR1 | 1 0 0 | RBG | 1 0 0 | P2 |
| 1 0 1 | LAR | 1 0 1 | PC-G | 1 0 1 | LPC |
| 1 1 0 | LDIR | 1 1 0 | 299-G | 1 1 0 | P 4 |
| 1 1 1 | 无操作 | 1 1 1 | 无操作 | 1 1 1 | 无操作 |

系统涉及到的微程序流程见下图**（图中各方框内为微指令所执行的操作，方框外的标号为该条微指令所处的八进制微地址）**。

控制操作为P4测试，它以CA1、CA2作为测试条件，出现了写机器指令、读机器指令和运行机器指令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存一个微地址单元随意填写。

机器指令的执行过程如下：首先将指令在外存储器的地址送上地址总线，然后将该地址上的指令传送至指令寄存器，这就是“取指”过程。之后必须对操作码进行P1测试，根据指令的译码将后续微地址中的某几位强制置位，使下一条微指令指向相应机器指令微程序的首地址，然后顺序执行该段微程序，完成该机器指令的功能。

在所有机器指令的执行过程中，“取指”和“译码”是公用微指令，对应于图6-2中01、02、31地址的微指令。31地址为“译码”微指令，该微指令的操作为P（1）测试，测试结果出现多路分支。本实验用指令寄存器的前4位（I7-I4）作为测试条件，出现5路分支，占用5个固定微地址单元，分别对应五个机器指令的微程序入口。



当全部微程序流程图设计完毕后，应将每条微指令代码化，表6-2即为将图6-2的微程序流程按微指令格式转化而成的“二进制微代码表”：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 微地址（二进制） | S3 S2 S1 S0 M CN WE 1A 1B | F1 | F2 | F3 | UA5...UA0 |
| 000000 | 0 0 0 0 0 0 0 0 0 | 111 | 111 | 110 | 010000 |
| 000001 | 0 0 0 0 0 0 0 0 0 | 101 | 101 | 101 | 000010 |
| 000010 | 0 0 0 0 0 0 0 1 0 | 110 | 111 | 111 | 011001 |
| 000011 | 0 0 0 0 0 0 0 0 0 | 010 | 100 | 111 | 000110 |
| 000110 | 1 0 0 1 0 1 0 0 0 | 000 | 001 | 111 | 000001 |
| 000111 | 0 0 0 0 0 0 0 1 0 | 000 | 111 | 111 | 000001 |
| 001000 | 0 0 0 0 0 0 0 1 1 | 000 | 111 | 000 | 000001 |
| 001001 | 0 0 0 0 0 0 0 0 0 | 100 | 000 | 111 | 000011 |
| 001010 | 0 0 0 0 0 0 0 0 0 | 101 | 101 | 101 | 000111 |
| 001011 | 0 0 0 0 0 0 1 0 1 | 111 | 000 | 111 | 000001 |
| 001100 | 0 0 0 0 0 0 0 0 0 | 101 | 101 | 101 | 001101 |
| 001101 | 0 0 0 0 0 0 0 1 0 | 001 | 111 | 101 | 000001 |
| 010000 | 0 0 0 0 0 0 0 0 0 | 101 | 101 | 101 | 010010 |
| 010001 | 0 0 0 0 0 0 0 0 0 | 101 | 101 | 101 | 010100 |
| 010010 | 0 0 0 0 0 0 0 1 0 | 100 | 111 | 111 | 010111 |
| 010011 | 0 0 0 0 0 0 0 0 0 | 111 | 111 | 111 | 000001 |
| 010100 | 0 0 0 0 0 0 0 1 1 | 100 | 111 | 111 | 011000 |
| 010111 | 0 0 0 0 0 1 1 0 1 | 111 | 001 | 111 | 010000 |
| 011000 | 1 1 1 1 1 1 1 1 0 | 111 | 001 | 111 | 010001 |
| 011001 | 0 0 0 0 0 0 0 1 0 | 110 | 111 | 000 | 001000 |

**表6-2**

本实验采用五条机器指令，根据上面所说的工作原理，设计参考实验程序如下：

地址（二进制） 机器指令（二进制） 助记符 说 明

0000 0000 0000 0000 IN AX, KIN 数据输入电路 AX

0000 0001 0010 0001 MOV Bx, 01H 0001H Bx

0000 0010 0000 0001

0000 0011 0001 0000 ADD Ax, Bx Ax+Bx Ax

0000 0100 0011 0000 OUT DISP, Ax Ax 输出显示电路

0000 0101 0100 0000 JMP 00H 00H PC

0000 0110 0000 0000

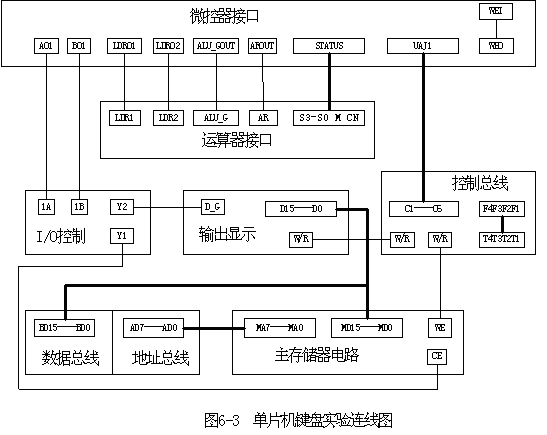
**注：其中MOV、JMP为双字长（32位），其余为单字长指令。对于双字长指令，第一字为操作码，第二字为操作数；对于单字长指令只有操作码，没有操作数。上述所有指令的操作码均为低8位有效，高八位默认为0。而操作数8位和16位均可。KIN和DISP分别为本系统专用输入、输出设备。**

###### 实验内容与步骤（40分）

**注意：在进行单片机键盘控制实验时，必须把开关K4置于“OFF”状态，否则系统处于自锁状态，无法进行实验！**

1、根据图6-2编写微程序和一段机器语言程序，并读出、运行验证其正确性。

2、实验连线如下图

99

1. 实验过程

**写微代码：**

1. 将开关K1K2K3K4拨到写状态即K1 off、K2 on、K3 off、K4 off，其中K1、K2、K3在微程序控制电路，K4在24位微代码输入及显示电路上。在监控指示灯滚动显示【CLASS SELECt】状态下按【实验选择】键，显示【ES--\_ \_ 】输入06或6，按【确认】键，显示为【ES06】，再按下【确认】键。
2. 监控显示为【CtL1=\_】，输入1显示【CtL1\_1】，按【确认】。
3. 监控显示【U-Addr】，此时输入【000000】6位二进制数表示的微地址，然后按【确认】键，监控指示灯显示【U\_CodE】，显示这时输入微代码【007F90】，该微代码是用6位十六进制数来表示前面的24位二进制数，注意输入微代码的顺序，先右后左，按【确认】键则显示【PULSE】，按【单步】完成一条微代码的输入，重新显示【U-Addr】提示输入第二条微代码地址。
4. 按照上面的方法输入表6-3微代码,观察微代码与微地址显示灯的对应关系（注意输入微代码的顺序是由右至左）。

|  |  |  |
| --- | --- | --- |
| 微地址（八进制） | 微地址（二进制） | 微代码（十六进制） |
| 00 | 000000 | 007F90 |
| 01 | 000001 | 005B42 |
| 02 | 000010 | 016FD9 |
| 03 | 000011 | 0029C6 |
| 06 | 000110 | 9403C1 |
| 07 | 000111 | 010FC1 |
| 10 | 001000 | 018E01 |
| 11 | 001001 | 0041C3 |
| 12 | 001010 | 005B47 |
| 13 | 001011 | 02F1C1 |
| 14 | 001100 | 005B4D |
| 15 | 001101 | 011F41 |
| 20 | 010000 | 005B52 |
| 21 | 010001 | 005B54 |
| 22 | 010010 | 014FD7 |
| 23 | 010011 | 007FC1 |
| 24 | 010100 | 01CFD8 |
| 25 | 010101 | 06F3C1 |
| 26 | 010110 | 011F41 |
| 27 | 010111 | 06F3D0 |
| 30 | 011000 | FF73D1 |
| 31 | 011001 | 016E08 |

**表6－3**

**读微代码及校验微代码：**

1. 先将开关K1K2K3K4拨到读状态即K1 off、K2 off、K3 on、K4 off，按【RESET】按钮对单片机复位，使监控指示灯滚动显示【CLASS SELECt】状态。
2. 按【实验选择】键，显示【ES--\_ \_ 】输入06或6，按【确认】键，显示【ES06】。按【确认】键。
3. 监控显示【CtL1=\_】时，输入2，按【确认】显示【U\_Addr】 ，此时输入6位二进制微地址，进入读代码状态。再按【确认】显示【PULSE】，此时按【单步】键，微地址指示灯显示输入的微地址，同时微代码显示电路上显示该地址对应的微代码，至此完成一条微指令的读过程。
4. 此时监控显示【U\_Addr】，按上述步骤对照表6-3表检查微代码是否有错误，如有错误，可按步骤2重新输入微代码。

**写机器指令**

1. 先将K1K2K3K4拨到运行状态即K1 on、K2 off、K3 on、K4 off，按【RESET】按钮对单片机复位，使监控指示灯滚动显示【CLASS SELECt】状态。
2. 按【实验选择】键，显示【ES--\_ \_ 】输入06或6，按【确认】键，显示【ES06】，再按【确认】。
3. 监控显示【CtL1=\_】,按【取消】键，监控指示灯显示【CtL2=\_】，输入1显示【CtL2\_1】表示进入对机器指令操作状态，此时拨动CLR清零开关（在控制开关电路上，注意对应的JUI应短接）对地址寄存器、指令寄存器清零，清零结果是微地址指示灯（uA5—uA0）和地址指示灯（A7—A0）全灭，清零步骤是使其电平高－低－高即CLR指示灯状态为亮－灭－亮。**如不清零则会影响机器指令的输入！！！**确定清零后，按【确认】。
4. 监控显示闪烁的【PULSE】，连续按【单步】键，当微地址显示灯显示“010100”，时按【确认】键，监控指示灯显示【data】，提示输入机器指令“00”或“0000”（两位或四位十六进制数）,输入后按【确认】，显示【PULSE】，再按【单步】，微地址显示灯显示“011000”，数据总线显示灯显示“0000000000000000”，即输入的机器指令。
5. 再连续按【单步】，当微地址显示灯再次显示“010100”时，按【确认】输入第二条机器指令。依此规律逐条输入表6－4的机器指令，输完后，可连续按【取消】或【RESET】退出写机器指令状态。**注意，每当微地址显示灯显示“010100”时，地址指示灯自动加1显示。如输入指令为8位，则高8位自动变为0。**

|  |  |
| --- | --- |
| 地址（十六进制） | 机器指令（十六进制） |
| 00 | 0000 |
| 01 | 0021 |
| 02 | 0001 |
| 03 | 0010 |
| 04 | 0030 |
| 05 | 0040 |
| 06 | 0000 |

**表6-4 机器指令表**

**读机器指令：**

在监控指示灯显示【CtL2=\_】状态下，输入2，显示【CtL2\_2】，表示进入读机器指令状态，按步骤4的方法拨动CLR开关对地址寄存器和指令寄存器进行清零，然后按【确认】键，显示【PULSE】，连续按【单步】键，微地址显示灯从“000000”开始，然后按“010000”、 “010010” 、“010111”方式循环显示。当微地址灯再次显示为“010000”时，输出显示数码管上显示写入的机器指令。读的过程注意微地址显示灯，地址显示灯和数据总线指示灯的对应关系。如果发现机器指令有误，则需重新输入机器指令。

注意：机器指令存放在RAM里，掉电丢失，故断电后需重新输入。

**6**

###### 实验结果与分析（40分）

**在实验过程中验证微程序和机器语言程序的正确性。**

**1、运行程序**

在监控指示灯显示【CtL2=\_】状态下，输入3，显示【CtL2\_3】，表示进入运行机器指令状态，按步骤4的方法拨动CLR开关对地址寄存器和指令寄存器进行清零,使程序入口地址为00H，可以【单步】运行程序也可以【全速】运行，运行过程中提示输入相应的量，运行结束后从输出显示电路上观察结果。

1).单步运行结果

在监控指示灯显示【run CodE】状态下，连续按【单步】键，可以单步运行程序。当微地址显示灯显示“001000”“000001”时，监控显示【dAtA】,提示输入数据，即被加数，输入 1206（此处学号后四位），按【确认】，再连续按【单步】，在微地址灯显示“001011”时，按【单步】，此时可由输出显示电路的数码管观察结果为 1207 同时数据显示灯显示“ 1207 ” 。

2).全速运行结果

在监控指示灯显示【run CodE】状态下，按【全速】键，则开始自动运行程序，在监控指示灯显示【dAtA】时输入数据，按【确定】键，程序继续运行，此时可由输出显示电路的数码管显示运算结果为“ 1207 ”。

1. **分析结果**

阅读程序可知，程序功能为：

即完成数据 1206H 和数据 0001H 做 加 操作，结果为 1207H ，实验结果显示灯为

1207H ，表示结果 正确 （正确或错误）

教师检查（在对应处打√）：

线路连接：正确 ，有错误 ；

实验结果：正确 ，有错误 ；

1. 分析（20分）

实验值和理论值比较结果为： 相同

根据实验原理，实现输入一个数据，减2以后输出的功能，如何修改相关程序： 将01H处的机器代码改为0022H将03H处的机器代码改为0060H。因为：地址：0000 0001 机器指令：0010 0001 做的操作为：MOV Bx, 01H ，只需要将机器指令改为：0010 0010 做的操作为：MOV Bx, 02H，即令减数是02H，地址：0000 0011 机器指令：0001 0000 做的操作为： ADD Ax, Bx ，只需要将机器指令改为：0110 0000 做的操作为：SUB Ax, Bx，即让输入的数减02H

