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** |
| **实验四、微程序设计实验** | **2023.11.20** | **XXY-A502** |

###### 实验目的

1. 掌握微程序控制器的组成原理和工作过程。
2. 理解微指令和微程序的概念，理解微指令与指令的区别与联系。

###### 预习要求（20分）

1. 复习微程序控制器工作原理；

2. 预习实验内容，填写表5-1中的内容；

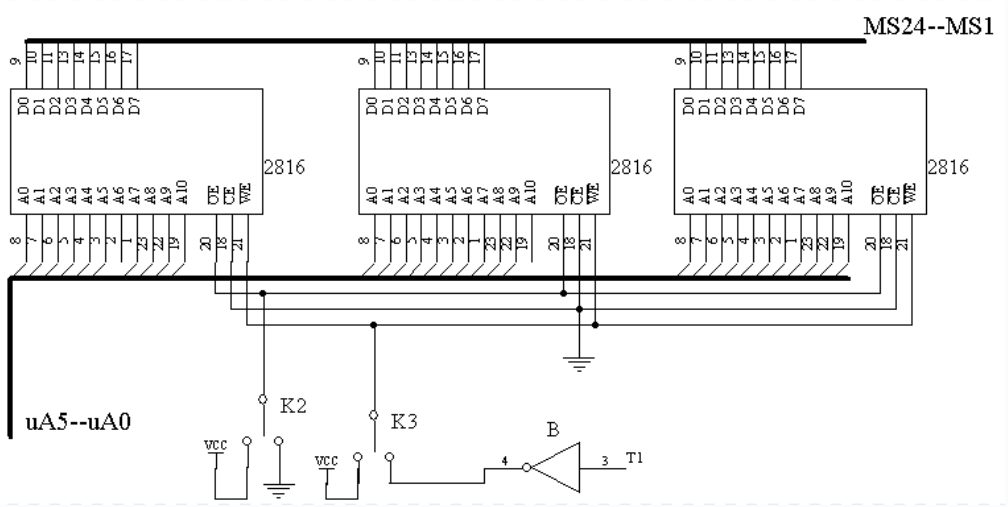
###### 实验设备：

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、电路图**

3片EPROM2716构成控制存储器电路如下：



**2.实验原理**

每条微指令由24位组成，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 |

S3S2S1S0MCn为运算器的方式控制；WE为外部器件的读写信号，‘1’表示写，‘0’表示读；1A、1B用于选通外部器件，通常接至底板IO控制电路的1A1B端，四个输出Y0Y1Y2Y3接外部器件的片选端。

Ｆ１、Ｆ２、Ｆ３三个字段的编码方案如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 | LRi | 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 |

LRi：寄存器堆的写控制，它与指令寄存器的第0位和第1位共同决定对哪个寄存器进行写操作；

LOAD：程序计数器的置数控制；

LDR1、LDR2：运算器的两个锁存控制，总线数据送暂存器（见实验一）；

LAR：地址寄存器的锁存控制，总线上的地址送地址寄存器；

LDIR：指令寄存器的锁存控制，指令送指令寄存器；

RAG（R0G）、RBG（R1G）、RCG（R2G）：寄存器AX(R0)、BX(R1)、CX(R2)的输出控制，输出到总线；

ALU-G：运算器的输出控制，运算结果送总线；

PC-G：程序计数器的输出控制，PC值送地址寄存器；

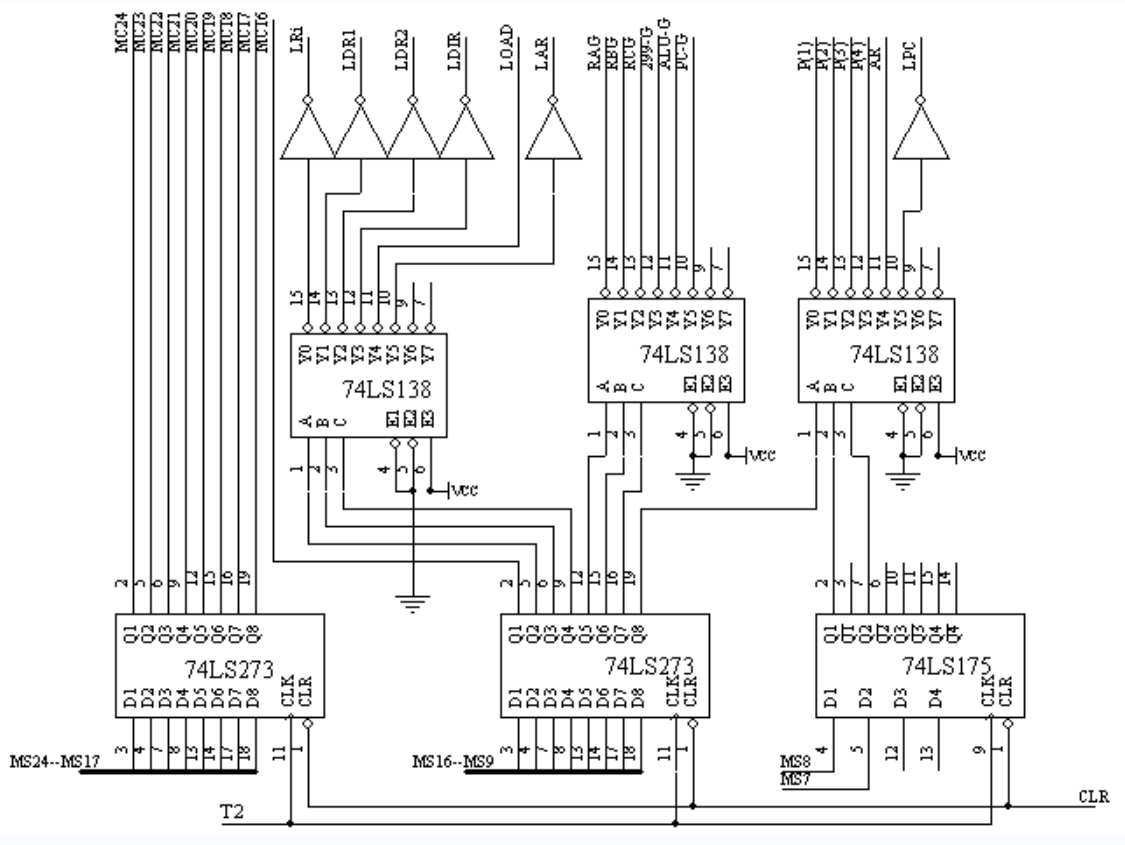
1. G：移位寄存器的输出控制；

P1、P2、P3、P4：位测试字，其功能是对机器指令进行译码，使微程序转入相应的微地址入口，从而实现微程序的顺序、分支和循环运行；

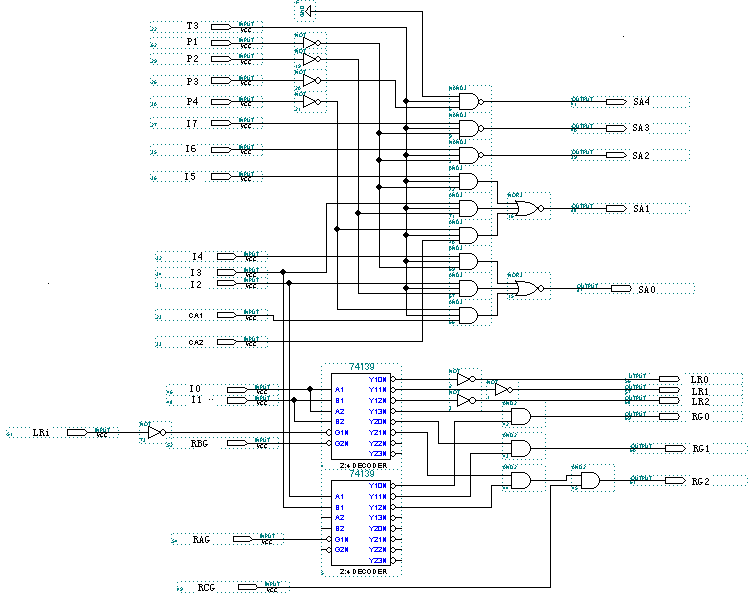
AR：运算器的进位输出控制；

LPC：程序计数器的时钟控制

微指令译码电路如下：

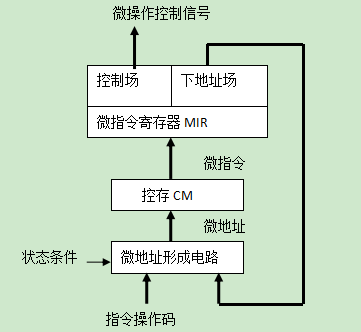


指令译码器电路如下：



微指令中的uA5-uA0为6位的后续微地址，本系统采用分段编码的指令格式，采用断定方式确定下一条微指令的地址， 微指令中设有“下地址场”，它指出下条微指令的地址，这使一条指令的微程序中的微指令在控存中不一定要连续存放。在微程序执行过程中，微程序控制部件中的微地址形成电路直接接受微指令下地址场信息来产生下条微指令地址，微程序的首址也由此微地址形成电路根据指令操作码产生。

如下图所示：



例如：编写几条微代码如下表，按照微指令格式及说明，分析微指令功能如下表5-1所示：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 微地址  （二进制） | 微代码  （十六进制） | ALU | F1 | F2 | F3 | 下地址 | 功能说明 |
| 000000 | 007F90 | 无有效操作 | 111  无操作 | 111  无操作 | 110  无操作 | 01 | P4测试，根据CA1CA2状态进行RAM读、写，转入01号微指令 |
| 000001 | 005B42 | 无有效操作 | 101  LAR | 101  PC-G | 101  LPC | 02 | PC→AR， PC+1，读指令地址 |
| 000010 | 016FC9 | 无有效操作 | 110  LDIR | 111 | 111 | 03 | RAM做读操作，取指令 RAM→D\_BUS→IR， |
| 000011 | 0029C6 | 无有效  操作 | 010  LDR2 | 100  RBG | 111 | 000011 | BX(R0)->DBUS->运算器第二个暂存器锁存控制 |
| 000110 | 9403C1 | 有有效  操作 | 000  LRi | 001  ALU-G | 111 | 000001 | ALU-G->DBUS，对寄存器进行写操作 |
| 001001 | 0041C3 | 无有效  操作 | 100  LDR1 | 000  RAG | 111 | 000011 | AX(R0)->DBUS->运算器的第一个暂存器锁存 |

###### 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、往EEPROM里写入一段微程序，并读出验证后，运行此程序观察微指令执行流程。

2、实验连线

实验连线图如图所示。连线时应按如下方法：对于横排座，应使排线插头上的箭头面向自己插在横排座上；对于竖排座，应使排线插头上的箭头面向左边插在竖排座上。

微程序接口

UAJ1

C1C2C3C4C5C6 F1F2F3F4

控制总线

T1T2T3T4

图 实验五键盘实验接线图

3、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05或5，按【确认】键，显示为【ES05】，再按下【确认】键。
2. 监控显示为【CtL1=\_】，表示对微代码进行操作。输入1显示【CtL1\_1】，表示写微代码，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】，这时输入微代码【000001】，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，此过程中可按【取消】键来取消上一次输入，重新输入。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】提示输入表5-3第二条微代码地址。
4. 按照上面的方法输入下表中的微代码,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（二进制） | 微代码（十六进制） |
| 000000 | 000001 |
| 000001 | 000002 |
| 000010 | 000003 |
| 000011 | 015FC4 |
| 000100 | 012FC8 |
| 001000 | 018E09 |
| 001001 | 005B50 |
| 010000 | 005B55 |
| 010101 | 06F3D8 |
| 011000 | FF73D9 |
| 011001 | 017E00 |

**读微代码：**

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05或5，按【确认】键，显示【ES05】。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，进入读微代码状态。再按【确认】显示【PULSE】，此时按【单步】键，监控显示【U\_Addr】，微地址指示灯显示输入的微地址，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。对照表5-3表检查微代码是否有错误，如有错误，可按步骤2写微代码重新输入这条微代码。

**微代码的运行**：

1. 先将开关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05或5，按【确认】键，显示【ES05】。按【确认】键。
3. 监控指示灯显示【CtL1=\_】，输入3，显示【CtL1\_3】，表示进入运行微代码状态，拨动CLR清零开关（在控制开关电路上，注意对应的JUI应短接）对程序计数器清零，清零结果是地址指示灯（A7—A0）和微地址显示灯（uA5—uA0）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，使程序入口地址为000000。

1）、单步运行

在监控指示灯显示【CtL1\_3】状态下，确认清零后，按【确认】键，监控指示灯滚动显示【Run CodE】,此时可按【单步】键单步运行微代码，观察微地址显示灯，显示 “000001”，再按【单步】，显示为“000010”，连续按【单步】，则可单步运行微代码，注意观察微地址显示灯和微代码的对应关系。

2）、全速运行

在控指示灯滚动显示【Run CodE】状态下，按【全速】键，开始自动运行微代码，微地址显示灯显示从“000000”开始，到“000001”、“000010”、“000011”、“000100”、“001000”、“001001”、“010000”、“010101”、“011000”、“011001”再到“000000”，循环显示。

###### 实验结果与分析（40分）

**验证微程序中微指令的正确性。**

分析相应微代码填入下表中。

1.分析结果（20分）

|  |  |  |
| --- | --- | --- |
| 微地址（二进制） | 微代码(十六进制/二进制) | 下地址（二进制） |
| 000011 | 015FC4/000000010101111111000100 | 000100 |
| 000100 | 012FC8/000000010010111111001000 | 001000 |
| 001000 | 018E09/000000011000111000001001 | 001001 |
| 001001 | 005B50/000000000101101101010000 | 010000 |
| 010000 | 005B55/000000000101101101010101 | 010101 |
| 010101 | 06F3D8/000001101111001111011000 | 011000 |
| 011000 | FF73D9/111111110111001111011001 | 011001 |
| 011001 | 017E00/000000010111111000000000 | 000000 |

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： 理论上，微代码可以控制下一条微代码的地址；在实验中，得出微代码的二进制表示的后六位决定了下一条微代码的微地址

试编写将AX中内容送BX，下一条跳转到20地址的微指令代码： 0001D0

由题可知可选用无效操作ALU即前六位为0，将AX内容送给BX即F1为000 F2为000 F3为111

跳转到20地址即010000，所以综合为000000000000000111010000 即0001D0

