**微程序控制控制器实验报告**

**计科1702 徐晟 17221023**

**计科1702 牛兴炜 15281129**

**一、详细针对实验指导书P59页 ADD R0, R1指令的微程序，说明每一条微指令的功能，这些功能是由哪些字段具体实现的。**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **微址** | **下址** | **CI3-0** | **SCC** | **MRW** | **I2-0** | **I8-6** | **I5-3** | **B口** | **A口** | **SST** | **SSHSC** | **DC2** | **DC1** |
| **00** | **00** | **1110** | **0000** | **100** | **001** | **011** | **001** | **0101** | **0101** | **000** | **001** | **111** | **000** |
| **01** | **00** | **1110** | **0000** | **100** | **011** | **010** | **000** | **0101** | **0101** | **000** | **001** | **011** | **000** |
| **02** | **00** | **1110** | **0000** | **001** | **000** | **001** | **000** | **0000** | **0000** | **000** | **000** | **001** | **000** |
| **03** | **04** | **0010** | **0000** | **100** | **001** | **011** | **000** | **0000** | **0001** | **001** | **000** | **000** | **000** |
| **04** | **30** | **0011** | **0000** | **100** | **001** | **011** | **000** | **0000** | **0001** | **001** | **000** | **000** | **011** |
| **30** | **3A** | **0011** | **0010** | **100** | **111** | **000** | **000** | **0000** | **0000** | **000** | **000** | **000** | **011** |
| **31** | **02** | **0011** | **0000** | **100** | **011** | **010** | **000** | **0101** | **0101** | **000** | **001** | **011** | **000** |

1. 本条微指令完成公共操作0→PC，DI#=0；

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **00** | **00** | **1110** | **0000** | **100** | **001** | **011** | **001** | **0101** | **0101** | **000** | **001** | **111** | **000** |

由于I8-6为011，即通用寄存器结果F→B，I2-I0为001，即RS的数据来源为A口和B口，A口B口均为0101，化为10进制为5，即寄存器为R5，I5-I3为001，实现运算功能S-R，A口B口的数据都是寄存器R5的数据，尽管R5还没有赋值，但是无论它的值是什么都要实现S-R赋给B口寄存器的功能，S与R都为R5，所以实现将0赋给R5。

其它指令如MRW为100，即无读写，SSH为001，即Cin=1，保证了运算器数据减法的正确性，DC2为111，即开关中断，不使用开关数据，DC1为000，开关内容到内部总线，由于DC2的缘故，开关中断，更不要说将开关内容传至总线了。

下址为00，CI3-0为1110，实现地址的递增+1，此时寻找微址为01的微指令。

2.本条微指令完成PC→AR，PC+1→PC；

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **01** | **00** | **1110** | **0000** | **100** | **011** | **010** | **000** | **0101** | **0101** | **000** | **001** | **011** | **000** |

DC2为011，即AR高位接收，通用寄存器R5便将地址存入了高位的地址寄存器。I2-0为011，数据来源为0和B口，B口即R5的数据，I5-3为000，即实现R+S的功能，I8-6为010，即将运算结果放到B口，并输出A，在这里运算结果为R+S，即0+R5，为R5，但注意到SSH为001，即Cin为1，实现了寄存器数据自增1操作，所以结果为R5+1并存到了A口，输出仍为B口数据R5。

下址为00，CI3-0为1110，实现地址的递增+1，此时寻找微址为02的微指令。

3.本条微指令完成MEM→IR；

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **02** | **00** | **1110** | **0000** | **001** | **000** | **001** | **000** | **0000** | **0000** | **000** | **000** | **001** | **000** |

MRW为001，即实现内存读的操作，储存器6116通过缓冲器245将数据放在内部总线上，DC2为001，即指令寄存器接收，指令寄存器IR从内部总线上获取数据，即完成MEM→IR的操作。

I2-0为001，数据来源R为A，S为B，I5-3为001，即实现运算功能S-R，I8-6为000，即实现F→Q，输出F的操作，Q中存着数据0，输出也为0。

下址为00，CI3-0为1110，即顺序执行，实现地址的递增+1，此时寻找微址为03的微指令。

4. 本条微指令完成/MAP操作功能；

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **03** | **04** | **0010** | **0000** | **100** | **001** | **011** | **000** | **0000** | **0001** | **001** | **000** | **000** | **000** |

MRW为100，即无读写，DC2为000，即NC，没有任何功能，DC1为000，开关没有打开，不存在将开关内容送到开关总线，相当于没有功能。

I2-0为001，数据来源R为A，S为B，I5-3为001，即实现运算功能S-R，I8-6为000，即实现F→Q，输出F的操作，Q中存着数据0，输出也为0。

下址为04，CI3-0为0010，即下地址传入2910，2910将地址传入MAPROM，实现了/MAP的操作功能，因为当/MAP有效时，D来源于MAPROM，用于实现从机器指令的操作码找到相应的微程序段首地址。

5.本条微指令执行ADD指令，DR DR+SR的操作；

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **04** | **30** | **0011** | **0000** | **100** | **001** | **011** | **000** | **0000** | **0001** | **001** | **000** | **000** | **000** |

用于完成两个累加器之间的相加操作。

I2-0为001，数据来源R为A口，A口为0000，即R0，S为B口，B口为0001即R1，I8-6为011，将运算结果存至B口，即存至R1，I5-3为000，实现功能R+S，综合以上即实现ADD指令R1 R1+R0，DC1和DC2的解释和第4条相同。SST为001，改变了CZVS的值，与ADD指令关系不大。

下址为30，CI3-0为0011，条件转移，下微指令转判中断请求，看SCC，SCC为0000，意思是必转，直接转到地址30即可，此时跳转至微址为30的微指令。

6.本条微指令完成STR→Q，CC#=INT#公共操作功能。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **30** | **3A** | **0011** | **0010** | **100** | **111** | **000** | **000** | **0000** | **0000** | **000** | **000** | **000** | **011** |

完成两个累加器之间的数据传送，DC1为011，把CZVS状态给到内部总线，再通过后续微操作将状态存入寄存器，之后如果要用的话，再取出。

I2-0为100，数据来源R为D，S为0，I5-3为000，实现功能R+S，I8-6为000，将运算结果存到Q寄存器，并且输出也为计算结果，不保存标志位的值，故SST为000，无存储器和 IO读写。

下址为3A，CI3-0为0011，条件转移，看SCC，SCC为0010，意思是/INC=0，有中断请求时转，给出/PL信号, 看/CC的值，如果/CC的值为低，则转移执行至下地址值；如果是高，顺序执行即可。

7.本条微指令完成PC→AR，PC+1→PC，CC#=0的公共操作。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **31** | **02** | **0011** | **0000** | **100** | **011** | **010** | **000** | **0101** | **0101** | **000** | **001** | **011** | **000** |

DC2为011，即AR高位接收，通用寄存器R5便将地址存入了高位的地址寄存器，实现了R5→AR。

I2-0为011，数据来源R为0，S为B，I5-3为000，实现运算功能R+S，I8-6为010，将运算结果存入B口，但是输出A口的数据，注意到SSH为001，即Cin为1，实现了R5的自增+1,。

下址为02，CI3-0为0011，条件转移，看SCC，SCC为0010，给出/MAP信号, 使D从映射部件得到。

**二、选做P60页 MVRD 的微程序，要求同一。**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **微址** | **下址** | **CI3-0** | **SCC** | **MRW** | **I2-0** | **I8-6** | **I5-3** | **B口** | **A口** | **SST** | **SSHSC** | **DC2** | **DC1** |
| **00** | **00** | **1110** | **0000** | **100** | **001** | **011** | **001** | **0101** | **0101** | **000** | **001** | **111** | **000** |
| **01** | **00** | **1110** | **0000** | **100** | **011** | **010** | **000** | **0101** | **0101** | **000** | **001** | **011** | **110** |
| **02** | **00** | **1110** | **0000** | **001** | **000** | **011** | **000** | **0000** | **0000** | **000** | **000** | **001** | **000** |
| **03** | **1D** | **0010** | **0000** | **100** | **000** | **001** | **000** | **0000** | **0000** | **000** | **000** | **001** | **000** |
| **1D** | **1C** | **0011** | **0000** | **100** | **011** | **010** | **000** | **0101** | **0101** | **000** | **001** | **011** | **000** |
| **1C** | **30** | **0011** | **0000** | **001** | **111** | **011** | **000** | **0000** | **0000** | **000** | **000** | **000** | **000** |
| **30** | **3A** | **0011** | **0010** | **100** | **111** | **000** | **000** | **0000** | **0000** | **000** | **000** | **000** | **011** |
| **31** | **02** | **0011** | **0000** | **100** | **011** | **010** | **000** | **0101** | **0101** | **000** | **001** | **011** | **000** |

1.本条微指令完成公共操作0→PC，DI#=0；

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **00** | **00** | **1110** | **0000** | **100** | **001** | **011** | **001** | **0101** | **0101** | **000** | **001** | **111** | **000** |

由于I8-6为011，即通用寄存器结果F→B，I2-I0为001，即RS的数据来源为A口和B口，A口B口均为0101，化为10进制为5，即寄存器为R5，I5-I3为001，实现运算功能S-R，A口B口的数据都是寄存器R5的数据，尽管R5还没有赋值，但是无论它的值是什么都要实现S-R赋给B口寄存器的功能，S与R都为R5，所以实现将0赋给R5。

其它指令如MRW为100，即无读写，SSH为001，即Cin=1，保证了运算器数据减法的正确性，DC2为111，即开关中断，不使用开关数据，DC1为000，开关内容到内部总线，由于DC2的缘故，开关中断，更不要说将开关内容传至总线了。

下址为00，CI3-0为1110，实现地址的递增+1，此时寻找微址为01的微指令。

2. 本条微指令完成PC→AR，PC+1→PC；

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **01** | **00** | **1110** | **0000** | **100** | **011** | **010** | **000** | **0101** | **0101** | **000** | **001** | **011** | **110** |

DC2为011，即AR高位接收，通用寄存器R5便将地址存入了高位的地址寄存器。I2-0为011，数据来源为0和B口，B口即R5的数据，I5-3为000，即实现R+S的功能，I8-6为010，即将运算结果放到B口，并输出A，在这里运算结果为R+S，即0+R5，为R5，但注意到SSH为001，即Cin为1，实现了寄存器数据自增1操作，所以结果为R5+1并存到了A口，输出仍为B口数据R5。

下址为00，CI3-0为1110，实现地址的递增+1，此时寻找微址为02的微指令。

3. 本条微指令完成MEM→IR；

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **02** | **00** | **1110** | **0000** | **001** | **000** | **011** | **000** | **0000** | **0000** | **000** | **000** | **001** | **000** |

MRW为001，即实现内存读的操作，储存器6116通过缓冲器245将数据放在内部总线上，DC2为001，即指令寄存器接收，指令寄存器IR从内部总线上获取数据，即完成MEM→IR的操作。

I2-0为001，数据来源R为A，S为B，I5-3为001，即实现运算功能S-R，I8-6为000，即实现F→Q，输出F的操作，Q中存着数据0，输出也为0。

下址为00，CI3-0为1110，即顺序执行，实现地址的递增+1，此时寻找微址为03的微指令。

4.本条微指令完成/MAP操作功能；

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **03** | **00** | **0010** | **0000** | **100** | **000** | **001** | **000** | **0000** | **0000** | **000** | **000** | **001** | **000** |

MRW为100，无读写， DC1为000，开关没有打开，不存在将开关内容送到开关总线，相当于没有功能。

DC2为001，即指令寄存器接收，指令寄存器IR从内部总线上获取数据，即完成MEM→IR的操作。

I2-0为100，数据来源R为A，S为B，I5-3为001，即实现运算功能S-R，I8-6为000，即实现F→Q，输出F的操作，Q中存着数据0，输出也为0。

下址为1D，CI3-0为0010，即下地址传入2910，2910将地址传入MAPROM，实现了/MAP的操作功能，因为当/MAP有效时，D来源于MAPROM，用于实现从机器指令的操作码找到相应的微程序段首地址。

5.本条微指令完成PC→AR，PC+1→PC，CC#=0操作；

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1D** | **1C** | **0011** | **0000** | **100** | **011** | **010** | **000** | **0101** | **0101** | **000** | **001** | **011** | **000** |

DC2为011，即AR高位接收，通用寄存器R5便将地址存入了高位的地址寄存器。

I2-0为011，数据来源为0和B口，B口即R5的数据，I5-3为000，即实现R+S的功能，I8-6为010，即将运算结果放到B口，并输出A，在这里运算结果为R+S，即0+R5，为R5，但注意到SSH为001，即Cin为1，实现了寄存器数据自增1操作，所以结果为R5+1并存到了A口，输出仍为B口数据R5。

下址为1C，CI3-0为0011，条件转移，下微指令转判中断请求，看SCC，SCC为0000，意思是必转，直接转到地址1C即可，此时跳转至微址为1C的微指令。

6.本条微指令完成MEM→DR，CC#=0操作；

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1C** | **30** | **0011** | **0000** | **001** | **111** | **011** | **000** | **0000** | **0000** | **000** | **000** | **000** | **000** |

DC2为000，即NC，没有任何功能，DC1为000，将开关内容送到内部总线，MRW为001，实现从内存中读取数据。

I2-0为111，数据来源R为D，S为0，I8-6为011，将运算结果给通用寄存器，并且将运算结果输出，I5-3为000，实现运算功能R+S，即D，由此实现了从内存中读取数据并存到累加器DR的操作MEM→DR。

下址为30，CI3-0为0011，条件转移，下微指令转判中断请求，看SCC，SCC为0000，意思是必转，直接转到地址30即可，此时跳转至微址为30的微指令。

7.本条微指令完成STR→Q，CC#INT#操作；

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **30** | **3A** | **0011** | **0010** | **100** | **111** | **000** | **000** | **0000** | **0000** | **000** | **000** | **000** | **011** |

完成两个累加器之间的数据传送，DC1为011，把CZVS状态给到内部总线，再通过后续微操作将状态存入寄存器，之后如果要用的话，再取出。

I2-0为100，数据来源R为D，S为0，I5-3为000，实现功能R+S，I8-6为000，将运算结果存到Q寄存器，并且输出也为计算结果，不保存标志位的值，故SST为000，无存储器和 IO读写。

下址为3A，CI3-0为0011，条件转移，看SCC，SCC为0010，给出/MAP信号, 使D从映射部件得到。

8.本条微指令完成PC→AR，PC+1→PC，CC#=0操作；

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **31** | **02** | **0011** | **0000** | **100** | **011** | **010** | **000** | **0101** | **0101** | **000** | **001** | **011** | **000** |

DC2为011，即AR高位接收，通用寄存器R5便将地址存入了高位的地址寄存器，实现了R5→AR。

I2-0为011，数据来源R为0，S为B，I5-3为000，实现运算功能R+S，I8-6为010，将运算结果存入B口，但是输出A口的数据，注意到SSH为001，即Cin为1，实现了R5的自增+1,。

下址为02，CI3-0为0011，条件转移，下微指令转判中断请求，看SCC，SCC为0000，意思是必转，直接转到地址02即可，此时跳转至微址为02的微指令。

**三、利用模拟程序的单微指令执行功能，说明监控程序的前两条机器指令编码是什么？这两条机器指令是什么指令。（即从PC=0及PC=1单元读出的指令），详细说明与这两条机器指令对应的微程序。说明每一条微指令的功能，这些功能是由哪些字段具体实现的。**

答：第一条机器指令为MVRD指令，编码如下为1000 1000

具体微程序如下：

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **微址** | **下址** | **CI3-0** | **SCC** | **MRW** | **I2-0** | **I8-6** | **I5-3** | **B口** | **A口** | **SST** | **SSHSC** | **DC2** | **DC1** |
| **00** | **00** | **1110** | **0000** | **100** | **001** | **011** | **001** | **0101** | **0101** | **000** | **001** | **111** | **000** |
| **01** | **00** | **1110** | **0000** | **100** | **011** | **010** | **000** | **0101** | **0101** | **000** | **001** | **011** | **110** |
| **02** | **00** | **1110** | **0000** | **001** | **000** | **011** | **000** | **0000** | **0000** | **000** | **000** | **001** | **000** |
| **03** | **1D** | **0010** | **0000** | **100** | **000** | **001** | **000** | **0000** | **0000** | **000** | **000** | **001** | **000** |
| **1D** | **1C** | **0011** | **0000** | **100** | **011** | **010** | **000** | **0101** | **0101** | **000** | **001** | **011** | **000** |
| **1C** | **30** | **0011** | **0000** | **001** | **111** | **011** | **000** | **0000** | **0000** | **000** | **000** | **000** | **000** |
| **30** | **3A** | **0011** | **0010** | **100** | **111** | **000** | **000** | **0000** | **0000** | **000** | **000** | **000** | **011** |
| **31** | **02** | **0011** | **0000** | **100** | **011** | **010** | **000** | **0101** | **0101** | **000** | **001** | **011** | **000** |

具体解释见第二题。

第二条指令为OUT PORT 指令，编码为1000 0110

具体微程序如下：

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **微址** | **下址** | **CI3-0** | **SCC** | **MRW** | **I2-0** | **I8-6** | **I5-3** | **B口** | **A口** | **SST** | **SSHSC** | **DC2** | **DC1** |
| **00** | **00** | **1110** | **0000** | **100** | **001** | **011** | **001** | **0101** | **0101** | **000** | **001** | **111** | **000** |
| **01** | **00** | **1110** | **0000** | **100** | **011** | **010** | **000** | **0101** | **0101** | **000** | **001** | **011** | **110** |
| **02** | **00** | **1110** | **0000** | **001** | **000** | **011** | **000** | **0000** | **0000** | **000** | **000** | **001** | **000** |
| **03** | **00** | **0010** | **0000** | **0001** | **000** | **001** | **000** | **0000** | **0000** | **000** | **000** | **000** | **000** |
| **12** | **14** | **0011** | **0110** | **100** | **111** | **001** | **000** | **0000** | **0000** | **000** | **000** | **011** | **010** |
| **13** | **30** | **0011** | **0000** | **010** | **011** | **001** | **000** | **0000** | **0000** | **000** | **000** | **000** | **001** |
| **30** | **3A** | **0011** | **0010** | **100** | **111** | **000** | **000** | **0000** | **0000** | **000** | **000** | **000** | **011** |
| **31** | **02** | **0011** | **0000** | **100** | **011** | **010** | **000** | **0101** | **0101** | **000** | **001** | **011** | **000** |

1.本条微指令完成公共操作0→PC，DI#=0；

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **00** | **00** | **1110** | **0000** | **100** | **001** | **011** | **001** | **0101** | **0101** | **000** | **001** | **111** | **000** |

由于I8-6为011，即通用寄存器结果F→B，I2-I0为001，即RS的数据来源为A口和B口，A口B口均为0101，化为10进制为5，即寄存器为R5，I5-I3为001，实现运算功能S-R，A口B口的数据都是寄存器R5的数据，尽管R5还没有赋值，但是无论它的值是什么都要实现S-R赋给B口寄存器的功能，S与R都为R5，所以实现将0赋给R5。

其它指令如MRW为100，即无读写，SSH为001，即Cin=1，保证了运算器数据减法的正确性，DC2为111，即开关中断，不使用开关数据，DC1为000，开关内容到内部总线，由于DC2的缘故，开关中断，更不要说将开关内容传至总线了。

下址为00，CI3-0为1110，实现地址的递增+1，此时寻找微址为01的微指令。

2. 本条微指令完成PC→AR，PC+1→PC；

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **01** | **00** | **1110** | **0000** | **100** | **011** | **010** | **000** | **0101** | **0101** | **000** | **001** | **011** | **110** |

DC2为011，即AR高位接收，通用寄存器R5便将地址存入了高位的地址寄存器。I2-0为011，数据来源为0和B口，B口即R5的数据，I5-3为000，即实现R+S的功能，I8-6为010，即将运算结果放到B口，并输出A，在这里运算结果为R+S，即0+R5，为R5，但注意到SSH为001，即Cin为1，实现了寄存器数据自增1操作，所以结果为R5+1并存到了A口，输出仍为B口数据R5。

下址为00，CI3-0为1110，实现地址的递增+1，此时寻找微址为02的微指令。

3. 本条微指令完成MEM→IR；

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **02** | **00** | **1110** | **0000** | **001** | **000** | **011** | **000** | **0000** | **0000** | **000** | **000** | **001** | **000** |

MRW为001，即实现内存读的操作，储存器6116通过缓冲器245将数据放在内部总线上，DC2为001，即指令寄存器接收，指令寄存器IR从内部总线上获取数据，即完成MEM→IR的操作。

I2-0为001，数据来源R为A，S为B，I5-3为001，即实现运算功能S-R，I8-6为000，即实现F→Q，输出F的操作，Q中存着数据0，输出也为0。

下址为00，CI3-0为1110，即顺序执行，实现地址的递增+1，此时寻找微址为03的微指令。

4.本条微指令完成/MAP操作功能；

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **03** | **00** | **0010** | **0000** | **100** | **000** | **001** | **000** | **0000** | **0000** | **000** | **000** | **001** | **000** |

MRW为100，无读写， DC1为000，开关没有打开，不存在将开关内容送到开关总线，相当于没有功能。

DC2为001，即指令寄存器接收，指令寄存器IR从内部总线上获取数据，即完成MEM→IR的操作。

I2-0为100，数据来源R为A，S为B，I5-3为001，即实现运算功能S-R，I8-6为000，即实现F→Q，输出F的操作，Q中存着数据0，输出也为0。

下址为1D，CI3-0为0010，即下地址传入2910，2910将地址传入MAPROM，实现了/MAP的操作功能，因为当/MAP有效时，D来源于MAPROM，用于实现从机器指令的操作码找到相应的微程序段首地址。

【注意】本次实验前两条指令为入口指令，后两条公共指令，每条指令的第五条为为接口指令。

5、本条指令完成PORT->AR操作功能

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **12** | **14** | **0011** | **0110** | **100** | **111** | **001** | **000** | **0000** | **0000** | **000** | **000** | **011** | **010** |

CI3-0 为0011为条件微转移，给出/PL信号, 看/CC的值，下地址值为14，顺序执行；SCC为0110，IR10=0时（IN指令）则顺序执行，两者功能相似。 MRW为100，无读写， DC1为010，送IR低位字节内容到内部总线 .

I2-0 为111数据来源为0和D, S为0 . I5-I3 为I000 实现加法。8-I6为001将Y也就是D输出。DC2 为011地址寄存器接收PORT.

综上，实现了PORT->AR得操作。

6、本条指令完成RO->IO操作功能

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **13** | **30** | **0011** | **0000** | **010** | **011** | **001** | **000** | **0000** | **0000** | **000** | **000** | **000** | **001** |

此条指令SCC为0000必定转移8位转移地址拼接命令码3在拼接必转得条件编码0000；MRW为010使用I/O进行写入，I2-0为011数据来源为B，R为0。

DC1为001送 ALU 输出到内部总线，且DC2为000无寄存器接收数据读主存， 读出内容 IR ; 微指令都是顺序执行，故 2910选 14# 命令.

及实现上述功能。

7.本条微指令完成STR→Q，CC#INT#操作；

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **30** | **3A** | **0011** | **0010** | **100** | **111** | **000** | **000** | **0000** | **0000** | **000** | **000** | **000** | **011** |

完成两个累加器之间的数据传送，DC1为011，把CZVS状态给到内部总线，再通过后续微操作将状态存入寄存器，之后如果要用的话，再取出。

I2-0为100，数据来源R为D，S为0，I5-3为000，实现功能R+S，I8-6为000，将运算结果存到Q寄存器，并且输出也为计算结果，不保存标志位的值，故SST为000，无存储器和 IO读写。

下址为3A，CI3-0为0011，条件转移，看SCC，SCC为0010，给出/MAP信号, 使D从映射部件得到。

8.本条微指令完成PC→AR，PC+1→PC，CC#=0操作；

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **31** | **02** | **0011** | **0000** | **100** | **011** | **010** | **000** | **0101** | **0101** | **000** | **001** | **011** | **000** |

DC2为011，即AR高位接收，通用寄存器R5便将地址存入了高位的地址寄存器，实现了R5→AR。

I2-0为011，数据来源R为0，S为B，I5-3为000，实现运算功能R+S，I8-6为010，将运算结果存入B口，但是输出A口的数据，注意到SSH为001，即Cin为1，实现了R5的自增+1,。

下址为02，CI3-0为0011，条件转移，下微指令转判中断请求，看SCC，SCC为0000，意思是必转，直接转到地址02即可，此时跳转至微址为02的微指令。