**大连海事大学**

**实践课程报告**

计算机组成原理课程实践

**学 院 计算机科学与技术学院**

**专 业 网络工程**

**课程名称 计算机组成原理课程实践**

**组序号 21**

**成员姓名 赤凯、梁渝宁、楚凡**

**指导教师 陈媛嫄、张少仲**

**起始日期 2021年7月12日——7月23日**

|  |  |  |
| --- | --- | --- |
| **小组完成的主要工作** | | 1. 复杂模型机设计实验（实验指导书上实验5.3）   (1)在 Proteus 上搭建电路图  (2)测试顺序、分支、循环、中断微程序  (3)设计三数加和微程序,并输入三组数据测试   1. 复杂模型机设计   （1)重新设计指令集、微指令集、微指令流程图  (2)设计新的寻址方式  (3)对新设计的内容设计微程序,进行测试 |
| **成**  **员**  **分工** | | **组长： 2220193816 赤凯**  **第一阶段电路图的连接，第二阶段指令的设计，实验报告的编写**  **成员1： 2220191761楚凡**  **第一阶段电路图的链接，功能的测试，实验报告的编写**  **成员2： 2220191447 梁渝宁**  **第一阶段电路图的链接，功能的测试，实验报告的编写** |
| **综合评语** | **（设计方案、实践环节、问题解答、设计报告）** | |

|  |  |
| --- | --- |
| 成  绩  评  定 | **成员1： 2220193816 赤凯**  **成员2： 2220191447 梁渝宁**  **成员3： 2220191761 楚凡** |

1. 复杂模型机设计实验（实验指导书上实验5.3或3.1）
   1. 实验原理

本实验的主要内容是掌握基于微程序控制器的 CPU 组成结构，了解 CPU 的中断工作机制，熟悉 CPU 微指令设计，掌握机器指令的微程序实现方法。本 实验将设计一个微程序 CPU,其中包括微程序控制器、运算器、存储器、寄存器 堆及外部 I/O 接口。定义一套较完备的机器指令集，编写每条机器指令对应的 微程序，在 CPU 电路上运行基于上述机器指令集的机器语言程序，并且用汇编 助记符(语言)加以注释。

* 1. 实验过程

（1）在 Proteus 上搭建好全部的实验电路，检查好序号和连线，标签等，将实验的各个部分做好标记。

（2）烧录微程序到 ROM 中。先新建出 txt 文件，把实验微程序全部写入到其中，再将 txt 文件改名为 asm 文件。

（3）测试实验指导书上的顺序结构程序。编译烧录 ADD.asm,初始化后先单步执行观察实验结果，后接入激励源自动执行观察实验结果。

（4）测试实验指导书上的分支结构程序。编译烧录 JS.asm,初始化后先单步执行观察实验结果，后接入激励源自动执行观察实验结果。

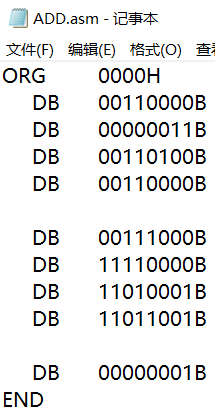
（5）测试实验指导书上的循环结构程序。编译烧录 SOP\_JZ.asm,初始化后先单步执行观察实验结果，后接入激励源自动执行观察实验结果。

（6）测试实验指导书上的中断程序。编译烧录 INT\_IRET.asm,初始化后先单步执行观察实验结果，后接入激励源自动执行观察实验结果。

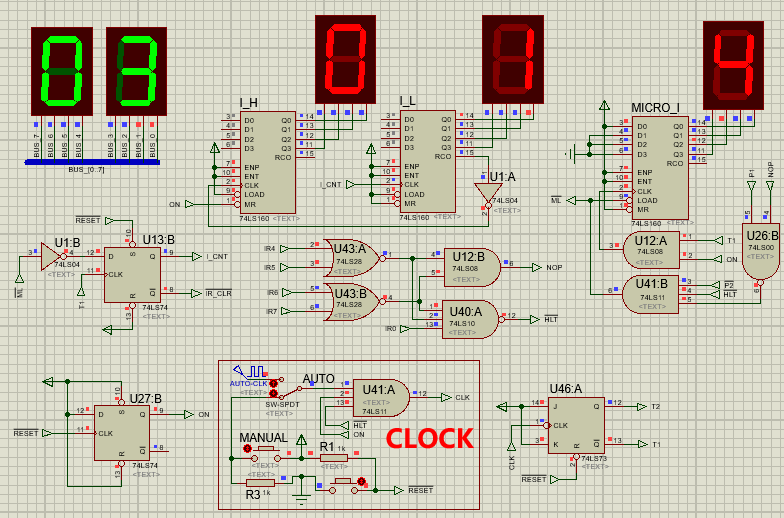
（7）自己设计三数相加微程序，基本思路是将实验要求的三组测试数据相加。编译烧录三数相加.asm，初始化后先单步执行观察实验结果，后接入激励源自动执行观察实验结果。

* 1. 实验结果

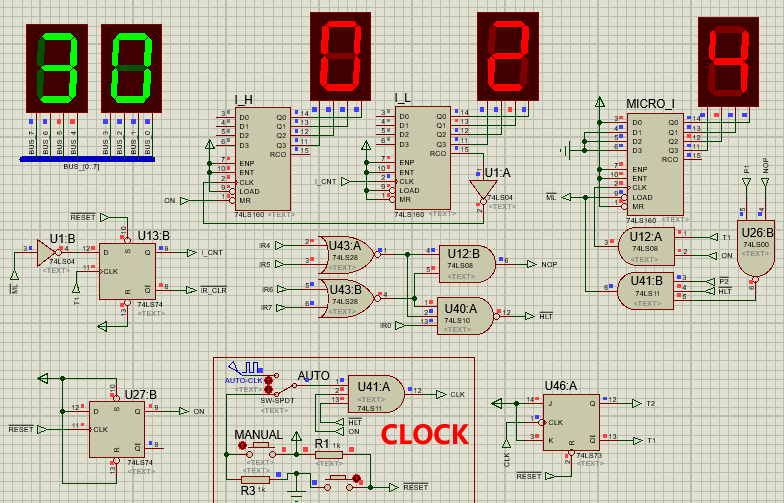
实验一：



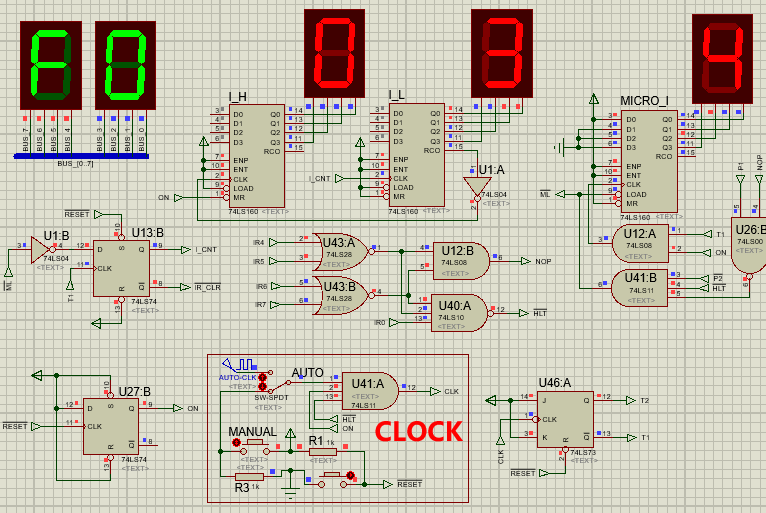
R0值：



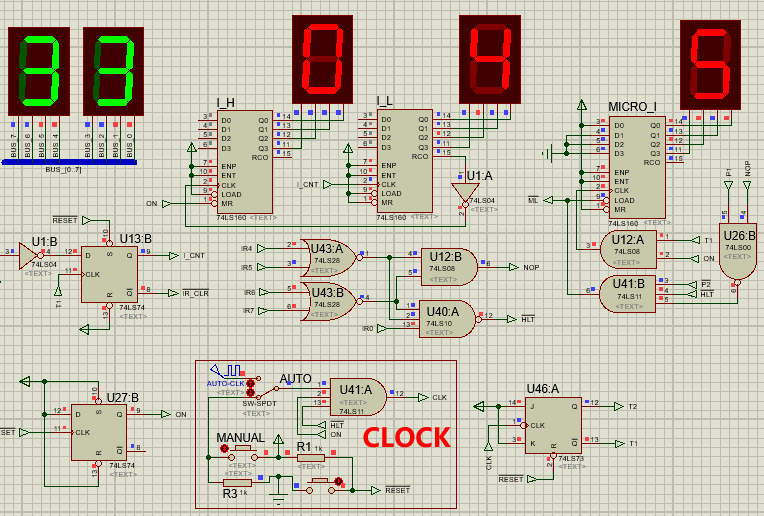
R1值：



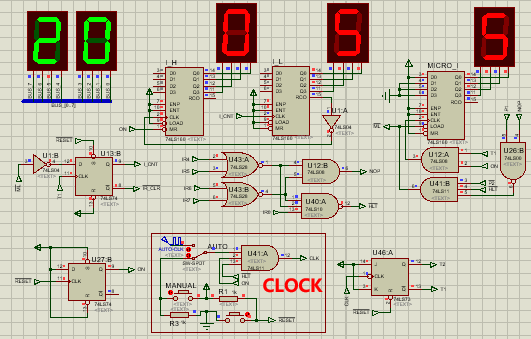
R2值：



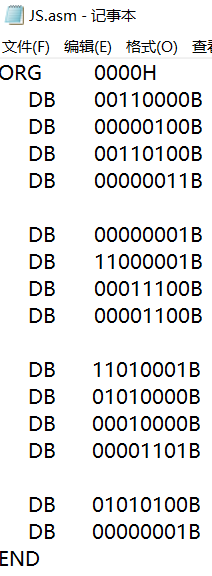
R0+R1的值：（先显示R0和R1的值分别为03和30，相加结果为33）



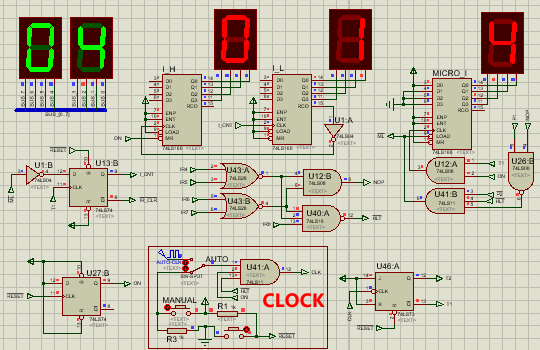
R2+R1的值：（再显示R2和R1的值分别为F0和30，相加结果为20）



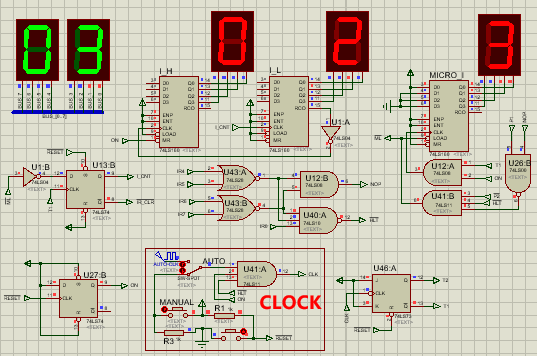
实验二：用于比较寄存器数据大小



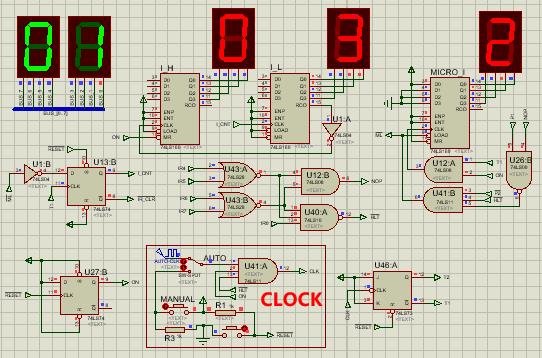
给R0赋值：



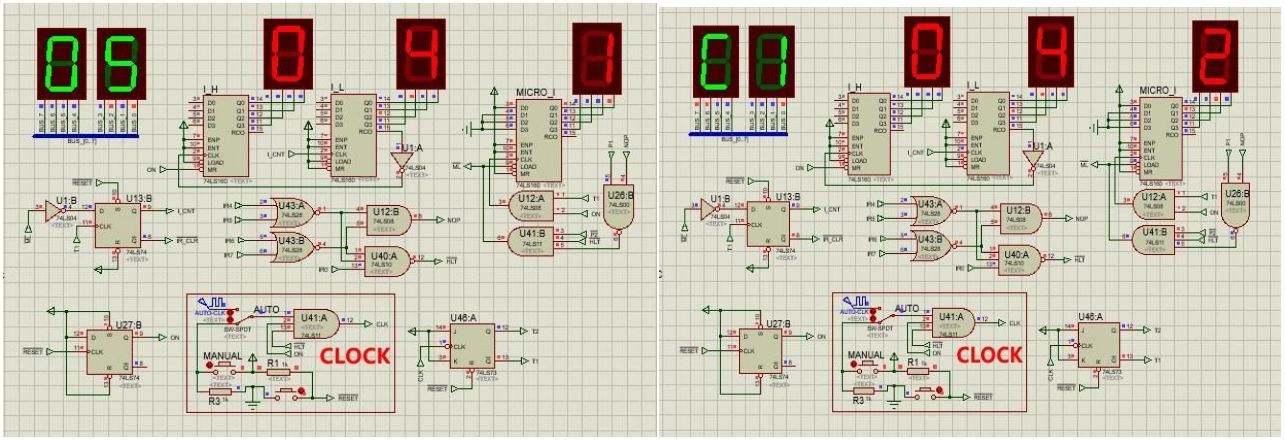
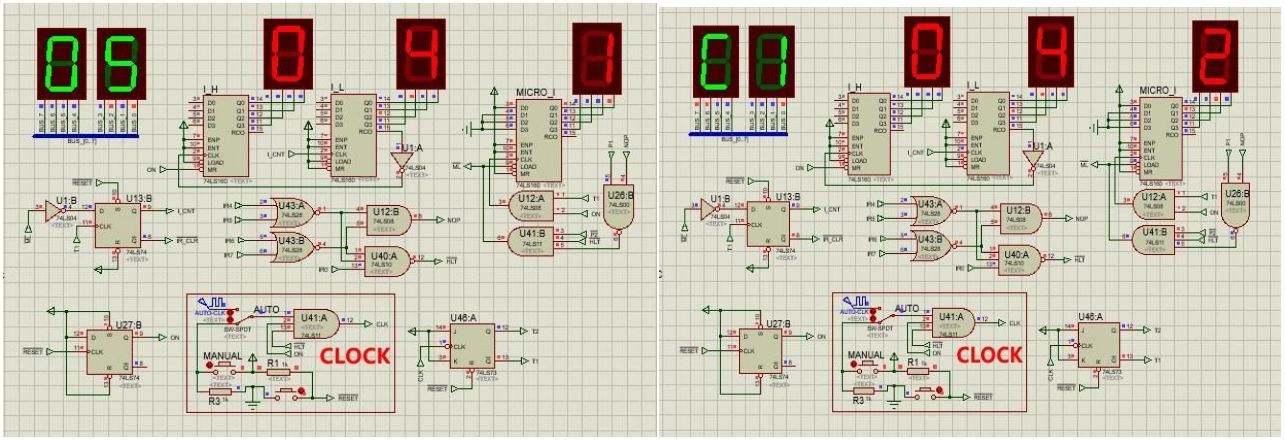
给R1赋值：

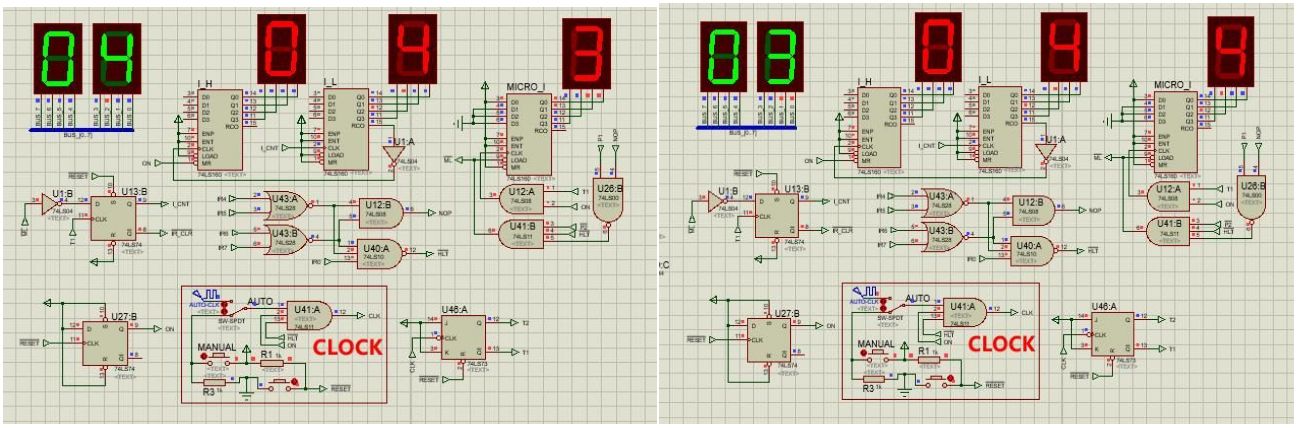
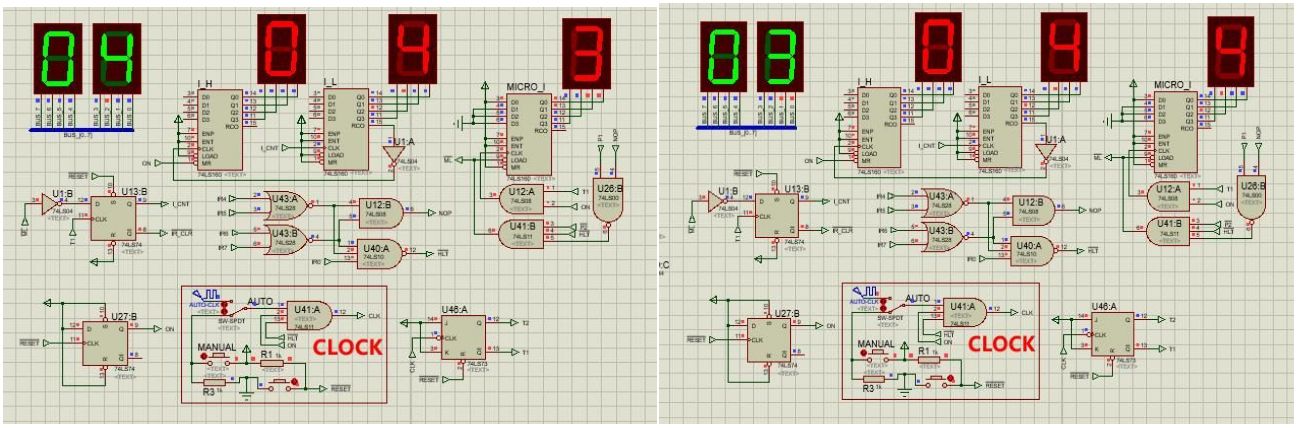


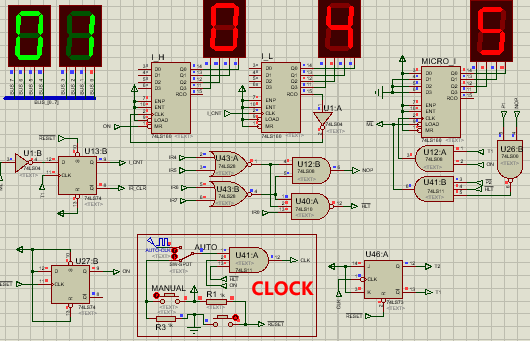
第一个断点：



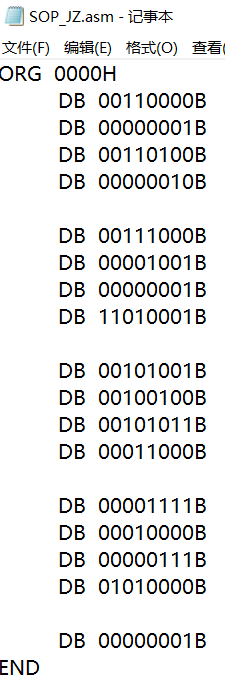
R0-R1：（直到第二个断点）

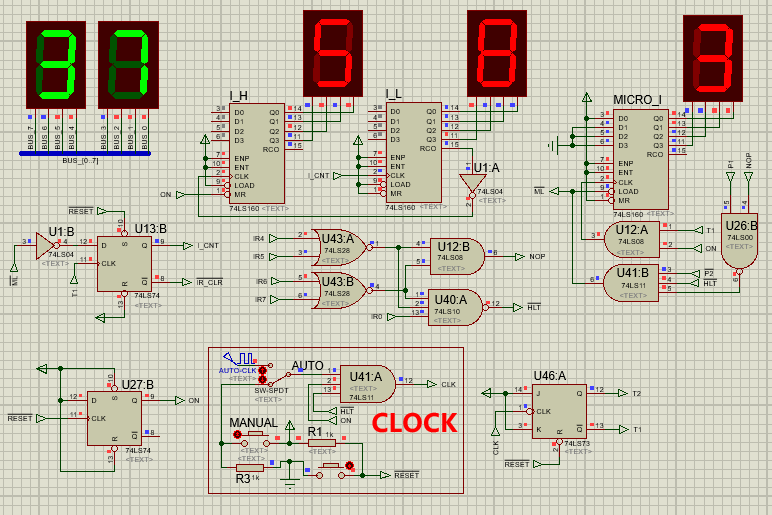






实验三：



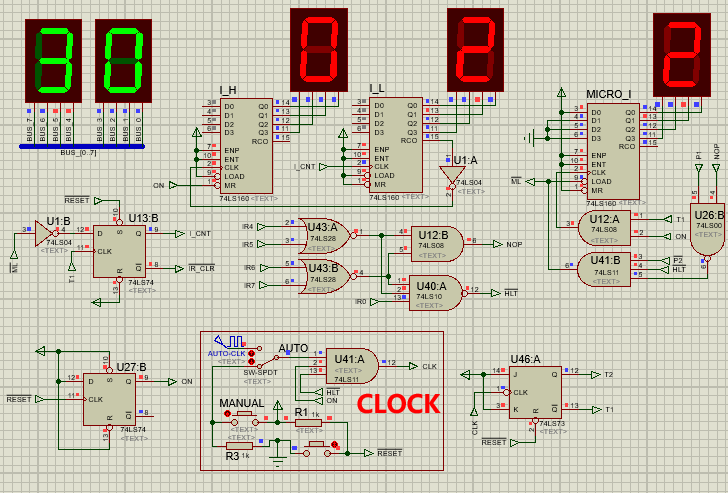
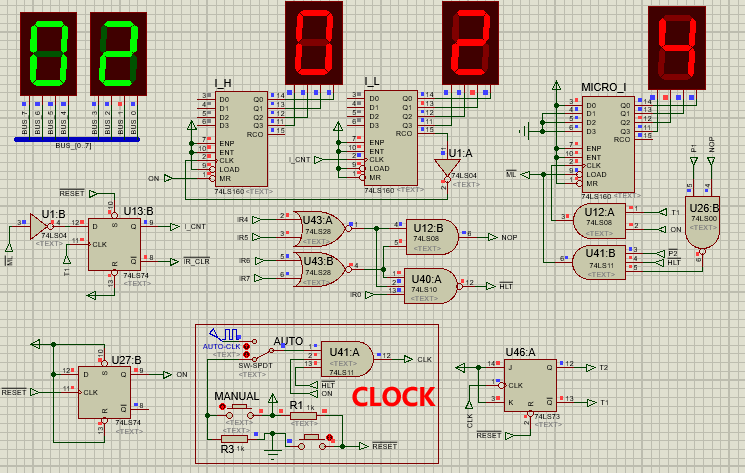


一直循环，得到最终结果（55 的 16 进制是 37H）。

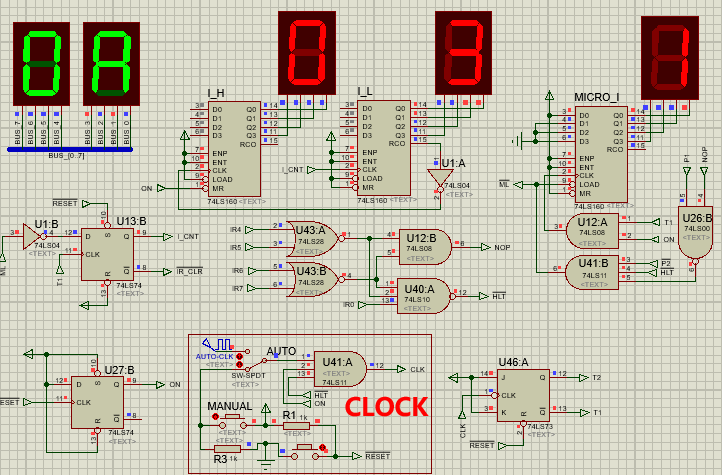
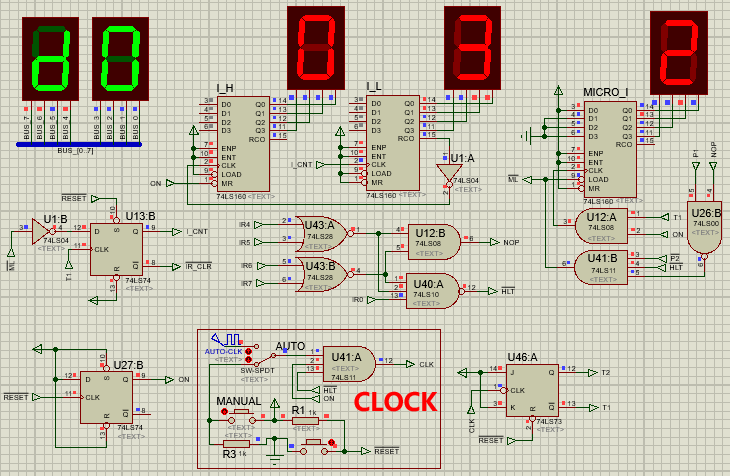
实验四：中断程序

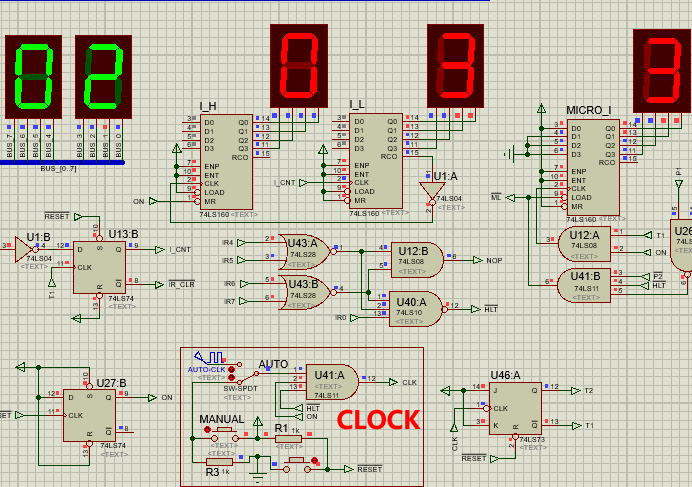
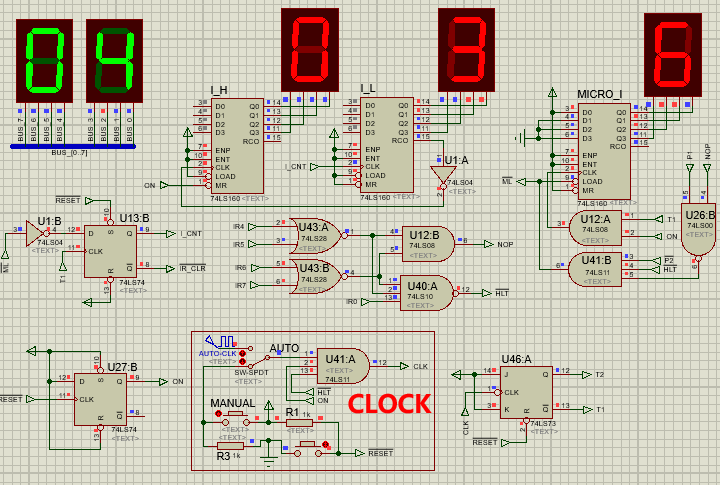
从第九行开始分析：

第九行、第十行：给R0赋值02H

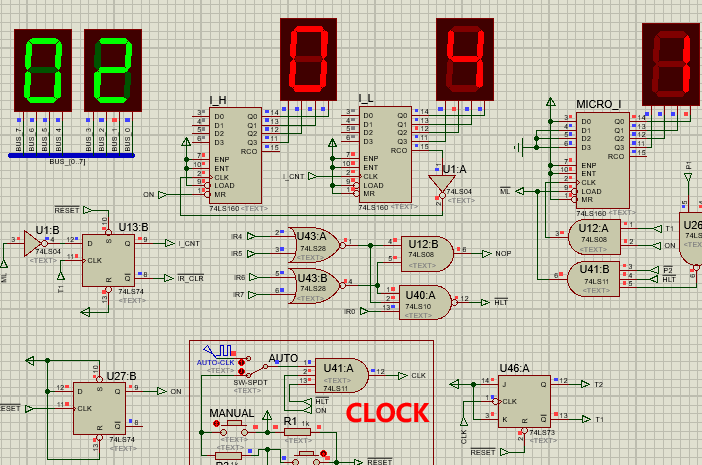
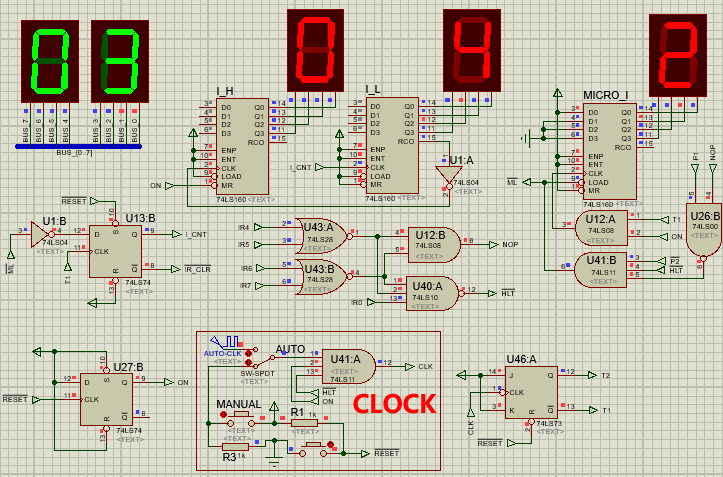
 

第十一行、第十二行：R0+02H=04H

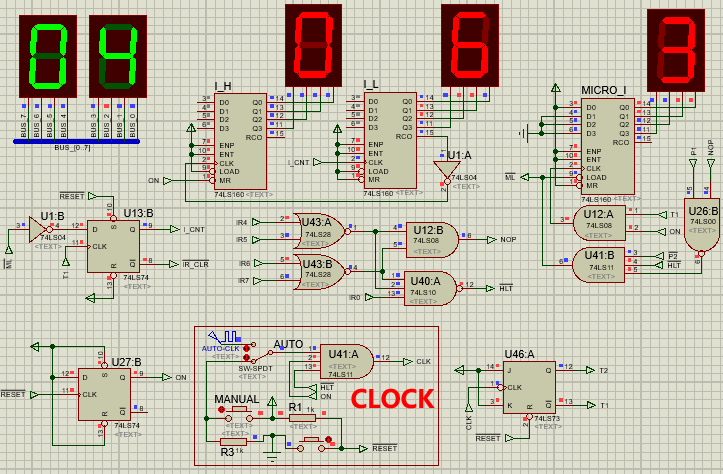
 

中断后进入中断向量地址：

跳过停机执行out，输出R0此时的值：

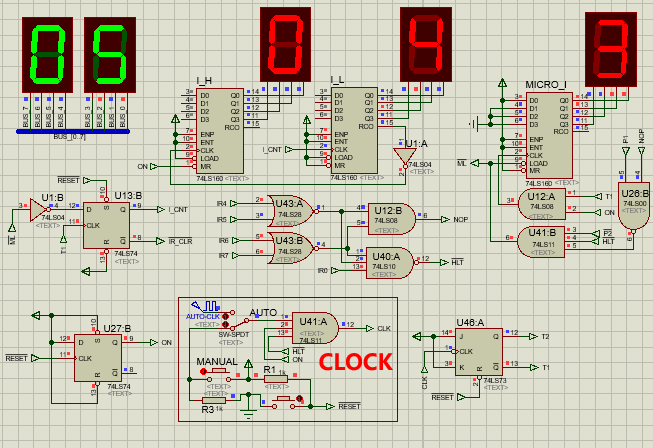
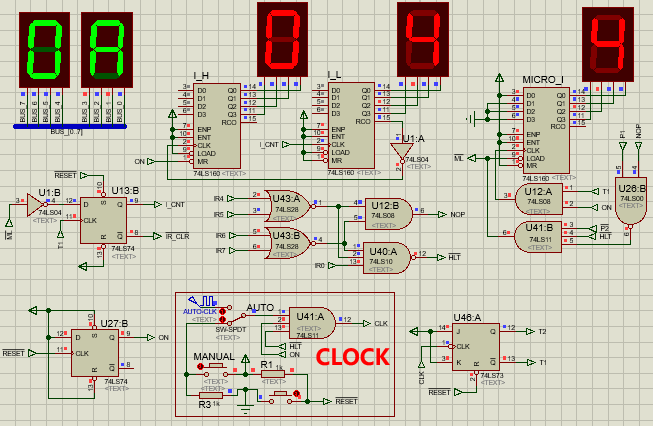


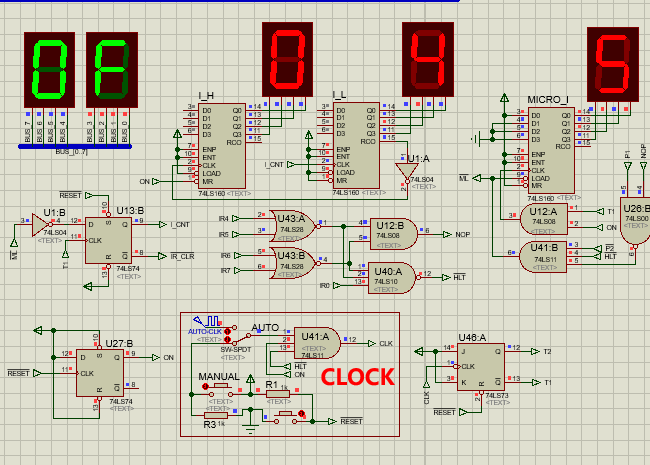
接着顺序执行到第八行进入主程序，循环ADDI，直到手动中断。

实验五：

修改R0、R1、R2的数据分别为05H、0AH、0EH。

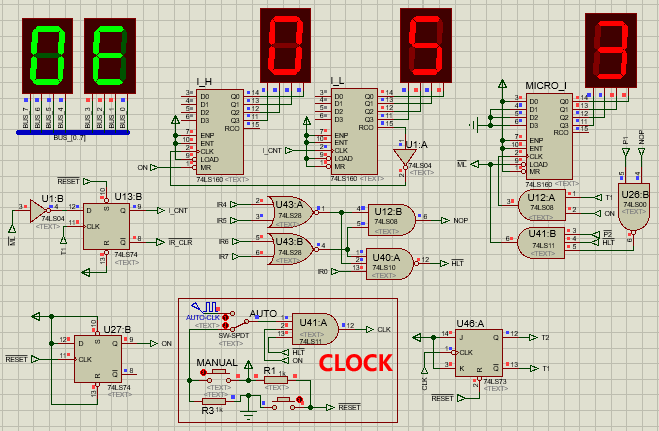
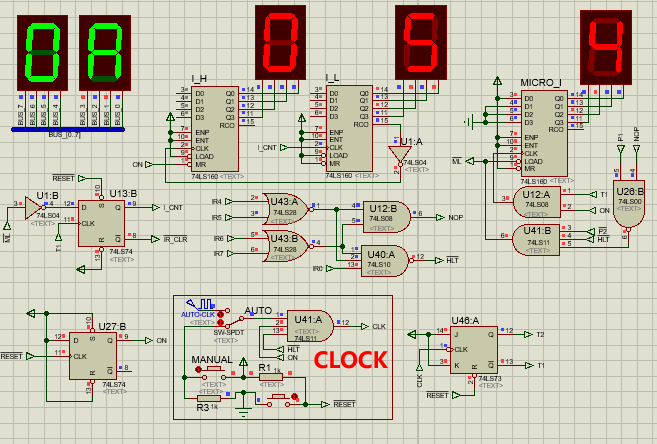
R0+R1：

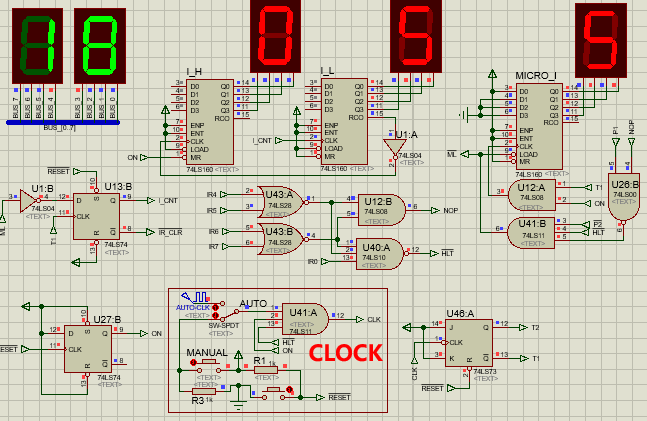
 



结果为0FH。

R2+R1：

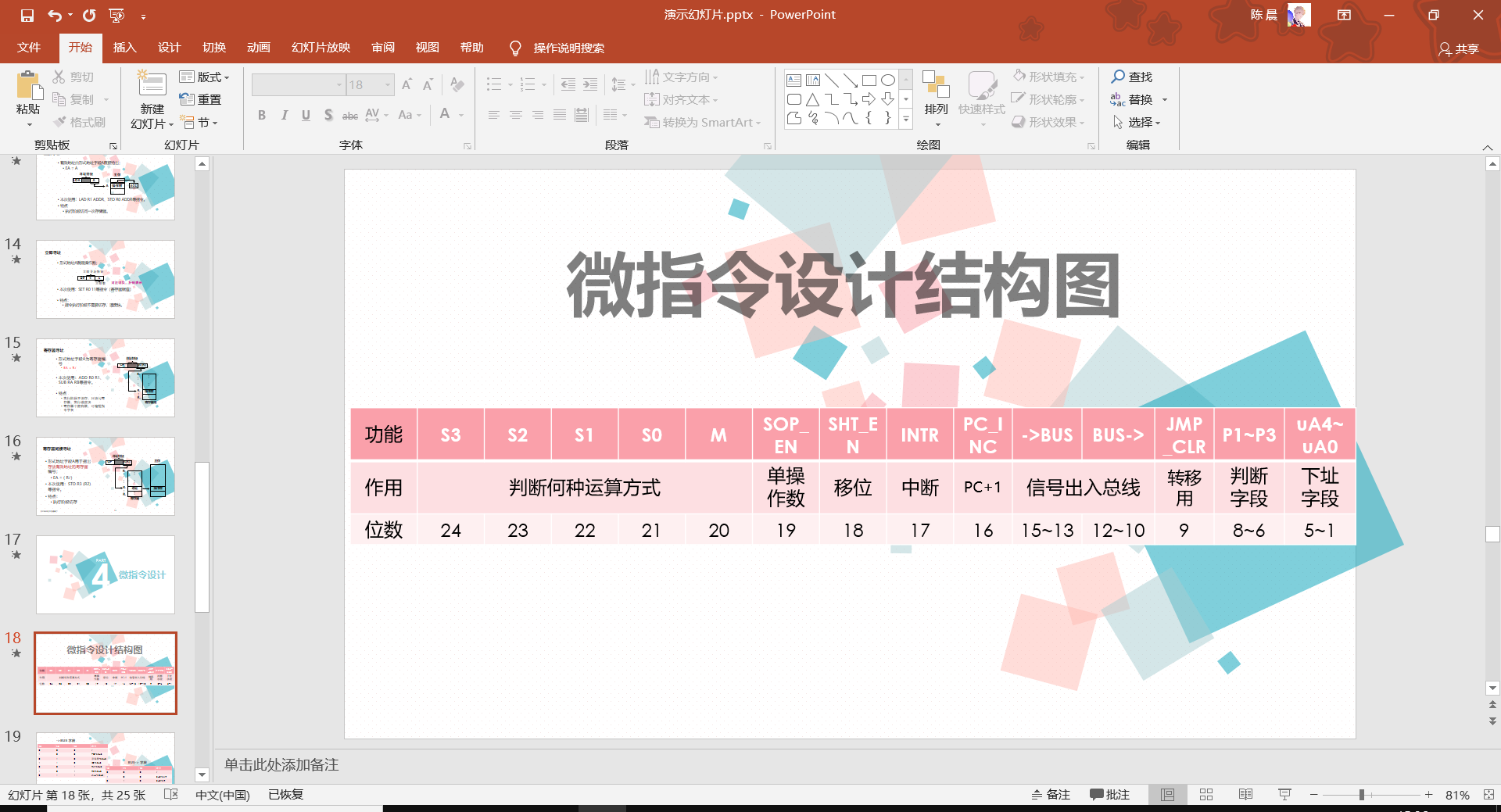
 

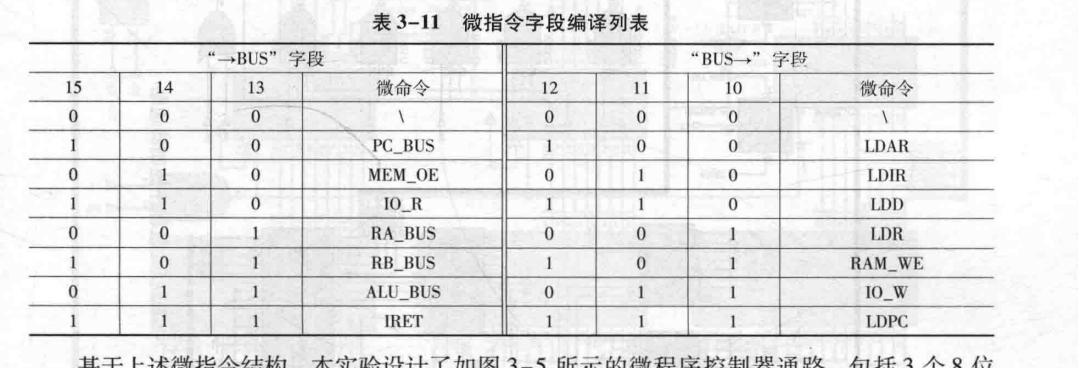


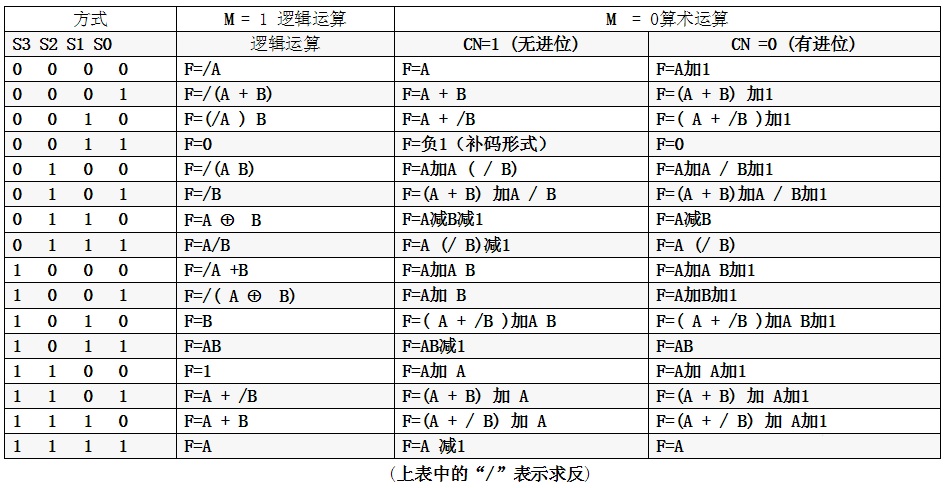
结果为18H。

1. 复杂模型机设计

通过对电路的分析得出微指令每位的含义：



****

****

* 1. **指令系统的设计、寻址方式的设计**

1. **指令系统设计： 通过修改微指令，设计一条相加求反和算数与的指令，，思路是通过控制ALU的信号来进行运算.**

**设计一条取反加一（补码）的指令，思路是将取反和加一两条指令操作连接在一起，主要是通过对ALU信号的控制实现。**

1. **寻址方式的设计：将相加指令ADD 的寄存器寻址改为寄存器间接寻址。思路是先在RAM指定地址中存入一个数据，在执行阶段先将第一个寄存器中的数据取出放入暂存器DA,再将第二个操作数中存入的地址取出放入AR,然后打开RAM，选中相应的地址，将其中的数据取出放入暂存器DA进行运算。**
   1. **微指令的设计、流程图**

**1. 指令系统设计**

**（1） /(A + B) 微指令如下：**

**00000000 01001000 00010101//PC\_BUS LDAR**

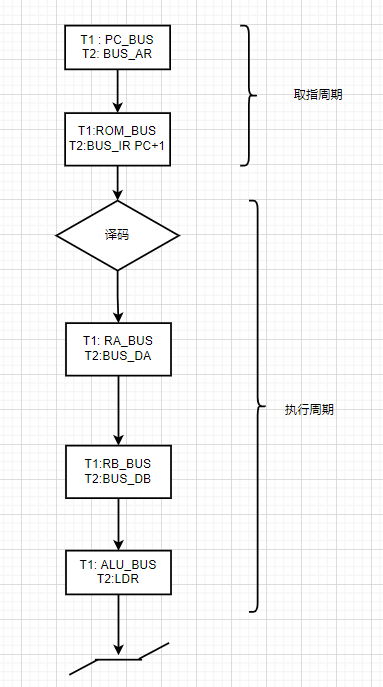
**00000000 10100100 10000000//PC\_INC MEM\_OE LDIR 10101**

**00011000 00011100 00111000//给ALU信号，RA->BUS,BUS->DA,**

**00000000 01011100 00010100//RB->Bus.BUS->DB**

**00000000 00110010 01000000 //ALU\_BUS LDR P2**

**微指令流程图：**



**（2）取反加一（～A+1）微指令**

**00000000 01001000 00010101//PC\_BUS LDAR**

**00000000 10100100 10000000//PC\_INC MEM\_OE LDIR 10101**

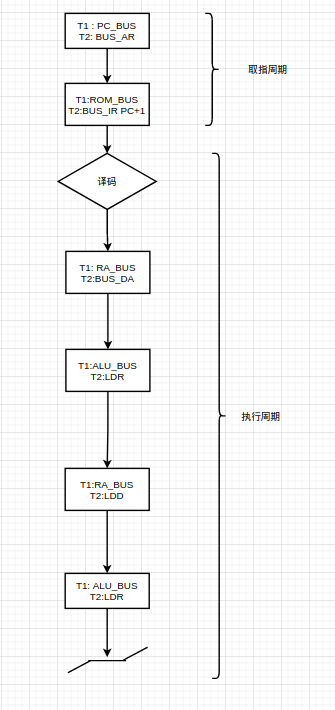
**00001000 00011100 00000011 // ALU 给取反信号，RA\_BUS LDD**

**00000000 00110010 00000100 // ALU\_BUS LDR**

**00000000 00011100 00010100 // ALU给加一信号，RA\_BUS LDD**

**00000000 00110010 01000000 //ALU\_BUS LDR P2**

**流程图：**



**2．寻址方式设计**

**ADD 间接寻址**

**00000000 01001000 00010101//PC\_BUS LDAR**

**00000000 10100100 10000000//PC\_INC MEM\_OE LDIR**

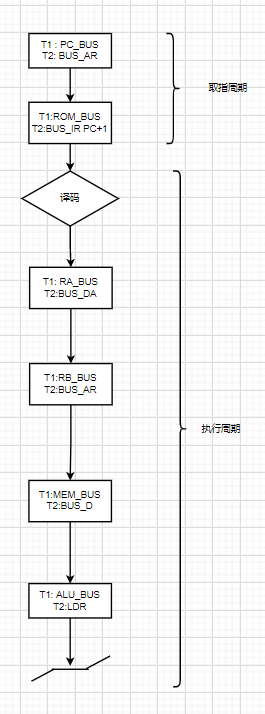
**10010000 00011100 00111000 // RA\_BUS BUS\_D p3**

**00000000 01011000 00011100 // RB\_BUS BUS\_AR**

**00000000 00101100 00010100 // MEM\_BUS BUS\_D**

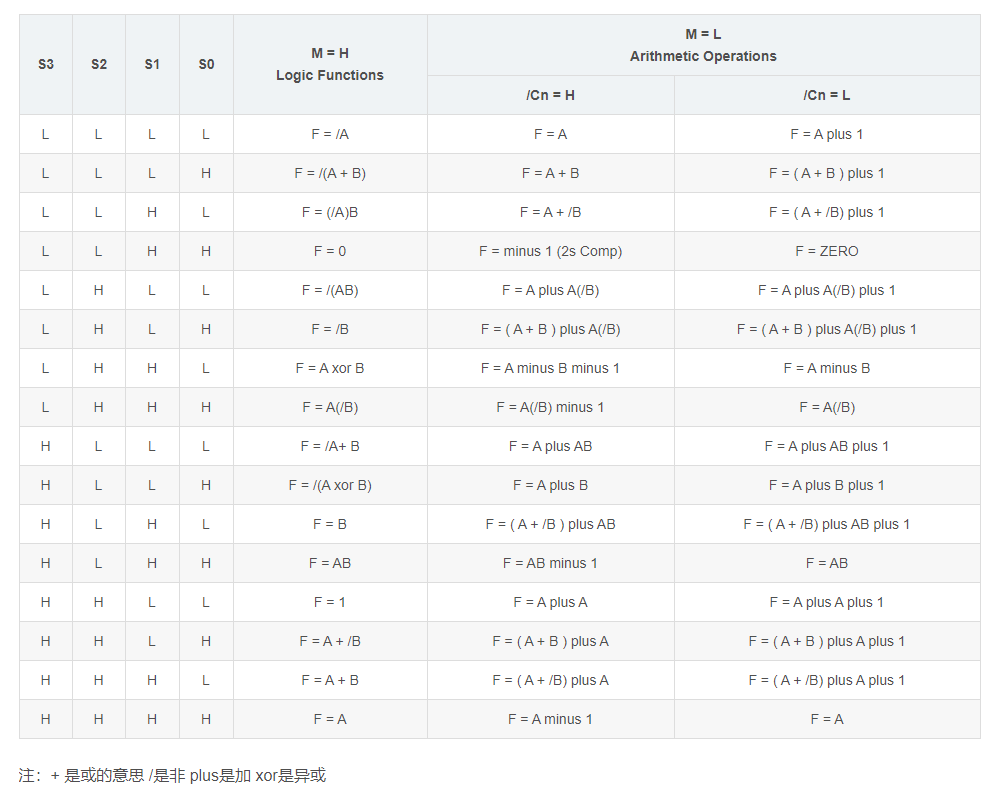
**00000000 00110010 01000000 //ALU\_BUS LDR P2**

**寻址方式流程图：**



* 1. **各组成部分详细设计过程（重点描述本组实际完成的工作）**

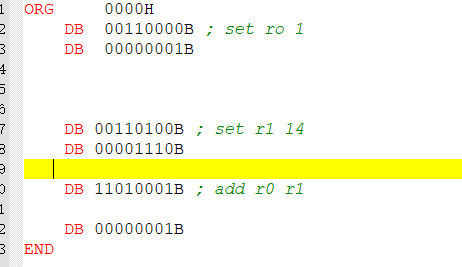
1. **指令系统设计：**
2. **确定需求。计划设计一个新的算数指令。**
3. **过程。通过对电路的研究，查阅相关的芯片参数，对微指令的各位进行研究。**
4. **具体设计。经过查阅ALU芯片的参数，**



**当我们给 S3——S0,M 相应的操作时，ALU进行相应的操作**

**现在给00011 进行 相加后取反的操作**

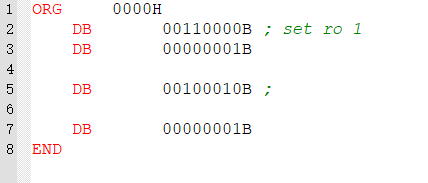
1. **验证程序**



1. **基于此设计一条取补码（取反加一）的指令。**

**思路是 先将操作数放入ALU中进行取反操作，再将ALU的结果放入R中暂存，然后将R中的值重新放入ALU进行加一操作。**

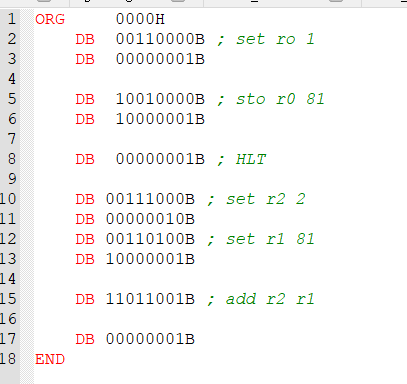
**验证程序**

****

1. **寻址方式设计**
2. **需求设计。对ADD进行改造，设计一个能进行间接寻址的ADD操作**
3. **具体设计**

**先将第一个操作数RA放入暂存器DA,并给予ALU加法信号。将第二个寄存器内的值通过BUS放入AR中，地址译码。操作码OP的第一位控制对RAM/ROM的读取，当为1时，读取RAM.然后将数据取出放入到暂存器DB中，与DA做运算后结果放入RB**

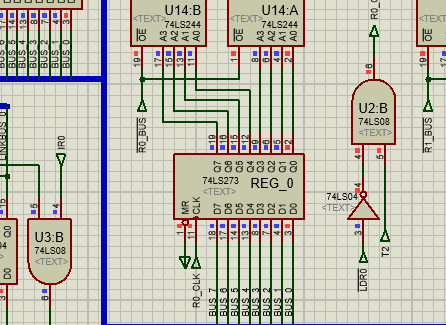
1. **验证程序**

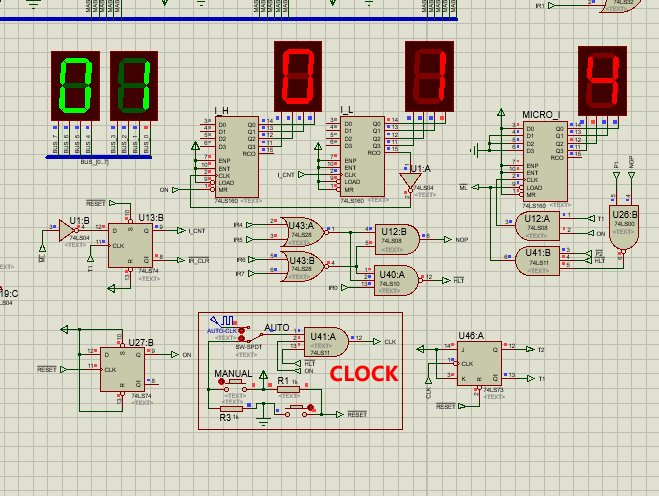


* 1. **实验结果**（拷屏或拍照）

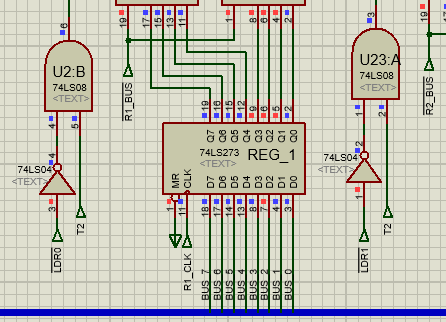
1. **指令设计**
2. **~(A+B)指令**

**将1放到R0 中**

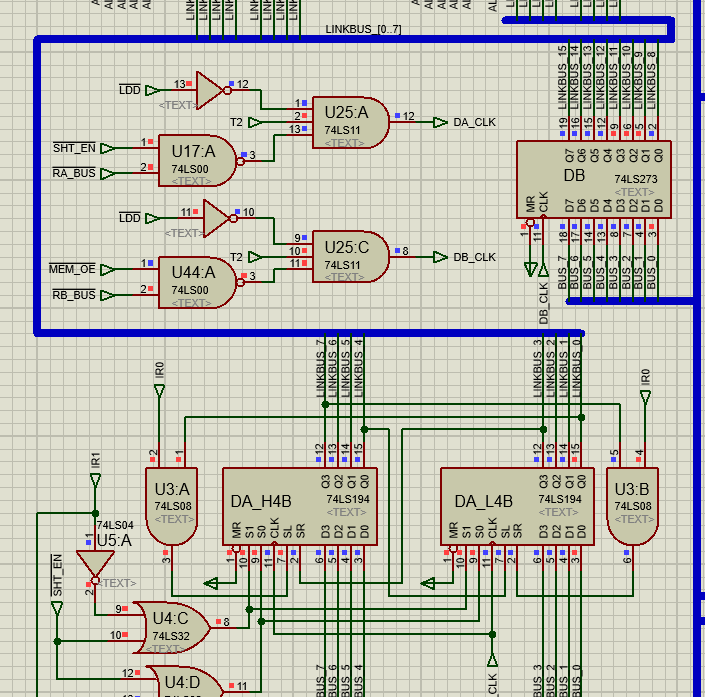




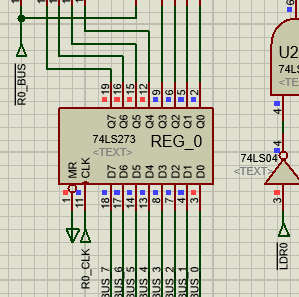
**将 14 放在 R1 中**



**将R0 的值 放到 DA，R1 放到 DB暂存器中**

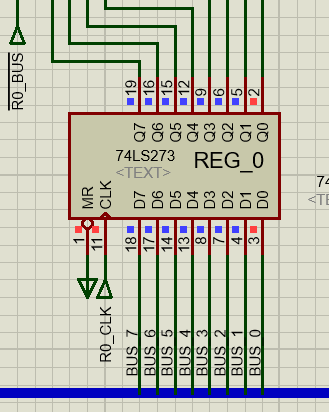


**ALU 经过 求和取反运算 将结果返回到 R0中**

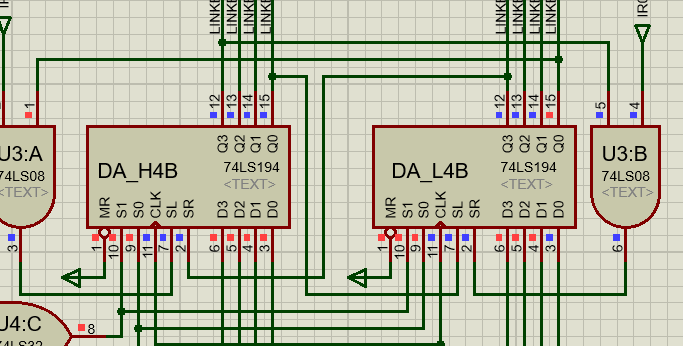


1. **~A+1 指令**

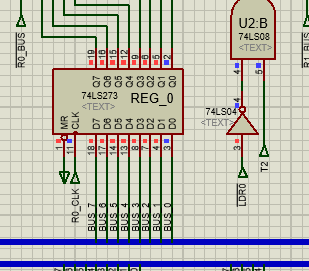
**将 RO 置为 1**

****

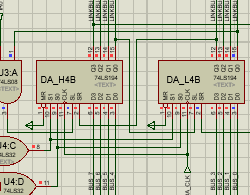
**将 RO 的值传入 ALU**

****

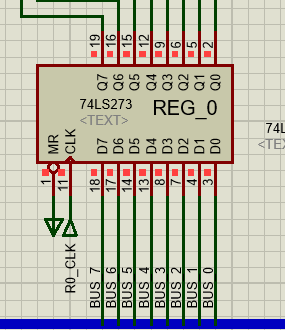
**将取反操作之后的值传入RO**

****

**再将RO的值传入ALU进行加一操作**

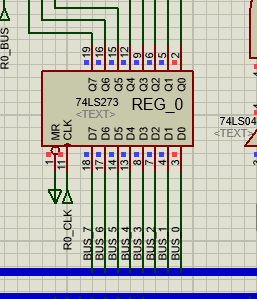
****

**将结果放入R0**

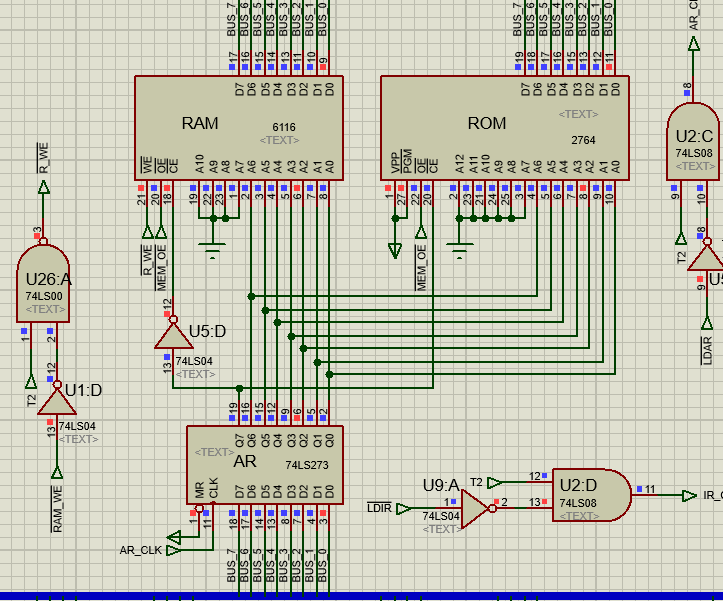
****

1. **取址方式设计**

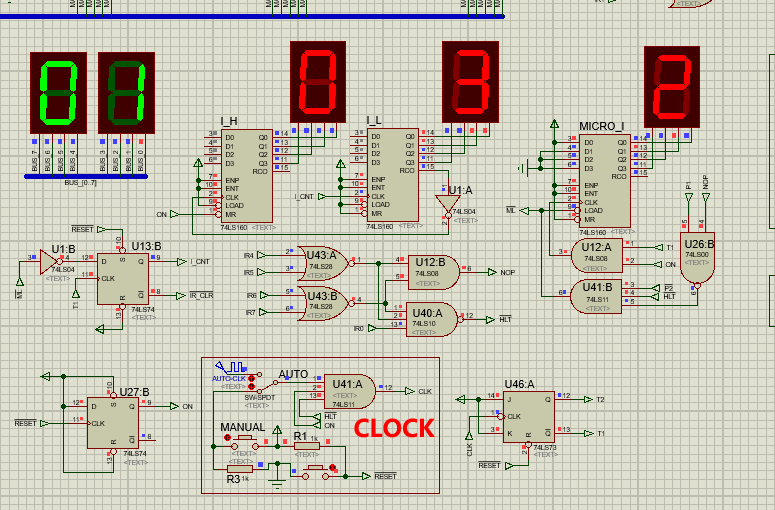
**将R0 置为 1**



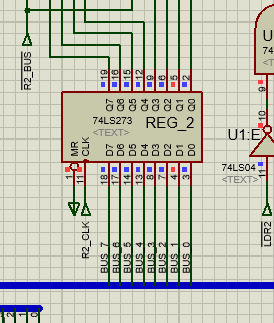
**将R0 中的值赋值给 81地址**



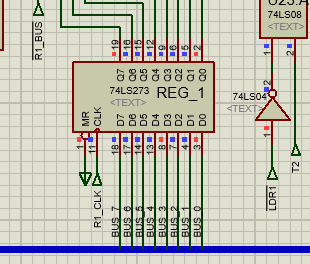
**中断**



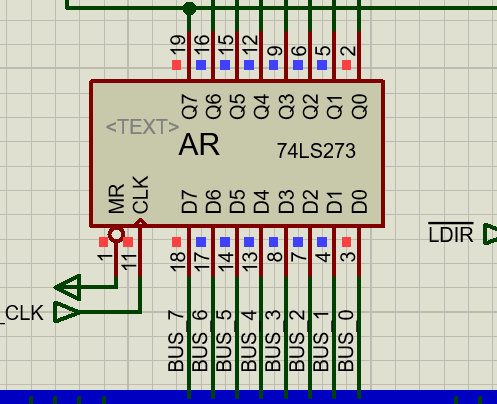
**将R2 置为 2**



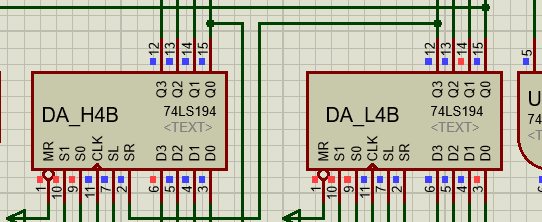
**将R1 置为 81**



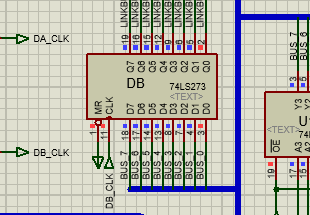
**将 81 送到 AR 中**



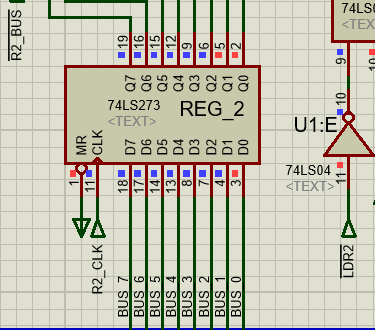
**将 R2 送入 DA 中**



**将RAM 中81的数据送入DB中**



**将ALU的结果送入R2**



1. **课程设计总结**

**通过此次复杂模拟机实验，我们对CPU构造和原理理解更加深入了。在第一个阶段，我们通过搭建电路图对CPU的构造进行了深入理解，并对基本功能进行了解。第二阶段，在第一阶段的基础上，我们通过查阅文档资料，清楚了微指令的构造和指令的执行过程，构造了新的指令和寻址方式。在此次实验中，我们小组互帮互助解决了各种问题，在实验过程中不仅加深了对知识的理解，还锻炼了动手能力和查阅文档的能力，更重要的是培养了团队合作的能力。**

1. **参考文献**

1、赖晓铮,2017.基于 PROTEUS 的计算机系统实验教程:逻辑、组成原理、体系结构、微机接口,北京:机械工程出版社

2、白中英、戴志涛,2019.计算机组成原理,北京:科学出版社