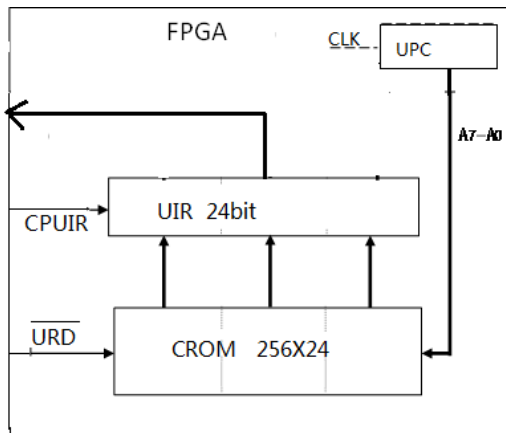


计算机组成与设计

课程实验报告

学号：202200400053		姓名：王宇涵	班级：2202
实验题目：控制器实验			
实验学时：2		实验日期：2024-06-13	
实验目的： 掌握存储逻辑控制器的工作原理并设计运行			
实验软件和硬件环境： 软件环境： QuartusII 软件 硬件环境： 1.实验室台式机 2.计算机组成与设计实验箱			
实验原理和方法： 目前控制器设计大都采用微程序设计方法，又称存储逻辑控制器。微程序控制器电路结构如图 13-1 所示。它由控制存储器 CROM、微程序 μPC 计数器和微指令寄存器 μIR 构成。 其中，微程序计数 μPC 向控制存储器提供 8 位微地址，在控存读信号 $\overline{\mu RD}$ 的作用下，读出一条长 24 位的微指令代码，并在打入命令 $CP \mu IR$ 的作用下，送入 μIR 。			
<div></div> <div>图 13-1 微程序控制器框图</div>			
每当按一次脉冲键便产生一个负脉冲，该脉冲的作用是：			
<div><ul style="list-style-type: none">• 作为读控存的命令 $\overline{\mu RD}$。• 负脉冲当作 $CP \mu IR$ 将读出的微指令打入微指令寄存器 μIR。• 负脉冲的上升沿使 $\mu PC + 1$ 形成下一条微指令的地址。• 负脉冲反相后的上升沿作为寄存器打入脉冲。</div>			
(1) 微程序计数器 μPC 的设计，完成 8 位具有加 1 功能和清除功能的计数器设计并封装，如图			

13-3。

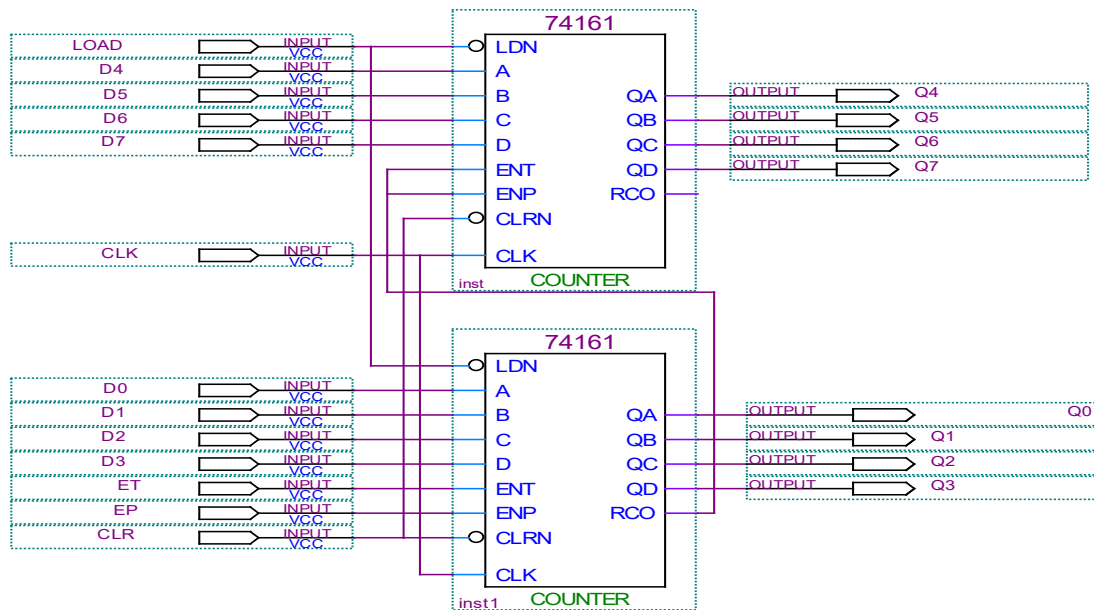


图 13-3 UPC 的设计图

其中，CLR：清零端，，低电平有效；CLR=0 时，Q7Q6Q5Q4Q3Q2Q1Q0=00000000；

LOAD：置数端，低电平有效；LOAD=0 时，在 CLK 的上升沿，

Q7Q6Q5Q4Q3Q2Q1Q0=D7D6D5D4D3D2D1D0；

当 CLR=1，LOAD=1，ET=1，EP=1 时，对 CLK 进行增 1 计数。

注意：本实验使用时，只有 clk、CLR 两引脚引出，其它引脚，ET、EP、LOAD 接高电平。

(2) 按图 13-1 完成微程序控制器的连线及引脚锁定。

* 用单脉冲驱动 μ PC 的计数脉冲 CPPC、 μ RD、CP μ IR。

* 将 μ PC 的 8 位输出锁定在 A7—A0 上。

* 实验平台工作于模式 5，将单脉冲锁定于键 8，将 μ PC 的复位端 CLR 锁定在键 7，输出 UIR 的 24 位接数显 3—数显 8。

(3) 设计并初始化控制存储器。

从 ROM 的 0 号单元开始写入、写入的微程序代码随意确定。

(4) 读出微指令

◆按 CPU 复位按键，清除微指令计数器。

◆按单脉冲键

读出 0 号单元中的微指令代码送 μ IR23- μ IR0 同时显示在数显上；

将微指令计数器 μ PC 十 1 为读下条微指令做准备。

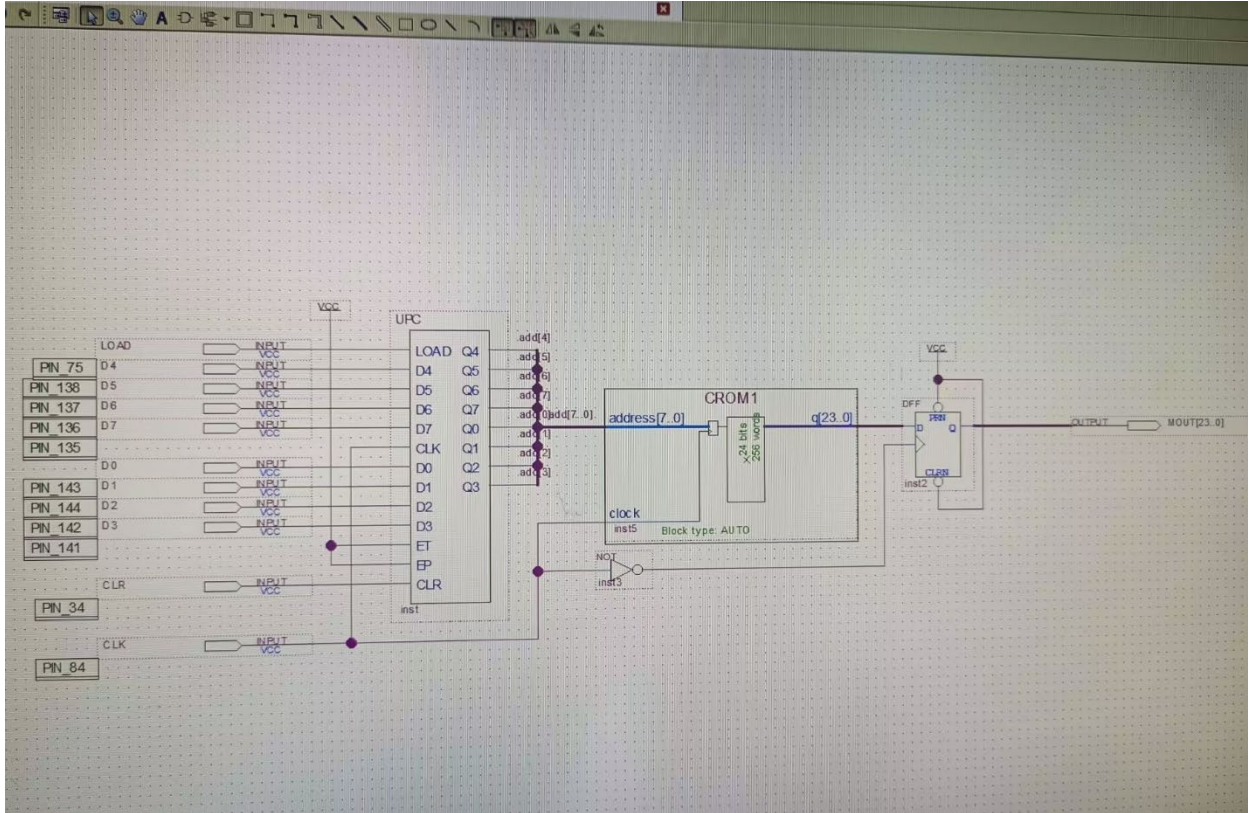
◆连续按单脉冲键

微指令连续从 ROM 中读出并显示。

注意：在读出过程中要和原先写入的代码比对看是否正确

实验步骤:

(1) 原理图输入：根据图所示电路，完成电路原理图设计。



(2) 管脚锁定：完成原理图中输入、输出的管脚锁定。

引脚分配如下：

CLK——键 8——PIO7——PIN84

CLR——键 7——PIO6——PIN34

LOAD——键 6——PIO5——PIN75

D7—DOUT8—DA7—PIO40—PIN135

D6——DOUT7——DA6——PIO41——PIN136

D5——DOUT6——DA5——PIO42——PIN137

D4—DOUT5—DA4—PIO43—PIN138

D3—DOUT4—DA3—PIO44—PIN141

D2——DOUT3——DA2——PIO45——PIN142

D1—DOUT2—DA0—PIO47—PIN144

D0—DOUT1—DA1—PIO46—PIN143

MOUT23——译码器 6 第 4 位——PIO39——PIN125

MOUT22——译码器 6 第 3 位——PIO38——PIN128

MOUT21——译码器 6 第 2 位——PIO37——PIN114

MOUT20——译码器 6 第 1 位——PIO36——PIN120

MOUT19——译码器 5 第 4 位——PIO35——PIN105

MOUT18——译码器 5 第 3 位——PIO34——PIN113

MOUT17——译码器 5 第 2 位——PIO33——PIN100

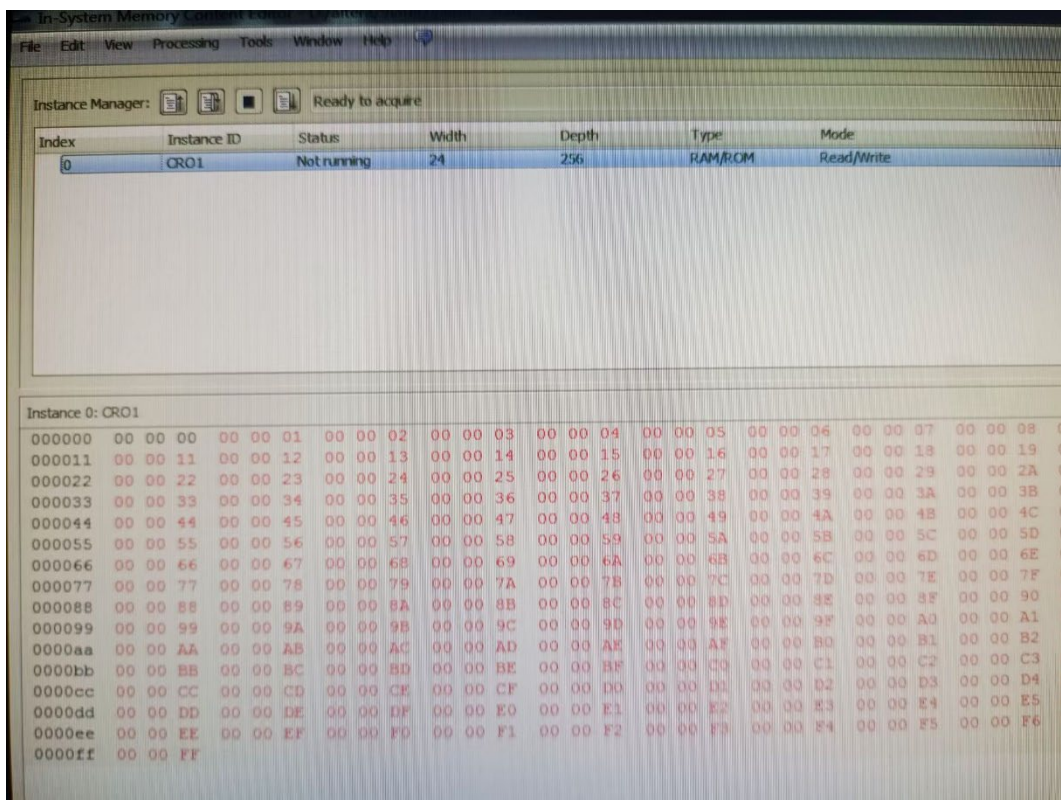
MOUT16——译码器 5 第 1 位——PIO32——PIN101

MOUT15——译码器 4 第 4 位——PIO31——PIN80

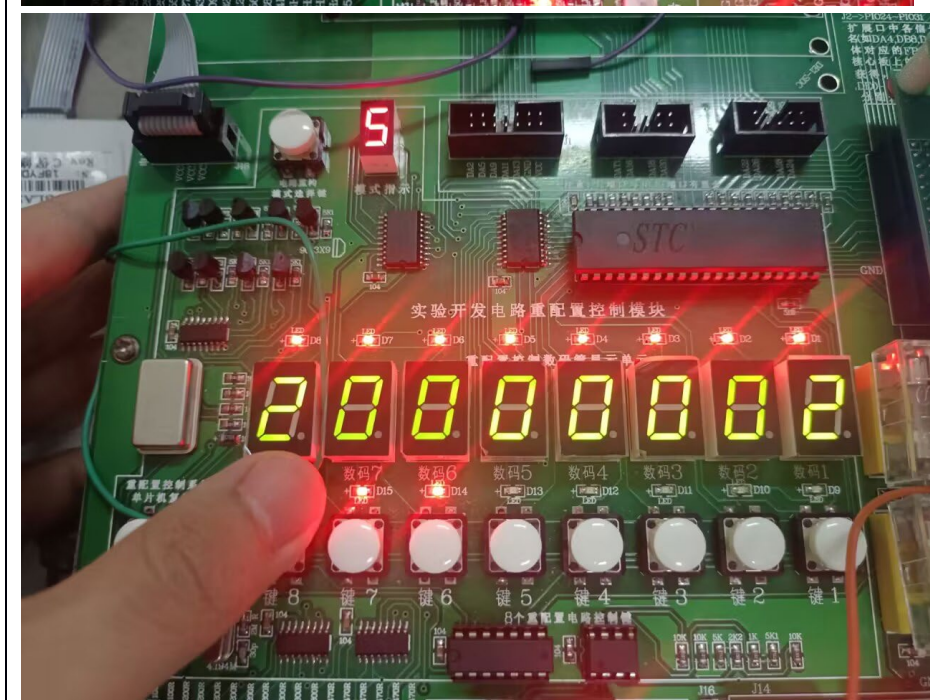
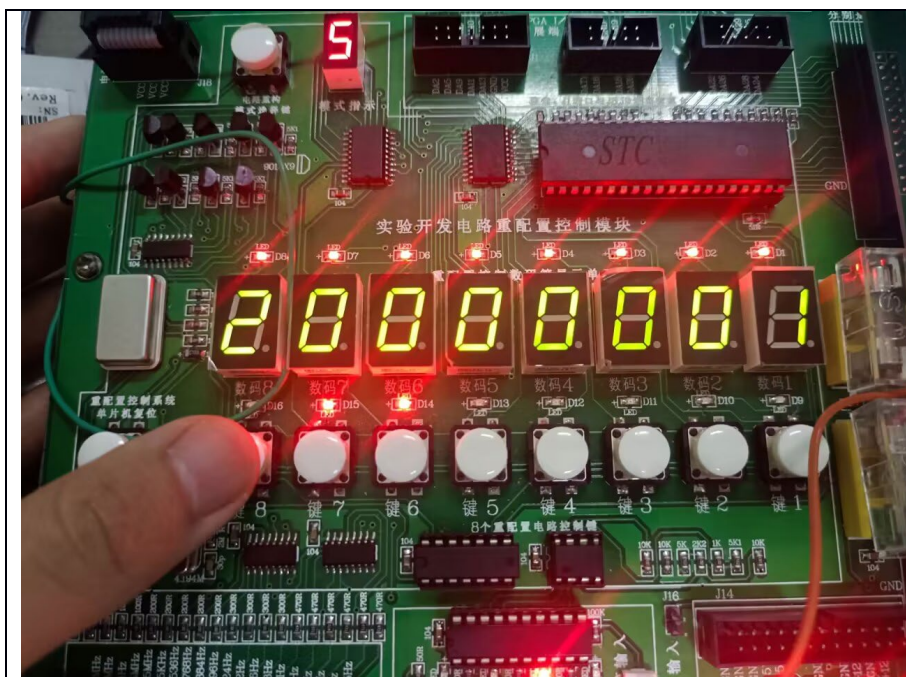
MOUT14——译码器 4 第 3 位——PIO30——PIN85
 MOUT13——译码器 4 第 2 位——PIO29——PIN73
 MOUT12——译码器 4 第 1 位——PIO28——PIN76
 MOUT11——译码器 3 第 4 位——PIO27——PIN71
 MOUT10——译码器 3 第 3 位——PIO26——PIN72
 MOUT9——译码器 3 第 2 位——PIO25——PIN68
 MOUT8——译码器 3 第 1 位——PIO24——PIN69
 MOUT7——译码器 2 第 4 位——PIO23——PIN54
 MOUT6——译码器 2 第 3 位——PIO22——PIN59
 MOUT5——译码器 2 第 2 位——PIO21——PIN50
 MOUT4——译码器 2 第 1 位——PIO20——PIN51
 MOUT3——译码器 1 第 4 位——PIO19——PIN46
 MOUT2——译码器 1 第 3 位——PIO18——PIN49
 MOUT1——译码器 1 第 2 位——PIO17——PIN43
 MOUT0——译码器 1 第 1 位——PIO16——PIN44

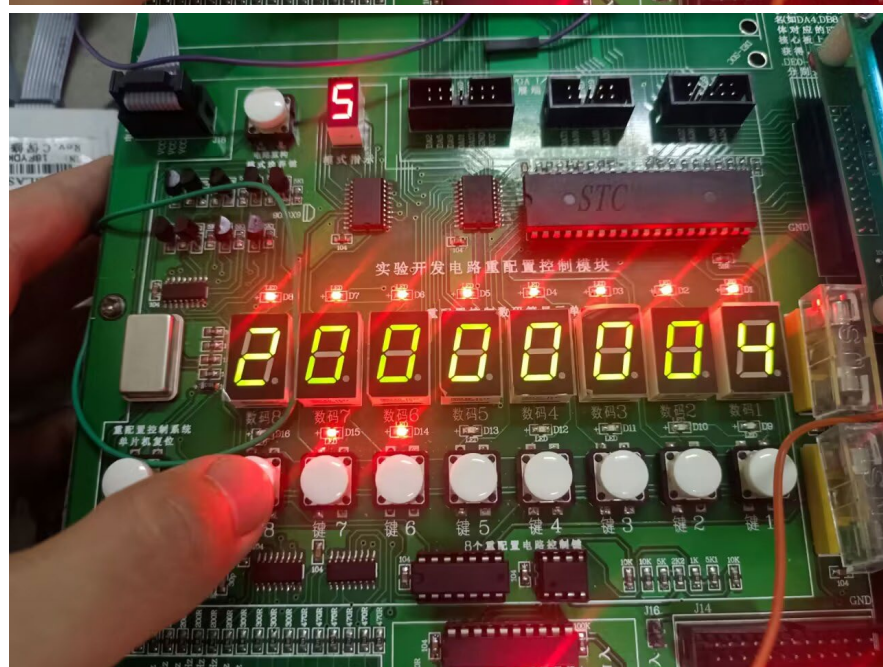
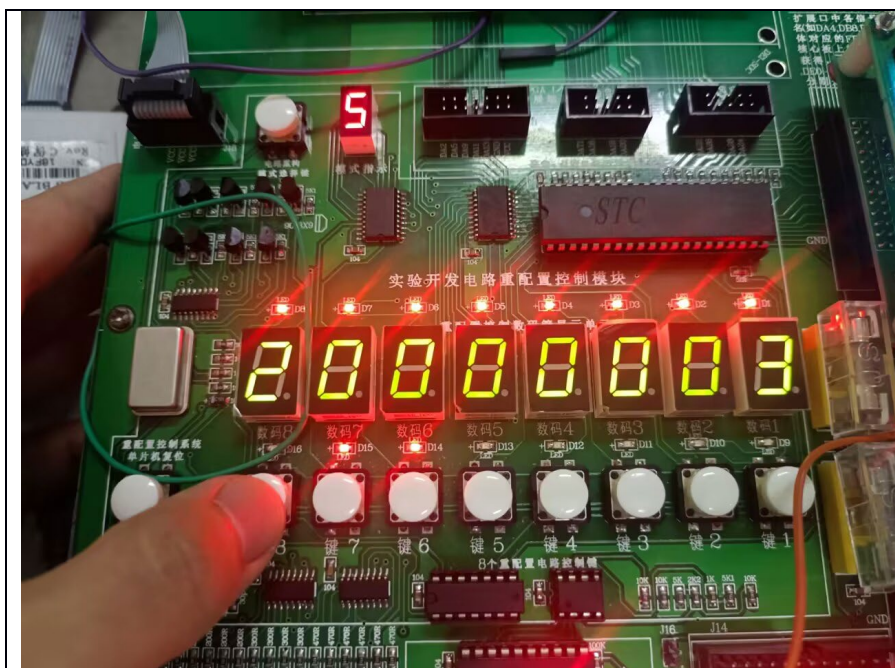
(3) 原理图编译、适配和下载：在 Quartus II 环境中选择 EP4CE6/10 器件，进行原理图的编译和适配，无误后完成下载。

(4) 功能测试：利用输入开关及发光二极管 LD 测试逻辑运算部件的功能并记录测试结果。我们初始化 ROM 如图,将地址处的值设置为该地址

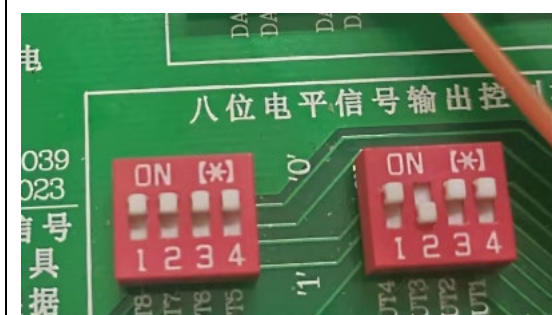


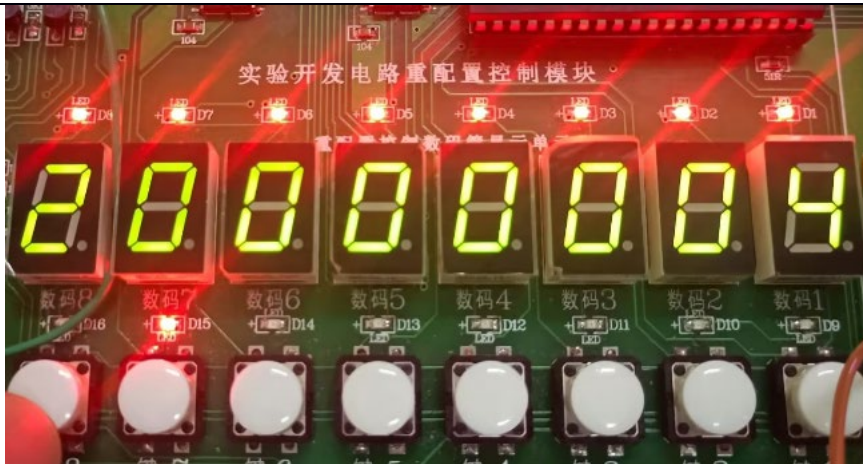
我们先进行 $PC = PC + 1$ 的测试, 将 LOAD 置为 1, CLR 置为 1, 不断按动脉冲键, 发现读出的值不断增加, 为预期的结果



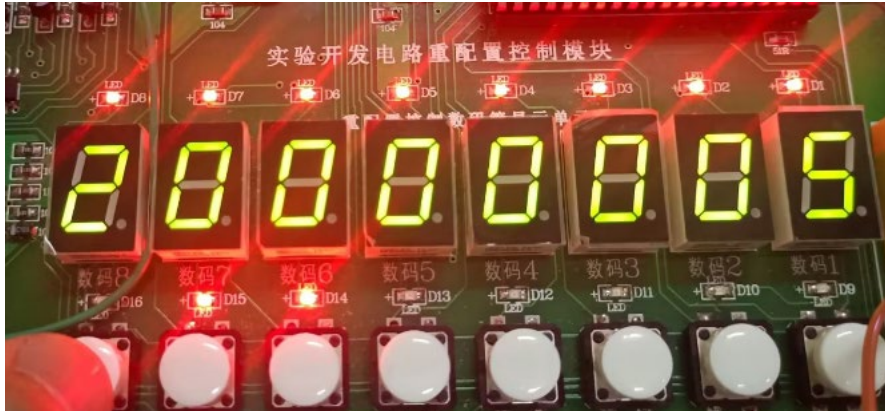


接下来我们测试初始化地址为 4 时, LOAD 置为 0, 预期的输出为地址 4 对应的值 4, 输出正确.





然后将 LOAD 置为 1, 继续进行自增操作, 发现变为了 5, 为预期结果, 成功!



(5) 生成元件符号。

仿真结果：

本次实验采取实际演示效果更明显, 因此不做仿真演示.

结论分析与体会：

通过实验，我深入理解了微程序控制器的工作原理。微程序控制器由控制存储器（CROM）、微程序计数器（PC）和微指令寄存器（IR）组成，能够实现复杂的控制逻辑。每次按下脉冲键产生的负脉冲，分别用于控制存储器的读取命令、微指令寄存器的加载和微程序计数器的递增。这个过程中，我学会了如何设计 8 位微程序计数器，包含加 1 和清除功能，并且明白了 CLR、LOAD 等控制信号的作用。

此外，通过将微程序控制器的各个部分连线并进行引脚锁定，我实际动手操作，掌握了硬件电路的连接与调试技巧。使用单脉冲驱动计数器、控制存储器读取、微指令寄存器加载等操作，使我更加理解了控制信号在微程序控制器中的作用。

在控制存储器初始化和微指令读取过程中，我学会了如何从 ROM 中写入和读出微指令代码，并通过与预先写入的代码进行比对，验证了读出过程的正确性。这部分实验强化了我对存储器和寄存器操作的理解，提升了调试和故障排除的能力。

总体而言，本次实验不仅增强了我对微程序控制器的理论理解，还通过实际操作提升了我的实践能力和问题解决能力。

期间遇到的一些问题如下:

1. 如何初始化 ROM?

答: 在创建的时候选择对应的初始化 hex 文件, 而不能中途修改的时候重新选择.

2. LOAD 的作用?

答: LOAD 低电平有效, 为 1 的时候代表不进行初始化地址数据的读入, 进行自增操作, 为 0 的时候代表进行初始化地址数据的读入.