

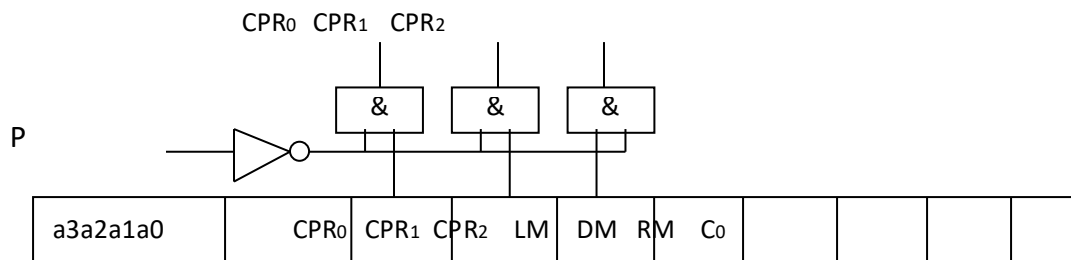
计算机组成与设计

课程实验报告

学号：202200400053		姓名：王宇涵	班级：2202
实验题目：综合实验			
实验学时：2		实验日期：2024-06-20	
实验目的： 锻炼综合电路设计的能力			
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：			
CPU 综合实验电路包括运算器电路和控制器电路。图 14-1 给出了 CPU 综合实验结构框图。			
运算器由三个寄存器 R0、R1、R2、移位器、加法器等构成，并组装在一起构成 ALU 算术逻辑运算部件，参照四位补码运算器电路框图所示。			
(1) 调用 ALU 模块、μPC 模块及门电路按 CPU 综合实验结构框图完成连线。			
(2) 管脚定义：实验平台工作于模式 5，ALU 的输入数据 a3-a0 依次锁定在μIR23-μIR420 上，CPR0、CPR1、CPR2 依次锁定在μIR7-μIR5 上，LM、DM、RM、C0 依次锁定在μIR4-μIR1 上，P 锁定在键 8 上。Q3-Q0 依次锁定在 D4-D1 上。			
<div></div>			
图 14-1 CPU 综合实验结构框图			
(3) 适配、下载			

(4) 编制微程序

微指令可确定如下格式：



$\mu IR_{23} \sim \mu IR_{20}$ μIR_7 μIR_6 μIR_5 μIR_4 μIR_0

将微指令格式分为两部分：前面部分 $\mu IR_{23} \sim \mu IR_{20}$ 可设置数据，后面部分 $\mu IR_7 \sim \mu IR_0$ 可确定微命令，例：需要 CPR0 脉冲，该位为 1，否则为 0；备用位填 0。

例题：编写一个 $0110 + 1000$ 的微程序。

寄存器分配：0110 送 R0、1000 送 R1、结果送 R2。

操作步骤 微指令 说明

0 1 1 0 \rightarrow R0; 60 00 80H 存入控制存储器 ROM 的 0 单元。

↓

1 0 0 0 \rightarrow R1; 80 00 40H 存入控制存储器 ROM 的 1 单元。

↓

$R0 + R1 \rightarrow R2$; 00 00 08H 存入控制存储器 ROM 的 2 单元。

(5) 功能检查

按 CPU 复位键清 μPC ，使之指向控制存储器的 0 号单元。

每按一次单脉冲键，便执行一条微指令。

按第 3 次单脉冲键，微程序执行完毕。

运算结果应存放在 R2 中，并用 LD3-0 指示。

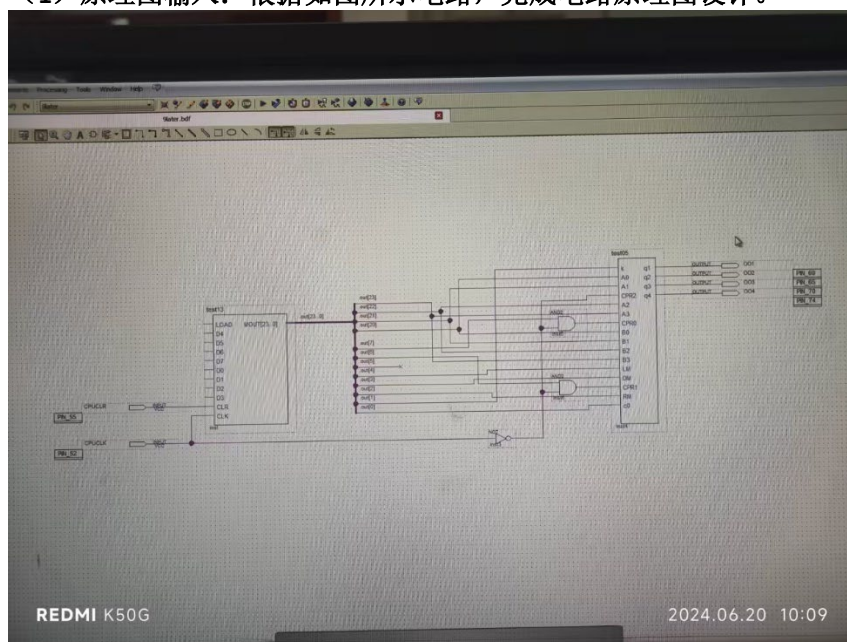
同学可编制多种多样的微程序来完成各种操作。

例 $2 \times (R0 + R1) \Rightarrow R2$;

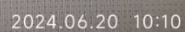
$1/2 (R0 + R1) \Rightarrow R2$;

实验步骤：

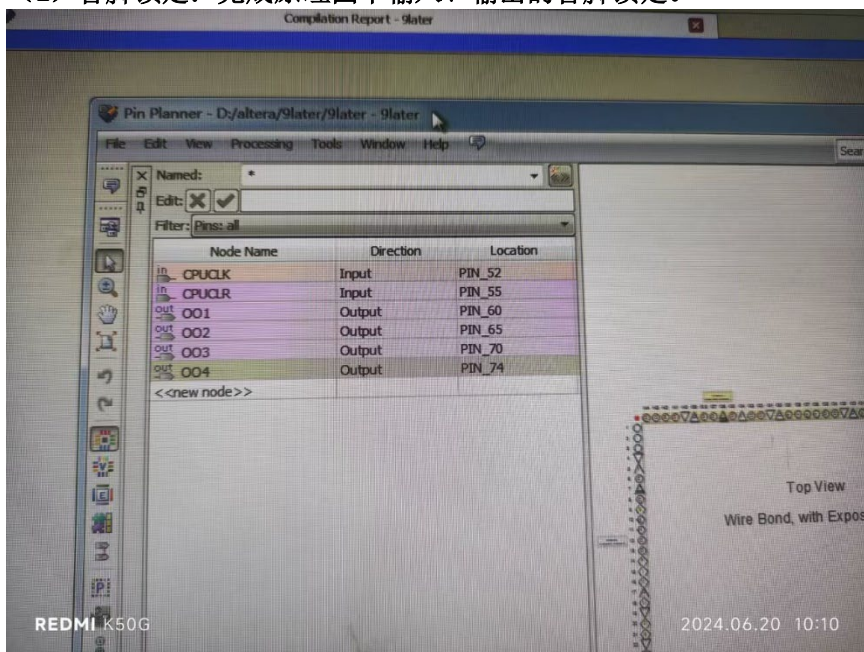
(1) 原理图输入：根据如图所示电路，完成电路原理图设计。



各个组件的具体电路图如下



(2) 管脚锁定：完成原理图中输入、输出的管脚锁定。



(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 测试功能并记录测试结果。

指令序列如图：

前五条指令分别为 600080H, 800040H, 000008H, 000010H, 000004H

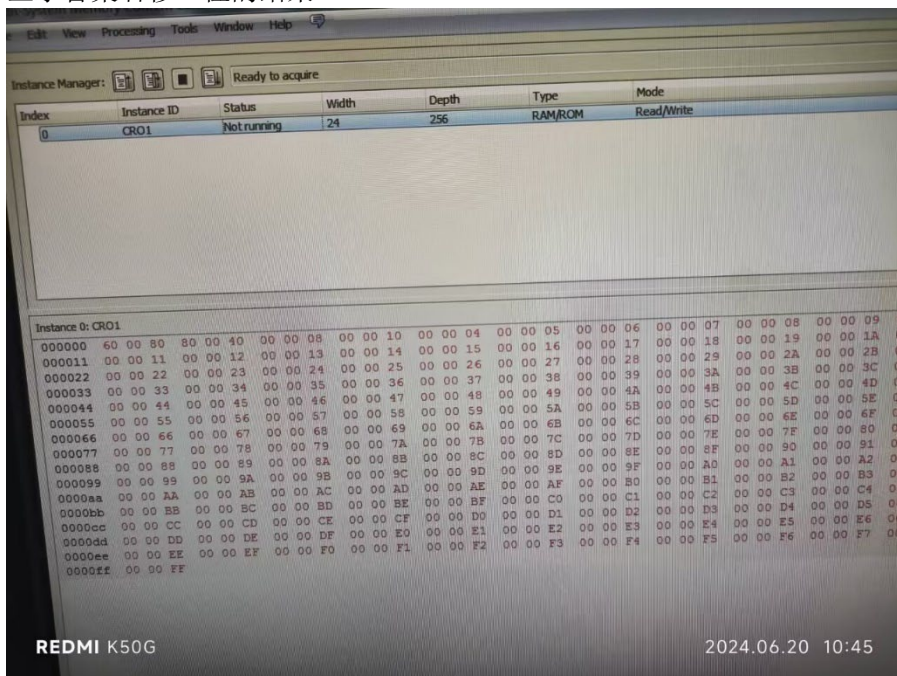
分别实现功能为：0 1 1 0→R0 将数字 6 存入控制存储器 ROM 的 0 单元；

1 0 0 0→R1 数字 8 存入控制存储器 ROM 的 1 单元；

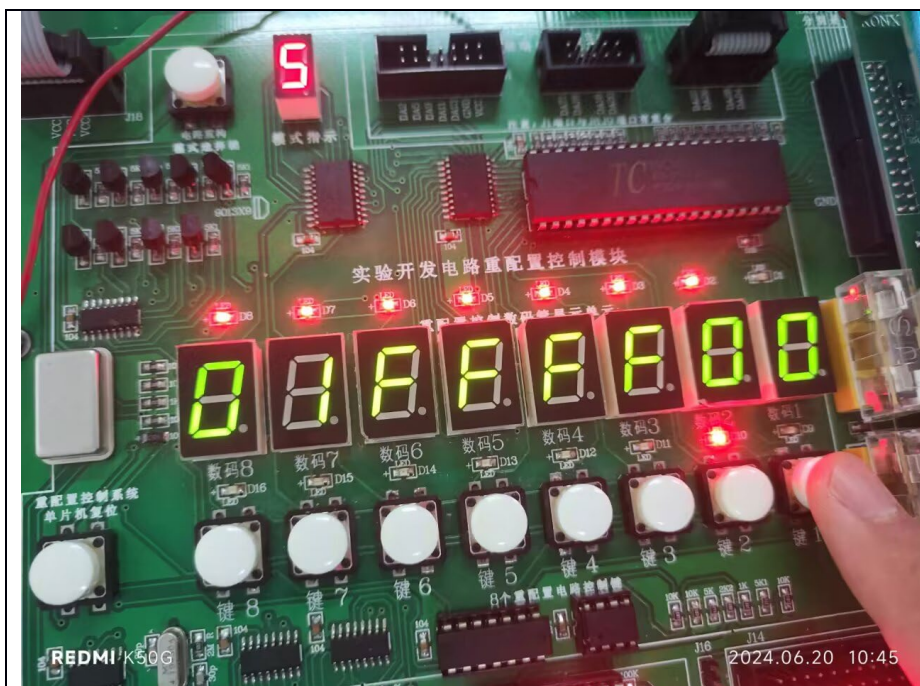
R0+R1→R2 将加法的计算结果存入控制存储器 ROM 的 2 单元并显示；

显示答案左移一位的结果；

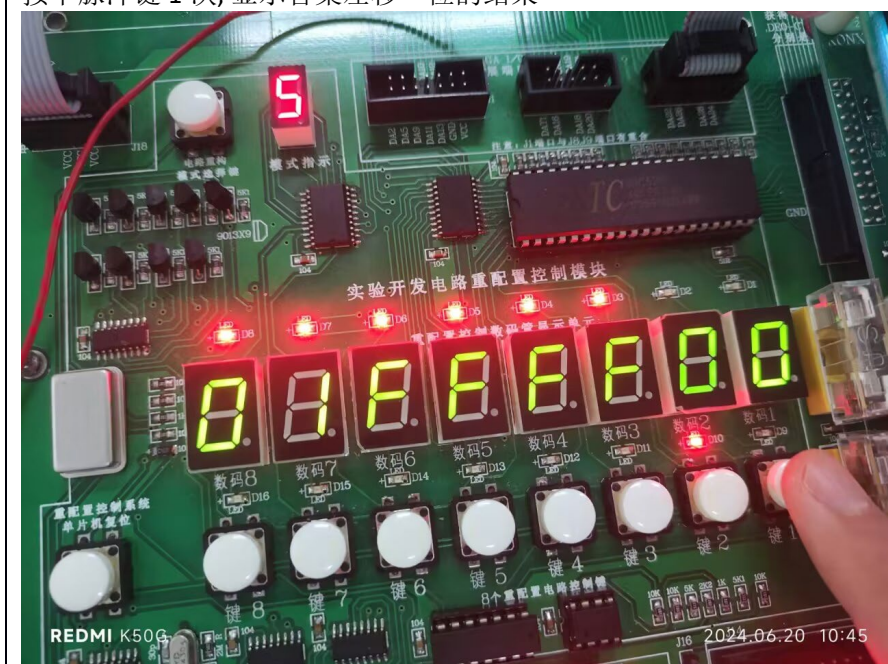
显示答案右移一位的结果



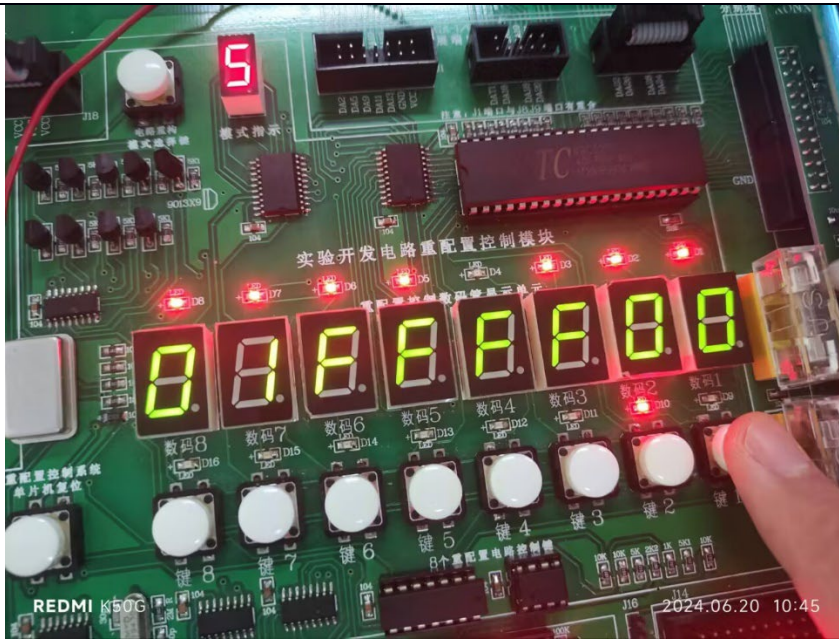
点击复位按钮，按下脉冲键 3 次，前三条指令执行完毕，R0+R1→R2 将加法的计算结果存入控制存储器 ROM 的 2 单元并显示



按下脉冲键 1 次,显示答案左移一位的结果



按下脉冲键 1 次,显示答案右移一位的结果



(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：

在完成了本学期最后一个计算机组成原理实验后，我对 CPU 内部结构有了更深刻的理解。通过构建和连接运算器（ALU）和控制器模块，我进一步理解了 CPU 的基本工作原理和内部结构，特别是寄存器的使用和微指令的执行过程，让我对 CPU 如何实现复杂的指令操作有了直观的认识。

我实践了寄存器与 ALU 的交互。实验中涉及将数据存入寄存器（R0、R1、R2）并利用 ALU 进行算术运算。通过实际操作，我了解到如何通过微指令控制数据流动和运算过程，以及寄存器在计算中的重要作用。

我掌握了微指令的编写与执行。实验要求编写微指令来实现特定的运算操作，例如将数据装载到寄存器中以及进行加法运算。这一过程让我理解了微程序控制器的设计思想以及微指令的格式和编写方法。

我体验了微程序控制器的功能。通过实验中的功能检查步骤，我熟悉了如何利用微程序控制器实现顺序控制和操作控制，这对于理解微程序控制器在实现复杂指令集中的作用非常有帮助。

通过这次实验，我不仅加深了对计算机组成原理的理解，还提升了实际操作和解决问题的能力，为以后的学习和工作奠定了坚实的基础。

期间也遇到了一些问题：

1. 如何理解实验的原理？

答：实验将以往的很多实验结合在了一起，具有较强的综合性，实现了按 PC 计数从内存中读取指令并顺序执行的功能，并将所需的结果进行输出。

2. 如何在不创建新的 ROM 情况下手动初始化已有 ROM 数据？

答：使用 In-system 的 tool 元件，可以直接进行暂时的修改，点击 write 按钮即可修改成功。但当重新打开 In-system 器件时数据仍未修改，需要再次手动修改。