**西安电子科技大学**

**计算机组成与体系结构 课程实验报告**

**实验名称 计组实验**

计算机科学与技术学院 2203013 班

成 绩

姓名 赵宇阳 学号 22009200439

同作者 无

实验地点 **E-Ⅱ-311** 实验批次 **15 10**

|  |
| --- |
| 指导教师评语：  指导教师：  年 月 日 |
| **实验报告内容基本要求及参考格式**  一、实验目的  二、实验所用仪器（或实验环境）  三、实验基本原理及步骤（或方案设计及理论计算）  四、实验数据记录（或仿真及软件设计）   1. 实验结果分析及回答问题（或测试环境及测试结果） |

# 一．存储器实验

**一、实验目的**

1、掌握 FPGA 中 lpm\_ROM 的设置，作为只读存储器 ROM 的工作特性和配置 方法；

2、用文本编辑器编辑 mif 文件配置 ROM，学习将程序代码以 mif 格式文件加载于 lpm\_ROM 中。

**二、实验环境**

QuartusII 9.0 64bit，计组综合实验箱

**三、实验基本原理及步骤**

ALTERA 的 FPGA 中有许多可调用的 LPM (Library Parameterized Modules)参数

化的模块库，可构成如 lpm\_rom、lpm\_ram\_io、lpm\_fifo、lpm\_ram\_dq 的存储器

结构。CPU 中的重要部件，如 RAM、ROM 可直接调用他们构成，因此在 FPGA 中利

用嵌入式阵列块 EAB 可以构成各种结构的存储器，lpm\_ROM 是其中的一种。

lpm\_ROM 有 5 组信号：地址信号 address[ ]、数据信号 q[ ]、时钟信号

inclock、outclock、允许信号 memenable，其参数都是可以设定的。由于 ROM 是

只读存储器，所以它的数据口是单向的输出端口，ROM 中的数据是在对 FPGA 现场

配置时，通过配置文件一起写入存储单元的。

实验步骤：

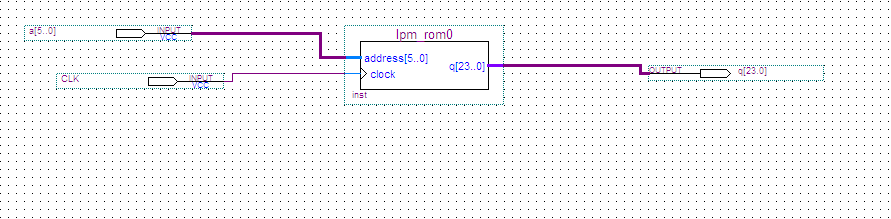
1.使用 lpm\_rom 元件绘制原理图，设置地址总线宽度为 6，数据总线宽度为 8；

2.编辑 ROM 初始化配置文件（.mif）；

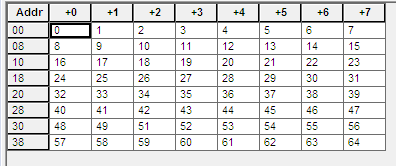
3.波形仿真并下载到实验箱验证。

**四、实验数据记录**

（1）原理图

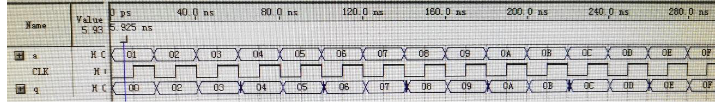


（2）mif表格



五、实验结果分析

波形仿真结果



经检验，结果正确

将程序下载到实验箱上进行验证，选择实验电路模式仍为 NO.0。其中 8 位数据

输出由数码管 3，4 显示。6 位地址由键 2、键 1 输入，键 1 负责低 4 位，地

址锁存时钟 CLK 由键 8 控制，每一次上升沿，将地址锁入。数码管 3，4 将显

示 ROM 中输出的数据。发光管 8 至 1 显示输入的 6 位地址值。

改变输入地址，外加读脉冲，通过实验箱上的数码管比较读出的数据，可以发现

与 mif 文件中数据一致，故结果正确。

**六、实验心得**

通过这次实验我学会了使用 mif 文件配置文件，并在验证的过程中熟悉了实验箱

的使用方法，增加了动手实践能力。明白了存储器的原理，可以通过试验箱读取存在文件里面的数据。

# 二．运算器实验

**一、实验目的**

1.了解简单运算器的数据传输通路；

2.掌握算术逻辑运算的工作原理。

**二、实验环境**

QuartusII 9.0 64bit，计组综合实验箱

**三、实验基本原理及步骤**

算术逻辑单元运算器 ALU181 根据 74LS181 的功能，用 VHDL 硬件描述语言编辑而

成，构成 8 位字长的 ALU。参加运算的两个 8 位数据分别为 A[7..0]和 B[7..0]，

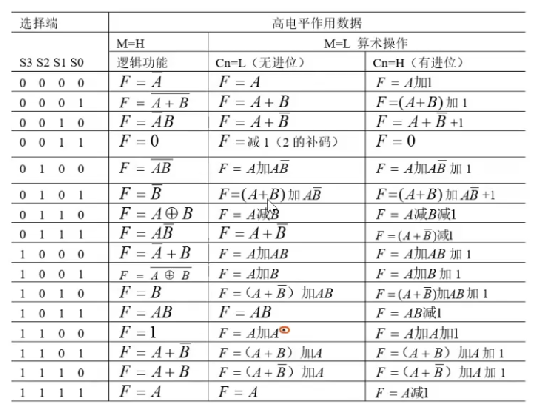
运算模式由 S[3..0]的 16 种组合决定，S[3..0]的值由 4 位 2 进制计数器

LPM\_COUNTER 产生，计数时钟是 Sclk；此外，设 M=0，选择算术运算，M=1 为逻辑

运算，CN 为低位的进位位；F[7..0]为输出结果，CO 为运算后的输出进位位。两个

8 位数据由总线 IN[7..0]分别通过两个电平锁存器 74373 锁入，ALU 功能如表所

示。



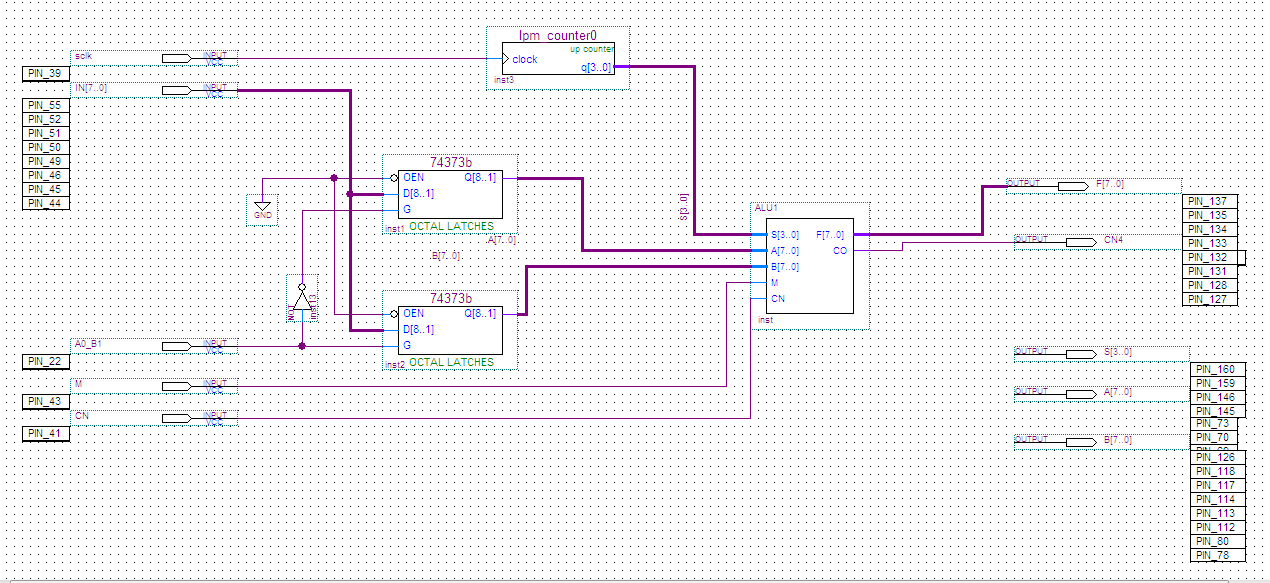
实验步骤： 1.用ALU181.vhd（已编写好）创建alu元件；

2.用alu 元件、锁存器等绘制原理图，可用lpm\_counter元件的输出作为S3..0

3.波形仿真并下载到实验箱验证。

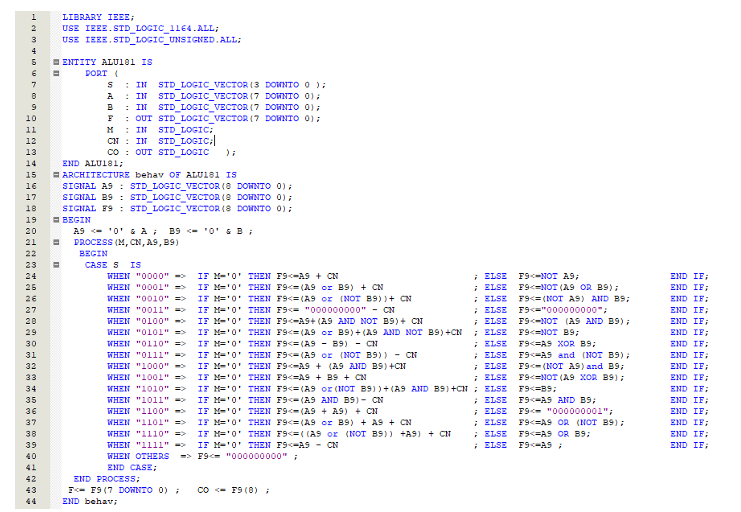
**四、实验数据记录**

**（1）原理图**



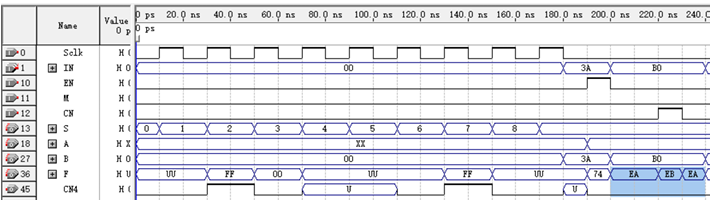
LPM\_COUNTER 输出 0 到F计数信号，作为ALU的模式选择信号；74LS373作 为操作数A的寄存器，通过控制寄存器的使能将数据输入端的数据分别送到A、B 端口，当使能端有效时，数据通过寄存器进入A端口，当使能端无效时，寄存器 输出端即A端保持数据，改变数据输入端的值，该数据直接传输到B端口，ALU 即输出运算结果。

（2）ALU181 源

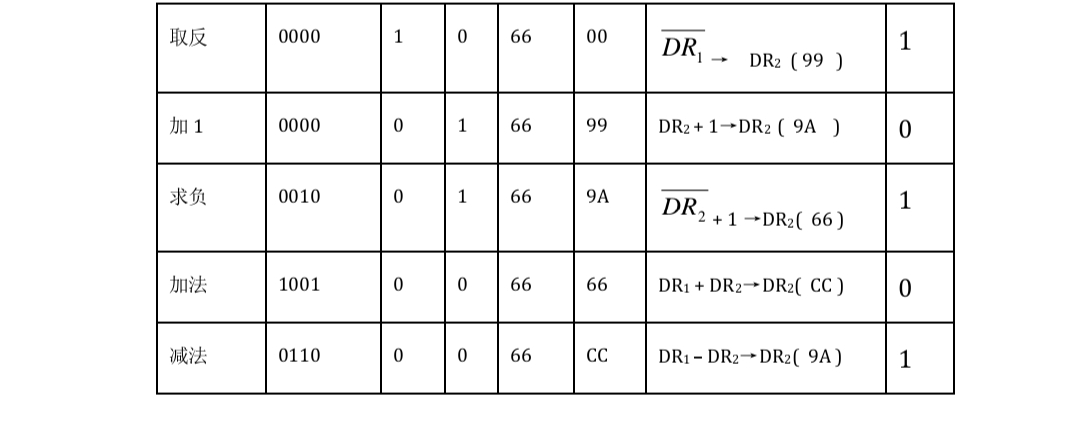
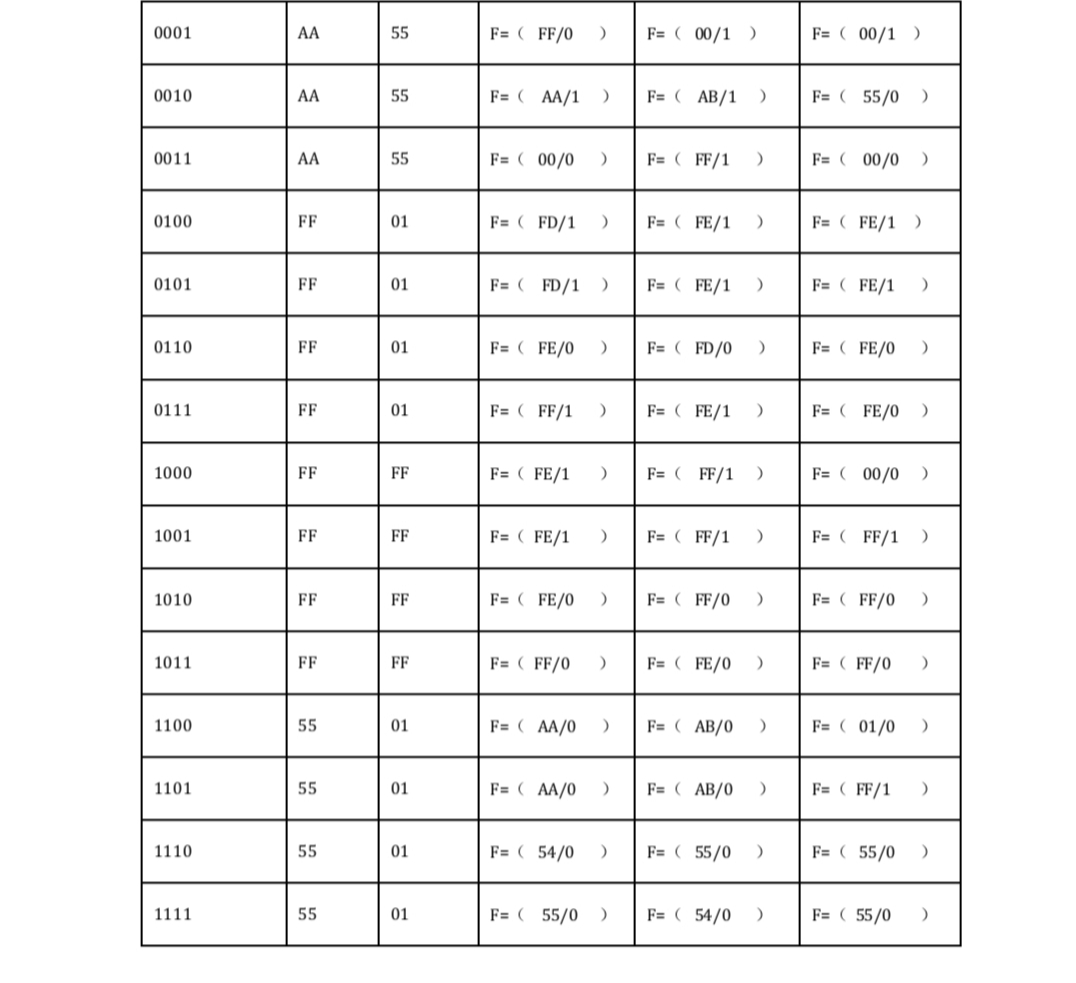


五、实验结果分析

波形仿真结果（节选）



该波形模拟了使用该ALU执行算术加法运算的过程。先在Sclk端输入9个计数 脉冲，M置0，使ALU工作在算术加法状态，IN端输入3A并给一个使能脉冲，将 3A 锁存至A端，改变IN端的值为B0，B0直通至B端，ALU的输出端F即输出算 术A+B的结果EA。此时再将CN（低位进位）置1，可以看到结果F也加1变成 EB。



**六、实验心得**

通过本次实验，对ALU的工作原理有了更深入的了解。 ALU 输入输出方式有多种。本例仅有一组8位输入和一组8位输出，故采用了单寄存器的方式输入，先将两个操作数之一锁存至ALU一输入端，再直接将另一操作数送至ALU另一输入端，即完成操作数的输入。模式有很多种，该实验有16种，选择不同对应的模式下，操作的方法不一样，得到的结果也不一样，所以需要查表运算。

三．节拍脉冲发生器时序电路实验

**一、实验目的**

1.掌握节拍脉冲发生器的设计方法和工作原理；

2.理解节拍脉冲发生器的工作原理。

**二、实验环境**

QuartusⅡ 9.0 64bit，计组综合实验箱

**三、 实验基本原理及步骤**

计算机之所以能够按照人们事先规定的顺序进行一系列的操作或运算，就是因为 它的控制部分能够按一定的先后顺序正确地发出一系列相应的控制信号。这就要求计算机必须有时序电路。控制信号就是根据时序信号产生的。本实验说明时序电路中节拍脉冲发生器的工作原理。

1.连续节拍发生电路 可由 4 个 D 触发器组成，产生 4 个等间隔的时序信号 T1-T4，其中 CLK1 为 时钟信号。当 RST1 为低电平时，T1 输出为“1”，而 T2、T3、T4 输出为 “0”；当 RST1 由低电平变为高电平后，T1-T4 将在 CLK1 的输入脉冲作用下， 周期性地轮流输出正脉冲。

2.单步节拍发生电路 原理图与连续节拍类似，其中 CLK1 为时钟信号。当 RST1 为低电平时，T1-T4 输出均为“0”；当 RST1 由低电平变为高电平后，T1-T4 将在 CLK1 的输入脉冲 作用下，仅输出组正脉冲。

3.单步/连续节拍发生电路 增加两个 2-1 多路选择器，S 是单步或连续节拍发生控制信号，当 S=0，选择 单步运行方式；当 S=1，选择连续运行方式。

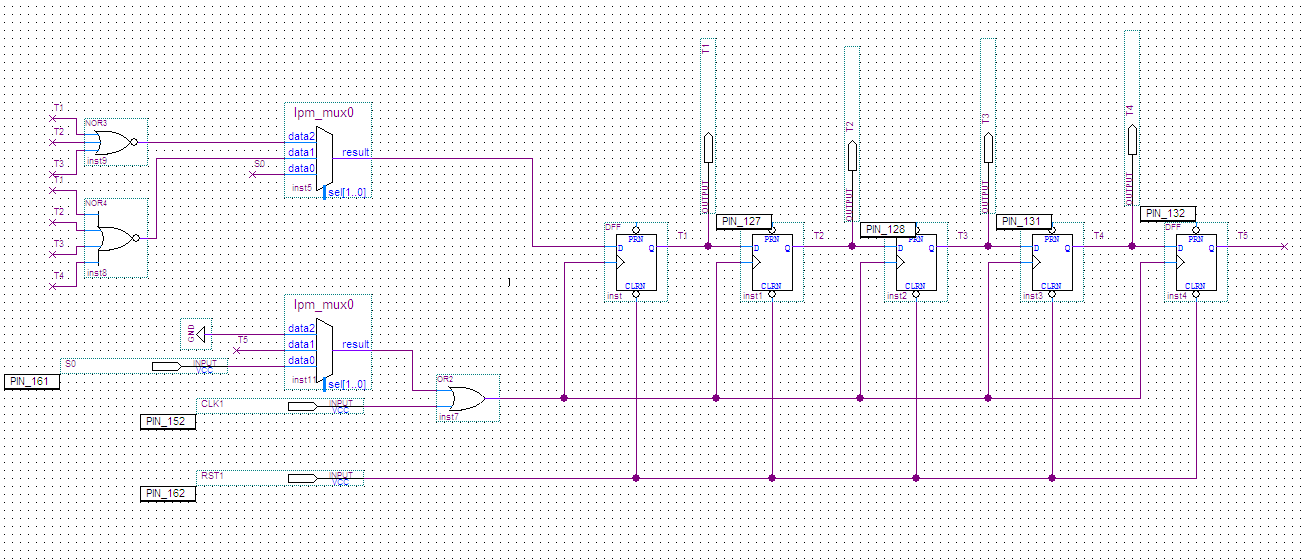
步骤： 1.画出原理图；

2.进行波形仿真，并选择其中的任意一个下载到实验箱进行验证。

**四、 实验数据记录**

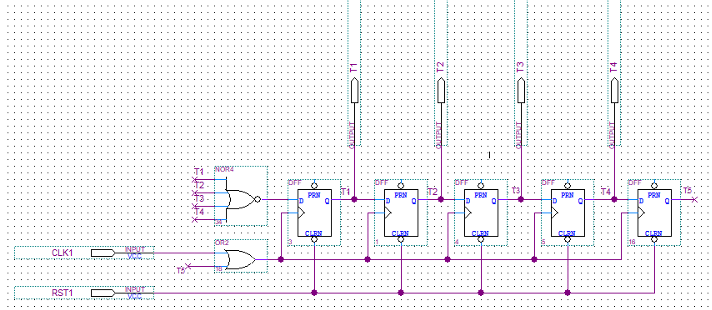
原理图如下

1. 连续节拍发生器



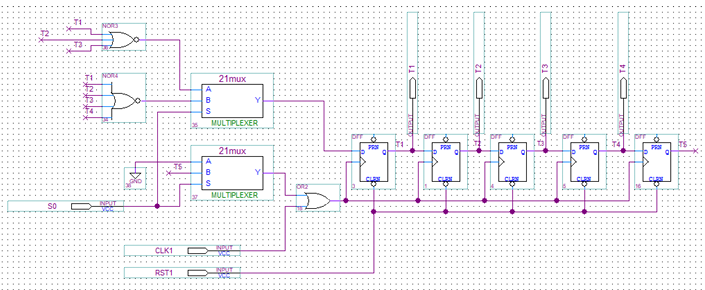
当T1-T3 都为0，T4为1时，一次节拍即将结束，或非门输出1送到第一个D触 发器输入端，作为下一轮节拍的启动。

（2）单步节拍发生器



当T1-T4 都为0，T5为1时节拍结束。T5使所有D触发器的时钟信号不变，触发 器状态不变，保持0 0 0 0 1。4输入或非门已将启动信号准备好，直到异步清 零信号到来，T5置0，触发器正常接收时钟信号，产生下一轮节拍。

（3）连续/单步节拍发生器

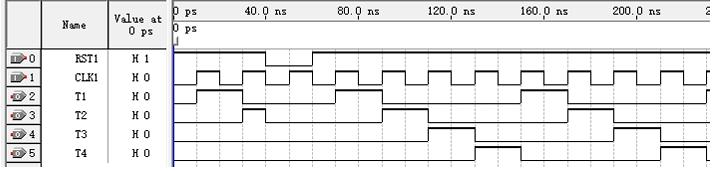


此为可切换连续/单步的节拍发生器。分别观察连续和单步的图，可以发现两种 节拍器可以设计成仅在输入端有所不同，故将两种融合，增加S端控制两种节拍 的切换。这里S为0时是单步节拍，S为1时是连续节拍。

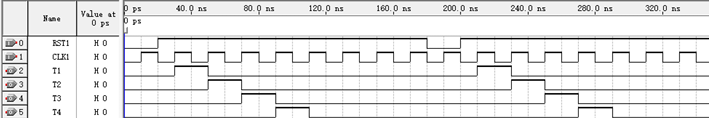
**五、实验结果分析**

波形仿真结果（节选）

（1）连续节拍



（2）单步节拍



之后选择一个原理图分配引脚并下载到实验箱上，通过发光管观察是否能产生连 续的脉冲信号，验证正确。



**六、实验心得**

该实验非常巧妙地使用了或非门作为节拍的启动器，并且根据原理的相似性设计了可切换的功能，而且可切换功能部分还可以再进一步改进以减少元件数量。实验原理图有一定的思维难度，光是读懂都花了一些时间，但是整体的思路还是比较清晰明确的。

**四：程序计数器** PC **和地址寄存器** AR

4.1 实验目的

（1）掌握地址单元的工作原理。

（2）掌握程序计数器 PC 的两种工作方式，加1计数和重装计数器初值

的实现方法；

（3）掌握地址寄存器从程序计数器获得数据和从内部总线获得数据的实

现方法。

4.2实验环境

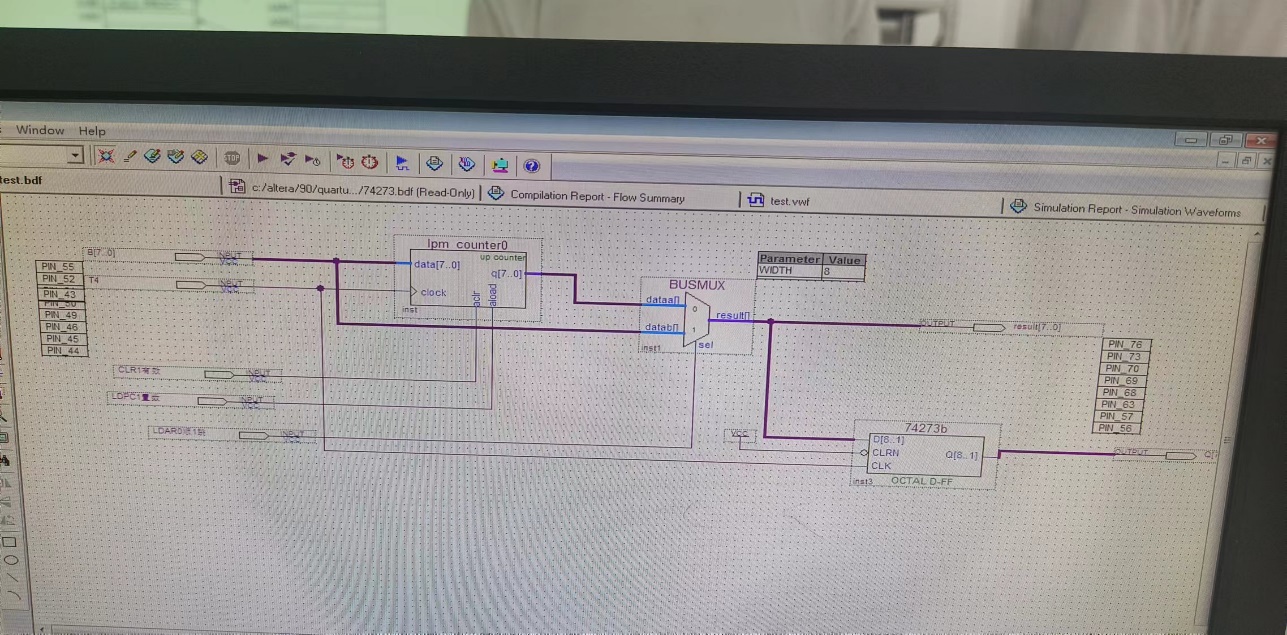
Quartus Ⅱ 开发集成环境，FPGA 硬件平台。

4.3 实验原理和实验步骤

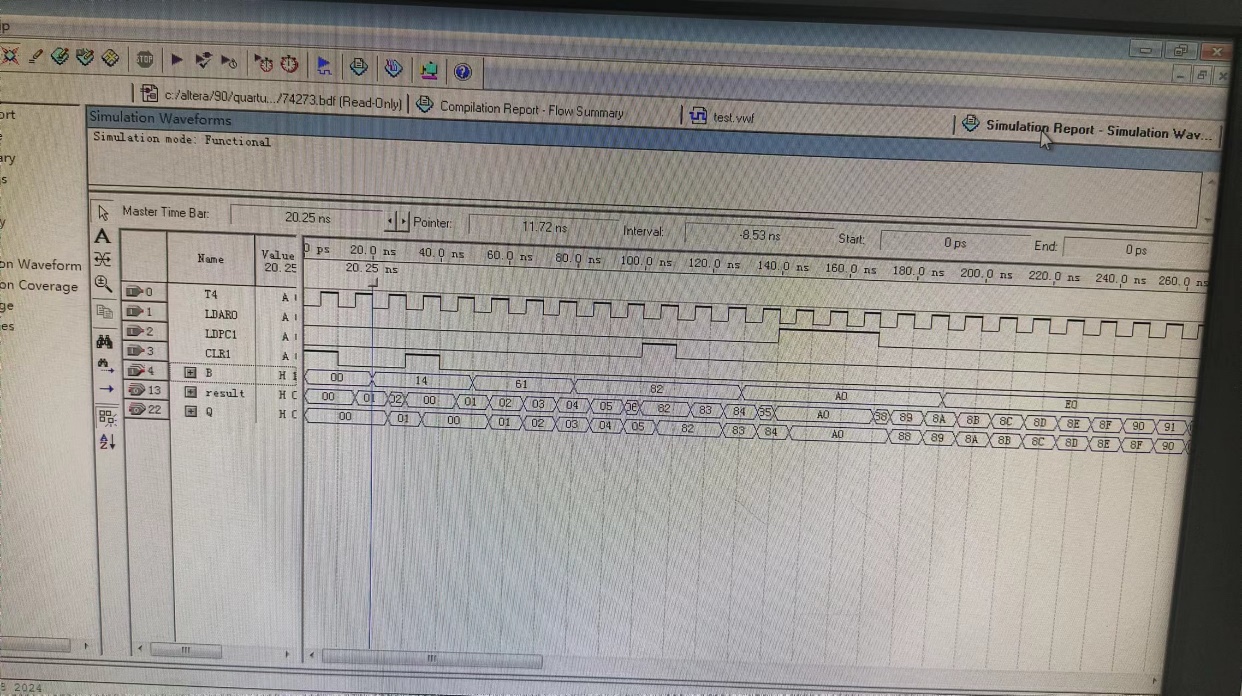
地址单元主要由三部分组成：程序计数器、地址寄存器和多路开关。

程序计数器 PC 用以指出下一条指令在主存中的存放地址，CPU 正是根 据 PC 的内容去存取指令的。因程序中指令是顺序执行的，所以 PC 有自增功能。程序计数器提供下一条程序指令的地址，在 T4 时钟脉冲的作用下具有自动加 1 的功能；在 LDPC 信号的作用下可以预置计数器的初值（如子程序调用或中断响 应等）。当 LDPC 为高电平时，计数器装入 data[ ]端输入的数据。aclr 是计数器的清 0 端，高电平有效（高电平清零）；aclr 为低电平时，允许计数器正常计数。

地址寄存器 AR（74273）锁存访问内存 SRAM 的地址。273 中的地址来自两个渠道。一 是程序计数器 PC 的输出，通常是下一条指令的地址；二是来自于内部数据总线的数据， 通常是被访问操作数的地址。 为了实现对两路输入数据的切换，在 FPGA 的内部通过总线多路开关 BUSMUX 进行选择。 LDAR 与多路选择器的 sel 相连，当 LDAR 为低电平，选择程序计数器的输出；当 LDAR 为高电平时，选择内部数据总线的数据。



4.4 实验数据记录



4.5 实验结果分析

T=0时，CLR=1清零，PC开始从00计数。正常开始计数，中途又清零一

次，重新计数到5时，LDPC1有效，PC置数到82，并从82开始依次计数。计数到85时发生跳转（即LDAR有效），不过PC仍然在时钟脉冲下从85继续置数，这是电路设计的局限性，即，使用了同一个时钟脉冲的原因，所以在跳转结束后，PC值为88，并从88开始依次计数。

**4.6实验心得：**

通过该实验了解到了cpu内部PC和AR寄存器的功能作用，PC的自增加一功能，通过选择器可以选择输出是从PC还是从AR，对cpu的掌握更加熟练了。

**五：总线控制实验**

5.1 实验目的

1．理解总线的概念及特性。

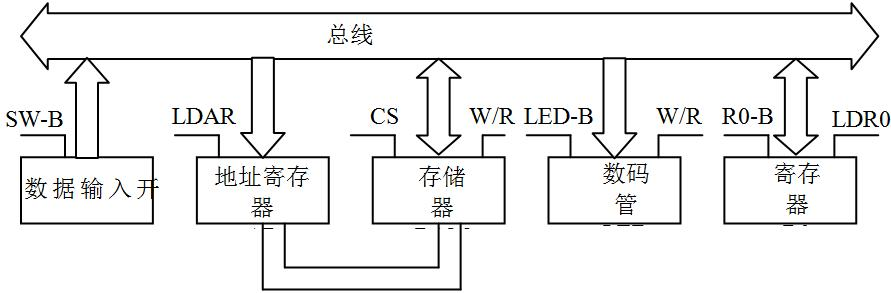
2．掌握总线传输控制特性。

5.2实验环境：

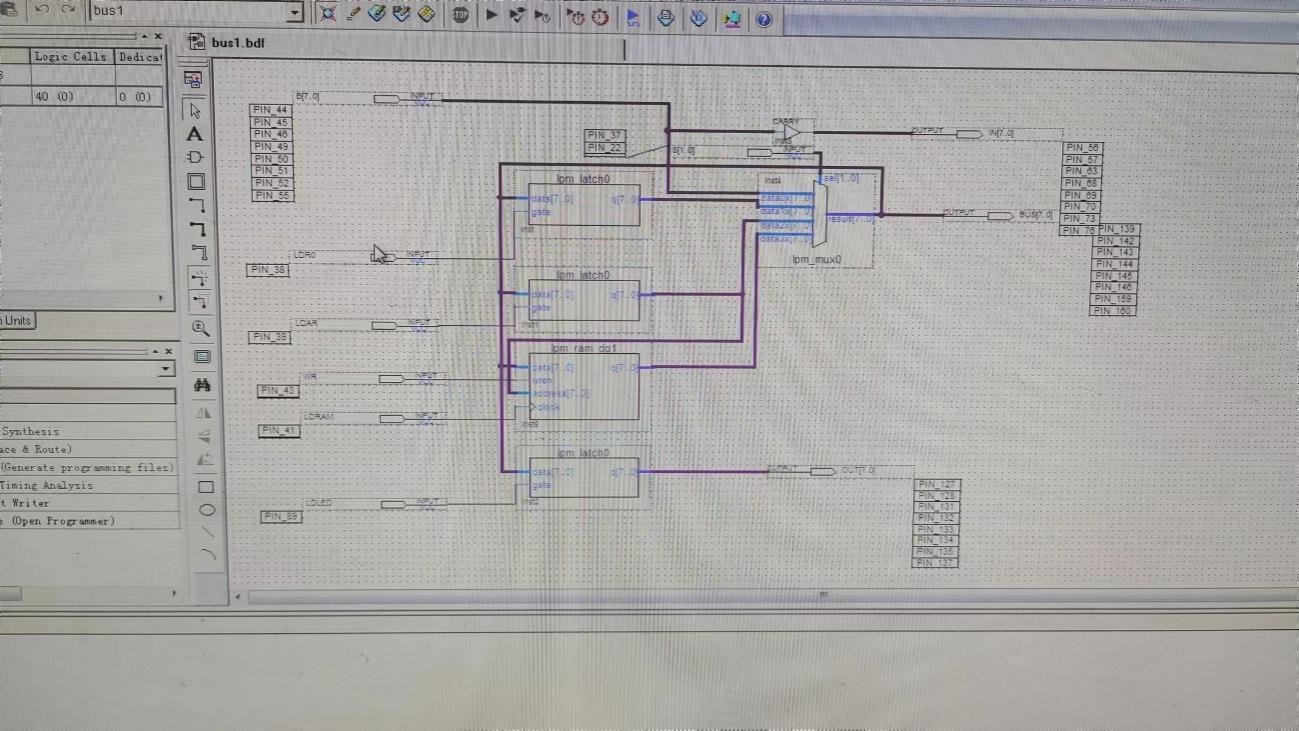
Quartus Ⅱ 开发集成环境，FPGA 硬件平台

5.3 实验原理和实验步骤

总线是多个系统部件之间进行数据传输的公共通路，是构成计算机系统的骨架。借助总线连接，计算机在系统各部件之间实现传送地址、数据和控制信息的操作。所谓总线就是指能为多个功能部件服务的一组公用信息线。



总线控制实验线路图



实验步骤（实验箱）：

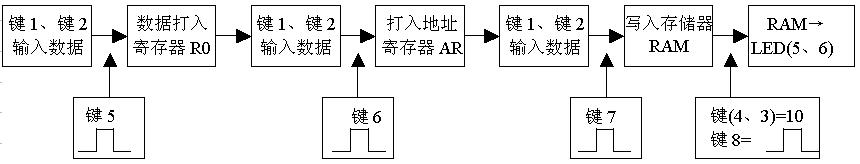
根据挂在总线上的几个基本部件，设计一个简单的流程。

1.输入设备将数据打入寄存器 R0。

2.输入设备将另一个数据打入地址寄存器 AR。

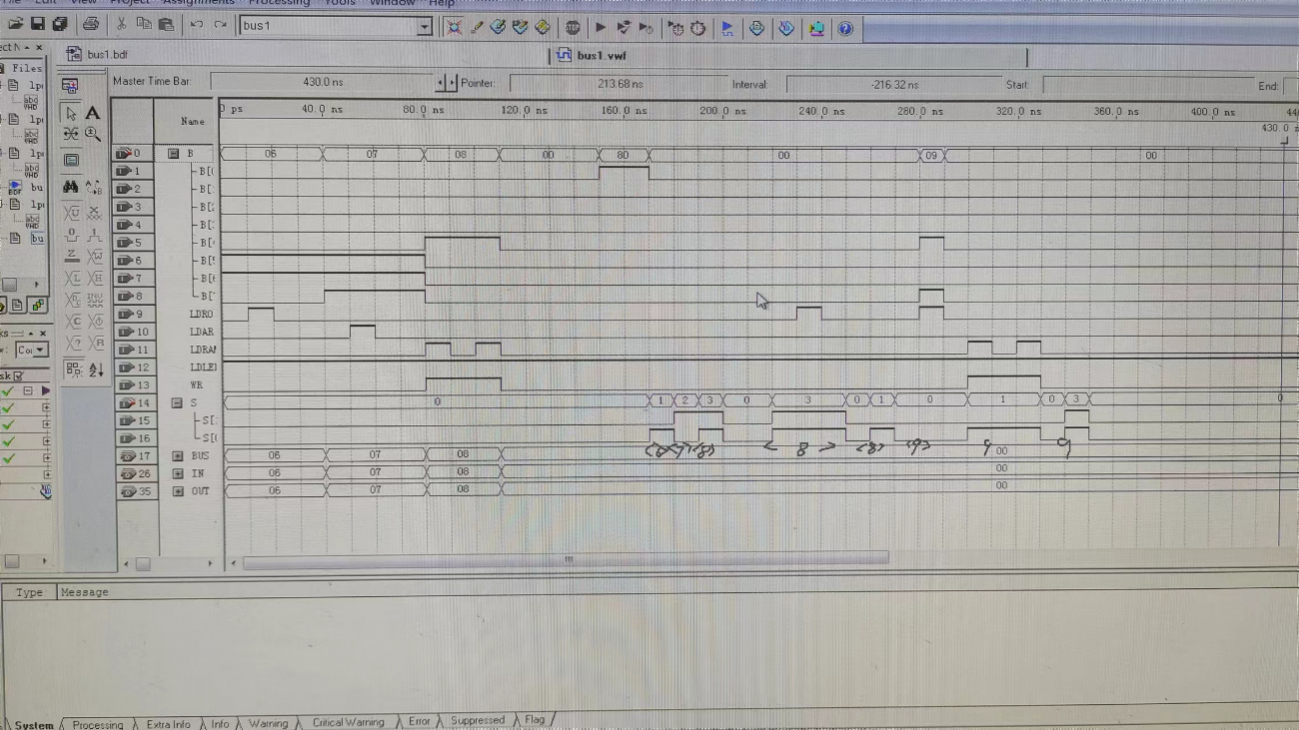
3.将寄存器 R0 中的数据写到当前地址的存储器中。

4.将当前地址的存储器中的数用数码管显示。



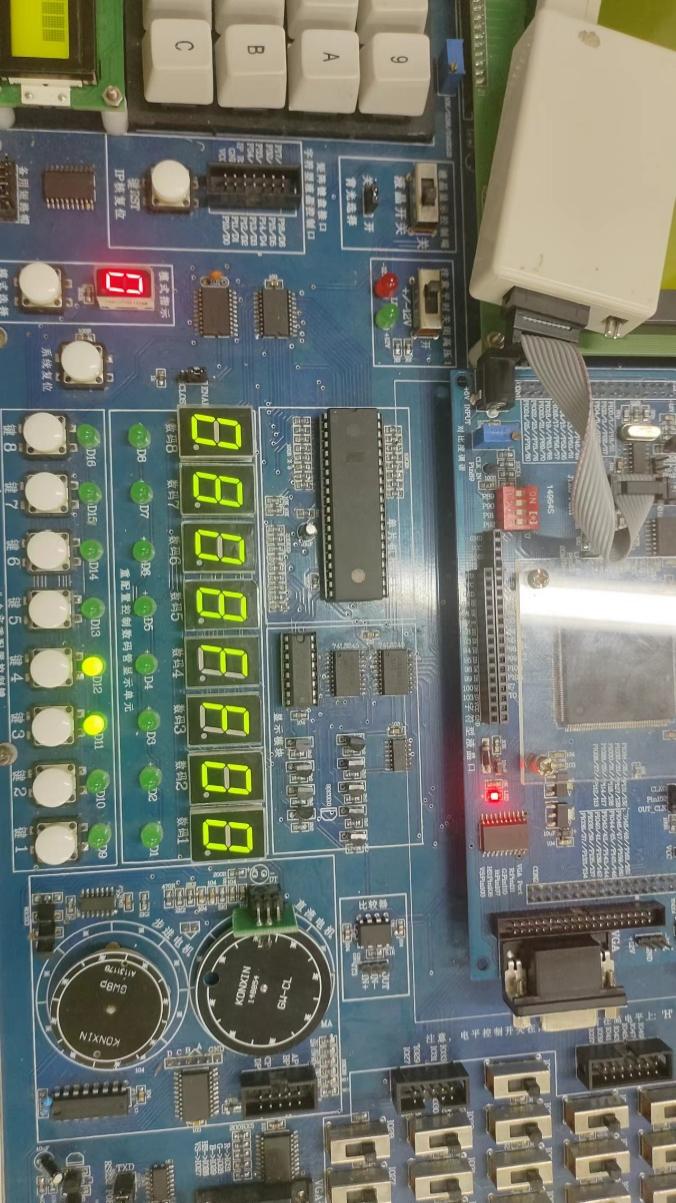
|  |  |  |
| --- | --- | --- |
| 练习题编号 | 功 能 | 助 记 符 |
| 1  2  3  4  5  6  7 | 把INPUT设置的数据写入R0  把INPUT设置的数据写入RAM某单元  把RAM某单元内容读入R0  把R0内容读入RAM某单元  把R0内容输出显示  把RAM某单元内容输出显示  把INPUT设置的数据直接输出显示 | IN R0，KEY  IN RAM，KEY  LD R0，RAM  ST RAM，R0  OUT LED，R0  OUT LED，RAM  OUT LED，KEY |

5.4 实验数据记录



未保存完整仿真图，总线中数据如标注所示

5.5 实验结果分析



S=0,分别使LDR0、LDAR、LDRAM与WR为高电平，以此分别向寄存器R0，AR与RAM中存入数据06、07、08。我们可以通过分别使S的值为1、2、3来使其数据输出到总线上显示出来。接着我们使S=3，并且使LDR0为高电平，这样R0会读取RAM中的数据，即R0中现在的数据变为了8，接着我们将R0中的数据改为9,再使S=1，将R0中数据输出到总线，使LDRAM与WR为高电平，从而让RAM读取R0中的数据变为9。

5.6回答问题

1．如何向RAM中输入多个数据，并在输出设备OUTPUT上显示这些数据？（将

3个数据写入RAM的不同地址中，再将它们分别读出，在OUT上显示）

确定要写入的地址，通过使LDAR有效，确定地址后，使WR和LDRAM有

效，写入数据即可。读取数据时，使S=0，使总线数据为该数据的地址，使LDA有效，重新读取该地址，存入地址寄存器后，使S=3，LDRAM=1，此时总线数据为确定地址下的数据，在输出设备OUTPUT显示即可。

2．传输过程中是否会在总线上发生数据冲突？若发生冲突应怎样避免？

在传输过程中，总线上确实有可能发生数据冲突。通常发生在多个设备或

内存映射单元同时尝试在总线上发送数据的情况。

为避免冲突，可以通过总线仲裁和调度，通过设置优先级等方法来避

免冲突。在软件层面，可以使用协议和同步机制来协调各个设备的操作。方法有很多，总的思想是确定总线使用顺序，使数据传输有序。

**5.7心得体会**：通过本次实验，学会了总线的控制流程，明白了AR，PC，R0寄存器以及存储器rom是怎么通过总线进行数据的输入和输出，读和写。可以通过键位1到9来实现不同的功能，列入把总线上的数据存在R0里面，或者写入到存储器里面，又或者把R0里面的存放的数据写入到存储器ROM里面，更加了解到了cpu是处理数据时总线是怎么工作的。