

# 西安电子科技大学

## 计算机组织与体系结构实验 课程实验报告

实验名称 计算机组织与体系结构

计算机 学院 1603019 班

姓名 张俊华 学号 16030199025

同作者

实验日期 2018 年 6 月 9 日

实验地点 E-II-311 实验批次 第六批

成 绩

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）
- 五、实验结果分析及回答问题（或测试环境及测试结果）

## 实验一 存储器实验

### 一、实验目的

1. 掌握 FPGA 中 lpm\_ROM 只读存储器配置方法。
2. 用文本编辑器编辑 mif 文件配置 ROM，加载于 ROM 中；
3. 验证 FPGA 中 megalpmROM 的功能。

### 二、实验原理

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 lpmrom、lpmramio、lpmfifo、lpmramdq 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，lpmROM 是其中的一种。lpmROM 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。

### 三、实验步骤

1. 用图形编辑，进入 megalpm 元件库，调用 lpmrom 元件，设置地址总线宽度 address[] 和数据总线宽度 q[]，分别为 6 位和 24 位，并添加输入输出引脚。
2. 设置图 3-1-1 为工程。
3. 在设置 lpmrom 数据参数选择项 lpmfile 的对应窗口中（图 3-1-2），用键盘输入 lpmROM 配置文件的路径（roma.mif），然后设置在系统 ROM/RAM 读写允许，以便能对 FPGA 中的 ROM 在系统读写。
4. 用初始化存储器编辑窗口编辑 lpmROM 配置文件（文件名.mif）。这里预先给出后面将要用到的微程序文件：roma.mif。rom\_a.mif 中的数据是微指令码（图 3-1-3）。
5. 全程编译。
6. 下载 SOF 文件至 FPGA，改变 lpmROM 的地址 a[5..0]，外加读脉冲，通过实验台上的数码管比较读出的数据是否与初始化数据(roma.mif 中的数据)一致。
7. 打开 QuartusII 的在系统存储模块读写工具，了解 FPGA 中 ROM 中的数据，并对其进行在系统写操作

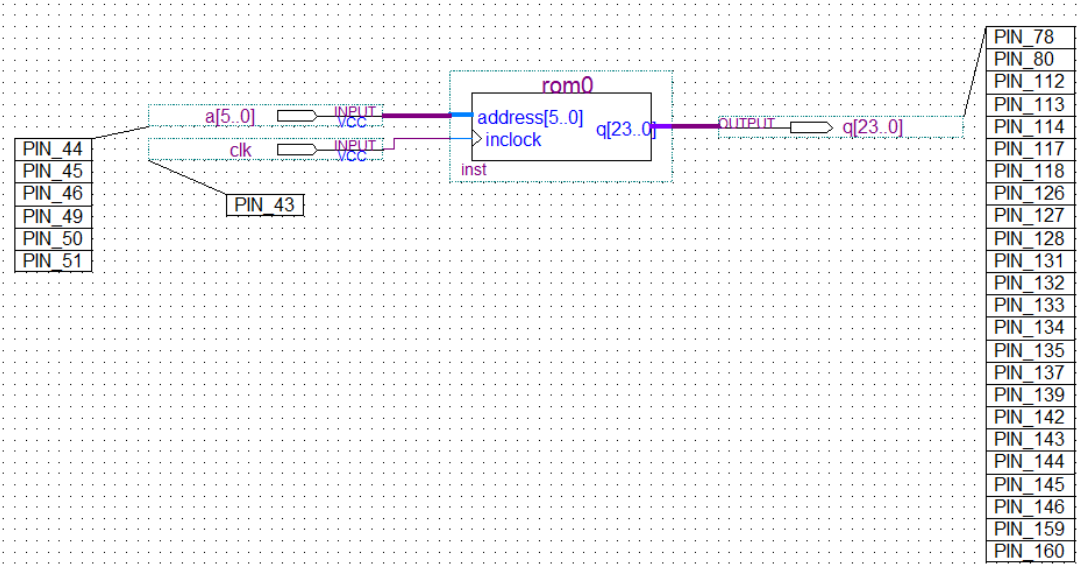


图 3-1-1 lpm\_ROM 的结构图

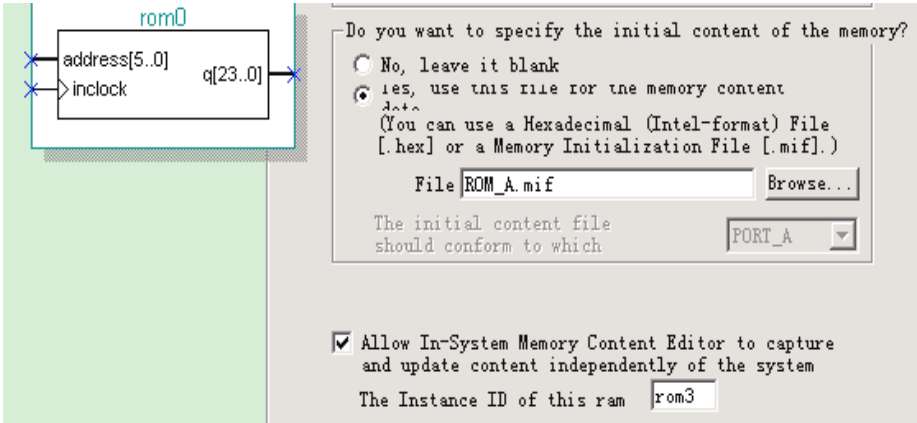


图 3-1-2 设置在系统 ROM/RAM 读写允许

Addr	+0	+1	+2	+3	+4	+5	+6	+7
00	018108	00ED82	00C050	00E004	00B005	01A206	959A01	00E00F
08	00ED8A	00ED8C	00A008	008001	062009	062009	070A08	038201
10	001001	00ED83	00ED87	00ED99	00ED9C	31821D	31821F	318221
18	318223	00E01A	00A01B	070A01	00D181	21881E	019801	298820
20	019801	118822	019801	198824	019801	018110	000002	000003
28	000004	000005	000006	000007	000008	000009	00000A	00000B
30	00000C	00000D	00000E	00000F	000010	000011	000012	000013
38	000014	000015	000016	000017	000018	000019	00001A	00001C

图 3-1-3 rom\_a.mif 中的数据

(7) 打开 QuartusII 的在系统存储模块读写工具，了解 FPGA 中 ROM 中的数据，并对其进行在系统写操作（图 3-1-4）。

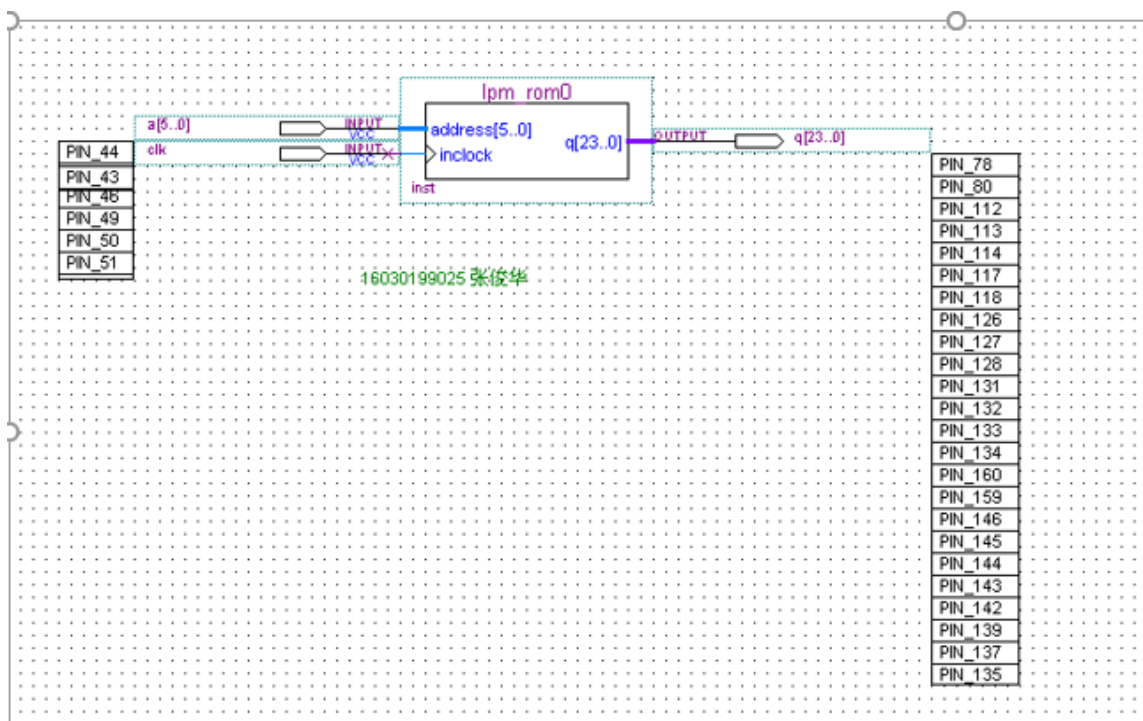
The screenshot shows the WinBox interface during an acquisition. The 'Acquisition in progress' window is active, displaying a table of acquisition data. The table has columns: Instance ID, Status, Width, Depth, Type, and Mode. The data shows a single row for 'rom3' with status 'Offloading...' and type 'RAM/ROM'. The 'JTAC Chain' window on the right shows 'JTAC ready' and 'Hardware: ByteElasterMV [LPT1]'.

Instance ID	Status	Width	Depth	Type	Mode
0 rom3	Offloading...	24	64	RAM/ROM	Read/Wri

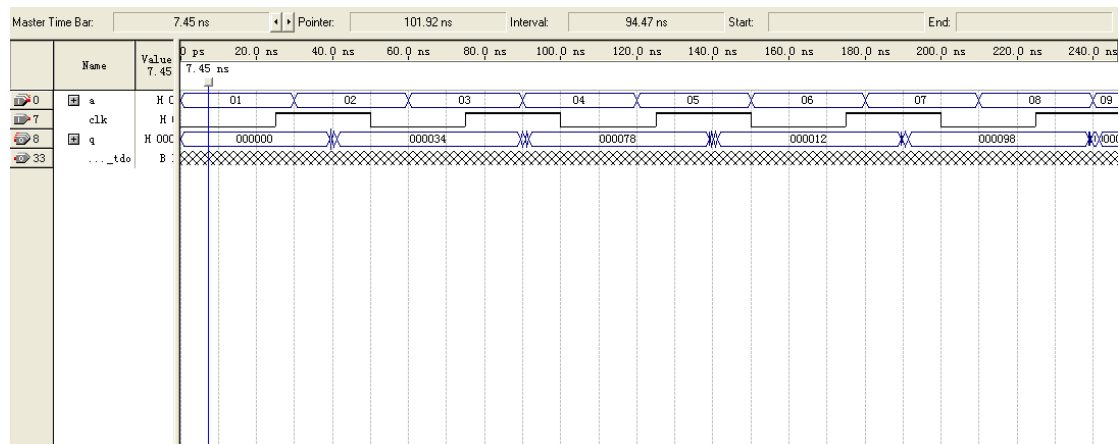
图 3-1-4 在系统存储模块读写

## 四、实验结果

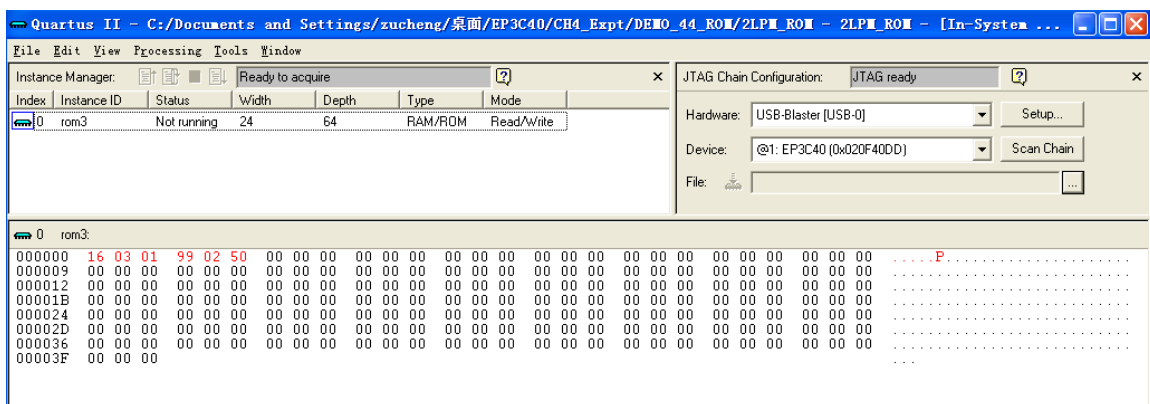
**LPM-ROM 仿真电路图如图所示:**



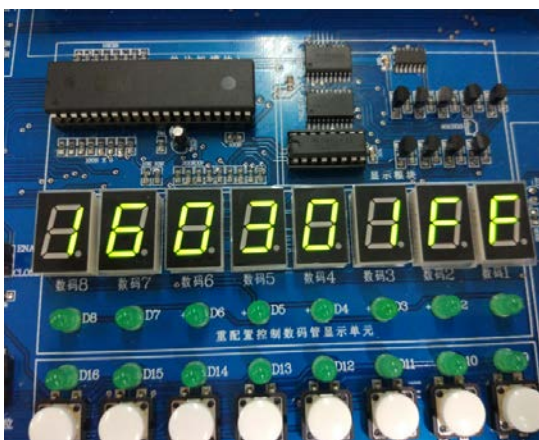
仿真波形图:



QuartusII 的在系统存储模块读写 ROM 数据如图所示:



该数据与实验箱数码管上显示内容相同:



## 实验二 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. 验证运算功能发生器的组合功能。
3. 掌握算术逻辑运算加、减、与的工作原理。

### 二、实验原理

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（图 2-1）；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。

### 三、实验步骤

1. 以原理图方式建立顶层文件工程

选择图形方式，电路图，从 Quartus II 的基本元件库中将各元件调入图形编辑窗口、连线，添加输入输出引脚。

2. 设计 ALU 元件

在 Quartus II 环境下，编辑器 Text Editor 输入 ALU181.VHD 算术逻辑单元文件，并将文件制作成一个可调用的原理图元件。

3. 芯片编程 Programming

打开编程窗口。将配置文件 ALU.sof 下载到 FPGA 中。

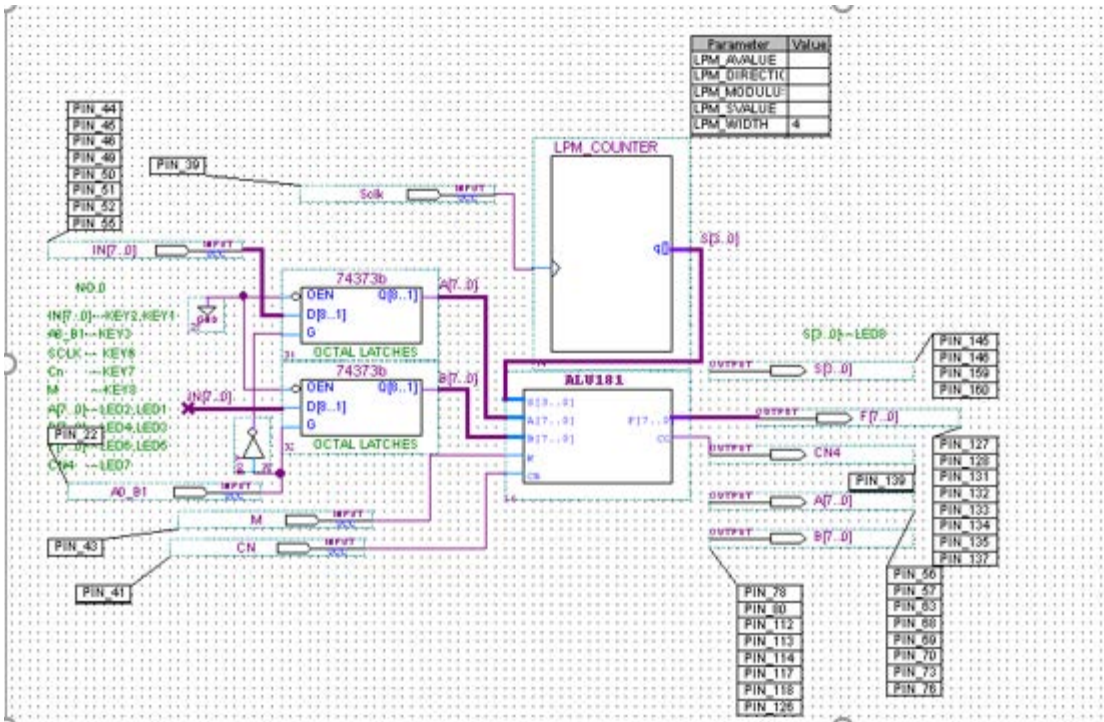
4. 选择实验系统的电路模式是 NO.0，验证 ALU 的运算器的算术运算和逻辑运算功能

根据表 2-1，从键盘输入数据 A[7..0]和 B[7..0]，并设置 S[3..0]、M、Cy，验证 ALU 运算器的算术运算和逻辑运算功能，记录实验数据。

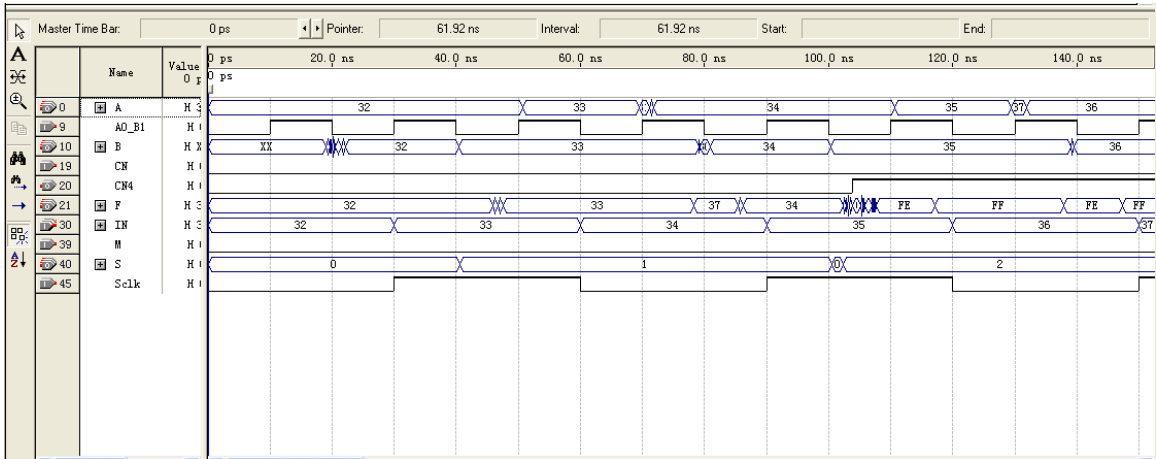
### 四、实验结果

运算器 仿真电路图如图所示：





仿真波形图：



验证 ALU181 的算术运算和逻辑运算功能表：

表 2-3

S3 S2 S1 S0	A[7..0]	B[7..0]	算术运算 M=0		逻辑运算 (M=1)
			cn=0 (无进位)	cn=1 (有进位)	
0000	AA	55	F= ( AA/0 )	F= ( AB/0 )	F= ( 55/1 )

0001	AA	55	F= ( FF/0 )	F= ( 00/1 )	F= ( 00/1 )
0010	AA	55	F= ( AA/1 )	F= ( AB/1 )	F= ( 55/0 )
0011	AA	55	F= ( 00/0 )	F= ( FF/1 )	F= ( 00/0 )
0100	FF	01	F= ( FD/1 )	F= ( FE/1 )	F= ( FE/1 )
0101	FF	01	F= ( FD/1 )	F= ( FE/1 )	F= ( FE/1 )
0110	FF	01	F= ( FE/0 )	F= ( FD/0 )	F= ( FE/0 )
0111	FF	01	F= ( FF/1 )	F= ( FE/1 )	F= ( FE/0 )
1000	FF	FF	F= ( FE/1 )	F= ( FF/1 )	F= ( 00/0 )
1001	FF	FF	F= ( FE/1 )	F= ( FF/1 )	F= ( FF/1 )
1010	FF	FF	F= ( FE/0 )	F= ( FF/0 )	F= ( FF/0 )
1011	FF	FF	F= ( FF/0 )	F= ( FE/0 )	F= ( FF/0 )
1100	55	01	F= ( AA/0 )	F= ( AB/0 )	F= ( 01/0 )
1101	55	01	F= ( AA/0 )	F= ( AB/0 )	F= ( FF/1 )
1110	55	01	F= ( 54/0 )	F= ( 55/0 )	F= ( 55/0 )
1111	55	01	F= ( 55/0 )	F= ( 54/0 )	F= ( 55/0 )

### 八种常用的算数与逻辑运算：

操作	S3 S2S1S0	M	Cn	DR1	DR2	运算关系及结果显示	Cn4
逻辑乘	1011	1	0	66	FF	$DR_1 \cdot DR_2 \rightarrow DR_2(66)$	0
传送	1111	1	0	66	66	$DR_1 \rightarrow DR_2(66)$	0
按位加	0110	1	0	66	66	$DR_1 \oplus DR_2 \rightarrow DR_2(00)$	0



取反	0000	1	0	66	00	$\overline{DR_1} \rightarrow DR_2 (99)$	1
加 1	0000	0	1	66	99	$DR_2 + 1 \rightarrow DR_2 (9A)$	0
求负	0010	0	1	66	9A	$\overline{DR_2} + 1 \rightarrow DR_2 (66)$	1
加法	1001	0	0	66	66	$DR_1 + DR_2 \rightarrow DR_2 (CC)$	0
减法	0110	0	0	66	CC	$DR_1 - DR_2 \rightarrow DR_2 (9A)$	1

### 认识和体会：

算术逻辑单元（ALU），是中央处理器 CPU 的一部分，用以计算机指令集中的执行算术与逻辑操作。通过这次实验，我了解了 ALU 的工作原理，掌握了 ALU 完成常用的算术和逻辑运算的方法，使我对计算机组成原理的学习有了更深的理解和印象。

## 实验三 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、实验原理

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。

#### 连续节拍发生电路：

可由 4 个 D 触发器组成，可产生 4 个等间隔的时序信号 T1~T4，其中 CLK1 为时钟信号，由实验台右边的方波信号源 clock0 提供，可产生 1Hz~12MHz 的方波信号频率。实验者可根据实验自行选择信号频率。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，机器进入连续运行状态（EXEC）。

### 单步节拍发生电路：

该电路每当 RST1 出现一个负脉冲后，仅输出一组 T1、T2、T3、T4 节拍信号，直到 RST1 出现下一个负脉冲

### 单步/连续节拍发生电路

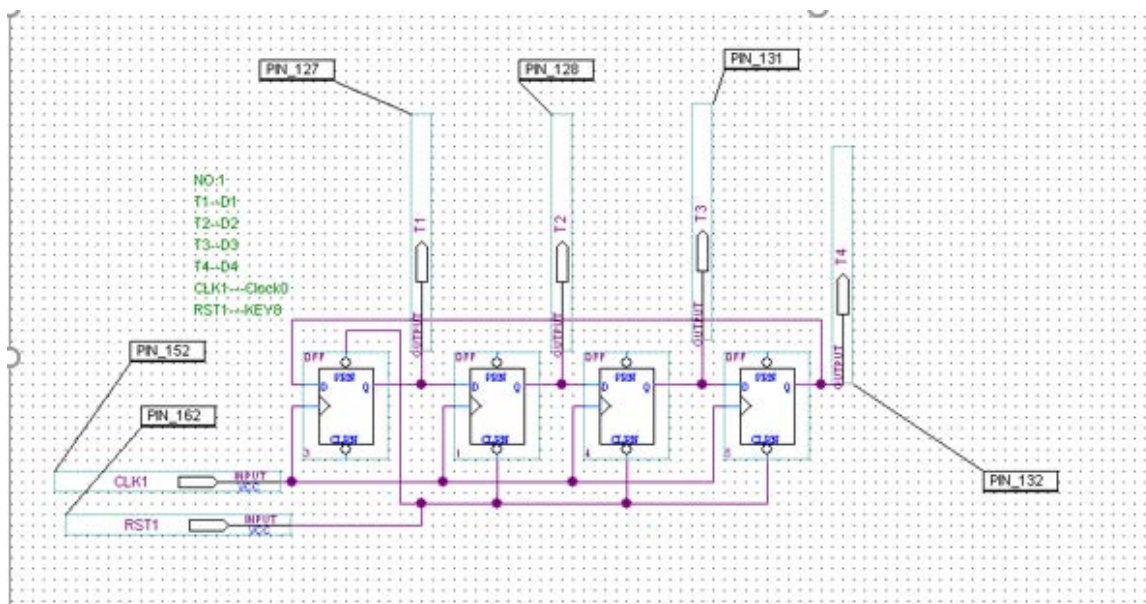
增加两个 2-1 多路选择器，可将图单步节拍发生电路改变为单步/连续节拍发生电路。S0 是单步或连续节拍发生控制信号，当 S0=0，选择单步运行方式；当 S0=1，选择连续运行方式。

## 三、实验步骤

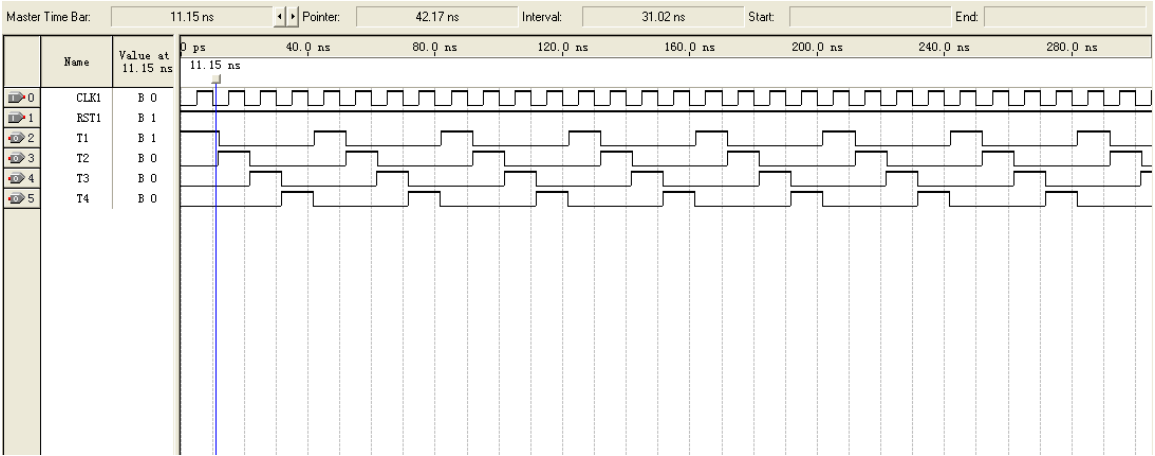
1. 硬件验证测试连续节拍发生电路（图 4-1-1），实验结果与仿真波形图比较
2. 硬件验证测试单步节拍发生电路（图 4-1-3），实验结果与仿真波形图比较
3. 硬件验证测试单步/连续节拍发生电路（图 4-1-5），实验结果与仿真波形图比较
4. 绘出相应的时序波形图。

## 四、实验结果

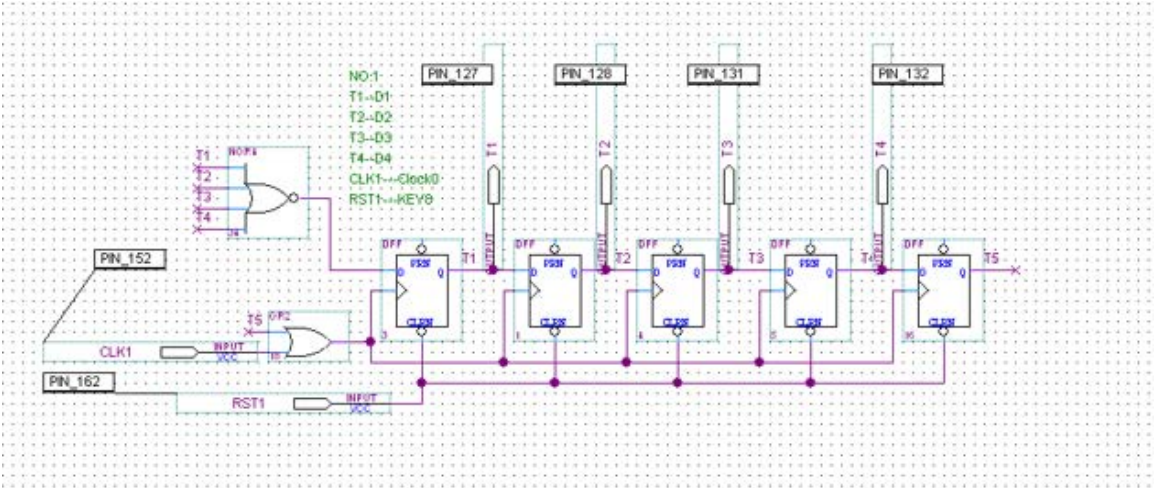
连续节拍发生电路图如图所示：



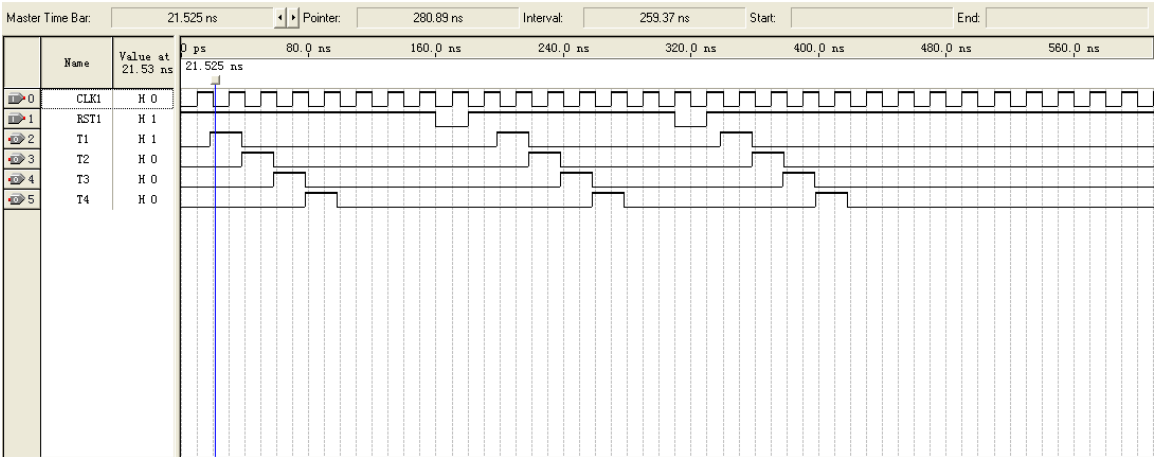
连续节拍发生电路时序波形图：



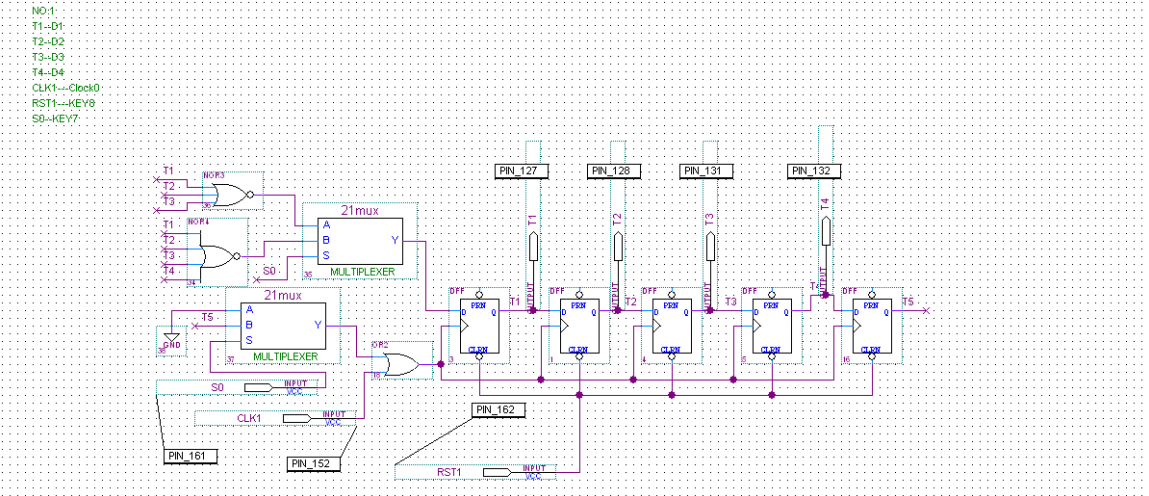
单步节拍发生电路图如图所示：



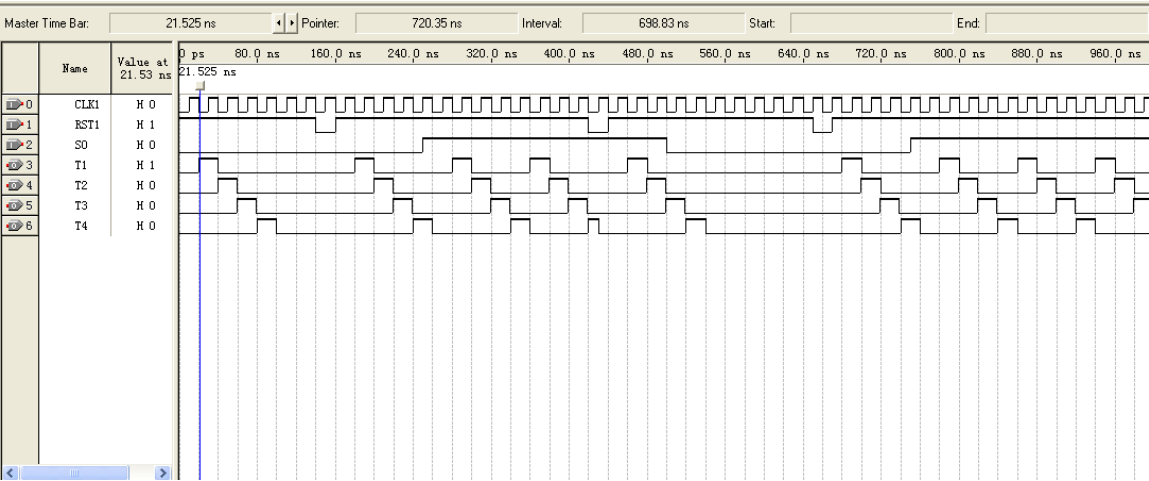
单步节拍发生电路仿真波形图：



连续/单步节拍发生电路图如图所示：



连续/单步节拍发生电路波形图：



实验分析：

本次实验，采用的是移位型节拍发生电路。**单步运行与连续运行的区别：**单步运行只在复位后 T1-T4 依次产生一个脉冲，对应与计算机中执行一条指令，而连续运行复位后 T1-T4 将依次产生脉冲不断循环下去。使用环境：单步运行用于调试器对单步指令运行的控制，连续运行用于程序的正常运行和连续运行控制连续指令执行。**如何实现单步/连续运行工作方式的切换？**通过 S0 切换，S0 为高电平时，复位后将连续运行，S0 为低电平，复位后将单步运行一次。

## 实验四 程序计数器 PC 与地址寄存器 AR 实验

### 一、实验目的

1. 掌握地址单元的工作原理。
2. 掌握的两种工作方式，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。

### 二、实验原理

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，如电路图 4-2-1 所示，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。

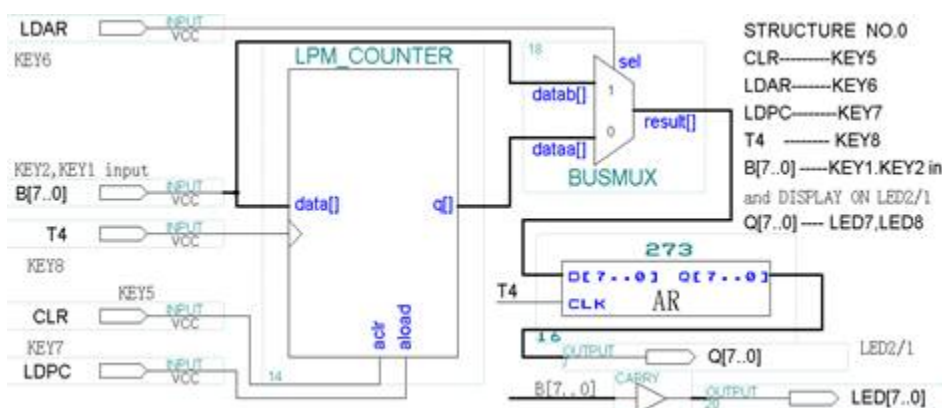


图 4-2-1 程序计数器原理图

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。



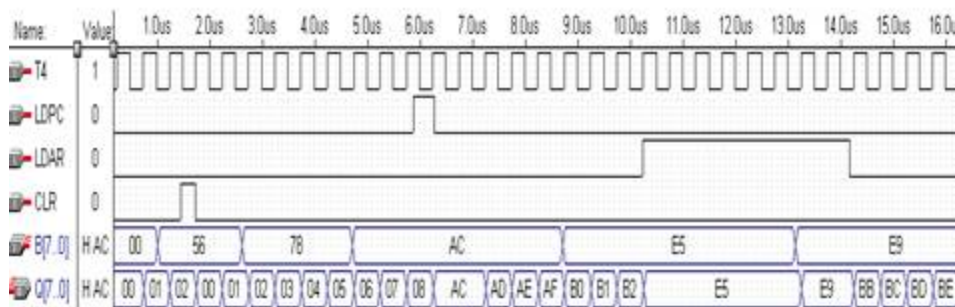


图 4-2-2 程序计数器工作波形

### 三、实验步骤

- 按照 图 4-2-1 程序计数器原理图编辑、输入电路，实验台选择 NO.0 工作模式。对输入原理图进行编译、引脚锁定、并下载到实验台。示例工程文件是 PC\_unit.bdf。硬件实验验证（与仿真波形图 4-2-2 比较！）。

实验说明：

（1）下载 pc\_unit.sof；

（2）用模式键选模式“0”，再按一次右侧的复位键；

（3）键 2 和键 1 可输入 8 位总线数据 B7..0；CLR（键 5）按 2 次(0à1à0)，产生一正脉冲，高电平清零；LDAR（键 6）=0 时，BUSMUX 输出程序计数器 PC 的值；LDAR=1 时，BUSMUX 输出 B[7..0]总线数据。LDPC（键 7）：程序计数器 PC 预置控制端，当 LDPC=1 时，将 B[7..0]总线数据装入程序计数器 PC；当 LDPC=0 时，程序计数器 PC 处于计数自动工作状态，对 T4 进行计数；T4（键 8）：程序计数器 PC 的计数时钟 CLK，键 8 按动两次产生一个计数脉冲。

- 通过 B[7..0]设置程序计数器的预加载数据。当 LDPC=0 时，观察程序计数器自动加 1 的功能；当 LDPC=1 时，观察程序计数器加载输出情况，

示例操作：

1、所有键置 0，键 2/1 输入 A5；按键 5àPC 计数器清 0(0à1à0)；

2、连续按动键 8，可以从数码 8/7 上看到 AR 的输出，即 PC 值；

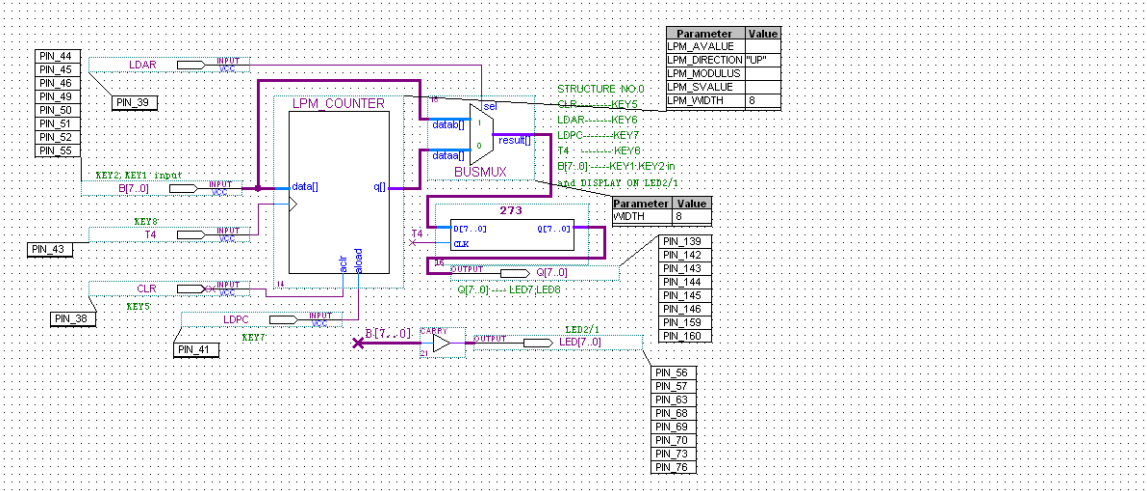
3、按键 6->'1'，选通直接输出总线上的数据 A5 作为 PC 值，按键 8，产生一个脉冲上升沿，即可看到 AR（显示在数码 8/7）的输出为 A5；

4、使键 6=0，仍选通 PC 计数器输出，这时键 2/1 输入 86，按键 7 产生一个上升脉冲(0à1à0)，即用 LDPC 将 86 加载进 PC 计数器；

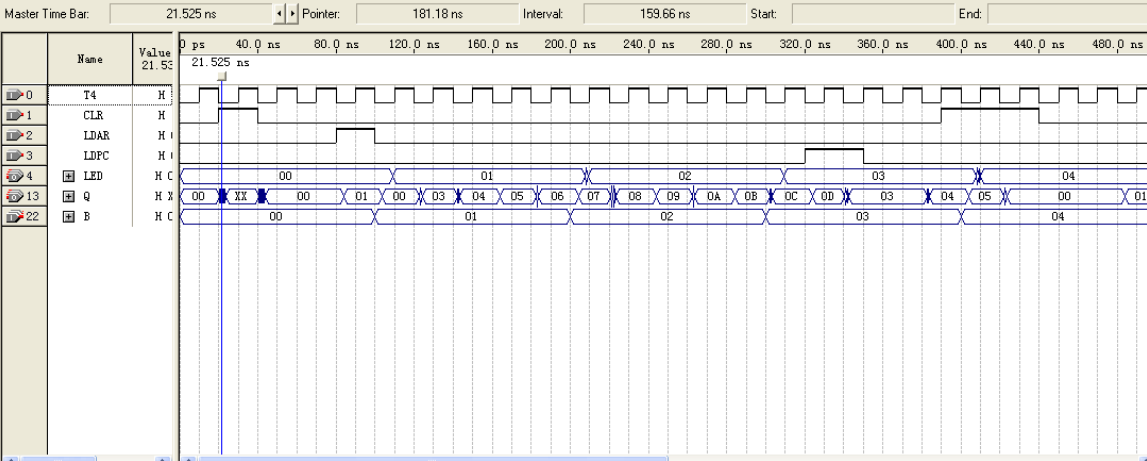
5、连续按动键 8，可以发现 AR 的输出在 86 上累加输出：86、87、88 等。

四、实验结果

实验电路图如图所示：



仿真波形如图所示：



实验结果讨论、分析：

本次实验，验证了程序计数器 PC 和 地址寄存器 AR 的工作原理，掌握了 PC 加一计数和重置初值的方法：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。

地址寄存器可以从程序计数器获得数据和从内部总线获得数据。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的数据。



## 实验五 总线控制实验

### 一、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。

### 二、实验原理

#### 总线的基本概念

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。

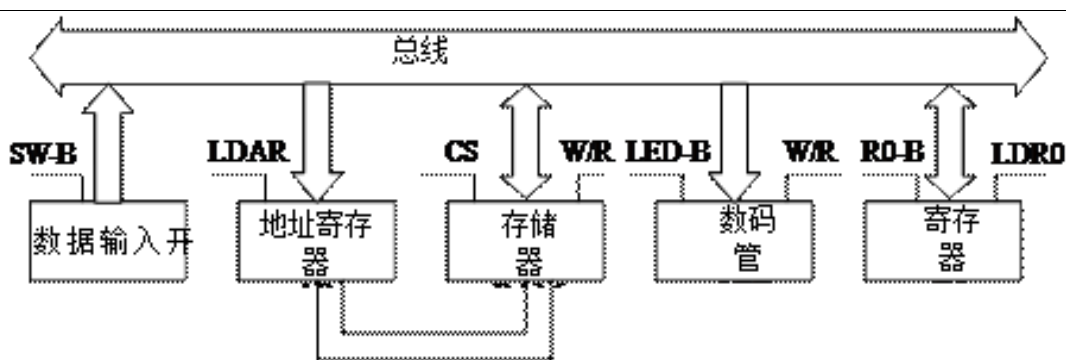


图 5-1 总线实验传输框图

#### 实验原理

实验所用总线实验传输框图如图 5-1 所示。它将几种不同的设备挂在总线上，有存储器、输入设备、输出设备、寄存器。这些设备在传统的系统中需要有三态输出控制，然而在 FPGA 的内部没有三态输出控制结构，因此必须采用总线输出多路开关结构加以控制。按照传输要求恰当有序地控制它们，就可以实现总线信息传输。

### 三、实验步骤

1. 实验电路如图 5-2 所示。写使能  $WE=1$  允许写， $=0$  禁止写，允许读；inclock 为数据 DATA 锁存时钟。具体操作可参考图 5-3。

2. 文件是 BUS-4.bdf，下载 BUS-4.sof 到实验台的 FPGA 中；

### 3. 实验内容 1

根据图 5-2 完成实验操作：选择实验模式“0”；再按一次右侧的复位键(用一接线将实验板上键 9 的输入端插针与适配板上 FPGA 的第 P196 针相连，以便能用键 9 控制 OUT 锁存器的时钟；)：初始状态；1、键 4、键 3 控制设备选择端：sel[1..0]=00（键 4/键 3=00，）；2、此时由键 2/键 1 输入的数据(26H，显示于数码管 2/1)直接进入 BUS（数码管 8/7 显示），键 5、6、7 为低电平；3、键 8=1（允许 RAM 写入）完成图 5-2 所示的操作：4、键 5 发正脉冲（0-1-0），将数据打入寄存器 R0；5、键 2/键 1 再输入数据(如 37H)；6、键 6 发正脉冲（0-1-0），将数据打入地址寄存器 AR；7、键 2/键 1 再输入数据(如 48H)；8、键 7 发正脉冲（0-1-0），将数据写入 RAM（此时必须键 8 输出‘1’，注意此时进入 RAM 的数据 48H 是放在地址 37H 单元的）；9、键 2/键 1 再输入数据(如 59H)；10、键 9 发正脉冲（0-1-0），将数据写入寄存器 OUT（数码管 6/5 将显示此数）；11、键 4、键 3 分别选择 sel[1..0]=00、01、10、11，从数码管 8/7 上观察被写入的各寄存器中的数据。

### 实验内容 2

先将数据 28H 写入 RAM 的地址（4AH），再将数据 1BH 送进 R0，最后将刚才写入 RAM 中地址（4AH）的数据读出送到 OUT 口。依据总线电路图 5-3，操作如下：

1、用一接线将实验板上键 9 的输入端插针与适配板上 FPGA 的第 P196 针相连，以便能用键 9 控制 OUT 锁存器的时钟；键 3、4、5、6、7、8 都为低电平，使键 4/键 3=00，即总线多路选择器 sel[1..0]=00，选择由键 2/键 1 输入的数据 4AH（地址），直接进入 BUS；

2、按键 6 两次（0-1-0），产生一个正脉冲，将地址数据 4AH（地址）锁入地址寄存器 AR，如图 5-3 所示，此数据直接进入 RAM 的 address 端；

3、按键 2/键 1，输入数据 28H（数据），此时直接进入总线 BUS，并进入 RAM 的 data 数据端；按键 8=1（RAM 写允许）；按键 7 两次，将数 28H 写入 RAM（地址为 4AH），最后按键 8=0，写禁止，读允许。

4、由键 2/键 1 输入的数据 1BH，按键 5 两次（0-1-0），产生一个正脉冲，即此数写入 R0 寄存器。

5、读 RAM 送到 OUT：由键 2/键 1 输入的数据 4AH，按键 5 两次，使 4AH 进入 AR；

6、按键 7 两次，RAM 中 4AH 单元中的数据 28H 输出，再使键 4/键 3=10，即总线多路选择器 sel[1..0]=10，此时 RAM 数据口的 28H 进入总线 BUS（可从数码管 8/7 上看到）；

7、按键 9 一次（此键是单脉冲），RAM 口的 28H 即被锁如输出口 OUT 寄存器，由数码管 6/5 显示。

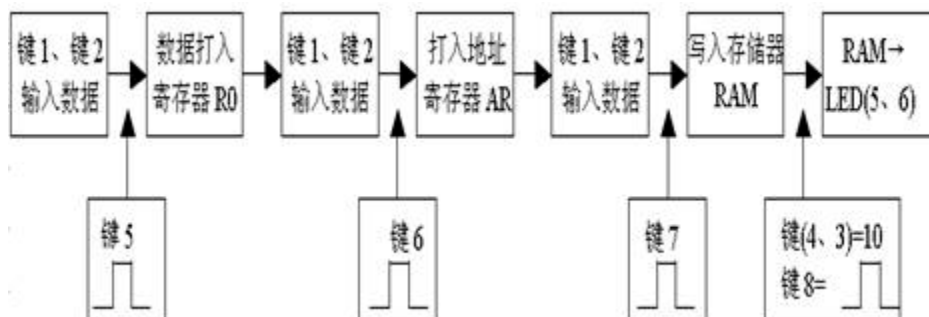


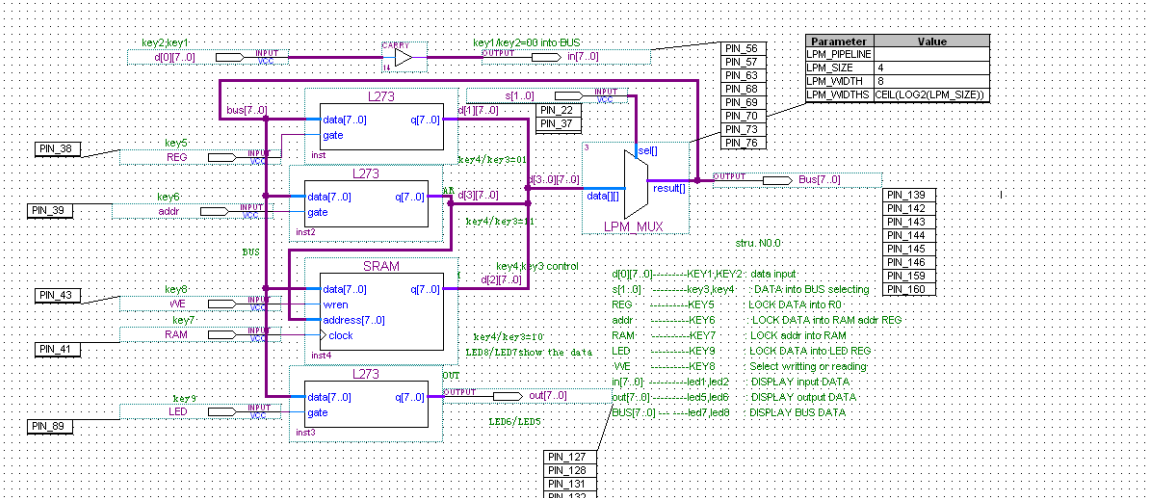
图 5-2 总线数据传输练习操作步骤

8、键盘/显示定义详细说明：

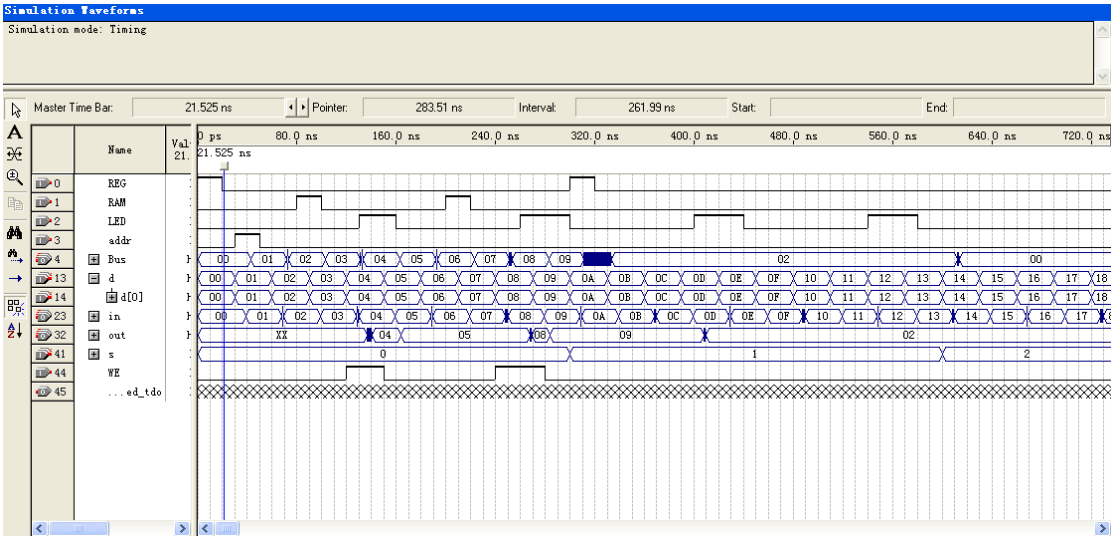
- 1) 键 2、键 1 输入 D[7..0]，输入的数据同时显示在数码 2 和数码 3 上。
- 2) 键 9、键 3 输入控制设备选择端 sel[1..0]，如图 5-2 所示，键 4/键 3 控制总线多路选择器，选择不同设备的数据进入总线：sel[1..0]= 00：输入设备 INPUT 数据进入总线 BUS；= 01：寄存器 R0 中的数据进入总线 BUS；= 11：地址寄存器 AR 的数据进入总线 BUS；= 10：存储器 RAM 的数据进入总线 BUS；
- 4) 总线 BUS 上的输出数据显示在数码 8 和数码 7 上；
- 5) 键 5 控制寄存器 R0 的输入选通锁存端；
- 6) 键 6 控制地址寄存器 AR 输入选通锁存端；
- 7) 键 7 控制 LPM\_RAM 数据 DATA 输入锁存端；
- 8) 键 8 控制 LPM\_RAM 写入允许 WE 端，=1 有效；
- 9) 键 9 控制输出设备 OUTPUT 的输入选通端，输出数据显示在数码 6 和数码 5 上，要求首先用一接线将实验板上键 9 的输入端插针与适配板上 FPGA 的第 P196 针相连。

## 四、实验结果

实验电路图如图所示：



仿真时序波形如图所示：



实验结果分析：

本次实验，通过总线，实现了系统部件之间进行数据传输。总线是计算机各种功能部件之间传送信息的公共通信干线它是由导线组成的传输线束，按照计算机所传输的信息种类，计算机的总线可以划分为数据总线、地址总线和控制总线，分别用来传输数据、数据地址和控制信号。总线是一种内部结构，它是cpu、内存、输入、输出设备传递信息的公用通道，主机的各个部件通过总线相连接，外部设备通过相应的接口电路再与总线相连接，从而形成了计算机硬件系统。在计算机系统中，各个部件之间传送信息的公共通路叫总线，微型计算机是以总线结构来连接各个功能部件的。

本次实验，总线将寄存器 R0、AR、RAM 相连，实现了输入数据的保存与输出。

**思考题：**

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)

答：利用键一，键二，键三分别把数据输入，然后再用控制信号产生不同的脉冲将数据存入 **RAM** 中，然后再用相同的脉冲将数据读出来。

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？

答：可能会发生冲突，解决方法：当它们有总线请求时，把它们唯一的仲裁号发送到共享的仲裁总线上，每个仲裁器将仲裁总线上得到的号与自己的号进行比较。如果仲裁总线上的号大，则它的总线请求不予响应，并撤消它的仲裁号。最后，获胜者的仲裁号保留在仲裁总线上。得到总线的使用权

## 实验心得

这学期的计算机组成原理实验的学习让我受益匪浅。在实验课即将结束之时，我对在这一学期的学习进行了总结，总结这一学期的收获与不足。取之长、补之短，在今后的学习和工作中有所受用。

没有接触过计算机或者对计算机不是特别了解的人可能觉得计算机特别神秘而且不知道为什么它可以实现那么复杂的功能，而就我们而言越是深入学习越是渴望了解其工作原理。很幸运这学期我们开设了计算机组成原理实验课，同时也开设了计算机组成原理专业课，之所以将计算机组成原理实验独立成一门课程，足见其重要性。也体现了现在大学教学中对大学生动手能力的重视。《计算机组成原理实验系统》这门课程是我们计算机专业一门很重要的专业课起着承上启下的作用。这学期的实验课，通过自己的亲自操作让我对计算机的基本结构，基本组成与结构原理有了更加深入的了解，这门课程对于我们了解现代计算机的各个组成部分及其工作原理有重要作用，通过自己动手做实验使我们更加深入的了解计算机系统中各种功能的实现方法。这学期的实验课由浅及深，分别从存储器实验，算术逻辑运算实验，程序计数器 PC、总线等几个方面先掌握计算机系统中一些基本的功能模块，通过对这些模块功能的学习，我们掌握了一些芯片的管脚功能及其工作原理，但是由于课程及实验时间的限制，我想我们学到的东西还是太少了，不过没关系，这毕竟为我们以后的学习打下了基础。

总之，这次实验给我提供了动手实验的机会，使我对计算机组成原理的相关知识有了更深的印象和认识。计算机组成原理是计算机专业的基础课。这门课对于使我们了解现代计算机的各个组成部分及其工作原理具有重要作用，对于我们后续课程的学习无疑也具有积极的意义。计算机专业是一个很渊博的专业，我们现在有很好的机会站在巨人的肩膀上学习，虽然通过这学期的实验学到了很多知识，但那只是计算机知识海洋中的一滴，我将继续努力对计算机组成原理方面进行深入研究，了解更多计算机方面的知识，为以后打下坚实的基础。