

西安电子科技大学

计算机组织与体系结构课程实验报告

实验名称 计算机组织与体系结构

计算机科学与技术学院 2003019 班

姓名 周灿 学号 20009200741

同作者 无

实验日期 2022 年 6 月 12 日

实验地点 线上 实验批次 计组实验3群

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 mega_lpm_ROM 的功能。

二、实验环境

线上，quartus2 软件。

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 3 组信号：地址信号 address[]、数据信号 q[]、时钟信号 clk。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。

(1) 用图形编辑，进入 mega_lpm 元件库，调用 lpm_rom 元件以及 lpm_ram 元件，设置地址总线宽度 a[] 和数据总线宽度 q[]，分别为 6 位和 8 位，并添加输入输出引脚。设置地址。

(2) 波形仿真

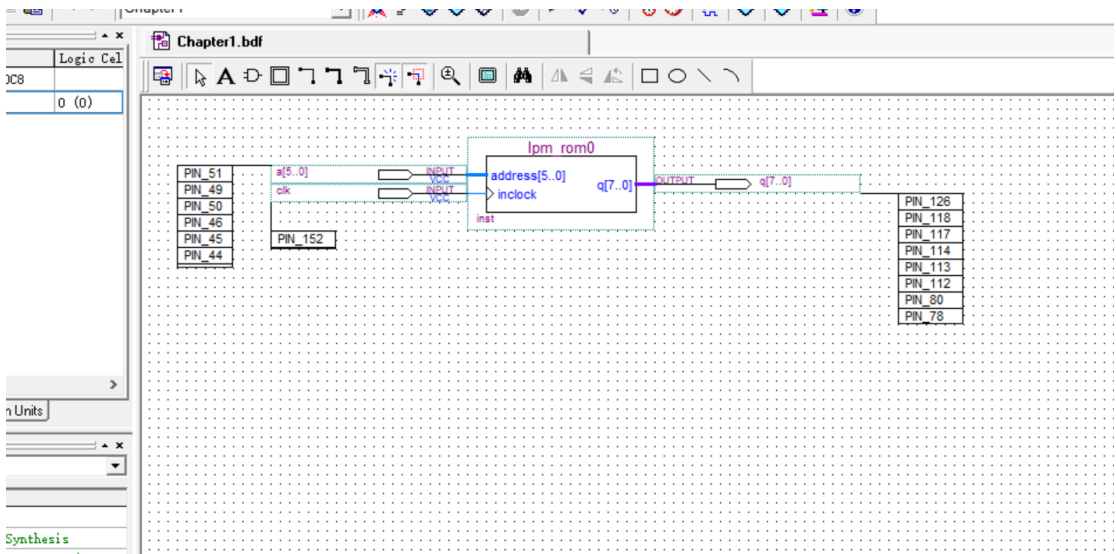
打开 QuartusII，选 File -> New other files -> Vector Waveform

File, 点击 OK 进入仿真界面。

点 Node Finder, 点 list 选中需要仿真的信号, :a[5..0], clk , [7..0]点击 OK。加入信号, 点快捷键仿真。

Lpm_rom

实验原理图:



地址设置为:

四、结果

Chapter1

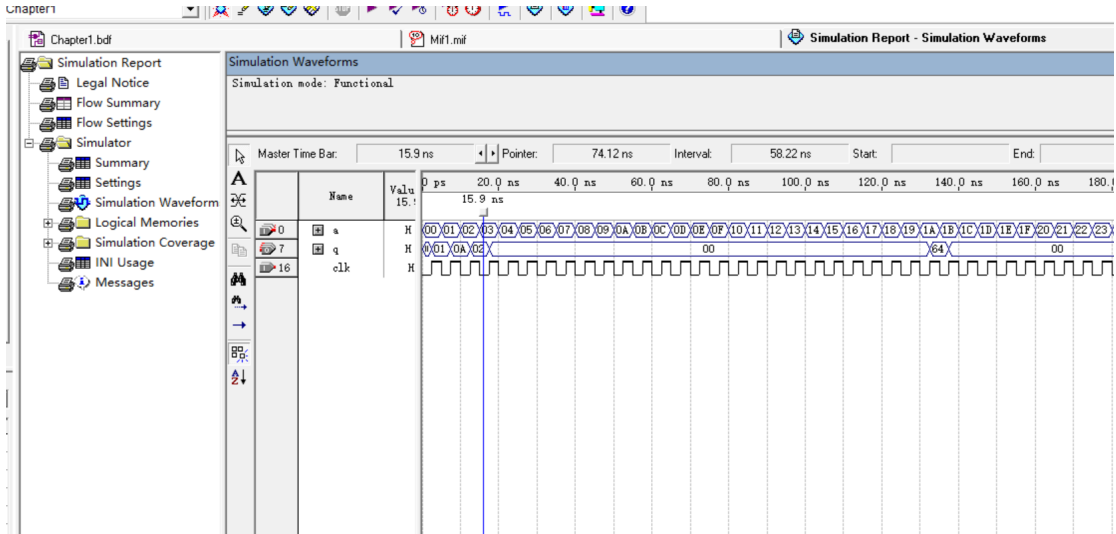
Chapter1.bdf

Mif1.mif

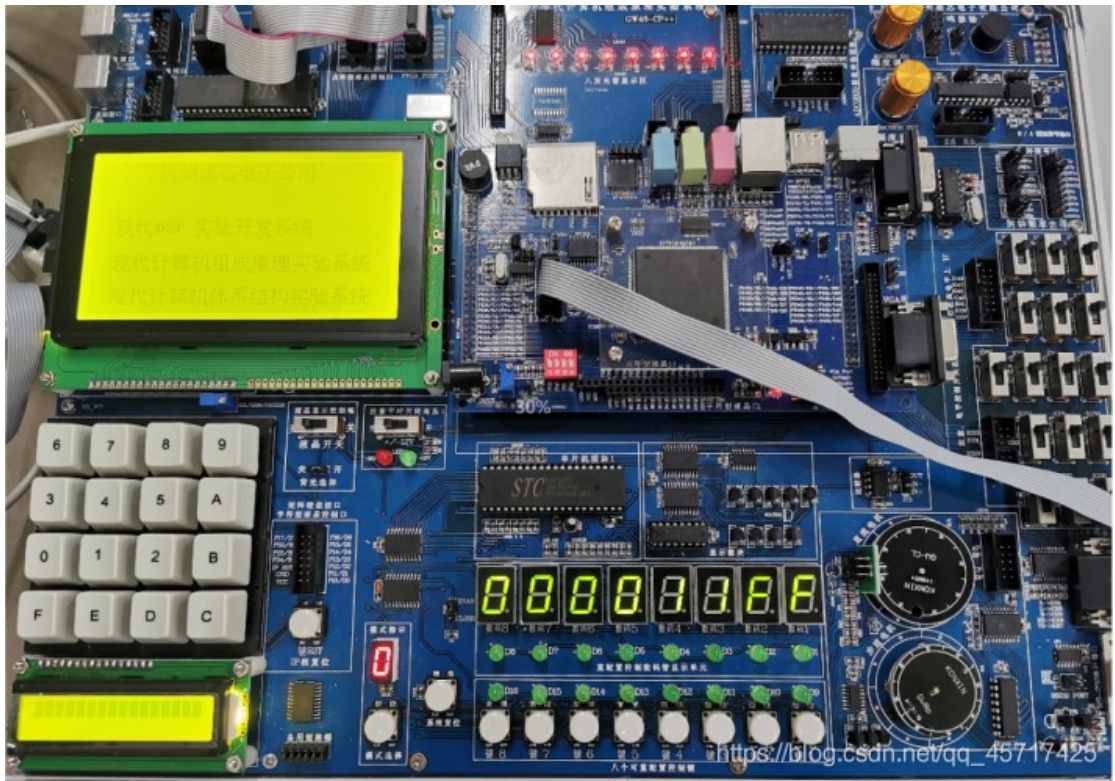
Addr	+0	+1	+2	+3	+4	+5	+6	+7
00	01	02	03	04	05	06	07	08
08	09	0A	0B	0C	0D	0E	0F	00
10	10	11	12	13	14	15	16	17
18	18	19	3F	3F	3F	3F	3F	3F
20	20	21	3F	3F	3F	3F	3F	3F
28	28	29	3F	3F	3F	3F	00	00
30	30	31	3F	3F	3F	00	00	00
38	38	39	3F	3F	00	00	00	00

Compile Design
Analysis & Synthesis
Fitter (Place & Route)

波形仿真



FPGA 实验板初始化状态



五、实验结果分析

该存储器通过时钟，在每个时钟周期输出此时输入的地址所存储的数据。

选择实验电路模式为 NO。0，24 位数据输出由数码 8 至数码 3 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 8/7/6/5/4/3 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。

例如：当由键 2、键 1 输入的地址为 000001 时，数码管 3 显示 000001F，表示读取的数据为十进制的 1；当由键 2、键 1 输入的地址为 010001 时，数码管 3 显示 000011FF，表示读取的数据为十进制的 17。由此可见，本次实验成功地验证了 FPGA 中 mega_lpm_ROM 的功能。

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。
- 4、验证实验台运算的 8 位加、减、与、直通功能。
- 5、按给定数据，完成几种指定的算术和逻辑运算。

二、实验环境

线上，quartus2 软件。

三、实验基本原理及步骤

算术逻辑单元 ALU 的数据通路如图 2-1 所示。其中运算器 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，而 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 2-1 所示。

选择端	高电平作用数据		
S3 S2 S1 S0	M=H	M=L 算术操作	
	逻辑功能	Cn=L（无进位）	Cn=H（有进位）
0 0 0 0	$F = \overline{A}$	$F = A$	$F = A \text{加} 01$
0 0 0 1	$F = \overline{A + B}$	$F = A + B$	$F = (A + B) \text{加} 1$
0 0 1 0	$F = \overline{AB}$	$F = A + \overline{B}$	$F = A + \overline{B} + 1$
0 0 1 1	$F = 0$	$F = \text{减} 1 \text{（} 2 \text{的补码）}$	$F = 0$
0 1 0 0	$F = \overline{AB}$	$F = A \text{加} A\overline{B}$	$F = A \text{加} A\overline{B} \text{加} 1$
0 1 0 1	$F = \overline{B}$	$F = (A + B) \text{加} A\overline{B}$	$F = (A + B) \text{加} A\overline{B} + 1$
0 1 1 0	$F = A \oplus B$	$F = A \text{减} B$	$F = A \text{减} B \text{减} 1$
0 1 1 1	$F = A\overline{B}$	$F = A + \overline{B}$	$F = (A + \overline{B}) \text{减} 1$
1 0 0 0	$F = \overline{A} + B$	$F = A \text{加} AB$	$F = A \text{加} AB \text{加} 1$
1 0 0 1	$F = \overline{A \oplus B}$	$F = A \text{加} B$	$F = A \text{加} B \text{加} 1$
1 0 1 0	$F = B$	$F = (A + \overline{B}) \text{加} AB$	$F = (A + \overline{B}) \text{加} AB \text{加} 1$
1 0 1 1	$F = AB$	$F = AB$	$F = AB \text{减} 1$
1 1 0 0	$F = 1$	$F = A \text{加} A \cdot$	$F = A \text{加} A \text{加} 01$
1 1 0 1	$F = A + \overline{B}$	$F = (A + B) \text{加} A$	$F = (A + B) \text{加} A \text{加} 1$
1 1 1 0	$F = A + B$	$F = (A + \overline{B}) \text{加} A$	$F = (A + \overline{B}) \text{加} A \text{加} 1$
1 1 1 1	$F = A$	$F = A$	$F = A \text{减} 1$

注 1、* 表示每一位都移至下一更高有效位，“+”是逻辑或，“加”是算术加

注 2、在借位减法表达上，表 2-1 与标准的 74181 的真值表略有不同。

1. 设计 ALU 元件

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

2. 以原理图方式建立顶层文件工程

选择图形方式。根据图 2-1 输入实验电路图，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。

将所设计的图形文件 ALU.bdf 保存到原先建立的文件夹中，将当前文件设置成工程文件，以后的操作就都是对当前工程文件进行的。

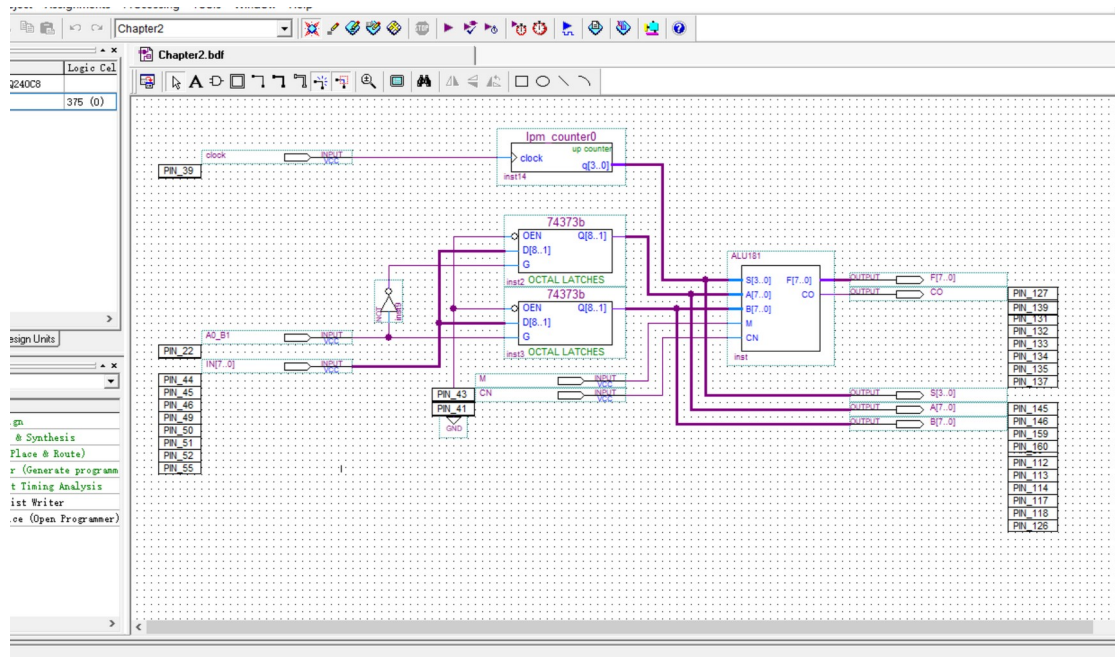
3. 器件选择

选择 Cyclone 系列，在 Devices 中选择器件 EP1C6QC240C8。编译，引脚锁定，再编译。引脚锁定后需要再次进行编译，才能将锁定信息确定下来，同时生成芯片编程/配置所需要的各种文件。

4. 芯片编程

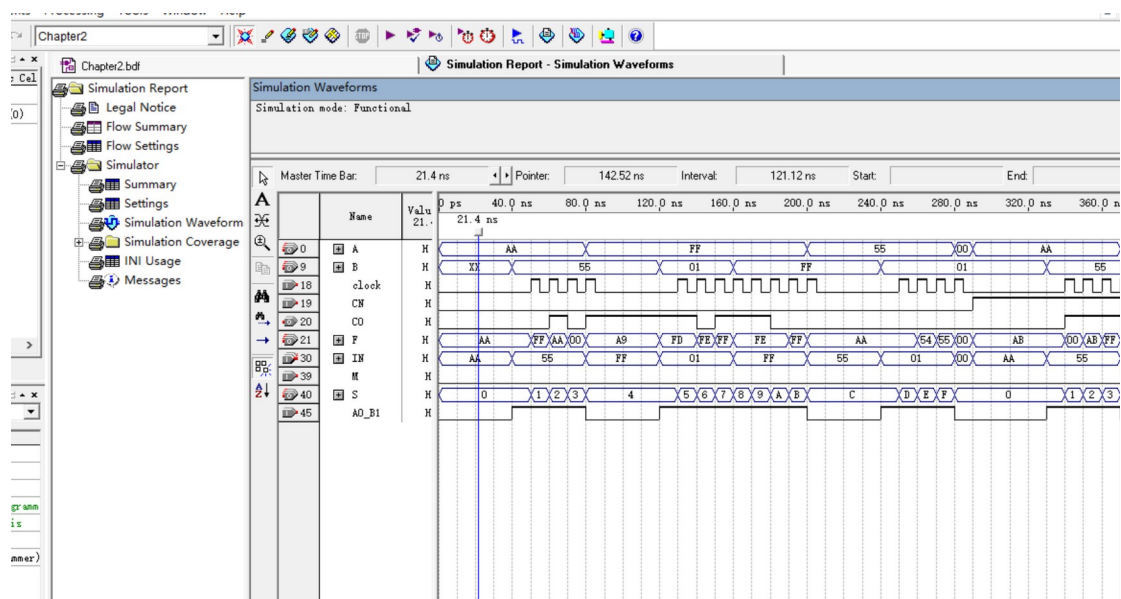
打开编程窗口。将配置文件 ALU.sof 下载进 GW48 系列现代计算机组成原理系统中的 FPGA 中。

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

仿真波形



填写表格

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	A A	5 5	F= (0AA)	F= (0AB)	F= (155)
0001	A A	5 5	F= (0FF)	F= (100)	F= (100)
0010	A A	5 5	F= (1AA)	F= (1AB)	F= (055)
0011	A A	5 5	F= (000)	F= (1FF)	F= (000)
0100	F F	0 1	F= (1FD)	F= (1FE)	F= (1FE)
0101	F F	0 1	F= (1FD)	F= (1FE)	F= (1FE)
0110	F F	0 1	F= (0FE)	F= (0FD)	F= (0FE)
0111	F F	0 1	F= (1FF)	F= (1FE)	F= (0FE)
1000	F F	F F	F= (1FE)	F= (1FF)	F= (000)
1001	F F	F F	F= (1FE)	F= (1FF)	F= (1FF)
1010	F F	F F	F= (0FE)	F= (0FF)	F= (0FF)
1011	F F	F F	F= (0FF)	F= (0FE)	F= (0FF)
1100	5 5	0 1	F= (0AA)	F= (0AB)	F= (0AB)
1101	5 5	0 1	F= (0AA)	F= (0AB)	F= (1FF)
1110	5 5	0 1	F= (054)	F= (055)	F= (055)
1111	5 5	0 1	F= (055)	F= (054)	F= (055)

五、实验结果分析

在验证 ALU 功能特性的时候，我发现当 S3 S2 S1 S0 = 1000, M=1, Cn=0 时，ALU 原本应该实现 F=A 非+B，但实际操作下来却发现并非如此，询问老师后得知可能是 ALU181 真值表与实际功能实现间存在极少数不对应的情况。

实验三 微控制器实验

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

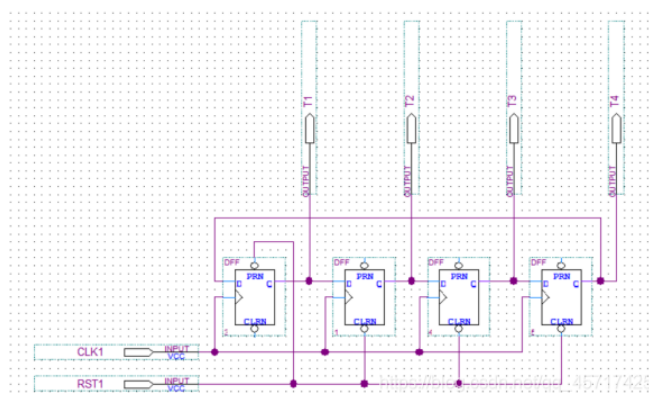
线上，quartus2 软件。

三、实验基本原理及步骤

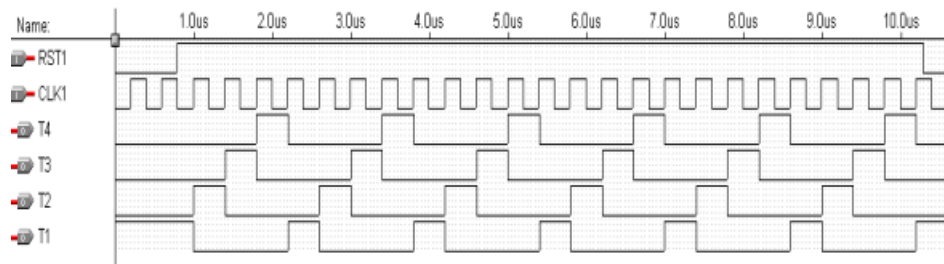
计算机之所以能够按照人们事先规定的顺序进行一系列的操作或运算，就是因为它的控制部分能够按一定的先后顺序正确地发出一系列相应的控制信号。这就要求计算机必须有时序电路。控制信号就是根据时序信号产生的。本实验说明时序电路中节拍脉冲发生器的工作原理。

1.连续节拍发生电路：

可由 4 个 D 触发器组成，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）。



T1~T4 以及 CLK1、RST1 的工作波形如图 3-2 所示。硬件实验验证方法如图 3-1 所示，下载.SOF 文件，选择实验模式 1，Clock0 接 4Hz，键 8 控制 RST1，高电平时可以看到，发光管 1、2、3、4 分别显示 T1、T2、T3、T4 的输出电平。



2.单步节拍发生电路：

将图 3-1 电路稍加改变即可得到图 3-3 所示的单步运行电路。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，波形如图 3-4 所示。

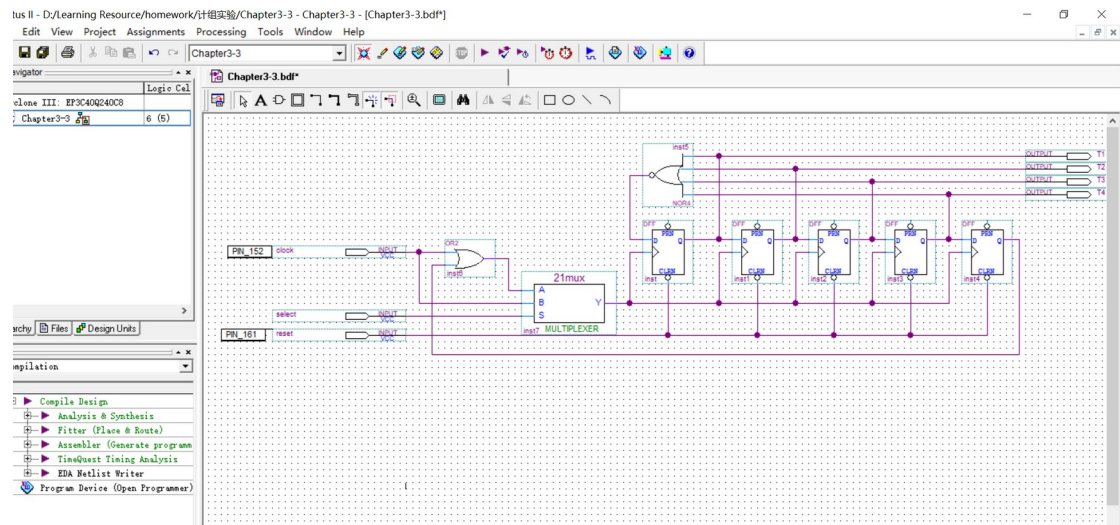
示例工程文件是 T5.bdf。硬件实验验证方法如图 4-1-3 所示，下载 T5.SOF 文件，选择实验模式 1，Clock0 接 4Hz（选择范围是 1Hz-50MH），键 8 控制 RST1。每出现一个负脉冲，发光管 1、2、3、4 分别显示 T1、T2、T3、T4 的输出电平一次。

3.单步/连续节拍发生电路：

增加两个 2-1 多路选择器，可将图 3-3 电路改变为图 3-5 所示电路。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。图 4-1-6 为此电路的仿真波形。

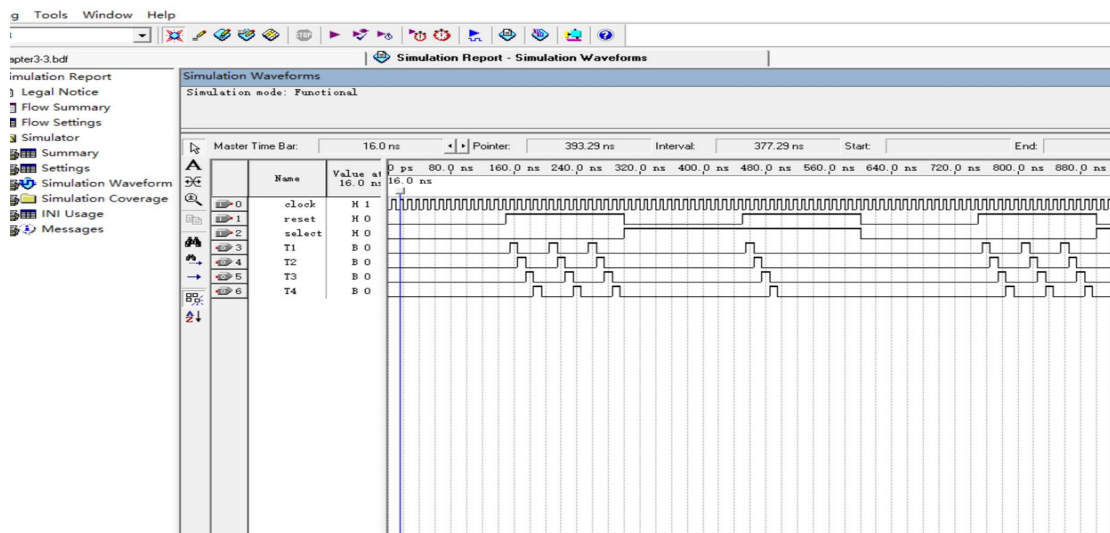
示例工程文件是 TS5.bdf。硬件实验验证：下载 TS5.SOF 文件，选择实验模式 1，Clock0 接 4Hz，键 8 控制 RST1，键 7 控制 S0，发光管 1、2、3、4 分别显

示 T1、T2、T3、T4 的输出电平。

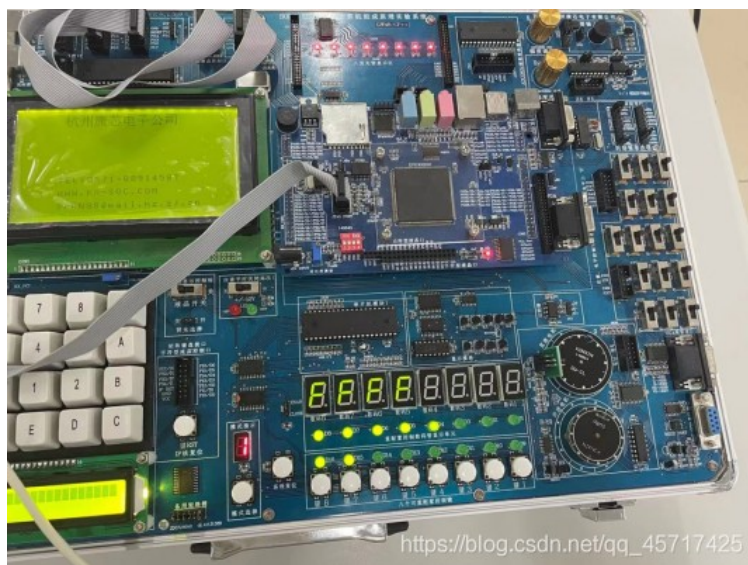


四、实验结果

单步/连续运行电路工作波形



实验箱硬件测试连线图



五、实验结果分析

节拍脉冲发生器实验：实验中选择实验模式 1，Clock0 选择信号频率为 4Hz。当按下按键 8 使 RST1 为低电平时，T1 输出为“1”，而 T2、T3、T4 输出为“0”，此时 D4、D3、D2、D1 不亮；当按下键 8 使 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，体现为 D4、D3、D2、D1 轮流亮光。

单步运行电路实验：选择实验模式 1，Clock0 接 4Hz，键 8 控制 RST1。按下键 8 使输出为低电平时，出现一个负脉冲，此时发光管 D1、D2、D3、D4 分别显示 T1、T2、T3、T4 的输出电平一次；当按下键 8 使输出为高电平时，发光管 D1、D2、D3、D4 不亮。

选择实验模式 1，Clock0 接 4Hz，键 8 控制 RST1，键 7 控制 S0，发光管 1、2、3、4 分别显示 T1、T2、T3、T4 的输出电平。当键 7 控制 S0 输出低电平时，选择的运行模式为单步运行。此时按下键 8 使输出为低电平时，出现一个负脉冲，此时发光管 D1、D2、D3、D4 分别显示 T1、T2、T3、T4 的输出电平一次，当按下键 8 使输出为高电平时，发光管 D1、D2、D3、D4 不亮。当键 7 控制 S0 输出高电平时，选择的运行模式为连续运行。当按下按键 8 使 RST1 为低电平时，T1 输出为“1”，而 T2、T3、T4 输出为“0”，此时 D4、D3、D2、D1 不亮；当按下键 8 使 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，体现为 D4、D3、D2、D1 轮流亮光。

实验四 微控制器实验

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](#)从程序计数器和内部总线获得数据的实现方法。

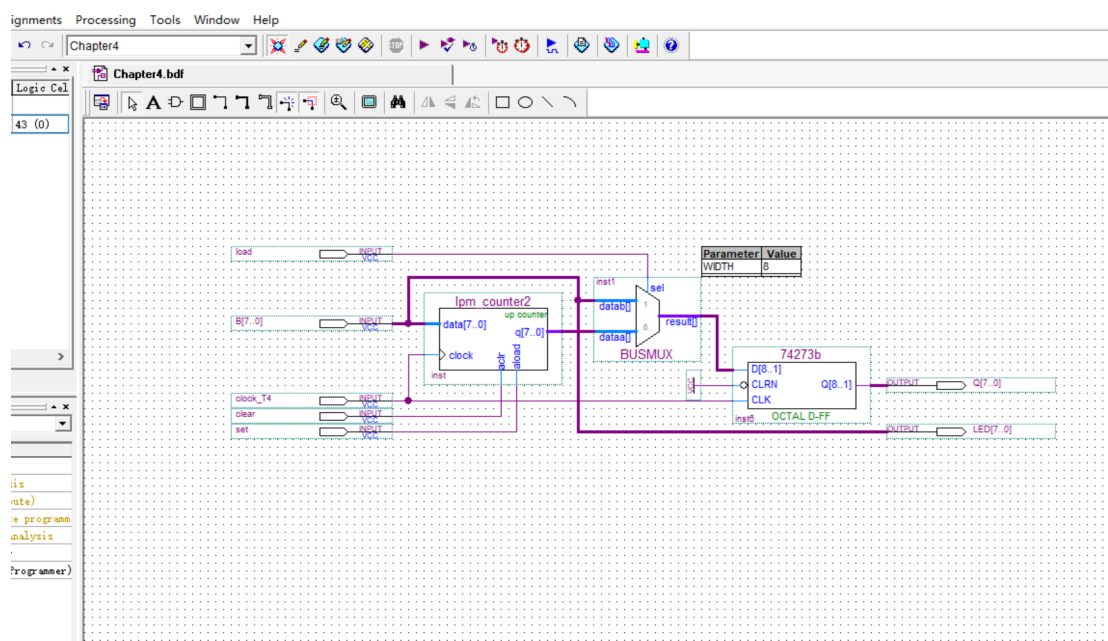
二、实验环境

线上，quartus2 软件。

三、实验基本原理及步骤

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，如电路图 4-1 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。

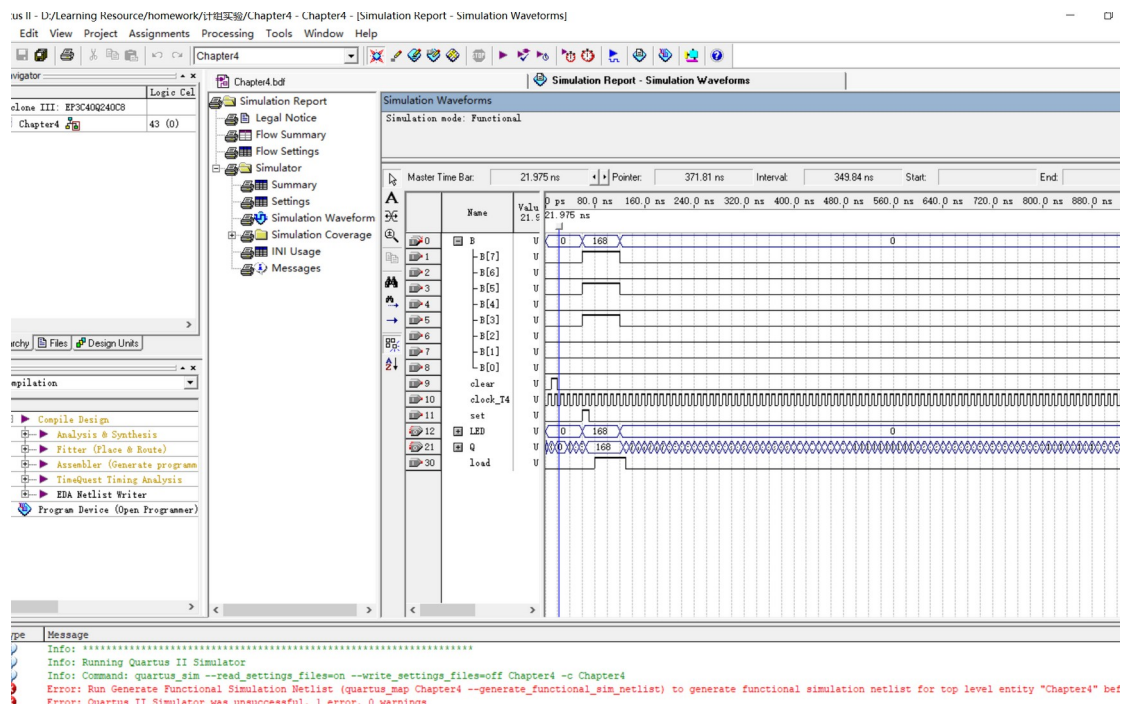


地址寄存器 AR（74273）锁存访问内存 SRAM 的地址。273 中的地址来自两个渠道。一是程序计数器 PC 的输出，通常是下一条指令的地址；二是来自于内部数据总线的数据，通常是被访问操作数的地址。

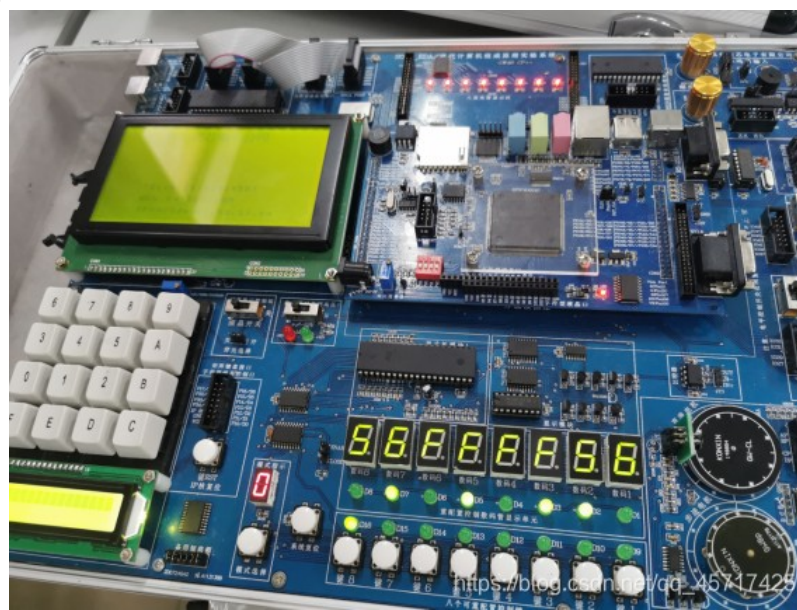
为了实现对两路输入数据的切换，在 FPGA 的内部通过总线多路开关 BUSMUX 进行选择。LDAR 与多路选择器的 sel 相连，当 LDAR 为低电平，选择程序计数器的输出；当 LDAR 为高电平时，选择内部数据总线的数据。

四、实验结果

实验结果之 PC_AR 工作仿真波形



实验箱实物图



五、实验结果分析

(1) 首先进行清零操作：将键 1-键 8 置 0，键 5 置 1，再按一次键 8 即可将 PC 清零；

(2) 连续按键 8（此时保持键 6 为 0，代表选通 PC），可以从数码管 8、7 上看到 AR 的输出（即 PC 值）在不断地自增 1（一般是取指操作），对应地，在 CPU 中可以实现顺序访问地址；

(3) 键 6 置 1，表示选通直接输出总线上的数据作为 AR 值（一般是取操作数操作），按下键 8，产生一个时钟脉冲的上升沿，可看到 AR 的输出（显示在数码管 8、7 上）即为总线上的数据值，值得注意的是，此时并未把地址加载到 PC 中！；

(4) 键 6 置 0，选通 PC 计数器输出，此时用键 2、1 输入一个自定义的十六进制数据至总线上，按键 7 产生一个上升脉冲，代表 LDPC 将总线数据加载至 PC；

(5) 在 (4) 的条件下，连续按动键 8，可以发现观察到 AR 中的数据值在 (4) 中输入的总线数据的基础上不断加 1。

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、实验环境

线上，quartus2 软件。

三、实验基本原理及步骤

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。

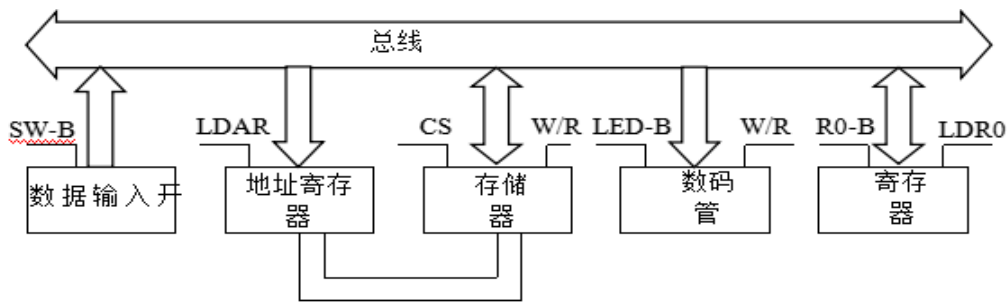


图 5-1 总线实验传输框图

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。

(1) 实验电路如图 5-2 所示。写使能 $WE=1$ 允许写， $WE=0$ 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、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。

（4）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 的地址端；

3、按键 2、键 1，输入数据 28H（数据），此时直接进入总线 BUS，并进

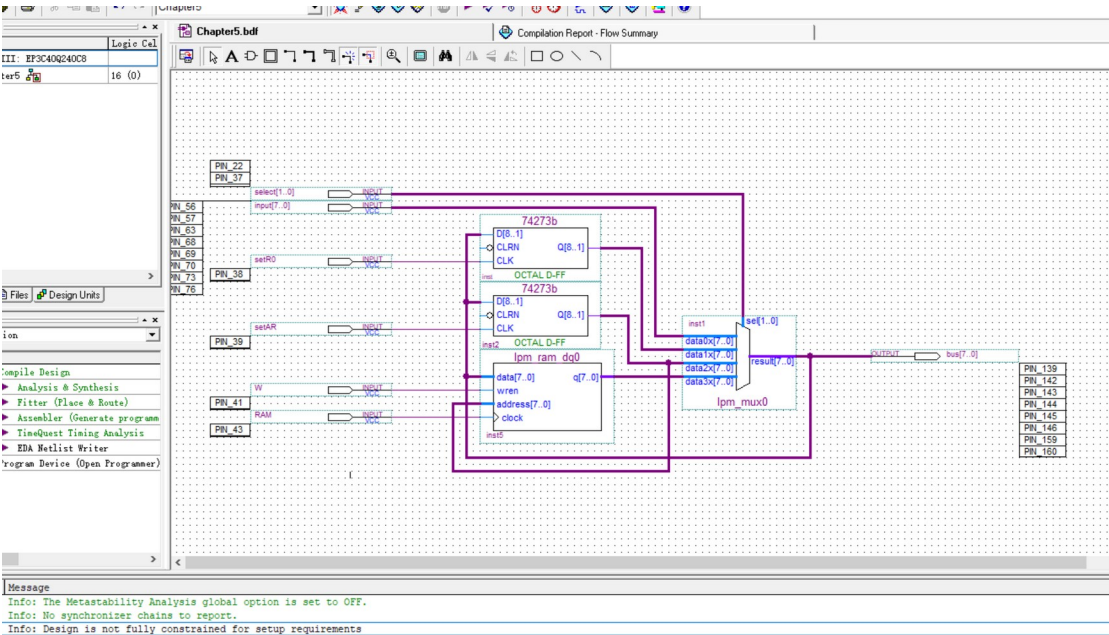
入 RAM 的 data 数据端；按键 8=1（RAM 写允许）；按键 7 两次，将数 28H 写入 RAM（地址为 4AH），最后按键 8=0，写禁止，读允许。

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；

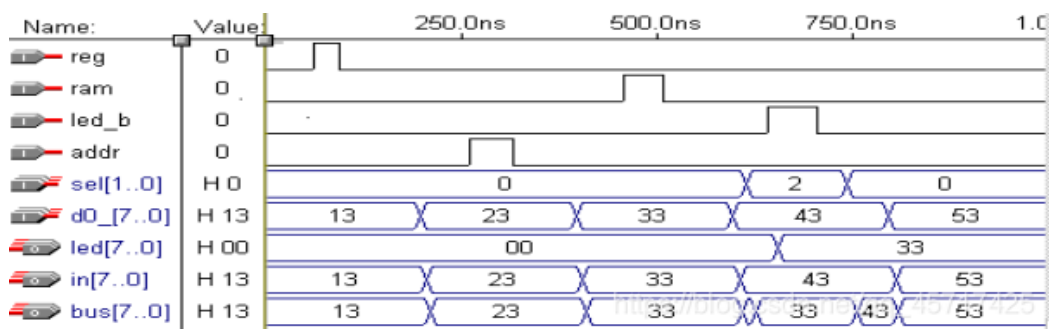
6、按键 7 两次，RAM 中 4AH 单元中的数据 28H 输出，再使键 4、键 3=10，即总线多路选择器 sel[1...0]=10，此时 RAM 数据口的 28H 进入总线 BUS（可从数码管 8、7 上看到）；

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。



四、试验结果

总线控制的时序仿真波形图



用实验箱验证实验结果



五、实验结果分析

本次实验中我验证了实验内容 1，具体操作步骤如下：

- (1) 选择实验模式“0”并置键 8 为高电平（允许 RAM 写入）；
- (2) 利用键 4、键 3 控制设备选择端并置键 4、键 3 为低电平；
- (3) 由键 2、键 1 输入数据 26H（显示于数码管 2、1），该数据直接进入总线（由数码管 8、7 显示）；
- (4) 键 5 发送一次正向脉冲，将数据打入寄存器 R0；
- (5) 键 2、键 1 输入数据 37H，键 6 发送一次正向脉冲，将数据打入地址

寄存器 AR;

(6) 键 2、键 1 再输入数据 48H, 键 7 发送一次正向脉冲, 将数据写入 RAM (此时必须键 8 输出高电平)。由数据锁入规则可以知道此时进入 RAM 的数据 48H 存储在地址为 37H 的存储单元中。

用实验箱验证实验结果图如下: