

实验2 计算机基本部件与总线 控制实验 (第4次课)

2025.10



哈尔滨工程大学计算机实验教学中心

总线传输实验

实验目的

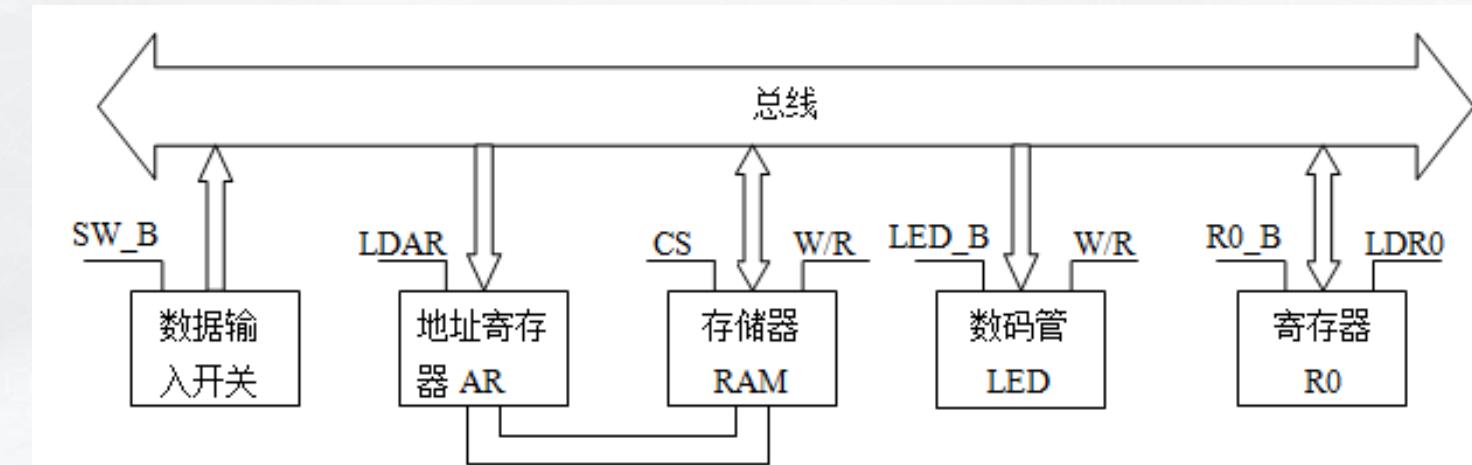
1. 掌握总线电路结构和工作原理。
2. 掌握Quartus Prime软件环境和FPGA实验台的使用方法。
3. 掌握利用框图输入法设计总线电路的方法。
4. 验证总线传输功能。

实验内容

完成总线电路的设计、仿真、编程下载和实验台演示。

总线概念

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。



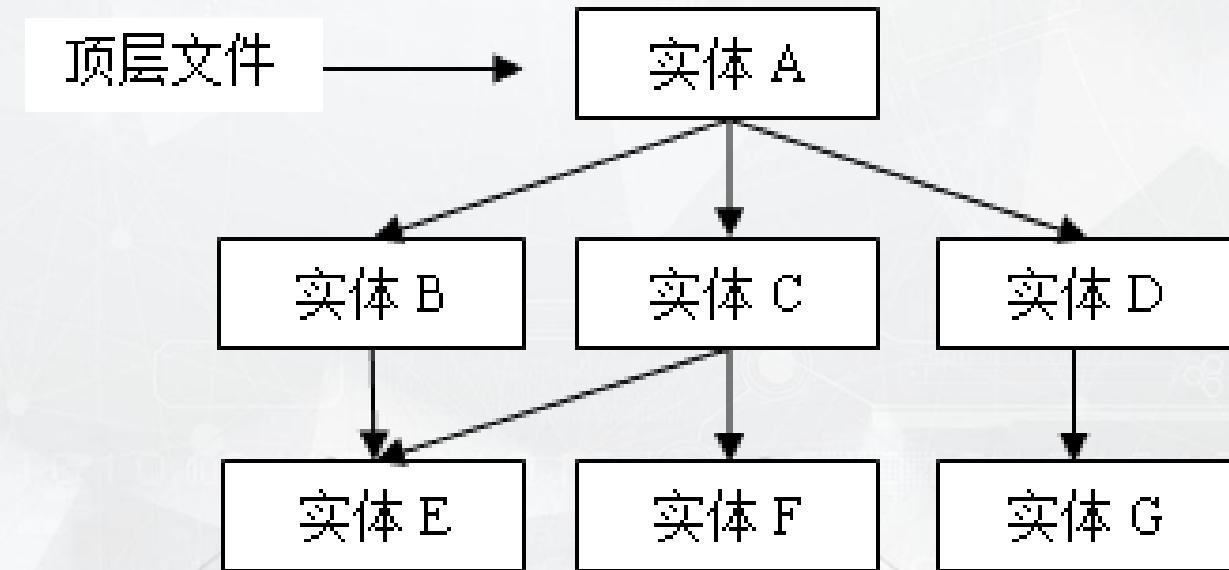
总线传输原理图

总线信息传输功能表

序号	功能	助记符	说明
1	把K1-K8设置的数据写入寄存器R1	IN R1, KEY	
2	把K1-K8设置的数据写入寄存器R2	IN R2, KEY	输入设备→总线→寄存器
3	把K1-K8设置的数据写入RAM某单元	IN RAM, KEY	输入设备→总线→存储器
4	把RAM某单元内容读入寄存器R1	LD R1, RAM	
5	把RAM某单元内容读入寄存器R2	LD R2, RAM	存储器→总线→寄存器
6	把寄存器R1内容写入RAM某单元	ST RAM, R1	
7	把寄存器R2内容写入RAM某单元	ST RAM, R2	寄存器→总线→存储器
8	把寄存器R1内容传到寄存器R2	MOV R2, R1	
9	把寄存器R2内容传到寄存器R1	MOV R1, R2	寄存器→总线→寄存器
10	把寄存器R1内容输出到led显示	OUT LAMP, R1	
11	把寄存器R2内容输出到led显示	OUT LAMP, R2	寄存器→总线→输出设备
12	把RAM某单元内容输出到led显示	OUT LAMP, RAM	存储器→总线→输出设备
13	把K1-K8设置的数据直接输出到led显示	OUT LAMP, KEY	输入设备→总线→输出设备

层次化设计方法

层次化设计的核心思想是“模块化”和“元件复用”。模块化是将一个数字系统划分为几个模块，每个模块可由更小的模块实现。



总线传输实验内容与要求

- ✓ 在Quartus Prime软件中，利用框图设计总线传输电路。总线上的设备有存储器、输入设备、输出设备、数据寄存器和地址寄存器。
- ✓ 完成仿真，仿真要求：
 1. 向RAM存储器中若干存储单元中依次写入数据，然后依次从这些存储单元中读出数据，并在输出设备上显示。
 2. 按照总线信息传输功能表，验证总线传输功能。
- ✓ 完成引脚锁定、在实验台上演示。

选做实验

RAM地址	RAM中存储的程序代码
00	00
01	10
02	0A
03	20
04	0B
05	30
06	0B
07	40
08	01
09	
0A	34
0B	

选做实验：在总线上增加指令寄存器IR，完成仿真：

模拟模型机执行程序时RAM地址的变化。

程序运行时，RAM地址的变化是：00、01、02、0A、03、04、0B、05、06、0B...。

从RAM读取的指令（00、10、20、30、40）送往总线，在送入指令寄存器IR。

实验任务与步骤

1、新建工程，新建框图文件 (*.bdf) ，设计输入总线电路图。添加自定义8位寄存器元件，采用元器件库中 lpm_ram_dq和lpm_mux。保存文件。

主菜单“File”→“New Project Wizard”，新建工程BUS（实体名）
主菜单“File”→“New”项，选择Block Diagram/Schematic File，新建框图文件，保存为 BUS.bdf。

2、设置器件

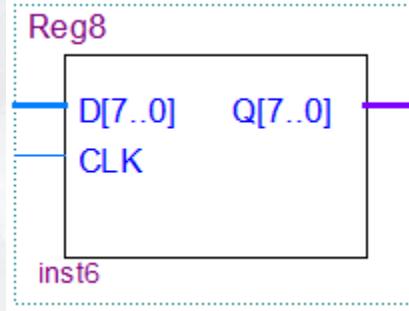
主菜单“Assignmemts”→“Device”项，选择Cyclone IV E系列
EP4CE55F23C8芯片

3、编译电路

主菜单“Processing”→“Start Compliation”项，启动编译

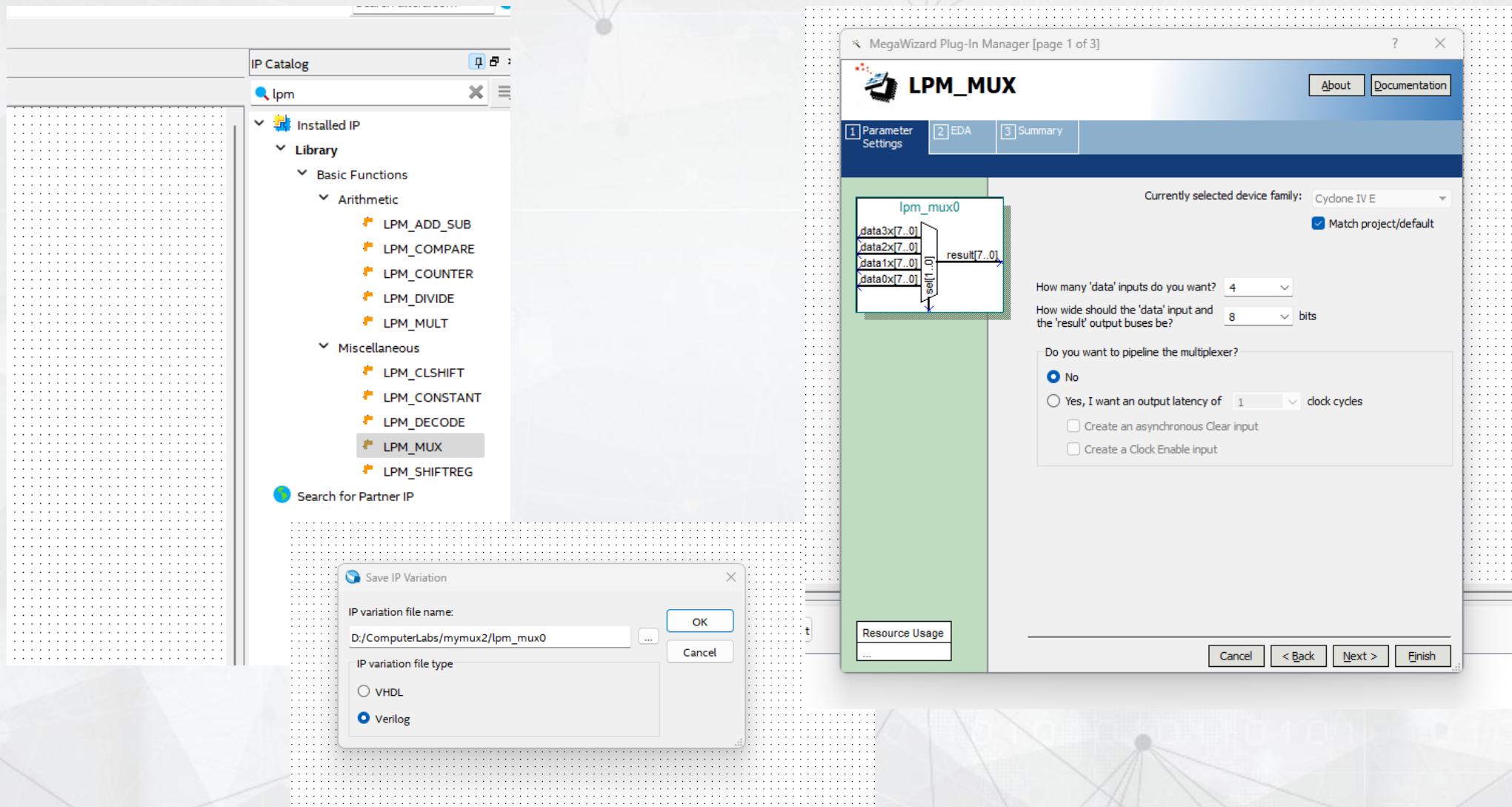
自定义8位寄存器元件

- 1.利用框图设计位寄存器电路， 电路设计文件Reg8.bdf， 将Reg8.bdf拷贝到总线工程目录
- 2.主菜单“File”→“Create/Update”项，
选择“Create Symbol Files for Current File”
由Reg8.bdf生成Reg8.bsf， 即生成自定义8位寄存器元件符号
- 3.在元器件库中，在Project目录下选择自定义元件Reg8，加入到总线电路图中

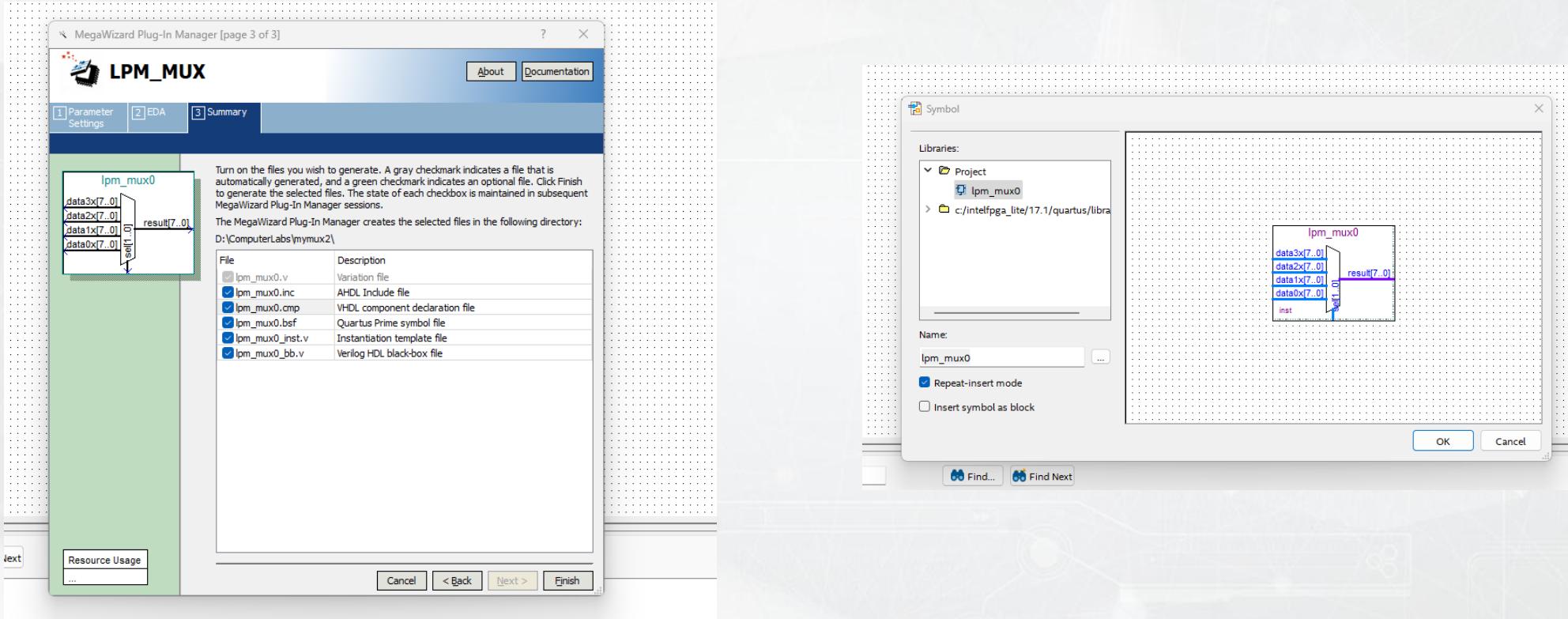


数据寄存器Reg8可以暂存8位数据。当 CLK上升沿到来时，输出端Q输出输入端D的值。即 $Q[7..0]=D[7..0]$

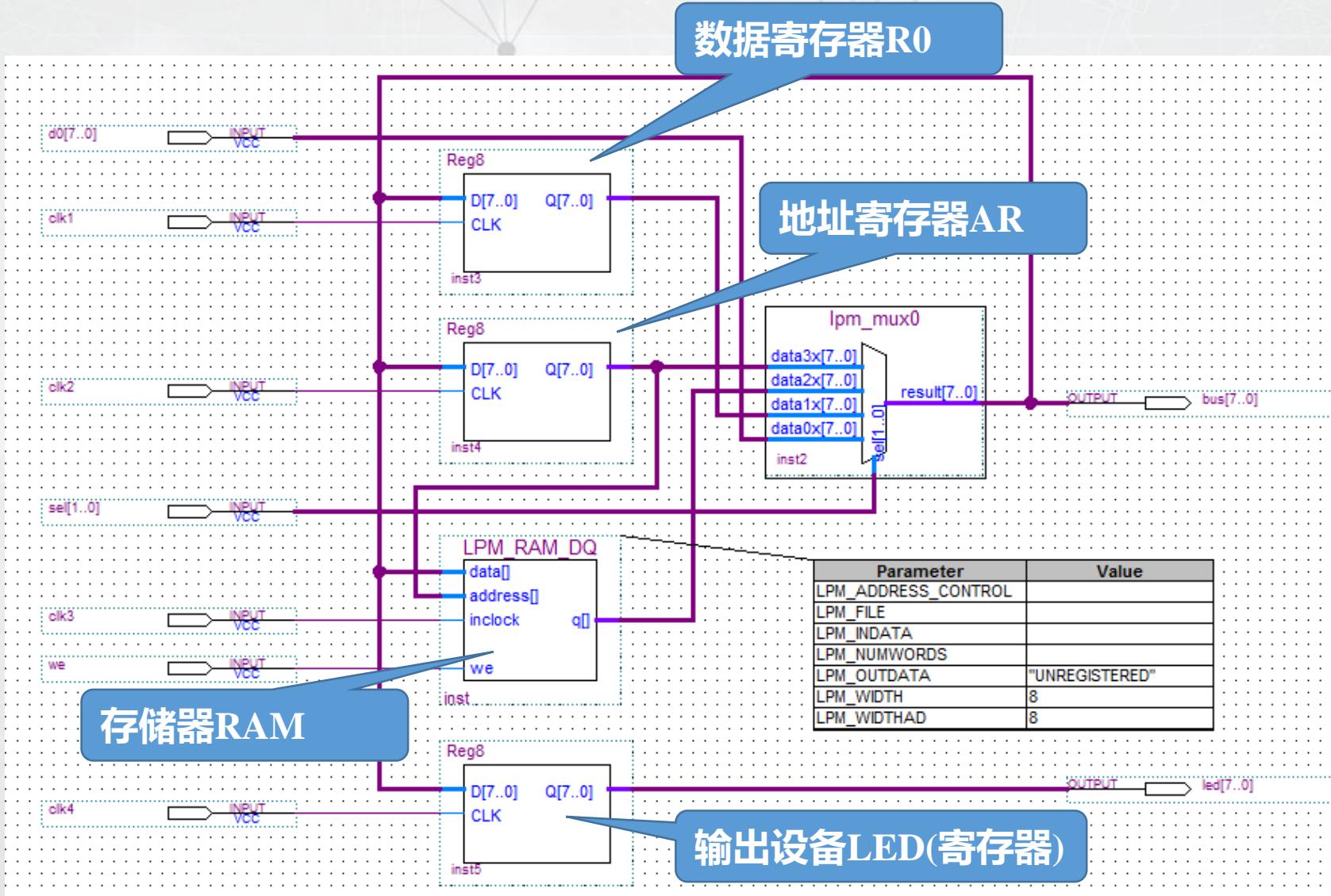
加入多路选择开关



加入多路选择开关



采用层次化设计方法设计总线电路



实验任务与步骤

4、新建波形图文件 (*.vwf), 设置仿真时间, 添加输入输出端口, 设置输入信号值, 保存文件。运行仿真。

建立仿真波形文件：主菜单“File”→“New”项，选择University Program VWF，新建*.vwf，打开波形编辑器。

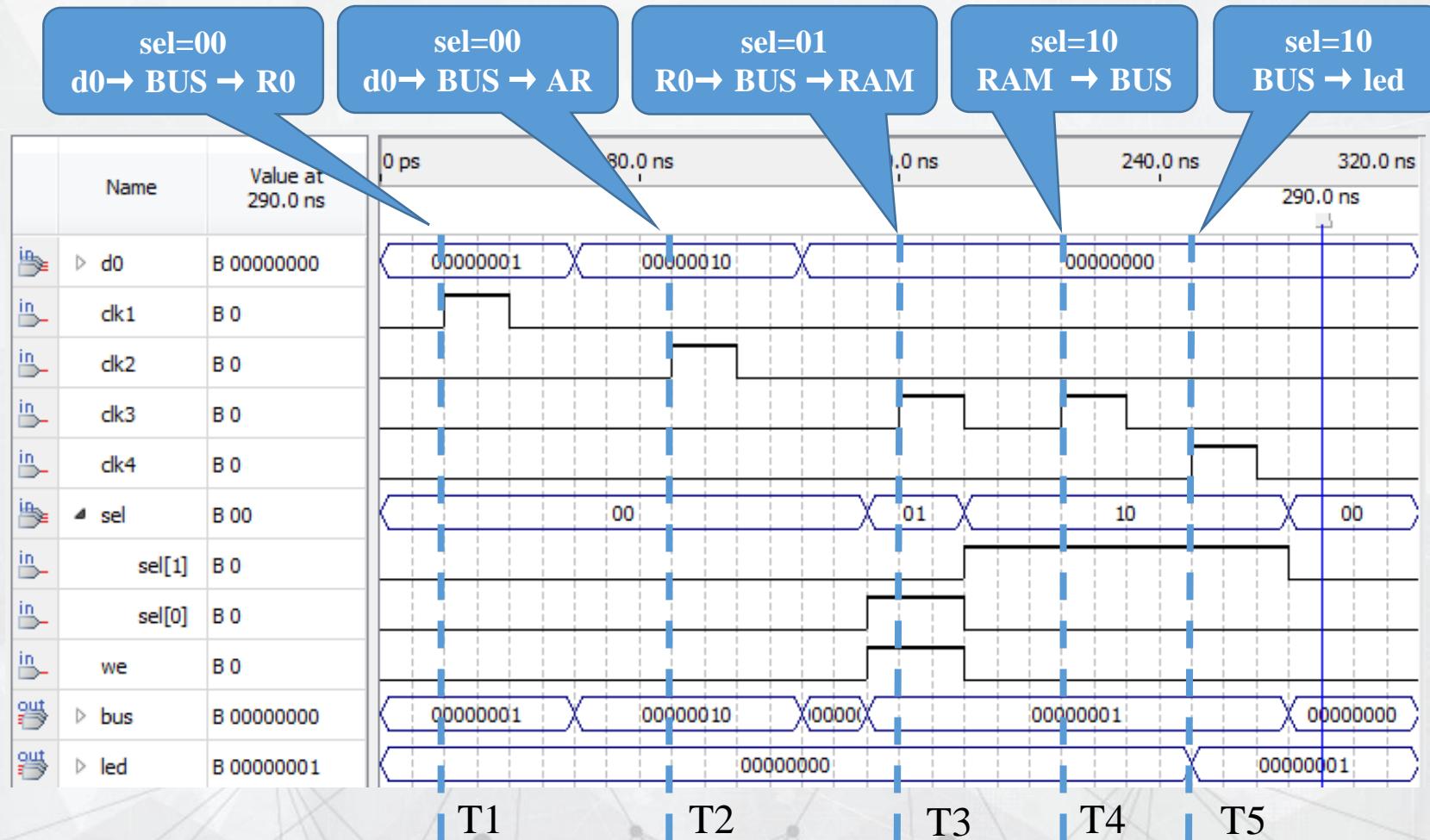
设置仿真时间：主菜单“Edit”→“Set End Time”项。

添加输入输出端口：波形编辑器窗口主菜单 “Edit” → “Insert”→“Insert Node or Bus”

运行仿真：波形编辑器窗口主菜单“Simulation”→“Run Functional Simulation”项。

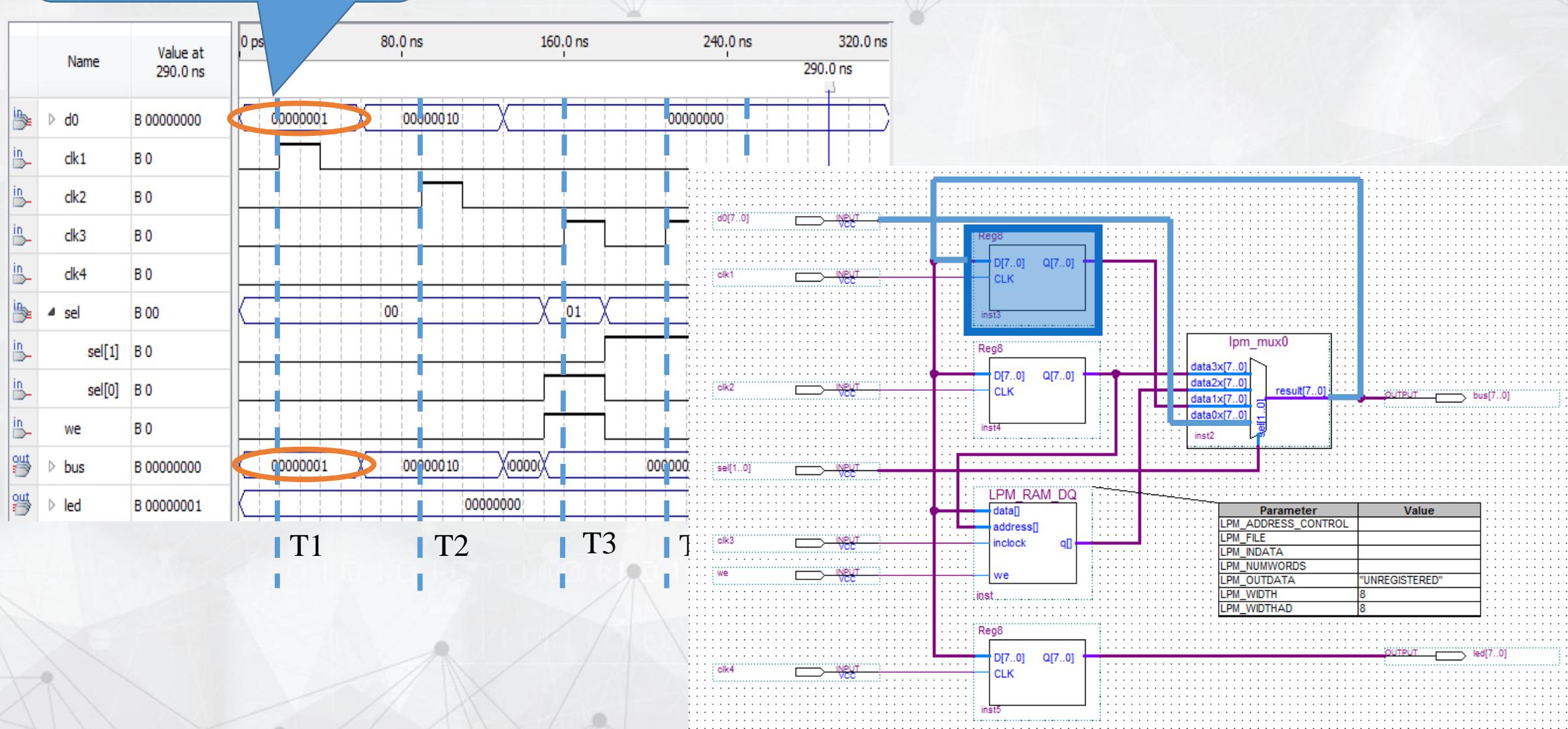
总线传输仿真波形示例

将数据01在写入地址为10的RAM存储单元，并输出到输出设备led上。仿真时间为320ns。



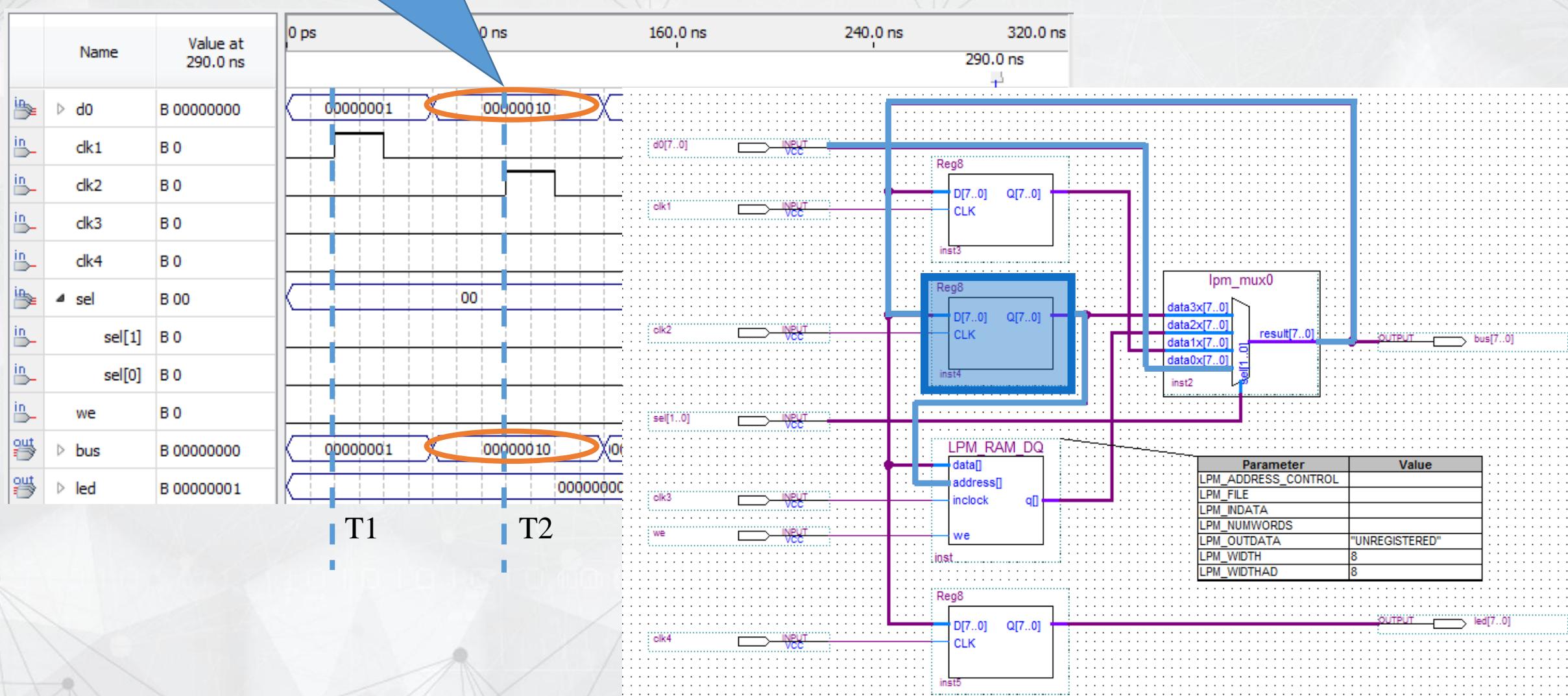
sel=00
d0→ BUS → R0

T1时刻：输入设备→总线→R0寄存器



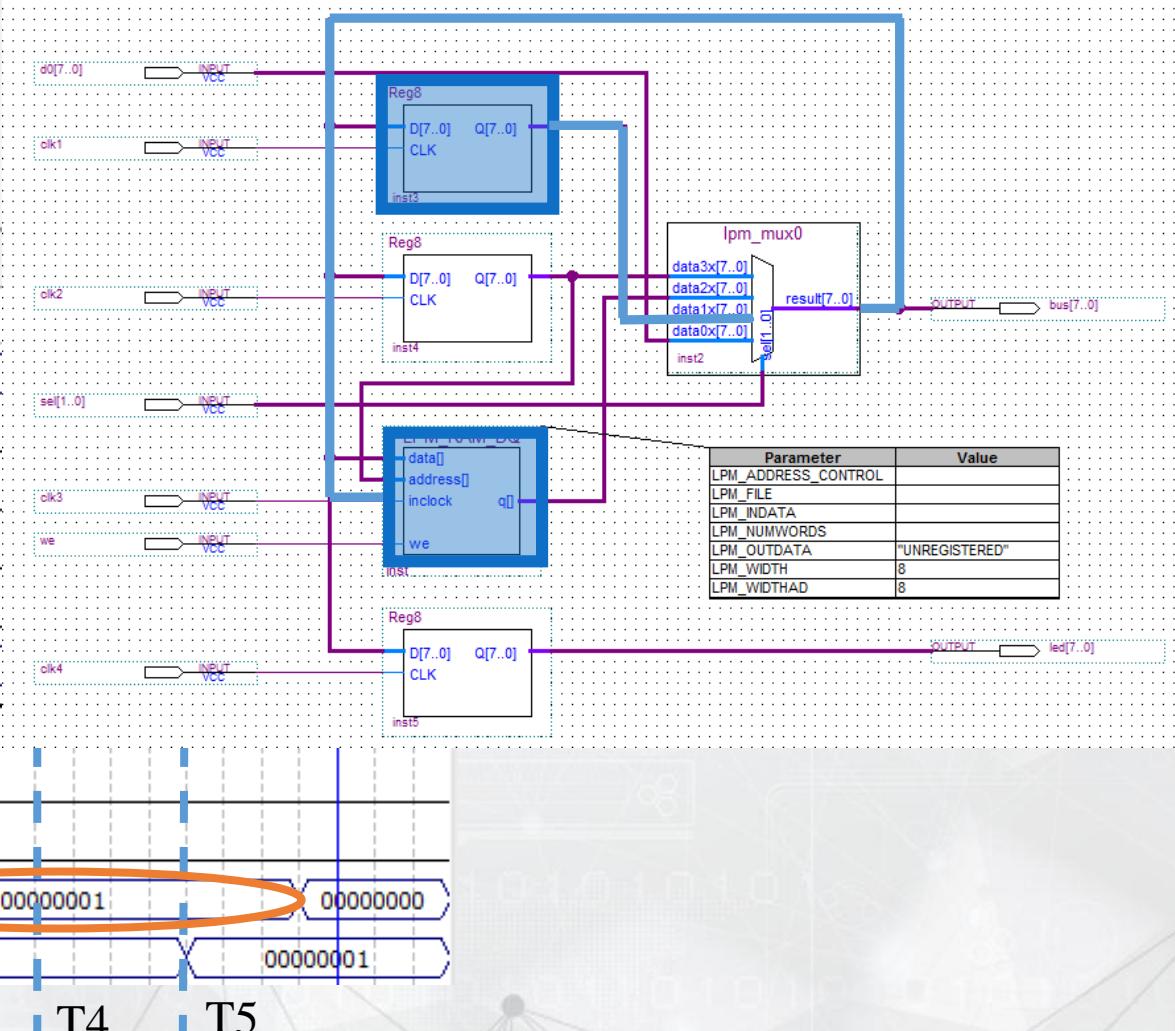
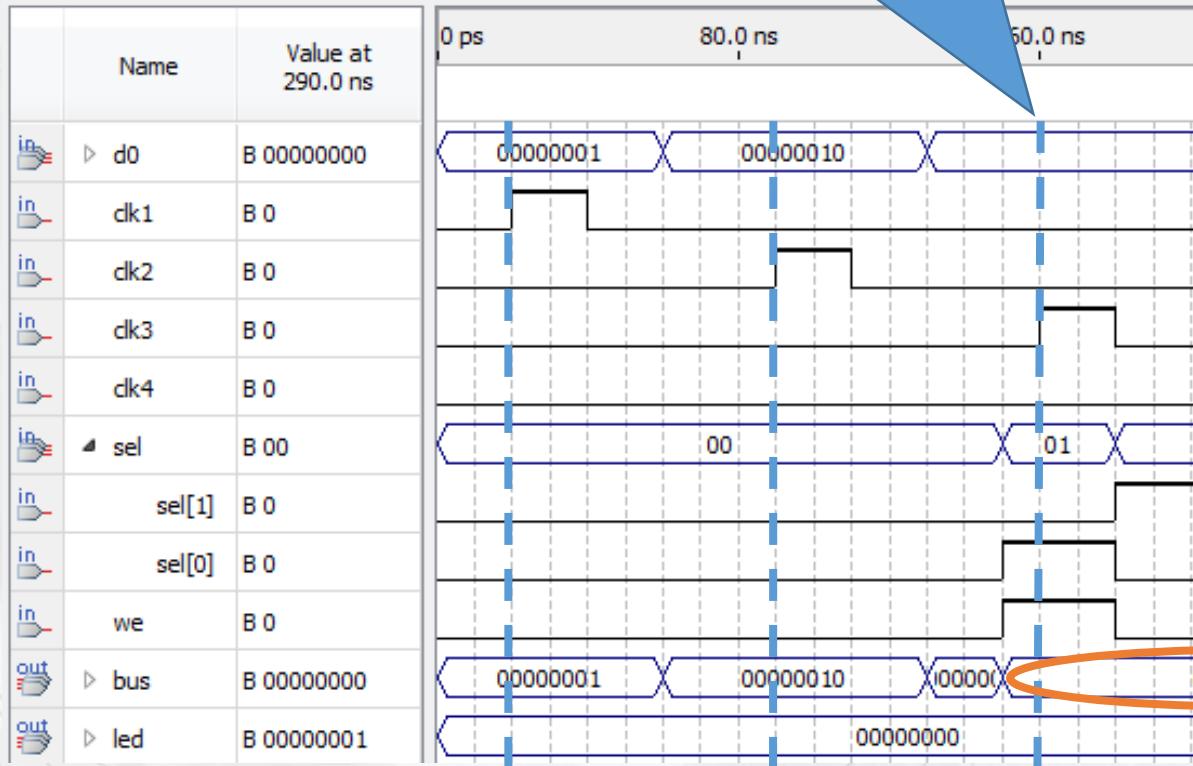
sel=00
d0 → BUS → AR

T2时刻：输入设备d0→总线→AR寄存器



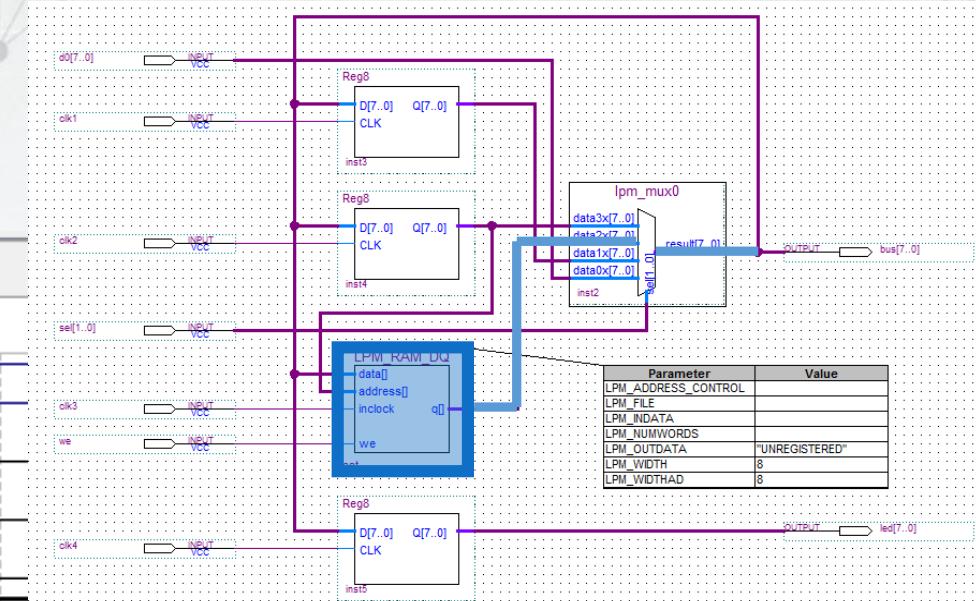
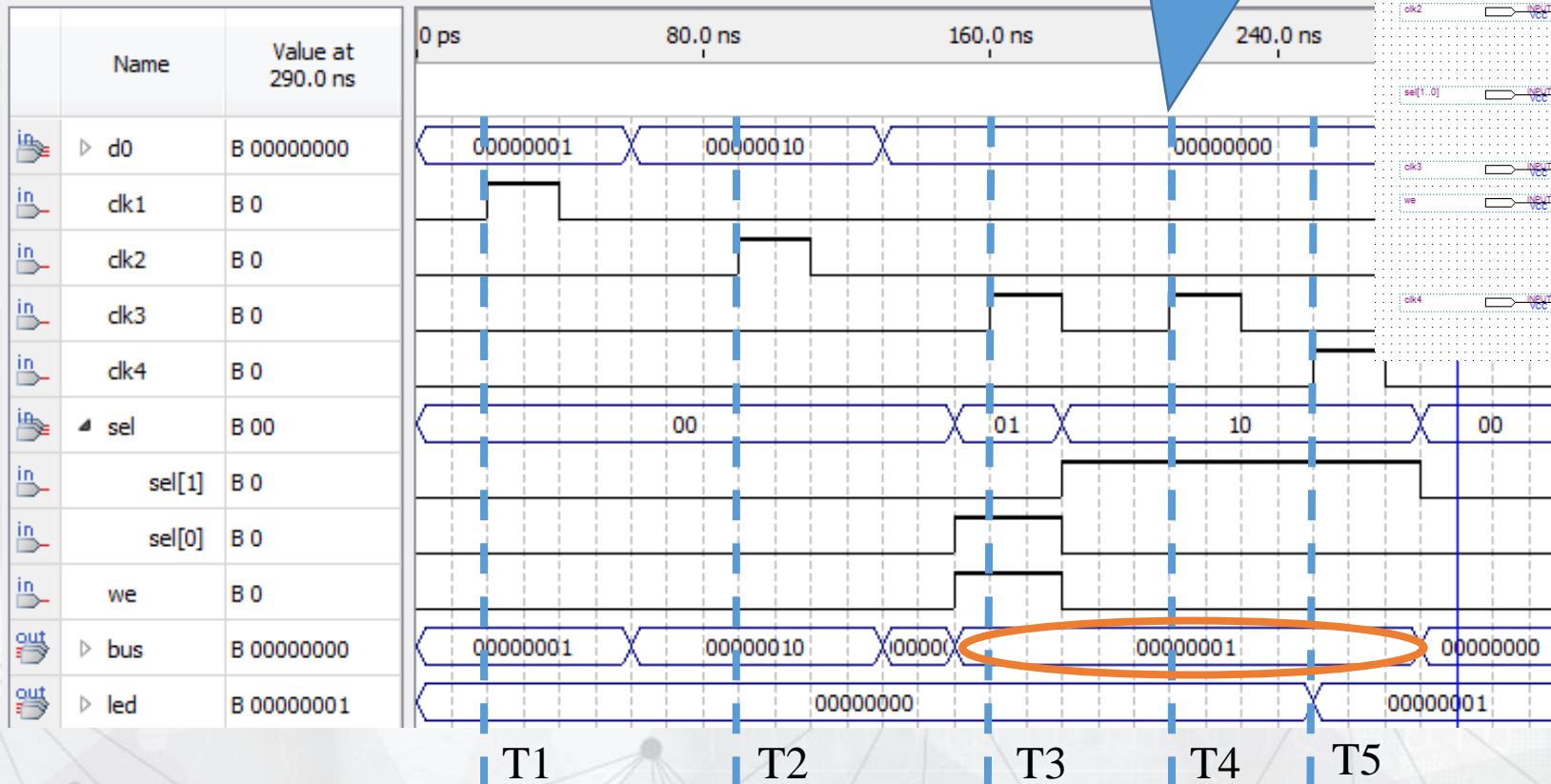
T3时刻： R0寄存器→总线→存储器RAM

sel=01
R0 → BUS → RAM

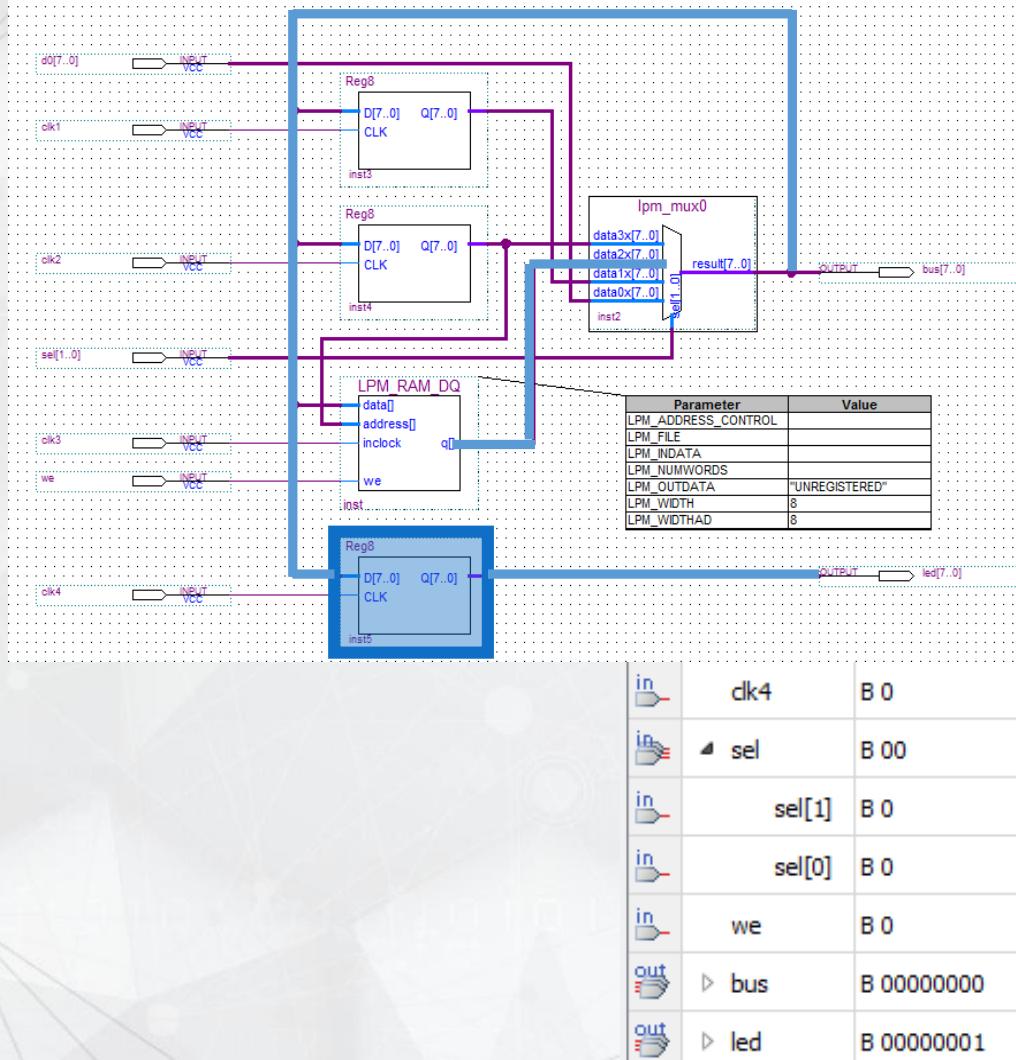


T4时刻：存储器RAM→总线

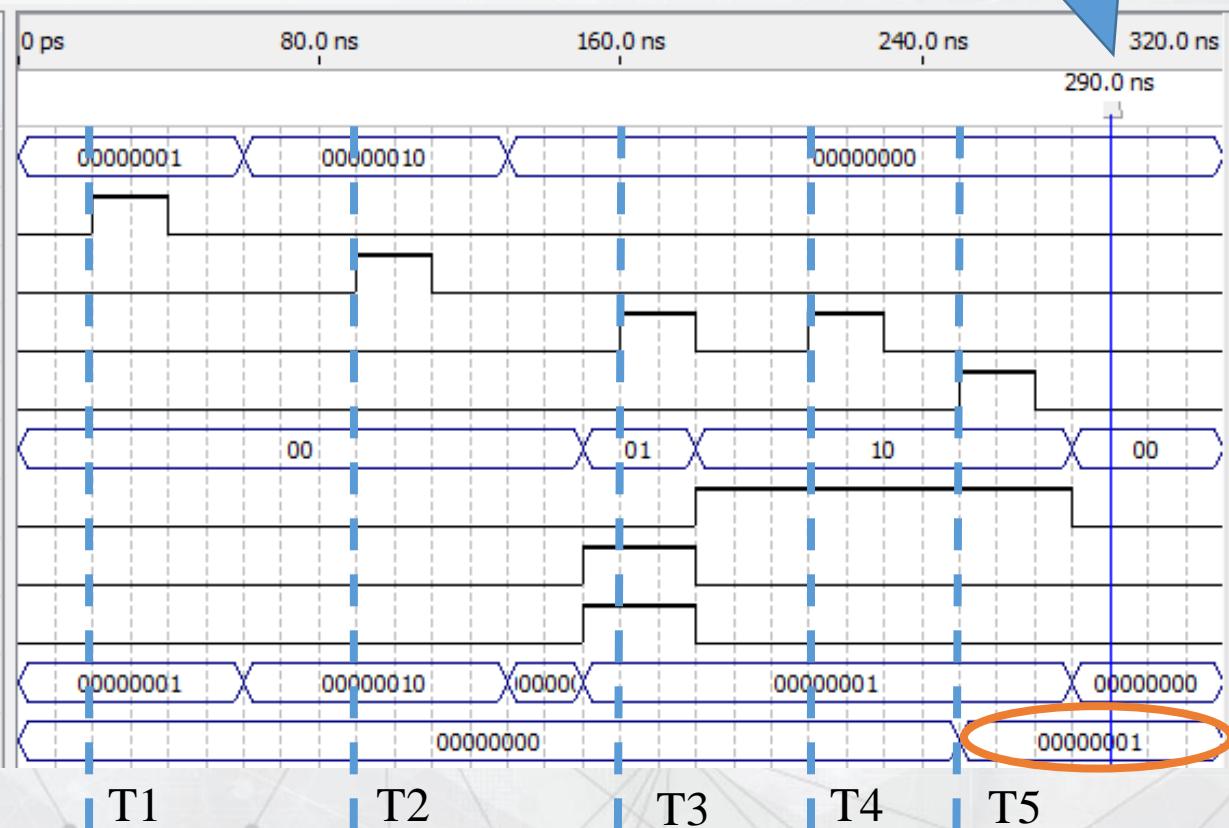
sel=10
RAM → BUS



T5时刻：存储器RAM→总线→输出设备led



sel=10
RAM → BUS → led



实验任务与步骤

5、选择KX-CDS实验台，选择合适的电路模式结构，例如NO.1，对照电路模式图和引脚表，查找引脚号。打开编程器，输入引脚号，对电路进行引脚锁定，编译工程。

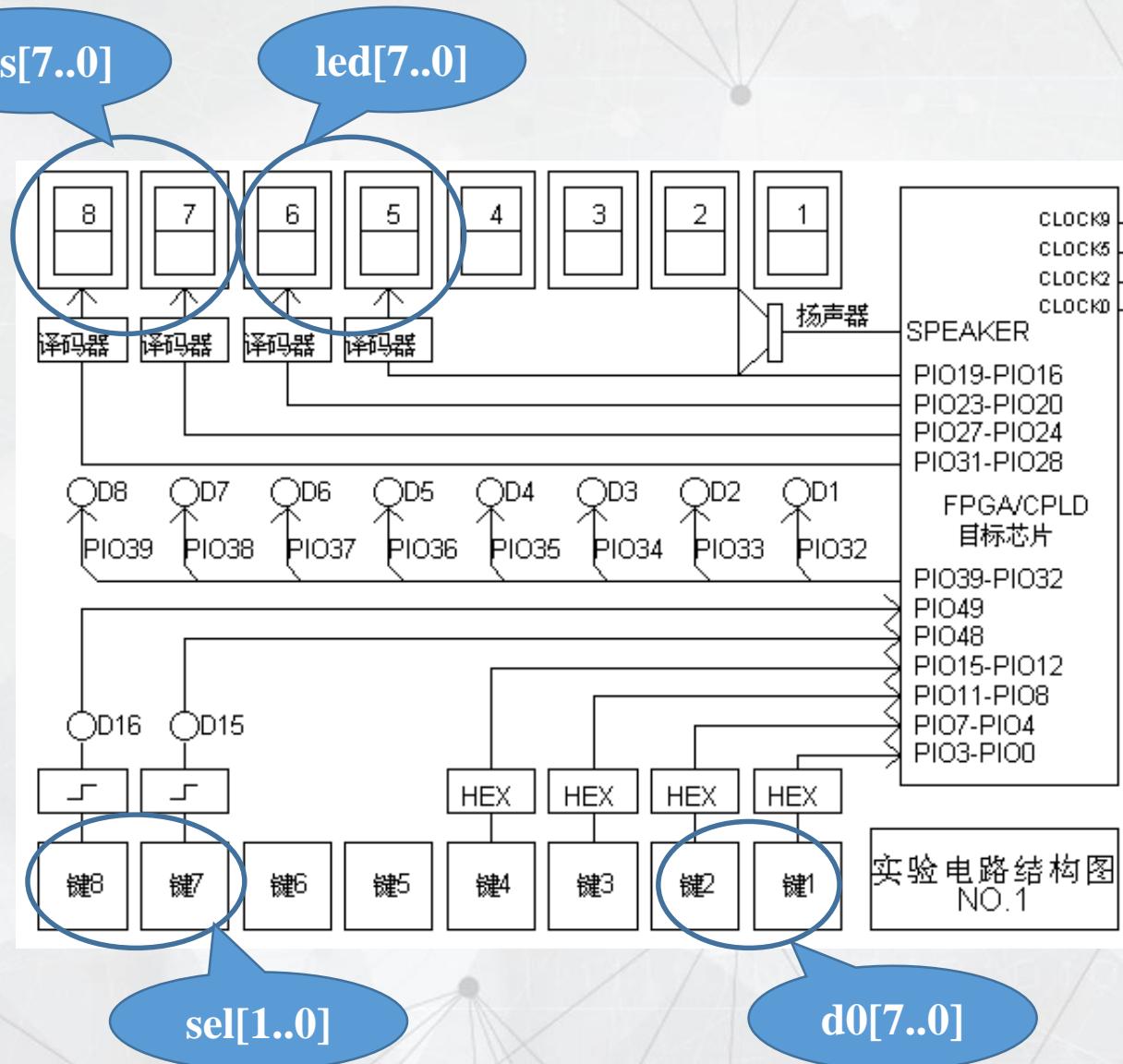
主菜单“Assignments”→“Pin”项，在Location栏中输入引脚号

6、下载sof文件到FPGA实验台，演示总线传输的功能。

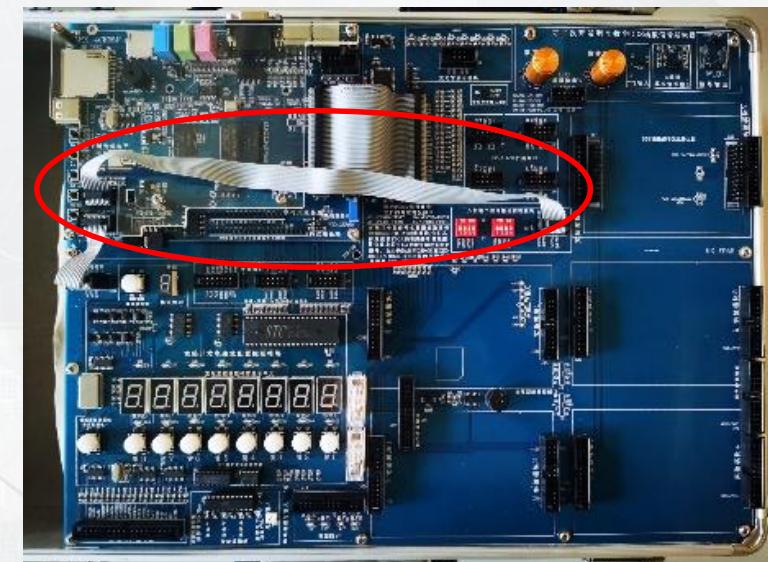
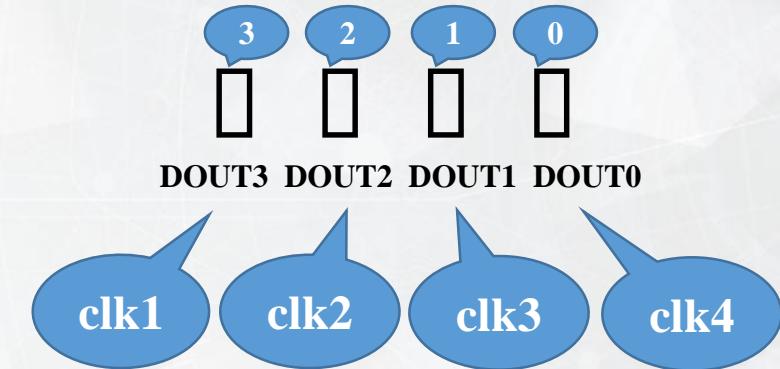
主菜单“Tools”→“Programmer”项，打开编程器，设置硬件，连接实验台。

**在Programmer窗口，点击Start按钮，Progress为100%时，
下载完毕**

FPGA实验台测试与演示 KX-CDS实验台



扩展板拨码开关



扩展板拨码开关需要连线

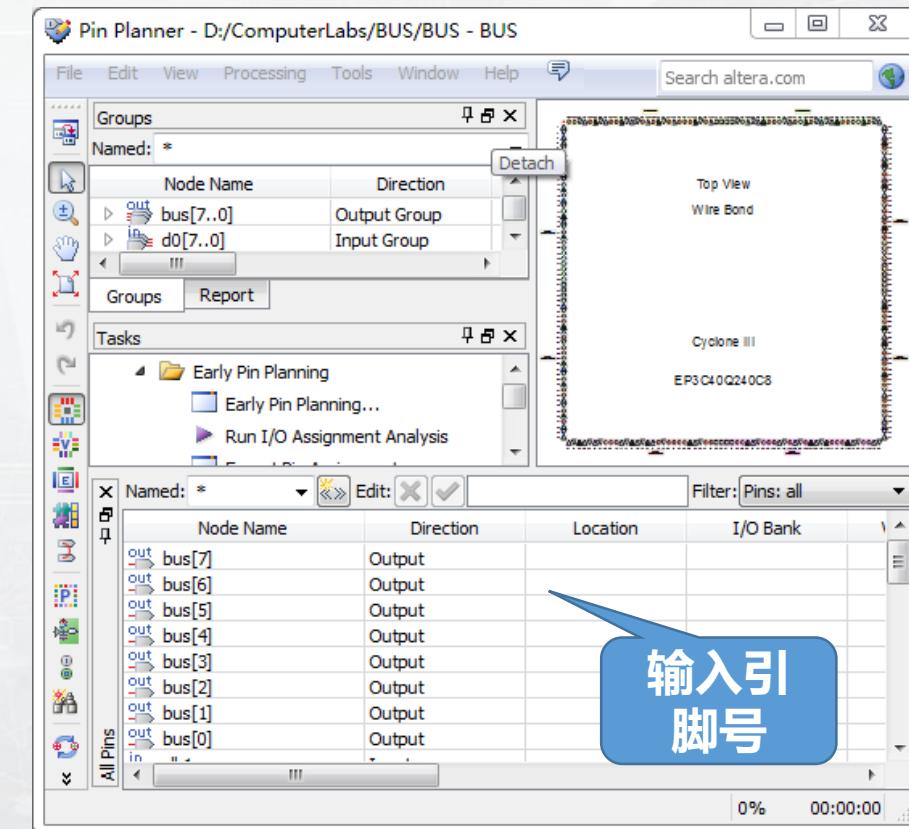
FPGA实验台引脚锁定

参照电路模式图No.1，确定引脚名称，再查找引脚表（QQ群文件），获得引脚号

引脚锁定方案(No.1)

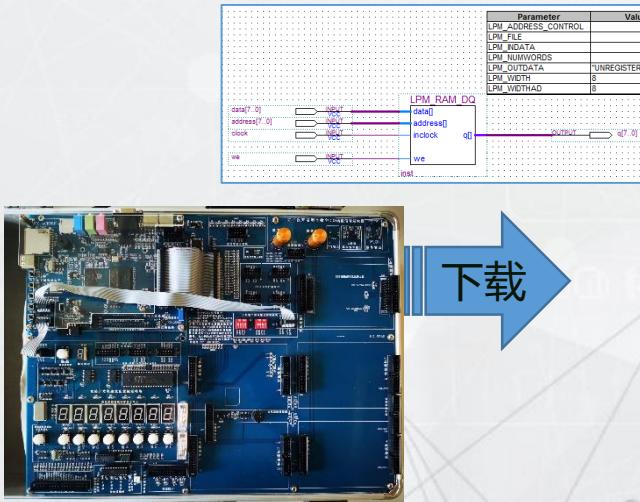
输入/输出端口	外设	引脚名称	引脚号
sel[1]	按键8		
sel[0]	按键7		
clk4	拨码开关1		
clk3	拨码开关2		
clk2	拨码开关3		
clk1	拨码开关4		
we	拨码开关5		
d0[7..4]	按键2		
d0[3..0]	按键1		
bus[7..4]	数码管8		
bus[3..0]	数码管7		
led [7..4]	数码管6		
led [3..0]	数码管5		

主菜单“Assignments”→“Pin”项，在Location栏中输入引脚号



BUS实验台演示

- ✓ 主菜单“Tools”→“Programmer”项，打开编程器，点击“Hardware Setup”按钮，选择USB-Blaser[USB-0]，
- ✓ 注意：实验台需要打开电源，并且将其JTAG接口与计算机通过USB线连接。勾选ProgramConfigure复选框。
- ✓ 在Programmer窗口，点击Start按钮，Progress为100%时，下载完毕。



现在开始实验！（第4次课）

实验2 计算机基本部件与总线控制实验

1、总线传输实验：参考教材6.5

实验台下载：参考教材2.3.5

KX-CDS型号FPGA实验台使用方法.pptx

2、完成电路设计、编译和仿真。要求向RAM连续写入若干数据，再依次读出数据，输出到led端。（全部数据写入RAM之后再读RAM）

3、选做内容：**利用Verilog语言实现各个电路，完成编译与仿真。**

4、**2个人一组。** 实体名后面加2个学号的后两位，例如mux21a0709

5、下次课预习：

实验3 基本模型机系统设计实验

时序信号发生器实验 参考教材278页-294页

程序计数器与地址寄存器实验 参考教材271-273

微程序控制器实验 参考教材278页-294页