

# 总线数据传输实验日志

## 一、实验代码

```
1  library IEEE;
2  use IEEE.std_logic_1164.all;
3  entity exp_bus is
4  port(clk in STD_LOGIC;
5  sw_bus,r1_bus,r2_bus,r3_bus in STD_LOGIC;
6  k in STD_LOGIC_VECTOR(7 downto 0);
7  lddr in STD_LOGIC_VECTOR(3 downto 1);
8  l in out STD_LOGIC_VECTOR(7 downto 0));
9  end exp_bus;
10 architecture rtl of exp_bus is
11 signal r1,r2,r3,bus_Reg:STD_LOGIC_VECTOR(7 downto 0);
12 begin
13 lddr<=process(clk,lddr,bus_Reg)
14 begin
15 if clk'event and clk='1' then
16 if lddr(1)='1' then r1<=bus_Reg;
17 elsif lddr(2)='1' then r2<=bus_Reg;
18 elsif lddr(3)='1' then r3<=bus_Reg;
19 end if;
20 end if;
21 end process;
22 bus_Reg<=k when (sw_bus='0' and r1_bus='1' and r2_bus='1' and r3_bus='1') else
23 r1 when (sw_bus='1' and r1_bus='0' and r2_bus='1' and r3_bus='1') else
24 r2 when (sw_bus='1' and r1_bus='1' and r2_bus='0' and r3_bus='1') else
25 r3 when (sw_bus='1' and r1_bus='1' and r2_bus='1' and r3_bus='0') else
26 (others=>'0');
27 l<=bus_Reg when (sw_bus='0' or r1_bus='0' or r2_bus='0' or r3_bus='0') else
28 (others=>'Z');
29 end rtl;
```

## 二、实验数据记录

(1) 记录波形仿真参数设置（End time 和 Grid size）。

End Time

Time: 4.0 us

Default extension options:

Extension value: Last clock pattern

End time extension per signal:

Signal Name	Direction	Radix	Extension value
clk	Input	Binary	Default extension value
+ k	Input	Hexadecimal	Default extension value
+ sw_r123	Group	Binary	Default extension value
+ lddr	Input	Binary	Default extension value
+ l	Bidir	Hexadecimal	Default extension value
+ l~result	Group	Hexadecimal	Default extension value

OK

Cancel

Grid Size

Base grid on

☐ Clock settings:

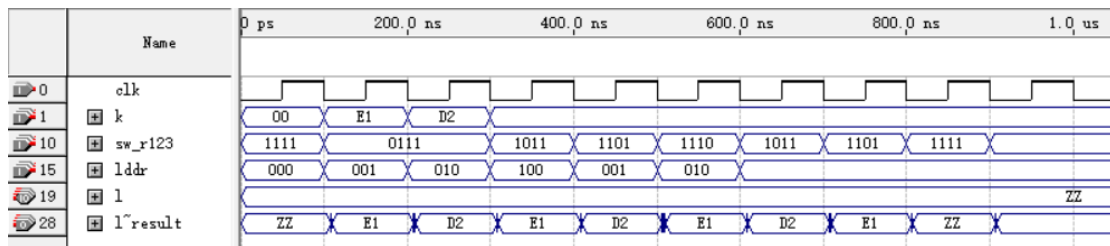
☒ Time period:
 

Period: 100.0 ns

OK

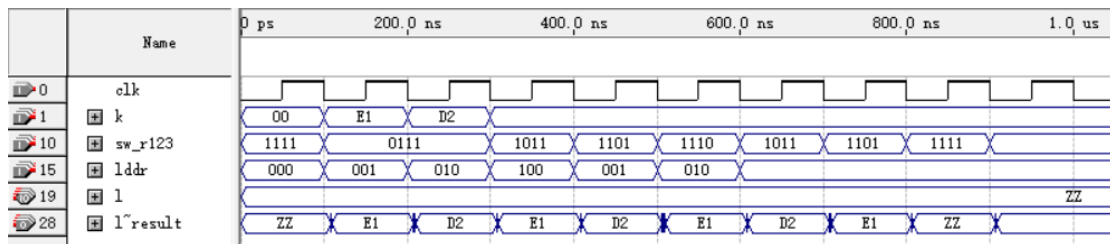
Cancel

- (2) 记录芯片设置及引脚设置。
- (3) 记录电路初始状态时 input 输入信号设置。



k 置为 00, sw\_bus,r1\_bus,r2\_bus,r3\_bus 低电平有效, 置为 1111, lddr 高电平有效, 置为 000.

(4) 记录全部实验内容中出现的数据。



0~100ns,设置初始状态;

100~200ns,将外部输入 E1 传入总线, 再写入 r1;

200~300ns,将外部输入 D2 传入总线, 再写入 r2;

300~400ns,将 r1 数据传入 r3;

400~500ns,将 r2 数据传入 r1;

500~600ns,将 r3 数据传入 r2;

600~700ns,将 r1 数据传入总线;

700~800ns,将 r2 数据传入总线;

800~900ns,操作结束, 停机。

### 三、思考题

1) 总线数据传输的基本特性是什么?

总线的基本特性是不允许挂在总线上的多个部件同时向总线发出信息; 但是, 允许挂在总线上的多个部件同时从总线上接收信息。

2) 从 74374 和 74244 内部电路结构图上说明它们的逻辑功能。

74374 相当于一个八位寄存器, 可以在控制信号的控制下输出 (低电平有效) 和输入 (高电平有效) 数据, 并实现存储; 74244 相当于一个八位三态门, 可以在控制信号 (低电平有效) 的控制下选择的通过数据。

3) 实验电路中 BIDIR 端口的用途是什么?

BIDIR 端口可以实现双向输入, 即既可以在仿真波形中看到总线上的结果, 也可以通过端口向总线输入数据。

4) 举例说明画电路图中连线 bus line(粗线)和 node line(细线)的区别。总线与支线命名方式是什么?

粗线比细线在画图上粗的多, 粗线可以连接传输多位数据, 细线只能传输一位数据。总线的命名方式为 x[n..m], 支线的命名方式为 x[n]...x[m]。

5) 实验需要互换 R1 和 R2 数据, 但是电路图中 R3 的连线有问题, 错在哪里? 为什么?

R3 的输出没有接入总线中，所以没法借助 R3 实现数据暂存。也就没有办法实现 R1 寄存器和 R2 寄存器的数据交换。

- 6) exp\_bus.vhd 代码中如何实现双向总线的定义与缓冲？  
通过 inout 设置端口来实现双向输入输出。
- 7) 写出 exp\_bus.vhd 代码中(others=>'Z')的其他描述方式？  
可以将 when 语句修改成 if else 语句来实现，如：else bus\_Reg<="ZZZZZZZZ"
- 8) 编写 VHDL 时如何实现多路选择器？  
可以通过 if 和 elsif 语句实现多路分支，分别对应不同的情况执行不同的语句来实现多路选择的效果。
- 9) 编写 VHDL 代码时如何为寄存器赋初值？  
在声明中间变量的时候在变量类型后面通过“=”进行赋值即可。
- 10) 结合 4.1.6 思考题（3,4,6），试述如何构造总线电路？  
通过粗线作为总线连接多个寄存器，并将总线引出接到 BIDIR 端口来观察总线值以及输入数据。