



XAPP491 (v1.0) 2006 年 10 月 4 日

Spartan-3 FPGA 系列中高效 PCB 布局的 LVDS 信号倒相

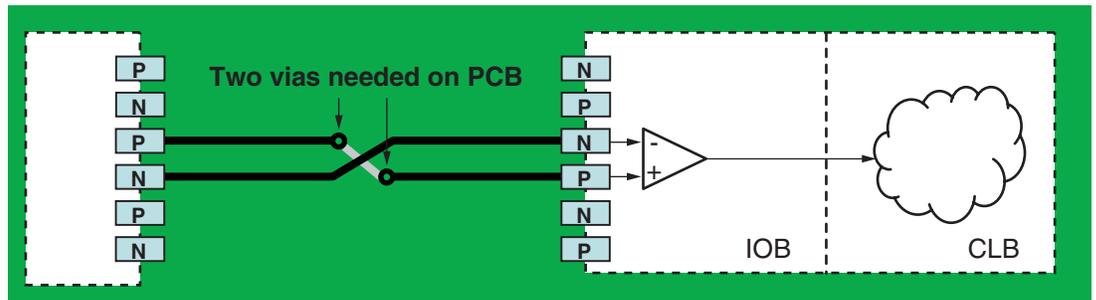
作者：Nick Sawyer 和 Gary Lawman

提要

在比较简单的未大量使用过孔的四层或六层 PCB 上，可能很难对 LVDS 或 LVPECL 这类差分信号布线。其原因是，驱动器上的正极引脚必须驱动接收器上的相应正极引脚，而负极引脚则必须驱动接收器的负极引脚。有时迹线以错误的方向结束，这实际上是向电路中添加了一个倒相器。本应用指南说明 Spartan™-3 FPGA 系列如何仅通过在接收器数据通路中加入一个倒相器即可避免大量使用过孔，并且在不要 PCB 重新设计的情况下即可解决意外的 PCB 迹线交换问题。这项技术同样适用于将 FPGA 用作驱动器的情况，而且交换迹线可使得在其他器件或连接器上的 PCB 布线更容易。

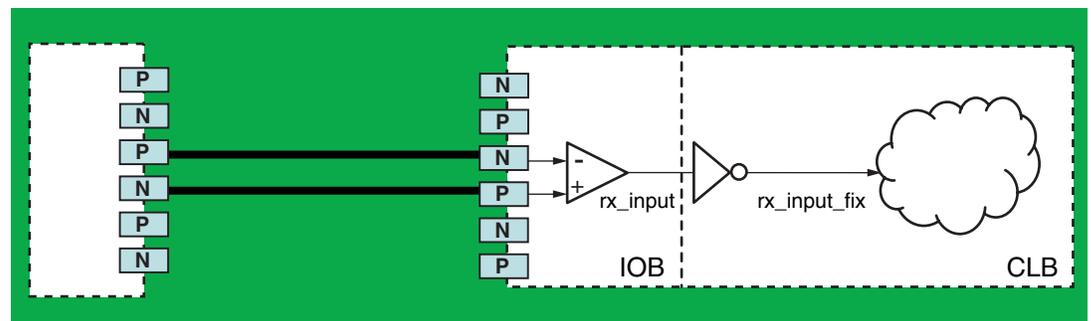
简介

图 1 所示为一个 PCB 示例，其中正极引脚驱动接收器的正极引脚，负极引脚驱动接收器的负极引脚。如果引脚被意外交换，那么 PCB 迹线实际上就变成了一个倒相器，这就可能需要电路板重新设计。



X491_01_041906

图 1: PCB 布局需要使用过孔来交换迹线



X491_02_041906

图 2: PCB 布局不需要使用过孔来交换迹线

图 2 说明 Spartan-3 FPGA 系列如何通过接收器数据通路中加入必要的倒相器来解决这一问题。有此功能，设计人员便可以为简化布线而任意交换迹线。因此，PCB 设计人员可以为实现最大信号完整性而自由布置差分对；所发生的任何交换都可以在 FPGA 内部得到校正。如果使

© 2006 Xilinx, Inc. All rights reserved. All Xilinx trademarks, registered trademarks, patents, and further disclaimers are as listed at <http://www.xilinx.com/legal.htm>. All other trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change without notice.

NOTICE OF DISCLAIMER: Xilinx is providing this design, code, or information "as is." By providing the design, code, or information as one possible implementation of this feature, application, or standard, Xilinx makes no representation that this implementation is free from any claims of infringement. You are responsible for obtaining any rights you may require for your implementation. Xilinx expressly disclaims any warranty whatsoever with respect to the adequacy of the implementation, including but not limited to any warranties or representations that this implementation is free from claims of infringement and any implied warranties of merchantability or fitness for a particular purpose.

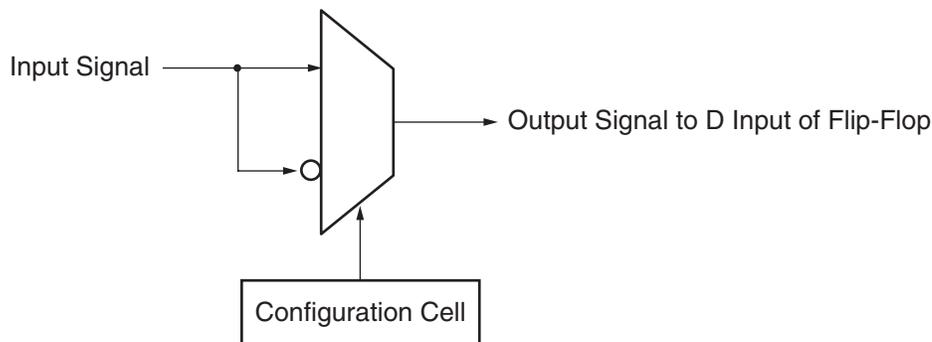
用 DCM (请参阅 “异步输入”), 则这一布线的自由仅适用于数据线, 而 “不” 适用于时钟线。交换线路绝对不会损坏器件。

吸收倒相器示例

在两种情况下, 可以向前吸收倒相器:

1. 当直接驱动触发器输入时
2. 当驱动到逻辑函数中时

在第一种情况下, Spartan-3 FPGA 系列在 CLB 触发器的直接 (D) 输入路径上有一个多路复用器 (如图 3 所示)。此多路复用器在真实输入信号和补充输入信号之间进行选择。它通过一个配置单元进行配置, 该单元由加载到此器件中的比特流进行初始化。用户在操作过程中不能访问多路复用器。



X491_03_041406

图 3: CLB 触发器前面的可编程倒相器

在第二种情况下, 倒相器直接就被吸收了。例如, 如果一个执行 “ $B = \sim A$ ” 的倒相器后面紧跟着一个执行 “ $D = B$ 与 C ” 的与门, 那么一种简单的替代方式就是一个无倒相器的执行 “ $D = \sim A$ 与 C ” 的与门; 换句话说, 就逻辑的利用和延迟而言, 倒相器吸收永远是 “自由” 的。

这种倒相器吸收的机制也适用于 IOB 输出触发器。同样, 如果是为简化 PCB 布局所需, 这种吸收可以 “自由” 地在 FPGA 的输出通路中加入倒相器。如果 FPGA 驱动某接插件, 而此接插件带有与此 FPGA 的 N 和 P 两个 LVDS 输出直接匹配的预定义引脚, 则此机制可以起到帮助作用。

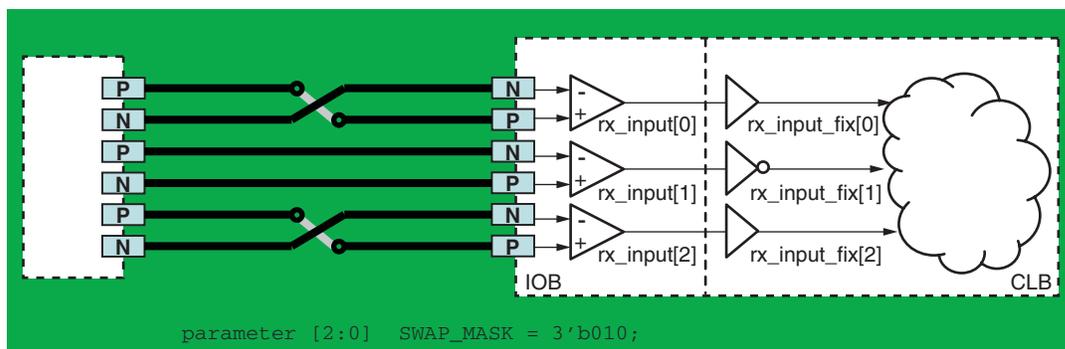
异步输入

图 2 是可以考虑的最简单的示例。接收到的经过交换的 LVDS 信号将用在 FPGA 内部的组合逻辑中。在此例中, 只需将一个简单的倒相器添加到代码中。此倒相器的代码以 Verilog 和 VHDL 语言表述如下:

```
Verilog:    assign rx_input_fix = ~rx_input;
VHDL:      rx_input_fix <= not rx_input;
```

可以将此倒相器吸收到由输入信号驱动的组合逻辑中, 也可以将其吸收到 FPGA 内部的某个触发器的 D 输入中, 但不能将其吸收到 FPGA 的 IOB 内的触发器、DCM 或 BUFGMUX 时钟缓冲器中。因此, 引脚交换的灵活性无法用到将用于数据时钟的时钟信号上。如果上述时钟只是系统的振荡器, 那么可以交换线路并且不需要再倒相, 这样做不会产生任何负面效应。

图 4 所示示例中的输入实际上是一条由 “n” 个信号对组成的总线。其中某些信号对是正确的, 而其他信号对则为了方便而进行了交换。此例中, 在设计中最好定义一个对应这 “n” 个输入的掩码。该掩码用来有选择地反转 (事实上不包括 “与”) 那些需要校正的位, 但不反转接收无误的位。在图 4 中, 位 0 和 2 是正确的, 而位 1 需要倒相。在代码中处理校正的最好方式是使用生成环, 它们可以例化输入缓冲器并且有选择地逐位执行倒相。



X491_04_041906

图 4: 一些总线迹线在 PCB 上交换了极性以简化布线

以下 Verilog 代码用生成环来执行接收倒相:

```

.
parameter [2:0] SWAP_MASK = 3'b010;
.
.
genvar i;
generate
for (i = 0; i <= 2; i = i + 1)
begin: loop0
IBUFDS
#(.IOSTANDARD("LVDS_25"), .IBUF_DELAY_VALUE("0"), .DIFF_TERM("FALSE"))
ibuf_d (.I(datain_p[i]), .IB(datain_n[i]), .O(rx_input[i]));
assign rx_input_fix[i] = rx_input[i] ^ SWAP_MASK[i];
end
endgenerate

```

以下 VHDL 代码用生成环来执行接收倒相:

```

.
constant SWAP_MASK : std_logic_vector(2 downto 0) := "010";
.
.
loop0: for i in 0 to 2 generate
ibuf_d: ibufds generic map
(IOSTANDARD => "LVDS_25", IBUF_DELAY_VALUE => "0", DIFF_TERM => FALSE)
port map
(i => datain_p(i), iB => datain_n(i), o => rx_input(i));
rx_input_fix(i) <= rx_input(i) xor SWAP_MASK(i);
end generate;

```

通过修改红色的粗体字符, 可以方便地将此机制扩展到不同的位宽。

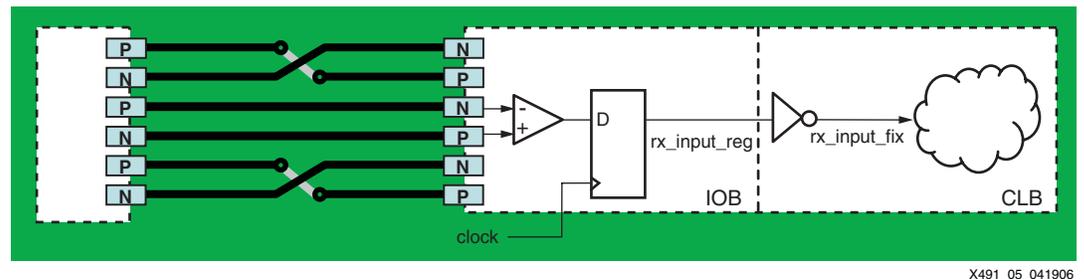
IOB 输入触发器的同步使用

通常, 输入信号寄存在 IOB 触发器中, 因为高速数据传输是使用 LVDS 的最常见的原因。可使用下列技术之一来寄存数据:

- 单数据速率 (SDR) 技术, 这种技术在 IOB 中只使用一个 (通常是正沿触发的) 触发器
或者
 - 双数据速率 (DDR) 技术, 这种技术使用正沿和负沿都触发的触发器对输入数据线进行采样
- 在两种情况下都不可能在输入放大器和触发器之间反转输入信号, 因为 IOB 模块中的触发器没有可倒相的输入。倒相器需要添加到 IOB 输入触发器后, 它们可以被吸收到其后的寄存逻辑或组合逻辑中。

SDR 示例

图 5 所示为 IOB 中带有触发器的 SDR 方案。



X491_05_041906

图 5: SDR 寄存接收器

以下代码说明针对相同生成环示例的 SDR 寄存情况。唯一的变化是增加了触发器例示。

Verilog 语言代码:

```
.
parameter [2:0] SWAP_MASK = 3'b010;
.
.
genvar i;
generate
for (i = 0; i <= 2; i = i + 1)
begin: loop0
IBUFDS#(.IOSTANDARD("LVDS_25"), .IFD_DELAY_VALUE("0"), .DIFF_TERM("FALSE"))
ibuf_d (.I(datain_p[i]), .IB(datain_n[i]), .O(rx_input[i]));
FD fd_d (.C(clkin), .D(rx_input[i]), .Q(rx_input_reg[i]));
assign rx_input_fix[i] = rx_input_reg[i] ^ SWAP_MASK[i];
end
endgenerate
```

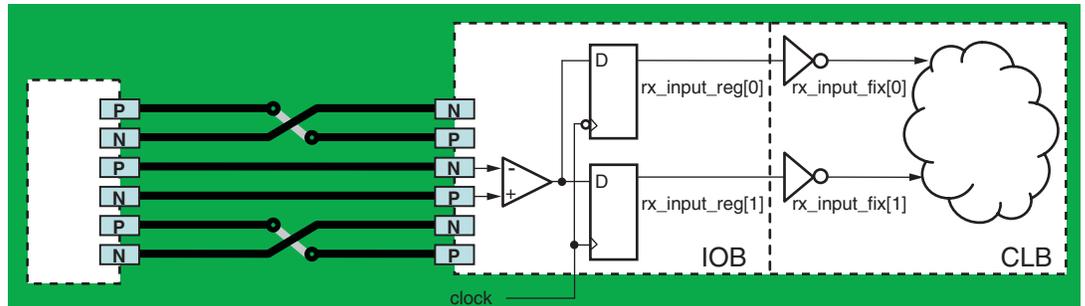
VHDL 语言代码:

```
.
constant SWAP_MASK : std_logic_vector(2 downto 0) := "010";
.
.
loop0: for i in 0 to 2 generate
ibuf_d: ibufds
generic map (IOSTANDARD => "LVDS_25", IFD_DELAY_VALUE => "0", DIFF_TERM => FALSE)
port map (i => datain_p(i), iB => datain_n(i), o => rx_input(i));
fd_d: fd port map (c => clkin, d => rx_input(i), q => rx_input_reg(i));
rx_input_fix(i) <= rx_input_reg(i) xor SWAP_MASK(i);
end generate;
```

通过修改红色的粗体字符，可以方便地将此机制扩展到不同的位宽。

输入 DDR 示例

图 6 所示为接收 DDR 方案，其中每条输入线生成两条可能需要倒相的内部数据线。对于 Spartan-3E FPGA 的 DDR 输入，推荐使用新型的 IDDR2 输入触发器结构。这种结构去除了从下降沿到下一上升沿的所有通路，从而可以使内部逻辑更容易设计。要了解有关 IDDR2 的详情，请参阅 [DS312](#) 《Spartan-3E FPGA 系列数据手册》。



X491_06_041906

图 6: DDR 寄存接收器

以下代码说明针对相同生成环示例的 DDR 寄存接收器情况。唯一的变化是增加了 Spartan-3E FPGA 的 IDDR2 示例。原 Spartan-3 器件需要略微不同的代码，因为它不包含 IDDR2 结构。所附 ZIP 文件包含完整细节（请参阅“设计文件”）。

Verilog 语言代码:

```

parameter [2:0] SWAP_MASK = 3'b010;
.
.
genvar i;
generate
for (i = 0; i <= 2; i = i + 1)
begin: loop0
IBUFDS#(.IOSTANDARD("LVDS_25"), .IFD_DELAY_VALUE("0"), .DIFF_TERM("FALSE"))
ibuf_d (.I(datain_p[i]), .IB(datain_n[i]), .O(rx_input[i]));
IDDR2 #(.DDR_ALIGNMENT("C0")) fd_ioc(.C0(clkin), .C1(notclk), .D(rx_input[i]),
.CE(1'b1), .R(1'b0), .S(1'b0), .Q0(rx_input_reg[i+3]),
.Q1(rx_input_reg[i]));
assign rx_input_fix[i] = rx_input_reg[i] ^ SWAP_MASK[i];
assign rx_input_fix[i+3] = rx_input_reg[i+3] ^ SWAP_MASK[i];
end
endgenerate

```

VHDL 语言代码:

```

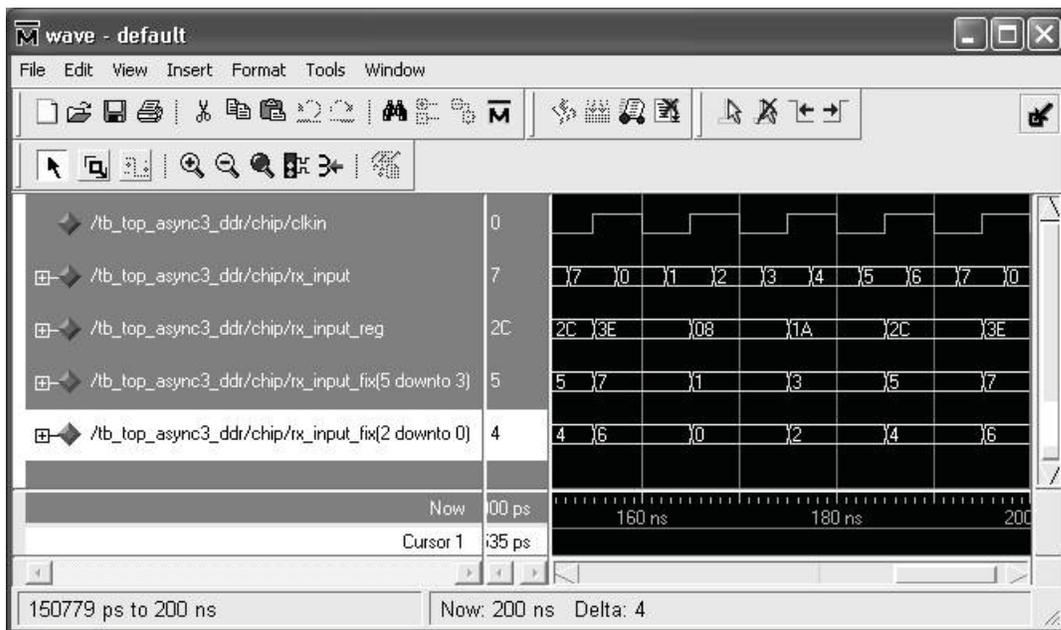
constant SWAP_MASK : std_logic_vector(2 downto 0) := "010";
.
.
loop0: for i in 0 to 2 generate
ibuf_d : ibufds
generic map (IOSTANDARD => "LVDS_25", IFD_DELAY_VALUE => "0", DIFF_TERM => FALSE)
port map (i => datain_p(i), iB => datain_n(i), o => rx_input(i));
fd_d : iddr2
generic map (DDR_ALIGNMENT => "C0")
port map (c0 => clkin, c1 => notclock, d => rx_input(i), ce => '1', r => '0',
s => '0', q0 => rx_input_reg(i+3), q1 => rx_input_reg(i));
rx_input_fix(i) <= rx_input_reg(i) xor SWAP_MASK(i);
rx_input_fix(i+3) <= rx_input_reg(i+3) xor SWAP_MASK(i);

```

通过修改红色的粗体字符，可以方便地将此机制扩展到不同的位宽。

在使用 DDR 技术时，位操作可能很重要。DDR 生成环示例生成一条总线，总线的低阶位聚集在时钟的下降沿，而其高阶位则聚集在下一个上升沿。图 7 是 DDR 设计模拟运行的屏幕截图，

其中显示了这种位集中现象。此模拟运行假设所有迹线都是正确的（即没有进行引脚交换），以清楚地显示具体位的结束位置。

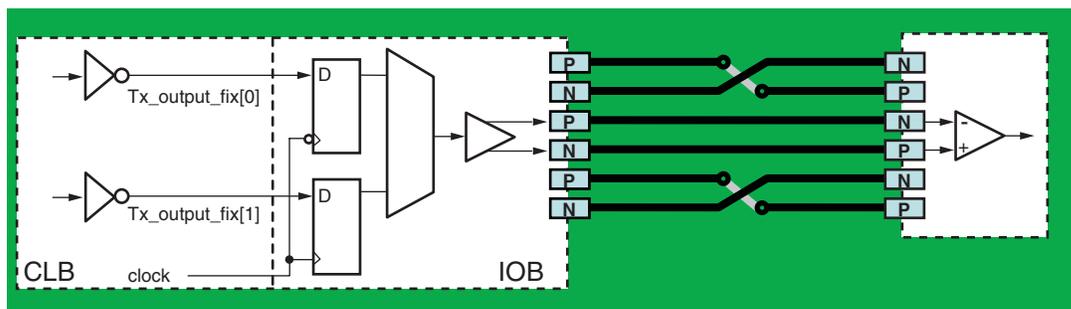


X491_07_041906

图 7: 输入 DDR 情形的模拟

输出 DDR 示例

图 8 所示为发射器的 DDR 方案，其中每对发送数据线由 Spartan-3E FPGA 中的 ODDR2（在 Spartan-3 FPGA 中是 FDDRSE）机制进行多路复用。这种情况下，在与极性需要倒相的 LVDS 输出相关联的每条线路中都增加了倒相器。因为这些倒相器在上述实现过程中被吸收到输出触发器中，所以它们不会改变电路的时序。



X491_08_092106

图 8: 双数据速率寄存发射器

以下代码说明生成环示例的发射器 DDR 寄存情形。原 Spartan-3 器件需要略微不同的代码，因为它不包含 ODDR2 结构。所附 ZIP 文件包含完整细节（请参阅“设计文件”）。

Verilog 语言代码:

```
parameter [2:0] SWAP_MASK = 3'b010 ;

genvar i ;
generate
for (i = 0 ; i <= 2 ; i = i + 1)
begin : loop0
OBUFDS #(.IOSTANDARD("LVDS_25"))
obuf_d (.I(tx_output_reg[i]), .O(dataout_p[i]), .OB(dataout_n[i]));
ODDR2 #(.DDR_ALIGNMENT("NONE")) fd_ioc (.C0(clkin), .C1(notclk),
.D0(tx_output_fix[i+3]), .D1(tx_output_fix[i]), .CE(1'b1), .R(1'b0),
.S(1'b0), .Q(tx_output_reg[i])) ;
assign tx_output_fix[i] = tx_output[i] ^ SWAP_MASK[i] ;
assign tx_output_fix[i+3] = tx_output[i+3] ^ SWAP_MASK[i] ;
end
endgenerate
```

VHDL 语言代码:

```
constant SWAP_MASK : std_logic_vector(2 downto 0) := "010" ;

loop0 : for i in 0 to 2 generate
ibuf_d : obufds generic map (IOSTANDARD => "LVDS_25")
port map (i => tx_output_reg(i), o => dataout_p(i), oB =>
dataout_n(i));
fd_d : oddr2 generic map (DDR_ALIGNMENT => "NONE")
port map (c0 => clkin, c1 => notclock, d0 => tx_output_fix(i),
d1 => tx_output_fix(i+3), ce => '1', r => '0', s => '0', q =>
tx_output_reg(i));
tx_output_fix(i) <= tx_output(i) xor SWAP_MASK(i) ;
tx_output_fix(i+3) <= tx_output(i+3) xor SWAP_MASK(i) ;
end generate ;
```

通过修改红色的粗体字符，可以方便地将此机制扩展到不同的位宽。

如上所述，在使用 DDR 技术时，位操作可能很重要。DDR 生成环示例生成一条总线，总线的低阶位在时钟的下降沿上发送，而其高阶位则在下一个上升沿上发送。

设计文件

本应用指南介绍的各种接收器和发射器示例的设计文件是针对所有 Spartan-3 器件和 Spartan-3E 系列器件编写的。可从 Xilinx 网站 ([xapp491.zip](http://www.xilinx.com/xapp491.zip)) 获得 Verilog 和 VHDL 两种语言的设计文件。所附 readme.txt 文件提供了最新详情。

结论

用 LVDS 进行设计时，有计划地谨慎使用 Spartan-3 FPGA 系列资源，可以大幅度降低 PCB 布局的复杂性，同时可以提高整体电路板信号的完整性。对于器件中集成的 LVDS 接收器和 LVDS 发射器来说也是如此，但输入时钟引脚除外；输入时钟引脚必须具有正确的极性。

修订历史

下表说明此技术文档的修订历史。

日期	版本	修订
2006 年 10 月 4 日	1.0	Xilinx 最初版本。