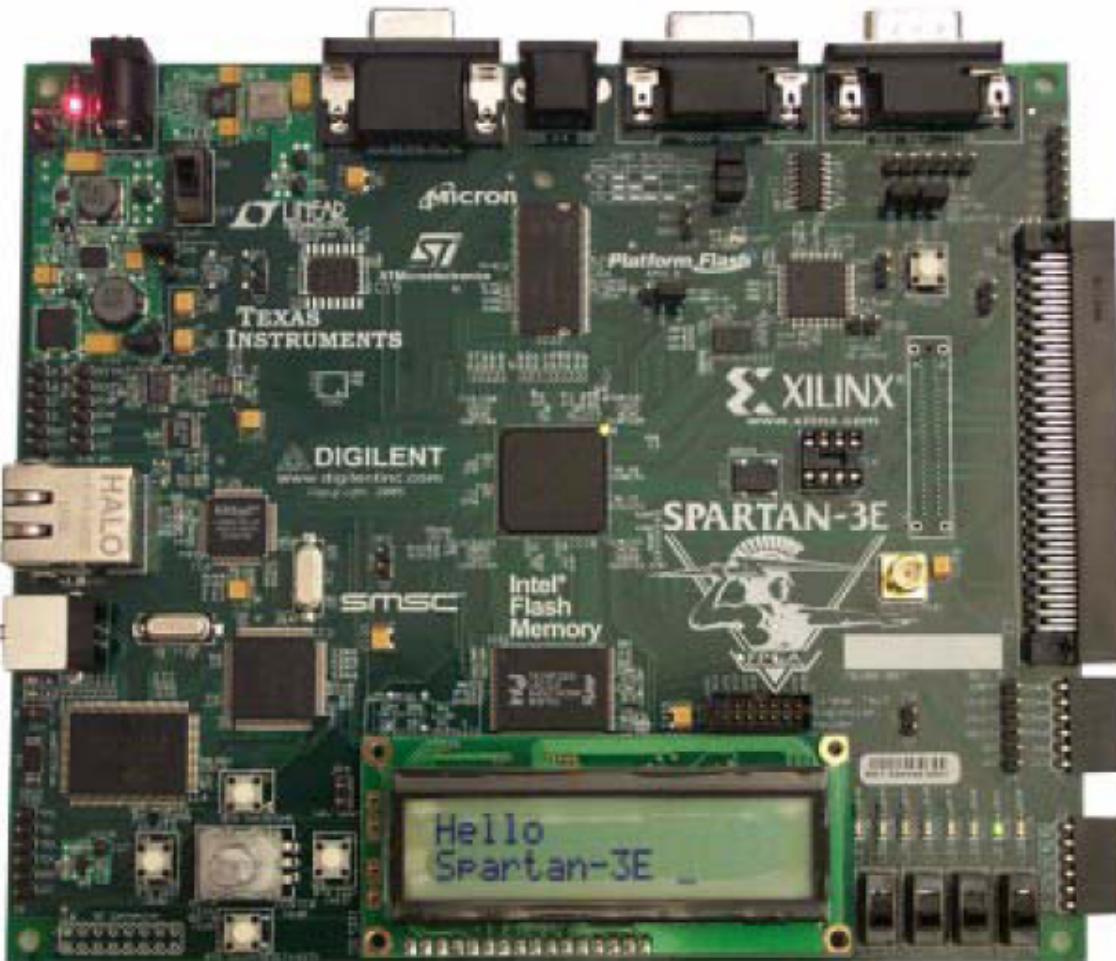


# Spartan-3E Starter Kit Board User Guide

v1.0) March 9, 2006

Click a component to jump to the  
related documentation. Not all  
components have active links.



- Chapter 1: Introduction and Overview**
- Chapter 2: Switches, Buttons, and Knob**
- Chapter 3: Clock Sources**
- Chapter 4: FPGA Configuration Options**
- Chapter 5: Character LCD Screen**
- Chapter 6: VGA Display Port**
- Chapter 7: RS-232 Serial Ports**
- Chapter 8: PS/2 Mouse/Keyboard Port**
- Chapter 9: Digital to Analog Converter (DAC)**
- Chapter 10: Analog Capture Circuit**
- Chapter 11: Intel StrataFlash Parallel NOR Flash PROM**
- Chapter 12: SPI Serial Flash**
- Chapter 13: DDR SDRAM**
- Chapter 14: 10/100 Ethernet Physical Layer Interface**
- Chapter 15: Expansion Connectors**
- Chapter 16: XC2C64A CoolRunner-II CPLD**
- Chapter 17: DS2432 1-Wire SHA-1 EEPROM**

# *Chapter 1: Introduction and Overview*

Spartan-3E入门实验板使设计人员能够即时利用Spartan-3E系列的完整平台性能。

**设备支持:** Spartan-3E、CoolRunner-II

**关键特性:** Xilinx 器件: Spartan-3E (50万门, XC3S500E-4FG320C), CoolRunner™-II (XC2C64A-5VQ44C)与Platform Flash (XCF04S-VO20C)

**时钟:** 50 MHz晶体时钟振荡器

**存储器:** 128 Mbit 并行Flash, 16 Mbit SPI Flash, 64 MByte DDR SDRAM

**连接器与接口:** 以太网10/100 Phy, JTAG USB下载,两个9管脚RS-232串行端口, PS/2类型鼠标/键盘端口, 带按钮的旋转编码器, 四个滑动开关,八个单独的LED输出, 四个瞬时接触按钮, 100管脚hirose扩展连接端口与三个6管脚扩展连接器

**显示器:** VGA显示端口, 16 字符- 2 线式 LCD

**电源:** Linear Technologies 电源供电, TPS75003三路电源管理IC

**市场:** 消费类, 电信/数据通信, 服务器, 存储器

**应用:** 可支持 32 位的 RISC 处理器, 可以采用 Xilinx 的 MicroBlaze 以及 PicoBlaze 嵌入式开发系统; 支持 DDR 接口的应用; 支持基于 Ethernet 网络的应用; 支持大容量 I/O 扩展的应用。

## **Choose the Starter Kit Board for Your Needs**

### **Spartan-3E FPGA Features and Embedded Processing Functions**

Spartan3-E FPGA 入门实验板具有 Spartan3-E FPGA 系列突出独特的特点和为嵌入式处理发展与应用提供了很大的方便。该板的特点如下:

Spartan3-E 特有的特征: 并行 NOR Flash 配置; 通过并行 NOR Flash PROM 实现 FPGA 的多种配置方式

嵌入式系统: MicroBlaze™ 32-bit 嵌入RISC处理器; PicoBlaze™ 8-bit 嵌入控制器; DDR 存储器接口

### **Learning Xilinx FPGA, CPLD, and ISE Development Software Basics**

Spartan3-E FPGA 入门实验板比其他的入门实验板先进、复杂。它是学习FPGA或CPLD设计和怎样运用ISE软件的基础。

### **Advanced Spartan-3 Generation Development Boards**

入门实验板示范了MicroBlaze™ 32-bit 嵌入式处理器和EDK的基本运用。其更先进的地方

在于配了额外的外设和FPGA逻辑，包括SP-305入门实验板。

## Key Components and Features

主要特征：1) XC3S500E(Spartan-3e): 多达232个用户I/O口；320个FPGA封装管脚；超过10000个逻辑单元。2) 4Mbit的Flash 配置PROM；3) 64个宏单元的XC2C64A CoolRunner CPLD；4) 64 MByte (512 Mbit) of DDR SDRAM, × 16 数据接口, 100+ MHz；5) 16 MByte (128 Mbit) of 并行NOR Flash (Intel StrataFlash): FPGA配置存储；MicroBlaze代码存储/映射；6) 16 Mbits of SPI serial Flash (STMicro): FPGA配置存储；MicroBlaze代码存储/映射；7) 16字符-2线式LCD显示屏；8) PS/2鼠标或键盘接口；9) VGA显示接口；10) 10/100以太PHY（要求FPGA内部具有以太MAC）；11) 2个9-管脚的RS-232端口（DTE和DCE两种类型）；12) FPGA/CPLD下载/调试USB接口；13) 50Hz时钟晶振；14) 1线式的SHA-1位流复制保护串行EEPROM；15) Hirose FX2扩展连接口；16) 3个管脚扩展连接器；17) 4个SPI-DAC转换器输出管脚；18) 2个SPI带可编程增益ADC输入管脚；19) ChipScope™软件调试接口；20) 带按钮的旋转编码器；21) 8个单独的LED输出；22) 4个滑动开关；23) 4个按钮开关；24) SMA时钟输入；25) 8管脚插槽辅助晶振

## Design Trade-Offs

### Configuration Methods Galore!

FPGA的一个典型应用就是使用单永久性存储器来存储配置信息。为了说明新的Spartan-3E的性能，入门实验板有3个不同的配置存储源，这需要一起正确使用。额外的配置功能使入门实验板比典型的Spartan-3E应用更复杂。

入门实验板包括JTAG可编程USB接口。片上的线路简化了器件的编程过程。在典型的应用中，JTAG编程硬件在片外或在一个单独的编程模块上，如XILINX USB电缆平台。

## Voltages for all Applications

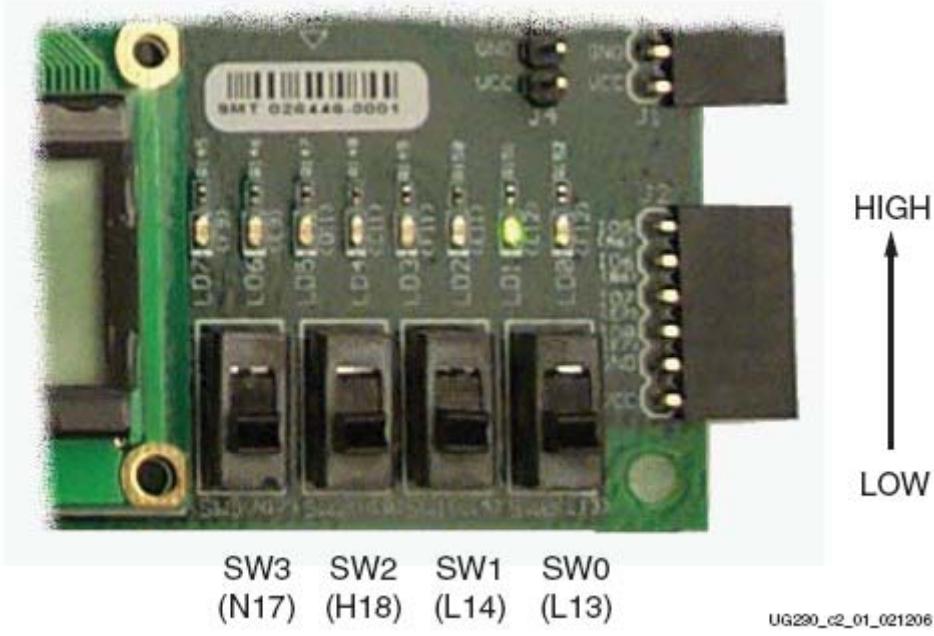
入门实验板利用TI公司的TPS75003芯片（专门为Spartan-3E的FPGA提供电源）作为三态输出调整仪。该调整仪适用多种FPGA的应用。但是，入门实验板包括DDR SDRAM，这需要它自身快速的电流来供给。简单地说，带USB接口的JTAG下载方式解决了需要单独配备1.8V电源的问题。

# Chapter 2: Switches, Buttons, and Knob

## Slide Switches

### Locations and Labels

入门实验板具有4个滑动开关，如图2.1所示。



*Figure 2-1: Four Slide Switches*

## Operation

当开关关上或上拉时，FPGA的管脚连接3.3V电源，即逻辑高电平。断开或下拉时，FPGA管脚接地，逻辑低电平。一般开关的机械闭合时间为2ms，这里没有使用活动的回弹线路，尽管这种线路可以很容易地加到FPGA上。

## UCF Location Constraints

图2.2为4个滑动开关提供了UCF约束、I/O口分配和I/O口标准。这里没有用到上拉电阻。但是，当开关处于中间转换位置时，它被定义为输入。

```

NET "SW<0>" LOC = "L13" | IOSTANDARD = LVTTL | PULLUP ;
NET "SW<1>" LOC = "L14" | IOSTANDARD = LVTTL | PULLUP ;
NET "SW<2>" LOC = "H18" | IOSTANDARD = LVTTL | PULLUP ;
NET "SW<3>" LOC = "N17" | IOSTANDARD = LVTTL | PULLUP ;

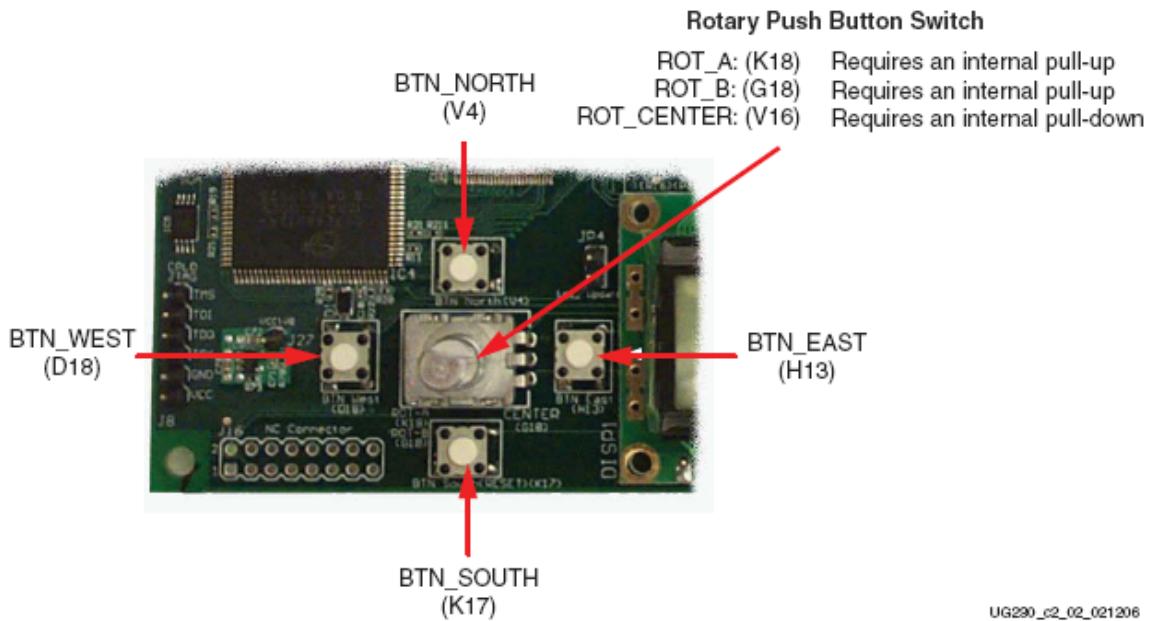
```

*Figure 2-2: UCF Constraints for Slide Switches*

## Push-Button Switches

### Locations and Labels

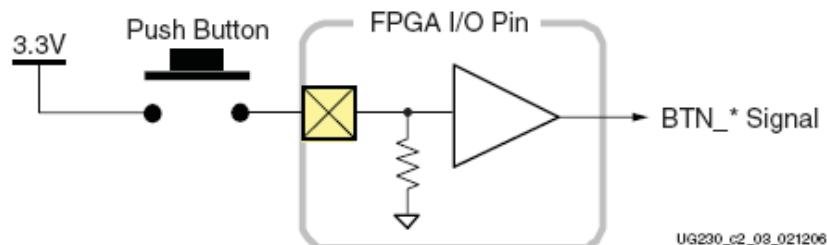
入门实验板有4个瞬时按钮开关，如图2.3所示。BTN\_NORTH、BTN\_EAST、BTN\_SOUTH、and BTN\_WEST。



注：a 所有的BTN\_\*按钮输入需要内部的下拉电阻；b 在一些FPGA应用中BTN\_SOUTH作为软复位使用。

## Operation

按下按钮，FPGA接到3.3v电源，如图2.4。没有按下时，鉴于内部下拉电阻的原因，FPGA管脚产生一个逻辑低电平。图2.5说明了怎样去定义下拉电阻的UCF。这里按钮上没有活动的回弹线路。



**Figure 2-4: Push-Button Switches Require an Internal Pull-Down Resistor in FPGA Input Pin**

在一些应用中，BTN\_SOUTH按钮开关充FPGA选择复位功能的软复位。

## UCF Location Constraints

图2.5为4个按钮开关提供了UCF约束、I/O口分配和I/O口标准。并为每个输入管脚定义下拉电阻。

```

NET "BTN_EAST" LOC = "H13" | IOSTANDARD = LVTTL | PULLDOWN ;
NET "BTN_NORTH" LOC = "V4" | IOSTANDARD = LVTTL | PULLDOWN ;
NET "BTN_SOUTH" LOC = "K17" | IOSTANDARD = LVTTL | PULLDOWN ;
NET "BTN_WEST" LOC = "D18" | IOSTANDARD = LVTTL | PULLDOWN ;

```

**Figure 2-5: UCF Constraints for Push-Button Switches**

## Rotary Push-Button Switch

## Locations and Labels

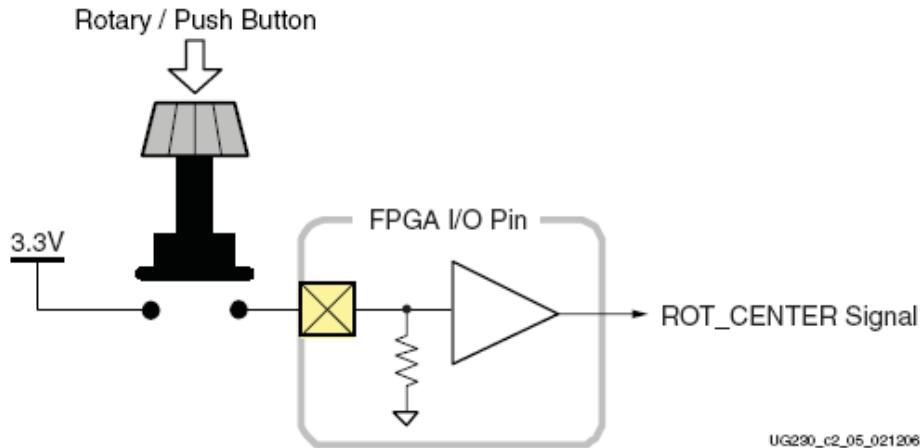
如图2.3所示，旋转按钮处4个分开的按钮的中间。该开发产生3个输出：2个轴状编码输出ROT\_A和ROT\_B。中心的按钮是ROT\_CENTER。

## Operation

旋转按钮有2个不同的功能。只要轴柄一转，按钮就旋转并输出值。该轴也可以按下，和按钮开关一样。

### Push-Button Switch

按下握柄或按钮，则FPGA接通3.3V，如图2.6所示。使用内部的下拉电阻产生低电平。图2.9说明了怎样去定义UCF的下拉电阻。这里按钮上没有活动的回弹线路。



UG290\_c2\_05\_021206

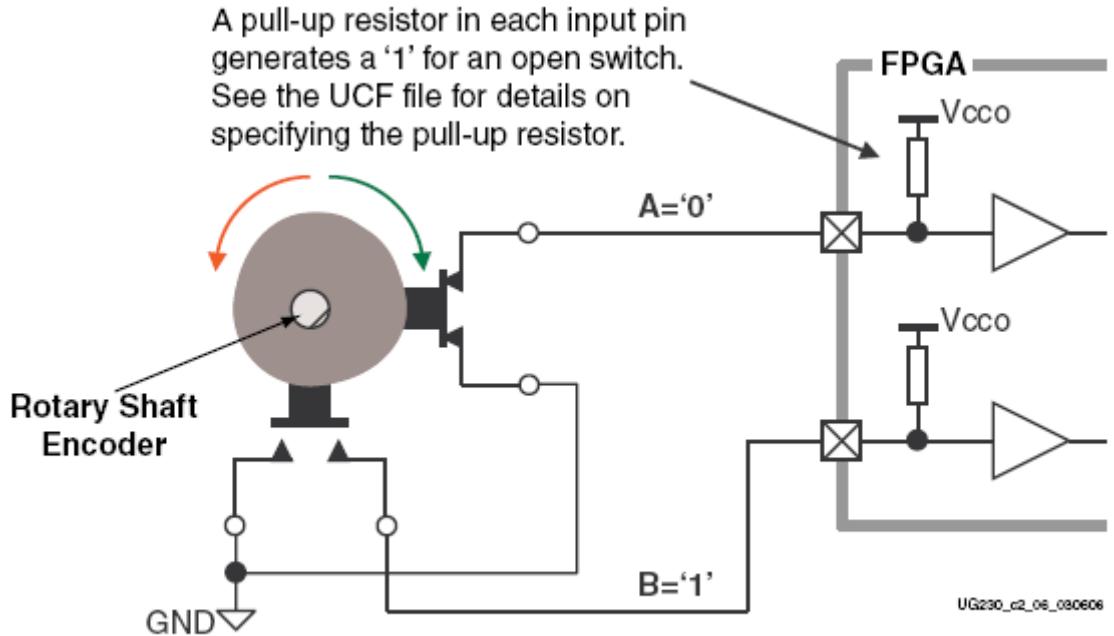
**Figure 2-6: Push-Button Switches Require Internal Pull-up Resistor in FPGA Input Pin**

### Rotary Shaft Encoder

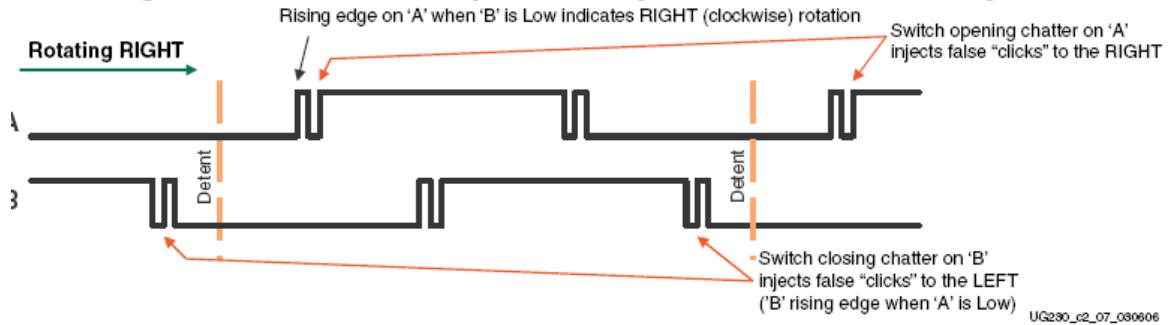
首先，旋转编码更像是个连接到中心的凸轮。旋转轴柄可操作两个按钮开关，如图2.7所示。

选择一个开关连接地，产生低电平。当开关开时，FPGA内部的上拉电阻将该信号拉为高电平。图2.9是对其UCF约束的描述，怎样去定义上拉电阻。

FPGA电路很方便地译码A和B的输入信号，但考虑到开关的机械特性，转换时会产生输入噪音。如图2.8所示，噪音错误地报告额外的旋转事件或甚至报告旋转相反的方向。



**Figure 2-7: Basic example of rotary shaft encoder circuitry**



**Figure 2-8: Outputs from Rotary Shaft Encoder May Include Mechanical Chatter**

### UCF Location Constraints

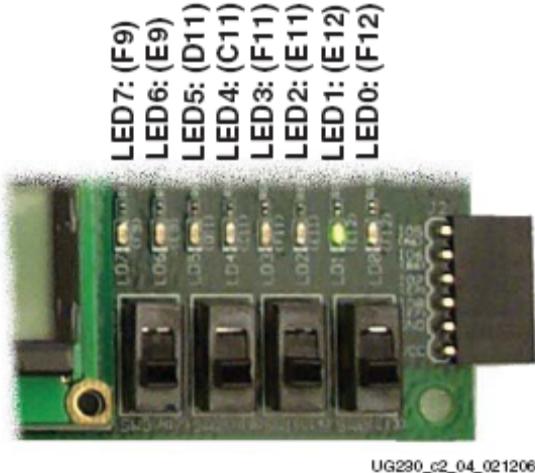
```
NET "ROT_A"      LOC = "K18" | IOSTANDARD = LVTTL | PULLUP    ;
NET "ROT_B"      LOC = "G18" | IOSTANDARD = LVTTL | PULLUP    ;
NET "ROT_CENTER" LOC = "V16" | IOSTANDARD = LVTTL | PULLDOWN ;
```

**Figure 2-9: UCF Constraints for Rotary Push-Button Switch**

### Discrete LEDs

#### Locations and Labels

入门实验板在滑动开关的上面有 8 个独立的贴片 LED。



*Figure 2-10: Eight Discrete LEDs*

## Operation

LED 一端接地，另一端通过 390 欧的限流电阻接到 Spartan-3E 上。要点亮一个 LED，向相应的控制位置高。

## UCF Location Constraints

NET "LED<7>" LOC = "F9"   IOSTANDARD = LVTTL   SLEW = SLOW   DRIVE = 8 ;
NET "LED<6>" LOC = "E9"   IOSTANDARD = LVTTL   SLEW = SLOW   DRIVE = 8 ;
NET "LED<5>" LOC = "D11"   IOSTANDARD = LVTTL   SLEW = SLOW   DRIVE = 8 ;
NET "LED<4>" LOC = "C11"   IOSTANDARD = LVTTL   SLEW = SLOW   DRIVE = 8 ;
NET "LED<3>" LOC = "F11"   IOSTANDARD = LVTTL   SLEW = SLOW   DRIVE = 8 ;
NET "LED<2>" LOC = "E11"   IOSTANDARD = LVTTL   SLEW = SLOW   DRIVE = 8 ;
NET "LED<1>" LOC = "E12"   IOSTANDARD = LVTTL   SLEW = SLOW   DRIVE = 8 ;
NET "LED<0>" LOC = "F12"   IOSTANDARD = LVTTL   SLEW = SLOW   DRIVE = 8 ;

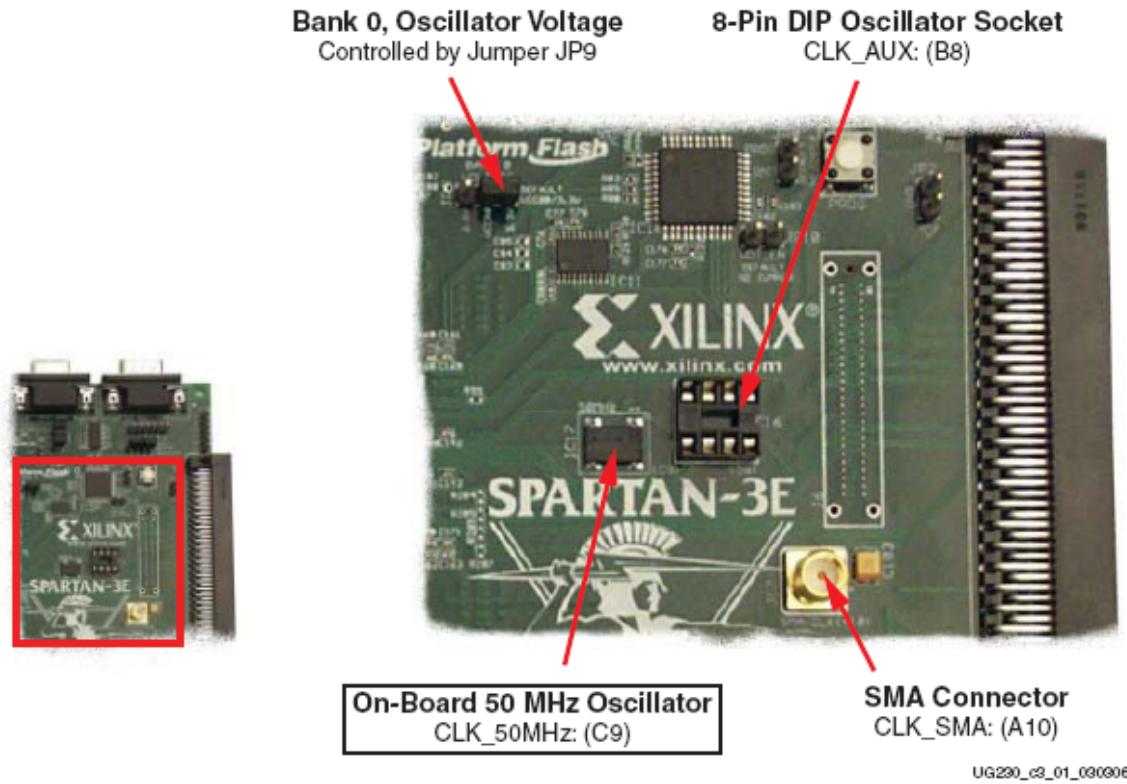
*Figure 2-11: UCF Constraints for Eight Discrete LEDs*

## Chapter 3: Clock Sources

### Overview

图 3.1 所示，入门实验板支持 3 个主时钟输入源。

- 1) 包括一个 50MHz 的时钟晶振
- 2) 通过 SMA 连接器，时钟可以板外供应。反之，FPGA 也可以提供时钟信号或其它高速信号给 SMA 连接器
- 3) 分列式 8-DIP 时钟晶振插槽



UG230\_c3\_01\_030006

Figure 3-1: Available Clock Inputs

## Clock Connections

每个时钟的输入直接连到 Bank0 的输入全局缓冲 I/O。表 3.1 所示，每个时钟输入也可以连接到相应的 DCM。

Table 3-1: Clock Inputs and Associated Global Buffers and DCMs

Clock Input	FPGA Pin	Global Buffer	Associated DCM
CLK_50MHz	C9	GCLK10	DCM_X0Y1
CLK_AUX	B8	GCLK8	DCM_X0Y1
CLK_SMA	A10	GCLK7	DCM_X1Y1

## Voltage Control

FPGA 的 I/O Bank0 的供给电压由跳线 JP9 控制。因此，这些时钟源也是由 JP9 控制的。默认情况下，JP9 置为 3.3v。板上的晶振是 3.3V 的，如果 JP9 为 2.5V，晶振的工作有可能达不到预期的效果。

## 50 MHz On-Board Oscillator

50MHz 的晶振占空比为 40%~50% 之间。其精度为 ±2500Hz 或 ±50ppm。

## Auxiliary Clock Oscillator Socket

如果 FPGA 需要除了 50MHz 外的频率，可以使用该 8 管脚的插槽。另外，也可以使用 FPGA 的 DCM 来产生，或者从板上的 50MHz 晶振结合其它频率来获得。

## SMA Clock Input or Output Connector

连接输入时钟信号到 SMA 连接器可以从外部提供一个时钟。FGPA 也可以在 SMA 上产生一个单端的时钟输出信号或高速信号给外部器件。

## UCF Constraints

### Location

```
NET "CLK_50MHZ" LOC = "C9" | IOSTANDARD = LVCMOS33 ;
NET "CLK_SMA" LOC = "A10" | IOSTANDARD = LVCMOS33 ;
NET "CLK_AUX" LOC = "B8" | IOSTANDARD = LVCMOS33 ;
```

Figure 3-2: UCF Location Constraints for Clock Sources

### Clock Period Constraints

```
# Define clock period for 50 MHz oscillator
NET "CLK_50MHZ" PERIOD = 20.0ns HIGH 40%;
```

Figure 3-3: UCF Clock PERIOD Constraint

## Chapter 4:

# FPGA Configuration Options

入门实验板支持多种 FPGA 的配置方法：

通过JTAG、USB接口直接将设计下载到FPGA。板上的USB-JTAG逻辑也提供对Platform Flash PROM 和 Xilinx XC2C64A CPLD的在线编程。

对板上的4 Mbit Xilinx XCF04S serial Platform Flash PROM进行编程，然后采用主串行模式对Platform Flash PROM上的存储信息配置到FPGA。

对板上的16 Mbit ST Microelectronics SPI serial Flash PROM进行编程，然后采用SPI模式对SPI serial Flash PROM上的存储信息配置到FPGA。

对板上的128 Mbit Intel StrataFlash parallel NOR Flash PROM进行编程，然后采用BPI Up或BPI Down对StrataFlash parallel NOR Flash PROM上的存储信息配置到FPGA。接着，可以采用Spartan-3E FPGA's MultiBoot 模式以两种不同的配置方式对一个FPGA进行轮流下载。

图4.1给出了USB下载/编程接口和永久存储器的位置。图4.2是多种配置方式的一些细节内容。

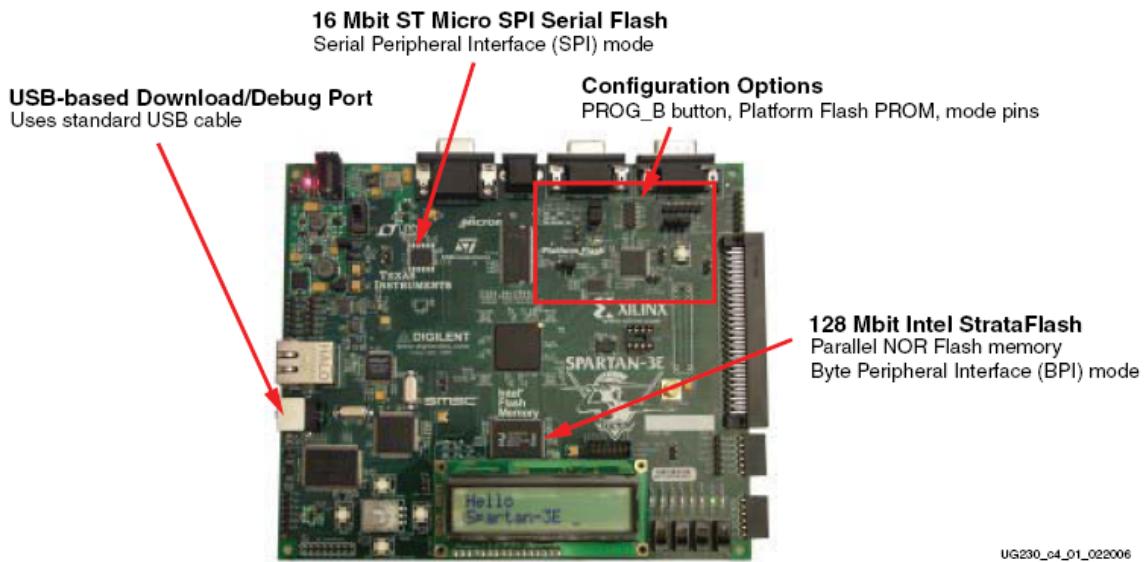


Figure 4-1: Spartan-3E Starter Kit FPGA Configuration Options

**Configuration Mode Jumper Settings (Header J30)**

Select between three on-board configuration sources

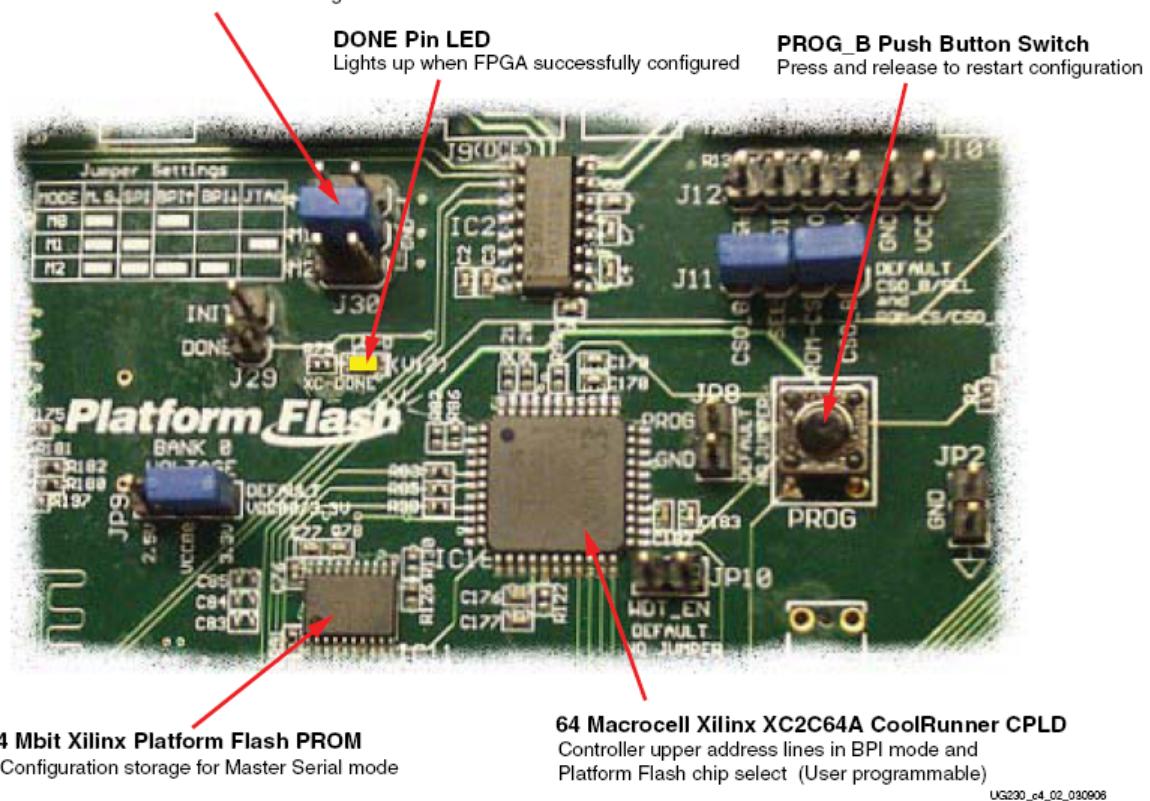


Figure 4-2: Detailed Configuration Options

当上电或PROG按钮被按下时，跳线的配置方式决定了FPGA采用哪种配置方式。

当FPGA成功配置时，DONE管脚的LED亮。

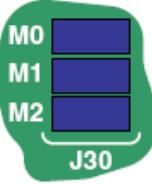
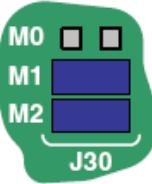
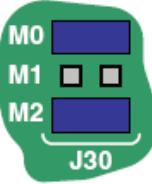
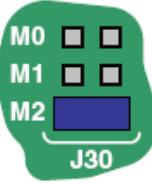
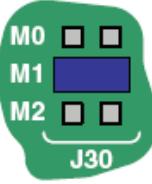
4 Mbit Xilinx Platform Flash PROM为FPGA的JTAG提供了一个简单的可编程配置存储单元。来自Platform Flash PROM的FPGA配置采用主串行模式。

采用BPI Up、BPI Down或MultiBoot配置方式以及StrataFlash parallel Flash PROM对FPGA进行下载时，64-macrocell XC2C64A CoolRunner II CPLD为其提供了额外的存储空间。CPLD 由

用户自己编程。

## Configuration Mode Jumpers

如表 4.1 所示，由 J30 的设置方式来控制 FPGA 的配置方式。

Configuration Mode	Mode Pins M2:M1:M0	FPGA Configuration Image Source	Jumper Settings
Master Serial	0:0:0	Platform Flash PROM	
SPI (see <a href="#">Chapter 12, “SPI Serial Flash”</a> )	1:1:0	SPI Serial Flash PROM starting at address 0	
BPI Up (see <a href="#">Chapter 11, “Intel StrataFlash Parallel NOR Flash PROM”</a> )	0:1:0	StrataFlash parallel Flash PROM, starting at address 0 and incrementing through address space. The CPLD controls address lines A[24:20] during BPI configuration.	
BPI Down (see <a href="#">Chapter 11, “Intel StrataFlash Parallel NOR Flash PROM”</a> )	0:1:1	StrataFlash parallel Flash PROM, starting at address 0x1FF_FFFF and decrementing through address space. The CPLD controls address lines A[24:20] during BPI configuration.	
JTAG	0:1:0	Downloaded from host via USB-JTAG port	

## PROG Push Button

PROG 按钮，迫使 FPGA 进行重新配置。只要按下该按钮，FPGA 就重新进行配置。

## DONE Pin LED

当 FPGA 成功配置时，DONE 管脚的 LED 亮，如果不亮说明配置失败。

## Programming the FPGA, CPLD, or Platform Flash PROM via USB

正如图4.1所示，入门实验板包含了内嵌式USB可编程逻辑和B型连接器的USB接口。通过USB电缆可将其连到主机的iMPACT可编程软件对FPGA、Platform Flash PROM或CPLD进行直接编程。对于并行或串行的Flash PROM并暂时不支持直接编程。

### Connecting the USB Cable

实验板包含一个标准的 USB A/B 型的电缆，如图 4.3 所示。

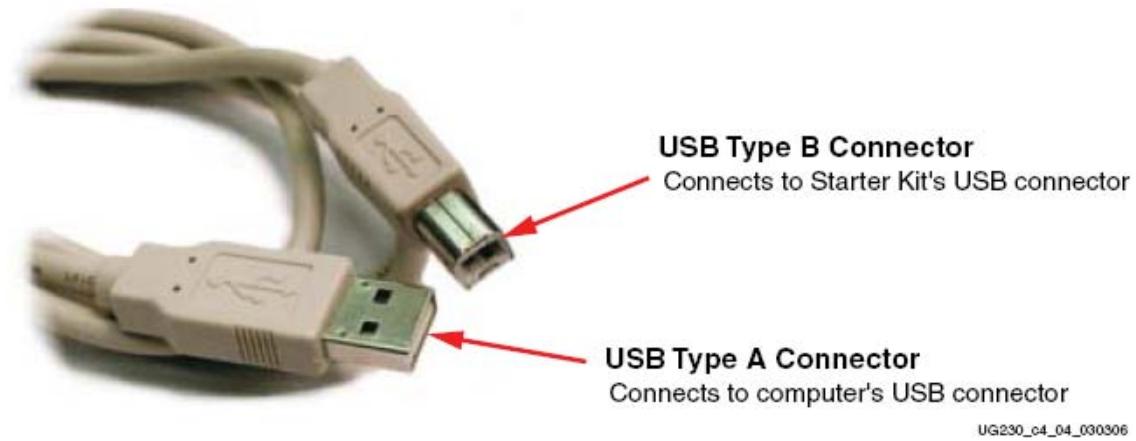


Figure 4-3: Standard USB Type A/Type B Cable

其中，A 型的连接口接到 PC 上，B 型接口接到板上。上电时，Windows 操作系统通过安装相应的驱动软件即可识别。

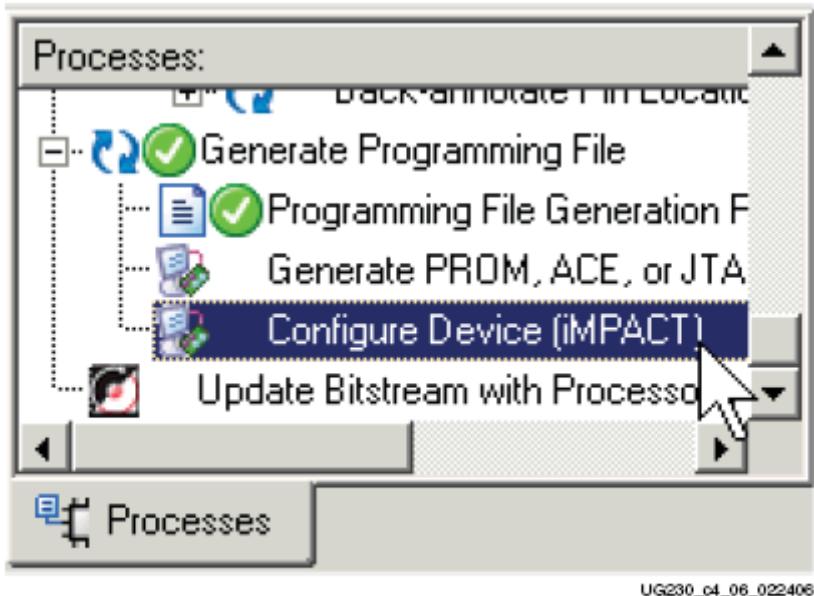


Figure 4-4: Connect the USB Type B Connector to the Starter Kit Board Connector  
绿色的 LED 亮表示 USB 连接正常。

## Programming via iMPACT

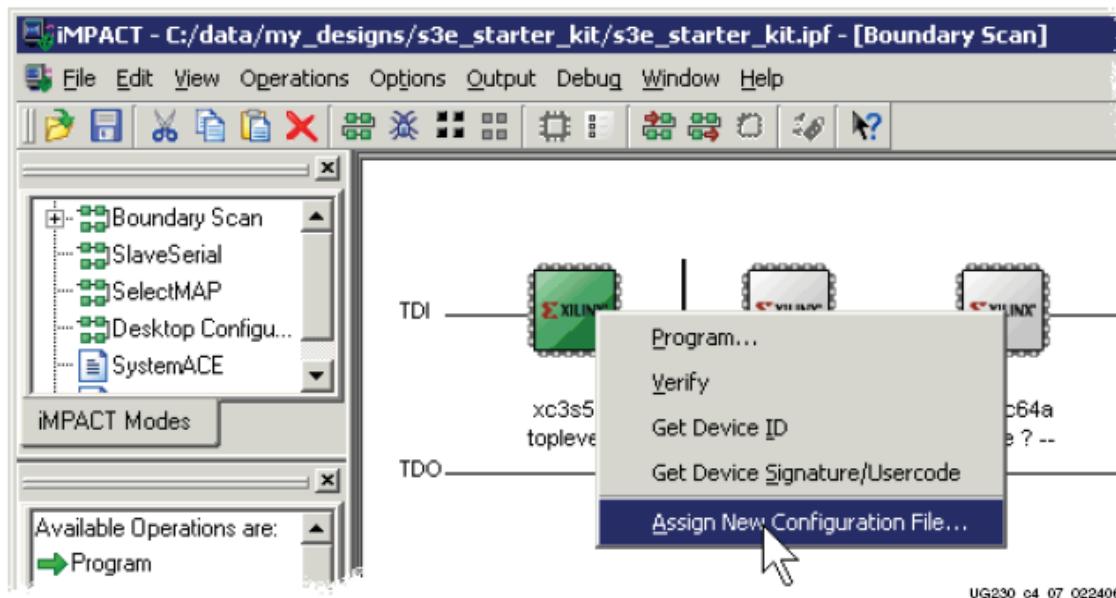
通过 iMPACT 和 USB 电缆将编译成功的 FPGA 设计下载到 FPGA 上。

将USB电缆连到板子并给其上电，然后双击Project Navigator 的 **Configure Device (iMPACT)**，如图4.5所示。



**Figure 4-5: Double-Click to Invoke iMPACT**

只要板连接正确, iMPACT就会自动识别JTAG编程文件, 如图4.6所示。如果没有即使识别出来, 可以点击连接链的第一个芯片, 然后右击FPGA, 选择**Assign New Configuration File**, 如图4.6所示。



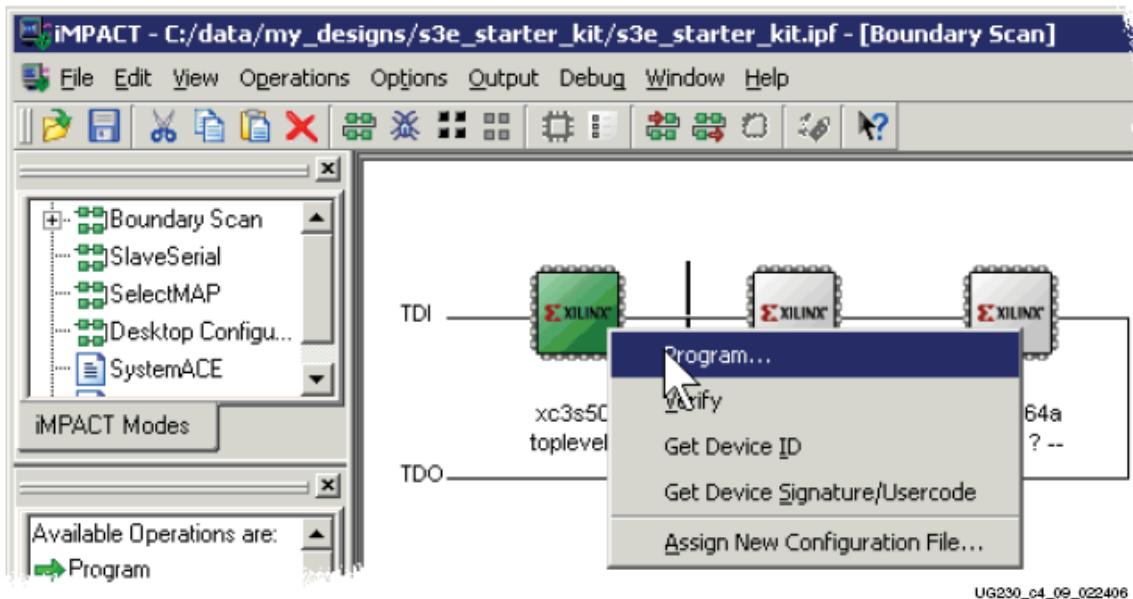
**Figure 4-6: Right-Click to Assign a Configuration File to the Spartan-3E FPGA**

如果FPGA的源配置文件采用默认的启动时钟——CCLK, iMPACT会给出警告提示, 如图4.7。该警告可以忽略。采用JTAG下载时, iMPACT 必须的启动时钟必须改为TCK JTAG时钟源。



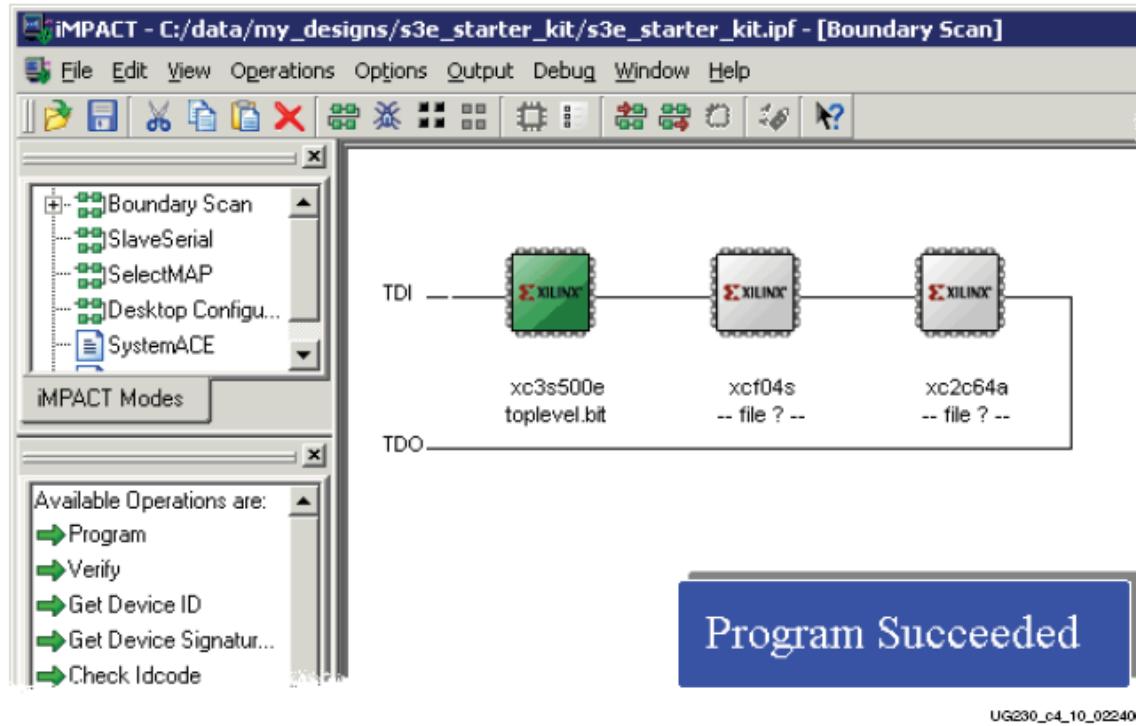
**Figure 4-7: iMPACT Issues a Warning if the StartUp Clock Was Not CCLK**

开始编程FPGA，右击FPGA，选择Program。IMPACT在编程时会自动报告进程状态。编程时间的长短取决于USB的接口 和IMPACT的设置。



**Figure 4-8: Right-Click to Program the Spartan-3E FPGA**

当FPGA编程成功时，IMPACT会显示成功，如图4.9所示。此时，可以在板上执行FPGA了，还有DONE管脚点亮。

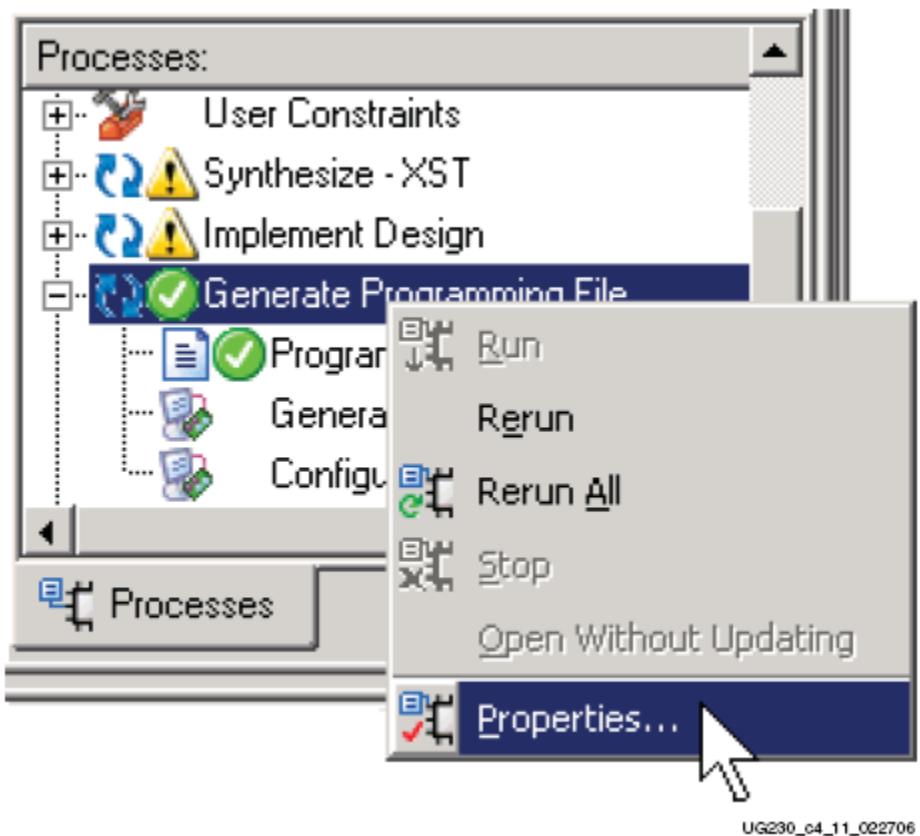


**Figure 4-9: iMPACT Programming Succeeded, the FPGA's DONE Pin is High**  
**Programming Platform Flash PROM via USB**

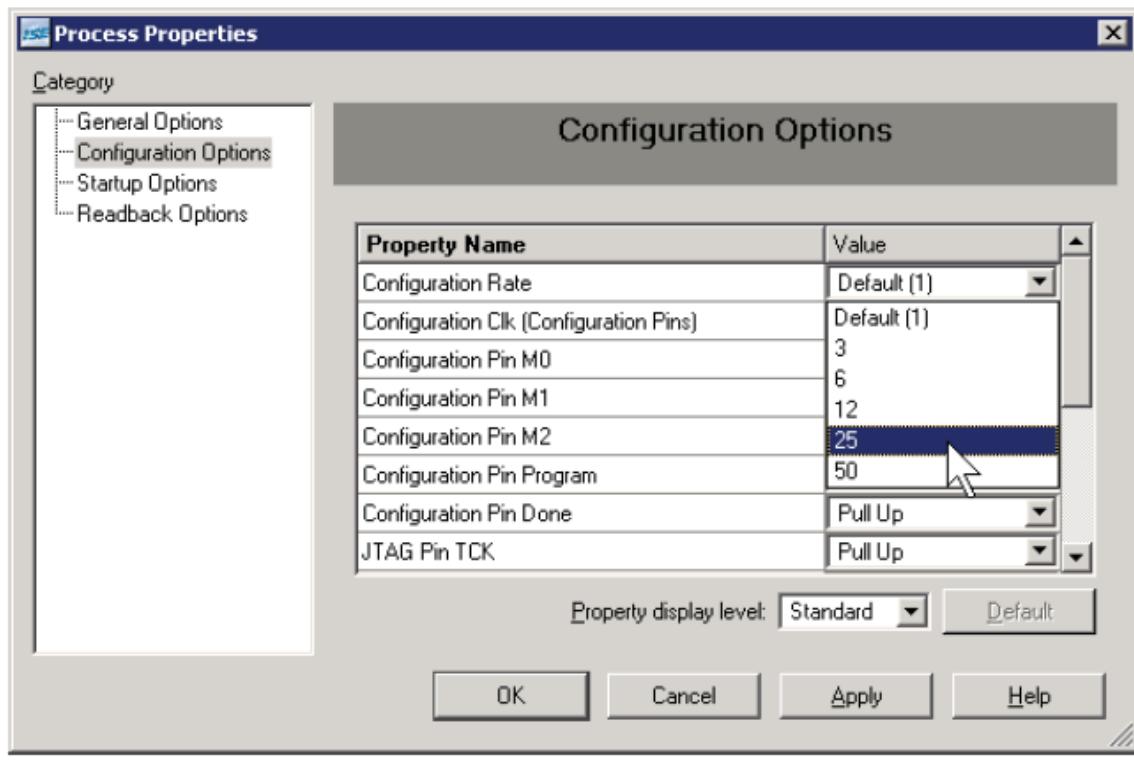
板上的USB-JTAG同样可以对Xilinx XCF04S serial Platform Flash PROM进行编程。以下是如何产生PROM文件和如何下载到FPGA上的步骤。

### Generating the FPGA Configuration Bitstream File

在产生PROM文件之前，要先生成FPGA位流文件。从外部的PROM下载到它自身时，FPGA提供了一个输出时钟—CCLK。FPGA内部的晶振CCLK的振动频率较慢，大约1.5MHz。大多数的外部PROM支持高频率时钟。增加CCLK的频率等同于减少FPGA的配置时间。Xilinx XCF04S Platform Flash支持25 MHz的CCLK频率。如图4.10所示。

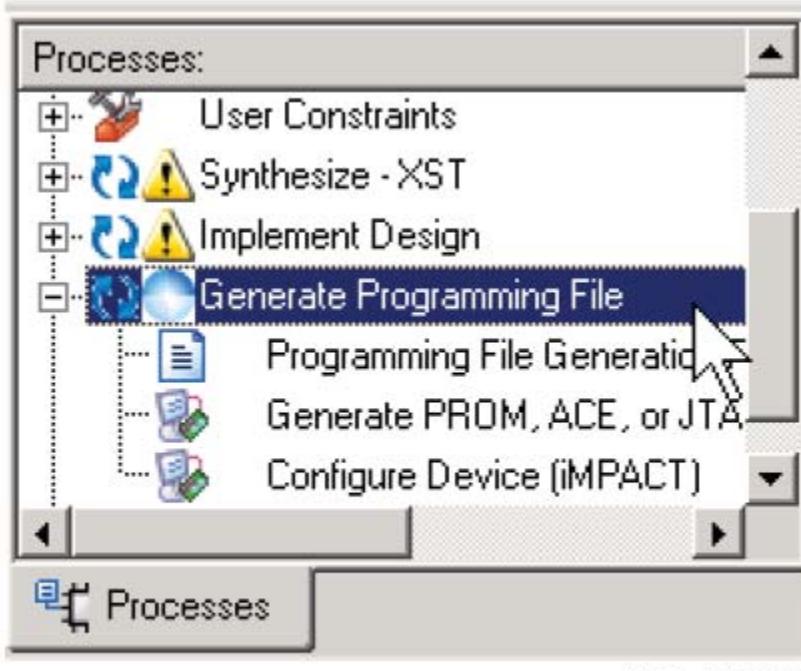


**Figure 4-10: Set Properties for Bitstream Generator**  
接着如图4.11所示进行相应的操作。



UG230\_c4\_12\_022706

**Figure 4-11: Set CCLK Configuration Rate under Configuration Options**  
接下来如图4.12所示。

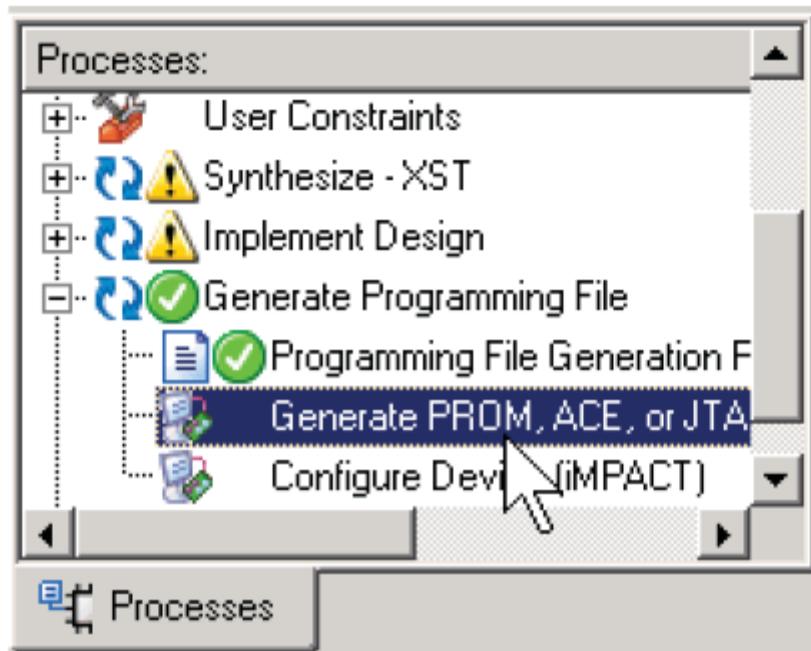


UG230\_c4\_13\_022706

**Figure 4-12: Double-Click Generate Programming File**

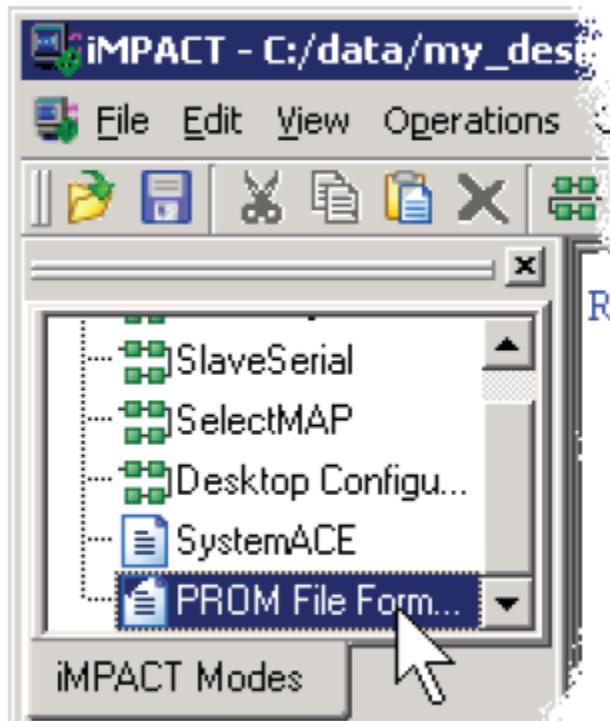
Generating the PROM File

产生程序文件之后，进行如图4.13所示的操作。



UG230\_c4\_14\_022706

**Figure 4-13: Double-Click Generate PROM, ACE, or JTAG File**  
启动IMPACT之后，双机PROM File Formatter，如图4.14所示。



UG230\_c4\_15\_022706

**Figure 4-14: Double-Click PROM File Formatter**  
接下来如图4.15、4.16、4.17、4.18进行相应的操作。

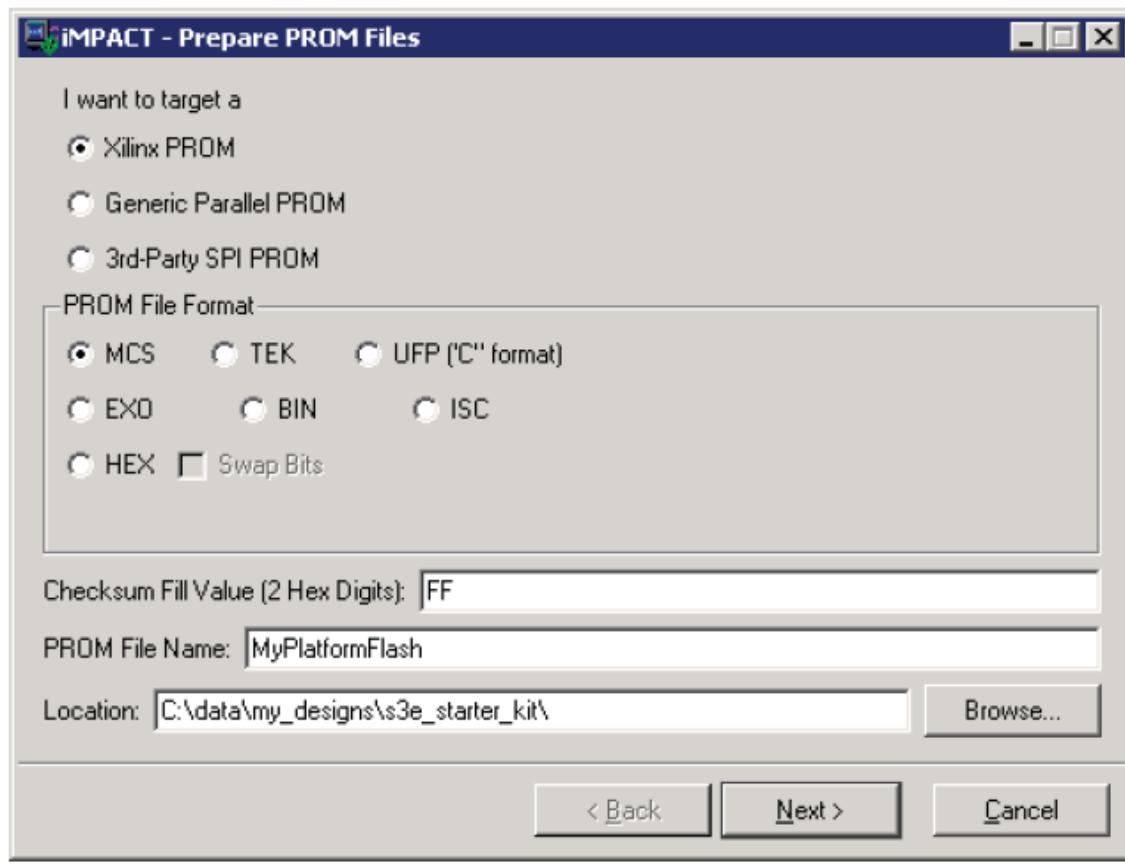


Figure 4-15: Choose the PROM Target Type, the Data Format, and File Location

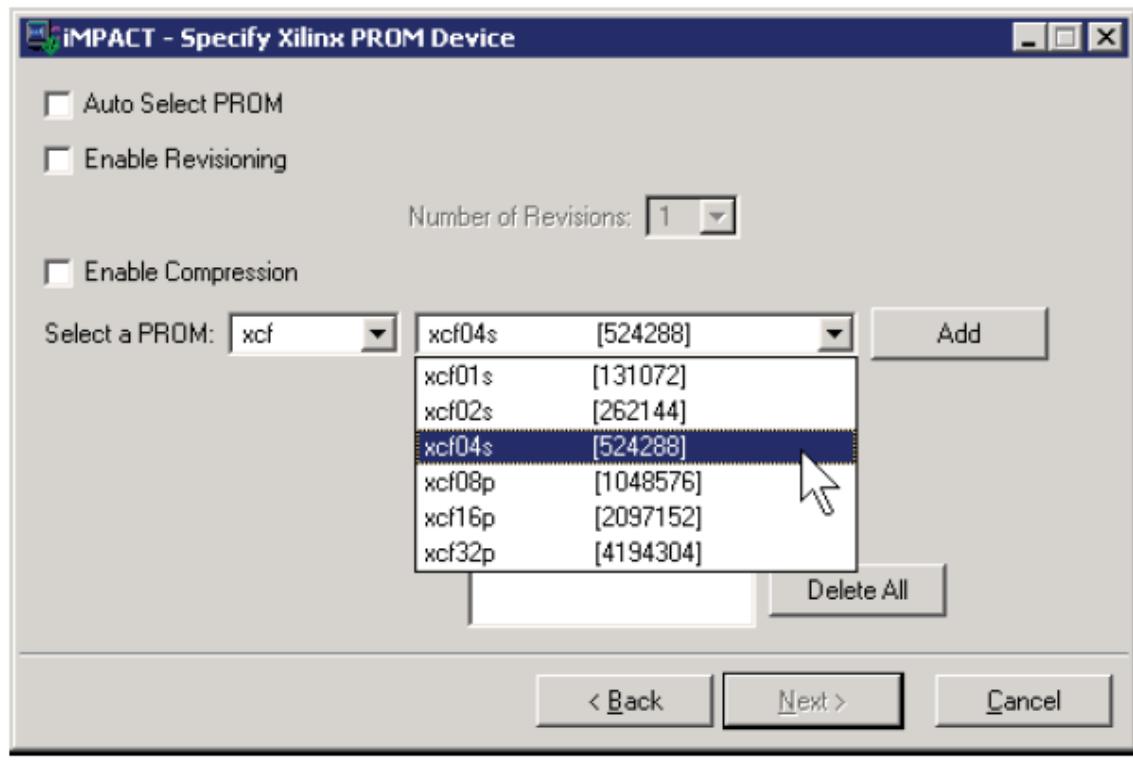


Figure 4-16: Choose the XCF04S Platform Flash PROM

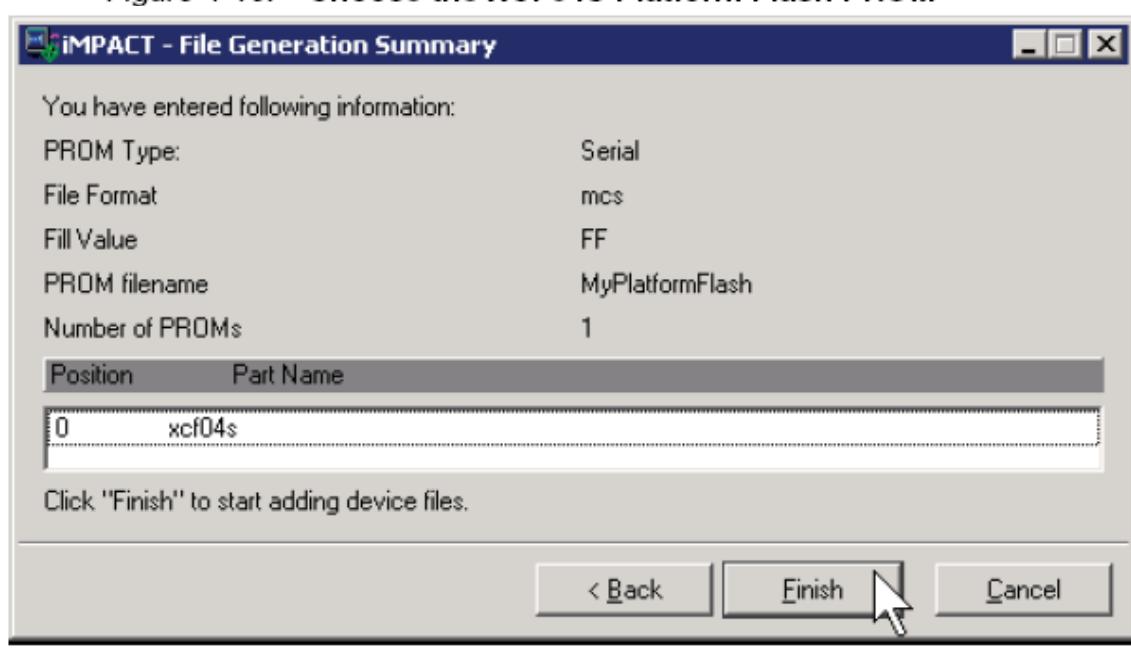
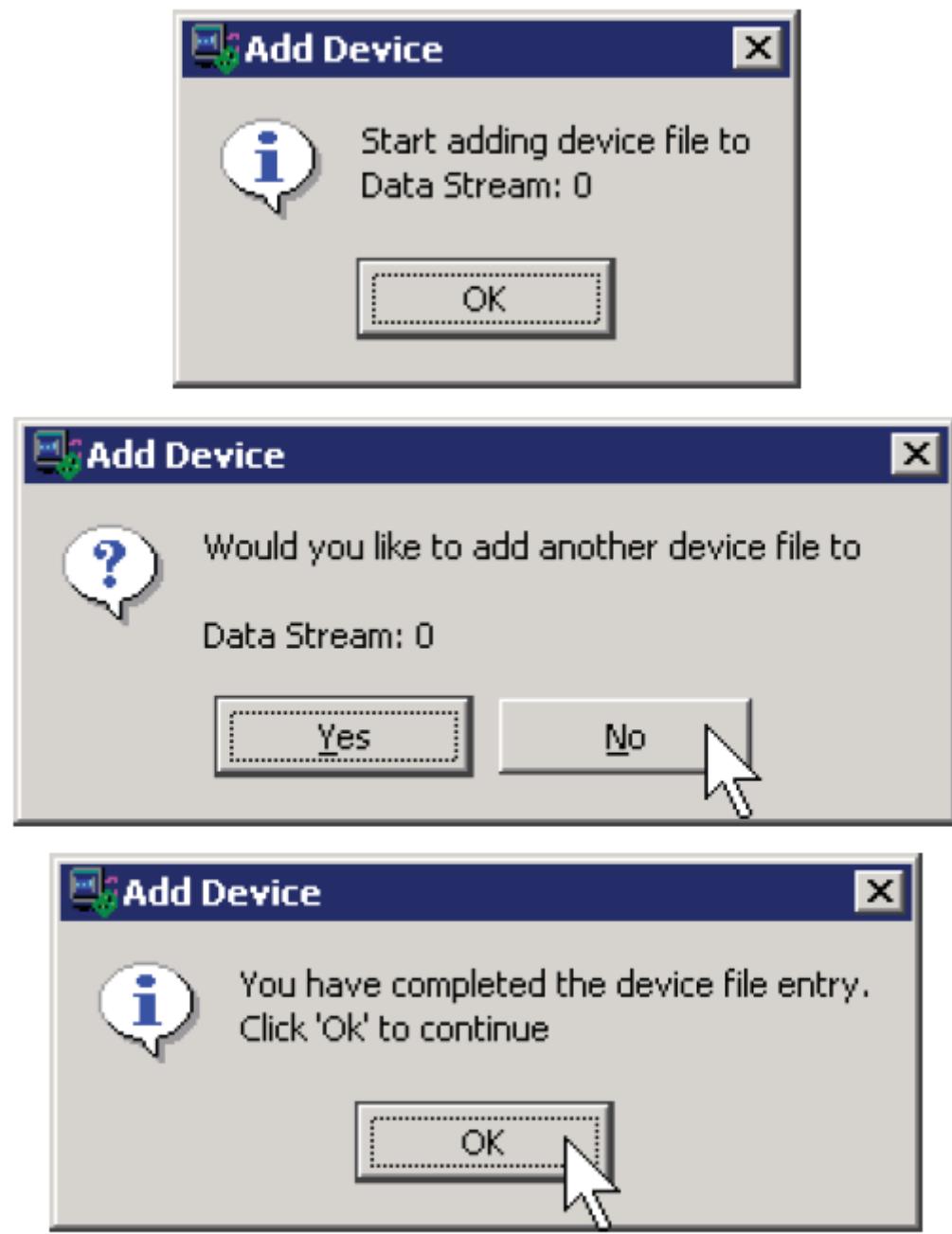
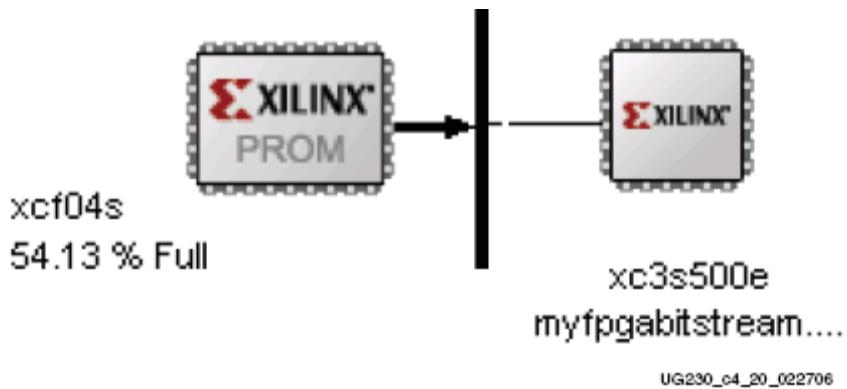


Figure 4-17: Click Finish after Entering PROM Formatter Settings



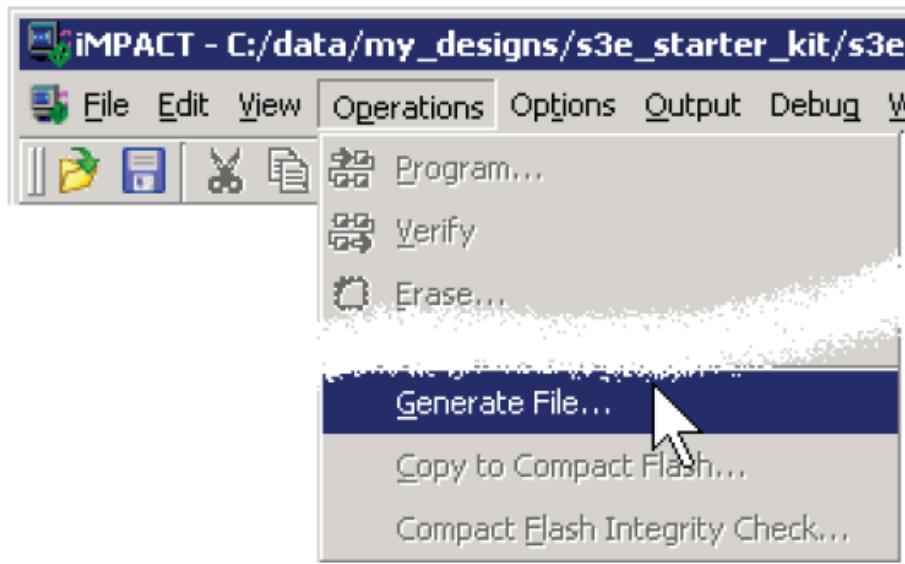
UG230\_c4\_19\_022706

**Figure 4-18: Enter FPGA Configuration Bitstream File(s)**  
PROM成功配置后，IMPACT显示位流文件占用PROM的存储空间，如图4.19所示。



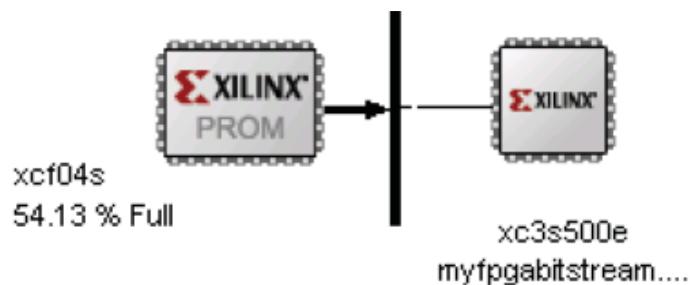
**Figure 4-19: PROM Formatting Completed**

要生成可执行PROM文件，点击Operation/Generate File，如图4.20所示。



**Figure 4-20: Click Operations → Generate File to Create the Formatted PROM File**

iMPACT将指出PROM文件成功产生，如图4.21所示。

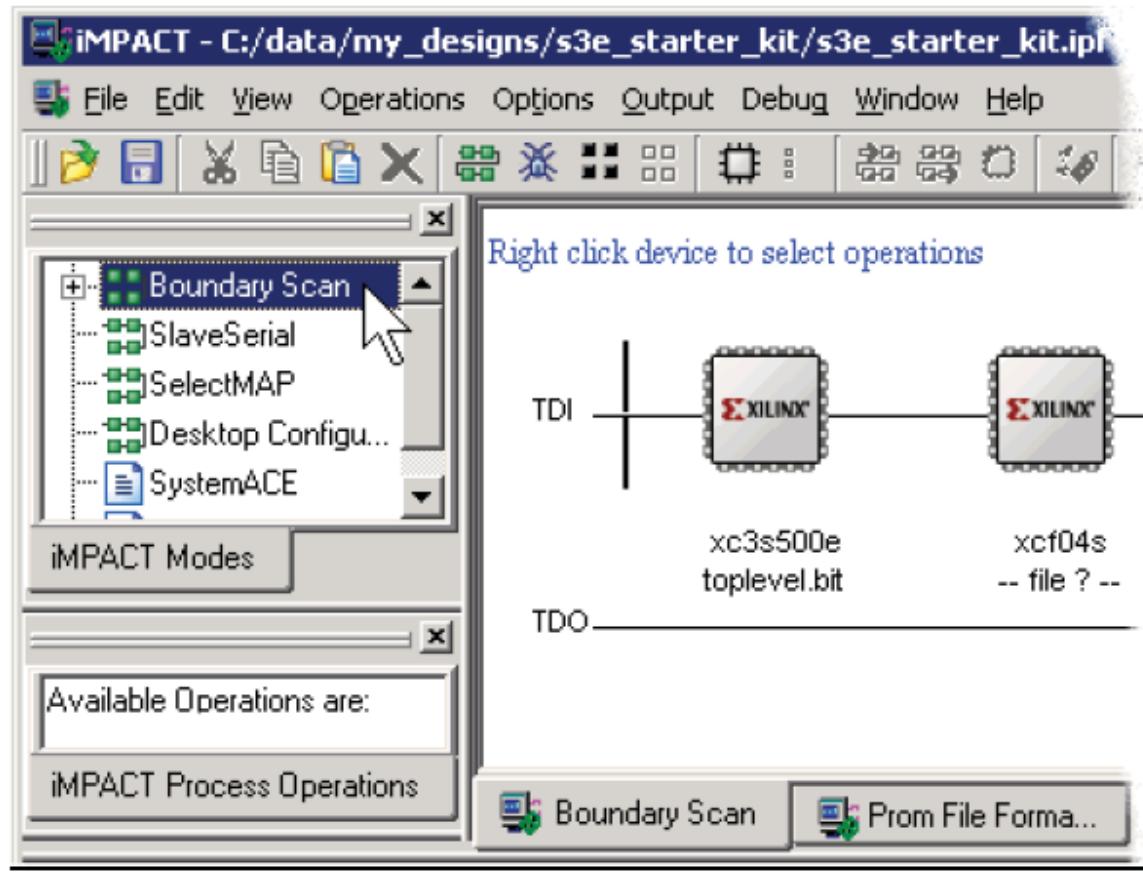


## PROM File Generation Succeeded

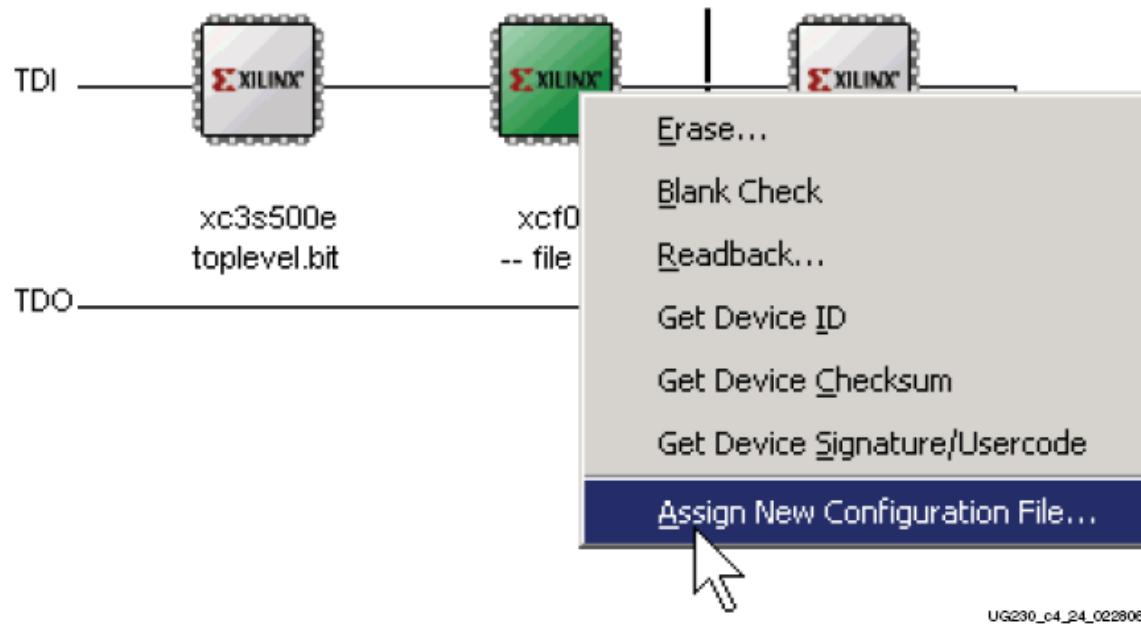
UG230\_c4\_22\_022706

**Figure 4-21: PROM File Formatter Succeeded**  
Programming the Platform Flash PROM

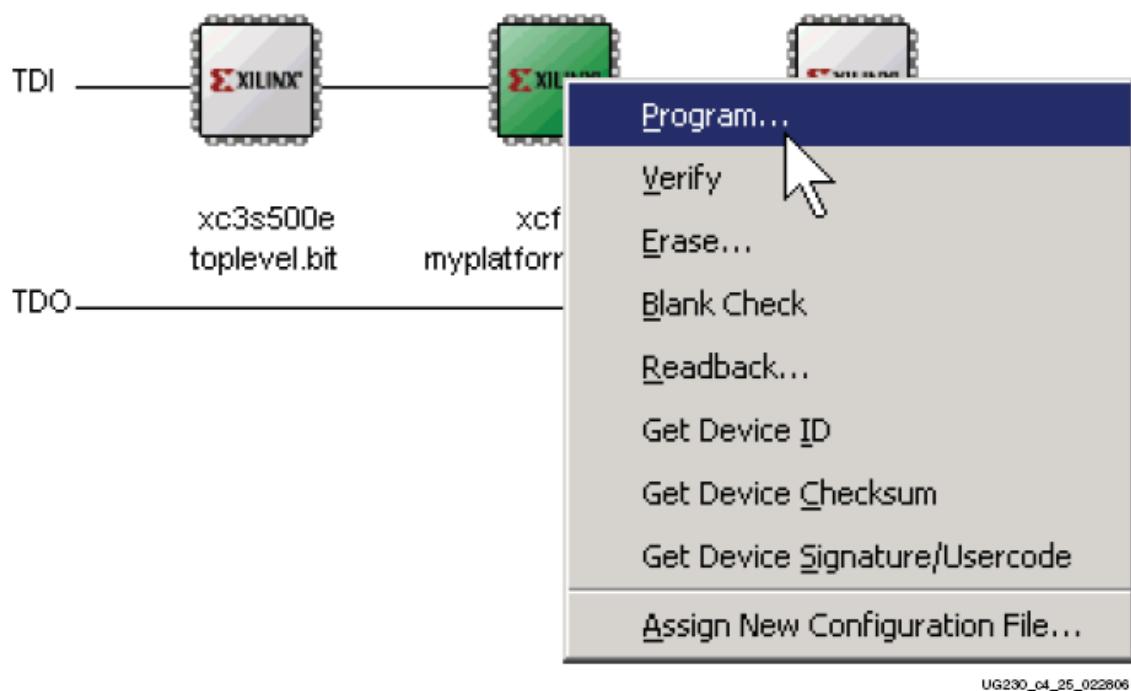
下面是通过USB—JTAG连接电路将PROM文件编程到Platform Flash PROM的简要步骤。具体如图4.22~4.25所示。



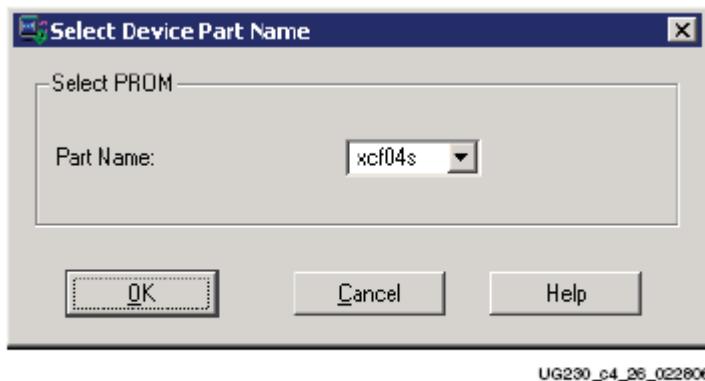
**Figure 4-22: Switch to Boundary Scan Mode**



*Figure 4-23: Assign the PROM File to the XCF04S Platform Flash PROM*



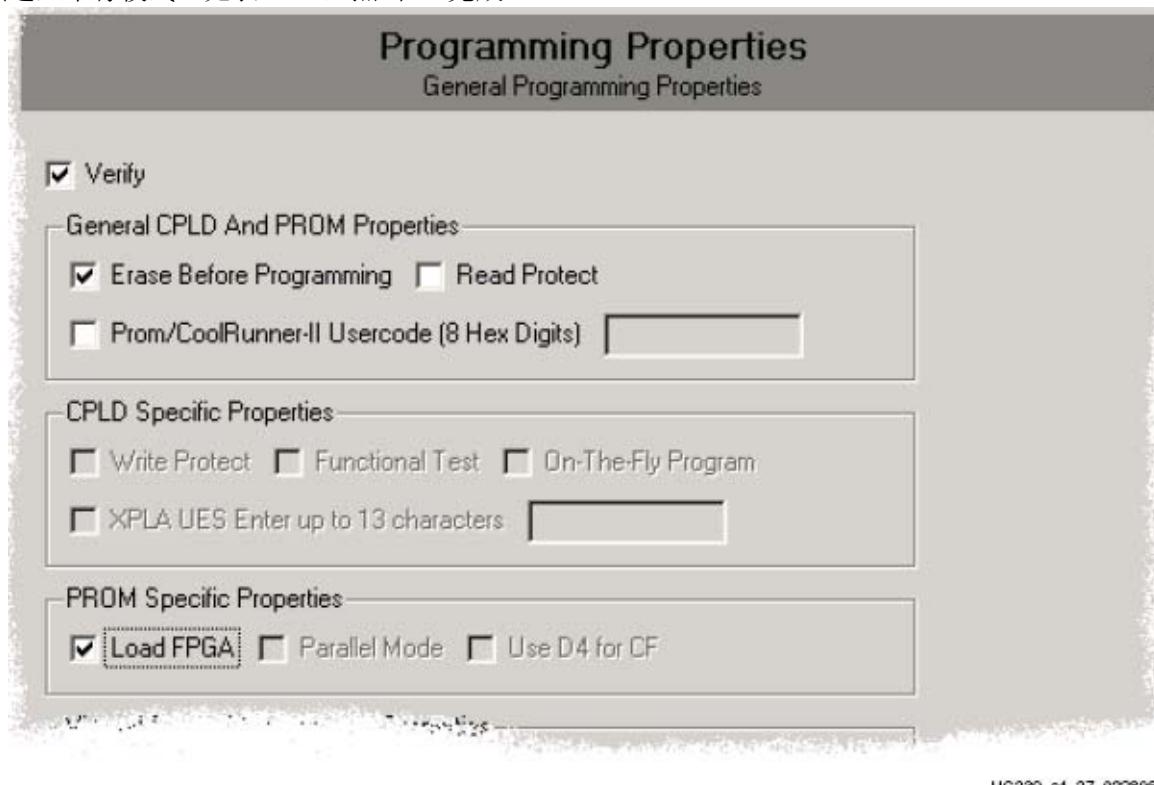
*Figure 4-24: Program the XCF04S Platform Flash PROM*



**Figure 4-25: Select XCF04S Platform Flash PROM**

在编程之前，如图4.26选择编程选项。检查Erase Before Programming选项，在编程之前完全擦除Platform Flash PROM，确保没有残余数据。Verify选项用来检查PROM正确编程和匹配下载配置位流文件。一般推荐这两个选项，即使是增加编程时间。

Load FPGA选项在Platform Flash PROM编程后立即强制重新配置FPGA。FPGA的配置模式必须选主串行模式（见表4.1），点击OK完成。



**Figure 4-26: PROM Programming Options**

IMPACT会指出编程成功与否。如果编程成功而Load FPGA选项没有检查，按下PROG\_B按钮空开关强制FPGA从Platform Flash PROM重新进行配置。如果FPGA成功配置，DONE LED 点亮。

# Chapter 5

## Character LCD Screen

### Overview

入门实验板显著的特征是 2 线 16 字符液晶显示器 LCD。尽管 LCD 支持 8 位的数据接口，为了与其它的 XILINX 的开发板保持兼容并且尽可能减少针脚数，FPGA 仅通过 4 位的数据接口线控制 LCD，如图 5.1 所示。

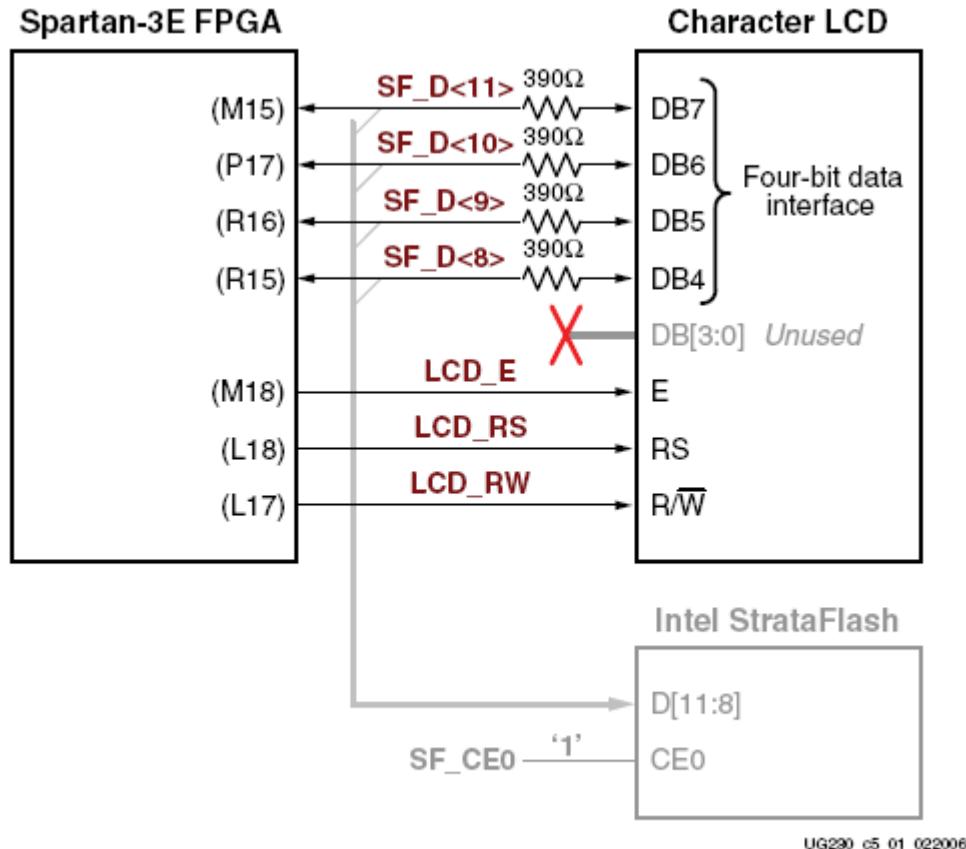


Figure 5-1: Character LCD Interface

LCD 通过使用 ASCII 标准和自定义字符可以有效地显示多种信息。但是，这些显示速度并不是很快。每半秒扫描一次以测试实际清晰度的界限。与 50MHz 时钟频率相比，这样的显示速度是慢的。PicoBlaze 处理器可以有效地控制显示时间和显示内容。

### Character LCD Interface Signals

表 5.1 给出了字符 LCD 接口的接口信号。

### Voltage Compatibility

字符 LCD 的供电电压是 +5V。FPGA 的 I/O 口信号的电压是 3.3V。但是，FPGA 的输出电平是通过 LCD 来识别是有效的低电平还是高电平。LCD 控制器接收 5V TTL 信号电平，FPGA 输出 3.3V 的 LCMOS 以满足 5V TTL 电压要求。

数据线上的 390 欧串联电阻，当 LCD 驱动一个逻辑高电平时，其用来防止了 FPGA 和 SrtataFlsah I/O 管脚的超负载。当 LCD\_RW 为高时，LCD 驱动数据线。在绝大多数应用中，LCD

作为只读外围设备，几乎没有从显示器读数据。

## Interaction with Intel StrataFlash

如图 5.1 所示，4 根 LCD 数据线与 StrataFlash 数据线 SF\_D<11:8>复用。正如表 5.2 所示，LCD/StrataFlash 存储器依赖于设计而交互使用。当存储器失能时 (SF\_CE0=1) ,FPGA 用作全读/写通道给 LCD。相反，当 LCD 读失能时 (LCD\_RW=0) ,FPGA 用作全读/写通道给存储器。

*Table 5-1: Character LCD Interface*

Signal Name	FPGA Pin	Function	
SF_D<11>	M15	Data bit DB7	Shared with StrataFlash pins SF_D<11:8>
SF_D<10>	P17	Data bit DB6	
SF_D<9>	R16	Data bit DB5	
SF_D<8>	R15	Data bit DB4	
LCD_E	M18	Read/Write Enable Pulse 0: Disabled 1: Read/Write operation enabled	
LCD_RS	L18	Register Select 0: Instruction register during write operations. Busy Flash during read operations 1: Data for read or write operations	
LCD_RW	L17	Read/Write Control 0: WRITE, LCD accepts data 1: READ, LCD presents data	

*Table 5-2: LCD/StrataFlash Control Interaction*

SF_CE0	SF_BYTE	LCD_RW	Operation
1	X	X	StrataFlash disabled. Full read/write access to LCD.
X	X	0	LCD write access only. Full access to StrataFlash.
X	0	X	StrataFlash in byte-wide (x8) mode. Upper data lines are not used. Full access to both LCD and StrataFlash.

### Notes:

- ‘X’ indicates a don’t care, can be either 0 or 1.

要是 StrataFlash 存储器为字节宽度×8 模式 (SF\_BYTE=0) ,FPGA 作为全同步读/写通道给 LCD 和存储器。在字节模式下，存储器没有用到 SF\_D<15:8>数据线。

## UCF Location Constraints

```

NET "LCD_E" LOC = "M18" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET "LCD_RS" LOC = "L18" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET "LCD_RW" LOC = "L17" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;

# The LCD four-bit data interface is shared with the StrataFlash.
NET "SF_D<8>" LOC = "R15" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET "SF_D<9>" LOC = "R16" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET "SF_D<10>" LOC = "P17" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET "SF_D<11>" LOC = "M15" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;

```

*Figure 5-2: UCF Location Constraints for the Character LCD Controller*

该2×16字符LCD内嵌一个Sitronix ST7066U图形控制器。其功能与下面几个一样：Samsung

S6A0069X or KS0066U、Hitachi HD44780和SMOS SED1278。

## Memory Map

该控制器有三个内部存储空间，每个都有专门用途。送数据给这些空间之前必须初始化。

### 1) DD RAM

显示数据RAM（DD RAM）存储字符编码。绝大多数应用中，都是与DD RAM相结合的。存储在DD RAM中的字符编码所涉及的特定的字符位图要么存在CG ROM字符设置中，要么存在用户自定义的CG RAM的字符设置中。

图5.3给出了显示器32位字符位置的默认地址。字符的最上行存储在地址0X00与0X0F之间。第二行的字符存储在地址0X40与0X4F之间。

	Character Display Addresses																Undisplayed Addresses		
1	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	...	27
2	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	...	67
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	...	40

Figure 5-3: DD RAM Hexadecimal Addresses (No Display Shifting)

从物理上讲，DD RAM 一共有 80 个字符位置，每行有 40 个字符。位置 0X10 到 0X27 和 0X50 到 0X67 之间的地址用来存储其它非显示数据。此外，这些位置也可以存储只有使用控制器的显示移位功能才能显示的字符。

往 DD RAM 读或写之前，DD RAM 地址命令得初始化地址计数器。写 DD RAM 数据使用写数据到 CG RAM 或 DD RAM 命令，读 DD RAM 使用从 CG RAM 或 DD RAM 命令读数据。

DD RAM 地址计数器要么在读或写之后保持常数，要么自动增加 1 或自动减 1。

### 2) CG ROM

字符产生器 ROM（CG ROM）包括每个事先定好的字符的字体位图，这样 LCD 屏才能显示，如图 5.4。字符编码存储在 DD RAM 中，每个字符的位置与 CG ROM 的位置按顺序对应。例如，0X53 的一个 16 进制的字符编码存储在 DD RAM 中的位置显示字符是“S”。0X53 最上面的轻咬位（高半位）等同于 DB[7:4]=0101 和最低的轻咬位（低半位）等同于 DB[3:0]=0011。如图 5.4 所示，字符“S”就显示在屏幕上。

英语/罗马字符存储在 CG ROM 相应的 ASCII 编码地址中。

字符 ROM 存储 ASCII 英文字符和日本字符。

控制器同样提供了 8 位自定义字符位图，存储在 CG RAM 中。这些 8 位的自定义字符编码显示时存储在 DD RAM 的 0X00 与 0X07 之间。

Upper Data Nibble			
DB7	DB6	DB5	DB4
0 0 0 0 0 0 0 1 1 1 1 1 1	0 0 0 1 1 1 1 0 0 1 1 1 1	0 1 1 0 0 1 1 1 1 0 0 1 1	0 0 1 0 1 0 1 0 1 0 1 0 1
xxxxx0000	CG RAM	0@P^P-タミαP	
xxxxx0001		!1AQAa@アチ4äd	
xxxxx0010		"2BRBr「イツメB@	
xxxxx0011		#3CScs」ウテモセ~	
xxxxx0100	CG RAM	\$4DTdt、エトヤムΩ	
xxxxx0101		%5EUeu・オナユGÜ	
xxxxx0110		&6FUVfυヲカニヨPΣ	
xxxxx0111		'7GW9wアキヌラQπ	
xxxxx1000		(8HXh×イクネリJX	
xxxxx1001		)9IViワテノル"4	
xxxxx1010		*:JZjズエコハレ i?	
xxxxx1011		+;K[k<オサヒロ×万	
xxxxx1100		,<L¥1 ヤシフワΦ円	
xxxxx1101		-=M]m)ユズヘンモ÷	
xxxxx1110		.>N^n→ヨセホ^R	
xxxxx1111		/?0_o←ソマ"ö■	
DB3 DB2 DB1 DB0			UG3230_c5_02_090906

Figure 5-4: LCD Character Set

### 3) CG RAM

字符产生器 RAM(CG RAM)提供空间给 8 位的自定义字符位图。每个自定义字符位由 8 行位图的 5 个点组成，如图 5.5 所示。

往 CG RAM 读或写之前 CG RAM 地址命令得初始化地址计数器。写 CG RAM 数据使用写数据到 CG RAM 或 DD RAM 命令，读 CG RAM 使用从 CG RAM 或 DD RAM 命令读数据。CG RAM 地址计数器要么在读或写之后保持常数，要么自动增加 1 或自动减 1。

图 5.5 举了个例子，产生一个特殊的西洋跳棋盘字符。自定义字符存储在第四 CG RAM 字符位置中，当 DD RAM 的位置是 0x03 时，其显示。写自定义字符时，使用设置 CG RAM 地址命令初始化 CG RAM 地址。前三行的地址位对应自定义字符位。后三行位对应字符地址的行地址。写数据到 CG RAM 或 DD RAM 命令用来写每个字符位行。“1”表示点亮。“0”表示熄灭。只有低 5 位的数据被用到。高三位的数据与位置无关。第 8 行的数据位一般为 0 以适于指针之用。

						Upper Nibble			Lower Nibble				
						Write Data to CG RAM or DD RAM							
A5	A4	A3	A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0
Character Address	Row Address		Don't Care			Character Bitmap							
0	1	1	0	0	0	-	-	-	0	0	0	0	0
0	1	1	0	0	1	-	-	-	0	0	0	0	0
0	1	1	0	1	0	-	-	-	0	0	0	0	0
0	1	1	0	1	1	-	-	-	0	0	0	0	0
0	1	1	1	0	0	-	-	-	0	0	0	0	0
0	1	1	1	0	1	-	-	-	0	0	0	0	0
0	1	1	1	1	0	-	-	-	0	0	0	0	0
0	1	1	1	1	1	-	-	-	0	0	0	0	0

Figure 5-5: Example Custom Checkerboard Character with Character Code 0x03 Command Set

表 5.3 简要的说明了 LCD 控制器的命令和位的定义。由于该显示屏是 4 位操作，每 8 位命令被送到 2 个 4 位的轻咬位（2 个半位）。高半位先送，低半位后送。

Table 5-3: LCD Character Display Command Set

Function	LCD_RS	LCD_RW	Upper Nibble				Lower Nibble			
			DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Clear Display	0	0	0	0	0	0	0	0	0	1
Return Cursor Home	0	0	0	0	0	0	0	0	1	-
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S
Display On/Off	0	0	0	0	0	0	1	D	C	B
Cursor and Display Shift	0	0	0	0	0	1	S/C	R/L	-	-
Function Set	0	0	0	0	1	0	1	0	-	-
Set CG RAM Address	0	0	0	1	A5	A4	A3	A2	A1	A0
Set DD RAM Address	0	0	1	A6	A5	A4	A3	A2	A1	A0
Read Busy Flag and Address	0	1	BF	A6	A5	A4	A3	A2	A1	A0
Write Data to CG RAM or DD RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0
Read Data from CG RAM or DD RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0

1) 失能

如果 LCD\_E 使能信号为低，所有其它输入 LCD 信号全被忽视。

2) 清屏

清屏后指针返回到原始位置——最左上角。该命令写一个空白内容（ASCII/ANSI 字符编码为 0x20）给所有的 DD RAM 地址。DD RAM 的 0X00 地址计数器置 0。清除所有的选择设置。I/D 控制位置 1（地址自动增加模式）。

执行时间：82us~1.64ms

3) 返回指针原始位

指针返回原始位——最左上角。DD RAM 的内容不受影响。所有的显示被移到原始位，如图 5.3 所示。DD RAM 的 0X00 地址计数器置 0。如果移位，显示返回原始状态。指针或光标移到字符位的最左上角。

执行时间：40us~1.6ms

#### 4) 进入模式设置

设置指针移动的方向，并规定是否移动显示。在读和写数据时，这些操作就完成了。

执行时间：40us

##### **Bit DB1: (I/D) Increment/Decrement**

0	Auto-decrement address counter. Cursor/blink moves to left.
1	Auto-increment address counter. Cursor/blink moves to right.

在每次写数据给 CG RAM 或 DD RAM 或从 CG RAM 或 DD RAM 读数据后，该位的 DD RAM 和 CG RAM 地址计数器要么自动增加 1 要么自动减少 1. 指针或光标的位置随之移动。

##### **Bit DB0: (S) Shift**

0	Shifting disabled
1	During a DD RAM write operation, shift the entire display value in the direction controlled by Bit DB1 (I/D). Appears as though the cursor position remains constant and the display moves.

#### 1) 显示关/断

显示关或断，控制所有的字符、指针和指针位置的字符光标。

执行时间：40us

##### **Bit DB2: (D) Display On/Off**

0	No characters displayed. However, data stored in DD RAM is retained
1	Display characters stored in DD RAM

指针使用字符最底行的 5 个点。指针出现在显示字符的下面。

##### **Bit DB1: (C) Cursor On/Off**

0	No cursor
1	Display cursor

##### **Bit DB0: (B) Cursor Blink On/Off**

0	No cursor blinking
1	Cursor blinks on and off approximately every half second

#### 2) 指针和显示移动

移动指针和显示并不改变 DD RAM 的内容。移动指针位置或显示往左或往右时并不需要写或读显示数据。

指针的位置功能是为了修改个别的字符，或向左或右滚动窗口来显示存储在 DD RAM 中的额外数据，可以移到每行的第 16 个符。当它移到第一行的第 40 个字符之处时，指针自动移到第二行。两行的显示移动在同一时间进行。

当显示数据重复移动时，两行水平移动。第二行不会移到第一行。

执行时间: 40us

Table 5-4: Shift Patterns According to S/C and R/L Bits

DB3 (S/C)	DB2 (R/L)	Operation
0	0	Shift the cursor position to the left. The address counter is decremented by one.
0	1	Shift the cursor position to the right. The address counter is incremented by one.
1	0	Shift the entire display to the left. The cursor follows the display shift. The address counter is unchanged.
1	1	Shift the entire display to the right. The cursor follows the display shift. The address counter is unchanged.

3) 功能设置

设置接口数据的长度，每行显示的个数，字符的字体。入门实验板支持单功能设置，其值为 0X28。

执行时间: 40us

7) 设置 CG RAM 地址

设置 CG RAM 的初始地址。该命令后，以后所有往显示屏的读或写操作的数据来自或去往 CG RAM。

执行时间: 40us

8) 设置 DD RAM 地址

设置 DD RAM 的初始地址。该命令后，以后所有往显示屏的读或写操作的数据来自或去往 DD RAM。

执行时间: 40us

9) 读忙标志和地址

读忙标志 (BF) 用来判断内部操作是否在进行，并读当前地址计数器的内容。

BF=1 说明内部操作正在进行。下个指令不被接收直到 BF 被清 0 或直到当前指令达到最大的执行时间。

该命令返回当前地址计数器的值。地址计数器为 CG RAM 和 DD RAM 所用。具体内容取决于最新公布的设置 CG RAM 地址或设置 DD RAM 设置命令。

执行时间: 1us

10) 写数据给 CG RAM 或 DD RAM

要是该命令在设置 DD RAM 地址命令之后，则写数据给 DD RAM；或是该命令在设置 CGRAM 地址命令之后，则写数据给 CGRAM。根据进入设置模式，在写操作之后，地址自动加 1 或自动减 1。进入设置模式同样决定显示移动。

执行时间: 40us

11) 从 CG RAM 或 DD RAM 读数据

要是该命令在设置 DD RAM 地址命令之后，则从 DD RAM 读数据；或是该命令在设置 CGRAM 地址命令之后，从 CG RAM 读数据。根据进入设置模式，在写操作之后，地址自动加 1 或自动减 1。进入设置模式同样决定显示移动。

执行时间: 40us

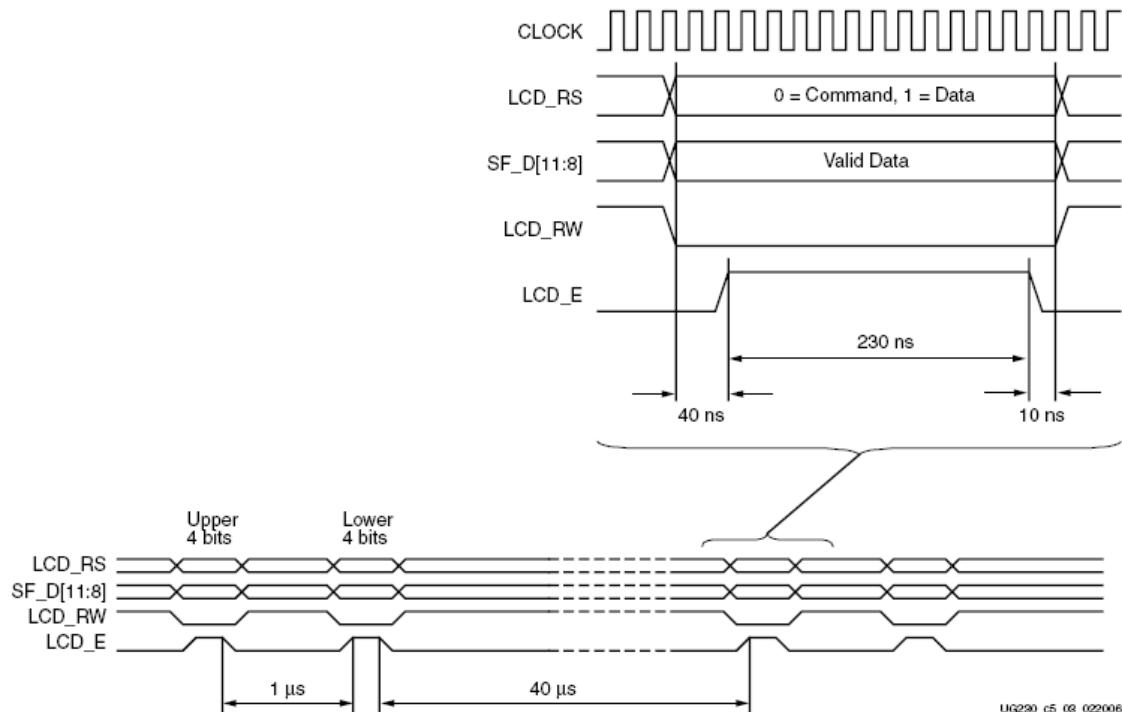
## Operation

### Four-Bit Data Interface

该板使用 4 位数据接口给字符 LCD。图 5.6 说明了向 LCD 写操作建立、保持允许的最长时间以及使能脉冲对时钟 (50MHz 或 T=20ns) 的偏移时间。

DF\_D<11:8>的数据值，寄存器选择信号（LCD\_RS）以及读/写（LCD\_RW）控制信号必须建立并在使能信号 LCD\_E 转向高电平之前至少稳定 40ns。使能信号必须保留高电平 230ns 或更长时间——等于或超过 12 时钟周期（50MHz）。

在许多应用中，LCD\_RW 信号可以永远置低，因为 FPGA 一般不会从显示屏读取数据。



**Figure 5-6: Character LCD Interface Timing**  
**Transferring 8-Bit Data over the 4-Bit Interface**

在初始化显示屏和建立通信之后，所有的命令和数据以 8 位形式传送给字符显示屏——用 2 个连续的 4 位传送。每个 8 位传送必须分为 2 个 4 位，其间隔时间至少 1us，如图 5.6 所示。先传高半位，再传低半位。一个 8 位的写操作在下个通信之前必须间隔至少 40us。在清屏命令之后，该延时必须增至 1.64ms。

### Initializing the Display

上电后，显示屏必须初始化建立所需的通信协议。该初始化步骤简单，完全适合高效的 8 位 PicoBlaze 嵌入式控制器。初始化之后，PicoBlaze 控制器除了简单驱动显示屏外，还用来更为复杂的控制或计算。

#### 1) 上电初始化

初始化的第一步是建立 FPGA 与 LCD 的 4 位的数据接口，具体如下：

- A: 等待 15ms 或更长，尽管 FPGA 完成配置后显示屏一般处于准备就绪状态。在 50MHz 时，15ms 时间等于 750000 时钟周期。
- B: 写 SF\_D<11:8>=0x3，LCD\_E 保持高电平 12 时钟周期。
- C: 等待 4.1ms 或更长，即在 50MHz 时，205000 时钟周期。
- D: 写 SF\_D<11:8>=0x3，LCD\_E 保持高电平 12 时钟周期。
- E: 等待 100us 或更长，即在 50MHz 时，5000 时钟周期。
- F: 写 SF\_D<11:8>=0x3，LCD\_E 保持高电平 12 时钟周期。
- G: 等待 40us 或更长，即在 50MHz 时，2000 时钟周期。
- H: 写 SF\_D<11:8>=0x2，LCD\_E 保持高电平 12 时钟周期。
- I: 等待 40us 或更长，即在 50MHz 时，2000 时钟周期。

## 2) 显示屏配置

上电初始化完成后，4位的数据接口就建立了。下一步就是配置显示屏了：

- A: 发一个功能设置命令，0x28，配置显示屏。
- B: 发一个进入模式命令，0X06，设置显示屏自动增地址指针。
- C: 发一个显示开/断命令，0x0c，开显示屏并失能指针和光标。
- D: 最后，发清屏命令，此后等待至少 1.64ms (82000 时钟周期)。

## Writing Data to the Display

写数据给显示屏，指定初始地址，紧接着是一个或多个数据值。写任何数据之前，发送一个设置 DD RAM 地址命令给 DD RAM 中指定的初始 7 位地址。见图 5.3 所示。

使用写数据给显示屏使用 CG RAM 或 DD RAM 命令。8 位数据值通过查表地址送给 CG ROM 或 CG RAM，如图 5.4 所示。CG ROM 或 CG RAM 中存储的位图驱动  $5 \times 8$  点阵给相应的字符。

如果地址计数器配置为自动增 1，正如前面所说的，这种应用方法可以次序写多个字符编码，每个字符自动存储并显示在下个位置。

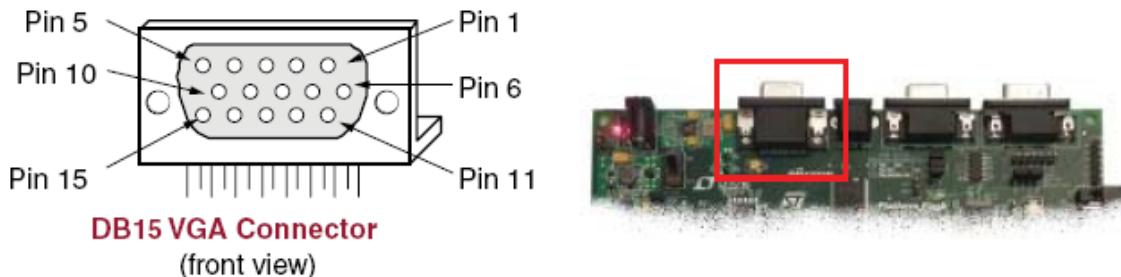
继续写字符，但是，在第一显示行的最后停止。剩余的数据不会自动在第二行显示，因为 DD RAM 的映射从第一行到第二行并不是连续的。

## Disabling the Unused LCD

如果 FPGA 的应用不使用 LCD 显示屏，置低 LCD\_E 管脚，失能它。同样，置低 LCD\_RW 管脚，阻止 LCD 屏回传数据。

# Chapter 6: VGA Display Port

入门实验板通过 DB15 连接器输出一个 VGA 显示端口。该端口通过标准的监视电缆直接连到 PC 监视器或平板 LCD 上。如图 6.1 所示，VGA 连接器在板上的最左端。



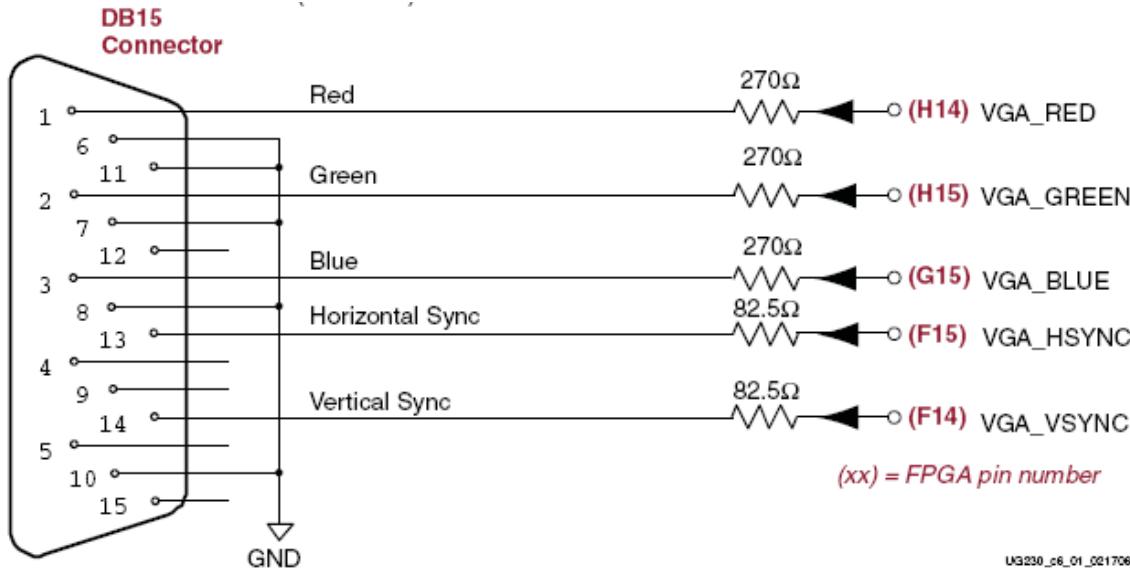


Figure 6-1: VGA Connections from Spartan-3E Starter Kit Board

FPGA通过串联电阻直接驱动5个VGA信号。每个颜色信号串一个电阻，每位的颜色信号分别是VGA\_RED、VGA\_BLUE、VGA\_GREEN。每个电阻与终端的75欧电缆电阻相结合，确保颜色信号保持在VGA规定的0V~0.7V之间。VGA\_HSYNC和VGA\_VSYNC信号使用LVTTL或LVCMS3I/O标准驱动电平。通过VGA\_RED、VGA\_BLUE、VGA\_GREEN置高或低来产生8中颜色，如表6.1所示。

Table 6-1: 3-Bit Display Color Codes

VGA_RED	VGA_GREEN	VGA_BLUE	Resulting Color
0	0	0	Black
0	0	1	Blue
0	1	0	Green
0	1	1	Cyan
1	0	0	Red
1	0	1	Magenta
1	1	0	Yellow
1	1	1	White

VGA 信号的时序由视频电气标准委员会（VESA）规定。以下提供的 VGA 系统和时序信息作为例子来说明 FPGA 在 640×480 模式下是如何驱动 VGA 监视器的。

## Signal Timing for a 60 Hz, 640x480 VGA Display

VGA 显示器基于 CRT，使用调幅模式，移动电子束（或阴极射线）在荧光屏上显示信息。LCD 使用矩阵开关给液晶加压，在每个像素点上通过液晶来改变光的介电常数。尽管下面的描述仅限于 CRT，LCD 已经发展到可以同 CRT 使用同样的时序信号了。因此，下面的讨论均适合 CRT 和 LCD。

在 CRT 显示器中，电流的波形通过蹄形磁铁产生磁场，使得电子束偏转，光栅在显示屏上横向显示，水平方向从左至右，垂直方向从上至下。如图 6.2 所示，当电子束向正方向移动时，信息才显示，即从左至右、从上至下。如果电子束从后返回左或顶边，显示屏并不显示任何信

息。在消隐周期——电子束重新分配和稳定于新的水平或垂直位时，丢失了许多信息。

显示协议定义了电子束的大小以及通过显示屏的频率，该频率是可调的。

现在的 VGA 显示屏支持多种显示协议，VGA 控制器通过协议产生时序信号来控制光栅。控制器产生同步脉冲 TTL 电平来设置电流通过偏转磁铁的频率，以确保像素或视频数据在适当的时间送给电子枪。

视频数据一般来自重复显示存储器中一个或多个字节——它们被分配到每个像素单元。入门实验板使用每个像素中的 3 位，产生表 6.1 中可能的一种颜色。控制器指定视频数据缓冲器以备电子束通过显示屏。然后，控制器接收并利用视频数据在适当的时间显示，电子束移动到指定的像素点。

根据图 6.2 所示，VGA 控制器产生水平同步时序信号 (HS) 和垂直同步时序信号 (VS)，调节在每个像素时钟视频数据的传送。像素时钟定义了显示像素信息的有效时间段。VS 信号定义显示的更新频率，或刷新屏幕信息的频率。最小的刷新频率是取决于显示器的亮度和电子束的强度，实际频率一般在 60~120Hz 之间。给定的刷新频率的水平线的数量定义了水平折回频率。

## VGA Signal Timing

表6.2的时序信号是 $640 \times 480$ ，像素时钟 $25\text{Mhz}$ ，刷新频率 $60\text{Hz} \pm 1$ 。图6.3说明了每个时序的联系。同步脉冲的时序宽度 (Tpw) 和前后门控的间隔 (Tfp 和 Tbp) 基于观察多种 VGA 的显示屏的结果。前后门控的间隔是前后同步脉冲时间。在这些时段信息不能显示。

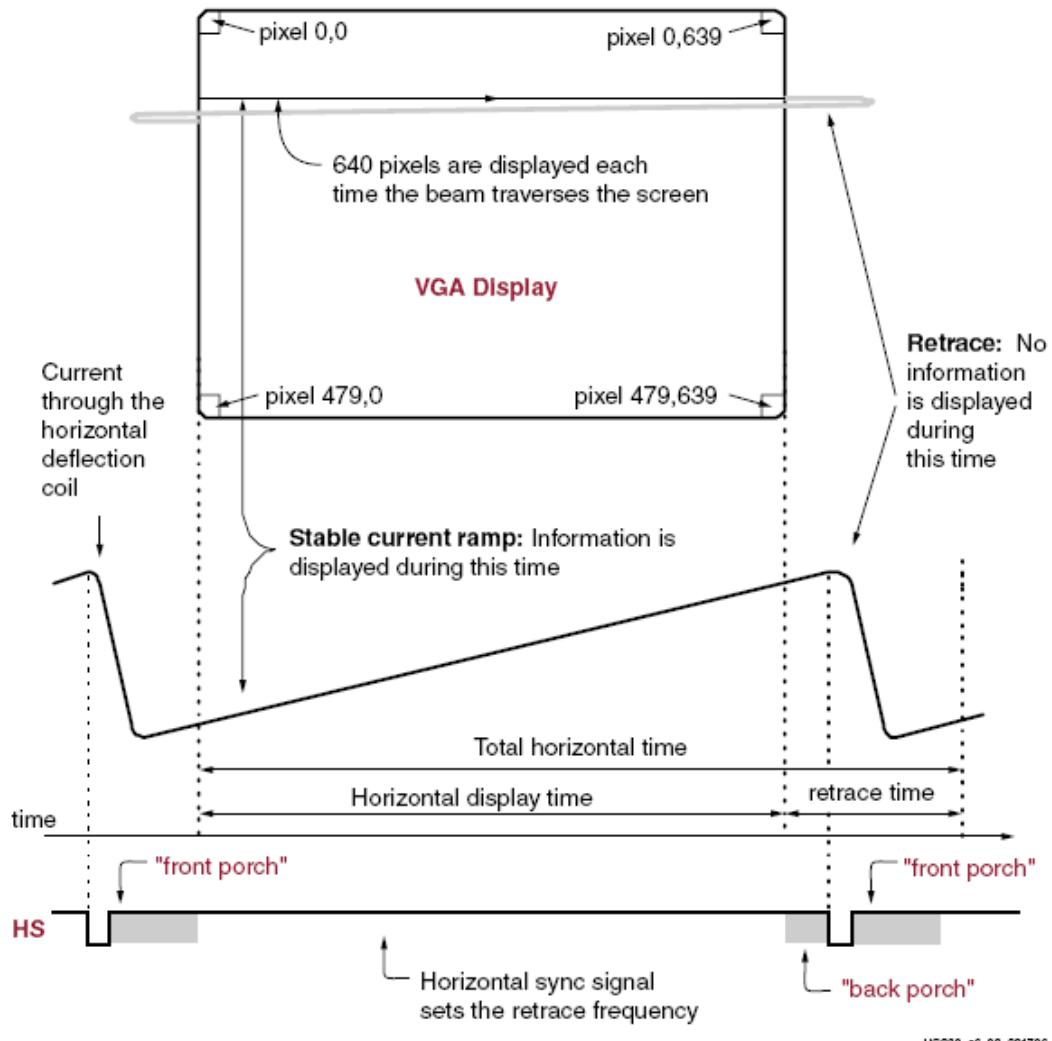
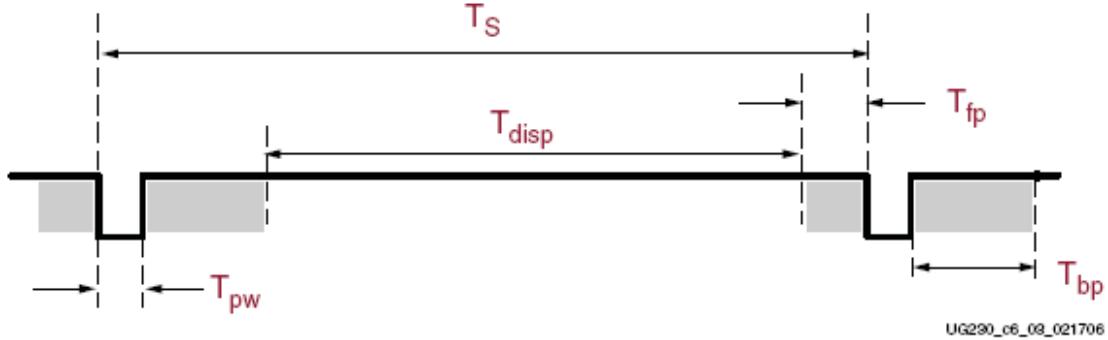


Figure 6-2: CRT Display Timing Example

Table 6-2: 640x480 Mode VGA Timing

Symbol	Parameter	Vertical Sync			Horizontal Sync	
		Time	Clocks	Lines	Time	Clocks
$T_S$	Sync pulse time	16.7 ms	416,800	521	32 $\mu$ s	800
$T_{DISP}$	Display time	15.36 ms	384,000	480	25.6 $\mu$ s	640
$T_{PW}$	Pulse width	64 $\mu$ s	1,600	2	3.84 $\mu$ s	96
$T_{FP}$	Front porch	320 $\mu$ s	8,000	10	640 ns	16
$T_{BP}$	Back porch	928 $\mu$ s	23,200	29	1.92 $\mu$ s	48



*Figure 6-3: VGA Control Timing*

一般来说，时钟计数器通过像素时钟来控制水平时序信号。译码计数器的值产生 HS 信号。在指定的行，计数器产生当前像素显示的位置。

一个独立的计数器产生垂直时序信号。垂直同步计数器在每个 HS 脉冲信号来临时自动加 1，译码值产生 VS 信号。计数器产生当前显示行。这两个计数器从地址到显示缓冲器连续计数。例如，入门实验板的 DDR SDRAM 提供了一个足够的显示缓冲区。

在 HS 脉冲的开始和 VS 脉冲的开始没有具体规定相对的时序关系。因此，计数器被分配到简单格式的视频 RAM 地址，或分配到同步脉冲产生器的最小译码逻辑。

## UCF Location Constraints

图 6.4 提供了 VGA 显示端口的 UCF 约束，包括 I/O 口管脚分配和 I/O 口使用标准，输出摆率和输出驱动电流。

```
NET "VGA_RED" LOC = 'H14' | IOSTANDARD = LVTTL | DRIVE = 8 | SLEW = FAST ;
NET "VGA_GREEN" LOC = 'H15' | IOSTANDARD = LVTTL | DRIVE = 8 | SLEW = FAST ;
NET "VGA_BLUE" LOC = 'G15' | IOSTANDARD = LVTTL | DRIVE = 8 | SLEW = FAST ;
NET "VGA_HSYNC" LOC = 'F15' | IOSTANDARD = LVTTL | DRIVE = 8 | SLEW = FAST ;
NET "VGA_VSYNC" LOC = "F14" | IOSTANDARD = LVTTL | DRIVE = 8 | SLEW = FAST ;
```

*Figure 6-4: UCF Constraints for VGA Display Port*

# Chapter 7: RS-232 Serial Ports

## Overview

图 7.1 所示，入门实验板有 2 个 RS-232 串口：母口 DB9 DCE 和公口 DTE 连接器。DCE 口通过标准的串口电缆直接连到 PC 和工作站上。没有经过调制解调器或交换电缆不能用。

DTE 连接器用来控制其他的 RS-232 外围设备，如调制解调器、打印机或通过 DCE 连接器完成简单的环回测试。

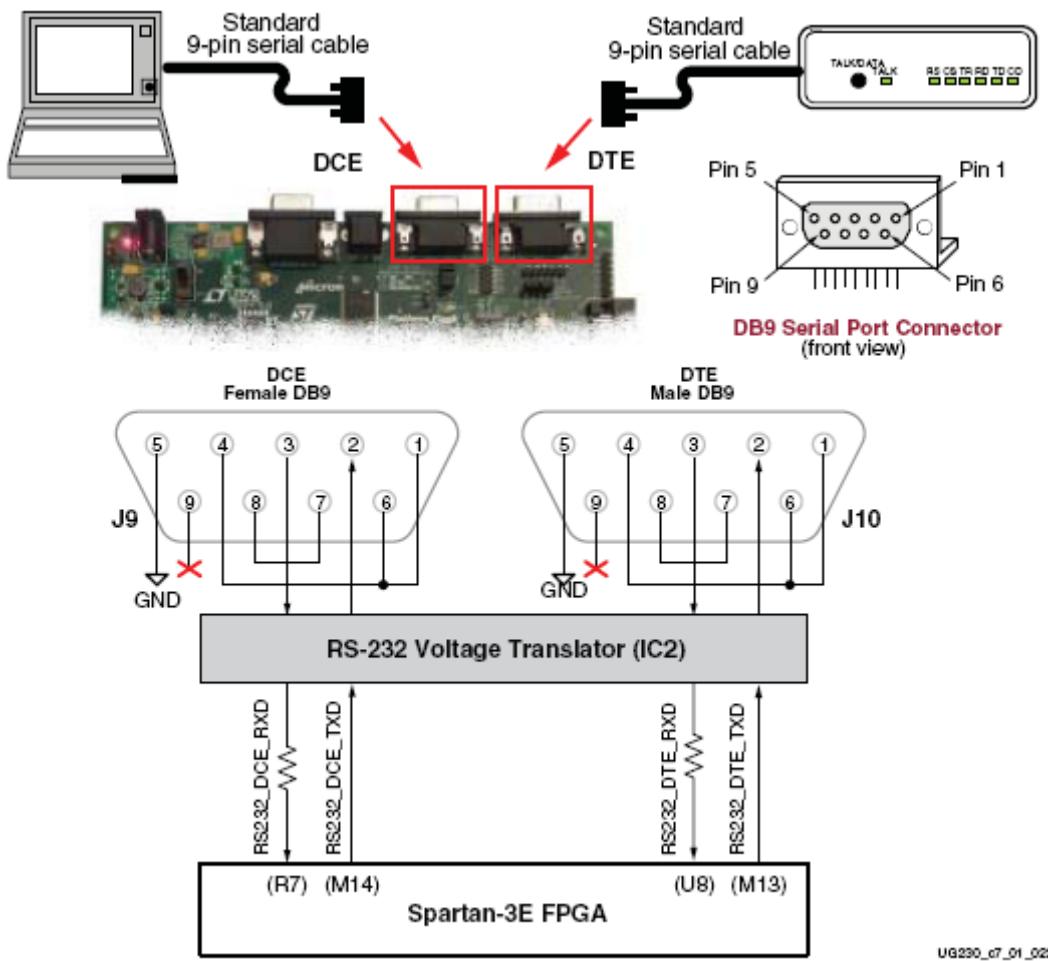


Figure 7-1: RS-232 Serial Ports

图 7.1 给出了 FPGA 和 2 个 DB9 连接器之间的连接。FPGA 使用 LVTTL 或 LVCMOS 电平提供串行数据给 Maxim 器件——用来电平转换的，以满足 RS-232 电压的电平。反之，Maxim 器件转换相应的 LVTTL 电平以满足 RS-232 串行输入数据给 FPGA。在 Maxim 与 FPGA 的 RXD 管脚之间串联一个电阻，以保护外部逻辑干扰。

连接器不支持硬件流控制。DCD、DTR 和 DSR 信号连接一起，如图 7.1 所示。同样，端口的 RTS 和 CTS 信号连接在一起。

## UCF Location Constraints

图 7.2 和 7.3 分别为 RS-232 的 DTE 和 DCE 端口提供了 UCF 约束，包括 I/O 口管脚分配和 I/O 口使用标准。

```

NET "RS232_DTE_RXD" LOC = "U8" | IOSTANDARD = LVTTL ;
NET "RS232_DTE_TXD" LOC = "M13" | IOSTANDARD = LVTTL | DRIVE = 8 | SLEW = SLOW ;

```

Figure 7-2: UCF Location Constraints for DTE RS-232 Serial Port

```

NET "RS232_DCE_RXD" LOC = "R7" | IOSTANDARD = LVTTL ;
NET "RS232_DCE_TXD" LOC = "M14" | IOSTANDARD = LVTTL | DRIVE = 8 | SLEW = SLOW ;

```

Figure 7-3: UCF Location Constraints for DCE RS-232 Serial Port

## Chapter 8 PS/2 Mouse/Keyboard Port

入门实验板包括一个 PS/2 的鼠标/键盘接口和标准的 6 管脚微 DIN 连接器——J14。图 8.1 给出了 PS/2 连接器，表 8.1 给出了连接信号。只有连接器的管脚 1 和 5 连到 FPGA。

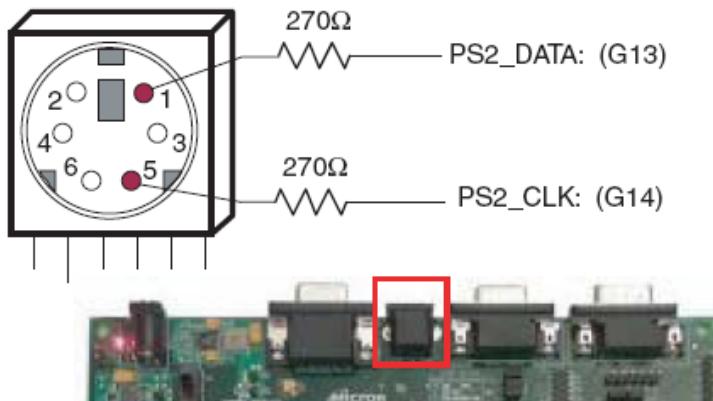


Figure 8-1: PS/2 Connector Location and Signals

Table 8-1: PS/2 Connector Pinout

PS/2 DIN Pin	Signal	FPGA Pin
1	DATA (PS2_DATA)	G13
2	Reserved	G13
3	GND	GND
4	+5V	—
5	CLK (PS2_CLK)	G14
6	Reserved	G13

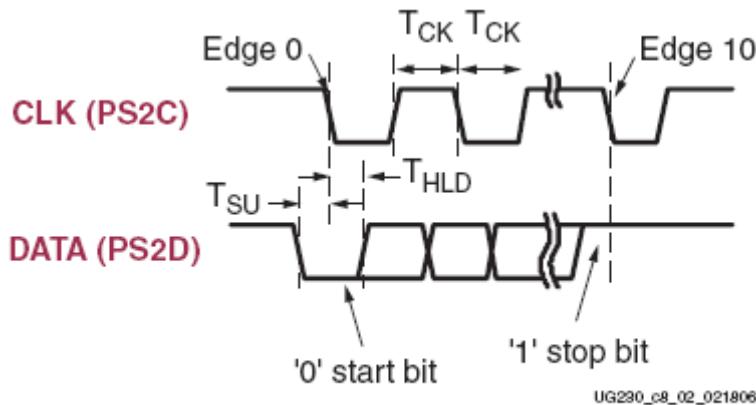
PC 鼠标和键盘都采用 2一线 PS/2 串行总线与主机进行通信，本入门实验板就属于此种情况。PS/2 总线包括时钟和数据线。鼠标和键盘通过识别信号的时序来驱动总线，它们均采用 11 位的字包括一个起始位、停止位和奇检验位。但是，鼠标和键盘的数据包打包方式不同。此外，

键盘允许双向数据交换，所以主机可以点亮键盘的状态灯。

表 8.2 和图 8.2 说明了 PS/2 总线的时序。只有当数据转换发生时，时钟和数据信号才被驱动，否则，它们以高电平的形式处于空闲状态。时序定义了鼠标与主机通信的信号要求或键盘与主机的双向通信。图 8.2 所示，当时钟信号是高电平时，敲击键盘或鼠标时往数据线写 1 位字，在时钟低电平时，主机 PC 读该数据线。

**Table 8-2: PS/2 Bus Timing**

Symbol	Parameter	Min	Max
T <sub>CCK</sub>	Clock High or Low Time	30 $\mu$ s	50 $\mu$ s
T <sub>SU</sub>	Data-to-clock Setup Time	5 $\mu$ s	25 $\mu$ s
T <sub>HLD</sub>	Clock-to-data Hold Time	5 $\mu$ s	25 $\mu$ s



**Figure 8-2: PS/2 Bus Timing Waveforms**

## Keyboard

键盘使用开集式驱动方式，这样键盘或主机均可以驱动 2一线总线。如果主机无需传送数据给键盘，则可以单线输入方式。

PS/2 式键盘采用扫描式编码来获取按键的数值。如今几乎所有的键盘都采用 PS/2 式。每个按键被按下时，都会产生一个独立的扫描码信号。按键的扫描码见图 8.3。

如果按键按下并且按住不放，键盘则每隔 100ms 重复发送扫描码信号。当按键释放时，键盘在扫描码之后发送一个“F0”。不管有没有按下 SHIF 键，键盘都发送同样的字符扫描码。

一些扩展键，按下时在扫描码之前发送一个“E0”，而且它们可能会发送多于一个的扫描码。当一个扩展键释放时，在扫描码之后发送一个“E0 F0”。

主机可以发送命令和数据给键盘。表 8.3 提供了一些常用的命令清单。

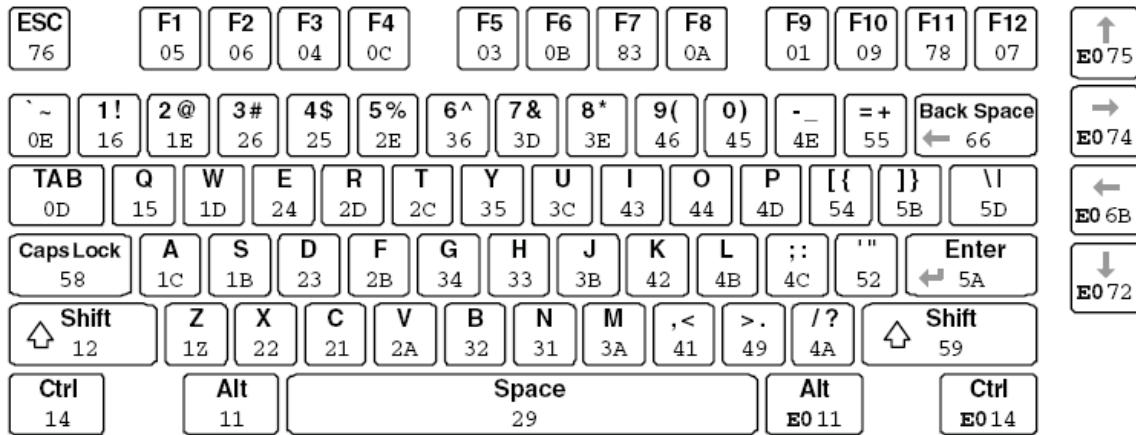


Figure 8-3: PS/2 Keyboard Scan Codes

Table 8-3: Common PS/2 Keyboard Commands

Command	Description																							
ED	Turn on/off Num Lock, Caps Lock, and Scroll Lock LEDs. The keyboard acknowledges receipt of an "ED" command by replying with an "FA", after which the host sends another byte to set LED status. The bit positions for the keyboard LEDs are shown below. Write a '1' to the specific bit to illuminate the associated keyboard LED.																							
	<table border="1"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr> <tr> <td colspan="5">Ignored</td><td>Caps Lock</td><td>Num Lock</td><td>Scroll Lock</td></tr> </table>								7	6	5	4	3	2	1	0	Ignored					Caps Lock	Num Lock	Scroll Lock
7	6	5	4	3	2	1	0																	
Ignored					Caps Lock	Num Lock	Scroll Lock																	
EE	Echo. Upon receiving an echo command, the keyboard replies with the same scan code "EE".																							
F3	Set scan code repeat rate. The keyboard acknowledges receipt of an "F3" by returning an "FA", after which the host sends a second byte to set the repeat rate.																							
FE	Resend. Upon receiving a resend command, the keyboard resends the last scan code sent.																							
FF	Reset. Resets the keyboard.																							

只有在空闲状态，数据线和时钟线均为高时，键盘才发送数据或命令给主机。

由于总线控制器是主方，在驱动总线之前，由键盘检查主方是否正在发送数据。时钟线可以用作清除发送信号线。如果主方将时钟线置低，直到时钟线被释放，键盘才能发送数据。

键盘以 11 位字（包括 0 起始位，接着是 8 位的扫描码（最低位 LSB 先传），再接着是奇校验位和终止位 1）发送数据给主机。当键盘发送数据时，它大约以 20~30KHz 的频率产生 11 个时钟周期的传送时间。在时钟的下降沿数据有效，如图 8.2 所示。

## Mouse

鼠标移动时产生一个时钟和数据信号，否则，这些信号保持高电平，说明处于空闲状态。鼠标每次移动，都发送 11 位字给主机。每个 11 位的字包括起始位 0，8 位的扫描码（最低位 LSB 先传），奇校验位和终止位 1。每个传送的数据总共包括 33 位，即第 0、11、22 位是起始位 0，第 10、21、32 位是终止位 1。三个数据域所包含的数据如图 8.4 所示。在时钟的下降沿数据有效，其时钟周期大约为 20~30KHz。

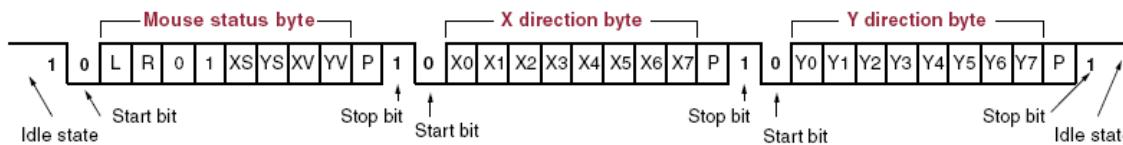
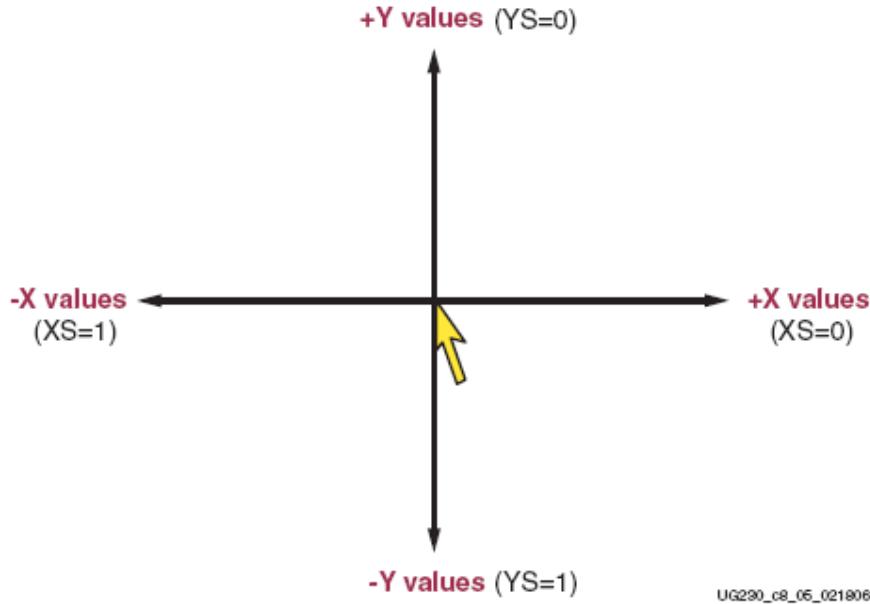


Figure 8-4: PS/2 Mouse Transaction

PS/2 式鼠标采用关联坐标体系，见图 8.5。鼠标右移时，X 域产生一个正值，左移时，产生负值。同理可得，鼠标上移时，Y 域产生一个正值，下移时，产生负值。XS 位和YS 位为每个值的状态位，1 表示负值。



**Figure 8-5: The Mouse Uses a Relative Coordinate System to Track Movement**

X 和 Y 值的大小代表鼠标移动速度的快慢。值越大，移动越快。状态字节的 XV 和 YV 位表示鼠标超出最大值，溢出。1 表示溢出。此时如果鼠标继续移动，则第 33 位每隔 50ms 重复发送一次。

状态字节的 L 和 R 域代表左右键按下。1 表示相应键按下。

## Voltage Supply

键盘的供给电压是 5V，而入门实验板的器件并不能承受 5V 电压，所以通信时串了个限流电阻，如图 8.1 所示。

## UCF Location Constraints

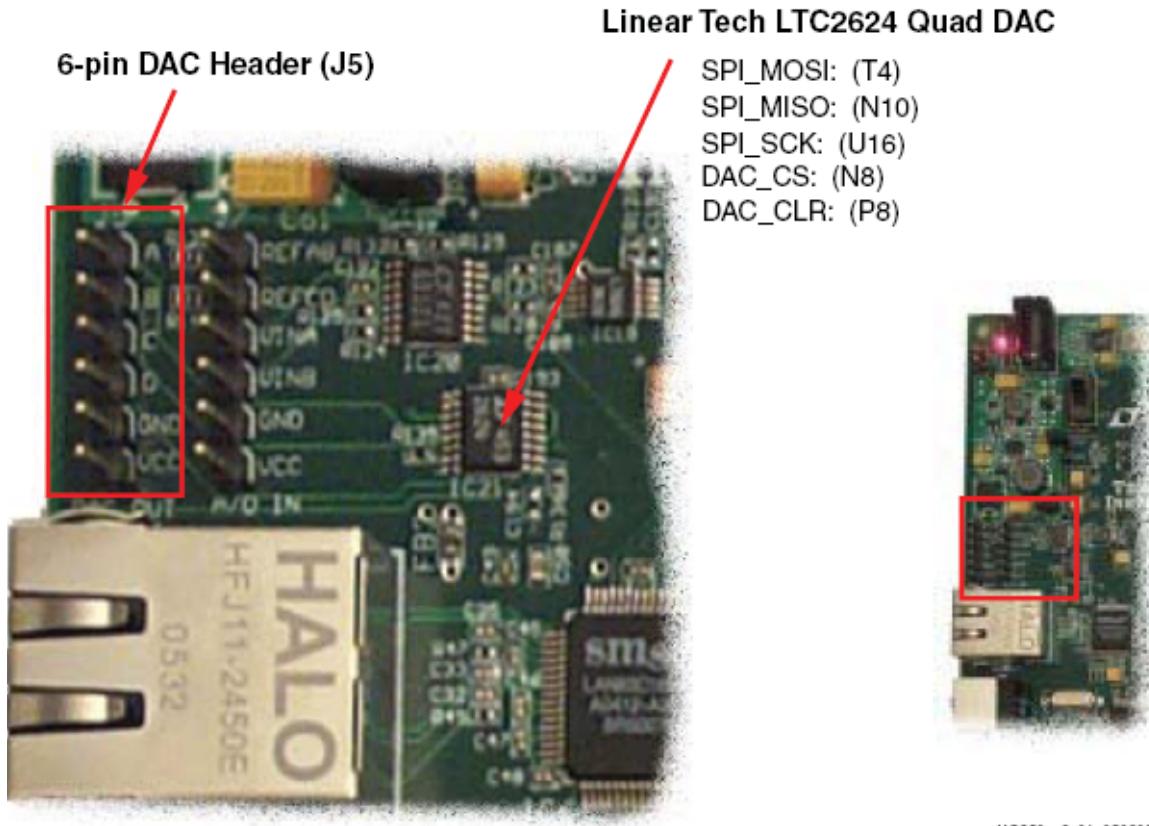
```
NET "PS2_CLK" LOC = "G14" | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = SLOW ;
NET "PS2_DATA" LOC = "G13" | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = SLOW ;
```

**Figure 8-6: UCF Location Constraints for PS/2 Port**

# Chapter 9

## Digital to Analog Converter (DAC)

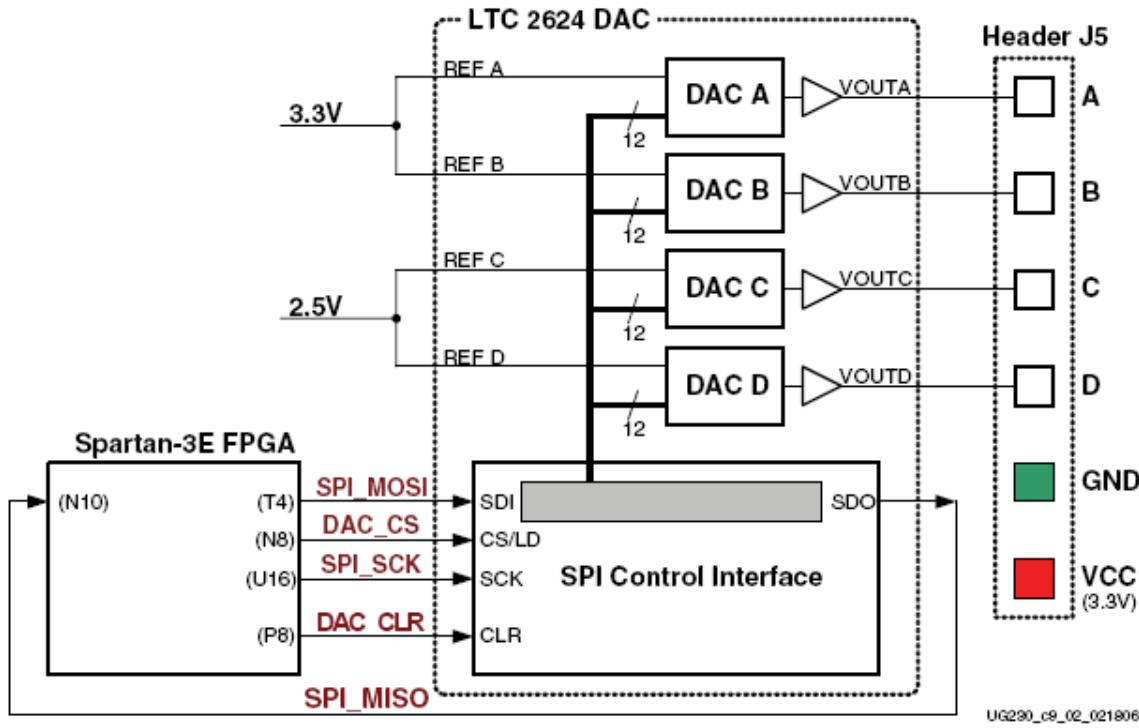
入门实验板包括一个兼容 SPI，4 通道，串行数字—模拟转换器（DAC）。该 DAC 器件是个线性工艺 LTC2624-12 位的。4 个输出连到 J5 接头——使用 Digilent 6 管脚的外设模数格式。



*Figure 9-1: Digital-to-Analog Converter and Associated Header*

## SPI Communication

图 9.2 所示, FPGA 通过 SPI 与 DAC 的输出通道进行数据通信。SPI 总线是全双工、同步、定向通道——采用简单的 4 线接口。在本例中, FPGA 作为主总线使用, 驱动数据时钟信号 (SPI\_SCK) 和传送串行数据 (SPI\_MOSI) 给从数据线——本例为 DAC。与此同时, 从数据线提供串行数据 (SPI\_MISO) 回传给主数据线。



UG290\_c9\_02\_021806

**Figure 9-2: Digital-to-Analog Connection Schematics Interface Signals**

表 9.1 列出了 FPGA 与 DAC 之间的接口信号。SPI\_MOSI、SPI\_MISO 和 SPI\_SCK 信号与其它的器件复用 SPI 总线。DAC\_CS 信号低电平有效送给 DAC。DAC\_CLR 信号低电平有效，同步复位送给 DAC。

**Table 9-1: DAC Interface Signals**

Signal	FPGA Pin	Direction	Description
SPI_MOSI	T4	FPGA → DAC	Serial data: Master Output, Slave Input
DAC_CS	N8	FPGA → DAC	Active-Low chip-select. Digital-to-analog conversion starts when signal returns High.
SPI_SCK	U16	FPGA → DAC	Clock
DAC_CLR	P8	FPGA → DAC	Asynchronous, active-Low reset input
SPI_MISO	N10	FPGA ← DAC	Serial data: Master Input, Slave Output

从 DAC 输出的串行数据主要送给多个 DAC。在多数应用中，该信号可以被视，尽管它可以使用 SPI 总线的全双工方式通信。

### Disable Other Devices on the SPI Bus to Avoid Contention

板上的 SPI 总线与其它器件复用。当 FPGA 与 DAC 进行数据通信时，要失能其它的器件，这点很重要，以避免总线冲突。表 9.2 提供了失能其它器件的信号与逻辑值。尽管 StrataFlash PROM 是个并行器件，它的最低数据位还是和 SPI\_MISO 信号线复用。

Table 9-2: Disabled Devices on the SPI Bus

Signal	Disabled Device	Disable Value
SPI_SS_B	SPI serial Flash	1
AMP_CS	Programmable pre-amplifier	1
AD_CONV	Analog-to-Digital Converter (ADC)	0
SF_CE0	StrataFlash Parallel Flash PROM	1
FPGA_INIT_B	Platform Flash PROM	1

### SPI Communication Details

表 9.3 说明了 SPI 总线时序的具体应用。每位的传输与接收均相对于 SPI\_SCK 时钟信号。总线完全是静态的并支持最高的 50MHz。但是，在最高速或接近最高速时，请使用 LTC2624 数据手册检查时序。

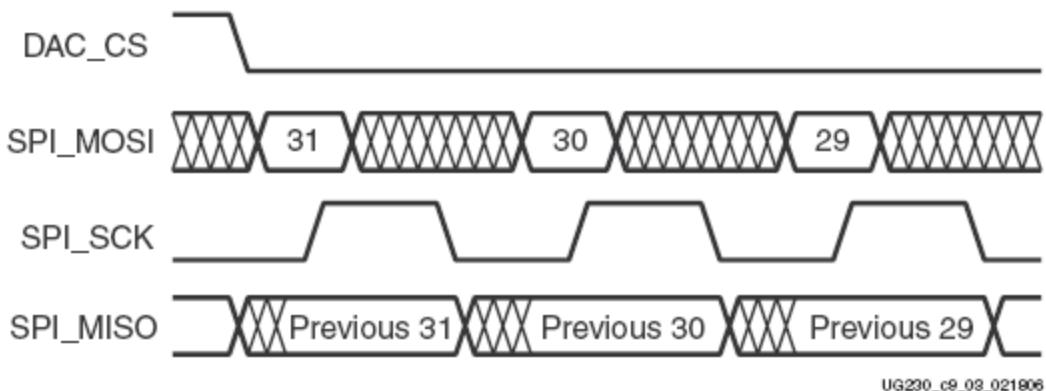


Figure 9-3: SPI Communication Waveforms

在从片选信号 DAC\_CS 置低后，FPGA 用 SPI\_MOSI 信号传送数据，MSB 位先传。LTC2624 在 SPI\_SCK 的上升沿采集输入数据 (SPI\_MOSI)。相对于时钟信号的上升沿，数据的有效时间至少是 4ns。

LTC2624 在 SPI\_SCK 的下降沿采集输入数据 (SPI\_MISO)。FPGA 在下个 SPI\_SCK 的上升沿采集该数据。FPGA 必须在 DAC\_CS 变低后的 SPI\_SCK 的第一个上升沿读取第一个 SPI\_MISO 的值。否则，31 位会丢失。

传完所有的 32 位数据后，FPGA 完成 SPI 总线处理，置从片选信号 DAC\_CS 为高。在高电平即去时，DAC 内部开始进行数—模转换。

### Communication Protocol

图 9.4 说明了 LTC2624 DAC 接口所需的通信协议。DAC 支持 24 位和 32 位协议。32 位的协议如下所示。在 D/A 转换器的内部，SPI 接口由 32 位的移位寄存器组成。每个 32 位的指令字随在数据值之后组成一条指令，一个地址。随着新指令进入 DAC，前 32 位指令字回送给主器件。从 DAC 回应的信息可以被忽视，尽管它对确认正确的通信有用。

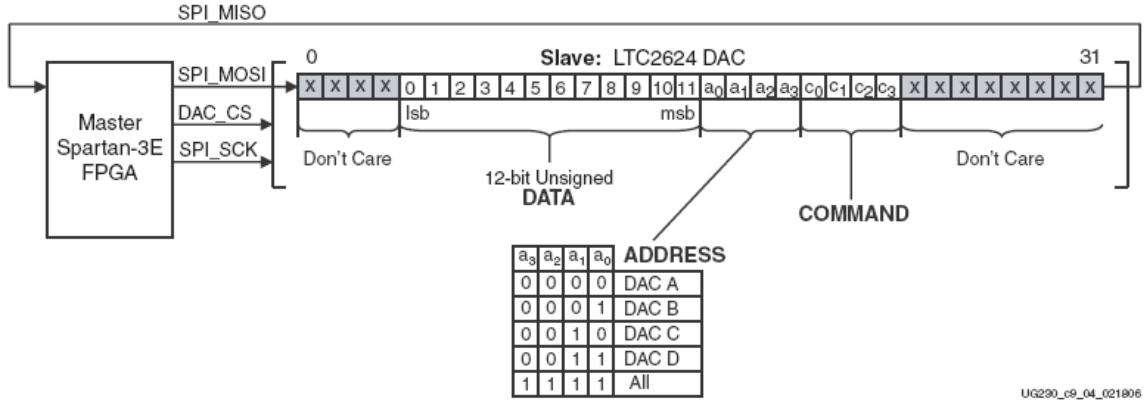


Figure 9-4: SPI Communications Protocol to LTC2624 DAC

FPGA 首先传送 8 个无效位或无关位，随后是 4 位的命令。最普通的使用命令是  $\text{COMMAND}[3:0]=0011$ ，这指的是立即更新 DAC 的输出数据值。随后，FPGA 通过 4 位的地址域选择一个或所有的输出通道。地址域之后，FPGA 发送 12 位无符号数据——DAC 转换成模拟值并输出到相应的输出端。最后，4 个额外的无效位或无关位紧随在 32 位命令字之后。

## Specifying the DAC Output Voltage

如图 9.2 所示，每个 DAC 输出电平与 12 位的无符号数字值  $D[11: 0]$  相一致，FPGA 经由 SPI 接口写给 DAC。

电压的输出见等式 9.1。参考电压  $V_{\text{REFERENCE}}$  在 4 个输出通道之间有所差异。通道 A 和通道 B 的参考电压是 3.3V，通道 C 和通道 D 是 2.5V。参考电压本身有  $\pm 5\%$  的波动，所以输出电压会有相应的轻微波动。

$$V_{\text{OUT}} = \frac{D[11:0]}{4096} \times V_{\text{REFERENCE}} \quad \text{Equation 9-1}$$

### DAC Outputs A and B

等式 9.2 提供了 DAC 输出通道 A 和 B 的输出电压。参考电压为  $3.5V \pm 5\%$ 。

$$V_{\text{OUTA}} = \frac{D[11:0]}{4096} \times (3.3V \pm 5\%) \quad \text{Equation 9-2}$$

### DAC Outputs C and D

等式 9.3 提供了 DAC 输出通道 C 和 D 的输出电压。参考电压为  $2.5V \pm 5\%$ 。

$$V_{\text{OUTC}} = \frac{D[11:0]}{4096} \times (2.5V \pm 5\%) \quad \text{Equation 9-3}$$

## UCF Location Constraints

```

NET "SPI_MISO" LOC = "N10" | IOSTANDARD = LVCMOS33 ;
NET "SPI_MOSI" LOC = "T4" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8 ;
NET "SPI_SCK" LOC = "U16" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8 ;
NET "DAC_CS" LOC = "N8" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8 ;
NET "DAC_CLR" LOC = "P8" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8 ;

```

Figure 9-5: UCF Location Constraints for the DAC Interface

# Chapter 10: Analog Capture Circuit

入门实验板包括一个双通道的模拟信号采集电路，由可编程缩放预放大器和模一数转换器(DAC)组成，如图 10.1 所示。

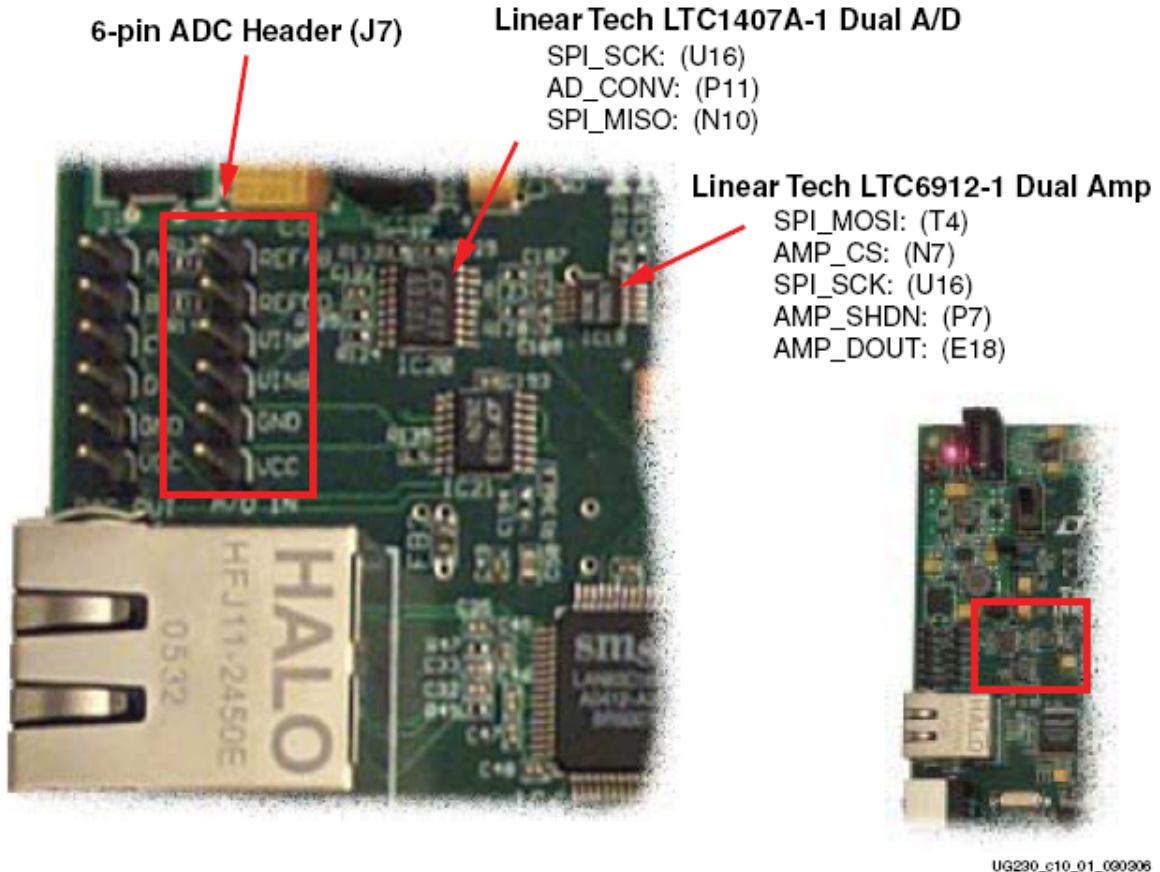


Figure 10-1: Two-Channel Analog Capture Circuit

模拟信号采集电路由线性工艺 LTC6912-1 可编程预放大器构成，其可以缩放从 J7 输入的模拟信号，如图 10.2 所示。预放大器的输出连接到线性工艺 LTC1407A-1 ADC。预放大器和 ADC 由 FPGA 连续编程或控制。

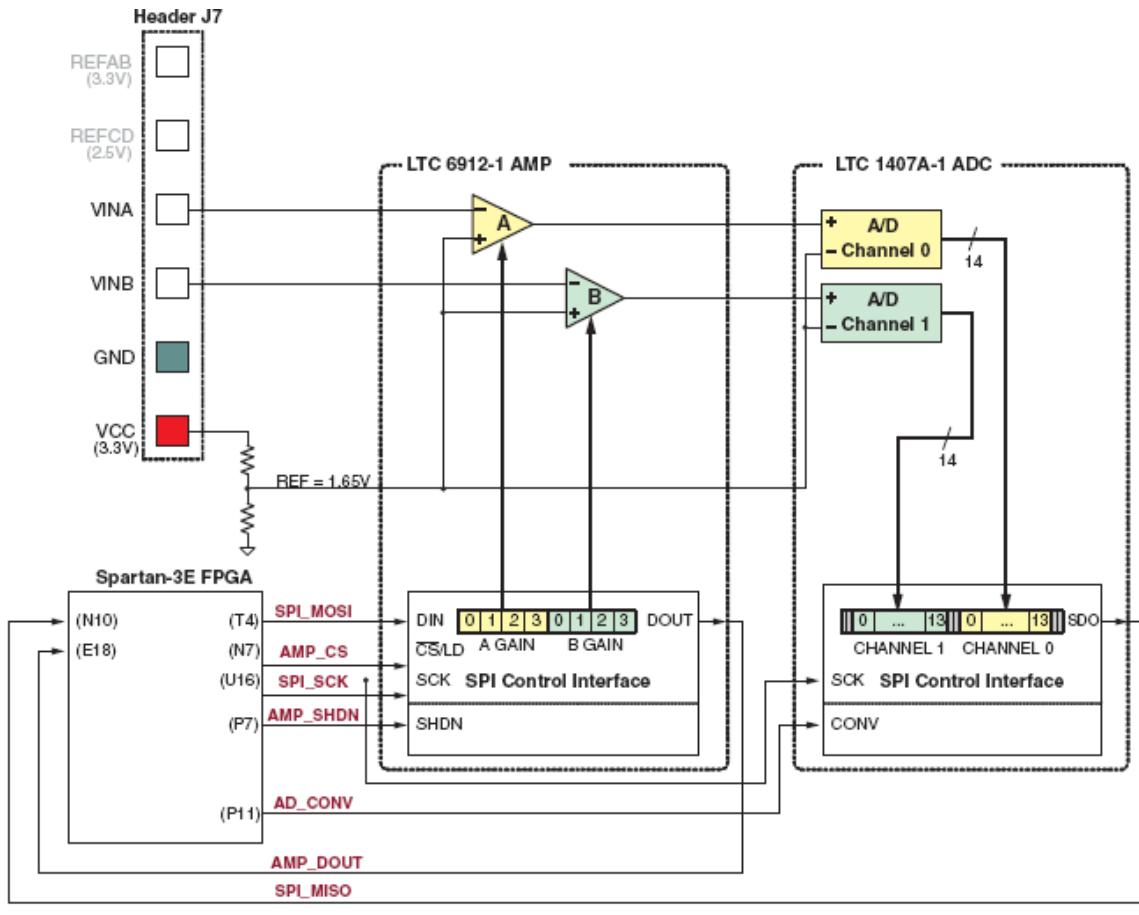


Figure 10-2: Detailed View of Analog Capture Circuit

## Digital Outputs from Analog Inputs

采集电路通过 VINA 和 VINB 端口转换模拟电压，并用相应的 14 位数字表示，D[13:0]，见等式 10.1。

$$D[13:0] = GAIN \times \frac{(V_{IN} - 1.65V)}{1.25V} \times 8192 \quad \text{Equation 10-1}$$

增益 (GAIN) 是当前设置的，并送给可编程预放大器。增益可以设置成多种值，VINA 和 VINB 允许多种输入值，具体见表 10.2 所示。

放大器和 ADC 的参考电压是 1.65V，一般由图 10.2 的分压电路产生。因而，1.65V 是输入电压减去 VINA 或 VINB 得到的。

ADC 的最大波动范围是  $\pm 1.25V$ ，中点在参考电压 1.65V 附近。因此，1.25V 出现在分母上以缩放模拟输入信号。

最后，ADC 输出一个 14 位 - 2 个辅助数字输出值。14 位 - 2 个辅助数字代表的值  $-2^{13} \sim 2^{13} - 1$  之间。因此，其分点的数量为 8192。

## Programmable Pre-Amplifier

LTC6912-1 提供 2 个独立的可编程增益转换放大器。放大器的目的是缩放 VINA 或 VINB 的输入电压。DAC 的最大转换范围是  $1.65 \pm 1.25V$ 。

## Interface

表 10.1 列出了 FPGA 和 SPI\_MOSI、SPI\_MISO 和 SPI\_SCK（这三个信号与其它器件复用 SPI 总线）之间的接口信号。AMP\_CS 是送给放大器的低电平有效的片选信号。

Table 10-1: AMP Interface Signals

Signal	FPGA Pin	Direction	Description
SPI_MOSI	T4	FPGA → AD	Serial data: Master Output, Slave Input. Presents 8-bit programmable gain settings, as defined in Table 10-2.
AMP_CS	N7	FPGA → AMP	Active-Low chip-select. The amplifier gain is set when signal returns High.
SPI_SCK	U16	FPGA → AMP	Clock
AMP_SHDN	P7	FPGA → AMP	Active-High shutdown, reset
AMP_DOUT	E18	FPGA ← AMP	Serial data. Echoes previous amplifier gain settings. Can be ignored in most applications.

## Programmable Gain

每个模拟通道有一个对应的可编程增益放大器，见图 10.2。模拟信号出现在连接 J7 的 VINA 或 VINB 输入端口，其放大是相对于 1.65V 的。

每个放大器的增益的编程范围是  $-1 \sim -100$ ，见表 10.2。

Table 10-2: Programmable Gain Settings for Pre-Amplifier

Gain	A3	A2	A1	A0	Input Voltage Range	
	B3	B2	B1	B0	Minimum	Maximum
0	0	0	0	0		
-1	0	0	0	1	0.4	2.9
-2	0	0	1	0	1.025	2.275
-5	0	0	1	1	1.4	1.9
-10	0	1	0	0	1.525	1.775
-20	0	1	0	1	1.5875	1.7125
-50	0	1	1	0	1.625	1.675
-100	0	1	1	1	1.6375	1.6625

## SPI Control Interface

图 10.3 给出了基于 SPI 的与放大器的通信接口。每个放大器的增益设置成 8 位的命令字，由 2 个 4 位的部分组成。最高位 B3 先送。

从放大器输出的 AMP\_DOUT 响应原先的增益设置。在大多数应用中，该值可以被忽视。

当 FPGA 声明 AMP\_CS 为低时，SPI 总线开始进行数据传输，见图 10.4。放大器在 SPI\_SCK 时钟信号的上升沿采集 SPI\_MOSI 的串行数据。放大器在 SPI\_SCK 的下降沿通过 AMP\_DOUT 输出串行数据。

放大器接口的速率相对较低，支持 10MHz 的时钟频率。

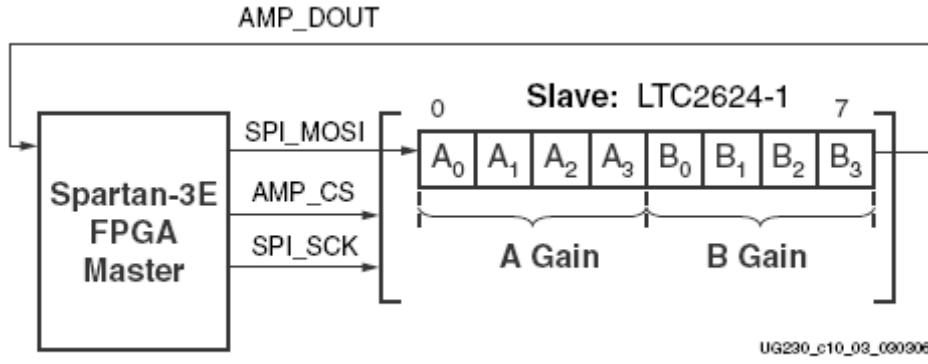


Figure 10-3: SPI Serial Interface to Amplifier

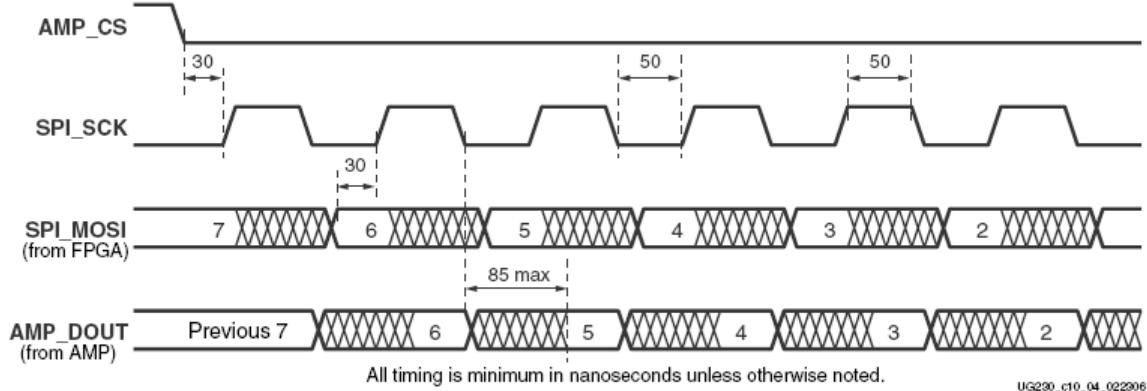


Figure 10-4: SPI Timing When Communicating with Amplifier

## UCF Location Constraints

```

NET "SPI_MOSI" LOC = "T4" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 6 ;
NET "AMP_CS" LOC = "N7" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 6 ;
NET "SPI_SCK" LOC = "U16" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8 ;
NET "AMP_SHDN" LOC = "P7" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 6 ;
NET "AMP_DOUT" LOC = "E18" | IOSTANDARD = LVCMOS33 ;

```

Figure 10-5: UCF Location Constraints for the DAC Interface

## Analog to Digital Converter (ADC)

LTC1407A-1 提供两个 ADC。当 AD\_CONV 信号有效时，这两个模拟输入端口同时进行采样。

### Interface

表 10.3 列出了 FPGA 和 ADC 的接口信号。SPI\_MOSI、SPI\_MISO 和 SPI\_SCK 这三个信号与其它器件复用 SPI 总线。送给 DAC 的从片选信号 DAC\_CS 低电平有效。DAC\_CLR 异步复位信号低电平有效。

Table 10-3: ADC Interface Signals

Signal	FPGA Pin	Direction	Description
SPI_SCK	U16	FPGA → ADC	Clock
AD_CONV	P11	FPGA → ADC	Active-High shutdown and reset.
SPI_MISO	N10	FPGA ← ADC	Serial data: Master Input, Serial Output. Presents the digital representation of the sample analog values as two 14-bit two's complement binary values.

## SPI Control Interface

图 10.6 提供了 SPI 总线传输给 ADC 的一个例子。

当 AD\_CONV 信号变高时，ADC 的两个通道同时采样。直到下个 AD\_CONV 信号有效时采样结果才被送出，即保留了一个采样周期。最大的采样频率大约是 1.5MHz。

ADC 将采集到的模拟信号以 14 位—2 个辅助二进制值的形式输出。

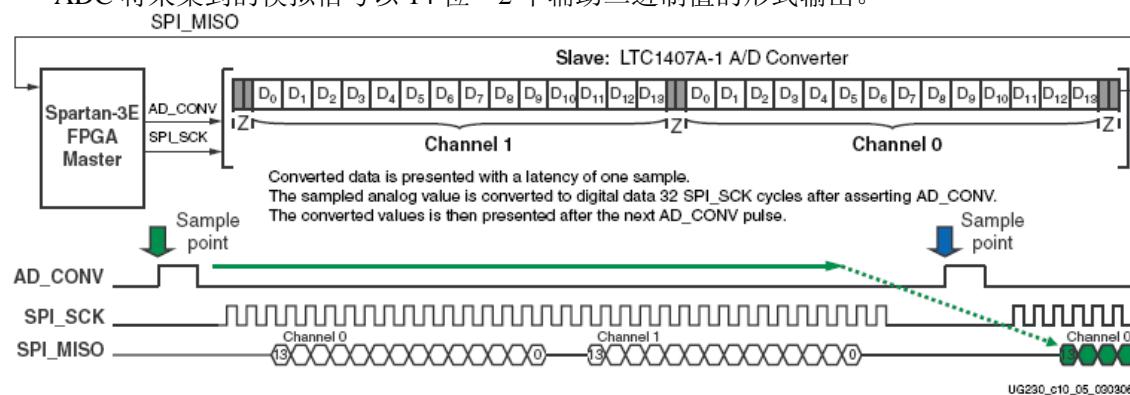


Figure 10-6: Analog-to-Digital Conversion Interface

图 10.7 给出了具体的传输时序。AD\_CONV 信号不再是原来意义上的 SPI 从片选使能信号了。ADC 让 SPI\_MISO 信号处于高阻态以保证获得足够的 SPI\_SCK 时钟周期。否则，ADC 阻止与其它 SPI 外设进行通信。图 10.6 所示，一次通信使用了 34 个时钟周期。ADC 在传输每个 14 位数据的前后使其输出保留 2 个时钟周期的三态状态。

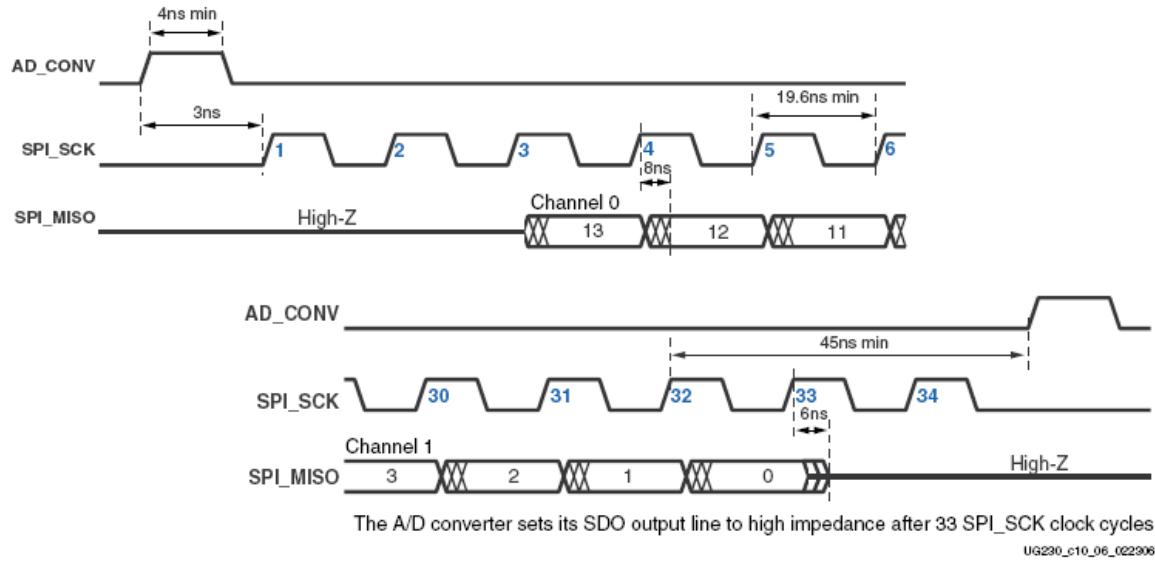


Figure 10-7: Detailed SPI Timing to ADC

## UCF Location Constraints

```
NET "AD_CONV" LOC = "P11" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 6 ;
NET "SPI_SCK" LOC = "U16" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8 ;
NET "SPI_MISO" LOC = "N10" | IOSTANDARD = LVCMOS33 ;
```

## Disable Other Devices on the SPI Bus to Avoid Contention

SPI总线与板上的其它器件复用。当FPGA与AMP或ADC进行通信时失能其它的器件是很重要的，这样可以避免总线冲突。表10.4提供了失能其它器件的信号及其相应的逻辑值。尽管StrataFlash PROM是个并行器件，它的最低位还是与SPI\_MISO信号线复用。如果FPGA设置为主串行配置模式时，Platform Flash PROM才有可能使能。

Table 10-4: Disable Other Devices on SPI Bus

Signal	Disabled Device	Disable Value
SPI_SS_B	SPI Serial Flash	1
AMP_CS	Programmable Pre-Amplifier	1
DAC_CS	DAC	1
SF_CE0	StrataFlash Parallel Flash PROM	1
FPGA_INIT_B	Platform Flash PROM	1

## Connecting Analog Inputs

通过一个DC(直流)隔离电容连接AC信号与VINA或VINB。

# Chapter 11: Intel StrataFlash Parallel NOR Flash PROM

如图11.1所示，实验板包括一个128Mbit（16Mbyte）的Intel StrataFlash parallel NOR Flash PROM。如图所示，StrataFlash的一些连线与板上的其它一些元件复用。

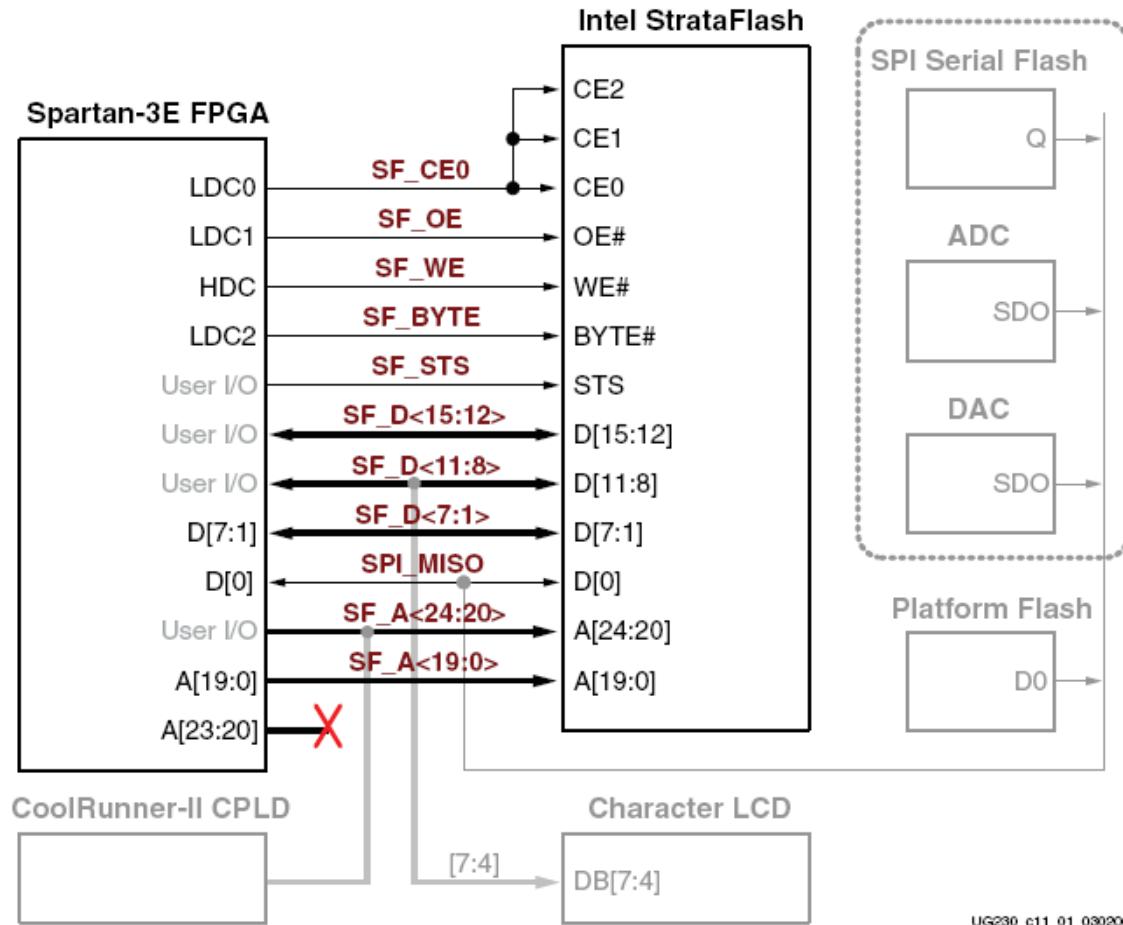


Figure 11-1: Connections to Intel StrataFlash Flash Memory

StrataFlash PROM有以下多种功能：

- 1) 存储单个FPGA的配置文件；2) 存储两个不同的FPGA配置文件，并利用Spartan-3E FPGA’s MultiBoot的特点进行轮流转换；3) 存储MicroBlaze处理器的代码；4) 存储MicroBlaze处理器的代码并在执行代码之前将代码映射到DDR存储器；4) 存储来自FPGA的永久数据。

## StrataFlash Connections

表11.1说明了FPGA与StrataFlash之间的连接图。

虽然XC3S500E FPGA每个配置信息仅需要稍微多于2Mbit的容量，但FPGA-to-StrataFlash之间的接口支持高达256Mbit的StrataFlash。实验板用的是128Mbit元件。地址线SF\_A24没用到。一般来说，StrataFlash元件连到XC3S500E支持8位的外围接口（BPI）配置。从FPGA出来的高4位并没有直接连到StrataFlash。相反，在配置时由XC2C64 CPLD控制这些管脚。正如表11.1所示，StrataFlash的一些连线与其它一些器件复用。

Table 11-1: FPGA-to-StrataFlash Connections

Category	StrataFlash Signal Name	FPGA Pin Number	Function
Address	SF_A24	A11	Shared with XC2C64A CPLD. The CPLD actively drives these pins during FPGA configuration, as described in <a href="#">Chapter 16, "XC2C64A CoolRunner-II CPLD"</a> . Also connects to FPGA user-I/O pins. SF_A24 is the same as FX2 connector signal FX2_IO<32>.
	SF_A23	N11	
	SF_A22	V12	
	SF_A21	V13	
	SF_A20	T12	
	SF_A19	V15	
	SF_A18	U15	
	SF_A17	T16	
	SF_A16	U18	
	SF_A15	T17	
	SF_A14	R18	
	SF_A13	T18	
	SF_A12	L16	
	SF_A11	L15	
	SF_A10	K13	
	SF_A9	K12	
	SF_A8	K15	
	SF_A7	K14	
	SF_A6	J17	
	SF_A5	J16	
	SF_A4	J15	
	SF_A3	J14	
	SF_A2	J12	
	SF_A1	J13	
	SF_A0	H17	

Table 11-1: FPGA-to-StrataFlash Connections

Category	StrataFlash Signal Name	FPGA Pin Number	Function
Data	SF_D15	T8	Upper 8 bits of a 16-bit halfword when StrataFlash is configured for x16 data (SF_BYTE=High). Connects to FPGA user I/O.
	SF_D14	R8	
	SF_D13	P6	
	SF_D12	M16	
	SF_D11	M15	
	SF_D10	P17	
	SF_D9	R16	
	SF_D8	R15	
	SF_D7	N9	
	SF_D6	M9	
	SF_D5	R9	
	SF_D4	U9	
	SF_D3	V9	
	SF_D2	R10	
	SF_D1	P10	
	SPI_MISO	N10	Bit 0 of data byte and 16-bit halfword. Connects to FPGA pin D0/DIN to support the BPI configuration. Shared with other SPI peripherals and Platform Flash PROM.
Control	SF_CE0	D16	StrataFlash Chip Enable. Connects to FPGA pin LDC0 to support the BPI configuration.
	SF_WE	D17	StrataFlash Write Enable. Connects to FPGA pin HDC to support the BPI configuration.
	SF_OE	C18	StrataFlash Chip Enable. Connects to FPGA pin LDC1 to support the BPI configuration.
	SF_BYTE	C17	StrataFlash Byte Enable. Connects to FPGA pin LDC2 to support the BPI configuration. 0: x8 data 1: x16 data
	SF_STS	B18	StrataFlash Status signal. Connects to FPGA user-I/O pin.

## Shared Connections

除了StrataFlash的一些连线与其它一些器件复用外，StrataFlash的一些存储空间也与其它器件复用。

## Character LCD

字符LCD采用4位的数据接口。显示数据的连线同样与StrataFlash PROM上的信号线

SF\_D<11:8>复用。如表11.2所示，FPGA通过SF\_CE0和LCD\_RW信号线来控制StrataFlash PROM的访问通道或字符LCD的读状态。

**Table 11-2: FPGA Control for StrataFlash and LCD**

SF_CE0	LCD_RW	Function
1	1	The FPGA reads from the character LCD.
0	0	The FPGA accesses the StrataFlash PROM.

## Xilinx XC2C64A CPLD

在配置时，Xilinx XC2C64A CoolRunner CPLD用来控制StrataFlash高5位的地址线SF\_A<24:20>。在BPI模式的高4位地址线A<23:20>没有连接。相反，FPGA的4个I/O口连接到StrataFlash PROM的高位地址线SF\_A<23:0>上。可以参阅16章了解CPLD更详细的信息。16 Mbyte StrataFlash PROM并没有用到最高有效位地址线SF\_A<24>。它是为同样封装形式更大容量的StrataFlash PROM提供扩展的。当然，SF\_A<24>信号线也连到FX2扩展接头的FX2\_IO<32>上。

## SPI Data Line

最低有效位数据线SF\_D<0>与来自串行SPI外设、SPI\_MISO和Platform Flash PROM的串行输出信号线复用，见表11.3。为了避免冲突，FPGA必须确保在任何时刻只能有一个数据线有效。

**Table 11-3: Possible Contention on SPI\_MISO (SF\_D<0>) Data**

Condition	Function
FPGA_M2 = Low FPGA_M1 = Low FPGA_M0 = Low INIT_B = High	Platform Flash outputs data on D0.
SF_CE0 = Low SF_OE = Low	StrataFlash outputs data.
AD_CONV = High SPI_SCK	Serial data is clocked out of the A/D converter
DAC_CS = Low SPI_SCK	DAC outputs previous command in response to SPI_SCK transitions.

## UCF Location Constraints

### Address

图11.2是StrataFlash地址线管脚的UCF约束。

```

NET 'SF_A<24>' LOC = 'A11' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_A<23>' LOC = 'N11' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_A<22>' LOC = 'V12' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_A<21>' LOC = 'V13' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_A<20>' LOC = 'T12' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_A<19>' LOC = 'V15' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_A<18>' LOC = 'U15' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_A<17>' LOC = 'T16' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_A<16>' LOC = 'U18' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_A<15>' LOC = 'T17' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_A<14>' LOC = 'R18' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_A<13>' LOC = 'T18' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_A<12>' LOC = 'L16' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_A<11>' LOC = 'L15' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_A<10>' LOC = 'K13' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_A<9>' LOC = 'K12' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_A<8>' LOC = 'K15' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_A<7>' LOC = 'K14' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_A<6>' LOC = 'J17' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_A<5>' LOC = 'J16' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_A<4>' LOC = 'J15' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_A<3>' LOC = 'J14' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_A<2>' LOC = 'J12' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_A<1>' LOC = 'J13' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_A<0>' LOC = 'H17' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;

```

**Figure 11-2: UCF Location Constraints for StrataFlash Address Inputs Data**

图11.3是数据线管脚的UCF约束。

```

NET 'SF_D<15>' LOC = 'T8' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_D<14>' LOC = 'R8' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_D<13>' LOC = 'P6' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_D<12>' LOC = 'M16' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_D<11>' LOC = 'M15' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_D<10>' LOC = 'P17' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_D<9>' LOC = 'R16' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_D<8>' LOC = 'R15' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_D<7>' LOC = 'N9' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_D<6>' LOC = 'M9' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_D<5>' LOC = 'R9' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_D<4>' LOC = 'U9' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_D<3>' LOC = 'V9' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_D<2>' LOC = 'R10' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET 'SF_D<1>' LOC = 'P10' | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET "SPI_MISO" LOC = 'N10' | IOSTANDARD = LVCMOS33 | DRIVE = 6 | SLEW = SLOW ;

```

**Figure 11-3: UCF Location Constraints for StrataFlash Data I/Os Control**

图11.4是控制线管脚的UCF约束。

```

NET "SF_BYTE" LOC = "C17" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET "SF_CBO" LOC = "D16" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET "SF_OE" LOC = "C18" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET "SF_STS" LOC = "B18" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET "SF_WE" LOC = "D17" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;

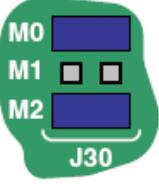
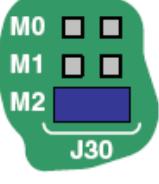
```

*Figure 11-4: UCF Location Constraints for StrataFlash Control Pins*

## Setting the FPGA Mode Select Pins

可以将FPGA设置为BPI UP 或BPI DOWN模式，见表11.4。

*Table 11-4: Selecting BPI-Up or BPI-Down Configuration Modes (Header J30 in Figure 4-2)*

Configuration Mode	Mode Pins M2:M1:M0	FPGA Configuration Image in StrataFlash	Jumper Settings
BPI Up	0:1:0	FPGA starts at address 0 and increments through address space. The CPLD controls address lines A[24:20] during BPI configuration.	
BPI Down	0:1:1	FPGA starts at address 0xFF_FFFF and decrements through address space. The CPLD controls address lines A[24:20] during BPI configuration.	

# Chapter 12: SPI Serial Flash

实验板包括一个STMicroelectronics M25P16 16 Mbit SPI serial Flash，其有多种应用。SPI Flash提供了可选的方案来配置FPGA——这是Spartan-3E FPGA 的一个新的特征，见图12.1所示。SPI Flash在配置后同样可以有其它的多种用途，如：1) 简单的永久性数据存储；2) 存储识别码、串号以及IP地址等等；3) 将存储的MicroBlaze处理器代码映射到DDR SDRAM上。

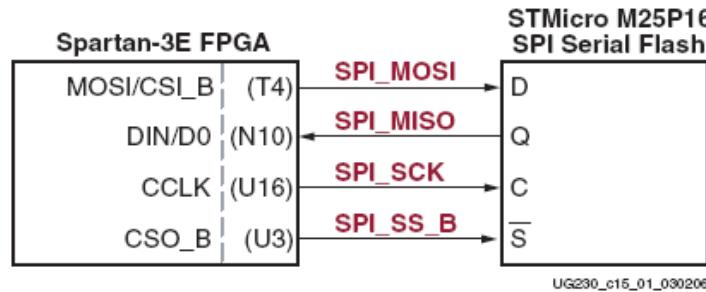


Figure 12-1: Spartan-3E FPGAs Have an Optional SPI Flash Configuration Interface  
Table 12-1: SPI Flash Interface Signals

Signal	FPGA Pin	Direction	Description
SPI_MOSI	T4	FPGA→SPI	Serial data: Master Output, Slave Input
SPI_MISO	N10	FPGA←SPI	Serial data: Master Input, Slave Output
SPI_SCK	U16	FPGA→SPI	Clock
SPI_SS_B	U3	FPGA→SPI	Asynchronous, active-Low slave select input

## UCF Location Constraints

图12.2是SPI Flash PROM的UCF约束。

```
# some connections shared with SPI Flash, DAC, ADC, and AMP
NET "SPI_MISO" LOC = "N10" | IOSTANDARD = LVCMOS33 ;
NET "SPI_MOSI" LOC = "T4" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 6 ;
NET "SPI_SCK" LOC = "U16" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 6 ;
NET "SPI_SS_B" LOC = "U3" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 6 ;
NET "SPI_ALT_CS_JP11" LOC = "R12" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 6 ;
```

Figure 12-2: UCF Location Constraints for SPI Flash Connections

## Configuring from SPI Flash

SPI Flash 的配置，FPGA的模式选择必须合理，SPI Flash必须包含一个有效的配置映射。

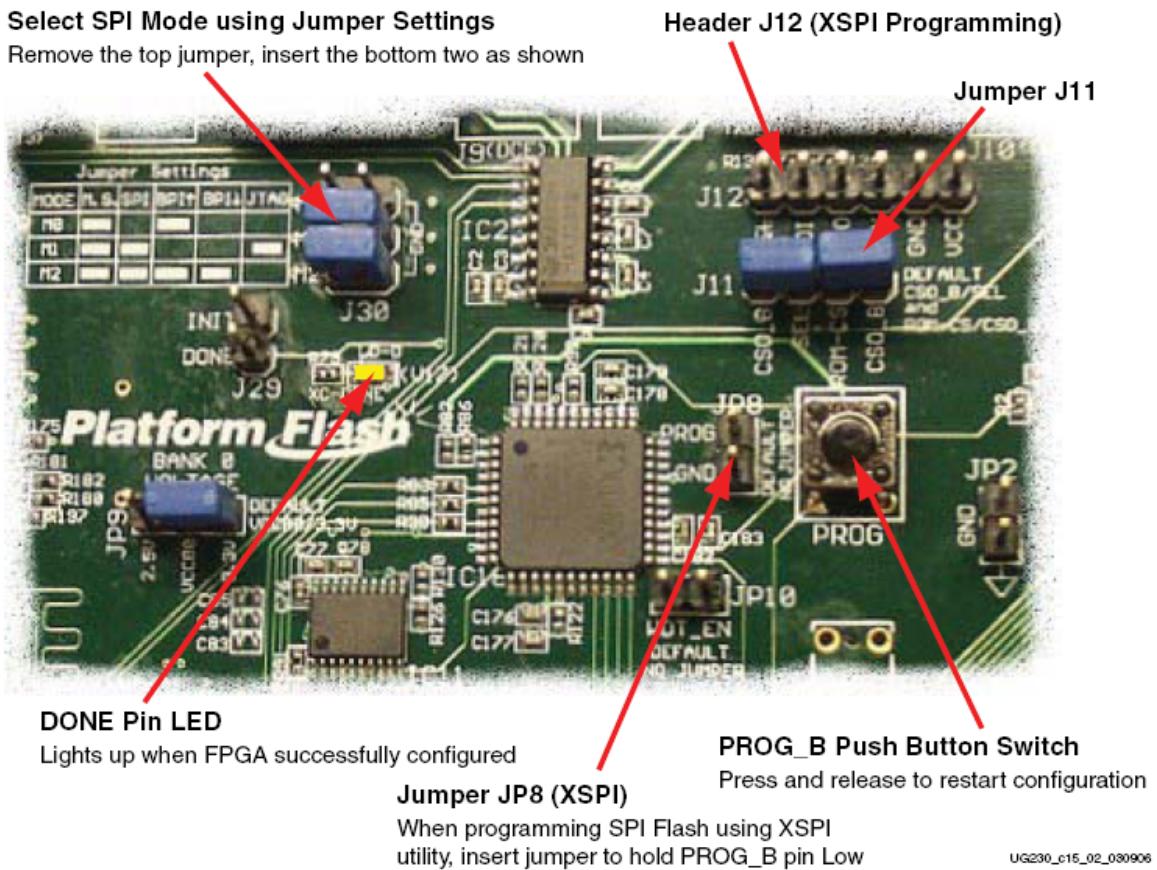
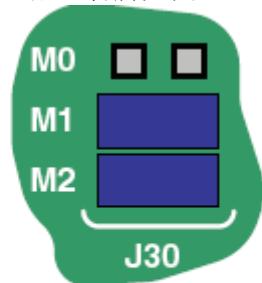


Figure 12-3: Configuration Options for SPI Mode

### Setting the FPGA Mode Select Pins

对于SPI模式的FPGA配置管脚见图12.4。跳线的位置见图12.3。



UG230\_c15\_03\_030206

Figure 12-4: Set Mode Pins for SPI Mode

### Creating an SPI Serial Flash PROM File

以下是如何将FPGA的位流文件配置到SPI Serial Flash PROM的步骤。

### Setting the Configuration Clock Rate

选择连接M25P16 SPI serial Flash时，FPGA支持12MHz的配置时钟，具体设置见图12.5。

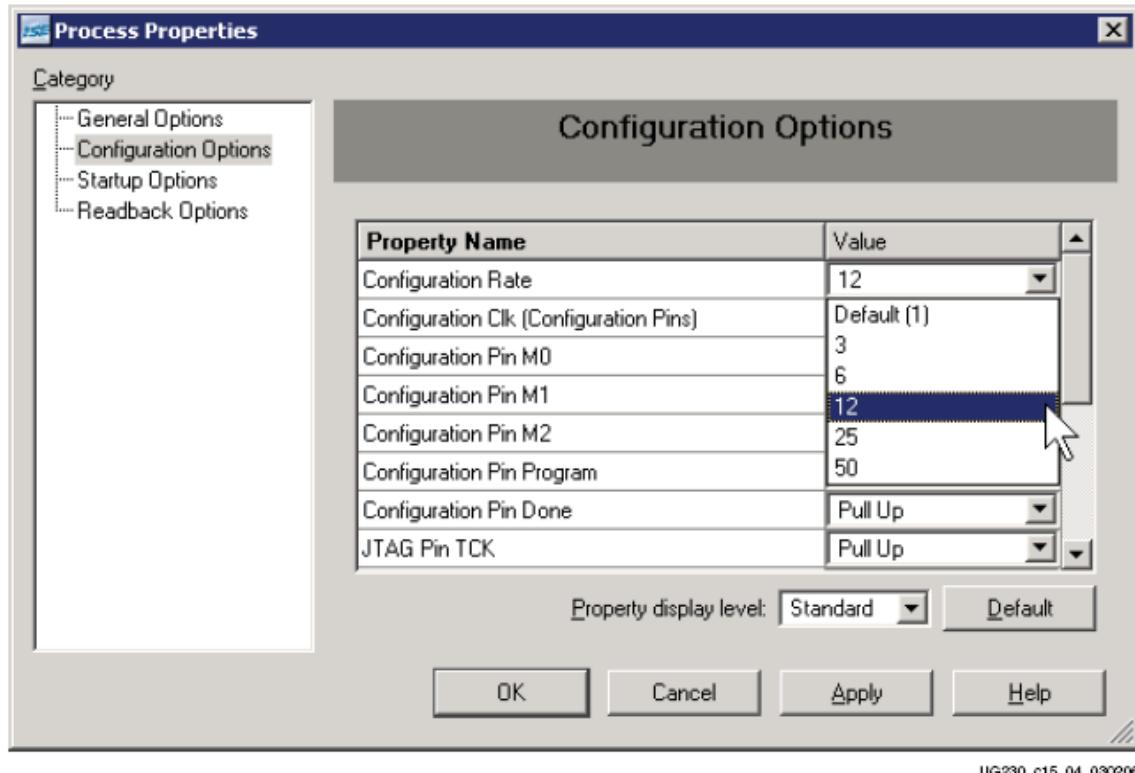


Figure 12-5: Set Configuration Rate to 12 MHz When Using the M25P16 SPI Flash Formatting an SPI Flash PROM File

产生配置文件之后，见图12.6操作。

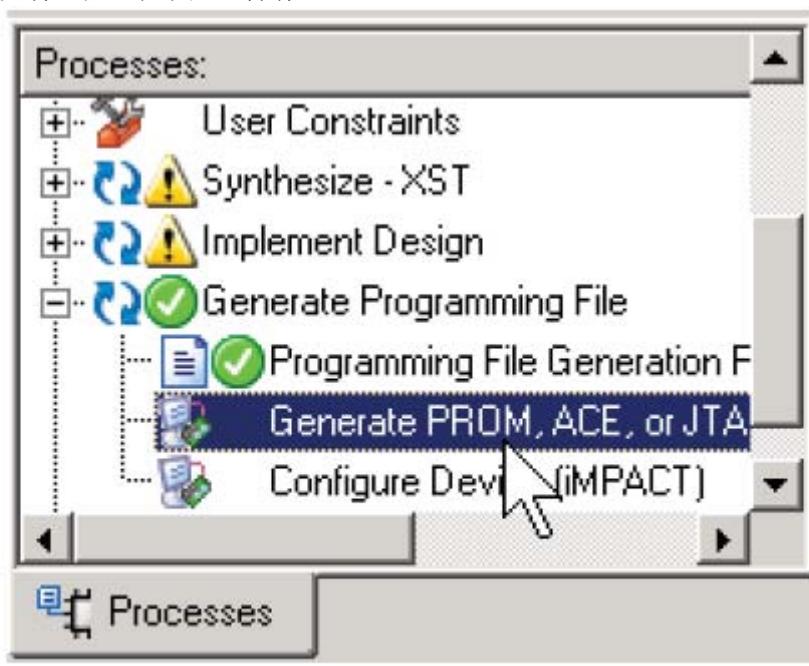
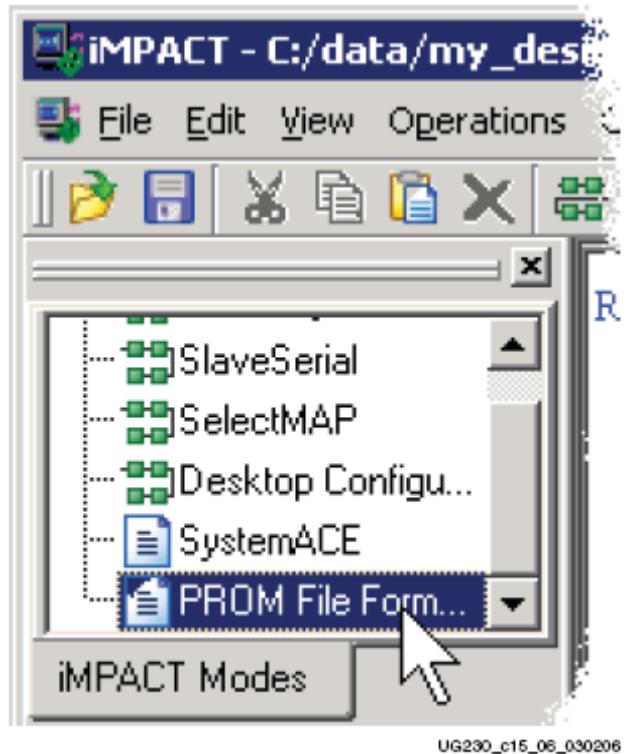


Figure 12-6: Double-Click Generate PROM, ACE, or JTAG File  
启动IMPATC之后，双击PROM File Formatter，见图12.7。



**Figure 12-7: Double-Click PROM File Formatter**

接下来如图12.8、12.9、12.10、12.11所示进行相应的操作。

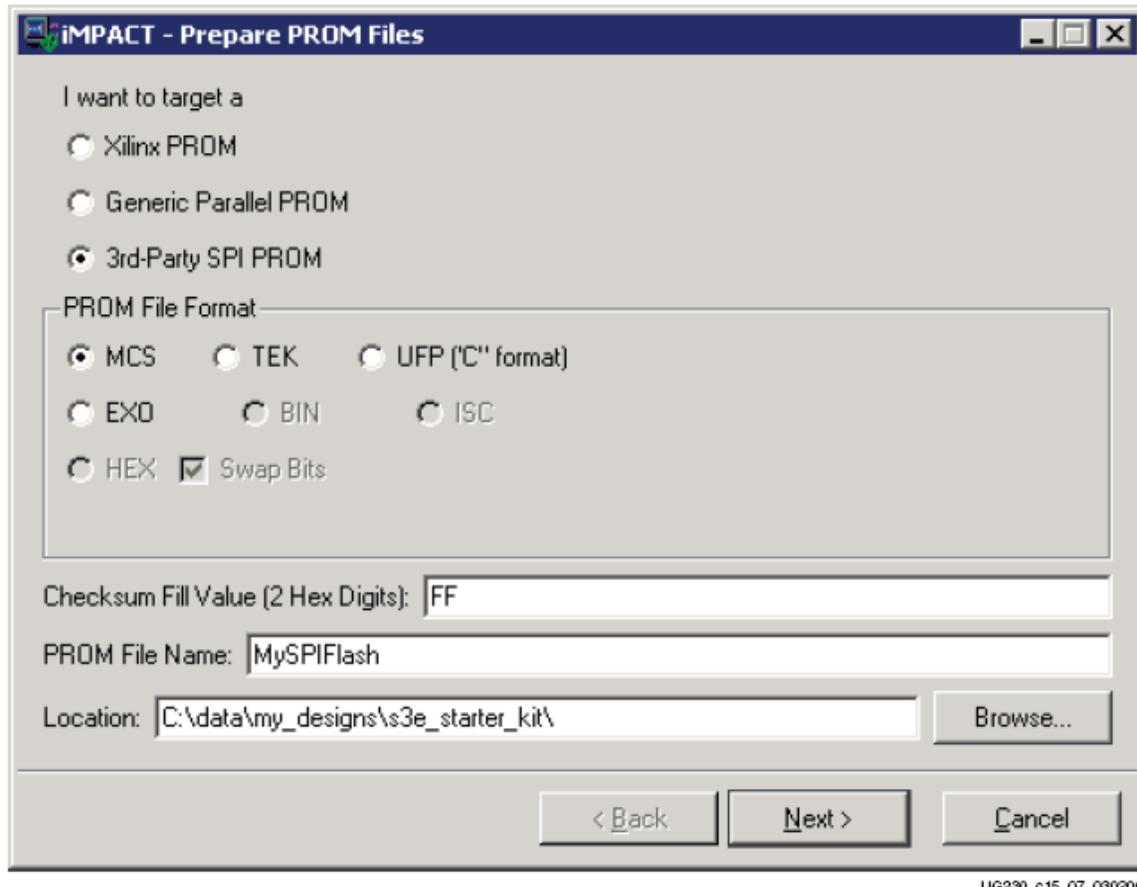


Figure 12-8: Choose the PROM Target Type, the Data Format, and File Location

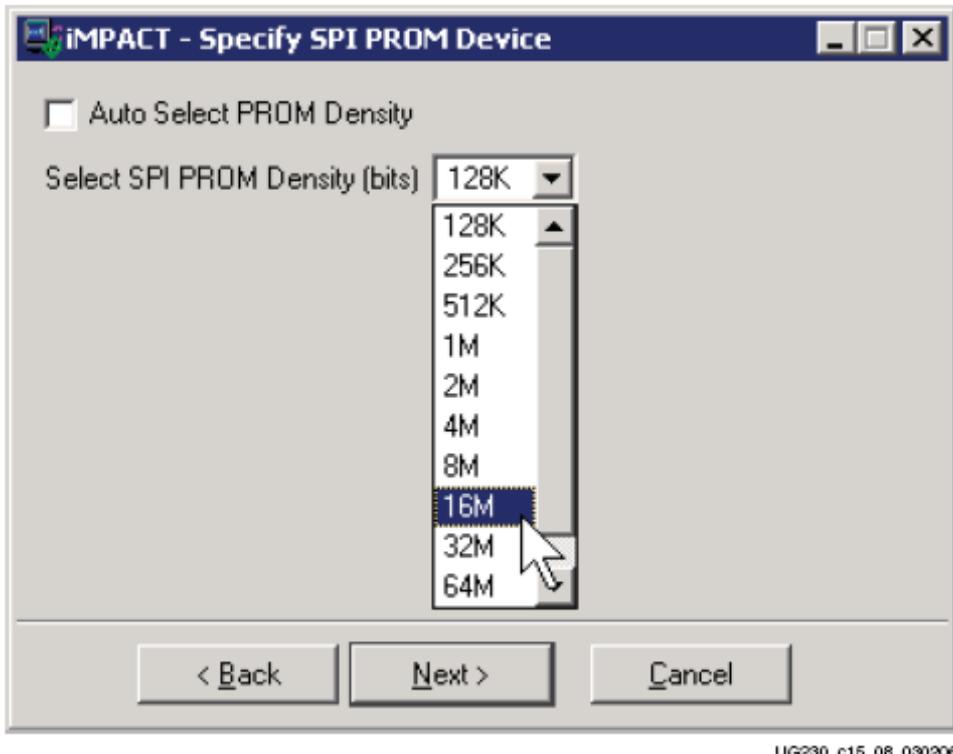


Figure 12-9: Choose 16M

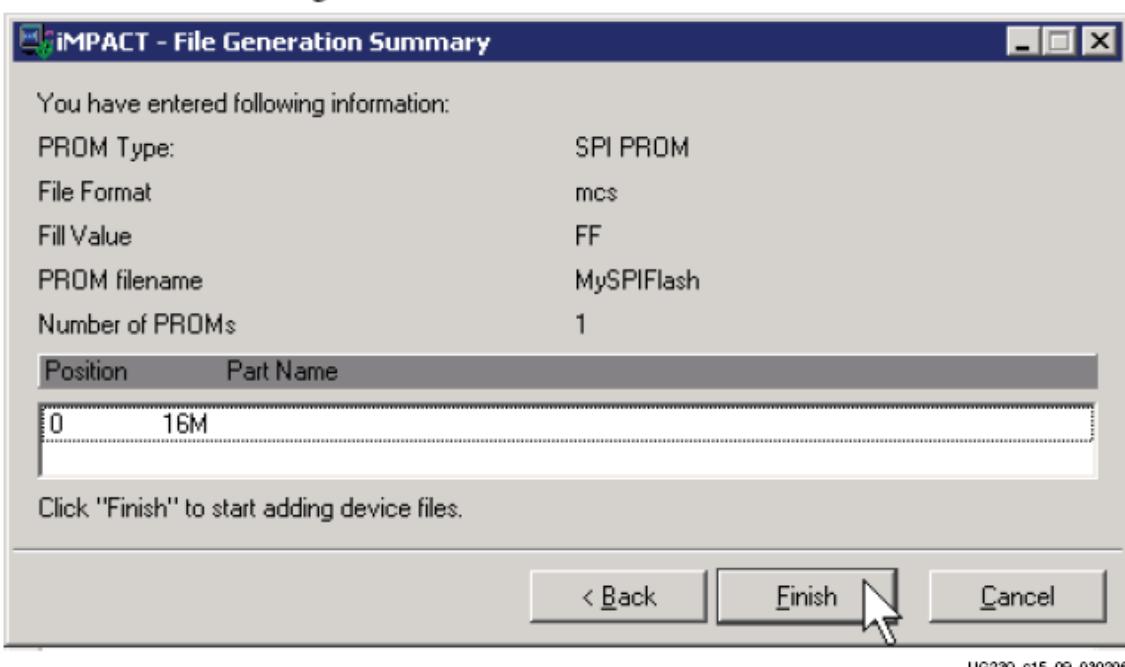
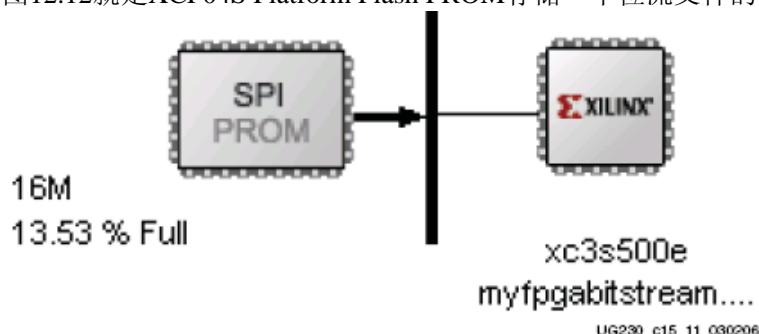


Figure 12-10: Click Finish after Entering PROM Formatter Settings



**Figure 12-11: Enter FPGA Configuration Bitstream File(s)**

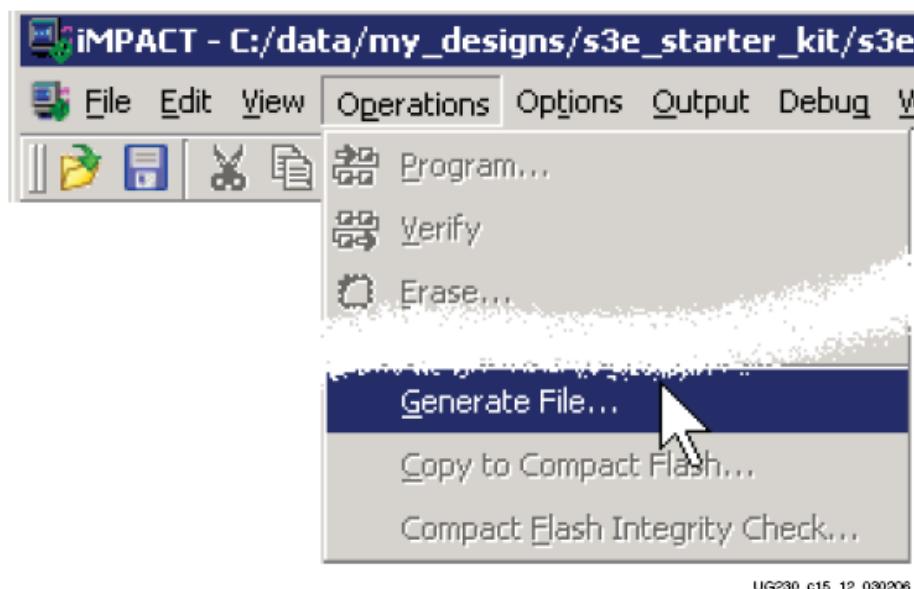
当PROM完成格式后，IMPACT会示出PROM的当前状态，所选的位流文件以及位流文件占用PROM的空间。图12.12就是XCF04S Platform Flash PROM存储一个位流文件的例子。



**Figure 12-12: PROM Formatting Completed**

To generate the actual PROM file, click **Operations → Generate File** as shown in Figure 12-13.

要产生当前PROM文件，如图12.13所示进行操作。



**Figure 12-13: Click Operations → Generate File to Create the Formatted PROM File**

如图12.14所示，IMPACT示出PROM文件成功生成。见图12.8的设置，PROM Formatter产生一个输出文件。在本例中，输出文件名为**MySPIFlash.mcs**。

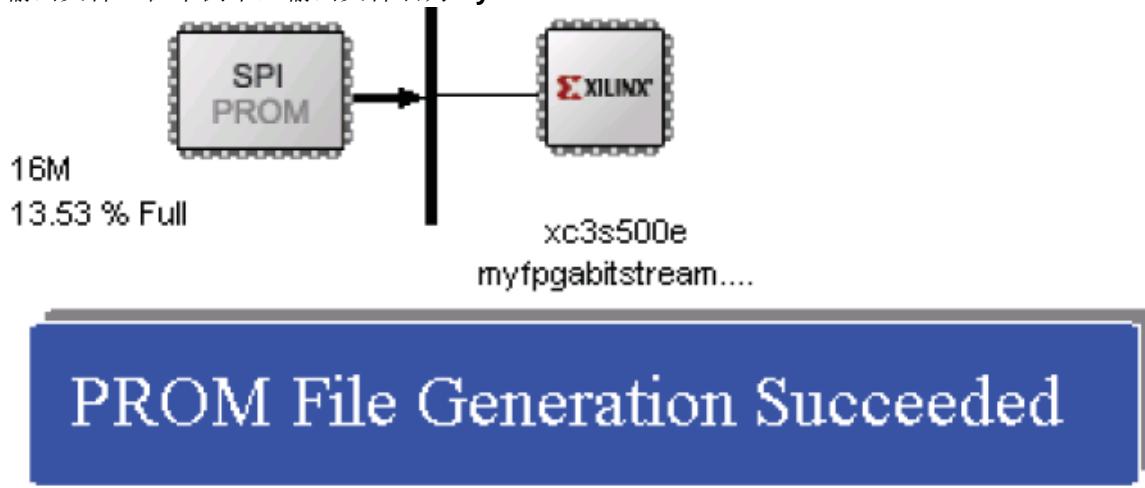


Figure 12-14: PROM File Formatter Succeeded

### Downloading the Design to SPI Flash

对SPI Flash进行编程有多种方法，如下：1) 采用带有XAPP445的XSPI编程软件。通过采用JTAG编程电缆（板上没有提供）由并行口下载到SPI Flash上；2) 采用基于SPI Flash的PicoBlaze参考设计。使用终端模拟器，如Hyperlink，通过PC的串口将SPI Flash的编程数据下载到FPGA。嵌入式PicoBlaze处理器对SPI serial Flash进行编程；3) 采用JTAG工具通过FPGA 的JTAG链接将SPI Flash编程到FPGA上。其链接见相应的参考资料；4) ISE8.2i提供有另外的编程支持。

### Downloading the SPI Flash using XSPI

以下是使用XSPI编程下载到SPI Flash的步骤。

#### Download and Install the XSPI Programming Utility

XSPI编程软件的应用见应用笔记XAPP445。

#### Attach a JTAG Parallel Programming Cable

XSPI编程软件采用JTAG并口编程电缆，如：带飞行引线的Xilinx Parallel Cable IV和Digilent JTAG3 编程电缆。

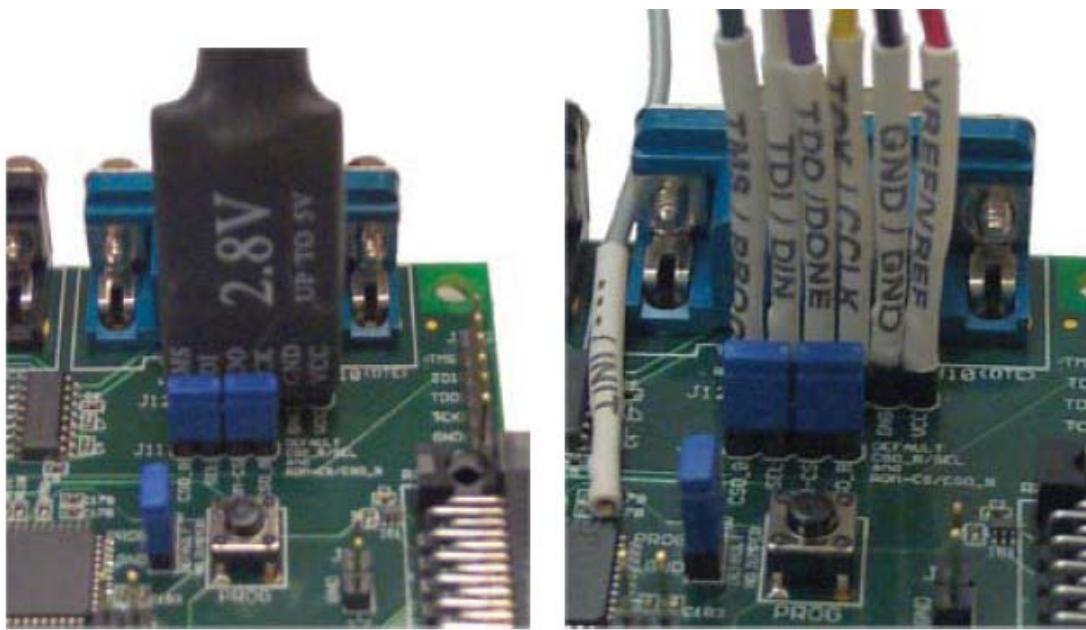
这些电缆并没有随实验板一起提供，但可以单独购买。

首先，关掉实验板的电源。

如果板上连有USB电缆，拔掉。因为同时连接USB电缆和并口电缆会使IMPACT无法正确识别。

将JTAG的一端连到J12，如图12.15a所示。J12的物理位置见图12.3所示。将J12的连接头直接连到SPI Flash管脚上，并不是连到JTAG链上。

将JTAG3电缆直接连到J12上。JTAG3的标签对着J11跳线。如果使用飞行引线，它们必须如图12.15b和表12.2所示连接。注意引线的颜色。灰色的INIT引线留着不用。



a) JTAG3 Parallel Connector

b) Parallel Cable III or Parallel Cable IV  
with Flying Leads

UG230\_c15\_14\_030206

*Figure 12-15: Attaching a JTAG Parallel Programming Cable to the Board*  
*Table 12-2: Cable Connections to J12 Header*

Cable and Labels	Connections					
J12 Header Label	SEL	SDI	SDO	SCK	GND	VCC
JTAG3 Cable Label	TMS	TDI	TDO	TCK	GND	VCC
Flying Leads Label	TMS/ PROG	TDI/ DIN	TDO/ DONE	TCK/ CCLK	GND/ GND	VREF/ VREF

#### Insert Jumper on JP8 and Hold PROG\_B Low

JTAG并行编程电缆直接与SPI Flash管脚相连。为了避免与FPGA的信号相冲突，确保连接FPGA的管脚处于高阻态。在JP8（在PROG按钮附近）安装一个跳线迫使FPGA的PROG\_B管脚为低，见图12.16所示。



a) No Jumper: FPGA Operational (default)    b) Jumper Installed: FPGA Held in Configuration State, I/Os in High Impedance

UG230\_c15\_15\_030206

*Figure 12-16: Installing the JP8 Jumper Holds the FPGA in Configuration State*

将实验板重新上电。

Programming the SPI Flash with the XSPI Software

打开一个命令提示符或DOS窗口，改变XSPI安装路径。

XSPI安装软件会有一个使用向导，附在XAPP445上。提示符下敲打xspi查看快速帮助。

在提示符下敲打如下命令，使用事先生成的SPI-formatted Flash 文件来编程SPI Flash。这种读写SPI Flash需要一个M25P16 SPI Flash，然后擦除、编程，最后写Flash。

```
C:\xspi>xspi -spi_dev m25p16 -spi_epv -mcs -i MySPIFlash.mcs -o output.txt
```

屏幕上会出现一个免责公告。按Enter键继续。整个编程过程大约1分钟多，如图12.17所示。

```
---< Press ENTER to accept notice and continue >---

Start : Mon Feb 27 13:37:07 2006

--> Checking SPI device [STMicro_M25P16_ver_00100] ID code(s)
  - density = [2097152] bytes
    = [16777216] bits
  - mfg_code = [0x20]
  - memory_type = [0x20]
  - density_code = [0x15]
+-----+
| Device ID code(s) check =====> [ OK ] |
+-----+

--> Operation: Erase
--> Operation: Program and Verify using file [MySPIFlash.mcs]
Programmed [283776] of [283776] bytes (w/ polling)
Verified [283776] of [283776] bytes (0 errors)

--> Total byte mismatches [0] (see [temp.txt])
Finish : Mon Feb 27 13:38:22 2006
Elapsed clock time (00:01:15) = 75 seconds
```

Figure 12-17: Programming the M25P16 SPI Flash with the XSPI Programming Utility

SPI Flash编程后，卸掉JP8，如图12.16a所示。如果编程正确，FPGA会从SPI Flash PROM自动配置，且DONE LED点亮。

## Additional Design Details

图12.18提供了SPI Flash接口使用的一些其它细节。在大多数应用中，该接口的简单应用正如图12.1所示那样。但是，实验板支持多种配置方式并给出了Spartan-3E的一些其它性能。

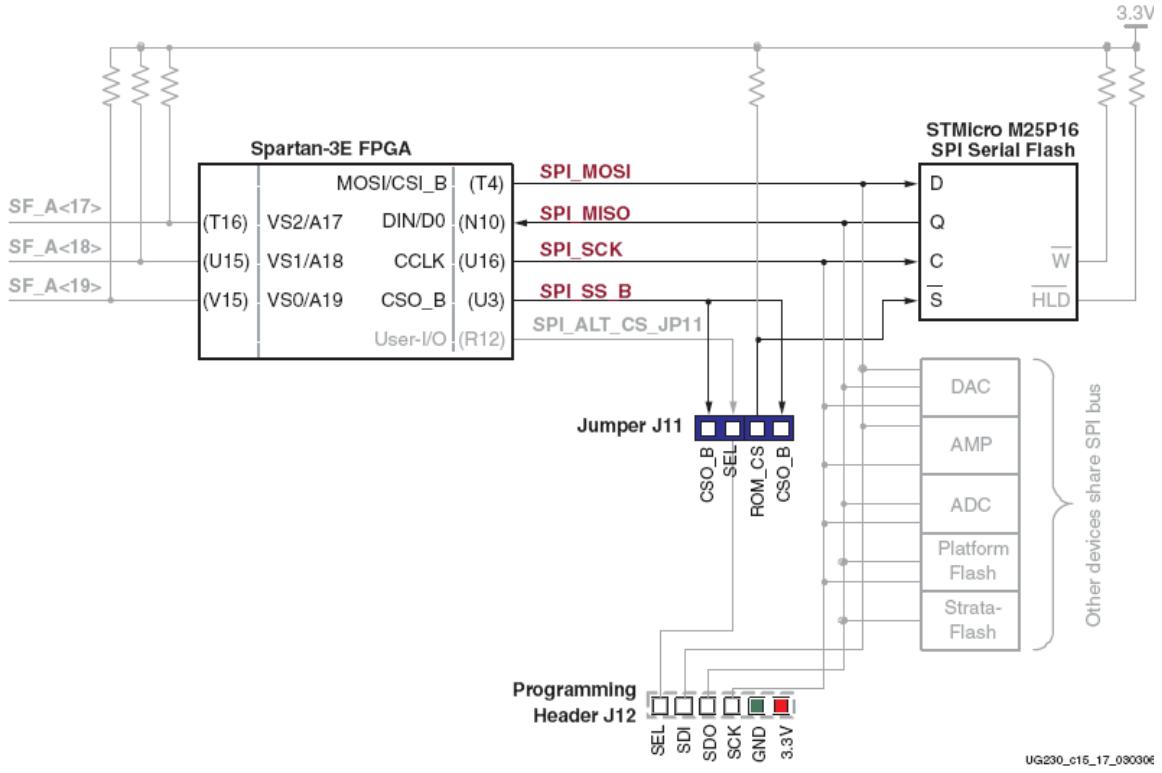


Figure 12-18: Additional SPI Flash Interface Design Details

### Shared SPI Bus with Peripherals

配置之后，SPI Flash的配置管脚可以作为设计的其它应用。实验板上的SPI总线与其它的带有SPI的外围器件复用，如图12.18所示。配置后，要访问SPI Flash存储器，FPGA必须失能其它复用PCI总线的器件。表12.3给出了这些器件的信号名和失能值。

Table 12-3: Disable Other Devices on SPI Bus

Signal	Disabled Device	Disable Value
DAC_CS	Digital-to-Analog Converter (DAC)	1
AMP_CS	Programmable Pre-Amplifier	1
AD_CONV	Analog-to-Digital Converter (ADC)	0
SF_CE0	StrataFlash Parallel Flash PROM	1
FPGA_INIT_B	Platform Flash PROM	1

### Other SPI Flash Control Signals

M25P16 SPI Flash有两个额外的输入控制。低电平有效的写保护输入信号W和总线保持输入信号没用，通过将外部上拉电阻置高来实现。

### Variant Select Pins, VS[2:0]

处于SPI配置模式时，FPGA通过采样三个管脚VS[2: 0]的值来判定哪个SPI的读命令送与SPI Flash。对于M25P16 Flash，VS[2: 0]=111是其正确的命令字。VS[2: 0]管脚通过外部的上拉电阻接到3.3V。在BPI配置模式时，VS[2: 0]管脚是并行永久性Flash的地址线A[19:17]，这些信号连接到StrataFlash parallel Flash PROM。在SPI配置后，VS[2: 0]管脚变成用户编程I/O口，允许完全访问StrataFlash parallel Flash PROM，而不管来自SPI Flash的FGPA配置。

### Jumper Block J11

在SPI配置模式时，FPGA通过置低CSO\_B管脚来选中SPI Flash。CSO\_B连到J11跳线上。该

跳线提供一个选项来移动板上的SPI Flash到不同的选择线上（SPI\_ALT\_CS\_JP11）。通过改变J11的设置以及将要使用的SPI Flash连到JP12上就可以测试不同的SPI Flash器件。默认下，这两个跳线都置在J11里头。

## Programming Header J12

如图12.15所示，J12使用一个JTAG编程电缆来编程板上的SPI Flash。

## Multi-Package Layout

STMicroelectronics能够自动识别M25Pxx SPI系列Flash的封装。实验板支持16Mbit器件所有的三种封装类型，如图12.19所示。默认情况下，实验板以8-lead, 8x6 mm MLP封装形式进行工作。多封装同样支持8-pin SOIC 和16-pin SOIC的封装。8-pin SOIC的管脚1和MLP在板上的左上角。然而，16-pin SOIC的管脚1却在右上角，原因是这种封装是90度的。16-pin SOIC封装形式的每个面有4个管脚没有连接到板上。这些管脚必须悬空。为何要支持多封装形式？一句话，灵活性。多封装布局提供了：

- 1) 在小密度和大密度的SPI Flash PROM之间轻松实现密度移植。并非所有的SPI Flash 的密度都可以在不同的封装形式里用。SPI Flash 的移植策略取决于FPGA提供的移植输出管脚的精细程度。
- 2) 在FPGA的密度移植之间实现PROM配置的兼容性（一致性）。Spartan-3E FPGA's FG320的管脚封装形式支持XC3S500E、XC3S1200E和XC3S1600E 的FPGA器件，而无须修改。SPI Flash 的多种封装形式允许在相关联的PROM之间实现灵活的兼容。从而在板上为FPGA运行最优尺寸的SPI Flash。
- 3) 提供安全措施。如果一种特定的SPI Flash密度在想要的封装里不可用，则可以安全地转换到不同的封装形式或不同的密度。

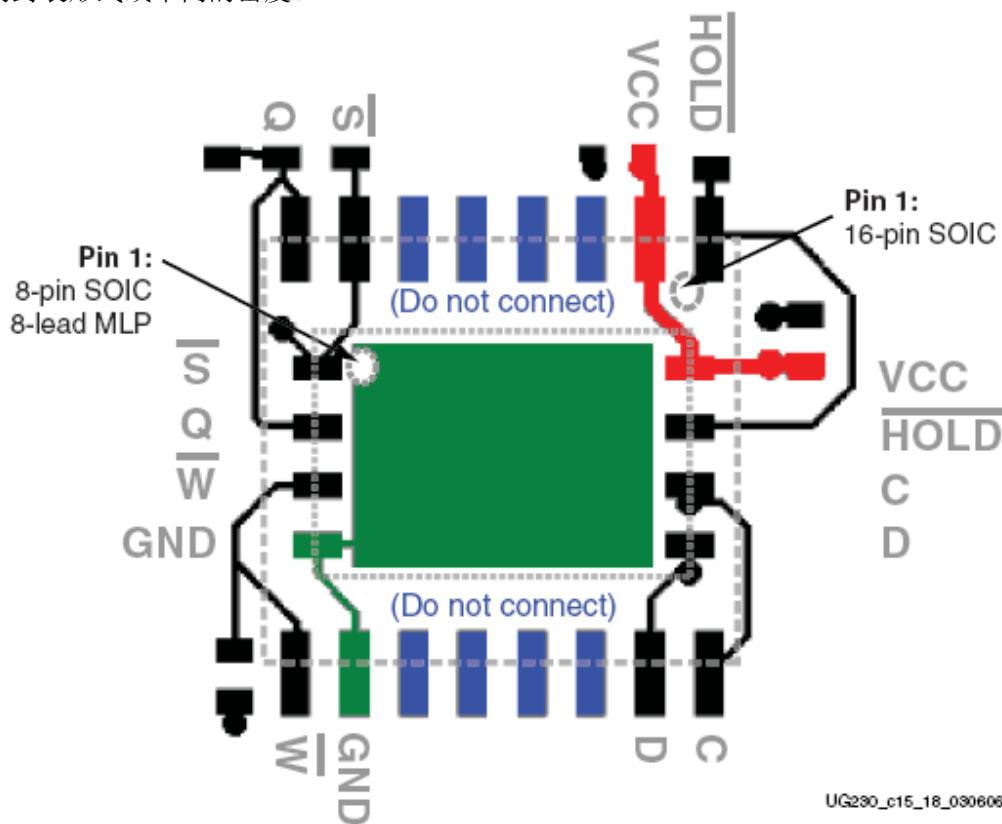


Figure 12-19: Multi-Package Layout for the STMicroelectronics M25Pxx Family

# Chapter 13: DDR SDRAM

实验板包括一个16位接口的512 Mbit (32M x 16) Micron Technology DDR SDRAM (MT46V32M16)，如图13.1所示。所有的 DDR SDRAM 接口的管脚连接到 FPGA's I/O Bank 3。I/O Bank 3 和DDR SDRAM 均由2.5V供电，由板上5V的输入电压经LTC3412调整得到。1.25V 的参考电压一般由2.5V电压经电阻分压得到，供给FPGA 和 DDR SDRAM。

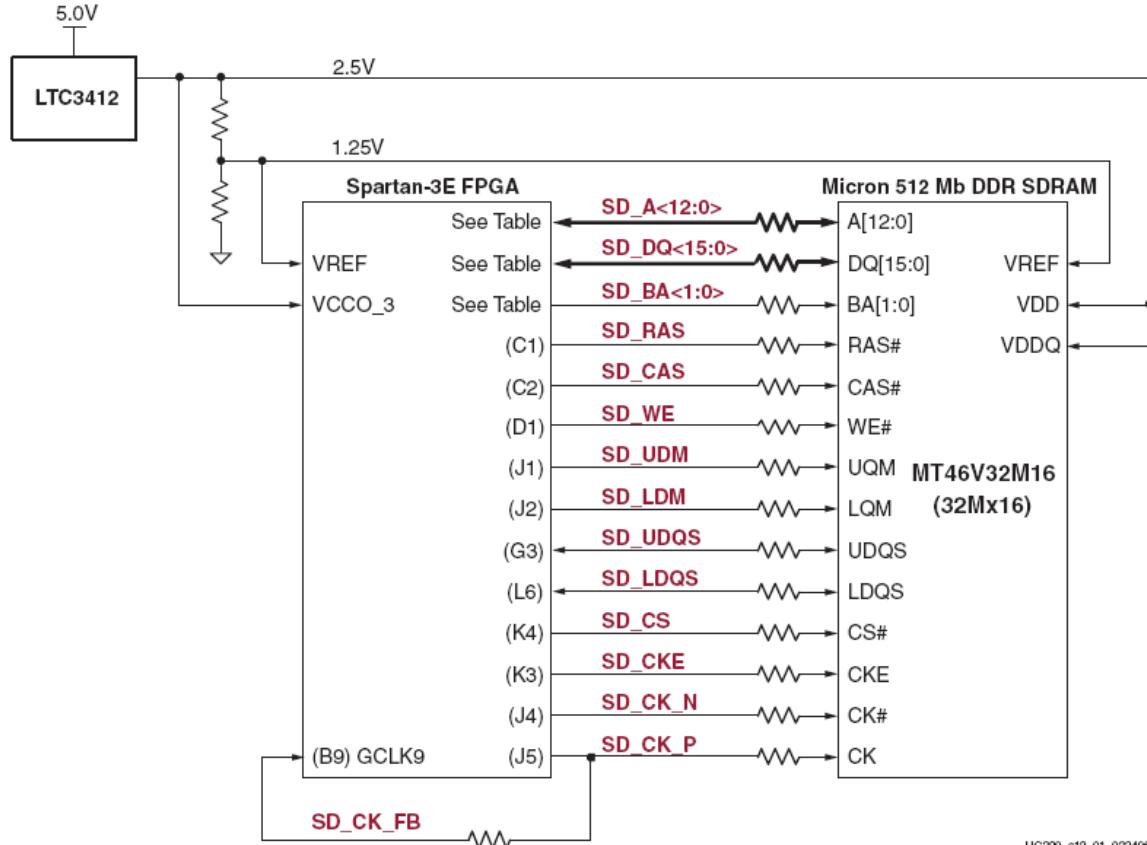


Figure 13-1: FPGA Interface to Micron 512 Mbit DDR SDRAM

所有的DDR SDRAM接口信号都是终端的。

差分时钟管脚SD\_CK\_P反馈给FPGA I/O Bank 0的管脚B9，以获得最佳通向FPGA's Digital Clock Managers (DCMs)的路径。当使用到MicroBlaze OPB DDR控制器时就要用到该路径了。从EDK 8.1i 开发软件同样可以获得MicroBlaze OPB DDR SDRAM 控制器的IP 核。

## DDR SDRAM Connections

表13.1给出了FPGA与DDR SDRAM之间的连接。

**Table 13-1: FPGA-to-DDR SDRAM Connections**

Category	DDR SDRAM Signal Name	FPGA Pin Number	Function
Address	SD_A12	P2	Address inputs
	SD_A11	N5	
	SD_A10	T2	
	SD_A9	N4	
	SD_A8	H2	
	SD_A7	H1	
	SD_A6	H3	
	SD_A5	H4	
	SD_A4	F4	
	SD_A3	P1	
	SD_A2	R2	
	SD_A1	R3	
	SD_A0	T1	

Table 13-1: FPGA-to-DDR SDRAM Connections (*Continued*)

Category	DDR SDRAM Signal Name	FPGA Pin Number	Function
Data	SD_DQ15	H5	Data input/output
	SD_DQ14	H6	
	SD_DQ13	G5	
	SD_DQ12	G6	
	SD_DQ11	F2	
	SD_DQ10	F1	
	SD_DQ9	E1	
	SD_DQ8	E2	
	SD_DQ7	M6	
	SD_DQ6	M5	
	SD_DQ5	M4	
	SD_DQ4	M3	
	SD_DQ3	L4	
	SD_DQ2	L3	
	SD_DQ1	L1	
	SD_DQ0	L2	
Control	SD_BA1	K6	Bank address inputs
	SD_BA0	K5	
	SD_RAS	C1	Command inputs
	SD_CAS	C2	
	SD_WE	D1	
	SD_CK_N	J4	Differential clock input
	SD_CK_P	J5	
	SD_CKE	K3	Active-High clock enable input
	SD_CS	K4	Active-Low chip select input
	SD_UDM	J1	Data Mask. Upper and Lower data masks
	SD_LDM	J2	
	SD_UDQS	G3	Data Strobe. Upper and Lower data strobes
	SD_LDQS	L6	
	SD_CK_FB	B9	SDRAM clock feedback into top DCM within FPGA. Used by some DDR SDRAM controller cores

## UCF Location Constraints

### Address

图13.2给出了DDR SDRAM地址线管脚的UCF约束。

NET "SD_A<12>" LOC = "P2"	IOSTANDARD = SSTL2_I ;
NET "SD_A<11>" LOC = "N5"	IOSTANDARD = SSTL2_I ;
NET "SD_A<10>" LOC = "T2"	IOSTANDARD = SSTL2_I ;
NET "SD_A<9>" LOC = "N4"	IOSTANDARD = SSTL2_I ;
NET "SD_A<8>" LOC = "H2"	IOSTANDARD = SSTL2_I ;
NET "SD_A<7>" LOC = "H1"	IOSTANDARD = SSTL2_I ;
NET "SD_A<6>" LOC = "H3"	IOSTANDARD = SSTL2_I ;
NET "SD_A<5>" LOC = "H4"	IOSTANDARD = SSTL2_I ;
NET "SD_A<4>" LOC = "F4"	IOSTANDARD = SSTL2_I ;
NET "SD_A<3>" LOC = "P1"	IOSTANDARD = SSTL2_I ;
NET "SD_A<2>" LOC = "R2"	IOSTANDARD = SSTL2_I ;
NET "SD_A<1>" LOC = "R3"	IOSTANDARD = SSTL2_I ;
NET "SD_A<0>" LOC = "T1"	IOSTANDARD = SSTL2_I ;

*Figure 13-2: UCF Location Constraints for DDR SDRAM Address Inputs*  
**Data**

图13.3给出了DDR SDRAM数据线管脚的UCF约束。

NET "SD_DQ<15>" LOC = "H5"	IOSTANDARD = SSTL2_I ;
NET "SD_DQ<14>" LOC = "H6"	IOSTANDARD = SSTL2_I ;
NET "SD_DQ<13>" LOC = "G5"	IOSTANDARD = SSTL2_I ;
NET "SD_DQ<12>" LOC = "G6"	IOSTANDARD = SSTL2_I ;
NET "SD_DQ<11>" LOC = "F2"	IOSTANDARD = SSTL2_I ;
NET "SD_DQ<10>" LOC = "F1"	IOSTANDARD = SSTL2_I ;
NET "SD_DQ<9>" LOC = "E1"	IOSTANDARD = SSTL2_I ;
NET "SD_DQ<8>" LOC = "E2"	IOSTANDARD = SSTL2_I ;
NET "SD_DQ<7>" LOC = "M6"	IOSTANDARD = SSTL2_I ;
NET "SD_DQ<6>" LOC = "M5"	IOSTANDARD = SSTL2_I ;
NET "SD_DQ<5>" LOC = "M4"	IOSTANDARD = SSTL2_I ;
NET "SD_DQ<4>" LOC = "M3"	IOSTANDARD = SSTL2_I ;
NET "SD_DQ<3>" LOC = "L4"	IOSTANDARD = SSTL2_I ;
NET "SD_DQ<2>" LOC = "L3"	IOSTANDARD = SSTL2_I ;
NET "SD_DQ<1>" LOC = "L1"	IOSTANDARD = SSTL2_I ;
NET "SD_DQ<0>" LOC = "L2"	IOSTANDARD = SSTL2_I ;

*Figure 13-3: UCF Location Constraints for DDR SDRAM Data I/Os*  
**Control**

图13.4给出了DDR SDRAM控制线管脚的UCF约束。

```

NET "SD_BA<0>" LOC = "K5" | IOSTANDARD = SSTL2_I ;
NET "SD_BA<1>" LOC = "K6" | IOSTANDARD = SSTL2_I ;
NET "SD_CAS" LOC = "C2" | IOSTANDARD = SSTL2_I ;
NET "SD_CK_N" LOC = "J4" | IOSTANDARD = SSTL2_I ;
NET "SD_CK_P" LOC = "J5" | IOSTANDARD = SSTL2_I ;
NET "SD_CKE" LOC = "K3" | IOSTANDARD = SSTL2_I ;
NET "SD_CS" LOC = "K4" | IOSTANDARD = SSTL2_I ;

NET "SD_LDM" LOC = "J2" | IOSTANDARD = SSTL2_I ;
NET "SD_LDQS" LOC = "L6" | IOSTANDARD = SSTL2_I ;
NET "SD_RAS" LOC = "C1" | IOSTANDARD = SSTL2_I ;
NET "SD_UDM" LOC = "J1" | IOSTANDARD = SSTL2_I ;
NET "SD_UDQS" LOC = "G3" | IOSTANDARD = SSTL2_I ;
NET "SD_WE" LOC = "D1" | IOSTANDARD = SSTL2_I ;
# Path to allow connection to top DCM connection
NET "SD_CK_FB" LOC = "B9" | IOSTANDARD = LVCMOS33 ;

```

*Figure 13-4: UCF Location Constraints for DDR SDRAM Control Pins*

### Reserve FPGA VREF Pins

图13.5是I/O Bank 3 五个参考输入电压VREF的管脚约束。这些管脚不能用于通用I/O。禁止软件使用这些管脚，其约束如下。

```

# Prohibit VREF pins
CONFIG PROHIBIT = D2;
CONFIG PROHIBIT = G4;
CONFIG PROHIBIT = J6;
CONFIG PROHIBIT = L5;
CONFIG PROHIBIT = R4;

```

*Figure 13-5: UCF Location Constraints for StrataFlash Control Pins*

# *Chapter 14*

## *10/100 Ethernet Physical Layer Interface*

入门实验板包括一个标准的微系统——LAN83C185 10/100以太物理层（PHY）接口和一个RJ-45连接器，如图14.1所示。FPGA内置一个以太介质控制器（MAC），入门实验板便可以连到标准的以太网上。板上的25M晶振控制所有的时序。

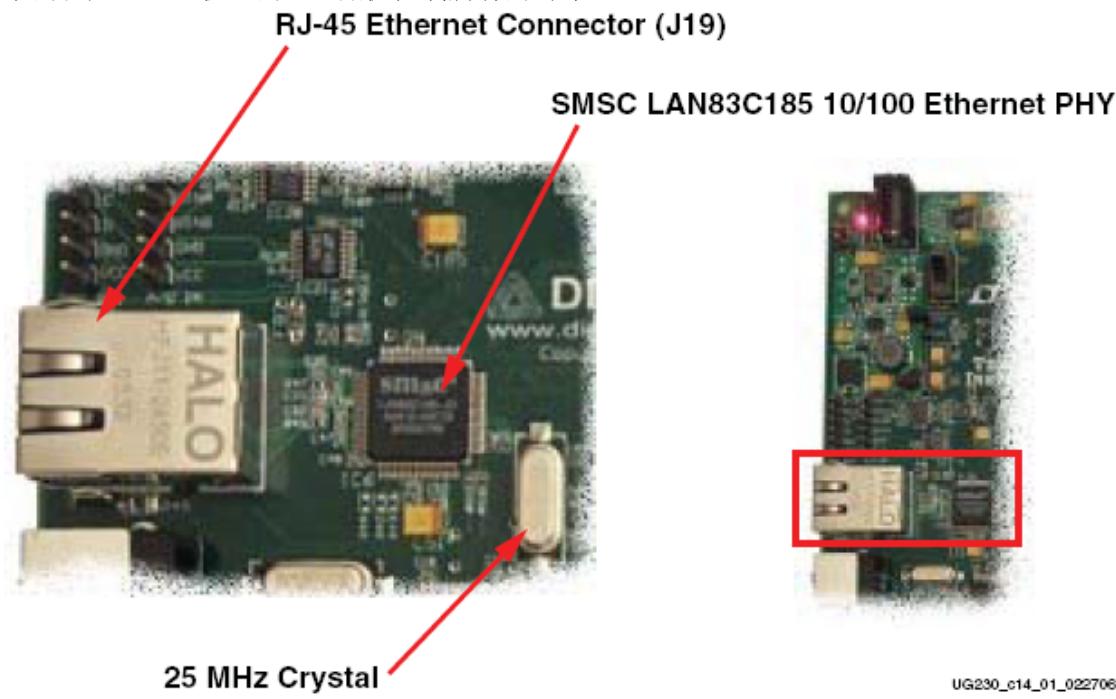
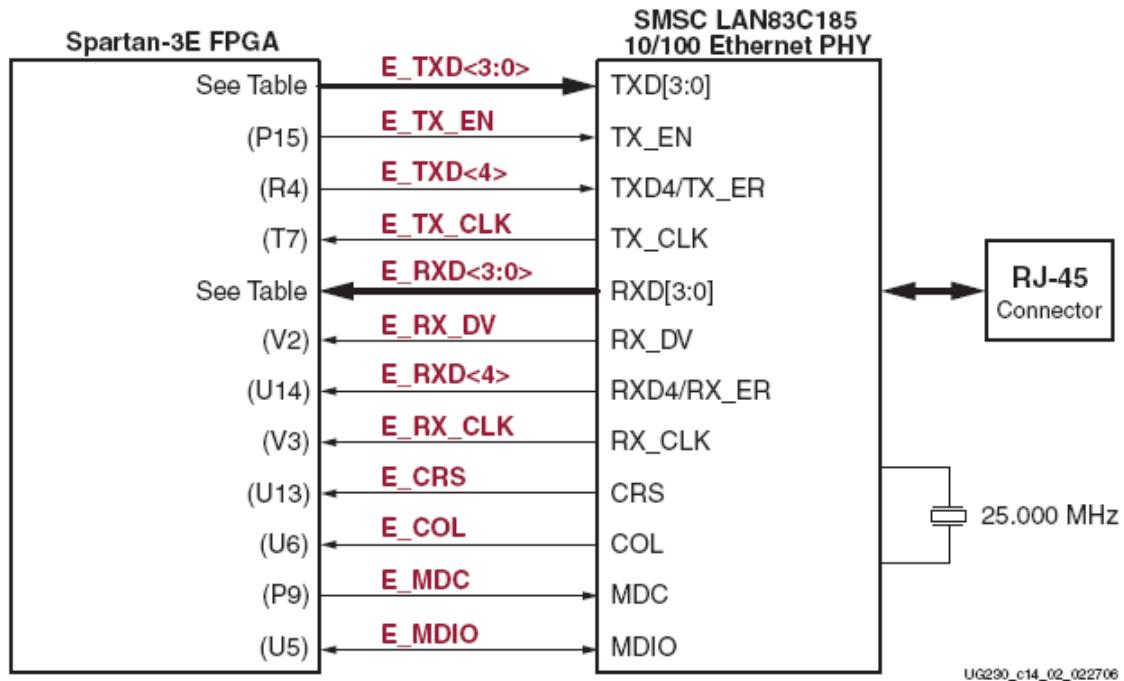


Figure 14-1: 10/100 Ethernet PHY with RJ-45 Connector

### **Ethernet PHY Connections**

FPGA通过MII连接到LAN83C185 Ethernet PHY，如图14.2。详细信息看表14.1。



UG290\_c14\_02\_022706

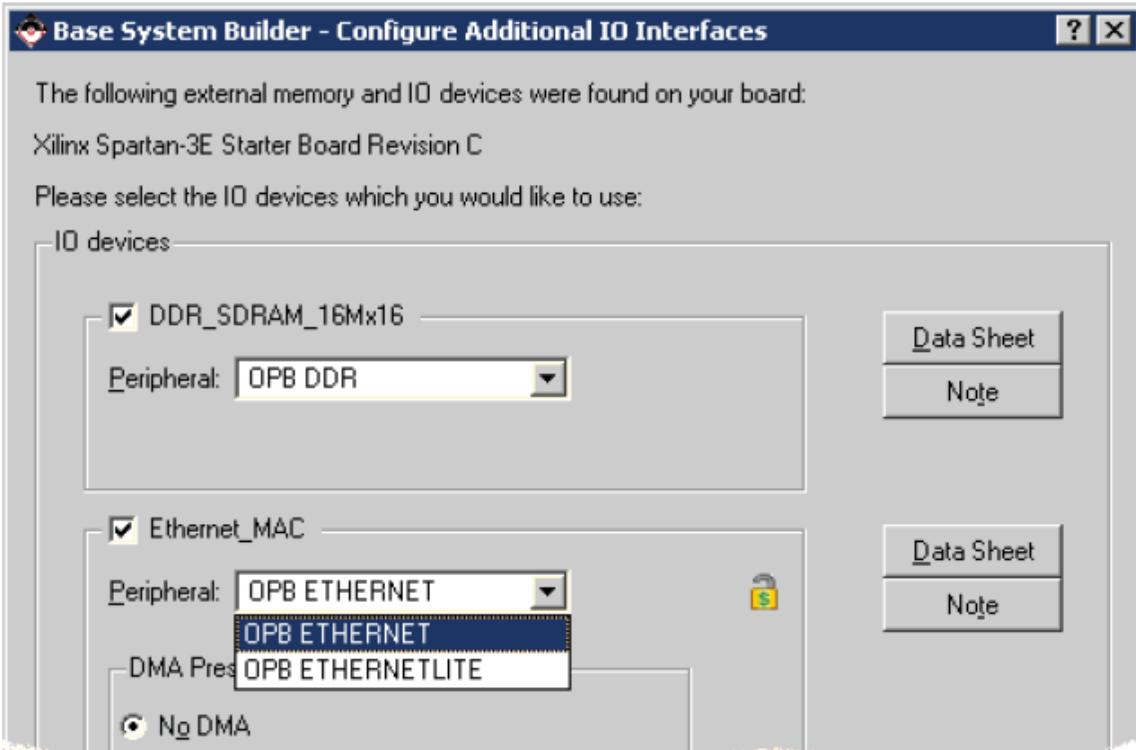
Figure 14-2: FPGA Connects to Ethernet PHY via MII

Table 14-1: FPGA Connections to the LAN83C185 Ethernet PHY

Signal Name	FPGA Pin Number	Function
E_TXD<4>	R6	Transmit Data to the PHY. E_TXD<4> is also the MII Transmit Error.
E_TXD<3>	T5	
E_TXD<2>	R5	
E_TXD<1>	T15	
E_TXD<0>	R11	
E_TX_EN	P15	Transmit Enable.
E_TX_CLK	T7	Transmit Clock. 25 MHz in 100Base-TX mode, and 2.5 MHz in 10Base-T mode.
E_RXD<4>	U14	Receive Data from PHY.
E_RXD<3>	V14	
E_RXD<2>	U11	
E_RXD<1>	T11	
E_RXD<0>	V8	
E_RX_DV	V2	Receive Data Valid.
E_RX_CLK	V3	Receive Clock. 25 MHz in 100Base-TX mode, and 2.5 MHz in 10Base-T mode.
E_CRS	U13	Carrier Sense
E_COL	U6	MII Collision Detect.
E_MDC	P9	Management Clock. Serial management clock.
E_MDIO	U5	Management Data Input/Output.

## MicroBlaze Ethernet IP Cores

PHY 主要为 MicroBlaze 所用。例如，MAC 是 EDK 开发平台的一部分。



MAC 核需要设计约束以满足任务要求。细节请参考 OPB—以太 MAC 数据手册。OPB 总线的时钟频率必须是 65MHz 或高于 100Mbps 以太传输速率和 6.5MHz 或快于 10Mbps 以太传输速率。

## UCF Location Constraints

图14.4是10/100 Ethernet PHY interface接口的UCF约束，包括I/O管脚分配和标准。

```

NET "E_COL"      LOC = "U6"      | IOSTANDARD = LVCMOS33 ;
NET "E_CRS"       LOC = "U13"     | IOSTANDARD = LVCMOS33 ;
NET "E_MDC"       LOC = "P9"      | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8 ;
NET "E_MDIO"      LOC = "U5"      | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8 ;
NET "E_RX_CLK"    LOC = "V3"      | IOSTANDARD = LVCMOS33 ;
NET "E_RX_DV"     LOC = "V2"      | IOSTANDARD = LVCMOS33 ;
NET "E_RXD<0>"  LOC = "V8"      | IOSTANDARD = LVCMOS33 ;
NET "E_RXD<1>"  LOC = "T11"     | IOSTANDARD = LVCMOS33 ;
NET "E_RXD<2>"  LOC = "U11"     | IOSTANDARD = LVCMOS33 ;
NET "E_RXD<3>"  LOC = "V14"     | IOSTANDARD = LVCMOS33 ;
NET "E_RXD<4>"  LOC = "U14"     | IOSTANDARD = LVCMOS33 ;
NET "E_TX_CLK"    LOC = "T7"      | IOSTANDARD = LVCMOS33 ;
NET "E_TX_EN"     LOC = "P15"     | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8 ;
NET "E_TXD<0>"  LOC = "R11"     | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8 ;
NET "E_TXD<1>"  LOC = "T15"     | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8 ;
NET "E_TXD<2>"  LOC = "R5"      | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8 ;
NET "E_TXD<3>"  LOC = "T5"      | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8 ;
NET "E_TXD<4>"  LOC = "R6"      | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8 ;

```

Figure 14-4: UCF Location Constraints for 10/100 Ethernet PHY Inputs

# Chapter 15: Expansion Connectors

实验板提供一个多样化的扩展连接器，以便于与其它板外的器件轻松连接。其扩展 I/O 的连接头如下：

- 1) 一个 100 管脚 hirose 扩展连接器与 FPGA 的 43 个用户 I/O 口相连，包括 15 个差分 LVDS I/O 对和 2 个只输入对；
- 2) 三个 6 管脚扩展连接器；
- 3) 为 Agilent or Tektronix 无线探针提供的焊盘垫。

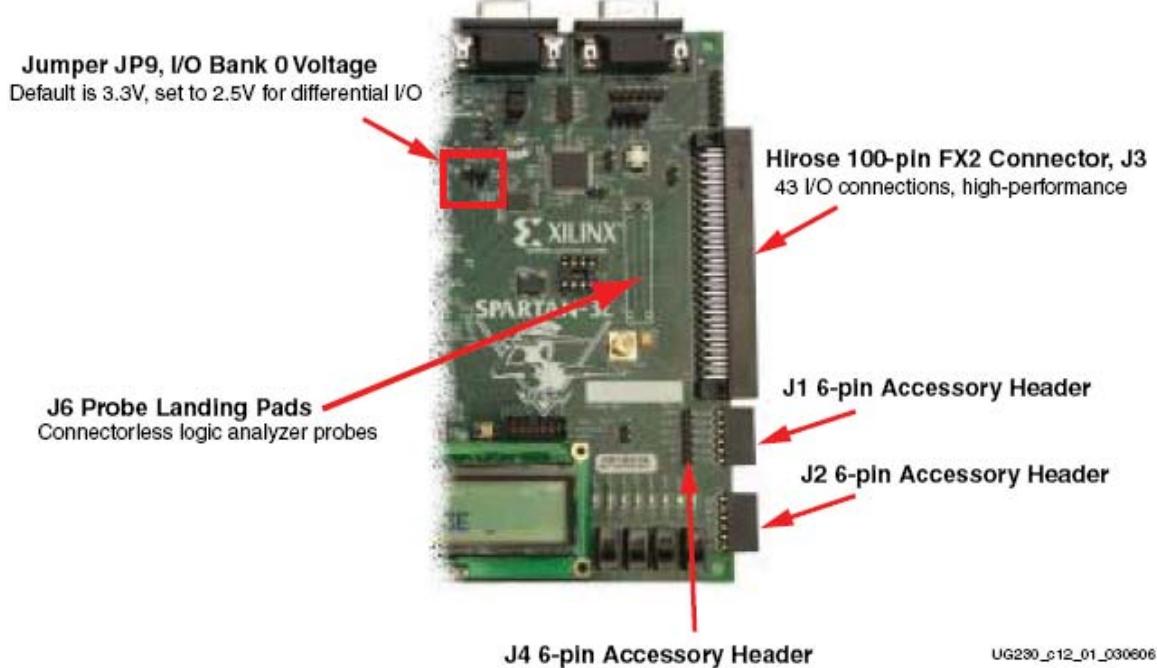


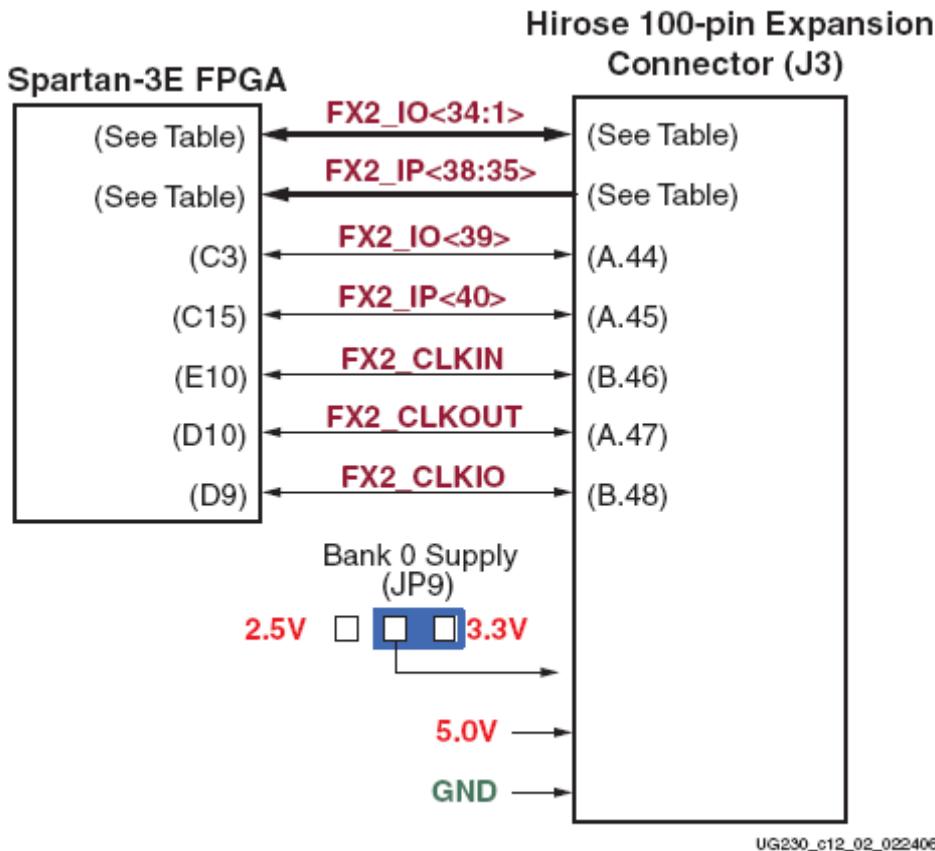
Figure 15-1: Expansion Headers

## Hirose 100-pin FX2 Edge Connector (J3)

100管脚扩展连接器在板上的右边沿，见图15.1所示。该连接器是FX2-100P-1.27DS Hirose1.27mm 的接头。本文中简称其为 FX2 连接器。

图15.2所示，43个FPGA的I/O管脚与FX2连接器相连。除了其中的5个只输入管脚外，双向的I/O均可驱动或接收信号。这5个管脚是：FX2\_IP<38:35>和FX2\_IP<40>。见表15.1，这些管脚中标为绿颜色的不能驱动FX2连接器，但可以接收信号。

尽管所有的三个连接到全I/O管脚，但这三个信号主要保留作为实验板与FX2连接器之间的时钟信号。



UG230\_c12\_02\_022406

Figure 15-2: FPGA Connections to the Hirose 100-pin Edge Connector

### Voltage Supplies to the Connector

实验板通过两个供给电源给 100 管脚 hirose FX 扩展连接器以及其它的板上附件提供电源，见图 15.2。5.0V 的电压给所有的板上附件的 5V 逻辑或调节器。

一个分离的电源单独供给FPGA的0号I/O口同样的电压。连接到Hirose 的FPGA的 I/O 接口都在0号上。默认下，0号的I/O的供给电压是3.3V。但是，使用JP9可以将电压转变为2.5V。一些FPGA的I/O标准，特别是差分标准诸如RSDS和LVDS——需要2.5V的输出电压。

为支持高速信号通过该连接器，FX2 连接器B面的大多数管脚接地。

### Connector Pinout and FPGA Connections

表15.1给出了100管脚hirose FX扩展连接器输出管脚与FPGA管脚的连线。FX2有2排连接器，每排有50个管脚，见表15.1淡黄色部分。

表15.1同样给出了与8位LED、3个6-管脚的附属接头（J1、J2、J4）以及无线调试接头J6的复用的连线。

Table 15-1: Hirose 100-pin FX2 Connector Pinout and FPGA Connections (J3)

Signal Name	FPGA Pin	Shared Header Connections					FX2 Connector		FPGA Pin	Signal Name
		LED	J1	J2	JP4	J6	A (top)	B (bottom)		
	VCCO_0						1	1		SHIELD
	VCCO_0						2	2	GND	GND
TMS_B							3	3		TDO_XC2C
JTSEL							4	4		TCK_B
TDO_FX2							5	5	GND	GND
FX2_IO1	B4		◆			◆	6	6	GND	GND
FX2_IO2	A4		◆			◆	7	7	GND	GND
FX2_IO3	D5		◆			◆	8	8	GND	GND
FX2_IO4	C5		◆			◆	9	9	GND	GND
FX2_IO5	A6			◆		◆	10	10	GND	GND
FX2_IO6	B6			◆		◆	11	11	GND	GND
FX2_IO7	E7			◆		◆	12	12	GND	GND
FX2_IO8	F7			◆		◆	13	13	GND	GND
FX2_IO9	D7				◆	◆	14	14	GND	GND
FX2_IO10	C7				◆	◆	15	15	GND	GND
FX2_IO11	F8				◆	◆	16	16	GND	GND
FX2_IO12	E8				◆	◆	17	17	GND	GND
FX2_IO13	F9	LD7				◆	18	18	GND	GND
FX2_IO14	E9	LD6				◆	19	19	GND	GND
FX2_IO15	D11	LD5				◆	20	20	GND	GND
FX2_IO16	C11	LD4				◆	21	21	GND	GND
FX2_IO17	F11	LD3				◆	22	22	GND	GND
FX2_IO18	E11	LD2				◆	23	23	GND	GND
FX2_IO19	E12	LD1					24	24	GND	GND
FX2_IO20	F12	LD0					25	25	GND	GND
FX2_IO21	A13						26	26	GND	GND
FX2_IO22	B13						27	27	GND	GND
FX2_IO23	A14						28	28	GND	GND
FX2_IO24	B14						29	29	GND	GND
FX2_IO25	C14						30	30	GND	GND
FX2_IO26	D14						31	31	GND	GND
FX2_IO27	A16						32	32	GND	GND
FX2_IO28	B16						33	33	GND	GND
FX2_IO29	E13						34	34	GND	GND

Table 15-1: Hirose 100-pin FX2 Connector Pinout and FPGA Connections (J3) (Continued)

Signal Name	FPGA Pin	Shared Header Connections					FX2 Connector		FPGA Pin	Signal Name
		LED	J1	J2	JP4	J6	A (top)	B (bottom)		
FX2_IO30	C4						35	35	GND	GND
FX2_IO31	B11						36	36	GND	GND
FX2_IO32	A11						37	37	GND	GND
FX2_IO33	A8						38	38	GND	GND
FX2_IO34	G9						39	39	GND	GND
FX2_IP35	D12						40	40	GND	GND
FX2_IP36	C12						41	41	GND	GND
FX2_IP37	A15						42	42	GND	GND
FX2_IP38	B15						43	43	GND	GND
FX2_IO39	C3						44	44	GND	GND
FX2_IP40	C15						45	45	GND	GND
GND	GND						46	46	E10	FX2_CLKIN
FX2_CLKOUT	D10						47	47	GND	GND
GND	GND						48	48	D9	FX2_CLKIO
5.0V							49	49		5.0V
5.0V							50	50		SHIELD

## Compatible Board

与Spartan-3E实验板的FX2相兼容的是来自Digilent的VDEC1 Video Decoder Board。

## Mating Receptacle Connectors

实验板使用Hirose FX2-100P-1.27DS的连接器。该接头可以连接到其它任何兼容的100管脚的连接器，包括贴板的和非带锁电缆的连接器。

## Differential I/O

FX2、J3支持高达15个差分I/O对和两对采用LVDS或RSDS I/O标准的只输入对，见表15.2所示。所有的I/O对支持差分终端输入（DIFF\_TERM）。选择对可通过外部终端电阻随意匹配。

这些信号并不布线到匹配差分阻抗，这需要考虑到最终的性能。但是，所有的路径对于最小的偏移有相似的长度。

Table 15-2: Differential I/O Pairs

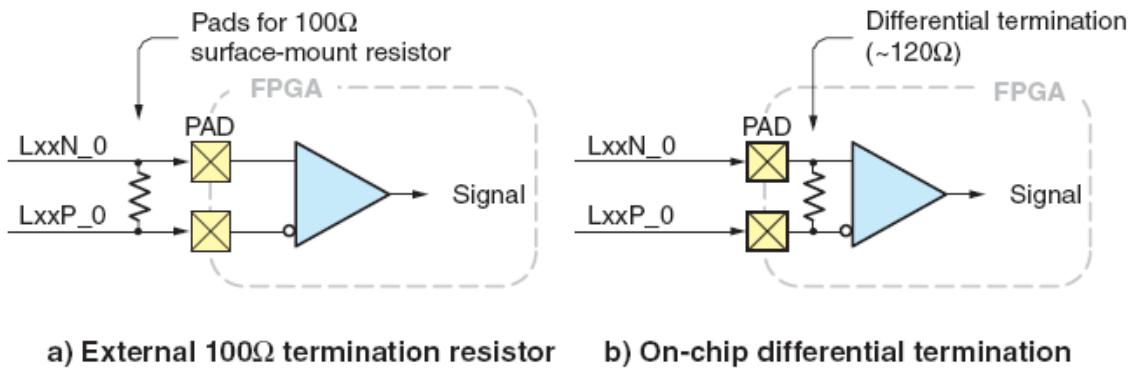
Differential Pair	Signal Name	FPGA Pins	FPGA Pin Name	Direction	DIFF_TERM	External Resistor Designator
1	FX2_IO1	B4	IO_L24N_0	I/O	Yes	
	FX2_IO2	A4	IO_L24P_0	I/O	Yes	
2	FX2_IO3	D5	IO_L23N_0	I/O	Yes	
	FX2_IO4	C5	IO_L23P_0	I/O	Yes	
3	FX2_IO5	A6	IO_L20N_0	I/O	Yes	
	FX2_IO6	B6	IO_L20P_0	I/O	Yes	
4	FX2_IO7	E7	IO_L19N_0	I/O	Yes	
	FX2_IO8	F7	IO_L19P_0	I/O	Yes	
5	FX2_IO9	D7	IO_L18N_0	I/O	Yes	
	FX2_IO10	C7	IO_L18P_0	I/O	Yes	
6	FX2_IO11	F8	IO_L17N_0	I/O	Yes	
	FX2_IO12	E8	IO_L17P_0	I/O	Yes	
7	FX2_IO13	F9	IP_L15N_0	I/O	Yes	
	FX2_IO14	E9	IP_L15P_0	I/O	Yes	
8	FX2_IO15	D11	IP_L09N_0	I/O	Yes	
	FX2_IO16	C11	IP_L09P_0	I/O	Yes	
9	FX2_IO17	F11	IO_L08N_0	I/O	Yes	R202
	FX2_IO18	E11	IO_L08P_0	I/O	Yes	
10	FX2_IO19	E12	IO_L06N_0	I/O	Yes	R203
	FX2_IO20	F12	IO_L06P_0	I/O	Yes	
11	FX2_IO21	A13	IO_L05P_0	I/O	Yes	R204
	FX2_IO22	B13	IO_L05N_0	I/O	Yes	
12	FX2_IO23	A14	IO_L04N_0	I/O	Yes	R205
	FX2_IO24	B14	IO_L04P_0	I/O	Yes	
13	FX2_IO25	C14	IO_L03N_0	I/O	Yes	R206
	FX2_IO26	D14	IO_L03P_0	I/O	Yes	
14	FX2_IO27	A16	IO_L01N_0	I/O	Yes	R207
	FX2_IO28	B16	IO_L01P_0	I/O	Yes	
15	FX2_IP35	D12	IP_L07N_0	Input		R208
	FX2_IP36	C12	IP_L07P_0	Input		
16	FX2_IP37	A15	IP_L02N_0	Input		R209
	FX2_IP38	B15	IP_L02P_0	Input		
17	FX2_CLKIN	E10	IO_L11N_0/ GCLK5	I/O	Yes	R210
	FX2_CLKOUT	D10	IO_L11P_0/ GCLK4	I/O	Yes	

## Using Differential Inputs

LVDS和RSDS差分输入需要输入终端。有两种可选措施。其一是使用外部终端电阻，见图15.3a。实验板为100欧的外部终端电阻提供焊盘垫。这些电阻并没有焊在板上。电阻的参考设计标在印刷纸上，见标15.2所示。焊盘垫分布在实验板的正反两面，在FPGA与FX2连接器之间。这些电阻并没有焊在板上。使用差分对15和16时，需要外部的终端电阻。

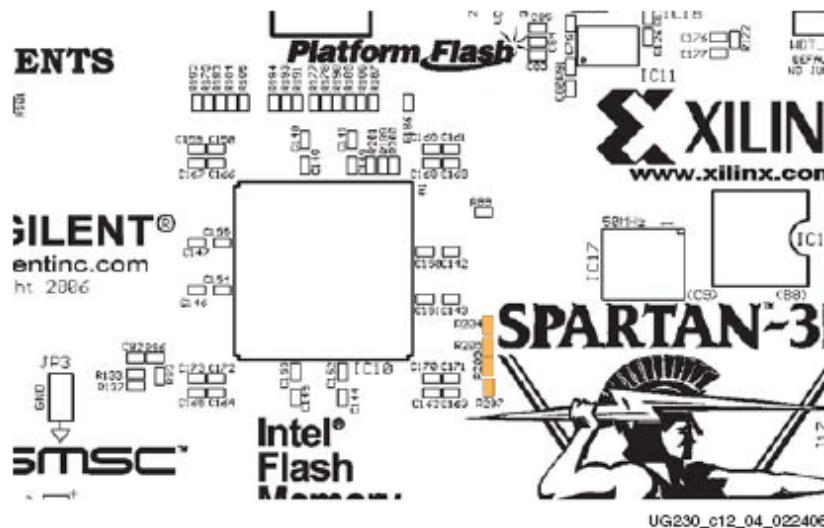
其二见图15.3b所示，利用Spartan-3E的特点——片上差分终端，它是使用差分I/O信号的DIFF\_TERM属性。每个差分I/O管脚包含一个类似于内部大约120欧的终端电阻的线路。片上差

分终端只能在差分I/O对上使用，不能用在只输入对上，如15和16对。



*Figure 15-3: Differential Input Termination Options*

图15.4和图15.5给出了输入差分终端电阻在板上正反两面的焊接位置。表15.2给出了哪个电阻连接具体的差分对。



**Figure 15-4: Location of Termination Resistor Pads on Top Side of Board**

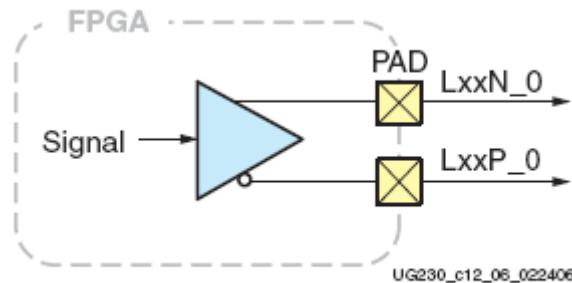


*Figure 15-5: Location of Termination Resistor Pads on Bottom Side of Board Using Differential Outputs*

差分输入信号并要求任何特定的电压。然而，LVDS和RSDS差分输出信号需要I/O Bank 0

的2.5V供给电压。实验板提供方案给I/O Bank 0供给3.3V或2.5V的电压。

如果在FX2连接器上使用差分输出，将JP9设置为2.5V。如果跳线设置错误，输出开关正确，信号电平则于规定的不符合。



*Figure 15-6: Differential Outputs*  
**UCF Location Constraints**

图15.7给出了FX2连接器管脚的UCF约束。

```

# ===== FX2 Connector (FX2) =====
NET "FX2_CLKIN" LOC = "E10" | IOSTANDARD = LVCMOS33 ;
NET "FX2_CLKIO" LOC = "D9" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_CLKOUT" LOC = "D10" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
# These four connections are shared with the J1 6-pin accessory header
NET "FX2_IO<1>" LOC = "B4" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IO<2>" LOC = "A4" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IO<3>" LOC = "D5" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IO<4>" LOC = "C5" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
# These four connections are shared with the J2 6-pin accessory header
NET "FX2_IO<5>" LOC = "A6" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IO<6>" LOC = "B6" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IO<7>" LOC = "E7" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IO<8>" LOC = "F7" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
# These four connections are shared with the J4 6-pin accessory header
NET "FX2_IO<9>" LOC = "D7" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IO<10>" LOC = "C7" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IO<11>" LOC = "F8" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IO<12>" LOC = "E8" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
# The discrete LEDs are shared with the following 8 FX2 connections
#NET "FX2_IO<13>" LOC = "F9" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
#NET "FX2_IO<14>" LOC = "E9" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
#NET "FX2_IO<15>" LOC = "D11" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
#NET "FX2_IO<16>" LOC = "C11" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
#NET "FX2_IO<17>" LOC = "F11" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
#NET "FX2_IO<18>" LOC = "E11" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
#NET "FX2_IO<19>" LOC = "E12" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
#NET "FX2_IO<20>" LOC = "F12" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IO<21>" LOC = "A13" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IO<22>" LOC = "B13" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IO<23>" LOC = "A14" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IO<24>" LOC = "B14" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IO<25>" LOC = "C14" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IO<26>" LOC = "D14" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IO<27>" LOC = "A16" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IO<28>" LOC = "B16" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IO<29>" LOC = "E13" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IO<30>" LOC = "C4" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IO<31>" LOC = "B11" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IO<32>" LOC = "A11" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IO<33>" LOC = "A8" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IO<34>" LOC = "G9" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IP<35>" LOC = "D12" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IP<36>" LOC = "C12" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IP<37>" LOC = "A15" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IP<38>" LOC = "B15" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IO<39>" LOC = "C3" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "FX2_IP<40>" LOC = "C15" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;

```

**Figure 15-7: UCF Location Constraints for Accessory Headers**

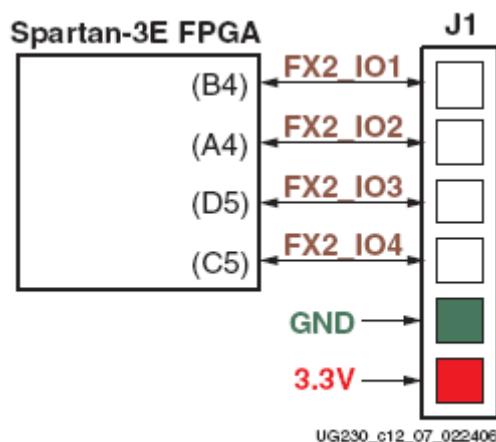
## Six-Pin Accessory Headers

6管脚扩展连接器使用Digilent Peripheral Modules提供了简易的I/O扩展接口。

### Header J1

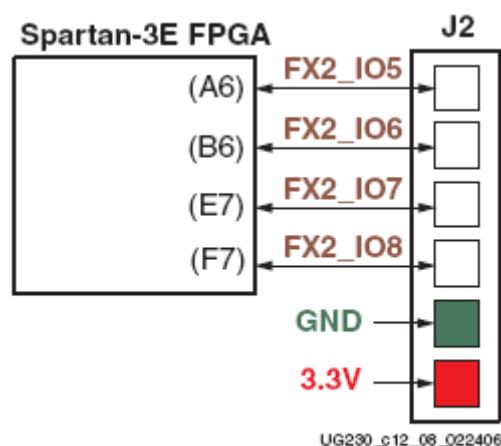
如图15.8所示，在板上的右边缘顶部是6管脚扩展连接器的J1。它采用6管脚90度的插槽。4个FPGA管脚（FX2\_IO<4:1>）连接到J1。这四个信号与FX2连接器复用。实验板向贴在J1的

扩展板提供3.3V的电压。



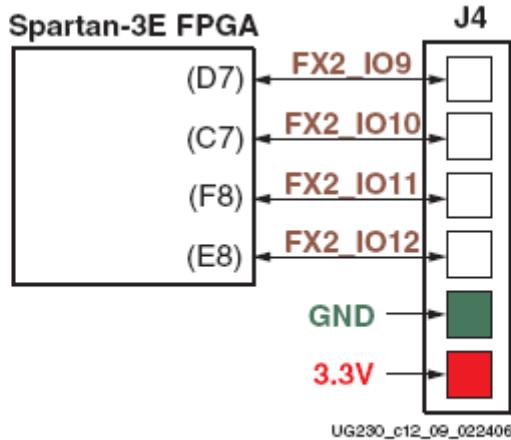
**Figure 15-8: FPGA Connections to the J1 Accessory Header Header J2**

如图15.9所示，在板上的右边缘底部是6管脚扩展连接器的J2。它采用6管脚90度的插槽。4个FPGA管脚（FX2\_IO<8:5>）连接到J2。这四个信号与FX2连接器复用。实验板向贴在J2的扩展板提供3.3V的电压。



**Figure 15-9: FPGA Connections to the J2 Accessory Header Header J4**

如图15.10所示，J4紧挨着J1。它采用0.1英尺圆心桩脚组成6管脚连接器。4个FPGA管脚（FX2\_IO<12:9>）连接到J4。这四个信号与FX2连接器复用。实验板向贴在J4的扩展板提供3.3V的电压。



*Figure 15-10: FPGA Connections to the J4 Accessory Header*  
**UCF Location Constraints**

图15.11是扩展连接器的UCF约束。

```
# ===== 6-pin header J1 =====
# These four connections are shared with the FX2 connector
#NET "J1<0>" LOC = "B4" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 6 ;
#NET "J1<1>" LOC = "A4" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 6 ;
#NET "J1<2>" LOC = "D5" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 6 ;
#NET "J1<3>" LOC = "C5" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 6 ;

# ===== 6-pin header J2 =====
# These four connections are shared with the FX2 connector
#NET "J2<0>" LOC = "A6" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 6 ;
#NET "J2<1>" LOC = "B6" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 6 ;
#NET "J2<2>" LOC = "E7" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 6 ;
#NET "J2<3>" LOC = "F7" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 6 ;

# ===== 6-pin header J4 =====
# These four connections are shared with the FX2 connector
#NET "J4<0>" LOC = "D7" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 6 ;
#NET "J4<1>" LOC = "C7" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 6 ;
#NET "J4<2>" LOC = "F8" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 6 ;
#NET "J4<3>" LOC = "E8" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 6 ;
```

*Figure 15-11: UCF Location Constraints for Accessory Headers*  
**Connectorless Debugging Port Landing Pads (J6)**

见图15.1，和J6连接器一样，实验板为无线连接器调试接口提供了焊盘垫。板上没有实物的连接器，相反，只是无线连接探针。像Agilent一样，向逻辑分析仪提供了一个接口。该调试接口主要为采用Agilent's FPGA动态探针的Xilinx ChipScope Pro软件所用。当然，也可使用FPGA Editor命令来使用Agilent 或Tektronix的探针，而无需ChipScope Pro软件。

表15.3给出了连接器的输出管脚。只有18个FPGA管脚连接到连接器上，连接器上的其它焊盘垫没有用到。这18个管脚与FX2连接器（J3）和6管脚扩展连接器（J1、J2、J4）。

**Table 15-3: Connectorless Debugging Port Landing Pads (J6)**

Signal Name	FPGA Pin	Connectorless Landing Pads		FPGA Pin	Signal Name
FX2_IO1	B4	A1	B1	GND	GND
FX2_IO2	A4	A2	B2	D5	FX2_IO3
GND	GND	A3	B3	C5	FX2_IO4
FX2_IO5	A6	A4	B4	GND	GND
FX2_IO6	B6	A5	B5	E7	FX2_IO7
GND	GND	A6	B6	F7	FX2_IO8
FX2_IO9	D7	A7	B7	GND	GND
FX2_IO10	C7	A8	B8	F8	FX2_IO11
GND	GND	A9	B9	E8	FX2_IO12
FX2_IO13	F9	A10	B10	GND	GND
FX2_IO14	E9	A11	B11	D11	FX2_IO15
GND	GND	A12	B12	C11	FX2_IO16
FX2_IO17	F11	A13	B13	GND	GND
FX2_IO18	E11	A14	B14		
		A15	B15		
		A16	B16		
		A17	B17		
		A18	B18		
		A19	B19		
		A20	B20		
		A21	B21		
		A22	B22		
		A23	B23		
		A24	B24		
		A25	B25		
		A26	B26		
		A27	B27		

# *Chapter 16:*

## **XC2C64A CoolRunner-II CPLD**

入门实验板包括一个Xilinx XC2C64A CoolRunner-II CPLD。CPLD是个用户可编程以及适于消费应用。CPLD的部分功能保留，以调整各种FPGA的配置存储器，即Xilinx Platform Flash PROM和Intel StrataFlash PROM。CPLD必须为用户的应用提供了一下功能。

1) 当FPGA为主串行配置模式 (FPGA\_M<2:0>=000) 时，向XCF04S Platform Flash PROM 提供一个低电平有效的使能信号。Platform Flash PROM在其它任何模式都失能。CPLD减少了跳线的个数，简化了FPGA配置存储器的相互连接。

2) 在BPI-UP配置模式 (FPGA\_M<2:0>=010) FPGA有效时，置StrataFlash PROM 地址线的5个最高位 A[24:20]为二进制的 00000。在BPI-DOWN配置模式 (FPGA\_M<2:0>=011) FPGA有效时，置StrataFlash PROM 地址线的5个最高位 A[24:20]为二进制的11111。在非BPI配置模式或FPGA的DONE管脚为高时，置A[24:20]为ZZZZZ。在BPI模式时，根据5位地址的配置来识别FPGA的实现形式。为什么要增加一个CPLD来模这种实现形式呢？以后的参考设计将说明了这一独特的配置性能。在典型的BPI模式应用中，CPLD并不需要用到。

除了需要CPLD外，用户I/O口管脚13~21和58个宏单元可为用户所用。跳线JP10 (WDT\_EN) 定义了CPLD的 XC\_WDT\_EN信号的状态。默认情况下，该跳线为空，信号线置为高电平。

XC\_PROG\_B从CPLD输出，如果用到的话，必须配置为开漏输出（如：要么置低有效，要么悬空为高阻态，不能为高）。该信号直接连到FPGA的PROG\_B可编程管脚。StrataFlash PROM地址的最高位是SF\_A<24>，同FX2连接器的信号FX2\_IO<32>一致。16M字节的StrataFlash PROM只用到低24位，SF\_A<23:0>，SF\_A<24>用来提供给更大容量的StrataFlash PROM。

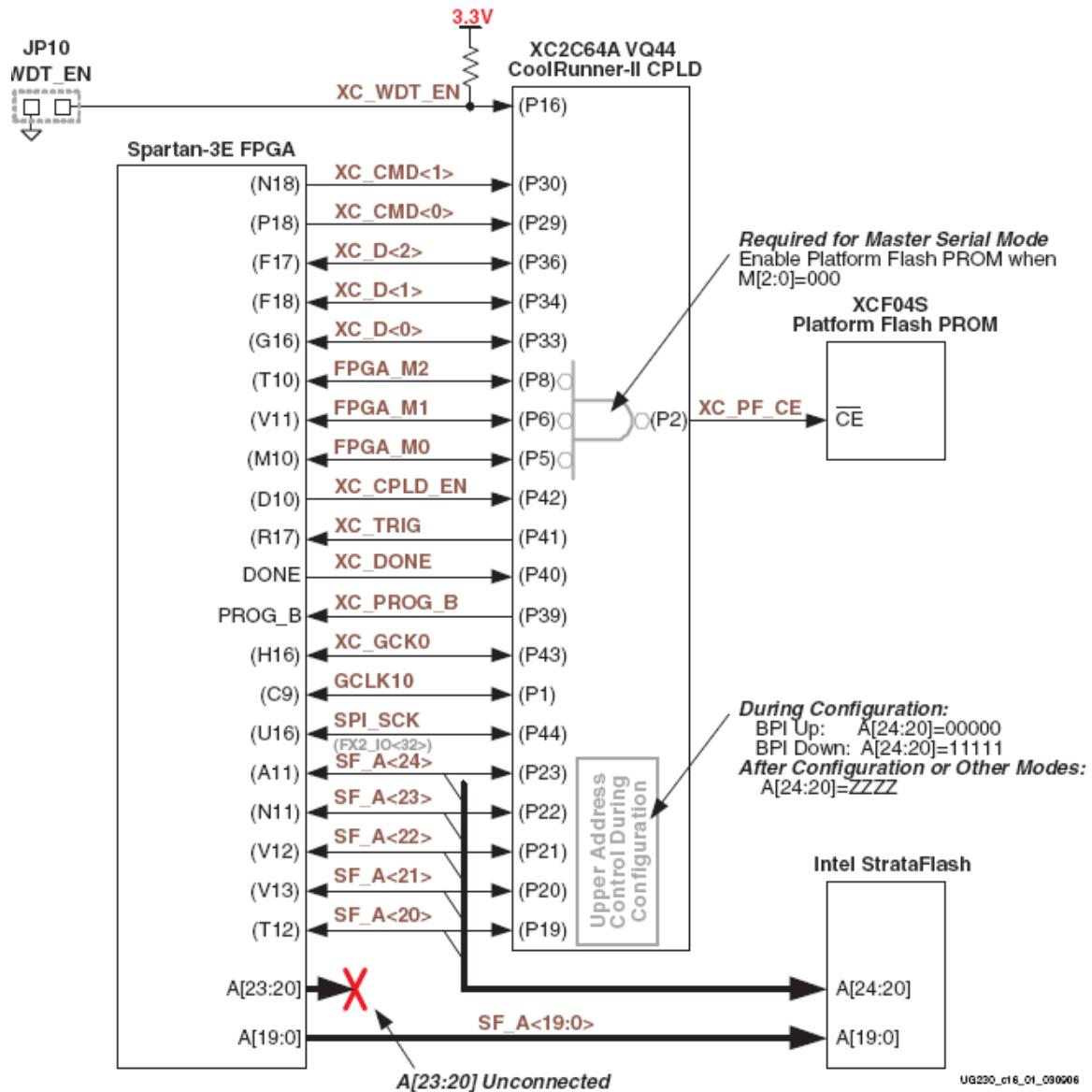


Figure 16-1: XC2C64A CoolRunner-II CPLD Controls Master Serial and BPI Configuration Modes

## UCF Location Constraints

### FPGA Connections to CPLD

```

NET "XC_CMD<1>" LOC = "N18" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET "XC_CMD<0>" LOC = "P18" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET "XC_D<2>" LOC = "F17" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET "XC_D<1>" LOC = "F18" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET "XC_D<0>" LOC = "G16" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET "FPGA_M2" LOC = "T10" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET "FPGA_M1" LOC = "V11" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET "FPGA_M0" LOC = "M10" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET "XC_CPLD_EN" LOC = "B10" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET "XC_TRIG" LOC = "R17" | IOSTANDARD = LVCMOS33 ;
NET "XC_GCK0" LOC = "H16" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET "GCLK10" LOC = "C9" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET "SPI_SCK" LOC = "U16" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
# SF_A<24> is the same as FX2_IO<32>
NET "SF_A<24>" LOC = "A11" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET "SF_A<23>" LOC = "N11" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET "SF_A<22>" LOC = "V12" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET "SF_A<21>" LOC = "V13" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
NET "SF_A<20>" LOC = "T12" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;

```

**Figure 16-2: UCF Location Constraints for FPGA Connections to CPLD CPLD**

Figure 16-3 provides the UCF constraints for the CPLD , including the I/O pin assignment and the I/O standard used.

```

NET "XC_WDT_EN" LOC = "P16" | IOSTANDARD = LVCMOS33 ;
NET "XC_CMD<1>" LOC = "P30" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "XC_CMD<0>" LOC = "P29" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "XC_D<2>" LOC = "P36" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "XC_D<1>" LOC = "P34" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "XC_D<0>" LOC = "P33" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "FPGA_M2" LOC = "P8" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "FPGA_M1" LOC = "P6" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "FPGA_M0" LOC = "P5" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "XC_CPLD_EN" LOC = "P42" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "XC_TRIG" LOC = "P41" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "XC_DONE" LOC = "P40" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "XC_PROG_B" LOC = "P39" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "XC_GCK0" LOC = "P43" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "GCLK10" LOC = "P1" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "SPI_SCK" LOC = "P44" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
# SF_A<24> is the same as FX2_IO<32>
NET "SF_A<24>" LOC = "P23" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "SF_A<23>" LOC = "P22" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "SF_A<22>" LOC = "P21" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "SF_A<21>" LOC = "P20" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "SF_A<20>" LOC = "P19" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;

```

**Figure 16-3: UCF Location Constraints for the XC2C64A CPLD**

# Chapter 17

## DS2432 1-Wire SHA-1 EEPROM

入门实验板包括一个集成了 SHA-1 的 Maxim DS2432 串行 EEPROM。图 17.1 所示，DS2432 EEPROM 使用 Maxim 的 1-Wire 接口，顾名思义，它是用单线进行电源和串行通信的。[\(1-Wire/SHA-1 器件是移动式令牌，可以基于随机质询使用加密性能完善的散列算法生成信息认证代码\(MAC\)\)](#)

DS2432 EEPROM 提供一个保护 FPGA 配置位流的方法（当然这种保护方法有很多），使得要复制其信息有困难。根据 Xilinx 应用笔记 XAPP780，提供了一个可行的方案。

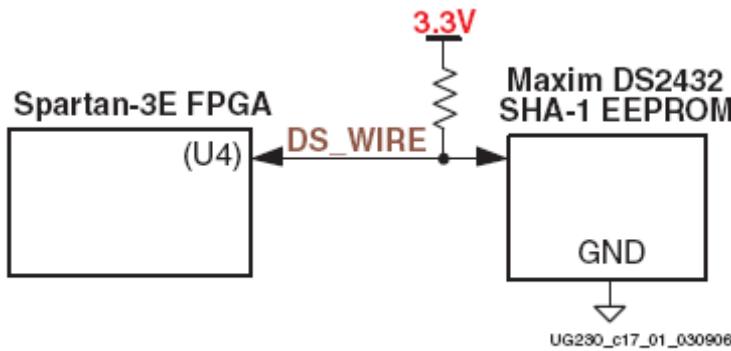


Figure 17-1: SHA-1 EEPROM  
UCF Location Constraints

```
NET "DS_WIRE" LOC = "U4" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8 ;
```

Figure 17-2: UCF Location Constraints for DS2432 SHA-1 EEPROM