



XAPP445 (v1.4) 2007 年 9 月 17 日

## 用 SPI Flash 存储器配置 Spartan-3E FPGA

作者：Arthur Khu

### 提要

本应用指南讲述 Spartan-3E 系列中的串行外设接口 (SPI) 配置模式。SPI 配置模式拓宽了 Spartan™-3E 设计人员可以使用的配置解决方案。SPI Flash 存储器件引脚少、封装外形小而且货源广泛。本指南讨论用 SPI Flash 存储器件配置 Spartan-3E FPGA 所需的连接，并且介绍 SPI 模式的配置流程。本指南还提供一种实用工具，用于在原型开发过程中对选定的 STMicroelectronics 和 Atmel SPI 器件进行在系统编程。

### 简介

Spartan-3E 系列继续支持常用的串行和从并配置模式，这类模式直接连接 Xilinx PROM。对于具有 JTAG 测试功能或者需要 FPGA 配置和 PROM 编程共用一个 JTAG 接口的系统，使用 Xilinx Platform Flash PROM 系列是一种便利的解决方案。

除了标准的串行和并行配置模式，Spartan-3E FPGA 还包括两种新配置模式：串行外设接口 (SPI) 和字节宽度外设接口 (BPI)。对于已经具有板上 SPI 或行业标准并行 NOR Flash 存储器件的系统，好处是可以选择从同一存储器件配置 FPGA。Xilinx 不提供或支持 SPI 和并行 NOR Flash 存储器件，但可以从第三方供应商购买这些器件。

本应用指南主要介绍 Spartan-3E 系列的新型 SPI 配置接口。基于 SPI 的 Flash 存储器得以流行，是因为它们像 Xilinx Platform Flash PROM 一样可以在配置后方便地访问，从而提供对 FPGA 的随机存取和非易失性数据存储。各供应商使用的 SPI 协议稍有不同。本指南讨论 SPI 基本原理和供应商差异。另外，本指南还讨论在 SPI 模式下配置 Spartan-3E FPGA 所需的连接和 SPI 编程方法。有关 SPI 配置模式的进一步详情，请参阅 [DS312](#) 《Spartan-3E FPGA 系列数据手册》。

本应用指南涉及的软件编程实用工具称为 XSPI。设计人员可以用此实用工具对选定的 STMicroelectronics 和 Atmel SPI Flash 存储器件进行在系统编程、擦除和验证操作。XSPI 使用 Xilinx 并行或 USB 下载线和个人主机对 SPI 存储器件执行所需操作。在使用 XSPI 时，必须将 SPI 引脚引出到电路板接口并如 [图 7](#) 所示连接到 Xilinx 并行下载线。XSPI 是免费的实用工具，仅供原型开发之用，不适于生产编程。第三方供应商提供对商品存储器的生产编程支持。有关生产编程器的货源信息，请与 SPI 供应商联系。

### SPI 基本原理

串行外设接口 (SPI) 是一种四线同步串行数据总线。这种数据链路最初是用作微控制器与外设之间的串行通讯接口。许多现代微控制器都包括内置的 SPI 接口，以用于支持与众多存储器、传感器、显示驱动器以及模数和数模转换器通讯。这种接口常见于嵌入式和消费市场，现在也可用来配置 Spartan-3E FPGA。

SPI 系统通常由一个主器件和至少一个从器件组成。对于 Spartan-3E 配置，FPGA 是 SPI 主器件，SPI Flash PROM 是从器件。SPI 接口使用 [表 1](#) 所列四种信号在主从器件之间进行通讯。

表 1: 主从器件的通用 SPI 信号

通用 SPI 信号	信号描述
SCLK	“串行时钟” 提供串行接口的时序。
MOSI	“主出从入” 由主器件用来指定要执行的指令或者向从器件传送数据。
MISO	“主入从出” 由主器件用来收集从器件传输的数据。
SS_n	“从器件选择”，低有效信号；当置为高时，此信号用来反选从器件并将 MISO 设置为高阻抗。

虽然所支持的供应商特有命令可以变化，但通用 SPI 物理接口是标准的全双工传输。SPI 主器件通过 SCLK 时钟信号控制所有时序。在选择信号 SS\_n 转为低之后，主器件经 MOSI 引脚控制将数据由主器件同步输出到从器件。在同一时钟周期内，数据经 MISO 信号控制由从器件同步输出到主器件。数据在一个时钟沿上从各器件同步输出，在该周期内的下一个对沿 (opposite edge) 上同步输入该器件。图 1 所示为主从器件的典型布局。

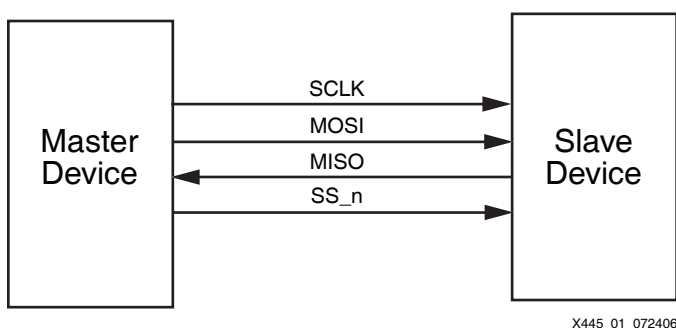


图 1: 典型 SPI 主从器件的结构

在图 2 中，SPI 主器件使用选择信号 SS\_n 选择从器件。然后，主器件使用单向 MOSI 输出将串行命令发送到从器件。从器件收到串行命令后将数据返回到主器件的单向 MISO 输入。

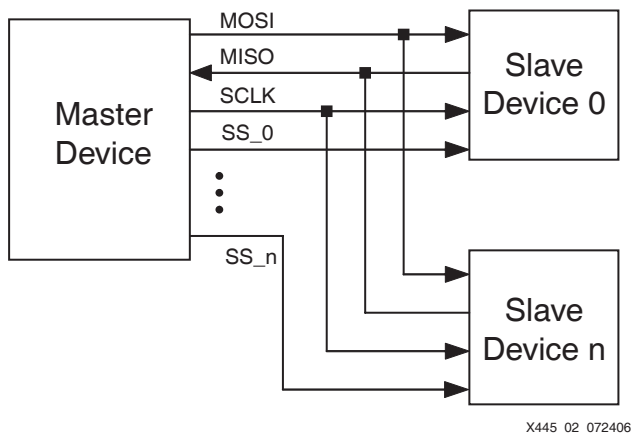


图 2: SPI 主器件与多个从器件的结构

默认情况下，Spartan-3E FPGA 有一个从器件选择信号，称为 CSO\_B。更多的 SPI 从器件需要自带选择信号，往往需要附加一个 FPGA I/O 引脚。MOSI、MISO 和 SCLK 信号由公共 SPI 总线上的所有器件共享。

## SPI Flash 存储器

SPI 物理接口是公共接口，但各 SPI Flash 供应商使用的信号命名约定略有不同。以下表 2 所列为 SPI Flash 存储器件与 FPGA SPI 配置的信号相互对照。除了通用 SPI 引脚（MOSI、MISO、SCLK 和 SS\_n），许多 SPI Flash 存储器供应商还采用其他 SPI 控制信号。FPGA 在配置中不使用 SPI 供应商特有控制信号，但在编程和配置过程中必须适当保留这些控制信号。有关进一步详情，请见相应的 SPI Flash 存储器数据手册。

表 2: SPI Flash 接口连接与引脚命名

SPI 信号	SPI Flash 引脚 <sup>(1)</sup>	Spartan-3E 连接
<b>通用 SPI 信号</b>		
MOSI	DATA_IN (D/SI)	MOSI
MISO	DATA_OUT (Q/SO)	DIN
SS_n	SELECT (S/CS)	CSO_B
SCLK	CLOCK (C/SCK)	CCLK
<b>其他供应商特有 SPI 控制信号</b>		
写保护	$\overline{\text{WR\_PROTECT}}$ ( $\overline{\text{W/WP}}$ )	FPGA 配置不需要此信号，但必须将其保持为高才能对 SPI Flash 进行编程或擦除。配置后可选择连接到 FPGA 的用户 I/O。
保持	$\overline{\text{HOLD}}$	FPGA 配置不需要此信号，但必须将其保持为高才能对 SPI Flash 进行编程或擦除。配置后可选择连接到 FPGA 的用户 I/O。
复位	$\overline{\text{RESET}}$	FPGA 配置不需要此信号，但在 FPGA 配置和 SPI 擦除或编程过程中必须将其保持为高。配置后可选择连接到 FPGA 的用户 I/O。请勿连接到 FPGA 的 PROG_B，因为这样做会导致与 SPI 编程冲突。
就绪 / 繁忙	$\text{RDY}/\overline{\text{BUSY}}$	仅某些封装具有此信号。FPGA 配置不需要此信号。配置后可选择连接到 FPGA 的用户 I/O。

### 注:

1. 此表所列为通用 SPI Flash 引脚名称与最常见供应商引脚名称的对照。各供应商所用 SPI 控制信号的子集可能不同。有关具体引脚的信息，请参见供应商的数据手册。

除了已讨论的 SPI 控制信号差异，SPI 命令也是供应商特有命令，不是通用命令。考虑到这种差异，Spartan-3E FPGA 具有三个变量选择引脚，用于确定供应商的操作以及定义从 SPI Flash 存储器读取数据所需的特定命令序列。VS[2:0] 引脚确定 FPGA 发出哪条 SPI Flash 命令来启动读操作以及在 FPGA 预期收到来自 SPI Flash 的有效数据之前需要插入的虚拟字节数。表 3 所示为 SPI Flash 存储器件所需的变量选择。

表 3: Spartan-3E 变量选择代码

VS[2:0]	SPI Flash 命令 <sup>(1)</sup>	地址长度	虚拟字节数 <sup>(1)</sup>
<1:1:1>	FAST READ (0x0B)	24 位	1 字节
<1:0:1>	READ (0x03)	24 位	0 字节
<1:1:0>	READ ARRAY (0xE8)	24 位	4 字节

### 注:

1. 各变量之间的 SPI Flash 命令和虚拟字节数不同。有关支持的读命令，请参阅 SPI Flash 存储器数据手册。

商品 SPI Flash PROM 的密度从 512 Kb 到 128 Mb 不等。表 4 所示为配置一个 Spartan-3E FPGA 所需的最小存储器密度。FPGA 比特流压缩使用 `bitgen -g` 压缩选项，可以将设计压缩到较小的 SPI 器件中。

表 4: Spartan-3E 典型配置的数据位要求

器件	配置数据位 (每器件)	所需最小 SPI Flash	保留空间
XC3S100E	581,344	1,024 Kb (1 Mb)	456 Kb
XC3S250E	1,353,728	2,048 Kb (2 Mb)	726 Kb
XC3S500E	2,270,208	4,096 Kb (4 Mb)	1.83 Mb
XC3S1200E	3,837,184	4,096 Kb (4 Mb)	348 Kb
XC3S1600E	5,969,696	8,192 Kb (8 Mb)	2.31 Mb

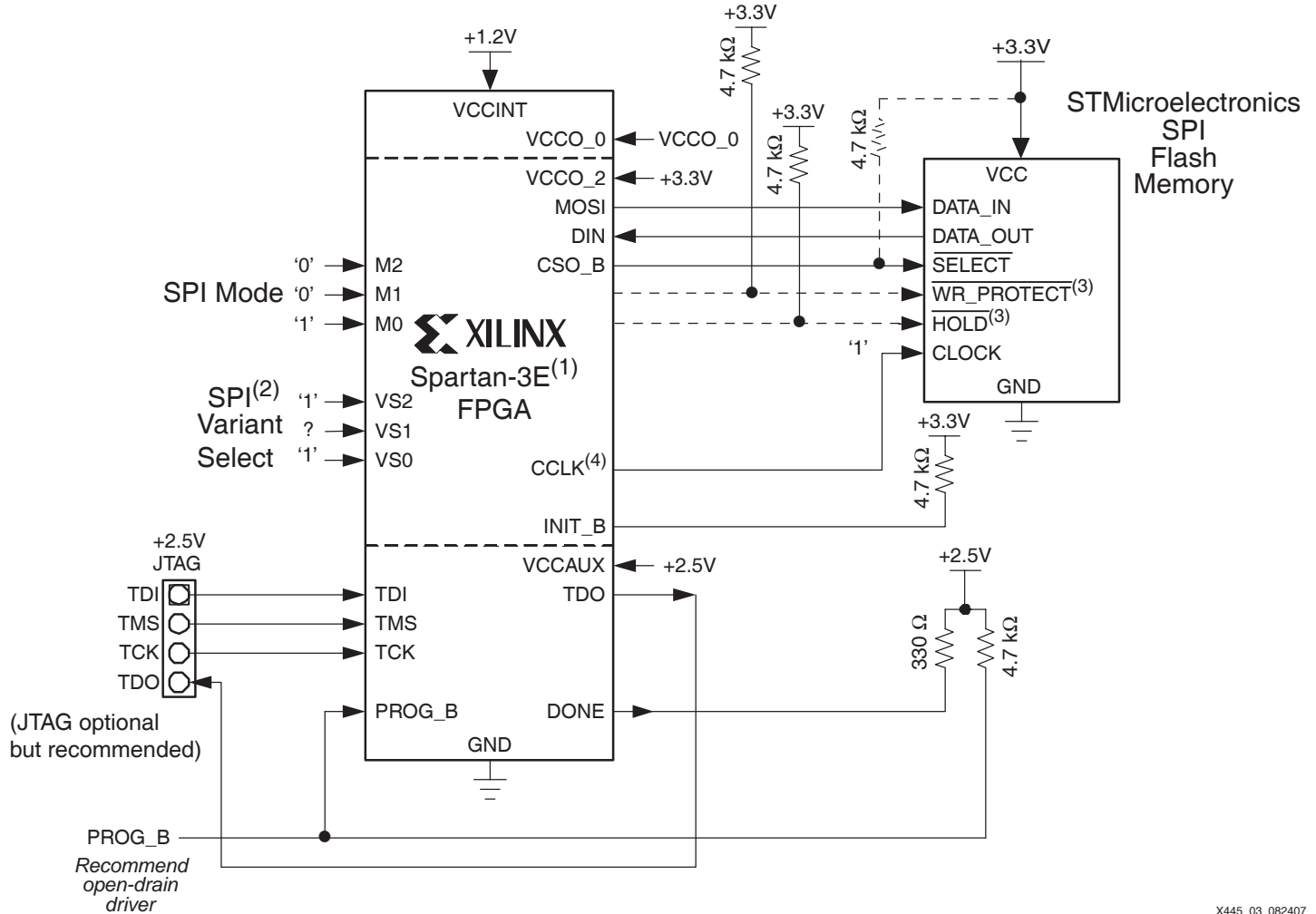
对于存储多个 FPGA 配置比特流的菊花链应用或存储其他用户数据（如嵌入式 MicroBlaze™ RISC 处理器核的代码）的应用，可以使用较大的 SPI 存储器件。只有 Stepping 1 及更新的 Spartan-3E 器件支持通过单 SPI Flash PROM 对多个 Spartan-3E FPGA 进行菊花链连接。

## 从 SPI Flash 存储器配置 Spartan-3E

如图 3 所示，在 SPI 串行 Flash 模式 ( $M[2:0] = 001$ ) 下，Spartan-3E FPGA 从 SPI 串行 Flash 存储器启动自我配置。在这种模式下，FPGA 用其内部晶振为所连接 STMicroelectronics SPI Flash 器件的时钟输入提供 CCLK 输出。

图 3 所示为支持 0x03 READ 命令或 0x0B FAST READ 命令的 SPI Flash PROM 的总体连接图。

图 4 所示为 Atmel DataFlash 串行 PROM 的连接图，这些 PROM 也使用基于 SPI 的协议。Atmel DataFlash 器件支持 READ ARRAY (0xE8) 命令。

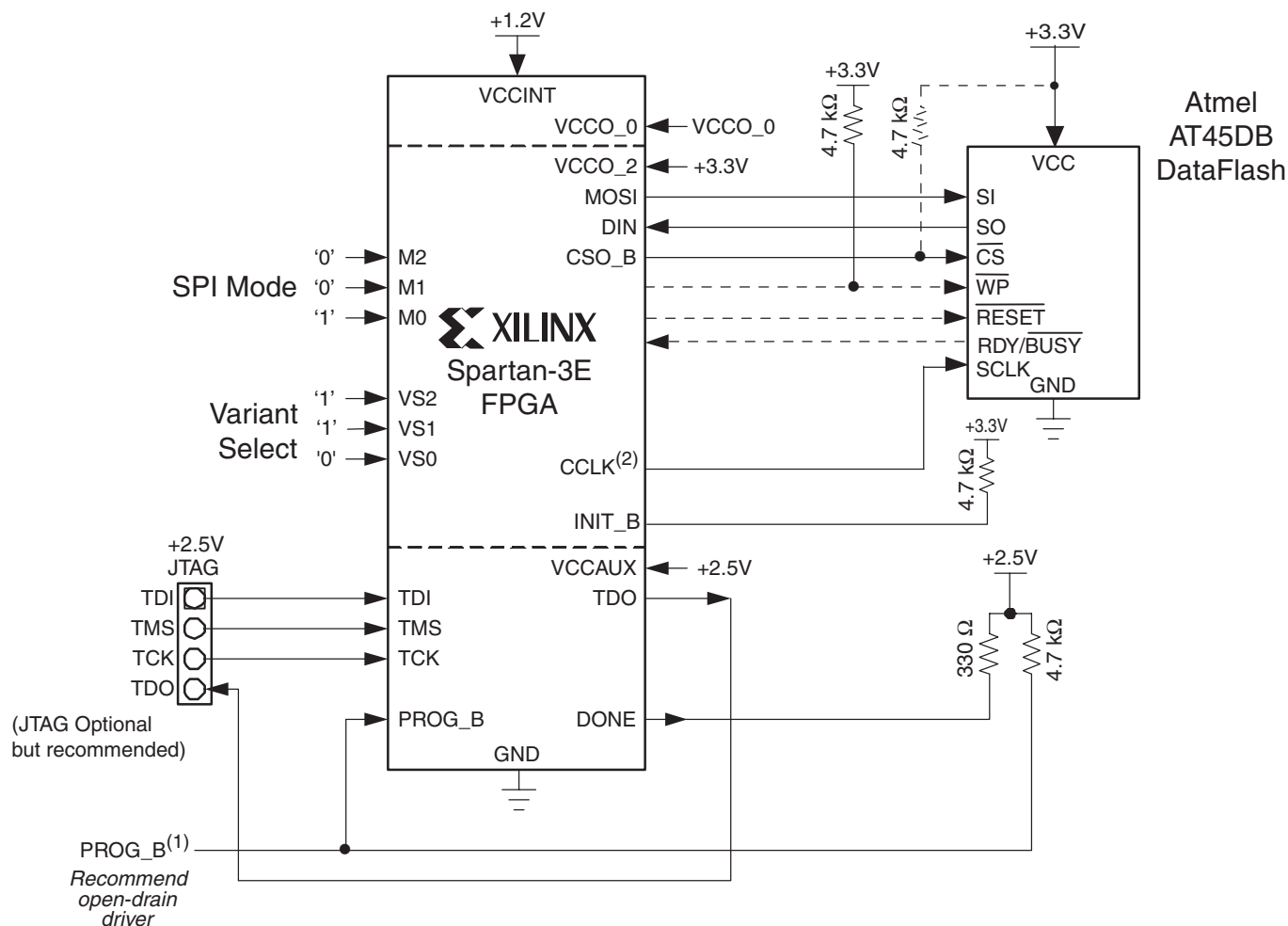


X445\_03\_082407

**注:**

1. 有关上电监视器考虑事项，请参阅《Spartan-3E FPGA 系列数据手册》。
2. 快读命令的变量选择是 111，读命令的变量选择是 101。
3. SPI 控制信号是供应商特有信号；有关进一步详情，请参阅 SPI 数据手册。
4. 有关确保良好 CCLK 信号完整性的设计建议，请参阅 [UG332](#) 《Spartan-3 系列配置用户指南》中的“CCLK 设计考虑事项”部分。

图 3: STMicroelectronics SPI 与 Spartan-3E FPGA 之间的快读命令 (0x0B) 和读命令 (0x03) 接口



X445\_04\_082407

- 注:**
1. 有关上电监视器考虑事项，请参阅《Spartan-3E FPGA 系列数据手册》。
  2. 有关确保良好 CCLK 信号完整性的设计建议，请参阅《Spartan-3 系列配置用户指南》中的“CCLK 设计考虑事项”部分。

图 4: Atmel DataFlash SPI 与 Spartan-3E FPGA 之间的读命令 (0xE8) 接口

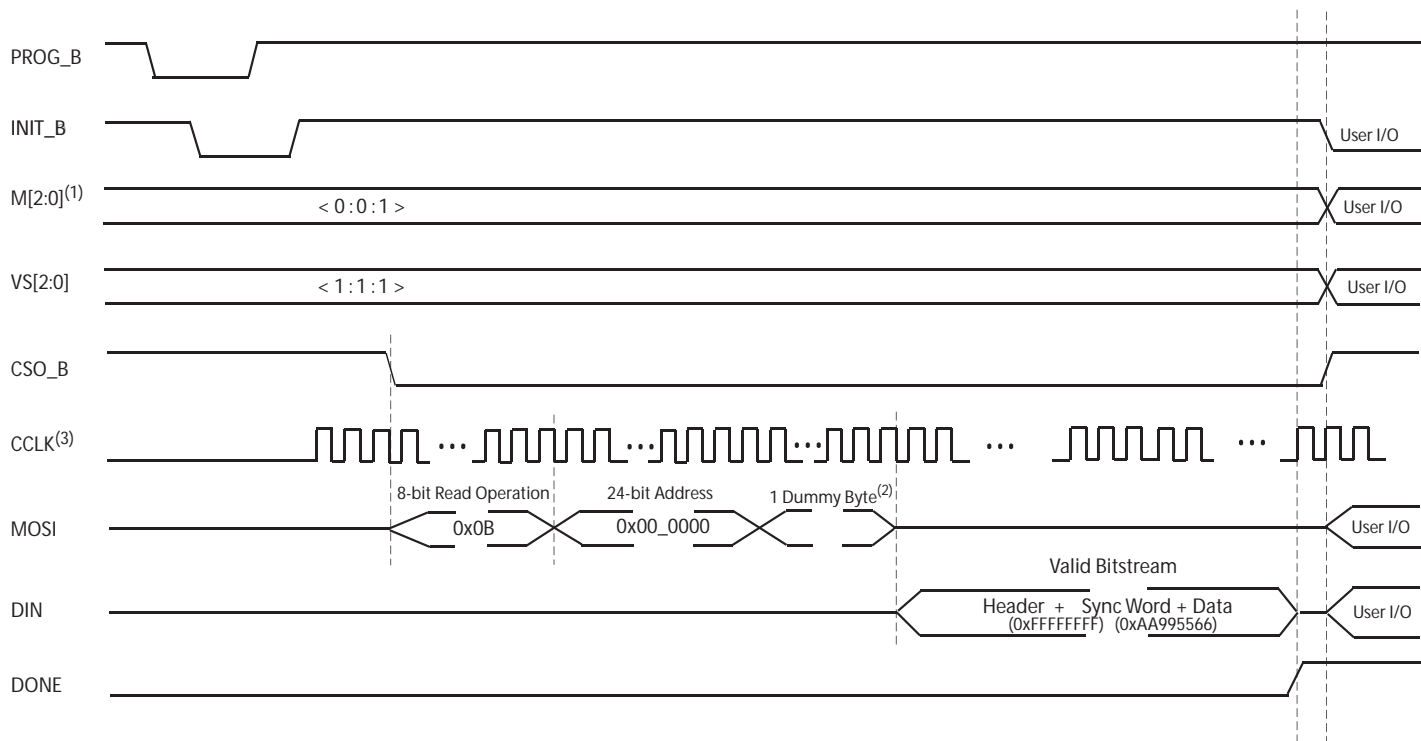
### SPI 配置序列

图 5 所示为 Spartan-3E SPI 配置流程概况。在上电或者向 PROG\_B 引脚发出低脉冲后，Spartan-3E FPGA 经过一个初始化序列，以清空内部 FPGA 配置存储器。在此序列开始处，DONE 和 INIT\_B 引脚均转为低。初始化完成后，INIT\_B 引脚转为高并采样 Spartan-3E 模式及变量选择引脚。

在 SPI 模式下，FPGA 对变量选择 (VS[2:0]) 引脚采样，以确定发出哪个 SPI 命令序列。当初始化之后发出 INIT\_B 信号时，模式引脚和变量选择引脚都必须处在正确的逻辑级，以确保正确采样。

在变量选择引脚选择 SPI 命令集之后，Spartan-3E FPGA 将 CSO\_B 选择信号置为低，并且开始通过 FPGA 的 CCLK 引脚对 SPI Flash 存储器进行时钟控制。接着，FPGA 发出 8 位读命令后跟 24 位起始地址 0x00\_0000 和目标命令集的适量虚拟字节。FPGA 从地址 0 开始读取 SPI Flash 存储器阵列，直到读完所需的配置位数。如果从存储器件读取到有效比特流，则发出 DONE 信号，以指示 Spartan-3E FPGA 配置成功。配置成功之后，所有 Spartan-3E SPI 引脚都成为可用的用户 I/O 引脚。

表 5 列出了配置过程中使用的 Spartan-3E 信号。如果将 FPGA 配置成 SPI 模式，则必须按表中所指定方式连接这些信号才能配置成功。



X445\_07\_032107

**注:**

1. VS[2:0] 设置可以因所用 SPI 器件供应商不同而变化。请按表 3 选择。
2. 所示为默认的起动顺序。
3. 发出的 MOSI 虚拟字节数因变量选择 (VS[2:0]) 选项不同而变化。请参考表 3。

图 5: FAST READ (0x0B) 的 Spartan-3E SPI 配置流程示例

表 5: Spartan-3E SPI 配置连接

引脚名称	FPGA 方向	描述	配置期间	配置之后
HSWAP	输入	<b>用户 I/O 上拉控制。</b> 如果配置期间为低，则启用所有 I/O 引脚中的上拉电阻上拉到相应 I/O 组的 V <sub>CC0</sub> 输入。 0: 在配置期间上拉。 1: 无上拉。	整个配置过程中，在某个有效的逻辑电平提供驱动。	用户 I/O
M[2:0]	输入	<b>模式选择。</b> 选择 FPGA 配置模式。	M2=0, M1=0, M0=1。在 INIT_B 上升沿采样。	用户 I/O
VS[2:0]	输入	<b>变量选择。</b> 指示 FPGA 如何与连接的 SPI Flash PROM 通讯。	表 3 所示为有效设置选项。如果在 INIT_B 上升沿采样，则必须处于有效设置。	用户 I/O
MOSI	输出	<b>串行数据输出。</b>	FPGA 将 SPI Flash 存储器读命令和起始地址发至 PROM 的串行数据输入。	用户 I/O
DIN	输入	<b>串行数据输入。</b>	FPGA 从 PROM 的串行数据输出接收串行数据。	用户 I/O

表 5: Spartan-3E SPI 配置连接 (续表)

引脚名称	FPGA 方向	描述	配置期间	配置之后
CSO_B	输出	芯片选择输出。低有效。	连接到 SPI Flash PROM 的芯片选择输入。如果 HSWAP=1, 则将此信号连接到 4.7kΩ 的上拉电阻上拉到 3.3V。	配置后将 CSO_B 置为高, 以禁用 SPI Flash 并将 MOSI、DIN 和 CCLK 引脚恢复为用户 I/O。可以选择重新使用此引脚以及令 MOSI、DIN 和 CCLK 引脚继续与 SPI Flash 通讯。
CCLK	输出	<b>配置时钟。</b> 由 FPGA 内部晶振生成。频率由 <i>ConfigRate</i> 比特流发生器选项控制。如果 CCLK 的 PCB 迹线较长或具有多个连接, 则端接此输出, 以保证信号完整性。	驱动 PROM 的时钟输入。	用户 I/O
DOUT	输出	<b>串行数据输出。</b>	有效时作驱动使用。在单 FPGA 配置中不使用。在菊花链配置中, 此引脚连接到链中下一 FPGA 的 DIN 输入。下游 FPGA 为从串模式。	用户 I/O
INIT_B	开漏双向 I/O	<b>初始化指示器。</b> 低有效。配置过程开始时置为低, 并在存储器清空过程保持为低。在存储器清空结束时释放, 此时对模式和变量选择引脚采样。在菊花链应用中, 此信号需要用 4.7kΩ 的外部上拉电阻上拉到 V <sub>CCO_2</sub> 。	在配置期间有效。如果 SPI Flash PROM 需要上电后有 > 2 ms 的唤醒时间, 则将 INIT_B 保持为低, 直到 PROM 就绪。如果在配置期间检测到 CRC 错误, 则 FPGA 将 INIT_B 置为低。	用户 I/O。如果在应用中不使用, 则将 INIT_B 置为高。
DONE	开漏双向 I/O	<b>FPGA 配置完成。</b> 在配置期间为低。在 FPGA 成功完成配置后转为高。需要用 330Ω 的外部上拉电阻上拉到 2.5V。	低表示 FPGA 尚未完成配置。	通过外部上拉电阻上拉到高。当为高时, 表示 FPGA 已成功完成配置。
PROG_B	输入	<b>启动 FPGA 编程。</b> 低有效。如果保持低有效达到或超过 300 ns, 则强制 FPGA 通过清空配置存储器并重新设置 DONE 和 INIT_B 引脚来重新启动配置过程。需要用 4.7kΩ 的外部上拉电阻上拉到 2.5V。如果从外部驱动, 则使用开漏或开集驱动器。	必须为高才能开始配置。	置低然后释放 PROG_B, 以便为 FPGA 重新编程。通过保持 PROG_B 为低以强制 FPGA I/O 引脚进入 Hi-Z 状态, 可以直接对 SPI Flash PROM 引脚编程。



## 为 SPI 存储器编程

为 SPI 存储器编程有三种主要方法：

- 第三方编程器（板外编程）
- 间接在系统编程
- 直接在系统编程

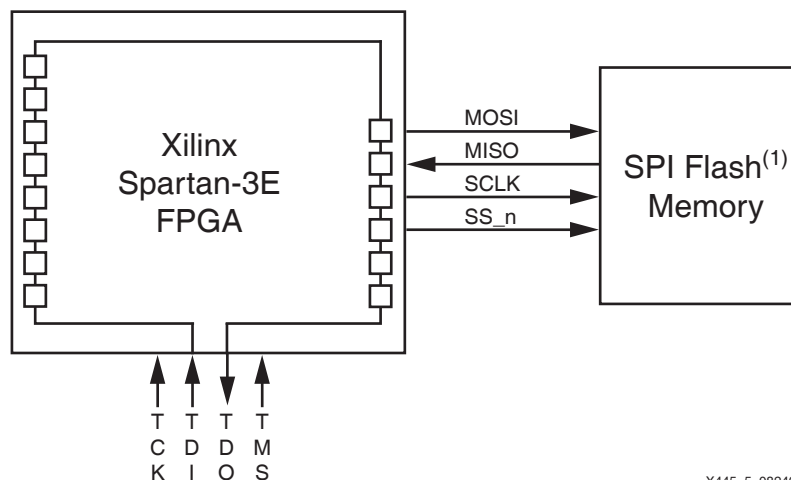
### 第三方编程器（板外编程）

可以用第三方编程器对 SPI 存储器进行板外编程。这种方法的优点是可以进行大量生产编程。其缺点是难以为 PCB 上的器件重新编程，除非设置 FPGA 或 SPI 总线上的其他某个控制器来执行这一操作。有些分销商提供批量器件编程服务。可以向 SPI PROM 供应商索取第三方编程供应商清单。有关编程文件的正确格式，请见第 11 页“编写 SPI PROM 文件”。

### 间接在系统编程

FPGA 具有 JTAG 测试功能，其中包括标准的 PRELOAD 和 EXTEST 命令。在使用这些命令时，可以用 JTAG 链驱动 FPGA 的引脚并对其采样，从而通过 PCB 上的迹线对 SPI 存储器的引脚发出编程指令序列。这种方法如图 6 所示。许多第三方 JTAG 工具供应商都支持这种方法，但此方法常常比其他编程方法慢得多。可以使用能理解 JTAG 协议并且能将实用数据转换成 SPI 总线能接收的数据的工具，通过 FPGA 的 JTAG 链为 SPI 存储器编程。

这种方法的优点是很少布线即可进行在系统编程。其缺点是 SPI 编程速度相对较低，并且需要支持所用特定 SPI 器件的第三方 JTAG 编程器。



注：

1. 在编程过程中，SPI Flash 存储器信号  $\overline{\text{RESET}}$ 、 $\overline{\text{W}}$  和  $\overline{\text{HOLD}}$  必须保持为高。

图 6: 用 JTAG EXTEST 命令为 SPI Flash 编程

### 直接在系统编程

对于需要在系统编程支持的系统，有适合生产和原型设计阶段的不同选项。对于生产编程，有些第三方 PROM 编程器利用带连线的插座转接器对 SPI flash 存储器进行在系统编程。对于原型编程，Xilinx 有一个称为 XSPI 的软件实用工具，可用于有限 SPI flash 存储器集的编程。XSPI 实用工具使用 Xilinx 并行下载线或 USB 下载线和主机，对 SPI Flash 存储器件直接进行在系统编程。

无论是使用 XSPI 实用工具还是使用第三方编程器进行直接在系统编程，都要确保驱动 MOSI、MISO、SCLK 和 SS\_n 信号的 FPGA 引脚均为高阻抗（浮动，Hi-Z）。将 FPGA SPI 信号置为高阻抗有三种方法：

- 保持 FPGA 的 PROG\_B 引脚为低，从而使所有 I/O 引脚为三态引脚。
- 也可以将 FPGA 的模式引脚改为 JTAG 模式 ( $M[2:0] = <1:0:1>$ )，并且向 PROG\_B 发送脉冲，从而强制所有 FPGA I/O 为高阻抗。
- 还可以从正常工作的 FPGA 应用内部引出控制信号，在使用 XSPI 对所连接 SPI Flash 编程时，用此信号使 MOSI、MISO、SCLK 和 SS\_n 信号为三态信号。

FPGA 的 HSWAP 引脚必须为低才能启用所有用户 I/O 引脚上的上拉电阻；此引脚必须为高才能禁用这些上拉电阻。

## XSPI 实用工具

Xilinx 提供一个称为 XSPI 的命令行实用工具，用来为 STMicroelectronics M25Pxx 和 M45PExx 以及 Atmel AT45DBxx 和 AT25Fxx SPI Flash 存储器件编程。可下载的文件 Xapp445.zip 中提供了 XSPI 实用工具的两个版本，其中包含 install\_xspi.zip 和 install\_xspi\_usb.zip 两个文件。文件 install\_xspi.zip 包含实用工具版本 **xspi.exe**，此版本使用并行下载线 Cable IV 或并行下载线 Cable III。文件 install\_xspi\_usb.zip 包含实用工具版本 **xspi\_usb.exe**，此版本使用平台下载线 USB 和并行下载线 Cable IV。文件 install\_xspi.zip 和 install\_xspi\_usb.zip 都包含数据库文件 .xdv，XSPI 实用工具的两个版本都引用此数据库文件。

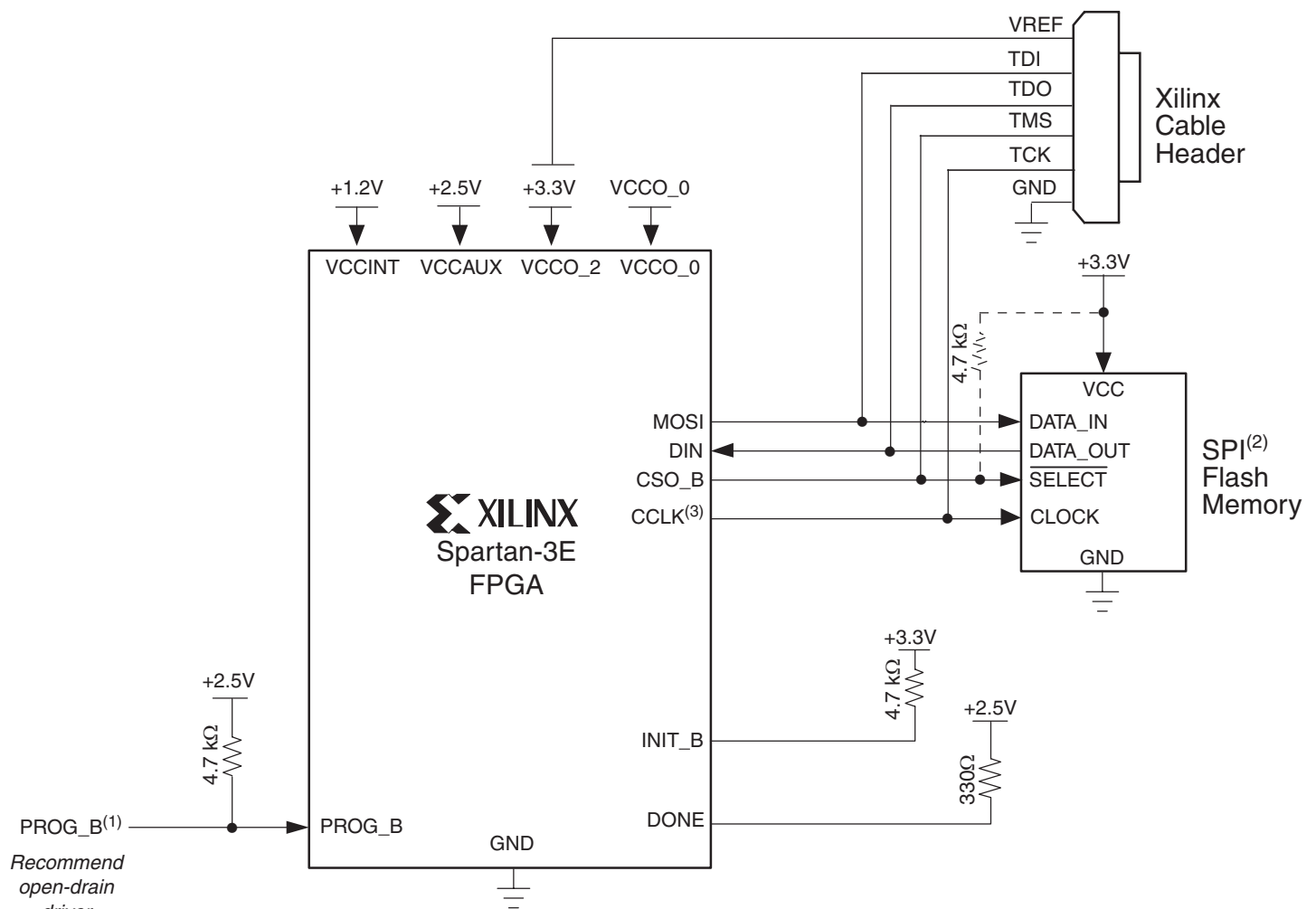
**注：** 启动 XSPI 实用工具任一版本的 PC 上必须安装 iMPACT。USB 版 XSPI 使用 iMPACT 的下载线驱动程序。如果使用 iMPACT 9.1i (或更新版本)，请使用 **xspi\_usb.exe** 版本 3.11b (或更新版本)，以便驱动程序兼容。默认情况下，**xspi\_usb.exe** 连接 USB 下载线。有关下载线选择选项 (-select\_cable N)，请通过键入 **xspi\_usb** 参阅 XSPI 命令。

XSPI 是 WinXP/2000 版命令行实用工具，该工具与 SPI 存储器件之间使用 Xilinx 并行下载线 Cable III、并行下载线 Cable IV 或下载线 USB 传递往来信号。这些下载线上的四个 JTAG 引脚连接到下述 SPI 引脚：

表 6: XSPI 的 Xilinx 并行下载线和 USB 下载线与 SPI Flash 存储器引脚的连接

Xilinx 下载线信号	SPI Flash 引脚
TCK	时钟 (C/SCK)
TMS	选择 (S/CS)
TDI	数据输入 (D/SI)
TDO	数据输出 (Q/SO)

当对 SPI Flash 存储器件进行在系统编程时，要保证在连接 Xilinx 下载线时 SPI 引脚上没有冲突。例如，在运行 XSPI 对 SPI flash 器件编程之前，Spartan-3E PROG\_B 引脚必须保持为低，以使 Spartan-3E SPI 接口信号为三态信号。另外，任何其他 SPI Flash 存储器控制信号（如 WRITE PROTECT、HOLD 或 RESET）都必须在电路板上保持为无效状态才能保证 SPI 编程正确。图 7 所示为编程接入点。



X445\_6\_082407

**注:**

1. 在 SPI 存储器编程期间，PROG\_B 必须置为低。
2. 在 SPI 存储器编程期间，SPI 供应商特有控制信号（如 RESET、HOLD 和 W/WP）必须保持为高。
3. 有关确保良好 CCLK 信号完整性的设计建议，请参阅《Spartan-3 系列配置用户指南》中的“CCLK 设计考虑事项”部分。

图 7: 用 XSPI 为 SPI Flash 编程的 Xilinx 下载线连接

**编写 SPI PROM 文件**

本部分提供为 SPI Flash 存储器创建 PROM 文件的指导原则。在将 Spartan-3E 比特流转换成 SPI 格式 PROM 文件之前，设计人员必须确认该比特流是用 **bitgen -g StartupClk:Cclk** 选项生成的。此选项使启动顺序与 Spartan-3E 内部时钟同步，从而确保 FPGA 功能正常。

Xilinx 软件工具 PROMGen 或 iMPACT 从 Spartan-3E 比特流生成 SPI 格式 PROM 文件。SPI 存储器首先串行输出数据的 MSB 字节，而 Xilinx PROM 则首先输出数据的 LSB。与标准 Xilinx PROM 文件相比，SPI 格式 PROM 文件在每字节内有位反转，因此，需要在 PROMgen 中使用 **-spi** 选项才能正确格式化。

XSPI 支持 **.hex** 和 **.mcs** 两种 SPI PROM 文件格式。以下所示为生成 SPI 格式 **.mcs** 文件的 PROMGen 命令行操作的示例。要生成 SPI 格式 PROM 文件 **.hex**，请用 **-p hex** 替换 **-p mcs** 选项开关：

```
promgen -spi -p mcs -o promdata.mcs -s 512 -u 0 inputfile.bit
```

**注:** 当选择第三方 SPI-PROM 流时，此格式化过程在 iMPACT 中自动完成。

表 7: PROMGen 的 SPI PROM 文件选项示例

PROMGen 选项	描述
-spi	用于保持从 SPI Flash 存储器件配置 Spartan-3E FPGA 所需的正确位顺序。
-p <format>	PROM 输出文件格式。XSPI 支持输出文件格式 .hex (-p hex) 或 .mcs (-p mcs)。第三方编程器可以接受所提及的其他 PROM 格式 (-p tek 或 -p exo)。有关详情, 请参阅第三方编程器的文档。
-s <size>	以千字节为单位指定 PROM 容量。对于此选项, PROM 容量必须是 2 的乘方, 其默认设置是 64 Kb。
-u <address>	从指定的起始地址按上行方向加载 .bit 文件。此选项必须在紧接输入比特流文件之前指定。

## XSPI 命令

安装 XSPI 实用工具之后, 在命令窗口中键入 **xspi** 或 **xspi\_usb**, 即可显示可用命令列表。如果找不到 xspi 命令, 请确保 xspi.exe 或 xspi\_usb.exe 和数据库文件 (.xdv) 是在当前工作目录或 PATH 环境变量中指定的子目录中。

以下示例所示为用 MCS 文件中的数据对 STMicroelectronics M45PE40 进行擦除、编程和验证的 **xspi** 命令组合:

```
xspi -accept_notice -mcs -spi_dev m45pe40 -spi_epv -i promdata.mcs -o result.out
```

表 8 所示为上述常用命令行操作的描述和示例。要了解关于所有可用选项的更多信息, 只要在命令提示符处键入 **xspi** 或 **xspi\_usb** 即可查看完整列表。以下命令对于 xspi 和 xspi\_usb 两版实用工具完全相同, 唯一的区别是各版本所提供的下载线支持。如果运行 **xspi\_usb.exe**, 只要在下表中用 **xspi\_usb** 取代 **xspi** 即可。

表 8: XSPI 命令

XSPI 命令	描述
通知接受	<p><b>xspi -accept_notice</b></p> <p>XSPI 实用工具发一次“嘀”音, 提示用户确认说明“这是在原型开发期间使用的编程工具”的通知。若接受此通知并跳过通知提示, 请添加选项:</p> <p><b>-accept_notice</b></p>
SPI 器件列表	<p><b>xspi -display_partlist</b></p> <p>此 XSPI 命令屏显正式支持的 SPI 存储器件。使用 <b>-spi_dev</b> 命令附带此列表中的器件名称之一, 随同在目标器件上需要执行的操作命令。</p>
检查器件 IDCODE 或密度代码	<p><b>xspi -spi_dev &lt;devicename&gt; -spi_idcheckonly</b></p> <p>显示从指定 SPI 器件读取的器件 ID 代码或密度代码。</p>
擦除	<p><b>xspi -spi_dev &lt;devicename&gt; -spi_e -o result.out</b></p> <p>此命令擦除 SPI 器件的内容。XSPI 实用工具轮询 SPI 器件的状态寄存器, 以确定擦除操作是否成功。如果已经超过最大轮询迭代数, 则说明器件连接不正确或 SPI 引脚上可能有冲突。在重试之前, 请检查 SPI 和 Xilinx 并行下载线引脚连接, 然后尝试读取器件 ID 代码。任何擦除错误都记录在 result.out 文件中。</p>
擦除 / 编程 / 验证	<p><b>xspi -spi_dev &lt;devicename&gt; -mcs -spi_epv -i promdata.mcs -o result.out</b></p> <p>此命令使用轮询擦除 SPI 内容, 然后对 SPI 器件的内容进行编程和验证。任何验证不匹配项都记录在 result.out 文件中。默认情况下, XSPI 认定数据是 HEX 格式。如果输入文件是 HEX 格式, 则去除 -mcs 选项:</p> <p><b>xspi -spi_dev &lt;devicename&gt; -spi_epv -i promdata.hex -o result.out</b></p>

表 8: XSPI 命令 (续表)

XSPI 命令	描述
编程	<b>xspi -spi_dev &lt;devicename&gt; -mcs -spi_p -i promdata.mcs -o result.out</b>
	编程操作将 promdata.mcs 文件的内容编入指定的 SPI 存储器件。如果出现编程错误, 则这些错误记录在 result.out 文件中。默认情况下, XSPI 认定数据是 HEX 格式。如果输入文件是 HEX 格式, 则去除 -mcs 选项: <b>xspi -spi_dev &lt;devicename&gt; -spi_p -i promdata.hex -o result.out</b>
验证	<b>xspi -spi_dev &lt;devicename&gt; -mcs -spi_v -i promdata.mcs -o result.out</b>
	验证命令从存储器位置 0 开始对照 MCS 文件中的数据比较 SPI 器件的内容。验证错误写入 result.out 文件。默认情况下, XSPI 认定数据是 HEX 格式。如果输入文件是 HEX 格式, 则去除 -mcs 选项: <b>xspi -spi_dev &lt;devicename&gt; -spi_v -i promdata.hex -o result.out</b>
编程和验证	<b>xspi -spi_dev &lt;devicename&gt; -mcs -spi_pv -i promdata.mcs -o result.out</b>
	编程和验证操作对 SPI 存储器件进行编程和验证。验证错误写入指定的 result.out 文件。promdata.mcs 文件中的字节存入从字节存储器地址位置 0 开始的连续存储器位置。默认情况下, XSPI 认定数据是 HEX 格式。如果输入文件是 HEX 格式, 则去除 -mcs 选项: <b>xspi -spi_dev &lt;devicename&gt; -spi_pv -i promdata.hex -o result.out</b>
读回	<b>xspi -spi_dev &lt;devicename&gt; -spi_r 1 -o result.out</b>
	读回命令读回 SPI 器件中存储的内容。从器件读回的数据以 mcs 模式 (-spi_r 1) 存储在 result.out 中。使用 -spi_r 0 将内容作为 HEX 格式文件读回。
空白检验	<b>xspi -spi_dev &lt;devicename&gt; -spi_b -o result.out</b>
	输出文件 result.out 包含在器件中检测到的所有非空白字节的地址和值。
Prom 文件 反转字节	<b>xspi -spi_dev &lt;devicename&gt; -mcs -reverse_bytes -spi_pv -i promdata.mcs -o result.out</b>
	XSPI 查找 HEX 或 MCS 输入文件中的 Xilinx FPGA 同步数据, 以确定数据字节是否按照为 SPI 器件编程的正确顺序排列。例如, 如果 PROM 文件生成时没有 PROMGen 中的 -spi 选项, XSPI 就会显示警告信息。使用 -reverse_bytes 选项反转编入 SPI Flash 存储器的文件的数据字节顺序。如果输入文件是 HEX 格式, 则去除 -mcs 选项: <b>xspi -spi_dev &lt;devicename&gt; -reverse_bytes -spi_pv -i promdata.hex -o result.out</b>
跳过同步字 检查	<b>xspi -spi_dev &lt;devicename&gt; -mcs -skip_syncword_check -spi_pv -i promdata.mcs -o result.out</b>
	XSPI 自动检查输入 PROM 文件头信息, 以确保存在有效的 Xilinx FPGA 同步模式。使用此选项禁止 Xilinx FPGA 同步头检查。如果输入文件是 HEX 格式, 则去除 -mcs 选项: <b>xspi -spi_dev &lt;devicename&gt; -skip_syncword_check -spi_pv -i promdata.hex -o result.out</b>
跳过 SPI IDCODE 检查	<b>xspi -spi_dev &lt;devicename&gt; -mcs -spi_epv -skip_idcheck -i promdata.mcs -o result.out</b>
	仅当 -spi_dev partname 中指定的 ST Microelectronics 或 Atmel 器件名称有多个器件版本并且其中较老的版本没有 IDCODE 指令时, 才使用此选项。如果输入文件是 HEX 格式, 则去除 -mcs 选项: <b>xspi -spi_dev &lt;devicename&gt; -spi_epv -skip_idcheck -i promdata.hex -o result.out</b>

以下所示为用 XSPI 实用工具成功执行擦除、编程和验证操作序列的示例（根据实用工具的版本，所示信息可能略有变化）。

**注：**当使用该版实用工具时，请用 `xspi_usb` 取代 XSPI 命令。

```
C:\xspi>xspi -spi_dev m45pe40 -mcs -spi_epv -i promdata.mcs -o result.out
```

```
-----
| ==> Checking SPI database [___xspi_database.xdv]
|       - version [ 001 . 4c ( 2007 January 31 ) ]: OK
|-----
```

```
xspi(tm) Version 1.27
Copyright (c) 2003-2007 Xilinx, Inc. All rights reserved.
Xilinx SPI Programming Utility
```

```
*****
**//=====\\**
**||
**|| NOTICE: XSPI SOFTWARE FOR XILINX PROTOTYPE DEVELOPMENT USE ONLY ||**
**||
**|| SOFTWARE PROVIDED "AS IS". ALL WARRANTIES, EXPRESS OR IMPLIED, ||**
**|| ARE HEREBY DISCLAIMED. SOFTWARE NOT AUTHORIZED FOR USE IN ||**
**|| PRODUCTION ENVIRONMENTS OR FOR USE IN OR WITH LIFE-SUPPORT OR ||**
**|| MISSION-CRITICAL APPLIANCES, SYSTEMS, OR DEVICES. ||**
**||
**|| This software is for use with SPI devices listed in the XSPI ||**
**|| device database (i.e., run XSPI with the command option ||**
**|| -display_partlist). Results when used with SPI devices from ||**
**|| other manufacturers are unknown. Please email all technical ||**
**|| questions and comments to: ||**
**||
**|| xspi@xilinx.com ||**
**||
**|| Refer to Xilinx appnote XAPP445 for software documentation. ||**
**\\=====//**
*****
```

```
==> Use "-accept_notice" option to accept notice automatically
```

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

```
Start : Thu Mar 22 17:49:08 2007
```

```
==> Checking SPI device [STMicro_M45PE40_ver_00100] ID code(s)
- density = [524288] bytes
          = [4194304] bits
- mfg_id = [0x20]
- memory_type = [0x40]
- memory_capacity = [0x13]
```

```
+-----+
| Device ID code(s) check ==> [ OK ] |
+-----+
```

```
=> Operation: Erase
```

```
=> Operation: Program and Verify using file [c:\promdata.mcs]
- device has [256] byte programming page/buffer
Programmed [283776] of [283776] bytes (w/ polling)
Verified [283776] of [283776] bytes (0 errors)
```

```
--> Total byte mismatches [0] (see [result.out])
```

```
Finish : Thu Mar 22 17:49:51 2007
```

```
Elapsed clock time (00:00:43) = 43 seconds
```

## 编程实用工具

请点击[这里](#)下载 XSPI 实用工具。

## 结论

SPI 接口与 Spartan-3E 相结合，允许设计人员使用多家供应商的小型 SPI 存储器进行配置。除了存储现有的用户数据，板上已有 SPI Flash 存储器的系统还可以借助单存储器资源存储配置数据。

## 修订历史

下表说明此技术文档的修订历史。

日期	版本	修订
2005 年 10 月 20 日	1.0	Xilinx 最初版本。
2006 年 1 月 3 日	1.1	修改第 14 页上的链接。增加 Atmel 技术文档。
2006 年 7 月 27 日	1.2	<ul style="list-style-type: none"> <li>• 多项更正和更新。</li> <li>• 更新以包括 USB 下载线支持。</li> <li>• 为简明起见，更新第 7 页图 5。</li> <li>• 第 12 页表 8 中的命令。</li> <li>• 更新第 14 页上的示例。</li> </ul>
2007 年 4 月 5 日	1.3	<ul style="list-style-type: none"> <li>• 增加 <code>xspi_usb.exe</code> 增强型并行下载线支持。</li> </ul>
2007 年 5 月 1 日	1.3.1	<ul style="list-style-type: none"> <li>• 纠正排版错误。</li> <li>• 增加对 ISE 9.1.03i 的 <code>xspi_usb.exe</code> 支持。</li> </ul>
2007 年 9 月 17 日	1.4	<ul style="list-style-type: none"> <li>• 更新第 5 页图 3、第 6 页图 4 和第 11 页图 7，以增加关于 CCLK 上信号完整性的注解。</li> <li>• 增加对 ISE 9.2.x 的 XSPI_USB 支持。</li> </ul>

## Notice of Disclaimer

Xilinx is disclosing this Application Note to you "AS-IS" with no warranty of any kind. This Application Note is one possible implementation of this feature, application, or standard, and is subject to change without further notice from Xilinx. You are responsible for obtaining any rights you may require in connection with your use or implementation of this Application Note. XILINX MAKES NO REPRESENTATIONS OR WARRANTIES, WHETHER EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL XILINX BE LIABLE FOR ANY LOSS OF DATA, LOST PROFITS, OR FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR INDIRECT DAMAGES ARISING FROM YOUR USE OF THIS APPLICATION NOTE.