XAPP126 (v1.2) July 22, 2003

Data Generation and Configuration for Spartan Series FPGAs

Author: Ashok Chotai and Hari Devanath

Summary

This application note describes various methods for configuring Spartan series FPGAs. Each configuration method is described in detail. Information on necessary software programs to run with input files required, output files produced, download cables used, and other hardware necessary to accomplish the task is discussed. This application note targets users who are new to Xilinx devices and the Alliance/Foundation series software tools. It is intended to make the configuration and debugging flows easy to understand.

Notes:

1. The final version of the Xilinx Software tools to support the Spartan and Spartan-XL families was ISE Foundation/Alliance 4.2i Service Pack 3. This version remains available for download on http://www.support.xilinx.com/
2. The Debug “capture” flow mentioned in this documentation was only supported in the original Foundation 3.3.08i and earlier software releases.

Introduction

A design is entered as a schematic or high-level HDL description and then implemented (translated, mapped, placed, and routed) into a Spartan series FPGA using Xilinx software tools. As a result, a bitstream is produced which is used to configure the FPGA. Xilinx FPGAs are built with static RAM. Therefore, they need a data source, such as a PROM or other intelligent interface, to load the configuration data upon power-up.

Configuration is the process of loading the bitstream into one or more FPGAs to define the functional operation of the internal blocks (function look-up table, flip-flop, multiplexer, buffers, pull up/down, slew rate, etc.) and their interconnections (pass transistors). Each configuration bit for the FPGA defines the state of a static memory cell that controls either an internal block or its interconnections. The device can be configured through standard configuration pins or through the Boundary Scan (JTAG) interface.

Xilinx Alliance/Foundation series software programs, the files required by these programs, the download cables used, and other hardware necessary to accomplish the configuration and debugging tasks are discussed in the following section.

Software Programs

There are three different steps (i.e., three different software programs) that can be required to run the configuration flow. These are integrated by the ISE Project Navigator GUI.

The three steps as seen in Project Navigator (and the software programs that are invoked in the background) are:

1. Generate Programming File (Bitgen)
2. Generate PROM File (PROMGen) (optional)
3. Configure Device (iMPACT©).

Generate Programming File (BitGen)

BitGen is a program that takes a mapped, placed, and fully routed NCD design file as its input and produces a configuration bitstream—a BIT file with a .bit extension (see Figure 1). The program can also produce an LL file which is used for debugging the device. The program is a part of the implementation process. Additionally, the BitGen program has the following options:

General Options: BIT, RBT, MSK, and LL file generation. Run DRC checks. Tie unused interconnects to known values.

Configuration Options (including I/O Programmability): Configuration Clock (CCLK) rate, Cyclic Redundancy Check (CRC) error check. TTL or CMOS thresholds for the inputs and outputs (Spartan family only), pull-up and pull-down control for the dedicated configuration pins (TDO, DONE, M0, and M1), Power Down Control (Spartan XL family only) and 5V Tolerant I/Os (Spartan XL family only).

Start-up Timing Options: User clock or configuration clock selection, control of the sequence of output events (DONE going High, user I/Os going active, and release of global Set/Reset).

Readback Options: User clock or configuration clock selection, bitstream verification, and in-circuit debugging control.

Bitgen can be run as a stand-alone tool from the UNIX/DOS command prompt or by running the "Generate Programming File" step in the ISE Project Navigator Graphical User Interface. For information on the command-line mode usage and options, refer to the BitGen chapter of the *Development System Reference Guide.* For instructions on graphical mode usage and options, refer to the help menu within the 4.2i Project Navigator.

Generate PROM File (PROMGen/PROM File Formatter)

PROMGen is the software program used to format single or multiple BIT files into a PROM file (MCS, TEK, or EXO) or a straight HEX file (see Figure 1). The PROM file is used to store the configuration data used to program FPGAs. PROMGen is typically run in the following situations:

- When the FPGA is configured using a serial or parallel PROM.
- When the FPGA is configured using a microprocessor or other intelligent interface.
- To concatenate bitstream files for multiple devices, when they are to be configured in a daisy chain.
- To split an existing PROM file into multiple PROM files.

PROMGen can be run as a stand-alone program from the UNIX/DOS command prompt or by running the "Generate PROM File" option under the "Generate Programming File" step in the Project Navigator GUI. For information on the Command-Line mode usage and options, refer to the PROMGen chapter in the *Development System Reference Guide*. For instructions on the graphical mode usage and options, refer to the Help menu within the 4.2i Project Navigator.

Configure Device (iMPACT© Software)

Xilinx's iMPACT© software is a full featured software tool used for configuring and programming all Xilinx devices (CPLDs, FPGAs, and PROMs). The tool is used to load the configuration data into Xilinx Spartan/XL FPGAs in the following modes:

- Slave Serial mode
- JTAG (IEEE 1149.1) mode
- Express mode (only Spartan-XL)

These modes are briefly discussed in this application note. For a detailed description of these configuration modes, refer to the "Configuration" section of the Spartan and Spartan XL data

sheets and/or the application notes suggested in the "References" section at the end of this document.

For a single device configuration, iMPACT© software uses the BIT file, RBT file, or a PROM file (MCS or EXO). For multiple devices, iMPACT© software uses only the PROM file. For instructions on the graphical mode usage and options, refer to the Help menu within the 4.2i iMPACT Programmer. Note that the only PROM file formats recognized by iMPACT programmer are MCS and EXO.

HW-130 Programmer Software

As the name implies, the HW-130 Programmer software comes with the HW-130 Programmer. The stand-alone HW-130 Programmer is used to load the configuration data into Xilinx serial PROMs.

Files for Configuration

There are various files that are used in the configuration flow. This section describes each file and its importance in the configuration flow.

The BIT file (configuration bitstream) is a binary representation of the logic design. The file is used to configure the Xilinx device. It can also be used to create a PROM file. Note that the bitstream file size is a constant for any given device, regardless of the amount of logic in it. Every bit in the device gets programmed, whether it is used or not. Also, note that the same bitstream is used for Master/Slave Serial modes and the JTAG mode. However, the Express mode bitstream has a different size and cannot be used for any other configuration mode. The Express mode is available only in the SpartanXL family. The bitstream size for all the devices is listed in the *Spartan Program Data* table (Table 17) in the Spartan/XL data sheet.

The RBT (raw bit) file is an ASCII representation of the BIT file and contains both the header and the configuration data. The main benefit of using the RBT file instead of the BIT file is that this file can be viewed using a common text editor. Also, the order in which the 1s and 0s appear in this file is actually what shows up on the DOUT pin of the FPGA during configuration. Monitoring the DOUT pin during the configuration process helps in identifying whether the bitstream gets loaded into the correct device.

The MSK (mask) file is a file used to compare relevant bit locations when reading back configuration data contained in an operating Xilinx device. This file is generated by BitGen if readback (an optional step in the configuration flow) is enabled. Basically, the mask file indicates which bits are configuration bits and which ones are not.

The LL (logic allocation) file contains the position information of CLB flip flops, latches, and the IOBs. The file is used if the user wants to perform an additional step of performing a manual readback operation on the device, which is essentially reading the internal states of the configured device.

The HEX file is an ASCII formatted file, where each hexadecimal digit represents four consecutive configuration bits from the BIT file. The file is typically used in microprocessor based configuration. The HEX file includes only configuration data. The bitstream header information stored in the BIT file is stripped out.

The PROM file is a formatted BIT file that can be used to program the PROM. The file can be MCS, TEK, or EXO format.

The BSDL (.**bsd**) file is the Boundary Scan description file. The file contains the description of test pins and test instructions. It is used for Boundary Scan (IEEE 1149.1) based configuration and testing. A BSD file is required for every Xilinx and non-Xilinx device in the chain.

Programming Equipment

This section describes the hardware equipment that is required to accomplish the device configuration. It assumes that the user already has other equipment, such as the system board

with Spartan/XL device on it, the cables necessary for probing and connecting the power source, and the 2.5V/3.3V/5V DC power supply.

Xilinx Download Cables

The Xilinx Parallel cables (PC III and PC IV), which connect to the parallel port of the PC, support Configuration and Readback Verify of FPGAs. The Parallel Download Cable connects to the parallel port of the PC. For more information on these cables, refer to the *Hardware User Guide*.

Xilinx HW-130/Third Party Programmer

HW-130 Programmer is used to program the Spartan/XL series serial PROMs. The *programmer supports MCS, EXO, and TEK file formats.* For more information, refer to the *HW-130 Programmer* data sheet or the on-line help which is available in the software.

The user can optionally use a third party PROM programmer. For a complete list of the programmers supporting Xilinx FPGAs, refer to the *Programmer Solutions Technical Tips* on the Xilinx web site.

Configuration Modes

This section describes the various configuration modes in which the FPGA can be connected.

The Spartan/XL FPGAs can be configured through standard configuration pins or through the Boundary Scan interface. Using the standard configuration pins, the Spartan family has two modes and the SpartanXL family has three modes. The two modes common to both the families are the Master Serial and Slave Serial modes. These modes are selected using the mode pin(s) on the device. In the Master Serial mode, the FPGA generates its own configuration clock (CCLK) to load the configuration data from an external memory source, such as a Xilinx serial PROM. In the Slave Serial mode, an external signal drives the CCLK input of the FPGA to load the configuration data from an external device such as a lead FPGA or an intelligent interface, such as a microprocessor. The Slave Serial mode is described in the application note, *XAPP098: The Low-Cost Efficient Serial Configuration of Spartan FPGAs.*

The SpartanXL family has an additional mode called Express mode. It has the same timing as Slave Serial mode, except that it is eight times faster because the data is processed one byte instead of one bit per CCLK cycle. The Express mode is described in application note, *XAPP122: The Express Configuration of SpartanXL FPGAs*. The bitstream file for this mode is created by the following command from the command prompt:

```
bitgen -g ExpressMode:Enable -g CRC:Disable -b infile outfile
```

Since the Express Mode does not support error detection using CRCs, it is necessary to disable this feature with the text "`-g CRC:Disable`". For more information on configuration modes, refer to the *Configuration and Test* section in the Spartan/XL data sheet.

Configuration through the Boundary Scan interface is discussed in detail in the "JTAG Flow" section of this application note.

Configuration Flows

Xilinx FPGAs can be configured in several ways. The various possible flows are: Debug, Any_File, JTAG, Processor, and PROM. The Debug flow is the only flow which has additional capability of reading the internal states of the configured device. The Debug, Any_File, and the JTAG flows need a Xilinx cable and Xilinx software. Any of these flows can be used based upon the requirements. These flows are summarized in Table 1 and shown graphically in Figure 1. For assistance with debugging problems encountered during configuration using these flows, refer to *The Configuration Problem Solver* located on the Xilinx web site.

Table 1: **Configuration Flows Overview**

Flow Name	Designer Goal	Program Equipment(1)	Software Required(2)	Files Needed
Debug (Verify only)	Prototype for a single device	Parallel Cable III or IV	• iMPACT© Software	BIT and MSK file
Any_File	Prototype for a single device	Parallel Cable III or IV.	• 4.2.03i ISE Design Tools • iMPACT© Software	BIT/RBT/PROM file
JTAG	Production/Prototype for single device OR multiple devices in daisy chain using Boundary Scan interface(3)	Parallel Cable III or IV	• 4.2.03i ISE Design Tools • iMPACT© Software	BIT and BSDL for each device
PROM	Prototype/Production stage for single device OR multiple devices in daisy chain	PROM HW-130/Third Party PROM Programmer	• 4.2.03i ISE Design Tools • PROM File Formatter • iMPACT© Software • HW-130/Third Party Programmer software	PROM file
Processor	Prototype/Production stage for single device OR multiple devices in daisy chain using Micro-processor	Microprocessor with memory	• 4.2.03i ISE Design Tools • PROM File Formatter • Microprocessor firmware(4)	HEX/PROM file

Notes:
1. The target board which includes the Spartan/XL device(s) must be already present.
2. All the software programs are part of Xilinx ISE series software tools, unless otherwise stated.
3. Boundary-scan (BSCAN) symbol must be instantiated in the design and you must have a separate BIT file for each FPGA in the chain.
4. Not supplied by Xilinx.

Debug Flow

Debugging involves Readback Verify and Readback Capture. Readback Verify consists of reading the programming data sent to the device and comparing it against the original bitstream data to ensure that the device is loaded with the correct data. This can be done using the iMPACT© tool. Verification requires both a bit file (.bit) and a mask file (.msk).

Readback Capture refers to reading the device's internal states (flip-flop outputs, CLB outputs, IOB outputs, and the RAM/ROM bits) to make sure the design performs correctly. Note that the debug process does not interfere with the normal operation of the device. Note also that Readback Capture is not supported by the iMPACT© flow. Legacy tools, i.e., Hardware Debugger with 3.3.08i, or earlier versions of it, support this flow only with the Xilinx XChecker cable. For a detailed description of this flow, refer to the *Answers Database* on http://www.support.xilinx.com (search for "readback capture").

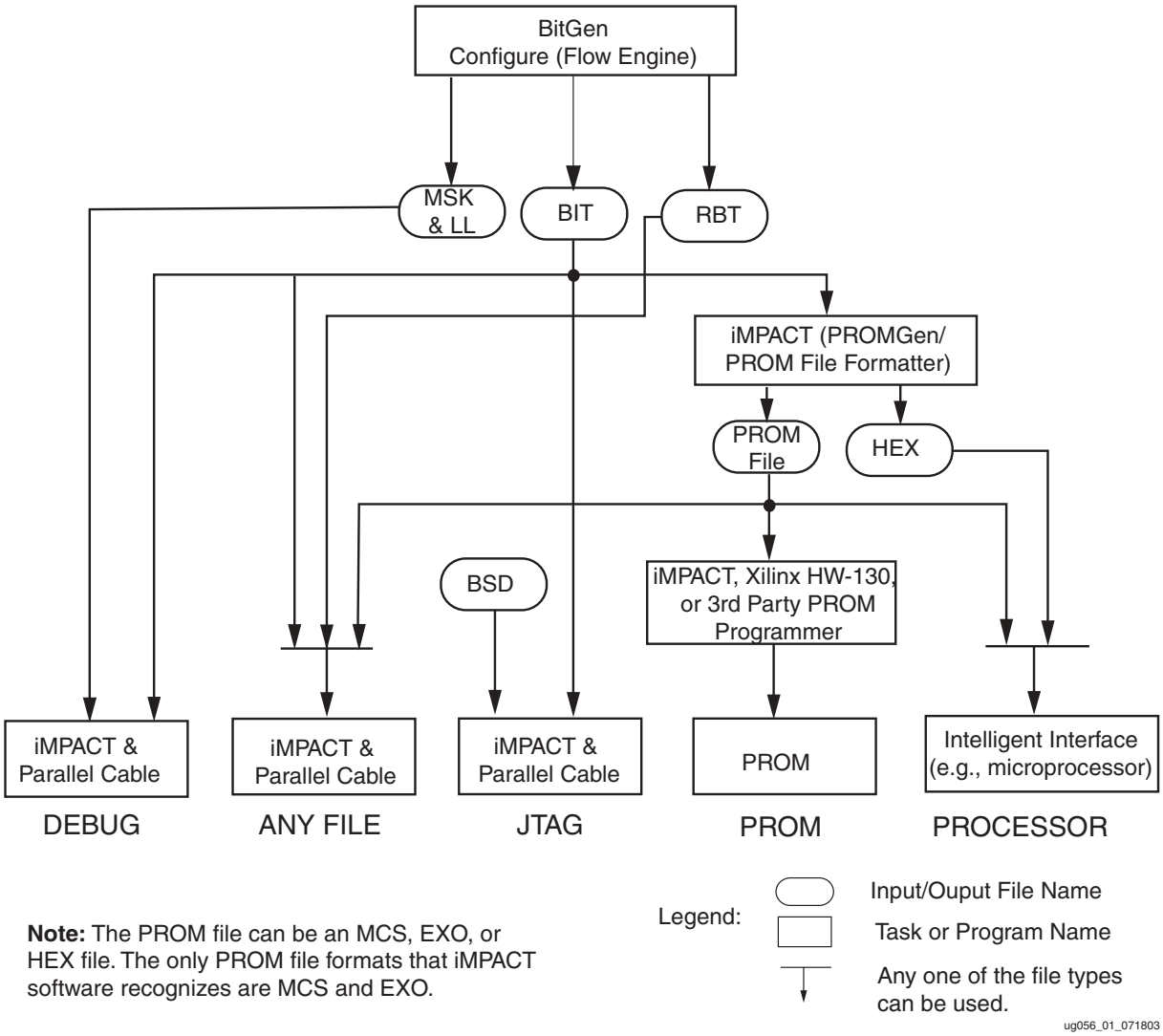

Figure 1: **Spartan/XL Configuration Flows**

Any_File Flow

Here, either Parallel Cable III or IV can be used to configure the device. The Parallel cable is connected to the parallel port of the PC. Also, you can use any of these files: BIT, RBT, or any PROM file (MCS, EXO) to configure the device. The flows for these three files are described in the next section.

Using the BIT file for Slave Serial Configuration

1. Implement the design to get the BIT file.
   a. For a step-by-step procedure to Synthesize and Implement your design in Project Navigator GUI, refer to *ISE Tutorials* available online or the On-line Documentation from the Help menu within the Project Navigator software.
2. Make the Parallel Cable connections.
   a. Make sure the mode pin(s) are set for Slave Serial mode. Although these pins have weak pull-up resistor during configuration, it is recommended to attach an external pull-up of 4.7 kΩ to make sure these pins are not floating.
   b. Connect the Parallel cable connector 1 to the target system as shown in Figure 2.

c. Power up the board using a 5V DC supply if it has Spartan device(s) and 3.3V supply if it has SpartanXL device(s).

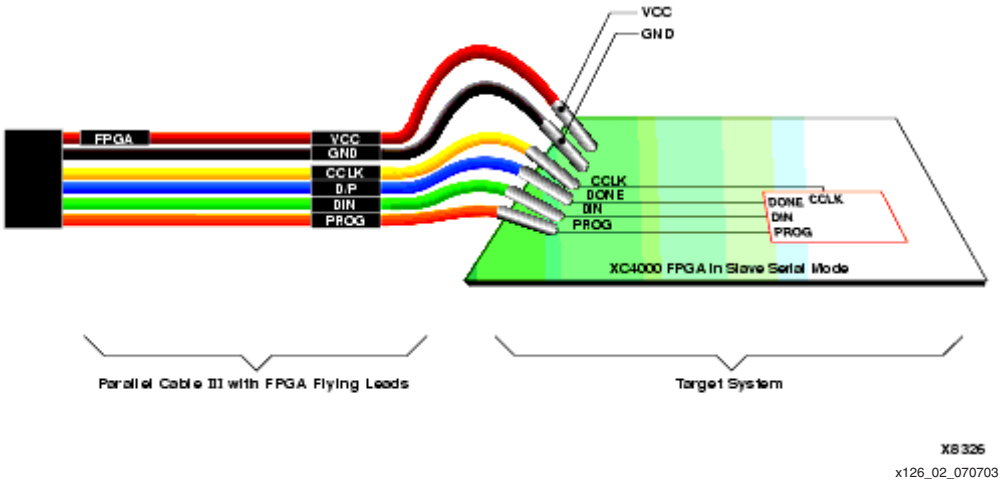

Figure 2: **Parallel Cable Connections**

3. Configure using the iMPACT© tool.
   a. Invoke the iMPACT© program from the Project Navigator (under Implement Design --> Generate Programming File --> Configure Device) or the standalone from the ISE Install (Start--> Programs --> Xilinx ISE 4 --> Accessories --> iMPACT©).
   b. Assign a BIT file to the target device, and configure the device. For more information on using iMPACT© software, refer to the *iMPACT© User Guide* through the on-line Help which is available in the software.

Using the RBT file for Slave Serial Configuration

1. Implement the design to get the BIT and RBT files.

   For a step-by-step procedure to Synthesize and Implement your design in Project Navigator GUI, refer to *ISE Tutorials* available online or the On-line Documentation from the Help menu within the Project Navigator software.

2. Follow the steps 2 and 3 described in the previous section "Using the BIT file for Slave Serial Configuration" to configure the device, assigning an RBT file instead of a BIT file.

Using the PROM file for Multiple Device Chain Slave Serial Configuration

For a single device, the user can use a BIT or RBT file. However, for multiple devices in a daisy chain, a PROM file must be used to concatonate the device bitstreams. No other file format (BIT, RBT) is acceptable.

1. Implement the design to get the BIT file.

   For a step-by-step procedure to Synthesize and Implement your design in Project Navigator GUI, refer to ***ISE Tutorials*** available on-line or the On-line Documentation from the Help menu within the Project Navigator software.

2. Create the PROM (MCS) file using the PROM File Formatter in the ISE install.

   For more information, refer to the PROM File Formatter Guide also available on-line through the ISE Help Menu.

3. Make the Parallel Cable connections.

   Connect the Parallel cable as shown in Figure 2. Follow the instructions described in step 2 of "Using the BIT file for Slave Serial Configuration" to power up the board.

4. Configure using the iMPACT© software.

   Invoke the iMPACT© program from the Project Navigator (under Implement Design --> Generate Programming File --> Configure Device) or standalone from the ISE Install (Start--> Programs --> Xilinx ISE 4 --> Accessories --> iMPACT©). Assign an MCS file to the target chain, and configure the device chain. For more information on using iMPACT© software, refer to the *iMPACT© User Guide* through the on-line Help which is available in the software.

JTAG Flow

If the user is testing a system using Boundary Scan, then it is possible to configure the FPGA through the same Boundary Scan (JTAG) pins. This minimizes additional configuration circuitry. The iMPACT© software and one of the two cables, Parallel Cable III or Parallel Cable IV, are used for this purpose. The programming file used is a BIT file. The BSDL files for all the Xilinx and non-Xilinx devices in the chain also must be available. The steps to perform configuration are described below:

1. Instantiate the BSCAN symbol in your schematics or HDL-based design.

   After configuration, the Spartan/XL Boundary Scan logic is no longer available unless the BSCAN block is instantiated in the design.

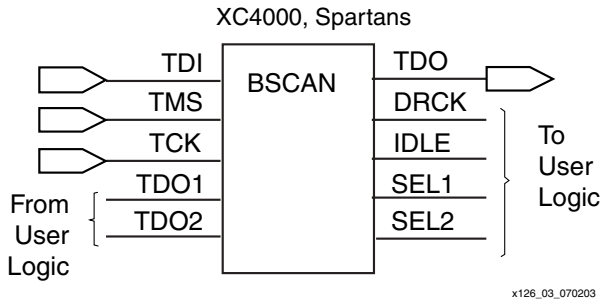

Figure 3: **Example of Boundary Scan Image**

   For more details refer to the *Libraries Guide* section of the 4.2i software version.

2. Implement the design to get the BIT file.

   For a step-by-step procedure to Synthesize and Implement your design in Project Navigator GUI, refer to the *ISE Tutorials* available on-line or the On-line Documentation from the Help menu within the Project Navigator software.

3. Make the Parallel Cable connections.

   Connect one end of the Parallel cable to the PC port (see Figure 4) and the other end to the target system. Power up the board using a 5V DC supply if it has Spartan device(s) and 3.3V supply if it has SpartanXL device(s). For more information, refer to Chapter 2, “Cables,” of the *iMPACT© User Guide*.

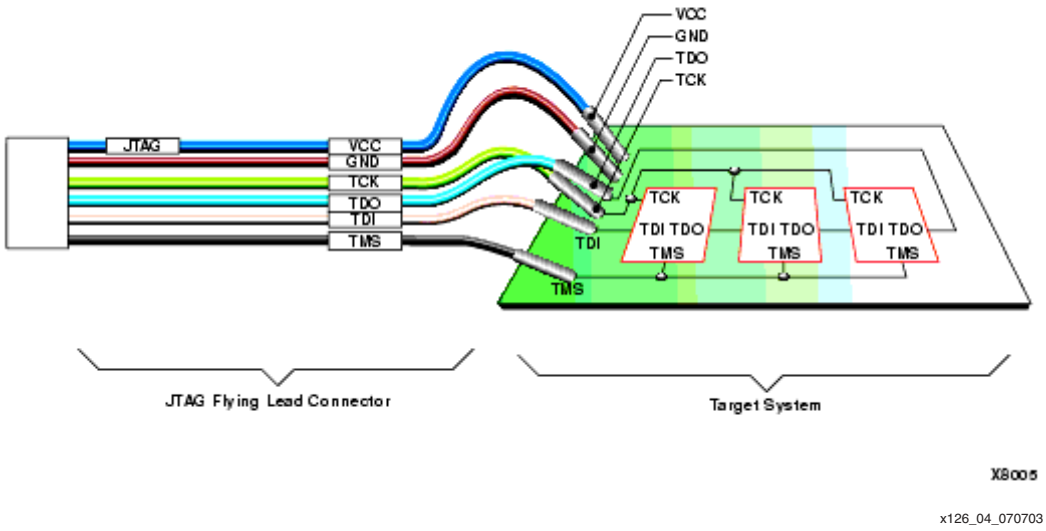

Figure 4: **Parallel Cable Connections**

4. Configure using iMPACT© software.

   Invoke the iMPACT© program from the Project Navigator (under Implement Design --> Generate Programming File --> Configure Device) or stand-alone from the ISE Install (Start--> Programs --> Xilinx ISE 4 --> Accessories --> iMPACT©). Assign a BIT file to the target device, and configure the device. For more information on using iMPACT© software, refer to the *iMPACT© User Guide* through the on-line help which is available in the software.

   For more information on Boundary Scan, refer to application note, *XAPP017: Boundary Scan in XC4000 and XC5200 Series Devices*, which also applies to the Spartan series.

PROM Flow

The PROM is used to store the configuration data for a single FPGA or a daisy chain of FPGAs.

1. Implement the design to get the BIT file.

   For a step-by-step procedure to Synthesize and Implement your design in Project Navigator GUI, refer to the *ISE Tutorials* available on-line or the On-line Documentation from the Help Menu within the Project Navigator software.

2. Create the PROM file using PROM File Formatter in the Design Manager.

   For more information, refer to the *PROM File Formatter Guide*.

3. Load the PROM file into a PROM.

   If the PROM used is a Xilinx In System Programmable (ISP) PROM with a JTAG interface, then iMPACT software can be used to program the PROM with the PROM file (MCS or EXO) using the JTAG flow. If the PROM used is a Xilinx One Time Programmable (OTP) PROM, then the HW-130 Programmer or a supported third-party PROM programmer can be used to program the PROM. Byte-wide PROMs can be used only for Spartan-XL Express mode configuration. For information on Programmers, refer to the *HW-130 Programmer Users Manual* or third-party supplied documentation.

4. Connect the PROM to an FPGA.

   Select the appropriate configuration mode according to the requirements. Connect the PROM to the FPGA by looking at a corresponding circuit diagram under the *Configuration and Test* section in Spartan/XL data sheet.

5. Configure the device.

   Power up the board using a 5V DC supply if it has Spartan device(s) and 3.3V DC power supply if it has SpartanXL device(s). Configure the device. If the PROM is correctly connected to the Spartan/XL and the FPGA mode pins are set appropriately, then configuration automatically commences following power up.

Processor Flow

Note that this section covers the topic in a very high level without going into specific technical details. The user is advised to learn more based on the type of processor that is being used. This method is also described in application note, *XAPP098: The Low-Cost Efficient Serial Configuration of Spartan FPGAs*.

1. Implement the design to get the BIT file.

   For a step-by-step procedure to Synthesize and Implement your design in Project Navigator GUI, refer to the *ISE Tutorials* available on-line or the On-line Documentation from the Help menu within the Project Navigator software.

2. Create the PROM file using PROM File Formatter in the Design Manager.

   Make sure that you create the file format that is acceptable by the intelligent interface. For more information, refer to the *PROM File Formatter Reference/User Guide*.

3. Load the PROM file.

   Load the PROM file into processor's RAM or any other configuration data storage device you are using.

4. Connect the processor to an FPGA.

   Select the appropriate configuration mode according to your requirements. Connect the intelligent interface to the FPGA as shown in the corresponding circuit diagram under the *Configuration and Test* section in the Spartan/XL data sheet.

5. Configure the device.

   Power up the board using a 5V DC supply if it has Spartan device(s) and 3.3V DC supply if it has SpartanXL device(s). Configure the device.

References

Data Sheets

- Spartan/XL data sheet:
  http://direct.xilinx.com/bvdocs/publications/ds060.pdf
- Spartan/XL series Serial PROMs data sheet:
  http://direct.xilinx.com/bvdocs/publications/ds030.pdf
- HW-130 Programmer Data Sheet:
  http://direct.xilinx.com/bvdocs/publications/ds019.pdf

Application Notes

- XAPP017: Boundary Scan in XC4000 and XC5200 Series Devices:
  http://www.xilinx.com/xapp/xapp017.pdf
- XAPP098: The Low-Cost Efficient Serial Configuration of Spartan FPGAs:
  http://www.xilinx.com/xapp/xapp098.pdf
- XAPP122: The Express Configuration of SpartanXL FPGAs
  http://www.xilinx.com/xapp/xapp122.pdf

- FPGA Configuration Application Notes:
  http://support.xilinx.com/apps/config.htm

Technical Support

- ISE Tutorial:
  http://support.xilinx.com ◊ Documentation ◊ Tutorials ◊ 4.1i/4.2i ISE and Foundation
- Configuration Problem Solver:
  http://support.xilinx.com ◊ Problem Solvers ◊ Configuration Problem Solver.
- JTAG Problem Solver:
  http://support.xilinx.com ◊ Problem Solvers ◊ JTAG Problem Solver.
- Programmer Solutions Technical Tips:
  http://support.xilinx.com ◊ Tech Tips ◊ Programmer Solutions (under Implementation Tools)
- Desktop Programmer & Download Cable Support:
  http://www.support.xilinx.com ◊ Hardware ◊ Configuration Solutions ◊ Desktop Programmer & Download Cable Support

Online Software Documents

For the following documents, go to the link below:

- Development System Reference Guide
- PROM File Formatter Reference/User Guide
- iMPACT© User Guide
- Libraries Guide for 4.2i software
  http://support.xilinx.com/support/sw_manuals/xilinx4/

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
07/22/01	0.01	Initial Xilinx release.