



XAPP973 (v1.4) March 8, 2010

## Indirect Programming of BPI PROMs with Virtex-5 FPGAs

Author: Stephanie Tapp

### Summary

Virtex®-5 FPGAs and ISE® software support configuration from and programming of industry-standard, parallel NOR flash memory (BPI PROMs). Industry standard BPI PROMs are an alternate solution for Virtex-5 FPGA designs whose requirements are not met by the Platform Flash XL configuration and storage device (see [\[Ref 1\]](#) for more information on Platform Flash XL). The iMPACT software, included in the ISE® development software tools, provides indirect programming for select BPI PROMs during prototyping. This application note demonstrates how to program a Numonyx StrataFlash P30 BPI PROM indirectly using iMPACT 11.4 and a Xilinx cable. In this solution, the Virtex-5 FPGA serves as a bridge between the IEEE Std 1149.1 (JTAG) bus interface and the BPI bus interface. The required hardware setup, BPI-UP PROM file generation flow, and BPI indirect programming flow are shown. The Virtex-5 FPGA BPI-UP configuration sequence is also described.

**Note:** Parallel NOR flash memory is referred to by the term BPI PROM throughout this document.

### Introduction

Xilinx FPGAs are CMOS configurable latch (CCL) based and must be configured at power-up from a non-volatile source. FPGA configuration is traditionally accomplished with a JTAG interface, a microprocessor, or the Xilinx PROMs (Platform Flash PROMs). In systems where the easiest solution is preferred, Master Serial mode with a Xilinx Platform Flash PROM is still the most popular configuration mode because it has:

- A direct JTAG interface for programming
- The smallest interface pin requirement for configuration
- Flexible I/O voltage support

Moreover, this solution is available for any Virtex-5 FPGA device (refer to [\[Ref 2\]](#) for more information).

In addition to the traditional methods, a direct configuration interface to third-party BPI PROMs is included on Virtex-5 FPGAs to address changing system requirements. Systems with a BPI PROM already onboard for random-access, non-volatile application data storage can benefit from consolidating the configuration storage into the same memory device.

Similar to the traditional configuration memories, BPI PROMs must be loaded with the configuration data. BPI PROMs have a single interface for programming, and three primary methods to deliver the data to this interface:

- Third-party programmers (off-board programming)
- In-system programming (ISP) with an embedded processor
- Indirect ISP (using JTAG or custom solution)

Production programming is often accomplished off-board with a third-party programmer or in-system with a JTAG tool vendor. During the prototyping phase, indirect ISP is preferred to easily accommodate design iterations.

Because BPI PROMs do not have a JTAG interface, extra logic is required to serve as a bridge between the iMPACT programmer (using a cable to drive the JTAG bus interface), and the BPI

PROM (connected to the FPGA's BPI bus interface). This extra logic must be downloaded into the FPGA by iMPACT before indirect programming is possible.

This application note is divided into three main sections. The first section discusses the hardware connections required for the indirect in-system programming of BPI PROMs for prototype designs. The second section shows the Xilinx software tool flows for generating a PROM file formatted for 16-bit BPI-UP mode and then for programming the select BPI PROMs. The third section provides a basic configuration flow overview for the FPGA after the BPI PROM is programmed and describes expectations when using this indirect setup.

## iMPACT Indirect In-System Programming with a Virtex-5 FPGA

The basic hardware setup required for the iMPACT indirect BPI PROM programming method is shown in [Figure 1](#).

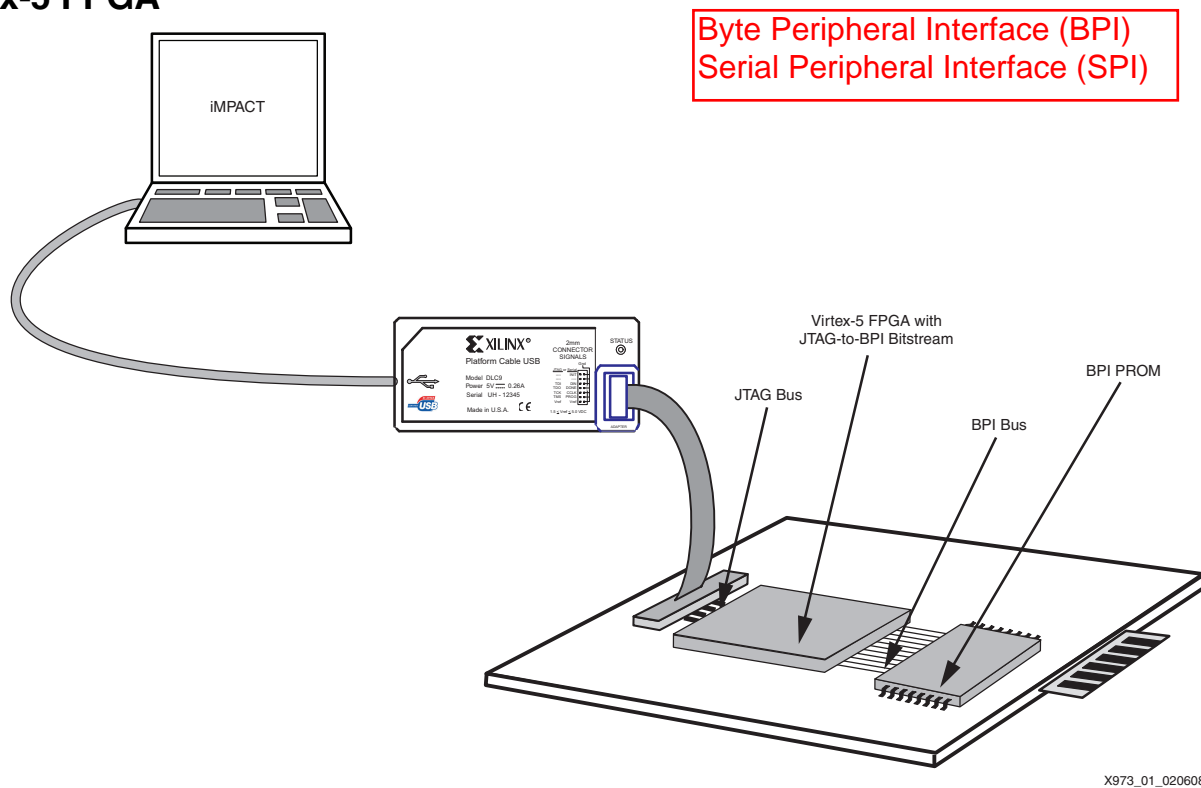


Figure 1: iMPACT Indirect BPI PROM Programming with a Virtex-5 FPGA

### Minimum Requirements

- Virtex-5 FPGA
- BPI PROM (refer to [Table 1](#))
- Xilinx Cable and Connector (refer to [Table 4, page 7](#))
- ISE iMPACT Software 11.4

**Note:** Indirect BPI PROM programming was introduced with limited device support in iMPACT 9.2i. This application note demonstrates the software flow and lists the device support in iMPACT 11.4.

The schematic in Figure 20-2 illustrates how to wire the ispJTAG connector, FPGA, and SPI Serial Flash.

[illegible]

- The download header has standard 0.1 inch pin-to-pin spacing.
- The 4.7K pull-down resistors prevent spurious clock pulses during  $V_{CC}$  ramp-up. Place the resistors close to their clock line to keep the stub length as short as possible.
- The CCLK frequency can be as high as 50MHz, so keep this trace fairly short.
- $V_{CCIO}$  for the bank that drives the signals to the SPI Serial Flash must match the SPI Serial Flash  $V_{CC}$  (today that voltage is 3.3V but this will change over time as Flash chip geometries decrease). For all packages these signals are located in bank 3.
- During configuration CCLK drives the SPI Serial Flash CLK pin, but once the FPGA completes configuration CCLK goes into tri-state. The CCLK pin is not accessible by user code so CCLK needs to be wired to a nearby General Purpose I/O pin (GPIO) to allow the FPGA fabric to supply a clock to the SPI Serial Flash. This pin is part of the Soft SPI Interface and is unique to each package. If the user embeds the Soft SPI Interface into their code then this pin, along with the other pins wired to the SPI Serial Flash, must be locked using the Pre-Map Preference Editor in ispLEVER. A complete list of these pins is found in Table 20-5.
- In addition to standard decoupling practices, place a decoupling capacitor close to the connector's  $V_{CCJ}$  pin. Any standard ceramic capacitor value may be used, for example 0.1  $\mu F$ , 0.01  $\mu F$ , etc.

In order to program SPI Serial Flash via ispJTAG the FPGA must contain the Soft SPI Interface. Programming the SPI Serial Flash with ispVM System and an ispDOWNLOAD cable makes this transparent to the user. The software simply loads a default Soft SPI Interface bitstream into the FPGA and then loads the user bitstream into the Flash. Once programming of the SPI Serial Flash is complete the FPGA configures itself by reading the Flash. Again, software makes all of this transparent to the user so that it is no different than programming any other serial boot device.

1. Connect the ispDOWNLOAD Cable to the appropriate header and apply power to the board.
2. Start the ispVM System software.
3. From the main window click on the **Scan** button located on the toolbar. The LatticeECP/EC device should be detected automatically (if it's not detected, check the ispJTAG connections and make sure the board is powered up). The resulting screen should be similar to Figure 20-3.

ICS ispVM6 System [New Scan Configuration Setup\*]

File Edit View Project ispTools Options Window Help

Icons: [File] [Folder] [Disk] [Floppy] [CD] [GO] [NEW] [ZVP] [ATE] [VME] [CPU] [HW 300] [LOG] [Help]

Index	Device List	FileName/R-Length	Operation	Status
1	LFEC/ACP20E		Bypass	N/A

Status

For Help, press F1

- 20-7