



XAPP501 (v1.3) June 10, 2002

Configuration Quick Start Guidelines

Author: Stephanie Tapp

Summary

This application note discusses the configuration and programming options for Xilinx Complex Programmable Logic Device (CPLD), Field Programmable Gate Array (FPGA), and PROM families and demonstrates some of the most popular configuration methods used for each family. This document includes configuration quick start guidelines for the Virtex™ Series, Virtex-II Series Platform FPGAs, Spartan™, XPLA3, XC9500, XC17S00, and XC18V00 families.

Introduction

To help meet the different requirements of system designers, various methods for configuring Xilinx FPGAs, CPLDs, and PROMs are available. This document describes the different configuration modes to assist the designer in selecting an appropriate configuration method. Examples are provided for some of the most popular configuration methods that allow for production and prototyping.

There are three general steps necessary to configure or program a Xilinx programmable logic device:

- **Step 1 - Design Entry**

Software design entry tools are used to create a design in VHDL, Verilog, ABEL, or Schematic. Refer to *Quick Start Books* at: http://toolbox.xilinx.com/docsan/3_1i/

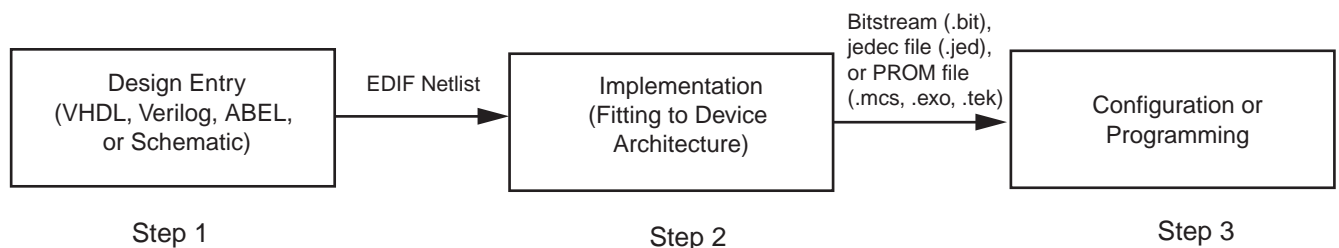
- **Step 2 - Implementation**

Software implementation tools are used to fit the design netlist into the desired Xilinx architecture and produce a configuration bitstream. Refer to *Development System Reference* at: http://toolbox.xilinx.com/docsan/3_1i/

- **Step 3 - Configuration or Programming**

Configuration is the process of downloading configuration data into an FPGA using an external data source, such as a PROM, CPLD, or microprocessor.

Programming is the process of loading the configuration data or program data into a CPLD or PROM. See [Figure 1](#).



x501_01_010901

Figure 1: Flow Overview to Configuration and Programming

© 2002 Xilinx, Inc. All rights reserved. All Xilinx trademarks, registered trademarks, patents, and further disclaimers are as listed at <http://www.xilinx.com/legal.htm>. All other trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change without notice.

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

Device Definitions

Xilinx produces three types of programmable logic devices. Below is a brief description of these product types to help the designer understand the configuration needs for each:

FPGAs

Xilinx FPGAs are volatile because they are based on SRAM technology. That is, the device loses its configuration if the power to the device is turned off. FPGAs typically utilize an external memory device, such as a PROM for production type environments to prevent the loss of configuration data in a power outage. FPGAs can be programmed using a cable without an external storage device.

PROMs

Xilinx PROM devices are non-volatile devices, which are typically used to store configuration data for Xilinx FPGAs. These devices are available in two different types:

- One-time programmable (XC17x)
- In-system programmable (ISP) reprogrammable PROMs (XC18V00).

CPLDs

Xilinx CPLDs are generally non-volatile, utilizing EEPROM or FLASH Cell technology. Since they do not require an external memory device, the non-volatile nature of Xilinx CPLDs leads to different requirements in the programming process.

Software Tools Overview

This section briefly describes the software available for Xilinx FPGA, CPLD, and PROM products. The software tools shown in **Table 1** serve two purposes for configuring and programming a Xilinx device:

1. Generate a configuration or programming file; a bitstream (**.bit** file) for FPGAs, a jedec (**.jed**) file for CPLDs, or a PROM file (**.mcs**, **.exo**, **.tek**) for PROMs.
 - Design Entry Tools - Generate a design netlist.
 - Implementation Tools - Generate a bitstream, **jedec** file, or PROM file from a design netlist.
2. Download the file to the device:
 - JTAG Programmer - Xilinx Software Download Tool used to program any chain of Boundary Scan (JTAG) compliant Xilinx devices.
 - Hardware Debugger – Xilinx Software Download Tool used to program FPGA devices in Slave Serial mode or SelectMAP mode.

Table 1: Available Xilinx Software Packages

	Foundation (DE>Imp>CP) ⁽¹⁾	Foundation ISE (DE>Imp>CP) ⁽¹⁾	WebPACK ISE (DE>Imp>CP) ⁽¹⁾	Alliance (Imp>CP) ⁽¹⁾	WebFITTER (DE>Imp) ⁽¹⁾
Virtex Series/ Virtex-II Series Platform FPGAs	X	X	XCV300E only ⁽²⁾	X	
Spartan-XL/II Families (FPGA)	X	X	Spartan-II only ⁽²⁾	X	
XC9500/XL/XV Families (CPLD)	X	X	X	X	X
XPLA3 Family (CPLD)			X		X

Table 1: Available Xilinx Software Packages (Continued)

	Foundation (DE>Imp>CP) ⁽¹⁾	Foundation ISE (DE>Imp>CP) ⁽¹⁾	WebPACK ISE (DE>Imp>CP) ⁽¹⁾	Alliance (Imp>CP) ⁽¹⁾	WebFITTER (DE>Imp) ⁽¹⁾
XC18V00 Family (PROM)	X	X	X	X	
XC17S00 Family (PROM)	X	X	X	X	

Notes:

1. DE = Design Entry Software included; IMP = Implementation Software included; CP = Configuration or Programming Software included.
2. The Virtex XCV300E and Spartan-II families are the FPGA device members supported in the latest version of WebPACK.

Selecting a Configuration Mode

Before deciding on the configuration environment, it is important to be aware of the available configuration modes. Table 2 shows the modes supported by each family with a description of each mode following the table.

Table 2: Modes Supported (by Family)

	JTAG Mode	SelectMAP/ Slave-Parallel Mode	Master-Serial Mode	Slave-Serial Mode	Express Mode	Standalone Programmer Mode
Virtex Series/ Virtex-II Series Platform FPGAs	X	X	X	X		
Spartan-XL/II Families (FPGA)	X	Spartan-II only	X	X	Spartan-XL only	
XC9500-XL/XV Families (CPLD)	X					X
XPLA3 Family (CPLD)	X					X
XC18V00 Family (PROM)	X					X
XC17S00 Family (PROM)						X

FPGA Supported Modes

JTAG or Boundary Scan Mode

JTAG or Boundary Scan mode is an industry standard (IEEE 1149.1, or 1532) serial programming mode. External logic from a cable, microprocessor, or other device is used to drive the JTAG specific pins, Test Data In (TDI), Test Mode Select (TMS), and Test Clock (TCK). This mode has gained popularity due to its standardization and ability to program FPGAs, CPLDs, and PROMs through the same four JTAG pins. The data in this mode is loaded at one bit per TCK.

:

SelectMAP/Slave-Parallel Modes

SelectMAP™ mode is supported by the Virtex Series, and Slave-Parallel mode is supported by the Spartan-II family. SelectMAP/Slave-Parallel modes allow parallel reading and writing through byte-wide ports. An external clock source, microprocessor, download cable or other FPGA is required. The data is loaded one byte per CCLK in this mode. This mode is typically used as a configuration mode on Virtex-E/Virtex-II or Spartan-II devices when configuration speed is an important factor.

Express Mode

Express mode is supported by the Spartan-XL devices. It is similar to the Slave Serial mode, except that data is processed one byte per CCLK cycle, instead of one bit per CCLK cycle. An external source is used to drive CCLK, while byte-wide data is loaded directly into the configuration data shift registers.

Master-Serial Mode

Master-Serial mode is supported by all Xilinx FPGA families, but not Xilinx CPLDs. It is the simplest configuration method for FPGAs. The FPGA loads configuration data from a serial PROM. Using the FPGA to provide the clock, it virtually load itself and utilizes its internal oscillator, which drives the configuration clock. The FPGA provides all the control logic. In this mode, data is loaded at one bit per CCLK.

Slave-Serial Mode

Slave-Serial mode, like Master-Serial, is supported by all Xilinx FPGA families, but not Xilinx CPLDs. It uses an external clock and allows for daisy-chain configurations. In this mode, an external clock, such as a microprocessor, another FPGA, or download cable are required. Data in this mode is loaded at one bit per CCLK.

CPLD/PROM Supported Modes**Standalone Programmer Mode**

Standalone Programmer mode is supported by Third Party Programmers and the HW-130 Programmer.

JTAG or Boundary Scan Mode

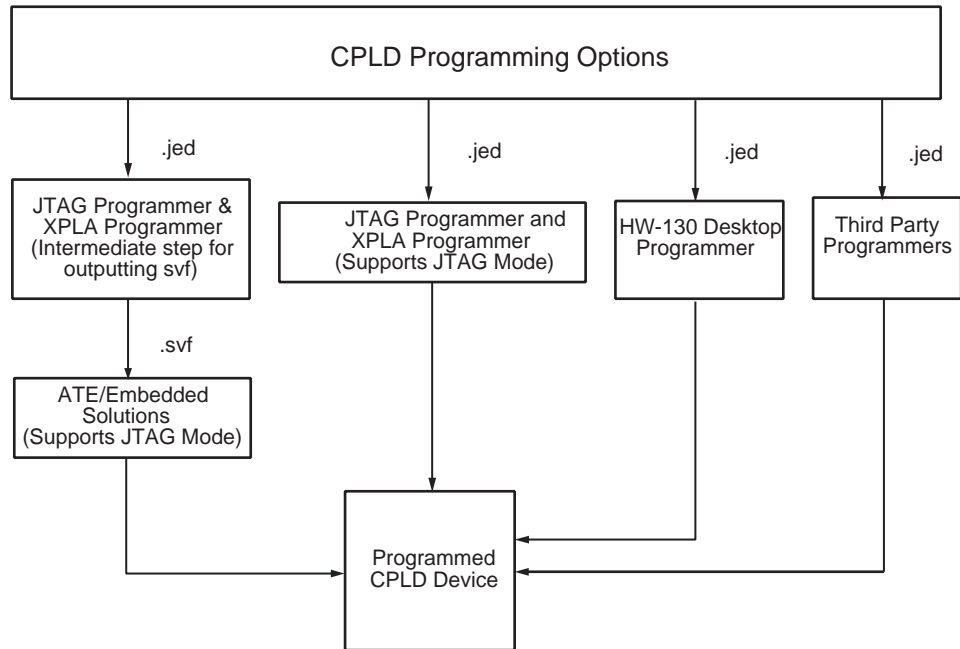
JTAG or Boundary Scan mode is an industry standard (IEEE 1149.1, or 1532) serial programming mode. External logic from a cable, microprocessor, or other device is used to drive the JTAG specific pins, Test Data In (TDI), Test Mode Select (TMS), and Test Clock (TCK). This mode has gained popularity due to its standardization and ability to program FPGAs, CPLDs, and PROMs through the same four JTAG pins. The data in this mode is loaded at one bit per TCK.

Programming and Configuration Options

The next section gives an overview of all configuration or programming flows for each family.

CPLD Programming Options

The CPLD programming flow starts after the CPLD fitter software generates the **jedec** file. The CPLD Fitter software is embedded in all Xilinx software packages shown in [Table 1](#). [Figure 2](#) shows the options available for downloading the **jedec** file into the CPLD device.



x501_02_010901

Figure 2: CPLD Programming Options

ATE/Embedded Solutions

Embedded solutions are becoming very popular as board densities increase and microprocessors become commonplace in many systems. For examples of how to use Xilinx CPLDs in ATE or embedded systems, see the websites at: <http://www.xilinx.com/isp/ate.htm> and <http://www.xilinx.com/isp/ess.htm>

JTAG Programmer

JTAG Programmer is a software tool that is used to download a **jedec** file to a CPLD using a cable through the Boundary Scan (JTAG) pins. This software program is included in all of the Xilinx software packages shown in [Table 1](#) with the exception of the WebFITTER. See the *JTAG Programmer Guide* at: http://toolbox.xilinx.com/docsan/3_1i/

HW-130

HW-130 is the Xilinx stand-alone programmer, which is used in prototyping environments for programming many of the Xilinx CPLDs.

For more information about this product, see the website at:

<http://www.xilinx.com/support/programr/ps.htm>

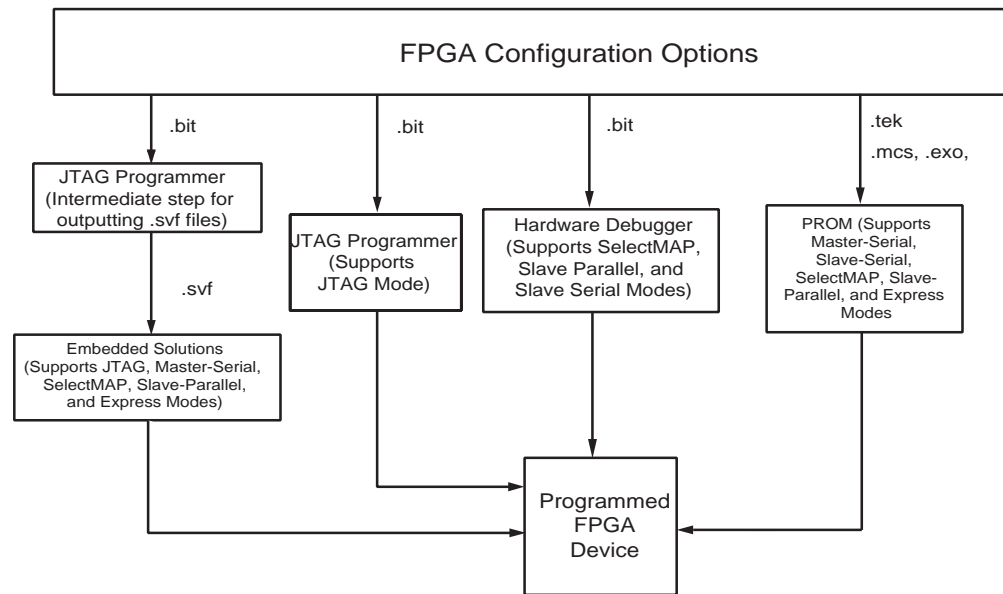
Third Party Programmers

Third Party Programmers, such as Data I/O and BP Microsystems, support many Xilinx CPLDs. For details on available third party support, see the website at:

<http://www.xilinx.com/support/programr/ps.htm>

FPGA Configuration Flows

Refer to **Table 1** for the software packages available to generate FPGA bitstreams. **Figure 3** shows the options for downloading the bitstream into the FPGA device.



x501_03_011901

Figure 3: FPGA Configuration Options

JTAG Programmer

JTAG Programmer is a software program that is used to download a bitstream to FPGA through a Xilinx Parallel III or MultiLINUX cable using the Boundary Scan (JTAG) connections. This software tool is included in all Xilinx software packages for FPGAs mentioned in **Table 1**. See the *JTAG Programmer Guide* at: http://toolbox.xilinx.com/docsan/3_1i/

Hardware Debugger

Hardware Debugger is a software tool that is used to download a bitstream into a FPGA using a Xilinx Parallel III or MultiLINUX cable. These software downloads can configure the device in Slave-Serial, SelectMAP™, or Slave-Parallel modes. This software tool is included in all Xilinx software packages for FPGAs mentioned in **Table 1** except WebPACK ISE. See the *Hardware Debugger Guide* at: http://toolbox.xilinx.com/docsan/3_1i/

Embedded Solutions

Embedded solutions are becoming very popular as board densities increase and microprocessors become commonplace in many systems. For examples of using Xilinx FPGA devices in embedded systems, see the website at:

http://www.xilinx.com/xlnx/xil_prodcat_product.jsp?title=isp_ess_page

PROM

A PROM is a companion memory device to the FPGA. Configuring the FPGA from a PROM is one of the most widely used configuration methods. The PROM must be configured with the data intended for the FPGA. When the PROM is placed in the system, the FPGA configures itself from the memory device. This method supports Master-Serial, Slave-Serial, SelectMAP, Slave-Parallel, and Express modes.

PROM Flows

PROMs, companion devices for the FPGA families, require an additional software tool, the PROM File Formatter. This software tool converts the bitstream targeted to an FPGA family into a PROM file. The PROM File Formatter software is included in the FPGA Xilinx software packages. Figure 4 shows the options available for downloading the PROM file into the PROM device.

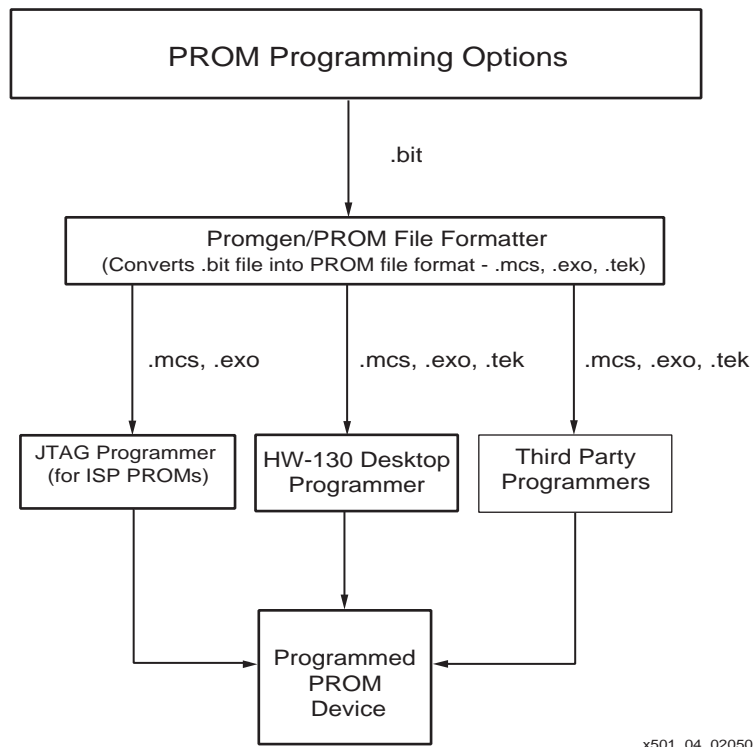


Figure 4: PROM Programming Options

JTAG Programmer

JTAG Programmer is a software tool that is used to download a PROM file into an ISP PROM (XC18V00). This software tool is included in all Xilinx software packages for PROMs mentioned in Table 1. See the *JTAG Programmer Guide* at the website:

http://toolbox.xilinx.com/docsan/3_1i/

PROMgen/PROM File Formatter

PROMgen/PROM File Formatter are software tools that create PROM files for serial or byte-wide configuration Xilinx PROMs. These tools convert .bit files into one of three popular standard PROM file formats: MCS, EXO, or TEK. The PROM File Formatter is simply the graphical user interface (GUI) version of the PROMgen command line software. This software tool is included in all Xilinx software packages for PROMs mentioned in Table 1. For this command line tool, refer to the *Development System Reference Guide* for the "PROMgen" section or the "Prom File Formatter" section at: http://toolbox.xilinx.com/docsan/3_1i/

HW-130

HW-130 is the Xilinx desktop programmer. The Xilinx desktop programmer is used in prototyping environments and can program both types of Xilinx PROMs.

For more information about this product, see the website:

<http://www.xilinx.com/support/programr/ps.htm>

Third Party Programmers

Third Party Programmers, such as Data I/O and BP Microsystems, support many Xilinx CPLDS. For details on available third party support, see the website at: <http://www.xilinx.com/support/programr/ps.htm>

Tips for Selecting a Configuration or Programming Method

The system designer needs to consider whether the system being developed is for a prototyping environment, a production environment, or both. After this question is answered, the configuration choices can be narrowed. The recommended options for a prototyping environment are discussed first.

Prototyping Environment

A prototyping environment requires easy access to the device for multiple reconfigurations. In this type of environment, two options work best:

Table 3: Prototyping Recommended Configuration Options

Options	CPLD	FPGA	PROM
Xilinx Configuration or Programming Software with Cable	JTAG & XPLA Programmer Software	Hardware Debugger Software	JTAG Programmer Software (ISP PROM Option)
Desktop Programmer	HW-130	Not Applicable (FPGAs are volatile devices)	HW-130

Production Environment

For a Production environment, reconfiguration is not as high a priority as fast programming times and robust configuration methods that retain configuration data. For this kind of environment, several options are listed in [Table 4](#).

Table 4: Production Recommended Configuration Options

Options	CPLD	FPGA	PROM
Embedded Solution	Custom code (refer to XAPP058)	Custom code (refer to XAPP058)	Custom code (refer to XAPP058 for ISP PROMs)
Automatic Test Equipment (ATE)	ATE/Third Party JTAG tools	Not Typical (FPGAs are volatile devices)	ATE/Third Party JTAG tools
Third Party Programmers (e.g., BP Microsystems, DataIO)	Programmers supported	Not Applicable (FPGAs are volatile devices)	Programmers supported
PROM	Not Applicable (CPLDs are non-volatile devices)	PROM supported	Not Applicable

Xilinx Common Configuration and Programming Setups

This section provides the Xilinx user a quick summary of some of the more common configuration and programming setups. The setups shown here are designed to be used for a prototyping and production environment. A brief description of the setup advantages, required software and hardware, and hardware connections are described for each setup.

CPLD JTAG Chain Setup

The CPLD JTAG Chain (XC9500/XL/XV or XPLA3) is the most common programming method for CPLDs, since it can be used for both prototyping and production environments.

- **Advantages:**
 - In-System Programming support
 - Requires only the four JTAG pins to configure and test the chain.
- **Hardware Used:**

Combination of Xilinx CPLD devices (XC9500/XL/XV or XPLA3)

 - For Prototyping: Xilinx Cable (Parallel III or MultiLINX)
 - For Production: Microprocessor (Embedded Solution), Automatic Test Equipment (ATE), or Third Party Programmer
- **Software Used:**

Xilinx CPLD programming **jedec** files (Foundation/Alliance or WebPACK software packages create the **jedec** file.)

 - For Prototyping: JTAG Programmer Software (for XC9500/XL/XV) or XPLA Programmer Software (for CoolRunner XPLA3)
 - For Production: Xilinx CPLD programming **jedec** files are converted to Xilinx Serial Vector Files (**.svf**) for use with a microprocessor, ATE, or Third Party Programmer. See XAPP058 at: <http://www.xilinx.com/xapp/xapp058.pdf>

See the Xilinx [ATE](#) and [Programmer](#) webpages for more information.

Hardware Considerations

The connections to implement a CPLD JTAG chain are shown in [Figure 5](#).

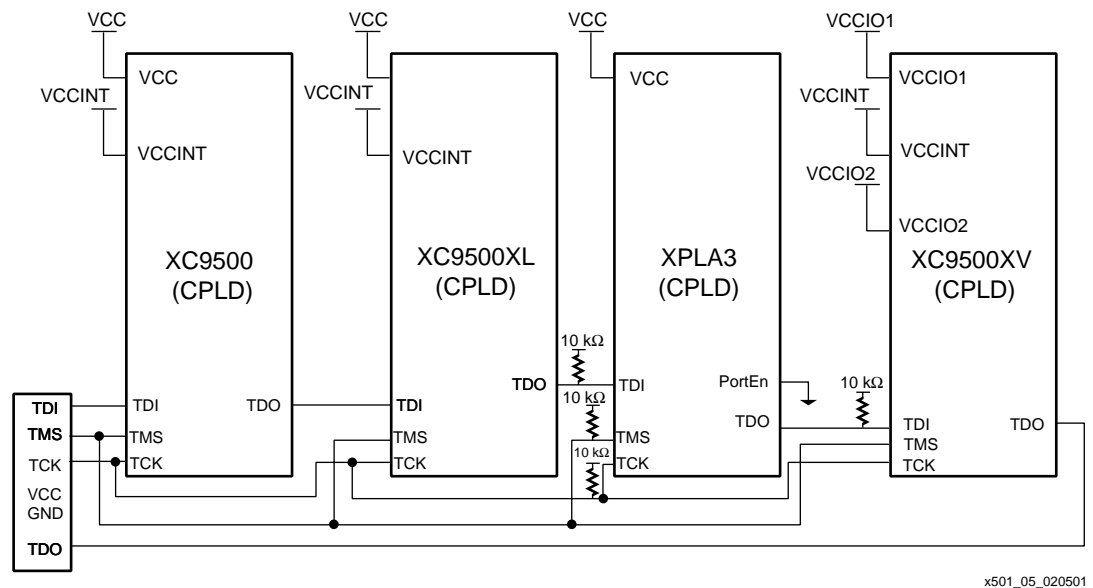


Figure 5: CPLD JTAG Chain Connections

When using a JTAG Chain with mixed voltage devices, as shown in [Figure 5](#), extra care must be taken to ensure the integrity between the devices. Refer to the device's data sheet for the appropriate voltage operating ranges. In this example, using a 5V, 3.3V, and 2.5V device in a single JTAG chain, although not as typical, allows consideration for a mixed voltage chain to be discussed.

To accommodate both the XC9500 (5V) and XC9500XV (2.5V) devices, the V_{CCIO} signals should be tied to 3.3V. This drives the TDO pin on all of the devices at 3.3V, which meets the voltage requirements. For the XC9500XV, only the V_{CCIO} bank (1 or 2) where the TDO pin is located needs to be driven at 3.3V.

When using a cable with this setup, it should be driven at 3.3 V. This ensures that the TMS and TCK TAP pin values are within the required voltage ranges for all of the specified devices in [Figure 5](#). Since the 5 V part receives slightly lower, but acceptable signal levels, good signal integrity is another practice that should be given appropriate consideration. Refer to the following sections for tips and techniques to ensure proper integrity strength and general practices on implementing a CPLD JTAG chain.

General JTAG Checklist

1. Include buffers on TMS and TCK signals interleaved at various points for larger JTAG chain of more than five devices to account for unknown device impedance.
2. Make sure the V_{CC} is within the rated value: 5V $\pm 5\%$ for the XC9500 device, 3.3V $\pm 10\%$ for XC9500XL/XPLA3 devices, and 2.5V $\pm 10\%$ for the XC9500XV device.
3. Provide both 0.1 μ F and 0.01 μ F capacitors at every V_{CC} point of the chip and attach them directly to the nearest ground.

XC9500/XL/XV Specific Checklist

1. The TDI and TMS (JTAG) pins have internal pullups for the XC9500/XL/XV families and do not require any external pullups since the pins are dedicated for JTAG use.
2. It is vital that the XC9500/XL/XV devices be provided with very clean (noise free) voltage supplied to the V_{CCINT} pins within the correct range.
3. The JTAG Clock for the XC9500/XL devices, TCK, has a maximum frequency of 10 MHz.
4. When deciding the placement of devices in a mixed voltage chain, care should be taken to ensure compatibility. For instance, the XC9500XV parts are not 5 V tolerant, and care should be taken to ensure the device inputs are driven by the appropriate voltage levels.

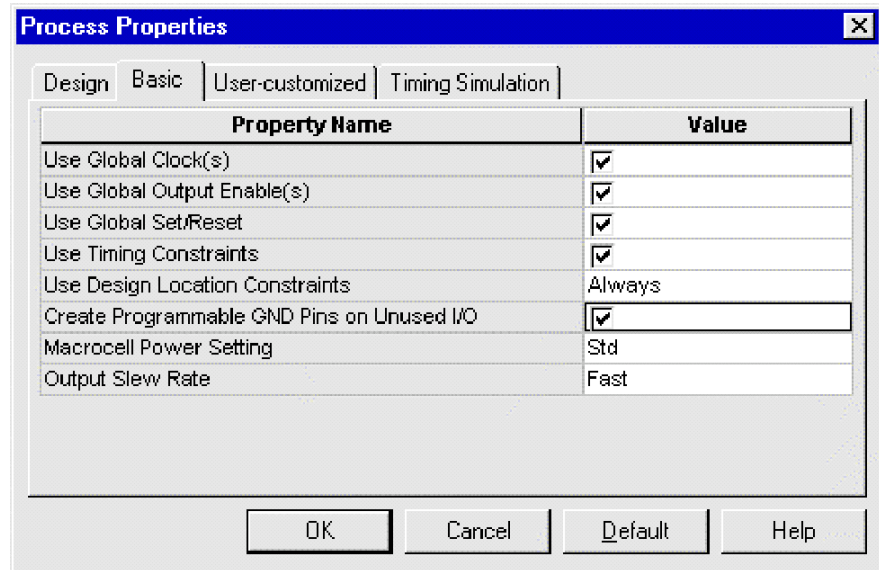
XPLA3 Specific Checklist

1. The PortEn pin should be connected to ground when using dedicated JTAG pins as recommended in this programming setup example.
Note: If the JTAG pins need to be used as dual-purpose I/Os, the PortEn pin should be jumpered out. The PortEn pin is driven high to re-establish connection with the JTAG pins.
2. The XPLA3 JTAG pins should have an external 10 k Ω resistor placed on them to prevent them from floating.
3. The JTAG clock for the XPLA3 devices, TCK, has a maximum frequency of 10 MHz.

Software Implementation Considerations

The software **jedec** files need to be created to program the devices. The Software Implementation Options that should be considered when creating the **jedec** programming files are shown in the following GUIs. The XC9500/XL/XV implementation options are the same for all Xilinx software packages. The implementation software default settings for the XC9500/XL/XV devices are recommended for this example with the exception of the option described below:

“Create Programmable GND Pins on Unused I/O”– By default this option is not selected. It is recommended to check this option to prevent unused I/O from floating and drawing additional power. See [Figure 6](#).

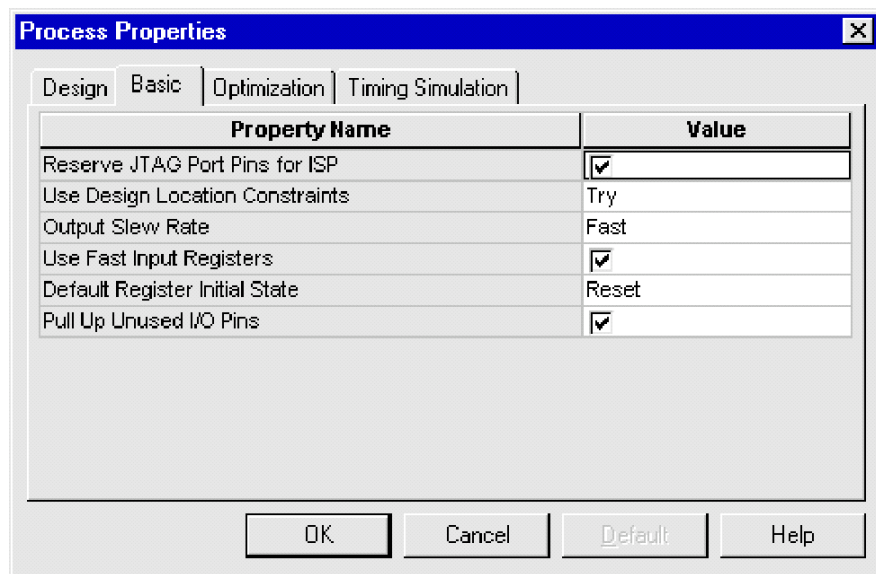


x501_06_011001

Figure 6: Xilinx XC9500/XL/XV Implementation Options in WebPACK ISE Software Package

The software default settings in [Figure 7](#) for the XPLA3 devices are recommended to be used for this example. Two of the most important options for programming a user should be aware of are described below:

- “Pull Up Unused I/O Pins” – By default, this option is selected. This is the recommended state to prevent additional power draw due to the CMOS I/O.
- “Reserve JTAG Port Pins for ISP” – By default, this option is selected and for purposes of this example, it is recommended to leave this option as default. This option would be deselected only if the user intended the JTAG pins to be used as dual-purpose I/Os.



x501_07_122800

Figure 7: XPLA3 Implementation Options in WebPACK ISE Software Package

Software Download Considerations

The last step needed to complete the implementation of a CPLD JTAG Chain is programming the bitstream to the device. With the JTAG Chain programming, this can be done for both prototyping and production environments.

Prototyping Environment

In a prototyping environment, it is very common to use a Xilinx programmer, which is comprised of a stand-alone downloadable software module that requires a Xilinx cable and access to the JTAG pins of the devices.

- When targeting the XPLA3 family, use the Xilinx XPLA Programmer to download the bitstream to the part. Always be certain to use the latest version of the software, which is available from: <http://www.xilinx.com/sxpresso/webpack.htm>
- When targeting the XC9500/XL family, use the Xilinx JTAG Programmer to download the bitstream to the part. Always be certain to use the latest version of the software, which is available from: <http://www.xilinx.com/sxpresso/webpack.htm>

Production Environment

In a production environment, ATEs or Third Party Programmers are more common. In general, these tools supply faster programming times, and in many cases, supply a means to program more devices at a given time.

For this environment, a **jedec** file is normally converted into a standard vector format (**.svf**) file. This file format is a widely accepted standard by vendors and is a common way to distribute programming files. This format is an optional output of the Xilinx programmers described above in the Prototyping section. A description of how these files are created in the Xilinx programmers can be found in the ATE guides. Chapter 2, "Creating SVF files" of the ATE document listed describes in detail how the files are generated, and is located at:

<http://www.xilinx.com/appnotes/svf2dts.pdf>

SPARTAN-XL Master Serial and JTAG Combination Setup

Serial mode and JTAG mode are two most commonly used configuration modes for the Spartan-XL devices.

- **Advantages:**
 - Versatile and cost effective solution.
 - Master Serial Mode provides a simple and robust configuration mode for production, and the JTAG Mode provides In-System programming support for flexibility when configuring a chain in the Prototyping stage.
- **Hardware Used:**

Any combination of Xilinx Spartan-XL and companion 17Sxx devices.

 - For Prototyping: Xilinx Cable (Parallel III or MultiLINX).
 - For Production: Xilinx PROM and supporting Programmer /Software.
- **Software Used:**

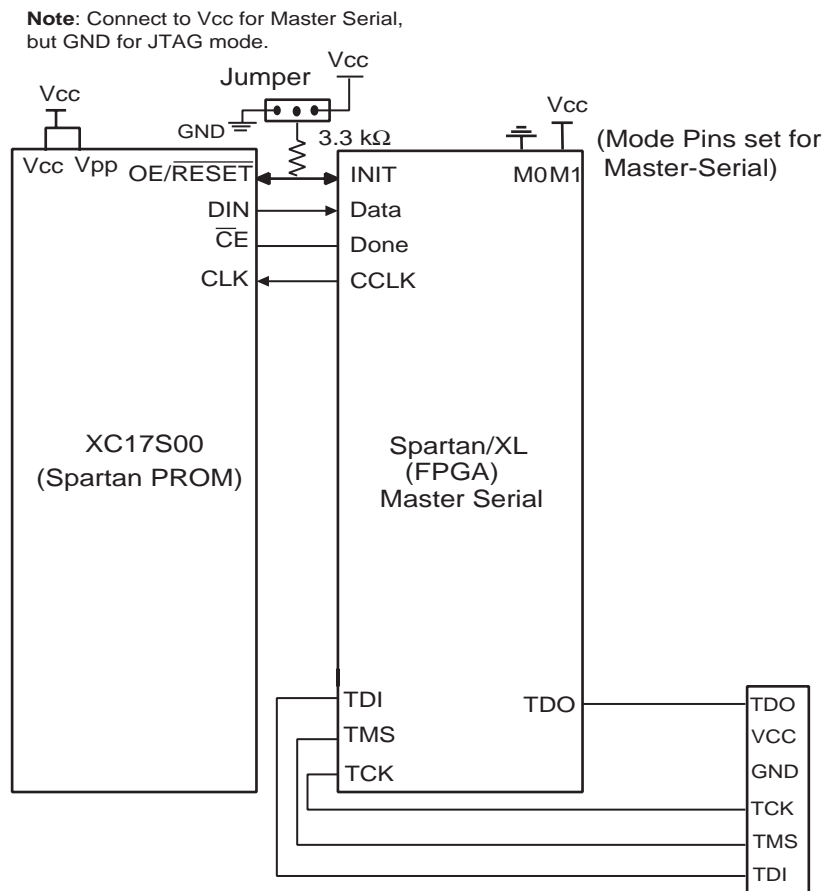
Xilinx FPGA programming bitstream (**.bit** files) (Foundation or Alliance software packages create the bitstream.)

 - For Prototyping: JTAG Programmer Software.
 - For Production: Xilinx Prom File Formatter (Included in the Foundation and Alliance software packages) to create Xilinx PROM files (**.mcs**, **.exo**), JTAG Programmer Software or appropriate Third Party Programmer software.

Hardware Considerations

The hardware connections used to implement the Master Serial/JTAG Mode combination are shown in **Figure 8**. Hardware items that the user should be aware of are listed below:

1. When configuring with Boundary Scan/JTAG mode, it is necessary to hold INIT pin low on power up to allow the device to enter the JTAG mode. See **Figure 8**.
(**Note:** Connect INIT to GND for JTAG mode.)
2. The mode pins should be set for Master-Serial mode for the Spartan-XL: M0 = 0 and M1 = 1
3. For the JTAG pins, it is recommended to place a pullup of 4.7 kΩ on the TMS pin to avoid inadvertent JTAG operations.
4. For INIT and DONE pin, it is recommended to allow access to the INIT and DONE pins for initial board debugging. These two pins provide valuable status information.
5. The maximum for the JTAG clock, TCK, for the Spartan-XL is 2 MHz - 5 MHz.
6. The CCLK in Master mode has a speed of the following depending on the mode:
 - a. Slow (0.5 MHz - 1.25 MHz) (bitgen option “Configuration Rate” seen in **Figure 10**)
 - b. Fast (4 MHz - 10 MHz) (bitgen option “Configuration Rate” seen in **Figure 10**)



x501_08_071001

Figure 8: Spartan-XL Master-Serial/JTAG Mode Connections

Software Design Entry Considerations

The Spartan-XL family requires the addition of a library element to keep the JTAG pins from being converted to regular I/O after configuration.

- For the Spartan-XL family, the JTAG pins are dual purpose. If JTAG operations are needed after initial device configuration for reconfiguring the device or for other JTAG functions, the BSCAN symbol is required to be placed in the design. This allows the JTAG pins to be dedicated and not converted to regular I/O after configuration, which is the default behavior. For details on instantiating the BSCAN symbol, refer to: http://toolbox.xilinx.com/docsan/3_1i/data/common/jtg/dppf/appf.htm

For a schematic example, the library element is shown in Figure 9. The four JTAG pins (TMS, TCK, TDI, and TDO) are the only pins that are necessary to connect to maintain the JTAG pins. The TDO1, TDO2, DRCK, SEL1, SEL2, and IDLE are special-purpose pins for custom logic and are recommended to be left unconnected.

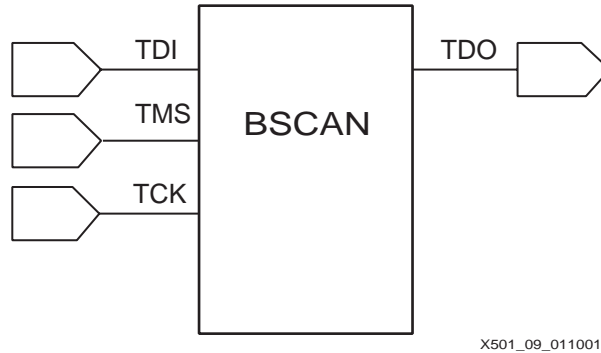


Figure 9: Boundary Scan Spartan-XL Schematic Symbol

Software Implementation Considerations

After the hardware setup has been established, the software bitstreams should be created to prepare for programming the devices. When using this configuration setup with the Spartan-XL devices, the following software implementation options (in Figure 10) need to be considered when generating the bitstream:

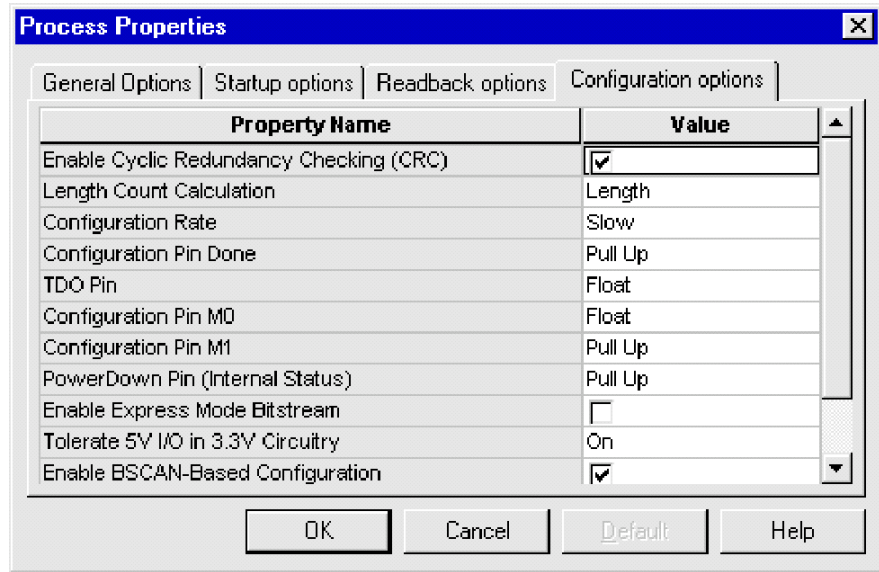


Figure 10: Spartan-XL Configuration Options in the Foundation ISE Software Package

The Spartan-XL configuration options are the same for all of the Xilinx software packages. The Spartan-XL default settings are recommended for this example. Two important options are:

- “Configuration Rate” – is an implementation option that controls the speed of the CCLK in Master Serial mode. The speeds are described in the “Hardware Considerations” section above.
- “Enable BSCAN-Based Configuration” – is an implementation option that, when disabled, inhibits the BSCAN-based configuration after the device is successfully configured. This feature allows board testing without the risk of reconfiguring the Spartan-XL device by toggling the TCK/TMS/TDI/TDO lines.

For more detailed information or information on command line options, refer to the [Development Systems Reference Guide](#).

Software Download Considerations

After the bitstream has been created, the download stage follows.

Prototyping

In this setup for prototyping, the Xilinx JTAG Programmer is commonly used. The Xilinx JTAG Programmer is comprised of a stand-alone downloadable software module, which requires a Xilinx cable to download the bitstream from the PC to the device. To achieve this, access to the four JTAG pins of the Spartan-XL device is required.

- Use the Xilinx Programming software, JTAG Programmer, to download the bitstream to the part
- Always be certain to use the latest version of the software.
<http://www.xilinx.com/sxpresso/webpack.htm>

Production

In the production environment, ensure that systems are not affected by power glitches or power down situations. The FPGA devices are volatile and a power outage erases the device contents. In order to ensure that the programmed data is not lost when the system is shut off, it is common to use a mode like Master Serial. The configuration data is permanently stored in the PROM in this example. Therefore, if a power outage occurs, the data in the PROM simply reconfigures the FPGA when power is regained. Below are the steps needed to program a XC17S00 PROM.

- Use the PROM File Formatter (GUI version) or PROMgen (command line version) to format the bitstream into a PROM file.

For the PROM File Formatter/PROMgen, go to: http://toolbox.xilinx.com/docsan/3_1i/

- Use a third party programmer to program the PROM. See the website at: <http://www.xilinx.com/support/programr/ps.htm>

Virtex Series or Spartan-II Master Serial and Boundary Scan (JTAG) Combination Setup

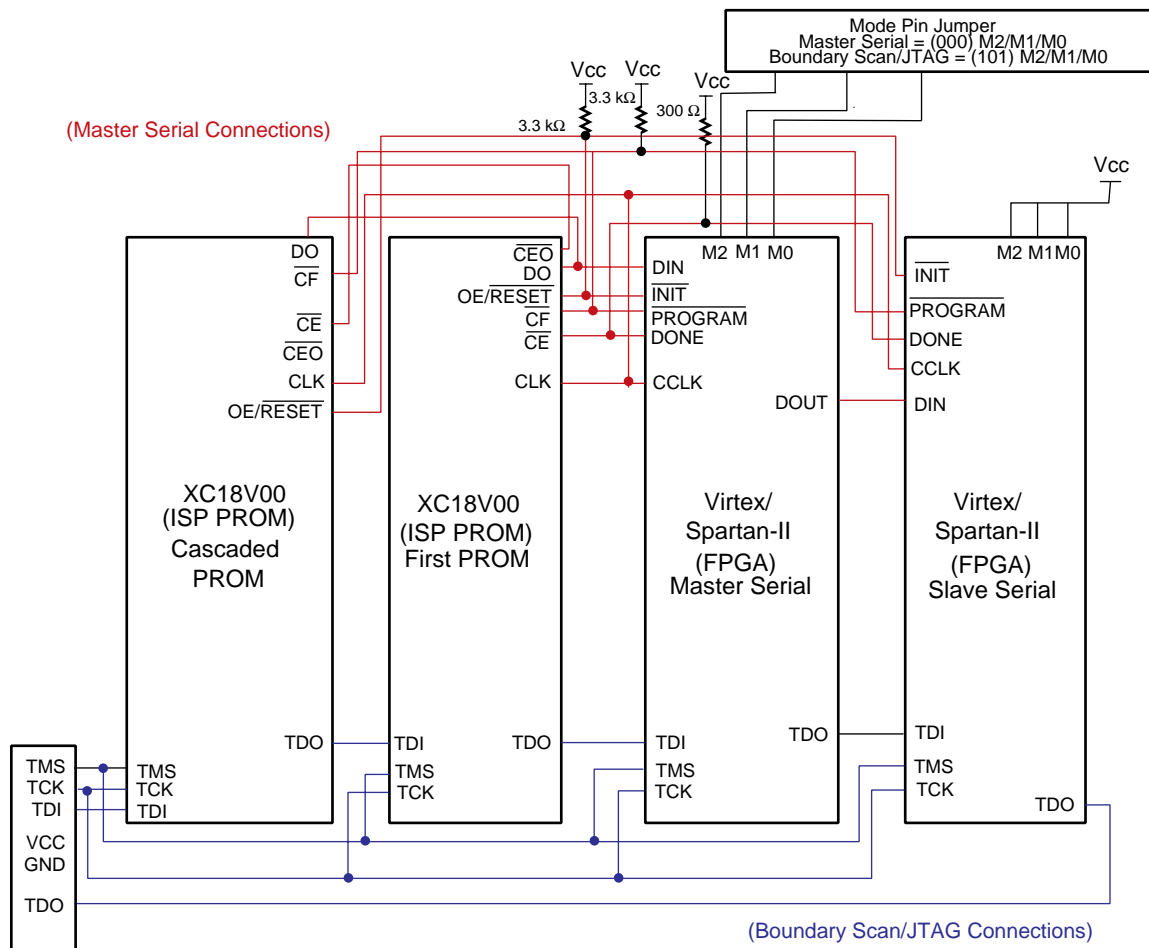
Master Serial mode and Boundary Scan (JTAG) mode are two of the most commonly used configuration modes for the Virtex Series and Spartan-II devices.

- **Advantages:**
 - The combination of the Master-Serial mode with the Boundary Scan (JTAG) mode gives the user a versatile setup which allows for easy debug and multiple configuration methods.
 - Master Serial mode provides a simple and robust configuration mode for production, and the Boundary Scan (JTAG) mode provides in-system programming support for flexibility when configuring a chain in the prototyping stage.

- **Hardware Used:** Any combination of Xilinx Virtex/Virtex-E or Spartan-II devices and companion 18Vxx companion devices.
 - For Prototyping: Xilinx Cable (Parallel III or MultiLINX)
 - For Production: Xilinx PROM and supporting Programmer /Software
- **Software Used:** Xilinx FPGA programming bitstream (.bit files) (Foundation or Alliance software packages create the bitstream.)
 - For Prototyping: JTAG Programmer Software
 - For Production: Xilinx Prom File Formatter (Included in the Foundation, Alliance software packages) to create Xilinx PROM files (.mcs, .exo), JTAG Programmer Software, or appropriate Third Party Programmer software.

Hardware Considerations

The hardware connections to implement the Master Serial/Boundary Scan (JTAG) Mode Combination are shown in Figure 11 for the Virtex Series or Spartan-II families.



x501_11_20501

Figure 11: Virtex/Spartan-II Master Serial and Boundary Scan (JTAG) Mode Connections

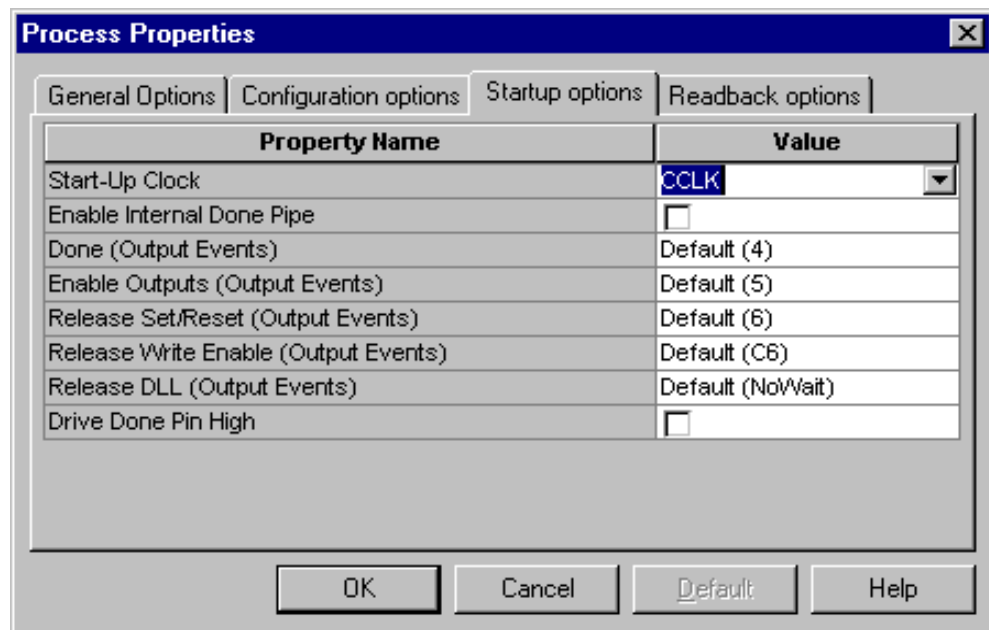
Additional hardware considerations are listed below:

- Refer to the appropriate family data sheets for voltage values required for each part.
- For Spartan-II or Virtex Series mode pins, the following settings are required:
 - Master-Serial mode – M0 = 0, M1 = 0, M2 = 0
 - Boundary Scan (JTAG) mode – M0 = 1, M1 = 0, M2 = 1

- Boundary Scan (JTAG) pins – For both Virtex and Spartan-II devices, internal programmable weak pullups are provided by default on the TMS, TCK, and TDI pins. TDO is floating.
- Output buffer can source up to 24 mA and sink up to 48 mA
- The maximum Boundary Scan (JTAG) clock, TCK, for the Virtex and Spartan-II devices is 33 MHz.

Software Implementation Considerations

Once the hardware setup has been established, the software bitstreams need to be created to prepare for programming the devices. When using this configuration setup with the Virtex or Spartan-II devices, the following software implementation options need to be considered when generating the bitstream. See [Figure 12](#).



x501_12_010301

Figure 12: Startup Options for Virtex or Spartan-II

The above defaults for the startup options should be used with the exception of the “Start-Up Clock.” This should be changed to “JTAGCLK” when creating a bitstream for a Boundary Scan (JTAG) download.

- Start-Up Clock – The bitstream must be generated with the appropriate startup clock option for the part to be configured properly. The “Start-Up Clock” option by default is set to “CCLK” for Master Serial Mode. When generating a bitstream for Boundary Scan (JTAG) Mode the option must be set to “JTAGCLK” in the pull-down menu of the GUI or using bitgen’s command line:
- For configuring using Boundary Scan (JTAG):

```
bitgen -g startupclk:jtagclk designName.ncd
```
- For configuring via Master-Serial:

```
bitgen -g startupclk:cclk designName.ncd
```

Notes:

1. If a setup is using these two configuration modes, two bitstreams must be generated, one for the Boundary Scan (JTAG) download and the other for the Master Serial download.

The Virtex or Spartan-II configuration options are the same for all of the Xilinx software packages. The default configuration option settings are recommended for this example, and some of the important options are described below:

- “Configuration Rate” is an implementation option which controls the speed of the CCLK in Master Serial Mode.
- The configuration options include programmable pull ups on the mode pins, PROGRAM, DONE, and CLK. The Boundary Scan (JTAG) pins should be left as default.

For more detailed information or information on command line options, refer to the [Development Systems Reference Guide](#).

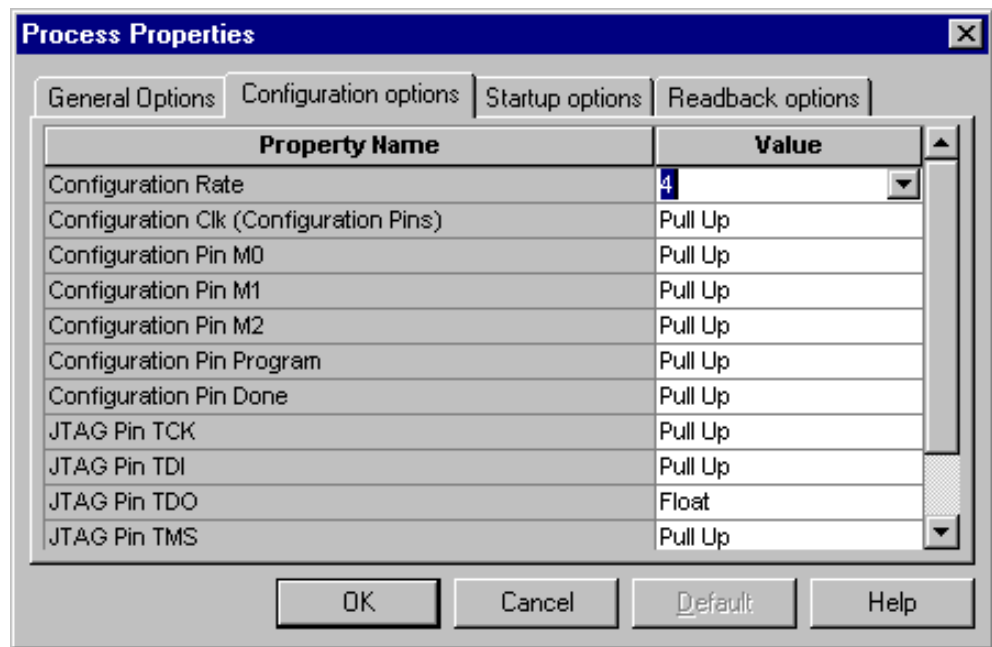
Software Download Considerations

After the bitstream has been created, follow the recommended programming options for this setup as discussed below. See [Figure 13](#).

Prototyping

For prototyping with this setup, the Xilinx JTAG Programmer is commonly used. The Xilinx JTAG Programmer is comprised of a stand-alone downloadable software module, which requires a Xilinx cable to download the bitstream from the PC to the device. To achieve this, access to the four Boundary Scan (JTAG) pins of the Virtex, Spartan-II, or the XC18V00 device is required.

- Use the Xilinx JTAG Programmer to download the bitstream to the part. Always be certain to use the latest software version at: <http://www.xilinx.com/sxpresso/webpack.htm>



x501_13_010301

Figure 13: Configuration Options for Virtex or Spartan-II Devices

Production

In the production environment, it is important that systems are not affected by power glitches or power down situations. Because the FPGA devices are volatile, a power outage erases the device contents. In order to ensure that the programmed data is not lost when the system is shut off, it is recommended to use a mode like Master Serial. The configuration data is permanently stored in the PROM in this example. If a power outage occurs, the data in the PROM reconfigures the FPGA when power is regained. Below are the steps needed to program a XC18V00 PROM.

- Use the PROM File Formatter (GUI version) or PROMgen (command line version) to format the bitstream into a PROM file. For the PROM File Formatter/PROMgen, see the website at: http://toolbox.xilinx.com/docsan/3_1i/
- An ATE/Third Party Programmer or JTAG Programming Tool can be used to download to the XC18V00 in system.

Support

For more information about configuration, there are a number of Xilinx-specific application notes, checklists, and websites that are helpful.

Configuration Application Notes

General:

- Configuring Xilinx FPGAs Using an XC9500 CPLD and Parallel PROM
<http://www.xilinx.com/xapp/xapp079.pdf>
- Xilinx In-System Programming Using an Embedded Microcontroller
<http://www.xilinx.com/xapp/xapp058.pdf>

Virtex Series:

- Using the Virtex SelectI/O Resource (including details on termination)
<http://www.xilinx.com/xapp/xapp133.pdf>
 - Virtex Configuration Architecture User Guide
<http://www.xilinx.com/xapp/xapp151.pdf>
 - Virtex FPGA Configuration and Readback
<http://www.xilinx.com/xapp/xapp138.pdf>
 - Configuring Virtex FPGAs from Parallel EPROMs with a CPLD
<http://www.xilinx.com/xapp/xapp137.pdf>
- Configuration and Readback of Virtex FPGAs Using (JTAG) Boundary-Scan
<http://www.xilinx.com/xapp/xapp139.pdf>

Spartan:

- Configuring Spartan-II FPGAs from Parallel EPROMs
<http://www.xilinx.com/xapp/xapp178.pdf>
- Spartan-II FPGA Family Configuration and Readback
<http://www.xilinx.com/xapp/xapp176.pdf>
- Boundary-Scan in XC4000, Spartan, and XC5200 Series Devices
<http://www.xilinx.com/xapp/xapp017.pdf>
- The Low-Cost, Efficient Serial Configuration of Spartan FPGAs
<http://www.xilinx.com/xapp/xapp098.pdf>

XC9500/XL:

- Planning for High Speed XC9500XL Designs
<http://www.xilinx.com/xapp/xapp115.pdf>
- Designing CPLD Multi-Voltage Systems
<http://www.xilinx.com/xapp/xapp144.pdf>
- A Quick JTAG ISP Checklist
<http://www.xilinx.com/xapp/xapp104.pdf>

XPLA3:

- In-System Programming of XPLA3 Devices
<http://www.xilinx.com/xapp/xapp343.pdf>

PROM:

- XC1700 and XC18V00 Design Migration Considerations
<http://www.xilinx.com/xapp/xapp161.pdf>

Configuration Checklist

Before designing a board with FPGAs/CPLDs, Xilinx recommends that consideration be given to the following issues concerning configuration.

Hardware:

- **Boundary Scan (JTAG) Related**

- If there are multiple FPGAs/CPLDs/(ISP)PROMs in a configuration chain, consider adding jumpers to isolate the devices to allow each to be loaded separately for debugging.
- Be sure to treat configuration or Boundary Scan (JTAG) clocks like any other high-speed clock.
- Consider adding extra spare pins to the design to bring signals out for later debugging.
- Place buffers on TMS or TCK signals interleaved at various points on the Boundary Scan (JTAG) circuitry to account for unknown device impedance.
- Put the rest of the JTAG chain into HIGHZ mode by selecting the HIGHZ preference on JTAG Programmer when having difficulty programming a particular part. This reduces any noise seen by the part.
- If free running clocks are delivered into a Boundary Scan (JTAG) device, it can be necessary to disconnect or disable their entry into these devices during ISP or Boundary Scan (JTAG) operations.

- **Cable Related:**

- Do not attach extension cables to the target system side of the cable, because this can compromise configuration data integrity and cause checksum errors.
- Attach cable configuration leads firmly to the target system.
- The Parallel Cable used with PCs draws less power from the target LCA board than the MultiLINX cable. This is because the MultiLINX cable utilizes an embedded microprocessor and has additional features like SelectMAP/USB support that the parallel cable does not have.
- Design for the ability to configure using a cable, as well as your selected production configuration method. This allows for programming options and a method for debugging the part easily.

- **Board Related:**

- Include the ability to set the mode pins to different values using jumpers.
- Remember signal integrity issues, termination, edge rates, ground bounce, and trace layout.
- For designs over 60 MHz, Xilinx recommends that simulation of board level switching be done to ensure there are no reflection or ground bounce issues.
- Keep in mind simultaneous switching output issues.
- Ensure that the ground pins are not overloaded to prevent ground bounce on the outputs.
- Unused I/Os can be set as outputs tied to ground to add extra grounding to a device.
- If there is space, add LEDs to the DONE pins so that it can be determined whether or not the FPGAs were configured.
- Put test points on the key configuration pins (DOUT, INIT, DONE, and PROGRAM) for debug and status information.
- Do not overload configuration signals. Check the Fan-out/Fan-in loading of the configuration signals.
- Use the Output Enable on your clock source. This can allow stopping the system clocks during configuration to check for cross-talk and other noise issues
- Be aware of switching noise that might cause erase time or programming time extension as the design progresses, especially for long chains.

- Make sure the V_{CC} is within the rated value for devices being used.
- Provide both 0.1 μ F and 0.01 μ F capacitors at every V_{CC} point of the chip, and attach them directly to the nearest ground.
- **Software:**
 - For troubleshooting JTAG Programmer issues, refer to the following:
http://toolbox.xilinx.com/docsan/3_1i/data/common/jtg/dppc/appc.htm
 - Xilinx WEBPACK has the latest JTAG Programmer software and can be used as a lab install for acquiring the download software without installing the entire software suite of tools at: <http://www.xilinx.com/products/software/webpowered.htm#webpack>
 - Read the documentation on the configuration options in the Foundation/Alliance 3.1i software at: http://toolbox.xilinx.com/docsan/3_1i/
 - Consider programmable pull up and pull down needs I/O pins that are set in the software for configuration.
 - Check if release of DONE after or before the DLL has locked on the system clock.
 - Do not put regular design signals on configuration pins.
 - The STARTUP block is not recommended for Virtex /Spartan-II designs. If using this feature, be sure the internal signals do not conflict and lock up the part after configuration. The STARTUP reset polarity and the regular system reset needs to be of the same polarity.
 - Check for system noise by running the IDCODE looping instruction in JTAG Programmer "Operations -> Idcode looping". This displays the edit window. Specify the number of loops desired. The operation should complete this correctly (100 percent); otherwise, there might be system noise.
 - Use the latest version of Xilinx Parallel III cable, which has a serial number greater than 5000.

Configuration Problem Solvers

The following software tools step the user through configuration problems based on the inputs the user provides the tool. Go to:

<http://www.xilinx.com/support/troubleshoot/psolvers.htm>

Conclusion

Configuration is a very important, but often an under-emphasized step when using PLDs. Understanding the options and using the guidelines in this application note should help to make the process of programming or configuring a PLD easier.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
02/14/01	1.0	Initial Xilinx release
02/26/01	1.1	Minor edits done.
08/02/01	1.2	Minor edits done.
06/10/02	1.3	Added "Virtex Series" and "Virtex-II Series Platform FPGAs."