XAPP693 (v1.1) January 19, 2005

A CPLD-Based Configuration and Revision Manager for Xilinx Platform Flash PROMs and FPGAs

Author: Don St. Pierre

Summary

This application note illustrates the use of a Xilinx CoolRunner™-II CPLD to monitor configuration data between a Xilinx Platform Flash Configuration PROM and a Xilinx Spartan™ or Virtex™ family FPGA. The intent is to ensure reliable configuration of the FPGA while providing revision control for one or more configuration files stored in the PROM.

This document assumes the reader is familiar with Xilinx FPGA configuration methodology. Please refer to applicable Spartan or Virtex family data sheets and/or user guides for a complete discussion of configuration requirements and procedures.

Introduction

The Xilinx Platform Flash PROMs offer a host of features, including an internal clock source and revision control for up to four pages of configuration data. They provide an easy to use, cost effective and reprogrammable medium for sharing larger Xilinx FPGA configuration bitstreams, and are available in densities from 1 Mbit to 32 Mbit. See DS123, “Platform Flash In-System Programmable Configuration PROMs,” for more details.

The reference design presented in this application note provides access control to the PROM's revisions, and ensures reliable configuration by monitoring configuration status signals from a Xilinx FPGA, such as INIT and DONE. In the event of a configuration failure, the Configuration Monitor reverts back to the “known-good” PROM file in Page 00.

Figure 1 shows a functional block diagram of the Platform Flash PROM.

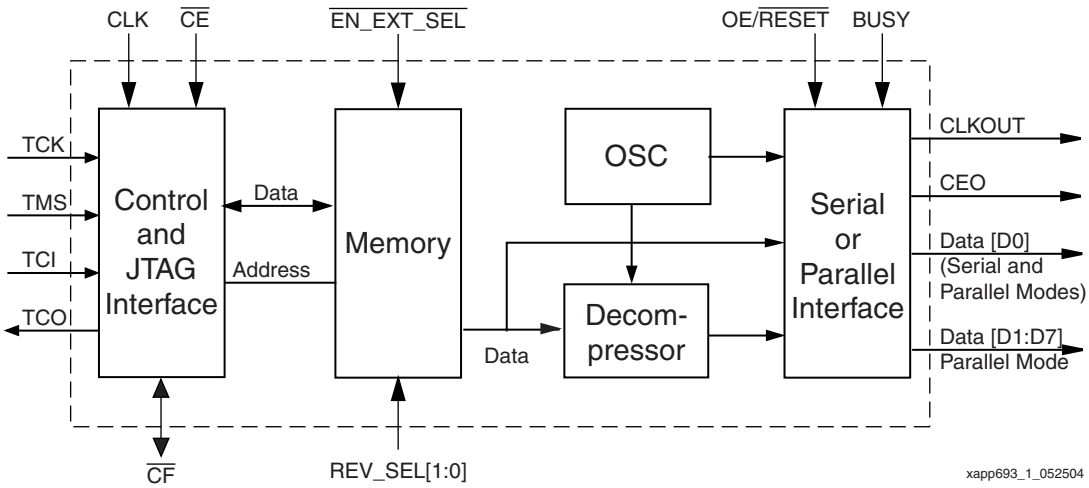

Figure 1: **Platform Flash PROM Block Diagram**

Configuration

Initialization and Timing

Figure 2 shows the configuration process flow for a Virtex-II Pro FPGA.

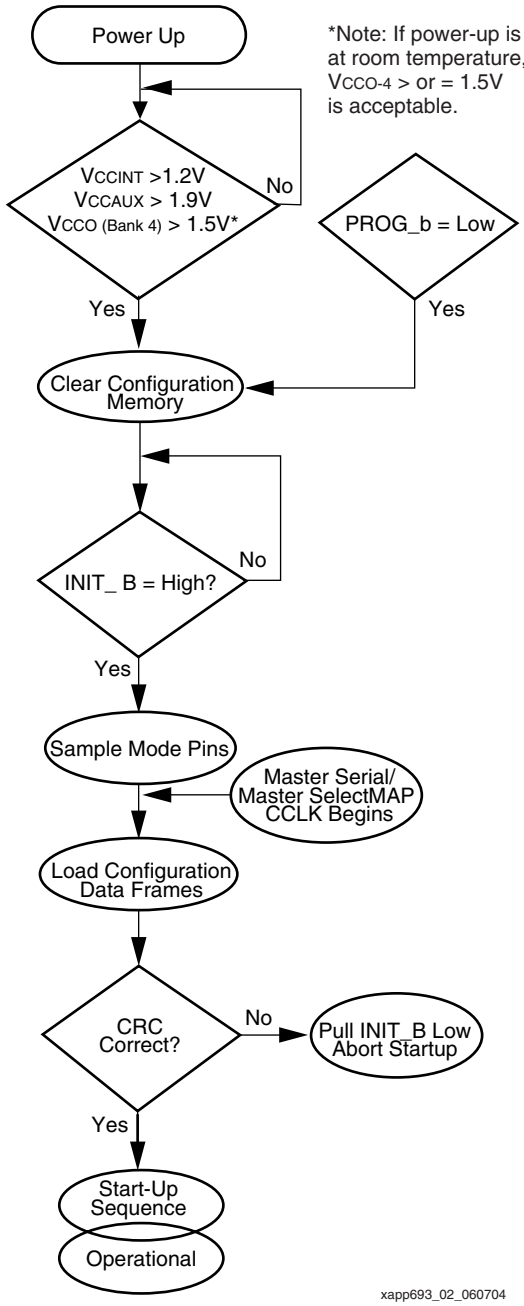

Figure 2: **Configuration Process**

Upon power-up, an FPGA's $\overline{\text{INIT}}$ pin is internally held Low while the FPGA initializes its internal circuitry and clears its configuration memory. Configuration will start as soon as the $\overline{\text{INIT}}$ pin transitions High, either after power-up or after the FPGA's $\overline{\text{PROGRAM}}$ pin is held Low and released.

The $\overline{\text{PROGRAM}}$ pin must be held low for a preset amount of time ($T_{PROGRAM}$). Refer to individual FPGA data sheets for this and other FPGA timing values. Figure 3 shows a representative example of Virtex-II Pro power-up configuration timing, and Table 1 shows the associated timing characteristics.

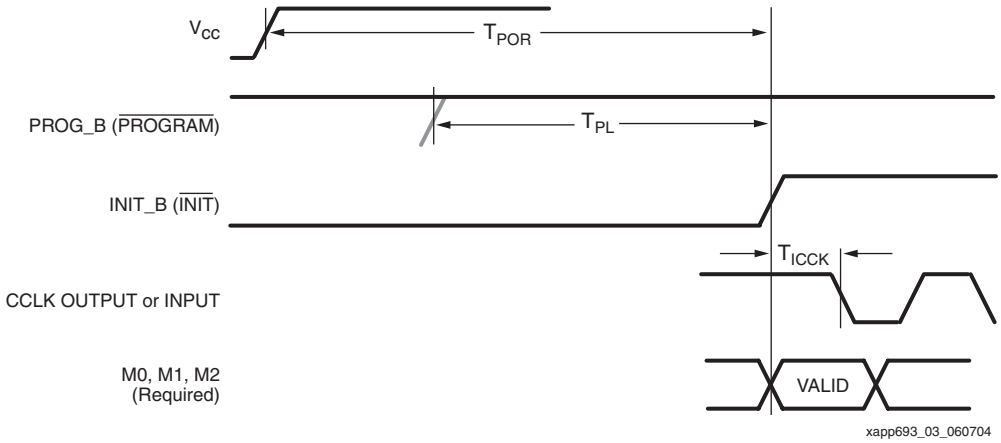

Figure 3: **Power-Up Timing Configuration Signals**

Table 1: **Power-Up Timing Characteristics**

Description	Symbol	Value	Units
Program Latency	T_{PL}	4	µs per frame, max.
Power-on Reset	T_{POR}	T_{PL} + 2	ms, max.
CCLK (output) Delay	T_{IOCK}		µs, min.
			µs, max.
Program Pulse Width	$T_{PROGRAM}$	300	ns, min.

Configuration Failures

Configuration consists of loading application-specific data into an FPGA's internal configuration memory. Configuration failures often occur in the form of either a failed CRC or a missed PREAMBLE.

If the INIT signal transitions from High to Low during configuration, the bitstream being sent to the FPGA fails its CRC, and the FPGA becomes "stuck" in a failed state (INIT and DONE Low). The only way to recover from this state is to pulse the FPGA's PROGRAM pin Low and initiate a reconfiguration. However, if the bitstream has somehow become corrupted, the CRC failure will likely occur again and again.

Another condition that can occur is absence of the PREAMBLE, or synchronization word, that normally signifies the beginning of configuration data. A bitstream created in byte-reversed order, for example, will not present the appropriate 32-bit synchronization word (for example, `0xAA995566`) to the FPGA. This scenario can be quite confusing because the FPGA never reports a CRC error (INIT going Low). The common misconceptions when this occurs are that the FPGA is "bad" or may require that more configuration clocks (CCLK) be sent. Rather, the most appropriate action in this scenario is to inspect the bitstream and verify that the synchronization word is correct.

Configuration Monitor Overview

The Configuration Monitor is a passive monitor that senses the FPGAs configuration status pins (*INIT* and *DONE)* and ensures reliable configuration by sensing common configuration failures. Figure 4 shows a system level block diagram of where the Configuration Monitor interfaces to the PROM and FPGA.

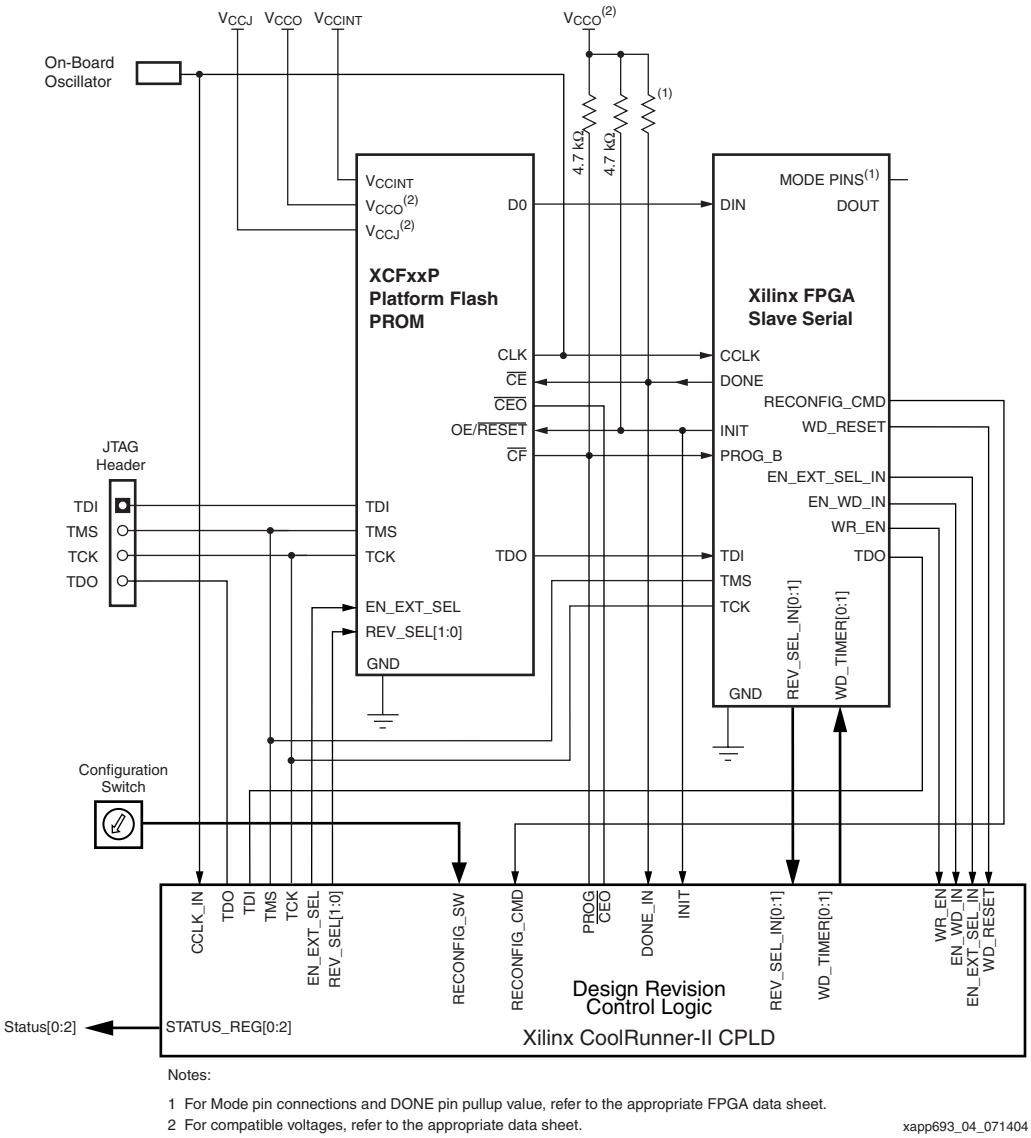

Figure 4: **System Level Block Diagram**

The Design Revision Control Logic box is where the Configuration Monitor sits. Table 2 shows the pin names and purpose.

Table 2: **Configuration Monitor Signal Names**

Pin Name	Direction	Comments
CCLK_IN	Input	10MHz Clock to drive CPLD state machine and related logic, as well as FPGA in Slave Configuration Mode
WD_RESET	Input	Input from FPGA to reset watchdog timer counter
INIT	Input	Configuration Status pin from FPGA
RECONFIG_SW	Input	Reconfiguration signal from the on-board switch to initiate reconfiguration (if installed)
RECONFIG_CMD	Input	Reconfiguration Command from the FPGA to initiate reconfiguration
DONE_IN	Input	DONE pin from FPGA
EN_EXT_SEL_IN	Input	Signal from FPGA to enable external revision control of the PlatformFlash PROM
EN_WD_IN	Input	Signal from the FPGA to enable watchdog timer operation
WR_EN	Input	Active Low write enable to control writing of the watchdog timer enable register, as well as EN_EXT_SEL and REV_SEL registers
CEO	Input	Output from the PROM. Acts as status signal when the PROM internal address counter has reached terminal count. This occurs for each bank in the PROM. If this occurs, and FPGA DONE pin has not gone high and INIT has not gone low, this signal is a simple way to determine if entire bitstream was sent to the FPGA. It saves cost of having to implement large counter inside the CPLD.
REV_SEL_IN(0 to 1)	Input	Signals from the FPGA to select which PROM revision will be used
PROG	Output	Active low PROGRAM pin on the FPGA
EN_EXT_SEL	Output	Output to control external bitstream revisioning on the PROM
WD_TIMER(0 to 1)	Output	Outputs of the watchdog timer for the FPGA to monitor
REV_SEL(0 to 1)	Output	Outputs to the PROM to select the bank revision used to reconfigure the FPGA
STATUS_REG(0 to 2)	Output	Status bit outputs showing the status of configuration attempts

Configuration Monitor Components

The Configuration Monitor reference design is modular, It contains a top-level Verilog™ file, *CONFIG_MONITOR*, with five levels of hierarchy:

- MONITOR_SM
- PROGRAM
- REVISION
- STATUS
- WATCHDOG

Each of the modules and their corresponding signals are described in the sections that follow.

Configuration Monitor Top-Level File: *CONFIG_MONITOR*

The top-level Verilog file provides connectivity between all lower level instantiated modules. A reset signal, RESET, is generated at the top level, and is sourced to multiple modules. RESET is a result of one of two conditions: RECONFIG_CMD or RECONFIG_SW. These signals are inputs to the top-level design, and provide the opportunity to restart the configuration from a board-level switch, or from a command initiated by the corresponding FPGA.

The Configuration Monitor reference design requires a free-running input clock for the configuration clock; CCLK. The signal CCLK_IN in the reference design port list is connected to the FPGA and PROM clock inputs, and driven by an on-board crystal oscillator. Having a free-running CCLK allows the reference design to be in lock-step with the configuration process. This also means that any FPGA using this application note needs to be set for Slave mode configuration. See the applicable FPGA device datasheets for configuration mode details.

Monitor State Machine: *MONITOR_SM*

The monitor state machine module is the heart of the configuration monitor design. Here, the CPLD monitors the configuration of an FPGA and controls PASS or FAIL of a given bitstream. Figure 5 shows the state diagram and transitions for the configuration monitor design.

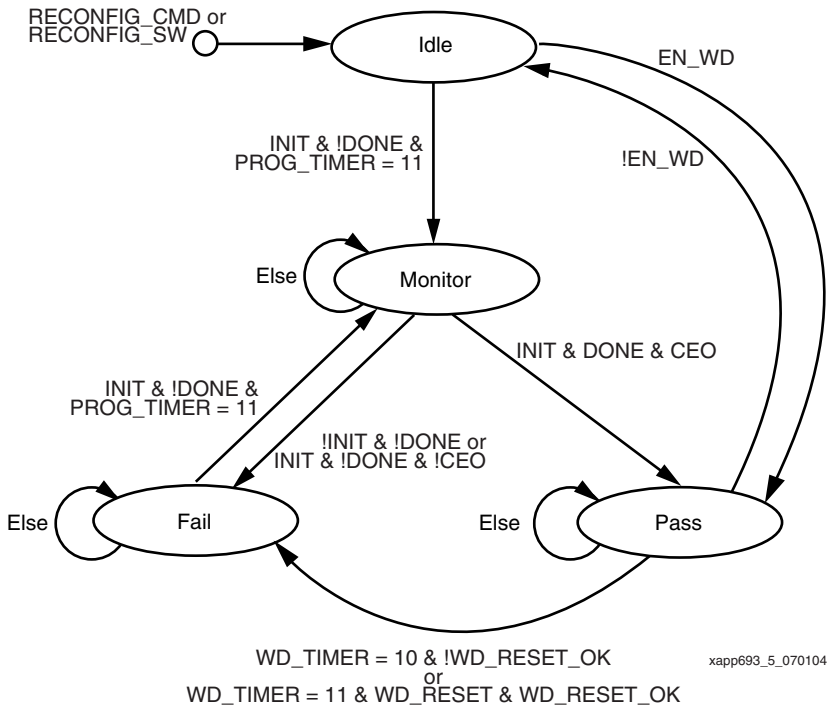

Figure 5: **Configuration Monitor Status Machine**

The state machine default and power-up state is the IDLE state. IDLE is reached when either RECONFIG_SW or RECONFIG_CMD asserts, or from the PASS state when the user selects no watchdog function control. The state machine moves the MONITOR state, to monitor the bitstream being sent to the target FPGA, once $\overline{\text{INIT}}$ transitions from LOW to HIGH.

In the MONITOR state, if $\overline{\text{INIT}}$ transitions LOW before DONE is HIGH, then a failed condition occurred and the state machine transitions to FAIL. If $\overline{\text{INIT}}$ and Chip Enable Output ($\overline{\text{CEO}}$) remain HIGH as DONE transitions from LOW to HIGH, then the state machine moves to the PASS state. The $\overline{\text{CEO}}$ pin from the PROM represents the end of the bitstream, and signifies that if $\overline{\text{INIT}}$ never transitioned from HIGH to LOW and DONE never went HIGH, then the FPGA likely never saw the sync-word.

In the FAIL state, the Configuration Monitor reference design re-asserts PROGRAM for 300 ns, per datasheet specification (Tprogram), and transition back to the MONITOR state. In the FAIL state, the REVISION module of the Configuration Monitor reference design changes the REV_SEL pins back to Page 0,0, the “known-good” bitstream. Once back in the MONITOR state, configuration commences with the new, known-good bitstream.

The PASS state enables the watchdog timer function if the user enables its operation. See “Watchdog Timer: WATCHDOG,” page 8 for details. If the user has EN_WD disabled, the state machine traverses back to IDLE. If EN_WD is enabled, the state machine stays in the PASS state, and monitors the count of the watchdog timer and reset from the FPGA. If the FPGA fails to “kick” the watchdog off (WD_RESET_OK) before a count of “01,” the design assumes failure and moves to the FAIL state to initiate a reconfiguration from the known-good bitstream. If WD_RESET_OK is true, and the FPGA continues to reset the watchdog timer (WD_TIMER) before its terminal count, the state machine stays in the PASS state.

Configuration Control: *PROGRAM*

The PROGRAM module ensures the FPGA PROG pin is held low for the minimum required time Tprogram (per the datasheet specification). See the applicable FPGA device datasheet.

A 10 MHz clock source driven into the SLOWCLK input pin of the CPLD provides a 100 ns. period. The program timer counter (PROG_TIMER) is a two-bit counter, with terminal count of 03h, or 300 ns.

The Configuration Monitor reference design consumes less than the maximum number of macrocells in a CoolRunner-II 32 macrocell CPLD. As such, this counter may be modified to allow for more bits, and a larger terminal count, if necessary.

The PROGRAM module monitors the state machine states. Since the IDLE state can see a reconfiguration command (or RECONFIG_SW), a provision had to be made to drive PROG to the FPGA from this state. If RESET transitions LOW, PROG is driven low, and the RECONFIGURE signal is driven HIGH. The RECONFIGURE signal keeps the PROG pin driven LOW, in the IDLE state, until the PROG_TIMER reaches its terminal count. If RECONFIGURE is LOW while in the IDLE state, there was no reconfiguration request, either from a switch or FPGA command, and the PROG pin stays HIGH.

Revision Select and Control: *REVISION*

The REVISION module provides control of the PROM revision select pins, as well as the PROM input EN_EXT_SEL. The EN_EXT_SEL pin to the PROM signifies external revision control. If LOW, the PROM’s internal revisions are selected based upon the state of the REV_SEL pins. The EN_EXT_SEL and REV_SEL registers are written to by driving WR_EN LOW.

The REVISION module also allows FPGA control of the EX_EXT_SEL, and REV_SEL pins. Therefore, with signals RECONFIG_CMD, EN_EXT_SEL and REV_SEL, the corresponding FPGA has complete control over which PROM revision it will use to reconfigure itself.

Status Control: *STATUS*

Configuration Monitor Status Register:

Status of the current configuration is displayed in a three-bit status register that is visible on the CPLD output pins. See Table 3 for the Configuration Monitor Status Register definition map.

Status Register Bit Codes (MSB:LSB)	Code
000	All good
001	Previous bitstream failed watchdog timer test
010	Previous bitstream failed CRC test
100	Previous bitstream failed PREAMBLE test (Timed out or the FPGA didn't see a sync word.)

Table 3: **Configuration Monitor Status Register**

Upon detection of a configuration failure, the previous state of the status register is ORed with the failing condition. This provides for a history of events until the system is reset (RECONFIG_SW) or a reconfiguration command is issued from the FPGA (RECONFIG_CMD).

For example, if an initial failure occurs because of an invalid CRC, the status register is set to 010. If, after reconfiguration from Page 00 (the known-good bitstream), the Configuration Monitor reconfigured the FPGA yet again due to failure to reset the watchdog timer, the resulting status register would then be 011. This specific condition could occur over and over, unless the user disables the watchdog timer, or reset the watchdog timer as prescribed. This is one of the primary reasons why the watchdog timer is disabled by default in the design.

If the FPGA detects a CRC on the data being sent, it drives $\overline{\text{INIT}}$ Low, and transitions to a failed state. The Configuration Monitor senses this condition, and sets the status register to 010, signifying that the CRC check failed, and begins the sequence to reconfigure the FPGA from the known-good configuration source (Page 00) in the Platform Flash PROM.

The Configuration Monitor uses the PROM's Chip Enable Output ($\overline{\text{CEO}}$) pin to determine when the internal PROM address counter has reached its terminal count. If an entire bitstream has been sent to the FPGA, and the FPGA's $\overline{\text{INIT}}$ pin never went low, and the DONE pin never went high, **and** the PROM $\overline{\text{CEO}}$ pin went low, it is probable that the FPGA never saw the bitstream PREAMBLE (sync word). In this scenario, the Configuration Monitor again initiates a reconfiguration sequence from the known-good configuration source (Page 00) in the Platform Flash PROM, and sets the status register to 100.

Watchdog Timer: *WATCHDOG*

The Configuration Monitor includes a selectable watchdog timer. The function of the watchdog timer is to ensure proper operation of the target FPGA after configuration. A counter, “WD_TIMER”, represents the count of the watchdog timer, and is visible on the CPLD output pins. If enabled, the watchdog timer will increment to a terminal count of 03h. If the target FPGA does not reset the timer before the terminal count, the Configuration Monitor reference design will initiate a reconfiguration from the known-good bitstream, Page 00 of the Platform Flash PROM.

If the user (corresponding FPGA) has EN_WD disabled, the Configuration Monitor state machine traverses back to IDLE. If EN_WD is enabled, the state machine stays in the PASS state, and monitors the count of the watchdog timer and reset from the FPGA. The EN_WD register is written to by driving WR_EN LOW.

If the FPGA fails to “kick” the watchdog off (WD_RESET_OK) before a count of 01h (100 ns.), the design assumes failure and moves to the FAIL state to initiate a reconfiguration from the known-good bitstream. If WD_RESET_OK is true, and the FPGA continues to reset the watchdog timer (WD_TIMER) before its terminal count, the state machine stays in the PASS state, and no reconfiguration is initiated.

Verilog Download

The Verilog source code and test bench is available for this design. THE DESIGN IS PROVIDED TO YOU “AS IS”. XILINX MAKES AND YOU RECEIVE NO WARRANTIES OR CONDITIONS, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, AND XILINX SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. This design has not been verified on hardware (as opposed to simulations), and it should be used only as an example design, not as a fully functional core. XILINX does not warrant the performance, functionality, or operation of this Design will meet your requirements, or that the operation of the Design will be uninterrupted or error free, or that defects in the Design will be corrected. Furthermore, XILINX does not warrant or make any representations regarding use or the results of the use of the Design in terms of correctness, accuracy, reliability or otherwise.

The reference design zip file is available online at**:**

http://www.xilinx.com/bvdocs/appnotes/xapp693.zip

Conclusion

This application note was designed to support the newest family of Xilinx PROMs, the XCF00 Platform Flash PROM, and provide example of how to ensure safe configuration. The reference design demonstrates design revision and configuration control.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
07/15/04	1.0	Initial Xilinx Release
01/19/05	1.1	Removed reference to VHDL code.