Configuration Issues: Power-up, Volatility, Security, Battery Back-up

XAPP 092 November 24, 1997 (Version 1.1)

Application Note by Peter Alfke

Summary

This application note covers several related subjects: How does a Xilinx FPGA power up, and how does it react to power-supply glitches? Is there any danger of picking up erroneous data and configuration? What can be done to maintain configuration during loss of primary power? What can be done to secure a design against illegal reverse-engineering?

Xilinx Families

XC2000, XC3000, XC4000, XC5200

Power-Up

Here is a detailed description of XC3000 Series, XC4000 Series and XC5200 device behavior during supply ramp-up and ramp-down.

When V_{cc} is first applied and is still below about 3 V, the device wakes up in the pre-initialization mode. HDC is High; $\overline{INIT}$, $\overline{LDC}$ and DONE or DONE/$\overline{PROG}$ (D/$\overline{P}$) are Low, and all other outputs are 3-stated with a weak pull-up resistor.

When V_{CC} has risen to a value above ~3 V, and a 1 and a 0 have been successfully written into two special cells in the configuration memory, the initialization power-on time delay is started. This delay compensates for differences in V_{CC} detect threshold and internal CCLK oscillator frequency between different devices in a daisy chain. The initialization delay counts clock periods of an on-chip oscillator (CCLK) which has a 3:1 frequency uncertainty depending on processing, voltage and temperature. Time-out, therefore, takes between 11 and 33 ms for a slave device, four times longer for a master device.

This factor of four makes sure that even the fastest master will always take longer than any slave. We assume that the worst- case difference between 33 ms and 4 x 11 ms is enough to compensate for the V_{CC} rise time spent between threshold differences (max 2 V) of devices in a daisy chain. Only in cases of very slow V_{CC} rise time (>25 ms), must the user hold RESET Low until V_{CC} has reached a proper level. Interconnecting the $\overline{INIT}$ pins of all devices in a daisy-chain is a better method of synchronizing start-up, but cannot be used with XC2000 devices, since they lack an $\overline{INIT}$ pin.

After the end of the initialization time-out, each device clears its configuration memory in a fraction of a millisecond, then tests for inactive $\overline{RESET}$ or $\overline{PROGRAM}$, stores the MODE value and starts the configuration process, as described in the Data Sheet. After the device is configured, the 5-V V_{CC} may dip to about 3.5 V without any significant consequences beyond an increase in delays (circuit speed is proportional to V_{cc}), and a reduction in output drive. If V_{cc} drops into the 3-V range, it triggers a sensor that forces the device back to the pre-initialization mode described above. All flip-flops are reset, HDC goes High; $\overline{INIT}$, $\overline{LDC}$ and D/$\overline{P}$ or DONE go Low, and all other outputs are 3-stated with a weak resistive pull-up. If V_{CC} dips substantially lower, the active outputs become weaker, but the device stays in this preinitialization mode. When V_{CC} rises again, a normal configuration process is initiated, as described above.

Sensitivity to V_{CC} Glitches

The user need not be concerned about power supply dips: The XC3000/XC4000/XC5200 devices stay configured for small dips and they are “smart enough” to reconfigure themselves (if a master) or to ask for reconfiguration by pulling $\overline{INIT}$ and D/$\overline{P}$ or DONE Low (if a slave). The devices will not lock up; the user can initiate re-configuration at any time just by pulling D/$\overline{P}$ or $\overline{PROGRAM}$ Low or, if D/$\overline{P}$ is Low, by forcing a High-to-Low transition on RESET.

Any digital logic device with internal data storage in latches or flip-flops is sensitive to power glitches. This includes every RAM, microprocessor, microcontroller, and peripheral circuit. Only purely combinatorial circuits can be guaranteed to survive a severe power glitch without any problem.

Xilinx SRAM-based FPGAs store their configuration in latches that lose their data when the supply voltage drops below a critical value (which is substantially below 3 V for the 5-V devices), but configuration data is extremely robust and reliable while V_{CC} stays above 3 V. All Xilinx configuration latches are implemented as cross-coupled complementary inverters with active pull-down n-channel transistors and active pull-up p-channel transistors. Both High and Low logic levels have an impedance of less than 5kΩ with respect to their respective supply rail.

Typical SRAM memory devices use passive poly-silicon pull-up resistors with an impedance of about 5,000 MΩ. A

current of one nanoamp (!) is sufficient to upset the typical SRAM cell, whereas it takes a million times more current to upset the Xilinx configuration latch.

This does not mean that SRAMs are unreliable, it just shows that the levels in Xilinx configuration latches are six orders of magnitude more resistant to upsets caused by external events, like cosmic rays or alpha particles. Xilinx has never heard about any occurrence of a spontaneous change in the configuration store in any of its ~50 million FPGA devices sold over the past twelve years.

Whereas most digital circuits rely on V_{cc} staying within specification, Xilinx FPGAs have an internal voltage monitoring circuit. For example, in the 5-Volt devices, whenever the supply voltage dips below 3 V, the internal monitoring circuit causes the Xilinx FPGA to stop normal operation. All outputs go 3-state, and the device waits for the supply voltage to rise closer to 4 V, when it either demands (slave or peripheral mode) or initiates (master mode) a reconfiguration. In the range between 5.5 and 3 V, all typical CMOS devices maintain their functionality and their data storage, they just get slower as the voltage goes down.

Xilinx has made sure that the FPGA cannot be corrupted by a power glitch. The most sensitive circuit is the low-voltage detector. It kicks in while all other configuration storage and user logic is still guaranteed to be functional. The voltage-monitoring feature in the Xilinx device can even be used to protect other circuitry, or it can be coordinated with external monitoring circuits.

There is no possibility of a V_{CC} dip causing the device to malfunction, i.e., to operate with erroneous configuration information.

- If V_{CC} stays above the trip point, the device functions normally, albeit at reduced speed, like any other CMOS device.
- If V_{CC} dips below the trip point, the device 3-states all outputs and waits for reconfiguration.

Xilinx production-tests the V_{CC}-dip tolerance of all XC3000 devices in the following way.

After the device is configured, V_{CC} is reduced to 3.5 V, and then raised back to 5.0 V. Configuration data is then read back and compared against the original configuration bit stream. Any discrepancy results in rejection of the device.

Subsequently, V_{CC} is reduced to 1.5 V and then raised to 5.0 V. The device must first go 3-state, then respond with a request for reconfiguration.

Both these tests are performed at high temperature (>85°C for commercial parts, >100°C for military). Any part failing any of these tests is rejected as a functional failure.

As a result of these careful precautions, we contend that Xilinx FPGAs are safer than all other types of circuitry (except purely combinatorial circuits). A microprocessor can loose the content of its address register, its accumulator or other control register due to an undetected power glitch, with disastrous consequences to the subsequent operation. A Xilinx FPGA detects the power glitch and always plays it safe by flagging the problem.

No complex system of any kind can function reliably when V_{cc} is unreliable. Xilinx FPGAs do the safest thing possible, whenever such problems occur.

Design Security

Some Xilinx customers are concerned about the security of their designs. How can they protect their designs against unauthorized copying or reverse-engineering?

We must distinguish between two very different situations:

- Configuration data in accessible from a serial or parallel EPROM or in a microprocessor's memory. This is the normal case.
- Configuration data is hidden from the user, since the design does not permanently store a source of configuration data. After the FPGA was configured, the EPROM or other source was removed from the system, and configuration is kept alive in the FPGA through battery-back-up.

Design Security when Configuration Data is Accessible

In the first case, it is obviously very easy to make an identical replica of the design by copying the configuration data and the pc-board interconnect pattern of the standard devices, but it is virtually impossible to interpret the bitstream in order to understand the design or make intelligent modifications to it. Xilinx keeps the interpretation of the bitstream a closely guarded secret. Reverse-engineering an FPGA would require an enormously tedious analysis of each individual configuration bit, which would still only generate an XACT view of the FPGA, not a usable schematic.

The best protection against a mindless copy is legal. The bitstream is easily protected by copyright laws that have proven to be more successfully enforced than the intellectual property rights of circuit designs.

The combination of copyright protection, and the almost insurmountable difficulty of creating any design variation for the intended function, provides good design security. The recent successes of small companies in reverse-engineering microprocessors and microprocessor support circuits show that a non-programmable device can actually be more vulnerable than an FPGA. For advice on legal protection of the configuration bitstream, see the following paragraphs.

Legal Protection of Configuration Bit-Stream Programs

The bit-stream program loaded into the FPGA may qualify as a “computer program” as defined in Section 101, Title 17 of the United States Code, and as such may be protectable under the copyright law. It may also be protectable as a trade secret if it is identified as such. We suggest that a user wishing to claim copyright and/or trade secret protection in the bit stream program consider taking the following steps.

Place an appropriate copyright notice on the FPGA device or adjacent to it on the PC board to give notice to third parties of the copyright. For example, because of space limitations, this notice on the FPGA device could read “©1996 XYZ Company” or, if on the PC board, could read “Bit Stream ©)1996 XYZ Company”.

File an application to register the copyright claim for the bit-stream program with the U.S. Copyright Office.

If practicable, given the size of the PC board, notice should also be given that the user is claiming that the bit- stream program is the user's trade secret. A statement could be added to the PC board such as: “Bit-stream proprietary to XYZ Company. Copying or other use of the bitstream program except as expressly authorized by XYZ Company is prohibited.”

To the extent that documentation, data books, or other literature accompanies the FPGA-based design, appropriate wording should be added to this literature providing third parties with notice of the user's claim of copyright and trade secret in the bit-stream program. For example, this notice could read: “Bit-Stream©)1996 XYZ Company. All rights reserved. The bit-stream program is proprietary to XYZ Company and copying or other use of the bit- stream program except as expressly authorized by XYZ Company is expressly prohibited.”

To help prove unauthorized copying by a third party, additional nonfunctional code should be included at the end of the bit-stream program. Therefore, should a third party copy the bit-stream program without proper authorization, if the non-functional code is present in the copy, the copier cannot claim that the bit-stream program was independently developed.

These are only suggestions, and Xilinx makes no representations or warranties with respect to the legal effect or consequences of the above suggestions. Each user is advised to consult legal counsel with respect to seeking protection of a bit-stream program and to determine the applicability of these suggestions to the specific circumstances.

If the user has any questions, contact the Xilinx legal department at 408-879-4984.

Design Security by Hiding the Configuration Data

If the design does not contain the source of configuration data, but relies on battery-back-up of the FPGA configuration, then there is no conceivable way of copying this design. Opening up the package and probing thousands of latches in undocumented positions to read out their data without ever disturbing the configuration is impossible.

This mode of operation offers the ultimate design security. It is being used by several Xilinx customers who have reason to be concerned about illegal pirating of their designs.

Battery Back-up and Powerdown

Since SRAM-based FPGAs are manufactured using a high-performance low-power CMOS process, they can preserve the configuration data stored in the internal static memory cells even during a loss of primary power. This is accomplished by forcing the device into a low-power non-operational state, while supplying the minimal current requirement of V_{CC} from a battery.

Circuit techniques used in XC3100, XC4000 and XC5200 devices prevent I_{CC} from being reduced to the level needed for battery back-up. Consequently, battery back-up should only be used for XC2000, XC2000L, XC3000, XC3000A and XC3000L devices.

There are two primary considerations for battery backup which must be accomplished by external circuits.

- Control of the Power-Down ($\overline{PWRDWN}$) pin
- Switching between the primary V_{CC} supply and the battery.

Important considerations include the following.

- Insure that $\overline{PWRDWN}$ is asserted logic Low prior to V_{CC} falling, is held Low while the primary V_{CC} is absent, and returned High after V_{CC} has returned to a normal level. $\overline{PWRDWN}$ edges must not rise or fall slowly.
- Insure “glitch-free” switching of the power connections to the FPGA device from the primary V_{CC} to the battery and back.
- Insure that, during normal operation, the FPGA V_{CC} is maintained at an acceptable level, 5.0 V ± 5% (±10% for Industrial and Military).

Figure 1 shows a power-down circuit developed by Shel Epstein of Epstein Associates, Wilmette, IL. Two Schottky diodes power the FPGA from either the 5.2 V primary supply or a 3 V Lithium battery. A Seiko S8054 3-terminal power monitor circuit monitors V_{CC} and pulls $\overline{PWRDWN}$ Low whenever V_{CC} falls below 4 V.

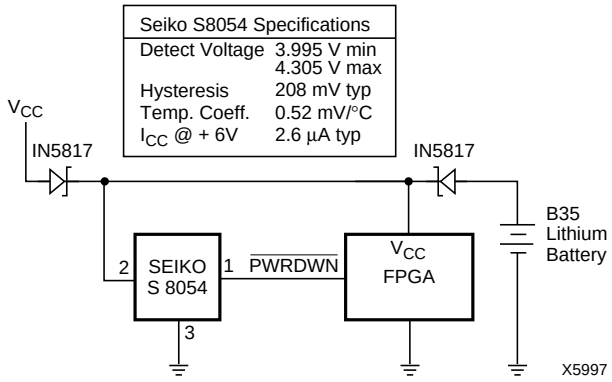

Figure 1: Battery Back-up Circuit

Powerdown Operation

A Low level on the $\overline{\text{PWRDWN}}$ input, while V_{CC} remains higher than 2.3 V, stops all internal activity, thus reducing I_{CC} to a very low level:

- All internal pull-ups (on Long lines as well as on the I/O pads) are turned off.
- The crystal oscillator is turned off
- All package outputs are three-stated.
- All package inputs ignore the actual input level, and present a High to the internal logic.
- All internal flip-flops or latches are permanently reset.
- The internal configuration is retained.
- When $\overline{\text{PWRDWN}}$ is returned High, after V_{CC} is at its nominal value, the device returns to operation with the same sequence of buffer enable and D/$\overline{P}$ as at the completion of configuration.

Things to Remember:

Powerdown retains the configuration, but loses all data stored in the device. Powerdown three-states all outputs and ignores all inputs. No clock signal will be recognized, and the crystal oscillator is stopped. All internal flip- flops and latches are permanently reset and all inputs are interpreted as High, but the internal combinatorial logic is fully functional.

Things to Watch Out for:

Make sure that the combination of all inputs High and all internal flip-flop outputs Low in your design will not generate internal oscillations or create permanent bus contention by activating internal bus drivers with conflicting data onto the same long line. These two situations are farfetched, but they are possible and will result in considerable power consumption. It is quite easy to simulate these conditions since all inputs are stable and the internal logic is entirely combinatorial, unless latches have been made out of function generators.

During powerdown, the V_{CC} monitoring circuit is disabled. It is then up to the user to prevent V_{CC} dips below 2.3 V, which might corrupt the stored configuration.

During configuration, the $\overline{\text{PWRDWN}}$ pin must be High, since configuration uses the internal oscillator. Whenever V_{CC} goes below 4 V, $\overline{\text{PWRDWN}}$ must already be Low in order to prevent automatic reconfiguration at low V_{CC}. For the same reason, V_{CC} must first be restored to 4 V or more, before $\overline{\text{PWRDWN}}$ can be made High.

$\overline{\text{PWRDWN}}$ has no pull-up resistor. A pull-up resistor would draw supply current when the pin is Low, which would defeat the idea of powerdown, where I_{CC} is only microamperes.