XAPP457 (v1.0) June 8, 2007

Powering and Configuring Spartan-3 Generation FPGAs in Compliant PCI Applications

Author: Eric Crabill

Summary

The PCI™ Local Bus Specification, Revision 3.0 (“the PCI specification”) defines a number of power and reset requirements. These requirements, when considered in an FPGA implementation, create several challenges that must be addressed for long term reliability and broad interoperability:

- Setting the V_{CCO} voltage, as related to device maximum ratings and reliability
- Determining if V_{CCO} can be powered directly from a bus power rail
- Designing V_{CCO} regulation for current and reverse current (if regulated)
- Distributing I/O placement to observe guidelines for simultaneously switching outputs
- Strapping the correct pre-configuration I/O pull-up resistor option
- Minimizing the ramp time of regulated power supplies
- Budgeting time for FPGA power-on and configuration

The information presented in this application note applies to compliant PCI applications using Spartan™-3 Generation FPGAs. For embedded systems where compliance is not required, this information helps designers make appropriate design decisions to ensure their systems are operable and reliable. This information is also relevant to other Xilinx FPGA families, as well as other PCI-SIG® technologies, such as the PCI-X™ and PCI Express® technologies.

Setting the V_{CCO} Voltage

For compliant PCI applications in Spartan-3 Generation FPGAs, the V_{CCO} voltage should nominally be +3.3V. XAPP653, *3.3V PCI Design Guidelines*, recommended a reduction in V_{CCO} voltage to reduce the gate oxide stress from electrically demanding transient behavior exhibited by PCI signals and by the overshoot/undershoot compliance test presented in the PCI specification. The reduction in V_{CCO} discussed in XAPP653 is no longer required for Spartan-3 Generation FPGAs based on revisions to the absolute maximum ratings specification for the V_{IN} parameter (the voltage applied to all user I/O pins). Existing designs that use this technique continue to be valid; however, all new designs should take advantage of the enhanced absolute maximum ratings specification.

Note: Spartan-3 Generation FPGAs are suitable and compliant for use in a 3.3V PCI bus environment only. Spartan-3 Generation FPGAs can be used as 3.3V PCI bus devices in a 5.0V PCI bus environment when provided with external protection as described in XAPP646, *Connecting Devices to a 5V PCI Bus*. However, the technique described in XAPP646 is not compliant with the PCI specification.

Powering V_{CCO} Directly from the Bus

The PCI specification defines a connector that provides a number of power rails, including +12V, +5V, +3.3V, and -12V. Any of these rails can be used with an appropriate voltage regulator to generate V_{CCO}, provided the maximum current draw on each bus power rail does not exceed the limit imposed by the PCI specification.

Significant cost savings can be realized if V_{CCO} can be powered directly from the +3.3V bus power rail, eliminating the need for a regulator. The PCI specification defines this power rail to be +3.3V ± 10% (3.0V minimum to 3.6V maximum). To determine if V_{CCO} can be powered directly from the +3.3V bus power rail, two requirements must be satisfied:

- The nominal V_{CCO} voltage must be +3.3V.
- The V_{CCO} tolerance must be ±10% or greater.

While all Spartan-3 Generation FPGAs use a nominal V_{CCO} voltage of +3.3V, only the Spartan-3A, Spartan-3AN, and Spartan-3A DSP FPGA families have a V_{CCO} tolerance of ±10% or greater. Therefore, only the Spartan-3A, Spartan-3AN, and Spartan-3A DSP FPGA families can use the +3.3V bus power rail directly as V_{CCO}. Coupled with their programmable V_{CCAUX} option (where V_{CCAUX} can be set to +3.3V), these families offer a compelling advantage in PCI bus applications, resulting in lower system cost.

For the Spartan-3 and Spartan-3E FPGA families, it is critical to note that V_{CCO} must *not* be powered directly from the +3.3V bus power rail. Instead, it must be regulated due to the tolerance requirement.

Embedded System Tip: If the system +3.3V rail is regulated with sufficient tolerance, it might be possible to directly power V_{CCO}, even for the Spartan-3 and Spartan-3E FPGA families.

Designing V_{CCO} Regulation

When local regulation of V_{CCO} is required, the first question is: “How much current is required to power the I/O circuitry?” A second important question also exists: “How much reverse current can be injected into the rail by charge pump action of the I/O protection diodes?” Answers to these questions are necessary to size the current capability of the regulator for the application.

The hardware-tested and field-proven design presented in XAPP653 contains the answers. The circuit shown in Figure 1 powers a complete 64-bit bus interface (66 MHz or 33 MHz) using a 500 mA linear voltage regulator. One resistor value in the feedback network has been modified from the original design, setting the rail voltage at +3.3V. The input and output capacitor values have been changed to match the “typical application” circuit provided in the regulator data sheet.

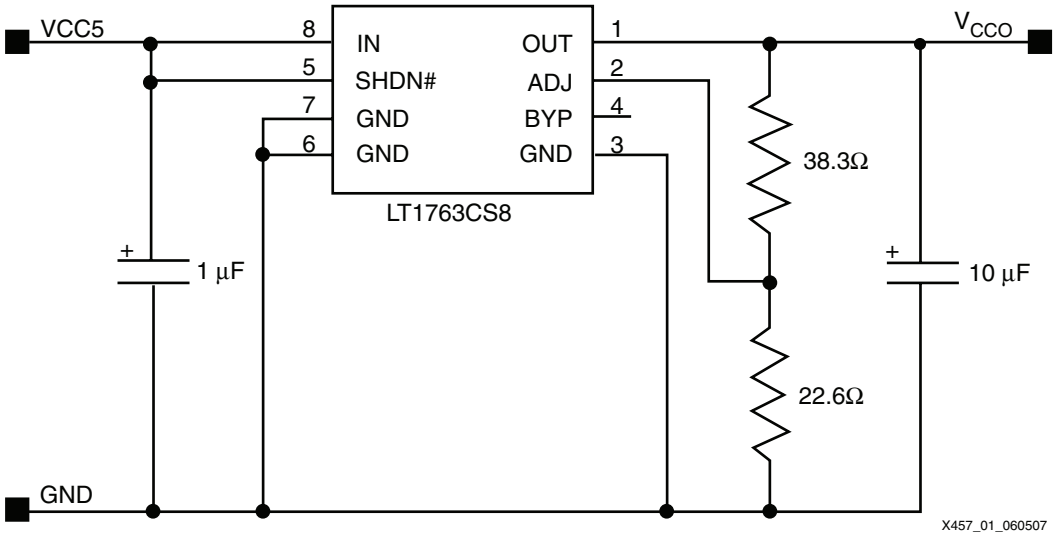

Figure 1: **XAPP653 Regulator, Modified for +3.3V**

Careful inspection of the original design reveals that the impedance of the feedback network is atypically low. Even with zero load, approximately 50 mA is lost through the feedback network. At first, this loss seems undesirable, but it serves an important purpose. The I/O protection diodes in Spartan-3 Generation FPGAs form a charge pump that can inject excess bus switching energy into the V_{CCO} and GND rails. This effect is pronounced when the device is not participating in a bus transaction, but switching activity from other devices generates overshoot/undershoot on the bus signals.

This effect poses a problem for a regulated V_{CCO} rail. Most regulators do not tolerate reverse current and might lose voltage regulation under such circumstances. One way to solve this problem, at the expense of slightly higher power consumption, is to shunt current to the GND rail as shown in Figure 2.

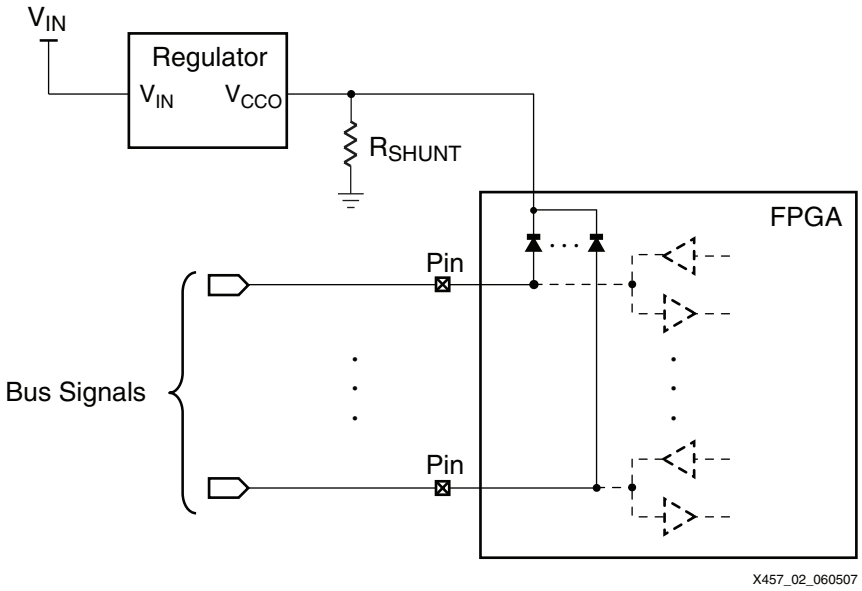

Figure 2: **Handling Reverse Current with a Shunt Resistor**

This solution can be implemented in a typical regulator application using a low-impedance resistor, R_{SHUNT}, connected between the regulator output and GND. In the case of a linear regulator with a feedback network, the feedback network impedance can be lowered while retaining the correct ratio for the desired output voltage, eliminating the need for an additional resistor.

Based on the design and characterization of XAPP653, regulation for a 64-bit bus interface consisting of 90 pins is adequately protected by a 50 mA quiescent current, which corresponds to approximately 0.6 mA per pin. Subtracting the 50 mA quiescent current from the 500 mA regulator capability yields 450 mA remaining to power the I/O circuitry. Although a 64-bit bus interface consists of 90 pins, the protocol limits the number of driven pins to 78 during any bus phase, which corresponds to approximately 5.8 mA per switching pin.

This information can be used to calculate conservative supply requirements for a 32-bit bus interface. A 32-bit bus interface consists of 50 pins, with the protocol limiting the number of driven pins to 40 during any bus phase. The approximate current requirement is 250 mA for a 32-bit bus interface.

Note: The Spartan-3A FPGA family uses a floating well structure in the I/O, which is typically associated with the absence of I/O protection diodes. However, Spartan-3A FPGA devices contain I/O protection diodes that are enabled specifically in PCI SelectIO™ modes. While few designs in Spartan-3A FPGA devices use a locally regulated V_{CCO}, those that do must be designed to handle the reverse current requirements discussed in this section.

Distributing I/O Placement to Observe Guidelines for SSOs

In most bus interface applications implemented in Spartan-3 Generation FPGAs, there is a subtle conflict in I/O placement. On one hand, it is desirable to have all I/O pins placed close to each other because internal timing constraints are easier to meet. On the other hand, it is desirable to have all I/O pins far apart from each other to minimize signal integrity challenges associated with simultaneously switching outputs (SSOs).

In the device data sheets for the Spartan-3 Generation FPGAs, Xilinx provides guidelines that describe the maximum number of I/O pins of a given output signal standard that should simultaneously switch in the same direction, while maintaining a safe level of switching noise. Meeting these guidelines ensures that the device operates free from the adverse effects of ground and power bounce.

Ground or power bounce occurs when a large number of outputs simultaneously switch in the same direction. The output drive transistors all conduct current to a common rail. Low-to-High transitions conduct to the V_{CCO} rail; High-to-Low transitions conduct to the GND rail. The resulting cumulative current transient induces a voltage difference across the inductance that exists between the output driver on the die and the V_{CCO} or GND rail. The inductance is associated with bonding wires, the package lead frame, and any other signal routing inside the package. Any SSO-induced difference in V_{CCO} or GND voltage on the die affects internal switching noise margins and ultimately signal quality.

Two questions arise:

1. How many SSOs are allowed?
2. How many SSOs exist in the bus interface?

How Many SSOs are Allowed?

The answer to the first question, which can be found in the device data sheet, is a function of the device size and the package selection. Use these parameters to look up the number of equivalent V_{CCO}/GND pairs per I/O bank on the device. Next, use the choice of SelectIO mode to look up the number of SSOs allowed per equivalent V_{CCO}/GND pair. The product of these values yields the number of SSOs allowed per bank for a specific device size and package selection. In general, BGA packages are superior to QFP packages when SSOs are a concern.

How Many SSOs Exist in the Bus Interface?

The answer to the second question is found by evaluating the bus protocol. For PCI bus implementations, it is impossible for all outputs to switch simultaneously. A generic analysis of output switching by an automated tool will over-report unless the tool is aware of the bus protocol.

Consider the case of a 64-bit bus interface. In this example, the full signal list consists of the following 90 signals:

- AD[63:0]
- CBE[7:0]
- PAR, PAR64, PERR#
- IRDY#, FRAME#, REQ64#
- TRDY#, STOP#, DEVSEL#, ACK64#
- INT#, PME#, SERR#
- REQ#, GNT#, IDSEL
- CLK, RST

Based on their specified use, some signals can be ignored. The CLK, RST, GNT#, and IDSEL signals are input-only pins and can be removed from further consideration. INT#, PME#, and SERR# do not switch often, only drive Low, and are exempt from most bus timing requirements.

These three signals can use a low-drive, slow-slew SelectIO mode and therefore can be removed from further consideration.

The list is reduced to the following 83 relevant output capable signals:

- AD[63:0]
- CBE[7:0]
- PAR, PAR64, PERR#
- IRDY#, FRAME#, REQ64#
- TRDY#, STOP#, DEVSEL#, ACK64#
- REQ#

Now consider what the bus transaction protocol does to this list. When the device is active, one of four transactions can be taking place:

1. Target Reads
2. Target Writes
3. Initiator Reads
4. Initiator Writes

Case 2 and Case 3 are not interesting because they represent bus activity where the output drivers for AD[63:0] are three-stated, resulting in a drastic reduction in SSOs. Case 1 and Case 4 are evaluated separately.

For Case 1, Target Reads, there are 71 relevant output capable signals:

- AD[63:0]
- PAR, PAR64
- TRDY#, STOP#, DEVSEL#, ACK64#
- REQ#

For Case 4, Initiator Writes, there are 78 relevant output capable signals:

- AD[63:0]
- CBE[7:0]
- PAR, PAR64
- IRDY#, FRAME#, REQ64#
- REQ#

For a 32-bit bus interface, removing the 64-bit extension signals AD[63:32], CBE[7:4], PAR64, ACK64#, and REQ64# from the signal list yields a maximum of 40 output pins simultaneously switching during an initiator write bus cycle. During a target read, there are no more than 37 output pins simultaneously switching. Remember that the signal sets for target reads and initiator writes are not identical.

I/O Placement

With answers to the SSO questions from the previous two subsections, it is possible to evaluate and modify proposed I/O placements for compliance with the SSO guidelines. This step is simple for designs that fit in a single bank. For I/O placements that span multiple banks, it is necessary to separately evaluate the signal sets for target reads and initiator writes in all used banks to make sure both transactions comply with the SSO guidelines.

For designs with proposed I/O placements that violate the SSO guidelines, two options exist. Space permitting, I/O placements can be artificially spread out into additional banks as far as internal timing constraints allow. Another option is to select a different part and package combination.

Note: In all designs, proper power and ground distribution systems are critical. Refer to XAPP623, *Power Distribution System Design: Using Bypass/Decoupling Capacitors* for additional information.

Strapping the Pre-Configuration I/O Pull-Up Resistor Option

Prior to the completion of configuration, the user I/O pins on Spartan-3 Generation FPGAs are in one of two states:

- three-stated

  or
- pulled up to V_{CCO} by internal pull-up resistors

A special pin called PUDC_B, HSWAP_EN, or HSWAP (depending on the specific Spartan-3 Generation FPGA family) controls the behavior. The selected behavior applies globally to all user I/O pins.

It is a violation of the PCI specification to connect anything other than a compliant I/O buffer to a bus signal; such a buffer typically does not include pull-up, pull-down, or weak keeper functionality.

For this reason, the pre-configuration I/O pull-up resistor option must be disabled by driving PUDC_B, HSWAP_EN, or HSWAP to a logic High value by properly strapping it on the board. Any other user I/O pins, which are not part of the bus interface and require a valid logic level before configuration completes, require external pull-up or pull-down resistors.

Furthermore, post-configuration PULLUP, PULLDOWN, or KEEPER attributes on the I/O pins of the bus interface must not be used.

Minimizing Ramp Time of Regulated Power Supplies

To ensure a successful power-on event, the Spartan-3 Generation FPGAs require their supply ramp times to fall within a specified range. These supply ramp time requirements apply to V_{CCINT}, V_{CCAUX}, and V_{CCO} and are specified for each family in the respective data sheet.

An intersection of the requirements for all members of the Spartan-3 Generation, for all three supplies, yields the following universal recommendations for production qualified devices:

- Every supply must ramp monotonically from GND.
- The supply ramp time must be greater than 0.2 ms, but less than 50 ms.

A successful power-on event for the Spartan-3 Generation FPGA does not guarantee a successful power-on event for the system. All PCI-SIG technologies specify time limits for events in the system power-on sequence. In some implementations, these specifications can be difficult to satisfy. Failing to meet the specifications might result in system crashes or other fatal application errors.

In general, it is necessary to ramp the supplies as quickly as possible. For this reason, regulation schemes that include fold back or soft-start should not be used. To form a meaningful time budget for system power-on events, a realistic estimate of the ramp time is required. This estimate is determined by the power supply design and is therefore not specified by Xilinx.

Budgeting Time Allowed for FPGA Power-On

Figure 3 shows the power-on timing requirements for the PCI bus. The deassertion of RST# is important because the host bridge broadcasts a bus initialization message at this time.

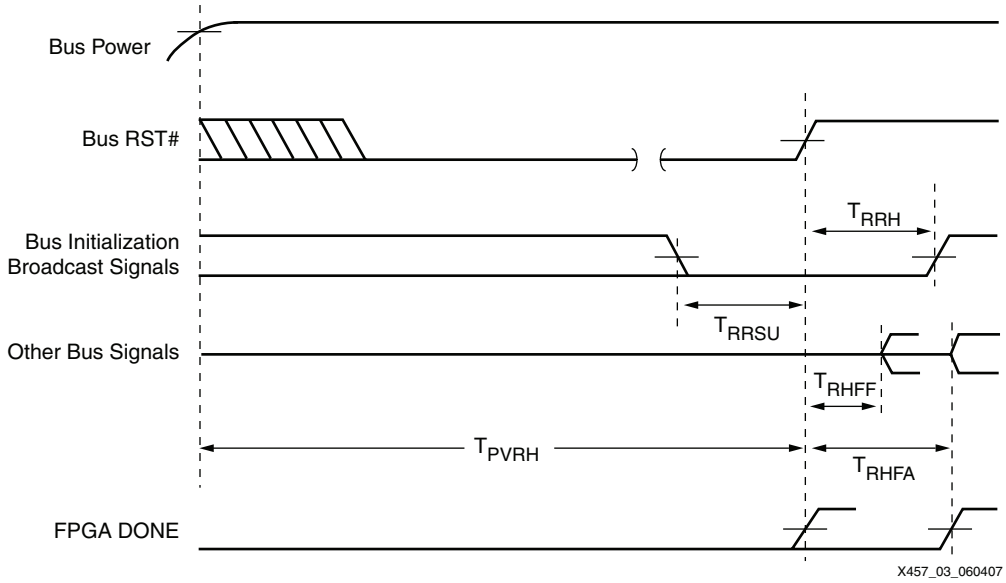

Figure 3: **Bus Interface Power-On Requirements**

Table 1 lists the important parameters from Figure 3 with their associated values. This message, which is part of the system interoperability protocol, must be received by all devices on the bus that are 64-bit capable, PCI-X capable, or both. For a basic 32-bit PCI mode device, it is not necessary for the device to receive the bus initialization message.

Table 1: **Power-On Timing Requirements**

Symbol	Parameter	Min	Units
T_{PVRH}	Power Valid to RST# High	100	ms
T_{RHFA}	RST# High to First Configuration Access	2^{25}	clocks

For a basic 32-bit PCI mode device, the PCI specification requires the device to be ready for bus activity within (T_{PVRH} + T_{RHFA}). The Spartan-3 Generation FPGA must be in user mode with DONE asserted before (T_{PVRH} + T_{RHFA}) has elapsed. Because T_{RHFA} is specified in clock cycles, it must be converted to time based on the bus clock frequency:

- 32-bit, 33 MHz PCI designs: T_{PVRH} + T_{RHFA} = 100 ms + 2^{25} · 30 ns = 1,107 ms
- 32-bit, 66 MHz PCI designs: T_{PVRH} + T_{RHFA} = 100 ms + 2^{25} · 15 ns = 603 ms

For any device capable of 64-bit or PCI-X operation, the PCI specification requires the device to be ready for the bus initialization message within T_{PVRH}. The FPGA must be in user mode with DONE asserted before T_{PVRH} has elapsed. This is a substantially smaller window of time.

- 64-bit or PCI-X capable designs: T_{PVRH} = 100 ms

Through these requirements, the specification sets an upper bound on the amount of time available for the power-on and configuration events of the Spartan-3 Generation FPGAs. For convenience, the maximum amount of time allowed between power valid and DONE assertion is named T_{PVDH}. T_{PVDH} can be 1,107 ms, 603 ms, or 100 ms depending on the type of interface (32-bit/33 MHz, 32-bit/66 MHz, or 64-bit). The following events lead to DONE assertion and must complete within T_{PVDH}.

- Power supply ramp time. This event is named T_{PSRAMP} for convenience. Refer to "Minimizing Ramp Time of Regulated Power Supplies," page 6 for a discussion of this event. T_{PSRAMP} can range from 0.2 ms to 50 ms.

- Internal power-on reset. The duration of this event, T_{POR}, is a function of device family and density. T_{POR} can range from 0 to 18 ms. For a specific device and density, consult the relevant device data sheet.
- Internal oscillator start-up. The duration of this event, T_{ICCK}, is a function of the configuration mode used. T_{ICCK} can range from 0 to 4 µs. For a specific device and configuration mode, consult the relevant device data sheet.
- Configuration bitstream loading. The duration of this event, named $T_{BITLOAD}$ for convenience, is a function of device family, density, configuration clock frequency, and configuration data port width. Consult the relevant device data sheet for more information. In general:

  $T_{BITLOAD}$ = (Bitstream Length in bits) / ((Clock Frequency in Hz) * (Configuration Port Width in bits))

  When using a configuration clock derived from a crystal oscillator, the nominal clock frequency can be used in the calculation. When the configuration clock is derived from a ring oscillator or other highly variable source (for example, the Spartan-3 Generation FPGA is using a “master” configuration mode), use the guaranteed minimum clock frequency in the calculation.
- Waiting for system-level silicon features to lock. The duration of this event, named T_{LOCK} for convenience, represents the optional capability of the device to delay the start of user mode until system-level silicon features such as the Digital Clock Manager (DCM) have achieved lock. By default, this feature is not enabled. Let the system-level silicon feature lock after configuration has completed, so that the device can enter user mode as soon as possible to receive the initialization message.

To guarantee a successful system power-on event:

$$T_{PSRAMP} + T_{POR} + T_{ICCK} + T_{BITLOAD} + T_{LOCK} < T_{PVDH}$$

The Configuration Time Estimator for PCI Applications spreadsheet (located in xapp457.zip) assists with evaluating the time required for power-on events leading to DONE assertion.

Embedded System Tip: When the bus reset is controlled by firmware, the deassertion of reset can be delayed to increase the available time for the Spartan-3 Generation FPGA to enter user mode. As a result, a lower-performance, lower-cost configuration scheme can be used.

Conclusion

System designers can use Spartan-3 Generation FPGA devices to build low-cost implementations of PCI-SIG technologies. Several points merit careful attention to achieve long term reliability and broad interoperability. An understanding of these items early in the design process minimizes the likelihood of a costly redesign to address latent design issues that might be discovered only when the product is in volume production.

References

The following documents provide supplemental information useful to this application note:

- DS099, *Spartan-3 FPGA Family Data Sheet*
  http://www.xilinx.com/bvdocs/publications/ds099.pdf
- DS312, *Spartan-3E FPGA Family Data Sheet*
  http://www.xilinx.com/bvdocs/publications/ds312.pdf
- DS529, *Spartan-3A FPGA Family Data Sheet*
  http://www.xilinx.com/bvdocs/publications/ds529.pdf
- DS557, *Spartan-3AN FPGA Family Data Sheet*
  http://www.xilinx.com/bvdocs/publications/ds557.pdf
- DS610, *Spartan-3A DSP FPGA Family Data Sheet*
  http://www.xilinx.com/bvdocs/publications/ds610.pdf

- XAPP623, *Power Distribution System Design: Using Bypass/Decoupling Capacitors*
  http://www.xilinx.com/bvdocs/appnotes/xapp623.pdf
- XAPP646, *Connecting Devices in the Virtex™ and Spartan Families to a 3.3V or 5V PCI Bus*
  http://www.xilinx.com/bvdocs/appnotes/xapp646.pdf
- XAPP653, *3.3V PCI Design Guidelines*
  http://www.xilinx.com/bvdocs/appnotes/xapp653.pdf
- PCI Local Bus Specification, Revision 3.0
  http://www.pcisig.com/specifications/conventional
- LT1763 Series 500 mA, Low Noise, LDO Micropower Regulators
  http://www.linear.com/pc/downloadDocument.do?navId=H0,C1,C1010,C1764,P1778,D3903

The XAPP457 Configuration Time Estimator for PCI Applications spreadsheet (located in xapp457.zip) is derived from an original work by the Xilinx Configuration Solutions Application Engineering Team.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
06/08/07	1.0	Initial Xilinx release.