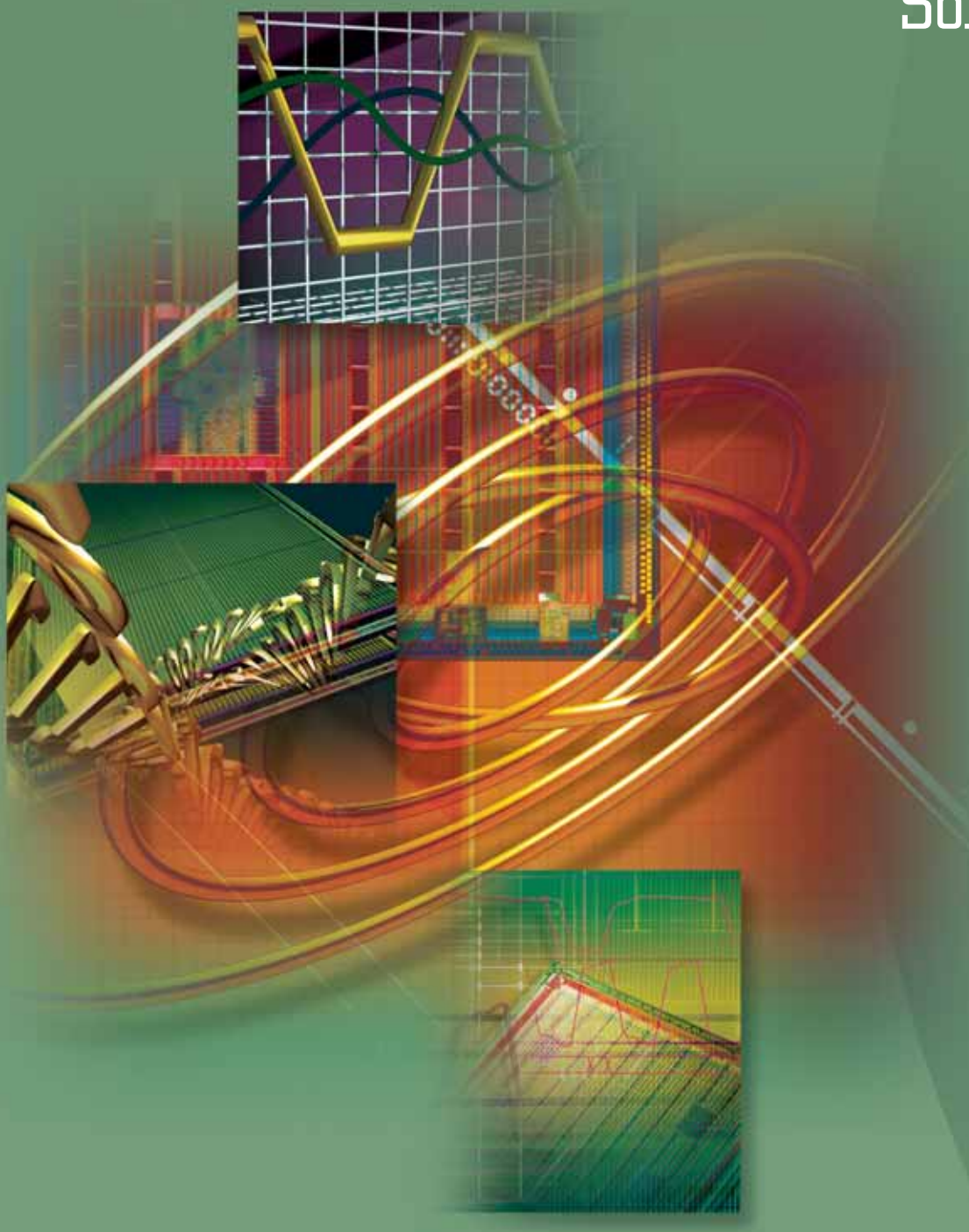
Issue 1
July 2005

Signal Integrity

Solution Guide



Being Heard Above the Noise

INSIDE

ARTICLES

Managing Signal Integrity

Extend Your Reach

Losing Less from Lossy Lines

WEBCAST

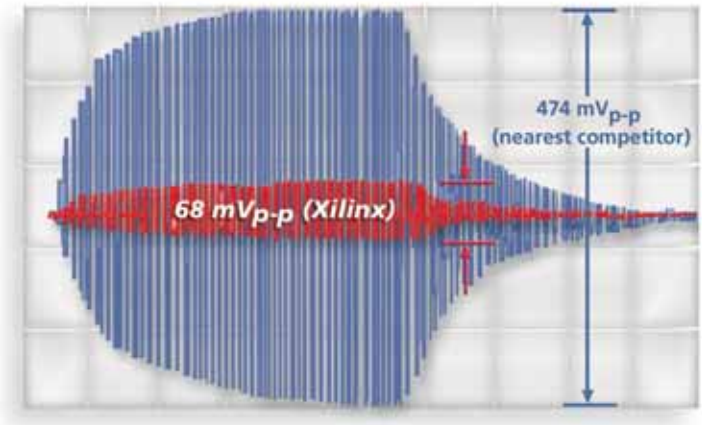
Dr. Howard Johnson: BGA Crosstalk

APPLICATION NOTES

Power Distribution System (PDS) Design: Using Bypass/Decoupling Capacitors

XILINX®

"High Signal Integrity Demands a Low-Noise Chip".



Design Example: 1.5 volt LVCMOS 4mA, I/O, 100 aggressors shown.

Best Signal Integrity: 7x Less SSO Noise

Virtex-4 FPGAs deliver the industry's best signal integrity, allowing you to pre-empt board issues at the chip level, for high-speed designs such as memory interfaces. Featuring a unique SparseChevron™ pin out pattern, the Virtex-4 family provides the highest ratio of VCCO/GND pins to user I/O pins available in any FPGA. By strategically positioning one hard power/ground pin adjacent to every user I/O on the device, we've reduced signal path inductance and SSO noise to levels far below what you can attain with a virtual ground or soft ground architecture.

The Industry's Highest Signal Integrity, Proven By Industry Experts

Incorporating continuous power and ground planes, plus integrated bypass capacitors (on the chip and inside the package), we're eliminating power-supply noise at its source. In addition, we provide on-chip termination resistors to control signal ringing. Guaranteed lowest jitter and custom slew-rate control give you industry-leading results for any design. The lab tests speak for themselves. As measured by signal integrity expert Dr. Howard Johnson, no competing FPGA comes close to achieving the low-noise benchmarks of Virtex-4 devices.

Visit ***www.xilinx.com/virtex4/sipi*** today, and choose the right high-performance FPGA before things get noisy.

Dr. Howard Johnson, author of *High-Speed Digital Design,* frequently conducts technical workshops for digital engineers at Oxford University and other sites worldwide. Visit www.sigcon.com to register.

XILINX®
The Programmable Logic Company℠

www.xilinx.com/virtex4/sipi

Breakthrough Performance At The Lowest Cost

© 2005 Xilinx, Inc. All rights reserved. XILINX, the Xilinx logo, and other designated brands included herein are trademarks of Xilinx, Inc. All other trademarks are the property of their respective owners.

Signal Integrity
Solution Guide

EDITOR IN CHIEF Carlis Collins
editor@xilinx.com
408-879-4519

MANAGING EDITOR Forrest Couch
forrest.couch@xilinx.com
408-879-5270

ASSISTANT MANAGING EDITOR Charmaine Cooper Hussain

XCELL ONLINE EDITOR Tom Pyles
tom.pyles@xilinx.com
720-652-3883

ADVERTISING SALES Dan Teie
1-800-493-5551

ART DIRECTOR Scott Blair

Xcell Publications

Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124-3400
Phone: 408-559-7778
FAX: 408-879-4780

© 2005 Xilinx, Inc. All rights reserved. XILINX, the Xilinx Logo, and otherdesignated brands included herein are trademarks of Xilinx, Inc. PowerPC is a trademark of IBM, Inc. All other trademarks are the property of their respective owners.

The articles, information, and other materials included in this issue are provided solely for the convenience of our readers. Xilinx makes no warranties, express, implied, statutory, or otherwise, and accepts no liability with respect to any such articles, information, or other materials or their use, and any use thereof is solely at the risk of the user. Any person or entity using such information in any way releases and waives any claim it might have against Xilinx for any loss, damage, or expense caused thereby.

Too Important to Ignore

Welcome to the first edition of the Xilinx® Signal Integrity Solution Guide. Feedback from our customers indicates that today, designers are caught between requirements for ever-increasing bit rates, faster edge rates, higher clock speeds, and technology advances that keep lowering operating voltages, reducing package sizes, and ball pitch, forcing more components into a smaller amount of board area.

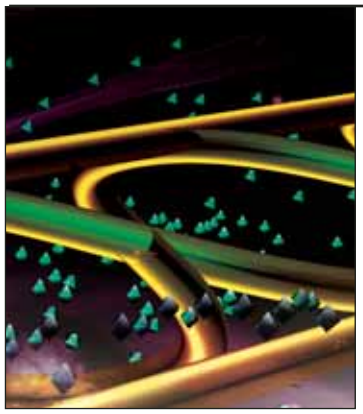
Take a look at present-day source-synchronous interfaces. DDR and QDR memory interface speeds are rapidly increasing, with DDR2 speeds at more than 500 Mbps. The bit rates are getting faster and the buses getting wider. With faster bit rates come faster edge rates, which now can be just a few hundred picoseconds.

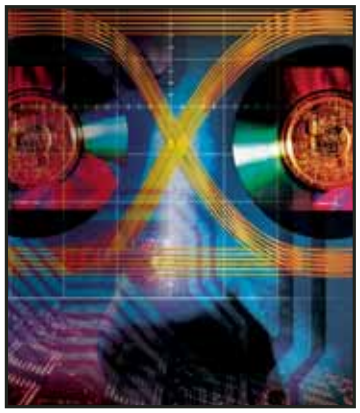
Faster is better, but there are a few hurdles to deal with. You've told us that you cannot allow yourself to ignore signal integrity and gamble that your system will work as designed. You might have to reduce the clock rate just to get the system to work, or be forced into a complete board re-design to correct signal integrity issues.

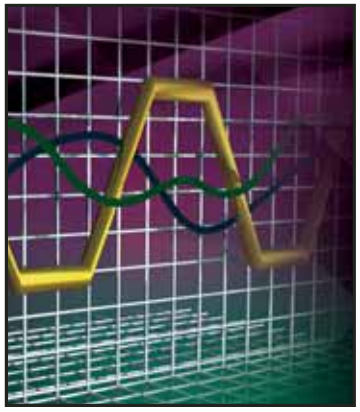
This Solution Guide contains some of the best material published recently by Xilinx on the topic of signal integrity. Included are recent *Xcell Journal* articles and application notes from Xilinx, Mentor Graphics, and Dr. Howard Johnson, a noted industry expert on high-speed digital design and signal integrity. Of particular interest is a complete illustrated transcript of Dr. Johnson's March 2005 online web seminar on BGA crosstalk and signal integrity in FPGAs.

Space limitations do not allow us to print all of the great material available on this topic. To access all of the signal integrity information available from Xilinx, visit the Signal Integrity Central website at *www.xilinx.com/signalintegrity*.

Andy DeBaets
Sr. Director, Systems &
Application Engineering,
Advanced Products Division







CONTENTS

"In any high-frequency inductive problem, the relevant magnetic field resides in the space between conductors, not in the conductors themselves. This field, not the conductors, causes all inductive effects."

ARTICLES

Managing Signal Integrity 5

Designing for Signal Integrity 8

Extend Your Reach 11

Ten Reasons Why Performing SI Simulations is a Good Idea 15

For Synchronous Signals, Timing Is Everything 16

Eyes Wide Open 20

Solving the Signal Integrity Challenge.....23

Xilinx/Micron Partner to Provide High-Speed Memory Interfaces 25

Losing Less From Lossy Lines.....27

WEBCAST

BGA Crosstalk 30

APPLICATION NOTES

Power Distribution System (PDS) Design
Using ByPass/Decoupling Capacitors 41

Learn more about power management solutions from Xilinx at: *www.xilinx.com/xcell/signal1/*

Managing Signal Integrity

Being heard above the noise.

by Steve Sharp
Sr. Marketing Manager
Xilinx, Inc.
steve.sharp@xilinx.com

Panch Chandrasekaran
Connectivity Marketing Manager
Xilinx, Inc.
panch.chandrasekaran@xilinx.com

The problem of signal integrity is a lot like trying to carry on a conversation at a crowded trade show. If you and the person you're talking to are in a quiet corner of the hall with some nice padded walls around you and not too many other people nearby, it isn't a problem. Try the same conversation in the middle of the exhibit floor with hundreds of people all around, noise from neighboring exhibit booths, and no walls to break up or absorb the sound, and you've got a problem.

Back in the good old days of logic design, we didn't give much thought to signal integrity. We had 5V power supplies, DIP packages with leads that actually went through the board, and high-speed microprocessors running at a heady 5 MHz.

If you paid a little attention to board layout and put a decent ceramic bypass capacitor next to each chip, you probably didn't have to worry about your signals. Ones stayed ones and zeroes stayed zeroes. Even 100 mV of noise on a signal wouldn't be enough to change its logic level.

Today, designers are caught between design requirements for ever-increasing bit rates, faster edge rates, higher clock speeds, and technology advances that keep lowering operating voltages, reducing package sizes and ball pitch, and forcing more components into a smaller amount of board area.

Signal Integrity Today

Take a look at present-day source-synchronous interfaces. DDR and QDR memory interface speeds are rapidly increasing, with DDR2 speeds at more than 500 Mbps. The bit rates are getting faster and the buses getting wider. With faster bit rates comes faster edge rates, which now can be just a few hundred picoseconds.

Faster is better, but there are a few hurdles to deal with. Parasitic inductance and capacitance – which didn't matter a whole lot at lower speeds – are suddenly very important. The resulting noise because of the parasitics is a big concern. Today it is common to have FPGAs with hundreds of I/Os switching, causing high levels of simultaneous switching output noise (SSN). This affects your system in many ways, especially causing jitter, which can reduce your timing margin or even cause system failure.

You cannot allow yourself to ignore signal integrity and gamble that your system will work as designed. You might be forced to reduce the clock rate just to get the system to work, or be forced into a complete board re-design to correct signal integrity issues.

What Kind of Integrity Do You Have?

Having good signal integrity usually means controlling unwanted noise on logic signals. Noise usually falls into one of two main domains:

- Level-related noise affects the logic level of the signal. If the noise is large enough, the signal may cross the threshold from a desired logic state to an undesired state and propagate into other logic.
- Time-related noise, or jitter, affects the position of a signal transition and causes setup/hold windows for data sampling to be violated, thereby allowing incorrect data to be sampled and propagated through the system.

The combination of level noise and jitter combine to reduce signal margins in both the voltage and time domains, effectively reducing the "eye" or window in which good data is available.

Controlling Noise

A well-designed package is critical to signal integrity. Noise can emanate from many sources in a system. If the noise source is on the board, there are some potential solutions once you find out where the problem is (probably after a long and laborious debug process). If the problem is in the package, you have little or no choice but to change the design, vendor, or parts. This is a time-consuming process that can affect product revenue significantly. For this reason, it is imperative to have a well-designed low-inductance package.

When speeds were still fairly low, short signal paths did not alter signal characteristics. Today, with rise times in the hundreds of picoseconds (even if bit periods are a few nanoseconds), the frequency components of signals run into gigahertz, causing even very short signal paths like package traces to impact signals.

For every signal line, there is an associated return path for the return currents. For single-ended signals, these return paths are usually GND or VCC reference planes. To maintain a 50 ohm line, the returns should be in close proximity to the signal.

Although PCB traces are less of a concern, you must pay close attention to vias. For large FPGAs the breakout region – the area between the package balls to the PCB – is extremely critical, as it comprises a dense concentration of signal vias.

SSN is generally observed as "ground bounce" and can be caused by two different phenomena.

First, noise because of via-field crosstalk is a function of loop inductance, which is a function of the proximity of ground/power reference pin locations to the signal pin. Signal pins farther away from a reference pin are more susceptible to noise.

This problem is exacerbated when a number of I/Os in the region switch simultaneously. Proper distribution of ground/power and signal pins in a package is extremely critical – in other words, a good pinout architecture.

Second, maintaining a clean power supply to the FPGA is also critical to maintain acceptable signal integrity. Noise margins are reduced as VCC values drop down to 1.2V.

Furthermore, any noise in the power rail translates to jitter at the output, shrinking available timing margins. As noise depends on package inductance and the number of simultaneously switching I/Os, optimal signaling requires a good low-inductance package.

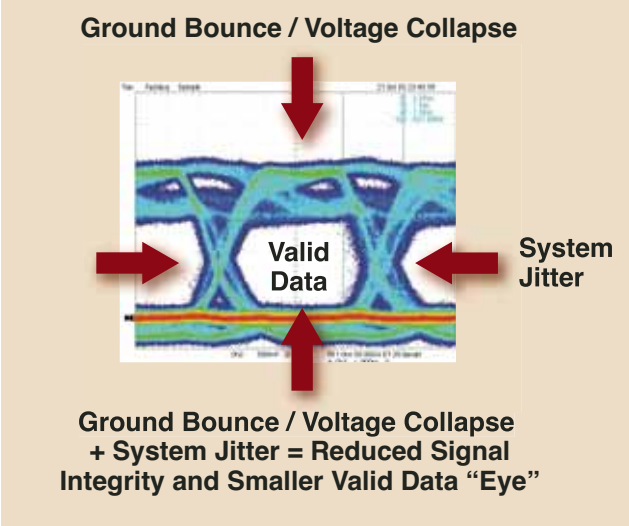


Figure 1 – How noise and jitter can affect an eye diagram.

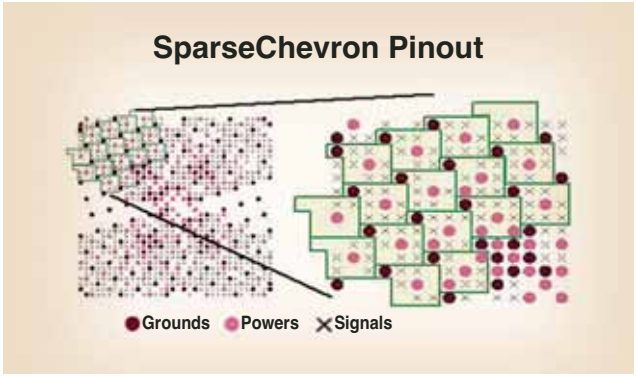


Figure 2 – Xilinx SparseChevron pinout places return pins adjacent to each signal pin.

Tackling the SSN Challenge

One package that tackles the SSN challenge is the Xilinx® Virtex™-4 FPGA package. Most notably, the package enables better noise performance on higher speed single-ended interfaces, which are more susceptible to noise than differential interfaces such as LVDS.

The pinout architecture of the package is responsible for roughly 80% of the total noise. The Virtex-4 FPGA package achieves optimal pin distribution through a tiled pattern – a regular array of signal, ground, and power pins called SparseChevron pinout (Figure 2).

The signal-to-ground-to-power ratio of the package is 8:1:1. Because both power and ground are equally effective as return current paths, the package effectively has a signal-to-return ratio of 4:1. Also, the pins are distributed so that every signal pin is adjacent to a return pin, ensuring that the return current loop is kept to a minimum.

Additionally, the abundance of return paths in any given area of the package provides a low impedance path for the return currents. The pinout also confines noise from an aggressor to a smaller area so that the influence of the aggressor drops rapidly with distance. Because crosstalk noise is cumulative, this results in a lower total SSN.

Simplifying Signal Termination

On-chip termination (active termination) removes external components and places termination closest to where it matters (driver or receiver).

To maintain the ideal 50 ohm line impedance, it is normal design practice to have termination resistors on each signal. For hundreds of signal I/Os, this can translate to many hundreds of external termination resistors. The physical challenges of placing the resistors on the board and their connections to the power and ground planes are not trivial.

The Xilinx Controlled Impedance Technology (XCITE) on-chip active I/O termination used in Virtex FPGAs solves many of the problems associated with signal termination. XCITE provides both parallel and serial equivalent options for single and differential termination. Impedance is controlled using an internal reference voltage and is available on all I/O pins. This active termination provides automatic temperature and volt-

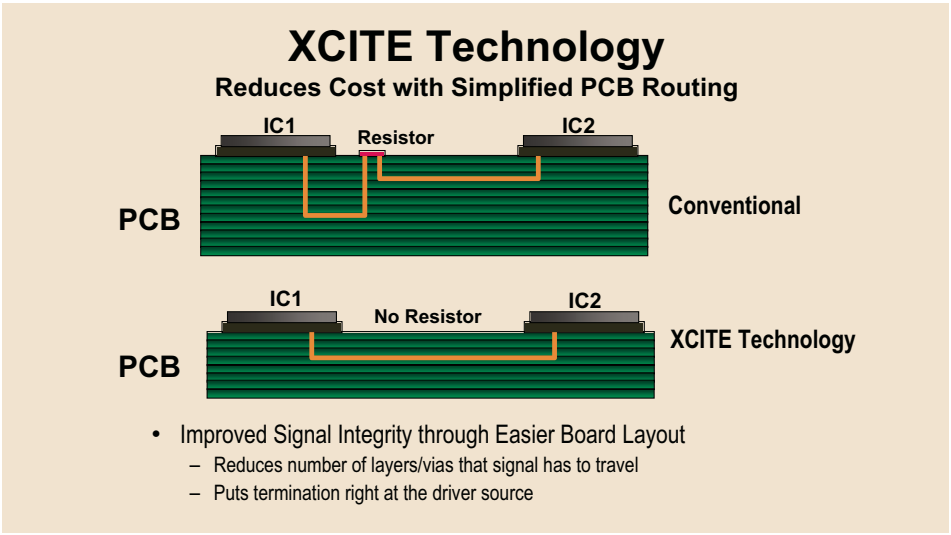


Figure 3 – Xilinx XCITE technology reduces cost through simplified PCB routing.

age compensation; puts the termination inside the buffer circuitry where it belongs; and saves board space and cost by eliminating hundreds of discrete resistors. Figure 3 shows the simplified board layout and signal trace paths using both conventional and Xilinx XCITE DCI termination technology.

Power Plane Integrity

Power and ground planes are important to maintaining signal integrity in FPGA designs. To maintain the characteristic impedance (Zo) across the frequency range of interest, reference planes for single-ended signals should be very low impedance.

Otherwise, the result is impedance discontinuities, causing jitter due to reflections. In addition, noisy power and ground planes affect circuit performance on the die, causing additional jitter. It is important to design packages with continuous power and ground planes to minimize impedance.

Typically, PCB designers use decoupling capacitors to filter out noise and maintain a clean power supply. For reducing high-frequency noise, decoupling capacitors are placed close to the noise source. Leading-edge ASICs and FPGAs are equipped with very low-inductance decoupling capacitors within the package to aid cleaning the power-supply noise.

Compensating for Signal Integrity Issues

Improving the signal integrity of your system will enhance the data valid window (the eye) of the high-frequency signals reaching your FPGA I/O pins. However, this is only half the battle. Even superior designs exhibit shrinking data valid windows, as shown in the 533 Mbps DDR2 SDRAM example shown in Figure 4. The input circuitry needs the capability to capture the data by centering the clock to the middle of the shrinking data valid window.

Virtex-4 FPGAs have unique ChipSync™ technology built into every I/O block that makes data capturing easier and more reliable. It includes a precision delay called IDELAY that generates the tap delays necessary to center data to the FPGA clock. Memory strobe edge detection logic, included in the I/O block, uses this precision delay to detect the edges of the memory strobe from which the pulse center can be calculated. Delaying the data by the number of delay taps counted between the first and second edges aligns the center of the data window with the edge of the FPGA clock output. The tap delays generated by this precision delay block allow alignment of the data and clock to within 75 ps resolution.

ChipSync technology also simplifies the design of differential parallel bus interfaces, with embedded SERDES blocks that serialize and de-serialize parallel interfaces to match the data rate to the speed of the internal FPGA circuits. Additionally, this technology provides per-bit and per-channel de-skew for increased design margins, simplifying the design of interfaces such as SPI-4.2, XSBI, and SFI-4, as well as RapidIO.

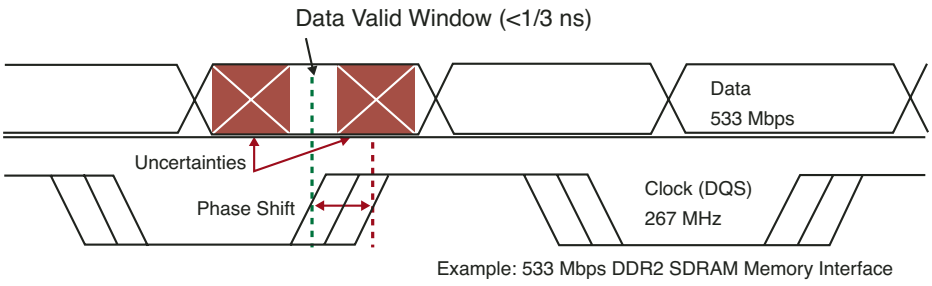


Figure 4 – Data valid window for a 533 Mbps DDR2 SDRAM interface

Conclusion

Signal integrity is a key issue in today's high-speed designs and will continue to be important as more high-speed signals are squeezed into smaller amounts of board space, packages get denser, and ball spacing shrinks.

Signal integrity issues can affect voltage and time domains, combining to reduce the window of available valid data in a system. If the issues become large enough, systems may not work at all or be extremely unreliable, forcing long and costly system redesigns.

It might never be possible to completely eliminate signal noise in a high-speed system. But paying attention to several key areas can minimize noise or adjust timing so that system performance is not compromised. These include using well-engineered, low-inductance packages; using devices with built-in power supply decoupling; using active signal termination where necessary; and choosing devices with the ability to adjust the relationship between the data valid window and the clock.

For more information, visit *www.xilinx.com/signalintegrity.*

Designing For Signal Integrity

You can use the Xilinx/Ansoft 10 Gbps Backplane Design Kit to predict interconnect performance.

by Suresh Sivasubramaniam
Senior Design Engineer
Xilinx, Inc.
suresh.subramaniam@xilinx.com

Lisa Murphy
Application Engineer
Ansoft
lmurphy@ansoft.com

The Xilinx® Virtex-4™ FX family of devices contains up to 24 RocketIO™ multi-gigabit transceivers, each capable of operating anywhere from 622 Mbps to 10.3125 Gbps. This seamless scalability, coupled with support for various emerging standards (Figure 1), allows you tremendous flexibility to upgrade today's designs to meet increasing bandwidth requirements.

To realize the full potential of this upgradeability to high-bandwidth processing applications, you must carefully design the serial interconnect channels on the PCB, be it line card or backplanes. Once the transfer characteristics of the physical channel are well understood, you can effectively employ features such as transmit pre-emphasis/voltage swing and receive equalization (Figure 2) to overcome losses and attenuation in the channel, thus ensuring high signal integrity at the receiver.

MK322 Evaluation Board Case Study

The MK322 platform is the primary board used for the electrical evaluation and characterization of the RocketIO X high-speed serial multi-gigabit transceivers in Virtex-II Pro™ X FPGAs. This board was specifically designed to evaluate and test the RocketIO X transceiver and is available for sale.

The SMA connectors on the board allow you to interface the board to a scope, to other boards, or for loopback tests. The physical channel for each transceiver is carefully optimized to ensure the highest signal quality at the SMAs (on the transmit path) or at the FPGA (on the receive path).

The data can significantly degrade after it has passed through the transmission path. Degradation includes loss of signal amplitude, reduction of signal rise time, and a spreading at the zero crossings. It is critical to model the transmission path when designing a high-performance, high-speed serial interconnect system. The transmission path may include long transmission lines, connectors, vias, and crosstalk from adjacent interconnect.

MK322 Board Stackup

The MK322 is a 12-layer board. The stack and trace geometries are designed for 100 Ohm differential and 50 Ohm single-ended signaling. The board material is standard FR4 (E_r = 4.2 and tanδ = 0.02). All trace and plane layers are 0.5 oz. copper (0.65 mil thick). The electrical channel of interest for our case study is routed as fol-

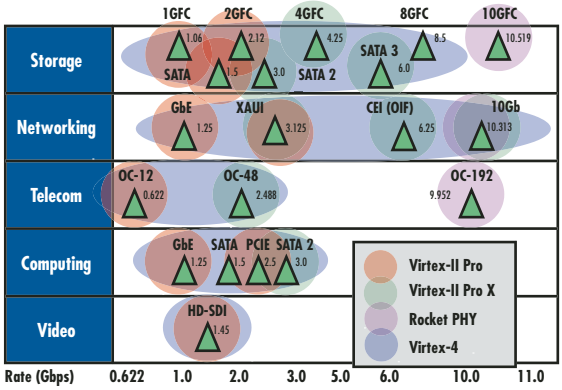


Figure 1 – Seamless scaling from 622 Mbps to 10 Gbps

Feature	Virtex-4	Benefit
Programmable Termination	Yes	Reduces reflections
Programmable Voltage Swing	Yes	Reduces power
Transmit Pre-Emphasis	Yes	Equalizes simple channels
Integrated AC Coupling	Yes	Direct interface to other devices, reduces component count
Receive Equalization	Linear and DFE	Equalizes stringent channel; allows legacy backplanes to be upgraded
Automatic EQ Settings Algorithm	Yes	Automatically finds optimum EQ setting for a given channel; eases design and ensures signal integrity

Figure 2 – Programmable pre-emphasis and equalization features in the Virtex-4 FX family

lows: microstrip on the top layer and transitions to layer 10 stripline through a GSSG differential via.

Differential Signal Topology

The differential signals are routed into and out of the board using Rosenberger™ high-performance coax-to-board SMA connectors. The signals are routed from the top-mounted connector to the FPGA using stripline transmission lines (layer 10), which transition to microstrip before interfacing with the FPGA BGA package. The actual trace layout for one Tx and Rx pair is shown in Figure 3.

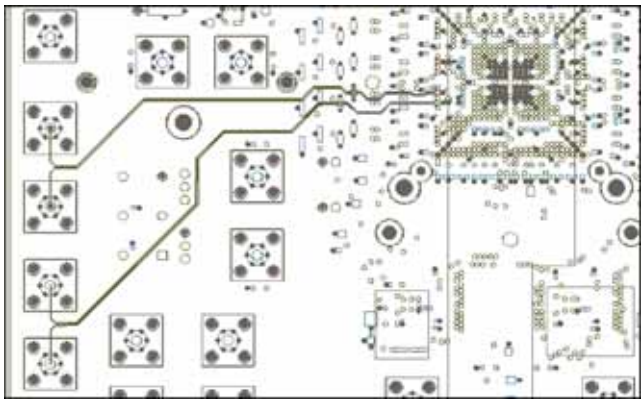


Figure 3 – Physical structure of a Tx and Rx differential pair on the MK322 board

Modeling and Simulation

The electrical channel comprises five main sections (Figure 4):

- The BGA package
- Microstrip transmission line
- Differential via (GSSG configuration, G- ground, S- signal)
- Stripline transmission line
- Connector

Let's look at each piece in turn.

BGA Package

The package model and the specific Tx pair of interest were extracted from the Cadence™ APD database and simulated using Ansoft HFSS. Figure 5 is a plot of the differential insertion loss (red) and return loss (blue) as computed by Ansoft HFSS.

For this particular differential pair, return loss is better than 15 dB, up to 22 GHz. Ansoft HFSS can output the differential S-parameters as Touchstone files. Typically, companies are reluctant to give out their package databases except under an NDA, because they contain sensitive design information. However, you can use S-parameters derived from the model for channel simulations.

Microstrip and Stripline Interconnect

We performed simulations for the stripline and microstrip structures using the two-dimensional quasistatic finite element simulator within Ansoft SI 2D Extractor. The stripline geometries were designed to provide nominally 100 Ohms differential impedance. Simulations confirmed that the impedance was within 7% of the nominal value (see Figure 6).

You can model PCB interconnects using various methods within Ansoft Designer™. The simplest is to use a coupled-line circuit model (like those found in popular high-frequency circuit simulators such as Ansoft Designer). In this instance, the interconnect is modeled with a uniform differential coupled transmission line without any discontinuities. On the other end of the modeling spectrum is the utilization of a full-wave planar EM field simulator based on the method of moments (MoM). Although accurate, MoM simulations are also the most computationally expensive method to predict interconnect performance.

A compromise that offers the accuracy of planar EM simulations with some of the speed of circuit simulation is offered by using a combination of the two. Figure 7 provides a comparison of the simulation results using the three different methods. As you can see in the figure, all methods predict similar performance. For an extended discussion of the trade-offs of the different approaches, please refer to the white paper accompanying the kit, available on the Xilinx SI Central website.

In addition, we parameterized each of the interconnect models. For example, in the microstrip interconnect model, the width, spacing, metal thickness, and physical length are parameters that can vary. For the initial simulations, these values were set to geometries specific to the MK322 board.

Differential Via

In keeping with good design practices that minimize unterminated stubs, layer 10 was used to transition from the microstrip

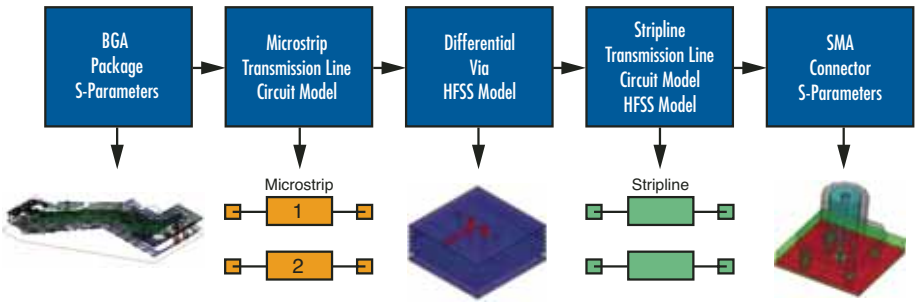


Figure 4 – The individual pieces comprising the full channel

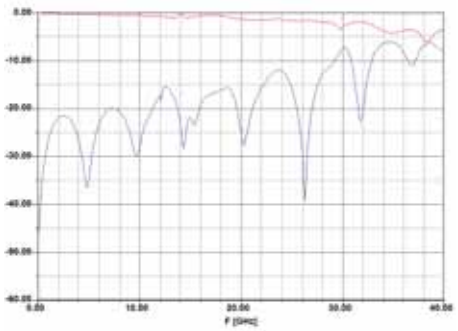
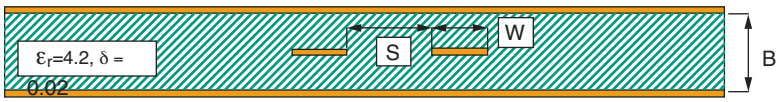


Figure 5 – Package model insertion loss (red) and return loss (blue) as computed by Ansoft HFSS



Layer	B	W	S	Zse	Zd	Zoom
S10	20.650	6.750	7.500	54.73	93.64	31.32

All dimensions are in mills

Figure 6 – Impedance for the stripline traces as extracted using Ansoft SI 2D Extractor

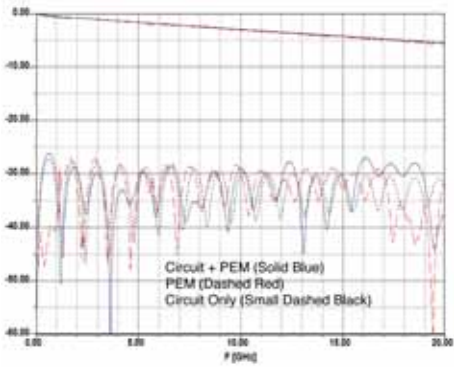


Figure 7 – A comparison of the three methods to simulate interconnects

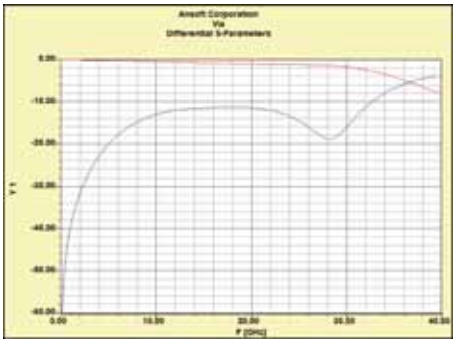


Figure 8 – Differential S-parameters for the via as computed by Ansoft HFSS

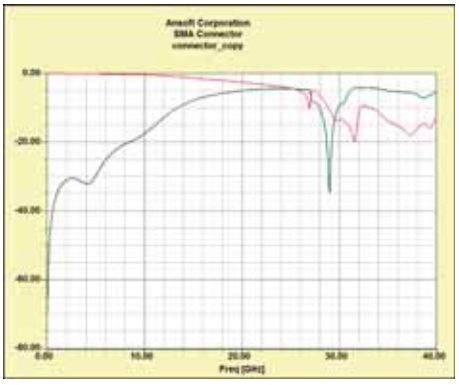


Figure 9 – Differential S-parameters for the SMA connector

to stripline using the throughhole differential via. The actual geometries for the ground-signal-signal-ground configuration were taken from Appendix D of the XFP specification (see pages 160-163 of the specification).

Several key variables for the via are parameterized, including spacing between signal vias, via radius, and antipad radius. Simulation results for the differential via structure are shown in Figure 8. The via structure shows excellent broadband insertion and return loss (> -10 dB) well beyond 20 GHz.

SMA Connector

The SMA connector used on the MK322 board is manufactured by Rosenberger (Part # 32K153-400). Rosenberger was gracious enough to provide us with the HFSS model for the connector, along with the optimized PCB footprint. The critical parameters for optimization involve the pad and antipad radii, as well as placement and spacing of several ground return vias around the center conductor. The ground vias around the center conductor allow the signal to transition from a radial coaxial field to a transverse electromagnetic mode (TEM) transmission line field in such a way that it minimizes any impedance mismatches. Figure 9 shows the insertion and return loss (> -10 dB up to 12 GHz) for the optimized SMA launch.

Full Channel Simulation

It is possible to cascade results generated from EM and circuit simulations on each of the individual components to get a full system simulation. Figure 10 is a snapshot of the schematic of the full channel, from the SMA connector, through the board to the Xilinx Virtex-II Pro X BGA package, set up for frequency domain analysis.

Figure 11 is a plot of the system simulation results displaying the insertion and return loss up to 40 GHz. As expected, the channel has a response similar to a low-pass filter. The majority of the energy for a baseband digital binary signal is contained within the first null of its power spectrum. For the rise time and signaling rate of this channel (30 ps, 10 Gbps), we are most concerned with the response up to 17 GHz. As seen in the plot, the insertion loss is roughly -10 dB and the return loss is below -10 dB up to 17 GHz.

You can also perform time domain simulations (see Figure 12) using the system simulator in Ansoft Designer. This simulator uses a convolution algorithm to process the frequency domain channel data with user-defined input bitstreams. Insertion and return loss is included in the simulation.

An ideal 10 Gbps pseudo-random bit source with a 0.5V p-p amplitude and 30 ps rise time was applied to the channel.

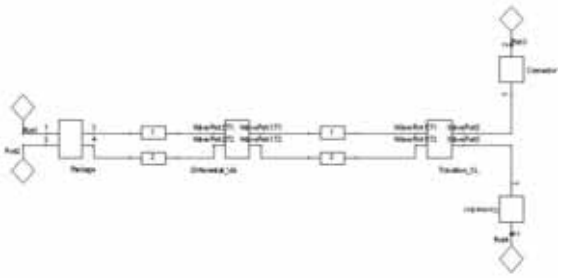


Figure 10 – Schematic of the full channel setup for frequency domain analysis within Ansoft Designer

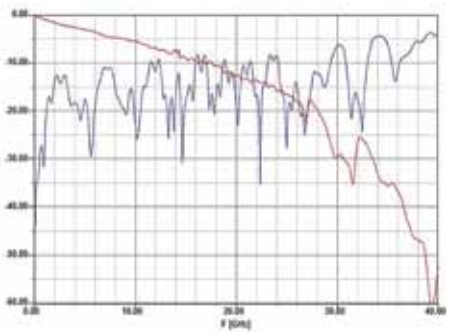


Figure 11 – Insertion and return loss for the full channel

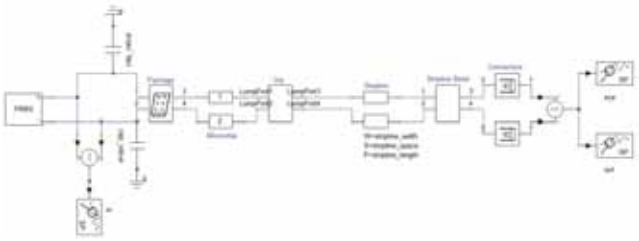


Figure 12 – Schematic showing setup for time-domain simulations

The channel was terminated in single-ended 50 Ohm impedances. The resulting eye diagram is shown in Figure 13, along with a measured eye diagram. There is excellent correlation between the measurement and simulation results. A very clear and open eye is achieved, as is expected from the frequency domain results.

For comparison to the measured eye, the driver capacitance was added to the channels. These capacitors are not part of the package model, because the passive channel will eventually be used with actual driver/receiver models that already include the capacitance. No pre-emphasis was used in the simulation. It should be anticipated that some pre-emphasis would sharpen up the time-domain response.

Extension of the Methodology

In creating the models, we emphasized that the critical variables that make up the physical structure are parameterized. Why parameterize? Although there are many reasons for doing so, let's show through some examples the power and utility of models that allow manipulation of critical variables.

A Longer Stripline Segment

In the original model, the nominal length for the stripline segment of the channel is 2.5 in. For whatever reason (board routing congestion is an obvious one), suppose that the stripline segment now needed to be 5 in. You can easily investigate the channel performance for this new scenario by changing the physical length variable (SL_L) in the model. Examples of such an analysis, for various trace lengths, are shown in Figure 14.

Increasing the length of the stripline segments results in significant eye degradation. Because every component of the channel is parameterized, you can explore the performance impact of different variables in each section of the channel when investigating design trade-offs. In fact, with exactly this intent in mind, we have made these models available as a Xilinx/Ansoft 10 Gbps Backplane Design Kit at *www.gigabitbackplanedesign.com*. Complete details on each of the models and the parameterized variables are available at this site.

Conclusion

Modern platform FPGA devices provide wide bandwidth processing and high-speed I/O. Serial I/O with speeds in the gigabit realm creates new challenges for PCB designers.

Models associated with this effort have been assembled into a 10 Gbps backplane design kit that you can use to predict performance of circuit board designs.

The design kit is available on the Xilinx "SI Central" website, enabling you to rapidly evaluate your own board designs. Visit *www.gigabitbackplanedesign.com* for more information.

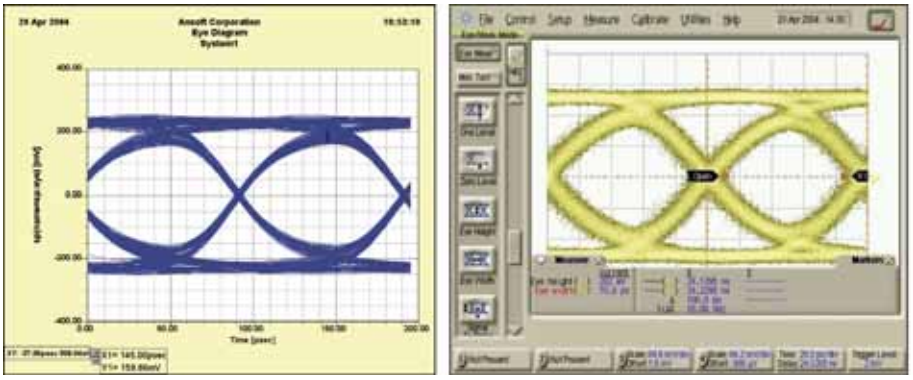


Figure 13 – Simulated (left) and measured (right) eye diagram for the full channel; the simulated eye is in excellent agreement with measurements

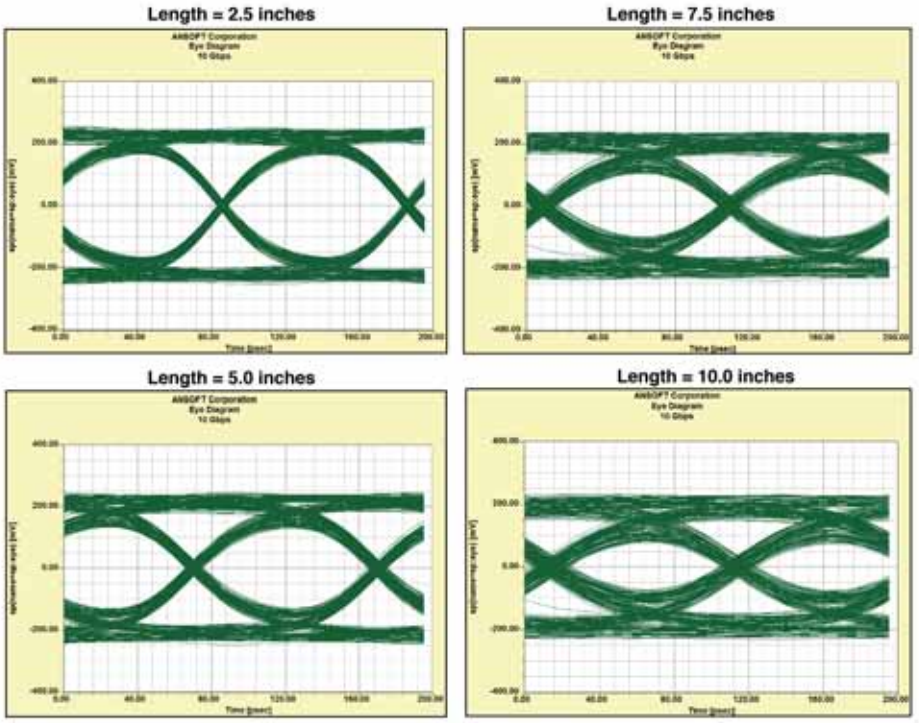
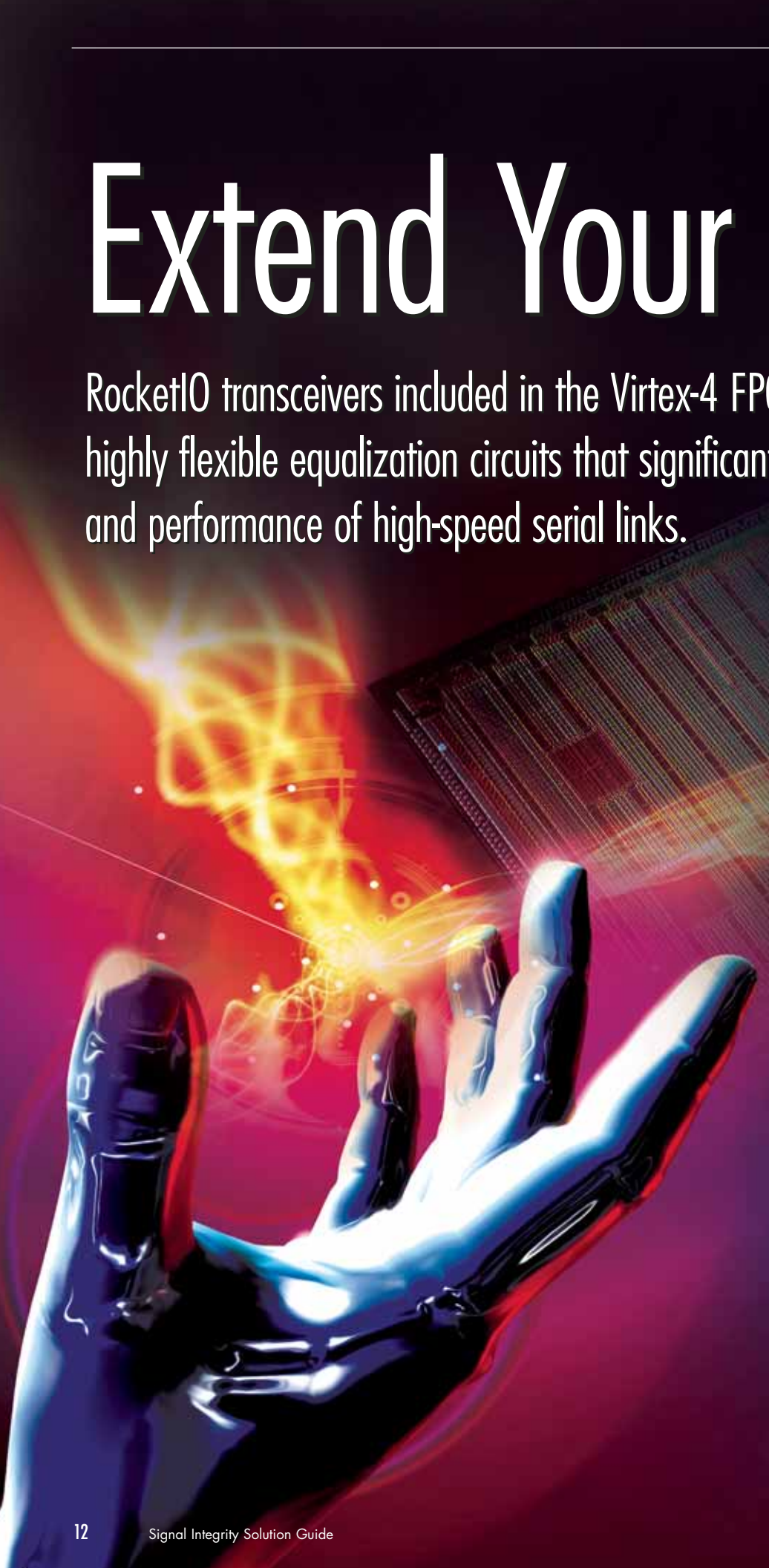


Figure 14 – Channel performance degrades due to losses in the transmission line as the trace length increases

Extend Your Reach

RocketIO transceivers included in the Virtex-4 FPGA family incorporate highly flexible equalization circuits that significantly extend the range and performance of high-speed serial links.



by Howard Johnson, Ph.D.
President
Signal Consulting, Inc.
howie03@sigcon.com

Mike Degerstrom
Senior Staff Signal Integrity Design Engineer
Xilinx, Inc.
mike.degerstrom@xilinx.com

Every multigigabit backplane, trace, and cable distorts the signals passing through it. This degradation may be slight or devastating, depending on the conductor geometry, materials, length, and type of connectors used.

Because they spend their lives working with sine waves, communications engineers like to characterize this distortion in the frequency domain. Figure 1 shows the channel gain, also called the frequency response, of a perfectly terminated typical 50 ohm stripline (or 100 ohm differential stripline). This stripline acts like a low-pass filter, attenuating high-frequency sine waves more than lower frequency waves.

Figure 2 illustrates the degradation inherent to a digital signal passing through 20 inches (.5 meters) of FR-4 stripline. The dielectric and skin-effect losses in the trace reduce the amplitude of the incident pulse and disperse its rising and falling edges. We like to call the received pulse, much smaller than normal, a "runt pulse." In a binary communication system, any runt pulse that fails to cross the receiver threshold by a sufficient margin causes a bit error.

For the purposes of this discussion, three things degrade the amplitude of the runt pulse in a high-speed serial link: losses in the traces or cables, reflections due to connectors and other signal transitions, and the limited bandwidth of the driver and receiver.

A classic test of dispersion appears in Figure 3. This particular waveform – adjusted so that the long flat portions of the test signal represent the worst-case, longest runs of ones or zeros available in your data code – displays the runt-pulse amplitude. In the absence of reflections, crosstalk, or other noise, this single waveform (as measured at the receiver) represents a worst-case test of channel dispersion. Longer traces introduce progressively more dispersion, eventually causing receiver failure at (in this example) a length of 1.5 meters.

One measure of signal quality at the receiver is voltage margin. This number equals the minimum distance (in volts) between the signal amplitude and the receiver threshold at the instant sampling occurs. In a system with zero reflections, crosstalk, or other noise you could theoretically operate with a very small voltage margin and still expect the system to operate perfectly.

In a practical system, however, you must maintain a healthy noise margin sufficient to soak up the maximum amplitude of all reflections, crosstalk, and other noise in the system, while still keeping the received signal sufficiently above the threshold to account for the limited bandwidth and noise inherent to the receiver.

Following the example in Figure 4, a runt-pulse amplitude equal to 85% of the nominal low-frequency signal amplitude exceeds the receiver threshold by only 35%, instead of the nominal 50%. A smaller runt pulse with amplitude 75% of the normal size would reduce the voltage margin by half – a huge hit to your noise budget, but still workable. For generic binary communication using no equalization, we would like to see the runt pulse arrive with amplitude never smaller than 70% of the low-frequency pulse amplitude.

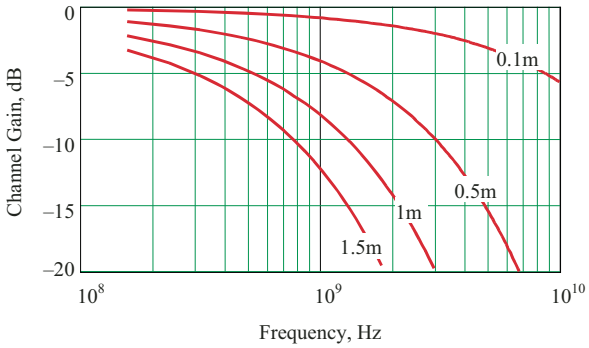


Figure 1 – The effective channel gain associated with a long PCB trace depends on the trace width, dielectric materials, length, and type of connectors used.

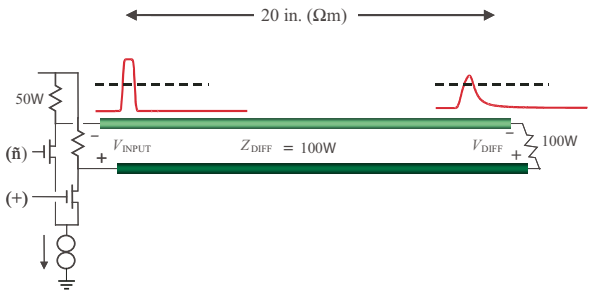


Figure 2 – Long traces reduce the amplitude of the input pulse and disperse its rising and falling edges.

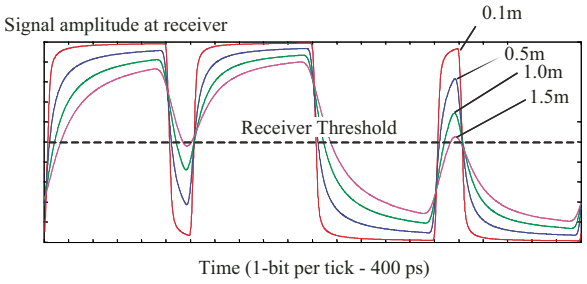


Figure 3 – This test waveform displays the worst-case runt-pulse amplitude.

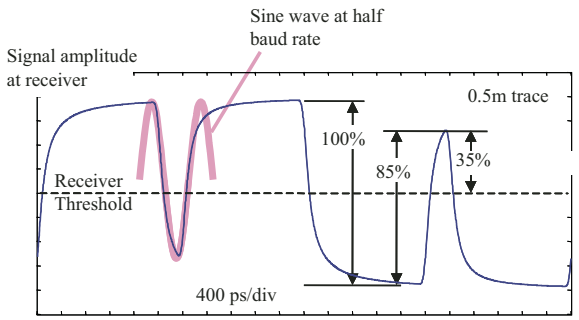


Figure 4 – A runt-pulse amplitude equal to 85% of the nominal low-frequency signal amplitude reduces the voltage margin above the threshold to only 35%, instead of the nominal 50%.

Runt-Pulse Degradation

On the left side of Figure 4 is a sine wave with a period of two baud. To the extent that the runt-pulse pattern (101) looks somewhat like this sine wave, you should be able to infer the runt-pulse amplitude from a frequency-domain plot of channel attenuation. Let's try it.

In Figure 4, the data waveform has a baud rate of 2.5 Gbps. One half this frequency (the equivalent sine wave frequency) equals 1.25 GHz. According to Figure 5, the half-meter curve gives you 4.5 dB of attenuation at 1.25 GHz. The same curve also shows 1.5 dB of attenuation at 1/10th this frequency, corresponding roughly to the lowest frequency of interest in an 8B10B coded data transmission system. The difference between these two numbers (-3 dB) approximates the ratio of runt-pulse amplitude to low-frequency signal amplitude at the receiver. With only -3dB degradation, the system satisfies our 70% frequency-domain criterion for solid link performance – precisely explaining why time-domain waveforms look so good at a half-meter.

Looking closely at Figure 4, the actual runt-pulse amplitude in the time domain is 85%, not quite as bad as the -3dB predicted by our quick frequency-domain approximation. This discrepancy arises partly from the harmonic construction of a square wave, where the fundamental amplitude exceeds the amplitude of the square wave signal from which it is extracted, and partly from the natural fuzziness inherent to any quick rule-of-thumb translation between the time and frequency domains. The simple frequency-domain criteria conservatively estimates these factors.

If your data code permits longer runs of zeros or ones than 8B10B coding, then you must use a correspondingly lower frequency as your "lowest frequency of interest." In the time domain, you will see the

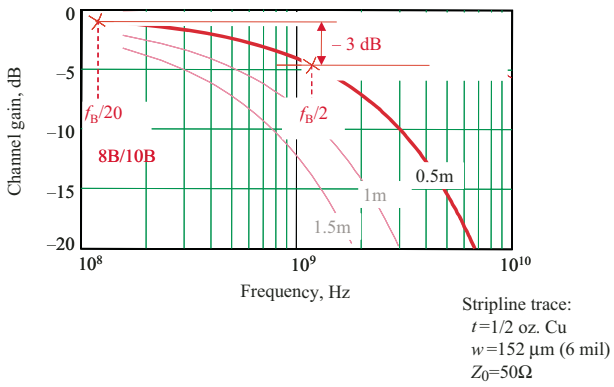


Figure 5 – The difference between high-frequency and low-frequency channel gain in this 2.5 Gbps system equals 3 dB.

received signal creep closer to the floor (or ceiling) of its maximum range before the runt pulse occurs, making it even more difficult for the worst-case runt pulse to cross the threshold.

As a rule of thumb, we look at the difference between the channel attenuation at the highest frequency of operation (the 101010 pattern) and the lowest frequency of operation (determined by your data coding run length) to quickly estimate the degree of runt-pulse amplitude degradation at the receiver. This simple frequency-domain method only crudely estimates link performance. It cannot substitute for rigorous time-domain simulation, but it can greatly improve your understanding of link behavior.

A channel with less than 1 dB of runt-pulse degradation works great with just about any ordinary CMOS logic family, assuming that you solve the clock skew problem either with low-skew clock distribution or by using a clock recovery unit at the receiver. A channel with as much as 3 dB degradation requires nothing more sophisticated than a good differential architecture with tightly placed well-controlled receiver thresholds. A channel with 6 dB of degradation requires equalization.

Transmit Pre-Emphasis

The Xilinx® Virtex™-4 RocketIO™ transceiver incorporates three forms of equalization that extend your reach on deeply degraded channels. The first is transmit pre-emphasis.

Figure 6 illustrates a simple binary waveform x[n] and the related first-difference waveform x[n]-x[n-1]. If you are familiar with calculus, you can think of the first-difference waveform as a kind of derivative operation. On every edge, the difference waveform creates a big kick. The transmit pre-emphasis circuit adds together a certain proportion of the main signal and the first difference waveform to superimpose the big kick at the beginning of every transition. As viewed by the receiver, each kick boosts the amplitude of the runt pulses without enlarging low-frequency portions of your signal, which are already too big.

The first-difference idea helps you see how pre-emphasis works, but that is not how it is built. The actual circuit sums not two but three delayed terms, called the pre-cursor, cursor, and post-cursor. This architecture gives you the capacity to realize both first and second differences by adjusting the coefficients associated with these three terms. Programmable 5-bit multiplying DACs control the three coefficients. The first and third amplitudes are always inverted with respect to the main center term, a trick that is accomplished by using the NOT-Q outputs of the first and third flip-flops. As an example, Figure 7 plots the frequency response corresponding to the particular coefficient set [-0.056, 0.716, -0.228].

Over the critical range from DC to 1.25 GHz, the pre-emphasis response rises smoothly – just the opposite of the plummeting curves drawn in Figure 5. The response peaks at 1.25 GHz. If you clock this pre-emphasis circuit at a higher data rate, the peak shifts correspondingly higher, always appearing just where you want it at a frequency equal to half the data rate.

Figure 8 overlays the pre-emphasis response with the channel response at 1 meter, showing a composite result (the equalized channel) that appears much flatter than either curve alone. In very simplistic terms, a flatter composite channel response should make a better-looking signal in the time domain.

The time-domain benefits of pre-emphasis appear in Figure 9. At shorter distances the signal appears over-equalized. The overshoot at each transition works fine in a binary system, assuming that the receiver has ample headroom to avoid saturation with the maximum-sized signal. At 1 meter, the signal looks quite nice, with very little runt-pulse degradation visible and (if you look closely) very little jitter. The 1.5 meter waveform now just meets the 70% criteria for runt-pulse success.

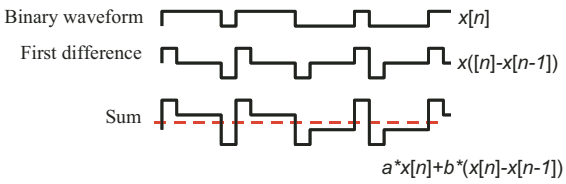


Figure 6 – The transmit pre-emphasis circuit creates a big kick at the beginning of every transition.

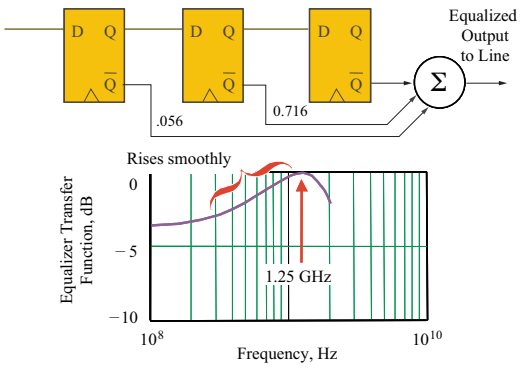


Figure 7 – Over the critical range from DC to 1.25 GHz, the pre-emphasis response rises smoothly.

Compared to a simple differential architecture, the pre-emphasis circuit has at least doubled the length of channel over which you may safely operate.

Linear Receive Equalizer

In addition to the pre-emphasis circuit, the RocketIO transceiver also incorporates a sophisticated 6-zero, 9-pole receive-based linear equalizer. This circuit precedes the data slicer. It comprises three cascaded stages of active analog equalization that may be individually enabled, turning on zero, one, two, or all three stages in succession.

Figure 10 presents the set of four possible frequency-response curves attainable with this receiver-equalization architecture. Each section of the equalizer is tuned to approximate the channel response of a typical PCB channel with an attenuation of about 3 dB at 2.5 GHz. With all stages on, you get a little more than 9 dB of boost at 2.5 GHz. Because the response keeps rising all the way to 5 GHz, this equalizer is useful for data rates up to and beyond 10 Gbps.

When setting up the equalizer, first select the number of sections of the RX linear equalizer that best match your overall channel response. Then fine-tune the overall pulse response using the 5-bit programmable coefficients in the transmit pre-emphasis circuit to obtain the lowest ISI, the lowest jitter, or a combination of both. After building the circuit, a clock phase adjustment internal to the receiver helps you map out bit error rate (BER) bathtub curves, so you can corroborate the correctness of your equalizer settings.

The flexibility provided by these two forms of equalization lets you interoperate with an amazing array of serial-link standards, meeting exact transmitted signal specifications and at the same time adding receiver-based equalization to keep your system working at the peak of performance.

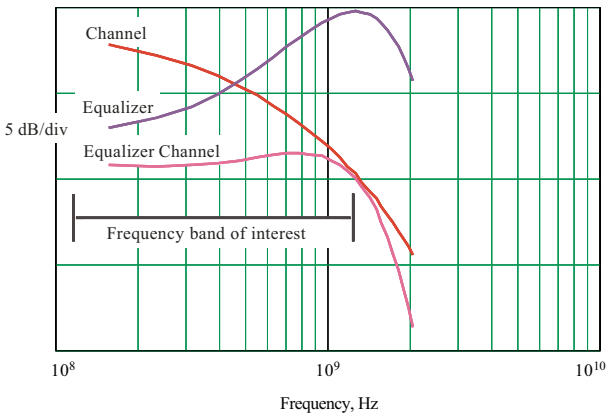


Figure 8 – Composing the pre-emphasis circuit with the channel produces an overall response much flatter than either curve alone.

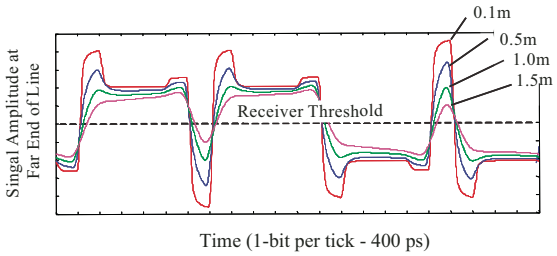


Figure 9 – A pre-emphasis circuit at least doubles the length of channel over which you may safely operate.

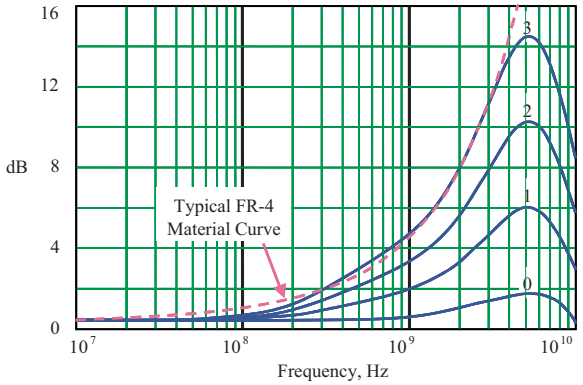


Figure 10 – The linear equalizer in the receiver may be set to one of four distinct response curves preprogrammed to match the response of various lengths of FR-4 PCB trace.

Decision-Feedback Equalizer

As a last defense against the slings and arrows of uncertain channel performance, the RocketIO transceiver includes a manually adjustable six-tap decision-feedback equalizer (DFE). This device is integrated into the slicer circuit at the receiver. The DFE is particularly useful with poor-quality legacy channels not initially designed to handle high serial data rates. It has the remarkable property of accentuating the incoming signal without exacerbating crosstalk.

Those of you familiar with signal processing will recognize that a DFE inserts poles into the equalization network, while a TX pre-emphasis circuit creates zeros. (A very accessible book about digital equalization, including DFE circuits, is John A.C. Bingham's "The Theory and Practice of Modem Design.")

Working together, the DFE, TX-pre-emphasis, and RX linear equalizer provide an incredibly rich array of possible adjustments.

Conclusion

For any channel with as much as 6 dB of runt-pulse degradation, a simple pre-emphasis adjustment easily doubles the length at which your link operates.

If you anticipate more than 6 dB of runt-pulse degradation, we strongly suggest that you simulate your system in detail before making the final equalizer adjustments. Contact your local Xilinx customer support office or visit the Xilinx website to obtain the necessary RocketIO models and associated design kits for modeling your channel. The modeling effort is well worth it, as equalization can substantially extend the reach of your circuits.

Howard Johnson, PhD, is the author of High-Speed Digital Design *and* High-Speed Signal Propagation. *He frequently conducts technical workshops for digital engineers at Oxford University and other sites worldwide. For more information, visit* www.sigcon.com, *or e-mail* howie03@sigcon.com.

Figures 1, 3, 4, and 9 are adapted with permission from Johnson and Graham, *High-Speed Signal Propagation: Advanced Black Magic*, Prentice-Hall, 2003.

Ten Reasons Why Performing SI Simulations is a Good Idea

by Austin Lesea
Principal Engineer, Advanced Products Group
Xilinx, Inc.

Not so long ago, the rise and fall times of signals, the coupling from one trace to another, and the de-coupling of power distribution on a PCB were tasks that were routinely handled by a few simple rules. Occasionally, you might use the back of an envelope, scribbling down a few equations to make sure that the design would work.

Those days are gone forever. Sub-nanosecond, single-ended I/O rise and fall times, 3 to 10 Gb transceivers, and tens of ampere power needs at around 1V have all led to increased engineering requirements.

Your choice is simple: simulate now and have a working result on the first PCB, or simulate later after a series of failed boards. The cost of signal integrity tools more than outweighs the cost of making the board over and over with successive failures.

In keeping with the theme of this special issue, here are my 10 best reasons why signal integrity engineering is a good idea:

1. You're tired of making PCBs over and over and still not having them work.

Seriously, without simulating all signals, as well as power and ground, you risk making a PCB that will just not work. IR (voltage) drop, inadequate bypassing or de-coupling, crosstalk, and ground bounce are just a few of the possible problems.

2. You're tired of being late to market and watching your competition succeed.

Every time you have to fix a problem with a PCB, it necessitates a new or changed layout, a new fabrication, and another assembly cycle. It also requires the re-verification of all parameters. Taking the time to do these things right has both monetary and competitive advantages.

3. You're tired of spending all this money, only to scrap the first three versions of PCBs and all of the components that went with them.

See reason number two.

4. Your eye pattern is winking at you.

If the eye pattern of a high-speed serial link is closing, or closed, it's likely that the link has a serious problem and will have dribbling errors – or worse, will be unable to synchronize at all. You must simulate every element of the design to assure an error-free channel.

5. All 1s or all 0s suddenly breaks the system.

Unfortunately, many systems do not have a choice of what data may be processed. Often the data pattern will create conditions that, if not simulated a priori, will cause errors in the system.

6. Hot and cold, fast and slow, and high and low voltages cause failures.

Without simulating the "corners" of the silicon used as well as the environmental factors, you're playing Russian Roulette with five of the six chambers loaded.

7. You cannot meet timing, and you are unable to find out why.

Poor signal integrity is the primary cause of adding jitter to all signals in a design. Ground bounce, crosstalk, and reflections all conspire to add jitter. And once added, jitter is virtually impossible to remove.

8. The FCC Part 15 or VDE EMI/RFI test fails every time you test a board.

Radiated and conducted radio frequency emissions, as well as susceptibility to radio frequency sources, is a sign of poor SI practices. Fixing the problem by shielding increases the system cost substantially, and may not even be possible in extreme cases.

9. Customers complain, but when you get the boards back, you don't find any problems.

One of the biggest problems with SI is that the errors and failures observed are difficult to correlate and sometimes impossible to find. Was it a problem with voltage, temperature, or with the data pattern itself? It might have been someone turning lights on and off (ground disturbance). Don't risk a return that cannot be fixed.

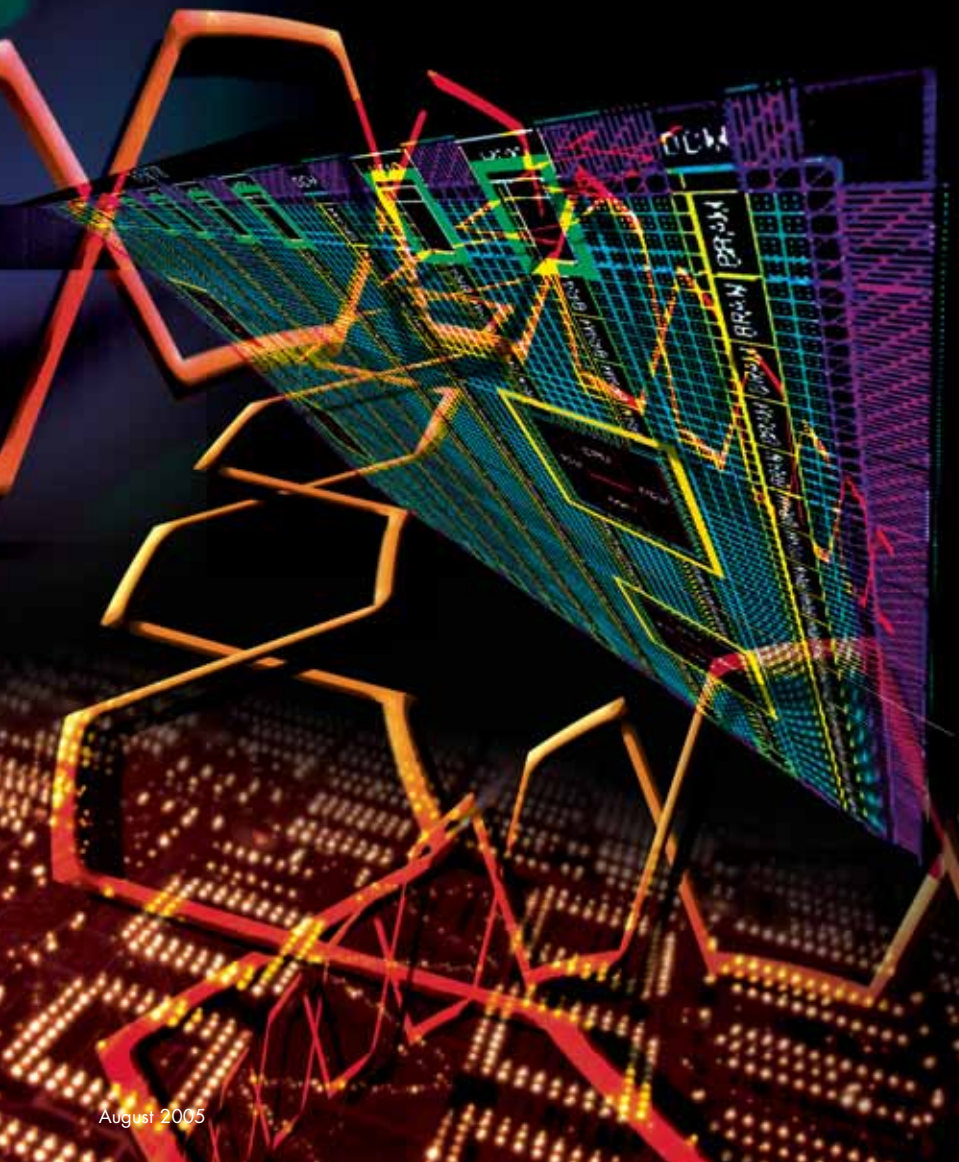
And last, but certainly not the least:

10. Your manager has suggested that you look for other employment.

Do not let this happen to you. Stay current, educated, and productive. Get the right tools to do the job. Realize that signal integrity engineering is a valuable and irreplaceable skill in great demand in today's design environments.

For Synchronous Signals, Timing Is Everything

Mentor Graphics highlights a proven methodology for implementing pre-layout Tco correction and flight time simulation with Virtex-II and Virtex-II Pro FPGAs.



by Bill Hargin
Product Manager, HyperLynx
Mentor Graphics Corp.
Bill_Hargin@mentor.com

We've all heard the phrase "timing is everything," and this is certainly the case for the majority of digital outputs on modern FPGAs. Timing-calculation errors of 10 or 20 percent were fine at 20 MHz, but at 200 MHz and above, they're absolutely unacceptable.

As Xilinx Senior Field Applications Engineer Jerry Chuang points out, "The toughest case usually is a memory or processor bus interface. Most designers know that they have to account for Tco (clock-to-output) as it relates to flight time, but don't really know how."

Another signal integrity engineering manager who preferred to remain anonymous explains, "We've got lots of things that hang on the hairy edge of working. That's one of the reasons why they give you so many knobs to turn on newer memory interfaces."

To complicate matters, manufacturer datasheets and application notes use multiple, often-conflicting definitions of many of the variables and procedures involved, requiring you to investigate the conventions used by manufacturer A versus manufacturer B. Most of the recently published signal integrity books either gloss over the subject or avoid it altogether. We hope that this article will serve to blow away some of the fog and reinforce some standard definitions.

System Timing for Synchronous Signals

An FPGA team will typically place and route an FPGA according to their specific timing requirements, leaving system-level timing issues to be negotiated later with the system-design team. With the sub-nanosecond timing margins associated with many signals, it's common for the system side to be faced with PCB floor-planning changes, part rotation, and sometimes the need to negotiate pin swaps with the FPGA team to accommodate timing goals. Proactive, pre-layout timing analysis and some careful accounting can keep both the FPGA and system teams from spending a month or more chasing timing problems.

Two classes of signals pose problems for FPGA designers and their downstream counterparts at the system level: timing-sensitive synchronous signals and asynchronous, multi-gigabit serial I/Os. We'll concentrate on parallel, synchronous designs in this article.

Margins

The system-timing spreadsheet for synchronous designs is based on two "classic" timing equations:

Tco_test(Max) + Jitter + TFlight(Max) + TSetup < TCycle

Tco_test(Min) + TFlight(Min) > THold

Or, once Tco_test is corrected, becoming Tco_sys, as outlined in this article:

Tco_sys(Max) + Jitter + Tpcb_delay(Max) + TSetup < TCycle

Tco_sys(Min) + Tpcb_delay(Min) > THold

Each net's timing is initially set up with a small, positive timing margin. This margin is allocated to the TFlight(Max) and TFlight(Min) values (or Tpcb_delay[Max] and Tpcb_delay[Min], respectively) in the preceding equations; these are timing contributions of the PCB interconnect between each net's driver and receivers.

If there is insufficient margin left to design the interconnects, either the silicon numbers need to be retargeted and redesigned, or the system speed must be slowed. Figure 1 shows how timing margins shrink relative to frequency.

There are two ways to come up with the interconnect values for the timing spreadsheet. Some signal integrity tools automatically make calculations that produce a single "flight-time" value. However, especially for designers just learning about the timing challenges of high-speed systems, a two-step approach is more instructive. First, you learn how to correct a datasheet's driver Tco value to match the behavior in your real system; second, you add the additional delay between the driver and each of its receivers.

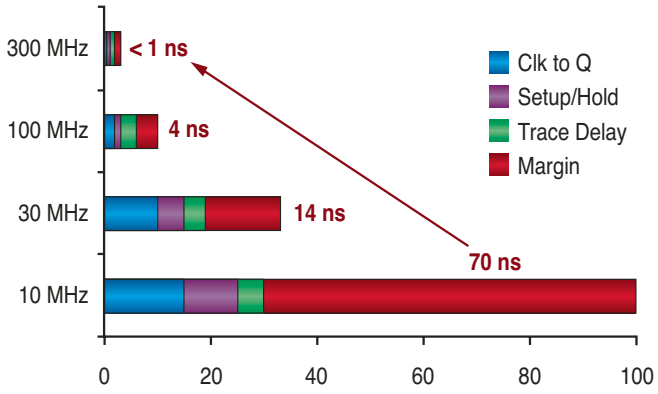


Figure 1 – Drastically narrowed system-timing margins, as clock frequency moves from 10 to 300 MHz, are shown in red.

Data Book Values

Initially, timing spreadsheets are populated with values from the silicon vendor's data book. You'll need first-order estimates from silicon designers on the values of Tco and setup and hold times for each system component. You can usually obtain this data from the component datasheet.

Test and Simulation Reference Loads

To arrive at the datasheet value for your drivers' Tco, standard simulation test loads (or reference loads) provide an artificial interface between the silicon designer and the system designer.

You'd prefer, of course, to have Tco specified into the actual transmission-line impedance you're driving on your PCB, but the silicon provider has no way of knowing what that will be. Knowing what loading the vendor assumed when publishing Tco is critical so that you can adjust for the difference between that load and your real one.

The Recipe for a Problem

As shown in Figure 2, if the reference load is significantly different from the actual load that the output buffer will see in your design, the sum of the datasheet and PCB-interconnect timing values will not represent actual system timing. Actual or total delay may be represented as:

Total Delay = Tco_sys + Tpcb_delay
≠ Tco_test + Tpcb_delay

where Tpcb_delay is the extra interconnect delay between the time at which the driver switches high or low until a given receiver switches.

Note that this "PCB delay" is not just the time it takes for a signal to travel along the trace (sometimes called "copper delay"

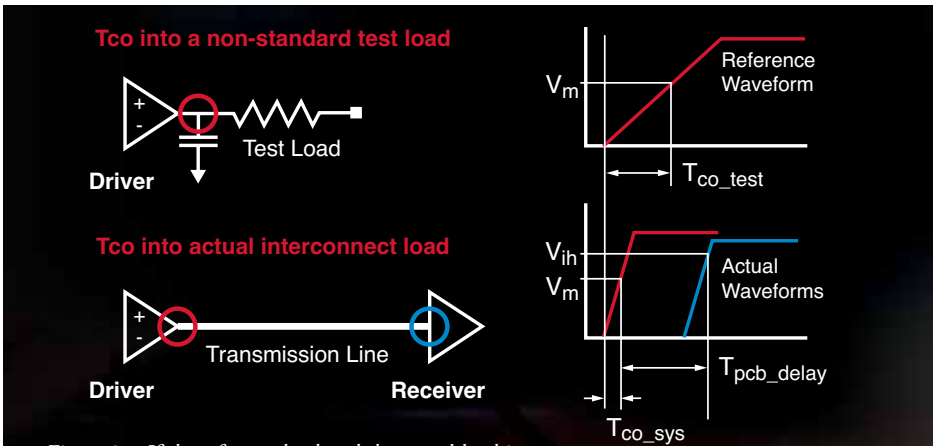


Figure 2 – If the reference load and the actual load in your design differ, you've got to make an adjustment in your system timing spreadsheet to compensate. The red driver waveforms illustrate the difference, and the impact, on Tco.

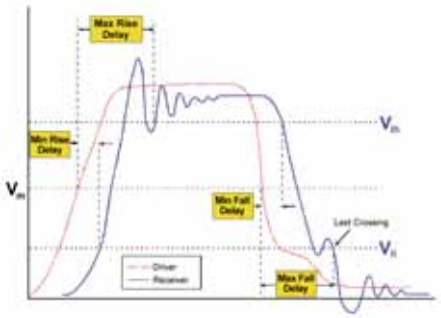


Figure 3 – "PCB delay" refers to the difference between the driver waveform switching through Vmeas and the waveform at the receiver as it switches through Vih (rising) or Vil (falling). Finding this value requires simulation, not just a simple "copper-delay" calculation.

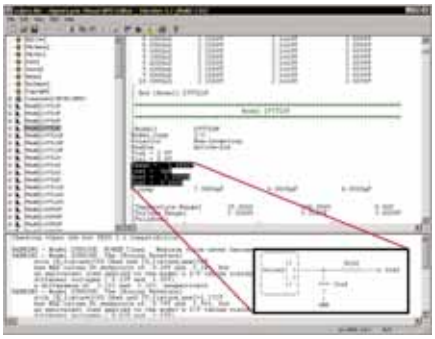


Figure 4 – Mentor Graphics' HyperLynx Visual IBIS Editor, a free tool for navigating the 50,000-plus lines of Xilinx Virtex-II Pro, Virtex-II, and Spartan IBIS models, shows reference load information for an LVTTL8F buffer as well as the assumed connections – from the IBIS specification – for Cref, Rref, and Vref in the insert.

or "propagation delay"). Here, Tpcb_delay accounts for effects such as ringing at the receiver, as shown in Figure 3. Its value could (on a poorly terminated net) easily be longer than the simple copper delay.

Calculating accurate timing involves more than finding Tpcb_delay. If the difference between Tco_sys and Tco_test is significant – even in the neighborhood of 100 ps – your board may not function properly if you don't account for the difference. But because Tco_test is a value created with an assumed test load, it almost never matches Tco_sys, the clock-to-output delay you'll see in your actual system.

For example, Lee Ritchey, author of "Get it Right the First Time" and founder of the consulting firm Speeding Edge, was hired to resolve a timing problem on a 200 MHz memory system. After digging into the design, he found that unadjusted datasheet values were used, based on Tco values that were measured on a 50 pF load rather than something resembling the design's 50 Ohm transmission-line load. As a result, this improper accounting "threw timing off by just over one nanosecond," he says. "That's 20 percent of the total timing budget, a major error."

In the following sections, we'll see how you can correct Tco_test to become Tco_sys, avoiding this type of error altogether.

The Process

Measuring Tco_test

To measure Tco_test, you need to set up a simulation with just the driver model and the datasheet test load. Though they're an optional sub-parameter in the IBIS specification, most IBIS models (including Xilinx IBIS models) contain a record of the test load (Cref, Rref, Vref) and the measurement voltage (Vmeas) to use with these values. Figure 4 shows these values for the LVTTL8F buffer in the Virtex-II Pro™ IBIS model, as well as a generic reference load diagram taken from the IBIS specification.

Once you've gathered these load values from the IBIS model, you simulate rising and falling edges, and for each, measure the time from the beginning of switching until the driver pin crosses the Vmeas threshold. These are the Tco_test values.

Obtaining "Tcomp," the Timing-Correction Value

Now you need to calculate a compensation value, Tcomp, that will convert the datasheet Tco value into the actual Tco you'll see in your system. Tcomp is the delay between the time the driving signal, probed at the output, crosses Vmeas into the silicon manufacturer's standard reference load, and the time it crosses Vmeas for your actual system load. Tcomp is then used as a modification to the Tco value from the vendor datasheet, as shown in Figure 5.

The revised computation of actual delay from the previous equation is then:

$$\text{Total Delay} = \text{Tco_sys} + \text{Tpcb_delay}$$
$$= (\text{Tco_test} + \text{Tcomp}) + \text{Tpcb_delay}$$

Note that Tcomp may be negative or positive, depending on whether the actual load in your system is smaller or larger than the standard test load. Traditionally, silicon vendors used capacitive test loads (like 35 pF) to measure Tco; almost all real PCB transmission lines do not present as heavy a load, so Tcomp is usually negative in this situation.

Xilinx, for its current generation of FPGAs, uses a 0 pF test load for output driver wave shape accuracy. Real transmission lines will represent a different load – some mixture of inductance, capacitance, and resistance. Because the transmission-line load is heavier than a 0 pF "open load," Tcomp will be positive. Simulation is the only way to accurately predict the exact value of Tcomp.

Simulating Tpcb_delay

At this point in the process, you've completed the first step in finding accurate delays for your timing spreadsheet, and you've compensated the datasheet Tco to match your real system load. Next, you need to determine Tpcb_delay, the additional delay caused by the interconnect from driver to receiver.

A signal integrity simulator is the only way to accurately do this, because only a simulator can account for subtle effects like reflections, receiver input capacitance, line loss, and so forth.

From here, we'll explore some detailed examples based on Xilinx-provided IBIS models – the process of calculating Tcomp and then using the HyperLynx™ simulator to determine an interconnect's Tpcb_delay through pre-layout topology analysis. You could enter the values that we come up with directly into your system-timing spreadsheet.

The process using Mentor Graphics' HyperLynx product is straightforward. You look up the manufacturer's test load in the IBIS model (see Figure 4), enter it in the LineSim schematic, set up your actual interconnect topology just below the reference load, and begin a simulation, probing at both drivers so that you can measure Tcomp and Tpcb_delay, as shown in Figure 6.

Running the Numbers on a Real Problem

An important design for an electronic equipment manufacturer had a Xilinx FPGA talking to a bank of SRAMs at 125 MHz, meaning the cycle time (Tcycle) was 8 ns.

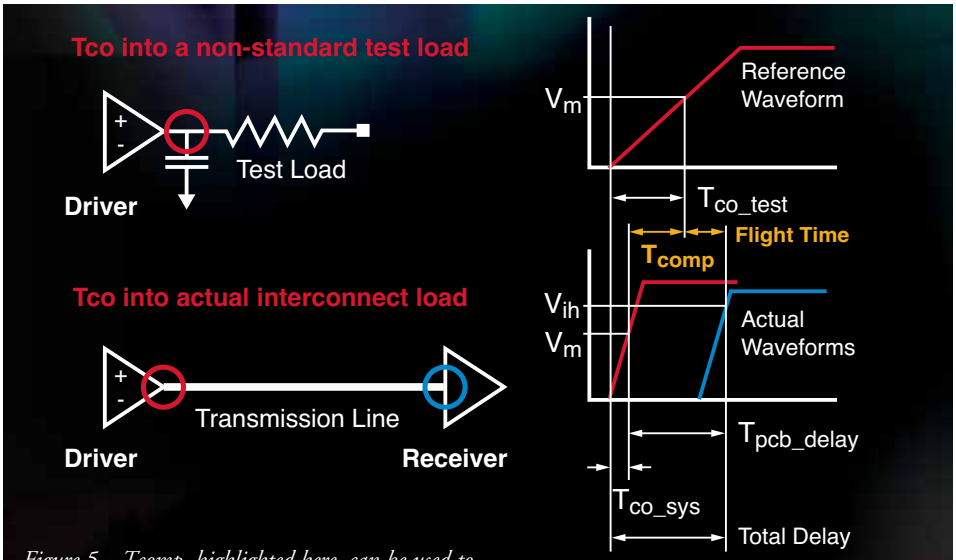


Figure 5 – Tcomp, highlighted here, can be used to "compensate" for data book Tco values in system timing calculations. Tcomp is positive when the actual load exceeds the reference load, and negative when the reference load is larger. Signal integrity tools actually use a one-step process that combines the effect of Tcomp and Tpcb_delay into a single value called "flight time" (see sidebar, "What is Flight Time?").

The Xilinx datasheet specified Tco as 4 ns (i.e., Tco_test). The SRAM's setup time was 2 ns.

Some of the traces connecting the FPGA to an SRAM were six inches long; a signal integrity simulation showed a worst-case maximum PCB delay (to the receiver's "far" threshold) of 2.5 ns. This yielded in the design's timing spreadsheet a total time of 4 + 2.5 + 2 = 8.5 ns (Tco_test + Tpcb_delay + Tsetup), violating the 8 ns cycle time.

However, the Tco value, when corrected for the actual design load, was 4-1.2 = 2.8 ns (Tco_sys = Tco_test + Tcomp), meaning that the actual total delay value was 2.8 + 2.5 + 2 = 7.3 ns (Tco_sys + Tpcb_delay + Tsetup), leaving an acceptable timing margin of 700 ps.

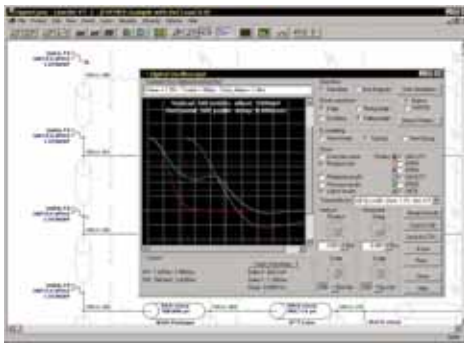


Figure 6 – Total Delay, Tco_test, Tcomp, Tco_sys, and Tpcb_delay, as well as flight time, are all measurable for this falling-edge waveform using Mentor Graphics' HyperLynx software.

Note that in this calculation, we measured to the time at which the receiver signal crossed the farthest-away threshold to get the worst-case, longest possible Tpcb_delay. For a rising edge, we measured to the last crossing of Vih; for a falling edge, to the last crossing of Vil.

Conclusion

For seamless interaction between the FPGA designer and the system designer, it's prudent to do as much pre-layout, "what-if" analysis as possible. And, though not covered explicitly in this article, you can also verify that your laid-out printed circuit boards meet your timing requirements using a post-layout simulator with batch analysis capabilities.

Some Mentor products that perform this type of analysis are HyperLynx, ICX, and XTK. Running these simulations, you're revising simulated representations of interconnect circuits in minutes as compared to the weeks required to spin actual PCB prototypes.

The new HyperLynx Tco simulator is available on Mentor Graphics' website, *www.mentor.com/hyperlynx/tco/*. Included with the Tco simulator are the Virtex-II Pro, Virtex-II™, and Spartan™ IBIS models; boilerplate schematics that will help you make adjustments to data book Tco values; and a detailed tutorial on Tco and flight-time correction that parallels this article.

What is "Flight Time"?

In this article, we've shown conceptually how Tco values specified into a silicon vendor's test load can be corrected on a per-net basis to give the actual clock-to-output (Tco) timing you'll see on your PCB, and then added to the additional trace delays between drivers and receivers to give accurate timing values. However, signal integrity (SI) tools actually deal with corrected timing values in a different (but equal) way.

The most convenient output from an SI tool is a single number – called "flight time" – shown in Figure 5 as (Total Delay - Tco_test) or (Tpcb_delay - Tcomp). You can add this value to the standard data book Tco values in your timing spreadsheet to give the same effect as the two-step process described in this article.

When an SI tool calculates timing values, it 1) simulates each driver model into the vendor's test load, measures the time for the output to cross the Vmeas threshold, and stores the value (Tco_test); 2) simulates the actual nets in the design and measures the time at which each receiver switches (Total Delay); and 3) for each receiver, subtracts the driver-switching-into-test-load time from the receiver time (Total Delay – Tco_test). The resulting flight time is a single number that can be added to each net's row in a timing spreadsheet, and that both compensates Tco_test for actual system loading and accounts for the interconnect delay between driver and receiver.

The term "flight time" is somewhat unfortunate, although it's become the industry standard. The name suggests the total propagation delay between driver and receiver, but the value calculated is actually the delay derated to compensate for the reference load. For old-style capacitive reference loads (e.g., 50 pF), flight time can even be negative.

Eyes Wide Open

The RocketIO Design Kit for ICX reduces the burden of implementing working multi-gigabit channels.

by Steve Baker
High Speed Architect, Systems Design Division
Mentor Graphics Corporation
steve_baker@mentor.com

If you're migrating from traditional bus standards such as PCI and ATA to serialized asynchronous architectures such as PCI Express™ and ATA-2, you've probably discovered that the tools for simulating the designs and models for the various buffers, connectors, transmission lines, and vias have become more complex.

Although setup and hold, crosstalk and single-ended delay are well understood, accurately modeling these new parts and their various complex behaviors adds to the job's complexity. To reduce the complexity of interacting with model and design parameters, Mentor Graphics and Xilinx have jointly developed the RocketIO™ Design Kit for ICX™ software, producing a design environment that allows you to fully confirm what's required to satisfy your design specifications.

The Design Kit

The RocketIO Design Kit for ICX is a companion to the standard Xilinx Signal Integrity Simulation (SIS) Kit and comprises a set of designs that match various Xilinx-supplied SPICE transmission line implementations. The kit is hierarchical, so all of the different elements – such as documentation, system configuration, simulation models, and ICX databases – are stored in different, relative location folders. These folders are located within the ICX kit in the same parent directory as the Xilinx SIS kit.

The design kit enables easy simulation analysis through the RocketIO menu and through existing features of ICX products, including eye-diagram, jitter, and inter-symbol interference analysis using predefined and custom multi-bit stimuli with lossy transmission line modeling.

Additionally, the IBIS 4.1 models, which ICX uses for simulation, reference the encrypted models supplied by Xilinx. You can progress from design to design through the kit's environment, learning more about the behavior of the RocketIO buffers with each design or simulation, such as what is achievable with these buffers in a multi-gigabit channel and what settings are required to maximize system performance.

> The custom menu is more full-featured, allowing direct simulation and eye diagram display of any of the 10 pairs from a single menu selection.

Standard Designs

The three standard designs supplied with the RocketIO Design Kit include:

- Correlation
- Example
- Evaluation.

You can also verify your own design, either in pre- or post-route states, in the kit's design area.

The Correlation Design

In a correlation design, the ICX database reproduces the interconnect scheme (Figure 1) from the Xilinx backplane example and uses the same drivers and receiver buffer models and parameters. The ICX database provides virtual "push button" operation so that you can run a signal integrity simulation and compare the resulting waveform with that provided in the Xilinx Rocket IO Design Kit in eye-diagram form. You can also verify that simulation results match those supplied by Xilinx with either the ICX self-contained simulation environment using ADMS SI or with HSPICE® as an external simulator called from within ICX.

The Example Design

The example design has an expanded set of transmission line examples to match the 10 examples that Xilinx supplies. Each of the 10 paths comprises a RocketIO transmitter connected to a Teradyne™ HSD five-row connector through two inches of differential board traces; 16 inches of differential board traces to a second Teradyne HSD five-row connector; and finally two inches of differential board traces from the second Teradyne HSD five-row connector to a RocketIO receiver.

The custom menu allows direct simulation and eye diagram display of any of the 10 pairs from a single menu selection. The menu also includes additional configuration and pulse train dialogs that you can use to change the simulation parameters, thus allowing investigations of RocketIO buffer behavior with these different settings and stimuli.

In the example design, because the transmission lines are fixed, you modify the various settings of the buffer itself and then conduct a simulation on whichever differential channel you want to investigate.

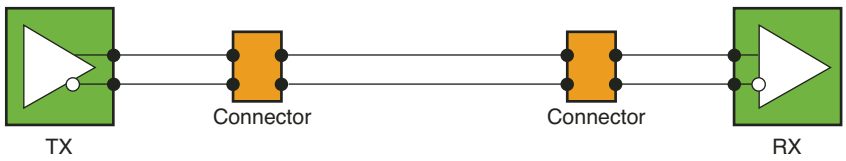


Figure 1 – Generic schematic of the design under simulation

The built-in RocketIO configuration utility allows changes to the temperature and bit duration settings when using the models directly from the Xilinx IBIS writer utility. It also gives you additional freedom to set the pre-emphasis level, driver/receiver termination values, and differential voltage swing when evaluating other possible solutions.

To enable different bit-patterns and speeds, you can also change the pulse train from the standard 3.125 GHz to your own specified pulse train using the pulse train generator. This utility allows you to specify bit patterns that can be used directly in ICX or exported as an ASCII file, in either SPICE PWL format or VHDL-AMS time vectors, toggling between state transitions.

The bit-patterns have an underlying pulse duration over which you can add jitter, where the peak-to-peak value specifies the six sigma points in picoseconds of this Gaussian random number. The pattern can be a user-defined set of ones and zeros, automatically defined as a random number of user-defined pattern length or as a pre-defined pattern. Pre-defined pattern styles include several pseudo-random bit sequences and Fibre Channel pulse trains (Figure 2).

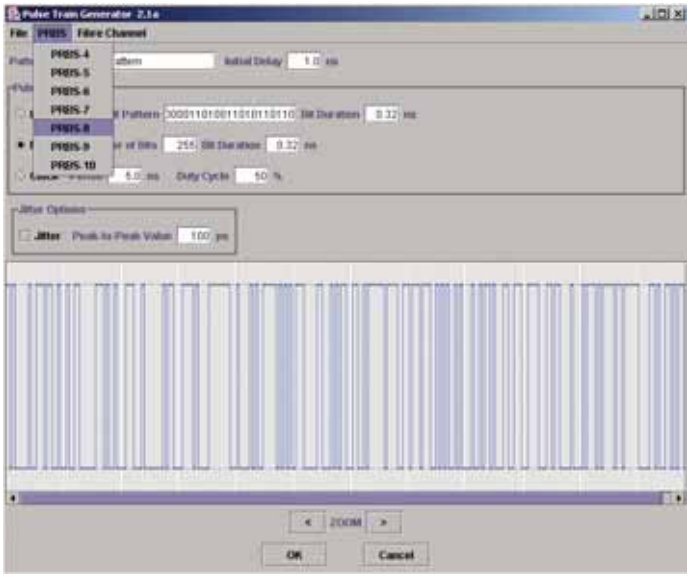


Figure 2 – Pulse train dialog showing pseudo-random bit pattern

The Evaluation Design

The evaluation design allows you to load a pre-defined cross section that matches one of the cross sections from the example design. In this virtual prototype environment, you can place actual parts, try "what-if" routing, and see the results in an eye diagram. As the IBIS part models include other buffers for Virtex-II Pro™ devices, you can simulate the whole of the FPGA rather than just the RocketIO channel.

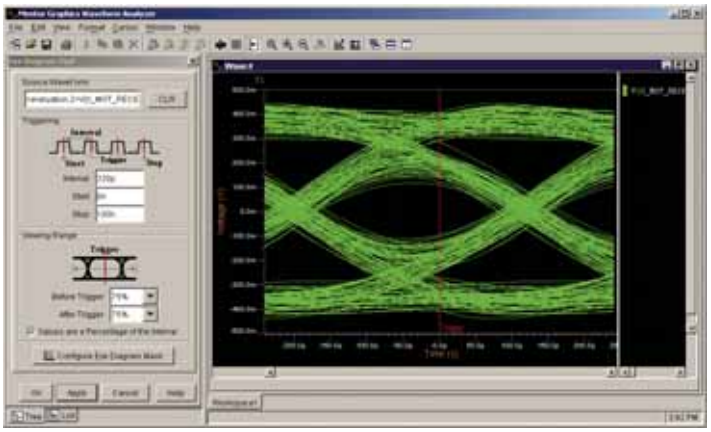


Figure 3 – A 3.125 GHz eye diagram from the evaluation design

This is where the channel's design is investigated in greater detail, as you initially place the devices to match your expected end design rather than using a fixed set of transmission lines. Using the electrical editor functionality of the IS floorplanner tool, you can add additional parts such as connectors or terminators and evaluate the impact of these on the resulting eye diagram. When working with these items, you can quickly determine the result of the different pre-emphasis settings. Additionally, you can see the impact of different routing strategies, including the fan-out pattern and tightly or loosely coupled differential pairs.

In the evaluation design, you can determine how much pre-emphasis is required to create the desired eye, as well as what level of noise is introduced on adjacent signals, on the board, or through the connector due to that level of pre-emphasis. The results of this virtual prototyping, as seen in the eye diagram in Figure 3, can be passed forward in the flow as constraints to drive the electrical design, as well as placement and routing examples.

Verification

The most advanced part of the kit allows you to simulate your design or system. The various parts of the system, backplane and plug-in cards, or just a single card with on-board channel, can be run through verification using the same complex pulse trains and model settings as before.

If required, you can modify settings to improve channel performance as measured by the eye. You can also define additional corner cases to evaluate best- and worst-case scenarios, including the impact of one pair on the other in terms of crosstalk; its impact on the shape and size of the eye; and the impact of other signals on the channel.

Conclusion

Iteration happens in any design process. The quicker decisions can be made in those iterations and the smaller the impact on existing design implementations, the happier we all are.

The RocketIO Design Kit for ICX allows you to make initial evaluations of the technology before any of the actual design implementation has occurred. As the design progresses forward from initial evaluations to the virtual prototype environment, you can confirm, in a pseudo-physical implementation, that the specifications can still be achieved, or use the kit to determine what changes are required to achieve the desired performance.

Finally, by verifying the placement, the routing of the multi-gigabit channels, or the whole design, you can confirm that you are within specification. For more information about the RocketIO Design Kit for ICX, visit *www.mentor.com/highspeed/resource/design_kits/icx-rocketio_designkit.html.*

Solving the Signal Integrity Challenge

Virtex-4 RocketIO transceivers bring blazing speed, and the ability to use it.

by Ryan Carlson
Director of Marketing, High Speed Serial I/O
Xilinx, Inc.
ryan.carlson@xilinx.com

The industry is moving away from parallel buses and relatively slow differential signals toward higher speed differential signaling schemes. These high-speed signals solve many design challenges: they offer new levels of bandwidth, they lower overall system cost, and they make designs easier by addressing the skew issues of large parallel buses.

However, with these improvements comes a new challenge: maintaining signal integrity. As signals push the limits of the media across which they are transmitted, the challenge of dealing with signal impairments becomes non-trivial, to say the least. The new Xilinx® Virtex-4™ RocketIO™ transceivers have incorporated multiple new features designed to solve this challenge.

Frequency-Dependent Loss

Several factors contribute to the frequency-dependent loss of a typical channel. Figure 1 shows the frequency response of 1 m of FR-4 trace. Dielectric loss and skin effect combine to create a significant loss above 1 GHz. With today's serial I/O standards approaching 10 Gbps, this loss becomes a critical design issue.

As a signal travels across a channel (like the one with a transfer function shown in Figure 1), a bit is degraded to the point where it interferes with neighboring bits; this is known as inter-symbol interference (ISI). Figure 2 shows the effect of ISI on a signal transmitted across a typical backplane channel. The high-frequency components are subject to losses that are greater than the low-frequency components. The edges that contain the high-frequency components are degraded, resulting in added jitter and eye closure. Additional techniques are needed to compensate for these losses.

Signal Integrity Features

The Virtex-4 RocketIO transceivers contain several features aimed at solving this problem. The first is transmit pre-emphasis. By modifying the signal before it is transmitted through a channel, transmit pre-emphasis can proactively compensate for some of the frequency-dependent loss of the channel.

Although most existing solutions use two-tap transmit pre-emphasis (addressing only the post-cursor ISI shown in Figure 2), the Virtex-4 RocketIO transceivers employ three-tap transmit pre-emphasis to address both pre- and post-cursor ISI. For signal rates above 3 Gbps, pre-cursor ISI becomes a non-negligible effect, and three taps of transmit pre-emphasis are needed to solve the problem.

In addition to transmit pre-emphasis, Virtex-4 RocketIO transceivers provide two different types of receive equalization. These options can be used in conjunction with transmit pre-emphasis to further improve signals degraded by lossy channels.

The first type of receive equalization works by amplifying the high-frequency components of the signal that have been attenuated by the channel (Figure 1). The transfer functions of this equalizer are programmable, and are shown in Figure 3.

The second type of receive equalization is called decision feedback equalization (DFE). This technique removes ISI effects by looking at consecutive bits and choosing the amount of equalization needed.

Both forms of receive equalization described above seek to amplify the high-frequency components of the desired signal. An advantage of DFE is that it does not amplify any crosstalk that may be associated with the signal. This technique can

therefore be useful for increasing the speed of legacy backplanes, where extensive crosstalk may exist.

All of these signal integrity features are fully programmable; they can be used independently or together, and each has multiple settings to equalize any channel. To fully take advantage of these hardware-based features, Xilinx also provides software-based reference designs that use bit error rate tests (BERT) to find the optimal settings for each unique application.

Integrated Receive Side AC-Coupling Capacitors

Many applications require AC-coupling capacitors to ensure compatibility between different Tx and Rx blocks. These capacitors require their own vias; at high speeds vias present yet another discontinuity to impair signal quality.

The Virtex-4 RocketIO transceivers integrate the AC-coupling capacitors on chip. This not only reduces external component count and design effort, but more importantly improves signal integrity by removing the need for extra vias in the board. These integrated AC-coupling capacitors can be optionally bypassed.

Conclusion

Signal integrity is an engineering challenge that accompanies the move to high-speed serial signaling. Once the system design has been optimized to minimize the physical effects of connectors, board materials, traces, vias, coupling capacitors, and cables, the remaining losses and channel effects need to be addressed by advanced silicon features.

Virtex-4 RocketIO transceivers are the industry's fastest integrated transceivers. Along with these leading-edge speeds, the RocketIO transceivers deliver multiple features designed to simultaneously address the signal integrity challenge that comes with them.

Xilinx has detailed information about high-speed design challenges, and the solutions available to solve them, at *www.xilinx.com/signalintegrity*. Instructional DVDs that describe various aspects of the signal integrity challenge can be purchased from the Xilinx online store by visiting *www.xilinx.com/store/*.

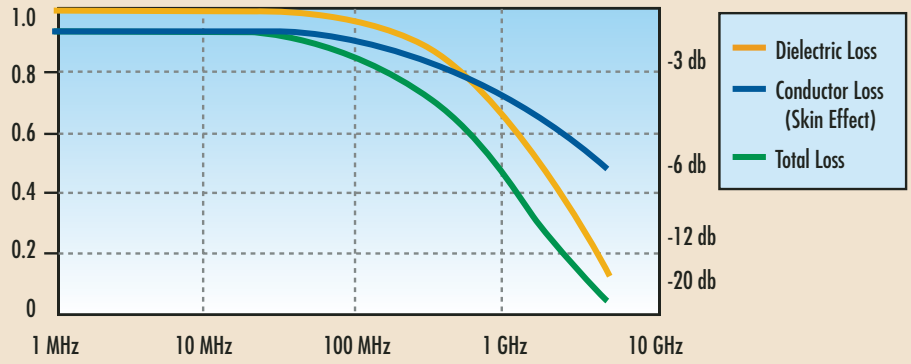


Figure 1 – Frequency-dependent loss

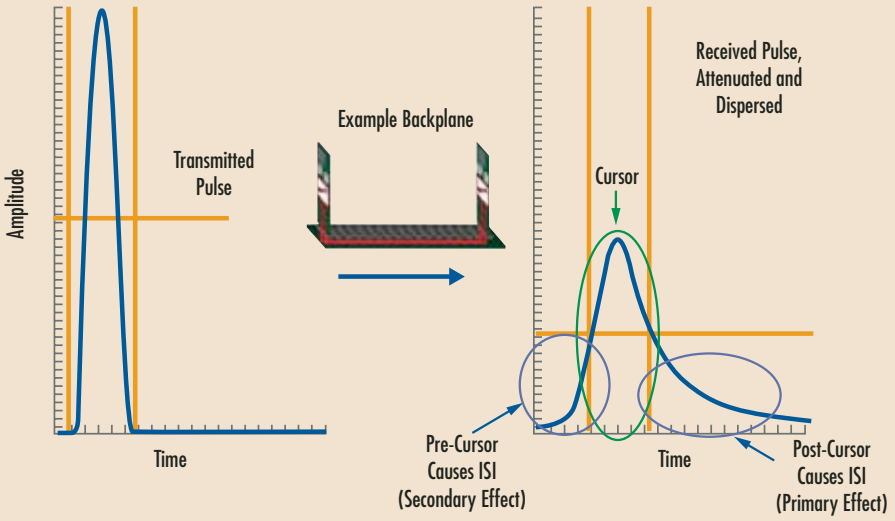


Figure 2 – A transmitted bit (left) and the result of inter-symbol interference (right)

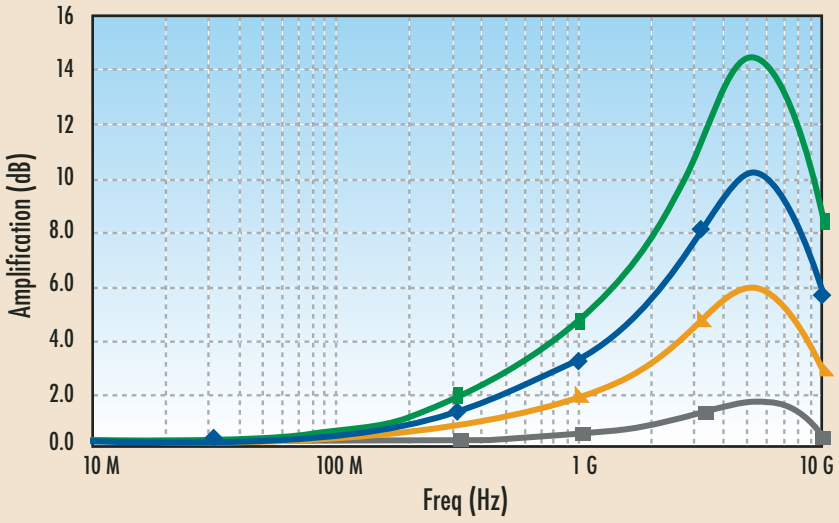


Figure 3 – Virtex-4 RocketIO receive equalization transfer functions

Xilinx/Micron Partner to Provide High-Speed Memory Interfaces

Micron's RLDRAM II and DDR/DDR2 memory combines performance-critical features to provide both flexibility and simplicity for Virtex-4-supported applications.

by Mike Black
Strategic Marketing Manager
Micron Technology, Inc.
mblack@micron.com

With network line rates steadily increasing, memory density and performance are becoming extremely important in enabling network system optimization. Micron Technology's RLDRAM™ and DDR2 memories, combined with Xilinx® Virtex-4™ FPGAs, provide a platform designed for performance.

This combination provides the critical features networking and storage applications need: high density and high bandwidth. The ML461 Advanced Memory Development System (Figure 1) demonstrates high-speed memory interfaces with Virtex-4 devices and helps reduce time to market for your design.

Micron Memory

With a DRAM portfolio that's among the most comprehensive, flexible, and reliable in the industry, Micron has the ideal solution to enable the latest memory platforms. Innovative new RLDRAM and DDR2 architectures are advancing system designs farther than ever, and Micron is at the forefront, enabling customers to take advantage of the new features and functionality of Virtex-4 devices.

RLDRAM II Memory

An advanced DRAM, RLDRAM II memory uses an eight-bank architecture optimized for high-speed operation and a double-data-rate I/O for increased bandwidth. The eight-bank architecture enables RLDRAM II devices to achieve peak bandwidth by decreasing the probability of random access conflicts.

In addition, incorporating eight banks results in a reduced bank size compared to typical DRAM devices, which use four. The smaller bank size enables shorter address and data lines, effectively reducing the parasitics and access time.

Although bank management remains important with RLDRAM II architecture, even at its worst case (burst of two at 400 MHz operation), one bank is always available for use. Increasing the burst length of the device increases the number of banks available.

I/O Options

RLDRAM II architecture offers separate I/O (SIO) and common I/O (CIO) options. SIO devices have separate read and write ports to eliminate bus turnaround cycles and contention. Optimized for near-term read and write balance, RLDRAM II SIO devices are able to achieve full bus utilization.

In the alternative, CIO devices have a shared read/write port that requires one additional cycle to turn the bus around. RLDRAM II CIO architecture is optimized for data streaming, where the near-term bus operation is either 100 percent read or 100 percent write, independent of the long-term balance. You can choose an I/O version that provides an optimal compromise between performance and utilization.

The RLDRAM II I/O interface provides other features and options, including support for both 1.5V and 1.8V I/O levels, as well as programmable output impedance that enables compatibility with both HSTL and SSTL I/O schemes. Micron's RLDRAM II devices are also equipped with on-die termination (ODT) to enable more stable operation at high speeds in multipoint systems. These features provide simplicity and flexibility for high-speed designs by bringing both end termination and source termination resistors into the memory device. You can take advantage of these features as needed to reach the RLDRAM II operating speed of 400 MHz DDR (800 MHz data transfer).

At high-frequency operation, however, it is important that you analyze the signal driver, receiver, printed circuit board network, and terminations to obtain good signal integrity and the best possible voltage and timing margins. Without proper terminations, the system may suffer from excessive reflections and ringing, leading to reduced voltage and timing margins. This, in turn, can lead to marginal designs and cause random soft errors that are very difficult to debug. Micron's RLDRAM II devices provide simple, effective, and flexible termination options for high-speed memory designs.

On-Die Source Termination Resistor

The RLDRAM II DQ pins also have on-die source termination. The DQ output driver impedance can be set in the range of 25 to 60 ohms. The driver impedance is selected by means of a single external resistor to ground that establishes the driver impedance for all of the device DQ drivers.

As was the case with the on-die end termination resistor, using the RLDRAM II

on-die source termination resistor eliminates the need to place termination resistors on the board – saving design time, board space, material costs, and assembly costs, while increasing product reliability. It also eliminates the cost and complexity of end termination for the controller at that end of the bus. With flexible source termination, you can build a single printed circuit board with various configurations that differ only by load options, and adjust the Micron RLDRAM II memory driver impedance with a single resistor change.

DDR/DDR2 SDRAM

DRAM architecture changes enable twice the bandwidth without increasing the demand on the DRAM core, and keep the power low. These evolutionary changes enable DDR2 to operate between 400 MHz and 533 MHz, with the potential of extending to 667 MHz and 800 MHz. A summary of the functionality changes is shown in Table 1.

Modifications to the DRAM architecture include shortened row lengths for reduced activation power, burst lengths of four and eight for improved data bandwidth capability, and the addition of eight banks in 1 Gb densities and above.

New signaling features include on-die termination (ODT) and on-chip driver (OCD). ODT provides improved signal quality, with better system termination on the data signals. OCD calibration provides the option of tightening the variance of the pull-up and pull-down output driver at 18 ohms nominal.

Modifications were also made to the mode register and extended mode register, including column address strobe CAS latency, additive latency, and programmable data strobes.

Conclusion

The built-in silicon features of Virtex-4 devices – including ChipSync™ I/O technology, SmartRAM, and Xesium differential clocking – have helped simplify interfacing FPGAs to very-high-speed memory devices. A 64-tap 80 ps absolute delay element as well as input and output DDR registers are available in each I/O element, providing for the first time a run-time center alignment of data and clock that guarantees reliable data capture at high speeds.

Figure 1 – ML461 Advanced Memory Development System

Xilinx engineered the ML461 Advanced Memory Development System to demonstrate high-speed memory interfaces with Virtex-4 FPGAs. These include interfaces with Micron's PC3200 and PC2-5300 DIMM modules, DDR400 and DDR2533 components, and RLDRAM II devices.

In addition to these interfaces, the ML461 also demonstrates high speed QDR-II and FCRAM-II interfaces to Virtex-4 devices. The ML461 system, which also includes the whole suite of reference designs to the various memory devices and the memory interface generator, will help you implement flexible, high-bandwidth memory solutions with Virtex-4 devices.

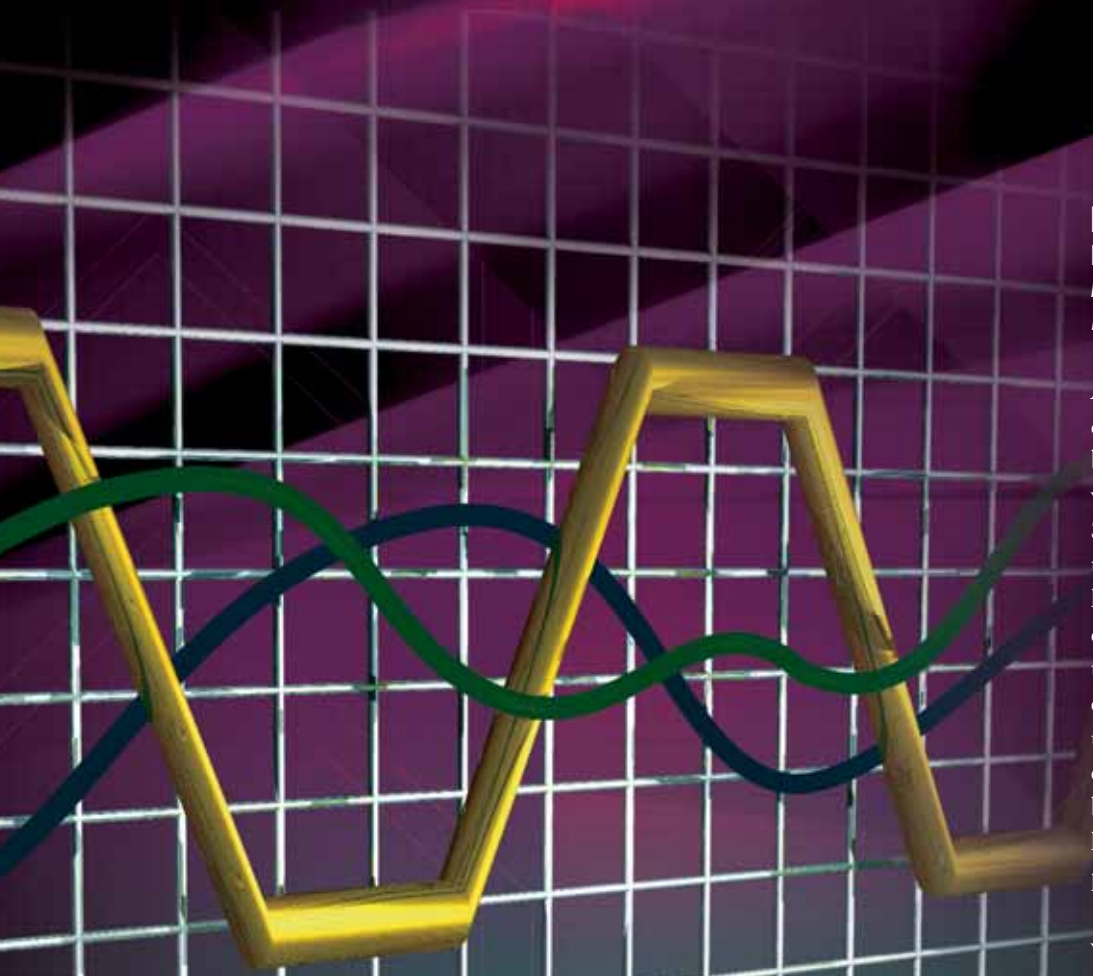
Please refer to the RLDRAM information pages at *www.micron.com/products/dram/rldram/* for more information and technical details.

FEATURE/OPTION	DDR	DDR2
Data Transfer Rate	266, 333, 400 MHz	400, 533, 667, 800 MHz
Package	TSOP and FBGA	FBGA only
Operating Voltage	2.5V	1.8V
I/O Voltage	2.5V	1.8V
I/O Type	SSTL_2	SSTL_18
Densities	64 Mb-1 Gb	256 Mb-4 Gb
Internal Banks	4	4 and 8
Prefetch (MIN Write Burst)	2	4
CAS Latency (CL)	2, 2.5, 3 Clocks	3, 4, 5 Clocks
Additive Latency (AL)	No	0, 1, 2, 3, 4 Clocks
READ Latency	CL	AL + CL
WRITE Latency	Fixed	READ Latency - 1 Clock
I/O Width	x4/ x8/ x16	x4/ x8/ x16
Output Calibration	None	OCD
Data Strobes	Bidirectional Strobe (Single-Ended)	Bidirectional Strobe (Single-Ended or Differential) with RDQS
On-Die Termination	None	Selectable
Burst Lengths	2, 4, 8	4, 8

Table 1 – DDR/DDR2 feature overview

Losing Less from Lossy Lines

Managing signal loss when designing with RocketIO transceivers.



by Bill Hargin
Product Manager, HyperLynx
Mentor Graphics
bill_hargin@mentor.com

A few months ago, I needed a few new hard drives: two desktop drives for home and a bigger 120 GB drive for my laptop, which was out of space. Browsing through a Seattle-area computer store, I compared the prices of Ultra ATA drives (older, parallel architectures) and Serial ATA (SATA) disk drives, discovering that the higher throughput of the SATA drives had resulted in price points that were twice those of the Ultra ATA drives. To me, this presented a crisp picture of the economic drivers behind the SERDES technology wave: higher throughput commanding a higher price, yet lower manufacturing costs.

You would think hardware designers would be making a mad dash toward serial design. However, I find that the "dash" toward SERDES seems to hover around 20 percent. Pondering this, I have come to believe that there are three primary reasons for the reticence among the remaining 80 percent:

1. Speed. Many applications are not pushing the speed envelope.
2. Resistance to change. Even innovative engineers are creatures of habit.
3. Real or perceived technical hurdles. SERDES design requires a different approach than wide, parallel bus design.

In this article, I'll address the third reason in detail. Although my focus is on Xilinx® Virtex™-II Pro RocketIO™ technology, the information would apply to any serial interface, including RocketIO transceivers in Virtex-4 devices.

Line Loss

An ideal lossless transmission line assumes that a signal propagates down the line with no energy loss. In other words, if a 1.0V signal with a 1.0 ns rise time enters one end of the line, the same 1.0V and 1.0 ns signal will come out the far end.

Figure 1 – Reciever waveforms for a lossless line, as well as 10, 20, and 40 in lines, showing both attenuation and rise-time degradation for a 2 GHz signal. (Simulated with Mentor Graphics HyperLynx.)

This is a good approximation when signal rise times are on the order of 1.0 ns (or slower), and with trace lengths of 10 in (or shorter). However, as rise times trend toward 100 ps and lengths get significantly longer (as in a backplane), the lossy effects of transmission lines begin to influence signal quality dramatically. As Figure 1 shows, these effects – both attenuation and rise-time degradation – vary directly with length at higher frequencies.

To describe and accurately predict the behavior of real interconnects, two important mechanisms that absorb energy from the signal must be modeled:

- Resistive loss. From DC through frequencies up to a few megahertz, the current in a trace moves through the entire cross-sectional area of the trace. At higher frequencies, however, current flows along the perimeter of a line rather than uniformly across the entire cross-section. As a result, the series resistance of the signal and return path conductors increases with the square root of frequency as the effective cross-section of the interconnect path is reduced. Resistive loss is also referred to as "skin effect." But no matter what you call it, it is the same phenomenon, and something you should be concerned about at high frequencies.
- Dielectric loss. The second important loss mechanism is dielectric loss, which is simply the conversion of electrical energy from the alternating electric field into heat. Dielectric loss, often specified in decibels per meter, increases with frequency and varies inversely with a material's "loss tangent" – a function of the material's resin type and molecular structure. Depending on resin content, "vanilla" FR-4 has a loss tangent ranging from 0.02-0.03. Lower loss tangent equates to more of the output signal getting to its destination, as well as higher material costs compared to FR-4. GETEK, for example, has a loss tangent of 0.012. Nelco 4000-13 is 0.01. And the loss tangent for Rogers 4003 is as low as 0.0027. For an actual design, you will want to discuss the tradeoffs with your board vendor.

Loss, Jitter, and ISI

I've often heard engineers use the terms inter-symbol interference (ISI), jitter, and loss to refer to the same thing: the unknown cause of a less-than-optimal signal or bitstream. In fact, these are different phenomena.

Random jitter is used to describe random events that result in a delay between the expected and actual signal transition. The distribution of random effects follows a classic Gaussian distribution, where the results vary wider (in time) as more data is observed. This data is typically provided by the driver manufacturer.

Deterministic jitter encompasses the list of systematic interconnect effects that will reoccur if you repeat the same stimulus. Dielectric and resistive loss, as well as crosstalk, reflections, via parasitics, return-path discontinuities, and any systematic aspect of an interconnect design contribute to deterministic jitter.

If the combined effects of random and deterministic jitter are significant enough, ISI will result, indicating that bit distinctions have become "blurry" at the receiver. This becomes particularly serious when rise-time degradation becomes comparable to the bit period of the signal. As a result, the shape of the received waveform will depend on the prior bit pattern (ISI).

These effects are best modeled using an oscilloscope that supports eye diagram analysis. Eye diagrams provide a visual display of the signal quality over many bit transitions with both deterministic and random jitter serving to close the "eye."

At a glance, an eye diagram will show if an interconnect is acceptable. Pass/fail criteria are often specified by an eye mask, shown as a blue hexagon on the left-hand side of Figure 2. Eye masks, which conform to the various SERDES specifications, define minimum and maximum keep-out regions where proper bit transitions should not appear for proper receiver interpretation of the driver's intent.

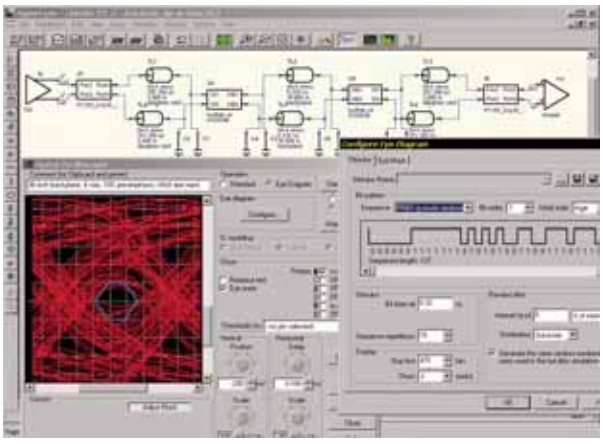


Figure 2 – The eye shows encroachment on the XAUI eye mask when using 10% pre-emphasis for a 36 in FR-4 transmission path with four vias and two Teradyne VHDM-HSD5 connectors (simulated with HyperLynx LineSim GHz).

Typically used bit patterns, or "pulse trains," include 8b/10b encoding and PRBS (pseudo-random bit stimulus). The 8b/10b data transmission scheme is considered ideal for high-speed local area networks and computer links. Realistic, long character sequences eventually hit some kind of worst-case history, but this can take a long time. A PRBS stimulus, as shown on the right-hand side of Figure 2, provides a means to force as much "action" on a serial data path as possible, in the smallest number of cycles. It is "pseudo-random" because it actually repeats after a pre-determined "n" number of bits.

Losing Less from Lossy Lines

Assuming that you have followed good routing rules (for example, achieving a consistent differential impedance of 100 ohms and avoiding excessive routing skew), there are five major ways to mitigate loss:

1. Reduce resistive loss by widening traces. Because resistive loss or "skin effects" are the result of a reduced cross-sectional area in a trace, wider traces result in a larger cross-sectional area, so the percent reduction due to resistive loss becomes smaller.

2. Reduce dielectric loss by shortening lines. Dielectric loss is a function of the material used and the length over which a signal is transmitted. Shortening the overall interconnect length (which may not be possible in many systems) can be an effective means of eliminating loss.

3. Reduce dielectric loss by employing lower loss tangent dielectrics. To reduce dielectric loss, more expensive materials – with lower loss tangents – can be considered, perhaps after exhausting less-expensive approaches.

4. Increase driver pre-emphasis. Boost the initial voltage level of each edge to compensate for high-frequency loss. (The trade-off here is additional power consumption.)

5. Equalization at the receiver. The incoming signal's lower frequency components are intentionally attenuated to artificially balance between high- and low-frequency components. The result is then amplified, with equalization of the low- and high-frequency signal components.

Although it is possible to estimate rise-time degradation and loss based on rules of thumb, the only way to get a realistic prediction of the impact from losses is to use a software simulator with the capability of simulating lossy lines. The HyperLynx Virtex-II Pro RocketIO Design Kit, along with HyperLynx simulation software, will enable you to simulate the preceding effects with Xilinx RocketIO technology.

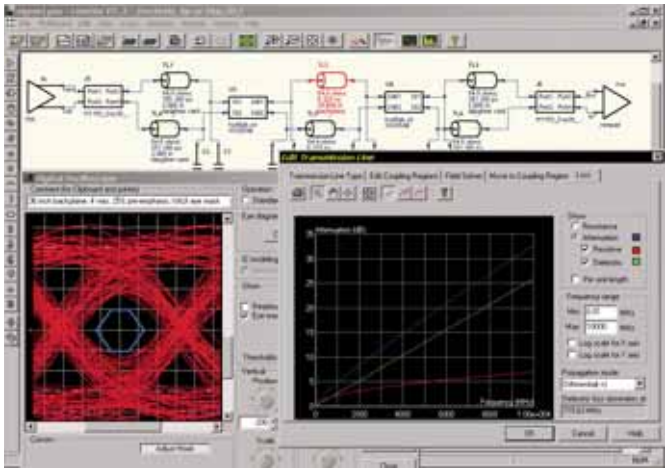


Figure 3 – Results with pre-emphasis bumped to 25% show significant improvement. Also shown is a graph of resistive and dielectric loss (dB) as a function of frequency (MHz) for the backplane (simulated with HyperLynx LineSim GHz).

The HyperLynx RocketIO Design Kit

In an effort to make multi-gigabit interconnect implementation as painless as possible, Xilinx and Mentor Graphics have teamed up to provide the RocketIO Design Kit for HyperLynx. Editable, pre-configured circuits in the kit are ready to simulate for both chip-to-chip applications and PCB backplanes – including connectors and pre-configured differential striplines.

The FR-4 traces used are a differential pair of centered striplines, 12 mils wide and 20 mils apart, with a 50 ohm characteristic impedance.

Figure 2 shows simulation results for the Xilinx backplane example with a 36 in transmission path and four vias. The eye in the figure shows encroachment on the XAUI (10 Gb Extended Attachment Unit Interface) eye mask when using 10 percent pre-emphasis. Here, we can clearly see the effects of ISI at the receiver.

Although you could engage any of the five primary loss-reduction mechanisms to open up the eye, let's assume that the 36 in backplane is a design requirement and that the four vias are inevitable. One of the remaining options is pre-emphasis, which can be bumped up to 20, 25, or 33 percent. Figure 3 shows the results at 25 percent pre-emphasis: the eye is wide open, which is great. With the wide margin around the XAUI mask's internal keep-out region, I would want to simulate this again at 20 percent pre-emphasis in an attempt to conserve power.

Conclusion

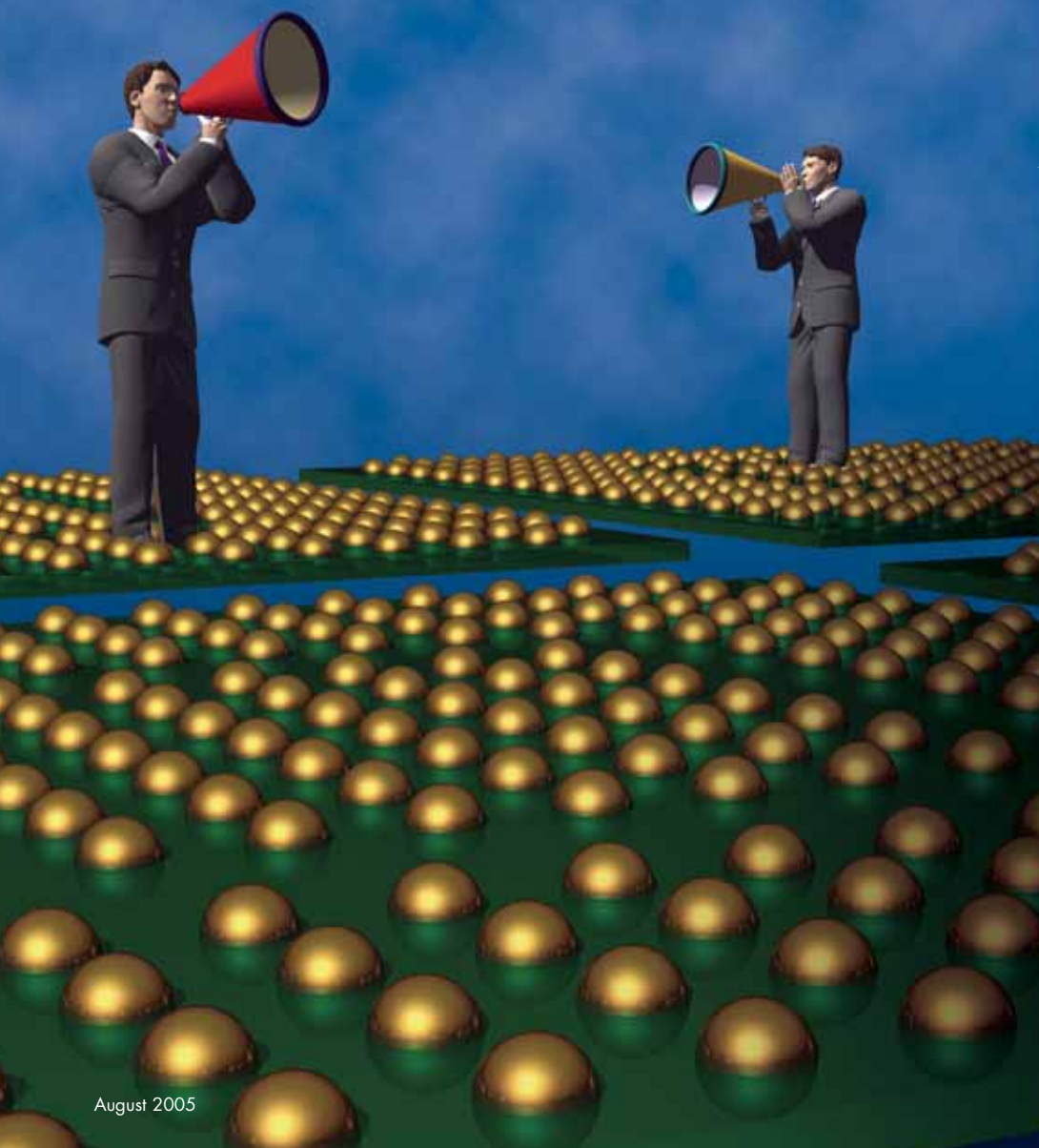
Although I simulated only one solution here, any permutation of the five ways to deal with loss is possible. Moreover, you could also examine non-ideal routing effects, including simulation of the impacts of differential length skew. With the availability of helpful simulation tools like HyperLynx, you don't need to fabricate prototypes – wondering whether you've over-designed or under-designed, spent too much or spent too little.

In a future extension of this theme, we'll revisit the Xilinx SIS Kit backplane design and examine the impact of the new equalization technology in the Virtex-4 implementation of RocketIO transceivers.

The HyperLynx RocketIO Design Kit is available from *www.mentor.com/hyperlynx.*

BGA Crosstalk

Xilinx Virtex-4 FPGA and Altera Stratix II FPGA



by Dr. Howard Johnson

Note: The following is an edited transcript of a TechOnline webcast presentation made on March 1, 2005[1] by Dr. Howard Johnson during which he discussed the details, measured lab results, and theory of crosstalk involving hundreds of outputs switching simultaneously in a high-speed Virtex™-4 FPGA package.

Measurement Setup

Thank you for inviting me to speak at this TechOnline forum.

This talk is about measurements, so I should begin by showing you how I made them. First, I obtained a test board comprising one Virtex-4 LX60 FPGA in the FF11480 package, plus an Altera® Stratix® II 2S60 FPGA in their F1120 package. On both these parts I arranged to bang a lot of outputs up and down, making a good deal of noise-hundreds of outputs switching at once.

While making all that noise I measured the crosstalk on one poor little guy that tried to remain "stuck at zero" or "stuck at one" during the test (Figure 1). This technique represents a realistic worst-case appraisal of the crosstalk emanating from any I/O driven out of the package during noisy events. It also indicates the crosstalk received by I/Os directed inwards towards the package. To see how this works we need to delve for a moment into the details of *inductive crosstalk*, because that is what this configuration delivers.

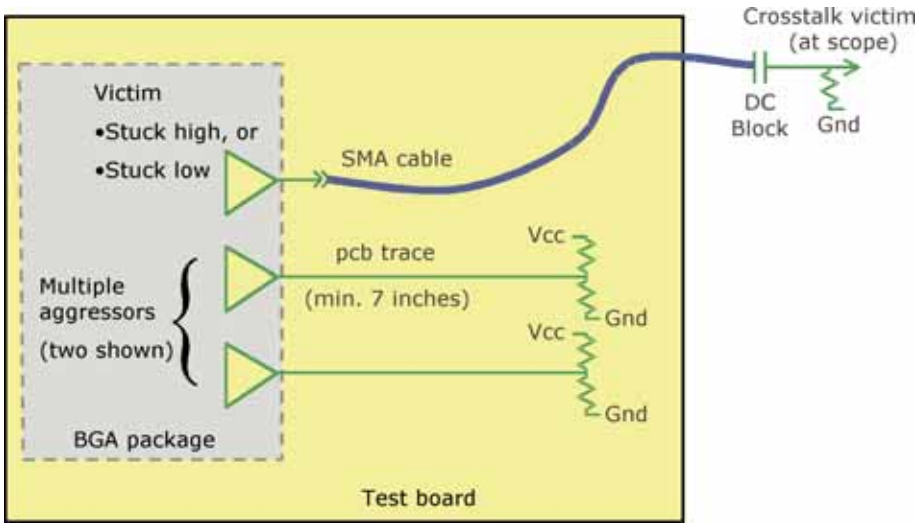


Figure 1 – Each FPGA on the test board transmits multiple aggressive signals while I observe noise emanating from a few static outputs.

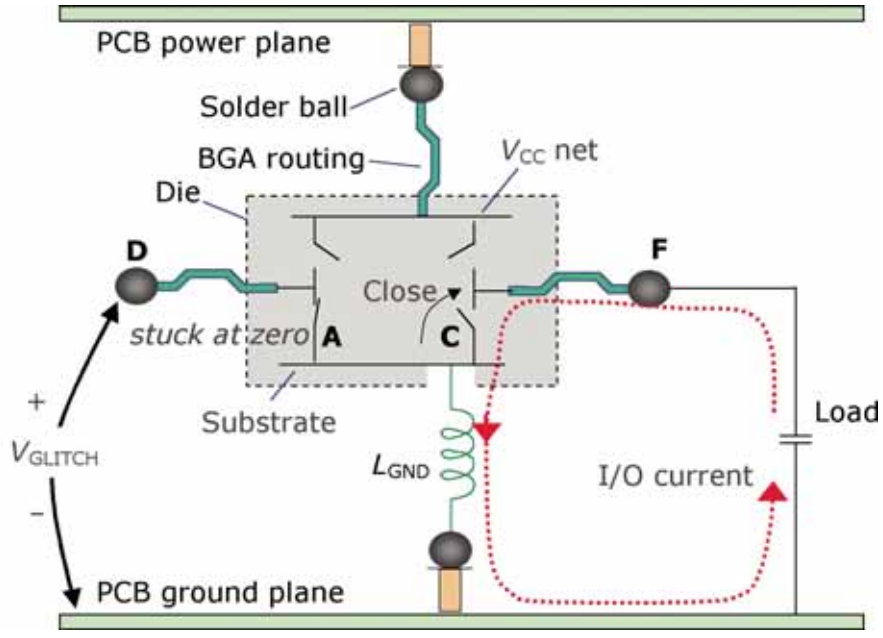


Figure 2 – This simplistic view of ground bounce ascribes inductance only to the ground connection.

Figure 2 illustrates a pair of PCB power and ground planes, between which I have connected a BGA package. Of course, the BGA package does not actually reside physically between the planes, but that is a convenient way to represent the circuit in a schematic view. The BGA balls and BGA routing appear in the diagram. The package holds two totem-pole drivers.

Suppose the load at **F** is initially charged HIGH. At time zero, switch **C** drives LOW, creating a huge I/O current transient. As this current flows through the finite inductance, L_{GND}, representing the matrix of ground balls underneath the BGA package, it creates a voltage disturbance on the chip substrate.

The victim **D**, which remains stuck low throughout the experiment, picks up this voltage glitch and transmits it straight out of the package to the scope. This simplistic view of the problem is a qualitatively good way to explain why switching events on one pin cause noise on another, but is quantitatively poor at predicting noise amplitudes.

To explain the inadequacy of the diagram, I will have to ask you to abandon the concept of inductance as a property of an individual wire or conductive pathway. You must instead perceive inductance as it truly exists: a property of the space between conductors.

In any high-frequency inductive problem, the relevant magnetic field resides in the space between conductors, not in the conductors themselves. This field, not the conductors, causes all inductive effects. In analog circuits, we are used to thinking of an inductor as a tightly wound coil. The coil concentrates its magnetic field in the space within the body of that inductive component.

The inductances we deal with are different. They occur as parasitic inductances, bound to magnetic fields that exist in the spaces between our signal and the return pathways. Figure 3 shows a better way to think about mutual inductance. This diagram shows the chip package on top, above the balls and vias. The three signal vias **D**, **E**, and **F** each penetrate the ground plane.

Switch **C**, when it closes, initiates the change in current drawn in red. This action fills the space between vias with an intense magnetic field. I have partitioned that field into three sections labeled L_1, L_2, and L_3. According to Faraday's law, the crosstalk measured at **D** varies in proportion to the total magnetic flux L_1 lying between via **D** and the nearest return-current position. Crosstalk at **E** varies with both fields L_1+L_2. Since L_1+L_2 exceeds L_1 alone, the crosstalk at **E** must exceed the crosstalk at **D**. That is, crosstalk varies strongly with the position of the victim via.

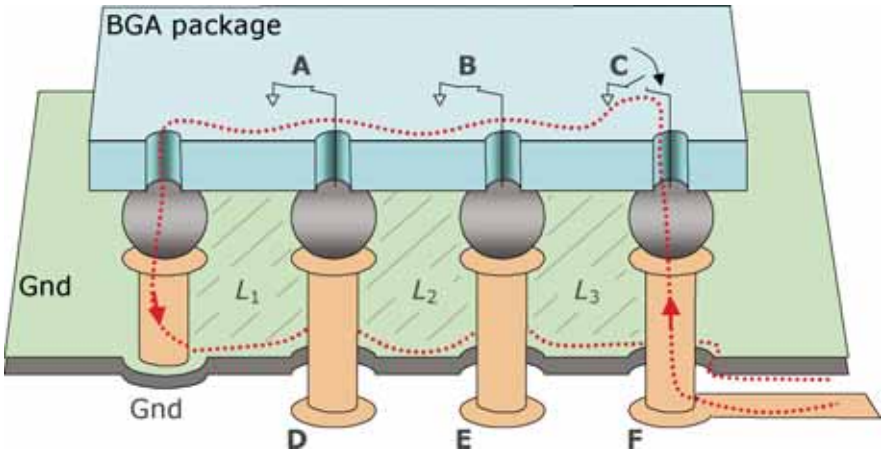


Figure 3 – Inductance exists not as a property of any individual wire or conductive pathway, but as a property of the space between conductors.

Measuring the crosstalk output from a stuck-at-zero (or stuck-at-one) I/O cell is a great way to determine the general level of BGA crosstalk input to receivers on that same package.

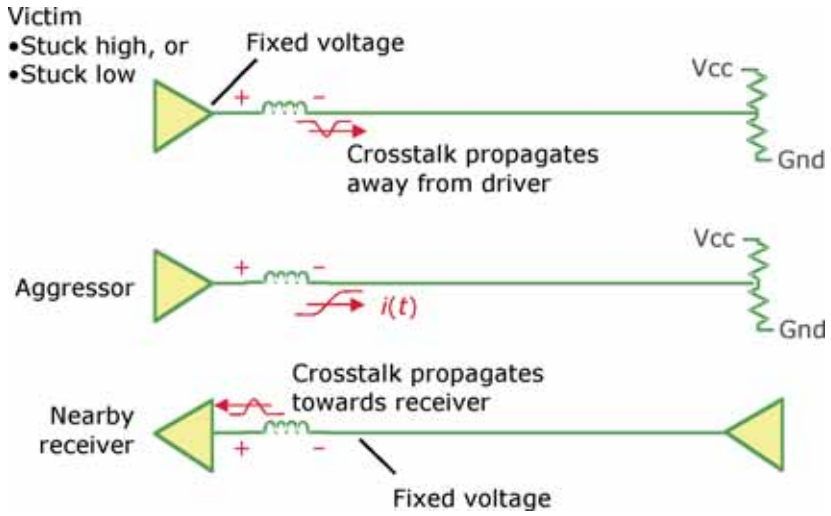


Figure 4 – Inductive crosstalk affects both drivers and receivers alike.

The variation in crosstalk with signal-pin position is not explained by Figure 2. To assign all the package inductance specifically to the ground pathway ignores the impact of signal positioning on observed crosstalk. In a real problem, ground bounce always varies with a signal's proximity to a good return connection. The signals closest to a good return suffer the least "ground bounce" or "power bounce".

In a complex multi-ground-pin package it is more accurate to think of ground bounce as a "crosstalk effect", rather than trying to partition the effect into discrete portions due to noise on the ground substrate, noise on the chip's internal Vcc rail, or noise accumulated in the ball-and-via field underneath the package.

This type of magnetic coupling varies with the separation between signal pathways and the length, or in this case, height = via + ball + package thickness. It also scales with the rise or fall time of the aggressive signal (specifically, the di/dt).

Capacitive crosstalk contributes very little to this setup. The measured results clearly support this, in that if there were significant capacitive crosstalk, when pin **F** switches LOW we would see pin **E** also glitch LOW. The fact that pin **E** glitches in the HIGH direction can only can only be ascribed to the inductive coupling explained in Figure 3.

If you understand how transformers work, Figure 4 makes it clear that inductive crosstalk affects both drivers and receivers alike. Inductive crosstalk acts like a low-impedance transformer hooked in series with your circuit. The primary of the transformer is the I/O current path. The various multiple secondary windings correspond to each of the victim vias. The inductive crosstalk voltage induced across each secondary is determined by the proximity of that secondary loop to the primary.

If a low-impedance connection (in the top case, a stuck-at-low driver) appears to the left of the transformer then the noise

voltages develop to the right. If the low-impedance connection (in the bottom case, an incoming line) appears to the right of the transformer then the noise voltages develop to the left, affecting the receiver. Either way, inductive crosstalk gets you.

Measuring the crosstalk output from a stuck-at-zero (or stuck-at-one) I/O cell is a great way to determine the general level of BGA crosstalk input to receivers on that same package.

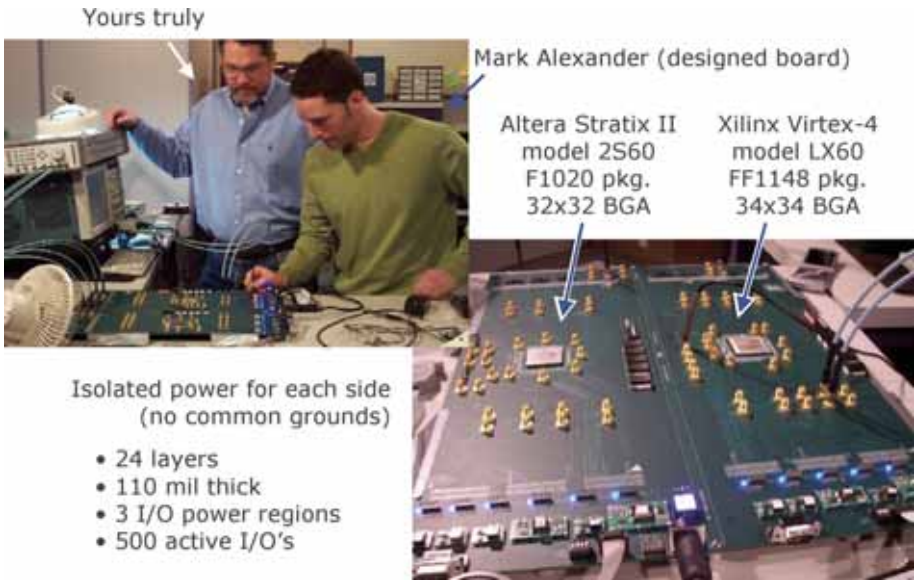


Figure 5 – On the left of the test board you can see the Altera Stratix II with 1020 balls, on the right is the Virtex-4 part with 1148 balls.

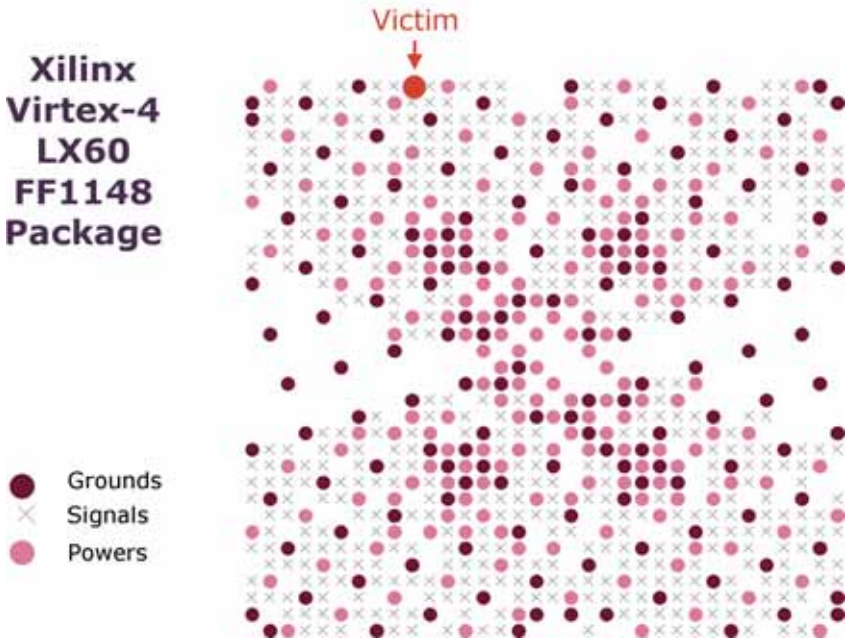


Figure 6 – The LX60 package includes a well-distributed array of power and ground balls to control crosstalk.

A picture of the test board appears in Figure 5. Mark Alexander did a fine job building this setup. It was designed as much as possible for a direct apples-to-apples comparison of the two parts. On the left you can see the Altera FPGA with 1,020 balls, on the right is the Xilinx® FPGA with 1,148 balls. The board has 24 layers, and is 110 mils thick. Three I/O voltage regions are included on the board (1.5, 2.5 and 3.3-volt) with each power plane sandwiched between grounds. To eliminate any concern about differences in power architecture, Mark came up with one common power arrangement that more than met both manufacturers guidelines, and used precisely the same setup on both parts. You can see the SMA fittings for viewing particular ports on each chip. Not every I/O is instrumented.

The Altera and Xilinx parts fit on opposite sides of the layout. All planes are completely isolated side-to-side, with separate power supplies. Not even the grounds touch, except through the instrumentation connections.

We ran all the tests shown here with only one side of the board active at a time, although we did test for interference between sides and saw no observable changes.

From the prior discussion, you should expect crosstalk to vary strongly with the proximity of the victim to a return pin. Building on that point let's look at the pattern of ground and power balls in the Virtex-4 LX60 FF1148 package.

Figure 6 depicts the array of return pins used in the FF1148 package. The dark dots represent ground balls; the lighter dots power balls.

Assuming perfect decoupling within the BGA package, the power and ground balls are equally effective as conduits for returning signal current. For the rest of this paper I will not differentiate between power and ground balls, but simply call them "return balls". The other marks (X) represent high-speed signals. Blank spots indicate low-speed programming signals, PLL control signals, and other non-high-speed functions.

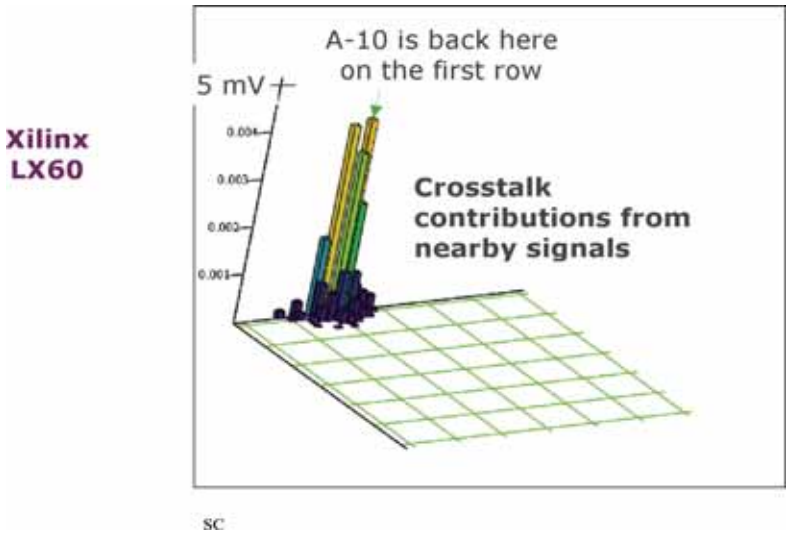


Figure 7 – Only the nearest aggressors contribute significantly to aggregate crosstalk at position A10.

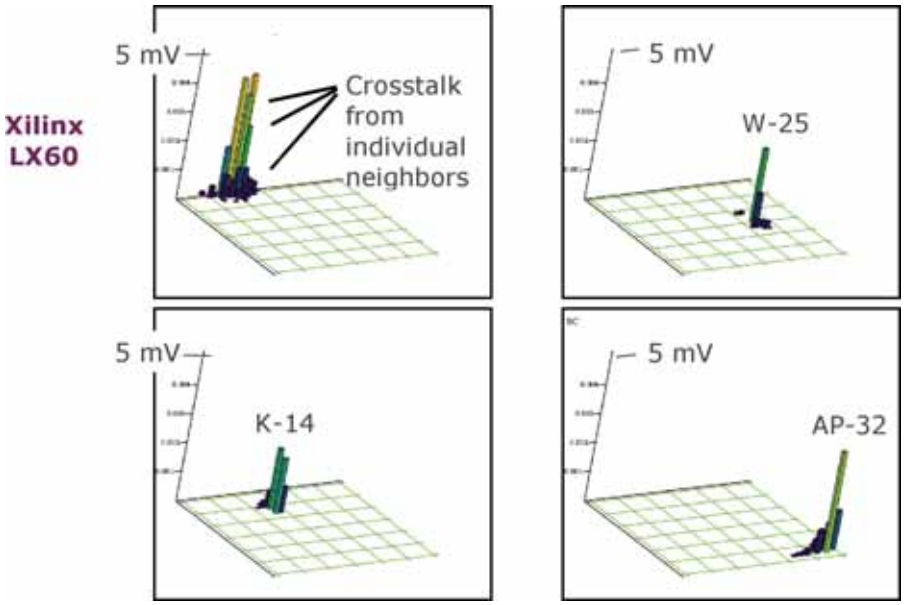


Figure 8 – No matter where you go, the main contributors to crosstalk always lie tightly grouped around the victim.

Some of the blank spaces on the left and right sides of the array are not populated in the Virtex-4 LX60 FF1148 package. We picked this configuration because it has about the same total number of active high-speed signal pins (500) as the Altera Stratix II 2S60.

The existence of all those return balls helps tremendously to reduce crosstalk from distant aggressors. If you were to do a "walking ones" test, moving the aggressor further and further away from the victim each time while observing crosstalk on the victim during the whole test, you would see crosstalk fall off geometrically each time you pass a return ball position.

To illustrate that point, let's look in detail at the distribution of crosstalk associated with one specific I/O pin. I will pick signal A10, which is up on the top row, about a third of the way in. By the way, I picked this pin for today's studies because our calculations predict pins located near the edge of the pattern should be a worst-case location for crosstalk performance.

Because there are so many power and ground balls in this package, and so evenly distributed, the crosstalk contributions around the particular signal A10 are tightly grouped around that particular location (Figure 7).

... moving the aggressor further and further away from the victim each time while observing crosstalk on the victim during the whole test, you would see crosstalk fall off geometrically each time you pass a return ball position.

Figure 8 examines a number of other locations. The story is the same in each case: crosstalk falls off geometrically as you pass beyond each successive ring of power or ground pins.

Some details behind the simulation used to generate these pictures are that the drivers are set to a di/dt of 9.05E+06 A/s, corresponding to a Xilinx 1.5V LVCMOS 4 ma fast driver, and all vias are set to an average depth of 0.035 in. below the surface of the PCB.

Since both power and ground pins act equally as conduits for returning signal current in this package, what we really have here is a well-distributed 4:1 ratio of signals to returns.

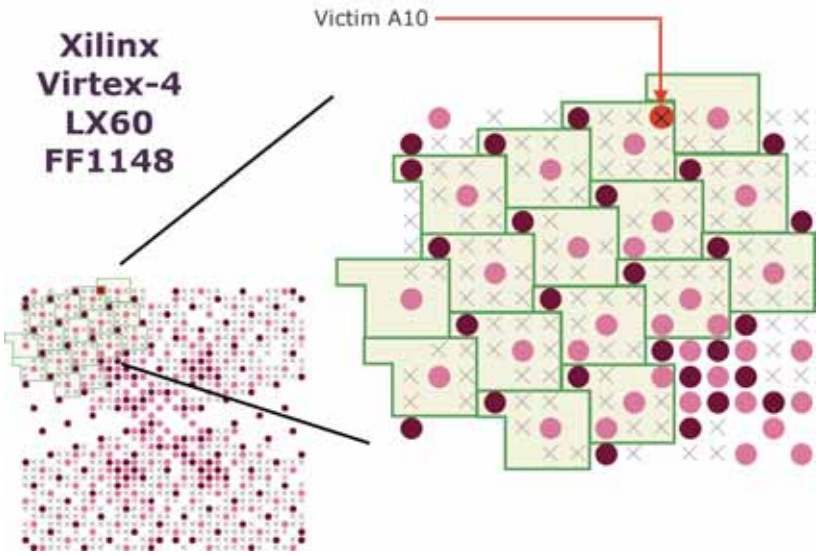


Figure 9 – The LX60 FF1148 package is tessellated with a regular array of power and ground pins, called a "sparse chevron" pattern.

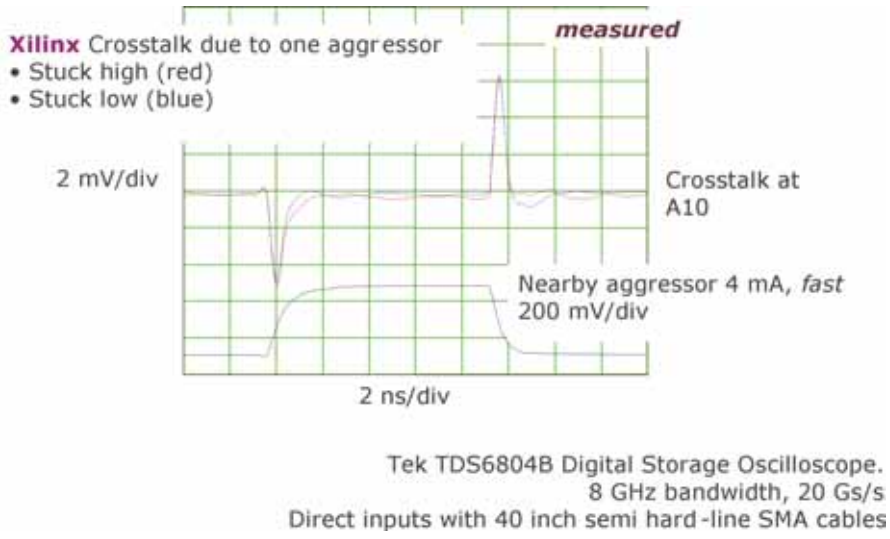


Figure 10 – The crosstalk waveforms move in a direction opposite the motion of the aggressive signal-proof that crosstalk in this experiment results from inductance, not capacitance.

What contains the distribution of returning signal current, and thus crosstalk, is the existence of lots of densely packed ground and power pins. These power/ground pins are tessellated in a regular array of ten elements called a "sparse chevron" (Figure 9). The overall ratio of signals to grounds to powers in this package is 8:1:1.

Since both power and ground pins act equally as conduits for returning signal current in this package, what we really have here is a well-distributed 4:1 ratio of signals to returns.

Figure 10 plots the first of our measured results. This data was acquired using a Tektronix TDS6804B Digital Storage Oscilloscope, 8 GHz bandwidth, 20 Gs/s, using direct inputs with 40-inch low-loss SMA cables. The bottom portion of the figure depicts a nearby aggressor, going first high and then low. The resulting crosstalk waveforms just above represent the crosstalk picked up with victim A10 stuck high (red) or stuck low (blue).

Note that the crosstalk waveforms move in a direction opposite to the motion of the aggressive signal. This movement proves conclusively that the crosstalk results from inductance, not capacitance. Capacitive crosstalk would have made the victim move the same direction as the aggressor. Certain tricks that work well to reduce capacitive crosstalk, like driving an input ball using a very low-impedance source, do nothing to improve the inductive crosstalk generated within the BGA ball field and vias underlying this FPGA package.

Crosstalk from this aggressor peaks at only six millivolts (about one percent). I made heavy use of averaging in the TDS5804 to improve the noise floor for these measurements, making possible the clarity of view represented in the diagrams.

The fact that the stuck-high and

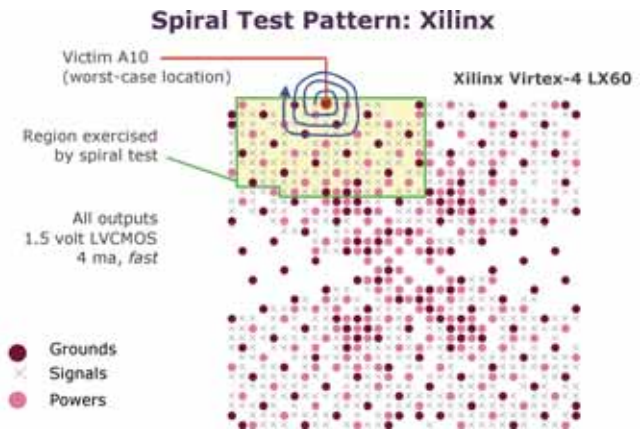


Figure 13 – The Xilinx "spiral test" exercises 100 nearest-neighbor aggressors in succession, working around and around location A10 to increasingly distant locations.

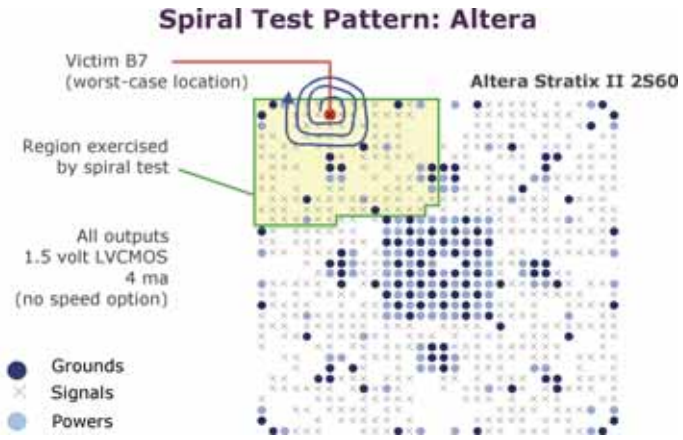


Figure 14 – The Altera spiral test exercises slightly different pins due to differences in the pin assignments, but encompasses the same number of I/Os at similar average distances.

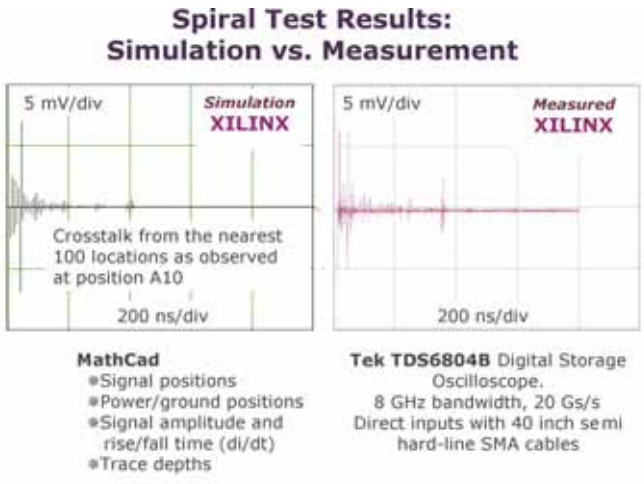


Figure 15 – If you have a theory about how something works and if you can make a simulator to predict what should happen, and the simulation matches measurement, then you probably have a good understanding of what is going on.

speed selection capability in the Stratix II.

Before I compare the packages to each other, let's compare theory with measurement (Figure 15). The combination of theory, simulation, and measurement is what Terry Morris at HP calls his "triumvirate of understanding". Terry points out that if you have a theory about how something works, and if you can make a simulator to predict what should happen, and the simulation matches your measurements, then you probably have a good understanding of what is going on.

In this case, I would say the simulation and measurement match fairly well. Both are telling us that the crosstalk pulses fall off rapidly as a function of distance from the victim A10, which remains stuck at zero during this measurement.

The Altera part displays two artifacts: more crosstalk on a pin-by-pin basis, and a pattern of crosstalk that does not fall off as quickly as crosstalk in the Xilinx package.

Even though there is not an exact match in the waveform pin-by-pin, the aggregate trend shines through. I conclude that this simulation technique realistically captures the crosstalk effect in these packages. Let's put the simulator away for now, but bring it back later to predict aggregate crosstalk at other locations around the package.

In the first comparative test result (Figure 16), I had to turn down the scale to fit both waveforms on the screen. The

stuck-low signals present nearly the same crosstalk indicates near-perfect performance of the internal bypassing within the FF1148 package.

Figure 11 shows a similar view, but with more aggressors. The aggressors turn on and off in sequential fashion, first one, and then the next, so you can see clearly the crosstalk resulting from each. Crosstalk falls off quickly as you make your way further and further from the victim.

Three Measures of Crosstalk

Having established the format of my measurements, let's look at the three comparative tests used to evaluate the Xilinx and Altera packages. The purpose of this work is to quantify the practical improvement in crosstalk made possible by the "sparse chevron" ground pin tessellation. These tests are called:

- **Spiral Test**
  100 nearest outputs, exercised individually
- **Accumulating Spiral Test**
  100 nearest outputs, aggregating in larger and larger groups
- **Hammer Test**
  500 outputs all together

I ran all three tests run on both packages. These packages each have solid internal power and ground planes plus a lot of internal bypassing. In that sense, they are very similar. They differ in their BGA pin assignments (Figure 12).

The two packages have nearly the same number of high-speed signal balls: 504 for Xilinx and 505 for Altera. The Xilinx package, being slightly larger, accommodates more power and ground balls (185 ground and 144 power for Xilinx, versus 114 and 103, respectively, for Altera). The biggest difference, though, lies not in the sheer number of ground and power balls but in their distribution. The Altera package has large areas devoid of returns. The Xilinx pattern is more evenly peppered with returns.

The Spiral Test (Figure 13) exercises the 100 nearest outputs, one at a time, each going up and then down just like in Figure 11. This figure shows the region of the Xilinx Virtex-4 LX60 FF1148 package used in this test. The victim location, A10, appears on the top row, in the middle of the aggressor region. Mark's spiral exercises the nearby aggressors in succession, working its way around and around location A10 to increasingly distant aggressors. During this test, all outputs are equipped with 1.5-volt LVCMOS 4ma fast drivers.

The Altera Spiral Test (Figure 14) exercises slightly different pins due to differences in the pin assignments, but encompasses the same number of I/Os at similar average distances. During this test, all outputs are equipped with 1.5-volt LVCMOS 4ma drivers, but there is no

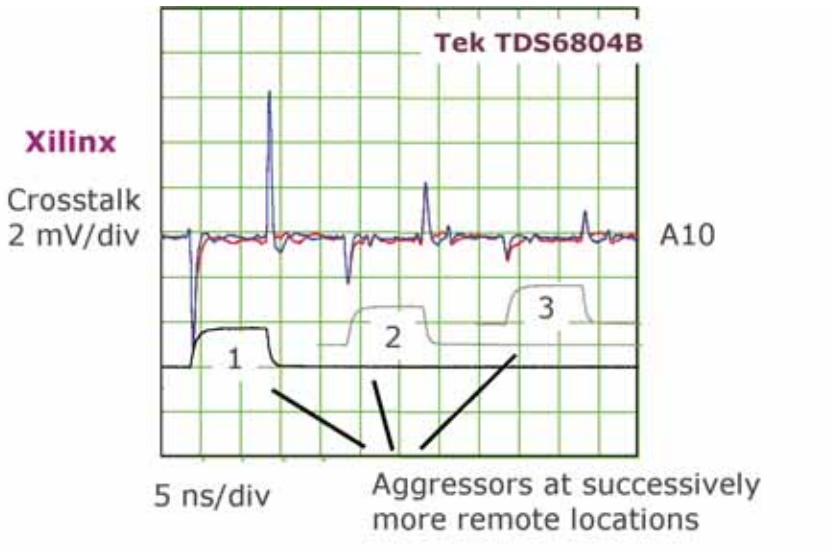


Figure 11– Crosstalk falls off quickly as you move the aggressor further and further from the victim.

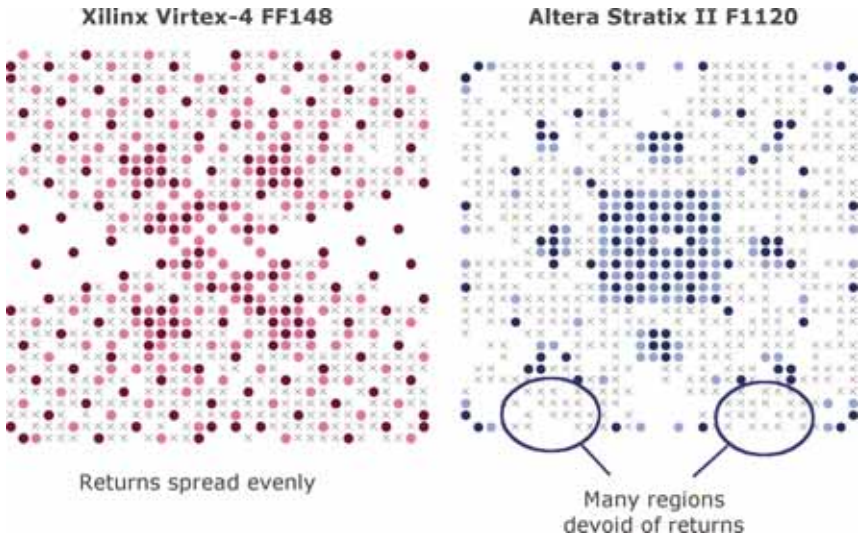


Figure 12 – Both packages have solid internal power and ground planes, plus a lot of bypassing, but very different power/ground pin assignments.

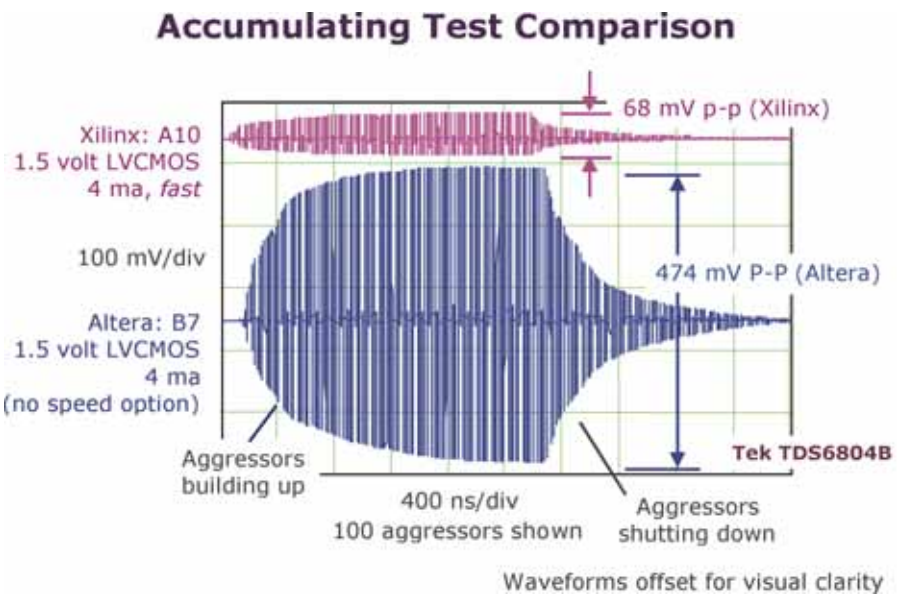


Figure 18 – This accumulating measurement shows aggregate crosstalk building up and then ramping down as you exercise different numbers of pins surrounding the victim.

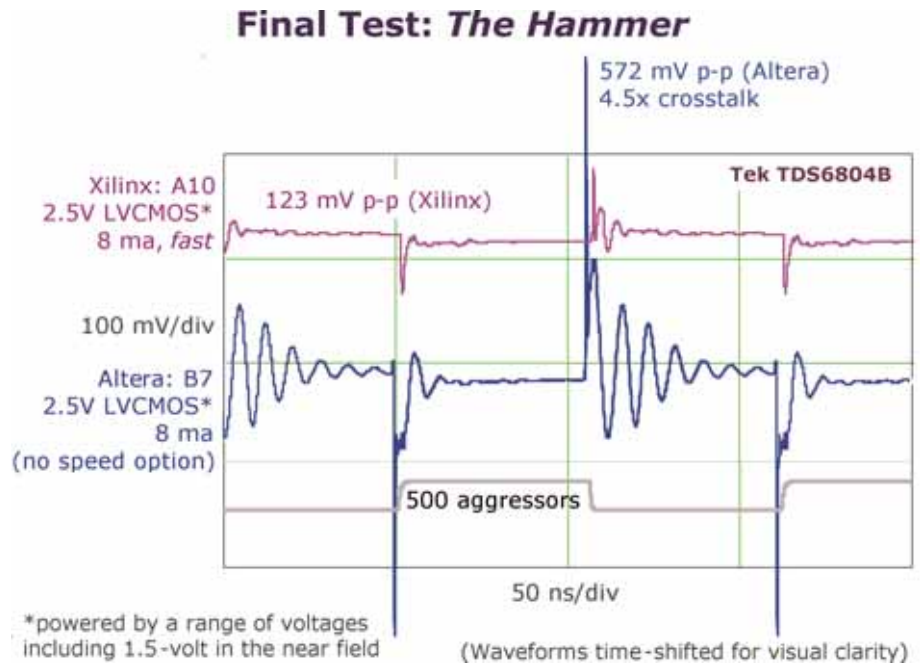


Figure 19 – The final test exercises 500 outputs on each part to maximize crosstalk. (2)

In this test the Altera part produced crosstalk waveforms 4.5 times larger than the Xilinx part.

The ratio of crosstalk voltages we measured (4.5:1) is not all due to differences in packaging. Both packaging and signal rise time contribute to this factor.

Figure 20 details the aggressive waveforms produced during the Hammer Test (measured at locations A11 and B6, respectively, on the Xilinx and Altera packages). Given the same driver settings the Altera part generates a current slope (di/dt) twice as large as the Xilinx part.

This 2:1 ratio of di/dt translates directly into more crosstalk for the Altera FPGA. The variation in power/ground pin outs accounts for the remaining portion of Altera's 4.5x increase in crosstalk.

Final Simulations

Now let's leave our measurements behind and go back to the simulator.

What I would like to do is address any concerns you may have about our selection of pin locations for testing, or other factors in the layout that may have skewed the results.

I will set up an ideal crosstalk simulation, making the following selections:

- Ignore exact pattern of trace layers, assuming an average trace depth of 0.035 in.
- Ignore differences in rise/fall time, assuming both parts produce di/dt = 2E+07 A/s.
- Ignore details of dog-bone offset, assuming a via-in-pad geometry for simplicity.

Under these conditions, Figure 21 displays the worst-case aggregate crosstalk for every pin on both devices, not just the few pins instrumented with SMA jacks. This chart pinpoints only the differences you would expect to see due to variations in the power/ground ball distribution between the two packages – not taking into account the inherent differences in di/dt due to different rise/fall times. It assumes all outputs are

The resulting crosstalk waveforms (Figure 18) clearly show the relative importance of remote balls in the overall aggregate crosstalk waveform.

top trace shows the Xilinx crosstalk at A10, the bottom shows Altera crosstalk at position B7. The Altera part displays two artifacts: more crosstalk on a pin-by-pin basis, and a pattern of crosstalk that does not fall off as quickly as crosstalk in the Xilinx package. If you add up all the individual crosstalk pulses in each waveform, you can see that the Altera crosstalk aggregates to a much higher level.

The Accumulating Crosstalk Test (Figure 17) exercises the same balls used in the one-at-a-time spiral test, but with different patterns.

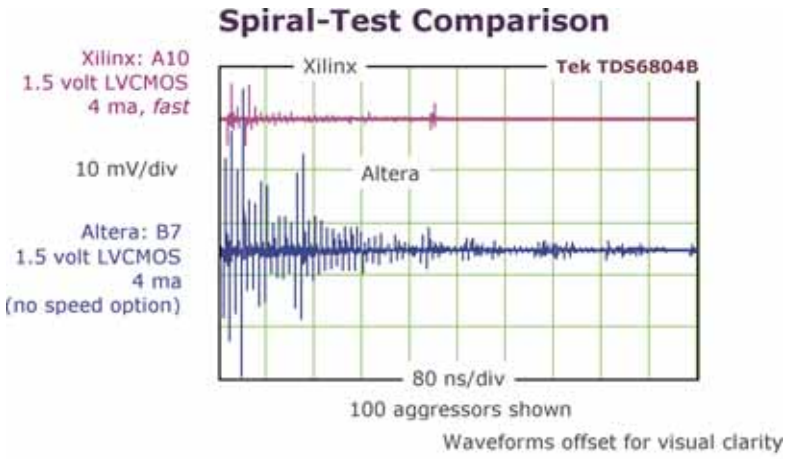


Figure 16 – In this spiral test you can clearly see the rapid fall-off of crosstalk on the Xilinx part.

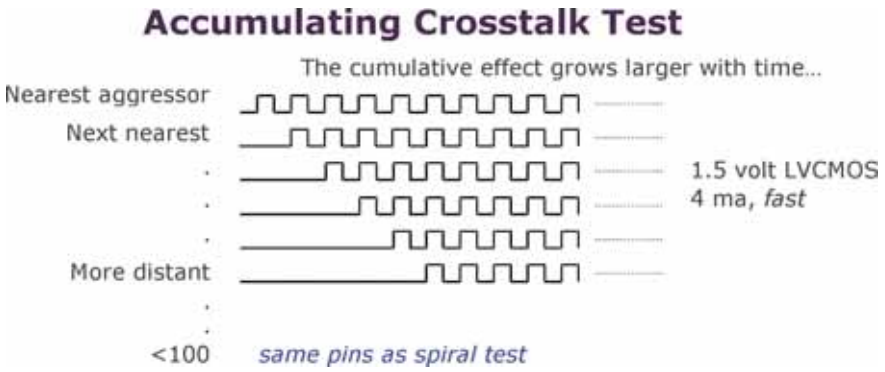


Figure 17 – The accumulating crosstalk test exercises the same balls used in the one-at-a-time spiral test, but with different patterns.

First, only the nearest ball fires off, going up and then down. Then it fires off a second time in conjunction with the next nearest ball. Then it does three at a time, then four, and so forth, until a large number of balls (100) are blasting up and down together.

The resulting crosstalk waveforms (Figure 18) clearly show the relative importance of remote balls in the overall aggregate crosstalk waveform. All aggressors in this shot are set to a 1.5-volt LVCMOS 4 ma driver (fast, for Xilinx, no speed option for Altera). As before, the Xilinx victim A10 remains stuck low. Similarly, the Altera victim B7 remains stuck low. The Xilinx component tops out at 68 mV p-p of crosstalk, the Altera component generates 474 mV p-p.

In the final test, Mark and I went all out to see how much crosstalk we could make. The Hammer Test (Figure 19) exercises 500 simultaneous outputs on each part. We wanted all the outputs running at the same voltage level for this test, but that wasn't possible with the test board architecture, so here's what we did. We configured each aggressor as a 2.5 volt LVCMOS 8 ma driver (fast, for Xilinx, but no speed option for Altera). The outputs are physically powered by a mix of voltages, including 1.5V (for at least 100 aggressors nearest the victim location), 2.5V, and 3.3V. Between the two parts, each voltage rail powered the same numbers of outputs in approximately the same positions.

Again, the victims are A10 and B7, this time stuck-at-high. The stuck-at-high setting displays BGA crosstalk along with some of the noise extant in the power system.

The stuck-low waveforms show the same high-frequency (short term) spike of crosstalk, but lack the resonant behavior visible in the power system after the main pulse. This resonance provides some clues about the efficacy of the overall power system bypassing network.

switching 1v p-p into 50 ohms with a rise/fall time of 1 ns.

Taken together with the BGA crosstalk theory from the beginning of this presentation and my actual measurements corroborating the crosstalk effects, Figure 21 paints a clear picture of the differences between the two packages under study.

I hope this presentation has been as interesting and informative for you to read as it has been fascinating (and challenging) for me to produce. If I have stimulated your interest in researching the problem further, I can suggest these related articles:

www.sigcon.com/Pubs/edn/DataCodingLowNoise.htm

www.sigcon.com/Pubs/edn/TimeforAllThings.htm

www.sigcon.com/Pubs/edn/assymnoisemargins.htm

www.sigcon.com/Pubs/news/3_9.htm
"Crosstalk and SSO Noise"

www.sigcon.com/Pubs/news/7_10.htm
"Scrambled Bus"

At my web site *www.sigcon.com* you will find a treasure trove of additional publications (282 at last count), plus a full schedule of my High-Speed Digital Design seminars, seminar course outlines, SiLab films, newsletters, Article indexes, and much more.

The Xilinx signal integrity site *www.xilinx.com/signalintegrity* holds a number of resources useful for high-speed designers, including information about my new SI tutorial for RocketIO™ serial transceivers, now available on DVD at *www.xilinx.com/store/dvd*.

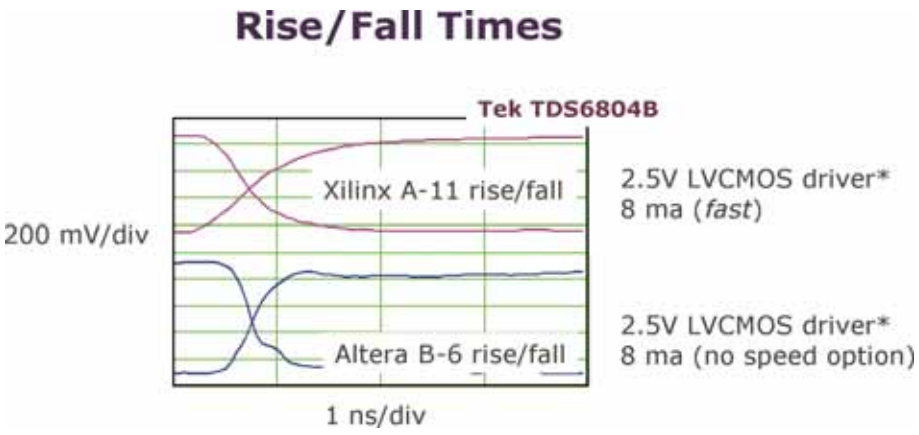


Figure 20 – The ratio of crosstalk voltages we measured (4.5:1) is not all due to differences in packaging; both packaging and signal rise time contribute to this factor.

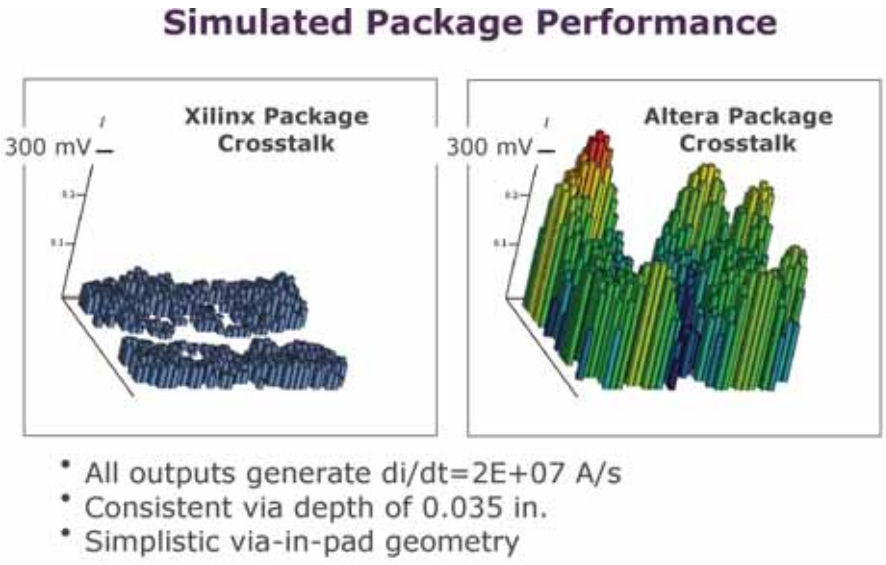


Figure 21 – This chart pinpoints only the differences you would expect to see due to variations in the power/ground ball distribution between the two packages – not taking into account the inherent differences in di/dt due to different rise/fall times.

Conclusion

The Altera component used in this test displayed crosstalk 4.5 times higher than the Xilinx component.

The Altera package suffers from two issues:

- Excessively fast signal rise/fall time
- Over-concentration of power/ground balls in core region

Together, these two effects combine to produce a 4.5:1 ratio of observed crosstalk in the Hammer Test (Figure 19).

Howard Johnson, PhD, author of High-Speed Digital Design and High-Speed Signal Propagation, *frequently conducts technical workshops for digital engineers at Oxford University and other sites worldwide.* www.sigcon.com, howie03@sigcon.com.

1 © 2005 Signal Consulting, Inc. All rights reserved. Xilinx is a registered trademark, and Virtex-4 a trademark, of Xilinx, Inc. Altera and Stratix are registered trademarks of Altera Corporation. All other company and product names may be trademarks of their respective companies.

2 Errata: horizontal axis corrected to read: 50 ns/div. The test repetition rate is 9 MHz.

XAPP623 (v2.1) February 28, 2005

Power Distribution System (PDS) Design: Using Bypass/Decoupling Capacitors

Author: Mark Alexander

Summary

This application note specifies how to build power distribution systems for Virtex™ devices. It also covers the basic principles of power distribution systems and bypass or decoupling capacitors. A step-by-step process is described where a power distribution system can be designed and verified. The final section discusses additional sources of power supply noise and provides resolutions.

Introduction

FPGA designers are faced with a unique task when it comes to designing power distribution systems (PDS). Most other large, dense ICs (such as large microprocessors) come with very specific bypass capacitor requirements. Since these devices are only designed to implement specific tasks in their hard silicon, their power supply demands are fixed and only fluctuate within a certain range. FPGAs do not share this property. Since FPGAs can implement an almost infinite number of applications at undetermined frequencies and in multiple clock domains, it can be very complicated to predict what their transient current demands will be.

Since exact transient current behavior cannot be known for a new FPGA design, the only choice when designing the first version of an FPGA PDS is to go with a conservative worst-case design.

Transient current demands in digital devices are the cause of ground bounce, the bane of high-speed digital designs. In low-noise or high-power situations, the power supply decoupling network must be tailored very closely to these transient current needs, otherwise ground bounce and power supply noise will exceed the limits of the device. The transient currents in an FPGA are different from design to design. This application note provides a comprehensive method for designing a bypassing network to suit the individual needs of a specific FPGA design.

The first step in this process is to examine the utilization of the FPGA to get a rough idea of its transient current requirements. Next, a conservative decoupling network is designed to fit these requirements. The third step is to refine the network through simulation and modification of capacitor numbers and values. In the fourth step, the full design is built and in the fifth step it is measured. Measurements are made consisting of oscilloscope and possibly spectrum analyzer readings of power supply noise. Depending on the measured results, further iterations through the part selection and simulation steps could be necessary to optimize the PDS for the specific application. A sixth optional step is also given for cases where a perfectly optimized PDS is needed.

Basic Decoupling Network Principles

Before starting into the PDS design flow, it is important to understand the basic electrical principles involved. This section discusses the purpose of the PDS and the properties of its components. It also describes important aspects of discrete capacitor placement and mounting as well as PCB geometry and stackup recommendations.

The purpose of a PDS is to provide power to the devices in a system. Each device in a system not only has its own power requirements for its operation, but also its own requirement for the cleanliness of that power. Most digital devices, including all Xilinx FPGAs, have a requirement that on all supplies, V_{CC} must not fluctuate more than 5% above or 5% below the nominal V_{CC} value. In this document V_{CC} is used generically to refer to all FPGA power supplies: V_{CCINT},

© 2003-2005 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.

V_{CCO}, V_{CCAUX}, and V_{REF}. Multi-gigabit transceiver (MGT) analog supplies ($AV_{CCAUXTX}$, $AV_{CCAUXRX}$, VTTX, VTRX) are not covered here. For specific instructions on these supplies, see the *RocketIO™ Transceiver User Guide* (**Reference #1**).

This requirement specifies a maximum amount of noise present on the power supply, often referred to as "ripple voltage." If the device requirements state that V_{CC} must be within ±5% of the nominal voltage, that means peak to peak voltage ripple must be no more than 10% of the nominal V_{CC}. This assumes that nominal V_{CC} is exactly the nominal value given in the datasheet. If this is not the case, then V_{RIPPLE} must be adjusted to a value correspondingly less than 10%.

The power consumed by a digital device varies over time, and this variance occurs on all frequency scales. Low frequency variance of power consumption is usually the result of devices or large portions of devices being enabled or disabled. This can occur on time scales from milliseconds to days. High frequency variance of power consumption is the result of individual switching events inside a device, and this happens on the scale of the clock frequency and the first few harmonics of the clock frequency.

Since the voltage level of V_{CC} for a device is fixed, changing power demands are manifested as changing current demand. The PDS must accommodate these variances of current draw with as little change in power supply voltage as possible.

When the current draw in a device changes, the power distribution system cannot respond to that change instantaneously. For the short time before the PDS responds, the voltage at the device changes. This is where power supply noise appears. There are two main causes for this lag in the PDS corresponding to the two major components of the PDS.

The first major component of the PDS is the voltage regulator. It observes its output voltage and adjusts the amount of current being supplied to keep the voltage constant. Most common voltage regulators make this adjustment on the order of milliseconds to microseconds. They are effective at maintaining output voltage for events at all frequencies from DC to a few hundred kilohertz (depending on the regulator). For all transient events that occur at frequencies above this range, there is a time lag before the voltage regulator can respond to the new level of demand. For example, if current demand in the device increases in a matter of nanoseconds, the voltage at the device sags by some amount until the voltage regulator can adjust to the new, higher level of current it must provide. This lag might take from microseconds to milliseconds, during which time the voltage sags.

The second major component of the PDS is the bypass or decoupling capacitors. In this application note, the words "bypass" and "decoupling" are used interchangeably. Their function is to act as local energy storage for the device. They cannot provide DC power, as only a small amount of energy is stored in them (the voltage regulator is present to provide DC power). The function of this local energy storage is to respond very quickly to changing current demands. The capacitors are effective at maintaining power supply voltage at frequencies from hundreds of kilohertz to hundreds of megahertz in the milliseconds to nanoseconds range. Decoupling capacitors are of no use for all events occurring above or below this range. For example, if current demand in the device increases in a few picoseconds, the voltage at the device sags by some amount until the capacitors can supply extra charge to the device. If current demand in the device changes and maintains this new level for a number of milliseconds, the voltage regulator circuit, operating in parallel with the bypass capacitors, effectively takes over for them, changing its output to supply this new current.

Figure 1 shows the major components of the PDS: the power supply, the decoupling capacitors, and the active device being powered (in this case, an FPGA).

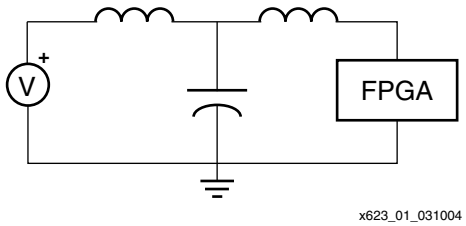


Figure 1: **Simplified PDS Circuit**

Figure 2 shows a further simplified PDS circuit, showing all reactive components decomposed to a frequency-dependent resistor.

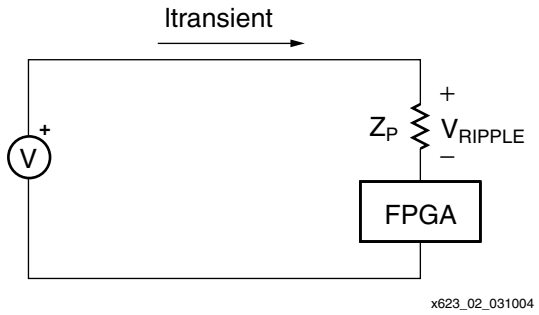


Figure 2: **Further Simplified PDS Circuit**

What is the Role of Inductance?

There is a property of the capacitors and of the PCB current paths that retards changes in current flow. This is the reason why capacitors cannot respond instantaneously to transient currents, nor to changes that occur at frequencies higher than their effective range. This property is called inductance.

Inductance can be thought of as momentum of charge. Where charge is moving at some rate through a conductor, this implies some amount of current. If the level of current is to change, the charge must move at a different rate. Because there is momentum (stored magnetic field energy) associated with this charge, it takes some amount of time for the charge to slow down or speed up. The greater the inductance, the greater the resistance to change, and the longer it takes for the current level to change.

The purpose of the PDS is to accommodate whatever current demands the device(s) could have, and respond to changes in that demand as quickly as possible. When these demands are not met, the voltage across the device's power supply changes. This is observed as noise. Since inductance retards the abilities of bypass capacitors to quickly respond to changing current demands, it should be minimized.

Figure 1 shows inductances between the FPGA device and capacitors, and between the capacitors and the voltage regulator. These inductances arise as parasitics of the capacitors themselves and of all current paths in the PCB. It is important that each of these be minimized.

Capacitor Parasitic Inductance

Of a capacitor's various properties, the capacitance value is often considered the most important. However in the domain of PCB PDS design, the property of parasitic inductance (ESL or Equivalent Series Inductance) is of the same or greater importance.

The one factor that influences parasitic inductance more than any other is the dimensions of the package. Very simply, physically small capacitors tend to have lower parasitic inductance than physically large capacitors. Just as a short wire has less inductance than a long wire, a short capacitor has less inductance than a long capacitor. Likewise, as a fat or wide wire has less inductance than a narrow wire, so too does a fat capacitor have less inductance than a narrow capacitor.

For these reasons, when choosing decoupling capacitors, the smallest package should be chosen for a given value. Similarly, for a given package size (essentially a fixed inductance value), the highest capacitance value available in that package should be chosen.

Surface-mount chip capacitors are the smallest capacitors available, making them a good choice for discrete bypass capacitors. For values from 2.2 µF down to very small values such as 0.001 µF, X7R or X5R type capacitors are usually used. These have low parasitic inductance, and an acceptable temperature characteristic. For larger values, such as 1000 µF, tantalum capacitors are used. These have low parasitic inductance and a relatively high equivalent series resistance (ESR), giving them a low-quality factor and consequently a very wide range of effective frequencies. They also provide a comparatively high capacitance value in a small package size, thus reducing board real-estate costs. In cases where tantalum

capacitors are not available, low-inductance electrolytic capacitors can be used. Other new technologies with similar characteristics are also available.

A real capacitor has characteristics not only of capacitance but also inductance and resistance. Figure 3 shows the parasitic model of a real capacitor. A real capacitor should be treated as an RLC circuit.

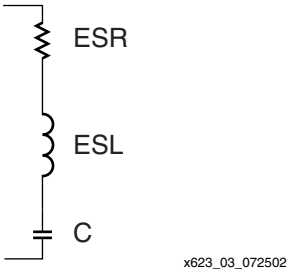


Figure 3: **Parasitics of a Real, Non-Ideal Capacitor**

Figure 4 shows the impedance characteristic of a real capacitor. Overlaid on this plot are the curves corresponding to the capacitor's capacitance and parasitic inductance (ESL). These two curves combine to form the total impedance characteristic of the RLC circuit formed by the parasitics of the capacitor.

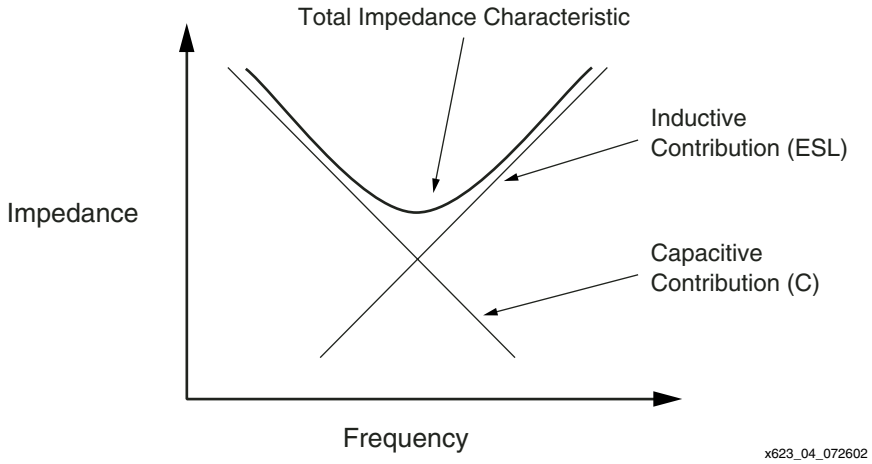


Figure 4: **Contribution of Parasitics to Total Impedance Characteristics**

As capacitive value is increased, the capacitive curve moves down and to the left. As parasitic inductance is decreased, the inductive curve moves down and to the right. Since parasitic inductance for capacitors in a given package is essentially fixed, the inductance curve remains fixed. As different capacitor values are selected in that same package, the capacitive curve moves up and down relative to the fixed inductance curve. The only way to decrease the total impedance of a capacitor for a given package is to increase the value of the capacitor. The only way to move the parasitic inductance curve down (and consequently lower the total impedance characteristic), is to connect additional capacitors in parallel.

Inductance from PCB Current Paths

The parasitic inductance of current paths in the PCB have two distinct sources: the capacitor mounting, and the power and ground planes of the PCB.

Mounting Inductance

In this context, the mounting refers to the capacitor's solder land on the PCB, the trace (if any) between the land and via, and the via itself.

The vias, traces, and pads of a capacitor mounting contribute anywhere from 300 pH to 4 nH of inductance depending on the specific geometry. Since the inductance of a current path is proportional to the area of the loop the current traverses, it is important to minimize the size of

this loop. The loop consists of the path through one power plane, up through one via, through the connecting trace to the land, through the capacitor, through the other land and connecting trace, down through the other via, and into the other plane, as shown in Figure 5.

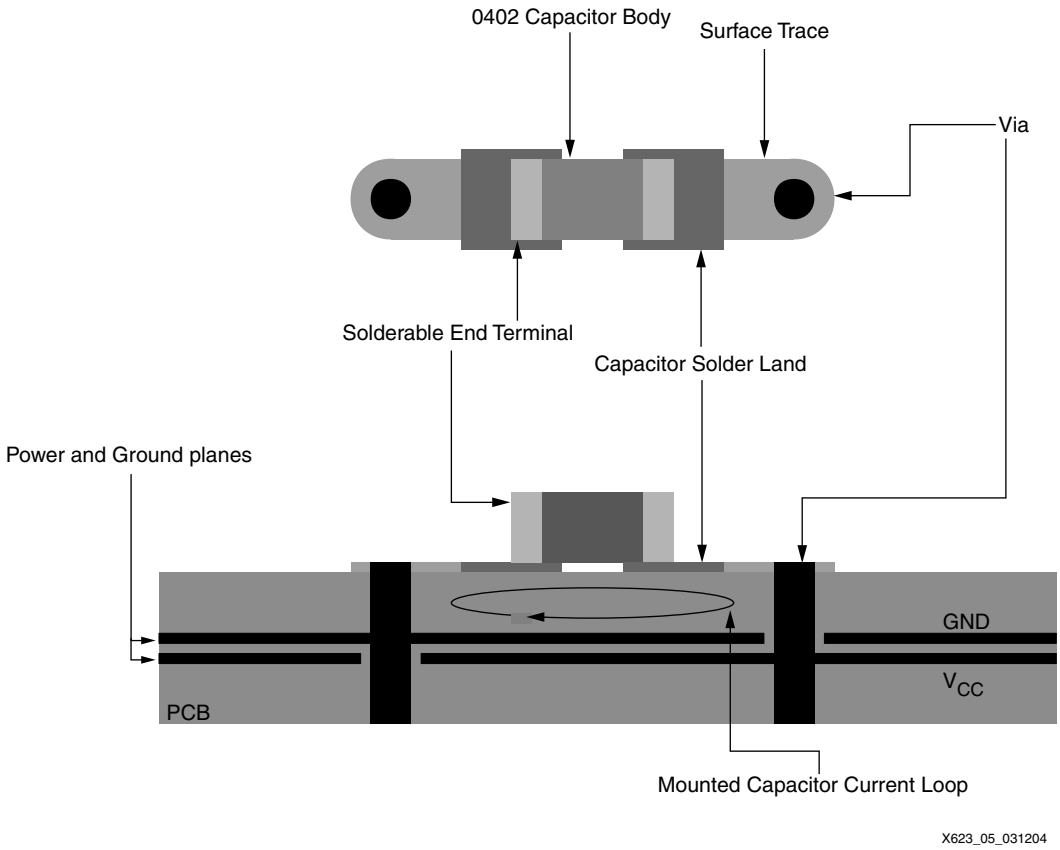


Figure 5: **Cutaway View of PCB with Capacitor Mounting**

By shortening the connecting traces, the area of this loop is minimized and the inductance is reduced. Similarly, by reducing the via length through which the current flows, loop area is minimized and inductance is reduced.

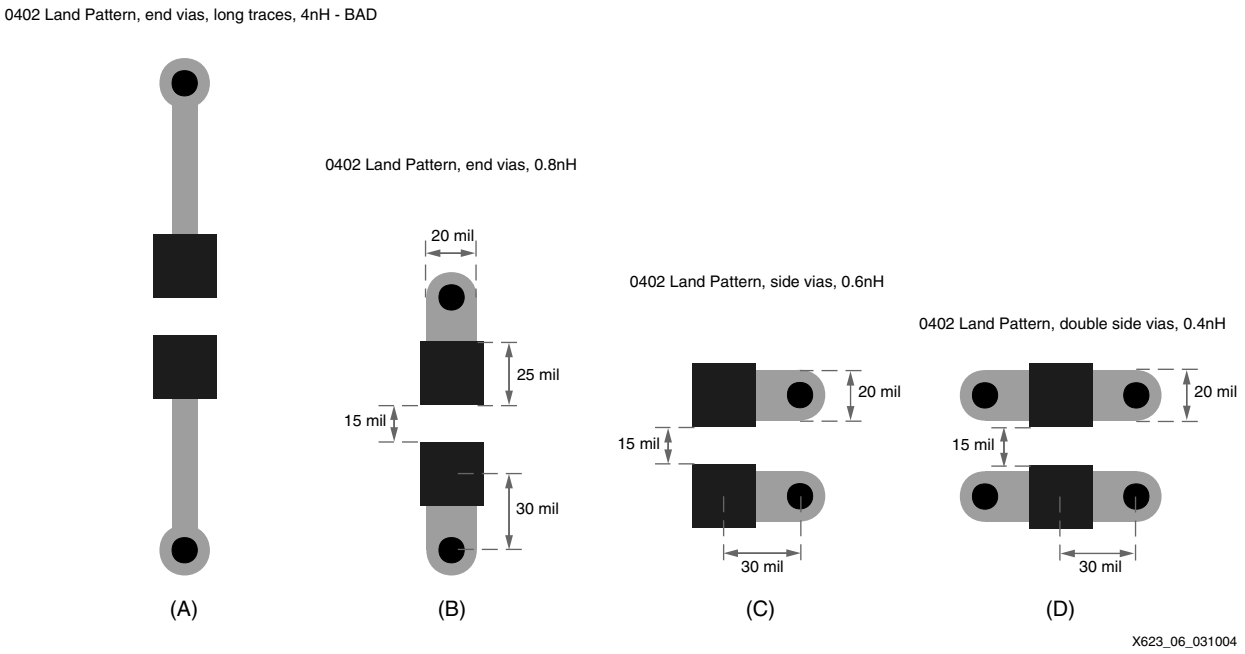


Figure 6: **Example Capacitor Land and Mounting Geometries**

The existence and/or length of a connecting trace has a big impact on parasitic inductance of the mounting. Wherever possible, there should be no connecting trace (Figure 6a) - the via should butt up against the land itself (Figure 6b). Additionally, the connecting trace should be made as wide as possible. Further improvements can be made to the mounting by placing vias to the side of capacitor lands (Figure 6c), or doubling the number of vias (Figure 6d). Currently, very few PCB manufacturing processes allow via-in-pad geometries, but this is another good option. The technique of using multiple vias per land is important when using ultra-low inductance capacitors, such as reverse aspect ratio capacitors (AVX's LICC).

Many times in an effort to squeeze more parts into a small area, PCB layout engineers opt to share vias among multiple capacitors. **This technique should not be used under any circumstances**. The capacitor mounting (lands, traces and vias) typically contributes about the same amount or more inductance than the capacitor's own parasitic inductance. If a second capacitor is connected into the vias of an existing capacitor, it only improves the PDS by a very small amount. It is better to reduce the total number of capacitors and maintain a one-to-one ratio of lands to vias.

Plane Inductance

The power and ground planes of a PCB have some amount of inductance associated with them. The geometry of these planes determines their inductance.

Since power and ground planes are by definition a planar structure, current does not just flow through them in one direction. It tends to spread out as it travels from one point to another, in accordance with a property similar to skin effect. For this reason, inductance of planes can be described as "spreading inductance," and is specified in units of henries per square. The square is dimensionless, as it is the shape of a section of plane, not the size, that determines its inductance.

Spreading inductance acts like any other inductance — to resist changes to the amount of current in a conductor. In this case, the conductor is the power plane or planes. This quantity should be reduced as much as possible, since it retards the ability of capacitors to respond to transient currents in the device. Since the X-Y shape of the plane is typically something the designer has little control over, the only controllable factor is the spreading inductance value. This is primarily determined by the thickness of the dielectric separating a power plane from its associated ground plane.

In high-frequency power distribution systems of the type discussed here, power and ground planes work in pairs. Their inductances do not exist independently of each other. The spacing (and the dielectric constant of the material) between power and ground planes determines the spreading inductance of the pair. The closer the spacing (the thinner the dielectric), the lower the spreading inductance. Table 1 gives approximate values of spreading inductance for different thicknesses of FR4 dielectric (**Reference #2**).

Table 1: **Capacitance and Spreading Inductance Values for Various Thicknesses of FR4 Power-Ground Plane Sandwiches**

Dielectric Thickness (mil, microns)	Inductance (pH/square)	Capacitance (pF/in^2, pF/cm^2)
4, 102	130	225, 35
2, 51	65	450, 70
1, 25	32	900, 140

Since closer spacing results in decreased spreading inductance, it is best, wherever possible, to place V_{CC} planes directly adjacent to GND planes in the stackup. Facing V_{CC} and GND planes are sometimes referred to as "sandwiches." While the use of V_{CC} – GND sandwiches was not necessary in the past for previous technologies, the speeds involved and the sheer amount of power required for fast, dense devices demands it.

Besides offering a low-inductance current path, power-ground sandwiches also offer some high-frequency decoupling capacitance. As plane area increases and as the separation between power and ground planes decreases, the value of this capacitance increases. At the same time, since the parasitic inductance of this capacitance is decreasing, its effective frequency band center frequency increases. Capacitance per square inch is also given in Table 1.

This capacitance alone is usually not enough to give power-ground sandwiches a compelling advantage. However, when viewed as a bonus on top of low spreading inductance, it is an advantage most designers gladly take.

PCB Stackup and Layer Order

The placement of V_{CC} and Ground planes in the PCB stackup (determined by layer order) has a significant impact on the parasitic inductances of power current paths. For this reason, PCB designers need to consider layer order in the early stages of the design cycle, putting high-priority supplies in the top half of the stackup and low-priority supplies in the bottom half of the stackup.

Power supplies with high transient current should have their associated V_{CC} planes close to the top surface (FPGA side) of the PCB stackup to decrease the distance in the vertical direction that currents travel through V_{CC} and GND vias before reaching the associated V_{CC} and GND planes. As mentioned in the previous section, every V_{CC} plane should have a GND plane adjacent to it in the stackup to reduce spreading inductance. Since high-frequency currents couple tightly due to skin effect, the GND plane adjacent to a given V_{CC} plane tends to carry the majority of the current complementary to that in the V_{CC} plane. For this reason, adjacent V_{CC} and GND planes are considered as a pair.

Not all V_{CC} and GND plane pairs can reside in the top half of the PCB stackup, because manufacturing constraints typically require the PCB stackup to be symmetrical about the center with respect to dielectric thicknesses and etched copper areas. The PCB designer must determine which V_{CC} and GND plane pairs have high priority or carry high-frequency energy, and which pairs have low priority or carry lower frequency energy.

Capacitor Effective Frequency

Every capacitor has a narrow frequency band where it is most effective as a decoupling capacitor. Outside this band, it does have some contribution to the PDS but in general it is much smaller. The frequency bands of some capacitors are wider than others. The ESR of the capacitor determines the quality factor (Q) of the capacitor, which determines the width of the effective frequency band. Tantalum capacitors generally have a very wide effective band, while

X7R and X5R chip capacitors, with their lower ESR, generally have a very narrow effective band.

The effective frequency band corresponds to the capacitor's resonant frequency. While an ideal capacitor only has a capacitive characteristic, real non-ideal capacitors also have a parasitic inductance ESL and a parasitic resistance ESR. These parasitics act in series to form an RLC circuit (Figure 3). The resonant frequency associated with that RLC circuit is the resonant frequency of the capacitor.

To determine the resonant frequency of an RLC circuit, the following equation is used:

$$F = \frac{1}{2\pi\sqrt{LC}} \qquad \text{Equation 1}$$

Alternatively, a frequency sweep SPICE simulation of the circuit could be performed, and the frequency where the minimum impedance value occurs would be the resonant frequency.

It is important to distinguish between the capacitor's self-resonant frequency and the effective resonant frequency of the mounted capacitor when it is part of the system. This is simply the difference between taking into account only the capacitor's parasitic inductance, and taking into account its parasitic inductance as well as that of the vias, planes, and connecting traces lying between it and the FPGA. The self-resonant frequency of the capacitor F_{RSELF} (the value reported in a capacitor datasheet), is considerably higher than its effective mounted resonant frequency in the system, F_{RIS}. Since the mounted capacitor's performance is what is important, it is the mounted resonant frequency that is used when evaluating a capacitor as part of a larger PDS.

The main contributors to mounted parasitic inductance are the capacitor's own parasitic inductance, the inductance of PCB lands and connecting traces, the inductance of vias, and power plane inductance. Vias traverse a full board stackup on their way to the device when capacitors are mounted on the underside of the board. These vias contribute something in the range of 300 pH to 1,500 pH on a board with a finished thickness of 60 mils; vias in thicker boards have higher inductance. Because there are two of these paths in series with each capacitor, twice this value should be added to the capacitor's parasitic inductance. This quantity, the parasitic inductance of the capacitor mounting, is designated L_{MOUNT}. To determine the total parasitic inductance of the capacitor in-system, L_{IS}, the capacitor's parasitic inductance L_{SELF} is added to the parasitic inductance of the mounting, L_{MOUNT}:

$L_{IS} = L_{SELF} + L_{MOUNT}$

Example

X7R Ceramic Chip capacitor (AVX capacitor data used here)

$C = 0.01\ \mu F$

$L_{SELF} = 0.9$ nH

$F_{RSELF} = 53$ MHz

$L_{MOUNT} = 0.8$ nH

To determine the effective in-system parasitic inductance (L_{IS}), add the via parasitics:

$L_{IS} = L_{SELF} + L_{MOUNT} = 0.9\text{ nH} + 0.8\text{ nH} = 1.7\text{ nH}$

$L_{IS} = 1.7$ nH

Plugging in the values from the example:

$$F_{RIS} = \frac{1}{2\pi\sqrt{L_{IS}C}}$$

$$F_{RIS} = \frac{1}{2\pi\sqrt{(1.7\times10^{-9}H)\cdot(1\times10^{-8}F)}} = 3.8\times10^{7}Hz$$

F_{RIS}: Mounted Capacitor Resonant Frequency: 38 MHz

Since a decoupling capacitor is only effective at a narrow band of frequencies around its resonant frequency, it is important that the resonant frequency be taken into account when choosing a collection of capacitors to build up a decoupling network.

Capacitor Anti-Resonance

One common problem associated with capacitors in an FPGA PDS is anti-resonant spikes in the PDS aggregate impedance. These spikes can be caused by bad combinations of energy storage devices in the PDS (such as discrete capacitors, parasitic inductances, power and ground planes). If the inter-plane capacitance of the power and ground planes has an especially low Z with a high-quality factor, the crossover point between the high-frequency discrete capacitors and this plane capacitance might exhibit a high-impedance anti-resonance peak. If the FPGA has high transient current demand at this frequency (acting as a stimulus), a large noise voltage results. The PDS can be improved only by bringing down the impedance of the anti-resonant spike. To mitigate this problem, either the characteristics of the high-frequency discrete capacitors or the characteristics of the V_{CC} and Ground planes must be changed.

Capacitor Placement

Capacitors need to be close to the device to perform the decoupling function. There are two basic reasons for this requirement.

First, increased spacing between the device and decoupling capacitor increases the distance travelled by the current in the power and ground planes, and hence, the inductance of the current path between the device and the capacitor. Since the inductance of this path (the loop followed by current as it goes from the V_{CC} side of the capacitor to the V_{CC} pin[s] of the FPGA, and from the GND pin[s] of the FPGA to the GND side of the capacitor[s]), is proportional to the loop area, decreasing its inductance is a matter of decreasing the loop area. Shortening the distance between the device and the decoupling capacitor(s) reduces the inductance resulting in a less impeded transient current flow. Because of the dimensions of PCBs, this reason tends to be less important with regard to placement than the second reason.

The second reason deals with the phase relationship between the FPGA noise source and the mounted capacitor. Their phase relationship determines the capacitor's effectiveness. For a capacitor to be effective in providing transient current at a certain frequency (for instance, the optimum frequency for that capacitor), it must be within a fraction of the wavelength associated with that frequency. The placement of the capacitor determines the length of the transmission line interconnect (in this case, the power and ground plane pair) between the capacitor and FPGA. The propagation delay of this interconnect is the relevant factor.

Noise from the FPGA falls into certain frequency bands, and different sizes of decoupling capacitors take care of different frequency bands. For this reason, capacitor placement is determined based on the effective frequency of each capacitor.

When the FPGA initiates a change in its current demand, it causes a small local disturbance in the voltage of the PDS (a point in the power and ground planes). For a decoupling capacitor to counteract this, the capacitor has to first see a voltage difference. There is a finite time delay between the start of the disturbance at the FPGA power pins and the start of the capacitor's view of the disturbance. This time delay is equal to the distance from FPGA power pins to capacitor, divided by the propagation speed of current through FR4 dielectric (the substrate of the PCB where the power planes are embedded). There is another delay of the same duration for the compensation current from the capacitor to reach the FPGA.

Therefore, for any transient current demand in the FPGA, there is a round-trip delay to the capacitor before any relief is seen at the FPGA. For placement distances greater than one quarter of a wavelength of some frequency, the energy transferred to the FPGA is negligible.

For decreasing distances less than a quarter wavelength, the energy transferred to the FPGA increases to 100% at zero distance. Efficient energy transfer from the capacitor to the FPGA requires placement of the capacitor at a fraction of a quarter wavelength of the FPGA power

pins. This fraction should be small because the capacitor is also effective at frequencies slightly above its resonant frequency, where the corresponding wavelength is shorter.

In practical applications, one tenth of a quarter wavelength is a good target. This leads to placing a capacitor within one fortieth of a wavelength of the power pins it is decoupling. The wavelength corresponds to F_{RIS}, the capacitor's mounted resonant frequency.

Example

0.001 µF X7R Ceramic Chip capacitor, 0402 package

L_{IS} = 1.6 nH

$$F_{RIS} = \frac{1}{2\pi\sqrt{LC}} = \frac{1}{2\pi\sqrt{1.6\times10^{-9} \times 0.001\times10^{-6}}} = 125.8\,MHz$$

Equation 2 calculates T_{RIS}, the mounted period of resonance, from F_{RIS}

$$T_{RIS} = \frac{1}{F_{RIS}} = \frac{1}{125.8\times10^{6}} = 7.95\ ns \quad \text{Equation 2}$$

Equation 3 computes the wavelength based on T_{RIS} and the propagation velocity in FR4 dielectric.

$$\lambda = Wavelength = \frac{T_{RIS}}{V_{PROP}} \quad \text{Equation 3}$$

where $V_{PROP} = 166\times10^{-12}\frac{s}{inch}$

$$\lambda = \frac{T_{RIS}}{V_{PROP}} = \frac{7.95\times10^{-9}}{166\times10^{-12}} = 47.9\ inches$$

$$R_{PLACE} = \frac{\lambda}{40} \quad \text{Equation 4}$$

$$R_{PLACE} = \frac{\lambda}{40} = \frac{47.9\ inches}{40} = 1.20\ inches$$

In this example, the effective frequency, equal to the resonant frequency, can be determined by Equation 1. This effective frequency is determined to be 125.8 MHz. The reciprocal of this is taken to give the resonant period, 7.95 ns using Equation 2. Using the propagation speed of current in FR4 (approximately 166 ps per inch), the wavelength associated with this capacitor is computed to be approximately 48 inches using Equation 3. As computed in Equation 4, one fortieth of this is 1.2 inches. Therefore the target placement radius (R_{PLACE}) for capacitors of this size is within 1.2 inches (3.0 cm) of the power and ground pins they are decoupling.

All other capacitor sizes follow in the same manner. A radius of 1.2 inches is not terribly difficult to achieve in current PCB technology. It does not require placing capacitors directly underneath the device on the opposite side of the PCB. It is acceptable for capacitors to be mounted around the periphery of the device, provided the target radius is maintained. The 0.001 µF capacitors are among the smallest in the decoupling network, so placement radii less than an inch are unnecessary. For larger capacitors, the target placement radius expands quickly as the resonant frequency goes down. A 4.7 µF capacitor, for example, can be placed anywhere on the board, as its target radius of 98 inches is much bigger than most PCBs (corresponding to the resonant frequency of 1.56 MHz).

Example Capacitor Layout

Figure 7 is an example of the bottom-side PCB artwork showing the capacitor layout. Black fill and hatch represents plated copper, red represents vias, blue represents silkscreen labels, and

purple represents package outlines. The FPGA footprint can be seen as the regular array of red via dots in the upper portion of the figure at the center. The absence of vias in a cross pattern at the center of the device indicates that solder lands on the top surface had their associated vias escape out toward the corners.

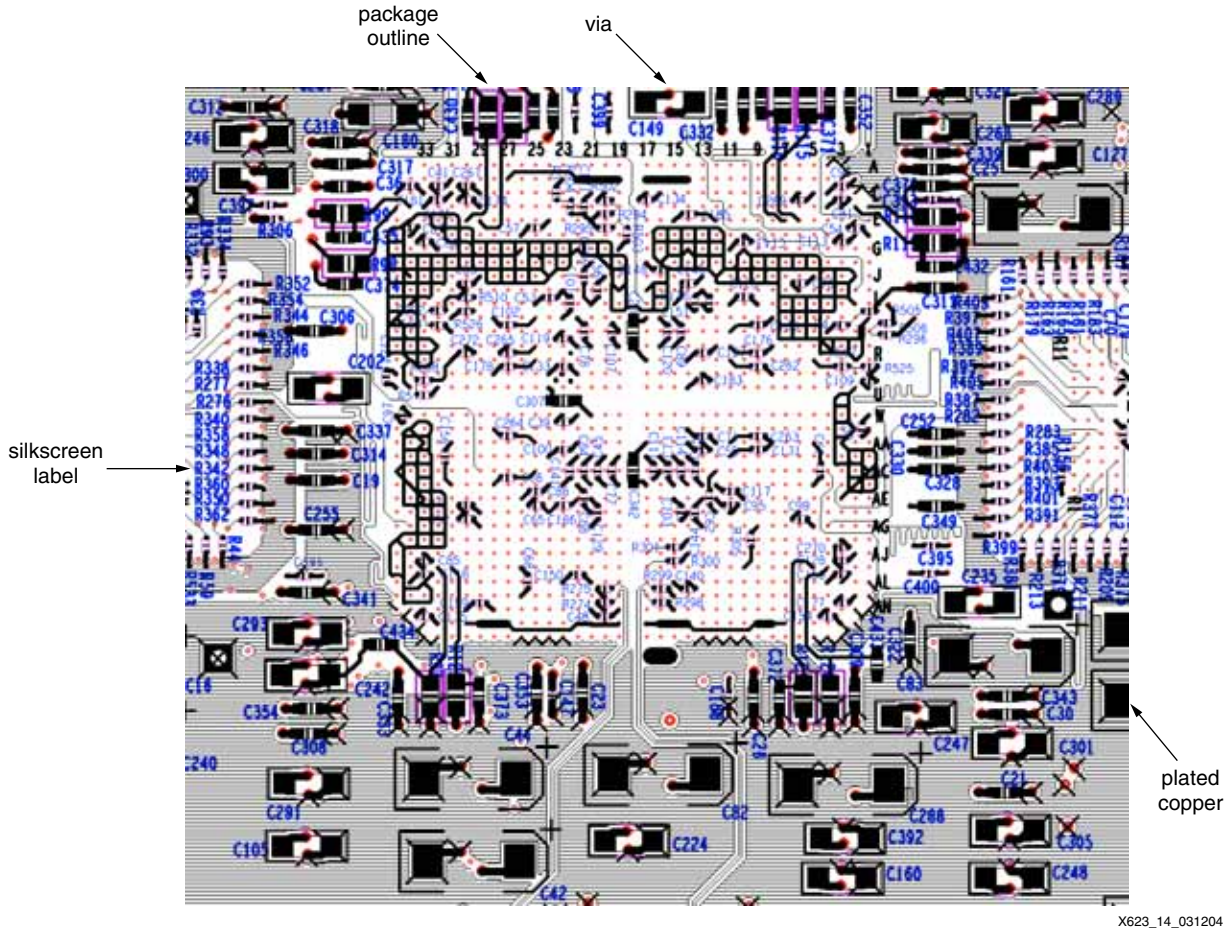


Figure 7: **Example PCB Layout showing Capacitor Placement on the Bottom Surface**

In this example, many of the high-frequency 0402 decoupling capacitors are placed within the footprint of the FPGA on the opposite side of the board (C150, C117). There are also a handful of 0603 decoupling capacitors and termination resistors (C307, R274). Larger capacitors are placed outside the footprint of the FPGA, moving farther away from the FPGA with increasing size (C247, C288).

Traces connecting capacitor lands to vias are kept as short as possible. Also for large-package capacitors with large separation between solder lands (C42, C224), vias are inserted in between the solder lands to reduce parasitic inductance of the mounting.

It is not necessary to place high-frequency capacitors within the footprint of the FPGA. It is perfectly acceptable to place all capacitors around the periphery of the device, provided all V_{CC} planes have a Ground plane adjacent to them, separated by a dielectric less than 4 mils in thickness. Also, in cases where V_{CC} and Ground plane pairs are in the top half of the stackup (closer to the device), it is advantageous to place capacitors on the top surface of the board, around the periphery of the device.

In cases where large numbers of external termination resistors are used, placement of the termination resistors takes priority over the decoupling capacitors. Moving away from the device in concentric rings, termination resistors should be closest to the device, followed by the smallest-value decoupling capacitors, then followed by larger-value decoupling capacitors.

PDS Design and Verification

Having discussed the basic operating principles of power distribution systems, this section introduces a step-by-step process for designing and verifying a PDS.

Step 1: Determining Critical Parameters of the FPGA

In designing the first iteration of the decoupling capacitor network, the basic objective is to have one capacitor per V_{CC} pin used on the device. Therefore, the *effective* number of V_{CC} pins for each supply must be determined.

Very few designs use 100% of the various resources in the FPGA. The FPGA package and the PDS inside it are very carefully sized to meet the needs of a fully utilized die without being overly conservative. The number of V_{CC} and GND pins on a package for a given device is determined based on the needs of a 100% utilized FPGA. The determining factor is not DC power handling abilities — it is transient current impedance. Decoupling capacitor requirements track very closely since they are based on the same factor. For this reason, the number of V_{CC} pins on each supply is used as an indicator of the number of capacitors needed on that supply. All supplies must be considered: V_{CCINT}, V_{CCAUX}, V_{CCO}, and V_{REF}.

It is only necessary to provide one capacitor per V_{CC} pin if all pins are used. There is no need to decouple V_{REF} pins if they are not used as V_{REF}. Conversely, V_{CCAUX} and V_{CCINT} pins must always be fully decoupled, i.e., they must always have one capacitor per pin. V_{CCO} can be pro-rated according to I/O utilization.

Pro-rating V_{CCO} Pins

The number of V_{CCO} pins used by a device can be determined based on the Simultaneously Switching Output (SSO) restrictions given in the device documentation (data sheet and user guide). A budget is calculated on a per-bank basis using these restrictions. The utilization of I/O resources in a bank determines the percentage of the budget used. This percentage effectively represents the percentage of V_{CCO} pins used by the device.

Examples: Using an XC2V3000 FF1152

Single bank and full device examples are provided.

Single Bank Example

In a hypothetical design, Bank 0 has 80 outputs in it. Each is configured as a 3.3V LVCMOS 12 mA Fast driver.

The limit for 3.3V LVCMOS 12 mA Fast drivers in the SSO table of the datasheet is 10 per V_{CC}/GND pair. There are 13 V_{CCO} pins per bank on this device. Therefore, the limit for this type of I/O driver is 130 per bank.

This bank uses 80 outputs. Therefore the percentage of the total bank 0 budget that is used is:

Bank 0 Percentage Used = Used/Limit = 80/130 = 62%

Full Device Example

In this example, the utilization of all I/Os for one device is listed in Table 2, as well as the per-bank SSO limits for each standard (Table 3), computed from the number per V_{CC}/GND pair SSO limits in the *Virtex-II Platform FPGA User Guide* (**Reference #3**).

Table 2: **I/O Utilization for Each Bank in Full Device Example**

Bank Number	Voltage	I/O Utilization	I/O Standard
Bank 0	3.3V	80	LVCMOS_12F
Bank 7	3.3V	80	LVCMOS_12F
Bank 1	1.5V	16	LVDCI
Bank 6	1.5V	16	LVDCI
Bank 2	1.8V	32	HSTL_1
		45	LVCMOS_12F

Table 2: **I/O Utilization for Each Bank in Full Device Example**

Bank Number	Voltage	I/O Utilization	I/O Standard
Bank 3	1.8V	32	HSTL_1
		45	LVCMOS_12F
Bank 4	1.8V	32	HSTL_1
		45	LVCMOS_12F
Bank 5	1.8V	32	HSTL_1
		45	LVCMOS_12F

Table 3: **SSO Limits for I/O Standards of Full Device Example**

I/O Standard	SSO Limit Per Bank
3.3V LVCMOS_12F	130
1.5V LVDCI	130
1.8V HSTL_1	260
1.8V LVCMOS_12F	117

The budget for banks 0, 7, 1, and 6 are computed as in the single-bank example. Banks 2, 3, 4, and 5, however, all have two I/O standards in them. For these banks, the budget is computed for each standard separately, and then the two are combined.

For Banks 2, 3, 4, and 5:

1.8V HSTL_1:

% used = used/limit = 32/260 = 13%

1.8V LVCMOS_12F:

% used = used/limit = 45/117 = 39%

Total budget for each bank:

13% + 39% = 52%

Table 4 summarizes the budgets for each bank of the device.

Table 4: **Budgets for Each Bank in Full Device Example**

Bank Number	Budget
Bank 0	62%
Bank 7	62%
Bank 1	12%
Bank 6	12%
Bank 2	52%
Bank 3	52%
Bank 4	52%
Bank 5	52%

The number of V_{CCO} pins used in a bank (Table 5) is simply the number of V_{CCO} pins in a bank times the percentage of the SSO budget used.

Table 5: **Number of V_{CCO} Pins Used**

Bank Number	Calculated	Number of Pins Used
Bank 0	13 pins × 62%	8 pins
Bank 7	13 pins × 62%	8 pins
Bank 1	13 pins × 12%	2 pins
Bank 6	13 pins × 12%	2 pins
Bank 2	13 pins × 52%	7 pins
Bank 3	13 pins × 52%	7 pins
Bank 4	13 pins × 52%	7 pins
Bank 5	13 pins × 52%	7 pins

Step 2: Designing the Generic Bypassing Network

A number of Xilinx test boards and customer designs were analyzed to discern some trends of successful PDS designs. In 80% to 100% utilized designs with power supply noise on the order of half the maximum allowed power supply noise ($V_{RIPPLE}/2$), the PDS generally has approximately one capacitor per V_{CC} pin on a per-supply basis. The generic bypassing network is designed with this range of capacitors in mind. The pro-rated number of V_{CCO} pins is used.

Given the number of discrete capacitors needed as determined above, a distribution of capacitor values adding up to that total number must be determined. To cover a broad range of frequencies, a broad range of capacitor values must be used. The proportion of high-frequency capacitors to low-frequency capacitors is an important factor.

The objective of a parallel combination of a number of values of capacitors is to keep a low and flat power supply impedance over frequencies from the 500 kHz range to the 500 MHz range. Both large value (low frequency) and small value (high frequency) capacitors are needed. Small value capacitors tend to have less of an impact on the total impedance profile, so a greater number of small value capacitors are needed to yield the same impedance level impact as a small number of large value capacitors.

To keep the impedance profile smooth and free of anti-resonance spikes, a capacitor is generally needed at least in every decade of the capacitor value range. The typical ceramic capacitor range generally spans values from 0.001 µF to 4.7 µF. The exact value of these capacitors is not critical. What is critical is having some capacitor value in every order of magnitude over this range. More values are better, as a flatter impedance profile is yielded.

A ratio of capacitors giving a relatively flat impedance is one where the quantity of capacitors is roughly doubled for every decade of decrease in size. In other words, if the bottom three values in the network were 1.0 µF, 0.1 µF and 0.01 µF, the network might have two 1.0 µF capacitor, four 0.1 µF capacitors, and eight 0.01 µF capacitors.

In addition, low-frequency capacitance in the form of tantalum, OS-CON, or electrolytic capacitors is needed. These large capacitors typically have a higher ESR than ceramic chip capacitors, making them effective over a wider range of frequencies. This also makes the capacitors less likely to contribute to anti-resonance spikes. For this reason, it is not necessary to maintain the rule of one value per decade. Generally, one value in the 470 µF to 1000 µF range is sufficient.

A set of percentages is helpful for calculating these ratios based on the total number of capacitors (Table 6).

Table 6: **Capacitor Value Percentages for a Balanced Decoupling Network**

Capacitor Value	Quantity Percentage	Capacitor Type
470 µF to 1000 µF	4%	Tantalum
1.0 to 4.7 µF	14%	X7R 0805
0.1 to 0.47 µF	27%	X7R 0603
0.01 to 0.047 µF	55%	X7R 0402

For every power supply except V_{REF}, these ratios should be roughly maintained. For V_{REF} supplies, the values should be distributed in a 50/50 ratio of 0.1 µF to 0.47 µF capacitors and 0.01 µF to 0.047 µF capacitors. Since the primary function of V_{REF} decoupling capacitors is to reduce the impedance of V_{REF} nodes thus reducing crosstalk coupling, very little low-frequency energy is needed. Therefore, only capacitors in the 0.01 µF - 0.47 µF range are necessary.

1.5V Supply Example

In this example, the 1.5V supply for the Virtex-II device supplies V_{CCO} for banks 1 and 6, and V_{CCINT}. There are 44 V_{CCINT} pins on this device. Banks 1 and 6 were previously calculated to use two pins each. Adding the 44 V_{CCINT} pins and the four V_{CCO} pins for Banks 1 and 6 equals 48 pins. Therefore, there should be 48 capacitors total on the 1.5V supply. Table 7 shows how the quantity of each value of capacitor is determined.

Table 7: **Calculation of Capacitor Quantities for 1.5V Supply Example**

Capacitor Value	Calculated	Quantity of Capacitors
680 µF	48 pins x 4% = 1.92	2
2.2 µF	48 pins x 14% = 6.72	7
0.47 µF	48 pins x 27% = 12.6	13
0.047 µF	48 pins x 55% = 26.4	26

This calculation gives a first-pass estimate of the capacitors necessary for the 1.5V supply. Changes can be made to the exact number of capacitors to accommodate different values and to make the supply more symmetric (e.g., using eight 2.2 µF capacitors instead of seven for a more standard the PCB layout). Capacitor values can also be modified according to the specific constraints of the design (e.g., a pre-existing BOM of capacitors). This process of capacitor selection must be repeated for each supply.

Step 3: Simulation

During simulation, the generic decoupling network is verified and in some cases refined. The designer can experiment with different values of capacitors or different packages to achieve an optimum power supply impedance profile for the constraints of the system. A number of levels of PDS design tools available from various EDA vendors are listed in Appendix D: EDA Tools for PDS Design and Simulation.

The simulation circuit is essentially a parallel combination of the decoupling capacitors with associated parasitics. The simulator calculates the aggregate impedance over the pertinent range of frequencies. The equivalent circuit can be created and analyzed in SPICE (see Appendix C: SPICE Simulation Examples for an example SPICE deck) or in one of the tools listed in Appendix D: EDA Tools for PDS Design and Simulation. A more limited but still effective approach is to plot the impedance profile in a spreadsheet tool (for example, Microsoft Excel).

Note that a lumped RLC simulation of this type does not reflect the distributed RLC properties of the V_{CC} and Ground planes of the PCB stackup. The effects of these planar structures usually begin to manifest in the 500 MHz range, and are dependent on the geometries of the planes (for example, length and width). These are difficult to predict without the use of a distributed model, such as what is offered by a tool like Speed2000, SIwave, Specctraquest

Power Integrity, or a full-mesh RLC SPICE simulation. For this reason, it is unwise to draw any conclusions from the results of a lumped RLC simulation above 500 MHz.

In using any of these tools to simulate a bypassing network, it is important to have accurate parasitic values. Obtaining accurate self-parasitic data from the capacitor vendor or from in-house testing is important. The mounting parasitics lying in the path between the bypass capacitor and the FPGA also need to be taken into account. These parasitics combined in series give the mounted capacitor parasitic resistance and inductance. The section on Mounting Inductance covers the details of mounting modeling. Appendix B: Calculation of Via Inductance lists equations for via parasitic inductance. A more accurate inductance number for a particular geometry can be obtained using a field solver such as Ansoft's HFSS. For the following simulation, a value of 0.8 nH to 0.9 nH in mounting inductance was added to each capacitor's parasitic self-inductance to come up with L_{IS}. This parameter reflects the inductance of small capacitor mountings in a board on the order of 60 mils thick. Thicker board stackups have a higher associated via inductance.

Figure 8 shows a simple impedance plot from a simulation of the parallel combination of these capacitors, taking into account their parasitics and the approximate parasitics of the PCB. An equivalent SPICE netlist is included in Appendix C: SPICE Simulation Examples. Table 8 lists the capacitor quantities, values, and parasitic values used in the simulation. The RLC characteristics of the V_{CC} and GND planes of the PCB are not taken into account.

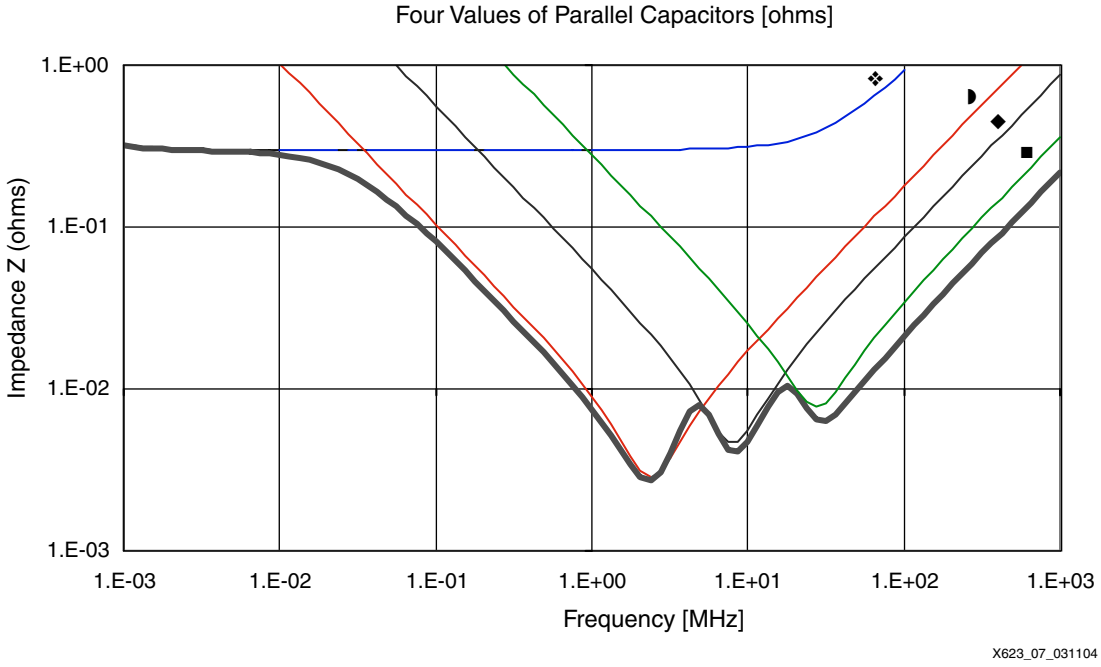


Figure 8: **PDS Impedance Versus Frequency Plot**

Table 8: **Values Used in Impedance Plot of Figure 8**

Quantity	Symbol	Package	Capacitive Values (µF)	Parasitic Inductance (nH)	Parasitic Resistance (ohms)
2	v	E	680	2.8	0.57
7	w	0805	2.2	2.0	0.02
13	u	0603	0.22	1.8	0.06
26	n	0402	0.022	1.5	0.20

This collection of capacitors is a good start. The impedance is below 0.033 Ω from 500 KHz to 150 MHz, and increases to 0.11 Ω at 500 MHz. Over this range there are no significant anti-resonance spikes. These capacitors are used in the board design.

Step 4: Building the Design

At this stage, the PCB is laid out with the final capacitor networks verified in simulation. The board is built. See earlier sections on capacitor placement and land geometries for detailed layout information.

Step 5: Measuring Performance

In the performance measurement step, measurements are made to determine whether the PDS is adequate for the devices it is serving. Determining whether or not a bypassing network is adequate for a given design is relatively simple. The measurement is performed with a high-bandwidth oscilloscope (1 GHz oscilloscope and 1 GHz probe at minimum), on a design running realistic test patterns.

Noise Magnitude Measurement

The measurement is taken either directly at the power pins of the device, or across a pair of unused I/O, one driven High and one driven Low. V_{CCINT} and V_{CCAUX} can only be measured at the PCB backside vias. V_{CCO} can also be measured this way, but more accurate results are obtained measuring fixed signals at unused I/Os in the same bank.

When making the noise measurement at the back-side of the board, it is necessary to take into account the parasitics of the vias in the path between the measuring point and FPGA, as any voltage drop occurring in this path is not accounted for in the oscilloscope measurement. Backside via measurements also have a potential pitfall. Many times, decoupling capacitors are mounted directly underneath the device, meaning that the capacitor lands may connect to these V_{CC} and GND vias directly with surface traces. These capacitors can confound the measurement, as they act like a short circuit for high-frequency AC current. To make sure such capacitors do not short the measurement, capacitors at the measurement site must be removed.

When measuring V_{CCO} noise, the measurement can be taken at a pair of I/O pins configured as strong drivers to logic 1 and logic 0. This technique, when performed correctly, can also show die-level noise.

Measuring the driven logic 1 against the driven logic 0 shows the degree of rail collapse at the die. Measuring a driven logic 0 against PCB ground shows the amount of ground bounce the die is experiencing relative to the PCB PDS. Since all grounds are common at the die and package levels of the device (excepting AGND of MGTs), a ground bounce measurement taken at an unused I/O pin shows the ground bounce of all supplies. Rail collapse measurements, on the other hand, only apply to V_{CCO}.

To make these measurements, the oscilloscope should be in infinite persistence mode, to acquire noise over a long time period (many seconds or minutes). If the design operates in a number of different modes, utilizing different resources in different amounts, these various conditions and modes should be in operation while the oscilloscope is acquiring the noise measurement. Noise measurements should be made at a few different V_{CC}/GND pairs on the FPGA to eliminate the effects of a local noise phenomena.

Figure 9 shows an instantaneous noise measurement taken at the V_{CCINT} pins of a sample design. Figure 10 shows an infinite persistence noise measurement of the same design. Since the infinite persistence measurement catches ALL noise events over a long period, it obviously yields more relevant results.

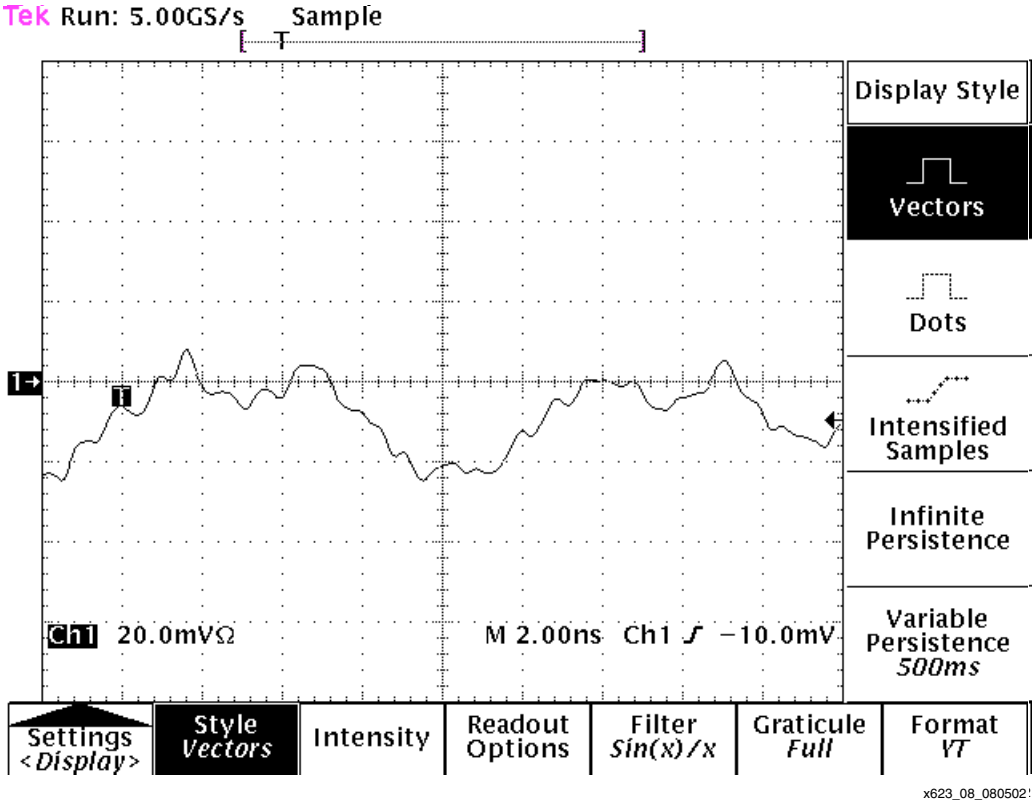


Figure 9: **Instantaneous Measurement of V_{CCO} Supply, with Multiple I/O Sending Patterns at 100 MHz**

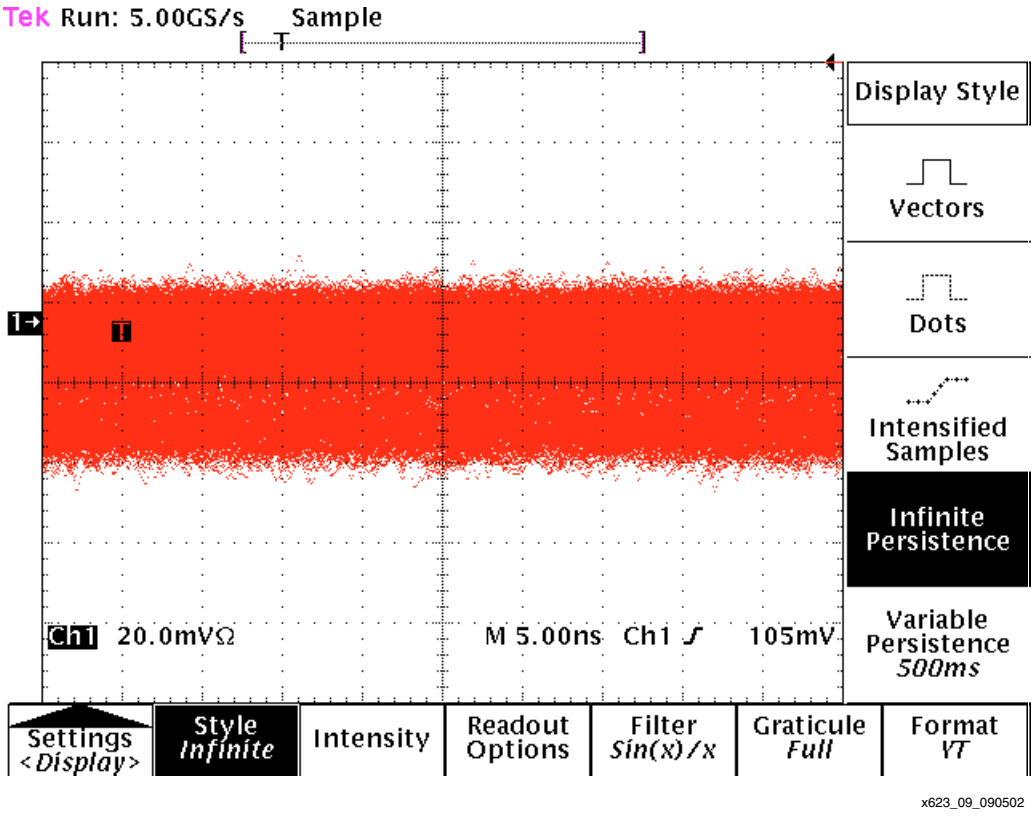


Figure 10: **Infinite Persistence Measurement of Same Supply**

This measurement represents the peak-to-peak noise. If it is greater than or equal to the maximum V_{CC} ripple voltage specified in the datasheet (10% of V_{CC}), then the bypassing network is not adequate. The maximum voltage ripple allowed for this particular supply, with a nominal value of 1.5V DC, is 10% of this, or 150mV. The scope shots show noise in the range of 60 mV. From this measurement, it is clear that the decoupling network is adequate.

If, however, the measurement showed noise greater than 10% of V_{CC}, the PDS would be inadequate. To have a working, robust design, changes should made to the PDS. A greater number of capacitors, different capacitance values, or different numbers of the various decoupling capacitor values will bring the noise down.

Having the necessary information to improve the decoupling network requires additional measurements. Specifically, measurement of the noise power spectrum is necessary to determine the frequencies where the noise resides. There are many ways to do this. A spectrum analyzer works well as does an oscilloscope with FFT math functionality. Alternatively, a long sequence of time-domain data can be captured from an oscilloscope and converted to frequency domain using MATLAB or other software supporting FFT. It is also possible to get a basic feel for the frequency content of the noise by simply looking at the time-domain waveform and measuring the individual periodicities present in the noise.

Noise Spectrum Measurements

A spectrum analyzer is a frequency-domain instrument. It shows the frequency content of a voltage signal at its inputs. When used to measure an inadequate PDS, the user can see the exact frequencies where the PDS is inadequate. Excessive noise at a certain frequency indicates a frequency where the PDS impedance is too high for the transient current demands of the device. Armed with this information, the designer can modify the PDS to accommodate the transient current at the specific frequency. This is accomplished either by adding capacitors with resonant frequencies close to the frequency of the noise or by lowering the PDS impedance at the critical frequency through other means.

The noise spectrum measurement should be taken in the same place as the peak-to-peak noise measurement — directly underneath the device, or at a pair of unused I/O driven High and Low. A spectrum analyzer takes its measurements through a 50 Ω cable, rather than through an active probe like the oscilloscope. One of the best ways to attach the cable for measurements is through an SMA connector tapped into the power and ground planes in the vicinity of the device. In most cases this is not available. Another way to attach the cable for measurement of noise in the power planes is to remove a decoupling capacitor in the vicinity of the device, and solder the center conductor and shield of the cable directly to the capacitor lands. Alternatively, a probe station can be used.

In most cases, distinct bands of noise at fixed frequencies are seen. These correspond to the clock frequency and its harmonics. The height of each band represents its relative power. The majority of the energy is usually contained in tight bands around 3 or 4 of the harmonics, with power falling off as frequency increases.

Figure 11 shows an example of a noise spectrum measurement. It is a screenshot of a spectrum analyzer measurement of power supply noise on V_{CCO}, with multiple I/O sending patterns at 150 MHz.

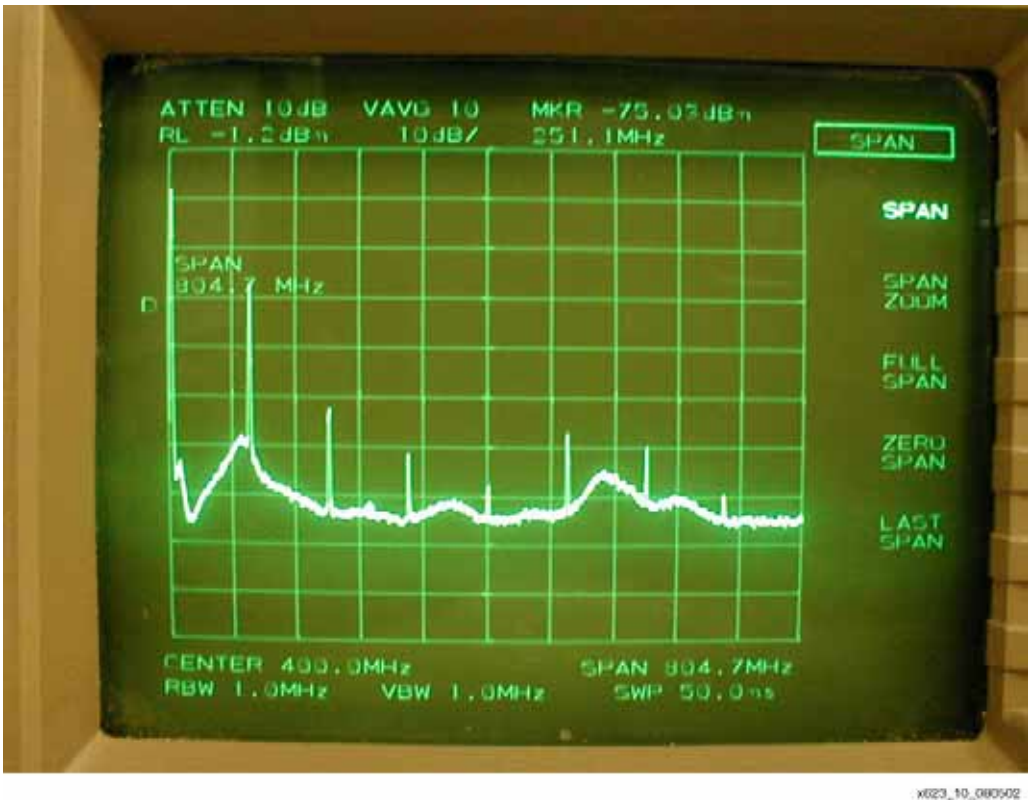


Figure 11: **Screenshot of Spectrum Analyzer Measurement of V_{CCO}**

The noise bands correspond to frequencies where the FPGA has a demand for current but is not receiving it from the PDS. This could be because there is not enough capacitance, or because there is enough capacitance but the parasitic inductance of the path separating the capacitors from the FPGA is too great. Whatever the cause, the impedance of the power supply at this frequency is too high. Conversely, at frequencies where there is very little or no noise, the impedance may be lower than it needs to be. To solve these problems, the bypassing network must be modified. New capacitor values, or different quantities of the original values should be chosen.

Step 6: Optimum Bypassing Network Design (Optional)

In cases where a highly optimized PDS is needed, further measurements can be taken to guide the design of a carefully tailored decoupling network. A network analyzer can be used to measure the impedance profile of a prototype PDS, giving an output similar to what was discussed in the simulation section. The network analyzer sweeps a stimulus across a range of frequencies and measures the impedance of the PDS at each frequency. Its output is impedance as a function of frequency.

Since the spectrum analyzer gives an output of voltage as a function of frequency, these two measurements can be used together to determine transient current as a function of frequency.

$$I(f) = \frac{V(f)\ From\ Spectrum\ Analyzer}{Z(f)\ From\ Network\ Analyzer}$$

Armed with an understanding of the design's transient current requirements, the designer can make better PDS choices. With maximum voltage ripple value from the datasheet, the value of impedance needed at all frequencies can be determined. This yields a target impedance as a

function of frequency. Given this, a network of capacitors can be designed to accommodate the transient current of the specific design.

This six-step process lays out a closed-loop method for designing and verifying a power distribution system. Its use ensures an adequate PDS for any design.

Other Concerns and Causes

If this step-by-step method does not yield a design meeting the required noise specifications, then other aspects of the system should be analyzed for possible changes.

Possibility 1: Excessive Noise from Other Devices on Board

When ground and/or power planes are shared among many devices, as is often the case, noise from an inadequately decoupled device can affect the PDS at other devices. RAM interfaces with inherently high transient current demands due to temporary periodic contention and high-current drivers are a common cause; large microprocessors are another. If unacceptable amounts of noise are measured locally at these devices, an analysis should be done on the local PDS and decoupling networks of the component.

Possibility 2: Parasitic Inductance of Planes, Vias, or Connecting Traces

In this case there is enough capacitance in the bypassing network, but too much inductance in the path from the capacitors to the FPGA. This could be due to a bad choice of connecting trace or solder land geometry, too long a path from capacitors to the FPGA, and/or a current path in power vias that traverse an exceptionally thick stackup.

In the case of inadequate connecting trace and capacitor land geometry, it is important to keep in mind the loop inductance of the current path. If the vias for a bypass capacitor are spaced a few millimeters from the capacitor solder lands on the board, the current loop area is greater than it needs to be (Figure 6a). Vias should be placed directly against capacitor solder lands (Figure 6b). Never connect vias to the lands with a section of trace (Figure 6a). Other improvements of geometry are via-in-pad (where the via is actually under the solder land), not shown, and via beside pad (where vias are not at the ends of the lands, but rather astride them), see Figure 6c. Double vias are a further improvement (Figure 6d).

If the inductance of the path in the planes is too great, there are two parameters that can be changed; the length of the electrical path, and the spreading inductance of the planes themselves.

The path length is determined by capacitor placement. Capacitors must be placed close to the power/ground pin pairs on the device being bypassed. This is especially important for the smallest capacitors in the network, since care has been taken to chose capacitors with low parasitic inductance. There is no use in connecting a low-inductance, high-frequency capacitor to a device through a high-inductance path. Larger capacitors inherently have a high parasitic self inductance allowing the proximity to the device to be less important.

The spreading inductance of the planes is controlled by the plane spacing and by the dielectric constant of the material between them. See section on “Plane Inductance”.

When boards are exceptionally thick (greater than 90 mils or 2.3 mm), vias have higher parasitic inductance. In these cases, the following changes to the design should be considered. The first is to move the V_{CC}/GND plane sandwiches close to the top surface the FPGA is on. The second is to place the highest frequency capacitors on the top surface. Both changes together reduce the parasitic inductance of the relevant current path.

Possibility 3: I/O Signals in PCB are Stronger Than Necessary

If noise in the V_{CCO} PDS is still too high after making refinements to the PDS, the I/O interface power can be scaled back. This goes for both outputs from the FPGA and inputs to the FPGA. In some cases, excessive overshoot on inputs to the FPGA can reverse-bias the clamp diodes in the IOBs. This can put large amounts of noise into V_{CCO}. If this condition is occurring, the drive strength of these interfaces should be decreased, or termination should be used (both on input and output paths).

Possibility 4: I/O Signal Return Current Travelling in Sub-Optimal Paths

Excessive noise in the PDS can be caused by I/O signal return currents. For every signal transmitted by a device into the PCB (and eventually into another device), there is an equal and opposite current flowing from the PCB back into the device's power/ground system. If there is no low-impedance return current path available, a less optimal, higher impedance path is used. When this occurs, voltage changes are induced in the PDS.

This situation can be improved by ensuring that every signal has a closely spaced and fully intact return path. Various strategies could be required including restricting signals to only a few of the available routing layers, and providing low-impedance paths for AC currents to travel between reference planes (decoupling capacitors at specific locations on the PCB).

Conclusion

This application note is an overview of the important principles of power distribution systems, followed by a step-by-step process for designing a PDS. This is an iterative method of PDS design, where the designer first creates a generic network, simulates and refines it, then measures it, and then refines it again based on the measured results is described. When this method fails to give an acceptable result, other possible contributors to the problem are explored. Through the use of this method, all PDS problems can be resolved.

References

1. Xilinx, Inc., *RocketIO Transceiver User Guide*, UG024
2. Larry D. Smith "Decoupling Capacitor Calculations For CMOS Circuits," Proceedings EPEP Conference, November 1984
3. Xilinx, Inc., *Virtex-II Platform FPGA User Guide*, UG002
4. Frederick W. Grover Ph.D. "Inductance Calculations: Working Formulas and Tables", D. Van Nostrand Company, Inc. 250 Fourth Avenue New York 1946

Appendix A: Glossary

Land: A section of exposed metal on the surface of a PCB where surface-mount devices are soldered

Network Analyzer: An instrument used to measure the frequency-domain characteristics of electrical networks. The electrical characteristics of power distribution systems are often measured using a network analyzer.

Oscilloscope: An instrument used to show the time-domain voltage of a signal. Power supply noise is the signal measured when establishing the magnitude of noise voltage on a power supply.

Sandwich: A pair of planes in a PCB stackup separated only by dielectric material, no signal layer is in-between. In most cases, one of these planes is at ground potential and the other plane carries power. Also known as buried capacitance.

Spectrum Analyzer: An instrument used to measure the frequency content of a signal. Power supply noise is the signal measured when establishing the characteristics of a power distribution system.

Stackup: The series of layers in a PCB is often referred to as a stackup. Multi-layered boards are comprised of alternating layers of signal routing or plane metal and dielectric material. The dielectric material also serves as a structural substrate.

Via: A vertical connection in a PCB, usually formed by drilling a hole through the PCB and plating the walls of this hole with conductive material. Vias make electrical connections between different layers of a PCB. Vias can represent impedance discontinuities when they are in a signal path, and represent additional parasitic inductance when they are in a power distribution path (both are undesirable). The parasitic inductance formula is shown in Appendix B: Calculation of Via Inductance.

Voltage Ripple: Power supply noise is often referred to as voltage ripple. The maximum voltage ripple corresponds to the maximum amount of power supply variation allowed by a part's absolute maximum ratings.

Appendix B: Calculation of Via Inductance

Via inductance is a major contributor to the parasitic inductance of a capacitor mounting. The dimensions of a via largely determine its parasitic inductance. Equation 5, from Grover (**Reference #4**), is used to determine the self-inductance of a single filled via based on its length and diameter. Dimensions are in inches and nanohenries.

$$L = 5.08 \times h \times \left[\ln\left(\frac{4 \times h}{d}\right) - 0.\dot{7}5 \right] \qquad \text{Equation 5}$$

Example

To calculate the inductance of a via going from the bottom surface of the board to the top surface of the board, use the board finished thickness for via length: the board finished thickness at 62 mils, the via diameter at 3 mils. There are 1000 mils in an inch.

h = 0.062 in
d = 0.003 in

$$L = 5.08 \times h \times \left[\ln\left(\frac{4 \times h}{d}\right) - 0.\dot{7}5 \right]$$

$$L = 5.08 \times 0.062 \times \left[\ln\left(\frac{4 \times 0.062}{0.003}\right) - 0.\dot{7}5 \right]$$

L = 5.08 x 0.062 x 3.67 nH

L = 1.15 nH

This result is the self-inductance of a single via. The self-inductance is only one part of the total inductance of the current loop the via is a part of. Since the mutual inductance of vias with opposing currents (power and ground) has its own effect on the total inductance, it should be taken into account when greater accuracy is desired. The mutual inductance of closely spaced complementary vias lowers the total inductance by a small amount.

Appendix C: SPICE Simulation Examples

This appendix demonstrates the method used to simulate decoupling capacitor networks in SPICE. HSPICE techniques are discussed here. Other variants of SPICE or dedicated PDS simulation software can also be used. The simulation referenced below is purely for illustrative purposes. Simulator details are beyond the scope of this discussion and are left to the readers' investigation. The HSPICE result is included in Figure 12. A schematic representation is included in Figure 13.

These capacitor networks represent the capacitance and parasitics of an 18-capacitor network. The general capacitor array impedance calculation follows these steps:

1. Formulate a netlist for the L-C-R network
2. Understand where the input node and output node are located
3. Apply an AC stimulus to the input port
4. Run an AC analysis on the L-C-R network
5. Measure the input current as well as the input AC voltage
6. Formulate Z = V/I
7. Plot the result using a log scale for ease of viewing

In this approach, the AC stimulus is set to 1A. The AC Analysis directive sweeps an AC current waveform across a prescribed set of frequency points. The number of frequency points per

decade is commented in the appended HSPICE netlist. With the AC current magnitude set to 1A, the impedance is calculated based on Z = V/I. Thus, V is the main calculated variable — the voltage at the capacitor array positive node.

Two other details complete the SPICE decks:

1. There is a DC bias resistor to ground
2. There is a small input resistor connecting the AC source to the L-C-R network (this is optional)

Item 1 is necessary to decrease simulation time. It allows SPICE to quickly calculate an operating point for the circuit prior to AC analysis. This is accomplished by providing SPICE a DC path to the L-C-R network (to ground by way of the bias resistor). Item 2 is optional, but convenient. It provides a component to monitor the input current to the L-C-R network.

For viewing the simulated impedance result in HSPICE, the `.net` directive is executed in order that HSPICE calculates Z_{IN} for direct plotting.

HSPICE Netlist

The HSPICE Netlist is available on the Xilinx FTP site:
http://www.xilinx.com/bvdocs/appnotes/xapp623.zip

HSPICE Output

Figure 12 shows the HSPICE output: Z_{IN}(MAG) using the AWAVES graphical viewer.

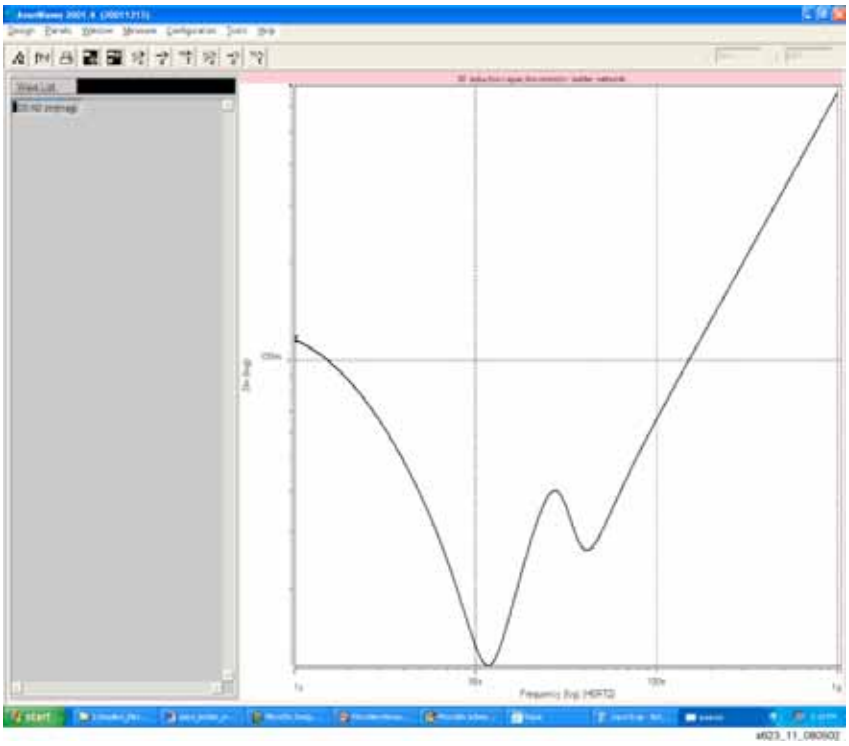


Figure 12: **HSPICE Output**

Schematic Circuit

Figure 13 shows a capacitor array with corresponding parasitic inductance and resistance.

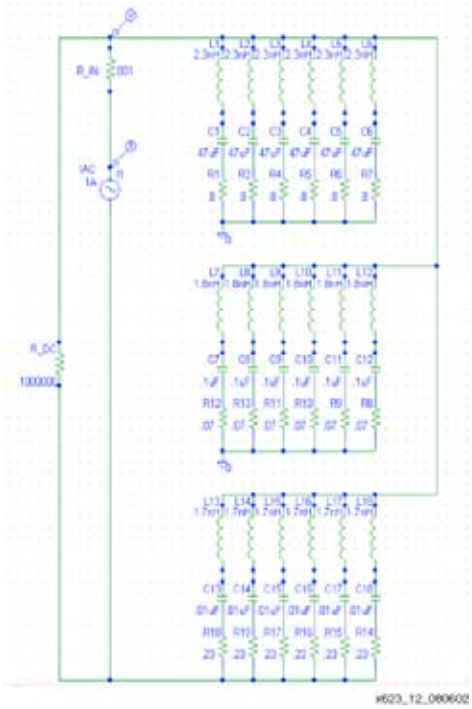


Figure 13: **Schematic Circuit**

Appendix D: EDA Tools for PDS Design and Simulation

Table 9 lists the some vendors of EDA tools for PDS design and simulation.

Table 9: **EDA Tools for PDS Design and Simulation**

Tool	Vendor	Website URL
SIwave	Ansoft	http://www.ansoft.com
Specctraquest Power Integrity	Cadence	http://www.cadence.com
Speed 2000	Sigrity	http://www.sigrity.com
Star HSPICE	Synopsys	http://www.synopsys.com
UCADESR3.exe	UltraCAD	http://www.ultracad.com

Get the Latest Virtex Info Delivered to Your Desktop

Would you like to write for Xcell Publications?

It's easier than you think.

Our Xcell Publishing Alliance can help you – from concept research and development, through planning and implementation, all the way to publication and marketing.

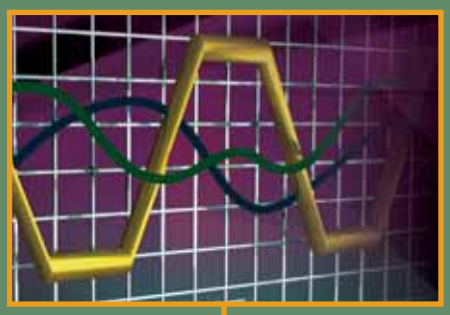
Submit articles for our Web-based *Xcell Online* or any of our printed publications and we will assign an editor and a graphics artist to work with you to make your work look as good as possible. Submit your book concepts and we will bring our broad industry resources to assist you in planning, research, writing, editing, and marketing.

For more information on this exciting and highly rewarding program, please contact:

Forrest Couch
Managing Editor, Xcell Publications
xcell@xilinx.com

Signal Integrity

Solution Guide



"Having good signal integrity usually means controlling unwanted noise on logic signals."

www.xilinx.com/xcell/signal1/

Published by

Xcell Publications

XILINX®
The Programmable Logic Company™

Corporate Headquarters
Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
Tel: (408) 559-7778
Fax: (408) 559-7114
Web: www.xilinx.com

European Headquarters
Xilinx
Citywest Business Campus
Saggart,
Co. Dublin
Ireland
Tel: +353-1-464-0311
Fax: +353-1-464-0324
Web: www.xilinx.com

Japan
Xilinx, K.K.
Shinjuku Square Tower 18F
6-22-1 Nishi-Shinjuku
Shinjuku-ku, Tokyo
163-1118, Japan
Tel: 81-3-5321-7711
Fax: 81-3-5321-7765
Web: www.xilinx.co.jp

Asia Pacific
Xilinx, Asia Pacific Pte. Ltd.
No. 3 Changi Business Park Vista, #04-01
Singapore 486051
Tel: (65) 6544-8999
Fax: (65) 6789-8886
RCB no: 20-0312557-M
Web: www.xilinx.com

Distributed By:

FORTUNE 2005 100 BEST COMPANIES TO WORK FOR

© 2005 Xilinx Inc. All rights reserved. The Xilinx name is a registered trademark; CoolRunner, Virtex, Spartan, Virtex-II Pro, RocketIO, System ACE, WebPACK, HDL Bencher, ChipScope, LogiCORE, AllianceCORE, MicroBlaze, and PicoBlaze are trademarks; and The Programmable Logic Company is a service mark of Xilinx Inc. PowerPC is a trademark of International Business Machines Corporation in the United States, or other countries, or both. All other trademarks are the property of their owners.

Printed in U.S.A.

PN 0010869