

A Signal Integrity Test Bed for PCB Buses

Jihong Ren and Mark R. Greenstreet*

Department of Computer Science, University of British Columbia
jihong,mrg@cs.ubc.ca

Abstract

Research in high-speed interconnect requires physical test to validate circuit models and design assumptions. At multi-Gbit/sec rates, physical implementations require custom circuit design, teams with many designers, long design cycles, and expensive test equipment. By building a “scale model” that operates at bit rates of 50-100 Mbits/sec, we obtain order of magnitude reductions in cost and design time. We present a simple, inexpensive test bed implemented using a PC and inexpensive graphics cards. To demonstrate the effectiveness of our test bed, we use it to validate novel methods for synthesizing crosstalk equalization filters.

1. Introduction

Increasing chip speeds and integration densities place ever greater demands on the performance of off-chip interconnect. Because the technologies for chip packaging, printed circuit boards, and connectors improve much more slowly than that for silicon, designers require higher and higher bandwidths from copper wires that have changed little over many fabrication generations. To meet this demand, designers make increasing use of on-chip signal processing to maximize the utilization of off-chip interconnect [1]. Effectively, this uses more transistors on-chip to compensate for the limitations of the interconnect between chips.

The design of high-speed interconnect is a multi-faceted effort that includes the design of high-speed DACs and ADCs, low jitter PLLs and DLLs, state-of-the-art packaging, detailed electrical modeling of the interconnect, and our research focus, equalizing filters. Complete designs require custom chip design, many designers, expensive equipment, and design cycles of a year or more. While such efforts are necessary to bring interconnect solutions to the point that they can be utilized by system architects, only a few research groups can undertake such resource-intensive efforts.

Smaller groups can focus on particular interconnect challenges by using a combination of simulation-based studies and “scale model” implementations. Simulations provide a practical way to examine issues in cutting edge interconnect. However, simulations inevitably rely on simplified models for the bus and connectors, as well as for clock jitter and other phenomena that degrade signal integrity. For example, accurate electrical models for buses and connectors are difficult to obtain without measurement as they require 3D solutions to Maxwell’s equations. Thus, we need a physical implementation while avoiding massive design efforts and expensive test equipment. We achieve this by targeting much lower bit-rates and deliberately designing PC board buses to exacerbate signal integrity issues at these lower frequencies. This “scale model” approach has numerous advantages: it is inexpensive to implement; the design cycle is dramatically shortened; and we can easily alter the test bed to examine the impact of varying individual components of the link. Unlike simulation alone, the physical implementation forces us to address the full range of issues arising in an actual design.

Section 3 describes how our test bed uses commodity PC graphics cards to provide analog channels operating at up to 300M samples/sec. We demonstrate the effectiveness of our test bed by using it to validate our designs of novel equalization filters. Section 2 summarizes our previous work in synthesizing optimal equalizing filters, and section 4 shows the use of our test bed for validating these filter designs. These experiments show the impact of timing jitter and inaccuracies in channel estimation. The test bed also confirms that our filters offer dramatic improvements in signal integrity. Its low cost and simplicity should allow our test bed design to be easily used and modified for a wide range of research in signal integrity and mixed-signal design.

2. Equalization

Figure 1 shows the structure of a typical channel with pre-equalization. A filter is assigned to each wire of the bus. Each filter takes as input a data bit and one or more neighbouring bits in each direction and outputs a predistorted signal for one wire of the bus. An ideal equalizing filter would

* This research has been supported by grants from BC-ASI, Intel, NSERC, and SUN Microsystems.

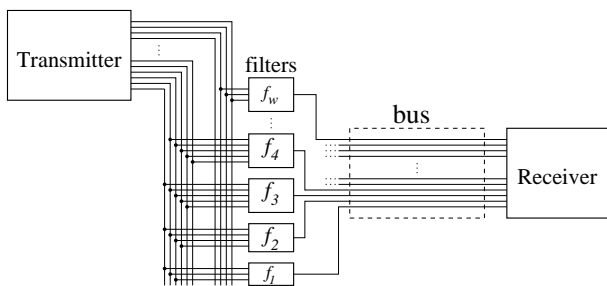


Figure 1. A typical channel with pre-equalizing filters for crosstalk cancellation.

have a transfer function that is the exact inverse of that of the bus with some delay, in which case the concatenation of the filter and the channel would convey data at arbitrarily high rates with perfect fidelity. However, power, bandwidth, and area constraints preclude implementing such a filter. Our research develops methods for synthesizing optimal filters under practical constraints.

Equalization has been used effectively with multi-Gbs links to compensate for high frequency losses [2, 3, 4] and for nearest neighbour crosstalk cancellation [5]. Typically, least squares (LSQ) optimization has been used to determine the filter coefficients. However, LSQ minimizes the average error, whereas standard measures of signal integrity such as eye-height correspond to the worst-case performance. Because the system is linear, contributions from each bit times on each wire can be considered independently. For the worst-case scenario, the total disturbance is simply the sum of the absolute values of those individual contributions from other wires and from other bit times. Based on this observation, we developed linear programming (LP) approaches for optimizing worst-case performance and presented practical implementations of the algorithms [6, 7]. Our methods allow the direct optimization of eye-height or an eye-mask as well as constraining the maximum output of the filter and maximum overshoot at the receiver and identification of the worst-case data patterns for the channel. Simulation results show that our LP based methods achieve significantly greater signal integrity and data rates than LSQ methods for channels with significant crosstalk. The test bed described here allows us to demonstrate these filters with real, physical channels.

3. The Testbed

We designed our test bed around inexpensive VGA graphics cards. Using dual-port, PCI graphics cards and a typical PC with five available PCI slots, we can implement a test bed with up to 30 analog channels. To use

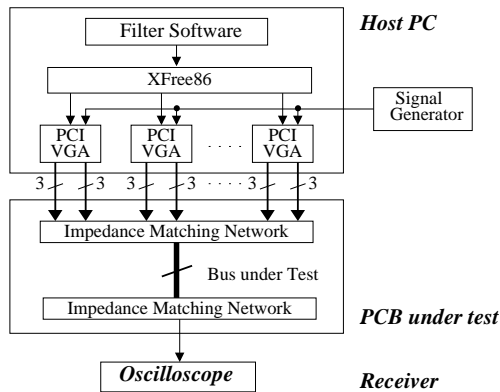


Figure 2. The Testbed

graphics cards as transmitters in a synchronous transmission system, we have to synchronize the video outputs so that they have identical pixel rates and so that frames and scanlines are aligned. Section 3.2 presents our solution to synchronizing the graphics cards. Typical graphics cards support pixel rates of up to 400MHz. To compensate for high-frequency losses, our equalizing filters have sample rates that are a small multiple of the data rate. For example, using four samples per data bit limits the data rate of our test bed to 60~100Mbits/sec. To show the effectiveness of the filter synthesis methods, we intentionally designed a bus with severe crosstalk and dispersive losses at such low data rates. Section 3.3 describes the issues involved in designing a sufficiently bad PCB bus. Finally, section 3.4 describes how we measure the test bed's electrical parameters to validate simulation models and provide a basis for filter synthesis.

3.1. System Overview

Figure 2 sketches our test bed. It consists of three major components: a host PC with multiple PCI graphics cards, a PCB with a bus under test and impedance matching networks, and an oscilloscope as a receiver. The bus under test is driven by the video outputs of the graphics cards. We used ATI Radeon 7000 VE graphics cards at a cost of about \$US 60 each. Each card provides separate VGA and DVI graphics ports providing a total of six analog channels per card. A run of seven boards cost \$350, mainly because of the area needed for a long bus. Not counting the oscilloscope and signal generator which are available in most electronic labs, our test bed can be built for less than \$2000.

After characterizing the bus, we use our filter synthesis methods to derive the filter coefficients and determine their worst-case input sequences. We then compute the filter outputs and store them as pixel values that are transferred to the

frame buffers of the graphics cards. As result, the RGB signals from the cards drive each wire of the bus with the values from the corresponding filter. At the receiver end, the oscilloscope records the bus outputs which we then download to the host PC for analysis.

3.2. Synchronizing multiple graphics cards

In our test bed, we match the pixel clocks of the graphics cards by modifying the graphics cards to operate from a common clock. We then use a software approach to align the frames and scanlines from the multiple graphics ports.

The Radeon 7000 graphics card derives its pixel clocks from a 27MHz crystal oscillator. We added connectors that allow us to override the oscillator with the signal from an external signal generator, an Agilent 33250A. Adding the connector for an external clock is the only physical change that we made to the graphics cards.

With the pixel clocks matched, we need to align the frames and scanlines to ensure sufficient scanline overlap to test our filters. This is called “genlock.” While graphics cards with hardware support for genlock are available, they are specialty devices with prices in the thousands of dollars. Using a large number of these expensive cards would increase the cost of our test bed by an order of magnitude. Instead, we adapted a software approach to genlock originally described in [8]. The main idea is to temporarily alter the length of the horizontal and vertical retrace times of each graphics port to bring all of the ports into alignment. The solution in [8] used software observation of the retrace intervals to achieve frame level synchronization across multiple PC’s running real-time Linux. Using the same general approach on a single PC running RedHat Linux and XFree86, we use an oscilloscope to make precise measurements of sync pulse alignments and achieve synchronization to within a few pixels.

VGA standards [9] specify the registers that control the video signal generation. To access these registers, we modified the graphics card driver in XFree86 [10] and extended the VIDMODE feature of XFree86 so that synchronization can be performed at the application level. With this method we achieve synchronization to within 15 pixels. In particular, this software approach is limited because the graphics card that we used restricts the number of pixels in a scanline to be a multiple of eight; it doesn’t provide software observation of the horizontal synchronization events; and changes made to the timing registers appear to cause some small random perturbation in hsync timing. For our application, the residual offset is not a problem and can be compensated by shifting the location of filter outputs in the frame buffer. Using the 1600×1200 , video mode (the longest supported scanline for the DVI channel) with a filter oversampling rate of four pixels per bit provides test sequences of

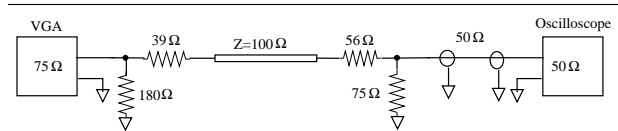


Figure 3. Impedance Matching Networks

up to 396 bits per scanline which is more than adequate for our purposes.

3.3. A Scale Model for High-speed Buses

The Radeon 7000 graphics cards provide 300MHz RAMDACs. With four filter samples per data bit, this sets the upper limit for the data rate of our test bed to be 75MHz. To obtain a signaling environment at these bit rates similar to that of a multi-Gbs bus, the bus geometry must promote crosstalk. To that end, we fabricated a bus that is 1 meter long with 6 mil (0.15mm) wide traces with 6 mil spacing running 93 mils (2.36mm) over the ground plane with a standard, FR-4 dielectric and 0.5oz (0.017mm) copper thickness. The large separation to the ground plane increases the inductive coupling to produce significant crosstalk at relatively low data rates. Using the HSPICE 2D field solver, we extracted a model for the bus and simulations showed that without equalization it should have a closed data eye at 70 Mbits/sec. The board is 15in \times 10in (38cm \times 25cm).

Achieving sufficient crosstalk for our experiments necessitated a bus geometry with a relatively high characteristic impedance, about 100 Ω . The RGB outputs from the graphics card are 75 Ω , and the input to the oscilloscope (an HP 54522A) is 50 Ω . To address these impedance mismatches, we considered transformers and resistive networks. To use transformers, we would have to restrict our data sequences to minimize the low-frequency content of the transmitted signals. On the other hand, insertion loss is the main disadvantage of using a resistive network. We chose the resistive network approach for its ease of implementation and the flexibility of being able to change the termination simply by changing resistor values. Figure 3 shows the resistive impedance matching network that we use. The voltage output from this network has about 40% of the magnitude of an implementation with ideal transformers.

Even with impedance matching networks, there are many discontinuities in the signal path. These include the VGA and DVI connectors, DVI-VGA adapters, header-pin connectors, and some non-impedance controlled segments on the board and between the VGA connectors and the header pins. In practice, connectors and chip packages are often the dominant sources of crosstalk and reflections for multi-Gbs links [11].

3.4. Bus Characterization

Channel estimation for high-speed buses and serial links is often performed using a time domain reflectometer (TDR) or a vector network analyzer (VNA). Our test lab has a 2-port VNA. A w -bit bus is a $2w$ -port network and requires $\binom{2w}{2}$ measurement configurations when using a 2-port VNA. For the nine wire bus described in section 4, this requires 153 test configurations with manual changes for each. Furthermore, VNA measurements would not include circuitry and interconnect on the graphics cards such as the DACs, graphic chip packaging, and graphics card copper traces. TDR approaches have similar limitations.

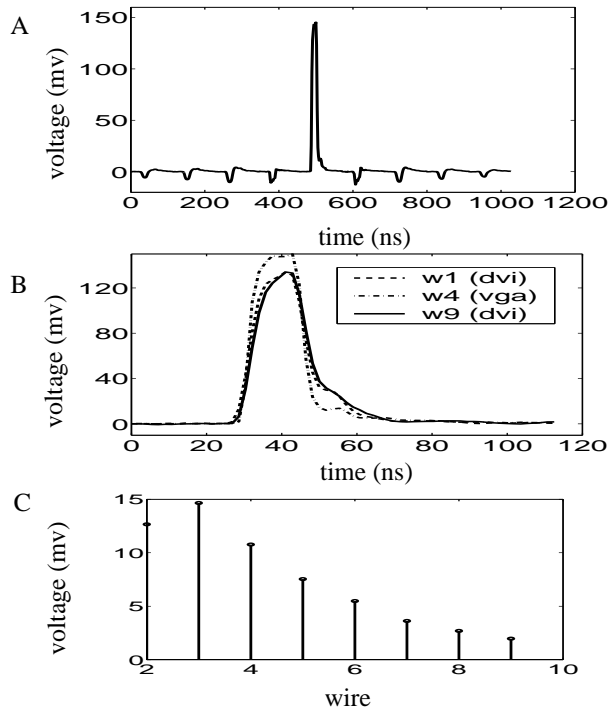
Instead of using the VNA, we perform channel estimation *in situ*. Our optimization methods are formulated with respect to the response on each bus output to a pulse whose width is one data bit time on each bus input. We program the graphics cards to produce such pulses and measure the responses directly using the oscilloscope. This method takes the entire channel including the DACs, connectors, resistive networks, and bus into account. For a w -bit bus, we only need w^2 configurations. This number can be further reduced by sending out bits consecutively on each wire. For example, by sending down a single bit input on three wires consecutively and using a multi-channel oscilloscope, the number of configurations is reduced to $w^2/(3(c-1))$, where c is the number of oscilloscope ports. That's 27 configurations for our nine wire bus with a two-channel oscilloscope. Unlike a VNA which uses a tuned front end, our approach is highly sensitive to noise. As discussed in section 4.1, we average the response to 100 measurements per configuration to obtain satisfactory accuracy.

4. Results

Using our test bed, we have performed some preliminary experiments with our filters. This section describes our *in situ* measurements of bit responses, measurements of filter effectiveness, and the impact of timing jitter. Pragmatism motivated us to choose a nine-wire bus for these experiments. As noted in section 3.4, the effort to characterize the bus response grows quadratically with bus width. We balance the incentive for a small bus to simplify characterization with the need for a wide enough bus to show large scale crosstalk.

4.1. Bus Characterization

Figure 4A shows a complete bit response on the middle wire of the bus, wire 5, given a single, bit-wide input on each wire consecutively. The response to the pulse on the wire itself (Figure 4B) has a peak at $128 \sim 148$ mv, with



A: the cascaded waveform of the responses on wire 5 given a single, bit-wide pulse on each wire of the bus;
B: the responses on wire 1, 4 and 9 given a single, bit-wide pulse on wire 1, 4 and 9 respectively;
C: the maximum magnitude on wire 1 of crosstalk from each other wire.

Figure 4. Bus Bit Responses.

wires driven directly by the VGA port having a higher peak than those driven through DVI→VGA adapters. This is apparently due to the high-frequency losses of the DVI→VGA adapter. Figure 4C shows the maximum crosstalk on wire 1 from other wires. Note that coupling from wire 3 is stronger than that from wire 2; otherwise, crosstalk decreases with distance as expected.

We observed a 5mv peak-to-peak noise floor. With coupling terms from 2mv to 15mv, we averaged the values from 100 measurements to estimate a bus model. We automated these measurements using GPIB programming of the oscilloscope. From the standard deviation of these measurements, we estimate that our bus model is accurate to within $\pm 3\%$. Using the average from 20 traces degrades the accuracy for filter synthesis to $\pm 8\%$ with little impact on end-to-end signal integrity. This suggests that simple, on-chip circuitry for channel estimation should suffice when implementing our filters in multi-Gbs links.

4.2. Filter Performance

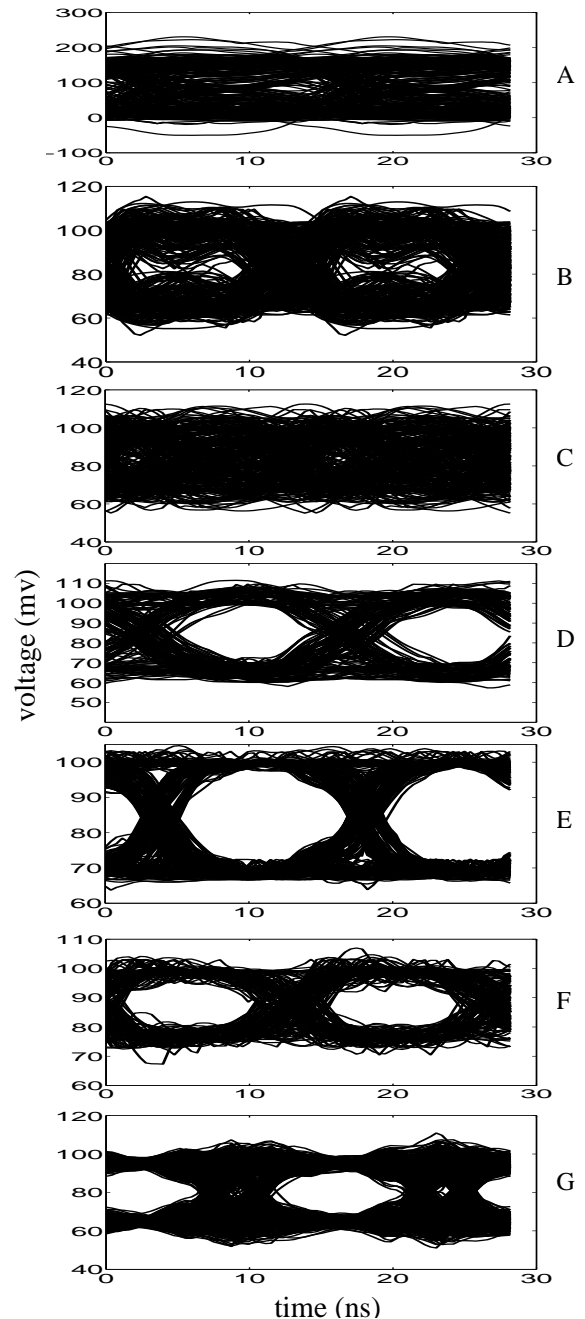
Implementing a filter where every output depends on the data values from every data input would be unacceptable in terms of power, latency, and die area. Thus, we consider filters where each output is computed from the input value for the wire itself and each of its k closest neighbours in both directions. For example, the filter shown in figure 1 is a design with $k = 2$. We write that a filter is $m \times k$ to indicate that the filter has m taps and considers k neighbours to each side. We test each filter using its worst-case input sequence (see [6, 7]) and random variations.

Figure 5 shows eye diagrams measured from the bus with various filters. As predicted by the HSPICE field solver and simulation, this bus has a closed eye at 70Mbit/s, figure 5A. For single-line pre-emphasis, neither the LP nor the LSQ synthesized filters opens the eye. As the LP method optimizes for the worst-case, it can make no progress and produces meaningless filter coefficients. By optimizing for the average case, the LSQ method produces a filter that provides some improvement in signal integrity. Thus, we show a single-line pre-emphasis filter designed by the LSQ method, figure 5B. While the LSQ method cannot open the eye with a 12×1 filter, the LP method achieves 10% eye height and 15% eye width in simulation that neglected jitter. However, due to jitter and voltage noise in the physical system, we do not observe this small eye opening using the test bed, figure 5C. By taking more wires into account, the 12×3 filters designed by the LP method produces a good eye opening, figure 5D. Figure 5E shows that an excellent eye is produced by a full-width filter using the LP method, while figure 5F shows that the corresponding LSQ filter is not nearly as effective. This shows the advantage of the LP method that maximizes eye masks directly. These observations are consistent with simulation results reported in our earlier work [6] and confirm the correctness and practicality of our filter design approach.

4.3. Jitter

High-speed I/O bandwidth should scale with technology as long as the timing uncertainties can be made to scale at the same rate [11]. Clock jitter and channel interference are the dominant causes of timing uncertainty. Although equalizing filters can greatly reduce channel interference, clock jitter can significantly degrade channel performance [12].

We face two timing issues in this test bed: subpixel misalignment and graphics card PLL jitter. The subpixel misalignment between video ports is the residual misalignment after shifting the location of filter outputs in the frame buffer (see section 3.2). We measure this subpixel misalignment and incorporate it into the channel model. Although pixel clock jitter is not specified by graphics card manufactur-



- A: without filter;
- B: with independent pre-emphasis on each wire, LSQ synthesized filters;
- C: with 12×1 LP synthesized filter for nearest-neighbour crosstalk cancellation;
- D: with 12×3 LP synthesized equalizing filter;
- E: with 12×9 LP synthesized equalizing filter;
- F: with 12×9 LSQ synthesized equalizing filter;
- G: simulation for 12×9 LP filter with extreme-case jitter.

Figure 5. Eye Diagrams at 70Mbit/sec/wire.

ers, it is significant. For the cards we used, the jitter appears to be random with a standard deviation of approximately 200ps. The excellent eye-diagrams shown in figure 5 suggest that although we synthesized the equalizing filters assuming perfect timing, the filter design approach tolerates moderate amounts of jitter.

With 200ps rms random jitter, to ensure a bit error rate (BER) of 10^{-12} , the system should tolerate 2.8ns peak-to-peak jitter. With randomly generated extreme-case jitter of either +1.4ns or -1.4ns, we simulated the transmission system with the 12×9 equalizing filter designed by the LP method. The open eye shown in figure 5G suggests that the system provides 10^{-12} BER and further confirms that the filter design approach has some jitter tolerance. However, compared with figure 5D, figure 5G shows reduction in both eye height and eye width. This suggests the importance of taking jitter into account while designing equalizing filters in order to guarantee a specified BER. We plan to incorporate jitter explicitly in our filter synthesis procedure and then use our test bed to validate its effectiveness.

5. Conclusions and Future Work

We have presented a simple, low-cost, signal integrity test bed for PCB buses. We use commodity graphics cards to provide a large number of analog channels at rates of 300-400M samples/sec. By designing buses with exaggerated crosstalk, we can model channels similar to those found in multi-Gbs links at a much lower rate. Because our approach uses commodity components and does not require custom-chip fabrication, we can perform experiments with an order of magnitude or more reduction in cost, time, and effort compared with full-speed links. This allows us to quickly evaluate novel signaling approaches.

We demonstrated our test bed by using it to validate our synthesis procedures for crosstalk cancelling equalization filters. Measurements from the test bed show that we can get results that are comparable to those predicted by simulation in the presence of many real-world non-idealities such as timing jitter and imperfect connectors. We plan to build upon this foundation and extend our methods to include explicit modeling of jitter as well as differential and multi-level signaling. Our test bed will allow simple and fast validation and demonstration of these techniques as we develop them and allow us to identify further areas for improving signaling.

In our work, we have focused on high-speed digital buses. Signal integrity with coupled channels occurs in many other situations including telephone subscriber loop connections [13]. The most obvious limitations of our approach are the blanking of output signals during horizontal and vertical retrace and the offline computation of filter outputs. The former could be overcome by combining

the outputs from two or three VGA ports to produce an uninterrupted signal. The latter might be addressed by using the graphics pipelines of the video cards for signal processing computations. With such changes, our test bed could be used to test a wide variety of mixed-signal applications.

Acknowledgements

Thanks to Dr. John Poulton for many helpful discussions and pointing us to soft genlock. Thanks to Dr. Roberto Rosales for expert assistance in the test lab.

References

- [1] M. Horowitz, C-K.K. Yang, and S. Sidiropoulos. High speed electrical signalling: Overview and limitations. *IEEE Micro*, 18(1):12–24, Jan.–Feb. 1998.
- [2] W.J. Dally and J.W. Poulton. Transmitter equalization for 4-Gbps signaling. *IEEE Micro*, 1:48–56, 1997.
- [3] V. Stojanovic, G. Ginis, and M.A. Horowitz. Transmit pre-emphasis for high-speed time-division-multiplexed serial-link transceiver. *IEEE Trans. Communications*, 38:551–558, 2001.
- [4] J.L. Zerbe, C.W. Werner, et al. Equalization and clock recovery for a 2.5-10 Gb/s 2-PAM/4-PAM backplane transceiver cell. *IEEE J. Solid-State Circuits*, pp. 2121–2130, 2003.
- [5] J.L. Zerbe, R.S. Chau, et al. A 2Gb/s/pin 4-PAM parallel bus interface with transmit crosstalk cancellation, equalization and integrating receivers. In *IEEE Int'l. Solid State Circuits Conf.*, pp. 430–432, 2001.
- [6] J. Ren and M.R. Greenstreet. Synthesizing optimal filters for crosstalk-cancellation for high-speed buses. In *Proc. 40th ACM/IEEE Design Automation Conference*, pp. 592–597, 2003.
- [7] J. Ren and M.R. Greenstreet. Crosstalk cancellation for realistic PCB buses. In *Proc. PATMOS 2004*. Springer, 2004.
- [8] J. Allard, V. Gouranton, et al. Softgenlock: Active stereo and genlock for PC cluster. In *Proc. Joint IPT/EGVE'03 Workshop*, Zurich, Switzerland, May 2003.
- [9] Hardware Level VGA and SVGA Video Programming Information Page. <http://web.inter.nl.net/hcc/S.Weijgers/FreeVGA/home.htm>.
- [10] The XFree86 Project. <http://www.xfree86.org>.
- [11] M. Lee, W.J. Dally, et al. CMOS high-speed I/Os - present and future. In *21st Int'l. Conf. Computer Design*, 2003.
- [12] G. Balamurugan and N. Shanbhag. Modeling and mitigation of jitter in multi-Gbps source-synchronous I/O links. *IEEE J. Solid-State Circuits*, pp. 2121–2130, 2003.
- [13] M.L. Honig, K. Steiglitz, and B. Gopinath. Multichannel signal processing for data communications in the presence of crosstalk. *IEEE Trans. Communications*, 38:551–558, 1990.