A Unified Optimization Framework for Equalization Filter Synthesis

Jihong Ren and Mark Greenstreet

Dept. of Computer Science, University of British Columbia
2366 Main Mall, Vancouver, BC, V6T 1Z4
{jihong,mrg}@cs.ubc.ca

ABSTRACT

We present a novel method for jointly optimizing FIR filters for pre-equalization, decision feedback equalization, and near-end crosstalk cancellation. The unified optimization problem is a linear program, and we describe sparse matrix techniques for its efficient solution. We illustrate our approach with uni- and bi-directional buses using differential signaling in both intra-board and cross-backplane scenarios.

Categories and Subject Descriptors

B.4.3 [**Input/Output and Data Communications**]: Interconnections (Subsystems); J.6 [**Computer Applications**]: Computer-aided Engineering

General Terms

Design, performance.

Keywords

crosstalk, equalizing filters, linear programming, optimal synthesis.

1. INTRODUCTION

On-chip speeds and integration densities have grown exponentially over the past several decades, creating a corresponding demand for high-bandwidth chip-to-chip communication. The ITRS roadmap projects a need for per-pin I/O bit-rates that track clock frequencies, reaching 9.6 Gbits/sec by year 2009 and nearly 25 Gbits/sec by 2018 [1, p. 23-25]. To meet these expectations, either the symbol rate or the number of levels per symbol must grow rapidly. Dielectric and skin effect losses limit the practicality of scaling to ever higher symbol rates. On the other hand, multi-level signaling requires extremely good signal integrity. For example, 8-PAM (three bits/symbol) requires the equivalent of a 2-PAM (binary) channel with a 93% eye-height. In practice, crosstalk, reflections, intersymbol interference (ISI), ground-bounce, timing jitter, and substrate noise degrade the signal integrity of real links. The first three of these, crosstalk, reflections, and ISI, are linear processes and can be mitigated with equalization filters [3, 5, 13, 15]. In this paper, we present novel techniques for the optimal synthesis of such filters. Equalization has also been proposed for jitter mitigation [2].

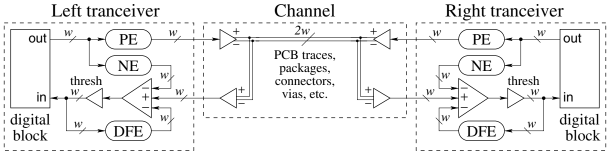

Figure 1: A link with equalization filters

High speed links often use pre-equalization by the transmitter, decision feedback equalization by the receiver, and near-end cross-talk cancellation for bidirectional links. Figure 1 shows a w-bit bidirectional link with differential signaling and these three forms of equalization. Each transceiver consists of a digital block that transmits and receives data, a pre-equalizer (PE), an equalizer for near-end crosstalk cancellation (NE), and a decision feedback equalizer (DFE). The channel includes the differential drivers and receivers as well as the packages, PC board buses, and connectors that provide the connections between the communicating chips. Each of the three equalizers brings its own strengths and weaknesses to the system, and designing an optimal channel requires managing these trade-offs effectively. For example, the PE filter can improve far-end signal integrity by boosting the high-frequency components of the transmitted signal; however, this exacerbates near-end crosstalk and places greater demands on the NE filter. The DFE can correct for many of the same effects of crosstalk and ISI as the pre-equalizer, without incurring the PEs downsides of increased power consumption and near-end crosstalk. However, the DFE is sensitive to errors in the received data stream, and creates a tight feedback cycle that isn't present in the other filters. Thus, practical designs make use of both pre-equalization and DFE.

While there has been intensive research in high-speed links in the past eight years, there has been little CAD tool support. In practice, designers rely on extensive simulations and trial-and-error prototyping. This paper presents a novel approach to jointly optimizing pre-equalizers, decision feed-

DAC 2005, June 13–17, 2005, Anaheim, California, USA.

back equalizers, and filters for near-end crosstalk cancellation. We show that optimizing an eye-mask for these three filters is a linear programming problem, and present efficient methods for solving these linear programs. In earlier work [10], we presented crosstalk cancelling pre-equalizers for single-ended, unidirectional links with relatively simple channel models. The present work addresses a comprehensive set of practical issues including the incorporation of differential and bidirectional channels, and modeling of realistic packaging and connector parasitics. We present simulation results for on-board links as well as backplane links connecting two daughter cards.

We describe our mathematical programming formulation for optimizing signal integrity in section 2. Section 3 presents an efficient implementation of the linear program solver for our problem. In section 4, we demonstrate our approach by examining the symbol rates at which various eye-heights can be achieved for several link configurations with channel models that include packaging and connector parasitics.

2. OPTIMAL FILTER SYNTHESIS

This section shows how to jointly synthesize optimal PE, DFE and NE filters to maximize eye height, optimize eye masks and minimize near-end crosstalk for simultaneous bidirectional signaling over differential pairs. We first state assumptions that simplify the presentation. We then describe eye-masks, a common measure of signal integrity, and show how they can be parameterized to obtain objective functions for optimization. Finally, we formulate the joint filter synthesis problem for pre-equalization, decision feedback equalization, and near-end crosstalk cancellation.

2.1 Simplifying Assumptions

We use the following assumptions:

1. The response of the channel is linear and can be approximated accurately by a finite length impulse response.

2. The channel is symmetric and transceivers on both ends have the same characteristics.

3. Each differential pair is used to convey binary (i.e. two-level) data. The input high and low levels are $+1$ and -1. Likewise, the output target levels are $+1$ and -1.

4. The high and low portions of the eye-mask are symmetric, and the same eye mask is used for every differential pair.

The first pertains to the physical interconnect and holds for most off-chip practical applications. The last three simplify the presentation. Extending the methods presented here to asymmetric channels, other signaling levels, single-ended links, other eye-mask shapes, etc., is straightforward.

2.2 Eye Masks

We formulate filter synthesis using parameterized eye masks. Figure 2 shows such a mask. The mask specifies constraints on the possible waveforms of a signal during each symbol period by excluding regions of undershoot and overshoot. Typically, eye-masks are given as polygons. We parameterize these polygons by specifying the vertices using the variable η. In figure 2, we specify constraints on overshoot

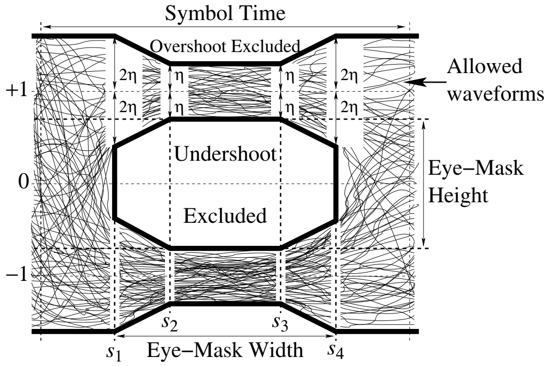

Figure 2: A Parameterized Eye Mask

and undershoot at four different sample times. More generally, we represent an eye mask as a set L of undershoot constraints and U of overshoot constraints. In particular, (s, α) in L has the interpretation that for all input patterns where the transmitted value for the wire and sample period of interest is $+1$, the received value must be at least $1-\alpha\eta$ at sample time s. We interpret the overshoot constraints from U in the corresponding manner. Minimizing η optimizes the eye mask.

2.3 Eye Mask Optimization

Eye-masks specify constraints that must hold for all possible input patterns; in other words, eye-masks specify constraints on the *worst-case* waveforms. This is a l_∞ optimization problem that can be solved by linear programming. In our earlier work [10, 11], both simulation results and measurements on a physical test bench have shown clear advantages of our l_∞ approach over traditional least-squares optimization for obtaining optimal eye masks. This earlier work only considered pre-equalizing filters for minimizing far-end crosstalk and inter-symbol interference. We now show how to formulate the joint optimization of pre-equalizing filters, decision feedback equalizers, and near-end crosstalk cancelling filters and show that linear programming can be used to find the optimal filter coefficients. In section 3, we present efficient techniques for solving these linear programs.

Consider a link as shown in figure 1. Each client transmits w_{link} symbols per symbol period. We call w_{link} the width of the link and say that the link has w_{link} lines. Using differential signaling, the bus between the transceivers has $2w_{\text{link}}$ wires. In the following we refer to each differential pair as simply a “pair;” thus, the bus has w_{link} pairs.

In practice, the tap rates for equalization filters are a small multiple of the symbol rate to allow the filters to compensate for high frequency losses of the bus. Furthermore, we perform our analysis with a sample rate greater than the filter tap rate to avoid quantization effects that can lead to severe overshoot. Let r_{tap} be the number of sample times per tap time, and let r_{sym} be the number of sample times per symbol time. In the examples in this paper, we use $r_{\text{tap}} = 4$ and $r_{\text{sym}} = 8$ (i.e. two taps per symbol). We define $\mathbb{M}_{\text{sym}}$ as the set of integer multiples of r_{sym}.

Consider the signal received at y, the input of the threshold circuit in the right transceiver. We exploit the linearity of the channel and filters to break the signal received at

y into several independent components. Let $\mathsf{y}(j,s)$ be the value received on line j at time s. We define:

y_{fe}: (the far end response) the data transmitted by the left-transceiver convolved with the impulse responses of the PE filter and the far-end response of the channel.

$\mathsf{y}_{\mathrm{dfe}}$: (the DFE filter output) The data received by the right-transceiver convolved with the impulse responses of the DFE filter.

$\mathsf{y}_{\mathrm{next}}$: (near end crosstalk) The data transmitted by the right-transceiver convolved with the near-end response of the channel.

y_{ne}: (the NE filter output) The data transmitted by the right-transceiver convolved with the impulse response of the NE filter.

Thus, $\mathsf{y} = \mathsf{y}_{\mathrm{fe}} + \mathsf{y}_{\mathrm{dfe}} + \mathsf{y}_{\mathrm{next}} + \mathsf{y}_{\mathrm{ne}}$.

Let f_{pe}, f_{dfe}, and f_{ne} be column vectors of the filter coefficients for the PE, DFE, and NE filters respectively. Let $\mathsf{y}_{\mathrm{fe}}(j,s)$ be the component y_{fe} for line j at time s. Consider a scenario where a value of $+1$ is output by the left-transceiver on line i for one symbol period starting at time 0 and a value of 0 is sent at all other times and on all other lines. We define a column vector $g_{\mathrm{fe}}(i,j,s)$ such that for this scenario the response at y on line j at time s is given by $g_{\mathrm{fe}}(i,j,s)'\,f_{\mathrm{pe}}$. We call g_{fe} the far-end bit response of the channel. Let $v(i,s) \in \pm 1$ be the data value sent on line i at time s. By the assumption that the channel is linear, we have

$$\mathsf{y}_{\mathrm{fe}}(j,s) \;=\; \sum_{i=1}^{w_{\mathrm{link}}} \sum_{k \in \mathbb{M}_{\mathrm{sym}}} v(i,k) g_{\mathrm{fe}}(i,j,s-k)'\,f_{\mathrm{pe}} \qquad (1)$$

We define g_{dfe}, g_{next}, and g_{ne} in the equivalent manner. Note that g_{fe} and g_{next} depend on the response of the bus whereas g_{dfe} and g_{ne} do not. The convolutions for calculating y_{fe} and $\mathsf{y}_{\mathrm{dfe}}$ use data from the left-transceiver, whereas the convolutions for $\mathsf{y}_{\mathrm{next}}$ and y_{ne} use data from the right.

Let δ_0 be the target delay for the channel. Typically, we choose δ_0 to be slightly larger than the LC delay of the channel. Thus $v(i, s-\delta_0)$ is the desired value for $\mathsf{y}(i,s)$. We focus on the case where $v(i,s-\delta_0) = +1$, and express the response of the channel as an "undisturbed" component plus a sum of "disturbances." The "undisturbed" response, $u(j,s)$ is

$$u(j,s) \;=\; g_{\mathrm{fe}}(j,j,\delta_0)'\,f_{\mathrm{pe}} \qquad (2)$$

We first consider disturbances arising from other bits sent by the left-transceiver including the contributions of the DFE filter. For simplicity, we assume that the link is error-free; thus, the received data stream matches the transmitted data stream. We do not address error handling here due to space limitations. We find the largest possible disturbance by choosing the signs of the other data bits to result in positive disturbances. Accordingly, the maximum far-end response, $\mathsf{y}_{\mathrm{far,max}}$, and maximum far-end disturbance, d_{fe}, are

$$\begin{aligned} \mathsf{y}_{\mathrm{far,max}}(j,s) \;&=\; \sum_{i=1}^{w_{\mathrm{link}}} \sum_{k \in \mathbb{M}_{\mathrm{sym}}} \\ &\left| g_{\mathrm{fe}}(i,j,s+k)'\,f_{\mathrm{pe}} + g_{\mathrm{dfe}}(i,j,s+k)'\,f_{\mathrm{dfe}} \right| \qquad (3) \\ d_{\mathrm{fe}}(j,s) \;&=\; \mathsf{y}_{\mathrm{far,max}}(j,s) - u(j,s) \end{aligned}$$

We now consider near-end interference. The contribution of $\mathsf{y}_{\mathrm{next}}+\mathsf{y}_{\mathrm{ne}}$ is purely a disturbance. By reasoning equivalent to the far-end case, we get

$$\begin{aligned} d_{\mathrm{ne}}(j,s) \;&=\; \sum_{i=1}^{w_{\mathrm{link}}} \sum_{k \in \mathbb{M}_{\mathrm{sym}}} \qquad (4) \\ &\left| g_{\mathrm{next}}(i,j,s+k)'\,f_{\mathrm{pe}} + g_{\mathrm{ne}}(i,j,s+k)'\,f_{\mathrm{ne}} \right| \end{aligned}$$

Here we used the assumption that the transceivers and channel are symmetric by assuming that f_{pe} is the pre-equalizer for the left transceiver in equations 2 and 3 and for the right in equation 4. This assumption could be removed by using separate filter coefficient vectors for the left and right transceivers.

We consider the pessimistic case that the left and right transceivers have independent clocks; thus, there is no fixed alignment between the period of the near-end disturbance and the received data. Instead, we assume that the worst-case near-end disturbance could happen at any time during the sampling interval. We overload d_{ne} and write $d_{\mathrm{ne}}(j)$ for the worst case near-end disturbance on line j with

$$d_{\mathrm{ne}}(j) \;=\; \max\nolimits_{s \in [0 \ldots r_{\mathrm{sym}}]} d_{\mathrm{ne}}(j,s) \qquad (5)$$

Equations 2, 3 and 5 yield the mathematical program for unified optimization for PE, DFE and NE filters for bidirectional signaling:

$$\begin{aligned} &\min_{f_{\mathrm{pe}}, f_{\mathrm{dfe}}, f_{\mathrm{ne}}} \quad \eta \ \text{ s.t.} \\ &\qquad \forall (s,\alpha) \in L.\ \forall j \in [1 \ldots w_{\mathrm{link}}]. \\ &\qquad\quad 1 + d_{\mathrm{fe}}(j,s) - u(j,s) + d_{\mathrm{ne}}(j) \leq \alpha\eta \qquad (6) \\ &\quad \wedge \quad \forall (s,\alpha) \in U.\ \forall j \in [1 \ldots w_{\mathrm{link}}]. \\ &\qquad\quad d_{\mathrm{fe}}(j,s) + u(j,s) - 1 + d_{\mathrm{ne}}(j) \leq \alpha\eta \end{aligned}$$

We note that u, d_{fe}, and d_{ne} are all linear in the filter coefficients. Thus, this is a linear programming problem. The optimization problem for unidirectional signaling is the same with $d_{\mathrm{ne}}(j) = 0$.

3. IMPLEMENTING THE OPTIMIZATION ALGORITHM

Given a bus impulse response and an eye mask specification, we set up a linear programming problem according to the formulation presented in section 2. In addition, we note that in practice, filters have limited overdrive ability. Thus, in addition to the constraints from equation 6, we add constraints to limit the magnitude of the filter output on each wire at each tap time. We now write equation 6 as

$$\min_{f,d,\eta,e} \eta \ \text{ s.t.}$$
$$\begin{bmatrix} -I & 0 & G & 0 \\ 0 & -I & F & 0 \\ -I & 0 & -G & 0 \\ 0 & -I & -F & 0 \\ X & 0 & H & -\alpha \\ X & 0 & -H & -\alpha \\ 0 & M & 0 & 0 \end{bmatrix} * \begin{bmatrix} d \\ e \\ f \\ \eta \end{bmatrix} \;\leq\; \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \\ \gamma \end{bmatrix} \qquad (7)$$

The rows with the G matrices compute d, the absolute values of the disturbances, the rows with H and $-H$ compute the maximum undershoot and overshoot respectively: H computes the undisturbed response; X computes the total of the disturbance terms; and α is a column vector of scaling terms for each measurement point of the eye mask.

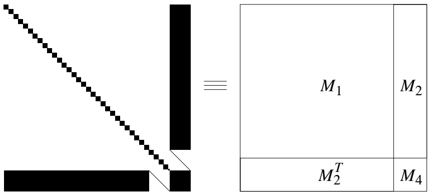

Figure 3: Sparsity pattern of $A^T\Lambda^2A$.

The rows with F and $-F$ compute e, the absolute values of the filter outputs. Finally, the row with M computes the maximum magnitude output for each wire at each sample time where γ is the output limit.

To implement the filter synthesis routines we implemented a customized version of Mehrotra's interior-point, predictor-corrector algorithm [9] using Matlab [14]. The size of the LPs presents the greatest challenge. The examples presented in section 4 have over 10,000 variables and over 15,000 constraints for a bidirectional bus with four differential pairs. With sixteen pairs, the LP has over 100,000 variables and 200,000 constraints. The number of variables and constraints of the linear programs grows quadratically with the number of lines in the bus and linearly with the number of filter coefficients. The large number of LP variables is primarily due to the number of disturbance terms d, which grows quadratically with the width of the bus. To implement a practical filter synthesis tool, we exploited the sparsity and structure of our particular constraint matrix.

Mehrotra's algorithm repeatedly solves for x in linear systems of the form:

$$A^T\Lambda^2Ax \quad = \quad y \tag{8}$$

where A is the constraint matrix from equation 7, and Λ is a diagonal matrix whose elements are updated with each iteration of the algorithm. Equation 8 is called the "normal equation" for the LP. Our particular LP formulation produces the sparsity pattern shown in figure 3 for the normal equation. We solve the normal equation by forming its Schur complement.

$$S = M_4 - M_2^T M_1^{-1} M_2 \tag{9}$$

where blocks M_1, M_2, and M_4 are as shown in figure 3. The size of the Schur complement S grows linearly with the width of the bus; hence it is a much smaller than the original normal equation. We solve this system by Cholesky decomposition using the Matlab function chol().

For our LPs, each block of M_1 is a rank-1 update of a diagonal matrix. This allows us to invert M_1 very efficiently based on the Sherman-Morrison formula [6], without forming M_1 or its inverse explicitly. Furthermore, the M_2 matrix is sufficiently large that computing $M_2^T M_1^{-1} M_2$ directly is very slow. Instead, we divide the A matrix into blocks that naturally arise from the formulation of the linear programming problem and rewrite $M_2^T M_1^{-1} M_2$ according to these blocks. We then re-order the multiplications to avoid producing large intermediate matrices. These techniques greatly speed up the LP solver. For example, for a LP with $\sim$ 50000 variables and $\sim$ 100000 constraints, it takes 4 seconds to form its schur complement on a 900MHz, UltraSparc III processor, while the brute force approach with Matlab built-in sparse matrix computation takes 1506 seconds.

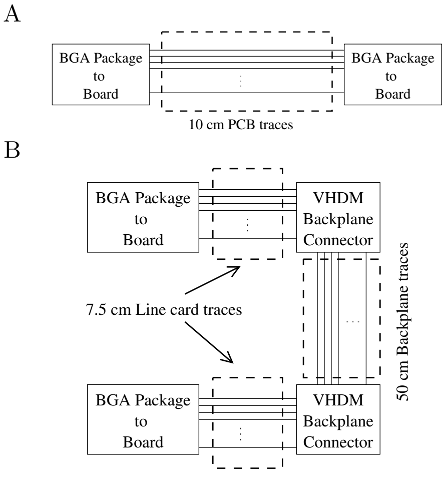

Figure 4: Off-chip link models.

4. EVALUATION

This section reports results using the filter synthesis procedure described in the previous sections for realistic channels. Our goal is to evaluate the synthesis procedure and explore the trade-offs and limitations for high-speed off-chip buses.

4.1 Link Models

We considered two basic configurations as shown in figure 4: a 10 cm point-to-point link between two chips on the same PC board; and a link across a 50 cm backplane. For the inter-board link, we used a model provided by Teradyne for their eight-row, VHDM-HSD connector. We modeled each DAC output as linearly slewing from its old value to its new value over the tap-period. This reflects the limited slew-rate of real DACs and avoids introducing unwanted high-frequency energy into the channel.

All PC board buses considered here use 6 mil traces with 8 mil spacing in 0.5 oz copper with ground planes on each side of the signal plane based on a design from Rambus [8]. We used the 2D field solver of HSPICE to obtain an electrical model for this bus. The bus has a 100Ω nominal impedance. We assumed a manufacturing tolerance of ±10% for the actual impedance, and extracted models with worst-case mismatches. All differential pairs are terminated with 100Ω resistors to ground.

For our initial experiments, we used the ball grid array model from Dally and Poulton [4, p. 39]. Our initial experiments showed that the chip package was the critical bottleneck for both configurations due to the 5 nH of chip-to-package and another 5 nH of board-to-chip inductances. After consulting with designers of high-speed links in industry [7], we created a model where these inductances were reduced to 0.5 nH and reduced the package capacitances by a factor of 3.

While we believe that these models provide a fairly realistic model for high-speed links, it is important to note that we did not include ground and Vdd bounce, timing jitter, or PC board vias. Thus, with a real, physical link, the actual eye-heights would be lower than the ones we report, and the effects of vias will be especially pronounced at high data

	Inter-board		Intra-board	
	bi	uni	bi	uni
P1, w/o filters	0.5	1.1	1.2	3.5
P1, w. filters	1.7	3.0	2.5	6.0
P2, w/o filters	0.5	1.4	2.5	10.0
P2, w. filters	4.0	5.0	12.5	> 25.0

bi = bidirectional link; uni = unidirectional link; P1 = original, Dally and Poulton BGA model; P2 = $L/10$, $C/3$ BGA model.

Table 1: Maximum Bit-Rates (in Gb/s/pair) for 50% Eye-Height

eye-height	Inter-board		Intra-board	
	bi	uni	bi	uni
50% (2-PAM)	4.0	4.0	12.5	> 25.0
83% (4-PAM)	< 1.0	3.3	2.5	15.0
93% (8-PAM)	< 1.0	< 1.0	< 1.0	8.0

Table 2: Maximum Bit-Rates (in Gb/s/pair) for Various Eye-Heights

rates. However, by using the same models for the channels with and without filters, we believe that our results accurately indicate the merits of our equalizing filters and our unified optimization framework.

4.2 Results

Table 1 shows the bit-rates that can be achieved for a variety of link configurations with four differential pairs. All filters were optimized using the parameterized eye-mask from figure 2. For all of the filters, the tap rate is twice the symbol rate. The pre-equalization filters compute their outputs for each differential pair based on the data input for that pair and for each of its immediate neighbours. For inter-board links, the pre-equalizer has 6 taps (i.e. 3 symbol times), and for intra-board links it has four taps.

The decision feedback equalizer only considers the current line. Like the pre-equalizer, the DFE filter has 6 taps for the inter-board links and 4 taps for the intra-board links. We assume a latency of two symbol periods in the DFE; thus, it can only correct for ISI and reflections that occur after that delay.

Reflections are particularly severe for the bidirectional links, and we designed the near-end (NE) filters with multiple segments corresponding to the delays of the principal reflections. For the inter-board links, the NE filters have four segments. The first two segments have six taps and consider nearest neighbours. The last two have six taps but only consider the line itself. The NE filters for the intra-board links have two segments. Both segments have four taps and consider nearest neighbours. For the inter-board NE filters the last two segments can ignore neighbours because there is sufficient high-frequency attenuation in the backplane to render such coupling negligible.

In our initial designs, near-end crosstalk severely limited the performance of the bidirectional links. This is because the peak of the near-end interference can occur anywhere in the received eye. To mitigate this, we introduced an integrating receiver [12] at the input of the thresholding element. We modeled it with a simple convolution over four consecutive sample points. This allowed bandwidth improvements of roughly 10% for most of the bidirectional scenarios and a much larger improvement of 60% for the intra-board link with the reduced inductance package. In all cases, the integrator reduced the performance of the unidirectional links. Thus, we report results for bidirectional links with an integrating receiver and with a simple, thresholding receiver for the unidirectional links.

From Table 1, we see that the equalizing filters double the channel bandwidth in all cases with even greater gains when using the low inductance package. Due to near-end cross-talk, bidirectional signaling always has a lower one-way bandwidth than unidirectional signaling. However, bidirectional signaling achieves greater total bandwidth for the inter-board links.

Table 2 explores the trade-off between symbol-rate and eye-height. All results are with the reduced inductance package. If a channel with two-level signaling achieves an eye-height of $(100 - E/(N-1))\%$, then a channel with N-level signaling and the same filters can achieve an eye-height of $(100 - E)\%$. Thus, 83% eye-height for two-level signaling provides 50% for four-level, and a 93% eye-height for two-level signaling provides 50% for eight-level. Because we are ignoring ground bounce, and clock-jitter, these are optimistic estimates, especially for multi-level signalling. The maximum output of the filter is three times the target value. Greater signal integrity can be achieved in the unidirectional case with greater overdrive, but we regarded the cost in power and the need for greater DAC resolution to preclude such designs.

Table 2 illustrates how a designer can use our synthesis procedure to explore design trade-offs. For example, it shows that for the channels that we considered, multi-level signaling is never advantageous for a bidirectional link – the total bandwidth of the link is much less than can be achieved with simple, two-level signaling. For unidirectional links, multi-level signaling is a more viable alternative. A designer would have to trade-off the advantages of a lower symbol rate against the increased complexity in the receiver circuitry. We note that because our models do not include PC board vias, the extremely high data rates obtained for intra-board, unidirectional links should be taken with many grains of salt. Even in this case, we see the advantage of automatic filter synthesis – it allows us to quickly identify scenarios where more detailed modeling of the channel is needed.

We also tried separate synthesis of the filters. We first synthesized an optimal pre-equalizer assuming no DFE or NE filters, and then synthesized the other two filters including the pre-equalizer in the channel. The unified approach performed much better in the bidirectional case when the filter output magnitude was unconstrained. This is because the pre-equalizer generated by separate synthesis would drive very high slew-rate transitions into the channel causing severe near-end crosstalk. Reducing the maximum filter output magnitude mitigates this effect. However, it also reduces the performance of the pre-equalizer. Hence manual adjustment and iterations are needed to achieve good performance for separate synthesis. The unified optimization automatically finds the best balance between the filters and achieves eye-heights 5-10% greater than those for separate optimization. Thus, in addition to producing better filters, the designer can explore a simpler design space without losing optimality.

Our filter design times range from less than one minute for a unidirectional link with four differential pairs, to about two minutes for a bidirectional link with four pairs and about forty minutes for a bidirectional link with 16 pairs. These times are for a 900MHz, UltraSparc III processor. The time is roughly equally divided between setting up the linear program and solving it.

5. GENERALIZATIONS

In section 2, we made several assumptions to simplify the presentation. We now revisit some of these.

The channel is symmetric. As noted following equation 4, this assumption can easily be removed. Disturbances from the two transceivers would have to be considered separately. This roughly doubles the number of variables and constraints in the LP, but the structure remains the same.

The eye mask is symmetric and the same for every differential pair. Our methods work with asymmetric eye-masks and with different eye-masks for each link of the channel. These changes would be reflected in the objective function. The size of the resulting LP would be unchanged.

The link is error-free. In practice, errors may occur, and following an error, the DFE filter may corrupt signal integrity rather than improve it. In practice, the DFE filter output is typically small enough to prevent cascading errors from becoming a serious issue. Consider a bus with an error-rate of 10^{-15}/symbol when the data input to the DFE is correct. For the sake of an example, we'll assume that the error rate increases to 10^{-6} when the DFE is processing an erroneous bit. Because the higher error rate only lasts for a few symbols, the impact on the overall error rate is negligible. More generally, our linear programming formulation provides a natural opportunity for restricting the magnitude of the DFE output to ensure that cascading errors are not a problem.

The filter synthesis problem is a linear program. As discussed in section 2.2, our linear programming formulation corresponds directly to eye-masks. While we parameterized the eye mask using a single scaling factor, η, we note that the linear programming framework provides a great degree of flexibility in formulating the objective function. Nevertheless, there are some natural aspects of filter design that do not fit into our linear programming framework. In particular, optimizing cascaded filters (such as the integrators discussed in section 4) and minimizing the power output of the transmitter require quadratic formulations. These are problems for future work.

6. CONCLUSIONS

Chip clock rates continue to grow at a much faster rate than improvements in off-chip interconnect. To bridge the gap, designers are using increasingly sophisticated on-chip equalization filters. We have presented a unified approach for synthesizing optimal filters for the three most common forms of equalization: transmitter pre-equalization, decision-feedback equalization, and near-end crosstalk cancellation. We formulate the objective function using parameterized eye-masks that give the designer great flexibility for specifying trade-offs between eye-height, eye-width, and other details of the eye-shape. We have shown that the resulting optimization problems can be solved using linear programming. The linear programs are large, but can be solved efficiently using sparse matrix techniques that exploit the specific structure of the constraint matrices.

To illustrate the use of our methods, we examined the design of bidirectional and unidirectional links for both intra-board and cross-backplane communication. Our models included chip packaging parasitics, impedance mismatches, dielectric and skin-effect losses, and connector parasitics. Automatic synthesis of optimal filters allows early identification of bandwidth bottlenecks and rapid evaluation of design trade-offs such as the use of multi-level signaling and integrating receivers.

7. REFERENCES

[1] International Technology Roadmap for Semiconductors. http://public.itrs.net/Files/2003ITRS/Test2003.pdf,2003.

[2] G. Balamurugan and N. Shanbhag. Modeling and mitigation of jitter in multiGbps source-synchronous I/O links. *21st Int'l. Conf. on Computer Design*, pages 254–260, 2003.

[3] W. Dally and J. Poulton. Transmitter equalization for 4-GBPs signaling. *IEEE Micro*, 1:48–56, 1997.

[4] W. Dally and J. Poulton. Digital Systems Engineering. Cambridge University Press, 1998.

[5] A. Fiedler, R. Mactaggart, et al. A 1.0625Gbps transceiver with 2x-oversampling and transmit signal pre-emphasis. In *Proc. of ISSCC97*, pages 238–239, 1997.

[6] G.H. Golub and C.F. Van Loan. Matrix Computations, 7th ed. *Johns Hopkins*, page 51, 1996.

[7] Ron Ho. Personal Communication, 2004

[8] R. Kollipara, G.J. Yeh, et al. Design, Modeling and Characterization of High Speed Backplane Interconnects. *High-Performance System Design Conf.*, 2003.

[9] J. Nocedal and S. Wright. *Numerical Optimization*, pages 395–417, Springer Press, 1999.

[10] J. Ren and M. Greenstreet. Crosstalk Cancellation for Realistic PCB Buses. In *Proc. of the 14th Int. Workshop on Power and Timing Modeling, Optimization and Synthesis (PATMOS)* , Springer – LNCS, 2004.

[11] J. Ren and M. Greenstreet. A Signal Integrity Test Bed for PCB Buses. In *Proc. of the Int'l. Conf. Computer Design*, pages 2004.

[12] S. Sidiropoulos and M. Horowitz. A 700-Mb/s/pin CMOS signaling interface using current integrating receivers. *IEEE J. Solid State Circuits*, 32(5):681-690, May 1997.

[13] V. Stojanovic, G. Ginis, and M. Horowitz. Transmit pre-emphasis for high-speed time-division-multiplexed serial-link transceiver. *IEEE Trans. on Communications*, 38:551–558, 2001.

[14] The Mathworks Inc. http://www.mathworks.com

[15] 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*, 38(12):2121-2130, Dec. 2003.