Equalized Interconnects for On-Chip Networks: Modeling and Optimization Framework

Byungsub Kim and Vladimir Stojanović
Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA02139, USA
{byungsub,vlada}@mit.edu.

Abstract—**This paper presents a modeling framework for fast design space exploration and optimization of equalized on-chip interconnects. The exploration is enabled by cross-layer modeling that connects the transistor and wire parameters to link performance, equalization coefficients, and architecture-friendly metrics (delay, energy-per-bit, and throughput density). Appropriate models are derived to speed-up the search by more than two orders of magnitude and make a million point design space searchable in less than two hours on a standard machine. With this approach we are able to find the best link design for target throughput, power and area constraints, thus enabling the architectural optimization of energy-efficient on-chip networks. For the same latency and throughput density, equalized interconnects optimized using the new methodology have up to 10x better energy-efficiency than optimized repeater interconnects.**

I. INTRODUCTION

Poor scaling of global on-chip wires [1] has alerted many researchers and industry teams in the past several years to work on changing the on-chip communication methods and architectures of future computing platforms. Although modular, multi-core processors [2, 3] and Systems-On-a-Chip have been designed to alleviate the wire scaling problem, these solutions rely heavily on the ability to design efficient and scalable on-chip networks. This scalability assumes efficient use of power and area resources while maximizing the throughput and minimizing the communication latency. With global power and area constraints, a more area-and-energy-efficient interconnect leaves more resources for computation, improving the functionality of the chip.

On-chip networks today are exclusively designed using wires with repeaters to overcome the poor scaling of wire throughput with length [2,3]. Significant effort has already been spent to include this repeated interconnect in the chip design and verification flow [4,5]. Repeaters, however, burn extra power and introduce a fundamental trade-off between wire throughput and latency. To improve energy-efficiency, low-swing on-chip buses have been explored [6], but they provide limited data rate scaling without signal conditioning. To improve the wire throughput, recent efforts have thus focused on using signal conditioning or non-linear repeaters [7-9]. However, so far very little has been done on designing interconnects with advanced signaling techniques to address timing, area and energy constraints *simultaneously*.

Analysis in this paper shows that equalized interconnects can provide both high energy-efficiency and high-throughput at the same time, while not compromising the latency of the link. The key to getting these benefits from equalized interconnect is to jointly optimize the circuit, wire and equalization parameters. This requires appropriate cross-layer modeling and design methodology.

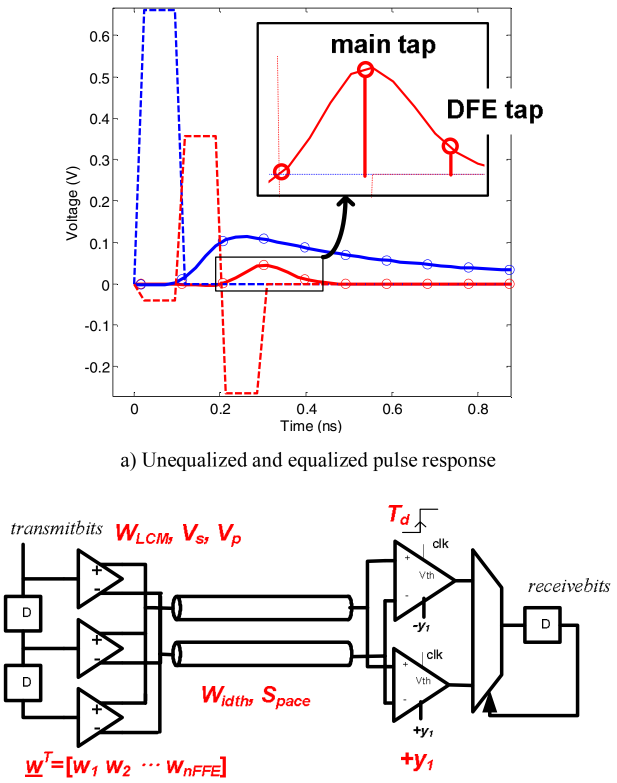
In this paper, we presents a framework for equalized interconnect optimization, by laying out the cross-layer design methodology and developing a tool for fast design space exploration. We first create the appropriate physical wire and circuit models, and then connect them to the channel model, to equalization algorithms and finally to the power and performance models. By deriving the closed form expression for optimal link delay and choosing the appropriate equalization algorithm, we are able to speed up the design space search by two orders of magnitude, making the joint circuit/wire optimization practical. Since utilization of power and metallization resources are the best indicators of the interconnect efficiency, in a way similar to the repeater bus optimization in [5], our equalized bus optimization framework uses energy-per-bit (power/data rate) and throughput density (data rate/interconnect pitch) as the metrics suitable for link abstraction at the on-chip network level.

II. EQUALIZED LINK DESIGN SPACE

A global on-chip wire acts as a low-pass filter, spreading and attenuating short transmitted pulses, Fig. 1a. The long tail of the received pulse causes intersymbol interference (ISI) to adjacent bits. A finite-impulse response (FIR) filter at the transmitter, Fig. 1b, can remove the long ISI tail, enabling higher data rate, provided that the receiver has enough precision to recover bits from signals with small amplitude. On-chip wires have a desirable property that this particular type of RC-dominated tail can be removed effectively with a small number of FIR taps (ideally two). However, a transmit

NEC fund, IBM faculty award.

1-4244-1382-6/07/$25.00 ©2007 IEEE



a) Unequalized and equalized pulse response

b) FFE+DFE link architecture and design parameters: W_{LCM}, V_s, V_p, $\underline{w}^T$, W_{idth}, S_{pace}, y_1,T_d

Figure 1. Equalization effects and link architecture

FIR filter (feed-forward equalizer – FFE) cannot improve the signal amplitude since transmit circuits have limited voltage swing. To improve the amplitude of the received signal we use a decision-feedback equalizer (DFE) filter to remove the interference based on the knowledge of previously received bits.

This equalized link architecture is very similar to the state-of-the-art off-chip high-speed links [10], where the first DFE tap is usually predictive due to speed constraints. The important difference in on-chip links is that good ISI compensation can be achieved with relatively short equalizer filters and transmitter impedance is not necessarily fixed to 50 Ω. The impedance should be tuned jointly with wire parameters to achieve the best performance metrics. When properly used, these additional degrees of freedom improve both the data rate density and energy-efficiency of equalized on-chip interconnects.

Fig. 1b shows the link architecture and parameters that should all be jointly optimized to achieve the optimal performance metrics (circuit parameters: driver size W_{LCM}, driver and pre-driver's supply voltages V_s, V_p; wire width and spacing W_{idth}, S_{pace}; and link parameters: equalization coefficients of transmitter and receiver $\underline{w}^T$, y_1 and the optimal sampling delay T_d). All these parameters are jointly optimized to minimize energy-per-bit cost for given architectural targets (data rate density and wire length), and link robustness (eye opening). Since this design space is huge (millions of data points for design parameters' ranges and resolutions of interest), we need a fast method to explore it and estimate the efficiency of this link architecture.

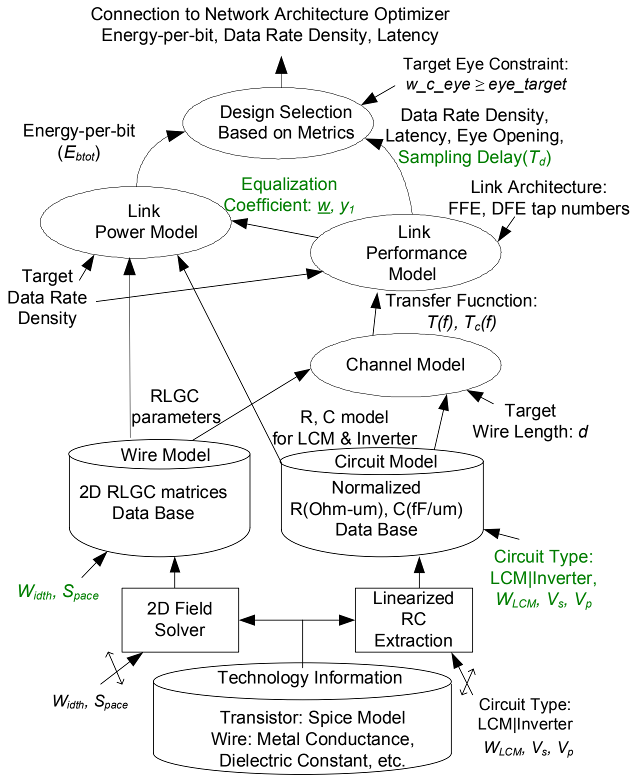


Figure 2. Tool flow

III. Fast Design Space Exploration

The cross-layer design space exploration and co-optimization is a complicated problem due to the metrics' non-linear (and sometimes non-closed form) dependencies on the underlying parameters. In this tool, we export simplified and abstracted low level models up to the link design level, and optimize the link parameters in terms of low level wire and circuit parameters. Fig 2 summarizes the overall tool flow.

At the physical level, the tool creates the RLGC database for different wire dimensions, and extracts normalized linear RC parameters for driver and pre-driver circuits.

The wire parameters such as wire width, space, conductivity, dielectric constant, etc., are fed into a 2D field solver to generate multi-port RLGC transmission line parameters for the wire models. The RLGC database parameterized by wire width and space is created for each metal layer.

Circuit parameters such as transistor spice model, driver and pre-driver supply voltages, are fed into the linearized RC model extractor for each circuit type: for example a Low Common-Mode driver (LCM) [6] and an inverter pre-driver. The extracted RC parameters are normalized by the transistor width and saved into a database.

At the electrical layer, the RLGC wire parameters together with the driver's linear RC model are combined to make the channel model. The formula of the through and crosstalk transfer functions *T*(*f*) and *T*$_c$(*f*) can be described with the RLGC matrices and the wire target length as well as the driver's RC model.

Different equalization algorithms can then be applied to produce data rate performance metrics and equalization coefficients needed for the link power model. With parameterized power and performance models, the tool can perform efficient design space search at the top level and find the optimal low level parameters. In the following sections we will describe each of the modeling levels in detail and outline the speed-ups that we achieved through adequate modeling.

A. Link Performance Model

The performance estimation is the most time consuming part of the flow due to numerical calculations to extract the performance and equalizer coefficients from a variety of different channels (for various wire and circuit sizes).

Appropriate performance model speeds-up the two major bottlenecks (finding optimal sampling time and equalization coefficients) by at least an order of magnitude each – resulting in more than two orders of magnitude faster design exploration. We derive the closed-form solution for near-optimal sampling time in terms of a channel's frequency response and also show that least-mean square based equalization yields similar results (albeit much faster) than a conservative eye maximization algorithm.

1) Optimal sampling delay

The received eye is affected by the channel attenuation, signal ISI, and crosstalk. In a well equalized channel, the optimal sampling time maximizes received signal amplitude, assuming that crosstalk and residual ISI distributions are relatively uniform across the bit time. For n_{FFE}-tap FFE transmitter with equalization coefficient vector $\underline{w}$, the frequency spectrum of the received signal *Y*(*f*) is

$$Y(f) = sinc(f)\,T(f)\,L(f)\,\underline{w}$$
$$L(f) = \begin{bmatrix} 1 & e^{-j2\pi f T_s} & \ldots & e^{-j2\pi f T_s (n_{FFE}-1)} \end{bmatrix} \tag{1}$$

where *sinc*(*f*), *T*(*f*), and T_s are the sinc function, transfer function of the channel, and the symbol time, respectively. *L*(*f*)$\underline{w}$ represents the FFE's transfer function. If we sample the received pulse response *y*(*t*) after latency time T_d, then the frequency response of the sampled sequence $Y_{ds}(f)$ can be approximated as

$$Y_{ds}(f) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} Y(f-2kf_N)e^{-j2\pi 2kf_N T_d}$$
$$\approx \frac{1}{T_s}\left(Y(f-2f_N)e^{-j2\pi 2f_N T_d} + Y(f) + Y(f+2f_N)e^{j2\pi 2f_N T_d}\right) \tag{2}$$

by assuming that the received signal spectrum *Y*(*f*) is band limited due to significant channel roll-off past 2x of the Nyquist frequency f_N. On the other hand, expressing $Y_{ds}(f)$ with the received signal samples y_k gives

$$Y_{ds}(f) = e^{-j2\pi f T_d}\sum_{k=-\infty}^{\infty} y_k e^{-j2\pi f k T_s} \approx e^{-j2\pi f T_d}\left(y_0 + y_1 e^{-j2\pi f T_s}\right) \tag{3}$$

where y_0 is the main tap of the sampled received pulse response, the value to be maximized. FFE will minimize all received pulse response samples except the main (y_0) and the first post-cursor (y_1), which is handled by the DFE. By setting the frequency *f* in (2) and (3) to zero and the Nyquist frequency (f_N), and ignoring frequency response above 2x of the Nyquist frequency, we obtain a set of equations to solve for y_0 and y_1:

$$Y_{ds}(0) \approx \frac{1}{T_s} Y_{ds}(0) \approx y_0 + y_1 \tag{4}$$

$$Y_{ds}(f_N) \approx \frac{1}{T_s}\left(Y(-f_N)e^{-j2\pi 2 f_N T_d} + Y(f_N)\right) \approx e^{-j2\pi f_N T_d}(y_0 - y_1). \tag{5}$$

Solving (4) and (5) for y_0 and y_1 allows us to approximate y_0 in terms of the received frequency spectrum at *f*=*0*, f_N, and the sampling delay T_d.

$$y_0 \approx \frac{Y(0)}{2T_s} + \frac{1}{T_s}\left|Y(f_N)\right|\cos\left(2\pi f_N T_d + \angle Y(f_N)\right) \tag{6}$$

Using (1) and (6), the T_d that maximizes the received signal is

$$T_d \approx -\frac{\angle Y(f_N)}{2\pi f_N} = \frac{T_s}{2} - \left(\frac{\angle\{L(f_N)\underline{w}\}}{2\pi f_N} + \frac{\angle T(f_N)}{2\pi f_N}\right) \tag{7}$$

where $T_s/2$ is time duration from the edge of the square pulse to its center, $-\frac{\angle\{L(f_N)\underline{w}\}}{2\pi f_N}$ is the delay of the FFE equalizer that is equal to (n_{main}-1)T_s where n_{main} is the index of the maximum valued equalization coefficient, and $-\frac{\angle T(f_N)}{2\pi f_N}$ is the channel delay at the Nyquist frequency.

2) Equalization Algorithms

We model a bundle of interconnects in a bus as a multi-port communication system with the additional requirement that all inputs have the same FFE coefficients (due to the symmetry of the bus structure and the same length of the wires). Including a crosstalk model into the equalization algorithm is critical since equalization affects not only the link performance but also the choice of the wire parameters like spacing and width, which in turn specify the crosstalk characteristics.

Our analysis shows that nearest neighbor crosstalk is by far the most dominant (due to attenuation and shielding). For a given sampling delay T_d, the channel transfer functions are converted into the time domain pulse responses and sampled at delay T_d with symbol-spaced sampling period T_s.

a) Maximizing the worst-case eye opneing (1_1-norm solution)

The preferred solution for conservative on-chip circuit design environment is to optimize the equalizer coefficients to satisfy the worst-case eye opening requirement. This problem can be formulated as a large linear program [11]. The equalized received sequence $\underline{y}$ and received crosstalk $\underline{c}$ can be written as

$$\begin{aligned} \underline{y} &= \mathbf{H}\underline{w} = [\underline{h}_1 \quad \underline{h}_2 \quad \dots \quad \underline{h}_n]\,\underline{w} \\ \underline{c} &= \mathbf{H}_\mathbf{c}\underline{w} = [\underline{h}_{c1} \quad \underline{h}_{c2} \quad \dots \quad \underline{h}_{cn}]\,\underline{w} \end{aligned} \tag{8}$$

where $\mathbf{H}$ and $\mathbf{H_c}$ are the Toeplitz matrices whose column vectors are the time shifted sampled pulse response vectors of $\underline{h}$ (through-channel) and $\underline{h}_c$ (crosstalk). The optimal T_d determines the main received signal sample, corresponding to the transmitted bit, y_{main} and the ISI term y_{isi}

$$\begin{aligned} y_{main} &= \underline{h}_{sig}{}^T\underline{w} \\ \underline{y}_{isi} &= \mathbf{H}_{\mathbf{isi}}\underline{w} \end{aligned} \tag{9}$$

where $\underline{h}_{sig}$ is the row vector of $\mathbf{H}$ at the index corresponding to T_d and $\mathbf{H_{isi}}$ is the intersymbol interference matrix derived from $\mathbf{H}$ by puncturing the row vectors that correspond to the main and post-cursor taps covered by DFE. The vector $\underline{y}_{isi}$ represents the ISI sequence generated by one bit transmission.

The worst case eye opening when independent and identically distributed (i.i.d.) random bit patterns are transmitted is

$$w\_c\_eye = \underline{h}_{sig}{}^T\underline{w} - \|\mathbf{H}_{\mathbf{isi}}\underline{w}\|_1 - \|\mathbf{H}_{\mathbf{c}}\underline{w}\|_1 \tag{10}$$

which gives the following formulation for optimal equalization coefficients $\underline{w}$

$$\begin{aligned} &maximize \quad w\_c\_eye \\ &subject\ to \quad \|\underline{w}\|_1 \le 1 \end{aligned} \tag{11}$$

where l_1-norm of $\underline{w}$ limits the maximum voltage swing of the FFE. By using the epigraph form and expanding the l_1-norm terms in (10) and (11) we get the linear program formulation

$$\begin{aligned} &maximize \quad eye \\ &subject\ to \quad eye - \underline{h}_{sig}^T\underline{w} + \underline{d}_j^T\mathbf{H}_{\mathbf{isi}}\underline{w} + \underline{d}_k^T\mathbf{H}_{\mathbf{c}}\underline{w} \le 0 \\ &\qquad\qquad\quad \underline{s}_n{}^T\underline{w} - 1 \le 0 \\ &\qquad\qquad\quad \forall\, j,k \mid \underline{d}_j, \underline{d}_k \in \{\pm 1\}^{length(\mathbf{H}_{\mathbf{isi}}\underline{w})} \\ &\qquad\qquad\quad \forall\, n \mid \underline{s}_n \in \{\pm 1\}^{length(\underline{w})} \end{aligned} \tag{12}$$

where the first $2^{length(\mathbf{H}_{isi}\ \underline{w})}$ constraints represent (10) and the next $2^{length(\underline{w})}$ constraints represent the l_1-norm of $\underline{w}.$ Since the number of constraints grows exponentially with the length of the pulse response, linear program in (12) can be rather slow, so we use the numerical gradient solver on original problem in (11) to directly compute the l_1-norms. Although it speeds up the computation, this method is still about an order of magnitude slower than the closed-form least-mean-square equalization (LMSE) algorithm described next.

b) LMSE (l_2-norm solution)

Although in general LMSE does not maximize the worst-case eye opening, we show in the results section that for this class of on-chip wire channels, the LMSE is very close to the worst-case eye opening solution. The LMSE finds the equalization coefficients that minimize the ISI and crosstalk energy (i.e. their l_2-norm) with closed form solution for equalization.

In the LMSE algorithm, we first set the main received signal to unity and then minimize the ISI and crosstalk energy from (8) and (9) using a method of Lagrange multipliers

$$y_{main} = \underline{h}_{sig}{}^T\underline{w} = 1 \tag{13}$$

$$E_N = \left|\underline{y}_{isi}\right|^2 + \left|\underline{c}\right|^2 = \underline{w}^T\left(\mathbf{H}_{\mathbf{isi}}^{\mathbf{T}}\mathbf{H}_{\mathbf{isi}} + \mathbf{H}_{\mathbf{c}}^{\mathbf{T}}\mathbf{H}_{\mathbf{c}}\right)\underline{w} \tag{14}$$

$$\underline{w}_{lmse} = \frac{\left(\mathbf{H}_{\mathbf{isi}}^{\mathbf{T}}\mathbf{H}_{\mathbf{isi}} + \mathbf{H}_{\mathbf{c}}^{\mathbf{T}}\mathbf{H}_{\mathbf{c}}\right)^{-1}\underline{h}_{sig}}{\underline{h}_{sig}{}^T\left(\mathbf{H}_{\mathbf{isi}}^{\mathbf{T}}\mathbf{H}_{\mathbf{isi}} + \mathbf{H}_{\mathbf{c}}^{\mathbf{T}}\mathbf{H}_{\mathbf{c}}\right)^{-1}\underline{h}_{sig}} \tag{15}$$

To satisfy the maximum voltage constraint $\|\underline{w}\|_1 \le 1$, $\underline{w}_{lmse}$ must be normalized

$$\underline{\tilde{w}}_{lmse} = \frac{\underline{w}_{lmse}}{\left\|\underline{w}_{lmse}\right\|_1}. \tag{16}$$

IV. Modeling Other Layers

In this section, the physical, channel and power models are described.

A. Physical Model

1) Circuit Model

Although there are many different types of on-chip wire drivers, in this modeling example we use the LCM driver, which was reported to be very power efficient [6]. Figure 3 shows a schematic and the equivalent linearized model of the LCM driver.

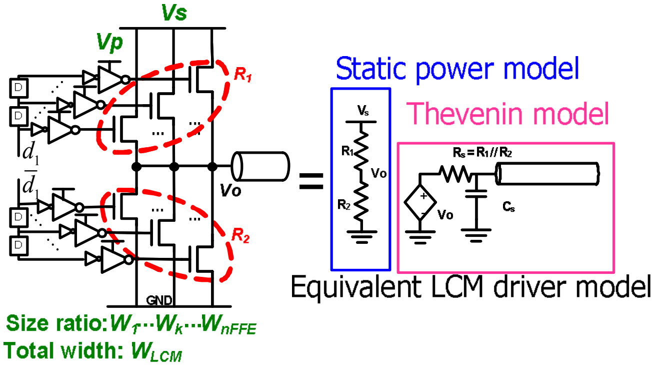


Figure 3. LCM driver and equivalent model

The gate voltage of each transistor is digitally controlled by an inverter-based pre-driver which has separate power supply voltage V_p. To keep the NMOS in triode region, typically driver's power supply V_s is chosen less than V_p, and V_p is set to V_{dd} in this paper.

An LCM driver consists of pull-up and pull-down pair segments. In each segment, the pull-up and pull-down NMOSs are sized for the same resistance, and the each segment corresponds to one of the FFE tap coefficients $\underline{w}^T=[w_1 .. w_{nFFE}]$ since the resistance of each segment is set to be proportional to the inverse of each tap coefficient w_i. The total LCM driver width (the sum of each segment's width) is parameterized by W_{LCM}.

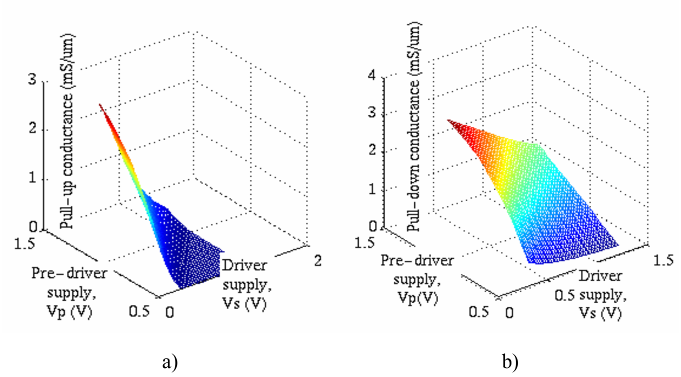
By turning on either pull-up or pull-down transistors in each segment, the LCM driver controls the total pull-up and pull-down resistance R_1 and R_2 while keeping the Thevenin resistance R_s constant. The Thevenin equivalent output voltage V_o is determined by R_1 and R_2's voltage division.

When different segments are driven with delayed data, this driver becomes an analog FIR filter [12], which can be used as an FFE. At the link model layer, our tool uses the linearized LCM model as a controlled voltage source with impedance R_s and parasitic capacitance C_s that captures the transient response of the LCM driver.

As shown in (17), R_s and C_s as well as the LCM driver's total input gate capacitance C_g can be expressed in terms of the total driver width w_{LCM} by using normalized technology-voltage-dependent constants: R_{LCM} (Ohm-um), C_{dLCM} (fF/um), and C_{gLCM} (fF/um).

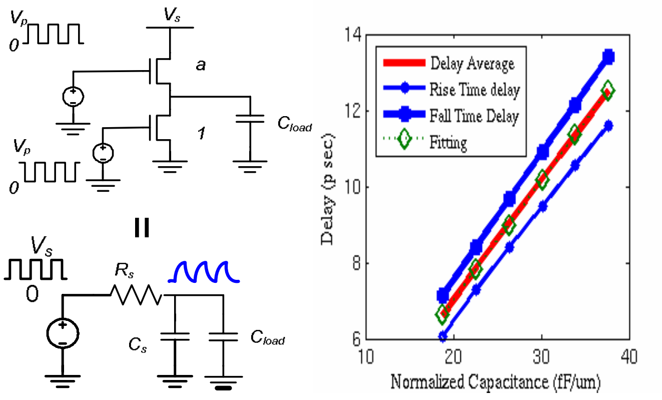
$$
\begin{aligned}
R_s &= R_{LCM} / w_{LCM} \\
C_s &= C_{dLCM}\, w_{LCM} \\
C_g &= C_{gLCM}\, w_{LCM}
\end{aligned}
\tag{17}
$$

The normalized conductance ($1/R_{LCM}$) is obtained by driving static current through triode-region transistor with various pre-driver supply V_p and driver supply V_s values and by setting output voltage as $V_s/2$. The voltage ranges (V_p, V_s) that cause large non-linearity error are eliminated from the design space. As shown in Fig. 4, the conductance is then expressed as a linear function of V_p and V_s within the range of interest.



a) b)

Figure 4. LCM driver's normalized a) pull-up and b) pull-down conductance



a) C_s extractions setup b) Delay vs. normalized load capacitance

Figure 5. LCM driver delay vs. load capacitance

The normalized capacitance (C_{dLCM}) is extracted by matching the delay of the LCM driver to an equivalent linearized model, Fig. 5a. Since the average of rising and falling delay can be modeled as $0.69R_s(C_{load}+C_s)$ as shown in Fig 5b, fitting the delay into linear formula of normalized load capacitance C_{load}/W_{LCM} gives us C_{dLCM}.

The pre-driving inverters are sized proportional to LCM driver gate capacitance with fanout factor EF using a linear model in (18) with normalized parameters: R_{Pre} (Ohm-um), $C_{g\text{Pre}}$ (fF/um), and $C_{d\text{Pre}}$ (fF/um).

$$
\begin{aligned}
R_{inv} &= R_{\text{Pre}} / w_{pre} \\
C_{ginv} &= C_{g\text{Pre}}\, w_{pre} \\
C_{dinv} &= C_{d\text{Pre}}\, w_{pre}
\end{aligned}
\tag{18}
$$

The normalized parameters for the pre-driving inverter are extracted using a similar procedure as for the LCM driver.

2) Wire Model

Wires are modeled as multi-port lossy transmission lines, with standard strip line bus model sandwiched between two DC planes as shown in Fig. 6. Since we assume the uniform structure in z-dimension, we use 2D field solver to get RLGC matrices for multi-port transmission line models. The RLGC matrices are parameterized by wire width and space but not the wire length, and depend only on the process and the metal layer. The S-parameters for a wire of length d can be derived from RLGC matrices using the telegrapher's equation.

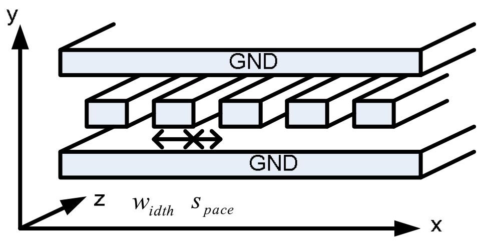


Figure 6. Wire's 2D structure model

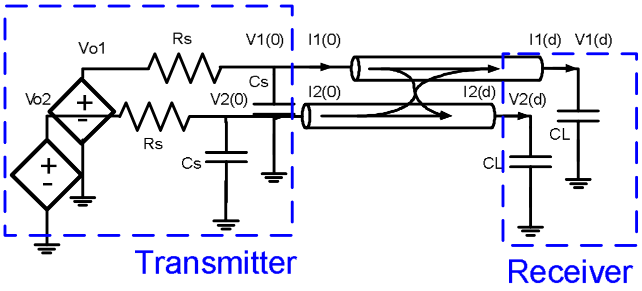


Figure 7. Physical channel model with crosstalk

Skin-effect is added through matrix $\mathbf{R_s}$.

$$\mathbf{R}(f)=\mathbf{R_o}+\sqrt{f}\,(1+j)\,\mathbf{R_s} \tag{19}$$

By assuming that the crosstalk from the first neighbor is the dominant crosstalk, we reduce the extracted m-by-m RLGC matrices of an m-port bus to 2-by-2 RLGC matrices for two neighboring ports. The corresponding 2-by-2 blocks capture diagonal terms (RLGC constant for through transfer function, written as r_o,l_o,g_o,c_o) and first off-diagonal terms (the crosstalk term from the first neighbor, written as r_c,l_c,g_c,c_c).

B. Channel Model

By combining the LCM driver's Thevenin equivalent model and the wire RLGC transmission line model in Fig. 6, we derive the closed form expression for the channel's frequency response, for the interconnect of length d. In Fig. 7, C_L represents the receiver input capacitance, while R_s, C_s, and V_{o1}, V_{o2} model two transmitters.

With $V_1(z,\omega)$, $V_2(z,\omega)$, $I_1(z,\omega)$, and $I_2(z,\omega)$ as the traveling voltages and currents of interconnect 1 and interconnect 2 along the wire at distance z from the driver, the telegrapher's equation for Fig. 7 becomes

$$-\frac{\partial}{\partial z}\begin{bmatrix}\underline{V}(z,\omega)\\ \underline{I}(z,\omega)\end{bmatrix}=\begin{bmatrix}0 & \mathbf{R}+j\omega\mathbf{L}\\ \mathbf{G}+j\omega\mathbf{C} & 0\end{bmatrix}\begin{bmatrix}\underline{V}(z,\omega)\\ \underline{I}(z,\omega)\end{bmatrix}$$
$$\underline{V}(z,\omega)=\begin{bmatrix}V_1(z,\omega)\\ V_2(z,\omega)\end{bmatrix},\ \underline{I}(z,\omega)=\begin{bmatrix}I_1(z,\omega)\\ I_2(z,\omega)\end{bmatrix},\ \underline{V}_o(\omega)=\begin{bmatrix}V_{o1}\\ V_{o2}\end{bmatrix} \tag{20}$$
$$\mathbf{R}=\begin{bmatrix}r_o & r_c\\ r_c & r_o\end{bmatrix},\ \mathbf{L}=\begin{bmatrix}l_o & l_c\\ l_c & l_o\end{bmatrix},\ \mathbf{G}=\begin{bmatrix}g_o & g_c\\ g_c & g_o\end{bmatrix},\ \mathbf{C}=\begin{bmatrix}c_o & c_c\\ c_c & c_o\end{bmatrix}$$

with boundary conditions for distance 0 and d.

$$\begin{aligned}\underline{V}_o(\omega)&=\underline{V}(0,\omega)+R_s\left(\underline{I}(0,\omega)+j\omega C_s\underline{V}(0,\omega)\right)\\ \underline{I}(d,\omega)&=j\omega C_L\underline{V}(d,\omega)\end{aligned} \tag{21}$$

Solving these equations gives closed form expression for through and crosstalk transfer functions, $T(\omega)$ and $T_c(\omega)$

$$\begin{aligned}T(\omega)&\approx T_{com}(\omega)+T_{diff}(\omega)\\ T_c(\omega)&\approx T_{com}(\omega)-T_{diff}(\omega)\end{aligned}$$

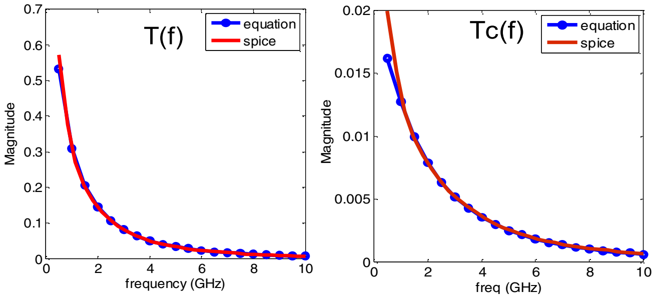


Figure 8. Frequency response matching : equations vs. SPICE

$$T_{com}(\omega)=\frac{e^{-d\sqrt{(z_o+z_c)(y_o+y_c)}}}{\left(j\omega C_L\sqrt{\frac{(z_o+z_c)}{(y_o+y_c)}}+1\right)\left(1+R_s\left(\sqrt{\frac{(y_o+y_c)}{(z_o+z_c)}}+j\omega C_s\right)\right)} \tag{22}$$
$$T_{diff}(\omega)=\frac{e^{-d\sqrt{(z_o-z_c)(y_o-y_c)}}}{\left(j\omega C_L\sqrt{\frac{(z_o-z_c)}{(y_o-y_c)}}+1\right)\left(1+R_s\left(\sqrt{\frac{(y_o-y_c)}{(z_o-z_c)}}+j\omega C_s\right)\right)}$$

where $\begin{bmatrix}z_o & z_c\\ z_c & z_o\end{bmatrix}=\mathbf{Z}=\mathbf{R}+j\omega\mathbf{L}$, $\begin{bmatrix}y_o & y_c\\ y_c & y_o\end{bmatrix}=\mathbf{Y}=\mathbf{G}+j\omega\mathbf{C}$ and the wire input admittance is approximately

$$Y_w(\omega)\approx\sqrt{\frac{y_o}{z_o}}\,. \tag{23}$$

Fig. 8 shows a good match between our equation-based transfer functions with 2-by-2 RLGC matrices and SPICE simulation with 6-by-6 W-element transmission line wire models, which enables fast design space exploration.

C. Power Model

The power model describes the energy dissipation of the pre-driver and driver circuits with output voltage determined by equalization coefficients.

From Fig. 3 model, the power consumption of the LCM driver has both active and static components. Active power is dissipated to swing the wire voltage and to switch the driver itself while static power is lost on the driver resistance stack. Once the equalization coefficients are set, the active energy-per-bit consumed from the supply is computed in frequency domain by taking integral of the voltage-current product of source V_o in Fig. 3

$$E_{active}=\int_{-\infty}^{\infty}\left|V_o(f)\right|^2\,\mathrm{Re}\left\{1/\left(R_s+\frac{1}{j2\pi fC_s+Y_w(f)}\right)^*\right\}df\,. \tag{24}$$

The energy used to drive the wire is

$$E_w=\int_{-\infty}^{\infty}\left|V_o(f)\right|^2\,\mathrm{Re}\left\{\left(Y_w(f)/\left(1+\left(j2\pi fC_s+Y_w(f)\right)R_s\right)\right)^*\right\}df\,. \tag{25}$$

The difference of these two is the energy consumed by the parasitic capacitance of the driver E_{active_drv}=E_{active}-E_w. The static power consumption of the LCM driver is a function of data-dependent time varying resistances R_1 and R_2

$$P_{sc\_drv}(t)=\frac{V_s^2}{R_1(t)+R_2(t)}=\frac{V_o(t)\left(V_s-V_o(t)\right)}{R_s} \tag{26}$$

and the corresponding energy-per-bit is

$$E_{sc\_drv}(t)=P_{sc\_drv}(t)T_s\,. \tag{27}$$

With maximum voltage constraint, the V_o voltage can be written in terms of n_{FFE} consecutive data bits $\underline{x}^T=[x_1\ x_2\ x_3\ ..\ x_{nFFE}]$, and the equalizer coefficients vector $\underline{w}$

$$V_o=\frac{V_s}{2}\left(1+\underline{w}^T\underline{x}\right)\cdot \tag{28}$$

By assuming random link data x_i, the average power of (26) becomes

$$\left\langle P_{sc\_drv}\right\rangle=\frac{V_s^2}{4R_s}\left(1-\left\|\underline{w}\right\|_2^2\right)\cdot \tag{29}$$

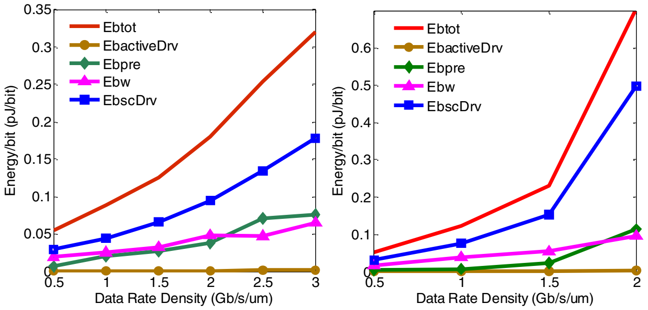
For pre-driver power, we assume that inverter-based pre-driver is sized for fanout of *EF* giving

$$E_{Pre}=\alpha\left(\frac{1/EF}{1-1/EF}\frac{C_{gPre}+C_{dPre}}{C_{gPre}}+1\right)C_{gLCM}w_{LCM}V_p^2 \tag{30}$$

where α is the average activity factor, and C_{gPre}, C_{dPre} are the transistor-width normalized pre-driver gate and diffusion capacitors, and C_{gLCM} is the gate capacitor of the LCM driver. Since the equalized interconnects use differential signaling scheme, the total energy consumption is twice the sum of these variables $E_{btot}=2(E_w+E_{active\_drv}+E_{sc\_drv}+E_{pre})$.

V. Results

In this section we use our tool to explore the link design space for a 10 mm long wire, in 90 nm, 1.2 V CMOS technology. We use the link architecture in Fig. 1b, with 3-tap LCM FFE and 1-tap loop-unrolled DFE receiver. To illustrate the results of joint wire and circuit sizing, we use this architecture and explore the link metrics trade-offs by varying the wire width and spacing, driver size, and supply voltage as well as worst-case eye requirement. In Fig. 9 we show a comparison of the optimized equalized interconnect and repeated interconnect. Repeaters are also optimized jointly with wires to reduce energy cost for the same throughput density and latency as equalized wires [5].



a) M8 wire

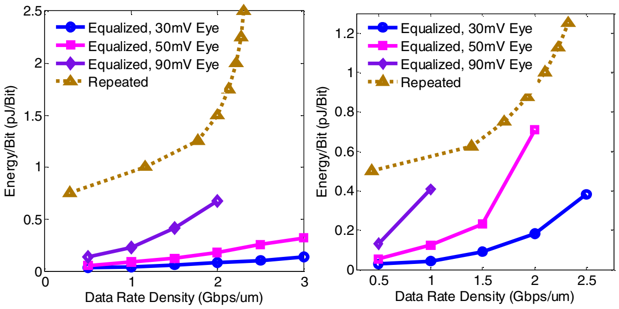
b) M2-6, wire

Figure 10. Energy breakdown of LCM equalized interconnect

For M8 wire, the equalized interconnect consumes much less power than repeated interconnect. In metal layers M2-6, equalized interconnect is still superior to repeaters but the benefit of equalization is smaller than at the M8 layer. Equalization at these lower metal levels enables their use for medium-to-long wires, therefore potentially alleviating the congestion at the top M8 layer in hierarchical on-chip networks. The energy-savings and data-rate density are a strong function of required worst-case eye opening since that determines the required voltage swing on a wire. Interestingly, the equalized interconnect not only decreases the energy-per bit for the same data rate density, but also pushes the trade-off to larger data rates.

In Fig. 10 the energy breakdown for each link component, with pre-driver fanout *EF=3*, shows dominant static power component. This is a shortcoming of the LCM drivers and indicates room for further improvement at the circuit level. Note that in on-chip applications, even with this static power component, LCM drivers are a lot more efficient than high-common mode differential pair drivers [12].

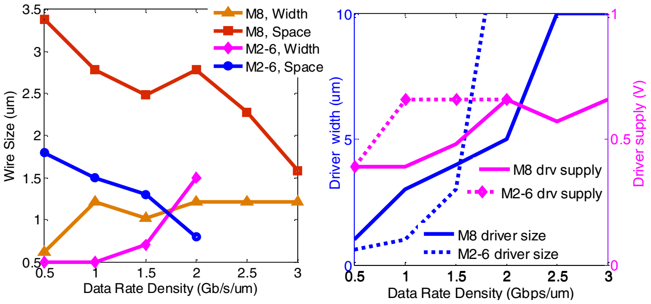
Optimal wire and circuit parameters are shown in Fig. 11. It is very interesting that the optimal wire widths are only 2-3x of the minimum width for M8 and 3-10x of M2-6. Spacing is



a) M8 10 mm wire

b) M2-6 10 mm wire

Figure 9. Comparison of repeated and equalized interconnect



a) Wire width and spacing

b) Driver size and supply

Figure 11. Optimal wire and circuit parameters

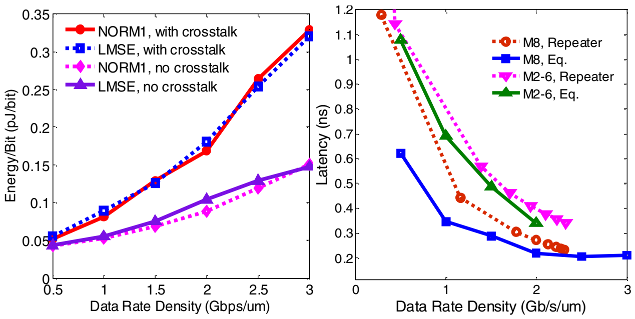


Figure 12. a) Comparison of equalization algorithms (worst-case eye vs. LMSE, M8 wire), b) Latency comparison

TABLE I. EXPLORATION TIME COMPARISON

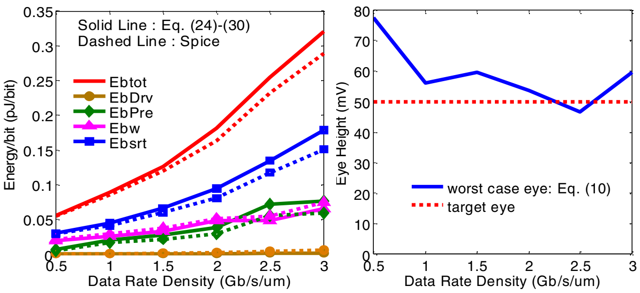
Design Space 423K points	LMSE optimal Td	Norm1 optimal Td	Brute Force (Norm1 with 20x oversample)
Run time (h)	0.74	9.7	180
Normalized	x1	x13	x244

up to 10x the minimum spacing for M8 and M2-6 primarily due to crosstalk. These results indicate that by using equalization we can improve the data rate on a tighter interconnect pitch, which in turn improves energy-efficiency, as opposed to just using very wide wires to push the data rate at a higher energy cost.

In Fig. 12a, we compare LMSE and worst-case eye (l_1-norm) equalization, showing very little difference between the algorithms, even with crosstalk, although worst-case eye method requires 13x more computation time than LMSE. The optimal delay formula speeds up the tool by another order of magnitude. Table I summarizes the total design exploration run time using closed form equalization and optimum delay expressions.

Fig. 12a also shows that crosstalk is a significant factor in energy cost per bit, and data rate density computation. Some improvement is still possible since we did not use differential wire criss-crossing [6,9] to make the crosstalk common-mode. Fig. 12b shows that equalized links have slightly better latency than repeaters.

Fig. 13 illustrates the accuracy of the energy and the worst-case eye prediction models ((24)-(30) and (10)), when compared with SPICE and field-solver simulations.



a) Energy per bit b) Worst-case eye opening

Figure 13. Verification of the model with SPICE simulations

VI. CONCLUSION

This paper shows that properly optimized equalized interconnect has the potential to improve both data rate density and energy-efficiency of on-chip networks. Equalization lowers the energy cost by decreasing the swing across the wire and removing the need for very wide wires by improving the data rate on narrower wires. Lowering the wire width and spacing, and improving the link data rate both improve the utilization of metal layers and energy-efficiency. By using the appropriate models, all levels of link design hierarchy are jointly optimized to provide the best trade-off between data rate density and energy-efficiency. Derived closed form expressions for optimal link latency and LMSE solution for equalizer coefficients improve the run time of our design exploration tool by more than two orders of magnitude, making million design point searches practical. This proves to be sufficient to cover the link design space of interest. The outcome is a set of architecture-friendly link metrics that enable the design of next generation hierarchical on-chip networks.

REFERENCES

[1] R. Ho, K.W. Mai and M.A. Horowitz "The future of wires," *Proceedings of the IEEE* vol. 89, no. 4, pp. 490-504, 2001.

[2] M.B. Taylor et al. "The Raw microprocessor: a computational fabric for software circuits and general-purpose programs," *IEEE Micro,* vol. 22, no. 2, pp. 25-35, 2002.

[3] D.C. Pham et al. "Overview of the architecture, circuit design, and physical implementation of a first-generation cell processor," *IEEE Journal of Solid-State Circuits,* vol. 41, no. 1, pp. 179-196, 2006.

[4] K. Banerjee and A. Mehrotra "A power-optimal repeater insertion methodology for global interconnects in nanometer designs," *IEEE Transactions on Electron Devices,* vol. 49, no. 11, pp. 2001-2007, 2002.

[5] T. Lin and L. T. Pileggi "Throughput-driven IC communication fabric synthesis," *IEEE International Conference on Computer Aided Design,* pp. 279-284, 2002.

[6] R. Ho, K. Mai and M. Horowitz "Efficient on-chip global interconnects," *Symposium on VLSI Circuits, Digest of Technical Papers*, pp. 271-274, 2003.

[7] D. Schinkel, E. Mensink, E.A. Klumperink, E. van Tuijl and B. Nauta "A 3-Gb/s/ch transceiver for 10-mm uninterrupted RC-limited global on-chip interconnects," *IEEE Journal of Solid-State Circuits,* vol. 41, pp. 297-306, 2006.

[8] A.P. Jose, G. Patounakis and K.L. Shepard "Near speed-of-light on-chip interconnects using pulsed current-mode signalling," *IEEE Symposium on VLSI Circuits*, pp. 108-111, 2005.

[9] S.R. Sridhara, N.R. Shanbhag and G. Balamurugan "Joint equalization and coding for on-chip bus communication," *IEEE Sixth International Symposium on Quality of Electronic Design*, pp. 642-647, 2005.

[10] V. Stojanović, A. Ho, B.W. Garlepp, F. Chen, J. Wei, G. Tsang, E. Alon, R.T. Kollipara, C.W. Werner, J.L. Zerbe and M.A. Horowitz "Autonomous dual-mode (PAM2/4) serial link transceiver with adaptive equalization and data recovery," *IEEE Journal of Solid-State Circuits,* vol. 40, no. 4, pp. 1012-1026, 2005.

[11] J. Ren and M. Greenstreet "A unified optimization framework for equalization filter synthesis," *Design Automation Conference*, pp. 638-643, 2005.

[12] H. Hatamkhani, K.-L.J. Wong, R. Drost, Chih-Kong Ken Yang, “A 10-mW 3.6-Gbps I/O transmitter,” *IEEE Symposium on VLSI Circuits*, June 2003, pp. 97-99.