# SIMPLE MODELS FOR POWER OPTIMIZATION ACROSS TRANSMISSION, EQUALIZATION AND DECODING

*Pulkit Grover, Anant Sahai, and Ji-Hoon Park*

UC Berkeley, Berkeley, CA-94720, USA
{pulkit, sahai, overlord}@eecs.berkeley.edu

## ABSTRACT

This paper provides simplistic models for optimization over transmit, equalization, and decoding power in a communication system for a given attainable rate, bandwidth, error-probability, and the channel taps between the transmitter and the receiver. For equalization, we focus on the digital feedback equalizer (DFE) and observe that the number of taps equalized should increase with the transmit power, and characterize the regime to where additional available power should be invested in transmission or equalization. For decoding, we focus on the message-passing decoder for any code, and provide a model for power consumed in the computational nodes and interconnects of the decoder, using which we formulate an optimization problem to minimize the total power consumed in transmission, equalization, and decoding.

## 1. INTRODUCTION

In a communication system, information travels from the sender to the receiver not merely through the communication channel, but also through processing circuits and chips on the transmitting and receiving modules. Conceptually, a theory of information *processing* is thus an integral part of the theory of information communication. Practically, the decreasing distances have brought the transmit power down to the level of processing power (or even lower [1]) and hence a theory of information that does not fully understand processing could also have diminished practical significance.

How do we build a theory of information that also includes processing computations? *Prima facie*, it seems that we have the tools: the existing theory of communication needs to be coupled with the theory of computation, because the latter intends to understand precisely the computational requirements for various problems. The greatness of the Turing machine model in the theory of computation is its simplicity and generality: it can be adapted to simulate the logic of any computer algorithm. The hope therefore is that the results from the study of Turing's machine can help understand computation for processing signals at the transmitting and receiving ends as well.

However, the results on Turing's machine fall short on two counts. First, the connection between relevant parameters such as power, area, and (more controversially) time required for computation and the abstraction of Turing's model is merely suggestive (see [2] for a deeper look into the issue). Second, and more importantly, a significant aspect of theory of information are fundamental bounds that limit the performance of *any* communication scheme through converses. Similar fundamental bounds on Turing's machine have been extremely hard to find[1].

For the purpose of understanding computation for an enhanced theory of information, it is therefore prudent to shift attention to more restricted models of computation that could be closer to implementation. Towards building a such a theory of information, we focused on models of decoding, borrowing ideas from the "VLSI model" of computation [4], and the model of communication complexity [5] in order to understand decoding power [1, 6, 7]. In [6], we derive fundamental lower bounds on neighborhood size of message-passing decoding that we use to obtain lower bounds on power consumed in computational nodes [1] and wires [7] in a VLSI implementation of a decoder. These bounds show not only that there is a fundamental tradeoff between transmit and decoding power, and but also that one needs to operate at a non-zero gap from capacity in order to minimize total power. Traditionally optimal codes that operate close to capacity are no longer the best because they require large number of decoding iterations and long wires in decoding implementation, contributing to a large decoding power.

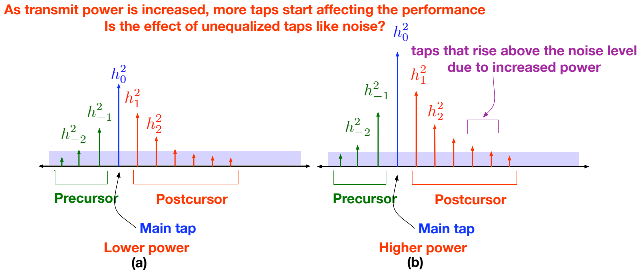

As transmit power is increased, more taps start affecting the performance
Is the effect of unequalized taps like noise?
$h_0^2$
$h_{-1}^2$
$h_{-2}^2$
$h_1^2$
$h_2^2$
Precursor
Postcursor
Main tap
Lower power
(a)
$h_0^2$
taps that rise above the noise level due to increased power
$h_{-1}^2$
$h_{-2}^2$
$h_1^2$
$h_2^2$
Precursor
Postcursor
Main tap
Higher power
(b)

Figure 1: An intuitive illustration of why the DFE power consumption should increase with transmit power consumption for full benefit: an increase in transmit power also increases the impact that the unaccounted paths can have on the performance. The number of taps (and thus also the power consumption) of the DFE therefore should increase in order to get the most out of the increased power. Somewhat surprisingly, our results show that the noise-level might not matter in the calculation of the number of taps that should be equalized.

Taking this understanding a step further, a communication system is not simply about the choice of a coding scheme. One also has to choose strategies for modulation/demodulation, equalization, etc. In this paper, we seek to extend our understanding to include equalization as well. What is equalization useful for? An example channel re-

We thank Jan Rabaey and Karthik Ganesan for useful discussions. We also gratefully acknowledge the support of NSF-CCF-0917212.

[1] The most obvious instantiation of this statement is the fact that the hypothesis $P \neq NP$ is still unproven. More interestingly, the best lower bound on the complexity of 3-SAT, an NP-complete problem, is slower than quadratic in the size of the input [3] even with space restrictions!

sponse for short-distance communication is shown in Fig. 1. Because of numerous objects that exist in indoor environments, the number of multipaths can often be high, introducing Inter Symbol Interference (ISI). Equalization is performed to reduce the effects of ISI. Because of the large number of filter taps, the power required for equalization in indoor environments can be large [8].

Unlike the vast freedom in choice of a code, the choice of equalization technique is limited to a few in practice, such as using OFDM followed by FFT, matched filtering, decision feedback equalizer (DFE), joint equalization and decoding, etc. (see e.g. [8] and the references therein). Empirical comparison of DFE with other techniques in high-rate short distance applications (e.g. for 60 GHz band in [9]) shows that a DFE is often more power efficient. For simplicity, to begin understanding equalization, we focus on just the DFE. The larger goal for the future is to obtain a fundamental understanding under a model of computation that is general enough to encompass these techniques as well as others that are yet to be discovered.

While an empirical comparison of these models of equalization has been performed in the literature (see e.g. [8]), these models often assume that all relevant paths have been equalized. As shown in Fig. 1, the impact of unequalized paths can increase with increasing transmit power. Because our focus is on minimizing the total system power, we need to allow for large transmit powers where more paths may become relevant than those at low transmit power. For instance, a question of interest is: suppose each tap requires $P_{tap}$ W of power. Given a working system, if we have an additional $P_{tap}$ W of power available, where should we invest it? In transmission, equalization, or decoding?

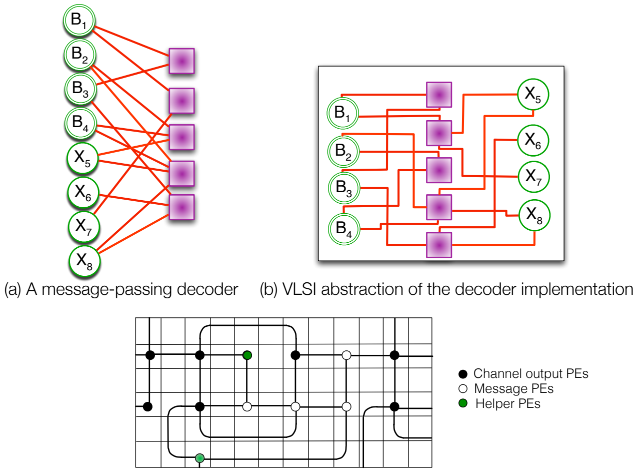
We first provide models for power consumption in a DFE equalizer and a decoder in Section 2. We use these models to analyze the required power for the DFE equalizer: we observe in Section 3.1 that if transmit power is increased, the extra "unequalized paths" (see Fig. 1(b)) can start affecting the performance. Interestingly, we show the impact of unaccounted taps can be much worse than that of noise of equal power. We also show that the DFE taps (and therefore, the DFE power consumption) should also be increased as transmit power is increased in order to maximize the benefit of increased power consumption. In Section 3.2, we analyze the decoding power based on models for node and wire power consumption at the decoder developed in Section 2.3. In Section 3.3, we then lay down the optimization framework for optimizing power consumption across transmission, equalization, and decoding.

## 2. SYSTEM MODEL AND POWER CONSUMPTION MODELS

### 2.1 System model

The communication system has a transmitter, who sends $k_b$ bits of information by coding them together using a block-code at rate $R$ bits per second. The goal is to receive these bits with an average bit-error probability of at most $P_e$. $P_T$ is the transmit power "over the air," the efficiency of the power amplifier is denoted by $\eta$. For simplicity, we assume that $\eta$ does not depend on $P_T$ even though this is only an approximation. Thus the total power required for transmission is $\frac{P_T}{\eta}$. The transmitter is assumed to use BPSK signaling to transmit the channel inputs $X_i$. The coded symbols are received at the receiver who first performs a channel equalization (using a DFE for the postcursor coupled with a linear equalizer [10] for the precursor).

In order to focus on unequalized taps, we ignore the effect of quantization introduced by the ADC and the option of using mixed analog and digital DFE. For simplicity of exposition, we assume that there are no taps in the precursor. Because of our assumption of using a linear equalizer for the precursor, the equalization of the precursor adds a constant amount to the total power consumption and can be ignored in the optimization.



(a) A message-passing decoder (b) VLSI abstraction of the decoder implementation

(c) VLSI abstraction of a general decoder implementation

Figure 2: (b) is an abstraction of the implementation of the decoder shown in (a). The graph that models this chip is the obvious one: each PE is represented by a node, and each wire by an edge.

We first need a model of decoding process in order to account for power consumed in decoding. In [1], we introduced a "VLSI model of decoding" (see Fig. 2) that is inspired by Thompson's "VLSI model from computation" [4]. The decoder consists of computational nodes (or processing elements, PEs) connected to each other using wires. These nodes are either (a) 'message' nodes that store the decoded bits after decoding, (b) 'channel output' nodes that store the channel outputs, (c) 'helper' nodes that act as intermediaries of processing by improving connectivity, or (d) any combination of (a), (b), and (c)

In [1], we ignore the power consumed in all nodes but the "channel output" nodes, that correspond to variable nodes in a decoder. In [7], we provided lower bounds on wire-lengths as a function of code performance, but did not connect the wire-lengths to power consumption. Here, we complete that link: in Section 2.3, we introduce models of decoding power consumption that abstract the power consumed in computational nodes as well as the decoder interconnects.

### 2.2 DFE power consumption

The total number of taps in the postcursor is denoted by $N$. The received signal at time $k$, $y(k)$ is given by

$$y(k) = \sum_{i=0}^{N} h_i X_{k-i} + Z_k, \tag{1}$$

where $Z_k$ is the additive white Gaussian noise and $h_i$ are the coefficients corresponding to various signal paths. The coefficient $h_0$ corresponds to the main path. The part that is equalized in the postcursor is assumed to be from taps 1 to $N_1$. Thus the taps $N_1+1$ to $N$ are assumed to not have been equalized (see Fig. 1). Further, we assume that each filter tap requires $P_{tap}$ Watts of power for equalization.

We assume that the receiver makes a thresholding hard decision on the bit value that is blind to the equalization before (and if) it starts decoding, and thus decoder effectively sees channel outputs corrupted by a binary channel of crossover probability $p_{ch}$.

## 2.3 A model for power consumed in decoding implementation

### 2.3.1 *Model for power consumed by computational nodes*

We assume that all nodes consume the same amount of power regardless of the clock speed and $V_{DD}$. We note that this assumption is inaccurate because power consumption can vary across nodes and can depend on the clock speed and $V_{DD}$.

The energy consumed in decoder nodes in $l$ iterations is, therefore,

$$E_{total;nodes} = n_{total} E_{node} \times l, \tag{2}$$

where $n_{total}$ is the total number of computational nodes at the decoder, $E_{node}$ is the energy consumed by a node in one iteration, and $l$ denotes the number of iterations.

In this simplistic picture, the power consumed in computational nodes is therefore given by:

$$P_{nodes} = \frac{n_{total} E_{node} \times l}{T_{dec}}, \tag{3}$$

where $T_{dec}$ is the time required for decoding that depends on the decoding throughput $R_{dec}$. The decoding throughput is given by

$$R_{dec} = \frac{k_b}{T_{dec}}, \tag{4}$$

where $k_b$ is the number of bits decoded in parallel by the decoder (in time $T_{dec}$). Thus,

$$P_{nodes} = \frac{n_{total} E_{node} l R_{dec}}{k_b}. \tag{5}$$

### 2.3.2 *Model for power consumed in decoder interconnects*

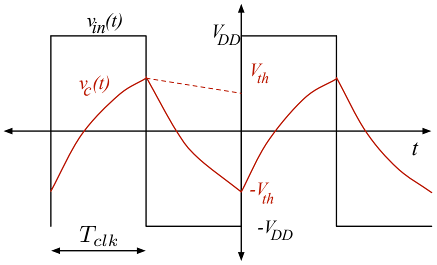


Figure 3: The charging and discharging of decoder interconnects. $v_c(t)$ is the capacitor voltage as it varies with time. $V_{th}$ is the required voltage to make the gates work.

Suppose each gate (that receives messages at the computational nodes from the connecting wires) requires an input voltage of $V_{th}$ volts in operate reliably. For simplicity, we assume that each wire is charged and discharged at each time instant, even though this is not true in practice[2]. Modeling each interconnect by the Elmore lumped model [11], the total energy consumed in charging a wire from zero volts to $V_{th}$ volts is given by $\frac{1}{2}C_{wire}V_{DD}V_{th}$, where $V_{DD}$ is the maximum voltage applied in order to charge the capacitance.

Now, $V_{th} = V_{DD}(1 - e^{-T_{clk}/r_{wire}C_{wire}})$, where $r_{wire}$ is the wire resistance, and $T_{clk}$ is the clock-cycle. For $T_{clk} \ll r_{wire}C_{wire}$ (i.e. corresponding to high throughputs[3]), $e^{-\frac{T_{clk}}{r_{wire}C_{wire}}} \approx 1 - \frac{T_{clk}}{r_{wire}C_{wire}}$. Thus, $V_{th} \approx V_{DD}\frac{T_{clk}}{r_{wire}C_{wire}}$. Since $V_{th}$ is fixed by the required gate voltage, we get $V_{DD} \approx \frac{V_{th}r_{wire}C_{wire}}{T_{clk}}$. Thus, the energy consumed in each wire in each clock-cycle can be approximated by:

$$E_{wire} = \frac{1}{2}C_{wire}V_{DD}V_{th} = \frac{1}{2}\frac{r_{wire}C_{wire}^2}{T_{clk}}V_{th}^2 \tag{6}$$

How do we calculate the capacitance of a wire? Using Elmore lumped model [11], $C_{wire} = C_{parallel-plate} + C_{fringe} + C_{interwire}$, where $C_{parallel-plate}$ is the capacitance of the wire with the substrate, $C_{fringe}$ is the fringing capacitance, and $C_{interwire}$ is the capacitance between two wires running close[4] to each other. $C_{parallel-plate}$ and $C_{fringing}$ are proportional to the wire-length $W$, and $C_{interwire}$ is proportional to the length of parallel tracks of the two wires. As a simplification, we assume that $C_{wire}$ itself is simply proportional to the length of the wire. This approximation appears reasonable when the interconnects are stacked closely together, which is often the case for decoding circuits. In order to obtain first order results, we observe that for random codes as well as for many structured code constructions (e.g. see [12–14]), both the average wire-length and the maximum wire-length scale linearly with the blocklength. This justifies another assumption: that the average wire-length can be approximated by the maximum wire-length.

Assuming that the state of the wire switches at each iteration, the total energy consumed in the interconnects in $l$ iterations can now be approximated as

$$\begin{aligned} E_{wires\,total} &= \sum_{i=1}^{n_{wire}} \frac{1}{2}\frac{r_i C_i^2}{T_{clk}} V_{th}^2 \times l \\ &= \frac{n_{wire}}{2}\frac{r_{wire}C_{wire}^2}{T_{clk}} V_{th}^2 \times l. \end{aligned}$$

Here $C_i$ is the total capacitance of the $i$-th wire, $n_{wire}$ is the total number of interconnects, and $C_{wire}$ and $r_{wire}$ are the capacitance and the resistance of the longest interconnect. Defining $C_{unit}$ and $r_{unit}$ as the total capacitance and average resistance per unit-length for an interconnect, the total *power* is approximately

$$\begin{aligned} P_{wires} &= \frac{n_{wire}}{2}\frac{r_{unit}C_{unit}^2}{T_{clk}T_{dec}} W_{max}^3 V_{th}^2 \times l \\ &\overset{(T_{dec}=T_{clk}l)}{=} \frac{n_{wire}}{2}\frac{r_{unit}C_{unit}^2}{T_{clk}^2 l} W_{max}^3 V_{th}^2 \times l \\ &= \frac{n_{wire}}{2}\frac{r_{unit}C_{unit}^2}{T_{clk}^2} W_{max}^3 V_{th}^2. \end{aligned}$$

[2] Messages tend to stabilize as decoding proceeds. However, using a non-fully parallel decoding architecture can slow this stabilization significantly.

[3] This approximation will only provide a conservative estimate of the required power because $V_{DD}$ scales exponentially in $V_{th}$ at large $T_{clk}$.

[4] The proximity is measured by comparing the distance between the wires to the width of the wire.

Further, the time required for decoding, $T_{dec} = \frac{k_b}{R_{dec}} = T_{clk} \times l$. This implies that $T_{clk} = \frac{k_b}{R_{dec} l}$. Thus,

$$P_{wires} = \frac{n_{wire}}{2} \frac{r_{unit} C_{unit}^2 V_{th}^2 R_{dec}^2}{k_b^2} W_{max}^3 l^2. \tag{7}$$

Thus, the power consumed in the interconnects increases as the cube of the length of the longest wire and as the square of the number of iterations.

## 3. ANALYSIS AND OPTIMIZATION OF TOTAL POWER CONSUMPTION

For simplicity, we assume a separation between equalization and decoding even though joint equalization-decoding architectures have been suggested in the literature [15].

### 3.1 Analysis of equalization power

The received signal is given by

$$\begin{aligned} y(k) &= \sum_{i=0}^{N} h_i X_{k-i} + Z_k \\ &= \left\{ \sum_{i=0}^{N_1} h_i X_{k-i} \right\} + \left( \sum_{i=N_1+1}^{N} h_i X_{k-i} \right) + Z_k, \end{aligned}$$

The term in simple brackets $(\cdot)$ is the term that has not been equalized at the receiver. The signal remaining after the action of DFE is given by

$$r(k) = h_0 X_k + \left( \sum_{i=N_1+1}^{N} h_i X_{k-i} \right) + Z_k.$$

The noise and the ISI from the "unequalized paths" together can cause the bit to be in error at the receiver (post-equalization). Under the event that the bit at time $j > 0$ is $b_j \text{sgn}(h_j)$, the signal power post-equalization is given by

$$P_{sig}(P_T, \vec{b}) = P_T \left( |h_0| - \sum_{j=N_1+1}^{N} b_j |h_j| \right)^2. \tag{8}$$

Based on hard-decision at the receiver, the raw bit-error probability $P_e^{(u)}$ is given by

$$\begin{aligned} P_e^{(u)} &= \sum\nolimits_{b_j \in \{-1,1\}} \tfrac{1}{2^{N-N_1}} \mathbb{Q} \left( \frac{\sqrt{2 P_{sig}(P_T, \vec{b})}}{\sqrt{\sigma_z^2}} \right) \qquad (9) \\ &\overset{(a)}{\approx} \tfrac{1}{2^{N-N_1}} \mathbb{Q} \left( \frac{\sqrt{2 P_{sig}(P_T, \vec{1})}}{\sqrt{\sigma_z^2}} \right), \end{aligned}$$

where $(a)$ follows from the observation that at high SNR, the summation is dominated by the term where $b_j = 1$ for all $j$ (because the $\mathbb{Q}$-function falls exponentially in its argument). With this approximation, the designer's goal is maximize the product term $\Pi := P_{sig}(P_T, \vec{1})$

$$\Pi = \sqrt{P_T} \times \left( |h_0| - \sum_{j=N_1+1}^{N} |h_j| \right), \tag{10}$$

If $P_{tap}$ W of power is available, where $P_{tap}$ is the required power to run one filter tap, where should it be invested? Investing in transmit power, it increases the transmit power by $\eta P_{tap}$, and thus brings a marginal gain of

$$Gain_{Tx} = \sqrt{\frac{P_T + P_{tap}\eta}{P_T}}, \tag{11}$$

to the product (here $\eta$ is the efficiency of the power amplifier at the transmitter). Invested in equalization power, it brings a marginal gain of

$$Gain_{eq} = \frac{|h_0| - \sum_{j=N_1+2}^{N} |h_j|}{|h_0| - \sum_{j=N_1+1}^{N} |h_j|} \tag{12}$$

(assuming $|h_{N_1+1}|$ is the unequalized tap of largest magnitude) to the product. The power should therefore be invested depending on which of the gains ((11) or (12)) is larger.

Notice that, counter to our intuition in Fig. 1, neither (11) nor (12) depend on the noise level $\sigma_z^2$! This is because the impact of unequalized filter-taps is not like noise. Instead, they *reduce the effective signal power* (see (10)). Of course, in some cases, they can add to the signal power too, but because of the exponential decay of the $\mathbb{Q}$ function, the dominant error probability term in (9) is that which corresponds to power reduction.

### 3.2 An analysis of the decoding power

How does the wire-length scale with the number of iterations? If the code is a sparse-graph code, assuming that the decoding algorithm runs only until the point that all the decoding neighborhoods are trees, $l = \frac{g}{2} - 1$, where $g$ is the girth of the code-graph, the longest wire scales exponentially in the girth [7, Theorem 1]. Thus, if one designs a code that has a larger girth than what is required by the number of iterations that will be executed, the interconnect power consumption can increase sharply.

What if the code is not a sparse-graph code? Even so, if the decoder implementation can be modeled as in Section 2.1, it is shown in [7] that the product of $W_{max}$ and the number of iterations $l$ is lower bounded as follows for any 2-D decoder chip (the bounds can easily be extended to 3-D chips)

$$W_{max} \times l \gtrsim c\sqrt{A_{node}} \frac{\sqrt{\log \frac{1}{P_e}}}{C(P_T) - R}, \tag{13}$$

where $A_{node}$ is the area of any computational node that stores channel outputs. Since interconnect power grows only quadratically in the number of iterations, but cubically in the wire-length, from (13) it seems it is best to run more iterations and use wire-lengths as small as possible, keeping the product $W_{max} l$ close to the lower bound. However, use of more iterations increases $P_{nodes}$, hence we need to look at the total decoding power.

#### Total decoding power consumption

From (5) and (7), the total power consumed at the decoder can therefore be modeled as

$$\begin{aligned} P_{dec} &= P_{nodes} + P_{wires} \\ &= \frac{n_{total} E_{node} R_{dec}}{k_b} l + \frac{n_{wire} r_{unit} C_{unit}^2 V_{th}^2 R_{dec}^2}{2 k_b^2} W_{max}^3 l^2. \end{aligned}$$

### 3.3 An optimization framework for total power consumption

How do we put the the analysis of Section 3.1 and 3.2 together? Since we assume that a hard decision is performed after equalization, it is tempting to think of $p_{ch}$ as the crossover probability of a (*memoryless*) BSC at the output of the DFE. However, one needs to be careful: while our bounds in [1, 7] assume that the obtained messages have independent errors, the unequalized taps in a DFE can introduce correlations in messages sent by variable nodes. Nevertheless, we ignore this correlation in the hope that the randomness in the code-structure will ensure that most of the channel outputs in most of the decoding neighborhoods will have independent errors.

Ignoring the above correlation, the total power is

$$P_{tot} = \min_{P_T : C(P_T) > R, N_1, code, l} \frac{P_T}{\eta} + P_{eq} + P_{dec}(P_e, R),$$

where $P_{eq} = P_{tap} N_1$, and $P_{dec} = P_{nodes} + P_{wire}$. The optimization is over code as well as the decoder implementation and the number of iterations $l$ for which the decoding is run. The decoding throughput $R_{dec}$ can be assumed to be equal to the rate $R$ of communication.

What does our analysis above tell us about this optimization? For instance, we can again ask the question: where should a small amount of extra available power be invested? From (11) and (12), we know how to choose between allocating this power to the equalizer or the transmitter. Decoding power makes this choice more complicated because of the plethora of possible codes and decoders. To get an estimate, we can lower bound this optimization by using the lower bounds on the decoding power (with help from the complexity lower bounds in [1, 7]) discussed in Section 3.2. As in [1, 7], we should not operate too close to the channel capacity, or over-design codes for error probabilities lower than required because both of these will require large decoding power. Further, as noted in [1], while LDPC codes might be a wise choice (because of the decay in error probability with the number of iterations), one should not use capacity-approaching LDPC codes because they require use of degree-2 nodes which exponentially slows down the pace of reduction of error probability [1]. Once this lower bound has been evaluated, we can then choose the code/decoder that performs closest to the lower bound. This approach could guarantee a limited gap from optimality.

## 4. DISCUSSIONS

In this paper we provided models for power consumed in decision feedback equalizer (DFE) and in the nodes and interconnects of a message-passing decoder. A more satisfying theory would not only allow for understanding other possible ways of equalizing (e.g. matched-filter, OFDM, rake receiver, etc.), but would also account for power consumed in encoding and ADC, and allow for joint processing techniques. One promising recent technique is that of turbo equalization [16] that is compatible with the message-passing decoding architecture and might therefore be amenable to an analysis similar to that in [1].

However, a satisfying theory could require a modeling of all (possible) equalization techniques, which will likely be extremely hard. This brings out the biggest obstacle in understanding computation for communication, or even more broadly, for embedded and cyber-physical systems [2]. While we would like to use models that are restrictive enough that interesting results on power, area, and time can be obtained, if the models only abstract the existing architectures, they may not suggest radically different and improved techniques from those the existing ones. Striking the right balance between model restrictions and generality could yield rich dividends.

## REFERENCES

[1] P. Grover, K. A. Woyach, and A. Sahai, “Towards a communication-theoretic understanding of system-level power consumption,” *Journal of Selected Areas in Communications, special issue on energy-efficient communications, to appear. Arxiv preprint arXiv:1010.4855*, 2011.

[2] E. Lee, “Computing needs time,” *Communications of the ACM*, vol. 52, no. 5, pp. 70–79, 2009.

[3] [Online]. Available: http://www.eecs.berkeley.edu/∼sseshia/172/lectures/PhaseTransitionsSAT.pdf

[4] C. D. Thompson, “A complexity theory for VLSI,” Ph.D. dissertation, Pittsburgh, PA, USA, 1980.

[5] A. C.-C. Yao, “Some complexity questions related to distributive computing(preliminary report),” in *STOC ’79: Proceedings of the eleventh annual ACM symposium on Theory of computing*. New York, NY, USA: ACM, 1979, pp. 209–213.

[6] A. Sahai and P. Grover, “A general lower bound on the decoding complexity of message-passing decoding,” in *preparation*, 2010. [Online]. Available: http://www.eecs.berkeley.edu/∼pulkit/papers/ComplexityITPaper.pdf

[7] P. Grover and A. Sahai, “Fundamental bounds on the interconnect complexity of decoder implementations,” in *Conference on Information Sciences and Systems (CISS)*, Baltimore, MD, Mar. 2011.

[8] J.-H. Park and B. Nikolic, “Mixed-signal power optimization of a multi-Gb/s equalizer in the 60-GHz band,” in preparation.

[9] J.-H. Park and D. Stepanovic, “A BER performance and power characterization of a 60GHz transceiver,” 2009, course project.

[10] J. Barry, E. Lee, and D. Messerschmitt, *Digital communication*. Springer Netherlands, 2004.

[11] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital integrated circuits*. Prentice Hall Englewood Cliffs, New Jersey, 2002.

[12] M. Mansour and N. Shanbhag, “High-throughput LDPC decoders,” *IEEE Tran. VLSI Systems*, vol. 11, pp. 976–996, 2003.

[13] J. Thorpe, “Low-density parity-check (LDPC) codes constructed from protographs,” *IPN Progress Report 42-154, JPL*, 2005.

[14] M. Mohiyuddin, A. Prakash, A. Aziz, and W. Wolf, “Synthesizing interconnect-efficient low density parity check codes,” in *Proceedings of the 41st annual Design Automation Conference*, ser. DAC ’04. New York, NY, USA: ACM, 2004, pp. 488–491.

[15] H. Lee and V. Gulati, “Iterative equalization/decoding of LDPC code transmitted over MIMO fading ISI channels,” in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2002*, vol. 3, 2002, pp. 1330–1336.

[16] M. Tuchler, R. Koetter, and A. Singer, “Turbo equalization: principles and new results,” *IEEE Transactions on Communications*, vol. 50, no. 5, pp. 754–767, 2002.