Towards Approaching Total-Power-Capacity: Transmit and Decoding Power Minimization for LDPC Codes

Karthik Ganesan†, Pulkit Grover‡, Jan Rabaey§, and Andrea Goldsmith†

Abstract

Motivated by recently derived fundamental limits on total (transmit + decoding) power for coded communication, this paper investigates how close regular LDPC codes can get to these fundamental limits. For two decoding algorithms (Gallager-A and Gallager-B), we provide upper and lower bounds on the required decoding power based on models of parallelized decoding implementations. As the target error-probability is lowered to zero, we show that the transmit power must increase unboundedly in order to minimize total (transmit + decoding) power. Complementing our theoretical results, we develop detailed physical models of decoding implementations using rigorous (post-layout) circuit simulations, and use them to provide a framework to search for codes that may minimize total power. Our results show that approaching the total-power channel capacity requires increasing the complexity of both the code design and the corresponding decoding algorithm as the communication distance is increased, or as the target error-probability is lowered.

Index Terms

Low-density parity-check (LDPC) codes; Iterative message-passing decoding; Total-power channel capacity; Energy-efficient communication; System-level power consumption; Circuit power consumption; VLSI complexity theory.

I. INTRODUCTION

Intuitively, the concept of Shannon capacity captures how much information can be communicated across a channel under specified resource constraints. While the problem of approaching Shannon capacity under solely transmit power constraints is well understood, modern communication often takes place at transmitter-receiver distances that are very short (e.g., on-chip communication [52], short distance wired communication [29], and extremely-high-frequency short-range wireless communication [2]). Empirically, it has been observed that at such short distances, the power required for processing a signal at the transmitter/receiver circuitry can dominate the power required for transmission, sometimes by orders of magnitude [29], [20], [9]. For instance, the power consumed in the decoding circuitry of multi-gigabit-per-second communication systems can be hundreds of milliwatts or more (e.g. [29], [8]), while the transmit power required is only tens of milliwatts [9]. Thus, transmit power constraints do not abstract the relevant power consumed in many modern systems.

Shannon capacity, complemented by modern coding-theoretic constructions [49], has provided a framework that is provably good for minimizing transmit power (e.g. in power constrained AWGN channels). In this work, we focus on a capacity question that is motivated by total power: at what maximum rate can one communicate across a channel for a given total power, and a specified error probability? Alternatively, given a target communication rate and error probability, what is the minimum required total power? The first simplifying perspective to this problem was adopted in [24], [22], where all of the processing power components at the transmitter and the receiver were lumped together. However, processing power is influenced heavily by

† Electrical Engineering, Stanford University. ‡ Electrical and Computer Engineering, Carnegie Mellon University. § Electrical Engineering and Computer Science, University of California - Berkeley. (Email correspondence should be addressed to karthik3@stanford.edu.) Preliminary results from this paper were presented at the 2012 Allerton Conference [14] and IEEE Globecom 2012 [13].
the specific modulation choice, coding strategy, equalization strategy, etc. [29], [20]. Even for a fixed communication strategy, processing power depends strongly on the implementation technology (e.g. 45 nm CMOS) and the choice of circuit architecture.

With simplistic encoding/decoding models, recent literature has explored fundamental limits [20], [21], [35], [33], [34], [5], [6] on transmit + encoding + decoding power. These limits hold for any code and any encoding/decoding algorithm, under specific implementation models. The models abstract power consumed in computational nodes [20], [33], [34] and wiring [35], [21], [5], [6] in the encoder/decoder implementations, and show that there is a fundamental tradeoff between transmit and encoding/decoding power. The simplicity of these models is needed\(^1\) in order to obtain universal limits.

In this work, we therefore pose the question (see Fig. 1): how small is the gap between known families of codes and decoding algorithms and fundamental limits on total\(^2\) power? To address this question, we first provide asymptotic upper and lower bounds (Sections III- IV) on required decoding power. Our novelty, in comparison with the fundamental limits [21], lies in performing these analyses by restricting our attention to regular LDPC codes and Gallager decoding algorithms. This choice is motivated by both the order-optimality of regular LDPC codes in some theoretical models of circuit power [20], and their practical utility in both short [1] and long [15] distance settings. Recent work of Blake and Kschischang [6] also studied the energy of LDPC decoding circuits, and an important connection to this work is highlighted in Section IV-D.

Within these restrictions we provide the following insights:

1) Wiring power, which explicitly brings out physical constraints in a digital system [47], costs more in the low error-probability limit than the power consumed in computational nodes. Thus, it is not sufficient to simply count the number of algorithmic operations and use this as a metric of decoding complexity if power is important.

2) Approaching Shannon capacity requires keeping transmit power near the Shannon limit even as the error probability approaches zero. However, when total-power-minimization is the goal, keeping transmit power bounded (e.g., by approaching Shannon capacity) can lead to suboptimal decoding power. For instance, we observe that (Theorems 3 and 4) at sufficiently low error-probability, it is more total-power-efficient to use uncoded transmission than regular LDPC codes with Gallager decoding, if using bounded transmit power. However, if transmit power is allowed to grow unboundedly, LDPC codes can outperform uncoded transmission in this total-power sense.

3) Under the assumption that processing power is dominated by nodes as opposed to wires, fundamental limits on total power can be achieved in the low error-probability limit via regular LDPC codes with Gallager-B decoding. However, when wires dominate, a large gap exists between fundamental limits and lower bounds on Gallager decoding.

To obtain insights on optimizing code choices for system implementations at practical error-probabilities, we then develop empirical models of decoding power consumption of 1 and 2 bit message-passing algorithms for regular LDPC codes (Section V-C). These models are constructed by simulating (post-layout) power consumption for simple codes and decoders, breaking down the circuit power into its constituents, and generalizing these constituents of power to structurally similar codes.

Shannon-theoretic analysis yields transmit-power-centric results, which are plotted as “waterfall” curves (with corresponding “error-floors”) demonstrating how close the code performs to the Shannon limit. There, the channel path-loss can usually be ignored because it is merely a scaling factor for the term to be optimized (namely the transmit power), thereby not affecting the optimizing code. Since we are interested in total power, the path-loss impacts the code choice. For simplicity of understanding,

\(^1\) In a nutshell, the models assume that any synchronous VLSI circuit is a set of computational nodes connected to each other using wires.

\(^2\) In this paper, since we focus on transmit and decoding power, we use the term total power to denote just the sum of transmit and decoding power.
path-loss is translated into a more relatable metric — communication distance — using a simple model for path-loss. The resulting question is illustrated in Fig. 1(b): At a given data-rate, what code and corresponding decoding algorithm minimize the transmit + decoding power for a given transmit distance and error-probability? In Section V-C, we present optimization results for this question in a 60 GHz communication setting using our models. This particular setting is chosen not just because of the short distance, but also because the results highlight another conceptual point we stress in this paper:

4) Approaching total-power capacity requires an increase in the complexity of both the code design and the corresponding decoding algorithm as the communication distance is increased, or as the target error-probability is lowered.

The results presented in this paper provide a framework for optimizing codes and decoders, which can be used to obtain optimal code-decoder choices for some (but not all) system designs. In particular, we only consider a limited set of coding strategies, and while the results and models presented here extend easily to irregular LDPC constructions, they are not necessarily applicable to all decoders. Second, modern transceivers [39] contain many other processing power sinks, including analog-to-digital converters (ADCs), digital-to-analog converters (DACs), power amplifiers, modulation, and equalizers, and the power requirements of each of these components can vary based on the coding strategy. While recent works have started to address fundamental limits [26] and modeling [27] of power consumption of system blocks from a mixed-signal circuit design perspective, tradeoffs with code choice of these components remain relatively unexplored. Hence, while analyzing decoding power is a start, and is especially relevant when decoding power is the dominant sink of energy, other system-level tradeoffs should be addressed in future work. It is also of great interest to understand tradeoffs at a network level (e.g., see [20]), where multiple transmitting-receiving pairs are communicating in a shared wireless medium. In such situations, one cannot simply increase transmit power to reduce decoding power: the resulting interference to other users needs to be accounted for as well.

The remainder of the paper is organized as follows. Section II states the assumptions and notation used in the paper. Sections II-C to II-F introduce theoretical models of VLSI circuits and decoding energy. These models are analyzed in Sections III and IV respectively, in the context of the question illustrated in Fig. 1(a). Section V discusses detailed physical models of decoding implementations, in the context of the question illustrated in Fig. 1(b). Section VI concludes the paper.

3For example, the resolution of ADCs used at the receiver may vary with the code choice by virtue of the fact that changing the rate of the code may require a change in signaling constellation (when channel bandwidth and data-rate are fixed).
II. SYSTEM AND VLSI MODELS FOR ASYMMETRIC ANALYSIS

Throughout this paper, we rely on the family of Bachmann-Landau notation [38] (i.e. “big-O” notation). For any two functions \( f(x) \) and \( g(x) \) defined on some subset of \( \mathbb{R} \), asymptotically (as \( x \to \infty \)), \( f(x) = \mathcal{O}(g(x)) \) if \( |f(x)| \leq c_2|g(x)| \); \( f(x) = \Omega(g(x)) \) if \( |f(x)| \geq c_1|g(x)| \); and \( f(x) = \Theta(g(x)) \) if \( c_3|g(x)| \leq |f(x)| \leq c_4|g(x)| \) for some positive real-valued constants \( c_1, c_2, c_3, c_4 \). All logarithm functions \( \log(\cdot) \) are natural logarithms unless explicitly stated otherwise.

A. Communication channel model

We assume the communication between transmitter and receiver takes place over an AWGN channel with flat-fading. The transmission strategy uses BPSK modulation, and a \((d_v, d_c)\)-regular LDPC code of design rate \( R = 1 - \frac{d_v}{d_c} \) [48] (which is assumed to equal the code rate). The blocklength of the code is denoted by \( n \), and the number of source bits is denoted by \( k = nR \). We further assume that \( d_v \geq 4 \) for reasons that have to do with how fast the error-probability decreases with the number of decoding iterations, which will become clear in Lemma 2. The decoder performs a hard-decision on the observed channel outputs before starting the decoding process, thereby first recovering noisy codeword bits transmitted through a Binary Symmetric Channel (BSC) of flip probability \( p_0 = Q(\sqrt{2SNR}) \). Here, \( SNR \) is the received signal-to-noise ratio and \( Q(\cdot) \) is the right-tail cumulative density function of the standard normal distribution, \( Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{u^2}{2}} du \). The transmit power \( P_T \) is assumed to be proportional to \( SNR \), modeling fixed distance and fixed fade-coefficient wireless communication. It follows via the Mill’s ratio inequalities [32], that \( p_0 = \Theta \left( \frac{e^{-\eta P_T}}{\sqrt{4\pi\eta P_T}} \right) \) for some constant \( \eta > 0 \), where \( SNR = \eta P_T \).

B. Decoding algorithm assumptions

We consider two decoding algorithms which were originally proposed in Gallager’s thesis [31], and are now called [48] “Gallager A” and “Gallager B”. Both algorithms are one-bit message-passing algorithms which were analyzed in [31] under the assumption that any two messages being passed in a given iteration are independent. For this assumption to hold, decoding can run only for the number of algorithmic iterations until which the decoding neighborhood around each variable-node is locally tree-like. In this work, we assume the decoder operates under this constraint. Any larger number of iterations introduces correlations in messages, rendering density-evolution analysis [49] of error-probability invalid\(^4\).

It is known that the girth [44] of the code determines how many such independent algorithmic iterations can be accommodated. This maximum number of independent iterations, which we denote as \( N_{\text{iter}} \), is explicitly \( N_{\text{iter}} = \left\lceil \frac{g-2}{2\ell} \right\rceil \) [44]. Where \( g = 2\ell \) is the girth and \( \ell \geq 2 \) for any LDPC code. Hence, we assume that \( g \geq 6 \) so that at least one decoding iteration is used. The target average bit-error probability is denoted by \( P_e \), decoding power by \( P_{\text{Dec}} \), and “total” power by \( P_{\text{total}} := P_T + P_{\text{Dec}} \). The minimum total power for a strategy is denoted by \( P_{\text{total, min}} \) and the optimizing transmit power by \( P_T^* \).

C. VLSI layout model for decoding implementation

Different models for analyzing the wiring and area complexity of VLSI circuits were introduced several decades ago in computer science, but the most commonly used one is attributed to Thompson [54]. Our model for the LDPC decoding circuit in this paper is an adaptation of Thompson’s model, and it entails the following assumptions:

\(^4\)In practice, decoding is often run for a larger number of iterations because at large blocklengths, error-probability may still decay as the number of iterations increase. However, in that case, density-evolution does not yield the correct error-probability as it will vary based on the specific code construction [16].
1) The VLSI circuit includes processing elements which perform computations and store data, and wires which connect them. The circuit is placed on a square grid of horizontal and vertical wiring tracks of finite width $\lambda > 0$, and contact squares of area $\lambda^2$ at the overlaps of perpendicular tracks.

2) Neighboring parallel tracks are spaced apart by width $\lambda$.

3) Wires carry information bi-directionally and can only cross orthogonally at the contact squares.

4) The layout is drawn in the plane. In other words, the model does not allow for more than two metal layers for routing wires in the manner that modern IC manufacturing processes do.

5) The processing elements in the circuit hold finite memory and are situated at the contact squares of the grid. They connect to wires routed along the grid.

6) Since wires are routed only horizontally and vertically, any single contact has access to a maximum of 4 distinct wires. To accommodate higher-degree nodes, a processing element requiring $x$ external connections (for $x > 4$) can occupy a square of side-length $x\lambda$ on the grid, with wires connecting to any side. No wires pass over the large square.

Hence, $\lambda$ can be thought of as a “toy-model” of the minimum feature-size metric which is often used to describe IC fabrication processes. Consequently, in the body of the paper we refer to this model as Implementation Model ($\lambda$). The decoder circuit is assumed to be implemented in a “fully-parallel” manner [8], i.e. a processing element never acts as more than one vertex in the Tanner graph [53] of the code. In such a decoder, the “processing elements” which Thompson refers to are exactly the variable-nodes and check-nodes of the LDPC code. This means, if we redraw any fully-parallel layout as an undirected graph, where processing nodes are represented as vertices and wires correspond to edges, there exists a graph isomorphism between this layout graph and the Tanner graph of the code. As an example, Fig. 2(a) shows the Tanner graph for a $(7, 4)$-Hamming code and Fig. 2(b) shows a fully-parallel layout of the decoder.

D. Time required for processing

In Sections II-E, II-F we will describe two models of energy consumption for the VLSI circuit. In order to later translate these energy models to power models, we need the time required for computation (the computation time is measured in seconds and is different from $N_{\text{iter}}$). In this section, we characterize the computation time.

The computations are assumed to happen in clocked iterations, with each iteration consisting of two steps. In the first step of each iteration, one-bit information messages (based on the Gallager A or B decoding algorithm) are passed from all

![Figure 2: The Tanner graph (a) of a $(7, 4)$-Hamming code and a fully parallel decoder (b) drawn according to Implementation Model ($\lambda$).](image)
variable-nodes to neighboring check-nodes along connecting wires. In the second step, each check-node computes a function of its inputs (according to the decoding algorithm) and passes a one-bit message back to each neighboring variable-node.

We denote the decoding throughput (number of source bits decoded per second) by $R_{\text{data}}$. Because a batch of $k$ source bits are processed in parallel, the time available for processing is $T_{\text{proc}} = \frac{k}{R_{\text{data}}}$ seconds. The required power for decoding is therefore $P_{\text{Dec}} = \frac{E_{\text{Dec}}}{T_{\text{proc}}} = \frac{E_{\text{Dec}}}{k} R_{\text{data}}$, which is simply the energy per source bit ($E_{\text{Dec}}/k$) multiplied by the (fixed) data rate.

E. Computational node model of decoding power

**Definition 1** (Node Model ($\xi_{\text{node}}$)). The energy consumed in each variable or check node during one decoding iteration is $E_{\text{node}}$. This constant can depend on $\lambda$, $d_v$, $d_c$, and $R_{\text{data}}$. The total number of nodes at the decoder is $n_{\text{nodes}} = n + (n - k) = 2n - k$. The total energy is $E_{\text{Dec}} = E_{\text{node}} n_{\text{nodes}} N_{\text{iter}}$. The decoding power is $P_{\text{Dec}} = E_{\text{node}} \frac{(2n-k)N_{\text{iter}}}{k} R_{\text{data}} = \xi_{\text{node}} N_{\text{iter}}$.

Here, $\xi_{\text{node}}$ is a constant that depends on $\lambda$, $d_v$, and $d_c$. This model assumes that the entirety of the decoding energy is consumed in computational processing nodes, and wires require no energy. In spirit, the model is simply counting the number of operations for the given message-passing decoding algorithm. We note that the node model accounts for the number of iterations because the leakage power in the nodes (which is the power expended even when there is no switching) is not negligible [12]. The next energy model complements the node model by accounting for energy consumed in wiring.

F. Message-passing wire model of decoding power

**Definition 2** (Wire Model ($\xi_{\text{wire}}$)). The decoding energy is $E_{\text{Dec}} = E_{\text{unit \_area}} A_{\text{wires}}$, where $E_{\text{unit \_area}}$ is the energy consumed in each unit-area of a wire and $A_{\text{wires}}$ is the total area occupied by the wires in the circuit. The decoding power is $P_{\text{Dec}} = E_{\text{node}} \frac{A_{\text{wires}}}{k} R_{\text{data}} = \xi_{\text{wire}} A_{\text{wires}}$, where $\xi_{\text{wire}}$ is a constant depending on $\lambda$, $d_v$, $d_c$, and $R_{\text{data}}$.

The wires in the decoder consume power whenever they are “switched,” i.e. when the message along the wire changes its value. The fraction of time the wires need to be switched is called the activity-factor of the wires. If the messages passed along the wires were completely random (i.e. Bernoulli($\frac{1}{2}$)) and independent over time, then each wire would need to be switched half the time (on average). In reality however, as decoding proceeds, the messages tend to stabilize, reducing switching and hence the power consumed in the wires. Thus the model here accounts for the fact that the activity factor is low after the first few iterations by ignoring the energy consumed in the wires after the first iteration.

In practice, either of the two power models can be a better approximation based on the simplicity of the computations and the complexity of the wiring required for the decoding algorithm.

III. ANALYSIS OF NODE MODEL

A. Approximation analysis of Gallager decoding algorithms

We first obtain bounds on the number of algorithmic decoding iterations required to attain a specific error-probability. These results are used in Section III-B to bound the total power under the Node Model.

**Lemma 1.** The required number of independent iterations $N_{\text{iter}}$ to attain error-probability $P_e$ with a Gallager-A decoder is

$$N_{\text{iter}} = \Theta \left( \frac{\log \frac{1}{P_e}}{P_T} \right).$$

5 Switching consumes energy because wires act as capacitors that need to be charged/discharged. If voltage is maintained, little additional energy is spent.
Lemma 2. The required number of independent iterations $N_{\text{iter}}$ to attain error-probability $P_e$ with a Gallager-B decoder with variable node degree $d_v \geq 4$ is given by

$$N_{\text{iter}} = \Theta \left( \frac{\log \frac{1}{P_e}}{\log \left( \frac{d_v - 1}{2} \right)} \right).$$

Proof: See Appendix B. Importantly, this does not hold for $d_v < 4$ because Gallager-A and Gallager-B are equivalent then.

B. Minimum total power of node model

In this section, we investigate the question: as $P_e \to 0$, which decoding algorithms, with associated optimal transmit power, minimize the total power under the Node Model of Section II-E?

1) Gallager-A decoding:

Corollary 1. Under the Node Model, the optimal total power using Gallager-A Decoding is

$$P_{\text{total, min}} = \Theta \left( \sqrt{\log \frac{1}{P_e}} \right) \quad (1)$$

which is achieved by transmit power

$$P_T^* = \Theta \left( \sqrt{\log \frac{1}{P_e}} \right). \quad (2)$$

Proof: Applying Lemma 1 to the Node Model, the power consumed by a Gallager-A decoder is given by

$$P_{\text{Dec}} = \Theta \left( \frac{\log \frac{1}{P_e}}{\eta P_T} \right) \quad (3)$$

and the total power is given by

$$P_{\text{total}} = P_T + P_{\text{Dec}} = P_T + \Theta \left( \frac{\log \frac{1}{P_e}}{P_T} \right). \quad (4)$$

Thus, if $P_T$ is bounded even as $P_e \to 0$, the total power $P_{\text{total, bdd}} P_T = \Theta \left( \log \frac{1}{P_e} \right)$. If instead $P_T$ is allowed to increase unboundedly, optimizing over $P_T$, the minimum total power is

$$P_{\text{total, min}} = \min_{P_T} P_T + \Theta \left( \frac{\log \frac{1}{P_T}}{P_T} \right) = \Theta \left( \sqrt{\log \frac{1}{P_e}} \right), \quad (5)$$

for which the optimal transmit power, $P_T^* = \Theta \left( \sqrt{\log \frac{1}{P_e}} \right).$ \hfill ■

2) Gallager-B decoding:

Corollary 2. Under the Node Model, the optimal total power using Gallager-B Decoding is

$$P_{\text{total, min}} = \Theta \left( \log \log \frac{1}{P_e} \right) \quad (6)$$

which is achieved by transmit power

$$P_T^* = \Theta \left( 1 \right). \quad (7)$$
**Proof:** Using Lemma 2 in the Node Model, the power consumed by a Gallager-B decoder is given by

\[ P_{\text{Dec}} = \Theta \left( \log \frac{1}{P_{e}} \right). \]  

(8)

Thus the total power is given by

\[ P_{\text{total}} = P_T + P_{\text{Dec}} = P_T + \Theta \left( \log \frac{1}{P_{e}} \right). \]  

(9)

The optimal total power as \( P_e \to 0 \) is given by

\[ P_{\text{total}, \text{opt}} = \Theta \left( \log \log \frac{1}{P_{e}} \right). \]  

(10)

In this case the optimal transmit power is constant, even as \( P_e \to 0 \).

\[ \blacksquare \]

C. Comparison with fundamental limits

Can we reduce the power under the Node Model via a better code, or a more sophisticated decoding algorithm? After all, Gallager-B is merely a one-bit message-passing algorithm, and belief-propagation requires the transmission of infinite-length log-likelihoods. It was shown in [20] that under the Node Model and a fully-parallelized decoding implementation such as Implementation Model (\( \lambda \)), the optimal total power is lower bounded by \( \Omega \left( \log \log \frac{1}{P_{e}} \right) \), matching Corollary 2.

In fact, using a code which performs closer to Shannon capacity can even reduce efficiency: if a capacity-approaching LDPC code is used instead of a regular LDPC code, the asymptotic performance under the Gallager-B decoding algorithm matches that of regular LDPCs under Gallager-A. This is because the error-probability decays only exponentially (and not doubly-exponentially) with the number of iterations under Gallager-B decoding if degree-2 variable nodes are present [18], and [51] shows that degree-2 variable nodes are required in order to achieve capacity (the fraction of degree-2 variable nodes required to attain capacity under message-passing decoding is characterized in [51]).

IV. Analysis of Wire Model

We now shift our focus toward analyzing the Wire Model described in Section II-F. We first state some bounds on the blocklength of LDPC codes which we will refer back to at several points in the remainder of the paper.

**Lemma 3.** For a given girth \( g \) of a \((d_v, d_c)\)-LDPC code, a lower bound on the blocklength \( n \) is given by

\[ n \geq [(d_v - 1)(d_c - 1)]^{\frac{g-2}{4}}, \]

and an upper bound on the blocklength is given by

\[ n \leq 2(d_v + d_c)d_v d_c (2d_v d_c + 1)^{\frac{g}{4}}. \]  

(11)

**Proof:** For the lower bound, see [35, Appendix I], and for the upper bound, see [35, Claim 2].

\[ \blacksquare \]

A. Bounds on wiring area of decoders

To make use of the energy model of Section II-F, we must characterize the total wiring area of the decoder. Techniques for obtaining upper and lower bounds on the total wire area for different computations were explored in often-forgotten computer
science works [54], [41], [43], [40]. In the following subsections, we introduce some graph theory concepts and we directly apply them to obtain bounds on the wiring of an LDPC decoder.

1) Lower bound on wiring area: We first give the trivial lower bound on the wiring area of the decoder for any regular LDPC code under Implementation Model ($\lambda$).

**Lemma 4.** For a $(d_v, d_c)$-regular LDPC code of blocklength $n$, the wiring area $A_{\text{wires}}$ under Implementation Model ($\lambda$) is

$$A_{\text{wires}} = \Omega(n).$$

**Proof:** There are $nd_v$ wires. Each wire has width $\lambda > 0$ and positive length (no two wires overlap completely). □

In his thesis [40], Leighton utilizes the crossing number (a property first defined by Turán [56]) of a graph as a tool for obtaining lower bounds on the wiring area of circuits. We use the following two definitions to introduce this property.

**Definition 3 (Graph Drawing).** A drawing of a graph $G$ is a representation of $G$ in the plane such that each vertex of $G$ is represented by a distinct point and each edge is represented by a distinct continuous arc connecting the endpoints, which does not cross itself. We assume that in any drawing, no edge passes through vertices other than its endpoints and no edges overlap others for any nonzero length (i.e. anything other than at discrete points where they might cross).

**Definition 4 (Crossing Number).** The crossing number of a graph $G$, $cr(G)$, is the minimum number of edge-crossings over all possible drawings of $G$. An edge-crossing is any point in the plane other than a vertex of $G$ where a pair of edges intersects.

Crossing numbers continue to be of interest to combinatorialists and graph-theorists, and many difficult problems on finding exact crossing numbers or bounds for various families of graphs remain open [46].

It follows that for any graph $G$, the wiring area ($A_{\text{wires}}$) of the corresponding circuit under Implementation Model ($\lambda$) is lower bounded as $A_{\text{wires}} \geq \lambda^2 cr(G)$. This is due to the fact that any VLSI layout of the type described in Section II-C is isomorphic to a drawing of $G$ in the sense of Definition 3. Therefore the minimum number of wire crossings of any layout of $G$ is $cr(G)$. Since every crossing has area $\lambda^2$, the inequality follows. From this, any lower bound on the crossing number of a computation graph also yields a lower bound on its circuit wiring area. The first lower bound on the crossing number of a general graph (often called the “crossing number lemma” [3]) was proved independently by Ajtai et al. [17] and Leighton [40]. In this paper, we make use of the following improvement from [11].

**Theorem 1 (Pach, Spencer, Tóth [11]).** Let $G = \{V, E\}$ be a graph with girth $g > 2\ell$ and $|E| \geq 4|V|$. Then $cr(G)$ satisfies

$$cr(G) \geq k_\ell \frac{|E|^\ell+2}{|V|^\ell+1},$$

where $k_\ell > 0$ is a constant dependent on $\ell$.

We now obtain lower bounds on wiring area based on the minimum number of independent iterations the code allows for.

**Lemma 5 (Crossing Number Lower Bound on $A_{\text{wires}}$).** For any $(d_v, d_c)$-regular LDPC code $C$ that allows for at least $N_{\text{iter}}$ independent decoding iterations with a decoder $D$ implemented in Implementation Model ($\lambda$), the decoder wiring area $A_{\text{wires}}$ is lower bounded in the order of $N_{\text{iter}}$ as

$$A_{\text{wires}} = \Omega(e^{\gamma N_{\text{iter}}}),$$
where $\gamma \in [\log ((d_v - 1)(d_c - 1)), 3 \log(2d_v d_c + 1)]$. If we further assume $d_v d_c \geq 4(d_v + d_c)$, we can tighten this to

$$A_{\text{wires}} = \Omega \left( e^{N_{\text{iter}}(\gamma + 2 \log d_v + 2 \log d_c - 2 \log (d_v + d_c))} \right).$$

**Proof:** Let $C$ be a $(d_v, d_c)$-regular LDPC code that allows for at least $N_{\text{iter}}$ independent decoding iterations. We know that the minimum girth, $g_{\text{min}}$ of $C$ must satisfy

$$g_{\text{min}} \geq 4N_{\text{iter}} - 2.$$

From Lemma 3, the blocklength $n$ of the code $C$ can be expressed in the order of $N_{\text{iter}}$ as

$$n = \Omega \left( e^{\gamma N_{\text{iter}}} \right),$$

where $\gamma \in [\log ((d_v - 1)(d_c - 1)), 3 \log(2d_v d_c + 1)]$. And from Lemma 4 we then have

$$A_{\text{wires}} = \Omega \left( e^{\gamma N_{\text{iter}}} \right).$$

(12)

Now, assume $d_v d_c \geq 4(d_v + d_c)$. Let $V_C, E_C$ denote the sets of vertices and edges in the tanner graph of $C$. The sizes are

$$|E_C| = nd_v,$$

$$|V_C| = n \left( 1 + \frac{d_v}{d_c} \right).$$

We can then carry out the following manipulations

$$d_v d_c \geq 4(d_v + d_c)$$

$$\Rightarrow nd_v \geq 4n \left( 1 + \frac{d_v}{d_c} \right).$$

Hence, $|E_C| \geq 4|V_C|$. Also, using the fact that $g_{\text{min}} > 4N_{\text{iter}} - 4$ we can apply Theorem 1 and write

$$A_{\text{wires}} = \Omega \left( e^{\gamma N_{\text{iter}}} \right).$$

(13)

How loose can (13) become? If $d_v d_c \geq 4(d_v + d_c)$, the loosest the bound can become is $A_{\text{wires}} = \Omega \left( e^{N_{\text{iter}}(\gamma + 2 \log 4)} \right)$.

2) **Upper bound on wiring area:** Since the total circuit area is always an upper bound on the area occupied by wires, we use an upper bound on the circuit area to obtain the following upper bound on the wiring area based on the maximum number of independent iterations that the code allows for.

**Lemma 6** (Upper bound on $A_{\text{wires}}$). For any $(d_v, d_c)$-regular LDPC code $C$, that allows for at most $N_{\text{iter}}$ independent decoding iterations, the decoder wiring area $A_{\text{wires}}$ is upper bounded in the order of $N_{\text{iter}}$ as

$$A_{\text{wires}} = O \left( e^{2\gamma N_{\text{iter}}} \right),$$

where $\gamma \in [\log ((d_v - 1)(d_c - 1)), 3 \log(2d_v d_c + 1)]$. 

**Proof:** Let $C$ be a $(d_v, d_c)$-regular LDPC code that allows for at most $N_{\text{iter}}$ independent decoding iterations. We know that the maximum girth, $g_{\text{max}}$ of $C$ must satisfy

$$g_{\text{max}} \leq 4N_{\text{iter}} + 6.$$  

From Lemma 3, the blocklength of any such code can be upper bounded in the order of $N_{\text{iter}}$ as

$$n = O\left(e^{3\gamma N_{\text{iter}}}\right), \quad (14)$$

where $\gamma \in \lfloor \log \left(\left((d_v - 1)(d_c - 1)\right)\right), 3 \log(2d_vd_c + 1)\rfloor$. Then, consider a “collinear” VLSI layout of the Tanner graph of $C$ which satisfies all the assumptions described in Section II-C. Arrange all variable-nodes and check-nodes in the graph along a horizontal line, leaving $\lambda$ spacing between consecutive nodes. The total length of this arrangement is then $O(n)$. Allocate a unique horizontal wiring track for each of the $nd_v$ edges in the Tanner graph. Then, every connection in the graph can be made with two vertical wires (one from each endpoint) which connect to the opposite ends of the dedicated horizontal track. The total height of this layout is then $O(n)$, and the total area is $O(n^2)$. An example collinear layout is given in Fig. 3. Substituting the result from (14) for $n$, we obtain the desired bound.

**B. Minimum total power of wire model**

We now present analogues of results in Section III-B, where we instead consider decoding power described by the Wire Model of Section II-F. We translate the wiring area bounds of Section IV-A to power bounds.

**Theorem 2** (Asymptotic bounds on $P_{\text{wires}}$). Under implementation Model ($\lambda$) and Wire Model ($\zeta_{\text{tech}}$), the decoding power...
for any regular LDPC code that is decoded for exactly $N_{\text{iter}}$ iterations is bounded as

$$P_{\text{wires}} = \Omega(e^{\gamma N_{\text{iter}}})$$

$$P_{\text{wires}} = O(e^{2\gamma N_{\text{iter}}})$$

where $\gamma \in \left[\log ((d_v - 1)(d_c - 1)), 3\log(2d_vd_c + 1)\right]$.

Proof: The result is a straightforward conclusion from Lemma 3, Lemma 5, and Lemma 6.

1) Gallager-A decoding:

Theorem 3. The optimal total power under Gallager-A decoding in the Wire Model ($\xi_{\text{tech}}$) for any regular LDPC code to achieve error-probability $P_e$ is bounded as

$$P_{\text{total,min}} = \Theta\left(\frac{\gamma}{\eta} \log \frac{1}{P_e}\right)$$

Further, if $P_T$ is bounded even as $P_e \to 0$, then the required power diverges as a polynomial in $\frac{1}{P_e}$, which is exponentially worse than using uncoded transmission.

Proof: See Appendix C.

2) Gallager-B decoding:

Theorem 4. The optimal total power under Gallager-B decoding in the Wire Model ($\xi_{\text{tech}}$) for any regular LDPC code to achieve error-probability $P_e$ is lower bounded as

$$P_{\text{total,min}} = \Omega\left(\log^2 \frac{1}{P_e}\right)$$

and upper bounded as

$$P_{\text{total,min}} = O\left(\log^6 \frac{1}{P_e}\right).$$

Further, if $P_T$ is bounded even as $P_e \to 0$, then $P_{\text{total,bdd}} P_T = \Omega\left(\log^2 \frac{1}{P_e}\right)$.

Proof: See Appendix D.

C. Comparison with fundamental limits

In [21], using another Wire Model, it is show that the total power required for any message passing decoding algorithm is fundamentally lower bounded by $\Omega\left(\log^2 \frac{1}{P_e}\right)$. In comparison, Theorem 4 shows that the total power for regular LDPCs for Gallager-B decoding diverges to infinity at least as fast as $\log^2 \frac{1}{P_e}$. The Wire Model of [21] and the one here have a small difference: while the power is assumed to be proportional to $A_{\text{wires}} N_{\text{iter}}$ in [21], here it is assumed to be simply proportional to $A_{\text{wires}}$. This difference in modeling is not significant for Gallager-B decoding: the number of iterations, $N_{\text{iter}}$ is $\Theta\left(\log \frac{1}{nP_{\text{eff}}}\right)$. Thus even if we adopted the Wire Model of [21] in this paper, the introduced discrepancy would be bounded by a multiplicative factor of $\log \log \frac{1}{P_e}$, which is small relative to the fractional powers of $\log \frac{1}{P_e}$ in play here.

Theorem 3 shows that coding can be useful: the Gallager-A algorithm can outperform uncoded transmission in total-power in the order sense. However, the gap in total power between the two is merely a multiplicative factor of $\log \log \frac{1}{P_e}$. While Theorem 4 shows that Gallager-B decoding increases the relative advantage of coding to a fractional power of $\log \frac{1}{P_e}$, the
difference between the upper bound and the power for uncoded transmission is minuscule. The exponent of \( \log \frac{1}{P_e} \) in the upper bound is an increasing function of both \( d_v \) and \( d_c \), approaching 1 as either gets large. Since Gallager-B decoding requires \( d_v \geq 4 \), the smallest exponent for regular codes occurs when \( d_v = 4 \) and \( d_c = 5 \). The numerical value of the exponent for these degrees is \( \approx 0.98 \), which suggests little order sense improvement over uncoded transmission. Hence, the wiring area at the decoder (particularly, how much better it can be than the bound of Lemma 6) is crucial in determining how much can be gained by using Gallager-B decoding instead of uncoded transmission.

D. The need for new code constructions

In fact, wiring complexity of the code is so crucial that inefficient constructions can cause the lower bound and upper bound of Theorem 4 to match. Recent work of Blake and Kschischang [6] has shown the following theorem, which holds for randomly generated regular-LDPCs and for any sequence of randomly generated LDPC codes which approach capacity.

**Theorem 5** (Blake, Kschischang [6]). Let \( \{C_i\}_{i=1}^{\infty} \) be a sequence of (regular or capacity-approaching) LDPC codes that are generated randomly with blocklengths \( \{n_i\}_{i=1}^{\infty} \) increasing unboundedly. Then \( \mathbb{P} \left( \{A_{\text{wires}} = \Omega(n_i)\} \right) \xrightarrow{\text{i.i.d.}} 1 \) almost surely.

By Theorem 4 then, randomly generated LDPC codes with Gallager-B decoding will have minimum total power that is \( \Theta \left( \log \frac{k}{P_e} \right) \), where \( 0.97 < k < 1 \), providing little order-sense improvement over uncoded transmission. The authors of [6] highlight the fact that Theorem 5 does not rule out the possibility that there may exist a subset of codes with measure 0 asymptotically that has sub-quadratic wiring area. One open problem that was highlighted at the end of Section IV-A2, specifically the problem of proving tight upper bounds on the crossing number for (even some classes of) semi-regular graphs, could potentially provide an answer to this issue. Thus, while randomly generated codes provide a means to approach Shannon-capacity, new code constructions are needed in order to approach total-power-capacity.

V. CIRCUIT SIMULATION EXPLORATION FOR FINITE-LENGTH CODES

At reasonable error probabilities (e.g. \( 10^{-8} \)) and short distances (e.g. less than five meters), asymptotic bounds cannot provide precise answers on which codes to use. For example, consider the following problem, shown graphically in Fig. 1(b).

**Problem 1.** Suppose we want to design a point-to-point communication system that operates over a given channel. We are given a target error-probability \( P_e \), communication distance \( r \), and system data-rate \( R_{\text{data}} \) that the link must operate at. Which code and corresponding decoding algorithm minimize the total (i.e. transmit + decoding) power?

Since the bounds of Sections II-IV are derived as \( P_e \to 0 \), they may not be applicable to most instances of Problem 1. In this section we therefore develop an optimization paradigm for jointly choosing codes and decoding algorithms to answer specific instances of Problem 1. We focus on one-bit Gallager-A [31] and two-bit [25] decoding algorithms, still restricting the number of algorithmic iterations to \( \lfloor \frac{2^n}{2^2} \rfloor \). Because of the effort required in implementing or even simulating a single decoder in hardware, we construct models for power consumed in decoding implementations of different algorithms based on post-layout simulations for simple decoders. The models developed attempt to capture detailed physical aspects (e.g. interconnect lengths and impedance parameters, propagation delays, silicon area, and power-performance tradeoffs) of implementations, in stark contrast with their theoretical counterparts of Sections II-IV. In Section V-C, we use these models to investigate solutions to some instances of Problem 1. We bridge the gap between the asymptotic bounds and the circuit simulations by comparing the bounds with the behavior of the optimal total power as \( r \) is fixed and \( P_e \to 0 \).
A. Note on channels and constellation size

To answer Problem 1 using precise numbers, additional physical assumptions about the channel (e.g. bandwidth, fading, path-loss, temperature, constellation size) are required in comparison to the model of Section II-A. The channel is still assumed to be binary-input AWGN with flat-fading. However, while Section II-A assumes BPSK modulation for all transmissions, due to the introduction of a data-rate constraint and fixed passband bandwidth $W$ (for fair comparison), the constellation size is required to vary based on the code rate. Explicitly, the transmission strategy is assumed to use either BPSK or square-QAM modulation, mapping codeword bits to constellation symbols. We assume the transmitter signals at a rate of $W$ symbols/s and that the minimum square constellation size ($M$) satisfying the system data-rate requirement is chosen: $M$ is always the smallest square of an even integer for which:

$$M \geq 2^{R_{\text{data}}/(W \times R)}.$$  

For calculating transmit power numbers, the thermal noise variance used is $\sigma_z^2 = kT W$, where $k$ is the Boltzmann constant ($1.38 \times 10^{-23} \text{ J/K}$), and $T$ is the temperature. The power is assumed to decay according to a power-law path-loss model $1/r^\alpha$, where $\alpha$ is the path-loss coefficient. The received $E_bN_0$ is obtained as a function of the system and channel parameters:

$$E_bN_0 = \frac{P_T}{kT W (\frac{\lambda}{\lambda})^\alpha \log_2(M)},$$  

where $\lambda$ is the wavelength of transmission at center frequency $f_c$ in Hz ($\lambda = 3 \times 10^8/f_c$). The channel error-probability for BPSK transmissions under this model is:

$$p_0 = Q\left(\sqrt{\frac{2E_b}{N_0}}\right),$$  

and the channel error-probability for $M$-ary square QAM is [7]:

$$p_0 = \sum_{k=1}^{\log_2(\sqrt{M})} \sum_{j=0}^{k-1} (\frac{1}{\sqrt{M}} - \frac{1}{\sqrt{M}})^{j} \times \left(2^{k-1} - \frac{j \times 2^{k-1}}{\sqrt{M}} + \frac{1}{2}\right) \times 2Q\left(\frac{2j+1}{\sqrt{3} \frac{4N_0 \log_2(M)}{(M-1)}}\right).$$  

We specifically mention here that the asymptotic bounds from Sections II-IV remain unchanged, even if we substitute $M$-ary QAM for BPSK as the signaling constellation. This follows from the fact that the RHS of equation (17) is a linear combination of $Q(\cdot)$ functions with argument linearly proportional to $\sqrt{\frac{E_b}{N_0}}$. Hence, even for $M$-ary QAM, $p_0 = \Theta\left(\frac{e^{-\phi \sqrt{P_T}}}{\sqrt{4\pi \sigma \rho T}}\right)$ for some constant $\phi \neq \eta$. Since the difference in this constant only changes the bounds on $N_{\text{iter}}$ in Lemmas 1 and 2 by a multiplicative constant, the analysis of Sections II-IV holds.

For the results presented in Section V-C, we assume the decoding throughput is required to be equal to $R_{\text{data}} = 7$ Gb/s. We assume a channel center frequency of $f_c = 60$ GHz and bandwidth of $W = 7$ GHz. The temperature $T$ is 300 K. The distances considered are much larger than the wavelength of transmission ($\approx 0.5$ cm) so the “far-field approximation” applies.

B. Simulation-based models of LDPC decoders

Given a code, decoding algorithm, and desired data-rate, calculating the required decoding power $P_{\text{Dec}}$ is a difficult task. Even within the family of regular LDPC codes and specified decoding algorithms, the decoder can be implemented in myriad ways. The choice of circuit architecture, implementation technology, and even process-specific transistor options can have a
significant impact on the decoding power [8], [29]. The models we present here are based on simulations of synchronous, fully-parallel decoding architectures in a 90 nm CMOS process with a standard threshold voltage. While this provides insight (see Section V-C), a solution to any instance of Problem 1 requires optimization of $P_T + P_{\text{Dec}}$ over not just super-exponentially many codes and decoding algorithms, but also all decoder architectures, implementation technologies, and process options.

1) Initial post-layout simulations: Our models for large-degree LDPC codes are constructed based on circuit simulations using STMicroelectronics 90 nm standard CMOS process with 7 metal-layers. First, post-layout simulations of one-bit and two-bit decoders for two simple codes were performed. The codes were both (3, 4)-regular LDPC codes (of girth 6 and 8) that were generated randomly using the guess-and-test algorithm in [28] and [30]. The CAD flow used is detailed in Appendix E, and a diagram of the flow is given in Fig. 5(b). The next section details how these results are generalized to larger codes.

2) Physical model of LDPC decoding: Even within our imposed restrictions on the LDPC code degrees, girth, and number of message-passing bits for decoding, constructing a decoding power model that applies to all combinations of these code parameters requires some simplifying assumptions:

1. Decoders are assumed to operate at a fixed supply voltage (chosen to be 0.6V: the minimum voltage used in our initial simulations for which all decoders meet timing constraints).
2. “Minimum-Blocklength” codes (found in [28] for $6 \leq g \leq 10$ or [57] for $g = 12$ and [10] for larger girths) are chosen for a given $g, d_v, d_c$. Hence the blocklength is expressed as a function of these parameters: $n_{g,d_v,d_c}^{(\text{min})}$.

We then model the minimum-achievable clock period $T_{\text{CLK}}$, maximum-achievable clock frequency $f_{\text{CLK}}$, and decoding throughput $R_{\text{Dec}}$ for each decoder as functions of $b, g, d_v, d_c$:

$$T_{\text{CLK}}(b, g, d_v, d_c) = T_{\text{VN}}(b, d_v) + T_{\text{stp}} + T_{\text{CLK-Q}}(b, g, d_v, d_c) + 2T_{\text{wire}}(b, g, d_v, d_c) + T_{\text{CN}}(b, d_c)$$  \hspace{1cm} (18)

$$f_{\text{CLK}}(b, g, d_v, d_c) = T_{\text{CLK}}^{-1}(b, g, d_v, d_c)$$  \hspace{1cm} (19)

$$R_{\text{Dec}}(b, g, d_v, d_c) = \frac{n_{g,d_v,d_c}^{(\text{min})} (1 - \frac{d_v}{d_c})}{\left[ \frac{g-2}{4} \right] \times T_{\text{CLK}}(b, g, d_v, d_c)}.$$  \hspace{1cm} (20)

In (18), $T_{\text{VN}}(\cdot, \cdot)$ and $T_{\text{CN}}(\cdot, \cdot)$ are critical-path delays through individual variable and check nodes respectively, $T_{\text{stp}}$ and $T_{\text{CLK-Q}}(\cdot, \cdot, \cdot, \cdot)$ are the setup and clock-to-Q delays of message-passing flip-flops, and $T_{\text{wire}}(\cdot, \cdot, \cdot, \cdot)$ is the propagation delay through a single message-passing interconnect. In essence, (18) formulates the critical path delay for the circuit by summing up the delays of all logic stages traversed in a single decoding iteration. Details for each component are given in Appendix F.

We model the components of decoding power as

$$P_{\text{Dec}}(b, g, d_v, d_c) = n_{g,d_v,d_c}^{(\text{min})} \times \left[ P_{\text{VN}}(b, d_v) + \frac{d_v}{d_c} P_{\text{CN}}(b, d_c) + 2bd_v \times P_{\text{wire}}(b, g, d_v, d_c) \right].$$  \hspace{1cm} (21)

In (21), $P_{\text{VN}}(\cdot, \cdot)$ and $P_{\text{CN}}(\cdot, \cdot)$ are the power consumed in individual variable and check nodes respectively, and $P_{\text{wire}}(\cdot, \cdot, \cdot, \cdot)$ is the power consumed in a single message-passing interconnect. Note that (21) is a sum of all power consumed in computations and wires of the decoder (the coefficients in (21) count the number of occurrences of each power sink in the decoder). The details of the node power models are given in Appendix G and the details of the wire power models are given in Appendix H.

3) Satisfying the communication data-rate: Fixing the supply voltage for a decoder and using the fastest possible clock speed only allows for a single decoding throughput. Hence, parallelism in order to meet the system data-rate requirement $R_{\text{data}}$ in Problem 1 is also modeled. For example, Fig. 4(a) shows the decoding power vs. decoding throughput for two hypothetical
decoders (‘A’ and ‘B’) at a fixed supply voltage. If an application demands throughput that is twice A’s throughput, two copies of decoder ‘A’ can be used in parallel. Together, they provide twice the throughput, and require twice the power of a single decoder ‘A’. Fig. 4(b) shows the communication system architecture which accommodates this choice. Two separate codewords are transmitted at twice the throughput of a single decoder ‘A’, and a multiplexer at the receiver passes a separate codeword to each of the parallel decoders, which decode the two codewords independently. We therefore allow any integer multiple of points on the curves of Fig. 4(a) to be achieved using this strategy. Though making such a design choice in practice would introduce additional hardware and a slight power consumption overhead, we ignore this cost in our analysis.

What if we want a decoding throughput that is, say, 1.5 times the throughput for a single decoder? In other words, can we interpolate between the points ‘A’ and ‘2A’ in Fig. 4(a)? In cases where integer multiples of a single decoder’s throughput do not exactly reach $R_{\text{data}}$, we first find the minimum number of parallel decoders that when combined exceed the required throughput. Calling this minimum number of decoders $Q$, we then assume that the clock frequency of each of the parallel decoders is slowed down until the overall throughput of the parallel combination is exactly $R_{\text{data}}$. Explicitly, the formula to determine this “underclocked” frequency $f_u$ is:

$$f_u = \frac{\lfloor \frac{g-2}{4} \rfloor \times R_{\text{data}}}{Q \times n_{g,d_e,d_c}^{(\text{min})} \times R}. \tag{22}$$

Because the decoding power is modeled as being linearly proportional to the decoder clock frequency (see Section V-B2 and Appendices G-H), we multiply each individual decoder’s power by the appropriate frequency scaling factor $\kappa = \frac{f_u}{f_{\text{CLK}}(b,g,d_v,d_c)}$, and then multiply the result by the number of parallel decoders to get the total power of the parallel combination:

$$P_{\text{parallel}} = Q \times P_{\text{Dec}}(b,g,d_v,d_c) \times \kappa. \tag{23}$$

Substituting (18) and using the explicit formula above for $f_u$, we replace $\kappa$ and carry out some algebra to obtain:

$$P_{\text{parallel}} = Q \times P_{\text{Dec}}(b,g,d_v,d_c) \times \frac{\lfloor \frac{g-2}{4} \rfloor \times R_{\text{data}}}{Q \times n_{g,d_e,d_c}^{(\text{min})} \times R} \times f_{\text{CLK}}(b,g,d_v,d_c) \tag{24}$$

$$= P_{\text{Dec}}(b,g,d_v,d_c) \times \frac{\lfloor \frac{g-2}{4} \rfloor \times R_{\text{data}}}{n_{g,d_e,d_c}^{(\text{min})} \times R} \times \frac{n_{g,d_e,d_c}^{(\text{min})} \times R}{\lfloor \frac{g-2}{4} \rfloor \times R_{\text{Dec}}(b,g,d_v,d_c)} \tag{25}$$

$$= P_{\text{Dec}}(b,g,d_v,d_c) \times \frac{R_{\text{data}}}{R_{\text{Dec}}(b,g,d_v,d_c)}. \tag{26}$$

Hence, we assume that any (throughput, power) point on the line through ‘A’ and ‘2A’ in Fig. 4(a) can be achieved in this manner (with the obvious exception of points that have negative throughput and power). Therefore, in our analysis in Section V-C, we assume the decoding throughput is exactly $R_{\text{data}}$ and we use the decoding power numbers obtained through this linear interpolation between the modeled points.

4) Comparing different coding strategies: Now, given a subset of codes and decoders, how should a system designer jointly choose a code and decoding algorithm to minimize the total system power? Within the channel model of Section V-A, consider specific instances of Problem 1: let path-loss coefficient $\alpha$ and $R_{\text{data}}$ be fixed. Then, for each choice of $(r, P_e)$, we can compare the required $P_T + P_{\text{Dec}}$ for each combination of code and decoding algorithm modeled in Section V-B2, and find the minimizing combination. A flowchart detailing the steps in this analysis is given in Fig. 5(a).
Fig. 4: (a): Power-throughput tradeoffs for hypothetical decoders. (b): Communication system architecture where two codewords are transmitted at twice the data-rate, and decoded in parallel.

Fig. 5: Flowcharts detailing steps involved in (a): solving an instance of Problem 1 at fixed $r$, $P_e$, and (b): the CAD flow used for simulating decoding power.

C. Example: 60 GHz point-to-point communication

An example plot which shows the minimum achievable $P_T + P_{Dec}$ for different $P_e$ values at a fixed distance $r = 2.8$ m and $\alpha = 3$ is given in Fig. 6. The plot also shows the curve of the optimizing $P_T$ which achieves the minimum $P_T + P_{Dec}$, and the Shannon-limit [50] for the AWGN channel. The horizontal distance between the optimizing $P_T$ curve and the $P_T + P_{Dec}$ curve in Fig. 6 corresponds to the optimizing $P_{Dec}$. As $P_e$ decreases, this decoding power increases, indicating an increase in the total-power-minimizing decoder’s complexity.

1) Connection with asymptotic bounds: To investigate at what $P_e$ values the asymptotic bounds on $P_{\text{wires}}$ from Section IV-B become relevant, an example plot including the upper and lower bounds on the wiring power of the optimizing decoder at a fixed distance $r = 4.4$ m and $\alpha = 3$ is given in Fig. 6(b). This particular distance is chosen for the plot, since the optimizing decoders happen to be 1-bit Gallager-A decoders, allowing for a fair comparison with the results of Theorem 3. Because multiplicative constants are not modeled in the asymptotic analysis, the bounds are scaled by the appropriate constant (the
Fig. 6: A plot of $\log_{10}(P_e)$ vs. minimum achievable $P_T + P_{\text{Dec}}$ for $\alpha = 3$ at a fixed distance of $r = 2.8m$ (a) and $r = 4.4m$ (b). The dashed curves show the optimizing $P_T$ and the Shannon-limit for the channels is also plotted. Wire model bounds for Gallager-A decoding from Theorem 3 are shown in (b).

power consumed in a single interconnect, found using the models of V-B2) to facilitate comparison of the behavior of total power curves in Fig. 6(b). The optimal total power curve sits within the upper and lower bounds on the wire model. The looseness between the upper and lower bounds increases significantly as $P_e \to 0$, which is expected given the multiplicative gap (linear in blocklength) between the two.

2) Joint optimization over code-decoder pairs: Based on the previous sections, it is clear that the form of the total power curve changes with communication distance. For improved understanding, we use two-dimensional contour plots in the $(r, P_e)$ space to evaluate choices of codes and decoders, as suggested by Fig. 1(b). An example is shown in Fig. 7(a), which compares code and decoding algorithm choices for path-loss coefficient $\alpha = 3$. In the top plot, the contours represent regions in the $(r, P_e)$ space where specific combinations minimize $P_T + P_{\text{Dec}}$, and in the bottom plot, regions in the $(r, P_e)$ space are divided based on the value of the minimum $P_T + P_{\text{Dec}}$. The best choices for these instances of Problem 1 turn out to be rate $\frac{1}{2}$ codes. Lower rate codes require large constellations for a 7 Gb/s data-rate, thus requiring large transmit power for the same $p_0$, and higher rate codes require larger decoding power due to increased complexity and size of higher degree nodes. Some tradeoffs between $P_T + P_{\text{Dec}}$ and code and decoder complexity can also be observed in Fig. 7(a): to achieve minimum power, message-passing bits $b$ must increase with $r$ and code girth $g$ must increase with decreasing $P_e$.

How does the inclusion of uncoded transmission as a possible strategy change the picture? Contour plots with uncoded transmission included are given in Fig. 7(b). Comparing Fig. 7(a) with Fig. 7(b), we see that when uncoded transmission is included, it overtakes areas in the $(r, P_e)$ space where $P_e$ is high and $r$ is very small. However, Fig. 7(b) suggests that simple codes and decoders can still outperform uncoded transmission at reasonably low $P_e$ and distances of several meters or more.

VI. CONCLUSIONS AND DISCUSSIONS

We have developed a framework to determine total (transmit + decoding) power for regular LDPC codes. We then compared the derived results with known fundamental limits. If only node power is considered, we show that it is order-optimal to transmit with bounded power with the Gallager-B decoding algorithm. Further, the minimum total power increases as $\log \log \frac{1}{P_e}$ which matches the fundamental limits on this component of power from [21]. However, if wiring power is considered, it turns out that transmitting at bounded transmit power is suboptimal, and in fact even worse than just using uncoded transmission. This suggests that measuring complexity of decoding by merely counting the number of operations (e.g. [55], [37], [4]) is insufficient
for understanding system-level power consumption.

We then asked the question as to when coding can indeed be useful in achieving an order-sense improvement in total power over uncoded transmission. It turns out that in the wire model, achieving any order-sense advantage over uncoded transmission requires that both transmit and decoding power diverge to infinity as $P_e \to 0$, which calls into question the assumption that one should approach Shannon capacity in order to minimize power.

Our work highlights an important question (see Section IV-D) that has received little attention in the coding-theoretic literature: design of codes that have good performance while maintaining small wiring area (see [45], [19], [5], [6], [35]). For wire power consumption, there is a significant gap between the bounds on power consumed by LDPC decoders derived here, and the fundamental limits derived in [21]. Nevertheless, it is entirely possible that regular LDPC codes do not achieve these fundamental limits, even in the order sense. It is therefore important to investigate wiring complexity and power consumption of other coding families, such as polar codes [4], and convolutional codes [42].

The simulation-based estimates of decoding power presented in Section V suggest that coding can be useful for minimizing total power, even in short-distance settings. For instance, they predict that simple regular-LDPC codes can achieve lower error probabilities than uncoded transmission in short-distance settings while still consuming the same total power, even at distances as low as 2 meters. However, in these regimes, it is possible that “classical” coding families (e.g. Hamming or Reed-Solomon codes [42]) might be even more efficient, and hence they need to be examined as well.
ACKNOWLEDGEMENTS

This work was supported in part by the Interconnect Focus Center of the Semiconductor Research Corporation, and in part by the NSF Center for Science of Information (CSoI) NSF CCF-0939370, as well as a seed grant on “Understanding Information-Energy Interactions” from the NSF CSoI. Grover’s research supported by NSF CCF-1350314 (NSF CAREER), and NSF ECCS-1343324 (NSF EARS) awards. We thank Anant Sahai, Subhasish Mitra, Yang Wen for stimulating discussions, Jose Moura for code constructions that we started our simulations with, and Jianying Luo for comments. We thank the students, faculty, staff and sponsors of the Berkeley Wireless Research Center and Wireless Foundations group at Berkeley. In particular, Brian Richards assisted with the simulation flow and Tsung-Te Liu advised us on modeling circuits. We acknowledge STMicroelectronics for the technology information used in our simulations.

APPENDIX A

PROOF OF LEMMA 1

Proof of Lemma 1: From [48, Eqn. (6)], the error-probability in the \(i\)th decoding iteration, denoted by \(p_i\), follows

\[ p_i = p_0 - p_0 \left[ 1 + \frac{(1 - 2p_{i-1})^{d_c-1}}{2} \right]^{d_c-1} + (1 - p_0) \left[ 1 - \frac{(1 - 2p_{i-1})^{d_c-1}}{2} \right]^{d_c-1}. \]  

(27)

Since the RHS of (27) is differentiable wrt \(p_{i-1}\), by Taylor’s Theorem \(\exists\) a real function \(R_1(x)\) with \(\lim_{x \to 0} R_1(x) = 0\) s.t:

\[ p_i = p_0(d_v - 1)(d_c - 1)p_{i-1} + R_1(p_{i-1}). \]  

(28)

The RHS of (28) is the first-order MacLaurin expansion of \(p_i\). Further, the remainder term \(R_1(p_{i-1})\) has Lagrange form:

\[ R_1(p_{i-1}) = \frac{1}{2} \frac{d^2p_i(x^*)}{dp_{i-1}^2} p_{i-1}^2, \]  

(29)

where \(x^* \in (0, p_{i-1})\). Calculating the second derivative of \(p_i\) wrt \(p_{i-1}\), we find that it is:

\[ \frac{d^2p_i}{dp_{i-1}^2} = -\frac{p_0}{2d_v} (d_v - 1)(d_v - 2)(d_c - 1)^2 \left[ 1 + (1 - 2p_{i-1})^{d_c-1} \right]^{d_c-3} (1 - 2p_{i-1})^{2(d_c-2)} \]

\[ - \frac{p_0}{2d_v} (d_v - 1)(d_v - 1)(d_c - 2) \left[ 1 + (1 - 2p_{i-1})^{d_c-1} \right]^{d_c-2} (1 - 2p_{i-1})^{d_c-3} \]

\[ + \frac{(1 - p_0)}{2d_v} (d_v - 1)(d_v - 2)(d_c - 1)^2 \left[ 1 - (1 - 2p_{i-1})^{d_c-1} \right]^{d_c-3} (1 - 2p_{i-1})^{2(d_c-2)} \]

\[ - \frac{(1 - p_0)}{2d_v} (d_v - 1)(d_v - 1)(d_c - 2) \left[ 1 - (1 - 2p_{i-1})^{d_c-1} \right]^{d_c-2} (1 - 2p_{i-1})^{d_c-3}. \]  

(30)

It is easily verified that the derivative of the RHS of (30) wrt \(p_{i-1}\) is nonnegative for \(p_{i-1} \in [0, \frac{1}{2}]\). Therefore, the RHS of (30) is maximized at \(p_{i-1} = \frac{1}{2}\), which gives the bound \(\frac{d^2p_i}{dp_{i-1}^2} \leq 0\). Hence, we always have

\[ R_1(p_{i-1}) \leq 0p_{i-1}^2 = 0. \]  

(31)

Plugging (31) into (28) and applying the resulting relation recursively, we obtain an upper bound on \(p_i\):

\[ p_i \leq (p_0(d_v - 1)(d_c - 1))^i \]  

(32)
Similarly, the RHS of (30) is minimized at \( p_{i-1} = 0 \). Plugging back into (29), we obtain the following:

\[
R_1(p_{i-1}) \geq -p_0(d_v - 1)(d_c - 1) \left( \frac{(d_v - 2)(d_c - 1)}{2} + (d_c - 2) \right) p_{i-1}^2 \geq -(p_0(d_v - 1)(d_c - 1))^2 p_{i-1}. \tag{33}
\]

Using (33) in (28), applying the relation recursively, and combining it with the bound of (32):

\[
(p_0(d_v - 1)(d_c - 1))^i (1 - p_0(d_v - 1)(d_c - 1))^i \leq p_i \leq (p_0(d_v - 1)(d_c - 1))^i. \tag{34}
\]

Note the leftmost term of (34) is negative when \( p_0 > \frac{1}{(d_v - 1)(d_c - 1)} \), the stability threshold for Gallager A decoding. However, there is some constant \( K_A > 0 \) s.t. transmit power \( P_T \geq K_A \) implies \( p_0 < \frac{1}{(d_v - 1)(d_c - 1)} \). Notice that if \( \bar{P}_T = \max\{P_T, K_A\} \), 

\( P_T = \Theta(\bar{P}_T) \). Hence, we can assume all sides of (34) are non-negative. Applying the Mill’s ratio bounds [32] on \( p_0 \) terms:

\[
\left( \frac{e^{-\eta \bar{P}_T}(d_v - 1)(d_c - 1)}{\sqrt{2\pi} \left( \sqrt{2\eta \bar{P}_T + \frac{1}{2\sqrt{2\eta \bar{P}_T}}} \right)^i} \right) \leq p_i \leq \left( \frac{e^{-\eta \bar{P}_T}(d_v - 1)(d_c - 1)}{\sqrt{4\pi \eta \bar{P}_T}} \right)^i. \tag{35}
\]

Inverting all sides of (35), taking \( \log(\cdot) \) on all sides, and replacing \( p_i \) by \( P_e \) and \( i \) by \( N_{\text{iter}} \):

\[
N_{\text{iter}} \left( \eta \bar{P}_T + \frac{1}{2} \log \eta \bar{P}_T + c_A \right) \leq \log \frac{1}{P_e} \leq N_{\text{iter}} \left( \eta \bar{P}_T + \frac{1}{2} \log \bar{P}_T - \log(2\eta \bar{P}_T + 1) + c_A^o \right). \tag{36}
\]

For some constants \( c_A^o \) and \( c_A \) depending on \( d_v, d_c, \eta \). The result then follows for large \( P_T \), small \( P_e \). \( \square \)

**APPENDIX B**

**PROOF OF LEMMA 2**

**Proof of Lemma 2:** Using [31, Eqn. 4.15], for \( d_v \) odd the error-probability in the \( i \)th decoding iteration, for Gallager-B decoding follows the recurrence relation

\[
p_i = p_0 - \frac{p_0}{2^{d_v-1}} \sum_{m=\frac{d_v-1}{2}}^{d_v-1} \left[ \frac{d_v - 1}{m} \right] \frac{1}{m} \left[ 1 - (1 - 2p_{i-1})^{d_c-1} \right]^{m} \left[ 1 - (1 - 2p_{i-1})^{d_c-1} \right]^{d_v-1-m} \tag{37}
\]

\[
+ \frac{1 - p_0}{2^{d_v-1}} \sum_{m=\frac{d_v-1}{2}}^{d_v-1} \left[ \frac{d_v - 1}{m} \right] \left[ 1 - (1 - 2p_{i-1})^{d_c-1} \right]^{m} \left[ 1 + (1 - 2p_{i-1})^{d_c-1} \right]^{d_v-1-m}.
\]

Gallager [31, Eqn. 4.17] shows that the \( \frac{d_v}{2} \)th order Maclaurin expansion is

\[
p_i = p_0 \left( \frac{d_v - 1}{d_c - 1} \right)^{d_v - 1} \pi_{-1}^{d_v - 1} + R_B(p_{i-1}). \tag{38}
\]

Here, \( R_B(p_{i-1}) \) takes the form

\[
R_B(p_{i-1}) = \frac{1}{(d_v + 1)!} \frac{d^{d_v + 1}}{dp_{i-1}^{d_v + 1}} \pi_{i-1}^{d_v + 1} \tag{39}
\]

where \( x^* \in (0, p_{i-1}) \). Clearly, the \( \frac{d_v}{2} \)th derivative of \( p_i \) is a sum of finite-degree polynomials in \( p_{i-1} \) with only positive degrees and finite coefficients. Hence, on the interval \((0, p_{i-1})\)

\[
c_1 p_{i-1}^{d_v + 1} \leq R_B(p_{i-1}) \leq c_2 p_{i-1}^{d_v + 1} \tag{40}
\]
for some constants \( c_1, c_2 \). Plugging (40) into (38),

\[
(p_0 \left( \frac{d_v - 1}{d_v - \frac{1}{2}} \right) (d_c - 1) \frac{d_v - 1}{d_v - \frac{1}{2}} + c_1 p_{i-1}) \frac{d_v - 1}{d_v - \frac{1}{2}} \leq p_i \leq \left( p_0 \left( \frac{d_v - 1}{d_v - \frac{1}{2}} \right) (d_c - 1) \frac{d_v - 1}{d_v - \frac{1}{2}} + c_2 p_{i-1} \right) \frac{d_v - 1}{d_v - \frac{1}{2}} .
\]

(41)

Hence we immediately have

\[
p_i = \Theta \left( p_0 \left( \frac{d_v - 1}{d_v - \frac{1}{2}} \right) (d_c - 1) \frac{d_v - 1}{d_v - \frac{1}{2}} \right).
\]

(42)

Applying (42) recursively

\[
p_i = \Theta \left( \left( p_0 \left( \frac{d_v - 1}{d_v - \frac{1}{2}} \right) (d_c - 1) \frac{d_v - 1}{d_v - \frac{1}{2}} \right)^{1+ \left( \frac{d_v - 1}{d_v - \frac{1}{2}} \right) + \left( \frac{d_v - 1}{d_v - \frac{1}{2}} \right)^2} \times \left( p_0 (d_c - 1) \right)^{1+ \left( \frac{d_v - 1}{d_v - \frac{1}{2}} \right) + \left( \frac{d_v - 1}{d_v - \frac{1}{2}} \right)^2} \right).
\]

(43)

Therefore, using \( f \lesssim g \) and \( f \gtrsim g \) as shorthand for \( f = O(g) \) and \( f = \Omega(g) \) respectively,

\[
p_i (d_v, g) \lesssim \left( p_0 \times c^i_B \right)^{1+ \left( \frac{d_v - 1}{d_v - \frac{1}{2}} \right) + \left( \frac{d_v - 1}{d_v - \frac{1}{2}} \right)^2} \times \frac{1}{c^i_B}.
\]

(44)

The constant in (43) is

\[
c^i_B = \left( (d_v - 1) (d_c - 1) \right)^{\frac{d_v - 1}{d_v - \frac{1}{2}}},
\]

and the constant in (44) is

\[
c^u_B = \left( (d_v - 1) (d_c - 1) \right)^{-\frac{d_v - 1}{d_v - \frac{1}{2}}}
\]

Inverting both sides of (43) and (44), taking \( \log(\cdot) \) on both sides, and replacing \( i \) by \( N_{iter} \) and \( p_i \) by \( P_c \)

\[
\left( \frac{d_v - 1}{2} \right)^{N_{iter}} \times \left( \log \frac{1}{p_0} - \log c^u_B \right) \lesssim \log \frac{1}{P_c} \lesssim \frac{1 - \left( \frac{d_v - 1}{2} \right)^{N_{iter}+1}}{1 - \left( \frac{d_v - 1}{2} \right)} \times \left( \log \frac{1}{p_0} - \log c^i_B \right) + \log c^i_B.
\]

(45)

Because \( N_{iter} \geq 0 \), the sum of the \( \log c^i_B \) terms of opposite sign on the RHS of (45) will be non-positive. We can therefore loosen the upper bound of (45) by canceling them

\[
\left( \frac{d_v - 1}{2} \right)^{N_{iter}} \times \left( \log \frac{1}{p_0} - \log c^u_B \right) \lesssim \log \frac{1}{P_c} \lesssim \frac{1 - \left( \frac{d_v - 1}{2} \right)^{N_{iter}+1}}{1 - \left( \frac{d_v - 1}{2} \right)} \times \left( \log \frac{1}{p_0} \right).
\]

(46)

We can further loosen the upper bound of (46) by noticing

\[
\frac{1 - \left( \frac{d_v - 1}{2} \right)^{N_{iter}+1}}{1 - \left( \frac{d_v - 1}{2} \right)} \leq \frac{(d_v - 1)^{N_{iter}+1}}{(d_v - \frac{1}{2})} \leq 2 \left( \frac{d_v - 1}{2} \right)^{N_{iter}+1}.
\]

(47)

Plugging in this upper bound back into (46)

\[
\left( \frac{d_v - 1}{2} \right)^{N_{iter}} \times \left( \log \frac{1}{p_0} - \log c^u_B \right) \lesssim \log \frac{1}{P_c} \lesssim \left( \frac{d_v - 1}{2} \right)^{N_{iter}+1} \times \left( 2 \log \frac{1}{p_0} \right).
\]
Replacing \( \log \frac{1}{P_e} \) with \( \log \sqrt{4 \pi \eta P_T} + \eta P_T \) and dividing all sides of (47) by \( P_T \)

\[
\left( \frac{d_v - 1}{2} \right)^{N_{\text{iter}}} \times \left( \eta + \frac{\log \sqrt{4 \pi \eta P_T}}{P_T} \right) \lesssim \frac{\log \frac{1}{P_T}}{P_T} \lesssim \left( \frac{d_v - 1}{2} \right)^{N_{\text{iter}} + 1} \times \left( \frac{\log 4 \pi \eta P_T}{P_T} + 2 \eta \right).
\] (48)

Taking \( \log(\cdot) \) on all sides of (48) we obtain

\[
N_{\text{iter}} \log \left( \frac{d_v - 1}{2} \right) + \log \left( \eta + \frac{\log \sqrt{4 \pi \eta P_T}}{P_T} \right) \lesssim \log \frac{\log \frac{1}{P_T}}{P_T} \lesssim (N_{\text{iter}} + 1) \log \left( \frac{d_v - 1}{2} \right) + \log \left( \frac{\log 4 \pi \eta P_T}{P_T} + 2 \eta \right).
\] (49)

And dividing all sides of (49) by \( \log \left( \frac{d_v - 1}{2} \right) \), we obtain

\[
N_{\text{iter}} + \frac{\log \left( \eta + \frac{\log \sqrt{4 \pi \eta P_T}}{P_T} \right)}{\log \left( \frac{d_v - 1}{2} \right)} \lesssim \frac{\log \frac{\log \frac{1}{P_T}}{P_T}}{\log \left( \frac{d_v - 1}{2} \right)} \lesssim N_{\text{iter}} + 1 + \frac{\log \left( \frac{\log 4 \pi \eta P_T}{P_T} + 2 \eta \right)}{\log \left( \frac{d_v - 1}{2} \right)}.
\] (50)

The result now follows for small \( P_e \) and large \( P_T \). A parallel analysis can be done for even values of \( d_v \) following the same steps as above with the Maclaurin expansion [31, Eqn. 4.18].

**Appendix C**

**Proof of Theorem 3**

**Proof of Theorem 3:** Let \( N_{\text{iter}}^{(P_e)} \) denote the number of independent Gallager-A decoding iterations required for a given regular LDPC code to achieve error-probability \( P_e \). Via Theorem 2 and Lemma 1, the total power is lower bounded by

\[
P_{\text{total}} = P_T + P_{\text{wires}} = P_T + \Omega \left( e^{\gamma N_{\text{iter}}^{(P_e)}} \right) = P_T + \Omega \left( e^{\frac{\log \left( \frac{1}{P_e} \right)}{\eta P_T}} \right) \left( \frac{1}{P_e} \right)^{\frac{2 \gamma}{\eta P_T}}.
\] (51)

Similarly, there is an upper bound for the total power

\[
P_{\text{total}} = O \left( P_T + \left( \frac{1}{P_e} \right)^{\frac{2 \gamma}{\eta P_T}} \right).
\] (52)

It follows that if \( P_T \) is not increased unboundedly as \( P_e \to 0 \), then the required decoding power diverges as a power of \( \frac{1}{P_e} \), which is exponentially larger than the power required for uncoded transmission. In order to find the optimizing transmit power, let \( L_{P_e}(P_T) \) denote the function in the \( \Omega \) expression of (51) and let \( U_{P_e}(P_T) \) denote the function in the \( O \) expression of (52):

\[
L_{P_e}(P_T) = P_T + \left( \frac{1}{P_e} \right)^{\frac{2 \gamma}{\eta P_T}}
\] (53)

\[
U_{P_e}(P_T) = P_T + \left( \frac{1}{P_e} \right)^{\frac{3 \gamma}{\eta P_T}}.
\] (54)

We start by analyzing the lower bound. To find the \( P_T \) which minimizes \( L_{P_e} \), we differentiate \( L_{P_e} \) and set it to 0

\[
\frac{dL_{P_e}}{dP_T} = 1 - e^{\frac{\gamma \log \frac{1}{P_T}}{\eta P_T}} \frac{\gamma \log \frac{1}{P_T}}{\eta P_T^2} = 0
\]

\[
\Rightarrow \frac{P_T^2}{\gamma \log \frac{1}{P_T}} = e^{\frac{\gamma \log \frac{1}{P_T}}{\eta P_T}}.
\] (55)
Now, let \( P = \frac{P_T}{\sqrt{\frac{\gamma \log \frac{1}{P_T}}{\eta}}} \). Substituting into (55), we get

\[
\begin{align*}
P^2 &= e^{\frac{\gamma \log \frac{1}{P_T}}{\eta}} \quad (56) \\
2P \log P &= \sqrt{\frac{\gamma \log \frac{1}{P_T}}{\eta}} \quad (57) \\
\log P e^{\log P} &= \frac{1}{2} \sqrt{\frac{\gamma \log \frac{1}{P_T}}{\eta}} \quad (58)
\end{align*}
\]

The positive, real valued solution to (58) is given by the principal branch \( W_0(\cdot) \) of the Lambert W function [23]. Explicitly, when \( x, z \in \mathbb{R}^+ \) satisfy the relation \( x = ze^z \), we say \( z = W_0(x) \). Hence we can write

\[
\begin{align*}
\log P &= W_0 \left( \frac{1}{2} \sqrt{\frac{\gamma \log \frac{1}{P_T}}{\eta}} \right) \quad (59) \\
P &= e^{W_0 \left( \frac{1}{2} \sqrt{\frac{\gamma \log \frac{1}{P_T}}{\eta}} \right)} \quad (60) \\
&= \frac{1}{2} \sqrt{\frac{\gamma \log \frac{1}{P_T}}{\eta}} \\
&= W_0 \left( \frac{1}{2} \sqrt{\frac{\gamma \log \frac{1}{P_T}}{\eta}} \right) \quad (61)
\end{align*}
\]

Rewriting \( P \) in terms of \( P_T \) we find the optimizing transmit power

\[
P_T^* = \frac{\gamma \log \frac{1}{P_T}}{\eta} \quad (62)
\]

The first two terms in the asymptotic expansion of the \( W_0(x) \) as \( x \to \infty \) are \( \log(x) - \log \log(x) \) [23]. In fact, \( \forall x \geq e \) [36]:

\[
\log(x) - \log \log(x) \leq W_0(x) \leq \log(x) - \frac{1}{2} \log \log(x) \quad (63)
\]

Using (63) in (62), the optimizing transmit power is bounded as

\[
P_T^* = \Omega \left( \frac{\frac{7}{4} \log \frac{1}{P_T}}{2 \log \frac{1}{2} \sqrt{\frac{2}{\eta}} + \log \log \frac{1}{P_T} - 2 \log \log \frac{1}{2} \sqrt{\frac{2}{\eta}} - 2 \log \log \log \frac{1}{2} \sqrt{\frac{2}{\eta}} \right) \quad (64)
\]

Plugging back into \( L_{P_T} \) in (53) we obtain

\[
P_{\text{total, min}} = \Omega \left( \frac{\frac{7}{4} \log \frac{1}{P_T}}{2 \log \frac{1}{2} \sqrt{\frac{2}{\eta}} + \log \log \frac{1}{P_T} - 2 \log \log \frac{1}{2} \sqrt{\frac{2}{\eta}} - 2 \log \log \log \frac{1}{2} \sqrt{\frac{2}{\eta}} + \frac{\frac{7}{4} \log \frac{1}{P_T}}{\log \log \frac{1}{P_T}} \right) \quad (64)
\]

Ignoring constants and non-dominating terms in the denominators of (64), we get the lower bound of Theorem 3:

\[
P_{\text{total, min}} = \Omega \left( \frac{\frac{7}{4} \log \frac{1}{P_T}}{\log \log \frac{1}{P_T}} \right)
\]
An identical minimization of $U_{P_e}$ in (53), reveals that the optimizing transmit power is upper bounded as

$$P^*_T = \mathcal{O} \left( \frac{2\gamma}{\eta} \log \frac{1}{P_e} \right).$$

Plugging back into $U_{P_e}$ in (54) we obtain

$$P_{\text{total},\min} = \mathcal{O} \left( \frac{2\gamma}{\eta} \log \frac{1}{P_e} \right). \quad (65)$$

Ignoring constants and non-dominating terms in the denominators of both the transmit and decoding power in (65), we get

$$P_{\text{total},\min} = \mathcal{O} \left( \frac{3\gamma}{\log \log \frac{1}{P_e}} \right),$$

completing the proof.

**APPENDIX D**

**PROOF OF THEOREM 4**

Proof of Theorem 4: Again, let $N^{(P_e)}_{\text{iter}}$ denote the number of independent Gallager-B decoding iterations required for a given regular LDPC code to achieve error-probability $P_e$. The total power is

$$P_{\text{total}} = P_T + P_{\text{wires}} \overset{(a)}{=} P_T + \Omega \left( e^{\gamma N^{(P_e)}_{\text{iter}}} \right) \overset{(b)}{=} P_T + \Omega \left( \left( \frac{1}{P_T} \right)^{\frac{\gamma}{\log \frac{d_v}{d_c}}} \right), \quad (66)$$

where (a) holds from Theorem 2, and (b) holds from Lemma 2. Using the upper bound from Theorem 2, we also know,

$$P_{\text{total}} = P_T + \mathcal{O} \left( \frac{\log \frac{1}{P_T}}{\frac{\log \frac{d_v}{d_c}}{\log \frac{d_v}{d_c}}} \right). \quad (67)$$

Then, considering the bounds on $\gamma$ in Theorem 2, we examine the exponent of $\frac{\log \frac{1}{P_T}}{\frac{\log \frac{d_v}{d_c}}{\log \frac{d_v}{d_c}}}$ in (66):

$$\gamma \geq \log \left( \frac{d_v - 1}{2} \right) \geq \log (d_v - 1) - \log 2 \overset{(a)}{=} \frac{\gamma}{\log \left( \frac{d_v - 1}{2} \right)} > 1 + \frac{\log (d_v - 1)}{\log (d_v - 1)}.$$ 

Because $d_c > d_v$ for any regular LDPC code,

$$1 + \frac{\log (d_c - 1)}{\log (d_v - 1)} > 1 + \frac{\log (d_v - 1)}{\log (d_v - 1)} = 2 \quad (69)$$

Also, because $d_v > 3$ for Gallager-B decoding, the denominator of (68) can be bounded by:

$$0 < 1 - \frac{\log 2}{\log (d_v - 1)} < 1. \quad (70)$$

Thus, using (69) and (70) in (68),

$$\frac{\gamma}{\log \left( \frac{d_v - 1}{2} \right)} > \frac{1 + \frac{\log (d_c - 1)}{\log (d_v - 1)}}{1 - \frac{\log 2}{\log (d_v - 1)}} > \frac{2}{1} = 2.$$
Substituting this lower bound on the exponent back into (66), we obtain:

\[
P_{\text{total}} = \Omega \left( P_T + \left( \frac{\log \frac{1}{P_e}}{P_T} \right)^2 \right). \tag{71}
\]

If the transmit power is bounded even as \( P_e \to 0 \), then the total power (and the decoding power) diverges

\[
P_{\text{total, bdd } P_T} = \Omega \left( \log^2 \frac{1}{P_e} \right). \tag{72}
\]

Differentiating the expressions on the RHS of (71) wrt \( P_T \) and setting to zero, the minimum total power is lower bounded as:

\[
P_{\text{total, min}} = \Omega \left( \log^2 \frac{1}{P_e} \right).
\]

Moving to the upper bound, via Theorem 2, we find that the exponent of \( \frac{\log \frac{1}{P_e}}{P_T} \) in (67) is

\[
2\gamma \leq \frac{6 \log(2d_v d_c + 1)}{\log(d_v - 1) - \log 2}. \tag{73}
\]

Then substituting (73) into (67), we get the bound

\[
P_{\text{total}} = O \left( P_T + \left( \frac{\log \frac{1}{P_e}}{P_T} \right)^{6 \log(2d_v d_c + 1)} \log(d_v - 1) - \log 2 \right). \tag{74}
\]

Differentiating the expressions inside the \( O (\cdot) \) of (74) wrt \( P_T \) and setting to zero, we obtain the upper bound of Theorem 4. \( \Box \)

**APPENDIX E**

**CAD FLOW DETAILS FOR SIMULATION AND POWER ESTIMATION**

Decoders are constructed in a hierarchical manner, where behavioral verilog descriptions of variable and check nodes are mapped to standard cells using logic synthesis and then placed and routed. Then, these nodes are connected according to the parity-check matrix of the codes using place and route, resulting in fully-parallel layouts for the decoders. Post-layout simulation is then performed, using extracted RC parasitics and typical corners for the ST 90nm CMOS process. Three components of the total power consumption (computational nodes, message-passing interconnects, and global clock-tree) are isolated by means of post-layout netlist modification. Starting from a nominal supply voltage of 1.2V down to a minimum of 0.6V, power reduction steps are taken by means of supply voltage and frequency scaling. During this process, no changes to the circuit architecture or transistor sizing in the decoder cells are made. Initial simulations are performed assuming that the received sequence at the output of the channel is all-zero. An assumption of an all-zero transmitted sequence (codeword) is easy to justify in theory [48] due to the linearity of the code, the symmetry of the channel, and the symmetry of the decoders with respect to ones and zeros. In practice however, the received sequence of bits has errors, and could therefore require more switching activity at the decoder than these all-zero simulations indicate. However, since the target error probabilities we consider (see Section V-C) correspond to channel flip probabilities of \( p_0 \approx 10^{-2} \) or less, the expected number of errors in the received sequence is small and the extra switching activity caused by bit flips is ignored. Assuming a random initial state for gates and wires, the power consumption for all valid codewords (that satisfy the parity-check constraints) should be the same. Thus, assuming small \( p_0 \), power consumption for any likely sequence received from the channel is estimated (with slight underestimation) by simply measuring the required power for the all-zero codeword.
APPENDIX F

CIRCUIT MODEL FOR CRITICAL PATH PROPAGATION DELAY

Logic synthesis is used to synthesize variable nodes of degrees $3 \leq d_v \leq 6$ and check nodes of degrees $4 \leq d_c \leq 13$ for message-passing algorithms of bits $1 \leq b \leq 2$. The synthesis tool outputs the area consumed by logic gates in each node and these areas can be expressed as functions of the node degrees and message-passing bits $A_{VN}(b,d_v)$ and $A_{CN}(b,d_c)$. The area occupied by logic gates is assumed to be proportional to the total capacitance of the variable or check node\(^6\).

Assuming the majority of standard cells are designed in the static CMOS logic style, the worst-case pull-up and pull-down resistance of the gates in any critical-path should be close to that of an inverter cell of nominal drive strength. Modeling each gate in the path as a lumped RC circuit, the critical path delay can be estimated by the product of the inverter resistance with the total capacitance of all the gates along the critical path. For the Gallager-A and two-bit decoding algorithms, each variable or check node computes some identical logic functions in parallel [25]. The number of such paths is equal to the node degree. In practice, the synthesis tool could be allowed to construct each path using different gates, but for our modeling efforts, they are forced to be as close to identical as possible\(^7\). Thus, each variable-node has $d_v$ critical paths, and each check-node has $d_c$ critical paths. We assume that the total capacitance is split equally between each critical path. Using the simulated critical path delays through variable nodes ($T_{VN}(1,3)$) and check nodes ($T_{CN}(1,4)$) from the Gallager-A decoders described in Section V-B1 as constants, critical path delays through nodes are modeled as:

$$T_{VN}(b,d_v) = T_{VN}(1,3) \times \frac{3 \times A_{VN}(b,d_v)}{d_v \times A_{VN}(1,3)} \quad (75)$$

$$T_{CN}(b,d_c) = T_{CN}(1,4) \times \frac{4 \times A_{CN}(b,d_c)}{d_c \times A_{CN}(1,4)} \quad (76)$$

Interconnect delay is assumed to be linearly proportional to the length and width of the interconnect. Estimating the length of individual interconnects requires an estimate of the decoder’s physical dimensions. Assuming place and route density\(^8\) for individual VNs and CNs remains constant regardless of node degree, physical area estimates for individual VNs and CNs are obtained by scaling the physical areas of Gallager-A VNs and CNs in Section V-B1 by ratios of logic gate areas. Adding some fixed empty spacing to all sides of each node\(^9\), the total area is estimated as a sum of node areas and empty area. A worst-case estimate for the interconnect lengths ($l_{wire}(b,g,d_v,d_c)$) is obtained by assuming that all interconnects span the Manhattan distance across the chip. Since routing for the decoders in Section V-B1 is done using minimum-width wires on the lower 5 metal layers of a 7-layer CMOS process\(^10\), average minimum width ($w_{min}$), sheet resistance ($R_{eq}$), and capacitance per-unit-length ($C_{unit}$)\(^11\) for these metal layers are calculated using technology information and are assumed as constants. Interconnect delay is then estimated assuming a distributed Elmore model [47]:

$$R_{wire}(b,g,d_v,d_c) = R_{eq} \times \frac{l_{wire}(b,g,d_v,d_c)}{w_{min}} \quad (77)$$

$$C_{wire}(b,g,d_v,d_c) = C_{unit} \times l_{wire}(b,g,d_v,d_c) \quad (78)$$

\(^6\)The delay, power, area, and structure of synthesized logic depend on the constraints and mapping effort given as inputs to the synthesis tool. To allow for a fair comparison between codes of different degrees, we only specify constraints for minimum delay and use equal mapping effort for each node.

\(^7\)During the synthesis process, the tool can be supplied with identical timing constraints for each path to ensure that they are balanced.

\(^8\)Defined here as the ratio of area occupied by logic gates to total physical area.

\(^9\)Outer boundaries of VNs and CNs consist of supply rails, which require minimum spacing that is dictated by the technology library.

\(^10\)Metal layers 6 and 7 are used to construct a global power grid for the entire decoder.

\(^11\)Including parallel-plate, fringing, and interwire components [47].
\[ T_{\text{wire}}(b, g, d_v, d_c) = \frac{1}{2} R_{\text{wire}}(b, g, d_v, d_c) \times C_{\text{wire}}(b, g, d_v, d_c). \] (79)

It is assumed that all decoders operate at the minimum clock period \( T_{\text{CLK}}(b, g, d_v, d_c) \) for which timing is met at the 0.6V supply voltage. This minimum allowable clock period that meets timing in flip-flop based synchronous circuits is bounded by the setup time constraint [47] for each flip-flop. The critical path for a full decoding iteration consists of a CLK-Q delay \( (T_{\text{CLK-Q}}(b, g, d_v, d_c)) \) of a message passing flip-flop, then an interconnect delay, then a CN delay, then another interconnect delay, and finally a VN delay. After this critical path, the arrival of input data at the flip-flop requires an additional margin \( T_{\text{st}} \) before the clock signal arrives. The intrinsic delay of the logic gate before the flip-flop is already accounted for in \( T_{\text{clk}} \), and the remainder of the setup time depends only on the input stage of the flip-flop. It is therefore accounted for by a constant, obtained from simulation. Since CLK-Q delay depends linearly on the capacitive loading at the flip-flop output (which is the capacitance of the message-passing interconnect), it is modeled as the simulated \( T_{\text{CLK-Q}}(1, 6, 3, 4) \) scaled by the estimated ratio of capacitive loading at the VN outputs:

\[ T_{\text{CLK-Q}}(b, g, d_v, d_c) = T_{\text{CLK-Q}}(1, 6, 3, 4) \times \frac{C_{\text{wire}}(b, g, d_v, d_c)}{C_{\text{wire}}(1, 6, 3, 4)}. \] (80)

### Appendix G

**Circuit Model for Computation Power**

The power consumption of a logic gate is approximately proportional to the total capacitance of the gate, but it consists of both dynamic power (which is also proportional to the activity-factor at the input of the gate), and static power (which has no dependence on the activity-factor) [47]. In simulation, the static power consumption of decoders at 0.6V supply is observed to be on average less than 5% of the total decoding power, and it is therefore ignored for these models. The individual VN power \( (P_{\text{VN}}(b, g, d_v, d_c)) \) and CN power \( (P_{\text{CN}}(b, g, d_v, d_c)) \) are therefore assumed to be proportional to the simulation results for Gallager-A nodes \( (P_{\text{VN}}(1, 6, 3, 4)) \) and \( (P_{\text{CN}}(1, 6, 3, 4)) \), scaled by the appropriate ratio of logic gate area to account for different capacitance. Given the small channel error probabilities considered in this work, it is assumed that most errors are corrected within the initial iterations of decoding. Under Gallager-A decoding, it is assumed that messages stabilize within 1 iteration and under two-bit decoding, it is assumed to take 2 iterations since the “strength” bits take an extra iteration to stabilize when an error is corrected. For these first few iterations, worst-case switching activity is assumed, but after these iterations, flip-flop access power (which occurs at every clock cycle) in variable nodes is assumed to be the only dynamic power consumed in the decoder. This gives an effective activity-factor for the rest of the decoder of \( a(b, g) = \frac{b}{1+b} \).

Flip-flops are modeled as a separate component of \( P_{\text{VN}}(b, g, d_v, d_c) \), due to their higher activity factor. The power consumption of a single flip-flop in the Gallager-A decoders \( P_{\text{FF}} \) is isolated, and is then multiplied by the total number of flip-flops in the variable-node \( (b \times d_v) \) to obtain the flip-flop component of the VN power. Since the dynamic power consumption of the decoder is also linearly proportional to the switching frequency, \( P_{\text{VN}}(1, 6, 3, 4) \) and \( P_{\text{CN}}(1, 6, 3, 4) \) are scaled by a ratio of estimated clock frequencies. The final models for the power of a single VN and CN are:

\[ P_{\text{VN}}(b, g, d_v, d_c) = \left[ b \times d_v \times P_{\text{FF}} + (P_{\text{VN}}(1, 6, 3, 4) - 3P_{\text{FF}}) \times \frac{a(b, g)}{a(1, 6)} \times \frac{A_{\text{VN}}(b, d_v)}{A_{\text{VN}}(1, 3)} \right] \times \frac{f_{\text{CLK}}(b, g, d_v, d_c)}{f_{\text{CLK}}(1, 6, 3, 4)} \] (81)

\[ P_{\text{CN}}(b, g, d_v, d_c) = \frac{a(b, g)}{a(1, 6)} \times P_{\text{CN}}(1, 6, 3, 4) \times \frac{A_{\text{CN}}(b, d_v)}{A_{\text{CN}}(1, 4)} \times \frac{f_{\text{CLK}}(b, g, d_v, d_c)}{f_{\text{CLK}}(1, 6, 3, 4)}. \] (82)
APPENDIX H

MODELING POWER CONSUMED IN ON-CHIP INTERCONNECTS

Using the interconnect capacitance estimate \( C_{\text{wire}}(b,g,d_v,d_c) \) and clock frequency \( f_{\text{CLK}}(b,g,d_v,d_c) \) from Appendix F, and the activity-factor \( a(b,g) \) from Appendix G, the power consumed by a single message-passing interconnect \( P_{\text{wire}}(b,g,d_v,d_c) \) in the decoder is modeled using the formula for the dynamic power consumed in interconnects [47]:

\[
P_{\text{wire}}(b,g,d_v,d_c) = a(b,g) \times C_{\text{wire}}(b,g,d_v,d_c) \times (0.6V)^2 \times f_{\text{CLK}}(b,g,d_v,d_c).
\] (83)

REFERENCES

[1] IEEE Std. 802.3an-2006: “Physical Layer and Management Parameters for 10 Gb/s Operation, Type 10GBASE-T,” amendment to IEEE Std. 802.3-2005. IEEE, Sept. 2006.

[2] IEEE Std. 802.11ad-2012: “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Enhancements for Very High Throughput in the 60 GHz Band,” amendment. IEEE, Dec. 2012.

[3] N. Alon and J. H. Spencer. The Probabilistic Method. John Wiley & Sons, 2004.

[4] E. Arikan. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. IEEE Trans. Inf. Theory, 55(7):3051–3073, July 2009.

[5] C. Blake and F. R. Kschischang. Energy consumption of VLSI decoders. Dec. 2014. http://arxiv.org/abs/1412.4130.

[6] C. Blake and F. R. Kschischang. On the energy complexity of LDPC decoder circuits. Feb. 2015. http://arxiv.org/abs/1502.07999.

[7] K. Cho and D. Yoon. On the general BER expression of one and two-dimensional amplitude modulations. IEEE Trans. Comm., 50(7):1074–1080, July 2002.

[8] A. Darabiha et al. Power reduction techniques for LDPC decoders. IEEE J. Solid-State Circuits, 43(8):1835–1845, Aug. 2008.

[9] C. Marcu et al. A 90 nm CMOS low-power 60 GHz transceiver with integrated baseband circuitry. IEEE J. Solid-State Circuits, 44(12):3434–3447, Dec. 2009.

[10] I. E. Bocharova et al. Some voltage graph-based LDPC tailbiting codes with large girth. In Proc. of ISIT, pages 732–736, Aug. 2011.

[11] J. Pach et al. New bounds on crossing numbers. Dis. Comp. Geo., 24(4):623–644, 2000.

[12] K. Ganesan et al. The power cost of over-designing codes. In Proc. of IEEE SiPS, pages 128–133, Oct. 2011.

[13] K. Ganesan et al. Choosing “green” codes by simulation-based modeling of implementations. In Proc. of IEEE GLOBECOM, pages 3286–3292, Dec. 2012.

[14] K. Ganesan et al. How far are LDPC codes from fundamental limits on total power consumption? In 50th Allerton Conf., pages 671–678, Oct. 2012.

[15] K. S. Andrews et al. The development of Turbo and LDPC codes for deep-space applications. Proc. IEEE, 95(11):2142–2156, Nov. 2007.

[16] L. Dolecek et al. Predicting error floors of structured LDPC codes: Deterministic bounds and estimates. IEEE J. Select. Areas Comm., 27(6):908–917, Aug. 2009.

[17] M. Ajtai et al. Crossing-free subgraphs. Ann. Dis. Math., 12:9–12, 1982.

[18] M. Lentmaier et al. An analysis of the block error probability performance of iterative decoding. IEEE Trans. Inf. Theory, 51(11):3834–3855, Nov. 2005.

[19] M. Mohiyuddin et al. Synthesizing interconnect-efficient low density parity check codes. In Proc. of 41st DAC, pages 488–491, June 2004.

[20] P. Grover et al. Towards a communication-theoretic understanding of system-level power consumption. IEEE J. Select. Areas Comm., 29(8):1744–1755, Sept. 2011.

[21] P. Grover et al. Fundamental limits on the power consumption of encoding and decoding. In Proc. of ISIT, pages 2716–2720, July 2012.

[22] P. Youssef-Massaad et al. Impact of Processing Energy on the Capacity of Wireless Channels. In Proc. of ISITA, Oct. 2004.

[23] R. M. Corless et al. A sequence of series for the Lambert W function. Proc. of 1997 ACM Intl. Symp. on Symb. and Alg. Comp., pages 197–204, 1997.

[24] S. Cui et al. Energy Constrained Modulation Optimization. IEEE Trans. Wireless Comm., 4(5):1–11, Sept. 2005.

[25] S. K. Chilappagari et al. Two-bit message passing decoders for LDPC codes over the binary symmetric channel. In Proc. of ISIT, pages 2156–2160, July 2009.

[26] T. Sundstrom et al. Power dissipation bounds for high-speed Nyquist analog-to-digital converters. IEEE Trans. Circuits Syst. I, 56(3):509–518, Mar. 2009.

[27] Y. Li et al. A system level energy model and energy-quality evaluation for integrated transceiver front-ends. IEEE Trans. VLSI, 15(1):90–103, Jan. 2007.
[28] Y. Wang et al. Construction of high-girth QC-LDPC codes. Technical Report TR2008-061, Mitsubishi Electric Research Laboratories, Sept. 2008.
[29] Z. Zhang et al. An efficient 10GBASE-T ethernet LDPC decoder design with low error floors. IEEE J. Solid-State Circuits, 45(4):843–855, April 2010.
[30] M. Fossorier. Quasi-cyclic low-density parity-check codes from circulant permutation matrices. IEEE Trans. Inf. Theory, 50(8):599–618, Aug. 2004.
[31] R. G. Gallager. Low-Density Parity-Check Codes. PhD thesis, MIT, 1960.
[32] R. D. Gordon. Values of mills' ratio of area to bounding ordinate and of the normal probability integral for large values of the argument. Ann. Math. Stat., 12(3):364–366, Sept. 1941.
[33] P. Grover and A. Sahai. Green codes: Energy-efficient short-range communication. In Proc. of ISIT, pages 1178–1182, July 2008.
[34] P. Grover and A. Sahai. Time-division multiplexing for green broadcasting. In Proc. of ISIT, pages 2517–2521, July 2009.
[35] P. Grover and A. Sahai. Fundamental bounds on the interconnect complexity of decoder implementations. In Proc. of 45th CISS, pages 1–6, Mar. 2011.
[36] A. Hoorfar and M. Hassani. Inequalities on the Lambert W function and hyperpower function. J. Inequal. Pure and Appl. Math, 9(2), 2008.
[37] I. Jacobs and E. Berlekamp. A lower bound to the distribution of computation for sequential decoding. IEEE Trans. Inf. Theory, 13(2):167–174, Apr. 1967.
[38] D. Knuth. Art of Computer Programming Volume 1: Fundamental Algorithms. Addison-Wesley Professional, 1997.
[39] T. H. Lee. The design of CMOS radio-frequency integrated circuits. Cambridge university press, 2004.
[40] F. T. Leighton. Layouts for the Shuffle-Exchange Graph and Lower Bound Techniques for VLSI. PhD thesis, MIT, 1982.
[41] C. E. Leiserson. Area-efficient graph layouts (for VLSI). In Proc. of 21st IEEE FOCS, pages 270–281, Oct. 1980.
[42] S. Lin and D. J. Costello. Error control coding. Prentice Hall, 2004.
[43] R. J. Lipton and R. Sedgewick. Lower bounds for VLSI. In Proc. of 13th ACM symp. on FOCS, pages 300–307, May 1981.
[44] J. Lu and J. M. F. Moura. Structured LDPC codes for high-density recording: Large girth and low error floor. IEEE Trans. Magn., 42(2):208–213, 2006.
[45] M. Mansour and N. R. Shanbhag. High-throughput LDPC decoders. IEEE Trans. VLSI, 11:976–996, Dec. 2003.
[46] J. Pach and G. Tóth. Thirteen problems on crossing numbers. Geombinatorics, 9(4):194–207, 2000.
[47] J. M. Rabaei, A. P. Chandrakasan, and B. Nikolic. Digital Integrated Circuits. Prentice Hall, 2002.
[48] T. J. Richardson and R. L. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. IEEE Trans. Inf. Theory, 47(2):599–618, Feb. 2001.
[49] T.J. Richardson and R.L. Urbanke. Modern Coding Theory. Cambridge University Press, 2007.
[50] C. E. Shannon. A mathematical theory of communication. Bell Syst. J., 27:379–423,623–656, July/October 1948.
[51] A. Shokrollahi. Capacity achieving sequences. Codes, Systems, Graphical Models, 123:153–166, 2001.
[52] V. Stojanović. Channel-limited high-speed links: modeling, analysis and design. PhD thesis, Stanford University, 2004.
[53] R. M. Tanner. A recursive approach to low complexity codes. IEEE Trans. Inf. Theory, 27(5):533–547, 1981.
[54] C. D. Thompson. A complexity theory for VLSI. PhD thesis, Carnegie Mellon University, 1980.
[55] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Trans. Inf. Theory, 13(2):260–269, Apr. 1967.
[56] K. Zarankiewicz. On a problem of P. Turán concerning graphs. Fund. Math., 41:137–145, 1954.
[57] T. Zhang and K. K. Parhi. Joint (3,k)-regular LDPC code and decoder/encoder design. IEEE Trans. Signal Processing, 52(4):1065–1079, Apr. 2004.