Fundamental limits on complexity and power consumption in coded communication

Pulkit Grover[†], Andrea Goldsmith[†], Anant Sahai[‡]
† Stanford University ‡ University of California, Berkeley
{pulkit@, andrea@ee.}stanford.edu, sahai@eecs.berkeley.edu

Abstract—**We provide fundamental information-theoretic bounds on the required communication complexity and computation power consumption for encoding and decoding of error-correcting codes in VLSI implementations. These bounds hold for all codes and all encoding and decoding algorithms implemented within the paradigm of our VLSI model. This model essentially views computation on a 2-D VLSI circuit as a computation on a network of connected nodes. The bounds are derived based on analyzing information flow in the circuit. They are then used to show that there is a fundamental tradeoff between the transmit and encoding/decoding power, and that the total (transmit + encoding + decoding) power must diverge to infinity at least as fast as $\sqrt[3]{\log \frac{1}{P_e}}$. On the other hand, for bounded transmit power schemes, the total power must diverge to infinity at least as fast as $\sqrt{\log \frac{1}{P_e}}$ due to the burden of encoding/decoding.**

I. Introduction

Information theory has been extremely successful in determining fundamental capacity limits to communication over a wide range of channels. However, these limits focus only on transmit power, ignoring the complexity and the associated power consumption for processing (e.g. encoding and decoding) the signals. At short distances, the empirical evidence suggests that transmit power does not necessarily dominate total power consumption [2], and hence the intuition from current theory can be misleading, at least in the context of coding. For example, while irregular LDPC codes approach capacity, experimentalists often shun the capacity-approaching constructions in favor of regular LDPC codes (e.g. [3]) that have faster convergence of decoding algorithms [4]. In fact, in many cases, uncoded transmission has been proposed [5] to do away with encoding and decoding altogether, even at the cost of significantly larger transmit power! Thus, this observation in practice is calling us to revise our theory: how *should* we jointly choose the code, encoder, and decoder so that the *total* power consumed is close to the minimum possible? And what is the minimum total power?

While total power minimization by itself has received little attention in information theory, a unified understanding of information theory and various notions of *complexity* has been a long-standing intellectual goal. The issue has been investigated for many code families, notions of complexity, and encoding/decoding algorithms (see [1] for a short survey). In comparison, Shannon's capacity results are *fundamental*: given the communication model, the channel capacity is the limit on the achievable rate for *any reliable communication strategy*.

An intellectual challenge in deriving fundamental results on encoding/decoding complexity and power is that a given code and technology (e.g. 45 nm CMOS), may have many encoding and decoding algorithms and implementation architecture. Truly fundamental bounds therefore need to be oblivious not only to the code construction, but also to the chosen encoding and decoding algorithms and implementation architecture. The goal of this paper is to provide such bounds. To that end, just as channel models take into account the limitations (e.g. noise, bandwidth, path-loss) imposed by communication channels, in Section III, we first provide an implementation model that captures the limitations of information flow in VLSI implementations.

Our model is essentially an adaptation of Thompson's model for VLSI computation [6], [7]. In his thesis [7], Thompson provides fundamental complexity and power bounds for his model for two problems: computing the Fourier transform, and sorting. The bounds essentially build on fundamental network information-theoretic limits on the required communication complexity[1] of computing the desired function. To ascertain the information bottlenecks in this "network," a communication graph representing the circuit is bisected into two roughly equal pieces by cutting as few "wires" as possible. Knowing the number of bits that need to be communicated across this cut-set, a simple application of the cut-set bounding technique [9, Pg. 376] provides the minimum run-time (the number of "clock-cycles", denoted here by τ) for the computation.

Work of El Gamal, Greene, and Pang [10] uses Thompson's VLSI model to derive lower bounds on the VLSI-area complexity of encoding/decoding. Instead of using Thompson's technique, the authors use a technique credited to Angluin [11] in [12][2]: in a single analysis step, the authors break the entire circuit into multiple small sub-circuits. Unfortunately, this technique does not extend to provide bounds on energy/power of encoding/decoding (this aspect is discussed in detail in Section III). Nevertheless, it does provide hope that such bounds can be derived.

In order to establish results on energy and power consumption,

We thank Abbas El Gamal, Salim El Rouayheb, Reza Mirghaderi, Sudeep Kamath, Jan Rabaey, Karthik Ganesan and Yang Wen for helpful discussions. This research was supported by the Interconnect Focus Center of the Semiconductor Research Corporation, the NSF Center for Science of Information (CSoI) and by NSF grant NSF-CCF-0917212. Because of space-limitations, the full proofs of results, and a more detailed literature survey, appear in [1].

[1] The VLSI theory of computation and the theory of communication complexity [8] developed almost simultaneously, and certainly share their origins.

[2] Angluin's manuscript [11] is so far unavailable to us.

in Section IV, we revert back to Thompson's technique [7]. Instead of breaking the circuit into multiple sub-circuits in one step, we analyze the computation by repeatedly bisecting the circuit until the resulting sub-circuits are sufficiently small. We then use an error-exponent-style analysis that yields non-asymptotic bounds for any given error probability and block-length. A similar approach was adopted in [13], where weaker techniques limited our results to just the encoding complexity. In this paper, we are able to derive stronger bounds than those in [13] that apply to encoding as well as decoding.

In Section IV, we provide a lower bounds on the energy and power consumption in encoding and decoding. Assuming communication over a Gaussian channel with average transmit power P_T, this lower bound shows that the energy consumed in encoding/decoding is at least $\Omega\left(k\sqrt{\log\frac{1}{P_e^{blk}}/P_T}\right)$. Optimizing over transmit power, we conclude that the *total* (transmit + encoding + decoding) power must scale at least as fast as $\Omega\left(\sqrt[3]{\log\frac{1}{P_e^{blk}}}\right)$. Further, if we use bounded transmit power even as P_e is driven to zero, then the lower bound is larger, and scales at least as fast as $\Omega\left(\sqrt{\log\frac{1}{P_e^{blk}}}\right)$! Thus, we conclude that the optimal strategy will increase transmit power as P_e is driven to zero in order to reduce the complexity and power consumption of encoding and decoding.

Although most of our results are non-asymptotic, we use asymptotic notation (*i.e.* the "big-Oh" notation) to convey an intuitive understanding. Vectors are denoted in bold (e.g. $\mathbf{X}_1^n$ is a vector of length n). For any set $\mathcal{A}$, $|\mathcal{A}|$ denotes its cardinality.

II. System model

We consider a point-to-point communication link. An information sequence of k fair coin flips $\mathbf{b}_1^k$ is encoded into 2^{nR} binary-alphabet codewords $\mathbf{X}_1^n$, hence the rate of the code is $R = \frac{k}{n}$ bits/channel use. The codeword $\mathbf{X}_1^n$ is modulated using BPSK modulation and sent through an Additive White Gaussian Noise (AWGN) channel of bandwidth W. The decoder estimates the input sequence $\widehat{\mathbf{b}}_1^k$ by first performing a hard-decision on the received channel symbols before using these hard-decisions $\mathbf{Y}_1^n$ to decode the input sequence. The overall channel is therefore a Binary Symmetric Channel (BSC) with raw bit-error probability $p_{ch} := \mathbb{Q}\left(\sqrt{\frac{\zeta P_T}{\sigma_z^2}}\right)$, where $\mathbb{Q}(x) = \int_x^\infty \frac{1}{2\pi} e^{-\frac{x^2}{2}} dx$, ζ is the path-loss associated with the channel, P_T is the transmit power of the BPSK-modulated signal, and σ_z^2 is the variance of the Gaussian noise in the hard-decision estimation. The encoder-channel-decoder system operates at an average block-error probability P_e^{blk} given by

$$P_e^{blk} = \Pr\left(\widehat{\mathbf{b}}_1^k \neq \mathbf{b}_1^k\right). \tag{1}$$

III. VLSI model of encoding/decoding implementation

A. Overview

As mentioned earlier, our model is an adaptation of Thompson's model [7]. The model assumes that any VLSI circuit is a set of computational nodes that are connected to each other using finite-width wires. In each clock-cycle, the nodes communicate with all the other nodes that they are connected to. The nodes can perform simple computations (e.g. a three input NAND) on the inputs received by the nodes. The computation terminates at a predetermined τ number of clock-cycles.

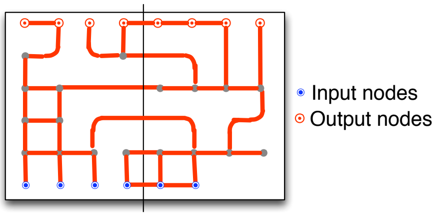

Fig. 1. The VLSI model of implementation and an example bisection. A bisection needs to divide only a specified subset of nodes into two roughly equal pieces. The above cut bisects the set of output nodes in the communication graph of the circuit.

B. Detailed model

We assume that the encoder $\mathcal{E}$ and the decoder $\mathcal{D}$ are implemented using the "VLSI model of computation." The model, which is detailed below, captures essentially the communication limitations in a VLSI circuit, allowing a network-information-theoretic analysis to be performed on the circuit itself.

- The circuit to compute the encoding/decoding function must be laid out on a grid of unit squares. It is composed of finite-memory computational nodes and interconnecting wires. Wires run along the edges and can cross at grid points. A computational node is a grid point that is either a logic element, a wire-connection, or an input/output pin.
- The wires are assumed to be bi-directional. In each clock-cycle, each node sends one bit of information to each of the nodes that it is connected to over these wires.
- The circuit is planar[3], and each node is connected to at most four other nodes using the bi-directional wires.
- The inputs of the computation (information bits for the encoder, and channel outputs for the decoder) are stored in separate *source nodes*, and the outputs of computation (codeword bits for the encoder, reconstructed information bits at the decoder) are stored in *output nodes*. The same node may act as a source-node and as an output node[4]. Also, each input value may enter the circuit at only the corresponding source node.
- Each wire has a finite-width λ specified by the technology chosen to implement the circuit. Further, the area of each node is at least λ^2.
- The processing is done in "batches," *i.e.*, a set of inputs is processed and outputs are released before the next set of inputs arrives into the source nodes.

The last assumption rules out "pipelining" [10] and sequential processing of inputs (e.g. by reading them from external storage while the computation is in process). An example implementation that lies outside our model is the decoding of a convolution code in a streaming manner, where bit estimates are outputted before the entire block is received.

Energy/power model: The energy consumed in computation is assumed to be given by $E_{proc} = \xi_{tech} A_{wires} \tau$, where ξ_{tech} is the "energy parameter" of the implementation that depends on the capacitance per-unit length of wiring in the circuit.

[3] Our results extend to multi-layered chips in a manner similar to that in [7].

[4] For instance, in LDPC decoders, the variable nodes act as source nodes as well as output nodes.

In order to translate energy to power consumption, we assume that the decoding throughput (the number of information bits encoded/decoded per second) of the encoder/decoder is the same as the data rate R_{data} (information bits/sec) across the channel. This assumption is necessitated by the requirement of avoiding buffer overflow at the receiver. Because the batch of k data bits are processed in parallel by the encoder/decoder, the amount of time available for the processing is $T_{proc} = \frac{k}{R_{data}}$ seconds. The required power for encoding/decoding is therefore $P_{proc} = \frac{E_{proc}}{T_{proc}} = \frac{E_{proc}}{k} R_{data}$, which is simply the energy-per-information-bit multiplied by the data rate.

Definition 1 (Channel Model (ζ, σ_z^2)): Channel Model (ζ, σ_z^2) denotes (as described in Section II) a BSC(p_{ch}) channel that is a result of hard-decision at the receiver across an AWGN channel of average transmit power P_T, path loss ζ and noise variance σ_z^2.

Definition 2 (Implementation Model (ξ_{tech}, λ)): Implementation Model $(\zeta, \xi_{tech}, \lambda)$ denotes the implementation model (as described in Section III) having minimum wire-width λ, and energy parameter ξ_{tech}.

C. A partial survey of existing results for the model

As noted earlier, Thompson derived fundamental complexity and power bounds for the VLSI model for computing the Fourier transform and sorting [7]. The idea used by Thompson is as follows: to ascertain the information bottlenecks in this "network," a communication graph representing the circuit (see Fig. 1) is bisected into two roughly equal pieces by "cutting" as few "wires" as possible. Knowing the number of bits that need to be communicated across this cut-set, a simple application of the cut-set bounding technique [9, Pg. 376] provides the minimum run-time (the number of "clock-cycles", denoted here by τ) for the computation. Although this minimum run-time can be reduced by increasing the "min-cut" (referred to as the "minimum bisection width" in the VLSI theory literature [6]), it comes at the cost of an increased wiring area.

Thus there is a fundamental tradeoff between the number of clock-cycles τ, and the wiring area A_{wires}. This is usually characterized by lower bounds on $A_{wires}\tau^2$. The energy consumption is intimately connected to a closely related quantity: the energy consumed in wiring can be approximated by the product $A_{wires}\tau$ [7]. The results have been extended to many other computational problems, such as multiplication, computing boolean functions, etc. (see [1] and the references therein).

The work of El Gamal *et al.* [10] uses Thompson's VLSI model, but (as noted earlier) uses instead a technique credited to Angluin [11]: in a single analysis step, the authors break the entire circuit into multiple small sub-circuits. The authors show that the product $A_{chip}\tau^2$ is at least $\Omega\left(nR^2 \log \frac{n}{P_e^{blk}}\right)$, where A_{chip} is the area of the smallest rectangle that encloses the circuit. As noted earlier, we need lower bounds on $A_{wires}\tau$ in order to obtain lower bounds on energy/power[5]. We therefore believe that the proof technique in [10] does not extend to provide bounds on $A_{wires}\tau$ (because $A_{wires} < A_{chip}$), and therefore would not yield bounds on energy/power of encoding/decoding.

D. Definitions for computation on the communication graph

For analyzing a computation process in the VLSI model, we define a communication graph for a circuit in the VLSI model.

Definition 3 (Communication graph): A communication graph G corresponding to a circuit implemented in the model described in Section III has vertices at the computational nodes and edges along the interconnecting wires.

The set of vertices is denoted by V, and of edges by E. The computation can be viewed as being performed on the communication graph G. In order to analyze the bottlenecks of information flow in G, we define bisections on G. A cut is said to "bisect" G (see Fig. 1) if roughly half of a specified subset of nodes lie on either side of the cut. More formally:

Definition 4 (Bisection [6], [7]): Let $S \subseteq V$ be a subset of the vertices, and $E_S \subseteq E$ be a subset of the edges in G. Then E_S *bisects* S *in* G if removal of E_S cuts V into sets V_1 and V_2 and S into sets $S_1 \subseteq V_1$ and $S_2 \subseteq V_2$ such that $\big||S_1| - |S_2|\big| \leq 1$.

Definition 5 (Minimum bisection width, and MBW-cut): The *minimum bisection width* (MBW) of S in G is defined as $\min\{|E_S| \text{ s.t. } E_S \text{ bisects } S \text{ in } G\}$. The corresponding cut is called a minimum-bisection-width-cut; or an MBW cut.

Definition 6 (Nested bisections): Let $S \subseteq V$ be a subset of the vertices, and $E_S \subseteq E$ be a subset of the edges in G that bisects S such that removal of E_S cuts G into G_1, G_2; of V into V_1 and V_2; and of S into $S_1 \subseteq V_1$ and $S_2 \subseteq V_2$. Let $E_{S,i}, i = 1, 2$, be subsets of edges in G_i that bisect S_i. The resulting partitions of sets S, V each into four disjoint subsets is called a *2-step nested bisection of S in G.*

The physical wires and computational nodes corresponding to the disjoint and mutually disconnected subsets that result from (nested) bisections constitute the *sub-circuits*. In our proof techniques, we will perform r-step nested bisections for some $r \in \mathbb{Z}^+$, conceptually breaking the original circuit into 2^r sub-circuits, indexed by $i = 1, 2, \ldots, 2^r$. By assumption, bits are passed across an edge in each direction at every clock-cycle.

Definition 7 (Bits communicated across a cut): Suppose the communication graph V of a circuit implemented in the Implementation Model (ξ_{tech}, λ) is partitioned into two disconnected sets on removal of edges E_{cut}. An ordered set of the bits passed along the edges in E_{cut} in either direction during the computation process is called the vector of bits communicated across the cut. The length of this vector is called the number of bits communicated across the cut.

If w is the width of a cut, the number of bits communicated across the cut in τ processing cycles is simply $2w\tau$, where the factor of 2 is because of the bi-directional nature of the wires. In this paper, we will mostly be interested in the bits communicated across an MBW-cut. In particular, we will be interested in the *vector of communicated bits across all MBW-cuts* in r-steps of nested bisections of a communication graph. This vector is simply the concatenation of vectors of bits

[5] In his thesis, Thompson also places emphasis on this issue: while Theorem 1 [7, Pg. 54] in his thesis obtains a lower bound on the area of the circuit's minimum enclosing rectangle (which includes unoccupied area; much like the result in [10]), for lower bounding purposes, he uses Theorem 2 [7, Pg. 54] that obtains a lower bound on the *wiring* area.

communicated across the MBW-cuts in the r steps of nested bisections. The length of this vector, which is the total number of bits communicated across the MBW-cuts in the r bisection steps, is denoted by $B_{total}^{[1:r]}$.

Definition 8 (Communication bits for a sub-circuit): An ordered collection of bits communicated during the computation process along the wires that have exactly one computational node inside a given sub-circuit is called the vector of communication bits for the i-th sub-circuit, denoted by $\vec{b}_{comm,i}^{[1:r]}$. The bits could have been communicated in either direction along the bi-directional wires during $s = 1, 2, \ldots, r$. The length of $\vec{b}_{comm,i}^{[1:r]}$, the vector of communication bits for the i-th sub-circuit, is denoted by $B_{subckt,i}^{[1:r]}$. Considering the sub-circuits at the end of r steps of successive bisections, because any wire that has exactly one computational node inside the sub-circuit must be connected to a exactly one node outside the sub-circuit, we have that $\sum_{i=1}^{2^r} B_{subckt,i}^{[1:r]} = 2B_{total}^{[1:r]}$.

IV. Main results

This section presents a sequence of results that culminate in our lower bounds on total power. The results are based on r-step nested bisections of the communication graph of the respective circuit, each step bisecting the respective "information nodes" (input nodes for encoding; output nodes for decoding). For clarity of exposition, we assume that k is a power of 2. Thus, at the end of r steps of nested bisections, there are 2^r sub-circuits, each with $\frac{k}{2^r}$ "information nodes." The results can be extended suitably to values of k that are not powers of 2.

Lemma 1 (Energy lower bounds): Under the Channel Model (ζ, σ_z^2) and Implementation Model (ξ_{tech}, λ), let k (a power of 2) be the number of information nodes at the encoder/decoder, and n be the nodes corresponding to channel input/output symbols. On performing $r < \log_2(k/2)$ steps of nested bisections on this circuit, let the total number of bits that pass across the MBW-cuts of the r stages of nested bisections be denoted by $B_{total}^{[1:r]}$. Then,

$$E_{proc} > \xi_{tech}\lambda^2 \frac{(\sqrt{2}-1)}{4\sqrt{2}} \sqrt{\frac{n}{2^r}} B_{total}^{[1:r]}, \qquad (2)$$

Proof overview: At each stage $s = 1, 2, \ldots, r$, we first obtain a lower bound on $A_{wires}\tau^2$ by lower bounding the sum $\sum_{i=1}^{2^s} A_{wires,i}^{(s)}\tau^2$ for given number of bits $B^{(s)}$ passed across MBW-cuts in the s-th stage, *i.e.*, $B^{(s)} = \sum_{i=1}^{2^s} B_i^{(s)}$, where $B_i^{(s)}$ is the number of bits passed across the MBW-cut of the i-th sub-circuit in the s-th stage. This gives r different lower bounds on the product $A_{wires}\tau^2$ for the encoding/decoding process, one for each stage, each depending on $B^{(s)}$ for the corresponding stage. Then, noting that $\sum_{s=1}^{r} B^{(s)} = B_{total}^{[1:r]}$, we obtain a lower bound on the product $A_{wires}\tau^2$ as a function of $B_{total}^{[1:r]}$. Using the simple lower bound on A_{wires}, namely, $A_{wires} \geq n\lambda^2$, we obtain a lower bound on $A_{wires}\tau$ which is proportional to the energy consumed in our implementation model. ■

Observe in Lemma 1 that while the total number of bits $B_{total}^{[1:r]}$ that cross MBW-cuts in r stages increases with r, so does the denominator 2^r. The following lemma (Lemma 2) uses the total number of communicated bits, $B_{total}^{[1:r]}$, to obtain a lower bound on error probability. Lemma 1 and Lemma 2 are used to obtain lower bounds on energy given the target P_e^{blk}. The value of r can then be chosen to obtain the best lower bound on energy.

Lemma 2 (P_e^{blk} lower bounds): Under Channel Model (ζ, σ_z^2) and Implementation Model (ξ_{tech}, λ), on performing $r < \log_2(k/2)$ steps of nested bisections on this circuit, let the total number of bits that pass across the MBW-cuts over the r stages be denoted by $B_{total,enc}^{[1:r]}$ at the encoder and $B_{total,dec}^{[1:r]}$ at the decoder. If $\min\{B_{total,enc}^{[1:r]}, B_{total,dec}^{[1:r]}\} < \frac{k}{4}$, then

$$P_e^{blk} \geq p_{ch} 2^{-\log_2\left(\frac{1}{2p_{ch}}\right)\frac{n}{2^{r-1}}\left(1 - \frac{\frac{k}{2} - 2\min\{B_{total,enc}^{[1:r]}, B_{total,dec}^{[1:r]}\}}{n}\right)},$$

where $p_{ch} = \mathbb{Q}\left(\sqrt{\frac{\zeta P_T}{\sigma_z^2}}\right)$.

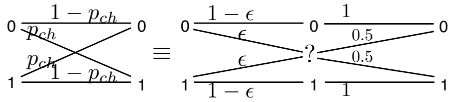

Fig. 2. For $\epsilon = 2p_{ch}$, the BSC on left is equivalent to the concatenation of erasure channel with another DMC shown on right.

Proof overview for decoding computation: Working with a BSC model turns out to be complicated: the problem is attempted in [13] where we derive looser bounds that apply to encoding, but not to decoding. Instead, we take an indirect route: we first find bounds on a BEC and use them for a BSC by observing that a BSC(p_{ch}) can be interpreted as a physically degraded BEC with erasure probability $2p_{ch}$ (see Fig. 2).

Consider the i-th decoder sub-circuit at the r-th (final) stage, $i = 1, 2, \ldots, 2^r$, strengthened with the knowledge of the erasure-outputs. Denote the bits communicated across the boundary of the i-th decoder sub-circuit in either direction by $\vec{b}_{comm,i}^{[1:r]}$ (summed over all the r stages, and shared across MBW-cuts across different stages), and their number by $B_{subckt,i}^{[1:r]}$. As noted earlier, the total number of bits communicated across MBW-cuts (over all the r bisection stages) is $\sum_{i=1}^{2^r} B_{subckt,i}^{[1:r]} = 2B_{total,dec}^{[1:r]}$. Define $\mathcal{S} := \{i : B_{subckt,i}^{[1:r]} < \frac{k}{2^r}\}$. A simple averaging argument shows that $|\mathcal{S}| > 2^{r-1}$.

The i-th decoder sub-circuit can only use the communication bits $\vec{b}_{comm,i}^{[1:r]}$ and the n_i channel outputs in order to decode the $\frac{k}{2^r}$ information bits in the sub-circuit. Assuming optimistically that the communication bits themselves can deliver some of the $\frac{k}{2^r}$ information bits in an error free manner, when $B_{subckt,i}^{[1:r]} < \frac{k}{2^r}$, the decoder still has to use the n_i available channel outputs to decode the remaining $\frac{k}{2^r} - B_{subckt,i}^{[1:r]}$ information bits. However, if the number of unerased channel outputs in the i-th sub-circuit is smaller than $\frac{k}{2^r} - B_{subckt,i}^{[1:r]}$, then this decoder sub-circuit likely makes an error (i.e. with probability $> \frac{1}{2}$, because it has to guess at least one bit). How many erasures should there be for this to happen? Clearly, the number of erasures $\#_{E,i}$ must be at least $n_i - \frac{k}{2^r} + B_{subckt,i}^{[1:r]} + 1$. Assuming (again, conservatively) that the first $\#_{E,i}$ bits are erased, and denoting this event by $\mathcal{W}_i$, the probability of this event is

$$\Pr(\mathcal{W}_i) = \epsilon^{n_i - \frac{k}{2^r} + B_{subckt,i}^{[1:r]} + 1} = 2p_{ch}(2p_{ch})^{n_i\left(1 - \frac{\frac{k}{2^r} - B_{subckt,i}^{[1:r]}}{n_i}\right)}.$$

Averaging this probability over $i \in \mathcal{S}$, using the convex-∪ nature of the exponential function, and using the lower bound of $\frac{1}{2}$ on the error probability under $\mathcal{W}_i$ yields the lower bound. ■

The derivation for the bound on P_e^{blk} given the number of bits passed in *encoding* makes no assumptions on the decoding model (the decoder is assumed to be the optimal decoder). Because of space limitations, that derivation is relegated to [1].

Theorem 1 (Energy in encoding/decoding): Under Channel Model (ζ, σ_z^2) and Implementation Model (ξ_{tech}, λ), let the energy consumed in encoding be denoted by E_{enc}, and that in decoding be denoted by E_{dec}. Then, for this model, for any $r < \log_2(k/2)$, $\min\{E_{enc}, E_{dec}\}$ satisfies either

$$P_e^{blk} \geq p_{ch} 2^{-\frac{n}{2^{r-1}} \log_2 \frac{1}{2p_{ch}} \left(1 - \frac{\frac{k}{2} - \frac{8\sqrt{2}}{(\sqrt{2}-1)\xi_{tech}\lambda^2} \sqrt{\frac{2^r}{n}} \min\{E_{enc}, E_{dec}\}}{n}\right)},$$

or $\min\{E_{enc}, E_{dec}\} > \xi_{tech}\lambda^2 \frac{\sqrt{2}-1}{16\sqrt{2}\sqrt{R}} \sqrt{\frac{k^3}{2^r}}$.

Proof: Immediate from Lemma 1 and Lemma 2. ■

Corollary 1 (Asymptotic lower bounds on energy): Under Channel Model (ζ, σ_z^2) and Implementation Model (ξ_{tech}, λ), in the limit of small P_e^{blk},

$$E_{proc} \gtrsim \xi_{tech}\lambda^2 \frac{\sqrt{2}-1}{24\sqrt{2}\sqrt{\left(1-\frac{R}{3}\right)}} k \sqrt{\frac{\log_2\left(\frac{p_{ch}}{P_e^{blk}}\right)}{\log_2\left(\frac{1}{2p_{ch}}\right)}}. \quad (3)$$

Proof overview: The proof follows by choice of r according to the following equation:

$$2^{r-1} \approx n \log_2\left(\frac{1}{2p_{ch}}\right)\left(1 - \frac{R}{3}\right) / \log_2\left(\frac{p_{ch}}{P_e^{blk}}\right), \quad (4)$$

a choice which can be made in the limit of small P_e^{blk}. ■

Corollary 2 (Asymptotic lower bounds on total power): Under Channel Model (ζ, σ_z^2) and Implementation Model (ξ_{tech}, λ), across a channel of path-loss ζ, in the limit $P_e^{blk} \to 0$, the total power is bounded as follows

$$P_{tot} \gtrsim \left(2^{\frac{1}{3}} + \frac{1}{2^{\frac{2}{3}}}\right) \eta^{\frac{2}{3}} \sqrt[3]{\ln\left(\frac{1}{P_e^{blk}}\right)}, \quad (5)$$

where $\eta = \frac{\sqrt{2}-1}{48\sqrt{1-R/3}} \sqrt{\frac{\sigma_z^2}{\zeta}} W R \lambda^2$, and W is the channel bandwidth.

Proof overview: In our hard-decision channel model, the term $\log \frac{1}{2p_{ch}}$ scales proportionally to received power ζP_T. Because we want to minimize the total power, and because E_{proc} is normalized by k in order to obtain power,

$$P_{total} \geq \min_{P_T} P_T + P_{proc} \approx \min_{P_T} P_T + \beta \sqrt{\frac{\log \frac{1}{P_e^{blk}}}{P_T}}, \quad (6)$$

for some constant β. As $P_e^{blk} \to 0$, choosing $P_T = \log^x \frac{1}{P_e^{blk}}$, to minimize the total power, x must satisfy $x = \frac{1}{2} - \frac{1}{2}x$. Thus, $x = \frac{1}{3}$, and we get that the total power, as well as the optimizing transmit power, must scale at least as fast as $\sqrt[3]{\log \frac{1}{P_e^{blk}}}$. It is also clear from (6) that if we use bounded P_T, then the total power scales at least as fast as $\sqrt{\log \frac{1}{P_e^{blk}}}$. ■

Plots for lower bounds on total power appear in [1].

V. Discussion and Conclusions

This paper provides fundamental limits on complexity and power consumed in VLSI implementations of encoding/decoding of error-correcting codes. The limits are derived by analyzing the network of interconnected nodes in VLSI implementations using simple information-theoretic tools (*i.e.* cut-set bounds). The underlying intuition behind our results is simple: exchange of bits in VLSI computation requires power. At the same time, to utilize the benefits offered by increased blocklengths, one needs to exchange bits. Otherwise, the large code can effectively be split into codes of smaller blocklengths which have worse error correction capability. In other words, in order to get the coding gains commensurate with large codes, there is more exchange of bits in the implementation, and hence more power consumption. This leads to a tradeoff in transmit power (to compensate for a weaker code) and computation power. Because of the simplicity of this intuition, we think that these results should extend to implementations that do not fit directly into the model (e.g. 3-D circuits, soft decisions, etc.).

While we account for power consumed in wires in the encoder/decoder, our model here ignores the power consumed in the computational nodes. These results therefore complement those in [4], [14], [15] that account for power consumed in the computational nodes, but ignore the power consumed in wires.

References

[1] P. Grover, A. Goldsmith, and A. Sahai, "Fundamental limits on transmission and computation power," Feb. 2012, extended version of paper submitted to ISIT'12. [Online]. Available: http://www.stanford.edu/~pulkit/files/ISIT12FullProofs.pdf

[2] S Cui, AJ Goldsmith and A Bahai, "Energy Constrained Modulation Optimization," *IEEE Trans. Wireless Commun.*, vol. 4, no. 5, pp. 1–11, Sep. 2005.

[3] Z. Zhang, V. Anantharam, M. Wainwright, and B. Nikolic, "An efficient 10 GBASE-T ethernet LDPC decoder design with low error floors," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 4, pp. 843–855, Apr. 2010.

[4] P. Grover, K. A. Woyach, and A. Sahai, "Towards a communication-theoretic understanding of system-level power consumption," *IEEE Jour. Selected Areas in Comm., Special Issue on Energy-Efficient Communications. Arxiv preprint arXiv:1010.4855*, vol. 29, no. 8, Sep. 2011.

[5] C. Marcu, D. Chowdhury, C. Thakkar, J.-D. Park, L.-K. Kong, M. Tabesh, Y. Wang, B. Afshar, A. Gupta, A.Arbabian, S. Gambini, R. Zamani, E. Alon, and A. Niknejad, "A 90 nm CMOS low-power 60 GHz transceiver with integrated baseband circuitry," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 12, pp. 3434 – 3447, 2009.

[6] C. D. Thompson, "Area-time complexity for VLSI," in *Proceedings of the 11th annual ACM symposium on Theory of computing (STOC)*. New York, NY, USA: ACM, 1979, pp. 81–88.

[7] ——, "A complexity theory for VLSI," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, USA, 1980.

[8] A. C.-C. Yao, "Some complexity questions related to distributive computing," in *Proceedings of the eleventh annual ACM symposium on Theory of computing*, 1979, pp. 209–213.

[9] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 1st ed. New York: Wiley, 1991.

[10] A. El Gamal, J. Greene, and K. Pang, "VLSI complexity of coding," in *The MIT Conf. on Adv. Research in VLSI*, Cambridge, MA, Jan. 1984.

[11] D. Angluin, "VLSI: On the merits of batching," Manuscript, 1982.

[12] K. F. Pang, "Complexity results in VLSI and distributed computing," Ph.D. dissertation, Stanford University, CA, USA, 1985.

[13] P. Grover, A. Goldsmith, A. Sahai, and J. Rabaey, "Information theory meets circuit design: Why capacity-approaching codes require more circuit area and power," in *Proceedings of the Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Sep. 2011.

[14] A. Sahai and P. Grover, "A general lower bound on the VLSI decoding complexity," in *preparation*, 2012. [Online]. Available: http://www.eecs.berkeley.edu/~pulkit/papers/ComplexityITPaper.pdf

[15] P. Grover, "Bounds on the tradeoff between rate and complexity for sparse-graph codes," in *the IEEE Information Theory Workshop (ITW)*, Lake Tahoe, CA, 2007.