Energy, Latency, and Reliability Tradeoffs in Coding Circuits

Christopher G. Blake and Frank R. Kschischang
Department of Electrical & Computer Engineering
University of Toronto
christopher.blake@mail.utoronto.ca frank@comm.utoronto.ca

Abstract

It is shown that fully-parallel encoding and decoding schemes with asymptotic block error probability that scales as \( O(f(n)) \) have Thompson energy that scales as \( \Omega(\sqrt{\ln f(n)n}) \). As well, it is shown that the number of clock cycles (denoted \( T(n) \)) required for any encoding or decoding scheme that reaches this bound must scale as \( T(n) \geq \sqrt{\ln f(n)} \). Similar scaling results are extended to serialized computation. The Grover information-friction energy model is generalized to three dimensions and the optimal energy of encoding or decoding schemes with probability of block error \( P_e \) is shown to be at least \( \Omega \left( n (\ln P_e(n))^{\frac{1}{2}} \right) \).

I. INTRODUCTION

EXPANDING on work started in [1] and more recently advanced in [2]–[4], we borrow a computational complexity model introduced in [5] that allows us to model the energy and number of clock cycles of a computation. We consider fundamental tradeoffs between the asymptotic energy, number of clock cycles, and block error probability for sequences of good encoders and decoders.

Definition 1. An \( f(n) \)-coding scheme is a sequence of codes of increasing block length \( n \), together with a sequence of encoders and decoders, in which the block error probability associated with the code of block length \( n \) is less than \( f(n) \) for sufficiently large \( n \).

We show, in terms of \( T(n) \) (the number of clock cycles of the encoder or decoder for the code with block length \( n \)) that an \( f(n) \)-coding scheme that is fully parallel has encoding and decoding energy \( (E) \) that scales as \( E \geq \Omega \left( \frac{n \ln f(n)}{T(n)} \right) \). We show that the energy optimal number of clock cycles for encoders and decoder \( (T(n)) \) for an \( f(n) \)-coding scheme scales as \( O \left( \sqrt{\ln f(n)} \right) \), giving a universal energy lower bound of \( \Omega \left( \sqrt{\ln f(n)n} \right) \). A special case of our result is that exponentially low probability of error coding schemes thus have encoding and decoding energy that scales at least as \( \Omega \left( n^{\frac{1}{2}} \right) \) with energy-optimal number of clock cycles that scales as \( \Omega \left( n^{\frac{3}{2}} \right) \). This approach is generalized to serial implementations.

Recent work on the energy complexity of good decoding has focused largely on planar circuits. However, circuits implemented in three-dimensions exist [6], and so we generalize the recent information friction (or bit-meters) model introduced by Grover in [3] to circuits implemented in three-dimensions and extend the technique of Grover to show that, in terms of block length \( n \), a bit-meters coding scheme in which block error probability is given by \( P_e(n) \) has encoding/decoding energy that scales as \( \Omega \left( n (\ln P_e(n))^{\frac{1}{2}} \right) \). We show how this approach can be generalized to an arbitrary number of dimensions.

In Section II we discuss prior work, and in particular we discuss existing results on complexity lower bounds for different models of computation for different notions of “good” encoders and decoders. The main technical results of this work are in Section III, where we study the Thompson energy model, and in Section IV where we study a multi-dimensional generalization of the Grover bit-meters model. In these sections we present lower bounds for decoders, as the derivation for encoding lower bounds is almost exactly the same. We provide an outline of the technique for encoder lower bounds in Section V. In Section VI we discuss limitations and weaknesses in the model used. In Section VII we discuss other energy models of computation. In Section VIII we discuss possible future work, and conjecture that similar tradeoffs may extend to circuits that perform inference.

Notation: We use standard Bachmann-Landau notation in this paper. The statement \( f(x) = O(g(x)) \) means that for sufficiently large \( x \), \( f(x) \leq cg(x) \) for some positive constant \( c \). The statement \( f(x) = \Omega(g(x)) \) means that for sufficiently large \( x \), \( f(x) \geq cg(x) \) again for some constant \( c \). The statement \( f(x) = \Theta(g(x)) \) means that there are two positive constants \( b \) and \( c \) such that \( b \leq c \) and for sufficiently large \( x \), \( bg(x) \leq f(x) \leq cg(x) \).

Part of this work was submitted for presentation at the 2016 International Symposium on Information Theory.
II. Prior Related Work: Computational Complexity Lower Bounds for Good Decoders and Encoders

The earliest work on computational complexity lower bounds for good decoding comes from Savage in [7] and [8], which considered bounds on the memory requirements and number of logical operations needed to compute decoding functions. However, wiring area is a fundamental cost of good decoding and the authors do not consider this. More recently, in [11], the authors use a model similar to our model, except the notion of “area” the authors use is the size of the smallest rectangle that completely encloses the circuit under consideration.

In [2], Grover et al. consider the same model that we do, and find Thompson energy lower bounds as a function of probability of block error probability for good encoders and decoders. Our analysis of the Thompson model differs from the approach of Grover et al. in a number of ways. Firstly, central to the work of Grover et al. is a bound on block error probability if inter-subcircuit bits communicated is low (presented in Lemma 2 in the Grover et al. paper), which is analogous to our result in (4) of the proof of Theorem 1. Our result simplifies this relationship using simple probability arguments. Secondly, the Grover et al. paper does not present what energy-optimal number of clock cycles are in terms of asymptotic probability of block error, nor do they present the fundamental tradeoff between number of clock cycles, energy, and reliability within the Thompson model that we present in this paper. Moreover, the technique of [2] does not extend to serial implementations.

In [4] we considered the corner case of decoding schemes in which block error probability asymptotically was less than \(\frac{1}{2}\) for serial and parallel decoding schemes. We did not, however, analyze schemes in terms of the rate at which block error probability approaches 0, nor did we compute energy-optimal number of clock cycles as we do herein.

There has also been some work on complexity scaling rules for encoding and decoding of specific types of codes. Low density parity check coding VLSI scaling rules have been studied in [9], [10] and polar coding scaling rules have been studied in [11]. The scaling rules presented in this paper are general and apply to any code.

Another computational model that has proven more tractable than the Turing Time complexity model is the constant depth circuit model (see [12] for a detailed description of this model). Super-polynomial lower bounds on the size of constant depth circuits that compute certain notions of “good encoding functions” (though not decoding) were derived in [13]. In this case, the notion of “good” considered was the ability to correct at least \(\Omega (n)\) errors at rates asymptotically above 0. Similar related work exists in [14] which discovered lower bounds on the formula-size of functions that perform good error control coding; similar bounds were later discovered in [15].

III. Thompson Model

A. Circuit Model

The model we will consider derives from Thompson [5]. The specific model we consider has been studied in [2], [4], [9], [10]. The reader should refer to [4] for details of the model. The important parameters to be extracted from the model are \(A\), the circuit area, and \(T\), the number of clock cycles in a computation. Since in this paper we are only concerned with scaling rules, we assume that both the technology constant and the wire width considered in [2], [4] are equal to 1. The energy of a computation is thus defined as \(E = AT\).

Note that a circuit can be associated with a graph in the natural way, in which a wire corresponds to an edge of the graph and a node corresponds to a vertex. An edge connects two vertices if their associated nodes are connected by wires. A diagram of a small circuit next to its associated graph is given in Fig. [1]

Lemma [2] presented below is derived in [5] and it relates the area of a circuit to its graph’s minimum bisection width, and is a key component of our Thompson model circuit lower bounds.

B. Definitions and Lemmas

To present the main results of this paper we shall present a sequence of definitions and lemmas similar to [2], [4].

Lemma 1. [4] Suppose that \(X, Y\), and \(\hat{X}\) are random variables that form a Markov chain \(X \rightarrow Y \rightarrow \hat{X}\) and \(X\) takes on values from a finite alphabet \(\mathcal{X}\) with a uniform distribution, (i.e., \(P(X = x) = \frac{1}{|\mathcal{X}|}\) for all \(x \in \mathcal{X}\)), \(Y\) takes on values from a finite set \(\mathcal{Y}\), and \(\hat{X}\) from a set \(\hat{\mathcal{X}}\). Suppose as well that \(\hat{X} \in \mathcal{X}\). Then:

\[
P(\hat{X} = X) \leq \frac{|\mathcal{Y}|}{|\mathcal{X}|}.
\]

Remark 1. We will interpret \(X\) as the set of symbols a particular subcircuit will need to estimate, \(\hat{X}\) as that subcircuit’s estimate of those symbols, and \(Y\) as the bits injected into the subcircuit during the computation. Note that this result mirrors the result of Lemma 4 in [3]. In this lemma, the author proves that if a circuit has \(\frac{r}{2}\) bits to make an estimate \(\hat{X}\) of a random variable \(X\) that is uniformly distributed over all binary strings of length \(r\), then that circuit makes an error with probability at least \(\frac{1}{9}\). Our lemma presented here includes this lemma as a special case by setting \(|\mathcal{Y}| = 2^r\) and \(|\mathcal{X}| = 2^r\). In this case we can infer: \(P(\hat{X} \neq X) \geq 1 - \frac{2^r}{2^r} \geq 1 - 2^{-\frac{r}{2}} > \frac{1}{9}\), where the last inequality is implied by \(r \geq 1\).
Proof: (of Lemma 1) See [4]. This flows from a simple application of the law of total probability and the definition of a Markov chain.

**Definition 2.** A bisection of a graph \( G = (V, E) \) of a set of vertices \( V' \in V \) is a set of edges \( E' \in E \) that, once removed from the graph, results in two disconnected subgraphs with vertices \( V_1 \) and \( V_2 \) in which \( ||V' \cap V_1| - |V' \cap V_2|| \leq 1 \). That is, it is the set of edges that, once removed, divides the vertices of \( V' \) roughly in half. The minimum bisection width of a set of vertices \( V' \) is the size of a smallest bisection.

Note that since a circuit is associated with a graph, we can discuss such a circuit’s minimum bisection width, that is the minimum bisection width of the graph with which it is associated. Herein we will consider bisecting the output nodes of a circuit.

**Lemma 2.** All circuits whose associated graphs have minimum bisection width \( \omega \) have circuit area \( A \geq \frac{\omega^2}{4} \).

Proof: See Thompson [5].

We now discuss the notion of nested minimum bisection, a concept introduced by Grover et al. in [2] and also used in [4] which we again present here so the paper is self contained.

Suppose that a circuit has \( k \) output nodes. If the output nodes of such a circuit are minimum bisected, this results in two disconnected subcircuits each with, roughly, \( \frac{k}{2} \) output nodes. These two subcircuits can each have their output nodes minimum bisected again, resulting in four disconnected subcircuits, now each with roughly \( \frac{k}{4} \) output nodes.

**Definition 3.** This process of nested minimum bisections on a circuit, when repeated \( r \) times, is called performing \( r \)-stages of nested minimum bisections. In the case of this paper, the set of nodes to be minimum bisected will be the output nodes. We may also refer to this process as performing nested bisections, and a circuit under consideration in which nested bisections have been performed as a nested bisected circuit. Note that we will omit the term “minimum” in discussions of such objects, as this is implicit.

Note that associated with an \( r \)-stage nested bisected circuit are \( 2^r \) subcircuits. Note as well that once a subcircuit has only one node, it does not make sense to bisect that subcircuit again. Suppose we are nested-bisecting the \( k \) output nodes of a circuit. In this case, one cannot meaningfully nested-bisect the output nodes of a circuit \( r \) times if \( 2^r > k \).

Note that each of the \( 2^r \) subcircuits induced by the \( r \)-stage nested bisection may have some internal wires, and also wires that were deleted and connect to nodes in other subcircuits. We can index the \( 2^r \) subcircuits with the symbol \( i \).

**Definition 4.** Let the number of wires attached to nodes in subcircuit \( i \) that were deleted in the nested bisections be \( f_i \). This quantity is the fan-out of subcircuit \( i \).

We shall also consider the bits communicated to a given subcircuit.

**Definition 5.** Let \( b_i = f_i T \), where we recall that \( T \) is the number of clock cycles used in the running of the circuit under consideration. This quantity is called the bits communicated to the \( i \)th subcircuit.

We can now define an important quantity.

**Definition 6.** The quantity \( B_r = \sum_{i=1}^{2^r} b_i \) is the inter-subcircuit bits communicated.

Note that each subcircuit induced by the nested bisections will each have close to \( \frac{k}{2^r} \) output nodes within them (a consequence of choosing to bisect the output nodes at each stage), however, each may have a different number of input nodes.

**Definition 7.** This quantity is called the number of input nodes in the \( i \)th subcircuit and we denote it \( n_i \).

Note that \( \sum_{i=1}^{2^r} n_i = n \) for all valid choices of \( r \). That is, the sum over the number of input nodes in each subcircuit is the total number of input nodes in the original circuit.

This now allows us to present an important lemma.

**Lemma 3.** All fully-parallel circuits with inter-subcircuit bits communicated \( B_r \) have product \( AT^2 \) bounded by:
where we define \( c_1 = \frac{\sqrt{2}-1}{32} \).

**Proof:** This result, from Grover et al. [2], flows from applying Lemma [2] recursively on the nested-bisected structure and optimizing.  

**Lemma 4.** All fully-parallel circuits with inter-subcircuit bits communicated \( B_r \) and number of input nodes \( n \) have product \( AT \) bounded by:

\[
AT \geq c_2 \sqrt{\frac{n}{2r}} B_r
\]

where we define \( c_2 = \frac{\sqrt{2}-1}{4\sqrt{2}} \).

**Proof:** See [2]. This result flows from the observation that \( A \geq n \) for a fully parallel circuit and then combining this inequality with (1).  

**Definition 8.** An \((n,k)\)-decoder is a circuit that computes a decoding function \( f : \{0,1\}^n \rightarrow \{0,1\}^k \). It is associated with a codebook, (and therefore, naturally, an encoding function, which computes a function \( g : \{0,1\}^k \rightarrow \{0,1\}^n \)), a channel statistic, \( P(y^n|x^n) \) (which we will assume herein to be the statistic induced by \( n \) channel uses of a binary erasure channel), and a statistic from which the source is drawn \( p(x^k) \) (which we will assume to be the statistic generated by \( k \) independent fair binary coin flips). The quantity \( n \) is the block length of the code, and the quantity \( k \) is the number of bits decoded.

**Definition 9.** The block error probability of a decoder, denoted \( P_e \), is the probability that the decoder’s estimate of the original source is incorrect. Note that this probability depends on the source distribution, the channel, and the function that the decoder computes.

**Definition 10.** A decoding scheme is an infinite sequence of circuits \( D_1, D_2, \ldots \) each of which computes a decoding function, with block lengths \( n_1 < n_2 < \ldots \) and bits decoded \( k(n_1), k(n_2), \ldots \). They are associated with a sequence of codebooks \( C_1, C_2, \ldots \) and a channel statistic.

We assume throughout this paper that the channel statistic associated with each decoder is the statistic induced by \( n \) uses of a binary erasure channel. Our lower bound results also apply to any channel that is a degraded erasure channel, including the binary symmetric channel. Our results in terms of binary erasure probability \( \epsilon \) can be applied to decoding schemes for the binary symmetric channel with crossover probability \( p \) by substituting \( p = 2\epsilon \).

**Definition 11.** We let \( P_e(n) \) denote the block error probability for the decoder with input size \( n \). We let \( R(n) = \frac{k(n)}{n} \) be the rate of the decoder with input size \( n \).

We also classify decoding schemes in terms of how their probability of error scales in the definition below.

**Definition 12.** An \( f(n) \)-decoding scheme is a decoding scheme in which for sufficiently large \( n \) the block error probability \( P_e(n) < f(n) \).

**Definition 13.** The asymptotic-rate, or more compactly, the rate of a decoding scheme is \( \lim_{n \rightarrow \infty} R(n) \), if this limit exists, which we denote \( R \).

Note that the rate of a decoding scheme may not be the rate of any particular codebook in the decoding scheme.

**Definition 14.** An exponentially-low-error decoding scheme is an \( e^{-cn} \)-decoding scheme for some \( c > 0 \) with asymptotic rate \( R > 0 \).

We will also consider another class of decoding schemes, one which can be considered less reliable.

**Definition 15.** A polynomially-low-error decoding scheme is a \( \frac{1}{n^t} \)-decoding scheme for some \( t > 0 \) with asymptotic rate \( R > 0 \).

We will also need to define a sublinear function, which will be used to deal with a technicality in Theorem [1]

**Definition 16.** A sublinear function \( f(n) \) is a function in which \( \lim_{n \rightarrow \infty} \frac{f(n)}{n} = 0 \).

C. Main Lower Bound Results

We can now state the main theorem of this paper.
Theorem 1. All \( f(n) \)-decoding schemes associated with a binary erasure channel with erasure probability \( \epsilon \) in which \( f(n) \) monotonically decreases to 0 and in which \(-\ln(f(n))\) is a sublinear function have energy that scales as

\[
E \geq c_3 \sqrt{\frac{\ln(f(n))}{\ln(\epsilon)}} k
\]

(2)

where \( c_3 = \frac{\sqrt{n} \ln(\sqrt{2} - 1)}{16\sqrt{2}} \) and \( AT^2 \) complexity that scales as:

\[
AT^2 \geq c_4 \frac{k^2 \ln(f(n))}{n \ln(\epsilon)}
\]

(3)

for another positive constant \( c_4 = \frac{\ln(2)(\sqrt{2} - 1)^2}{32} \).

Proof: Associated with each decoder is its \( B_r \), the inter-subcircuit bits communicated. We can choose \( r \) to be any function of \( n \) so long as \( 2^r < n R(n) = k(n) \). From here on, we will suppress the dependence of \( r(n) \), \( k(n) \), and \( R(n) \) on \( n \). For ease of notation, let \( N = 2^r \) be the number of subcircuits induced by the \( r \)-stages of nested bisections. Consider any specific sufficiently large circuit in our decoding scheme, and suppose that \( B_r < \frac{k}{2} \). Then there exists at least \( \frac{N}{2} \) subcircuits in which \( b_i < \frac{k}{N} \) (where we recall \( b_i \) is the bits communicated to the \( i \)th subcircuit from Definition 5). Suppose not, i.e., that there are \( \geq \frac{N}{2} \) subcircuits with \( b_i \geq \frac{k}{N} \). Then, \( B_r \geq \frac{kN}{2} = \frac{k}{2} \), violating the assumption that \( B_r < \frac{k}{2} \). Call the set of at least \( \frac{N}{2} \) subcircuits with bits communicated to them less than \( \frac{k}{N} \) \( Q \). Using a similar averaging argument, we claim that within \( Q \) there must be one subcircuit in which \( n_i \leq \frac{2n}{N} \). If not, if all \( \frac{N}{2} \) subcircuits in \( Q \) have greater than \( \frac{2n}{N} \) input bits injected into them, then the total number of inputs nodes in the entire circuit is greater than \( \frac{2n}{N} N = n \), but there are only \( n \) input nodes in the entire circuit. Thus, there is at least one subcircuit in \( Q \) in which \( b_i < \frac{k}{N} \) and \( n_i \leq \frac{2n}{N} \).

Suppose that all the input bits injected into this special subcircuit are erased. Then, that subcircuit makes an error with probability at least \( \frac{1}{2} \) by Lemma 1 since it will have to form an estimate of \( \frac{k}{N} \) bits by only having injected into it fewer than \( \frac{k}{N} \) bits. Thus, if \( B_r \leq \frac{k}{2} \) then:

\[
P_e \geq P(\text{error}[\text{all } n_i \text{ bits erased}]) P(\text{all } n_i \text{ bits erased})
\]

\[
\geq \frac{1}{2} e^{n_i}
\]

where this first inequality flows from summing one term in a law of total probability expansion of the probability of block error, and the second from lower bounds on these probabilities.

Combining this observation with the fact the \( n_i \leq \frac{2n}{N} \) gives us the following observation:

\[
\text{if } B_r \leq \frac{k}{2} \text{ then } P_e \geq \frac{1}{2} e^{n_i} \geq \frac{1}{2} e^{\frac{2n}{N}}
\]

(4)

This is true for any valid choice of \( r \).

Now suppose that our decoding scheme is an \( f(n) \)-decoding scheme. We choose \( r \) to be

\[
r = \left \lfloor \log_2 \frac{2n \ln(\epsilon)}{\ln(2) \ln(f(n))} \right \rfloor
\]

so that

\[
N = 2^r \approx \frac{2n \ln(\epsilon)}{\ln(2) \ln(f(n))}.
\]

(5)

This is a valid choice of \( r \) because \( N \) cannot grow faster than \( O(n) \) because we assumed \( P_e(n) \) was monotonically decreasing (easily checked by inspection). Note as well that \( N \) increases with \( n \) because of the sub-linearity assumption of \( -\ln(f(n)) \). Then, if \( B_r \leq \frac{k}{2} \), by directly substituting into (4),

\[
P_e \geq \frac{1}{2} \exp \left( \frac{\ln(\epsilon)2\ln(2)n \ln(f(n))}{2n \ln(\epsilon)} \right)
\]

\[
= \frac{1}{2} \exp \left( \ln(2) \ln(f(n)) \right) = f(n).
\]

In other words, if \( B_r \leq \frac{k}{2} \) then our decoding scheme is not an \( f(n) \)-decoding scheme. Thus, for this choice of \( r \), \( B_r > \frac{k}{2} \).
Thus, by Lemma 4:

\[ E \geq c_2 \sqrt{\frac{n}{2^{\left\lfloor \frac{2 \ln(e)}{\ln(2) \ln(f(n))} \right\rfloor}}} k \]

\[ \geq c_2 \sqrt{\frac{n}{2^{\left\lfloor \frac{2 \ln(e)}{\ln(2) \ln(f(n))} \right\rfloor + 1}}} k \]

\[ \geq c_2 \sqrt{\frac{n}{2^{\left( \frac{2 \ln(e)}{\ln(2) \ln(f(n))} \right)}}} k \]

\[ \geq c_3 \sqrt{\frac{\ln(f(n))}{\ln(e)}} k \]

where we substituted the value for \( N \) in the first line, used the fact that \( \lfloor x \rfloor \leq x + 1 \) in the second, and simplified the lines that followed, proving inequality (2) of the theorem. As well, by Lemma 1, using \( B_r > \frac{k}{2} \) for this choice of \( r \), following a similar substitution as in the previous paragraph:

\[ AT^2 \geq c_1 \frac{B_r^2}{2^r} \]

\[ \geq c_1 \frac{k^2}{4 \left( 2^{\left\lfloor \frac{2 \ln(e)}{\ln(2) \ln(f(n))} \right\rfloor} \right)} \]

\[ \geq c_1 \frac{k^2}{4 \left( 2^{\left\lfloor \frac{2 \ln(e)}{\ln(2) \ln(f(n))} \right\rfloor + 1} \right)} \]

\[ = \frac{c_1 k^2}{8 \left( \frac{2 \ln(e)}{\ln(2) \ln(f(n))} \right)} \]

\[ = \frac{c_1 k^2 \ln(2)}{16 \ln(e)} \]

and the inequality in (3) flows from substituting the appropriate value for \( c_1 \) as defined in Lemma 1.

Corollary 1. All exponentially low error decoding schemes have energy that scales as

\[ E \geq \Omega \left( \frac{n^{\frac{3}{2}}}{p(n)} \right) \]

for all functions \( p(n) \) that increase without bound. In other words, all exponential probability of error decoding schemes have energy at least that scales very close to \( \Omega \left( n^{\frac{3}{2}} \right) \). Moreover, any such scheme that has energy that grows optimally, i.e. as \( AT = O \left( n^{\frac{3}{2}} \right) \), must have \( T(n) = \Omega \left( n^{0.5} \right) \).

Proof: Note that an exponentially low error decoding scheme has \( P_e \leq e^{-cn} \). Thus, such a scheme is also an \( e^{-c p(n)} \)-decoding scheme, for any increasing \( p(n) \). The result then directly flows by substituting \( f(n) = e^{-c p(n)} \) into (2) of Theorem 1.

For the second part of the corollary, suppose that for some constant \( c \), a decoding scheme has

\[ AT = \Theta \left( n^{\frac{3}{2}} \right) \]

We have as well from (3) and substituting \( f(n) = e^{-c p(n)} \)

\[ AT^2 \geq \Omega \left( \frac{n^2}{p(n)} \right) \]

where we use the fact that \( k = Rn \) (since by definition exponentially-low error decoding schemes have asymptotic rate greater than 0).

Suppose that

\[ T < O \left( \frac{n^{\frac{3}{2}}}{g(n)} \right) \]

for a \( g(n) \) that grows with \( n \), i.e., that \( T \) asymptotically grows slower than \( O \left( n^{\frac{3}{2}} \right) \). Then, to satisfy (7) we need
\[ AT^2 \geq \Omega \left( \frac{n^2}{p(n)} \right) \]  

for all increasing \( p(n) \), implying \[ A \geq \Omega \left( \frac{ng(n)^2}{p(n)} \right). \]

To see this precisely, suppose otherwise and then it is easy to see that, combined with Corollary 1, the inequality in (9) will be unsatisfied. If this is true, however, then the product
\[ AT \geq \Omega \left( \frac{ng(n)^2}{p(n)} \right) \frac{n^2}{g(n)} = \Omega \left( \frac{n^2 g(n)}{p(n)} \right). \]

Since this is true for all increasing \( p(n) \), it is true for, say, \( p(n) = \ln g(n) \), implying that the product \( AT \) grows strictly faster than \( \Omega \left( \frac{n^3}{p(n)} \right) \), contradicting the assumption of (6).

We generalize Corollary 1 to decoding schemes with different asymptotic block error probabilities below:

**Theorem 2.** All \( f(n) \)-decoding schemes with asymptotic rate greater than 0 in which \( f(n) \) is sub-exponential with energy that scales as \( E = \Theta \left( \sqrt{\ln f(n)n} \right) \) (that is, their energy matches the lower bound of (2) of Theorem 1) must have \( T(n) = \Omega \left( \sqrt{n f(n)} \right) \). Moreover, for all decoding schemes in which \( T(n) \) is faster than this optimal, \( E \geq \Omega \left( \frac{n \ln f(n)}{T(n)} \right) \).

**Proof:** Suppose that
\[ AT = \Theta \left( \sqrt{\ln f(n)n} \right) \]  

Note that from (3),
\[ AT^2 \geq \Omega \left( n \ln f(n) \right). \]  

As well, suppose \( T(n) \leq O \left( \sqrt{\ln f(n)/g(n)} \right) \) for some increasing \( g(n) \). Then, from the bound (11), \( A \geq \Omega \left( n \sqrt{\ln f(n) g^2(n)} \right) \) (to prove this, suppose otherwise and derive a contradiction). This implies then that \( AT \geq \Omega \left( \sqrt{\ln f(n)n g(n)} \right) \), contradicting (10).

Moreover, for all \( T(n) \) growing slower than that required for optimal energy, this implies that \( A \geq \Omega \left( \frac{n \ln f(n)}{T(n)} \right) \), which implies \( E \geq \Omega \left( \frac{n \ln f(n)}{T(n)} \right) \).

**Corollary 2.** All polynomially-low error decoding schemes have energy that scales at least as
\[ E \geq \Omega \left( n \sqrt{\ln n} \right). \]  

If this optimal is reached, then \( T(n) \geq \Omega(\sqrt{\ln n}). \)

**Proof:** This energy lower bound flows from letting \( f(n) = \frac{1}{n^2} \) and then substituting this value into (2). The time lower bound flows from directly applying Theorem 2.

**D. Serial Decoding Scheme Scaling Rules**

Let the number of output nodes in a particular decoder be denoted \( j \) (in a decoding scheme this will be a function of \( n \)).

**Definition 17.** A serial decoding scheme is one in which \( j \) is constant.

In [4] we considered the case of allowing the number of output nodes \( j \) to increase with increasing block length. We required an assumption that such a scheme be output regular, which we define below.

**Definition 18.** An output regular circuit is one in which each output node of the circuit outputs exactly one bit of the computation at specified clock cycles. This definition excludes circuits where some output nodes output a bit during some clock cycle and other output nodes do not during this clock cycle. An output regular decoding scheme is one in which each decoder in the scheme is an output regular circuit.

**Theorem 3.** All serial \( f(n) \)-decoding schemes have energy that scales as \( \Omega \left( n \ln f(n) \right) \).

**Proof:** The \( \Omega \left( n \ln f(n) \right) \) lower bound flows from following the arguments of the proof of Theorem 2 in [4], by showing that any decoding scheme in which the area scales less than \( O \left( \ln f(n) \right) \) cannot be an \( f(n) \)-decoding scheme.
Theorem 4. All output regular increasing-output node $f(n)$-decoding schemes have energy that scales as $\Omega \left( n \ln f(n) \right)^{\frac{1}{n}}$.

Proof: From the derivations preceding equation (13) in [4], following a similar argument as in this paper, we divide the circuit into $M = \Theta \left( \frac{n}{f} \right)$ epochs as before, and divide the subcircuits into $N = \Theta \left( \frac{A}{\ln f(n)} \right)$ subcircuits through nested bisections. With this choice, we can follow the same arguments used in Theorem 3 in [4], and derive that all $f(n)$-decoding schemes must have

$$AT \geq \Omega \left( n \ln (f(n)) \right)^{\frac{1}{n}}.$$ 

IV. INFORMATION FRICTION IN THREE-DIMENSIONAL CIRCUITS

The “information friction” computational energy model was introduced by Grover in [3] and further studied by Vyavahare et al. in [16] and Li et al. in [17]. We generalize (and slightly modify) this model to three dimensions and use a similar approach to Grover to obtain some non-trivial lower bounds on the energy complexity of three dimensional bit-meters decoder circuits, in terms of block length and probability of error. We will discuss how this approach can be generalized to models in arbitrary numbers of dimensions. We present the model below and then prove our main complexity result.

- A circuit is a grid of computational nodes at locations in the set $\mathbb{Z}^3$, where $\mathbb{Z}$ is the set of integers. Some nodes are inputs nodes, some are output nodes, and some are helper nodes. Note that Grover [3] considers this model in terms of a parameter characterizing the distance between the nodes, but since we are concerned with scaling rules, we will assume that they are placed at integer locations, allowing us to avoid unnecessary notation. The Grover paper considered scaling rules in which nodes are placed on a plane, in which the number of dimensions $d = 2$. In our results we will discuss the case of $d = 3$ and afterwards discuss how the approach can be generalized to an arbitrary number of spatial dimensions.
- A circuit is to compute a function of $n$ binary inputs and $k$ binary outputs.
- At the beginning of a computation, the $n$ inputs to the computation are injected into the input nodes. At the end of the computation the $k$ outputs should appear at an output node. A node can be both input and output.
- A node can communicate messages along its links to any other node, and can receive bits communicated to them from any other node.
- Each node has constant memory, and can compute any computable function of all the inputs it has received throughout the computation that is stored in their memory, to produce a message that it can send to any other node.
- We associate a computation with a directed multi-graph, that is, a set of edges linking the nodes. For every computation, there is one edge per bit communicated along a link in the computation’s associated multi-graph. The “cost” of an edge in such a multi-graph is the Euclidean distance between the two nodes that it connects. Note that if a node communicates $m$ bits to another node in a computation, then that computation’s associated multi-graph must have $m$ edges connecting the two nodes. This multi-graph is called a computation’s communication multi-graph.
- The energy, or the bit-meters, denoted $\beta$ of a computation is the sum of the costs of all the edges in the computation’s associated multi-graph (that is, the sum of the Euclidean distances of all the edges).

We consider a grid of three-dimensional cubes, with “inner cubes” nested within them. This object is a generalization of the “stencil” object defined by [3].

Definition 19. An $(L, \lambda)$—nested cube grid is an infinite grid of cubes, with side length $L$ and inner cube side length $L (1 - 2\lambda)$. Note that the inner cubes are centered within the outer cubes. Fig. 2 shows a diagram of one cube in a $(L, \lambda)$—nested cube grid, to which the reader can refer to visualize this nested cube structure. A set of nested cube grid parameters is valid if $L > 0$ and $0 < \lambda < \frac{1}{2}$.

Note that a nested cube grid can be placed conceptually on top of a bit meters circuit. We will consider placing a nested cube grid in parallel with the Cartesian 3-space that defines our circuit. We can specify the position of a nested cube grid that is parallel to a set of Cartesian coordinates by calling one of the corners of an outer cube the origin, and then specify the location of its origin. A particular set of parameters for a nested cube grid and a location for its origin (called its orientation) induces a set of subcircuits, defined below.

Definition 20. A subcircuit, associated with a particular orientation of a nested cube grid, is the part of a bit-meters circuit within a particular outer cube.

Nodes in any subcircuit can thus be considered to be either inside an inner cube or outside an inner cube. For any circuit with finite number of nodes there will thus be some cubes that contain computational nodes, and some that do not. We can label the subcircuits that contain nodes with the index $i$. The number of input nodes in cube $i$ we denote $n_i$. The number of output nodes in subcircuit $i$ we denote $k_i$. Furthermore, we denote the number of input nodes within the inner cube of subcircuit $i$ as $k_{in,i}$.

Definition 21. We define $k_{in} = \sum k_{in,i}$, which is the number of output nodes within inner cubes, which we will often simply refer to with the symbol $k_{in}$. 

We will show in Lemma 6 that there exists a nested cube grid orientation in which $k_{in}$ is high.

**Definition 22.** The internal bit meters of a subcircuit $i$ is the length of all the communication multigraph edges completely within subcircuit $i$, plus the length of the parts of the edges within subcircuit $i$. This quantity is denoted with the symbol $\beta_i$. Note that $\beta = \sum_{\text{all subcircuits } j} \beta_j$ (where we may have to sum over some subcircuits that do not contain any nodes).

Since a computation has associated with it its communication multi-graph, for a given subcircuit we can consider the subgraph formed by all the paths that start outside of the cube and end inside the inner cube. We can group all the vertices of this graph that start outside the outer cube and call this the source, and group all vertices inside an inner cube and call it the sink. For this graph we can consider its min-cut, the minimum set of edges that, once removed, disconnects the source from the sink.

**Definition 23.** The number of bits communicated from outside a cube to within an inner cube, or, bits communicated, is the size of this minimum cut. For a particular subcircuit $i$ we refer to this quantity with the symbol $b_i$.

Remark 2. This quantity is analogous (but not the same) as the quantity $b_i$ for the Thompson circuit model from Definition 5, and thus we use the same symbol. The reader should not confuse these symbols; the Thompson model definition applies to discussions in Section III, and the bit-meters model definition applies in this section, Section IV.

If the $n_i$ internal bits of a subcircuit are fixed, then the subcircuit inside an inner cube will compute a function of the messages passed from outside the outer cube. Clearly, the size of the set of possible messages injected into this internal cube is $2^{b_i}$ (since $b_i$ is the min cut of the paths leading from outside to inside.)

**Lemma 5.** All subcircuits with bits communicated $b_i$ have internal bit meters at least $b_i\lambda L$.

**Proof:** This result flows from Menger’s Theorem [18], [19], which states that any network with min-cut $b_i$ has at least $b_i$ disjoint paths from source to sink. Each of these paths must have length at least $\lambda L$ from the triangle inequality. 

Remark 3. This lemma makes rigorous the idea that to communicate $b_i$ bits from outside a subcircuit to within its inner square, the bit-meters this takes is proportional to the distance from outside an outer square to within an inner square $(\lambda L)$ and the number of bits communicated.

In the lemma below we show that there exists an orientation of any nested cube grid such that $k_{in}$ is high.

**Lemma 6.** For all three dimensional bit-meters circuits with $k$ output nodes, all valid nested cube grid parameters $L$ and $\lambda$, there exists an orientation of an $(L, \lambda)$-nested cube grid in which the number output nodes within inner cubes ($k_{in}$) is bounded by:

$$k_{in} \geq (1 - 2\lambda)^3 k$$

Remark 4. Note that the relative volume of the inner cubes is $(1 - 2\lambda)^3$. This lemma says there exists an orientation of any nested cube grid in which the fraction of output nodes within inner cubes is at least this fraction, so this result is not surprising.

**Proof:** This is a natural generalization of the Grover result (See Lemma 2 of [3]), which uses the probabilistic method. We consider placing the origin of an $(L, \lambda)$-nested cube grid uniformly randomly within a cube of side length $L$ centered at the origin in the Cartesian 3-space. We index the $k$ output nodes by $i$. Let $1_{in,i}$ be the indicator random variable that is equal to 1 if output node $i$ is within an inner cube. Then, given the uniform measure on the position of the cube, the quantity $k_{in}$
is a random variable. We observe:

\[
\begin{align*}
k_{\text{in}} &= \sum_{i=1}^{k} 1_{\text{in},i}, \text{ thus } \\
E(k_{\text{in}}) &= E\left(\sum_{i=1}^{k} 1_{\text{in},i}\right) \\
&= \sum_{i=1}^{k} E(1_{\text{in},i}) \\
&= \sum_{i=1}^{k} (1 - 2\lambda)^3 \\
&= k(1 - 2\lambda)^3
\end{align*}
\]

where in (13) we use the observation that, for each output node, the probability that it is in an inner square is proportional to the relative area of the inner square. Thus, the expected value of \(k_{\text{in}}\) is \(k(1 - 2\lambda)^3\) and so there must be at least one nested cube grid orientation in which \(k_{\text{in}}\) is greater than or equal to that value.

\[\text{Lemma 7. For all valid nested cube parameters } L \text{ and } \lambda, \ n_i \leq (L + 1)^3 \text{ and thus for sufficiently large } L \ n_i \leq 2L^3.\]

\[\text{Proof: Intuitively, there cannot be more than on the order of } L^3 \text{ inner nodes in a cube of volume } L^3. \text{ The } (L + 1)^3 \text{ bound comes from considering the corner case of a cube whose sides exactly touch output nodes.} \]

We can now state the main results of this section.

\[\text{Theorem 5. All 3D-bit-meters decoders for a binary erasure channel with erasure probability } \epsilon \text{ of sufficiently large block length with block error probability } P_e \text{ have bit-meters } \beta \text{ bounded by:} \]

\[\beta > \frac{27}{512} \left( \frac{\ln (4P_e)}{2\ln(\epsilon)} \right)^{\frac{2}{3}} k.\]

\[\text{Proof: We consider the number of bits communicated from outside a subcircuit } i \text{ to within the inner cube of subcircuit } i \text{ (} b_i \text{). It must at least be } k_{\text{in},i} \text{ to overcome the case that all the input nodes in the entire cube are erased. If this does not happen, then one of the output nodes must guess at least one bit, making an error with probability at least } \frac{1}{2}, \text{ formally justified by Lemma 4. This allows us to argue that:} \]

\[P_e \geq P(\text{error|all } n_i \text{ output bits are erased}) \geq \frac{1}{2} e^{n_i}. \]

If \(\beta < \lambda L k_{\text{in}}\) then there exists a subcircuit indexed by \(i\) in which \(b_i < k_{\text{in},i}\). Suppose otherwise, i.e. that \(b_i \geq k_{\text{in},i}\) for all \(i\), then:

\[\beta \geq \sum_{\text{all subcircuits } i} \lambda L b_i = \lambda L \sum b_i \geq \lambda L \sum k_{\text{in},i} = \lambda L k_{\text{in}}\]

where we apply Lemma 5 after the first inequality, and for convenience suppress the subscript on the summation sign after the first instance. This contradicts our assumption that \(\beta < \lambda L k_{\text{in}}\).

We choose the parameter \(L\) in terms of probability of error in order to derive a contradiction if a circuit does not have high enough bit-meters. Specifically, we choose

\[L = \left( \frac{\ln (4P_e)}{2\ln(\epsilon)} \right)^{\frac{2}{3}}. \]

Consider the nested cube structure that has \(k_{\text{in}} \geq (1 - 2\lambda)^3 k\) that must exist by Lemma 6. If \(\beta \leq \lambda L k_{\text{in}}\) then there must exist a subcircuit \(i\) that has less than \(k_{\text{in},i}\) bits injected into it from outside the subcircuit to within its inner cube. Thus:

\[\text{if } \beta \leq \lambda L k_{\text{in}} \text{ then } P_e \geq \frac{1}{2} e^{n_i} \geq \frac{1}{2} e^{2L^3} \geq 2P_e\]
where (a) flows from \(14\), (b) from Lemma \(7\) and (c) from the evaluation of this expression by substituting \(15\). This is a contradiction. Thus, all bit meters decoders must have

\[
\beta > \lambda L k_{in} \\
\beta > \lambda (1 - 2\lambda)^{3} L k \\
\geq \lambda (1 - 2\lambda)^{3} \left( \frac{\ln(4P_{e})}{2\ln(\epsilon)} \right)^{\frac{1}{4}} k.
\]

The second inequality flows from the fact that we are considering the nested cube structure in which \(k_{in} \geq (1 - 2\lambda)^{3} k\) that must exist by Lemma \(6\). We may choose any valid \(\lambda\) to maximize this bound, and letting \(\lambda = \frac{1}{8}\) gives us:

\[
\beta > \frac{27}{512} \left( \frac{\ln(4P_{e})}{2\ln(\epsilon)} \right)^{\frac{1}{4}} k.
\]

Remark 5. Note that this argument naturally generalizes to \(d\)-dimensional space, in which all \(d\)-dimensional bit-meters decoders have energy that scales as \(\beta \geq \Omega \left( \left(\ln(P_{e})\right)^{\frac{1}{2}} k \right)\). The key step in the proof to be altered is in a modification of Lemma \(7\) and a choice of \(L = c \left( \frac{\ln(4P_{e})}{\ln(\epsilon)} \right)^{\frac{1}{4}}\) in line \(15\) of the proof for some constant \(c\) that may vary depending on the dimension. This implies, among other things, that exponentially low probability of error decoding schemes implemented in \(d\)-dimensions have bit-meters energy that scales as \(\Omega \left( n^{1 + \frac{1}{d}} \right)\). Obviously, the most engineering-relevant number of dimensions \(d\) for this type of analysis are \(d = 2\) and \(d = 3\).

### V. Encoder Lower Bounds

In terms of scaling rules, all the decoder lower bounds presented herein can be extended to encoder lower bounds. The main structure of the decoder lower bounds (inspired by \(2, 3\)) involves dividing the circuit into a certain number of subcircuits. Then, we argue that if the bits communicated within the circuit is lower, there must be one subcircuit where the bits communicated to it are less than the bits it is responsible for decoding. If all the inputs bits in that circuit are erased, the decoder must make an error with probability at least \(1/2\).

In the encoder case, we also take inspiration from \(2, 3\). In this case, the \(n\) outputs of the encoder circuit can be divided into a certain number of subcircuits. Then we consider the bits communicated out of each subcircuit. This quantity must be proportional to the number of output bits in each subcircuit. Otherwise, there will be at least one subcircuit where the number of bits communicated out is less than the number of output nodes in the subcircuit. Call these bits that were not fully communicated out of this subcircuit \(Q\). Suppose that once the output bits of the encoder are injected into the channel, all the bits in \(Q\) are erased. Now, the decoder must use the other bits of the code to decode. But, the subcircuit containing \(Q\) in the encoder communicated less than \(|Q|\) bits to the other outputs of the encoder. By directly applying Lemma \(1\), we see that no matter what function the decoder computes, it must make an error with probability at least \(1/2\). An argument of this structure and following exactly the structure of Theorems \(1, 2, 3, 4, 5\) for the decoders gives us the following theorems, whose proofs are omitted.

**Theorem 6.** All fully-parallel \(f(n)\)-encoding schemes with number of clock cycles \(T(n)\) have energy

\[
E(n) \geq \Omega \left( \frac{n \log(f(n))}{T(n)} \right)
\]

with optimal lower bound of \(E \geq \Omega \left( n \sqrt{\log f(n)} \right)\) when \(T(n) \geq \sqrt{\log(f(n))}\).

All serial, \(f(n)\)-encoding schemes have energy that scales as

\[
E(n) \geq \Omega \left( n \log(f(n)) \right).
\]

All increasing output node, output-regular \(f(n)\)-encoding schemes have energy that scales as

\[
E(n) \geq \Omega \left( n^{\log^{1/5} (f(n))} \right).
\]

Finally, all three-dimensional, bit-meters encoding schemes associated with block error probability \(P_{e}\) have energy that scales

\[
E(n) \geq \Omega (n \ln P_{e}).
\]
VI. LIMITATIONS OF RESULTS

There are a number of weaknesses in the models we have used. Firstly, our results are asymptotic. For some set block error probability and rate, there may be a specific circuit that reaches this block error probability using a circuit design methodology that does not generalize to scale in a way as predicted by our theorems.

Note that our quantity $T$ refers to number of clock cycles, which reflects one of the main “time costs” in a circuit computation. In real circuits, the “time cost” of a computation involves two parameters: the number of clock cycles required, and the time it takes to do each clock cycle. In our model, we do not consider the time per clock cycle. In real circuits, this quantity often varies with wire lengths. We do not consider this in our model.

A particular weakness of the Thompson model we use is that it does not consider a quantity called switching activity factor. In circuit design, this quantity is the fraction of the circuit that “switches” during the course of the computation. And yet, our model assumes a switching activity factor of 1. Thus, in terms of scaling rules, the Thompson model should be considered applicable only to computational schemes in which the switching activity factor does not change with increasing input sizes. On the other hand, the information-friction model accounts for the possibility of schemes in which switching activity factor changes with increasing block length, so, combined with the results of Grover, we conjecture that there may be similar tradeoffs we derive apply.

VII. OTHER ENERGY MODELS OF COMPUTATION

There has been some work on energy models of computation different from the Thompson energy models and Grover information friction models, and herein we provide a short review.

In [20], Bingham et al. classify the tradeoffs between the “energy” complexity of parallel algorithms and “time” complexity for the problem of sorting, addition, and multiplication using a model similar to, but not the same as the model we use. In the grid model used by these authors, a circuit is composed of processing elements laid out on a grid, in which each element can perform an operation. In this model the circuit designer chooses over the speed of each operation, but this comes at an energy cost. Real circuits run at higher voltages can result in lower delay for each processing element but higher energy [21]. The model used by the authors in [20] captures some of this fundamental tradeoff. Note that our model assumes constant voltage. Non-trivial results that show how real energy gains can occur by lowering voltages in decoder circuits have been studied in [22], but we do not study this here.

Another energy model of computation was presented by Jain et al. in [23]. This model introduced an augmented Turing machine, a specialization of the traditional Turing machine [24]. The authors introduce a transition function, mapping the current instruction being read, the current state, the next state and the next instruction to the “energy” required to make this transition. This model (once the transition function is clearly defined for a specific processor architecture) would be good for the algorithm designer at the software level. However, we do not believe this model informs the specialized circuit designer. The Thompson model which we analyze, on the other hand, can include, as a special case, the energy complexity of algorithms implemented on a processor, as our model allows for a composition of logic gates to form a processor.

Landauer [25] derives that the energy required to erase one bit of information is at least $kT \ln 2$, where $k$ is Boltzmann’s constant, and $T$ is the temperature. Thus, a fundamental limit of computation comes from having to erase information. Of course, it may be possible to do reversible computation in which no information is erased that can use arbitrarily small amounts of energy, but such circuits must be run arbitrarily slowly. This suggests a fundamental time-energy tradeoff different from the tradeoff discussed herein. Landauer [26], Bennett [27] and Lloyd [28] provide detailed discussions and bibliographies on this line of work. Demaine et al. [29] extract a mathematical model from this line of work and analyze the energy complexity of various algorithms within this model. Note that the Thompson model we use is one informed by how modern VLSI circuits are created, even though they operate at energies far above ultimate physical limits.

VIII. FUTURE WORK

Currently, our work on lower bounds has not been extended to other channels, like the additive white Gaussian noise channel. Perhaps more interesting, however, is the question, do there exist polynomially low probability of error decoding schemes with energy that closely matches of Corollary 2, i.e., one with energy that scales as $\Omega \left( \frac{1}{\sqrt{n}} \right)$? This may have significantly lower energy than an exponentially-low error decoding scheme, and may provide sufficient error control performance. We do not know whether such a decoding scheme exists and this remains an important open question. It may be that decoding strategies with energy that scales like this are already invented but have simply not been analyzed in terms of their energy complexity.

The decoding problem for communication systems is a special case of the more general problem of inference. Well known algorithms used for inference, for example the Sum-Product Algorithm [30] and variational methods [31], include Gallager’s low-density parity-check decoding algorithms as a special case [32]. Thus, we conjecture that there may be similar tradeoffs between energy, latency, and reliability in circuits that perform inference.
REFERENCES

[1] A. El Gamal, J. Greene, and K. Pang, “VLSI complexity of coding,” The MIT Conf. on Adv. Research in VLSI, 1984.
[2] P. Grover, A. Goldsmith, and A. Sahai, “Fundamental limits on the power consumption of encoding and decoding,” in Proc. 2012 IEEE Int. Symp. Info. Theory, 2012, pp. 2716–2720.
[3] P. Grover, “Information friction and its implications on minimum energy required for communication,” IEEE Trans. Inf. Theory, vol. 61, no. 2, pp. 895–907, Feb 2015.
[4] C. G. Blake and F. R. Kschischang, “Energy consumption of VLSI decoders,” IEEE Trans. Inf. Theory, vol. 61, no. 6, pp. 3185–3198, June 2015.
[5] C. D. Thompson, “Area-time complexity for VLSI,” Proc. 11th Ann. ACM Symp. Theory of Comput., pp. 81–88, 1979.
[6] Y. Xie, J. Cong, and S. S. Sapatnekar, Three-dimensional integrated circuit design: EDA, design and microarchitectures. New York, NY, USA: Springer Verlag, 2010.
[7] J. E. Savage, “Complexity of decoders: I-classes of decoding rules,” IEEE Trans. Inf. Theory, vol. 15, no. 6, pp. 689–695, Nov 1969.
[8] ——, “The complexity of decoders – part ii: Computational work and decoding time,” IEEE Trans. Inf. Theory, vol. 17, no. 1, pp. 77–85, January 1971.
[9] C. G. Blake and F. R. Kschischang, “On the energy complexity of LDPC decoder circuits,” CoRR, vol. abs/1502.07999, Feb. 2015. [Online]. Available: http://arxiv.org/abs/1502.07999
[10] K. Ganesan, P. Grover, J. Rabaei, and A. Goldsmith, “On the total power capacity of regular-ldpc codes with iterative message-passing decoders,” Selected Areas in Communications, IEEE Journal on, vol. 34, no. 2, pp. 375–396, Feb 2016.
[11] C. G. Blake and F. R. Kschischang, “On scaling rules for energy of VLSI polar encoders and decoders,” 2016, in preparation.
[12] S. Lovett and E. Viola, “Bounded-depth circuits cannot sample good codes,” 2012, available at author’s homepage: http://www.ccs.neu.edu/home/viola/papers/LoV.pdf.
[13] K. L. Rychkov, “A modification of khrapchenko’s method and its applications to bounds on the complexity of pi-schemes and coding functions,” Met. Disk. Anal. Theor. Graph. Skhem., vol. 42, pp. 91–98, 1985.
[14] A. Kojevnikov and A. S. Kulikov, “Lower bounds on formula size of error-correcting codes,” 2007, unpublished manuscript, available at author’s homepage: http://logic.pdmi.ras.ru/ arist/papers/hamming.pdf.
[15] P. Vayvalhare, M. Mahzoon, P. Grover, N. Limaye, and D. Manjunath, “Information friction limits on computation,” in Communication, Control, and Computing (Allerton), 2014 52nd Annual Allerton Conference on, Sept 2014, pp. 93–100.
[16] T. Li, M. Bakshi, and P. Grover, “Energy-efficient decoders for compressive sensing: Fundamental limits and implementations,” CoRR, vol. abs/1411.4253, 2015. [Online]. Available: http://arxiv.org/abs/1411.4253
[17] K. Meng, “Zur allgemeinen kurventheorie,” Fund. Math., vol. 10, pp. 96–115, 1927.
[18] F. Göring, “Short proof of menger’s theorem,” Discrete Mathematics, vol. 219, pp. 295–296, 2000.
[19] B. D. Bingham and M. R. Greenstreet, “Modeling energy-time trade-offs in VLSI computation,” IEEE trans. Computers, vol. 61, no. 4, April 2012.
[20] B. Hoeneisen and C. A. Mead, “Fundamental limitations in microelectronics – i. MOS technology,” Solid-State Electronics, vol. 15, pp. 819–829, 1972.
[21] F. Leduc-Primeau, F. R. Kschischang, and W. Gross, “Modeling and energy optimization of LDPC decoder circuits with timing violations,” CoRR, vol. abs/1503.03880, 2015. [Online]. Available: http://arxiv.org/abs/1503.03880
[22] R. Jain, D. Molnar, and Z. Ramzan, “Towards a model of energy complexity for algorithms [mobile wireless applications],” in 2005 IEEE Wireless Communications and Networking Conference, vol. 3, March 2005, pp. 1884–1890.
[23] A. M. Turing, “On computable numbers, with an application to the entscheidungsproblem,” Journal of Math, vol. 58, 1936.
[24] R. Landauer, “Irreversibility and heat generation in the computing process,” IBM Journal of Research and Development, vol. 5, no. 3, pp. 183–191, July 1961.
[25] ——, “Dissipation and noise immunity in computation and communication,” Nature, vol. 335, no. 27, Oct. 1988.
[26] C. H. Bennett, “The thermodynamics of computation - a review,” International Journal of Theoretical Physics, vol. 21, no. 12, 1982.
[27] S. Lloyd, “Ultimate physical limits to computation,” Nature, vol. 406, August 2000.
[28] E. D. Demaine, J. Lynch, G. J. Mirano, and N. Tyagi, “Energy-efficient algorithms,” in Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, ser. ITCS ’16. New York, NY, USA: ACM, 2016, pp. 321–332. [Online]. Available: http://doi.acm.org/10.1145/2840728.2840750
[29] F. Kschischang, B. Frey, and H. A. Loeliger, “Factor graphs and the sum-product algorithm,” IEEE Trans. Inf. Theory, vol. 47, no. 2, pp. 498–519, Feb 2001.
[30] M. J. Wainwright and M. I. Jordan, “Graphical models, exponential families, and variational inference,” Foundations and Trends in Machine Learning, vol. 1, no. 1–2, 2008.
[31] R. Gallager, “Low-density parity-check codes,” IRE Trans. Inf. Theory, vol. 8, no. 1, pp. 21–28, 1962.