On the VLSI Energy Complexity of LDPC Decoder Circuits

Christopher G. Blake and Frank R. Kschischang, *Fellow, IEEE*

***Abstract*— Sequences of randomly generated bipartite configurations are analyzed; under mild conditions almost surely such configurations have minimum bisection width proportional to the number of vertices. This implies an *almost sure* $\Omega(n^2/d_{\max}^2)$ scaling rule for the energy of directly-implemented low-density parity-check (LDPC) decoder circuits for codes of block length n and maximum node degree $d_{\max}$. It also implies an $\Omega(n^{3/2}/d_{\max})$ lower bound for serialized LDPC decoders. It is also shown that *all* (as opposed to almost all) capacity-approaching, directly-implemented non-split-node LDPC decoding circuits, have energy, per iteration, that scales as $\Omega\left(\chi^2 \ln^3 \chi\right)$, where $\chi = (1 - R/C)^{-1}$ is the reciprocal gap to capacity, R is code rate, and C is channel capacity.**

***Index Terms*— LDPC codes, circuits, energy, complexity.**

I. Introduction

THIS paper uses an adaptation of Thompson's [1] VLSI model to derive lower bounds on the VLSI energy complexity of low-density parity-check (LDPC) codes, an important family of error control codes introduced by Gallager [2].

The first result is an "almost-sure" scaling rule for the energy complexity of LDPC decoders. In particular, we analyze ensembles generated according to a uniform configuration distribution. We show, subject to some mild conditions, that the minimum bisection width of a randomly-generated bipartite configuration asymptotically almost surely has minimum bisection width proportional to the number of vertices. This implies an $\Omega(n^2/d_{\max}^2)$ lower bound on the energy of directly-implemented LDPC decoders (see Definition 5) and a $\Omega(n^{3/2}/d_{\max})$ lower bound on the energy of serialized decoders (see Definition 14).

We also show that a capacity-approaching sequence of "non-split-node directly-implemented" LDPC decoders (see Definition 8) must have energy that scales as $\Omega(\chi^2 \ln^3(\chi))$, where $\chi = (1 - R/C)^{-1}$ is the *reciprocal gap to capacity*, where R is the code rate and where C is the capacity of the channel over which the code is transmitted. This lower bound contrasts with the universal lower bound of $\Omega(\chi^2\sqrt{\ln(\chi)})$ derived in [3].

Manuscript received February 25, 2015; revised September 23, 2016; accepted January 22, 2017. Date of publication February 23, 2017; date of current version April 19, 2017. This paper was presented in part at the 2014 IEEE North American School of Information Theory and the 2015 IEEE International Symposium on Information Theory.

The authors are with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: christopher.blake@mail.utoronto.ca; frank@comm.utoronto.ca).

Communicated by H. Pfister, Associate Editor for Coding Theory.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2017.2673805

The $\Omega(\chi^2 \ln^3(\chi))$ result applies to decoding circuits where messages are passed on a Tanner graph induced by a parity-check matrix of the underlying code. This lower bound does not apply to decoding algorithms that use modified Tanner graphs with punctured variable nodes like those used for the non-systematic irregular repeat accumulate (IRA) codes of [4] or the compound low-density generator matrix (LDGM) codes of [5]. However, computations show that the $\Omega(n^2/d_{\max}^2)$ and $\Omega(n^{3/2}/d_{\max})$ almost sure lower bounds apply to the non-systematic IRA construction of [4] for many parameters.

We begin the paper in Section II with a discussion of prior related work. In Section III we introduce the main definitions and the circuit model considered. Then, in Section IV, after defining some properties of node-degree lists, we present the main theorem. We proceed to show how this theorem allows us to find scaling laws for the energy of LDPC decoders in Section V. In Section VI we derive a $\Omega\left(\chi^2 \ln^3 \chi\right)$ scaling rule for capacity-approaching sequences of non-split-node LDPC decoders. In Section VII we discuss some open problems related to this work and in Section VIII we make some concluding remarks.

II. Prior Work

A. Related Work on Circuit Complexity and LDPC Codes

Ganesan *et al.* [6] assume that the average wire length in a VLSI instantiation of a Tanner graph is proportional to the longest wire, and that the length of the longest wire is proportional to the diagonal of the circuit upon which the LDPC decoder is laid out. The implication of these assumptions is an $\Omega\left(n^2\right)$ scaling rule for the area of directly-implemented LDPC decoders, which is the same result as Corollary 2 of this paper. However, these assumptions are taken as axioms without being fully justified; there certainly can exist bipartite Tanner graphs that can be instantiated in a circuit without such area. We show that, in fact, the $\Omega\left(n^2\right)$ scaling rule is justified for *almost all* directly-implemented Tanner graphs (so long as some mild conditions are satisfied).

More recently, Ganesan *et al.* [7] analyze the VLSI complexity of certain classes of LDPC decoding algorithms, including how the number of iterations required for such algorithms scales with block error probability. Moreover, the authors show that a judicious choice of node-degree distributions can optimize the total (transmit + decoding) power for coded communication using LDPC codes by simulating real circuits and their code performance. The Ganesan *et al.* paper complements our paper; we do not analyze how the number of iterations depends on target block error probability,

nor do we simulate any actual circuit performance. Neither the Ganesan *et al.* paper nor our paper consider the performance of LDPC codes whose interconnection complexity, and not just degree distribution, is optimized. This open problem is discussed further in Section VII.

B. Related Work on Graph Theory

This paper uses a combinatorial approach to derive almost sure lower bounds on the minimum bisection width of randomly generated configuration. This contrasts with a common approach that considers a graph's Laplacian (See [8, Definition 8.6.15]). Fiedler [9], shows that the second largest eigenvalue of a graph's Laplacian, λ_2, can be used to find a lower bound of $\frac{\lambda_2 n}{4}$ on the graph's minimum bisection width. Bezrukov *et al.* [10] find bounds on the bisection width of graphs that are related to this λ_2 value. Diaz *et al.* [11] provide almost sure upper bounds for the bisection width of randomly generated regular graphs. Luczak and McDiarmid [12] also study the minimum bisection width of graphs generated according to a distribution different from ours. Furthermore, our analysis is of random *bipartite* graphs, as opposed to random regular graphs. As well, our result makes only weak assumptions on the node degree distribution, without requiring a degree-regularity assumption, in contrast to previous work.

III. Preliminaries

A. Graph Theory Definitions

The main result of our paper involves the minimum bisection width (MBW) of a graph.

Definition 1: Let $G = (V, E)$ be a graph and let $V' \subseteq V$. A subset of the edges $E_s \subseteq E$ *bisects* V' in G if removal of E_s cuts V into unconnected sets V_1 and V_2 in which $\left|\left|V_1 \cap V'\right| - \left|V_2 \cap V'\right|\right| \leq 1$. The sets $V_1 \cap V'$ and $V_2 \cap V'$ are considered the *bisected sets of vertices*. A *minimum bisection* is a bisection of a graph whose size is minimum over all bisections. The *minimum bisection width of* V' is the size of a minimum bisection of V'. The *minimum bisection width* of the graph is the minimum bisection width of all the vertices V. We let $\phi_G(V')$ denote the minimum bisection width of a set of vertices V' in G.

Note that finding the minimum bisection width of a graph is NP-Complete [13].

B. Circuit Model

In this paper, the definition of a *circuit* is adapted from Thompson [14] and is considered to be a mathematical object consistent with the circuit axioms which we specify in [3]. Readers should consult this reference to understand precisely the model that we are discussing; we review briefly the main parts of this model below, as well indicate some minor simplifications introduced in this paper.

- A *circuit* is a grid of squares with computational nodes, wires, and wire crossings. Wires connect computational nodes in the circuit. The nodes compute functions of their binary inputs synchronously at each clock cycle. Wires are bidirectional: we assume that they may pass a message in both directions each clock cycle. Inputs are injected into input nodes and outputs appear at output nodes.
- One of the key circuit parameters we consider is the circuit area (A) which is equal to the number of grid squares occupied. Note that in [3] we define area as the product of the number of grid squares occupied times the square of the wire width. Since we are concerned with scaling rules, in this paper we just assume that the wire width is unity.
- Another key circuit parameter is τ, the number of clock cycles used in the computation.
- The energy of a computation is defined as $E = A\tau$. Note again that in [3] we borrowed notation from [15] relating energy and the area-time product by a proportionality constant; for simplicity in this paper we just assume that this proportionality constant is unity.

The modified Thompson model that we consider in this paper is meant to subsume the wiring complexity as a fundamental cost of computation. In the field of error control coding, this interconnection complexity has been shown to be a significant factor in the energy of a computation in [16] and [17]. Though this model assumes a planar implementation, [1] shows that if we allow L layers, this can decrease circuit area by a factor of at most L^2; thus if the model allows a constant number of layers L, our lower bound results can be modified by a factor of L^2.

C. Relationship Between Circuit Model and Graphs

Note that a circuit is a collection of nodes connected by wires. Each of the computational nodes of a circuit can be thought of as a vertex of a graph, $G = (V, E)$. However, since more than one wire can connect two nodes, this object may actually be a multi-graph. The wires of a circuit correspond to the edges of the graph. In particular, two vertices v_1 and v_2 are connected in the graph G by an edge if and only if there is a wire connecting the two computational nodes that correspond to v_1 and v_2. We let $d_{\max}(G)$ denote the maximum node degree of a graph G.

D. LDPC Decoders

LDPC codes are linear codes first studied by Gallager [2]. Given a parity-check matrix H with m rows for a code of length n, the *Tanner graph* of H is a bipartite graph where one part of the graph contains n vertices called *variable nodes* and the other part is composed of m *check nodes*. Each check node is associated with a row of the parity-check matrix and each variable node with a column. A check node is connected to a variable node if and only if the row associated with the check node has a 1 in the column corresponding to the variable node.

Since there are many possible parity-check matrices for a given linear code, there are many possible Tanner graphs associated with that code. An LDPC decoding algorithm for a code is a message-passing procedure where messages are passed over the edges of a particular Tanner graph of the code.

We consider two possible paradigms to implement LDPC decoding algorithms with a circuit: a *directly-implemented* and

a *serialized* technique. To be precise, we will use the following graph-theoretic terminology.

Definition 2 ([8, Definition 2.2.7]): The *contraction* of an edge e connecting vertices u and v is the removal of e and the replacement of u and v with a vertex whose incident edges are the same edges incident on u and v, except of course e.

Definition 3: The reverse of edge contraction is *vertex splitting*. This is a process that replaces a vertex v with two vertices v_1 and v_2 with an edge between them. For every edge incident on v there is exactly one edge either connecting to v_1 or to v_2. We say that v_1 and v_2 are *split* from vertex v.

Definition 4 ([8, Definition 6.2.13]): A graph G is a *minor* of a graph G' if the graph G can be obtained by deleting vertices and edges of G' and contracting edges of G'. We say in this case that G' *contains* G *as a minor*. Equivalently, G' contains G as a minor if there is a sequence of vertex splits of G that results in a subgraph of G'.

Definition 5: A circuit is a *directly-implemented LDPC decoder* associated with an LDPC code with a Tanner graph T if its graph contains T as a minor.

Definition 5 allows Tanner graph computational nodes to be split and thus logic gates associated with the computation for a single check or variable nodes can appear in different parts of the circuit.

Definition 6: Let G' be obtained by successively splitting the vertices of a graph G with labelled vertices. When a labelled vertex is split, move the label arbitrarily to one of the new vertices. The labelled vertices of G' we term the *vertices of G' corresponding to G*, or the *G-corresponding vertices*.

Definition 7: Let G' be obtained by successively splitting the vertices of a graph G. Consider a vertex v in the graph G. Then its *v-descendants* in G' are those vertices that were originally v, or split from v, or split from a vertex that was split from v, and so on. Consider a labelling of the G-corresponding vertices of G'. A particular labeled vertex v is considered the *parent* of all those vertices descended from the vertex of G from which v descended.

Definition 8: Consider a circuit which contains Tanner graph T as a minor. Consider each set of vertices descended from a vertex in T. If the wires connecting these vertices do not cross any other wires in the circuit, then such a circuit is said to be a *non-split-node directly-implemented LDPC decoder*.

Lemma 1: Let v be a G-corresponding vertex of G', where G' is obtained from G by vertex splitting. Then the number of edges leading from the descendants of v to the rest of the graph is not more than $d_{\max}$ (in fact, not more than the degree of the original vertex).

Proof: This is easily observed by drawing a circle around a node and then successively splitting this vertex. The number of edges exiting the circle does not increase as the vertices are split. ∎

Definition 9: We let $A_{\min}(G)$ be the minimum circuit area of a circuit whose associated graph is G.

The following lemma adapted from Thompson [1] is important for our discussion:

Lemma 2: If a graph G has minimum bisection width $\phi_G(V')$ for a set V' of vertices, then the area of a circuit implementing this graph is lower bounded by

$$A_{\min}(G) \geq \frac{\phi_G^2(V')}{4}.$$

Proof: Thompson's proof for the minimum bisection width of all the vertices of the graph [1, Th. 2] applies just as well to the minimum bisection width of a subset of the vertices. ∎

Let G' be a graph obtained by vertex splitting G. We let V'_G be the G-corresponding vertices of G' so that $\phi_{G'}(V'_G)$ is the minimum bisection width of the G-corresponding vertices of G'. This is of course dependent on how the nodes are labeled during the splitting process. Thus to be more precise this represents the smallest MBW of the nodes corresponding to G over all labellings.

The graph for a directly-implemented LDPC decoder is obtained by vertex splitting and adding edges and vertices to the original Tanner graph. Adding edges and vertices cannot decrease the MBW of V'_G, but vertex splitting might. However, vertex splitting can only decrease the MBW of V'_G by a limited amount, as the following lemma proves.

Lemma 3: If G' is obtained by a sequence of vertex splits of a graph $G = (V, E)$ with no isolated vertices, and G has maximum node degree $d_{\max}(G)$, then

$$\phi_{G'}(V'_G) \geq \frac{\phi_G(V)}{d_{\max}(G)+1} \geq \frac{\phi_G(V)}{2d_{\max}(G)}.$$

Proof: Suppose not, *i.e.,* that $\phi_{G'}(V'_G) < \frac{\phi(G)}{1+d_{\max}(G)}$. Note that G can be obtained by contracting the vertices of G'. Consider a minimum bisection of the G-corresponding vertices of G', and place one side of the bisection on the left side and the other the right side. There will be some vertices on the left side that are descended from vertices on the right side, and vice versa. We call such vertices *bisection-crossing descendants*.

If such a bisection of V'_G has ω' edges crossing it, then there are at most ω' G-corresponding vertices of G' that have bisection-crossing descendants. To see this, observe that the set of descendants of a vertex must be connected by paths using only their vertices, so there must be at least one unique edge crossing the bisection for each G-corresponding vertex of G' that has a bisection-crossing descendant.

We shall now show how to construct a bisection of G with at most $\omega'(1 + d_{\max}(G))$ edges. Simply move all the bisection-crossing descendants of G to the side of their parent, while keeping the G-corresponding vertices of G' on the same side of the bisection. Then contract all the vertices that were split in obtaining G' from G. By Lemma 1, for each H-corresponding vertex of G, we observe that at most $d_{\max}$ edges will connect the descendants of v to vertices on the opposite side of the bisection. Thus, moving the vertices to the side of their descendant can at most add $\omega' d_{\max}(G)$ edges crossing the bisection, and the resulting bisection has width at most $\omega' + \omega' d_{\max}$.

Thus, we have constructed a bisection of G of width less than $\frac{\phi_G(V)}{1+d_{\max}(G)}(1 + d_{\max}(G)) = \phi_G(V)$, a contradiction. ∎

Note that if a graph G'' contains G as a minor, then it contains a subgraph G' that is obtained by a sequence of vertex splits of G. This allows us to conclude:

Lemma 4: If a graph G'' contains a graph $G = (V, E)$ as a minor, then the minimum bisection width of the nodes of G' corresponding to G is at least $\phi_G(V)/(1 + d_{\max}(G))$. Thus, by applying Lemma 2, the circuit area of G'' is at least

$$A_{\min}(G'') \geq \frac{\phi_G^2(V)}{4(1 + d_{\max}(G))^2} \geq \frac{\phi_G^2(V)}{16 d_{\max}^2(G)}$$

E. Serialized LDPC Decoders

Not all LDPC decoders are directly-implemented. This motivates considering a more general class of LDPC decoder. Our definition of a serialized circuit includes both serialization of the message-passing step (for example, by introducing an interleaver that works over multiple clock cycles to pass messages from node to node), and serializing computation steps (by having one computational node perform the computation for multiple check or variable nodes, but at different clock cycles). The key idea is that a serialized circuit *simulates* a *joined Tanner graph*, which we will define in this section.

To do so we first define a computation's communication multi-graph.

Definition 10: The *communication directed multigraph*, or *communication graph* for a circuit operated for τ clock cycles is the graph obtained by replacing each computational node with a vertex and replacing each wire between two computational nodes (u and v, say) with 2τ edges, τ of them directed from u to v and τ of them directed from v to u.

Definition 11: Two unconnected vertices v_1 and v_2 of a graph can be *joined* by removing the two vertices and replacing them with a single vertex v. Each edge connecting v_1 or v_2 to a vertex (denoted a) in the original graph is replaced with an edge connecting v to a.

Definition 12: A graph obtained by first splitting the nodes of a Tanner graph T and then joining nodes that are not associated with variable node inputs is a *joined Tanner graph* obtained from T.

For a joined Tanner graph T', we let $j_{\max}(T')$ be the maximum number of vertices joined to form a single vertex. Often its dependence on T' will be suppressed.

Definition 13: A communication graph K *simulates* a graph G if there is a subset of vertices of K in a one-to-one correspondence with the vertices of G, and for each edge in G, there is a path connecting the two corresponding vertices in K. Moreover, these paths are mutually edge-disjoint.

We can now define a serialized LDPC decoder.

Definition 14: A *serialized LDPC decoder* for a Tanner graph T is a circuit that simulates a joined Tanner graph obtained from T during each iteration.

Note that if a particular node of such a circuit corresponds to a vertex formed by joining j nodes then there must be at least j clock cycles performed each iteration.

In the sections that follow, we prove an $\Omega(n^2/d_{\max}^2)$ scaling rule for the energy of directly-implemented LDPC decoder circuits in Corollary 2 and an $\Omega(n^{3/2}/d_{\max})$ lower bound for serialized LDPC decoders in Theorem 2.

IV. Main Theorem

Our main theorem is fundamentally graph-theoretic in nature and applies to graphs generated according to a standard uniform random configuration distribution.

Definition 15: Consider the set of bipartite graphs $G = (V_L \sqcup V_R, E)$ (where the symbol $\sqcup$ is the *disjoint union symbol*) in which $|V_L| = n$, $|V_R| = m$. Let V_L be called the left nodes and V_R the right nodes. Order the left nodes and right nodes in terms of increasing node degree. Let l_i be the degree of the ith left node in the graph, and let r_i be the degree of the ith right node in the graph. Then we say that $L = (l_1, l_2, \ldots, l_n) \in (\mathbb{N})^n$ is the *left node degree list* and $R = (r_1, r_2, \ldots, r_m) \in (\mathbb{N})^m$ the *right node degree list*.

Note that the node degree lists are non-standard; often it is the node degree distribution that is considered. However, in Appendix A we show how to present our results in terms of the more standard node degree distributions.

Given a list $Z = \{z_1, z_2, \ldots, z_n\}$ with $z_1 \leq z_2 \leq \ldots \leq z_n$, we define

$$S_{\text{top}}(Z) = \sum_{i=\lfloor \frac{n}{2} \rfloor}^{n} z_i. \tag{1}$$

Note that implicitly this function takes as input the size of the input vector.

Denote the set of bipartite graphs with left and right node degree lists L and R as $\mathcal{G}(L, R)$. Note that the number of edges in each particular graph in $\mathcal{G}(L, R)$ is $|E| = \sum_{i=1}^{n} l_i = \sum_{i=1}^{m} r_i$.

For convenience of counting, we will consider not the set of graphs with a particular degree list, but rather the set of *configurations* with this degree list. We can associate each node in a graph with a number of sockets equal to its degree. This node and socket configuration model is a standard way to consider the set of bipartite graphs that form the Tanner graphs of LDPC ensembles, and in particular is discussed thoroughly in [18].

Definition 16: A set of left nodes and right nodes with an ordered labeling of the sockets of each node, together with a permutation mapping the left node sockets to the right node sockets is called a *configuration*.

Let the set of configurations with node degree lists L and R be denoted $\mathcal{B}(L, R)$. Clearly, $|\mathcal{B}(L, R)| = |E|!$. Since a configuration is merely a graph with a labeling of sockets for each node, graph properties, including minimum bisection width, can be extended to describe configurations in the natural way.

Let

$$B_a = \{G \in \mathcal{B}(L, R) : \exists \text{ a bisection } K \subseteq E \text{ such that } |K| = a\}$$

be the set of configurations that have a bisection of size a. Note that B_a does not represent the set of configurations in $\mathcal{B}(L, R)$ with *minimum* bisection width a, but rather the set of configurations with *some* bisection of size a.

Let B_a^* be the set of configurations in $\mathcal{B}(L, R)$ that have a bisection of size a or less, *i.e.*,

$$B_a^* = \bigcup_{i=0}^{a} B_i.$$

Given a left node node degree list L of length n and right node degree list R of length m, where L and R are ordered by increasing degree and $m \leq n$, we define

$$\delta(L, R) = \frac{\max\left(S_{\text{top}}(L), S_{\text{top}}(R)\right)}{n} \tag{2}$$

We also let

$$\sigma(L, R) = \frac{|E|}{n} - \delta(L, R).$$

For notational convenience we will abbreviate these two quantities as δ and σ and their dependence on the node degree distribution under discussion is to be implicit. Note that $|E| = \delta n + \sigma n$.

Consider a configuration with left degree list L and right degree list R. For a given subset of vertices V' we can divide this set into two disjoint sets, $V'_L = V' \cap V_L$ and $V'_R = V' \cap V_R$. Let $S_{\text{left}}\left(V'\right) = \sum_{v \in V'_L} \deg(v)$ and $S_{\text{right}}\left(V'\right) = \sum_{v \in V'_R} \deg(v)$ denote the number of left and right sockets in V', respectively.

Lemma 5: For any bipartite configuration $G = (V_L \sqcup V_R, E)$ with left degree list L and right degree list R, where $|V_L| = n$ and $|V_R| = m$, for any collection V' of $\frac{n+m}{2}$ vertices, $\min\left(S_{\text{left}}(N_V), S_{\text{right}}(N_V)\right) \leq n\delta$.

Proof: Suppose not. This implies that both $S_{\text{left}}(V') > n\delta$ and $S_{\text{right}}(V') > n\delta$. Divide the vertices in V' into the left nodes V'_L and right nodes V'_R. It must be that $\left|V'_L\right| + \left|V'_R\right| = \frac{n+m}{2}$. Thus, it must be that $\left|V'_L\right| \leq \frac{n}{2}$ or $\left|V'_R\right| \leq \frac{m}{2}$ (otherwise their sum would exceed $\frac{m+n}{2}$). Let us consider the case in which $\left|V'_L\right| \leq \frac{n}{2}$ (the other case leads to an analogous argument). If $\left|V'_L\right| \leq \frac{n}{2}$ and $S_{\text{left}}(V') > n\delta$ then, in particular $S_{\text{left}}\left(V'\right) > S_{\text{top}}(L)$. But $S_{\text{top}}(L)$ by (1) is the sum of the highest degree left nodes. A collection of at most half these nodes cannot exceed this quantity, leading to a contradiction. ∎

We will need the following lemma for our proof:

Lemma 6: The quantity $m!n!$, subject to the conditions $0 \leq n \leq m \leq Z \leq m + n \leq Y$ and that Y, Z, m and n are all integers cannot exceed $Z!(Y - Z)!$.

Proof: See Appendix B. ∎

We can now give the main technical lemma of this paper, which states that the set of configurations with a small bisection is small, which will imply that with high probability a Tanner graph has MBW proportional to n.

Lemma 7: If a configuration $G = (V_L \sqcup V_R, E)$ with $|V_L| = n$ and with degree lists L and R is generated according to the uniform configuration distribution, then the probability that this configuration is in the set B_a^* when

$$0 \leq a \leq \sigma(L, R)n \tag{3}$$

is upper bounded by

$$P\left(B_a^*\right) \leq \frac{(a+1)\, n^2 \binom{n}{\frac{n}{2}}^2 \binom{|E|}{a}^4 a!\,(\delta(L,R)n)!\,(\sigma(L,R)n - a)!}{(\delta(L,R)n + \sigma(L,R)n)!}.$$

Proof: This follows from a counting upper-bounding argument, where the key idea is to overcount a set of objects that is larger than B_a^*, namely the set of "quadrant configurations" with a bisection of size a or less.

Let the set of configurations in $\mathcal{B}(L, R)$ having a bisection of size a be denoted by B_a. Then we can say that, according to the uniform configuration distribution, the probability of the event of generating a configuration with a bisection of size a is given by:

$$P(B_a) = \frac{|B_a|}{|E|!},$$

recalling that $|E|$ is the number of edges in the configurations of $\mathcal{B}(L, R)$.

We will now bound the number of configurations in $\mathcal{B}(\Lambda, P)$ with a bisection of size a, and we will assume that $a < \sigma n$. To do so, we will define a "quadrant configuration", show that the number of quadrant configurations with a bisection of size a is greater than or equal to B_a, and then upper bound the number of quadrant configurations with a bisection of size a or less.

A *quadrant configuration* of a bipartite configuration $G = (V_L \sqcup V_R, E)$ is an ordered-tuple $Q = (G, T_L, T_R, B_L, B_R)$ where the vertices are divided into 4 disjoint sets, the *top left nodes* (T_L), the *top right nodes* (T_R), the *bottom left nodes* (B_L), and the *bottom right nodes* (B_R), in which $T_L, B_L \subseteq V_L$, $T_R, B_R \subseteq V_R$ and $||T_R \cup T_L| - |B_R \cup B_L|| \leq 1$. Vertices in T_L and T_R are considered to be *top nodes* or and similarly for the bottom nodes.

Note that every bipartite graph has at least one quadrant configuration induced by arbitrarily dividing the vertices in half, and denoting one half of these vertices top nodes and the other half bottom nodes. Thus, the set of quadrant configurations with a particular degree distribution is at least as big as the set of configurations with a particular degree distribution. Because a quadrant configuration $Q = (G, T_L, T_R, B_L, B_R)$ contains a graph G, graph properties can be extended to describe a quadrant configuration.

Denote the set of quadrant configurations with set node degree lists L and R in which a is the number of edges connecting top nodes to bottom nodes as Q_a. Note that the dependence of Q_a on a particular node degree distribution is implicit. Observe that every configuration with a bisection of size a has a corresponding quadrant configuration in Q_a created in the natural way by denoting one bisected set of vertices as the top nodes, and the other the bottom nodes. Thus $|B_a| \leq |Q_a|$.

For ease of discussion, we will assume that the total number of nodes $m + n$ in the set of configurations under discussion is even, so that $\frac{m+n}{2}$ is an integer.

Denote the set of quadrant configurations in Q_a in which there are i top left nodes and j edges connecting top left nodes to the bottom right by $Q_a^{i,j}$. This of course implies that there are $\frac{m+n}{2} - i$ top right nodes and $a - j$ edges leading from the bottom left to the top right nodes. We can see in Figure 1 an example of such an element that we are counting for the case of $n = 8$ and $a = 4$, $i = 4$ and $j = 2$. Note then that

$$Q_a = \bigcup_{i=0}^{n} \bigcup_{j=0}^{a} Q_a^{i,j}.$$

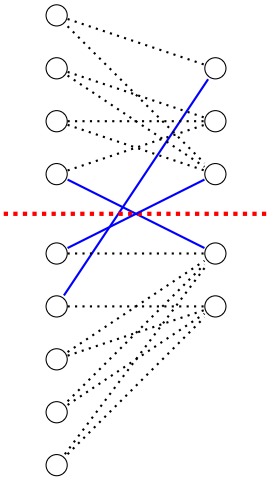


Fig. 1. An example of a particular quadrant configuration associated with a left node degree list L in which all the nodes have 2 sockets and a right node degree list $R = (2, 3, 4, 4, 5)$. The number of left nodes $n = 9$ and the number of right nodes $m = 5$. The configuration drawn is a quadrant configuration in $Q_3^{4,1}$ for the particular degree lists. Recall that the superscript denotes that there are $i = 4$ top left nodes and $j = 1$ edges leading from top left nodes to bottom right nodes. The subscript indicates that there are $a = 3$ edges between top and bottom vertices. The edges forming the bisection are solid lines. The horizontal dotted line indicates where the bisection occurs.

We bound the size of $Q_a^{i,j}$ by counting all quadrant configurations with a bisection of size a that are the edges connecting top nodes to bottom nodes. In the following, for compactness, we let $\delta = \delta(L, R)$ and $\sigma = \sigma(L, R)$. We have

$$\left|Q_a^{i,j}\right| \leq \underbrace{\binom{n}{i}}_{a}\underbrace{\binom{m}{\frac{m+n}{2}-i}}_{b} \cdot \underbrace{\binom{|E|}{j}\binom{|E|}{j}\binom{|E|}{a-j}\binom{|E|}{a-j}}_{c} \cdot \underbrace{(j)!\,(a-j)!}_{d}\underbrace{(\delta n)!\,(\sigma n - a)!}_{e}, \tag{4}$$

where

a) Represents a choice of i top left nodes.
b) Represents a choice of $\frac{m+n}{2} - i$ top right nodes.
c) The quantity $\binom{|E|}{j}$ is an upper bound on the number of choices of j sockets that will have edges that cross the bisection line chosen from the top variable nodes, and $\binom{|E|}{j}$ is an upper bound on the number of choices for the bottom right sockets to which these edges will be connected. For a configuration in $B_a^{i,j}$ there must also be $a - j$ edges leading from the bottom left to the top right. The quantity $\binom{|E|}{a-j}$ is an upper bound on the number of choices of sockets in the bottom left that can have edges crossing the middle bisection, and similarly $\binom{|E|}{a-j}$ is an upper bound on the number of choices for the sockets connected in the top right.
d) Counts the number of permutations of edges that join the top half to the bottom half (first counting the j connections from the top left nodes to the bottom right nodes, then the $a - j$ connections from the bottom left nodes to the top right nodes).
e) This step involves permuting the connections of the remaining sockets in the top half and the bottom half. However, at this point it is not clear how many sockets are in the top half or the bottom half. However, we can upper bound the number of permutations possible. The number of sockets available in the top left nodes must equal the number of sockets available in the top right nodes (because in order to construct a valid configuration this must be true). By construction, the total number of nodes in the top left and top right is $\frac{m+n}{2}$, and thus the number of sockets available cannot exceed δn, by Lemma 5. Suppose the number of sockets available for all the top left nodes is M and the sockets available in the bottom left nodes is N. Then there are at most $M!N!$ ways to permute these. We also know that $M + N = |E| - a$ (since the total number of sockets available on one side of the constructed quadrant configuration is $|E|$ and a have been used to cross between top nodes and bottom nodes), and that $M \leq \delta n$ and $N \leq \delta n$. Subject to these restrictions, a direct application of Lemma 6 implies $M!N! \leq (\delta n)!\,(|E| - \delta n - a)! = (\delta n)!\,(\sigma n - a)!$

The proof mostly follows from simplification of these bounds and the details of the rest of the proof are given in Appendix C. ∎

Consider a sequence of random configurations $G_1, G_2, \ldots$ where each G_i in the sequence is a configuration generated according to the uniform configuration distribution, in which the ith configuration is drawn according to node degree lists L_i and R_i. Note that the randomness for each element of such a sequence does not come from the degree lists: we are assuming that these lists are fixed. It is the interconnections between nodes that is random. We specifically concern ourselves with a sequence in which the number of left nodes n increases without bound. For such a sequence, denote the number of left nodes of the ith configuration as n_i. We will abbreviate the quantities $\delta(L_i, R_i)$ and $\sigma(L_i, R_i)$ with the symbols δ_i and σ_i respectively. We let $B_{a,i}^*$ be the set of configurations with node degree lists (L_i, R_i) with a bisection of size a or less.

Theorem 1: Consider a sequence of right and left node degree lists (L_i, R_i), and a sequence bipartite configurations G_i where G_i is generated according to the random configuration distribution with node degree list (L_i, R_i). If

$$\limsup_{i\to\infty}\left[2\ln(2) + \delta_i\left(\ln\left(\frac{\delta_i}{\delta_i+\sigma_i}\right)\right) + \sigma_i\left(\ln\left(\frac{\sigma_i}{\delta_i+\sigma_i}\right)\right)\right] < 0 \tag{5}$$

then there exists some $\beta > 0$ in which

$$\lim_{i\to\infty} P\left(\{G_i \in B_{\beta n_i,i}^*\}\right) \to 0$$

where $\{G_i \in B_{\beta n_i,i}^*\}$ is the event that the ith configuration has a bisection of size βn_i or less. In particular, this is true for

any $0 < \beta < \sigma$ *that satisfies:*

$$\lim_{i\to\infty} 2\ln(2) + 4\mathcal{H}\left(\frac{\beta}{\delta_i+\sigma_i}\right) + \beta\left(\ln\left(\frac{\beta}{\sigma_i-\beta}\right)\right) + \delta_i\left(\ln\left(\frac{\delta_i}{\delta_i+\sigma_i}\right)\right) + \sigma_i\left(\ln\left(\frac{\sigma_i-\beta}{\delta_i+\sigma_i}\right)\right) < 0. \quad (6)$$

Remark 1: Stated less precisely, this theorem says that in the limit, a random bipartite configuration will, with high probability, have no small bisections.

Proof: This proof involves algebraic manipulation of the expression in Lemma 7 and showing that if the conditions of the theorem are satisfied, the limit evaluates to 0. The details of this computation are given in Appendix D. ∎

In the corollaries that follow, we consider a sequence of configurations generated according to the uniform configuration distribution. Let ϕ_i be the *minimum bisection width of the* i*th configuration*. Note that this symbol is a random variable. Theorem 1 now has an obvious corollary.

Corollary 1: If there is a sequence of configurations as described in Theorem 1, in which the condition in (5) is satisfied then $\lim_{i\to\infty} P(\phi_i \geq \beta n_i) = 1$ *for some* $\beta > 0$.

Proof: Note that B_a^* is the event that a random configuration has a bisection of size a or less. The complement of this event is the event that a random configuration has no bisection of size a or less, and thus equal to the event that a random configuration has minimum bisection width greater than or equal to a. The corollary flows directly from this observation. ∎

Remark 2: This Corollary and the results that follow can be slightly strengthened, because we know that the probability that a bisection exists with size less than βn approaches 0 exponentially quickly. Let $I_{\phi_i/n_i<\beta}$ be the event that the graph with n_i left nodes has a bisection less than βn. We easily observe that $\sum_n P(\phi_i/n_i > \beta) < \infty$ and so by the Borel-Cantelli Lemma, the probability that a bisection of size less than βn_i occurs infinitely often is 0. Thus, $P(\liminf_{i\to\infty} \phi_i/n_i \geq \beta) = 1$ for some $\beta > 0$.

V. Almost Sure Bounds on Sufficiently High Rate LDPC Decoder Circuits

To apply our results to LDPC decoder circuits, we first define a few terms in order to make our claims precise.

Definition 17 [19]: For a given parity check matrix H for a code of block length n and rate R, we define $\Delta(H)$ as the number of 1s in the matrix divided by nR, and call this quantity the *density* of the matrix H.

Definition 18: For a code of rate R associated with a channel with capacity C, let $\chi = (1 - R/C)^{-1}$ be the *reciprocal gap to capacity*.

Definition 19: Consider a sequence of codes and decoders for a particular channel. We let the block error probability of the ith code in the sequence be $P_{\mathrm{e},i}$. Then such a sequence is *vanishing-error-probability* if $\lim_{i\to\infty} P_{\mathrm{e},i} = 0$.

The following result, which is a simple implication of Sason and Urbanke [19] which we present using our notation shows that as capacity is approached the density of a code's parity check matrix must approach infinity. We will use this result in Corollary 2 and Theorem 3.

Lemma 8 ([19, Th. 2.1]): Consider a sequence of parity check matrices $\{H_i\}$ *for a channel with capacity* C. *Let* $\{\chi_i\}$ *denote the reciprocal gap to capacity of the* i*th code. Let the density of the* i*th parity matrix be* $\Delta(H_i)$. *Suppose that there is a decoder for each of these codes, and thus each code for matrix* H_i *has an associated block error probability (*$P_{\mathrm{e},i}$*). Suppose as well that in the limit of increasing* i $P_{\mathrm{e},i}$ *approaches* 0. *Then, there is some constant* K_1 *such that, for sufficiently large* i,

$$\Delta(H_i) > K_1 \ln(\chi_i).$$

A. Energy Complexity of Directly-Implemented LDPC Decoders

Corollary 2: Consider a vanishing-error-probability LDPC coding scheme where each code in the scheme is generated according to a uniform configuration distribution. Suppose that each decoder in the scheme is a directly-implemented LDPC decoder. If such a scheme has asymptotic rate sufficiently close to capacity, then for this scheme $\lim_{i\to\infty} P\left(A_i \geq cn_i^2/d_{\max}^2\right) = 1$ *for some constant* $c > 0$, *where* $d_{\max}$ *is the maximum node degree (possibly a function of* n*). Energy is bounded similarly.*

Proof: Note that Lemma 8 implies that as rate approaches capacity, the parity-check matrix density must approach infinity. But this implies that the associated Tanner graph has number of edges per node approaching infinity. Then obviously the quantity δ must approach infinity. We can use this observation that for codes of sufficient closeness to capacity the expression

$$2\ln(2) - (\delta+\sigma)\mathcal{H}\left(\frac{\delta}{\delta+\sigma}\right) < 0 \quad (7)$$

must be satisfied.

To see this, note that δ approaches ∞ for a capacity-approaching code. What happens to σ is either (a) $\lim_{n\to\infty} \frac{\delta}{\delta+\sigma} < 1$ or (b) $\lim_{n\to\infty} \frac{\delta}{\delta+\sigma} = 1$, or (c) this limit does not exist. Note that this value cannot exceed 1 because necessarily $\sigma \leq \delta$.

In the case of (c), it must be that the value of σ alternates and no limit can be defined. In this case, however, we should consider the specific subsequence of decoders in which either (a) or (b) applies. It will be clear that since for each subsequence the appropriate scaling rule holds, thus it must be true for the entire sequence.

In case (a): in the limit, $\ln\left(\frac{\delta}{\delta+\sigma}\right) < 0$ and so $\delta\left(\ln\left(\frac{\delta}{\delta+\sigma}\right)\right) \to -\infty$, as δ approaches ∞. Since $\sigma\left(\ln\left(\frac{\sigma}{\delta+\sigma}\right)\right) < 0$ in any case (a consequence of $\sigma \leq \delta$), thus in the limit the inequality (7) will be satisfied.

For case (b), in which $\ln\left(\frac{\sigma}{\delta+\sigma}\right) \to -\infty$, note that σ is positive, so $\sigma\left(\ln\left(\frac{\sigma}{\delta+\sigma}\right)\right) \to -\infty$, and thus in the limit (7) will also be satisfied.

If the scheme has asymptotic rate sufficiently close to capacity, then for sufficiently large block lengths in this

scheme the node degree list satisfies the sufficient condition of Theorem 1, and the code's Tanner graph has MBW at least βn_i for some $\beta > 0$ with probability approaching 1. In this case, the decoder, which contains a subgraph obtained by splitting the vertices of the Tanner graph, but have MBW at least $\beta n_i/(2d_{\max})$ by Lemma 3. In this case, Lemma 4 implies $A_i \geq (\beta n_i)^2/(16d_{\max}^2)$ and thus:

$$\lim_{i\to\infty} P\left(A_i \geq \frac{(\beta n_i)^2}{16 d_{\max}^2}\right) = 1$$

as expressed in the theorem statement. A similar bound is obviously then true for the energy per iteration of such circuits. ∎

B. Serialized-Check Node Decoders

In this section we generalize our results to serialized circuits. To develop this theory, however, we need to define some new terminology. In particular, we will generalize the notion of minimum bisection width by considering collections of bipartitions of the nodes of a graph.

Definition 20: A *bipartition* of a set X is the partition of the set into two disjoint sets X_1 and $X \setminus X_1$.

We will represent a bipartition by a single set contained within it.

Definition 21: Given a set of vertices V of a graph G, a *bisection of* V is a bipartition of V into V_1 and V_2 such that $||V_1| - |V_2|| \leq 1$.

We see that a bisection is an example of a bipartition. What we will be interested in is collections of bipartitions that are "zig-zaggable". It is the zig-zaggable property of the bisections of a graph that allows Thompson to prove in [1] that $A \geq \phi_G^2(V)/4$ for a circuit with graph $G = (V, E)$ with MBW $\phi_G(V)$.

Definition 22: Let X be a nonempty finite set. If $\emptyset \subseteq A \subset B \subseteq X$, a *simple chain* from A to B is a sequence $S_1 \subset S_2 \subset \ldots \subset S_L$ with $S_1 = A$ and $S_L = B$ and $|S_{i+1} \setminus S_i| = 1$ for $i = 1, 2, \ldots, L-1$.

Consider a subset (denoted $\mathcal{C}$) of the bipartitions of a set X.

Definition 23: A subset of the bipartitions of a set X is *zig-zaggable* if the following conditions hold:

1) All simple chains from $\emptyset$ to X contain an element of $\mathcal{C}$.
2) If A and B are subsets of X such that $A \subseteq B$, and there is a set D in $\mathcal{C}$ such that $A \subseteq D \subseteq B$, then all simple chains from A to B contain an element of $\mathcal{C}$.

Lemma 9: The collection of bisections of a set are zig-zaggable.

Proof: A set C induces a bisection of X if an only if $|C| = \lfloor |X|/2 \rfloor$ or $|C| = \lceil |X|/2 \rceil$. A simple chain from $\emptyset$ to X results in a sequence of bipartitions where the size of one of the sets of the bipartitions increases by 1 each time. One of these bipartitions must thus be a bisection. For property 2, suppose that a simple chain from A to B contains a set C that induces a bisection. Then, either A or B are bisections, or they are not and then $|A| < \lfloor |X|/2 \rfloor$ and $|B| > \lceil |X|/2 \rceil$, and then any simple chain from A to B will include a bisection. ∎

We will show however that a more general collection of bipartitions is zig-zaggable.

Definition 24: The *width* of a bipartition of a set of vertices of a graph is the number of edges connecting the vertices between the two sets of the bipartition.

Definition 25: The $\mathcal{C}$-bipartition width of a graph with respect to a collection of bipartitions $\mathcal{C}$ is the minimum width of all bipartitions in $\mathcal{C}$.

Using the definition of zig-zaggable, we can now easily adapt Thompson's proof [1] and derive the following lemma:

Lemma 10: Let $\mathcal{C}$ be a zig-zaggable collection of bipartitions of a graph G, and let $\omega_{\mathcal{C}}$ be the $\mathcal{C}$-bipartition width of the graph. Then

$$A_{\min}(G) \geq \frac{\omega_{\mathcal{C}}^2}{4}.$$

Proof: A detailed proof is given in Appendix E that essentially follows the proof of Thompson [1, Th. 2]. The author constructs on the order of $\omega_{\mathcal{C}}$ bisections of the nodes by drawing zig-zags across the circuit, each of which have on the order of $\omega_{\mathcal{C}}$ wires crossing them. These bisections must exist precisely because of the zig-zaggable property of the bisections of the graph. Thus, this proof extends to any zig-zaggable collection of bipartitions. ∎

Consider a joined Tanner graph as in Definition 12. Such a graph is obtained by splitting a Tanner graph T to obtain T' and then joining vertices. We can assign to each vertex of the joined Tanner graph a number equal to the number of T-corresponding vertices of T' that were joined in forming it. Each of these values is the *weight* of the vertex. The weight of T-corresponding vertices that were not joined are assigned the value 1, and the others are given weight 0. For a vertex v we let $w(v_i)$ be its weight.

Definition 26: A *κ-weighted bisection* of a collection of positive weighted nodes V is a bipartition $\{V_1, V_2\}$ of the vertices such that $\left|\sum_{v\in V_1} w(v_i) - \sum_{v\in V_2} w(v_i)\right| \leq \kappa$. That is, it is a bipartition where the sum of the weights of their nodes is within κ of being equal.

Lemma 11: The collecton of κ-weighted bisections of a graph with non-negative weighted vertices with maximum weight less than or equal to κ is zig-zaggable.

Proof: This proof follows essentially the same form as Lemma 9. The key idea is that the maximum weight of a vertex is κ, so any simple path between subsets of the vertices has the weight of the subsets increase by at most κ each step. ∎

Lemma 12: Let T be a Tanner graph with maximum node degree $d_{\max}$, let T' be a split Tanner graph obtained from T, and let T'' be a joined Tanner graph obtained by joining vertices of T'. Let the maximum number of vertices joined in a single vertex be $j_{\max}$. Let the minimum bisection width of T be ω. Then, the minimum $j_{\max}$-weighted bisection width of T'' is at least $\omega/(2d_{\max}) - j_{\max}d_{\max}$.

Proof: Suppose not, *i.e.,* that there is a $j_{\max}$-weighted bisection of width ω' such that $\omega' < \omega/(2d_{\max}) - j_{\max}d_{\max}$. Note that, by Lemma 3, T' has MBW of its T-corresponding vertices at least $\omega/(2d_{\max})$. We shall show how to construct a bisection of T' with width less than this. Firstly, consider the $j_{\max}$-weighted bisection of T'. Then, unjoin all the vertices, resulting in a bipartition of T'. Form a bisection of the

T-corresponding vertices of T' by moving T-corresponding nodes one by one from the side with the most vertices to the side with the least vertices until a bisection is formed. Each time a vertex is moved it increases the edges crossing the bisection by at most $d_{\max}$. A bisection is formed by moving no more than $j_{\max}$ nodes (since the original bipartition had difference in number of nodes at most $j_{\max}$). This constructs a bisection of T' with width less than $\omega/(2d_{\max})$, a contradiction. ∎

Theorem 2: If a sequence of Tanner graphs is generated uniformly according to the conditions of Theorem 1 and $d_{\max}(n) < \sqrt{n}$ for sufficiently large n, then a sequence of serialized- LDPC decoders based on these Tanner graphs have

$$\lim_{i\to\infty} P\left(A_i\tau_i \geq \frac{cn_i^{1.5}}{d_{\max}(n)}\right) = 1$$

for some $c > 0$.

Proof: From Definition 14, a serialized LDPC decoder must have a single node for each node of its joined Tanner graph. Consider a particular decoder of sufficiently large block length n. We consider two cases, that (a) $j_{\max} \geq \sqrt{n}$ and (b) $j_{\max} < \sqrt{n}$.

Case (a): The area of the circuit is at least n because there must be at least one node for each variable node. Consider the node that joined $j_{\max}$ nodes. Then $\tau \geq \sqrt{n}$ because at least $\sqrt{n}$ outputs must appear at that node. Thus $A\tau \geq n^{1.5}$.

Case (b). We consider the event that the Tanner graph of this code has MBW $\omega = cn$.

By Lemma 12, the $j_{\max}$-weighted bisection width of the joined Tanner graph is at least $c'n - \sqrt{n}d_{\max}$, where $c' = c/(2d_{\max})$

Let the $j_{\max}$-weighted bisection width of the circuit (and not the associated Tanner graph) be W. Now consider a minimum $j_{\max}$-weighted bisection of that circuit. Thus, there must be at least

$$\tau \geq \frac{c'n - \sqrt{n}d_{\max}}{2W}$$

clock cycles per iteration to communicate $c'n - \sqrt{n}d_{\max}$ bits across the bisection (where the factor of 2 comes from the bidirectionality assumption of the wires). By Lemma 10 we have

$$A \geq \frac{W^2}{4}$$

implying

$$A\tau^2 \geq \frac{\left(c'n - \sqrt{n}d_{\max}\right)^2}{16}.$$

As well, because there are at least n check nodes,

$$A \geq n$$

and so we get

$$A^2\tau^2 \geq \frac{n\left(c'n - \sqrt{n}d_{\max}\right)^2}{16}$$

which implies

$$A\tau \geq \frac{n^{1/2}\left(c'n - \sqrt{n}d_{\max}\right)}{16} \geq \Omega\left(\frac{n^{3/2}}{d_{\max}}\right)$$

where we have substituted $c' = c/2d_{\max}$ to obtain the last inequality. The theorem is then implied by Corollary 1 which shows that the MBW of the Tanner graph is proportional to n with probability approaching 1. ∎

C. Applicability and Limitations of Result

According to the definition of the uniform configuration distribution, it is possible that two or more edges can be drawn between the same two nodes. This type of conflict is usually dealt with by deleting even multi-edges and replacing odd multi-edges with a single edge [18, Definition 3.15]. This leads to a potential problem with the applicability of our theorem: what happens if the edges that we delete form a minimum bisection of the induced graph? In that case it is possible that the graph we instantiate on the circuit has a lower minimum bisection width than that which we calculated, and thus could possibly have less area. However, in the limit as n approaches infinity for a standard LDPC ensemble, the graph is locally tree-like [18, Th. 3.49] with probability approaching 1. This implies that the probability that the number of multi-edges in a randomly generated configuration is some fraction of n must approach 0 (or else the graph would not be locally tree-like, contradicting the theorem). Hence, even if we did delete these multi-edges from the randomly generated configuration, this could at most decrease the minimum bisection width by the number of deletions, but this number of deletions, with probability 1, cannot grow linearly with n. Hence, the minimum bisection width must still, with probability 1, grow linearly with n, and our scaling rules are still applicable.

In this paper we have considered Tanner graphs generated according to the uniform random configuration distribution, a commonly used method to analyze the performance of LDPC codes [18]. This does not mean that there do not exist good LDPC coding schemes with slower scaling laws. The scaling rule might be avoided if a different random generation rule for the Tanner graph is used. For example, perhaps the variable nodes and check nodes could be placed uniformly scattered through a grid and then the randomly placed edges, instead of being chosen uniformly over all possible edges, are chosen uniformly over a choice of edges connecting variable and check nodes that are "close" to each other. Whether or not such a sequence of LDPC codes would have good performance is unclear. However, in the following section we can obtain scaling rules that are true for *all* directly-implemented capacity-approaching LDPC decoders with vanishing error probability, not just almost all.

VI. Bounds for All Directly-Implemented Non-Split-Node LDPC Decoder Circuits

Definition 27: A sequence of codes and decoders in which the ith code has rate R_i for a channel with capacity C is *vanishing-error-probability capacity-approaching* if $\lim_{i\to\infty} R_i = C$ and block error probability approaches 0 as i is increased.

Definition 28: The *crossing number* of a graph is the minimum number of edges that cross in any planar embedding of that graph.

Note that since a crossing takes at least one grid square in any circuit, the crossing number obviously is a lower bound on circuit area.

Ganesan *et al.* [7] related the following lemma from Pach *et al.* [20] to understand the complexity of LDPC decoding. We will also use this result to understand a scaling rule for all, as opposed to almost all, directly implemented LDPC decoders.

Lemma 13 [20]: Let G be a graph with $|E| > 4|V|$ edges and girth greater than $2r$ for some integer $r > 0$. Then the crossing number of such a graph is bounded by

$$cr(G) \geq c_r \frac{|E|^{r+2}}{|V|^{r+1}} \tag{8}$$

for some constant c_r.

Theorem 3: The energy, per iteration, of any vanishing-error-probability capacity-approaching sequence of non-split-node directly-implemented LDPC decoders must have asymptotic energy per iteration lower bounded by

$$E \geq \chi^2 \ln^3(\chi).$$

Proof: Lemma 8 implies that the number of edges in a Tanner graph, per bit, scales as $\Omega(\ln \chi)$. From [21] and [22] note that the minimum block length of any code must scale as

$$n \geq c_3 \chi^2 \tag{9}$$

for a constant $c_3 > 0$. We then use Lemma 13, and the observation that a Tanner graph has girth at least 2, to conclude that a non-split-node directly-implemented decoder must have at least $\Omega(n \ln^3(\chi))$ wire crossings. ∎

Note that if a the LDPC codes are constrained to have girth greater than $2r$ then this argument can be extended to show that a sequence of such decoders must have area bounded by $\Omega(\chi^2 \ln^{r+2} \chi)$.

It may be that directly-implemented LDPC decoders can improve upon this lower bound by splitting up check and variable node subcircuits (and not localizing these computations in one area of the circuit). In actual VLSI design this may happen automatically by circuit design software, so this limits the applicability of this theorem.

The lower bound of Theorem 3 is applicable to all non-split-node directly-implemented LDPC decoding schemes. However, using a punctured code construction, Pfister *et al.* [4] construct a capacity-approaching ensemble of codes that avoids the complexity blowup of Lemma 8. Theorem 3 does not apply to such constructions. We considered the check-regular ensemble of [4, Th. 2] and computed whether this ensemble satisfies the conditions of Theorem 1. By varying the parameters ε from 0.05 to 0.3 in increments of 0.05 and the parameter p from 0.05 to 0.95 in increments of 0.05, computations show that the only values of these parameters that did not satisfy the conditions were $p = 0.05$ when $\varepsilon = 0.15, 0.2, 0.25, 0.3$. Thus, for most parameters checked we conclude that decoders based on these ensembles satisfy the almost-sure scaling rules of Corollaries 2 and Theorem 2.

Comparison to Universal Lower Bounds

We note that this lower bound on directly-implemented Tanner graphs contrasts with the lower bounds in [3], which show an $\Omega(\chi^2 \ln^{1/2}(\chi))$ lower bound for the energy complexity of fully-parallel decoding algorithms as a function of gap to capacity. This result means that non-split-node directly-implemented LDPC decoders are *necessarily* asymptotically worse than this lower bound. Of course, it is not known whether the lower bounds of the paper in [3] are tight. It may also be that splitting check nodes in the circuit could overcome this lower bound and get closer to the universal lower bound, but our result does not address this case.

VII. Open Problems

There are still some unanswered questions related to the computational complexity of LDPC decoding, and error control coding in general. We discuss some of these problems below.

- This paper finds an "almost sure" scaling rule for the energy of VLSI LDPC decoders. However, it does not exclude the possibility that there are good LDPC codes whose decoding energy scales more slowly than this (they may simply occur with vanishing probability). Thus, there may be some good LDPC codes with "lower energy" Tanner graphs that still provide good code performance. Intuition suggests that for a given channel a code with a "lower energy" LDPC decoder may have higher probability of error. A general analysis of this fundamental tradeoff is an open question.
- The dependence on maximum node degree of our scaling rules is somewhat surprising. In our definitions of LDPC decoders, we consider a graph that contains the Tanner graph as a minor. It may be that high degree nodes can be split to decrease the minimum bisection width of a graph and thus possibly decrease circuit area. A formal analysis of how vertex splitting might decrease circuit area remains an open question.
- It may be that edges of a Tanner graph that connect vertices are too far away on the decoder can be modified to connect closer nodes, with a small cost in error probability. As well, there exist some algorithmic level modifications [23] that may allow energy savings. A theoretical analysis of such techniques may be informative.
- For a given application, what technique can be used to choose the "energy-optimal" error control code? Can this analysis improve the energy of real communication systems? Ganesan *et al.* [7] discuss this question and show how, for a reasonable system model, the performance optimal code depends on the circuit technology used and the nature of the channel.
- A theoretical energy analysis for almost any other type of error control code also remains generally unexplored (polar codes [24] and spatially coupled codes [25] may be particularly suitable for this type of analysis). Some early theoretical work on the VLSI energy complexity for polar coding exists in [26].

- The $\Omega(n^2/d_{\max}^2)$ lower bound on directly-implemented LDPC decoders can be reached for node degree distributions of bounded degree (see for example our simple construction in [3]). However, it is not known whether the $\Omega(n^{3/2}/d_{\max})$ lower bound for serialized decoders can be reached.

VIII. Conclusion

The main contribution of this paper is graph theoretic. We have shown that subject to a mild condition on node degrees, almost all Tanner graphs have a minimum bisection width that scales as $\Omega(n)$ where n is the number of left nodes. We have used this to show that almost all directly-implemented LDPC decoders must have circuit area, and thus energy, that scales as $\Omega(n^2)$. As well, we show that almost all serialized decoders have energy complexity, per iteration, that scales as $\Omega(n^{3/2}/d_{\max})$. We have further presented a general theorem on the area of circuits that instantiate any graph to bound the area of any sufficiently large LDPC decoder generated from a capacity-approaching distribution. Note that our results show that directly-implemented LDPC decoders cannot reach the lower bounds presented in [3], thus indicating that either the lower bound cited is not tight, or non-split-node directly-implemented LDPC codes are asymptotically not optimal from this energy perspective. It may also be that both are true, namely that known lower bounds are not tight and LDPC codes are not asymptotically optimal. This remains an open question.

Appendix A
Definition of $\delta(L, R)$ in Terms of Node Degree Distributions

In the discussions in this paper, we define a quantity δ in (2) in terms of node degree lists. Given a bipartite graph G and number of left nodes n, number of right nodes m, and node degree lists R and L, one can easily construct the more standard node degree distribution. This definition is adapted from [18].

Definition 29: For a bipartite graph G, let ρ_i be the fraction of right nodes in G of degree i and let λ_i be the fraction of left nodes of degree i. Then $P = \{\rho_1, \ldots\}$ is the *right node degree distribution* and $\Lambda = \{\lambda_1, \ldots\}$ is the *left node degree distribution*.

We note that the sum of the entries of both P and Λ defined above must be 1, and that Λ and P are functions of the left and right node degree lists. Since it is more common to consider distributions in terms of their node degree distributions, we will state the quantity $\delta(L, R)$ of (2) used in Theorem 1 in terms of Λ and P, the left and right node degree distributions.

Consider a sequence $X = \{x_1, x_2, \ldots\}$, such that $\sum x_i = 1$ and $0 \le x_i \le 1$ for each i. Define:

$$M(X) = \max\left\{m : \sum_{i=m}^{\infty} x_i \ge \frac{1}{2}\right\}$$

Note that for a graph with right node degree distribution P, there are at least half the right nodes with degree $M(P)$ or greater.

We define

$$\xi(X) = \frac{1}{2} - \sum_{i=M(X)+1}^{\infty} X_i$$

so that $\xi(X) + \sum_{i=M(X)+1}^{\infty} X_i = \frac{1}{2}$. We let

$$S'_{\text{top}}(X) = M(X)\xi(X) + \sum_{i=M(X)+1}^{\infty} i x_i.$$

Then it is obvious to see that the quantity δ from (2) can be computed in terms of the graph's node degree distribution as:

$$\delta(L, R) = \max\left(S'_{\text{top}}(\lambda), S'_{\text{top}}(P)\right).$$

Appendix B
Proof of Lemma 6

Proof: Since $m < Z$, $n < Z$, the product $m!n! < Z!Z!$, so if $Y - Z > Z$ then obviously $m!n! < Z!(Y - Z)!$ Thus we consider the case that $Y - Z \le Z$. Since $m!n!$ is increasing in m and n, we shall also assume that $m + n = Y$. We now argue that $Z!(Y - Z)!$ maximizes $m!n!$ and is achieved when $m = Y - Z$ and $n = Y$.

Suppose that $c \ge d \ge 1$ for positive integers c and d. We have

$$\frac{c+1}{d} > 1 \tag{10}$$

implying

$$\frac{(c+1)!\,(d-1)!}{c!d!} > 1.$$

This implies

$$c!d! < (c+1)!(d-1)!.$$

Thus, any product $m!n!$ in which $n \le m < Z$ and $m + n = Y$ can be increased by increasing m by 1 and decreasing n by 1 (which still preserves $m + n = Y$). ∎

Appendix C
Proof of Lemma 6 Continued

For the sake of simplicity, we will further loosen these bounds by upper bounding each of the factors a, b, c, and d. Each of these bounds is easily verified:

a) We note that $\binom{n}{i} \le \binom{n}{\frac{n}{2}}$.
b) Since $m \le n$, thus $\binom{m}{\frac{m+2}{2}-i} \le \binom{n}{\frac{n}{2}}$.
c) $\binom{|E|}{j}\binom{|E|}{a-j}\binom{|E|}{j}\binom{|E|}{a-j} \le \binom{|E|}{a}^4$ which is implied by $a \le \sigma n \le \frac{|E|}{2}$.
d) $(j)!\,(a-j)! \le a!$ which flows directly from the observation that $\binom{a}{j} \ge 1$.

Combining these gives us the following bound:

$$\left|Q_a^{i,j}\right| \le \binom{n}{\frac{n}{2}}^2 \binom{|E|}{a}^4 a!\,(\delta n)!\,(\sigma n - a)!.$$

We can bound $|Q_a|$ by summing over our upper bound on $\left|Q_a^{i,j}\right|$:

$$|Q_a| \leq \sum_{i=1}^{n}\sum_{j=1}^{n}\left|Q_a^{i,j}\right| \leq n^2\binom{n}{\frac{n}{2}}^2\binom{|E|}{a}^4 a!\,(\delta n)!\,(\sigma n - a)!. \quad (11)$$

We of course are not concerned with the probability of a bisection of size a, but rather with the probability of a bisection of size a or less. We denote the set of configurations with a bisection of size a or less by Q_a^* and since $Q_a^* = \bigcup_{i=0}^{a} Q_a$:

$$\left|Q_a^*\right| \leq \sum_{i=0}^{a}|Q_i|.$$

We will now show that the expression in (11) is an non-decreasing function of a for $0 < a \leq \frac{|E|-1}{2}$. Let the right side of the expression be denoted d_a, then it is easy to show that $\frac{d_{a+1}}{d_a}$ is greater than or equal to 1. It is easy to show that

$$\frac{d_{a+1}}{d_a} = \frac{\binom{|E|}{a+1}^4(a+1)}{\binom{|E|}{a}^4(\sigma n - a)}.$$

Expanding the binomial coefficients in the numerator and denominator and simplifying gives us

$$\frac{d_{a+1}}{d_a} = \frac{(|E|-a)^4}{(a+1)^3(\sigma n - a)}.$$

This quantity will be greater than or equal 1 if $|E|-a \geq a+1$ and $|E| - a \geq \sigma n - a$. Note that $a < \sigma n$ (an assumption of our lemma) implies $2a < 2\sigma n \leq |E|$. Since a and $|E|$ are both integers, this implies $2a \leq |E| - 1$, from which we can see that the first inequality is satisfied. The second is satisfied by the fact that $\sigma n \leq |E|$. We thus observe that,

$$\begin{aligned}\left|B_a^*\right| &\leq \left|Q_a^*\right| \\ &\leq \sum_{i=0}^{a}|Q_i| \\ &\leq \sum_{i=0}^{a} n^2\binom{n}{\frac{n}{2}}^2\binom{|E|}{a}^4 a!\,(\delta n)!\,(\sigma n - a)! \\ &\leq (a+1)\,n^2\binom{n}{\frac{n}{2}}^2\binom{|E|}{a}^4 a!\,(\delta n)!\,(\sigma n - a)!. \quad (12)\end{aligned}$$

We note that the number of possible multi-graphs with our given node degree distribution is at least $(\delta n + \sigma n)!$. We can now bound the probability of the event B_a^* with:

$$P\left(B_a^*\right) \leq \frac{\left|B_a^*\right|}{(\delta n + \sigma n)!} \quad (13)$$

$$\leq \frac{(a+1)\,n^2\binom{n}{\frac{n}{2}}^2\binom{|E|}{a}^4 a!\,(\delta n)!\,(\sigma n - a)!}{(\delta n + \sigma n)!} \quad (14)$$

where we have simply applied the upper bound for the size of B_a^* of (12).

Appendix D
Proof of Theorem 1

We first prove a simple lemma:

Lemma 14: Suppose $g(n) = O\left(n^k\right)$ for some $k > 0$ and is positive for sufficiently large n, and there is a sequence $n_1, n_2, \ldots$ that increases without bound. Then:

$$\lim_{i\to\infty} g(n_i)\exp(n_i f(n_i)) = 0 \text{ if } \limsup_{n\to\infty} f(n) < 0.$$

Proof: Since $\limsup_{n\to\infty} f(n) < 0$ and the sequence n_i increases without bound, then for sufficiently large i, $f(n_i) < -c$ for some $c > 0$. Then, for sufficiently large i,

$$g(n_i)\exp(n_i f(n_i)) \leq g(n_i)\exp(-cn_i).$$

Clearly, $\lim_{i\to\infty} g(n_i)\exp(-cn_i) = 0$ and because $g(n)$ is positive for large enough n,

$$g(n_i)\exp(-n_i f(n_i)) > 0$$

for large enough i. The limit thus follows from the squeeze theorem. ∎

Consider first a specific random configuration in the sequence with block length n and node degree distributions that result in values for δ and σ. We will use the bounds of Lemma 7 and then apply well known approximations. Firstly, we use the well-known bounds derived from Stirling's approximation [27, Question 5.8] that

$$e^{\left(1+n\ln\left(\frac{n}{e}\right)\right)} \leq n! \leq e^{\left(1+(n+1)\ln\left(\frac{n+1}{e}\right)\right)}$$

and that

$$\binom{n}{k} \leq \exp\left(n\mathcal{H}\left(\frac{k}{n}\right)\right)$$

where $\mathcal{H}(x) = -x\log x - (1-x)\log(1-x)$. We use base e as opposed to base 2 in order to conveniently simplify the expressions that follow. Applying these bounds appropriately to the bound in Lemma 7, and grouping terms that grow polynomially into an arbitrary polynomial term $g(n)$ we get that:

$$\begin{aligned}P\left(B_a^*\right) &\leq g(n)(a+1)\exp\Bigg[2n\mathcal{H}\left(\frac{1}{2}\right) + 4n\mathcal{H}\left(\frac{a}{|E|}\right) \\ &\quad + a\ln\left(\frac{a+1}{e}\right) + \delta n\ln\left(\frac{\delta n+1}{e}\right) \\ &\quad + (\sigma n - a)\ln\left(\frac{\sigma n - a + 1}{e}\right) \\ &\quad - (\delta n + \sigma n)\ln\left(\frac{\delta n + \sigma n}{e}\right)\Bigg].\end{aligned}$$

We now let $a = \beta n$, which will satisfy the condition specified in (3) for $\beta < \sigma$. We substitute $\mathcal{H}(1/2) = \ln 2$ and $|E| = \delta n + \sigma n$, and combine polynomial terms into $g(n)$, and

then use algebraic manipulation to give us:

$$P\left(B^*_{\beta n}\right) \leq g\,(n) \exp\Bigg[2n\ln(2) + 4n\mathcal{H}\left(\frac{\beta}{\delta+\sigma}\right) + \beta n \ln\left(\frac{\beta+\frac{1}{n}}{\sigma-\beta+\frac{1}{n}}\right) + \sigma n \ln\left(\frac{\sigma-\beta+\frac{1}{n}}{\delta+\sigma}\right) + \delta n \ln\left(\frac{\delta+\frac{1}{n}}{\delta+\sigma}\right)\Bigg].$$

By factoring the n term and by applying Lemma 14, we see that the above expression will approach 0 if

$$\limsup_{i\to\infty} 2\ln(2) + 4\mathcal{H}\left(\frac{\beta}{\delta+\sigma}\right) + \beta\ln\left(\frac{\beta+\frac{1}{n}}{\sigma-\beta+\frac{1}{n}}\right) + \sigma\ln\left(\frac{\sigma-\beta+\frac{1}{n}}{\delta+\sigma}\right) + \delta\ln\left(\frac{\delta+\frac{1}{n}}{\delta+\sigma}\right) \leq 0$$

where we recall again that the dependence on i in this expression comes from the n terms and the δ and σ terms (whose dependence on i we have suppressed). This is true if

$$\limsup_{i\to\infty} 2\ln(2) + 4\mathcal{H}\left(\frac{\beta}{\delta+\sigma}\right) + \beta\ln\left(\frac{\beta}{\sigma-\beta}\right) + \sigma\ln\left(\frac{\sigma-\beta}{\delta+\sigma}\right) + \delta\ln\left(\frac{\delta}{\delta+\sigma}\right) \leq 0.$$

Also note that this is the condition on β given in (6). To derive the condition in (5), we find the limit as β approaches 0 of this expression, and treating the other terms as constants, giving us:

$$2\ln(2) + \sigma\left(\ln\left(\frac{\sigma}{\delta+\sigma}\right)\right) + \delta\left(\ln\left(\frac{\delta}{\delta+\sigma}\right)\right) \leq 0$$

where we have applied the easily verifiable facts that $\lim_{x\to 0}\mathcal{H}\left(\frac{x}{c}\right) = 0$ and $\lim_{x\to 0} x\left(\ln\left(\frac{x}{\sigma-x}\right)\right) = 0$ to get rid of the second and third terms in the expression. Thus, if this condition is satisfied, by the definition of a limit, there exists a sufficiently small β in which $\lim_{i\to\infty} P\left(B^*_{\beta n}\right) = 0$.

Appendix E
Proof of Lemma 10

In this proof, adapted with only slight differences from Thompson's proof [1, Th. 2], we show that if a circuit's graph has a $\mathcal{C}$-bipartition width ω, then at least $\omega^2/4$ grid squares of the circuit are occupied. To do so, we adapt the zig-zag argument of Thompson and construct on the order of ω curves that $\mathcal{C}$-bipartition the circuit, each which must have ω connections crossing the curve, implying that there are close to ω uncounted nodes adjacent to the curve. The details require defining sequences of curves which increase the number of nodes on their left side by 1 each step, which we call the "initial sweep" sequences and the "zig-zag raise" sequences.

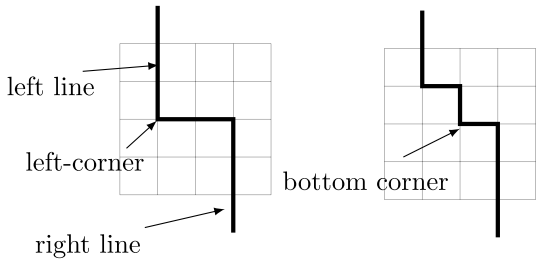


Fig. 2. An example of a zig-zag (on the left) with its left corner, left line, and right line labelled, and the curve resulting when its left corner is indented (on the right). Note that indenting a curve at most adds one more node to the left side.

Let the grid of a circuit form a Cartesian coordinate system so that all nodes occupied are in the top left quadrant. Draw the smallest rectangle aligned with the circuit grid that encloses the circuit. All points outside this rectangle are considered outside the circuit. The top of this rectangle is the top of the circuit, and the bottom is the bottom of the circuit.

Definition 30: A *zig-zag* of width a is a curve drawn on a circuit composed of a vertical line starting outside the circuit leading to coordinate (x, y), a horizontal line connecting (x, y) to $(x+a, y)$, and then a vertical line from this point to below the circuit. The point (x, y) is called the *left corner* of the zig-zag. The vertical line on the left is called the *left line* and on the right the *right line*. An example of a zig-zag with its left corner, left line, and right line labelled is given in Figure 2.

Definition 31: A curve with a left corner at coordinate (x, y) can be *indented* at this coordinate by replacing the curve with a new curve where the edges connecting coordinates $(x, y+1)$ to (x, y) and then to $(x+1, y)$ are replaced with two edges connecting $(x, y+1)$ to $(x+1, y+1)$ and $(x+1, y+1)$ to $(x+1, y)$. The point $(x+1, y)$ is the *bottom corner* of the indentation.

The reader should refer to Figure 2 to see an example of a curve that is indented, and a labelling of the bottom corner of the resulting indented curve.

Definition 32: An *initial sweep* of a circuit is a sequence of curves beginning with a width 1 zig-zag with left corner at location $(0, 0)$. The left corner of the zig-zag is successively indented until a zig-zag with left corner at the top of the circuit is obtained. Then, this process is repeated starting with a width 1 zig-zag with left corner at $(1, 0)$. This process is continued until a zig-zag in which all occupied grid squares are to its left is obtained. Figure 3 show an example of the sequence of curves in an initial sweep for a small circuit.

The idea of an initial sweep is that the first curve has no circuit nodes to its left, and eventually the curve has all nodes to its left; in between the amount of nodes to the left of the curve increases by at most 1 each time. This means that there will be a $\mathcal{C}$-bipartition of the circuit induced by one of the curves in the initial sweep, a consequence of property 1 of a zig-zaggable set of bipartitions.

Definition 33: A curve that $\mathcal{C}$-bipartitions the circuit in an initial sweep is the *initial curve*.

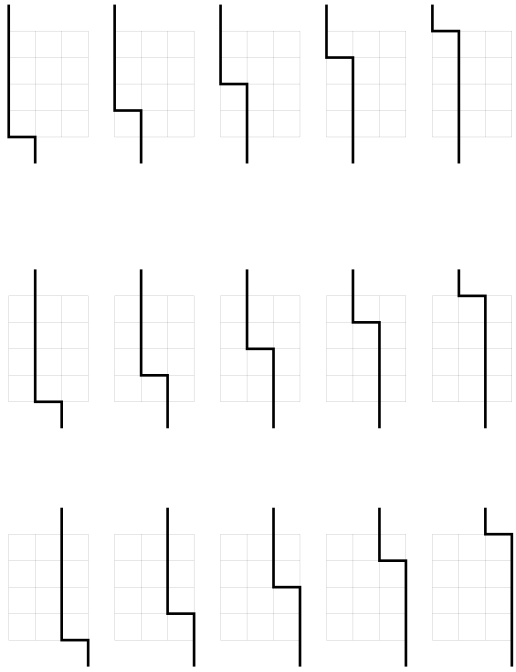


Fig. 3. An example of the curves, in order from left to right, top to bottom, of an initial sweep of a circuit of height 4 and width 3. On each row each successive curve is created by indenting the previous curve's left corner. Each curve of an initial sweep has at most one more circuit node on its left side. We see that an initial sweep must eventually bisect the nodes of the circuit. For the same reason, at least one curve of the initial sweep must $\mathcal{C}$-bipartition the nodes for any zig-zaggable set of bipartitions $\mathcal{C}$, a consequence of property 1 of zig-zaggable sets of bipartitions.

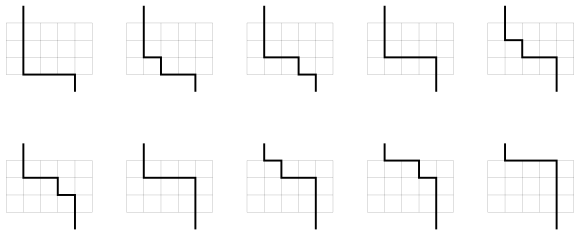


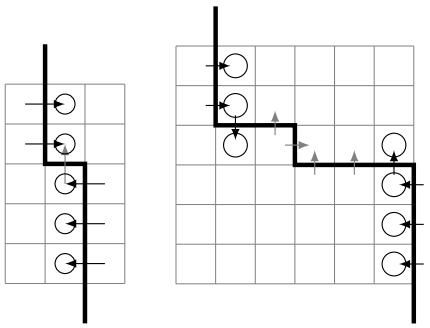
Fig. 4. An example of the curves, in order from left to right, top to bottom, of a zig-zag raise of width 3 for a circuit of height 3.

We let the left line of the initial curve have x-coordinate ℓ.

Note that the left-corner (at location (x, y) of a width a zig zag can be indented, resulting in a new curve. The resulting bottom corner can be indented again in total a times, and the end result is a new zig-zag of width a (where a is positive integer), this time with left-corner at $(x, y + 1)$ (one unit higher than the initial zig-zag). The indenting process can be performed on this new zig-zag. This can be done repeatedly until a zig-zag with left-corner at the top of the circuit is obtained.

Definition 34: We call a curve resulting from such a sequence of indentations an *indented zig-zag*.

Definition 35: The sequence of curves generated by this sequence of indentations is called a *zig-zag raise*. The curves corresponding to a zig-zag raise for a small circuit are given in Figure 4.



(a) (b)

Fig. 5. (a) This figure is adapted from [1, Figs. 3 and 4]. An example of an initial curve with arrows crossing grid edges that could form possible connections between the left side and the right side. For all but the gray arrow, we can conclude that if a connection exists across the edge the arrow crosses, there is a unique grid square (which in the diagram contains a circle) occupied in the boundary column. Thus, if ω edges must cross the curve, at least $\omega - 1$ nodes in the boundary column must be occupied. (b) An example of an indented zig-zag obtained from a zig-zag raise. The grid squares with circles in them are the grid squares in the boundary columns that are adjacent to an edge of the curve. Arrows cross edges where a connection between the left side and the right side can be made. For each black arrow, if a connection is there in the circuit, then the circle to which the arrow points must contain an occupied node. Note that there are at most 4 crossings that do not involve a node with an boundary column (which are denoted by gray arrows). Thus, if ω crossings must exist across the indented zig-zag, there must be at least $\omega - 4$ circles occupied.

Definition 36: Given an indented zig-zag, the column to the right of the left line and to the left of the right line are called the *boundary columns*.

Definition 37: The computational nodes to the left of the indented zig-zag are called the *left nodes* and the nodes to its right are called the *right nodes*.

Now, consider performing a zig-zag raise starting from a zig-zag of width 3 with left-corner at $(0, \ell - 1)$. Let S_1 be the left-nodes of the first curve of the zig-zag raise and S_3 be the left-nodes of the last curve of the zig-zag raise. We let S_2 be the left-nodes of the initial curve. Since obviously $S_1 \subseteq S_2 \subseteq S_3$ and $S_2 \in \mathcal{C}$, applying Property 2 of zig-zaggable bipartitions (See Definition 23) means that there must be some curve of the zig-zag raise that is a $\mathcal{C}$-bipartition.

By the same argument, for any j, we can construct a width $2j + 1$ indented zig-zag that $\mathcal{C}$-bipartitions the circuit, by starting with a zig-zag with left line at $x = \ell - j$ and performing a zig-zag raise. A possible indented zig-zag that results from this process for $j = 2$ is given in Figure 5(b).

Definition 38: Two grid squares of a circuit are *connected* across a grid edge if they are adjacent at the grid edge and either they contain wires that are connected or one square contains a node attached to a wire in the other square. Such a pair of grid squares is called a *connection*.

Since each of these curves $\mathcal{C}$-bipartitions the circuit, there must be at least ω connections across each curve.

Figure 5(a) shows that the initial curve must have at least $\omega - 1$ grid squares occupied in its boundary column.

As well, Figure 5(b) shows a width 5 indented zig-zag and shows that if ω edges must cross the indented zig-zag, then there must be at least $\omega - 4$ grid squares in the boundary column occupied. This is because for all but 4 of the possible

connections, if they are connected then this implies a unique grid square in the boundary column is occupied. It is then easy to generalize that an indented zig-zag of width $2k+1$ must have at least $\omega - 2k$ occupied nodes in its boundary columns.

Since the boundary columns of each of the bipartitions constructed do not intersect, summing up the lower bound on the number of grid squares occupied implies that the number of grid squares occupied in the circuit is bounded as:

$$\begin{aligned}
&\omega - 1 + \sum_{i=1}^{\lfloor \frac{\omega}{2} \rfloor} \omega - 2i \\
&\quad \geq \omega - 1 + \omega \left\lfloor \frac{\omega}{2} \right\rfloor - \left\lfloor \frac{\omega}{2} \right\rfloor \left(\left\lfloor \frac{\omega}{2} \right\rfloor + 1 \right) \\
&\quad \geq \omega - 1 + \left\lfloor \frac{\omega}{2} \right\rfloor \left(\omega - \left\lfloor \frac{\omega}{2} \right\rfloor - 1 \right) \\
&\quad \geq \frac{\omega^2}{4} \qquad (15)
\end{aligned}$$

The last inequality flows from the fact that either $\frac{\omega}{2}$ is an integer and $\lfloor \frac{\omega}{2} \rfloor = \frac{\omega}{2}$ or $\lfloor \frac{\omega}{2} \rfloor = \frac{\omega}{2} - \frac{1}{2}$. We see in both cases that the expression in (15) is greater than $\omega^2/4$ for $\omega > 2$. As Thompson observed in his proof of the theorem, the case when $\omega = 1$ is trivial.

ACKNOWLEDGMENTS

The authors would like to thank the Associate Editor, H. Pfister, and the anonymous reviewers for their detailed and thoughtful comments on an earlier version of this paper.

REFERENCES

[1] C. D. Thompson, "A complexity theory for VLSI," Ph.D. dissertation, Dept. Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, 1980.
[2] R. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.
[3] C. G. Blake and F. R. Kschischang, "Energy consumption of VLSI decoders," *IEEE Trans. Inf. Theory*, vol. 61, no. 6, pp. 3185–3198, Jun. 2015.
[4] H. D. Pfister, I. Sason, and R. Urbanke, "Capacity-achieving ensembles for the binary erasure channel with bounded complexity," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2352–2379, Jul. 2005.
[5] C.-H. Hsu and A. Anastasopoulos, "Capacity-achieving codes with bounded graphical complexity and maximum likelihood decoding," *IEEE Trans. Inf. Theory*, vol. 56, no. 3, pp. 992–1006, Mar. 2010.
[6] K. Ganesan, P. Grover, and A. Goldsmith, "How far are LDPC codes from fundamental limits on total power consumption?" in *Proc. Allerton Conf. Commun., Control, Comput.*, Monticello, IL, USA, Oct. 2012, pp. 671–678.
[7] K. Ganesan, P. Grover, J. Rabaey, and A. Goldsmith, "On the total power capacity of regular-LDPC codes with iterative message-passing decoders," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 375–396, Feb. 2016.
[8] D. B. West, *Introduction to Graph Theory*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2001.
[9] M. Fiedler, "A property of the eigenvectors of non-negative symmetric matrices and its application to graph theory," *Czechoslovak Math. J.*, vol. 25, no. 4, pp. 619–633, 1975.
[10] S. L. Bezrukov, R. Elsässer, B. Monien, R. Preis, and J. P. Tillich, "New spectral lower bounds on the bisection width of graphs," *Theor. Comput. Sci.*, vol. 320, nos. 2–3, pp. 155–174, Jun. 2004.
[11] J. Diaz, M. J. Serna, and N. C. Wormald, "Bounds on the bisection width for random d-regular graphs," *Theor. Comput. Sci.*, vol. 382, nos. 2–3, pp. 120–130, Aug. 2007.
[12] M. J. Luczak and C. McDiarmid, "Bisecting sparse random graphs," *Random Struct. Algorithms*, vol. 18, no. 1, pp. 31–38, Jan. 2001.
[13] M. Garey, D. Johnson, and L. Stockmeyer, "Some simplified NP-complete graph problems," *Theor. Comput. Sci.*, vol. 1, no. 3, pp. 237–267, 1976.
[14] C. D. Thompson, "Area-time complexity for VLSI," in *Proc. ACM Symp. Theory Comput.*, Atlanta, GA, USA, Apr. 1979, pp. 81–88.
[15] P. Grover, A. Goldsmith, and A. Sahai, "Fundamental limits on the power consumption of encoding and decoding," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2012, pp. 2716–2720.
[16] J. Thorpe, "Design of LDPC graphs for hardware implementation," in *Proc. Int. Symp. Inf. Theory*, Sep. 2002, p. 483.
[17] K. Ganesan, P. Grover, and J. Rabaey, "The power cost of over-designing codes," in *Proc. IEEE Workshop Signal Process. Syst.*, Oct. 2011, pp. 128–133.
[18] T. Richardson and R. Urbanke, *Modern Coding Theory*. New York, NY, USA: Cambridge Univ. Press, 2008.
[19] I. Sason and R. Urbanke, "Parity-check density versus performance of binary linear block codes over memoryless symmetric channels," *IEEE Trans. Inf. Theory*, vol. 49, no. 7, pp. 1611–1635, Jul. 2003.
[20] J. Pach, J. Spencer, and G. Tóth, "New bounds on crossing numbers," *Discrete Comput. Geometry*, vol. 24, no. 4, pp. 623–644, 2000.
[21] R. G. Gallager, *Information Theory Reliable Communication*. New York, NY, USA: Wiley, 1968.
[22] V. Strassen, "Asymptotic estimates in Shannon's information theory," in *Proc. 3rd Trans. Prague Conf. Inf. Theory*, Prague, Czech Republic, 1962, pp. 689–723.
[23] T. Mohsenin, D. N. Truong, and B. M. Baas, "A low-complexity message-passing algorithm for reduced routing congestion in LDPC decoders," *IEEE Trans. Circuits Syst.*, vol. 57, no. 5, pp. 1048–1061, May 2010.
[24] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.
[25] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 803–834, Feb. 2011.
[26] C. G. Blake and F. R. Kschischang, "On scaling rules for the energy of polar encoders and decoders," *CoRR*, Feb. 2016. [online]. Available: https://arxiv.org/abs/1602.04034
[27] R. G. Gallager, *Information Theory and Reliable Communication*. New York, NY, USA: Wiley, 1968.

Christopher G. Blake received his B.A.Sc. degree from the University of Toronto in 2009 and his M.A.Sc. degree from the Massachusetts Institute of Technology in 2011, both in electrical engineering. He received his PhD at the University of Toronto in the Department of Electrical and Computer Engineering in 2016. He is interested in coding theory and computational complexity theory.

Frank R. Kschischang (S'83–M'91–SM'00–F'06) received the B.A.Sc. degree (with honors) from the University of British Columbia, Vancouver, BC, Canada, in 1985 and the M.A.Sc. and Ph.D. degrees from the University of Toronto, Toronto, ON, Canada, in 1988 and 1991, respectively, all in electrical engineering. He is the Distinguished Professor of Digital Communication in the Department of Electrical and Computer Engineering at the University of Toronto, where he has been a faculty member since 1991. During 1997-98, he was a visiting scientist at MIT, Cambridge, MA; in 2005 he was a visiting professor at the ETH, Zurich, and in 2011 and again in 2012-13 he was a visiting Hans Fischer Senior Fellow at the Institute for Advanced Study at the Technical University of Munich.

His research interests are focused primarily on the area of channel coding techniques, applied to wireline, wireless and optical communication systems and networks. In 1999 he was a recipient of the Ontario Premier's Excellence Research Award and in 2001 (renewed in 2008) he was awarded the Tier I Canada Research Chair in Communication Algorithms at the University of Toronto. In 2010 he was awarded the Killam Research Fellowship by the Canada Council for the Arts. Jointly with Ralf Kötter he received the 2010 Communications Society and Information Theory Society Joint Paper Award. He is a recipient of the 2012 Canadian Award in Telecommunications Research. He is a Fellow of IEEE, of the Engineering Institute of Canada, and of the Royal Society of Canada.

During 1997-2000, he served as an Associate Editor for Coding Theory for the IEEE TRANSACTIONS ON INFORMATION THEORY, and from 2014 to 2016 he served as this journal's Editor-in-Chief. He also served as technical program co-chair for the 2004 IEEE International Symposium on Information Theory (ISIT), Chicago, and as general co-chair for ISIT 2008, Toronto. He served as the 2010 President of the IEEE Information Theory Society. He received the Aaron D. Wyner Distinguished Service Award in 2016.