Finite-Alphabet Message Passing using only Integer Operations for Highly Parallel LDPC Decoders

Tobias Monsees, Dirk Wübben, Armin Dekorsy
Department of Communications Engineering
University of Bremen, 28359 Bremen, Germany
Email: {tmonsees, wuebben, dekorsy}@ant.uni-bremen.de

Oliver Griebel, Matthias Herrmann, Norbert Wehn
University of Kaiserslautern, 67663 Kaiserslautern, Germany
{griebel, herrmann, wehn}@eit.uni-kl.de

***Abstract*—In this paper, we present a new design of Finite Alphabet (FA) Message Passing (MP) decoders using only integer operations. We utilize Discrete Density Evolution with a multidimensional Lookup-Table (mLUT) design for Variable Node (VN) updates to consider all input messages jointly for reducing the information loss compared to the frequent sequential LUT design approaches. From this mLUT design, we derive a minimum-integer computation (MIC) decoder that allows for different bit-widths for node operations and message exchanges between nodes. The mLUT operations for VN updates are replaced by low complexity signed integer additions and threshold operations, and the Check Node (CN) updates simplify to a minimum search over integers. For a (816,406) regular LDPC code, we show that our 3-bit MIC decoder achieves the communication performance of the corresponding mLUT decoder and outperforms a 4-bit state-of-the-art Min-Sum (MS) decoder. We show that the node implementations on a 22 nm FD-SOI technology yield an improved area and energy efficiency over the respective MS implementation. To the best of our knowledge, this is the first time that an implementation improvement for the VNs and CNs is shown when using FA MP.**

***Index Terms*—LDPC code, finite alphabet message passing, mutual-information based design, implementation efficiency**

I. Introduction

Beyond 5G and 6G wireless communication systems target peak data rates of 100 Gb/s to 1 Tb/s with processing latencies between 10-100 ns [1]. For such high data rate and low latency requirements, the implementation of the Forward Error Correction (FEC) decoder, that is the most complex and computationally intense component in the baseband processing chain, is a major challenge [2].

Low-Density Parity Check (LDPC) codes [3] are FEC codes with capacity approaching error correction performance [4] and are part of many communication standards, e.g. DVB-S2x, Wi-Fi, and 3GPP 5G-NR. In contrast to other advanced FEC codes, like Polar and Turbo codes, the decoding of LDPC codes is dominated by data transfers [2] making high throughput decoders very challenging in which area and power is largely dominated by the routing. Two sets of nodes, the variable and check nodes, iteratively exchange messages over the edges of a bipartite graph (Tanner graph of the code). Very high throughput decoders can be achieved by mapping the Tanner graph one-to-one on hardware, i.e., instantiating a dedicated processing unit for each node and hardwiring the edges of the Tanner graph. Unrolling and pipelining the decoding iteration can further increase the throughput [5]. Such Full-Parallel (FP) decoders can achieve in principal throughputs towards 1 Tb/s. However, due to the large routing challenge, such decoders are limited to smaller block sizes in the order of about 1 kbit. This problem is even exacerbated in very advanced technology nodes since routing does scale much worse than transistor density.

Finite Alphabet (FA) Message Passing (MP) decoding has been introduced as a method to mitigate the routing challenges in unrolled LDPC decoders by reducing the bit-widths of the exchanged messages and, thus, the number of necessary wires [6]. In contrast to the conventional MP decoding algorithms like the Belief Propagation (BP) and its approximations, e.g. Min-Sum (MS), Offset Min-Sum (OMS) and Normalized Min-Sum (NMS) [7], the node operations in FA decoding are designed to fulfil a pre-specified design criterion. The Lookup-Table (LUT)-decoder [8], [9] is designed to maximize mutual information between a code bit and its message under the assumption that all operations are decomposed into a sequence of two inputs. In contrast to Mutual Information-Maximizing Quantized Belief Propagation (MIM-QBP) [10] and Reconstruction-Computation-Quantization (RCQ) [11], we derive a representation of the multidimensional Variable Node (VN) LUT that aims to minimize the local node complexity by using only low-range integers and avoids complexity drawbacks of a LUT implementation [12]. The challenge with these decoders is the trade-off between reduced routing complexity by smaller number of quantization and increased node complexity due to the LUTs, i.e. the area and power of the VN node of state-of-the-art (SoA) FA MP based decoders is typically larger than its NMS equivalent. We will show that our new approach will solve this problem and decrease the complexity of the node implementation. The new contributions of this paper[1] are summarized as follows:

This work was funded by the German ministry of education and research (BMBF) under grants 16KIS1180K, 16KIS1185 (FunKI) and 16KISK016, 16KISK004 (Open6GHub).

[1]*Notation:* Random variables are denoted by sans-serif letters x, random vectors by bold sans-serif letters $\boldsymbol{\mathsf{x}}$, realizations by serif letters x and vector valued realizations by bold serif letters $\boldsymbol{x}$. Sets are denoted by calligraphic letters $\mathcal{X}$. The distribution $p_{\mathsf{x}}(x)$ of a random variable x is abbreviated as $p(x)$. $\mathsf{x} \rightarrow \mathsf{y} \rightarrow \mathsf{z}$ denotes a Markov chain and $\mathbb{R}$, $\mathbb{Z}$, $\mathbb{F}_2$ denotes the real numbers, integers and Galois field 2, respectively.

- we propose the Minimum-Integer Computation (MIC) decoder that processes only low-range integers and realizes both update functions by integer operations,
- in our evaluation, we show that the MIC decoder using only integers of low-range has communication performance of an information maximizing multidimensional LUT (mLUT) decoder,
- in a 22nm Fully-Depleted Silicon-on-Insulator (FD-SOI) implementation we show that the decoder achieves better area and energy efficiency than a SoA NMS decoder.

II. Information Optimized MP for LDPC Codes

A. System Model

We consider a regular LDPC code defined by its parity check matrix $\boldsymbol{H} \in \mathbb{F}_2^{M\times N}$ with d_V ones per column and d_C ones per row. The information word $\boldsymbol{u} \in \mathbb{F}_2^K$ is encoded into the codeword $\boldsymbol{c} \in \mathbb{F}_2^N$ and the Binary Phase Shift Keying (BPSK) modulated vector $\boldsymbol{x} = \boldsymbol{1} - 2\boldsymbol{c}$ is transmitted over an Additive White Gaussian Noise (AWGN) channel leading to the receive vector $\boldsymbol{y} \in \mathbb{R}^N$ given by $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}$ with noise variance σ_n^2. The corresponding Tanner graph consists of VNs for each codebit and Check Nodes (CNs) for each check equation with an edge connecting VN n and CN m if $\boldsymbol{H}_{m,n} = 1$. For the iterative BP decoding, *extrinsic information* for the codebits c_n are exchanged on these edges. We denote the Variable-Node to Check-Node (VN-to-CN) messages as $L_{n\to m} \in \mathbb{R}$ and the Check-Node to Variable-Node (CN-to-VN) messages as $L_{n\leftarrow m} \in \mathbb{R}$. In the first iteration, all VN-to-CN messages are initialized by channel Log-Likelihood Ratios (LLRs) $L^{(0)}_{n\to m} = L(y_n) = \frac{2}{\sigma_\mathsf{n}^2} y_n$ for $n = 1, ..., N$ with $m \in \mathcal{N}_n$ and $\mathcal{N}_n$ defining the set of CNs connected to VN n. At iteration i, the CN update for all CNs $m = 1, ..., M$ and its connected VNs $n \in \mathcal{M}_m$ is

$$L^{(i)}_{n\leftarrow m} = \underbrace{2\,\mathrm{arctanh}\left(\prod_{j\in\mathcal{M}_m\setminus n} \tanh\left(\frac{L^{(i-1)}_{j\to m}}{2}\right)\right)}_{f_{\mathrm{C,BP}}(\cdot)} . \quad (1)$$

The VN update for VNs $n = 1, ..., N$ is

$$L^{(i)}_{n\to m} = \underbrace{L(y_n) + \sum_{v\in\mathcal{N}_n\setminus m} L^{(i)}_{n\leftarrow v}}_{f_{\mathrm{V,BP}}(\cdot)} , \forall m \in \mathcal{N}_n \quad (2)$$

and the bit decision $\hat{c}^{(i)}_{n,\mathrm{BP}}$ is determined by the sign of (2) by adding all messages $L^{(i)}_{n\leftarrow v}$ for $v \in \mathcal{N}_n$ in the summation term.

B. Information Maximizing Quantizer

For FA MP decoding, each receive signal y_n is mapped to a n_Q-bit signed integer $z_n \in \mathcal{Z}$ using the scalar n_Q-bit channel quantizer $Q_\mathsf{ch} : \mathcal{Y} \to \mathcal{Z} = \{-\frac{2^{n_\mathrm{Q}}}{2}, ..., -1, 1, ..., \frac{2^{n_\mathrm{Q}}}{2}\}$. The quantizer Q_ch is designed on symbol level to maximize mutual information $I(\mathsf{x};\mathsf{z})$ considering an AWGN with *design* noise variance σ_w^2. The underlying optimization problem

$$Q^\star_\mathsf{ch} = \underset{Q_\mathsf{ch}}{\mathrm{argmax}}\, I(\mathsf{x};\mathsf{z}) , \quad \text{s.t.} \quad |\mathcal{Z}| = 2^{n_\mathrm{Q}} , \quad (3)$$

is a clustering problem with Kullback-Leibler divergence as similarity measure and a special case of the underlying optimization problem of the Information Bottleneck Method (IBM) [9], [13]. The optimum quantizer for binary input has been derived in [14]. It yields a deterministic function where the joint distribution between symbol x and the quantizer output z given by

$$p(z|x) = \int_{\mathcal{I}_z} \frac{1}{\sqrt{2\pi\sigma_\mathsf{w}^2}} \exp\left(-\frac{(y-x)^2}{2\sigma_\mathsf{w}^2}\right) dy , \quad (4)$$

with $\mathcal{I}_z = \{y \mid Q^\star_\mathsf{ch}(y) = z\}$ as the interval of receive signals y mapped to the quantizer output z.

We denote the LLRs of the quantizer output $z \in \mathcal{Z}$ by $L(z) = \log\left(\frac{p(z|x=1)}{p(z|x=-1)}\right)$. The transformation $z \mapsto L(z)$ is *information lossless* in the sense that $I(\mathsf{x};\mathsf{z}) = I(\mathsf{x};\mathsf{L}(\mathsf{z}))$. In the following, we review an important property of the solution in (3) that will be used again in Sec. III. First, we define the set of all LLRs $\mathcal{L}_\mathsf{y} = \{L(y) \,|\, y \in \mathcal{Y}\}$ in order to specify an equivalent quantizer for LLRs $Q_\mathrm{L} : \mathcal{L}_\mathsf{y} \to \mathcal{Z}$ that achieves the same quantization result $Q_\mathrm{L}(L(y)) = Q^\star_\mathsf{ch}(y) = z$.

Lemma 1 (LLR separation [14]): There exist an optimal quantizer $Q^\star_\mathsf{ch}$ of (3) with an equivalent representation Q_L where two preimages $\mathcal{B}_{z'} = \{L \in \mathcal{L}_\mathsf{y} \,|\, Q_\mathrm{L}(L) = z'\}$ and $\mathcal{B}_{z''} = \{L \in \mathcal{L}_\mathsf{y} \,|\, Q_\mathrm{L}(L) = z''\}$ with $z', z'' \in \mathcal{Z}$ for $z' \neq z''$ are separated by a real valued threshold.

For a discrete set $\mathcal{Y}$, Dabirnia et al. show that this property holds for all information maximizing quantizers [15]. Furthermore, Lemma 1 is a special case of the more general separating hyperplane condition [16].

C. Multidimensional LUT (mLUT) Decoder Design

The mLUT decoder follows the same MP structure of the BP decoder but uses only signed integers as messages on the Tanner graph and signed integer based VN and CN updates. In the mLUT decoder design, we design the VN update for each iteration to maximize *extrinsic information* of a codebit c_n. For the CN update, we utilize an integer based minimum search. The design procedure is based on Density Evolution (DE) with quantization [8] where it is assumed that messages of the same type are independent and identically distributed (i.i.d.) during one iteration, i.e. we can omit the codebit index n. Based on these assumptions it is sufficient to design only one VN update function per iteration. In the following, we keep the iteration index i only in the distribution of messages for simplicity of notation. For the design of the first CN update, we assume that the VN-to-CN messages are initialized by signed integers $t_j \in \mathcal{Z}$ that follow the same distribution $p^{(0)}(t_j|c_j) = p(z|c)$ for $j = 1, ..., d_\mathrm{C} - 1$ determined by (4), using the BPSK relation $x = 1-2c$. For the CN update, we consider the distribution of the $d_\mathrm{C} - 1$ VN-to-CN messages $\boldsymbol{t} = [t_1, ..., t_{d_\mathrm{C}-1}] \in \mathcal{T}^{d_\mathrm{C}-1}_{(0)} = \mathcal{Z}^{d_\mathrm{C}-1}$ and the codebit c at iteration i, given by

$$p^{(i)}(\boldsymbol{t}|c) = \left(\frac{1}{2}\right)^{d_\mathrm{C}-2} \sum_{\boldsymbol{c}:\bigoplus \boldsymbol{c}=c} \prod_{j=1}^{d_\mathrm{C}-1} p^{(i-1)}(t_j|c_j) , \quad (5)$$

where $\bigoplus \boldsymbol{c} = c_1 \oplus ... \oplus c_{d_\text{C}-1}$ is the modulo 2 sum of connected codebits. We utilize the signed integer based minimum search [17] as CN update

$$a = f_{\text{C,min}}(\boldsymbol{t}) = \left(\prod_{j=1}^{d_\text{C}-1} \text{sign}(t_j)\right) \min\{|t_1|, ..., |t_{d_\text{C}-1}|\}. \quad (6)$$

The distribution between the CN-to-VN message $a \in \mathcal{A}_{(i)} = \mathcal{T}_{(i-1)}$ and the codebit c depends on the CN update, i.e.

$$p^{(i)}(a|c) = \sum_{\boldsymbol{t} \in \mathcal{T}_{(i-1)}^{d_\text{C}-1} : f_{\text{C,min}}(\boldsymbol{t}) = a} p^{(i)}(\boldsymbol{t}|c), \quad (7)$$

that can be calculated efficiently in a recursive fashion.

For designing the VN update, we require the joint distribution of the discrete channel information z together with the CN-to-VN messages a_m combined in $\boldsymbol{a} = [z, a_1, ..., a_{d_\text{V}-1}] \in \mathcal{Z} \times \mathcal{A}_{(i)}^{d_\text{V}-1} = \mathcal{V}_{(i)}$ and the codebit c, determined by

$$p^{(i)}(\boldsymbol{a}|c) = p(z|c) \prod_{m=1}^{d_\text{V}-1} p^{(i)}(a_m|c), \quad (8)$$

where $p^{(i)}(a_m|c) = p^{(i)}(a|c)$ for $m = 1, ..., d_\text{V} - 1$. Given the distribution (8), the VN update $f_\text{V}^{(i)}(\mathsf{a}^{(i)})) = \mathsf{t}^{(i)}$ that maximizes Mutual Information (MI) $I(\mathsf{c}; \mathsf{t}^{(i)})$ is determined as the solution of the optimization problem ($\mathsf{c} \to \mathsf{a}^{(i)} \to \mathsf{t}^{(i)}$)

$$f_{\text{V,MI}}^{(i)} = \underset{f_\text{V}}{\text{argmax}}\ I\left(\mathsf{c}; \mathsf{t}^{(i)}\right) \text{ s.t. } |\mathcal{T}_{(i)}| = 2^{n_\text{M}}. \quad (9)$$

Parameter n_M defines the bit-width of the messages exchanged between VN and CN and controls the complexity of the message exchange. For the remainder, $n_\text{M} = n_\text{Q}$ is assumed for simplicity. The optimization problem in (9) equals the channel quantization problem for binary input (Sec. II-B). The optimal deterministic function is a mLUT that conserves more mutual information compared to a design that decomposes the optimization problem into a sequence of 2-dimensional LUTs [8], [9] with the drawback of higher implementation costs for realizing multidimensional LUTs. The distribution of the VN-to-CN messages for the next iteration in (5) is given by

$$p^{(i)}(t_j|c_j) = p^{(i)}(t|c) = \sum_{\boldsymbol{a} \in \mathcal{V}_{(i)} : f_{\text{V,MI}}^{(i)}(\boldsymbol{a}) = t} p^{(i)}(\boldsymbol{a}|c), \quad (10)$$

for $j = 1, ..., d_\text{C} - 1$. For the design of the mutual information maximizing mLUT decoder, we iterate over (5), (7), (8), (9) and (10) and declare convergence if $I(\mathsf{c}; \mathsf{t}^{(i)})$ approaches one after i_max number of iterations.

III. Minimum-Integer Computation Decoder Design

In this section, we replace the information maximizing mLUT for VN updates in (9) by simple integer computations. To motivate the derivation, we consider first the VN update using LLRs for the discrete input messages of n_Q-bit followed by subsequent quantization to n_Q bits for the output. Second, we avoid the inherent addition of real valued LLRs by decomposing this operation into i) mapping of the n_Q-bit input messages into n_R-bit signed integers with $n_\text{R} \geq n_\text{Q}$, ii) execution of integer additions, iii) threshold quantization to n_Q bits. For a sufficiently large n_R these operations replace exactly the mLUT functionality. Furthermore, by decreasing the bit resolution n_R this approach allows to find information optimized approximations with lower VN update complexity.

A. Equivalent LLR Quantizer

Similar to (2), the VN output corresponding to the LLR of the message vector $\boldsymbol{a}$ equals the addition of the LLRs of the channel output z and of the individual messages a_m

$$L^{(i)}(\boldsymbol{a}) = \log\left(\frac{p^{(i)}(\boldsymbol{a}|c=0)}{p^{(i)}(\boldsymbol{a}|c=1)}\right) = L(z) + \sum_{m=1}^{d_\text{V}-1} L^{(i)}(a_m). \quad (11)$$

As a result of Lemma 1, a $|\mathcal{T}_{(i)}| - 1$ threshold quantizer

$$f_{\text{V,MI}}^{(i)}(\boldsymbol{a}) = Q_{\text{V,L}}^{(i)}\left(L^{(i)}(\boldsymbol{a})\right) \quad (12)$$

can be determined that achieves the same output as the information optimal mLUT in (9).

B. Computations over Integers

Our approach is to find an equivalent representation of (12) where the computations over real numbers are replaced by computations over signed integers and an integer quantization function. To that end, we define the signed integer based node calculation[2] as

$$f_{\text{V,IC},s}^{(i)}(\boldsymbol{a}) = Q_{\text{V},s}^{(i)}\left(\underbrace{\lfloor sL(z)\rceil}_{r=\phi^{(i)}(z)} + \sum_{m=1}^{d_\text{V}-1} \underbrace{\lfloor sL^{(i)}(a_m)\rceil}_{r_m=\phi_\text{V}^{(i)}(a_m)}\right), \quad (13)$$

with real valued scaling factor $s > 0$ and integer mappings $\phi^{(i)} : \mathcal{Z} \to \mathcal{R}_s^{(i)} \subset \mathbb{Z}$ and $\phi_\text{V}^{(i)} : \mathcal{A}_{(i)} \to \mathcal{R}_{s,\text{V}}^{(i)} \subset \mathbb{Z}$. The scaling factor s might be different per iteration. We denote the output of the integer mappings as $r = \phi^{(i)}(z)$ and $r_m = \phi_\text{V}^{(i)}(a_m)$, respectively. We interpret $W_s^{(i)}(\boldsymbol{a}) = \lfloor sL(z)\rceil + \sum_{m=1}^{d_\text{V}-1} \lfloor sL^{(i)}(a_m)\rceil \in \mathcal{W}_s^{(i)} \subset \mathbb{Z}$ as an scaled signed integer equivalent of $L^{(i)}(\boldsymbol{a})$ in (11). For any real valued scaling factor $s > 0$, we can construct a quantizer $Q_{\text{V,L},s}^{(i)}(sL^{(i)}(\boldsymbol{a}))$, where any two different preimages are separated by a single threshold and where the thresholds of $Q_{\text{V,L},s}^{(i)}$ are scaled values of the thresholds of $Q_{\text{V,L}}^{(i)}$ in (12). By varying the scaling factor s in (13), we can reduce the sum of relative errors $e^{(i)}(s) = \sum_{\boldsymbol{a} \in \mathcal{V}_{(i)}} |1 - \frac{W_s^{(i)}(\boldsymbol{a})}{sL^{(i)}(\boldsymbol{a})}|$ to find a corresponding integer calculation (13) with the costs of increasing the values (i.e. required bit resolution for representation) in the integer range $\mathcal{R}_s^{(i)}$ and $\mathcal{R}_{s,\text{V}}^{(i)}$ if we increase the scaling factor s. To that end, we define the condition for exact representation via the signed integer based node calculation defined in (13). We use the preimages of the information maximizing VN update in (9) $\mathcal{A}_t^{(i)} = \{\boldsymbol{a} \in \mathcal{V}_{(i)} : f_{\text{V,MI}}^{(i)}(\boldsymbol{a}) = t\}$ and the set of values of $W_s^{(i)}(\boldsymbol{a})$ that are related to one specific t, given by $\mathcal{W}_{s,t}^{(i)} = \{W_s^{(i)}(\boldsymbol{a}) \in \mathcal{W}_s^{(i)} : \boldsymbol{a} \in \mathcal{A}_t^{(i)}\}$. The output of the

[2] With $\lfloor\cdot\rceil$ as round to nearest integer (away from 0 if fraction part is .5)

information maximizing VN update can be calculated exactly if different sets $\mathcal{W}_{s,t}^{(i)}$ are disjoint, i.e.

$$\mathcal{W}_{s,t'}^{(i)} \cap \mathcal{W}_{s,t''}^{(i)} = \emptyset \quad \forall t'', t' \in \mathcal{T}_{(i)} \text{ for } t'' \neq t' \,. \tag{14}$$

Condition (14) ensures that the output of $f_{\text{V,MI}}^{(i)}(\boldsymbol{a})$ in (12) can be represented exactly by using a suitable quantization function $Q_{\text{V},s}^{(i)}$ in (13).

Our aim is to keep the single threshold separation property to simplify the quantization operation $Q_{\text{V},s}^{(i)}$. For this purpose, we define a restricted condition of (14) by *non-overlapping intervals* as

$$\left[W_{s,t',\text{L}}^{(i)}, W_{s,t',\text{R}}^{(i)}\right] \cap \left[W_{s,t'',\text{L}}^{(i)}, W_{s,t'',\text{R}}^{(i)}\right] = \emptyset \,, \forall t'', t' \in \mathcal{T}_{(i)} \,, \tag{15}$$

with $W_{s,t,\text{L}}^{(i)} = \min \mathcal{W}_{s,t}^{(i)}$ and $W_{s,t,\text{R}}^{(i)} = \max \mathcal{W}_{s,t}^{(i)}$. For exact representation, we search for the minimum value of s such that condition (15) is satisfied. For n_R-bit limited integer calculation, we ensure that the integer valued range $\mathcal{R}_s^{(i)}$ and $\mathcal{R}_{s,\text{V}}^{(i)}$ of the integer mappings in (13) satisfy

$$\max_z |\phi^{(i)}(z)| < 2^{n_\text{R}} \quad , \quad \max_a |\phi_\text{V}^{(i)}(a)| < 2^{n_\text{R}} \tag{16}$$

for all iterations $1 \leq i \leq i_\text{max}$ and select the output cluster that maximizes mutual information if condition (15) is not fulfilled.

IV. Node Hardware Architectures

With the definition of the MIC node update functions in (6) and (13), we can derive respective FP hardware architectures, that constitute the main building blocks for an unrolled FP LDPC decoder.

A. Check Node Functional Unit (CFU)

The Check Node Functional Unit (CFU) of the MIC decoder is similar to that of a conventional NMS or OMS decoder. It is composed of a minimum search stage with a subsequent output stage for each output message. For a detailed description, we refer to the relevant literature, e.g. [5]. In contrast to a conventional CFU, there is no need to perform a Sign-Magnitude (SM)/Two's Complement (TC) conversion as well as the message normalization, respectively. Therefore, respective units are omitted.

B. Variable Node Functional Unit (VFU)

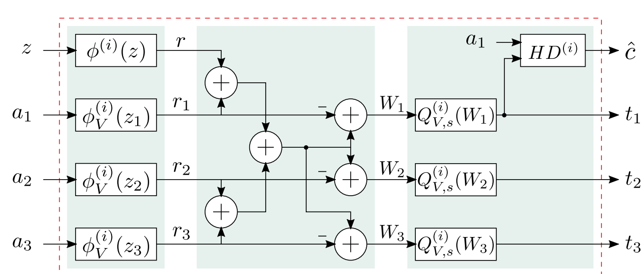

Fig. 1. Variable Node Functional Unit for a degree $d_\text{V} = 3$ VN.

The Variable Node Functional Unit (VFU) for a degree $d_\text{V} = 3$ MIC VN is depicted in Fig. 1. It is composed of an adder tree (center), similar to a conventional full-parallel VFU [5], embedded by respective input (left) and output (right) stages. The discrete channel information z, the incoming and outgoing messages a_m and t_m, with $1 \leq m \leq d_\text{V}$ are each $n_\text{M} = n_\text{Q}$ bit, whereas the corresponding reconstructed messages r and r_m are each n_R bit wide. The input and output stages perform the required integer mappings. Furthermore, the input and output stages perform a number format conversion from sign-magnitude to two's complement and vice versa. The latter is already taken into account in the design of the one dimensional LUTs ϕ, ϕ_V and $Q_{\text{V},s}$. Similarly, the hard-decision bit $\hat{c}$ is obtained by evaluation of the respective two dimensional LUT.

V. Results and Discussion

The MIC approach is evaluated in terms of FEC performance and implementation efficiency for a $(816, 406)$ regular LDPC code with VN degree $d_\text{V} = 3$ and CN degree $d_\text{C} = 6$.

A. Frame Error Rate (FER)

Fig. 2 shows the FER performance of the MIC decoder for integer bit-widths $n_\text{R} \in \{4, 5, 6\}$ with $n_\text{M} = n_\text{Q} = 3$ quantization and message bit-width. The decoders were designed for a target FER of 10^{-3}. As baseline, we include the FER of a double precision floating point BP decoder without quantization of receive signals as well as the FERs for NMS decoder with scaling factor $\gamma = 0.75$ and a $n_\text{M} = n_\text{Q} = 3$ mLUT decoder. All decoders use the flooding schedule and are limited to 10 decoding iterations for practical reasons.

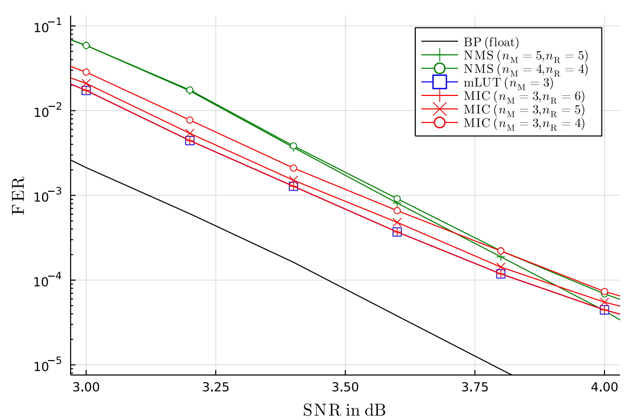

Fig. 2. FER of the MIC decoder for different integer bit-widths n_R in comparison with various SoA decoders at ten iterations.

For $n_\text{R} = 6$ the MIC decoder achieves the same FER as the mLUT decoder by meeting condition (16). For lower integer resolutions $n_\text{R} = 5$ and $n_\text{R} = 4$, we observe a performance degradation of less than 0.1 dB for the considered target FER. Thus, n_R can be used to trade-off computational complexity and performance of the decoder. In contrast to conventional FA decoder designs, the bit-width of the exchanged messages n_M, that is crucial for the routability of the decoder, can remain at small values (here $n_\text{M} = 3$).

For the NMS decoder with $n_\mathrm{Q}=3$, a message quantization of at least $n_\mathrm{M}=n_\mathrm{R}=4$ is required to achieve a FER close to that of the MIC decoder. An increase to $n_\mathrm{M}=n_\mathrm{R}=5$ shows a slight improvement, but overall the performance does not exceed the MIC decoder with $n_\mathrm{R}=4$. This is also the case for higher message resolutions in the NMS decoder, which are omitted in this plot for better clarity.

B. Implementation Results

Implementation efficiency of our MIC approach is evaluated on node level. I.e. VFUs and CFUs are implemented in a 22 nm FD-SOI technology using a Super-Low-V_T (SLVT) standard library under worst-case Process, Voltage and Temperature (PVT) conditions. Synthesis is performed with Synopsys Design Compiler. The MIC node implementations are then compared with their counterparts from a SoA NMS decoder. In detail, we compare the $n_\mathrm{M}=3$, $n_\mathrm{R}=4$ MIC nodes with the $n_\mathrm{M}=n_\mathrm{R}=4$ NMS nodes since they provide comparable FER performance, both with $n_\mathrm{Q}=3$. We also extrapolate the results to decoder level.

TABLE I
POST-SYNTHESIS RESULTS OF VN AND CN FUNCTIONAL UNITS

	VFU			CFU		
	NMS	**MIC**	**Imp.**	**NMS**	**MIC**	**Imp.**
Critical Path $[ns]$	0.24	0.24	0%	0.29	0.15	48%
Area $[um^2]$	200	179	11%	269	143	47%
Power $[uW]$	382	322	15%	518	252	51%

Table I shows the corresponding synthesis results. Here, we only list the results of the first iteration VFU, as the variations in timing, area and power over the iterations are negligible. When looking at the results, the length of the critical paths of the VFUs for NMS and MIC are similar, but area and power of the MIC VFU are decreased by 11% and 15%, respectively. The improvements in the CFU are much larger with around 50% in critical path, area and power. The large gains in the CFU stem from i) a lower message bit width n_M (3 vs. 4 bit), ii) the removal of the message normalization unit, and iii) the removal of the SM/TC conversion unit. A similar observation was made in [12], where a 5 bit MS decoder was compared to a 3 bit Min-LUT decoder for a $d_\mathrm{V}=6$, $d_\mathrm{C}=32$ code, where the CFU area was reduced by around 50%, but the VFU area increased by 100%.

We can perform a first fast extrapolation of these area results to an unrolled FP decoder architecture. For this, we consider a code that contains 816 VNs and 406 CNs. This extrapolation yields a logic area reduction by about 25%. Routing area is first order also reduced by 25% since the the bit widths, and thus the wiring, reduces from 4 bits to 3 bits. It has to be mentioned that the real savings would be larger since the routing area grows rather quadratically than linearly for routing dominated architectures. Moreover it is expected that the power is also largely reduced. However, the implementation of a full decoder is out of scope of this paper.

But this simplified extrapolation shows that an unrolled LDPC decoder architecture can be implemented much more efficiently with our new MIC approach compared to a NMS decoder without loss in FER performance. Furthermore, this approach enables the possibility to address a major issue for high throughput LDPC decoder implementations, i.e. the implementations of larger code block sizes that is constraint by routing congestion for SoA NMS decoder architectures.

VI. CONCLUSION

We presented the Minimum-Integer Computation (MIC) decoder based on FA MP where the multidimensional mutual information maximizing Variable Node (VN) Lookup-Table (LUT) is replaced by an equivalent low-range integer calculation. This equivalent integer representation calculates the output of the VN LUT exactly or approximately depending on the bit resolution of the internal integer calculations. For the implementation, we showed that the MIC design has lower complexity than the Normalized Min-Sum (NMS) decoder with comparable performance.

REFERENCES

[1] W. Saad, M. Bennis, and M. Chen, "A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems," *IEEE Network*, vol. 34, no. 3, pp. 134–142, 2020.
[2] C. Kestel, M. Herrmann, and N. Wehn, "When Channel Coding Hits the Implementation Wall," in *IEEE 10th International Symposium on Turbo Codes Iterative Information Processing (ISTC 2018)*, 2018.
[3] R. Gallager, "Low-Density Parity-Check Codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.
[4] D. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, no. 2, 1999.
[5] P. Schläfer, N. Wehn, M. Alles, and T. Lehnigk-Emden, "A New Dimension of Parallelism in Ultra High Throughput LDPC Decoding," in *SiPS 2013 Proceedings*, 2013, pp. 153–158.
[6] R. Ghanaatian, A. Balatsoukas-Stimming, T. C. Müller, M. Meidlinger, G. Matz, A. Teman, and A. Burg, "A 588-Gb/s LDPC Decoder Based on Finite-Alphabet Message Passing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 2, pp. 329–340, 2018.
[7] J. Chen and M. Fossorier, "Density Evolution for Two Improved BP-Based Decoding Algorithms of LDPC Codes," *IEEE Communications Letters*, vol. 6, no. 5, pp. 208–210, 2002.
[8] F. J. C. Romero and B. M. Kurkoski, "LDPC Decoding Mappings that Maximize Mutual Information," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 9, pp. 2391–2401, Aug. 2016.
[9] J. Lewandowsky and G. Bauch, "Information-Optimum LDPC Decoders based on the Information Bottleneck Method," *IEEE Access*, Jan. 2018.
[10] X. He, K. Cai, and Z. Mei, "On Mutual Information-Maximizing Quantized Belief Propagation Decoding of LDPC Codes," in *IEEE Global Communications Conference (GLOBECOM 2019)*, Dec. 2019.
[11] L. Wang, R. D. Wesel, M. Stark, and G. Bauch, "A Reconstruction-Computation-Quantization (RCQ) Approach to Node Operations in LDPC Decoding," in *IEEE Global Comm. Conf. (GLOBECOM)*, 2020.
[12] A. Balatsoukas-Stimming, M. Meidlinger, R. Ghanaatian, G. Matz, and A. Burg, "A Fully-Unrolled LDPC Decoder based on Quantized Message Passing," in *IEEE Works. on Signal Proc. Systems (SiPS)*, 2015.
[13] B. M. Kurkoski, "On the Relationship Between the KL Means Algorithm and the Information Bottleneck Method," in *11th International ITG Conference on Systems, Communications and Coding (SCC)*, Feb. 2017.
[14] B. M. Kurkoski and H. Yagi, "Quantization of Binary-Input Discrete Memoryless Channels," *IEEE Transactions on Information Theory*, vol. 60, no. 8, pp. 4544–4552, 2014.
[15] M. Dabirnia, A. Martinez, and A. G. i Fàbregas, "A Recursive Quantizer Design Algorithm for Binary-Input Discrete Memoryless Channels," *IEEE Transactions on Communications*, vol. 69, no. 8, 2021.
[16] D. Burshtein, V. D. Pietra, D. Kanevsky, and A. Nadas, "Minimum Impurity Partitions," *The Annals of Statistics*, vol. 20, no. 3, 1992.
[17] M. Meidlinger, G. Matz, and A. Burg, "Design and Decoding of Irregular LDPC Codes Based on Discrete Message Passing," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1329–1343, 2020.