entropy

MDPI

Article

Minimum-Integer Computation Finite Alphabet Message Passing Decoder: From Theory to Decoder Implementations towards 1 Tb/s

Tobias Monsees [1,*], Oliver Griebel [2], Matthias Herrmann [2], Dirk Wübben [1], Armin Dekorsy [1] and Norbert Wehn [2]

[1] Department of Communications Engineering, University of Bremen, 28359 Bremen, Germany
[2] Microelectronic Systems Design Research Group, TU Kaiserslautern, 67663 Kaiserslautern, Germany
* Correspondence: tmonsees@ant.uni-bremen.de; Tel.: +49-421-218-62407

Abstract: In Message Passing (MP) decoding of Low-Density Parity Check (LDPC) codes, extrinsic information is exchanged between Check Nodes (CNs) and Variable Nodes (VNs). In a practical implementation, this information exchange is limited by quantization using only a small number of bits. In recent investigations, a novel class of Finite Alphabet Message Passing (FA-MP) decoders are designed to maximize the Mutual Information (MI) using only a small number of bits per message (e.g., 3 or 4 bits) with a communication performance close to high-precision Belief Propagation (BP) decoding. In contrast to the conventional BP decoder, operations are given as discrete-input discrete-output mappings which can be described by multidimensional LUTs (mLUTs). A common approach to avoid exponential increases in the size of mLUTs with the node degree is given by the sequential LUT (sLUT) design approach, i.e., by using a sequence of two-dimensional Lookup-Tables (LUTs) for the design, leading to a slight performance degradation. Recently, approaches such as Reconstruction-Computation-Quantization (RCQ) and Mutual Information-Maximizing Quantized Belief Propagation (MIM-QBP) have been proposed to avoid the complexity drawback of using mLUTs by using pre-designed functions that require calculations over a computational domain. It has been shown that these calculations are able to represent the mLUT mapping exactly by executing computations with infinite precision over real numbers. Based on the framework of MIM-QBP and RCQ, the Minimum-Integer Computation (MIC) decoder design generates low-bit integer computations that are derived from the Log-Likelihood Ratio (LLR) separation property of the information maximizing quantizer to replace the mLUT mappings either exactly or approximately. We derive a novel criterion for the bit resolution that is required to represent the mLUT mappings exactly. Furthermore, we show that our MIC decoder has exactly the communication performance of the corresponding mLUT decoder, but with much lower implementation complexity. We also perform an objective comparison between the state-of-the-art Min-Sum (MS) and the FA-MP decoder implementations for throughput towards 1 Tb/s in a state-of-the-art 28 nm Fully-Depleted Silicon-on-Insulator (FD-SOI) technology. Furthermore, we demonstrate that our new MIC decoder implementation outperforms previous FA-MP decoders and MS decoders in terms of reduced routing complexity, area efficiency and energy efficiency.

Keywords: LDPC code; decoding; finite alphabet message passing; information bottleneck; implementation efficiency

check for updates

Citation: Monsees, T.; Griebel, O.; Herrmann, M.; Wübben, D.; Dekorsy, A.; Wehn, N. Minimum-Integer Computation Finite Alphabet Message Passing Decoder: From Theory to Decoder Implementations towards 1 Tb/s. *Entropy* **2022**, *24*, 1452. https://doi.org/10.3390/e24101452

Academic Editor: Syed A. Jafar

Received: 26 August 2022
Accepted: 8 October 2022
Published: 12 October 2022

1. 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 a Forward Error Correction (FEC) decoder, which is one of the most complex and computationally intense components 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 competitive FEC codes, like Polar and Turbo codes, the decoding of LDPC codes is dominated by data transfers [2] making very high-throughput decoders in advanced silicon technologies challenging, especially from routing and energy efficiency perspectives. For example, in a state-of-the-art 14 nm silicon technology, the transfer of 8 bits on a 1 mm wire costs about 1 pJ, whereas the cost of an 8 bit integer addition is only 10 fJ, which is two orders of magnitude less than the wiring energy cost. During Message Passing (MP) decoding, two sets of nodes, the Check Node (CN) and Variable Node (VN), iteratively exchange messages over the edges of a bipartite graph (Tanner graph of the LDPC code). High-throughput decoding can be achieved by mapping the Tanner graph one-to-one onto hardware, i.e., dedicated processing units are instantiated for each node and the edges of the Tanner graph are hardwired. Unrolling and pipelining the decoding iterations can further boost the throughput towards 1 Tb/s [5], called unrolled full parallel (FP) decoders in the following. However, FP decoders imply large routing challenges, since every edge in the Tanner graph corresponds to $2 \cdot I \cdot n_{\mathrm{E}}$ wires, with I being the number of decoding iterations and n_{E} being the quantization-width of the exchanged messages. Moreover, to enable good error correction performance, the Tanner graph exhibits limited locality and regularity, which makes efficient routing even more difficult. This problem is even exacerbated in advanced silicon technologies, as routing scales much worse than transistor density [6].

Finite Alphabet Message Passing (FA-MP) decoding has been investigated as a method to mitigate the routing challenges in FP LDPC decoders to reduce the bit-width, i.e., the quantization-width n_{E}, of the exchanged messages and, thus, the number of necessary wires [7–9]. In contrast to conventional MP decoding algorithms like the Belief Propagation (BP) and its approximations, i.e., Min-Sum (MS), Offset Min-Sum (OMS) and Normalized Min-Sum (NMS) [10], FA-MP use non-uniform quantizers and the node operations are derived by maximizing MI between exchanged messages. Nodes in state-of-the-art FA-MP decoders have to be implemented as Lookup-Tables (LUTs). Since the size of the LUT exponentially increases with the node degree and n_{E}, investigations were performed to decompose this multidimensional LUT (mLUT) into a chain or tree with only two-input LUTs (denoted as sequential LUT (sLUT) in this paper) yielding only a linear dependency of the node degree but at the cost of a decreased communications performance [11,12]. The Minimum-LUT (Min-LUT) decoder [13] approximates the CN update by a simple minimum search and can be implemented as Minimum-mLUT (Min-mLUT) or Minimum-sLUT (Min-sLUT), i.e., with mLUT or sLUT for VNs, respectively. Other approaches, e.g., Mutual Information-Maximizing Quantized Belief Propagation (MIM-QBP) [14–16] and Reconstruction-Computation-Quantization (RCQ) [17,18], are adding non-uniform quantizers and reconstruction mappings to the outputs and inputs of the nodes, respectively, and performing the standard functional operations inside the nodes, e.g., additions for VNs and minimum search for CNs. The reconstruction mappings generally increase the bit resolution required for node internal representation and processing. It can be shown that this approach is equivalent in terms of error correction performance compared to the mLUT, if the internal quantization after the reconstruction mapping is sufficiently large.

Based on the framework of MIM-QBP and RCQ, the proposed MIC decoder [19] realizes CN updates by a minimum search and VN updates by integer computations that are designed to realize the information maximizing mLUT mappings either exactly or approximately. In this paper, we provide more detailed explanations, extend the discussion to irregular LDPC codes and present a comprehensive implementation analysis. The new contributions of this paper (*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.) are summarized as follows:

- We provide a novel criterion for the resolution of internal node operations to ensure that the MIC decoder can always replace the information maximizing VN mLUT exactly;
- we show that this MIC decoder has the same communication performance compared to an MI maximizing Min-mLUT decoder;
- we make an objective comparison between different FA-MP decoder implementations (Min-mLUT, Min-sLUT, MIC) in an advanced silicon technology and compare them with a state-of-the-art MS decoder for throughput towards 1 Tb/s;
- we show that our MIC decoder implementation outperforms state-of-the-art FP decoders in terms of routing complexity, area efficiency and energy efficiency and enables the processing of larger block sizes in state-of-the-art FP decoders since the routing complexity is largely reduced.

The remainder of this paper is structured as follows: Section 2 reviews the system model, conventional decoding techniques for LDPC codes such as BP and NMS decoding, and Information Bottleneck (IB) based quantization. Section 3 describes the Min-mLUT and Min-sLUT decoder design for regular and irregular LDPC codes. In Section 4, we introduce the proposed MIC decoder and, in Section 5, we discuss the MIC decoder implementation along with a detailed comparison with state-of-the-art FP MP decoders. Finally, Section 6 concludes the paper.

2. Preliminaries

This section briefly reviews the transmission model, conventional decoding techniques for LDPC codes, and the quantizer design based on IB.

2.1. Transmission Model

The transmission model is shown in Figure 1. An information word $\boldsymbol{u} \in \mathbb{F}_2^K$ is encoded into the codeword $\boldsymbol{c} \in \mathbb{F}_2^N$ via a binary LDPC code [3] of rate $R = \frac{K}{N}$. The Binary Phase Shift Keying (BPSK) modulated vector $\boldsymbol{x} = \mathbf{1} - 2\boldsymbol{c}$ is transmitted over an Additive White Gaussian Noise (AWGN) channel leading to the received vector $\boldsymbol{y} \in \mathbb{R}^N$ given by $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}$ with AWGN $\boldsymbol{n}$ of variance σ_n^2. A particular LDPC code is defined via a sparse parity check matrix $\boldsymbol{H} \in \mathbb{F}_2^{M\times N}$. The Tanner graph [20] of an LDPC code is a visual representation of its parity check matrix $\boldsymbol{H}$ and consists of a CN for each parity check equation χ_m with $m = 1, ..., M$ and a VN for each codebit c_n with $n = 1, ..., N$. An edge connects VN n and CN m if and only if $\boldsymbol{H}_{m,n} = 1$. The degree of a node is determined by the number of connected edges. Furthermore, the fraction of edges that is connected to a node of a specific degree is characterized by the edge-degree distributions

$$\lambda(\xi) = \sum_{d_V \in \mathcal{D}_V} \lambda_{d_V} \xi^{d_V - 1} \quad \text{and} \quad \rho(\xi) = \sum_{d_C \in \mathcal{D}_C} \rho_{d_C} \xi^{d_C - 1} \tag{1}$$

where λ_{d_V} is the fraction of edges that are connected to VNs of degree $d_V \in \mathcal{D}_V$, and ρ_{d_C} denotes the fraction of edges that is connected to CNs of degree $d_C \in \mathcal{D}_C$.

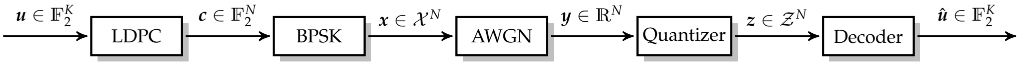

$\boldsymbol{u} \in \mathbb{F}_2^K$
LDPC
$\boldsymbol{c} \in \mathbb{F}_2^N$
BPSK
$\boldsymbol{x} \in \mathcal{X}^N$
AWGN
$\boldsymbol{y} \in \mathbb{R}^N$
Quantizer
$\boldsymbol{z} \in \mathcal{Z}^N$
Decoder
$\hat{\boldsymbol{u}} \in \mathbb{F}_2^K$

Figure 1. Transmission model for transmission of LDPC encoded messages over an AWGN channel with quantization prior to FEC decoding.

2.2. Iterative Decoding via Belief-Propagation (BP)

LDPC codes are usually decoded by iterative BP, where *extrinsic information* for each codebit c_n is propagated along the edges of the resulting Tanner graph. Figure 2 shows the CN χ_1 that generates *extrinsic information* for the VN c_n by processing the incoming Variable Node to Check Node (VN-to-CN) messages from the other VNs connected to CN χ_1, i.e., c_1 and c_2. For BP decoding, we define 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 messages are initialized by the channel LLRs

$$L_{n\to m}^{(0)} = L(y_n) = \tfrac{2}{\sigma_\text{n}^2} y_n \,. \tag{2}$$

In iteration i, a CN m generates *extrinsic information* for the connected VNs $n \in \mathcal{M}_m$ via the box-plus operation

$$L_{n\leftarrow m}^{(i)} = 2\,\text{arctanh}\left(\prod_{j\in\mathcal{M}_m\setminus n} \tanh\left(\tfrac{1}{2} L_{j\to m}^{(i-1)}\right)\right), \quad \forall n \in \mathcal{M}_m \,. \tag{3}$$

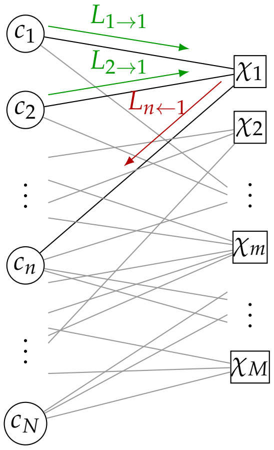


Figure 2. Illustrative example of a CN update on a Tanner graph. The CN χ_1 generates the CN-to-VN message $L_{n\leftarrow1}$ for the VN c_n based on the VN-to-CN messages $L_{1\to1}$ and $L_{2\to1}$ from VN c_1 and c_2, respectively.

In case of Normalized Min-Sum (NMS) decoding, the CN update (3) is approximated by

$$L_{n\leftarrow m}^{(i)} \approx \gamma \left(\prod_{j\in\mathcal{M}_m\setminus n} \text{sign}\left(L_{j\to m}^{(i-1)}\right)\right) \min_{j\in\mathcal{M}_m\setminus n} \left|L_{j\to m}^{(i-1)}\right|, \quad \forall n \in \mathcal{M}_m \,. \tag{4}$$

where γ is the normalization factor. In the case of $\gamma = 1$, (4) is the CN update of the MS decoder.

In similar fashion, a VN n generates *extrinsic information* for the connected CNs $m \in \mathcal{N}_n$ by adding the corresponding LLRs

$$L_{n\to m}^{(i)} = L(y_n) + \sum_{v\in\mathcal{N}_n\setminus m} L_{n\leftarrow v}^{(i)}\,, \quad \forall m \in \mathcal{N}_n \,. \tag{5}$$

The final bit decision $\hat{c}_{n,\text{BP}}^{(i)}$ at iteration i is determined by

$$\hat{c}_{n,\text{BP}}^{(i)} = \frac{1}{2}\left(1 - \text{sign}\left(L(y_n) + \sum_{v\in\mathcal{N}_n} L_{n\leftarrow v}^{(i)}\right)\right). \tag{6}$$

2.3. Information Bottleneck Based Quantizer Design

For the design of our proposed MIC decoder, we utilize MI maximizing quantization to design an information optimized processing chain that uses only quantizer labels instead of real valued representations [12]. To that end, we first review the principle idea of the MI based quantizer design approach. The considered system model is visualized in Figure 3. The observed signal $y \in \mathcal{Y}$ is mapped to a compressed representation $z \in \mathcal{Z}$ via the scalar quantization function $Q : \mathcal{Y} \to \mathcal{Z}$. The objective is to find a quantizer function $Q^\star$ that maximizes MI $I(\mathrm{x};\mathrm{z})$ between the relevant source $x \in \mathcal{X}$ and the quantizer output $Q(y) = z \in \mathcal{Z}$ under the condition that the three random variables form a Markov chain $\mathrm{x} \to \mathrm{y} \to \mathrm{z}$. Given the joint distribution $p(x, y) = p(y|x)p(x)$, the mapping of the information maximizing quantizer $Q^\star$ is determined by solving the optimization problem

$$Q^\star = \underset{Q}{\operatorname{argmax}}\, I(\mathrm{x};\mathrm{z}) \quad \text{s.t.} \quad |\mathcal{Z}| = 2^{n_\mathrm{Q}} < |\mathcal{Y}| \tag{7}$$

where the number of possible quantizer outputs is set to 2^{n_Q}. The optimization problem in (7) is a special case of the Information Bottleneck Method (IBM) [12,21–23]. The optimal solution is a deterministic quantization function where the conditional probability of the quantizer output z given the relevant source x is

$$p(z|x) = \sum_{y \in \mathcal{Y}_z} p(y|x) \tag{8}$$

with $\mathcal{Y}_z = \{y \in \mathcal{Y} \mid Q^\star(y) = z\}$ as the set of observed signals y that are mapped to one specific quantizer output z. Since the maximum of (7) depends only on the cardinality of $\mathcal{Z}$, we utilize a convenient signed integer based representation $\mathcal{Z} = \{-\frac{2^{n_\mathrm{Q}}}{2}, \ldots, -1, 1, \ldots, \frac{2^{n_\mathrm{Q}}}{2}\}$ that simplifies the MIC decoder processing. For the special case where the relevant source x is a binary random variable (i.e., $|\mathcal{X}| = 2$), the algorithm that finds the optimal quantizer via dynamic programming has been derived in [24]. We denote the LLRs of the quantizer output $z \in \mathcal{Z}$ by

$$L(z) = \log\left(\frac{p(z|x = +1)}{p(z|x = -1)}\right). \tag{9}$$

An important property of the MI maximizing quantizer for binary input is that any two different sets of LLRs $\mathcal{L}_{z'} = \{L(y) \,|\, y \in \mathcal{Y}_{z'}\}$ and $\mathcal{L}_{z''} = \{L(y) \,|\, y \in \mathcal{Y}_{z''}\}$ for $z', z'' \in \mathcal{Z}$ and $z' \neq z''$ are separated by a single threshold [19,24,25]. This property will be exploited in the design of the MIC decoder in Section 4.

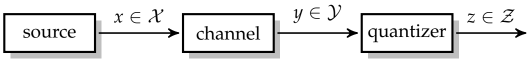


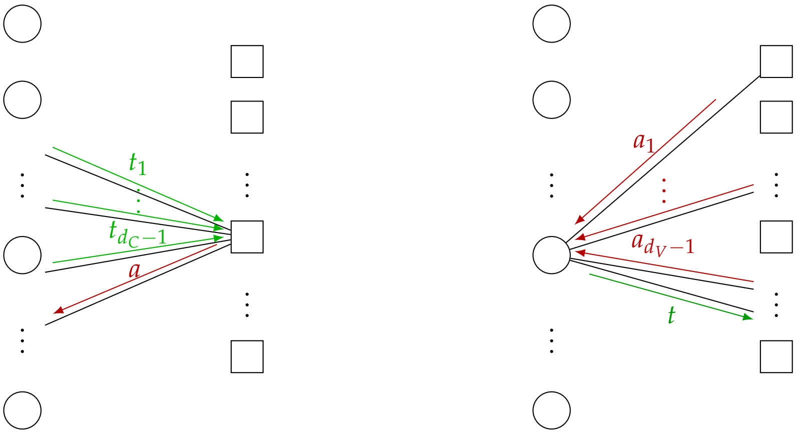
Figure 3. Considered system model for quantizer design.

3. LUT Decoder Design

This section describes the design of the LUT decoder that is optimized via Discrete Density Evolution (DDE) [11] to maximize *extrinsic information* between the codebits and its messages, under the assumption that the Tanner graph is cycle free. In contrast to the BP algorithm, the LUT is optimized to process the quantizer labels z in (7) directly and the bit resolution of the message exchange on the Tanner graph is limited to n_E bits, e.g., 3 or 4 bits. Furthermore, we exploit the signed integer-based representation to simplify the CN update by using the label-based minimum search [13]. In the Min-mLUT decoder design, the VN update functions are optimized to maximize MI. For the Min-sLUT decoder design, the VN update is decomposed into a sequence of two-dimensional updates that generally results in a MI loss compared to the Min-mLUT decoder design.

In the following, we review the calculation of the CN and VN distributions for each iteration that are required for the design of the MI maximizing VN update. As illustrated in Figure 4, we omit the iteration index i and consider messages of an arbitrary CN and

VN for CN degrees $d_C \in \mathcal{D}_C$ and VN degrees $d_V \in \mathcal{D}_V$ to calculate the distributions that are required for the Min-mLUT design.



(**a**) CN update (**b**) VN update

Figure 4. Illustrative example for generation of extrinsic information in case of LUT decoding using discrete messages. (**a**) visualizes a CN that generates the CN-to-VN message a based on incoming VN-to-CN messages $t_1, ..., t_{d_C-1}$. In (**b**), a VN generates the VN-to-CN message t based on incoming CN-to-VN messages $a_1, ..., a_{d_V-1}$.

3.1. Check Node LUT Design

The LUT decoder design is based on discrete alphabets $\mathcal{Z}$, $\mathcal{T}$ and $\mathcal{A}$ for the channel information, the VN-to-CN and the CN-to-VN messages, respectively. For the first iteration, the VN-to-CN messages t_j for $j = 1, ..., d_V - 1$ are initialized by the signed integer valued channel information, i.e., $t_j = z_j \in \mathcal{Z}$. The distribution of the $d_C - 1$ VN-to-CN messages $\boldsymbol{t} = [t_1, ..., t_{d_C-1}] \in \mathcal{T}^{(d_C-1)}$ and an arbitrary codebit c of a check equation χ is [11]

$$p_{d_C}(\boldsymbol{t}|c) = \left(\frac{1}{2}\right)^{d_C-2} \sum_{\boldsymbol{b}:\bigoplus \boldsymbol{b}=c} \prod_{j=1}^{d_C-1} p(t_j|b_j) \tag{10}$$

with $\bigoplus \boldsymbol{b} = b_1 \oplus ... \oplus b_{d_C-1}$ as the modulo 2 sum of connected codebits. The VN-to-CN messages t_j are processed by a CN update function that generates quantized output messages $a \in \mathcal{A}$ that are represented only by n_E bits.

Given the distribution in (10), the CN update (We keep the node degrees d_C or d_V as index of random variables to indicate that the distribution changes with the corresponding degrees.) $f_{d_C}(\mathbf{t}_{d_C}) = \mathsf{a}_{d_C}$ that maximizes MI is determined by the solution of the quantization problem for binary input ($\mathsf{c} \to \mathbf{t}_{d_C} \to f_{d_C}(\mathbf{t}_{d_C}) = \mathsf{a}_{d_C}$)

$$f_{d_C}^{\mathrm{MI}} = \underset{f_{d_C}}{\operatorname{argmax}}\ I(\mathsf{c}; \mathsf{t}_{d_C}) \text{ s.t. } |\mathcal{A}| = 2^{n_E} \quad \text{for } d_C \in \mathcal{D}_C\,. \tag{11}$$

As discussed in Section 2.3, the optimal solution of (11) is found via dynamic programming.

However, we utilize the minimum update [13] as a CN update for all iterations as an approximation of the MI maximizing CN update in (11). We observed that the output of the minimum update is quite close to the optimal IB update. As visualized for a degree 3 CN in Figure 5, the difference between the optimal IB CN and the minimum update can be interpreted as an additive *correction* LUT where only a small fraction of entries are nonzero. For the label-based minimum search, the CN update rule reads

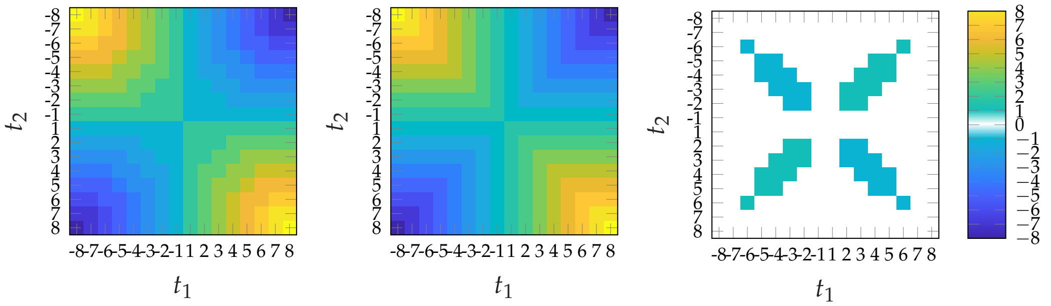
$$a = f_{d_C}^{\min}(\boldsymbol{t}) = \left(\prod_{j=1}^{d_C-1} \operatorname{sign}(t_j)\right) \min\{|t_1|, ..., |t_{d_C-1}|\}\,. \tag{12}$$

If the CN update function is given, the conditional distribution of the CN-to-VN messages $a \in \mathcal{A} = \mathcal{T}$ is

$$p_{d_C}(a|c) = \sum_{\boldsymbol{t}\in\mathcal{T}^{(d_C-1)}:\, f^{\min}_{d_C}(\boldsymbol{t})=a} p_{d_C}(\boldsymbol{t}|c) \quad \text{for } \; d_C \in \mathcal{D}_C\,. \tag{13}$$

In the design via DDE, the connections between VNs and CNs are considered on average by the degree distribution [26]. Hence, the design considers only the marginal CN-to-VN message distribution $p(a|c)$ that includes averaging over all possible CN degrees by

$$p(a|c) = \sum_{d_C\in\mathcal{D}_C} p_{d_C}(a|c)\rho_{d_C}\,. \tag{14}$$



(**a**) MI maximizing update (**b**) Minimum update (**c**) *correction* LUT

Figure 5. Graphical representation of a discrete CN update using $n_\mathrm{E} = 4$ bit input messages t_1 and t_2 and a color-coded output message $a \in \mathcal{A} = \{-4, ..., -1, 1, ..., 4\}$. Subfigure (**a**) shows the MI maximizing update $f^{\mathrm{MI}}_3(t_1, t_2)$ and subfigure (**b**) the minimum update $f^{\min}_3(t_1, t_2)$. The difference $f^{\mathrm{MI}}_3(t_1, t_2) - f^{\min}_3(t_1, t_2)$ in subfigure (**c**) contains only a few non-zero elemets and can be interpreted as a *correction* LUT.

3.2. Variable Node LUT Design

For designing the VN update, we require the joint distribution of the discrete channel information $z \in \mathcal{Z}$ together with the CN-to-VN messages $a_m \in \mathcal{A}$ combined in $\boldsymbol{a} = [z, a_1, ..., a_{d_V-1}] \in \mathcal{Z} \times \mathcal{A}^{d_V-1} = \mathcal{V}$ and a codebit c [11]

$$p_{d_V}(\boldsymbol{a}|c) = p(z|c)\prod_{m=1}^{d_V-1} p(a_m|c) \tag{15}$$

where $p(a_m|c) = p(a|c)$ for $m = 1, ..., d_{V,\max} - 1$ and $\mathcal{V}$ is the set of all possible states of the vector $\boldsymbol{a}$, i.e., $|\mathcal{V}| = 2^{n_\mathrm{Q}+(d_V-1)n_\mathrm{E}}$. Given the distribution (15), the individual degree-dependent VN update $g_{d_V}(\mathsf{a}_{d_V}) = \mathsf{t}_{d_V}$ that maximizes MI $I(\mathsf{c};\mathsf{t}_{d_V})$ is determined as the solution of the optimization problem $(\mathsf{c} \to \mathsf{a}_{d_V} \to g_{d_V}(\mathsf{a}_{d_V}) = \mathsf{t}_{d_V})$

$$g^{\mathrm{MI}}_{d_V} = \underset{g_{d_V}}{\operatorname{argmax}}\; I(\mathsf{c};\mathsf{t}_{d_V}) \text{ s.t. } |\mathcal{T}| = 2^{n_\mathrm{E}} \quad \text{for } \; d_V \in \mathcal{D}_V\,. \tag{16}$$

The parameter n_E defines the bit-width of the messages exchanged between VN and CN and controls the complexity of the message exchange. The optimization problem in (16) is the channel quantization problem for binary input (Section 2.3). The optimal solution is a deterministic input–output relation that can be stored as a d_V dimensional LUT with $2^{n_\mathrm{Q}+(d_V-1)n_\mathrm{E}}$ entries, e.g., for $d_V = 6$ and $n_\mathrm{E} = n_\mathrm{Q} = 4$, we have approximately 16.8 million entries. Furthermore, the communication performance can be increased by considering the degree distribution in the design of the node updates [13,26]. The gain in communication performance generally depends on the degree distribution and the message resolution n_E [13]. However, a comparison of the different design approaches in [13,26] is beyond the scope of this paper. The distribution of the VN-to-CN messages for the next iteration in (10) is

$$p_{d_V}(t|c) = \sum_{\boldsymbol{a}\in\mathcal{V}\,:\,g^{\mathrm{MI}}_{d_V}(\boldsymbol{a})=t} p(\boldsymbol{a}|c) \ \text{ for } \ d_V \in \mathcal{D}_V\,. \tag{17}$$

Again, the marginal distribution is determined by averaging over all possible VN degrees, i.e.,

$$p(t_j|c_j) = p(t|c) = \sum_{d_V\in\mathcal{D}_V} p_{d_V}(t|c)\lambda_{d_V}\,, \quad \text{with } j = 1,\ldots,d_{C,\max}\,. \tag{18}$$

In case of a regular LDPC code, there is only one possible degree for all VNs and CNs, i.e., the summation term in (14) and (18) vanishes but all other steps remain the same.

For the design of the MI maximizing Min-mLUT decoder, we start with an initial VN-to-CN distribution $p(t_j|c_j)$ and iterate over (10), (13)–(18) and declare convergence if $I(\mathsf{c};\mathsf{t})$ approaches the maximum value of one bit for binary input after I number of iterations.

3.3. Sequential LUT Design

For the sequential design approach sLUT, the node update is split into a sequence of degree two updates that are optimized independently to maximize MI. This approach serves as an approximation of the mLUT design described in Section 3.2 and reduces the number of possible memory locations within each update. In general, multidimensional optimization without decomposition conserves more MI compared to a design that decomposes the optimization problem into a sequence of two-dimensional updates [11,12] or more general nested tree decompositions [13].

4. Minimum-Integer Computation Decoder Design

The MI maximizing Min-mLUT decoder realizes the discrete VN updates by LUTs with $2^{n_\mathrm{Q}+(d_V-1)n_\mathrm{E}}$ entries leading to prohibitively large implementation complexity. Nevertheless, determining these multidimensional LUTs in the laboratory is feasible with sufficient computing resources. Thus, the idea is to search for the MI maximizing mLUTs but implement the corresponding discrete functions by relatively simple operations in order to avoid performance degradations. As visualized in Figure 6, the computational domain framework [14,16] replaces the VN update by an operation that is decomposed into

(i) mappings ϕ_V and ϕ of the n_E-bit CN-to-VN messages a_m and n_Q-bit channel information z into *node internal* n_R-bit signed integers with $n_\mathrm{R} \geq n_\mathrm{E}$ and $n_\mathrm{R} \geq n_\mathrm{Q}$, respectively;

(ii) execution of integer additions for n_R-bit signed integers;

(iii) threshold quantization Q_V to n_E bits determining the VN-to-CN message t.

For the MIC decoder design, we derive a criterion for sufficient internal node resolution n_R such that the mLUT mapping is replaced exactly. Note that the information maximizing mLUT is generated offline and is replaced by an integer function that replaces its functionality exactly or approximately during execution.

To keep the notation simple, we omit the dependency on the iteration index i and node degree in this section.

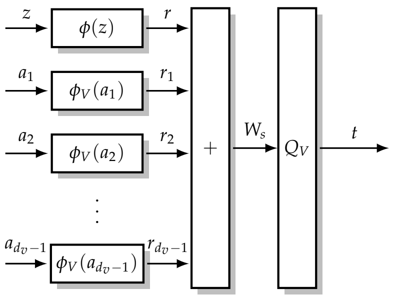


Figure 6. VN update for computational domain framework [14,16]. The n_Q-bit channel information $z \in \mathcal{Z}$ and the n_E-bit CN-to-VN messages $a_1, ..., a_{d_V-1} \in \mathcal{A}$ are transformed to n_R-bit signed integers. This transformation generally increases the required bit resolution for the representation, i.e., $n_\text{R} \geq n_\text{Q}$ and $n_\text{R} \geq n_\text{E}$. The internal signed integers are summed and quantized back into a n_E-bit VN-to-CN message $t \in \mathcal{T}$.

4.1. Equivalent LLR Quantizer

To motivate the integer calculation of the MIC approach, we review the connection between the equivalent LLR quantizer and the VN update of the BP algorithm. Analogous to the VN update of the BP algorithm in (5), the LLR of the combined message vector $\boldsymbol{a} \in \mathcal{V}$ equals the addition of the LLRs of the channel output z and of the individual messages a_m, i.e., for every possible combination $\boldsymbol{a} \in \mathcal{V}$, the LLR of the combined message is

$$L(\boldsymbol{a}) = \log\left(\frac{p(\boldsymbol{a}|c=0)}{p(\boldsymbol{a}|c=1)}\right) = L(z) + \sum_{m=1}^{d_V-1} L(a_m)\ . \tag{19}$$

The LLRs $L(a_m)$ of the individual messages are determined by (14) during DDE. As described in Section 2.3, the information maximizing quantizer for binary input separates the LLR $L(\boldsymbol{a})$ by using a $|\mathcal{T}|-1$ threshold quantizer $Q_\text{L} : \mathbb{R} \to \mathcal{T}$, i.e., the relation

$$t = g^\text{MI}(\boldsymbol{a}) = Q_\text{L}(L(\boldsymbol{a})) = Q_\text{L}\left(L(z) + \sum_{m=1}^{d_V-1} L(a_m)\right) \tag{20}$$

can be determined that achieves the same output as the information optimal mLUT in (16). However, to ensure that (20) produces the same output as the information optimal mLUT, calculations over real numbers are required. In the next subsection, we show that we can exploit (20) to find a calculation that requires only a finite resolution. We also provide a condition to limit the resolution that is required for exact calculation of the information optimal mLUT.

4.2. Computations over Integers

The VN update structure using the computational domain framework is visualized in Figure 6. As suggested by [14,16], a possible choice for the integer mappings $\phi_v(m)$ and $\phi_{ch}(z)$ is given by scaling and rounding the corresponding LLRs $L(m)$ and $L(z)$, respectively. In addition to [14,16], we provide further insights on the optimal choice of the scaling factor based on the relation between the VN update of the BP algorithm and the MI maximizing quantizer design. More precisely, based on the established relation in (20), we define an integer mapping for the channel information z and the CN-to-VN messages a_m in order to replace the computations over real numbers by computations over signed integers (With $\lfloor\cdot\rceil$ as round to nearest integer (away from 0 if fraction part is .5))

$$g^\text{MIC}(\boldsymbol{a}) = Q_V(W_s(\boldsymbol{a})) = Q_V\left(\underbrace{\lfloor sL(z)\rceil}_{r=\phi(z)\in\mathcal{R}} + \sum_{m=1}^{d_V-1} \underbrace{\lfloor sL(a_m)\rceil}_{r_m=\phi_\text{V}(a_m)\in\mathcal{R}_\text{V}}\right) = Q_V\left(r + \sum_{m=1}^{d_V-1} r_m\right). \tag{21}$$

Compared to (20), the LLRs $L(z)$ and $L(a_m)$ have been multiplied by a non-negative scaling factor s and quantized to the next n_{R}-bit signed integer r and r_m, respectively. Subsequently, the sum of integers is limited again to n_{E} bits by threshold quantizer Q_V. We can interpret the scaling and rounding operation also directly as a mapping of signed integer messages z and a_m to n_{R}-bit signed integer messages

$$r = \phi(z) \in \mathcal{R} \quad \text{and} \quad r_m = \phi_{\mathrm{V}}(a_m) \in \mathcal{R}_{\mathrm{V}} \tag{22}$$

that requires n_{R} bits for the representation, depending on the scaling factor s.

In the following, we show that we can always find a threshold quantizer $Q_V : \mathcal{W} \to \mathcal{T}$ that maps the summation $W_s(\boldsymbol{a})$ into a VN-to-CN message $t \in \mathcal{T}$ that is identical to the VN-to-CN message of the information optimal VN update in (20), i.e., $t = g^{\mathrm{MIC}}(\boldsymbol{a}) = g^{\mathrm{MI}}(\boldsymbol{a})$. First, we consider the set of messages $\mathcal{A}_t = \{\boldsymbol{a} \in \mathcal{V} : g^{\mathrm{MI}}(\boldsymbol{a}) = t \in \mathcal{T}\}$ that are mapped into a specific output t via the information maximizing VN update $g^{\mathrm{MI}}(\boldsymbol{a})$ in (16). Thus, we can identify a corresponding set of integers $\mathcal{W}_t = \{W_s(\boldsymbol{a}) \in \mathcal{W} : \boldsymbol{a} \in \mathcal{A}_t\}$. By varying the scaling factor s, we can always find a scaling value $s^\star \leq \frac{d_V}{\Delta_{\min}}$ such that the sets of integer values $\mathcal{W}_t$ for all $t \in \mathcal{T}$ are *non-overlapping intervals*, i.e.,

$$[D_{t'}, E_{t'}] \cap [D_{t''}, E_{t''}] = \emptyset \quad \forall t'', t' \in \mathcal{T}\ , \tag{23}$$

with $D_t = \min \mathcal{W}_t$ and $E_t = \max \mathcal{W}_t$. Condition (23) ensures that any two different clusters t' and t'' can be separated by a simple threshold operation. The value $\Delta_{\min}$ is the minimum separation between the LLRs $L(\boldsymbol{a})$ of the elements of any two neighbouring clusters in (20) and is always larger than zero since Q_{L} is a threshold quantizer. If we consider a scaled version of the LLRs $sL(\boldsymbol{a})$ with any real valued scaling factor $s > 1$, we can always find a threshold quantizers $Q_{L,s}$ that achieves the same output as the information optimal mLUT. Scaling the LLRs $L(\boldsymbol{a})$ by a factor of $\frac{d_V}{\Delta_{\min}}$ ensures that the minimum separation between any two neighbouring clusters is d_V. Since the influence of the rounding operation can be bounded by $-\frac{d_V}{2} \leq W_s(\boldsymbol{a}) - sL(\boldsymbol{a}) < \frac{d_V}{2}$, scaling with a factor of at least $\frac{d_V}{\Delta_{\min}}$ ensures that any two neighbouring clusters $\mathcal{W}_t$ and $\mathcal{W}_{t+1}$ are separated by at least one integer and, thus, condition (23) is satisfied. Hence, we can always find a corresponding integer function $g^{\mathrm{MIC}}(\boldsymbol{a})$ in (21) that generates exactly the same output as $g^{\mathrm{MI}}(\boldsymbol{a})$ in (20).

Furthermore, an approximate integer calculation is found if the integer valued range of ϕ and ϕ_{V} are limited to n_{R}-bits

$$\max_z |\phi(z)| < 2^{n_{\mathrm{R}}} < 2^{n_{\mathrm{R}}^\star}\ , \quad \max_a |\phi_{\mathrm{V}}(a)| < 2^{n_{\mathrm{R}}} < 2^{n_{\mathrm{R}}^\star} \tag{24}$$

where $n_{\mathrm{R}}^\star = \lceil \log_2(\lfloor s^\star L_{\max} \rceil) \rceil + 1$ is the bit resolution that is required for exact representation if the largest magnitude of the individual LLRs in (20) is $L_{\max}$. If condition (23) is not fulfilled, we select the output cluster that maximizes MI. If (24) is satisfied, the required bit resolution of the summation $W_s(\boldsymbol{a})$ in (21) is limited by

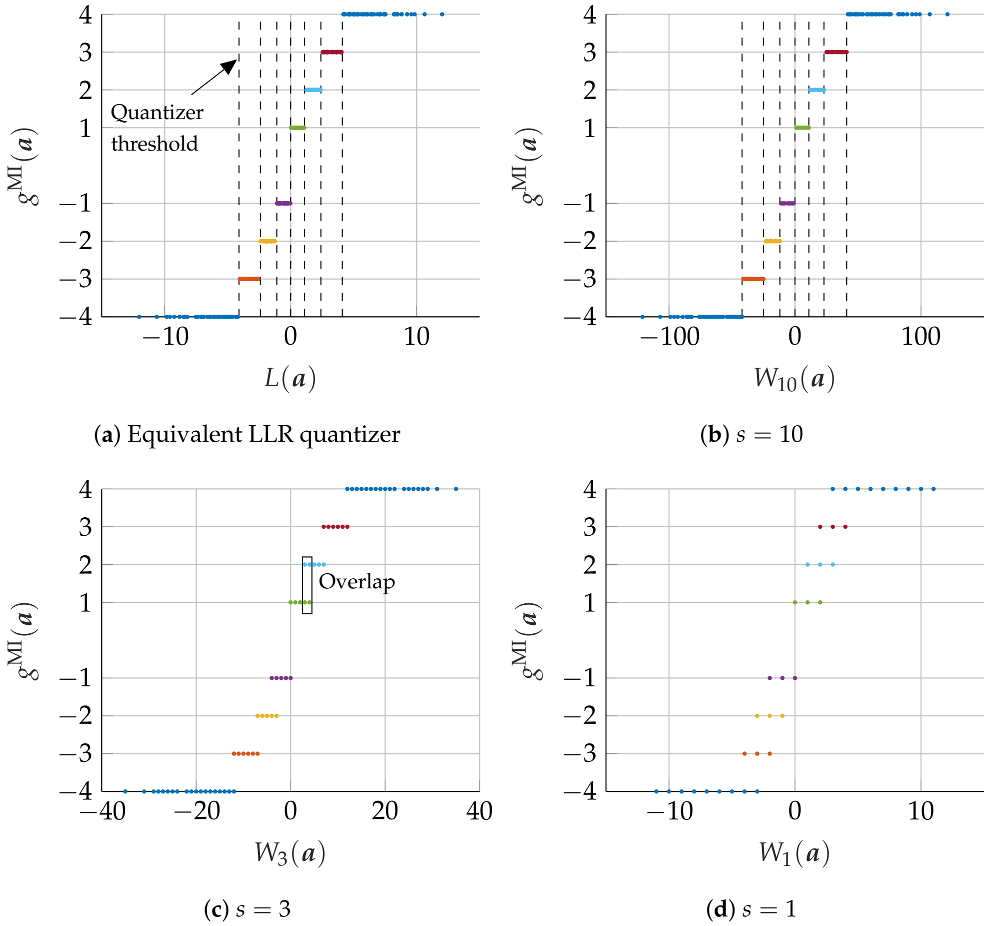
$$n_{\mathrm{W}} = \lceil \log_2 \Big(d_V (2^{(n_{\mathrm{R}}-1)} - 1) \Big) \rceil + 1\ . \tag{25}$$

To consider the influence of this new mapping in the design of subsequent iterations, we also update the VN-to-CN distribution in (17).

We note that the MIC design approach can also be applied for the design of CN operations and can also be used to generate exact or approximate representations of nested tree decompositions similar to the sLUT method. However, the corresponding investigations are beyond the scope of this paper.

Illustrative Example for MIC Calculations

To illustrate the proposed MIC approach, we consider the design of a VN node update for a (d_V=3, d_C=6) regular LDPC codes at iteration $i = 1$ with design $E_b/N_0 = 2.5$ dB. Figure 7a shows the equivalent LLR quantizer (20) with $2^{n_{\mathrm{E}}}$ *non-overlapping* clusters on the real number line, i.e., $\mathcal{T} = \{-4, ..., -1, 1, ..., 4\}$. E.g. all LLRs $L(\boldsymbol{a})$ between 0 and 1.1 are mapped into cluster $t = 1$. The threshold values are shown by dashed lines in Figure 7a. Additionally, Figure 7b–d show the output of the integer addition in (21) on the x-axis and the output clusters of the optimal mLUT on the y-axis if the scaling factor is set to $s = 10$, $s = 3$, and $s = 1$, respectively.



(**a**) Equivalent LLR quantizer

(**b**) $s = 10$

(**c**) $s = 3$

(**d**) $s = 1$

Figure 7. Visualization of the relationship between the result of the calculation in the computational domain and the assignment to mutual information maximizing mLUT mapping. Subfigure (**a**) shows the addition the real valued LLRs of (20) on the x-axis and the mutual information maximizing mLUT assignment of (16) on the y-axis. In Subfigure (**b**)–(**d**), the values on the x-axis are replaced by the corresponding integer additions of (21) for different scaling factors $s \in \{1, 3, 10\}$.

In the case of $s = 10$, all output clusters are separated by using seven integer thresholds, which are indicated by dashed lines in Figure 7b. In this case, the integer computation fully replaces the original mLUT functionality by using only signed integers of low-range. To clarify the example, the numeric values of the corresponding LLRs and integer mappings of (19) and (21) for $s = 10$ are shown in Table 1. For example, the quantized receive message $z = 2$ corresponds to an LLR of $L(z) = 1.56$ leading to the n_{R}-bit signed integer message $r = \phi(z) = \lfloor 15.6 \rceil = 16$. After summation of r and r_m, all results $12 \leq W_{10}(\boldsymbol{a}) \leq 23$ are again mapped back to the n_{E} message $t = 2$.

Table 1. Numeric values of integer based VN update g_{10}^{MIC} with scaling parameter $s = 10$.

Cluster index z, a, t	± 4	± 3	± 2	± 1
Channel LLR $L(z)$	± 5.07	± 2.90	± 1.56	± 0.49
Integer mapping $\phi(z)$	± 51	± 29	± 16	± 5
Message LLR $L(a)$	± 3.46	± 2.08	± 1.02	± 0.25
Integer mapping $\phi_V(a)$	± 35	± 21	± 10	± 3
Interval $\mathcal{W}_t$	$\pm[121, 42]$	$\pm[41, 25]$	$\pm[23, 12]$	$\pm[11, 1]$

For $s = 3$ and $s = 1$, the integer range is further reduced, but the original mLUT functionality cannot be represented exactly since some integer additions are mapped to more than one output cluster of the original mLUT (e.g., some values of $W_s(\boldsymbol{a})$ are mapped into cluster $t = 1$ and $t = 2$ as highlighted in Figure 7c). If some values of $W_s(\boldsymbol{a})$ are assigned to more than one cluster of the information maximizing mLUT mapping, a merging of these values into a single cluster is required. This merging generally leads to an inevitable loss of information. In order to find a corresponding threshold quantizer for this case, we select the output cluster that minimizes the information loss under the condition that (23) is fulfilled.

4.3. FER Results

In this section, we discuss the communication performance of the proposed MIC decoder for an irregular LDPC code from the 802.11n standard [27] of length $N = 648$ with rate $R = 0.75$ and edge degree distributions $\lambda(\xi) = 0.2083\xi^1 + 0.3333\xi^2 + 0.25\xi^3 + 0.2083\xi^5$ and $\rho(\xi) = \frac{1}{3}\xi^{13} + \frac{2}{3}\xi^{15}$. The realization of the MIC decoder is characterized by three quantization parameters and specified by MIC($n_\text{E}, n_\text{Q}, n_\text{R}$). In contrast, the Min-mLUT decoder with label based minimum operation as CN update has only two parameters and is denominated by Min-mLUT(n_E, n_Q). Figure 8 shows the Frame Error Rate (FER) performance of Min-mLUT and MIC for $n_\text{E} = n_\text{Q} = 3$ and $I = 10$ iterations, but varying resolution of internal messages $n_\text{R} \in \{4, 5, 6, 7, 11\}$ for MIC.

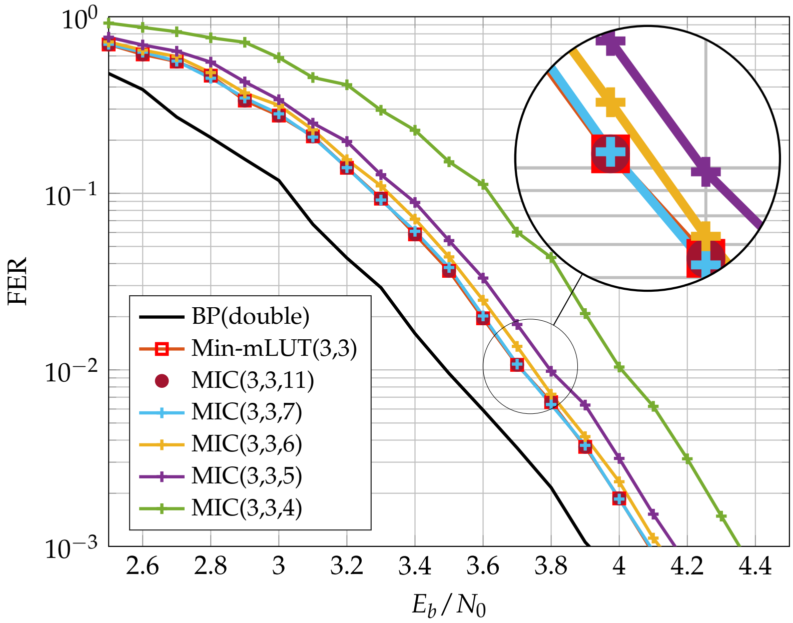


Figure 8. FER performance of $n_\text{E} = n_\text{Q} = 3$ bit Min-mLUT and MIC decoders using different internal message resolutions n_R for VN update.

The BP decoder with double precision serves as our benchmark simulation. The Min-mLUT decoder with $n_{\mathrm{E}} = n_{\mathrm{Q}} = 3$ bit quantization for the message exchange and channel information results in a minor performance degeneration of only 0.2 dB at a FER of 10^{-3} w.r.t. the benchmark simulation. In comparison, the proposed MIC decoder that replaces the VN update of the Min-mLUT decoder by using the computational domain framework with internal messages of size $n_{\mathrm{R}} = 4$ results in a loss of 0.25 dB compared to the Min-mLUT decoder. The performance gain of the MIC decoder by using $n_{\mathrm{R}} = 5$ compared to $n_{\mathrm{R}} = 4$ is around 0.1 dB. The MIC decoder with $n_{\mathrm{R}} = 7$ has basically identical FER performance compared to the Min-mLUT decoder. If $n_{\mathrm{R}} = 11$, the MIC decoder represents the mLUT functionality exactly by meeting the criterion (23), but the gain in communication performance compared to the MIC decoder with $n_{\mathrm{R}} = 7$ is negligible. Additionally, MIC decoding does not require LUTs with up to 262k entries for each iteration.

5. Finite Alphabet Message Passing (FA-MP) Decoder Implementation

In this section, we investigate the implementation complexity of different LUT-based FA-MP decoders in terms of area, throughput, latency, power, area efficiency, and energy efficiency and compare them with a state-of-the-art Normalized Min-Sum (NMS) decoder. As already stated, we focus on unrolled full parallel (FP) decoder architectures that enable throughput towards 1 Tb/s. The architecture template is shown in Figure 9. Input to the decoder are compressed messages z from the channel quantizer. The decoder uses two-phase decoding. Hence, each iteration consists of two stages: one stage comprises M Check Node Functional Units (CFUs) and the second stage N Variable Node Functional Units (VFUs). The stages are connected by hardwired routing networks, which implement the edges of the Tanner graph. Since the decoding iterations are unrolled, the decoder consists of $2 \cdot I$ stages. Deep pipelining is applied to increase the throughput. For more details on this architecture, the reader is referred to [5].

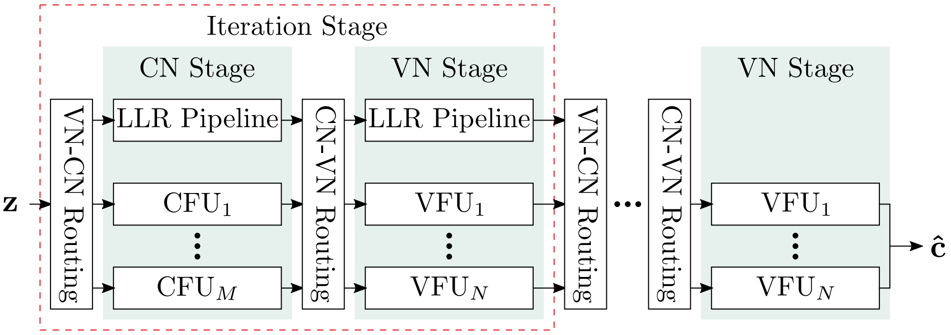


Figure 9. Unrolled full-parallel decoding architecture.

In FP decoders that use the NMS algorithm, node operations are implemented as additions and minimum searches on uniformly quantized messages [5]. In contrast, node functionality in Finite Alphabet (FA) decoders is implemented as LUTs. Implementing a single LUT as memory is impractical in Application-Specific Integrated Circuit (ASIC) technologies since the area and power overhead would be too large. Hence, a single LUT is transformed into n_E Boolean functions $b : B^{inp} \rightarrow B$ with inp being the number of inputs of the LUT, which is the node degree multiplied by n_{E}. b can consist of up to 2^{inp} product terms if b is represented in sum-of-product form. State-of-the-art logic synthesis tools try to minimize b such that it can be mapped onto a minimum number of gates. Despite this optimization, the resulting logic can be very large for higher node degrees and/or n_{E}, making this approach unsuitable for efficient FP decoder implementation. It was shown in [7] that the mLUT can be decomposed into a set of two-input sLUTs arranged in a tree structure, which largely reduces the resulting logic at the cost of a small degradation in error correction performance. To compare these approaches with our new decoder, we implemented four different types of FP decoders:

- NMS decoder with extrinsic message scaling factor of 0.75;
- Two LUT-based decoders: in these decoders, we implemented the VN operation by LUTs and the CN operations by a minimum search on the quantized messages. The latter corresponds to the CN Processor implementation of [7]. The LUTs are implemented either as a single LUT (mLUT), or as a tree of two-input LUTs (sLUT);
- Our new MIC decoder in which the VN is replaced by the new update algorithms, presented in the previous section.

For MIC and LUT based decoders, we investigated message quantization $n_{\mathrm{E}} = 3$ and $n_{\mathrm{E}} = 4$. The reference is an NMS decoder with $n_{\mathrm{E}} = 4$ and $n_{\mathrm{E}} = 5$, respectively. For all decoders, the channel and message quantization were set to be identical, i.e., $n_{\mathrm{E}} = n_{\mathrm{Q}}$. We used a different code for our implementation investigation than in the previous sections. This code has a larger block size, which implies increased implementation complexity. The code is a $(816, 406)$ regular LDPC code with $d_V = 3$ and $d_C = 6$ and the number of decoding iterations is $I = 8$.

We applied a Synopsys Design Compiler and IC Compiler II for implementation in a 28 nm Fully-Depleted Silicon-on-Insulator (FD-SOI) technology under worst-case Process, Voltage and Temperature (PVT) conditions (125 °C, 0.9 V for timing, 1.0 V for power). A process with eight metal layers was chosen. Metal layers 1 to 6 are used for routing, with metals 1 and 2 mainly intended for standard cells. The metal layers 7 and 8 are only used for power supply. Power numbers were calculated with back-annotated wiring data and input data for a FER of 10^{-4}. All designs were optimized for high throughput with a target frequency of 1 GHz during synthesis and back-end. To assess the routing congestion, we fixed the utilization to 70 % for all designs as a constraint. The utilization specifies the ratio between logic cell area and total area (=logic cell area plus routing area). Thus, by fixing this parameter, all designs have the same routing area available in relation to their logic cell area.

5.1. FER Performance of Implemented FA-MP Decoders

Figures 10 and 11 show the FER performance for the different decoders. We compare the NMS decoder with the MIC decoder and the two LUT-based decoders. The LUTs of the FA-MP decoders are elaborated to a design Signal-to-Noise-Ratio (SNR) optimized at an FER of 10^{-4}. It should be noted that this may result in an error floor behavior below the target FER. This phenomenon can be mitigated by selecting a larger design SNR at the cost of decreased performance in the waterfall region [13]. For comparison, we also added the BP performance with double precision floating point number representation.

In the previous section, we showed that, for the $(648, 486)$ code, the MIC decoder achieves the same error correction performance as the Min-mLUT decoder for $n_{\mathrm{R}} = 7$. A similar observation was made for the $(816, 406)$ code considered here. In our implementation comparison, we reduced n_{R} such that the MIC's FER stays below that of the NMS at the target FER of 10^{-4}. In this way, we obtained an $n_{\mathrm{R}} = 5$, which yields a small degradation in the MIC FER compared to the Min-sLUT and Min-mLUT decoders, but outperforms the NMS decoder. We observe that the MIC and Min-mLUT decoders with one bit smaller message quantization n_{E} have better error correction capability than the NMS decoder at the target FER. In addition, due to the low message quantization and the resulting low dynamic range, the NMS runs into an error floor below FER 10^{-4}.

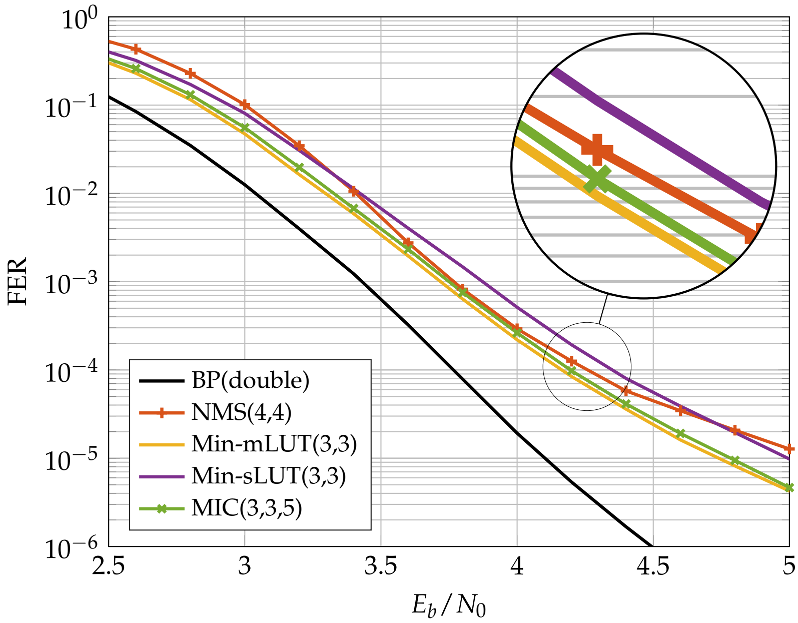


Figure 10. Communication performance of $n_{\mathrm{E}} = 3$ bit FA-MP decoders.

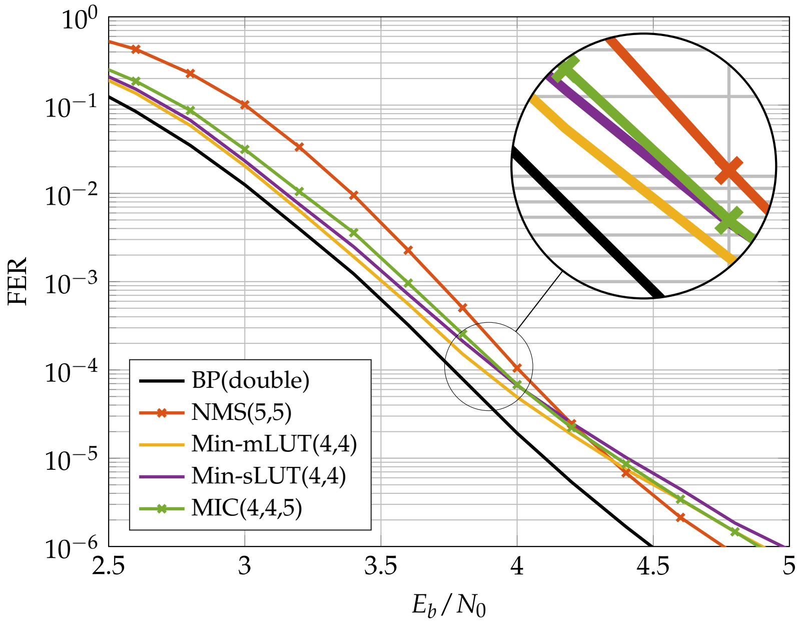


Figure 11. Communication performance of $n_{\mathrm{E}} = 4$ bit FA-MP decoders.

5.2. FD-SOI Implementation Results

Table 2 shows the implementation results for MIC(3,3,5), Min-mLUT(3,3), Min-sLUT(3,3) and NMS(4,4) decoders, whereas Table 3 shows the implementation results for MIC(4,4,5), Min-mLUT(4,4), Min-sLUT(4,4) and NMS(5,5) decoders. As already stated, we fixed the target frequency to 1 GHz and the utilization to 70% for all decoders. Maximum achievable frequency f, final utilization, area A and power consumption P were extracted from the final layout data. From these data, we can derive the important implementation metrics: throughput, latency, area efficiency and energy efficiency. Since the decoders are pipelined, the coded decoder throughput T is $f \cdot N$. The latency is $1/f \cdot 26$ (each iteration consists of three pipeline stages, decoder input and output are also buffered, yielding $8 \cdot 3 + 2 = 26$ pipeline stages in total). The area efficiency is defined as T/A and the energy efficiency as P/T.

The Min-mLUT decoder has the largest area, the worst area efficiency, and the worst energy efficiency. We see an improvement in these metrics for the Min-sLUT at the cost of a slightly decreased error correction performance. The difference in the implementation metrics largely increases when $n_\mathrm{E} = 3$ changes to $n_\mathrm{E} = 4$. The area increases by a factor of 10 for the Min-mLUT(4,4), but only by a factor of 2.7 for the Min-sLUT(4,4) decoder. Moreover, we had to reduce the utilization to 50 % to achieve a routing convergence for the Min-mLUT(4,4) decoder. The large area increase is explainable with the increase of the LUT sizes from 512 to 4096 entries per LUT when increasing n_E from 3 to 4. Moreover, the frequency largely breaks down, yielding a very low area efficiency and energy efficiency. The Min-sLUT decoders scale better with increasing n_E. Both Min-sLUT decoders outperform the corresponding NMS decoders in throughput and efficiency metrics.

The MIC decoder has the best implementation metric numbers in all cases. It outperforms all other decoders in throughput, area, area efficiency and energy efficiency while having the same or even slightly improved error correction performance compared to the other decoders. It can also be seen that the MIC decoder has a lower routing complexity compared to the Min-sLUT and the NMS decoder. We observe a large drop in the frequency from 595 MHz down to 183 MHz (70 % decrease) when comparing NMS(4,4) with NMS(5,5) under the utilization constraint of 70 %. The large drop in the frequency is explainable with the increased routing complexity for the given routing area constraint that yields longer wires and corresponding delays. This problem is less severe for the Min-sLUT, where the frequency drops from 670 MHz to 492 MHz (27 % decrease). The MIC achieves the highest frequency for all cases and drops from 775 MHz to 633 MHz (18% decrease), only. This shows that the MIC scales much better with increasing n_E.

It should be noted that the CFU implementation is identical for the MIC, Min-mLUT and Min-sLUT decoders. Compared to the corresponding NMS, the CFU implementation is less complex [19] due to: (i) a 1 bit smaller message quantization, (ii) the omission of the scaling unit, and (iii) the omission of the sign-magnitude to two's complement conversion. Hence, the CFU complexity of the FA-MP is always lower than that of the NMS independent of the respective CN degree. Moreover, in contrast to the NMS decoder, the messages from the CFUs to the VFUs are transmitted in sign-magnitude representation via the routing network which reduces the toggling rate and thus the average power consumption.

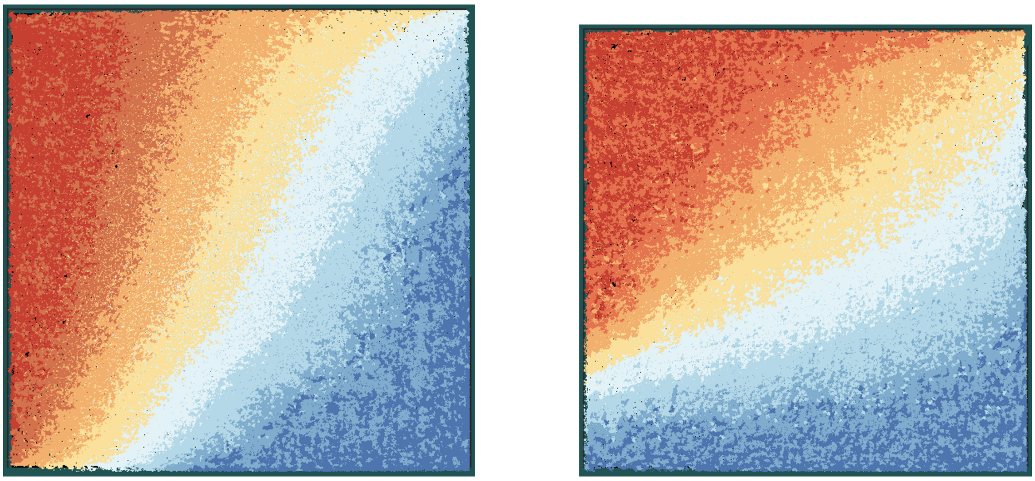
Table 2. Post-layout results of FA-MP decoders with $n_\mathrm{E} = n_\mathrm{Q} = 3$, $n_\mathrm{R} = 5$.

	MIC	**Min-mLUT**	**Min-sLUT**	**NMS**
n_E, n_Q	3	3	3	4
E_b/N_0 @ FER 10^{-4} [dB]	4.20	4.16	4.35	4.26
Utilization [%]	70	68	71	71
Frequency [MHz]	775	662	670	595
Coded Throughput [Gb/s]	633	540	547	486
Area [mm^2]	2.73	4.23	2.86	3.04
Area Efficiency [Gb/s/mm^2]	231.6	128	190	159.7
Latency [ns]	33.5	39.3	35.8	43.7
Power [W]	4.49	5.07	4.38	4.39
Energy Efficiency [pJ/bit]	7.10	9.4	8.0	9.0

Table 3. Post-layout results of FA-MP decoders with $n_\mathrm{E} = n_\mathrm{Q} = 4$, $n_\mathrm{R} = 5$.

	MIC	**Min-mLUT**	**Min-sLUT**	**NMS**
n_E, n_Q	4	4	4	5
E_b/N_0 @ FER 10^{-4} [dB]	3.94	3.87	3.93	4.01
Utilization [%]	69	49	66	69
Frequency [MHz]	633	267	492	183
Coded Throughput [Gb/s]	516	218	401	149
Area [mm^2]	3.66	40.51	7.82	3.99
Area Efficiency [Gb/s/mm^2]	141.1	5.4	51.3	37.4
Latency [ns]	41.1	97.2	48.0	142.0
Power [W]	5.61	11.85	8.68	2.25
Energy Efficiency [pJ/bit]	10.9	54.3	21.6	15.1

Figure 12 shows the layout of the MIC and the NMS decoder in the same scale. Each color represents one iteration stage, which is composed of CFUs, VFUs, and the routing between the nodes (see also Figure 9). When comparing the same iteration stages (same color) of the two decoders, we can observe that the iteration stages in the MIC decoder are smaller than the corresponding iteration stages in the NMS decoder, although the frequency of the MIC decoder is more than three times higher compared to the NMS decoder. This shows once again that the MIC has a lower implementation complexity, especially from a routing perspective.



(**a**) NMS(5,5), (3.99 mm^2) (**b**) MIC(4,4,5), (3.66 mm^2)

Figure 12. Layout of decoders in the same scale; each color indicates one iteration stage from dark red (first iteration) to dark blue (eighth iteration).

Our analysis shows that the new MIC approach largely improves the implementation efficiency and exhibits better scaling compared to the state-of-the-art sLUT and NMS implementations of FP decoder architectures. This enables the processing of larger block sizes, which is mainly due to the reduced routing complexity. Larger block sizes improve the error correction capability and further increase the throughput of FP architectures.

6. Conclusions

This paper provides a detailed investigation of the Minimum-Integer Computation (MIC) decoder for regular and irregular Low-Density Parity Check (LDPC) codes. The MIC decoder utilizes the computational domain framework to realize Variable Node (VN) updates by an equivalent low-range signed integer computation and Check Node (CN) updates by a minimum search. For the VN update, we provide further insights for the design of an Mutual Information (MI) maximizing signed integer computation. To discuss implementation issues on FA-MP decoding architectures, we exemplified this on different LUT-based decoder designs. Furthermore, we compared MIC to state-of-the-art Normalized Min-Sum (NMS) decoder implementations to show the improvement in area efficiency and energy efficiency.

Author Contributions: Conceptualization, T.M., O.G., M.H., D.W., A.D. and N.W.; Funding acquisition, A.D., D.W. and N.W.; Investigation, T.M., D.W., O.G. and M.H.; Software, T.M., O.G. and M.H.; Validation, T.M., O.G., M.H., D.W. and N.W.; Visualization, T.M.; Writing—original draft, T.M., O.G., M.H., D.W. and N.W.; Writing—review and editing, T.M., O.G., M.H., D.W., A.D. and N.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the German Ministry of Education and Research (BMBF) projects "FunKI" (grants 16KIS1180K and 16KIS1185) and "Open6GHub" (grants 16KISK016 and 16KISK004).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Saad, W.; Bennis, M.; Chen, M. A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems. *IEEE Netw.* **2020**, *34*, 134–142. https://doi.org/10.1109/MNET.001.1900287.
2. Kestel, C.; Herrmann, M.; Wehn, N. When Channel Coding Hits the Implementation Wall. In Proceedings of the IEEE 10th International Symposium on Turbo Codes Iterative Information (ISTC 2018), Hong Kong, China, 3–7 December 2018. https://doi.org/10.1109/ISTC.2018.8625324.
3. Gallager, R. Low-Density Parity-Check Codes. *IRE Trans. Inf. Theory* **1962**, *8*, 21–28. https://doi.org/10.1109/TIT.1962.1057683.
4. MacKay, D. Good error-correcting codes based on very sparse matrices. *IEEE Trans. Inf. Theory* **1999**, *45*, 399–431. https://doi.org/10.1109/18.748992.
5. Schläfer, P.; Wehn, N.; Alles, M.; Lehnigk-Emden, T. A New Dimension of Parallelism in Ultra High Throughput LDPC Decoding. In Proceedings of the IEEE Workshop on Signal Processing Systems (SIPS 2013), Taipei, Taiwan , 16–18 October 2013; pp. 153–158. https://doi.org/10.1109/SiPS.2013.6674497.
6. IEEE. International Roadmap for Devices and Systems, 2021 Update, More Moore. 2021. Available online: https://irds.ieee.org/images/files/pdf/2021/2021IRDS_MM.pdf (accessed on 8 August 2022)
7. Balatsoukas-Stimming, A.; Meidlinger, M.; Ghanaatian, R.; Matz, G.; Burg, A. A Fully-Unrolled LDPC Decoder based on Quantized Message Passing. In Proceedings of the IEEE Workshop on Signal Processing Systems (SIPS 2015), Hangzhou, China, 14–16 October 2015. https://doi.org/10.1109/SiPS.2015.7345024.
8. Ghanaatian, R.; Balatsoukas-Stimming, A.; Müller, T.C.; Meidlinger, M.; Matz, G.; Teman, A.; Burg, A. A 588-Gb/s LDPC Decoder Based on Finite-Alphabet Message Passing. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2018**, *26*, 329–340. https://doi.org/10.1109/TVLSI.2017.2766925.
9. Lee, J.S.; Thorpe, J. Memory-efficient decoding of LDPC codes. In Proceedings of the International Symposium on Information Theory (ISIT 2005), Adelaide, SA, Australia, 4–9 September 2005; pp. 459–463. https://doi.org/10.1109/ISIT.2005.1523376.
10. Chen, J.; Fossorier, M. Density Evolution for Two Improved BP-Based Decoding Algorithms of LDPC Codes. *IEEE Commun. Lett.* **2002**, *6*, 208–210. https://doi.org/10.1109/4234.1001666.
11. Romero, F.J.C.; Kurkoski, B.M. LDPC Decoding Mappings that Maximize Mutual Information. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 2391–2401. https://doi.org/10.1109/JSAC.2016.2603708.
12. Lewandowsky, J.; Bauch, G. Information-Optimum LDPC Decoders based on the Information Bottleneck Method. *IEEE Access* **2018**, *6*, 4054–4071. https://doi.org/10.1109/ACCESS.2018.2797694.
13. Meidlinger, M.; Matz, G.; Burg, A. Design and Decoding of Irregular LDPC Codes Based on Discrete Message Passing. *IEEE Trans. Commun.* **2020**, *68*, 1329–1343. https://doi.org/10.1109/TCOMM.2019.2944159.
14. Kang, P.; Cai, K.; He, X.; Li, S.; Yuan, J. Generalized Mutual Information-Maximizing Quantized Decoding of LDPC Codes With Layered Scheduling. *IEEE Trans. Veh. Technol.* **2022**, *71*, 7258–7273. https://doi.org/10.1109/TVT.2022.3162579.
15. He, X.; Cai, K.; Mei, Z. On Mutual Information-Maximizing Quantized Belief Propagation Decoding of LDPC Codes. In Proceedings of the IEEE Global Communications Conference (GLOBECOM 2019), Waikoloa, HI, USA, 9–13 December 2019. https://doi.org/10.1109/GLOBECOM38437.2019.9013335.
16. He, X.; Cai, K.; Mei, Z. Mutual Information-Maximizing Quantized Belief Propagation Decoding of LDPC Codes. *arXiv* **2019**, arXiv:1904.06666. Available online: https://arxiv.org/abs/1904.06666 (accessed on 7 October 2022).

17. Wang, L.; Wesel, R.D.; Stark, M.; Bauch, G. A Reconstruction-Computation-Quantization (RCQ) Approach to Node Operations in LDPC Decoding. In Proceedings of the IEEE Global Communications Conference (GLOBECOM 2020), Taipei, Taiwan, 7–11 December 2020. https://doi.org/10.1109/GLOBECOM42002.2020.9348139.
18. Wang, L.; Terrill, C.; Stark, M.; Li, Z.; Chen, S.; Hulse, C.; Kuo, C.; Wesel, R.D.; Bauch, G.; Pitchumani, R. Reconstruction-Computation-Quantization (RCQ): A Paradigm for Low Bit Width LDPC Decoding. *IEEE Trans. Commun.* **2022**, *70*, 2213–2226. https://doi.org/10.1109/TCOMM.2022.3149913.
19. Monsees, T.; Wübben, D.; Dekorsy, A.; Griebel, O.; Herrmann, M.; Wehn, N. Finite-Alphabet Message Passing using only Integer Operations for Highly Parallel LDPC Decoders. In Proceedings of the IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication (SPAWC 2022), Oulu, Finland, 4–6 July 2022. https://doi.org/10.1109/SPAWC51304.2022.9833953.
20. Tanner, R. A recursive approach to low complexity codes. *IEEE Trans. Inf. Theory* **1981**, *27*, 533–547. https://doi.org/10.1109/TIT.1981.1056404.
21. Kurkoski, B.M. On the Relationship Between the KL Means Algorithm and the Information Bottleneck Method. In Proceedings of the 11th International ITG Conference on Systems, Communications and Coding (SCC), Hamburg, Germany, 6–9 February 2017.
22. Tishby, N.; Pereira, F.C.; Bialek, W. The Information Bottleneck Method. In Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA, 22–24 September 1999; pp. 368–377.
23. Hassanpour, S.; Wübben, D.; Dekorsy, A. Overview and Investigation of Algorithms for the Information Bottleneck Method. In Proceedings of the 11th International Conference on Systems, Communications and Coding (SCC), Hamburg, Germany, 6–9 February 2017.
24. Kurkoski, B.M.; Yagi, H. Quantization of Binary-Input Discrete Memoryless Channels. *IEEE Trans. Inf. Theory* **2014**, *60*, 4544–4552. https://doi.org/10.1109/TIT.2014.2327016.
25. Burshtein, D.; Pietra, V.D.; Kanevsky, D.; Nadas, A. Minimum Impurity Partitions. *Ann. Stat.* **1992**, *20*, 1637–1646. https://doi.org/10.1214/aos/1176348789.
26. Stark, M.; Wang, L.; Bauch, G.; Wesel, R.D. Decoding Rate-Compatible 5G-LDPC Codes With Coarse Quantization Using the Information Bottleneck Method. *IEEE Open J. Commun. Soc.* **2020**, *1*, 646–660. https://doi.org/10.1109/OJCOMS.2020.2994048.
27. IEEE. IEEE Standard for Information Technology—Local and Metropolitan Area Networks—Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput. In *IEEE Std 802.11n-2009 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, and IEEE Std 802.11w-2009)*; IEEE: Piscatvey, NJ, USA, 2009; pp. 1–565. https://doi.org/10.1109/IEEESTD.2009.5307322.