Freezing-based Scheme for the Fixed-point Implementation of Min-sum LDPC Decoding

Mehrzad Malmirchegini
ECE Department of University of New Mexico,
Albuquerque, New Mexico, USA
Email: mehrzad@ece.unm.edu

Shadi Abu-Surra and Thomas Henige and Eran Pisek
Dallas Technology Lab, Samsung Electronics,
Richardson, Texas, USA
Email: {sasurra,thenige,episek}@sta.samsung.com

***Abstract*—In this paper, we statistically analyze the impact of the quantization and the fixed-point implementation on the performance of the min-sum LDPC layered decoding algorithm. In particular, we show how the growth of the LLR (Log of Likelihood Ratio) values under finite precision results in the saturation and performance degradation. We then propose a freezing-based min-sum decoding algorithm for the fixed-point implementation. We show that the proposed approach overcomes the saturation problems at high SNRs and improves the decoding performance drastically. Furthermore, we discuss an optimum uniform quantization scheme, which minimizes the quantization error of the channel LLR values.**

***Index Terms*—Belief propagation, Fixed-point implementation, LDPC codes, min-sum decoding algorithm.**

I. Introduction

Low Density Parity Check codes (LDPC) have recently received considerable attention in the error control coding field, due to the low complexity of their decoding algorithms [1]. Furthermore, this class of codes performs near the Shannon bound for long enough block lengths.

Among different families of decoding algorithms, the well-known belief propagation (BP) algorithm provides a good decoding performance [2]. However, BP algorithm requires large hardware complexity. A simplified variation of BP algorithm, called min-sum decoding algorithm, significantly reduces the hardware complexity at the cost of performance degradation. Recently, the normalized min-sum decoding algorithm has been preferred in many practical and finite precision applications since it provides acceptable decoding performance as compared to BP algorithm for regular codes [3]. However, for the most irregular LDPC codes, the normalized min-sum decoding algorithm performs poorly under finite precision implementation [4], [5]. Most of irregular LDPC codes have a large amount of low-degree variable nodes. These variable nodes require more number of iterations to converge as compared to high-degree variables nodes. Furthermore, finite precision decoding further decreases the convergence rate of the low-degree variable nodes due to the quantization effects. In [6], [7], authors propose different variations of min-sum algorithm to improve the decoding performance at the cost of slower convergence rate and higher hardware complexity. For instance, in [6], the authors proposed that the variable nodes can use a down-scaled intrinsic information iteratively to improve the reliability of extrinsic information at the variable nodes. The down-scaling in intrinsic values reduces the convergence speed. Furthermore, the down scaling factor needs to be optimized for a specific number of iterations. In this paper we propose a low-complex freezing-based min-sum decoding algorithm to overcome the saturation problem and improve the decoding performance for the fixed-point implementation.

The paper is organized as follows. In Section II, we introduce the protograph LDPC codes and provide the background of min-sum layered decoding algorithm. In Section III, we statistically analyze the impact of the quantization and the fixed-point implementation on the performance of the min-sum layered decoding algorithm. Furthermore, we discuss an optimum uniform quantization scheme, which minimizes the quantization error of the channel LLR values. In Section IV, we propose the freezing-based min-sum decoding algorithm. We show the simulation results in Section V. We then conclude the paper in Section VI.

II. LDPC codes and min-sum layered decoding

A low-density parity-check code is defined by a sparse $M' \times N'$ parity check matrix, where M' represents the number of parity checks and N' represents the number of codeword's bits. The parity check matrix $\mathbf{H}$ of an LDPC code can be illustrated by a Tanner graph, where $\mathcal{C}_{\mathbf{H}}$ and $\mathcal{V}_{\mathbf{H}}$ denote the set of check nodes and variable nodes respectively. LDPC codes are usually decoded using message passing algorithms. One important subclass of these algorithms is belief propagation (BP) algorithm. In this algorithm, the CNs and VNs pass their beliefs or probabilities over the corresponding Tanner graph. There exists several realization of BP algorithm. The most well-known realization is the Sum-Product (SP). However, this algorithm requires large amount of hardware complexity. On the other hand, min-sum algorithm is a suboptimal variation of SP, which provides less complexity in price of degradation in performance.

Consider the binary phase-shift keying (BPSK) modulation and additive white Gaussian noise (AWGN) channel. The reception corresponding to the jth bit can be represented as: $y_j = \sqrt{P} b_j + n_j$, where P denotes the transmitted power, $b_j \in \{-1, 1\}$ and n_j is a zero-mean Gaussian variable with variance of σ^2. We define the conditional Log-Likelihood Ratios (LLR) as follows: $L_j^{\text{pr}} = \log \frac{\text{prob}(b_j=0|y_j)}{\text{prob}(b_j=1|y_j)} = \frac{2}{\sigma^2} y_j$. If b_js are equiprobable, then we have the following distribution

for L_j^{pr}:

$$p(x) = \text{prob}(L_j^{\text{pr}} = x) = \frac{\sigma}{4\sqrt{2\pi}}\Big(e^{-\frac{1}{8}\sigma^2(x-\frac{2\sqrt{P}}{\sigma^2})^2} + e^{-\frac{1}{8}\sigma^2(x+\frac{2\sqrt{P}}{\sigma^2})^2}\Big) \quad (1)$$

The LLR values (L_j^{pr}) are the input to the channel decoder block. We next discuss min-sum layered decoding algorithm [1] over protograph codes and show how the LLR values are utilized in the decoding process.

A. Protograph LDPC codes and min-sum layered decoding algorithm

The structure-based family of LDPC codes has found considerable attentions as it simplifies the architecture of the decoder drastically. Among the members of this family, protograph-based LDPC codes have been frequently appeared in the industry [8]. A protograph is a relatively small Tanner graph, from which a larger graph can be built up by the following copy-and-permute procedure. The protograph is copied Z times, where Z denotes the lifting factor. The identical edges will be then permutated among the corresponding replicas. Let the $M \times N$ matrix $\mathbf{H}_p$ denote the adjacency matrix of the protograph. In this work, we only consider protographs with no parallel edges, and so the entries of $\mathbf{H}_p$ belong to $\{0, 1\}$. We also define $\mathbf{I}_{Z\times Z}^{(s)}$ as a $Z \times Z$ identity matrix, circularly shifted to the right by s positions. The adjacency matrix of the derived LDPC code, i.e. $\mathbf{H}$, can be characterized as follows: 1) Replace every "0" in $\mathbf{H}_p$ by $\mathbf{0}_{Z\times Z}$, where $\mathbf{0}_{Z\times Z}$ denotes the $Z\times Z$ zero matrix. 2) Replace every "1" in $\mathbf{H}_p$ by $\mathbf{I}_{Z\times Z}^{(s)}$, where the values of s are randomly chosen from $[0, \cdots, Z-1]$. Therefore, the size of matrix $\mathbf{H}$ is $MZ \times NZ$. In protograph-based LDPC codes, the Z replicas of each CN in the protograph form a layer. Hence, the CNs of the LDPC code can be represented by: $\mathcal{C}_{\mathbf{H}} = \bigcup_{l=1}^{M} \mathcal{C}^l$, where $\mathcal{C}^l = \{c_1^l, \cdots, c_Z^l\}$ denotes the set of check nodes in the lth layer. Note that each layer consists of a set of contention-free CNs (i.e., only one CN can access a given VN memory at a precise time), which can be processed in parallel without contention. In our design, each decoding iteration consists of M sub-iterations, corresponding to each layer. For each layer, all the check nodes update their values in parallel. This process will go on in serial for different layers until all the check nodes update their values. These types of decoding algorithms are mainly called "Layered Decoding" in LDPC literatures [9]–[11]. Let $\mathcal{N}_{c_i^l} = \{p | v_p \text{ is connected to } c_i^l\}$ denote the index sets of the neighbors of the check node c_i^l. The min-sum layered decoding algorithm can be then summarized as follows:

- Initialization: For $k = 0$, the check-to-variable messages, $\rho_{c_i^l \to v_j}(0)$, for $1 \le i \le Z$ and $1 \le l \le M$ are initialized to zero. Furthermore, $L_j^{\text{ps},0}(0) = L_j^{\text{pr}}$, for $1 \le j \le NZ$, where $L_j^{\text{ps},l}(k)$ denotes the posterior LLR value corresponding to the jth variable node at iteration k and layer l.
- At iteration k and layer l:
  - Each check node at lth layer receives the following messages from its VN neighbors:
  $$\rho_{v_j \to c_i^l}(k) = L_j^{\text{ps},l-1}(k) - \rho_{c_i^l \to v_j}(k-1) \quad (2)$$
  for $j \in \mathcal{N}_{c_i^l}$ and $1 \le i \le Z$. We also have $L_j^{\text{ps},0}(k) = L_j^{\text{ps},M}(k-1)$.
  - Each check node will then send the following value to its VN neighbors
  $$\rho_{c_i^l \to v_j}(k) = \frac{\varepsilon_{c_i^l, v_j}(k)}{\alpha} \min_{p \in \mathcal{N}_{c_i^l} \setminus \{j\}} |\rho_{v_p \to c_i^l}(k)| \quad (3)$$
  where $\varepsilon_{c_i^l, v_j}(k) = \prod_{p \in \mathcal{N}_{c_i^l} \setminus \{j\}} \text{sgn}\big(\rho_{v_p \to c_i^l}(k)\big)$ with sgn(.) denotes the sign operator. α is the scaling factor. Define $m_i^l(k) \triangleq \arg\min_{p \in \mathcal{N}_{c_i^l}} |\rho_{v_p \to c_i^l}(k)|$ and $n_i^l(k) \triangleq \arg\min_{p \in \mathcal{N}_{c_i^l} \setminus m_i^l(k)} |\rho_{v_p \to c_i^l}(k)|$. Therefore, Eq. 3 will be simplified as:
  $$\rho_{c_i^l \to v_j}(k) = \begin{cases} \frac{\varepsilon_{c_i^l, v_j}(k)}{\alpha} |\rho_{v_{m_i^l(k)} \to c_i^l}(k)|, j \ne m_i^l(k); \\ \frac{\varepsilon_{c_i^l, v_j}(k)}{\alpha} |\rho_{v_{n_i^l(k)} \to c_i^l}(k)|, j = m_i^l(k). \end{cases}$$
- The posterior LLR value is updated as:
  $$L_j^{\text{ps},l}(k) = \rho_{v_j \to c_i^l}(k) + \rho_{c_i^l \to v_j}(k). \quad (4)$$
- Hard decisions are then made based on the sign of the posterior LLR values. The syndrome of the codeword is then checked in order to detect the error.

In the next section, we investigate the impact of the quantization and the fixed-point implementation on the performance of the min-sum layered decoding algorithm.

III. Fixed-point effects

In order to characterize the trade-offs between hardware complexity and decoding performance, the effects of finite precision should be analyzed. The finite-precision implementation of LDPC decoding requires the following steps 1- Quantization of the channel LLR values 2- Fixed-point implementation of decoding algorithm. For the first step, we devise an optimum uniform quantizer (in Mean Square Error (MSE) sense) for the channel LLR values and characterize the underlying trade-offs. We then analyze the performance of min-sum decoding algorithm under finite precision constraint. Our analysis in this section will be accounted as the basis for the proposed algorithm in the subsequent section.

A. LLR quantization

In this part, we devise an optimum uniform quantizer (in MSE sense) for the channel LLR values. In Eq. 1, we showed the distribution of LLR for the BPSK modulation over AWGN channel. Let $Q^{q,\Delta}(x)$ denote the q-bit quantizer with the step size Δ, defined as follows:

$$Q^{q,\Delta}(x) \triangleq \begin{cases} (2^{q-1}-1)\Delta, & x \ge \frac{2^q-3}{2}\Delta; \\ \lfloor \frac{x}{\Delta} + \frac{1}{2} \rfloor \Delta, & -\frac{2^q-1}{2}\Delta < x < \frac{2^q-3}{d}\Delta; \\ -2^{q-1}\Delta, & x \le -\frac{2^q-1}{2}\Delta. \end{cases}$$

Fig. 1 shows mean square quantization error of LLR values, i.e $E\Big\{\big[L_j^{\text{pr}} - Q^{q,\Delta}\big(L_j^{\text{pr}}\big)\big]^2\Big\}$, as a function of quantizer step size (Δ). As can be seen, for a fixed number of bits, quantization with small values of step size introduces large amount of error due to the saturation. However, large values of step size will

[1] This algorithm is known as the normalized min-sum layered decoding in LDPC literatures, however, for simplicity we call it min-sum layered decoding in this paper.

not capture the required resolution for high-probable values of LLR, which again results in a significant error. Therefore, there exists some tradeoffs for the value of the step size. The following theorem characterizes the optimum uniform quantizer, which minimizes the mean square quantization error of LLR values.

Theorem 1. *Define* $f_{\mu,s}(x,y) \triangleq \frac{1}{\sqrt{2\pi}s}\left(e^{-\frac{(x-\mu)^2}{2s^2}} - e^{-\frac{(y-\mu)^2}{2s^2}}\right)$ *and* $g_{\mu,s}(x,y) \triangleq Q\left(\frac{x-\mu}{s}\right) - Q\left(\frac{y-\mu}{s}\right)$, *where* $Q(z) = \int_z^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2} dz$ *denotes the Q-function. Let* Δ_{opt} *denote the optimum value of* Δ, *which minimizes MSE of LLR quantization. We then have,* $\Delta_{opt} = \min_\Delta C^q_{P,\sigma}(\Delta)$, *where*

$$C^q_{P,\sigma}(\Delta) = \frac{4}{\sigma^4}(P+\sigma^2) + \sum_{i=-2^{q-1}}^{2^{q-1}-1} \frac{1}{2}\Delta^2 i^2 \Big(g_{\frac{2\sqrt{P}}{\sigma^2},\frac{2}{\sigma}}(\alpha_n^{q,\Delta}, \alpha_{n+1}^{q,\Delta}) + g_{-\frac{2\sqrt{P}}{\sigma^2},\frac{2}{\sigma}}(\alpha_n^{q,\Delta}, \alpha_{n+1}^{q,\Delta}) \Big) - \Delta i \Bigg[\frac{4}{\sigma^2} \Big(f_{\frac{2\sqrt{P}}{\sigma^2},\frac{2}{\sigma}}(\alpha_n^{q,\Delta}, \alpha_{n+1}^{q,\Delta}) + f_{-\frac{2\sqrt{P}}{\sigma^2},\frac{2}{\sigma}}(\alpha_n^{q,\Delta}, \alpha_{n+1}^{q,\Delta}) \Big) + \frac{2\sqrt{P}}{\sigma^2} \Big(g_{\frac{2\sqrt{P}}{\sigma^2},\frac{2}{\sigma}}(\alpha_n^{q,\Delta}, \alpha_{n+1}^{q,\Delta}) - g_{-\frac{2\sqrt{P}}{\sigma^2},\frac{2}{\sigma}}(\alpha_n^{q,\Delta}, \alpha_{n+1}^{q,\Delta}) \Big) \Bigg] \quad (5)$$

and $\alpha_n^{q,\Delta} = \begin{cases} -\infty, & n = -2^{q-1}; \\ \frac{2n-1}{2}\Delta, & -2^{q-1}+1 \le n \le 2^{q-1}-1; \\ \infty, & n = 2^{q-1}. \end{cases}$.

Proof: The optimum Δ, in mean square sense, can be characterized as follows: $\Delta_{\text{opt}} = \min_\Delta \int_{-\infty}^{\infty} \left[x - Q^{q,\Delta}(x)\right]^2 p(x)dx$. We then have,

$$\int_{-\infty}^{\infty} \left[x - Q^{q,\Delta}(x)\right]^2 p(x)dx = \sum_{i=-2^{q-1}}^{2^{q-1}-1} \int_{\alpha_i^{q,\Delta}}^{\alpha_{i+1}^{q,\Delta}} (x-\Delta i)^2 p(x)dx = \sum_{i=-2^{q-1}}^{2^{q-1}-1} \Bigg[\Delta^2 i^2 \times \int_{\alpha_i^{q,\Delta}}^{\alpha_{i+1}^{q,\Delta}} p(x)dx - 2\Delta i \int_{\alpha_i^{q,\Delta}}^{\alpha_{i+1}^{q,\Delta}} xp(x)dx + \int_{\alpha_i^{q,\Delta}}^{\alpha_{i+1}^{q,\Delta}} x^2 p(x)dx \Bigg]$$

$$= \sum_{i=-2^{q-1}}^{2^{q-1}-1} \Bigg[\frac{1}{2}\Delta^2 i^2 \Big(g_{\frac{2\sqrt{P}}{\sigma^2},\frac{2}{\sigma}}(\alpha_n^{q,\Delta}, \alpha_{n+1}^{q,\Delta}) + g_{-\frac{2\sqrt{P}}{\sigma^2},\frac{2}{\sigma}}(\alpha_n^{q,\Delta}, \alpha_{n+1}^{q,\Delta}) \Big) - 2\Delta i \int_{\alpha_i^{q,\Delta}}^{\alpha_{i+1}^{q,\Delta}} xp(x)dx \Bigg] + \frac{4}{\sigma^4}(P+\sigma^2) \quad (6)$$

The integration part can be easily derived using change of variables. ■

Fig. 1 also shows that Δ_{opt} is an increasing function of the channel SNR. For this simulation, we fix the transmitted power at $P = 10$watt and change the channel noise power in order to get different channel SNRs. It can be seen from Eq. 1 as channel SNR increases, $p(x)$ becomes wider, which results in a larger values for Δ_{opt}. Next, we characterize the impact of the fixed-point implementation on the performance of the LDPC decoding.

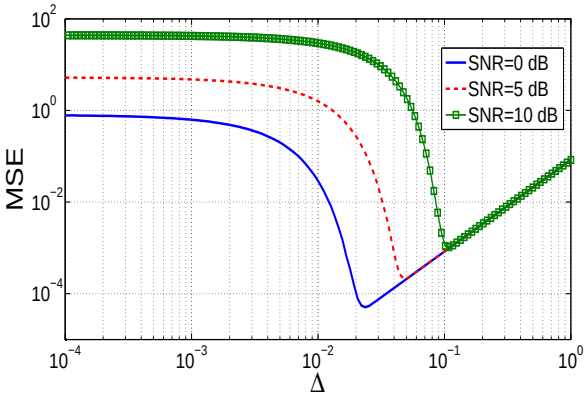

Fig. 1. Mean square quantization error v.s. quantizer step size for $q = 8$ and $P = 10$watt.

B. Fixed-point implementation of min-sum layered decoding algorithm

In this part, we analyze the impact of fixed-point implementation on the performance of the min-sum layered decoding algorithm without early stopping (i.e. using fixed number of iterations). For our analysis, we consider two modes of 1) floating-point operations and 2) fixed-point operations. Furthermore, we pick the layered decoding design in our analysis. For the floating-point operations, we utilize the min-sum algorithm, where all the operations occur in floating-point mode. However, in the fixed-point implementation, we assume that the mathematical operations happen over the fixed number of bits. In this case, if an operation results in a number which exceeds the maximum allowed number of bits, the decoder clips the value to the nearest allowable value.

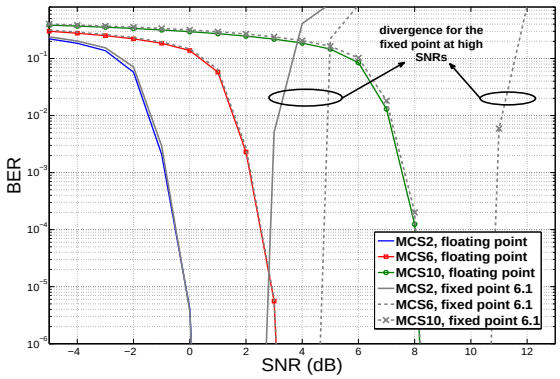

Fig. 2. Comparison between floating-point and fixed-point implementation of min-sum layer decoding algorithm for (672,336) irregular LDPC code and different modulation schemes in IEEE 802.11ad standard. The figure shows the performance after 8 number of iterations.

Fig. 2 shows the performance of both floating and fixed point implementations for 672-bit LDPC codes with rate of the $\frac{1}{2}$. It also compares the decoding performance for different modulation schemes of IEEE 802.11ad standard [12] (more information about this standard as well as different Modulation

and Coding Schemes (MCS) are provided in Section V). As can be seen, for small values of SNR, the fixed point scheme provides a similar performance as the float point one. However, after a certain point as channel SNR increases, the fixed point scheme diverges. In order to understand this behavior, we numerically analyze the Probability Density Function (PDF) of the extrinsic messages.

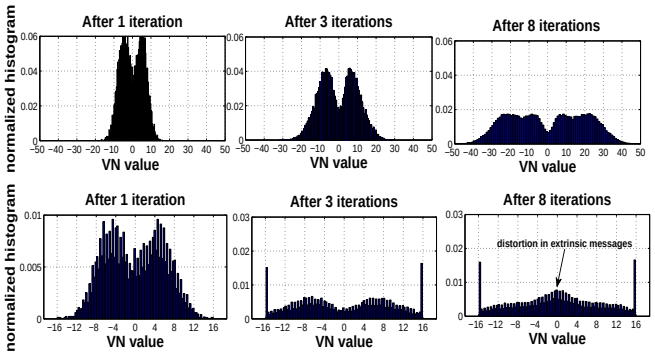

Fig. 3. Normalized histogram of the VNs of degree 3 for (672,336) irregular LDPC and channel SNR of 0dB – (top) floating-point min-sum layered decoding, (bottom) 4.3 fixed-point min-sum layered decoding.

Fig. 3 shows the PDF of the extrinsic messages updated at VNs for both floating and fixed-point schemes. For the fixed-point analysis, we consider 4.3 quantization scheme. [2]. Fig. 3 (top) shows the dynamic of the VNs (of degree 3) for the floating-point scheme. As can be seen, as the number of iterations increases the LLR values start growing. The higher value of LLR in each VN represents less uncertainty in the corresponding vote. However, in fixed-point implementation, we have a few number of bits to store the values of extrinsic messages. Therefore, these values become saturated as they pass the allowable range of variations. For instance, Fig. 3 (bottom) shows the impact of saturation at $k = 3$. Another impact of the saturation can be observed at $k = 8$. As can be seen, after 8 iterations, the extrinsic values will be around zero with higher probability as compared to $k = 3$. This behavior is not desirable. From Eq. 2 and Eq. 4, we can easily show that $L_j^{\text{ps},l}(k) = L_j^{\text{ps},l-1}(k) + \rho_{c_i^l \to v_j}(k) - \rho_{c_i^l \to v_j}(k-1)$. Assume $L_j^{\text{ps},l-1}(k) = Q_{\max}^{q,\Delta}$, where $Q_{\max}^{q,\Delta} > 0$ denotes the maximum allowable value of the quantizer. If $\rho_{c_i^l \to v_j}(k) > \rho_{c_i^l \to v_j}(k-1)$, then $L_j^{\text{ps},l}(k) = Q_{\max}^{q,\Delta}$. However, if $\rho_{c_i^l \to v_j}(k) < \rho_{c_i^l \to v_j}(k-1)$, then we have $L_j^{\text{ps},l}(k) < L_j^{\text{ps},l-1}(k) = Q_{\max}^{q,\Delta}$. Therefore, saturation can decrease the posterior probability, as we observed in Fig. 3 (bottom). In the next section, we propose an algorithm, which overcomes the saturation defects.

[2]This quantization scheme implies 4bits for the integer and 3bits for the fraction part.

IV. A FREEZING-BASED LOW-COMPLEX MIN-SUM ALGORITHM

In this section, we propose a freezing-based min-sum algorithm, which overcomes the saturation problem. In Fig. 3, we observed that as the number of iterations increases, the LLR values start growing up. However, due to the saturation, these values can not exceed the allowable range. In such cases, the saturated LLR values start moving toward zero, which is not desirable. In order to overcome this problem, we freeze the posterior LLR values as soon as they reach the saturation regions.

Let $I_j^l(k)$ denote the saturation indicator of the posterior LLR values corresponding to the jth variable node at time k and layer l. We then have $I_j^l(k) = F_R(L_j^{\text{ps},l}(k))$, where

$$F_R(z) = \begin{cases} 0, & -2^{R-1} < z < 2^{R-1} - 1; \\ 1, & \text{otherwise.} \end{cases} \tag{7}$$

Define the saturation function as

$$S_R(z) \triangleq \begin{cases} -2^{R-1}, & z \le -2^{R-1}; \\ z, & -2^{R-1} < z < 2^{R-1} - 1; \\ 2^{R-1} - 1, & z \ge 2^{R-1} - 1. \end{cases} \tag{8}$$

for $z \in \mathbb{Z}$. The proposed fixed-point and freezing-based min-sum decoding algorithm can be summarized as follows:

- Initialization: For $k = 0$, the check-to-variable messages are initialized to zero. Furthermore, $L_j^{\text{ps},0}(0) = Q^{R+1,\Delta_{\text{opt}}}(L_j^{\text{pr}})$ and $I_j^l(0) = 0$ for $1 \le j \le NZ$.
- At iteration k and layer l:
  - Each check node at lth layer receives the following messages from its VN neighbors:

$$\rho_{v_j \to c_i^l}(k) = \begin{cases} S_R\Big(L_j^{\text{ps},l-1}(k) - \rho_{c_i^l \to v_j}(k-1)\Big), & I_j^l(k) = 0; \\ S_R\Big(L_j^{\text{ps},l-1}(k)\Big), & I_j^l(k) = 1. \end{cases} \tag{9}$$

  for $j \in \mathcal{N}_{c_i^l}$ and $1 \le i \le Z$. We also have $L_j^{\text{ps},0}(k) = L_j^{\text{ps},M}(k-1)$.
  - Each check node will then send the following value to its VN neighbors

$$\rho_{c_i^l \to v_j}(k) = \begin{cases} S_R\big(\frac{\varepsilon_{c_i^l,v_j}(k)}{\alpha} |\rho_{v_{m_i^l(k)} \to c_i^l}(k)|\big), & j \ne m_i^l(k); \\ S_R\big(\frac{\varepsilon_{c_i^l,v_j}(k)}{\alpha} |\rho_{v_{n_i^l(k)} \to c_i^l}(k)|\big), & j = m_i^l(k). \end{cases}$$

  where $m_i^l(k)$, $n_i^l(k)$ and $\varepsilon_{c_i^l,v_j}(k)$ are as defined in Section II-A.
- The posterior LLR value is updated as:

$$L_j^{\text{ps},l}(k) = \begin{cases} \rho_{v_j \to c_i^l}(k) + \rho_{c_i^l \to v_j}(k), & I_j^l(k) = 0; \\ L_j^{\text{ps},l-1}(k), & I_j^l(k) = 1. \end{cases}$$

- Hard decisions are then made based on the sign of the posterior LLR values. The syndrome of the codeword is then checked in order to detect the error.

Fig. 4 shows the block diagram of the freezing-based min-sum layered decoding algorithm. As can be seen, in the proposed algorithm, the check node operations occur in R bits. However, variable nodes require R+1 bits. This design

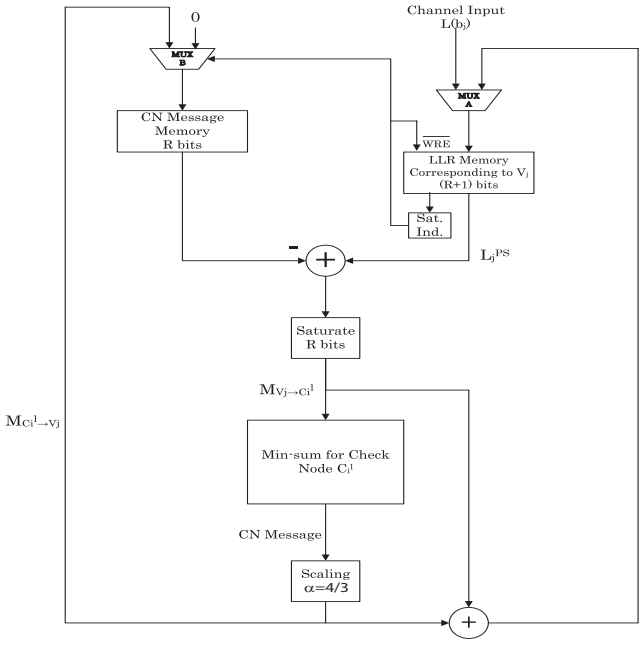

Fig. 4. Block diagram of the proposed freezing-based min-sum layered decoding algorithm.

requires only one saturation unit, which further reduces the implementation complexity. As we mentioned in the algorithm, an indicator bit is required to monitor the saturation status of each variable node. In order to decrease the number of required bits, this bit is generated from the stored LLR values using the circuit logics. The saturation indicator bit activates the writing disable switch of the jth variable node. Furthermore, this bit is connected to the select input of Multiplexer B, which activates the zero input of the Multiplexer, when the variable node value is in the saturation region. As can be seen, in this design Multiplexer B is placed before CN memory unit, hence, it is not part of the critical path. However, this design results in one time-step delay in Eq. 9 when $I_j^l(k)$ becomes one. Our simulation results, however, confirm that the performance loss is negligible as compared to the original case, where Multiplexer B is placed in critical path and after the memory unit. Fig. 5 shows the PDF of the extrinsic

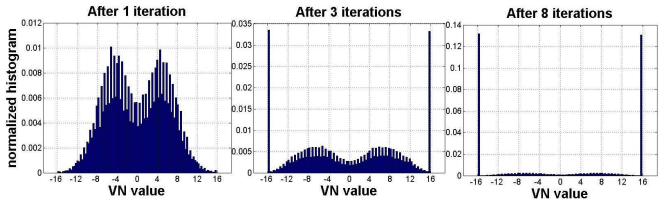

Fig. 5. Normalized histogram of the VNs of degree 3 for (672,336) irregular LDPC and channel SNR of 0dB for the proposed freezing-based min-sum layered decoding algorithm.

messages updated at VNs for the proposed freezing-based min-sum layered decoding algorithm. As can be seen the proposed algorithm overcomes the saturation problem of Fig. 3 (bottom). For instance, after 8 iterations, the extrinsic values will be around zero with negligible probability as compared to $k = 3$.

V. Simulation Results

Next, we show the performance of the proposed algorithm. All the simulations were performed using WiGigIEEE\802.11ad LDPC codes [12]. Here, we show the simulation results for the following schemes: MCS2, MCS6 and MCS10, where they utilize BPSK, QPSK and 16QAM modulations respectively. Furthermore, all these schemes use 672-bit irregular protograph LDPC codes with the rate of $\frac{1}{2}$.

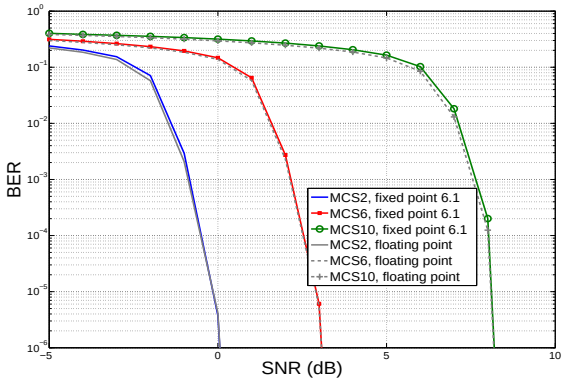

Fig. 6. Comparison between floating-point and the proposed freezing-based min-sum layered decoding algorithm for (672,336) irregular LDPC code and different modulation schemes in IEEE 802.11ad standard.

Fig. 6 compares the performance of the proposed freezing-based min-sum layered decoding algorithm with the floating-point one after 8 decoding iterations. The freezing-based algorithm is implemented under 6.1 bit constraint. Furthermore, Fig. 6 also compares the decoding performance for different modulation schemes. As can be seen, the proposed algorithm overcomes the saturation problem and provides a very close performance to the floating-point scheme, which is the ultimate reference for the fixed-point implementation.

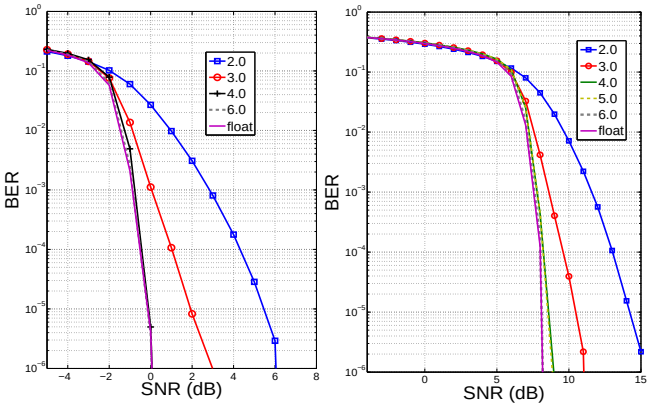

Fig. 7. Bit precision for freezing-based min-sum layered decoding algorithm– (left) MSC2 and (right) MCS10.

Fig. 7 (left) and Fig. 7 (right) compare the performance of the proposed algorithm with the floating-point min-sum layered decoding under different bit precisions for MCS2 and MCS10 of IEEE 802.11ad standard respectively. As can be seen, as the number of bits increases the performance of the proposed algorithm becomes better. Furthermore, for both modulation schemes and under 4.0 bit constraint, the proposed freezing-based algorithm provides the similar performance to the floating-point approach.

As can be seen, the proposed fixed-point decoder reaches the the floating-point performance under low hardware complexity. The impact of quantization on the decoding performance has been discussed in several literatures. For instance, in [13] authors investigated the impact of fixed-point quantization on the performance of sum-product decoder and proposed a dually-quantized sum-product decoder to mitigate the error floor. They reported the results for the SNRs between 2.5dB and 6.5dB. However, the proposed freezing-based approach, requires less hardware complexity and provides a wider operating range of SNRs for a fixed number of iterations. As another alternative, authors in [6] suggested that the variable nodes can use a down-scaled intrinsic information iteratively to improve the reliability of extrinsic information at the variable nodes for the fixed-point implementation. This approach requires different weights for different VNs, depending on their degrees, which increases the decoder complexity. Furthermore, the down scaling factor needs to be optimized for a specific number of iterations. As another drawback, the down-scaling weights reduce the convergence speed of the high degree variable nodes. The proposed freezing-based approach in this paper, requires less hardware complexity. Furthermore, we compared the results with the floating point alternative, which is the ultimate reference.

VI. Conclusion

In this paper, we analyzed the impact of the quantization and the fixed-point implementation on the performance of the min-sum layered decoding algorithm. We then proposed a freezing-based min-sum decoding algorithm for the fixed implementation. We showed that the proposed approach overcomes the saturation problems at high SNRs and improves the decoding performance drastically. Furthermore, we devised an optimum uniform quantization scheme, which minimizes the quantization error of the channel LLR values. Our simulation results show that the proposed algorithm can almost reach the floating-point decoding performance under 4.0 quantization scheme.

References

[1] R. G. Gallager, *Low-Density Parity-Check Codes*. The MIT Press, Cambridge, Massachusetts, 1963.

[2] M. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," *IEEE Transactions on Communications*, vol. 47, pp. 673 –680, may 1999.

[3] J. Zhao, F. Zarkeshvari, and A. Banihashemi, "On implementation of min-sum algorithm and its modifications for decoding low-density parity-check (ldpc) codes," *IEEE Transactions on Communications*, vol. 53, pp. 549–554, april 2005.

[4] J. Zhang, M. Fossorier, and D. Gu, "Two-dimensional correction for min-sum decoding of irregular ldpc codes," *IEEE Communications Letters*, vol. 10, mar 2006.

[5] J. Chen, R. Tanner, C. Jones, and Y. Li, "Improved min-sum decoding algorithms for irregular ldpc codes," in *International Symposium on Information Theory*, sept. 2005.

[6] D. Oh and K. Parhi, "Performance of quantized min-sum decoding algorithms for irregular ldpc codes," in *IEEE International Symposium on Circuits and Systems*, pp. 2758 –2761, may 2007.

[7] S. Kim, G. Sobelman, and H. Lee, "Adaptive quantization in min-sum based irregular ldpc decoder," in *IEEE International Symposium on Circuits and Systems*, may 2008.

[8] J. Thorpe, "Low-density parity-check (ldpc) codes constructed from protographs," *Tech. Rep. 42-154, IPN Progress Report*, aug 2003.

[9] Z. Cui, Z. Wang, and Y. Liu, "High-throughput layered ldpc decoding architecture," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, pp. 582 –587, april 2009.

[10] D. Hocevar, "A reduced complexity decoder architecture via layered decoding of ldpc codes," in *IEEE Workshop on Signal Processing Systems, 2004. SIPS 2004.*, pp. 107 – 112, oct. 2004.

[11] E. Eleftheriou, T. Mittelholzer, and A. Dholakia, "Reduced-complexity decoding algorithm for low-density parity-check codes," *Electronics Letters*, vol. 37, pp. 102 –104, jan 2001.

[12] "IEEE 802.11ad. PHY/MAC complete proposal specification (approved as IEEE 802.11ad D0.1)," may 2010. http://tinyurl.com/2fqlkxx.

[13] Z. Zhang, L. Dolecek, B. Nikolic, V. Anantharam, and M. Wainwright, "Design of ldpc decoders for improved low error rate performance: quantization and algorithm choices," *IEEE Transactions on Communications*, 2009.