This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.

Quantization Effects in Low-Density Parity-Check Decoders

Zhengya Zhang, Lara Dolecek, Martin Wainwright, Venkat Anantharam, and Borivoje Nikolić
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, CA 94720, USA

Abstract–**A class of combinatorial structures, called absorbing sets, strongly influences the performance of low-density parity-check (LDPC) decoders. In particular, the quantization scheme strongly affects which absorbing sets dominate in the error-floor region. Absorbing sets may be characterized as weak or strong. They are a characteristic of the parity check matrix of a code. Conventional quantization schemes applied to a (2209,1978) array-based LDPC code can induce low-weight weak absorbing sets and, as a result, elevate the error floor. Adaptive quantization schemes alleviate the effects of weak absorbing sets, and, as a result, only the strong ones dominate the error floor of an optimized decoder implementation. Another benefit of an adaptive quantization scheme is that it performs well even in very few iterations.**

I. Introduction

Low-density parity-check (LDPC) codes have been demonstrated to perform very close to the Shannon limit when decoded iteratively [1]. Sometimes, excellent performance is only observed up until a moderate bit error rate (BER), because at lower BER, the waterfall curve changes slope leading to a so-called error floor [2]. Such error floors are a major factor in limiting the employment of LDPC codes in high-throughput applications.

In previous work [3], we have developed an FPGA-based emulation system to investigate the causes of error floors. We showed that error floor behavior is strongly connected to the quantization. A high-precision decoder implementation lowers the error floor to a level that is dominated by non-codeword absorbing sets. Absorbing sets are intrinsic to the parity check matrix of the code. More absorbing sets emerge in a low-precision implementation.

This paper presents an analysis of quantization effects on the performance of LDPC codes. Compared to some related work [4]–[6], our study is performed under the constraints of limited wordlength and limited number of decoding iterations, and focuses on the composition of dominant absorbing sets in the error floor region. In this process, we classify the absorbing sets in terms of their absorbing strength. Most importantly, we propose improved quantization techniques that alleviate the effects due to weak absorbing sets.

In Section II, we provide background on the array-based LDPC code, a precise definition of absorbing sets, and the sum-product message-passing decoding algorithm. We then investigate the quantization effects through a series of experiments using the (2209,1978) array code. The results are presented in Section III. We develop methods to improve upon standard quantization approaches, thereby reducing the effects of weak absorbing sets and lowering the error floor. In Section IV, we study the effect of finite number of decoding iterations on the error floor performance. Section V concludes the paper and considers directions for future work.

II. Background

A. Array-based LDPC Code

Array-based LDPC codes [7] are regular LDPC codes parameterized by a pair of integers (p, γ), where $\gamma \le p$, p is an odd prime. The **H** matrix ($H_{p,\gamma}$) is given by

$$H_{p,\gamma} = \begin{bmatrix} I & I & I & \cdots & I \\ I & \sigma & \sigma^2 & \cdots & \sigma^{p-1} \\ I & \sigma^2 & \sigma^4 & \cdots & \sigma^{2(p-1)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ I & \sigma^{\gamma-1} & \sigma^{(\gamma-1)2} & \cdots & \sigma^{(\gamma-1)(p-1)} \end{bmatrix}, \qquad (1)$$

where σ denotes a $p \times p$ permutation matrix of the form

$$\sigma = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}. \qquad (2)$$

Array codes have been demonstrated to have very good performance [7]. The structure of an array code makes it suitable for efficient encoder and decoder implementations [8], [9]. The investigations covered in this paper are based on a (2209,1978) array code with $p = 47$ and $\gamma = 5$.

B. Definition of an Absorbing Set and Connectivity of a Variable Node in Absorbing Sets

Let $G = (V, F, E)$ be a bipartite graph with the vertex set $V \cup F$, where V and F are disjoint, and with the edge set E, such that there exists an edge $e(i, j) \in E$ iff $i \in V$ and $j \in F$. One can associate a bipartite graph $G_H = G_H(V, F, E)$ with a parity check matrix **H**, such that the set V corresponds to the columns of **H**, the set F corresponds to the rows of **H**, and $E = \{e(i, j) \mid \mathbf{H}(j, i) = 1\}$. Such a graph G_H is commonly referred to as the Tanner graph of the parity check matrix **H** [10], [11].

1-4244-0353-7/07/$25.00 ©2007 IEEE

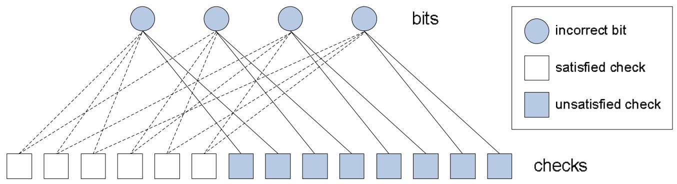

Figure 1. Illustration of the (4,8) absorbing set.

For a subset D of V, let $\mathcal{E}(D)$ (resp. $\mathcal{O}(D)$) be the set of neighboring vertices of D in F with even (resp. odd) degree with respect to D. Given an integer pair (a, b), an (a, b) absorbing set is a subset A of V of size a, with $\mathcal{O}(A)$ of size b, and with the property that each element of A has strictly fewer neighbors in $\mathcal{O}(A)$ than in $F \setminus \mathcal{O}(A)$ [12].

An example of an (a, b) absorbing set with $a = 4$, $b = 8$ is given in Fig. 1, where full circles constitute the set A, full squares constitute the set $\mathcal{O}(A)$, empty squares constitute the set $\mathcal{E}(D)$, $E(A, \mathcal{O}(A))$ is given by solid lines, and $E(A, \mathcal{E}(A))$ is given by dashed lines. Observe that each element in A has more even-degree than odd-degree neighbors.

We introduce the notation p:q to describe the connectivity of a variable (bit) node in an absorbing set with p connections to satisfied check nodes and q connections to unsatisfied check nodes. In the (4,8) absorbing set shown in Fig. 1, each variable node has a 3:2 connection. A bit remains incorrect in the absorbing state because it receives more incorrect messages from the satisfied checks than correct messages from the unsatisfied checks [3]. In a regular LDPC code with column weight γ, γ is $p + q$. A variable node in an absorbing set can have a connectivity of some type $(\gamma - q)$:q, with $q \geq 0$ and $q < \gamma/2$.

C. The Sum-Product Message-Passing Decoding Algorithm

Low-density parity-check codes are usually iteratively decoded using the sum-product algorithm. The algorithm can be visualized using a Tanner graph, where soft messages are exchanged between variable nodes and check nodes and convergence can usually be achieved within a small number of iterations. In the first step of the algorithm, variable nodes x_i are initialized with the prior log-likelihood ratios defined in (3) using the channel outputs y_i:

$$L^{pr}(x_i) = \log \frac{\Pr(x_i = 0 \mid y_i)}{\Pr(x_i = 1 \mid y_i)}. \qquad (3)$$

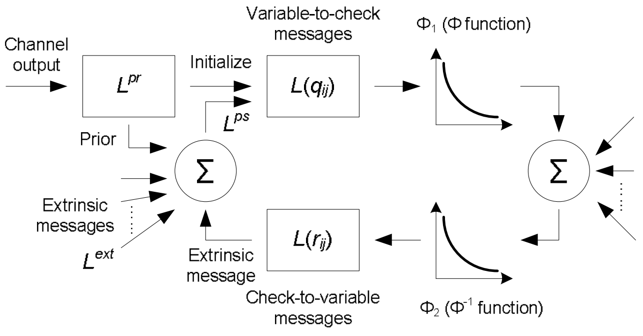

Figure 2. A message-passing decoder.

The formulation of (3) assumes a BPSK modulation and an additive white Gaussian noise (AWGN) channel.

The variable nodes send messages to the check nodes following the edges defined by the Tanner graph. The variable-to-check messages are summed at each check node and the marginalized sum is returned to the corresponding variable node. Each variable node then updates its decision based on the prior and the information received from all the neighboring check nodes. The marginalized sum is used as the variable-to-check message in the next iteration. An illustration of the iterative decoding procedure is shown in Fig. 2. The figure is simplified in the representation of the posterior LLR, as only the marginalized L^{ps} can be used as the variable-to-check message for the next decoding iteration.

Variable-to-check and check-to-variable messages are calculated using equations (4), (5), and (6). The messages q_{ij} and r_{ij} refer to the variable-to-check and check-to-variable messages that are passed between the i^{th} variable node and the j^{th} check node. In representing the connectivity of the Tanner graph, $Col[i]$ refers to the set of all the check nodes adjacent to the i^{th} variable node and $Row[j]$ refers to the set of all the variable nodes adjacent the j^{th} check node.

$$L(q_{ij}) = \sum_{j' \in Col[i]} L(r_{ij'}) - L(r_{ij}) + L^{pr}(x_i), \qquad (4)$$

$$L(r_{ij}) = \Phi^{-1}\left(\sum_{i' \in Row[j]} \Phi\left(\left|L(q_{i'j})\right|\right) - \Phi\left(\left|L(q_{ij})\right|\right) \right) \times \left(\operatorname{sgn}\left(L(q_{ij})\right) \prod_{i' \in Row[j]} \operatorname{sgn}\left(L(q_{i'j})\right) \right), \qquad (5)$$

$$\Phi(x) = -\log\left(\tanh\left(\frac{1}{2}x\right) \right), \quad x \geq 0. \qquad (6)$$

The iterative decoding algorithm is allowed to run until all the parity-check equations are satisfied or when an upper limit on the iteration number is reached, whichever occurs earlier. After this a hard decision is performed based on the posterior LLR (7) – (9).

$$L^{ext}(x_i) = \sum_{j' \in Col[i]} L(r_{ij'}), \qquad (7)$$

$$L^{ps}(x_i) = L^{ext}(x_i) + L^{pr}(x_i), \text{and} \qquad (8)$$

$$\hat{x}_i = \begin{cases} 0, & if \quad L^{ps}(x_i) \geq 0 \\ 1, & if \quad L^{ps}(x_i) < 0. \end{cases} \qquad (9)$$

D. Quantization Procedures

Practical high data throughput implementations of the iterative decoding algorithm only approximate the ideal operation of the decoder. Such approximations are inevitable for two reasons: (a) wordlength is necessarily limited, which causes saturation and quantization effects, thereby making it difficult to accurately represent messages and carry out arithmetic operations; and (b) the number of iterations is limited, so that the effectiveness of iterative decoding cannot be fully realized.

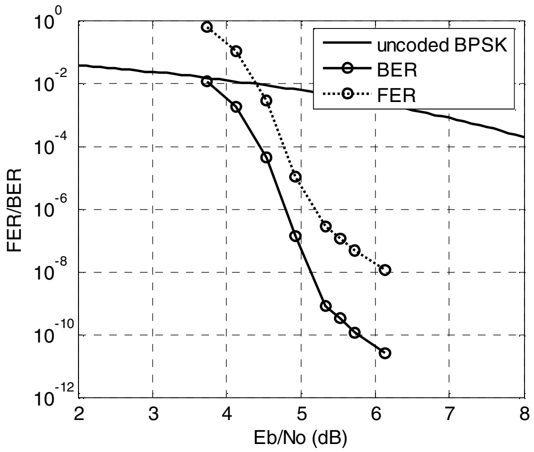

Figure 3. Performance of the (2209, 1978) array code implemented using a 4.2 uniform quantization.

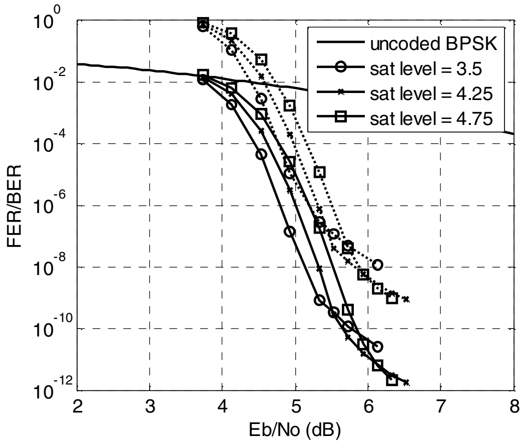

Figure 4. The effect of adjusting the strength of extrinsic messages in a 4.2 uniform quantized implementation.

The imperfections are illustrated by considering a pass through the decoding loop shown in Fig. 2. The channel output is saturated and quantized before it is saved as the prior log-likelihood ratio, L^{pr}. During the first phase of message passing, variable-to-check messages pass through the Φ transformation defined in (6), then the summation (which includes marginalization), and finally the Φ^{-1} transformation. The Φ function is its own inverse, so the two transformations are identical. We refer to them as Φ_1 and Φ_2. The Φ function is approximated by discretization. The input and output of the function are saturated and quantized, thus the characteristics of this function cannot be fully captured especially in the regions approaching infinity and zero.

In the second phase of message passing, check-to-variable messages are combined with the prior to produce the posterior probability, L^{ps}. The hard decision is based on L^{ps}, but L^{ps} is not always accurately represented because its constituents, L^{pr} and L^{ext}, are inaccurate. The prior, L^{pr}, is the saturated and quantized channel output; the extrinsic message, L^{ext}, is the sum of check-to-variable messages, which originate from the saturated and quantized outputs of the Φ_2 function. There is additional error propagation that is lumped to L^{ext}. The inaccurate L^{ps} causes a decoder to perform worse than theoretically possible. This deficiency is usually manifested as a loss in performance in the cliff region and/or a rise of the error floor.

The saturation and quantization effects are related to the fixed-point number format that is used in the processing and storage of data. We use the notation *m.f* to represent a signed fixed-point number with *m* bits to the left of the radix point to represent integer values and *f* bits to the right of the radix point to represent fractional values. Such a fixed-point representation translates to a quantization step size (or precision) of 2^{-f} (in a uniform quantization) with a maximum value of $2^{m-1} - 2^{-f}$ and a minimum value of -2^{m-1}. Note that there is an asymmetry between the maximum value and the minimum value because 0 is represented with a positive sign in this number format. Values above the maximum or minimum are saturated (i.e., clipped). The wordlength of this fixed-point number is $m + f$. As an example, a 4.2 signed fixed-point uniform quantization translates to a quantization step size of 0.25, with a maximum of 7.75 and a minimum of -8.

III. Finite Wordlength Effects

The wordlength of the decoder design is limited to 6 bits in all the experiments. A maximum of 200 decoding iterations is performed to isolate the quantization effect from the iteration number effect. We begin by using a 4.2 fixed-point uniform quantization scheme on the prior, the variable-to-check and check-to-variable messages [13], as well as the input and output of the Φ function. The performance of this implementation is shown in Fig. 3.

Based on our emulation results, the failures in the error floor region are entirely due to absorbing sets. We take 58 error samples at 6.1 dB and 6.3 dB of SNR in the error floor region. A partial list of absorbing sets is shown in Table I. The structure of the dominant (4,8) absorbing set is illustrated in Fig. 1. The other structures are not drawn for the lack of space. Among the other absorbing sets listed in Table I, the (6,8), (8,6), and (8,8) structures have the least number of variable nodes with 3:2 connections.

A. Strength of Extrinsic Messages

Extrinsic messages are the outputs of the Φ_2 function. Moving the saturation level of the Φ_2 function (i.e., $\Phi_2(0)$) adjusts the strength of the (most confident) extrinsic messages, thus affecting the decoder performance. Fig. 4 shows that stronger extrinsic messages lower the error floor (note that the error floor performance ceases to improve as the strength of extrinsic messages reaches a certain level) but worsen the performance in the cliff region.

TABLE I
A Partial List of Absorbing Sets in a 4.2 Decoder Implementation
(SNR = 6.1 dB and 6.3 dB, $\Phi_2(0) = 3.5$, Total 58 Cases)

Absorbing sets	(4,8)	(5,9)	(6,8)	(8,6)	(8,8)
Cases (out of 58)	45	6	5	0	2

This observation can be intuitively explained as strong extrinsic messages permitting the correct bits to exert strong influences on the incorrect bits (favorable influences) and the incorrect bits to exert strong influences on the correct bits (adverse influences). The SNR level determines whether the favorable influences overpower the adverse influences or the opposite. In a low-SNR cliff region, more bits are received incorrectly. A large number of strong adverse influences tend to encourage excessive error propagation, which stalls the convergence. At a high SNR level, very few bits are received incorrectly. Strong extrinsic messages allow the favorable influences to significantly outnumber and overpower the adverse influences (and the incorrect priors), which makes it more difficult to get trapped by an absorbing set.

The above describes the average behavior of a message-passing decoder. An absorbing set is a special configuration at the high SNR level where seemingly satisfied checks gather enough adverse influences that outnumber the favorable influences, therefore we conjecture that strengthening extrinsic messages uniformly is not likely to change an absorbing configuration. We verify this conjecture by strengthening the extrinsic messages and observe the failure cases in the error floor region. Partial lists of the absorbing sets are shown in Table II. We observe that the (4,8) absorbing set remains the dominant cause of error floors when the extrinsic messages are strengthened.

B. Differentiation among Extrinsic Messages

At a high SNR level or when the decoder starts to converge, the variable-to-check messages usually grow larger. We observe that the decoder is essentially operating on the lower right corner of the Φ_1 curve and subsequently on the upper left corner of the Φ_2 curve. We refer to these corners as the operating regions of the Φ_1 and Φ_2 functions. Differentiating extrinsic messages requires more output levels of the Φ_2 function in its operating region, which also necessitates more precise inputs to the Φ_2 function. The requirements can be both satisfied if the quantization scheme is designed more intelligently by dividing the decoder into two quantization domains illustrated in Fig. 5. For instance, Domain A uses a 4.2 fixed-point uniform quantization while Domain B could use a more precise 1.5 fixed-point uniform quantization. The 6-bit wordlength is preserved to maintain the decoder complexity. The functions Φ_1 and Φ_2 separate the two domains. The input to Φ_1 is in a 4.2 quantization and the output of Φ_1 is in a 1.5 quantization. The Φ_2 function assumes the opposite quantization assignment.

TABLE II
A PARTIAL LIST OF ABSORBING SETS IN A 4.2 DECODER IMPLEMENTATION
(a) SNR = 6.1 DB AND 6.3 DB, $\Phi_2(0) = 4.25$, TOTAL 58 CASES

Absorbing sets	(4,8)	(5,9)	(6,8)	(8,6)	(8,8)
Cases (out of 58)	48	2	6	2	0

(b) SNR = 6.1 DB AND 6.3 DB, $\Phi_2(0) = 4.75$, TOTAL 58 CASES

Absorbing sets	(4,8)	(5,9)	(6,8)	(8,6)	(8,8)
Cases (out of 58)	38	3	10	2	2

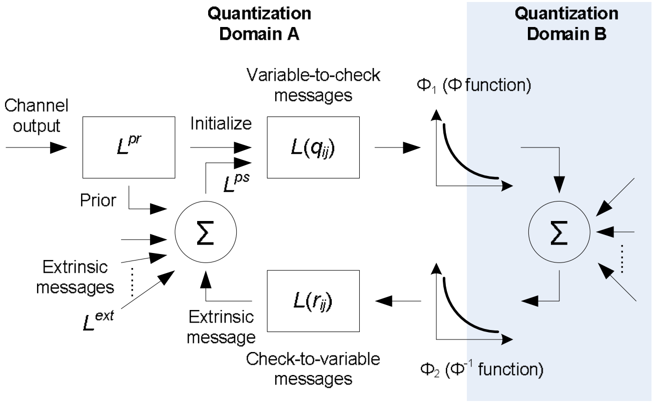

Figure 5. Division of the quantization domains.

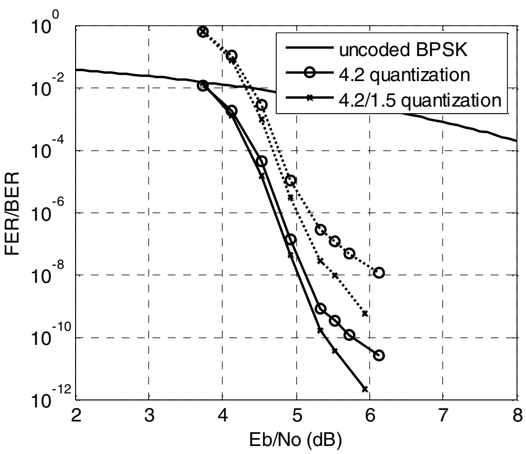

Figure 6. Performance comparison between the 4.2 quantization and the 4.2/1.5 adaptive quantization.

Fig. 6 shows that the 4.2/1.5 adaptive quantization outperforms the 4.2 quantization in both the cliff and the error floor regions. The performance advantage of the 4.2/1.5 adaptive quantization is attributed to more levels in the operating region of the Φ_2 function, which enable a more accurate representation of the extrinsic messages. Intuitively, more accurate extrinsic messages reduce the excessive error propagation from numerous incorrect bits at a low SNR, thus the performance in the cliff region is improved. It is also possible to precisely differentiate extrinsic messages with more quantization levels, so that the favorable influences could potentially obtain a stronger representation than the adverse influences, and the absorbing-set errors become less likely.

The above explanation is verified by the failures in the error floor region. The partial list of absorbing sets is shown in Table III.(a). The (4,8) absorbing sets observed in the 4.2 quantization are almost completely removed using the adaptive quantization approach. The error floor is now dominated by (6,8) and (8,6) absorbing sets. These absorbing sets are illustrated in Fig. 7 using the same notations as in Fig. 1.

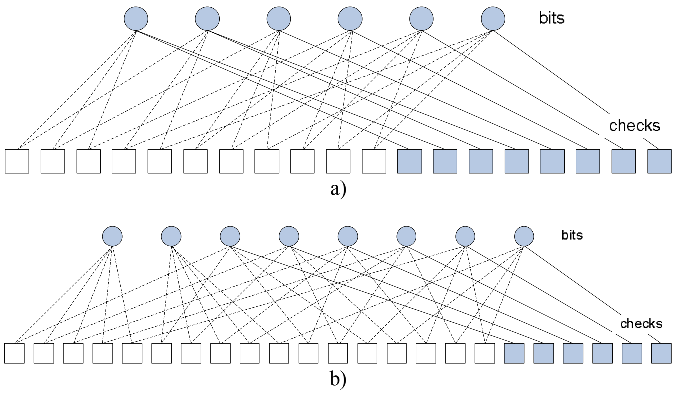

Figure 7. Illustration of a) the (6,8) and b) the (8,6) absorbing set.

The (8,6) absorbing set consists of 4:1 and 5:0 variable node configurations. The 5:0 configuration is the strongest absorbing scenario in which all the extrinsic messages agree to the wrong decision. The 4:1 configuration is also a strong absorbing scenario where 4 incorrect messages considerably overpower the single correct message. Therefore, the (8,6) absorbing set can be considered a strong absorbing set.

The (6,8) absorbing set consists of 4:1 and 3:2 variable node configurations. Because of the weaker 3:2 configurations, the (6,8) absorbing set is considered weaker than the (8,6) absorbing set. In fact, we can apply stronger extrinsic messages as in Section III.A by raising the saturation level of the Φ_2 function such that it becomes more difficult to enter the (6,8) absorbing state. Table III.(b) shows that the importance of (6,8) absorbing set is lowered, thereby leaving the (8,6) absorbing set as the only dominant cause of error floors.

Note that strong absorbing sets are defined relative to weak absorbing sets. In poor decoder implementations, the performance can be dominated by weak and often low-weight absorbing sets, which lead to a higher error floor. The quantization can be improved such that the effects of weak absorbing sets are alleviated. The error floor performance is eventually dominated by strong absorbing sets.

C. Representation of Channel Likelihoods

In an AWGN channel, the prior log-likelihood ratio, i.e., the input to the decoder, is given by $L^{pr}(x_i) = 2y_i/\sigma^2$, where y_i is the channel output and σ^2 is the noise variance. At a SNR level in the range of interest, 5 dB for instance, the input $L^{pr}(x_i) \sim N(\pm 11.33, 22.65)$. A 4.2 quantization scheme is not sufficient to capture the distribution of the input. Moreover, it clips correct priors and incorrect priors disproportionately. Simple calculations show that the 4.2 quantization scheme clips approximately 76% of the correct priors; in contrast, it only clips 0.071% of the incorrect priors.

An adaptive quantization approach provides more flexibility in choosing the most suitable quantization in each domain. A 6.0 uniform quantization can be selected to replace the 4.2 quantization in Domain A. The 6.0 quantization accepts a higher dynamic range, which permits the correct priors to assume a stronger representation without being clipped excessively. Variable nodes backed by stronger correct priors cannot be easily pulled into an absorbing configuration, thus the likelihood of absorbing-set errors is reduced. The performance of the 6.0/1.5 adaptive quantization design is shown in Fig. 8. The error floor performance improves over the 4.2/1.5 adaptive quantization.

TABLE III
A PARTIAL LIST OF ABSORBING SETS IN A 4.2/1.5 DECODER IMPLEMENTATION
(a) SNR = 5.5 DB AND 5.7 DB, $\Phi_2(0) = 5.5$, TOTAL 58 CASES

Absorbing sets	(6,8)	(7,9)	(8,6)	(8,8)	(9,5)	(10,4)
Cases (out of 58)	20	2	18	4	0	0

(b) SNR = 5.5 DB AND 5.7 DB, $\Phi_2(0) = 6.25$, TOTAL 58 CASES

Absorbing sets	(6,8)	(7,9)	(8,6)	(8,8)	(9,5)	(10,4)
Cases (out of 58)	6	1	23	3	6	4

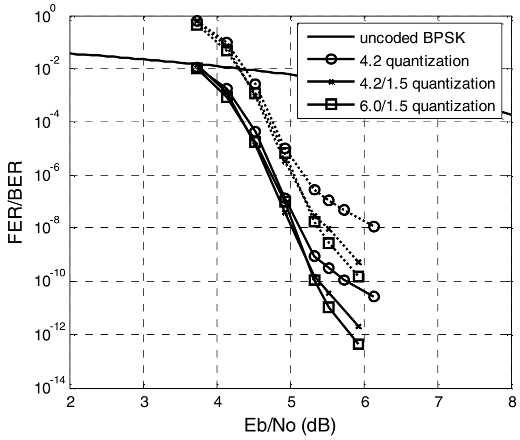

Figure 8. Performance comparison among different quantization schemes.

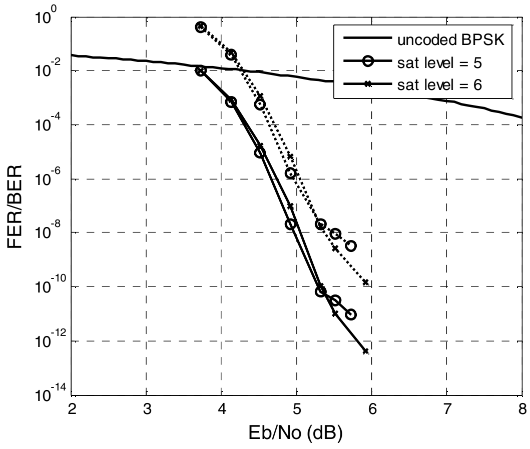

Figure 9. Performance of the 6.0/1.5 quantization when the saturation level of the extrinsic messages is adjusted.

The adaptive quantization approach can be fine-tuned by adjusting the saturation level for extrinsic messages as in Section III.A. Fig. 9 shows that there exists an optimal saturation level for extrinsic messages at each SNR region. At a low SNR, a lower saturation level results in a better performance in the cliff region. As the SNR increases towards the error floor range, a higher saturation level improves the error floor performance. An adaptive saturation level for extrinsic messages brings the optimal performance at each SNR level.

D. Applicability of the Results

The concept of strong and weak absorbing sets applies to other LDPC code constructions. We observe that moving the saturation level for extrinsic messages shifts the performance curve of a (2048,1723) RS-LDPC code [14] in the cliff region. An adaptive quantization approach further reduces the implementation loss in the cliff region. We also observe that in a (2048,1765) RS-LDPC code, some conventional quantization schemes could result in weak absorbing sets dominating the error floor performance. The quantization approach can be improved to alleviate the effects of weak absorbing sets and lower the error floor.

IV. Limited Number of Decoding Iterations

The number of decoding iterations is usually very limited in practice, as it determines the latency and throughput of the system. As an example, in the IEEE 802.3an 10GBASE-T standard which supports 10 Gb/s Ethernet over twisted-pair cabling, the maximum number of iterations for the LDPC decoder is limited to around 6 to 8 [15]. In the following investigation, we choose the maximum number of decoding iterations to be 10, because the results would be relevant in some practical applications.

Fig. 10 shows the change in performance as the upper limit on the iteration number is reduced from 200 to 10 using a 4.2 uniform quantized decoder. Observe that a good performance in the cliff region can be achieved with as few as 10 iterations. The loss in performance in the cliff region is due to an insufficient number of iterations for the decoding to converge. The 10-iteration waterfall eventually “lands” on the 200-iteration error floor, so that the two error floors almost overlap. Analysis of the failures in this region confirms that the (4,8) absorbing set, the dominant cause of error floors in the 200-iteration decoder, causes the 10-iteration decoder to fail. This result suggests that the absorbing behavior usually happens very quickly and the absorbing structure emerges in full strength within 10 decoding iterations. Non-convergent errors, however, become less significant in a high SNR region.

The quantization effects observed previously are still applicable in a 10-iteration decoder. Strengthening extrinsic messages slows down the convergence. Fig. 11 confirms that the performance loss in the cliff region is widened with a higher saturation level for the extrinsic messages. As the strength of extrinsic messages is increased further, an oscillation behavior can occur [3] and it usually cannot be resolved within a small number of iterations, leading to a higher error floor.

The adaptive quantization approach improves the error floor performance even within 10 iterations. In particular, observe in Fig. 12 that the 6.0/1.5 adaptive quantization performs better in the error floor region than the 4.2 or the 4.2/1.5 quantization. This behavior can be understood as the excessive input clipping (using the 4.2 input quantization) of correct priors weakening the stabilities of the bits that are received correctly. These bits become less immune to adverse influences, thus the convergence is slowed down and absorbing-set errors become more likely.

V. Conclusions and Future Work

Limited wordlength and limited number of iterations are the fundamental constraints in any practical implementation of the iterative decoding algorithm. Quantization schemes under the limited wordlength constraint can be suboptimal, thus allowing weak absorbing sets of relatively small size to dominate, thereby leading to a higher error floor. In this work, we showed that an adaptive quantization approach improves the fidelity of extrinsic messages and channel likelihoods. It mitigates the effects of weak absorbing sets and lowers the error floor. Moreover, we established that adaptive saturation level for the extrinsic messages can be combined with the adaptive quantization approach to achieve the optimal performance in both the cliff and the error floor regions.

We also showed that a good performance can be achieved even with very few decoding iterations. Results based on 10 decoding iterations demonstrate that the loss due to non-convergent errors is small and the error floor performance does not significantly improve after the first 10 iterations. The performance is still strongly influenced by the quantization. Reducing the saturation level of extrinsic messages narrows the loss due to non-convergent errors and the adaptive quantization approach improves the error floor performance substantially.

In this work, we isolated weak absorbing sets from strong absorbing sets and we showed that quantization has a significant effect on the composition of absorbing sets in the error floor region. In future work, we intend to enumerate the possible absorbing sets and evaluate their influences, as it would allow us to predict of the location and the slope of the error floors. We also intend to explore the complexity dimension by investigating the error floor performance using the min-sum or a modified min-sum decoding algorithm.

Acknowledgement

The authors would like to thank Pierre-Yves Droz and Henry Chen for help with the BEE2 hardware emulation platform. This research was supported in part by NSF CCF grant #0635372, Marvell Semiconductor and Intel Corporation through the University of California MICRO program. NSF CNS RI grant #0403427 provided the computing infrastructure.

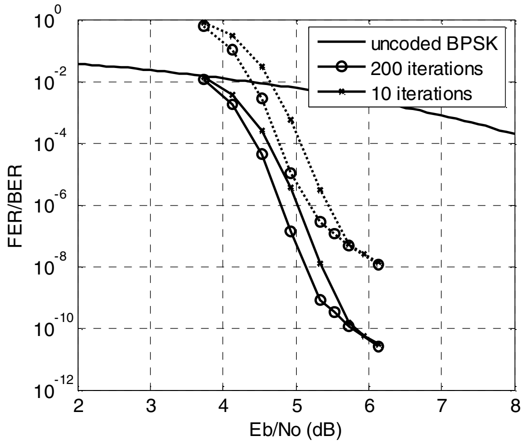

Figure 10. Limited number of decoding iterations using the 4.2 quantization with a relatively low Φ saturation level: Φ(0)=3.5.

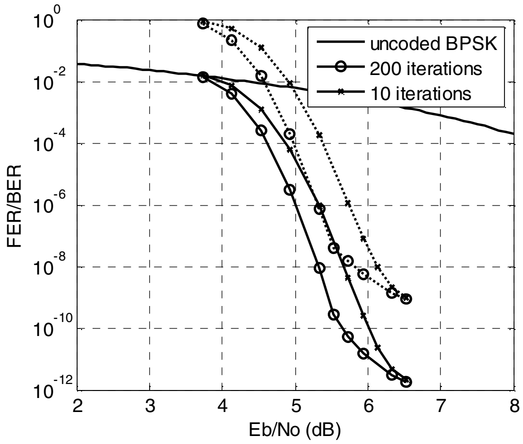

Figure 11. Limited number of decoding iterations using the 4.2 quantization with a higher Φ saturation level: Φ(0) = 4.25.

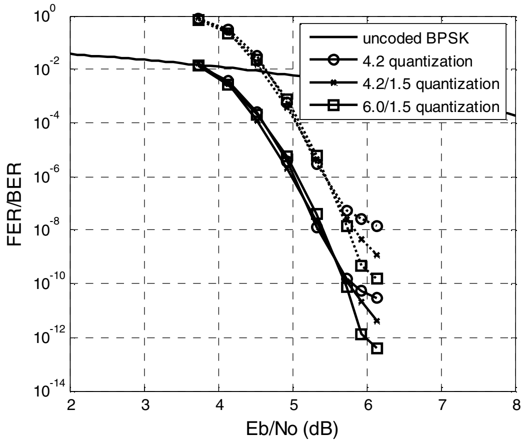

Figure 12. Quantization effect on the decoder performance using only 10 decoding iterations.

REFERENCES

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

[2] T. Richardson, "Error floors of LDPC codes," in *Proc. of the Allerton Conference on Communications, Control, and Computing*, Monticello, IL, pp. 1426-1435, Oct. 2003.

[3] Z. Zhang, L. Dolecek, B. Nikolic, V. Anantharam, and M. Wainwright, "Investigation of error floors of structured low-density parity-check codes by hardware emulation," in *Proc. of IEEE GLOBECOM*, San Francisco, CA, Nov. 2006.

[4] J. Chen, A. Dholakia, E. Eleftheriou, M.P.C. Fossorier, X. Hu, "Reduced-complexity decoding of LDPC codes," *IEEE Trans. on Communications*, vol. 53. no. 8, pp. 1288-1299, Aug. 2005.

[5] J. Zhao, F. Zarkeshvari, A. Banihashemi, "On implementation of min-sum algorithm and its modifications for decoding low-density parity-check (LDPC) codes," *IEEE Trans. on Communications*, vol. 53. no. 4, pp.549-554, Apr. 2005.

[6] J. Thorpe, "Low-complexity approximations to belief propagation for LDPC codes," available at http://www.systems.caltech.edu/~jeremy/research/papers/.

[7] J. Fan, "Array codes as low-density parity-check codes," *Second Int. Symp. on Turbo Codes and Related Topics*, Brest, France, pp. 543-46, Sept. 2000.

[8] E. Eleftheriou and S. Olcer, "Low-density parity-check codes for digital subscriber lines," in *Proc. of IEEE Int. Conf. on Communications*, New York, NY, pp. 1752-1757, Apr. 2002.

[9] S. Olcer, "Decoder architecture for array-code-based LDPC codes," in *Proc. of IEEE GLOBECOM*, San Francisco, CA, pp. 2046-2050, Dec. 2003.

[10] G.D. Forney, "Codes on graphs: normal realizations," *IEEE Trans. on Information Theory*, vol. 47, pp.520-548, Feb. 2001.

[11] N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation, Linkoping University, Sweden, 1996.

[12] L. Dolecek, Z. Zhang, V. Anantharam, M. Wainwright, B. Nikolic, "Analysis of absorbing sets for array-based LDPC codes," to appear in *Proc. of IEEE Int. Conf. on Communications*, Glasgow, UK, Jun. 2007.

[13] L. Sun, H. Song, Z. Keirn, and B.V.K.V. Kumar, "Field programmable gate array (FPGA) for iterative code evaluation," *IEEE Trans. on Magnetics*, vol. 42. no. 2, pp. 226-231, Feb. 2006.

[14] I. Djurdjevic, J. Xu, K. Abdel-Ghaffar, and S. Lin, "A class of low-density parity-check codes constructed based on Reed-Solomon codes with two information symbols," *IEEE Communications Letters*, vol. 7, no. 7, pp. 317-319, Jul. 2003.

[15] *Specific requirements Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications Amendment 1: Physical Layer and Management Parameters for 10 Gb/s Operation, Type 10GBASE-T*, IEEE Standard 802.3AN-2006, Sep. 2006.