Design of Fixed-Point Processing Based LDPC Codes Using EXIT Charts

Xin Zuo, Robert G. Maunder and Lajos Hanzo
School of ECS, University of Southampton, SO17 1BJ, United Kingdom.
Email: {xz3g08, rm, lh}@ecs.soton.ac.uk, http://www-mobile.ecs.soton.ac.uk

***Abstract*—The computational complexity and hence energy consumption of a low-density parity-check (LDPC) decoder employing fixed point (FP) computation is commensurate with the operand width (OW) of the FP representation employed. Therefore low OWs are desirable, but, if the OW is too low, then an inevitable performance degradation will be introduced. For that reason it is desirable to determine the minimum OW that does not impose a significant performance degradation. Previous efforts have advocated different OWs based on results obtained using time-consuming bit-error ratio (BER) simulations. However, these BER simulations are extremely time consuming, owing to the requirement of considering a range of channel signal-to-noise ratios (SNRs). Therefore, in this paper, we propose the employment of extrinsic information transfer (EXIT) charts to overcome this drawback. Furthermore, EXIT-chart analysis has the additional benefit of offering insights into the specific causes of the performance degradations encountered. Finally, a FP scheme having an overall OW of 6 bits is proposed for the implementation of the min-sum algorithm (MSA).**

I. Introduction

Low-density parity-check (LDPC) codes [1] exhibit an outstanding error correction capability [2], when using the sum-product algorithm (SPA) [3]. However, owing to the high dynamic range of its variables, the optimal SPA requires a floating point implementation, which is associated with a high complexity. Therefore, in order to mitigate this problem, the SPA can be transformed into the logarithmic domain, to yield the 'Log-SPA', which replaces the sum and product operations of the SPA with the lower-complexity 'boxplus' [4] operations to be detailed in Section II and sum operations. Furthermore, by approximating the boxplus operator of the Log-SPA with the minimum-finding operation, the min-sum algorithm (MSA) [5]–[7] achieves a further complexity reduction at the cost of marginally degrading the LDPC performance by a few tenths of a dB [8]. Both the Log-SPA and MSA operate on the basis of logarithmic likelihood ratios (LLRs), which have a low dynamic range and can be represented by fixed point (FP) operands, having a low operand width (OW).

In a hardware implementation, it is this OW that determines both the size of the memory, as well as the area of the data path and hence the energy consumption imposed. It is therefore desirable to use a low OW. However if this is set too low, then the decoder will suffer from a performance degradation, owing to the detrimental effects of the limited fine-resolution and dynamic-range on the represented LLRs. Hence the authors of [7]–[13] attempted to determine the minimum OW that does not impose a significant performance degradation. However, owing to the different algorithms and techniques that they considered, these efforts reached differing conclusions, as summarized in Table I. This problem is further aggravated by the need to consider diverse combinations of various parameters, such as the LDPC code length, the check and variable node degrees, the number of decoding iterations performed and the channel conditions. We argue that since it is unfeasible to exhaustively search all combinations of these parameters, there is no consensus on the most desirable OWs in the cited studies. For this reason, it is typically necessary to re-evaluate the most desirable OW, whenever a new LDPC code is developed. The efforts of [7]–[13] relied upon bit-error ratio (BER) simulations. However, these simulations are very time consuming, since they have to consider different numbers of decoding iterations, as well as a range of channel signal-to-noise ratios (SNRs). Furthermore, the BER simulations only reveal the magnitude of the performance degradation, without offering any insight into its specific causes, namely whether the dynamic-range or the fine-resolution of the FP operands employed is insufficient, for example.

TABLE I
Summary of previously proposed schemes.

Reference	LDPC code[1]	Decoding algorithm	Scheme[2]
[7]	R and IR	Log-SPA	U-Q 7 bits
		offset-MSA [7]	U-Q 4-6 bits
[8]	R and IR	offset-MSA	U-Q 4-6 bits
[9]	not mentioned	SPA	U-Q 12 bits
		SPA[3]	U-Q 6 bits
[10]	R	offset-MSA	FP(2,3,2/3)[4]
[11]	IR	Log-SPA	FP(4,4,4)
[12]	IR	offset-MSA	NU-Q 5 bits
[13]	IR	offset-MSA	U-Q 6 bits

As a remedy, in this paper, we propose the employment of extrinsic information transfer (EXIT) charts [14] to investigate the FP implementation of LDPC decoders. This semi-analytical approach has the advantage of avoiding the requirement of iterative decoding. Furthermore, EXIT chart investigations are independent of both the LDPC code length and the number of decoding iterations performed. As a result, our approach significantly reduces the simulation time required to comprehensively investigate the desirable OW of the LLRs, when a new LDPC code is developed. Additionally, we will demonstrate that EXIT charts offer insights into the specific causes of performance degradation, which are unavailable when using BER simulations [15].

We commence in Section II by detailing the LDPC decoding algorithm and the FP representation considered. Our EXIT chart simulations are described in Section III and the process of analysing the results is detailed in Section IV. Finally, Section V will conclude that an overall LLR OW of 6 is the minimum value that avoids imposing a significant performance degradation in the particular LDPC family considered.

II. Fixed Point Decoder and Log-SPA Operation

As shown in the transmitter schematic of Figure 1, the LDPC encoder is employed to transform the sequence of $(N-M)$ information bits $\mathbf{u}$ into the N-bit encoded sequence $\mathbf{c}$, which is transmitted

[1]Here, 'R' represents regular LDPC codes constructed from a random unstructured Parity Check Matrix (PCM), which was designed to avoid from length-4 cycles. Meanwhile 'IR' represents corresponding irregular LDPC codes.

[2]Here, 'U-Q' represents a uniform quantization of the LLRs in the Log-SPA and MSA, or of the Likelihood Ratios (LRs) of the SPA. Meanwhile, 'NU-Q' represents a non-uniform quantization. We define the notation $\mathrm{FP}(x,y,z)$ in Section II.

[3]In fact, [9] adopts the "parity likelihood ratio algorithm", which is equivalent to the SPA.

[4]In this scheme, the fraction part of the FP representation is reduced from $z=3$ to $z=2$ when the integer part is clipped.

978-1-4244-8325-9/11/$26.00 ©2011 IEEE

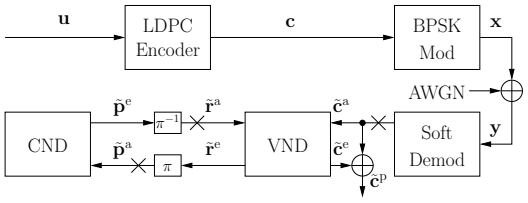


Fig. 1. A schematic of an LDPC coding scheme, consisting of an encoder and an iterative decoder .

through an additive white Gaussian noise (AWGN) channel. In the receiver of Figure 1, the LDPC decoder may be considered to be an iteratively-operated serial concatenation of two decoders that are separated by an interleaver π of length D [14]. More specifically, the variable node decoder (VND) comprises N variable nodes with the same degree[5] of d_{v}, while the check node decoder (CND) comprises M check nodes with degree d_{c}, where we have $Md_{\mathrm{c}} = Nd_{\mathrm{v}} = D$. Note that the soft demodulator provides the decoder with LLR sequences represented by FP numbers. In each decoding iteration, the VND uses the forward-backward algorithm (FBA) [7], [8] to convert the *a priori* LLR sequences $\tilde{\mathbf{c}}^{\mathrm{a}}$ and $\tilde{\mathbf{r}}^{\mathrm{a}}$ into the extrinsic LLR sequence $\tilde{\mathbf{r}}^{\mathrm{e}}$. Meanwhile, the CND uses the FBA to convert $\tilde{\mathbf{p}}^{\mathrm{a}}$ into $\tilde{\mathbf{p}}^{\mathrm{e}}$. While the VND FBA generates extrinsic LLRs by summing *a priori* LLRs, the CND FBA combines LLRs using the boxplus operator [4], according to

$$\begin{aligned} \tilde{a} \boxplus \tilde{b} &= \operatorname{sign}(\tilde{a})\operatorname{sign}(\tilde{b}) \min(|\tilde{a}|, |\tilde{b}|) \\ &\quad + \log\left(1 + e^{-|\tilde{a}+\tilde{b}|}\right) - \log\left(1 + e^{-|\tilde{a}-\tilde{b}|}\right) \quad (1) \\ &\approx \operatorname{sign}(\tilde{a})\operatorname{sign}(\tilde{b}) \min(|\tilde{a}|, |\tilde{b}|). \quad (2) \end{aligned}$$

Note that while the CND employs the boxplus operator of (1) in a Log-SPA LDPC decoder, the CND of an MSA LDPC decoder approximates this using the minimum-finding operator of (2).

In an FP implementation of an Log-SPA or MSA LDPC decoder, a two's complement representation [16] may be adopted for the LLRs. This binary representation consists of an operand's fraction part comprising z bits and an integer part comprising y bits, the most significant of which is used as the sign bit. Here, it is the OW y that dictates the dynamic-range of the FP representation, while its fine-resolution is dictated by the OW z. In order to avoid the overflow that would eventually occur by repeatedly incrementing a two's complement number, saturation is employed at a given maximum during the sum and boxplus operations. Furthermore, the integer part of the *a priori* LLRs is clipped to values in the range that could be represented if only $x \leq y$ bits were employed for their representation, as indicated by the $\times$ symbols in Figure 1. For the remainder of the paper, we adopt the notation $\mathrm{FP}(x, y, z)$ to define the OWs of particular FP representations.

Note that while the terms of the form $\log(1+e^{-\tilde{c}})$ can be computed accurately in a floating point LDPC decoder, FP decoders typically rely on a look-up table (LUT) for these computations. Since $\log(1 + e^{-\tilde{c}}) < 1$ for $\tilde{c} > 0$, it is the OW z of the fraction part in the two's complement representation that dictates the design of the LUT. More specifically, Figure 2 illustrates the values from the set $\{1, 2, 3, \ldots\} \cdot 2^{-z}$ that most closely approximate $\log(1 + e^{-\tilde{c}})$ for various values of $\tilde{c}$, which are also selected from that set.

[5]The extension of our proposed methods to irregular LDPC codes is trivial.

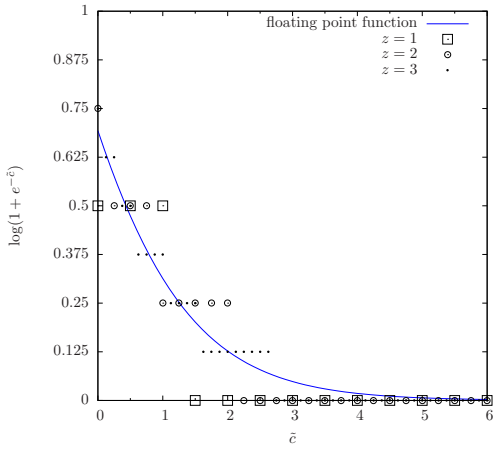


Fig. 2. The approximation of the function $\log(1 + e^{-\tilde{c}})$ using a LUT with limited fraction OWs.

III. EXIT Chart Analysis

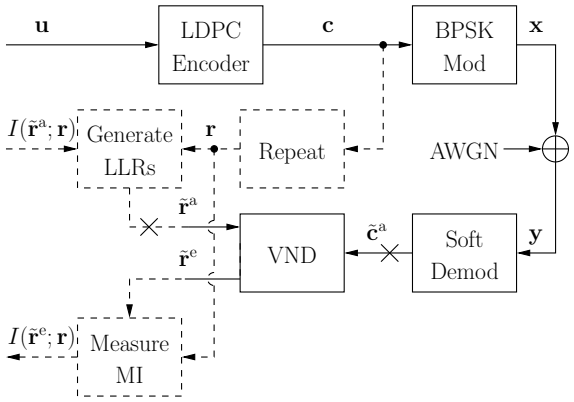
The iterative exchange of increasingly more reliable extrinsic information between the VND and CND can be characterised using EXIT charts. These employ the mutual information (MI) $I(\tilde{\mathbf{x}}; \mathbf{x})$ to quantify the reliability of the information in the LLR sequence $\tilde{\mathbf{x}}$ about the contents of the bit sequence $\mathbf{x}$. Here, the MIs have values in the range of $[0, 1]$, where '0' indicates that the LLRs contain no information about the corresponding bits, while '1' indicates perfect MI. More specifically, an EXIT function plots the MI of the extrinsic LLRs $\tilde{\mathbf{x}}^{\mathrm{e}}$ that are output by a decoder as a function of the MI of the corresponding input *a priori* LLRs.

Figure 3(a) shows how the schematic of Figure 1 may be adapted to plot the VND EXIT function, which is exemplified in Figure 4(a). More specifically, as the dashed boxes and paths of Figure 3(a) show, each bit of the encoded sequence $\mathbf{c} = [c_1, c_2, \cdots, c_N]$ is repeated d_{v} times to form the longer D-bit sequence $\mathbf{r}$, where we have $r_{(n-1)\cdot d_{\mathrm{v}}+1} = r_{(n-1)\cdot d_{\mathrm{v}}+2} = \cdots = r_{n\cdot d_{\mathrm{v}}} = c_n$ for the n^{th} variable node[6]. The EXIT chart simulation of Figure 3(a) generates Gaussian-distributed *a priori* LLR sequences $\tilde{\mathbf{r}}^{\mathrm{a}}$ having MIs $I(\tilde{\mathbf{r}}^{\mathrm{a}}; \mathbf{r})$ across the range of $[0, 1]$ [14, (18)]. Following the operation of the VND, the MI $I(\tilde{\mathbf{r}}^{\mathrm{e}}; \mathbf{r})$ of the extrinsic LLR sequence $\tilde{\mathbf{r}}^{\mathrm{e}}$ is measured using the MI histogram based method [7] [14, Equation (19)]. Figure 4(a) plots $I(\tilde{\mathbf{r}}^{\mathrm{e}}; \mathbf{r})$ as a function of $I(\tilde{\mathbf{r}}^{\mathrm{a}}; \mathbf{r})$, when using the floating point Log-SPA or MSA for variable node degrees in the set $d_{\mathrm{v}} \in \{2, 4, 8, 16\}$. Note that the Log-SPA and MSA have identical VND's EXIT functions, since the distinction between these algorithms affects only the CND operation. Furthermore, since the simulation based investigations outlined in Figure 3(a) consider the AWGN channel, the results of Figure 4(a) are specific to an AWGN channel E_b/N_0 of 3 dB. Higher or lower E_b/N_0 values would move the EXIT functions upwards or downwards, respectively.

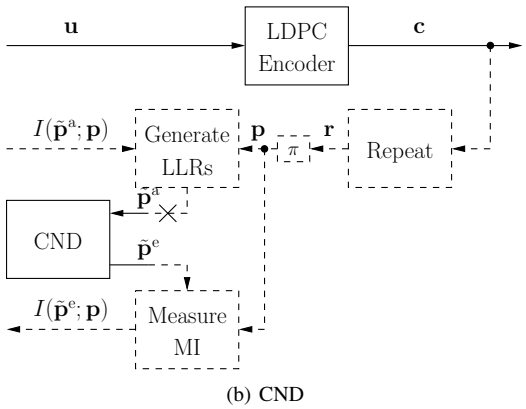
Similarly, the investigation detailed in Figure 3(b) are used to plot the EXIT function of the CND, as exemplified in Figure 4(b). Here, the dashed-box interleaver π permutes the bit sequence $\mathbf{r}$, in order to obtain the Md_{c}-bit sequence $\mathbf{p}$. Figure 4(b) plots $I(\tilde{\mathbf{p}}^{\mathrm{e}}; \mathbf{p})$ as a function of $I(\tilde{\mathbf{p}}^{\mathrm{a}}; \mathbf{p})$, when using the floating point Log-SPA or MSA for check node degrees in the set $d_{\mathrm{c}} \in \{4, 8, 16, 32\}$. Observe

[6]The extension to an irregular LDPC code is trivial.

[7]Note that unlike the averaging method of [17, Equation (8)], the histogram based method of evaluating the MI can accurately quantify the degradation imposed by sub-optimal decoding algorithms, which produce LLRs that do not satisfy the consistency condition [17].



(a) VND



(b) CND

Fig. 3. Adapted schematics used to depict the generations of EXIT functions for the (a) VND, (b) CND.

that the Log-SPA and MSA have different EXIT functions for the CND, since the former operates on the basis of (1), while the latter employs (2). More specifically, the MSA CND EXIT function can be seen to achieve extrinsic MIs that are almost 0.005 lower than these of the Log-SPA. Note however that it is conventional to plot these EXIT functions on inverted axes [14]. Also note that the CND EXIT function is independent of the channel's E_b/N_0 value.

As described in [14], the VND's and CND's EXIT function may be plotted in the same figure to obtain an EXIT chart. When the channel SNR is sufficiently high, the VND's EXIT function will be above the CND's EXIT function, and they will not intersect before reaching the $(1, 1)$ point of perfect convergence to the minimum attainable BER. The emergence of the resultant so-called open EXIT tunnel implies that iterative decoding convergence towards a minimum BER may indeed be achieved. Observe in Figure 4(b) that the EXIT function values corresponding to the sub-optimal MSA are slightly higher than those corresponding to the optimal Log-SPA. This implies that a slightly higher channel SNR is required to create an open EXIT tunnel, when the MSA is employed. These results demonstrate that the performance degradation imposed by sub-optimal decoding algorithms can be investigated by considering the quantization-induced narrowing of EXIT tunnels. In the next section, we use this technique to identify desirable OWs for FP implementations of LDPC codes. Note that the vertical bars in Figure 4, Figure 5 and Figure 6 indicate the spread of the EXIT bands [18], when employing an interleaver length of D. These bands allow the EXIT chart to accurately characterize the iterative decoding process, even when very short interleaver lengths are employed, in which case the spread of the bands is increased, but the EXIT functions remain unchanged.

IV. Results and Discussions

In this section, we determine the minimum required OWs for FP implementations of LDPC codes in a systematic manner. We commence by using EXIT chart simulations to determine a desirable value for the fraction OW z in isolation. Next, we consider the clipped integer OW x in isolation, before jointly considering z, x and the integer OW y. We shall consider the degradations imposed by limited OWs to be acceptable, provided that they are less than the difference between the CND's EXIT functions obtained using the MSA and Log-SPA, namely a maximum MI degradation of 0.005. Note that in each case, we found that the spread of the EXIT bands was unaffected by the OW and that they accurately predicted the path of the iterative decoding trajectories, even when employing short interleaver lengths.

A. Fraction Operand Width

Figure 4 demonstrates the effect of employing fraction OWs in the set $z \in \{0, 1, 2, 3, 4, 5\}$ upon the VND's and CND's EXIT functions. Here, effectively infinite values have been selected for the integer OWs x and y, as implied by the notation $\mathrm{FP}(\infty, \infty, z)$. This allows us to consider the effect of the fraction OW in isolation.

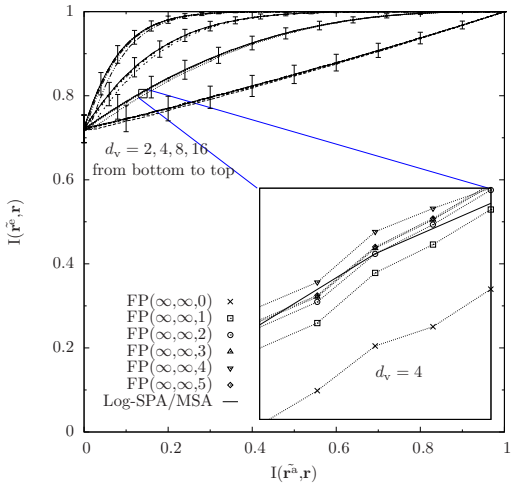
Figure 4 shows that the gradually reduced fraction OW z bends the CND's and VND's EXIT curves towards each other, narrowing the tunnel between them. Due to the narrowed tunnel, more iterations would be required for the decoder to converge towards the minimal attainable BER. The results also show that the degradation imposed varies only slightly with the node degrees d_{v} and d_{c}.

The CND results of Figure 4(b) demonstrate that the degradation imposed by a fraction OW of $z \geq 2$ is less than that imposed by the MSA. Likewise, $z \geq 1$ imposes only a modest degradation for the VND. In order to simplify the LDPC decoder's architecture, it is desirable to adopt the same fraction OW z in both the VND and CND. For this reason, we recommend a fraction OW of $z = 2$ for striking an attractive trade-off between the complexity imposed and the performance attained.

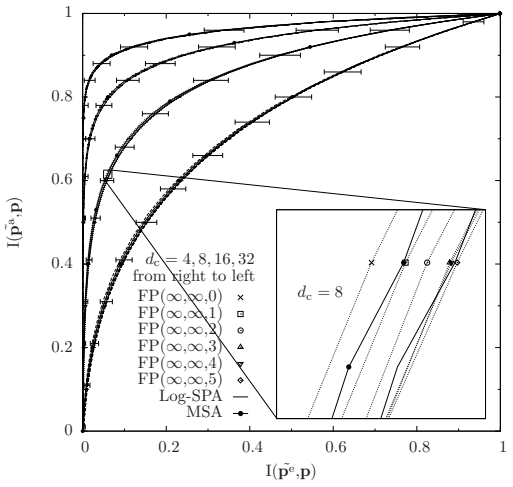
B. Clipping Range

The effect of employing clipping OWs in the set $x \in \{1, 2, 3, 4, 5, 6\}$ upon the VND's and CND's EXIT functions is considered in Figure 5. Here, the notation $\mathrm{FP}(x, \infty, \infty)$ is employed to show that effectively infinite values have been selected for the integer and fraction OWs y and z, respectively. In this way, the effect of the clipping OW is considered in isolation.

Figure 5 shows that for both the VND and CND, a higher degradation is imposed by lower clipping OWs x, which may indeed be expected owing to the reduced range that this implies. As shown in Figure 5(b), the degradation imposed on the CND EXIT functions is similar to that caused by employing a limited fraction OW z, but the effect is more significant. In the case of the VND, the degradation causes a droop in the EXIT functions for low *a priori* MIs of $I(\tilde{\mathbf{r}}^{\mathrm{a}};\mathbf{r}) < 0.1$, when $x \in \{1, 2\}$. Briefly, this may be explained by the fact that $x \in \{1, 2\}$ causes most *a priori* LLRs to become saturated, effectively transforming the VND into a hard-decision decoder. While Figure 5(a) shows that the shape of the droop is affected by the variable node degree d_{v}, nevertheless the magnitude of the droop remains largely unaffected. Note that for $d_{\mathrm{v}} = 2$ and $x = \{1, 2\}$, the VND's EXIT function fails to reach the $(1, 1)$ point at the top-right corner of the EXIT chart. As a result, the EXIT-tunnel would never become open, regardless of how high the channel E_b/N_0 value is, hence preventing iterative decoding convergence to the minimum attainable BER.

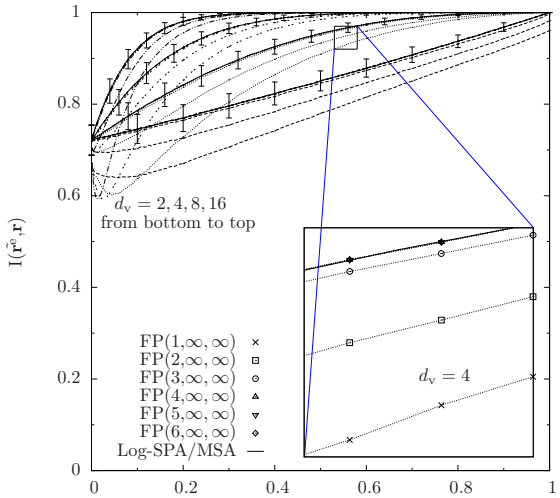


(a) VND

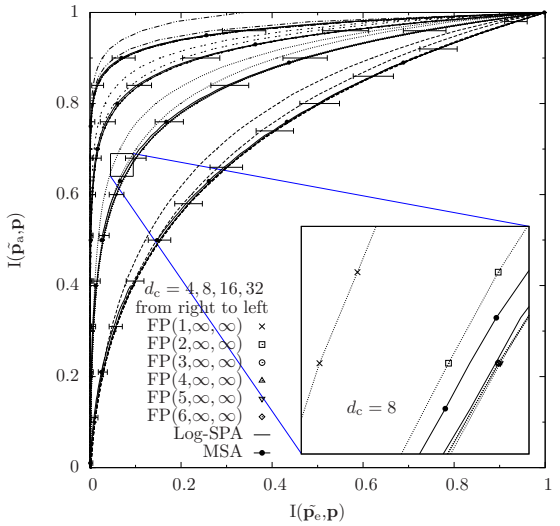


(b) CND

Fig. 4. The EXIT functions for FP implementations of LDPC codes employing various fraction OWs z, as well as various variable and check node degrees, for communication over an AWGN channel having a E_b/N_0 of 3 dB.



(a) VND



(b) CND

Fig. 5. The EXIT functions for FP implementations of LDPC codes employing various clipped integer OWs x, as well as various variable and check node degrees, for communication over an AWGN channel having a E_b/N_0 of 3 dB.

Based on both the VND's and CND's results of Figure 5, to keep the tunnel open until the top-right point is reached and also to ensure that the degradation imposed by a clipping OW is less than that imposed by the MSA, we suggest that an attractive trade-off between the complexity imposed and the performance attained may be struck by using $x = 3$.

C. Integer Operand Width

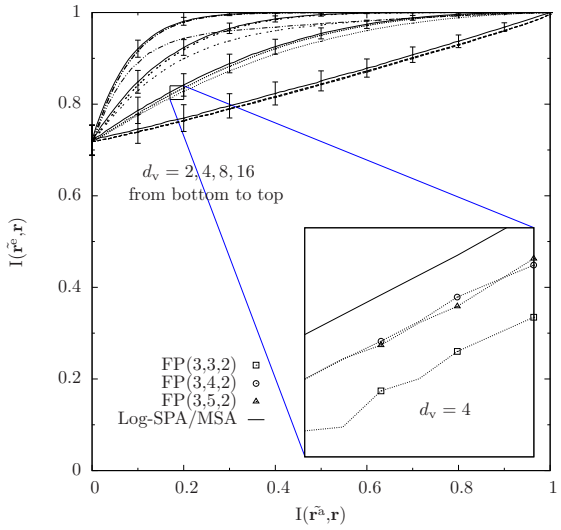
Let us now combine the conclusions of Sections IV-A and IV-B with the consideration of the integer OW y. More specifically, we adopt the values of $x = 3$ and $z = 2$, while considering integer OWs from the set $y \in \{3, 4, 5\}$.

For both the VND and CND, Figure 6 shows that a more severe degradation is imposed by lower integer OWs y, which is not unexpected owing to the reduced range that this implies. As in the case of limited fraction OWs, this degradation is manifested as a slight EXIT-tunnel narrowing across the entire range of MIs. However, unlike for limited fraction OWs, Figure 6(a) shows that the degradation is more pronounced for higher variable node degrees d_v.

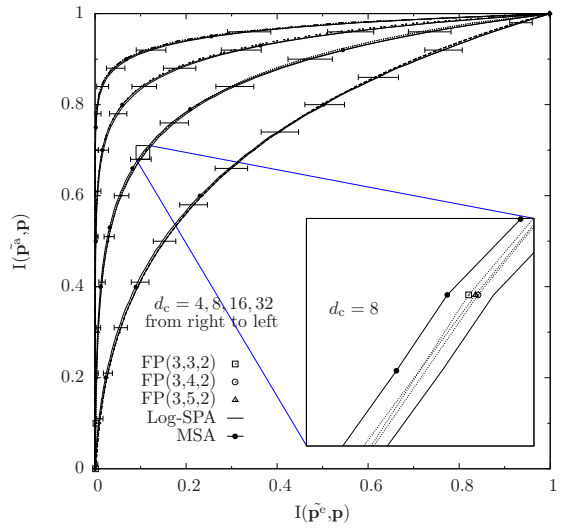
Regardless of the node degrees, it can be seen for both the VND and CND that an integer OW of $y = 4$ is sufficient to avoid a degradation that is significantly higher than that imposed by the MSA. For this reason, we conclude that the $\mathrm{FP}(3, 4, 2)$ representation facilitates LDPC implementations that strike the most desirable trade-off between the complexity imposed and the performance attained.

V. Conclusions

In this paper, we have proposed an EXIT chart based method to investigate desirable parameters of FP implementations of LDPC codes, in order to strike an attractive trade-off between their complexity and performance. Our technique may also be employed to investigate this trade-off in other reduced-complexity versions of LDPC codes. Since this technique does not require the simulation of iterative decoding at different number of iterations or at multiple E_b/N_0 values, it is significantly less time consuming than the conventional BER-based method. For example, when employing a half-rate length-500 regular LDPC code having $d_v = 3$ and $\mathrm{FP}(3, 4, 2)$, a total of 8×10^8 VND



(a) VND



(b) CND

Fig. 6. The EXIT functions for FP implementations of LDPC codes employing various integer OWs y, as well as various variable and check node degrees, for communication over an AWGN channel having a E_b/N_0 of 3 dB.

operations and 6×10^8 CND operations are required, to obtain the CND's and VND's EXIT functions, respectively. Here, we consider 2000 frames for each of 100 *a priori* MI values in the range of $[0, 1]$. However, a BER simulation for the same LDPC code requires approximately 10^{11} VND operations and 10^{11} CND operations to consider 10 different E_b/N_0 values in the range of $[0, 4.5]$ dB. Here the decoding iterations are continued only until convergence is achieved. Therefore, the resultant factor of 100 complexity reduction allows comprehensive investigations into the effects of multiple parameters to be conducted. This is particularly beneficial, since it is typically necessary to investigate the most desirable OWs whenever a new LDPC code is designed. Furthermore, EXIT chart analysis can offer insights into the causes of the performance degradation that is imposed by reduced-complexity versions of LDPC codes. For example, the analysis of Section IV revealed that a droop in the VND EXIT function implies that the *a priori* LLRs are excessively clipped.

In Section IV, we proposed a systematic approach for determining the desirable OWs for FP implementations of LDPC codes. Our results indicate that as a rule of thumb an overall OW of $y + z = 6$ may offer an attractive trade-off between an LDPC code's complexity and performance. As summarized in Table I, this result agrees with the conclusions of [7]–[13], [19], where the slight differences observed may be attributed to the different algorithms and quantization schemes considered.

ACKNOWLEDGEMENT

The financial support of the European Union under the auspices of the Newcom and Phoenix projects, as well as that of the RC-UK under the auspices of the India-UK Advanced Technology Centre known as IN-ATC is gratefully acknowledged.

REFERENCES

[1] R. G. Gallager, "Low density parity check codes," *IRE Transactions on Information Theory*, vol. IT, no. 5, pp. 21–28, 1962.

[2] D. J. C. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 33, no. 6, pp. 457–458, 1997.

[3] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[4] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 429–445, Aug. 2002.

[5] M. P. 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, no. 5, pp. 673–680, Mar. 1999.

[6] E. Eleftheriou, T. Mittelholzer, and A. Dholakia, "Reduced-complexity decoding algorithm for low-density parity-check codes," *Electronics Letters*, vol. 37, no. 2, pp. 102–104, Jan 2001.

[7] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X. Hu, "Reduced-complexity decoding of ldpc codes," *IEEE Transactions on Communications*, vol. 53, no. 8, pp. 1288–1299, 2005.

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

[9] L. Ping and W. K. Leung, "Decoding low density parity check codes with finite quantization bits," *IEEE Communications Letters*, vol. 4, no. 2, pp. 62–64, Feb. 2000.

[10] T. Zhang, Z. Wang, and K. Parhi, "On finite precision implementation of low density parity check codes decoder," in *IEEE International Symposium Circuits and Systems*, vol. 4, May. 2001, pp. 202–205.

[11] G. Masera, F. Quaglio, and F. Vacca, "Finite precision implementation of ldpc decoders," in *Communications, IEE proceedings-*, vol. 152, no. 6, Dec. 2005, pp. 1098–1102.

[12] D. Oh and K. Parhi, "Optimally quantized offset min-sum algorithm for flexible ldpc codes," in *Asilomar Conference Signals, Systems and Computers*, Oct. 2008, pp. 1886–1891.

[13] ——, "Min-sum decoder architecture with reduced word length for ldpc codes," *IEEE Transactions on Circuits and Systems I*, vol. 57, no. 1, pp. 105–115, Jan. 2010.

[14] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, no. 10, pp. 1727–1737, 2001.

[15] L. Li, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, "Design of fixed-point processing based turbo codes using extrinsic information transfer charts," in *IEEE 72nd Vehicular Technology Conference Fall*, Ottawa, Canada, Sep. 2010, pp. 1–5.

[16] G. Montorsi and S. Benedetto, "Design of fixed-point iterative decoders for concatenated codes with interleavers," *IEEE Journal on Selected Areas in Communications*, vol. 19, pp. 871–882, May 2001.

[17] J. Hagenauer, "The turbo principle - Tutorial introduction and state of the art," in *Proc. Int. Symp. on Turbo Codes*, Brest, France, Sep. 1997, pp. 1–11.

[18] J. W. Lee and R. E. Blahut, "Generalized EXIT chart and BER analysis of finite-length turbo codes," in *Proc. IEEE Global Telecommunications Conf.*, vol. 4, San Francisco, CA, USA, Dec. 2003, pp. 2067–2072.

[19] E. Boutillon, C. Douilard, and G. Montorsi, "Iterative decoding of concatenated convolutional codes : Implementation issues," in *Proceedings of the IEEE*, vol. 95, no. 6, Jun. 2007, pp. 1201–1227.