Finite-Length Quasi-Synchronous LDPC Decoders

François Leduc-Primeau, Warren J. Gross

Finite-Length Quasi-Synchronous LDPC Decoders

François Leduc-Primeau and Warren J. Gross
Dept. of Electrical & Computer Engineering, McGill University, Montréal, Canada

***Abstract*—Quasi-synchronous systems aim to reduce energy consumption by allowing timing violations in a synchronous circuit, while performance guarantees are provided by analyzing the system with a suitable deviation model. This paper studies the performance of quasi-synchronous LDPC decoders for regular codes of finite length. We present an approach to accurately predict the decoding performance and energy consumption of the decoder for a specific average channel quality and maximum number of iterations. These analytical results are then compared with gate-level circuit simulations of a quasi-synchronous decoder.**

I. Introduction

The scale of static and dynamic variations in CMOS process technologies is increasing rapidly, with the result that design guardbands put in place to ensure the reliable operation of the circuit are quickly becoming important contributors to the overall delay of the circuit [1]. A possible approach to reducing the guardbands is to tolerate occasional timing violations in the circuit. We call systems that achieve this without a compensation mechanism in hardware *quasi-synchronous* (QS) systems. For this approach to work, it is necessary to be able to accurately characterize the performante of the QS system while accounting for the *deviations* caused by timing violation events. Low-density parity-check (LDPC) decoders are ideal candidate for a QS implementation because only their average error-correction performance is of interest, and because the high level of parallelism of the decoding algorithm could allow tolerating even static variations as long as they are not significantly correlated across processing units.

The behavior of faulty LDPC decoders has attracted numerous contributions. Notable previous works include the analysis in [2] of the Gallager-A and Sum-Product algorithms when messages exchanged in the decoder are affected by noise, and an analysis of faulty finite-alphabet LDPC decoders where deviations are modeled using conditional distributions [3]. However, these analyses use models that are not directly applicable to deviations caused by timing violations. An analysis of the Min-Sum algorithm was performed in [4] for the case where computations are reliable, but messages are stored in an unreliable memory. Finally, a deviation model that takes into account the state dependence of timing violations is proposed in [5] for the special case of binary-output circuits.

In this paper, we use a deviation modeling approach proposed in [6] that can generate accurate memoryless deviation models of QS circuits, and show how this deviation model can be used to predict the performance and energy consumption of a QS decoder operating on a finite-length code. Two approaches can be used to predict the error correction performance. First, the deviation model can be used within a Monte-Carlo simulation based on a high-level software implementation of the algorithm. More interestingly, for moderately long codes, the performance can be predicted from density evolution (DE) results, allowing to determine the performance of a particular finite-length QS decoder very rapidly. As a result, the operating condition can be optimized specifically in terms of the chosen code length, signal-to-noise ratio (SNR), and maximum number of iterations.

II. Decoder Architecture

In this paper, we restrict our attention to regular LDPC codes, in which all variable nodes have degree d_v, and all check nodes have degree d_c. The decoder that we consider implements the Offset Min-Sum (OMS) algorithm, a well known approximation to the Sum-Product algorithm that achieves similar performance. An LDPC decoder proceeds by exchanging belief messages between the variable and check nodes of the code's Tanner graph, and a decoder implementation can be divided into variable-node processor (VNP) blocks that implement the variable node (VN) computation, and check-node processor (CNP) blocks that implement the check node (CN) computation. We implement a row-layered decoder architecture, in which all the messages sent by a given CN are computed in parallel, but VN messages are generated one edge at a time. By using this architecture with the corresponding row-layered message-passing schedule, the number of decoding iterations can be reduced approximately by a factor of two [7], and for this reason it is widely used for circuit implementations (e.g. [8]).

One CNP block can be combined with d_c VNP blocks to form a processor unit. Each VNP block in the unit takes as input the VN total belief Λ_i, as well as the intrinsic belief λ_i^ℓ, which corresponds to the message that was previously received by the VN on edge $\ell \in \{1, 2, \ldots, d_v\}$ (note that $\Lambda_i = \sum_\ell \lambda_i^\ell$). After sending messages to the CNP and receiving messages back, each VNP outputs the updated values of Λ_i and λ_i^ℓ. This processor can be pipelined, and in this architecture we divide it into two pipeline stages. The complete architecture is composed of a number of processor units and of a memory.

The values Λ_i and λ_i^ℓ are represented using a fixed-point representation. When using the OMS algorithm, these values can be thought of as integers without loss of generality. The belief totals Λ_i are initialized using the channel outputs. We assume that codewords are transmitted over an Additive White Gaussian Noise (AWGN) channel. For a channel output y_i and channel noise variance σ_w^2, the initial belief total $\Lambda_i^{(0)}$ is given by rounding

$$\Lambda_i^{(0)} = \frac{\alpha y_i}{\sigma_w^2} \tag{1}$$

to the nearest integer. The number of bits used to represent the messages, the value of α, and the offset parameter of the OMS algorithm are chosen based DE results obtained for a reliable decoder (these values are given in Section IV).

Note that to generate the circuit results in Section IV, we actually use a decoder containing a single processor unit, since the number of processors has no impact on decoding performance or energy consumption. Since we focus on the energy required by the computations, we assume that the memory is reliable and consumes no power.

III. Performance Analysis

A. Deviation Model

The signal propagation delays in CMOS circuits vary according to many factors, but in this paper we focus on the delay variations associated with the state of the circuit, and assume that the operating condition condition of the circuit, denoted γ, is deterministic. The fact that propagation delays depend on the state of the circuit makes it challenging to model deviations caused by timing violations, since deviations depend not only on the circuit's current input but also on the previous one.

We will model deviations occurring in a QS LDPC decoder by using the approach proposed in [6], where the dependence on the circuit's state is replaced with a dependence on the statistical distribution of the input, yielding a memoryless model that is nonetheless accurate. Since we are usually interested in the progress made by the decoder from one iteration to the next, we can summarize the effect of deviations on the decoder by considering that every message $\nu_{i,j}^{(t)}$ exchanged from a variable node i to a check node j in iteration t is replaced by a faulty message $\mu_{i,j}^{(t)}$. We then define a family of conditional distributions of $\mu_{i,j}^{(t)}$ given $\nu_{i,j}^{(t)}$, indexed by the operating condition γ, and by the message error rate at the beginning of the iteration $p_e^{(t-1)}$. To further improve the model's accuracy, we also condition the distributions on the value of the transmitted bit x_i corresponding to variable node i. Omitting the node indices i and j, the model can be denoted as

$$\mathbb{P}^{(\gamma,p_e^{(t-1)})}_{\mu^{(t)}|\nu^{(t)},x}\left(\mu \mid \nu, x\right). \tag{2}$$

The model is said to be *weakly symmetric* if

$$\mathbb{P}^{(\gamma,p_e^{(t-1)})}_{\mu^{(t)}|\nu^{(t)},x}\left(\mu \mid \nu, x=1\right) = \mathbb{P}^{(\gamma,p_e^{(t-1)})}_{\mu^{(t)}|\nu^{(t)},x}\left(-\mu \mid -\nu, x=-1\right)$$

holds for any given realization of the channel noise, and in that case it is shown in [6] that DE can be used to determine the performance of the QS decoder when the code length tends to infinity.

The deviation model in (2) describes the iterative behavior of the decoder, but when studying the decoding of a finite-length code, we are also interested in measuring the error rate at the output of the decoder, which depends on the a-posteriori estimate represented by the VN total Λ_i. To do this, we define a second deviation model similar to (2) but that provides the conditional distribution of the faulty belief total in terms of the ideal one, that is in terms of the value that would be computed if the last decoding iteration had been reliable. Denoting the ideal belief total of some VN after iteration t by $K^{(t)}$, the output deviation model is given by

$$\mathbb{P}^{(\gamma,p_e^{(t-1)})}_{\Lambda^{(t)}|K^{(t)},x}\left(\Lambda \mid K, x\right). \tag{3}$$

Note that the output deviation model could be simplified further since in most cases the output is only used to take a binary decision on the value of the transmitted bit.

B. Deviation and Energy Measurements

In order to quickly cover the range of possible input distributions that will be seen by the circuit, we approximate the distribution of extrinsic messages in the decoder as a one-dimensional (1-D) Normal distribution, that is a Normal distribution with a mean μ and variance σ^2 such that $\mu = \alpha\sigma^2$, where α is the constant in (1). Note that this approximation is only used to measure deviations, and not to measure the error rate of the decoder. The 1-D Normal distribution can be equivalently described in terms of its error rate parameter $p_e = \frac{1}{2}\operatorname{erfc}\left(\frac{1}{\sqrt{2\sigma^2}}\right)$. We select a small set of p_e values, and for each value, measure the conditional distributions in (2) and (3) by performing Monte-Carlo simulations on the gate-level representation of the processing circuit. This is repeated for every operating condition γ of interest to construct the complete deviation model. Simultaneously, we also record the switching activity in the circuit in terms of p_e, in order to estimate the circuit's power consumption and construct an energy model that depends on p_e.

Even though the deviation measurements are performed by assuming a specific noise variance σ_w^2 for the AWGN channel, the deviation model does not depend on σ_w^2, and we have observed that the model remains accurate for other variance values.

C. Density Evolution

Since messages in the decoder are quantized, the error correction performance of the decoder on a cycle-free graph can be evaluated exactly by performing DE with discrete distributions [9]. We perform DE in the usual way, but we take into account the layered message-passing schedule used by the hardware implementation, and apply an additional transformation on the distribution of VN-to-CN messages according to (2). For each decoding iteration, we also compute the distribution of the VN total belief Λ_i. Note that Λ_i is obtained by summing d_v CN-to-VN messages together (rather than $d_v - 1$ in the case of extrinsic messages). After evaluating the distribution of the ideal total, we apply (3) to obtain the distribution of Λ_i.

D. Finite-length Performance

The performance of a finite-length LDPC code differs from the performance given by performing DE on a cycle-free graph for two reasons. First, good LDPC codes necessarily contain cycles [10], which introduce harmful correlation among the messages exchanged by a belief propagation decoder. Second, because the codewords have a finite length, the channel noise realization associated with a particular transmitted frame has a varying distribution. We now discuss how the error-correction performance and energy results obtained by using DE with the

deviation model can be used to determine the performance and energy consumption of a quasi-synchronous decoder operating on a a finite-length code, in the case of a transmission through the AWGN channel.

To illustrate the variations in channel noise, it is helpful to first consider a binary symmetric channel with parameter p_o. In a codeword of infinite length, a fraction p_o of the transmitted bits will be received in error, with probability 1. However, in a finite-length codeword, the number of incorrect bits in a codeword will vary according to a Binomial distribution. Denoting by p_{obs} the fraction of incorrect bits observed in the codeword, the *observed* channel can be described by a binary symmetric channel (BSC) with parameter p_{obs}. For a code of length N, this parameter has the distribution

$$\mathbb{P}_{p_{\text{obs}}}(p_{\text{obs}}) = \binom{N}{Np_{\text{obs}}} p_o^{Np_{\text{obs}}}(1-p_o)^{N-Np_{\text{obs}}}. \quad (4)$$

For large N, this distribution can be approximated by a Gaussian distribution with mean p_o and variance $p_o(1-p_o)/N$.

In the case of the AWGN channel, the observed channel cannot be described by a single parameter. Nonetheless, it is shown in [11] that approximating the noise realization by a 1-D Gaussian distribution with error-rate parameter p_{obs} leads to accurate predictions of the performance of moderately long codes. This is also corroborated by our own results presented in Section IV.

Since the decoding performance concentrates to the cycle-free case as the code length goes to infinity, we can expect the cycles to have a smaller impact on performance as the code length increases. In fact, it was shown in [11] that the frame and bit error rates of moderately long codes can be determined accurately based only on the channel threshold associated with the code ensemble and with the decoding algorithm.

The results presented in [11] apply to the case where the number of decoding iterations is large. We present a similar method that is able to predict the decoding performance for any number of decoding iterations. Being able to predict the bit and frame error rate as a function of the number of iterations is important to explore tradeoffs involving latency. In addition, predicting the frame error rate as a function of decoding iterations will allow a precise evaluation of the energy consumption of the decoder.

Let us denote by $P^{(t)}_{\gamma,N}(p_o)$ the output bit error rate of the decoder after t iterations, when using a code of length N and an operating parameter γ, and when the average channel error rate is p_o. Similarly, let $P^{(t)}_{\gamma,\infty}(p_o)$ denote the output bit error rate after t iterations for an infinite-length code, which can be obtained using DE. Recall that in this case, the observed channel has parameter $p_{\text{obs}} = p_o$ with probability 1. If we ignore the effect of cycles, the distinction between $P^{(t)}_{\gamma,N}(p_o)$ and $P^{(t)}_{\gamma,\infty}(p_o)$ is only due to the variability of the observed channel. Therefore, we can express the finite-length bit-error rate (BER) as

$$P^{(t)}_{\gamma,N}(p_o) = \int_0^{\frac{1}{2}} P^{(t)}_{\gamma,\infty}(p_{\text{obs}})\, \phi_{\mathcal{N}}\left(p_{\text{obs}};\, p_o, \frac{p_o(1-p_o)}{N}\right) \mathrm{d}p_{\text{obs}}, \quad (5)$$

where $\phi_{\mathcal{N}}(x;\mu,\sigma^2)$ is the probability density function of a normal random variable with mean μ and variance σ^2. The function $P^{(t)}_{\gamma,\infty}(p_o)$ can be evaluated at discrete intervals using DE, and linear interpolation used to construct a continuous function. Since a single deviation model is valid for all p_{obs} values of interest, no additional circuit simulation is required.

The frame-error rate (FER) $R^{(t)}_{\gamma,N}(p_o)$ can be evaluated in a similar way. In a cycle-free graph and for a given received frame with an observed noise distribution of p_{obs}, a frame remains in error if at least one bit is estimated incorrectly. Therefore, when the channel noise corresponds to an error rate of exactly p_{obs}, the FER $R^{(t)}_{\gamma,\infty}(p_{\text{obs}})$ is given by

$$R^{(t)}_{\gamma,\infty}(p_{\text{obs}}) = 1 - \left(1 - P^{(t)}_{\gamma,\infty}(p_{\text{obs}})\right)^N. \quad (6)$$

In the finite-length case, we take the expectation over the possible channel noise realizations, and obtain

$$R^{(t)}_{\gamma,N}(p_o) = \int_0^{\frac{1}{2}} R^{(t)}_{\gamma,\infty}(p_{\text{obs}})\, \phi_{\mathcal{N}}\left(p_{\text{obs}};\, p_o, \frac{p_o(1-p_o)}{N}\right) \mathrm{d}p_{\text{obs}}. \quad (7)$$

The energy consumption of the decoder during iteration $t+1$ is modeled as a function of the message error rate $p_e^{(t)}$ and of the current operating condition γ. Since $p_e^{(t)}$ is fully determined by the channel distribution p_{obs}, by γ, and by t, we can also express the energy as a function of these quantities. We denote the energy consumption of iteration t for a fixed observed channel p_{obs} by $E^{(t)}_{\gamma,\infty}(p_{\text{obs}})$. If we assume that the decoder is terminated as soon as a valid codeword is found, the decoding energy required to decode a particular frame depends on the number of iterations that was required for that frame. The average energy consumed by the decoder to decode each codeword is therefore

$$E_{\gamma,N}(p_o) = \int_0^{\frac{1}{2}} \left(\sum_{t=1}^{L} E^{(t)}_{\gamma,\infty}(p_{\text{obs}}) R^{(t-1)}_{\gamma,\infty}(p_{\text{obs}})\right) \cdot \phi_{\mathcal{N}}\left(p_{\text{obs}};\, p_o, \frac{p_o(1-p_o)}{N}\right) \mathrm{d}p_{\text{obs}}, \quad (8)$$

where L is the maximum number of decoding iterations. If the latency of one iteration is T_γ and is constant throughout the decoding, we can also obtain the average energy-delay product (EDP) metric of the decoder as

$$\text{EDP}_{\gamma,N}(p_o) = E_{\gamma,N}(p_o) \cdot T_\gamma\ . \quad (9)$$

If the latency is not constant, it simply appears inside the sum in (8).

E. Operating Condition Selection

In CMOS circuits, there exists a tradeoff between energy consumption and latency. For this reason, it is interesting to seek to minimize the product of energy and delay, or EDP. The EDP objective provides more flexibility in finding solutions because the clock period of the circuit can be chosen arbitrarily, whereas the supply voltage is constrained to the points for which the standard cell libraries have been characterized.

The EDP metric of the decoder varies in terms of the channel quality. Since a finite-length code of reasonable length

cannot achieve a low error rate near the ensemble threshold, the optimization should be performed at the SNR where the decoder is likely to be operated. We choose an SNR of 1.7 dB, which corresponds to a channel output error rate of $p_o = 0.112$.

When the decoder is to be operated at a single operating condition γ, one way to minimize EDP is simply to evaluate (9) for every possible γ. Using MATLAB, (8) can be evaluated in a few seconds on a laptop CPU.

IV. Results

A. Experimental Setup

We construct a $(3, 6)$ code of length $N = 32000$ using a random construction free of 4-cycles. The code was generated using MacKay's freely available `code5` generator [12]. By studying the performance of a reliable implementation of the decoder, we set α in (1) to 4 and the OMS correction offset to 1, while the number of quantization bits used to represent messages is set to 6 in order to obtain decoding performance similar to a floating-point implementation.

The processing unit of the decoder is synthesized using `Cadence Encounter` [13] to TSMC's 65 nm process. The synthesis uses the *tcbn65gplus* cell library and is performed at the default supply voltage of $V_{\mathrm{dd}} = 1\,\mathrm{V}$, with a target clock period of $2\,\mathrm{ns}$. After synthesis, we generate timing annotations for the circuit at several V_{dd} values, in increments of $0.05\,\mathrm{V}$. A testbench then instantiates the gate-level version of the processing unit. The testbench connects the processor with the memory, and implements an early termination mechanism that stops the decoder as soon as a valid codeword is found. The early termination mechanism itself is not affected by timing violations. The received frames that are processed by the decoder are generated by transmitting random codewords modulated with binary phase-shift keying through an AWGN channel.

In addition to the simulation of the decoder circuit, the performance of the QS decoder can be evaluated by building the deviation model into a high-level software implementation of the decoding algorithm. Deviations applied to VN-to-CN messages are sampled from (2), and deviations applied to the VN total beliefs are sampled from (3). At every decoding iteration, the message error rate parameter $p_e^{(t-1)}$ of the deviation model is updated to track the error rate progress of the particular frame being decoded. Note that unlike the predictions based on DE, the high-level Monte-Carlo approach is expected to be accurate for any code length.

B. Discussion

We present the bit and frame error rate of the decoder operated at $(V_{\mathrm{dd}} = 0.85\,\mathrm{V}, T_{\mathrm{clk}} = 2.2\,\mathrm{ns})$ in Fig. 1, and at $(V_{\mathrm{dd}} = 0.85\,\mathrm{V}, T_{\mathrm{clk}} = 2.3\,\mathrm{ns})$ in Fig. 2. In both cases, the error rate is evaluated with three different methods. The "gate-level" results show the error rate measured using a gate-level simulation of the decoder, with timing annotations corresponding to the chosen operating condition. The "sw MC" results are generated using the high-level software implementation of the decoding algorithm. These high-level software results actually use a flooding message-passing schedule while doubling the maximum number of iterations. Finally, the "DE-based" results are generated by evaluating (5) and (7) numerically. The gate-level simulations are of course the most computationally intensive, and for this reason can only be generated for relatively high error rates. On the other hand, once the deviation models have been generated, the curves based on DE can be generated in less than a minute.

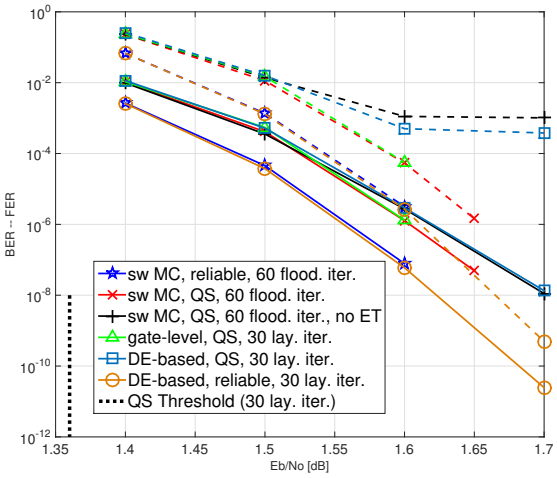

Fig. 1. Bit (solid curves) and frame (dashed curves) error rate of the $(3, 6)$ code of length 32,000 for a QS decoder at $\gamma = (0.85\,\mathrm{V}, 2.2\,\mathrm{ns})$, compared with a reliable decoder.

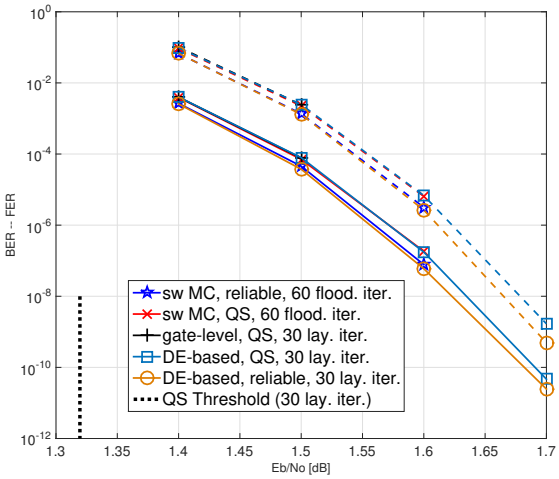

Fig. 2. Bit (solid curves) and frame (dashed curves) error rate of the $(3, 6)$ code of length 32,000 for a QS decoder at $\gamma = (0.85\,\mathrm{V}, 2.3\,\mathrm{ns})$, compared with a reliable decoder.

We can see in Fig. 1 that operating the decoder at $\gamma = (0.85\,\mathrm{V}, 2.2\,\mathrm{ns})$ results in an important loss of coding gain. At this operating condition, up to 3% of the messages are affected by deviations. Despite the large deviation rate, both the high-level simulation and the DE results can accurately predict the performance of the decoder when the SNR is 1.4 or 1.5 dB. Starting at 1.6 dB, the DE result predicts an error floor that is not observed in the decoder that uses early termination. On the other hand, when early termination is not used ("no ET"), the DE result accurately predicts the performance. The ability of a reliable early termination mechanism to eliminate an early floor

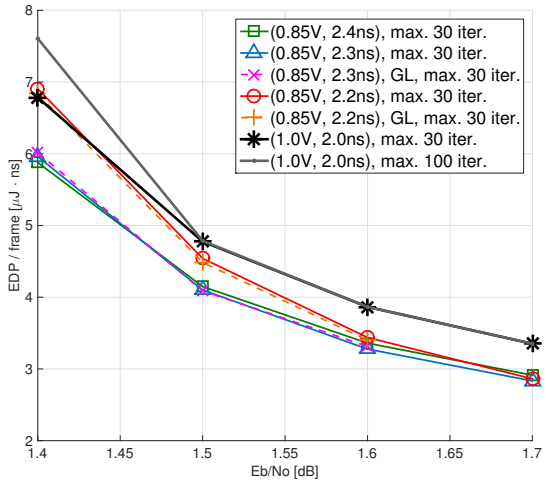

Fig. 3. Average EDP required to decode a frame of the $(3, 6)$ code of length 32,000 at various operating conditions, using layered message-passing. The dotted curves are based on the frame error rate measured on the gate-level model.

caused by deviations was previously noted in [14], and can be explained by the fact that when deviations occur independently from one iteration to the next, the probability that deviations prevent termination decreases exponentially in the number of additional iterations.

When the operating condition is instead selected as $\gamma = (0.85\,\mathrm{V}, 2.3\,\mathrm{ns})$, the deviation rate remains below 1%, and the error correction performance of the QS decoder becomes very similar to the performance of a reliable decoder. Both the DE-based results and the high-level software accurately predict the performance.

Finally, Fig. 3 shows the EDP metric at three different operating conditions. We can first see that since an early termination mechanism is used, the maximum number of iterations only has an impact on EDP at low SNR values. Indeed, the impact of decoding iterations that have a low probability of being required is negligible on the average energy and EDP. For this reason, the erroneous error floor obtained with the DE-based result at $\gamma = (0.85\,\mathrm{V}, 2.2\,\mathrm{ns})$ has no impact on the EDP evaluation. The first interesting thing to note from Fig. 3 is that a QS decoder can actually have a worse EDP than a reliable decoder, even as its coding gain is degraded. The reason is that the slower convergence of the QS decoder can overcome the reduction in the energy of each decoding iteration. Also note that the optimal operating condition can be different depending on SNR. The operating condition that yields the best EDP at 1.7dB is $\gamma = (0.85\,\mathrm{V}, 2.3\,\mathrm{ns})$, providing a gain of 15% with respect to the reliable decoder.

To assess the accuracy of the EDP results, we would ideally like to measure the energy used by the gate-level decoder at each iteration. However, this would require measuring switching activity separately for each iteration, which is cumbersome. Instead, we assume that the energy model $E^{(t)}_{\gamma,\infty}(p_{\mathrm{obs}})$ is accurate, since it is measured on a test circuit that is identical to the decoder circuit. We then evaluate the energy consumption using the frame error rate $R^{(t)}_{\gamma,N}(p_o)$ measured on the gate-level decoder. The resulting EDP is shown as dotted curves in Fig. 3. These curves confirm that the frame error rate at each decoding iteration is predicted accurately, which leads to accurate energy predictions, as long as the energy model itself is accurate.

V. Conclusion

In this paper, we performed gate-level simulations of an Offset Min-Sum LDPC decoder affected by timing violations, and showed that its decoding performance can be accurately predicted using the memoryless deviation model introduced in [6]. We introduced an analytical approach based on density evolution that allows predicting the BER and FER of the QS decoder after any number of iterations, which in turn can be used to obtain accurate estimates of its energy consumption or EDP metric.

The EDP gain results presented are only due to the fact that different circuit inputs activate different paths through a circuit and cause different delays. We expect that much larger gains can be obtained once process and operating condition variations are taken into account.

Acknowledgements

The authors wish to thank CMC Microsystems for providing access to the Cadence tools and TSMC 65nm CMOS technology. Warren Gross is a member of ReSMiQ and of SYTACom.

References

[1] A. Rahimi, L. Benini, and R. K. Gupta, "Variability mitigation in nanometer CMOS integrated systems: A survey of techniques from circuits to software," *Proceedings of the IEEE*, vol. 104, no. 7, pp. 1410–1448, Jul. 2016.

[2] L. Varshney, "Performance of LDPC codes under faulty iterative decoding," *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4427–4444, Jul. 2011.

[3] E. Dupraz, D. Declercq, B. Vasic, and V. Savin, "Analysis and design of finite alphabet iterative decoders robust to faulty hardware," *IEEE Trans. on Communications*, vol. 63, no. 8, pp. 2797–2809, Aug. 2015.

[4] A. Balatsoukas-Stimming and A. Burg, "Density evolution for min-sum decoding of LDPC codes under unreliable message storage," *IEEE Commun. Lett.*, vol. 18, no. 5, pp. 849–852, May 2014.

[5] S. Brkic, O. Al Rasheed, P. Ivanis, and B. Vasic, "On fault tolerance of the Gallager B decoder under data-dependent gate failures," *IEEE Commun. Lett.*, vol. 19, no. 8, pp. 1299–1302, Aug. 2015.

[6] F. Leduc-Primeau, F. R. Kschischang, and W. J. Gross, "Modeling and energy optimization of LDPC decoder circuits with timing violations," *CoRR*, vol. abs/1503.03880, 2015. [Online]. Available: http://arxiv.org/abs/1503.03880

[7] E. Sharon, S. Litsyn, and J. Goldberger, "Efficient serial message-passing schedules for LDPC decoding," *IEEE Trans. on Information Theory*, vol. 53, no. 11, pp. 4076–4091, Nov 2007.

[8] C. Roth, P. Meinerzhagen, C. Studer, and A. Burg, "A 15.8 pJ/bit/iter quasi-cyclic LDPC decoder for IEEE 802.11n in 90 nm CMOS," in *2010 IEEE Asian Solid State Circuits Conf.*, Nov. 2010, pp. 1–4.

[9] S.-Y. Chung, J. David Forney, T. J. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045dB of the Shannon limit," *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 58–60, Feb. 2001.

[10] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, 2008.

[11] R. Yazdani and M. Ardakani, "Waterfall performance analysis of finite-length LDPC codes on symmetric channels," *IEEE Trans. on Communications*, vol. 57, no. 11, pp. 3183–3187, Nov 2009.

[12] D. J. C. MacKay, "Encyclopedia of sparse graph codes." [Online]. Available: http://www.inference.phy.cam.ac.uk/mackay/codes/data.html

[13] Cadence Design Systems Inc., "Encounter digital implementation system." [Online]. Available: http://www.cadence.com/products/di/edi_system

[14] C. K. Ngassa, V. Savin, E. Dupraz, and D. Declercq, "Density evolution and functional threshold for the noisy min-sum decoder," *IEEE Trans. on Communications*, vol. 63, no. 5, pp. 1497–1509, May 2015.