



HAL
open science

A Min-Sum Iterative Decoder Based on Pulsewidth Message Encoding

Kevin Cushon, Camille Leroux, Saied Hemati, Shie Mannor, W.J. Gross

► **To cite this version:**

Kevin Cushon, Camille Leroux, Saied Hemati, Shie Mannor, W.J. Gross. A Min-Sum Iterative Decoder Based on Pulsewidth Message Encoding. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2010, 57 (11), pp.893 -897. 10.1109/TCSII.2010.2082970 . hal-00670670

HAL Id: hal-00670670

<https://hal.science/hal-00670670>

Submitted on 15 Feb 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Min-Sum Iterative Decoder Based on Pulsewidth Message Encoding

Kevin Cushon, *Student Member, IEEE*, Camille Leroux, *Member, IEEE*, Saied Hemati, *Senior Member, IEEE*, Shie Mannor, *Senior Member, IEEE*, and Warren J. Gross, *Senior Member, IEEE*

Abstract—In this brief, we introduce a new iterative decoder implementation called pulsewidth-modulated min-sum (PWM-MS), in which messages are exchanged in a pulsewidth-encoded format. The advantages of this method are low switching activity, very low complexity check nodes, low routing congestion, and excellent energy efficiency. We implement a fully parallel PWM offset MS decoder for a (660, 484) regular (4, 15) low-density parity-check code with 4-bit quantization in 0.13- μm CMOS, with a core area of 5.76 mm² (4.24-mm² cell area or 556K equivalent AND gates). In postlayout simulations, this decoder achieves an average information throughput of 5.71 Gb/s and an energy consumption of 65.8 pJ per information bit at a signal-to-noise ratio of 5.5 dB. Our results show a 21% reduction in area, a 0.6-dB improvement in coding gain, and an energy efficiency improvement of 19% over the comparable bit-serial MS decoder architecture. We also demonstrate 3-bit implementations, in which the coding gain is traded off for further improvements in throughput, area, and energy efficiency.

Index Terms—Low-density parity-check (LDPC) codes, low power, low switching activity, min-sum (MS) iterative decoding, pulsewidth modulation (PWM).

I. INTRODUCTION

ITERATIVE decoding techniques play a prominent role in modern forward-error-correction applications. Originally proposed for low-density parity-check (LDPC) codes [1], the interest in iterative decoding was renewed with the introduction of Turbo codes [2], where it was shown that these methods could achieve near-capacity error-correction performance for reasonable complexity. It was later discovered that LDPC codes offered similar near-capacity performance [3]. Other notable examples of iteratively decoded codes include serially concatenated codes and product codes [4]. Today, Turbo and LDPC codes remain the most eminent examples of iteratively decoded codes, as evidenced by their inclusion in numerous current and upcoming communications standards.

Since an iterative decoder can account for a large portion of the silicon area and consumed power in a receiver, they are a prime target for power reduction efforts. Several different approaches to low-power iterative decoding have been taken.

Manuscript received April 1, 2010; revised July 12, 2010; accepted September 13, 2010. Date of publication October 28, 2010; date of current version November 17, 2010. This paper was recommended by Associate Editor Z. Wang.

The authors are with the Department of Electrical and Computer Engineering, McGill University, Montreal, QC H3A 2A7, Canada (e-mail: kevin.cushon@mail.mcgill.ca; camille.leroux@mcgill.ca; saied.hemati@mcgill.ca; shie.mannor@mcgill.ca; warren.gross@mcgill.ca).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSII.2010.2082970

Among these are bit-serial (BS) architectures such as [5] and [6], in which probability updates are exchanged bit serially over single wires. This has the effect of reducing computational complexity and routing congestion.

Another way to reduce power consumption is to reduce switching activity. In [7], differential density evolution is used to analyze the switching activity of various message exchange formats in an LDPC decoder. Differential BS message passing is found to significantly reduce interleaver switching activity compared with conventional BS message passing.

Various analog iterative decoders have also been proposed and built [8], [9]. Such analog decoders have been tested to have much better energy efficiency than digital decoders. However, due to process variations and imprecision, it is not practical to scale analog decoders to large codeword lengths, whereas the current LDPC code standards specify codeword lengths of hundreds or thousands of bits. Analog decoders also have much lower throughput compared with digital decoders.

In this brief, we propose a new approach to low-power iterative decoding called *pulsewidth-modulated min-sum* (PWM-MS). In PWM-MS, messages are exchanged over single wires, with their magnitudes encoded using a digital pulse of variable width. This technique has a number of advantages in the power domain: Single-wire messages reduce overall wire lengths and parasitic capacitance, it has a lower average switching activity in the interleaver compared with BS-MS, and it has very simple computational units.

The remainder of this brief is structured as follows. Section II provides a brief background of the algorithmic notions necessary to understand this brief and the complexity problems faced in MS decoding. Section III introduces pulsewidth message encoding and shows the advantages it has over other message exchange techniques. Section IV details how pulsewidth message encoding gives rise to low-complexity MS decoder architectures. Section V contains the implementation results for PWM-MS LDPC decoders. Finally, concluding remarks and a summary are presented in Section VI.

II. MS ITERATIVE DECODING

Many algorithms for iterative decoding have been formulated, but of those, many can be described as instances of a general algorithm known as belief propagation (BP) over a graphical model of the code [10]. The basic principle of this algorithm is the exchange of messages, or “beliefs,” between processing nodes associated with each vertex of this graph. More specifically, it has been noted that these algorithms fall

under two general subcategories of BP: the sum-product algorithm and the MS algorithm (MSA) [11].

Let us consider the conventional form of the MSA, which is applied to a Tanner graph representation of a code. The decoder is initialized based on the information received at the channel output. Decoding is performed by iteratively passing messages through the edges of the Tanner graph between the variable nodes and the check nodes. In the k th iteration ($k > 0$), in the log-likelihood ratio (LLR) domain, the messages that are passed between a variable node v and a check node c are defined as follows:

$$m_{v \rightarrow c}^{(k)} = L_v + \sum_{c' \in C_i \setminus c} m_{c' \rightarrow v}^{(k-1)} \quad (1)$$

$$m_{c \rightarrow v}^{(k)} = \prod_{v' \in V_j \setminus v} \text{sgn}(m_{v' \rightarrow c}^{(k)}) \cdot \min_{v' \in V_j \setminus v} (|m_{v' \rightarrow c}^{(k)}|) \quad (2)$$

where C_i and V_j are the sets of the check and variable nodes incident to the variable node v and the check node c , respectively. The message $m_{a \rightarrow b}^{(k)}$ is the message sent from node a to node b at iteration k . The initial round of messages $m_{a \rightarrow b}^{(0)}$ are set to zero. L_v is the initial weight of the variable node v in the LLR domain, which is based on the received value from the channel.

The estimation of the transmitted bit in each variable node is computed by taking the sum of all incoming messages from the check nodes, plus the initial channel information L_v . The decoding process continues until a stopping criterion is reached; this could be once the estimated bits from all the variable nodes form a valid codeword or when an iteration limit k_{\max} is reached.

While the variable node processing consists of simple addition, the check-node processors represent a large part of the complexity in an MS iterative decoder. The reason for this is that each check node must compute the minima of all incoming extrinsic messages for each of its outputs. It is also possible to apply heuristics to the check-node computation, reducing complexity at the cost of some error-correction performance. Split-row MS [12] and the single minimum with correction technique for BS-MS in [5] are examples of this. Pulswidth-encoded messages, on the other hand, are naturally suited to MS decoding, as finding the minima of these messages is very simple, eliminating this source of complexity.

III. PULSWIDTH MESSAGE ENCODING

In digital CMOS circuits, the causes of power dissipation can be classified as either static or dynamic. Dynamic power refers to power dissipated when circuit nodes switch from one voltage level to another. Static power refers to constant leakage currents that are present even when the circuit is inactive.

Although static power consumption has been observed to increase as a share of total power at smaller manufacturing process sizes, dynamic power is generally the dominant factor in iterative decoder integrated circuits. For instance, in [12], in a fully parallel iterative decoder in 65-nm CMOS, static power accounts for only 0.5% to 1% of the total power. Furthermore, recent developments in manufacturing processes have shown that gate leakage currents can be greatly reduced.

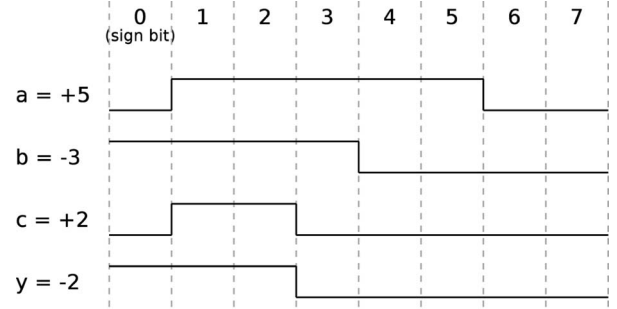


Fig. 1. Some examples of pulswidth-encoded messages. The message y is the check-node result produced from the inputs a , b , and c .

Auth *et al.* [13] demonstrate 45-nm transistors with gate leakage currents reduced by two to three orders of magnitude compared with 65-nm transistors. Subthreshold currents, which are another source of static power consumption, can be reduced by using transistors with high threshold voltages. For these reasons, in this work, we exclusively focus on reducing dynamic power consumption.

In an iterative decoder, the switching activity generated by message passing has a particularly high priority for reduction. One reason for this is that message passing represents a great volume of exchanged information, so representing these messages in a power-efficient format can have a large impact on the circuit's overall power consumption [7]. In addition, the routing networks required to support message passing can necessitate very long wires with multistage buffer chains, particularly in large codes, leading to a high effective parasitic capacitance for these signals.

In our proposed pulswidth message encoding, the messages between the variable and check nodes are exchanged in sign-magnitude format, with the magnitude determined by the width of a digital pulse, and transferred over single wires. Fig. 1 shows some examples of one possible pulswidth encoding scheme, in which the first bit of each message is a sign bit. The subsequent bits indicate the magnitude, with the signal held at logic 1 for a number of clock cycles equal to the magnitude. The length of a decoding iteration in clock cycles is thus equal to the maximum message magnitude, plus 1 for the sign bit. Since this number can be set arbitrarily, it is possible to have a nonpower-of-2 number of quantization levels.

One of the key motivations for using such a message encoding scheme is the very low switching activity. Using the encoding scheme shown in Fig. 1, for instance, each message has a maximum of one 0-to-1 transition per decoding cycle. This low activity translates directly to low dynamic power consumption. The switching activity could be reduced even further using a transition-based encoding scheme, in which the signal holds the same state as the sign bit and then transitions to indicate the magnitude. However, this would result in a more complex CMOS circuit implementation as it requires transition-sensitive circuitry. For the remainder of this brief, we consider only the level-based scheme, as indicated in Fig. 1.

Since the activity factor varies as the decoding process proceeds, it is more relevant to characterize the switching activity using the total transition count observed during decoding [7]. Fig. 2 plots the simulated average number of transitions per

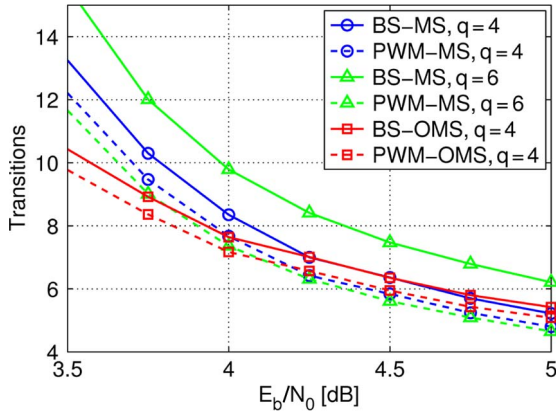


Fig. 2. Average number of transitions per edge per decoded codeword for the (660, 484) LDPC code used in this work.

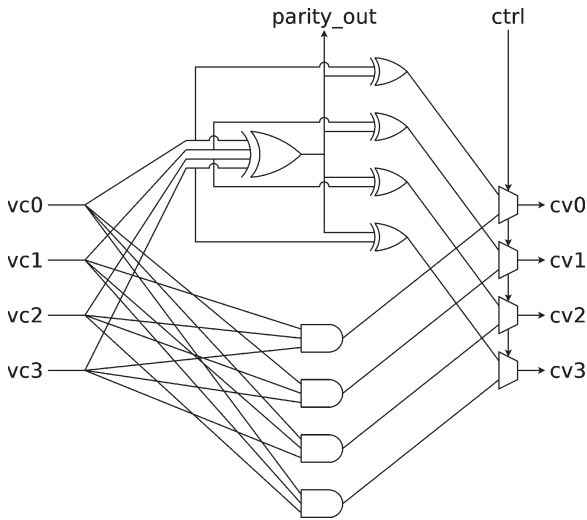


Fig. 3. PWM-MS check node of degree $d_c = 4$.

edge in the Tanner graph per decoded codeword for PWM-MS, PWM offset MS (PWM-OMS), and BS-MS to serve as comparison. This result shows that PWM has a lower switching activity than BS-MS for all given values of the number of quantization bits q and SNR. The difference is relatively minor at $q = 4$ (about 5%); however, at $q = 6$, this advantage increases to over 25%. This is because the switching activity for PWM-MS remains roughly constant with q . In BS-MS, the average transition count increases with the number of bits. The results for OMS (PWM-OMS and BS-OMS) are also plotted, with PWM-OMS showing a similar advantage.

IV. LOW-COMPLEXITY ARCHITECTURE OF PWM-MS DECODERS

In addition to low switching activity, another major advantage of PWM-MS is that it results in a very low complexity check node. As shown in the check-node schematic in Fig. 3, the minimum of a group of messages can be computed with a single AND gate. The check node is fully combinatorial; thus, the CMOS implementation is very compact, and power is further saved by the lack of sequential elements. The XOR gate network is used to determine the signs of the outgoing messages on each edge, whereas the AND gates determine

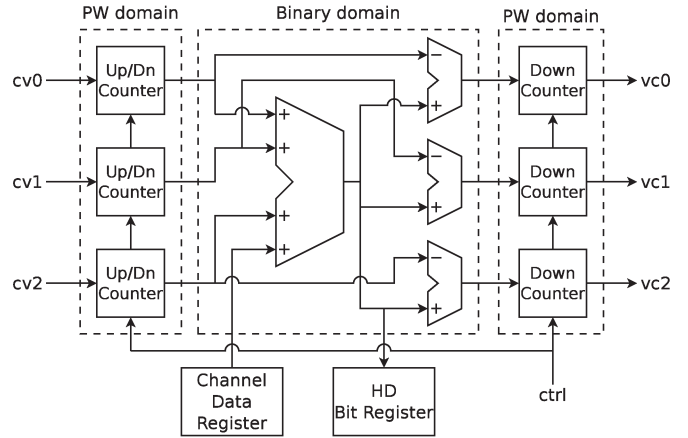


Fig. 4. PWM-MS variable node of degree $d_v = 3$.

the magnitudes in accordance with (2). A control signal $ctrl$ selects the appropriate output during the sign and magnitude computation phases. A parity output is used for convergence detection and early termination logic.

Not only is this check-node architecture very compact, but it also offers an exact implementation of the MS check-node function. There is no performance loss resulting from the application of heuristics.

The PWM-MS variable node is shown in Fig. 4. It consists of an up/down counter on each incoming edge, which converts the incoming PWM message to 2's complement binary. An adder network computes the hard-decision bit estimation, as well as the outgoing messages on each edge. Down counters on each outgoing edge take the binary messages as input and encode them in a PWM format. In the variable node, the $ctrl$ input signals the start of a new iteration, which resets the up/down counters to zero and loads the down counters.

One notable property of this variable node design is that it can be used to implement OMS for a negligible increase in complexity, as opposed to other decoder architectures including BS. This is because the sign-magnitude format of PWM allows the offset to be efficiently applied in the variable node. The offset is implemented by setting the output counters to transition when their internal count reaches a number equal to or less than a predetermined offset value, rather than zero. Furthermore, applying an offset reduces the maximum message magnitude. With pulswidth message encoding, this reduces the number of clock cycles per iteration and increases throughput accordingly. For instance, with 4-bit quantization and an offset of 1, the maximum message magnitude decreases from 7 to 6, and the number of clock cycles per iteration decreases from 8 to 7, increasing throughput by 12.5%. It is therefore very advantageous to use PWM-OMS over PWM-MS.

V. IMPLEMENTATION RESULTS

We implemented PWM-OMS decoders in three different configurations in IBM CMRF8SF standard- V_t 0.13- μm CMOS to obtain estimates for silicon area, throughput, and power consumption. We also implemented PWM-OMS and PWM-MS decoder prototypes on a Xilinx Virtex-5 LX330 field-programmable gate array (FPGA) for verification and

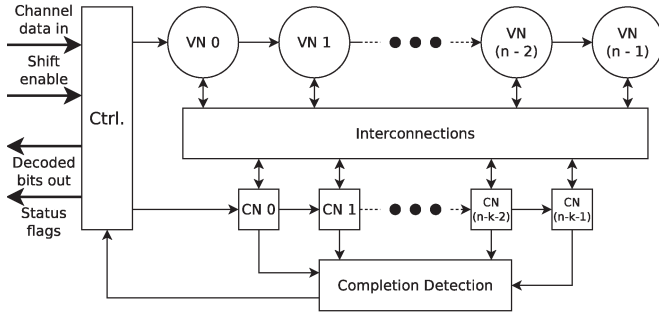


Fig. 5. Block diagram of the implemented decoder.

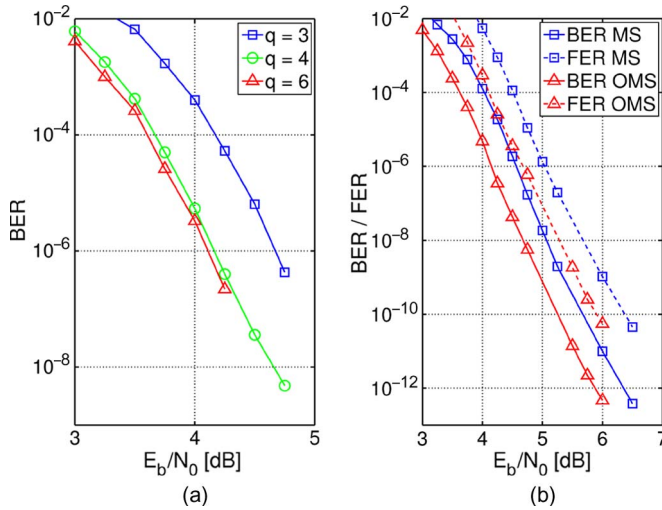


Fig. 6. Decoding performance (a) for different quantization levels of OMS with an offset of 1 and (b) for 4-bit PWM-MS and PWM-OMS decoders implemented on the FPGA.

bit-error-rate (BER)/frame-error-rate (FER) measurements. In all cases, the LDPC code used is a (660, 484) regular (4, 15) progressive-edge-growth-based LDPC code. This code has also been used in [5] for verifying the BS approach. This section contains discussion of these results, as well as comparisons with other decoders.

Fig. 5 shows a top-level block diagram of the implemented decoder. It is a fully node-parallel architecture, and the interconnections between the nodes are each a single wire. Early termination logic uses parity checks from the check nodes to detect convergence. A state machine *ctrl* generates control signals and acts as the off-chip interface.

The impact of the number of quantization bits on the BER performance of PWM-OMS is shown in Fig. 6(a). In terms of the number of quantization bits, $q = 4$ results in a very small performance loss compared with $q = 6$, whereas $q = 3$ incurs another loss of 0.4–0.5 dB. Fig. 6(b) shows the BER and FER performance of the FPGA decoder prototypes. For this code, OMS gives a performance gain of 0.4–0.5 dB over conventional MS. All BER tests use a maximum of 15 decoding iterations. These FPGA prototype results exactly match the results obtained with a software model. These performance tests used an emulated Gaussian channel [14].

Table I shows the application-specific integrated circuit implementation results for our decoder in 0.13- μm CMOS, alongside the BS approximate MS decoder in [5]. Comparisons

TABLE I
IMPLEMENTATION RESULTS AND COMPARISON WITH [5]

	This work			[5]
Architecture	PWM-OMS			Bit-serial approx. MS
LDPC code	(660, 484)			(660, 484)
Process and supply voltage	0.13 μm / 1.2V			0.13 μm / 1.2V
Configuration	A	B	C	n/a
Quantization width [bits]	4	3	3	4
Pipelining	no	no	yes	yes
Clock cycles / iteration	7	3	4	4
Clock frequency [MHz]	150	150	250	300
Core area [mm^2]	5.76	4.50	4.54	7.3
Cell area [mm^2]	4.24	3.31	3.30	5.26
Gate equivalent [Kgates]	556	434	433	690
Max. iterations	15			15
Coding gain [dB] ^a	0.5	-0.1	-0.1	-0.1
Avg. information throughput (5.5 dB) [Gbps]	5.71	9.18	12.2	17.1
Avg. power (5.5 dB) [mW]	376	465	819	1383
Avg. energy per decoded bit (5.5 dB) [pJ/bit]	65.9	50.7	67.1	80.9
T/P per unit area [Gbps / mm^2]	0.99	2.04	2.69	2.34
T/P per unit area per unit power [Gbps / ($\text{mm}^2 \cdot \text{W}$)]	2.64	4.39	3.28	1.69

^aRelative to conventional min-sum with $q = 4$ and 15 max. iterations.

between our decoder and this one are particularly relevant as they both use the same process size, supply voltage, and LDPC code. They also have architectural similarities—both are fully node parallel with single-wire node interconnections, and both were primarily designed for low-power applications. Throughput and power for our designs were calculated using postlayout simulations, with back-annotated delays and wiring parasitics, and includes the clock trees.

We implemented three configurations of PWM-OMS decoders. Configuration A is optimized for decoding performance. Configuration B is optimized for energy efficiency, whereas configuration C is optimized for throughput.

Since our decoder architecture uses OMS and no heuristics in the check node, configuration A achieves a coding gain of approximately 0.5 dB over conventional 4-bit MS with a maximum of 15 iterations, as shown in Fig. 6(b). On the other hand, [5] incurs a loss of 0.1 dB due to a check-node approximation. We note that, with $q = 3$, PWM-OMS can achieve approximately the same BER performance as the 4-bit approximate MS used in [5]. Hence, in configurations B and C, we trade off this 0.6-dB coding gain for reduced area, higher throughput, and better energy efficiency.

Configuration A has seven clock cycles per decoding iteration. In standard MS, eight cycles would be required—one for the message sign, plus seven for the 3 bits of magnitude. However, since we implement OMS, we apply an offset of 1, reducing the maximum magnitude to 6. Likewise, configuration B takes only three cycles per decoding iteration (one for the sign and two for the magnitude). Pipelining adds a cycle of latency; thus, configuration C requires four cycles.

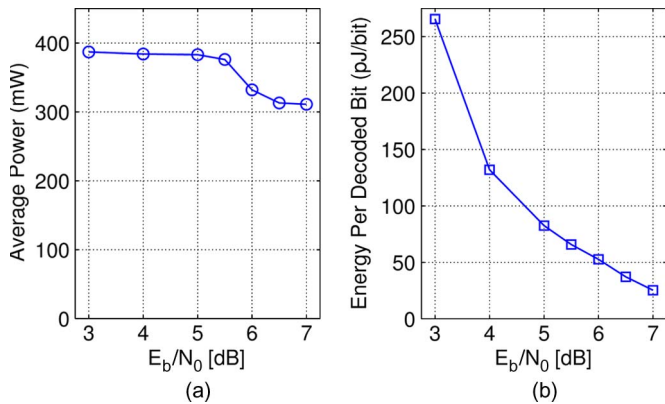


Fig. 7. (a) Average power consumption and (b) energy efficiency for a PWM-OMS decoder with $q = 4$.

Area is reduced relative to BS-MS due to the simpler check node. We define average throughput as the throughput achieved by immediately beginning decoding of a new codeword once the current one has converged. Average throughput is therefore determined by the average number of iterations required for convergence and varies for different values of SNR. Raw throughput is lower, due to both the lower clock frequencies and (in the case of configuration A) higher number of clock cycles per iteration. In terms of throughput per unit area, however, configuration C is 15% higher than [5]. We also define another metric that takes power consumption into account: throughput per unit area per unit power. In these terms, all three PWM-OMS implementations are superior to [5]. Energy efficiency, which is defined in terms of energy consumed per decoded bit, is also better in PWM-OMS. Plots of average power consumption and energy efficiency for the A decoder over a range of SNR values are shown in Fig. 7. Energy efficiency was determined by dividing the average power by the average throughput at each given value of SNR.

The complexity and energy efficiency of PWM-OMS also favorably compare with other recent LDPC decoders, although differences in the process size and LDPC code make direct comparisons difficult. One example is the partially parallel OMS decoder in [15], which implements a (2048, 1723) Reed–Solomon-based code in 65-nm CMOS. It achieves an average information throughput of 40 Gb/s at 5.5-dB SNR, with power of 2800 mW and energy of 69.8 pJ/bit.

Another recent architecture is the 90-nm fully node- and bit-parallel MS decoder in [16], which achieves an information throughput of 13.21 Gb/s, with average power of 1323 mW and energy of 98 pJ/bit at 5-dB SNR. It should be noted that these figures are without early termination, which was used to greatly increase throughput and decrease energy in [5], [15], and this work. However, since [16] partitions long internode wires with registers, early termination would be less effective due to added complexity and increased latency. In addition, this architecture requires parallel check nodes, which become very complex at high degrees, whereas in this work, the check-node complexity is extremely low, even at high degrees.

If we assume energy consumption per information bit scales quadratically with feature size, the scaled energy values of [15] and [16] are 279 and 204 pJ/bit, respectively, giving this work respective improvements of 76% and 68%.

VI. CONCLUSION

In this brief, we have presented a new iterative decoding technique called PWM-MS and implemented it for a (660, 484) regular (4, 15) LDPC code. The advantages of this architecture include very low complexity check nodes and low message exchange switching activity. Our implementation of an OMS decoder with 4 bits of message quantization achieved a core area of 5.76 mm² and an average information throughput of 5.71 Gb/s at 5.5-dB SNR. It also achieved an energy efficiency value of 65.9 pJ/bit at this SNR and a coding gain of 0.5 dB over conventional MS, which can be traded off for additional improvements in area, throughput, and energy by reducing the number of quantization levels. These results for area and energy represent respective improvements of 21% and 19% compared with the similar BS-MS decoder architecture. These results also favorably compare with other recent LDPC decoder architectures.

REFERENCES

- [1] R. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.
- [2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. ICC*, May 1993, vol. 2, pp. 1064–1070.
- [3] D. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.
- [4] R. Pyndiah, "Near-optimum decoding of product codes: Block turbo codes," *IEEE Trans. Commun.*, vol. 46, no. 8, pp. 1003–1010, Aug. 1998.
- [5] A. Darabiha, A. Chan Carusone, and F. Kschischang, "Power reduction techniques for LDPC decoders," *IEEE J. Solid-State Circuits*, vol. 43, no. 8, pp. 1835–1845, Aug. 2008.
- [6] T. Brannon, R. Hang, G. Block, V. C. Gaudet, B. Cockburn, S. Howard, C. Giasson, K. Boyle, P. Goud, S. S. Zeinoddin, A. Rapley, S. Bates, D. Elliott, and C. Schlegel, "A scalable LDPC decoder ASIC architecture with bit-serial message exchange," *Integration, VLSI J.*, vol. 41, no. 3, pp. 385–398, May 2008.
- [7] V. Gaudet, C. Schlegel, and R. Dodd, "LDPC decoder message formatting based on activity factor minimization using differential density evolution," in *Proc. ITW*, Sep. 2007, pp. 571–576.
- [8] S. Hemati, A. Banihashemi, and C. Plett, "A 0.18- μ m CMOS analog min-sum iterative decoder for a (32, 8) low-density parity-check (LDPC) code," *IEEE J. Solid-State Circuits*, vol. 41, no. 11, pp. 2531–2540, Nov. 2006.
- [9] C. Winstead, N. Nguyen, V. Gaudet, and C. Schlegel, "Low-voltage CMOS circuits for analog iterative decoders," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 4, pp. 829–841, Apr. 2006.
- [10] F. Kschischang and B. Frey, "Iterative decoding of compound codes by probability propagation in graphical models," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 219–230, Feb. 1998.
- [11] N. Wiberg, *Codes and Decoding on General Graphs*. Linköping, Sweden: Linköping Univ., 1996.
- [12] T. Mohsenin, D. Truong, and B. Baas, "A low-complexity message-passing algorithm for reduced routing congestion in LDPC decoders," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 5, pp. 1048–1061, May 2010.
- [13] C. Auth, M. Buehler, A. Cappellani, C.-H. Choi, G. Ding, W. Han, S. Joshi, B. McIntyre, M. Prince, P. Ranade, J. Sandford, and C. Thomas, "45 nm high- k + metal gate strain-enhanced transistors," *Intel Technol. J.*, vol. 12, no. 2, pp. 77–85, Jun. 2008.
- [14] A. Ghazel, E. Boutillon, J.-L. Danger, G. Gulak, and H. Laamari, "Design and performance analysis of a high speed AWGN communication channel emulator," in *Proc. PACRIM*, 2001, vol. 2, pp. 374–377.
- [15] Z. Zhang, V. Anantharam, M. J. Wainwright, and B. Nikolic, "An efficient 10GBASE-T ethernet LDPC decoder design with low error floors," *IEEE J. Solid-State Circuits*, vol. 45, no. 4, pp. 843–855, Apr. 2010.
- [16] N. Onizawa, T. Hanyu, and V. Gaudet, "Design of high-throughput fully parallel LDPC decoders based on wire partitioning," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 3, pp. 482–489, Mar. 2010.