Ultra-Low-Power Error Correction Circuits: Technology Scaling and Sub-V_{T} Operation

Chris Winstead, *Senior Member, IEEE*, and Joachim Neves Rodrigues, *Senior Member, IEEE*

***Abstract*—Techniques are evaluated for implementing error correction codes in wireless applications with severe power constraints, such as bio-implantable devices and energy harvesting motes. Standard CMOS architectures are surveyed and compared against alternative implementations, including known sub-V_{T} analog decoding techniques. Novel sub-V_{T} digital designs are proposed, and their power efficiency is evaluated as a function of operating voltage and clock frequency. Sub-V_{T} implementation is predicted to offer 29× gain in power consumption for a (3,6) low-density parity-check decoder of length $N = 512$ operating at a throughput of 200 Mb/s, compared to standard digital implementation of the same design.**

***Index Terms*—Analog decoders, biomedical implants, error correction codes (ECC), sub-threshold, ultra-low voltage.**

I. Introduction

ERROR correction codes (ECC) are widely used to optimize performance in wireless communication devices. Some advanced ECC techniques—generically described as *iterative message passing* algorithms—are able to approach theoretical limits on performance, which is defined by the minimum signal energy needed to maintain a reliable operation. Unfortunately, the message-passing ECC implementations typically consume more than 100 mW of power [1]–[10]. Advanced ECC options are therefore inaccessible to micro-power communication devices, which are increasingly important for biomedical applications, personal area networks, distributed sensor networks, and machine-to-machine interfaces.

ECC solutions to improve communication performance in biomedical devices were previously examined [11], [12]. Body-area-networking devices that utilize ECC circuits have also been described [13]. Previous articles consider using sub-threshold (sub-V_{T}) analog decoding circuits as a means of minimizing the power cost of ECC modules [14]–[19]. Analog decoding circuits have been studied for more than 10 years and several analog circuit techniques have been devised to maximize power efficiency. Although many different low-power ECC implementations have been described in the literature, to date there has not been any comprehensive study to evaluate the comparative efficiencies of alternative analog and digital implementations.

Contribution: In this paper, we evaluate and compare the efficiencies of analog and digital message-passing ECC decoders that are reported in the literature. Furthermore, we extend this analysis to include the estimated performance of a new sub-V_{T} digital implementation, which is predicted to achieve better power efficiency than previous analog or digital designs. In order to provide a fair comparison, a generalized scaling theory is applied to correct for differences in process technology, operating speed, and supply voltage. The results of our analysis reveal that high-performance ECC designs—whether analog or digital—are unable to meet acceptable power limits for bio-compatible electronics. Finally, we demonstrate medium-performance sub-V_{T} digital decoders that meet bio-compatible power constraints. The medium-performance digital decoders are also shown to consume less energy per bit than previously reported analog decoders with comparable performance.

The remainder of this paper is organized as follows. Section II presents an overview of the most prevalent high-performance ECC implementations and considerations for low-power implementation. Section III proposes a technology normalization method for comparing ECC designs implemented in different technologies. Section IV describes the design and characterization of a sub-V_{T} digital ECC design and Section V offers conclusions.

II. ECC Implementations

A. Typical ECC Characteristics

During the past two decades, significant effort was invested to implement high-performance iterative ECC decoders based on Turbo [20] and low-density parity-check (LDPC) codes [21], [22], which are known to approach the theoretical Shannon Limit on communication performance. LDPC and Turbo codes may be classified as concatenated codes, which are composed of simpler sub-codes. Decoding is accomplished iteratively by exchanging *messages* between sub-decoders. The messages are interpreted as local decoding estimates for each of the sub-codes, and by combining all local informationß, a message-passing decoder obtains dramatically improved performance. The final decoded result is obtained after 5–25 iterations of message passing, depending on the particular code.

Table I lists a selection of common decoder implementations for iterative message-passing codes. Decoders are first classified by their standard channel type, which include hard- and soft-information channels. In the former, only binary data is received, whereas in the latter, the decoder processes real-valued samples from the analog channel. Soft-information decoders are more complex and achieve higher performance. Performance also tends to improve as the code's block size, N, is increased, with a corresponding increase in complexity. Current standards utilize codes with N equal to several hundred or

Manuscript received July 19, 2012; revised September 18, 2012; accepted October 27, 2012. Date of publication January 15, 2013; date of current version February 1, 2013. This work was supported by the US National Science Foundation under award ECCS-0954747 and by the Swedish Vetenskapsrådet under grant 621-2011-4540. This brief was recommended by Associate Editor M. Alioto.

C. Winstead is with the Department of Electrical and Computer Engineering, Utah State University, Logan, UT 84322-4120 USA (e-mail: chris.winstead@usu.edu).

J. N. Rodrigues is with the Department of Electrical and Information Technology, Lund University, 221 00 Lund, Sweden (e-mail: Joachim.Rodrigues@eit.lth.se).

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

Digital Object Identifier 10.1109/TCSII.2012.2231040

TABLE I
CLASSIFICATION OF BLOCK AND LDPC DECODER IMPLEMENTATIONS

Channel	Algorithm	Complexity	Architecture	Performance
Hard	Syndrome[1]	minimal	parallel	low
	Gallager[1]	low	parallel	moderate
Soft	BP	very high	parallel	very high
	Layered BP	high	semi-parallel	very high
	Layered MS	moderate	semi-parallel	high
	Analog BP[2]	moderate	parallel	high
	Analog MS[2]	moderate	parallel	high
	Turbo BP	very high	sliding-window	very high
	Turbo SOVA	moderate	sliding-window	high

[1] Sub-$V_{\rm T}$ digital designs considered in Sec. IV.
[2] Sub-$V_{\rm T}$ analog designs reported previously, and compared in Sec. III.

thousand bits. Table I also lists non-iterative syndrome type decoders, which are among the least complex (and lowest performance) HD decoders. Syndrome decoders utilize lookup tables (LUTs) that are efficient for very small N, but cannot be scaled up for high-performance decoding.

The more sophisticated message-passing decoders listed in Table I include Gallager, Belief Propagation (BP), and Min-Sum (MS) algorithms. Gallager-style decoders operate by toggling bits in a large Boolean network [21]. BP is a more complex algorithm that achieves near-optimal performance, but at a very high implementation cost [22], [23]. The MS algorithm is an approximate version of the BP algorithm that obtains significantly reduced complexity with a mild performance loss [7]. Most recent BP and MS implementations use a semi-parallel digital architecture to improve efficiency. In a typical semi-parallel architecture, messages are partitioned into memory blocks and parallel operations are performed on non-overlapping pairs of memory blocks. Turbo decoders are another code class that utilize the BP algorithm. Semi-parallel implementations are obtained for Turbo codes by using "sliding-window" algorithms. For the Turbo case, the soft-output Viterbi algorithm is often used, which can be considered as an approximation of BP in the Turbo case.

The earliest iterative ECC implementations required up to 1 W of power [7], which prompted some researchers to propose analog implementations [24]. Analog implementations have been demonstrated for both BP and MS algorithms, which operate by exchanging analog currents or voltages between nonlinear processing cells. Analog BP decoders are usually implemented using sub-$V_{\rm T}$ CMOS circuits and analog MS circuits have been demonstrated using super-V_T current-steering circuits [17]. Analog architectures are necessarily parallel and obtain moderate throughput (10 Mb/s to 1 Gb/s) through parallelism gain on large codes.

Although sub-$V_{\rm T}$ analog decoders show promise for low-power implementation, they tend to suffer from speed and performance degradation that are usually attributed to interface limitations. Analog decoders require an array of Sample-and-Hold (S/H) cells that perform serial-to-parallel conversion on the decoder's analog input samples. S/H circuits suffer from charge leakage, resulting in a gradual accumulation of error on samples, $Err \propto T_H$, where T_H is the maximum hold time. For a serial-to-parallel converter, $T_H = N/f$, where N is the code's block size and f is the throughput in bits per second. If a maximum S/H error tolerance is specified, then an analog decoder cannot be operated for N greater than $N_{\max} \propto Err_{\max} \times f_{\max}$, where $f_{\max}$ is the maximum S/H sample rate. Hence, the analog decoders' block size is limited and the achievable performance is correspondingly limited.

The available codes and decoding algorithms are subject to a wide range of parameters and channels, making it difficult to state a single definition of "performance." For the purposes of this paper, we define performance generally as the minimum input signal-to-noise ratio (SNR) required to achieve a specified bit error rate on a reference binary-input additive white Gaussian noise channel. In this definition, the rate-normalized SNR is used, equal to E_B/N_0, where E_B is the energy per data bit and N_0 is the channel's noise power spectral density. We also define an algorithm's "complexity" as that of the arithmetic operations that comprise the algorithm.

B. Low-Power Considerations

Low-power applications—particularly those emerging in the bio-medical and body-area-networking domains—demand extremely low power consumption for safe long-term use. For bio-implanted devices, power density must be no greater than 0.8 mW/mm^2 to avoid tissue damage through heating effects [25], which predicts a chip-scale limit on the order of 10 mW. This limit is exceeded by all reported digital BP, MS, and Turbo architectures, which lie in the range of 50 mW to 1 W [1]–[10]. The imagined bio-medical applications have low to moderate speed requirements, below 1 Gb/s. It was previously argued that sub-$V_{\rm T}$ analog decoders are a suitable match for these requirements [11], but our comparison analysis in Section III shows that sub-$V_{\rm T}$ analog decoders still consume more than 1 mW at moderate speed.

We now propose that Gallager-style algorithms provide a competitive option, and our analysis in Section IV demonstrates that sub-$V_{\rm T}$ Gallager designs achieve similar performance to analog decoders with lower power consumption. Hard-decision (HD) Gallager decoders offer several benefits over analog implementations. First, the system's overall complexity is reduced by using a hard-information demodulator, instead of a specialized analog interface. Second, Gallager decoders are implemented using mature digital-synthesis strategies and are straightforwardly mapped into any standard CMOS process.

C. HD Decoder Architecture

This paper focuses primarily on efficient implementation of HD Gallager-style decoders. Throughout this work, we also include comparison results for a simple $N = 7$ Hamming code syndrome decoder, which serves as a corner point for lowest complexity, lowest power, and poorest performance among useful ECC implementations. The syndrome decoder is highly efficient for small N, consisting of a small binary matrix multiplication and a LUT. The N-bit input vector r is first multiplied into the code's $N \times M$*parity-check matrix* H, yielding the M-bit *syndrome vectors* $= rH^T$. If the syndrome vector is non-zero, the input sequence contains at least one error. The LUT implements a one-to-one mapping between syndrome vectors and error patterns, hence $e = LUT(s)$. Finally, the errors are corrected when the error pattern is EXORed with the original input, yielding the decision vector $d = r \oplus e$.

The parity-check part of the syndrome decoder is implemented using a set of EXOR parity-check operations. The number of parity-checks M is linear with N; however, the number of LUT patterns grows with 2^M. As a result, syndrome decoders are only efficient for very small N. Iterative message-passing algorithms provide an alternative method of estimating the error pattern without requiring a LUT. These methods were

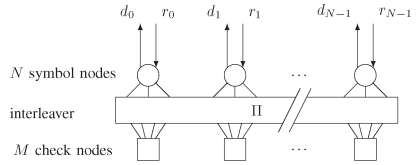

Fig. 1. Parallel architecture for an LDPC decoder, consisting of symbol nodes, check nodes, and an interleaver.

first reported by Gallager for decoding LDPC codes with sparse parity-check matrices and large N.

Among the Gallager-style decoders, the Gallager-A (GA) algorithm has the simplest logic and is therefore expected to provide the lowest power implementation. The GA decoder consists of N symbol nodes and M check nodes, as shown in Fig. 1. The decoder receives N input bits from the noisy channel, $r_0, \ldots, r_{N-1}$ and produces N output bits $d_0, \ldots, d_{N-1}$ via processing at each of N symbol nodes. The symbol nodes exchange binary messages with M check nodes. The interconnect between symbol and check nodes is governed by the interleaver Π, which implements a pseudo-random permutation on the message wires.

The GA decoding algorithm is described elsewhere [21] and briefly summarized as follows. Each symbol node S_j sends messages equal to its top-level input r_j to all adjacent check nodes. Then, each check node C_i computes the total parity P_i of its local messages, defined as the EXOR-sum over all messages. If $P_i = 0$, then the check node simply echoes the messages back to its adjacent symbol nodes. If $P_i = 1$, then at least one of the adjacent bits is erroneous, so the check node responds by inverting each of the messages. At the symbol nodes, unanimous-vote logic is applied, so that message values are updated if all incoming messages are equal, i.e., the channel bit is flipped if all adjacent parity-checks indicate an error. This process is repeated through several iterations so that erroneous bits are incrementally toggled and corrected.

III. Comparison of ECC Solutions

A. Technology Scaling Trends in ECC Implementations

Sub-V_T analog decoders initially proved to be much more power efficient than digital options, but that gain has diminished recently due to steady improvements by technology migration. Decoder power efficiency is conventionally measured in Joules per decoded bit, defined as $Jpb \triangleq PT/f_cN$, where P is the decoder's average power consumption, N is the number of decoded bits per frame, T is the number of iterations per frame, and f_c is the clock frequency. For standard CMOS technologies, the Jpb figure is proportional to the devices' power-delay-product, which improves with each successive process generation.

Power-efficiency trends for digital designs are predicted by generalized process scaling rules, which account for typical differences in supply voltage and terminal capacitance between different technology nodes [26]. For digital designs, the Jpb efficiency is predicted to scale by a factor of the power ratio γ_D, defined as

$$\gamma_D = \frac{C_2 V_{\mathrm{DD2}}^2}{C_1 V_{\mathrm{DD1}}^2} = \xi^2 \tag{1}$$

where ξ is the technology scale ratio, C the device terminal capacitance, and the subscripts "1" and "2" indicate the original reported technology and target technology, respectively. The terminal capacitances depend on device geometry and may vary significantly within a design, but they are expected to scale proportionally with ξ, and the supply voltage V_{DD} scales with $\xi^{1/2}$.

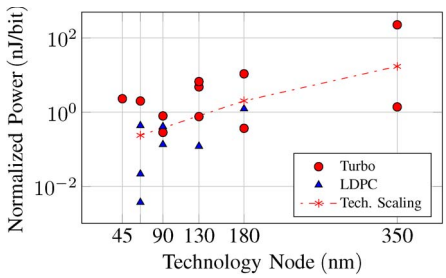

Fig. 2. Technology scaling trends in digital ECC implementations, including Turbo decoders [8]–[10] and LDPC decoders [1]–[7]. Also shown is the trend line predicted by Dennard scaling theory [26].

Fig. 2 shows the power efficiency of reported digital ECC chips, along with a dashed line indicating the improvements that are expected due to scaling theory. Decoder efficiency tends to follow the trend predicted by scaling theory, indicating that improvements are largely attributable to technology scaling, rather than design innovations. The earliest designs reported in Fig. 2, implemented in 350-nm CMOS technology, exceeded 1 nJ/bit. By comparison, sub-V_{T} analog designs were measured as low as 40 pJ/bit [18]. As process technology advanced, efficiency improved for digital decoders, and some recent designs achieve less than 10 pJ/bit.

Scaling trends have diminished the advantages of sub-V_{T} analog decoders in high-speed applications, but analog implementations may still prove to be more efficient in biocompatible applications, which operate at a lower speed. When digital designs are operated at low speed, the dynamic power is reduced but the static leakage power remains constant. This tends to degrade the decoder's Jpb efficiency, as discussed in Section III-C. The benefits of technology scaling are expected to diminish in future process generations, in which the threshold and supply voltages are now approaching physical limits. We may therefore expect that digital decoder implementations will not benefit as strongly from technology scaling in the future.

B. Technology-Adjusted Performance of ECC Implementations

In this section, we present a normalized comparison of reported iterative decoders in which scaling theory is used to eliminate variations due to process technology and clock frequency. This normalization strategy predicts the performance of decoders implemented in a common 65-nm reference technology, operating at a common throughput of 200 Mb/s. By normalizing decoder results to 65 nm, it is possible to make meaningful associations between design choices and power efficiency, and to make same-technology comparisons of alternative implementations. Several of the more recent LDPC decoders are already implemented in 65 nm, so the scaling adjustment is used primarily to adjust figures for previous designs implemented in 90-nm and larger technologies.

Digital designs may be scaled by application of the power ratio (1), assuming that dynamic power consumption is much greater than leakage losses for the reported digital designs. For scaling analog designs, the scaling ratio is somewhat different because analog ECC implementations are usually based on continuous-time processing with current-mode circuits. The efficiency is therefore linear with V_{DD} and the power ratio is

$$\gamma_A = \frac{C_2 V_{\mathrm{DD2}}}{C_1 V_{\mathrm{DD1}}} = \xi^{\frac{3}{2}}. \tag{2}$$

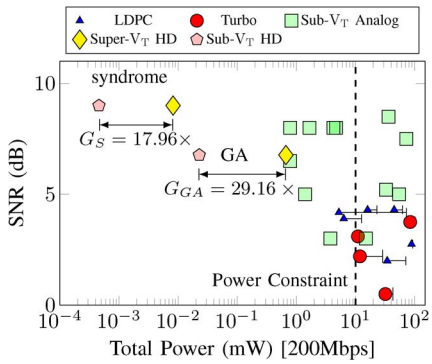

Fig. 3. Comparison of scaled power and performance for reported decoder implementations, including LDPC decoders [1]–[7], Turbo decoders [8]–[10], and sub-V_T analog decoders [14]–[19]. Also shown are the HD decoders synthesized for this work. All power numbers are scaled to a 65-nm technology and normalized to a fixed throughput of 200 Mb/s. The sub-V_T power gains for syndrome and GA decoders are indicated as G_S and G_{GA}, respectively. The SNR is reported as the E_b/N_0 required to achieve an error rate of 10^{-5}. The LDPC and Turbo data points include error bars to indicate the range of uncertainty in static power consumption.

Because digital efficiency scales with ξ^2 while analog efficiency scales with $\xi^{3/2}$, technology scaling should provide greater benefits for digital implementations than analog. This analysis explains why digital designs have closed the efficiency gap with analog designs during the past decade.

Fig. 3 shows the results of our scaling adjustments, revealing the tradeoff between required SNR and decoder power consumption for LDPC, Turbo, and analog decoders. Two low-complexity HD designs are also shown, including a small syndrome decoder and a larger GA decoder, which were synthesized using a standard CMOS cell library. The same decoders were also charaterized for sub-V_T operation, as explained in Section IV. The adjusted results for digital LDPC and Turbo designs account for both dynamic and static power, as explained in Section III-C. Because the static power is not reported for most designs, error bars are shown to indicate the range of uncertainty. In addition to the estimated static power consumption, the digital designs are further adjusted by adding the minimum power needed to operate an analog-to-digital converter, which is required of the digital designs but is not needed for the analog and HD designs. According to Murmann's comprehensive survey [27], the minimum energy requirement in 65-nm technology is 6.5 pJ per sample, or 1.3 mW when sampling at 200 MS/s. This has been added to each of the digital points.

For bio-compatible applications, 10 mW is the chip-scale maximum power constraint, as indicated by a vertical dashed line in Fig. 3. Although this constraint is met by several of the reported analog and LDPC designs, it is not necessarily sufficient since 10 mW is a constraint on the entire implanted device, not just the decoder. Some margin must be left for other functions, including the RF front-end, additional signal processing, and any circuits required for stimulation or actuation functions. The decoder itself should therefore be required to operate well below the 10-mW constraint. As a result, the high-performance digital implementations must be ruled out for use in bio-implantable electronics. The digital HD implementations, on the other hand, are able to operate significantly below the 10-mW limit and have performance similar to analog decoders, as seen in Fig. 3. In Section IV, we demonstrate that when using a sub-V_T supply voltage, the digital HD decoders consume well below 1 mW.

C. Leakage Power in ECC Implementations

For the scaling approach used in Fig. 3, the decoders' power consumption is adjusted to account for differences in reported throughput. For each decoder, the total power is $P_{\text{tot}} = P_s + P_d$, where P_s is a static part due to leakage current and P_d is a dynamic part due to switching. The switching part is proportional to clock frequency and a linear adjustment is applied via the digital power ratio (1). The static part, P_s, does not scale with frequency and therefore establishes a lower bound on power consumption as speed is reduced. Unfortunately, P_s is usually not reported in the literature on iterative decoders. In the rare cases where P_s is reported, it ranges from 3 mW [28] to 41 mW [29].

We used two approaches to account for uncertainty in P_s for LDPC and Turbo decoders. First, we observe that $P_s > 1$ mW in all reported digital implementations (e.g., [3], [29]), which, second, identifies that the reported relative leakage power consumption is between 0.5% [3] and 5% [29] of the total power when operating at the decoder's maximum speed. Fig. 3 reports the minimum estimated P_s and error bars indicate the maximum estimated P_s using these methods.

IV. Sub-V_T Energy Characterization

Sub-threshold (sub-V_T) or weak-inversion operation of digital circuits is an efficient technique to reduce static and dynamic dissipation. The drawback of sub-V_T operation is a severe degradation of the transistor's performance, i.e., propagation delay and reliability. Thus, sub-V_T operation may be considered for designs with low to moderate throughput requirements, and for highly parallel architectures where speed can be recovered through parallel operation. The GA decoder is a highly parallel architecture that is suitable for sub-V_T implementation. In this section, we present power analysis for GA and syndrome decoders that were synthesized for a 65-nm low-power high threshold (LP-HVT), which has a threshold voltage $V_T < 700$ mV. The LP-HVT technology was previously demonstrated to be functional down to $V_{DD} = 250$ mV [30].

The GA and syndrome decoders were synthesized using a commercially available standard LP-HVT super-V_T CMOS technology. When operating at 200 Mb/s, the power consumption of the super-V_T decoders were estimated to be 664 μW (GA) and 8.23 μW (syndrome), as reported in Fig. 3. To evaluate the gains obtained by operating in the sub-V_T domain, their power consumption was estimated as a function of clock frequency and supply voltage.

Toggling information for power estimation is obtained by simulating a fully routed design (including clock tree) with back-annotated timing information. The design is characterized by employing an energy model that uses parameters retrieved from critical path information as well as a traditional *value change dump*-based power simulation. The sub-V_T characterization flow considers static and dynamic energy and provides an accurate energy profile, verified by silicon measurements of previously fabricated sub-V_T circuits [30].

The decoders' sub-V_T energy profiles are shown in Fig. 4(a) and (b). The minimum voltage for reliable operation is indicated by the vertical dashed line in Fig. 4(a). The energy minimum ($E_{\min}$) is found at supply voltages 176 mV and 151 mV for the syndrome and the GA decoder, respectively. At the energy minimum, voltage static and dynamic energy have an equal share, whereas at higher supply voltages, the contribution from switching increases. At such a low V_{DD}, however, the circuit

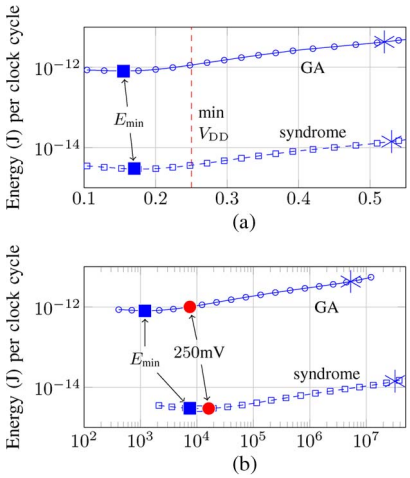

Fig. 4. Sub-V_{T} energy profiling of the $N = 7$ Hamming syndrome and $N = 512$ GA decoders. The minimum energy ($E_{\min}$) points are indicated as squares and realizable minima (operating at 250 mV supply) are indicated by circles. The projected points at 200 Mb/s are indicated by asterisks.

cannot operate reliably [30]. The obtainable energy minima are therefore achieved at 250-mV supply, operating at clock speeds 16 kHz (GA) and 7.5 kHz (syndrome), corresponding to throughputs of 8 Mb/s and 52 kb/s, respectively. At these speeds, the actual power consumption figures are 7.5 nW (GA) and 47.9 pW (syndrome). In order to benchmark these results alongside the comparison data in Fig. 3, the points corresponding to 200 Mb/s are indicated by asterisks ($*$) in Fig. 4(a) and (b). To achieve a throughput of 200 Mb/s, the decoders are operated at 4.7 MHz (GA) and 28 MHz (syndrome), with corresponding increased dynamic power. At this throughput, the decoders are estimated to have a power consumption of 22.77 μW (GA) and 458 nW (syndrome), operating with supply voltages 0.52 V and 0.53 V, respectively.

V. Conclusion

ECCs provide significant performance benefits for wireless communication circuits, but state-of-the-art ECC implementations require too much power to be integrated within the power constraints of bio-implantable devices.

Our results show that iterative HD methods, such as the GA decoder, achieve lower power consumption than the best analog decoders with similar SNR performance. We furthermore showed that sub-mW operation can be realized through sub-V_{T} digital implementation of HD algorithms, which reduces power by nearly 30 times compared to standard super-V_T implementation.

References

[1] S.-W. Yen, S.-Y. Hung, C.-L. Chen, H.-C. Chang, S.-J. Jou, and C.-Y. Lee, "A 5.79-Gb/s energy-efficient multirate LDPC codec chip for IEEE 802.15.3c applications," *IEEE J. Solid-State Circuits*, vol. 47, no. 9, pp. 2246–2257, Sep. 2012.

[2] 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.

[3] T. Mohsenin, D. N. Truong, and B. M. 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.

[4] K. He, J. Sha, L. Li, and Z. Wang, "Low power decoder design for QC-LDPC codes," in *Proc. IEEE Int. Symp. Circuits Syst.*, Jun. 2, 2010, pp. 3937–3940.

[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] M. Mansour and N. Shanbhag, "A 640-Mb/s 2048-bit programmable LDPC decoder chip," *IEEE J. Solid-State Circuits*, vol. 41, no. 3, pp. 684–698, Mar. 2006.

[7] A. Blanksby and C. Howland, "A 690-mW 1-Gb/s 1024-b, rate-1/2 low-density parity-check code decoder," *IEEE J. Solid-State Circuits*, vol. 37, no. 3, pp. 404–412, Mar. 2002.

[8] C.-Y. Lin, C.-C. Wong, and H.-C. Chang, "A multiple code-rate Turbo decoder based on reciprocal dual trellis architecture," in *Proc. IEEE Int. Symp. Circuits Syst.*, 30 May–Jun. 2, 2010, pp. 1496–1499.

[9] W.-T. Lee, S.-H. Lin, C.-C. Tsai, T.-Y. Lee, and Y.-S. Hwang, "A new low-power turbo decoder using HDA-DHDD stopping iteration," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2005, pp. 1040–1043.

[10] P. Ciao, G. Colavolpe, and L. Fanucci, "A parallel VLSI architecture for 1-Gb/s, 2048-b, rate-1/2 Turbo Gallager code decoder," in *Proc. Euromicro Symp. Digital Syst. Des.*, Sep. 2004, pp. 174–181.

[11] Y. Luo, C. Winstead, and P. Chiang, "125Mbps ultra-wideband system evaluation for cortical implant devices," in *Proc. IEEE Eng. Med. Biol. Conf.*, 2012, pp. 779–782.

[12] C. Winstead and Y. Luo, "Error correction circuits for bio-implantable electronics," in *Proc. IEEE Midwest Symp. Circuits Syst.*, 2012, pp. 158–161.

[13] H. Sjöland, J. B. Anderson, C. Bryant, R. Chandra, O. Edfors, A. J. Johansson, N. S. Mazloum, R. Meraji, P. Nilsson, D. Radjen, J. N. Rodrigues, S. M Y. Sherazi, and V. Owall, "A receiver architecture for devices in wireless body area networks," *IEEE J. Emerging Sel. Topics Circuits Syst.*, vol. 2, no. 1, pp. 82–95, Mar. 2012.

[14] M. Moerz, T. Gabara, R. Yan, and J. Hagenauer, "An analog 0.25-μm biCMOS tailbiting MAP decoder," in *Proc. IEEE Int. Solid-State Circuits Conf.*, 2000, pp. 356–357.

[15] A. Demosthenous and J. Taylor, "A 100-Mb/s 2.8-V CMOS current-mode analog Viterbi decoder," *IEEE J. Solid-State Circuits*, vol. 37, no. 7, pp. 904–910, Jul. 2002.

[16] D. Vogrig, A. Gerosa, A. Neviani, A. Gi. Amat, G. Montorsi, and S. Benedetto, "A 0.35-μm CMOS analog Turbo decoder for the 40-bit rate 1/3 UMTS channel code," *IEEE J. Solid-State Circuits*, vol. 40, no. 3, pp. 753–762, Mar. 2005.

[17] S. Hemati, A. Banihashemi, and C. Plett, "A 0.18- 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.

[18] C. Winstead, N. Nguyen, V. C. 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.

[19] R. Meraji, J. B. Anderson, H. Sjoland, and V. Owall, "An analog (7,5) convolutional decoder in 65-nm CMOS for low power wireless applications," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2011, pp. 2881–2884.

[20] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Proc. IEEE Int. Conf. Commun.*, May 1993, vol. 2, pp. 1064–1070.

[21] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[22] D. MacKay and R. Neal, "Near Shannon limit performance of low density parity check codes," *Electron. Lett.*, vol. 33, no. 6, pp. 457–458, Mar. 1997.

[23] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996.

[24] H. A. Loeliger, F. Lustenberger, M. Helfenstein, and F. Tarkoy, "Probability propagation and decoding in analog VLSI," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 837–843, Feb. 2001.

[25] T. Seese, H. Arasaki, G. M. Saidel, and C. R. Davies, "Characterization of tissue morphology, angiogenesis, and temperature in the adaptive response of muscle tissue to chronic heating," *Lab Invest.*, vol. 78, no. 12, pp. 1553–1562, Dec. 1998.

[26] R. H. Dennard, J. Cai, and A. Kumar, "A perspective on today's scaling challenges and possible future directions," *Solid State Electron.*, vol. 51, no. 4, pp. 518–525, Apr. 2007.

[27] B. Murmann, ADC Performance Survey 1997–2012. [Online]. Available: http://www.stanford.edu/murmann/adcsurvey.html

[28] Y. Sun, J. Cavallaro, and T. Ly, "Scalable and low power LDPC decoder design using high level algorithmic synthesis," in *Proc. IEEE Int. SOCC*, Sep. 2009, pp. 267–270.

[29] A. Cevrero, Y. Leblebici, P. Ienne, and A. Burg, "A 5.35 mm2 10GBASE-t Ethernet LDPC decoder chip in 90 nm CMOS," in *Proc. IEEE Solid State Circuits Conf.*, Nov. 2010, pp. 1–4.

[30] O. C. Akgun, J. N. Rodrigues, Y. Leblebici, and V. Owall, "High-level energy estimation in the sub-VT domain: Simulation and measurement of a cardiac event detector," *IEEE Trans. Biomed. Circuits Syst.*, vol. 6, no. 1, pp. 15–27, Feb. 2012.