

# A High-Throughput Low-Power Soft Bit-Flipping LDPC Decoder in 28 nm FD-SOI

Kevin Cushon<sup>†</sup>, Per Larsson-Edefors<sup>†</sup> and Peter Andrekson<sup>‡</sup>

<sup>†</sup>Department of Computer Engineering, Chalmers University of Technology, Gothenburg, Sweden

<sup>‡</sup>Department of Microtechnology and Nanoscience, Chalmers University of Technology, Gothenburg, Sweden

**Abstract**—We present a low-density parity check (LDPC) decoder using the adaptive degeneration (AD) algorithm with a (3600, 3000) LDPC code, integrated in 1.85 mm<sup>2</sup> in 28 nm FD-SOI. With early termination and variable latency decoding, this decoder achieves an optimal energy efficiency of 0.16 pJ/bit and information throughput of 13.6 Gbps with a core supply voltage of 0.4 V. At a core supply voltage of 1.0 V, it achieves 0.58 pJ/bit energy efficiency and 181 Gbps throughput. With constant latency equal to the maximum number of iterations, it achieves optimal energy efficiency of 0.52 pJ/bit and information throughput of 7.2 Gbps at a supply voltage of 0.55 V, and 1.9 pJ/bit energy and 24 Gbps throughput at 1.0 V. The net coding gain at a bit error rate of 10<sup>-12</sup> is 8.7 dB.

## I. INTRODUCTION

Many modern communication systems require forward error correction (FEC) with very high performance in order to meet stringent throughput and error rate requirements in noisy channels. Soft-decision low-density parity check (LDPC) codes are commonly used in such systems. However, decoders using iterative message-passing algorithms such as the min-sum algorithm (MSA) are very costly in terms of silicon area and power, making performance-cost tradeoffs necessary. Prior decoder implementations have addressed these problems with voltage-frequency scaling (VFS) in conjunction with partially parallel or layered architectures [1] [2] [3], serialized message passing [4], bi-directional message passing circuitry [5], or using refresh-free embedded dynamic random access memory (eDRAM) in lieu of registers [6].

In this work, we present an LDPC decoder application-specific integrated circuit (ASIC) based on the adaptive degeneration (AD) algorithm, a low complexity soft decision bit-flipping algorithm. The characteristics of this algorithm and architecture are very favorable for achieving high throughput and low energy consumption, which we demonstrate via measurements conducted on a decoder fabricated in 28 nm FD-SOI with a core area of 1.85 mm<sup>2</sup>. Depending on operating mode and supply voltage, this ASIC achieves throughput of up to 181 Gbps, and energy consumption as low as 0.16 pJ per information bit. This represents an improvement of 2 to 5 times greater throughput per unit area compared to recently published MSA decoder ASICs, and 30 times lower energy consumption. While it achieves lower coding gain than MSA, this tradeoff is very good for systems where throughput and energy consumption are a high priority.

In Section II of this paper, we give a brief description of the AD decoding algorithm and an overview of the implemented

chip. In Section III, we present measured results for error rate performance, throughput, and power consumption. We also analyze these results and present comparisons with a selection of relevant previously published results for fabricated and non-fabricated LDPC decoders. Finally, we present some concluding remarks in Section IV.

## II. ALGORITHM AND SYSTEM ARCHITECTURE

The AD algorithm is initialized with the log-likelihood ratios (LLRs) of symbols received from the channel, which are loaded into the variable node (VN) memories. The messages to the check nodes (CNs) are the sign bits of these memories. These sign bits also constitute the decoded hard decision bits. The CNs perform modulo-2 addition on their inputs; the 1-bit result constitutes the CN-to-VN message. The VNs add, convert, and scale their incoming messages. A degeneration factor, opposite in sign to each VN memory, is also added to the message sum. The degeneration magnitude is usually very small, and serves to produce a “decay and oscillate” behavior in VNs with deadlocked inputs. In many cases, this is sufficient to break the deadlock and allow decoding to complete successfully. However, this is not sufficient to correct stronger trapping sets (i.e., conditions where a small number of erroneous bits reinforce one another through falsely satisfied parity checks). In these cases, if the number of unsatisfied parity checks has not decreased over the previous few iterations, then the degeneration factor is globally set to a larger magnitude for the next iteration only. This technique has been shown to be highly effective for breaking up trapping sets and lowering the error floor. Schematics of the VN and CN are shown in Fig. 1. For a more detailed analysis and description of the algorithm, we refer interested readers to [7].

With only a single memory per VN, and CNs of any degree implementable with an XOR gate, the AD algorithm has lower computational complexity than the MSA. Furthermore, since inter-node messages are single bits, fully parallel decoders with long block lengths can be implemented in VLSI without encountering routing congestion issues.

The LDPC code chosen for this implementation is a (3600, 3000) random regular code with VN degree 6 and CN degree 36. This code was designed to perform well with AD decoding, as it works best with moderate even-numbered VN degrees and large block lengths. The number of LLR quantization bits  $q$  is set to 5, with a standard fixed-point number format of 1 sign bit, 3 integer bits, and 1 fractional bit.

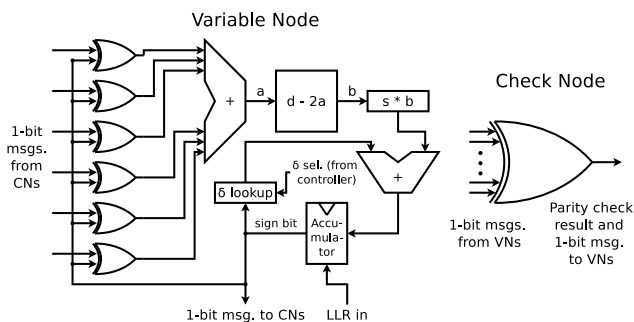


Fig. 1. Simplified variable node and check node schematics. The parameter  $d$  is the degree of the VN, and  $s$  is a constant scaling factor. The degeneration factor  $\delta$  is always opposite in sign to the accumulator, and its magnitude can vary based on an input from the decoder controller.

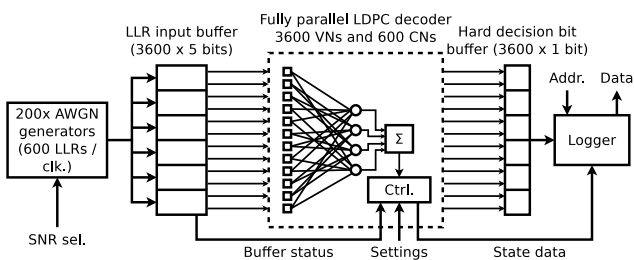


Fig. 2. Chip block diagram showing the overall system and main components.

A block diagram of the chip as fabricated is shown in Fig. 2. In addition to the fully parallel decoder, the chip also contains 200 additive white Gaussian noise (AWGN) generators to generate input vectors entirely on-chip. Each generator consists of a 128-bit xorshift+ pseudo-random number generator (PRNG) [8], and flexible decision trees to map the PRNG output to 5-bit LLRs. The decision thresholds are hard-wired according to the LLR probability distribution functions for signal-to-noise ratios (SNRs) in the region of interest, with the desired SNR selected via dedicated input pins. This technique limits input generation capability to a few pre-defined SNRs, but is very fast and compact. The PRNGs are seeded with random hard-coded values, but can also be re-seeded by an additional PRNG, or set to an arbitrary state through the scan chain. Each generator produces 3 LLRs per clock cycle, so a full frame of 3600 LLRs is generated in 6 clock cycles.

Also present on the chip are buffer memories for the input LLRs and output hard decision (HD) bits, both with a capacity of 1 full frame. In order to minimize their dynamic power consumption, these are implemented as addressable register files rather than shift registers. Finally, a logger counts completed frames, frame errors, and bit errors, and also records key internal state data. This data is readable off-chip via a conventional address/data interface.

### III. CHIP IMPLEMENTATION AND MEASUREMENT RESULTS

The chip is implemented in STMicroelectronics 28 nm FD-SOI technology with 8 metal layers for routing (6 thin, 2 thick). Fig. 3 shows an annotated die microphotograph, and

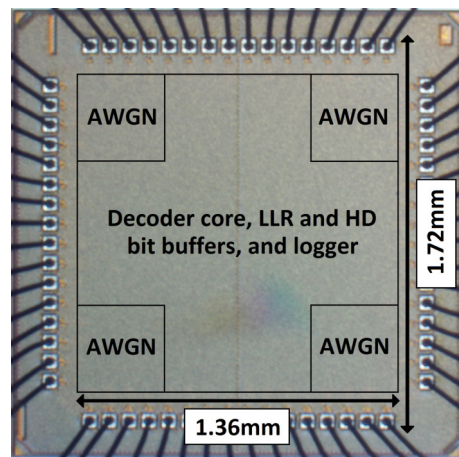


Fig. 3. Die microphotograph with annotations.

TABLE I  
CHIP AND TECHNOLOGY SUMMARY

| Technology        | ST 28 nm FD-SOI       |
|-------------------|-----------------------|
| Metal stack       | 6 thin, 2 thick, 2 RF |
| Core voltage (V)  | 0.36 – 1.1            |
| Std. cell library | Low $V_t$             |
| Core area         | 1.85 mm <sup>2</sup>  |
| Die area          | 2.96 mm <sup>2</sup>  |

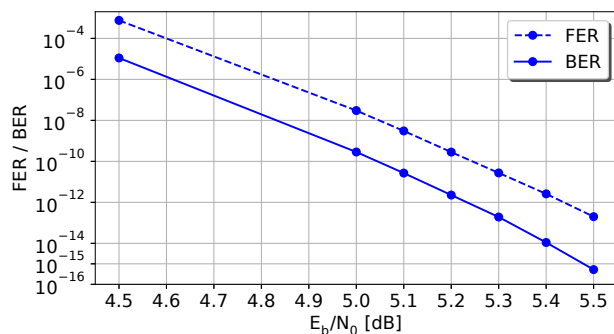


Fig. 4. Measured FER and BER for the (3600, 3000) LDPC code.

Table I contains a summary of the fabrication technology and the physical characteristics of the chip. It has a core area of 1.36 x 1.36 mm (1.85mm<sup>2</sup>) and a die area (including pads) of 1.72 x 1.72 mm (2.96mm<sup>2</sup>). The AWGN generators are 0.53 mm<sup>2</sup>, while the decoder (together with I/O buffer memories) occupy the remaining 1.32 mm<sup>2</sup>.

Measured frame error rate (FER) and bit error rate (BER) performance is plotted in Fig. 4. The maximum number of decoding iterations is set to 49 for these measurements. There is no error floor above a BER of 10<sup>-15</sup>, so this design would be suitable for low BER applications such as storage and optical fiber communication. The SNR (defined as  $E_b/N_0$ ) at a BER of 10<sup>-12</sup> is 5.23 dB, which corresponds to a net coding gain (NCG) of 8.7 dB.

For the following power consumption and throughput measurements, we report results using 49 maximum decoding

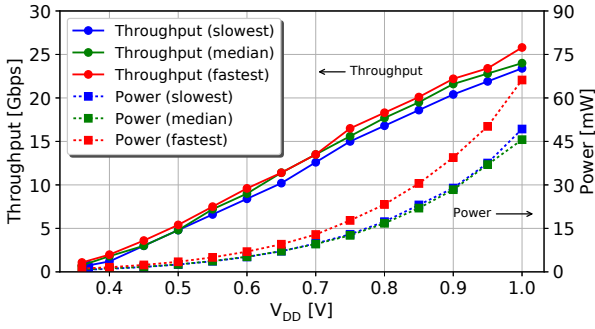


Fig. 5. Power and information throughput at  $E_b/N_0 = 5.4$  dB without ET.

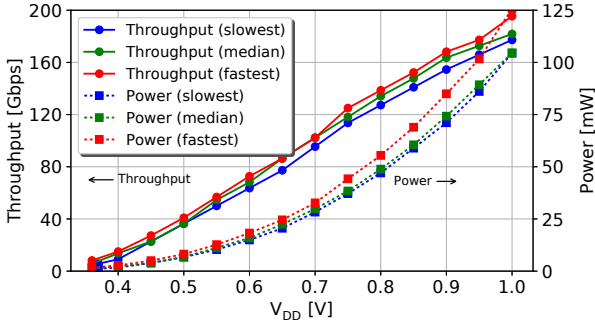


Fig. 6. Power and information throughput at  $E_b/N_0 = 5.4$  dB with ET.

iterations, and we select an SNR of 5.4 dB as representative of a typical operating point for this decoder. The mean iteration count at this SNR is 5.6.

We also report results for two different operating modes: without early termination (ET), and with ET. Without ET, the decoder stops decoding when all parity checks are met, but does not begin the next frame until a total of 50 clock cycles have elapsed (49 decoding iterations of 1 clock cycle each, plus 1 cycle to load and unload the decoder). This represents the minimum throughput and constant latency case. The average duty cycle for this case is 13%. With ET, the decoder stops decoding as soon as all parity checks are met, and begins decoding the next frame as soon as it is available. In this environment, the decoder has an average duty cycle of 94% – since the AWGN generators require 6 clock cycles to generate a frame, the decoder is input-constrained if it finishes decoding in fewer cycles. While much more efficient in this environment, we note that practical use of ET to raise throughput and reduce idle time would require a larger input buffer than the one implemented on this chip, as well as a system that is tolerant of variable decoding latency.

We also report results for the slowest, median, and fastest of the received chips in order to account for process variation. The maximum clock frequency for the median chip ranges from 12.5 MHz at 0.36 V, to 400 MHz at 1.0 V.

Fig. 5 plots power consumption and information throughput across a range of core supply voltages for the case without ET,

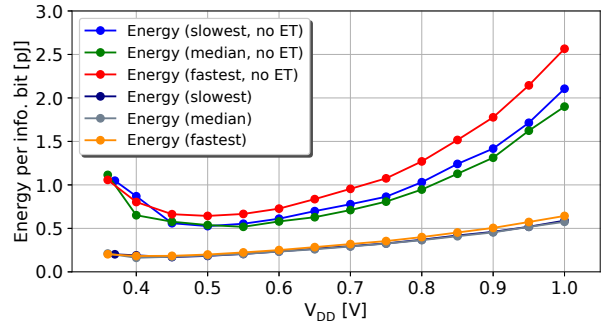


Fig. 7. Energy per decoded bit at  $E_b/N_0 = 5.4$  dB, with and without ET.

while power and information throughput with ET are plotted in Fig. 6. These results exclude the dynamic power of the AWGN generators, but include power consumption from all other sources (i.e., the decoder core, register buffers, logger, and full-chip static power). At the highest tested supply voltage of 1.0 V, the decoder achieves throughput of 24 Gbps without ET, and 181 Gbps with ET.

Fig. 7 plots energy per decoded information bit for both cases, with and without ET. As expected, energy is higher in the without-ET case, because of additional static energy consumption while the decoder is idle. In the without-ET case, the energy optimum for the median chip is 0.52 pJ/bit, and occurs at a supply voltage of 0.55 V. The corresponding throughput at this voltage is 7.2 Gbps. At the maximum supply voltage of 1.0 V, energy consumption is 1.9 pJ/bit. In the with-ET case, the energy optimum for the median chip is 0.16 pJ/bit at a supply voltage of 0.4 V. Throughput at these operating conditions is 13.6 Gbps. Energy does not dramatically increase with supply voltage as it does for the without-ET case, since the decoder spends much less time idle and so leakage energy does not become a large fraction of the total.

Table II presents a summary of key data and comparisons with a selection of previously published LDPC decoders. A state-of-the-art MSA-based LDPC decoder ASIC implemented in 28 nm technology is presented in [3]. Our implementation achieves 4 times higher area efficiency and over 30 times lower energy per bit. However, it should be noted that [3] is a flexible decoder supporting 4 different code rates, and thus incurs additional area, speed, and power costs to support this flexibility. Our decoder and the ones in [1] and [9] implement a single LDPC code.

While [1] is an older decoder design implemented in 65 nm CMOS, it is also highly relevant for comparison because it employs a highly parallel high-throughput architecture, and uses an LDPC code with similar characteristics to the one used in this work (i.e., VN degree of 6 and a rate of approximately 5/6). After process scaling, we obtain area and energy efficiency figures for [1] that are comparable to [2] and [3]. Since our AD-based decoder trades off some coding gain in exchange for greatly improved area- and energy efficiency, it is also relevant to compare it with reduced-complexity

TABLE II  
SUMMARY AND COMPARISONS WITH PRIOR WORKS

|  | This work       |      | [3]             |      | [1]               |                   | [9]                    |                   |
|--|-----------------|------|-----------------|------|-------------------|-------------------|------------------------|-------------------|
| <b>Technology</b>                        | 28 nm FD-SOI    |      | 28 nm CMOS      |      | 65 nm CMOS        |                   | 65 nm CMOS             |                   |
| <b>Source of results</b>                 | Fabricated ASIC |      | Fabricated ASIC |      | Fabricated ASIC   |                   | Post-layout simulation |                   |
| <b>Algorithm</b>                         | AD              |      | MSA             |      | Offset MSA        |                   | Split-row              |                   |
| <b>Scheduling</b>                        | Flooding        |      | Layered         |      | Layered           |                   | Flooding               |                   |
| <b>Block size</b>                        | 3600            |      | 672             |      | 2048              |                   | 2048                   |                   |
| <b>Code rate</b>                         | 5/6             |      | 1/2 – 13/16     |      | 0.84              |                   | 0.84                   |                   |
| <b>Max. iterations</b>                   | 49              |      | 4               |      | 14                |                   | 11                     |                   |
| <b>NCG @ BER = 10<sup>-12</sup> (dB)</b> | 8.7             |      | n/a             |      | 9.0               |                   | n/a                    |                   |
| <b>Core area (mm<sup>2</sup>)</b>        | 1.85            |      | 0.78            |      | 2.14 <sup>†</sup> |                   | 1.94 <sup>†</sup>      |                   |
| <b>Core voltage (V)</b>                  | 0.5             | 1.0  | 0.58            | 0.9  | 0.7               | 1.2               | 0.7                    | 1.3               |
| <b>Clock freq. (MHz)</b>                 | 80              | 400  | 100             | 470  | 160 <sup>†</sup>  | 1120 <sup>†</sup> | 56 <sup>†</sup>        | 312 <sup>†</sup>  |
| <b>Throughput (Gbps)</b>                 | 36.4            | 181  | 3.0             | 18.0 | 10.7 <sup>†</sup> | 76.3 <sup>†</sup> | 26.5 <sup>†</sup>      | 148 <sup>†</sup>  |
| <b>Decoder power (mW)</b>                | 6.84            | 105  | 12.7            | 170  | 69 <sup>†</sup>   | 1382 <sup>†</sup> | 29.8 <sup>†</sup>      | 652 <sup>†</sup>  |
| <b>Area eff. (Gbps / mm<sup>2</sup>)</b> | 19.7            | 97.8 | 3.8             | 23.6 | 5.0 <sup>†</sup>  | 35.7 <sup>†</sup> | 13.7 <sup>†</sup>      | 76.5 <sup>†</sup> |
| <b>Energy eff. (pJ / bit)</b>            | 0.19            | 0.58 | 8.2             | 18.0 | 6.45 <sup>†</sup> | 17.6 <sup>†</sup> | 1.1 <sup>†</sup>       | 4.5 <sup>†</sup>  |

<sup>†</sup> Scaled to 28 nm using scaling factors of 1.6 for clock frequency, 0.4 for area, and 0.3 for energy. These factors are based off our own observations converting LDPC decoder designs from 65 nm CMOS to 28 nm FD-SOI.

MSA decoders which do the same, such as split-row [9] and simplified variable-weight min-sum (svwMS) [10]. In particular, [9] presents post-layout simulation results for a split-row LDPC decoder in 65 nm CMOS. After applying the same scaling, our AD decoder has 20-30% greater area efficiency and approximately 7 times lower energy consumption.

In terms of error correction performance, [1] achieves 0.3 dB higher NCG using a code with similar rate and shorter block length compared to this work. It is more difficult to draw a comparison with [3], since it uses much shorter irregular codes with different rates and a high error floor. However, it would be reasonable to assume correction performance similar to [1] with the same code. Deep BER results are not provided for the split-row decoder of [9], but it reports a NCG loss of 0.3 dB at a BER of 10<sup>-7</sup> compared to [1] using the same LDPC code. In general, we would expect lower NCG from AD compared to MSA, but due to the lower logic and wiring complexity of AD, it is feasible to implement longer block lengths to mitigate this loss – an advantage that we leverage by using a longer 3600-bit LDPC code.

When normalized for process and block length, this work has about half the silicon area of the other decoders summarized in Table II, yet energy consumption is many times lower. This discrepancy can be explained by their architectural differences and the corresponding effect on dynamic power consumption. A fully parallel AD decoder architecture has the

advantage that it is highly static after the initial frame load. There is no bulk data movement, interleaving of frames in a pipeline, or re-use of computational units with different inputs multiple times in a single iteration. This is true for any fully parallel decoder architecture, such as [9]. However, AD has an additional advantage in that the outputs of the VNs change only when their sign bits change, which further suppresses switching activity. As a result, switching activity and dynamic power are very low in proportion to the silicon area.

#### IV. CONCLUSION

We fabricated and tested an LDPC decoder ASIC using the low complexity soft bit-flipping AD algorithm and a (3600, 3000) LDPC code. The chip has a core area of 1.85mm<sup>2</sup> in 28 nm FD-SOI. Depending on operating conditions, it achieves throughput of up to 181 Gbps, and energy consumption as low as 0.16 pJ per information bit. It achieves greater throughput per unit area and consumes 7 to 30 times less energy per bit than state-of-the-art MSA-based LDPC decoders. Thus, this design is highly suitable for high-throughput, low-power applications where some coding gain can be traded off.

#### ACKNOWLEDGEMENTS

This work was funded by a grant from the Knut and Alice Wallenberg Foundation.

The authors would also like to thank Christoffer Fougstedt, Erik Ryman, Lars Norén, and Stavros Giannakopoulos for their help with chip measurements and microphotography.

#### REFERENCES

- [1] 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, April 2010.
- [2] M. Weiner, M. Blagojevic, S. Skotnikov, A. Burg, P. Flatresse, and B. Nikolic, “A scalable 1.5-to-6Gb/s 6.2-to-38.1mW LDPC decoder for 60GHz wireless networks in 28nm UTBB FDSOI,” in *IEEE ISSCC 2014*, Feb 2014, pp. 464–465.
- [3] M. Li, J. W. Weijers, V. Derudder, I. Vos, M. Rykunov, S. Dupont, P. Debacker, A. Dewilde, Y. Huang, L. Van der Perre, and W. Van Thillo, “An energy efficient 18Gbps LDPC decoding processor for 802.11ad in 28nm CMOS,” in *IEEE A-SSCC 2015*, Nov 2015, pp. 1–5.
- [4] A. Darabiha, A. Chan Carusone, and F.R. Kschischang, “Power reduction techniques for LDPC decoders,” *IEEE J. Solid-State Circuits*, vol. 43, no. 8, pp. 1835–1845, Aug. 2008.
- [5] A. Cevrero, Y. Leblebici, P. Ienne, and A. Burg, “A 5.35 mm<sup>2</sup> 10GBASE-T Ethernet LDPC decoder chip in 90 nm CMOS,” in *IEEE A-SSCC 2010*, Nov 2010, pp. 1–4.
- [6] Y. S. Park, D. Blaauw, D. Sylvester, and Z. Zhang, “A 1.6-mm<sup>2</sup> 38-mW 1.5-Gb/s LDPC decoder enabled by refresh-free embedded DRAM,” in *2012 Symposium on VLSI Circuits (VLSIC)*, June 2012, pp. 114–115.
- [7] K. Cushon, P. Larsson-Edefors, and P. Andrekson, “Low-Power 400-Gbps Soft-Decision LDPC FEC for Optical Transport Networks,” *J. Lightw. Technol.*, vol. 34, no. 18, pp. 4304–4311, Sept 2016.
- [8] S. Vigna, “Further scramblings of Marsaglia’s xorshift generators,” *Journal of Computational and Applied Mathematics*, vol. 315, pp. 175–181, 2017.
- [9] 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*, vol. 57, no. 5, pp. 1048–1061, May 2010.
- [10] F. Angarita, J. Valls, V. Almenar, and V. Torres, “Reduced-Complexity Min-Sum Algorithm for Decoding LDPC Codes With Low Error-Floor,” *IEEE Trans. Circuits Syst. I*, vol. 61, no. 7, pp. 2150–2158, July 2014.