Check node unit for LDPC decoders based on one-hot data representation of messages

O. Boncalo✉, A. Amaricai, V. Savin, D. Declercq and F. Ghaffari

A novel check node unit architecture for low-density parity check (LDPC) decoders, which avoids the usage of carry-based comparators for the computation of the required first and second minimum values, is presented. It relies on a one-hot representation of the input messages' magnitude, obtained by q-to-2^q decoders. The two minimums are computed using an OR tree and a modified leading zero counter. The proposed architecture is imprecise, as the second minimum is not computed correctly when it is equal to the first one. The implementation results and the analysis of the error correction capability show that the proposed imprecise unit is highly suited for high rate LDPC codes; it presents up to 30% better hardware cost, a higher working frequency, while the loss of the decoding capability is negligible with respect to standard implementations.

Introduction: low-density parity check (LDPC) codes have been widely adopted in many communication and storage systems, because of their excellent performance under iterative message passing algorithms, such as the belief propagation [1] or the min-sum (MS) decoders [2]. For wired transmissions or storage systems, high-throughput decoders for high rate LDPC codes are often used, and the target throughput requirements (from 10 to 30 Gbit/s [3]) must be achieved with low-complexity decoders.

Among the low-complexity decoding algorithms, MS is probably the most efficient solution, since it provides a good trade-off between coding gain and complexity. The check node unit (CNU) update equation of the MS algorithm requires the search of the two smallest values (first and second minimum) among the check node degree, d_c, input messages [4, 5] and the first minimum index. For high rate applications, where the d_c is very large, this leads to large hardware complexity (interconnects and processing units), especially when fully parallel solutions are used. A typical CNU implementation uses two modules: a sorting module, which arranges the input pairs, and a compare-select tree, based on a compare-select module [4, 5]. The latter block selects two minimum out of the incoming inputs. Both the sorting and the compare-select module use carry propagate comparators and multiplexers. To further improve the implementations of MS decoders, several methods to approximate the second minimum with reduced hardware resources at the check node have been recently proposed [6–9]. Most of these techniques rely on the estimation of the second minimum value by a modified version of the first minimum value [6, 7] and/or on the partial hardware reuse of the circuit used to compute the first minimum [8, 9].

In this Letter, we propose a novel CNU implementation for computing the first and second minimum, which always computes accurately the first minimum value, but computes the second minimum in an imprecise manner. The main novelty is represented by the fact that we employ 'a one-hot representation of the magnitude values that reduces the comparators tree to an OR tree'. The two minimums are obtained using a bitwise OR operation between the new representation of the input messages and by employing a modified leading zero counter (LZC). Hence, the usage of carry propagate comparators is no longer required. With respect to [6–9], our approach is different, as we do not rely on the first minimum value to infer the value of the second minimum.

Proposed CNU: The proposed CNU (as depicted in Fig. 1) uses the following building blocks: q-to-2^q decoders, the OR reduction tree block, modified LZC and first minimum index computation.

This solution relies on an alternative representation of the magnitude values: an array of bits with only one non-zero bit position corresponding to the input value (e.g. the 3-bit value 2 is represented by the 8-bit vector 00000100). q-to-2^q decoders are used to obtain the new representation of the input messages. To compute the two minimum values, a bitwise OR operation is performed between these representations (e.g. a bitwise OR between 1, 2, 4 and 5 will result in the vector 00110110). The least significant one in the vector obtained after the OR operation represents the value of the first minimum, while the second least significant one represents the value of the second minimum. To extract these values, a modified LZC is used; the conventional LZC computes only the position of the least significant one. The proposed CNU cannot compute correctly the second minimum when its value is equal to the first minimum (e.g. for the following inputs 1, 2, 1, 5, 4 the computed first minimum is 1, while the result for the second minimum is 2 instead of 1). Hence, the CNU operation is imprecise.

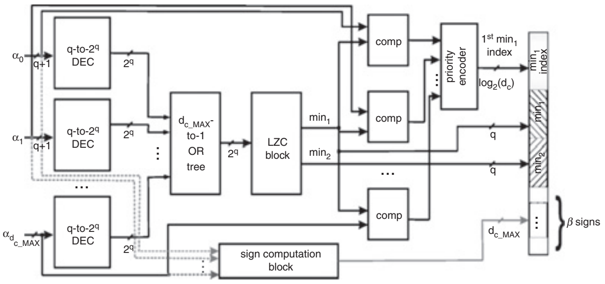

Fig. 1 *Proposed CNU architecture*

The index of the first minimum is computed in two stages: first compare the first minimum with the inputs, and then use a priority encoder during the second stage.

Implementation results: Tables 1 and 2 present the synthesis estimations (cost and performance) for the proposed check node unit architecture, as well as for the comparator-based standard architecture used in [4, 5] LDPC decoders. The results have been obtained using the Xilinx ISE 14.7 synthesis tool for Xilinx Virtex-7 FPGA (speed grade 2) for a 4-bit quantisation of the check node message (1-bit sign and 3-bits of magnitude). The proposed architecture uses 3-to-8 decoders. We analyse the effect of varying the number of inputs on cost and frequency. These check node degrees d_c correspond to different codes and code rates. Both the proposed architecture and the standard check node unit have a single pipeline stage (latched inputs and outputs).

Table 1: Cost estimates for the baseline [4, 5] and the proposed CNU for Virtex-7 FPGA (in look-up tables – flip-flop pairs)

d_c	10	15	20	32	40	64	72
Baseline	141	195	274	430	591	873	976
Proposed	113	155	204	332	383	612	685

Table 2: Frequency estimates for the baseline [4, 5] and the proposed CNU for Virtex-7 FPGA (in MHz)

d_c	10	15	20	32	40	64	72
Baseline	272	246	209	197	182	166	153
Proposed	254	216	197	185	**197**	**176**	**175**

Table 1 suggests that the proposed implementation of the CNU presents a 9 to 30% better cost with respect to the baseline implementation. Significant cost improvements can be obtained if we further increase the number of CNU inputs. Regarding the performance estimates, for smaller d_c, the proposed implementation presents a lower frequency compared with the baseline. However, for a higher number of inputs (such as 40), the proposed CNU equals and surpasses the performance of the baseline CNU.

Error correction performance: In this Section, we evaluate the decoding performance degradation because of the introduced impreciseness in our proposed CNU. The evaluation is performed for MS and self-corrected MS (SCMS) [10] based LDPC decoders. Regular LDPC codes with $d_v = 3$ and $d_c = 6$, 9, 12, 18, 30, corresponding to coding rates $R = 1/2$, 2/3, 3/4, 5/6, 9/10 have been considered for analysis. The bit error rate (BER) performance over the additive white Gaussian noise channel with quadrature phase-shift keying modulation is shown in Fig. 2. For both decoders, exchanged messages are quantised on 4 bits (hence CNU operations are performed on 3-bit values). BER curves have been derived for each coding rate. MS is denoted with black and SCMS with red. Solid curves correspond to the exact CNU and dashed curves to the imprecise CNU proposed in this Letter. At BER = 10^{-5}, the SNR degradation induced by the imprecise CNU varies between 0.25 dB (at $R = 1/2$) and 0.15 dB (at $R = 9/10$) for the MS decoder, and between 0.12 dB (at $R = 1/2$) and 0.08 dB (at $R = 9/10$) for the SCMS decoder.

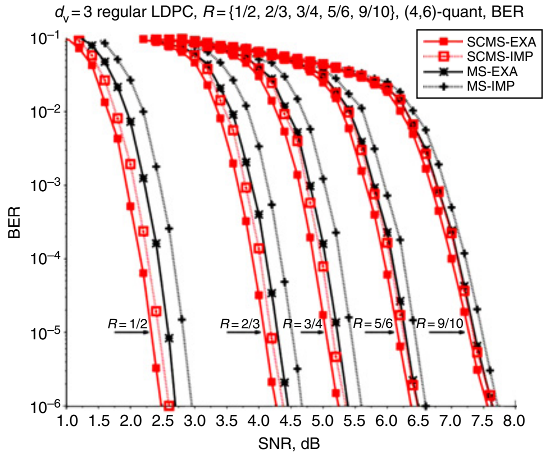

Fig. 2 *BER performance of MS and SCMS decoders, with exact and imprecise CNU*

Conclusion: We have introduced a novel type of CNU architecture for LDPC decoders. It relies on q-to-2^q decoders, OR trees and a modified LZC to compute the first two minimum required for check node operations. It is based on an imprecise operation, as it cannot compute correctly the second minimum when first and second minimums are identical. Synthesis results for Xilinx Virtex-7 FPGAs indicate that the proposed implementation presents from 9 to 30% better hardware cost with respect to the baseline CNU architecture [4, 5]. High cost gains are obtained for a high d_c which corresponds to high rate LDPC codes. Regarding latency, the proposed CNU has better delay for d_c greater than 40. The error correction capability analysis shows that the introduced impreciseness leads to a decoding performance loss of 0.15 dB for MS decoding and of 0.08 dB for SCMS decoding for high rate LDPC codes. This makes the proposed CNU best suited for LDPC codes with high d_c, having significant cost gain and lower latency, while the loss in decoding performance is negligible with respect to the conventional approach.

Acknowledgments: This work was supported by the Franco-Romanian (ANR-UEFISCDI) Joint Research Programme ‘Blanc–2013’ – project DIAMOND.

11 January 2015
doi: 10.1049/el.2015.0108
One or more of the Figures in this Letter are available in colour online.

O. Boncalo and A. Amaricai (*University Politehnica Timisoara, Vasile Parvan Blvd., Nr. 2, Timisoara, Romania*)

✉ E-mail: oana.boncalo@cs.upt.ro

V. Savin (*CEA-LETI, MINATEC Campus, Grenoble, France*)

D. Declercq and F. Ghaffari (*ETIS – ENSEA/University of Cergy-Pontoise/CNRS, France*)

References

1 Fossorier, M., Mihaljevic, M., and Imai, H.: ‘Reduced complexity iterative decoding of low-density parity check codes based on belief propagation’, *IEEE Trans. Commun.*, 1999, **47**, (5), pp. 673–680

2 Chen, J., Dholakia, A., Eleftheriou, E., Fossorier, M., and Hu, X.-Y.: ‘Reduced-complexity decoding of LDPC codes’, *IEEE Trans. Commun.*, 2005, **53**, (8), pp. 1288–1299

3 IEEE P802.3an, 10GBASE-T: http://www.ieee802.org/3/an

4 Park, Y.S., Blaauw, D., Sylvester, D., and Zhang, Z.: ‘Low-power high-throughput LDPC decoder using non-refresh embedded DRAM’, *IEEE J. Solid-State Circuits*, 2014, **49**, (3), pp. 783–794

5 Weiner, M., Nikolic, B., and Zhang, Z.: ‘LDPC decoder architecture for high-data rate personal-area networks’. Proc. Int. Symp on Circuits and Systems, Rio De Janeiro, Brazil, May 2011, pp. 1784–1787

6 Zhang, C., Wang, Z., Sha, J., Li, L., and Lin, J.: ‘Flexible LDPC decoder design for multigigabit-per-second applications’, *IEEE Trans. Circuits Syst. I*, 2010, **57**, (1), pp. 116–124

7 Mohsenin, T., Truong, D., and Baas, B.: ‘A low-complexity message passing algorithm for reduced routing congestion in LDPC decoders’, *IEEE Trans. Circuits Syst. I*, 2010, **57**, (5), pp. 1048–1061

8 Angarita, F., Valls, J., Almenar, V., and Torres, V.: ‘Reduced-complexity min-sum algorithm for decoding LDPC codes with low error-floor’, *IEEE Trans. Circuits Syst. I*, 2014, **61**, (7), pp. 2150–2158

9 Lee, H-C., Cheng, C.-C., and Ueng, Y.-L.: ‘Hardware-friendly probabilistic Min-Sum algorithm for fully-parallel LDPC decoders’. Int. Symp. on Turbo Codes and Iterative Information Processing 2014, Bremen, Germany, August 2014, pp. 102–106

10 Savin, V.: ‘Self-corrected Min-Sum decoding of LDPC codes’. Int. Symp. on Information Theory (ISIT), Toronto, Canada, July 2008, pp. 146–150