TABLE V
TEST SESSIONS BY INTER-CORE MULTILEVEL PARTITION WITH mDFS-L

Circuit	Λ	V	ΛΛ	VV	ΛV	VΛ
$ac3_{\times 10}$	149	147	144	142	141	142
$ac3_{\times 100}$	418	415	412	391	406	385
$ac3_{\times 1000}$	974	965	973	910	959	907
Normalized	107.46%	106.49%	106.62%	100.63%	105.02%	100.00%

C. Inter-Core Partition With Multilevel Framework

The multilevel framework can deal with multicore systems with complicated interconnection structures efficiently, as the nets are partitioned into disjoint sets that can be easily processed. In order to demonstrate the effectiveness of this scheme, some very complicated circuits are created by duplicating the same core many times and generating random interconnect nets to connect these cores. Various multilevel schemes, including single-pass configurations (V and Λ) and two-pass configurations (VV, ΛΛ, VΛ, and ΛV), have been experimented, and the results are shown in Table V.

Experimental results show that various configurations of the multilevel framework achieve roughly the same level of performance. Overall, two-pass configurations (V, V) and (V, Λ) give better results; however, the difference is not significant.

A more important fact is that the multilevel framework achieves good test generation results for complicated circuits. The required test sessions grow at a much lower rate than the cores do. For a 10× increase in the scale of cores, the increase in test sessions is only about 2.3× to 2.8×.

REFERENCES

[1] M. Kaneko and K. Sakaguchi, "Oscillation fault diagnosis for analog circuits based on boundary search with perturbation model," in *Proc. Int. Symp. Circuits Syst.*, 1994, pp. 93–96.

[2] K. Arabi and B. Kaminska, "Oscillation-based test strategy for analog and mixed-signal integrated circuits," in *Proc. VLSI Test Symp.*, 1996, pp. 476–482.

[3] Z. M. Santo, F. Novak, and S. Macek, "Design of oscillation-based test structures for active RC filters," *IEE Proc. Circuits Devices Syst.*, vol. 147, no. 5, pp. 295–302, Oct. 2000.

[4] G. Huertas, D. Vazquez, E. J. Peralias, A. Rueda, and J. L. Huertas, "Practical oscillation-based test of integrated filters," *IEEE Design Test Comput.*, vol. 19, no. 6, pp. 64–72, Nov.–Dec. 2002.

[5] W. C. Wu, C. L. Lee, M. S. Wu, J. E. Chen, and M. Abadir, "Oscillation ring delay test for high performance microprocessor," *J. Electron. Test., Theory Appl.*, vol. 16, nos. 1–2, pp. 147–155, 2000.

[6] M.-S. Wu, C. L. Lee, C. P. Chang, and J. E. Chen, "A testing scheme for crosstalk faults based on the oscillation test signal," in *Proc. Asian Test Symp.*, 2002, pp. 170–175.

[7] K. S.-M. Li, C. Su, Y.-M. Chang, C.-L. Lee, and J.-E. Chen, "IEEE standard 1500 compatible oscillation ring test methodology for interconnect delay and crosstalk detection," *J. Electron. Test., Theory Appl.*, vol. 23, no. 4, pp. 341–355, Aug. 2007.

[8] K. S.-M. Li, C.-L. Lee, Y.-W. Chang, C. Su, and J.-E. Chen, "IEEE standard 1500 compatible interconnect diagnosis for delay and crosstalk faults," *IEEE Trans. Comput.-Aided Design*, vol. 25, no. 11, pp. 2513–2525, Nov. 2006.

[9] K. S.-M. Li, C. Su, Y.-M. Chang, C.-L. Lee, and J.-E. Chen, "Multilevel full-chip routing with testability and yield enhancement," *IEEE Trans. Comput.-Aided Design*, vol. 26, no. 9, pp. 1625–1636, Sep. 2007.

[10] Y.-W. Chang and S.-P. Lin, "MR: A new framework for multilevel full-chip routing," *IEEE Trans. Comput.-Aided Design*, vol. 23, no. 5, pp. 793–800, May 2004.

[11] T.-C. Chen and Y.-W. Chang, "IMF: Interconnect-driven multilevel floorplanning for large-scale building-module designs," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 2005, pp. 159–164.

[12] G. Karypic and V. Kumar. (1998). *hMetis: A Hypergraph Partitioning Package* [Online]. Available: http://glaros.dtc.umn.edu/gkhome/metis/hmetis/download

Block-Circulant RS-LDPC Code: Code Construction and Efficient Decoder Design

Seong-In Hwang and Hanho Lee

***Abstract*— This brief presents a method for constructing block-circulant (BC) Reed–Solomon-based low-density parity-check (RS-LDPC) codes and an efficient decoder design. The proposed construction method results in a BC form of a parity-check matrix from a random parity-check matrix for RS-LDPC codes. A decoder architecture and switch network for BC-RS-LDPC code are then developed based on the new BC parity-check matrix. Thus, an efficient decoder architecture dedicated to a promising class of high-performance BC-RS-LDPC codes is presented for the first time. Moreover, a (2048, 1723) BC-RS-LDPC decoder architecture is designed to demonstrate the efficiency of the presented techniques. Synthesis results show that the proposed decoder requires 1.3-M gates and can operate at 450 MHz to achieve a data throughput of 41 Gb/s with eight iterations.**

***Index Terms*— Belief propagation, block-circulant (BC) parity-check matrix, layered decoding, low-density parity-check (LDPC) codes.**

I. INTRODUCTION

Low-density parity-check (LDPC) codes were discovered by Gallager in the early 1960s [1]. Since the rediscovery of LDPC codes in the late 1990s as near-Shannon limit codes with iterative decoding, many research efforts have been spent on the construction of these codes [2]–[4]. Many applications, such as digital video broadcasting, 10 Gb ethernet (10GBASE-T) [5], broadband wireless access (WiMax), and fourth generation of wireless standards long term evolution, have included LDPC codes. Specifically, 10GBASE-T ethernet adopts (2048, 1723) Reed–Solomon-based LDPC (RS-LDPC) codes. In [6], the authors first introduced RS-LDPC codes, which have large girth and good performance. But RS-LDPC codes have random parity-check matrix that leads to large hardware complexity. On the other hand, the parity-check matrix of quasi-cyclic (QC) LDPC codes [7] consists of circulant permutation matrices. QC-LDPC codes improve hardware efficiency and are proper to highly paralleled decoder architecture. Additionally, an LDPC decoder that has a QC parity-check matrix has a short critical path delay and a simple switch network (SN) structure. Thus, RS-LDPC codes need to have a QC form. Block-circulant (BC) LDPC codes were studied in [8] and [9]. Their BC parity-check matrices are composed by some smaller matrices which are usually cyclic-shifting results of a simple matrix. The parity-check matrix of a common BC-LDPC code is sparse and BC, which is very useful in decoder design [9]. However, to the best of our knowledge, no construction method of a BC parity-check matrix for RS-LDPC codes has been reported.

This brief presents a novel construction method of BC-RS-LDPC code and its efficient decoder design. The proposed construction method generates a BC parity-check matrix for RS-LDPC codes. BC-RS-LDPC codes are able to implement a simpler SN than the conventional Banyan network [10] and quasi-cyclic switch network (QSN) [11]. As a result, a decoder architecture based on

Manuscript received December 21, 2011; revised May 15, 2012; accepted July 21, 2012. Date of publication September 7, 2012; date of current version June 21, 2013. This work was supported by the Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology under Grant 2012R1A1A2007740.

The authors are with the Department of Information and Communication Engineering, Inha University, Incheon 402-751, Korea (e-mail: feverzt@nate.com; hhlee@inha.ac.kr).

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/TVLSI.2012.2210452

BC-RS-LDPC codes has high data throughput and low hardware complexity.

The rest of this brief is organized as follows. Section II describes the design issues related to RS-LDPC codes. Section III presents the proposed construction method of BC-RS-LDPC codes. The SN and overall decoder architecture for BC-RS-LDPC codes are described in Section IV. In Section V, the implementation and comparison are presented. Finally, the conclusion is drawn in Section VI.

II. Design Issues Related to RS-LDPC Codes

RS-LDPC codes were constructed using the strong point of RS codes, which is a maximum-distance separable (MDS) code. Therefore, RS-LDPC codes are MDS codes as well [6]. An MDS code has a minimum distance that is equal to $d_m = n - k - 1$, where n and k denote the code length and code dimension, respectively. Note that the MDS code has the greatest error-correcting capability. The construction steps of RS-LDPC codes [8] are described as follows.

1) The generator matrix G_b can be obtained from the generator polynomial of RS codes.
2) A shortened RS code C_b can be obtained by a linear combination of two rows of G_b.
3) C_b is divided into cosets.
4) Codewords in each coset are expanded into a symbol location vector.
5) Parity-check matrix **H** can be obtained by concatenating symbol location vector sets of cosets.

The null space of the **H** matrix is an RS-LDPC code. Since two rows have at most one 1-component in common, there are no four "1" in **H** that make four corners of a rectangle. Therefore, the Tanner graph of **H** is free of cycle length 4 and its girth is at least 6. **H** can be divided into submatrices. Each submatrix is a permutation but not always a circulant matrix.

In the third step, to divide C_b into cosets, two rows of the generator matrix G_b are used. Because these two rows are independent, linear combinations of these two rows are also independent and a random parity-check matrix is constructed. This random parity-check matrix results in high hardware complexity. Therefore, to eliminate this random property, RS-LDPC codes with a QC form are required. Furthermore, the BC parity-check matrix is sparse and BC, which results in a very efficient decoder design. In Section III, the proposed construction methods of the QC-RS-LDPC codes and BC-RS-LDPC codes are described.

III. Construction of QC-RS-LDPC Codes and BC-RS-LDPC Codes

A. Construction of QC-RS-LDPC Codes

Fig. 1 shows a flowchart for constructing QC-RS-LDPC codes. In constructing QC-RS-LDPC codes, symbol location vectors are created by the following steps.

1) Select one codeword **c** in C_b.
2) Expand codeword **c** into symbol location vector $\mathbf{z}(\mathbf{c})$.
3) Generate the next row by performing a cyclic shift of $\mathbf{z}(\mathbf{c})$.
4) Repeat $p^2 - 2$ times, and then $Z(C_b^{(i)})$ can be obtained.

The proposed QC-RS-LDPC codes retain the two structural properties of RS-LDPC codes, which were introduced in [6].

1) No two rows in the symbol location vector set $Z(C_b^{(i)})$ have any 1-component in common.
2) No two rows in the different symbol location vector set, $Z(C_b^{(i)})$ and $Z(C_b^{(j)})$, have more than one 1-component in common.

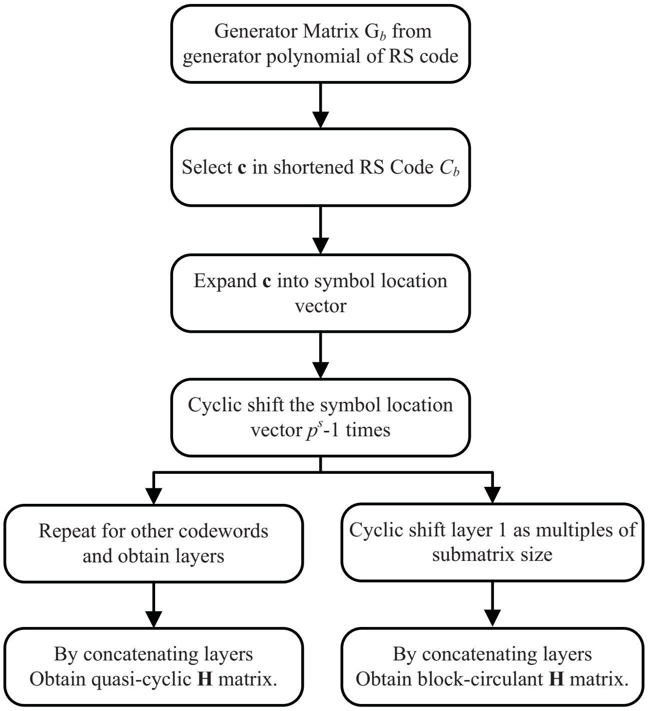

Generator Matrix G_b from generator polynomial of RS code
Select **c** in shortened RS Code C_b
Expand **c** into symbol location vector
Cyclic shift the symbol location vector p^s-1 times
Repeat for other codewords and obtain layers
Cyclic shift layer 1 as multiples of submatrix size
By concatenating layers Obtain quasi-cyclic **H** matrix.
By concatenating layers Obtain block-circulant **H** matrix.

Fig. 1. Flowchart for constructing QC-RS-LDPC codes and BC-RS-LDPC codes.

The QC-RS-LDPC code was constructed based on the RS codes, and each row is generated by performing cyclic shifting number set of the other layer. Thus, QC-RS-LDPC codes maintain these two properties and have submatrices that have only circulant permutation. QC-RS-LDPC codes are able to use SNs such as a Banyan network or QSN. Table I shows a performance comparison of the SNs used in RS-LDPC codes and QC-RS-LDPC codes. SNs for QC-RS-LDPC codes have a short critical path delay and less memory than a Benes network [12], which is used for RS-LDPC codes. However, since the Banyan network and QSN have high hardware complexity, it is necessary to optimize the SN for efficient decoder design.

B. Construction of BC-RS-LDPC Codes

This section proposes the construction method for BC-RS-LDPC codes that have a simple SN. The parity-check matrix of BC-RS-LDPC codes not only has the properties of RS-LDPC codes, but also has the structural properties of QC-RS-LDPC codes. In addition, the parity-check matrix of the proposed BC-RS-LDPC codes is a BC matrix. Fig. 1 shows the flowchart for constructing BC-RS-LDPC codes. Construction steps of BC-RS-LDPC codes are described as follows.

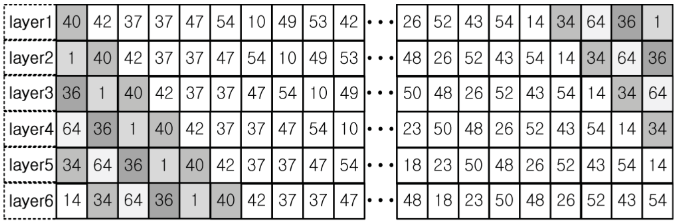
1) Select one codeword **c** in set of codewords C_b.
2) Expand codeword **c** into symbol location vector $\mathbf{z}(\mathbf{c})$. This is the first row of the symbol location vector set of coset 1.
3) Generate the next row by performing a cyclic shift of the symbol location vector $\mathbf{z}(\mathbf{c})$.
4) Repeat $p^s - 2$ times and obtain $Z(C_b^{(1)})$ for layer 1.
5) Cyclic shift layer 1 as one submatrix size p^s, and layer 2 is obtained.
6) By concatenating layers, BC parity-check matrix **H** is obtained.

Similarly, using these steps, the rest of the layers can be obtained. All layers of the parity-check matrix have ρ circulant permutation numbers. Fig. 2 shows an illustration of the BC parity-check matrix for (2048, 1723) BC-RS-LDPC codes. This code is used for the

TABLE I
IMPLEMENTATION RESULTS OF THE SN ARCHITECTURES FOR ONE SUBMATRIX USING 90-nm CMOS 1.1-V LIBRARY

Design	(2048, 1723) RS-LDPC	(2048, 1723) QC-RS-LDPC	
	Benes network [12]	Banyan network [13]	QSN [11]
Total no. of gates (gates)	12 673	9087†	12 216†
Memory size (Bits)	67 584	1152	1152
Critical path delay (ns)	5.5	3.2	3.3
Clock rate (MHz)	180	300	290

† Including the module for generating control signals.



layer1	40	42	37	37	47	54	10	49	53	42	•••	26	52	43	54	14	34	64	36	1
layer2	1	40	42	37	37	47	54	10	49	53	•••	48	26	52	43	54	14	34	64	36
layer3	36	1	40	42	37	37	47	54	10	49	•••	50	48	26	52	43	54	14	34	64
layer4	64	36	1	40	42	37	37	47	54	10	•••	23	50	48	26	52	43	54	14	34
layer5	34	64	36	1	40	42	37	37	47	54	•••	18	23	50	48	26	52	43	54	14
layer6	14	34	64	36	1	40	42	37	37	47	•••	48	18	23	50	48	26	52	43	54

Fig. 2. Illustration of BC parity check matrix for (2048, 1723) BC-RS-LDPC code.

10GBASE-T standard [5]. It presents the cyclic permutation number of the submatrix. The weight of each row is 32 and the weight of each column is 6. As shown in Fig. 2, each layer has a set of numbers that results from the cyclic shifting number set of the other layer.

Fig. 3 shows the bit error rate (BER) performance comparison for (6, 32)-regular (2048, 1723) RS-LDPC code, QC-RS-LDPC code, and BC-RS-LDPC code. For the decoding process, the belief propagation algorithm [2], [3] and the offset min-sum algorithm [13] were used. For performance computation, we assume BPSK transmission over an additive white Gaussian noise channel. Fig. 3 also shows that the proposed QC-RS-LDPC code and BC-RS-LDPC code have almost the same BER performance as the conventional RS-LDPC code has. At a BER of 10^{-7}, the proposed BC-RS-LDPC code performs 2.04 dB from the Shannon limit and achieves a 6.2-dB coding gain over the uncoded BPSK. Although the BC-RS-LDPC code has a simple BC parity-check matrix, there is no performance degradation.

IV. PROPOSED BC-RS-LDPC DECODER ARCHITECTURE

A. Overall Decoder Architecture

The block diagram of the proposed BC-RS-LDPC decoder is illustrated in Fig. 4. The overall decoder was designed based on the offset min-sum algorithm with a layered decoding algorithm. The proposed decoder mainly consists of variable nodes (VNs) for VN computation, check nodes (CNs) for CN computation, and SNs for routing messages. The decoding process of the proposed decoder is as follows.

1) *Initialization:* From communication channel, floating point messages arrive and pass through the SN. The messages are stored in the posterior memory and sent to the SN for CN computation.

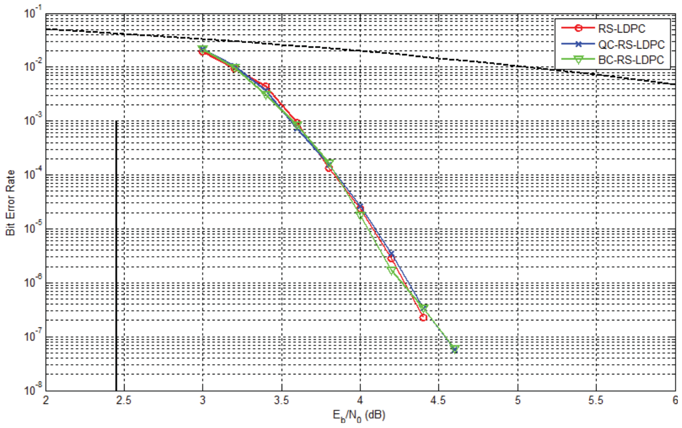


Fig. 3. BER performance of (2048, 1723) RS-LDPC code, QC-RS-LDPC code and BC-RS-LDPC code.

2) *CN Computation:* Messages from VNs that pass through the SN are used for comparison and product in CN block. As mentioned earlier, the CN block has four stages for computing the minimum values and the product of the signs.

3) *Posterior Computation:* Sixty-four results from sixty-four sub-blocks of the CN are routed in the SN and return to the VNs. The results are used for the computation of posterior messages. Then posterior messages are used for parity checking. A typical parity-check equation corresponding to $\hat{\mathbf{c}} \times \mathbf{H}^T = 0$ is computed, where $\mathbf{c} = \{c_1, c_2, c_3, \ldots, c_M\}$ denotes the estimated information bits. If the decoding succeeds, the decoding procedure is terminated; otherwise the steps are repeated.

B. SN for BC-RS-LDPC Codes

Fig. 5 illustrates the submatrix of the parity-check matrix using the conventional construction method and the proposed construction method. The proposed QC-RS-LDPC codes and BC-RS-LDPC codes have only cyclic permutation. In particular, since BC-RS-LDPC codes have a more simple parity-check matrix than QC-RS-LDPC codes, a simple SN can be implemented. Each layer of the BC-RS-LDPC code has ρ cyclic permutation numbers in which all numbers are the same, but the order of the cyclic permutation numbers is different. The SN for (2048, 1723) BC-RS-LPC code is shown in Fig. 6. In the input control block, inputs are fed to different fixed wire blocks according to a control signal, and the fixed wire blocks operate cyclic permutation. Therefore, the proposed SN consists of the fixed wire block and the input control block. Each sub-block of the fixed wire block has a simple structure in which inputs are directly connected to outputs. Thus, the proposed SN has a very short switching time. Table II shows the performance comparison of several SNs. SNs for (2048, 1723) RS-LDPC code, whose parity-check matrix has a 64 × 64 submatrix, are used. That means that the SN consists of 32 sub-blocks in which each sub-block has 64 inputs and 64 outputs. SNs were implemented using a SYNOPSYS design tool and a 90-nm CMOS technology.

Because each subnetwork of a Banyan SN requires around a 9-K gate count, 64 subnetworks require around 580-K gates. On the other hand, the proposed SN for BC-RS-LDPC code requires around 100-K gates. The critical path delay of the proposed SN is 1.4 ns, which is much shorter than the critical path delay of a Banyan SN. While the Banyan SN has seven stages, the proposed SN has only one stage except for the input control block. Due to a reduced critical path delay, the proposed SN can operate at a high clock frequency.

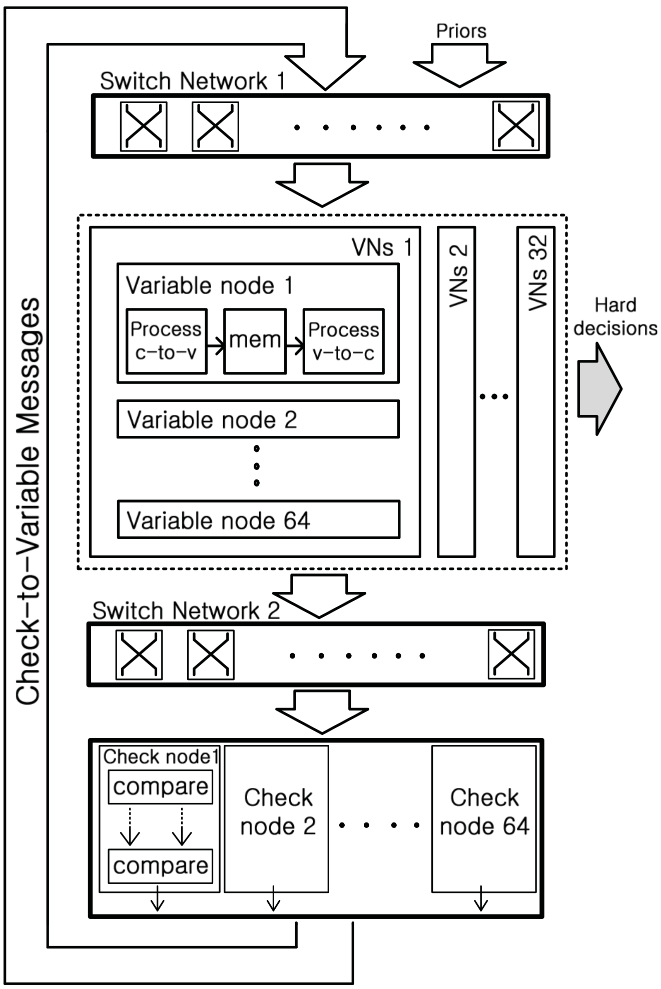
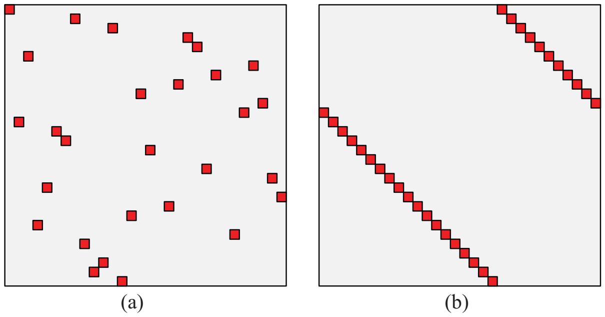


Fig. 4. Block diagram of the proposed BC-RS-LDPC decoder.


(a)

(b)

Fig. 5. Illustration of the submatrix of parity-check matrix using (a) conventional construction method and (b) proposed construction method.

Furthermore, the proposed SN of BC-RS-LDPC code does not need memory to store the control signal, because the input control block is controlled by a clock signal.

C. Pipeline

The proposed decoder architecture has a four-stage pipeline. Fig. 7 shows the four-stage pipeline during one iteration. Four stages in each row are for one layer of (2048, 1723) BC-RS-LDPC code. (2048, 1723) BC-RS-LDPC code has a total of six layers. In the first stage, the VNs to CN messages are computed and routed. The next two stages process the CN computations. In these stages, the minimum values are obtained. The CN block has four compare–select stages. In the fourth stage, messages from the CN block are routed in the SN and are used for VN computation. In the VN computation block, messages from the CN computation block update posterior messages and VN messages. By using the SN of BC-RS-LDPC code, the decoding latency is significantly reduced.

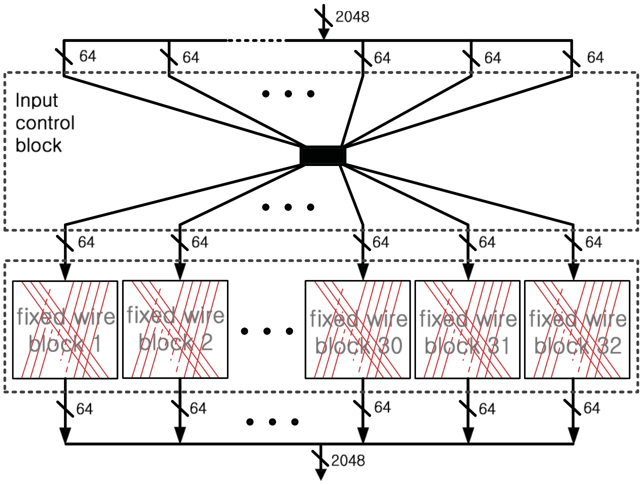


Fig. 6. Proposed SN architecture for (2048, 1723) BC-RS-LDPC code.

TABLE II
IMPLEMENTATION RESULTS OF THE SN ARCHITECTURES FOR (2048, 1723) RS-LDPC USING 90-nm CMOS 1.1-V LIBRARY

Design	Benes network	Banyan network	Proposed SN
Number of stages	13	7	1
Critical path delay (ns)	5.5	3.2	1.4
Required memory (bits)	67 584	1152	None
Gate counts (gates)	811 K	582 K	103 K

V. PERFORMANCE COMPARISON

The proposed (2048, 1723) BC-RS-LDPC decoder was modeled in Verilog HDL and then simulated to verify the functionality using a test pattern generated from a C simulator. After complete verification of the design functionality, the proposed decoder was synthesized using appropriate time and area constraints. Both simulation and synthesis steps were carried out using a SYNOPSYS design tool and a 90-nm CMOS technology optimized for 1.1 V supply voltage. A layered decoding method was also used to improve the speed of convergence and memory requirements. The proposed decoder was designed with an offset min-sum algorithm which has offset factor $\beta = 1.25$. For quantization, a $(q, f) = (6, 2)$ quantization scheme was adopted, where q and f were the total bit sizes of the intrinsic message and the number of fractional bits, respectively.

Table III compares the proposed decoder with previous decoders, where efficiency is defined by the throughput-to-gate-count ratio (Gb/s/M gates). The proposed decoder can operate at a clock frequency of 450 MHz.

The data decoding throughput was calculated according to $T = (f \times N)/(C \times I_{\mathrm{avg}})$, where f is the operating frequency, N is the block length of the code, C is the cycles per iteration, and I_{avg} is the

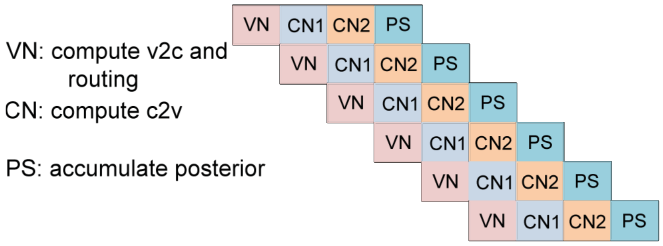


Fig. 7. Four-stage pipeline diagram for (2048, 1723) BC-RS-LDPC decoder.

TABLE III
PERFORMANCE COMPARISON OF (2048, 1723) RS-LDPC DECODERS

Design	Darabiha [15]	Sha [16]	Zhang [14]	Proposed design	
CMOS tech.	90 nm	90 nm	65 nm	90 nm	90 nm
Max. iterations	8	8	8	8	8
Gate counts (gates)	2.23 M	0.82 M	-	1.3 M	
Clock freq. (MHz)	250	400	100	100	450
Decoding latency (ns)	-	-	960	720	180
Throughput (Gb/s)	16	10	6.7	9.1	41
Efficiency (Gb/s/ M gates)	7.17	12.2	-	-	31.54

average number of iterations. At a clock frequency of 450 MHz, the data-decoding throughput was estimated to be approximately $(450 \times 2048)/(9 \times 2.5) = 41$ Gb/s using a maximum of eight decoding iterations. Table III shows that the proposed decoder can provide a higher throughput and better efficiency than the decoders presented in [14]–[16] based on the same code and the same number of iterations. Compared with the decoder in [15], the proposed decoder has much higher data throughput and fewer gate counts. Although the proposed decoder has more gate counts than the decoder in [16], it achieves much higher data throughput and efficiency. The decoder in [14] has 12 stages per iteration, whereas the proposed decoder has 9 stages per iteration. Thus, at the same clock frequency, the proposed decoder has shorter decoding latency and higher data throughput. The proposed BC-RS-LDPC decoder has not only the highest decoding throughput but also a low hardware complexity.

VI. CONCLUSION

A construction method of BC-RS-LDPC code and an efficient BC-RS-LDPC decoder architecture was proposed. The proposed method enables short critical path delay, high throughput, and low hardware complexity. Since BC architecture is simple and easy to implement, the routing complexity is reduced significantly by replacing the conventional SN with a fixed wire SN. The proposed decoder architecture based on BC-RS-LDPC codes has a high data throughput of 41 Gb/s and low hardware complexity. Thus, it can be adopted not only for the application of IEEE 802.3 10GBASE-T Ethernet but also for 40-Gb/s ethernet.

REFERENCES

[1] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.

[2] D. Mackay and R. Neal, "Good codes based on very sparse matrices," in *Proc. 5th IMA Conf. Cryptography Coding*, Oct. 1995, pp. 100–111.

[3] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[4] N. Alon and M. Luby, "A linear time erasure-resilient code with nearly optimal recovery," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 1732–1736, Nov. 1996.

[5] *IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks—Specific Requirements Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*, IEEE Standard 802.3an, Sep. 2006.

[6] I. Djurdjevic, J. Xu, K. Abdel-Ghaffar, and S. Lin, "A class of low-density parity check codes constructed based on Reed–Solomon codes with two information symbols," *IEEE Commun. Lett.*, vol. 7, no. 7, pp. 317–319, Jul. 2003.

[7] M. Fossorier, "Quasi cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Trans. Inf. Theory*, vol. 50, no. 8, pp. 1788–1793, Aug. 2004.

[8] O. Milenkovic, I. B. Djordjevic, and B. Vasic, "Block-circulant low-density parity-check codes for optical communication systems," *IEEE J. Quantum Electron.*, vol. 10, no. 2, pp. 294–299, Mar.–Apr. 2004.

[9] X. Ying and W. Guo, "On the construction of quasi-systematic block-circulant LDPC codes," *IEEE Commun. Lett.*, vol. 11, no. 11, pp. 886–888, Nov. 2007.

[10] S. Olcer, "Decoder architecture for array-code-based LDPC codes," in *Proc. IEEE Global Commun. Conf.*, vol. 22. Dec. 2003, pp. 2046–2050.

[11] X. Chen, S. Lin, and V. Akella, "QSN—a simple circular-shift network for reconfigurable quasi-cyclic LDPC decoders," *IEEE Trans. Circuit Syst. II, Exp. Briefs*, vol. 57, no. 10, pp. 782–786, Oct. 2010.

[12] V. E. Benes, "Optimal rearrangeable multistage connecting networks," *Bell Syst. Technol. J.*, vol. 43, no. 4, pp. 1641–1656, Jul. 1964.

[13] J. Chen and M. P. C. Fossorier, "Density evolution for two improved BP-based decoding algorithms of LDPC codes," *IEEE Commun. Lett.*, vol. 6, no. 5, pp. 208–210, May 2002.

[14] 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.

[15] A. Darabiha, A. C. 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.

[16] J. Sha, J. Lin, Z. Wang, and M. Gao, "Decoder design for RS-based LDPC codes," *IEEE Trans. Circuit Syst. II, Exp. Briefs*, vol. 56, no. 9, pp. 724–728, Sep. 2009.