LDPC Decoder Implementation Using FPGA

Mahdie Kiaee
Department of Computer Engineering
Shahed University
Tehran, Iran
m.kiaee@shahed.ac.ir

Hossein Gharaee
Iran Telecom. Research Center,
Tehran, Iran
Gharaee@itrc.ac.ir

Naser Mohammadzadeh
Department of Computer Engineering
Shahed University
Tehran, Iran
mohammadzadeh@shahed.ac.ir

***Abstract*— This paper presents a partial-parallel LDPC decoder based on sum-product algorithm with high throughput. The hardware implementation of decoder considers design issues with respect to FPGA and time scheduling is proposed based on modified TPMP[1] algorithm in order to reduce the number of clock cycles, hardware resources and power. The decoder is implemented for a code length of 672 whit rate of 3/4, maximum throughput of 3360 Mbps in maximum frequency of 280 MHz and provides power of 150 mW.**

Keywords*— LDPC decoder; hardware implementation; FPGA; time scheduling; TPMP algorithm

I. Introduction

Low Density Parity Check (LDPC) codes are a kind of linear block codes that were discovered by Gallagher in 1962. Some advantage of these codes includes error correction performance as close as to the Shannon limit and sparse parity check matrix [1]. Also, the intermediate complexity of decoding with a considerable level of parallelism in hardware implementation made these codes eligible in the new wireless communication systems [2].

LDPC codes are used in industrial standards like wireless LAN (IEEE 802.11n), Mobile WiMax (IEEE 802.16e) and 10 Gb/s Ethernet (10GBASE-T). In addition, LDPC codes are widely used in satellite television, space communications, magnetic storage in hard disk drives and optical networking [1, 3-5].

The decoding of LDPC codes is based on belief propagation algorithm which is known as Sum-Product algorithm (SPA) and needs complex Calculations. Min-sum algorithm (MSA) is a Simple kind of SPA [6]. The Sum-Product algorithm has better performance in error correction than Min-Sum algorithm, but simple check node unit in MSA needs smaller area and low memory [1]. So, designing a decoder based on SPA with simple check node process unit is an important factor in decoder architecture. LDPC decoders have complex calculations that cause to low delay and high throughput and also need strong hardware architecture [7].

There are three methods for hard ware implementation of LDPC decoders including: 1.fully-parallel 2.serial and 3.semi-parallel [8].

[1] Two-phase-Message-passing

The fully-parallel architecture implements all check nodes and variable nodes of parity check matrix as a process unit, the serial architecture implements only one check node unit and variable node unit and semi-parallel architecture is between fully-parallel and serial [8].

Regardless of parallelism, time scheduling has strong effect on the implementation of decoder. The timing of decoder is done in two ways including Tow-phase-Message-passing (TPMP) decoding and Turbo-Decode-Message-Passing (TDMP) decoding that is known as layered decoding. Convergence of decoding in layer decoding is two times faster than TPMP with 50% reduction in iteration [9, 10].

Two-phase-message-passing algorithm has optimal error-correction performance, but large exponential numbers, look-up table and multiplicative operations increase hardware difficulty [11]. Some simplified algorithms are proposed based on TPMP. Sum-Product Log-Domain algorithm uses log-likelihood ratio (LLR) which avoids exponential computations and numerical instability. Min-Sum algorithm has less complexity but suffers from heavy performance loss. The layer decoding is the most common way in LDPC decoding. In recent years, most of decoders were based on TPMP, but today due to higher convergence speed of TDMP, using the layer decoding is spread [12].

A. Related work

In TPMP, updating of check nodes and variable nodes is performed in discrete units, but in layered architecture, parity check matrix is divided into *m* layers and processing is done layer by layer [10, 13-15]. In [16], an innovative dual-shift stochastic-detection (DSSD) technique is proposed to Deal with partial-parallel cascaded TPMP decoder weaknesses that mitigates computational resources [16]

There are different Parallelism methods in check node and variable node units. In first type, M check node units (CNUs) and N variable node units (VNUs) operate in parallel and per iteration needs $2\times Z$ clock cycles for row and column processing. In the second type, *Z* check node processing units and *Z* variable node processing units can
operate simultaneously. So $W_c + W_v$ clock cycles are needed for column and row updating. Different methods such as *Z/4* of CNUs and *Z/4* of VNUs, or *Z* CNUs and N**Z* VNUs or *2*Z* CNUs and *Z* VNUs are implemented too [17]. The overall architecture of layer decoding is shown in Fig. 1. In [18], a fully pipelined QC-LDPC decoder is presented including M

check nodes and N variable nodes with high parallel degree that is implemented based on Quasi-Cyclic features and layered decoding through efficient utilization of permutation network and small check node design.

In layer decoding deriving calculated values from RAM memory should be done with no data conflict. So, the idle time to calculate the correct information in layer decoding is provided for each layer in [19]. In [20], a half-row layered decoder with reduced routing network is presented in which variable node parallelism equals to half of code length. Routing network is eliminated by changing the shift value of each block data sent from variable node to check node. In half-row design, the parallel degree for variable node is reduced to 336 from 672 but the degree of check node remains 42. Due to the complexity of check node unit and high density of hardware resources used in check node unit and permutation network, proposing a reduced parallel degree architecture is necessary [17].

The RAM memory that is used to store medial messages is an important factor for the design of decoder. In some designs, registers due to faster and spread access at the expense of high power and high area are preferred [21]. In most recent designs of LDPC codes, permutation network is used for transferring medial messages that most of the hardware resources are allocated to it [13, 14, 17-19, 22-24]. Therefore, removal of permutation network without creating complexity in routing between nodes is important. Fixed connections, in addition to reducing hardware resources, increase the flexibility of decoder for both regular and non-regular parity check matrix. Also provides the capability of reconfiguration to implement both structured and random codes without the limitations of parity check matrix [25]. Parity check matrixes of LDPC decoders include regular [11, 26] and irregular [27] structures. Irregular codes have a better performance than the regular codes and provide better protection for input code word and provided greater reliability for data, because the codes with greater degree converge faster and assist the codes with less degree. The architecture design for irregular codes is more challengeable and should be designed flexible to support different matrixes [19].

There is multi-dimensional design space for LDPC decoder consist of scheduling, optimization of check node simplification, parallelism, pipeline stage optimization and etc. Designing a high throughput decoder should be based on multi-dimensional optimizations that depend on the limitations such as power, area and hardware resources [20].

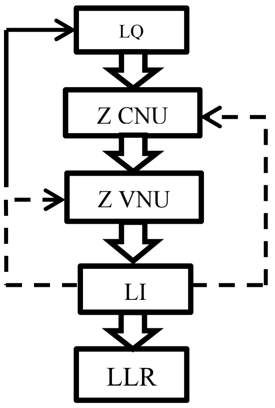

Fig. 1.overall architecture of layer decoding [4]

The architecture of [23] shows that optimal designing requires a tradeoff between hardware complexity, throughput, and performance.

Many decoding method are proposed According to the iterative decoding which include high and low parallel degree. Decoders with high parallelism have short decoding delay and high throughput, but the silicon area is large. In contrast, decoders with low degree of parallelism require less processing units and memories with higher density instead of separate registers, so the area is smaller and Lower throughput is provided [12].

In this article, a LDPC decoder for LAN (IEEE 802.11ad) standard, 3/4 code rate and the code length of 672 is implemented. The architecture of LDPC code is based on sum-product algorithm. In order to creating an acceptable trade-off between parallelism level and maximum throughput, a proposed scheduling with semi-parallel structure is used. By creating pipeline and simultaneity in variable and check node operations with a proper degree of parallelism, the decoding clock cycles will decrease.

The rest of the paper is organized as follows. Section 2, reviews the construction of LDPC codes and the decoding Process, Section 3 describes the proposed decoder architecture and pipeline schedule. In Section 4, implementation results are presented and discussed. Section 5 concludes the paper.

B. Introduction of LDPC codes

A LDPC code is defined with a parity check matrix $H_{m\times n}$ which n is code length (number of bits in code word) and m is the number of parity check equations. Parity check matrix is shown by a bipartite tanner graph consists of variable nodes (VNs) for each column and check nodes (CNs) for each row. The tanner graph of parity check matrix H is shown in Fig. 2 [6].

C. Sum-Product Log-Domain algorithm

For simplifying the computation in sum-product algorithm, the log likelihood ratio of prior (messages received from channel) and posterior (medial messages transferred between check nodes and variable nodes) probability is used. The decoding process consists of three steps including initializing variable nodes with *L* vector, check nodes processes and variable nodes processes.

According to LLR, the log likelihood ratio of prior probability of *i*th bit is shown in (1). Then the L_i values are put into vector $L = [L_1\ L_2 \dots\ L_n]$.

$$L_i = log\left(\frac{1-P_i^{int}}{P_i^{int}}\right) = LLR\left(P_i^{int}\right) \qquad (1)$$

The check node and variable node processes are shown in (2) and (3):

$$E_{ij} = 2\tanh^{-1}\left(\prod_{i'\in B_j, i'\neq i} \tanh\frac{B_{ij}}{2}\right) \qquad (2)$$

$$A_j = L_j + \sum_{i'\in M_j} E_{i'j} \qquad (3)$$

In final step, Hard-Decision is made based on (3) to derive the vector z. The code word is estimated in (4) [19].

$$H.Z^T = 0 \qquad (4)$$

II. ARCHITECTURE DESIGN

As mentioned in previous sections, in addition to parallelism, the timing of decoding affects the design of decoder. In layer decoding, the data dependency between each is a considerable issue that affects the pipeline. Compared with layer decoding, TPMP decoding has less data dependency with the expense of doubling the number of iterations in the same BER [20].

In decoding process due to complexity of As mentioned in previous sections, in addition to parallelism, the timing of decoding affects the design of decoder. In layer decoding, the data dependency between each is a considerable issue that affects the pipeline. Compared with layer decoding, TPMP decoding has less data dependency with the expense of doubling the number of iterations in the same BER [20].

In decoding process due to complexity of E_{ij} in log likelihood ratio decoding, the E_{ij} converts to the (5): [3]

$$E_{ij} = 2\tanh^{-1}\left(\prod_{i' \in B_j, i' \neq i} \tanh\frac{B_{ij}}{2}\right) \qquad (5)$$

So, hardware complexity is reduced due to two kinds of LUT for $\tanh$ and $\tanh^{-1}$. In order to simplifying the operation of check node, E_{ij} is divided into several modules, including tanh LUT, multiplier Unit and $\tanh^{-1}$ LUT. Matrix row multiplication is conducted by using cyclic shift register which reduces hardware resources with hardware reuse. Block diagram of check node unit is shown in Fig. 3. After n clock cycles required for variable node processes and checking all the possibilities of A_j, the binary vector of length n bits (the number of columns of H matrix) will be start. Block diagram of variable node is shown in Fig. 4.

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

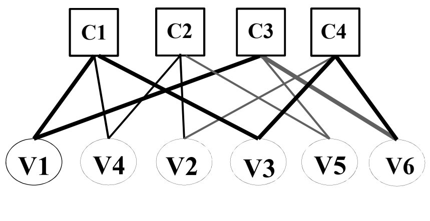

Fig. 2.tanner graph of parity check matrix H

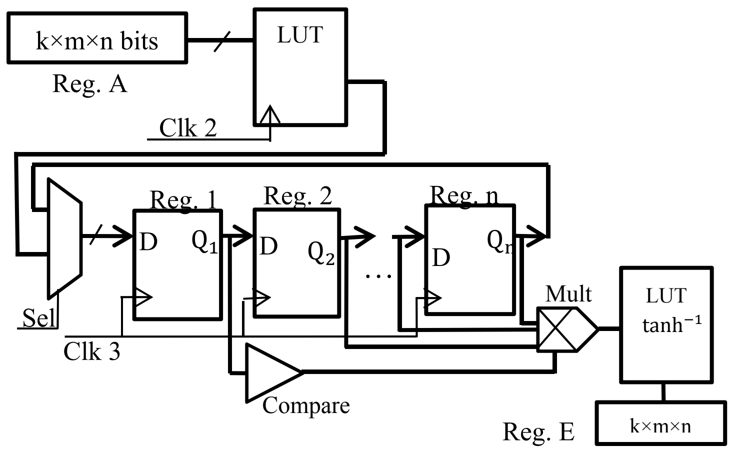

Fig. 3. Block diagram of check node unit

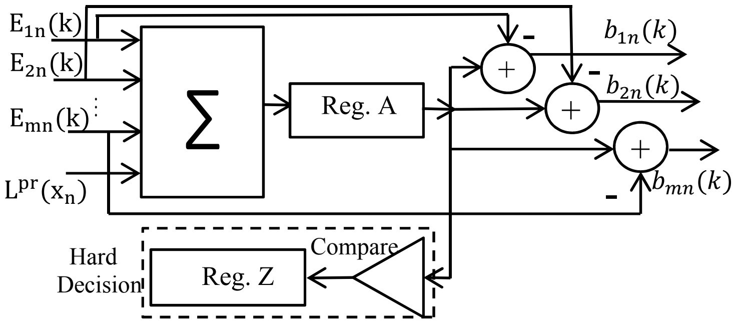

Fig. 4.Block diagram of variable node

III. TIME SCHEDULING AND PIPELINE

Because of better error correction performance of TPMP decoding than layer decoding, a new time scheduling is proposed based on TPMP with simultaneous operation of rows and columns without the interfering of data. In proposed architecture, the number of clock cycles equals to m+5 which m is the number of rows in parity check matrix and the constant 5 is related to decoding operations. In previously proposed layered architectures the number of clock cycles required for row and column processing is $2 \times Z$ or $(W_r + W_c) \times N_{layer}$ according to parallelism degree. So, the number of clock cycles for a decoder according to parity check matrix with $m \times n$ cyclic sub-matrix of size $Z \times Z$, is different based on row and column weight. Reducing the clock cycles without increasing the parallelism between CNUs and VNUs is not possible. In TPMP, if all three operations on check node block be conducted sequentially, $3 \times m$ clocks are required. Operation of variable node unit is done in n clocks which equal to columns of parity check matrix. Finally $3 \times m + n + 1$ clock cycles are required that constant 1 value is related to hard decision.

As mentioned above, decoding speed and throughput are the major issues in LDPC decoder implementation. By overlapping of check node and variable node operations, the clock cycles and hardware resources will be decreased with increasing of decoding throughput.

In this article, two sets of pipelines related to check nodes and variable nodes processing are provided. The first proposed pipeline is related to medial operations of CNU, including $tanh$, row multiplication and $tanh^{-1}$. The Second one is related to simultaneity of check node and variable node operations. If the size of parity check matrix H considered 4×6, decoding timing without simultaneity is shown in Fig. 5. Three stages of check node operations are done simultaneously and Variable node operations consist of column addition begin after check node process and take *n* clock cycles. Check node operations include r_m, Tr_m, MTr_m and $T^{-1}r_m$ are sequentially related to row initialization, tanh of row elements, row multiplication and $tanh^{-1}$ of row elements and add c_n shows addition of elements of a column. Second kind of pipeline is shown in Fig. 6 which consists of simultaneous operation of CNU and VNU. After check node operation of 1st and 2nd row in 5th clock, addition of variable nodes begin with 3rd row operation and the operations of other columns will be done in the same way. Thus by proposed two-stage pipeline, the number of clock cycles has reduced.

IV. Parallelism

Simultaneous operations of variable nodes and check nodes reduce clock cycle, Moreover pipeline between processing units and re-use of the hardware, reduce hardware complexity.

In addition, cyclic shift registers in check node processing unit reduce multipliers. Also, due to simultaneous operations in VNU, only two rows of the

CNU				
Cycle 1	r_1			
Cycle 2	Tr_1	r_2		
Cycle 3	MTr_1	Tr_2	r_3	
Cycle 4	$T^{-1}r_1$	MTr_2	Tr_3	r_4
Cycle 5		$T^{-1}r_2$	MTr_2	Tr_2
Cycle 6			$T^{-1}r_2$	MTr_2
Cycle 7				$T^{-1}r_3$

Fig. 5.Pipeline of CNU and VNU separately

CNU&VNU				
Cycle 1	r_1			
Cycle 2	Tr_1	r_2		
Cycle 3	MTr_1	Tr_2	r_3	
Cycle 4	$T^{-1}r_1$	MTr_2	Tr_3	r_4
Cycle 5		$T^{-1}r_2$	MTr_3	Tr_4
Cycle 6		$cr_1 + cr_2$	$T^{-1}r_3$	MTr_4
Cycle 7			$cr_{12} + cr_3$	$T^{-1}r_4$
Cycle 8				$cr_{123} + cr_4$

Fig. 6.Pipeline of CNU and VNU simultaneously

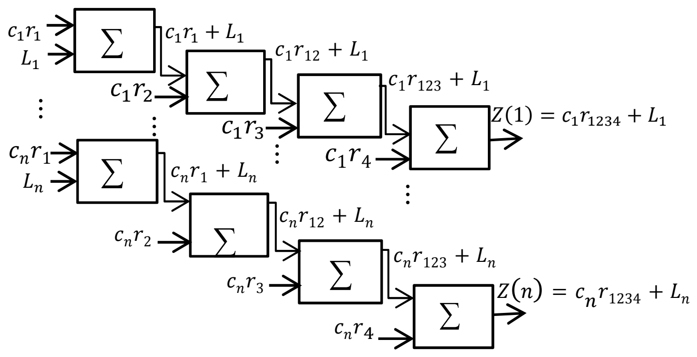

Fig. 7. parallelism of variable node unit adders

TABLE I. Complexity of clock cycles

		TPMP		layered		Proposed timing	
Parallel units	VN	$n \times Z$	1	n	Z	1	N
	CN	$m \times Z$	1	m	Z	1	m
Clock cycle		2	$(m + n) \times Z$	$2 \times Z$	$Wc + Wr$	$Z \times m + 5$	$Z + 5$

TABLE II. Synthesis results of LDPC decoder for IEEE 802.11ad on XILINX VIRTEX6

	[18]	[16]	[23]	Proposed		
Code Length	682-MSA	4489-MSA	2304-MSA	4489	2204	672
Number of Slice Registers	N FIFO	32435 Block RAM= N+M	3086 Block RAM=15	91596	44970	13712
Number of Slice LUTs	35668	63453	13555	60562	58632	56832
Number of occupied slice	13229	-	4446	23744	11872	2968
Number of fully Used LUT- FF pairs	-	29760	3086	39946	19973	6090

matrix elements are added together in a column order per clock cycle. So, the number of required adders equal to the number of matrix columns. Tree adder is proposed for VNU, which only one addition for per column is done in each clock cycle. Variable nodes parallelism is shown in Fig. 7. Another parallel operation in decoding corresponds to check nodes and variable node processors.

Given that the number of clocks needed to complete per iteration is equal to m + 5, the greater size of m will also increase the number of clock cycles. Thus, by parallelism of several decoding units including CNU and VNU, the number of clock cycles reduces. Reducing the number of clock cycles depends on the amount of increasing of parallelism degree. By two parallel decoding units operating together, the number of clock cycles reduced to 1/2.

The number of CNUs and VNUs working in parallel, equal to the number of layers. In fact, the number of row layers or column layers in parity check matrix of LDPC code defines the parallel degree of decoding and processing is done for each layer separately. Proposed parallelism in this article reduces the decoder processor units. Table 1 compares the complexity of clock cycle in proposed method with earlier scholars.

V. IMPLEMENTATION RESULT

A LDPC decoder which supports LAN (IEEE 802.11ad) standard has been synthesized on a XILINX VIRTEX6. The base parity check matrix size is 4×16 with the sub-matrix of size 42×42. Table 2 shows the fixed-point simulation results of TPMP decoding mode for the rate of

TABLE III. Throughput and area comparison of proposed and published LDPC decoder implementation

	[19]	[4]	[24]	This work
Code rate	5/6	1/2	1/2, 2/3, 3/4, 5/6	1/2, 3/4
Code length	1296	2304	1944	672
Algorithm	MSA TDMP	MSA TDMP	MSA TDMP	SPA TPMP
Freq. (MHZ)	230	100	250	280
Max Iteration	5	10	1-7	10
Throughput (Mbps)	767	183	672	3360
Area (cm^2)	3.12	6.25	3.67	3.4
Power (mw)	-	242	171.07	150

1/2 and 3/4 and 672-bit code length. The iteration number of our proposed decoder is set to 10 iterations. The VLSI implementation results show that the proposed decoder occupies an area of $3.4mm^2$ and achieves maximum decoding throughput of 3360 Mbps with maximum 10 iterations in TPMP decoding mode. The estimated power consumption is 150 mW when decoding at 280 MHz. In Table 2, the logic resource utilization of the FPGA is shown and the proposed architecture is compared with [16, 18], and [23]. Table 3 compares this decoder with the state of-the-art LDPC decoder of [4, 19] and [24]. In the proposed decoder in [18] and [23] permutation network is used for transferring medial messages between variable nodes and check nodes and [16] implements a layered decoder with Min-Sum algorithm. The recent architectures are compared with proposed architecture with new time scheduling in this article. Table 3 shows the results.

VI. CONCLUSION

An efficient partially parallel decoder architecture based on sum-product algorithm for LDPC decoder has been proposed in this paper. The proposed architecture has eliminated the permutation network for transferring data between check nodes and variable nodes, so complexity overhead of the switch network has removed. With synchronization between process units, based on proposed pipeline between VNUs and CNUs, the number of clock cycles, hardware resources and power has reduced. The decoder Block Processing Unit is proposed for rate 1/2 and 3/4 of length 672. The decoder have been implemented on FPGA the results show that decoders can achieve throughput of 3360Mbps.

REFERENCES

[1] X. Pan, X.-f. Lu, M.-q. Li, and R.-f. Song, "A high throughput LDPC decoder in CMMB based on virtual radio," in *Wireless Communications and Networking Conference Workshops (WCNCW), 2013 IEEE*, 2013, pp. 95-99.

[2] A. Shevchenko, R. Maslennikov, and A. Maltsev, "Comparative analysis of different hardware decoder architectures for IEEE 802.11 ad LDPC code," in *MELECON 2014-2014 17th IEEE Mediterranean Electrotechnical Conference*, 2014, pp. 415-420.

[3] J. Kim and W. Sung, "Rate-0.96 LDPC decoding VLSI for soft-decision error correction of NAND flash memory," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 22, pp. 1004-1015, 2014.

[4] T. Heidari and A. Jannesari, "Design of high-Throughput QC-LDPC Decoder for WiMAX standard," in *2013 21st Iranian Conference on Electrical Engineering (ICEE)*, 2013, pp. 1-4.

[5] F. Angarita, J. Valls, V. Almenar, and V. Torres, "Reduced-complexity min-sum algorithm for decoding LDPC codes with low error-floor," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 61, pp. 2150-2158, 2014.

[6] B. Belean, S. Nedevschi, and M. Borda, "Application specific hardware architecture for high-throughput short-length LDPC decoders," in *Intelligent Computer Communication and Processing (ICCP), 2013 IEEE International Conference on*, 2013, pp. 307-310.

[7] J. Andrade, G. Falcao, and V. Silva, "Flexible design of wide-pipeline-based WiMAX QC-LDPC decoder architectures on FPGAs using high-level synthesis," *Electronics Letters,* vol. 50, pp. 839-840, 2014.

[8] C. Beuschel, "Fully programmable LDPC decoder hardware architectures," Universität Ulm, 2010.

[9] S.-I. Hwang and H. Lee, "Block-circulant RS-LDPC Code: code construction and efficient decoder design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 21, pp. 1337-1341, 2013.

[10] S. Huang, D. Bao, B. Xiang, Y. Chen, and X. Zeng, "A flexible LDPC decoder architecture supporting two decoding algorithms," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, 2010, pp. 3929-3932.

[11] Y.-H. Chen, C.-L. Chu, and J.-S. He, "FPGA implementation and verification of LDPC minimum sum algorithm decoder with weight (3, 6) regular parity check matrix," in *Electronic Measurement & Instruments (ICEMI), 2013 IEEE 11th International Conference on*, 2013, pp. 682-686.

[12] B. Xiang, D. Bao, S. Huang, and X. Zeng, "An 847–955 Mb/s 342–397 mW dual-path fully-overlapped QC-LDPC decoder for WiMAX system in 0.13 m CMOS," *IEEE Journal of Solid-State Circuits,* vol. 46, pp. 1416-1432, 2011.

[13] Y. S. Park, D. Blaauw, D. Sylvester, and Z. Zhang, "Low-power high-throughput LDPC decoder using non-refresh embedded DRAM," *IEEE Journal of Solid-State Circuits,* vol. 49, pp. 783-794, 2014.

[14] X. Zhao, Z. Chen, X. Peng, D. Zhou, and S. Goto, "High-parallel performance-aware LDPC decoder IP core design for WiMAX," in *2013 IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2013, pp. 1136-1139.

[15] Y. Sun and J. R. Cavallaro, "VLSI architecture for layered decoding of QC-LDPC codes with high circulant weight," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 21, pp. 1960-1964, 2013.

[16] M. H. L. Lim and W. L. Goh, "High-throughput dual-shift stochastic-detection quasi-cyclic LDPC decoder," in *Information, Communications and Signal Processing (ICICS) 2013 9th International Conference on*, 2013, pp. 1-5.

[17] M. Awais, A. Singh, E. Boutillon, and G. Masera, "A novel architecture for scalable, high throughput, multi-standard LDPC decoder," in *Digital System Design (DSD), 2011 14th Euromicro Conference on*, 2011, pp. 340-347.

[18] S. A. Zied, A. T. Sayed, and R. Guindi, "Configurable low complexity decoder architecture for Quasi-Cyclic LDPC codes," in *Software, Telecommunications and Computer Networks (SoftCOM), 2013 21st International Conference on*, 2013, pp. 1-5.

[19] Z. Luan, Y. Pei, and N. Ge, "A fast convergence and area-efficient decoder for quasi-cyclic low-density parity-check codes," in *2013 19th Asia-Pacific Conference on Communications (APCC)*, 2013, pp. 458-462.

[20] M. Li, F. Naessens, P. Debacker, P. Raghavan, C. Desset, M. Li, *et al.*, "An area and energy efficient half-row-paralleled layer LDPC decoder for the 802.11 AD standard," in *SiPS 2013 Proceedings*, 2013, pp. 112-117.

[21] Y. S. Park, D. Blaauw, D. Sylvester, and Z. Zhang, "A 1.6-mm 2 38-mW 1.5-Gb/s LDPC decoder enabled by refresh-free embedded DRAM," in *2012 Symposium on VLSI Circuits (VLSIC)*, 2012, pp. 114-115.

[22] C.-W. Sham, X. Chen, F. C. Lau, Y. Zhao, and W. M. Tam, "A 2.0 Gb/s throughput decoder for QC-LDPC convolutional codes," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 60, pp. 1857-1869, 2013.

[23] J. Wetcharungsri, N. Buabthong, S. Jantarachote, P. Sangwongngam, and K. Sripimanwat, "Field-programmable gate array implementation of low-density parity-check codes decoder and hardware testbed," in *TENCON Spring Conference, 2013 IEEE*, 2013, pp. 104-107.

[24] C. Yu, H.-S. Chuang, B.-S. Lin, P.-H. Cheng, and S.-J. Chen, "Improvement on a block-serial fully-overlapped QC-LDPC decoder for IEEE 802.11 n," in *2014 IEEE International Conference on Consumer Electronics (ICCE)*, 2014, pp. 446-447.

[25] R. Li, J. Zhou, Y. Dou, S. Guo, D. Zou, and S. Wang, "A multi-standard efficient column-layered LDPC decoder for software defined radio on GPUs," in *2013 IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2013, pp. 724-728.

[26] J. C. Porcello, "Designing and implementing low density parity check (ldpc) decoders using fpgas," in *2014 IEEE Aerospace Conference*, 2014, pp. 1-7.

[27] M. Karkooti, P. Radosavljevic, and J. R. Cavallaro, "Configurable, high throughput, irregular LDPC decoder architecture: Tradeoff

analysis and implementation," in *IEEE 17th International Conference on Application-specific Systems, Architectures and Processors (ASAP'06)*, 2006, pp. 360-367.