A LOW-COMPLEXITY IMPLEMENTATION OF QC-LDPC ENCODER IN RECONFIGURABLE LOGIC

Georgios Tzimpragos[*†], *Christoforos Kachris*[†], *Dimitrios Soudris*[*] *and Ioannis Tomkos*[†]

[*]National Technical University of Athens, 15780 Zographou Campus, Athens, Greece
Email: {getzim, dsoudris}@microlab.ntua.gr
[†]Athens Information Technology, 19002 Peania, Athens, Greece
Email: {kachris, itom}@ait.edu.gr

ABSTRACT

Low Density Parity Check(LDPC) codes are a special class of error correction codes widely used in communication and disk storage systems, due to their Shannon limit approaching performance and their favorable structure. In this paper, a methodology for optimized hardware multiplication by constant matrices in GF(2) is introduced and then applied to the Quasi-Cyclic LDPC encoding algorithm. Taking advantage of the fact that the parity check matrix rarely changes, the signals in many cases are hard-wired into the LUTs and thus the cyclic-shifters and block-memories conventionally used are eliminated. Therefore, the proposed framework leads to less complex, mapped to reconfigurable logic designs, whereas it combines the performance of hard-wired solutions (high throughput, low latency) and the flexibility of the software and its hardware counterparts. These advantages in terms of hardware savings and throughput prove that the proposed encoder scheme is suitable for high-speed applications, such as long-haul optical transmission, where speed and resources utilization are a major issue.

I. INTRODUCTION

In computer science, telecommunications and information theory, *forward error correction (FEC)* is a technique with great practical significance in data transmission over unreliable, noisy communication channels and finds application in the mobile, satellite and optical communication, and disk storage systems. Generally speaking, the main idea of *error control coding (ECC)* is to provide the transmitter with ways to encode data signals in a redundant way following certain relations, in order to enable automatic error detection and correction in the received signals. The publication of Claude Elwood Shannon's landmark paper in 1948 [1], describing the maximum rate at which data can be transmitted over a noisy communications channel of a specified bandwidth, was the starting point of the study of error control coding and since then much research has been devoted to the optimization of encoding and decoding methods for error control in noisy environments.

Low Density Parity Check (LDPC) codes are linear error correcting codes, defined by a sparse parity-check matrix. These codes were originally developed by Gallager in the 1960s [2] and have demonstrated bit error rate (BER) performance close to the Shannon limit. However, they have been unnoticed for a long time, because their computational complexity was very high for the existent hardware technology. The reappearence of Turbo codes in 1992 though, led to their rediscovery a few years later by MacKay and Neal [3]. Compared with Turbo codes, LDPCs in many cases demonstrate better characteristics, such as parallelism in decoding and simple computation operations, while having high performance. Moreover, a special subclass of LDPC codes, called *Quasi-Cyclic LDPC (QC-LDPC) codes*, come along with an even more efficient implementation, while maintaining great performance. Quasi-cyclic codes are codes in which a cyclic shift of one codeword results in an other codeword and due to their advantageous structure they require less memory as compared with the conventional LDPC codes, whereas their encoding is proved to be linear with code length.

In this paper, we focus on the development of an optimized hardware archtitecture for multiplication by constant matrices in GF(2) and the presentation of a novel design methodology for QC-LDPC encoders. To prove the effectiveness of the proposed technique we apply it to the encoding algorithm of *block-type LDPC (B-LDPC)* codes. The B-LDPC codes are a distinguished class of QC-LDPC codes, characterized by a suitable for efficient hardware implementation encoding algorithm, good noise threshold, and low error floor [4].

Overall, the main contributions of this paper are the following:

The research leading to these results is partially supported by the ASTRON project (Adaptive Software-defined Terabit Transceiver for flexible Optical Networks) with funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under grant agreement n. 318714.

- presentation of an optimized hardware archtitecture for multiplication by constant matrices in GF(2),
- introduction of a design framework for QC-LDPC encoders, based on a look-up table (LUT) method and advanced mapping to the FPGA resources,
- a low-complexity, high-throughput implementation of QC-LDPC encoder in reconfigurable logic, based on the proposed framework.

The remainder of the paper is organized as follow. In Section 2, after a brief introduction to QC-LDPC codes, their use as error correction codes in communication standards is presented. Moreover, a suitable encoding algorithm is reviewed. In Section 3 the proposed design methodology is at first described and then applied to B-LDPC codes. The differences between our solution and existing are also highlighted. In Section 4 experimental results and comparisons with other solutions, in terms of throughput and area, indicate the gain achieved by the proposed scheme. Finally, we give concluding remarks in Section 5.

II. BACKGROUND

II-A. QC-LDPCs in Communication Standards

Quasi-cyclic low-density parity-check (QC-LDPC) codes have received much attention as a family of forward error correction codes due to their favorable structure and excellent error correction performance. Generally, a binary QC-LDPC code is specified by a parity-check matrix, which consists of square sub-matrices of the same size over the (Galois field) GF(2), which are the zero matrix or circulant permutation matrices. Equation 1 provides a base of a QC-LDPC code's parity-check matrix in the form described above

$$Hb = \begin{bmatrix} P^{a_{11}} & \cdots & P^{a_{1N_b}} \\ \vdots & \ddots & \vdots \\ P^{a_{M_b1}} & \cdots & P^{a_{M_bN_b}} \end{bmatrix} \quad (1)$$

, where P is a *z*x*z* permutation matrix given by

$$P^1 = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 1 & 0 & \cdots & 0 \end{bmatrix} \quad (2)$$

Note that P^i is the circulant permutation matrix, which shifts the identity matrix I *i* times to the right ($0 \leq i \leq z$).

Especially, some communication standards, such as IEEE 802.11 and IEEE 802.16, have adopted the QC-LDPCs as error correction codes for their channel coding scheme [5], [6] and support various code rates and block lengths. It should be noticed that the QC-LDPC codes employed by the mentioned IEEE standards own the special property of the parity check matrices of B-LDPCs and thus their encoding can be done in an efficient way, without reference to the code generation matrix , as described in [4].

II-B. Encoding Algorithm

Assuming that the parity check matrix H has full rank, it is divided into the following form according to the Richardson - Urbanke encoding method [7].

$$\mathsf{H} = \begin{pmatrix} \mathsf{A} & \mathsf{B} & \mathsf{T} \\ \mathsf{C} & \mathsf{D} & \mathsf{E} \end{pmatrix}$$

, where A is *(Mb - 1)*z* $\times$ *Kb*z*, B is *(Mb - 1)*z* $\times$ z, T is *(Mb - 1)*z* $\times$ *(Mb - 1)*z*, C is z $\times$ *Kb*z*, D is z $\times$ z, E is z $\times$ *(Mb - 1)*z*, and *Nb = Mb + Kb*.

Let *c* be a a codeword of the code specified by H. Then the following equation (syndrome check) has to be satisfied.

$$H * c^T = 0 \quad (3)$$

, where *s* denotes the *Kb*z* information bits and *p1* and *p2* symbolize the z and *(Mb - 1)*z* parity bits (c = [s p1 p2]), respectively. From the above equation, *p1* can be obtained by:

$$p1^T = \phi^{-1} * (E * T^{-1} * A * s^T + C * s^T) \quad (4)$$

, but the inverse of the matrix $\phi = (E * T^{-1} * B + D)$ is not sparse, and thus the computational complexity is high. In general, while the LDPC decoder can operate in linear time, it may be hard to perform low-complexity encoding of these codes [8].

However, in their paper *"Quasi-Cyclic LDPC Codes for Fast Encoding"* [4], *S. Myung et al.* proposed an encoding algorithm for these codes with linearly scaled complexity, based on the idea to choose the matrix ϕ as the identity matrix or a circulant permutation matrix in general (B-LCPC codes). In that way, the encoding procedure of B-LDPCs is simplified and can be summarized in the following steps.

- Step1: Compute $A * s^T$ and $C * s^T$
- Step2: Compute $E * T^{-1} * A * s^T$
- Step3: Compute $p1^T = E * T^{-1} * A * s^T + C * s^T$
- Step4: Compute $p2^T$ by $T * p2^T = A * s^T + B * p1^T$

III. HARDWARE IMPLEMENTATION

In this section we present a novel hard-wire oriented methodology for implementing efficient QC-LDPC encoder architectures with high throughput and low resources consumption, based on an optimized hardware archtitecture for multiplication by constant matrices in GF(2). The overall design framework is based on a look-up table (LUT) method and advanced mapping to the FPGA resources.

Since the parity check matrix of the QC-LDPC codes is composed of circulant shifted identity and zero matrices, the key problem of designing the encoder is the efficient multiplication of the input data by a number of circular shift unit matrices. In contrast with other proposed designs, in the presented scheme there is no need for cyclic shifters

in order to carry out the required arithmetic operations. Noting that in many cases the code rate does not change regularly, the parity check matrix Hb is defined as constant and thus we proceed with an implementation approaching the efficiency of hard-wired solutions. To maintain the flexibility of its counterparts and support different code rates and block lengths use of the reconfigurable nature of FPGAs should be made.

To make things clear we should provide a simple example.

Assuming we want to execute the operation:

$$q = L * x^T \tag{5}$$

, where L is a z x $3z$ matrix consisting of z x z submatrices, x is a $3z$ x 1 matrix and q is a z x 1 matrix.

$$\begin{bmatrix} q_{z-1} \\ q_{z-2} \\ \vdots \\ q_1 \\ q_0 \end{bmatrix} = \begin{bmatrix} P^{a_2} & P^{a_1} & P^{a_0} \end{bmatrix} * \begin{bmatrix} x_{3z-1} \\ x_{3z-2} \\ \vdots \\ x_1 \\ x_0 \end{bmatrix} \tag{6}$$

$$\begin{bmatrix} q_{z-1} \\ q_{z-2} \\ \vdots \\ q_0 \end{bmatrix} = P^{a_2} * \begin{bmatrix} x_{3z-1} \\ x_{3z-2} \\ \vdots \\ x_{2z} \end{bmatrix} \oplus P^{a_1} * \begin{bmatrix} x_{2z-1} \\ x_{2z-2} \\ \vdots \\ x_z \end{bmatrix} \oplus P^{a_0} * \begin{bmatrix} x_{z-1} \\ x_{z-2} \\ \vdots \\ x_0 \end{bmatrix} \tag{7}$$

, where $\oplus$ here denotes the bitwise modulo-2 addition (XOR).

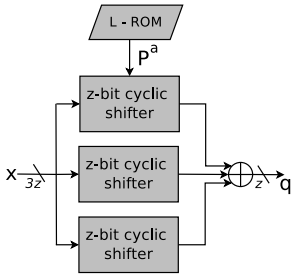

Fig. 1. Straightforward implementation of Eq.7. Colored components can be removed with the proposed method.

According to conventional approaches, this operation is implemented with the use of z-bit cyclic shifters(for the matrix multiplication) as shown in Figure 1. However, in the proposed solution as the matrix L is regarded constant, we skip the shift operation and connect the signals to the right LUTs (used as XOR operators because all calculations are in GF(2)) directly. Based on the proposed technique there is also no need for block-memories (ROM) to get used and further the number of required components is reduced. Therefore, the derived architecture exhibits lower hardware complexity (i.e. a smaller number of gates). Additionally the maximum number of the required resources is totally predictable.

To prove the effectiveness of the proposed technique we apply it to the encoding algorithm described above. Figure 2 depicts an architecture overview of the QC-LDPC encoder defined by the parity-check matrix given by the IEEE 802.11n standard (Rate = 5/6 and subblock size z = 81).

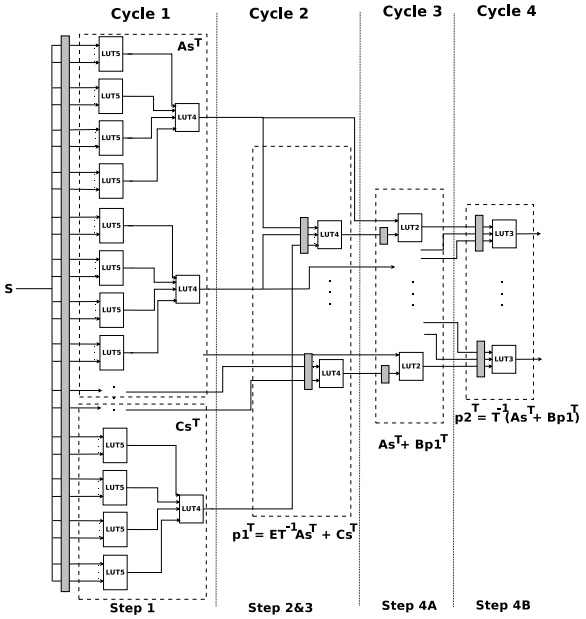

Fig. 2. Architecture overview for the LDPC code with R = 5/6 and z = 81 bits, given by the IEEE 802.11n standard

The hardware verification of the LDPC encoding circuit can be realized by the outcome of Eq. 3.

IV. IMPLEMENTATION RESULTS

This section is devoted to the presentation of IEEE 802.11 LDPC encoder implementation results, defined by the parity- check matrix given by the standard for codeword length n = 1944 bits, rate 5/6 and block size z = 81 bits. The target reconfigurable medium is a Xilinx Virtex5 (xc5vlx20t).

In more detail, Table I provides a comparison between existing and the proposed QC-LDPC encoder in terms of throughput, latency and resources utilization. All the included designs were applied to LDPC codes with the same structure and rate. Compared with its counterparts, the proposed implementation has a higher maximum frequency, requires less clock cycles, and therefore it achieves significant higher throughput. As it is demonstrated by the results, skipping cyclic-shifters and proceeding with a hard-wire oriented implementation leads to important gains in term of encoding speed. Further performance enhancement could be accomplished by the use of pipeline techniques, which will come along with an increase in the number of occupied slices (more registers will be required) though.

As far as area is concerned, the presented solution is more advantageous than its counterparts. As can be seen in Table I, it requires less LUTs/LEs(Logic Elements) than

Table I. Throughput and latency comparison of QC-LDPC encoders

[a]	[9]	[10]	[11]	[12]	[13]	[14]	**Proposed Design**
Code Rate	5/6	5/6	5/6	5/6	5/6	5/6	5/6
Codeword length(bits)	1944	1944	1944	2304	2304	3072	1944
Frequency (MHz)	-	-	-	60	150.69	100	290
Clock cycles	24	73-83	24	-	51	-	4
Throughput (Gbps)	-	-	-	0.36	5.67	19.2	117.45
LUTs/LEs	-	-	-	11,430	12,306	-	1,782
Block-Memories	✓	✓	✓	✓	-	✓	✗
FFs	1701	162	1053	-	-		2187
XOR gates	1620	243	1053	-	-	688	-
z-bit barrel shifter	20	1	1	-	-	-	0
2nd-stage cyclic shifter	0	0	11	-	-	-	0
Target medium	-	-	-	STRATIX EPIS80F1506c6	STRATIX EP1S25F672C6	Virtex-II	Virtex5 (xc5vlx20t)

[a]- denotes that the associated implementation result of this design is not available.

[12], [13], whereas although it utilizes more FFs than [9], [10], [11], the fact that no barrel shifters are employed, makes it area efficient. For comparison reasons, we should mention that in a state-of-the-art Xilinx Virtex6 device a 32-bit barrel shifter uses 96 LUTs. Hence, it is obvious that avoiding the z-bit barrel shifters(in our case $z = 81$ bits) brings on important resources savings. Finally, we should point out that in the presented scheme, in contrast with the others, no block-memories were used.

V. CONCLUSION

In this work, a novel design methodology for QC-LDPC encoders has been presented. To prove the low-complexity and high-throughput of the implementations within the proposed framework, this technique has been applied to the encoding algorithm of Block-type LDPC codes. The selected algorithm is based on the special structure of the parity check matrices of these codes and can be applied to IEEE 802.11, IEEE 802.16 and other LDPC codes with similar properties. The proposed hard-wire oriented design methodology comes along with advanced mapping to the FPGA resources and leads to significant gains in the encoding speed, while keeping the resources utilization low. Moreover, the required flexibility is not sacrificed by taking advantage of the reconfigurable nature of FPGAs. Due to these advantages, the proposed encoding scheme is proven to be suitable even for high-speed applications, such as long-haul optical transmission.

VI. REFERENCES

[1] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, July, October 1948.

[2] R. G. Gallager, *Low-Density Parity-Check Codes*. MIT Press, 1963.

[3] D. J. MacKay and R. M. Neal, "Near Shannon Limit Performance of Low Density Parity Check Codes," *Electronics Letters*, vol. 32, pp. 1645–1646, 1996.

[4] S. Myung, K. Yang, and J. Kim, "Quasi-cyclic ldpc codes for fast encoding," *Information Theory, IEEE Transactions on*, vol. 51, no. 8, pp. 2894–2901, 2005.

[5] "IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Ame," *IEEE 802.11*, pp. –.

[6] "IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Broadband Wireless Access Systems," *IEEE Std 802.16-2009 (Revision of IEEE Std 802.16-2004)*, pp. 1–2080, 2009.

[7] T. Richardson and R. Urbanke, "Efficient encoding of low-density parity-check codes," *Information Theory, IEEE Transactions on*, vol. 47, no. 2, pp. 638–656, 2001.

[8] A. Mahdi, N. Kanistras, and V. Paliouras, "An encoding scheme and encoder architecture for rate-compatible QC-LDPC codes," in *Signal Processing Systems (SiPS), 2011 IEEE Workshop on*, 2011, pp. 328–333.

[9] Z. Cai, J. Hao, P. Tan, S. Sun, and P. S. Chin, "Efficient encoding of IEEE 802.11n LDPC codes," *Electronics Letters*, vol. 42, no. 25, pp. 1471–1472, 2006.

[10] J. Perez and V. Fernandez, "Low-cost encoding of IEEE 802.11n," *Electronics Letters*, vol. 44, no. 4, pp. 307–308, 2008.

[11] Y. Jung, Y. Jung, and J. Kim, "Memory-efficient and high-speed LDPC encoder," *Electronics Letters*, vol. 46, no. 14, pp. 1035–1036, 2010.

[12] H. Yasotharan and A. Carusone, "A flexible hardware encoder for systematic low-density parity-check codes," in *Circuits and Systems, 2009. MWSCAS '09. 52nd IEEE International Midwest Symposium on*, 2009, pp. 54–57.

[13] S. Kopparthi and D. Gruenbacher, "Implementation of a Flexible Encoder for Structured Low-Density Parity-Check Codes," in *Communications, Computers and Signal Processing, 2007. PacRim 2007. IEEE Pacific Rim Conference on*, 2007, pp. 438–441.

[14] Z. He, S. Roy, and P. Fortier, "Encoder architecture with throughput over 10 Gbit/sec for quasi-cyclic LDPC codes," in *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, 2006, pp. 4 pp.–.