

A New Design for Reducing Logic Utilizations in FPGA-Based Stochastic LDPC Decoders

Ghania Zerari and Abderrezak Guessoum

"LATSI" Laboratory, Department of Electronics, University of Blida, Blida, Algeria

Email: ghania.zerari@gmail.com

Rachid Beguenane

Department of Electrical and Computer Engineering, Royal Military College of Canada, Kingston, Canada

Email: rachid.beguenane@rmc.ca

Abstract—This paper presents a new fully parallel stochastic Low-Density Parity-Check (LDPC) decoding approach, to implement field programmable gate array (FPGA) based LDPC-decoders. The improved configuration is synthesised to optimize the FPGA logic utilisation and to decrease the decoding latency. To attain the target complexity reduction, the routing of the proposed sub variable node (VN) internal memory is designed to exploit only one slice distributed RAM. Furthermore, an efficient VN initialization, using the channel input probability, is performed to improve the decoder convergence, without requiring additional resources and without incorporating output saturated-counters. The Xilinx FPGA implementation, of (48, 24), (200, 100) and (1024, 512) regular codes, shows that the proposed decoding approach reach high performance along with reduction of logic utilisation.

Index Terms—Low-Density Parity-Check (LDPC) decoder, stochastic decoding, field programmable gate array (FPGA)

I. INTRODUCTION

The necessity for growing the throughput of recent communication systems capacity, requires high performance error correcting code. In 1962, Gallager introduced the primarily version of Low-density parity-check (LDPC) codes [1], offering an outstanding procedure for correcting transmission errors, added by the channel noises. The performance capacity of the LDPC codes [2] is exploited by a variety of digital communication norms. 10GBASE-T (IEEE 802.3an), DVB-S2, WiFi (IEEE 802.11) and WiMAX (IEEE 802.16e) standards attest this great performance.

It has been clearly revealed that the higher throughput is achieved by the fully parallel decoding solutions, nevertheless they enlarged the FPGA logic utilisations and the hardware complexity. A several of LDPC decoding implementations have been explored to achieve high throughput results [3]-[5]. Numerous reduced-complexity and stochastic LDPC decoding architectures are elaborated [6]-[8]. The recent stochastic decoding approach affirmed their adaptability for parallel decoding method [9]-[18]. Additionally and for supplementary

silicon area reduction, different LDPC decoding architectures and strategies, using the stochastic methods, are proposed.

Nevertheless, an area-efficient architecture for ASIC-Based stochastic LDPC decoder can not systematically generates an efficient FPGA logic utilisation. It is straightforward that, the six bits counter ASIC implementation needs less silicon area compared to 64 bits memory. However an opposite result is achieved by the FPGA implementation. A Xilinx FPGA 64 bits memory implementation can be routed using only one LUT, in contrast with the six bits counter implementation. This paper presents a new and improved field programmable gate array (FPGA) based stochastic Low-Density Parity-Check (LDPC) decoding architecture, to implement fully parallel LDPC-decoders. The proposed technique is designed to reduce the FPGA logic utilization and to improve the convergence, even for short codes. A preliminary work was published in [19]. To validate the gain of the improved approach, an FPGA implementation using Xilinx Virtex-6 VLX240T is performed. The paper is organized as follows. In Section II, we give a summary of the LDPC stochastic decoding process and architecture. In section III, the configuration of the new developed stochastic LDPC decoding is presented. Results of the Xilinx FPGA implementation are presented in section IV, and the conclusion is given in the last section.

II. LDPC STOCHASTIC DECODING STRUCTURE

The regular LDPC codes are investigated in our target decoder. LDPC decoder can be characterized by a factor graph which uses N VNs and $(N-K)$ CNs. The K information bits is encoded, with (N, K) LDPC code, in to N encoded bits, where $N > K$. The design of an LDPC decoder is based on the $M \times N$ parity check matrix H . N defines the number of variable nodes (VNs) while M defines the number of CNs. A dv degree VN has $(dv+1)$ ports, one of which gets the channel probability and the other dv are connected to different CNs, by bidirectional ports. In like manner, the dc degree CN has dc bidirectional ports, which are connected to different VNs, and one parity-check output port.

Manuscript received July 25, 2017; revised October 9, 2017.

The non stochastic LDPC fully parallel decoder uses fixed-point operands to symbolize the probabilities, exchanged between the VNs and the CNs of the factor graph. Stochastic LDPC decoders perform the computing by a bit-serial iterative process. In this structural design, the received probabilities P_{ch} from the channel are converted to Bernoulli sequences as random bits sequences. Several encoded stochastic sequences can be produced for the same probability. In a $\{a_i\}$ Bernoulli sequence of m bits, in which $a_i \in \{0, 1\}$, the estimated probability value is computed as

$$P_{ch}(m) = \frac{\sum_{i=1}^m a_i}{m} \quad (1)$$

$Cout$, Pc_{out_i} and Pc_{in_i} represent the parity-check output, the CN outputs and the CN probabilities inputs respectively, where $Pc_{in_i} = Pr(cin_i=1)$ is the probability of each CN inputs, in which $i \in \{1, 2, \dots, dc\}$ and dc is CN degree. The output probability Pc_{out_i} can be computed as

$$\left. \begin{aligned} Pc_{out_1} &= Pc_{in_2} \oplus \dots \oplus Pc_{in_{dc}} \\ Pc_{out_i} &= Pc_{in_1} \oplus \dots \oplus Pc_{in_{i-1}} \oplus Pc_{in_{i+1}} \oplus \dots \oplus Pc_{in_{dc}} \\ Pc_{out_{dc}} &= Pc_{in_1} \oplus \dots \oplus Pc_{in_{dc-1}} \end{aligned} \right\} \quad (2)$$

Fig. 1 illustrates the configuration of a degree dc CN used in the conventional stochastic decoder. The parity-check output $Cout$, of dc degree CN, can be computed according to (3).

$$Cout = Pc_{in_1} \oplus \dots \oplus Pc_{in_{dc}} \quad (3)$$

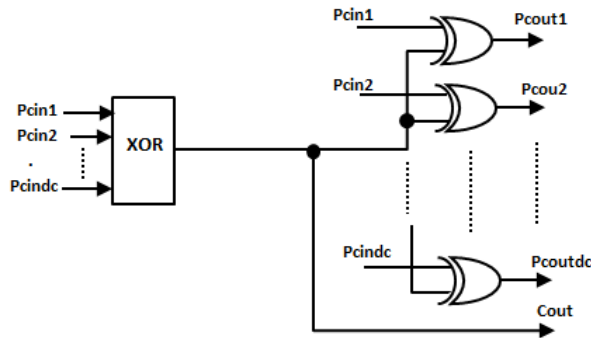


Figure 1. The structure of the-check node

As an example, we consider a stochastic variable node with two input bits ($dv=2$). $Pvin_1$ and $Pvin_2$ denote the probability of the two input bits. The variable node output probability $Pvout$ can be computed as

$$Pvout = \frac{Pvin_1 \cdot Pvin_2}{Pvin_1 \cdot Pvin_2 + (1 - Pvin_1) \cdot (1 - Pvin_2)} \quad (4)$$

In case that the VN inputs are same, one of the input bits will be transmitted to the output. This state is named the agreement state. When the inputs are not identical, the variable node requires an advanced method to generate the output bit. This state is named disagreement state or the hold state. One of the advanced stochastic method bit generation (ASMBG) can be used.

$$Pvout(N) = \begin{cases} \text{Bit using (4)} & \text{if } Pa = Pb \\ \text{Bit using ASMBG} & \text{otherwise} \end{cases} \quad (5)$$

Fig. 2 shows the recent stochastic two inputs variable node main architecture.

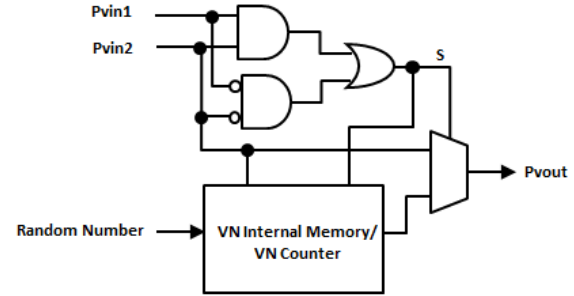


Figure 2. The structure of recent stochastic degree-2 variable node

The stochastic LDPC decoding algorithm can be summarised as follows:

Algorithm 1 Stochastic LDPC decoding

Initialization

1. Load the LLRs corresponding probabilities P_{ch} for each variable node (one DC) and transform P_{ch} to Bernoulli sequence a_i (each DC).

2. Initialize the variable nodes internal memories (16 to 32 DCs for 32 bits memory [10]) or the internal saturated counter (one DC [16]).

Iterations

3. Variable to check node : At each decoding cycle, the variable node computes there inputs bits using (5) and sends there outputs bits to the corresponding check nodes.

4. Check to variable node : At each decoding cycle, the check node computes there inputs bits using (2) and sends there outputs results to the corresponding variable nodes. Simultaneously, the check nodes send there outputs states using (3) to the syndrome checker.

5. If $xH^T = 0$ or the maximum of DCs is reached, terminate the decoding process. Otherwise, go to step 3.

III. PROPOSED FPGA-BASED LDPC STOCHASTIC CONFIGURATION

It has been proven that initializing the first VN output bit, based on the received probability channel, enhance the stochastic decoder convergence [15]. Furthermore, it has been shown that the LDPC stochastic decoding process, which there VNs use the latest output bits as code bits, offer comparable BER performance to the version with output saturating up/down [17]-[18]. The proposed VN exploits the two referenced characteristics, in addition to adopt an internal memory-based configuration, analogous to DS and EM versions.

The output VN probability will be calculated as follows.

$$Pvout(t) = \begin{cases} Pst & \text{if } t = 1 \\ Pa \text{ or } Pb & \text{if } t \neq 1 \text{ \& } Pa = Pb \\ \text{Bit from FPGA LUT RAM} & \text{otherwise} \end{cases} \quad (6)$$

$$\text{where } Pst = \begin{cases} 1 & \text{if } Pch \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

The proposed VN initialisation is completed with ($t = 1$). In the disagreement state ($t \neq 1$), the novel architecture generate a new bit by reading randomly the VN internal memory (IM). The length of the IM can be enlarged up to LUT FPGA RAM length. Based on (6), the hardware implementation of the improved decoding structure does not call an extra FPGA logic utilisation. Furthermore, the proposed technique computes the received probability without any additional decoding cycle.

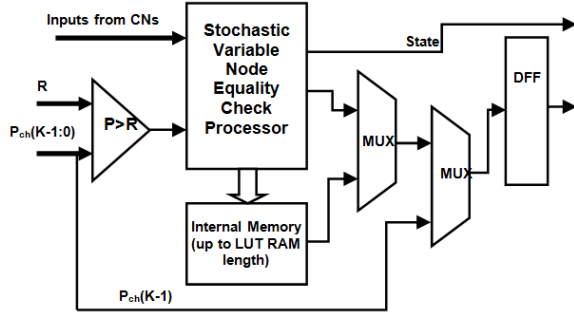


Figure 3. The structure of the proposed stochastic VN.

The $P_{ch}(k-1)$ bit is transmitted to the VN output DFF in the first cycle. Following the first cycle and until the last one, the multiplexer sends the variable node processor output bit to the VN output DFF. In such manner and comparably to the CSS process, the larger part of variable nodes outputs start with a correct bit and pass around the conventional stochastic initialization. Fig. 3 represents the main organization of the new variable node. The proposed CN have two types of inputs and two outputs.

The inputs are the $CNin_i$ signals and $State_i$ signals, in which $i \in \{1, 2, \dots, dc\}$. The inputs are linked to the VNs outputs signals. The first outputs signals are the $CNout_i$, which are sent to VN inputs. The second output is the parity-check output state $CNstate$. $CNstate$ outputs are connected to the syndrome checker unit. The parity-check output state $CNstate$, of dc degree CN, is calculated with the same method using by (3) and can be written as follows.

$$CNstate = \sum_{i=1}^{dc} \oplus CNin_i \quad (7)$$

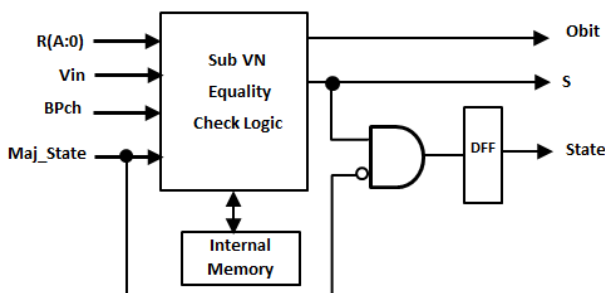


Figure 4. The structure of the proposed stochastic sub variable node

where $\sum \oplus$ is the bitwise XOR operation and dc is the parity-check node degree.

Each CN outputs signals $CNout_i$ uses the $CNin_i$ signals and $State_i$ signals, to generate the $CNout_i$ signals

according to the expression (8). The $CNstate$ result given by (7) can be exploited.

$$CNout_i = \left((CNstate) \wedge \left(\bigvee_{i=1}^{dc} \overline{state_i} \right) \right) \oplus CNin_i \quad (8)$$

where \bigvee is the bitwise NOR operation and \wedge is the bitwise AND operation.

The proposed stochastic LDPC decoding algorithm can be outlined as follows:

Algorithm 2 The proposed Stochastic LDPC decoding

Initialization

1. Load the LLRs corresponding probabilities P_{ch} simultaneously with initializing the variable node output (one DC) and transform P_{ch} to Bernoulli sequence a_i (each DC).

Iterations

2. Variable to check node : At each decoding cycle, the variable node computes there inputs bits using (6) and sends there outputs bits and states to the corresponding check nodes.

3. Check to variable node: At each decoding cycle, the check node computes there inputs bits using (8) and sends there outputs results to the corresponding variable nodes. Simultaneously, the check nodes send there outputs states using (7) to the syndrome checker.

4. If $xH^T = 0$ or the maximum of DCs is reached, terminate the decoding process. Otherwise, go to step 2.

IV. IMPLEMENTATION RESULTS

The main objective of the proposed architecture is to enhance the FPGA-based LDPC decoding performance, without extra FPGA logic utilisation. To validate the improvement of the new design, a (1024, 512), (200, 100) and (48, 24) LDPC Regular codes are implemented on Xilinx Virtex-6 VLX240T field programmable gate array (FPGA) device, with different techniques.

The implemented CNs use comparable configuration to CNs adopted by the DS and the CSS decoders. Fig. 5 gives the structure of degree dc CN.

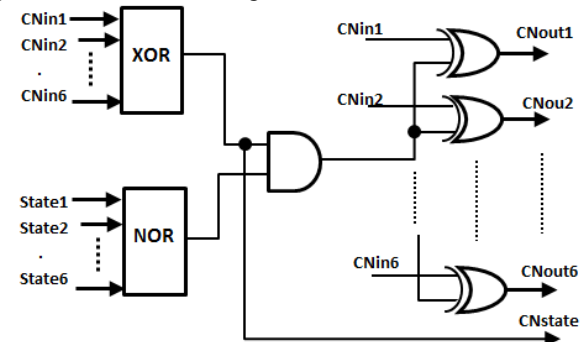


Figure 5. The structure of the proposed degree dc CN

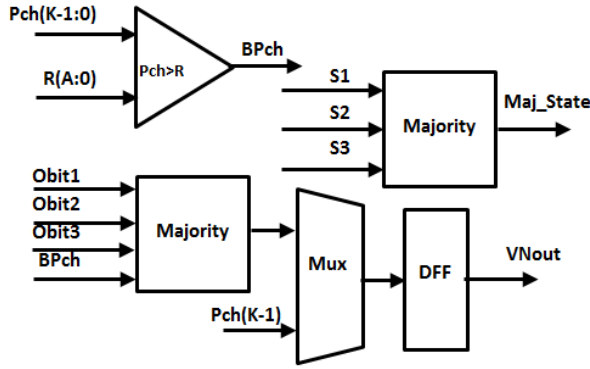


Figure 6. The block diagram of the connection to associate three variable sub nodes

Fig. 6 presents the block diagram of the connection to associate sub nodes of the degree-3 VN of the proposed (1024, 512) and (200, 100) and (48, 24) LDPC codes. The degree-3 Variable Node is composed by 3 degree-3 sub-node. The majority state and the random address signals are connected to all sub-nodes. One of the 3 degree-3 sub-node input is connected to probability signal by a comparator. The other two sub-node inputs are connected to the check node output. The three S signals are joint to produce the majority state signal. The FPGA implementation of VN Internal Memory is achieved by using the (LUT) Slice FPGA distributed RAMs.

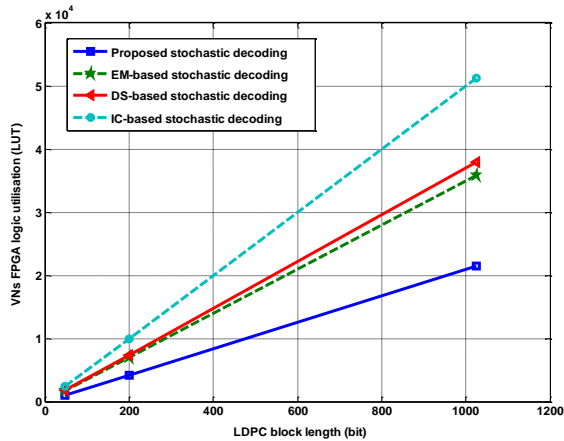


Figure 7. Result and the comparison of the FPGA logic utilisations implementation of the variable nodes

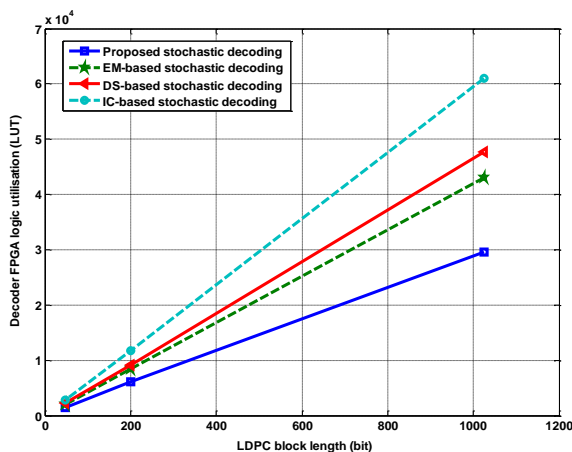


Figure 8. Result and the comparison of the FPGA logic utilisations implementation of the stochastic decoders

The Fig. 7 and Fig. 8 show the result and the comparison of the FPGA logic utilisations implementation of the variable nodes unit and the decoder respectively. The proposed decoder achieves an average reduction about of 50% compared to the one-step initialized counter and an average reduction about of 35% compared to DS and EM decoders. In addition to the logic utilisation reduction, the new architecture offers additional performance even for short codes. The EM decoder provides an FPGA implementation result close to the DS variety.

The implementation of 2×1 bits up to 64×1 bits exploits only one LUT in Virtex-6 Xilinx FPGA. Therefore, the proposed LDPC decoder version can be implemented using up to 64-bit VN internal memory without requiring additional FPGA resources. As we can see, the FPGA implementation of the one-step initialized counter based decoder need additional resources, compared to the EM and the DS versions. This drawback is caused by the employment of an initializing counters instead of the VN internal memories used in DS. The additional reduction of FPGA logic utilisation seen for the new proposed decoder is principally obtained as a result of the unemployment of VN output saturated counter.

V. CONCLUSION

In this work, we investigated the logic utilisation of FPGA-based implemented LDPC stochastic decoders. Hence, an improved fully parallel stochastic LDPC decoding configuration was introduced, which can outperform all state of the arte editions. The improvement was validated by making an efficient stochastic variable node. A reduction of decoding latency and complexity, in addition of BER amelioration, are achieved even for short codes. A Xilinx Virtex-6 VLX 240T FPGA implementation results validated the advancement of the new method. An average reduction of 35% and 85% of logic utilisation and average decoding cycles respectively, compared to DS method are reached.

REFERENCES

- [1] R. G. Gallager, "Low density parity check codes," *IRE Trans. Inf. Theory*, vol. 8, pp. 21–28, Jan. 1962.
- [2] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electron. Lett.*, vol. 32, no. 18, pp. 1645–1646, 1996.
- [3] The IEEE 802.16 Working Group. [Online]. Available: <http://www.ieee802.org/16/>
- [4] The IEEE 802.3an 10GBASE-T Task Force. [Online]. Available: <http://www.ieee802.org/3/an>
- [5] IEEE Working Group for 40 Gb/s and 100 Gb/s Operation Std. [Online]. Available: <http://www.ieee802.org/3/>
- [6] The Digital Video Broadcasting Standard. [Online]. Available: <http://www.dvb.org>
- [7] The IEEE 802.11n Working Group Std. [Online]. Available: <http://www.ieee802.org/11/>
- [8] V. Gaudet and A. Rapley, "Iterative decoding using stochastic computation," *Electron. Lett.*, vol. 39, no. 3, pp. 299–301, Feb. 2003.
- [9] S. S. Tehrani, W. Gross, and S. Mannor, "Stochastic decoding of LDPC codes," *IEEE Commun. Lett.*, vol. 10, no. 10, pp. 716–718, Oct. 2006.

- [10] S. S. Tehrani, S. Mannor, and W. J. Gross, "Fully parallel stochastic LDPC decoders," *IEEE Trans. Signal Process.*, vol. 56, no. 11, pp. 5692–5703, Nov. 2008.
- [11] S. Tehrani, A. Naderi, G. A. Kamendje, S. Mannor, and W. Gross, "Tracking forecast memories in stochastic decoders," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2009, pp. 561–564.
- [12] S. S. Tehrani, A. Naderi, G. A. Kamendje, S. Mannor, and W. Gross, "Tracking forecast memories for stochastic decoding," *J. Signal Process. Syst.*, pp. 1–11, 2010.
- [13] S. S. Tehrani, A. Naderi, G. A. Kamendje, S. Hemati, S. Mannor, and W. Gross, "Majority-based tracking forecast memories for stochastic LDPC decoding," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4883–4896, Sep. 2010.
- [14] A. Naderi, S. Mannor, M. Sawan, and W. J. Gross, "Delayed stochastic decoding of LDPC codes," *IEEE Trans. Signal Process.*, vol. 59, no. 11, pp. 5617–5626, Nov. 2011.
- [15] M. Maamoun, R. Bradai, A. Naderi, R. Beguenane, and M. Sawan, "Controlled start-up stochastic decoding of LDPC codes," in *Proc. IEEE Int. NEWCAS Conference*, Paris, France, June 2013.
- [16] D. Wu, Y. Chen, Q. Zhang, Y. Ueng, and X. Zeng, "Strategies for reducing decoding cycles in stochastic LDPC decoders," *IEEE Trans. Circuits and Systems II*, vol. 63, no. 91, pp. 873–877, Sept. 2016.
- [17] K. L. Huang, V. Gaudet, and M. Salehi, "Output Decisions for Stochastic LDPC Decoders," in *Proc. 48th Annual Conference on Information Sciences and Systems*, Princeton, USA, March 2014.
- [18] K. L. Huang, "Efficient Algorithms for Stochastic Decoding of LDPC Codes," Doctor of Philosophy, Northeastern University, Boston, USA, May. 2016.
- [19] G. Zerari and A. Guessoum, "Improved controlled start-up stochastic LDPC decoder for efficient FPGA-based implementation," in *Proc. 4th International Conference on Electrical Engineering*, Boumerdes, Algeria, Dec. 2015.