Analysis and Implementation of On-the-Fly Stopping Criteria for Layered QC LDPC Decoders

Andrei Hera[1], Oana Boncalo[1], Constantina-Elena Gavriliu[1], Alexandru Amaricai[1], Valentin Savin[2],David Declercq[3], Fakhreddine Ghaffari[3]
[1] University Politehnica Timisoara, Timisoara, Romania
[2] CEA-LETI, MINATEC Campus, Grenoble, France
[3] ETIS-ENSEA, Cergy-Pontoise, France

***Abstract*—This paper presents an analysis of existing stopping criteria for layered architecture used for quasi-cyclic (QC) LDPC decoders. Furthermore, it proposes a novel imprecise method for early termination in layered decoders. The analysis is performed under the same framework in order to provide a fair and accurate comparison between existing methods, and our new solution. The developed hardware modules have been designed independently from the decoder architecture, with the only constraint that the decoder scheduling is layered. Synthesis estimates for Xilinx Virtex-7 devices and the decoding performance analysis indicate that the new stopping criterion presents the best cost/performance trade-off for most of the considered LDPC codes.**

***Keywords*—LDPC Decoder, Error Correction Codes, Layered Architecture, Early Termination**

I. Introduction

Modern communication systems have an increased need for higher and higher data rates combined with the requirement of reducing power consumption. Low-Density Parity Check (LDPC)[1] codes form a class of error correction codes that have a decoding performance close to the Shannon limit, and have been adopted in many standards such as DVB-S2, WiMAX, WLAN [2][3]. A structured subclass of these codes, named Quasi-Cyclic LDPC (QC-LDPC), relies on special code construction methods, and makes them suitable for hardware (HW) implementations [5]. Practical implementations of LDPC decoders deal with different *scheduling strategies*, according to the order in which the information is being processed during a decoding iteration. There are two major solutions proposed in the literature: *flooding scheduling* [6] and *layered scheduling* [7]. The latter allows a flexible trade-off between throughput and hardware resources. Furthermore, it presents a faster convergence compared to the flooding schedule.

Another important characteristic of any decoding architecture is represented by stopping rule used in the decoding process. Two approaches exist: *(i)* execute a fixed number of iterations *(ii)* stop the decoding if the syndrome is checked (i.e. all parity checks are satisfied for all equations), or if the maximum allowed number of iterations have been reached (in case no syndrome check has been reached). Furthermore, simulation analysis for different codes have shown that the average iteration number is dependent on signal to noise ratio (SNR), message quantization and the LDPC code (defined by the parity check matrix). Since the average number of iteration to decode a noisy frame is typically much smaller than the maximum allowed iterations, especially for long codewords, stopping the decoder earlier may lead to significant reduction in energy consumption [8]-[12]. For layered architectures, implementing the stopping criterion is not a trivial task, as a straightforward early-termination implementation may have increased HW complexity and can reduce the decoder throughput. Thus, the energy saving obtained by stopping the decoder may not compensate for the energy increase due to the early termination circuit. Therefore, it is important that the early termination circuit has minimum complexity in order to limit area and power overhead of the extra logic. This work focuses on reviewing existing techniques for layered architectures that allow early decoding termination, with minimum area overhead. Furthermore, we propose a new low cost early termination technique for stopping the decoder.

This article is organized as follows: Section II presents the layered scheduling for QC-LDPC decoding; the early termination strategies for layered architecture are described in Section III; Section IV presents the proposed on-the-fly new stopping criterion; Section V is dedicated to the analysis of these strategies.

II. Layered Scheduling for QC-LDPC Decoder

LDPC codes are linear codes defined by sparse parity check matrices H and can be represented by bipartite Tanner graphs. These types of graphs connect two kinds of computational units: variable node units, corresponding to the codeword bits, and check nodes units, corresponding to the parity check equations. LDPC decoding is performed by message-passing (MP) algorithms, which rely on messages exchange along the edges of the Tanner graph [4].

Quasi-Cyclic (QC) LDPC codes are a class of LDPC codes which feature highly structured parity check matrices, defined by blocks of circulant matrices [5]. This feature has led to their wide use in many telecommunication standards. A QC-LDPC code is defined by a base matrix B, with integer entries $b_{i,j} \geq -1$. The parity check matrix is obtained by expanding the base matrix B by a factor m. Each entry of B is replaced by a square matrix of size $m \times m$, as follows: all-zero matrix is used

ISBN 978-83-63578-06-0

to replace -1 entries, while entries $b_{i,j} \geq 0$ are replaced by a circulant matrix, obtained by right-shifting the identity matrix by $b_{i,j}$ positions. A *horizontal layer* of *H* is defined as the set of *m* consecutive rows corresponding to one row of *B*.

Regarding decoding algorithms, Min-Sum (MS) and its variants (such as offset-MS, normalized MS, etc) are more frequently used, due to their low computational complexity. Layered MS starts with the initialization of the of the a-posteriori log-likelihood ratios (AP-LLR - γ_v) with the channel values. It performs the following steps: (1) update the variable node messages (α_v) (2) update the check node message (β_c) (3) update the AP-LLR. More details about layered MS decoders can be found in [7]. Table I presents the notations which will be used in paper.

III. Early Termination Strategies for Layered Decoders

A. Review of On-the-Fly Stoping Criterias

Multiple strategies have been employed for performing early termination. They are achieved by monitoring in some way the decoder's state. The efficiency of such a method lies in implementing a low cost strategy (small energy consumption overhead, low cost), that does not degrade either the decoder performance or its throughput. This work focuses on analyzing low cost early termination methods - *on the fly stopping criteria* - which can be easily integrated in the layered decoding flow.

The work in [8] proposes a low complexity solution, that does not affect decoding performance, called *SignStability*. It monitors AP-LLR messages' signs fluctuations. For iteration *I* and layer number *L* the newly computed γ_v signs are compared to the ones computed previouslly (i.e. sign of input AP-LLRs). If all signs match, then a counter is incremented by 1, otherwise the counter is reset. Decoding stops when the counter reaches a value equal to n_L.

The authors of [9] propose strategy based on verifying the parity check constraints. After updating the γ_v messages for layer *L*, the parity check constraints for the check nodes within layer *L* are verified sequentially. Decoding stops when all the on the fly parity check equations are satisfied, for all n_L layers. This method is called *on the fly syndrome check* (*OTF Syndrome*).

Another low complexity stopping criterion called *SDA* (*soft decision aided*) is presented in [10]. At a given moment it checks that all γ_v magnitudes are greater than a predefined (statically configured) threshold. If the condition holds for all layers, the messages are considered to be reliable and decoding is stopped. The decision assumes to have detected a codeword.

A slightly more complex criterion compared to the aforementioned is proposed in [11]. The following two conditions need to be satisfied for stopping the decoder: (i) each γ_v sign remains constant over two successive iterations; (ii) the absolute value of the smallest γ_v message is larger than a predefined threshold (we will refer to this method as *Sun2008*).

TABLE I. Notations

Symbol	Description
L	Current layer being processed
I	Current iteration
M	Number of rows in the parity check matrix
N	Number of columns in the parity check matrix
n_L	Number of layers in the parity check matrix
α_v	Variable node message computed by the v^{th} VNU
β_v	Check node message computed by the v^{th} CNU
γ_v	AP-LLR message computed for the v^{th} position VNU
d_c	Check node degree (number of inputs for a check node –row wise)
d_v	Variable degree (number of inputs for a variable node - column wise)

The work from [12], proposes several complex *safe* conditions (by safe the authors mean decision will not change with successive iterations) to halt decoding. For our evaluation, we have selected two such methods: SSD and SSiA. SSD requires: Check node Satisfaction and Sign Stability (condition SS) together with condition D. *SSiA* is an alternative set of conditions, derived from the first, which is more suitable for practical implementations. *SSiA* checks that the signs of the CN inputs match the outputs, and satisfy the CNs (*SSi*) combined with the requirement that the absolute value of the check node messages increase with respect to the previous iteration. Authors of [12], claim no decoding performance degradation, and a high degree of generality for their criterions (i.e. they aply for all types of codes).

$$SS: \begin{cases} \text{sign}\left(\gamma_v^{(m)}\right) = \text{sign}\left(\gamma_v^{(m-1)}\right) = \cdots = \text{sign}\left(\gamma_v^{(m-d+1)}\right), & v = 0, 1, \ldots \\ \prod_{v \in VNU_c} \text{sign}\left(\gamma_v^{(m)}\right) = 1, & c = 0, 1, \ldots, M-1 \end{cases} \quad (1)$$

$$\boldsymbol{D}: \Delta\beta_v^{(m-k)} \text{sign}\left(\gamma_v^{(m-k)}\right) \geq 0 \quad where \begin{cases} \Delta\beta_v^{(m)} = \beta_v^{(m)} - \beta_v^{(m-d)} \; (currentit - previt) \\ v = 0, 1, \ldots, N-1 \\ k = 0, 1, \ldots, \quad d-2 \end{cases} \quad (2)$$

The *sign* function outputs -1 for negative values, +1 for positive and 0 when the input parameter is equal to 0.

While decoding performance has been theoretically analyzed for some of these methods [12], others have been introduced as optimizations to a particular architecture [8]. Implementation results of LDPC codes with stopping criterion have been derived for different technologies especially when targeting low power designs (e.g. ASIC CMOS 65 nm [9], ASIC CMOS 90 nm [11]). One contribution of this work is to study all of these approaches under the same conditions, and to derive simulation and implementation characterizations of their performance.

B. Proposed On-the-Fly Imprecise PC Termination Criteria

We propose a low cost, low overhead early termination method with minimal or no performance degradation for layered LDPC decoders that is based on parity checks (PC). Low implementation cost is achived by only considering the signs of the newly updated γ_v messages. Parity check equations for each processed layer are computed. If they are satisfied, layer by layer, for n_L consecutive layers, a counter is incremented yielding *a partial syndrome check* as LLR messages fluctuate during layered decoding. Decoding ends, when three partial syndrome checks are obtained for a codeword.

```
Iter_Cnt = 0;
For it=0÷MAX_IT
    Intra_It_Cnt = 0;
    Foreach layer L
        process_layer();
        Intra_It_Cnt += layer_pc();
    If (Intra_It_Cnt == n_L)
        Iter_Cnt = Iter_Cnt + 1;
    If (Iter_Cnt == 3)
        Go to data_offloading();
data_offloading();
```

Layer_pc() returns 1 if all parity checks are satisfied for layer *L*. Here M_L is used to denote all the lines from the parity matrix H, coresponding to layer *L*.

$$\prod_{v \in VNU_c} \text{sign}\left(\gamma_v^{(m)}\right) = 1, \; c \in M_L \tag{3}$$

Because each AP-LLR value is updated several times during an iteration, the parity checks are computed for different values of the AP-LLRs . As shown in [13], these values fluctuate even after a codeword has been reached, making the pin-pointing of the exact moment of a codeword being decoded difficult. Partial syndrome computation (equivalent to the OTF method) is computed as the sum of all the parity checks coresponding to all the layers from the curent iteration. Depending on the LDPC code, chosing this as stopping criterion, yields up to two order of magnitude in performance loss (see Table II – OTF syndrome FER results).

Regarding the cost for the proposed solution, computing the parity checks for a layer, requires circulant size *dc* input XOR gates. The inputs are the updated γ_v message signs. Two counters are needed: a $\lceil log_2 n_L \rceil$ bit counter for partial syndrome check computation, that accounts for the satisfied PCs, and a 2 bit counter is needed for the satisfied iterations. For the decoder's control unit, a signal, when asserted, notifies that decoding can be stopped under the assumption that a codeword has been detected. All other processing is aborted, and decoding offloading can start.

IV. Analysis and Discussions

We have evaluated the stopping criteria within a common simulation framework under the same setup parameters: message quantization, maximum number of iterations, and a set of LDPC codes taken from the literature. We have considered both regular and irregular codes, with different rates. Ideally, decoding stops when a codeword is detected. This corresponds to the case, where at the end of each iteration AP-LLR sign bits are checked against all parity checks (i.e. conventional syndrome check). During decoding performance analysis, we have considered the conventional syndrome check as our *baseline* (best performance in FER, and minimum average iterations). For all other techniques, we have compared the results obtained from simulations against the baseline. We have run simulations for several SNR ranges (depending on each code and rate) and we have monitored two metrics: *FER* and *average number of iterations*.

The decoder used is layered Min-Sum (MS). We have employed a (4, 6) quantization scheme, meaning 4 bits for β messages, and 6 bits for γ, with the following codes and rates: WiMAX Rates 1/2, 5/6 [2], WPAN Rates 1/2, 3/4, 7/8 [3] and four non-standard regular matrices (rates 1/2 and 3/4) [14]. These regular matrices all have the same codeword length, being 1296. Circulants sizes are defined as m=54 for the low rate matrices and m=27 for the high rate matrices, with (d_v, d_c) degrees (3, 6) and (4, 16).

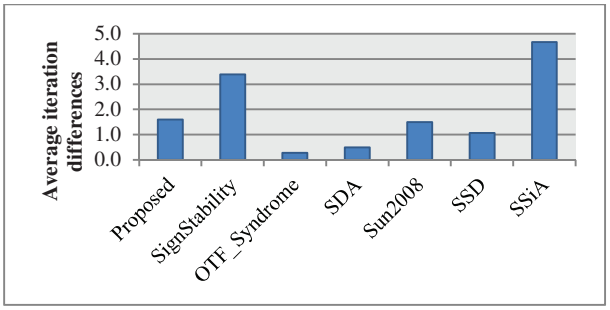

Fig. 1. Difference of average iteration increase compared to the baseline for all methods.

TABLE II. FER SIMULATION RESULTS

Code	Early termination method			
	Proposed	***OTF Syndrome***	***SDA***	***Sun 2008***
WiMAX Rate ½	1.2e-04 at 2.6 dB 4.3e-05 at 3.0 dB	-	10x - 100x at 2.6, 2.8 , 3.0 dB	-
WiMAX Rate 5/6	-	-	3.9e-05 at 6.8 dB 2.8e-05 at 7.0 dB	-
***Dv=3* Rate 1/2 - no FER degradation**				
***Dv=3* Rate ¾**	1.8e-04 at 5.8 dB 2.1e-04 at 6.2 dB	-	-	1.0e-05 at 6.2 dB
***Dv=4* Rate ½**	-	10x - 100x at 3.6, 3.8, 4.0 dB	-	-
***Dv=4* Rate ¾**	-	10x - 100x at 5.8, 6.0, 6.2 dB	10x - 100x at 6.0 and 6.2 dB	-
WPAN Rate ½	-	-	1.9e-03 at 3.0 dB	-
WPAN Rate 3/4	-	-	1.0e-04 at 6.2 dB 2.8e-05 at 6.6 dB	-
WPAN Rate 7/8	-	10x - 100x at 7.4, 7.8, 8,0 dB	2.0e-04 at 8.0 dB	-

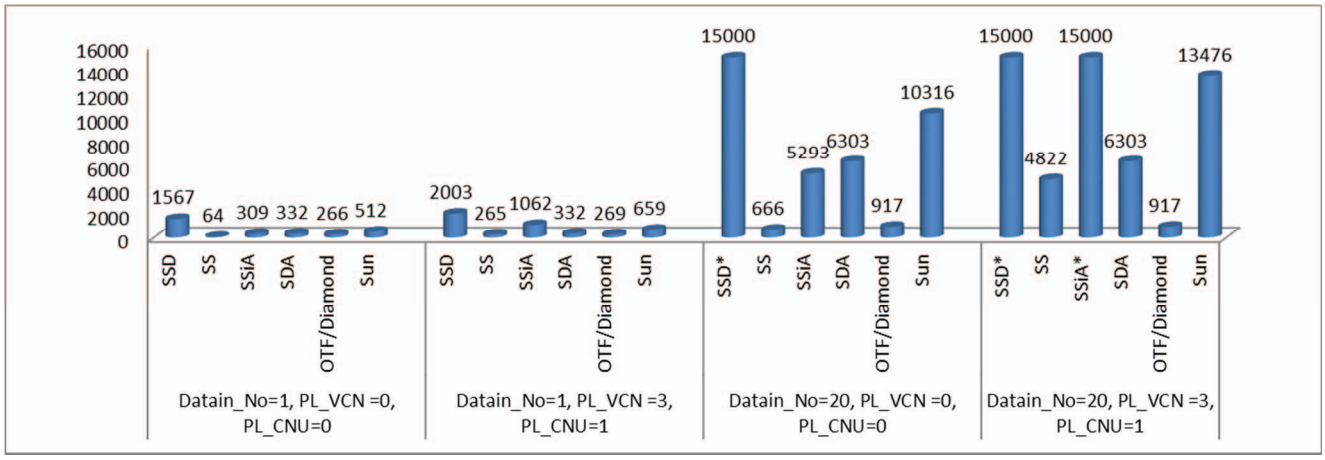

Fig. 2. Synthesis cost estimates for several LDPC architeture variations.

The maximum allowed number of iteration has been fixed to 20. Simulation results yielded no FER degradation for *SignStability*, *SSD* and *SSiA*, for all SNR ranges simulated. Table II highlights the differences compared to the baseline for all other techniques. A significant FER degradation has been obtained for *OTF Syndrome* and *SDA* (up to 2 orders of magnitude for some codes), and a negligible SNR degradation (< 0.2 dB) has been observed for the proposed criteria, as well as *Sun2008*. Furthermore, for WPAN rate 1/2, *SignStability* and *SSiA* fail to reach the stopping condition, reaching the maximum allowed number of iterations.

Regarding the HW implementation, these modules have been designed as off-the-shelf modules, which can be added to an existing LDPC decoder architecture with minimal changes to an existing HW model. Integration wise, the most significant overhead is in the control unit. It must be able to take into account the early termination signal and stop decoding. Furthermore, the implemented modules are fully parameterized (i.e. quantization, circulant size, datapath, number of pipeline stages etc.).

We have obtained synthesis results for WiMAX codes, with circulant size 96 for *Xilinx Virtex 7 - 7vx330tffg1157-3* FPGA platform using Xilinx ISE 14.7. In order to assess the cost of the stopping criteria for different architectures, we have considered the following setup of the hardware architecture: *(i)* number of AP-LLR messages processed in parallel by the processing units (performing both variable and check-node message processing), *(ii)* depth of the pipeline for data-path processing, and *(iii)* pipeline depth for the check-node unit update. These are denoted in Fig.2 with *Datain_No*, *PL_VCN*, and *PL_CNU* respectively. Working frequencies, are in the range 400÷600 MHz. For implementation cost larger than 15000 LUT-FF pairs we have marked them with (*) and have saturated their value. Results show that cost corresponding to methods which require the sign/value from a previous iteration increase with the number of pipeline stages, due to the needed delay buffers. This is the case for *SignStab*, *Sun2008*, *SSiA*, *SSD*. Even for a small number of pipeline stages (3 for the whole data-path, and 1 for check node unit processing), a method such as *SignStability* can have a cost increase of 10x. However, this is not the case for methods that only check results from the current iteration (i.e. *SDA*, *OTF,* and the proposed method,). These latter methods are only influenced by the number of input messages. Another interesting conclusion from the simulation results is that the proposed criterion provides similar performance for all codes analyzed, yielding an almost constant average iteration increase of around 1.6. Furthermore, with two minor exceptions, the FER performance of our new method was not degraded with respect to the baseline.

SignStability was found to be a fast and reliable early termination method for many codes. It is suitable for LDPC architectures with no pipeline, or very few pipeline stages. Performance (FER and average iteration) is code dependent. Results show significant performance loss for codes which have d_v=2. To mitigate this limitation, we have only performed SignStab on the codeword source bits. The efficiency of this solution is code dependent. While it has allowed successful decoding stopping for some matrices (WPAN 1/2), it has proved less efficient for WiMAX ½, where performance drops considerably with respect to the baseline for SNRs above 2.4 dB. Furthermore, for WIMAX rate ½ code, sign stability method yields the highest average iteration number for which decoding can be stopped. *SSiA* has shown a similar limitation being unable to detect codewords for WPAN rates 1/2 and 3/4. *SSiA* and *SSD* are expensive compared to other solutions. For the codes considered in this study the conditions appeared to be a case of overdesign. In most of the cases, they yield both an increase in average iteration count. Other criteria such as *Sun2008*, *and SDA* are also reliable, but require at least 70% more LUT-FF pairs with respect to the least expensive one (*SignStability* or Proposed method depending on the LDPC architecture parameters).

V. Conclusion

Our study highlights the strengths and weaknesses of the existing early termination criteria, and we have proposed a new method for on-the-fly stopping criterion. Its strong points are: *(i)* decoder performance is similar for all codes considered in this study; *(ii)* criterion cost has a weak dependence on the underlying LDPC architecture characteristics (i.e. pipeline); *(iii)* cost overhead is small ($\approx$ 1% cost overhead with respect to a low cost serial AP-LLR processing MS decoder such as [15]). From the studied stopping criteria, we derive that two have proved cost-efficient: the ones based on parity checks, and the one relying on sign stability. More complex stopping criteria might offer zero performance degradation for all kinds of LDPC codes, but they are in many cases overdesigns, and cost-prohibitive.

Acknowledgments

This work has been supported by the Franco-Romanian (ANR-UEFISCDI) Joint Research Program "Blanc–2013" – project DIAMOND.

References

[1] R. G. Gallager, Low Density Parity Check Codes, M.I.T. Press, 1963, Monograph.

[2] IEEE-802.16e, "Physical and medium access control layers for combined fixe and mobile operation in licensed bands," 2005, amendment to Air Interface for Fixed Broadband Wireless Access Systems

[3] IEEE 802.15.3c-2009 "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPAN)", 2009, Amendment 2: Millimeter-wave-based Alternative Physical Layer Extension

[4] R. M. Tanner, "A Recursive Approach to Low Complexity Codes," *IEEE Trans. Information Theory*, vol. 27, no. 5, pp. 533–547, 1981.

[5] M.P.C. Fossorier, "Quasicyclic Low-Density Parity-Check Codes from Circulant Permutation Matrices," *IEEE Trans. on Information Theory*, vol. 50, no. 8, pp. 1788–1793, 2004

[6] F. R. Kschischang and B. J. Frey, "Iterative decoding of compound codes by probability propagation in graphical models," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 219–230, 1998.

[7] D. E. Hocevar, "A reduced complexity decoder architecture via layered decoding of LDPC codes," *Proc. of IEEE Workshop on Signal Processing Systems (SIPS)*, 2004,

[8] J. Li, G. He, H. Hou, Z. Zhang, J. Ma, "Memory Efficient Layered Decoder Design with Early Termination for LDPC Codes", *Proc. IEEE Int. Symp. On Circuits and System(ISCAS)*, 2011.

[9] E. Amador, R. Knopp, R. Pacalet, V. Rezard "On-the-fly Syndrome Check for LDPC Decoders", *Proc. 6th Int. Conf. on Wireless and Mobile Communications (ICWMC)* ,2010.

[10] C.-H. Lin, T.-H.Huang, C.-C.Chen and S.-Y.Lin, "Efficient layer stopping technique for layered LDPC decoding", *Electronic Letters*, Vol. 49, Issue 16, 2013.

[11] Y. Sun, J. R. Cavallaro, "A low-power 1-Gbps reconfigurable LDPC decoder design for multiple 4G wireless standards", *Proc. 28th IEEE SOC Conference*, 2008.

[12] M. Ferrari, S. Belini, A. Tomasoni "Safe Early Stopping of Layered LDPC Decoding", *IEEE Communications Letters*, Vol. 19, Issue 3, 2014.

[13] D. Shin, K. Heo, S. Oh, J. Ha "A Stopping Criterion for Low-Density Parity-Check Codes" *Proc. IEEE 65th Vehicular Technology Conference*, 2007

[14] i-RISC/Deliverable 6.1 "Report on Reliability Aware Synthesis and LDPC Decoders Built with Unreliable Components" online: http://www.i-risc.eu/home/liblocal/docs/iRISC_Deliverables/i-RISC_D6.1.pdf

[15] A. Amaricai, O. Boncalo, V. Savin "Memory Efficient Implementation of Self-Corrected Min-Sum LDPC Decoder" *21st IEEE International Conference on Electronics Circuits and Systems*, Marseille, Dec. 2014.