ACCELERATING AND DECELERATING MIN-SUM-BASED GEAR-SHIFT LDPC DECODERS

Joao Andrade, Gabriel Falcao and Vitor Silva

Instituto de Telecomunicações, Dept. of Electrical and Computer Eng., University of Coimbra, Portugal

ABSTRACT

Low-Density Parity-Check (LDPC) decoders typically implement a single decoding algorithm or update rule, which narrows down the design space of the decoder and maintains its overall simplicity. However, gear-shift techniques combine multiple decoding algorithms, update rules and quantization of the log-likelihood ratios (LLRs), allowing wider design space explorations as more parameters can be fine-tuned to a particular need. Gear-shift LDPC decoders have been shown to improve both the decoding throughput and the energy efficiency per bit decoded, while achieving similar capacity compared to traditional approaches that only use one algorithm. In this paper, we incorporate gear-shift techniques based on the Min-Sum algorithm (MSA) and Self-Corrected Min-Sum algorithm (SCMSA) using variable quantization steps. The proposed design allows bit error rate (BER) performances close to the more powerful SCMSA running only a selected number of iterations using the most powerful update rule.

Index Terms— LDPC Codes, Gear-Shift, Min-Sum, Self-Correction

1. INTRODUCTION

LDPC codes are capacity-approaching coding schemes widely used in modern digital communication and storage systems [1]. Traditionally, the focus on LDPC decoding algorithms is finding a numerically simple decoding algorithm which yields minimal BER degradation. Algorithms such as the MSA [2], SCMSA [3, 4] and improved differential binary (IDB) [5], to name a few, were developed with this principle in mind. Furthermore, quantization strategies were also developed, static and dynamic [6, 7], which minimize the required data bitwidth for the decoder design to have high coding gains. On the other hand, gear-shift techniques have been developed [6, 7, 8, 9, 10, 11], which combine the versatility of adaptive quantization schemes with different decoding algorithms, or node update rules, in order to reach more flexible tradeoffs in both BER performance, energy efficiency of the decoder, decoding latency and throughput. Existing gear-shift decoders work on the principle that a

powerful decoding algorithm is not needed in all decoding iterations, but is required in order to improve on the least powerful algorithm work carried out in the previous iterations. Typically, gear-shift decoders are accelerating – they move from the least to most powerful – whereas a decelerating decoder would commute from most to least powerful algorithm.

In this paper, we propose a gear-shift Min-Sum-based decoder which combines the MSA and the SCMSA. We show that the self-correction can be turned off for a considerable fraction of iterations, either in acceleration or deceleration. Furthermore, as the self-correction increases the bitwidth required, we developed an adaptive bitwidth scheme which reduces the average number of bits required to represent each data element on a standard decoding algorithm.

Algorithm 1 Standard and Self-Corrected Min-Sum	
Initialization: $\alpha_{nm}^{(0)} = \gamma_n$	(1)
CN processing:	
$\beta_{mn}^{(i)} = \prod_{n' \in N(m) \setminus n} \operatorname{sign}\left(\alpha_{n'm}^{(i-1)}\right) \times \min_{n' \in N(m) \setminus n} \alpha_{n'm}^{(i-1)} $	(2)
BN processing:	
$\alpha_n^{(i)} = \gamma_n + \sum_{\substack{m' \in M(n) \setminus n}} \beta_{m'n}^{(i)}$	(3)
$\tilde{\alpha}_{mn}^{(i)} = \alpha_n^{(i)} - \beta_{mn}^{(i)}$	(4)
(a, i, j, (i-1)) = ((i)) = (i-1) (i-1)	~

$$\alpha_{mn}^{(i)} = \begin{cases} 0, d_H \{ sgn(\alpha_{mn}^{(i-1)}) \oplus sgn(\alpha_{mn}^{(i)}) \} = 1 \land \alpha_{mn}^{(i-1)} \neq 0 \\ \tilde{\alpha}_{mn}^{(i)}, \text{ for MSA and otherwise in the SCMSA case.} \end{cases}$$
(5)

2. LDPC CODES

LDPC codes are linear block codes defined as: $\mathbf{c} \times \mathbf{H}^{T} = \mathbf{0}$, $\mathbf{c} \in \mathbf{C}$, where \mathbf{H} is the parity-check matrix which defines the LDPC code, \mathbf{c} is a codeword from the set \mathbf{C} of possible codewords. \mathbf{H} is an adjacency matrix to the bipartite graph with columns defining Bit Nodes (BNs) and rows defining Check Nodes (CNs) [12].

LDPC soft-decoding algorithms are based on messagepassing over the Tanner graph of the code [12] and the MSA and the SCMSA are formalized in Algorithm 1 [3, 4]. The

This work was supported by Instituto de Telecomunicações and the Portuguese Foundation for Science and Technology (FCT) under grants UID/EEA/50008/2013 and SFRH/BD/78238/2011.



Fig. 1. ROI of the gear-shift decoder is located between the SCMSA BER curve and the MSA. The BER curves were plotted for the DVB-S2 rate 1/3, N = 64800 bits.

channel demodulator produces a likelihood estimate – in this case an LLR – on the received bits, γ_n (1). CNs will gather $\alpha_{mn}^{(i)}$ messages from their adjacent nodes – denoted as the set N(m) – and will produce β messages (2). Likewise, BN will gather β messages to produce new estimates on α messages. CN processing is the same for the MSA and the SCMSA (2); and the BN processing is defined by (4) and by (5), respectively for the former and the latter.

3. GEAR-SHIFT LDPC DECODERS

The region of interest (ROI) of the gear-shift decoder that combines the standard and the self-corrected MSA is shown in Figure 1. The objective of the gear-shift design is to achieve the BER performance of the most powerful update rule, i.e. the SCMSA, while running the least powerful update rule – the MSA – for the longest number of iterations possible. As such, the ROI is delimited by the SCMSA and the MSA BER performance.Throughout this work, we refer to the fixed-point precision as Qx.y, with x the number of integer bits including the sign bit, y the number of decimal bits and x+y the bitwidth of the LLR.

3.1. Variable Quantization Bits

LDPC decoders only require a limited number of fixedprecision bits to operate within a negligible margin of high precision floating-point equivalent BER [4, 6]. This observation, compounded by the fact that for low signal-to-noise ratio (SNR) values the magnitude of LLRs is also low and take a considerable number of iterations to grow in magnitude before being clipped due to the limited precision, allows us to explore narrower bitwidths for the initial decoding iterations. While a dynamic LLR quantization watchdog could be employed to adjust in the face of variable SNR conditions, we prefer to keep the simplicity of the decoding design. We statically define a threshold for increasing the bitwidth of each internal data word, by performing an *a-priori* sweep of the most suitable threshold candidates, which commutes from a low precision bitwidth b_{low} to a high precision bitwidth b_{high} at iteration threshold t_b . A natural advantage of using the self-correction as a gear in the decoding process is the ability to reduce the required bitwidth from X_f to $X_s = X_f - 2$ [4]. While in bit-parallel implementations this comes as reducing the switching activity on said wires to 0, on bit-serial or pulse-width implementations, there is a reduction of at least 2 clock cycles when the self-correction is deactivated, and a greater reduction is possible when combined with variable quantization.

3.2. Gearbox: Standard and Self-Correction

Activation and deactivation of the self-correction on the MSA can be seen as shifting a gear. When the least powerful decoder, in this case the MSA, is running in the first iterations and the most powerful is running at trailing iterations, we can stretch the metaphor into an accelerating decoder, whereas the opposite can be though as a decelerating decoder. The rationale for development of an accelerating decoder and a decelerating one takes advantage of the following aspects, provided that the SNR of the channel is no known to the LDPC decoder.

For low SNR values, LLRs are also low and thus, the advantage of a decelerating decoder comes from its increased decoding capabilities in its initial iterations. When LLRs have grown, the decoder then shifts into the algorithm with the least decoding capabilities, as it will not face a significant degradation of BER. On the other hand, for high SNR conditions, both decoders might be at the error-floor region, which as seen in Figure 1 is around 10^{-12} for both the MSA and the SCMSA decoding algorithms. Thus, the accelerating decoder will run the least powerful decoder and to make sure that the BER is improved for SNR values in the ROI, the decoder "revs up" by activating the self-correction. The drawback of the decelerating approach is the breaking of the principle of only consuming more energy once a certain threshold has been reached on the least powerful update rule and the transmitted word has not been successfully decoded.

As done previously with the quantization bits threshold, we performed an *a-priori* sweep exploring the threshold of acceleration and deceleration. The gear-shift decoder commutes from the least to the most powerful algorithm, turning on the self-correction at the trailing decoding iterations, when accelerating, and does the opposite when decelerating. The *a-priori* sweep found the most suitable threshold, t_a , to perform the activation or deactivation of the self-correction. We designate this as shifting a gear, and thus, the combined effect



Fig. 2. BER of the (A)ccelerating and the (D)ecelerating gear-shift decoders.

of the standard MSA and the SCMSA can be thought of as a gearbox. The remaining challenge is then to combine both the quantization and the gear-changing thresholds.

4. EXPERIMENTAL RESULTS

Experimental validation was performed using the the DVB-S2 N = 64,800 bits rate 1/3 LDPC code, running a maximum of 50 decoding iterations [13, 14]. The architecture of the decoder is based on the bit-parallel architecture in [15].

4.1. Accelerating vs. Decelerating Decoding

The BER performance of the accelerating and decelerating decoders without variable quantization are shown in Figure 2a). As seen, there are clear differences between the BER of each strategy. Whereas the decelerating decoders – running the SCMSA and then the MSA – show a degrading BER when compared to the SCMSA BER, they follow a similar trajectory and gradient. The accelerating decoders on the other hand, show both BER trajectory and gradient closer to that of the MSA BER. This suggests that the initial decoding iterations heavily impact on the final BER. Running the more powerful SCMSA in the beggining yielded better BER than the least powerful MSA algorithm first.

4.2. Variable Quantization Shifting

Introducing a two-step variable quantization of the LLRs's bitwidth in the previous design yields the BER performance shown in Figure 2b), for the best performing combinations of $(t_a, t_b) = (20, 15)$, combinations of low precision bitwidths $b_{low}, b_{high} \in Q\{4.2, 5.2, 6.2\}$. A further BER degradation is observed for both gear-shift strategies. Namely, the dominant effect in terms of BER curve trajectory on high SNR regions is the bitwidth selected as the lowest precision.The

decoder using $(b_{low}, b_{high}) = (Q4.2, Q5.2)$ yields the same error-floor as the $(b_{low}, b_{high}) = (Q4.2, Q6.2)$.

Table 1. Difference (%) in decoding iterations of the gearshift decoders A_3 and D_3 compared to SCMSA and MSA.

$\frac{SNR}{Algorithms}SCMSA - A_3MSA - A_3SCMSA - D_3MS$	$A - D_3$
-1.00 -8.97 3.04 -10.08	1.04
-0.90 -22.41 0.58 -18.39	0.48
-0.80 -24.37 3.90 -21.05	2.13
-0.70 -20.25 -0.45 -21.89	-6.94
-0.60 -18.89 -4.91 -22.65	-14.08
-0.50 -18.75 -8.37 -23.56	-19.38

5. COMPLEXITY ANALYSIS

The proposed gear-shift decoder is implementable following the architecture shown in Figure 3 [15]. Essentially, the selfcorrection is appended to the end of the datapath of the BN unit. On the CN unit, there is only one operation related with the knowledge of a previously introduced erasure [4]. The data representation is such that 2 extra bits are required, one for $sgn(\alpha_{mn}^{(i-1)})$ and an erasure flag stating whether $\alpha_{mn}^{(i-1)} =$ 0. The complexity of operating this particular LDPC architecture is shown in Table 2, expressed in numeric operations and consequent logic blocks required. As seen, an extra comparator, XOR and AND are required to implement the selfcorrection, plus an additional 2 bits making the bitwidth of the LLR word X_f . The decoding unit runs in MSA mode, when the selector s defines $\tilde{\alpha}_{mn}^{(i)}$ as output. In Table 1, the number of decoding iterations required for the A_3 and D_3 decoders is shown, and understandably, is greater than what the SCMSA requires (columns $SCMSA - \{A_3, D_3\}$). In the waterfall region ([-1.0, -0.80] dB), it is also higher than what the MSA



Fig. 3. Datapath of the BN and the CN units. As shown, the self-correction can be appended to the MSA unit, and can be turned on or off according to the update rule in place by fixing the MUX selector s to let $\tilde{\alpha}_{mn} = \alpha_{mn}$.

Table 2. Numerical complexity of the MSA and SCMSA and hardware resources of their implementation [15].

Operations/Algorithm	CN		Resources/Algorithm	CN	BN
Operations/Argorithin	Comparisons	XOR	Resources/Algorium	Shared MSA and SCMSA Blocks	
MSA & SCMSA (2)	$2d_v - 2$	$d_v - 1$	FIFO depth	$\max\{d_c\}$	$\max\{d_b\}$
	BN		Registers	6	2
	Additions	XOR	Selectors	4	2
MSA (4)	d_v	-	XORs	2	-
SCMSA (5)	d_v	d_v	d_v	Self-Correction	
			Comparator	1	-
			XOR	-	1
d_v and d_c are the BN and CN weight respectively [4]			AND	-	1

requires (columns $MSA - \{A_3, D_3\}$), since the difference in iterations is positive. However, in the range $[-0.7, \infty[$ dB, the gear-shift designs start running less iterations than the MSA. Whenever this difference is negative, the gear-shift decoder is running with lower efficiency as it requires more decoding iterations to decode successfully. Namely, in the waterfall -1.0 dB data point, the improvement is in the order $2\sim 4\%$ less decoding iterations required as opposed to MSA approaches, respectively for the accelerating and the decelerating gear-shift decoder. However, there will be an increase in the number of decoding iterations as the SNR conditions improve and the decoders enter the error-floor region.

6. RELATION TO PRIOR WORK

The concept of gear-shift decoding has been introduced in [8] and has seen considerable improvements for hardware realization using bit-serial techniques and on digit-online decoders [9, 10]. The latter works focus also on variable bitwidth representations of data to achieve higher power savings, the main subject of [6, 7]. In [11], a tri-mode decoding which combines the versatility of 3 decoding algorithms to achieve a good tradeoff between coding losses and power gains is proposed, which combines distinct update rules. Tuning of the data representation bitwidth was explored similarly to [6, 7, 10], with varying degrees of success. When using algorithms with very low bitwidth requirements, higher gains are possible when compared to the proposed Min-Sumbased approach which is more sensitive to low bitwidths. The proposed gear-shift decoder allows for portions of the hardware to have no switching activity, thereby increasing its energy efficiency without the implementation of a significant logic control module [10], and without the need to implement hardware modules which can be used by widely distinct update rules [10, 11]. Moreover, the characteristics of the proposed gear-shift MSA decoder exhibit a better decoding performance when decelerating instead of when accelerating, in the waterfall and beginning of error floor region, as it is commonly found in prior work.

7. CONCLUSIONS

The presented decoder provides a tradeoff between the MSA and SCMSA BER, depending on the set of parameters chosen: data bitwidth and how long and how is the self-correction active – accelerating or decelerating the decoding procedure. In the proposed architecture, the self-correction consumes a low number of extra logic modules and negligible control is required. Bitwidth requirements increase due to the selfcorrection, but this does not add to the energy required to decode each bit when running the MSA. The benefits of this approach are not limited to bit-parallel implementations, but are also applicable to pulse-width decoders. Furthermore, the MSA-based gear-shift decoder exhibits a better performance when decelerating rather than when accelerating.

8. REFERENCES

- R. G. Gallager, *Low-Density Parity-Check Codes*, MIT Press, Cambridge, 1963.
- [2] N. Wiberg, H.-a. Loeliger, and R. Kotter, "Codes and Iterative Decoding on General Graphs," in *IEEE Symp.* on Inf. Theory, 1995, p. 468.
- [3] V. Savin, "Self-corrected Min-Sum Decoding of LDPC codes," in *IEEE Symp. on Inf. Theory*, 2008, pp. 146– 150.
- [4] J. Andrade, G. Falcao, V. Silva, J. P. Barreto, N. Goncalves, and V. Savin, "Near-LSPA performance at MSA complexity," in *Proc. IEEE Int. Conf. on Commun.*, 2013, pp. 3281–3285.
- [5] Kevin Cushon, Saied Hemati, Camille Leroux, Shie Mannor, and Warren J. Gross, "High-Throughput Energy-Efficient LDPC Decoders Using Differential Binary Message Passing," *IEEE Trans. Signal Processing*, vol. 62, no. 3, pp. 619–631, 2014.
- [6] T. Mohsenin, H. Shirani-mehr, and B. M. Baas, "LDPC Decoder with an Adaptive Wordwidth Datapath for Energy and BER Co-Optimization," *Hindawi Journal of VLSI Design*, vol. 2013, no. 1, pp. 1–14, 2013.
- [7] C.-H. Chung, Y.-L. Ueng, M.-C. Lu, and M.-C. Lin, "Adaptive quantization for low-density-parity-check decoders," in *IEEE Symp. on Inf. Theory and its Applications*, 2010, pp. 13–18.
- [8] M. Ardakani and F. R Kschischang, "Gear-shift Decoding," *IEEE Trans. Commun.*, vol. 54, no. 7, pp. 1235– 1242, 2006.

- [9] A. Darabiha, A. C. Carusone, and F. R Kschischang, "A Bit-serial Approximate Min-sum LDPC Decoder and FPGA Implementation," in *Proc. IEEE Int. Symp. on Circuits and Systems*, 2006, pp. 4–pp.
- [10] K. Cushon, S. Hemati, S. Mannor, and W. J. Gross, "Energy-efficient gear-shift LDPC decoders," in *Proc. IEEE Int. Conf. on Application-specific Systems, Architectures and Processors*. IEEE, 2014, pp. 219–223.
- [11] X. Chen, Q. Huang, S. Lin, and V. Akella, "FPGAbased Low-complexity High-throughput Tri-mode Decoder for Quasi-cyclic LDPC Codes," in Annual Allterton Conf. on Communication, Control and Computing, 2009, pp. 600–606.
- [12] M. Tanner, "A Recursive Approach to Low Complexity Codes," *IEEE Trans. Inform. Theory*, vol. 27, no. 5, pp. 533–547, Sep 1981.
- [13] A. Morello and V. Mignone, "DVB-S2: the Second Generation Standard for Satellite Broad-band Services," *Proceedings of the IEEE*, vol. 94, no. 1, 2006.
- [14] G. Falcao, J. Andrade, V. Silva, and L. Sousa, "GPUbased DVB-S2 LDPC Decoder with High Throughput and Fast Error Floor Detection," *Electronics Letters*, vol. 47, no. 9, pp. 542–543, 2011.
- [15] G. Falcao, M. Gomes, V. Silva, L. Sousa, and J. Cacheira, "Configurable M-factor VLSI DVB-S2 LDPC Decoder Architecture with Optimized Memory Tiling Design," *EURASIP Journal on Wireless Commun. and Networking*, vol. 2012, no. 1, pp. 98, March 2012.