One Minimum Only Trellis Decoder for Non-Binary Low-Density Parity-Check Codes

Jesús O. Lacruz, Francisco García-Herrero, Javier Valls *Member, IEEE*, David Declercq *Senior Member, IEEE*

Abstract

A one minimum only decoder for Trellis-EMS (OMO T-EMS) and for Trellis-Min-max (OMO T-MM) is proposed in this paper. In this novel approach, we avoid computing the second minimum in messages of the check node processor, and propose efficient estimators to infer the second minimum value. By doing so, we greatly reduce the complexity and at the same time improve latency and throughput of the derived architectures compared to the existing implementations of EMS and Min-max decoders. This solution has been applied to various NB-LDPC codes constructed over different Galois fields and with different degree distributions showing in all cases negligible performance loss compared to the ideal EMS and Min-max algorithms. In addition, two complete decoders for OMO T-EMS and OMO T-MM were implemented for the (837,726) NB-LDPC code over GF(32) for comparison proposals. A 90 nm CMOS process was applied, achieving a throughput of 711 Mbps and 818 Mbps respectively at a clock frequency of 250 MHz, with an area of $\mathbf{19.02mm^2}$ and $\mathbf{16.10mm^2}$ after place and route. To the best knowledge of the authors, the proposed decoders have higher throughput and area-time efficiency than any other solution for high-rate NB-LDPC codes with high Galois field order.

Index Terms

NB-LDPC, OMO T-EMS, OMO T-MM, Check node processing, low-latency, VLSI design

J. Lacruz is with the Electrical Engineering Department, Universidad de Los Andes, Mérida, 5101, Venezuela. (e-mail: jlacruz@ula.ve)

F. García, and J. Valls are with the Instituto de Telecomunicaciones y Aplicaciones Multimedia, at Universitat Politècnica de València, 46730 Gandia, Spain (e-mail: fragarh2@epsg.upv.es, jvalls@eln.upv.es).

D. Declercq is with the ETIS Laboratory, ENSEA/Univ. Cergy-Pontoise/CNRS-UMR-8051, 6, Avenue du Ponceau, F-95000, Cergy-Pontoise, France (e-mail: david.declercq@ensea.fr).

I. Introduction

Since the first non-binary low-density parity-check (NB-LDPC) decoder architecture was proposed for the Q-ary Sum-of-Product algorithm (QSPA) [1], hardware designers have been working to derive solutions that allow the use of NB-LDPC codes in a wide range of communication and storage systems. Good error correction, high throughput and small area remain the challenge of any NB-LDPC decoder designer.

Extended Min-Sum (EMS) [2] and Min-Max (MM) [3] algorithms were proposed, with the aim of reducing the complexity involved in the check node processor, which is the bottleneck of QSPA algorithm. Although the decoding process is simplified by means of using forward-backward for the extraction of check-to-variable messages, these metrics penalize the maximum throughput achievable when they are implemented in hardware.

To avoid the use of forward-backward, in [4] the Trellis Extended Min-Sum (T-EMS) was proposed. With T-EMS, the degree of parallelism is increased using only combinations of the most reliable Galois field (GF) symbols to compute the check-to-variable messages. The decoder presented in [4] was outperformed in [5] where an extra column is added to the original trellis with the purpose of generating the check-to-variable messages in a parallel way. The main drawback of the approach presented in [5] is that requires a lot of area and pipeline stages, reducing the overall efficiency of the decoder. In [6] the hardware implementation of a T-EMS decoder is described, reaching the highest throughput found in literature. Previous trellis-based proposals, such as the ones from [7], [8] and [9], applied partial-parallel decoding as a way to obtain the output messages in the check node processor.

In [10] a decoder architecture named Relaxed Min-Max (RMM) is proposed. RMM makes an approximation for the second minimum calculation and hence, generates the check-to-variable messages with less complexity. The main drawbacks for this approach are: i) the check node output messages are derived serially, reducing the overall throughput of the decoder and increasing latency; and ii) the proposed approach suffers of an early degradation in the error floor region, due to the way of deriving the second minimum.

In this paper, we introduce a novel second minimum approximation based on the statistical analysis of the check node messages named as One Minimum Only (OMO) decoder. The motivation to perform this approximation is that the two-minimum finder duplicates the critical path and increases the complexity of the check node processor. In addition to the second minimum estimator proposed in [10], we analyze two other estimators: one based on a slight modification of the one-minimum finder, and a last one which linearly combines the first two estimators, and showed the best performance in simulations. The proposed OMO decoder can be applied to both T-EMS and Trellis Min-max obtaining OMO T-EMS and OMO T-MM decoders respectively. By avoiding the use of two-minimum finders [11] in the check node, we were able to reduce both area and latency of the check node update without introducing any performance loss compared to the original EMS or Min-max algorithms.

The OMO T-EMS and OMO T-MM check node architectures have been implemented and included in a layered scheduling decoder. A 90nm CMOS process has been employed and a (837,726) NB-LDPC code over GF(32) has been chosen to show the efficiency of our approach for high order fields and high rate codes. The OMO T-EMS and OMO T-MM decoders achieve 100% and 159% higher efficiency (Mbps / Million Gates) compared to the most efficient decoder found in literature [10] respectively, with about 30% less latency and 40% higher throughput than the solution from [6] depending on whether EMS or Min-max version is implemented.

The rest of the paper is organized as follows: in Section II we introduce the nomenclature and the main concepts of T-EMS algorithm. The proposed approach for the second minimum estimation of T-EMS algorithm, OMO T-EMS, is presented in Section III, including and analysis of performance for different NB-LDPC codes and showing that can be extended to Min-max algorithm without loss of generality. Section IV includes the hardware implementation of the proposed check node and the overall decoder. Moreover, synthesis and post place and route results of the design and comparisons with other architectures are also included. Finally, conclusions are outlined in Section V.

II. Trellis - Extended Min-Sum algorithm

NB-LDPC codes are characterized by a sparse parity check matrix **H** where each non-zero element $h_{m,n}$ belongs to Galois field $GF(q=2^p)$. We consider regular NB-LDPC codes with constant row weight d_c and column weight d_v. Decoding algorithms for NB-LDPC codes use iterative message exchange between two types of nodes called check nodes (CN) (M rows of **H**) and variable nodes (VN) (N columns of **H**).

Let $\mathcal{N}(m)$ ($\mathcal{M}(n)$) be the set of VN (CN) connected to a CN (VN) m (n). Let $Q_{m,n}(a)$ and $R_{m,n}(a)$ be the edge messages from VN to CN and from CN to VN for each symbol $a \in GF(q)$ respectively. $L_n(a)$ denotes the channel information and $Q_n(a)$ the *a posteriori* information.

Algorithm 1: T-EMS Algorithm

```
  Input: $\mathbf{Q_{m,n}}$ , $z_n = \arg\min_{a \in GF(q)} Q_{m,n}(a)\ \forall\ n \in \mathcal{N}(m)$
  for $j = 1 \to d_c$ do
1     $\Delta Q_{m,n_j}(\eta_j = a + z_{n_j}) = Q_{m,n_j}(a)$
  end
2 $\beta = \sum_{j=1}^{d_c} z_{n_j} \in GF(q)$
3 $\Delta Q(a) = \min_{\eta'_j(a) \in conf(n_r,n_c)} \sum_{j=1}^{d_c} \Delta Q_{m,n_j}(\eta'_j(a)), a \in GF(q)$
  for $j = 1 \to d_c$ do
4     $\Delta R_{m,n_j}(a + \eta'_j(a)) = \min(\Delta R_{m,n_j}(a + \eta'_j(a)), \Delta Q(a) - \Delta Q_{m,n_j}(\eta'_j(a)))$
5     $R_{m,n_j}(a + \beta + z_{n_j}) = \lambda \cdot \Delta R_{m,n_j}(a), a \in GF(q)$
  end
  Output: $\mathbf{R_{m,n}}$
```

Let $\mathbf{c} = c_1, c_2, \cdots, c_N$ and $\mathbf{y} = y_1, y_2, \cdots, y_N$ be the transmitted codeword and received symbol sequence respectively, with $\mathbf{y} = \mathbf{c} + \mathbf{e}$ and $\mathbf{e}$ is the error vector introduced by the communication channel. The log-likelihood ratio (LLR) for each received symbol is obtained as $L_n(a) = \log[P(c_n = z_n|y_n)/P(c_n = a|y_n)]$ ensuring that all values are non-negative where z_n is the symbol associated to the highest reliability.

Trellis Extended Min-Sum (T-EMS) algorithm [4] presents a way of implementing the original EMS algorithm [2], avoiding the use of the forward-backward metrics and increasing the degree of parallelism of the CN. Algorithm 1 includes the original T-EMS CN algorithm, where the first and fifth steps perform the transformation of incoming messages ($\mathbf{Q_{m,n}}$) from "normal" to delta domain (N $\to \Delta$) and from delta domain to normal domain ($\Delta \to$ N) for the CN outgoing messages ($\mathbf{R_{m,n}}$) respectively. For the N $\to \Delta$ transformation, syndrome β of the CN must be obtained (Step 2 of Algorithm 1) using the incoming tentative hard decision z_n for each CN message.

The extra column ($\Delta Q(a)$) calculation is derived on step 3 using the configuration sets originally proposed in [2], with the aim of building the output messages using only the most reliable information. The configuration set $conf(n_r, n_c)$ is defined as the set of at most n_r symbols that satisfy the parity equation, deviating at most n_c times from the combination (path) of symbols with the highest reliability. Considering only the case when $n_r = 1$ and $n_c = 2$, the extra column $\Delta Q(a)$ is built with the combination of the most reliable messages for each GF(q) symbol i.e. with the minimum value message, $min_1(a)$.

Once the $\Delta Q(a)$ values are derived, the outgoing CN messages in delta domain $\Delta \mathbf{R_{m,n}}$, are generated in Algorithm 1 using step 4 which provides all the values for extrinsic CN outgoing messages. For the intrinsic values, $min_1(a)$ and $min_2(a)$ are used as it is explained in detail in [5].

It is important to remark that the $min_1(a)$ values are used for both, $\Delta Q(a)$ and $\Delta R_{m,n}(a)$ generation while $min_2(a)$ is only used to compute $\Delta R_{m,n}(a)$ (in the case of $n_r = 1$ and $n_c = 2$). Additionally, the extraction of the position of the first minimum is also required ($min_{1_{pos}}(a)$), since this information is used to derive the path for each extra column value in the trellis. However, the two minimum values must be processed using a two-minimum finder before the extra column calculation. This two-minimum finder increases the critical path for $min_1(a)$ due to the $min_2(a)$ extraction.

In next section we propose a novel approach to approximate the second minimum, which at the same time that reduces the critical path to get the first minimum, achieves an accurate estimation of the second one without degrading the performance of the original T-EMS and Min-max algorithms.

III. One Minimum Only Trellis Decoder

As shown in Section II, the two-minimum finder represents an important part of the CN architecture. On the other hand, the hardware architectures to implement the minimum finder processor [11] introduce the same delay for both $min_1(a)$ and $min_2(a)$, which is not optimal for EMS and Min-max algorithms.

This observation is our principal motivation for creating a novel check node architecture which approximates the computation of the second minimum, reducing the delay for the first one and hence improving the latency and the throughput of the decoder as it can be seen in next sections. Our proposed approach has been tested on multiple NB-LDPC codes with different GF(q) and degree distribution, showing in all cases a negligible performance loss compared with the T-EMS and Min-max algorithms. In order to simplify the description of our proposal, we will focus on T-EMS, however, this method can be directly derived to Min-max algorithm without any loss of generality. However, we will provide performance and implementation results of this new solution for both EMS and Min-max decoders.

In the rest of the section, different estimators of the second minimum values are considered, and a statistical analysis of their distribution compared to the true second minimum is made. The analysis is done for the (837,726) NB-LDPC code over GF(32), where **H** is generated using the methods proposed in [12], with circulant sub-matrices of size $(q-1)\times(q-1)$. However, other codes with different GF and degree distribution have been tested obtaining the same behavior.

A. Estimators for the second minimum value

A first natural solution for the estimation of min_2 is to make use of a scaled version of the first minimum min_1, described in equation (1):

$$min_2''(a) = min_1(a) \times \gamma_p \tag{1}$$

This approximation has been already proposed in [10]. However, by just applying equation (1) the value of the minimum is usually underestimated if we apply a γ_p value that mimics as much as possible the behavior of EMS or Min-max in the waterfall region [1]. As it can be seen in Fig.1, where we draw the distributions of the true $min_2(a)$ and their proposed estimators, the value of $min_2''(a)$ is on average smaller than the real $min_2(a)$, which leads to an important performance degradation in the error floor region.

A second possible estimator makes advantage of a re-use of the hardware architecture. Using a radix-2 one-minimum finder is possible to determine an early estimation for the second minimum. In Fig. 3, a one-minimum tree finder is presented. In the figure, we include an extra multiplexor in the last stage, that allows extracting the looser term, denoted $min_2'''(a)$. By doing so and just using an extra multiplexor, this term can be used as an early estimator of the second minimum, which represents an upper-bound on the true minimum value. If the true $min_2(a)$ value is located in the other half part of the tree that $min_1(a)$ ($d_c/2$ branches of the minimum tree finder not connected to $min_1(a)$), then we obtain $min_2'''(a) = min_2(a)$. In the other cases, $min_2'''(a) > min_2(a)$. Hence, the resultant value corresponds to an provable upper bound on the true $min_2(a)$. A systematic over-estimation of the second minimum value could lead also to performance degradation of the complete decoder, and we propose to combine $min_2''(a)$ and $min_2'''(a)$ in order to get an estimator with a better statistical behavior.

[1] γ_p is selected as the mean value of the ratio between $min_2(a)$ and $min_1(a)$. $\gamma_p = min_2(a)/min_1(a)$

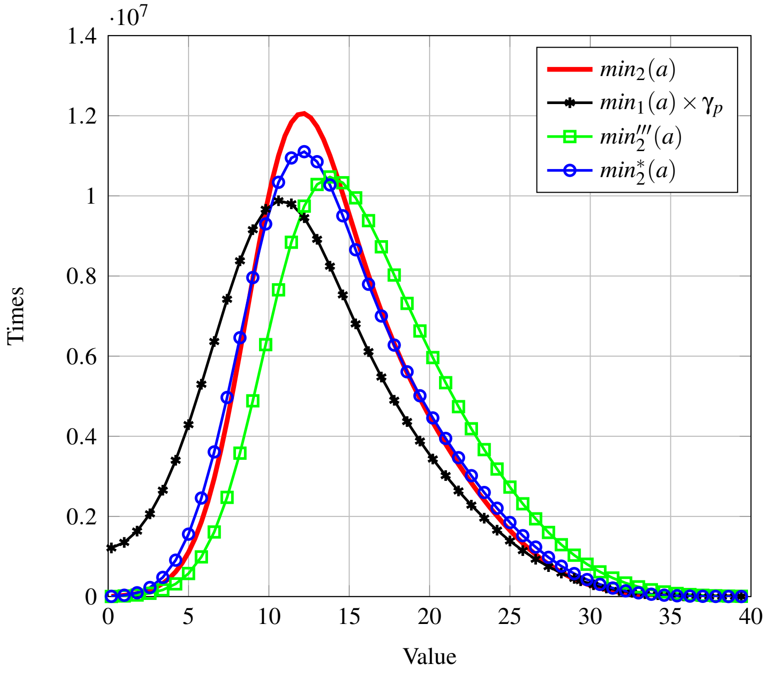

Fig. 1. Histograms for the different estimators of $min_2(a)$. The γ_p value was set to 1.125.

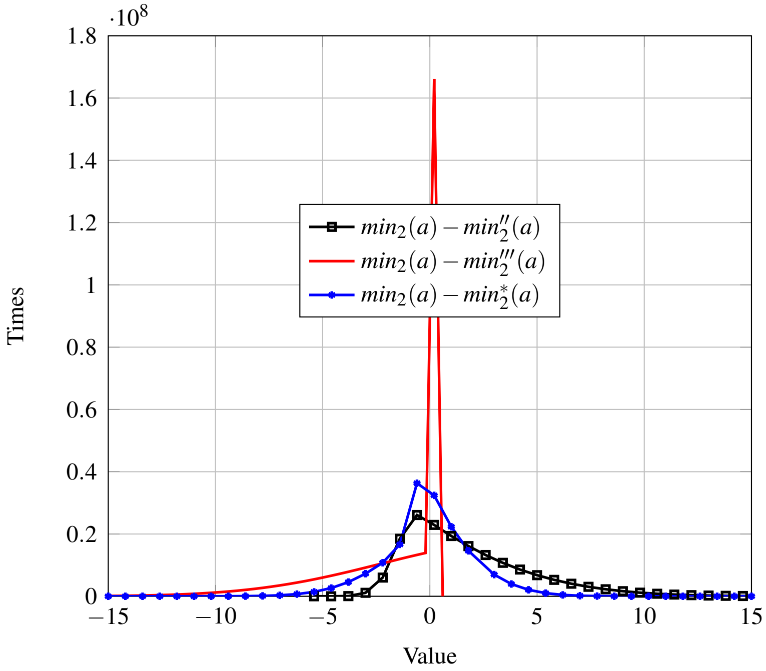

Fig. 2. Histograms showing the error distribution of different estimators of $min_2(a)$. The γ_p value was set to 1.125.

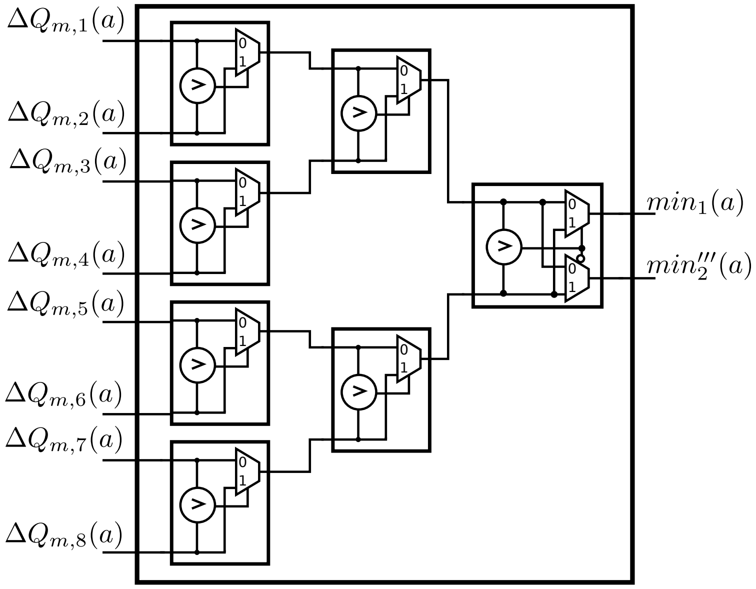

Fig. 3. Second minimum estimation based on a radix-2 one-minimum finder. Example for an eight inputs tree.

As we will demonstrate with a statistical analysis in the next section, both $min_2''(a)$ and $min_2'''(a)$ are biased estimators, one over-estimating the true second minimum, and the other one under-estimating the true second minimum. We therefore propose in this paper to combine those two estimators, by using a linear combination of the two preceding estimators, in the following way:

$$min_2^*(a) = \frac{min_2''(a) + min_2'''(a)}{2} = \frac{min_1(a) \times \gamma_p + min_2'''(a)}{2} \tag{2}$$

Compared to the real $min_2(a)$ values, $min_2^*(a)$ presents a similar behavior in the histogram shown in Fig. 1 which implies that the proposed estimation has similar statistical behavior than the exact $min_2(a)$ values.

The operations involved to implement (2) can be performed after $min_1(a)$ and $min_2'''(a)$ values are obtained (using the hardware structure in Fig. 3). Therefore, the second minimum estimation can be made at the same time that $\Delta Q(a)$ values are obtained, to finally calculate check-to-variable output messages.

In the next section, we analyse the statistical behavior of each of the three proposed estimators.

B. Statistical analysis of the different estimators

In Fig. 2, we plot the distributions of the difference between the proposed estimators, $\widehat{min_2}(a)$ being defined following one of the equations (1)-(2), and the true minimum, *i.e.* $p\left(min_2(a)-\widehat{min_2}(a)\right)$. We performed this analysis by computing for each iteration and for different E_b/N_0 values the difference between the real second minimum at the check node and each one of the estimators. The information for this analysis is computed based on all the M check nodes of the parity check matrix.

From the shape of the distributions, we can see that $\widehat{min_2}(a)=min_2''(a)$ seems to be biased and skewed to the positive values of the difference, which means that not only $min_2''(a)$ under-estimates the true minimum, but also that the difference is not symmetric around its bias. Of course, as it was expected for $\widehat{min_2}(a)=min_2'''(a)$, which represents a upper bound on the true second minimum, we get the opposite behavior, as the distribution of $min_2(a)-min_2'''(a)$ is left biased and skewed. In order to better measure the performance of each estimator, we have computed the first four cumulants of the distributions $p\left(min_2(a)-\widehat{min_2}(a)\right)$, and reported their values in Table I after 1 and 15 decoding iterations. The first cumulant of the distribution is the *mean*, and measures the bias of the estimator, a value of zero indicating that the estimator is unbiased. The second cumulant is the *square-root of the variance*, which indicates the spread of the estimator around the mean value. The third cumulant is the *skewness*, and is a measure of the symmetry of the distributions. A zero skewness indicates that an estimator does not favor positive or negative difference with the true value $min_2(a)$. Finally, the fourth cumulant, called the *kurtosis* is a measure of the flatness of the tails of the distribution. A low value of the kurtosis indicates that very large outliers values of the difference with the true minimum do not appear with high probability. The kurtosis value for a Gaussian distribution is equal to 3.

As we can see from those tables, $min_2''(a)$ typically tends to under-estimates the true $min_2(a)$ value, since both the mean and the skewness are positive, while $min_2'''(a)$ over-estimates the true $min_2(a)$ value, since both the mean and the skewness are negative (which was expected as $min_2'''(a)$ is actually an upper bound of $min_2(a)$). As we can see, the third estimator that we propose, namely $min_2^*(a)$, is a better estimation than the other 2, with respect to all statistics. First it is practically unbiased at the first iteration, although a slight positive bias seems to appear

TABLE I

STATISTICAL PROPERTIES OF THE DIFFERENT $\widehat{min_2}(a)$ ESTIMATORS AFTER $I=1$ AND $I=15$ DECODING ITERATIONS.

	$p\left(min_2(a)-min_2''(a)\right)$	$p\left(min_2(a)-min_2'''(a)\right)$	$p\left(min_2(a)-min_2^*(a)\right)$
mean ($I=1/I=15$)	1.70 / 2.09	-1.74 / -1.67	-0.07/0.16
σ ($I=1/I=15$)	2.94 / 2.65	2.83 / 2.74	2.04 / 1.92
Skewness ($I=1/I=15$)	1.17 / 1.09	-0.11 / -0.25	-0.0011 / -0.0025
Kurtosis ($I=1/I=15$)	4.51 / 4.18	5.82 / 5.71	4.05 / 3.99

at iteration 15. The skewness is almost zero, which tells us that, on average, the decoder will under-estimate or over-estimate the min_2 with the same frequency. Finally, both the variance and the kurtosis of $p\left(min_2(a)-min_2^*(a)\right)$ are the minimum among the three estimators, which indicates that values very different than the true minimum will appear less often with $min_2^*(a)$ than with the other two estimators. With respect to those indicators, $min_2^*(a)$ is a better estimator of min_2 than $min_2''(a)$ or $min_2'''(a)$. We will confirm in the next section that $min_2^*(a)$ also provides the maximum gain in error correction performance for the overall LDPC decoder.

C. Frame Error Rate Performance

To prove the correct behavior of the proposed OMO T-EMS and OMO T-MM algorithms, we performed Frame Error Rate (FER) simulations for NB-LDPC codes with different degree distributions and Galois field values, from GF(4) to GF(32), assuming transmission over Binary Phase Shift Keying (BPSK) modulation and Additive White Gaussian Noise (AWGN) channel. In this subsection we only include the performance for two different codes, the (837,726) NB-LDPC code over GF(32) with $d_c=27$ and $d_v=4$ and the (2212,1896) NB-LDPC code over GF(4) with $d_c=28$ and $d_v=4$. For the rest of the codes we obtained similar results. We compare the proposed OMO T-EMS and OMO T-MM approaches to T-EMS [6] and Relaxed Min-Max (RMM) algorithms [10].

Fig. 4 shows the frame error rate (FER) simulation results of the (837,726) NB-LDPC code. For this code, the proposed OMO T-EMS algorithm in its floating point version (fp) achieves the same performance as T-EMS algorithm without any performance loss. Both algorithms use 15 iterations (it) and a scaling factor $\lambda=0.5$. In addition, the OMO T-MM algorithm has a coding gain of 0.12dB compared to the RMM from [10]. Comparing the quantized version of OMO T-EMS algorithm to RMM algorithm, OMO T-EMS algorithm with 7 bits (7b) for the datapath

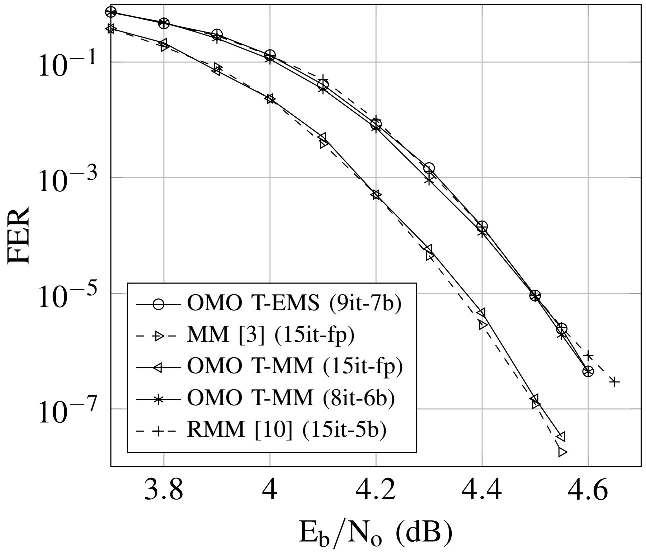

Fig. 4. FER performance for the (837,726) NB-LDPC code over GF(32) with AWGN channel. Layered schedule is applied to all algorithms. $\lambda = 0.5$ for TEMS and OMO T-EMS algorithms. $\gamma_p = 1.125$ for OMO T-EMS algorithm. $\gamma_p = 1.5$ for OMO T-MM algorithm.

and 9 iterations achieves the same performance as RMM [10] with 5 bits for the datapath and 15 iterations, so the proposed approach requires less iterations than the method from [10] to achieve the same performance. For the quantized version of the OMO T-MM the performance with 6 bits (6b) and 8 iterations achieves the same than the RMM decoder.

On Fig. 5, we have plotted the performance of the T-EMS decoder with 15 iterations, and for all the approximations of the second minimum discussed in this paper. The curve labeled T-EMS uses the exact computed value of min_2. As we can see, the fact that min_2'' and min_2''' do not estimate accurately the second minimum has indeed an impact on the overall decoder performance, and especially in the error floor region, where a strong early flattening appears for both approximations (especially for min_2''). On the other hand, our proposed approximation min_2^* has absolutely no performance loss compared to the T-EMS with the exact minimum computation, both in the waterfall and the error floor regime. It results that the complexity gains provided by the OMO-T-EMS comes at no performance loss, at least for the codes that we simulated.

In Fig. 6, simulations for the (2212,1896) NB-LDPC show a negligible performance loss of 0.03dB for a FER = 10^{-7} comparing T-EMS to OMO T-EMS. The same happens when we compare Min-max algorithm to OMO T-MM, just a negligible difference of 0.04dB is introduced

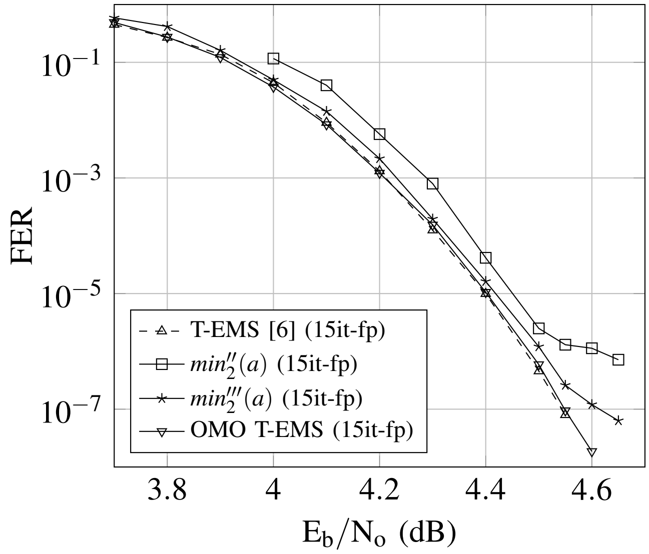

Fig. 5. FER performance for the (837,726) NB-LDPC code over GF(32) with AWGN channel for the estimators of the second minimum value. $\gamma_p = 1.125$ for OMO T-EMS algorithm.

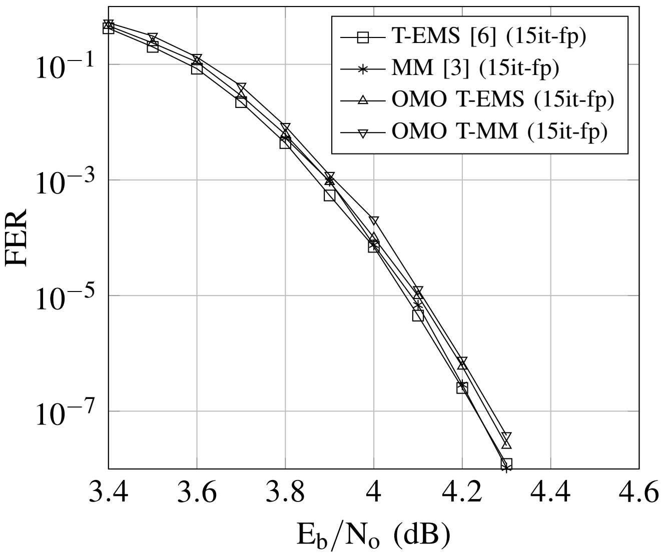

Fig. 6. FER performance for the (2212,1896) NB-LDPC code over $GF(4)$ with AWGN channel. Layered schedule is applied to all algorithms. $\lambda = 0.5$ for T-EMS and OMO T-EMS algorithms. $\gamma_p = 2.5$ for OMO T-EMS algorithm. $\lambda = 0.75$ for MM and OMO T-MM algorithms. $\gamma_p = 1.125$ for OMO T-MM algorithm.

by the approximation. The γ_p values in all simulations are adjusted using the mean of the ratio min_2/min_1 as we said before.

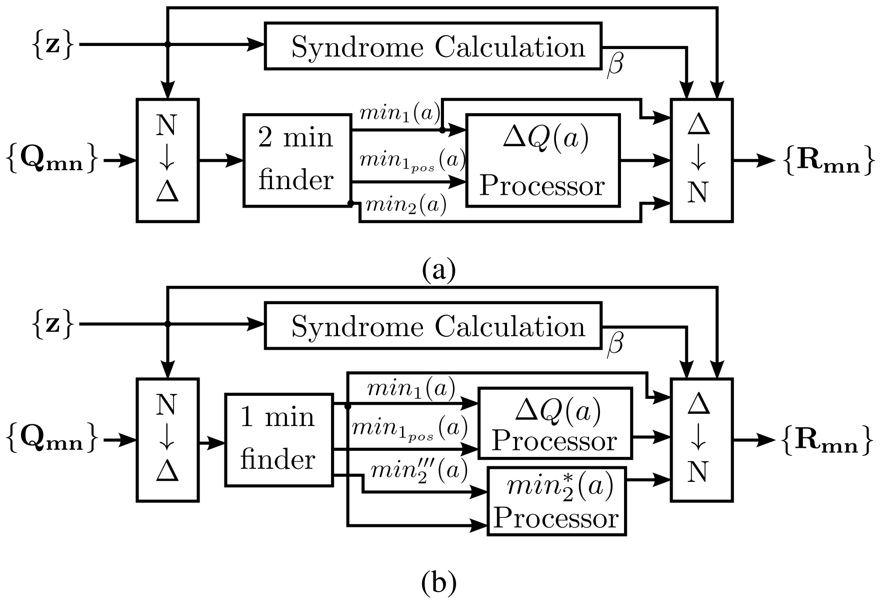

Fig. 7. Check node top architecture for T-EMS algorithm (a). Proposed OMO T-EMS/ OMO T-MM check node architecture (b).

IV. OMO T-EMS and OMO T-MM Hardware Architectures

In this section the hardware architectures for the proposed OMO T-EMS and OMO T-MM are introduced. Since the main contribution of this paper focuses in the CN processing, first we detail the implementation results for the OMO T-EMS and OMO T-MM CN architectures comparing them to other existing solutions. Finally, we present the results for the complete decoders with horizontal layered schedule.

A. Check Node Architecture

In Fig. 7.a the original T-EMS hardware structure [6] is included, while the proposed OMO T-EMS CN structure is presented in Fig. 7.b. It can be observed that the main advantage of our approach is to avoid the use of two-minimum finders and apply one-minimum finders, reducing the total complexity and the delay for the $min_1(a)$ values, introducing the novel second-minimum estimation. To do this approximation, the block labeled "$min_2^*(a)$ *Processor*" is responsible to implement the Eq. (2). The $min_2^*(a)$ Processor does not introduce any additional delay since the processing is made at the same time that the $\Delta Q(a)$ values are computed. Moreover, it is important to remark that the OMO decoding technique can be directly implemented for a Min-max decoder obtaining the same advantages.

For both OMO T-EMS and OMO T-MM algorithms, the row-wise search of the most reliable messages $min_1(a)$ implies that the one-minimum finder must have d_c inputs and includes an extra multiplexor in the last stage to extract the $min_2'''(a)$ values as shown in Fig. 3. For the CN $q-1$ one-minimum finders are required, each one formed by (d_c-1) w-bit comparators and $(d_c \times w)$ 2-input multiplexors, where w is the number of bits for the datapath. On the other hand, compared with the two-minimum finders [11], the critical path is reduced by half due to the reduction of the hardware spent on the second minimum computation, which will impact greatly on the obtained throughput.

To make a fair comparison with the two-minimum finder used in conventional designs, we must add extra resources to implement (2), which reduce to $2 \times (q-1)$ w-bit adders for the one-minimum finders. This value is calculated considering that the implementation of (1) and (2) need only two additional adders.

On the other hand, a conventional two-minimum finder [11] requires $2 \times d_c$ w-bit comparators and $3 \times d_c \times w$ 2-input multiplexors. Considering the same number of equivalent gates for an adder and a comparator (w bits both of them), the two-minimum finder has three times more multiplexors and two times more comparators than the one minimum finder plus the second minimum estimation implementing (2).

For the $N \rightarrow \Delta$ and $\Delta \rightarrow N$ transformation, the approach used is similar to the one proposed in [13], requiring $2 \times q \times p \times d_c \times w$ 2-input multiplexors to perform both transformations. The check node's syndrome β is calculated adding all d_c tentative hard decision symbols z_n by means of a GF(q) adder in a tree structure fashion needing $p \times (dc-1)$ XOR gates.

The extra column values are generated using a configuration processor similar to the one proposed in [6] using $(q-1) \times (q/2-1)$ w-bit adders or comparators to generate the tentative extra column values depending on whether EMS or Min-max check node is implemented. To select the most reliable value, $q-1$ one minimum finders are required, each one formed by $(q/2-1)$ w-bit comparators and $(q/2 \times w)$ 2-input multiplexors. To compute the path info, $(q-1)(2 \times \lceil \log d_c \rceil + (q/2-1) \times w)$ 2-input multiplexors, $(q-1)(\lceil \log d_c \rceil)$ XOR gates and $(q-1)(\lceil \log d_c \rceil - 1)$ OR gates are implemented.

The resources required for the CN implementation of OMO T-EMS and OMO T-MM are summarized in Table II and compared with the approaches from [10] and [6]. VHDL was used for the description of the hardware and the total gate account was derived after synthesis using Cadence RTL Compiler. The hardware implementation was performed for the (837,726) NB-

TABLE II

CN COMPLEXITY COMPARISONS. FOR THE (837,726) NB-LDPC CODE OVER GF(32)

Architecture	Datapath	Logic Gates (NAND)	Memory (bits)
Relaxed Min-Max [10]	5 bits	152594	52080
T-EMS ($n_r = 2$, $n_c = 2$) [6]	6 bits	304260	-
OMO T-EMS ($n_r = 1$, $n_c = 2$)	7 bits	190780	-
OMO T-MM ($n_r = 1$, $n_c = 2$)	6 bits	165700	-

LDPC code over $GF(32)$, with $d_c = 27$ and $d_v = 4$.

As it can be observed, although the CN in [10] has less NAND gates than our proposals, their CN requires to store intermediate messages due to the serial processing, increasing the gate account of the CN to 230714 NAND gates (considering that storing one bit of RAM memory is equivalent in terms of area to 1.5 NAND gates [10], [14], [15]). Hence, our proposals requires at most 18% less logical resources than the CN presented in [10], even considering that we use two extra bits.

For T-EMS decoder presented in [6] we did not provide separate results for the CN architecture, however we obtained these results considering the main differences with our new proposal. The CN from [6] needs about four times more hardware than our OMO approaches for the extra column values computation due to the use of the first and second minimum for the extraction of the extra column values. As we can see in Table II OMO T-EMS and OMO T-MM require 37% less logical resources than [6].

B. Complete decoder architecture

The proposed CN architecture has been included in an horizontal layered schedule decoder (with one CN cell (Fig. 7) and d_c VN units. Each VN processor includes dual-port memories that store the LLR values ($Q_n(a)$) and avoid adding extra latency. On the other hand, due to the layered schedule, a shift register is required to store the “last iteration” CN output information ($R_{m,n}(a)$).

Since, only one CN cell is implemented in the decoder, M clock cycles are required to complete one decoding iteration. This value is increased due to the pipeline stages (k) introduced in the decoder ($k \times d_v$ clock cycles are added) with the aim of achieving the desired clock frequency (f_{clk}). As after processing one entire circulant sub-matrix the pipeline registers must be empty before processing a new one, reducing the logical path of the decoder has a great impact in the maximum throughput achieved by the decoder (Eq. (3)). Finally, $q-1$ additional clock cycles are required to load the LLR values and output the estimated codeword of the decoder.

$$Throughput = \frac{f_{clk}[\text{MHz}] \times N \times p}{it \times (M + d_v \times k) + (q-1)} \left[\frac{\text{Mb}}{\text{s}}\right] \tag{3}$$

With OMO T-EMS we reduce the critical path of the CN, so we only require 8 pipeline stages to achieve a clock frequency of $f_{clk} = 250MHz$ after place and route with Cadence SOC encounter tools and employing a 90 nm CMOS library in which the area of a NAND gate is $3.13\mu m^2$. The total area of the decoder is 19.02 mm^2 with a core occupation of 60% and a gate account of $(19.02 \times 0.6)/3.13 = 3.6$ Million of NAND gates.

For OMO T-MM the number of pipeline stages is 8 and the maximum clock frequency is $f_{clk} = 250MHz$. The total area is 16.10 mm^2 with a core occupation of 70% and a gate account of $(16.1 \times 0.7)/3.13 = 3.6$ Million NAND gates.

It is important to remark that the library used to implement both OMO T-EMS and OMO T-MM do not include optimized RAM memories, so each bit of RAM is implemented as a register, and hence, the area for the memories is about three times larger. Due to this, the total number of equivalent NAND gates is overestimated compared to the results found in literature that always assume optimized memories. For this reason we include in Table III, for comparison purposes, the equivalent number of NAND gates assuming that each bit of RAM is implemented with and area of 1.5 NAND gates.

To achieve the same performance as in [10] and [6], our OMO T-EMS approach requires only 9 decoding iterations, as can be seen in Fig. 4, therefore the total latency of the decoder is 1435 clock cycles, which corresponds to a throughput of 729 Mbps (3). For the OMO T-MM 1279 clock cycles are required to get the same FER performance as RMM or T-EMS, so a maximum throughput of 818 Mbps is reached.

OMO T-EMS and OMO T-MM decoders have been compared to the most efficient NB-LDPC decoder designs to the best knowledge of the authors. The results of the comparisons have been included in Table III, where we have scaled the results in [10] to include all throughput results over 90 nm CMOS process [16].

The throughput of both OMO T-EMS and OMO T-MM decoders is higher than any decoder proposed in literature for high order fields and high rate NB-LDPC codes (see Table III), because of the improvements made at the check node processor.

Our approaches require in the worst case (OMO T-EMS) less than half area than [15] and achieve at least 3.2 times higher throughput, so our most complex solution is 13 times more efficient. [2]

On the other hand, the decoder presented in [10] has been considered since it was the most efficient one until now and it uses (1) as a method to approximate the second minimum, which gives benefits in terms of area but introduces early performance degradation (Fig. 4). Despite this, OMO T-EMS has 8.8 times less latency than [10] achieving 4.75 times higher throughput with a decoder 49.7% more efficient in terms of area over throughput (for a 90 nm CMOS process). On the other hand, OMO TMM has 61.2% higher efficiency than [10] with 9.9 times less latency and 5.3 times higher throughput.

Finally, our OMO T-EMS approach has been compared against the T-EMS decoder presented in [6]. Making use of the novel approach for the second minimum estimation, the latency is reduced on 33% with respect to [6] with an increment in throughput of 50%. The area was also reduced in 25%, so the efficiency is 50% higher.

Is important to remark that the proposed approach is focused on high-rate NB-LDPC codes. However, efficient NB-LDPC decoders suitable for lower rate codes have been proposed in the literature [17]- [18]. These architectures make a parallel processing of messages.

V. Conclusions

In this paper a new method to estimate the second minimum value in message of the check node processor of NB-LDPC decoders is proposed. This solution avoids the use of two-minimum finders, greatly reducing the check node complexity. The simplifications applied to the T-EMS and T-MM algorithms reduce latency and area with respect to the original proposal, without

[2] Note that [15] is the only proposal that also provides post place and route results.

TABLE III

Comparison of the proposed NB-LDPC layered decoders to other works from literature. For the (837,726) NB-LDPC code over GF(32)

Algorithm	T-Max-log QSPA [15]	RMM [10]	T-EMS [6]	OMO T-EMS / OMO T-MM
Report	Post-layout	Synthesis	Synthesis	Post-layout
Technology	90 nm	180 nm	90 nm	90 nm
Quantization (w)	7 bits	5 bits	6 bits	7 bits / 6 bits
Gate Count (NAND)	8.51M	871K	2.75M	2.07M / 1.79M
f_{clk} (MHz)	250	200	250	250
Iterations	5	15	12	8
Latency (clock cycles)	4460	12675	2160	1435 / 1279
Throughput (Mbps)	223	66	484	729 / 818
Throughput (Mbps) 90 nm	223	154	484	729 / 818
Efficiency 90 nm (Mbps/M-gates)	26.2	176.8	176	352 / 456
Area (mm^2)	46.18	-	-	19.02 / 16.10

introducing any significant performance loss. The proposed check node architecture was included in a complete decoder with layered schedule achieving 729 Mbps of throughput after place and route on a 90nm CMOS process for OMO T-EMS and 818 Mbps for OMO T-MM. The designed decoder nearly doubles the efficiency of the best solutions found in literature for high order fields and high rate codes.

Acknowledgment

This work was supported in part by the Spanish Ministerio de Ciencia e Innovación under Grant TEC2011-27916 and in part by the Universitat Politècnica de València under Grant PAID-06-2012-SP20120625. The work of F. García-Herrero was supported by the Spanish Ministerio de Educación under Grant AP2010-5178. David Declercq has been funded by the Institut Universitaire de France for this project.

References

[1] M. Davey and D. MacKay, “Low-density parity check codes over GF(q),” *IEEE Communications Letters*, vol. 2, no. 6, pp. 165–167, 1998.

[2] D. Declercq and M. Fossorier, “Decoding Algorithms for Nonbinary LDPC Codes Over GF(q),” *IEEE Transactions on Communications*, vol. 55, no. 4, pp. 633–643, 2007.

[3] V. Savin, “Min-Max decoding for non binary LDPC codes,” in *IEEE International Symposium on Information Theory*, 2008, pp. 960–964.

[4] E. Li, K. Gunnam, and D. Declercq, “Trellis based Extended Min-Sum for decoding nonbinary LDPC codes,” in *8th International Symposium on Wireless Communication Systems (ISWCS)*, 2011, pp. 46–50.

[5] E. Li, D. Declercq, and K. Gunnam, “Trellis-Based Extended Min-Sum Algorithm for Non-Binary LDPC Codes and its Hardware Structure,” *IEEE Transactions on Communications*, vol. 61, no. 7, pp. 2600–2611, 2013.

[6] E. Li, D. Declercq, K. Gunnam, F. García-Herrero, J. Lacruz, and J. Valls, “Low Latency T-EMS Decoder for NB-LDPC Codes,” in *Conference Record of the Forty Seventh Asilomar Conference on Signals, Systems and Computers (ASILOMAR), 2013*, 2013.

[7] X. Zhang and F. Cai, “Reduced-latency scheduling scheme for min-max non-binary LDPC decoding,” in *IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), 2010*, Dec 2010, pp. 414–417.

[8] ——, “Reduced-Complexity Decoder Architecture for Non-Binary LDPC Codes,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 7, pp. 1229–1238, July 2011.

[9] M. Punekar and M. Flanagan, “Trellis-based check node processing for low-complexity nonbinary LP decoding,” in *IEEE International Symposium on Information Theory Proceedings (ISIT), 2011*, July 2011, pp. 1653–1657.

[10] F. Cai and X. Zhang, “Relaxed Min-Max Decoder Architectures for Nonbinary Low-Density Parity-Check Codes,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. PP, no. 99, pp. 1–1, 2012.

[11] C.-L. Wey, M.-D. Shieh, and S.-Y. Lin, “Algorithms of Finding the First Two Minimum Values and Their Hardware Implementation,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 11, pp. 3430–3437, 2008.

[12] B. Zhou, J. Kang, S. Song, S. Lin, K. Abdel-Ghaffar, and M. Xu, “Construction of non-binary quasi-cyclic LDPC codes by arrays and array dispersions - [transactions papers],” *IEEE Transactions on Communications*, vol. 57, no. 6, pp. 1652–1662, 2009.

[13] J. Lin, J. Sha, Z. Wang, and L. Li, “Efficient Decoder Design for Nonbinary Quasicyclic LDPC Codes,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 5, pp. 1071–1082, 2010.

[14] X. Chen and C.-L. Wang, “High-Throughput Efficient Non-Binary LDPC Decoder Based on the Simplified Min-Sum Algorithm,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 11, pp. 2784 –2794, nov. 2012.

[15] Y.-L. Ueng, K.-H. Liao, H.-C. Chou, and C.-J. Yang, “A High-Throughput Trellis-Based Layered Decoding Architecture for Non-Binary LDPC Codes Using Max-Log-QSPA,” *IEEE Transactions on Signal Processing*, vol. 61, no. 11, pp. 2940–2951, 2013.

[16] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital integrated circuits: a design perspective*. Pearson Education, 2003.

[17] J. Lin and Z. Yan, “An Efficient Fully Parallel Decoder Architecture for Nonbinary LDPC Codes,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. PP, no. 99, pp. 1–1, 2013.

[18] Y. S. Park, Y. Tao, and Z. Zhang, “A 1.15Gb/s fully parallel nonbinary LDPC decoder with fine-grained dynamic clock gating,” in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013*, Feb 2013, pp. 422–423.