

# Low-complexity Multi-Bit Iterative Decoders: Algorithms and Hardware Architectures

Bane Vasic  
University of Arizona  
vasic@ece.arizona.edu

Shiva Planjery, and David Declercq  
Université de Cergy-Pontoise-CNRS  
{shiva.planjery,declercq}@ensea.fr

**Abstract**—We present a new type of iterative decoders for low-density parity check (LDPC) codes that use simple Boolean functions for variable node processing and surpass the belief propagation decoders in the error floor region. We discuss the error performance, guaranteed error correction capability and speed and implementation complexity of these decoders as factors crucial for application that require extremely high speeds. We also present a hardware architecture and show that our decoders require only half an silicon area to achieve the same throughput as best efficient min-sum decoders, while improving the frame error performance in the error floor region.

## I. SUMMARY

Iterative message-passing decoding algorithms and low-density parity-check (LDPC) codes of small column-weights ( $d_c \leq 5$ ) offer a good error-correction capability, and are standardized in numerous communications systems. However, the complexity of the existing algorithms is still high and the decoding speed is limited which ignited a significant effort in searching for alternatives which offer better trade-offs among complexity, decoding speed and error performance.

In a series of recent papers [2], [3] we introduced a new type of decoders, called finite alphabet iterative decoders (FAIDs) for LDPC codes. In these decoders, the messages take values in a small finite alphabet and the variable-to-check messages are updated using the check-to-variable messages and the value received from a channel through a Boolean function judiciously designed to improve the error floor performance over binary symmetric channel. We have shown that a FAID outperforms a floating-point BP decoder in the error-floor region with only seven levels in the alphabets (3-bit word message length). In addition, a multiplicity of decoders with different map functions can be used in parallel in serial to further improve the performance at the cost of higher complexity [3].

In the our talk we will present this novel type of iterative decoders designed for discrete memoryless channels with finite alphabets. In addition to low complexity and excellent error floor performance, these decoders are amenable to analysis and unlike the existing message passing decoders they offer error correction guaranties, a feature highly desirable in high speed

This work is supported in part by the National Science Foundation under Grants CCF-0963726 and CCF-1314147 and Seventh Framework Programme of the European Union, under Grant Agreement number 309129. We also acknowledge earlier contributions of Xinmiao Zhang and Fang Cai in hardware emulation.

applications. We present the performance analysis of a new class of iterative algorithms for LDPC codes on the BSC and discuss their hardware implementation and complexity. This part of the presentation will be based on the recent invited paper [4].

## II. FAID DECODERS

An  $N_s$ -level FAID operates on the finite message alphabet  $\mathcal{M} = \{-L_s, \dots, -L_2, -L_1, 0, L_1, L_2, \dots, L_s\}$ , where  $L_i \in \mathbb{R}^+$  (set of  $N_s = 2s+1$  increasing positive message levels) and  $\mathcal{Y}$ , the set of channel values. The sign of a message  $x \in \mathcal{M}$  can be interpreted as an estimate of the bit associated with the variable node to or from which  $x$  is being passed (positive for zero and negative for one), and the magnitude  $|x|$  as a measure of how reliable this value is, and for the BSC,  $\mathcal{Y} = \{\pm C\}$ , where  $C \in \mathbb{R}^+$ , and the value  $y_i \in \mathcal{Y}$  for node  $v_i$  is determined by  $y_i = (-1)^{r_i} C$ , i.e., we use the mapping  $0 \rightarrow C$  and  $1 \rightarrow -C$ .

Let  $m_1, \dots, m_{l-1}$  denote the extrinsic incoming messages to a node with degree  $l$ . The function  $\Phi_c : \mathcal{M}^{d_c-1} \rightarrow \mathcal{M}$  used for updating a check node with degree  $d_c$  is classical min-sum function.

The function  $\Phi_v : \mathcal{Y} \times \mathcal{M}^{d_v-1} \rightarrow \mathcal{M}$  used for update at a variable node with degree  $d_v$  is defined as

$$\Phi_v(y_i, m_1, m_2, \dots, m_{d_v-1}) = Q \left( \sum_{j=1}^{d_v-1} m_j + \omega_i \cdot y_i \right) \quad (1)$$

where the function  $Q(\cdot)$  is defined below based on a threshold set  $\mathcal{T} = \{T_i : 1 \leq i \leq s+1\}$  with  $T_i \in \mathbb{R}^+$  and  $T_i > T_j$  if  $i > j$ , and  $T_{s+1} = \infty$ .

$$Q(x) = \begin{cases} \text{sgn}(x)L_i, & \text{if } T_i \leq |x| < T_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

The weight  $\omega_i$  is computed using a symmetric function  $\Omega : \mathcal{M}^{d_v-1} \rightarrow \mathbb{R}^{\geq 0}$  (set of non-negative real numbers) whose input arguments are the  $d_v - 1$  incoming messages.

On the other hand, the function  $\Phi'_v$  used in the min-sum decoders is

$$\Phi'_v(y_i, m_1, \dots, m_{d_v-1}) = y_i + \sum_{j=1}^{d_v-1} m_j. \quad (2)$$

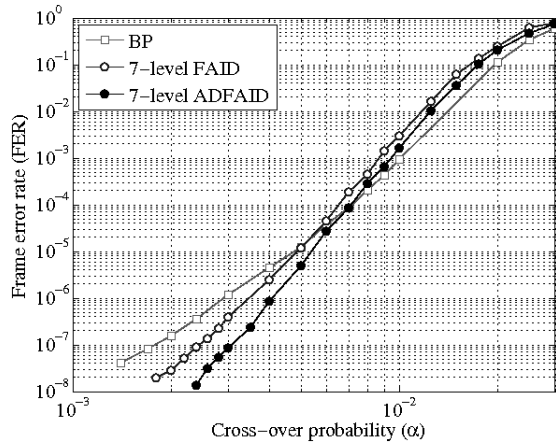


Fig. 1. FER comparison on the (732, 551) structured code with  $d_{min} = 12$

For both FAID and min-sum decoders, the hard-decision of the bit  $i$  is calculated as the sign of  $y_i + \sum_{j=1}^{d_v} m_j$  in each decoding iteration.

### III. GUARANTEED ERROR CORRECTION CAPABILITY

Deriving provable statements in terms of guaranteed error correction of iterative decoders still remains a challenge. We will present the technique of *decimation*, for analysis of FAIDs. It involves fixing certain bits of the codeword to a particular value, and incorporating this into the variable node update function of FAIDs in different novel ways, thus constituting two new classes of decoders, namely, *decimation-enhanced FAIDs* and *adaptive decimation-enhanced FAIDs* for LDPC codes on the BSC. We then show that decimation leads to powerful 3-bit precision FAID that reduces the number of iterations required to correct a fixed number of errors while maintaining the good performance of the original FAID. With the help of this scheme, we provide insights into how the decoders can be analyzed to derive provable results related to their guaranteed error correction capability. Fig. 1 shows an example of the performance improvement on a QC code with  $d_{min} = 12$ .

### IV. VLSI ARCHITECTURES AND COMPLEXITY

In [4] we developed an efficient VLSI implementation architecture for QC-LDPC codes. Bit serial check node processing is adopted and shown to result in higher efficiency. In [4] we also introduced a new bit-serial check node unit architecture based on the design in [5]. The check node unit in [5] does not compute second minimum and uses minimum+1 instead which may lead to performance loss and higher error floor. In [4], both minimum and second minimum are computed at the expense of small area overhead. We also showed that the variable node update function can be implemented efficiently due to the symmetry of the Boolean function. An optimized interleaved data scheduling scheme is also developed to maximize the hardware utilization efficiency.

We focus on regular quasi-cyclic (QC) LDPC codes with column-weight  $d_v = 3$ . The parity check matrix of a QC-LDPC code is an array of  $r \times t$  sub-matrices of dimension  $L \times L$ . Each sub-matrix is either a cyclically shifted identity matrix or a zero matrix which greatly reduces hardware complexity. As an illustration, we present the synthesis results for the our decoders operating on a (7807, 7177) QC-LDPC code with  $L = 211$ ,  $r = d_c = 37$ , and  $t = d_v = 3$  and comparison with the Min-sum decoder in [5], which is among the most efficient existing designs. Our decoder outperforms the min-sum decoder with 5-bit word length in the error floor region for the 3-bit message length. The proposed check and variable node units with  $w = 3$  are synthesized using SMIC 180nm CMOS technology and compared with those with  $w = 5$  for the min-sum decoder from [5].

sudden increase in the area is listed as the maximum achievable clock frequency in these tables.

TABLE I  
SYNTHESIS RESULTS OF (7807, 7177) QC-LDPC DECODERS USING 180nm CMOS TECHNOLOGY [4]

	Min-sum [5] ( $w = 5$ )	Proposed FAID ( $w = 3$ )
Area ( $mm^2$ ) (normalized)	69.3 (1)	36.1 (0.52)
Max. freq. (Mhz)	384	384
Latency (# clks)	180	180
Throughput (Gbps) (15 iter.)	33.3	33.3

The synthesis results of the overall (7807, 7177) QC-LDPC decoders are listed in Table I. Compared to the Min-sum decoder in [5], the number of VNUs needed in our FAID design is reduced to less than 1/4, and each VNU is smaller. Moreover, the registers for storing the v-to-c messages, Min1, and Min2 in the FAID is less since the word length is shorter.

The critical paths of both decoders lie in the check node units and they are the same. Also both decoders require the same number of clock cycles in each decoding iteration. Hence, their achievable throughputs are the same. Assuming that 15 decoding iterations are carried out, the latency of both decoders is  $15 \times (2 \times 6) = 180$  clock cycles. Considering that two data blocks are interleaved, the achievable throughput is  $(2 \times 7807) \times 384 / 180 = 33.3$  Gbps. The proposed FAID needs only 52% the area of the min-sum decoder in [5].

### REFERENCES

- [1] S. Planjery, D. Declercq, L. Danjean, and B. Vasic, "Finite alphabet iterative decoders for LDPC codes surpassing floating-point iterative decoders," *Electronics Letters*, vol. 47, no. 16, pp. 919-921, Aug. 2011.
- [2] S. K. Planjery, D. Declercq, L. Danjean, and B. Vasic, "Finite alphabet iterative decoders, Part I: Decoding beyond belief propagation on the binary symmetric channel," *accepted to IEEE Trans. Commun.*, Aug. 2013.
- [3] D. Declercq, B. Vasic, S. K. Planjery, E. Li, "Finite alphabet iterative decoders, Part II: Improved guaranteed error correction of LDPC codes via iterative decoder diversity," *accepted to IEEE Trans. Commun.*, Aug. 2013.
- [4] F. Cai, X. Zhang, D. Declercq, B. Vasic, and S. K. Planjery, "Low-complexity finite alphabet iterative decoders for LDPC codes," *IEEE Transactions on Circuits and Systems - Part I: Regular Papers*, submitted.
- [5] A. Darabiha, A. C. Carusone, and F. R. Kschischang, "A bit-serial approximate min-sum LDPC decoder and FPGA implementation," *Proc. IEEE Intl. Symp. Circuits and Syst.*, May 2006.