# TRACKING FORECAST MEMORIES IN STOCHASTIC DECODERS

Saeed Sharifi Tehrani, Ali Naderi, Guy-Armand Kamendje, Shie Mannor, and Warren J. Gross

Department of Electrical and Computer Engineering

McGill University, Montreal, Quebec, Canada

e-mail:{sshari9,shie,wjgross}@ece.mcgill.ca, ali.naderi@mail.mcgill.ca, gkamendje@cim.mcgill.ca

# ABSTRACT

This paper proposes Tracking Forecast Memories (TFMs) as a novel method for implementing re-randomization and decorrelation of stochastic bit streams in stochastic channel decoders. We show that TFMs are able to achieve decoding performance similar to that of the previous methods in the literature (i.e., edge memories or EMs), but they exhibit much lower hardware complexity. TFMs significantly reduce the area requirements of ASIC implementations of stochastic decoders.

*Index Terms*— Iterative (channel) decoding, stochastic decoding, low-density parity-check codes, ASIC.

#### 1. INTRODUCTION

LDPC codes are one of the most powerful classes of errorcorrecting codes known to date. These codes are able to provide decoding performance approaching the Shannon capacity limit [1] and have been considered for recent digital communication standards including IEEE 802.3an (10GBASE-T), IEEE 802.16 (WiMAX), and IEEE 802.11n (WiFi). LDPC codes can be iteratively decoded using the well-known Sum-Product Algorithm (SPA) or its approximation, the min-sum algorithm. The SPA is a message passing algorithm in which two types of processing nodes, known as variable nodes (VNs) and parity-check nodes (PNs), iteratively exchange their beliefs (in the form of probability messages) about the correctness of the received symbols from the channel. The connections between VNs and PNs are defined by the parity-check matrix of the code. This can be also graphically described by a bipartite factor graph [2] (see Fig. 1). Despite the excellent performance of LDPC codes under the SPA, the hardware complexity of SPA is high and hence low-complexity LDPC decoding has been a focal point of research in recent years.

Stochastic decoding [3] is a newly proposed approach for low-complexity iterative decoding of error-correcting codes on graphs. In this approach, probability messages are encoded into streams of bits using Bernoulli sequences in a way that the likelihood of observing a '1' in a stream is equal to the encoded probability. For example, for an (encoded) probability of 0.9, the likelihood of observing a '1' is 0.9 and the likelihood of observing a '0' in the stream is 0.1. The stochastic representation results in simple hardware structure for VNs and PNs. More importantly, because of its bit-serial nature, stochastic representation significantly reduces the number of wires between VNs and PNs. These advantages are key in the case of fully-parallel LDPC decoders where the number of processing nodes is high and the routing congestion (caused by complex interconnections between nodes) is a major problem (see [4–6]).

Stochastic decoding is, however, prone to the latching problem [7, 8]. It has been observed that cycles in a code graph correlate stochastic messages in a way that a group of nodes stick into fixed states for several decoding iterations and dramatically corrupt the convergence of the decoder (see [8]). The latching problem prevented early stochastic methods from being applied to practical LDPC codes. To overcome this problem, the concept of edge memories (EMs) was introduced in [8]. EMs are assigned to outgoing edges of VNs in a factor graph (see Fig. 1) and are used to rerandomize and hence decorrelate stochastic streams. EMs significantly reduce the chance of latching in the decoder enabled demonstrations of stochastic LDPC decoding of practical LDPC codes. The potential of EM-based stochastic decoding for high-throughput LDPC decoder hardware implementations was recently validated in [6]. The stochastic LDPC decoder in [6] is able to provide a maximum throughput of 1.66 Gb/s. The EM-based stochastic decoding method was also extended to the important classes of BCH, Reed-Solomon and Turbo product codes in [9].

In this paper we propose replacing EMs with TFMs in stochastic decoders. We provide examples of stochastic decoders that decode a (1056,528) LDPC code chosen from the IEEE 802.16 (WiMAX) standard and show that TFMs can provide similar decoding performance to EMs while having much lower hardware complexity. We also investigate the impact of TFMs on the overall area of stochastic decoders. The remainder of this paper is organized as follows. Section 2 provides background on stochastic LDPC decoding. Section 3 introduces TFMs. Section 4 compares the decoding performance and hardware complexity of TFMs with EMs. Section 5 presents the conclusions.



**Fig. 1**. Block diagram of a stochastic decoder for a (n, k) LDPC code based on the code's factor graph. In stochastic decoding, EMs are assigned to outgoing edges of VNs and are depicted as gray rectangles.

### 2. REVIEW OF STOCHASTIC LDPC DECODING

This section provides a brief review of stochastic LDPC decoding. The reader may refer to [6, 8, 10] for details.

Fig. 1 shows the block diagram of a stochastic decoder and Fig. 2 show the structure of a stochastic VN. In stochastic decoding channel probabilities are converted to streams of stochastic bits. Each VN receives one bit (i.e., CH<sub>in</sub> in Fig. 2) in every decoding cycle. A VN has two modes of operation: the *hold state* and the *regular (non-hold) state*. A hold state refers to the case where the input bits of a VN disagree and the VN refers to its memory to generate the output bit for an edge. The regular state refers to the case where there is an agreement between input bits and the VN directly copies the input bit to the output bit. Note that a stochastic VN uses one EM per edge and hence a degree- $d_v$  VN uses  $d_v$  EMs. When a VN is in the regular state, it directly uses the newly generated bit (called a regenerative bit [6]) as the output bit. When a hold state occurs the VN refers to the corresponding EM for the edge and (pseudo) randomly picks a bit from it. Stochastic decoding continues by VNs and PNs sending and receiving stochastic bits, one bit in each direction on every edge during each decoding cycle (DC). A stochastic PN checks if the incoming bits from VNs have an even parity. A stochastic PN has a simple structure based on XOR gates as shown in Fig. 3.

The operation of EMs are essential for proper stochastic decoding. An EM can be implemented as a shift-register with a single selectable bit. In this respect, in regular states an EM is updated with regenerative bits from the VN. In the hold state, a bit is (pseudo) randomly chosen from EM. This is done by generating a (pseudo) random address that changes in each decoding cycle (the RA signals in Fig. 2). As shown in [10] and [6], the random addresses for EMs in a stochastic decoder can be significantly shared and hence the decoder can use a simple *randomization engine* (based on short LFSRs) to produce the required random numbers for the whole decoder.



Fig. 2. The structure of a degree-2 VN based on EMs.  $A_{in}$  and  $B_{in}$  are received bits from PNs and  $CH_{in}$  is a received bit from the channel. Similarly,  $A_{out}$  and  $B_{out}$  are output bits for PNs and  $CH_{out}$  is the output bit of the VN.



Fig. 3. The structure of a degree-4 PN.

## 3. TRACKING FORECAST MEMORIES

Despite their good decoding performance, EMs occupy considerable silicon area when implemented in ASICs (see Section 4.2). As shown in Fig. 1, EMs are assigned to each outgoing edge of VNs. Consequently, the complexity of EMs significantly affects the area requirements of ASIC stochastic decoders because for decoding a state-of-the-art LDPC code (e.g., with n > 1000) the decoder uses thousands of EMs. In this respect, less-complex solutions that can provide similar decoding performance are important.

Our proposed method, called Tracking Forecast Memories (TFMs), addresses the above-mentioned problem. TFMs replace EMs in VNs. They measure the probabilities of outgoing stochastic streams from VNs and update them based on the method of *successive relaxation*. Let  $b(t) \in \{0, 1\}$ be the outgoing regenerative bit from a VN and P(t) be the probability stored by the TFM for the corresponding stochastic stream ( $0 \le P(t) \le 1$ ). The TFM recursively updates the P(t) probability in regular states as follows:

$$P(t+1) = (1 - \beta(t)) P(t) + \beta(t)b(t),$$
(1)



Fig. 4. General block diagram of a TFM. The multiplication can be replaced by a shift of the bit wires when  $\beta(t)$  is chosen as a negative power of 2.

where  $0 < \beta(t) < 1$  is the relaxation coefficient. The relaxation mechanism implemented in (1) efficiently estimates P(t) and follows its trend if P(t) changes. Also, similar to a low-pass filter, it filters noise. This memory-based mechanism is named TFM since it is based on tracking the past observations while emphasizing recent outcomes.

The operation of a TFM-based VN is similar to an EMbased VN. When the hold state occurs for the VN, the TFM is not updated, instead, P(t) is compared against a (pseudo) random number (probability), R(t), and the new outgoing bit, b'(t), is generated as follows: b'(t) = 1 when  $P(t) \ge R(t)$ and b'(t) = 0 when P(t) < R(t). In regular states, the TFM is updated as in (1) and b'(t) = b(t).

Fig 4 shows the block diagram of a TFM. In this figure, the U signal determines the regular states (U = 1) or hold states (U = 0). Also, R(t) is a (pseudo) random number varying in every decoding cycle. Note that for the sake of lower hardware complexity, the multiplication involved in the TFM operation can be replaced by a shift of the bit wires, when  $\beta(t)$  is chosen as a negative power of 2. For example, for the simulation results in this paper we used a constant relaxation coefficient of  $\beta(t) = 1/16$ .

### 4. DECODING PERFORMANCE AND HARDWARE COMPLEXITY

#### 4.1. Bit-Error-Rate Decoding Performance

Fig. 5 shows the bit-error-rate (BER) decoding performance of the proposed method for decoding a (1056,528) irregular LDPC code chosen from IEEE 802.11n (WiMAX) standard. For the sake of comparison, this figure also shows the decoding performance of the SPA and the EM-based stochastic decoder from [6] which decodes the same code and uses 64-bit, 48-bit, and 32-bit EMs for degree-6, degree-3 and degree-2 VNs, respectively. Note that for the case of "ideal" stochastic decoding, floating-point implementation is used and random numbers in the decoder are not shared. For true-bit simulations, fixed-point implementation of TFMs is considered and



**Fig. 5**. Decoding performance results for a (1056,528) LDPC code (FP:floating-point, FX:fixed-point, DC:decoding cycle).

the random numbers are generated by the same randomization engine used in the stochastic decoder in [6], also, received symbols from the channel are quantized to six bits as in [6].

As shown in Fig. 5, the floating-point TFM outperforms the EM-based decoder at high BERs (low SNRs) and provides performance within 0.5 and 0.25 dB of the floating-point SPA with 32 and 16 iterations, respectively. The decoder with 9bit fixed-point TFMs is able to provide similar performance compared to the EM-based decoder in [6]. In addition, it is possible to provide similar decoding performance with 8-bit fixed-point TFMs at low BERs (where practical LDPC decoders are expected to be used). The decoder with 7-bit fixedpoint TFMs has about 0.2 dB loss at low BERs (high SNRs) compared to the EM-based decoder. In summary, it can be concluded that 8 or 9-bit fixed-point TFMs are sufficient to provide similar performance as EMs for stochastic decoders.

### 4.2. Hardware Complexity Comparison

To compare the complexity of TFMs versus EMs, we synthesized TFM and EM circuits in the ST Microelectronics 90 nm 1V CMOS technology by using the Cadence RTL-Compiler v6.1 synthesis tool. Table 1 shows the area, delay and the equivalent 2-input (NAND) gate count of the synthesized circuits. As shown, TFMs provide a significant area improvement compared to EMs. For example, an 8-bit TFM requires about 63% and 30% less area compared to 64-bit and 32bit EMs, respectively. The impact of the area-efficiency of TFMs on the overall area of VNs is also significant. To show this impact, Table 1 also compares a degree-6 TFM-based VN (with six 8-bit TFMs) with a degree-6 EM-based VN (with six 64-bit EMs). The TFM-based VN uses about 58% less area compared to the EM-based VN. When compared individu-

Module	Area $(\mu m^2)$	2-input gates	Delay (ps)
32-bit EM	2354	429	382
64-bit EM	4506	821	414
7-bit TFM	1249	228	658
8-bit TFM	1649	300	666
9-bit TFM	1924	350	669
degree-6 VN	26686	4861	652
(with 6 EMs)			
degree-6 VN	11011	2006	712
(with 6 TFMs)			

Table 1. Synthesis results in CMOS 90nm technology.

**Table 2.** Synthesis results for EM-based and TFM-based(576,228) stochastic LDPC decoders in CMOS 90nm technology.

Module	Area ( $\mu m^2$ )	2-input gates
EM-based decoder	4275615	778801
TFM-based decoder	1742334	317365

ally, TFMs are slower than EMs. For example, the delay of a 64-bit EM is about 38% less than the delay of an 8-bit TFM. However, it is important to note that the impact of the slower speed of TFMs on the overall delay of the VNs is much less. For example, as Table 1 shows, the overall delay of a TFM-based VN is 712 ps which is about 9% more than the delay of an EM-based VN.

To study the effects of TFMs on the overall area of stochastic decoders we implemented an EM-based and a TFM-based fully-parallel stochastic LDPC decoder which decodes a (576,288) irregular LDPC code. This LDPC code also belongs to the IEEE 802.16 (WiMAX) standard and has the same degree distribution as the LDPC code used in Section 4.1. The TFM-based decoders uses 8-bit TFMs and the EM-based decoder uses 64-bit EMs. Both decoders are synthesized in the ST Microelectronics 90 nm 1V CMOS technology and are clocked at 400 MHz. Note that as shown in [6], with this clock frequency a fully-parallel stochastic LDPC decoder is able to provide a high throughput of more than 1 Gb/s at low BERs (high SNRs). Table 2 summarizes the synthesis results for the two decoders. As shown, TFMs have significant effects on the complexity of the decoder. The area and the gate count of the TFM-based decoder is about 40% of the area and the gate count of the EM-based decoder.

## 5. CONCLUSIONS

The paper proposed TFMs for re-randomization and decorrelation of stochastic bit streams in stochastic channel decoders. We showed that TFMs are able to provide similar decoding performance as EMs for decoding state-of-the-art LDPC codes while having much lower hardware complexity. The paper also showed that TFMs have significant impact on the overall area of ASIC implementations of stochastic LDPC decoders.

#### 6. REFERENCES

- D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electron. Lett.*, vol. 32, no. 18, pp. 1645–1646, 1996.
- [2] F. Kschischang, B. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, Feb 2001.
- [3] V. Gaudet and A. Rapley, "Iterative decoding using stochastic computation," *Electron. Lett.*, vol. 39, no. 3, pp. 299–301, Feb. 2003.
- [4] A. Blanksby and C.J. Howland, "A 690-mw 1-Gb/s 1024-b rate-1/2 low-density parity-check code decoder," *IEEE J. of Solid-State Circuits*, vol. 37, no. 3, pp. 404– 412, March 2002.
- [5] A. Darabiha, A. Chan Carusone, and F. R. Kschischang, "Block-interlaced LDPC decoders with reduced interconnect complexity," *IEEE Trans. on Circuits and Systems-II: Express briefs*, vol. 55, no. 1, pp. 74–78, Jan. 2008.
- [6] S. Sharifi Tehrani, S. Mannor, and W. J. Gross, "Fully parallel stochastic LDPC decoders," *IEEE Trans. on Signal Processing*, vol. 56, no. 11, pp. 5692–5703, Nov. 2008.
- [7] Chris Winstead, "Error-control decoders and probabilistic computation," in *Tohoku Univ. 3rd SOIM-COE Conf.*, Sendai, Japan, Oct. 2005, pp. 349–352.
- [8] S. Sharifi Tehrani, W. J. Gross, and S. Mannor, "Stochastic decoding of LDPC codes," *IEEE Comm. Lett.*, vol. 10, no. 10, pp. 716–718, Oct. 2006.
- [9] S. Sharifi Tehrani, C. Jego, B. Zhu, and W. J. Gross, "Stohastic decoding of linear block codes with highdensiy parity-check matrices," *IEEE Trans. on Signal Processing*, vol. 56, no. 11, pp. 5733–5739, Nov. 2008.
- [10] S. Sharifi Tehrani, S. Mannor, and W. J. Gross, "An area-efficient FPGA-based architecture for fully-parallel stochastic LDPC decoding," in *Proc. of the IEEE Workshop on Signal Processing Systems (SiPS)*, Shanghai, China, Oct. 2007, pp. 255–260.