International Journal of Technical Innovation in Modern Engineering & Science (IJTIMES)
Impact Factor: 5.22 (SJIF-2017), e-ISSN: 2455-2585
Volume 4, Issue 6, June-2018

An Area delay Efficient S-box Decoding Architecture using LDP Codings

J. Bhavani[1], G. Venkata Rao[2,]
Dept. of ECE, Lakireddy Bali Reddy College of Engineering, jajulabhavani04@gmail.com
[2]Dept. of ECE, Lakireddy Bali Reddy College of Engineering, venkat8312005@gmail.com

Abstract— This brief presents an area-efficient relaxed half stochastic non-binary low-density parity-check (NB-LDPC) decoder. Furthermore, the hardware complexity of variable node units (VNUs) is reduced through a truncated architecture, which only keeps the most reliable n probability density functions. To accelerate the majority logic decoding of difference set low density parity check codes the error detection in memory applications was proposed. This is useful as majority logic decoding can be implemented serially with simple hardware but requires a large decoding time. For memory applications, the increase of the memory access time takes place. S-Box Codes are the class of linear block codes which provide near capacity performance on large collection of data transmission channels.

Index Terms—Non-binary low-density parity-check (LDPC) codes, relaxed half-stochastic (RHS) algorithm, stochastic decoding

I. INTRODUCTION

Non binary low-density parity-check (NB-LDPC) codes, investigated by Davey and Mackay, are defined on the null space of a parity-check matrix H, in which the nonzero entries are the elements of a Galois field GF($q = 2p$). Compared to binary LDPC codes, it has been shown that NB-LDPC codes can achieve better bit-error-rate (BER) performance for higher order modulation schemes or multi-input–multi-output communication systems. Despite its remarkable decoding capability, the complicated computational units and huge memory usage are main challenges to implement an NB-LDPC decoder. In this respect, a large amount of literatures with hardware oriented algorithms have been reported. Declercq and Fossorier proposed the fast Fourier transform (FFT) algorithm to transfer the complicated convolution operations into simpler multiplications. The extended min-sum (EMS) algorithm was proposed to simplify check node unit (CNU) and truncate message vectors from field size q to a limited number nm. In the min-max decoding algorithm was introduced to further reduce the complexity of CNU with acceptable performance degradation. On the other hand, the authors and showed a simplified serial algorithm for a generalized bit-flipping algorithm and a message compression algorithm to reduce the complexity and storage resources required by the variable node unit (VNU).

Stochastic computation is a promising decoding method for error control codes. It is a bit-serial representation of probability and has a great potential to reduce complicated computation. Due to its sequential property, a single wire and simple logic gates can be used to manipulate multiple bits arithmetic. The state-of-the-art works provide high-throughput and area efficient binary stochastic LDPC decoders for IEEE applications. In recent years, stochastic decoding provides alternative methods for NB-LDPC codes. However, more than thousands of decoding cycles in the shift register edge memory (SR-EM) and tracking forecast memory (TFM) algorithms remain a bottleneck for high-throughput decoders. The relaxed half-stochastic (RHS) algorithm provides higher convergence speed, but it is more complicated than the SR-EM or TFM algorithm due to the conversion of two domains. A field-programmable gate array (FPGA) implementation based on the adaptive multi set stochastic algorithm (AMSA) is introduced to reduce runtime and area cost, but the hardware complexity is still high due to a large amount of memory. Nowadays, pursuing a high-throughput and low cost decoder is still a design challenge of stochastic NB-LDPC decoding.

In this brief, we propose an area-efficient RHS decoding architecture for NB-LDPC codes. A cumulative TFM with concealing channel values (CTFM-CC) algorithm is proposed to reduce the operations and maintain BER performance. A truncated TFM architecture, as well as its updating criterion, is designed to achieve lower complexity of VNUs. A dynamic random number generation method is provided to generate an output symbol of VNU.

II. EXISTED SYSTEM

A stochastic NB-LDPC decoder consists of three types of processing units, including VNUs, CNUs, and permutation node units (PNUs), as shown in Fig. 1. It is noted that the message representation in the stochastic NB-LDPC decoder is a symbol over GF(q) rather than a probability vector of length q. Therefore, received channel values are converted to stochastic symbol streams over GF(q). The occurrence of a specific element in the stochastic symbol stream is equal to the probability of symbol to be converted. The location of elements in the stream is not important; in other words, the representation of the stochastic symbol stream is not unique.

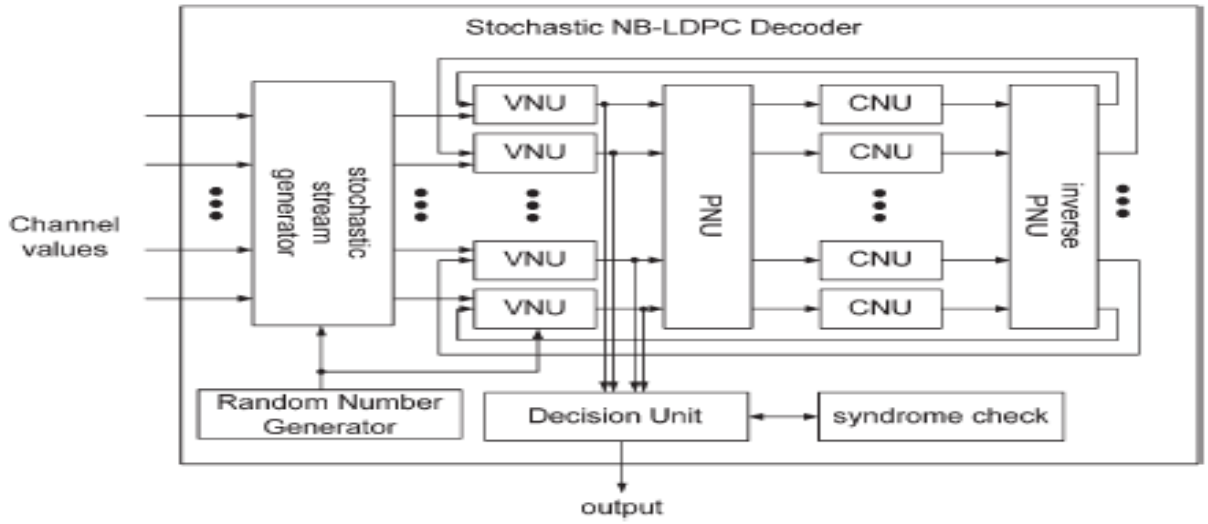

Fig.1: Architecture of stochastic NB-LDPC decoder

In the early research of stochastic LDPC decoding, the main challenge is performance degradation. The reason is the hold state problem, which refers to a case where a set of variable nodes are stuck to a fixed state. Therefore, variable nodes would employ memory devices to track previous symbols and create the random symbol z. The SR-EM and TFM algorithms were proposed to accurately generate a symbol.

However, it is hard to implement these algorithms for practical applications due to long decoding cycles. To enhance the rate of decoding convergence, the RHS algorithm for stochastic decoding of NB-LDPC codes was proposed. It was a hybrid decoding technique such that VNUs are operated in the probability domain, but CNUs are operated in the stochastic domain. In other words, the messages to be computed are

probabilities in the VNUs, but they are symbols in the CNUs. Compared to other related stochastic works, the number of iterations in an RHS decoder is significantly decreased. Moreover, the complexity of CNUs in a sum-product algorithm (SPA) decoder can be reduced by the stochastic approach.

A. CNU: The complexity of CNUs is the highest in the SPA decoder because of the convolution operations. Since the check equations of the RHS algorithm are operated in the symbol domain, the convolution operators can be simplified to finite field summations. Therefore, a CNU is implemented by EXCLUSIVEOR gates.

B. PNU: In the RHS algorithm, a PNU is fulfilled by a finite field multiplier since the output of PNU equals the product of the input and the corresponding nonzero element in H. In general, a lookup table and a combination of XOR gates is used for the small and high-order field, respectively, since the table size is proportional to the field size.

C. VNU: Since VNUs are operated in the probability domain but inputs/outputs are still symbols, two conversions are required. The first one is a stochastic-to-SPA message conversion. It is performed by a successive-relaxation method, in which the probability mass function (PMF) of each symbol γ is stored in TFMs. This updating rule is identical to that in the TFM algorithm.

III. PROPOSED SYSTEM

In this extended version, we investigate the uniformity problem and the need for re masking in more detail. We prove that under certain circumstances, it is enough to re mask only a fraction of the shares. Moreover, we argue that if there is enough re masking, we do not need to share functions uniformly. This observation helps us to further reduce the area and randomness requirements. We provide two new implementations.

The first one is similar to the one in, but it uses at least three shares in all the operations, including the linear ones. We use it to investigate the increase in security when moving from at least two to at least three shares, and to quantify the associated cost. The second implementation is based on the one in but modified according to our findings regarding uniformity and re masking.

Our three implementations need the same number of clock cycles to complete the calculation, and allow us therefore to focus on some trade-offs between area and additional randomness. For round operations and key schedule which requires only one S-box instance and loads the plaintext and key byte-wise in row-wise order we use a serial implementation.

The data unit consists of: the initial round of key addition and a final round. The need of the architecture of a standard round composed of both the transformation and the inverse transformation is for encryption and decryption respectively are performed using

the same hardware resources. This implementation generates one set of sub key and reuses for calculating all other sub keys in real-time.

1. Byte Sub: In this architecture each block is replaced by the substitution in S-Box table consisting of the byte of the block.

2. Shift Row: In this transformation the rows of the block state are shifted over different offsets. The amount of shifts is determined by the block length. The shift row operation using combinational logic considering the offset by which a row should be shifted was implemented by proposed architecture.

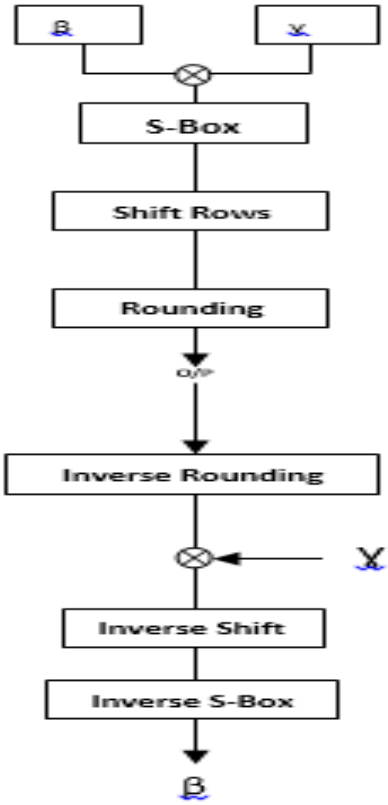

Fig. 2: Proposed System

If this technique is used to protect cascaded functions, then extra measures like the binary data discussed in the previous section

Page numbers, headers and footers must not be used.

IV. RESULTS

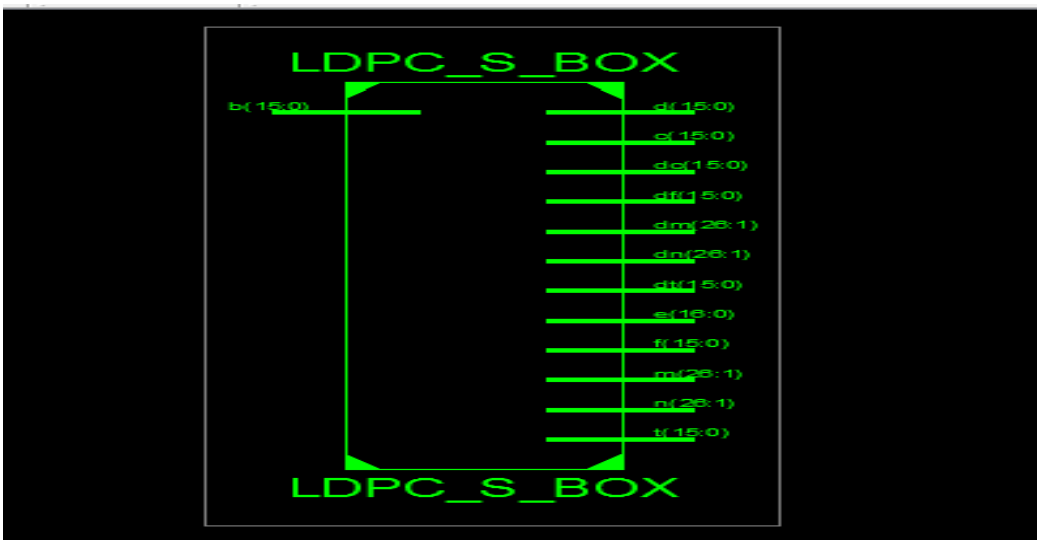

Fig.3: RTL-schematic for Ldpc codes

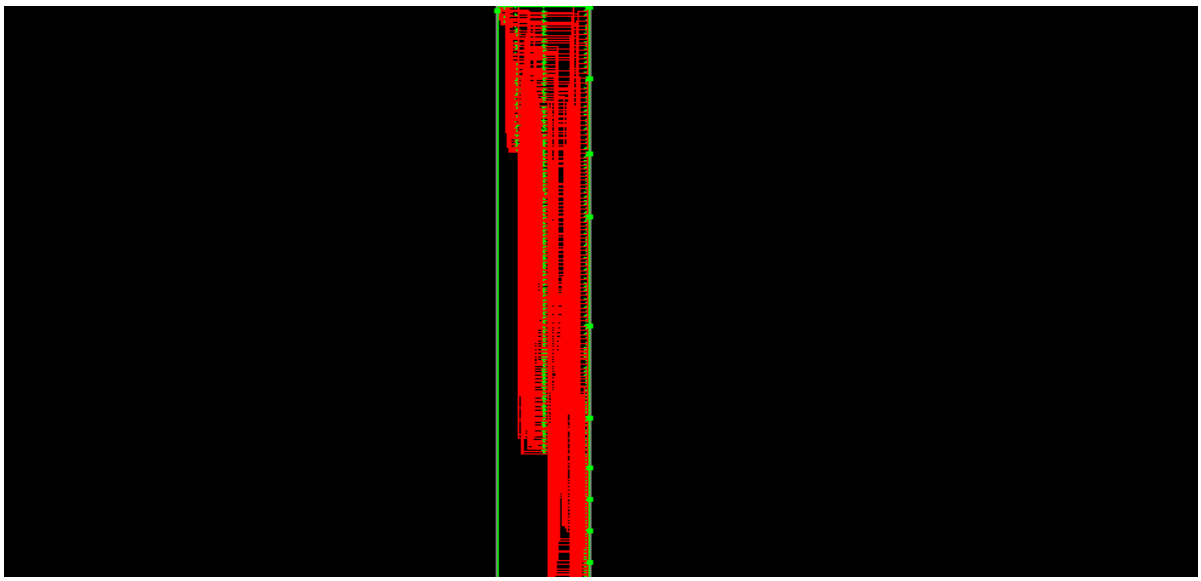

Fig.4: technology schematic of Ldpc codes

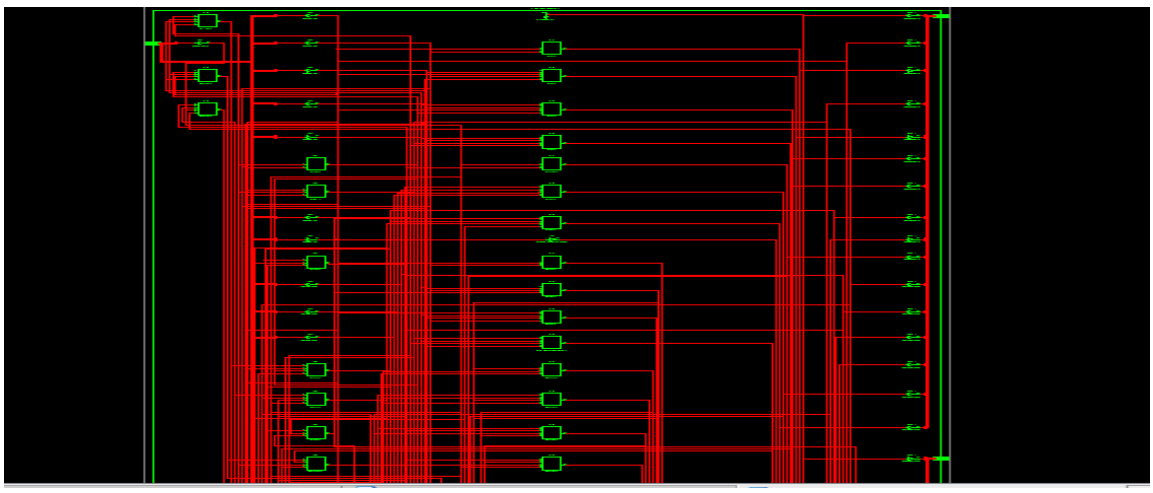

Fig. 5.Expandable view of Technology Schematic of Ldpc codes

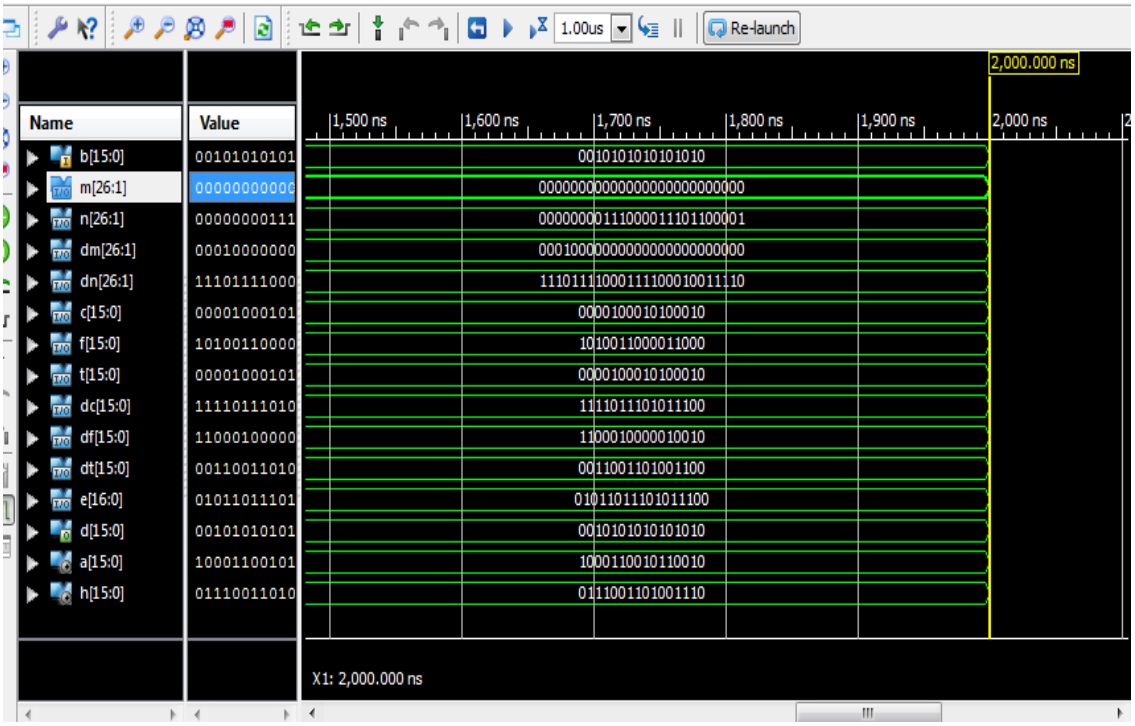

Fig. 6 Simulation Result of Ldpc codes

V. CONCLUSION

In this brief, the detection of errors during the first iterations of serial one step Majority Logic Decoding of S-Box codes has been studied. To reduce the decoding time by stopping the decoding process is the main objective when no errors are detected. The tested combinations of errors affecting up to four bits are detected in the first three iterations of decoding is showed by simulation results. These results extend the Ones recently presented for S-Box codes, making the modified one step majority logic decoding more attractive for memory applications. The larger choice of word lengths and error correction capabilities was now by designer

VI. REFERENCES

[1] M. Davey and D. Mackay, "Low-density parity check codes over GF(q)," *IEEE Commun. Lett.*, vol. 2, no. 6, pp. 165–167, Jun. 1998.

[2] X. Jiang, Y. Yan, X.-G. Xia, and M. H. Lee, "Application of nonbinary LDPC codes based on Euclidean geometries to MIMO systems," in *Proc. Int. Conf. WCSP*, 2009, pp. 1–5.

[3] D. Declercq andM. Fossorier, "Decoding algorithms for nonbinary LDPC codes over GF(q)," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 633–643, Apr. 2007.

[4] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urard, "Lowcomplexity decoding for non-binary LDPC codes in high order fields," *IEEE Trans. Commun.*, vol. 58, no. 5, pp. 1365–1375, May 2010.

[5] V. Savin, "Min-Max decoding for non binary LDPC codes," in *Proc. IEEE ISIT*, 2008, pp. 960–964.

[6] X. Zhang and F. Cai, "Reduced-complexity decoder architecture for nonbinary LDPC codes," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 7, pp. 1229–1238, Jul. 2011.

[7] Y. L. Ueng *et al.*, "An efficient layered decoding architecture for nonbinary QC-LDPC codes," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 2, pp. 385–398, Feb. 2012.

[8] X. Chen, S. Lin, and V. Akella, "Efficient configurable decoder architecture for nonbinary quasi-cyclic LDPC codes," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 1, pp. 188–197, Jan. 2012.

[9] F. Garcia-Herrero, M. J. Canet, and J. Valls, "Nonbinary LDPC decoder based on simplified enhanced generalized bit-flipping algorithm," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 6, pp. 1455–1459, Jun. 2014.

[10] J. Lin and Z. Yan, "An efficient fully parallel decoder architecture for nonbinary LDPC codes," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 12, pp. 2649–2660, Dec. 2014