International Journal For Technological Research In Engineering
Volume 1, Issue 9, May-2014 *ISSN (Online): 2347 - 4718*

DESIGN AND SYNTHESIS OF BINARY LDPC DECODER

Ms. Rashmi H[1], Mr. Naveen K B[2], Mr. Ananda Raju M B[3]
[1]PG Student VLSI Design and Embedded System
[2]Asst Professor, [3]HOD Department of Electronics & Communication Engineering
BGSIT, B. G. Nagar, Karnataka, India.

Abstract: This paper is aimed to optimize the binary LDPC decoder which has been shown to outperform commonly used codes for many communications and storage channels. Currently proposed binary LDPC decoder architectures have very high rates close to the channel capacity for the large code word length, in this research work we had presented a decoding algorithm based on belief propagation and provide details of the hardware resources required for an implementation with synthesis result for decoder.
Keywords: LDPC, Decoder algorithm.

I. INTRODUCTION

Low density parity-check codes, after their rediscovery in late 90's, have attracted vast research attenuation due to their excellent error correcting performance and highly parallel iterative decoding method. They have become the industry standard for error correction coding and adopted for instance in the Digital Video Broadcasting (DVB) and the IEEE wimax. A satisfying LDPC decoder usually means: good error correction performance, low hardware complexity, High throughput. In other case, the major drawback of codes over higher order fields is their decoding algorithm complexity. Binary LDPC codes are decoded using belief propagation their factors graphs .moreover, the decoding complexity of Galois field codes scale exponentially as (2pdc) and, where dc is the maximal number of nonzero entries in the parity check matrix row. In the recent year ,a few hardware non-binary decoder implementations have been presented in literature [4] & [5].some variations of classical belief propagation decoding algorithm with no need of multiplication large number of multipliers is available for designed disposal .it is desirable to make use of the multiple cores. In this paper, we present the decoder structure the efficiently uses all the FPGA resources.

II. PROBLEM FORMULATION

The performance for short block length codes is significantly higher for non-binary codes (over higher order GF fields), but on the other hand, the decoding complexity is increased. Therefore the hardware implementation of a decoder is still a challenging task.

III. DESIGN DESCRIPTION

Figure shows the schematic block diagram of LDPC Decoder. With the assertion of start ldpc decoding signal, the decoder core starts receiving data from the pins info data and Parity data at a rate, 360 LLR values per clock cycle. Once a

complete frame of data is received, the decoding process starts. When the decoding process finishes, the LDPC core asserts done ldpc decoding signal indicating the end of decoding. Hard decoding pass signal indicates the success of ldpc decoding. The decoded data is transmitted out at a rate 360 bits per clock.

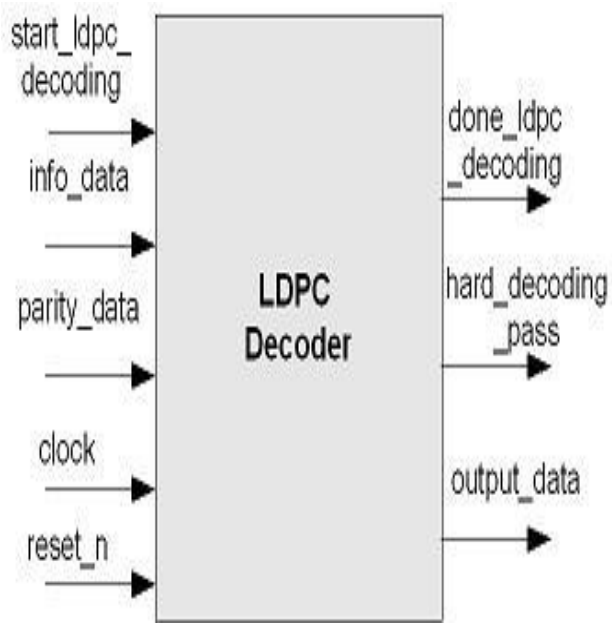

Fig. 1: LDPC Decoder

IV. PROPOSED METHODOLOGY

The decoding algorithm has the following four stages.

Initialization:
The channel LLR Values are assigned to the edges that goes out from Bit Node Unit.

Check Node Update:
Check Node Units receives the data from Bit Node edges and process the data according to Min Sum algorithm. The processed data is transmitted to Bit Node Units.

Bit Node Update:
All the edges that go out from Check node to Bit node are added to compute a sum value. This Sum value is used for Hard Decoding. Bit Node value is updated by subtracting its value from sum value.

Hard Decoding:

Depending on the sign of Sum value, the transmitted data is determined. Using these values, the parity check equations are computed. If all parity equations are satisfied then the decoder stops, otherwise another Check Node and Bit Node Update is performed.

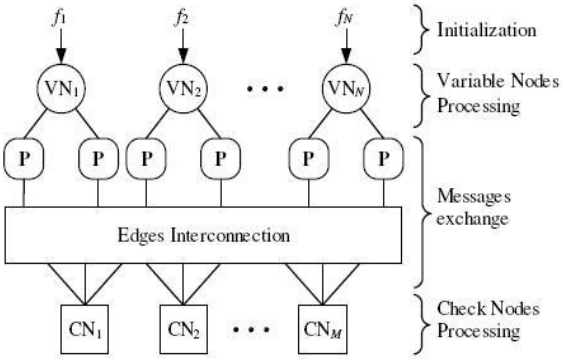

Fig. 2: Graph representation of decoder algorithm

V. ALGORITHM VARIATION

MacKay et al. [2] provided an efficient method for calculating r_{mn}^{a} in (4) by means of partial sums, which can be seen as an application of BCJR algorithm [6]. However it is well known that the complexity of this algorithm can be scaled down if its dual form in the frequency domain is used [7]. The improvement is based on observation that (4) can be reformulated as a convolution operation in the Galois field. This convolution can be evaluated efficiently using the product of p-dimensional two-point FFT and then inverse FFT (IFFT) of the result. The thorough study of the achieved complexity improvement can be found e.g. in [8]. Another algorithm reformulations use logarithm domain with log-density [9] or log-density-ratio representation of messages [10]. Logarithm domain algorithms require fewer quantization levels due to its lower sensitivity to quantization effects [11]. Moreover, the product operations in variable nodes processing and tentative decoding become summations, which entail less chip area and energy consumption in hardware implementation. On the other hand in check nodes processing, calculation of a complex nonlinear operation ⊞ [10] is required. To cope with it some algorithms with reduced complexity have been proposed, such as Min-Max decoding [12] or extended min-sum (EMS) decoding [13], also with truncated message vector [14]. Remark that such a complexity reduction is inevitably related with some decoding performance degradation. In this article we propose similar to proposed in [3] mixed domain algorithm formulation with the main difference being that check nodes operate in the real domain. Such an approach is intended specifically for FPGA decoder implementation and the motivation is as follows: CNs operating in the real domain requires multiplication operations, but contemporary FPGA devices contain a large amount of hardware multipliers that remain unused otherwise. It is then desirable to distribute the computational resources among multipliers (CNs) and adders / subtractors (VNs) implemented in the basic FPGA resources. Since the CNs processing includes FFT and IFFT, which is calculated in the real domain anyway, our proposition requires only two domain changes per iteration (on the input and output of the CNs) instead of four domain changes in [3]. Therefore we can save logic resources required for log (.) and exp (.) functions calculation.

VI. EXPERIMENTAL RESULT

Fig. 3: Simulation result of LDPC Decoder.

Figure shows the simulated result for LDPC Decoder, where the 32 bit input data with clock set to 1, and reset set to 0 based on the input data stream is simulated based on the above methodology.

VII. CONCLUSION

In this project we presented the LDPC decoder design approach targeted for the FPGA devices, however the VHDL description is designed in a way allowing the use with other hardware platforms too. Effectiveness of the presented realization is based on balanced utilization of all types of FPGA resources, particularly making use of the multiplier cores. In order to achieve this, the modified decoding algorithm formulation has been proposed at first. Then we presented the decoder structure, also pointing out some important aspects of FPGA implementation, such as messages normalization and pipeline processing. The synthesis results for serial architecture show that the designed decoder can be implemented in a small part (several percent) of the modern FPGA device resources with moderate throughput.

REFERENCES

[1] David J. C. MacKay, "Good Error-Correcting Codes Based on Very Sparse Matrices," IEEE Trans. Inf. Theory, vol. 45, pp. 399–431, March 1999.
[2] Matthew C. Davey and David MacKay, "Low-Density Parity Check Codes over GF (q)," IEEE Commun. Lett., vol. 2, pp. 165–167, June 1998.
[3] Christian Spagnol, Emanuel Mihai Popovici, and William Peter Marnane, "Hardware Implementation of GF (2m) LDPC Decoders," IEEE Trans. Circuits Syst. I, vol. 56, pp. 2609– 2620, December 2009.

[4] Jun Lin, Jin Sha, Zhongfeng Wang, and Li Li, "Efficient Decoder Design for Nonbinary Quasicyclic LDPC Codes," IEEE Trans. Circuits Syst. I, vol. 57, pp. 1071–1082, May 2010.
[5] Timo Lehnigk-Emden and Norbert Wehn, "Complexity Evaluation of Non-binary Galois Field LDPC Code Decoders," in Proc. (IEEE) 6th International Symposium on Turbo Codes & Iterative Information Processing, Brest, France, September 2010.
[6] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," IEEE Trans. Inf.Theory, vol. 20, pp. 284–287, March 1974.
[7] L. Barnault and D. Declercq, "Fast Decoding Algorithm for LDPC over GF(2q)," in Proc. (IEEE) Information Theory Workshop, Paris, France, April 2003.
[8] Amir Bennatan and David Burshtein, "Design and Analysis of Nonbinary LDPC Codes for Arbitrary Discrete-Memoryless Channels," IEEE Trans. Inf. Theory, vol. 52, pp. 549–583, February 2006.
[9] Hongxin Song and J. R. Cruz, "Reduced-Complexity Decoding of Q-ary LDPC Codes for Magnetic Recording," IEEE Trans. Magn., vol. 39, pp. 1081–1087, March 2003.
[10] Henk Wymeersch, Heidi Steendam, and Marc Moeneclaey, "Log-domain Decoding of LDPC Codes over GF(q)," in Proc. (IEEE) International Conference on Communications, Paris, France, June 2004.
[11] Henk Wymeersch, Heidi Steendam, and Marc Moeneclaey, "Computational Complexity and Quantization Effects of Decoding Algorithms for Non-binary LDPC Codes," in Proc. (IEEE) International Conference on Acoustics, Speech, and Signal Processing, Montreal, Canada, May 2004.
[12] Valentin Savin, "Min-Max decoding for non-binary LDPC codes," in Proc. (IEEE) IEEE International Symposium on Information Theory, Toronto, Canada, July 2008.
[13] David Declercq and Marc Fossorier, "Decoding Algorithms for Nonbinary LDPC Codes Over GF (q)," IEEE Trans. Commun., vol. 55, pp.633–643, April 2007.
[14] Adrian Voicila, David Declercq, Francois Verdier, Marc Fossorier, and Pascal Urard, "Low-Complexity Decoding for Non-Binary LDPC Codes in High Order Fields," IEEE Trans. Commun., vol. 58, pp. 1365–1375, May 2010.
[15] C. L. Wu and T. Y. Feng, "On a class of multistage interconnection networks," IEEE Trans. Comput., vol. C-29, pp. 694–702, 1980.
[16] Hao Zhong and Tong Zhang, "Block-LDPC: A Practical LDPC Coding System Design Approach," IEEE Trans. Circuits Syst. I, vol. 52, pp. 766–775, April 2005.
[17] Xiao Yu Hu, Evangelos Eleftheriou, and Dieter M. Arnold, "Regular and Irregular Progressive Edge-Growth Tanner Graphs," IEEE Trans. Inf. Theory, vol. 51, pp. 386– 398, January 2005.

