International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682
Vol.2, No.10, October 2013 DOI:10.15693/ijaist/2013.v2i10.81-85

Design Of Wave Pipelined Based Ldpc Decoder And Implementation Using Adaptive Message Control

T.SATHIYA
II YEAR M.E VLSI DESIGN
Department of ECE
PSNA College of Engg & Tech,
Dindigul.

Mr.P.THIRUMURUGAN
ASSISTANT PROFESSOR
Department of ECE
PSNA College of Engg & Tech,
Dindigul.

Dr.G.ATHISHA
PROFESSOR AND HEAD
Department of ECE
PSNA College of Engg & Tech,
Dindigul

Abstract

A new decoder architecture for low-density parity check (LDPC) codes is proposed in this paper to reduce the hardware operational complexity in VLSI implementations. The low decoding complexity of LDPC Decoder Architecture is achieved by employing adaptive message control (AMC) with wave pipelined to reduce the amount of memory accesses and arithmetic operations. The proposed LDPC decoder architecture can significantly reduce hardware operations and power consumption as compared with existing work. The architecture is synthesized on a Xilinx 9.2i as Synopsys tool targeted to CPLD.

I. INTRODUCTION

Low-density parity-check (LDPC) codes [1] have recently been adopted for several data communication applications due to their superior coding performance and parallelizable decoder architecture. LDPC codes allow a fine-level parallel message-passing decoding in which all the check and variable nodes are updated concurrently. This parallelism can potentially be used to build a decoder with Multi- Gbit/sec throughput. The major obstacle for efficient implementation of fully-parallel LDPC decoders is interconnect complexity which is the result of random location of 1's in the code's parity-check matrix. In this paper, we propose a bit-serial scheme for fully-parallel LDPC decoders. Bit-serial computation allows variable and check nodes to communicate multi-bit messages over single wires, hence reducing the interconnect complexity. In addition, we introduce a new approximation to the check update function in min-sum decoding. In this approximation, in each check node only one minimum magnitude is calculated over all the check node inputs. Depending on the number of inputs that share the same minimum magnitude, a corrective constant is then added in order to generate the proper check outputs. We show that with 4-bit quantization this approximation reduces the check node area by 48% while introducing less than 0.1 dB loss in BER performance. We illustrate feasibility of bit-serial LDPC decoding by implementing a (480, 355) RS-based LDPC decoder on a single Altera Stratix EP1S80 FPGA device based on the new proposed check node architecture. The decoder operates at maximum clock frequency of 61 MHz, performs 15 decoding iterations per frame and achieves 650 Mbps throughput. In message-passing LDPC decoding, a large number of messages need to be updated and transfered between check and variable nodes in each iteration. Previous works have proposed several approaches for representing and updating these messages. In [5], analog signals are used to represent the extrinsic messages. In analog decoders the exponential voltage-current relationship of a transistor is used to realize the message-passing update functions. Although analog decoders have the advantage of low power consumption, they become impractical for decoding long LDPC codes due to the noise and process mismatch. More conventional LDPC decoders often use multi-bit digital signals to represent the messages. In partially-parallel decoders [6], [7], the messages are transferred between the nodes through memory. This architecture reduces the decoder area by sharing the processing units, but this comes at the cost of reduced throughput. To achieve higher throughput, in the fully-parallel decoder presented in [8], all check and variable nodes are directly instantiated in hardware. Using this architecture, a throughput of 1 Gbps with 64 iterations per frame is reported. The major challenge in the implementation of fully parallel LDPC decoders is the complex and random interconnection between the variable and check nodes. This problem is worsened when multi-bit buses are used to realize the edges in the code Tanner graph.

II. RELATED WORK

Among various decoding algorithms for LDPC codes, Min-sum (MS) decoding algorithm has less mathematical complexity. It is considered as an approximation to the iterative SP (Sum-Product) algorithm [5]. Although the performance of MS is generally few tenths of dB less than that of SP decoding, the product term present in the SP decoding makes the architecture more complex, thereby focusing research on MS decoding[6]. The implementation has trade-off between decoding performance and hardware complexity. Various architectures [8–10] for implementation are dealt in the literature. In serial architecture, the area is lesser but it has latency overhead. While considering parallel architecture, the latency overhead is overcome but the area gets increased as compared to the serial architecture. To optimize the trade-off between the area, power and latency overhead, the concept of wave pipelining has been introduced. Also the concept of wave pipelining has improved the performance of the LDPC encoder in terms of area, power and latency. Different from these existing work targeting hardware implementation cost, the focus of this paper is to reduce the hardware operational complexity in nonbinary LDPC decoder architectures. This enables efficient decoding suitable for emerging applications such as underwater acoustic sensor networks [17] that are under the severe resource (e.g., energy) constraints. It was reported [4] that memory accesses and arithmetic operations are the two major contributors to the operating cost in LDPC decoders. A lot of research effort aims at reducing the decoding complexity of non binary LDPC codes at the algorithm level [7]–[9], [11]. To deal with the problem that computational complexity increases exponentially.

III. PROPOSED ARCHITECTURE

We propose Adaptive Message Control (AMC) with Wave Pipelining in LDPC Decoder Architecture.The Proposed method adjusts the message length adaptively, which can reduce the message length at the required performance. The inputs to the variable nodes are from the check nodes. Unlike the check nodes, from the first iteration itself the inputs are fed from the check node. In each iteration the variable nodes get updated by the messages from the corresponding check nodes (i). Figure shows the architecture of variable node unit of Min-sum decoder. Intermediate registers are included between the iterations which explain the concept of conventional pipelining.

Min-Sum Algorithm

Min-sum (MS) decoding algorithm is an approximation of the iterative sum-product (SP) algorithm. Even though the

LDPC Decoding:

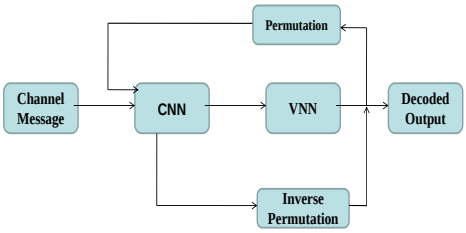

Fig.1 LDPC Decoding

performance of MS is generally a few tenths of a dB lower than that of SP decoding, it is most robust to quantization errors when implemented with fixed-point operations. Moreover min-sum is of reduced complexity and the knowledge of noise power is unnecessary in this case. In MS, the hardware for the check node function is simple when compared to the SP algorithm.

In MS decoding, similar to SP algorithm, the extrinsic messages are passed between check and variable nodes in the form of log likelihood ratios (LLRs). Let $Z_{mn}^{(i)}$ represents the LLR value for bit *n* sent from variable node v_n to the check node c_m in the i^{th} iteration and similarly $\varepsilon_{mn}^{(i)}$ represents the LLR value for bit *n*, sent from check node c_m to variable node v_n in the i^{th} iteration. Suppose W = $\{w_1, w_2,.....,w_N\} \in C$ and Y $=\{y_1,y_2,.....,y_N\}$ are the transmitted codeword and the received sequence respectively. The MS decoding algorithm consists of the following steps:

1. Initialize the iteration counter, *i* to 1 and let I_M be the maximum number of iterations allowed.

2. Initialize $Z_{mn}^{(0)}$ to the aposterior LLR,

$\lambda_n = \log(P(v_n : 0|y_n)/P(v_n = 1|y_n))$ for $1 \le n \le N$, $m \in M(n)$.

3. Update the check nodes, i.e., for $1 \le m \le M$, $n \in N(m)$ calculates,

$$\varepsilon_{mn}^{(i)} = \min_{n' \in N(m)\backslash n} \left| Z_{mn'}^{(i)} \right| \prod_{n' \in N(m)\backslash n} sgn(Z_{mn'}^{(i)})$$

(1)

4. Update the variable nodes, for $1 \le n \le N, m \in M(n)$, calculates,

$$Z_{mn}^{(i)} = \sum_{m' \in M(n)\backslash m} \varepsilon_{m'n}^{(i)}$$

(2)

5. Apply a hard decision, i.e., compute $\hat{W} = \{\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_N\}$ where element $\hat{w}_n$ is calculated as,

$$\hat{w}_n = \begin{cases} 0, & if\ \lambda_n + \sum_{m \in M(n)} \varepsilon_{mn}^{(i)} \ge 0 \\ 1, & otherwise \end{cases}$$

(3)

If $\hat{W}H^T = 0$ (or) $i = I_M$ stop the decoding and go to step 6, otherwise set $i=i+1$ and go to step 3.

6. Output $\hat{W}^{(i)}$, as the decoder output.

The conventional method for decreasing the clock period of the system is by pipelining the combinational block, where the combinational block is divided into 2 and the registers are inserted in between them. It leads in the reduction in the critical path. In this work, LDPC decoder architecture has been designed with Min-Sum algorithm. The check node module and the variable nodes are depicted in the following fig.2,3. The inputs to the check nodes are from the variable nodes except for the first iteration. For first iteration, the inputs are fed from the channel, a posterior LLR and from the second iteration onwards, the inputs are fed from the variable nodes. According to the PCM matrix, connection exists between the nodes and accordingly the messages are iterated.

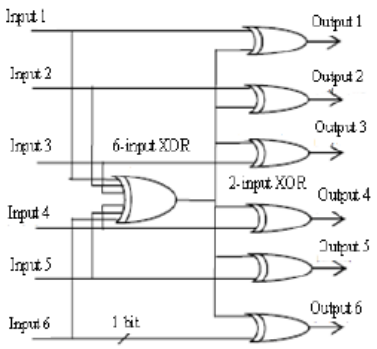

Fig.2 Check Node Unit

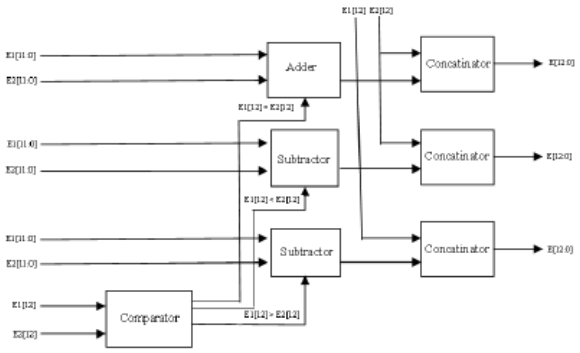

Fig.3 Variable Node Unit

The controller is present to check whether the hard decision has been achieved. If the hard decision is not achieved the iteration count gets incremented and the updating of the CNU and VNU takes place. At each iteration, the hard decision is checked. The output of each iteration is stored in the intermediate registers and fed to the input of second iteration. This explains the concept of conventional pipelining. The presence of the intermediate registers increases the area, which is an overhead. Even though the latency overhead gets improved when compared to the serial architecture, the area gets increased. The concept of wave-pipelining is incorporated to optimize the trade-off between the area and the latency overhead.

IV. SIMULATION RESULTS

In proposed architecture, the higher throughput, less power consumption and less area are achieved .The architecture is implemented using spartan3E family and XC3S500E device in Xilinx 9.2i.The proposed system is written in Verilog HDL language and synthesized in Xilinx 9.2i and stimulated using Modelsim 5.7. Dynamic power is defined as amount of power consumed by switching activities of FF, where as static power is power

consumed by leakage current. In 200MHz operation the Coprocessor consumes 79mW in static and 96mW in dynamic in the total summation of 175mW.

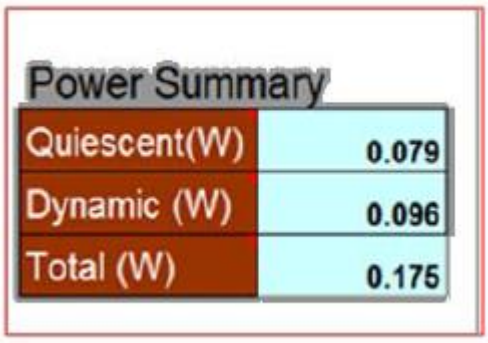

Fig.4 Power summary

V. CONCLUSION

Thus I, conclude that the proposed LDPC decoder architecture can significantly reduce hardware operations and power consumption as compared with existing work. As we use adaptive message control with wave pipelined based LDPC decoder architecture, the amount of memory access and arithmetic operations are reduced. The power consumption is compared with the existing system and shown in Fig.5 as a table. The architecture is synthesized on a Xilinx 9.2i as Synopsys tool targeted to CPLD.

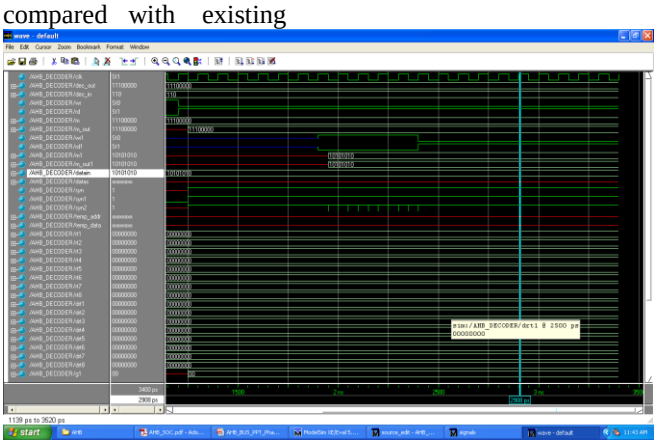

Fig.5 Simulation Result

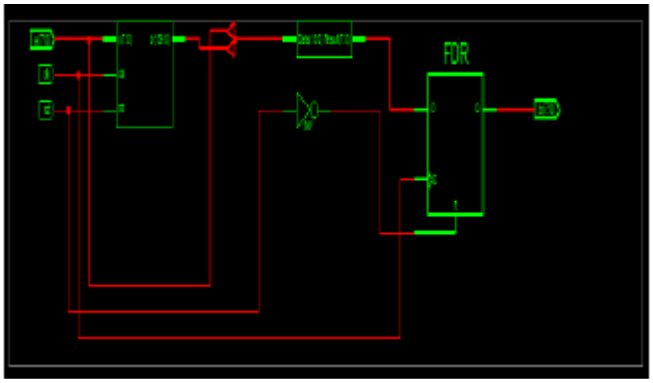

Fig.6 RTL View of Proposed Architecture

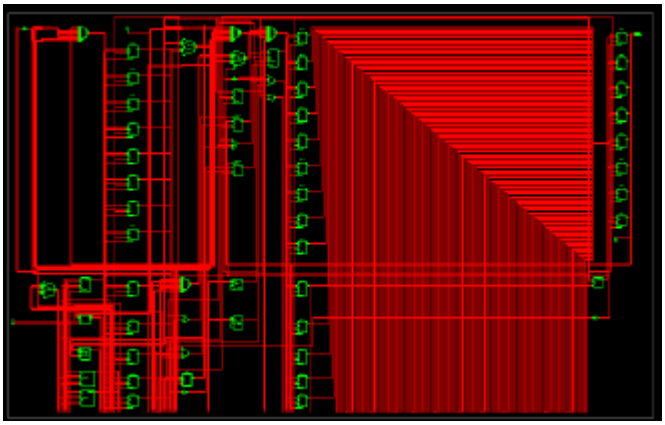

Fig.7 Technology Schematic View

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	682	9,312	7%	
Number of 4 input LUTs	1,987	9,312	21%	
Logic Distribution				
Number of occupied Slices	1,034	4,656	22%	
Number of Slices containing only related logic	1,034	1,034	100%	
Number of Slices containing unrelated logic	0	1,034	0%	
Total Number of 4 input LUTs	2,008	9,312	21%	
Number used as logic	1,987			
Number used as a route-thru	21			
Number of bonded IOBs	18	232	7%	
IOB Flip Flops	8			
Number of GCLKs	1	24	4%	
Total equivalent gate count for design	22,689			
Additional JTAG gate count for IOBs	864			

Fig.8 Device Utilization Results

VI.REFERENCES

[1] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT press, 1963.
[2] N. Wiberg, *Codes and decoding on general graphs, PhD thesis*. Linkoping: Linkoping University, 1996.
[3] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. on Information Theory*, vol. 47, pp. 498–519, Feb. 2001.
[4] A. B. F. Zarkeshvari, "On implementation of min-sum algorithm for decoding low-density parity-check (LDPC) cpdes," in *IEEE Globecom conference*, 2002.

[5] F. Lustenberger, *On the design of analog VLSI iterative codes*, *PhD thesis*. Zurich: Swiss Federal Institute of Technology, 2000.
[6] E. Yeo, P. Pakzad, B. Nikolic, and V. Anantharam, "VLSI architectures for iterative decoders in magnetic recording channels," *IEEE Transactions on Magnetics*, vol. 37, pp. 748–755, March 2001.
[7] T. Zhang and K. K. Parhi, "A 54 MBPS (3, 6)-regular FPGA LDPC decoder," in *IEEE Workshop on Signal Processing Systems*, San Diego, CA, 2002.
[8] A. J. Blanksby and C. J. Howland, "A 690-mW 1-Gb/s 1024-b, rate-1/2 low-density parity-check decoder," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 3, Mar. 2002.
[9] V. Gaudet and A. Rapley, "Iterative decoding using stochastic computation," *Electronics Letters*, vol. 39, no. 3, pp. 299–301, February 2003.
[10] M. Ardakani and F. R. Kschischang, "Gear-shift decoding," in *Proc. 21st Biennial Symp. on Comm.*, Queen's University, Canada, 2002.
[11] I. Djurdjevic, J. Xu, K. Abdel-Ghaffar, and S. Lin, "A class of lowdensity parity-check codes constructed based on Reed-Solomon codes with two information symbols," *IEEE Comm. Letters*, vol. 7, no. 7, July 2003.
[12] X.-Y. Hu, E. Eleftheriou, D.-M. Arnold, and A. Dholakia, "Efficient implementation of the sum-product algorithm for decoding LDPC codes," in *IEEE Global Telecommunications Conference*, vol. 2, San Antonio, TX, 2001, pp. 1036–1036E.
[13] A. Darabiha, A. Chan Carusone, and F. R. Kschischang, "Blockinterlaced fully-parallel LDPC decoders with reduced interconnect complexity," *submitted to IEEE Transactions on VLSI Systems*.
[14] X. Zhang and F. Cai, "Efficient partial-parallel decoder architecture for quasi-cyclic nonbinary LDPC codes," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 2, pp. 402–414, Feb. 2010.

Author Profile

T.Sathiya received the **B.E.** degree in electronics and communication engineering from the R.V.S College of Engineering and technology, dindigul, Anna University, Chennai, India, in 2011.Currently doing **M.E.** in electronics and communication engineering (VLSI Design) in PSNA College of Engineering and technology, Dindigul, India. Her research interest includes wireless communication networks (**WiFi,WiMax**), Mobile Ad hoc networks ,Sensor Networks , Communication networks.