Area efficient FPGA based LDPC decoder using Stochastic decoding scheme

Aquib Abdul Quadir Quraishi
School of electronics engineering (SENSE)
VIT University, Chennai Campus
Chennai, TN, India
quraishi.mdaquibaquadir@vit.ac.in

Abstract: This paper demonstrates the FPGA implementation of area efficient LDPC decoder using stochastic decoding scheme .Error correcting codes in communication systems are of high importance to achieve stability in wireless communication system. This paper presents the overview of designing an area efficient LDPC decoding architecture using stochastic decoding scheme. The Implementation results of fully parallel LDPC decoder displays 1% of slice using Xilinx vertex-4, 4vfx12sf363-10 device. The implementation of decoder architecture using stochastic decoding algorithm has reduced the number of slices required thereby reducing reducing the number of resources used and thus increasing the area efficiency of the decoder. The decoding scheme has also reduced the interconnect complexity of decoder architecture.

I. INTRODUCTION

LDPC codes are also abbreviated as low density parity check codes are very important topic of wide research as they are widely capable of correcting error in wireless communication devices. LDPC codes have the performance ability of correcting error near to Shannon-limit codes. LDPC codes play an important part in providing stability to wireless communication devices like Wimax. Despite of the high importance of LDPC codes VLSI implementation of LDPC codes was very much difficult because of the complexity involved in their implementation of LDPC codes. For efficient VLSI implementation of LDPC decoder various algorithms were proposed [1] . Though the VLSI implementation of LDPC decoder using the proposed algorithms were successful, but the proposed decoding architecture met with some severe issues of area efficiency, complex decoding architecture and routing and interconnect complexity LDPC codes are iteratively decoded by using the Sum- product algorithm. Sum product algorithm iteratively decodes LDPC by means of belief propagation. Some other algorithm proposed were Bit Flip algorithm, min-sum algorithm. In Bit Flip algorithm check bits are satisfied only if sum of adjacent variable nodes are satisfied .The disadvantage of this Algorithm is that it is successful if graph has good successful properties. The SPA passes the message over the edges of bipartite graph (2) .SPA offers reduced complexity but at the cost of 0.5 to 1 dB decoding loss [3,4].

Stochastic decoding technique is the recently proposed technique for developing a decoding architecture. Stochastic decoding technique results in decoding architecture with less number of nodes thereby reducing the complexity of the decoding architecture and increasing the area efficiency of the LDPC decoder. Stochastic decoding is based on stochastic computation techniques proposed in 1960.It was proposed with the aim of using the calculation to design low precision circuits [5].In stochastic decoding technique unlike other decoding technique computation is done using stochastic streams. The probabilities are represented using stochastic streams. The representation of stochastic streams results in simplicity of calculation thereby avoiding any complex calculations .Stochastic computation technique was first utilized for decoding error correcting codes like LDPC [6,7] and Hamming codes [8,9]. The decoder discussed in this paper is designed to work on binary codes.

Chitra K
School of electronics engineering (SENSE)
VIT University Chennai Campus
Chennai, TN, India
chitra.krishnan@vit.ac.in

II. LDPC DECODER IMPLEMENTATION

The main difficulty in VLSI implementation of LDPC decoder is to have area efficient architecture which will be successful in passing the message during the iterative belief propagation decoding. Two categories of decoders are available for LDPC decoding scheme.

- *Fully parallel decoders*: They are directly instantiated on bipartite graph of LDPC code to hardware
- *Partially parallel:* This category of decoders are used in mapping number of variable nodes and check nodes to a particular single unit in time division multiplexing node

Each variable node and check node are separately treated as a unit that decodes node. The connection to all nodes are made by an interconnecting network that resembles connectivity of bipartite .The advantage of fully parallel decoder is that it has a very high decoding throughput [10,11]. The biggest drawback of fully parallel decoder is the interconnection complexity. The fully parallel decoder has a very high degree of routing complexity and interconnection congestion. The reason for the interconnect complexity is the random-like connection which exists between variable nodes and check nodes. Despite of the above drawbacks fully parallel decoder is preferred over partially parallel decoder for decoding LDPC codes due to its high speed of operation and high throughput. The tradeoffs between complexity of hardware and speed of decoding are appropriately targeted by partially parallel decoder. The implementation of partially parallel decoder requires structured bipartite graph of LDPC code for efficiently realizing partially parallel passing of messages. The generation of LDPC codes is based on random construction which results in bipartite graph topology which is not properly structured. Due to this drawback partially parallel decoders are not suited for LDPC decoding. This paper proposes an efficient decoding algorithm scheme that will be very much suitable to have an area efficient LDPC decoding architecture using Xilinx vertex 4 device.

III. STOCHASTIC COMPUTATION AND DECODING OF LDPC.

Stochastic computation is based on the principle of transformation of probabilities into streams of stochastic bits .These bits are generated by the use of Bernoulli sequences. Here stream bit is equal to one with the probability with which it is transformed. To have a better understanding let us say for example a frame of 10 bits has been used .The frame having 7 bits equal to 1, denotes the probability of 1 to be 0.7.An important point about stochastic sequences are that it cannot be necessarily a frame of bits Instead stochastic sequences can also be sequences of bits , where no framing is used[12]. The process of transforming the probability into stochastic streams can never be unique. Different stochastic streams can have same probabilities. This idea is clearly explained in example 1.Example 1 shows different possible combination of stochastic streams each having the same probabilities of 6/10. The stochastic representation allows having simple solutions for complex probability operations. The multiplications and division operations can be executed with simple AND gate. Fig 1 shows simple AND gate used for multiplication operations.

0111011010….P=6/10

1110100101….P=6/10

1100110011…P=6/10

Example 1 . Different stochastic stream having the same probability of 6/10.

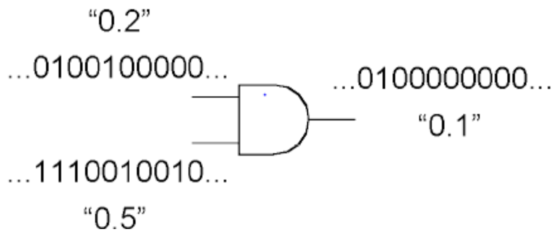

Figure.1. Multiplication of stochastic streams using and gate.

The simple structure for stochastic is of high importance to decode LDPC codes. The stochastic decoding scheme involves probabilities from channel to be transformed to stochastic streams..The decoding procedure is initiated by exchange of bits between stochastic variable node and parity check node. Let the input probabilities be denoted by Pa and Pb. Here Pa=probability of (a =1) and Pb =probability of(b=1) . The equation for operation of variable node is given by equation (1).

$$P_c = \frac{P_a P_b}{P_a P_b + (1-P_a)(1-P_b)}. \quad (1)$$

The equation for the degree 3 (*dv*=3) variable node operation. Similarly the equation for degree 3 (*dc*=3) parity check operations for input nodes is given by equation (2)

$$P_c = P_a(1-P_b) + P_b(1-P_a). \quad (2)$$

In case of inequality of input bits a and b, previous output bits are used by stochastic variable node structure *i.e. ci=ci-1* .This condition for variable node is referred to as hold state. The decoding operations in stochastic structure proceeds by exchange of bits between parity check node and variable node. This exchange of bits is done along each edge of factor graph. Here the output is calculated at the end of each decoding cycle referred to as DC. In this architecture the variable node output is passed to saturating up-down counter. The saturating up-down counter increments if it gets 1 as the input and it decrements after receiving 0 as the input from the output of variable node. The purpose of using saturating up down counter here is that it stops incrementing/decrementing if all the bits have been received .The hard decision is applied to contents of the counter after fix number of decoding cycle to determine the decode word. It is achieved by use of sign bit of up-down counter. Here 1 sign bit which determines +1 symbol and 0 sign bit determines -1 symbol in a BPSK transmission

[13,14,15]. The routing congestion problem is reduced significantly along with simple architecture of variable node and parity check nodes using stochastic decoding scheme since bits represents an edge in factor graph. The variable node structure of degree 3 and parity check node of degree 3 is given in Fig 2 A and 2B

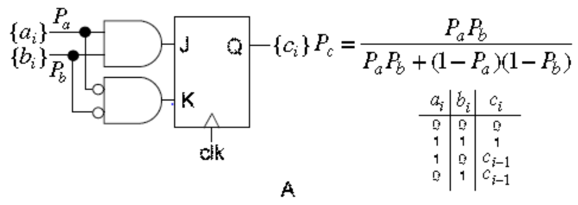

Fig 2.A The variable node structure of degree 3 using JK flip-flop.

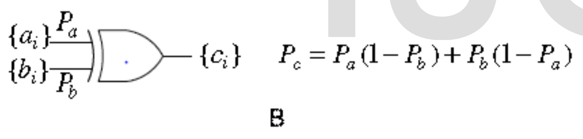

Figure.2.B The structure of degree 3 parity check node.

The disadvantage of variable node structure shown in fig 2A is that it was successful in decoding only short length codes. They also faced severe issues like latching (lock-up) issues of stochastic nodes due to sensitivity of stochastic decoders to the level of switching activity. The above problems can lead to severe degradation of BER decoding performance of LDPC. The above problem of lockup could be solved by increasing switching activity. The problem of degradation in BER performance can be solved by using noise dependant scaling [16] and inclusive of edge memories .This paper presents an implementation of fully parallel (1024,512) LDPC decoder having degree 3 variable node (*dv*=3) and degree 6 (*dc*) parity check node. Here variable node and parity check node are 2 decoding structures for (1024,512) LDPC codes using stochastic decoding scheme .

IV. STOCHASTIC DECODING ARCHITECTURE

The stochastic decoding scheme has variable node and parity check node as decoding architecture. Variable node and edge memories shown in Fig 3.1

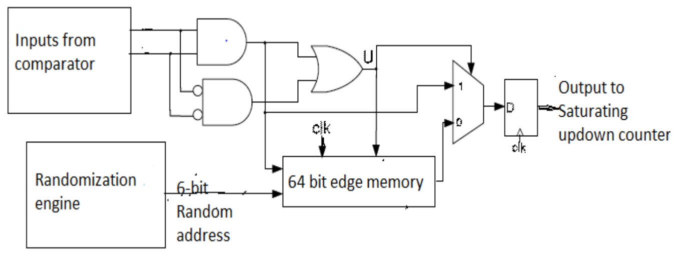

Figure . 3.1. Degree 3 variable node architecture.

$$\text{Pc (n)}= \sum_{i=1}^{n} [((\mathrm{P}a_{i.}.\mathrm{P}b_{i.} + \overline{P}_{a_i}.\overline{P}_{b_i})] .\bar{\text{u}} \qquad (3)$$

The equation for variable node architecture is given by equation 3. The input to the variable node architecture is given by generating stochastic streams [16]. The stochastic streams are generated by comparing the bits generated by AWGN with random bits. The probability is taken as 1 if the bit received by AWGN is greater than random bit considered. The stochastic bits thus generated are given as inputs (a_1,a_2) .Fig 3.1 to the variable node structure. Edge memories play a very important part in variable node decoding structure to avoid lock-up problem. Edge memories acts like the memories assigned to shit registers and are given to M bit shift registers. Each edge memory is used by each variable node. In previously proposed work on stochastic decoding architecture [16] edge memories were updated with the reception of 1.The bits in EM are selected by the address generated by Randomization engine. Randomization engine consist of ten 16-bit linear feedback shift register to generate 6 bit address. This paper displays the update of edge memories frequently after receiving the bits, thus reducing the problem of lock up to a great extent. We use 64 bit edge memories assigned to 64 bit shift registers for implementing(1024,512) LDPC decoder.64 bit EM are generated using four 4-input LUTs. The 64 bit EM is created by cascading of four 4 input LUT.

V. IMPLEMENTATION AND PERFORMANCE RESULTS

The implementation result shown in below Table 1 summarizes the number of slices required for the implementation of LDPC decoder. The number of slices consumed in individual module is given in Table 2, 3, 4.The efficient implementation of LDPC decoder can be done by successfully implementing the edge memories in the decoding architecture. The number of slices consumed by each individual module as well total decoder module determines the area that will be consumed in the total decoding architecture. The area consumptions of interleave is not considered since the interleave consists of only routing congestions. The whole area of decoder includes the area of interleave. The total number of slices utilized by decoder is total sum of all slices utilized in decoder. This paper displays reduced number of slices in comparison to its previous papers. The below mentioned results shows that percentage of LUTs utilized are very much less (approximately equal to 1%).The performance of stochastic decoder is very much close to floating point. Fig 4.1 and 4.2 shows the Technology schematic of variable node and check node architecture of decoder.

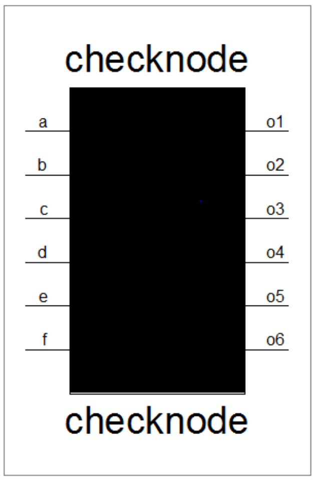

Figure 4.2 Complete decode architecture for degree 6 parity check node using Xilinx vertex-

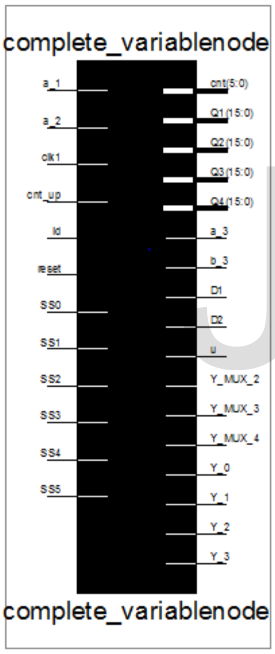

Figure 4.1 Complete decode architecture for degree 3 variable node Xilinx vertex-4

TABLE 1 OVER ALL CONSUMPTION OF SLICES

Modules	Slices consumed in this paper	Slices consumed in[16]
64 bit EM	60 0ut of 5472, 1% utilized	-
Degree 6 check-node	1 out of 5472 0% utilized	5
Degree 3 variable node	61 out of 5472 1% utilized	10 for 3 EM

TABLE 2 IMPLEMENTATION RESULT FOR 64 BIT EM

64 BIT EDGE MEMORY	SLICES CONSUMED
Slices consumed:	60 of 5472 1%
Utilization of Slice Flip Flops:	73 of 10944 0%
Utilization of 4 input LUTs:	40 of 10944 0%
IOs utilized:	85
Number of bonded IOBs:	85 of 240 35%

TABLE 3 IMPLEMENTATION RESULT FOR CHECK NODE OF DC=6

Check node	Slices consumed
Utilization of Slices:	0 of 5472 0%
Utilization of IOs:	5
Utilization of bonded IOBs:	5 of 240 2%

IOB Flip Flops:	1

TABLE 4 IMPLEMENTATION RESULT FOR VARIABLE NODE OF DV=6

Complete_Variable node	Slices consumed
Utilization of Slices:	61 of 5472 1%
Utilization of Slice Flip Flops:	76 of 10944 0%
Utilization of 4 input LUTs:	43 of 10944 0%
Utilization of IOs:	94
Utilization of bonded IOBs:	92 of 240 35%

VI. CONCLUSIONS

Thus the decoding architecture has been implemented using FPGA. Xilinx vertex 4 device. The paper discussed the different components used in construction of LDPC decoder and the number of slices used in constructing them. The decoding scheme used in this paper utilizes minimum number of slices approximately. The total number slice consumed by Variable node decoder is 121 out of 16416 slices which is 1% of the total available slices.

VII. REFERENCES

1] R. G. Gallagher, "*Low Density Parity Check Codes*, Cambridge, MA: MIT Press, 1963".

[2] F. Kschischang, B. Frey, and H. Lonelier, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, Feb 2001

[3] Anastasopoulos, "A comparison between the sum productand the min-sum iterative detection algorithms based on density evolution," in *Proceedings of the IEEE Global Telecommunications Conference*, Nov. 2001, vol. 2, pp. 1021–1025.

[4]F. Guilloud, E. Boutillon, and J.-L. Danger, "λ-min decoding algorithm of regular and irregular LDP Codes," in *Proceedings of 3rd International Symposium on Turbo Codes (ISTC 03)*, Brest, France, 1-5 Sept.2003, pp. 451–454

[5] B. Gaines, *Advances in Information Systems Science*, chapter 2, pp.37–172,Plenum,NewYork,1969.

[6] A. Rapley, C. Winsted, V. Gaudet, and C. Schlegel,"Stochastic iterative decoding on factor graphs," in *Proceedings. of the 3rd Int. Symp. on Turbo Codes and Related Topics*, Brest, France, Sept. 2003, pp. 507–510.

[7] A. Anastasopoulos, "A comparison between the sum-product and the min-sum iterative detection algorithms based on density evolution," in *Proceedings of the IEEE Global Telecommunications Conference*, Nov. 2001,vol. 2, pp. 1021–1025.

[8] 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

[9] A.A.Dinu,,M.N. Cirstea, and M. McCormick," Stochastic implementation of motor controllers," in *Proceedings. of the IEEE Int. Symp. on Industrial Electronics*, July 2002, pp. 639–644.

[10] C. Howland and A. Blanksby, "Parallel decoding architectures for lowdensity parity check codes," in *Proc. of 2001 IEEE Int. Symp. on Circuits and Systems*, Sydney, May 2001.

[11] A. J. Blanksby and C. J. Howland, "A 690-mW 1-Gb/s 1024-b, rate-1/2 low-density parity-check code decoder," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 3, pp. 404–412, March 2002.

[12] S. Sharifi Tehrani, S. Mannor, and W. J. Gross, "Surveyor stochastic computation on factor graphs," to appearing the *Proc. of the 37th International Symposiumon Multiple-Valued Logic*, May 2007.

[13] A. Rapley, C. Winstead, V. Gaudet, and C. Schlegel,"Stochastic iterative decoding on factor graphs," in *Proc. of the 3rd Int. Symp. on Turbo Codes and Related Topics*, Brest, France, Sept. 2003, pp. 507–510

[14] W. J. Gross, V. Gaudet, and A. Milner, "Stochastic implementation of LDPC decoders," in *the 39th Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 2005.

[15] S. Sharifi Tehrani, W. J. Gross, and S. Mannor,"Stochastic decoding of LDPC code," *IEEE Communications Letters*, vol. 10, no. 10, pp. 716–718, Oct. 2006.

[16] Sharifi Tehrani, Shie Mannor and Warren J. Grossb,"An area efficient FPGA based architecture for fully parallel stochastic LDPC decoding" *in Proceedings of IEEE 2007 Global Telecommunications conference ,Nov 2007,vol.3.pp.*1021-1034.

b