Proceedings of the 5th WSEAS Int. Conf. on Instrumentation, Measurement, Circuits and Systems, Hangzhou, China, April 16-18, 2006 (pp230-234)

Self Reliability based Weighted Bit-Flipping Decoding for Low-density Parity-check Codes

JIN SHA, MINGLUN GAO, ZHONGJIN ZHANG,LI LI

Institute of VLSI design
Key Laboratory of Advanced Photonic and Electronic Materials
Nanjing University
Nanjing, China, 210093

ZHONGFENG WANG

School of EECS
Oregon State University
Corvallis, OR 97331-5501,USA

Abstract - Low Density Parity Check (LDPC) codes can be decoded in various ways, namely the Bit-Flipping (BF) algorithm, the Weighted BF algorithm (WBF), the Belief Propagation (BP) algorithm and so on. These algorithms provide a wide range of tradeoffs among decoding complexity, decoding speed, and error rate performance. In this paper, a novel self-reliability-based weighted bit-flipping decoding scheme for low-density parity-check codes is proposed. Improvement in performance is observed in comparison with the modified weighted bit-flipping decoding scheme, and the decoding complexity can be significantly reduced as well.

Key-Words: - Low-density parity-check (LDPC) codes, decoding algorithm, bit flipping, self reliability

1. Introduction

Low-density parity-check (LDPC) codes, originally introduced by Gallager [1], have attracted a great deal of research interest in the information theory community. There have been a wide variety of iterative algorithms proposed for the decoding of LDPC codes [1–4]. Each algorithm provides a different performance/complexity tradeoff. Bit-flipping based LDPC decoding algorithms, such as weighted bit-flipping (WBF) [2] algorithm, modified weighted bit-flipping (MWBF) algorithm [3] and reliability ratio based weighted bit-flipping decoding algorithm [4] are considered as good trade-off between error-correcting performance and decoding complexity, comparing with belief-propagation (BP) decoding algorithm. In this letter, a novel Self-Reliability-based Weighted Bit-Flipping (SRWBF) decoding algorithm is proposed. This decoding method performs much better in contrast with MWBF, and has significantly lower computation complexity in contrast with other BF based algorithms.

In the following, a brief overview of these bit-flipping based algorithms is explored and SRWBF algorithm is presented. Then gives the simulation result and complexity analysis.

2. Low Density Parity Check Codes

Assume an LDPC code with the parity check matrix **H** = $[H_{mn}]$ that contains mostly zeros and only a small number of ones. The parity check matrix **H** has N columns and M rows. In this paper, only regular LDPC codes are considered. We describe an LDPC code by (N,K,d_v,d_c) LDPC code where N, K denote the length of the codeword and the information bits, respectively, and d_v, d_c denote the column and row weight,, respectively. The code rate is given by K/N.

The structure of LDPC codes is well represented by Tanner graph as illustrated in Fig.1. This graph is constituted by two kinds of nodes, namely the variable nodes and the check nodes. Consequently, (N,K) LDPC code has N variable nodes and M check nodes. An edge exists between a variable node and a check node if and only if there is a “1” in the corresponding entry in the parity check matrix. A (N,K) (j,k) LDPC code has a Tanner graph in which all the variable nodes have degree j and all the check nodes have degree k.

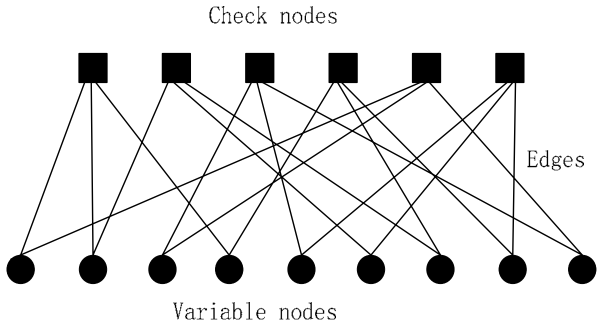

Fig.1 a Tanner graph example

3. Bit Flipping based algorithms

In the following, assume the received real value sequence is $\mathbf{y} = (y_1, y_2, \ldots, y_n)$. Let the corresponding binary hard decision sequence be $\mathbf{z} = (z_1, z_2, \ldots, z_n)$.

3.1 WBF Algorithm

The standard WBF algorithm [2] initially calculates the syndrome vector and finds the most unreliable message node participating in each individual check:

$$s_m = \sum_{n=1}^{N} z_m H_{mn} \tag{1}$$

$$y_m^{\min} = \min_{n \in N(m)} |y_n| \tag{2}$$

Here, $|y_n|$ denotes the absolute value of the nth message node's soft value, while y_m^{min} is the lowest magnitude of all message nodes participating in the mth check. $N(m)$ denotes the set of variable nodes that participate in mth check node.

For each variable node, compute

$$E_n = \sum_{m \in M(n)} (2s_m - 1) y_m^{\min} \tag{3}$$

Where, $M(n)$ is the set of check nodes in which nth variable node participates. Flip the bit with the largest E_n. Then, update the syndrome vector and do iterations until $\boldsymbol{s} = \boldsymbol{0}$ or iteration limit is reached.

3.2 MWBF Algorithm

As seen in (3), the WBF algorithm only considers the check-node based information during the evaluation of the error-term E_n. The MWBF algorithm [3] improves the decoding performance by taking into consideration the intrinsic message for each bit in the E_n computation.

$$E_n = \sum_{m \in M(n)} (2s_m - 1) y_m^{\min} - \alpha |y_n| \tag{4}$$

In MWBF, α is used to denote the weight coefficient for the soft value $|y_n|$. A drawback of the MWBF algorithm is that, for LDPC codes having different column weights, or operating at different SNRs, a different value for α has to be found specifically. In a recent publication [5], a further improvement of MWBF algorithm and a theoretical method for α calculation is proposed. However, the computation complexity will be increased and the decision of α is still a trouble for practices.

3.3 RRWBF Algorithm

RRWBF performs best among these BF-based algorithms. In addition, it eliminates the requirement of α parameter decision. In [6], the RRWBF algorithm is changed to a simplified version. Equation (5) gives its method for error term evaluation.

$$E_n = \frac{1}{|y_n|} \sum_{m \in M(n)} (2s_m - 1) T_m \tag{5}$$

$$T_m = \sum_{n \in N(m)} |y_n| \tag{6}$$

All these three algorithms stated above have one thing in common, the computation of error term E_n needs to get magnitude information from their neighbor variable nodes. This means that each edge in the Tanner graph must be mapped to $2*q$ wires to transmit the information (q is the quantization bits). The total number of wires should be $2*N*d_v$, and it is the same as the fully parallel soft LDPC decoder architecture. In [7], a rate 1/2, 1024 bit fully parallel LDPC decoder is implemented, where logic density is reduced to 50% to accommodate the complexity of the interconnect fabric. For longer code, the routing complexity will be more intolerable. LDPC codes achieve outstanding performance only with large code word lengths. Thus, the routing problem remains with these BF-based algorithms.

Proceedings of the 5th WSEAS Int. Conf. on Instrumentation, Measurement, Circuits and Systems, Hangzhou, China, April 16-18, 2006 (pp230-234)

4. SRWBF Algorithm

In the WBF or the MWBF algorithm, the reliability information of a check equation is based on the minimum absolute value of the received symbols participating in the check. In the RRWBF algorithm, the reliability information of a check equation is based on the sum value of the received symbols participating in the check. In evaluating the reliability of a check node, neither of these two methods is precise. The authors of [5] give a more precise method by excluding the bit itself in finding the minimum absolute value.

Two kinds of information need to be considered in evaluating the error term for each bit: the information from check node and the intrinsic information. In fact, without the message passing, the information each variable node can receive from their neighbors is very limited. It is noticed that the $2s_m-1$ term may bring enough information from check nodes. Hence, the self reliability $|y_n|$ should be considered more in contrast to the reliability of the neighbor variable nodes participating in same check nodes. In consideration of this, a new self reliability ratio based weighted bit-flipping decoding algorithm is introduced. The new error term used is:

$$E_n = \sum_{m \in M(n)} (2s_m - 1)/|y_n| \tag{5}$$

The ignorance of the reliability of neighbor variable nodes can greatly reduce the decoding complexity. This will be explained more clearly in section five.

The steps of SRWBF algorithm are listed below:

Initial: Receive the block $\boldsymbol{y}$, and get the hard-decision sequence $\boldsymbol{z}$

Step 1: Compute the syndrome vector $\boldsymbol{s}$ according to equ.(1)

Step 2: Compute E_n according to equ.(5)

Step 3: Flip the bit with the biggest E_n, and update the syndrome vector.

Step 4: Return to step 2, do iterations until $\boldsymbol{s} = \boldsymbol{0}$ or iteration limit is reached.

5. Simulation results

The achievable performance of the proposed algorithm is characterized as below. The algorithm will be benchmarked against BP, WBF, MWBF algorithms. These schemes will be invoked for decoding a (1008,504) regular (3,6) Gallager code [5] and the (2048,1723) RS-based (6,32) regular LDPC code [6] using a BPSK modulation scheme communicating over an AWGN channel. The second code is one of the only two options proposed for 10GBase-T Ethernet standard.

For each code, a maximum number of N/10 iterations are allowed. Hence, for codes having different length, 10% of the coded bits have the chance to be corrected. Iteration operation will be terminated as long as the resultant syndrome vector becomes an all-zero vector or when the maximum affordable iteration times has been exhausted.

Fig.1 and Fig.2 show that for both codes, SRWBF provides a considerable improvement over the conventional MWBF in error performance. For the RS-based LDPC code, at BER 10^{-5}, the required SNR can be reduced by 0.25dB.

Although, for the sake of brevity, only simulation results for two LDPC codes are presented here, more codes have been simulated and similar improvements have been observed.

Some simulations for finite geometry (FG) LDPC codes [2] are also conducted. However, results are not satisfied. SRWBF performs better than WBF but worse than MWBF. The reason may be too large number of checks is involved in the FG codes. For the (273,191) PG-LDPC code, each variable node can get information from 17 check nodes. The performance of SRWBF suffers from the ignorance of all these check node information. So, SRWBF does not perform well with finite geometry LDPC codes.

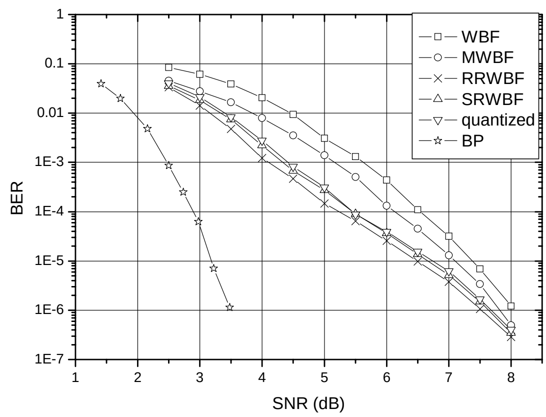

Fig.1. BER performance of (1008,504) (3,6) regular LDPC codes decoded by WBF, MWBF and proposed algorithms

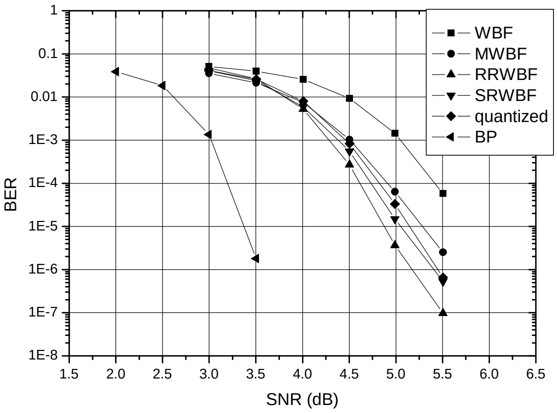

Fig.2. BER performance of (2048,1723) regular (6,32) RS-based LDPC codes decoded by BP，WBF, MWBF and proposed algorithms

6. Complexity and finite precision analysis

The proposed self reliability based LDPC decoding algorithm is more hardware friendly, comparing with other BF-based algorithms.

Firstly, the number of the wires needed by SRWBF is analyzed. It can be seen in equ.5 that, to compute the error term E_n, each variable node needs two input information: s_m and $|y_n|$. $|y_n|$ is the magnitude of the received value, which is stored inside each variable node. s_m is received from its neighbor check nodes and they are only one bit wide. Similarly, each check node only needs the sign of every variable node to compute s_m (equ.1), which is also one bit wide. Then, each edge in the Tanner graph maps into one pair of wires only. On the contrary, other BF based algorithms usually need to communicate the magnitude information between variable nodes and check nodes. So, it can save $(q-1)/q$ (q is the quantization bits) wires in contrast to the traditional BF-based algorithms. This can totally eliminate the routing problem for implementing decoders with large codeword length.

Secondly, the computation complexity is analyzed. At first glance, the divider imposed in each variable node will make the decoder design complex. However, some modifications can be applied to equation (5) to make it hardware intimate:

$$t_n = 1/y_n \tag{6}$$

$$E_n = \sum_{m \in M(n)} (2s_m - 1) \times t_n \tag{7}$$

Note that the division operation in (6) is performed to the received value for each bit. In fact, the received sequence $\boldsymbol{y}$ comes into the decoder serially bit by bit. Therefore only one divider is needed at the first receiving of y. t_n is stored in every variable node instead of y_n. This divider can be implemented by using a LUT, which is of very high speed and area saving for small input word length.

The finite word length effect on the performance of SRWBF is conducted. Through simulation, a four bits quantization scheme is suggested considering the tradeoff between hardware complexity and decoding performance, one bit for the sign, one bit for the integer part and two for fraction part. The simulation result of the four bit quantized SRWBF algorithm is also shown in Fig.1 and Fig.2. For the Gallager code with N=1008, the quantization induces little performance loss. And for the RS-based LDPC code, the loss is less than 0.2dB at BER=10^{-6}. So, the SRWBF is robust against quantization error.

The multiplication in Equation (7) can also be very simple. The ($2s_m$ -1) term has very limited possibilities. For example, for an LDPC code with d_v = 3, this term can only be 1 or 3. Or, for an LDPC code with d_v = 6 (RS-based), this term can only be 2, 4 or 6. Generally, a 2 bits ∗ 5 bits multiplier is enough to do the (7) operation in each variable node.

As a comparison, other algorithms usually need more operations in calculating $\boldsymbol{y_{min}}$ or complex multiplication which is very hardware hungry and time-consuming.

Table1. logic function usage per iteration in each variable node(vn) and each check node(cn) for three bit-flipping based decoding algorithms

	MWBF		RRWBF		SRWBF	
OPER	CN	VN	CN	VN	CN	VN
ADD	d_c -1	d_v	d_c -1	d_c -1	-	-
MUL	-	1	-	-	-	1(simple)
DIV	-	-	-	1	-	-
XOR	1	-	1	-	1	-

Table 1 shows the logic function usage per iteration in each variable node and check node for the BF-based decoding algorithms discussed in this paper. It shows that SRWBF has significantly low complexity.

7. Conclusion

In this paper, a novel self reliability based weighted bit-flipping algorithm is proposed. This decoding method can outperform the modified weighted bit-flipping algorithm. In addition, the SRWBF is robust against quantization error and has significantly lower computation complexity in contrast to other BF based algorithms, which is a valuable advantage.

6 Acknowledgments

The work presented in this paper was supported by the Foundation of High-Tech of Jiangsu Province of China under Grant No.BG2005030; the National Nature Science Foundation of China under Grant No.90307011.

References:

[1] R. G. Gallager, “Low density parity check codes,” IRE Trans. Info. Theory, vol. IT-8, pp. 21-28, 1962.

[2] Kou, Y., Lin, S., and Fossorier, M.: ‘Low-density parity-check codes based on finite geometries: a rediscovery and new results’, IEEE Trans. Inf. Theory, 2001, 47, pp. 2711–2736

[3] Zhang, J., and Fossorier, M.P.C.: ‘A modified weighted bit-flipping decoding of low-density parity-check codes’, IEEE Commun. Lett., 2004, 8, (3), pp. 165–167

[4] Guo, F., and Hanzo, L.: ‘Reliability ratio based weighted bit-flipping decoding for low-density parity-check codes’, Electron. Lett., 2004, 40, (21), pp. 1356–1358

[5] Ming Jiang, Chunming Zhao, Zhihua Shi, and Yu Chen: ‘An improvement on the Modified Weighted Bit Flipping Decoding Algorithm for LDPC codes’ IEEE Commun. Lett., vol.9, No.9, pp. 814–816. September, 2005

[6] C.-H.Lee, and W.Wolf : ‘Implementation-efficient reliability ratio based weighted bit-flipping decoding for LDPC codes’, Electron. Lett, 2005, 41,(13),pp. 755-757

[7] A.Blanksby and C.J.Howland, “A 220mW 1-Gbit/s 1024-Bit Rate-1/2 Low Density Parity Check Code Decoder,” Proc IEEE CICC, Las VEGAS, NV, USA, pp.293-6, May 2001.

[8] D.J.C.Mackay. *Encyclopedia of Sparse Graph Codes* [Online]. Available: http://www.inference.phy.cam.ac.uk/mackay/codes/data.html.

[9] 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.