Journal of Theoretical and Applied Information Technology
31st March 2013. Vol. 49 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A REDUCED-COMPLEXITY LDPC DECODING ALGORITHM WITH CHEBYSHEV POLYNOMIAL FITTING

[1]SHAOBO SHI, YUE QI, LEI LI, MIN YAO, QIN WANG

[1]PhD,School of Computer and communication engineering, University of Science and Technology Beijing, Beijing 100083, China

E-mail: [1]shaobo1985@gmail.com

ABSTRACT

BP decoding algorithm is a high-performance low-density parity-check (LDPC) code decoding algorithm, but because of its high complexity, it can't be applied to the high-speed communications systems. So in order to further reduce the implementation complexity with the minimum affection to the system performance, , we proposed a BP algorithm based on Chebyshev polynomial fitting for its good approximation performance in this paper. This method can transform the complicated exponential function into polynomial with the error at 10^{-3} level in the process of decoding , which can reduce the consumption of memory resources considerably. At the same time, a Chebyshev unit is proposed in the process of implementation with less multiplier; we implemented a semi-parallel LDPC decoder with this method on Altera CycloneII FPGA. The simulation results show that the degradation of this method is only 0.4dB compared with conventional BP algorithm.

Keywords: *low-density parity-check (LDPC) code;BP algorithm;Chebyshev Polynomial Fitting;*

1. INTRODUCTION

Gallager proposed LDPC(Low Density Parity Check) code [1]. The performance of this code is very close to Shannon limit, which has only less than 0.1dB difference [2]. For the better performance in decoding, LDPC codes are widely used in various current communications systems. Such as DVB-S, WLAN and WiMAX [3] and other popular wireless communication system.

Belief Propagation decoding algorithm is considered as an optimal iterative decoding algorithm [1]. However, a large amount of multiplication and exponentiation lead to high complexity of implementation in BP algorithm, and it is not suitable for high-speed communication system. Currently, there are two main methods to reduce the implementation complexity in LDPC code decoding. The first one is to transform the complex formula in the original BP algorithm intoa easy to implement one, thereby reduce the complexity of implementation. The famous Min-Sum algorithm[4] is the typical method, which adopt the operation of simple sign function and addition to replace the multiplication and exponentiation in original BP algorithm. But, compared with the traditional BP algorithm, the performance degradation of Min-Sum is about 1 dB[5][6]. While, the other method is the Look-up-table based on ROM[7]. This method has a comprehensive range of application for its simple structure and nice accuracy. However, there is a limit in improving performance, since to get the better accuracy need much more memory. Therefore, we should find a reasonable method that can provide better performance than Min-Sum and use lesser memory than Look-up-table.

This paper proposed a Reduced-Complexity BP algorithm based on Chebyshev Polynomial Fitting, we called it CPF-BP. In this method the exponential operations are transformed into a high proximate polynomial in the conventional BP algorithm. We use only handful of multipliers, shifters and adders to implement this method in the implementation, so this is an algorithm that is suitable for VLSI implementation.t. and a simulation and implementation of encoding/decoding with three-order polynomial CPF-BP are showed in this paper.

In section I the LDPC code and BP algorithm are introduced. In section Ⅱ, the CPF-BP algorithm is presented And in section Ⅲ, the hardware implementation is provided. The section Ⅳ and Ⅴ will show the simulation result of this method and the conclusion.

2. LDPC CODE AND BP ALGORITHM

Journal of Theoretical and Applied Information Technology
31st March 2013. Vol. 49 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The input to the ANN is the value of exponent of reactive power load-voltage characteristic (n_q) and the output is the desired proportional gain (K_P) and integral gain (K_I) parameters of the SVC. Normalized values of n_q are fed as the input to the ANN the normalized values of outputs are converted into the actual value. The process of

2.1 LDPC Code

LDPC code is one kind of linear block code, and it can be presented by Tanner graph or check matrix H, both of them is equivalent. As showed in figure1 and figure2:

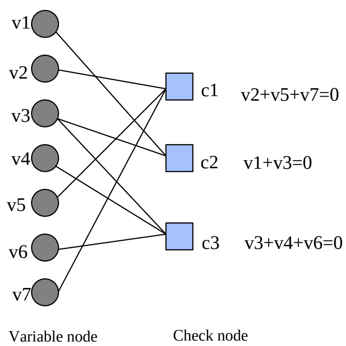

Figure1: Tanner graph of LDPC encoder

$$A=\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Figure2: The check matrix corresponding to Tanner graph

In figure1, the left nodes represent the variable nodes V, corresponding to the column of check matrix. The right nodes represent the check nodes C, corresponding to the row of check matrix. When there is a link between i-th V and j-th C in Tanner graph, the value of [j,i] in check matrix is 1.

The BP algorithm of LDPC code is one decoding algorithm that based on message propagation and iteration of Tanner graph. The basic mathematic theory is:

Bayes Principle:

$$P(X\mid Y)=P(Y\mid X)*P(X)/P(Y) \qquad (1)$$

Judgment criteria:

$$\begin{aligned} & if: P(X=0\mid Y) >= P(X=1\mid Y) \\ & \quad then: X=0 \\ & else: \\ & \quad then: X=1 \end{aligned} \qquad (2)$$

2.2 BP algorithm

In this paper we assumed the accepted vector is y, the length of y is N, the Check matrix H is a $M\times N$, and defined these variable:

p_n^x is the probabilistic values when the value of Nth bit is x; $q_{m,n}^x$ is the probabilistic values when the value of Nth bit is x, but the probabilistic values is got from the check function except Mth one; $r_{m,n}^x$ is the probabilistic values when the value of Nth bit is x, and this bit must meet the Mth check function;

The basic process of conventional BP algorithm is described as following:

Initialization: calculate the prior probability of every variable node in the initial channel.

We have this formula:

$$p_n^1=\frac{1}{1+e^{-2y_n/\sigma^2}} \qquad (3)$$

$p_n^0=1-p_n^1$, for every node that $H_{m,n}=1$ in the check matrix, $q_{m,n}^0=p_n^0, q_{m,n}^1=p_n^1$

Horizontal step: get the probability of every check node from the outcome of step1.

For every (m,n) in the check matrix:

$$\Delta r_{m,n}=\prod_{n'\in L(m)\backslash n}(q_{m,n'}^0-q_{m,n'}^1)$$

(4)

And, $r_{m,n}^0=(1+\Delta r_{m,n})/2, r_{m,n}^1=(1-\Delta r_{m,n})/2$

Vertical step: get the posterior probability from the values of step3.

For every variable node:

$$q_{m,n}^0=\alpha p_n^0\prod_{m'\in M(n)\backslash m}r_{m',n}^0,\quad q_{m,n}^1=\alpha p_n^1\prod_{m'\in M(n)\backslash m}r_{m',n}^1 \qquad (5)$$

α is the normalization coefficient of equation $q_{m,n}^0+q_{m,n}^1=1$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Soft Decision: calculate the posterior probability of variable node:

$$q_n^0 = \partial p_n^0 \prod_{m \in M(n)} r_{m,n}^0, \qquad q_n^1 = \partial p_n^1 \prod_{m \in M(n)} r_{m,n}^1 \tag{6}$$

∂ is the normalization coefficient of equation $q_n^0 + q_n^1 = 1$

Now, according to the Judgment criteria we can get the value of the every bit by these:

$$\begin{aligned} &if: q_n^0 > 0.5 \\ &\quad then: x = 0 \\ &else: \\ &\quad then: x = 1 \end{aligned} \tag{7}$$

If the number of iteration not achieves the maximum, then the algorithm will repeat the step 2 to step 4 to get the decoding result.

From the formula (3), we can see that the exponential function can bring too much computing burden to iteration process, and is hard to implement. In the practical applications, the solutions of this problem are the Look-up-table and function approximation. As mentioned before, Look-up-table has its boundedness in practical project. So the function approximation is another important solution for this problem, and we adopt the Taylor series usually. But Taylor series is based on design point, and there will be a increasing error when the data is far away the design point. So in this paper we adopt the method of chebyshev polynomial fitting, and this is a new and effective method of function approximation test by our simulation.

2.3 Chebyshev Polynomials

In the problem of Continuous function with a polynomial approximation, when $f(x) \in C[a,b]$,

it is impossible to make $P_n(x) = a_0 + a_1 x + ... + a_n x^n$, $a_0, a_1, ..., a_n$ are and real number, equal to $f(x)$ absolutely, but the polynomial $P_n^*(x)$ can be found to make the difference $\| f - P_n^* \|_\infty = \min_{a \le x \le b} | f(x) - P_n^*(x) |$, this is called the best uniform approximation.

If $f(x) \in C[-1,1]$, through Fourier series expansion:

$$f(x) \sim \frac{C_0^*}{2} + \sum_{k=1}^{\infty} C_k^* T_k(x) \tag{8}$$

Here

$$C_k^* = \frac{2}{\pi} \int_{-1}^{1} \frac{f(x) T_k(x)}{\sqrt{1-x^2}} dx (k = 0,1,...). \tag{9}$$

$$T_k(x) = \cos(k \arccos x), |x| \le 1$$
(10)

Fitting functhion replace the initial probability function in BP algorithm

In this part, we will introduce how to fitting the probability function with our method.

Suppose the x is the input data of decoder, and SNR=1, σ can be got through formula: $\sigma = \sqrt{10^{-SNR/10}}$, so $x \in [-2,2]$ with noise can be got. Actually, in our test system, the received bits almost are [2,-2], under the AWGN. We consider the situation that the x is in [0,2], and in this process of fitting we will use function replacement.

Suppose $f(x) = \frac{1}{1 + e^{-2x/\delta^2}}$, and then assign $x = t + 1 (-1 \le t \le 1)$, $x \in [0,2]$

So, we have $f(t) = f(x-1)$, $t \in [-1,1]$, and assign $t = \cos\theta, \theta \in [0, \pi]$.

Through the formula (8), (9), and (10), and combine the formula above, we have this reduction:

$$\begin{cases} C_k^* = \frac{2}{\pi} \int_0^{\pi} f(\cos\theta) \cos k\theta d\theta (k = 1,...) \Rightarrow \\ C_k = \frac{2}{\pi} \int_0^{\pi} \frac{1}{1 + e^{-2*(\cos a - 1)/\sigma^2}} * \cos(ka) da, k = 1,2,...n \end{cases} \tag{11}$$

Then, we can calculate the Chebyshev coefficients C_k, and get the fitting function we want:

$$f(t) = \frac{C_0}{2} + C_1 * \cos(\arccos t) + C_2 * \cos(2 * \arccos t) + C_3 * \cos(3 * \arccos t)$$
(12)

We replace the t through $t = x - 1$ in formula (12), and get the expression of the x:

$$P_n = C_0' + C_1' x + C_2' x^2 + C_3' x^3 \tag{13}$$

For the same principle, we can get another fitting function in the region $x \in [-2,0]$, and just pay attention to one point that we must assign $x = t - 1 (-1 \le t \le 1)$. So the expression of the x:

Journal of Theoretical and Applied Information Technology
31st March 2013. Vol. 49 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$P_n = C_0'' + C_1''x + C_2''x^2 + C_3''x^3 \quad (14)$$

From the figure3 we can see that there is a negligible error (10^{-3} level) between the chebyshev polynomial fitting function (CPF) --formula (13)(14) and the initial probability function (Primitive) --formula (3). So we can consider it is nearly equivalent, and can combine formula (13) and (14) to replace the complex initial probability function:

$$p_n^1 = \frac{1}{1+e^{-2y_n/\sigma^2}}$$

(15)

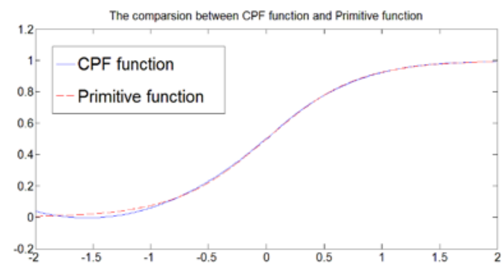

Figure3: Comparison between CPF and Primitive function.

3. THE IMPLEMENTATION OF BP ALGORITHM BASED ON CPF METHOD

In the process of BP decoding algorithm implement[8][9], we classifies nodes as Check Node Unit (CNU) and Variable Node Unit (VNU), horizontal and vertical processing treatment in decoding algorithm are respectively used to process the probability updating of Check Node Unit and Variable Node Unit, in the same time, this paper use Chebyshev polynomial to do initial processing.

In this paper, the overall block diagram of the decoder is shown as follows:

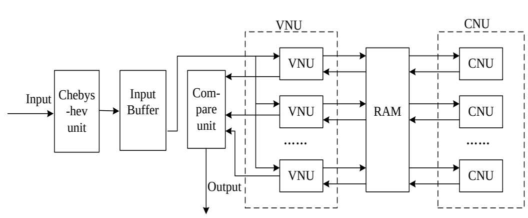

Figure 4: Architecture of Decoder

From above, when use $P_n = C_0' + C_1'x + C_2'x^2 + C_3'x^3$ to fit formula, implement of this formula, we need some multipliers, adders, and shifters. When the scope of x is determined, Chebyshev coefficients C_k are fixed. So in order to reduce the complexity of circuit design, as show in Figure5, we simplifies coefficients in formula with form of exponent2, then

$$P_n = C_0 + (2^{-a} + 2^{-b}) * x + (2^{-c} + 2^{-d}) * x^2 + (2^{-e} + 2^{-f}) * x^3,$$

in binary domain, multiplying with 2^{-n} can be replaced by shift operator, so we can use these computational component to implement the CPF function in decoding circuit.

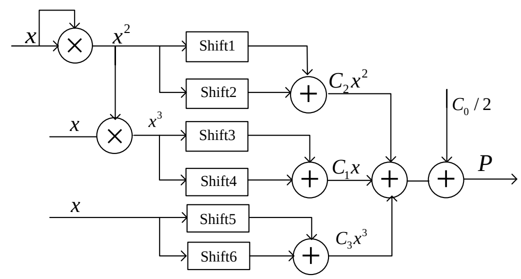

Figure 5: Architecture of CPF Unit

A. Hardware Implementation

To avoid enormous storage resources of parallel structure, this paper adopts semi-parallel structure, to instead nodes parallel computing by serial computing, and use RAM as buffer.

From the figure5, we can see that the new CPF unit added to the system, consumes 3 multiplier, 6 shifter, and 7 adder compared with the conventional BP algorithm. But CPF-BP decoding algorithm saved ROM resource which is used in lookup table method.

This paper uses Verilog hardware description language, based on Altera CycloneII series EP2C20F484C8, call its multiplier IP core, then to do synthesis, layout and experimental verification. Length of code is 2056 bits, the iteration number is 6. The FPGA synthesis results are shown as table1:

Table1:FPGA synthesis result

Logic resources（LEs）	Memory units（bits）	Maximum clock frequency（MHz）
10,918	172,544	55.9

4. SIMULATION

We simulate the LDPC code with conventional BP algorithm and CPF-BP algorithm respectively. Under the AWGN channel, the Packet length is

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2056bits, coding rate is 1/2, and with BPSK modulation.

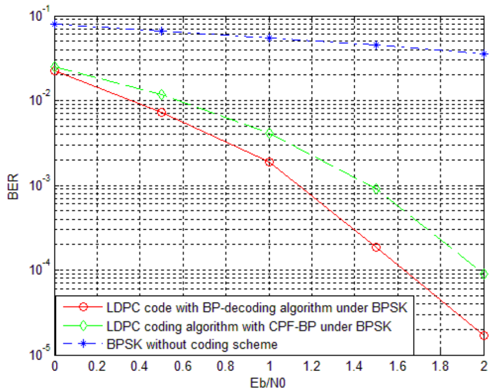

Figure 6 Performance comparison

As shown in figure 6, the CPF-BP algorithm obtains good performance of decoding. With the increased of value of Eb/N0, the BER of BP-CPF is not decreased significantly. When the level of BER is 10^{-4}, the degradation is just 0.4dB compared with conventional BP algorithm, and this is much better than the 1dB of Min-Sum algorithm in the same level of BER. The reason of this good performance is that the error that is between the initial probability function and the CPF function is controlled ideally. However, with the increased of iteration number, the degradation result caused by error cannot avoid.

5. CONCLUSION

In this paper, we proposed a reduced-complexity BP algorithm based on chebyshev polynomial fitting(CPF-BP) under the LDPC decoding background. We used this fitting function to instead the exponential function in order to reduce the complexity of implementation. From the result of simulation and FPGA synthesis, the degradation of this algorithm is 0.4dB, which can be negligible compared with conventional BP algorithm. And we use only 16 computational units to implement this complex function and save lots of memory space. In conclusion, the CPF-BP decoding algorithm has a better performance and low-complexity in VLSI implementation.

REFRENCES:

[1] GallagerRG.*low-densityparity-checkcodes*［M］.Cambridge, Mass : MITPress, 1963.

[2] T. J. Richardson, A. Shokrollahi, and R. Urbanke, “Design of provably good low-density parity-check codes,” *IEEE Trans. Inform. Theory*,Apr. 1999, submitted for publication.

[3] Digital Television Terrest rial Broadcasting Standard Special,Group. GB 2060022006 : Framing St ructure , Channel Coding and Modulation for Digital Television Terrest rial Broadcasting System [S] . Beijing : Chinese Standard Express , 2006

[4] E. Eleftheriou, T.Mittelholzer, and A. Dholakia, “A reduced-complexity decoding algorithm for low-density parity-check codes,” *Electron. Lett.*,Oct. 2000, submitted for publication.

[5] Anastasopoulos. A. “A comparison between the sum-product and the min-sum iterative detection algorithms based on density evolution”.*Global Telecommunications Conference, 2001. GLOBECOM* . IEEE Volume 2, Issue , 2001 Page(s):1021 - 1025 vol.2

[6] Bhattad. K, “LDPC Code Design for Min-Sum Based Decoding”. *Texas A\&M University, Technical Report* WCL-TR-07-102, 2006

[7] T. Theocharides, G. Link, “Implementing LDPC Decoding on Network-On-Chip”. *18th International Conference on VLSI Design*, 2005 Page(s):134 – 137

[8] Lee. WL and Wu. A, “VLSI implementation for low density parity check decoder”. *The 8th IEEE International Conference on Electronics, Circuits and Systems ICECS 2001*, 2002 Page(s):1223 - 1226

[9] Tong Zhang,"*Eficient VLSI Architectures for Error-Correcting Coding*", Ph.D thesis of University of Minnesota, 2002Page(s):92-111 chapter 4