



US 20090006924A1

(19) **United States**

(12) **Patent Application Publication**  
**Ho**

(10) **Pub. No.: US 2009/0006924 A1**

(43) **Pub. Date: Jan. 1, 2009**

(54) **FAST DECODING METHOD FOR LOW DENSITY PARITY CHECK (LDPC) CODE**

**Publication Classification**

(76) Inventor: **Chia-Lu Ho**, Jhongli City (TW)

(51) **Int. Cl.**  
**G06F 11/10** (2006.01)

Correspondence Address:

**HDSL**  
**P.O. BOX 220746**  
**CHANTILLY, VA 20153-0746 (US)**

(52) **U.S. Cl.** ..... **714/758; 714/E11.032**

(57) **ABSTRACT**

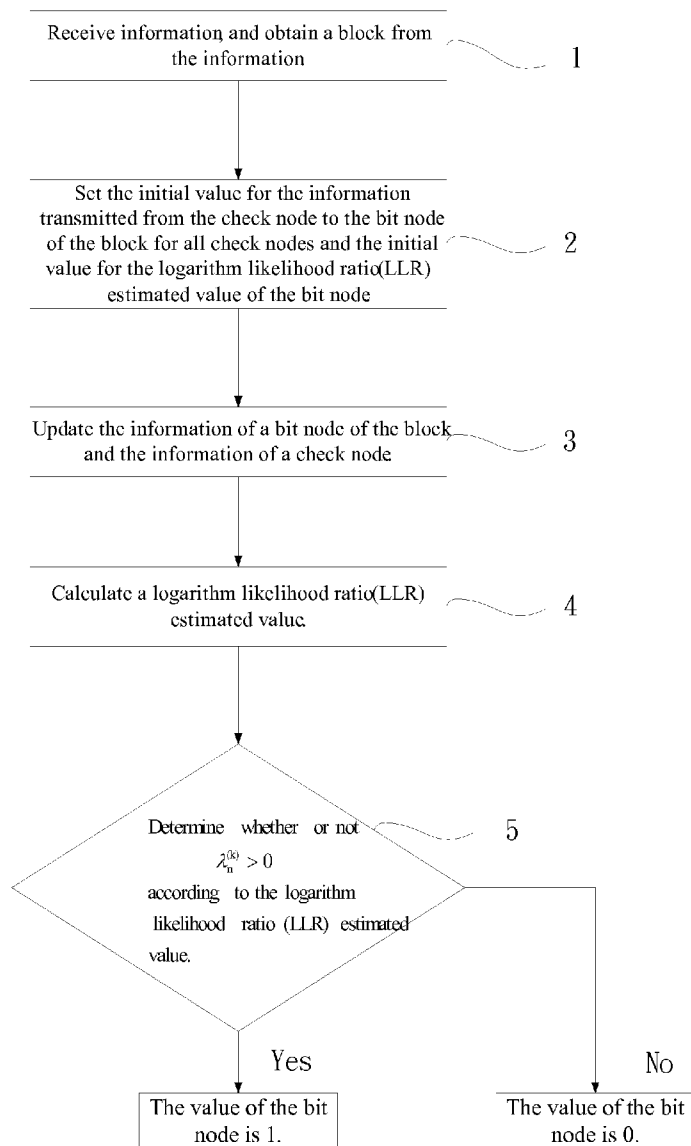
(21) Appl. No.: **11/838,970**

A fast decoding method for low density parity check (LDPC) code obtains a block from the information received by a digital communication system, and computes the information of a bit node and a check node by a simplified method, and determine the actual value of the bit node from the logarithm likelihood ratio (LLR) estimated value of each bit node. The simplified method of the invention provides a more effective simplified circuit and a faster computation speed than the traditional functions while maintaining the same error code rate as that before the simplification.

(22) Filed: **Aug. 15, 2007**

(30) **Foreign Application Priority Data**

Jun. 27, 2007 (TW) ..... 096123314



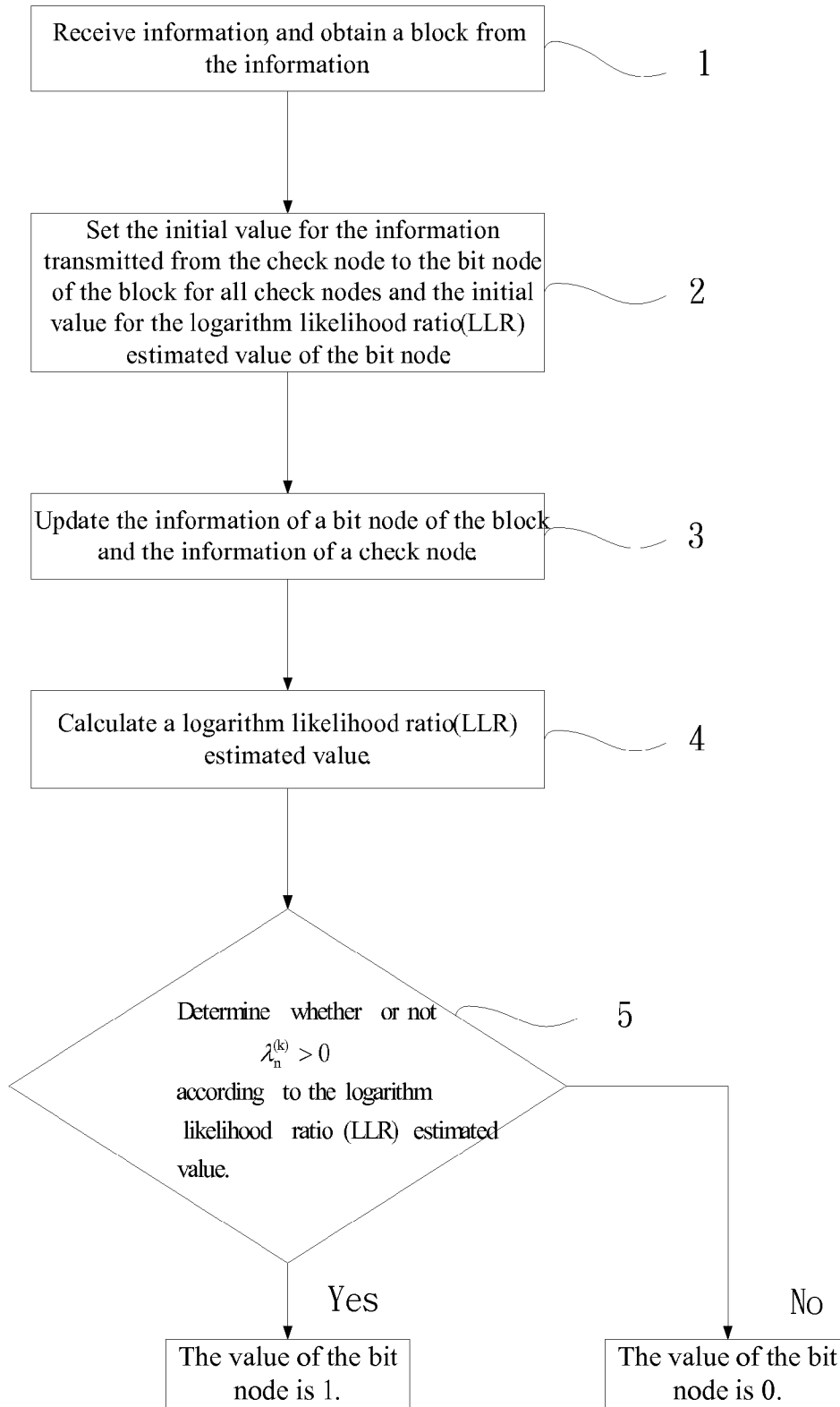


FIG. 1

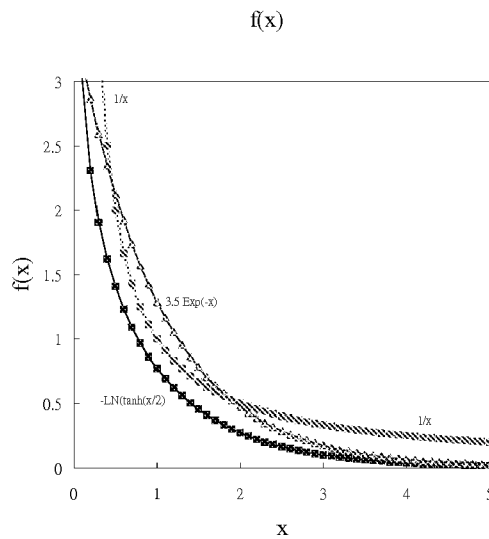


FIG. 2

Decoding Method	No. of Errors Bits (when 51299 bits are transmitted)	BER	Computation Time (sec)
$3.5e^{-x}$	3356	0.0655	9
$1/x$	5091	0.0994	11
Min-Sum Algorithm	4964	0.0969	9
$-\ln \tanh(x/2)$	3128	0.0609	14

FIG. 3

**FAST DECODING METHOD FOR LOW DENSITY PARITY CHECK (LDPC) CODE**

FIELD OF THE INVENTION

**[0001]** The present invention relates to a decoding method, and more particularly to a simplified decoding method for low density parity check (LDPC) code that can improve computation speed by using the message passing (MP) algorithm.

BACKGROUND OF THE INVENTION

**[0002]** Forward error correction (FEC) decoding methods such as the Hamming, BCH, Reed Solomon (RS), and turbo codes are used extensively in the industry. However, the level of complexity of these traditional decoding methods increases exponentially with code length. The low density parity check (LDPC) code plays an important role in a digital communication system mainly because its coding/decoding techniques can achieve excellent performance that approaches the Shannon limit.

**[0003]** The coding/decoding theory proposed by Claude E. Shannon in 1948 shows that each Gaussian noise channel has a capacity, and a codeword error caused by such a channel noise can be reduced to arbitrary desired level if the data in the digital communication system is encoded appropriately and the code rate of the codeword is smaller than the channel capacity.

**[0004]** In 1962, Robert G. Gallager proposed a simple structured parity check code, which is known as the low density parity check (LDPC) code due to its sparse parity check matrix. This method, however, did not catch much attention at that time. Until 1981, Michael Tanner graphically specified the low density parity check (LDPC) code by a bipartite graph according to a graph theory with a sum-product message passing algorithm. In 1997, David J. C. MacKay and Radford M. Neal rediscovered the low density parity check (LDPC) and obtained an outstanding result that approaches the Shannon limit.

**[0005]** In addition, the structure of LDPC code is simpler than that of turbo code. Unlike a traditional sequential method that has to wait till all stages of the computation are completed, the decoding of a LDPC code adopts a recurrent parallel processing method, which can be stopped at anytime to obtain a solution. Moreover, some part of the algorithm of this decoding method simply uses addition operations making the calculation much faster than the traditional ones.

**[0006]** A low density parity check (LDPC) code can be represented in terms of a parity check matrix H, which is used to be converted a generating matrix G. To generate a codeword, we multiply a raw information vector X with the generating matrix G to yield GX=C. The codeword C should satisfy the constraint CH<sup>T</sup>=0, otherwise errors had incurred in the transmission of the codeword C. And such errors should be corrected at the receiver decoding stage.

**[0007]** The parity check matrix H is a m×n matrix that treats each row of the matrix H as a check node, and each column as a bit node. Therefore, there are m check nodes and n bit nodes.

**[0008]** The traditional low density parity check (LDPC) algorithm calculates the message u<sub>m</sub><sup>(k)</sup>→n sent from the check node m to the bit node n in the k-th recursion as follows:

$$u_{m \rightarrow n}^{(k)} = -sf\left(\sum_{i \in N(m) \setminus n} f(|d_{m-i}^{(k-1)}|)\right) \tag{1}$$

$$u_{m \rightarrow n}^{(k)} \approx -s \min_{i \in N(m) \setminus n} |d_{m-i}^{(k-1)}| \tag{2}$$

$$u_{m \rightarrow n}^{(k)} \approx -\alpha s \min_{i \in N(m) \setminus n} |d_{m-i}^{(k-1)}|, 0.8 \leq \alpha \leq 1, \tag{3}$$

$$u_{m \rightarrow n}^{(k)} \approx s \max\left(\min_{i \in N(m) \setminus n} |d_{m-i}^{(k-1)}| - \beta, 0\right), \beta > 0,$$

where

$$s = \Pi \operatorname{sign}(-d_{m-i}^{(k-1)}) \text{ and } f(x) = -\ln \tanh(x/2).$$

**[0009]** Equation (2) is called a min-sum approximation, which has the drawbacks of loss of information and compensation issue. In Equation (3) two correction α and β terms have been added to Equation (2). After a recursion has been taken place, the logarithm likelihood ratio (LLR) representing the message produced at the n<sup>th</sup> bit node is λ<sub>n</sub><sup>(k)</sup> given by

$$\lambda_n^{(k)} = \frac{2}{\sigma^2} r_n + \sum_{m \in M(n)} u_{m \rightarrow n}^{(k)},$$

where σ<sup>2</sup> is the variance of a Gaussian channel noise, the value σ of its square root is the standard deviation of the channel Gaussian noise, r<sub>n</sub> is the value of the n<sup>th</sup> sample in the received sample vector r=[r<sub>1</sub>, . . . , r<sub>m</sub>, . . . , r<sub>N</sub>], and N is the length of the codeword. Here the sign of the logarithm likelihood ratio (LLR) of the estimated value is used for determining whether or not the value of each bit node is 1 or 0.

SUMMARY OF THE INVENTION

**[0010]** Because of the traditional low density parity check code (LDPC) decoding method implements a complicated function f(x)=-ln tan h(x/2), the inventor of the present invention based on years of experience developed and designed a fast decoding method for the low density parity check (LDPC) code.

**[0011]** It is a primary objective of the present invention to provide a fast low density parity check (LDPC) decoding method, such that when a check node is calculated, the following simplified mathematical function can be used for the calculation:

$$f(x)=1/x, \text{ or } f(x)=3.5 e^{-x}.$$

**[0012]** The simplified mathematical function reduces the level of complexity and computation time of the traditional low density parity check (LDPC) decoding method for calculating f(x)=-ln tan h(x/2) while the error rate (BER) has incurred no significant increase. For the design and manufacturing technology of integrated circuits advance, LDPC code has become a practical one for both theoretical and commercial applications. Furthermore, the LDPC code can be used together with the fast low density parity check (LDPC) decoding method in accordance with the invention to achieve the high-speed, high-capacity and high-reliability effects.

**[0013]** To make it easier for our examiner to understand the value of the invention, its structure, innovative features, and

performance, we use a preferred embodiment together with the attached drawings for the detailed description of the invention.

BRIEF DESCRIPTION OF THE DRAWING

[0014] FIG. 1 is a flow chart of a decoding method of the present invention;

[0015] FIG. 2 shows a comparison between a simplified function of the invention and a traditional function; and

[0016] FIG. 3 shows simulated results of a simplified function of the invention and a traditional function.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0017] The present invention relates to a fast decoding method for low density parity check code (LDPC), and the decoding method can be applied to cable, wireless and optical fiber communications having an error-tolerance decoding system, an IC decoding algorithm or a data storage and recovery that require a highly reliable decoding algorithm. Referring to FIG. 1 for a flow chart of a decoding procedure of the present invention, the procedure comprises of the following steps:

[0018] Step 1: Receive information, and obtain a block from a codeword block of the information;

[0019] Execute a message passing (MP) algorithm, which comprises the following steps:

[0020] Step 2: Set the initial value  $u_m^{(0)}=0$  for the information transmitted from the check node to the bit node of the block for all check nodes  $m \in \{1, \dots, M\}$  and all bit nodes  $n \in N(m)$  connected to the check nodes, and set the initial value

$$\lambda_n^{(0)} = \frac{2}{\sigma^2} r_n$$

for the logarithm likelihood ratio (LLR) estimated value of the bit node of the block for all bit nodes  $n \in \{1, \dots, N\}$ , where  $\sigma^2$  is the Gaussian noise variable in the channel, the value  $\sigma$  of its square root is the noise standard deviation,  $r_n$  is the value of the  $n^{th}$  received sample in  $r=[r_1, \dots, r_n, \dots, r_N]$  and  $N$  is the codeword length;

[0021] Step 3: Set the recursion counter  $k=1, 2, \dots, k_{max}$ , and the recursion counter can be set at an appropriate number of times as needed. And the recursion method is used to update the information of a bit node of the block,

$$d_{m \leftarrow n}^{(k-1)} = \lambda_n^{(k-1)} - u_{m \rightarrow n}^{(k-1)},$$

where  $d_{m \leftarrow n}^{(k-1)}$  is the message sent from the bit node  $n$  to the check node  $m$  in the  $(k-1)^{th}$  recursion;

[0022] After the message of the bit node is updated, a simplified mathematical function is used to update the message of the check node, and it is updated by calculating a

$$\sum_{i \in N(m) \setminus n} f(|d_{m \leftarrow i}^{(k-1)}|)$$

function, and the simplified mathematical function  $f(x)$  in the

$$\sum_{i \in N(m) \setminus n} f(|d_{m \leftarrow i}^{(k-1)}|)$$

function is  $f(x)=1/x$  or the  $f(x)=3.5 e^{-x}$  function, and the  $d_{m \leftarrow i}^{(k-1)}$  is the message sent from the bit node  $n$  to the check node  $m$  in the  $(k-1)^{th}$  recursion;

[0023] Substitute the value of the

$$\sum_{i \in N(m) \setminus n} f(|d_{m \leftarrow i}^{(k-1)}|)$$

function into the function

$$u_{m \rightarrow n}^{(k)} = f\left(\sum_{i \in N(m) \setminus n} f(|d_{m \leftarrow i}^{(k-1)}|)\right)$$

for a calculation to complete an update of the check node. The function  $f$  used in the computation is set to be  $f(x)=1/x$  or  $f(x)=3.5 e^{-x}$ , such that when the value of  $u_{m \rightarrow n}^{(k)}$  is calculated, the simplified mathematical function acts as a key engine function which can reduce the computation time greatly;

[0024] Step 4: After the information of the check node is updated and  $k$  times of recursion are calculated, the logarithm likelihood ratio (LLR) message  $\lambda_n^{(k)}$  of the bit node is:

$$\lambda_n^{(k)} = \frac{2}{\sigma^2} r_n + \sum_{m \in M(n)} u_{m \rightarrow n}^{(k)}$$

[0025] Step 5: Determine the message of each bit node as the logarithm likelihood ratio (LLR) value by the following conditions: if the logarithm likelihood ratio (LLR) value  $\lambda_n^{(k)}$  is greater than 0, then the  $n^{th}$  bit node is 1, and if the logarithm likelihood ratio (LLR) estimated value  $\lambda_n^{(k)}$  is less than zero, then the  $n^{th}$  bit node is 0.

[0026] Referring to FIG. 2 for a comparison between the simplified function of the invention and the traditional function, if  $f(x)=3.5 e^{-x}$ , then it is closer to the traditional function  $f(x)=-\ln \tan h(x/2)$  than  $f(x)=1/x$ , but the computation speeds of both functions are faster than the computation speed of the traditional function. Referring to FIG. 3 for the simulated results of the decoding by the simplified function of the present invention and the traditional function, the simplified function reduces the computation time (by approximately 35%) while BER does not have any significant increase.

[0027] The fast low density parity check (LDPC) decoding method of the invention improves the prior art as described below:

[0028] 1. The simplified function  $f(x)=1/x$  or  $f(x)=3.5 e^{-x}$  is simpler and easier to implement than the traditional function, and thus the computation time is much reduced.

[0029] 2. Unlike the min-sum algorithm, the decoding method does not need to find the minimum value before removing other values, and thus there is no concern of losing information or making necessary compensations.

[0030] 3. The function  $f(x)=3.5 e^{-x}$  can avoid round-off errors better than  $f(x)=-\ln \tan h(x/2)$  caused by a limited bit of operations, thus a highly reliable sample value will not be discarded.

[0031] It is to be understood, however, that even though numerous characteristics and advantages of the present invention have been set forth in the foregoing description, together with details of the structure and function of the invention, the disclosure is illustrative only, and changes may be made in detail within the principles of the invention to the full extent indicated by the broad general meaning of the terms in which the appended claims are expressed.

What is claimed is:

1. A fast decoding method for low density parity check (LDPC) code, comprising the steps of:

- (a) receiving information, and obtaining a block from a codeword in a block of the information;
- (b) setting an initial value of a bit node information sent from a check node of the block, and an initial value of a logarithm likelihood ratio (LLR) estimated value of the bit node of the block;
- (c) updating the check node information by using a simplified mathematical function, after the bit node information of the block is updated;
- (d) calculating a logarithm likelihood ratio (LLR) estimated value of the bit node by the updated check node information, for determining the value of each bit node according to the logarithm likelihood ratio (LLR) estimated value.

2. The fast decoding method for low density parity check (LDPC) code as recited in claim 1, wherein the check node is updated by calculating a

$$\sum_{i \in N(m) \setminus n} f(|d_{m-i}^{(k-1)}|)$$

function first, and a simplified mathematical function of  $f(x)$  in the summation

$$\sum_{i \in N(m) \setminus n} f(|d_{m-i}^{(k-1)}|)$$

is a  $f(x)=1/x$  function, and at the  $(k-1)^{th}$  recursion the message  $d_{m-i}^{(k-1)}$  is sent from the bit node  $i$  to the check node  $m$ ; and then the value of the

$$\sum_{i \in N(m) \setminus n} f(|d_{m-i}^{(k-1)}|)$$

function is substituted into another function given by

$$u_{m \rightarrow n}^{(k)} = -s f\left(\sum_{i \in N(m) \setminus n} f(|d_{m-i}^{(k-1)}|)\right)$$

for a calculation that updates the check node message, and the simplified mathematical function

$$f\left(\sum_{i \in N(m) \setminus n} f(|d_{m-i}^{(k-1)}|)\right)$$

in the other

$$u_{m \rightarrow n}^{(k)} = -s f\left(\sum_{i \in N(m) \setminus n} f(|d_{m-i}^{(k-1)}|)\right)$$

function for a calculation that updates the check node message.

3. The fast decoding method for low density parity check (LDPC) as recited in claim 1, wherein the check node is updated by calculating a

$$\sum_{i \in N(m) \setminus n} f(|d_{m-i}^{(k-1)}|)$$

function first, and a simplified mathematical function of  $f(|d_{m-i}^{(k-1)}|)$  in the summation

$$\sum_{i \in N(m) \setminus n} f(|d_{m-i}^{(k-1)}|)$$

is a  $f(x)=3.5 e^{-x}$  function, and at the  $(k-1)$ th recursion the message  $d_{m-i}^{(k-1)}$  is sent from the bit node  $i$  to the check node  $m$ ;

and then the value of the

$$\sum_{i \in N(m) \setminus n} f(|d_{m-i}^{(k-1)}|)$$

function is substituted into another function given by

$$u_{m \rightarrow n}^{(k)} = -s f\left(\sum_{i \in N(m) \setminus n} f(|d_{m-i}^{(k-1)}|)\right)$$

for a calculation that updates the check node message.

4. The fast decoding method for low density parity check (LDPC) as recited in claim 1, wherein the value of each bit node is determined by a condition that if a logarithm likelihood ratio (LLR) estimated value is greater than zero, then the  $n^{th}$  bit node is 1, and if the logarithm likelihood ratio (LLR) estimated value is less than zero, then the  $n^{th}$  bit node is 0.

\* \* \* \* \*