Design of Improved LDPC Encoder for CMMB Based on SIMD Architecture
Hang Yin, Weitao Du and Nanhao Zhu

Abstract—This paper designs and implements a novel parallel LDPC encoder. It is based on LU decomposition, according to the inherent characteristics of LDPC Parity-Check Matrix in CMMB. It is applied to design CMMB baseband exciter, which can support 2 different code rates (1/2 and 3/4). The SIMD parallel architecture is proposed to solve the encoding delay caused by iteration of LU algorithm, full pipeline and multistage Ping-Pong buffer structure are also used to improve throughput in high-speed encoding. It meets the requirements both in real-time performance and resource utilization. Furthermore, this method is generic and can be adapted easily for other LDPC codes; thus, it has a significant practical value.

I. INTRODUCTION

The error correction coding is a key technology in data communication and disk storage system. Low Density Parity Check Code (LDPC) is a kind of linear block code which is first presented by Dr. Gallager in 1960 [1]. LDPC was considered as impractical because of its large storage space, due to the limitation of computation capability at that time. In 1996, MacKay et al. (and Neal) presented a LDPC under the n-ary symbol domain based on BP (Belief Propagation) with the combination of iterative decoding, which demonstrated that it had an approaching performance compared to Shannon [2]. In 2001, Chung et al. designed an irregular LDPC based on irregular two-dimensional plot, with rate of 0.5 and length of 10^7 bits, whose performance is only 0.0045dB compared to Shannon, and now it is the nearest code to Shannon in the existing code [3]. On the other hand, some recently presented national standards of digital media, such as Digital Television Terrestrial Broadcasting (DTTB), China Mobile Multimedia Broadcasting (CMMB) and Digital Video Broadcasting-Satellite (DVB-S2) use LDPC as the solution in their channel encoding parts [4].

Usually, the parity-check matrix of LDPC is highly scattered, but not for generator matrix. The use of linear block code as the universal encoding method will make encoding complexity proportional to the square of the code length $O(N^2)$, which is hard to be acceptable when the code is longer. Although LDPC is capable of better performance and lower decoding complexity when comparing to Turbo, it must face some big challenges such as higher encoding complexity, huge hardware overhead and encoding delay, even from the point of view of today’s VLSI technology. With the development of existing communication system, high data throughput under reliable transmission is the inevitable tendency and the difficulty in designing LDPC encoder is how to select the appropriate algorithm and hardware structure to achieve the balance between resources and efficiency.

Traditional encoding algorithm is to use H matrix to get G matrix, and then encodes. Meanwhile, Gauss elimination exists in this process, but it destroys the sparsity of matrix such that the getting of G matrix is unscattered. Traditional algorithm needs high complexity and hardware overhead, so it is unavailable in practical use. Many authors have addressed this issue for both encoding and decoding of LDPC codes [5]-[8]. The first universal encoding algorithm with linear complexity based on LU decomposition algorithm was presented by Neal in 1999 [9]. The matrix inversion can be changed to the process of forward iteration and backward iteration; as long as the algorithm satisfies the condition that parity check bit block matrix which has the full rank. Its encoding complexity depends on the sparse level of decomposed L and U matrixes.

Although LU decomposition can provide linear complexity encoding algorithm while reduce the hardware overhead to the acceptable level, the inherent iteration characteristic of LU algorithm make it unavailable to use universal full pipeline structure to improve its encoding throughput and this cannot satisfy the increasing data transmission requirement of modern communication system. After analyzing the hardware implementation of LU algorithm, this paper draws the Single instruction, multiple data (SIMD) thinking in computer architecture to present a new and parallel LU algorithm encoder, which significantly improves its data throughput with only a small increase in hardware overhead.

II. LDPC ENCODING ALGORITHM BASED ON LU DECOMPOSITION

A. Encoding Based On LU Decomposition Algorithm

For LDPC used in (n, k) system, its parity-check matrix is divided into two sub-matrixes $H = [A \ B]$. B is the signal bit corresponding to $[(n-k) \times k]$ matrix, while A is the parity-check bit corresponding to $[(n-k) \times (n-k)]$ matrix. (1) can be gotten from $H \times C^T = 0$, $C = [s \ p]$ here is the code length.

$$[B \ A] \times \begin{bmatrix} s^T \\ p^T \end{bmatrix} = 0 \tag{1}$$

$s^T$ is the column vector with the length of k and presents k...
bits of information bit, $p^T$ is the column vector with the length of $(n-k)$ and presents $(n-k)$ bits length of parity. The essence of encoding is to calculate parity $p$.

$$p^T = A^{-1} \times B \times s^T$$

From the above formula, the calculation of parity bit is divided into the two parts, B multiplies by matrix vector of $s$ and the inverse matrix of matrix by the results vector. The multiplication of sparse matrix and vector quantity is about the linear complexity, but the inverse process of matrix A will seriously destruct its sparsity, so the complexity of will square increase with the size of matrix A, and this is usually unacceptable for longer LDPC. On the other hand, a full ranked matrix can be written as $A=L \times U$ via matrix elementary transformation. L is the matrix of next delta, and U is the matrix of last delta, so LU decomposition can change the calculation of $A^{-1}$ to forward and backward iteration. The following formula can be got when substituting $L \times U$ for $A$:

$$L \times (U \times p^T) = B \times s^T$$

We can get equation (4), so the complexity of LU decomposition based encoding depends on the sparse level of matrices L and U.

$$p^T = U^{-1} \times \left[ L^{-1} \times (B \times s^T) \right]$$

LU decomposition processes by judging the elements on the diagonal, which is by judging whether main element is 1 and because of the different policy of pivoting the final sparse level of matrices L and U are also different. Optimized partial pivoting strategy was used for LU decomposition, which based on minimum weight of row. Meanwhile as the exchange of a row is introduced in the process of LU decomposition, this will introduce row permutation matrix P in the calculation of parity-check, and then can get the following formulas:

$$P \times A = L \times U$$

$$p^T = U^{-1} \times \left[ L^{-1} \times ((P \times B) \times s^T) \right]$$

The LDPC encoding algorithm based on LU decomposition is mainly divided into three stages: pre-processing stage, encoding stage, code generation stage. From the above analysis, the encoding process can also be divided into another three steps: matrix multiplication of $B^T \times S^T$, forward iteration and backward iteration. In the process of implementation of encoding, to get the desired iteration matrix, LU decomposition is preprocessed by Matlab, and the only thing that the encoder will do is to complete the relative matrix multiplication and iteration calculation. The following figure 1 shows the flow of LDPC encoding, which is based on LU decomposition.

### B. LU Decomposition Of CMMB

The standard of China Mobile Multimedia Broadcasting adopts highly structuralized binary LDPC code with the length of 9216 and provides two code rates; one code rate is 0.5 with the row weight of 6 and column weight of 3. Another code rate is 0.75 with the row weight of 12 and column weight of 3, both of them are system code [10]. CMMB standard gives the sparse parity matrix, but no generation matrix, which is to make the decoding parity matrix as a quasi-cyclic form. The parity check matrix can be made up of moving 36 bits to the right every 18 lines or 9 lines for code rate of 0.5 and 0.75 respectively. H matrix pattern of 2 code rates are showed below in Figure 2:

![Fig. 1. The Flow of LDPC Encoding based on LU Decomposition](image)

(a) Code rate 0.5

(b) Code rate 0.75

**Fig. 2. H Matrix Pattern of 2 Code Rate LDPC**

We can get full rank parity-check matrix in system code form after using code bit mapping vector in the standard to rearrange the parity-check matrix. Since the standard does not use the quasi-cyclic LDPC code, only universal encoding can be used. CMMB standard supports different system payload data rate from 2.046Mbps to 16.243Mbps (8MHz bandwidth) as well as rate from 0.409Mbps to 3.248Mbps (2MHz bandwidth), which needs the data throughput of the encoder to satisfy a higher desire and the encoding technology is a difficult point. Table 1. showed LU decomposition results of 1/2 and 3/4 code rate.

| TABLE I. STORAGE RESOURCES CONSUMPTION OF LDPC ENCODER |
|---|---|---|---|
| Code Rate | B | L | U |
| 0.5 | 14158 | 31982 | 19824 |
| 0.75 | 20736 | 9010 | 4540 |
III. LDPC Encoder Based on SIMD Architecture

A. The Presentation Based On SIMD Architecture Optimization

According to LU decomposition based LDPC encoding, we consider information bit is \([s_0, s_1, \ldots, s_m, 1]\) and then we get the vector \([z_0, z_1, \ldots, z_k, 1]\) after the matrix-vector multiplier. This matrix-vector multiplier first calculates \(z_0\) to \(z_k\) sequentially and then we get vector \([y_0, y_1, \ldots, y_k, 1]\) after the forward iteration operation in the second level of multiplier. The forward iteration begins from the lower bit, that is means that it first calculates \(y_0, y_1, \ldots\), and finally \(y_k, 1\). There is no need to begin calculating forward iteration when all the calculation of matrix vector is over. We can start the iteration operation after some buffer time which is according to the analysis of matrix \(L\). The vector after backward iteration is \([c_0, c_1, \ldots, c_k, 1]\), which is the multiplication of upper triangular matrix and vector \([y_0, y_1, \ldots, y_k, 1]\). The calculation of the is from the last row of the matrix, so we must first use \(y_k, 1\) to calculate \(c_k, 1\) until we calculate \(c_0\). In this module, \(y_k, 1\) is the first element used in the calculation and it is also the last bit got from its last module, so the backward iteration should begin only after the forward iteration. If there exists column transition in the LU decomposition, we must finish it according to the mapping vector and then get the final parity-check bit. What’s more, we must finish matrix vector multiplication, forward iteration, backward iteration and code output in order before getting a complete LDPC encoding, this process must be done step by step and the calculation of every code is dependent on its last code.

According to the above analysis, we discover that LDPC encoding based LU decomposition can be realized by two levels iterations. The essence of forward and backward iteration operation is exclusive or operation according to the number of 1 in the matrix \(L\) and \(U\), and also determines the number of iteration clock cycle. It makes the calculation of the next sub-vector in the target vector depend on all the previous sub-vectors, due to the recursive and serial computation structure of iteration. It considerably limits the improvement of data throughput, and it is difficult to reduce encoding delay and improve data throughput by enhancing the parallelism.

For current LU decomposition based LDPC encoder structure, only by optimizing internal operation cycle of matrix decomposed compression, grading pipeline and improving the system’s operation clock frequency cannot satisfy the increasing throughput requirement in today’s communication system. We are trying to adopt SIMD architecture to optimize the hardware structure of LU decomposed LDPC encoder because the great advantage in processing of large vector data.

B. Encoder Structure Based On SIMD Architecture

Single instruction, multiple data (SIMD) is a kind of technology with one central controller to command multiple processing units. It executed a same operation separately in each vector of a set of data to realize spatial parallelism. SIMD architecture abstract module can administrate multiple process units under the same control unit, and all the process units will receive the same instruction broadcasted by the control unit, but the operation objects are different data. The characteristic of this hardware structure is that the parallel working mechanism of multiple processing units is under controlled by one control unit.

After investigated several methods and the existent hardware implementation of LDPC encoder based on LU algorithm, the paper presents a new LDPC encoder based on SIMD parallel structure. It significantly improves the encoding throughput with only few increases in hardware overhead; meanwhile this will not bring high complex control logic, and it is easy to realize. The following figure shows the structure of LDPC encoder based on SIMD architecture.

Comparing to original hardware structure, LDPC encoder based on SIMD architecture can use the operation instructions which are produced every time to operate several different groups of data simultaneously, on the other hand, encoder processing throughput is also greatly improved via structuralized parallelism.

Figure 3 is the system structure of LDPC encoder based on SIMD architecture, and it can be adapted easily for other systematic LDPC codes. This structure consists of the following function units.

SCU (SIMD Control Unit): In charge of time scheduling of data interface, read address generation of matrix LU, encoding control of PU unit.

MMC (Matrix Vector Memory Cell): storage unit, B, L, U are sparse matrixes, using the policy of memorizing row/column position of element 1 and its flag position to reduce the overhead of storage resources.

CPU (Coding Processing Unit): It is to complete matrix vector multiplication, forward iteration, backward iteration and some relative encoding operation like code output according to instructions which are sent by SCU. The essence of vector multiplication in matrix CPU and iteration is XOR operation unit of full pipeline structure. On the other hand, operating intermediate variable will continuously update according to index of sparse matrix and at this moment read / write operation will be on the same RAM address which leads to read / write collision when RAM is trying to update its data. When the flag appears on the read / write collision of RAM,
the design of bypass advanced pipeline can use the values in bypass register to update data, keep the continuity of unit pipeline and this design also improve the efficiency of CPU.

SIMD is a generic architecture, and the structure of its processing unit is not fixed. In order to improve the operation efficiency of processing unit and further improve the entire performance of the encoder, we can use existent improved encoder structure to optimize coding processing unit.

IV. HARDWARE IMPLEMENTATION AND TEST

This paper designs an improved LDPC Encoder in CMMB Based on SIMD Architecture. All the matrixes and vectors used in the encoding experience will be stored in on-chip RAM. It was implemented on Altera Cyclone III EP3C120 device; whole design was synthesized and routed under Quartus II 9.2 environment. To compare the performance of above SIMD structure optimized LDPC encoder, here we present the design of multi-pipelines structure LDPC encoder of Ref. [11] as the contrast. It was implemented on Altera Stratix II EP2S90 device. Table 2 shows a summary of synthesis results.

Both of the structures take a little logic resources, but more storage resources are consumed. The system payload throughput rate is only 16.10 Mbit/s, due to the encoding delay caused by the long code length based on multi-pipelines structure. It cannot satisfy system requirement when need to work at lower frequency. Based on the proposed SIMD structure, the LDPC encoder consist of five coding processing unit.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Reference</th>
<th>SIMD structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic Element</td>
<td>581</td>
<td>2,228</td>
</tr>
<tr>
<td>Storage Element(bit)</td>
<td>2,015,232</td>
<td>1,291,654</td>
</tr>
<tr>
<td>Throughtput 100MHz</td>
<td>0.5</td>
<td>31.86Mbit/s</td>
</tr>
<tr>
<td>Throughtput 100MHz</td>
<td>0.75</td>
<td>52.37 Mbit/s</td>
</tr>
</tbody>
</table>

The Clock Frequency is 100 MHz.

As the Table. 2 shows, the data throughput of LDPC encoder has a significant improvement, and also with a little increase in hardware and memory overhead. The system payload throughput rate can be up to 31.86 Mbit/s when it works at 100MHz system clock frequency. According to the above analysis, in SIMD structure, the number of CP unit can still be added to further improve encoder throughput; therefore the encoder base on SIMD structure is possible to make the encoder satisfy a higher system data throughput.

V. CONCLUSION

Based on SIMD processing structure, this paper adopts improved LU algorithm to design and implement the high performance low-complexity LDPC encoder, which satisfies China Mobile Multimedia Broadcasting indicators and another two standard code rate. This encoding structure solves low encoding rate brought by iteration of LU algorithm and it also greatly improve the encoding data throughput. This encoder has been validated on Altera EP3C120 FPGA, and the experiments demonstrate that data throughput provided by the encoder can fully satisfy data rate of CMMB system, also this encoder can leave extra processing ability for the follow-up units. Furthermore, this method is generic and can be adapted easily for other LDPC codes; thus, it has a significant practical value for engineering application.

ACKNOWLEDGMENT

This work was supported by the National Major Science and Technology Projects through funded by the China’s Ministry of Science and Technology (No. 2010ZX03005-001). The authors would like to thank the reviewers and the Associate Editor for their assistance with improving this paper by their valuable feedback on our manuscript.

REFERENCES

[12]