

A power-saving 1Gbps Irregular LDPC Decoder based on high-efficiency message-passing

Wenming Tang, Wen Ji, Xianghui Wei, Takeshi Ikenaga, Satoshi Goto

Graduate School of Information, Production and System, Waseda University
2-7 Hibikino, Wakamatsuku, Kitakyushu-shi, 808-0135 Japan
E-mail: twm@fuji.waseda.jp

Abstract: In this paper we proposed a partially-parallel decoder for irregular LDPC codes from IEEE802.11n standards. Our proposed decoder adopts high-efficiency message-passing algorithm and uses the min-sum algorithm handle the message-passing to reduce the hardware implementation complexity and area, and keep high throughput. Considering reducing the power consumption, we used half-registers and half-memory to store the temporary intrinsic messages. The wasted motion of shift-register was suppressed., This strategy would save us higher as 30% power under good channel condition. The synthesis result in TSMC 0.18um COMS technology demonstrated that for (1296,324) irregular LDPC code achieved high throughput (1.05Gbps) at the frequency of 200MHz, with 6% area reduction.

keywords— Low-Density Parity-Check, Message-Passing Algorithm, register, memory, TSMC 0.18um

1. Introduction

Low-Density Parity-check (LDPC) code was proposed by Gallager in the early 1960's [1], and it was recovered by MacKay and Neal in 1996[2]. With very good error correction capability, Low-density parity-check(LDPC) code had attracted wide attention over within 0.0035dB of the Shannon limit by Chung.[3] Compared with the traditional Turbo Codes, for the length data based LDPC code performs much better on error correction. Also the LDPC decoding algorithm can support high speed parallel processing. So far LDPC code has attracted much attention all around the world and been recognized as the best error correction code in the near future, demonstrated only 0.04dB close to Shannon limit at BER. Recently, the inherent parallelism of the well-known message-passing algorithm that can realize row and column operations independently for LDPC codes which makes it very suitable for LDPC decoder hardware implementation.

The parity check matrix for irregular LDPC codes is illustrated in Figure.1. The matrix is composed of 6×24 (144) sub-blocks, each of which is 54bit×54bit square matrix. Each element inside the matrix denotes the number through which the corresponding sub-block could be obtained by right-shifting the identity matrix. For instance, the Row two Column one (2,2) sub-block can be gotten through right-shifting the identity matrix by 21. The “-” denotes Zero matrix.

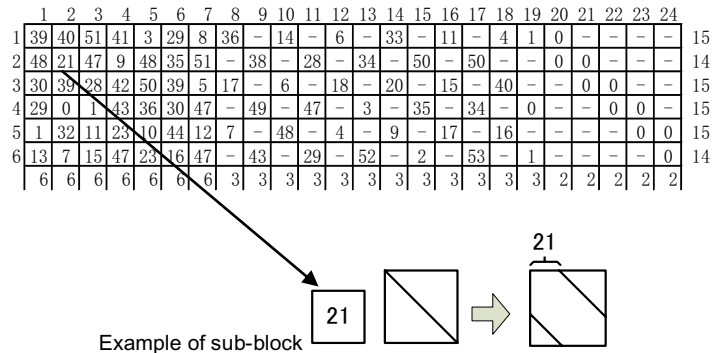


Figure 1: LDPC code parity check matrixes

There are two kind of parallel LDPC decoders for irregular LDPC decoder, one is Full-Parallel decoder and the other one is Partial-Parallel decoder.[9] The latter occupies less hardware area and shows better correcting performance than the former [5],[6],[7].

Although the partially-parallel irregular LDPC decoder proposed in [7] can improve the throughput performance, it suffers a main problem that the power consumption. Considering the fact that throughput is an essential metric in real-time communication systems applying LDPC code, we propose saving power consumption method and decoder architecture for irregular LDPC decoder, in this paper, to achieve a nearly error correcting performance.

We proposed irregular LDPC decoder is implemented under TSMC 0.18μm CMOS technology. The parity check matrix shown in Figure.1 is employed, which is one of the parity check matrixes specified by IEEE 802.11n Standard. With the equal error correcting performance, proposed design provides a 6% hardware area reduction and a 30% power consumption saving, compared with the design utilizing the LDPC decoder was proposed in [7].

The rest of the paper is organized as follows. Section II and Section III introduces the MIN-SUM Algorithm and High-efficiency message-passing algorithm separately. Section IV discusses our LDPC decoder method for reducing chip area and saving power consumption. Section V shows the detailed architecture of our decoder. In Section VI we demonstrate our implementation results and compare results. Finally Section VII is concludes.

2. Min-Sum algorithm

LDPC codes can be decoded iteratively using message-passing algorithm. But the hardware design was difficult. So we used MIN-SUM algorithm in Ref.[4] for easily to designed the hardware architecture

Initialization : Compute the log likelihood ratio(LLR) λ_n for bit node($n = 1, 2, \dots, N$), and set $\beta_{mn} = \lambda_n$ for each (m, n) satisfying $H_{mn}=1$.

Phase 1 : For all the check nodes C_m in the order to corresponding to $m = 1, 2, \dots, M$, compute the message α_{mn} with the following equation, where each set (m, n) satisfies $H_{mn}=1$.

$$\alpha_{mn} = \left(\prod_{n' \in A(m) \setminus n} \text{sign}(\beta_{mn'}) \right) \times \min_{n' \in A(m) \setminus n} |\beta_{mn'}| \quad (1)$$

Phase 2 : For all the bit nodes b_n in the order to corresponding to $n = 1, 2, \dots, N$, compute the message β_{mn} with the following equation, where each set (m, n) satisfies $H_{mn}=1$.

$$\beta_{mn} = \lambda_n + \sum_{m' \in B(n) \setminus m} \alpha_{m'n} \quad (2)$$

Tentative decision : Compute all the tentative LDPC code word \hat{y}_n for $n=1, 2, \dots, N$ with Eq.(3)

$$\hat{y}_n = \begin{cases} 0, \text{sign}(\lambda_n + \sum_{m' \in B(n)} \alpha_{m'n}) = 1 \\ 1, \text{sign}(\lambda_n + \sum_{m' \in B(n)} \alpha_{m'n}) = -1 \end{cases} \quad (3)$$

Parity Check : If the tentative codeword \hat{y}_n satisfies the parity check equation Eq.(4), or if the maximum number of iterations is reached then stop the algorithm, otherwise go to Phase 1, and continue iterations.

$$H \cdot (\hat{y}_1, \hat{y}_1, \dots, \hat{y}_N)^T = 0 \quad (4)$$

3. High-efficiency message-passing algorithm

We used the high efficient message-passing algorithm to improve data throughput. The high-efficiency message-passing algorithm is characterized as follows[5]:

- (1) The column operations for the bit nodes are performed in conjunction with the row operations for the check nodes.
- (2) The column operations are performed for every bit node connected to each of check nodes which are updated by the row operations in parallel.

High efficiency message-passing algorithm flow is shown in Figure2.

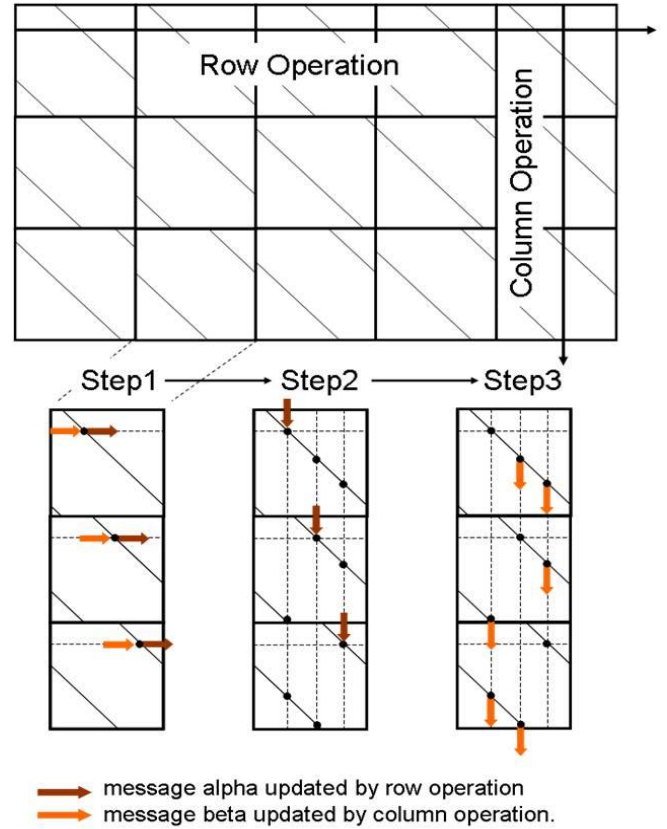


Figure 2. High-efficiency message-passing algorithm

Step 1: The row operations of the three block rows are performed in parallel simultaneously and each of them update the message of α in the respective row.

Step 2: In the block columns, every column operation is performed by using the updated message α value from the row with the same sequence number: the first column operation uses α value from the first row; the second column uses that from the second row and so on.

Step 3: Then every column operation updates all the message β of each respective column.

4. A compressing method for data store

4.1 Row compressing method

First of all, Comparison module of four-input is set for compute the first minimum and the second minimum. The modulation method is used to compare the β value of LDPC input. The module diagram shown in Figure3.

The method only saves the first and the second minimum value. Also, it saves the place of the first value and the sign value of all data.

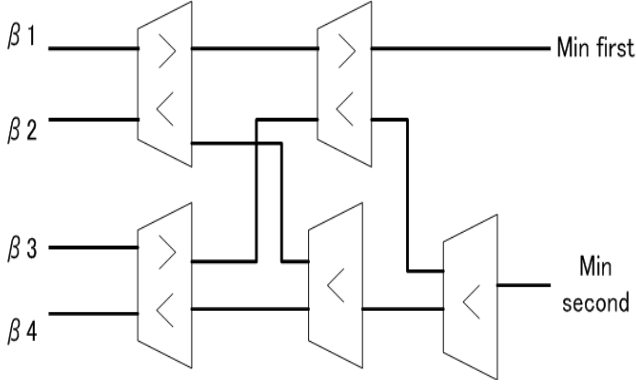


Figure 3.comparing β data of minimum first and second module

4. 2 Column compressing method

In Min-sum algorithm, we use the two steps for updating the β value.

$$\beta_{mn} = \lambda_n + \sum_{m' \in B(n) \setminus m} \alpha_{m'n}$$

Equation of column processing i.e.

$$\beta_k = \lambda + \alpha_1 + \dots + \alpha_{k-1} + \alpha_{k+1} + \dots + \alpha_n$$

Step 1: compute and store β_{all}

$$\beta_{all} = \lambda + \alpha_1 + \dots + \alpha_k + \dots + \alpha_n \quad (1)$$

$$\beta_k = \beta_{all} - \alpha_k \quad (2)$$

Step 2: Use the value of $\Delta\alpha$ to update β_{all}

$$\beta_{all}^{+1} = \beta_{all}^{-1} + \Delta\alpha_k \quad (3)$$

and save the new β_{all} data.

5. Hardware architecture

In this section, we illustrate the detailed architecture of our proposed irregular LDPC decoder. Row operation module, Column operation module and Pipeline architecture will be present in details in this section.

5. 1 Row Operation Module

In the row operation module, our decoder uses Section4 present method to get the minimum value and the second minimum value in the Section5.1 method. in the module setting, we use the three CFU module handle data at the same time to keep high throughput like the Ref[8]. But we propose the local memory to store data, throughput is less Ref[8] a little, but the local memory kill the useless movement of shift register to saved power consumption.

5. 2 Column Operation Module

In the column operation module, for Processing of Huge data, we use the shift-register to store the tentative data. Our decoder uses Section4 present method to get the small

update of value. Form the Δ register get the data of CFU handled to update the β . The flow showed in Figure. 4

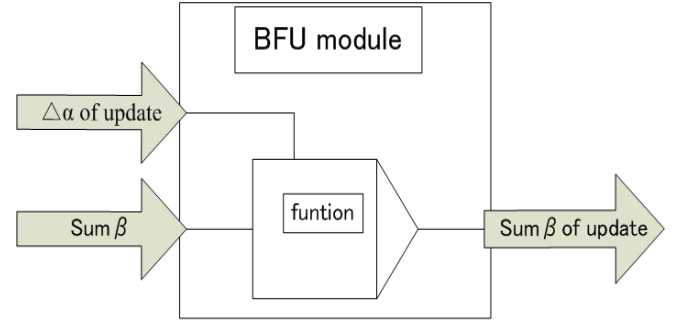


Figure 4.BFU module Diagram

5. 3 Block Diagram

Figure 5 is the block diagram of we proposed LDPC decoder. There are mainly six parts of modules, Including Controller modules, Row and Column operation modules, and Parity Check module. For we present method we proposed Shift-Register module and memory module Architecture for storing tentative data.

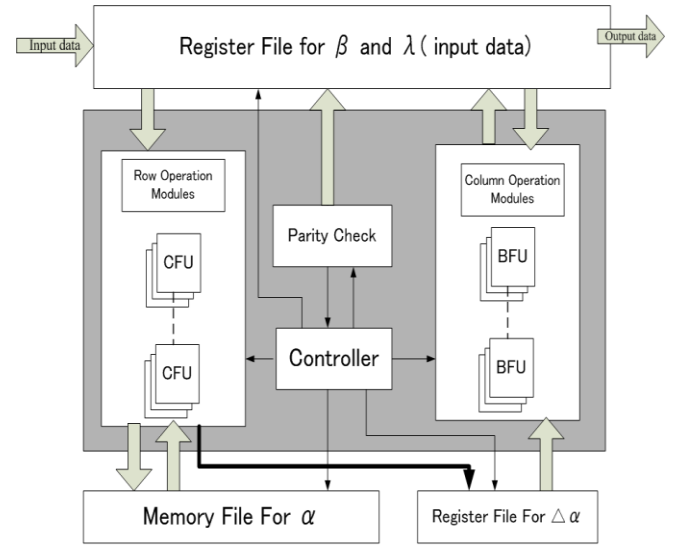


Figure 5 Block Diagram

6. Implementation result

In this section we discuss the practical implementation. By employing the parity check matrix specified for 1296bit (code rate 3/4) irregular LDPC code in IEEE 802.11n Standard, Figure 5 shows a block diagram of the proposed irregular LDPC decoder. The same bit error performance as shown in Figure 6, but the throughput becomes lower. At the same time, we proposed data save method can be reduce the decoder area and a useless movement of shift register is suppressed. Therefore we can save power consumption. We implement proposed decoder under the TSMC 0.18um CMOS process. The synthesis results are listed in Table1.

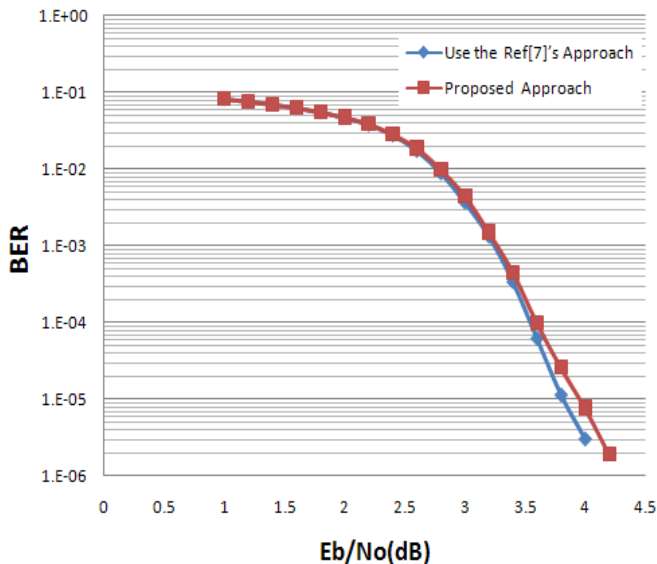


Figure 6. BER performance comparison

7. Conclusion

In this paper we proposed a high throughput partially-parallel irregular LDPC decoder. This decoder adopts min-sum algorithm and high-efficiency message-passing algorithm and compressing method of data store to reduce the implementation complexity and area. We use same memory to change shift-register for storing the “compressing alpha data”. Because a useless movement of shift register is suppressed. We can save power consumption efficiently as the channel condition becomes better. Our decoder can achieve low power and almost the same bit error rate performance on high throughput.

Table 1 Implementation Result

	Ref[6]	Ref[7]	Use the Ref[7]'s Approach	Proposed Approach
Design rule	TSMC 0.18um CMOS			
LDPC code	802.11n 648bit rate 1/2 irregular LDPC	802.11n 1296bit rate 1/2 irregular LDPC	802.11n 1296bit rate 3/4 irregular LDPC	
Memory Area(um ²)	7,082,885	No use	No use	2,093,358
Total Area(um ²)	13,090,549	24,424,648	17,936,840	17,166,638
Average Process@SNR =3.0dB	54Mbps (Iteration = 5) @200Mhz	837Mbps (Iteration = 6) @200Mhz	1.2Gbps (Iteration = 6) @200Mhz	1.05Gbps (Iteration = 6) @200Mhz
Power[mW] @200MHz,SNR =3.0,1.62v	765.85	755.36	641.87	454.93

8. Acknowledgement

This research was supported by CREST, JST.

References

- [1] R.G.Gallager, Low Density Parity Check Codes. Cambrider, MA: MIT press. 1963
- [2] D.J.C.MacKay, "Good error-correcting codes based on very sparse matrices," IEEE Trans. on Information Theory, Vol.47, pp.489-519, 2001.
- [3] S.Y Chung, G.D. Forney Jr., T.J. Richardson,R.L. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," IEEE Communications letters, Vol.5, No.2 , Feb.2001.
- [4] F.Zarkeshvari, A.H.Banihashemi, "On implementation of min-sum algorithm for decoding low-density parity-check (LDPC) codes," IEEE Global Telecommunications Conference, GLOBECOM'02, Volume2, pp.1349-1353,2002.
- [5] K.Shimizu, T.Ishikawa, N.Togawa, T.Ikenaga, S.Goto, "Partially-parallel LDPC decoder based on high efficiency message passing algorithm," Proc. IEEE International Conference on Computer Design 2005, pp.503-510, Oct. 2005
- [6] Kazunori Shimizu,et al ,"ASIC Implementation of LDPC Decoder Accelerating Message-Passing Schedule", Student Design Contest, IEEE International Solid State Circuits Conference(ISSCC), San Francisco, Feb. 5. 2006.
- [7]Yuta ABE, Xing LI, Kazunori SHIMIZU, Takeshi IKENAGA, Satoshi GOTO, "High-Throughput Design of High-Efficiency Message Passing partially parallel irregular-LDPC decoder based on 802.11n", ICESIT2008, Bangkok, Thailand, Feb. 2008
- [8]X.Li,Y. Abe, K.Shimizu, Z.Qiu, T.Ikenaga, S.Goto, "Cost-efficient partially-parallel irregular LDPC decoder with message passing schedule", International Symposium on Inetgrated Circuit(ISIC-2007),Sep.2007.
- [9]E. Liao, E. Yeo and B. Nikolic, "Low-density parity-check code constructions for hardware implementation," Proceedings 2004 IEEE International Conference on Communications, ICC'04,Paris, pp.2573-2577,June,2004.