TELKOMNIKA, Vol. 11, No. 5, May 2013, pp. 2699 ~ 2709
ISSN: 2302-4046

Low Cost Quasi Binary Weighting Addition Log-SPA LDPC Decoders

Po-Hui Yang*, Ming-Yu Lin
Department of Electronic Engineering, National Yunlin University of Science and Technology
123 University Road, Section 3, Douliou, Yunlin 64002, Taiwan, telp:+886-5-5342601-4340
*Corresponding author, e-mail: phyang@yuntech.edu.tw

Abstract

This paper proposes a new addition method, which can be applied to reduce the hardware cost of LDPC decoders using log-SPA, which traditionally has the best lowest bit error rate (BER) but the highest hardware requirement. The proposed quasi-binary weighting addition can be simply implemented by OR gates. With auxiliary pseudo-carry circuit, the BER performance can reach a fair level. In the log-SPA message-passing path, numeric transform reduction architecture is proposed for further hardware reduction. Synthesized and numerical results show that the new proposed architecture achieved an up to 32% and 18% total hardware reduction, compared with traditional log-SPA decoders, and the simplest sign-min architecture, respectively, with fair BER performance.

Keywords*: LDPC decoder, forward error correction code, log-SPA*

1. Introduction

New VLSI technology allows the low-density parity-check (LDPC) code [1, 2] to be realized in hardware. The excellence of the error correction performance of the LDPC code has introduced extensive discussions and studies [3-6]. The original LDPC decoder adopting a sum-product algorithm (SPA) required such large hardware that many new methods were proposed for hardware reduction [7-15]. Logarithm field transfer, called log-SPA [16], is a known method for replacing multipliers with adders and having the best bit error rate (BER) performance. Although adopting logarithmic operations to turn the original multiplication into addition, a large number of addition devices have also become hardware burdens. Many hardware-simplifying methodologies have been proposed to deal with mass addition operations. For example, parallel architecture [9, 12] and reorganized adder trees with re-maps skills [11] have been proposed to improve effectively the complexity of the hardware. However, a large number of adders and big look-up tables (LUT) are still the heavy hardware loadings. The proposed sign-min algorithm (SMA) architecture [8] has no additions in check node operations to save on hardware at the expense of BER performance. Mixed-signal circuits have been used by researchers to reduce the power and hardware requirements [17, 18]. However, this solution hinders the realization of technology migration, especially in all digital processes. For SMA to have good BER performance and to require additional compensation circuits [7, 13], hardware and BER are treated as tradeoffs. Having log-SPA with naturally good BER performance, a simplified circuit is utilized to replace the binary adders, which become the major focus of this study

2. LDPC Decoder Algorithm

Considering an LDPC code visualized by a Tanner graph consisting of *M* check nodes and *N* variable nodes, a set of the check nodes connected to variable nodes is denoted as $M(n)$ and the set of the variable nodes connected to check nodes is denoted as $N(m)$. $M(n)\backslash m$ represents the set $M(n)$ with the m^{th} check nodes excluded and $N(m)\backslash n$ represents the set $N(m)$ with n^{th} variable nodes excluded. During the decoding processes of the LDPC code, the SPA updates the soft information iteratively between check nodes and

Received January 19, 2013; Revised March 18, 2013; Accepted March 25, 2013

variable nodes. For notation simplification, $\lambda_{n\to m}(u_n)$ is defined the soft information sent from the variable node n to the check node m as

$$\lambda_{n\to m}(u_n) = \log\left\{\frac{q_{n\to m}(0)}{q_{n\to m}(1)}\right\}, \tag{1}$$

where the quantities $q_{n\to m}(x)$ are the probability information sent from the variable node n to the check node m along a connecting edge of a Tanner graph, indicating $P(c_n = x)$. Also, $\Lambda_{m\to n}(u_n)$ is defined as the soft information sent from the variable node n to the check node m as

$$\Lambda_{m\to n}(u_n) = \log\left\{\frac{r_{m\to n}(0)}{r_{m\to n}(1)}\right\}, \tag{2}$$

where the quantities $r_{m\to n}(x)$ are the probability information sent from the check node m to the variable node n along a connecting edge of a Tanner graph, indicating $P(c_n = x)$. Next, the iteration procedures of the SPA used in LDPC codes can be summarized as follows.

Step 1. Initialization. For the domain of log-likelihood, the log-likelihood ratio (LLR) is obtained for the case of the transmitted bit u_n = 0 and u_n = 1 given the received bit $y_n \in \{0,1\}$ which may be corrupted by the channel noise, like

$$L(u_n) = \log\left\{\frac{P(u_n = 0|y_n)}{P(u_n = 1|y_n)}\right\}. \tag{3}$$

The valves of $\lambda_{n\to m}(u_n)$ and $\Lambda_{m\to n}(u_n)$ are initialized to the values

$$\lambda_{n\to m}(u_n) = L(u_n) \tag{4}$$

and

$$\Lambda_{m\to n}(u_n) = 0, \tag{5}$$

respectively.

Step 2. Check node to variable node (update check nodes). According to the standard SPA, a check node m gathers all the incoming LLR messages and evaluates the LLR message sent to the variable node n, which can be expressed as

$$\Lambda_{m\to n}(u_n) = sign \prod_{n'\in N(m)\setminus n} \lambda_{n'\to m}\left(\frac{u_{n'}}{2}\right) \times \phi^{-1}\left(\sum_{n'\in N(m)\setminus n} \phi\left[\lambda_{n'\to m}\left(\frac{u_{n'}}{2}\right)\right]\right), \tag{6}$$

where

$$\phi(x) = -\log\left[\tanh\left(\frac{x}{2}\right)\right] = \log\left(\frac{e^x + 1}{e^x - 1}\right) \tag{7}$$

and

$$\phi^{-1}(x) = \phi(x). \tag{8}$$

Step 3. Variable node to check node (update variable node). According to the standard SPA, a variable node n passes the LLR message to all the connected check nodes, which can be expressed as

$$\lambda_{n\to m}(u_n) = L(u_n) + \sum_{m' \in M(n) \setminus m} \Lambda_{m'\to n}(u_n). \quad (9)$$

Step 4. Check stop criterion. The overall LLR message of a variable node n indicating the probability to decode the variable node n to 1 or 0 can be obtained by adding up all the incoming LLR messages to the variable node n as

$$\lambda_n(u_n) = L(u_n) + \sum_{m \in M(n)} \Lambda_{m\to n}(u_n). \quad (10)$$

After each iteration, a hard decision on the variable node n is made, i.e., the variable n is decoded as "one" when $\lambda_n(u_n) \geq 0$, and decoded as "zero" otherwise. If all the decision bits constitute a valid codeword, the algorithm stops and outputs the decoding result. Otherwise, the algorithm repeats Steps 2-4.

3. Log-SPA LDPC Decoder Hardware

The hardware structures of traditional log-SPA for the check node is shown in Figure 1 and the datapath of the variable node is shown in Figure 2, respectively. Focusing on the check node architecture is important because the hardware and computation complexities of the check node are high. Generally, the check node can be implemented by signing and adding two independent operational parts [9].

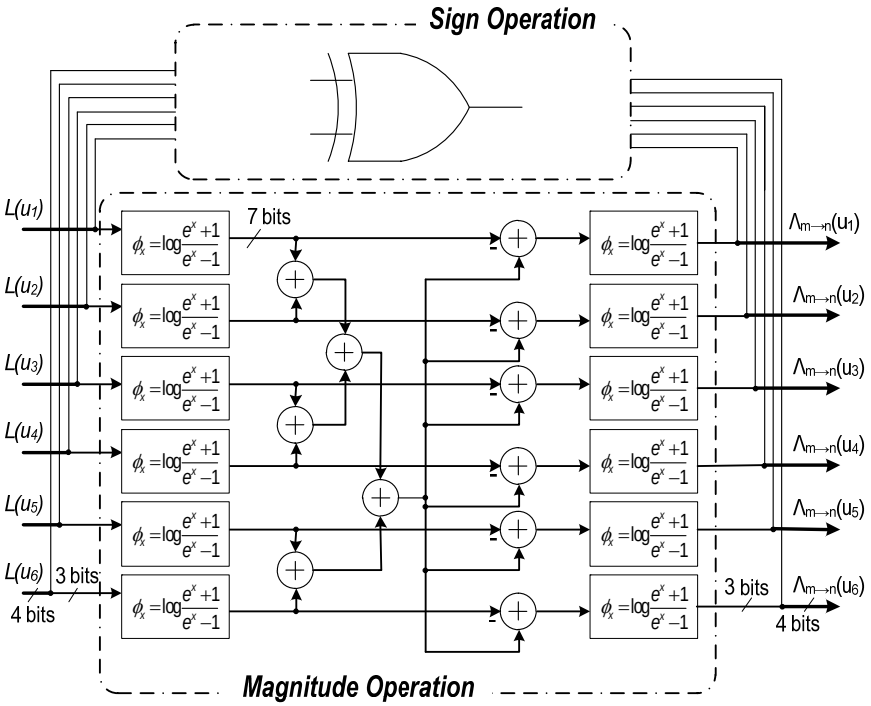

Figure 1. The traditional Log-SPA check node architecture.

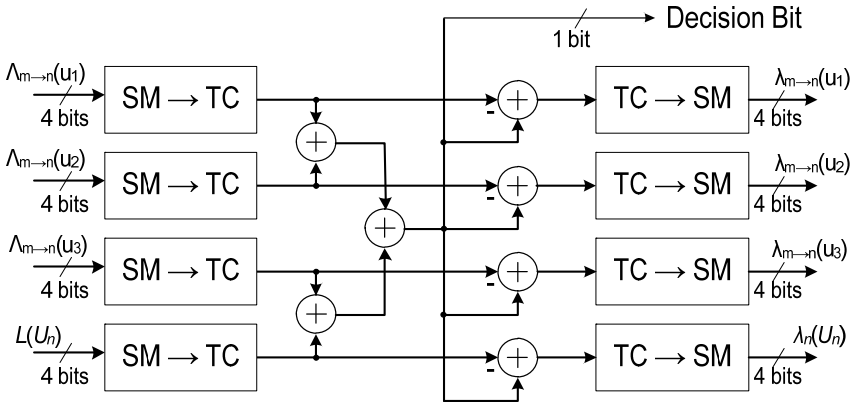

Figure 2. Datapath of traditional SPA variable node.

The quantizers restrict the input range for incoming LLR values, rounding off the input values to the nearest value, the Q-LLR, in a set ofquantization levels. Table 1 is an example of a 4-bit signed quantization table. While the message is updated in the check node by (6), the input values have to be mapped into a hyperbolic tangent (*tanh*) function. Then, the calculated check node is converted back to LLR by (8) for variable node message updating.

Table 1. Quantization Range

LLR	Q-LLR
≤ 000.101 (≤ 0.625)	000 (0)
000.110--001.101 (0.75--1.625)	001 (1)
001.110--010.101 (1.75--2.625)	010 (2)
010.110--011.101 (2.75--3.625)	011 (3)
011.110--100.101 (3.75--4.625)	100 (4)
100.110--101.101 (4.75--5.625)	101 (5)
≥ 101.101 (≥ 5.625)	110 (6)

The function of*tanh* is usually implemented by an LUT. Based on the characteristic curve of (7), which is shown in Figure 3, the input and the output mapping tables could be built as Table 2 (LUT 3-7) and Table 3 (LUT 7-3), respectively. Noticeably, the bit length of the quantized LLR value is a trade-off between hardware complexity and BER performance.

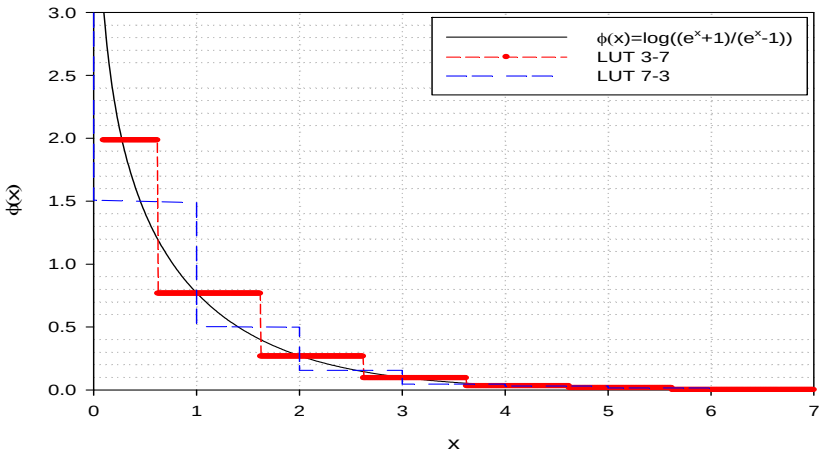

Figure 3. Quantization curve for lookup-tables.

Table 2. 3-bit to 7-bit LUT

Q-LLR	$\phi(x)$
000 (0)	1.111111 (1.984375)
001 (1)	0.110001 (0.765625)
010 (2)	0.010001 (0.265625)
011 (3)	0.000111 (0.109375)
100 (4)	0.000010 (0.031250)
101 (5)	0.000001 (0.015625)
110 (6)	0.000000 (0.000000)
111 (7)	0.000000 (0.000000)

Table 3. A 7-bit to 3-bit LUT Example for Traditional Architecture

Q-LLR		$\phi(x)$
1.111111--1.100000 (1.984375--1.500000)		000 (0)
1.011111--0.100000 (1.484375--0.500000)		001 (1)
0.011111--0.001010 (0.484375--0.156250)		010 (2)
0.001001--0.000011 (0.140625--0.046875)		011 (3)
0.000010	(0.031250)	100 (4)
0.000001	(0.015625)	101 (5)
0.000000	(0.000000)	110 (6)

The logarithm domain operation, a practical solution, replaces the multiplicative operation by additiveoperation to avoid intensive multiplicative calculations on conventional SPA. consequently, the speed bottleneck and the hardware cost are shifted from multipliers to adders, and become significant on large LDPC decoders. the min-sum algorithm [10] is the main LDPC hardware architecture which reduces hardware cost effectively. however, its BER performance is unsatisfactory. based on the original Log-SPA and the min-sum hardware reduction essences, some simple logic gates are proposed to replace the traditional adders in the check node. In the next section, the proposed idea is explained in detail.

4. The Proposed Scheme

4.1. Quasi Binary Weighting Addition Check Node

As shown in Figure 1, when the input LLR values are sent into the check node for addition operation, they are first mapped into the desired quantized value shown in Table 2. The corresponding value in Table 2 is the *tanh* function after reasonable quantization. Based on easier circuits to replace the traditional adder, the value characteristics of the*tanh* function in Table 2 are further observed. The characteristics of the summed value could be replaced by the OR gate in the addition operation of the check node (few value errors are further discussed). OR operation is performed directly, with bits of the OR gate under the same binary weighting. In the OR operation shown in the middle of Figure 4, the output of the bottom right OR gate is $\phi_1[6] \mid \phi_2[6] \mid \phi_3[6] \mid \phi_4[6] \mid \phi_5[6]$, in which $\phi_6[6]$ is missing because the upload of $\Lambda_{m\_n}(u_6)$ would not contain the value of $\phi_6[6:0]$. The proposed new circuits do not really execute the addition in traditional binary adders. Besides, in the application of the LDPC decoder, having the OR gate close to the binary addition operation is possible because of the value characteristics in the mapping table. The proposed new method is therefore called quasi binary weighting addition. In terms of overall operation, the 7-bit updated values should be mapped to 3-bit values before the quasi binary weighting addition check node value is updated to a variable node. Table 4, modified from the traditional one (e.g., Table 3), is doing the mapping for updated values, where ”x” stands for “don't care”. Further, all values in the check node are the estimated probabilities that do not require precise calculation as long as iterations can be done with convergence. As a result, the check node hardware complexity can be reduced with few computational errors. The use of logical OR-gates, shown in Figure 4, is proposed to replace the binary adders in the check node with simpler logic circuits.

Table 4. A Modified 7-bit to 3-bit LUT

$\phi(x)$	Q-LLR
1xxxxxx	000 (0)
01xxxxx	001 (1)
001xxxx	010 (2)
0001xxx	011 (3)
00001xx	100 (4)
000001x	101 (5)
0000001	110 (6)
0000000	111 (7)

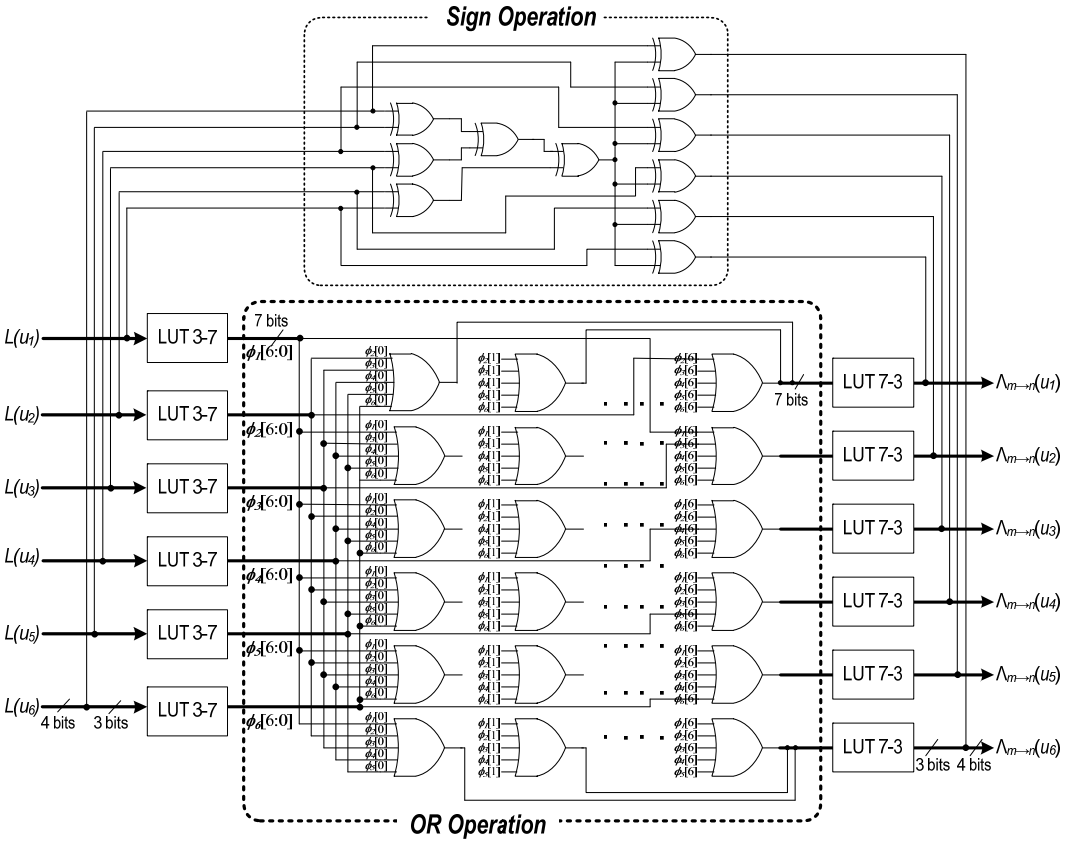

Figure 4.The pseudo-carry circuit.

The performance of the proposed quasi binary weighting addition for performing the SPA has been evaluated by numerical experiments. The code used for simulation is (1008, 3, 6) regular LDPC code [19]. For comparisons, two algorithms, namely, the original SPA, LUT 3-7 [9], and the SMA [7], are used to observe any improvement on the performance of the proposed scheme. Simulation results are shown in Figure 9, where the BER performance versus the signal-to-noise ratio (SNR) is compared among the SPA, LUT 3-7 [9], the SMA [7], and the proposed quasi binary weighting addition. The maximum iteration number is set to eighty in the simulation. In Figure 9, BER performance of the proposed quasi binary weighting addition outperforming the SMA does not lose too much. As a result of the new check node architecture that uses simple addition and a regular LUT, the OR operation architecture has about 15% hardware reduction when compared with the traditional LUT 3-7 architecture based on a 0.18μ m standard cell library. In addition, the proposed quasi binary weighting addition architecture can work up to 100 MHz decoding speed as the proposed architecture uses concise logical gates to perform additions in the LDPC.

4.2. Quasi Binary Weighting Addition With Pseudo-Carry

Without the carry sign in addition, our new addition method in check nodes leads to some numerical errors and bad BER performance. To compensate for the summation error in the OR operation architecture, a concise circuit is developed in place of the traditional adder carry-out circuits. A pseudo-carry circuit is then designed to deal with loss carry-out problem. The pseudo-carry circuit can be simply implemented by combinational logic circuits. Figure 5 is a design example. When the quasi binary weighting adder receives the mapped values, the pseudo-carry circuits detect the logical high for each same binary weighting column of bits. If a column of bits has more than one "1" , a pseudo-carry-out circuit is sent to the higher-order-bit column of the succeeding OR operation.

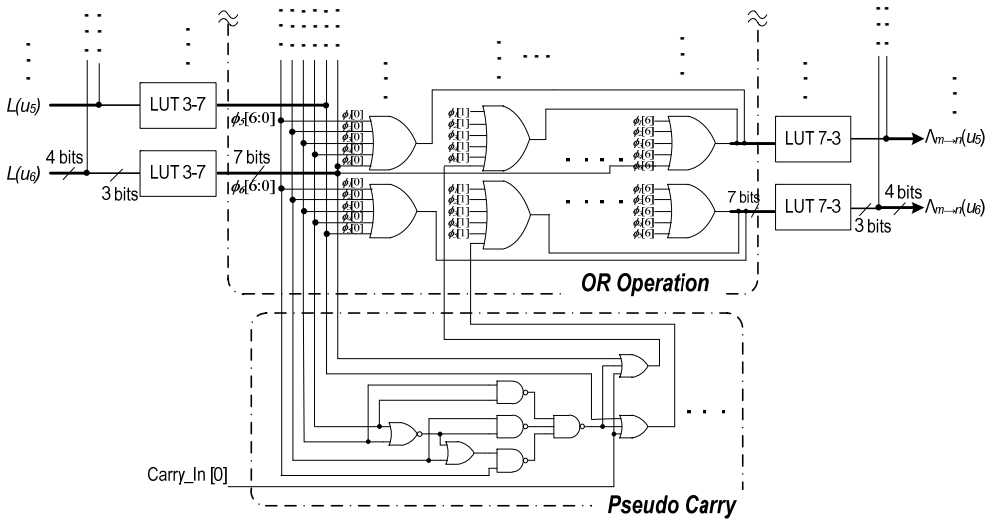

Figure 5.Proposed quasi binary weighting addition with pseudo-carry circuits.

Numerical simulation results, shown in Figure 9, illustrate that the quasi binary weighting addition, LDPC decoder, equipped with pseudo carry, has only 0.08 dB BER loss when compared with the log-SPA parallel architecture [9]. On account of the added pseudo-carry circuit, the total LDPC hardware has about 6% overhead compared with the original quasi binary weighting addition, yet the BER performance improves to about 0.2 dB over the quasi binary weighting addition architecture. Although extra pseudo-carry circuits increase the hardware, the total hardware cost is still less than adder-based log-SPA. Further, the new architecture, which adopts proper LUT, requires less hardware than selector-based sign-min.

5. Design for Hardware Reductions

A unified TC-based architecture is proposed to simplify the numeric transfer between TC and sign magnitude in traditional log-SPA, and to reduce the hardware without affecting BER performance loss. Further, the bit-width of LUTs is examined to come out with optimum bits for the data path.

5.1. Low Hardware Cost LUT

Figure 6 depicts BER performance versus bit-width of mapping table from 3-bit to 7-bit for Φ (x), using the (1008, 504) matrix [19]. The 3-bit to 4-bit mapping table (LUT 3-4) and the 3-bit to 7-bit (LUT 3-7) mapping tables have only 0.1 dB BER performance error. However, they have a three bit difference influencing the succeeding great deal of adders. The synthesized result shows that adopting LUT 3-4 has about 16% total area reduction in the LUT 3-7. In consideration of hardware cost, LUT 3-4, shown in Table 6, is adopted rather than using the LUT 3-7 used in [9] .

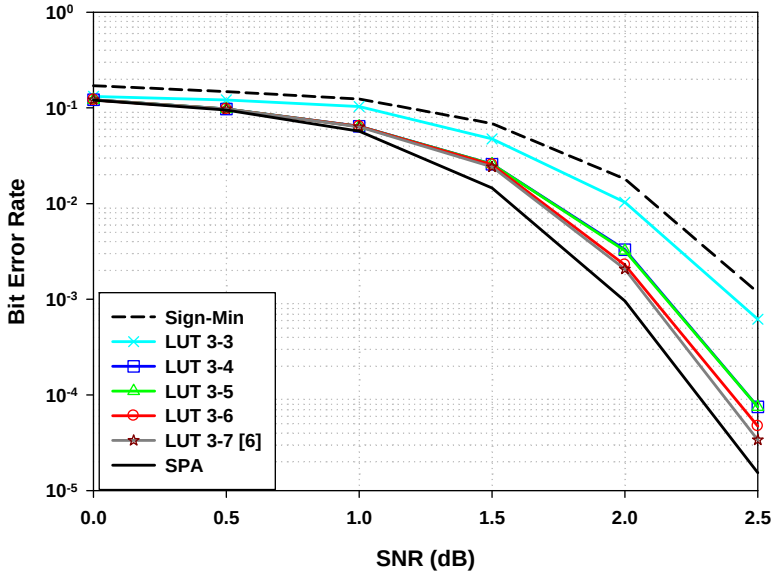

Figure 6. BER performance of different LUT for (1008, 504) LDPC code with eighty decoding iterations.

Table 6. 3-bit to 4-bit LUT

Q-LLR	$\phi(x)$
000 (0)	1.111 (1.875)
001 (1)	0.110 (0.750)
010 (2)	0.010 (0.250)
011 (3)	0.001 (0.125)
100 (4)	0.000 (0.000)
101 (5)	0.000 (0.000)
110 (6)	0.000 (0.000)
111 (7)	0.000 (0.000)

Table 7. New 3-bit to 4-bit LUT

Q-LLR	$\phi(x)$
000 (0)	1.111 (1.875)
001 (1)	0.110 (0.750)
010 (2)	0.010 (0.250)
011 (3)	0.001 (0.125)
100 (4 , -4)	0.000 (0.000)
101 (-3)	0.001 (0.125)
110 (-2)	0.010 (0.250)
111 (-1)	0.110 (0.750)

5.2. TC Based Message Passing Architecture

As illustrated in Figure 7 (a), message passing between check node and variable node is sign-magnitude in the traditional log-SPA architecture [9], where the variable node processes the TC values and the check node processes the SM values, so that the numeric transform is processed twice during one updating. If message passing is TC-based numbers, as shown in Figure 7 (b), the variable node does not need SM values for TC transfer. Consequently, after summation processing at the variable node, the calculated TC values are then sent to the check node directly. The only numeric transfer left for the TC architecture is on the check node outputs, and the LUT is modified as shown in Table 7, including the input LLR. Figure 8 shows the effectiveness of hardware reduction of TC message-passing architecture. The new proposed architecture is smaller than traditional LUT 3-7 [9] and LUT 3-4 hardware by 26% and

12% respectively. The comparison is made by the (1008, 504) matrix [19] with 0.18 μm standard cell library, and the decoding iteration set at eighty.

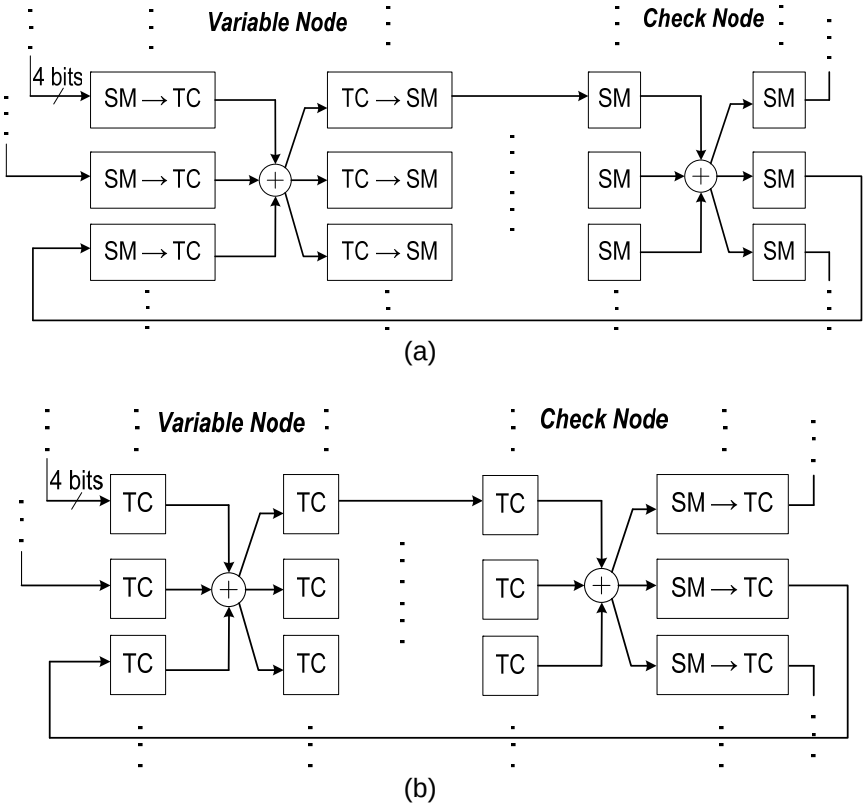

Figure 7. Datapath of variable node and check node message passing, (a) traditional architecture, (b) proposed TC based architecture.

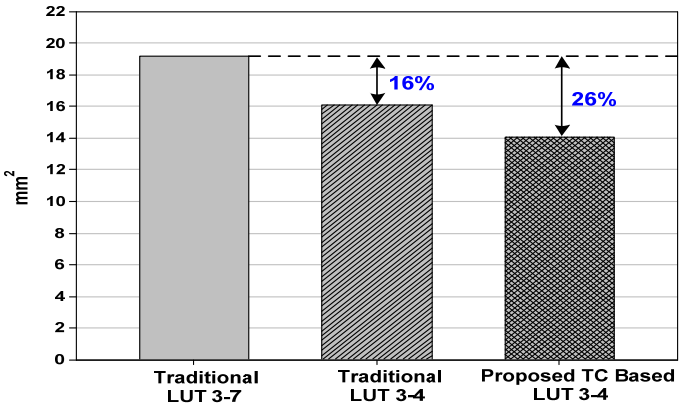

Figure 8. Hardware reductions with new TC based message passing architecture.

5.3. Low Hardware Cost Designs

For low hardware cost ASIC implementation, not only was the quasi binary weighting addition architecture adopted, but the optimized bit-width of LUTs was also employed for better hardware efficiency. The new low-cost log-SPA LDPC decoder, composed of OR pseudo-carry

architecture, the TC-based message passing data path, and the LUT 3-4 mapping table, are implemented. Simulation results, shown in Figure 9, demonstrate the new decoder (OR pseudo-carry) performing a reasonable BER. When comparing it with the original SPA, it only has 0.2 dB loss at 10^{-3} BER. Hardware comparisons, using the (1008, 504) matrix [19] and a 0.18 μ m standard cell library under the same logic synthesis conditions, are shown in Figure 10. For the lowest hardware cost, the quasi binary weighting addition with TC-based architecture is the suggested combination.

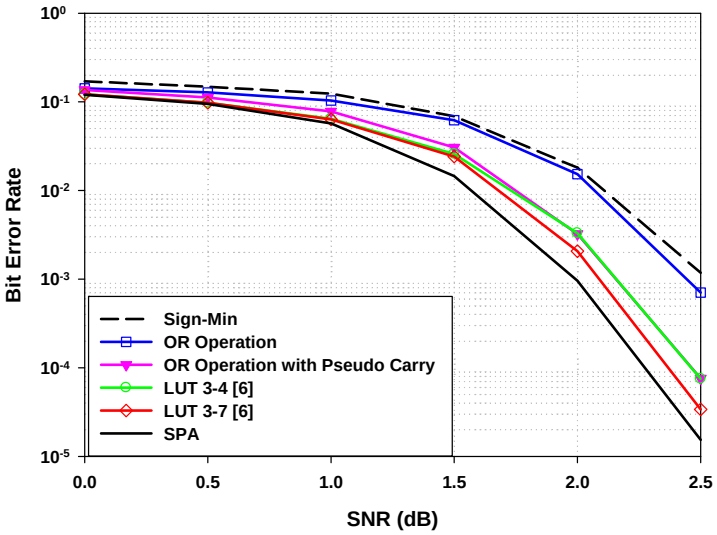

Figure 9. BER comparison for (1008, 504) LDPC code with eighty decoding iterations.

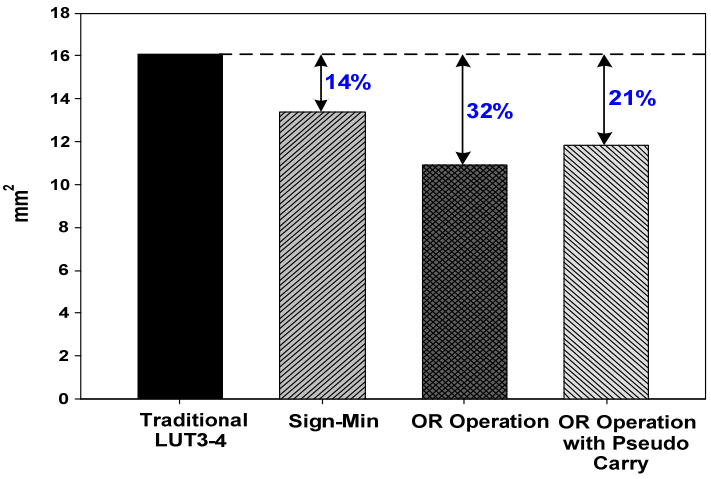

Figure 10. Area comparison of low hardware cost architectures using 0.18µm standard cell library.

6. Conclusion

For error correction hardware implementation, such as in an LDPC decoder, errors are allowed in the numerical algorithm. According to the characteristic, the value follows certain rules when the numerical range of mapping table is slightly adjusted, and replaces traditional adders with simple OR gate to execute additions. Numerical simulation results further prove that the proposed quasi binary weighting addition could successfully proceed with LDPC decoding. Moreover, the modified TC-based message passing techniques can effectively simplify the

numeric transfer hardware. The proposed hardware reduction techniques can be adopted independently. For example, the whole decoder of quasi binary weighting addition, with TC-based architecture and LUT 3-4, presents the best hardware efficiency. For better BER performance, adding pseudo-carry circuits in quasi binary weighting addition and using LUT 3-7 could fulfill the demand. Compared with traditional log-SPA architecture, the new proposed structure has effectively made a hardware reduction of up to 32%. Further, compared with the simplest selector-based sign-min, the new architecture has 11% to 18% hardware reductions and improves the coding gain from about 0.3 to 0.1 dB at a low BER value.

References

[1] RG Gallager. Low-Density Parity-Check Code. *IRE Trans. Inform. Theory.* 1962; 1(T-8): 21-28.
[2] Bane Vasic and Erozan M. Kurtas. Coding and Signal Processing for Magnetic Recording Systems. New York: CRC Press. 2004.
[3] DJC Mackay. Good Error-Correcting Codes Based on Very Sparse Matrices. *IEEE Trans. Inform. Theory*. 1999; 45: 399-431.
[4] SY Chung, GD Forney, TJ Richardson, and R Urbanke. On The Design of Low-Density Parity-Check Codes within 0.0045 dbof The Shannon Limit. *IEEE Commun. Lett*. 2001; 5: 58-60.
[5] Hendra Setiawan, Yuhei Nagao, Masayuki Kurosaki, and Hiroshi Ochi. IEEE 802.11n Physical Layer Implementationon Field Programmable Gate Array. *TELKOMNIKA*. 2011; 10(1): 67-74.
[6] Ali M Fadhil, Haider M AlSabbagh. Performance Analysis for Bit Error Rate of DS- CDMA Sensor Network Systems with Source Coding. *TELKOMNIKA*. 2012; 10(1): 165-170.
[7] XY Hu, E Eleftheriou, DM. Arnold, and A Dholakia. *Efficient Implementations of The Sum-Product Algorithm for Decoding LDPC Codes.* Proc. IEEE Globecom. 2001: 1036-1036.
[8] S Papaharalabos, P Sweeney, BG Evans, G Albertazzi, A Vanelli-Coralli, and GE Corazza. *Performance Evaluation of a Modified Sum-Product Decoding Algorithm for LDPC Codes*. Proc. ISWCS. 2005: 800-804.
[9] AJ Blanksby and CJ Howland. A 690-mW 1 Gb/S 1024-B Rate-1/2 Low-Density Parity-Check Code Decoder. *IEEE J. Solid-State Circuits*. 2002; 37(3): 404-412.
[10] CC Lin, KL Lin, CC Chung, and CY Lee. *A 3.33Gb/S (1200,720) Low-Density Parity Check Code Decoder*. Proc. 31st ESSCIR. 2005: 211-214.
[11] PH Yang, JC Chen, YT Chan, and MY Lin. *A Simplified Addition Operation Log-SPA LDPC Decoder.* Proc. 14th APCC. 2008: 1-4.
[12] Y Zhang, Z Wang, and KK Parhi. *Efficient High-Speed Quasi-Cyclic LDPC Decoder Architecture.* Proc. 38th ACSSC. 2004: 540-544.
[13] S Tong, P Wang, D Wang and X Wang. Box-minus operation and application in sum-product algorithm. *Electron. Lett.* 2005; 41: 197-198.
[14] S Papaharalabos, and PT Mathiopolous. Simplified Sum-Product Algorithm for Decoding LDPC Codes with Optimal Performance. *Electron. Lett*. 2009; 45: 116-117.
[15] Wen Ji, T Ikenaga, and S Goto. *A High Performance LDPC Decoder for IEEE802. 11n Standard*. Proc. ASP-DAC. 2009: 127-128.
[16] G Masera, F Quaglio, and F Vacca. Implementation of a Flexible LDPC Decoder. *IEEE Trans. Circuits Syst. II*. 2007; 54: 542-546.
[17] DJC MacKay. Online Database of Low-Density Parity-Check Codes. Available at http://wol.ra.phy.cam.uk/mackay/codes/data.html.
[18] DJC MacKay. Good Error-Correcting Codes Based on Very Sparse Matrices. *IEEE Trans. Inform. Theory*. 1999; 45: 399-431.
[19] TJ Richardson and RL Urbanke. The Capacity of Low-Density Parity Check Codes Under Message-Passing Decoding. *IEEE Trans. Inform. Theory*. 2001; 47: 599-618.
[20] MPC Fossorier, M Mihaljevic, and H Imai. Reduced Complexity Iterative Decoding Of Low Density Parity Check Codes Based On Belief Propagation. *IEEE Trans. Commun.* 1999; 47: 673-680.
[21] XY Hu, E Eleftheriou, DM Amold, and A Dholakia. *Efficient Implementation of the Sum-Product Algorithm for Decoding LDPC Codes.* Proc. Int. Conf. GLOBECOM. San Antonio 2001; 2: 1036–1036.
[22] Chien-Ching Lin, Kai-Li Lin, Hsie-Chia Chang, Chen-Yi Lee. *A 3.33Gb/S (1200,720) Low-Density Parity Check Code Decoder.* Proc. Int. Conf. ESSCIRC. 2005: 211-214.
[23] Ahmad Darabiha, AC Carusone and FR Kschischang. *A Bit-Serial Approximate Min-Sum LDPC Decoder and FPGA Implementation*. Proc. Int. Conf. ISCAS. 2006.
[24] Qi Wang, Shimizu K, Ikenaga T, Goto S. *A Power-Saved 1Gbps Irregular LDPC Decoder Based on Simplified Min-Sum Algorithm.* Proc. Int. Conf. VLSI-DAT. 2007: 25-27.
[25] Gunnam KK, Choi GS, Yeary MB. *A Parallel VLSI Architecture for Layered Decoding for Array LDPC Codes.* Proc. Int. Conf. VLSI-Design. 2007: 738-743.
[26] IEEE Std 802.16 g-2007. Air Interface for Broadband Wireless Access Systems. New York: IEEE Press 2007.