An LDPC Decoder Chip Based on Self-Routing Network for IEEE 802.16e Applications

Chih-Hao Liu, Shau-Wei Yen, Chih-Lung Chen, Hsie-Chia Chang, Chen-Yi Lee, *Member, IEEE*, Yar-Sun Hsu, and Shyh-Jye Jou

***Abstract*—An LDPC decoder chip fully compliant to IEEE 802.16e applications is presented. Since the parity check matrix can be decomposed into sub-matrices which are either a zero-matrix or a cyclic shifted matrix, a phase-overlapping message passing scheme is applied to update messages immediately, leading to enhance decoding throughput. With only one shifter-based permutation structure, a self-routing switch network is proposed to merge 19 different sub-matrix sizes as defined in IEEE 802.16e and enable parallel message to be routed without congestion. Fabricated in the 90 nm 1P9M CMOS process, this chip achieves 105 Mb/s at 20 iterations while decoding the rate-5/6 2304-bit code at 150 MHz operation frequency. To meet the maximum data rate in IEEE 802.16e, this chip operates at 109 MHz frequency and dissipates 186 mW at 1.0 V supply.**

***Index Terms*—Decoder architectures, IEEE 802.16, iterative decoders, LDPC codes, phase-overlapping, self-routing, WiMax.**

I. Introduction

LOW-DENSITY parity-check (LDPC) code, a linear block code defined by a very sparse parity-check matrix, was firstly introduced by Gallager [1]. The LDPC code has been proved to approximate the Shannon limit based on the iterative sum-product algorithm (SPA) and is capable of parallel implementation for higher decoding speed. Newly high-speed communication systems such as IEEE 802.11n, UWB [2], DVB-S2 [3], and IEEE 802.16e [4], [5] have considered employing LDPC codes to enhance their performance. LDPC code can be described by a bipartite graph, in which the bit nodes and the check nodes represent the information bits and the parity check equations respectively. Gallager's two-phase message passing algorithm [1] decodes a codeword by updating the messages between check nodes and bit nodes iteratively. Since the data dependency between check nodes and bit nodes, results in a limited decoding throughput. Turbo-decoding message-passing (TDMP) based on the soft-input soft-output (SISO) decoder was proposed in [6] to allow updating both check node and bit node concurrently. The trellis-based TDMP algorithm was applied for the specific 2048-bit, (3,6)-regular architecture-aware LDPC (AA-LDPC) [7], [8]. However, the complexity for transforming parity check matrix to trellis will be enhanced in an irregular LDPC code. In the IEEE 802.16e system [9], an irregular parity check matrix can be decomposed into several cyclic-shifted identity or zero matrices. We propose a phase-overlapping message passing algorithm for the LDPC decoder in this paper. The phase dependence between nodes in different rows (or sub-matrices) during decoding operation can be decoupled. As a result, the messages generated by check nodes in the pervious row can be passed to bit nodes immediately. Throughput can be improved by increasing the processing elements of the sub-matrix or the row.

Signal routing congestion is another challenge in implementing the message passing circuits of LDPC decoders. Fully parallel LDPC decoder for 1024-bit, rate-1/2 LDPC code with specified physical routing algorithm has been proposed in [10]. Partial-parallel LDPC decoders have been reported to reduce connections among edge nodes [11]–[13]. Bi-directional crossbar switch was exploited for regular LDPC decoders by fixed size forward and backward switch networks [8]. Note that signal routing congestion may constrain the crossbar switch size due to routing path conflict. The applied parity check matrix is irregular, and includes variable sizes of sub-matrices and code rates. Matrix permutation is also applied to transform the original parity check matrix into the architecture-aware structure [14]. The decoding process for multi-rates and multi-sizes LDPC codes in IEEE 802.16e is irregular and difficult to support all code rates under variable matrix sizes [4]. Flexible barrel shifter is applied to switch variable size messages for IEEE 802.16e LDPC decoders [5]. With only one 96-size permutation network, we propose a self-routing switch network that can merge 19 different sub-matrices sizes as defined in IEEE 802.16e [15].

The phase-overlapping message passing algorithm is proposed to decouple the architecture dependence among nodes of different rows, leading to improve overall decoding throughput. Moreover, a self-routing mechanism is developed to resolve the inherent blocking issue in switch network, where source messages are combined with routing information during permutation. As a result, signal routing congestion in the variable size switch network can be reduced significantly with only one permutation network (the size is 96) that provides 19 different switch network sizes.

The remainder of this paper is as follows. Section II introduces IEEE 802.16e LDPC code structure and the phase-overlapping message-passing algorithm. The corresponding architecture and memory structure are presented in Section III. Section IV describes the proposed architecture of self-routing

Manuscript received April 17, 2007; revised October 17, 2007.

C.-H. Liu and Y.-S. Hsu are with the Department of Electrical Engineering, National Tsing-Hua University, Hsinchu, Taiwan 30013, R.O.C. (e-mail: jrhaulu@gmail.com).

S.-W. Yen, C.-L. Chen, H.-C. Chang, C.-Y. Lee, and S.-J. Jou are with the Department of Electronics Engineering, National Chiao-Tung University, Hsinchu, Taiwan, R.O.C. (e-mail: cylee@faculty.nctu.edu.tw).

Digital Object Identifier 10.1109/JSSC.2007.916610

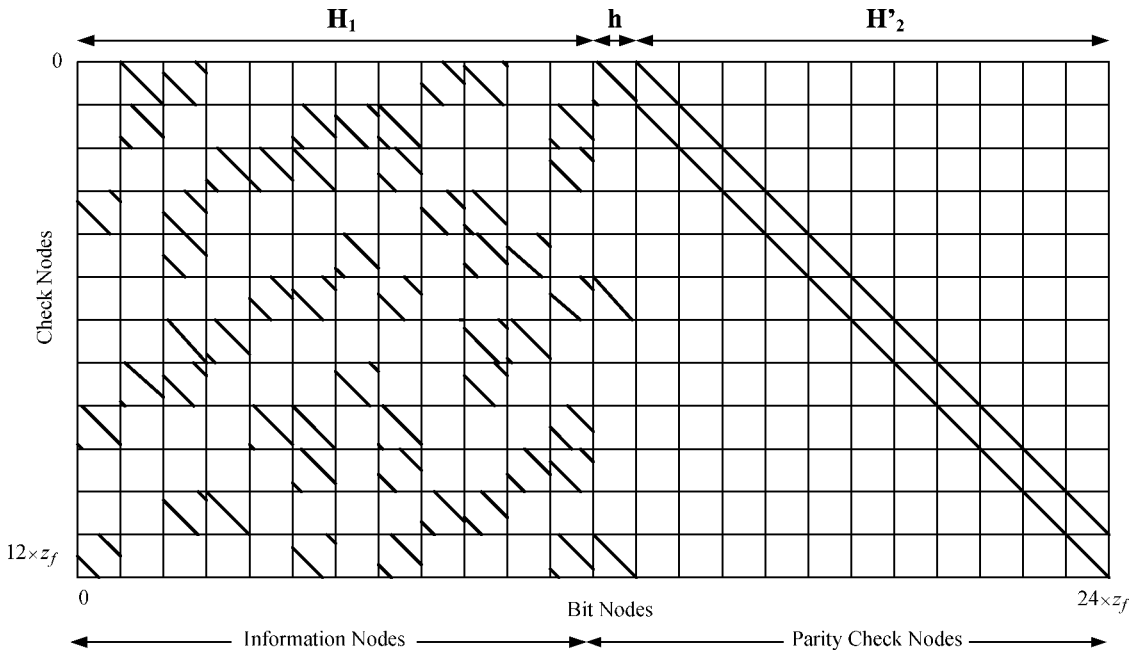

Fig. 1. Structure of the parity check matrix for a rate 1/2 IEEE 802.16e LDPC code.

switch network which can cover all sizes of sub-matrix as defined in IEEE 802.16e. Finally, the measurement results of LDPC decoder chip are shown in Section V and the final conclusion is presented in Section VI.

II. LDPC Codes and Decoding Algorithm

A. Code Structure of 802.16e

In the IEEE 802.16e system, the sub-matrix size is defined by the expansion factor z_f. The parity-check matrix $\mathbf{H}$ can be decomposed into several $z_f \times z_f$ sub-matrices, and each one is either the zero matrix or the cyclic-shifted identity matrix [9]. The parity check matrix size is based on both of the code rate and z_f. The 19 variable expansion factors defined in the IEEE 802.16e specification [9] range from 24 to 96 with an increment of four. Note that the size of matrix $\mathbf{H}$ is $M \times N$ where M is the number of parity check equations, and N is the code length. Moreover, there is a $m_b \times n_b$ base matrix $\mathbf{H}_\mathrm{b}$ with $m_b = M/z_f$ and $n_b = N/z_f$, where m_b and n_b are the sub-matrix number in a column and row respectively. The code rate is determined by the value of m_b/n_b, where the maximum value of m_b and the constant value of n_b defined in the IEEE 802.16e system are 12 and 24 respectively. The $\mathbf{H}$ is extended from the base matrix $\mathbf{H}_\mathrm{b}$ by replacing each 1 in $\mathbf{H}_\mathrm{b}$ with a $z_f \times z_f$ circular right shifted identity matrix and each 0 in $\mathbf{H}_\mathrm{b}$ with a $z_f \times z_f$ zero matrix. A structure for rate-1/2 parity check matrix $\mathbf{H}$ is shown in Fig. 1. Note that $\mathbf{H}$ can be partitioned into two parts: $\mathbf{H_1}$ and $\mathbf{H_2}$, where $\mathbf{H_1}$ is the information nodes and $\mathbf{H_2}$ is the parity check nodes. $\mathbf{H_2}$ can also be partitioned into two parts: $\mathbf{h}$ and $\mathbf{H_2'}$, where $\mathbf{H_2'}$ has a dual-diagonal structure.

B. Min Sum Decoding Algorithm

The belief-propagation (BP) algorithm [1], [16] provides an efficient and powerful approach to decode LDPC codes. Let S_m be the event that the parity check equation for the check node m is satisfied. In each decoding iteration, the check node m updates its outgoing message by the probability $P\{S_m|x_{n'} = b\}$, for all $n' \in N(m)$ and $b \in \{0,1\}$. After the bit node n receives all the messages from the check nodes in $M(n)$, the bit node updates its message according to the probability $P\{x_n = b|S_{m'}, y_n\}$, where $m' \in M(n)$, and y_n is the value received from the channel. Each bit node can accumulate more reliable information from the others by exchanging information between the bit nodes and the check nodes iteratively. The iterative decoding process operates until a valid codeword is found or the decoding iteration exceeds a predefined number. If the probabilistic messages are represented by log-likelihood ratios (LLR), the belief-propagation (BP) decoding can be described as follows:

1) **Initialization:**
Under the assumption of equal priori probability, $P(b_n = 0) = P(b_n = 1) = 0.5$ the decoder calculates, the intrinsic information of the bit node n, by

$$p_n = \log \frac{P(y_n|b_n = 0)}{P(y_n|b_n = 1)}. \tag{1}$$

The message from bit node n to check node m, denoted by q_{nm}, is initialized by $q_{nm} = p_n$, while the message from check node m to bit node n, denoted by r_{mn}, is set to zero.

2) **Iterative Decoding:**
   (a) Bit node updating:
   Bit node n updates the message to check node m by

$$q_{nm} = p_n + \sum_{m' \in \{M(n) \backslash m\}} r_{m'n} \tag{2}$$

where the set $\{M(n) \setminus m\}$ contains all elements in $M(n)$ excluding m. Meanwhile the decoder can make a hard decision that the nth bit by $x(n) = 0$ if $(p_n +$

$\sum_{m' \in \{M(n) \setminus m\}} r_{m'n}) > 0$ and $x(n) = 1$ otherwise. The decoding process stops when a valid codeword $x = [x_0, x_1, \ldots\ldots\ldots\ldots, x_N]^T$ is found while $Hx = 0$, otherwise, the decoding moves toward the phase of check node updating. If the iteration number exceeds a predefined value, the decoder claims a decoding failure and terminates the decoding procedure.

(b) Check node updating:
The check node m updates r_{mn}, the message to the bit node n, according to the messages received from $\{N(m) \setminus n\}$ in which n is excluding.

$$r_{mn} = \prod_{n' \in \{N(m) \setminus n\}} sign(q_{n'm}) \times \Psi^{-1}\left(\sum_{n' \in \{N(m) \setminus n\}} \Psi\left(|q_{n'm}|\right)\right) \quad (3)$$

$$\Psi(a) = \Psi^{-1}(a) = \log \frac{1 + e^{-a}}{1 - e^{-a}}. \quad (4)$$

The nonlinear function Ψ increases the complexity for hard implementation. Some approximation schemes had been proposed to simplify the hardware implementation for the check node operation. The min-sum algorithm [17] discards the smaller terms in the summation of (2) to approximate the check node updating by

$$r_{mn} = \prod_{n' \in \{N(m) \setminus n\}} sign(q_{n'm}) \min_{n' \in \{N(m) \setminus n\}} \{|q_{n'm}|\}. \quad (5)$$

However, there exists a performance loss between the min-sum algorithm and the BP algorithm since the min-sum algorithm always over-estimates the check node output magnitude. Several low-complexity approximations using a correction factor have then been introduced to compensate the performance loss. Moreover, in order to achieve a better performance, a normalized factor β can be applied to compensate the approximation error [17]:

$$r_{mn} \approx \prod_{n' \in \{N(m) \setminus n\}} sign(q_{n'm}) \times \left(\min_{n' \in \{N(m) \setminus n\}} \{|q_{n'm}|\} \times \beta\right). \quad (6)$$

C. Phase-Overlapping Message Passing

The parity check matrix in IEEE 802.16e system can be decomposed into at most 12 rows. Each row comprises 24 cyclic-shifted sub-matrices with 19 different sizes. Within each row, the sub-matrices are processed serially. Therefore, the throughput can be enhanced by increasing the parallelism in the computation of rows and sub-matrices.

In the SPA algorithm [1], both of the first and the second phase are initiated by the check node and the bit node respectively. After the check node operation, the messages will be delivered to the bit nodes, and then, the bit nodes accumulate the corresponding messages received from the check nodes. Thus, the decoding speed is restricted due to the data dependency. Layer decoding has been proposed to decouple the data dependency and improves the decoding speed [18]. A message passing scheme that leads to a higher decoding speed is applied to the LDPC decoder. Because the parity check matrix has been decomposed into sub-matrices, both of the first and the second phase can be overlapped. As a result, the new messages from the check nodes can be passed to the corresponding bit nodes immediately.

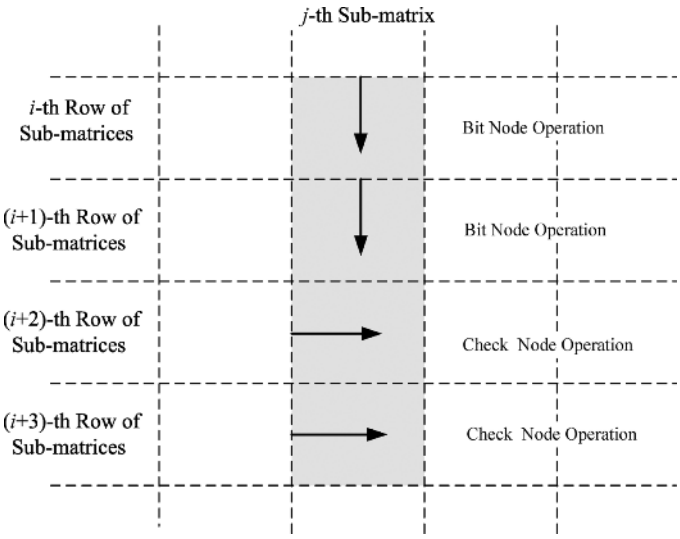

Fig. 2. Phase-overlapping message passing flow.

As shown in Fig. 2, the check nodes and the bit nodes are operated in horizontal and vertical direction respectively. In the first decoding iteration, the input messages of the check nodes are initiated as the probability ratio of the corresponding bits. The input messages of the bit nodes at the ith and the $(i+1)$th rows which are derived from the previous check node operations are accumulated. When the bit node phase reaches the ith and the $(i+1)$th rows, the check node phase can deal with the $(i+2)$th and the $(i+3)$th rows. At the end of this iteration, the accumulated sums are used to update the input messages of the check nodes for the next iteration. The iterative decoding stops when either a valid codeword is found by hard-decision result of the accumulated sums when the iteration exceeds a predefined maximum number. Comparing with SPA algorithm without layer decoding, two sub-matrices in adjacent rows can be operated simultaneously, resulting in 50% improvement in decoding throughput. After the completion of each row, the decoder accumulates the partial sum to perform the bit node operations.

III. Proposed Decoder Architecture

The architecture of the phase-overlapping message passing LDPC decoder is shown in Fig. 3. It mainly contains two edge node processor clusters in Fig. 3. The first one is the check node processor (CNP) which is used in the first phase, and the second one is the bit-node processor (BNP) which generates the sum of messages in the second phase. The number of processing cells in each cluster is 96 which can completely fulfill the maximum size of sub-matrices. Moreover, the messages are routed by a reconfigurable core network consisting of two self-routing switch

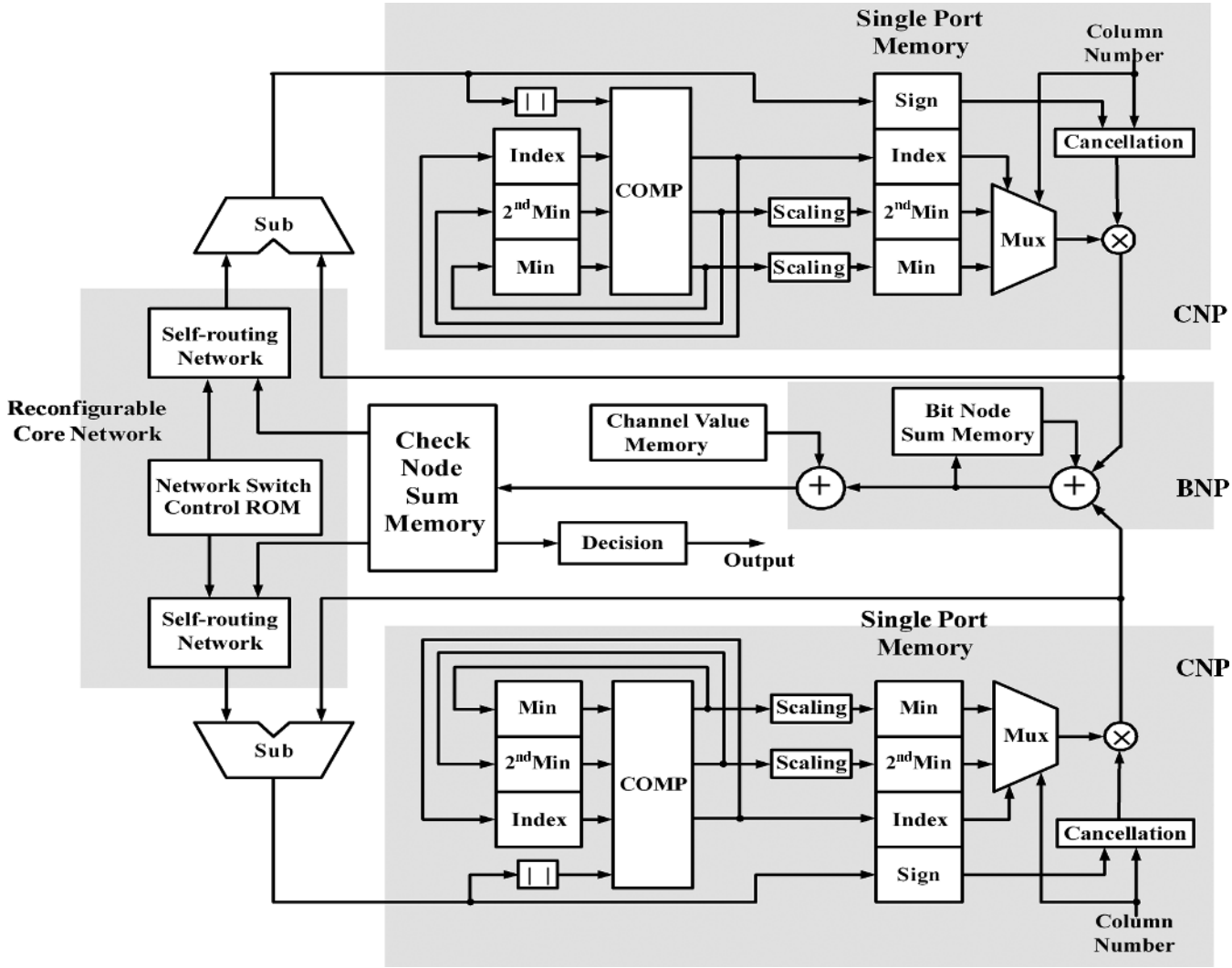

Fig. 3. Architecture of the LDPC decoder chip.

networks. The memory buffers retain all of the exchanged messages and the received channel values used by each node processor. The shift size of each sub-matrix at different code rates is stored in two ROMs.

Fig. 4 presents the memory structure containing four functional blocks. The first one is the bit node sum memory, which is for storing and updating the partial sum messages generated from bit node processor. The second one is the check node sum memory which is used for retaining the final messages sum generated during the previous iteration and will be adopted by the check node processor during the next iteration. Note that the messages in the bit node sum memory and the check node sum memory will be updated by the bit node updating engine. The third one is the minimum message memory group that stores the minimum messages correspondingly, and will be updated based on the output messages of check node processors. In the minimum message memory group, the first three memories are used to store the minimum messages, the second minimum messages, and the minimum index information respectively. The rest 24 memories are reserved for the minimum sign messages derived from different columns of the parity check matrix. The last one is the channel value memory that keeps the 6-bit channel probabilistic information. A buffer management unit is also allocated to control the channel value memory whose word length is equal to the sub-matrix size.

The phase-overlapping message passing scheduler combined with the buffer management unit arranges hardware resources of each iteration and controls the order of message passing between memories and node processors. During the decoding iteration, the configurable core network routes two 96×96 messages in parallel from memories to the node processors through two 96×96 self-routing switch networks.

A. Message Scheduling With Buffer Management

The phase-overlapping message passing scheduler manages the message passing and controls the message transfer sequence. The decoding message will strictly follow the instruction of each permutation matrix according to the phase-overlapping message passing algorithm. Moreover, memory access conflict can be avoided to reduce idle time effectively. Not only the memory access bandwidth, but also the core network utilization needs to be managed in the decoding operation. The switch network

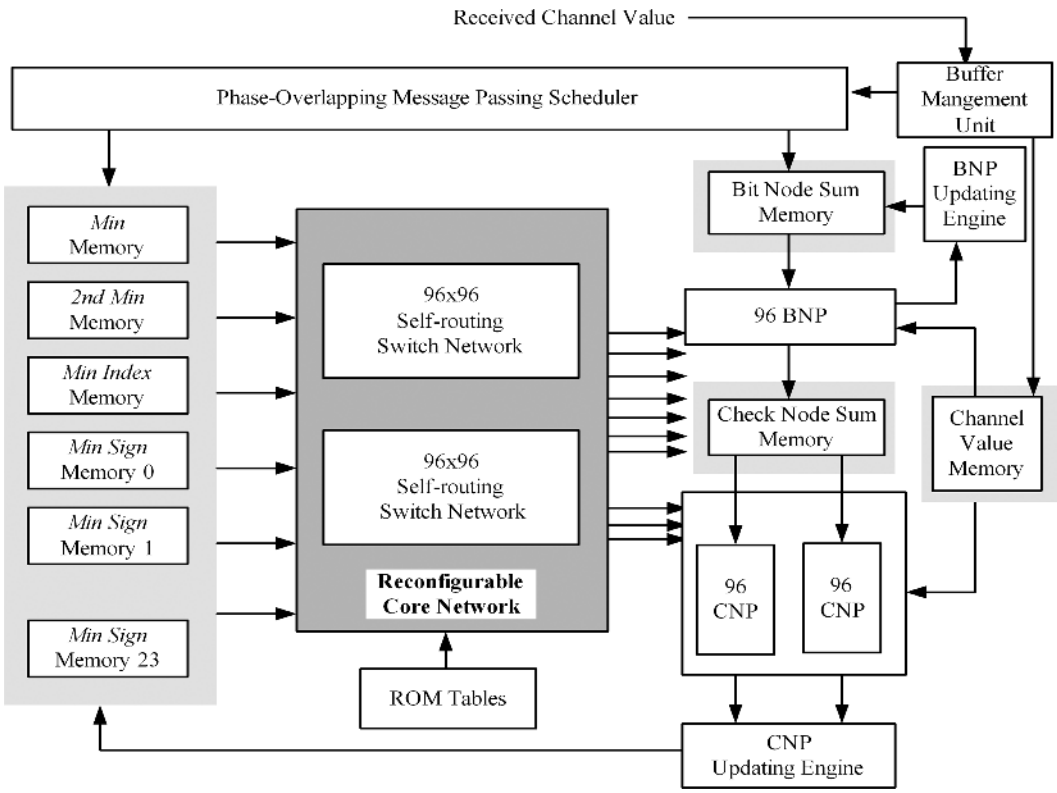

Fig. 4. Memory data structure of the proposed decoder.

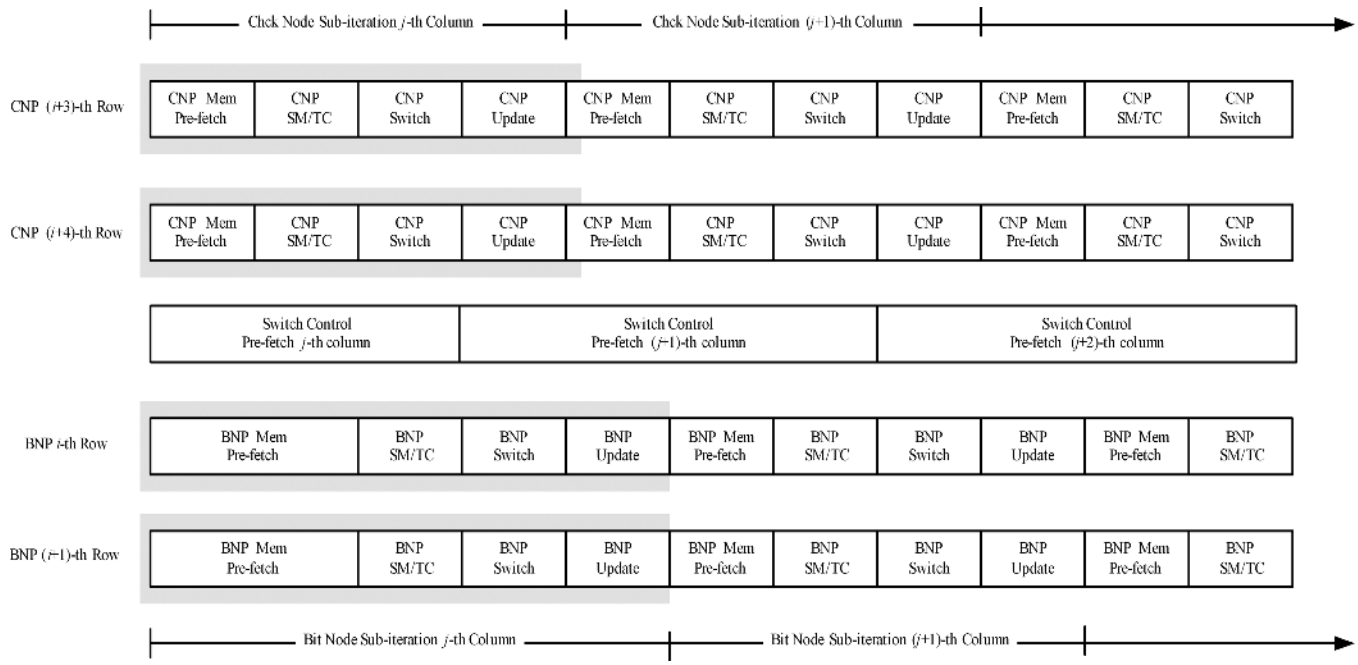

Fig. 5. The scheduling flow for message passing in the decoding process.

bandwidth can be shared by both of the check node and bit node processors. Hence, the central scheduler and the buffer management unit are applied to control the regular operation at the different rows. Two sub-matrices at adjacent rows can be operated simultaneously.

Fig. 5 shows the scheduling flow for message passing in the decoding process. This decoding process is specifically regularized and separated into four stages: memory pre-fetch, sign magnitude transfer, incoming messages switching, and out-going message updating. The memory pre-fetch process will generate the memory read address. The sign magnitude transfer converts the message from the sign magnitude (SM) notation to the 2's complement (TC). The incoming message switching process receives the messages after the format transformation, and switches the received messages to the node processors through the switch network. The out-going message

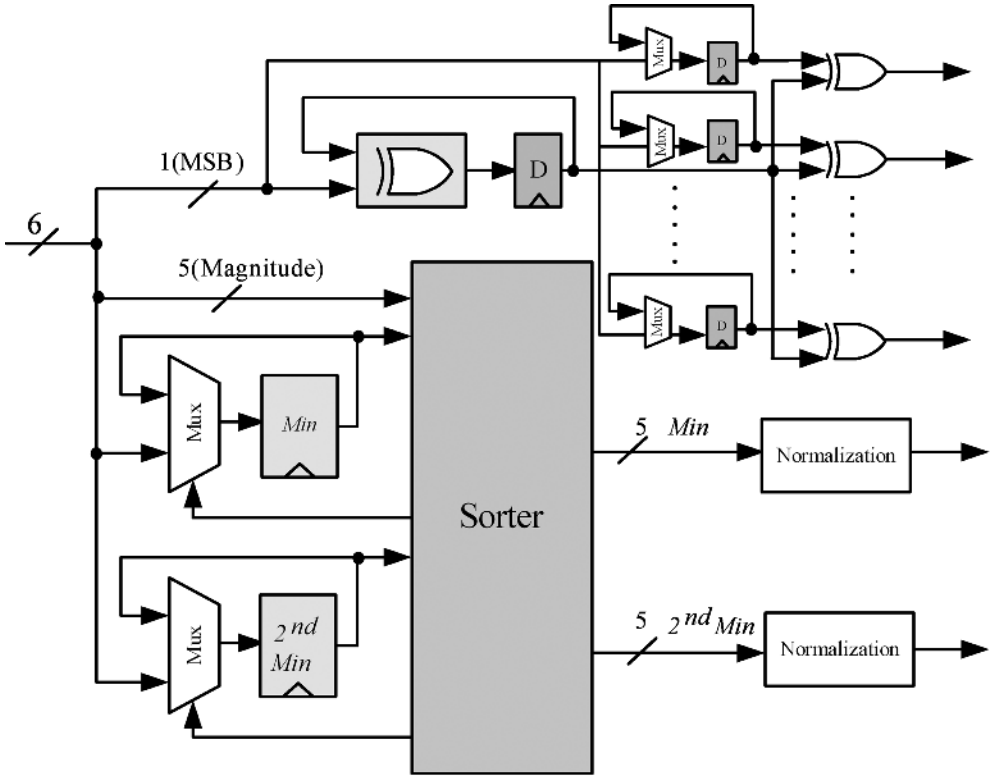

Fig. 6. Cell structure of the proposed check node.

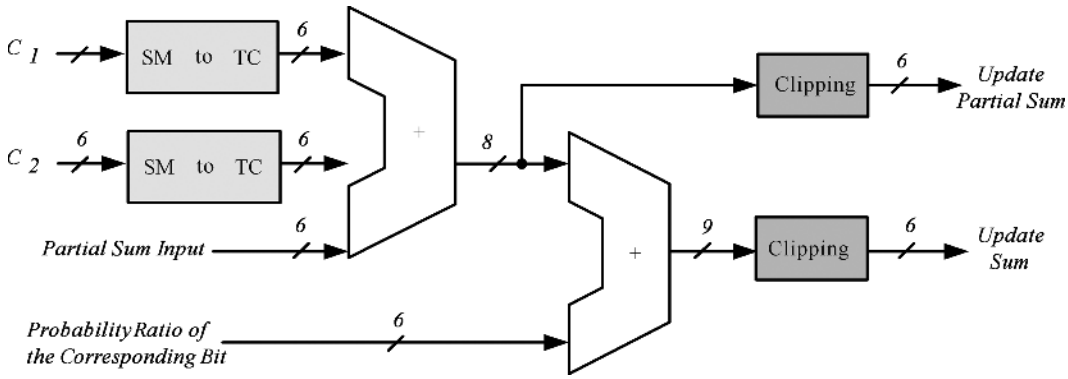

Fig. 7. Cell structure of the proposed bit node.

updating process controls the memory write address. Finally, the output messages after the computation in the edge node processors will be simultaneously updated and transferred to the message memory.

B. Node Processing Cell

The check node cell can be implemented by a sorter that searches the minimum magnitude. The sorter can be further modified to enhance the decoding speed by simultaneously updating all edges in connection with the same check node. Fig. 6 illustrates the proposed check node cell with the sign magnitude notation of 6-bit input. The check node can be divided into two parts: one is 1-bit sign-multiplication and the other is 5-bit sorter (that searches for the minimum value and the second minimum value from the inputs). The new messages generated by check nodes will be delivered to the corresponding bit nodes. The output messages of each check node are the combination of the sign bit (which is generated by the minimum sign processing element) and the new magnitude (which is either "min" or "2nd min" of the sorter).

Fig. 7 shows the block diagram of the bit node cell. The bit node cell receives the the probability ratio of the corresponding bits and the message linked to the same bit node. All inputs with the sign magnitude (SM) notation are firstly converted to the 2's complement (TC) representation and then summed up to perform the updating. The summed values are also clipped to avoid the overflow.

IV. Message Passing Switch Network

A. Variable Size Switch Network

Basically, the parity check matrix size is determined based on the code rate and the expansion factor z_f of sub-matrix. The 19 variable expansion factors in the IEEE 802.16e specification [9] range from 24 to 96 with an increment of four, and the variety causes the difficulty in applying the fixed size crossbar

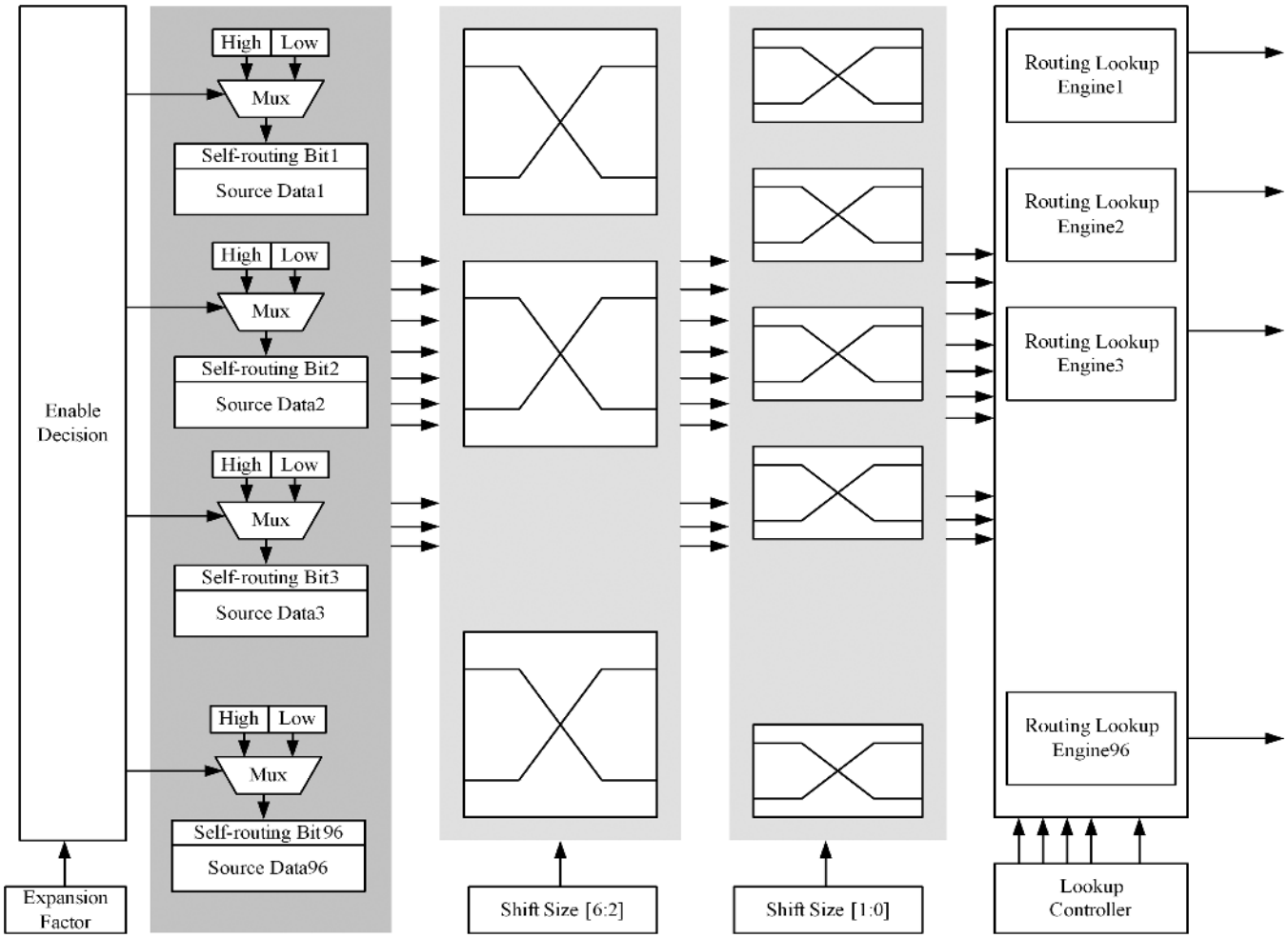

Fig. 8. Structure of the self-routing switch network.

switches, such as Banyan networks [19], Benes networks [20] and 64×64 dual bi-directional networks [8]. Multiple switches with different expansion factors lead to the signal routing congestion as well as the lower chip density [19]. The flexible barrel shifter with multi-stage multiplexers was applied to switch variable size messages for IEEE 802.16e LDPC decoders, and this will increase the signal congestion and the area of the switch network [5]. The routing decision mechanism in traditional switch network, preventing path conflict and blocking, controls both forward and backward routing paths of switch networks. But this will increase the signal routing complexity. Thus, a new shifter-based structure with only one permutation network [15] is proposed to complete the message routing for all code rates and code lengths. Each self-routing switch network is configurable for different expansion factors and shift size. Moreover, the blocking issue can be resolved by embedding self-routing information into the routing path.

B. Self-Routing With Embedded Routing Information

A self-routing switch network is proposed to enable parallel message to be routed without congestion. Fig. 8 illustrates the switch network architecture, where 96 messages are routed in parallel through the proposed four-stage switch network. Note that the size of sub-matrix is z. The message exchange operations in the four stages are as follows: the first stage is the combination of source messages with the self-routing bits, the second stage is the coarse permutation, the third stage is the fine permutation, and the fourth stage is the routing lookup scheme.

The 96 self-routing bits embedded in the routing messages are determined at the first stage, and are inserted into the corresponding source messages as shown in Fig. 8. Among the 96 source messages, the first zth message are meaningful and the others are dummy. The messages with self-routing bit equal to one means meaningful. At the second and the third stage, the 96 data, including the self-routing bits and the messages (or dummy messages), are permuted together according to the 7-bit shift size. Note that the most significant five bits are used to perform the coarse permutation by the scale of four at the second stage, and the last two significant bits are reserved to perform the fine permutation at the third stage. At the fourth stage, we have to choose z data from the 96 routed data based on the self-routing bits after the permutation. Fig. 9(a) shows that the first routing decision data constructed from the 96th to the $(96 - z + 1)$th routed data and the second routing decision data constructed from the zth to the first routed data. Fig. 9(b) illustrates the 96 lookup engines and compares the corresponding self-routing bits in parallel according to the expansion factor and shift size.

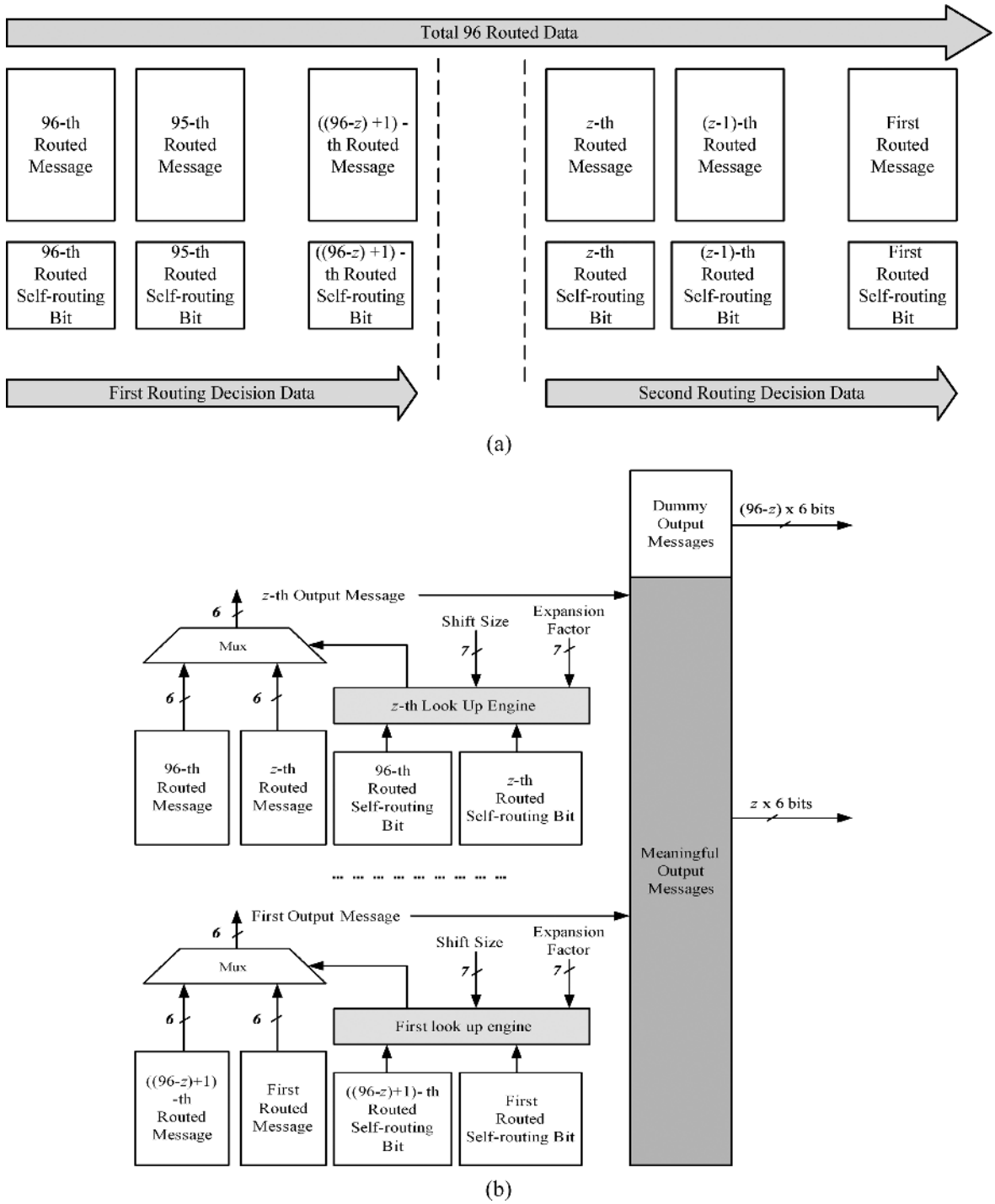

Fig. 9. Block diagram of the lookup engine: (a) the first routing decision data and the second routing decision data; (b) selection of the output messages from routing decision data using 96 look up engines.

Ninety-six out messages will be selected from the first routing decision data and the second routing decision data. The 96th routed message will be selected as the zth output message when the 96th routed self-routing bit is available (self-routing bit $= 1$ implies the available condition) and the zth routed self-routing bit is unavailable. The operation of the lookup engine will determine the expected messages based on the shift size and the expansion factor when both of the 96th routed self-routing bit and the zth routed self-routing bit are available.

V. Chip Measurement

Fig. 10 presents the fixed-point simulation results with different decoding iterations for the rate-1/2 and 2304-bit code. Note that the iteration number can be set according to the channel condition, and the chip throughput will be varied by means of controlling the iteration number. The maximum iteration number is set to 20 because of the trade-off between throughput and BER performance.

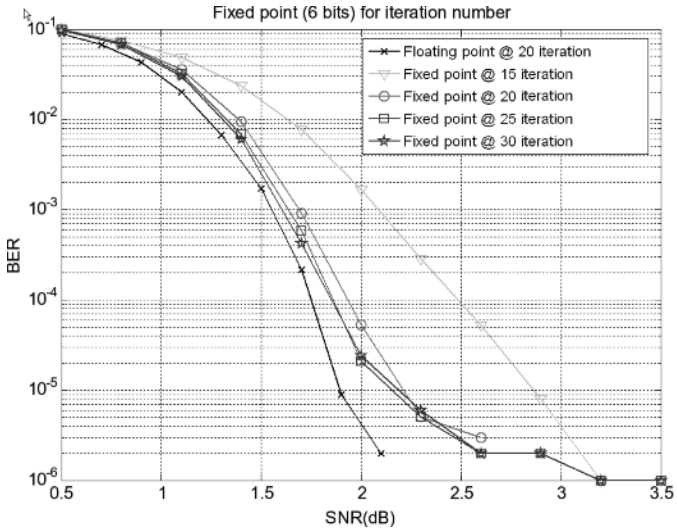

Fig. 10. Performance of fixed point simulation at rate-1/2 2304-bits code word.

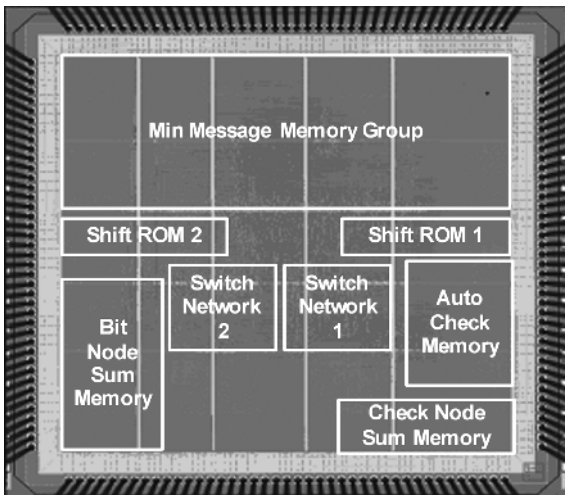

Fig. 11. Die photo of the LDPC decoder chip.

As shown in the micrograph (see Fig. 11), the decoder chip was implemented in a 90 nm 1P9M CMOS process, and its operation is programmable according to four parameters: code rate, expansion factor, sub-matrix shift size, and iteration number. Fig. 12 is the shmoo plot that indicates the maximum measured operation frequency at 1.0 V is 150 MHz. Under such operating frequency, we illustrate the chip throughput, ranging from 1.23 Gb/s to 0.105 Gb/s, for the 2304 bits code length in Fig. 13.

In IEEE 802.16e [21], the chip operating at frequency 109 MHz achieves the maximum 63.36 Mb/s data rate within 20 iterations and dissipates 186 mW at 1.0 V supply. The decoder chip occupies 6.25 mm^2 area. 380 k logic gates and 89 k bits memory with a 14 k bits dual-port SRAM for auto-check module are integrated together in this specific area. Note that the built-in auto-check module will compare the decoding result with the expected codewords stored in the memory. The chip parameters are listed in Table I, and the comparison with other decoders is shown in Table II, the energy efficiency is

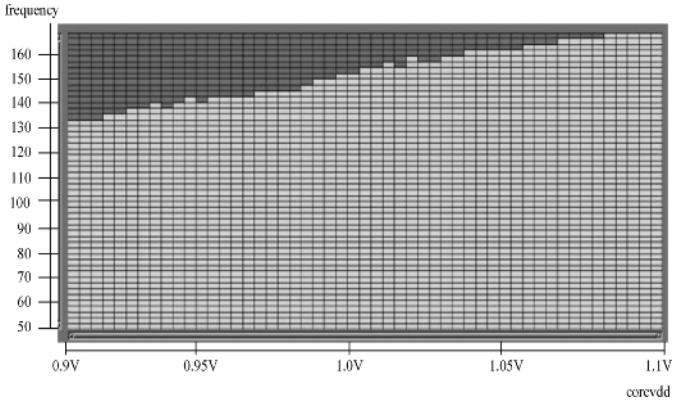

Fig. 12. Shmoo plot of chip testing.

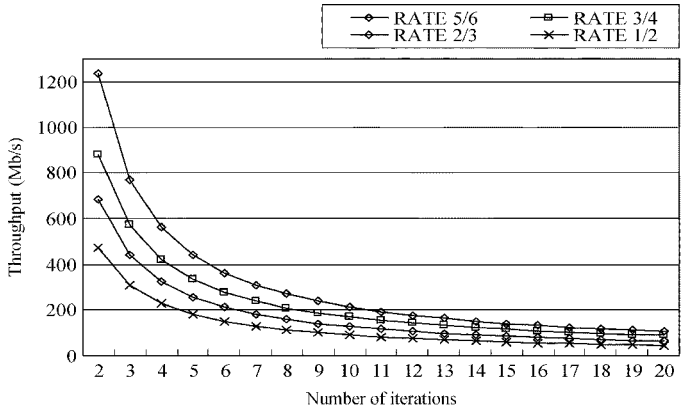

Fig. 13. Decoding throughput for different code rates from two to 20 iterations at operation frequency 150 MHz.

TABLE I
FEATURES OF THE LDPC DECODER IN IEEE 802.16e

System	IEEE 802.16e
Code Length	576 ~ 2304 Bits (Code Length = 24 x Expansion Factor)
Code Rate	1/2, 2/3A, 2/3B, 3/4A, 3/4B, 5/6
Expansion Factor	24 ~ 96 Total 19 Factors
Parity Check Matrix	Irregular
Modulation	QPSK/16QAM/64QAM
Total Memory	89,856 bits (14k bits for Auto-check Function)
IO Pins	208 Pin CFQP
Technology	Standard 90 nm 1P9M CMOS 1.0V Core 2.5V IO

derived as follows:

$$\text{Energy Efficiency} = \frac{\text{Power consumpation}}{\text{Throughput} \times \text{Iteration numbers}}. \quad (7)$$

TABLE II
OVERALL COMPARISON BETWEEN THE PROPOSED IEEE 802.16e LDPC DECODER AND THE EXISTING LDPC DECODERS

	Proposed IEEE 802.16e LDPC Decoder	Synthesis IEEE 802.16e LDPC Decoder IP [5]	TDMP LDPC Decoder [7]	Parallel LDPC Decoder [9]	Se-Hyeon Kang [12]
Maximum Throughput /Operating Frequency, Iterations	105 Mb/s (150MHz, 20 iterations)	619 Mb/s (333MHz, 10 iterations, Synthesis Result)	640 Mb/s (125MHz, 10 iterations)	1.0 Gb/s (64MHz, 64 iterations)	985 Mb/s (200MHz, 8 iterations)
Area	6.25 mm^2 (With Auto-check Memory)	3.834 mm^2 (Synthesis Result)	14.3 mm^2	52.5 mm^2	10.08 mm^2
Gate Counts	380k (Logic Part) 590k (Memory Part)	——	220k (Logic Part)	1750k	543k
Message Switch Network/Area	Self-routing Network (0.058 mm^2 at 400MHz)	Flexible Barrel Shifter (0.511 mm^2 at 333 MHz)	Bi-direction Bi-network (3.28 mm^2)	Fully Parallel	Matrix Permutation
Power	264 mW at 150MHz	——	787 mW at 125MHz	690 mW at 64MHz	——
Energy Efficiency	125 (pJ/Bit/Iter.)	——	123 (pJ/Bit/Iter.)	10.9 (pJ/Bit/Iter.)	——
Block Length	576~2304	576~2304	2048	1024	1024
Code Rate	1/2, 2/3A, 2/3B, 3/4A, 3/4B, 5/6	1/2, 2/3A, 2/3B, 3/4A, 3/4B, 5/6	8/16 : 1/16 : 14/16	1/2	1/2
Parity Check Matrix	Quasi-Cyclic	Quasi-Cyclic	(3,6)-regular	Irregular	(3,6)-regular
Sub-matrix Size	24~96	24~96	64	——	——
Operating Mode	114	114	7	single	single
CMOS Technology	90nm, 1.0V	130nm, 1.2V	180nm, 1.8V	160nm, 1.5V	180nm, 1.8V

VI. CONCLUSION

With the self-routing switch network, a 6.25 mm^2 LDPC decoder chip fully compliant to IEEE 802.16e applications is presented. This chip dissipates 264 mW power when decoding a rate-5/6 2304-bit LDPC code at 150 MHz and 1.0 V supply voltage; the throughput can achieve 105 Mb/s in 20 iterations. Additionally, the self-routing switch network enables to support the permutation function that can fulfill the requirement of different sub-matrix sizes. Signal routing congestion in the variable size switch network can be reduced significantly with only one permutation network that provides 19 different switch network sizes. Moreover, the phase-overlapping message passing algorithm is implemented to achieve the high throughput as specified in IEEE 802.16e with low hardware cost.

ACKNOWLEDGMENT

The authors thank Dr. Chien-Ching Lin and Yen-Chin Liao for layout assistance and comments for paper writing, and the National Chip Implementation Center for chip measurement assistance.

REFERENCES

[1] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.
[2] C.-C. Lin, K.-L. Lin, C.-C. Chung, and C.-Y. Lee, "A 3.33 Gb/s (1200, 720) low-density parity check code decoder," in *Proc. ESSCIRC*, 2005, pp. 211–214.
[3] P. Urard, E. Yeo, L. Paumier, P. Georgelin, T. Michel, V. Lebars, E. Lantreibecq, and B. Gupta, "A 135 Mb/s DVB-S2 compliant codec based on 64800b LDPC and BCH codes," in *IEEE ISSCC Dig. Tech. Papers*, 2005, pp. 446–447.
[4] X.-Y. Shi, "VLSI designs of LDPC codec for IEEE 802.16e system," Masters thesis, National Taiwan Univ., Taipei, Taiwan, R.O.C., 2006.
[5] T. Brack, M. Alles, F. Kienle, and N. When, "A synthesizable IP core for WIMAX 802.16E LDPC code decoding," in *Proc. IEEE 17th Int. Symp. Personal, Indoor and Mobile Radio Communications*, Sep. 2006, pp. 1–5.
[6] M. M. Mansour and N. R. Shanbhag, "High-throughput LDPC decoders," *IEEE Trans. Very Large Scale Integr.(VLSI) Syst.*, vol. 11, no. 6, pp. 976–996, Dec. 2003.
[7] M. M. Mansour and N. R. Shanbhag, "Design methodology for high-throughput memory-efficient programmable decoder cores for architecture-aware low-density parity-check codes," in *Proc. IEEE Workshop on Signal Process. Syst (SiPS'03)*, Seoul, Korea, Aug. 2003, pp. 159–164.
[8] M. M. Mansour and N. R. Shanbhag, "A 640-Mb/s 2048-bit programmable LDPC decoder chip," *IEEE J. Solid-State Circuits*, vol. 41, no. 3, pp. 634–698, Mar. 2006.
[9] *Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands*, IEEE P802.16e-2005, 2005.
[10] A. J. Blanksby and C. J. Howland, "A 690 mW 1 Gb/s 1024b rate 1/2 low density parity check code decoder," *IEEE J. Solid-State Circuits*, vol. 37, no. 3, pp. 404–412, Mar. 2002.
[11] T. Zhang, Z. Wang, and K. K. Parhi, "On finite precision implementation of low density parity check codes decoder," in *Proc. IEEE ISCAS*, Sydney, Australia, May 2001, vol. 4, pp. 202–205.
[12] S. Kim, G. E. Sobelman, and J. Moon, "Parallel VLSI architectures for a class of LDPC codes," in *Proc. IEEE ISCAS*, Phoenix-Scottsdale, AZ, May 2002, vol. 2, pp. 93–96.
[13] H. Chen, "A FPGA and ASIC implementation of rate 1/2, 8088-b irregular low density parity check decoder," in *Proc. IEEE GLOBECOM*, 2003, vol. 1, pp. 113–117.
[14] S.-H. Kang and I.-C. Park, "Loosely coupled memory-based decoding architecture for low density parity check codes," *IEEE Trans. Circuits Syst. I*, vol. 53, no. 5, pp. 1045–1056, May 2006.
[15] C.-H. Liu, C.-C. Lin, H.-C. Chang, C.-Y. Lee, and Y.-S. Hsu, "Method and apparatus for switching data in communication systems," Taiwan and US patent pending.

[16] J. L. Fan, *Constrained Coding and Soft Iterative Decoding*. Boston: Kluwer Academic, 2001.
[17] J. Chen and M. Fossorier, "Near optimum universal belief propagation based decoding of lower-density parity check codes," *IEEE Trans. Commun.*, vol. 50, pp. 406–414, Mar. 2002.
[18] D. E. Hocevar, "A reduced complexity decoder architecture via layered decoding of LDPC codes," in *Proc. IEEE Workshop on Signal Processing Systems*, Austin, TX, Oct. 2004, pp. 107–112.
[19] F. Quaglio, F. Vacca, C. Castellano, A. Tarable, and G. Masera, "Interconnection framework for high-throughput, flexible LDPC decoders," in *Proc. Design Automation and Test in Europe*, Mar. 2006, vol. 2, pp. 6–10.
[20] J. Tang, T. Bhatt, V. Sundaramurthy, and K. K. Parhi, "Reconfigurable shuffle network design in LDPC decoders," in *Proc. Application-Specific Systems, Architecture and Processors, 2006 (ASAP'06)*, Steamboat Springs, CO, Sep. 2006, pp. 81–86.
[21] "Mobile WiMAX—Part I: A technical overview and performance evaluation," WiMAX Forum, Aug. 2006.

Chih-Hao Liu received the B.S. and Masters degrees from the Department of Power Mechanical Engineering, National Tsing-Hua University, Hsinchu, Taiwan, R.O.C., in 1998 and 2000, respectively. From January 2001 to August 2006, he was with Industrial Technology Research Institute, as an engineer for switch network and WiMax integration circuit design. He is currently working toward the Ph.D degree in the Department of Electrical Engineering, National Tsing-Hua University, Hsinchu, Taiwan. His research interests include switch network architecture design, communication integration circuit design, coding theory and digital communication.

Shao-Wei Yen received the B.S. and Masters degrees from the Department of Electronics Engineering, National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 2004 and 2006, respectively. He is currently working toward the Ph.D degree in the Institute of Electronics Engineering, National Chiao Tung Univerisity. His research interests include digital communication, coding theory, and VLSI signal processing.

Chih-Lung Chen received the B.E. and M.S. degrees from the Department of Electronics Engineering, National Chiao-Tung University, Hsinchu, Taiwan, R.O.C., in 2004 and 2006, respectively. He is currently pursuing the Ph.D. degree in electronics engineering at National Chiao-Tung University. His general research interests include VLSI implementation of error control codes and wireless communication systems.

Hsie-Chia Chang was born in Keelung, Taiwan. He received the B.S., M.S., and Ph.D. degrees in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1995, 1997, and 2002, respectively.

From 2002 to 2003, he was with OSP/DE1 in MediaTek Inc., working in the area of decoding architectures for Combo SoC. In February 2003, he joined the faculty of the Department of Electronics Engineering, National Chiao Tung University, as an Assistant Professor. His current research interests include algorithms and circuit architectures in signal processing, especially for channel coding and crypto-systems, and joint source/channel coding for cross-layer communications.

Chen-Yi Lee (M'01) received the B.S. degree from National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1982, and the M.S. and Ph.D. degrees from Katholieke University Leuven (KUL), Belgium, in 1986 and 1990, respectively, all in electrical engineering.

From 1986 to 1990, he was with IMEC/VSDM, working in the area of architecture synthesis for DSP. In February 1991, he joined the faculty of the Electronics Engineering Department, National Chiao Tung University, Hsinchu, Taiwan, where he is currently a Professor and Dean of Research and Development Office. His research interests mainly include VLSI algorithms and architectures for high-throughput DSP applications. He is also active in various aspects of high-speed networking, system-on-chip design technology, very low power designs, and multimedia signal processing. In these areas, he has published more than 150 papers and holds decades of patents.

Dr. Lee served as the Director of Chip Implementation Center (CIC), an organization for IC design promotion in Taiwan (2000/8 $\sim$ 2003/12), and the microelectronics program coordinator of Engineering Division under National Science Council of Taiwan (2003/1 $\sim$ 2005/12). He was the former IEEE CAS Taipei Chapter Chair and is currently a member of IEEE.

Yar-Sun Hsu received the B.S. and M.S. degrees in electronics engineering from National Chiao Tung University, Taiwan, R.O.C., and the Ph.D. degree from Rensselaer Polytechnic Institute, Troy, NY.

He joined IBM T. J. Watson Research Center, Yorktown Heights, NY, in 1982 after working for General Electric Company in New York for three years. Since then, he has been involved in the research of scalable parallel cluster system, multiprocessor system, switching interconnection network, VLSI technology, and CMOS chip design. In 1988 he became the manager of a system department working on the design of IBM Scalable Power Parallel System, cache coherence protocol for multiprocessor systems, performance evaluation and visualization, and scalable parallel I/O. In 2002, he joined the Department of Electrical Engineering, National Tsing Hua University, Taiwan, as a Professor.

Dr. Hsu received one IBM Outstanding Technical Achievement Award, three IBM invention plateau awards, two IBM supplemental invention awards, and three IBM Research Division technical achievement awards. He has also received the best system paper award from the ACM SIGMETRICS Conference in 2000, the best paper award from International Computer Symposium in 2004, and the outstanding teaching award from National Tsing Hua University in 2006. His current interests include MPSoC architecture, on-chip interconnection network, cluster system, parallel I/O, and SoC design.

Shyh-Jye Jou was born in Taiwan, R.O.C., in 1960. He received the B. S. degree in electrical engineering from National Chen Kung University in 1982, and the M.S. and Ph.D. degrees in electronics from National Chiao Tung University in 1984 and 1988, respectively.

He was with the Electrical Engineering Department of National Central University, Chung-Li, Taiwan, from 1990 to 2004 and became a Professor in 1997. Since 2004, he has been a Professor of the Electronics Engineering Department of National Chiao Tung University, and became the Chairman from 2006. He was a Visiting Research Associate Professor in the Coordinated Science Laboratory at University of Illinois at Urbana-Champaign during the 1993–1994 academic years. In the summer of 2001, he was a Visiting Research Consultant in the Communication Circuits and Systems Research Laboratory of Agere Systems, USA. His research interests include design and analysis of high-speed, low-power digital integrated circuits, and communication integrated circuits and systems.

Dr. Jou has served on the technical program committees in several international conferences including Custom Integrated Circuits Conference (CICC1994–1996) and Asian Solid-State Circuits Conference (A-SSCC) 2005 to 2007.