A 5.79-Gb/s Energy-Efficient Multirate LDPC Codec Chip for IEEE 802.15.3c Applications

Shao-Wei Yen, Shiang-Yu Hung, Chih-Lung Chen, Hsie-Chia Chang, Shyh-Jye Jou, and Chen-Yi Lee, *Member, IEEE*

***Abstract*—An LDPC codec chip supporting four code rates of IEEE 802.15.3c applications is presented. After utilizing row-based layered scheduling, the normalized min-sum (NMS) algorithm can reduce half of the iteration number while maintaining similar performance. According to the unique code structure of the parity-check matrix, a reconfigurable 8/16/32-input sorter is designed to deal with LDPC codes in four different code rates. Both sorter input reallocation and pre-coded routing switch are proposed to alleviate routing complexity, leading to 64% input reduction of multiplexers. In addition, an adder-accumulator-shift register (AASR) circuit is proposed for the LDPC encoder to reduce hardware complexity. After implemented in 65-nm 1P10M CMOS process, the proposed LDPC decoder chip can achieve maximum 5.79-Gb/s throughput with the hardware efficiency of 3.7 $\mathrm{Gb/s/mm^2}$ and energy efficiency of 62.4 pJ/b, respectively.**

***Index Terms*—IEEE 802.15.3c, low-density parity-check (LDPC) codes, row-based layered scheduling.**

I. Introduction

LOW-DENSITY parity-check (LDPC) code, which is a popular linear block code with a simple decoding algorithm and good error-correcting capability, was first introduced by Gallager [1] and has attracted many research interests after rediscovery by Mackay [2]. An LDPC decoder is based on the iterative belief-propagation (BP) algorithm and is capable of parallel implementation, leading to a much higher decoding speed than other channel decoders. Therefore, newly high-speed wireless communication systems such as IEEE 802.15.3c [3] and IEEE 802.11ad [4] have also adopted LDPC codes to achieve better error correcting performance and Gb/s data rate with multiple code rates and block sizes.

However, the circuit implementation of LDPC decoder is still a great challenge when higher parallel implementation is adopted. In the first silicon implementation of a fully parallel decoder [5], it was reported that the size of the decoder was determined by routing congestion rather than the gate count. In the later works, Lin [6] and Chen [7] showed that the decoders could achieve 3.3 and 11.5 Gb/s in the 0.18-μm and 90-nm CMOS processes, respectively. However, these works only focus on the single LDPC parity-check matrix. For the designs of various block sizes and code rates, it is difficult to meet the high-throughput (over Gb/s) target while maintaining low hardware complexity. For those works [8], [9] on WiMAX applications, both of the two works could only achieve Mb/s throughput due to multiple block sizes and code rates.

TABLE I
LDPC Codes Parameters of IEEE 802.15.3c

Code	Code rate	Codeword n(bits)	Information k(bits)
LDPC(672,336)	0.5	672	336
LDPC(672,420)	0.625	672	420
LDPC(672,504)	0.75	672	504
LDPC(672,588)	0.875	672	588

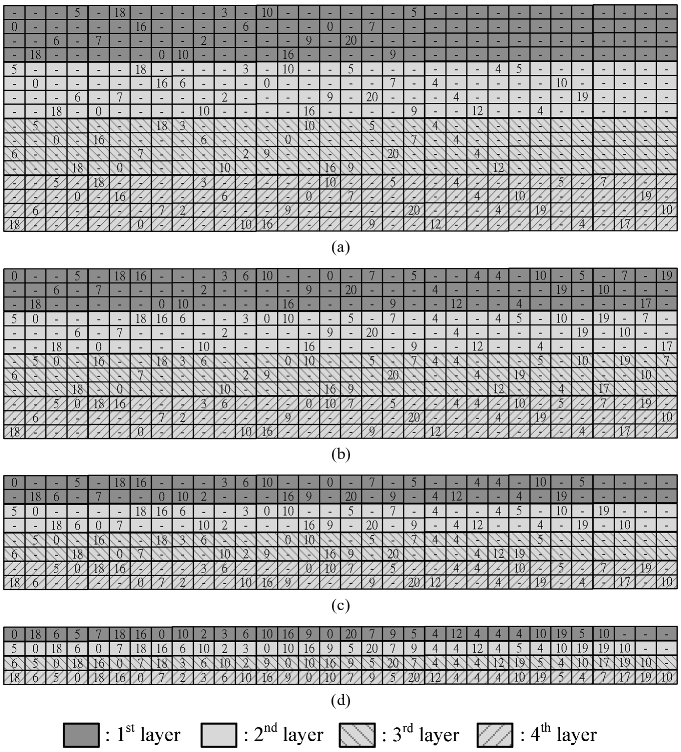
The IEEE 802.15.3c wireless personal area network (WPAN) standard defines the wireless communication protocol for indoor high-data-rate applications, and the data rate can achieve over 5 Gb/s. Four irregular LDPC codes are provided in the standard. The supported code rates, information block lengths, and codeword block lengths are listed in Table I. The parity-check matrix H can be decomposed into several $z \times z$ submatrices, and each submatrix is either the zero matrix or the cyclic left-permutation identity matrix. Fig. 1 represents four LDPC codes, where the numbers indicate the permutation amounts and "-" indicates the zero matrix. By observing the four LDPC codes, there are some structural features which can be used to simplify hardware. Each parity-check matrix can be horizontally and equally divided into four layers. To clearly describe the structure property, take the first layer of four LDPC codes, as the examples illustrated as the deep gray part in Fig. 1. Assume that H_i represents the ith row in the first layer. In Fig. 1(a) and (b), the merged row of H_1 and H_2 in (672, 336) LDPC code is equivalent to H_1 in the (672, 420) LDPC code. The equivalent case means that the permutation amounts of two rows at the same column indices are the same while ignoring the zero matrix. Moreover, H_3 and H_4 in the (672, 336) LDPC code are also equivalent to H_3 and H_4 in the (672, 420) LDPC code, respectively. In Fig. 1(b) and (c), the merged row of H_2 and H_3 in the (672, 420) LDPC code is equivalent to H_2 in the (672, 504) LDPC code and H_1 in the (672, 420) LDPC code is equivalent to H_1 in the (672, 504) LDPC code. In Fig. 1(c) and (d), the merged row of H_1 and H_2 in the (672, 504) LDPC code is equivalent to H_1 in the (672, 588) LDPC code. Therefore, the

Manuscript received April 22, 2011; revised February 06, 2012; accepted March 15, 2012. Date of publication May 10, 2012; date of current version August 21, 2012. This paper was approved by Associate Editor Bevan Baas. This work was supported by the National Science Council of Taiwan under Grant NSC 100-2220-E-009-029 and Grant NSC 100-2220-E-009-024.

The authors are with the Department of Electronics Engineering and the Institute of Electronics, National Chiao-Tung University, Hsinchu, Taiwan, R.O.C. (e-mail: shouway.ee89@nctu.edu.tw; hcchang@si2lab.org; jerryjou@mail.nctu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSSC.2012.2194176



-	-	-	5	-	18	-	-	-	-	3	-	10	-	-	-	-	-	-	5	-	-	-	-	-	-	-	-	-	-	-	-
0	-	-	-	-	-	16	-	-	-	-	6	-	-	-	0	-	7	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	6	-	7	-	-	-	-	2	-	-	-	-	9	-	20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	18	-	-	-	-	-	0	10	-	-	-	-	16	-	-	-	-	9	-	-	-	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	18	-	-	-	-	3	-	10	-	-	5	-	-	-	-	-	-	4	5	-	-	-	-	-	-	-
-	0	-	-	-	-	-	16	6	-	-	-	0	-	-	-	-	-	7	-	4	-	-	-	-	-	10	-	-	-	-	-
-	-	-	6	-	7	-	-	-	-	2	-	-	-	-	9	-	20	-	-	-	4	-	-	-	-	-	19	-	-	-	-
-	-	18	-	0	-	-	-	-	10	-	-	-	-	16	-	-	-	-	9	-	-	12	-	-	4	-	-	-	-	-	-
-	5	-	-	-	-	-	18	3	-	-	-	-	-	10	-	-	5	-	-	4	-	-	-	-	-	-	-	-	-	-	-
-	-	0	-	16	-	-	-	-	6	-	-	-	0	-	-	-	-	-	7	-	4	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	7	-	-	-	-	2	9	-	-	-	-	-	20	-	-	-	4	-	-	-	-	-	-	-	-	-
-	-	-	18	-	0	-	-	-	-	10	-	-	-	-	16	9	-	-	-	-	-	-	12	-	-	-	-	-	-	-	-
-	-	5	-	18	-	-	-	-	3	-	-	-	-	-	10	-	-	5	-	-	4	-	-	-	-	5	-	7	-	-	-
-	-	-	0	-	16	-	-	-	-	6	-	-	-	0	-	7	-	-	-	-	-	4	-	10	-	-	-	-	-	19	-
-	6	-	-	-	-	-	7	2	-	-	-	-	9	-	-	-	-	-	20	-	-	-	4	-	19	-	-	-	-	-	10
18	-	-	-	-	-	0	-	-	-	-	10	16	-	-	-	-	9	-	-	12	-	-	-	-	-	-	4	-	17	-	-

(a)

0	-	-	5	-	18	16	-	-	-	3	6	10	-	-	0	-	7	-	5	-	-	4	4	-	10	-	5	-	7	-	19
-	-	6	-	7	-	-	-	-	2	-	-	-	-	9	-	20	-	-	-	4	-	-	-	-	-	19	-	10	-	-	-
-	18	-	-	-	-	-	0	10	-	-	-	-	16	-	-	-	-	9	-	-	12	-	-	4	-	-	-	-	-	17	-
5	0	-	-	-	-	18	16	6	-	-	3	0	10	-	-	5	-	7	-	4	-	-	4	5	-	10	-	19	-	7	-
-	-	-	6	-	7	-	-	-	-	2	-	-	-	-	9	-	20	-	-	-	4	-	-	-	-	-	19	-	10	-	-
-	-	18	-	0	-	-	-	-	10	-	-	-	-	16	-	-	-	-	9	-	-	12	-	-	4	-	-	-	-	-	17
-	5	0	-	16	-	-	18	3	6	-	-	-	0	10	-	-	5	-	7	4	4	-	-	-	5	-	10	-	19	-	7
6	-	-	-	-	-	7	-	-	-	-	2	9	-	-	-	-	-	20	-	-	-	4	-	19	-	-	-	-	-	10	-
-	-	-	18	-	0	-	-	-	-	10	-	-	-	-	16	9	-	-	-	-	-	-	12	-	-	4	-	17	-	-	-
-	-	5	0	18	16	-	-	-	3	6	-	-	-	0	10	7	-	5	-	-	4	4	-	10	-	5	-	7	-	19	-
-	6	-	-	-	-	-	7	2	-	-	-	-	9	-	-	-	-	-	20	-	-	-	4	-	19	-	-	-	-	-	10
18	-	-	-	-	-	0	-	-	-	-	10	16	-	-	-	-	9	-	-	12	-	-	-	-	-	-	4	-	17	-	-

(b)

0	-	-	5	-	18	16	-	-	-	3	6	10	-	-	0	-	7	-	5	-	-	4	4	-	10	-	5	-	-	-	-
-	18	6	-	7	-	-	0	10	2	-	-	-	16	9	-	20	-	9	-	4	12	-	-	4	-	19	-	-	-	-	-
5	0	-	-	-	-	18	16	6	-	-	3	0	10	-	-	5	-	7	-	4	-	-	4	5	-	10	-	19	-	-	-
-	-	18	6	0	7	-	-	-	10	2	-	-	-	16	9	-	20	-	9	-	4	12	-	-	4	-	19	-	10	-	-
-	5	0	-	16	-	-	18	3	6	-	-	-	0	10	-	-	5	-	7	4	4	-	-	-	5	-	-	-	-	-	-
6	-	-	18	-	0	7	-	-	-	10	2	9	-	-	16	9	-	20	-	-	-	4	12	19	-	-	-	-	-	-	-
-	-	5	0	18	16	-	-	-	3	6	-	-	-	0	10	7	-	5	-	-	4	4	-	10	-	5	-	7	-	19	-
18	6	-	-	-	-	0	7	2	-	-	10	16	9	-	-	-	9	-	20	12	-	-	4	-	19	-	4	-	17	-	10

(c)

0	18	6	5	7	18	16	0	10	2	3	6	10	16	9	0	20	7	9	5	4	12	4	4	4	10	19	5	10	-	-	-
5	0	18	6	0	7	18	16	6	10	2	3	0	10	16	9	5	20	7	9	4	4	12	4	5	4	10	19	19	10	-	-
6	5	0	18	16	0	7	18	3	6	10	2	9	0	10	16	9	5	20	7	4	4	4	12	19	5	4	10	17	19	10	-
18	6	5	0	18	16	0	7	2	3	6	10	16	9	0	10	7	9	5	20	12	4	4	4	10	19	5	4	7	17	19	10

(d)

▇ : 1st layer □ : 2nd layer ▧ : 3rd layer ▨ : 4th layer

Fig. 1. Matrix permutation indexes of structured parity-check matrices H. (a) (672,336) LDPC code. (b) (672,420) LDPC code. (c) (672,504) LDPC code. (d) (672,588) LDPC code.

first layers of the four LDPC codes are equivalent in combination of rows and the other three layers have the same properties. Furthermore, the column degree of each layer is at most one. Based on these structural properties, the permutation blocks and computation units can be simplified and reused for four LDPC codes.

There are several challenges in architecture design for the 802.15.3c standard, such as high-throughput requirement, complicated signal routing of irregular parity-check matrices, and hardware overhead to support four code rates. According to these design challenges, we propose a low-power area-efficient LDPC decoder for 802.15.3c applications. In order to meet the throughput requirement, row-based scheduling and normalized min-sum (NMS) decoding algorithm [10], [11] are employed to reduce half of the maximum iterations. Based on the structure of parity-check matrices, an input reallocation technique for sorters and pre-coded routing switches are proposed to support all code rates in shared hardware and to reduce routing complexity of message passing between check nodes and variable nodes. For the purpose of low-power design, an early termination method is applied to avoid redundant decoding time and power when the codeword satisfies the check equation. For the encoder scheme, adder-accumulator-shift register is proposed to reduce the storage and computation units. Finally, a complete LDPC codec architecture, including encoder, AWGN channel, and decoder, is implemented for chip testing.

This paper is organized as follows. The NMS decoding algorithm with row-based layered scheduling and the encoding scheme are introduced in Section II. The proposed hardware architecture and the detailed functional blocks are demonstrated in Section III. Section IV shows the simulation results for hardware parameters configuration. Based on the proposed architecture, Section V shows the implementation for testing plan and the measurement results. Finally, a conclusion is given in Section VI.

II. Decoding Algorithm and Encoding Scheme

To achieve high-throughput and low-power requirement, the decoding algorithm should provide simple operation and better decoding efficiency. Therefore, the NMS algorithm with row-based layered scheduling is presented in the following. Besides, the encoding scheme is decomposed into vector accumulation and a modified accumulator is proposed to reduce the hardware cost.

A. NMS Algorithm With Row-Based Layered Scheduling

In order to speed up the decoding convergence, two scheduling methods including row-based layered and column-based shuffle scheduling are proposed [10], [12]. Both of these two scheduling methods allow the updated information to be utilized immediately, leading to similar decoding performance as the standard BP algorithm in fewer iterations. The difference between these two scheduling methods is the message updating procedure, where the row-based layered scheduling partitions the check nodes horizontally and the column-based shuffle scheduling partitions the variable nodes vertically.

Compared with the fully parallel architecture, the check node units (CNUs) can be shared between different groups while adopting row-based layered scheduling, and the required amount for variable node units (VNUs) remains the same as the block length. On the contrary, column-based shuffle scheduling is suitable for long block length by sharing the VNUs among different divided groups. Obviously, all of these features reveal that the row-based layered scheduling architecture is more suitable for the LDPC code of short codeword length and low row degree. All of the LDPC codes of IEEE 802.15.3c have short codeword length $(N = 672)$ and low row degrees less than 32. Moreover, the parity-check matrices of the four LDPC codes have the same layer structural property described in Section I, indicating that row-based layered scheduling is a good candidate for implementation of IEEE 802.15.3c LDPC codes. Since there is bit-error-rate (BER) performance loss due the approximation error by the min-sum algorithm, the NMS algorithm which uses a normalized factor β is utilized to improve performance in our approach.

In the row-based layered scheduling, the rows of parity-check matrix H are horizontally divided into G layers. The message updating between check nodes and variable nodes within a layer is done in one subiteration where one iteration indicates the process of completing all layers of H sequentially. The log-likelihood ratio (LLR) of intrinsic information of the nth variable node is denoted by P_n. The message from the mth check node to the nth variable node and the message from nth variable node to mth check node are defined as r_{mn} and q_{nm}, respectively. $N(m)$ denotes the set of variable nodes connecting to the mth check node and $N(m)\backslash n$ represents the set of $N(m)$ excluding the nth variable node. $M(n)$ denotes the set of check nodes connecting to the nth variable node and $M(n)\backslash m$ represents the set

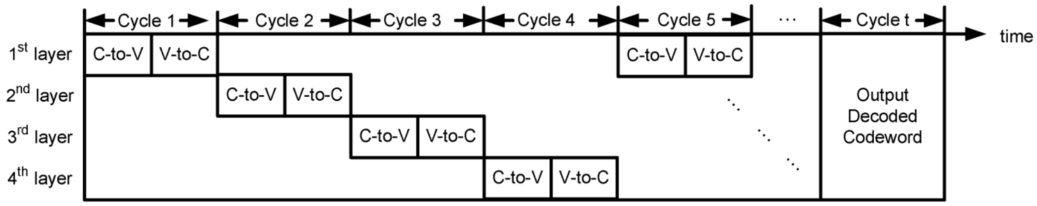


Fig. 2. Decoding flow of row-based layered scheduling.

of $M(n)$ excluding the mth check node. The *a posteriori* LLR of the nth bit is denoted by Z_n. The updating procedure of NMS algorithm with row-based layered scheduling is carried out as follows.

1) Initialization: For all variable nodes n and check nodes m, set $q_{nm} = P_n$ and $r_{mn} = 0$.
2) Iterative Decoding: For $0 \leq i \leq G \times I_{\max} - 1)$, perform the following four steps.
   (a) Check nodes to variable nodes updating step: For $g = i (\mathrm{mod}\ G)$ and each $m \in$ check nodes in the gth layer, we have

$$r_{mn} = \prod_{n' \in \{N(m)\backslash n\}} sign(q_{n'm}) \times \left(\min_{n' \in \{N(m)\backslash n\}} |q_{n'm}| \times \beta \right). \tag{1}$$

   (b) Variable nodes to check nodes and *a posteriori* LLR updating step:
   For all variable nodes n, we have

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

$$Z_n = P_n + \sum_{m' \in M(n)} r_{m'n}. \tag{3}$$

   (c) Hard decision: Let X_n be the nth bit of the decoded coded word. If $Z_n < 0$, $X_n = 1$, else $X_n = 0$.
   (d) Stopping criterion: The iterative decoding continues until the decoded codeword satisfies all the parity-check equations or maximum number of iteration $I_{\max}$ is reached.

Fig. 2 represents the decoding flow based on $G = 4$. Notice that C-to-V and V-to-C denote the check nodes to variable nodes updating step as (1) and variable nodes to check nodes updating step as (2) and (3), respectively. In conventional BP algorithm, V-to-C starts after completing C-to-V for all check nodes. Since the row-based layered scheduling divides the parity-check matrix into several layers horizontally, V-to-C follows C-to-V for check nodes in one layer and uses the newest updated message immediately. In each clock cycle, only check nodes of one layer are updated and r_{mn} in the layer are immediately utilized for all variable nodes. Since C-to-V is executed sequentially through different layers, the CNUs can be reused.

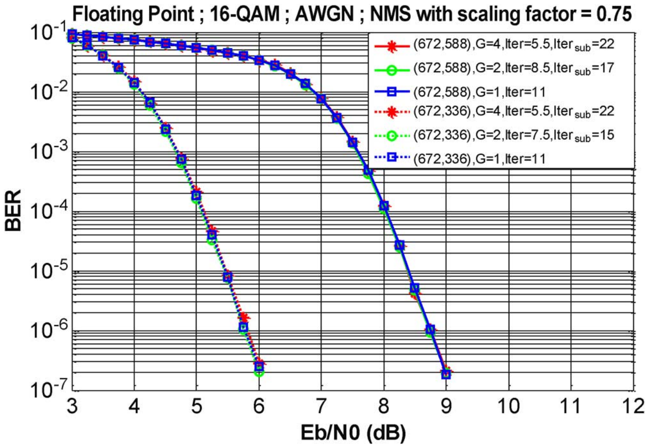


Fig. 3. BER performance under different number of layer groups.

TABLE II
COMPARISON OF DIFFERENT LAYER NUMBERS

Layer number	$G = 1$	$G = 2$	$G = 4$
Critical path delay (ns)	6.5	6.0	5.5
Throughput for (672,336)	4.69	3.73	3.05
Throughput for (672,588)	8.22	5.76	5.34
Gate count (k)	1612	957	647

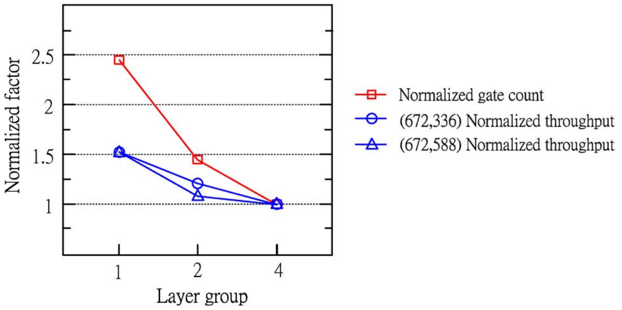


Fig. 4. Area and throughput under different number of layer groups.

The number of layer group would affect the hardware cost, throughput, and BER performance, which are a tradeoff of each other, so it is essential to explore the effect of different numbers of layer groups. Fig. 3 shows BER performance for (672, 336) and (672, 588) LDPC codes under different numbers of layer groups (G). Although the decoding iterations are different for the different G, similar BER performances can be achieved. Based on the synthesis results, Table II shows the critical path delay, throughput, and gate count of different layered scheduling for $G = 1, 2,$ and 4. The gate count can be greatly reduced due to the hardware sharing of CNUs. Although the critical path delay decreases slightly, the overall throughput decreases because it requires more cycles to complete one iteration. Fig. 4 shows that the four-layer groups $(G = 4)$ are efficient in consideration of hardware efficiency, and both the throughput and gate count are normalized according to the smallest one. In addition, the throughput requirement of IEEE 802.15.3c, 5.39 Gb/s for

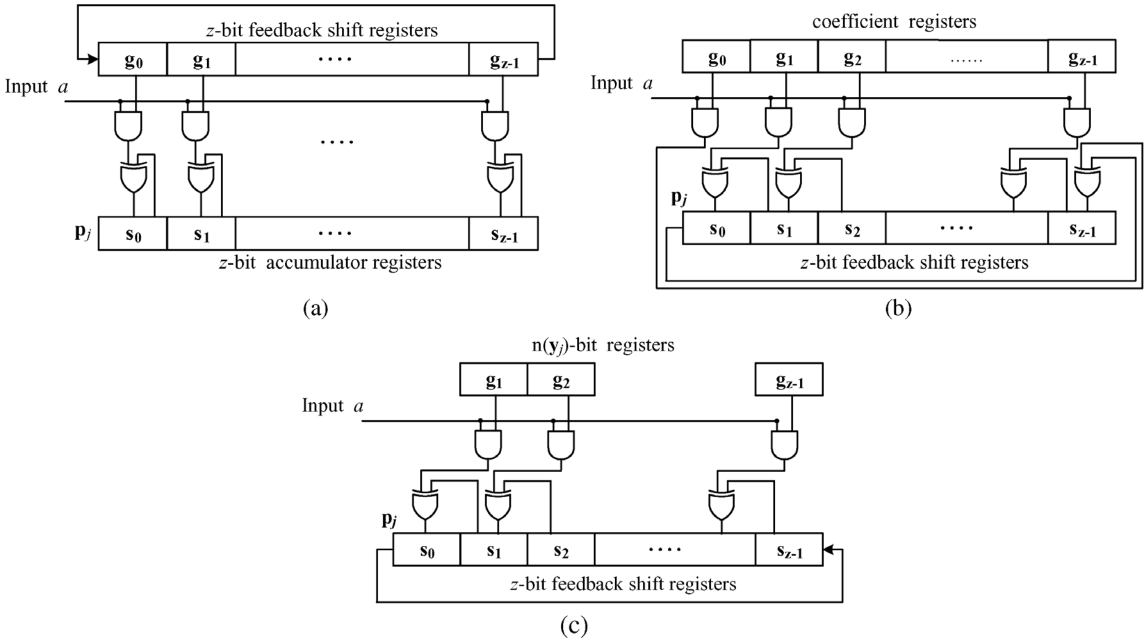


Fig. 5. Encoder circuit. (a) SRAA structure. (b) Basic AASR structure. (c) AASR structure with hardware reduction.

$(672, 588)$ LDPC code and 3.08 Gb/s for $(672, 336)$ LDPC code, can be still achieved with $G = 4$ architecture.

B. Encoding Scheme

Since the LDPC codes provided by 802.15.3c belong to the category of QC-LDPC codes [13], these codes have encoding advantage over other LDPC codes. The encoding of QC-LDPC codes can be accomplished by using simple shift registers instead of large storage memory for the generator matrix $\mathbf{G}$.

In the 802.15.3c standard, only parity-check matrix $\mathbf{H}$ is defined. The generator matrix G can be derived from the property of $\mathbf{HG^T} = \mathbf{0}$. Based on the systematic encoding in (n, k) QC-LDPC codes, the generator matrix has the following form:

$$\mathbf{G}=\begin{bmatrix}\mathbf{G}_0\\ \mathbf{G}_1\\ \vdots\\ \mathbf{G}_{kb-1}\end{bmatrix} = \left[\begin{array}{cccc|ccc}\mathbf{I} & \mathbf{O} & \cdots & \mathbf{O} & \mathbf{G}_{0,0} & \cdots & \mathbf{G}_{0,mb-1}\\ \mathbf{O} & \mathbf{I} & \cdots & \mathbf{O} & \mathbf{G}_{1,0} & \cdots & \mathbf{G}_{1,mb-1}\\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\ \mathbf{O} & \mathbf{O} & \cdots & \mathbf{I} & \mathbf{G}_{kb-1,0} & \cdots & \mathbf{G}_{kb-1,mb-1}\end{array}\right] \quad (4)$$

where $\mathbf{I}$ is a $z \times z$ identity matrix, and $\mathbf{O}$ is a $z \times z$ zero matrix. $\mathbf{G}_{i,j}$ is a $z \times z$ circulant matrix for $0 \le i \le kb - 1 (kb = k/z)$ and $0 \le j \le mb - 1 (mb = (n-k)/z)$. Assume an information sequence $\mathbf{a} = (\mathbf{a}_0, \mathbf{a}_1, \cdots, \mathbf{a}_{kb-1})$ is transmitted, where $\mathbf{a}_i = (a_{iz}, a_{iz+1}, \cdots, a_{iz+z-1})$ consists of z consecutive information bits. Then, the codeword $\mathbf{v} = (\mathbf{a}, \mathbf{p}_0, \mathbf{p}_1, \cdots, \mathbf{p}_j, \cdots, \mathbf{p}_{mb-1})$ can be obtained by $\mathbf{v} = \mathbf{aG}$. Due to the systematic generator matrix $\mathbf{G}$, the associated codeword $\mathbf{v}$ is encoded in the systematic form, and $\mathbf{p}_j$ are z parity bits. The generation of $\mathbf{p}_j$ can be obtained by

$$\mathbf{p}_j = \mathbf{a}_0\mathbf{G}_{0,j} + \mathbf{a}_1\mathbf{G}_{1,j} + \cdots\cdots + \mathbf{a}_{kb-1}\mathbf{G}_{kb-1,j}. \quad (5)$$

Each term of the right-hand side in (5) could be further expressed as

$$\mathbf{a}_i\mathbf{G}_{i,j} = a_{iz}\mathbf{g}_{i,j}^{(0)} + a_{iz+1}\mathbf{g}_{i,j}^{(1)} + \cdots\cdots + a_{iz+z-1}\mathbf{g}_{i,j}^{(z-1)} \quad (6)$$

where $\mathbf{g}_{i,j}^{(l)}$, the lth row of submatrix $\mathbf{G}_{i,j}$, is l-bits (right) cyclic shift of $\mathbf{g}_{i,j}^{(0)}$. According to (5) and (6), the encoder can be constructed based on shift-register adder-accumulator (SRAA) circuit shown in Fig. 5(a) [14]. The circuit uses AND gates and XOR gates to perform addition in bit level, and shift registers are used to process cyclic shift operation of $\mathbf{g}_{i,j}^{(0)}$. At the beginning, $\mathbf{g}_{0,j}^{(0)}$ is stored in the coefficient shift registers and the accumulator registers are set to zero. The information sequence $\mathbf{a}$ is shifted into encoder circuit bit by bit and the circuit performs the shift operation and accumulation. For example, information a_0 is fed into the encoder, and $a_0\mathbf{g}_{0,j}^{(0)}$ is obtained at the outputs of AND gates and stored in the accumulator registers. Then the next information a_1 is fed into encoder, and the feedback shift registers shift one bit to right then the new content of shift registers becomes $\mathbf{g}_{0,j}^{(1)}$. Therefore, $a_0\mathbf{g}_{0,j}^{(0)} + a_1\mathbf{g}_{0,j}^{(1)}$ is obtained by adding the original content of the accumulator registers. Every z cycles, $\mathbf{a}_i\mathbf{G}_{i,j}$ is generated as (6) and the content of shift registers is refreshed as $\mathbf{g}_{i+1,j}^{(0)}$, for $0 \le i \le kb - 1$. Based on (5), $\mathbf{p}_j$ can be derived by repeating kb times of (6). Notice that the numbers of registers, AND gates and XOR gates are based on the submatrix size z.

By observing the generator matrix $\mathbf{G}$ of the $(672, 336)$ LDPC code, the content of $\mathbf{g}_{i,j}^{(0)}$ is almost zeros. Moreover, the output of the XOR gate only changes when the content of the corresponding shift register is one, resulting in inefficient storage and operation units. Therefore, the modified architecture called adder-accumulator-shift register (AASR) is presented to eliminate redundant hardware. The main idea is to combine the shift operation and accumulation together such that the content of coefficient registers is constant. In order to give a clear description,

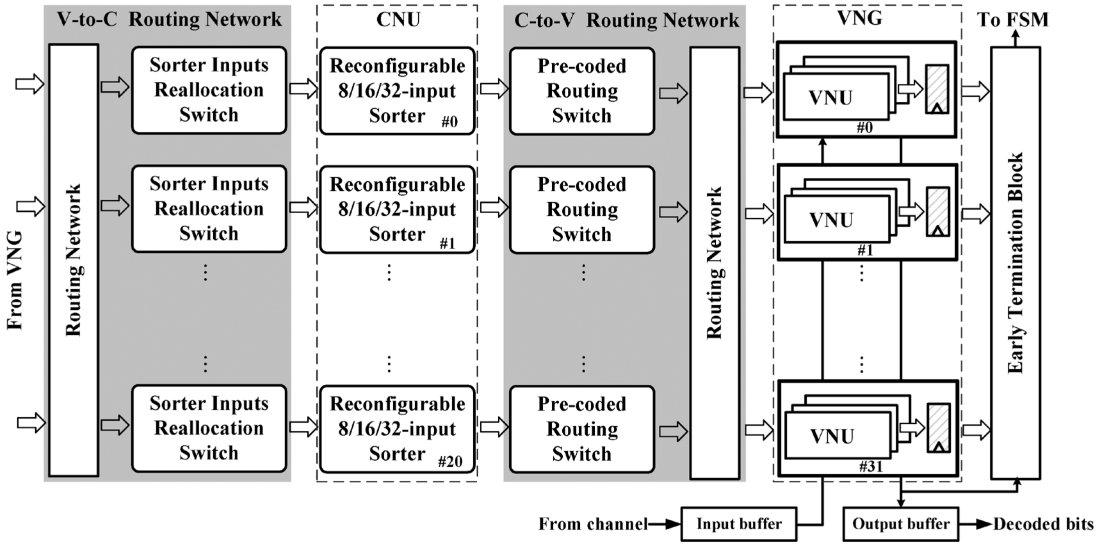


Fig. 6. Overall architecture of the LDPC decoder.

TABLE III
COMPARISON OF THE PROPOSED AASR AND THE CONVENTIONAL SRAA

	Proposed AASR	SRAA
Register	$z+n(\mathbf{y}_j)$	$2z$
AND gate	$n(\mathbf{y}_j)$	z
XOR gate	$n(\mathbf{y}_j)$	z
ROM size	$kb \times n(\mathbf{y}_j)$	$kb \times z$

Fig. 5(b) shows the proposed AASR architecture without hardware reduction. Moreover, assume $\mathbf{g}_{i,j}^{(0)}$ can be expressed into bit level as

$$\mathbf{g}_{i,j}^{(0)} = (g_0, g_1, g_2, \cdots, g_{z-1}). \tag{7}$$

Since the content of coefficient registers is fixed, the outputs of AND gates in AASR (from left side to right) are in the form of

$$(a_i g_0, a_i g_1, a_i g_2, \cdots, a_i g_{z-1}). \tag{8}$$

In SRAA architecture, the content of accumulator register s_0 is

$$a_{iz} g_0 + a_{iz+1} g_{z-1} + a_{iz+2} g_{z-2} + \cdots \tag{9}$$

To obtain $a_{iz}g_0 + a_{iz+1}g_{z-1}$, the right-cyclic shift operation in SRAA is replaced of the left-cyclic shift of accumulation. Hence, the inputs of XOR gate come from the right side of the accumulator register and AND gate.

To eliminate redundant hardware, we define the parity-section vector $\mathbf{y}_j$ as

$$\mathbf{y}_j = \text{bitwise OR}\left(\mathbf{g}_{0,j}^{(0)}, \mathbf{g}_{1,j}^{(0)}, \cdots, \mathbf{g}_{kb-1,j}^{(0)}\right), \text{ for } 0 \le j \le mb-1. \tag{10}$$

In Fig. 5(c), the example of parity-section vector $\mathbf{y}_j = (0,1,1,0,0,\ldots,0,0,1)$ and $n(\mathbf{y}_j) = 3$ denotes the total number of 1 appeared in $\mathbf{y}_j$. Therefore, the register, the AND gate, and the XOR gate can be removed where the position of $\mathbf{y}_j$ is 0 without affecting the accumulation result. Since $\mathbf{y}_j$ is always smaller than $\mathbf{z}$ in the generator matrix, the numbers of registers for $\mathbf{y}_j$, AND gates, and the XOR gates can be reduced according to $n(\mathbf{y}_j)$. Table III shows the comparison of conventional SRAA and the proposed AASR. For (672, 336) LDPC code and (672, 588) LDPC code, the reductions of parity-section weighting $n(\mathbf{y}_j)$ in average are 54.2% and 10.8%, respectively. For the configurable encoder which supports four code rates, the reduction of $n(\mathbf{y}_j)$ is only 12.7% due to (672, 420) LDPC code, in which the reduction of $n(\mathbf{y}_j)$ is almost 0%.

III. PROPOSED LDPC DECODER ARCHITECTURE

The architecture of the LDPC decoder is shown in Fig. 6. It contains five main blocks, including 21 CNUs, 32 VNGs, V-to-C routing network, C-to-V routing network, and early termination block. The check node units update from check node to variable node of each layer. Each variable node group consists of 21 VNUs, and the VNUs process variable node to check node updating. For reducing the overhead supporting the multiple LDPC codes, the common characteristics are found by analyzing the provided parity-check matrices to simplify the routing networks between CNUs and VNUs. The early termination block determines whether the decoding process should be terminated or not according to syndrome checking. The details of each block except the early termination block, which will be shown in Section V, are described in the following sections.

A. CNU: Reconfigurable 8/16/32-Input Sorter

Since the min-sum algorithm is adopted, the check node unit requires sorters to find the first minimum value (min), the index of min (min_index) and the second minimum value (2nd_min) of q_{nm}. Among the four LDPC codes, the number of rows ranges from 84 to 336 and the row degree ranges from 8 to 32. For example, (672, 336) LDPC code has 336 rows with row degree equal to 8 while (672, 588) has only 84 rows with row degree equal to 32. The irregular numbers of rows and row degrees correspond to the different numbers of input sorter and check node units. In order to achieve the requirement of four code rates, a reconfigurable 8/16/32-input sorter is proposed.

Since a 32-input sorter can be composed of four 8-input sorters, one CNU of (672, 588) LDPC code could be composed of four CNUs of (672, 336) LDPC code. The required

TABLE IV
DATA DEPENDENCY OF VNU

	cycle=0	cycle=1	cycle=2	cycle=3	cycle=4	cycle=5
Z	P	$P+m_{c\to v}^{(0)}$	$P+\sum_{l=0}^{1} m_{c\to v}^{(l)}$	$P+\sum_{l=0}^{2} m_{c\to v}^{(l)}$	$P+\sum_{l=0}^{3} m_{c\to v}^{(l)}$	$P+\sum_{l=1}^{4} m_{c\to v}^{(l)}$
$D[0]$	0	$m_{c\to v}^{(0)}$	$m_{c\to v}^{(1)}$	$m_{c\to v}^{(2)}$	$m_{c\to v}^{(3)}$	$m_{c\to v}^{(4)}$
$D[1]$	0	0	$m_{c\to v}^{(0)}$	$m_{c\to v}^{(1)}$	$m_{c\to v}^{(2)}$	$m_{c\to v}^{(3)}$
$D[2]$	0	0	0	$m_{c\to v}^{(0)}$	$m_{c\to v}^{(1)}$	$m_{c\to v}^{(2)}$
$D[3]$	0	0	0	0	$m_{c\to v}^{(0)}$	$m_{c\to v}^{(1)}$
$m_{v\to c}$	P	$P+m_{c\to v}^{(0)}$	$P+\sum_{l=0}^{1} m_{c\to v}^{(l)}$	$P+\sum_{l=0}^{2} m_{c\to v}^{(l)}$	$P+\sum_{l=1}^{3} m_{c\to v}^{(l)}$	$P+\sum_{l=2}^{4} m_{c\to v}^{(l)}$
Z'	$P+m_{c\to v}^{(0)}$	$P+\sum_{l=0}^{1} m_{c\to v}^{(l)}$	$P+\sum_{l=0}^{2} m_{c\to v}^{(l)}$	$P+\sum_{l=0}^{3} m_{c\to v}^{(l)}$	$P+\sum_{l=1}^{4} m_{c\to v}^{(l)}$	$P+\sum_{l=2}^{5} m_{c\to v}^{(l)}$

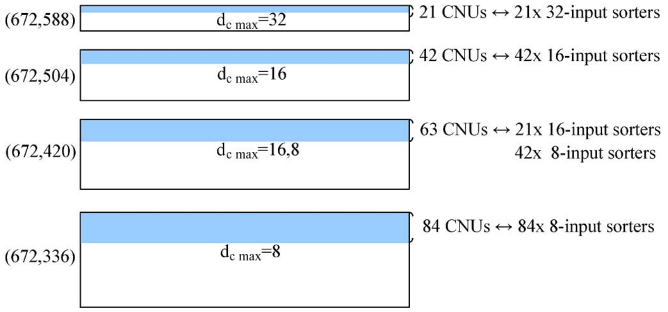

(672,588)
dc max=32
21 CNUs ↔ 21x 32-input sorters
(672,504)
dc max=16
42 CNUs ↔ 42x 16-input sorters
(672,420)
dc max=16,8
63 CNUs ↔ 21x 16-input sorters
42x 8-input sorters
(672,336)
dc max=8
84 CNUs ↔ 84x 8-input sorters

Fig. 7. Sorter requirement in a single layer of four LDPC codes.

numbers of sorters and sizes of sorters for four LDPC codes are illustrated in Fig. 7, where d_c denotes the row degree and the gray regions indicate one layer of LDPC codes. For each layer in (672, 588) LDPC code, it requires 21 32-input sorters to process check node updating as (1) within one clock cycle, and sorters can be reused as 42 16-input sorters for (672, 504) LDPC code by partitioning a 32-input sorter into two 16-input sorters. Similarly, the number of sorters can also be reused for (672, 420) and (672, 336) LDPC codes. Fig. 8 shows the architecture of reconfigurable 8/16/32-input sorter. A 16-input sorter is composed of two 8-input sorters and one extra 4-input comparator, as well as the 32-input sorter can be formed with two 16-input sorters and one 4-input comparator in the same way. According to Fig. 7, the selections of sorters for different code rates are determined by the required sizes of sorters in the gray regions.

B. VNG

Fully parallel variable node unit is proposed to accelerate message updating and to maintain high decoding throughput. There are 32 VNG and each VNG contains 21 VNUs for updating one submatrix. VNU sums up the channel value and passing message from the CNUs by using (2) and (3). Since the column degree is at most one for each layer in the $G = 4$ row-based layered scheduling, there is only one C-to-V message sending to the VNU, and one V-to-C message is updated in each variable node updating operation. Fig. 9 shows the architecture of one VNG with single input and single output VNUs. The message q_{nm} sending to mth check node is the summation of all incoming message excluding message from mth check node and the shift registers D[0]–D[3] are used to reserve the incoming message from the check node for deriving q_{nm}.

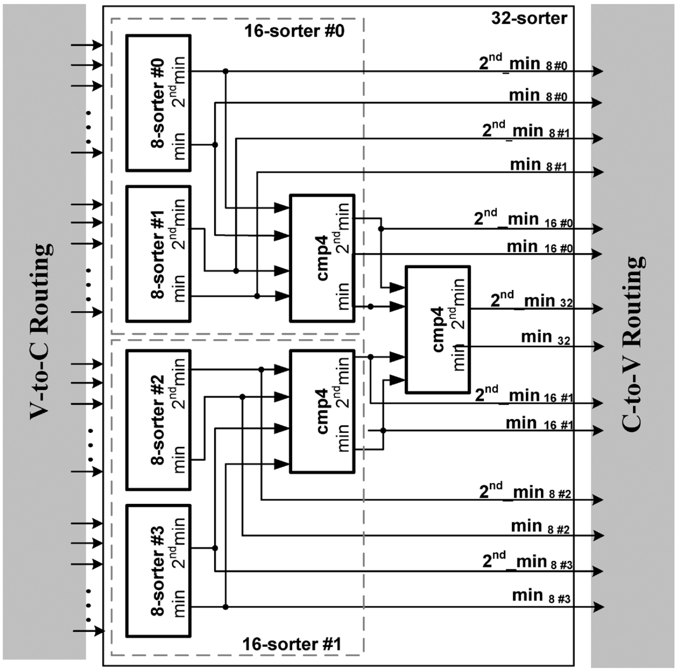


Fig. 8. Architecture of the reconfigurable 8/16/32-input sorter.

At the initialization step, the content of shift registers D[0]–D[3] is set to zero and Z is set to the channel value P. For each decoding process, the min or 2nd_min values are chosen from CNU according to the min_index. Then the sign-magnitude to two's-complement conversion is carried out. The result is sent to the shift register D[0] for next updating and D[3] is used to process variable node updating. Table IV illustrates the detail data flow of Fig. 9, where $m_{c\to v}^{(i)}$ denotes the message from the check node to variable node at $\mathrm{cycle} = i$.

C. V-to-C Routing Network: Sorter Inputs Reallocation Switch

Although the irregular check node degree can be solved with shared sorter, the interconnections between CNU and VNG are still very complicated and will dominate critical path delay. This is because q_{nm} for each sorter input will be different in terms of four LDPC codes and four updating layers. Table V is the example of the first input source for a eight-input sorter, and the v_i in the table represents the ith variable node which is

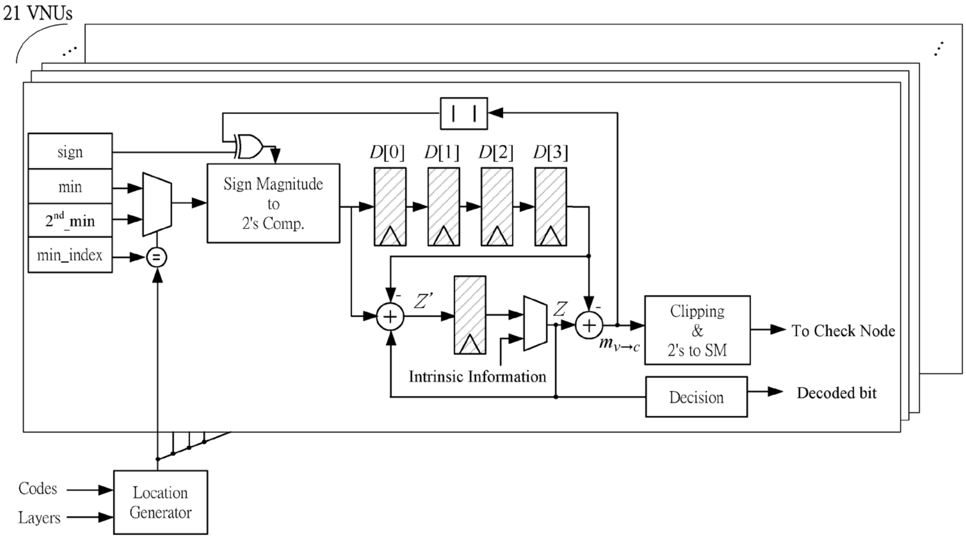


Fig. 9. Architecture of VNG.

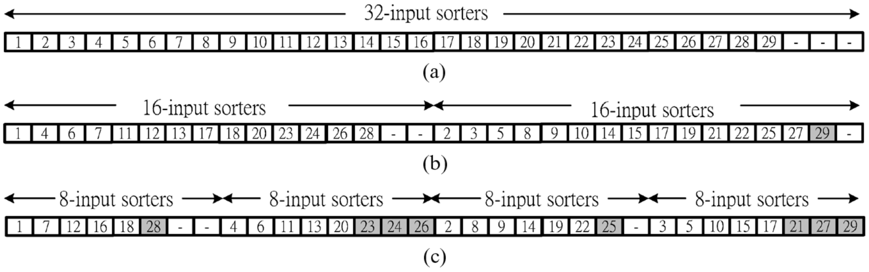


Fig. 10. Illustration of sorter inputs exchanging network. (a) One-way 32-input. (b) Two-way 16-input. (c) Four-way eight-input.

TABLE V
AN EXAMPLE OF THE INTERCONNECTION FOR A SORTER INPUT

Initial	(672,336)	(672,420)	(672,504)	(672,588)
Layer 0	v_39	v_39	v_356	v_508
Layer 1	v_60	v_60	v_377	v_509
Layer 2	v_81	v_81	v_345	v_523
Layer 3	v_18	v_18	v_366	v_514
Improved	(672,336)	(672,420)	(672,504)	(672,588)
Layer 0	v_39			
Layer 1	v_60			
Layer 2	v_81			
Layer 3	v_18			

connected to the sorter input. The original part shows that each sorter input requires a 12-to-1 multiplexer to arrange the input source. However, the overall 21 reconfigurable 8/16/32-input sorters require $672 (= 21 \times 32)$ 12-to-1 multiplexers, and such complicated interconnections will cause routing congestion and increase delay time of critical paths. Therefore, the reallocation of sorter input is exploited to alleviate the routing complexity.

The input orders of the sorter are rearranged such that the variable node sources are fixed for different layers. Fig. 10 illustrates the sorter inputs reallocation switch of the first layer. The number indicates the original order of the variable node group. First, Fig. 10(a) shows the initial order specified by (672, 588) LDPC code. Second, the 32-input sorter is decomposed into two 16-input sorters and the order is arranged to meet the interconnection requirement of (672, 504) LDPC code, as shown in Fig. 10(b). The arrangement is shown in

$$\begin{aligned}&\mathrm{sort}_{32}(\mathrm{V}_1, \mathrm{V}_2, \mathrm{V}_3, \cdots, \mathrm{V}_{28}, \mathrm{V}_{29})\\&=\mathrm{comp}_4\left(\mathrm{sort}_{16}(\mathrm{V}_1, \mathrm{V}_4, \cdots, \mathrm{V}_{28}), \mathrm{sort}_{16}(\mathrm{V}_2, \mathrm{V}_3, \cdots, \mathrm{V}_{29})\right)\end{aligned} \tag{11}$$

where sort_{32}, sort_{16}, and comp_4 represent the 32-input sorter, the 16-input sorter, and the four-input comparator, respectively.

Finally, a 16-input sorter is decomposed in the same way and is shown in

$$\begin{aligned}&\mathrm{sort}_{16}(\mathrm{V}_1, \mathrm{V}_4, \cdots, \mathrm{V}_{26}, \mathrm{V}_{28})\\&\quad= comp_4\left(\mathrm{sort}_8(\mathrm{V}_1, \cdots, \mathrm{V}_{28}), \mathrm{sort}_8(\mathrm{V}_4, \cdots, \mathrm{V}_{26})\right).\end{aligned} \tag{12}$$

Fig. 10(c) represents the final reallocation result and it leads to the fixed variable node source for each layer as shown in Table V. With the reallocation of sorter inputs, 12-to-1 multiplexers are reduced to 4-to-1 multiplexers, and the selection signal only depends on the index of layers.

D. C-to-V Routing Network: Pre-Coded Routing Switch

The C-to-V routing network encounters the similar problem as the V-to-C routing network. The inputs of each variable

TABLE VI
EXAMPLE OF INCOMING MESSAGES FROM SORTER FOR VNG0

	(672,336)	(672,420)	(672,504)	(672,588)
Layer 0	8-sorter#1 of CNU#0	16-sorter#0 of CNU#0		32-sorter of CNU#0
Layer 1	8-sorter#0 of CNU#16	16-sorter#0 of CNU#16		32-sorter of CNU#16
Layer 2	8-sorter#2 of CNU#15		16-sorter#1 of CNU#15	32-sorter of CNU#15
Layer 3	8-sorter#3 of CNU#3		16-sorter#1 of CNU#3	32-sorter of CNU#3

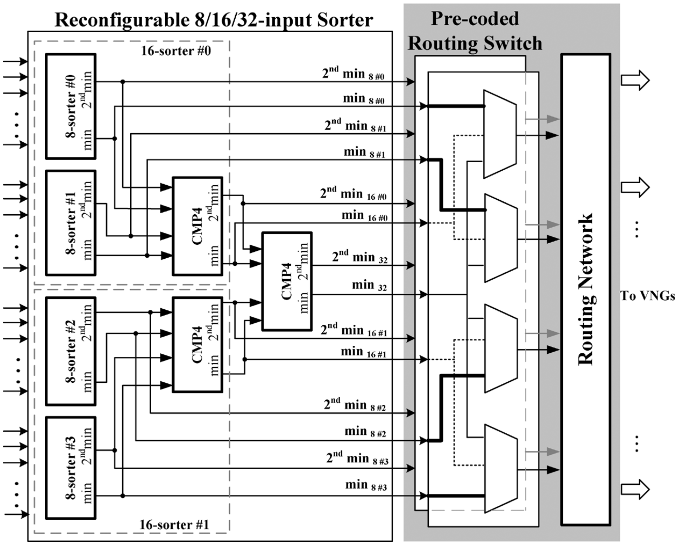


Fig. 11. Connection of pre-coded routing switch.

node come from the results of reconfigurable 8/16/32-inputs sorter. Table VI shows an example of inputs for VNG0 to imply that 12-to-1 multiplexer is also required for the routing network. Moreover, the data path contains min, 2nd_min, and min_index, which causes large bit widths of data paths. Therefore, the routing complexity is higher than that of the V-to-C network.

Based on the code structure of parity-check matrices, a pre-coded routing switch is proposed. The inputs of multiplexer come from three kinds of sorter outputs, eight-sorter, 16-sorter, and 32-sorter in Fig. 8, where the 16-sorter is composed of eight-sorters and the 32-sorter is composed of 16-sorters. While one kind of sorter output is required for the variable node, the other sorter outputs which have the same sorter sources are no longer demanded. For example, 16-sorter #0 is required for layer 0 in (672, 420) LDPC code, so the inputs from the 8-sorter #1 and 32-sorter can be ignored. Therefore, Fig. 11 illustrates the pre-coded routing switch which processes the coarse selection. The outputs of four multiplexers are corresponding to four layers and each multiplexer is selected by the code rate of LDPC code. For example, (672, 336) LDPC code requires only the results of eight-sorters, the message in bold lines at Fig. 11 are chosen in each multiplexer. Hence, the input sources of each variable node are reduced to four.

The original C-to-V routing network requires 672 12-to-1 multiplexers while the proposed architecture requires pre-coded routing switches (containing 21 × 4 3-to-1 multiplexers) followed by the modified C-to-V routing network (containing 672 4-to-1 multiplexers) as shown in Fig. 11. Because the 12 operands of four 3-to-1 multiplexers come from the same CNU, the 3-to-1 multiplexers can be merged into the CNU as the local network. More importantly, the global wires becomes 672 4-to-1 multiplexers' inputs instead of 672 12-to-1 multiplexers' inputs such that the global routing complexity can be greatly eased. Besides, the latency from the 3-to-1 multiplexer to the 4-to-1 multiplexer is roughly equal to the one of the 12-to-1 multiplexer, so the proposed routing network does not introduce additional latency while reducing the routing complexity.

The routing wire amount is proportional to the number of multiplexers' inputs. Based on the simulation results and the code structure, the bit width (w) of message is 6 and row degree (dc) is 32. The overall input number of multiplexers before optimization is

$$\begin{aligned}\mathrm{Mbit} =& \text{V-to-C Routing} + \text{C-to-V Routing}\\ =& z\times dc\times w\times 12\\ &+ N\times(\min+2^{\mathrm{nd}}\_\mathrm{min} + \mathrm{min\_index} + \mathrm{sign})\times 12\\ =& z\times dc\times w\times 12+N\times(2(w-1)+\log_2 dc+1)\times 12\\ =& 21\times 32\times 6\times 12 + 672\times 16\times 12\\ =& 177408.\end{aligned}$$

Based on Table V, one input of sorter requires a 12-to-1 multiplexer where the data bit width is 6 b. For row-based layered scheduling, the number of required 32-input sorter $(dc = 32)$ is z, then V-to-C routing is $z \times dc \times w \times 12$ multiplexers' inputs. Based on Table VI, one variable node requires a 12-to-1 multiplexer where the data bit width is 16 (including minimum value, second minimum value, index of minimum and sign magnitude). Due to fully parallel VNUs, the number of VNUs is $N = 672$ equal to the codeword length and C-to-V routing contains $N \times (\min + 2^{\mathrm{nd}}\_\mathrm{min} + \mathrm{min\_index} + \mathrm{sign}) \times 12$ multiplexers' inputs. As a result, the overall input number of multiplexer before optimization is 177408, and such large amount of global routing wires leads to the difficulty in APR step.

The overall input number of multiplexers after optimization is

$$\begin{aligned}\mathrm{Mbit}_R =& \text{V-to-C Routing} + \text{C-to-V Routing}\\ &+ \text{Pre-coded Network}\\ =& z\times dc\times w\times 4\\ &+ N\times(\min+2^{\mathrm{nd}}\_\mathrm{min} + \mathrm{min\_index} + \mathrm{sign})\times 4\\ &+ 4\times z\times(\min+2^{\mathrm{nd}}\_\mathrm{min}+\mathrm{min\_index}+\mathrm{sign})\times 3\\ =& 21\times 32\times 6\times 4+672\times 16\times 4+4\times 21\times 16\times 3\\ =& 59136 + 4032\\ =& 63168.\end{aligned}$$

For both V-to-C and C-to-V routing, the 12-to-1 multiplexer is reduced to 4-to-1. However, the pre-coded routing switch requires additional multiplexers. Based on three kinds of input sorters (8, 16, and 32) and four layers, four 3-to-1 multiplexers are required for one reconfigurable sorter and the bit width is the same as C-to-V routing network. There are z reconfigurable sorters to perform layered scheduling. So the number of additional multiplexers' inputs is $4 \times z \times (\min + 2^{\mathrm{nd}}\_\mathrm{min} + \mathrm{min\_index} + \mathrm{sign}) \times 3$. The overall elimination of multiplexer inputs for the proposed architecture is $((177408 - 63168)/177408) = 64\%$, leading to lower

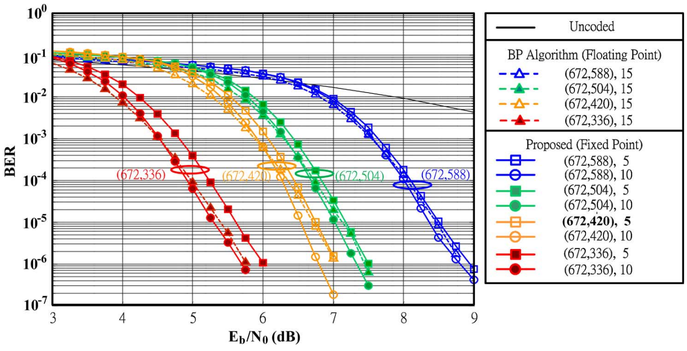


Fig. 12. BER performance for performance comparison under AWGN channel with 16-QAM modulation.

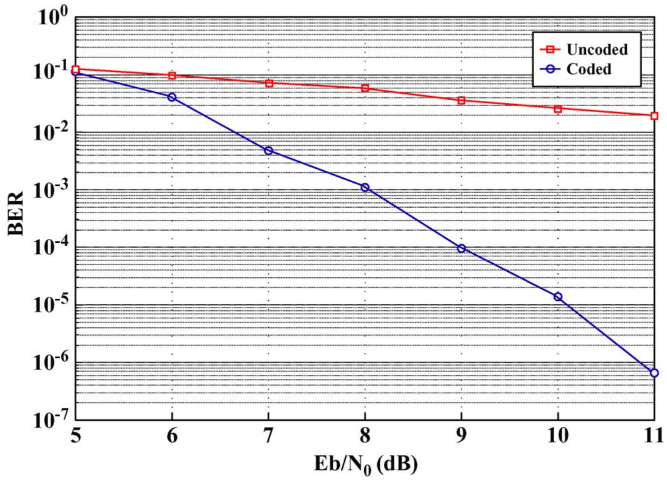


Fig. 13. BER performance of 802.15.3c system simulation with LDPC (672,336) under LOS of 1-m indoor environment, rms of 3.2 ns, 50 ppm CFO, and 50 ppm SCO.

routing overhead and high hardware utilization. Among the three terms in Mbit_R, the global wires, including C-to-V routing and V-to-C routing, gets $((177408-59136)/177408) \sim 66.6\%$ reduction of global routing network. Nevertheless, the local wires slightly increases 4032 due to the pre-coded routing switches.

IV. Simulation Results

The performance of the LDPC decoder is determined by several factors such as the iteration number, the precision of input symbols, and the normalized factor of NMS algorithm. Fig. 12 shows the BER performance of four LDPC codes under AWGN channel with 16-QAM modulation. The legend of each line denotes the LDPC codes with the maximum iteration number. The 6-b input symbol (4-b integer, 2-b decimal fraction) and the normalized factor of 0.75 is sufficient compared with the floating point simulation. Moreover, the proposed decoder can achieve similar performance of standard BP algorithm with half of the iteration number.

Furthermore, the proposed decoder is also applied to the 802.15.3c system simulation [15], [16]. The system consists of line-of-sight (LOS) channel, root-mean-square (rms) delay, carrier frequency offset (CFO), and sample clock offset (SCO) effects. Fig. 13 represents the baseband BER performance with (672, 336) LDPC code under OFDM mode, AWGN channel, and QPSK modulation. It shows that the proposed decoder can significantly improve the performance. At the system requirement of signal-to-noise ratio (SNR) equal to 11 dB, the BER of 10^{-6} can be achieved.

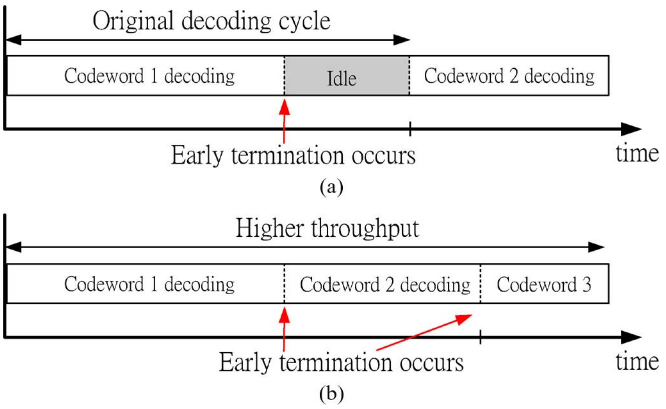


Fig. 14. Additional operating modes by early termination. (a) Low-power mode. (b) High-throughput mode.

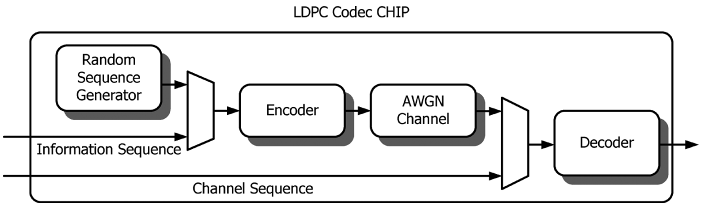


Fig. 15. Data flow of the proposed chip.

TABLE VII
Gate Count and Power Distribution of Key Module

Module		Gate count	Power
Decoder	VNUs	352.3k	258.2mW
	CNUs	170.5k	124.3mW
	Others	124.2k	50.9mW
Encoder		27.3k	3.8mW
AWGN		43.0k	9.4mW

V. Chip Implementation and Measurement Results

Through the simulation, the LDPC decoder generally corrects most of the codeword within the first few iterations, so it can be considered to early terminate the decoder to reduce power consumption or to increase throughput. In the proposed decoder, the syndrome checking method is used to determine whether the decoding process should be terminated or not. The decoded codeword $\mathbf{X}$ is sent to the early termination block to calculate the syndrome $\mathbf{S} = \mathbf{HX}$. Once the syndrome $\mathbf{S}$ is zero, the iterative decoding is terminated by gating the clock signal.

As shown in Fig. 14, there are two additional operation modes based on the early termination block. For the low-power issue, Fig. 14(a) illustrates that the decoder is idle when early termination occurs. The average power can be reduced since the power consumption is small at idle time. For high-throughput issue, the next codeword can be decoded immediately when the previous

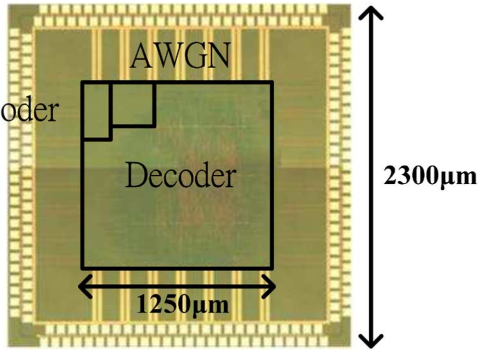


Fig. 16. Die photograph.

TABLE VIII
COMPARISON OF HIGH-THROUGHPUT LDPC DECODERS

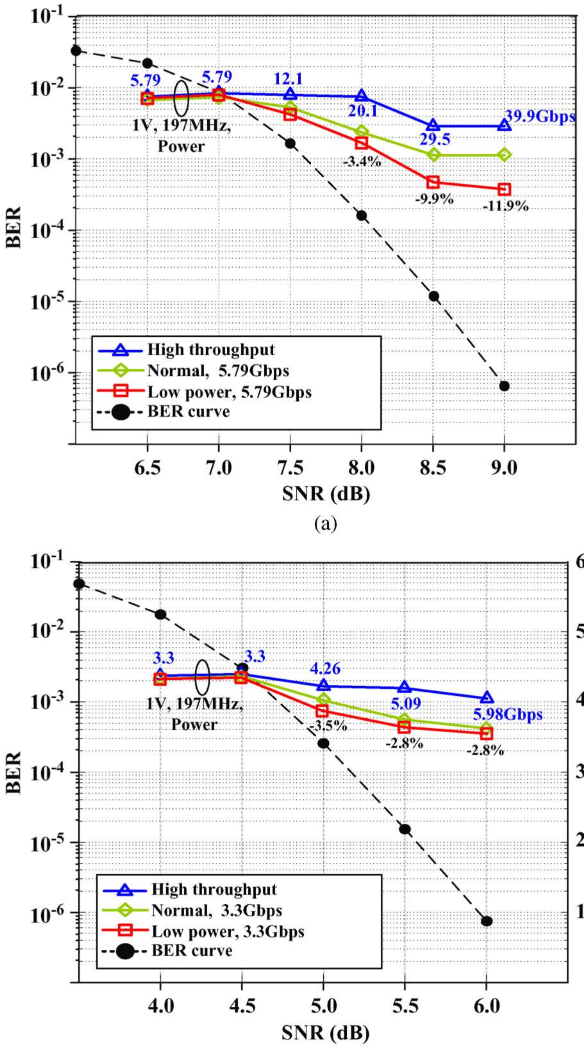
	This work		JSSC'08 [19]	JSSC'10 [20]	
CMOS Technology	65-nm		130-nm	65-nm	
Code Spec	(672,k) k=336,420,504,588		(660,480)	(2048,1723)	
Code Rate	1/2,5/8,3/4,7/8		0.73	0.84	
Decoder Gate Count (k)	647		690	NA	
Core Area (mm^2)	1.56		7.3	5.35	
Iteration (Imax)	5		15	8	
Input Quantization (bit)	6		4	4	
Supply Voltage (V)	1.0		1.2	1.2	
Clock Frequency (MHz)	197		300	700	
	Imax	Iavg	Imax	Imax	Iavg
Throughput (Gbps)	5.79 [1]	39.9 [1]	2.44 [2]	14.9 [3]	47.7 [3]
Power (mW)	361	450	1383	NA	2800
Energy Efficiency (pJ/bit)	62.4	11.3	566	NA	58.7
Hardware Efficiency (Gbps/mm^2)	3.70	25.54	0.333	2.78	8.91
Hardware Efficiency (Mbps/K-gate)	8.94	61.66	3.54	NA	NA

[1] (672,588) LDPC at $\mathrm{SNR} = 9$ dB, $\mathrm{BER} = 10^{-6}$, and 16-QAM modulation
[2] $\mathrm{SNR} = 5.5$ dB, $\mathrm{BER} < 10^{-8}$, and BPSK modulation
[3] $\mathrm{SNR} = 5.5$ dB, $\mathrm{BER} < 10^{-12}$, and BPSK modulation.

codeword is detected as a legal codeword. Hence, the decoding throughput can be enhanced by reducing decoding cycles.

The proposed LDPC encoder and decoder are integrated together for the completeness of chip testing [17]. Fig. 15 shows the data flow of the whole chip. The information can be fed from external or internal random sequence. According to the given SNR parameter, AWGN emulator generates noise value based on Box-Muller algorithm [18]. The input source of the decoder is fed from either external channel sequence or the internal AWGN channel.

The proposed LDPC codec is fabricated in 65-nm 1P10M CMOS technology and the total gate count is 703 k. Table VII shows the area and power distribution for key modules based on post-layout simulation results, and the module named as others is consist of the routing networks and the early termination block. The die photograph with the key modules is shown in Fig. 16. The chip size is 5.30 mm^2, while the core occupies 1.56 mm^2 with 73.3% utilization density.



(a)

(b)

Fig. 17. BER and Power consumption of different operating modes. (a) LDPC (672, 588). (b) LDPC (672, 336).

From the measurement results, the proposed decoder can achieve 5.79-Gb/s throughput with five decoding iterations under 1-V supply voltage. For LDPC (672, 588) at $\mathrm{SNR} = 9$ dB, the average power dissipation is 361 mW under the clock frequency of 197 MHz. Fig. 17 shows the power consumption of different operating modes and SNR conditions for LDPC (672, 588) and (672, 336). The left side on the y-axis indicates the BER performance under different SNR conditions, and the right side on the y-axis shows the power consumption under different SNR conditions. The dotted line with circles represents the BER performance results, and the solid lines present the power consumptions for different operating modes. The percentages shown in the figures represent the power reduction compared with the normal decoding without early termination. For the LDPC (672, 588) case at $\mathrm{SNR} = 9$ dB, the maximum throughput can achieve 39.9 Gb/s for the high-throughput mode while 11.9% of the power dissipation can be reduced for the low-power mode.

The chip summary and the comparison with the state of the art of high-throughput LDPC decoders are listed in Table VIII.

Imax and Iavg indicate the numbers of maximum iteration and average iteration with early termination at high-throughput mode. Instead of other designs supporting only one specific code rate, the proposed LDPC can support four code rates for HSI mode of IEEE 802.15.3c applications. In the aspects of either the energy or the hardware efficiency, our LDPC decoder can improve at least five times and 30%, respectively.

VI. Conclusion

In this paper, an area-efficient and energy-efficient LDPC codec supporting four code rates of IEEE 802.15.3c application is proposed. To meet the high-throughput requirement, the NMS algorithm with row-based layered scheduling is employed to reduce half of iterations. As an area-efficient design, several architectures, including the reconfigurable 8/16/32-input sorter architecture, sorter inputs reallocation switch, and pre-coded routing switch, are proposed to eliminate 64% inputs of multiplexers. Moreover, the early termination scheme is utilized to further diminish redundant decoding time and power consumption while detecting the valid decoded codeword. For the encoder architecture, the AASR circuit is proposed to encode codeword with fewer registers and operation units. After fabricated in a 65-nm CMOS process, the test chip with only 1.562 mm^2 can achieve the maximum throughput of 5.79 Gb/s with energy efficiency of 62.4 pJ/bit.

Acknowledgment

The authors would like to thank National Chip Implement Center (CIC), Taiwan, and United Microelectronics Corporation (UMC), Taiwan, for technology support.

References

[1] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT, 1963.

[2] D. J. C. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *Electron. Lett.*, vol. 33, no. 6, pp. 457–458, Mar. 1997.

[3] *Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specications for High Rate Wireless Personal Area Networks (WPANs)*, IEEE Std. P802.15.3c, 2009.

[4] *PHY/MAC Complete Proposal Specification*, Std. IEEE 802.11-10/0433r, IEEE 802.11 Task Group AD, May 2010.

[5] A. J. Blanksby and C. J. Howland, "A 690-mW 1-Gb/s 1024-b, rate-1/2 low-density parity-check code decoder," *IEEE J. Solid-State Circuits*, vol. 37, no. 3, pp. 404–412, Mar. 2002.

[6] C. C. Lin, K. L. Lin, H. C. Chang, and C. Lee, "A 3.33 Gb/s (1200,720) low-density parity check code decoder," in *Proc. IEEE ESSCIRC*, Sep. 2005, pp. 211–214.

[7] C. L. Chen, K. S. Lin, H. C. Chang, W. Fang, and C. Y. Lee, "A 11.5-Gbps LDPC decoder based on CP-PEG code construction," *Proc. IEEE ESSCIRC*, pp. 412–415, Sep. 2009.

[8] C. H. Liu, S. W. Yen, C. L. Chen, H. C. Chang, C. Y. Lee, Y. S. Hsu, and S. J. Jou, "An LDPC decoder chip based on self-routing network for IEEE 802.16e applications," *IEEE J. Solid-State Circuits*, vol. 43, no. 3, pp. 684–694, Mar. 2008.

[9] X. Y. Shih, C. Z. Zhan, C. H. Lin, and A. Y. Wu, "An 8.29 mm2 52 mW multi-mode LDPC decoder design for mobile WiMAX system in 0.13 m CMOS process," *IEEE J. Solid-State Circuits*, vol. 43, no. 3, pp. 672–683, Mar. 2008.

[10] M. Mansour and N. Shanbhag, "High-throughput LDPC decoders," *IEEE Trans. Very Large-Scale Integr. (VLSI) Syst.*, vol. 11, no. 6, pp. 976–996, Dec. 2003.

[11] J. Zhao, F. Zarkeshvari, and A. Banihashemi, "On implementation of min-sum algorithm and its modifications for decoding low-density parity-check (LDPC) codes," *IEEE Trans. Commun.*, vol. 53, no. 4, pp. 549–554, Apr. 2005.

[12] J. Zhang and M. Fossorier, "Suffled iterative decoding," *IEEE Trans. Commun.*, vol. 53, no. 2, pp. 209–213, Feb. 2005.

[13] M. P. C. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Trans. Inf. Theory*, vol. 50, no. 8, pp. 1788–1793, Aug. 2004.

[14] Z. Li, L. Chen, L. Zeng, S. Lin, and W. Fong, "Efficient encoding of quasi-cyclic low-density parity-check codes," *IEEE Trans. Commun.*, vol. 54, pp. 71–81, Jan. 2006.

[15] F. C. Yeh, T. Y. Liu, T. C. Wei, W. C. Liu, and S. J. Jou, "A SC/OFDM dual mode frequency-domain equalizer for 60 GHz Multi-Gbps wireless transmission," in *Proc. IEEE VLSI-DAT*, Apr. 2011, pp. 1–4.

[16] Y. S. Huang, W. C. Liu, and S. J. Jou, "Design and implementation of synchronization detection for IEEE 802.15.3c," in *Proc. IEEE VLSI-DAT*, Apr. 2011, pp. 1–4.

[17] S. Y. Hung, S. W. Yen, C. L. Chen, H. C. Chang, S. J. Jou, and C. Y. Lee, "A 5.7 Gbps row-based layered scheduling LDPC decoder for IEEE 802.15.3c applications," in *Proc. IEEE Asian Solid-State Circuits Conf.*, Nov. 2010, pp. 1–4.

[18] A. Alimohammad, S. F. Fard, B. F. Cockburn, and C. Schlegel, "A compact and accurate Gaussian variate generator," *IEEE Trans. Very Large-Scale Integr. (VLSI) Syst.*, vol. 16, no. 5, pp. 517–527, May 2008.

[19] A. Darabiha, A. C. Carusone, and F. R. Kschischang, "Power reduction techniques for LDPC decoders," *IEEE J. Solid-State Circuits*, vol. 43, no. 8, pp. 1835–1845, Aug. 2008.

[20] Z. Zhang, V. Anantharam, M. J. Wainwright, and B. Nikolic, "An efficient 10GBASE-T ethernet LDPC decoder design with low error floors," *IEEE J. Solid-State Circuits*, vol. 45, no. 4, pp. 843–855, Apr. 2010.

Shao-Wei Yen received the B.S. and M.S. degrees from the National Chiao-Tung University, Hsinchu, Taiwan, in 2004 and 2006, respectively, both in electronics engineering, where he is currently working toward the Ph.D. degree in electronics engineering.

His general research interests include VLSI implementation of error control codes and digital communication.

Shiang-Yu Hung received the B.S. and M.S. degrees from National Chiao-Tung University, Hsinchu, Taiwan, in 2008 and 2010, respectively, both in electronics engineering.

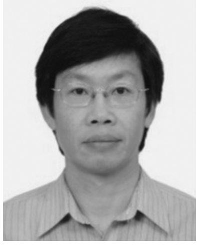
He is currently with MediaTek Inc., as a Digital Hardware Engineer. His general research interests include VLSI implementation of error control codes and wireless communication systems.

Chih-Lung Chen received the B.S. degree and M.S. degrees from National Chiao-Tung University, Hsinchu, Taiwan in 2004 and 2006, respectively, in electronics engineering, where he is currently working toward the Ph.D. degree in electronics engineering.

His general research interests include VLSI implementation of error control codes and wireless communication systems.

Hsie-Chia Chang received the B.S., M.S., and Ph.D. degrees from National Chiao Tung University, Hsinchu, Taiwan, in 1995, 1997, and 2002, respectively, all in electronics engineering.

From 2002 to 2003, he was with OSP/DE1 in MediaTek Corporation, working in the area of decoding architectures for Combo single chip. In February 2003, he joined the faculty of the Electronics Engineering Department, National Chiao-Tung University, where he has been a Professor since August 2010. His research interests include algorithms and VLSI architectures in signal processing, especially for error control codes and crypto-systems. Recently, he has also committed himself to designing high code-rate ECC schemes for flash memory and multi-Gb/s chip implementations for wireless communications.



Shyh-Jye Jou 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, Hsinchu, Taiwan, in 1984 and 1988, respectively.

He joined the Electrical Engineering Department, National Central University, Chung-Li, Taiwan, from 1990 to 2004, and became a Professor in 1997. Since 2004, he has been a Professor with the Electronics Engineering Department, National Chiao Tung University, and became the Chairman from 2006 to 2009. In August 2011, he became the Dean of Office of International Affair, National Chiao Tung University. He was a Visiting Research Professor with the Coordinated Science Laboratory, University of Illinois, Urbana-Champaign, during the 1993–1994 and 2010 academic years. In the summer of 2001, he was a Visiting Research Consultant with the Communication Circuits and Systems Research Laboratory of Agere Systems. He has published more than 100 IEEE journal and conference papers. His research interests include design and analysis of high-speed, low-power mixed-signal integrated circuits, communication and Bio-Electronics integrated circuits and systems.

Dr. Jou was the Guest Editor of the November 2008 issue of the IEEE JOURNAL OF SOLID-STATE CIRCUITS. He served as the Conference Chair of IEEE International Symposium on VLSI Design, Automation and Test (VLSI-DAT) and International Workshop on Memory Technology, Design, and Testing. He also served as Technical Program Chair or Co-Chair in IEEE VLSI-DAT, International IEEE Asian Solid-State Circuit Conference, IEEE BIOMEDICAL CIRCUITS AND SYSTEMS, and other international conferences. He received the Outstanding Engineering Professor Award, Chinese Institute of Engineers at 2011.

Chen-Yi Lee (M'01) received the B.S. degree from National Chiao Tung University, Hsinchu, Taiwan, 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 180 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 (2000/8–2003/12), and the microelectronics program coordinator of Engineering Division under National Science Council of Taiwan (2003/1–2005/12). He was the former IEEE Circuits and Systems Society Taipei Chapter Chair.