A Multi-mode LDPC Decoder for IEEE 802.16e Mobile WiMAX

Kyung-Wook Shin* and Hae-Ju Kim**

Abstract—**This paper describes a multi-mode LDPC decoder which supports 19 block lengths and 6 code rates of Quasi-Cyclic LDPC code for Mobile WiMAX system. To achieve an efficient implementation of 114 operation modes, some design optimizations are considered including block-serial layered decoding scheme, a memory reduction technique based on the min-sum decoding algorithm and a novel method for generating the cyclic shift values of parity check matrix. From fixed-point simulations, decoding performance and optimal hardware parameters are analyzed. The designed LDPC decoder is verified by FPGA implementation, and synthesized with a 0.18-μm CMOS cell library. It has 380,000 gates and 52,992 bits RAM, and the estimated throughput is about 164 ~ 222 Mbps at 56 MHz@1.8 V**

Index Terms—**Low-density parity-check (LDPC) codes, LDPC decoder, layered decoding, min-sum (MS) algorithm; WiMAX**

I. INTRODUCTION

With the increasing demands for high data rate wireless and multimedia applications, forward error correction (FEC) coding schemes become more important for reliable communication. Recent years, low-density parity-check (LDPC) codes, which were first proposed by R. Gallager [1] in the early 1960's and re-discovered by MacKay and Neal [2], have been receiving a lot of attention due to their remarkable error correction capabilities near the Shannon's limit. Many current and next generation communication standards such as WLAN (IEEE 802.11n) [3], Mobile WiMAX (IEEE 802.16e) [4], DVB-S2 [5] and 10GBaseT (IEEE 802.3an) have adopted or are considering the use of LDPC codes. Recently, there are many interesting research works on LDPC codes and decoders, including code construction, decoding algorithm and architecture, multi-mode or/and multi-standard design [6-9].

LDPC codes can be efficiently decoded by belief propagation (BP) algorithm or sum-product (SP) algorithm [10, 11]. It was originally formulated in the form of two-phase decoding, which is based on iterative exchanges of messages between check nodes and variable nodes on Tanner graph. Although the SP algorithm has high-precision message passing and excellent error-correcting performance, it is well known that hardware implementation is inefficient because its complex computations require large hardware. To reduce the complexity of the check node operation of the BP algorithm, min-sum (MS) algorithm [12, 13] was introduced. Since MS algorithm has lower computation complexity with little scarification of performance, it is preferred for hardware implementation of LDPC decoder. The layered decoding of SP and MS algorithms is possible, which has a decoding schedule based on layer sequence. It has the advantages of faster convergence speed and lower hardware complexity than the two-phase decoding.

LDPC decoder architectures can be classified into three categories: fully parallel, serial and partly parallel. LDPC decoder design has a tradeoff between error-

Manuscript received Jul. 15, 2011; revised Nov. 21, 2011.

* Kumoh National Institute of Technology - EE, Kyungbuk 730-701, Korea
** Silicon Works, Inc., Daejeon, Korea
E-mail : kwshin@kumoh.ac.kr

correction performance, hardware complexity and decoding throughput. Although various decoding algorithms and architectures for LDPC codes have been proposed until recent years, there are many challenges including flexible architecture for multi-mode and multi-standard operations, memory requirements and hardware complexity for high throughput.

In this paper, we propose a design technique to reduce the size of check node memory of MS-based layered LDPC decoder. In addition, a decoder architecture supporting 114 operating modes for 19 code lengths and 6 code rates of Mobile WiMAX LDPC codes and a prototype design are presented.

This paper is organized as follow: Section II briefly introduces Quasi-Cyclic (QC)-LDPC code for Mobile WiMAX system and layered decoding algorithm. In Section III, we describe our LDPC decoder architecture and memory reduction technique with comparisons. The implementation results of our LDPC decoder are given in Section IV.

II. LDPC CODES AND DECODING ALGORITHMS

1. LDPC Code Structure of Mobile WiMAX System [4]

The QC-LDPC code of the Mobile WiMAX system is defined by a parity-check matrix (PCM) **H** of size $M\times N$ where N is code length and M is the number of parity-check bits in the code. The PCM **H** is expanded from a binary base matrix $\mathbf{H_b}$ of size $m_b\times n_b$ where $m_b=M/z_f$ and $n_b=N/z_f$ with expansion factor z_f. Each LDPC code in the set of LDPC codes for Mobile WiMAX is defined by a PCM **H** as

$$H=\begin{bmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,n_b} \\ P_{2,1} & P_{2,2} & \cdots & P_{2,n_b} \\ \cdots & \cdots & \cdots & \cdots \\ P_{m_b,1} & P_{m_b,2} & \cdots & P_{m_b,n_b} \end{bmatrix}. \quad (1)$$

where $P_{i,j}$ is either one of a set of $z_f\times z_f$ circularly right-shifted identity matrices (i.e., permutation matrices) or a $z_f\times z_f$ all-zero matrix. In the IEEE 802.16e specification, 19 expansion factors which represent permutation matrix sizes are defined ranging from 24 to 96 with an increment of 4. The maximum value of m_b and the constant value of n_b are 12 and 24, respectively. Table 1 summarizes the LDPC code parameters of the IEEE 802.16e standard.

Table 1. LDPC code parameters of IEEE 802.16e

parameters	specification			
Code length (N)	576+96f (0≤f≤18)			
Sub-matrix size (z_f)	24+4f (0≤f≤18)			
Code rate (R)	1/2, 2/3(A,B), 3/4(A,B), 5/6			
No. of layers (m_b)	1/2	2/3(A,B)	3/4(A,B)	5/6
	12	8	6	4
Operating mode	114			

The non-zero identity matrices (called sub-matrices) are circularly right-shifted by a particular shift value defined in Eq. (2). For code rates 1/2, 3/4(A, B), 2/3B, and 5/6, the shift values $\{p(f,i_f,j_R)\}$ for a code size corresponding to expansion factor z_f are derived from $\{p(i_f,j_R)\}$ by Eq. (2-a), where $z_0=96$ denotes the maximum size of sub-matrix for code length N=2304. For code rate 2/3A, the shift values are derived by Eq. (2-b) using a modulo function.

$$p(f,i_f,j_R)=\begin{cases} p(i_f,j_R), & p(i_f,j_R)\le 0 \\ \left\lfloor \dfrac{p(i_f,j_R)\cdot z_f}{z_0} \right\rfloor, & p(i_f,j_R)>0 \end{cases} \quad (2\text{-a})$$

$$p(f,i_f,j_R)=\begin{cases} p(i_f,j_R), & p(i_f,j_R)\le 0 \\ \operatorname{mod}(p(i_f,j_R),z_f), & p(i_f,j_R)>0 \end{cases} \quad (2\text{-b})$$

The PCM can be effectively represented by a bipartite graph called Tanner graph as shown in Fig. 1. It has two types of nodes, check node (CN) and variable node (VN). The VNs represent the bits of a codeword and the CNs implement the parity-check equations. Each of CNs is connected to VN, where the elements of each row in the PCM are "1"s.

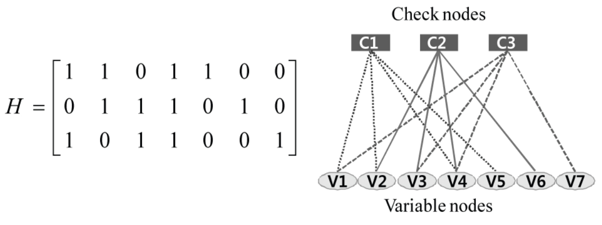

$$H=\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Fig. 1. PCM of a LDPC code and its Tanner graph.

2. Decoding Algorithms

LDPC codes can be effectively decoded using message passing algorithm which iteratively exchanges soft-information between each side of Tanner graph. The exchanged messages are the log-likelihood ratios (LLR) of the received bits in the codeword, defined as Eq. (3) where *x* and *y* are the transmitted codeword and the received codeword, respectively. During decoding process, the LLRs, which measure the reliability of the received bits based on channel observation, are propagated and updated between VNs and CNs until the decoded information bits satisfy check matrix constraint.

$$LLR(x) = \log\left(\frac{p(x \mid y=0)}{p(x \mid y=1)}\right) \tag{3}$$

The standard BP algorithm can be decomposed into two phases, CN processing and VN processing as follow:

i) CN processing: at the *q-th* iteration, each CN receives messages from the connected VNs, and computes CN updating message as Eq. (4)

$$L_{j\to i}^{q} = \left(\prod_{i'\in\omega(j)\backslash\{i\}} \alpha_{i'j}^{q-1}\right)\cdot\phi\left(\sum_{i'\in\omega(j)\backslash\{i\}}\phi\left(\beta_{i'j}^{q-1}\right)\right). \tag{4}$$

where $\alpha_{ij}^{q-1} = sign(L_{i\to j}^{q-1})$ and $\beta_{ij}^{q-1} = \left|L_{i\to j}^{q-1}\right|$. In Eq. (4), function $\phi(x)$ is defined as

$$\phi(x) = -\log\left(\tanh\left(\frac{x}{2}\right)\right) = \log\left(\frac{e^{x}+1}{e^{x}-1}\right) \tag{5}$$

ii) VN processing: at the *q-th* iteration, each VN receives messages from the connected CNs, and computes VN updating message as Eq. (6)

$$L_{i\to j}^{q} = F_i + \sum_{j'\in\Omega(i)\backslash\{j\}} L_{j'\to i}^{q} \tag{6}$$

In addition, each VN computes a refined estimation on the transmitted bit that is a posteriori probability (APP) by adding up the extrinsic information of all connected CNs to the channel value F_i as Eq. (7). The sign of z_i^q can be interpreted as the hard decision on the received bit.

$$z_i^q = F_i + \sum_{j\in\Omega(i)} L_{j\to i}^{q} \tag{7}$$

The arithmetic complexity of the SP algorithm can be greatly reduced by MS algorithm using an approximation $\sum\phi(x) \approx \phi(\min(x))$ and $\phi(\phi(x)) = |x|$. With the MS algorithm, the CN processing defined in Eq. (4) can be written as

$$L_{j\to i}^{q} \approx \left(\prod_{i'\in\omega(j)\backslash\{i\}} \alpha_{i'j}^{q-1}\right)\cdot \min_{i'\in\omega(j)\backslash\{i\}}\left(\beta_{i'j}^{q-1}\right) \tag{8}$$

The function $\phi(x)$, which is typically implemented as look-up table (LUT) in hardware, can be replaced by finding a minimum value in all VN messages, thus computational complexity is significantly reduced. Additionally, it is well known that the MS algorithm is generally less sensitive to quantization errors than SP algorithm, and hence smaller finite word-length can be used to reduce logic complexity.

The layered belief propagation (LBP) algorithm [14], which is a variation of the standard BP algorithm, treats the PCM as a group of concatenated horizontal layers. Each layer represents a row of sub-matrices in Eq. (1). The LBP algorithm repeats the decoding of each horizontal layer, and updates APP messages to be passed to next layer. *Algorithm 1* shows a pseudo code of

Algorithm 1. Pseudo code of layered MS LDPC decoding

```
Initialization:
 $\forall(i,j)\ with\ H(i,j)=1,\ set\ F_i = 2r_i/\sigma^2,\ z_i[0]=F_i,\ L_{j\to i}^{0}=0$
for q from 1 to I begin
 for k from 1 to L begin
   1) VN Processing-1;
      $\forall(i,j)\ with\ H_k(i,j)=1:$
      $L_{i\to j}^{q}[k] = z_i^q[k-1] - L_{j\to i}^{q-1}[k]$
   2) CN Processing;
      $\forall(i,j)\ with\ H_k(i,j)=1:$
      $L_{j\to i}^{q}[k] = \left(\prod_{i'\in\omega(j)\backslash\{i\}} sign(L_{i'\to j}^{q})\right)\cdot \min_{i'\in\omega(j)\backslash\{i\}} L_{i'\to j}^{q}$
   3) VN Processing-2;
      $z_i^q[k] = L_{i\to j}^{q}[k] + L_{j\to i}^{q}[k]$
 end
   4) Hard Decision;
      $c_i^q = \begin{cases} 0, & z_i^q[L] \ge 0 \\ 1, & z_i^q[L] < 0 \end{cases}$
     if $c^q H^T = 0$ then goto End of decoding;
 end
End of decoding
```

layered decoding. Iteration consists of L sub-iterations, and sub-iteration corresponds to one layer processing. In *Algorithm 1*, the CN processing computes Eq. (8) and updates the check-to-variable messages with two smallest magnitudes (min0 and min1) and the sign bits of variable-to-check messages. The VN processing, which is composed of two sub-operations, computes the APPs given in Eq. (7) and updates the variable-to-check messages. It is well known that the layered decoding can reduce the number of iterations since the latest extrinsic messages are passed to and are employed by subsequent layers within the current iteration.

III. LDPC Decoder Architecture

This section describes the proposed LDPC decoder architecture supporting 114 operation modes of QC-LDPC codes for Mobile WiMAX system, which implements the layered MS decoding algorithm shown in *Algorithm 1*. Design considerations to achieve efficient implementation are as follows: *(i)* block-serial scheduling is adopted for multi-mode operation and small area. *(ii)* An efficient way for storing CN messages is devised to minimize memory requirement. *(iii)* An efficient scheme for generating shift values for multiple code lengths and code rates is proposed. *(iv)* An optimized configuration of HROM to store the base model PCMs is employed. *(v)* The word-length of messages is determined by fixed-point simulation considering tradeoff between hardware complexities and decoding performance.

1. Overall Architecture

Fig. 2 shows the top-level architecture of our LDPC decoder which is composed of five parts: four banks of decoding function unit (DFU), CN memory, APP memory, H-ROM & SVG for storing and generating shift values, and a permuter block. Since the sub-matrix size z_f is defined by $z_f=24+4f$ $(0\leq f\leq 18)$ ranging from 24 to 96, a high degree of flexibility is required in decoder architecture. Our decoder exploits a partially parallel architecture which processes one sub-matrix in a clock cycle using *96* DFUs. Since the parallelism factor z_f varies from 24 to 96 with an increment of 4, the 96 DFUs are grouped into four banks and each bank consists of 6 sub-groups of 4 DFUs which are selectively activated according to the parallelism factor z_f. It results in a simple control as well as a reduction of overall power consumption by deactivating the banks and sub-groups that are not being used. Each DFU is independent from all others since there is no data dependence between adjacent CNs. The CN memory stores CN updating messages to be used in the next iteration processing, and the APP memory holds APP messages to be used in the next layer processing.

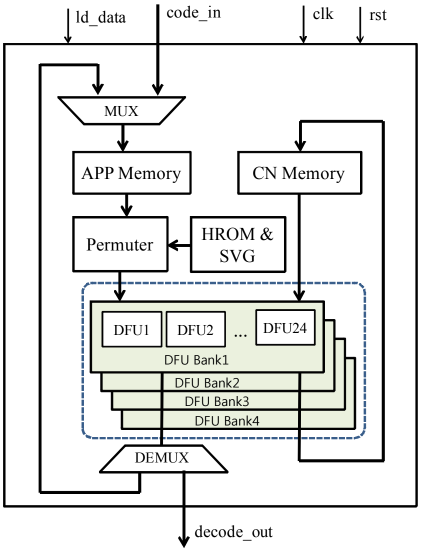


Fig. 2. Proposed architecture for multi-mode LDPC decoding.

As mentioned in Section I, the QC-LDPC code of Mobile WiMAX has 114 operation modes for supporting 19 code lengths and 6 code rates, and sub-matrices of a PCM corresponding to an operating mode have particular circular shift values as defined in Eq. (2). Since the Mobile WiMAX standard defines only 6 base model PCMs for the largest code length (N=2304), we need to generate the rest 108 PCMs using Eq. (2). As is well known, the division, floor function $\lfloor x \rfloor$ and modulo function in Eq. (2) require complicated hardware. In this paper, we devised an efficient circuit for generating shift values of 108 PCMs as shown in Fig. 3. The z_f/z_0 LUT block in Fig. 3 stores the quantized values of z_f/z_0 with some correction factors for compensating quantization errors. The shift values for all code lengths and code rates are generated by multiplying the shift values of 6 base model PCMs stored in HROM with the quantized values stored in LUT. Our method for generating the shift values is implemented in 5,531 gates including HROM to store 6 base model PCMs, and it uses only 13.4% of gate counts needed in

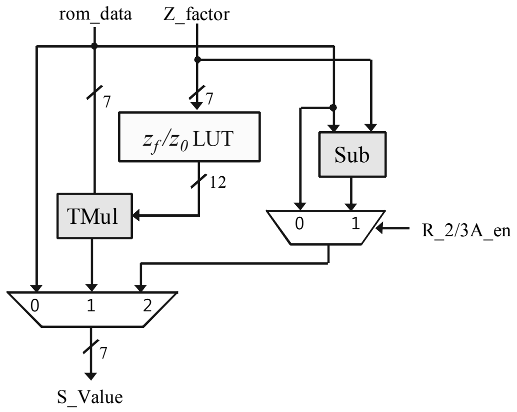


Fig. 3. Shift value generator (SVG) block.

direct implementation using LUTs in [15] and [17].

2. Decoding Function Unit

Fig. 4 shows the data path of DFU that implements the layered MS decoding algorithm described in *Algorithm 1*. It consists of SM-TC and TC_SM blocks converting sign-magnitude to 2's complement and vice versa, a min_detector finding two minimum values, an adder, a subtractor, and a FIFO. Each DFU reads CN messages of the previous iteration from CN memory and the APP messages of the previous layer from APP memory, calculates VN messages, and then finds minimum (*min0*) and semi-minimum (*min1*) values from VN messages for each layer over all VNs. The *min0* and *min1* are the new CN messages to be used to update VN. It also computes the new APP messages by adding the new CN messages to the current VN messages. These new CN and APP messages are stored in CN memory and APP memory, respectively, so that they are used in the next iteration and in the next layer processing.

3. Check Node Memory

The word-length of the messages processed in DFU and stored in memories influences the hardware costs of DFU and memory requirements. The APP memory stores messages of one layer, but CN memory needs to store CN messages of the entire layers. Therefore, CN memory requires a large hardware overhead. In this paper, we focus on a technique to reduce efficiently the size of CN memory. Fig. 5(a) shows the basic structure of conventional CN memory which stores the entire CN messages of *L* layers. With fixed-point word-length of *w*-bits, the total size of CN memory is ($w{\times}z_f{\times}S_l{\times}L$) bits, where S_l denotes the number of non-zero sub-matrices in a layer. For the code length 2304 and code rate 1/2 with word-length of *w=8* bits, the size of CN memory

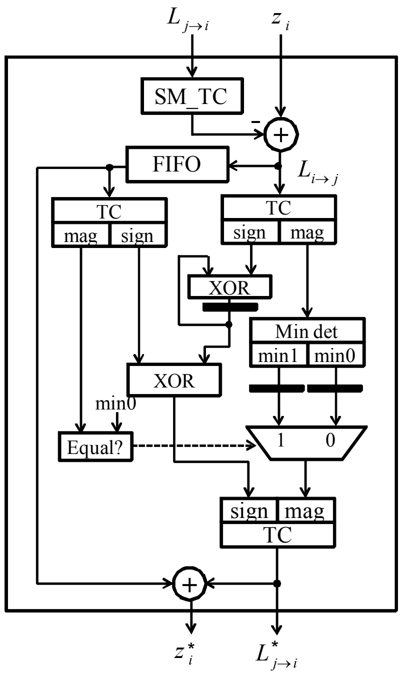
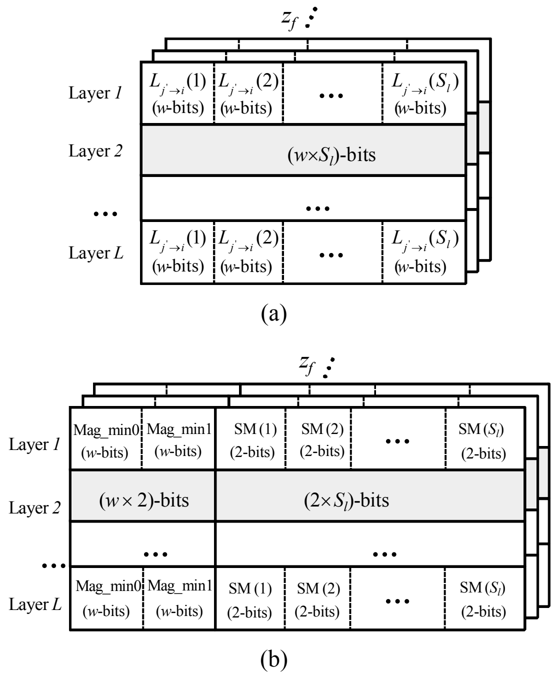


Fig. 4. Datapath of DFU.



(a)

(b)

Fig. 5. CN memory configurations (a) Conventional, (b) Proposed.

becomes 64,512 bits.

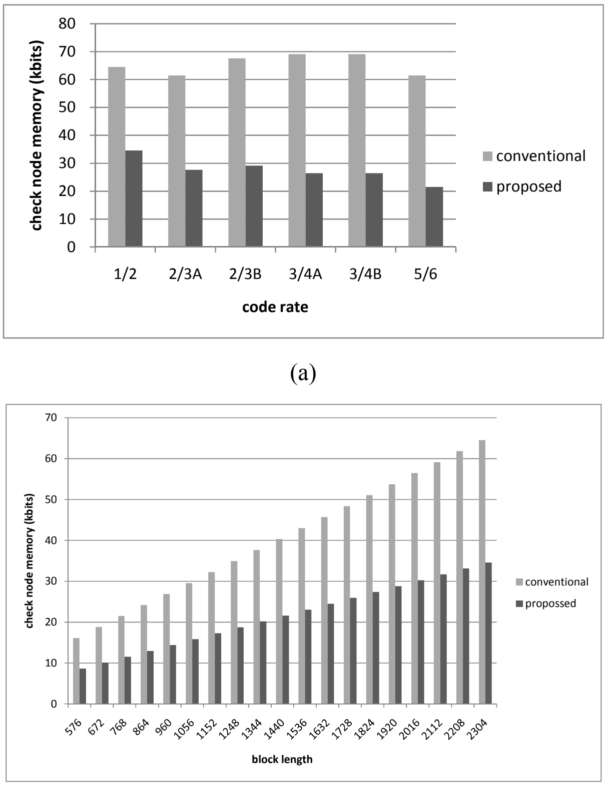
To reduce the CN memory size, we propose a new memory structure as shown in Fig. 5(b). Note that each layer consists of S_l non-zero sub-matrices whose size is $z_f{\times}z_f$. It means that each layer can be considered as a matrix having z_f rows by $z_f{\times}S_l$ columns, where S_l denotes the number of non-zero sub-matrices in a layer. From the CN processing of *Algorithm 1*, each layer has z_f *min1*s and $z_f{\times}(S_l\text{-}1)$ *min0*s. The key idea of our memory reduction comes from the fact that there is no need to store all the $z_f{\times}(\ S_l\text{-}1)$ *min0*s since they have the same value. Therefore, we can store only z_f *min0*s rather than $z_f{\times}(S_l\text{-}1)$ *min0*s for each layer. As shown in Fig. 5(b), our CN memory stores z_f *Mag_min0*s and *Mag_min1*s with ($z_f{\times}S_l$) *SM* data for each layer. The *Mag_min0* and *Mag_min1* represent the magnitudes of *min0* and *min1*, respectively. The 2-bit *SM* indicates the sign (i.e., positive or negative) and the type of minimum (i.e., *min0* or *min1*). The basic concept for storing the CN messages in a compressed way is similar to the method in [17-19], but our CN memory structure and implementation are different from them. Since CN memory does not store all the *min0*s, we need to restore the 2's complement value of each CN message using the *Mag_min0*s, *Mag_min1*s and *SM* information stored in the CN memory. SM_TC block as shown in Fig. 4 is used in DFU to convert the sign-magnitude value to 2's complement value. The hardware overhead of the SM_TC block is trivial when compared to the amount of CN memory reduction.

Table 2 shows a comparison of CN memory sizes for code length 2304 and code rate 1/2 for Mobile WiMAX. For word-length of *w=8* bits, the proposed method requires only 34,560 bits, which reduces CN memory by 46% compared to the conventional method. Fig. 6 compares the CN memory sizes for various code rates and code lengths of Mobile WiMAX system. Note that much higher memory reduction is obtained for the higher code rate.

Table 2. Comparison of CN memory sizes

Conventional	Proposed	
$w\times z_f\times S_l\times L$ $(8\times 96\times 7\times 12=64{,}512)$	*Mag_min0, Mag_min1*	*SM*
	$w\times z_f\times 2\times L$ $(8\times 96\times 2\times 12=18{,}432)$	$2\times z_f\times S_l\times L$ $(2\times 96\times 7\times 12=16{,}128)$
	Total size of CN memory (bits)	
	18,432+16,128=34,560	

*For rate-1/2 (2304,1152) code of Mobile WiMAX

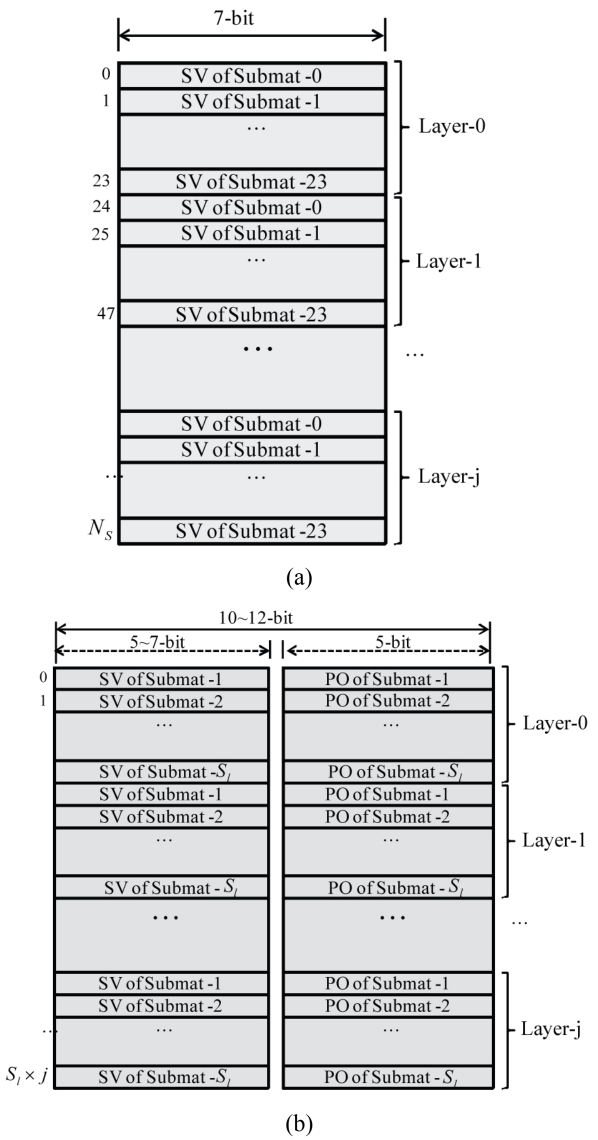


(b)

Fig. 6. Comparison of CN memory sizes (a) For various code rates (code length: 2304), (b) For various code lengths (code rate: 1/2).

4. HROM

The QC-LDPC code for Mobile WiMAX system has 6 base model PCMs which are stored in HROM and used for generating shift values of other 108 PCMs. Most of the sub-matrices in base model PCMs are zero matrices. For example, the PCM for code length 2304 and code rate 1/2 has 212 zero sub-matrices out of 288 sub-matrices. From this observation, we can reduce HROM size by storing only non-zero sub-matrices as depicted in Fig. 7(b), instead of storing all the sub-matrices as shown in Fig. 7(a). In Fig. 7, N_s denotes the total number of sub-matrices in a PCM and S_l denotes the total number of non-zero sub-matrices in a layer. In our method, the shift values and positions of non-zero sub-matrices are stored with 10~12 bits, thus we can reduce the HROM size by 17% compared to the conventional method storing all the sub-matrices including zero sub-matrices.



(a)

(b)

Fig. 7. HROM configuraions (a) Conventional, (b) Proposed.

5. Fixed-point Simulation

For hardware implementation, it is important to decide an optimal word-length of messages since there are a tradeoff between hardware cost and bit-error-rate (BER) performance. Fig. 8 shows the fixed-point simulation results of our decoder for various word-lengths of messages. In the fixed-point simulation, code length 2304 and code rate 1/2 was chosen and the maximum number of iteration was fixed to 8. As shown in Fig. 8, the BER performance is very poor when the word-length of integer part is less than 5 bits, and the BER performance has trivial difference for the word-lengths of integer part greater than 5 bits. Based on this analysis, word-length of 8 bits (5 bits for integer part and 3 bits for fractional part) was chosen for our LDPC decoder. Fig. 9 shows the fixed-point simulation results for various code lengths and code rates with maximum iteration set to 8.

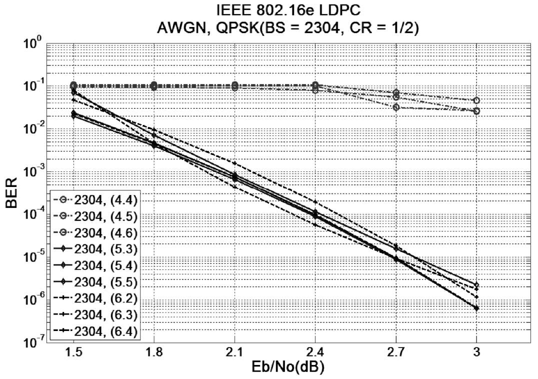
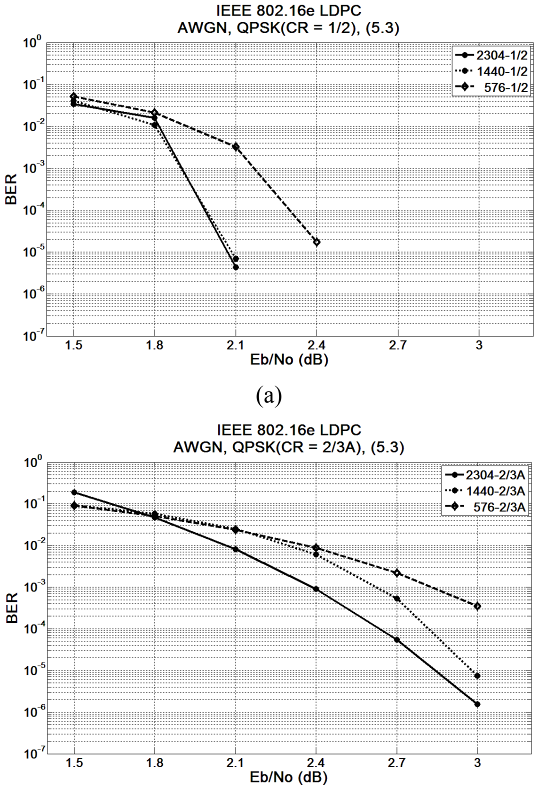


Fig. 8. BER performance for various word-lengths.



(a)

(b)

Fig. 9. Fixed-point simulation results (a) Code rate 1/2, (b) Code rate 2/3A.

IV. IMPLEMENTATION RESULTS

The LDPC decoder was implemented as a synthesizable

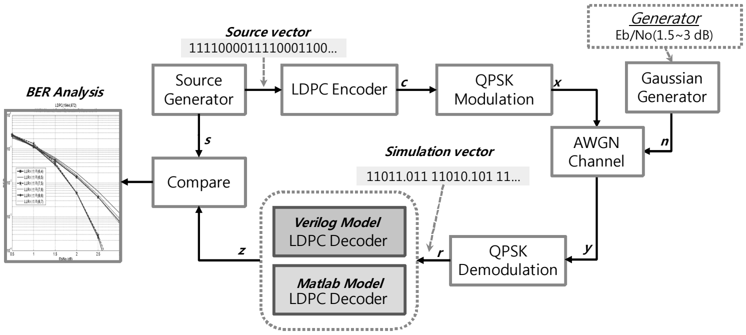


Fig. 10. Verification and performance evaluation flow.

Verilog HDL model, and its decoding performance was evaluated by simulation and FPGA implementation. As shown in Fig. 10, test vectors in the range of Eb/No= 1.5~3.0 dB with 0.3 dB step are generated and decoding performance is analyzed using Matlab. Fig. 11 shows BER performance of the decoder for code length 2,304 and code rate 1/2, which was obtained by functional simulation of Verilog HDL model with maximum iteration set to 8.

Fig. 12(a) shows the setup using Xilinx XC5vx50t-1ff1136 device to verify the decoder. RS232 transceiver and wrapper modules are embedded on FPGA along with the LDPC decoder to interface with RS232 port. Test data generated by Matlab are sent to FPGA with appropriate control signals for decoding, and the decoded data obtained from FPGA are used to analyze the decoding performance. Fig. 12(b) shows a part of FPGA verification results, indicating that the decoded output from FPGA are identical with the functional simulation results, thus the designed LDPC decoder works correctly.

The decoder synthesized using a 0.18-μm cell library has 380,000 gates and total 52,992 bits RAM including 18,432 bits of APP memory. Since our decoder processes one sub-matrix in a clock cycle, it requires total $((S_l+1)\times L)$ clock cycles to finish one iteration, where S_l denotes the number of non-zero sub-matrix in a layer and L denotes the number of layers. The estimated throughput is 164~222 Mbps at 56 MHz@1.8 V. Table 3 compares our decoder with the state-of-the-art LDPC decoders for Mobile WiMAX. Note that our decoder requires the smallest memory and comparable gate counts.

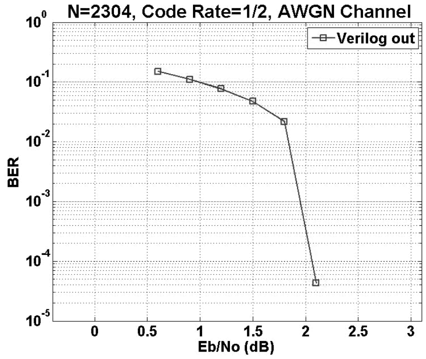
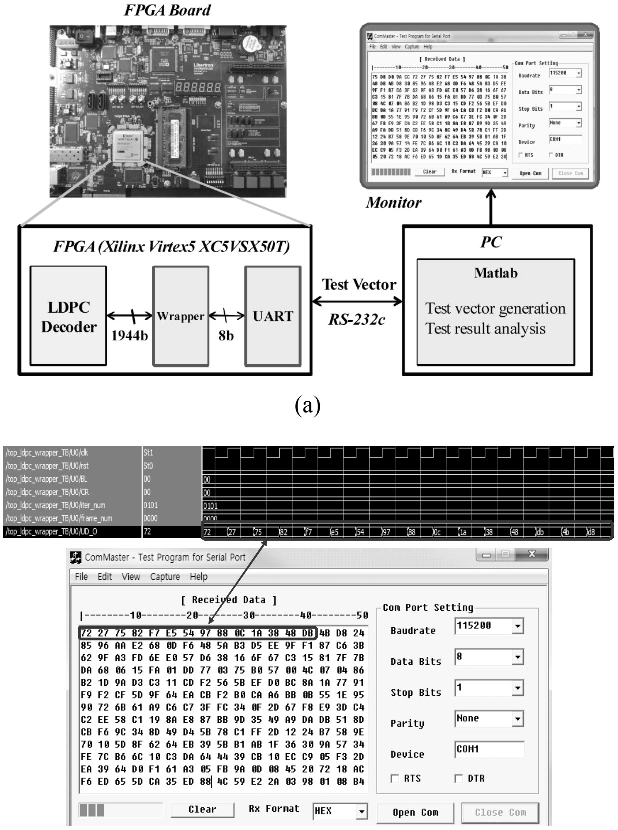


Fig. 11. BER performance of the designed decoder.



(a)

```
/top_ldpc_wrapper_TB/U0/clk        St1
/top_ldpc_wrapper_TB/U0/rst        St0
/top_ldpc_wrapper_TB/U0/BL         00     00
/top_ldpc_wrapper_TB/U0/CR         00     00
/top_ldpc_wrapper_TB/U0/iter_num   0101   0101
/top_ldpc_wrapper_TB/U0/frame_num  0000   0000
/top_ldpc_wrapper_TB/U0/UD_O       72     72 27 75 82 f7 e5 54 97 88 0c 1a 38 48 db 4b d8
```

ComMaster - Test Program for Serial Port

```
[ Received Data ]
|--------10--------20--------30--------40--------50
72 27 75 82 F7 E5 54 97 88 0C 1A 38 48 DB 4B D8 24
85 96 AA E2 68 0D F6 48 5A B3 D5 EE 9F F1 87 C6 3B
62 9F A3 FD 6E E0 57 D6 38 16 6F 67 C3 15 81 7F 7B
DA 68 06 15 FA 01 DD 77 03 75 B0 57 00 4C 07 04 86
B2 1D 9A D3 C3 11 CD F2 56 5B EF D0 BC 8A 1A 77 91
F9 F2 CF 5D 9F 64 EA CB F2 B0 CA A6 BB 0B 55 1E 95
90 72 6B 61 A9 C6 C7 3F FC 34 0F 2D 67 F8 E9 3D C4
C2 EE 58 C1 19 8A E8 87 BB 9D 35 49 A9 DA DB 51 8D
CB F6 9C 34 8D 49 D4 5B 78 C1 FF 2D 12 24 B7 58 9E
70 10 5D 8F 62 64 EB 39 5B B1 AB 1F 36 30 9A 57 34
FE 7C B6 6C 10 C3 DA 64 44 39 CB 10 EC C9 05 F3 2D
EA 39 64 D0 F1 61 A3 05 FB 9A 0D 08 45 20 72 18 AC
F6 ED 65 5D CA 35 ED 88 4C 59 E2 2A 03 98 01 08 B4
```

Com Port Setting

Baudrate: 115200

Data Bits: 8

Stop Bits: 1

Parity: None

Device: COM1

□ RTS □ DTR

Rx Format: HEX

(b)

Fig. 12. (a) Verification setup, (b) FPGA verification result of the designed decoder.

Table 3. Comparison of LDPC decoders for WiMAX

	[15]	[16]	[17]	[18]	This paper
Code length	576~2304	2304	576~2304	576~2304	576~2304
Code rate	1/2	1/2	1/2~5/6	1/2~5/6	1/2~5/6
Multi-mode	19	1	114	114	114
Gate counts	420 K	N/A	380 K	N/A	380 K
Memory (bits)	58,368	87,752	89,856	82,944	52,992
Frequency (MHz)	83.3	950	150	260	56
Throughput (Mbps)	60~222	2,200	105	205	164~222
Technology	130 nm	90 nm	90 nm	130 nm	180 nm

V. CONCLUSIONS

In this paper, a multi-mode LDPC decoder supporting 19 code lengths and 6 code rates for Mobile WiMAX system is described. It adopts block-serial architecture which processes a sub-matrix of $z_f \times z_f$ in parallel using 96 DFUs grouped into four banks to support 114 operation modes. A novel memory reduction technique which results in a significant reduction of CN memory and HROM compared to conventional approach is also exploited. The design techniques of this paper can be applied to any other QC-LDPC code decoders including IEEE 802.11n and DVB-S2 systems.

ACKNOWLEDGMENTS

This research was supported by Basic Science Research Program through the *National Research Foundation of Korea (NRF)* funded by the Ministry of Education, Science and Technology (No. 2010-0005589), and IDEC.

REFERENCES

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

[2] D.J.C. MacKay and R.M. Neal, “Near Shannon limit performance of low density parity check codes,” *IEE Electronic Letters*, Vol.32, No.18, pp.1645-1646, Aug., 1996.

[3] Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: enhancements for higher throughput*, IEEE Std. P802.11 n/D7.0*, 2008.

[4] IEEE 802.16e. Air interface for fixed and mobile broadband wireless access systems, IEEE P802.16e/D12 Draft, Oct., 2005.

[5] *Digital Video Broadcasting (DVB-S2) Via Satellite* [Online]. Available: http://www.dvb.org

[6] T. Mohsenin and B. Baas, “Trends and Challenges in LDPC Hardware Decoders,” *Forty-Third Asilomar Conf. on Signals, Systems and Computers*, pp.1273-1277, 2009.

[7] S. Huang, B. Xiang, B. Huang, Y. Chen, and X. Zeng, “A Flexible Architecture for Multi-Standard LDPC Decoders,” IEEE 8th Int. Conf. on ASIC (ASICON '09), pp.493-496, 2009.

[8] Y. Jung, Y. Jung, and J. Kim, “Design of Low Complexity and High Throughput Encoder for Structured LDPC Codes,” Journal of the IEEK, Vol.46-SD, No. 10, pp.61-69, Oct., 2009.

[9] J.-Y. Park, S.-J. Lee, K.-S. Chung, S.-M. Cho, J.-S. Ha, and Y.-H. Song, “A Memory-efficient Partially Parallel LDPC Decoder for CMMB Standard,” Journal of the IEEK, Vol.48-SD, No.1, pp.22-30, Jan., 2011.

[10] D.J.C. MacKay, “Good error-correcting codes based on very sparse matrices,” *IEEE Trans. Information Theory*, Vol.45, No.3, pp.399-431, Mar., 1999.

[11] J. Chen, A. Dholakia, E. Eleftheriou, M.P.C. Fossorier, and X.Y. Hu, “Reduced-Complexity Decoding of LDPC Codes,” *IEEE Trans. on Communications*, Vol.53, pp.1288-1299, Aug., 2005.

[12] S.Y. Chung, T. Richardson, and R. Urbanke. “Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation,” *IEEE Trans. on Information Theory*, Vol.47, No.657-670, Feb., 2001.

[13] M.P.C. Fossorier, M. Mihaljevic and H. Imai, “Reduced complexity iterative decoding of low-density parity check codes,” *IEEE Trans. on Communications,* Vol.47, pp.673-680, May, 1999.

[14] M.M. Mansour and N.R. Shanbhag, “High-throughput LDPC decoders,” *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, Vol.11, No.6, 976-996, Dec., 2003.

[15] X.-Y. Shih, C.-Z. Zhan, C.-H. Lin, and A.-Y. Wu,

"An 8.29 mm^2 52 mW Multi-Mode LDPC Decoder Design for Mobile WiMAX System in 0.13 um CMOS Process," *IEEE J. Solid-State Circuits*, Vol.43, No.3, pp.672-683, Mar., 2008.

[16] K. Zhang, X. Huang, and Z. Wang, "High-Through- put Layered Decoder Implementation for Quasi-Cyclic LDPC Codes," *IEEE J. Selected Areas in Communications*, Vol.27, No.6, pp.985-994, Aug., 2009.

[17] 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.

[18] S. Huang, D.B., B. Xiang, Y. Chen and X. Zeng, "A Flexible LDPC Decoder Architecture Supporting Two Decoding Algorithms," *Proceedings of 2010 IEEE Int. Sym. on Circuits and Systems (ISCAS)*, pp.3929-3932, 2010.

[19] J. Sha, Z. Wang, M. Gao and L. Li, "Multi-Gb/s LDPC Code Design and Implementation," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, Vol.17, No.2, 262-268, Feb., 2009.

Kyung-Wook Shin received the B.S. degree in Electronic Engineering from Korea Aerospace University, Gyeonggi-do, Korea, in 1984, and the M.S. and Ph.D. degrees in Electronic Engineering from Yonsei University, Seoul, Korea, in 1986 and 1990, respectively. In 1990, he worked at Electronics and Telecommunications Research Institute (ETRI) as a Research Engineer. In 1991, he joined Kumoh National Institute of Technology (KIT), Gumi, Korea, where he is currently a professor in the School of Electronic Engineering. He was a Visiting Scholar at the University of Illinois at Urbana-Champaign and the University of California at San Diego in 1995 and 2003, respectively, on his sabbatical leave from KIT. His research interests are in the area of high-speed/low-power VLSI design for digital communication, DSP and cryptography applications. He authored or co-authored over 140 research papers in journals and conference proceedings. He received "Best Teacher" Award from KIT in 2009.

Hae-Ju Kim received the B.S and M.S degrees in Electronic Engineering from Kumoh National Institute of Technology (KIT), Gumi, Korea, in 2009 and 2011, respectively. He is currently a Staff Design Engineer at Silicon Works Co., Ltd., Korea. His current research interests include encryption-decryption algorithm, error correcting codes and LED driver IC design.