An Efficient Overlapped LDPC Decoder with a Upper Dual-diagonal Structure

Yong Ki Byun, Jong Kang Park, Soongyu Kwon, and Jong Tae Kim*

***Abstract*—A low density parity check (LDPC) decoder provides a most powerful error control capability for mobile communication devices and storage systems, due to its performance being close to Shannon's limit. In this paper, we introduce an efficient overlapped LDPC decoding algorithm using a upper dual-diagonal parity check matrix structure. By means of this algorithm, the LDPC decoder can concurrently execute parts of the check node update and variable node update in the sum-product algorithm. In this way, we can reduce the number of clock cycles per iteration as well as reduce the total latency. The proposed decoding structure offers a very simple control and is very flexible in terms of the variable bit length and variable code rate. The experiment results show that the proposed decoder can complete the decoding of codewords within 70% of the number of clock cycles required for a conventional non-overlapped decoder. The proposed design also reduces the power consumption by 33% when compared to the non-overlapped design.**

***Index Terms*—Low-density parity-check(LDPC) codes, FEC, upper dual-diagonal, quasi-cyclic**

I. INTRODUCTION

Low density parity check (LDPC) codes are one of the forward error correcting (FEC) codes which are able be adopted in communication systems, such as WLAN, WiMAX/WiBro, DVB-S2, 10 GBaseT, and are even applicable to storage systems [1, 2]. LDPC codes perform near Shannon's limit and are generally improved by a randomly constructed and long bit length parity check matrix, *H*. However, due to short cycle times and limited memory size, it is difficult to make a good parity check matrix. Moreover, an LDPC decoder must have several parity check matrices corresponding to each code rate in order to support variable code rates; this may require a greater circuit area to implement. The algebraic method with an extension factor can make it easy to construct a long bit length parity check matrix [3]. The puncturing and shortening method can be used to support a variable code rate with a low area requirement [4].

The sum-product algorithm (SPA) is one of the better known message-passing techniques [5]. LDPC codes generally use the SPA to decode codewords. The SPA consists of three steps, namely, initialization, check node update (CNU), and the variable node update (VNU) and decision. The decoder mainly uses iteration for the last two steps. Because the CNU and VNU operate exclusively, the LDPC decoder has a low hardware utilization efficiency (HUE). If these two steps could be operated concurrently, we would increase the HUE and reduce the latency incurred in completing the decoding process. Several methods to accomplish this have been reported [6-8]. The decoders find a sequence which can partially execute the CNUs and VNUs concurrently. However, such scheduling techniques generally require complex and inflexible sequence controls for CNU/VNU and address map controls of the memory blocks. Its control block should be also changed when the parity check matrix changes due to a change in the code rate.

Manuscript received Apr. 27, 2012; revised Oct. 30, 2012.
School of Information and Communication Engineering, Sungkyunkwan University 300 cheoncheon-dong, Jangan-gu, Suwon-si, Gyeonggi-do, 440-746, Korea
E-mail : jtkim@skku.edu

The main contribution of this paper is to propose an overlapped LDPC decoder structure with a very simple control. It is based on a quasi-cyclic and (upper) dual-diagonal parity check structure used to support both the variable bit length decoding and efficient encoding architectures. The dual-diagonal and quasi-cycle structures for LDPC codes provide the efficient way to implement low complex encoder in large code length without significant loss of bit error rate (BER) performance [3, 9-11, 19, 20]. These methods make it possible for parts of the CNU and VNU to be operated concurrently.

II. THE SUM-PRODUCT ALGORITHM AND DUAL-DIAGONAL QUASI-CYCLIC STRUCTURE

1. The Dual-diagonal Structure

The LDPC encoding process is given by $v = sG$, where v is an n-bit codeword, s is the k-bit information, and G is the generation matrix. The generation matrix is obtained by the Gaussian elimination of H. However, Gaussian elimination removes the sparseness of H. Therefore, the encoding process becomes complex. If the parity check matrix H possesses a dual-diagonal structure, we can encode the information very efficiently into a systematic codeword without needing Gaussian elimination. If we make a sparse H matrix consisting of two parts, H_1 and H_2, we can express H=[H_1|H_2], where H_1 is made up of an $m \times k$ matrix, and H_2 is an $m \times m$ dual-diagonal matrix.

Therefore, the generation matrix can be described by $G = [I \mid H_1^T H_2^{-T}]$. H_1^T is still sparse; H_2^{-T} is an upper triangular matrix, as shown in Fig. 1(b). Therefore, by using a differential encoder, the encoding process is simplified [12]. We can apply a quasi-cyclic property to the dual-diagonal structure, which results in H_1 consisting of $L \times L$ sub-matrixes. H_2 can be constructed in two ways. One possibility is the full $mL \times mL$ dual-diagonal structure, such as the one shown in Fig. 1(a). The other method uses a dual-diagonal structure with an $L \times L$ non-shifted identity matrix and an $L \times L$ zero matrix. Both quasi-cyclic dual-diagonal structures will have similar properties, efficient encoding, and a variable bit length. Moreover, the dual-diagonal structure allows the code rate to be adjusted without any complex control. A variable code rate is simply implemented by shortening and puncturing [4].

2. The Quasi-cyclic LDPC Decoder Implementation

In the decoding process, the check nodes and variable nodes need only each other's messages in the CNU and VNU. If we have an $mL \times nL$ quasi-cyclic structure, then there are two ways of grouping the nodes. In the first method, the L nodes are assigned to one group. This requires m clock cycles to complete all of the CNUs and n clock cycles to complete all of the VNUs. The other way is to assign m nodes to each parity check group and n nodes to each variable group. This requires $2 \times L$ clock cycles to complete all of the CNUs and the VNUs. When we construct the parity check matrix with a large L factor, the former method has a more parallel property compared to the latter. Therefore, the semi-parallel architecture implemented herein is grouped using the first method in verifying the proposed design.

Generally, LDPC decoders update the check nodes corresponding to H from the first row to the last row in that order in the CNU step and update the variable nodes from the first column to the last column in that order in the VNU step in SPA. Because each VNU and CNU requires each other's messages, they cannot operate concurrently. There are a few studies on overlapped LDPC decoders which execute CNUs and VNUs concurrently [6]. The conventional overlapped LDPC decoders do not process CNUs and VNUs in order. The decoders find a sequence which can execute CNUs and VNUs partially. However, it is difficult to find such a sequence. Its control block must be changed when the parity check matrix changes in the case of a variable code rate.

III. THE OVERLAPPED (UPPER) DUAL-DIAGONAL LDPC DECODER

1. The Upper Dual-diagonal Structure

Fig. 1 shows upper-dual-diagonal structure, H_2, and its inverse transpose matrix, H_2^{-T}.

We can use the conventional differential encoder structure and still obtain the same efficiency. The upper-

$$H_2 = \begin{bmatrix} 1 & 1 & & & & \\ & 1 & 1 & & & \\ & & 1 & 1 & & \\ & & & \ddots & \ddots & \\ & & & & 1 & 1 \\ & & & & & 1 \end{bmatrix} \qquad H_2^{-T} = \begin{bmatrix} 1 & & & & & \\ 1 & 1 & & & & \\ 1 & 1 & 1 & & & \\ \vdots & & & \ddots & & \\ 1 & 1 & \cdots & 1 & 1 & \\ 1 & 1 & 1 & \cdots & 1 & 1 \end{bmatrix}$$

(a) (b)

Fig. 1. The upper-dual-diagonal structure (a) the upper-dual-diagonal form, (b) the inverse transpose matrix of H_2.

dual-diagonal structure in the overlapped LDPC decoder will reduce the number of clock cycles required for the computation to a larger extent than the dual-diagonal structure. The detailed explanation for this is given in section 3.2.

2. The Proposed Dual-diagonal LDPC Decoder Structure

The proposed structure shown in Fig. 2 requires only the (upper) dual-diagonal parity check matrix structure. All of the decoding operations are executed in the CNU and the VNU blocks with accessing the memory block. The semi-parallel scheduling is controlled by the sequence control block. It has flexibility in the change of the parity check matrix. It allows the decoders to accept the variable code rate property without any large control overhead. In conventional overlapped decoders, to change the LDPC code rate, even if the decoder uses shortening or puncturing, a new control is required in order to find the decoding sequences that allow the decoders to operate concurrently.

The proposed overlapped LDPC decoder reduces the clock latency needed to decode the received signal and simplifies the decoding algorithm by using an (upper) dual-diagonal structure. In the proposed LDPC decoder, the CNUs are executed from the first row to the last row

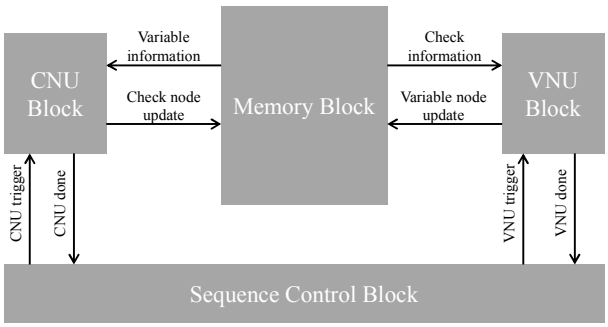
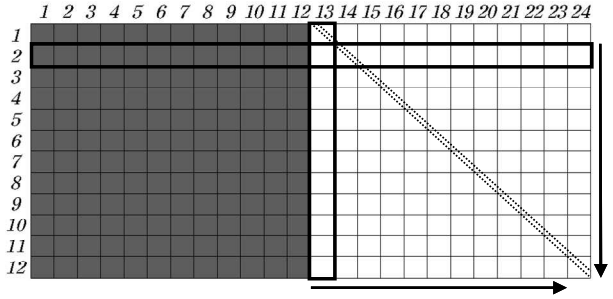


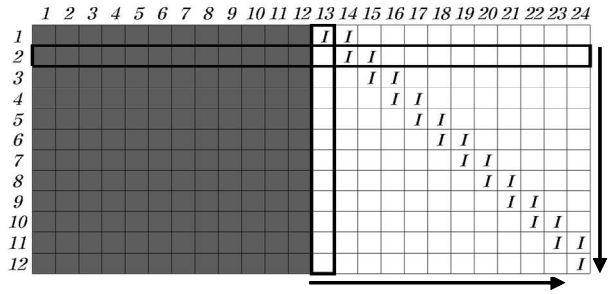
Fig. 2. The block diagram for the proposed design.

in order, as is in the case of the basic conventional non-overlapped LDPC decoder. The VNUs start from the first column of the dual-diagonal matrix, H_2. The rest of the H_1 variable nodes are then executed in order. The proposed overlapped (upper) dual-diagonal LDPC decoder expands the conventional SPA decoding steps from three steps to four. These four steps consist of "Initialization," "Only Check Node Update," "Overlapped Check Node and Variable Node Update", and "Only Variable Node Update." In the first step, Initialization, the initialization process is executed in the same manner as the conventional decoder. Next, in the Only Check Node Update step, the check nodes are updated from the first row in parity matrix H until the variable node corresponding to the first column in H_2 is ready to be executed. Then, in the Overlapped Check Node and Variable Node Update step, the check node update process is continued in an orderly fashion and the variable node update process starts from the nodes corresponding to the first column in H_2. That is, the CNUs and VNUs are executed concurrently in the third step. In the Only Variable Node Update step, the rest of the columns in H_2 and H_1 are executed. Tentative decisions can be obtained easily from the variable node update process. Therefore, all of the tentative decisions are executed with each variable node update. Once the four steps are completed, the decoder determines whether or not the decoding should iterate the decoding process from Step 2. Fig. 3 shows two examples of semi-parallel quasi-cyclic upper-dual-diagonal structures. Both of these examples have a 12×12 H_1 parity check matrix consisting of a quasi-cyclic matrix which has the extension factor, L. Fig. 3(a) shows a full upper-dual-diagonal matrix H_2. It is extended to 12(L)×12(L) using extension factor L. Fig. 3(b) shows a 12×12 upper-dual-diagonal mother matrix. Each 1-element of H_2 is substituted by the identity matrix using extension factor L.

In this example, after the Initialization step, the Only Check Node Update executes the check node update of the L nodes in one clock cycle. Then, if the VNU starts from the L variable nodes corresponding to the 13 column of the mother matrix, the CNUs and VNUs can be executed concurrently. Therefore, in the Overlapped Check Node and Variable Node Update step, the CNUs for rows 2-12 and the VNUs for column 13-23 can be



(a) The full upper-dual-diagonal structure



(b) The upper-dual-diagonal structure using $L \times L$ identity matrix

Fig. 3. The quasi-cyclic upper-dual-diagonal parity check matrix structure.

executed concurrently in 11 clock cycles. If the parity check matrix were dual-diagonal, rather than upper-dual-diagonal, step 2 would operate for rows 1 and 2 in 2 clock cycles, and the overlapped operation of the CNUs and VNUs would be executed in 10 clock cycles. This means that the use of LDPC decoders employing an upper-dual-diagonal structure can reduce the number of clock cycles to a larger extent than LDPC decoders using a dual-diagonal structure. The clock latency for both non-overlapped and the proposed designs are generalized as shown in Fig. 4.

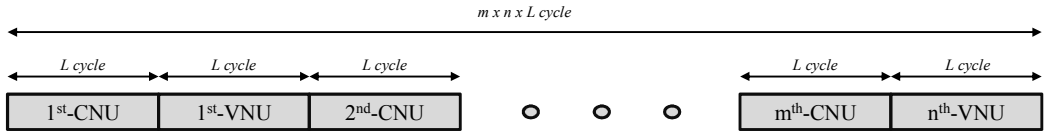
The reduced clock ratio for the overlapped design is:

$$R_{reduced} = \frac{NC_{init} + (NC_{onlyCNU} + NC_{overlappedNU} + NC_{onlyVNU}) \times N_{iter}}{NC_{init} + (NC_{CNU} + NC_{VNU}) \times N_{iter}} \approx \frac{NC_{onlyCNU} + NC_{overlappedNU} + NC_{onlyVNU}}{NC_{CNU} + NC_{VNU}} \quad (1)$$

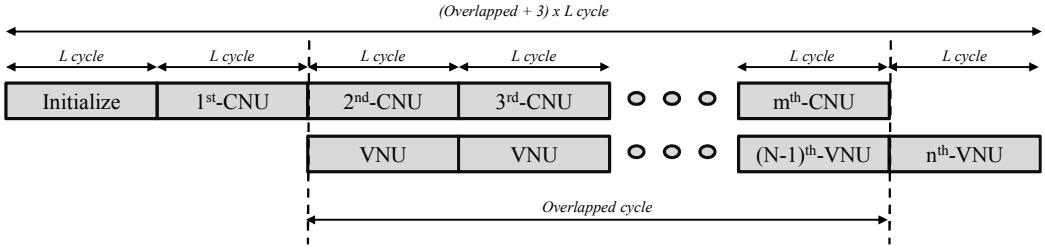
where NC means the number of clock cycles required for each step. The initialization step is completed in just one clock cycle by using buffers. For higher iteration numbers, NC_{init} can be disregarded. Therefore, in the case of the given R=1/2 WiBro spec., we can reduce the clock latency by 30.6%, as compared to the non-overlapped decoder. At R=2/3, 3/4 and 5/6, the corresponding reduced clock ratios are 22.9%, 18.3% and 13.1% from (1), respectively.

IV. THE EXPERIMENT RESULTS

In order to evaluate the decoding latency, we used a WiBro system, which is based on Mobile-WiMax [13], an 12×24 H matrix and a 2304-bit codeword through an AWGN channel with an Eb/No of 2.5 dB and a clock rate of 100 MHz. The H matrix specified in the standard, was modified by the upper dual-diagonal structure as shown



(a) clock latency for non-overlapped design



(b) clock latency for the proposed design

Fig. 4. The comparison for the decoding latency.

in Fig. 3. The number of iterations was fixed at 12. The proposed overlapped LDPC decoder consumed 3,040 ns from the initialization step to complete decoding, whereas the non-overlapped decoder consumed 4,335 ns. We saved about 30% of the clock cycles for the coded data with a 1/2 code rate. The reduction in clock latency was obtained only through the decoding sequence control.

Next, in order to validate the design efficiency of our approach at the implementation stage, we designed the proposed overlapped LDPC decoder and a conventional non-overlapped decoder in the RTL (register-transfer level). Two decoders are non-overlapped and overlapped designs with the same *H*-matrix and have the same decoding capability. The non-overlapped design is the fully-serial architecture of the LDPC decoder and it has the meaning of "smallest size but slowest speed" for given LDPC codes. We then synthesized both designs using a 90-nm process library. Table 1 shows the synthesized areas of both decoders. The overhead circuit area for the controls in the proposed design is about 10,000 gates; additional controls are required for the iterative step, "Overlapped Check Node and Variable Node Update" as described in section 3.2. This increase is only a small percentage of the overall area.

The overlapped design proposed in this paper, reduces 30% clock cycles compared to the non-overlapped one which is the reference design having the same parity check matrix. That means, even if we lower the target clock speed of the proposed design by 30% than the one of non-overlapped design, both designs have the same decoding rate. Different timing constraints mean the different target clock speeds as the design goal, defined to the non-overlapped and the overlapped designs. Moreover, such a timing margin enables to further apply the voltage scaling to the proposed design.

The energy consumption shown in Table 1 indicates that the reduced latency of our design decreases the total energy consumption for the same decoding capability with the voltage and frequency scaling technique, compared to the non-overlapped design. For the comparison, the experiment equalizes the output decoding rate as 531Mbps in two cases. Two different design constraints are defined to achieve the target speed, 531 Mbps. In order to simulate the energy consumption, we synthesized a non-overlapped design for 1.0 [V] that had a timing constraint of 10 ns and an overlapped design with a timing constraint of 13 ns for 0.7 [V] using the logic synthesis. Although the overlapped design had a less tighter timing constraint than the non-overlapped design, due to the difference in the clock cycle latency, both designs have nearly the same throughput of 531 Mbps and also satisfy the given timing constraint. We measured the power consumption with 5000-set test vectors (about 10^7-bits). The result shows that the proposed design with a supply voltage of 0.7 [V] requires only 67% of the energy of the non-overlapped design with a supply voltage of 1.0 [V]. It is clear that more power savings are achieved by voltage scaling of overlapped architecture than increased current driving by 10,000 gates overhead. Conversely, if the proposed design uses the increased supply with 1.0 [V], the decoding speed is increased up to 696 Mbps but, the total energy consumption is also slightly increased to 1.9 mJ.

Table 1. The comparison results of the conventional and proposed decoders

		Non-overlapped decoder, Vdd=1.0 [V]	Proposed LDPC Decoder, Vdd=0.7 [V]
Gate count	Initialization Block	87 K	
	Register files	213 K	
	CNU Block	180 K	
	VNU Block	280 K	
	Top Block (with control block)	970 K	980 K
Power Consum-ption	Avg. Power [nW]	109.2	75.2
	Total Energy for decoding 10^7 bits [mJ]	**2.4**	**1.6**

Table 2 summarizes the comparison results between the existing designs and the proposed design for WiBro/WiMax system. The technology library applied is different from each other, but our design has the competitive decoding rate for given clock speed and the circuit size. For FPGA designs, the gate counts are roughly estimated by the percentage of the macro-cells occupied and the supported total gate counts for given FPGA. Each memory bits are also converted to 8 equivalent logic gates.

Table 2. The comparison results to the existing studies

Items	[14]	[15]	[16]	[17]	[18]	This work
Technology	130 nm	180 nm	Altera EP2C70	Xilinx XC2V8000	Xilinx XC2V8000	90 nm
Main clock speed [MHz]	83	100	100	110	61	77
Throughput [Mbps]	61	68	350	278	70	513
Estimated gate counts	1.03M	0.50M	-	>1.5M	>0.97M	0.98M

V. CONCLUSION

In this paper, we presented an SPA, and explained its quasi-cyclic structure and dual-diagonal structure for the purpose of achieving an efficient implementation which can handle a variable bit length and variable code rate supporting the WiMax/WiBro specification. By using the proposed structure, we were able to reduce the overall clock latency needed to decode codewords by about 30% and reduced the energy consumption by 33% as compared to the conventional decoder design.

REFERENCES

[1] S. Jeon and B. V. K. V. Kumar, “Performance and Complexity of 32 k-bit Binary LDPC Codes for Magnetic Recording Channels,” *IEEE Trans. Magnetics*, vol. 46, pp. 2244-2247, Jun, 2010.

[2] H. Zhong, T. Zhong, and E. F. Haratsch, “Quasi-Cyclic LDPC Codes for the Magnetic Recording Channel: Code Design and VLSI Implementation,” *IEEE Trans. Magnetics*, vol. 43, pp. 1118-1123, Mar, 2007.

[3] M. P. C. Fossorier, "Quasi-Cyclic Low-Density Parity-Check Codes From Circulant Permutation Matrices," *IEEE Trans. Info. Theory*, vol. 50, pp. 1788-1793, Aug, 2004.

[4] M. Jiang, C. Wang, Y. Zhang, and C. Zhao, "An improved variable length coding scheme using structured LDPC codes," *IEEE Int. Conf. on WCSP*, Oct, 2010.

[5] Z. Wang, Z. Cui, and J. Sha, "VLSI Design for Low-Density Parity-Check Code Decoding," *IEEE Circuit and System Mag.*, vol. 11, pp. 52-69, Feb, 2011.

[6] Y. Chen and K. K. Parhi, "Overlapped message passing for quasi-cyclic low-density parity check codes," *IEEE Trans. Circuits and Syst.*, vol. 51, pp. 1106-1113, June, 2004.

[7] Y. Dai and Z. Yan, "Optimal Overlapped Message Passing Decoding of Quasi-Cyclic LDPC Codes," *IEEE Trans on VLSI*, Vol. 16, pp.565-578, May, 2008.

[8] L. Liu and C.-J.R. Shi, “Sliced Message Passing: High Throughput Overlapped Decoding of High-Rate Low-Density Parity-Check Codes,” *IEEE Trans. Circuits and Syst.*, Vol. 55, pp.3697-3710, 2008.

[9] S. Myung, K. Yang, and J. Kim, "Quasi-cyclic LDPC codes for fast encoding," *IEEE Trans Inform. Theory*, vol. 51, pp. 2894-2901, Aug. 2005.

[10] T. Richardson and R. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 638-656, Feb.2001.

[11] Wenming Liu, Guangxi Zhu, Yongqiang Deng, and Yejun He, "Deterministic quasi-regular LDPC codes," *VTC-2005-Fall*, vol. 1, pp 512- 516, 2005.

[12] M. Yang, W. Ryan, and Y. Li, "Design of efficiently encodable moderate-length high-rate irregular LDPC codes," *IEEE Trans. Comm.*, vol. 52, pp. 564–571, April, 2004.

[13] “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*, Oct. 2005.

[14] X. Shih, C. Zhan, C. Lin, and A. Wu, “A 19-mode 8.29 mm2 52-mW LDPC decoder chip for IEEE 802.16e system,” *Proc. of Symp. on VLSI Circuits*, pp. 16-17, 2007.

[15] T.-C. Kuo and J. A. N. Willson, “A flexible decoder IC for WiMAX QC-LDPC codes,” *Proc. of Custom Integrated Circuits Conf.*, pp.527–530, 2008.

[16] A. Blad and O. Gustafsson, “FPGA implementation of rate-compatible QC-LDPC code decoder,” *proc. of 20-th ECCTD*, pp.277.280, 2011.

[17] K. Gunnam, G. Choi, M. Yeary, and M. Atiquzzaman, “VLSI architectures for layered decoding for irregular LDPC codes of WiMax,” *Proc. of IEEE Int. Conf. Commun.*, pp. 4542–4547, 2007.

[18] K. Zhang and X. Huang, “A low-complexity rate-compatible LDPC decoder,” *Proc. of Signals,*

Systems and Computers Conf. Record of the Forty-Third Asilomar, pp.749-753, 2009.

[19] C. Yoon, E. Choi, M. Cheong and S. Lee, "Arbitrary Bit Generation and Correction Technique for Encoding QC-LDPC Codes with Dual-Diagonal Parity Structure", *Proc. of IEEE conf. WCNC*, pp.662-666, 2007.

[20] Z. He, S. Roy, and P. Fortier, "Encoder Architecture with Throughput Over 10 Gbit/sec for Quasi-cyclic LDPC Codes," *Proc. of ISCAS*, pp.3269-3272, 2006.

Yong Ki Byun received BS and MS degrees in electrical, electronic and computer engineering in 2002, 2007 from Sungkyunkwan University, Korea. His current research interests include the digital logic design, embedded system design and low power designs for SoC.

Jong Kang Park received BS and MS degrees in electrical, electronic and computer engineering in 2001, 2003 and Ph.D. degree in electric and electronics engineering from Sungkyunkwan University, Korea in 2008. His current research interests include the digital logic design, embedded system design and soft error analysis and tolerance techniques for VLSI designs.

Soongkyu Kwon received BS and MS degrees in electrical, electronic and computer engineering from Sungkyunkwan University in 2008 and 2010, respectively. He is a Ph.D. student in the department of IT convergence at Sungkyunkwan University. His current research interests include embedded system design and electronic system level design for SoC.

Jong Tae Kim is a Professor at the School of Electronic and Electrical Engineering, Sungkyunkwan University, where he has been since 1995. He received the BS degree in electronics engineering from Sungkyunkwan University in Korea in 1982 and the MS and PhD degrees in electrical and computer engineering at the University of California, Irvine, in 1987 and 1992, respectively. From 1991 to 1993 he was with the Aerospace Corporation in Elsegundo, California. He was a full-time lecturer at Chunbuk National University in Korea from 1993 to 1995. His research interests include SoC design and design methodology, embedded systems, and multi-core processor architecture.