디지털산업정보학회 논문지 제10권 제4호-2014년 12월

효율적인 정도 생성기 및 새로운 순열 기법을 가진 LT 코덱 구조*

하산 타릭** · 최 광 석***

A LT Codec Architecture with an Efficient Degree Generator and New Permutation Technique

Hasan, Md. Tariq · Choi, Goang Seog

<Abstract>

In this paper, a novel hardware architecture of the LT codec is presented where non-BP based decoding algorithm is applied. Novel LT codec architecture is designed with an efficient degree distribution unit using Verilog HDL. To perform permutation operation, different initial valued or time shifted counters have been used to get pretty well permutations and an effect of randomness. The codec will take 128 bits as input and produce 256 encoded output bits. The simulation results show expected performances as the implemented distribution and the original distribution are pretty same. The proposed LT codec takes 257.5 cycle counts and 2.575μs for encoding and decoding instead of 5,204,861 minimum cycle counts and 4.43s of the design mentioned in the previous works where iterative soft BP decoding was used in ASIC and ASIP implementation of the LT codec.

Key Words : Erasure channel, LT codes, LT codec, Degree distribution, Fountain Code

Ⅰ. Introduction

The binary erasure channel (BEC), introduced by Elias in 1955 [1], is the simplest real world channel model where data either can arrive correctly or be lost because of buffer overflows or excessive delays experienced in transmission[2]. LT codes, introduced by Michael Luby, are the first true rateless fountain codes to solve the erasure problem.

Instead of sending message bits along with the redundant bits like other channel codes, the message information is distributed among a number of encoded bits in LT codes. Therefore, if some of the transmitted bits are lost or erased in the BEC, the source information can be recovered from the remaining intact bits. Because of its efficiency and effectiveness, LT codes are being adopted in a form called Raptor codes, which is actually a concatenation of LT codes and LDPC [3], in many

* 이 논문은 2013년도 조선대학교 연구비 지원을 받아 연구되었음.

** Khulna University 전자통신공학과 조교수
*** 조선대학교 정보통신공학과 교수(교신저자)

applications like the multimedia broadcast/multicast system (MBMS) of the 3GPP, IP datacast [4-5], etc. In this paper, we present a new hardware architecture of an LT COder and DECoder (CODEC) for the codes as Luby proposed in [2], which includes fully functional degree generator, a generator matrix generation unit, an encoder and a decoder.

H. Wang proposed two BEC models for different OSI layers and a hardware architecture of the LT coder [6]. Kai, in [7], presented a hardware architecture for an encoder and soft-decision LT decoder with routers and reverse routers for the connectivity between input and output nodes. Based on [11], Alam, in [8], described an ASIC implementation of an LT codec architecture with a soft BP decoding algorithm and a modified degree distribution, and compared the performance in terms of area and cycle counts for TSMC and Samsung libraries.

Ⅱ. LT Codes

LT codes are the erasure recovery codes, in which instead of sending the original message and redundant bits as encoded output bits, the message information is distributed among the encoded output bits. This is done in way that the complete message can be recovered from a subset of the encoded bits even if some of the encoded bits are erased or lost [9].

An LT encoder receives K number of message bits $(s_1, s_2, s_3, \ldots, s_k)$, which are also known as variable nodes u, and d_n degree from $\rho(d)$ degree distribution. d_n message bits are chosen uniformly at random from u to obtain encoded bits, known as check nodes c, by modulo-2 operation, and the corresponding column of the generator matrix is generated. Eq. (1) can be used to describe the encoding process mathematically where n is the position of the check node and corresponds to the column number of the matrix G [10].

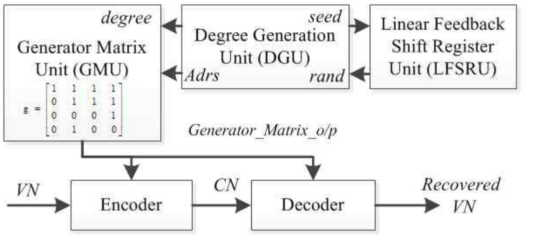
$$c_n = \sum_{k=1}^{K} s_k G_{kn} \tag{1}$$

Degree distribution is important for encoding, decoding and the overall efficiency evaluation. Luby proposed two distributions- ideal soliton distribution (ISD) and robust soliton distribution (RSD). Besides these distributions, other distributions, such as the Pareto degree distribution, the power degree distribution, Shokrallahi-proposed distributions for Raptor codes in [3].

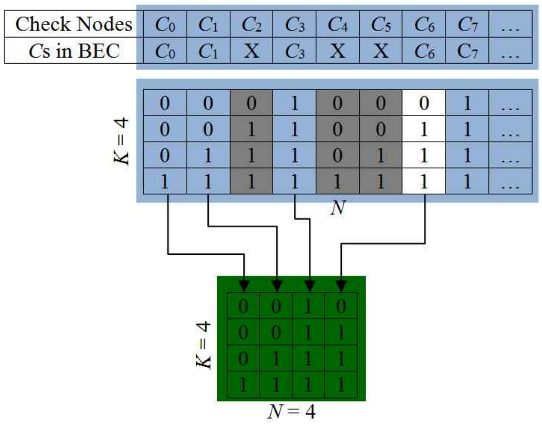
Ⅲ. Proposed Hardware Architecture of LT Codec

Hardware design for LT codec is quite challenging because of its random code construction and the variable data length characteristics. The proposed architecture is shown in Fig. 1.

The linear feedback shift register (LFSR) generates a random number; the degree generation unit (DGU) generates degrees according to RSD or any other predefined definition. In the generator matrix unit (GMU), the permutation and matrix construction operations are performed. It makes edge connections by assigning 1 (or 0 for no edge connection) between a check and a variable node depending on the magnitude of the degree. If the degree is 2, two 1s are inserted in the column of the generator matrix.



<Fig. 1> Complete architecture of the proposed LT codec



Check Nodes	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	…
Cs in BEC	C_0	C_1	X	C_3	X	X	C_6	C_7	…

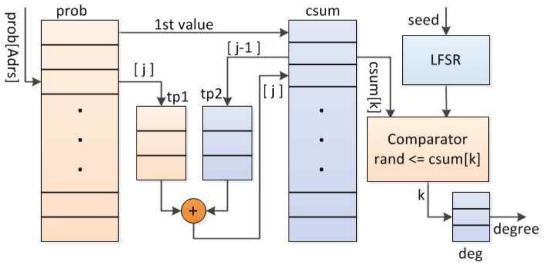
0	0	0	1	0	0	0	1	…
0	0	1	1	0	0	1	1	…
0	1	1	1	0	1	1	1	…
1	1	1	1	1	1	1	1	…

0	0	1	0
0	0	1	1
0	1	1	1
1	1	1	1

<Fig. 2> Check node selection and corresponding generator matrix formation for a 4-bit message and the encoded output bits

Finally, the encoder generates N encoded output bits, where each of the checks nodes is generated by d variable nodes. Usually, these encoded output bits are transmitted through a BEC. The decoder receives the transmitted bits and decodes it. If some of the bits are erased or lost in the BEC, it discards those bits and the corresponding column from the matrix, G, which is shown in Fig. 2 by grey color and letter X. Usually, at the receiver end, an LT decoder has all the units like encoder to generate the matrix. However, in this proposed design, the decoder also uses encoder's generated matrix. Fig. 1 shows the overall architecture of the LT codec as Luby proposed. In the figure, some functional units are integrated to a single unit to make it manageable. Each unit is described in brief in the following sections.

3.1 Hardware Architecture of the Degree Generation Unit



<Fig. 3> Hardware architecture of the degree generation unit. tp_1 and tp_2 are the temporary registers for holding the probability (*prob*) and the cumulative sum (*csum*) respectively. *prob* and *csum* are memories.

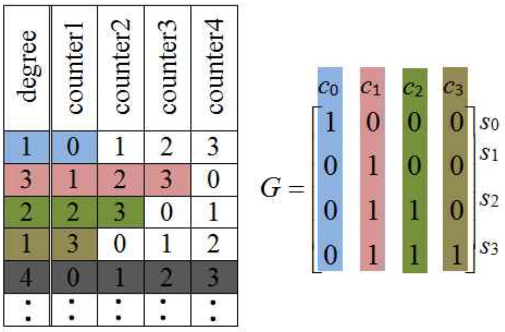
Degree distribution is an important factor determining the performance of LT codes. Fig. 3 shows the proposed degree generation unit (DGU). It receives a random number form a linear feedback shift register unit (LFSRU), and the number depends on the seed. Fig. 3 shows the complete architecture of the DGU and its outputs are considered as degree. Predefined probabilities are stored in the prob (probability) memory. However, any other degree distribution can be loaded at any time. This feature can be used when a superior degree distribution is found. From the stored probabilities, the cumulative sum is calculated according to the Eq. (2).

$$csum[j] = csum[j-1] + prob[j] \tag{2}$$

The calculation makes of two temporary registers tp_1 and tp_2, and the results are stored in memory csum. The LFSR-generated rand is compared with the magnitude of csum. There will be at least one value of csum for which rand is less than or equal to csum. The first value is accepted, and degree is chosen from the memory deg as deg[k]. Here, deg stores the predefined degrees according to the probabilities or distribution, and k is the index found from the comparison.

3.2 Hardware Architecture of the Generator Matrix Unit

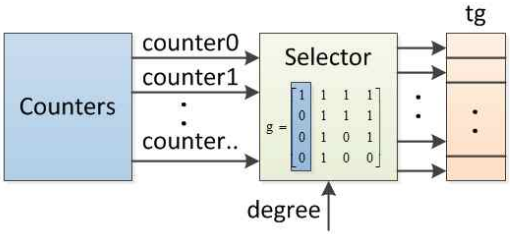
The generator matrix unit (GMU) receives the degree from the DGU. This degree indicates how many 1s are present in a column of the generator matrix G. If the degree is 3, then there will be three 1s in a column as mentioned above. In the LT encoding process, d variable nodes are randomly and uniformly selected. To do that it is better to have the permutations of K message bits. The design of this unit becomes critical as K grows. In this architecture, K counters with different initial values are used to obtain the effect of all the permutations and the randomness. The idea is shown in Fig. 4.



degree	counter1	counter2	counter3	counter4
1	0	1	2	3
3	1	2	3	0
2	2	3	0	1
1	3	0	1	2
4	0	1	2	3
:	:	:	:	:

$$G = \begin{array}{c} \\ \\ \\ \\ \\ \end{array}\begin{array}{cccc|c} c_0 & c_1 & c_2 & c_3 & \\ 1 & 0 & 0 & 0 & s_0 \\ 0 & 1 & 0 & 0 & s_1 \\ 0 & 1 & 1 & 0 & s_2 \\ 0 & 1 & 1 & 1 & s_3 \end{array}$$

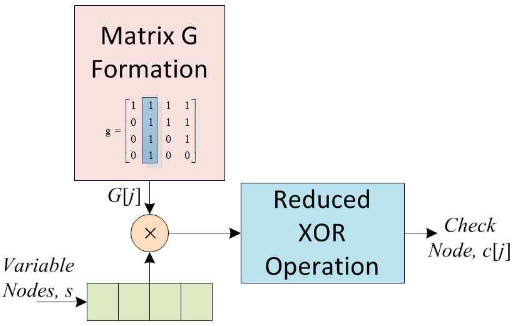
<Fig. 4> Generator matrix generation using degrees and different counters with different initial values



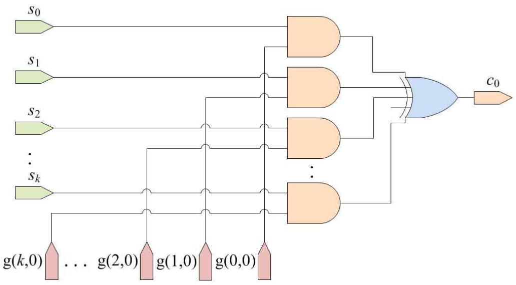
<Fig. 5> GMU architecture with different counters with different initial values to ensure non−overlapping counts

The different initial values of the counters ensure that the counts are non-overlapping. For example, consider four counters with different initial values.

For the column or address 01, the received degree is 3. Then the GMU selects first three counters and a 1 is stored in every row position indicated by the corresponding counter's count. In this example, counter1= 1, counter2 = 2, and counter3 = 3. In column c_1, 1 is inserted in row positions 1, 2, and 3; rest of the positions will have 0s. At the end of the procedure, a single column is



<Fig. 6> A column is multiplied with the message bits *s* to obtain *d* message bits, and then the bits are added together by modulo−2 operation, which produces a single−bit check node.



<Fig. 7> Multiplication and modulo−2 addition can be implemented by this simple logical operation

generated and stored in a temporary register tg, as shown in Fig. 5, and sent to the encoder and the decoder.

3.3 Hardware Architecture of the LT Encoder

In this unit, a check node is generated from a column of the generator matrix, G and the input variable nodes, s. Fig. 6 shows the encoder module, and Fig. 7 shows the logical operation in detail. The column number of the matrix is used as the address.

The address is used to select a column of G and the column entries are sent for a logical multiplication or AND operations with the variable nodes, s. In the multiplication, the corresponding d variable nodes are selected because the column contains d 1s and (K-d) 0s. These selected variable nodes are added together by modulo-2 operation, which is actually a reduced XOR operation, and the result of addition provides the encoded output of a single check node. There will be N such units, as shown in Fig. 7, for N check nodes.

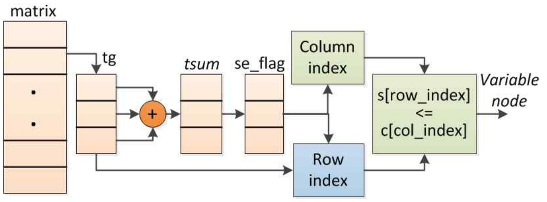
3.4 Hardware Architecture of the LT Decoder

This is one of the challenging units of the LT codec. An LT decoder has four major functional units.

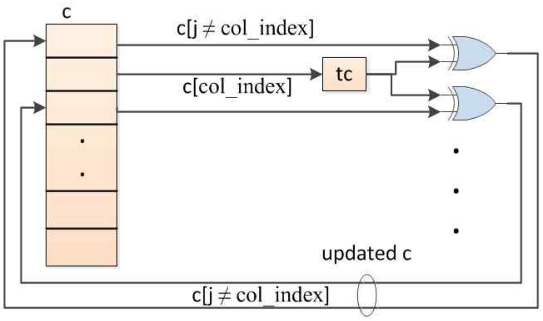
- Identifying the check nodes with a single edge, i. e., determining the row index and the column index of a single edge-check node in the matrix, G.
- Assigning the value of c to s.
- Updating the check-node register and
- Updating the matrix by assigning zeroes to every column of a particular row index.

Fig. 8 shows the architecture for the first two major operations. A complete column is copied to the temporary register tg, and the column-sum is calculated and stored in tsum. If the value of tsum is 1, a 1 is stored in the se_flag register at the address of the column index. The se_flag register is used to determine the row position or the index of the single edge in a column. Then, the c value is assigned to the s register at the row index as s[row_index] = c[col_index] as shown in Fig. 9.

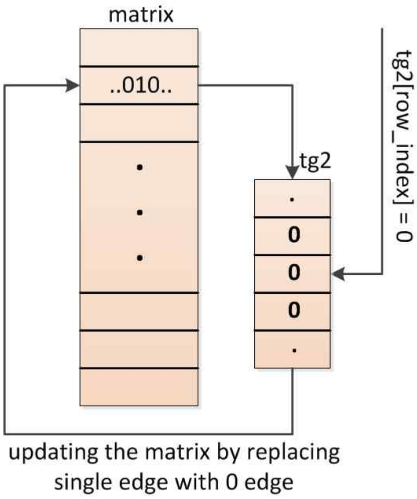
Now, the register where c is stored is updated to



<Fig. 8> Decoder unit: column-sum calculator, finding the row and the column indexes of a single-edge check node. The value of the check node is assigned to a variable node using the indices



<Fig. 9> The register where *c* is stored is updated by performing an XOR operation with every check node that is connected with the newly recovered s and the *c* value that is recently assigned to the s



<Fig. 10> Each column of the generator matrix is updated in its respective row position by replacing the single-edge value 1 with 0

make it independent of the newly recovered variable node s. This design is shown in Fig. 9.

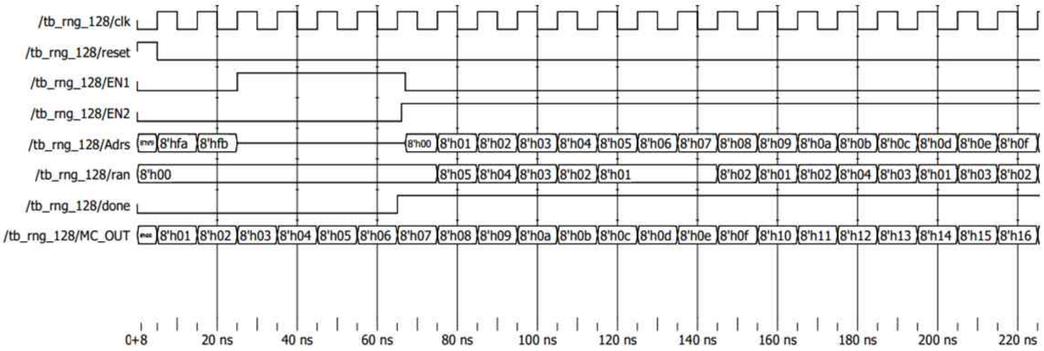
The value of c which was recently assigned to s is stored in the temporary register tc. Now, every c that is connected to the recently recovered s is updated by an XOR operation with tc; the updated c values are stored in the respective indexes. The next step is to update the matrix G, to make the check nodes free or disconnected from the newly recovered node s.

This is shown in Fig. 10. Here, each column of the generator matrix is copied to the temporary register tg_2. tg_2 is updated as tg_2 [row_index] = 0 and all the contents of tg_2 are stored in the matrix G according to the column indexes. This step is repeated N times to update the matrix.

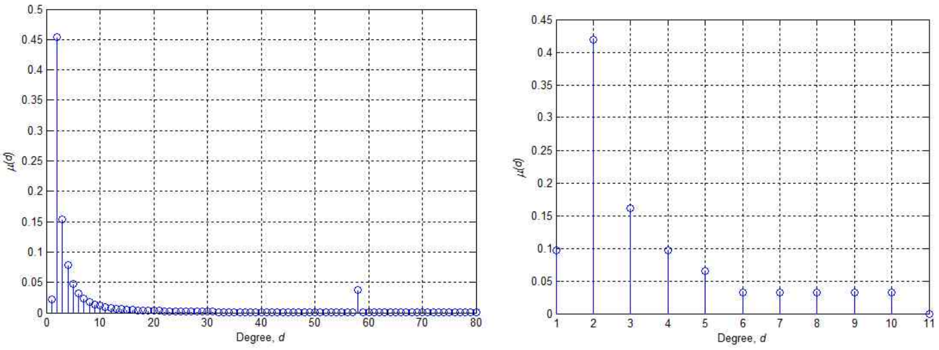
Ⅳ. Hardware Implementation and Simulation Results

In this paper, an LT codec architecture is presented as Luby proposed the code. Since the LT code is a fountain code, it ideally generates limitless bits at the output. However, this is not practically feasible. Therefore, the architecture was designed for 128 message bits and 256 encoded output bits. Verilog HDL was used for RTL, and ModelSim® was used to compile and simulate the design. For the generation of random numbers, an LFSR was used with some seeds to obtain randomness. Fig. 11 shows the output waveform of the DGU.

The signal ran shows a random number for the respective address, which is used as a degree in

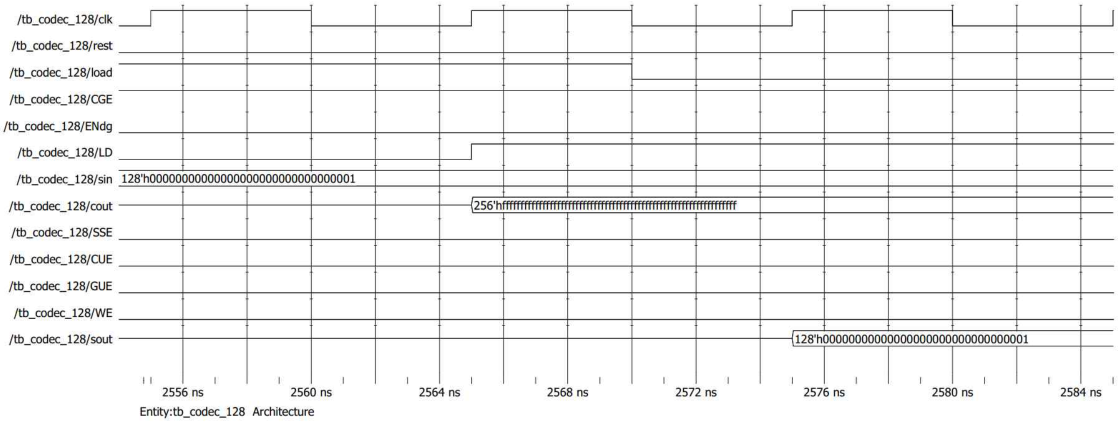


<Fig. 11> *ran* is the output random number generated by the DGU according to a distribution and it exactly follows the distribution



(a) (b)

<Fig. 12> Performance of the random number generator, DGU: (a) RSD degree distribution for $K = 128$, $\delta = 0.2$, and $c = 0.03$ and (b) RSD degree distribution implemented by the proposed hardware



<Fig. 13> Input-output waveforms of the LT codec

every encoding session. Any distribution can be implemented with some approximations, by comparing the LFSR-generated random value with the cumulative sum of probabilities. Fig. 12 shows the performance of the random number generator where (a) shows the original RSD and (b) shows the hardware implemented RSD.

Note that the original and the hardware generated distributions are almost similar. Finally, Fig. 13 shows the input-output waveforms of the codec. Here, sin represents the input variable nodes or s, cout represents the encoded output or the check nodes generated by the encoder, and sout indicates the decoded or the recovered variable nodes.

Here, 127 0s and a 1 were used as message bits and the encoded output was 256 1s. These 256 1s were used by the decoder to recover the original message bits 128'h000…1. This is shown in Fig. 13.

Ⅴ. Conclusions

In this paper, a new and fully functional LT codec hardware architecture with specially designed degree generator was proposed. The permutation was performed by multiple counters with different initial values and increment to obtain randomness in the permutation and in the generation of the generator matrix. The encoder and the decoder units were designed as Luby's proposed the code in his paper. Any degree distribution can be used with this architecture. The waveforms and distribution figures show that the design meets the expected performance. The encoder and the decoder performed quite satisfactorily and it takes 257.5 cycle counts to perform the operation and does not have complex mathematical operations.

References

[1] P. Elias, "Coding for Two Noisy Channels," in 3rd London Symp. Information Theory, London, U. K, 1955.

[2] M. Luby, "LT Codes," in 43 rd Annual IEEE Symposium on Foundations of Computer Science, 16-19 November 2002.

[3] A. Shokrollahi, "Raptor Codes," IEEE Transactions on Information Theory, vol. 52, no. 6, 2006, pp. 2551-2567.

[4] T. Mladenov, S. Nooshabadi and K. Kim, "Efficient Incremental Raptor Decoding Over BEC for 3GPP MBMS and DVB IP-Datacast Services," IEEE Transactions on Broadcasting, vol. 57, no. 2, June 2011, p. 313.

[5] Technical Specification Group Services and System Aspects: Multimedia Broadcast / Multicast Services (MBMS), Protocols and Codecs (Release 6), 3rd Generation Partnership Project (3GPP), Technical Report 3GPP (2005).

[6] H. Wang, "Hardware Designs for LT Coding," Delft University of Technology, Netherlands, 2006.

[7] K. Zhang, X. Huang and C. Shen, "Soft Decoder Architecture of LT Codes," in IEEE Workshop on Signal Processing Systems, Washington, 2008.

[8] S. M. S. Alam and G. Choi, "Design and

Implementation of LT CODEC Architecture with Optimized Degree Distribution," IEICE Electronics Express, pp. 1-10, May 29, 2013.

[9] Kongtao Wang, Zhiyong Chen and Hui Liu, "A Novel Decoding Scheme for LT-Codes in Wireless Broadcasting Systems," IEEE Communications Letters, VOL. 17, NO. 5, MAY 2013.

[10] D. MacKay, "Fountain Codes," in IEE Proceedings- Communications, 2005.

[11] 무하마드 아심 · 최광석, "무선채널에서 결합 부늣 부호들의 성취율 평가," 디지털산업정보학회 논문지, 제8권, 제1호, 2012년 3월, pp. 147-155.

■ 저자소개 ■

하산 타릭
Hasan, Md. Tariq

2014년 9월~ 현재
Khulna University 전자통신공학과 조교수
2014년 8월 조선대학교 정보통신공학과 (공학석사)
2001년 2월 Khulna University 전자통신공학과 (공학사)

관심분야 : 통신 VLSI ASIC, ASIP 설계, 채널부호
E-mail : mdthasan@gmail.com

최광석
Choi, Goang Seog

2006년 3월~현재
조선대학교 정보통신공학과 교수
2002년 2월 고려대학교 전자공학과(공학박사)
1989년 2월 부산대학교 전자공학과(공학석사)
1987년 2월 부산대학교 전자공학과(공학사)

관심분야 : 통신 및 디지털 미디어 SoC 설계
E-mail : gschoigs@chosun.ac.kr

논문접수일: 2014년 10월 20일
수 정 일: 2014년 11월 27일
게재확정일: 2014년 12월 2일