

Design and Implementation of a Latency Efficient Encoder for LTE Systems

Soo Yun Hwang, Dae Ho Kim, and Kyoung Son Jhang

The operation time of an encoder is one of the critical implementation issues for satisfying the timing requirements of Long Term Evolution (LTE) systems because the encoder is based on binary operations. In this paper, we propose a design and implementation of a latency efficient encoder for LTE systems. By virtue of 8-bit parallel processing of the cyclic redundancy checking attachment, code block (CB) segmentation, and a parallel processor, we are able to construct engines for turbo codings and rate matchings of each CB in a parallel fashion. Experimental results illustrate that although the total area and clock period of the proposed scheme are 19% and 6% larger than those of a conventional method based on a serial scheme, respectively, our parallel structure decreases the latency by about 32% to 65% compared with a serial structure. In particular, our approach is more latency efficient when the encoder processes a number of CBs. In addition, we apply the proposed scheme to a real system based on LTE, so that the timing requirement for ACK/NACK transmission is met by employing the encoder based on the parallel structure.

Keywords: Long term evolution, modem, data channel encoder, latency, 8-bit parallel processing, timing requirement.

Manuscript received Sept. 28, 2009; revised Dec. 23, 2009; accepted Jan. 18, 2010.

This work was supported by the IT R&D program of MKE/KEIT [2006-S-001-04, Development of Adaptive Radio Access and Transmission Technologies for the 4th Generation Mobile Communications].

Soo Yun Hwang (phone: +82 42 860 5569, email: syhwang@etri.re.kr) and Dae Ho Kim (email: daeho@etri.re.kr) are with the Internet Research Laboratory, ETRI, Daejeon, Rep. of Korea.

Kyoung Son Jhang (email: sun@cnu.ac.kr) is with the Department of Computer Engineering, Chungnam National University, Daejeon, Rep. of Korea.
doi:10.4218/etrij.10.0109.0584

I. Introduction

The mobile radio network technology family of the 3rd Generation Partnership Project (3GPP), as well as its predecessor, the European Telecommunications Standards Institute (ETSI), which includes a global system for mobile communications (GSM) [1], enhanced data rate for GSM evolution (EDGE) [2], universal mobile telecommunication system (UMTS) [3], and high-speed packet access (HSPA) [4] technologies, now accounts for over 85% of all mobile subscribers worldwide. The further increase in demand for high data rates in new applications such as mobile TV, online gaming, multimedia streaming, and so on has motivated 3GPP to work on the Long Term Evolution (LTE) project since late 2004 [5].

LTE [6]-[8] is a fourth-generation (4G) wireless broadband technology. It is aimed at a commercial deployment in 2010 and will offer high peak data rates (100 Mbps for downlink and 50 Mbps for uplink), low latency (10 ms round-trip delay), improved system capacity and coverage, multi-antenna support, efficient support for packet data transmission, flexible bandwidth operations (up to 20 MHz), and seamless integration with existing systems [9], [10]. The multiple input multiple output (MIMO) [11] concept has been recognized as a key new technology to achieve the required bandwidth efficiencies in LTE systems, and the orthogonal frequency division multiplexing (OFDM) [12] technique is the main physical layer design element for the LTE transmission scheme combined with either frequency division multiplexing (FDM) or time division multiplexing (TDM) as a multiple access method for shared data traffic channels. In the field of recent wireless mobile communications, many companies and institutes are designing and developing LTE systems spanning across many countries [13].

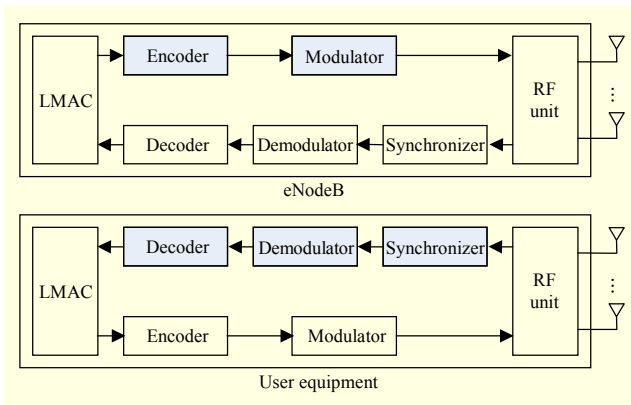


Fig. 1. Conventional modem architecture for LTE systems.

Figure 1 shows a conventional modem architecture for LTE systems. In Fig. 1, except for low medium access control (LMAC) and RF unit blocks, the shaded blocks are related with a downlink path, and the others are associated with an uplink path. The two paths consist of transmission (Tx) and reception (Rx) parts according to the side of eNodeB or user equipment (UE). The Tx part of each path is constructed with an encoder and modulator. The Rx part of each path is constructed with a synchronizer, demodulator, and decoder. Among them, the process time of the encoder is a critical implementation factor to meet the timing requirements of LTE systems since the encoder is based on binary operations. The encoder block performs the channel coding operations of the data and control streams from the LMAC layer to offer transport and control services over a radio Tx link. In particular, the data channel encoder has the longest latency because its block size is very long. In this paper, we present one way to reduce the latency of the data channel encoder. Through 8-bit parallel processing of the CRC attachment, CB segmentation, and parallel processor, we can execute the engines for turbo codings and rate matchings of each CB in a parallel manner, significantly decreasing the latency of the data channel encoder.

In the next section, we briefly explain the LTE specifications for the data channel coding scheme, while section III describes our encoder with a parallel structure for LTE systems. We present the experimental results in section IV and give concluding remarks in section V.

II. Coding Scheme of Data Channel for LTE Systems

Figure 2 shows the coding mechanism of a data channel for LTE systems [7]. In the first data channel processing step, a 24-bit CRC is calculated for and appended to each transport block (TB). The CRC allows for Rx error detection in the decoded TB. The corresponding error indication is then used, for instance, by a hybrid automatic retransmission request

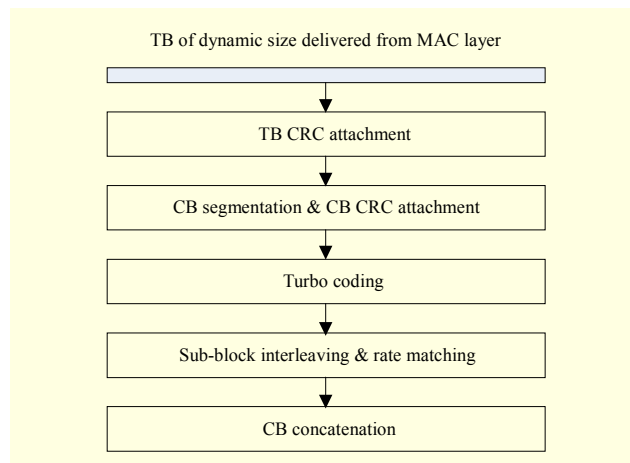


Fig. 2. Data channel coding scheme for LTE systems.

(HARQ) protocol as a trigger for requesting retransmissions. The LTE turbo coder internal interleaver is only defined for a limited number of CB sizes with a maximum block size of 6,144 bits. If the TB, including the TB CRC, exceeds this maximum CB size, the CB segmentation, as shown in Fig. 2, is applied before turbo coding. The CB segmentation implies that the TB is segmented into smaller CBs that match the set of CB sizes defined for the turbo coder. The CB segmentation also implies that an additional (24-bit) CRC is calculated for and appended to each CB. Having a CRC per CB allows for early detection of correctly decoded CBs and corresponding early termination of the iterative decoding of the CB. This can be used to reduce the terminal processing effort and power consumption. It should be noted that in the case of no CB segmentation, that is, in case of a single CB, no extra CB CRC is applied.

Turbo encoding reuses two wideband-code-division-multiple-access/high-speed downlink packet access (WCDMA/HSPA) eight-state constituent encoders which operate at a 1/2 rate, implying an overall code rate of 1/3. However, the WCDMA/HSPA turbo encoder internal interleaver has been replaced for LTE systems by quadratic permutation polynomial (QPP)-based interleaving [14]. The QPP interleaver provides a mapping from the non-interleaved input bits to the interleaved output bits according to

$$c(i) = (f_1 \cdot i + f_2 \cdot i^2) \bmod K, \quad (1)$$

where i is the index of the bit at the output of the interleaver, $c(i)$ is the index of the same bit at the input of the interleaver, and K is the CB size. The values of parameters f_1 and f_2 depend on the CB size K . The LTE specification [7] lists all supported CB sizes, ranging from a minimum of 40 bits to a maximum of 6,144 bits, together with the associated values for parameters f_1 and f_2 . In contrast to the WCDMA/HSPA turbo code interleaver, a QPP-based interleaver for the CB is maximum

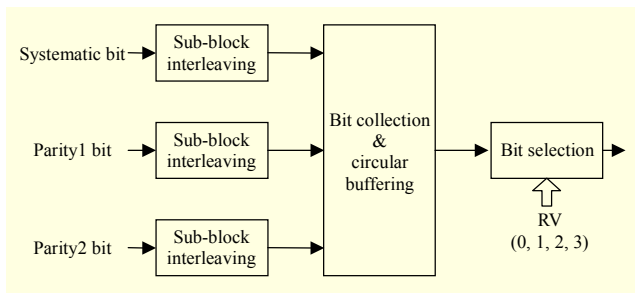


Fig. 3. Sub-block interleaving and rate matching.

contention free [15], implying that the decoding can be straightforwardly parallelized without risk of contention when the different parallel processes are accessing the interleaver memory. For very high data rates to be supported by LTE systems, the improved possibilities for parallel processing offered by QPP-based interleaving will substantially simplify the turbo coder implementation.

The task of rate matching and physical layer HARQ functionality is to extract from the blocks of code bits delivered by the channel encoder the exact set of bits to be transmitted within a given transmission time interval (TTI). As illustrated in Fig. 3, the outputs of the turbo encoder (systematic, parity1, and parity2 bits) are first separately interleaved by the sub-block interleaving. The interleaved bits are then inserted into what can be described as a circular buffer with the systematic bits inserted first, followed by alternating the insertion of parity1 and parity2 bits.

The bit selection then extracts consecutive bits from the circular buffer to an extent that fits the assigned resource. The set of bits to extract depends on the redundancy version (RV) corresponding to different starting points for the extraction of coded bits from the circular buffer. There are four different alternatives for the RV as shown in Fig. 3. At the last step of the transport channel processing, the rate matched CBs are successively connected to each other through the CB concatenation.

In the encoding scheme of a data channel for LTE systems, the unit of input data for the CRC attachment and CB segmentation is a TB, while that of the turbo coding and rate matching is a CB. Also, the data dependency does not exist among the CBs and a QPP-based interleaver for the CB is maximum contention free, so that it is possible to process the turbo codings and rate matchings for each CB in a parallel behavior that will be discussed later.

III. Proposed Parallel Structure for an Encoder

Figure 4 shows the proposed parallel structure for an encoder based on LTE systems. The encoder is composed of a TB CRC

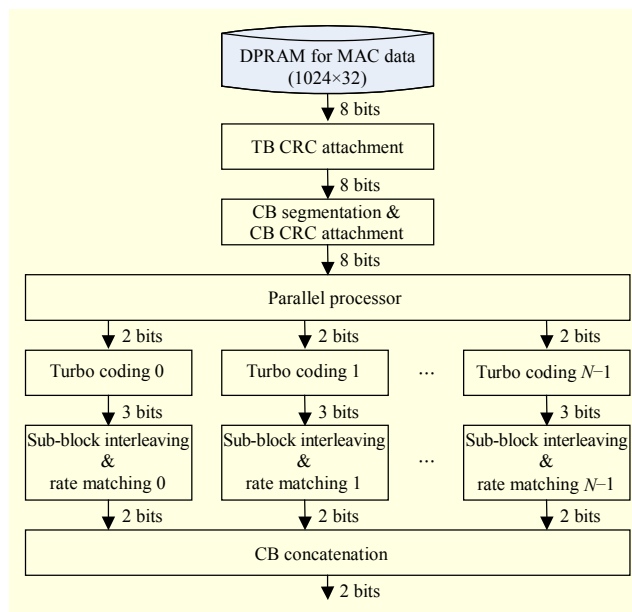


Fig. 4. Proposed parallel structure for an encoder ($N = \text{no. of CBs}$).

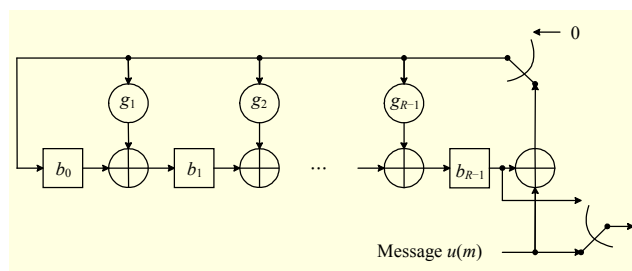


Fig. 5. Conventional CRC attachment based on a serial form.

attachment, a CB segmentation and CB CRC attachment, a parallel processor, N turbo codings, N sub-block interleavings and rate matchings, and a CB concatenation.

To maximize the parallelism of each turbo coding and rate matching, the CRC attachment and CB segmentation should deal with the multiple bits. Consequently, we improve them to handle 8-bit data per clock cycle. The reason for choosing 8 bits is that the TB and CB sizes are a multiple of 8 [7] and this makes the implementation easier. It is possible to extend the data bit width to further increase the parallelism. However, the method is somewhat complex in realization and requires more area overhead such as FPGA slices and memory.

First, we employ the matrix multiplication scheme [16] to enhance the CRC attachment. Figure 5 shows the conventional structure of a serial-based CRC attachment with the generator polynomial given as

$$g(X) = X^R + \sum_{r=0}^{R-1} g_r \cdot X^r, \quad R = N - K, \quad g_0 = 1, \quad (2)$$

where X is an indeterminate element.

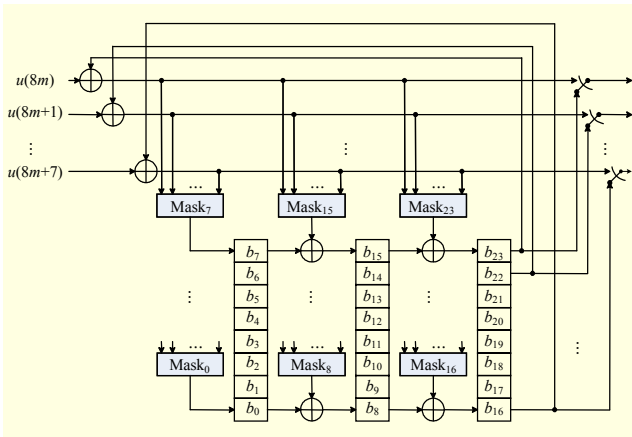


Fig. 6. Proposed CRC attachment based on 8-bit parallel processing.

The state vector of shift register $\mathbf{b}(m)=[b_{R-1}(m) \cdots b_0(m)]^T$ repeats the transition to $\mathbf{b}(m+1)$ by the m -th message bit $u(m)$ at time index $m (=0, 1, \dots, K-1)$, and the register generates the output values when the switching occurs after the K -cycle. From these operations and the matrix multiplication scheme, the formulation (3) is obtained as

$$\mathbf{b}(m+1) = \begin{bmatrix} g_{R-1} & 1 & & & \\ g_{R-2} & & 1 & & \\ \vdots & & & \ddots & \\ g_1 & & & & 1 \\ g_0 & 0 & 0 & \dots & 0 \end{bmatrix} \cdot \mathbf{b}(m) + \mathbf{g} \cdot u(m), \quad (3)$$

where \mathbf{g} is defined as $[g_{R-1} \cdots g_0]^T$ and the matrix operations are accomplished by multiplication and addition on $GF(2)$. Equation (4) is obtained by mathematical induction:

$$\mathbf{b}(m+S) = \mathbf{A}^S \cdot \mathbf{b}(m) + \begin{bmatrix} \mathbf{A}^{S-1} \cdot \mathbf{g} & \dots & \mathbf{A}^0 \cdot \mathbf{g} \end{bmatrix} \cdot \begin{bmatrix} u_0 \\ \vdots \\ u_{S-1} \end{bmatrix}, \quad (4)$$

where R -dimensional \mathbf{A} indicates the matrix multiplied with $\mathbf{b}(m)$ in the right-hand side of (3). Figure 6 shows a block diagram of the CRC attachment, where R and S are 24 and 8, respectively.

In Fig. 6, the mask blocks are generated by an inner-product between input message vectors and original binary vectors, and the original binary vectors are produced by the generator polynomials. The LTE specification [7] defines the generator polynomials of the CRC attachment for TB (24A) and CB (24B) as

- $g(x)^{24A} = x^{24} + x^{23} + x^{18} + x^{17} + x^{14} + x^{11} + x^{10} + x^7 + x^6 + x^5 + x^4 + x^3 + x + 1$,
- $g(x)^{24B} = x^{24} + x^{23} + x^6 + x^5 + x + 1$.

Table 1. Masking patterns for the generator polynomials $g(x)^{24A}$ and $g(x)^{24B}$ (8-bit hexadecimal representations).

Mask id	$g(x)^{24A}$	$g(x)^{24B}$	Mask id	$g(x)^{24A}$	$g(x)^{24B}$
23	f8	ff	11	88	04
22	08	00	10	e9	08
21	11	00	9	2b	10
20	23	00	8	57	21
19	46	00	7	af	42
18	8c	00	6	a6	84
17	e1	00	5	b5	f7
16	3b	00	4	92	10
15	76	00	3	dc	20
14	ed	00	2	41	40
13	22	01	1	82	80
12	44	02	0	fc	ff

Table 1 illustrates the masking patterns based on the generator polynomials $g(x)^{24A}$ and $g(x)^{24B}$.

The CRC attachment, processing 8-bit data per clock cycle, is implemented by the aforementioned scheme.

The CB segmentation includes a QPP-based internal interleaver as well as a CRC attachment for the CB. Therefore, the internal interleaver also has to process an 8-bit data per clock cycle. A QPP-based interleaver for the CB is maximum contention free, implying that the decoding can be straightforwardly parallelized without risk of contention when the different parallel processes are accessing the interleaver memory. Therefore, it is possible to process the internal interleaver in parallel behavior. Equation (5) for 8-bit parallel processing is simply obtained from (1).

$$c(8m+i) = \{f_1 \cdot (8m+i) + f_2 \cdot (8m+i)^2\} \bmod K, \quad (5)$$

where $m \in \{0, 1, \dots, K/8-1\}$ and $i \in \{0, 1, \dots, 7\}$. The message-bit-attached CRC for each CB should be stored in buffers before applying it to the corresponding turbo coding due to the generation of the interleaved bits. To utilize this buffering latency more efficiently, both the generation of the interleaving sequences and the store of the message-bit-attached CRC are performed at the same time, so the interleaved bits are prepared simultaneously with the message bits. These operations are executed in a parallel processor. Figure 7 illustrates the internal structure of the parallel processor and the generation process of the message and interleaved bits for a CB.

The parallel processor consists of a de-multiplexer to route the data path for each turbo coding and the memory to store the message and interleaved bits. The memory is based on

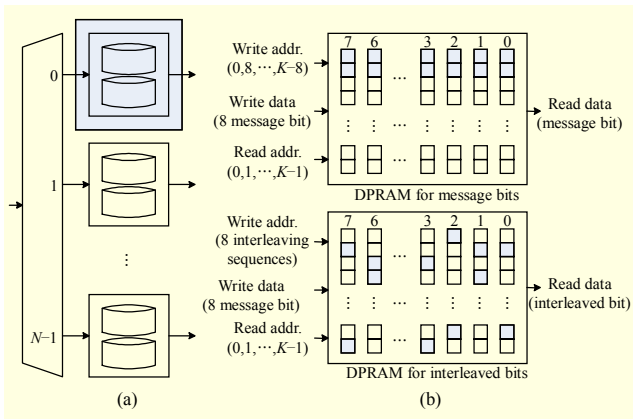


Fig. 7. (a) Internal structure of the parallel processor ($N = \text{no. of CBs}$) and (b) generation process of the message and interleaved bits for a CB.

8×768 matrix forms, and this formation makes us simply write an 8-bit data per clock cycle. The write addresses of DPRAM for message bits are generated in serial order such as 0, 8, 16, ..., $K-8$. The read addresses of DPRAM for message bits are generated in serial order such as 0, 1, 2, ..., $K-1$. The write addresses of DPRAM for interleaved bits are created by 8 interleaving sequences. The read addresses of DPRAM for interleaved bits are created in serial order, such as 0, 1, 2, ..., $K-1$.

The addresses of two DPRAMs consist of 13 bits. The upper 10 bits indicate the row address ranging from 0 to 767 of the matrixes, and the lower 3 bits indicate the column address ranging from 0 to 7 of the matrixes. The write data for the two DPRAMs are equal to each other in the form of 8 message bits, while their read data are classified into two types: message and interleaved bits. The read data of two DPRAMs are concurrently applied to the corresponding turbo coding after the completion of the write operations of the input message bits.

By virtue of 8-bit parallel processing of the CRC attachment, CB segmentation, and parallel processor, the turbo codings and rate matchings for each CB are performed in a parallel fashion. Figure 8 shows the internal structure and interface of the turbo codings, rate matchings, and CB concatenation. The input data of the turbo coding for a CB has 2 bits as message and interleaved bits, and the input data of the rate matching for a CB has 3 bits as systematic, parity1, and parity2 bits. It is possible to implement the turbo coding and rate matching in a parallel fashion such as a 2-bit per clock cycle or more. However, the method is complex in implementation and requires more area overhead such as FPGA slices and memory. Also, we do not feel it is necessary to employ turbo coding and rate matching based on the parallel processing because the

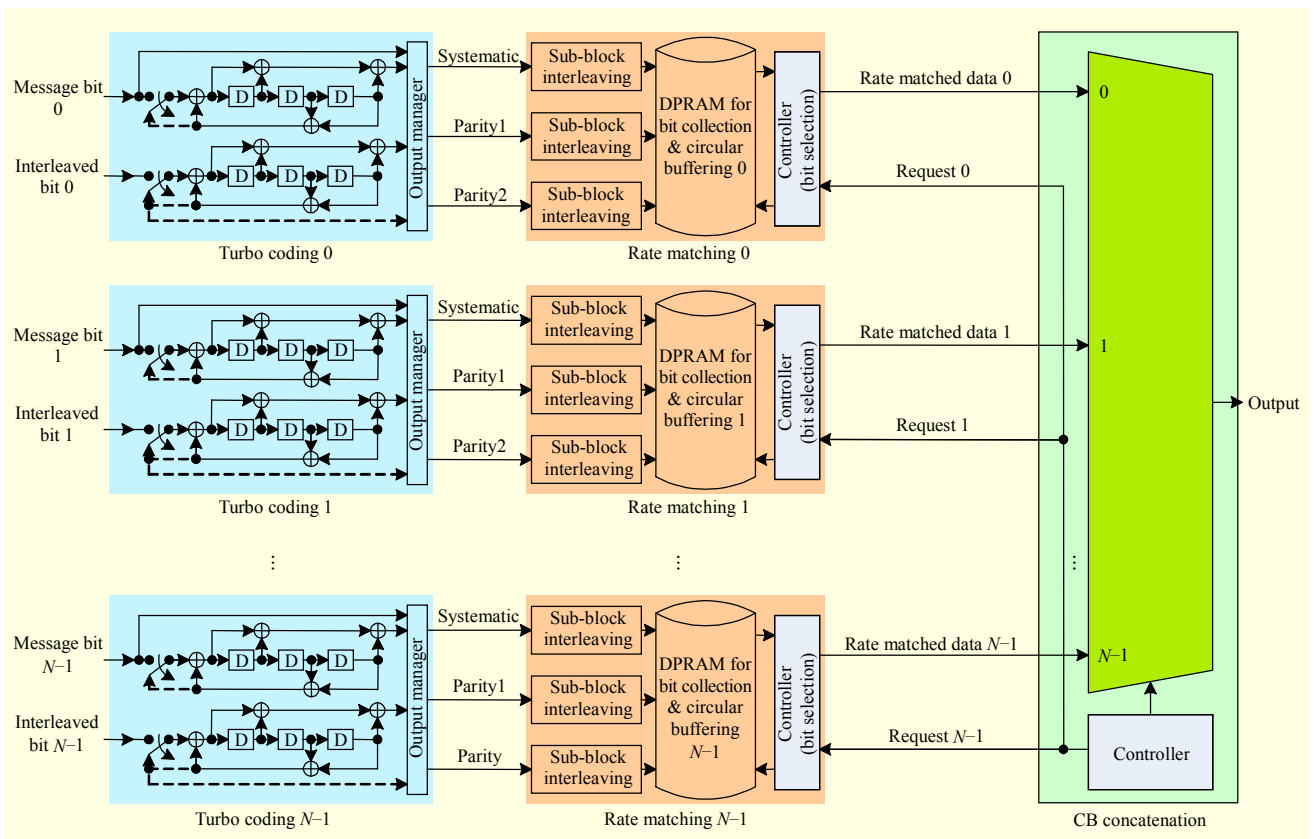


Fig. 8. Internal structure and interface of the turbo codings, rate matchings, and CB concatenation ($N = \text{no. of CBs}$).

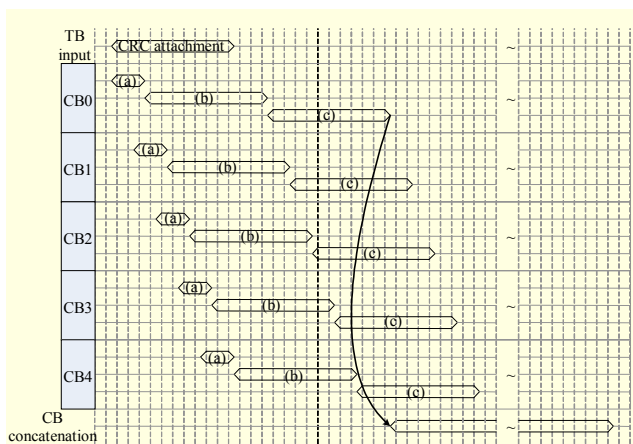


Fig. 9. Example timing diagram of an encoder based on 8-bit parallel processing: (a) CB segmentation including CRC attachment and internal interleaver, (b) turbo coding, and (c) rate matching.

turbo codings and rate matchings for each CB are performed in a parallel fashion due to the parallel processing of the CRC attachment, CB segmentation, and parallel processor. As a result, the latency of an encoder is reduced without additional area overhead.

The output data of the rate matching is 2 bits and is applied to the CB concatenation. The CB concatenation is responsible for combining the rate matched data for each CB in serial order from 0 to $N-1$, and this function is carried out by the controller and multiplexer shown in Fig. 8.

In particular, the handshake protocol based on request and response signals is employed between each rate matching and CB concatenation to control the data flow more efficiently. The controller in the CB concatenation requests the output of the rate-matched data stored in dedicated DPRAM for bit collection and circular buffering to controllers in the rate matching without additional idle cycles among CBs, and the receivers offer 2-bit rate matched data based on bit selection from DPRAM to the sender as a response. Hence, the output of the CB concatenation is successively generated without extra idle cycles among CBs.

Figure 9 illustrates an example timing diagram focusing on the data flow of an encoder based on 8-bit parallel processing with 5 CBs. The CB segmentations for each CB are done in a serial manner, as shown in Fig. 9(a), and the turbo codings and rate matchings, shown in Figs. 9(b) and 9(c), respectively, for each CB are achieved in a parallel fashion. In addition, the CB concatenation can simply start the output of an encoder after the completion of the rate matching for the first CB because the rate-matched data for the next CBs are prepared by the parallel processing among the CBs, so there are no extra idle cycles in the output of the CB concatenation.

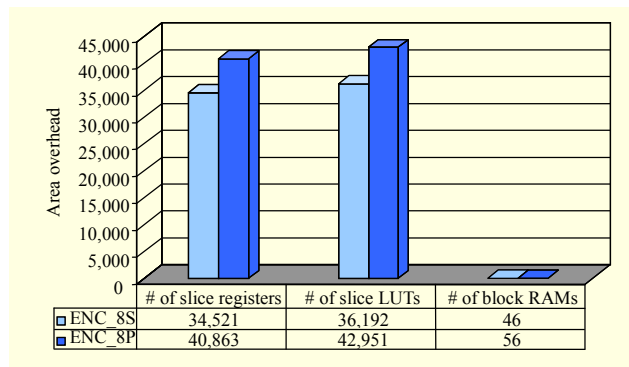


Fig. 10. Comparison of the area overhead.

Table 2. Comparison of the clock period.

	ENC_8S	ENC_8P
Clock period (ns)	5.253	5.588

IV. Experimental Results

1. Implementation

We implement the data channel encoders supporting up to 5 CBs for LTE systems. The encoders are implemented using a synthesizable RTL Verilog targeting XILINX FPGA (XC5VSX240T-1ff1738), and the XILINX design tool (ISE 10.1i) is used to measure the total area and clock period. The implemented encoders are

- an encoder based on 8-bit parallel processing (abbreviated as ENC_8P), and
- an encoder based on 8-bit serial processing (abbreviated as ENC_8S).

We implement the ENC_8S for comparison with the proposed ENC_8P in area overhead and latency. The ENC_8S consists of an 8-bit-based CRC attachment, an 8-bit-based CB segmentation, turbo coding, rate matching, and CB concatenation. We also except the encoder based on 1-bit serial processing (abbreviated as ENC_1S) for the experiments because the latency of ENC_1S is very long, making it unable to meet the timing requirements of LTE systems. In practical terms, the operation time of ENC_1S exceeds the TTI boundary (1 ms) for processing 5 CBs. Figure 10 and Table 2 show the synthesis results of the ENC_8P and ENC_8S.

It is apparent that the slice register, slice LUT, block RAM, and clock period of the ENC_8P are 18%, 19%, 22%, and 6% larger than those of the ENC_8S, respectively. The reason for this is that the ENC_8P requires four more engines for turbo coding and rate matching to handle the parallel processing among the CBs than the ENC_8S. However, we consider this to be non-critical since our engine for both turbo coding and

Table 3. Simulation scenarios for performance analysis.

	Type1	Type2	Type3	Type4	Type5
Minimum TB size	16	6,200	12,384	18,568	24,496
No. of CBs	1	2	3	4	5
No. of all supported CB sizes	188	48	32	24	20

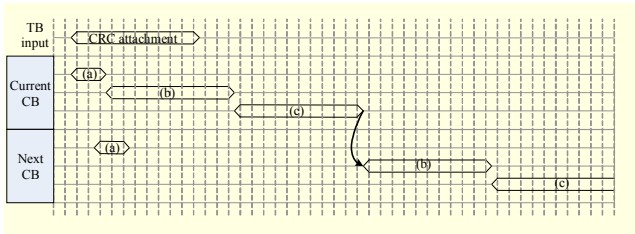


Fig. 11. Example timing diagram of an ENC_8S: (a) CB segmentation, (b) turbo coding, and (c) rate matching.

rate matching occupies a very small fraction of the total FPGA chip area. Actually, an engine takes about 1% of the total area in case of XC5VSX240T. Therefore, it is possible to support more CBs by using more engines for turbo coding and rate matching with a small rate of increase of the total FPGA chip area.

2. Performance Analysis

A Modelsim II simulator is utilized to measure the latency of the encoders. The clock frequency for simulations is 61.44 MHz, which is equivalent to that of an FPGA platform for evaluation. Table 3 charts the simulation scenarios for performance analysis. The LTE specification [7] prescribes all supported CB sizes. The total number of CB sizes is 188, and the CB sizes increase from a minimum of 40 bits to a maximum of 6,144 bits at irregular differences. Thus, we perform the performance simulations in all possible CB sizes according to the number of CBs.

The CB concatenation of the ENC_8P can initiate the output of the encoder after the completion of the rate matching of the first CB, while that of the ENC_8S cannot start the output of the encoder after the completion of the rate matching of the first CB since the turbo coding for the next CB can only perform the operation after the completion of the rate matching for the current CB, as shown in Fig. 11.

Therefore, the rate matched data of each CB are generated in a serial manner with an additional idle cycle among the CBs, which leads to a delay of the start point of the CB concatenation due to consecutive generation of the output data, as shown in Fig. 12. In the figure, the start points of the CB concatenations are T3, T4, T5, T6, and T7 when the number of CBs is 1, 2, 3, 4, and 5, respectively.

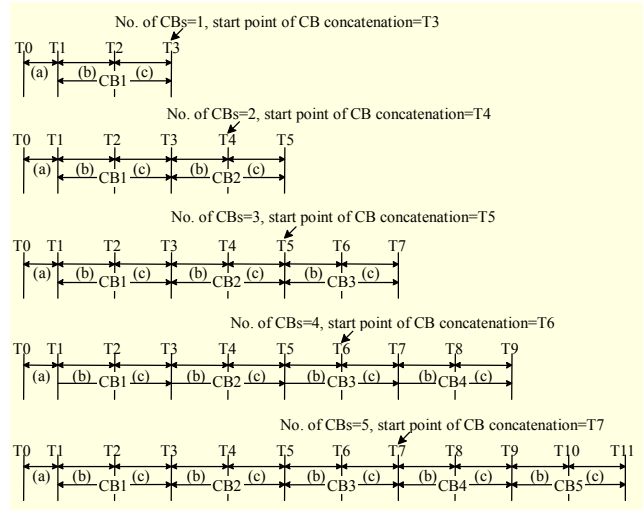


Fig. 12. Start points of the CB concatenations for an ENC_8S: (a) CB segmentation of the first CB, (b) turbo coding of each CB, and (c) rate matching of each CB.

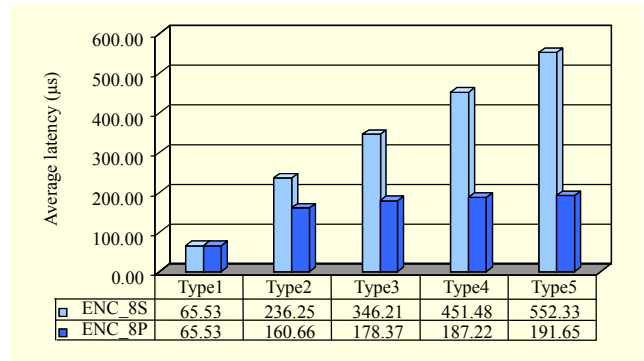


Fig. 13. Comparison of the ENC_8P with the ENC_8S in average latency.

As a result, the latency of the ENC_8S is longer than that of the ENC_8P, and we observe the simulation results shown in Fig. 13.

In case type1, where only one CB exists, the latencies for the ENC_8S and ENC_8P are equal to each other since the parallelism among the CBs does not take place and an engine is enough to process all TB input, while the latencies of the ENC_8P are decreased by 32%, 48%, 59%, and 65% compared with those of the ENC_8S in cases of type2, type3, type4, and type5, where there are many CBs, respectively. From Fig. 13, it is evident that ENC_8P is more latency efficient as the number of CBs grows, which means that the parallelism among the CBs increases.

Through the performance simulations, the latencies of the ENC_8P and ENC_8S are obtained as

$$\frac{((K + 24)/8 + (K + 4) * 2)}{clock_freq}, \quad (6)$$

$$\frac{((K + 24)/8 + (K + 4) * (C + 1))}{clock_freq.}, \quad (7)$$

respectively, where K is the CB size, C indicates the number of CBs, and $clock_freq.$ denotes the clock frequency (61.44 MHz). In addition, the element “+24” in “ $K+24$ ” means the CRC bits for CB segmentation and the one “+4” in “ $K+4$ ” stands for the tail bits for turbo coding. Therefore, the performance improvement (in a percentage) of the ENC_8P over the ENC_8S is formulated as

$$\frac{(K + 4) * (C - 1)}{((K + 24)/8 + (K + 4) * (C + 1))} * 100. \quad (8)$$

3. Evaluation

We apply our scheme to a UE baseband transport channel encoder unit (UCEU) for an IMT-advanced new mobile access (NeMA) system based on LTE to show the efficiency of the proposed scheme.

The UCEU is included in the uplink path and consists of PUSCH and PUCCH encoders. The PUSCH (PUCCH) encoder is composed of a UL-SCH encoder, a CQI encoder, an ACK encoder, a RI encoder, and a channel interleaver (a CQI encoder and an ACK encoder). Among them, the UL-SCH encoder has the longest latency since that is a data channel encoder. Therefore, we apply the proposed scheme to a UL-SCH encoder. The UCEU supports two code words (CWs), and the maximum TB size and CB number are 30,576 and 5, respectively, so that the UCEU has a 30 Mbps peak data rate per CW. The device for FPGA targeting is a XC5VSX240T-1ff1738, and the clock frequency is 61.44 MHz. Figure 14 depicts the UE platform for a NeMA system. In the figure, the units labeled (a), (b), (e), (f), and (g) are included in the downlink path, while those labeled (h) and (i) are contained in the uplink path. In addition, the unit labeled (c) controls the external interfaces of the platform, and the one labeled (d) calculates the complex computations and sets the register parameters for each unit. The UCEU operates in the FPGA (i) of the figure.

Figure 15 shows the synthesis results based on FPGA chip utilization of the UCEU. The UCEU occupies 54% (82,213/149,760) of the slice register, 56% (85,300/149,760) of the slice LUT, 31% (165/516) of the block RAM, and 23% (245/1,056) of the multiplier. In addition, the measured clock frequency from the place and route tool of XILINX is 98.21 MHz, so the UCEU meets the area requirement of the NeMA platform.

To meet the ACK/NACK timing requirement, and upon detection of a downlink shared data transmission in TTI $n-4$

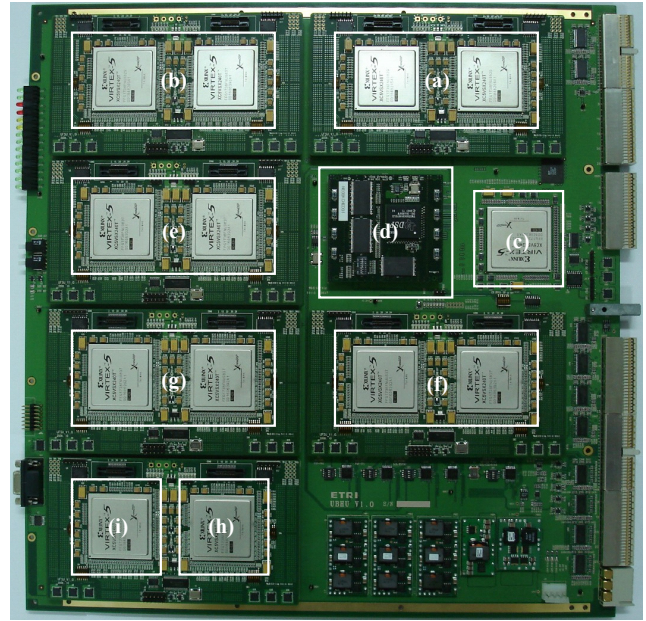


Fig. 14. UE platform for NeMA system: (a) synchronizer, (b) demodulator, (c) controller, (d) DSP, (e) MIMO detector, (f) turbo decoder, (g) viterbi decoder and derate matcher, (h) modulator, and (i) encoder.

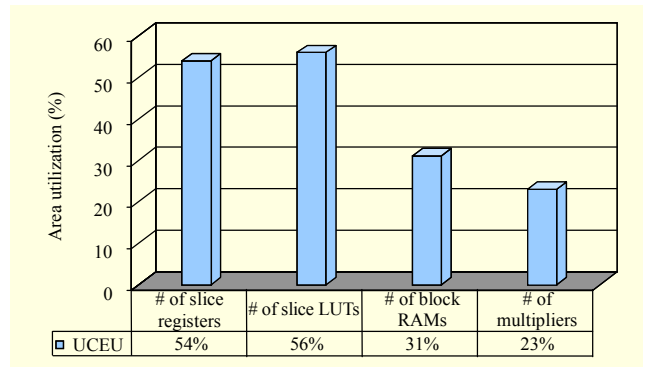


Fig. 15. FPGA chip utilization of the UCEU.

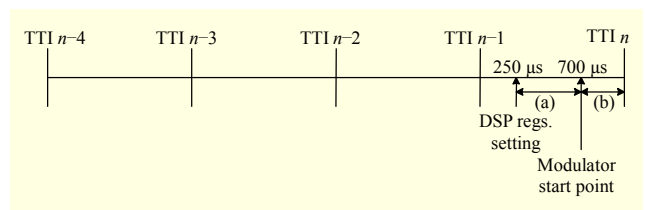


Fig. 16. ACK/NACK timing requirement for LTE systems: (a) 450 μ s and (b) 300 μ s.

intended for the UE, for which an ACK/NACK shall be provided, the UE shall transmit the ACK/NACK response in TTI n [8] through the uplink Tx link, as shown in Fig. 16.

The DSP configures the register parameters of the UCEU at 250 μ s of every TTI, and the UCEU starts the operation at the same time. The latency of the modulator is 300 μ s, so the

UCEU has to complete all operations and data transmissions to the modulator during 450 μ s. If the conventional data channel encoder based on a serial structure is applied to the UCEU, it may not satisfy the timing requirement of 450 μ s. In particular, if the number of CBs increases to over four, it will be difficult to meet the timing requirement. However, the UCEU employing the proposed scheme is satisfied with the timing requirements in all possible cases. Actually, the latencies of the UCEU take under 400 μ s in all scenarios of an NeMA system.

V. Conclusion

In this paper, we have presented one way to improve the structure of a data channel encoder to decrease the latency. With an 8-bit parallel processing of the CRC attachment, CB segmentation, and parallel processor, we are able to carry out the engines for turbo codings and rate matchings of each CB in a parallel manner. Experimental results show that although the total area and clock period of the proposed scheme are 19% and 6% larger than those of the conventional method based on serial type, respectively, our parallel structure decreases the latency by about 32% to 65% compared with a serial structure. In particular, our scheme is more latency-efficient when the encoder handles numerous CBs. Moreover, we apply our scheme to a NeMA system based on LTE, so that the timing requirement for ACK/NACK transmission is satisfied by employing the encoder based on the proposed scheme. We expect that it will be beneficial to apply our approach to high-speed data communication systems that require low latency.

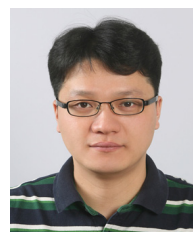
For future work, we feel that the power consumption according to the increment of the total area of the encoder with parallel structure needs to be measured. We are likewise looking at the applicability of the proposed scheme to the decoder corresponding to the encoder.

References

- [1] Y. Zhao, "Standardization of Mobile Phone Positioning for 3G Systems," *IEEE Comm. Mag.*, vol. 40, no. 7, July 2002, pp. 108-116.
- [2] D. Molkdar et al., "An Overview of EGPRS: The Packet Data Component of EDGE," *Elec. & Comm. Eng. J.*, vol. 14, no. 1, Feb. 2002, pp. 21-38.
- [3] W. Kamer et al., "Link Error Analysis and Modeling for Video Streaming Cross-Layer Design in Mobile Communication Networks," *ETRI J.*, vol. 29, no. 5, Oct. 2007, pp. 569-595.
- [4] J. Peisa et al., "High Speed Packet Access Evolution—Concept and Technologies," *Proc. IEEE Int. Conf. VTC Spring*, Apr. 2007, pp. 819-824.
- [5] J. Berkmann et al., "On 3G LTE Terminal Implementation—

Standard, Algorithms, Complexities and Challenges," *Proc. Int. Conf. WCMC*, Aug. 2008, pp. 970-975.

- [6] 3GPP TS 36.211, "Physical Channels and Modulation (Release 8)," 2008.
- [7] 3GPP TS 36.212, "Multiplexing and channel coding (Release 8)," 2008.
- [8] 3GPP TS 36.213, "Physical Layer Procedures (Release 8)," 2008.
- [9] 3GPP TR 25.913, "Requirements for Evolved UTRA (E-UTRA) and Evolved UTRAN (E-UTRAN) (Release 9)," 2009.
- [10] J. Liu et al., "Design and Analysis of LTE Physical Downlink Control Channel," *Proc. IEEE Int. Conf. VTC Spring*, 2009, pp. 1-5.
- [11] W.C. Jeong, J.M. Chung, and D. Liu, "Characteristic-Function-Based Analysis of MIMO Systems Applying Macroscopic Selection Diversity in Mobile Communications," *ETRI J.*, vol. 30, no. 3, June 2008, pp. 355-364.
- [12] C.D. Chung, "Correlatively Coded OFDM," *IEEE Trans. Wireless Commun.*, vol. 5, no. 8, Aug. 2006, pp. 2044-2049.
- [13] D.S. Cho et al., "Implementation of an Efficient UE Decoder for 3G LTE System," *Proc. Int. Conf. Telecommun.*, June 2008, pp. 1-5.
- [14] J. Sun and O.Y. Takeshita, "Interleavers for Turbo Codes Using Permutation Polynomials over Integer Rings," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, Jan. 2005, pp. 101-119.
- [15] O.Y. Takeshita, "On Maximum Contention-Free Interleavers and Permutation Polynomials over Integer Rings," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, Mar. 2006, pp. 1249-1253.
- [16] S.Y. Hwang et al., "An Improved Implementation Method of the Gold Sequence Generator," *Proc. IEEE Int. Conf. Consumer Electron.*, Apr. 2008, pp. 1-4.



Soo Yun Hwang received the BS in computer engineering from Hannam University, Daejeon, Korea, in 2002, and his MS and PhD in computer engineering from Chungnam National University, Daejeon, Korea, in 2004 and 2008, respectively. Since 2006, he has been working with ETRI, Daejeon, Korea, where he currently works in a high-speed modem research team as a senior member of engineering staff. He has participated in various projects including the development of a flexible on-chip network based system on a chip platform targeting H.264 decoder, a user equipment modem apparatus based on the 3rd Generation Partnership Project Long Term Evolution, and the adaptive radio access and transmission technologies for the 4th Generation Mobile Communications. His current research interests include CAD for VLSI, system on a chip design methodology, on-chip communication architecture, and high-speed modem designs.



Dae Ho Kim received his BS and MS in electronics engineering from Kyongpook National University, Daegu, Korea, in 1989 and 1991, respectively, and the PhD in electronics engineering from Chungnam National University, Daejeon, Korea, in 2005. He is currently a team leader of the high-speed modem research team at ETRI, Daejeon, Korea. His research interests include broadband wireless communications, DSP and VLSI applications, and multimedia signal processing.



Kyoung Son Jhang graduated from Seoul National University in 1986 with a Bachelor of Computer Engineering degree and received his MS and PhD at the same university in 1988 and 1995, respectively. Upon graduation, he joined Hannam University at Daejeon, Korea, as a faculty member in 1996. He then moved to Chungnam National University where he still works as a professor teaching systems programming and digital hardware design. Currently, his major interests include fault-tolerant hardware designs, electronic design automation, and digital system design.