

Design and Architecture of Low-Latency High-Speed Turbo Decoders

Ji Won Jung, In Ki Lee, Duk Gun Choi, Jin Hee Jeong,
Ki Man Kim, Eun-A Choi, and Deock Gil Oh

In this paper, we propose and present implementation results of a high-speed turbo decoding algorithm. The latency caused by (de)interleaving and iterative decoding in a conventional maximum a posteriori turbo decoder can be dramatically reduced with the proposed design. The source of the latency reduction is from the combination of the radix-4, center to top, parallel decoding, and early-stop algorithms. This reduced latency enables the use of the turbo decoder as a forward error correction scheme in real-time wireless communication services. The proposed scheme results in a slight degradation in bit error rate performance for large block sizes because the effective interleaver size in a radix-4 implementation is reduced to half, relative to the conventional method. To prove the latency reduction, we implemented the proposed scheme on a field-programmable gate array and compared its decoding speed with that of a conventional decoder. The results show an improvement of at least five fold for a single iteration of turbo decoding.

Keywords: Turbo code, radix-4, center-to-top, parallel decoding, early-stop, FPGA.

Manuscript received Jan. 31, 2005; revised June 9, 2005.

The material in this work was presented in part at IT-SoC 2004, Seoul, Korea, Oct. 2004.

Ji Won Jung (phone: +82 51 410 4424, email: jwjung@mail.hhu.ac.kr), In Ki Lee (email: popularity1@bada.hhu.ac.kr), Duk Gun Choi (email: dkchoi@baba.hhu.ac.kr), Jin Hee Jeong (email: jhlovetalk@hotmail.com), and Ki Man Kim (email: kimkim@mail.hhu.ac.kr) are with the Department of Radio and Information Communication Engineering, Korea Maritime University, Busan, Korea.

Eun-A Choi (email: eachoi@etri.re.kr) and Deock Gil Oh (email: dgoh@etri.re.kr) are with Digital Broadcasting Division, ETRI, Daejeon, Korea.

I. Introduction

For wireless communications, channel coding is an important tool for improving communications reliability. In satellite communications and 3rd generation mobile communication system, turbo codes have been adopted for high-speed data transmissions. Turbo codes, introduced in [1] and [2], have been shown to perform near the capacity limit in additive white Gaussian noise (AWGN) channels. Such a powerful coding technique offers great promise for improving the reliability of communication over wireless channels. Wireless communication systems are moving today from the conventional narrow-band voice service to the wide-band multimedia service requiring high-speed turbo decoders. Important issues in high-speed applications of turbo decoders are decoding delay and computational complexity. To solve the latency problem, in this paper four decoding algorithms are presented and combined into one decoder architecture. The first algorithm is the radix-4 decoding algorithm, where the previous state at $t=k-2$ goes forward to the current state at $t=k$, and the reverse state at $t=k+2$ goes backwards from the current one such that the time interval from $t=k-2$ to $t=k$ is merged into $t=k$. Therefore, we can decode two source data bits at the same time without any performance degradation while reducing the block size buffered in memory. In order to apply the radix-4 algorithm, we derived the branch metric (BM), forward state metric (FSM), α , backward state metric (BSM), β , and log likelihood ratio (LLR), λ .

The second algorithm is the center-to-top algorithm. In a conventional scheme, the decoder must wait for finishing the BSM (or FSM) calculations before calculating the LLR. The center-to-top method doesn't need to wait. The decoder

calculates the FSM (left to right) and BSM (right to left), simultaneously. When the FSM and BSM reach the same point, then the decoder begins to calculate the LLR.

The third algorithm is the full parallel decoding algorithm. Different from the original turbo decoder consisting of two decoders concatenated in a serial fashion, we propose a parallel decoder structure using the parallel sum, where the two decoders operate in parallel and update each other immediately and simultaneously after each one has completed its decoding. In decoding the estimated data, we use the sum of the LLR outputs of the parallel decoders to reduce the latency to half while maintaining the same performance level.

The fourth algorithm is the hard-decision-aided (HDA) algorithm implementing the early-stop algorithm. It compares each decision generated by the two decoders, and when the two sets of decisions match, it stops decoding the current block and outputs the hard decision bits.

These schemes are desirable in designing a high-speed turbo decoder because they provide significant reductions in decoder memory requirements as well as allowing increased parallelism. To verify the high-speed feature of the decoder, we implemented the proposed combined schemes in a field-programmable gate array (FPGA), Altera FLEX10K, and compared its decoding speed to a conventional design. The proposed architecture is demonstrated for a 4-state, $R = 1/2$ turbo maximum a posteriori (MAP) decoder.

The paper consists of the following sections. In section II, we present and analyze the four algorithms. Section III describes computer simulation results of the new high-speed turbo decoder based on combining these algorithms. In section IV, we present the optimal parameters and design of the FPGA-based high-speed turbo decoder. Section V concludes the paper.

II. High-Speed Decoding Algorithm

1. Scheme 1: Radix-4 Algorithm

Using the unified approach to state metrics, a 2^{v-1} -state trellis can be iterated from time index $n-k$ to n by decomposing the trellis into 2^{v-k} sub-trellises, each consisting of k iterations of a 2^k -state trellis. Each 2^k -state's sub-trellis can be collapsed into an equivalent one-stage radix- 2^k trellis by applying k levels of look-ahead to the recursive update. Collapsing the trellis does not affect the decoder performance since there is a one-to-one mapping between the collapsed trellis and radix-2 trellis. An example of the decomposition of a 4-state radix-2 into an equivalent radix-4 trellis using one stage of look-ahead is shown in Fig. 1, where $K=3$, $g_1=(111)$, $g_2=(101)$ with K denoting the constraint length.

For the radix-4 trellis iterations, time k to $k-2$, the backward

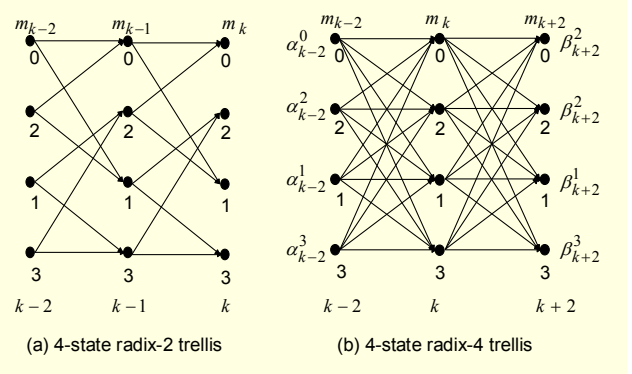


Fig. 1. Four-state radix-2 to radix-4 trellis.

state for calculating α_k^m , and time $k+2$ to time k , the forward state for calculating β_k^m , can be expressed as

$$\begin{aligned} m_{k-2}(d_{k-1}, d_k, m_k) &= (d_{k-1} \oplus d_k \oplus m_{0,k}) \parallel (d_k \oplus m_{0,k} \oplus m_{1,k}) \\ m_{k+2}(d_{k+1}, d_{k+2}, m_k) &= (d_{k+1} \oplus d_{k+2} \oplus m_{1,k}) \parallel (d_{k+1} \oplus m_{0,k} \oplus m_{1,k}), \end{aligned} \quad (1)$$

where d_k and m_k denote the uncoded data bit state values at time k , and $a \parallel b$ denotes the concatenation of a and b . We have that $m_k = \{m_{0,k}, m_{1,k}, \dots, m_{v-1,k}\}$ corresponds to the encoder state with m_{k-2} used for calculating α_k^m and m_{k+2} used for calculating β_k^m at time k .

A. Derivation of δ_k^m , α_k^m , β_k^m and λ_k^m

We consider an AWGN channel with zero mean and variance σ^2 . In a radix-4 MAP decoder, the branch metric $\delta_k^{p,m}$ for the received signal at time k , can be expressed as

$$\begin{aligned} \delta_k^{p,m} &= \Pr(D_k = p, S_k = m_k, R_k) \\ &= \Pr(D_k = i_{k-1} \parallel i_k, S_k = m_k, R_k) \\ &= \Pr(d_{k-1} = i_{k-1}, S_k = m_{k-1}, R_{k-1}) \\ &\quad \cdot \Pr(d_k = i_k, S_k = m_k, R_k) \\ &= \Pr(x_{k-1} | d_{k-1} = i_{k-1}, S_{k-1} = m_{k-1}) \\ &\quad \cdot \Pr(y_{k-1} | d_{k-1} = i_{k-1}, S_{k-1} = m_{k-1}) \zeta_{k-1}^{i_{k-1}} / 2^v \\ &\quad \cdot \Pr(x_k | d_k = i_k, S_k = m_k) \\ &\quad \cdot \Pr(y_k | d_k = i_k, S_k = m_k) \zeta_k^{i_k} / 2^v \\ &= K_k \exp\left(\frac{2}{\sigma^2} (x_{k-1} d_{k-1} + y_{k-1} Y_{k-1} + x_k d_k + y_k Y_k)\right), \end{aligned} \quad (2)$$

where i_k is the information bit value at time k , and $p = i_{k-1} \parallel i_k$. Therefore, we have $i = \{0, 1\}$ and $p = \{00, 01, 10, 11\}$; K_k is a constant, x_k and y_k are received in-phase and quadrature signals; and Y_k is the coded bit as a function of input bit d_k and encoder

state m_k . The forward state metric at time k and state m denoted by α_k^m can be expressed as

$$\begin{aligned} \alpha_k^m &= \Pr(R_1^{k-2} | D_k = p, S_k = m_k, R_k^N) = \Pr(R_1^{k-2} | S_k = m_k) \\ &= \sum_{m'} \sum_{p=0}^3 \Pr(D_{k-2} = p, S_{k-2} = m', R_1^{k-2} | S_k = m_k) \\ &= \sum_{m'} \sum_{p=0}^3 \Pr(D_{k-1} = p, S_{k-2} = m', R_1^{k-4}, R_{k-2} | S_k = m_k) \quad (3) \\ &= \sum_{p=0}^3 \alpha_{k-2}^{b(p, m_k)} \delta_{k-2}^{p, b(p, m_k)}, \end{aligned}$$

where $R_k^n = \{R_k, R_{k+1}, \dots, R_n\}$, and $b(p, m_k)$ is the previous $k-2$ time stage given input p and state m_k , that is,

$$b(p, m_k) = m_{k-2}(d_{k-1}, d_k, m_k). \quad (4)$$

In a similar way, after the whole block of data is received, we can recursively calculate probability β_k^m from probability β_{k+2}^m , that is,

$$\begin{aligned} \beta_k^m &= \Pr(R_k^N | S_k = m_k) \\ &= \sum_{m'} \sum_{p=0}^3 \Pr(D_k = p, S_{k+2} = m', R_k^N | S_k = m_k) \\ &= \sum_{m'} \sum_{p=0}^3 \Pr(D_k = p, S_{k+2} = m', R_{k+2}^N, R_k | S_k = m_k) \quad (5) \\ &= \sum_{p=0}^3 \beta_{k+2}^{f(p, m_k)} \delta_k^{p, m_k}, \end{aligned}$$

where $f(p, m_k)$ is the next $k+2$ time stage given input p and state m_k . With (1), $f(p, m_k)$ is expressed as

$$f(p, m_k) = m_{k+2}(d_{k+1}, d_{k+2}, m_k) \quad (6)$$

when taking into account the fact that events after time k are not influenced by observations up to time k , that is, R_1^k and R_{k+2}^N are independent. Finally, with (2), (5) and (6), the LLR of the radix-4 algorithm can be written as

$$\begin{aligned} \lambda_k^{p, m} &= \Pr(D_k = p, S_k = m | R_1^N) \\ &= \Pr(D_k = p, S_k = m, R_1^N) / \Pr(R_1^N) \\ &= \Pr(D_k = p, S_k = m, R_1^{k-2}, R_k^N) / \Pr(R_1^N) \\ &= \Pr(R_1^{k-2} | D_k = p, S_k = m, R_k^N) \Pr(R_{k+2}^N | D_k = p, S_k = m, R_k) \\ &\quad \cdot \Pr(D_k = p, S_k = m, R_k) / \Pr(R_1^N) \\ &= \alpha_k^m \beta_{k+2}^{f(p, m)} \delta_k^{p, m} / \Pr(R_1^N). \quad (7) \end{aligned}$$

In order to decode the two bits, $D_k = d_{k-1} \| d_k$, the following equation can be used:

$$D_k = \max \left\{ \sum_m \lambda_k^{00}(m), \sum_m \lambda_k^{01}(m), \sum_m \lambda_k^{10}(m), \sum_m \lambda_k^{11}(m) \right\}. \quad (8)$$

The overall configuration of the radix-4 turbo MAP decoder is depicted in Fig. 2. $L_c(x_j^{00}), L_c(x_j^{01}), L_c(x_j^{10}), L_c(x_j^{11})$ and $L_e(d_k^{00}), L_e(d_k^{01}), L_e(d_k^{10}), L_e(d_k^{11})$ denote the extrinsic information for correction at next MAP module. It is composed of two constituent decoders, MAP1 and MAP2, concatenated serially with symbol interleavers. Each MAP decoder calculates four LLRs, $\lambda_k^{00}, \lambda_k^{01}, \lambda_k^{10}$, and λ_k^{11} , where λ_k^{00} corresponds to the LLR value for two input bits '00'.

2. Scheme 2: Center-to-Top Algorithm

Another algorithm is the center-to-top (CT) algorithm. In

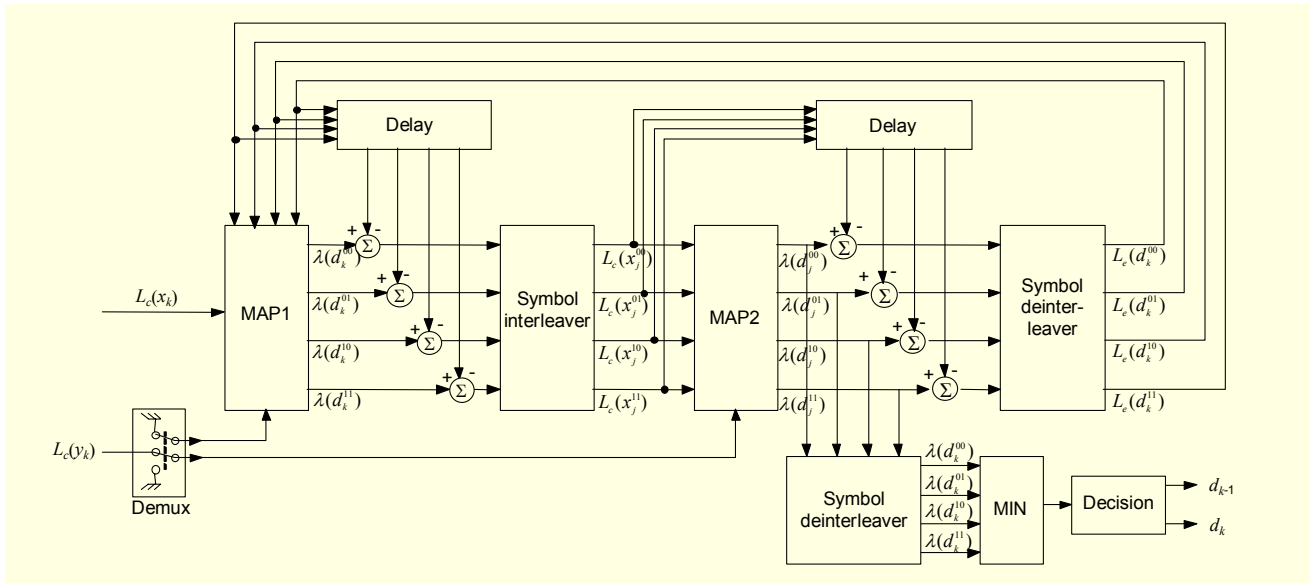


Fig. 2. Block diagram of turbo decoder based on radix-4.

conventional algorithms the decoder must wait for finishing the BSM (or FSM) calculations to calculate the LLR. The CT algorithm doesn't need to wait. The decoder calculates the FSM (left to right) and BSM (right to left), simultaneously. When the FSM and BSM reach the same point, the decoder begins to calculate the LLR. Figure 3 illustrates the operation of the CT algorithm. The procedure of the CT algorithm is as follows.

Step 1: Initialize the forward state metric and backward state metric.

$$\alpha_0^i(s_0^i(m)) = \begin{cases} 1, & \text{if } m = 0 \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

$$\beta_N^i(s_b^i(m)) = \begin{cases} 1, & \text{if } m = 0 \\ 0, & \text{otherwise} \end{cases}$$

Step 2: After receiving the whole set of received symbols, forward state metrics (left to right) and backward state metrics (right to left) are calculated simultaneously.

$$\hat{\alpha}_k^i(m) = \exp\left(\frac{2}{\sigma^2}(x_k i + y_k Y_k(i, m))\right) \sum_{j=0}^1 \hat{\alpha}_{k-1}^j(S_b^j(m)) \quad (10)$$

for $k = 0, \dots, N/2 - 1$,

$$\hat{\beta}_k^i(m) = \sum_{j=0}^1 \hat{\beta}_{k+1}^j(m) \exp\left(\frac{2}{\sigma^2}(x_{k+1} j + y_{k+1} Y_{k+1}(j, S_f^i(m)))\right) \quad (11)$$

for $k = N - 1, \dots, N/2$.

Step 3: At the middle point, begin to calculate the LLR.

$$L(d_k)_{right} = \log \frac{\sum_m \alpha_k^1(m) \beta_k^1(m)}{\sum_m \alpha_k^0(m) \beta_k^0(m)} \quad \text{for } k = N/2, \dots, N-1 \quad (12)$$

$$L(d_k)_{left} = \log \frac{\sum_m \alpha_k^1(m) \beta_k^1(m)}{\sum_m \alpha_k^0(m) \beta_k^0(m)} \quad \text{for } k = N/2 - 1, \dots, 0 \quad (13)$$

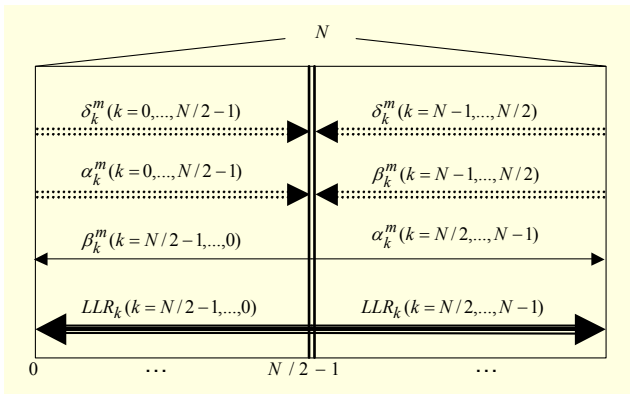


Fig. 3. CT algorithm.

3. Scheme 3: Parallel Decoding Algorithm [4]

The turbo decoding configuration operates in serial mode, that is, "MAP 1" processes data before "MAP 2" starts its operation, and so on. Another possibility, as presented in [4] and shown in Fig. 4 for two codes, is that all decoders operate in parallel at any given time.

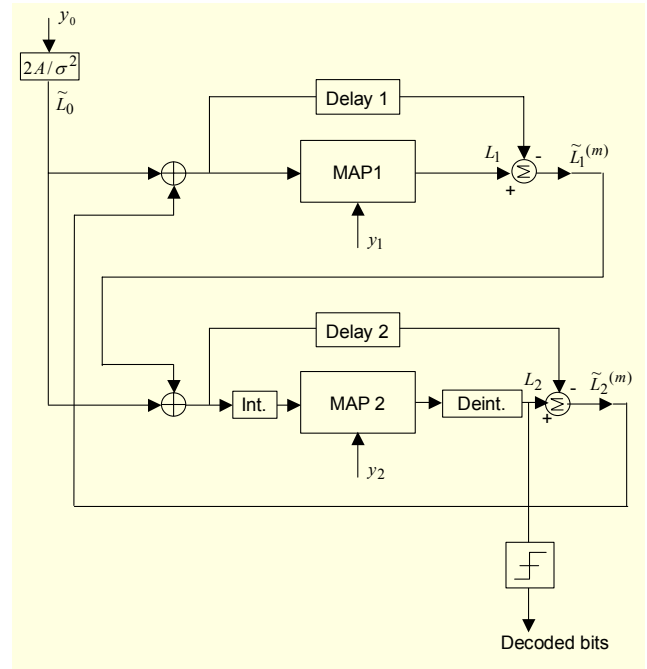


Fig. 4. Parallel decoder structure.

4. Scheme 4: Early Stop Algorithm

The hard decision (HAD) algorithm is used as an early-stop algorithm. It compares each decision generated by the two decoders, and when the two sets of decisions match, it stops decoding on the current block and outputs the hard decision bits. Table 1 shows the average number of iterations in an HAD algorithm. At an E_b/N_0 of 2 dB, it requires about 2.8 iterations relative to 8 for a given performance. This means that the decoding speed is improved or the power consumption (cost) is reduced by 64.6%.

Table 1. The average number of iterations according to E_b/N_0 (the predetermined number of iterations is 8).

E_b/N_0	Average number of iterations	Decoding speed improvement (%)
1	6.21	22.4
1.5	4.31	46.1
2	2.83	64.6

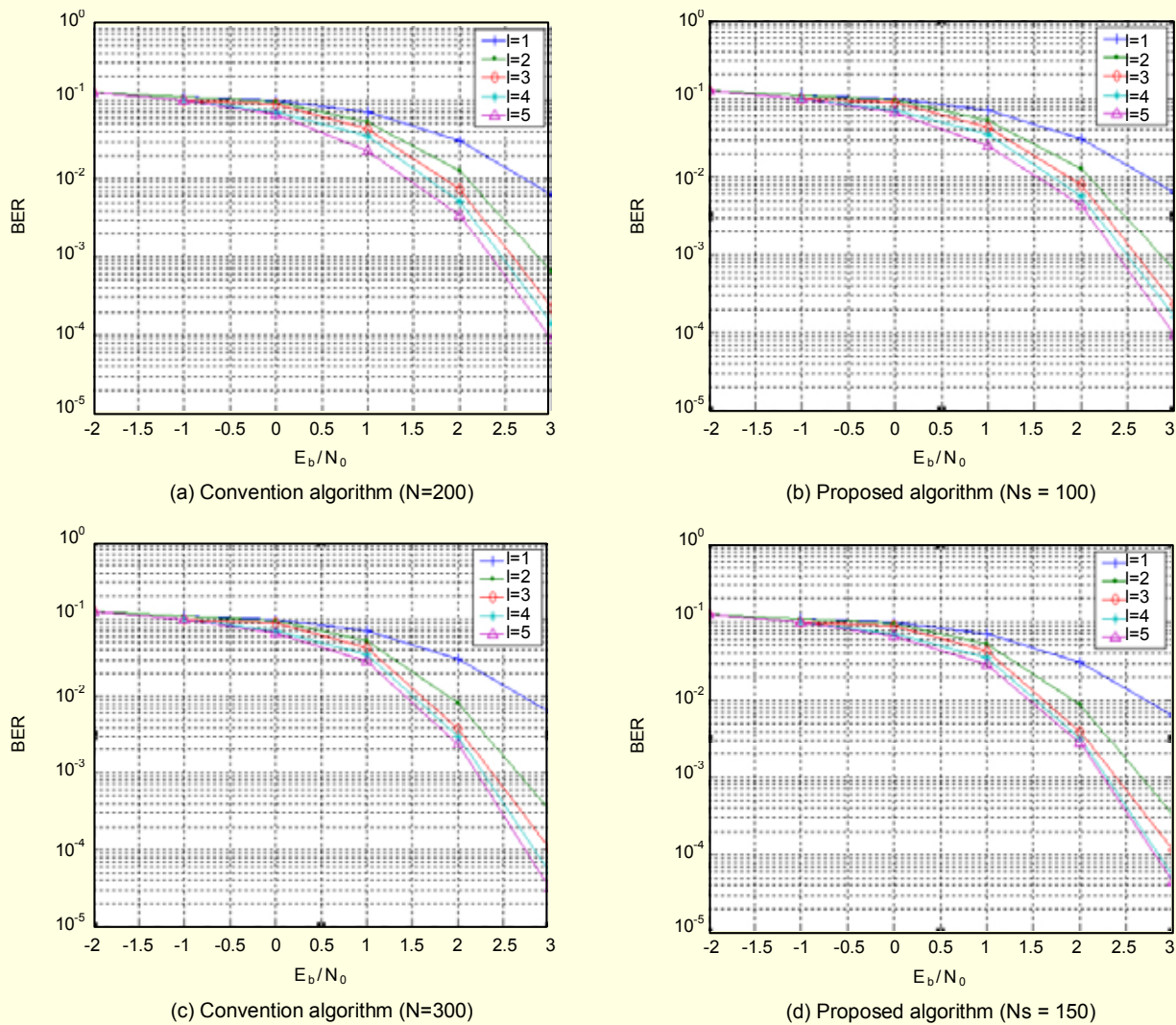


Fig. 5. Performance of the proposed decoder over an AWGN channel compared with that of a conventional algorithm as a function of interleaver size and number of iterations.

III. Computer Simulation Results

The bit-error rate (BER) performance of the new high-speed turbo decoder architecture combining the four schemes is analyzed in this section. For a comparison purpose, Fig. 5 shows the performance of the new decoder and a conventional one using $K=3$ turbo codes with generator polynomials $g_1=(111)$, $g_2=(101)$ as a function of interleaver size N and as a function of the number of iterations I . In the radix-4 method, the symbol interleaver, takes an information stream of length N_s composed of 2-bit words and feeds it to a random interleaver. From the figure, it can be verified that the performances of the proposed decoder architecture is very close to the conventional decoder for small block sizes (less than 300 bits). For large interleaver sizes (more than 300 bits),

the performance of the new decoder is slightly degraded relative to the conventional one because the randomness of the symbol interleaver is reduced. Therefore, it can be concluded that the proposed decoder reduces the decoding latency while maintaining almost the same performance of a conventional algorithm.

IV. Design of High-Speed Turbo Decoder

A schematic diagram of the high-speed turbo decoder implementation is shown in Fig. 6(a). A detailed block diagram of the radix-4 and CT decoder, MAP1 and MAP2, in this figure is shown in Fig. 6(b). In Fig. 6(a), MAP 1 and MAP 2 decoders are operated in parallel, with their outputs being log-likelihood ratios of input bits as shown in Fig. 6(b). For

instance, LLR_{00}^{\rightarrow} denotes log-likelihood ratios of input bits “00”, and the arrow \rightarrow denotes the direction of the LLR calculation, that is, left-to-right. To add the extrinsic information to the received symbols exactly in the next iteration, the decoder needs to re-order the blocks. This re-ordering of the blocks is equivalent to arranging the LLRs in a last-in, first-out or first-in, first-out fashion. As shown in Fig. 6(b), the decoder consists of four major units, the radix-4 branch metric unit (R4BMCU), radix-4 backwards state metric unit (R4BSMCU), radix-4 forwards state metric unit (R4FSMCU), and radix-4 for LLR calculator unit (R4LLRCU). The radix-4 algorithm is as described in subsection II.1 and needs four R4BMCUs, two R4FSMUs/ R4BSMUs, and four R4LLRCUs. After receiving the whole set of received symbols, the quantized I and Q samples are fed to the R4BMCU. The branch metrics between the branch codeword “0000” and the received symbols is denoted by $bm0000$. The R4BMCU calculates the branch metrics for four samples of received data. To apply the CT algorithm, the R4BMCU1 calculates the branch metrics in the direction of left-to-right. R4BMCU2 calculates the branch metrics in the direction of right to left. Each branch metric is buffered in memory to be used for R4BSMCU/R4FSMCU. In calculating the FSM (or BSM) in the interval from 0 to $N/2 - 1$ (or $N-1$ to $N/2$), we don’t need to calculate the branch metrics but simply refer to the memory of R4BMCU2 (or R4BMCU1). R4FSMCU, R4BSMCU, and R4LLRCU need an E-Table. The argument of the exponential function, that is, the input of an E-Table, was designed to be equal to the address lookup table (LUT). The E-Tables are implemented using an internal ROM (EAB, embedded array block) without using any external ROM.

In order to implement optimally the high-speed decoder, we determined the optimum quantized bits of each block shown in Fig. 6, the demodulator output, in-phase and quadrature signals, R4BMCU output of b_q , R4BSMCU output of s_q , and log-likelihood ratio l_q . By fixed-point computer simulation, the output of the demodulator is quantized to 8 bits. The internal parameters of the turbo decoder were always saturated to 9 bits, and the optimum quantization bits of the turbo decoder derived from the fixed-point simulations are listed in Table 2.

The results of the timing simulation of the high-speed turbo decoder are shown in Table 3. The interleaver size, N , was set to 64 bits due to the limited capacity of the embedded memory of Flex10k100GC503-4. The results from iterative decoding over an AWGN channel at $E_b/N_o=1$ dB is also presented in Fig. 7. Since the error correcting performance increases with the number of iterations, we concluded that the high-speed decoder implemented in this test-bed of Flex10k100GC503-4, pictured in Fig. 8, is performing well.

In order to compare the decoding speed of a conventional turbo decoder to the proposed one, we implemented a

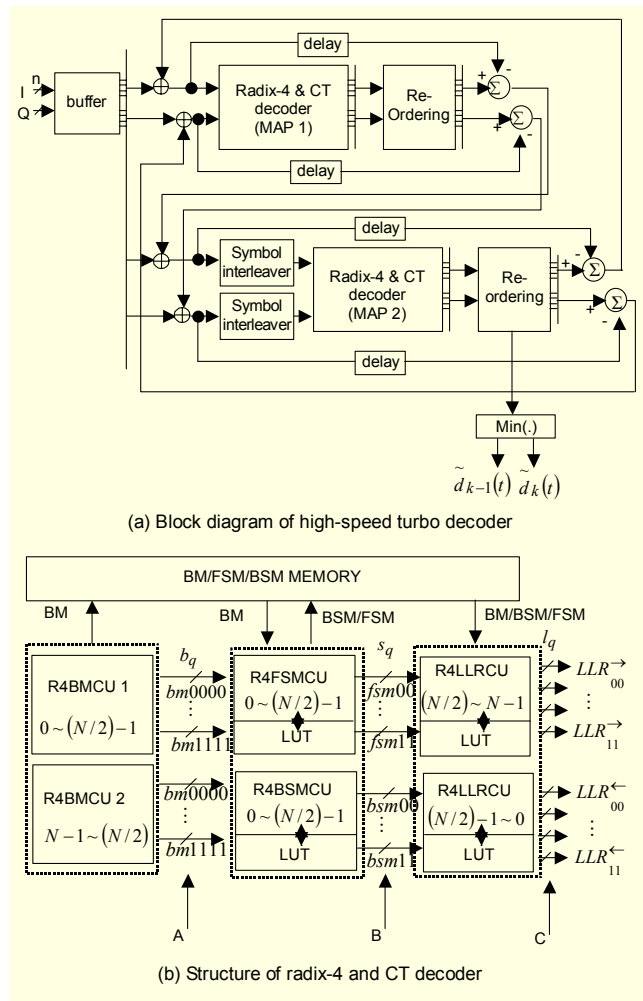


Fig. 6. Schematic diagram of proposed turbo decoder.

Table 2. The optimum quantized bits of the turbo decoder.

	Number of optimized quantization bits
$I, Q(n)$	8,8
b_q	9
s_q	9
l_q	9

conventional decoder using the same procedures. The required maximum clock of the device for the conventional and proposed decoders was 44 ns and 58 ns, respectively, based on the number of required logics and memories. The total delay latency of the proposed decoder is 27 ns for a single iteration, which is five times faster than that of a conventional algorithm, demonstrating the value of combining various latency improvement schemes into a single decoder design and architecture.

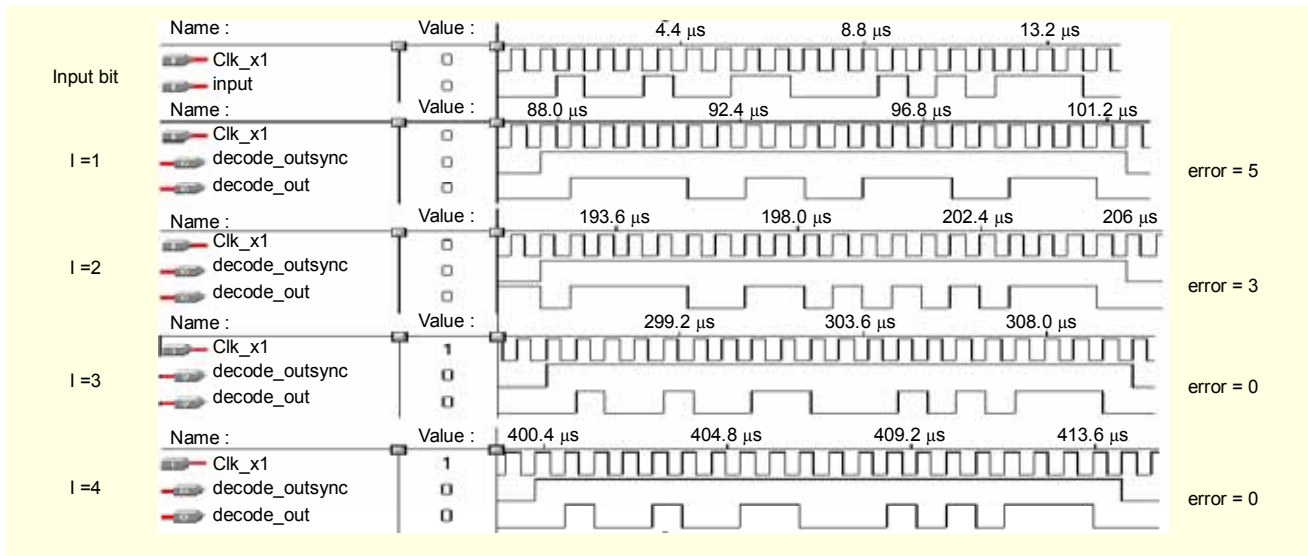


Fig. 7. Decoding process per iteration ($E_b/N_0=1$ dB).



Fig. 8. FPGA test-board for tests on number of iterations ($E_b/N_0=1$ dB).

Table 3. Comparison of decoding speed between a conventional method and the proposed method at each step.

	Radix-2	Radix-4
Main clock	44 ns	58 ns
BM delay (point A)	66 μ s	8 μ s
FSM/BSM delay (point B)	106.6 μ s	19 μ s
LLR output (point C)	136 μ s	27 μ s
Decoding speed	0.47 (Mbps)	2.37 (Mbps)

V. Conclusion

To extend the application area of turbo decoding to real time services, it is important to reduce the latency in the decoding

process of turbo decoders. In this paper, we proposed a new high-speed turbo decoding algorithm and architecture that provides a solution. Two new algorithms are presented, radix-4 and CT algorithms that are combined to two other techniques to reduced latency. The different decoding algorithms are incorporated into a decoder design architecture to show how much the decoding latency can be reduced. Fixed on $N=64$ and $F=1$, the results of the implementation in a Flex10k100GC503-4 confirmed that the decoding speed of the proposed decoder is superior to conventional algorithms. The error-correcting performance of the proposed algorithm is the same as that of the conventional one in the case of small block sizes (less than 300 bits) but is slightly degraded for larger interleaver sizes because the randomness of the symbol interleaver is reduced. Given that in the case of voice and data transmission in wireless communications the frame size tends to be very small, this decoder architecture is suitable to high-speed wireless communications.

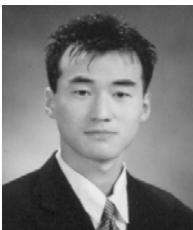
References

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes," *Proc. ICC93*, May 1993, pp. 1064-1070.
- [2] P. Robertson, E. Villebrun, and P. Hoeher, "A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain," *ICC95*, 1995, pp. 1009-1013.
- [3] D. Divsalar and F. Pollara, "Serial and Hybrid Concatenated Codes with Applications," *Proc. of the Int'l Symp. on Turbo Codes & Related Topics*, Sept. 1997, pp. 80-87.
- [4] S. Benedetto et. al., "Soft Output Decoding Algorithm in Iterative Decoding of Turbo Codes," *TDA progress rep. 42-124*, Jet Propulsion Lab., Pasadena, CA, Feb. 1996, pp. 63-87.

- [5] P.Hoeher, "New Iterative (Turbo) Decoding Algorithms," *Proc. of the Int'l Symp. on Turbo Codes & Related Topics*, Sept. 1997, pp. 63-70.
- [6] S.S. Pietrobon, "Implementation and Performance of a Serial MAP Decoder for Use in an Iterative Turbo Decoder," *Proc. IEEE Int. Symp. on Information Theory*, 1995, pp. 471-480.
- [7] S.S. Pietrobon, "Implementation and Performance of a Turbo/MAP Decoder," *Int'l J. of Satellite Comm.*, vol. 16, 1998, pp. 23-46.
- [8] Bernard Sklar, "A Primer on Turbo Code Concepts," *IEEE Comm. Magazine*, Dec. 1997.
- [9] D. Divsalar and F. Pollara, "Multiple Turbo Codes for Deep-Space Communications," *TDA Progress rep. 42-141*, Jet Propulsion Lab., Pasadena, CA, May 1995, pp. 66-77.
- [10] S. Benedetto and G. Montorsi, "Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schemes," *IEEE Trans. on Information Theory*, vol. 42, no. 2, Mar. 1996, pp. 409-429.
- [11] Digital Video Broadcasting (DVB) Interaction Channel for Satellite Distribution System, ETSI Reference EN 301 799, v1.2.2, Dec. 2000.
- [12] C. Douillard et al., The Turbo Code Standard for DVB-RCS, *Proc. of the 2nd Int Symp. on Turbo Codes*, Brest, France, Sept. 2000, pp. 551-554.



Ji Won Jung received his BS in 1989, MS in 1991, and PhD in 1995 from Sungkyunkwan University, Seoul, Korea, all in electronics engineering. From Nov. 1990 to Feb. 1992, he was with LG Research Center, Korea. From Sept. 1995 to Aug. 1996, he was with Korea Telecom (KT). From Aug. 2001 to July 2002, he was an Invited Researcher at the Communication Research Center Canada supported by NSERC. Since 1996, he joined the Department of Radio Science and Engineering, Korea Maritime University, Busan, Korea. His research interests are channel coding, digital modem, FPGA design technology, and digital broadcasting system.



In Ki Lee received his BS in 2003, MS in 2005, from Korea Maritime University, Busan, Korea. Since July 2005, he joined the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea, where he is currently a Researcher. His main research interests are channel coding, digital modem, FPGA design technology, and digital broadcasting system.



Duk Gun Choi received his BS in 2004, from Korea Maritime University, Busan, Korea. His research interests are channel coding, digital modem, FPGA design technology, and digital broadcasting system.



Jin Hee Jeong received BS degree in 2005 from Korea Maritime University, Busan, Korea. She is in Korea Maritime Graduate University. Her research interests are channel coding, digital modem, and FPGA design technology.



Ki Man Kim (M'95) received his BS in 1988, MS in 1990, and PhD in 1995 from Yonsei University, Seoul, Korea, all in electronics engineering. From 1995 to 1996, he worked for Yonsei Medical Center as a fellow, where he was involved in developing medical diagnosis equipments. In 1996, he joined the Department of Radio Science and Engineering, Korea Maritime University, Busan, Korea, where he is currently an Associate Professor. His main research interests include speech enhancement, microphone array, synthetic aperture sonar, virtual sound, and hardware implementation by using digital signal processor/CPLD.



Eun-A Choi received the BS degree in mathematics and the MS degree in information communications engineering from the Chonbuk National University (CNU), Chonbuk, Korea, in 1998 and 2000, respectively. She has been with the Digital Broadcasting Research Division, ETRI, Daejeon, Korea, as a Member of Research Staff since her graduation from CNU. In ETRI, she has been involved with Development Projects of Broadband Adaptive Satellite Communications and Broadcasting Convergence System, where she develops design algorithms for variable channel codecs. Her research interests include channel codes, digital communications, satellite communications and satellite broadcasting.



Deock Gil Oh received his BS degree in Electronics Engineering Department from Seoul National University (SNU) in 1980. He received the MS and PhD degrees in electronics engineering from SNU, Korea, in 1984 and 1996, respectively. He joined ETRI in 1982, where he is currently working as a Team Leader in Digital Broadcasting Research Division. His research interests include Wireless Access Technology, Mobile Communication and Broadcasting System and Future Generation Wireless Access System.