

An FPGA Implementation of High-Speed Flexible 27-Mbps 8-State Turbo Decoder

Duk Gun Choi, Min-Hyuk Kim, Jin Hee Jeong, Ji Won Jung, Jong-Tae Bae, Seok-Soon Choi, and Young Yun

In this paper, we propose a flexible turbo decoding algorithm for a high order modulation scheme that uses a standard half-rate turbo decoder designed for binary quadrature phase-shift keying (B/QPSK) modulation. A transformation applied to the incoming I-channel and Q-channel symbols allows the use of an off-the-shelf B/QPSK turbo decoder without any modifications. Iterative codes such as turbo codes process the received symbols recursively to improve performance. As the number of iterations increases, the execution time and power consumption also increase. The proposed algorithm reduces the latency and power consumption by combination of the radix-4, dual-path processing, parallel decoding, and early-stop algorithms. We implement the proposed scheme on a field-programmable gate array and compare its decoding speed with that of a conventional decoder. The results show that the proposed flexible decoding algorithm is 6.4 times faster than the conventional scheme.

Keywords: Coset mapping, radix-4 algorithm, dual-path processing, FPGA.

I. Introduction

Iterative decoding based on a symbol-by-symbol soft-in/soft-out decoding algorithm has attracted significant attention due to its near-Shannon-limit error performance [1]. As a powerful coding technique, a turbo code offers great promise to improve the reliability of communication systems such as those based on the Digital Video Broadcasting standard for Return Channel via Satellite (DVB-RCS) [2]. However, digital transmission via satellite can be severely affected by rain-induced signal fades. Finding an effective rain-fade countermeasure has been an important design objective for satellite communication systems, especially those offering broadband multimedia services at millimeter wavebands. There are various schemes to deal with this issue, such as up-link power control, flexible modulation and transmission, flexible channel coding, and so on. Flexible channel coding has received much attention recently, and it is a powerful scheme providing high reliability and high spectral efficiency over rain-fading channels [3]. The flexible channel-coding scheme uses different channel coding techniques depending on weather conditions. For example, under a clear sky, a more spectrum-efficient modulation and coding scheme such as an 8-PSK (phase shift keying) or 16-QAM (quadrature amplitude modulation) turbo trellis-coded modulation can be used to provide a higher data rate, while under heavy rain conditions, a quadrature PSK (QPSK) modulation with a half-rate turbo code can be employed to maintain acceptable performance.

The other important issues in high-speed applications of turbo decoders are decoding delay (or latency) and computational complexity. Like the maximum *a posteriori* probability (MAP) decoding algorithm, an iterative decoder processes the received symbols recursively to improve the

Manuscript received May 04, 2006; revised Dec. 08, 2006.

Duk Gun Choi (phone: +82 55 280 2662, email: dkchoi@stxengine.co.kr) was with Department of Radio Science and Engineering, Korea Maritime University, Busan, Korea, and is now with Department of Electrical Engineering, STX Engine, Changwon, Korea.

Min-Hyuk Kim (email: golden80@nate.com), Jin Hee Jeong (email: jihlovetalk@hotmail.com), Ji Won Jung (phone: + 82 51 410 4424, email: jwjung@mail.hhu.ac.kr), Jong-Tae Bae (email: ms43bjt@hotmail.com), Seok-Soon Choi (email: af560hg@hotmail.com), and Young Yun (email: yunyoung@hhu.ac.kr) are with the Department of Radio Science and Engineering, Korea Maritime University, Busan, Korea.

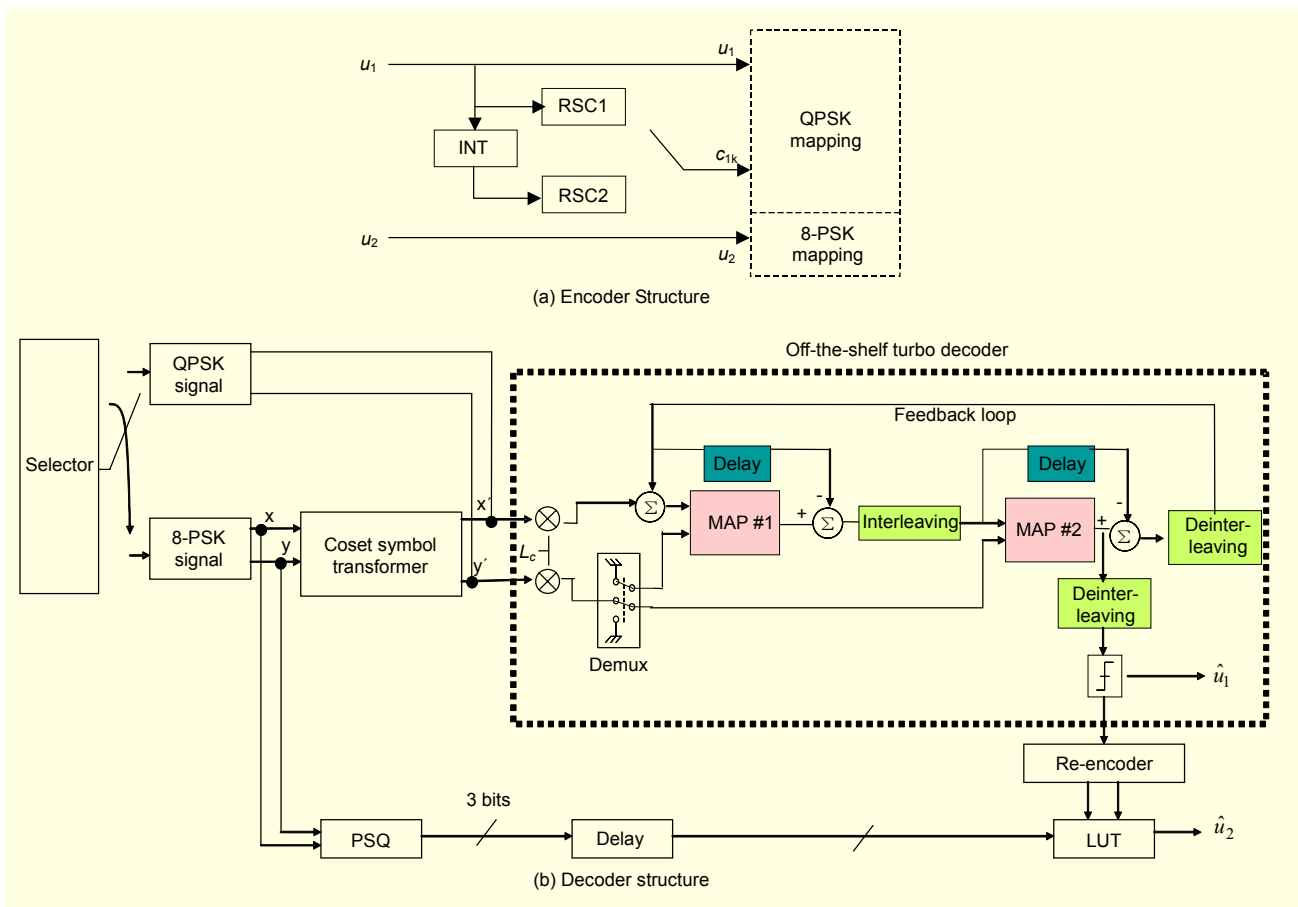


Fig. 1. Two-thirds rate turbo-coded pragmatic TCM encoder/decoder.

reliability of each symbol based on constraints which define the code. In the first iteration, the decoder only uses the channel output, and generates soft output for each symbol. The output reliability measures of the decoded symbols at the end of each decoding iteration are used as input for the next iteration. Therefore, the latency and complexity due to the iterative process make it difficult to implement the decoding algorithm in hardware or for high-speed wireless applications. To solve the latency problem, we propose the following four decoding algorithms which are combined into one decoder architecture: the radix-4, dual-path processing, parallel decoding, and early-stop. The resulting flexible turbo decoder is implemented using the coset symbol transformer [4], and it is suitable for high speed applications.

II. System Model

Trellis-coded modulation (TCM) is now a well-established method in digital communication systems. It is capable of achieving a coding gain within 3 to 6 dB range of the Shannon channel capacity for a trellis-coded 8-PSK system compared to an uncoded QPSK system [5]. The application

of turbo codes to TCM has received much attention in the literature. Hauro and others have proposed a new turbo trellis-coded modulation scheme [6]. As noted above, unlike in [6], we have chosen to combine turbo codes with a pragmatic concept. The result is the turbo-coded pragmatic TCM (TCPTCM). In this section, a TCPTCM with a two-thirds rate is presented. For the sake of clarity, only the case of an 8-PSK modulation is considered, but it can be easily generalized to an M-PSK modulation, where M equals a power of 2. Figure 1 shows the flexible turbo encoder/decoder structure with a two-thirds rate TCPTCM. The encoder consists of two recursive systematic convolutional (RSC) codes, and an interleaver (INT). The decoder consists of an off-the-shelf turbo decoder (DEC1, DEC2) with a half rate, a phase sector quantizer (PSQ), a coset symbol transformer (CST), and a re-encoder (RE). As a flexible concept, this structure may be easily expanded to high-order modulation schemes. The decoding procedure requires a standard turbo decoder without any modification in calculating the log-likelihood ratio, forward/backward state metric, and branch metric. As described in [4], the CST transforms the received 8-PSK symbols to QPSK symbols as

$$\begin{aligned} x' &= \sqrt{2} \cos(2(\varphi + 5\pi/8)), \\ y' &= \sqrt{2} \sin(2(\varphi + 5\pi/8)). \end{aligned} \quad (1)$$

where φ denotes the phase of the received 8-PSK signal, the x' and y' projections in the transformed QPSK constellation are obtained from the received 8-PSK symbol, and $\sqrt{2}$ is the scaling factor to project onto QPSK points with $(\pm 1, \pm 1)$.

In [5], the received 8-PSK symbols are used, which are directly input to a binary turbo decoder. However, in [4], the transformed QPSK symbols are used in order to use a binary turbo decoder. A PSQ gives information about the location of the received vector. The detailed algorithm is described in [4].

III. High Speed Flexible Turbo Decoder Algorithm

Since convolutional turbo codes are very flexible and are easily adapted to a large number of block sizes and coding rates, they have been adopted in the DVB-RCS standard. The use of an RCS terminal (RCST) includes individual and collective installation (such as SMATV) in the domestic environment. However, the applications of turbo codes are limited to low data-rate services because of the decoding speed limitation. Therefore, it is highly desirable to develop a high-speed turbo decoder. To solve the latency problem of a turbo decoder, four algorithms are proposed: the radix-4 algorithm, the dual-path processing algorithm, the full parallel decoding algorithm, and the early-stop algorithm based on the hard-decision-aided (HDA) scheme. The decoding iteration progresses until a certain stopping condition is satisfied. Then, hard decisions are made based on the reliability measures of the decoded symbol at the last decoding iteration [7].

1. Radix-4 Algorithm

In the radix-4 decoding algorithm [7], 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 to the current one such that the time interval from $t=k-2$ to $t=k$ is merged at time $t=k$. Therefore, we can decode two source data bits at the same time without any performance degradation while reducing the block size buffered in memory. 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 for 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 the radix-2 trellis. An example of

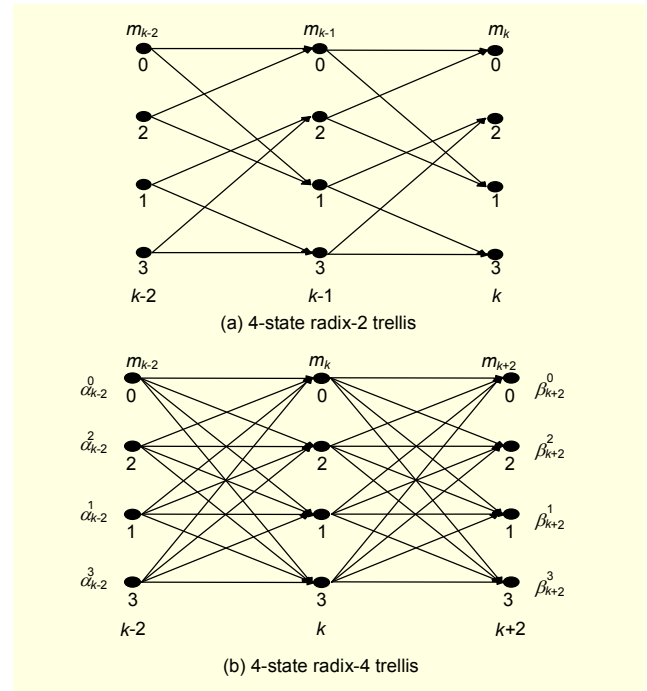


Fig. 2. Collapsing a 4-state radix-2 to a radix-4 trellis.

decomposition of a 4-state radix-2 into an equivalent radix-4 trellis using one stage of look-ahead is shown in Fig. 2, where $v=4$, $g_1=(15)_{\text{octal}}$, $g_2=(17)_{\text{octal}}$ with v denoting the constraint length [4].

2. Dual-Path Processing Algorithm

In a conventional scheme, the decoder must wait for the backward state metric (BSM) or forward state metric (FSM) calculation to be finished before calculating the extrinsic information. The dual-path processing method [7] does not 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 extrinsic information [4]. Figure 3 shows the operation of the dual-path processing.

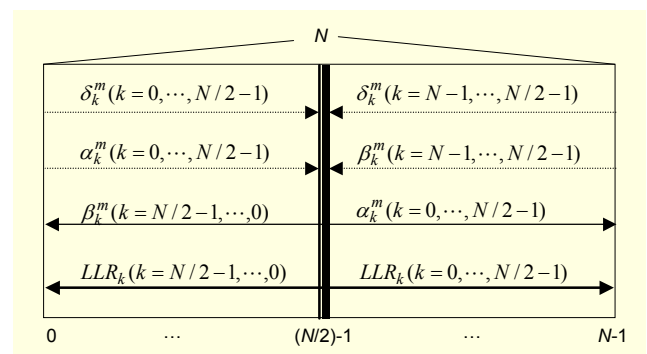


Fig. 3. Dual-path processing algorithm.

The procedure of the dual-path processing is as follows:

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

$$\alpha_0^i(s_0^i(m)) = \begin{cases} 1 & \text{for } m = 0, \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

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

where $\alpha_k^i(m)$ and $\beta_k^i(m)$ are the FSM and BSM at time k , for information bit i , and state m .

Step 2. After receiving the whole set of received symbols of N , simultaneously calculate FSMs (left to right) and BSMs (right to left):

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

for $k=0, \dots, N/2-1$, (3)

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

for $k=N-1, \dots, N/2$. (4)

Step 3. At the middle point, begin to calculate the log likelihood ratios (LLRs).

$$L(\vec{d}_k) = \log \frac{\sum_m \alpha_k^1(m) \beta_k^1(m)}{\sum_m \alpha_k^0(m) \beta_k^0(m)} \text{ for } k = (N/2), \dots, (N-1), \quad (5)$$

$$L(\overleftarrow{d}_k) = \log \frac{\sum_m \alpha_k^1(m) \beta_k^1(m)}{\sum_m \alpha_k^0(m) \beta_k^0(m)} \text{ for } k = (N/2)-1, \dots, 0, \quad (6)$$

where $L(\vec{d}_k)$ represents LLR outputs in the direction from right to left and $L(\overleftarrow{d}_k)$ represents LLR outputs in the direction from left to right.

3. Parallel Decoding Algorithm

Unlike the original turbo decoder consisting of two decoders concatenated in a serial fashion, the parallel decoder structure [7], [8], uses two decoders which operate in parallel and update each other simultaneously immediately after each one has completed its decoding. Unlike [8], to decode the estimated data, we use the sum of the LLR outputs of the parallel decoders to reduce the latency to one half while maintaining the same performance level.

4. Early Stop Algorithm

The decoding iteration continues processing until a certain stopping condition is satisfied, then hard decisions are made based on the reliability measures of the decoded symbols at the last decoding iteration. The HDA algorithm is used as an early-stop algorithm [9]. 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. In Table 1, “serial mode” means that the two MAP decoders are serially concatenated as shown in Fig. 1(b), and “parallel mode” means that the two MAP decoders are parallelly concatenated as shown in Fig. 4. Table 1 shows the average number of iterations in an HDA algorithm. At an E_b/N_0 of 6 dB, it requires only two iterations, which means that the decoding speed is improved or the power consumption cost is reduced by 74.9%. In the first iteration of the parallel mode, the extrinsic information of the first MAP decoder is not fed into the second MAP decoder. Therefore, the iterations of

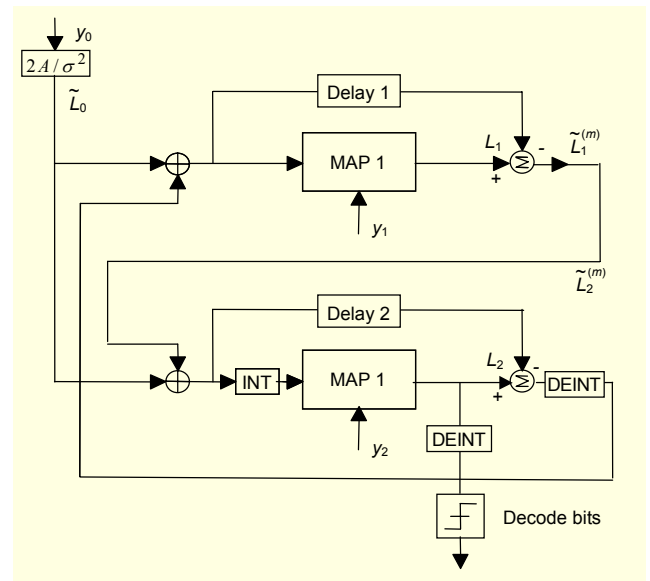


Fig. 4. Parallel decoder.

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

| E_b/N_0 (dB) | Serial mode | | Parallel mode | |
|----------------|------------------------------|--------------------------------|------------------------------|--------------------------------|
| | Average number of iterations | Decoding speed improvement (%) | Average number of iterations | Decoding speed improvement (%) |
| 4 | 3.03 | 62.1% | 4.75 | 40.6% |
| 5 | 2.03 | 74.6% | 2.67 | 66.6% |
| 6 | 1.85 | 76.9% | 2.01 | 74.9% |

the parallel mode are needed more than those of the serial mode to maintain the same performance.

5. Simulation Results

The bit-error rate (BER) performance of the proposed high-speed flexible turbo decoder architecture which employs the four algorithms described in the previous section is analyzed in this section. For comparison, Fig. 5 shows the performance of the new decoder and a conventional one using $v=4$ turbo codes with generator polynomials $g_1=(17)_{\text{octal}}$ and $g_2=(15)_{\text{octal}}$ as a function of the 8-PSK modulation scheme with an interleaving size of 212, and a two-thirds rate. The performance of the proposed decoder is almost the same as that of the conventional decoder. The reason for performance degradation of the source in the E_b/N_0 range of 3 dB to 5 dB is that the extrinsic information of the first MAP decoder is not fed into the second MAP decoder in parallel mode structure.

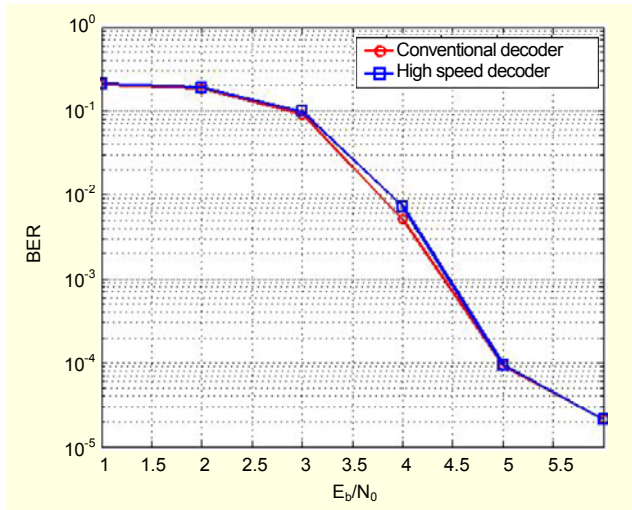


Fig. 5. Performance of the proposed decoder over an AWGN channel compared with that of a conventional algorithm.

IV. Design of the Flexible High-Speed Turbo Decoder

In this section, we present the flexible turbo decoder architecture with a two-thirds rate turbo-coded pragmatic TCM decoder using the off-the-shelf half-rate turbo decoder described in section II. Based on the high-speed algorithms described in section II, the entire architecture of the flexible turbo decoder of Fig. 1(b) is shown in Fig. 6. Our high-speed decoder can support both a half-rate turbo decoder BPSK modulation scheme and a two-thirds rate 8-PSK modulation scheme.

A schematic diagram of the high-speed turbo decoder implementation is shown in Fig. 7(a). A detailed signal flow of

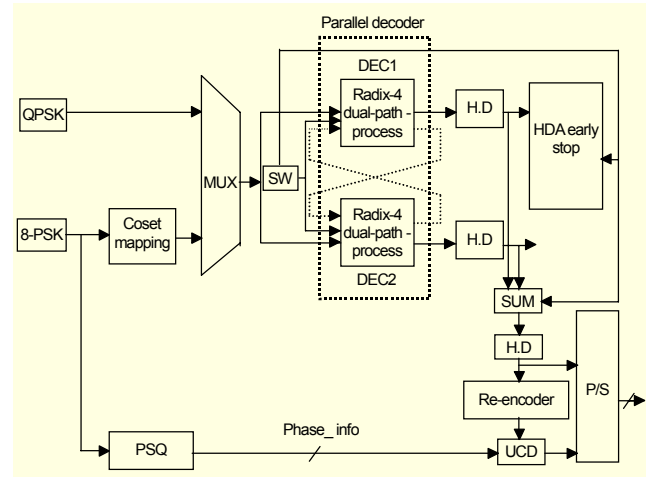


Fig. 6. Flexible turbo decoder structure.

the algorithm is shown in Fig. 7(b). In Fig. 7(a), the MAP 1 and MAP 2 decoders are operated in parallel, with their outputs being LLRs of input bits as shown in Fig. 7(b). For instance, $LLR0_{N/2-N}$ denotes log-likelihood ratios of input bits “00”, $N/2 - N$ denotes the time from $N/2$ to N , and the arrow (\rightarrow) denotes the direction of the LLR calculation, that is, left-to-right. The arithmetic logic unit (ALU) calculates the extrinsic information using LLR outputs, the received symbols, and the previous extrinsic information. To add the extrinsic information exactly in the next iteration, the decoder needs the dual port RAM (128×36) buffered ALU block. As shown in Fig. 7(b), the decoder consists of six major units: the Radix-4 Forward Branch Metric unit (R4FBMu), Radix-4 Backward Branch Metric unit (R4BBMu), Radix-4 Forward State Metric unit (R4FSMu), Radix-4 Backward State Metric unit (R4BSMu), Forward LLR unit (FLLRu), and Backward LLR unit (BLLRu). After the whole set of symbols is received, the quantized I and Q samples are fed to the R4FBMu and R4BBMu at the same time. The branch metrics between the branch codeword 0000 and the received symbols is denoted by $bm0000$. The R4FBMu and R4BBMu calculate the branch metrics for four samples of received data from left to right and from right to left simultaneously. As shown in Fig. 7(b), I_n ($n=1, 2$) and Q_n ($n=1, 2$) are fed to the R4FBMu to calculate the forward branch metric, and I_n ($n=1, 2$) and Q_n ($n=1, 2$) are also fed to the R4BBMu to calculate the backward branch metric. For the second iteration, the R4FBMu and R4BBMu need extrinsic information, Ex_n , that is, the LLR of input bit i_1i_2 . The index n of Ex_n denotes $2 \times i_2 + i_1$. The dR4FSMu calculates the forward state metrics from left to right, and the R4BSMu calculates the backward state metrics from right to left. The data calculated by the R4FSMu and R4BSMu is buffered in two separate 64×72 dual-port RAMs, R4FSM_RAM and

R4BSM_RAM. When the R4FSMu and R4BSMu reach the same point, the FLLRu and BLLRu begin to calculate the LLR of information n using the data read from the R4FSM_RAM and R4BSM_RAM, respectively. The ALU block shown in Fig. 7(a) begins to calculate the extrinsic information and store extrinsic information in a 128×36 dual-port RAM for the next iteration.

For the optimized implementation of the high-speed decoder, we determined the optimum quantized bits of each block shown in Fig. 7, that is, r_q bits of received in-phase and quadrature signals, b_q bits of the R4FBMu and R4BBMu outputs, s_q bits of the R4FSMu and R4BSMu outputs, and l_q

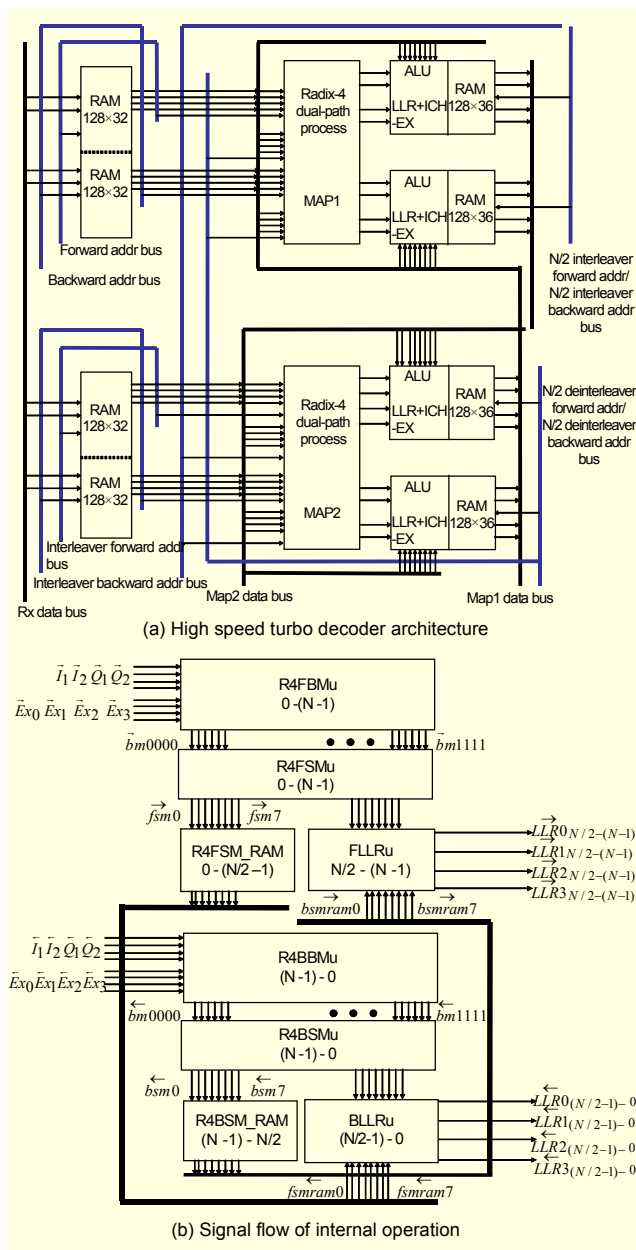


Fig. 7. Block diagram of proposed high-speed turbo decoder.

Table 2. Optimum quantized bits of the adaptive turbo decoder. (rate = 2/3, 8-states, N=212 bits, 3 iterations, 8-PSK).

| | Number of optimized quantization bits |
|-------|---------------------------------------|
| r_q | 8 |
| b_q | 9 |
| s_q | 9 |
| l_q | 9 |

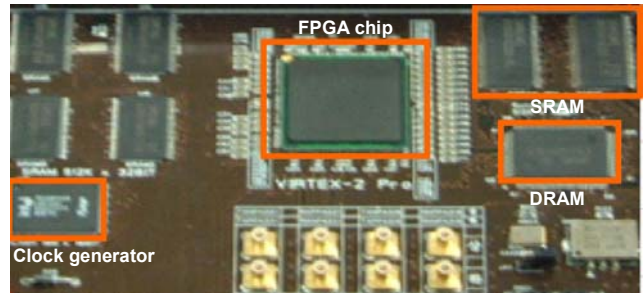


Fig. 8. Implementation of the adapted high-speed turbo decoder.

bits of the FLLR and BLLR. 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. The optimum quantization bits of the turbo decoder derived from the fixed-point simulations are listed in Table 2.

We implemented the flexible turbo decoder using a very high-speed hardware description language (VHDL) and verified its operation by register transfer level (RTL) simulation. During the verification, some symbol errors were added to the C-description. We then confirmed that the errors were corrected while being processed in the RTL simulation. The VHDL and C processes communicate with each other through the program language interface. The decoder implemented in VHDL was synthesized for the Xilinx FPGA commercial chip (VIRTEX2P (XC2VP30-5FG676)) with three million gates as shown in Fig. 8. The received data file generated by the C process at an E_b/N_0 rate of 5 dB is fed into the internal memory of a Xilinx FPGA chip.

To compare the decoding speeds of the conventional decoder and the high-speed flexible turbo decoder, we implemented the conventional decoder using the same procedure. Based on Table 1, since the required iteration number is 3 at an E_b/N_0 rate of 5 dB, we fixed the iteration number of both decoders to 3. The maximum operating clock cycle is 18 ns. The common parameters of both decoders are an interleaving size of 212, 8-states, and an 8-PSK modulation scheme. For the conventional decoder, just the serial mode and radix-2 method were employed. Table 3 shows comparison of the decoding speeds of the conventional and high-speed decoders. When

Table 3. Comparison of decoding speed between a conventional method and the proposed method ($N=212$, iteration = 3, 8-state, 8-PSK, main clock speed = 18 ns).

| | Conventional decoder | Radix-4 + serial mode | Radix-4 + parallel mode | Radix-4 + parallel mode + dual-path process |
|-------------------------|----------------------|-----------------------|-------------------------|---|
| Execution time (clocks) | 2861 | 1431 | 768 | 446 |
| Decoding speed | 4.11M | 8.23M | 15.33M | 26.4M |

combining the radix-4, parallel, and dual-path process algorithms, the proposed high-speed decoder is 6.4 times faster than the conventional decoder.

V. Conclusion

In this paper, we presented a high-speed flexible turbo decoding algorithm with two coded bits per symbol, based on a realization of rate $n/(n+1)$ trellis-coded scheme using an off-the-shelf turbo decoder originally designed for a standard half-rate turbo decoder for B/QPSK modulation [7]. Though the proposed decoder exhibits a small loss of less than 0.2 dB compared to the conventional turbo TCM, it needs less hardware, consumes less power, and reduces the receiver cost. The proposed approach may be extended to variable coding rates (5/6 and 8/9), depending on how many uncoded bits are assigned. Also, it can be used for 2^m -QAM constellations with $m \geq 4$.

To extend the application area of the turbo decoder to real time services, it is important to reduce its decoding latency. In this paper, we proposed a new high-speed turbo decoding implementation architecture. Two new low latency versions of the decoder were presented which employ the radix-4, dual-path processing, parallel mode, and early-stop algorithms. With the parameters of $N=212$, 8-states, 3-iterations, and an 8-PSK modulation scheme, we implemented the high-speed flexible turbo decoder on a Xilinx FPGA chip (VIRTEX2P (XC2VP30-5FG676)). From the test results, we confirmed that the proposed decoder is 6.4 times faster than the conventional algorithm. The DVB-RCS standard [2] requires a decoding speed of 10 Mbps, while our implementation reaches a speed of 26.4 Mbps. Therefore, our high-speed decoding algorithms and implementation methodology can be used not only for DVB-RCS but for other high-speed wireless applications.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon

Limit Error-Correcting Code and Decoding: Turbo Codes," *Proc. ICC'93*, 1993.

- [2] "Digital Video Broadcasting Standard for Return Channel via Satellite (DVB-RCS)," ETSI TR 101 790, vol. 2.1, 2003.
- [3] J.W. Jung and X.Huang, "Performance Analysis and Optimum Design of Pragmatic Code for Rain-Attenuation Compensation in Satellite Communication," *AIAA Conference*, Montreal, May 2002.
- [4] B.E. Wahlen and C.Y. Mai, "Turbo Coding Applied to Pragmatic Trellis-Coded Modulation," *IEEE Communications Letters*, vol. 4, no. 2, Feb. 2000, pp. 65-67.
- [5] H. Ogiwara et al., "Improvement of Turbo Trellis-Coded Modulation System," *IEICE Trans. Fundamentals*, vol. E81-A, no. 10, Oct. 1998.
- [6] E.A. Choi, J.W. Jung, N.S. Kim, Y.I. Kim, and D.G. Oh, "A Simplified Decoding Algorithm Using Symbol Transformation for Turbo Pragmatic Trellis-Coded Modulation," *ETRI Journal*, vol. 27, no. 2, Apr. 2005, pp. 223-226.
- [7] J.W. Jung et al., "Design and Architecture of Low-Latency High-Speed Turbo Decoder," *ETRI Journal*, vol. 27, no. 5, Oct. 2005, pp. 525-532.
- [8] S.H. Yoon and Y. Bar-Ness, "A Parallel MAP Algorithm for Low Latency Turbo Decoding," *Communications Letters*, vol. 6, no. 7, Jul. 2002, pp. 288-290.
- [9] R.Y. Shao, S. Lin, and M.P.C. Fossorier, "Two Simple Stopping Criteria for Turbo Decoding," *IEEE Trans. Communications*, vol. 47, no. 8, pp. 1117-1120.
- [10] L.R. Bahl et al., "Optimal Decoding of Linear Code for Minimizing Symbol Error Rate," *Trans. on Info. Theory*, vol. IT-20, Mar. 1994, pp. 248-287.



Duk Gun Choi received his BS, MS degrees from Korea Maritime University, Busan, Korea, in 2004 and 2006, respectively. In 2006, he joined the STX Company as a researcher. His research interests are channel coding, digital modem, FPGA design technology, and digital broadcasting system.



Min-Hyuk Kim received the BS degree in radio sciences and engineering from Korea Maritime University, Busan, Korea, in 2006. He is currently working toward the MS degree at Korea Maritime University. His research interests are channel coding, digital modem, field-programmable gate-array (FPGA) design

technology, and digital broadcasting systems.



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.



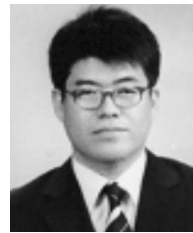
Ji Won Jung received his BS, MS, and PhD degrees from Sungkyunkwan University, Seoul, Korea, in 1989, 1991, and 1995, respectively, all in electronics engineering. From November 1990 to February 1992, he was with the LG Research Center, Anyang, Korea. From September 1995 to August 1996, he was with Korea Telecom (KT). From August 2001 to July 2002, he was an Invited Researcher with the Communication Research Center Canada [supported by Natural Sciences and Engineering Research Council of Canada (NSERC)]. Since 1996, he has been with the Department of Radio Science and Engineering, Korea Maritime University, Busan, Korea. His research interests are channel coding, digital modem, field-programmable gate-array (FPGA) design technology, and digital broadcasting systems.



Jong-Tae Bae received the BS degree in radio sciences and engineering from Korea Maritime University, Busan, Korea, in 2007. He is currently working toward the MS degree at Korea Maritime University. His research interests are channel coding, digital modem, field-programmable gate-array (FPGA) design technology, and digital broadcasting systems.



Seok-Soon Choi received the BS degree in radio sciences and engineering from Korea Maritime University, Busan, Korea, in 2007. He is currently working toward the MS degree at Korea Maritime University. His research interests are channel coding, digital modem, field-programmable gate-array (FPGA) design technology, and digital broadcasting systems.



Young Yun received the BS degree in electronic engineering from Yonsei University, Seoul, Korea, in 1993, the MS degree in electrical and electronic engineering from the Pohang University of Science and Technology, Pohang, Korea, in 1995, and the PhD degree in electrical engineering from Osaka University, Osaka, Japan, in 1999. From 1999 to 2003, he was an engineer with the Matsushita Electric Industrial Company Ltd., Osaka, developing MMICs for wireless communications. In 2003, he joined the Department of Radio Sciences and Engineering, Korea Maritime University, Busan, Korea, where he is currently an Assistant Professor. His research interests include design and measurement for RF/microwave and millimeter-wave ICs as well as design and fabrication for high electron-mobility transistors (HEMTs) and heterostructure bipolar transistors (HBTs).