An FPGA Implementation of High-Speed Adaptive Turbo Decoder

Min Huyk Kim* *Associate Member*, Ji-Won Jung* *Regular Member*,
Jong Tae Bae*, Seok Soon Choi* *Associate Members*, In Ki Lee** *Regular Member*

ABSTRACT

In this paper, we propose an adaptive turbo decoding algorithm for high order modulation scheme combined with originally design for a standard rate-1/2 turbo decoder for 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. Adaptive turbo decoder process the received symbols recursively to improve the performance. As the number of iterations increase, the execution time and power consumption also increase as well. The source of the latency and power consumption reduction is from the combination of the radix-4, dual-path processing, parallel decoding, and early-stop algorithms. We implemented the proposed scheme on a field-programmable gate array (FPGA) and compared its decoding speed with that of a conventional decoder.From the result of implementation, we confirm that the decoding speed of proposed adaptive decoding is faster than conventional scheme by 6.4 times.

Key Words : Coset Mapping, Radix-4 Algorithm, Dual-Path Processing, FPGA

I. Introduction

Iterative decoding based on symbol-by-symbol soft-in/soft-out decoding algorithm has significant attention, due to its near Shannon-limit error performance. As a powerful coding technique, a turbo code offers a great promise for improving the reliability of communication systems such as those based on the DVB standard for Return Channel via Satellite(DVB-RCS). However, digital transmission via satellite can be severely affected by rain-induced signal fade. Rain-fade countermeasure has been one important design objective of any satellite communication systems, especially those offering broadband multimedia services. There are various schemes to deal with this issue, such as up-link power control, adaptive modulation and transmission, and adaptive channel coding, etc. Among them, the adaptive channel coding receives much attention recently, and it is a most powerful scheme to warrant the high reliability and high spectral efficiency over the rain-fade channel. The adaptive channel coding scheme uses different channel coding techniques, depending on weather conditions. For example, under clear sky, a more spectrum-efficient modulation and coding such as an 8-PSK or 16-QAM turbo trellis-coded modulation can be used to provide a higher data rate; while under heavy rain condition, a QPSK with a half rate turbocode can be employed to maintain an acceptable performance.

In this paper we present a single turbo decoder need to decode all modulation schemes in order to need less hardware and less power consumption, and to reduce the receiver cost. Anovel decoding procedure allows the use of an off-the-shelf turbo decoder originally designed for a standard rate-1/2 turbo decoder over B/QPSK modulation, to decode some trellis coded modulation schemes over 2^m-PSK/QAM constellations, with $m \geq 3$.

This paper presents a new method for an adaptive turbodecoding algorithm using the coset sym-

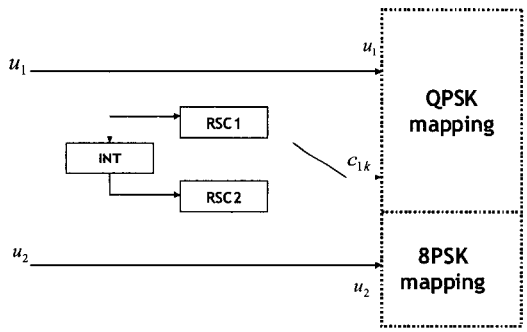
* 한국해양대학교 전파공학과 위성통신 연구실 (ms43bjt@hhu.ac.kr)
** 한국전자통신연구원 광대역 무선 멀티미디어 연구팀
논문번호 : KICS2006-03-146, 접수일자 : 2006년 3월 28일, 최종논문접수일자 : 2007년 3월 30일

bol transformation where received 8-PSK signals are transformed to QPSK signals after some manipulations. The novel adaptive turbo codes with two coded bits per symbol is based on a realization of rate n/(n+1) trellis coded scheme using an off-the-self turbo decoder, originally design for a standard rate-1/2 turbo decoder for B/QPSK modulation.

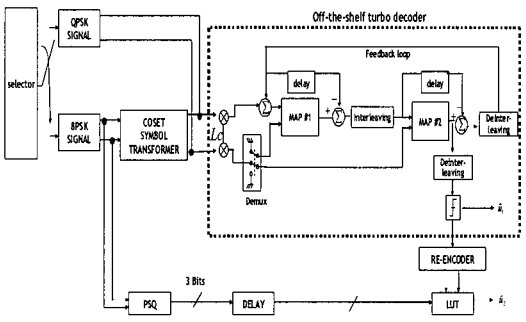
Important issues in high-speed applications of turbo decoders are decoding delay and computational complexity. Like maximum a posterior probability (MAP) decoding, iterative decoder processes the received symbols recursively to improve the reliability of each symbol based on constrains that specify the 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 iterations are used as input for next iteration. Therefore, the latency and complexity caused by several iterations and high computation order, it can be difficult to implement the decoding in hardware and to apply the high-speed wireless applications. To solve the latency problem, in this paper four decoding algorithms are presented and combined into one decoder architecture such as the radix-4, dual-path processing, parallel decoding, and early-stop algorithms.

Ⅱ. Adaptive Turbo Decoding Algorithm

Trellis coded modulation(TCM) is now a well-established method in a digital communication systems capable of achieving 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. The application of turbo codes to TCM has received much attention in the literature. Hauro et al. have proposed a new turbo trellis coded modulation scheme[4]. As noted above, unlike in, we have chosen to combine turbo codes with a pragmatic concept. Turbo-coded pragmatic TCM(TCPTCM) is chosen as an adaptive turbo code. In this section, a TCPTCM with a rate-2/3 is presented. For the sake of clarity, only the case of 8-PSK modulation is considered, but it is easily generalized to M-PSK for M a power of 2. Fig. 1 shows the adaptive turbo encoder/decoder structure with a rate of 2/3 TCPTCM. Encoder consists of two RSC (Recursive Systematic Convolutional) codes, and interleaver (INT). Decoder consists of an off-the shelf turbo decoder (DEC1, DEC2) with rate 1/2, a phase sector quantizer(PSQ), coset symbol transformer(CST), and a re-encoder(RE). As an adaptive concept, this structure may be easily expanded to high-order modulation schemes. The decoding procedures require a standard turbo decoder without any modification in calculating the log-likelihood ratio, forward/backward state metric, and branch metric.



(a) Encoder Structure



(b) Decoder Structure

Fig 1. Rate-2/3 turbo-coded pragmatic TCM encoder/decoder

2.1 Mapping of bits to signal

Two information bits (u_1, u_2) are encoded to produce three coded bits (u_2, u_1, c_{1k}), which are

mapped onto 8-PSK signal points, where c_{1k} is the output of the standard turbo encoder and is punctured. The signal points are labeled by a triplet (u_2, u_1, c_{1k}), as shown in Fig 2.

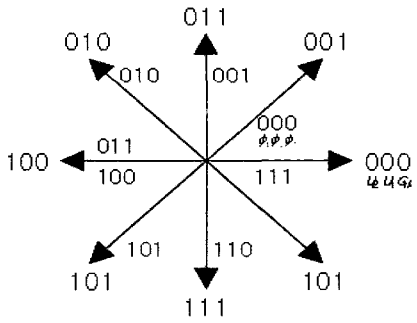


Fig 2. Labels and phase information of 8-PSK constellation

2.2 Coset Symbol Transformer (CST)

Let (x,y) denote the I- and Q- channel values of the received 8-PSK symbols r(t). Let φ denote the phase of the received signal.

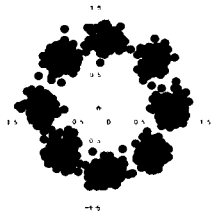
$$r(t) = x + jy \tag{1}$$

$$\varphi = \tan^{-1}(y/x) \tag{2}$$

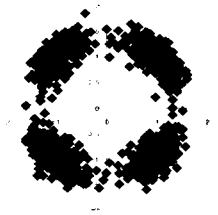
In order to use the turbo decoder with rate-1/2, a transformation is applied such that the 8-PSK points are mapped into QPSK points labeled by (u_1, c_{1k}), as shown in Fig. 2. The x' and y' projections in the transformed QPSK constellation are obtained from the received 8-PSK symbols by (3).

$$x' = \sqrt{2}\cos(2(\varphi + 5\pi/8))$$
$$y' = \sqrt{2}\sin(2(\varphi + 5\pi/8)) \tag{3}$$

$\sqrt{2}$ is the scaling factor to project onto QPSK points with (±1, ±1), $5\pi/8$ is the phase rotation constant to map into the QPSK point with(u_1, c_{1k}) : (11)→$\pi/4$, (01)→$3\pi/4$, (00)→$5\pi/4$ and (10)→$7\pi/4$. Fig. 3 shows the received 8-PSK symbols transformed into QPSK symbols at *Eb/No* of 12dB.



(a) Received 8-PSK symbols



(b) Transformed into QPSK symbols

Fig 3. Transformed QPSK symbols(Eb/N0 of 12dB)

2.3 Phase Sector Quantizer(PSQ)

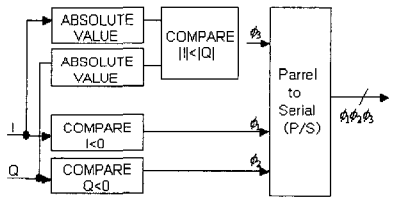


Fig 4. Phase sector quantizer and Soft decision mapping block

The signal vector space is quantized to determine the locations of received symbols, and PSQ in Fig. 4 is used for decoding the uncoded bits, u_2. The PSQ is designed with the assumption that the in-phase(I) and quadrature(Q) components of the received signals r(t), will be converted to q quantization bits. The circuit is shown in Fig. 4. Three comparators and two absolute generations produce the three bits of phase information indicating one of the sectors. Each of the three phase information bits gives information about the location of the received vector: $\emptyset_2$ and $\emptyset_3$ indicate the quadrant and the remaining one bit indicates the location within the quadrant. When |I| < |Q|, ϕ_1 is one. In Fig. 2, the numbers in

8-PSK constellation are denoted, ϕ_1, ϕ_2, ϕ_3.

CST outputs, (x', y'), are used by an iterative MAP turbo decoder to produce estimates, $\tilde{u}_1$, of u_1. The value $\tilde{u}_1$ is then encoded by the rate 1/2 turbo code encoder providing estimates, $\tilde{u}_1$ and $\tilde{c}_{1k}$ of u_1 and c_{1k}. After the turbo decoder estimates the coded bits, it remains to determine the uncoded bit. This is accomplished by making a threshold decision. Using the estimated code bits, $(\tilde{u}_1, \tilde{c}_{1k})$, and phase information, (ϕ_1, ϕ_2, ϕ_3), we have to determine the uncoded bit, $\tilde{u}_2$ of u_2. Due to the structure of the turbo decoding algorithm, every decoder delays the data by a fixed number of symbol periods. The phase information bits ϕ_1, ϕ_2, ϕ_3 must be delayed by the amount of turbo decoding delay to match with the reconstructed code sequence. A simple look-up table(LUT) as shown in Table 1 can be used to estimate $\tilde{u}_2$. As an example, if re-encoded bits $(\tilde{u}_1, \tilde{c}_{1k})$ are (00) and the phase information (ϕ_1, ϕ_2, ϕ_3) is (111), then $\tilde{u}_2$ =0.

Table 1. Look-up table for estimating $\tilde{u}_2$

estimated coded bits $(\tilde{u}_1, \tilde{c}_{1k})$	ϕ_1, ϕ_2, ϕ_3	$\tilde{u}_2$
00	(000),(001),(110),(111)	0
	(010),(011),(100),(101)	1
01	(000),(001),(010),(111)	0
	(011),(100),(101),(110)	1
10	(000),(101),(110),(111)	0
	(001),(010),(011),(100)	1
11	(000),(001),(010),(011)	0
	(100),(101),(110),(111)	1

2.4 Simulation Results

The performance computer simulations to compare the performance of a turbo-coded pragmatic TCM(TCPTCM) is compared to pragmatic TCM (PTCM) and published simulation results of the turbo-coded TCM method in [4]. TCM method of Hauro et al. in [4]. The simulation results are plotted (Fig. 5) for comparable 2-bps/Hz systems: 1) uncoded QPSK; 2) rate-2/3 pragmatic TCM(PTCM) with a single 64-state convolutional encoder; 3) rate 2/3 turbo-coded pragmatic TCM(TCPTCM, in Fig. 1 with two 16-state RSC encoders, a 500-bit random interleaver, and five decoding iterations; and 4) rate-2/3 turbo TCM with two 16-state RSC encoders, 500-bit random interleaver, and five decoding iterations[4]. At a BER of 10-5, TCPTCM achieves a 2 dB gain relative to PTCM. Compared to [4], the proposed decoder exhibits a small loss of less than 0.2 dB.

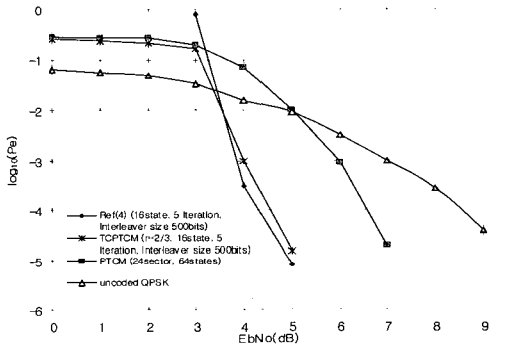


Fig 5. Bit error rate performance comparison between pragmatic TCM(PTCM), turbo-coded pragmatic (TCPTCM), and turbo TCM(TTCM)

Ⅲ. High Speed Turbo Decoder Algorithm

Since convolutional turbo codes are very flexible codes, easily adaptable to a large rate of block sizes and coding rates, they have been adopted in the DVB standard for Return Channel via Satellite(DVB-RCS). The use of RCST(RCS Terminal) includes individual and collective installation(e.g. SMATV) in domestic environment. However, the applications of turbo codes are limited to specific data such as low data-rate services because of their limit of decoding speed. Therefore, it is highly required to develop the high-speed turbo decoder. To solve the problem with latency of turbo decoder, four kinds of algo-

rithms are introduced. The first algorithm is radix-4 algorithm and the second algorithm is the dual-path processing algorithm. The third algorithm is the full parallel decoding algorithm. The fourth algorithm is the early-stop algorithm based on hard-decision-aided (HDA) scheme. The decoding iteration processes 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.

3.1 Radix-4 Algorithm

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. 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. 6, where $v=3$, $g_1=(7)_{octal}$, $g_2=(5)_{octal}$ with v denoting the constraint length.

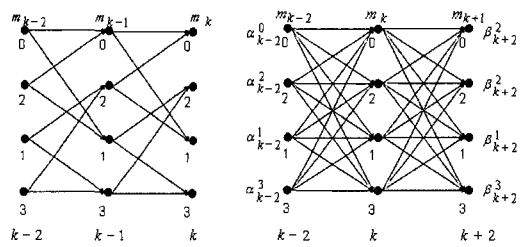


Fig 6. Four-state radix-2 to radix-4 trellis.

3.2 Dual-Path Processing Algorithm

In a conventional scheme, the decoder must wait for finishing the backward state metric (BSM) (or forward state metric (FSM)) calculations before calculating the extrinsic information. The dual-path processing 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 extrinsic information. Fig. 7 shows the operation of dual-path processing.

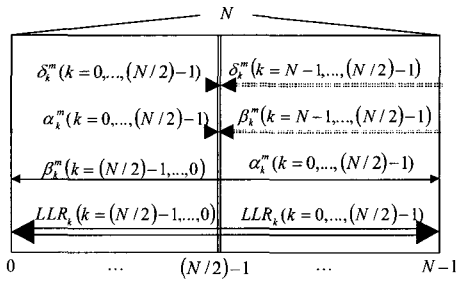


Fig 7. Dual-Path Processing Algorithm.

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

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

$$\begin{aligned} \alpha_0^i(s_0^i(m)) &= 1 \quad for \quad m=0 \\ &= 0, \quad else \\ \beta_N^i(s_b^i(m)) &= 1 \quad for \quad m=0 \\ &= 0, \quad else \end{aligned} \tag{4}$$

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

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

$$\hat{\alpha}_k^i(m) = \exp\left(\frac{2}{\sigma^2}(x_k i + y_k Y_k(i,m))\right) \sum_{j=0}^{1} \hat{\alpha}_{k-1}^j(S_b^j(m)) \quad (k=0,\ldots,(N/2)-1) \tag{5}$$

$$\hat{\beta}_k^i(m) = \sum_{j=0}^{1} \hat{\beta}_{k+1}^j(m) \exp\left(\frac{2}{\sigma^2}(x_{k+1} j + y_{k+1} Y_{k+1}(j,S_j^i(m)))\right) \quad (k=(N-1),\ldots,(N/2)) \tag{6}$$

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

$$L(\vec{d}_k)=\log\frac{\sum_m \alpha_k^1(m)\beta_k^1(m)}{\sum_m \alpha_k^0(m)\beta_k^0(m)}(k=(N/2),\ldots,(N-1)) \quad (7)$$

$$L(\overleftarrow{d}_k)=\log\frac{\sum_m \alpha_k^1(m)\beta_k^1(m)}{\sum_m \alpha_k^0(m)\beta_k^0(m)}(k=(N/2)-1,\ldots,0) \quad (8)$$

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

3.3 Parallel Decoding Algorithm

Different from the original turbo decoder consisting of two decoders concatenated in a serial fashion, we present 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.

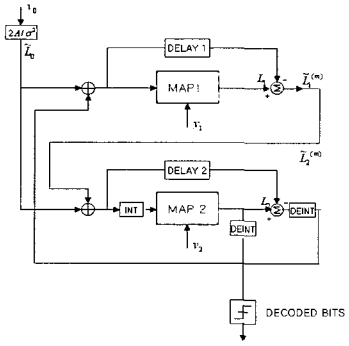


Fig 8. Parallel Decoder

3.4 Early Stop Algorithm

The decoding iteration processes until a certain stopping condition is satisfied,hard decisions are made based on the reliability measures of the decoded symbol at the last decoding iteration. HADalgorithm 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 thecurrent block and outputs the hard decision bits. Table 2 shows the average number of iterations in an HADalgorithm. At an E_b/N_0 of 6 dB, it requires about 2.01 iterations relative to 8 for a given performance in parallel mode of turbo-coded pragmatic(TCPTCM) N=500, 8 states, radix-4 . Thismeans that the decoding speed is improved or the power consumption (cost) is reduced by 74.9 %.

Table 2. The average number of iterations according to Eb/No (the predetermined number of iterations is 8).

EbNo[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%

3.5 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. 9 shows the performance of the new decoder and a conventional one using v=3 turbo codes with generator polynomials $g_1=(7)_{octal}$, $g_2=(5)_{octal}$ as a function of interleaving size 212 and as a function of the number of iteration 8. The performance of the new decoder using radix-4 and parallel method and the conventional decoder are similar. Therefore, we increase the decoding speed without degradation of performance.

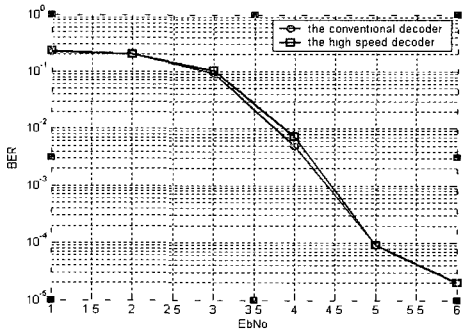


Fig 9. Performance of the proposed decoder over an AWGN channel compared with that of a conventional algorithm

Ⅳ. Design of the Adaptive High-Speed Turbo Decoder

In this section, as shown in Fig. 10, we propose the adaptive turbo decoder architecture with a rate of 2/3 turbo-coded pragmatic TCM decoder using the off-the-shelf half rate turbo decoder based on Fig. 1. Based on reduced latency and power consumption version of turbo decoder such as radix-4 algorithm, dual-path processing, parallel algorithm, and early top algorithm, Fig. 10 shows the entirely architecture of the adaptive turbo decode we design. Our decoder can support both half rate turbo decoder with BPSK modulation scheme and rate of 2/3 with 8PSK modulation scheme.

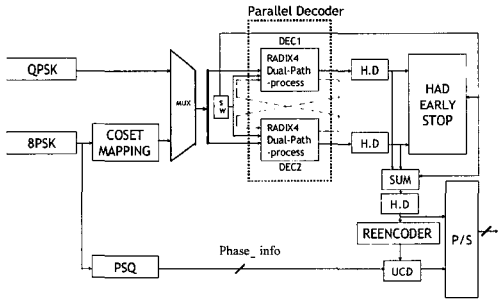
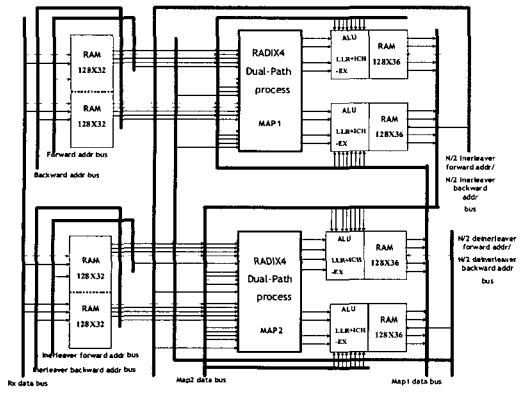


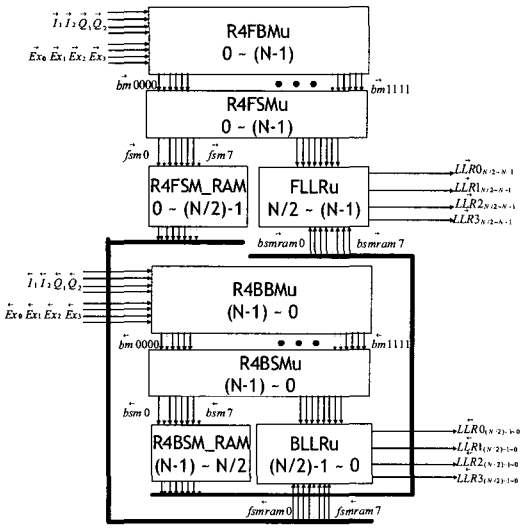
Fig 10. Adaptive turbo decoder structure

A schematic diagram of the high-speed turbo decoder for implementation is shown in Fig. 11(a). A detailed signal flow of internal MAP based on high-speed algorithms is shown in Fig. 11(b). In Fig. 11(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. 11(b). For instance, $\vec{LLR}0_{N/2\sim N}$ denotes log-likelihood ratios of input bits "00", N/2 ~ N denotes From the time of N/2 to N, and the arrow $\rightarrow$ denotes the direction of the LLR calculation, that is, left-to-right. The ALU (Arithmetic Logic Unit) calculates the extrinsic informationusing LLR outputs, the received symbol and the previous extrinsic information. To add the extrinsic information exactly in the next iteration, the decoder needs the dual port RAM(128x36) buffered ALU block. As shown in Fig. 11(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), Backward LLRunit (BLLRu).After receiving the whole set of received symbols, 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 in the direction of left to right and right to left simultaneously. As shown in Fig. 11(b), $I_n(n=1,2)$ and $Q_n(n=1,2)$ are fed to R4FBMu to calculate forward branch metric and $I_n(n=1,2)$ and $Q_n(n=1,2)$ are also fed to R4BBMu to calculate backward branch metric. From the second iteration, R4FBMu and R4BBMu need extrinsic information, Ex_n, that is LLR of input bit "$i_1 i_2$". The index n of Ex_n means $2*i_2+i_1$. R4FSMu calculates forward state metrics in the direction left-to-right and R4BSMu calculates backward state metrics in the direction right-to-left. The data calculated from R4FSMu and R4BSMu are buffered in dual-port RAM with size of 64x72. Therefore we need the two dual-port RAM that is R4FSM_RAM and R4BSM_RAM. When the R4FSMu and R4BSMu reach the same point, then FLLRu and BLLRu begin to calculate the LLR of information n using the data read from R4FSM_RAM and R4BSM_RAM respectively. ALU block shown in Fig. 11(a) begins to calculate the extrinsic information and store extrinsic information in dual-port RAM with size of 128x36 in order to input to next iteration.

In order to implement optimally the high-speed decoder, we determined the optimum quantized bits of each block shown in Fig. 11. r_q bits of received in-phase and quadrature signals, b_q bits of R4FBMu and R4BBMu outputs, sq bits of



(a) High speed turbo decoder architecture



(b)Signal flow of internal MAP

Fig 11. The block diagram of proposed high-speed turbo Decoder

R4FSMu and R4BSMu outputs, and lq bits of 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, and the optimum quantization bits of the turbo decoder derived from the fixed-point simulations are listed in Table 3.

We design theadaptive turbo decoder using VHDL(Very high-speed Hardware Description Language) and verified its operation by RTL simulation. During the verification, some errors were added in C-description. Then we confirmed that the errors were corrected while being processed in RTL simulation. VHDL and C language

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

	Number of optimized quantization bits
rq	8
bq	9
sq	9
lq	9

process communicate with each other through program language interface. The decoder designed by VHDL was synthesized using the Xillinx FPGA chip (VIRTEX2P(XC2VP30- 5FG676)) as shown in Fig. 12, The received data file generated by C language process at *Eb/No* of 5[dB] is fed into internal memory of Xillinx FPGA chip.

We designed the high-speed adaptive turbo decoder with 8-state, N=212, R = 2/3, and 8PSK modulation scheme. In order to compare the decoding speed of the conventional serial turbo decoder based on radix-2 trellis structure.To compare the decoding speed between conventional and high-speed adaptive turbo decoder, we implemented the conventional decoder using the same procedures. Based on Table 2, since the required iteration number is 3 in the case of Eb/N0=5[dB], we fixed the iteration number to 3. The maximum operating clock speed ofthe conventional and proposed decoders is 18[*ns*] Table 4 shows comparison of decoding speed between conventional and high-speed decoder. In the case of combining the radix-4 and parallel and dual-path process, the proposed high-speed decoder is faster than conventional one by 6.4 times.

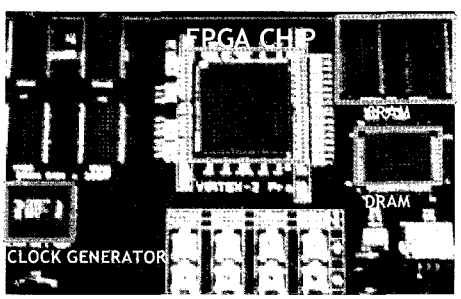


Fig 12. The implement chip-set of the adapted high-speed turbo decoder

Table 4. Comparison of decoding speed between a conventional method and the proposed method (N=212, iteration = 3, 8-state, 8PSK, 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

V. Conclusion

In this paper, we presented the adaptive turbo decoding algorithmwith two coded bits per symbol, based on a realization of rate n/(n+1) trellis coded scheme using an off-the-shelfturbo decoder originally designed for a standard rate-1/2 turbo decoder for the B/QPSK modulation. Compared to conventional turbo TCM, though the proposed decoder exhibits a small loss of less than 0.2 dB, but it needs less hardware, consumes less power, and reduces the receiver cost. The proposed approach may be extended to a variable coding rate(rate-5/6 and rate-8/9), which depends 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 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 implementation architecture. Two new low latency version of decoder are presented, radix-4 algorithm and dual-path processing that are combined to parallel mode and early-stop algorithm. Fixed on the parameters of N=212, iteration=3, 8-states, 3 iterations, and 8PSK modulation scheme, we designed the adaptive high-speed turbo decoder using the Xilinx chip(VIRTEX2P(XC2VP30-5FG676)). From the results, weconfirmed that the decoding speed of the proposed decoder is faster than conventional algorithms by 6.4 times.

References

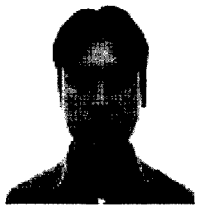
[1] C. Berrou, A. Glavieux, and P.Thitimajshima, "Near Shannon Limit Error-Correcting Code and Decoding: Turbo Codes", in Proc. Of ICC'93, 1993.

[2] L. R. Bahl et al, "Optimal Decoding of Linear Code for Minimizing Symbol Error Rate", Trans. on info. Theory, Vol. IT-20, pp.248-287, Mar. 1994.

[3] S.S.Pietrobo, "Implementation and Performance of a Turbo/MAP Decoder," to be appear in International Journal of Satellite Communications.

[4] Haruo Ogiwara, et al, "Improvement of Turbo Trellis-Coded Modulation System", IEICE Trans. Fundamentals, vol.E81-A, No.10, October 1998.

김 민 혁 (Min Hyuk Kim) 준회원



2006년 2월 : 한국해양대학교 전파공학과(공학사)

2006년 3월~현재 : 한국해양대학교 전파공학과 석사과정

<관심분야> 위성통신, 이동통신, 변 · 복조기술, 채널코딩, FPGA 기술 등

정 지 원 (Ji-Won Jung) 정회원

1989년 2월 : 성균관대학교 전자공학과(공학사)

1991년 2월 : 성균관대학교 전자공학과(공학석사)

1995년 2월 : 성균관대학교 정보공학과(공학박사)

1991년 1월~1992년 2월 : LG 정보통신연구소 연구원

1995년 9월~1996년 8월 : 한국통신 위성통신연구실 선임연구원

1997년 3월~1998년12월 : 한국전자통신연구원 초빙연구원

1996년 9월~현재 : 한국해양대학교 전파공학과 정교수

2001년 8월~2002년 8월 : 캐나다 NSERC Fellowship (Communication Research Center 근무)

<관심분야> 위성통신, 이동통신, 변.복조기술, 채널코딩, FPGA 기술 등

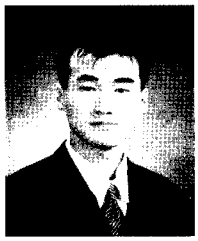
배 종 태 (Jong Tae Bae) 준회원

2007년 2월 : 한국해양대학교 전파공학과(공학사)
2007년 3월~현재 : 한국해양대학교 전파공학과 석사과정
<관심분야> 위성통신, 이동통신, 변·복조기술, 채널코딩, FPGA 기술 등

최 석 순 (Seok Soon Choi) 준회원

2007년 2월 : 한국해양대학교 전파공학과(공학사)
2007년 3월~현재 : 한국해양대학교 전파공학과 석사과정
<관심분야> 위성통신, 이동통신, 변·복조기술, 채널코딩, FPGA 기술 등

이 인 기 (In Ki Lee) 정회원



2003년 8월 : 한국해양대학교 전파공학과(공학사)
2003년 9월~2005년 8월 : 한국해양대학교 전파공학과(공학석사)
2005년 7월~현재 : 한국전자통신연구원 광대역 멀티미디어 연구팀 연구원
<관심분야> 위성통신, 이동통신, 변.복조기술, 채널코딩, FPGA 기술 등