

An FPGA Design of High-Speed Turbo Decoder

Ji-Won Jung*, Jin-Hee Jung*, Duk-Gun Choi*, In-Ki Lee* *Regular Members*

ABSTRACT

In this paper, we propose a high-speed turbo decoding algorithm and present results of its implementation. The latency caused by (de)interleaving and iterative decoding in conventional MAP turbo decoder can be dramatically reduced with the proposed scheme. The main cause of the time reduction is to use radix-4, center to top, and parallel decoding algorithm. The reduced latency makes it possible to use turbo decoder as a FEC scheme in the real-time wireless communication services. However the proposed scheme costs slight degradation in BER performance because the effective interleaver size in radix-4 is reduced to an half of that in conventional method. To ensure the time reduction, we implemented the proposed scheme on a FPGA chip and compared with conventional one in terms of decoding speed. The decoding speed of the proposed scheme is faster than conventional one at least by 5 times for a single iteration of turbo decoding.

Key Words : Radix-4 Algorithm, Center to top Algorithm, Parallel Algorithm, Early Stop

I. Introduction

In satellite communications and 3rd generation mobile communication system, turbo codes are adopted for high-speed data transmission. Recently, the trend of wireless communication is changed from the conventional narrow-band voice service to the wide-band multimedia service. Therefore, it is highly required to develop the high-speed turbo decoder. The important problems of high-speed applications of turbo decoder are decoding delay and computational complexity. To solve the problem with latency of turbo decoder, two types of decoding algorithms are proposed in the paper. The first type is radix-4 decoding algorithm. The previous state at $t-k-2$ going forwards to a current state at $t=k$ and the reverse state at $t=k+2$ going backwards from a current. Time interval from $t-k-2$ to $t=k$ is merged into $t=k$. Therefore, we can decode two bits source data at the same time without performance degradation. And we may reduce the block size buffered in memory. In order to apply

the radix-4 algorithm, we have derived the branch metric (BM), forward state metric (FSM), α , backward state metric (BSM), β , and log likelihood ratio (LLR), λ . Another type is center to top algorithm. In conventional scheme, to calculate the LLR, decoder must wait for until finishing the BSM (or FSM) calculations. But center-to top method doesn't need to wait. Decoder calculates FSM (left to right) and BSM (right to left) simultaneously. When FSM and BSM reach at same point, decoder begins to calculate the LLR.

In this paper, we combined the two types of proposed scheme and parallel decoding algorithm. The proposed scheme is desirable in designing of high-speed turbo decoder because it provides significant reductions in memory requirements of the decoder, as well as allowing increased parallelism. To testify high-speed decoder, we implemented the proposed scheme onto an FPGA chip (Altera FLEX10K) and compared with conventional one in terms of decoding speed. The proposed scheme is demonstrated in a 4-states, $R = 1/2$ turbo MAP decoder.

* 한국해양대학교 전파공학과 위성통신 연구실(dkchoi@bada.hhu.ac.kr),
논문번호 : KICS2004-11-270, 접수일자 : 2004년 11월 08일

The paper consists of the following Sections. In Section 2, we analyze the three types of high-speed turbo decoding algorithm. Section 3 describes computer simulation results of new high-speed turbo decoder. In Section 4, we find optimal parameters for implementation and design the high-speed turbo decoder. Section 5 concludes the paper.

II. High-Speed Decoding Algorithm

2.1 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} subtrellises, each consisting of k iterations of a 2^k -state trellis. Each 2^k -states subtrellis can be collapsed into an equivalent on stage radix- 2^k trellis by applying k levels of lookahead to the recursive update. Collapsing the trellis does not affect decoder performance since there is one-to-one mapping between collapsed trellis and radix-2 trellis. An example of the decomposition for a four-state radix-2 into an equivalent radix-4 trellis using one stage of lookahead is shown in Figure 1. (K=3 $g_1=(7)_{octal}$, $g_2=(5)_{octal}$ K means constrain length)

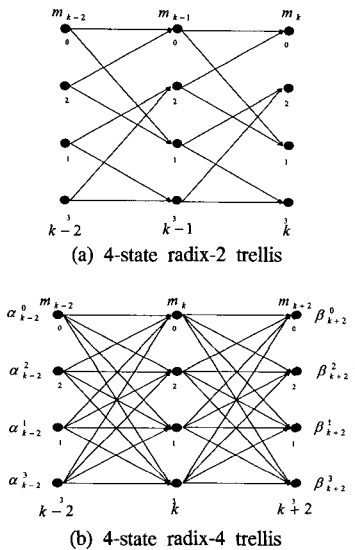


Figure 1. Trellis Structure

For radix-4 trellis iterations time k to $k-2$ time backward state for calculating the α_k^m and from time $k+2$ to time k forward state for calculating the β_k^m can be expressed

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

where d_k and m_k mean uncoded data bit state values at time of k , and $(a \parallel b)$ means concatenate the a and b . We have that $m_k = \{m_{0,k}, m_{+1,k}, \dots, m_{v-1,k}\}$ corresponding to the encoder state. m_{k-2} is used for calculating the α_k^m and m_{k+2} is used for calculating the β_k^m at k times.

2.1.1 Derivation of $\delta_k^m, \alpha_k^m, \beta_k^m$ and λ_k^m .

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

$$\begin{aligned} \delta_k^{p,m} &= P_r(D_k = p, S_k = m_k, R_k) \\ &= P_r(D_k = i_{k-1} \parallel i_k, S_k = m_k, R_k) \\ &= K_k \exp\left(-\frac{2}{\sigma^2} (x_{k-1}d_{k-1} + y_{k-1}Y_{k-1} + x_k d_k + y_k Y_k)\right) \end{aligned} \quad (2)$$

where i_k is the value of information bit at time k , and $p = i_{k-1} \parallel i_k$. Therefore we have that $i = \{0,1\}$, and $p = \{00,01,10,11\}$. K_k is a constant and x_k and y_k are received in-phase signal and quadrature signal. Y_k is coded bit as a function of the input bit d_k and encoder state m_k . Forward state metric at time k and m state α_k^m can be expressed ;

$$\begin{aligned} \alpha_k^m &= P_r(R_1^{k-2} \mid D_k = p, S_k = m_k, R_k^N) = P_r(R_1^{k-2} \mid S_k = m_k) \\ &= \sum_{p=0}^3 \alpha_{k-2}^{b(p,m_k)} \frac{p_b(p,m_k)}{\delta_{k-2}^{p,m_k}} \end{aligned} \quad (3)$$

where, $R_k^N = \{R_k, R_{k+1}, \dots, R_n\}$

$b(p, m_k)$ is previous $k-2$ time stage given an input p and state m_k

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

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

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

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

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

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

$$\lambda_k^{p, m} = \alpha_k^m \beta_{k+2}^{f(p, m)} \delta_k^{p, m} / P_r(R_1^N) \quad (7)$$

In order to decode the two bits $D_k = d_{k-1} \parallel d_k$, Equation(8) can be used.

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

The overall configuration of radix-4 turbo MAP decoder is depicted in Figure 2, which is composed of two constituent decoder (MAP1 and MAP2) concatenated serially with symbol interleavers. Each MAP calculates four LLR, $\lambda(d_k^{00}), \lambda(d_k^{01}), \lambda(d_k^{10}), \lambda(d_k^{11})$. Where $\lambda(d_k^{00})$ denotes corresponds to LLR value for two input bits '00'.

2.2 Scheme 2 : CT(Center to Top) algorithm

Another type is center to top(CT) algorithm. In conventional algorithm, to calculate the LLR, decoder must wait for until finishing the BSM (or

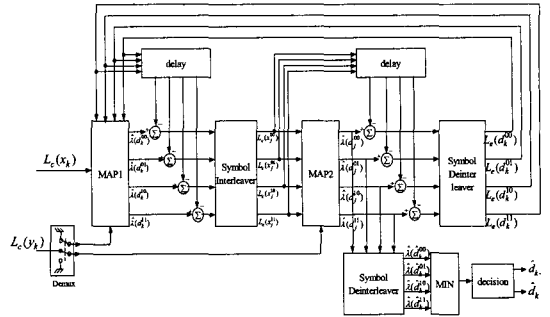


Figure 2. Block diagram of turbo decoder based on radix-4

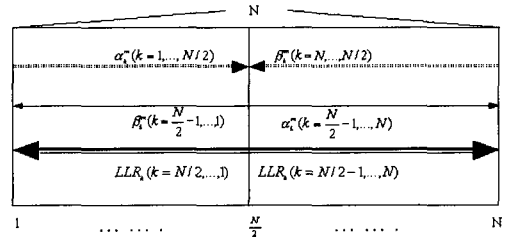


Figure 3. CT algorithm

FSM) calculations. But CT algorithm doesn't need to wait. Decoder calculates FSM(left to right) and BSM(right to left) simultaneously. When FSM and BSM reach at same point, decoder begins to calculate the LLR. Figure 3 illustrates the operation of CT

2.3 Scheme 3 :Parallel Decoding Algorithm(4)

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

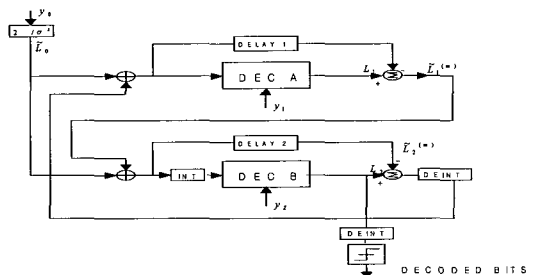


Figure 4. Parallel decoder structure

III. Computer Simulation Results

In this section, we analyze the bit-error rate (BER) performance for new high-speed turbo decoder that is incorporated three schemes. For comparison between new high-speed turbo decoder and conventional one, we produce the performance in Figure 5 using by K=3 turbo codes with generator polynomial $g_1=(7)_{octal}$, $g_2=(5)_{octal}$ as function of interleaver size N (N_s means symbol interleaver size for radix-4 method) and number of iteration I . Symbol interleaver in radix-4 method, an information stream of length N_s composed of 2-bit words is fed into random interleaver. From the Figure 5, it can be verified that the performances of proposed scheme is very matched to conventional one in small block size (less than 300 bits). For large interleaver size (more than 300 bits), since randomness of symbol interleaver is reduced, the performance of the new scheme is slightly degraded than conventional one. Therefore it can be concluded that the proposed decoder will considerably reduce decoding latency while it maintains almost the same performance with conventional algorithm.

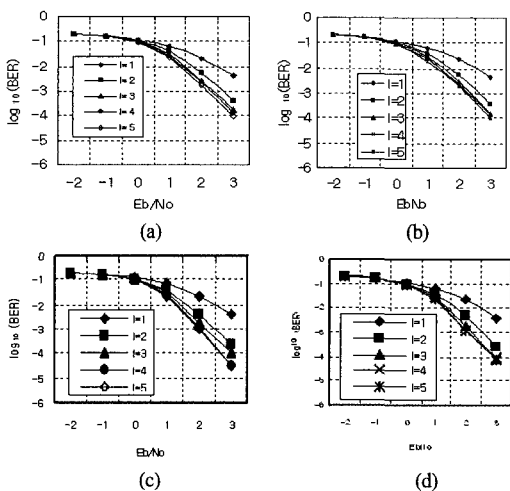
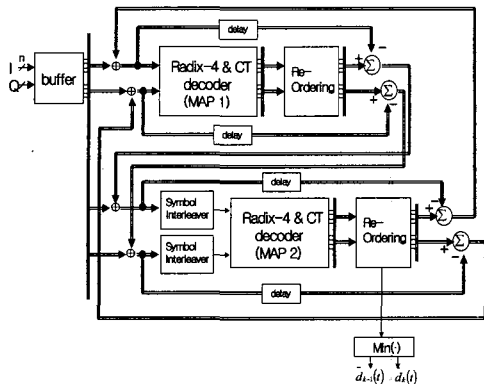


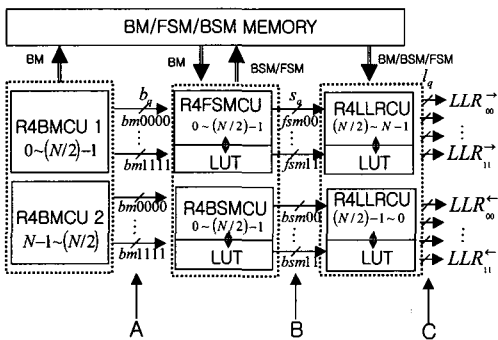
Figure 5. Performance of proposed algorithm over Gaussian channel compared with that of conventional algorithm in according to interleaver size and number of iteration
 (a) Conventional algorithm ($N_s=200$)
 (b) Proposed algorithm ($N_s=200$)
 (c) Conventional algorithm ($N_s=300$)
 (d) Proposed algorithm ($N_s=150$)

IV. Design of High-Speed Turbo Decoder

The schematic diagram of high-speed turbo decoder for implementation is shown in Figure 6(a). The detailed block diagram of Radix-4 & CT decoder (MAP1, MAP2) in Figure 6(a) is shown in Figure 6(b). In Figure 6(a), MAP 1 and MAP 2 are operated in parallel, the output of MAP 1 (or MAP 2) are log-likelihood ratios of input bits as shown in Figure 6(b). For example, LLR_{00} denotes log-likelihood ratios of input bits "00", and arrow \rightarrow denoting the direction of LLR calculation. To add the extrinsic information to the received symbols exactly in the next iteration, decoder needs Re-ordering blocks. Re-ordering blocks arrange the LLR such as LIFO (Last-In First-Out) or FIFO (First-In First Out). As shown in Figure 6(b), the decoder consists of four major units; R4BMCU (Radix-4 Branch Metric Unit), R4BSMCU (Radix-4 Backwards State Metric Unit), R4FSMCU (Radix-4 Forwards State Metric Unit), and R4LLRCU (Radix-4 for LLR Calculator Unit). Here, R4 means Radix-4 algorithm described in subsection 2.1. In high-speed turbo decoder described in Section 2, it needs four R4BMCUs, two R4FSMUs/R4BSMUs, and four R4LLRCUs. After receiving the whole received symbols, Quantized I and Q samples are fed to R4BMCU. bm_{0000} means branch metrics between branch codeword "0000" and received symbols. R4BMCU calculates branch metrics for 4 samples of the received data. To apply CT algorithm, 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) of interval from 0 to $(N/2)-1$ (or $N-1$ to $N/2$), we don't need to calculate the branch metrics, just refer to memory of R4BMCU2 (or R4BMCU1). R4FSMCU, R4BSMCU, and R4LLRCU need E-Table. The argument of the exponential function, input of E-Table, was designed to be equal to the address looked up table (LUT). In this paper, we implement the E-Table by internal ROM (EAB, Embedded Array Block) without using any external ROM.



(a)Block diagram of high-speed turbo decoder



(b) Structure of Radix-4 and CT decoder

Figure 6. Schematic diagram of proposed turbo decoder

In order to implement high-speed decoder, we have to decide the optimum quantized bits of each block in Figure 6, i.e., demodulator output, in-phase signal and quadrature signal, R4BMCU output of b_q , R4BSMCU output of s_q , and log-likelihood ratio l_q . By fixed-point computer simulation, Output of demodulator is quantized to 8 bits. The internal parameters of turbo decoder are always saturated to 9 bits. The optimum quantization bits of turbo decoder by fixed-point simulation are listed in Table 1.

The result of timing simulation for high-speed turbo decoder is shown in Table 2. The interleave size, N , set to 64 bits due to limited capacity of embedded memory of Flex10k100GC503-4. The results from iterative decoding over AWGN channel at $E_b/N_0=1\text{dB}$ is also presented in Figure 7. Since error is corrected according to increase the iteration, we confirmed that the high-speed decoder implemented onto Flex10k100GC503-4 is doing well.

Table 1. The optimum quantized bits of turbo decoder

	Number of optimized quantization bits
I,Q(n)	8,8
bq	9
Sq	9
lq	9

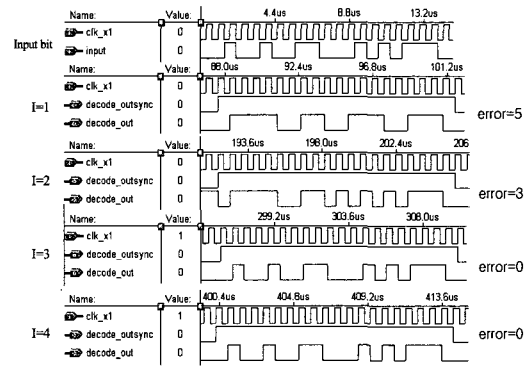


Figure 7. Decoding process for Iteration ($E_b/N_0=1\text{dB}$)

In order to compare decoding speed between conventional turbo decoder and proposed one, we implemented conventional decoder with same procedures of high-speed decoder. We knew that the required maximum clock of device for conventional and proposed decoder was 44ns and 58ns, respectively. The reason that proposed algorithm needs more clock period is mainly due to the increased number of required logics and memories, and so on. The total delay latency of proposed algorithm is 27us after single iteration, which is faster than that of conventional algorithm by 5 times. Therefore, decoding speed of proposed algorithm is faster than that of conventional one due to applying the Radix-4, CT algorithm, and parallelism.

Table 2. Comparison of speed between conventional method and proposed method at each point

	radix2	radix4
main clock	44ns	58ns
BM delay(point A)	66us	8us
FSM / BSM delay(point B)	106.6us	19us
LLR output(point C)	136us	27us
Decoding speed	0.47[Mbps]	2.37[Mbps]

V. Conclusion

To extend the application area of turbo decoding to real time services, the most important thing is to reduce the latency in decoding process of turbo decoder. In this paper, we proposed a new high-speed turbo decoding algorithm that is believed to be a proper solution. Two types of new algorithms are presented, Radix-4 and CT algorithm, another type is parallel decoding algorithm in Reference[4]. Three different kinds of decoding algorithms are incorporated to show how much decoding latency can be reduced. Fixed on $N=64$ and $I=1$, the results of implement onto Flex 10k100GC503-4, we confirmed that the decoding speed of proposed algorithm is superior than conventional algorithm. The performance of the proposed algorithm is the same as that of the conventional one in the case of small block size (less than 300 bits). For larger interleaver size (more than 300 bits), since randomness of symbol interleaver is reduced, the performance of the proposed algorithm is slightly degraded than conventional scheme. However, in the case of voice and data transmission in a wireless communications, the frame size is very small. Actually, in case of designing by ASIC, its speed may faster than an FPGA. Therefore, the proposed algorithm that reduce the decoding latency and enhances the decoding speed can be applied to the high-speed wireless communications.

REFERENCES

[1] C.Berrou, A.Glavieus, and P.Thitimajshima, "Near Shanon Limit Error-Correcting Coding and Decoding: Turbo-Codes," in proc. ICC93, pp.1064-1070, May 1993.

[2] P.Robertson, E.Villebrun, and P.Hoeher, "A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain," ICC95, pp.1009-1013. 1995.

[3] D.Divsalar and F.Pollara, "Serial and Hybrid Concatenated Codes with Applications," Proceedings of the International Symposium

on Turbo Codes & Related Topics, pp.80-87 September 1997.

[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, pp. 63-87, February 1996.

[5] P.Hoeher, "New Iterative(Turbo) Decoding Algorithms," Proceedings of the International Symposium on Turbo Codes & Related Topics, pp.63-70, September 1997.

[6] S.S. Pietrobon, "Implementation and performance of a serial MAP decoder for an iterative turbo decoder," in Proc., IEEE Int. Symposium on Information Theory, pp. 471-480, 1995.

[7] S.S.Pietrobon, "Implementation and Performance of a Turbo/MAP Decoder," International Journal of Satellite Communications, vol.16, . pp.23-46, 1998.

[8] Bernard Sklar, "A Primer on Turbo Code Concepts," IEEE Communications December 1997.

[9] D. Divsalar and F. Pollara, "Multiple Turbo Codes for Deep-Space Communications," TDA progress rep. 42-141, Jet Propulsion Lab., Pasadena, CA, 66-77, May 1995.

[10] S.Benedetto and G.Montorsi, "Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schemes," IEEE Transaction on Information vol.42, no. 2, pp.409-429, March 1996.

정지원 (Ji-Won Jung)

정회원



1989년 2월 성균관대학교 전
자공학(학사)
1991년 2월 성균관대학교 전
자공학(석사)
1995년 2월 성균관대학교 전
자공학(박사)
1991년 1월~1992년 2월 LG

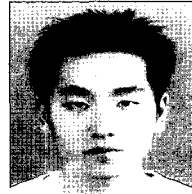
정보통신연구소 연구원

1995년 9월~1996년 8월 한국통신 위성통신연구실
선임연구원

1997년 3월~1998년 12월 한국전자통신연구원 초
빙연구원
1996년 9월~현재 한국해양대학교 전파공학과 조교수
2001년 8월~2002년 8월 캐나다 NSERC Fellow-
ship(Communication Research Center 근무)
<관심분야> 위성통신, 이동통신, 변·복조기술, 채널
코딩, FPGA 기술 등

최 덕 군(Duk Gun Choi)

정회원



2004년 8월 한국해양대학교 전파
공학(학사)
2004년 9월~현재 한국해양대
학교 전파공학 석사과정
<관심분야> 변·복조기술, 채널
코딩, FPGA 기술, 위성통신
등

정 진 희(Jin-Hee Jung)

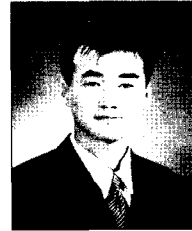
준회원



2005년 2월 한국해양대학교 전파
공학과(학사)
2005년 3월~현재 한국해양대학
교 전파공학과 석사과정
<관심분야> 변·복조기술, 위성
통신, 채널코딩, FPGA 기술
등

이 인 기(In-Ki Lee)

준회원



2003년 8월 한국해양대학교 공학
(학사)
2003년 9월~현재 한국해양대
학교 공학석사 과정
<관심분야> 채널 코딩, 변·복
조 기술, FPGA 기술, 위성
통신 등