



이인기, 최덕균, 정지원, 윤영, 김기만, 민경석

한국해양대학교 전자공학과  
E-Mail : popularity@bada.hhu.ac.kr

# Design Methodology for High-Speed Turbo Decoder Architectures

## 서론

차세대 이동통신 및 위성통신 시스템에서 요구되는 멀티미디어 통신 등 고속 대용량 데이터 통신을 위해서는 Shannon 이론에 의거한 전송 한계에 거의 근접한 강력한 에러정정 능력을 가진 터보부호의 사용이 절실히 요구된다. 차세대 무선 통신시스템은 고속 데이터 전송 및 동영상 등을 포함한 무선 멀티미디어 전송에 기반을 두고 있으며, 고속 데이터, 동영상 등의 무선 멀티미디어 서비스를 쌍방에게 제공하기 위해서는 고속 데이터 전송에 따른 고속 터보 복호기 개발이 필수적이다. 현재의 터보부호 응용 및 적용범위는 기존의 Viterbi 복호기와는 달리 처리할 데이터양과 연산 작용이 매우 복잡해 처리속도가 저속이어서 음성, 데이터 등의 저속 서비스에만 적용될 예정이며, 고속 데이터, 동영상 등의 고속 서비스로의 적용은 불가능한 실정이다. 따라서 하드웨어의 복잡도와 성능의 적절한 절충을 위한 계산량의 간소화와 터보 복호기의 적절한 설계 파라미터 설정 등의 과정이 요구되어진다. 이러한 문제점을 해결하기 위해서 본 논문에서는 두가지 방식의 decoding 알고리즘을 제시한다. 첫 번째 방식은 radix-4 decoding algorithm 이다. MAP기반의 Turbo-Code 복호시, Radix-2 방식은 임의의 k시점에서 복호할 때 k-1 시점에서의 순방향 state metric,  $\alpha_{k-1}^m$ , k+1 시점에서 역방향 state metric  $\beta_{k+1}^m$  그리고 k 시점에서 branch metric  $\delta_k^m$  를 이용하여 k 시점에서 "0" 과 "1"에 대한 Log Likelihood Ratio인 LLR을 구하여 1비트를 복호한다. 이에 반해 radix-4방식은 두 상태 이전인 k-2시점에서  $\alpha_{k-2}^m$ , 두 상태 이후인 k+2시점에서  $\beta_{k+2}^m$  를 구하여 k 시점에서 2 비트를 동시에 복호하기 때문에 radix-2 방식보다 속도가 2배 빠르며, 저장되는 메모리도 절반으로 감소할 수 있다. 또 다른 방식은 center to top algorithm이다. 기존의 방식에서 LLR을 계산하기 위해서 decoder는 BSM(or FSM) 계산과정이 끝나기를 기다려야 한다. 그러나 center to top 방식에서는 기다릴 필요 없이 FSM과 BSM을 동시에 계산하면서 FSM 과 BSM이 일정 시점을 지나면 decoder는 LLR을 계산하기 시작한다.

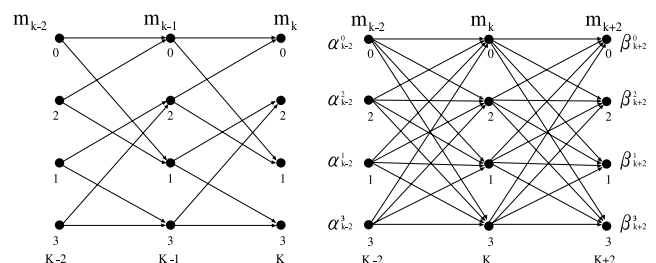
본 논문에서는 앞의 두 가지 방식과 병렬 decoding algorithm을 결합하였다. 제시하는 방식은 병렬화를 증가시킬 뿐만 아니라 decoder의 많은 메모리 요구량을 크게 감소시키기 때문에 고속 turbo 복호기 설계에 바람직하다. 고속 복호기를 테스트하기 위해 논문에 제시한 방식을 FPGA chip(Altera FLEX10K)에 구현하였다. 이 논문은 다음과 같이 구성되어 있다. 2장에서는 세 가지 유형의 고속 turbo 복호 알고리즘을 소개하고, 3장에서는 컴퓨터 시뮬레이션 결과를 분석하며, 4장에서는 구현을 위한 최적의 파라미터를 찾고 고속 turbo 복호기를 설계하며 5장에서는 결론을 말한다.

## High-Speed Decoding Algorithm

### Scheme 1 : Radix-4 algorithm

Turbo-Code 복호기를 고속화를 할 수 있는 방법은 고속 덧셈/곱셈 알고리즘을 적용하는 방법과 복호기 구조를 한 클럭에 한 비트를 복호하는 기존의 radix-2 방식이 아니라 한 클럭에 2비트를 복호하는 radix-4방식의 적용이 필수적이다. 따라서 본 연구 목표는 고속 덧셈/곱셈 알고리즘, radix-4 복호 방식을 이용하여 기존의 Turbo-Code 복호속도를 2배 이상으로 고속화시켜 MAP(Maximum A Posteriori)기반의 터보 복호기를 실현화하는 것이다. 이를 해결하면, 고속 데이터 전송 및 동영상 등을 포함한 무선 멀티미디어 서비스를 차세대 이동/위성 통신 단말기에 적용 가능하게 할 수 있다.

MAP기반의 Turbo-Code 복호시, Radix-2 방식은 임의의 k시점에서 복호할 때 k-1 시점에서의 순방향 state metric  $\alpha_{k-1}^m$ , k+1 시점에서 역방향 state metric  $\beta_{k+1}^m$  그리고 k 시점에서 branch metric  $\delta_k^m$  를 이용하여 k 시점에서 "0" 과 "1"에 대한 Log Likelihood Ratio인 LLR을 구하여 1비트를 복호한다. 이에 반해 radix-4방식은 두 상태 이전인 k-2시점에서  $\alpha_{k-2}^m$ , 두 상태 이후인 k+2시점에서  $\beta_{k+2}^m$  를 구하여 k 시점에서 2비트를 동시에 복호하기 때문에 radix-2 방식보다 속도가 2배 빠르며, 저장되는 메모리도 절반으로 감소할 수 있다. 과거 2단의 수신 비트를 입력 받아 한꺼번에 처리하는 Radix-4 방식의 trellis 구조는 Radix-2 방식의 2개 시점을 하나의 시점으로 간주하여 처리하며 Radix-2 방식에서 Radix-4방식으로의 trellis 구조 변경은 [그림 1]과 같다.



(a) 4-상태 Radix-2 trellis 구조

(b) 4-상태 Radix-4 trellis 구조

[그림 1] 4-상태 Radix-2, Radix-4의 trellis 구조

실제 MAP의 구현은 Log 차원에서 수행되므로, 두 상태를 하나의 상태로 간주하여 결합하는 radix-4 구조는 branch codeword와 수신된 비트와의

해밍거리를 구하는 viterbi 복호기의 구현 예처럼 단순하지가 않으며, Log차원에서 결합하기 때문에 복잡한 수식을 통하여 증명하고 난 뒤 구현해야 한다.

(a) Radix-4방식의 새로운 branch metric  $\delta_k^m$  계산방법

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

(b) Radix-4방식의 새로운 순방향 state metric  $\alpha_k^m$  계산방법

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

$b(p, m_k)$ 는 시점 k이고 m state에서 입력이 p일 때 S시점 k-2 만큼 뒤 쪽으로 향하는 state의 번호를 뜻한다.

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

(c) Radix-4방식의 새로운 역방향 state metric  $\beta_k^m$  계산방법

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

와 같이 나타낼 수 있으며,  $f(p, m_k)$ 는 입력이 p이고 state  $m_k$ 일 때 k+2 시점만큼 앞으로 향하는 state의 번호를 뜻한다.

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

(d) Radix-4방식의 Iteration시 새로운 APP  $\lambda_k^{p,m}$  계산방법

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

(e) Radix-4방식의 Iteration시 새로운 extrinsic  $Z_k$  계산방법

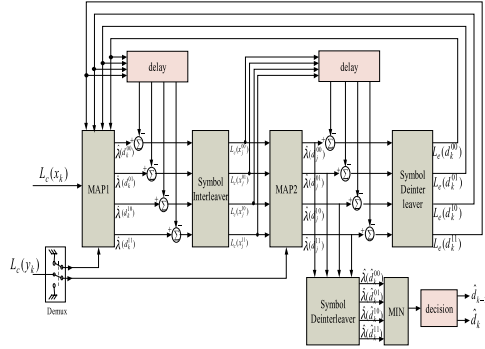
turbo iteration시 extrinsic 정보  $L_e(d_k^p)$ ,  $p = \{00, 01, 10, 11\}$ 을 첫번째 MAP에서의 branch metric  $\delta_k^{p,m}$ 은 나타낼 수 있다.

$$\delta_k^{p,m} = K_k \exp\left(-\frac{2}{\sigma^2}(\mathcal{X}_{k-1}d_{k-1} + y_{k-1}Y_{k-1} + \mathcal{X}_k d_k + y_k Y_k)\right) \times L_e(d_k^p)$$

(f) Radix-4방식의 복호비트를 결정하기 위한 새로운 LLR 계산방법

$$LLR(D_k) = \text{MIN} \left\{ \sum_m \lambda_k^{00}(m), \sum_m \lambda_k^{01}(m), \sum_m \lambda_k^{10}(m), \sum_m \lambda_k^{11}(m) \right\}$$

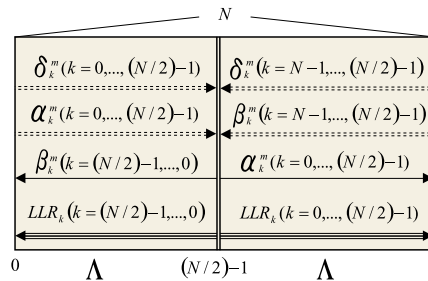
따라서, Radix-4 구조의 복호기는 다음 [그림 2]와 같다.



[그림 2] Radix-4 기반의 터보 복호기 구조

**Scheme 2 : CT(Center to Top) algorithm**

한 클럭에 두 비트를 복호하는 radix-4 알고리즘과는 달리 1 비트만 복호 하면서도 복호지연을 절반으로 감소하는 알고리즘을 제시한다. 본 장에서 제시한 알고리즘은 Turbo 복호기의 구현시 문제점은 메모리 크기 및 복호지연이 가장 큰 문제점인데 메모리 크기에 비례하여 복호지연이 발생하므로 메모리를 절반으로 줄이면서 복호지연 및 복호속도를 2배 향상시킨 알고리즘이다. [그림 3]은 본 연구에서 제시할 고속 turbo 복호기의 복호과정에 대한 그림이다.



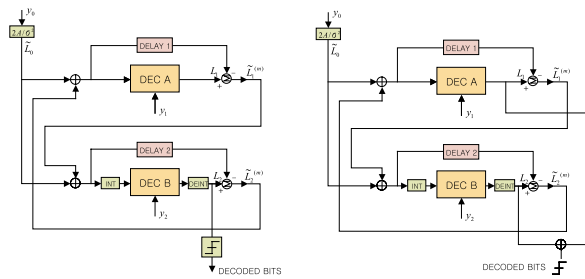
[그림 3] 고속 turbo 복호기 복호과정

[그림 3]에서 알 수 있듯이 순방향 상태 메트릭  $\alpha$ 와  $\beta$ 를 동시에 구한다면 메모리 절반 위치에 있을 때 중간 지점부터 LLR을 구하면서 복호를 한다. 그러므로 복호시점이 역방향 상태 메트릭을 모두 구한 다음 복호를 하는 기존의 방식과는 달리 복호 메모리의 절반 위치에서 복호를 하기 때문에 복호지연이 기존의 방식에 비해 1/2 감소한다.

**Scheme 3 : Parallel Decoding Algorithm**

터보 부호의 복호기는 복호화 알고리즘으로 MAP 알고리즘이나 SOVA 알고리즘을 이용하는 2개의 SISO(soft input soft output)복호기가 직렬 모드로 연결되어서 반복적으로 정보들을 서로 교환하는 구조이다. 복호기의 가장 큰 문제점은 첫 번째 복호기(MAP1)에서 복호하고 난 뒤 두 번째 복호기에서 복호하는데 이를 병렬로 처리할 수 없다는 것이 가장 큰 단점이다.

터부호등과 같이 연판정 입출력 값을 이용하여 직렬적으로 구성된 복호기 구조에서는 반복을 위해서 첫번째 복호기에 해당하는 복호기를 이용하여 복호하고 난 뒤에 첫 번째 복호기에 나온 연판정 값을 이용하여 두 번째 복호기에서 데이터를 복호하고 이의 과정을 여러 번 반복하면서 성능을 향상시킨다. 따라서 두 번째 복호기는 항상 첫 번째 복호기가 복호를 하고 난 후에 그 값을 이용해야 하기 때문에 지연으로 인한 속도 저하를 가져올 수 있다. Turbo 부호의 복호기를 병렬로 구성하여 속도를 고속화시키는 알고리즘이 최근에는 발표되고 있으나, 성능이 약간 감소하고 인터리버/디인터리버로 인한 지연은 개선시킬 수가 없다는 단점이 있다. 직렬 모드 복호기가 하나의 출력 시퀀스 생성하는 반면 본 연구에서는 그림4와 같이 병렬 모드 복호기는 동시에 2개의 출력 시퀀스를 제공하는 형태이다. 일반적인 병렬 모드는 두 번째 복호기(MAP2)의 LLR값을 강판정하여 복호하는 형태지만 그림4 오른쪽(Parallel Sum)은 첫 번째 복호기(MAP1)와 두 번째 복호기(MAP2)에서 나온 연판정 값을 합해서 복호를 한다.

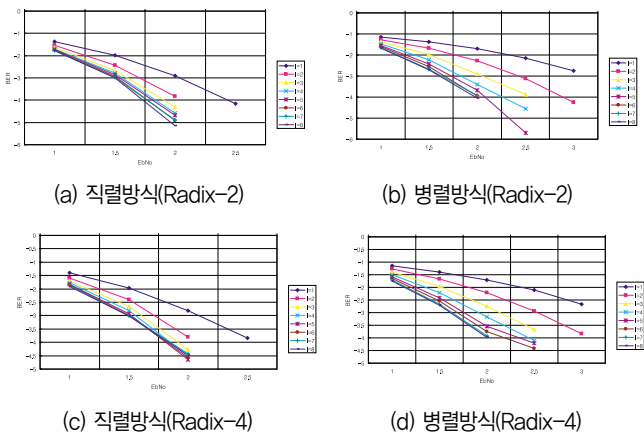


(a) 일반적인 병렬 (b) Sum을 이용한 병렬구조

[그림 4] 병렬모드 터보 복호기 구성

### Computer Simulation Results

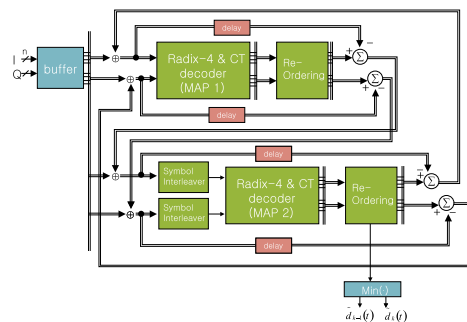
이번 장에서는 세 가지 알고리즘을 결합한 성능을 분석하였다. 제시한 알고리즘과 기존의 복호기의 성능 비교를 위해서 [그림 5]에서는 8상태 다양한 반복횟수를 두어 시뮬레이션 하였다. 아래 [그림 5]는 상태수 8, N=500, 그리고 Golden interleaver를 적용시키고 반복횟수에 따라 직렬구조와 일반적인 병렬구조에 대한 성능 분석한 그래프이다. [그림 5]로부터 제시된 방식의 성능은 기존의 방식과 거의 일치함을 증명할 수 있다. 따라서 제시한 방식의 복호기는 기존방식의 복호기와 거의 같은 성능을 가지면서 상당한 복호 지연을 줄일 수 있음을 결론지을 수 있다.



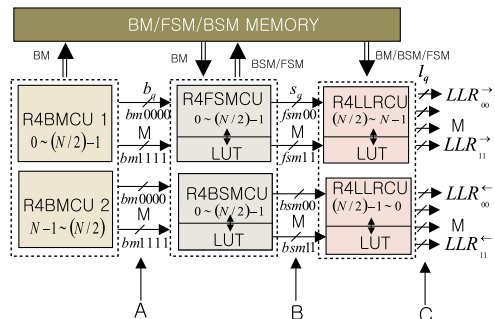
[그림 5] 성능곡선

### Design of High-Speed Turbo Decoder

[그림 6](a)은 구현을 위한 고속 turbo 복호기의 schematic diagram이다. [그림 6](b)는 [그림 6](a)에서 Radix-4 & CT decoder(MAP1, MAP2)의 세부적인 블록도이다. [그림 6](a)에서 MAP1과 MAP2는 병렬로 동작하며 MAP1의 출력은 그림 6(b)에서의 LLR(log likelihood ratio)이다. 예를 들어 LLR<sub>00</sub>은 입력 비트 "00"의 log-likelihood ratio를 의미한다. 그리고 화살표 "→"는 LLR 계산 방향을 표시한다. 다음 iteration시 외부정보(extrinsic information)를 정확히 수신 심벌에 첨가하기 위해서 복호기는 Re-ordering 블록이 필요하다. Re-ordering 블록은 LIFO(Last-In First-Out) 나 FIFO(First-In First-Out)처럼 LLR을 정렬한다. [그림 6](b)에서 보듯이, 복호기는 다음의 4가지 주요한 블록들로 구성된다. R4BMCU(Radix-4 Branch Metric Unit), R4BSMCU(Radix-4 Backwards State Metric Unit), R4FSMCU(Radix-4 Forwards State Metric Unit), R4LLRCU(Radix-4 for LLR Calculator Unit). 여기서 R4는 앞의 2장에서 설명한 Radix-4 알고리즘을 의미한다. 2장에서 말했듯이 고속 turbo 복호기에서는 4개의 R4BMCUs, 두개의 R4FSMUs/R4BSMUs 그리고 4개의 R4LLRCUs가 필요하다. 수신 심벌 전체를 저장한 후 양자화 된 I,Q의 샘플 들은 R4BMCU로 이동한다. bm0000은 branch codeword 와 수신 심벌 사이의 branch metric을 의미한다. R4BMCU는 수신 데이터중의 4 샘플의 branch metric을 계산한다. CT 알고리즘을 적용하기 위해 R4BMCU1은 왼쪽에서 오른쪽 방향으로 branch metric을 계산하고, R4BMCU2은 오른쪽에서 왼쪽 방향으로 branch metric을 계산한다. 각각의 branch metric은 R4BSMCU/R4FSMCU 계산을 위해 버퍼에 저장한다. 0에서 N/2-1(or N-1에서 N/2)간격의 FSM(or BSM)계산과정에서 branch metric 연산이 필요 없다. 단지 R4BMCU2(or R4BMCU1)의 메모리를 참조한다. R4FSMCU, R4BSMCU, 그리고 R4LLRCU 블록에서는 연산을 위해 E-table이 필요하다. 이 논문에서는 외부의 ROM을 별도로 사용하지 않고 internal ROM(EAB, Embedded Array Block)을 이용하여 E-table을 구현하였다.



(a) Block diagram of high-speed turbo decoder



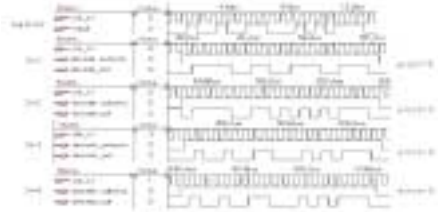
(b) Structure of Radix-4 and CT decoder

[그림 6] Schematic diagram of proposed turbo decoder

고속 turbo 복호기를 구현하기 위해 [그림 6]의 각각의 블록의 최적의 양자화 비트를 결정해야 한다. 즉 demodulator 출력, in-phase 신호 그리고 quadrature 신호, R4BMCU 출력  $b_p$ , R4BMCU의 출력  $S_p$ , 그리고 log-likelihood ratio  $L_p$ . fixed point 컴퓨터 시뮬레이션을 통해 demodulator의 추력은 8bit로 양자화 하였다. turbo 복호기 내부 파라미터는 9비트로 양자화 하였다. 시뮬레이션에 의한 turbo 복호기의 최적의 양자화 비트는 다음 표1에 나열하였다.

[표 1] turbo 복호기 최적의 양자화 비트

I,Q(n)	8,8
bq	9
Sq	9
lq	9



[그림 7] 반복 회수에 따른 복호과정의 파형 (Eb/No=1[dB])

[그림 7]은 고속 turbo 복호기에서의 timing simulation 결과이다. 여기서 Flex10k1000GC503-4의 embedded 메모리의 용량으로 인해 interleaver의 크기 N은 64로 고정하였다. [그림 7]의 결과에 따르면 반복회수의 증가에 따라 에러가 정정됨을 확인할 수 있다. 기존 방식의 복호기와 제시한 고속 turbo 복호기의 복호 속도를 비교하기 위해 고속 복호기와 같은 절차로 기존 방식의 복호기를 구현하였다. [표 2]는 기존 방식과 제시된 고속 복호기의 각 단계별 복호 속도이다. 기존 방식과 제시한 방식의 main 클럭은 각각 44ns,58ns이다. 제시된 방식은 더 많은 logic과 메모리를 요구하기 때문에 제시된 알고리즘에서는 클럭의 주기가 더 길어야 한다. 제시된 방식의 총 연산 시간은 27 $\mu$ s이며 이것은 기존의 방식보다 약 5배 정도의 속도를 보임을 확인할 수 있다.

[표 2] 기존 방식과 제시된 방식의 각 단계별 복호 속도

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

### Conclusion

실시간 서비스의 Turbo 복호기의 적용 분야를 넓히기 위해서 가장 중요한 문제는 복호과정의 지연시간을 줄이는 것이다. 이 논문에는 3가지의 알고리즘(Radix-4, Center to Top, Parallel)을 제시한다. 또한 어느 정도 복호지연이 감소되는지 확인을 위해 3가지 다른 알고리즘을 결합하여 구현하였다. interleaver의 크기는 64, 반복회수는 1로 고정하여 FLEX10k1000GC503-4에 구현해 본 결과, 기존의 방식보다 제시한 방식의 복호 속도가 매우 뛰어나 증명하였다. 또한 제시한 방식의 성능역시 기존 방식보다 거의 열화없음을 보였다. 고속 무선 통신에서 터보 복호기의 지연 및 연산의 복잡성으로 고속 무선통신에서의 적용이 불가능 한데 본 연구에서는 고속 및 저복잡

도 알고리즘을 이용하여 이를 개선하였다. 본 연구에 적용한 고속 및 저복잡도 알고리즘은 세 가지로 요약 할 수 있으며, 기존의 터보코드는 수신단에서 N개의 수신비트를 저장한 후 복호를 시작하므로 항상 N 비트 만큼의 복호 지연을 가지므로 Center-to-Top 방식으로 복호지연시간을 N/2만큼 단축하고, ② 기존의 방식은 한 클럭에 1비트를 복호하는 구조를 가지는데 반해 본 연구에서는 Radix-4 방식으로 한 클럭에 2비트의 정보를 복호하므로 복호 속도를 2배 향상시켰으며, ③ 기존의 방식은 두개의 복호기를 직렬 모드로 구성하므로 첫 번째 복호기의 수행후 두 번째 복호기가 수행됨으로 지연을 가져오므로 본 연구에서 병렬 구조를 이용하여 두개의 복호기를 동시에 사용함으로써 지연 감소시켰다. 본 연구에서 제시한 적응형 고속 터보복호기는 이론상으로 기존의 방식에 비해 고속화가 가능하다. (Radix-4 적용시 2배, Parallel 구조 설계시 2배, CT 방식 적용시 2배 정도가 기존의 방식에 비해 고속화가 가능하므로, 약 30배 정도의 고속화가 가능하다.) 위의 세 가지 방식을 결합하여 interleaver의 크기는 64, 반복회수는 1로 고정하여 FLEX10k1000GC503-4에 구현해 본 결과, 기존의 방식보다 제시한 방식의 복호 속도가 약 5배 가량 뛰어남을 보였다. 따라서 본 연구에서 제시한 적응형 고속 터보 복호 알고리즘 및 성능분석을 토대로, 기존 방식 보다 수십 배가 고속인 복호기의 개발이 가능하며, 이는 향후 적응형 시스템 및 무선 멀티미디어 통신(DMB, DVB-RCS등) 에로의 적용이 가능하다.

### Acknowledgement

This work was supported by IT SoC Promotion Group, Korea IT Industry Promotion Agency(Human Resource Development Project for IT SoC key Architect).

### [참고문헌]

- [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 ofTurbo 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, September1997.
- [6] S.S. Pietrobon, "Implementation and performance of a serial MAP decoder forin an iterative turbo decoder," in Proc., IEEE Int. Symposium on InformationTheory, 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 CommunicationsDecember 1997.
- [9] D. Divsalar and F. Pollara, " Multiple Turbo Codes for Deep-SpaceCommunications," 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 onParallel Concatenated Coding Schmes," IEEE Transaction on Informationvol.42, no. 2,pp.409-429, March 1996.