

터보복호기를 위한 SOVA 복호기의 설계

김기보, 김종태

성균관대학교 전기전자 및 컴퓨터 공학부 컴퓨터 시스템 합성연구실

VLSI Design of SOVA Decoder for Turbo Decoder.

Ki Bo Kim, Jong Tae Kim

School of Electrical and Computer Engineering, SungKyunkwan University

Abstract - Soft Output Viterbi Algorithm is modification of Viterbi algorithm to deliver not only the decoded codewords but also a posteriori probability for each bit. This paper presents SOVA decoder which can be used for component decoder of turbo decoder. We used two-step SMU architectures combined with systolic array traceback methods to reduce the complexity of the design. We followed the specification of CDMA2000 system for SOVA decoder design.

1. 서 론

3세대 디지털 통신 시스템인 IMT-2000 시스템에서는 음성통신 뿐만 아니라, 데이터 통신, 영상통신 서비스까지 함께 제공한다. 따라서 정보의 전송량이 크며 보다 빠른 데이터 처리 능력이 요구된다. IMT-2000 시스템에서는 채널코덱으로 IS-95a 시스템에서 사용되는 길쌈 부호와 함께 터보부호를 사용한다. 터보부호는 구현의 어려움과 time delay가 크다는 단점에도 불구하고, 그 높은 오류정정 능력으로 인해 IMT-2000 시스템에 채택되었다. [1]에 따르면 프레임 크기를 65536 비트로 하여 18회 반복 복호한 결과 10^{-5} BER을 Eb/NO 0.7 dB에서 얻어내는 등 터보 부호는 이론상 Shannon 한계에 근접한다고 알려져 있다.

터보 복호기는 복호 알고리즘에 따라 MAP(Maximum A Posteriori)과 SOVA(Soft Output Viterbi Algorithm)를 이용하여 구성할 수 있다. MAP알고리즘은 SOVA에 비해서 0.7dB 정도의 성능향상이 있으나[2], 구현에 있어서는 SOVA 보다 더 복잡하다.

본 논문에서는 IMT-2000 시스템용 터보복호기에 적용 가능한 SOVA 복호기의 설계에 대해서 설명한다. SOVA 복호기의 설계 사양은 동기식 IMT-2000 시스템의 표준안인 CDMA2000 시스템의 사양을 따랐다. 설계한 SOVA복호기는 systolic array 구조[3]와 two-step 아키텍처[4]를 사용하여 복잡도를 줄였다.

2. 본 론

2.1 Soft Output Viterbi Algorithm

SOVA는 Viterbi 알고리즘에서 soft output이 가능하도록 수정된 알고리즘이다. 터보복호기의 반복복호를 위해서는 soft output 값이 반드시 필요하다. 여기서 soft output 값은 복호 신뢰도라고 생각할 수 있다. 즉 SOVA복호기에서 복호한 결과가 어느 정도 정확한지를 나타내는 척도이다.

SOVA는 기본적으로 비터비 알고리즘을 통해서 복호를 하며 다음과 같은 연산을 추가하여 soft output 값을 구한다.

k 시간에서 경로 m 의 확률을 다음과 같이 쓸 수 있다.

$$P_k^m = C e^{M_k^m}$$

여기서 C 는 일정한 상수 값이며 M_k^m 은 경로값(path metric)이다. AWGN 채널이고 부호율이 $1/N$ 일 때 경로값 M_k^m 는 다음과 같이 연산된다.

$$M_k^m = M_{k-1}^m + \sum_{n=1}^N x_{k,n}^m L_c y_{k,n} + x_{k,1}^m L(u_k)$$

여기서 $x_{k,n}^m$ 은 경로 m 에서의 N 비트 발생부호 중 n 번째 비트, $y_{k,n}$ 은 k 시간에 채널로부터 수신한 n 번째 수신 신호, L_c 는 채널 신뢰도 ($= \frac{4E_s}{N_0}$)를 나타낸다.

P_k^m 을 생존경로(survivor path)와 다투는 경쟁경로(competing path)의 확률값, M_k^m 을 경쟁경로의 경로값이라고 하면 잘못된 경로 즉 경쟁경로를 선택할 확률은 다음과 같다.

$$\Phi_k^m = \frac{P_k^m}{P_k^m + P_k^{m'}} = \frac{1}{1 + e^{(M_k^{m'} - M_k^m)}}$$

올바른 경로를 선택할 확률과 잘못된 경로를 선택할 확률간의 log likelihood ratio는 다음 식으로 결정된다.

$$\Delta_k^m = \log \frac{1 - \Phi_k^m}{\Phi_k^m} = (M_k^m - M_k^{m'})$$

즉 생존경로와 경쟁경로의 경로값들의 차이가 LLR 값이 된다. 이때 Δ_k^m 이 $+\infty$ 에 가까울수록 경로 m 이 올바른 경로가 될 확률이 높고 $-\infty$ 에 가까울수록 경로 m' 이 올바른 경로가 될 가능성이 높다고 말할 수 있다. LLR 값 Δ_k^m 으로부터 출력할 신뢰도 값인 $L(u_k)$ 을 얻기 위해서는 Δ_k^m 값을 갱신(update) 과정이 필요하다. 갱신 규칙은 다음과 같다.

$$\text{for } (j=k : j \leq k+U : j++)$$

$$\{ \text{if } (u_j^s \neq u_j^c) L_j^s = \min(L_j^s, \Delta_k^m) \}$$

여기서 u_j^s 는 생존경로상의 j 번째 복호어, u_j^c 는 경쟁 경로상의 j 번째 복호어, L_j^s 는 j 번째 출력 복호어의 신뢰도값, U 는 갱신깊이(update depth)를 말한다.

갱신 규칙에서 보듯이 생존경로와 경쟁경로를 각각 비교하여 다른 복호어 값을 갖는 구간에 대해서만 갱신을 수행한다. 또, 하드웨어 구현을 좀더 간단하게 하기 위해서 U 라는 갱신깊이 만큼의 윈도우를 사용하여 연산량을 줄인다. 복호성능에 영향을 미치는 않는 갱신 깊이는 일반적으로 구축장 크기의 2배로 알려져 있다.[5]

2.2 Design architecture

2.2.1 Two Step SOVA

SOVA는 비터비 알고리즘에 따라 지로값(branch metric) 및 경로값(path metric)의 계산, 최우 복호경로의 판별/부호등의 연산과 함께 연판정값 출력을 위해

필요한 생존경로와 경쟁경로 사이의 차이값의 계산, 차이값의 갱신 등의 연산이 필요하다. 여기서 주목할 점은 차이값의 갱신은 생존경로에 대해서만 필요하며 나머지 경로들에 대한 차이값 갱신은 연판정 출력값에는 영향을 미치지 않는다. 따라서 필요한 갱신연산의 수를 줄이고 큰 비트 크기를 갖는 차이값들의 wiring에 따르는 회로 크기의 증가를 막기 위해서 two-step 방식으로 갱신연산을 수행한다(4). two-step 방식은 먼저 생존경로를 판별한 후 생존경로에 대해서만 신뢰도값을 갱신한다. 따라서 연산결과에 영향을 주지 않는 나머지 경쟁경로들에 대한 갱신연산을 생략하여 연산량을 줄일 수 있다. 그림 1에 SOVA 복호기에서 two-step 연산을 수행하는 SMU부분의 블록도를 보였다.

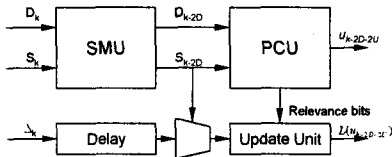
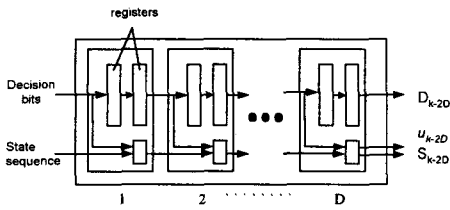


그림 1 two-step SOVA-SMU architecture

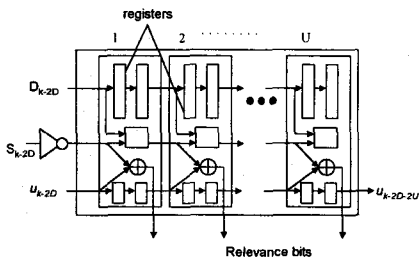
Two-step SOVA 구조를 통해서 회로크기를 40% 정도 줄일 수 있다.[4] 하지만 생존경로의 복호와 갱신연산을 동시에 하지 못하고 순차적으로 수행해야 하므로 그 만큼 연산의 지연(latency)은 더 커진다.

2.2.2 Systolic array

비터비 복호기에서는 역추적 복호법의 한가지 방법으로 systolic array 구조를 사용한다. 본 연구에서는 two-step SOVA의 복호부(SMU)와 경로비교부(PCU)의 설계에 systolic array 구조를 이용하였다. SMU와 PCU의 세부 블록도는 그림 2와 같다.



(a) SMU



(b) PCU

그림 2 SMU와 PCU의 블록도

2.3 설계 사양

본 연구에서는 SOVA 복호기의 설계사양으로 동기방식 IMT-2000 시스템의 표준안인 CDMA2000의 사양을 따랐다.

표 1에는 설계한 SOVA 복호기의 설계 사양을 보였다.

구축장 크기	4
부호율	1/3
생성다항식	(13, 15, 17) ₈
연판정 비트수	3bit (8단계)
신뢰도값의 양자화단계	16단계
decoding depth	20
update depth	8
path metric bit	9 bit

표 1 SOVA복호기의 설계사양

CDMA2000 시스템에서는 가변 부호율을 사용한다. 즉 1/2, 1/3, 1/4등의 부호율이 사용되는데 같은 부호기를 사용하면서 출력단의 천공(puncturing) 패턴만 다르게 하여 다른 부호율을 만든다. 따라서 본 연구에서는 천공패턴과 무관하게 적용 가능한 부호율인 1/3로 설계하였다.

2.4 SOVA 복호기의 설계

그림 3에는 설계한 SOVA 복호기의 블록도를 보였다. 점선으로 표시된 부분은 SOVA 복호기에 포함된 비터비 복호 블록에 해당된다.

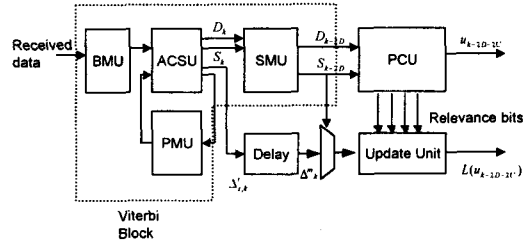


그림 3 SOVA 복호기의 전체 블록도

BMU (Branch Metric Unit)에서는 지로값을 계산한다. BMU에 입력되는 값은 채널로부터 수신한 신호 $y_{k,1}, y_{k,2}, y_{k,3}$ 와 이전 단계의 복호결과로 얻은 신뢰도값 ($L(u_k)$)이다. 여기서 $y_{k,1}$ 은 systematic 부호어이며 $y_{k,2}, y_{k,3}$ 는 parity 부호어 이다. 각각의 연판정 단계는 8단계로 3비트씩의 데이터 크기를 갖는다. 신뢰도값 $L(u_k)$ 는 [5]에 따르면 3비트 8단계의 양자화 단계면 충분하다고 하나 본 설계에서는 보다 정확한 연산을 위해서 4비트 16단계로 설계를 하였다. 부호율 1/3일 때 발생 가능한 부호어 ($x_{k,1}^m, x_{k,2}^m, x_{k,3}^m$)의 종류는 모두 8가지(000, 001, ..., 111)이다. 따라서 $x_{k,1}^m, x_{k,2}^m, x_{k,3}^m$ 와 $y_{k,1}, y_{k,2}, y_{k,3}$ 사이의 지로값의 가지수도 모두 8가지가 된다. BMU에서는 이상의 8가지의 지로값을 구한 다음 해당되는 ACS 블록에 필요한 지로값을 전달해준다.

ACSU(Add Compare Select Unit)에서는 BMU에서 전달받은 지로값과 PMU에 누적된 경로값을 합하여 새로운 경로값을 구한다. 한 state로 수렴하면서 경쟁하는 두 경로 중 보다 큰 경로값을 갖는 경로를 생존 경로로 결정하고 생존경로의 경로값을 PMU에 누적시킨다. 또 결정된 생존경로에 대한 경로정보를 SMU에 전달한다. 경로값은 지속적으로 누적되므로 overflow를 막기 위해서 일정한 정규화(normalization) 연산이 필요하다. 본 설계에서는 [6]에 정리된 modulo metric rescaling 방법을 사용하였다. 이 방법은 별도의 정규화 블록이 필요하지 않아서 회로가 더 간단해 지고 정규화에 필요한 연산을 크게 줄일 수 있는 이점이 있다.

ACS는 완전 병렬형 구조를 사용하였다. 구축장의 크기가 작기 때문에 state 수가 $2^{4-1} = 8$ 개이므로 완전 병렬형 구조로 설계해도 회로의 크기를 많이 차지하지 않으며 복호속도를 빠르게 할 수 있기 때문이다.

PMU(Path Metric Unit)는 경로값을 저장하고 있는 레지스터들로 이루어져 있다. PMU의 데이터 비트 수는 (7)에 제시된 계산식에 따라서 부호 비트를 포함해 9비트로 구성하였다.

SMU는 2.2.1에서 설명한대로 systolic array 구조를 사용하여 설계하였다.(그림 2(a)) decoding depth는 복호경로가 수렴하기에 충분하다고 알려진 구축장의 5배 크기로 설계하였다.[8]

PCU(Path Comparison Unit)는 SMU에서 결정된 생존경로와 그 경쟁경로에 대해서 비교를 수행하는 블록이다. PCU역시 systolic array 구조를 사용하였다. PCU에서는 경쟁경로에 대해서 역추적이 이루어지며 역추적 결과로 얻은 경쟁경로상의 j 번째 복호어 u_j^k 가 생존경로상의 j 번째 복호어 u_j^s 와 다를 때 신뢰도 값의 갱신 작업이 필요하며 이 갱신 작업의 수행을 지시하는 relevance bit 신호를 Update Unit에 인가하게 된다. Update unit에서는 PCU에서 받은 relevance bit 신호에 따라 해당하는 신뢰도 값의 갱신 작업을 수행한다. Update unit은 PCU와 같이 pipeline 형태로 이루어져 있다. 갱신 깊이는 구축장 크기의 2배인 8을 사용하였다. 따라서 PCU와 Update unit은 모두 8개의 소블럭이 pipeline 형태로 나열된 구조이다. Update unit의 세부 블록도를 그림 4에 보였다. UPE는 relevance bit에 신호가 인가되면 다음 연산을 수행한다.

$$L_j^s = \min(L_j^k, \Delta_k^m)$$

즉 UPE의 두 입력값 (L_j^k, Δ_k^m)을 비교하여 더 작은 값을 다음 UPE로 전달한다. Update unit의 최종 출력값은 SMU에서 연산되어 출력된 복호어의 신뢰도 값 $L(u_k)$ 이다. 이상과 같은 동작을 통해서 SOVA 복호기는 복호어 u_k 와 신뢰도값 $L(u_k)$ 를 출력한다.

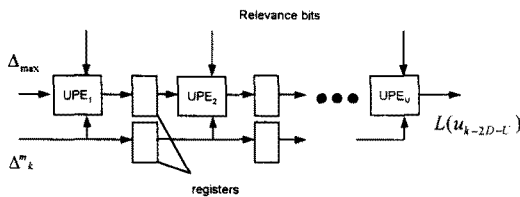


그림 4 Update unit

2.5 회로의 논리합성

설계한 SOVA 복호기에 대해서 논리 합성과 동작검증을 수행하였다. 표 2는 two-step, systolic array 구조를 갖는 SOVA 복호기를 표 1에서 제시한 사양으로 설계한 후 논리 합성한 결과이다. 설계는 VHDL을 이용하여 있으며 회로의 동작검증은 SynopsysTM의 Vhdl System Simulator (VSS)를 사용하였다. 논리합성을 위해서 SynopsysTM의 Design Compiler를 사용하였으며 Lsi-10K 라이브러리를 사용하였다. 비교를 위해서 systolic array 구조로 설계된 같은 사양을 갖는 비터비 복호기의 합성결과를 함께 제시하였다. 그림 5에는 합성한 SOVA 복호기의 합성결과를 보였다.

구분	비터비복호기	SOVA 복호기
게이트수	13626	38081

표 2 논리 합성결과

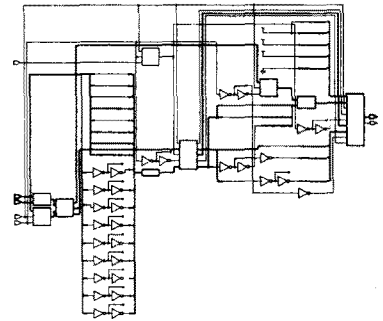


그림 5 논리 합성된 SOVA 복호기

3. 결론

본 연구에서는 IMT-2000 시스템에서 채널코덱으로 사용되는 터보복호기에 응용 가능한 SOVA 복호기를 설계하였다. 설계된 SOVA 복호기는 two-step 아키텍처와 systolic array 구조를 사용하여 회로의 크기를 줄였으며 IMT-2000 시스템의 동기식 표준안인 CDMA2000 사양에 맞추어 설계하였다. 설계한 SOVA 복호기는 초기의 지연시간이 지난 후 매 클럭마다 복호어와 복호 신뢰도 값을 출력할 수 있다. 터보복호기는 회로의 복잡도가 매우 큰 편이다. 따라서 복호기의 복호성능을 저하시키지 않는 한계 내에서 회로의 크기를 최소화하는 설계 방법을 찾아내는 연구가 필요하다.

[참고 문헌]

- [1] C.Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-Codes", ICC'93, Geneva, Switzerland, pp. 1064-1070, May 1993.
- [2] Papke, L., Robertson, P. and Villebrun, E., "Improved decoding with the SOVA in a parallel concatenated (turbo-code) scheme," Proc. of ICC '96, Dallas, TX, USA, June, 1996, pp. 102-106.
- [3] T.K. Truong, M.T. Shih, I.S. Reed, E.H. Satorius, "A VLSI Design for a Trace-Back Viterbi Decoder", IEEE Trans. Comm., vol. 40, No. 3 pp. 616-624, March 1992.
- [4] O. Joeressen, M. Vaupel, H. Meyr, "High-Speed VLSI Architectures for Soft-output Viterbi decoding", in Proc. Int. Conf. Applicat. Specific Array Processors, Oakland, CA : IEEE Computer Society Press, Aug. 1992, pp. 373-384.
- [5] O. Joeressen, M. Vaupel, H. Meyr, "Soft-Output Viterbi Decoding : VLSI Implementation Issues", Proceedings of the 1993 IEEE 43th Vehicular Technology Conference, 1993.
- [6] A.P.Hekstra, "An Alternative to Metric Rescaling in Viterbi Decoders", IEEE Trans. on Comm., Vol. 37, NO. 11, November 1989.
- [7] R. Burger, G. Cesana, M. Paolini, M. Turolla, S. Vercelli, "A Fully Synthesizable Parameterized Viterbi Decoder", in IEEE Custom Integrated Circuits Conference, pp. 27-30, 1999.
- [8] I. Onyszchuk, "Truncation Length for Viterbi Decoding", IEEE Trans. on Comm., Vol. 39, pp. 1023-1026, July 1991.