

3GPP 규격의 터보코드 복호를 위한 SOVA 복호기의 하드웨어 구현

정회원 김 주 민*, 고 태 환, 정 덕 진*

VLSI implementation of a SOVA decoder for 3GPP complied turbo code using FPGA

Jumin Kim*, Taehwan Ko, Duckjin Chung* *Regular Members*

요 약

차세대 멀티미디어 이동통신인 IMT-2000의 규격에서는 3GPP와 3GPP2에서 모두 터보 코드를 채널 코덱으로 채택하고 있다. 그 중 3GPP에서는 용도에 따라 길쌈부호와, 제한길이 4인 1/3 터보코드를 선택적으로 사용하도록 정의 되어 있다. 터보코드는 복호기의 출력으로 경관정 복호 비트에 대한 신뢰도 값을 동시에 생성하여, 이를 이용한 반복복호로 우수한 BER 특성을 얻을 수 있어야 한다. 본 논문에서는 먼저 3GPP 규격의 터보 복호기에 적용할 수 있는 내부 복호기로서 SOVA 복호기를 설계하였다. 또한 터보 복호기에서의 연관정 출력값의 중요성을 감안하여, 누적메트릭 정규화에 있어서 신뢰도 값에 영향을 주지않는 구조를 제안하여 적용하였다. 본 연구에서는 효율적인 구조의 3GPP SOVA 복호기를 설계하기 위하여 C++를 이용하여 알고리즘에 대한 성능을 검증하였으며, 이를 기반으로 VHDL을 이용하여 복호기를 설계하였다. 마지막으로 Altera사의 EPF10K100GC503 FPGA를 이용하여 복호기를 하드웨어로 구현하였다.

ABSTRACT

According to the IMT-2000 specification of 3GPP(3rd Generation Partnership Project) and 3GPP2, Turbo codes is selected as a FEC(forward error correction) code for even higher reliable data communication. In 3GPP complied IMT-2000 system, channel coding under consideration is the selective use of convolutional coding and Turbo codes of 1/3 code rate with 4 constraint length. In order to design a low complexity and high performance SOVA decoder for Turbo Codes, we suggest a new path metric normalization. Further more, we analyze the decoding performance with respect to update depth and find out the optimal value of it by using computer simulation. Based on the simulation result, we designed a SOVA decoder using VHDL and implemented it into the Altera EPF10K100GC503 FPGA.

I. 서론

1993년 Berrou, Glavieux, Thitimajshima^[1]이 터보코드를 제안한 이래 터보코드는 샤논의 한계에 근접하는 오류정정 성능을 보임으로써 세계적으로 많은 연구가 진행되고 있다. Berrou의 논문에 따라

면, 1/2 코드에서 $E_b/N_0=0.7\text{dB}$ 에서 10-5의 비트오율(bit error rate)을 보였다. 디지털 통신에 있어서 수신부에서 받는 신호는, 랜덤 잡음, 페이딩, 다중경로에 따른 위상 왜곡등으로 인하여 랜덤 및 군집 오류를 포함하여 수신되게 된다. 따라서 이러한 오류를 감지, 정정하기 위한 코딩기법이 계속 연구되어 왔으며, 이러한 과정에서 터보코드가 탄생되었다.

* 인하대학교 전자전기컴퓨터공학부 집적회로연구실(joshua1k@dreamwiz.com)

논문번호 : 010085-0427, 접수일자 : 2001년 4월 27일

※ 본 연구는 2000년도 시스템 IC 2010 사업의 지원으로 수행하였음.

현재 IMT 2000은 일본과 유럽을 중심으로 한 3GPP(W-CDMA) 규격과 북미를 중심으로 한 3GPP2(cdma2000)규격으로 나뉘어져 진행되고 있으며 이 두 규격 모두에서 길쌈부호와 터보부호를 병행하여 채택하고 있다. 이러한 목적에 터보코드를 사용하기 위해서는 반복부호 알고리즘의 최적화가 필수적이다. 현재까지 터보코드의 복호에 관한 많은 연구가 진행되어 왔으며, 터보 코드의 복호를 위해서 사용되는 알고리즘은 MAP(Maximum a posteriori)방법과 SOVA(soft output viterbi algorithm)이다. 이중 SOVA는 MAP보다 비교적 낮은 복잡도를 가지기 때문에 하드웨어를 구현하는데 있어서 좀더 용이하다.^[1]

SOVA 알고리즘은 1989년 Hagenauer가^[2] 연판정 출력을 생성시킬수 있도록 제안한 이래로 1995년 Joecressen 등이 16상태 길쌈부호의 복호기로서, 고속 SOVA복호기를 설계하였다.^[3] 그 이후 하드웨어 설계에 관한 연구도 많이 진행되고 있으며 특히 복호기 내부 모듈의 저전력화, 저복잡도를 위한 연구가 꾸준히 진행되고 있으나, IMT-2000 규격의 터보복호기에 적용할 수 있는 SOVA복호기 전체의 구현은 거의 전무한 실정이다. 본 연구에서는 3GPP 규격의 (13,15) 1/3 터보 코드를 복호하기 위한 터보복호기의 핵심 부분인 SOVA 복호기를 레지스터 교환 방식을 기반으로 설계 하였다. 또한 신뢰도값을 갱신하는 단계를 이용하는 SOVA의 경우에 신뢰도 갱신 길이는 추가 latency를 야기시키고^[3], 하드웨어의 복잡도를 증가시키므로 이의 최적화 역시 필요하다. 따라서 본 논문에서는 AWGN(Additive White Gaussian Noise) 환경과 BPSK 모듈레이션을 가정하고, 3GPP 터보코드에 있어서 신뢰도 갱신 길이에 따른 복호기의 BER 성능을 시뮬레이션을 통하여 최적화하여 하드웨어 설계시에 적용하였다.

II. 이론적 배경

1. 3GPP 규격의 터보 부호기

터보 부호기는 두개이상의 회귀길쌈부호를 인터리버를 사이에 두고 병렬로 연결하는 구조로인 병렬 연쇄 길쌈 부호기를 의미한다. 3GPP의 표준규격으로 제안된 1/3 코드의 터보부호기는 그림 1에 나타낸 바와 같이 메모리 길이가 3이고 생성다항식이 (13,15)인 동일한 두 개의 회귀부호기를 인터리버를 이용하여 병렬로 동작시켜 심벌을 만드는 구조이다.

그림 1에 보인 터보 부호기에서 x_k 는 메시지를,

z_k 는 첫 번째 패리티를, 그리고 z'_k 는 두 번째 패리티를 나타낸다. 따라서 위의 터보 부호는 시스템에 톱 부호이므로 프레임 길이가 K인 입력 시퀀스 x_1, \dots, x_K , 그 출력 비트열은 $x_1, z_1, z'_1, x_2, z_2, z'_2, \dots, x_K, z_K, z'_K$ 가 되며, 모든 입력비트가 입력된 후에 두번째 내부 부호기는 동작을 정지시키고 첫번째 내부 부호기의 출력을 자신의 입력으로 사용하여 내부 메모리를 중단시킨다. 그리고 난 후 첫번째 내부 부호기의 동작을 정지시키고, 두번째 내부 부호기의 출력을 그 자신의 입력으로 사용하여 중단시킨다. 따라서 트렐리스를 중단시키기 위한 비트열은 $x_{K+1}, z_{K+1}, x_{K+2}, z_{K+2}, x_{K+3}, z_{K+3}, x'_{K+1}, z'_{K+1}, x'_{K+2}, z'_{K+2}, x'_{K+3}, z'_{K+3}$ 이 된다.^[4]

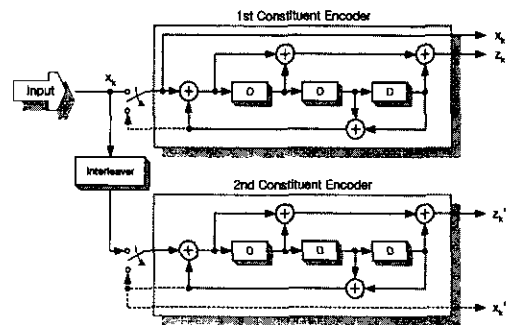


그림 1. 3GPP 규격의 터보 부호기

2. SOVA 복호기

SOVA 알고리즘은 최대 우호 시퀀스를 찾아 송신된 메시지를 복원한다. SOVA를 이용한 복호방법은 기존의 경판정 비터비복호의 방법을 그대로 적용하면서 복호된 비트에 대한 신뢰도 값을 계산하는 부가 첨가되어 그 출력으로서 경판정과 함께 신뢰도를 나타내는 연판정을 포함하게 된다.

그림 2에 도시한 바와 같이 송신된 코드는 잡음 환경에서 변질되어 $y_{1,k}, y_{2,k}, y_{3,k}$ 로 수신단에 입력된다. 첫번째 내부 SOVA 복호기는 $y_{1,k}, y_{2,k}$ 를 입력으로 초기 복호를 수행하여 경판정 비트열과 경판정 비트열에 해당하는 likelihood에 관한 열을 출력한다. 출력된 likelihood 정보는 간단한 연산을 통하여 외부 정보열로 변환된다. 생성된 외부정보 열은 인터리빙되어 두번째 SOVA복호기의 입력의 사전정보로 사용된다. 두번째 SOVA 복호기는 인터리빙된 $y_{1,k}$ 와 $y_{3,k}$ 그리고 첫번째 복호기로부터 생성된 외부정보를 입력으로 복호하게 된다. 두번째 복호기의 외부정보는 디인터리빙되어 다시 첫번째 복호기의

사전정보로 사용되므로써 반복복호를 수행할 수 있게 된다.

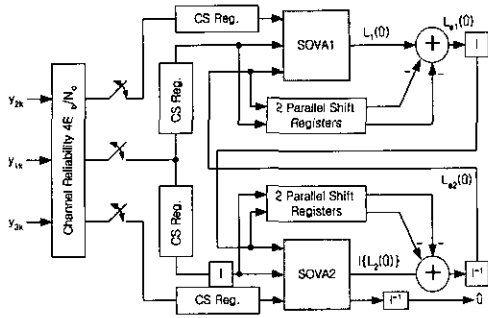


그림 2. SOVA를 이용한 터보 복호기

부가 백색 가우시안 잡음(Additive white Gaussian noise) 환경을 가정하고 1/N 코드인 경우 가지 매트릭은

$$BM_k^{(m)} = \sum_{i=1}^N x_{k,i}^{(m)} L_{\mathcal{Y}_{k,i}} + u_k^{(m)} L(u_k)_A \quad (1)$$

이며, 여기서 BPSK 모듈레이션을 가정하였을 때 $L_c=4Eb/No$, $L(u)_A$ 는 복호기의 사전 확률(a priori value)이며, $u_k(m)$ 는 생존 시퀀스의 시간 k에서의 복호값을 의미한다. 첫번째 SOVA 복호기를 기준으로 보았을 때 그 입력은 메시지에 해당하는 $y_{k,1}$ 과 첫번째 부호기의 출력에 해당하는 첫번째 패리티 $y_{k,2}$, 그리고 사전정보의 세가지이며, 1/2의 코드율을 보이므로 누적 매트릭은

$$PM_k^{(m)} = PM_{k-1}^{(m)} + x_{k,2}^{(m)} L_{\mathcal{Y}_{k,2}} + u_k^{(m)} (L_{\mathcal{Y}_{k,1}} + L(u_k)_A) \quad (2)$$

으로 표현할 수 있다. 터보 복호과정에서 이전 복호에서 출력된 각 u_k 에 대한 외부정보는 다음 단의 복호 과정에서 사전정보 $L(u)_A$ 로 사용된다. 이제 시점 k에서 경로 m이 생존하여 선택될 확률은

$$P_k^m = C e^{PM_k^{m/2}} \quad (3)$$

이고, C는 상수이다. 여기서 $P_k^{m'}$ 을 경쟁경로 m' 이 선택될 확률이고, 시점 k에서 경쟁경로의 누적 경로 매트릭이 $PM_k^{m'}$ 라고 가정하면, 시점 k에서 잘못된 경로를 선택할 확률은

$$\Psi_k^m = \frac{P_k^m}{P_k^m + P_k^{m'}} \quad (4)$$

가 된다. 여기서 SOVA의 신뢰도 값은

$$\Delta_k^m = \log \frac{1 - \Psi_k^m}{\Psi_k^m} = \frac{(PM_k^m - PM_k^{m'})}{2} \quad (5)$$

이다. 식(5)에 따르면 Δ_k^m 은 항상 양의 값을 가지며 이와 같은 신뢰도 값이 크면 클수록 옳은 경로를 선택할 확률이 증가된다. 터보 복호에 있어서 likelihood 값은

$$L(\hat{u}_k) = L(u_k)_A + L_{\mathcal{Y}_{k,1}} + L(\hat{u}_k)_E \quad (6)$$

이다. 여기서 $L(\hat{u}_k)_E$ 는 복호과정에서 나타나는 외부정보를 표시한다. 반복 복호에서 첫 번째 복호과정에서 $L(u)_A$ 는 0으로 초기화되며, 그 이후부터는 이전 복호과정으로부터 입력되는 외부정보로 대체되어 복호에 적용된다. 따라서 이러한 외부정보는 복호성능 향상에 중요한 역할을 하는 파라미터중 하나이다. 그런데 SOVA에서의 경판정을 포함한 likelihood는

$$L(\hat{u}_k) = \hat{u}_k \frac{1}{2} (PM_k^m - PM_k^{m'}) = \hat{u}_k \Delta_k^m \quad (7)$$

가 되므로 복호시에 다음단으로 출력되는 외부정보는 식 (6)으로부터

$$L(\hat{u}_k)_E = L(\hat{u}_k) - (L(u_k)_A + L_{\mathcal{Y}_{k,1}}) \quad (8)$$

가 된다. 따라서 SOVA의 과정에서 생성되는 식(7)의 likelihood에서 외부정보인 $L(\hat{u}_k)_E$ 를 구하기 위해서는 식(8)을 적용해야 할 것이다.

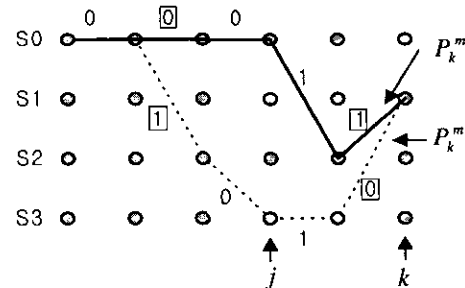


그림 3. 최대 우호 경로에 따른 신뢰도 갱신과정

더불어 SOVA의 likelihood를 계산하는데 있어 중요한 것은 신뢰도 값의 갱신이다. Hagenauer가 제안한 신뢰도 갱신은 그림 3에서 보인바와 같이

시점 k 에서 생존 시퀀스 u_j^m 의 j 번째 ($j < k$) 비트가 경쟁 시퀀스 $u_j^{m'}$ 의 j 번째 비트와 다른 경우에 \log likelihood 값 L_j^m 은 그 값과 Δ_k^m 의 값 중 최소값으로 갱신된다.^{[5][6]} 즉

$$u_j^m \neq u_j^{m'} \Rightarrow L_j^m = \min(L_j^m, \Delta_k^m) \quad (9)$$

이 된다. 그림 3에 likelihood 갱신과정을 보였으며, 여기서 임의의 시간에서 생존경로와 경쟁경로의 복호값이 다른 경우, 즉 갱신이 발생하는 부분을 상자로 표시하였다.

III. 실험 및 결과

1. 복호기 구성

개발된 SOVA 복호기는 기존의 비터비 복호기의 기본 구조를 그대로 가지고 있으면서, 복호비트의 신뢰도값을 연산하여 추가하는 기능을 위한 하드웨어로의 수정을 하였으며, 신뢰도값 갱신을 위한 생존메트릭 메모리 장치를 추가하였다. 개발 SOVA의 특징은 3GPP 규격의 제한길이 $K=4$ 인 (13,15)8 1/3 터보코드의 터보 복호를 위한 내부 SOVA 복호기로서, 가지경로메트릭 연산장치(branch calculation unit), 가산비교선택장치(add compare selection unit), 신뢰도갱신기능이 첨가된 생존메모리장치(soft SMU), 생존경로판단 및 신뢰도값 전달기능의 생존메모리장치(Hard SMU), 출력신호동기화장치(Output control unit), 그리고 전체복호기 모듈을 제어하는 제어모듈(Control module)로 구성되었다.

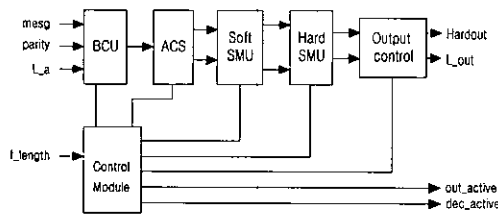


그림 4. 내부 SOVA 복호기의 블록 다이어그램

그림 4에 보인바와 같이, 입력 메시지와 패리티, 그리고 사전정보가 가지 메트릭 연산장치(BCU)으로 입력되고 계산된 가지 메트릭이 가산비교연산부(ACS)를 통과한다. 그 결과로 발생하는 각 스테이트에서 복호된 비트와 누적 메트릭의 차를 출력하여 Soft 값의 update가 수행되는 생존메모리장치(Soft SMU)를 통과한 후 update이 일어나지 않는

Hard SMU를 거쳐 output control 장치에서 경판정 출력 값과 그에 대응하는 신뢰도 값이 출력된다. 각 모듈의 동작은 Control module에서 제어하게 된다.

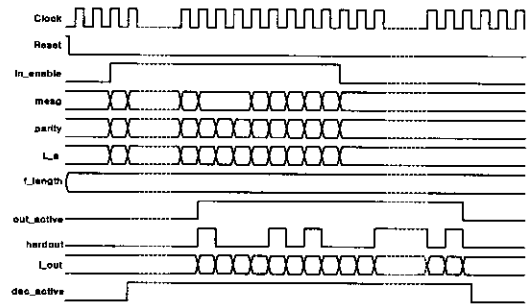


그림 5. 내부 SOVA 복호기의 타이밍 도

그림 5에 도시한 바와 같이 입력정보인 메시지(msg), 패리티(parity), 그리고 사전정보확률(L_a)는 입력신호를 알리는 신호인 in_enable과 동기되어 전송되며, 프레임 길이를 나타내는 f_length는 복호단계 이전에 고정된 값이 된다. 이를 입력으로 하여 복호기가 동작하면 control module에서 BCU와 ACS를 차례로 reset시켜 순차적으로 동작시키고 복호기가 동작하고 있다는 것을 나타내는 신호인 out_active를 high로 만든다. 또한 복호 latency후에 출력신호의 동기를 위하여 out_active신호를 생성하며 이에 동기되어 경판정출력과 경판정 값에 대한 likelihood값이 출력된다.

1.1 가지경로 메트릭 연산장치

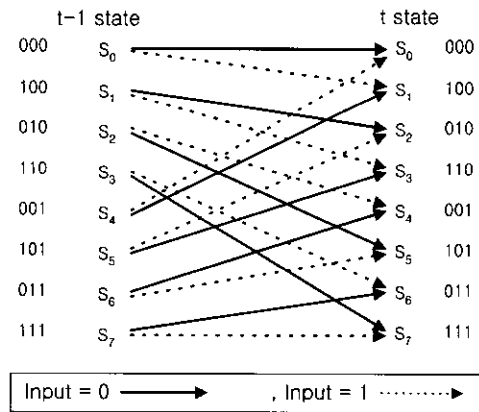


그림 6. 3GPP 터보코드의 트렐리스 도

그림 6에 도시한 3GPP 규격의 터보코드 트렐리스

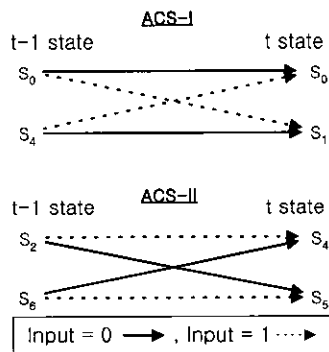
스도에서 보았을때, 식 (1)과 (2)로부터 가지경로 메트릭 연산은 입력 메시지 비트와 그 메트릭의 상태 천이에 따른 출력값에 의하여 부호가 결정된다. 따라서 모든 가지경로에 대한 메트릭은 입력과 출력이 0/0, 0/1, 1/0, 1/1인 경우에 따라 다음과 같이 연산을 적용하였다.

$$\begin{aligned}
 Brnch00 &= \overline{L_a} + \overline{mesg} + \overline{parity} \\
 Brnch01 &= \overline{L_a} + \overline{mesg} + parity \\
 Brnch10 &= L_a + mesg + \overline{parity} \\
 Brnch11 &= L_a + mesg + parity
 \end{aligned}
 \tag{10}$$

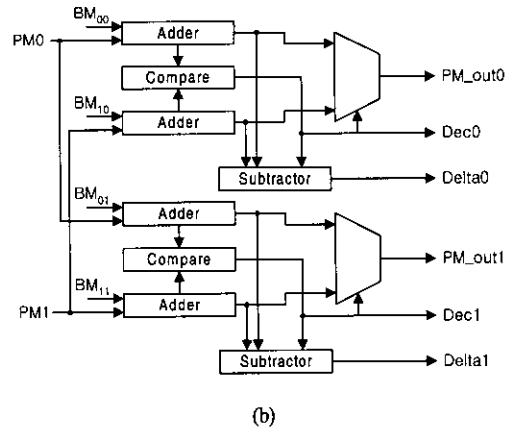
입력 soft bit은 모두 4비트로 하였으며, 여기서 연산된 결과인 가지 메트릭은 mesg, parity, 그리고 L_a가 각각 가장 큰 값이거나 가장 작은 값인 경우의 최소요구 비트인 6비트로 결정하였다. 부호의 적용은 2s complement를 적용하였다.

1.2 Add-compare-select unit(ACSU)

누적경로 메트릭 연산을 위해서는 각 상태에서 상태천이에 따른 가지 메트릭을 할당하여야 하는데 이를 위해서는 기본 연산기능을 공통된 PE(process element)로 설계하여야 한다. 그림 6의 trellis에서 입력값과 출력값에 대한 천이의 공통점으로부터 두 종류의 버터플라이 구조를 얻을 수 있으며 이를 ACS-I PE와 ACS-II PE로 구분하였다. 3GPP 구조의 터보 코드는 state가 8이므로 그림 7(a)에 도시한 바와 같은 두 종류의 구조를 가진 버터플라이가 각각 2개씩 존재하게 된다. 그림 7(b)에 ACS-I PE의 구조를 도시하였다. 여기서 BM뒤의 아래첨자의 첫 번째 0은 그림 7(a)의 버터플라이에서 t-1시간에서,



(a)



(b)

그림 7. (a) 필요한 버터플라이 구조
(b) ACS-I PE의 RTL구조

위에 존재하는 state를 의미하고 두번째 0은 t-1시간에서 아래에 존재하는 state를 의미한다. 또한 Delta는 시간 t의 state에서 보았을때 생존경로의 누적메트릭과 경쟁경로의 누적메트릭의 차가 된다. PM_out은 생존경로의 누적 메트릭이고, Dec는 생존경로의 상태천이를 일으키는 입력값, 즉 시간 t에서 해당 state에서의 복호 비트를 의미한다. ACS-II의 경우도 ACS-I과 같은 방법으로 설계하였다.

또한, 복호과정 동안의 메트릭의 누적으로 인한 오버플로우를 방지하기 위하여 기존의 비터비 복호기에서는 오버플로우 발생시에 임계치를 각 스테이트의 누적 메트릭에서 감하거나, 또는 메트릭의 최소값을 감하는 방법으로 메트릭의 정규화를 수행한다. 그러나 이는 정규화에서의 지연을 야기시키며,

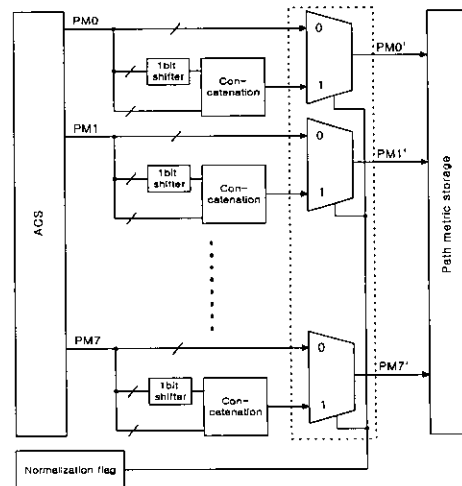


그림 8. 제안된 누적메트릭 정규화 모듈

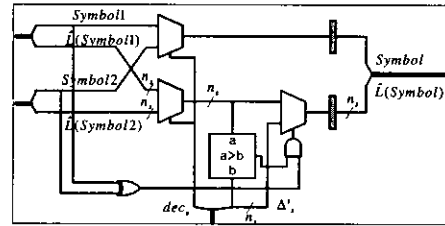
복잡도를 증가시킨다. 따라서 본 논문에서는 SOVA 복호기를 위한 정규화 방법을 제안한다.

그림 8에 도시한 바와 같이 누적메트릭이 임계치 이상이 되면, 전체 메트릭을 구성하는 비트를 신뢰도에 영향을 주는 하위 비트와 영향을 미치지 않는 상위 비트의 두 부분으로 나누어, 신뢰도 값에 영향을 미치지 않는 상위 비트만 1비트 쉬프트연산을 적용하여 정규화 한다. 이를 이용하면, 메트릭 비트수에 해당하는 전체 감산 연산대신에 메트릭 일부분의 1비트 쉬프트연산으로 간단하게 정규화를 시킬수 있으며, 전체 정규화를 위한 하드웨어의 복잡도 감소 및 지연을 극소화할 수 있다.

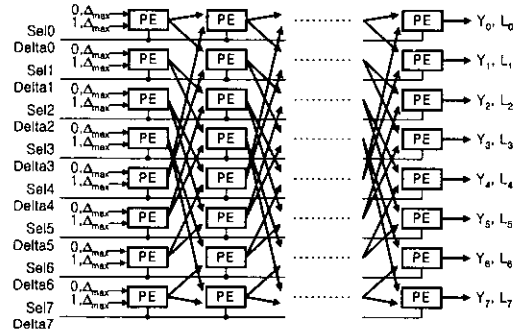
1.3. Soft SMU

본 연구에서 개발한 SOVA 복호기는 복호방식을 Register Exchange 구조를 사용하였으며, Soft bit의 갱신알고리즘이 구현되는 Soft SMU를 그림 9에 도시하였다. 각 시간 t에서 각 state에서 생존경로로 결정된 가지의 복호값 Sel0~Sel7을 입력으로 받고 식(9)에 나타낸 바와 같은 Hageneuer의 갱신절차에 따라 신뢰도 값의 갱신을 수행하면서 Register Exchange를 진행하여 각 복호 비트의 신뢰 값을 구한다. 이와 같은 갱신은 그림 9에서 도시한 바와 같은 PE에서 수행되며 3GPP규격의 터보 코드인 경우 8state가 필요하므로 8개의 PE로 구성된 하나의 열을 10stage반복하여 구성하였다. 이의 출력으로는 갱신이 완료된 시점에서의 likelihood값과 그 시점, 그 스테이트에서의 복호 비트이다. 그런데, 이러한 SMU의 스테이지 수가 증가하면 좀더 신뢰성 있는 복호성능을 얻을 수는 있지만 하드웨어의 복잡도가 매우 증가하게 된다. 따라서 3GPP규격의 터보 코드의 복호에 있어서 알맞은 갱신길이를 선택하는 것이 매우 중요하다. 본 연구에서는 C++로 개발한 터보코드 검증 프로그램을 통하여 시뮬레이션을 수행하여 그 결과를 그림 10에 도시하였다. 그림 10은 갱신길이 delta에 따른 SOVA복호기의 BER 성능곡선이며, 이를 얻기 위하여 100bit 입력프레임과 5회의 반복복호를 적용하였다. 시뮬레이션 결과로 부터 갱신길이 delta가 5인 경우 delta=10, 20, 30에 비하여 많은 성능저하를 보임을 알 수 있으며, delta가 10 인 경우 적은 복잡도로 높은 복호성능을 얻을 수 있음을 알 수 있다. 따라서 앞서 기술한 바와 같이 갱신을 위한 SMU는 10 stage로 설계 하였다. 그러나 비터비 복호기의 복호 비트가 신뢰성을 얻기 위해서는 제한길이의 5배~6배정도의 stage가 필

요하므로 Soft SMU에서 출력된 판정값은 신뢰성있는 복호 비트가 아니다. 따라서 신뢰성있는 복호를 위해서는 10 stage이상의 복호단이 추가로 필요하며, 여기서는 likelihood의 갱신은 제외시키며, 단지 생존 경로를 따라 전송해주면 된다.



(a)



(b)

그림 9. (a) 신뢰도 갱신기능의 PE (b) Soft SMU 구조도

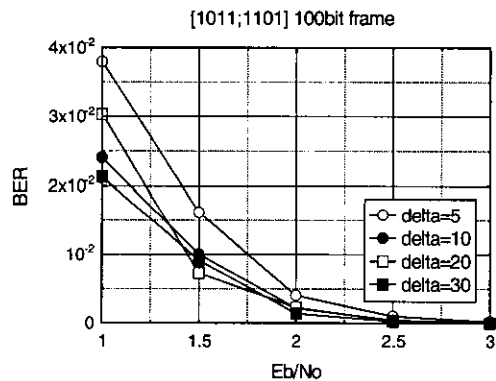


그림 9. 신뢰도 갱신길이 delta=5, 10, 20, 30일때 BER성능

1.4. Hard SMU

앞서 기술한 이유로 추가적인 10 stage는 그림 11에 도시한 바와 같은 PE로 구성된 register exchange SMU를 이용한다. 전체 PE의 연결도는 그림 9의 Soft SMU와 유사한 구조이지만 PE내부

의 동작은 다르다. 그림 11의 Hard SMU를 통과하여 출력되는 각 state에서 복호비트와 likelihood값은 하나의 신뢰성있는 값으로 merge되어 출력된다. 따라서 이를 출력제어 블록에서 경관정 비트열과, 경관정 비트의 정보가 포함된 likelihood값으로 최종 연산하여 out_active 신호에 동기하여 출력한다.

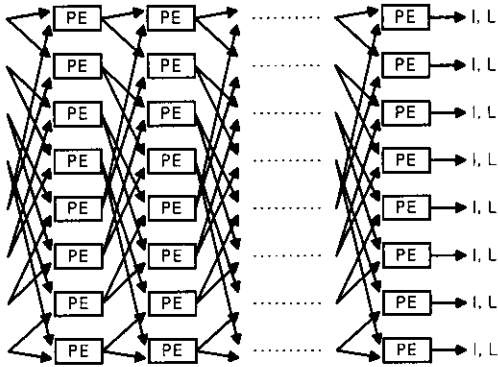


그림 11. Hard SMU 구조도

2. 테스트 결과

테스트를 위하여서는 메시지 열을 터보 부호화하여 체계적 부호열과, 첫번째 RSC 인코더를 통과한 패리티열을 입력으로 사용하며, 첫번째 반복복호이므로 사전 정보는 0으로 놓는다. functional simulation은 OrCad 및 Synopsys Vss에서 모두 수행하였다. 이를 Altera사의 FPGA에 이식하기 위하여 compiler로서 OrCad express (Leonardo Spectrum)을 사용하였으며, P&R은 Altera사의 Maxplus2를 사용하였다. 검증을 위한 테스트 벡터는 랜덤으로 generation된 100bit frame을 C언어로 코딩한 터보 부호화기를 통과시켜 메시지 비트열과 패리티 비트열을 얻고 이를 각각 가장 신뢰성 높은 값으로 4bit soft 비트로 하여 text로 저장하였다. Simulation을 위해서 testbench에서 이 file을 읽어서 입력 벡터로 사용하였으며, FPGA 테스트를 위해서는 National instrument사의 DIO-32HS 디지털 입출력 보드를 사용하여 Labview로 coding하므로써 앞의 입력 벡터를 시뮬레이션 시와 동일하게 입력하였다. 복호 결과를 관찰하기 위해서 HP1663A Logic Analyzer를 사용하여 복호 출력을 측정하였다. 설계된 복호기는 EPF10K100 GC503-3 FPGA에 이식하였으며, MaxplusII에서의 timing simulation으로부터 최대 동작속도가 13.8MHz로 측정되었다.

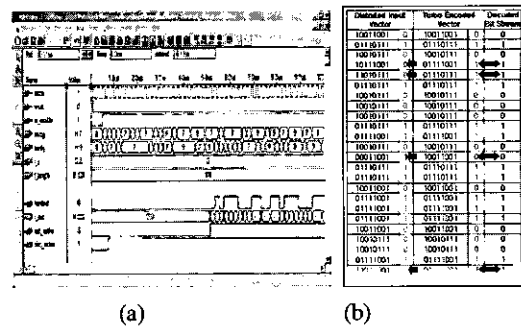
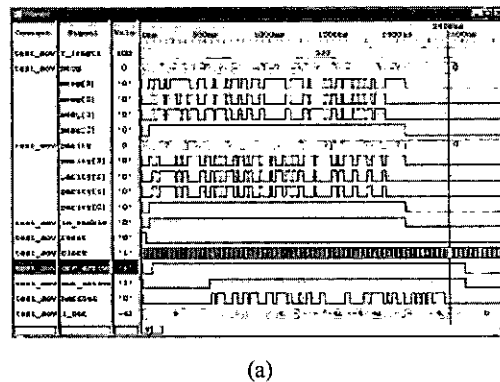


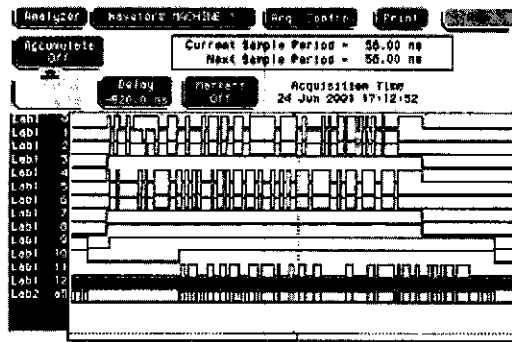
그림 12. SOVA 복호기의 시뮬레이션 결과 (a) waveform (b) 왜곡입력에 대한 오류정정 비교

그림 12(a)에 설계된 SOVA 복호기의 시뮬레이션 결과를 도시하였으며 그림 12(b)에 오류정정의 여부를 테이블로 나타내었다. 그림 12(b)의 왼쪽 열은 임의의 오류가 첨가된 복호기의 입력 벡터이며, 가운데 열은 오류가 첨가되지 않았을 경우의 벡터이다. 그림 12(b)의 세번째 열에 보인바와 같이 복호된 비트열은 오류가 모두 정정되었음을 알 수 있다. 마지막으로 100bit 프레임 입력열에 대한 전체 복호기 동작 시뮬레이션 결과와 FPGA를 이용한 측정결과를 그림 13에 도시하였다.

측정에는 2MHz의 클럭을 적용하였으며 이를 그림 13(b)의 Lab1 12에서 볼 수 있다. 그림 13(b)에서 Lab1 0~Lab1 3은 차례로 msg[3]~msg[0]이고, Lab1 4~Lab1 7은 parity[3]~parity[0]이며, Lab1 8은 in_enable신호로서 입력신호가 전송됨을 나타내며, Lab1 9는 Reset신호이며 Lab1 12는 clock이다. 마지막으로 Lab2 all은 10비트 신뢰도 정보이며 경관정 복호결과와 동시에 출력된다. 그림 13(a)의 시뮬레이션 결과와 그림 13(b)의 측정결과로 부터 설계된 SOVA 복호기가 성공적으로 동작함을 할 수 있다.



(a)



(b)

그림 13. (a) 시뮬레이션 결과 (b) 측정결과

IV. 결론

본 연구에서는 3GPP 규격의 터보 복호기를 위한 내부 SOVA 복호기를 설계하였다. 누적경로메트릭의 정규화에 필요한 하드웨어의 복잡도와 연산지연을 동시에 감소시키기 위하여 누적메트릭을 분할하여 1 비트 쉬프트 연산으로 정규화하는 방법을 제안하여 설계에 적용하였다. 더불어 신뢰도 갱신길이의 최적화를 위하여 C++ 언어를 이용하여 SOVA 복호기의 성능을 분석하였으며, 분석결과 신뢰도 갱신을 위한 SOVA SMU는 10단으로 결정되었으며 이를 설계에 적용하였다. 따라서 설계한 3GPP 규격의 터보 코드 복호기를 위한 내부 SOVA 복호기에서, 복호 방식은 register exchange 방식을 선택하였고, 전체 복호깊이는 20 stage, 그 중 신뢰도 갱신을 위한 갱신 길이를 시뮬레이션 결과로부터 10 stage로 설계 하였다. 설계는 VHDL로 모델링을 하여 OrCad express 및 Synopsys Vss에서 검증하였고, 이를 Altera사의 Maxplus2를 이용하여 EPF10K100 GC503-3 FPGA에 이식하였다. 하드웨어 동작검증은 Labview를 사용하여 시뮬레이션 입력 벡터와 동일한 벡터를 생성하여 National instrument사의 DIO-32HS 디지털 입출력 보드를 통하여 FPGA에 입력하였으며, 그 복호결과를 HP1663A 로직 분석기에서 확인하여 동작검증 결과와 비교하여 동작을 확인하였다. 본연구 결과는 고성능 터보 복호기의 설계에 있어서 필수적인 요소가 되리라 사료된다

참 고 문 헌

[1] C. Berrou, A.Glavieux, and P. Thitimajshima, Near Shannon limit error-correcting coding and

decoding: TURBO-Code, in Proc. Ieee. Int. Conf. Commun., Geneva, Switzerland, pp. 1064-1070, May 1993

[2] Joachim Hagenauer, and Peter Hoehner, "A Viterbi Algorithm with Soft-Decision Outputs and its Applications," Proceedings of IEEE Globecom Conference, Dallas TX, USA, pp. 1680-1686, 1989

[3] O. J. Joeressen, and H. Meyr, "A 40Mb/s Soft-Output Viterbi Decoder," IEEE Journal of Solid-state Circuits, Vol. 30, No. 7, pp. 812-818, July, 1995

[4] 3G TS 25.212 v.3.2.0 Technical specification, pp. 15-16, Mar., 2000.

[5] Joachim Hagenauer, Lutz Papke, Decoding Turbo -Codes with the Soft Output Viterbi Algorithm, IEEE International Symposium on Information theory, pp. 164, 1994

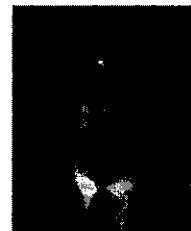
[6] Gerhard Bauch, Volker Franz, A comparison of Soft-In/Soft-Out Algorithm for Turbo-Detection, Proceedings of the International Conference on Telecommunication, pp. 259-263, Jun. 21, 1998,

[7] Lang Lin, Roger S. Cheng, On The Tail Effect of SOVA-Based Decoding For Turbo Codes, Global Telecommunications Conference, 1997. GLOBECOM97., IEEE, vol. 2, pp. 644-648, 1997

[8] Lang Lin, Roger S. Cheng, Improvements in SOVA-Based Decoding for Turbo Codes, IEEE International Conference on Communications, Vol. 3/3, pp. 1473-1478, Jun. 8, 1997

김 주 민(Jumin Kim)

정희원



1993년 2월 : 인하대학교
전자재료공학과졸업
1995년 8월 : 인하대학교
전자재료공학과 공학석사
1999년 3월~현재 : 인하대학교
전자재료공학과 박사과정
<주관심 분야> FEC 알고리즘,

VLSI 구현, 신호처리

고 태 환(Tachwan Ko)

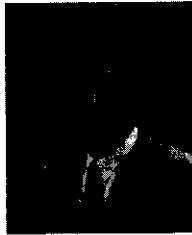
정회원



2000년 2월 : 인하대학교
전자재료공학과졸업
1999년 3월~현재 : 인하대학교
전자재료공학과석사과정
<주관심 분야> FEC 알고리즘,
VLSI 구현, 신호처리

정 덕 진(Duckjin Chung)

정회원



1970년 2월 : 서울대학교
전기공학과졸업
1984년 8월 : Utah State Univ.
전기공학과 공학석사
1988년 8월 : Univ. of Utah
전기공학과 공학박사
1980년 ~ 1988년: 한국전자통신연구원 책임연구원
1999년 ~ 현재: 인하대학교 정보통신공학부 교수
<주관심 분야> FEC 알고리즘, 인공지능, Genetic
Algorithm, VLSI 구현, 신호처리