

# Block-wise 알고리즘을 이용한 터보 코드 복호기의 구현

정희원 박구현\*, 윤석현\*\*, 강법주\*\*\*, 진익수\*\*\*, 홍대식\*, 강창언\*

## Implementation of a Turbo-code Decoder Using a Block-wise Algorithm

Goo-hyun Park\*, Suk-hyon Yoon\*\*, Bub-joo Kang\*\*\*, Ik-soo Jin\*\*\*, Dae-sik Hong\* and  
Chang-eon Kang\* *Regular Members*

### 요 약

터보 코드는 Maximum A Posteriori(MAP) 알고리즘을 적용한 반복 복호 시에 최적의 성능을 보인다. 하지만, MAP 알고리즘의 과도한 메모리 요구량과 계산의 복잡도는 터보 코드 복호기 구현에 큰 장애가 된다. 본 논문에서는 터보 코드 복호를 위한 MAP 복호기의 설계와 구현을 위한 구체적인 방법을 제시한다.

설계된 MAP 복호기는 빠른 복호 속도와 낮은 복잡도를 위하여 확률비를 이용한 브랜치 메트릭과 절단에 의한 상태 메트릭 정규화 방법을 사용하고 있으며, block-wise 알고리즘을 적용하여 메모리 요구량을 현저하게 감소시킨다. 또한, 파이프라인 구조의 시분할 처리 방법을 통하여 재귀 연산의 한계를 극복함으로써 MAP 복호기의 복잡도를 Soft Output Viterbi Algorithm(SOVA) 복호기 수준으로 낮추며, 다양한 프레임 크기를 가지는 시스템에 효율적으로 적용할 수 있는 다층 제어 구조도 제안한다.

제안된 block-wise 알고리즘을 이용한 터보 코드 복호기는 약 20,000 게이트의 Field Programmable Gate Array(FPGA) 단일 칩상에 설계되고, FPGA 상에서 8번 반복 복호 하였을 때에 약 300 kbps의 복호 능력을 가진다. 그리고 모든 하드웨어적인 환경을 고려하여도, 이상적인 MAP 복호기를 사용한 경우에 대하여 약 0.1~0.2dB의 성능 저하만을 보인다.

### ABSTRACT

This paper deals with the Turbo-code decoder for cdma-2000 systems. In this paper, several efficient methods for the implementation of a MAP algorithm are studied. In several decoding methods for the Turbo-code, only the MAP algorithm can achieve acceptable performance within 1dB of the value corresponding to the Shannon limit. However, the implementation of MAP algorithm for Turbo-code system has several hardships. Therefore, to overcome these practical problems, several implementation methods of the MAP decoder are proposed.

By using the novel time-shared process of a pipe-lined structure, the restriction of recursion process on state metric can be efficiently conquered, and the complexity of the MAP decoder can be reduced to the order of a SOVA(Soft Output Viterbi Algorithm) decoder. And, an efficient structure for the controller is also proposed for cdma-2000 system.

The designed decoder has been implemented in only one 20,000 gate circuit. It has been validated by VHDL, which has been compared with initial simulation(C programs). The designed decoder has 300 kbps decoding processing ability with 8 times iterations on a FPGA circuit, and has a deviation of about 0.1-0.2 dB over the ideal MAP decoder, even if all hardware environments were considered.

\* 연세대학교 전기·전자 공학과 정보통신연구실

\*\* LG전자 정보통신 차세대통신연구소 이동통신방식실

\*\*\* 한국전자통신연구원(ERTI:Electronics and Telecommunications Research Institute)

논문번호 : 00277-0720, 접수일자 : 2000년 7월 20일

※ 본 연구는 한국전자통신연구원 연구과제 및 두뇌 한국21사업 핵심분야에 의하여 지원되었음.

## I. 서론

차세대 이동통신 방식으로 가장 강력하게 추진되고 있는 cdma-2000 시스템에서는 강력한 오류 정정 코드인 터보 코드를 데이터 통신을 위한 오류 정정 코드로서 채택하고 있다. 터보 코드는 MAP 알고리즘을 이용한 반복 복호 시에 Shannon의 한계에 거의 도달하는 성능을 보이지만, MAP 알고리즘의 과도한 복잡도는 터보 코드 복호기 구현의 가장 큰 장애가 된다.

MAP 복호기의 복잡도를 감소시키기 위한 방법으로 Max-Log-MAP 알고리즘이 제안되어 곱셈 연산의 부담은 줄었으나 과도한 메모리 요구량은 여전히 큰 문제점으로 남았다<sup>[7]</sup>. 이를 해결하기 위하여 Viterbi는 block-wise 알고리즘을 제안하였다<sup>[4]</sup>. Block-wise 알고리즘을 구조적인 측면에서 복호기의 복잡도를 줄일 수 있는 방법으로 메모리의 요구량을 현저히 감소시킬 수 있었으나, 연산량은 오히려 늘어나게 되어 연산기 복잡도의 증가를 초래한다<sup>[14]</sup>. 따라서 block-wise MAP 알고리즘을 적용할 수 있는 효율적인 복호기의 구조가 필요하게 되었다. 본 논문은 block-wise MAP 복호기 구현을 위한 새로운 방법들을 제시하고 있다.

본 논문은 다음과 같이 구성되었다. II장에서는 터보 코드의 개념을 소개하고 터보 코드의 복호를 위한 MAP 알고리즘과 반복 복호 방법을 설명한다. III장에서는 터보 코드 복호기의 구현을 위한 기본 구조를 제시하고 제안된 알고리즘과 구조를 설명한다. IV장에서는 제안된 복호기의 구조를 적용하여 FPGA를 이용하여 설계하고, V장에서는 제안된 복호기를 모의 실험하고 일반적인 터보 코드 복호기와 비교한다. 그리고 마지막으로 VI장에서 결론을 맺는다.

## II. 터보 코드와 MAP 복호기

터보 코드는 두 개의 구성 코드를 사용하는 병렬 연쇄 컨벌루션 코드로서, Berrou, Glavieux, Thitimajshima<sup>[1]</sup>에 의해 코딩 이득과 복잡도에 있어서 주목할만큼 발전된 코딩 방식으로서 제안되었다<sup>[1]</sup>. 일반적으로 터보 코드라고 알려진 병렬 연쇄 컨벌루션 코드의 인코더는 그림1에 나타난 바와 같이 두 개의 컨벌루션 인코더를 병렬 연결시키고, 두 인코더는 인터리버를 사용하여 분리해 주는 형태로

구성된다<sup>[2,3]</sup>.

복호 시에 하나의 복호기가 다른 하나의 복호기에서 출력된 신뢰도 값을 이용하게 된다면 성능의 향상을 기대할 수 있다. 이와 같이 정보를 순환시켜 복호하는 방법을 반복 복호(iterative decoding)라 하며, 이때 하나의 복호기에서 나머지 하나의 복호기에 전달해주는 신뢰도 값을 외부정보(extrinsic information)라고 한다<sup>[4,6]</sup>. 반복 복호 알고리즘을 적용한 연쇄 코드 복호기에는 수신된 정보와 외부 정보를 이용하여 신뢰도 값을 연산하여 출력하는 부분이 필요하게 되는데, 이것을 Soft-In Soft-Out (SISO) 모듈이라 한다<sup>[7]</sup>. SISO 모듈을 이용한 병렬 연쇄 코드 복호기를 그림2에 나타내었다.

SISO 모듈을 구성하는 방법은 SOVA를 이용하는 방법과 MAP 알고리즘을 이용하는 방법 등이 있고, 일반적으로 MAP 알고리즘을 사용하는 방법이 SOVA를 사용하는 방법보다 우수한 성능을 보인다<sup>[8-10]</sup>. MAP 알고리즘은 Bahl 등에 의하여 처음으로 제안되었고 뛰어난 성능을 보이지만 코드워드가 많아지면 복호할 때 복잡도가 증가하게 되는 단점을 갖고 있다. 그리고 한 프레임의 정보 비트들을 모두 전송받은 후에 복호를 하게 되므로, 비터비 알고리즘보다는 시간 지연도 길어지게 된다. 수신된 시간  $k$ 에서 수신된 심볼을  $y_k$ 라고 할 때, 정보비트  $\hat{a}_k$ 의 대수 근사화 확률비(Log Likelihood Ratio : LLR)를 유도하면 식 (1)과 같다<sup>[11]</sup>.

$$L(\hat{a}_k) = \frac{\sum_{s_k} \sum_{s_{k-1}} (a_{k-1}(S_{k-1}) \cdot \gamma_1(y_k, S_{k-1}, S_k) \cdot \beta_k(S_k))}{\sum_{s_k} \sum_{s_{k-1}} (a_{k-1}(S_{k-1}) \cdot \gamma_0(y_k, S_{k-1}, S_k) \cdot \beta_k(S_k))} \quad (1)$$

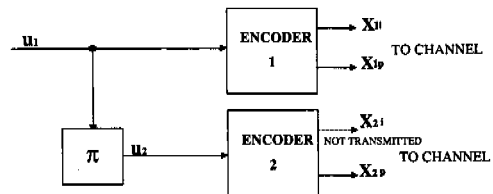


그림 1. 병렬 연쇄 코드의 인코더 구조

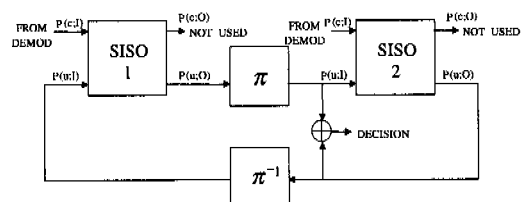


그림 2. 병렬 연쇄 코드의 반복적인 복호기 구조

식(1)에서  $\alpha_k(S_k)$ 와  $\beta_k(S_k)$ 는 각각 순방향 재귀 상태 메트릭(forward recursion state metric)과 역방향 재귀 상태 메트릭(backward recursion state metric)이고  $\gamma_k(S_k, S_{k-1}, S_k)$ 는 스테이트 천이확률을 나타내는 브랜치 메트릭(branch metric)이다<sup>[11]</sup>.

MAP 알고리즘은 우수한 복호 능력을 가지지만 계산의 복잡도 때문에 실제 시스템에 적용하기 어렵다. 이는 주로 곱셈 연산에 의해 발생하게 되는데, 이러한 문제를 해결하기 위하여 곱셈 연산을 log연산을 통해 덧셈으로 변환시키는 Max-Log-MAP 알고리즘이 주로 사용된다<sup>[7][12-13]</sup>. Max-Log-MAP 알고리즘을 적용한 변형된 대수 근사화 확률비는 다음 식과 같다.

$$\hat{a}_k \approx \begin{matrix} \max_{S_k, S_{k-1}} & (\ln \alpha_{k-1}(S_{k-1}) + \ln \gamma_k(y_k, S_{k-1}, S_k) + \ln \beta_k(S_k)) \\ - \max_{S_k, S_{k-1}} & (\ln \alpha_{k-1}(S_{k-1}) + \ln \gamma_0(y_k, S_{k-1}, S_k) + \ln \beta_k(S_k)) \end{matrix} \quad (2)$$

### III. 하드웨어 구현을 위한 알고리즘

이상적인 MAP 복호기는 많은 계산량을 필요로 하므로 하드웨어 구현 시에 많은 복잡도와 복호 시간을 필요로 한다. 따라서 하드웨어의 구현에는 logarithm 을 적용한 Max-Log-MAP 알고리즘을 보정해서 사용한다. 본 논문에서는 보다 효율적인 구현을 위하여 여러 가지 알고리즘을 제시한다.

#### 1. 확률비를 이용한 브랜치 메트릭

Max-Log-MAP 알고리즘에서 logarithm을 적용한 상태의 천이 확률 값을 브랜치 메트릭이라 한다. 수신된 시간  $k$ 에서 수신된 심볼을  $y_k$ 라 하고 트랜미스의  $S_{k-1}$  상태에서  $S_k$  상태로 천이할 때, 수신된 심볼을 구성하는 각 비트의 대수 확률비를 비트 메트릭(BIT Metric : BTM), 수신된 심볼의 대수 확률비를 브랜치 메트릭(Branch Metric : BM)이라 정의한다. 이 때, 비트 메트릭과 브랜치 메트릭은 각각 다음 식과 같다.

$$BM(y_k, S_{k-1}, S_k) = \ln \gamma_k(y_k, S_{k-1}, S_k) = \sum_{j=1}^M \ln f(y_{k,j}, x_{k,j}) = \sum_{j=1}^M (BTM_{i,j}(y_{k,j}, S_{k-1}, S_k)) \quad (3)$$

$$BTM_{i,j}(y_{k,j}, S_{k-1}, S_k) = \ln (f(y_{k,j}, x_{k,j})) = \ln \left( \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( -\frac{(y_{k,j} - x_{k,j})^2}{2\sigma^2} \right) \right) = \begin{cases} C_k + \frac{y_{k,j}}{\sigma^2}, & (x_{k,j} = +1) \\ C_k - \frac{y_{k,j}}{\sigma^2}, & (x_{k,j} = -1) \end{cases} \quad (4)$$

식(4)에서  $C_k$ 는 최종 출력인 대수 근사화 확률비의 연산 시에 소거되는 상수이다. 식에서 보는 것과 같이 코드율이  $1/M$  인 코드의 브랜치 메트릭을 구하기 위해서는  $2 \times M$ 개 비트 메트릭의 조합으로 이루어진  $2^M$ 가지 종류의 확률 분포를 계산해야 한다. 브랜치 메트릭을 보다 쉽게 구하기 위하여 확률 비를 이용한 비트 메트릭(BTM)을 정의하면 다음과 같다.

$$BTM_{i,j}(y_{k,j}, S_{k-1}, S_k) = \ln \left( \frac{f(y_{k,j}, x_{k,j})}{f(y_{k,j}, x_{k,j} = -1)} \right) = \ln \left( \frac{\frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( -\frac{(y_{k,j} - x_{k,j})^2}{2\sigma^2} \right)}{\frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( -\frac{(y_{k,j} - (-1))^2}{2\sigma^2} \right)} \right) = \begin{cases} \frac{2 \cdot y_{k,j}}{\sigma^2}, & (x_{k,j} = +1) \\ 0, & (x_{k,j} = -1) \end{cases} \quad (5)$$

식 (5)에서 보인 바와 같이, 확률비를 이용한 방법에서는 '1'의 비트 메트릭은 단순히 수신된 신호( $y_{k,j}$ )와 선형관계를 가지게 되고 '0'의 비트 메트릭은 항상 고정된다. 그러므로 브랜치 메트릭을 구하기 위하여 참조하는 비트 메트릭의 종류가  $M$ 개로 줄어들게 된다. 또한, 대칭되는 두 개의 비트 메트릭 사이의 차이가 줄어들게 되어 Max 연산의 보상을 위한 테이블이 단순해지는 장점이 있다. Max-Log-MAP 알고리즘은 Max 연산에 의한 오차를 보상하기 위하여 Jacobian logarithm을 사용하게 되고 이는 다음 식과 같다.

$$\ln(e^{\delta_1} + e^{\delta_2}) = \max(\delta_1, \delta_2) + f_c(|\delta_1 - \delta_2|) \quad (6)$$

이때 제안된 방법을 사용하면 두 메트릭 간의 차이( $|\delta_1 - \delta_2|$ )가 줄어들게 되어 보상 회로에서 참조하게 되는 보상 범위와 보상 값의 범위가 줄어드는 장점을 가진다. 예를 들어, cdma-2000 시스템에서 양자화 레벨을  $Q=16$ 으로 하고 입력 신호의 범위를  $\pm 2$ 로 결정했을 경우의 보상 테이블의 복잡도를 표1에 나타내었다.

표 1. 보상 테이블의 복잡도 비교

	보수 연산을 이용한 방법	확률비를 이용한 방법
보상 범위 ( $ \delta_1 - \delta_2 $ )	0 ~ 11	0 ~ 5
보상 값의 수	6 levels	3 levels

표1을 보면 확률비를 이용한 방법을 사용하였을 경우에 기존의 보수 연산을 이용한 방법에 비해 보 상 회로의 복잡도가 반으로 감소함을 알 수 있다.

### 2. Block-wise MAP 알고리즘

MAP 복호기는 전체 프레임에 대한 역방향 상태 메트릭을 모두 저장해야 함으로 많은 메모리를 요구한다. 예를 들어, cdma-2000 시스템에서 메모리의 크기는 한 상태에 8비트씩 할당할 경우 한 단계의 상태 저장에는 64(=8×8)비트가 필요하고, 최대 6,144 정보비트 프레임에 대해 393,216(=6,144×64)비트의 메모리가 요구된다. 그러므로, 메모리 요구량을 줄이기 위해 변형된 역방향 상태 연산을 사용하며 이것을 block-wise MAP 알고리즘이라 정의하고 그림3에 나타내었다<sup>[14]</sup>.

변형된 역방향 상태 연산은 전체 프레임에 대한 수행 과정을 작은 크기의 부-블록들로 나누어 각 부-블록단위로 수행하는 방법이다. 이 방법에는 신뢰할 수 있는 역방향 상태 연산의 초기값이 필요하므로 T만큼의 트레이닝 구간을 필요로 하게 되고 이 트레이닝 구간은 잘 알려진 대로 5×K(K : 구속장 길이) 이상이면 충분하다.

Block-wise MAP 알고리즘을 사용할 경우 상태 메트릭을 저장하기 위한 메모리 요구량은 하나의 부-블록의 상태 메트릭을 저장할 만큼의 크기로 감소한다. 만일 부-블록의 크기를 32정보 비트로 결정하였다면 상태 메트릭을 저장하기 위한 메모리 요구량은 단지 2,048(=32×64)비트에 불과하다. 게다가 재귀 상태 메트릭 연산과 외부정보 연산간에 부-블록단위의 파이프라인 구조의 적용이 가능하므로 역방향 재귀 상태 메트릭 연산을 기다리기 위한 복호 지연 시간도 현저하게 감소한다. 하지만, block-wise MAP 알고리즘을 적용하면 역방향 트레이닝을 위한 상태 메트릭 연산이 늘어나게 되어 추가의 재귀 상태 메트릭 프로세서를 사용하거나 시간 지연을 감수해야 한다.

### 3. 파이프라인 구조의 시분할 처리

Max-Log-MAP 알고리즘의 순방향과 역방향의 상태 메트릭은 각각 다음과 같다.

$$\ln \alpha_k(S_k) = \max_{S_{k-1}, i} (\ln \alpha_{k-1}(S_{k-1}) + \ln \gamma(y_k, S_{k-1}, S_k)) + C_{\alpha}$$

$$\ln \beta_k(S_k) = \max_{S_{k+1}, i} (\ln \beta_{k+1}(S_{k+1}) + \ln \gamma(y_k, S_k, S_{k+1})) + C_{\beta}$$

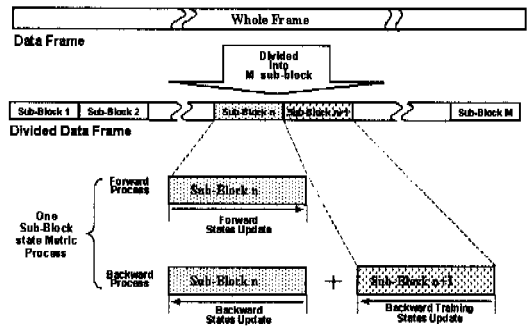
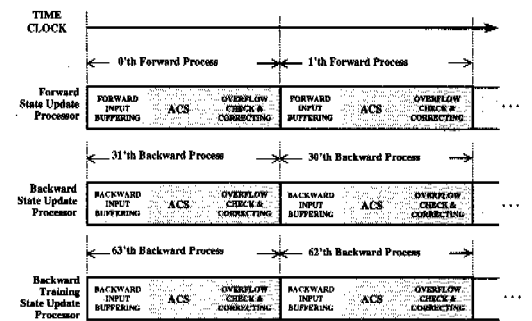
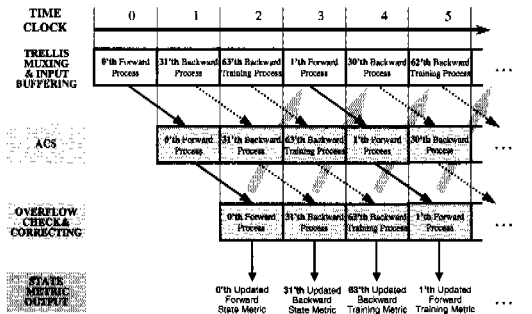


그림 3. Block-wise MAP 알고리즘 개념도



(a) 일반적인 상태 메트릭 연산과정



(b) 제안된 파이프라인 구조의 시분할 처리 과정

그림 4. 파이프라인 구조의 시분할 처리

식(7)과 식(8)에서 보인 바와 같이 MAP 알고리즘의 상태 메트릭 연산은 재귀적인 연산이다. 즉, 순차적으로 연산을 수행하여야 함으로 다음 단계의 상태 메트릭을 연산하기 위해서는 현재 상태의 메트릭 연산이 완료되어야 한다. 그러므로 각각의 상태 메트릭의 연산 과정의 구현 시에 파이프라인 구조를 적용할 수 없고, 한 단계의 상태 메트릭 연산 과정은 ACS 외에 상태 메트릭의 정규화 부분과 Max 연산의 보정 부분이 필요하므로 많은 복호 시간을 요구하게 된다. 게다가 block-wise 알고리즘을 적용하면 역방향 트레이닝을 위한 상태 메트릭 연

산이 늘어나게 됨으로 상태 메트릭의 효율적인 연산을 위한 구조가 필요하다. 이러한 문제점을 해결하기 위하여 상태 메트릭 연산에 파이프라인 구조의 시분할 처리 방법을 제안하고 이를 그림4에 나타내었다.

파이프라인 구조의 시분할 처리 방법은 세 개의 독립적인 상태 메트릭 재귀 연산을 동시에 효율적으로 처리하기 위한 방법이다. 제안된 방법은 상태 메트릭의 연산과정을 3구간으로 분리하여 파이프라인 구조를 적용한 뒤, 시분할 처리 방식을 적용하여 입력단에 순방향, 역방향, 역방향 트레이닝 브랜치 메트릭을 순차적으로 입력시키는 방법이다. 이때, 순방향 트렐리스와 역방향 트렐리스의 차이점을 해결하기 위하여 입력 버퍼링 부분에서 멀티플렉싱을 수행하게 된다. 제안된 방법은 각각의 재귀 연산 과정에 충분한 시간을 보장하면서도 3개의 재귀 연산을 하나의 연산기에서 한번의 재귀 연산 시간 동안에 처리할 수 있다. 그러므로, 세 개의 연산기가 필요한 block-wise MAP 알고리즘의 재귀 상태 메트릭 연산을 시간의 지연 없이 하나의 변형된 연산기로 수행할 수 있을 뿐 아니라 하드웨어의 증가도 거의 없다.

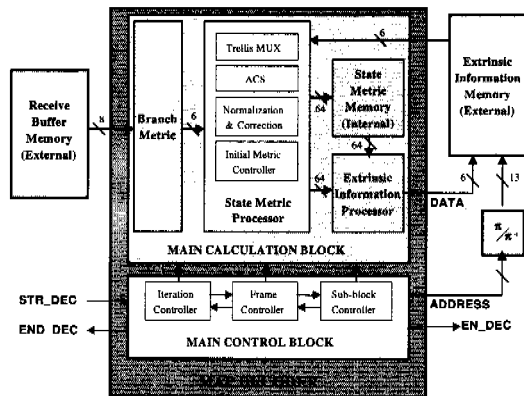


그림 5. 설계된 터보 코드 복호기의 구조

#### IV. 터보 코드 복호기의 설계

이 장에서는 II, III장에서 설명된 알고리즘과 구조를 적용한 cdma-2000 시스템의 터보 코드 복호기를 FPGA로 구현한다. 설계된 터보 코드 복호기의 전체 구조를 그림5에 나타내었다.

그림5에서 수신 버퍼 메모리와 외부 정보 메모리는 반복 복호를 위하여 프레임 단위로 수신 데이터

를 저장한다. 외부 정보 메모리에 저장된 LLR 값은 반복 복호 시에 인터리버나 디인터리버를 거쳐 다음번 반복 복호 시에 외부 정보 입력으로 사용된다. 이때 인터리버는 논리적인 인터리버를 대신하기 위하여 메모리를 이용한 인터리빙 방법을 사용하였다.

#### 1. Block-wise MAP 복호기 연산부 설계

##### 브랜치 메트릭 모듈

브랜치 모듈은 수신된 신호의 샘플링 값을 이용하여 모든 종류의 브랜치 메트릭 값을 구하는 모듈이다. 브랜치 모듈의 입력인 수신 신호의 샘플링 값은 각 4비트로 샘플링된 0000(2) 부터 1111(2)까지의 무부호 정수를 사용한다<sup>[15]</sup>. 이것은 상태 메트릭의 모듈 구조에서 오버플로우만 존재하고 언더플로우가 없는 상태 메트릭을 무부호 정수로 처리하기 위함이다. 설계된 브랜치 메트릭 모듈을 그림6에 나타내었다.

그림6에서 나타낸 것과 같이 각 비트 메트릭에 4비트를 사용하고 3가지 비트 메트릭의 합으로 구성되는 브랜치 메트릭에 6비트를 사용하였다. 브랜치 메트릭의 연산기는 III장에서 제안한 확률비를 이용한 비트 메트릭의 계산법을 사용한다. 따라서 브랜치 메트릭은 식(3)에서 처럼 '1'의 비트 메트릭만을 더하여 구해짐으로 단순한 가산기로 구성된다.

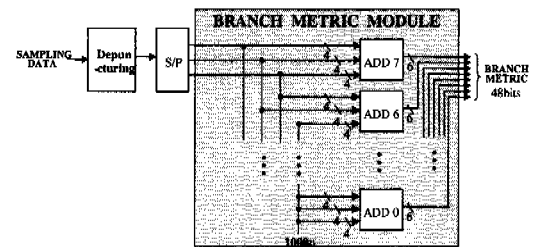


그림 6. 브랜치 메트릭 모듈

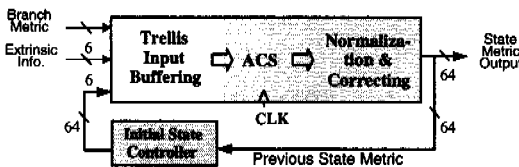
##### 상태 메트릭 모듈

상태 메트릭 모듈은 브랜치 메트릭과 이전 복호 과정의 LLR값을 외부 정보로 사용하여 재귀 상태 메트릭의 연산을 수행하는 부분이다. 따라서 상태 메트릭 모듈은 (6×8=)48 비트의 브랜치 메트릭, 6비트의 외부 정보 그리고 (8×8=)64 비트의 이전 상태 메트릭을 이용한 재귀 연산을 수행하여 상태 메트릭을 갱신한다.

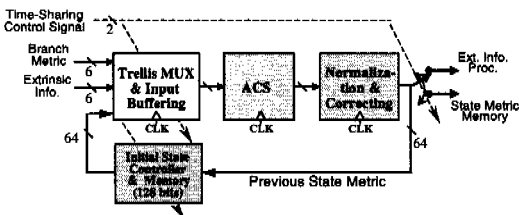
상태 메트릭의 연산은 Max-Log-MAP 알고리즘을 기반으로 수행됨으로 상태 메트릭의 오버플로우를 방지하기 위한 정규화(normalization) 부분이 필

수적이므로 절단에 의한 정규화 방법을 사용하여 8 비트의 상태 메트릭의 유지한다. 또한 Max 연산에 의한 성능 열화를 극복하기 위하여 보상 테이블을 이용한 Max연산의 보상(correcting)을 수행한다.

설계된 상태 메트릭 모듈은 block-wise MAP 알고리즘을 적용하기 위하여 부-블록 단위의 트렐리스 초기화를 위한 초기화 메모리를 가지고 있다. 그리고 파이프라인 구조의 시분할 처리방법을 적용하기 위하여 상태 메트릭 연산과정을 클럭으로 구분하는 파이프라인 구조와 트렐리스의 연결을 바꾸어주는 트렐리스 MUX(Multiplexer) 부분이 추가되었다. 일반적인 상태 메트릭 모듈과 본 논문에서 설계된 상태 메트릭 모듈을 그림7에 각각 나타내었다.



(a) 일반적인 상태 메트릭 모듈



(b) 제안된 상태 메트릭 모듈

그림 7. 상태 메트릭 모듈

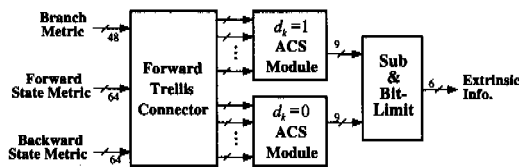


그림 8. 외부 정보 모듈

그림7-(b)에서 초기화 메모리 부분은 부-블록 단위의 재귀 상태 메트릭 연산에 마지막 값을 저장하고 다음 부-블록의 재귀 상태 메트릭 연산이 시작될 때에 상태 메트릭을 초기화하는 역할을 한다. 상태 메트릭 모듈에서 출력되는 3가지 재귀 상태 메트릭 중에서 순방향과 역방향 트레이닝 상태 메트릭의 마지막 값만이 상태 초기화 메모리에 저장 된다. 따라서 초기화 메모리의 크기는 단지 128(=64×2) 비

트에 불과하다. 그리고 출력되는 역방향 상태 메트릭만이 상태 메트릭 메모리로 저장되고, 읽기와 쓰기를 동시에 수행하기 위하여 상태 메트릭 메모리는 부-블록 크기의 2배에 해당하는 4,096(=64×32×2) 비트의 크기를 가진다.

외부 정보 모듈

외부 정보 모듈은 그림8에서 나타낸 것과 같이 이전에 계산된 브랜치 메트릭과 상태 메트릭을 이용하여 LLR을 연산하여 반복 복호를 위한 외부 정보를 생성한다. 외부 정보 모듈의 입력 중에서 브랜치 메트릭과 순방향 상태 메트릭은 상태 메트릭 모듈의 출력을 사용하고 역방향 상태 메트릭은 상태 메모리로부터 불러 온다. ACS 모듈의 최대값을 찾는 과정은 많은 시간을 필요로 하지만 파이프 라인 구조의 적용이 가능하므로 복호 시간에 큰 영향을 미치지 않는다. 출력단의 비트 제한기는 성능의 저하 없이 하드웨어의 복잡도를 줄이기 위하여 10 비트의 외부 정보의 비트 수를 6비트로 제한하는 역할을 한다<sup>[15]</sup>.

2. Block-wise MAP 복호기 제어부 설계

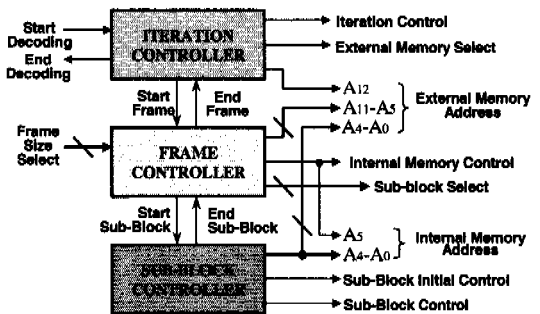


그림 9. Block-wise MAP 복호기 제어부 블록도

표 2. 부-블록 초기화 제어의 종류

제어 (SEL)	순방향 초기화 제어	역방향 초기화 제어	트레이닝 초기화 제어
0 (000(2))	-	-	All Zero
1 (001(2))	-	B_init_memory	All Zero
2 (010(2))	Init Value	B_Init Memory	All Zero
3 (011(2))	F_init_memory	B_Init Memory	All Zero
4 (100(2))	F_init_memory	B_Init Memory	-
5 (101(2))	F_init_memory	Init Value	-
6 (110(2))	F_init_memory	-	-

◆ Init Value: 상태 메트릭을 트렐리스의 초기값으로 설정  
 ◆ All Zero: 상태 메트릭을 모두 zero 값으로 설정

- ◆ F\_init\_memory: 이전 부분블록의 순방향 메트릭 연산의 마지막 값으로 설정
- ◆ B\_init\_memory: 이전 부분블록의 역방향 트레이닝 메트릭 연산의 마지막 값으로 설정

설계된 Block-wise MAP 복호기의 제어부는 block-wise 알고리즘과 다양한 프레임 크기를 가지는 시스템에 효율적인 적용하기 위하여 3단계의 다층 제어 구조를 가진다. 제어부는 부제어(sub-control) 블록, 프레임 제어(frame-control) 블록, 그리고 반복-제어(iteration control) 블록의 나뉘어 있으며 이를 그림9에 나타내었다.

다층 제어 구조에서 부제어 블록은 단지 부분블록 단위의 제어 신호만을 생성한다. 그러므로, 프레임 크기와 관계없이 반복적인 동작만을 수행하게 되고, 다양한 프레임 크기에 따른 제어 신호들과 반복 복호에 관련된 신호들은 프레임 제어 블록과 반복 제어 블록에서 각각 생성한다. 설계된 제어부의 제어 흐름도를 그림10에 나타내었다.

실제적으로 복호기 부제어 블록의 동작은 프레임의 크기와 상관 없이 프레임 상의 위치에 따른 7가지 종류의 초기화 동작만을 결정해주면 되며 이것을 표2에 나타내었다. 따라서, 프레임-제어 블록은 프레임 크기에 대한 정보를 외부로부터 전달 받은 후, 프레임 크기에 맞는 부분블록의 순서를 선택하여 부제어 블록에 알리고 부제어 블록은 순서에 맞는 7가지 제어 동작 중에 한가지를 실행하게 된다. 그리고 반복 제어 블록의 제어에 따라 정해진 반복 복호 횟수 만큼 수행한 뒤 결과를 출력하게 된다.

이러한 다층 제어 구조는 다양한 프레임 크기를 가지는 구조에 대하여 부분블록 연산을 재배열함으로써 간단히 적용할 수 있으며, 또한 block-wise 구조와 다양한 프레임 크기를 동시에 가지는 시스템에서 보다 효율적으로 동작한다.

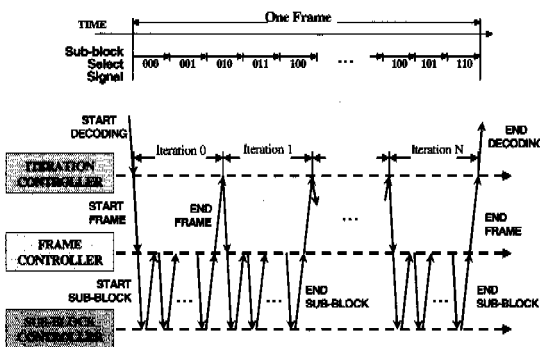


그림 10. Block-wise MAP 복호기의 제어 흐름도

## V. 설계된 터보 코드 복호기의 성능

이 장에서는 본 논문에서 설계된 터보 코드 복호기의 성능을 분석한다. 첫째, 설계된 복호기의 복잡도를 비교하며, 둘째, 복호기의 복호 속도를 비교한다. 그리고 마지막으로 구현된 복호기의 BER 성능을 이상적인 경우와 비교 설명한다.

### 1. 복호기의 복잡도

MAP 알고리즘을 이용하여 SISO 모듈을 구현하면 SOVA를 이용한 SISO 모듈의 같은 복호 속도를 내기 위해서 두 개의 재귀 상태 메트릭 연산기(순방향, 역방향)가 필요하고, 상태 메트릭을 저장하기 위한 많은 메모리도 요구된다. 하지만 block-wise MAP 알고리즘을 사용하면 메모리 요구량을 줄이고, 본 논문에서 제안된 시분할 파이프라인 구조를 적용하여 연산기의 개수를 줄이며, MAP 복호기는 하나의 재귀 상태 메트릭 연산기만 사용하여 SOVA 수준의 복호 속도와 메모리 요구량을 가질 수 있다. 터보 코드의 반복 복호 알고리즘에 따른 6,144 비트 프레임 복호에 필요한 메모리 요구량을 표3에 나타내었다. 표3에 보인 바와 같이 설계된 복호기는 MAP 알고리즘을 SOVA 수준의 복잡도로서 구현하였음을 볼 수 있다.

### 2. 복호기의 복호 속도

복호기의 설계에 있어서 하드웨어의 복잡도와 복호 속도 사이에는 trade-off 관계가 성립한다. 기존의 MAP 복호기의 경우에는 재귀 상태 메트릭 연산기의 수를 줄이면 복호 시간이 늘어나게 된다. 하지만 제안된 시분할 파이프라인 구조를 적용하면 한 개의 변형된 재귀 연산기만 사용하고도 세 개의 재귀 연산기를 사용한 경우와 같은 복호 속도를 가진다.

또한 block-wise MAP 알고리즘의 사용으로 재귀 상태 메트릭 연산과 외부 정보의 연산 사이에 파이프라인 구조의 적용이 가능하여 오히려 복호 시간을 줄일 수 있다. 표4에서 터보 코드 6,144 정보 비트 프레임 복호를 위한 SISO 복호 연산을 한번 수행에 필요한 시간을 비교하였다.

표4의 결과를 보면 같은 복잡도를 가지는 조건하에서 block-wise MAP 알고리즘만을 사용하여 복호기를 설계하면 메모리의 요구량은 줄어들지만 복호에 걸리는 시간은 오히려 늘어나는 것을 볼 수 있다. 이것은 트레이닝 연산으로 인하여 상태 메트릭의 계산량이 늘어나는 것에 기인한다.

표 3. 터보 코드 복호기 메모리 요구량 비교

SOVA 복호기	일반적인 MAP 복호기	Block-wise MAP 복호기	시분할 파이프라인 Block-wise MAP 복호기
$6144 \times 4 \times 4 =$ 98,304 bits	$6144 \times 4 \times 4 =$ 98,304 bits	$6144 \times 4 \times 4 =$ 98,304 bits	$6144 \times 4 \times 4 =$ 98,304 bits
0 bits	$6144 \times 2 \times 8 \times 8 =$ 786,432 bits	$32 \times 2 \times 8 \times 8 =$ 4,096 bits	$32 \times 2 \times 8 \times 8 =$ 4,096 bits
$6144 \times 2 \times 6 =$ 73,728 bits	$6144 \times 2 \times 6 =$ 73,728 bits	$6144 \times 2 \times 6 =$ 73,728 bits	$6144 \times 2 \times 6 =$ 73,728 bits
0 bits	0 bits	0 bits	288 = 128 bits
172,032 bits	958,464 bits	176,128 bits	176,256 bits

◆ Sampling:Q=16, Extrinsic Info:6 bits, State Metric:8 bits

표 4. 터보 코드 복호기의 복호 시간의 비교

	SOVA 복호기	일반적인 MAP 복호기	Block-wise MAP 복호기	시분할 파이프라인 Block-wise MAP 복호기
상태 메트릭 연산시간	$6144 \times n =$ $6,144 \times n$ clocks	$6144 \times 3 \times 2 =$ 36,864 clocks	$32 \times (192+2-7) \times 3 \times 3 =$ 53,856 clocks	$32 \times (192+2) \times 3 =$ 18,624 clocks
외부 정보 연산시간	$32 \times n =$ $32 \times n$ clocks	$6144+15 =$ 6,159 clocks	15 = 15 clocks	15 = 15 clocks
총 계	$6,174 \times n$ clocks	43,023 clocks	53,871 clocks	18,639 clocks

- ◆ 일반적으로,  $2 < n < 3$  (n : 스테이트 메트릭 update에 소요되는 시간 / 클럭)
- ◆ 동일한 복잡도 가지고 있을 경우 (하나의 상태 메트릭 연산기만을 사용.)

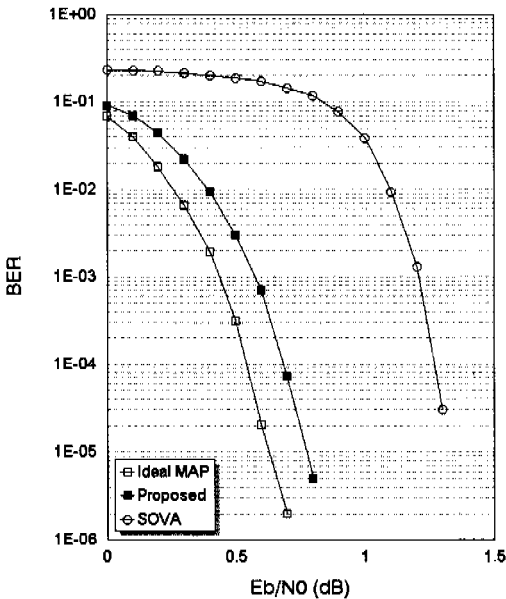


그림 11. 설계된 터보 코드 복호기의 성능

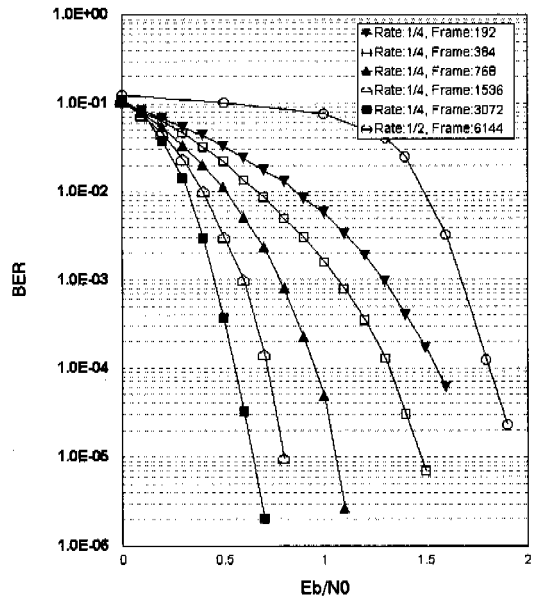


그림 12. cdma-2000 환경에서 설계된 복호기의 성능



하지만 block-wise MAP 알고리즘에 제안된 시분할 파이프라인 구조를 적용하면 복호 시간이 줄어들어 복호기의 성능이 개선됨을 알 수 있다.

### 3. BER 성능

그림11에서 코드율인 1/3이고, 프레임 크기가 3,072비트인 cdma-2000의 터보 코드를 AWGN 환경에서 5번 반복 복호한 경우의 모의 실험 결과를 나타내었다. 모의 실험은 설계된 복호기의 모든 하드웨어를 고려하여 설계된 복호기와 동일한 연산을 수행하여 실시하였다. 그림11에서 보인 바와 같이 보면 설계된 터보 코드 복호기는 MAP 알고리즘을 적용함으로써 SOVA를 사용한 경우보다 0.5dB 정도의 좋은 성능을 보인다. 그리고, 모든 하드웨어를 위한 모든 설정을 적용하였음에도 이상적인 MAP 복호기를 적용한 경우에 비하여 약 0.1~0.2 dB의 성능 저하만을 가지는 것을 볼 수 있다. 설계된 터보 코드 복호기를 사용하여 cdma-2000 시스템의 다양한 코드율과 프레임 크기에 적용한 경우, 복호기의 성능을 그림12에 나타내었다.

## VI. 결론

MAP 알고리즘을 이용한 터보 코드 복호 방법은 최적의 성능을 보임에도 불구하고 MAP 알고리즘 구현을 위한 하드웨어의 복잡도로 인하여 실제적인 적용에 많은 어려움을 가지고 있다. 오랜 시간동안 MAP 복호기 복잡도를 줄이기 위한 여러 가지 연구가 있었고 많은 성과를 거두었다. 하지만 알고리즘에 대한 연구에 비해 효율적인 구조에 대한 연구는 상대적으로 많이 이루어지지 못하였다.

본 논문에서는 MAP 복호기를 효율적으로 설계하기 위한 방법에 대하여 연구하였다. 본 논문에서 제안된 MAP 복호기는 확률비를 이용한 브랜치 메트릭의 계산법을 사용하여 브랜치 메트릭 연산기와 Max 연산 보정기의 복잡도를 절반 정도로 감소시켰다. 그리고 block-wise MAP 알고리즘을 적용하여 상태 메트릭을 저장하기 위한 메모리 요구량과 역방향 재귀 연산을 기다리기 위한 복호 시간의 지연을 현저히 줄이고, block-wise MAP 알고리즘에 의하여 늘어나는 연산량을 해결하기 위하여 파이프라인 구조의 시분할 처리 방법을 제안하였다. 제안된 방법은 재귀 상태 메트릭 연산의 재귀적 구조라는 제약을 극복하여 변형된 하나의 재귀 상태 메트

릭 연산기만으로 세 개의 연산기에서 처리하던 연산량을 시간 지연 없이 수행하도록 하였다.

제안된 MAP 복호기는 SOVA 복호기 정도의 복잡도만을 가지면 복호 속도도 SOVA 복호기에 크게 뒤지지 않는다. 설계된 터보 코드 복호기는 약 20k gates의 복잡도와 약 4k 비트의 내부 메모리만을 요구하며, 하나의 FPGA 소자와 3개의 메모리 소자만을 이용하여 구현되었다. 그리고, FPGA 상에서 8번 반복 복호 하였을 때에 약300kbps의 복호 능력을 가지며, AWGN 환경에서 MAP 알고리즘을 적용한 이상적인 터보 코드 복호기에 약 0.1~0.2 dB의 성능 저하만을 가진다.

## 참고 문헌

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes," Proceeding of ICC' 93, Geneva, Switzerland, pp. 1064-1070, May1993.
- [2] J. Y. Couleaud, High Gain Coding Schemes for Space Communications, ENSICA Final Year Report, University of South Australia, The Level, Australia, September 1995.
- [3] G. Battail, C. Berrou, and A. Glavieux, Pseudo-Random Recursive Convolutional Coding for Near-Capacity Performance, Comm. Theory Mini-Conference, GLOBECOM '93, Houston, Texas, December 1993.
- [4] Sergio Benedetto and Guido Montorsi, "Unveiling Turbo-Codes: Some Results on Parallel Concatenated Coding Schemes," IEEE Trans. on Inform. Theory, vol. IT-42, pp. 409-428, Mar. 1996.
- [5] J. Hagenauer and P. Hoehner, "A Viterbi Algorithm with Soft-Decision Outputs and its Applications," in Proc., IEEE Globecom Conf.(Dallas, TX, Nov. 1989), pp. 1680-1686, 1989.
- [6] Joachim Hagenauer, Elke Offer, and Lutz Papke, "Iterative Decoding of Binary Block and Convolutional Codes," IEEE Trans. on Inform. Theory, vol. IT-42, pp. 429-445, Mar. 1996.
- [7] S. Benedetto, D.Divsalar, G. Montorsi, and F.

Pollara, Soft-Output Decoding Algorithms in Iterative Decoding of Turbo Codes, The Telecommunications and Data Acquisition Progress Report 42-124, October-December 1995, Jet Propulsion Laboratory, Pasadena, California, pp. 63-87, February 15, 1996.

- [8] Steven S. Pietrobon and Adrian S. Barbucescu, "A Simplification of the Modified Bahl Decoding Algorithm for Systematic Convolutional Codes," Int. Symp. Inform. Theory & Its Applic., pp. 1073-1077, Sydney, Australia, Nov. 1994.
- [9] Jean Yves Couleaud, "High Gain Coding Schemes for Space Communication," Signal Processing Research Institute, University of South Australia, Sept. 1995.
- [10] Adrian S. Barbucescu, "Iterative Decoding of Turbo Codes and Other Concatenated Codes," Ph.D. Thesis, University of South Australia, Feb. 1996.
- [11] L. R. Bahl, K. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," IEEE Trans. on Inform. Theory, vol IT-20, pp 284-287, Mar. 1974.
- [12] J. A. Erfanian, S. Pasupathy, and G. Gulak, Reduced complexity symbol detectors with parallel structure for isi channels, IEEE Trans. Comm., vol. 42, pp. 1661-1671, February/March/April 1994.
- [13] P. Robertson, E. Villebrun and P. Hoeher, A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in Log Domain Proceedings of ICC'95, Seattle, Washington, pp.1009-1013, June 1995.
- [14] Andrew J. Viterbi, An Intuitive Justification and a Simplified Implementation of the MAP Decoder for Convolutional Codes, IEEE Journal on Selected Areas in Communications, pp260-264, Feb, 1998.
- [15] 표준 모델 단말기 반복조기에 관한 연구, 중간 연구보고서, ETRI, pp70-73, July, 1998.

박 구 현(Goo-hyun Park)

정회원



1998년 2월 : 단국대학교  
전자공학과 졸업 (공학사)  
2000년 2월 : 연세대학교 대학원  
전기, 컴퓨터공학과 졸업  
(공학석사)  
현재 : 연세대학교 대학원  
전기, 전자공학과  
공학박사과정

<주관심 분야> 이동통신시스템, 대역확산통신, 채널 코딩

윤 석 현(Suk-hyon Yoon)

정회원



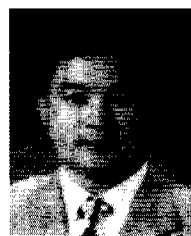
1994년 2월 : 연세대학교  
전자공학과 졸업 (공학사)  
1996년 2월 : 연세대학교 대학원  
전자공학과 졸업  
(공학석사)  
2000년 8월 : 연세대학교 대학원  
전기, 컴퓨터공학과 졸업  
(공학박사)

현재 : LG전자 정보통신 차세대통신연구소

<주관심 분야> 이동통신시스템, 채널코딩, 디지털 신호처리

강 범 주(Bub-joo Kang)

정회원



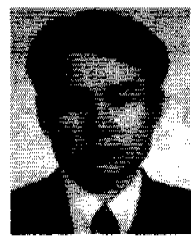
1983년 2월 : 경희대학교  
전자공학과 졸업 (공학사)  
1985년 8월 : 연세대학교 대학원  
전자공학과 졸업 (공학석사)  
1999년 2월 : 연세대학교 대학원  
전자공학과 졸업 (공학박사)

현재 : 한국전자통신연구원(ETRI)

<주관심 분야> 이동통신시스템, 대역확산통신, 채널 코딩, 디지털 신호처리

진 익 수(Ik-soo Jin)

정회원



1987년 2월 : 연세대학교  
전기공학과 졸업 (공학사)  
1989년 2월 : 연세대학교 대학원  
전기공학과 졸업(공학석사)  
1995년 8월 : 연세대학교 대학원  
전기공학과 졸업(공학박사)

현재 : 한국전자통신연구원(ETRI)

<주관심 분야> 대역확산통신, CDMA 이동무선통신,  
변복조, 채널코딩

홍 대 식(Dae-sik Hong)

종신회원

한국통신학회 논문지 제21권, 제8호 참조

강 창 언(Chang-eon Kang)

종신회원

한국통신학회 논문지 제23권, 제2호 참조