

# 반복 복호 횟수 감소를 통한 저전력 LDPC 복호기 설계

준회원 이준호\*, 박창수\*, 정회원 황선영\*

## Design of a Low-Power LDPC Decoder by Reducing Decoding Iterations

Jun-Ho Lee\*, Chang-Soo Park\* Associate Members, Sun-Young Hwang\* Regular Member

### 요약

LDPC 부호는 4G 이동통신 시스템에 적합한 오류 정정 부호이다. 그러나 알고리즘의 특성상 좋은 BER 성능을 위해서는 반복 복호에 의한 많은 연산량이 요구된다. 본 논문에서는 복호 지연과 전력 소모에 대한 복호기의 성능을 증가시키기 위하여 반복 복호 횟수를 줄이는 알고리즘에 대하여 제안한다. 제안된 알고리즘은 현재 LLR 복호값과 이전 LLR 복호값 사이의 변화를 측정하고 변화 방향을 예측하며, 패리티 검사식을 만족시켜 수렴속도를 높이고 LLR 값의 sign 비트를 반전시킨다. 실험결과, 제안한 방법은 BER 성능의 감소 없이 반복 복호 횟수를 약 33% 정도 줄이는 것이 가능하며 감소된 반복 복호 횟수에 비례하여 소모 전력도 감소시킬 수 있다.

**Key Words** : Low-Density Parity-Check (LDPC) codes, iterative decoding, low power algorithm

### ABSTRACT

LDPC (Low Density Parity Check) code, which is an error correcting code determined to be applied to the 4th generation mobile communication systems, requires a heavy computational complexity due to iterative decodings to achieve a high BER performance. This paper proposes an algorithm to reduce the number of decoding iterations to increase performance of the decoder in decoding latency and power consumption. Measuring changes between the current decoded LLR values and previous ones, the proposed algorithm predicts directions of the value changes. Based on the prediction, the algorithm inverts the sign bits of the LLR values to speed up convergence, which means parity check equation is satisfied. Simulation results show that the number of iterations has been reduced by about 33% without BER performance degradation in the proposed decoder, and the power consumption has also been decreased in proportional to the amount of the reduced decoding iterations.

### I. 서론

4G 이동통신 시스템과 고속 통신 등 최근의 디지털 이동통신 환경은 멀티미디어 데이터의 폭발적인 수요 증가로 신뢰성 높은 고속 데이터 전송이 요구되고 있다. 고속의 데이터 전송, 가격 대비 전

송률 최적화와 주파수 효율성의 향상 등을 특징으로 하는 4G 이동통신 시스템의 핵심기술로는 OFDMA (Orthogonal Frequency Division Multiple Access), SDR (Software Defined Radio), MIMO (Multiple-Input Multiple-Output), LDPC (Low Density Parity Check) 부호, Smart Antenna 등이

※ 본 논문은 2007년도 「서울시 산학연 협력 사업」의 「나노 IP/SoC 설계 기술 혁신 사업단」의 지원으로 이루어졌으며 Simulation 및 synthesis에 IDEC에서 지원받은 CAD tool을 사용하였음.

\* 서강대학교 전자공학과 대학원 CAD & ES연구실

논문번호 : KICS2007-02-050, 접수일자 : 2007년 2월 9일, 최종논문접수일자 : 2007년 8월 2일

있다. 그중에서 광대역 데이터와 고속 패킷 전송을 지원하기 위한 강력한 오류 정정 부호화 기법은 차세대 이동통신 시스템에 있어서 핵심적인 요소이다. 4G 이동통신 시스템에서 요구되고 있는 상당히 낮은 오류 확률에서 터보 부호는 성능의 한계를 나타내고 있어, 그래프를 기반으로 한 새로운 종류의 부호 방법에 대한 관심이 높아지게 되었다<sup>[1]</sup>. 이에 따라 새롭게 주목을 받게 된 오류 정정 부호가 LDPC 부호(Low Density Parity Check Code)이다<sup>[2]</sup>. LDPC 부호는 Shannon의 채널 용량 한계에 거의 근접하는 성능을 보이므로 데이터 BER  $10^{-6}$ 급, 영상 BER  $10^{-9}$ 급 이하의 고품질의 신뢰도를 요구하는 4G 이동통신 시스템에 적합한 부호화 방식이다<sup>[3][4]</sup>.

LDPC 부호는 1960년대 Gallager에 의해 처음 제안되었으며<sup>[5]</sup> 반복 부호가 가능한 선형 블록 부호(Linear Block Code)로써 터보부호와 마찬가지로 Shannon limit에 가까운 성능을 가진 부호이다. 알고리즘이 복잡하고 부호의 연산에 필요한 저장 매체와 부호 과정의 연산을 구현할 수 없어 40여년 동안 관심을 받지 못했으나 1996년에 MacKay와 Neal에 의해 구현되었다<sup>[4][6]</sup>. Tanner는 LDPC 부호들을 일반화하여 이분(Bipartite) 그래프를 사용하여 표현하였고 합곱(Sum-Product) 알고리즘을 소개하면서 cycle free 그래프에서 최적임을 증명하였다<sup>[7]</sup>. LDPC 부호는 패리티 검사 행렬인 H 행렬의 원소들이 대부분 '0'인 선형 블록 부호이며 패리티 검사식을 이용하여 확률적인 반복 부호방법을 사용함으로써 성능의 향상을 가져온다. 충분히 큰 블록크기를 가진 부호기에서 오류가 있는 블록은 패리티 검사식을 만족시키지 못하므로 확률적인 반복 부호를 거쳐 정정되지 않은 오류들을 대부분 검출할 수 있다<sup>[7][8]</sup>. 즉, 오류가 포함된 상태로 부호를 마친 경우는 거의 존재하지 않으며, 부호가 성공하거나 오류로 인해 부호 실패하는 두 가지 경우가 대부분이다. 부호율의 가변이 간단하고 그래프 기반의 반복 부호 방법을 사용함으로써 터보부호에 비해 ASIC이나 FPGA를 사용한 VLSI 구현에 용이한 구조를 갖게 되어 완전 병렬 처리에 의한 고속 전송이 가능하다<sup>[9]</sup>. 이는 LDPC 부호의 가장 큰 장점 중의 하나이며 1Gb/s의 높은 데이터 전송률을 요구하는 고성능의 자기기록장치, 10Gb/s의 광통신이나 고속 무선 통신 환경과 같은 응용분야에 적용이 가능하다.

Blanksby는 이분 그래프 상의 데이터 교환을 직접적으로 맵핑하여 일반적인 환경에서 1Gb/s의 전송률을 갖는 완전 병렬화한 LDPC 부호기를 설계하

였다<sup>[9]</sup>. 전력 소모는 데이터 스위칭의 활성화도에 비례한다. 낮은 SNR에서는 수신신호에 대한 복호가 쉽지 않으므로 최대 반복 부호 횟수만큼 매번 반복 부호를 시도하여 데이터 스위칭의 활성화도가 높아지며, 높은 SNR에서는 최대 반복 부호 횟수보다 적은 반복 횟수로도 복호가 가능하므로 데이터 스위칭의 활성화도가 낮아지게 된다. 좋은 BER 성능을 위해서는 블록 크기가 커져야하고 반복 부호 횟수가 많아져야 한다. 큰 블록 크기에서는 많은 연산량과 저장 공간이 요구되며 복호지연도 늘어난다<sup>[6][10]</sup>. 반복 부호 횟수가 많아질 경우에도 복호지연이 늘어나고 많은 전력소모와 함께 전송률이 떨어지는 문제점이 발생한다<sup>[11]</sup>.

본 논문에서는 LDPC 복호기에 버퍼와 덧셈기, 그리고 비교기를 추가하여 반복 부호 알고리즘의 중간 복호결과를 분석 예측하고 부호 값의 sign을 후보정하여 반복 부호 횟수를 줄이는 방안을 제시한다. 낮은 SNR 환경에서도 반복 부호 횟수를 줄임으로써 전체적인 부호기의 복잡도와 전력소모를 감소시킬 수 있다<sup>[9][11]</sup>. 본 논문의 구성은 다음과 같다. II절에서는 LDPC 부호의 알고리즘과 동작 원리에 대해 설명하고, III절에서는 제안한 반복 부호 횟수 감소를 위한 LDPC 부호 알고리즘을 제시한다. IV절에서 실험을 통해 제안한 방법의 성능을 검증하고 마지막으로 V절에서 결론을 맺는다.

## II. LDPC 부호 알고리즘

LDPC 부호의 부호화 과정에서는 채널을 통해 부호화 과정으로부터 수신된 부호어에 대한 확률 값을 그림 1에 보인 것과 같이 이분 그래프 상에서 비트 노드와 체크 노드 사이에 반복적으로 교환하고 갱신한다. 매회 복호된 부호어 c에 대해 패리티 검사 행렬 H와 패리티 검사를 수행하여 패리티 검사식  $H \cdot c^T = 0$ 을 만족하면, 미리 정해진 최대 반복 부호 횟수를 만족하지 않아도 부호를 완료하고 반복 부호를 멈춘다<sup>[2][6]</sup>.

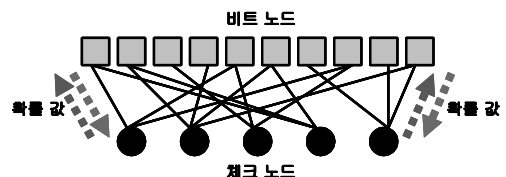


그림 1. 이분 그래프 상의 데이터 교환.

2.1 이분 그래프와 데이터 교환

복호화 과정에서 채널로부터 얻는 LLR 값을 이용하여 연관정할 수 있는 합곱 알고리즘은 패리티 검사 행렬  $H$  상의 '1'의 위치와 개수에 의해 결정된 그림 1의 이분 그래프로 나타낼 수 있다<sup>[12]</sup>. 이분 그래프의 노드에는 LLR 값을 갖는 비트 노드와 체크 노드라는 두 가지 종류의 변수가 할당된다. 합곱 알고리즘은 이분 그래프 상에서 나뉜 두 노드 사이에 확률 값을 전파하는 데이터 교환과 같은 의미로 해석된다. 합곱 알고리즘은 많은 변수들로 이루어진 복잡한 전체 함수의 값을 직접 계산하는 것이 어려우므로 간단한 지역 함수들의 곱으로 구한다. 각 지역 함수들이 작은 수의 변수의 함수라면 이를 이용하여 전체 함수의 값을 효과적으로 계산할 수 있다<sup>[12]</sup>. 그림 1에서 체크 노드는 지역 함수 노드에 해당하고, 수신된 부호어의 확률 값을 이루는 비트 노드는 변수 노드에 해당한다. 합곱 알고리즘은 이분 그래프 상에서 확률에 대한 데이터를 연결된 노드들 사이에 서로 교환하면서 각 노드들이 최대 LLR 값을 만족시키는 부호어에 수렴하도록 반복적으로 갱신하는 과정에 사용된다. 합곱 알고리즘은 로그 영역에서 곱셈연산 과정을 덧셈연산으로 변환하여 연산량과 하드웨어 면적을 줄임으로써 실제 구현 시 연산을 효과적으로 실행할 수 있다<sup>[13][14][15]</sup>.

2.2 복호화 과정

LDPC 복호의 데이터 교환은 반복적으로 수행되며, LLR을 이용한 단계별 복호 방법은 다음과 같다<sup>[14]</sup>.

단계 1 (초기화 단계): 모든 비트 노드를 수신 신호에 대한 LLR 값으로 초기화한다. 반복 복호 횟수는 '1'로 초기화한다. 식 (II-1)은  $N$ 의 크기를 갖는 수신 신호  $y$ 에 대한 LLR 값을 보이고 식 (II-2)는 Check-to-Bit 단계에서 저장되는 행렬  $r$ 에 대한 LLR 값의 초기화를 보인다.

$$LLR(y_i) = \log \frac{P(y_i = 1)}{P(y_i = 0)} \quad \text{식 (II-1)}$$

$$LLR^{(0)}(r_{ji}) = 0 \quad \text{식 (II-2)}$$

단계 2 (Bit-to-Check 단계): 비트 노드 연산과정이며, 비트 당 현재의 사후 확률 추정치 연산을 한다. 비트 노드의 LLR 값을 비트 노드로부터 이분 그래프 상에 연결된 체크 노드로 전달한다.

$$LLR^{(k)}(q_{ji}) = \sum_{j' \in M(i) \setminus \{j\}} LLR^{(k-1)}(r_{j'i}) + LLR(y_i) \quad \text{식 (II-3)}$$

단계 3 (Check-to-Bit 단계): 체크 노드 연산과정이며, 각 체크 노드는 모든 패리티 검사 행렬  $H$ 의 행에 대한 패리티 검사를 수행한다. 체크 노드의 LLR 값을 체크 노드로부터 이분 그래프 상에 연결된 비트 노드로 전달한다.

$$LLR^{(k)}(r_{ji}) = (-1)^{\left( \prod_{i' \in N(j) \setminus \{i\}} \text{sgn}(LLR^{(k)}(q_{ji'})) \right)} \cdot \Psi \left( \prod_{i' \in N(j) \setminus \{i\}} \Psi(LLR^{(k)}(q_{ji'})) \right) \quad \text{식 (II-4)}$$

$$\Psi(x) = -\log \left( \tanh \left( \frac{x}{2} \right) \right), \Psi = \Psi^{-1} \quad \text{식 (II-5)}$$

단계 4 (연산 결과 출력 단계): 복호기로부터 추정 결과 값을 출력한다. 식 (II-7)은 비트 노드에서 출력 노드로 전달하는 데이터의 사후 확률 LLR 값을 보인다. 식 (II-7)의 추정 결과 값에 대해 0을 기준으로 크면 '1' 작으면 '0'으로 복호 비트를 추정한다. 여기서  $LLR(q_i)$ 는 사후 확률 값,  $LLR(r_{ji})$ 는  $i$ 번째 비트 노드에 연결된 체크 노드들의 값,  $y$ 는 수신 신호,  $i$ 는  $N$ 의 크기를 갖는 비트 노드의 인덱스,  $j$ 는  $M$ 의 크기를 갖는 체크 노드의 인덱스,  $j'$ 는 해당 인덱스를 제외한다는 의미를 갖는다.

$$LLR^{(k)}(q_i) = \sum_{j' \in M(i)} LLR^{(k)}(r_{ji'}) + LLR(y) \quad \text{식 (II-6)}$$

$$\hat{v}_i^{(k)} = \begin{cases} 1 & \text{if } LLR^{(k)}(q_i) > 0 \\ 0 & \text{if } LLR^{(k)}(q_i) < 0 \end{cases} \quad \text{식 (II-7)}$$

단계 5 (반복 복호 결정 단계): 복호 과정 수행 후 각 비트의 확률 추정 결과 값이 패리티 검사식  $H \cdot c^T = 0$  식을 만족시키면 복호를 중단한다. 만약, 만족하지 못하면 단계 2부터 단계 4까지 반복하며, 정해진 최대 반복 복호 횟수만큼 반복 복호를 수행한 후 종료한다.

III. 제안된 반복 복호 알고리즘

LDPC 복호기의 전체 복잡도가 반복 복호 횟수

에 의존하므로 복잡성의 증가에 따른 복호 지연과 실시간 처리의 어려움이라는 문제점을 가지고 있다 [11]. II절에서 언급된 LDPC 복호 과정에서는 최대 반복 복호 횟수가 미리 정해져있으며 매회 복호된 부호어  $c$ 에 대한 패리티 검사를 수행하여 패리티 검사식  $H \cdot c^T = 0$  을 만족하면 최대 반복 복호 횟수를 만족하지 않더라도 반복 복호를 멈춘다. 한번의 복호 과정은 이분 그래프 상에 연결된 비트 노드와 체크 노드 사이에 확률 값을 가진 데이터를 반복적으로 교환하여 복호값을 보다 신뢰성 있는 값으로 갱신한다. 반복 복호 횟수를 줄임으로써 전체적인 복호기의 복잡도와 복호 지연과 더불어 전력 소모도 줄일 수 있다. 본 절에서는 반복 복호 과정에서 현재 LLR 복호값과 이전 LLR 복호값 사이의 변화를 측정하고 변화의 방향을 예측하며, 패리티 검사식을 만족시켜 수렴속도를 높이도록 LLR 값의 sign 비트를 반전시키는 방법을 제안한다.

### 3.1 LLR 값의 변화

그림 2는 AWGN(Additive White Gaussian Noise) 채널에서 BPSK(Binary Phase Shift Keying) 변조방식으로 전송했을 때, 오류가 발생한 비트의 반복 복호마다 변화하는 LLR 값을 보인다. 복호 비

트는 LLR 값의 sign이 양수이면 '1', 음수이면 '0'으로 결정되며, 반복 복호 과정에서 오류가 발생한 비트의 LLR 값은 sign이 복호 전과 반대가 되는 방향의 특정 값으로 수렴한다<sup>[14]</sup>. 세 번째 반복 복호 단계는 LLR 값의 sign에 대응하는 복호 비트가 패리티 검사식을 만족하여 오류가 정정된 경우이다. 오류가 정정된 LLR 값의 변화 패턴을 보면 LLR 값 변화를 나타내는 기울기가 오류가 발생한 LLR 값 sign의 반대방향으로 진행되는 것을 알 수 있으며, 변화 방향을 예측할 수 있다면 반복 복호의 수렴 속도를 향상시켜 반복 복호 횟수를 감소시킬 수 있다.

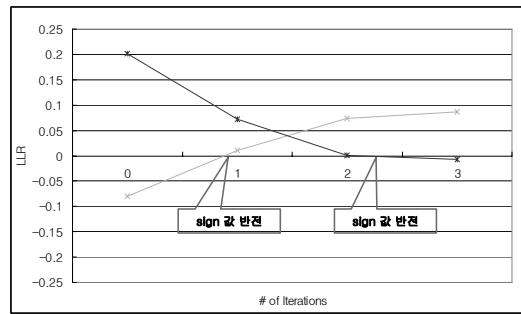
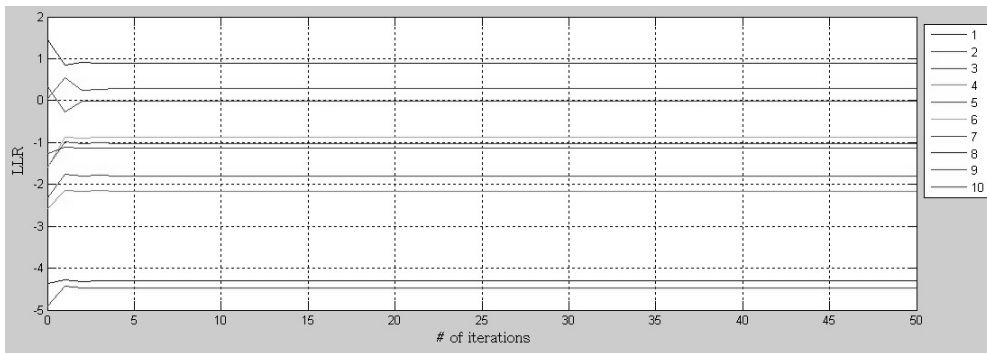
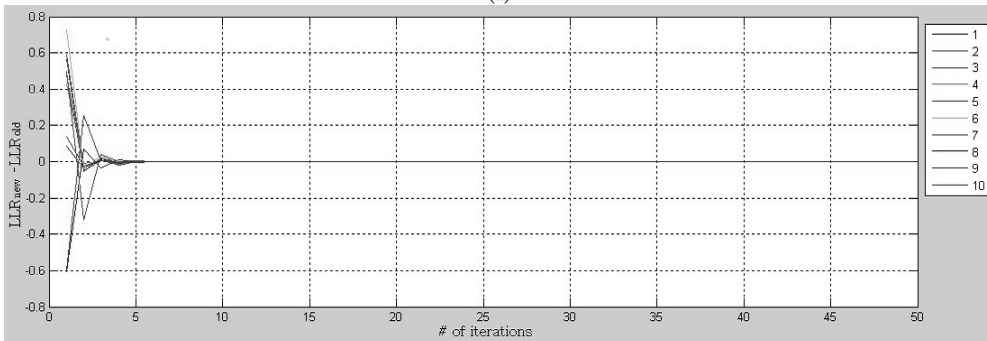


그림 2. 반복 복호 과정에서 LLR 값의 변화 예시.



(a)



(b)

그림 3. 반복 복호 과정에서 복호 실패한 LLR 값의 변화와 변화량. (a) LLR 값의 변화, (b) LLR 값의 변화량.

그림 3은 반복 복호 과정에서 복호 실패한 LLR 값의 변화와 변화량을 보인다. 그림 3 (a)는 반복 복호 중에 오류가 정정되지 못하고 미리 정해진 최대 반복 복호 횟수만큼 반복 복호됨을 보인다. 그림 3 (b)는 그림 3 (a)의 LLR 값의 변화량을 나타낸 그래프이며 변화량이 감소하여 일정 반복 복호 횟수 이후에는 변화가 거의 없음을 보인다. 반복 복호 초기에만 LLR 값의 변동 폭이 크며 반복 복호가 진행될수록 변동 폭이 감소하는 특성을 갖는다.

### 3.2 제안된 알고리즘

제안된 알고리즘은 random 정보 비트를 부호화하여 AWGN 채널을 통해 전송된 부호어의 반복 복호 과정에서 새 복호값  $L_{new}$ 와 이전 복호값  $L_{old}$  사이의 변화량  $\Delta L (=L_{old}-L_{new})$ 를 측정하고 분석하여  $L_{new}$ 와  $L_{old}$  사이의 변화 방향을 예측한다. 그림 2의 오류가 정정되는 비트의 복호 과정은 복호값이 반대 sign 방향으로 변화하는 패턴을 보이므로 변화 방향을 예측하여 sign 비트를 미리 반전시킴으로써 패리티 검사식을 만족시켜 반복 복호 횟수를 감소할 수 있다. 그림 4에 반복 복호 과정에서 복호값 LLR의 변화 패턴을 보인다. 그림 4 (a)는  $L_{old}$ 와  $L_{new}$ 의 sign이 같고  $|L_{new}|$ 가  $|L_{old}|$ 보다 작으므로  $L_{new}$ 가 반대 sign 방향으로 변화할 수 있는 패턴을 보이며 sign 반전의 가능성이 있음을 보인다. 그림 4 (b)는 그림 4 (a)와 기울기 패턴은 유사하나  $L_{new}$ 의 sign이 이미 반전되었으므로 제안 알고리즘을 적용하지 않는다. 그림 4 (c)는  $L_{old}$ 와  $L_{new}$  sign이 같지만  $|L_{new}|$ 가  $|L_{old}|$ 보다 크므로 sign이 반전의 가능성이 없음을 알 수 있다. 따라서 이 세 경우 중 그림 4 (a)의 패턴만이 sign 반전 가능성이 있으며 이 경우에 한해서만 sign을 반전시킨다. 복호값  $L_{old}$ 와  $L_{new}$  사이의 변화량은  $\Delta L$ 이고, Slope는  $\Delta L$ 의 sign

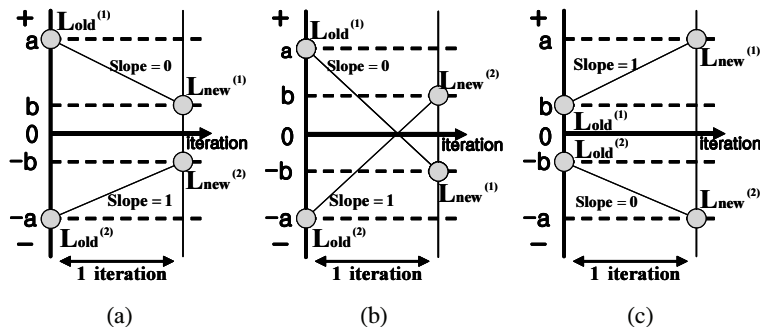


그림 4. 반복 복호 과정에서 복호값 LLR의 변화 패턴. (a)  $|L_{new}| < |L_{old}|$ ,  $\text{sgn}(L_{new}) = \text{sgn}(L_{old})$ , (b)  $|L_{new}| < |L_{old}|$ ,  $\text{sgn}(L_{new}) \neq \text{sgn}(L_{old})$ , (c)  $|L_{new}| > |L_{old}|$ ,  $\text{sgn}(L_{new}) = \text{sgn}(L_{old})$ .

이 양수면 ‘1’, 음수면 ‘0’이다. 그림 4 (a)에서 Slope와 복호값  $L_{new}$ 의 sign이 반대이므로,  $\text{XOR}(\text{Slope}, \text{sgn}(L_{new})) = 1$ 로 나타낼 수 있다. 예를 들어, 복호값  $L_{old}$ 와  $L_{new}$ 의 LLR 값이 +0.75와 +0.25일 때, 그림 4 (a)에서 보인바와 같이  $L_{new}$ 가 반대 sign으로 변화하려는 패턴을 보인다. 변화량  $\Delta L$ 은 -0.5이고  $L_{new}$ 는 +0.25이므로, Slope는 ‘0’의 값을 갖고  $\text{sgn}(L_{new})$ 는 ‘1’의 값을 갖게 되어  $\text{XOR}(\text{Slope}, \text{sgn}(L_{new})) = 1$ 로 보일 수 있다.

### 3.3 Threshold 적용

Threshold  $\Theta_{LLR}$ 를 적용하면  $|L_{new}|$ 가  $\Theta_{LLR}$ 보다 작을 때만  $L_{new}$ 의 sign 비트를 반전하므로 보다 정확하게 변화 방향을 예측할 수 있다. Threshold  $\Theta_{LLR}$ 는 반복 복호 횟수에 영향을 받지 않고 고정된 값을 갖는다. 제안된 알고리즘은  $\text{XOR}(\text{Slope}, \text{sgn}(L_{new})) = 1$ 과  $|L_{new}| < \Theta_{LLR}$  조건을 만족할 때 sign 비트를 반전하도록 한다. 그림 5는  $|L_{new}|$ 가  $\Theta_{LLR}$ 보다 작으며  $\text{XOR}(\text{Slope}, \text{sgn}(L_{new})) = 1$ 를 만족함을 보이며,  $L_{new}$ 의 sign 비트가 예측에 의해 반전되었다. 복호값의 변화 방향을 예측하여  $L_{new}$ 의 sign 비트를 반전시킴으로써 수렴속도를 높여 반복 복호 횟수를 줄일 수 있다.

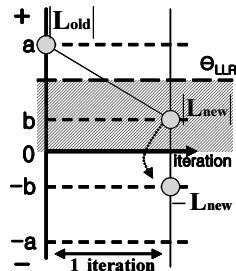


그림 5. Threshold  $\Theta_{LLR}$ 를 만족하여  $L_{new}$ 의 sign이 반전되는 경우.



3.4. 제안된 알고리즘의 하드웨어 구조

제안한 알고리즘의 하드웨어 구조를 그림 6에 제시하였다. LDPC 복호기는 Bit-to-Check 블록과 Check-to-Bit 블록으로 구성되며, 각 구성에서는 비트 노드와 체크 노드의 연산이 사용된다<sup>[9][10]</sup>. 그림 6은 기존 Bit-to-Check 블록의 비트 노드 연산 과정에 비퍼, 비교기, ADDER, XOR와 AND gate를 추가한 구조이다. 여기서,  $q_{ji}$ 와  $r_{ji}$ 는 비트 노드와 체크 노드 연산에서 중간 값들이 저장되는 matrix,  $q_i$ 는 사후 확률 값,  $c$ 는 복호된 추정값,  $y_i$ 는 수신 신호,  $i$ 는  $N$ 의 크기를 갖는 비트 노드의 인덱스,  $j$ 는  $M$ 의 크기를 갖는 체크 노드의 인덱스이다. 그림 6에서 데이터 흐름 ①은 이전 데이터의 사후 확률  $q_i$ 를 나타내며  $L_{new-Lold}$  연산을 위해  $L_{old}$ 를 버퍼에 저장한다. Slope ④는  $L_{old}$  ③과  $L_{new}$  ②를 입력 받은  $L_{new-Lold}$  연산의 결과  $\Delta L$ 이며, MSB인 sign 값을 출력한다.  $\Theta_{LLR}$ 를 SM의 magnitude로 제어하므로  $L_{new}$  ⑤의 2's complement를 SM으로 변환하는 과정을 거친다. 데이터 흐름 ⑥은  $L_{new}$  ⑤의 magnitude이며,  $\Theta_{LLR}$  ⑦과 함께 비교기에 입력된다. 비교기의 출력 ⑧은 ⑥ 값이 ⑦ 값보다 작을 때 '1'을 출력한다. XOR는 Slope ⑩과  $L_{new}$ 의 MSB인  $c$  ⑨를 입력받아 복호값의 변화할 가능성을 판단한다. AND연산은 ⑧ 값과 ⑪ 값을 입력받아 sign 비트 반전 조건을 만족하는지 판정한다.

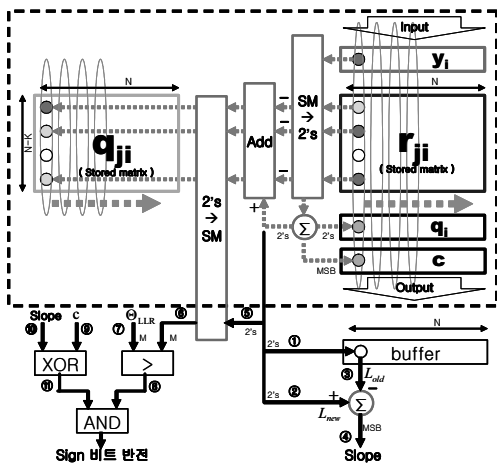


그림 6. 기존 블록에 추가된 제안한 알고리즘의 하드웨어 구조

IV. 실험 결과

복호기의 성능측정을 위한 소프트웨어 시뮬레이션은 Matlab 언어로 구현되었으며, 시스템의 입력

블록은 랜덤한 분포를 갖는 이진값을 발생하여 생성행렬과 곱하는 부호화 과정을 거쳐 부호어를 생성한다. 변조 모듈에서는 BPSK가 사용되어 이진 신호 '0'과 '1'은 각각 '-1'과 '+1'로 변조된다<sup>[16]</sup>. 채널 환경은 가산성 백색 가우시안 노이즈(AWGN)가 사용되었으며, 복조 과정에서는 연관정 값으로 복조한다. LDPC 복호기는 LLR가 적용된 반복 복호를 수행한다. 표 1에는 실험에 사용된 조건을 제시하였다. 본 알고리즘의 성능 실험에서 사용된 패리티 검사 행렬은 모두 네 가지로 블록 크기는 504, 1008, 4000인 패리티 검사 행렬을 사용하였으며 부호율 모두 0.5로, 최대 반복 복호 횟수는 50회로 고정하였다. 또한 실험에 사용된 패리티 검사행렬은 모두 regular 형태로 column weight가 각각 3, 9, 3, 3이며 row weight가 각각 6, 15, 8, 6이다.

표 1. 실험에 사용된 조건

파라미터	패리티검사 행렬			
	H1	H2	H3	H4
블록 크기 (N)	504	504	1008	4000
정보 비트 크기 (K)	252	252	504	2000
패리티 비트 크기 (N-K=M)	252	252	504	2000
열 당 1의 개수 (Wc)	3	9	3	3
행 당 1의 개수 (Wr)	6	15	8	6
최대 반복 복호 횟수	50	50	50	50
부호율	0.5	0.5	0.5	0.5
채널 변조 방식	AWGN, BPSK			
$E_b/N_0$ [dB]	1 ~ 2.5 (0.5 단위)			

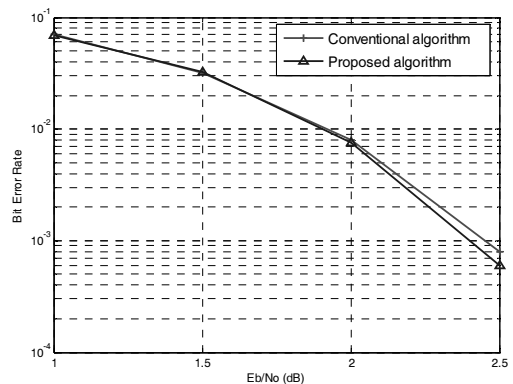


그림 7. BER 성능 비교

그림 7은 패리티 검사행렬 H1에 대한 기존 알고리즘과 제안된 알고리즘의 BER 성능 비교를 보인다. 제안된 알고리즘의 그래프는  $\Theta_{LLR}$ 가 '1'인 sign 반전 알고리즘이 적용된 LDPC 복호기의 BER 성

능 변화를 보인다.  $\Theta_{LLR}$  값은 반복적인 실험을 통해 오류 비트의 LLR 값을 측정한 결과와 제안된 알고리즘을 다양한  $\Theta_{LLR}$  값을 주어 실험한 결과 통계적으로 1의 값일 때 좋은 성능을 예측할 수 있다. 제안된 알고리즘의 BER 성능은 기존 LDPC 알고리즘과 큰 차이가 없다.

그림 8에 각 패리티 검사행렬에 대한 기존 알고리즘과 제안된 알고리즘의 프레임 당 평균 반복 복호 횟수 비교를 보이며 표 2에 반복 회수 감소율을 보인다. 모든 패리티 행렬에 대해  $E_b/N_0$ 가 낮은 dB에서 높은 dB로 증가함에 따라 제안된 알고리즘이 기존 알고리즘보다 상대적으로 적은 프레임 당 평균 반복 복호 횟수를 갖는다. 패리티 검사행렬 H1과 H2는 같은 블록크기와 같은 코드율을 갖는 패리티 검사 행렬이지만 H2의 경우 row와 column

weight의 크기가 더 크다. 그림 8의 (a), (b)와 표 2의 결과를 보면 weight가 큰 패리티 검사행렬 H2가 H1에 비해 기본적인 복호 성능이 더 좋은 것을 볼 수 있으며 복호 성능이 좋은 패리티 검사 행렬을 사용했을 때 알고리즘에 의한 성능 향상이 줄어든 것을 볼 수 있다. 패리티 검사행렬 H1과 H3, H4는 같은 코드율과 비슷한 weight를 가지며 블록 크기가 H1은 504, H2는 1008, H3는 4000로 다른 패리티 검사 행렬이다. 시뮬레이션 결과 그림 8의 (a), (c), (d)와 표 2에 보인 바와 같이 weight가 같은 경우 패리티 검사 행렬인 경우 블록의 크기가 커질수록 성능 개선 폭이 다소 감소하는 것을 확인할 수 있다.

Blanksby는 채널 환경 2.5dB를 고려할 때의 일반적인 환경에서 클럭 주파수 64MHz일 때, 1Gb/s

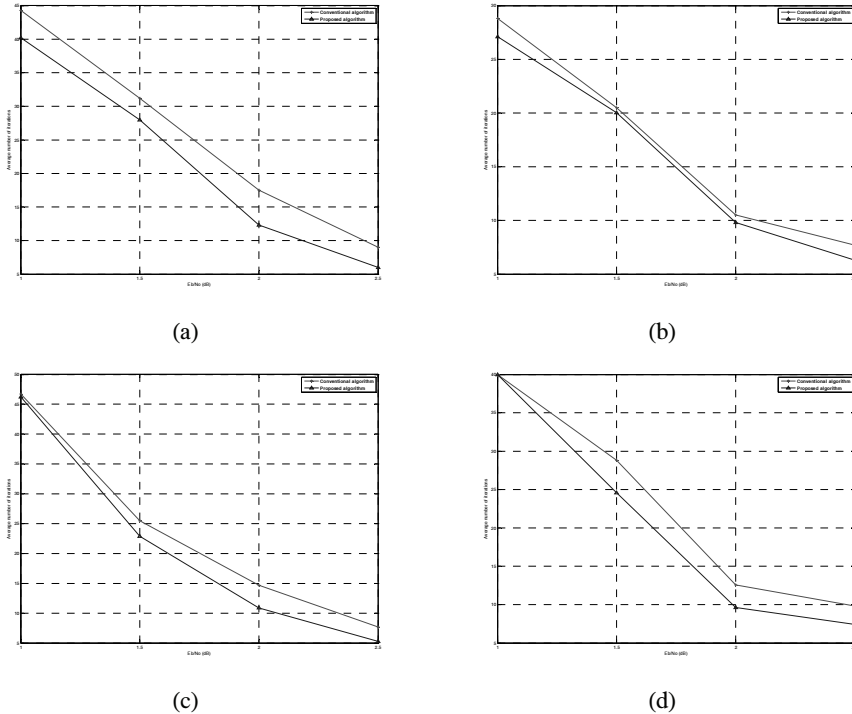


그림 8. 프레임 당 평균 반복 복호 횟수 비교 (a) 패리티 검사 행렬 H1, (b) H2, (c) H3, (d) H4.

표 2. 프레임 당 평균 반복 복호 횟수 비교

알고리즘 Eb/No (dB)	H1			H2			H3			H4		
	기존	제안	감소율 (%)	기존	제안	감소율 (%)	기존	제안	감소율 (%)	기존	제안	감소율 (%)
1.0	44.3	40.2	<b>9.3</b>	28.8	27.1	<b>5.9</b>	46.7	46.2	<b>1.1</b>	40	40	<b>0.0</b>
1.5	31.2	28.0	<b>10.3</b>	20.5	20.0	<b>2.4</b>	25.5	22.9	<b>10.2</b>	28.8	24.6	<b>14.6</b>
2.0	17.5	12.3	<b>29.7</b>	10.5	9.8	<b>6.7</b>	14.7	10.9	<b>25.8</b>	12.6	9.6	<b>23.8</b>
2.5	9.0	6.0	<b>33.3</b>	7.7	6.3	<b>18.2</b>	7.7	5.3	<b>31.1</b>	9.8	7.4	<b>24.5</b>

의 전송률을 갖고 복호 수행에 약 600mW의 전력을 소모하는 완전 병렬화한 LDPC 복호기를 설계하였다<sup>9)</sup>. 전력 소모는 데이터 스위칭의 활성도에 비례하였고, 낮은 dB에서는 정상적으로 복호할 수 없으므로, 최대 반복 복호 횟수만큼 매번 반복 복호를 시도하여 데이터 스위칭의 활성도가 높아진다. 채널 환경이 양호한 Eb/No가 높은 dB에서는 최대 반복 복호 횟수보다 적은 반복 횟수로도 복호가 가능하므로 데이터 스위칭의 활성도가 낮아진다. 표 3은 제안된 알고리즘의 전력 소모와 면적 비교이다. 2.5dB에서 최대 약 33%의 반복 복호 횟수를 감소시켰으며, 복호기의 전력 소모 600mW 에서 약 200mW의 전력을 감소시킬 수 있다. 면적은 전체 시스템 혹은 복호기의 면적 대비 약 1% 가량 증가된다.

표 3. 제안된 알고리즘 적용 시 전력 소모와 면적 비교

	전력 소모 (mW)	면적 (게이트 수)
알고리즘 적용전	600	1,750K
알고리즘 적용후	400	1,765K

### V. 결론

4G 이동통신 시스템과 최근의 디지털 이동통신 환경에서 요구하는  $10^{-6} \sim 10^{-9}$  수준의 고품질의 신뢰도를 요구하는 고속 데이터 전송에 있어서 상당히 낮은 오류 확률을 보이는 LDPC 부호가 주목을 받고 있다. LDPC 부호의 특성상 블록 크기가 크고 반복 복호 횟수가 많을수록 오류 정정 성능이 높아진다. 특히 반복 복호 횟수가 많아지면 복호지연이 늘어나고 전송률이 떨어지며 전력소모가 많아지게 되므로 휴대 전화와 같은 휴대용 기기에 적합하도록 저 전력 복호기의 설계가 요구된다.

본 논문에서는 LDPC 복호기의 LLR가 적용된 연판정 반복 복호 과정에서 복호된 결과를 분석하고 복호값의 변화 방향을 예측하여 복호값의 sign을 반전시키는 후보정을 함으로써 반복 복호 횟수를 줄일 수 있는 알고리즘을 제안하였다. 동일한 채널 환경에서 BER 성능이 같은 경우 줄어든 복호 횟수 만큼 하드웨어를 사용하지 않으므로 전력 소모를 감소시킬 수 있다. 제안한 알고리즘을 적용한 경우 BER 성능의 감소 없이 프레임 당 반복 복호 횟수를 최대 약 33% 줄일 수 있음을 확인하였다. 클럭 주

파수 64MHz에서 1Gb/s의 전송률과 복호에 600mW의 전력을 소모하는 완전 병렬화한 LDPC 복호기에 제안된 알고리즘을 적용할 경우, 약 200mW의 전력 소모를 감소시킬 수 있으며 추가되는 게이트에 의해 면적은 약 1% 정도 증가하였다. 알고리즘의 특성상 높은 전송률을 유지하면서 전력 소모를 줄일 수 있으며, 만약 전력 소모를 그대로 유지한다면 전송률을 증가시킬 수 있다. 추가 연구 과제로는 제안된 알고리즘에 필요한 하드웨어를 최소화 하는 연구와 다양한 블록크기에 따른 반복 복호 횟수 성능에 대한 평가가 필요하다.

### 감사의 글

본 논문은 2007년도 「서울시 산학연 협력 사업」의 「나노 IP/SoC 설계 기술 혁신 사업단」의 지원으로 이루어졌으며 Simulation 및 synthesis에 IDEC에서 지원받은 CAD tool을 사용하였음.

### 참 고 문 헌

- [1] 황승구, “4세대 이동통신 무선 전송 기술 동향,” 주간기술동향, 통권 1168호, 2004년 10월.
- [2] T. Richardson and R. Urbanke, “The Renaissance of Gallager’s Low-Density Parity-Check Codes,” IEEE Communication Magazine, vol. 41, no. 8, pp. 126-131, Aug. 2003.
- [3] T. Richardson and R. Urbanke, “The Capacity of Low-density Parity-check Codes under Message-passing,” IEEE Trans. Inform. Theory, vol. 47, no. 2, pp. 599-618, Feb. 2001.
- [4] D. MacKay and R. Neal, “Near Shannon Limit Performance of Low Density Parity Check Codes,” IEE Electronics Letters, vol. 32, no. 18, pp. 1645-1646, Aug. 1996.
- [5] R. Gallager, “Low-Density Parity-Check Codes,” IRE Trans. Inform. Theory, vol. IT-8, pp. 21-28, Jan. 1962.
- [6] D. Mackay, “Good Error-correcting Codes Based on Very Sparse Matrices,” IEEE Trans. Inform. Theory, vol. 45, no. 2, pp. 399-431, Mar. 1999.
- [7] R. Tanner, “A Recursive Approach to Low Complexity Codes,” IEEE Trans. Inform.



Theory, vol. 27, no. 5, pp. 533-547, Sep. 1981.

[8] V. Zyablov and M. Pinsker, "Estimation of the Error Correction Complexity of Gallager Low Density Parity Codes," Problems of Information Transmission, vol. 11, no. 1, pp. 23-26, Jan. 1976.

[9] A. Blankby and C. Howland, "A 690-mW 1-Gb/s 1024-b, Rate-1/2 Low-Density Parity-Check Code Decoder," IEEE J. Solid-State Circuits, vol. 37, no. 3, pp. 404-412, Mar. 2002.

[10] V. Sorokine, F. Kschischang, and S. Pasupathy, "Gallager Codes for CDMA Applications - Part II: Implementations, Complexity, and System Capacity," IEEE Trans. Communications, vol. 48, no. 11, pp. 1818-1828, Oct. 2000.

[11] S. Hong and W. Stark, "Design and Implementation of a Low Complexity VLSI Turbo-code Decoder Architecture for Low Energy Mobile Wireless Communications," J. VLSI Signal Processing System, vol. 24, no. 1, pp. 43-57, Feb. 2000.

[12] N. Wiberg, H. Loeliger, and R. Kotter, "Codes and Iterative Decoding on General Graphs," in Proc. Information Theory, pp. 468, Sep. 1995.

[13] R. Morelos-Zaragoza, The Art of Error Correcting Coding. Wiley: New York, 2002.

[14] J. Fan, Constrained Coding and Soft Iterative Decoding. Kluwer Academic Pub.: Norwell, MA, 2001.

[15] F. Kshischang and B. Frey, "Iterative Decoding of Compound Codes by Probability Propagation in Graphical Models," IEEE Journal on Selected Area in Communications, vol. 16, no. 2, pp. 219-230, Feb. 1998.

[16] B. Sklar, Digital Communications: Fundamental and Applications. 2nd Edition, Prentice Hall Inc.,: New Jersey, 2001.

**이 준 호 (Jun-Ho Lee)**

준회원



2005년 2월 서강대학교 전자공학과 졸업  
 2007년 2월 서강대학교 전자공학과 석사학위 취득  
 2007년 3월~현재 LG전자 NC 사업본부 재직중  
 <관심분야> Turbo code, LDPC 부호

**박 창 수 (Chang-Soo Park)**

준회원



2002년 2월 서강대학교 전자공학과 졸업  
 2004년 8월 서강대학교 전자공학과 석사학위 취득  
 2005년 3월~현재 서강대학교 전자공학과 대학원 CAD & Embedded Systems 연구실 박사과정  
 <관심분야> SoC 설계, Embedded System Design

**황 선 영 (Sun-Young Hwang)**

정회원



1976년 2월 서울 대학교 전자공학과 졸업  
 1978년 2월 한국 과학원 전기 및 전자공학과 공학석사 취득  
 1986년 10월 미국 Stanford 대학 전자공학 박사학위 취득  
 1976~1981 삼성 반도체

주식회사 연구원, 팀장

1986년~1989년 Stanford 대학 Center for Integrated System 연구소 책임 연구원

Fairchild Semiconductor Palo Alto

Reaserch Center 기술 자문

1989년~1992년 삼성전자(주) 반도체 기술 자문

1989년 3월~현재 서강대학교 전자공학과 교수

<관심분야> SoC 설계 및 framework 구성, CAD 시스템, Computer Architecture 및 Embedded System Design 등