

멀티미디어 기반 해상통신을 위한 DVB-S2 기반 고속 LDPC 복호를 위한 알고리즘에 관한 연구

정 지원*, 권해찬*, 김영주**, 박상혁***, 이성로°

A Study on High Speed LDPC Decoder Algorithm Based on DVB-S2 Standard

Ji Won Jung*, Hae Chan Kwon*, Yeong Ju Kim**, Sang Hyuk Park***, Seong Ro Lee°

요약

본 논문에서는 멀티미디어 기반의 해상통신을 위한 DVB-S2 기반 고속 LDPC 복호를 위한 알고리즘을 제안하였다. 체크 노드 연산중에 비트 노드 연산을 수행하여 기존의 LDPC 복호 알고리즘에 비해 반복횟수를 줄일 수 있는 horizontal shuffle scheduling 알고리즘을 기반으로 하여 복호 속도를 보다 고속화 할 수 있는 알고리즘을 제안하였다. 기존의 체크 노드 연산은 하나의 메모리에서 값을 가져오기 때문에 체크 노드 연산과정에서 많은 지연이 발생하는데 이를 dc개의 병렬구조로 설계함으로써 체크 노드 연산과정의 지연을 줄일 수 있고 따라서 고속 복호가 가능하다. 이를 시뮬레이션 한 결과, 최대 반복 30회를 수행하였을 때 HSS 알고리즘은 326 Mbit/s, 제안한 알고리즘은 2.29 Gbit/s로 약 7배 이상의 복호 throughput을 얻을 수 있었다.

Key Words : DVB-S2, Horizontal Shuffle Scheduling(HSS), Sj memory, edge memory, Check Node Update(CNU)

ABSTRACT

In this paper, we proposed high speed LDPC decoding algorithm based on DVB-S2 standard for applying marine communications in order to multimedia transmission. For implementing the high speed LDPC decoder, HSS algorithm which reduce the iteration numbers without performance degradation is applied. In HSS algorithm, check node update units are update at the same time of bit node update. HSS can be accelerated to the decoding speed because it does not need to separate calculation of the bit nodes, However, check node calculation blocks need many clocks because of just one memory is used. Therefore, this paper proposed partial memory structure in order to reduced the delay and high speed decoder is possible. The results of the simulation, when the max number of iteration set to 30 times, decoding throughput of HSS algorithm is 326 Mbit/s and decoding speed of proposed algorithm is 2.29 Gbit/s. So, decoding speed of proposed algorithm more than 7 times could be obtained compared to the HSS algorithm.

※ 이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 대학중점연구소 지원사업으로 수행된 연구임(2009-0093828)

※ 본 연구는 교육과학기술부와 한국연구재단의 지역혁신 인력양성사업으로 수행된 연구결과임

◆ 주저자 : 한국해양대학교 전파공학과, jwjung@hhu.ac.kr, 정희원

° 교신저자 : 목포대학교 정보전자공학과, srlee@mokpo.ac.kr, 정희원

* 한국해양대학교 전파공학과, khanm100@naver.com

** 목포대학교 컴퓨터공학과, xfile7@mokpo.ac.kr

*** 목포대학교 정보통신공학과, parkshyuk@mokpo.ac.kr

논문번호 : KICS2013-01-031, 접수일자 : 2013년 1월 14일, 최종논문접수일자 : 2013년 3월 8일

I. 서 론

해상 통신은 최근 멀티미디어 기반을 둔 고속 통신이 요구되는 실정이다^[1]. 위성을 이용한 해상통신은 DVB-S2 기반을 둔 전송방식의 표준안에 근거를 두어야한다. 본 논문에서는 멀티미디어 기반을 둔 해상 통신에 적용키 위해 DVB-S2 전송 방식 중 LDPC(Low Density Parity Check) 복호에 비중을 두어 연구를 한다. LDPC는 차세대 통신시스템의 오류정정 방식으로 주목받고 있으며, 오류율이 거의 Shannon limit에 근접하는 오류정정 부호 기술로 현재 위성의 DVB-S2, Wibro 기반의 802.16e 등에서 채택되는 무선통신에서 각광받는 강력한 오류정정부호기술이다^[2].

LDPC 부호는 최신 위성 방송 및 무선 통신에서 적용되는 오류정정 알고리즘으로 채널 부호화 방식을 주로 적용하고 있다. 광대역 위성 방송에 적용될 수 있는 오류 정정 부호는 고속 데이터 전송에 효율적이고 성능이 우수한 복호기의 적용이 필수 불가결하며, 이는 DVB-S2에서 제시된 LDPC 부호화 방식은 초고화질 다채널 실감 방송 서비스 및 멀티미디어 기반을 둔 고속 통신에 적합하다. 유럽식 위성 방송 표준안인 DVB-S2에 적용되는 사논의 채널 용량 한계에 근접한 LDPC 부호는 터보 부호에 비해 복호화의 복잡도가 낮을 뿐 아니라 좋은 거리 특성으로 오류마루 현상이 나타나지 않고, 완전 병렬 처리로 고속 처리가 가능한 장점이 있다^[3,4]. 하지만 LDPC 부호는 큰 블록 사이즈(N=64800) 및 많은 반복 횟수를 요구하고 있다. 또한 고속화를 위한 구현상 문제점의 대부분은 복호부에 있으며, 이를 어떻게 100 Mbps급 이상의 전송률을 가지게 하는가에 있다. 따라서 복호 시 병렬 처리, 병렬 처리에 따른 메모리 설계가 중요하다.

이에 본 논문에서는 비트 노드 계산을 체크 노드 계산 후에 하지 않고 체크 노드 계산 중에 수행하는 HSS(Horizontal Shuffle Scheduling) 방식^[5]을 기반으로 하는 복호기 구조에서 기존의 방식 보다 고속화 할 수 있는 알고리즘을 연구 하였으며, 기존의 row weight 개수인 dc개의 직렬 구조에서 dc개의 병렬 구조로 메모리를 효율적으로 설계함으로써 고속화가 가능하게끔 하였다.

본 논문의 2장에서는 DVB-S2 기반 LDPC 부호의 복호 알고리즘에 대해 알아보고 3장에서는 HSS 복호 알고리즘에 대해 알아보겠다. 그리고 4장에서는 3장에서 살펴본 HSS 복호 알고리즘을 기반으로

고속 복호를 위해 제안한 알고리즘에 대해 알아보고 5장에서 결론을 맺는다.

II. DVB-S2 기반 LDPC 복호 알고리즘

LDPC 복호과정은 전송된 심벌을 비트 노드와 체크 노드에서 각각의 확률값을 구하여 반복을 통해 전송된 비트를 결정하는 것이다. 전체적인 복호 과정은 그림 1과 같다.

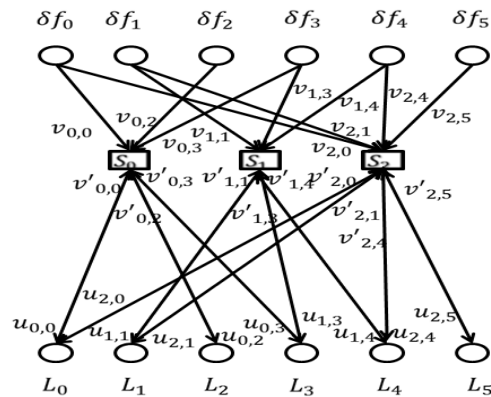


그림 1. LDPC 전체 복호 과정
Fig. 1. Total Decoding Process of LDPC

DVB-S2 기반 LDPC 복호 알고리즘은 수신비트에 채널 추정 값을 구하는 초기화 과정, 체크 노드 확률을 구하는 CNU(Check Node Update) 과정, 비트 노드 확률을 구하는 BNU(Bit Node Update) 과정으로 총 세 가지 단계로 나눌 수 있다.

단계 1 : 초기화 과정

첫 번째 단계는 초기화 단계로, 식 (1)과 같이 수신데이터에 채널 추정 값인 $-\frac{2}{\sigma^2}$ 를 곱해준 δf_n 을 각 비트 노드에 저장한다. σ^2 은 잡음 분산을 나타낸다.

$$\delta f_n = -\frac{2}{\sigma^2} r_n \quad (1)$$

단계 2 : CNU 과정

하나의 체크 노드에서 dc개의 row weight를 가진다고 가정할 때, 각각의 체크 노드로 들어오는 옛 지 값은 식 (2)와 식 (3)으로 구할 수 있다.

$$g(a,b) = \text{sign}(a) \times \text{sign}(b) \times \min(|a|, |b|) - LUT_g(a,b) \quad (2)$$

여기서,

$$LUT_g(a,b) = \ln(1 + e^{-|a+b|}) - \ln(1 + e^{-|a-b|}) \quad (3)$$

를 나타낸다.

따라서 $LUT(x) = \ln(1 + e^{-x})$ 의 look up table로 위의 CNU를 계산할 수 있다. 각각의 비트 노드들은 각각 연결된 모든 엣지의 값을 더함으로써 복호 값인 LLR(Log Likelihood Ratio) 값을 구할 수 있다. 이는 식 (2.4)로 나타낼 수 있다.

$$L_n = \sum u_{m,n} \quad (4)$$

$u_{m,n}$ 은 그림 1에서 n번째 비트노드에서 m번째 체크노드로 연결되는 식 (2)의 엣지 값을 나타낸다.

단계 3 : BNU 과정

CNU에서 각 체크 노드로 연결된 비트들에 대한 업데이트가 이루어지고 각 비트 노드의 LLR 값을 구한 후, 마지막 단계로 LLR 값과 비트 노드로 입력된 각 엣지 값들을 이용하여 다시 체크 노드로 보내는 엣지의 값을 구하기 위한 BNU 연산을 해야 한다. BNU 연산 역시 CNU와 마찬가지로 CNU 연산 후, 비트 노드로 입력된 자신의 엣지 값을 제외한 다른 엣지 값으로만 연산을 행한다. 이러한 연산 방법은 식 (2.5)와 같이 간단히 나타낼 수 있다.

$$v'_{m,n} = L_n - u_{m,n} \quad (5)$$

$v'_{m,n}$ 은 그림 1에서 m번째 체크노드에서 n번째 비트노드로 연결되는 엣지 값을 나타낸다. 위의 세 단계 중 처음 초기화 단계 후에는 CNU 과정과 BNU 과정만을 반복하며, 복호 데이터를 구하게 된다.

III. HSS 복호 알고리즘

기존의 LDPC 복호기의 복호 순서는 우선 수신 데이터를 이용하여 비트 노드를 초기화 한 후 각각의 체크 노드에 연결된 비트 노드 값을 이용하여 체크 노드 업데이트를 한다. 체크 노드 업데이트 후 다시 각각의 비트 노드에 대해 업데이트를 하고, 이러한 연산을 계속 반복한다. 기존의 복호 방식에 의해 체크 노드 업데이트 연산이 모두 끝난 후 비트 노드 업데이트를 하기 때문에 한번의 반복에도 많은 지연이 발생하여 고속의 LDPC 복호를 할 수 없다.

이를 극복하기 위해 HSS 복호 방법을 연구하였다. HSS 방식은 기존의 방식과는 달리 체크 노드 업데이트 연산을 하면서 비트 노드 업데이트 연산을 동시에 하는 것이 가능하다. HSS 복호 방식의 흐름도는 그림 2와 같다.

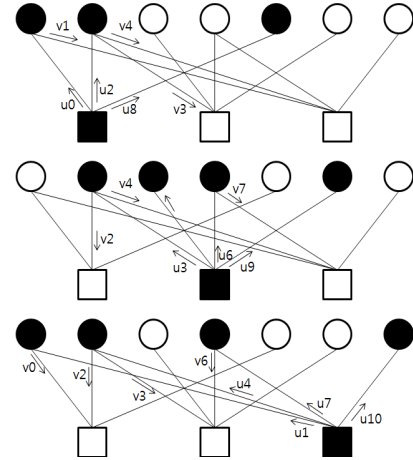


그림 2. HSS 복호의 흐름도
Fig. 2. HSS Decoding Flow

HSS 방식을 이용한 LDPC 복호 과정 중 각 비트 노드의 값은 다음 식에 의해 구할 수 있다.

$$S_i = LLR_i + \sum_{i=1}^{dv} u_i \quad (6)$$

여기서 S_i 는 비트 노드의 최종 값을 나타내고, LLR 은 수신 데이터를 나타낸다. i 는 비트 노드이고, dv 는 비트 노드에 연결된 edge 수이다. 그리고 u_i 는 체크 노드 업데이트를 통해 얻어진 각 edge 값이다. 체크 노드 업데이트는 식 (7)와 같다.

$$u_j = \bigoplus_{k=1, k \neq j}^{d_c} v_k \quad (7)$$

v 는 비트 노드에서 체크 노드로 향하는 엣지를 나타내고, d_c 는 체크 노드에 연결된 엣지 수이다. 위 식에서 \oplus 는 다음과 같이 구할 수 있다.

$$|v_i| \oplus |v_j| = \min(|v_i|, |v_j|) - offset \quad (8)$$

$$sign(|v_i| \oplus |v_j|) = sign(v_i) \times sign(v_j) \quad (9)$$

체크 노드 업데이트의 값을 이용하여, 각 비트

노드의 값은 식 (10)에 의해 구해진다.

$$v_i = S_i - v'_i \quad (10)$$

v' 는 이전 반복에서의 엣지 값을 나타낸다. 이와 같이 한 번의 반복이 끝나게 된다. 이러한 과정이 반복되면서 모든 반복이 끝나거나, 신뢰성 있는 데이터가 나올 때, 복호 과정은 끝나게 된다.

HSS 방식은 한 번의 반복에서 비트 노드가 row weight만큼 업데이트 되기 때문에 기존의 방식에 비해 좋은 성능을 가진다. 이는 기존의 알고리즘 보다 요구되는 반복횟수가 많이 줄어들음을 의미한다. 그림 3은 부호화율 1/2에서 기존 LDPC 복호 방식과 HSS 복호방식의 반복횟수에 따른 성능을 나타낸다.

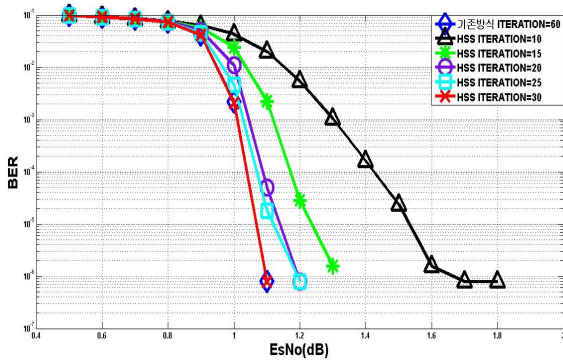


그림 3. 기존방식과 HSS방식의 반복횟수에 따른 성능비교 (r=1/2)
Fig. 3. Performance comparison by iteration number of HSS

성능비교에 사용된 데이터는 약 100만개이고, AWGN 환경에서 시뮬레이션을 하였다. 시뮬레이션의 한 블록의 크기는 64800이다. 기존의 방식은 반복 횟수를 고정 하였고, HSS 방식은 반복 횟수를 조정하면서 성능을 비교하였다. 성능 비교 결과 기존의 방식과 HSS 방식의 반복 횟수 30회에서 거의 비슷한 성능을 보임을 알 수 있다. 이는 HSS 방식을 사용할 때, 한 번의 반복에서의 속도 증가 효과가 있음과 동시에 반복 횟수 역시 줄일 수 있으며, 기존의 방식에 비해 성능 열화도 거의 발생하지 않음을 의미한다.

다음 그림 4는 HSS 방식을 적용한 LDPC 복호 구조를 나타낸다.

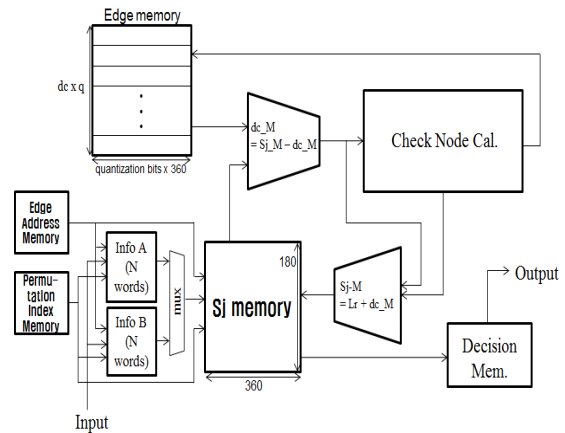


그림 4. HSS 방식을 적용한 LDPC 복호기 구조
Fig. 4. Structure of LDPC decoder based on HSS

HSS 방식을 적용한 복호기의 구조를 보면 크게 메모리 부분과 CNU 부분으로 나눌 수 있다. 특히 수신 데이터를 저장하는 메모리와 Sj메모리에서 데이터를 읽어 올 때는 H 매트릭스에 따라 랜덤하게 메모리 액세스가 되어야 하기 때문에 이를 구현할 수 있는 여러 인덱스들이 필요하다.

HSS 방식을 적용한 LDPC 복호기는 BNU 계산을 위한 블록을 따로 만들지 않는다. 그 이유는 CNU 블록에서 나온 출력 값들을 바로 Sj 메모리에 업데이트 시키기 때문에 BNU를 위한 연산만을 따로 하지 않기 때문이다. 즉, LDPC 복호를 위한 연산 블록으로는 CNU 만을 필요로 한다.

IV. HSS 기반 고속 복호 알고리즘

4.1. 고속 복호 알고리즘

HSS를 적용한 LDPC 복호 알고리즘에서는 체크 노드 연산과정에서 edge메모리와 Sj메모리를 각각 하나씩 사용하였기 때문에 체크 노드 연산 과정에서 dc개의 데이터를 읽어오고 체크 노드 연산 후 다시 edge메모리와 Sj메모리에 저장하는데 지연이 발생하게 된다. 이러한 문제를 해결하고 고속 복호를 위해 체크 노드 연산에 필요한 dc개의 데이터를 한 번에 읽어와 연산하고 한 번에 저장할 수 있는 방안인 edge 메모리와 Sj메모리를 각각 dc개로 분리하는 알고리즘을 제안하였다. 그림 5는 edge메모리와 Sj메모리를 각각 dc개로 분리한 알고리즘을 나타낸다.

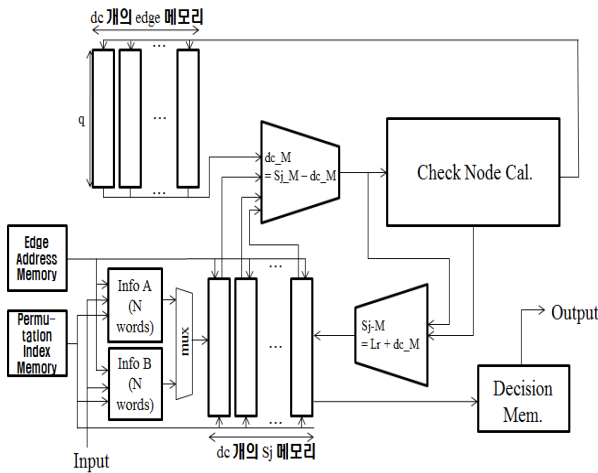


그림 5. edge메모리와 Sj메모리를 dc개로 분리한 복호 알고리즘
Fig. 5. Decoding algorithm of dc split edge memory and Sj memory

edge메모리와 Sj메모리를 각각 dc개로 분리하게 되면 체크 노드 연산과정에서 동시에 dc개의 데이터를 연산할 수 있게 된다. 이렇게 되면 한 번의 클럭으로 dc개의 데이터를 동시에 처리 가능하기 때문에 고속 복호가 가능하게 된다. 동시에 dc개의 데이터를 읽고 저장할 수 있게 되면 체크 노드 연산을 병렬로 처리할 수 있기 때문에 이 과정에서도 소요되는 클럭을 줄일 수 있다. 이때에 필요한 클럭 수는 $2^q \geq dc$ 를 만족하는 가장 작은 q로 나타낼 수 있다. 이에 따라 부호화율이 1/2인 경우 dc가 7이기 때문에 3 클럭의 연산으로 체크 노드 연산을 할 수 있게 된다.

다음 그림은 부호화율이 1/2인 경우 dc가 7일 때 체크 노드 연산의 병렬 구조를 나타낸다.

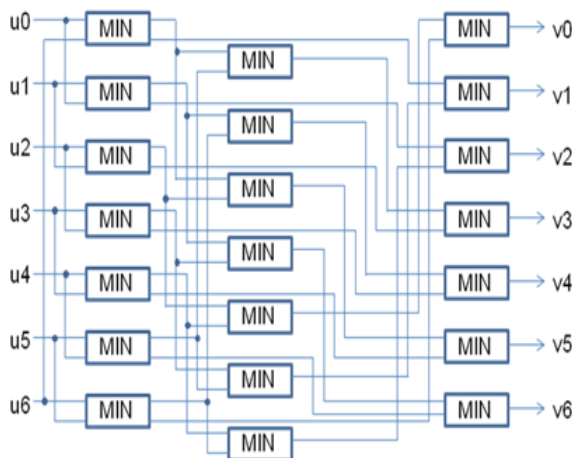


그림 6. 체크 노드 연산의 병렬 구조
Fig. 6. Parallel structure of check node calculation

그림 6에서 보듯이 dc개의 체크 노드 연산에 3 클럭이 소요됨을 알 수 있다.

다음은 HSS 알고리즘과 제안한 알고리즘을 적용시켰을 때의 성능을 비교한 그림이다. 부호화율은 1/2이고 이때 dc의 개수는 7이다.

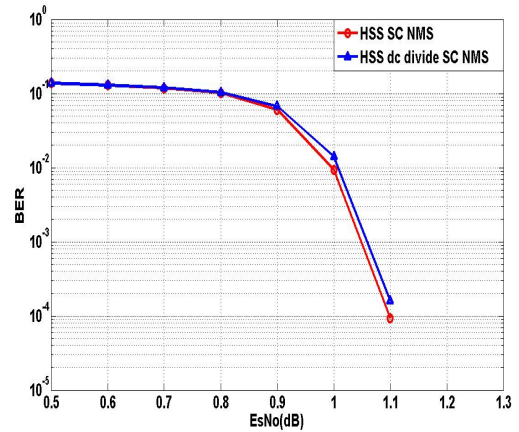


그림 7. HSS 알고리즘과 제안한 알고리즘의 성능 비교
Fig. 7. Performance Comparison by HSS and proposed algorithm

edge메모리와 Sj메모리를 dc개로 나누는 알고리즘과 기존의 HSS 알고리즘의 성능을 비교한 결과 거의 비슷한 성능을 보임을 알 수 있다. 이는 제안한 알고리즘으로 성능의 열화없이 고속 LDPC 복호를 구현할 수 있다는 것을 나타낸다.

4.2. 제안한 알고리즘의 속도 비교

표 1은 부호화율 1/2에서 HSS 알고리즘과 제안한 알고리즘의 복호 속도를 비교한 표이다.

표 1. HSS 알고리즘과 제안한 알고리즘의 복호 속도
Table 1. Decoding speed by HSS and proposed algorithm.

	required clock (1 iteration)	maximum required clock (30 iterations)	throughput(10ns clock period)
HSS algorithm	662 clock	19860 clocks	326 Mbit/s
Proposed algorithm	94 clock	2820 clocks	2.29 Gbit/s

부호화율1/2에서 HSS 알고리즘은 1회 반복에 662 클럭이 필요하며 전체 반복횟수를 기존 LDPC에 비해 절반으로 줄일 수 있기 때문에 최대 반복

시 요구되는 클럭은 19860 클럭이 소요된다. 이에 비해 제안한 알고리즘에서는 1회 반복에 94 클럭이 필요하고 최대 반복 시 2820 클럭이 소요된다. 이에 따라 10ns에서 throughput을 계산하였을 때 HSS 알고리즘은 326 Mbit/s, 제안한 알고리즘에서는 2.29 Gbit/s로 HSS 알고리즘에 비해 약 7배의 throughput을 얻었다.

edge 메모리와 Sj 메모리를 dc개로 나눈 제안한 알고리즘은 1회 반복에서 상대적으로 HSS 알고리즘 보다 적은 클럭이 필요하다. 그 이유는 체크 노드 연산과정에서 dc개 만큼 분리되어 한번에 데이터를 읽고 저장하는 것이 가능하고 또한 체크 노드 연산을 병렬로 처리 가능하기 때문에 HSS 알고리즘보다 적은 클럭이 필요하게 되고 고속 복호가 가능하다.

V. 결 론

위성을 이용한 해상통신은 DVB-S2 기반을 둔 전송방식의 표준안에 근거를 두어야 하며, 따라서 본 논문에서는 멀티미디어 기반을 둔 해상 통신에 적용키 위해 DVB-S2 전송 방식 중 LDPC 복호에 비중을 두어 연구를 하였으며, 복호기 구조에서 HSS 방식을 기반으로 보다 복호과정을 고속화할 수 있는 알고리즘을 제안하였다. HSS 알고리즘은 비트 노드 연산을 체크 노드 연산 후에 수행하지 않고 체크 노드 연산 중에 수행하기 때문에 기존의 LDPC 복호 방식에 비해 반복횟수를 줄일 수 있고 고속 복호가 가능하다. 이에 HSS 알고리즘을 기반으로 복호 속도를 더욱 높일 수 있는 알고리즘으로 기존의 row weight 개수수인 dc개의 직렬 구조에서 dc개의 병렬 구조로 비트 노드 메모리와 edge메모리를 나누어 체크 노드 연산을 고속으로 할 수 있는 알고리즘을 제안하였다. edge메모리와 비트 노드메모리를 dc개로 나눈 제안한 알고리즘은 1회 반복에서 상대적으로 HSS 알고리즘 보다 적은 클럭이 필요하다. 그 이유는 체크 노드 연산과정에서 dc개 만큼 분리되어 한번에 데이터를 읽고 저장하는 것이 가능하고 또한 체크 노드 연산을 병렬로 처리 가능하기 때문에 HSS 알고리즘보다 적은 클럭이 필요하게 되어 고속 복호가 가능하다.

dc개의 직렬구조에서 dc개의 병렬 구조로 메모리 구조를 변경함으로써 복호 속도를 개선할 수 있으나 이는 메모리의 낭비를 초래하며, 메모리 효율성 면에서는 다소 떨어진다 할 수 있다. 하지만 메모리 효율성의 측면을 감안하더라도 고속화를 극대화 할

수 있는 알고리즘으로 기존의 방식보다 약 7배 이상의 고속 복호가 가능하리라 사료된다.

References

- [1] D. W. Jang and H. D. Choi, "A study on trans of a technology and standardization for modernizing marine mobile radio communications", *Korea Inst. Commun. Inform. Sci. Winter Conf. (KICS)*, pp. 53-54, Gwangwon-do, Korea, Feb. 2012.
- [2] R. G. Gallager, "Low-density parity-check codes," *IRE trans. inform. theory*, vol. 8, no. 1, pp. 21-28, Jan. 1962.
- [3] "Digital Video Broadcasting(DVB); Second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications (DVB-S2)." *European Telecommun. Standard Inst. (ETSI) En 302 307 V1.2.1*, Apr. 2009.
- [4] D. J. C. Mackay and R. M. Neal, "Near Shannon limit performance of low-density parity-check codes," *Electron. Lett.*, vol. 32, no. 18, pp. 1645-1646, Aug. 1996.
- [5] A. Segard, F. Verdier, D. Declercq, P. Urard. "DVB-S2 compliant LDPC decoder integrating Horizontal Shuffle Scheduling," *Int. Symp. Intelligent Signal Process. Commun. (ISPACS 2006)*, pp. 1013-1016, Dec. 2006.
- [6] J. Dielissen, A. Hekstra, and V. Berg. "Low cost LDPC decoder for DVB-S2," in *Proc. 2006 Design, Automation Test in Europe (DATE'06)*, vol. 2, pp. 1-6, Munich, Germany, Mar. 2006.
- [7] V. Savin. "Self-corrected min-sum decoding of LDPC codes," in *Proc. IEEE Int. Symp. Inform. Theory (IEEE ISIT)*, pp. 146-150, July 2008.
- [8] S. M. Kim, C. S. Park, and S. Y. Hwang, "A Novel Partial Parallel Architecture for High-throughput LDPC Decoder for DVB-S2," *IEEE Trans. Consum. Electron.*, vol. 56, no. 2, pp. 820-825, May 2010
- [9] X.-Y. Hu, E. Eleftheriou, D.-M. Arnold, and

A. Dholakia, "Efficient implementations of the sum-product algorithm for decoding LDPC codes." in *Proc. IEEE Globecom*, pp. 1036-1036E, San Antonio, TX, Nov. 2001.

정 지원 (Ji Won Jung)



1989년 2월 성균관대학교 전자공학과(공학사)
 1991년 2월 성균관대학교 전자공학과(공학석사)
 1995년 2월 성균관대학교 정보공학과(공학박사)
 1991년 1월~1992년 2월 LG

정보통신연구소 연구원
 1995년 9월~1996년 8월 한국통신 위성통신연구실 선임연구원
 1997년 3월~1998년 12월 한국전자통신연구원 초빙 연구원
 1996년 9월~현재 한국해양대학교 전파공학과 정교수
 2001년 8월~2002년 8월 캐나다 NSERC Fellowship (Communication Research Center 근무)
 <관심분야> 위성통신, 이동통신, 변.복조기술, 채널코딩, FPGA 기술 등

권 해 찬 (Hae Chan Kwon)



2013년 2월 한국해양대학교 전파공학과 석사과정
 <관심분야> 위성통신, 이동통신, 변·복조기술, 채널코딩, FPGA 기술 등

김 영 주 (Yeong Ju Kim)



2005년 2월 조선대학교 전산통계학과 졸업
 2008년 2월 목포대학교 정보컴퓨터교육학 석사 졸업
 2010년 3월~현재 목포대학교 컴퓨터공학과 박사과정
 <관심분야> 데이터베이스, 데이터마이닝, RFID, USN, 텔레매틱스, 임베디드 시스템

박 상 혁 (Sang Hyuk Park)



2012년 2월 목포대학교 정보통신공학과 졸업
 2012년 3월~현재 목포대학교 정보통신공학과 석사과정
 <관심분야> 해상무선통신, e-Navigation, Cognitive Radio, 4G 이동통신

이 성 로 (Seong Ro Lee)



1987년 2월 고려대학교 전자공학과 공학사
 1990년 2월 한국과학기술원 전기및전자공학과 공학석사
 1996년 8월 한국과학기술원 전기및전자공학과 공학박사
 1997년 9월~현재 목포대학교 공과대학 정보전자공학과 교수
 <관심분야> 디지털통신시스템, 이동 및 위성통신시스템, USN/텔레매틱스응용분야, 임베디드시스템