

논문 2006-43SD-9-5

연산기와 메모리 재사용을 이용한 효율적인 DVB-S2 규격의 LDPC 복호기 구조

(Architecture of an LDPC Decoder for DVB-S2 using reuse Technique
of processing units and Memory Relocation)

박 재 근*, 이 찬 호**

(Jaegeun Park and Chanho Lee)

요 약

Low Density Parity Check (LDPC) code는 최근 그 우수한 성능으로 인하여 4세대 무선 이동 통신용 채널 코딩으로 주목받고 있다. 또한 유럽 디지털 위성 방송 규격인 DVB-S2는 LDPC 코드를 채널 코딩방식으로 채택하였다. 본 논문에서는 인코더와 디코더 양쪽 모두 효율적으로 하드웨어 구현이 가능한 hybrid H-matrix 구조를 이용한 DVB-S2 LDPC 복호기 구조를 제안한다. Hybrid H-matrix는 semi-random 방식과 partly parallel 방식을 결합하여 부호기와 복호기를 동시에 효율적으로 구현할 수 있다. 제안된 복호기 구조에서는 다양한 코드율에 사용되는 Variable Node processor Unit (VNU)을 재사용하기 위한 새로운 VNU와 최적화된 블록 메모리 배치 방법을 이용하였다. 제안된 구조를 이용하여 코드율 1/2의 DVB-S2 LDPC 복호기를 설계하였고 그 결과를 기존의 복호기와 비교하였다.

Abstract

Low-density parity-check (LDPC) codes are recently emerged due to its excellent performance. The standard for European high definition satellite digital video broadcast, DVB-S2 has adopted LDPC codes as a channel coding scheme. This paper proposes a DVB-S2 LDPC decoder architecture using a hybrid parity check matrix which is efficient in hardware implementation for both decoders and encoders. The hybrid H-matrices are constructed so that both the semi-random technique and the partly parallel structure can be applied to design encoders and decoders. Using the hybrid H-matrix scheme, the architecture of LDPC decoder for DVB-S2 can be very practical and efficient. In addition, we show a new Variable Node processor Unit (VNU) architecture to reuse the VNU for various code rates and optimized block memory placement to reuse. We design a DVB-S2 LDPC decoder of code rate 1/2 using the proposed architecture. We estimate the performance of the DVB-S2 LDPC decoder and compare it with other decoders.

Keywords : LDPC code, semi-random, Hybrid H-matrix, partly parallel structure, DVB-S2

I. 서 론

LDPC(Low Density Parity Check) code는 block code의 일종으로 1962년에 Gallager에 의해 처음 제안

되었다^[1]. 그러나 그 당시의 기술력으로는 구현이 불가능한 복호의 복잡도로 인해서 실용화 되지 못하고 한동안 잊혀 오고 있었다. 그러나 1995년에 Mackay와 Neal은 이를 재발견하였고, Gallager의 방식이 간단한 확률적 복호 법에 의해서 성능이 매우 우수함을 보였다^[2]. 최근에는 백색 잡음(additive white Gaussian noise: AWGN) 채널에서 Shannon의 채널 용량에 불과 0.0045dB 떨어진 임계 치를 갖는 LDPC 코드가 Chung 등에 의해 발견되었다^[3]. 이는 LDPC 코드가 재발견되기 이전에 가장 우수한 채널 코딩 기법이었던 터보 코

* 학생회원, 송실대학교 전자공학과
(Dept. of Electronic Engr., Soongsil University)

** 정회원, 송실대학교 정보통신전자공학부
(School of Electronic Engr., Soongsil University)

※ 본 연구는 송실대학교 교내연구비 지원으로 이루어졌음

접수일자: 2006년7월1일, 수정완료일: 2006년8월18일

드와 비교했을 때 더 작은 최소 거리(minimum distance)를 가지기 때문에 오류 마루(error floor) 현상이 거의 나타나지 않아 더 좋은 bit error rate(BER)를 가진다는 장점이 있다. 또한 합곱(sum-product) 알고리즘을 기반으로 한 반복 복호 과정을 완전히 병렬로 처리할 수 있기 때문에 복호 속도가 빠르다는 장점도 있다^[4]. 이러한 이유로 고속과 고성능을 요구하는 4세대 이동 통신용 채널 코딩으로 LDPC 코드가 주목받고 있다. 현재 유럽 방식 고화질 위성 방송용 규격인 DVB-S2 규격은 LDPC 코드와 BCH (Bose-Chaudhuri-Hocquenghem) 코드를 결합한 연접부호를 적용하고 있다^[5].

본 논문에서는 semi-random 방식과 partly parallel 방식을 결합하여 인코더와 디코더 모두 효율적인 구현이 가능한 hybrid H-matrix 구조를 이용한 DVB-S2 규격의 LDPC 디코더 구조를 제안한다. 기존의 hybrid H-matrix를 DVB-S2 LDPC 코드의 64,800 프레임을 갖는 구조에 맞게 변형을 하였다. 또한 디코더의 면적을 감소시키기 위하여 one-port 방식의 VNU(Variable Node processing Unit)도 이용할 수 있도록 하였고 동작 모드에 따라 연산기를 재사용하고 메모리 재배치를 통해 면적 감소시키는 구조를 택하였다. 이 구조를 바탕으로 코드율 1/2 동작 모드의 디코더를 구현하였다.

II. Hybrid H-matrix 구조

1. Semi-random 방식

Semi-random 방식을 이용한 H-matrix의 구조는 H_d 와 H_p 두 부분으로 분리된 형태로 구성된다^{[6][7]}. 여기서 H_d 는 임의의 행렬 형태를 가지며, H_p 는 deterministic한 형태로 이중 대각선(dual-diagonal) 정방행 행렬을 사용함으로써 인코더 부분의 복잡도를 줄이게 된다^[8]. 생성된 H_d -matrix를 사용한 H-matrix는 $H=[H_d, H_p]$ 의 구조를 가지며, 이때 코드워드는 $C=[d, p]^t$ (d :정보 비트, p :패리티 비트)의 형태로 만들어진다.

2. Partly parallel 방식

Partly parallel 방식은 fully parallel 방식이 가지는 하드웨어 크기가 크다는 문제점을 해결한 방식이며 Zhang 등이 처음 제안했다^[9]. 이 방식은 우선 디코더의 구조를 구성한 후에 shift된 항등 행렬을 이용하여 H-matrix를 만든다. 그러나 이러한 방식은 코드율의 변화에 민감하며 성능이 떨어지는 단점이 있다^[10]. 이것을

개선한 방식이 행렬 확장 방식으로 ($M_s \times N_s$) 크기의 기저 행렬과 ($p \times p$) 크기의 shift된 항등 행렬로 구성된다. 기저 행렬을 확장시킴으로써 설계의 유연성이 증가된다. 이러한 구조의 H-matrix를 사용하여 디코더를 구현할 경우 모든 노드가 프로세서를 가질 필요 없이 시분할 다중화 방식으로 공유할 수 있다^[9].

Partly parallel 방식을 사용하여 디코더를 구현할 경우, CNU(Check Node processor Unit)와 VNU(Variable Node processor Unit)의 개수는 각각 기저 행렬의 열과 행의 개수와 같은 M_s 개와 N_s 개만을 필요로 한다^[10]. H-matrix의 크기가 같을 경우, Fully parallel 방식의 디코더는 각각 pM_s 개의 CNU와 pN_s 개의 VNU를 필요로 하며 이것은 partly parallel 방식의 디코더보다 p 배 많은 수이다. Fully parallel 방식의 디코더가 2 cycle만에 한 번의 반복 연산이 가능한데 비하여 partly parallel 방식의 디코더는 $2p$ cycle을 필요로 한다. Partly parallel 방식은 디코더를 효율적으로 구현할 수 있도록 하지만 인코더 구조에 대해서는 여전히 복잡도가 크다는 문제점을 가지고 있다.

3. Hybrid H-matrix 구조

LDPC 코드의 인코더와 디코더의 하드웨어 구현에 대한 방법들을 알아보았으나 여전히 문제점이 남아있다. Semi-random 방식을 사용할 경우에는 인코더의 구현이 간단하지만 디코더의 구현이 굉장히 복잡하다. 반면 partly parallel 방식을 사용할 경우에는 디코더의 구현이 비교적 간단하지만 인코더의 구현에 문제가 있다.

이러한 하드웨어 구현의 문제점들을 해결하고자 hybrid H-matrix 구조의 LDPC 코드가 제안되었다^[6]. Hybrid H-matrix 구조는 앞에서 살펴본 semi-random 방식과 partly parallel 방식을 결합한 구조이다. 그림 1은 이러한 hybrid H-matrix의 구조를 보여주고 있다. Hybrid H-matrix는 $[H_d | H_p]$ 의 systematic한 구조를 가지는데, H_d 는 partly parallel 방식을 따르는 행렬이며, H_p 는 semi-random 방식을 따르는 이중 대각선 행렬을

$$\begin{array}{c}
 H^d \qquad \qquad \qquad H^p \\
 \left[\begin{array}{ccccc}
 T_{1,1} & O & O & T_{1,4} & O \\
 O & T_{2,2} & T_{2,3} & O & O \\
 O & O & O & O & T_{3,5}
 \end{array} \right]
 \left[\begin{array}{cccccc}
 1 & 0 & \dots & \dots & 0 & 0 \\
 1 & 1 & \dots & \dots & 0 & 0 \\
 \vdots & \vdots & \dots & \dots & \vdots & \vdots \\
 \vdots & \vdots & \dots & \dots & \vdots & \vdots \\
 0 & 0 & \dots & \dots & 1 & 0 \\
 0 & 0 & \dots & \dots & 1 & 1
 \end{array} \right]
 \end{array}$$

그림 1. Hybrid H-matrix의 구조
 Fig. 1. Structure of hybrid H-matrix.

사용하고 있다. 이러한 구조로 인하여 인코더에서는 semi-random 방식의 장점을, 디코더에서는 partly parallel 방식의 장점을 살릴 수 있다. 따라서 hybrid H-matrix 구조를 사용할 경우 인코더와 디코더 양쪽 모두 실제로 사용 가능할 정도의 하드웨어로 구현이 가능하다.

III. DVB-S2용 LDPC 복호기

차세대 DVB-S2 규격은 채널 코딩 방식으로 BCH 코드와 LDPC 코드를 이용하고 있다. DVB-S2의 LDPC 코드를 이용한 복호기는 hybrid H-matrix의 구조를 적용하여 구현이 가능하다. DVB-S2의 LDPC 코드의 블록 길이는 64,800 bits의 표준(noraml) 프레임과 16,200 bits의 짧은(short) 프레임을 사용할 수 있다. 방송을 위한 블록 길이는 64,800을 이용한다^[5].

1. DVB-S2 LDPC 행렬 구조와 hybrid H-matrix

DVB-S2 LDPC 행렬은 hybrid H-matrix와 매우 유사한 형태로 Hd와 Hp로 이루어져 있다. Hp는 semi-random 방식을 따르는 이중 대각선 행렬로 동일하지만 Hd가 다른 형태이다. DVB-S2의 LDPC 코드의 Hd 행렬은 코드율에 따라 제공되는 11개의 테이블을 이용하여 식 (1)에 따라 인코딩 하는 방식으로 구성된다^[5].

$$\{x + m \text{ mod } 360 * q\} \text{ mod } (n_{ldpc} - k_{ldpc}) \quad (1)$$

여기서 x는 인코딩할 때의 패리티 비트의 주소를 나타내고 m은 정보 비트의 순서이며, q는 코드율에 따라 값이 주어지는 상수이다. n_{ldpc} 는 전체 프레임 비트 수로서 표준 프레임일 경우에는 64,800을 나타내고 k_{ldpc} 는 그 중에 정보 비트 수를 나타내어 코드율이 1/2일

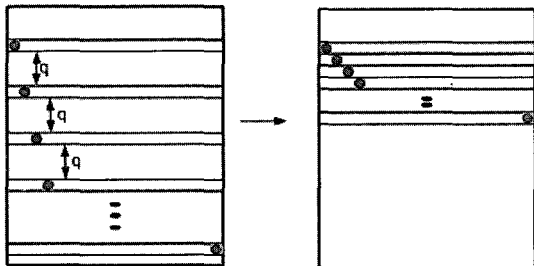


그림 2. Hd 기저행렬을 항등 행렬로 만들기 위해 열 위치를 변경한 모습

Fig. 2. Change of position of rows of Hd base matrix for making an identity matrix.

경우에는 32,400이다. (1) 식에서 보면 정보 비트가 증가할 때마다 패리티 비트의 주소가 q배씩 일정하게 증가한다. 그림 2처럼 패리티 비트 주소가 일정하게 증가하는 것을 열 위치를 변경함으로써 hybrid H-matrix의 확장 행렬중 하나의 항등 행렬로 구성할 수 있다.

코드율이 1/2일 경우, Hd의 기저 행렬(90 X 90)의 첫 열은 항등 행렬 (360 X 360) 8개와 O 행렬(360 X 360) 82개로 총 90개 행으로 구성할 수 있다. Hd의 기저행렬이 (90 X 90)이므로 hybrid H-matrix를 이용하게 되면 90개의 VNU가 필요하게 된다^[7].

2. 재사용가능한 VNU

DVB-S2 규격에는 11개의 코드율이 있어 각각의 코드율별로 Hd의 기저 행렬이 다르게 구성이 되며, 연산에 필요한 VNU와 CNU 개수가 다르게 된다. 연산에 필요한 VNU와 CNU 개수와 위치에 따라 필요한 VNU가 표 1에 나타나 있다.

DVB-S2의 Hd 행렬에서 코드율별로 VNUdf degree와 필요한 VNUdf 수가 표 1의 2행과 3행에 나타나 있다. Hd의 기저 행렬의 열의 수가 필요한 VNUd의 수이고, VNUd는 코드율에 따라 제공되는 테이블에 의해 VNUdf의 수와 VNUd3의 수로 나누어진다. VNUp의 수가 CNU의 수와 같고 CNU degree 또한 코드율별로 달라진다. k는 정보 비트의 수이다. 여기서 degree는 행렬 상에서의 행과 열 각각에 존재하는 1의 개수를 의미한다.

이러한 조건에서는 모드 별로 VNU와 CNU가 각기

표 1. 코드율별로 필요한 VNU와 CNU 개수
Table 1. numbers of VNUs and CNUs for each code rate.

code rate	VNUdf degree	VNUdf	VNUd3	VNUp CNU (q)	CNU degree	k (un-coded)
1/4	12	15	30	135	4	16,200
1/3	12	20	40	120	5	21,600
2/5	12	24	48	108	6	25,920
1/2	8	36	54	90	7	32,400
3/5	12	36	72	72	11	38,880
2/3	13	12	108	60	10	43,200
3/4	12	15	120	45	14	48,600
4/5	11	18	126	36	18	51,840
5/6	13	15	135	30	22	54,000
8/9	4	20	140	20	27	57,600
9/10	4	18	144	18	30	58,320

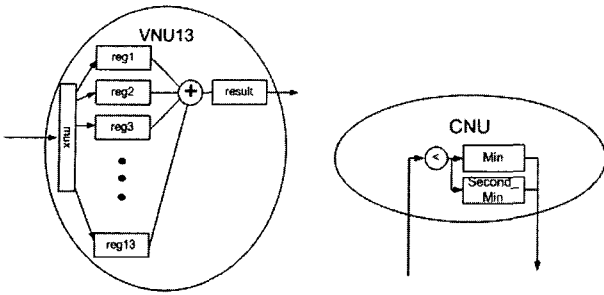


그림 3. One-port 구조의 VNU와 CNU
 Fig. 3. structure of one-port VNU and one-port CNU.

표 2. Multi-port VNU와 one-port VNU의 연산기 수와 레지스터 수 비교

Table 2. Comparison of number of processors and registers for a multi-port VNU with those of a one-port VNU.

	signed magnitude sum	number of register
multi-port VNU	$d+1 * (d \text{ operands})$	$d+n^{**}$
one-port VNU	1	d

*d: VNU의 degree, **: pipeline 레지스터 개수($d/2$ 이하)

존재해야 하기 때문에 하드웨어의 크기가 커지는 단점이 있다. 또한 행과 열의 degree 수에 따라 VNU와 CNU의 포트 수가 결정되기 때문에 재사용하기 힘들다. 이러한 문제를 해결하여 하나의 VNU 구조와 CNU 구조를 이용하여 코드율별로 재사용하기 위해 그림 3의 구조를 제안한다. VNU13은 degree가 13인 VNU를 나타내는데, degree가 달라질 경우 읽어오는 정보의 개수를 제한하여 다른 VNU degree라도 재사용이 가능하다. degree 값을 d 라고 할 때, multi-port VNU의 degree 만큼의 피연산자를 이용하여 d 개의 피연산자를 갖는 signed magnitude 덧셈 연산기가 $d+1$ 개가 필요한 반면, one-port의 VNU구조는 2개의 피연산자를 갖는 signed magnitude 덧셈 연산기가 1개로 연산이 가능하다. 연산을 위한 소요 클럭 수가 $d*2$ 만큼 증가하지만, 메모리에서 연산된 값을 쓰면서 새로운 데이터를 읽어오는 병렬 프로세싱이 가능하기 때문에 d 만큼의 클럭 수로 줄일 수 있다.

표 2는 multi-port VNU와 one-port VNU를 사용할 때 연산기 수와 레지스터 수를 비교한 것이다. Multi-port VNU가 조합 회로로 구현되기 때문에, 연산 로직이 커지게 되면 동작 주파수가 줄어들게 된다. 이를 피하기 위하여 파이프라인 레지스터가 추가 될 수 있으므로 multi-port VNU의 레지스터 수는 최소 d 개에 파이프라인 레지스터 수만큼 증가한다.

3. 효율적인 블록 메모리 배치

Partly parallel 방식으로 복호하기 위해서는 모든 CNU가 동시에 연산하고, 다음 순서에 모든 VNU가 동시에 연산하게 된다. 한 순간에 모든 CNU가 필요한 데이터를 읽어오기 위해서 필요한 데이터 수만큼의 블록 메모리가 필요하다. 코드율이 3/5일 때, CNU가 읽어올 데이터가 792개로 제일 많아서 최대 필요한 블록 메모리는 792개이다. 반대로 VNU가 읽어올 데이터 또한 792개이다. 코드율 3/5를 기준으로 하여 메모리를 배치하고 나서 각각 코드율별로 가장 적게 연산기를 사용하기 위해 VNU를 기준으로 배치를 하였다. 왜냐하면 VNU는 코드율이 달라도 같은 degree를 사용하는 경우가 있어서 재사용률이 CNU에 비하여 더 높기 때문이다. 결과적으로 degree가 최대 13까지 처리 가능한 VNU13이 15개, degree가 최대 12까지 처리 가능한 VNU12가 21개, degree가 최대 3까지 처리 가능한

표 3. 모든 동작 모드에 필요한 multi-port VNU 개수
 Table 3. Number of multi-port VNUs for all code rates.

degree	VNU13	VNU12	VNU8	VNU11	VNU4	VNU3	VNU2
개수	15	36	36	18	20	144	135

표 4. 코드율별로 사용되는 VNU 개수
 Table 4. Number of VNUs for each code rate.

code rate	VNU _d		VNU _d 3		VNU _p	
	VNU13	VNU12	VNU3		VNU3	VNU12
1/4	VNU13	VNU12	VNU3		VNU3	VNU12
	15	0	30		127	2
1/3	VNU13	VNU12	VNU3		VNU3	VNU12
	15	5	40		111	2
2/5	VNU13	VNU12	VNU3		VNU3	VNU12
	15	9	48		99	2
1/2	VNU13	VNU12	VNU3		VNU3	
	15	21	54		90	
3/5	VNU13	VNU12	VNU3		VNU3	VNU13
	15	21	72		65	*15
2/3	VNU13	VNU12	VNU3		VNU3	VNU12
	12	0	108		9	9
3/4	VNU13	VNU12	VNU3	VNU12	VNU12	
	15	0	115	1	14	
4/5	VNU13	VNU12	VNU3	VNU12	VNU12	
	15	3	115	4	12	
5/6	VNU13	VNU12	VNU3	VNU12	VNU12	
	15	0	115	5	10	
8/9	VNU13	VNU12	VNU3	VNU12	VNU12	
	15	5	115	9	3	
9/10	VNU13	VNU12	VNU3	VNU12	VNU12	
	15	3	115	10	3	

*재사용

code rate 2/3							
0	177	116	150	235	34	236	8
code rate 3/5							
105	392	269	9	118	302	184	289
396							

그림 4. 코드율 2/3과 3/5일 때, 연산에 필요한 블록 메모리 인덱스
 Fig. 4. Indices of block memory needed to operate when code rates are 2/3 and 3/5, respectively.

VNU3이 115개가 필요하다.

One-port VNU를 사용하지 않는 DVB-S2 복호기 구조에서 필요한 multi-port VNU 수는 표 3과 같다. 표 4는 배치된 블록 메모리에 연결된 최소한의 VNU 수를 나타낸다. 가장 효율적으로 블록 메모리를 배치하여도 VNU3이 40개, VNU2가 68개가 더 필요하다. 그러나 표 3과 비교하면 one-port VNU구조를 사용하는 것이 하드웨어 면적 측면에서 훨씬 이득임을 알 수 있다.

현재 배치된 메모리 블록을 순차적으로 색인하여 코드율 별로 CNU가 연산에 필요한 메모리 블록 인덱스들이 거의 다르다. 그림 4는 코드율 2/3, 3/5일 때 CNU가 읽어야 할 블록 메모리 인덱스를 나타내는데 모두 값이 다름을 알 수 있다. 결국 one-port CNU를 사용하더라도 코드율별로 재사용이 불가능하므로 하드웨어 면적과 와이어 복잡도의 감소가 거의 없다. 그러므로 multi-port 구조의 CNU를 사용하는 것이 전체 throughput 측면에서 효율적이다. 최종적으로 CNU는 CNU degree 만큼의 포트를 갖는 블록이 각각 q개씩 필요하다.

4. DVB-S2 LDPC 복호기 구조

제안하는 DVB-S2 LDPC 복호기 구조는 그림 5와 같다. Shuffling 네트워크는 블록 메모리와 VNU, CNU의 연산 모듈 인터페이스를 연결해주는 네트워크 로직이다. 300MHz로 동작했을 때의 예상 성능이 표 5에 나타나 있다. 그리고 multi-port VNU 구조로 구현되었을 경우의 예상 성능도 포함되어 있다.

코드율 3/5와 2/3일 때, 1/2인 경우 보다 throughput이 감소된 이유는 VNUdf의 degree가 1/2일 때보다 더 많기 때문이다. 코드율 5/6도 4/5보다 감소되었다. 그러나 각 경우에 한계 throughput만 넘기지만 하면 위성 방송 시청에는 문제가 없다.

표 6은 DVB-S2 LDPC 복호기를 구현한 논문들의 성능과 multi-port VNU 구조로 구현된 성능을 비교한 것이다^{[11][12][13][14]}. [11]과 [12]는 130nm CMOS technology

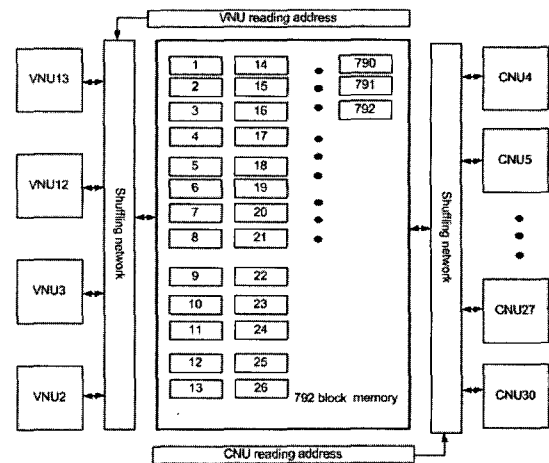


그림 5. DVB-S2 LDPC 복호기 구조
 Fig. 5. Architecture of DVB-S2 LDPC decoder.

표 5. 300MHz로 동작했을 때의 one-port VNU와 여러 동작 주파수별 multi-port VNU일 때의 예상 성능

Table 5. Estimated performance of one-port VNU at 300MHz and multi-port VNU at various operating frequencies.

code rate	one-port VNU (Mbps)@300MHz		multi-port VNU		
	1 iteration	30 iteration	(Mbps) @100MHz 30 iter	(Mbps) @200MHz 41 iter	(Mbps) @270MHz 30 iter
1/4	964	32	37	54	101
1/3	1285	42	50	73	135
2/5	1542	51	60	87	162
1/2	2700	90	75	109	202
3/5	2314	77	90	131	243
2/3	2400	80	100	146	270
3/4	2892	96	112	164	303
4/5	3323	110	120	175	324
5/6	3000	100	125	182	337
8/9	8000	266	133	195	360
9/10	8100	270	135	197	364

를 이용한 결과이고 [13]은 90nm CMOS technology를 이용한 결과이다. [14]의 경우는 16,200의 짧은 프레임에 코드율 1/4로 구현한 복호기이다. 본 논문은 one-port VNU에 비해 구현이 비교적 쉬운 multi-port VNU 구조로 64,800 프레임에 코드율 1/2로 구현하였다. 동작 주파수는 0.35um CMOS technology를 이용하여 얻은 결과로 90nm CMOS technology일 경우에는 더 큰 동작 주파수와 throughput을 얻을 것으로 예상된다.

표 6의 2열을 보면 제안한 구조의 면적이 [13]을 제외하고는 작다. [13]의 경우는 throughput이 감소하는 대신 면적과 와이어 복잡도를 최소화하기 위한 구조이다. 이는 one-port VNU 구조의 설계 목적과 비슷하다. 그러므로 [13]은 one-port VNU 구조로 구현한 복호기

표 6. DVB-S2 LDPC decoder 구현 사례 비교
Table 6. Comparison of DVB-S2 LDPC decoders.

	[11]	[12]	[13]	[14]**	Proposed
Gate-counts (logics)	2M 32.8mm ²	11.55mm ²	1.1mm ²	2,454,480	0.5M
Memory (bits)	2.921M 16.7mm ²	11.189 mm ²	3.1mm ²	797,520	1,749,600
wire complexity	*(360xq)	*(360xq)	*(45xq)	*(360xq)	*(630xq)
Throughput	90Mbps (41-77iter)	255Mbps (30 iter)	90Mbps (30 iter)	NA	75Mbps (30 iter)
processor	360	360	**45	360	180
operating freq.	200MHz	270MHz	300MHz	NA	100MHz
Technology	130nm	130nm	90nm	NA	0.35um

*q=6bits, **number of data paths

*** Short frame (16,200), code rate 1/4

와 비교해야 공정한 구조 비교 방식일 것이다. One-port VNU 구조의 동작 주파수가 300MHz일 때 예상 성능이 유사함을 표 5의 3행에서 확인할 수 있다. 표 6의 3열은 연산에 필요한 메모리도 다른 구조에 비해 제한한 구조가 작다는 것을 보여준다. [14]의 경우는 16,200의 짧은 프레임이므로 필요 메모리가 훨씬 작다. 제한한 구조가 와이어 복잡도가 다소 크다는 점을 제외하면 면적이나 메모리 그리고 throughput 측면에서 모두 우수함을 알 수 있다.

IV. 결 론

부호기와 복호기의 복잡도를 감소시키는 hybrid H-matrix의 방식을 이용하여 DVB-S2 규격의 LDPC 복호기 구조를 제안하였다. 그 결과 30번 반복 복호를 하여도 방송의 경우 1/2 이상의 코드율에서 90Mbps@300MHz 정도의 성능이 예측되어 충분히 실효성이 있다. 11개의 코드율에 따라 달라지는 VNU는 one-port VNU를 재사용하여 면적과 와이어 복잡도를 감소시킬 수 있다. 그리고 hybrid H-matrix의 multi-port VNU 구조로 코드율 1/2의 복호기 구현 결과 기존 구현 결과에 비해 면적과 메모리가 작으면서 성능이 우수함이 입증 되었다. DVB-S2 LDPC 복호기에 적용된 hybrid H-matrix 구조가 다양한 응용 서비스에서의 채널 코덱 기능을 충분히 수행할 수 있으리라 기대된다.

참 고 문 헌

- [1] R. G. Gallager, "Low density parity check codes", IRE Trans. Inform. Theory, vol. IT-8, pp. 21-28, Jan. 1962.
- [2] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes", Electron. Lett., vol. 32, pp. 1645-1646, Aug. 1996.
- [3] S.-Y. Chung, G. D. Forney Jr., T. J. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045dB of the Shannon limit", IEEE Commun. Lett., vol. 5, pp. 58-60, Feb. 2001.
- [4] Tong Zhang, Z. Wang, and K. K. Parhi, "On finite precision implementation of low-density parity-check codes decoder", in Proc. of 2001 IEEE Int. Symp. on Circuits and Systems (ISCAS), vol. 4, pp. 202-205, Sydney, Australia, May 2001.
- [5] DVB-S2 ETSI EN 302 307 V1.1.1 (2005-03), Digital Video Broadcasting-Satellite version 2, ETSI, 2005.
- [6] 한재선, 이찬호, "Efficient LDPC coding using a hybrid H-matrix", 2003 SOC design conference, pp.884-887, 2003.
- [7] Chanho Lee, Jaegeun Park, Jee Myong Lee, and Kwang Yup Lee, "Design of encoder and decoder for LDPC codes using hybrid H-matrix," IFIP VLSI-SoC, pp.317-320 Perth, Australia. Oct. 2005.
- [8] Li Ping, W. K. Leung, and Nam Phamdo, "Low density parity check codes with semi-random parity check matrix", IEE Electronics Lett., vol. 35, pp. 38-39, Jan. 1999.
- [9] T. Zhang, and K. K. Parhi, "VLSI implementation-oriented (3,k)-regular low-density parity check codes", IEEE Workshop, signal processing systems(SiPS), pp. 25-36, Antwerp, Belgium, Sept. 2001.
- [10] Hao Zhong and Tong Zhang, "Design of VLSI Implementation-Oriented LDPC codes", IEEE Vehicular Technology Conference, Orlando, USA, Oct. 2003.
- [11] P. Urard et.al. "A 135Mb/s DVB-S2 compliant codec based on 64800b LDPC and BCH codes", In IEEE Solid-state Circuits Conference (ISSCC), vol. 1, pp.446-609, Feb. 2005.
- [12] F. Kienle, T. Brack, and N. Wehn. "A synthesizable IP core for DVB-S2 LDPC code decoding", In IEEE Conference on Design Automation and Test in Europe (DATE), vol. 3,

pp.100-105, Mar. 2005.

- [13] J. Dielissen, A Hekstra, and V.Berg, "Low cost LDPC decoder for DVB-S2", In IEEE Conference on Design Automation and Test in Europe (DATE), vol 2, pp.06-10, Mar. 2006.
- [14] 김민정, "차세대 유럽형 디지털 위성 방송을 위한 LDPC decoder 설계", 이화여자대학교 과학기술대학원 석사학위 논문, 2005.

저 자 소 개



박 재 근(학생회원)
 2005년 숭실대학교
 전자공학과 학사졸업.
 2005년~현재 숭실대학교
 전자공학과 석사재학.
 <주관심분야 : 채널 코덱의 구현,
 SoC 설계방법론>



이 찬 호(정회원)
 1987년 서울대학교
 전자공학과 학사졸업.
 1989년 서울대학교 대학원
 전자공학과 석사졸업.
 1994년 University of California,
 Los Angeles 전자공학과
 박사졸업.
 1994년 8월~1995년 2월 삼성전자 반도체연구소
 선임연구원.
 1995년 3월~현재 숭실대학교 정보통신전자
 공학부 부교수.
 <주관심분야 : SoC on-chip-network, 3D 그래픽
 프로세서 설계, 채널코덱의 구현, SoC 설계방법
 론, H.264 codec 구현>