

# 높은 throughput 성능을 갖는 DVB-S2 LDPC 부호의 복호기 구현

준회원 김성운\*, 정회원 박창수\*, 황선영\*

## Implementation of High Throughput LDPC Code Decoder for DVB-S2

Seong-Woon Kim\* *Associate Member*, Chang-Soo Park\*, Sun-Young Hwang\* *Regular Members*

### 요 약

본 논문은 광대역 위성 서비스를 위한 유럽 전기통신 표준화기구의 2세대 표준인 DVB-S2에서 사용하는 LDPC 부호의 throughput을 증가시키기 위한 새로운 복호기 구조를 제안한다. 제안한 구조는 IRA 구조의 LDPC 부호가 가지는 특징을 이용해 360개의 비트노드와 체크노드를 각각 그룹핑한다. 노드 그룹을 구현한 연산모듈은 각각 로컬 메모리를 가지고 있고, 전달받은 메시지는 자신의 로컬 메모리에서만 읽는다. 제안한 구조는 메시지 라우팅 로직을 이용해 에지로 연결된 노드 그룹의 로컬 메모리에 메시지를 저장함으로써 메모리 충돌이 없고 순차적인 메모리 접근을 가능하게 하여 복호기의 throughput을 증가시킨다. 제안한 DVB-S2 LDPC 복호기 구조는 TSMC 90nm 공정으로 합성하였고 F. Kienle과 J. Dielissen이 각각 제안한 기존의 구조보다 throughput이 각각 104%, 478%가 증가함을 확인하였다.

**Key Words** : LDPC, DVB-S2, Decoder, Local memory, High throughput

### ABSTRACT

This paper proposes a novel LDPC code decoder architecture to improve throughput for DVB-S2, a second generation standard of ETSI for satellite broad-band applications. The proposed architecture clusters 360 bitnodes and checknodes into groups utilizing the property of IRA-LDPC code. Functional modules which perform calculations for bitnode groups and checknode groups have local memories and store the messages from the other type of functional modules connected by edges at their local memories. The proposed architecture can avoid memory conflicts by accessing stored messages sequentially, hence, increases throughput in the proposed DVB-S2 LDPC code decoder architecture. The proposed architecture was synthesized using the TSMC 90nm technology. Synthesis results show that throughput of the proposed architecture is improved by 104% and 478%, respectively, when compared with those of the architectures proposed by F. Kienle and J. Dielissen.

### I. 서 론

DVB-S2는 광대역 위성 서비스를 위한 유럽 전기

통신 표준화기구(ETSI)의 2세대 표준이다<sup>[1][2]</sup>. DVB-S2는 1세대 표준인 DVB-S에 비해 30%의 용량 이득을 가지며, 이 용량 이득은 DVB-S2에서 적용한 고차

※ 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음(IITA-2008-(C1090-0801-0012))

\* 서강대학교 전자공학과 대학원 CAD & ES 연구실

논문번호 : KICS2008-07-302, 접수일자 : 2008년 7월 7일, 최종논문접수일자 : 2008년 8월 21일

변조 방식과 새로운 FEC 시스템에 의해 가능하였다<sup>[2]</sup>. DVB-S2는 Convolution 부호와 Reed-Solomon 부호를 연결한 FEC(Forward Error Correcting) 시스템을 가진 DVB-S와 달리 BCH 부호와 LDPC 부호의 연결하여 FEC 시스템을 구성하였다.

LDPC 부호는 Gallager가 제안한 선형 블록 부호로서, Tanner 그래프로 표현할 수 있는 패리티 검사 행렬  $H$ 로 정의된다<sup>[3][4]</sup>. Tanner 그래프는 LDPC 부호를 일반화하여 표현한 이분(bipartite) 그래프이며 비트노드와 체크노드, 에지(edge)로 이루어진다. 에지는  $H$  행렬의 원소가 1인 행을 인덱스로 가지는 비트노드와 열을 인덱스로 가지는 체크노드 사이에 존재한다. 간단한 선형 블록 부호인 (7,4) 해밍 부호에 대한  $H$  행렬과 Tanner 그래프의 예를 그림 1에 보인다.

LDPC 부호는 Tanner 그래프 상의 비트노드와 에지로 연결된 체크노드 사이에 확률값을 가진 메시지를 반복적으로 전달하면서 신뢰성 있는 확률값을 가지도록 하는 메시지 전달 알고리즘들을 사용해 복호한다<sup>[5]</sup>. 메시지 전달 알고리즘에서 사용하는 심볼의 확률값은 실제 확률값을 사용하는 방법과 LLR(Log Likelihood Ratio)를 사용하는 방법이 있다. LLR을 사용하는 방법은 복호 과정에서 실제 확률값의 곱셈 연산을 덧셈 연산으로 대체할 수 있어 실제 확률값을 사용하는 방법보다 효율적이다<sup>[6]</sup>.

일반적으로 LDPC 부호는 매우 큰 부호화 복잡도를 가진다<sup>[7][8]</sup>. DVB-S2에서는 LDPC 부호의 부호화 복잡도 문제를 해결하기 위해 IRA(Irregular Repeat-Accumulate) 구조를 가지는 LDPC 부호를 사용한다<sup>[9]</sup>. IRA 구조의 LDPC 부호는 선형 부호화 복잡도를 가지며 일반적인 LDPC 부호에 비해 크게 열화되지 않는 성능을 가진다<sup>[10]</sup>. DVB-S2에서는 부호화 테이블을 사용하여 LDPC 부호를 표현하였으며, 부호화 테이블을 사용한 간단한 연산과정을 통해 DVB-S2에서 사용하는 LDPC 부호의 패리티 검사 행렬을 생성할 수 있다.

DVB-S2 표준은 HDTV와 같은 고품질 위성방송, 양방향 데이터 통신 등 다양한 서비스를 제공한다<sup>[11]</sup>. 제공하는 서비스의 종류가 다양하고 데이터의 용량

도 커짐에 따라 수신기에서 데이터 처리에 소요되는 시간이 증가한다. 따라서 충분한 데이터 처리 시간을 확보하기 위해 FEC 시스템의 throughput을 높게 할 필요가 있다. F. Kienle는 255Mbps의 throughput을 가지는 복호기 구조를 제안하였다<sup>[11]</sup>. F. Kienle가 제안한 구조는 DVB-S2 LDPC 부호의 특징인 360을 주기로 가지는 주기성을 이용해 360개의 연산모듈을 사용하였다. 많은 수의 연산모듈을 사용해 throughput에 비해 면적이 크다는 단점을 가진다. J. Dielissen은 면적을 최소화 하기 위해 45개의 연산모듈과  $H$  행렬을 재배치한 행렬을 사용하여 복호를 수행하는 복호기 구조를 제안하였다<sup>[12]</sup>. 그 결과 90Mbps의 비교적 낮은 throughput을 갖는다.

본 논문에서는 기존의 복호기 구조보다 높은 throughput을 갖는 DVB-S2 LDPC 복호기 구조를 제안한다. 제안한 복호기 구조는 Tanner 그래프 상에서 특정한 패턴을 가지는 360개의 비트노드와 체크노드를 각각 비트노드 그룹과 체크노드 그룹으로 그룹핑한다. 또 노드 그룹은 연산모듈로 구현하고, 연산모듈은 로컬 메모리를 가지도록 하여 메모리 충돌이 없는 단순화 된 메모리 접근 패턴을 가진다.

본 논문의 II절에서는 DVB-S2 LDPC 부호와 기존의 LDPC 복호기 구조를 소개하고 III절에서는 DVB-S2 LDPC 부호의  $H$  행렬의 특징을 이용한 DVB-S2 LDPC 복호기의 구조를 제안한다. IV절에서는 제안한 복호기 구조의 성능을 평가하며, 마지막으로 V절에서 결론을 맺는다.

## II. DVB-S2 LDPC 부호

본 절에서는 DVB-S2에서 사용하는 LDPC 부호와 기존에 연구된 DVB-S2 LDPC 부호의 복호기 구조를 간단히 소개한다.

### 2.1 DVB-S2 LDPC 부호

DVB-S2 LDPC 부호는 64,800 비트의 부호어 길이를 가진다. DVB-S2는 긴 부호어 길이로 인한 부호화 복잡도를 최소화 할 수 있도록 선형 부호화 복잡도를 가지는 IRA 구조의 LDPC 부호를 사용한다. DVB-S2 LDPC 부호기는  $k$ 의 길이를 가지는 정보 블록  $\mathbf{i} = (i_0, i_1, \dots, i_{k-1})$ 를 입력으로 받아 부호화 테이블을 사용한 간단한 연산을 통해  $n-k$  길이의 패리티 블록  $\mathbf{p} = (p_0, p_1, \dots, p_{n-k-1})$ 를 생성하고  $\mathbf{i}$ 와  $\mathbf{p}$ 를 연결함으로써 부호어  $\frac{k}{n}$ 의 부호어  $\mathbf{c} = (i_0, i_1, \dots, i_k, p_0, p_1,$

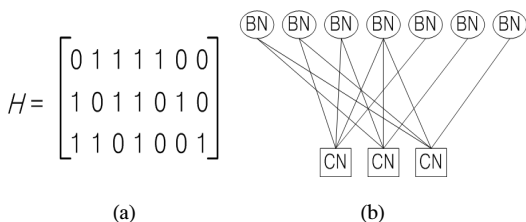


그림 1. (7,4) Hamming 부호 (a) H 행렬, (b) Tanner 그래프

...  $p_{n-k-1}$ )를 생성한다<sup>[1]</sup>. DVB-S2 LDPC 부호는  $\frac{1}{4}$ 부터  $\frac{9}{10}$ 까지 모두 11개의 부호율을 지원하며, 각 부호율에 따른 부호화 테이블을 가지고 있다. DVB-S2 LDPC 부호는 모든 패리티 비트를 0으로 초기화 한 후 식 (1)의 연산을 모든 정보 비트와 부호화 테이블에 대해 수행함으로써 패리티 비트를 업데이트한다.

$$\begin{aligned}
 p_j &= p_j \oplus i_m & (1) \\
 j &= \{x + q(m \bmod 360)\} \bmod (n - k) \\
 m &= 0, 1, 2, \dots, k - 1
 \end{aligned}$$

식 (1)에서  $q$ 는 부호율에 의해 정의되는 값이고  $n$ 과  $k$ 는 각각 부호어의 길이와 정보 블록의 길이이며,  $x$ 는 부호화 테이블 중  $\lfloor \frac{m}{360} \rfloor$  번째 열의 값이다. DVB-S2 LDPC 부호는 부호화 과정에서 식 (1)의 덧셈 연산이 반복적으로 수행되므로 IRA 구조를 가진다. DVB-S2 LDPC 부호의  $H$  행렬 구조는 그림 2에 제시하였으며<sup>[9]</sup>, 그림 2의  $H$  행렬을 표현한 Tanner 그래프는 그림 3과 같다. 그림 2의  $H$  행렬은  $A$  행렬과  $B$  행렬로 나뉜다.  $A$  행렬은 부호어  $c = (i, p)$  중 정보 블록인  $i$ 와 곱해 패리티 검사식을 생성하고  $B$  행렬은 패리티 블록인  $p$ 와 곱해 패리티 검사식을 생성한다.  $H$  행렬을 표현한 그림 3의 Tanner 그래프의 비트노드는 정보 비트노드와 패리티 비트노드로 구분된다. 정보 비트노드는 부호어 중 정보 블록인  $i$ 에 해당하는 위치의 비트노드이고 체크노드와 연결이 불규칙적이다. 패리티 비트노드는 부호어 중 패리티 블록인  $p$ 에 해당하는 위치의 비트노드이고, 체크노드와 지그재그의 규칙적인 형태로 연결된다. 파선으로 나타낸 에지는 불규칙적인 연결 관계의 에지이고 실선으로 나타낸 에지는 규칙적인 연결 관계의 에지이다.

$$H = \left[ \begin{array}{c|c} A_{(n-k) \times k} & B_{(n-k) \times (n-k)} \end{array} \right]$$

$$= \left[ \begin{array}{cccc|cccc}
 \alpha_{0,0} & \alpha_{0,1} & \dots & \dots & \alpha_{0,k-1} & 1 & 0 & \dots & \dots & \dots & 0 \\
 \alpha_{1,0} & \alpha_{1,1} & \dots & \dots & \alpha_{0,k-1} & 1 & 1 & 0 & \dots & \dots & \vdots \\
 \alpha_{2,0} & \alpha_{2,1} & \dots & \dots & \alpha_{0,k-1} & 0 & 1 & 1 & \dots & \dots & \vdots \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 \alpha_{n-k-2,0} & \alpha_{n-k-2,1} & \dots & \dots & \alpha_{n-k-2,k-1} & \dots & \dots & \dots & 1 & 1 & 0 \\
 \alpha_{n-k-1,0} & \alpha_{n-k-1,1} & \dots & \dots & \alpha_{n-k-1,k-1} & 0 & \dots & \dots & 0 & 1 & 1
 \end{array} \right]$$

그림 2. DVB-S2 LDPC 부호의 패리티 검사 행렬

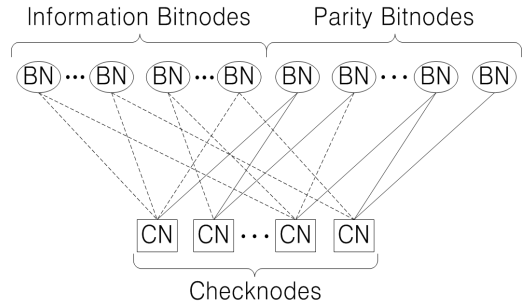


그림 3. DVB-S2 LDPC 부호의 Tanner 그래프

### 2.2 DVB-S2 LDPC 부호의 기존 복호기 구조

LDPC 복호기의 구조는 크게 직렬 복호기 구조와 병렬 복호기 구조, 부분 병렬 복호기 구조로 구분된다<sup>[13]</sup>. 직렬 복호기 구조는 적은 수의 비트노드 연산모듈과 체크노드 연산모듈을 가지고 있으며 각각의 연산모듈이 모든 비트노드와 체크노드의 연산을 수행한다. 연산모듈의 수가 적어 복호기의 면적은 작지만, throughput 성능이 좋지 않다<sup>[14]</sup>. 병렬 복호기 구조는 Tanner 그래프의 각 노드가 직접 비트노드 연산모듈과 체크노드 연산모듈로 구현된다. 하나의 노드가 하나의 연산모듈과 대응되어 빠른 속도로 복호가 가능하지만 복호기의 면적이 크다<sup>[15]</sup>. DVB-S2 LDPC 부호는 긴 부호어 길이로 인해 병렬 구조의 복호기는 실제로 구현할 수 없고 직렬 구조의 복호기는 DVB-S2에서 요구하는 throughput을 만족할 수 없다. 따라서 부분 병렬 구조의 복호기만이 구현 가능하다<sup>[11]</sup>.

F. Kienle는 360개의 연산모듈을 사용한 부분 병렬 구조의 복호기 구조를 제안하였다<sup>[11]</sup>. 제안된 복호기 구조는 Tanner 그래프 상의 에지를 정보 비트노드-체크노드 에지와 패리티 비트노드-체크노드 에지로 나눈다. 모든 정보 비트노드-체크노드 에지는 부호화 테이블에 의해 정해진 360개의 정보 비트노드-체크노드 에지를 사용하여 표현 가능하다. 360개의 에지는 각각 메모리 뱅크로 구현되고 360개의 메모리 뱅크를 사용해 에지를 통해 전달되는 모든 메시지를 저장한다. 360개의 연산모듈은 비트노드 연산과 체크노드 연산을 모두 수행하며 각 메모리 뱅크로부터 메시지를 전달받는다. 연산모듈은 한 클럭 사이클마다 새로운 메시지를 입력받아 메시지 전달 알고리즘에 따라 전달할 메시지를 계산하고, 계산한 값을 shuffling 네트워크를 통해 메모리에 저장하여 다음 반복 과정에서 사용한다. 이상의 과정을 통해 제안된 복호기는 255Mbps의 throughput을 가진다.

J. Diehlissen은 DVB-S2 LDPC 부호의  $H$  행렬이  $360 \times 360$  크기의 블록들로 구성된 사실을 이용한 복호기 구조를 제안하였다<sup>[12]</sup>. 제안된 복호기 구조는 메시지 전달 알고리즘에 근접한 성능을 가지고 하드웨어 복잡도를 감소시킬 수 있는 UMP-BP 알고리즘과 staggered 복호 기법을 사용하였다. UMP-BP 알고리즘은 체크노드 연산 과정에서 LUT을 사용하지 않고, 전달받은 메시지의 최소값을 비트노드에 전달한다. Staggered 복호 기법은 반복 복호 과정에서 메모리에 저장되어 있는 메시지를 읽은 후 다음 반복 복호 과정이 시작될 때까지 메시지를 유지하지 않고, 메시지를 읽은 후 곧바로 다음 반복 복호를 위한 메시지를 해당 위치에 저장하는 기법이다. 참고문헌 [12]의 복호기 구조는 Staggered 복호 기법과 UMP-BP 알고리즘을 사용해 메모리 사용을 참고문헌 [11]의 구조에 비해 절반으로 줄일 수 있었다. 반면에 제안된 복호기는 90Mbps의 throughput을 가진다.

### III. 제안한 LDPC 복호기 구조

#### 3.1 노드 그룹핑

DVB-S2 LDPC 부호의  $H$  행렬이 가지는 특징은 식 (2)와 식 (3)으로 표현할 수 있다.

$$\alpha_{r+q \times i, (c+i) \bmod 360 + \lfloor \frac{c}{360} \rfloor \times 360} = \alpha_{r,c} \quad (2)$$

$$i = 0, 1, 2, \dots, 359$$

$$r = 0, 1, \dots, q-1$$

$$c = 0, 1, \dots, k-1$$

$$\beta_{(r+1) \bmod (n-k), (c+1) \bmod (n-k)} = \beta_{r,c} \quad (3)$$

$$\beta_{0,0} = 1$$

$$\beta_{1,0} = 1$$

$$r = c = 0, 1, \dots, n-k-1$$

식 (2)와 식 (3)에서  $n$ 은 부호어의 길이,  $k$ 는 정보 블록의 길이이며,  $\alpha$ 와  $\beta$ 는 각각  $A$  행렬과  $B$  행렬의 원소이고  $r, c$ 는 각각  $\alpha$ 와  $\beta$ 의 열과 행 인덱스이다. 식 (2)와 식 (3)으로 표현된  $H$  행렬의 특징에 의해 Tanner 그래프의 비트노드와 체크노드는 비트노드 그룹과 체크노드 그룹으로 그룹핑 할 수 있다. 부호율  $\frac{1}{2}$ 의 DVB-S2 LDPC 부호는 부호어 길이  $n=64800$ 이고 정보어 길이  $k=32400$ 이며,  $q=90$ 의 값을 가진다<sup>[1]</sup>. 따라서 식 (2)와 식 (3)은 부호율  $\frac{1}{2}$ 에서 식 (4)와 식 (5)로 표현된다.

$$\alpha_{r+90 \times i, (c+i) \bmod 360 + \lfloor \frac{c}{360} \rfloor \times 360} = \alpha_{r,c} \quad (4)$$

$$i = 0, 1, 2, \dots, 359$$

$$r = 0, 1, \dots, 89$$

$$c = 0, 1, \dots, 32399$$

$$\beta_{(r+90) \bmod 32400, (c+90) \bmod 32400} = \beta_{r,c} \quad (5)$$

$$\beta_{0,0} = \beta_{1,0} = 1$$

$$r = c = 0, 1, \dots, 32399$$

$\alpha_{r,c} = 1$ 은 인덱스  $c$ 의 정보 비트노드와 인덱스  $r$ 의 체크노드 사이에는 에지가 존재함을 의미한다. 식 (4)는  $\alpha_{r,c} = 1$ 일 때  $r$ 로부터 90씩 증가한 인덱스의 체크노드와  $c$ 로부터 1씩 증가한 인덱스의 정보 비트노드 사이에는 에지가 존재함을 나타낸다. 이 때, 정보 비트노드 인덱스는 1 증가 후에 모듈로(modulo) 360 연산을 수행하므로 항상 연속적인 360개의 비트노드만을 가리킨다. 이 사실을 이용해 360개의 연속적인 정보 비트노드를 하나의 정보 비트노드 그룹으로 그룹핑하고, 인덱스가 90씩 증가하는 360개의 체크노드를 하나의 체크노드 그룹으로 그룹핑한다. 식 (5)는  $\beta_{r,c} = 1$ 일 때  $r$ 과  $c$ 로부터 각각 90씩 증가한 인덱스의 체크노드와 패리티 비트노드 사이에는 에지가 존재함을 나타낸다. 식 (5)의 사실을 이용해 90씩 증가하는 360개의 비트노드로 이루어진 패리티 비트노드를 하나의 패리티 비트노드 그룹으로 그룹핑한다.

부호율  $\frac{1}{2}$ 에 대한 그룹핑 결과를 그림 4에 보인다. 64,800개의 비트노드는 90개의 정보 비트노드 그룹과 90개의 패리티 비트노드 그룹으로 그룹핑하고, 32,400개의 체크노드를 90개의 체크노드 그룹으로 그룹핑한다. 각 노드 그룹은 노드 연산을 수행하는 연산모듈로 구현하며 연산모듈 사이의 메시지 전달은 메시지 라우팅 로직과 메모리로 구현한다.

#### 3.2 메모리 접근 패턴

메시지 전달 알고리즘에 의한 복호 과정을 하드웨어로 구현할 때, 메시지 전달은 메모리를 통해 이루어진다. 각 연산모듈은 전달할 메시지를 메모리에 저장하고, 해당 연산모듈의 상대 연산모듈은 저장된 메시지를 읽어 메시지 연산 후 메모리에 저장한다. 본 논문에서는 모든 연산모듈이 로컬 메모리를 가지는 구조를 제안한다. 연산모듈은 로컬 메모리에서 순차적으로 메시지를 읽어 에지로 연결된 연산모듈로 전달할 메시지를 계산한다. 하나의 연산모듈에서 계

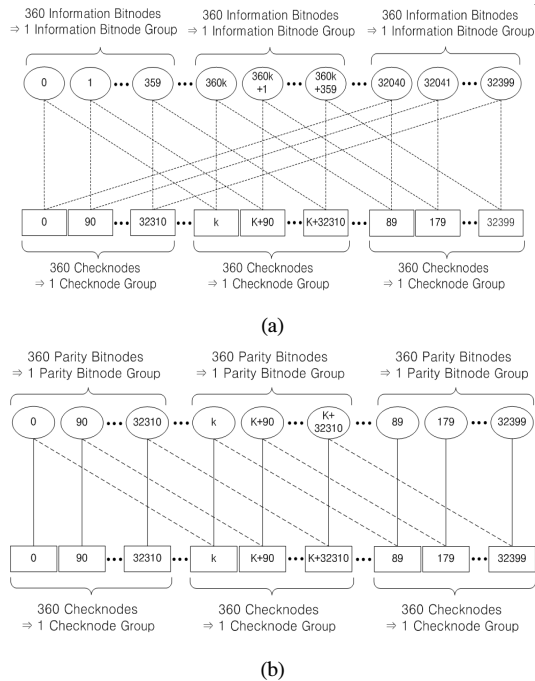


그림 4. 비트노드 그룹과 체크노드 그룹. (a) 정보 비트노드 그룹과 체크노드 그룹의 Tanner 그래프. (b) 패리티 비트노드 그룹과 체크노드 그룹의 Tanner 그래프

산된 메시지는 전달되어질 연산모듈에 따라 다른 값을 가지므로, 메시지를 정확한 위치의 메모리에 저장해야 할 필요가 있다. 제한한 구조는 라우팅 로직을 사용하여 메시지가 정확한 위치에 저장되도록 한다. 메시지 라우팅 로직은 연산모듈로부터 출력된 메시지가 Tanner 그래프 상에서 에지로 연결된 연산모듈의 로컬 메모리에 저장될 수 있도록 메시지의 경로를 정해준다. 각 연산모듈은 자신의 로컬 메모리에서 메시지를 읽고 Tanner 그래프 상에서 에지로 연결된 연산모듈의 로컬 메모리에 메시지를 저장하므로, 제한한 복호기 구조에는 메모리 충돌이 없다.

그림 5는 부호율  $\frac{1}{2}$  일 때, 비트노드 그룹과 체크노드 그룹으로 구성된 Tanner 그래프이고 각 노드 그룹 안의 숫자는 노드 그룹을 구성하는 노드의 인덱스이다. 그림 5에서 노드 그룹 사이의 에지는 내부 노드 사이의 에지의 집합과 같다. 즉 하나의 노드 그룹은 내부 노드와 연결된 노드를 포함하는 노드 그룹과만 연결된다. 비트노드 그룹과 체크노드 그룹은 각각 정렬된 인덱스를 가지는 360개의 노드로 구성되기 때문에, 비트노드 그룹과 체크노드 그룹의 연결은 각 노드 그룹 내부의 비트노드와 체크노드의 순차적 연결과 같다. 따라서 메시지 전달 알고리즘을

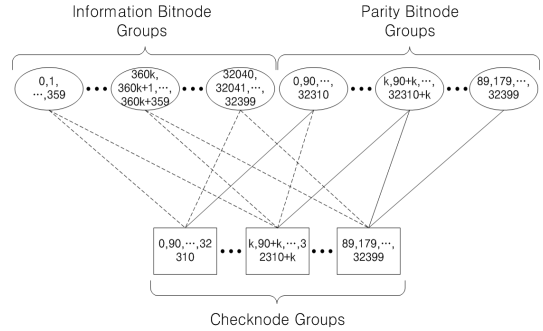


그림 5. 노드 그룹을 사용한 Tanner 그래프

사용한 복호 과정에서 하나의 노드 그룹에서 전송하는 메시지는 다른 하나의 노드 그룹으로만 전달되며, 노드 그룹 사이의 메시지 전달은 내부 노드 사이의 순차적 메시지 전달과 같다.

연산모듈이 전달하는 메시지는 에지로 연결된 연산모듈의 로컬 메모리에 저장된다. 메시지를 상대 연산모듈의 로컬 메모리로 전달하는 과정은 메시지 라우팅 로직이 수행한다. 하나의 연산모듈에 할당된 360개 노드에서 출력되는 모든 메시지는 같은 연산모듈로만 전달되므로 에지로 연결된 두 연산모듈 사이의 메시지 전달은 연산모듈과 로컬 메모리 사이의 메시지 전달 경로로 구현된다. 메시지 라우팅 로직은 H 행렬로부터 추출한 연산모듈 사이의 연결 관계를 바탕으로 모든 연산모듈의 출력으로부터 상대 연산모듈의 로컬 메모리까지 메시지 전달 경로를 각각 할당한다. 연산모듈에서 출력되는 메시지는 할당된 메시지 전달 경로를 통해 상대 연산모듈의 로컬 메모리에 저장된다. 연산모듈은 전달받은 메시지를 로컬 메모리로부터 읽어 에지로 연결된 연산모듈로 전달하는 메시지를 계산한다. 로컬 메모리의 첫 주소에는 연산모듈의 첫 노드가 필요로 하는 메시지가 아닌, 메시지를 전달한 연산모듈의 첫 노드로부터의 전달된 메시지가 저장되어 있다. 연산모듈이 첫 번째로 읽어야 하는 메시지는 H 행렬에 의해 초기값으로 정의된 인덱스를 가지는 노드로부터 전달받은 메시지이다. 따라서 각 로컬 메모리의 메모리 컨트롤러는 해당 연산모듈이 첫 번째로 읽어야 할 메모리 주소를 가지고 있어, 연산모듈이 초기 주소로부터 순차적으로 메시지를 읽도록 한다.

본 논문에서 제안한 메모리 접근 패턴을 부호율  $\frac{1}{2}$ 의 예를 통해 그림 6에 보인다. 그림 6의 BFM은 비트노드 연산모듈이고 CFM은 체크노드 연산모듈이

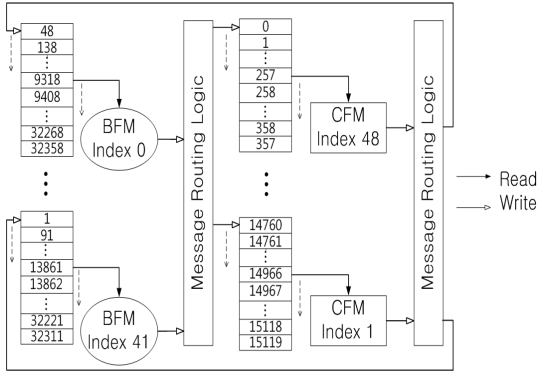


그림 6. 메모리 접근 패턴

며, 연산모듈 안의 숫자는 연산모듈의 인덱스를 나타낸다. 메모리 안의 숫자는 해당 메모리에 저장된 메시지를 전달한 노드의 인덱스이다. 인덱스 0의 비트노드 연산모듈은 인덱스 48의 체크노드 연산모듈과 연결된다. 연산모듈에서 출력된 메시지는 메시지 라우팅 로직을 거쳐 상대 연산모듈의 로컬 메모리의 첫 주소부터 저장되므로, 인덱스 0의 비트노드 연산모듈의 로컬 메모리에는 인덱스 48의 체크노드 연산모듈에서 보낸 메시지가 순차적으로 저장된다. 인덱스 0의 비트노드 연산모듈이 읽어야 할 첫 번째 메시지는 인덱스 9318의 체크노드에서 보낸 메시지이므로 해당 메시지가 저장되어 있는 주소에서 첫 번째 메시지를 읽고, 이후로는 1씩 증가된 주소에서 메시지를 읽는다.

### 3.3 연산모듈

Tanner 그래프의 비트노드는 비트노드 연산모듈로 구현하고, 체크노드는 체크노드 연산모듈로 구현한다. 본 논문에서는 비트노드 그룹과 체크노드 그룹을 각각 비트노드 연산모듈과 체크노드 연산모듈로 구현한다.

DVB-S2 LDPC 부호는 비균일 부호로서 부호율  $\frac{1}{2}$ 의 경우, 비트노드의 웨이트는 8, 3, 2이고 체크노드의 웨이트는 7이다. 그림 7은 웨이트가 8인 비트노드를 구현한 비트노드 연산모듈의 구조이다. 64,800개의 비트노드를 총 180개의 비트노드 연산모듈로 구현하였으며, 각각의 비트노드 연산모듈은 360개의 비트노드에 대한 비트노드 연산을 순차적으로 수행한다. 비트노드 연산모듈은 체크노드로부터 받은 메시지인  $r_{ji}$ 와 채널 LLR인  $LLR(y_i)$ 을 모두 더하여 전송된 심볼을 판정한다. 또 모든 메시지를 더한 값으

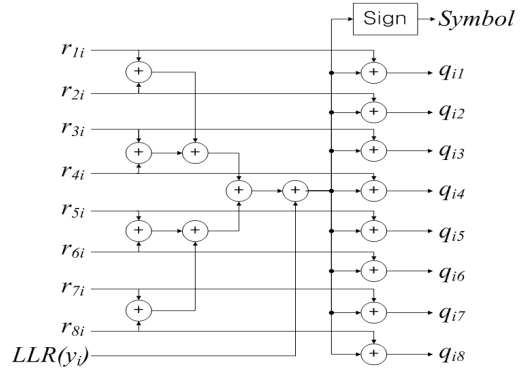


그림 7. 비트노드 연산모듈

로부터 체크노드에서 받은 메시지를 각각 빼주어 체크노드로 전달할 메시지를 계산한다. 비트노드 연산모듈은 16개의 덧셈기로 구성된 5 단계의 덧셈기 트리를 사용해 체크노드로 전달하는 메시지를 계산한다. 덧셈기 트리는 각 단계마다 레지스터를 배치한 파이프라인 구조를 가지고 있어 높은 클럭 주파수에서도 동작 가능하다. 메시지는 한 클럭 사이클 단위로 비트노드 연산모듈에 입력되므로, 모든 비트노드 연산은 latency를 포함해 총 365 클럭 사이클을 필요로 한다.

그림 8은 부호율  $\frac{1}{2}$ 에서의 체크노드 연산모듈을 나타낸다. 하나의 체크노드 연산모듈은 360개의 체크노드에 대한 체크노드 연산을 순차적으로 수행한다. 체크노드 연산은 메시지의 값을 계산하는 부분과 부호를 계산하는 부분으로 나뉜다. 메시지의 값을 계산하는 부분은 입력받은 메시지의 절대값만을 사용한다. 따라서 제안한 구조의 체크노드 연산모듈은 2의 보수로 표현된 메시지를 부호화된 크기 표현으로 변

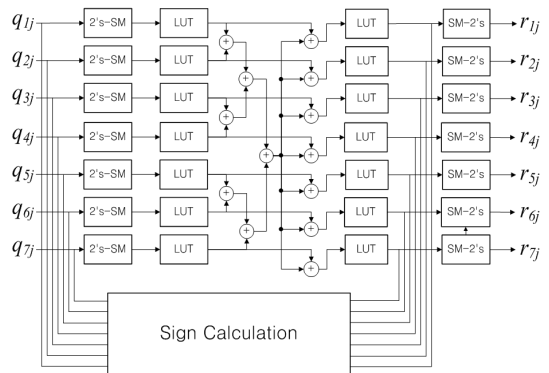


그림 8. 체크 연산모듈

환하여 부호 부분을 제외한 값 부분만을 사용한다. 체크노드 연산모듈 내부에서 사용하는 메시지의 길이를 1비트 줄여, 체크노드 연산에 필요한 LUT의 면적을 반으로 줄일 수 있고 덧셈기 입력의 길이도 줄일 수 있다. 체크노드 연산모듈은 덧셈 과정을 덧셈기 트리로 구현하고, 연산모듈의 각 구성 요소 사이에 레지스터를 배치한 파이프라인 구조를 가진다. 체크노드 연산모듈은 각 노드에 대한 입력을 한 클럭 사이클 마다 읽도록 구현되어, 체크노드 연산모듈 내부의 모든 체크노드 연산을 수행하는데 총 367 클럭 사이클이 소요된다.

### 3.4 제안한 복호기의 전체적인 구조

그림 9는 제안한 복호기의 블록 다이어그램을 보인다. 메모리 안의 괄호로 표현된 숫자는 메모리를 구성하는 메모리 뱅크의 수이다. 각 연산모듈의 로컬 메모리는 연산모듈의 입력 수만큼 메모리 뱅크로 구성되어, 연산모듈에 필요한 입력을 한 클럭 사이클에 제공한다. 부호율  $\frac{1}{2}$ 의 경우 체크노드 연산모듈은 각각 7개의 뱅크로 이루어진 로컬 메모리를 가지고 비트노드 연산모듈은 각각 8개와 3개, 그리고 2개의 뱅크로 이루어진 로컬 메모리를 각각 가진다. 그림 9에서 실선으로 표현된 화살표는 메시지의 흐름을 나타내고, 파선으로 표현된 화살표는 컨트롤 신호의 흐름을 나타낸다. 복호기의 복호 과정은 주 컨트롤러에 의해 제어된다. 주 컨트롤러는 입력 메모리 컨트롤러와 메시지 메모리 컨트롤러를 통해 메시지 전달 과정을 제어한다.

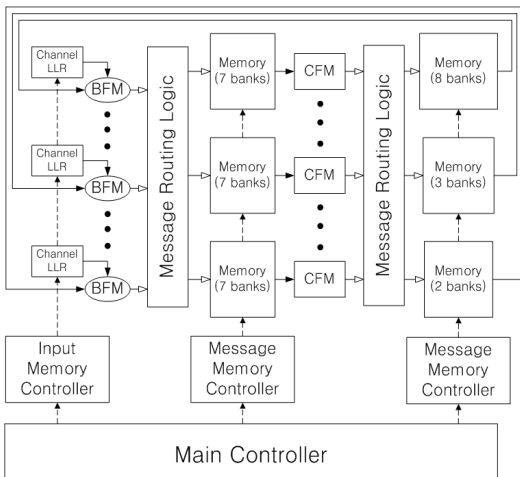


그림 9. 제안한 구조의 블록 다이어그램

복호 과정이 시작되면 주 컨트롤러는 각 메모리 컨트롤러의 메모리 주소값을 초기화한다. 초기화 과정으로 메모리 컨트롤러는  $H$  행렬에 의해 정해진 읽기 주소와 0 번지의 쓰기 주소를 가지고, 각 주소는 반복 복호 과정 중 1씩 증가한다. 따라서 연산모듈은  $H$  행렬에 의해 정해진 읽기 주소의 로컬 메모리로부터 메시지를 읽고 전달하는 메시지를 계산한다. 계산된 메시지는 메시지 라우팅 로직을 통해 에지로 연결된 연산모듈의 로컬 메모리에 0 번지부터 순차적으로 저장된다.

## IV. 구현 결과

제안한 구조의 복호기는 Synopsys 사의 Design Compiler를 사용해 TSMC 90nm 공정으로 합성하였다. 복호기는 메모리, 연산모듈, 메시지 라우팅 로직, 컨트롤러의 네 부분으로 나누어 각각의 면적을 표 1에 나타내었다. 복호기의 throughput은 식 (6)과 같이 구할 수 있다.

$$T = \frac{I}{\text{number of cycles}} \cdot f_{\text{cycle}} \quad (6)$$

$I$ 는 정보 블록의 길이이고  $\text{number of cycle}$ 은 LDPC 부호 한 프레임을 복호하는 데 소요되는 클럭 사이클이며  $f_{\text{cycle}}$ 은 클럭 주파수이다. 제안된 복호기는 한 번의 반복 과정에서 비트노드 연산과 체크노드 연산을 위해 각각 365 클럭 사이클과 367 클럭 사이클을 필요로 하며, 메모리 주소 초기화를 위해 1 클럭 사이클을 필요로 한다. 또 복호 시작 시 채널 LLR 값을 입력 받는데 360 클럭 사이클이 필요하다. 제안한 복호기의 최대 반복회수는 25이고, 300MHz에서 동작 가능하므로 제안된 복호기의 throughput은 520Mbps이다. 표 1에서 제안한 복호기 구조의 throughput을 기존 구조의 throughput과 함께 보인다.

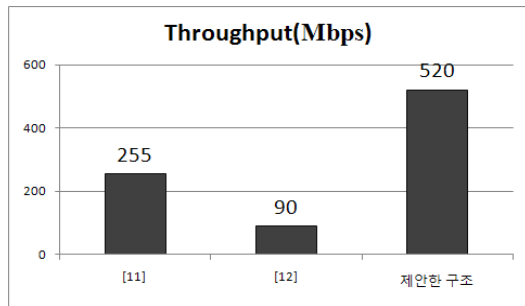
표 1은 제안한 복호기 구조의 구현 결과를 면적과 throughput 측면에서 기존 복호기 구조인 [11]과 [12]와 함께 보인다. 참고문헌 [11]에서 제안한 복호기 구조는 ST microelectronics의 0.13um 공정을 사용하여 합성하였으나, 90nm 공정에 맞게 스케일링 하였다[12]. 제안한 DVB-S2 LDPC 복호기 구조는 모든 연산모듈이 로컬 메모리를 가지는 구조로서, 비트노드와 체크노드 사이의 양방향 메시지 전달을 위한 메모리를 모두 가진다. 메모리 충돌을 고려할 필요가 없고 순차적인 메모리 접근이 가능하므로 throughput이 증가한

표 1. 기존 복호기 구조와 제안한 복호기 구조의 성능비교

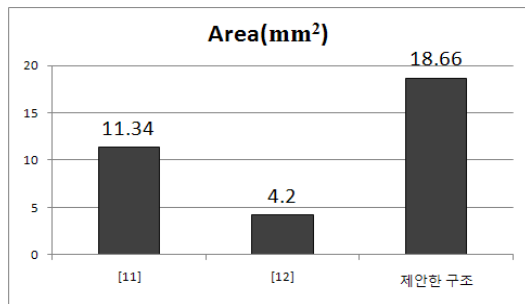
| Area(mm <sup>2</sup> ) |                   | 문헌 [11] | 문헌 [12] | 제안한 구조 |
|------------------------|-------------------|---------|---------|--------|
| memory                 | input             | 1.0     | 1.8     | 1.0    |
|                        | message           | 4.5     | 1.2     | 12.2   |
|                        | controller        | 0.04    | 0.1     | 1.61   |
| function               | functional logic  | 5.4     | 0.8     | 3.59   |
|                        | controller        | 0.1     | 0.1     | 0.001  |
|                        | shuffling/routing | 0.3     | 0.2     | 0.26   |
| total                  |                   | 11.34   | 4.2     | 18.66  |
| Throughput(Mbps)       |                   | 255     | 90      | 520    |

다. 제안한 복호기의 구조와 참고문헌 [11][12]의 구조를 비교하여 그림 10과 그림 11에 보인다.

그림 10은 기존 복호기 구조와 제안한 복호기 구조의 성능을 나타낸다. 기존 복호기 구조인 참고문헌 [11]의 구조는 255Mbps의 throughput 가지고 참고문헌 [12]의 구조는 90Mbps의 throughput을 가지는데 비해 제안한 구조의 복호기는 520Mbps의 throughput을 가진다. 반면 모든 연산모듈이 각각의 로컬 메모리를 가지므로 면적이 증가한다. 참고문헌 [11]의 구조는 1



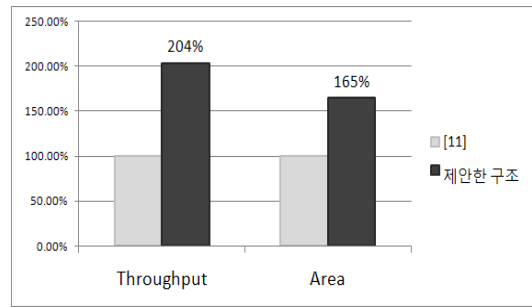
(a)



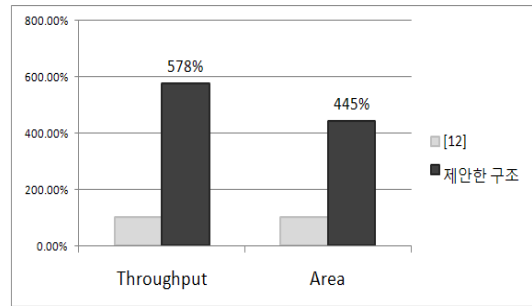
(b)

그림 10. 기존 복호기 구조와 제안한 복호기 구조의 성능 (a) Throughput, (b) 면적

1.34mm<sup>2</sup>의 면적을 가지고 참고문헌 [12]의 구조는 4.2mm<sup>2</sup>의 면적을 가지는데 비해 제안한 구조는 18.66mm<sup>2</sup>의 면적을 가진다. 그림 11은 기존 복호기 구조와 제안한 복호기 구조의 성능을 비율로 나타낸다. 제안한 구조의 복호기는 기존 복호기 구조인 [11]의 구조에 비해 104% 증가된 throughput 가지고 [12]의 구조에 비해 478% 증가된 throughput을 가진다. 반면 [11]의 구조에 비해 65% 증가된 면적을 가지고 [12]의 구조에 비해 345% 증가된 면적을 가진다.



(a)



(b)

그림 11. 기존 복호기 구조와 제안한 복호기 구조의 성능비교 (a) 참고문헌 [11]의 구조와 제안한 구조의 비교 (b) 참고문헌 [12]의 구조와 제안한 구조의 비교

## V. 결론

본 논문에서는 DVB-S2 LDPC 부호의 복호기 구조를 제안하였다. DVB-S2 LDPC 부호는 IRA 구조를 가진다. IRA 구조의 LDPC 부호가 가지는 특징을 이용해 Tanner 그래프 상의 비트노드와 체크노드를 비트노드 그룹과 체크노드 그룹으로 그룹핑하였다. 그룹핑 한 노드 그룹은 Tanner 그래프 상에서 그룹 내의 노드와 에지로 연결된 노드를 포함하는 노드 그룹과 연결되었다. 노드 그룹을 구현한 연산모듈은 각각 로컬 메모리를 가지며 전달받은 메시지는 로컬 메모리에서 순차적으로 읽도록 하였다. 제안한 구조



의 DVB-S2 LDPC 복호기는 로컬 메모리의 사용으로 메모리 충돌의 가능성을 없애고 메모리 접근 패턴을 단순화할 수 있어 throughput 성능을 향상시켰지만, 전체적인 면적이 증가하였다. 제안한 구조의 복호기는 참고문헌 [11]의 구조에 비해 약 104%, 참고문헌 [12]의 구조에 비해 약 478%의 throughput 성능 향상이 있었다. 반면에 참고문헌 [11]의 구조에 비해 약 65%, 참고문헌 [12]의 구조에 비해 약 345% 증가한 면적을 가진다.

본 논문에서는 IRA 구조를 가지는 DVB-S2 규격을 위한 LDPC 부호의 복호기 구조를 제안하였다. 제안한 구조의 핵심은 노드의 그룹핑과 로컬 메모리를 통한 메모리 접근 패턴의 단순화이다. 본 논문에서는 DVB-S2 LDPC 부호의 복호기 구조를 제안하였지만, IRA 구조의 LDPC 부호를 사용하는 다른 응용에도 적용 가능하다.

### 참 고 문 헌

- [1] ETSI, Digital video broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broad-band satellite applications: EN 302 307 V1. 1.1, 2005.
- [2] A. Morello and V. Mignone, "DVB-S2: The Second generation standard for satellite broad-band services," Proceedings of the IEEE, Vol.94, No.1, pp.210-227, Jan. 2006.
- [3] R. Gallager, "Low-density parity-check codes," IEEE Transactions on Information Theory, Vol.8, No.1, pp.21-28, Jan. 1962.
- [4] R. Tanner, "A recursive approach to low complexity codes," IEEE Transactions on Information Theory, Vol.27, No.5, pp.533-547, Sep. 1981.
- [5] D. MacKay, "Good error-correcting codes based on very sparse matrices," IEEE Transactions on Information Theory, Vol.45, No.2, pp.399-431, Mar. 1999.
- [6] F. Kschischang and B. Frey, "Iterative decoding of compound codes by probability propagation in graphical models," IEEE Journal on Selected Areas in Communications, Vol.16, No.2, pp.219-230, Feb. 1998.
- [7] T. Richardson and R. Urbanke, "Efficient encoding of low-density parity-check codes," IEEE Transactions on Information Theory, Vol.47, No.2, pp.638-656, Feb. 2001.
- [8] 배슬기, 김준성, 송홍엽, "순환 행렬과 eIRA 부호를 이용한 효율적인 LDPC 부호화기 설계", 한국통신학회 논문지, 제 31권, 2C호, pp.123-129, 2006년 2월.
- [9] M. Gomes, G. Falcao, V. Silva, V. Ferreira, A. Sengo, and M. Falcao, "Flexible parallel architecture for DVB-S2 LDPC decoders," in Proc. GLOBECOM. IEEE, Washington DC, pp.3265-3269, Nov. 2007.
- [10] H. Jin, A. Khandekar, and R. McEliece, "Irregular Repeat-Accumulate Codes," in Proc. 2nd Int. Symp. on Turbo Codes & Related Topics, Brest, France, pp.1-8, sep. 2000.
- [11] F. Kienle, T. Brack, and N. Wehn, "A synt hesizable IP core for DVB-S2 LDPC code decoding," in Proc. DATE, Munich, Germany, Vol.3, pp.100-105, Mar. 2005.
- [12] J. Dielissen, A. Hekstra and V. Berg, "Low cost LDPC decoder for DVB-S2," In Proc. DATE, Munich, Germany, Vol.2, pp.1-6, Mar. 2006.
- [13] E. Yeo and V. Anantharam, "Iterative decoder architectures," IEEE Communications Magazine, Vol.41, No.8, pp.132-140, Aug. 2003.
- [14] E. Yeo, P. Pakzad, B. Nikolic, and V. Anantharam, "High throughput low-density parity-check decoder architectures," in Proc. GLOBECOM. IEEE, San Antonio, TX, pp.3019-3024, Nov. 2001.
- [15] A. Blanksby and C. Howland, "A 690-mW 1-Gb/s 1024-b, rate-1/2 low-density parity-check code decoder," IEEE Journal of Solid-State Circuits, Vol.37, No.3, pp.404-412, Mar. 2002.

김 성 운 (Seong-Woon Kim)

정회원



2007년 2월 서강대학교 전자공학과 졸업  
2007년 3월~현재 서강대학교 전자공학과 대학원 CAD & ES 연구실 석사과정  
<관심분야> ECC decoder design for mobile, Low power VLSI design

황 선 영 (Sun-Young Hwang)

정회원



1976년 2월 서울대학교 전자공학과 졸업  
1978년 2월 한국 과학원 전기 및 전자공학과 공학석사 취득  
1986년 10월 미국 Stanford 대학 전자공학 박사학위 취득  
1976년~1981년 삼성 반도체 주식회사 연구원 팀장  
1986년~1989년 Stanford 대학 Center for Integrated System 연구소 책임 연구원  
1989년~1992년 삼성전자(주) 반도체 기술 자문  
2002년 4월~2004년 2월 서강대학교 정보통신대학원장  
1989년 3월~현재 서강대학교 전자공학과 교수  
<관심분야> CAD 및 임베디드 시스템 설계, NoC/SoC 시스템 설계, HW/SW co-design, DSP/VLSI 시스템 설계

박 창 수 (Chang-Soo Park)

정회원



2002년 2월 서강대학교 전자공학과 졸업  
2004년 8월 서강대학교 전자공학 석사학위 취득  
2005년 8월~현재 서강대학교 전자공학과 대학원 CAD & ES 연구실 박사과정

<관심분야> SoC design, Embedded System Design