

# 고속 UWB 시스템의 LDPC 디코더 구조 설계

정회원 최성우\*, 종신회원 이우용\*, 정회원 정현규\*

## LDPC Decoder Architecture for High-speed UWB System

Sung-Woo Choi\* *Regular Member*, Woo-Yong Lee\* *Lifelong Member*,  
Hyun-Kyu Chung\* *Regular Member*

### 요약

본 논문은 대표적인 고속 UWB 시스템인 MB-OFDM UWB 시스템에서 데이터 전송율을 올리면서 복호 성능을 보장하기 위해서 제안하고 있는 LDPC 부호에 대한 연구 결과를 보인다. 하드웨어 효율적인 복호기의 구조를 제안하기 위해서 LLR(log likelihood ratio) 계산 알고리즘과 체크노드 갱신 알고리즘을 시뮬레이션 하여 효율적인 방법을 선택 하였고, LDPC 디코딩 알고리즘의 반복 횟수를 결정하였다. 그리고 본 논문은 LDPC 디코더의 UWB 응용에 필요한 요구사항을 만족시키기 위한 LDPC 부호의 구조를 제시하였다. 이 구조는 FPGA를 통하여 합성되어 구현성을 검토하였으며, 기존 QC-LDPC 부호의 FPGA 합성 결과와 비교하여 높은 throughput을 제공함을 확인하였다. 이 구조를 이용하면 BP 알고리즘에 비해서 약 0.2dB의 성능열화를 포함하지만, 고속 데이터 전송에 적합한 LDPC 부호기를 구현할 수 있다.

**Key Words** : UWB, LDPC, Min-sum, Layered

### ABSTRACT

MB-OFDM UWB system will adopt LDPC codes to enhance the decoding performance with higher data rates. In this paper, we will consider algorithm and architecture of the LDPC codes in MB-OFDM UWB system. To suggest the hardware efficient LDPC decoder architecture, LLR(log-likelihood-ratio) calculation algorithms and check node update algorithms are analyzed. And we proposed the architecture of LDPC decoder for the high throughput application of Wimedia UWB. We estimated the feasibility of the proposed architecture by implementation in a FPGA. The implementation results show our architecture attains higher throughput than other result of QC-LDPC case. Using this architecture, we can implement LDPC decoder for high throughput transmission, but it is 0.2dB inferior to the BP algorithm.

### 1. 서론

MB-OFDM(Multi-band Orthogonal Frequency Division Multiplexing) UWB(Ultra wide band) 시스템은 3.1-10.6 GHz 대역을 사용하고, 53.3-480 M bps의 전송 속도를 가지는 WPAN 시스템이다<sup>[1]</sup>. 서비스 반경은 53.3 Mbps 경우 10 m 내외이며, 480 Mbps 인 경우는 1 m 내외 이다. 현재, 기존의 UWB 시스템

의 전송 용량을 1G bps 급으로 확대하는 표준안을 진행 중에 있다. 전송 용량을 확장하기 위하여 기존의 컨벌루션 코딩보다 복호 성능이 우수한 LDPC 부호를 채널코딩으로 사용할 것이 제안 되었다. LDPC 부호는 터보 부호와 더불어 성능 면에서는 최적의 부호라 할 수 있다. LDPC 부호는 병렬처리의 제한이 크지 않아서 데이터의 고속 처리에 유리한 장점이 있다. 현재, LDPC 부호는 IEEE 802.11n, IEEE 802.16e 등에서

※ 본 연구는 지식경제부 및 한국산업기술평가위원회의 IT산업원천기술개발사업 [2009-F-047-01, 개방형 mmWave 무선 인터페이스 플랫폼 기술 개발]의 일환으로 수행하였음.

\* 한국전자통신연구원 인터넷연구부 무선통신연구부 (csw9908, wylee, hkchung@etri.re.kr)

논문번호 : KICS2009-06-249, 접수일자 : 2009년 6월 17일, 최종논문접수일자 : 2010년 3월 5일

이미 사용되고 있으며, 고속 병렬처리 기술이 발전하면서 이동 및 무선 통신영역에서 각광 받고 있다. 본 논문은 고속 UWB 시스템의 채널 부호로써 사용될 LDPC에 대하여 전반적인 연구 결과를 보인다. 먼저, 고속 UWB 시스템에 적용될 LDPC 기술의 요구사항을 분석하고, 하드웨어에 영향을 주는 알고리즘 구조를 고찰한다. 그리고 높은 데이터 전송율을 위한 LDPC 디코더의 구조를 제시하고, 시뮬레이션을 통한 복호성능과 하드웨어 합성 결과를 보인다.

## II. MB-OFDM UWB system

### 2.1 시스템 개요

기존 MB-OFDM UWB 시스템은 3.1-10.6 GHz 주파수 대역을 대역폭이 528MHz 인 14개의 부대역으로 나누어서 사용한다. 물리계층의 전송방식으로는 OFDM을 이용하며, 128 포인트 IFFT/FFT를 사용한다. 그 중 100개의 서브 캐리어는 QPSK 나 DCM(Dual Carrier Modulation)으로 맵핑되고, 22 개의 서브캐리어는 파이프릿 신호로 사용된다<sup>1,2</sup>. 최근 논의 되고 있는 새로운 표준은 이 시스템의 전송율을 1Gbps 급으로 확장하기 위한 것이다. 주요 내용은 데이터 부채널에는 16-QAM에 상응하는 변조 방식을 사용하여 데이터 비트 수를 늘리고, 이에 대한 성능을 보장하기 위해서 LDPC 부호와 같은 강력한 부호를 채택하는 것이다. 현재 논의되는 LDPC 부호의 길이는 OFDM 3개 심볼에 해당하는 길이이며, 한 OFDM 심볼은 400비트이므로, 한 코드워드는 1200 비트이다. 본 장은 논의 되고 있는 LDPC 부호에 특징을 알아보고, 구현을 위한 요구사항을 분석한다.

### 2.2 Quasi-cyclic 블록 LDPC 부호

LDPC 부호의 패리티 체크 행렬을  $H$  라 하면,  $H$  행렬의 크기가  $m \times n$  일 경우,  $n$ 은 코드워드의 길이이고,  $m$ 은 패리티 체크 행의 개수이다. LDPC 부호는 원래 정보에 패리티 비트를 추가하는 systematic 부호로써, 정보 비트 수  $k$ 는  $n - m$  이다. 근래,  $H$  행렬을 구조적으로 설계하여, LDPC 인코더와 디코더의 구조를 간단히 하려는 연구가 진행 되어 왔다. 그 방법 중 하나는 IEEE 802.11n과 IEEE 802.16e에서 채택한 구조인 quasi-cyclic 블록 LDPC 부호이다. Quasi-cyclic LDPC 부호의  $H$  행렬은 cyclic-shift 된 identity 행렬 또는 제로 행렬을 요소 행렬로 가진다. 제로 행렬을 포함한 블록 LDPC 부호를 irregular quasi-cyclic LDPC 부호라 한다. Irregular quasi-

cyclic LDPC 부호에서  $H$  행렬을 몇 개의 블록으로 나누고 선형 변환으로 인코딩을 수행하여 LDPC 인코더의 복잡성을 개선할 수 있는데, 이것이 quasi-cyclic 블록 LDPC 부호이다. Quasi-cyclic LDPC (QC-LDPC) 부호는  $H$  행렬을 나타내기 위해서 기본 행렬을 사용한다. 기본 행렬은  $m_b \times n_b$ 의 행렬로 표현되며, 기본 행렬의 원소는 cyclic-shift 된 identity 행렬의 shift 값을 표현한다. 본 논문의 LDPC 부호는  $30 \times 30$ 인 identity 행렬을 요소 행렬로 하고,  $n_b = 40$ 이다. 부호율에 따른 기본 행렬의 구조는 표1과 같다.

표 1. 부호율에 따른 LDPC 부호 구조

code rate	data rate	$m_b$	$n_b$	degree of check node
1/2	640 Mbps	20	40	6
5/8	800 Mbps	15	40	9
3/4	960 Mbps	10	40	12
4/5	1024 Mbps	8	40	15

### 2.3 LDPC 구현 요구사항 분석

LDPC 디코더의 구현 요구사항 중 LDPC 디코더의 throughput은 디코딩 알고리즘의 반복 횟수에 가장 큰 영향을 받는다. LDPC 디코딩은 belief propagation (BP) 또는 Message passing 으로 불리는 알고리즘 의 해서 이루어진다<sup>3</sup>. 이 알고리즘에서는 체크-비트 메시지와 비트-체크 메시지를 미리 정한 반복 횟수만큼 반복 계산하며 갱신한다. 일반적으로 BP 알고리즘에서 비트노드에서 메시지를 모두 계산하고, 체크노드에서 메시지를 계산하는 것을 2-phased 디코딩이라하고, 순차적으로 체크노드와 비트노드를 갱신해나가는 방법을 layered 디코딩이라 한다. 잘 알려진 바와 같이, layered 디코딩의 반복 횟수가 2-phased 디코딩에 비해 절반 정도로 줄어든다. 먼저 BP 알고리즘의 반복 횟수를 결정하기 위하여 시뮬레이션을 수행하였다. 그림 1은 2-phased 디코딩과 layered 디코딩의 반복횟수에 따른 복호 성능을 보여준다.

Layered 디코딩의 6회 반복이 Layered 디코딩 10회 반복보다 약 0.2dB의 성능 열화를 가진다. 여기서, 디코더의 요구사항을 분석하기 위해서 layered 디코딩을 사용한 partially parallel LDPC 디코더 구조를 이용하였다<sup>3</sup>. 병렬화를 위해서 체크노드 업데이트 블록은 요소행렬의 크기만큼  $z$  병렬로 가정하였고, 입력되는 비트-체크 메시지들은 체크노드 degree 에 따라서

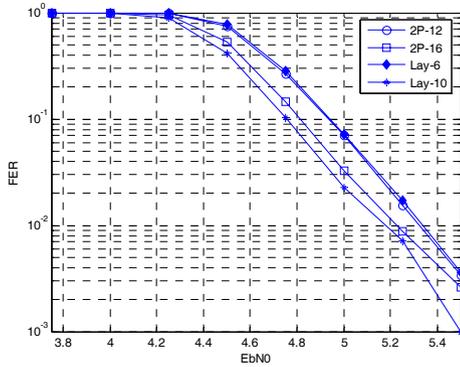


그림 1. 2-phased 디코딩과 layered 디코딩의 반복횟수에 따른 부호 성능 비교

serial로 입력되는 것으로 가정하였다. 그리고 layered BP 알고리즘의 반복횟수는 6회로 가정 하였으며, 행의 차수는 표1의 degree of check node 이다. 본 시스템은 매 3 개의 OFDM 심볼 동안 (937.5=3x312.5 ns) 코드워드의 복호 결과가 연어져야 한다. 상기 구조를 구현하기 위한 LDPC 디코더의 클럭 주파수를 표2에 나타냈다.

표 2를 보면 최대 864 MHz의 클럭이 요구된다. 이 클럭의 구현은 어렵기 때문에, 더욱 병렬화 방법을 도입해야 한다.

MB-OFDM UWB 시스템은 인터리버를 위하여 6 OFDM 심볼을 이용한다. 따라서 모뎀 수신부도 6심볼의 latency가 발생한다. 컨벌루션 코드를 사용한 경우, 비터비 디코더 자체의 latency는 trace back 방법을 이용하더라도 1 OFDM 심볼을 넘지 않는다. 일반적으로 LDPC 부호를 사용할 때는 인터리버/디인터리버를 사용하지 않으므로, LDPC 디코더의 latency가 6 OFDM 심볼과 같으면, 이전 규격과 호환성을 유지할 수 있으므로 상위 계층에서 제어가 용이하다. 본

표 2. Data rate에 따른 클럭 요구사항

Data rate (Mbps)	$m_b$	Required time for 1 clock (ns)	Rate (MHz)
640	20	$937.5/(6 \times 20 \times 6) = 1.302$	768
800	15	$937.5/(6 \times 15 \times 9) = 1.157$	864
960	10	$937.5/(6 \times 10 \times 12) = 0.163$	768
1024	8	$937.5/(6 \times 8 \times 15) = 0.163$	768
937.5/(BP 알고리즘 반복 횟수 x 행 개수 x 행 차수)			

LDPC 부호는 3 OFDM 심볼 블록 코디이므로, 디코더는 기본적으로 3 심볼의 latency를 가진다. 따라서 LDPC 디코더의 동작 latency가 3 OFDM 심볼 구간을 넘지 않아야 한다.

### III. LDPC 알고리즘 구조 설계

본 장에서는 LDPC 디코더의 하드웨어 구현 이슈에 있어서 중요한 영향을 끼치는 요소를 알아보고, 그것을 효율적인 하드웨어로 구현하는 방법을 제시하도록 한다.

#### 3.1 Log likelihood ratio 계산

LDPC 디코더는 LLR(log likelihood ratio)를 입력 받는다.  $k$  번째 심볼의  $i$  번째 비트  $y_{k,i}$ 가 LLR 결정 블록에 입력되었을 때, LLR 값은 아래 식과 같다.

$$A(y_{k,i}) = \ln \sum_{a_i \in A} e^{-\frac{(y_{k,i} - a_i)^2}{2\sigma^2}} - \ln \sum_{b_i \in B} e^{-\frac{(y_{k,i} - b_i)^2}{2\sigma^2}} \quad (1)$$

식(1)에서, A는  $y_{k,i} = 0$ 일 때 생성 가능한 좌표 값들의 집합이고, B는  $y_{k,i} = 1$ 일 때 생성 가능한 좌표 값들의 집합이다. 식(1)을 하드웨어로 설계하기 위해서는 많은 하드웨어가 소모 되므로, 이것을 해결하기 위한 sub-optimal LLR 기법들이 소개되었다[5]. 식(1)에  $\log(e^{\delta_1} + e^{\delta_2} + \dots + e^{\delta_n}) \approx \max_{i \in 1, \dots, n} \delta_i$ 을 적용하면 식 (1)을 간단하게 구할 수 있다. 이 방법을 Max-Log-MAP 알고리즘이라 한다. 그런데 이 근사화 과정에서 발생하는 에러를 줄이기 위해서 식 (2)와 같은 Jacobian logarithm을 이용할 수 있다.

$$\ln(e^{\delta_1} + e^{\delta_2}) = \max(\delta_1, \delta_2) + f_c(|\delta_1 - \delta_2|) \quad (2)$$

여기서,  $f_c(x)$ ,  $x > 0$  는 max 에 의한 에러를 보상하기 위한 것으로,  $\ln(1 + e^{-x})$  이다. 식 (2)를 초기 값으로 가정 하고,  $\delta = \ln(e^{\delta_1} + \dots + e^{\delta_{n-1}})$ 을  $n-1$  항까지의 계산 결과라 한다면,

$$\ln(e^{\delta_1} + \dots + e^{\delta_n}) = \max(\delta + \delta_n) + f_c(|\delta - \delta_n|) \quad (3)$$

을 이용하여  $n$ 항까지 계산할 수 있다. 이 방법을 Log-MAP 알고리즘이라 한다. Log-MAP 알고리즘의  $f_c(x)$ 는 미리 계산되어 저장 될 수 있다. 각 시스템은

complexity와 성능의 관계를 고려하여 LLR을 위한 알맞은 구조를 선택해야 한다.

### 3.2 체크노드 알고리즘

LDPC 코드를 복호하기 위한 알고리즘으로 메시지 BP(belief propagation) 알고리즘이 사용되고 있다. BP 알고리즘에서 비트노드와 체크노드 메시지 갱신을 위한 식은 다음과 같다.  $i$ -번째 iteration을 위하여 비트노드  $n$ 으로부터 체크노드  $m$ 으로 전달되는 메시지를  $Z_{mn}^{(i)}$ 라 하고, 체크노드  $m$ 으로부터 비트노드  $n$ 으로 전달되는 메시지를  $L_{mn}^{(i)}$ 라 하면, BP 알고리즘은 매 iteration 마다 다음 2단계를 거친다.

#### 3.2.1 체크노드 연산

$$L_{mn}^{(i)} = 2 \tanh^{-1} \left( \prod_{n' \in \mathcal{N}(m) \setminus n} \tanh \left( \frac{z_{mn'}^{(i-1)}}{2} \right) \right) \quad (4)$$

#### 3.2.2 비트노드 연산

$$z_{mn}^{(i)} = L_n^{(0)} + \sum_{m' \in \mathcal{M}(n) \setminus m} L_{m'n}^{(i)} \quad (5)$$

위 식 (4)에서  $\tanh$  함수는 테이블 형식으로 구해질 수 있다. 그러나 계산의 단순화를 위하여 BP-based 알고리즘 혹은 Min-Sum 알고리즘은 위 식 (4)를 아래와 같이 변경 한다<sup>[6]</sup>.

$$L_{mn}^{(i)} \approx \prod_{n' \in \mathcal{N}(m) \setminus n} \text{sgn}(z_{mn'}^{(i-1)}) \cdot \min_{n' \in \mathcal{N}(m) \setminus n} |z_{mn'}^{(i-1)}| \quad (6)$$

위 BP-based 알고리즘의 복호 성능 향상을 위하여 normalized BP-based 알고리즘이 사용되고 있다. 이 알고리즘은 식 (6)에 normalization factor를 곱한 형태이다.

$$L'_{mn}^{(i)} \leftarrow \alpha \cdot L_{mn}^{(i)} \quad (7)$$

식 (7)의  $\alpha$ 는 SNR 과 iteration 마다 달라질 수 있다. 그러나 간단한 구현을 위하여 일반적으로 한 가지 상수로 결정한다.

### 3.3 Layered 디코딩

II.3 절에서 이미 언급 한 바와 같이, Layered 디코딩 구조는 2 phased 디코딩에 비하여 디코딩 알고리즘 반복횟수가 약 1/2 줄어든다. Layered 디코딩에서

는 상기 수식 (4), (6)이 다음과 같이 세단계로 변경된다.

$$z_{mn}^{(i-1)} = z_n^{(i-1)} - L_{mn}^{(i-1)} \quad (8)$$

$$L_{mn}^{(i)} \approx \prod_{n' \in \mathcal{N}(m) \setminus n} \text{sgn}(z_{mn'}^{(i-1)}) \cdot \min_{n' \in \mathcal{N}(m) \setminus n} |z_{mn'}^{(i-1)}| \quad (9)$$

$$z_n^{(i)} = z_{mn}^{(i)} - \alpha \cdot L_{mn}^{(i)} \quad (10)$$

Layered 디코딩에서 식(8)~(10)을 계산하는 것을 sub-iteration이라 정의하고, QC-LDPC 부호에서 한 iteration은  $m_b$ (H행렬의 행 개수)개의 sub-iteration으로 구성된다.

### 3.4 제안된 알고리즘 구조

본 논문은 BP 알고리즘과 최대한 근사한 성능을 보이면서, 하드웨어 효율적인 알고리즘을 찾아내고자 하였다. 그림 2를 보면, BP 알고리즘과 normalized BP-based 알고리즘의 normalized factor에 따른 성능을 비교한다.

FER=0.08 인 지점을 살펴 볼 때, 결과적으로 BP 알고리즘이 normalized BP 알고리즘에 비하여 0.2 dB 정도 우수한 성능을 보이고 있음을 알 수 있고, 그림에서 normalized factor가 0.85 이거나 0.75 인 경우 유사한 성능을 나타낸다.

본 논문에서 제안하고자 하는 구조는 체크노드 갱신을 위해서 normalized BP-based 알고리즘을 사용하고, LLR을 계산하기 위해서 Max-Log-MAP을 사용하는 것이다. Max-Log-MAP 알고리즘을 사용하면, 식 (1)은 다음과 같이 변경할 수 있다.

$$\begin{aligned} A(y_{k,i}) &= \max_{a_i \in A} \frac{1}{2\sigma^2} (y_{k,i} - a_i)^2 - \max_{b_i \in B} \frac{1}{2\sigma^2} (y_{k,i} - b_i)^2 \\ &= \frac{1}{2\sigma^2} \left[ \max_{a_i \in A} (y_{k,i} - a_i)^2 - \max_{b_i \in B} (y_{k,i} - b_i)^2 \right] \end{aligned} \quad (11)$$

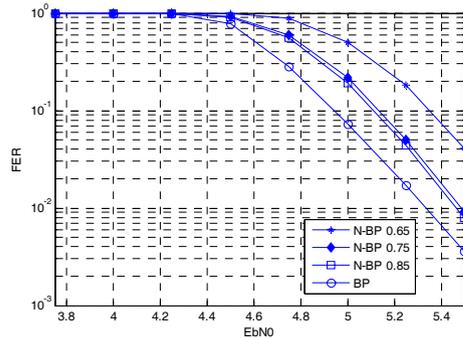


그림 2. BP 알고리즘과 normalized-BP 알고리즘의 복호 성능 비교

Normalized BP-based 알고리즘은 Minimum 값에 의존적이므로 입력되는 LLR의 상대적인 값에 의해서 결정된다. 그러므로 normalized BP-based 알고리즘을 사용할 때, 식 (11)의  $\sigma^2$ 이 생략 가능하다. 위 식 (11)을 이용하면, 채널에 대한 정보 없이 간단하게 LLR을 구할 수 있으므로, 그에 관한 하드웨어 복잡성을 제거할 수 있다. 그리고 체크노드 갱신을 위해서 normalized BP-based 알고리즘을 사용하므로, BP 알고리즘에 비하여 하드웨어는 많이 간단해지지만, min-sum 알고리즘에 비해서는  $\alpha$ 를 곱하기 위한 곱셈기가 추가된다. 여기서 본 구조는 시뮬레이션 결과로써 확인된  $\alpha = 0.75$ 를 사용한다.

$$\alpha \cdot L_{mn}^{(i)} = 0.75 \cdot L_{mn}^{(i)} = L_{mn}^{(i)} - 0.25 \cdot L_{mn}^{(i)} \quad (12)$$

식 (12)에서 0.25를 곱하는 것은 2비트 shift right 이므로, min-sum 알고리즘에 비교해서 감산기와 word length 가 2비트 추가된 레지스터만 하나 더 추가하면 normalized BP-based 알고리즘을 구현 할 수 있다. 그림 3은 제안된 체크노드 알고리즘을 사용한 경우, LLR 의 입력비트수가 6비트 일 때, 체크노드 메시지의 비트수를 결정하기 위해서 실험한 결과 이다. 9 비트일 때, 성능 열화가 거의 없음을 알 수 있다. 따라서 제안된 구조는 체크-비트 메시지와 비트-체크 메시지의 비트 수를 9로 결정하였다. 그림 4는 제안된 LDPC 디코더의 성능을 기존 알고리즘과 비교한 것으로써, 알고리즘 반복 횟수는 6회이다. 시뮬레이션 결과에 의하면, 반복 횟수가 많은 경우에 비해서 0.4dB 성능 열화가 있고, BP 알고리즘을 사용한 경우에 비해서는 0.2dB 정도의 성능 열화가 있다.

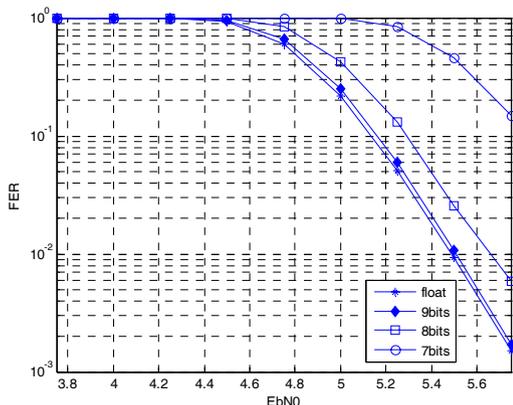


그림 3. 노트간 메시지 비트수에 따른 복호 성능 비교

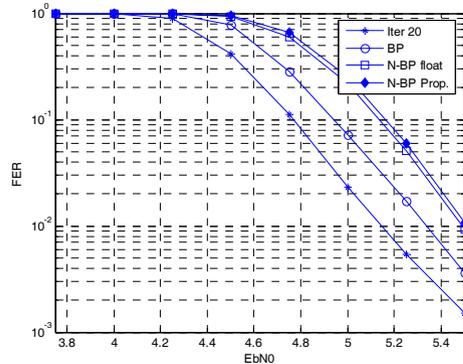


그림 4. 제안된 LDPC 디코더의 복호 성능 평가

#### IV. 제안된 LDPC 디코더 구조

4.1 Layer 기반 partially parallel LDPC 디코더  
 앞 II-3 장에서는 제안하는 LDPC 디코더의 기본 구조인 layer 기반 partially parallel LDPC 디코더의 구조를 일부 설명 하였다. 이 구조에서 체크노드 계산을 위한 입력이 시리얼 이면, 내부 구조는 간단하지만 동작 시간이 오래 걸리며, 입력을 모두 병렬로 받으면, 계산 시간은 적게 걸리지만 구조가 복잡해진다. 그리고 전 병렬 구조이면 irregular 한 체크노드 degree 일 때, 비효율적이다. 따라서 본 논문은 체크노드 블록의 입력 메시지를 3 병렬로 하는 layer 기반 partially parallel LDPC 디코더를 제안 한다. 제안하는 LDPC 디코더의 블록도는 그림 5와 같다.

입력되는 코드워드는 1200 비트이며, 요소 행렬의 크기는  $30 \times 30$  이다. 제안하는 디코더의 체크노드 블록은 요소행렬 내부 30행의 체크노드 연산을 동시에 수행하도록 30개의 체크노드 연산부로 구성된다. 제안하는 디코더의 체크노드 블록은 요소행렬 내부

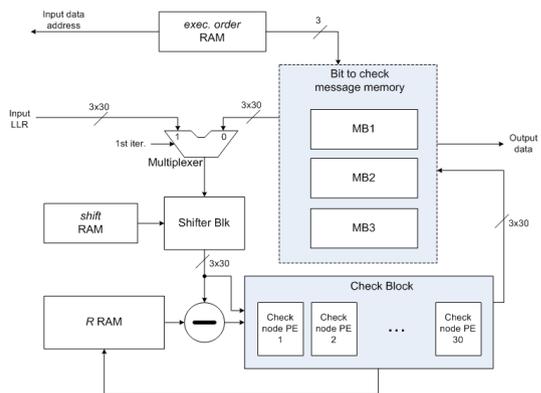


그림 5. 제안하는 LDPC 디코더 구조

30행의 체크노드 연산을 동시에 수행하도록 30개의 체크노드 연산부로 구성된다.(중복삭제요망) 본 디코더는 3 병렬 메시지 입출력 방식이므로, 체크노드 블록, 메모리 블록, 시프터 블록은 3x30 병렬로 데이터를 교환한다. 세 개의 메모리 뱅크(MB1, MB2, MB3)는 비트-체크 메시지를 저장하기 위한 공간이며, H 행렬의 40개 컬럼을 세 개로 나누어서 관리한다. MB1은 14개 컬럼을 저장하고, MB2와 MB3는 13개 컬럼을 저장한다. 따라서 각 MB의 주소는 4비트로 관리할 수 있다. 메모리 뱅크(MB1, MB2, MB3)의 주소는 exec\_order RAM에 병렬로 저장되어 있으며, execution\_order에 따라서 출력된다. Execution\_order는 BP 알고리즘의 한 iteration 구간에서 동작하는 행의 순서를 결정하고, 같은 행에서는 체크노드 업데이트 블록으로 입력되는 비트노드 메시지의 순서를 결정하는 값이다. 본 시스템의 excution\_order는 40이며, 다음 식을 통해 계산한다.

$$40 = (\text{deg. of node}) \times (m_b) / (\text{병렬수}) = 6 \times 20 / 3 = 40 \quad (13)$$

여기서,  $m_b$ 는 H 행렬의 행의 크기이다. 따라서 execution RAM의 크기는 40x12이며, 추가로 새로운 데이터 rate를 지원하기 위해서는 이 RAM에 새로운 execution\_order를 추가해야 한다.

시프터 블록은 3개의 30x30 barallel 시프터로 구성된다. Shift RAM은 execution\_order만큼 시프터 블록을 위한 cyclic shift 값 3개를 병렬로 저장한 것이다. 이 시프트 값은 해당 컬럼의 이전 시프트 값과 현재 필요한 시프트 값과의 차이를 저장한다.

R RAM 블록은 모든 체크-비트 메시지 값을 execution\_order만큼 저장한 것으로 가장 큰 부분을 차지한다. 체크-비트 메시지를 저장하는 메모리의 크기는 다음과 같다.

$$m_{bits} = (\text{메시지크기}) \times (\text{병렬수}) \times (\text{exec. order}) \times (\text{요소행렬크기}) = 9 \times 3 \times 40 \times 30 = 32,400 \quad (14)$$

#### 4.2 병렬 체크노드 연산 모듈

그림 6은 제안하는 체크노드 연산부의 구조를 나타낸다.

체크노드 연산부는 stage1~stage5로 구성된다. 각 stage는 3개의 체크-비트 메시지를 계산한다. 5 stage로 구성된 구조로써, 이 구조를 이용하여 최대 15개의 체크-비트 메시지를 계산할 수 있다. 만약 체크노드

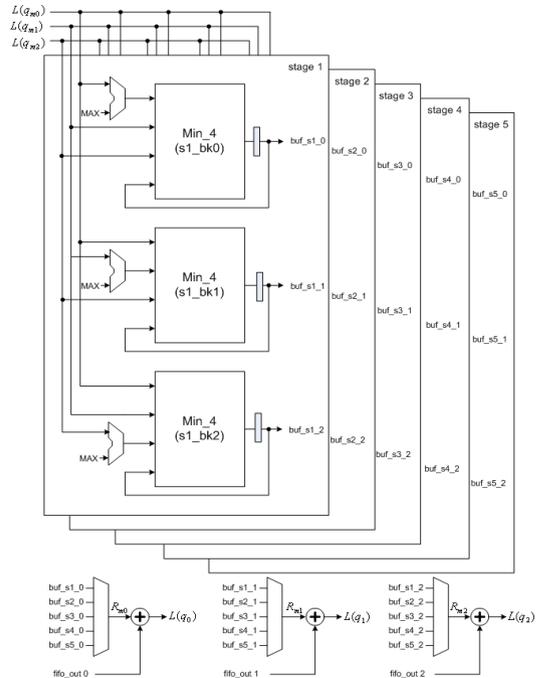


그림 6. 체크노드 프로세싱 블록 구조

드 degree가 증가하면, stage를 추가함으로써 디코더의 구조를 확장할 수 있다. 그림에서 Min\_4 블록은 4개의 메시지를 입력 받아서 normalized BP-based 알고리즘을 이용하여, 체크-비트 메시지를 계산하는 블록이다. Min\_4 블록 입력의 피드백 패스는 이전 클럭의 계산값이 현재 입력과 다시 계산되게 하기 위한 경로이다. 또한 Min\_4 블록의 한 입력은 멀티플렉서로 MAX 값을 선택할 수 있다. 이것을 선택하면, 식 (6)을 통하여 체크-비트 메시지를 계산할 때, 해당 비트노드에서 오는 메시지 값을 minimum 계산 결과에 반영하지 않도록 한다. 제안된 LDPC 디코더는 체크노드 계산이 3배 빠르게 진행되므로, 표 2의 시리얼 방식에 비해 1/3배의 동작 클럭으로 구현 할 수 있다.

### V. 시뮬레이션 및 FPGA 합성 결과

#### 5.1 LDPC 디코더 성능

제안된 LDPC 디코더의 UWB 채널 하에서의 동작을 검증하기 위하여 시뮬레이션을 수행하였다. 시뮬레이션을 위해서 페이로드 데이터는 1000 bytes로 하였으며, 각 EbN0 당 데이터 프레임을 30,000 번 전송하였다. UWB 채널 모델을 사용한 경우 각 모델 별로 정해진 100개의 프로파일들이 같은 횟수 선택 되도록 하였다. 시뮬레이션에는 OFDM 심볼 동기화 채널

estimation을 포함시켰으며, 주파수 오프셋은 고려하지 않았다. 이를 위하여 IEEE 802.15.3a의 채널 모델을 사용하였다<sup>[4]</sup>. 채널 모델 1~4의 특징은 다음 표와 같다.

데이터 전송율이 640Mbps 일 때, CM1, CM2, CM3 환경에서의 FER은 다음 그림 7과 같다.

규격에서는 FER이 8%이하 일 때, 링크 성공으로 본다. CM1 환경에서 필요한 EbN0는 10.5dB 이며, CM2 환경에서는 11dB 이지만, CM3 이상 열악한 환경에서는 LDPC 코드를 사용하더라도 전송이 될 수 없음을 보여준다.

표 3. UWB 채널 모델 특징

채널 특징	CM1	CM2	CM3	CM4
Mean access delay (nsec)	4.9	9.4	13.8	26.8
RMS delay (nsec)	5	8	14	26
채널환경 (서비스 반경)	LOS (0-4m)	NLOS (0-4m)	NLOS (4-10m)	NLOS (>10m)

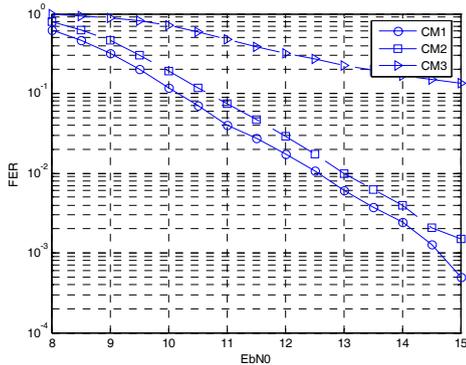


그림 7. UWB 채널 환경에서의 제안된 디코더 성능

### 5.2 하드웨어 합성 결과

제안된 LDPC 디코더의 구현성을 검토하기 위하여 제안된 구조를 verilog-HDL을 사용하여 코딩하고, Xilinx FPGA(Xilinx 4Vfx140-11ff1517)로 합성 하였다. FPGA 합성결과는 표 4와 같다.

제안된 디코더와 비교를 위하여, FPGA 합성 결과를 가지는 논문들 중에서 QC-LDPC 부호를 사용하면 같은 코드워드 길이를 가지는 논문을 선택하였다. 제안된 디코더가 throughput 측면에서 가장 좋은 결과를 보임을 알 수 있다. 본 LDPC 디코더의 하드웨어 복잡성의 증가는 병렬화 정도와 메시지 비트의 크기에 기인한다. 논문 [2]의 경우 시리얼 체크노드 입력

표 4. 하드웨어 합성 결과

	Proposed	ICUWB <sup>[2]</sup>
slices	53069	5996
LUT	101155	9497
SFF	11441	4310
BRAM	66	13
Memory (bits)	50890	
Clock (MHz)	61	
Total area (%)		
Throughput (Mbps)	152~228	100

구조로써, 본 구조보다 훨씬 적은 routing slices 와 LUT를 사용한다. 본 구조는 3-parallel 체크노드를 사용하였으며, 메시지 비트 크기도 9비트로써 다른 일반적인 예보다 많이 사용한다. 본 LDPC 디코더의 동작 클럭 61MHz는 FPGA 결과이므로, 현재 ASIC 공정으로 구현되면, 약 200MHz 이상 구현이 가능할 것으로 예상된다. 결과 비교를 위해서, 시리얼 체크노드를 기준으로 계산했던 표 2의 클럭 요구사항을 3 병렬 구조 기준으로 변경해야 한다. 제안된 디코더의 예상 동작클럭은 표 2의 동작클럭×1/3을 거의 만족 시킬 수 있다.

## VI. 결론

본 논문은 대표적인 고속 WPAN 시스템인 MB-OFDM UWB 시스템에 1Gbps 급 전송속도를 지원하는 과정에서 채택된 LDPC 디코더의 효율적인 구조를 제시하고자 하였다. 1Gbps급 LDPC 복호기의 요구사항을 분석한 결과 고속 동작을 위한 병렬처리가 많이 요구됨을 확인하였다. 본 논문은 병렬화를 대비하여 LLR 계산 블록과 체크노드 블록의 간단한 알고리즘 구조를 제시하였다. 제시한 알고리즘을 이용하면, 이상적인 BP 알고리즘에 비해서 0.2dB의 성능 열화가 발생한다. 그리고 Wimedia UWB의 높은 throughput을 얻기 위해서, 3 병렬 메시지 입출력 방식의 partially parallel LDPC 디코더를 제안하였다. 제안된 구조는 FPGA 합성한 결과, 다른 FPGA 합성 결과 보다 우수한 throughput을 제공함을 확인하였다. 향후 제안된 구조를 ASIC으로 제작한다면, 1Gbps급 MB-OFDM UWB의 데이터 전송률을 제공 할 수 있을 것이다.

참 고 문 헌

- [1] W. Abbott et al., "Multiband OFDM physical layer specification version 1.2 (draft)," *WiMedia Alliance*, Feb. 2007.
- [2] T. Lehnigk-Emden, C. Brehm, T. Brack, N. Wehn, "Energy Consumption of Channel Decoders for OFDM-based UWB Systems," in *Proc. IEEE International Conference on Ultra-Wideband*, 24-26 Sept. 2007, pp.447-452.
- [3] D. Hocoavar, "A reduced complexity decoder architecture via layered decoding of LDPC codes," in *Proc. IEEE Workshop on Signal Processing Systems (SiPS '04)*, Austin,USA, Oct. 2004, pp. 107 - 112.
- [4] Jeff Foerster, "Channel Modeling Sub-committed Report Final," *IEEE P802.15-02/368r5-SG3a*, Nov 2002.
- [5] P. Robertson, E. Vibbebrun, R. Hoehner, "A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain", in *Proc. IEEE Int. Conf. Commun., seattle*, 18-22 June 1995, pp.1009-1013
- [6] J. Chen and M. Fossorier, "Near optimum universal belief propagation based decoding of low-density parity check codes," *IEEE Trans. Commun.*, vol. 50, pp. 406 - 414, Mar. 2002.

이 우 용 (Woo-yong Lee)

중신회원



1989년 2월 고려대학교 전자공학과 학사  
 1991년 2월 한국과학기술원 전기 및 전자 공학과 석사  
 1997년 2월 한국과학기술원 전기 및 전자 공학과 박사  
 1997년 3월~현재 ETRI 초고속무선통신연구팀 팀장

<관심분야> 차세대 WPAN, 멀티기가 무선통신

정 현 규 (Hyun-Kyu Chung)

정회원



1985년 2월 서울대학교 전기 공학과 (학사)  
 1988년 2월 :KAIST 전기 및 전자 공학과(석사)  
 2002년 2월 Polytechnic University, Brooklyn, New Youk(박사)

1988년~1993년 한국통신 연구 개발단 전임연구원  
 1993년~1998년 SK 텔레콤 중앙연구소 책임연구원  
 2001년 Lucent Technologies, Whippany, New Jersey, MTS

2001년~현재 ETRI 무선통신연구부 부장

<관심분야> 이동통신, MIMO-OFDM, MMR, 무선 전송기술, Propagation Channel, MIMO 채널모델

최 성 우 (Sung-woo Choi)

정회원



1999년 2월 전북대학교 전자공학과 학사

2001년 2월 : 전북대학교 전자 공학과 석사

2001년 3월~현재 ETRI 초고속무선통신연구팀 선임연구원

<관심분야> 차세대 WPAN, 채널코딩, 신호처리 설계