

다중 블록길이를 지원하는 IEEE 802.11n LDPC 복호기 구조

나영현* · 신경욱*

*금오공과대학교 전자공학부

An Architecture for IEEE 802.11n LDPC Decoder Supporting Multi Block Lengths

Young-Heon Na*, Kyung-Wook Shin*

*School of Electronic Engineering, Kumoh National Institute of Technology

E-mail : yhna@kumoh.ac.kr

요 약

본 논문에서는 IEEE 802.11n 표준에 제시된 3가지 블록길이(648, 1,296, 1,944)를 지원하는 효율적인 LDPC (Low-Density Parity Check) 복호기 구조를 제안한다. LDPC 복호기의 핵심 블록인 DFU(Decoding Function Unit)의 연산 복잡도와 하드웨어 복잡도를 효율적으로 감소시킬 수 있도록 최소합 알고리즘과 블록직렬 방식의 layered 구조를 적용하였다. 또한 효율적인 다중 블록길이의 구현을 위해 PCM 값을 저장하는 H-ROM의 최적화 방법을 제안하였으며, 이를 통해 ROM의 크기를 약 42% 감소시켰다. 또한, 레이어 간의 효율적인 메모리 읽기/쓰기 방법을 적용하여 복호기 동작을 최적화시켰다.

ABSTRACT

This paper describes an efficient architecture for LDPC(Low-Density Parity Check) decoder, which supports three block lengths (648, 1,296, 1,944) of IEEE 802.11n standard. To minimize hardware complexity, the min-sum algorithm and block-serial layered structure are adopted in DFU(Decoding Function Unit) which is a main functional block in LDPC decoder. The optimized H-ROM structure for multi block lengths reduces the ROM size by 42% as compared to the conventional method. Also, pipelined memory read/write scheme for inter-layer DFU operations is proposed for an optimized operation of LDPC decoder.

키워드

LDPC(Low-Density Parity-Check), IEEE 802.11n, Wireless LAN, Min-sum algorithm

I. 서 론

LDPC 코드는 1962년 로버트 갤러거 (R. Gallager)의 MIT 박사학위 논문으로 제안된 채널부호화 기법이다^[1]. LDPC 부호는 복호화의 복잡도가 낮을 뿐만 아니라 우수한 거리 특성으로 오류마루 현상이 나타나지 않으며, 완전 병렬처리에 의한 고속연산 구현이 용이한 장점이 있다^[2]. LDPC 부호는 오류정정 및 검출능력을 나타내는 최소 거리가 부호의 블록길이에 비례하여 증가하는 특성이 있으며, 이것은 블록길이에 비례하여 복호 성능이 점진적으로 개선됨을 의미한다^[3].

본 논문에서는 최소합 알고리즘을 사용하여 IEEE 802.11n^[4]의 3가지 블록길이와 부호화율 1/2을 지원하는 LDPC 복호기 구조에 대해 연구하였다. 본 논문의 2장에서는 802.11n의 LDPC 복호 알고리즘에 대해서 기술하며, 3장에서는 효율적인 LDPC 복호기 구조를 위해 블록길이에 따라 가변적으로 동작하는 DFU 배열의 구조, 최적화된 H-ROM 구조, 그리고 Permuter의 효율적인 구현방법을 기술하였다. Verilog HDL로 설계된 회로를 Modelsim을 이용하여 기능검증을 하였다.

II. LDPC 복호 알고리즘

IEEE 802.11n 무선 랜 표준에는^[4] 3가지의 블록길이 (1944, 1296, 648)와 각 블록길이에 대해 4가지의 부호화율 (1/2, 2/3, 3/4, 5/6)을 지원하여 총 12가지의 패리티 검사 행렬(Parity Check Matrix; PCM) 규정하고 있다. LDPC 부호의 복호는 PCM에 의해 구성되는 태너 그래프 상의 변수노드(variable node)와 검사노드(check node) 사이에서 반복적인 메시지 전달과정에 의해 이루어진다. 대표적인 LDPC 복호 알고리즘으로 합-곱 알고리즘(sum-product algorithm; SPA), SPA의 계산 복잡도를 감소시킨 LLR-SPA, 그리고 LLR-SPA를 근사화한 최소합 알고리즘(min-sum algorithm; MSA)^[5] 등이 있다.

본 논문에서 사용하는 최소합 알고리즘(MSA)^[5]은 LLR-SPA의 함수 $\phi(x)$ 는 x 가 작을 때 큰 값을 가지며, x 가 커지면 거의 0에 근접하는 특성을 근사화한 알고리즘이다. 식(1)에서 함수 $\phi(x)$ 의 값은 x 값이 매우 작을 때 가장 큰 영향을 주므로 $\sum \phi(X) \approx \phi(\min(X))$ 로 근사화될 수 있다. 함수 $\phi(x)$ 는 $x > 0$ 일 때 역함수와 본함수 값이 같으므로 $\phi(\phi(\min(X))) = \min(X)$ 로 변환

되어 식(2)와 같이 근사화될 수 있다.

$$L_{j \rightarrow i} = \left(\prod_{i' \in \omega(j) \setminus \{i\}} \alpha_{i'j} \right) \cdot \phi \left(\sum_{i' \in \omega(j) \setminus \{i\}} \phi(\beta_{i'j}) \right) \quad (1)$$

$$\phi \left(\sum_{i' \in \omega(j) \setminus \{i\}} \phi(\beta_{i'j}) \right) \approx \phi \left(\phi \left(\min_{j' \in \omega(j) \setminus \{i\}} \beta_{i'j} \right) \right) = \min_{j' \in \omega(j) \setminus \{i\}} \beta_{i'j} \quad (2)$$

따라서 식(1)은 식(2)에 의해 식(3)과 같이 근사화될 수 있으며 이를 최소합 알고리즘이라 한다. 반복복호 과정에서 식(3)과 식(4)에 의해 사후확률 값 z_i 가 식(5)와 같이 계산된다.

$$L'_{j \rightarrow i} = \left(\prod_{j' \in \omega(j) \setminus \{i\}} \alpha_{i'j} \right) \min_{j' \in \omega(j) \setminus \{i\}} \beta_{i'j} \quad (3)$$

$$L_{i \rightarrow j} = z_i - L'_{j \rightarrow i} \quad (4)$$

$$z'_i = L_{i \rightarrow j} + L'_{j \rightarrow i} \quad (5)$$

본 논문에서는 $\phi(x)$ 의 연산과정을 피할 수 있고, LUT를 사용하지 않기 때문에 연산 복잡도와 하드웨어 면적이 감소하는 장점이 있는 최소합 알고리즘을 사용하여 다중 블록길이를 지원하는 LDPC 복호기를 구조를 제안하였다.

III. LDPC 복호기 구조 및 기능검증

본 논문에서는 IEEE 802.11n 표준의 3가지 블록길이와 부호화율 1/2을 지원하는 LDPC 복호기 구조를 제안하였다.

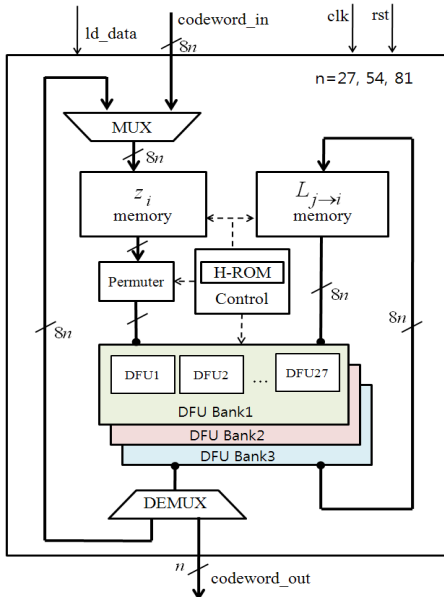


그림 1. 제안된 LDPC 복호기의 구조

제안된 LDPC 복호기의 전체 구조는 그림 1과 같으며,

복호연산을 수행하는 DFU(Decoding Function Unit), PCM을 저장하는 H-ROM, PCM의 오른쪽 순환시프트 정보에 따라 데이터를 분배하는 permuter, 반복복호 과정의 중간 결과 값을 저장하는 메모리, 그리고 전체 동작을 제어하는 제어블록으로 구성된다. LDPC 복호과정의 연산을 처리하는 DFU를 3개의 뱅크(bank) 구조로 구현하여 블록길이에 따라 27, 54, 81개의 DFU 배열이 선택적으로 동작하도록 하였다.

1. DFU 구조

DFU는 오류가 포함되어 있는 입력 코드워드의 오류를 정정하기 위해 코드워드들의 LLR을 취합하여 원래의 코드워드에 가장 가까운 LLR을 예측하는 연산을 수행한다. 본 논문에서는 최소합 알고리즘을 기반으로 한 layered 구조의 DFU를 설계하였으며, 설계된 DFU의 구조는 그림 2와 같다.

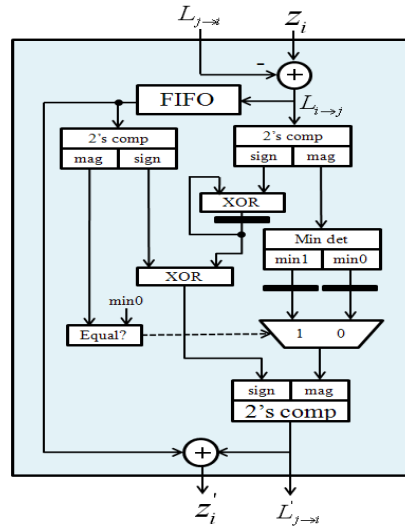


그림 2. DFU의 구조

DFU는 최소값 검출기, 부호비트 누산기, 데이터의 지연을 위한 FIFO(First-In First-Out) 메모리, 덧셈기, 뺄셈기, 비교기 그리고 수체계(2의 보수/부호-크기) 변환기 등으로 구성된다. 그림 3은 설계된 DFU의 기능검증 결과이다. DFU는 코드워드 8개의 읽기/쓰기에 8 클럭 사이클, 복호연산에 1 클럭 사이클 그리고 3 클럭 사이클의 stall을 필요로 하며, 따라서 하나의 레이어를 복호하기 위해 12 클럭 사이클이 소요된다. 그림 3은 k -번째 레이어의 복호 결과이며, 8개의 코드워드 z_i 와 검사노드 갱신값 $L_{j \rightarrow i}$ 를 입력받아 12 클럭 사이클 동안 복호하여 8개의 새로운 z'_i 와 $L'_{j \rightarrow i}$ 를 갱신한다. k -번째 레이어에서 $L_{i \rightarrow j}$ 가 FIFO에 저장되는 동안 검사노드는 변수노드 연산결과 $L_{i \rightarrow j}$ 를 받아 최소값(min0)과 준최소값(min1)을 검출한다. $(k+1)$ -번째 레이어에서는 FIFO에 저장된 k -번째 레이어의 $L_{i \rightarrow j}$ 값들과 비교하여 최소값이 검출된 위치에 $L_{j \rightarrow i}$ 를 준최소값으로 갱신하고, 나머지는 위치에는 최소값으로 $L_{j \rightarrow i}$ 를 갱신한다. z_i 는 최소

값과 준최소값으로 갱신된 $L'_{j \rightarrow i}$ 와 k -번째 레이어에서 FIFO에 저장된 $L_{i \rightarrow j}$ 값을 받아 $(k+1)$ -번째 레이어의 새로운 z'_i 로 갱신된다. 설계된 DFU의 $(k+1)$ -번째 레이어는 메모리에서 데이터를 읽어와 복호가 수행되는 동안 k -번째 레이어의 복호된 데이터를 메모리에 갱신하는 동작을 수행한다.

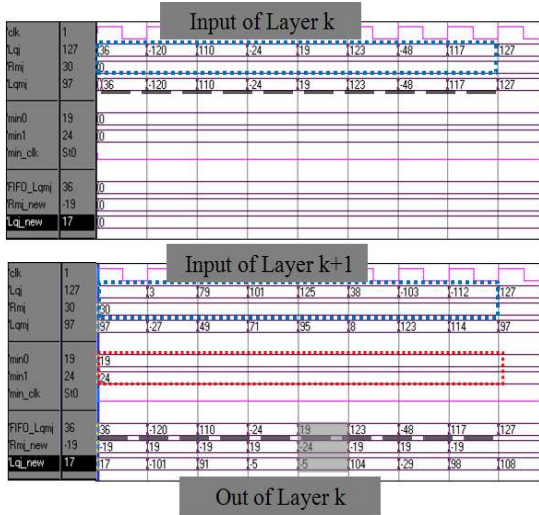


그림 3. DFU의 기능검증 결과

2. H-ROM 구조

H-ROM은 LDPC 복호화에 사용되는 PCM 값을 저장하는 메모리이다. IEEE 802.11n 표준의 PCM은 부호화를 1/2에 대해 3가지 블록길이(B=648, 1,296, 1,944)를 지원하며, 12개의 레이어와 레이어 당 24개의 부행렬을 가져, 총 288개의 부행렬로 구성된다. 부행렬의 크기는 블록길이에 따라 $Z_f \times Z_f$ (단, $f = 27, 54, 81$)가 된다. 총 288개의 부행렬 중에는 다수의 영(0) 행렬이 포함되어 있다. B=1,944비트와 1,296비트인 경우는 202개의 영행렬이 포함되어 있고, B=648비트인 경우는 200개의 영행렬이 포함되어 있다. 따라서 모든 행렬의 정보를 저장하는 대신에, 영 행렬을 제외한 부행렬들의 위치 정보와 순환 시프트 정보만을 H-ROM에 저장하는 방법을 고안함으로써 하드웨어가 최소화되도록 하였다.

IEEE 802.11n의 PCM을 구성하는 부행렬들은 블록길이 B에 따른 최대 시프트 거리가 25비트(B=648인 경우), 51비트(B=1,296인 경우), 79비트(B=1,944인 경우)이다. 시프트 정보를 저장하기 위해서는 블록길이에 따라 5~7비트의 정보가 필요하다. 본 논문에서는 영행렬을 제외한 7개의 부행렬과 영 행렬 1개를 추가하여 레이어를 8개의 부행렬로 구성하고, 그림 4(b)와 같이 총 96 (=12×8)개의 부행렬을 시프트 정보와 부행렬 위치로 식별하도록 하였다. 부행렬의 위치 정보를 나타내기 위해 0~23번지가 필요하므로 5비트의 표현이 필요하다. 하나의 부행렬을 ROM에 저장하기 위해서는 총 12비트가 필요하며, 전체 PCM을 저장하기 위해서는 3,456비트(=3×12×96)의 ROM 용량이 필요하다. 이와 같은 본 논문의 부행렬 저장방법은 그림 4(a)와 같이 모든

부행렬의 시프트 정보를 모두 저장하는 방법의 ROM 용량 6,048비트(=3×7×288)에 비해 약 42%의 ROM 용량을 감소시킬 수 있다.

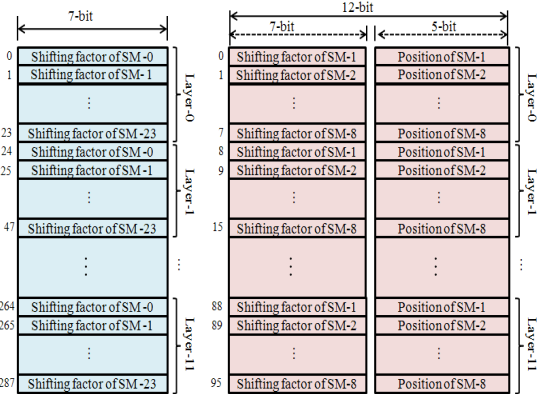


그림 4. PCM 값 저장 방법

3. Permuter 구조

Permuter는 PCM에서 제공되는 항등행렬의 오른쪽 순환 시프트를 구현하는 블록이다. 본 논문에서는 데이터를 임의의 비트만큼 오른쪽 순환 시프트(Cyclic Right Shift; CRS)시키는 회로를 그림 5와 같이 구현하였다.

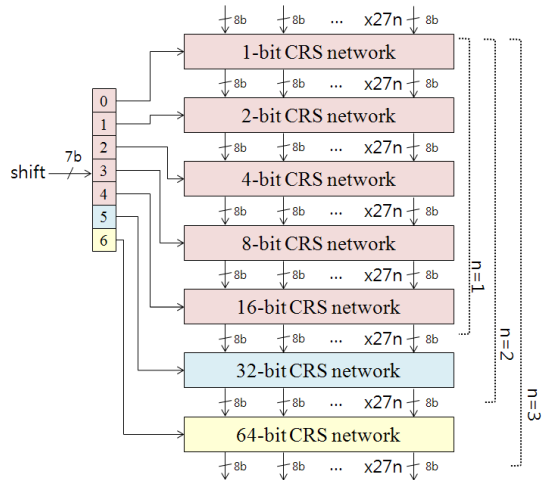


그림 5. Permuter의 구조

결정변수 메모리(z_i memory)에 저장된 데이터는 다음 레이어의 연산을 위해 CRS 회로를 통해 오른쪽 순환 시프트되고, 그 결과가 DFU에서 연산처리 된다. IEEE 802.11n의 부호화율 1/2에 대한 블록길이 648, 1,296, 1,944비트를 지원하는 PCM은 CRS의 크기가 0에서부터 최대 79인 부행렬들로 구성된다. 따라서 79 이하의 모든 수를 표현하기 위해 Permuter는 1비트(2^0)에서 64비트(2^6)의 7가지의 CRS 동작이 구현되어야 한다. 본 논문의 Permuter는 부행렬의 CRS 크기에 따라 7개의 CRS 네트워크 단계가 복합적으로 수행되므로 불

필요한 회로 중복이 제거되도록 설계되었다. 그림 6은 설계된 Permuter 블록의 기능검증 결과이다. Permuter 블록에 데이터 1이 입력되는 경우에 CRS 값이 1씩 증가 할수록 8비트씩 시프트되는 것을 확인할 수 있다.

data_in	1	1					
shift	0	0	1	2	3	4	5
data_out	1	1	256	65536	16777216	4294967...	1099511...

그림 6. Permuter 블록의 기능검증 결과

4. 메모리 구조 및 제어

블록지컬 방식의 LDPC 복호기는 DFU에 의한 복호연산 중간 결과값들을 메모리에 저장하고, 읽어오는 과정이 반복적으로 수행된다. 본 논문에서는 이와 같은 메모리 읽기/쓰기 동작의 동시 수행을 위해 dual-port 메모리를 사용하였다. 복호기 구조의 메모리 읽기/쓰기 동작은 메모리 어드레싱이 중요하다. 메모리 어드레싱은 PCM과 관계가 있으며, 특히 복호중인 블록이 레이어의 몇 번째에 위치하고 있는지가 중요하다. 그림 7은 레이어 간의 메모리 어드레싱 타이밍도를 보여준다. 블록지컬 방식의 layered 복호에서는 레이어 간의 메모리 쓰기/읽기 과정에서 데이터 해저드(data hazard)가 발생하게 된다. 이러한 데이터 해저드를 방지하기 위해 그림 7과 같이 stall을 삽입하여 정상적인 메모리 어드레싱이 가능하도록 하였다. 그림 8은 부호화율 1/2, B=1,944비트에 대한 메모리 어드레싱 동작의 기능검증 결과이다. 데이터 해저드를 방지하기 위해 각 레이어마다 3 클럭 사이클의 stall이 삽입되어 정상적인 메모리 읽기/쓰기 동작이 이루어지는 것을 확인하였다.

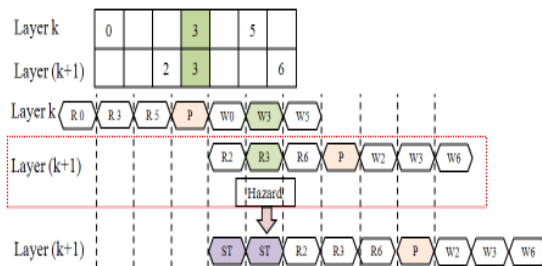


그림 7. PCM과 메모리 어드레싱 타이밍도

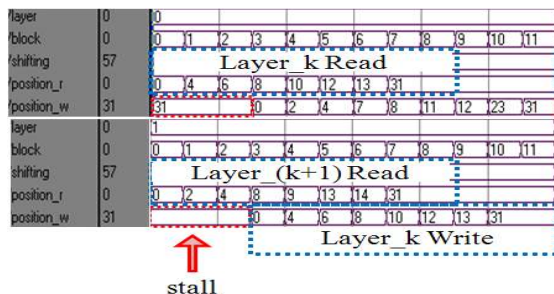


그림 8. 메모리 어드레싱 기능 검증

IV. 결 론

본 논문에서는 다중 블록길이의 부호화율 1/2을 지원하는 IEEE 802.11n용 LDPC 복호기의 최적화된 구조를 제안하였다. 다중 블록길이의 효율적인 구현을 위해 PCM 값을 저장하는 H-ROM의 최적화 방법을 제안하였으며, 이를 통해 H-ROM 메모리 용량을 약 42% 감소시켰다. 연산 복잡도와 하드웨어 복잡도를 효율적으로 감소시킬 수 있도록 최소합 알고리즘 기반의 layered 구조로 DFU를 설계하고 검증하였다. 향후 다중 블록길이를 지원하는 LDPC 복호기의 FPGA 구현을 통해 하드웨어 동작을 검증할 예정이다.

참고문헌

- [1] T.J. Richardson and R.L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, pp. 599-618, Feb. 2001.
- [2] D.C. MacKay, "Near Shannon Limit Performance of Low Density Parity Check codes." *Electronics letter*, vol. 33, no. 6, Mar. 1997.
- [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 Shnnon limit," *IEEE Communication letters*, vol. 5, pp. 58-60, Feb., 2001.
- [4] IEEE P802.11n/D3.07, "Draft Amendment to Standard for Information Technology- Telecommunications and information exchange between systems-Local and Metropolitan networks-specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Enhancements for Higher Throughput", 2008.
- [5] F. Zarkeshvari and A. Banihashemi, "On implementation of min-sum algorithm for decoding low-density parity-check (LDPC) codes", *Proc. IEEE GLOBECOM*, vol. 2, pp. 1349-1353, Nov. 2002.

※ 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업의 결과임(No. 2010-0005589)

※ 반도체설계교육센터(IDEC)의 CAD Tool 지원에 감사드립니다.