

완전 병렬 구조의 LDPC 복호기 VHDL 설계

정지원*

*한국해양대학교 진파공학과 교수

VHDL Design of full Parallel Structure for LDPC Decoder

Ji-Won Jung*

*Department of Radio Science Engineering, Korea Maritime University, Busan 606-701, Korea

요 약 : 본 논문에서는 LDPC 부호화 알고리즘으로는 Linear-Congruence 방법을 사용하였고, QPSK 변조 방식에 적용하여 최적 설계 파라미터를 분석하였고, 이를 토대로 양자화 비트수에 따른 성능 분석 및 잡음분산에러의 영향에 따른 최적 설계 방법을 분석하였으며, 완전 병렬 구조의 LDPC 부호를 VHDL 설계 하였다.

핵심용어: LDPC, Linear-Congruence, QPSK변조, VHDL

ABSTRACT : This paper analyzed optimal design parameters for LDPC coded system using the Linear-Congruence encoding method and QPSK modulation scheme. Based on these Parameters, This paper proposed performance evaluation according to number of quantized bits and noise-variance, and designed LDPC codes by VHDL

KEY WORDS : LDPC coded system, Linear-Congruence encoding method, QPSK modulation scheme.

1. 서 론

현재 채널 부호로는 길쌈 부호, 블록 부호 (BSH, RS등), 연쇄 부호등이 널리 사용되고 있으나, 이들 부호의 성능은 Shannon Limit와

다소 큰 차이를 보여 왔다. 최근에 실제적인 복호 알고리즘에서 Shannon의 채널 용량 한계에 근접하는 Turbo 코드와 LDPC 코드가 관심의 대상이 되고 있다[1][2]. LDPC 부호는 터보 부호에 비해 정정되지 않는 오류들을 대

* jwjung@hhu.ac.kr 051)410-4424

부분 검출할 수 있고, 복호화의 복잡도가 낮을 뿐 아니라 좋은 거리 특성으로 오류마루 현상이 나타나지 않고, 완전 병렬 처리로 고속 처리가 가능한 장점이 있다. 반면에 부호화 부분의 높은 복잡도가 LDPC코드의 단점이었으나 최근에 삼각행렬 분해법, Linear-Congruence 방법 등 부호화 방법이 제안되고 있다[3][4].

본 논문에서는 LDPC 부호화 알고리즘으로 Linear-Congruence 방법을 사용하였고, QPSK 변조 방식에 적용하여 최적 설계 파라미터를 분석하였고, 이를 토대로 양자화 비트수에 따른 성능 분석 및 잡음분산에러의 영향에 따른 최적 설계 방법을 분석하였으며, 완전 병렬 구조의 LDPC 부호를 VHDL 설계 하였다.

2. LDPC 부호화 알고리즘

Fig. 1은 비트의 정보원이 LDPC 부호기에 입력되어 n 개의 부호어가 출력되어 (n, k) LDPC code라 부르며, 채널을 통과하여 수신되는 n 개의 수신 부호는 check to bit node에서 각 bit 노드에 있는 정보를 갱신하고 bit to check node에서는 check node 정보를 갱신하면서 복호되는 비트 r 이 $H \cdot r=0$ 이 만족하도록 반복한다. 복호시에는 검사 행렬 H 원소의 1의 위치를 나타내는 ROW, COLUMN의 index를 저장하여야 한다.

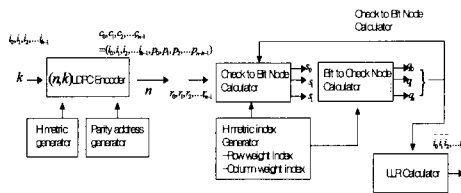
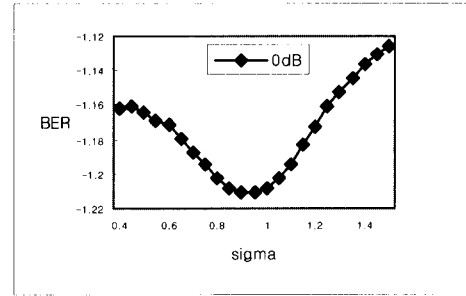
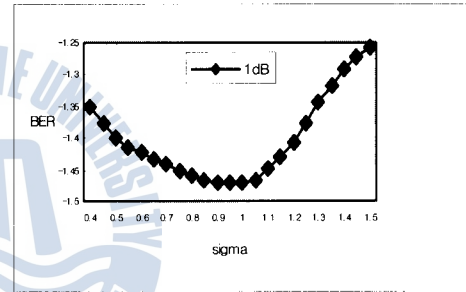


Fig. 1 Block diagram for LDPC encoder/decoder

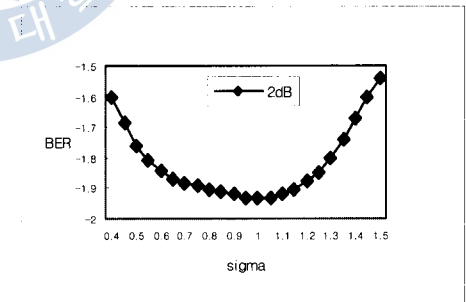
3. 잡음 분산 평가에러에 의한 L_c 값 추정



(a) $E_b/N_0 = 0$ [dB]



(b) $E_b/N_0 = 1$ [dB]



(c) $E_b/N_0 = 2$ [dB]

Fig.2 BER for Noise-Variance

Fig. 2는 잡음평가 에러를 가로축으로 하여 $N=192, R=1/2$ 인 부호기에 대해 iteration=30회 하여 복호한 BER 값을 도시한 것이다. 여기

서 σ 를 채널 잡음의 표준 편차, σ^* 를 평가 에러를 포함한 표준 편차라 정의 하면 가로축의 인덱스 χ 는 σ^* 가 $\sigma^* = \chi \times \sigma$ 만큼의 평가 에러를 포함하고 있음을 의미한다. Fig. 2에서 알 수 있듯이 채널 신뢰도 값 $L_c = 2/\sigma^2$ 를 추정할 때 최소한 σ^* 이 $[0.8\sigma - 1.15\sigma]$ 의 범위내에 있도록 하여야 할 것이다. 또한 σ^* 이 실제 σ 보다 다소 작게 유지할 필요가 있다. L_c 값을 {1,2,4}로 이산화시키므로 실제 L_c 값과 이산치 L_c 값의 차이로 인한 오차가 발생하게 된다. 여기에서 발생하는 오차를 잡음 평가 에러로 환산하면, $\sigma^* = \sigma/1.15$ 이 된다. 이 오차를 잡음 평가 에러로 환산하면 $\sigma^* = 1.12 \times \sigma$ 이 된다. 이 경우에는 Fig. 2에서 예측할 수 있듯이 성능열화가 초래될 가능성이 있다.

4. LDPC구현을 위한 블록도

복호기 구조는 크게 Check to Bit 블록과 Bit to Check 블록으로 구성되어지고 Fig. 3과 같다.

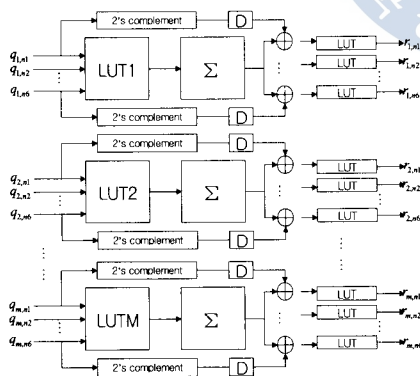


Fig. 3. Check to Bit node Block

초기화 되어 나온 q 값을 Look up Table에 대응되는 값으로 환산하고 각 행별로 parity check 행렬의 1의 위치의 값을 모두 더한 후 자기 자신을 빼주고, 그 결과를 다시 LUT에

대응되는 값으로 변환시키면 최종적으로 Check to Bit 결과 값을 구할 수 있다. 이때 LUT 값은 양의 값만을 갖기 때문에 부호 부분은 따로 처리해 주어야 한다.

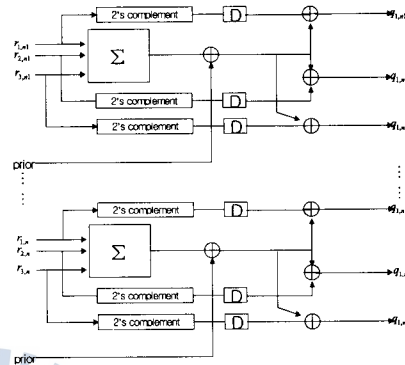


Fig. 4. Bit to Check node Block

각 열마다 parity check 행렬의 1의 위치는 column weight에 의해 결정된다. Fig. 4는 column weight가 3인 Bit to Check node 블록이다. 각 열의 r 값을 모두 더한 값에 초기 값을 더해서 나온 결과를 hard decision해서 얻은 결과가 복호화를 통해 얻은 전송되어진 데이터가 된다. 만약 이 값과 parity check 행렬의 곱이 0이 되지 않으면 자기 자신 r 값을 빼고 각 과정을 반복한다. Log Domain으로 계산된 복호기는 위 그림들의 블록도를 살펴 보면 parity check 행렬상의 1의 위치의 값을 모두 곱해야 하는 복잡한 과정을 단지 덧셈기와 Look up Table로 간단하게 구현되어진다.

4.1. Integer Simulation

Fig. 5에서 n 비트를 입력하면 bit-shifting 하면 m 비트가 출력되고, 여기서 $L_c = \{1,2,4\}$ 를 사용한다. m bit 가 초기화 되어 지면 Checkto Bit node 블록으로 m 비트가 입력되

어서 다시 r 비트로 출력 되어 진다. 출력 되어진 r 비트를 Bit to Check node 블록으로 입력하면, q 비트가 출력되어져서 다시 Check to Bit 블록으로 입력되는데 iteration을 반복 하게 된다. Bit to Check node 블록에서 q 비트가 출력될 때 또한 LLR을 계산하기 위해 L 비트가 출력 되어 진다. 이 표현을 (n,m,r,q,L) 이라 하면 $(3,5,5,5,5)$, $(4,6,6,6,6)$, $(5,7,7,7,7)$, $(6,8,8,8,8)$, $(7,9,9,9,9)$ 로 표현 할 수 있다. 이에 따라 r_k 의 양자화 비트수에 따른 성능 곡선을 보면 Fig. 6과 같다.

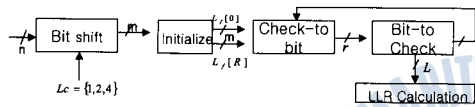


Fig. 5 Decoding block diagram of LDPC code

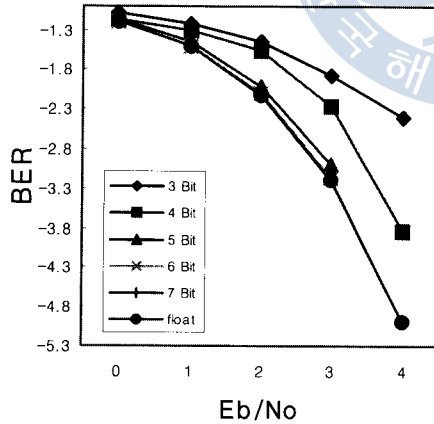


Fig. 6 BER according to number of quantization bits

5. 완전 병렬 구조의 VHDL 설계안

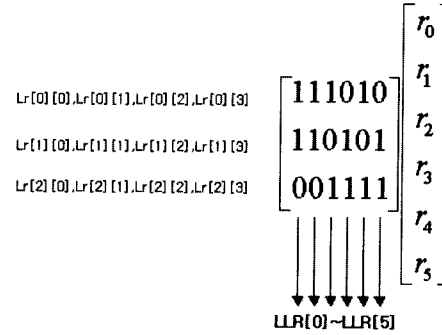


Fig. 7 LLR assignment for implementation

bit node의 확률은 $LLR[0] \sim LLR[5]$ 로서 Fig. 7에 표시 하였다. 이 때, bit node의 update 된 값의 column 열의 1의 위치를 동시에 알 수가 없다. 따라서 1의 위치를 나타내기 위해서는 column 열에 대한 1의 위치별로 카운터를 써야 하기 때문에 순차적으로 진행 될 수밖에 없고, 완전 병렬 구조가 이루어 질 수 없다. 따라서 본 논문에서 구현한 완전 병렬 구조 LDPC 복호기는 앞 절에서 명시한 것과 같이 LDPC 복호기는 크게 Check to Bit node block과 Bit to Check node block으로 구성되 어지고

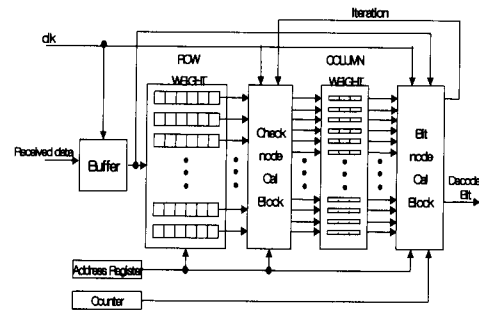


Fig. 8. Full parallel structure for LDPC decoder

Fig. 8에서 보는 것과 같이 완전 병렬 구조의 LDPC 복호기를 구성하는 것은 크게 수신된 데이터를 저장하는 buffer 부분과 Check to Bit node 블록에 해당하는 Check node 계산 블록 과 Bit to Check node 블록에 해당하는 Bit node 계산 블록으로 구성할 수 있다.

6. 시뮬레이션 결과

Fig. 9에서 Fig. 11까지는 3절과 4절에서 구현 최적의 파라미터를 적용하여 hardware 구현을 위한 부호화율 1/2 일 때 VHDL 시뮬레이션 한 결과이다. Fig. 9는 buffer에서 출력되는 데이터를 나타내었고, Fig. 10은 bit to check 과정에서 갱신된 값이 출력되는 값을 나타내었고, Fig. 11은 check to bit 과정에서 갱신되는 값이 출력되는 과정을 나타내었다.

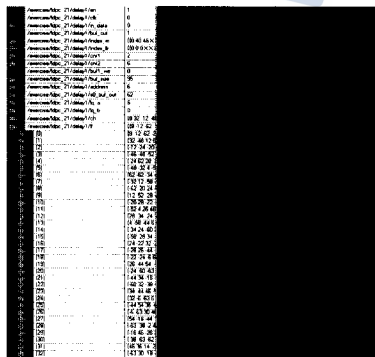


Fig. 9. Buffer VHDL result of LDPC decoder

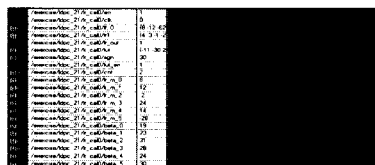


Fig. 10. Bit to Check VHDL result of LDPC decoder

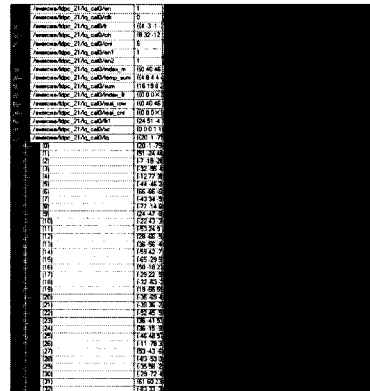


Fig. 11 Check to Bit VHDL result of LDPC decoder

7. 결 론

본 논문에서는 LDPC 부호화 알고리즘으로 Linear-Congruence 방법을 사용하였고, QPSK 변조 방식에 적용하여 LDPC 복호기 구현을 위한 최적 설계 파라미터를 도출하였으며, 시뮬레이션 결과 양자화 비트수는 6 비트, LLR은 8비트가 최적의 파라미터임을 알 수 있다. 또한, 잡음분산에러의 영향에 따른 채널 추정 파라미터를 성능별로 분석하여 구현시 간편하게 하기 위해 이산적인 값으로 할당하였다. 앞에서 도출한 최적의 파라미터를 이용하여 완전 병렬 구조의 LDPC 복호기 구현을 하였으며, 이를 MODELSIM 컴파일러를 이용하여 VHDL로 설계하였다. 타이밍 시뮬레이션 결과, 완전 병렬 구조의 복호기와 기존의 부분 병렬 복호기는 성능면에서 에러정정능력이 일치함을 확인하였다.

참 고 문 헌

[1] C. Berrou, A.Glavieux and P. Thitimajshima, Near Shannon limit error correcting coding

- and decoding : Turbo-codes, in *Proc.Int Conf. Communications(ICC)*, pp.1064-1070, 1993.
- [2] J. Campello, D. S. Modha, and S. Rajagopalan, Designing ldpc codes using bit-filling, in *Proc. Int. Conf. Communications(ICC), Helsinki, Finland, 2001.*
- [3] R.G.Gallager, Low density parity check codes, in Research Monograph series, MIT Press, Cambridge, Mass, 1963.
- [4] D.J.MacKay, Good error-correcting codes based on very sparse matrices, *IEEE Trans. Inform. Theory*, IT-45, pp.399-431, Mar. 1999.
- [5] M.Sipser and D.A.Spielman, Expander Codes, *IEEE Trans. Inform. Theory*, vol. 42, pp. 1710-1722, Aug. 1996.
- [6] Seong-Jun Cho, Eun-A Choi, Ji-Won Jung, Sang-Jin Park, In-Ki Lee, Tae-Gil Lee and Nae-Soo Kim, Establishment of optimal Parameters for Implementing the LDPC Decoder, in *Proc. CIC'2003*, pp. 544, Oct. 2003.

원고접수일 : 2007년 1월 4일

원고채택일 : 2007년 1월 19일