논문 2009-46SD-11-12

Quasi-Cyclic Low Density Parity Check 복호기의 다양한 설계 관점에 대한 성능분석

(Performance Analysis on Various Design Issues of Quasi-Cyclic Low Density Parity Check Decoder)

정 수 경*, 박 태 근**

(Sukyung Chung and Taegeun Park)

요 약

본 논문은 LLR-BP 복호 알고리즘을 사용하는 LDPC 복호기의 하드웨어 구조 분석하고 효율적인 복호기의 설계 방법들을 제시하였다. 또한 설계 시 복호 성능 및 하드웨어 복잡도에 영향을 미칠 수 있는 다양한 설계 이슈들을 제시하고 복호 성능의 변화를 모의실험을 통하여 분석하였다. 오류확률을 전달하는 메시지의 양자화는 정수부 3비트, 소수부 4비트를 할당하였고, 복호 성능이 저하되지 않도록 사전정보에 정수부 2비트, 소수부 4비트를 할당하였으며 LUT로 구현되는 $\Psi(x)$ 함수를 조합회로인 PWL 블록으로 대체하여 하드웨어 구조의 개선에 대해 논의하였다. 복호 시간을 단축하기 위하여 중첩 스케줄링을 적용하고, 각 복호기 구조 및 설계 변수들의 제한에 따른 하드웨어 자원을 비교함으로써, 하드웨어 복잡도를 분석하였다.

Abstract

In this paper, we analyze the hardware architecture of Low Density Parity Check (LDPC) decoder using Log Likelihood Ration-Belief Propagation (LLR-BP) decoding algorithm. Various design issues that affect the decoding performance and the hardware complexity are discussed and the tradeoffs between the hardware complexity and the performance are analyzed. The message data for passing error probability is quantized to 7 bits and among them the fractional part is 4 bits. To maintain the decoding performance, the integer and fractional parts for the intrinsic information is 2 bits and 4 bits respectively. We discuss the alternate implementation of $\Psi(x)$ function using piecewise linear approximation. Also, we improve the hardware complexity and the decoding time by applying overlapped scheduling.

Keywords: Low density parity check(LDPC) codes, quasi-cyclic LDPC, Belief Propagation, fixed-length analysis

Ⅰ. 서 론

디지털 통신에서의 채널 부호화란, 데이터의 송수신 과정에서 생긴 오류를 정정해줄 수 있는 오류정정부호(Error Correcting Codes)의 적용을 의미하며, 고품질 대량 정보전송 서비스를 지원하는 오늘날의 무선 통신 시스템에서 채널 부호의 적용은 선택이 아닌 필수로 자리매김 하고 있다. 이러한 채널 부호로 최근 주목받고 있는 LDPC(low Density Parity Check) 부호는 1962년 Gallager[1]에 의해 처음으로 제안된 것으로, 상대적으로 1의 개수가 0의 개수에 비해 현저히 적은 $(n-k)\times n$ 이진 행렬(sparse parity check matrix)을 패리티 검사 행렬로 갖는 이진 선형 블록부호로 정의된다[1~2].

최초로 정의된 (n, j, k) LDPC 부호는 부호어의 길이가 n이고, 열(column)이 포함하는 1의 개수를 j, 각각의 행(row)이 갖는 1의 개수로 k를 결정하여 패리티

* 학생회원, ** 정회원, 가톨릭대학교 정보통신전자공학부
(Department of Information, Communication, and Electronic Engineering, The Catholic University of Korea)

※ 본 연구는 2009년도 가톨릭대학교 교비연구비의 지원으로 이루어졌음.
접수일자: 2009년7월16일, 수정완료일: 2009년11월4일

검사 행렬의 각 행과 열이 모두 동일한 개수의 1을 갖도록 균일하게 구성하며, Gallager는 이 부호를 반복 복호 하면 부호어의 길이 n이 커짐에 따라 오류확률이 지수적으로 0에 접근한다는 것을 보였다. 그러나 논문 발표 당시 LDPC부호는 반복복호가 가지는 많은 계산량 등을 비롯하여 그 당시 컴퓨터 기술로는 모의실험이 어려운 한계 등으로 인해 학계의 주목을 받지 못하였지만, 최근에 이러한 단점을 극복할 수 있는 여러 가지 방법들이 제시되고 있다. LDPC 부호의 성능은 Shannon 한계에 근접하여 우수한 성능을 나타내며 소형 무선 단말기 등에서의 수십 Mbps 이상의 데이터 처리가 요구되는 시스템에서는 반복 복호 시 계산량이 증가하는 Turbo 부호보다 병렬처리의 제한이 크지 않은 LDPC 부호가 유리하다[2~4].

LDPC 부호의 구현에 관한 연구는 낮은 복잡도와 높은 오류정정 능력을 갖는 부호기와 복호기의 설계인데, 부호기의 복잡도를 줄이기 위해 LDGM(Low Density Generator Matrix) 부호로 G 행렬을 구성하는 연구[5]와 더불어 복호기의 성능 향상을 위해서 패리티 검사 행렬을 구성하는 연구가 가장 활발하게 진행되고 있다. 패리티 검사 행렬을 구성하는 방법에는 랜덤하게 패리티 검사행렬을 생성하는 방법, 블록 단위의 구조적 패리티 검사행렬을 생성하는 방법 등이 있다. 최근에는 Richardson의 부호화 기법[6]을 사용할 수 있도록 구조적 LDPC 부호를 생성하는 연구들이 진행되고 있으며, 이들의 연구는 LDPC 부호의 복호기에 초점이 맞추어져 있으며 이들의 효율적인 VLSI구조에 대한 연구도 함께 진행되고 있다.

본 논문에서는 LLR-BP(Log Likelihood Ration -Belief Propagation) 복호 알고리즘 기반의 QC-LDPC (Quasi Cyclic Low Density Parity Check) 복호기의 성능 및 복잡도에 영향을 미칠 수 있는 다양한 설계 이슈들을 제시하고, 각 설계 변수들의 변화에 따른 성능을 비교 분석하여 효율적인 QC-LDPC 복호기 설계에 적용할 수 있는 설계 방법을 제안한다.

본 논문의 제 Ⅱ장에서는 QC-LDPC 부호의 기본적인 부/복호기의 구조와 복호 과정, 대표적인 복호 알고리즘에 대해 설명한다. 제 Ⅲ장에서는 QC-LDPC 복호기의 설계 시 고려해야 할 다양한 설계 이슈들을 제시하고, 여러 관점에서 효율적인 복호기의 설계에 대해 논의한 후, 모의실험을 통하여 성능을 비교 분석한다. 마지막으로, 제 Ⅳ장에서는 본 논문의 결론을 제시한다.

Ⅱ. QC-LDPC 코드

LDPC 부호의 패리티 검사행렬을 구성하는 방법에는 균일한 방법과 비균일한 방법이 있는데, 균일한 것보다 비균일하게 구성된 LDPC 부호의 성능이 더 좋은 것으로 알려져 있다. 그러나 이후 패리티 검사 행렬을 여러 개의 부행렬로 구성한 구조적 부호의 오류 확률이 비균일한 LDPC 부호에 가까운 오류성능을 내는 것이 실험을 통해 밝혀지면서[3], 하드웨어 구현에 유리한 구조적 부호의 규칙성 및 부행렬의 단순성 등을 이용한 VLSI 구현에 QC-LDPC 부호가 연구되고 있다.

LDPC 부호는 다른 선형 블록 부호들과 같이 생성행렬과 패리티 검사행렬 의존도가 높으므로 부호화기의 경우 다분히 패리티 검사행렬 H의 종류와 형태에 의존하게 된다. 복호기의 구조 또한 설계자가 고려한 시스템 변수의 우선순위에 따라 다양한 형태가 존재할 수 있는데, 대부분의 H 원소들이 '0'인 $m \times n$ 행렬은 아래 그림 1과 같이, 검사 노드(Check node)의 개수가 m이고, 변수 노드(Variable node)의 개수가 n개인 이분 그래프(Bipartite graph)로 생각할 수 있다.

LDPC 부호의 패리티 검사행렬은 1의 개수가 매우 적어 블록이 커져도 반복복호를 통하여 복호가 가능하

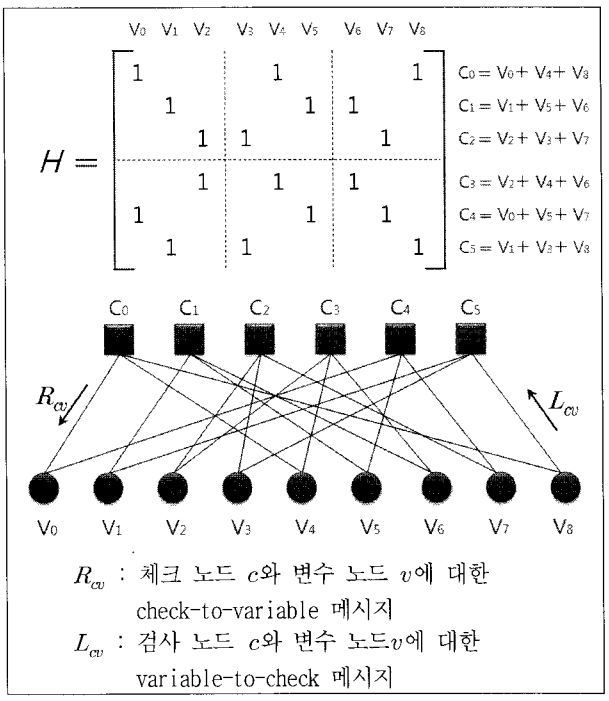

그림 1. LDPC부호의 검사행렬 예제와 이분 그래프
Fig. 1. The example of QC-LDPC parity check matrix and its bipartite graph.

며 터보 부호처럼 Shannon의 채널 용량 한계에 근접하는 성능을 보인다[3]. LDPC 부호의 복호를 수행하는 BP(Belief Propagation) 또는 합곱(Sum-Product) 복호기는 제한적인 거리 복호기(bounded distance decoder)보다 성능이 우수하며, 채널로부터 얻는 우도(likelihood)를 이용할 수 있는 연판정 복호기로서, 연집오류가 발생하는 채널에서도 연집오류를 정정할 수 있도록 일반화 될 수 있다[2].

LLR-BP(Log Likelihood Ration-Belief Propagation) 복호 알고리즘은 복호기의 검사노드와 변수 노드 간의 계산 복잡도를 낮추기 위해 실제의 확률을 사용하는 대신 log-likelihood ration을 사용하며, S_{cv}를 R_{cv}의 부호 부분, 검사 노드 c에 연결된 변수 노드들의 집합을 $N(c)$, 변수 노드 v와 $\frac{2r_v}{\sigma^2}$에 연결된 변수 노드들의 집합을$M(v)$라 할 때, 반복복호 알고리즘은 다음 표 1에 나타낸 것과 같이 요약될 수 있다.

먼저 LDPC 부호로 부호화된 부호열을 c_n이라 가정하면, 송신되는 부호열 s_n은 BPSK를 가정할 때

표 1. BP 복호 알고리즘의 복호 과정
Table 1. Decoding procedure on BP Algorithm.

Initialization : 패리티 검사행렬의 원소가 $H_{m,n}=1$인 모든 위치에서의 노드 값을 채널로부터 추정한 $\frac{2r_v}{\sigma^2}$ 정보로 초기화한다.

Check-node Update :

$$S_{cv}=\prod_{n\in N(c),n\neq v} sign(L_{cn}) \quad (1)$$

$$R_{cv}=-S_{cv}\Psi\left(\sum_{n\in N(c),n\neq v}\Psi(L_{cn})\right) \quad (2)$$

Variable node Update :

$$L_v=\sum_{c\in M(v)} R_{cv}-\frac{2r_v}{\sigma^2} \quad (3)$$

$$L_{cv}=L_v-R_{cv} \quad (4)$$

Decision : L_v를 양자화 한다. $sign(L_v)$를 계산하여, 0 혹은 1로 복호된 코드워드 x_n을 구성한다.

Termination : x가 $Hx=0$을 만족하면 복호를 마친다. $Hx\neq 0$이면 다시 Check-node Update단계로 돌아가 반복 복호를 수행한다. 미리 정해진 최대복호 횟수 이내에 종료되지 않는 경우에는 알고리즘을 강제 종료한다.

$s_n=2c_n-1$로 전송되며, 채널을 통해 잡음이 섞여 복호기에 수신되는 신호는 r_v로 표현할 수 있다. AWGN 채널에서 입력신호의 확률분포가 균일한 경우 채널의 intrinsic information은 $\frac{2r_v}{\sigma^2}$이며, 여기서 σ^2은 잡음의 분산을 나타낸다.

$\Psi(x)=\log(\tanh(\frac{|x|}{2}))$는 각 노드 계산이 이루어질 때 구현되는 비선형 함수이다. 각각의 알고리즘이 계산되는 Node Processing Unit으로 CNFU과 VNFU이 있으며, 부호가 계산되는 부분과 LUT값을 참조해 계산하는 덧셈기 트리(adder tree), L_v의 양자화 부분, sign-magnitude 포맷과 2의 보수 포맷간의 변환 부분 등으로 구성되어 있다.

III. 다양한 설계이슈에 대한 성능분석

1. 실험환경

본 논문에서의 성능분석을 위한 모의실험 환경을 그림 2와 같이 나타내었다. 실험에 사용한 패리티 검사행렬은 NASA 표준문서[7]의 QC-LDPC H행렬을 사용하였으며, C언어로 구현된 랜덤 발생기로부터 발생된 정보를 메시지로 사용하였다. NASA QC-LDPC 부호화기로부터 부호화된 신호 '1'과 '0'은 각각 '1'과 '-1'로 변조시키는 BPSK(Binary Phase Shift Keying)변조 후, 평균이 0이고, 분산이 No/2 인 AWGN(Additive White Gaussian Noise) 채널을 통하여 정보가 전송된다고 가정한다. 하나의 코드워드의 길이는 N = 8176 이고, 메시지의 길이는 7154 비트, 잉여 패리티 비트 P는 1022 비트이다. 따라서 이 블록 부호의 부호율은 7/8이고, 최대 반복복호 횟수는 15, Eb/No는 3.2dB에서 4.0dB까지 0.1혹은 0.2dB의 변화를 가진 실험 조건을 갖는다.

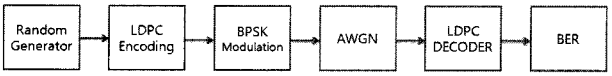

그림 2. 모의실험을 위한 시스템 블록도
Fig. 2. System block diagram under simulation.

2. 고정 소수점 비트 할당

고정 소수점 데이터를 (q,f)라고 나타낼 때 q는 데이터의 전체 비트 수이고, f는 소수점 이하 비트수라고 하자. 복호기 내의 모든 데이터 값을 고정 소수점을 이

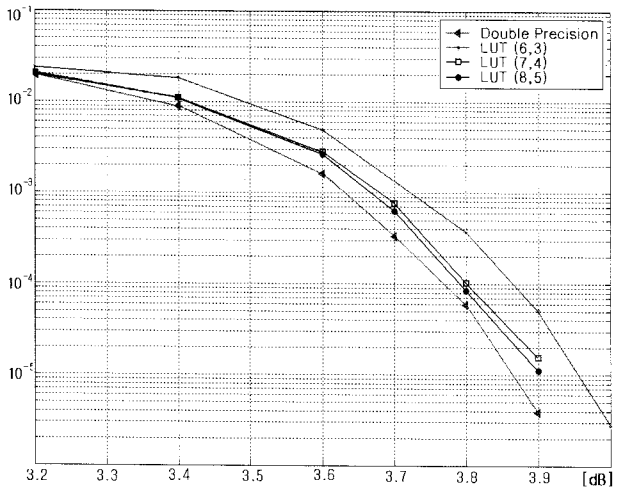

그림 3. 고정 소수점 비트 할당에 따른 오율
Fig. 3. Bit error rate according to the bit allocation.

용하여 구현할 때 비트 수를 적게 할당하면 하드웨어 상의 복잡도는 줄어들지만 양자화 오차가 커지고 비트 수를 많이 할당하게 되면 정확도가 높아져 복호기의 성능은 향상되지만 하드웨어의 비용이 증가함은 물론 동작 속도도 느려진다. 그러므로 고정 소수점을 이용하여 구현할 때 성능 저하를 줄이기 위해서는 데이터의 비트 수를 적절하게 선택하는 것이 중요하다. 아래 그림 3에 복호기 내의 메시지 정보를 $q=6,7,8$일 때, $f=3,4,5$로 (부호 비트 미포함) 할당하여 양자화 하였을 때의 성능 변화를 나타내었다.

먼저 정수부 비트 수를 고정시키고 소수부 비트 수를 변경시킬 경우 전체 비트 수도 같이 변하게 하여 성능 분석을 하였다. 소수점 이하 비트수가 커짐에 따라 성능이 향상됨을 확인할 수 있다. 소수점 이하 비트수가 4일 때 까지는 비트 오율의 값이 크게 차이를 보이며 향상되나, 5비트 이상부터는 메시지를 양자화하지 않고 실수로 복호한 경우와 비교하여 비트 오율의 차이가 크지 않았다. 그러나 소수점 비트가 3으로 할당된 $(6,3)$의 경우는 $(7,4)$, $(8,5)$의 경우보다 소수점 이하의 비트수가 부족하여 성능이 떨어진다. 이 실험을 통해 성능을 분석한 결과 고정 소수점 연산에서 전체 비트 수를 최적으로 고정시킬 경우 소수점 이하 4비트일 때의 결과가 실수 값을 이용한 결과와 근접함을 확인하였다. 따라서 전체 복호기 내에 전달되는 모든 데이터는 $q=7$, $f=4$인 $(3,4)$ 고정 비트 할당을 갖도록 하였다.

전체 메시지의 비트 할당 외에 복호기 내의 하드웨어 복잡도를 줄이기 위해서 상대적으로 덜 중요한 정보에 대한 고정비트를 감소시켜 비용 면에서 효율적으로 설계할 수 있다. 먼저 채널 intrinsic information z값은 실제로 수신되는 r을 받아서 LDPC 복호기의 내부에서 반복 복호에 사용되는 정보인데, 위에서 결정한 (q,f)에 따라 모두 $(q+1)\times N$ 크기의 메모리가 필요하다. 그러나 z값은 반복복호의 초기화와 비트 결정에 사용되는 값으로, 반복 복호되면서 누적되거나 변하는 값이 아니고 초기에 수신된 r에 많이 의존한다. 변조 방식이 BPSK이고, 삽입된 에러가 AWGN에 의한 것이므로, 전체 z비트 중 정수부에 해당하는 $z-f$는 주로 -1, 0, 1 값에 치중해 있으므로 기존의 정수부 비트보다 최소 1비트 적은 값을 할당하여 초기화해도 성능 면에서 거의 차이가 없는 BER 커브를 나타낸다.

또한 주로 Look-up table(LUT)로 구현되는 $\Psi(x)$값은 입력 x에 따른 출력 값을 미리 저장해 두고 $\Psi(x)$ 값을 읽어내는 메모리이기 때문에, 입력에 따른 출력의 종류가 많을수록 LUT의 면적이 증가한다. LUT는 소수부 비트의 길이 f를 선택하는 것이 중요한데, 그것은 작은 값을 많이 사용하여 복호가 이루어지기 때문이다. 따라서 이전 단계에서 언급한 대로 입력 x에 대한 고정 소수비트 f가 결정되면 그에 따른 최소값의 입력으로 $\Psi(x)$의 출력 값이 결정되므로 자연스럽게 LUT의 정수비트는 결정된다. 예를 들어 x의 $f=4$일 경우 LUT 입력의 최소값은 $x=0.0625$가 되며, $\Psi(x)$는 3.4661로 정수비트는 2로 결정된다. 하지만 LUT의 크기를 감소시키기 위해 이를 위한 비트의 할당을 따로 고려할 수 있으며, 이 경우 하드웨어 비용이 감소하는 만큼 복호기의 성능저하가 예상된다.

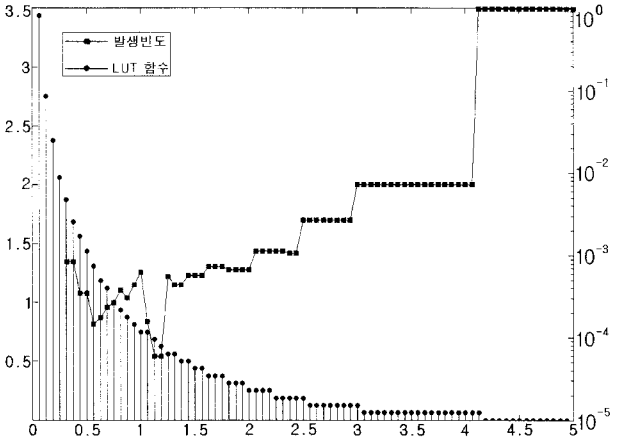

그림 4. 양자화 $\Psi(x)$ 값과 발생빈도
Fig. 4. Quantization value of $\Psi(x)$ and its occurance rate.

표 2. PWL 구간 구분과 $\Psi(x)$의 MSE
Table 2. PWL block range and Mean Squre Error of $\Psi(x)$.

분할	I/O	1영역	2영역	3영역	4영역	5영역	6영역	7영역	MSE
PWL4	x	~2.5	~3.0	~4.125	4.8175~				0.1569
	$\Psi(x)$	1.5~0.1875	0.125	0.0625	0.0				
PWL5	x	~0.5625	~2.5	~3.0	~4.125	4.8175~			0.02300
	$\Psi(x)$	2.75~1.4375	1.5~0.1875	0.125	0.0625	0.0			
PWL6	x	~0.4375	~1.5	~2.5	~3	~4.125	4.8175~		0.01553
	$\Psi(x)$	2.75~1.4375	1.1875~0.375	0.3125~0.1875	0.125	0.0625	0.0		
PWL7	x	~0.4375	~0.75	~1.75	~2.5	~3.0	~4.125	4.8175~	0.01299
	$\Psi(x)$	2.875~1.4375	1.375~0.9375	0.875~0.4375	0.375~0.1875	0.125	0.0625	0.0	

3. $\Psi(x)$ 함수의 근사화

Gallager 함수라고도 하는 $\Psi(x)$ 값은 각각의 체크 노드와 변수 노드가 전달하는 LLR 메시지를 계산하기 위해 필요한 값으로, LDPC 복호기에서 주로 유한 비트 폭을 갖는 LUT로 구현된다. LUT는 각 반복 복호 단계에서 식 (2)에 나타난 것과 같이 체크노드 계산에 중복 사용되는 값이다. 따라서 LUT는 체크 노드 개수의 두 배만큼 사용되며, 이것이 체크 노드의 회로 복잡도에 큰 영향을 미친다.

일반적으로 ROM 메모리로 구성되는 LUT를 $\Psi(x)$의 근사값을 갖는 조합회로로 대신한다면 체크 노드의 덧셈기 트리 등을 설계할 때 파이프라인을 구성하는 등 여러 가지 개선의 여지가 생길 수 있다. 먼저 $\Psi(x)$함수의 값을 근사화 하기 전에 정확도를 높여야 할 부분을 찾기 위해 실제 LDPC 복호 과정에서 많이 참조하는 $\Psi(x)$값을 조사하면 그림 4와 같다.

그림에서 $\Psi(x)$값이 0이 되는 값의 빈도가 98.6%, 0.0625값의 빈도가 0.7218%로 함수 값이 커짐에 따라 참조하는 빈도가 급격히 감소함을 알 수 있다. 따라서 BP 복호 알고리즘을 구현할 때 빈도가 높은 함수 값이 정확해야 누적되는 오차를 최소화 할 수 있다. 이를 근거로 $\Psi(x)$함수의 값을 입력 x의 범위에 따라 1차함수의 PWL(Piecewise linear approximation)로 근사화하면 $\Psi(x)$값을 LUT 메모리 대신 조합회로만으로 구현할 수 있다. 본 연구에서는 빈도가 높아 복호에 절대적인 영향을 미치는 $\Psi(x)$값의 최소값 3개를 상수로 선택하고, 나머지 구간을 1차함수 조각으로 구현하였다. 상수값 3개와 1차함수 조각의 개수가 1,2,3,4일 때 각각의 블록을 PWL4, PWL5, PWL6, PWL7로 정의하였고 각 구간별 정보 및 각각의 MSE(Mean Square Error)는 표 2에 나타나 있다.

7bit 균일 양자화를 가정한 LUT로 LLR 메시지를 계산한 결과와 PWL 방법으로 $\Psi(x)$를 근사화 하였을 때의 복호 성능을 BER로 비교하면 그림 5와 같으며, 구간을 잘게 분할하였을 경우의 BER이 LUT에 근접하게 나타난다. 이 실험 결과로 보아 PWL4는 성능에서 현저한 저하를 보이며 PWL5~PWL7의 경우는 LUT와 비교할 만한 성능을 보이므로 설계환경에 따라 LUT의

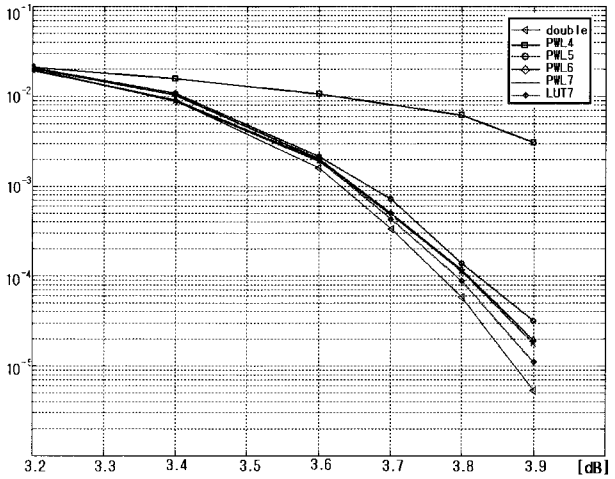

그림 5. PWL 복호 성능의 비교
Fig. 5. Performance comparison of the PWL decoding.

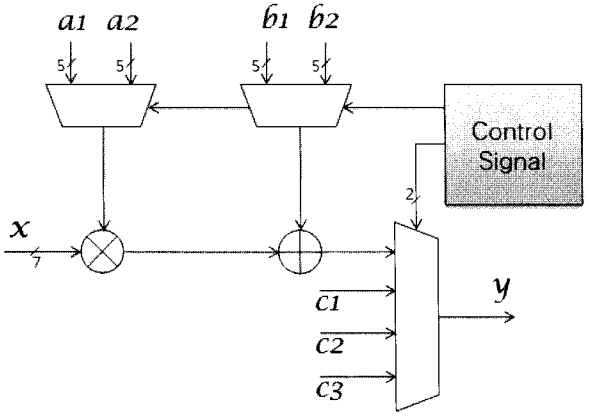

그림 6. PWL5 로직 블럭도
Fig. 6. Logic block diagram for PWL5.

대체가 가능하다.

그림 6은 입력구간을 5구간으로 나누고 2개의 1차함수 회로를 갖는 PWL5 블록을 하드웨어로 설계할 때 블록 다이어그램을 나타낸다. PWL로 LUT를 대체할 경우 적은 로직을 사용하여 구현하게 됨으로써, 전체적인 복호기 구조에 파이프라인 등을 적용할 때 메모리를 사용하는 것보다 유리하며 여러 가지 최적화 기법들을 적용할 수 있는 여지를 가진다. 여기에서 a_1, a_2는 1차함수의 기울기를, b_1, b_2는 y축 절편을 나타내며, c_1, c_2, c_3는 최소값 3개의 상수를 나타낸다.

4. Scheduling

스케줄링은 효율적인 메모리 관리를 위해 적용되는 것은 물론, 하드웨어 모듈이 요구하는 데이터를 가장 빠른 시간에 주고받음으로서 성능을 향상시키는 기법이다[8]. 이러한 기법을 검사 노드와 변수 노드로의 메시지 전달이 차지하는 비중이 큰 LDPC 복호기에 적용한다면, 가장 이상적일 경우 각 복호 단계를 중첩시킴으로서 복호 시간을 절반가량 감소시킬 수 있다. 하지만 이렇게 단순히 각 노드간의 메시지 전달을 중첩시켜 복호기를 구성할 경우에 검사 노드와 변수 노드 사이의 데이터 의존도 때문에 메모리 충돌이 일어날 수 있으므로 효율적인 메모리 스케줄링이 필요하다[9].

일반적으로 VNFU(Variable Node Functional Unit)과 CNFU(Check Node Functional Unit)의 연산은 CNFU의 계산이 끝나고 VNFU의 계산을 순차적으로 수행하여야 한 번의 반복복호가 종료된다. 그러나 각 Node Functional Unit들이 H 행렬의 행과 열에 대해 중첩되어 계산될 수 있다면 복호에 소요되는 시간을 단축시킬 수 있다.

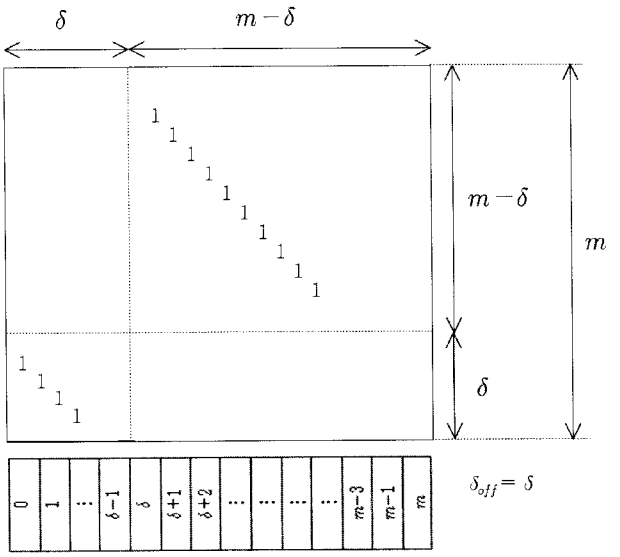

그림 7. 오프셋을 갖는 $m \times m$ 부행렬
Fig. 7. An $m \times m$ submatrix with offset.

QC-LDPC 부호의 패리티 검사 행렬을 구성하는 '1'은 항등행렬이기 때문에 규칙적인 배열을 갖는다. 패리티 검사 행렬은 1이 시작하는 열의 위치인 δ_{off}(offset)만 다르고 크기가 같은 $J \times K$개의 정방행렬을 부행렬로 가지며, 다음 그림 7과 같이 나타낼 수 있다.

오프셋이 $\delta_{off} = \delta$ 으로 표현되는 $m \times m$ 부행렬에서 1의 개수는 m개이므로 메모리가 m개 필요하며, $0, 1, 2, \cdots, \delta, \delta+1, \cdots, m-1$까지의 메모리 주소를 갖는다. 부행렬의 첫 번째 행(row)에 1이 존재하는 메모리 주소를 $M(\delta)$라고 할 때, $0, 1, \cdots, \delta-1$까지의 주소를 갖는 영역을 A, $\delta, \delta+1, \cdots, m-1$까지의 주소를 갖는 메모리 영역을 B라고 정할 수 있다. 이 때, 검사 노드에서 계산하는 시간이 m 싸이클, 변수 노드에서 계산하는 시간이 m 싸이클이므로 1회 반복복호에 걸리는 시간은 $2m$ 싸이클이 된다. 여기에 CNFU과 VNFU의 계산을 중첩하면 B 영역 크기에 비례하여 복호시간이 감소한다. 실제로 $B-1$ 싸이클 만큼의 복호시간이 절약되며 최대 복호횟수가 n일 때 최대 $n(B-1)$회까지 클럭 수를 줄일 수 있다.

그러나 NPU(Node Processing Unit)의 효율을 높이기 위한 부행렬의 메모리 중첩은 패리티 검사 행렬을 구성하는 각 부행렬의 오프셋에 따라 달라지며, CNFU와 VNFU는 일반적인 (J, K) 반병렬 구조의 복호기에서 $J+K$개가 존재한다. 이들 각각의 NPU의 계산은 $J \times K$개 부행렬의 데이터 메모리가 서로 다른 각 부행

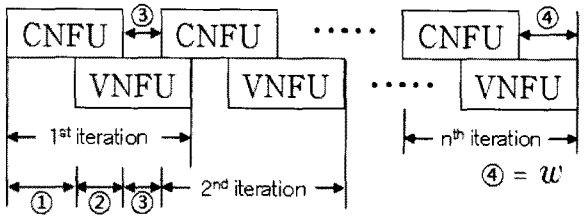

그림 8. CNFU과 VNFU의 실제 스케줄링
Fig. 8. practical scheduling of CNFU and VNFU.

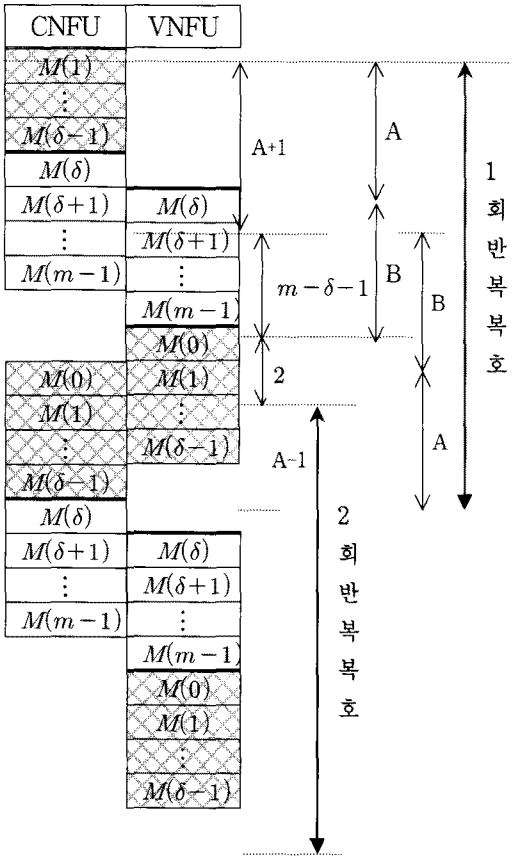

그림 9. 제안된 스케줄링과 메모리 주소
Fig. 9. Proposed memory scheduling and memory address.

렬의 오프셋을 고려하면서도 충돌 없이 처리되도록 고려해야 한다. 따라서 위에서 언급한 오버래핑을 적용한 스케줄링에서 $2A$ 영역은 줄일 수 없지만 $2B$ 영역은 줄일 수 있다.

메모리 영역의 비교를 통해 알아본 것과 같이, 스케줄링을 적용하지 않은 부호어의 반복복호시간 $2mn$에 비해, 스케줄링을 적용한 복호시간 $(m+2)n+(\delta+1)$은 부행렬의 오프셋 δ가 부행렬의 크기 m보다 훨씬 작을수록 중첩으로 얻을 수 있는 효율이 크다. 각 NPU들이 동시에 메모리에 접근하는 경우, 각 NPU에 메모리 주소를 전달하는 주소발생기(address generator)는 오프셋이 서로 다른 부행렬을 동시에 복호하기 위해서 $\max\{\delta_{j,k}\}$를 기준으로 주소를 발생시켜야 하며, 1회 복호 시 $m-\max\{\delta_{j,k}\}-1$ 만큼의 복호 시간을 절약할 수 있다. 중첩된 스케줄의 블록도는 그림 8을 통해 확인할 수 있으며, 복호시 메모리 중첩은 그림 9에 상세히 나타나 있다. 복호시간은 식 (5)와 같이 나타낼 수 있다.

$$\begin{aligned} Total_{dec} &= (①+②+③)\times n+④ \\ &= n(\delta+(m-\delta)+2)+(\delta-1)\ [\text{cycle}] \\ &= n(m+2)+(\delta-1)\ [\text{cycle}] \end{aligned} \tag{5}$$

오버래핑의 스케줄링을 적용한 경우 전체 복호 사이클에서 기대할 수 있는 최소 복호 시간은 N개의 부호어를 복호하는 경우 고려되는 복호 소요시간은 $Time_{dec}=N\times[n(m+2)+(\delta-1)]$이며, 스케줄링을 적용하지 않았을 때 걸리는 부호어의 반복 복호시간은 $2mnN$이 된다. LDPC의 복호는 정해진 최대 반복 횟수 내에서 메시지 전달을 통해 오류를 정정하기 때문에 중첩된 스케줄링을 적용한 경우, 높은 SNR보다 반복복호의 최대 횟수를 채워야 하는 낮은 SNR에서 더 많은 사이클을 절약할 수 있어 효율적이다.

5. Decoder Configuration

LDPC 복호기 구조는 크게 병렬구조(Fully-parallel)[10], 반 병렬구조(Partly-parallel)[9, 11~12], 직렬구조(Fully-serial)로 구분할 수 있다. 패리티 검사행렬이 비균일하며 랜덤한 방식으로 구성되었을 때 복호기는 설계자에 따라 NPU의 구성, 데이터 스케줄링 및 내부 결선의 라우팅 등이 고려되어 복잡한 반면, QC-LDPC 부호의 패리티 검사행렬은 비교적 단순한 구성으로 구현이 매우 직접적이다.

직렬구조의 LDPC 복호기는 NPU의 면적을 가장 적게 필요로 하여 하드웨어 자원을 공유함으로써 칩 면적을 최소화 시킬 수 있다. 이 구조를 갖는 복호기는 복호과정에서 요구되는 잦은 메모리 접근과 매우 느린 동작속도로 고속 데이터 전송률을 지원하기 어렵다.

병렬구조의 LDPC 복호기는 가장 많은 NPU가 필요하여 하드웨어 비용이 높다. 하지만 이분 그래프의 병렬 구조를 하드웨어로 옮겨 놓은 것과 같이 단순한 데이터 흐름을 갖고 빠른 복호 속도를 갖는다. 하지만 데

이터 라우팅을 위한 방대한 내부 결선으로 큰 블록 길이를 갖는 경우 구현이 곤란하다.

반 병렬구조는 직렬구조와 병렬구조의 장단점을 절충한 형태이며 칩 면적, 메모리 효율, 처리속도 등에서 많은 장점을 갖는다[9, 11]. 이를 구현할 경우 패리티 검사 행렬은 여러 개의 부행렬로 구성된 QC-LDPC 부호에 적합하며, 이 경우 병렬화 지수 f_p를 높일수록 복호 속도가 개선되나, 그만큼의 하드웨어 면적이 추가되므로 장단점을 고려하여 f_p와 어플리케이션의 요구 성능에 적합하게 병렬화 하는 것이 필요하다. 이러한 설계 변수들의 변화에 따른 시스템 자원의 사용을 변수로 정리한 것이 아래 표 3에 나타나 있다.

(J, K) QC-LDPC 부호는 $J \times K$개의 부행렬로 구성된 패리티 검사 행렬을 갖는다. 이 H 행렬의 부 행렬이 각각 $m \times m$ 정방행렬로 구성되어 있고, 복호기의 패리티 검사행렬이 일반적이라고 할 때 각 복호기의 구성에 따른 성능과 요구 면적은 다음과 같은 파라미터로 정리될 수 있다.

여기서 w는 그림 8의 ④에 해당하는 크기를 갖는 복호 대기 시간이며, n번째 반복 복호를 가정할 경우 오버래핑 된 복호의 마지막 사이클에서 소요되는 시간이다. 병렬화 지수 f_p가 2 이상을 갖는 구조는 스케줄링

표 3. 설계 변수와 복호기 구조에 따른 하드웨어 비용

Table 3. Hardware cost according to design factors and decoder configurations.

(Code length: N, Parity length: P, Code rate: (N-P)/N)

설계 변수		병렬 구조	반 병렬 구조	직렬 구조
CNPU 개수		P	$J \times f_p$	1
VNPU 개수		N	$K \times f_p$	1
LUT($\Psi(x)$) 개수		$P+N$	$(J+K)f_p$	2
반복 복호 당 클럭 수	BP 복호	2	$\frac{2m}{f_p}$	$2N$
	중첩된 BP $n(m+w)$	$n+w$	$n \times \frac{m}{f_p}+w$	$n \times N+w$
주소발생기		0	$J \times K$	$(N-K)N$
Address Decoder		0	$J \times K$	$(N-K)N$
메모리		Scatterd latches	$J \times K$ memory blocks	1 memory block

과 더불어 데이터 스위칭이 좀 더 효율적으로 이루어지도록 NPU간 데이터 통신을 위한 네트워크를 고려해야 한다. Nasa 표준의 부호화기를 이용하여 복호기를 구현한다고 할 때, 병렬구조의 경우는 단 2 cycle, 직렬구조의 경우는 한 부호어 복호에 16352 cycle 이라는 시간이 요구된다. 이 때 반병렬 구조를 적용하면 부호어 1개의 복호에 1022 cycle이 소요되는데, 적절한 비용증가를 감안하여 병렬화 지수 f_p를 높이면 설계자가 원하는 성능과 복잡도를 갖도록 복호기를 구현할 수 있다.

Ⅳ. 결 론

본 논문에서는 다양한 설계 관점에서 LDPC 복호기의 하드웨어 구현을 살펴보고, 분석하여 효율적인 복호기의 설계 방법들을 제시하였다. 오류확률을 전달하는 메시지의 양자화에 있어서, 하드웨어 복잡도와 복호 성능간의 균형을 고려하여 정수부 3비트, 소수부에 4비트를 할당한 (7,4) 양자화를 선택하였다. 채널로부터의 사전정보를 저장하는 버퍼의 구현이 수반되는 복호기인 경우에 정수부에 2, 소수부에 4비트를 할당한 (6,4) 양자화를 하여도 복호 시 성능 저하가 거의 없는 LDPC 복호기를 설계할 수 있다.

기존에 LUT 메모리로 구현되는 비선형 $\Psi(x)$ 함수를 PWL 근사방법으로 대체하여 복호기의 주요 연산 구조가 모두 조합회로로만 구현할 때 다양한 문제들을 분석하였다. 이렇게 함으로써 파이프라인 등, 구조적인 개선을 용이하게 할 수 있고 ROM 테이블을 제거할 수 있다. 오버래핑을 적용한 스케줄링은 효율적인 메시지 전달을 할 수 있을 뿐만 아니라 반복복호에 소요되는 시간을 단축하여 하드웨어 효율을 높일 수 있다.

어떤 구조로 복호기를 설계하는가에 따라서 복호기에 적용될 설계 변수들의 제한이 달라짐을 각 구조에 병렬화 지수 f_p를 적용하였을 때 필요한 하드웨어 자원을 비교하였다. 분석된 내용들을 구현할 복호기에 맞게 적절히 선택하면 적은 비용으로 우수한 성능을 갖는 QC-LDPC 복호기를 설계할 수 있다.

감사의 글

저자들은 본 연구를 위하여 설계 환경을 제공하여 준 IDEC(IC Design Education Center)에 감사드린다.

참 고 문 헌

[1] R. G. Gallager, "Low-Density Parity-Check Codes," *IRE Trans. Inform. Theory*, Vol. IT-B, pp, 21-28, Jan. 1962.

[2] J. C. Moreira and P. G. Farrell, *Essentials of Error-Control Coding*, John Wiley&Sons, 2006.

[3] David J. C. MacKay et al, "Good error-correctiong codes based on very sparse matrices," *IEEE Trans. on Inform. Theory*, Vol. 45, pp. 399-431, Mar. 1999.

[4] K. S. Andrews, S. Dolinar, C. R. Jones, F. Pollara and J. Hamkins, "The Development of Turbo and LDPC Codes for Deep-Space Applications," *Proceedings of IEEE*, Vol. 95, no. 11, Nov. 2007.

[5] Joon-Sung Kim and Hong-Yeop Song, "Concatenated LDGM Codes with Single Decoder," *IEEE Communications Letters*, Vol. 10, no. 4, April. 2006.

[6] T. J. Richardson and R. L. Urbanke, "The Capacity of Low-Density Parity-Check Codes Under Message-Passing Decoding," *IEEE Trans. Inform. Theory*, Vol. 47, pp. 599-618, Feb. 2001.

[7] Goddard Space Flight Center, GSFC-STD-9100, <http://standards.gsfc.nasa.gov>, Mar. 2008.

[8] Keshab K. Parhi, *VLSI Digital Signal Processing Systems : Design and Implementation*, John Wiley&Sons, 1999.

[9] Daesun Oh and K. K. Parhi, "Low Complexity Decoder Architecture for Low-Density Parity-Check Codes," *Journal of Signal Processing Systems*, Apr. 2008.

[10] A. J. Blanksby and Chris J. Howland, "A 690-mW 1-Gb/s 1024-b, Rate-1/2 Low-Density Parity-Check Code Decoder," *IEEE Solid-State Circuits*, Vol. 37, no. 3, Mar. 2002.

[11] Z. Wang and Z. Cui, "Low-Complexity High-Speed Design for Quasi-Cyclic LDPC Codes," *IEEE Trans. VLSI Systems*, Vol. 15, no. 1, Jan. 2007.

[12] Marjan Karkooti and Joseph R. Cavallaro, "Semi-Parallel Reconfigurable Architectures for Real-Time LDPC Decoding," in Proc. of IEEE conf. on Information Technology: Coding and Computing, Vol. 1, pp. 579-585, Apr. 2004.

저 자 소 개

정 수 경(학생회원)
2008년 가톨릭대학교 반도체시스템공학과 학사.
2008년~현재 가톨릭대학교 정보통신전자공학과 석사과정.

<주관심분야 : LDPC Decoder, VLSI설계, 임베디드 시스템 등>

박 태 근(정회원)-교신저자
1985년 연세대학교 전자공학 학사
1988년 Syracuse Univ. Computer 공학석사.
1993년 Syracuse Univ. Computer 공학박사.

1991년~1993년 Coherent Research Inc. VLSI 설계 엔지니어.
1994년~1998년 현대전자 System IC 연구소 책임연구원.
1998년~현재 가톨릭대학교 정보통신전자공학부 교수.

<주관심분야 : VLSI 설계, CAD, 병렬처리 등>