

# 저밀도 부호의 고속 부호화 알고리즘

이광재\* · 손영수\*\*

Fast Encoding Algorithm of Low Density Codes

Kwang-Jae Lee\* · Young-Su Son\*\*

요 약

본 논문에서 우리는 저밀도 부호의 부호화 문제를 고찰한다. 특히 순환 치환 행렬을 이용한 저밀도 부호의 고속 부호화 알고리즘을 제시하였다. 고속 부호화 알고리즘은 고속 Hadamard 변환과 유사하며 결과적으로 제안 기법은 저밀도 부호의 간단하며 고속화된 부호화 알고리즘임을 보여주고 있다.

ABSTRACT

In this paper, we consider the encoding scheme of low density codes. In particular, we propose the fast encoding algorithm of the low density codes using circulant permutation matrices. The fast encoding algorithm is similar to the fast Hadamard transform, and the results show that the proposed scheme is a simple and fast encoding algorithm of the low density codes.

키워드

LDC, Circulant Permutation Matrices, JIBM, Hadamard  
저밀도 부호, 순환 치환 행렬, 재킷 역 블록 행렬, 아다마르

## 1. 서 론

LDPC(Low Density Parity Check) 부호와 같은 저밀도 부호에서 다루어야 될 주요 문제 중 하나는 부호화 복잡도를 개선하는 것이며, 구조적 LDPC(structured-LDPC) 부호군과 근사 하삼각 분해 방법 등이 잘 알려져 있다[1-3]. 구조적 LDPC 부호는 패리티 검사 행렬 구조에 제한을 두어서 부호화 계산량을 줄이는 방법으로서 이중 대각 행렬(dual-diagonal matrix) 형태와 QC(Quasi-Cyclic) 부호 등이 있다. 이중 대각 구조의 부호는 LDPC 패리티 검사 행렬에서 패리티에 해당하는 부분이 이중 대각

행렬 형태가 되면 간단한 부호화 기법을 사용하여 부호화를 할 수 있다는 장점을 이용한다. QC 부호는 선형 블록 부호로서 부호어를 2 이상 오른쪽 혹은 왼쪽으로 순환 이동시켜 생성한 부호어도 역시 부호어가 되는 부호이다. QC-LDPC 부호는 이런 순환 행렬의 특징을 이용하여 효율적인 부호화를 가능하게 한다[4-5].

한편, 리차드슨은 LDPC 부호의 근사 하삼각 분해에 근거하여 효율적인 일반적 부호화 알고리즘을 소개했다. 이러한 리차드슨 형태의 계산은 여러 서브 행렬의 연산 과정들에 근거한다[6]. 그러나 패리티 검사 행렬의 선택이 적절하게 되지 않으면 변환과 행렬 처

\* 한려대학교 경영학과(kjlee3@naver.com)

\*\* 교신저자(corresponding author) : 한려대학교 경영학과(yss5733@naver.com)

접수일자 : 2014. 01. 07

심사(수정)일자 : 2014. 03. 10

게재확정일자 : 2014. 04. 11

리과정의 복잡도는 더 증가할 수 있다.

본 논문에서는 재킷 행렬 및 재킷 역 블록 행렬로 구성된 저밀도 부호(LDC:Low Density Codes)의 고속 부호화 알고리즘을 고찰한다. 저밀도 부호는 패리티 검사 행렬의 원소 중에서 "1"의 수가 부호 길이의 1/2 이하인 이진 부호이며, 재킷 행렬 및 재킷 역 블록 행렬로 구성된 저밀도 부호는 기본 재킷 행렬과 연속적으로 감소하는 차수를 갖는 계수 행렬간의 곱, 즉 순환 치환 행렬의 형태가 된다. 이 구성은 재킷 행렬의 명확한 블록 순회 저밀도 행렬 분해(block circular sparse matrices factorization)를 제공한다. 또한 대부분의 구조화된 LDPC 부호들 역시 재킷 행렬 및 재킷 역 블록 행렬로 나타낼 수 있다[7-8].

본 논문의 구성은 다음과 같다. 먼저 2장에서 재킷 행렬의 이의 특성을 간략히 소개하고, 3장에서는 재킷 역 블록 행렬의 분해 및 고속 알고리즘에 대해, 그리고 4장에서 이에 따른 결론을 기술한다.

## II. 재킷 행렬

$m \times n$  크기의 재킷 행렬  $[J]_{m \times n}$ 에서, 이의 원소는  $[J]_{m \times n} = [L_{i,j}]_{m \times n}$ ,  $i = \{0,1,\dots,m\}$ ,  $j = \{0,1,\dots\}$ 이며, 이 행렬의 역 형태는 단지 각 원소 단위의 역(inverse)과 전치(transpose) 즉,  $([J]_{m \times n})^{-1} = ([1/L_{i,j}]_{m \times n})^T$ 의 관계를 갖는다. 여기서  $(x)^T$ 는  $(x)$ 의 전치를 나타내며, 정규화 상수는 생략하였다.

이를 벡터 공간으로 표시하면 다음과 같다.

$$[J]_{m \times n} = \begin{bmatrix} L_0 \\ L_1 \\ \vdots \\ L_{m-1} \end{bmatrix}, [J]_{m \times n}^{-1} = \frac{1}{C} \begin{bmatrix} 1/L_0 \\ 1/L_1 \\ \vdots \\ 1/L_{m-1} \end{bmatrix} \quad (1)$$

여기서, C는 정규화 상수,  $L_i = (L_0, L_1, \dots, L_j)$ ,  $1/L_i = (1/L_0, 1/L_1, \dots, 1/L_{n-1})$ 이다.

$m \times n$  크기의 행렬  $[JB]_{m \times n}$ ,  $i \in \{0,1,2,\dots,m-1\}$ ,  $j \in \{0,1,2,\dots,n-1\}$ 에서  $[JB]_{m \times n} = [Lb_{ij}]_{m \times n}$ 이 성립하는 경우를 고려한다. 부 행렬  $LB_{ij} = [lb_{gh}]_{k \times k}$

는  $k \times k$  정방 행렬,  $g, h \in \{0,1,\dots,k-1\}$ 이며, 원소  $lb_{gh}$ 는 유한군  $G_n$ 을 형성한다.  $[JB]_{m \times n}$ 의 역 행렬은 블록의 역으로부터 용이하게 얻을 수 있다. 즉,  $[JB]_{m \times n}^{-1} = ([Lb_{ij}]_{m \times n})^T$ 와 같으며, 정규화 상수는 생략하였다.

예를 들어, 재킷 행렬을 다음과 같이 쓸 수 있다.

$$[J]_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & i & -1 \\ 1 & i & -i & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}, ([J]_4)^{-1} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -i & -1 \\ 1 & -i & i & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \quad (2)$$

여기서, 군  $G_4$ 에 속하는  $\{1, i, -1, -i\}$ 를 가지며,  $i = \sqrt{-1}$ 이다. 이제 관련된 작은 재킷 행렬로서의 원소를  $g \in \{\pm 1, \pm i\}$ 로 정의하면, 앞서의 경우와 동일한 관계를 갖게 되므로 이렇게 구성된 행렬을 재킷 역 블록 행렬(JIBM ; Jacket inverse block matrices)이라 한다[7-8].

## III. JIBM의 분해 및 고속 알고리즘

기존의 고속 알고리즘 및 행렬 분해와는 다르게 치환 행렬 상의 재킷 역 블록 행렬의 설계는 곱셈 군  $G_p$ 와 관련된 경우이며, 여기서 p는 소수이다. 이 경우, 일반적으로 덧셈은 존재하지 않기 때문에 일반적인 DFT(discrete Fourier transform) 또는 Hadamard 분해와 유사한 몇몇 연산은 가능하지 않다[9-10].

먼저,  $\{\alpha^0, \alpha^1\}$ 로 이루어진  $G_2$ 상에서의 JIBM을 고찰한다. 여기서,  $\alpha^2 = 1$ 과  $\alpha$ 는 복소 단위  $\exp(\frac{2\pi i}{p})$ 로 매핑할 수 있음을 고려하면,  $\alpha = -1$ , 군  $\{\alpha^0, \alpha^1\}$ 는  $\{1, -1\}$ 과 관련된다. 따라서 JIBM은 다음과 같은 Hadamard 행렬로 볼 수 있다.

$$J = \begin{bmatrix} \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 \end{bmatrix} \Delta \begin{bmatrix} (-1)^0 & (-1)^0 \\ (-1)^0 & (-1)^1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (3)$$

또한, 고차인 경우에는 다음과 같으며,

$$J_N = J_{N/2} \otimes J = J_{N/2} \otimes \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (4)$$

이는 고속 Hadamard 변환과 동일한 것이다. 따라서 행렬 분해 알고리즘은 다음과 같이 쓸 수 있다.

$$\begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^0 & \alpha^1 \\ \alpha^0 & \alpha^0 & \alpha^1 & \alpha^1 \\ \alpha^0 & \alpha^1 & \alpha^1 & \alpha^0 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \quad (5)$$

이를 좀 더 상세하게 나타내면 다음과 같다.

$$\begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^0 & \alpha^1 \\ \alpha^0 & \alpha^0 & \alpha^1 & \alpha^1 \\ \alpha^0 & \alpha^1 & \alpha^1 & \alpha^0 \end{bmatrix} = \begin{bmatrix} \alpha^0 & \alpha^0 & & \\ & \alpha^0 & \alpha^1 & \\ & & \alpha^0 & \alpha^0 \\ & & & \alpha^0 \end{bmatrix} \begin{bmatrix} \alpha^0 & & & \\ & \alpha^0 & & \\ & & \alpha^0 & \\ & & & \alpha^0 \end{bmatrix} \begin{bmatrix} \alpha^0 & \alpha^0 & & \\ & \alpha^0 & \alpha^1 & \\ & & \alpha^0 & \alpha^0 \\ & & & \alpha^0 \end{bmatrix} \quad (6)$$

$$= \left( [I] \otimes \begin{bmatrix} \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 \end{bmatrix} \right) \left( \alpha^0 [I] \right) \left( \begin{bmatrix} \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 \end{bmatrix} \otimes [I] \right)$$

팩터 그래프(factor graph)는 그림 1과 같이 나타낼 수 있다.

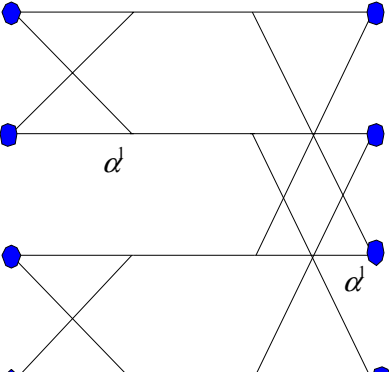


그림 1. 크기 4인 JIBM의  $G_2$  상에서의 고속 부호화 팩터 그래프

Fig. 1 The factor graph of the fast encoding of the  $G_2$ , JIBM with size 4

이러한 구성의 일반식은 다음과 같이 쓸 수 있다.

$$J_N = \left( [I]_{N/2} \otimes \begin{bmatrix} \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 \end{bmatrix} \right) \left( \alpha^0 [I]_N \right) \left( \begin{bmatrix} \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 \end{bmatrix} \otimes [I]_{N/2} \right) \quad (7)$$

군  $G_3$ 에 대한 JIBM은 다음과 같다.

$$J_3 = \begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^2 \\ \alpha^0 & \alpha^2 & \alpha^1 \end{bmatrix} \quad (8)$$

단지 곱셈 군(multiplication group)만이 사용될 경우 이 행렬이 분해될 수 없음은 분명하다. 이러한 문제를 해결하기 위해 모두  $\alpha^0$ 로 이루어진 열을 몇 개 추가하면 다음과 같은 변형된 형태를 얻게 된다.

$$MJ_3 = \begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^2 & \alpha^0 \\ \alpha^0 & \alpha^2 & \alpha^1 & \alpha^0 \\ \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \end{bmatrix} \quad (9)$$

이는 다음과 같이 나타낼 수 있다.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^2 & \alpha^0 \\ \alpha^0 & \alpha^2 & \alpha^1 & \alpha^0 \\ \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \end{bmatrix} = \begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^2 & \alpha^0 \\ \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^2 & \alpha^1 & \alpha^0 \end{bmatrix} \quad (10)$$

$$\begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^2 & \alpha^0 \\ \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^2 & \alpha^1 & \alpha^0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^0 & \alpha^2 \\ \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^2 & \alpha^0 & \alpha^1 \end{bmatrix} \quad (11)$$

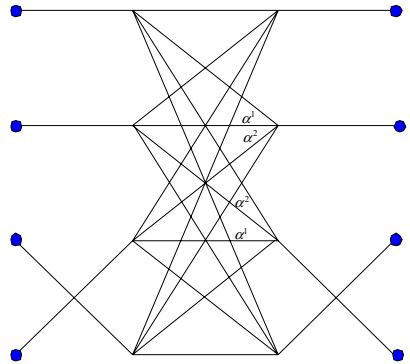


그림 2. 크기 3인 JIBM의  $G_3$  상에서의 4점 고속 부호화

Fig. 2 4-point fast encoding of JIBM with size 3 over  $G_3$

따라서, 변형된 재킷 행렬은 다음과 같이 된다.

$$MJ_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^0 & \alpha^2 \\ \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^2 & \alpha^0 & \alpha^1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (12)$$

4점의 경우에 대해 그림 2와 같이 도시할 수 있다.

또한, 이 행렬을 2점 버터플라이 형태로 나타내면 다음과 같다.

$$\begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^0 & \alpha^2 \\ \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^2 & \alpha^0 & \alpha^1 \end{bmatrix} = \begin{bmatrix} \alpha^0 & \alpha^0 & 0 & 0 \\ \alpha^0 & \alpha^1 & 0 & 0 \\ 0 & 0 & \alpha^0 & \alpha^0 \\ 0 & 0 & \alpha^0 & \alpha^1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & \alpha^0 & \alpha^0 \\ 0 & 0 & \alpha^0 & \alpha^2 \\ \alpha^0 & \alpha^0 & 0 & 0 \\ \alpha^0 & \alpha^2 & 0 & 0 \end{bmatrix} \\ = \begin{bmatrix} \alpha^0 & \alpha^0 & 0 & 0 \\ \alpha^0 & \alpha^1 & 0 & 0 \\ 0 & 0 & \alpha^0 & \alpha^0 \\ 0 & 0 & \alpha^0 & \alpha^1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha^0 & \alpha^0 & 0 & 0 \\ \alpha^0 & \alpha^2 & 0 & 0 \\ 0 & 0 & \alpha^0 & \alpha^0 \\ 0 & 0 & \alpha^0 & \alpha^2 \end{bmatrix} \quad (13)$$

따라서, 변형된 재킷 행렬은 다음과 같이 된다.

$$MJ_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^0 & \alpha^2 \\ \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^2 & \alpha^0 & \alpha^1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \\ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \left( \begin{bmatrix} \alpha^0 & \alpha^0 & 0 & 0 \\ \alpha^0 & \alpha^1 & 0 & 0 \\ 0 & 0 & \alpha^0 & \alpha^0 \\ 0 & 0 & \alpha^0 & \alpha^1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha^0 & \alpha^0 & 0 & 0 \\ \alpha^0 & \alpha^2 & 0 & 0 \\ 0 & 0 & \alpha^0 & \alpha^0 \\ 0 & 0 & \alpha^0 & \alpha^2 \end{bmatrix} \right) \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (14)$$

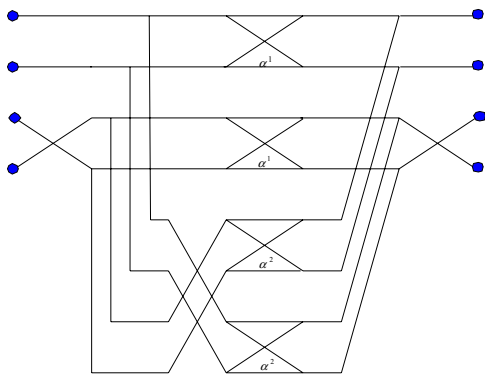


그림 3. 크기 3의 JIBM을 위한 2점 고속 부호화  
Fig. 3 2-point fast encoding for JIBM with size 3

식 (14)에서 4점의 경우를 두 개의 유사한 부분으로 분해하였으며, 그림 3과 같이 도시할 수 있다.

앞에서 언급한 것처럼 크기 3의 고속 부호화는 크기 4인 행렬을 사용한다. 따라서 출력 결과는 크기 3으로 변경되어야 하며, 이를 실현하기 위한 간단한 방법은 다음과 같다.

$$\begin{bmatrix} x_0 & x_1 & x_2 \end{bmatrix} \times \begin{bmatrix} d & d & d \\ d^0 & d^1 & d^2 \\ d^0 & d^2 & d^1 \\ d^0 & d^2 & d^1 \end{bmatrix} = \begin{bmatrix} d^0x_0+d^1x_1+d^2x_2 & d^0x_0+d^1x_1+d^2x_2 & d^0x_0+d^1x_1+d^2x_2 \end{bmatrix} \quad (15)$$

처음 세 개의 결과는 크기 3의 경우와 동일하게 된다.

$$\begin{bmatrix} x_0 & x_1 & x_2 \end{bmatrix} \times \begin{bmatrix} d^0 & d^0 & d^0 \\ d^0 & d^1 & d^2 \\ d^0 & d^2 & d^1 \end{bmatrix} = \begin{bmatrix} d^0x_0+d^1x_1+d^2x_2 & d^0x_0+d^1x_1+d^2x_2 & d^0x_0+d^1x_1+d^2x_2 \end{bmatrix} \quad (16)$$

$J_3$ 에 근거한 고차인 경우를 고려하면 다음과 같이 된다.

$$\begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^2 \\ \alpha^0 & \alpha^2 & \alpha^1 \end{bmatrix} \otimes \begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^2 \\ \alpha^0 & \alpha^2 & \alpha^1 \end{bmatrix} \\ = \begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^2 & \alpha^0 & \alpha^1 & \alpha^2 & \alpha^0 & \alpha^1 & \alpha^2 \\ \alpha^0 & \alpha^2 & \alpha^1 & \alpha^0 & \alpha^2 & \alpha^1 & \alpha^0 & \alpha^2 & \alpha^1 \\ \alpha^0 & \alpha^0 & \alpha^0 & \alpha^1 & \alpha^1 & \alpha^1 & \alpha^2 & \alpha^2 & \alpha^2 \\ \alpha^0 & \alpha^1 & \alpha^2 & \alpha^1 & \alpha^2 & \alpha^0 & \alpha^2 & \alpha^0 & \alpha^1 \\ \alpha^0 & \alpha^2 & \alpha^1 & \alpha^1 & \alpha^0 & \alpha^2 & \alpha^2 & \alpha^1 & \alpha^0 \\ \alpha^0 & \alpha^0 & \alpha^0 & \alpha^2 & \alpha^2 & \alpha^2 & \alpha^1 & \alpha^1 & \alpha^1 \\ \alpha^0 & \alpha^1 & \alpha^2 & \alpha^2 & \alpha^0 & \alpha^1 & \alpha^1 & \alpha^2 & \alpha^0 \\ \alpha^0 & \alpha^2 & \alpha^1 & \alpha^2 & \alpha^1 & \alpha^0 & \alpha^1 & \alpha^0 & \alpha^2 \end{bmatrix} \\ = \begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^0 & 0 & 0 & \alpha^0 & 0 & 0 \\ \alpha^0 & \alpha^1 & \alpha^2 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^0 & 0 & 0 & \alpha^0 & 0 & 0 \\ \alpha^0 & \alpha^2 & \alpha^1 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^0 & 0 & 0 & \alpha^0 & 0 & 0 \\ 0 & 0 & 0 & \alpha^0 & \alpha^0 & \alpha^0 & 0 & 0 & 0 & \alpha^0 & 0 & 0 & \alpha^2 & 0 & 0 \\ 0 & 0 & 0 & \alpha^0 & \alpha^1 & \alpha^2 & 0 & 0 & 0 & \alpha^0 & 0 & 0 & \alpha^1 & 0 & 0 \\ 0 & 0 & 0 & \alpha^0 & \alpha^2 & \alpha^1 & 0 & 0 & 0 & \alpha^0 & 0 & 0 & \alpha^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & 0 & 0 & \alpha^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha^0 & \alpha^1 & \alpha^2 & \alpha^0 & 0 & 0 & \alpha^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha^0 & \alpha^2 & \alpha^1 & \alpha^2 & 0 & 0 & \alpha^2 & 0 & 0 \end{bmatrix}$$

$$= [I] \otimes \begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^2 \\ \alpha^0 & \alpha^2 & \alpha^1 \end{bmatrix} \left( \begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^2 \\ \alpha^0 & \alpha^2 & \alpha^1 \end{bmatrix} \otimes [I] \right) \quad (17)$$

이의 팩터 그래프는 그림 4와 같이 나타낼 수 있다.

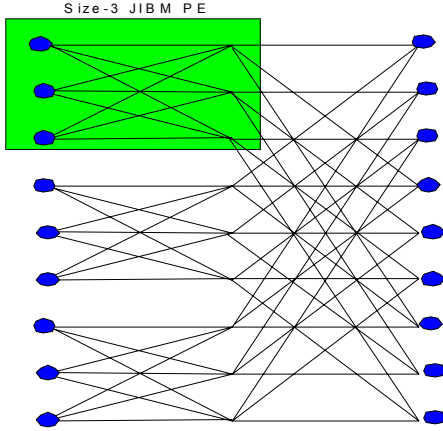


그림 4. 크기 3의 고차 JIBM의 구현

Fig. 4 Higher order of the size-3 JIBM implementation

행렬 크기가  $6 = 2 \times 3$ 인 경우의 재킷 행렬에 의한 LDC는 다음과 같다.

$$J_6 = J_2 \otimes J_3 = \begin{bmatrix} \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 \end{bmatrix} \otimes \begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^2 \\ \alpha^0 & \alpha^2 & \alpha^1 \end{bmatrix}$$

$$= \begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^2 & \alpha^0 & \alpha^1 & \alpha^2 \\ \alpha^0 & \alpha^2 & \alpha^1 & \alpha^0 & \alpha^2 & \alpha^1 \\ \alpha^0 & \alpha^0 & \alpha^0 & \alpha^1 & \alpha^1 & \alpha^1 \\ \alpha^0 & \alpha^1 & \alpha^2 & \alpha^1 & \alpha^2 & \alpha^0 \\ \alpha^0 & \alpha^2 & \alpha^1 & \alpha^1 & \alpha^0 & \alpha^2 \end{bmatrix}$$

$$\begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^2 & \alpha^0 & \alpha^1 & \alpha^2 \\ \alpha^0 & \alpha^2 & \alpha^1 & \alpha^0 & \alpha^2 & \alpha^1 \\ \alpha^0 & \alpha^0 & \alpha^0 & \alpha^1 & \alpha^1 & \alpha^1 \\ \alpha^0 & \alpha^1 & \alpha^2 & \alpha^1 & \alpha^2 & \alpha^0 \\ \alpha^0 & \alpha^2 & \alpha^1 & \alpha^1 & \alpha^0 & \alpha^2 \end{bmatrix} = \begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 & 0 & 0 & 0 \\ \alpha^0 & \alpha^1 & \alpha^2 & 0 & 0 & 0 \\ \alpha^0 & \alpha^2 & \alpha^1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha^0 & \alpha^0 & \alpha^0 \\ 0 & 0 & 0 & \alpha^0 & \alpha^1 & \alpha^2 \\ 0 & 0 & 0 & \alpha^0 & \alpha^2 & \alpha^1 \end{bmatrix} \begin{bmatrix} \alpha^0 & 0 & 0 & 0 & 0 & 0 \\ \alpha^0 & 0 & 0 & 0 & 0 & 0 \\ \alpha^0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha^1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha^1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha^1 \end{bmatrix} \quad (18)$$

이의 팩터 그래프는 그림 5와 같이 나타낼 수 있다.

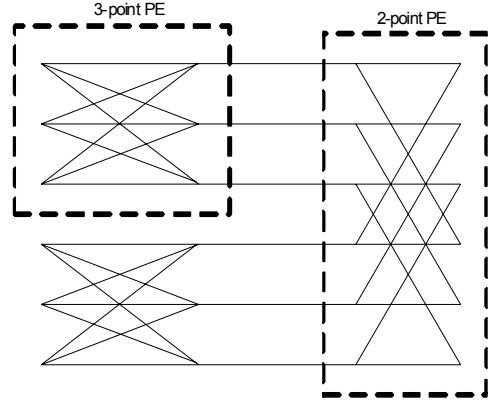


그림 5. 고차 LDC 행렬의 고속 부호화

Fig. 5 The higher order LDC matrix fast encoding

표 1. 제안 고속 부호화의 계산량

Table 1. Computations of the proposed fast encoding

	Conventional	Proposed $N=2^n$	Proposed $N=3^n$
Additions	$N \times N$	$N \log_2 N$	$(N+1) \log_2 (N+1)$
Multiplications	$N \times N$	$\frac{1}{2}(N \log_2 N)$	$\frac{1}{2}((N+1) \log_2 (N+1))$

#### IV. 결론

본 논문에서는 순환 치환 행렬과 JIBM을 이용한 저밀도 부호의 간단한 고속 부호화 알고리즘을 구성하였다.

제안 부호화 기법에서의 연산 특성을 표 1에 나타내었으며, 군  $G_2$ 의 경우,  $N=2^n$  크기의 행렬은  $N \log_2 N$ 의 덧셈과  $\frac{1}{2}(N \log_2 N)$ 의 곱셈을 필요로 하며, 군  $G_3$ 의 경우,  $N=3^n$  크기의 행렬은  $(N+1) \log_2 (N+1)$ 의 덧셈과  $\frac{1}{2}((N+1) \log_2 (N+1))$ 의 곱셈을 필요로 한다. 제안 고속 부호화 알고리즘은 고속 Hadamard 변환과 유사하며 결과적으로 제안 기법

은 저밀도 부호의 간단하며 고속화된 부호화 알고리즘을 보여주고 있다.

참고 문헌

[1] R. G. Gallager, "Low density parity check codes," *PhD thesis, MIT, Cambridge, Mass., Sept., 1960.*

[2] R. G. Gallager, *Low density parity check codes. Number 21 in research monograph series.* MIT Press, Cambridge, Mass., 1963.

[3] Y. Liu and H.-J. Suh, "A Comparison of Raptor Code Using LDGM and LDPC code," *J. of The Korea Institute of Electronic Communication Sciences*, vol. 8, no. 1, 2013, pp. 65-70.

[4] K.-J. Lee, "A design of ALT LDPC codes using circulant permutation matrices," *J. of The Korea Institute of Electronic Communication Sciences*, vol. 7, no. 1, 2012, pp. 117-124.

[5] K.-J. Lee and S.-S. Kang, "An F-LDPC codes based on Jacket pattern," *J. of The Korea Institute of Electronic Communication Sciences*, vol. 7, no. 2, 2012, pp. 317-325.

[6] T. J. Richardson and R. L. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Trans. JT*, vol. 47, Feb. 2001, pp. 638-656.

[7] M. Lee, B. S. Rajan, and J. Park, "A generalized reverse jacket transform," *IEEE Trans. Circuits Syst. II*, vol. 48, no. 7, 2001, pp. 684-690.

[8] W.-P. Ma and M. Lee, "Fast reverse jacket transform algorithms," *IEE Electronics Letters*, vol. 39, no. 18, Sept. 2003, pp. 1312-1313.

[9] M.-R. Moon, H. Jia, G.-Y. Hwang, M.-H. Lee, and K.-J. Lee, "Low density codes construction using Jacket matrices," *J. of The Institute of Electronics Engineers of Korea*, vol. 42, no. 8, 2005, pp. 517-526.

[10] N. Ahmed and K. R. Rao, *Orthogonal transforms for digital signal processing.* Berlin, Germany : Spring-Verlag, 1975.

저자 소개



**이광재(Kwang-Jae Lee)**

1986년 전북대학교 전자공학과 졸업 (공학사)

1990년 전북대학교 대학원 전자공학과 졸업(공학석사)

2006년 전북대학교 대학원 전기공학과 졸업(공학박사)

1995년~현재 한려대학교 경영학과 교수

※ 관심분야 : 부호이론, 이동통신



**손영수(Young-su Son)**

1988년 조선대학교 전산통계학과 졸업 (이학사)

1991년 조선대학교 전산통계학과 졸업 (이학석사)

1999년 조선대학교 전산통계학과 졸업(이학박사)

1996년~현재 한려대학교 경영학과 교수

※ 관심분야 : 프로그래밍어론, 데이터베이스, 인공지능, 퍼지제어 등