

# Single Parity Check 부호를 적용한 3차원 Turbo Product 부호의 효율적인 복호 알고리즘

하 상 철\*, 안 병 규\*, 오 지 명\*\*, 김 도 경\*\*, 허 준°

## Effective Decoding Algorithm of Three dimensional Product Code Decoding Scheme with Single Parity Check Code

Sang-chul Ha\*, Byung-kyu Ahn\*, Ji-myung Oh\*\*, Do-kyoung Kim\*\*, Jun Heo°

### 요 약

본 논문에서는 single parity check 부호(SPC)를 포함하는 3차원 turbo product 부호(TPC)의 효율적인 복호 기법을 제안한다. 일반적으로 TPC의 부호율을 극대화하기 위한 목적으로 부호 길이가 짧은 축에서 SPC 부호를 적용한다. 그러나 SPC 부호가 오류 정정 능력이 없는 부호이기 때문에 3차원 TPC를 Chase-Pyndiah 복호 알고리즘만으로 복호할 경우, 2차원 TPC에 비하여 성능 개선이 거의 발생하지 않는다. 본 논문에서는 이를 개선하기 위해 다음의 2가지 기법을 복호 과정에 적용하였다. 우선 SPC 부호로 이루어진 축에서는 구현 복잡도를 낮추기 위하여 min\*-sum 알고리즘을 복호 방법으로 적용하였으며, 반복 복호 방식으로는 성능 개선을 위해 직렬 복호 방식을 변형한 방식을 이용하였다. 마지막으로 이를 적용한 TPC 시뮬레이터의 성능을 비교 분석하고, 실제 하드웨어 구현 과정에서 고려해야 할 부분을 소개한 후, VHDL을 이용하여 3차원 TPC를 설계하였다.

**Key Words** : Turbo Product code, Chase-Pyndiah algorithm, Single Parity Check, Iterative decoding, FPGA implementation

### ABSTRACT

In this paper, we propose a decoding scheme that can apply to a three dimensional turbo product code(TPC) with a single parity check code(SPC). In general, SPC is used an axis with shortest code length in order to maximize a code rate of the TPC. However, SPC does not have any error correcting capability, therefore, the error correcting capability of the three-dimensional TPC results in little improvement in comparison with the two-dimensional TPC. We propose two schemes to improve performance of three dimensional TPC decoder. One is min\*-sum algorithm that has advantages for low complexity implementation compared to Chase-Pyndiah algorithm. The other is a modified serial iterative decoding scheme for high performance. In addition, the simulation results for the proposed scheme are shown and compared with the conventional scheme. Finally, we introduce some practical considerations for hardware implementation.

※ 본 연구는 2015.06~2016.05 LIG-Nex1의 지원을 받아 수행되었습니다.[과제명: Turbo Product Code(Turbo Block Code) 설계 및 구현]

◆ First Author : Korea University school of electrical engineering, cjf1230@korea.ac.kr, 학생회원

° Corresponding Author : Korea University school of electrical engineering, junheo@korea.ac.kr, 종신회원

\* Korea University school of electrical engineering, bk440@korea.ac.kr, 학생회원

\*\* LIG nex1 communication R&D Lab, jimyungoh@lignex1.com, 정회원

\*\* LIG nex1 communication R&D Lab, kimdokyoung@lignex1.com, 정회원

논문번호 : KICS2016-04-081, Received April 30, 2016; Revised July 27, 2016; Accepted August 4, 2016

## I. 서 론

Turbo Product 부호(TPC)는 1954년 Elias 에 의해 product 부호의 개념이 소개된 것을 바탕으로 연구가 시작되었다. 하지만 초창기 product 부호의 복호 알고리즘이 경 입력 경 출력(HIHO) 알고리즘이어서 낮은 성능, 복잡도 등의 이유로 많이 활용되지 못하였다.

1998년 Pyndiah<sup>[1]</sup>는 이러한 TPC의 경 입력 경 출력으로 인해 야기되는 낮은 성능의 단점을 극복하고, 우수한 성능을 가질 수 있는 Chase -Pyndiah 알고리즘<sup>[2]</sup>을 제안하였다. 이 알고리즘을 바탕으로 Pyndiah 는 연 입력 연 출력(SISO) 알고리즘을 적용하고, 복호 과정에 있어서 반복 복호 알고리즘을 도입하였다. 이렇게 구성된 TPC는 비교적 짧은 길이의 블록 부호들을 고차원 형태로 결합함으로써 최소거리(Minimum distance)가 각 부호의 최소거리의 곱 형태가 되는 부호를 만들어 낼 수 있으며, 복호 과정 중 검사 배열 집합을 구할 때 전역 탐색을 하지 않고 신뢰도가 가장 낮은 p개의 위치만을 사용하므로 상대적으로 낮은 복잡도의 구현이 가능해졌다.

TPC의 구성에는 Hamming, BCH, RS, SPC 부호<sup>[3]</sup> 등 다양한 종류의 블록 부호가 사용될 수 있으며, 사용하는 선형 블록 부호의 개수에 따라 차원의 확장이 가능하다. 이와 관련하여 2007년 Xiaoxiao Wu<sup>[4]</sup>는 BCH 부호를 이용한 기존 2차원 TPC를 3차원으로 확장할 때, 복호 성능을 개선할 수 있는 3차원 TPC의 반복 복호 기법을 제시하였다. 그 중 SPC 부호는 3차원 TPC를 구성할 때 부호율을 높이기 위한 목적으로 상대적으로 길이가 짧은 차원에서 주로 이용된다. 하지만 이와 같이 다차원 TPC의 복호 과정에서 한 축에 SPC를 사용한 경우의 복호 방식에 대해서는 아직 명확하게 정의된 형식을 찾을 수 없었다. 또한, 기존의 Chase-Pyndiah 알고리즘을 3차원 TPC 복호 과정의 모든 축에 적용하면 SPC 부호가 오류정정 능력이 없는 부호이기 때문에 SPC 부호를 이용하여 3차원 복호기를 구성하여도 차원의 증가로 인한 복호 성능의 향상을 기대하기 어렵다.

따라서 본 논문에서는 SPC 부호를 적용한 3차원 TPC의 복호 성능을 개선하기 위한 두 가지 방법을 제시한다. 우선 SPC 부호를 적용하여 구성된 길이가 짧은 축의 복호 과정에 min\*-sum 알고리즘을 적용한다. 다음으로 반복 복호 방식으로 기존 직렬 반복 복호 과정을 변형한 방식을 적용하여 3차원 TPC를 구성하는 방법을 제시한다. 추가적으로 제안 알고리즘을 VHDL 을 통해 하드웨어를 구현하고, 설계 및 구현 과정에서

고려해야 할 사항을 제시한다. 마지막으로 시뮬레이션을 통한 성능을 기존 기법과 비교 분석한다.

본 논문의 구성은 다음과 같다. II장에서는 TPC의 대표적인 복호 알고리즘인 Chase -Pyndiah 알고리즘과 제안한 3차원 TPC에서 적용할 수 있는 min\*-sum 알고리즘 그리고 반복 복호 기법에 대하여 설명한다. 뒤이어 III장에서는 제안한 3차원 TPC 기반의 시뮬레이션 결과와 하드웨어 구현을 위한 설계 방법을 제시한다. 마지막 IV장에서는 최종 결론을 맺는다.

## II. 본 론

이 장에서는 X축과 Y축은 각각  $(n_1, k_1, h_1)$ ,  $(n_2, k_2, h_2)$  확장 해밍 부호<sup>[5]</sup>를 사용하고 Z축은  $(n_3, k_3, h_3)$  single parity check 부호를 기반으로 하여 구성된 3차원 TPC의 복호 방법에 대해 알아보고, 복호 성능을 개선할 수 있는 2가지 기법을 소개한다. 여기서  $n_i$ 는 부호어 길이,  $k_i$ 는 정보 비트 길이,  $h_i$ 는 최소 해밍 거리를 의미하며 밑 첨자  $i$ 는 차원을 의미한다.

### 2.1 TPC의 Chase-Pyndiah 복호 알고리즘

일반적인 2차원 TPC의 복호 과정에는 전송 중에 오류가 포함되어 수신된 부호어 중에서 확률이 가장 높은 일부를 후보군으로 정하여 복호하는 Chase-Pyndiah 알고리즘이 주로 사용된다. 제안한 3차원 TPC의 X축과 Y축의 복호 과정 또한 Chase-Pyndiah 알고리즘을 적용하였으며 다음과 같은 순서로 복호 과정이 진행된다.

- 1) 수신신호를 R이라 할 때, 이 값을 바탕으로 경 판정을 진행한다. 또한 수신된 값들 중 신뢰도가 가장 낮은 값 p개의 위치를 수신정보의 LLR(Log Likelihood Ratio) 값을 바탕으로 선택한다.
- 2) p개의 신뢰도가 낮은 비트의 위치에 1이 위치하도록 하여  $2^p$ 개의 테스트패턴을 생성한다.
- 3) 경 판정 부호어와 테스트 패턴의 논리적 배타합을 통해  $2^p$ 개의 부호어를 만든다.
- 4)  $2^p$ 개의 부호어를 신드롬 복호하여  $2^p$ 개의 새로운 부호어 집단  $\Omega$ 을 형성한다.
- 5) 구해진 부호어 집단  $\Omega$ 가  $2^p$ 개의 부호어를 가지고 있을 때, 각각의 부호어를  $C^i (i = 1, 2, 3, \dots, 2^p)$ 라고 하고,  $C^i$  중 수신신호 R과의 유클리드 거리  $|R - C^i|^2$ 가 가장 짧은 부호어를 D,  $\Omega$ 에 속한 부호어에서  $d_j \neq c_j$ 를 만족하는 부호어 중 가장 유

표 1. 경쟁 부호어 C의 유무에 따른 외부 정보  
Table 1. Extrinsic information according to existence of competitor codeword C

If C exists	If C doesn't exist
$w_j = \left( \frac{ R - C^2 -  R - D ^2}{4} \right) d_j - r_j$	$w_j = (\beta \times d_j)$

클리드 거리가 짧은 부호어를 C라고 하면 새로운 외부 정보 W의 j번째 자리의 값인  $w_j$ 는 다음 표 1과 같이 두 가지 경우로 나누어 표현 가능하다. 여기서  $d_i$ 는 부호어 D의 i번째 자리,  $c_i$ 는 부호어 C의 i번째 자리에 해당하는 값을 의미한다. 여기서  $\beta$ 는 절반 반복 횟수 m에 따른 신뢰도를 보정하는 값으로  $\beta(m) = \{0.2, 0.4, 0.6, 0.8, 1.0, 1.0, 1.0, 1.0\}$  의 값을 가진다.

- 6) 외부 정보를 이용한 m 번째 반복 복호 과정을 위한 연성 입력의 갱신은 다음 식(1)과 같이 진행된다.

$$[R(m+1)] = [R] + \alpha(m)[W(m)] \quad (1)$$

이 때,  $[R(m+1)]$ 은 m+1번째 반복 복호 과정에서의 연성 입력 값이며,  $[R]$ 은 수신 신호,  $[W(m)]$ 은 m 번째 반복 복호에서 얻은 외부 정보, 무게 인수  $\alpha(m) = \{0.0, 0.2, 0.3, 0.5, 0.7, 0.9, 1.0, 1.0\}$ 은 반복 복호 과정의 높은 비트 오류를 보정해 주기 위한 값을 의미한다.

### 2.2 SPC부호를 적용한 3차원 TPC복호 알고리즘

이번 장에서는 2차원에서 3차원으로 TPC를 확장하는 과정에서 SPC 부호를 적용한 경우, Chase-Pyndiah 복호 알고리즘을 적용한 것에 비하여 성능을 개선할 수 있는 2가지 방법을 제안한다.

#### 2.2.1 Min\*-sum 알고리즘을 적용한 3차원 TPC 복호 기법

3차원 TPC의 나머지 두 축과 달리 상대적으로 부호어의 길이가 짧은 SPC 부호로 구성된 Z축에 대해서는 복호 방식으로 Chase-Pyndiah 알고리즘을 사용하지 않고, LDPC<sup>[6]</sup>의 복호에서 주로 사용되는 합-곱 알고리즘<sup>[7]</sup>(sum-product algorithm, SPA)을 사용한다. 합-곱 알고리즘은 변수 노드(Variable node) 갱신과 검사 노드(Check node) 갱신의 두 단계로 이루어지며, 이 때 적용되는 SPC 부호의 2분할 그래프는 그림 1과 같이 단일 검사 노드와 Z축의 부호 길이와 동일한 개수의 변수 노드를 가진다. 그리고 이 때, SPA

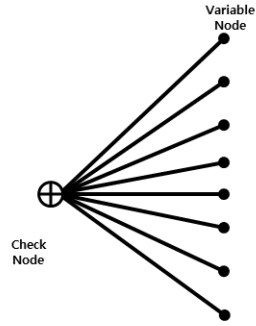


그림 1. (8,7) SPC의 2분할 그래프  
Fig 1. Bipartite graph of (8,7) SPC

에서 외부 정보를 구하는 방법으로 사용되는 검사 노드 갱신은 아래의 식(2)를 이용하여 구할 수 있다. 여기서  $w_i(m)$ 은 검사 노드에서 변수 노드로의 메시지로 m번째 복호로 나오는 i번째 자리의 외부 정보를 의미하며,  $r_j(m)$ 은 변수 노드에서 검사 노드로의 메시지로 m번째 복호 연 입력의 j번째 자리 값을 의미한다.

$$w_i(m) = 2 \times \tanh^{-1} \left[ \prod_{j \neq i} \tanh \{r_j(m)\} \right] \quad (2)$$

그러나 위의 식 (2)의 검사 노드 갱신 식은 실제 하드웨어 구현 시에는 tanh 연산의 표현의 어려움으로 인해 그대로 적용하는 것이 어렵다. 이 문제를 해결하기 위해 합-곱 알고리즘의 log-MAP을 통해 복잡도를 낮춘 min\*-sum 방식을 적용하였다. 이러한 min\*-sum 방식은 아래 식(3)과 (4)에서와 같이 검사 노드 갱신 과정에서 LUT를 사용하여 연산이 이루어진다.

$$w_i(m) = g(r_1(m), \dots, r_{i-1}(m), r_{i+1}(m), \dots, r_n(m)) \quad (3)$$

여기서 검사 노드 연산기는 다음과 같은 g()함수 연산을 통하여 검사 노드 갱신 과정을 진행한다.

$$g(a,b) = \text{sign}(a) \times \text{sign}(b) \times \{ \min(|a|, |b|) \} + LUT_g(a,b) \quad (4)$$

$$LUT_g(a,b) = \log(1 + e^{-|a+b|}) - \log(1 + e^{-|a-b|})$$

식 (4)에서 sign(a), sign(b)는 a, b의 부호에 따라 1, -1중 하나로 결정되는 부분이며,  $LUT_g(a,b)$ 는 a, b의 범위에 따라 위 식의 값과 가장 가까운 값으로 양자화 된 표를 만들어 사용한다.

2.2.2 3차원 TPC의 반복 복호 기법

3차원 TPC의 기본적인 반복 복호 기법으로는 이전 축의 복호 결과인 외부 정보를 이용하여 다음 축의 복호를 진행하는 표 2의 첫 번째 그림과 같은 직렬 방식의 복호 방식<sup>[4]</sup>이 사용된다.

이와 같은 기존 TPC의 반복 복호 방식은 모든 축을 동일한 복호 알고리즘으로 복호하는 경우, 차원의 확장으로 인한 성능 개선을 기대할 수 있다. 하지만 본 논문에서 구성한 것과 같이 길이가 짧은 한 축에서 나머지 축과는 달리 오류 정정 능력이 없는 SPC를 적용하여 반복 복호를 진행할 경우, 기존의 직렬 방식에서는 Z축의 복호 과정으로 인한 성능 개선을 기대하기 어렵다. 그러나 두 번째 그림과 같이 하나의 축의 복호 과정에 해당 축을 제외한 나머지 두 축의 복호 결과인 외부 정보를 합한 값을 이용하여 다음 복호 과정에서 적용될 입력을 생성하면 기존의 방법에 비해 성능 개선이 가능하다. 제안된 복호 기법에서의 성능 개선은 기존의 직렬 복호 기법이 이전 축의 외부 정보만을 이용해서 다음 축의 복호에 이용하는 것에 비해 나머지 두 축의 이전 복호 과정에서 사용된 외부 정보를 이용하는 것으로부터 발생한다.

이 때, 각 축의 연 판정 입력은 식(5)와 같이 구성되며  $W$ 는 각 축의 복호 과정의 결과 값,  $\alpha$ 는 무게인수를 의미한다.

$$\begin{aligned}
 R_x(m) &= R + \alpha_y(m-1) W_y(m-1) + W_z(m-1) \\
 R_y(m) &= R + \alpha_x(m) W_x(m) + W_z(m-1) \\
 R_z(m) &= R + \alpha_x(m) W_x(m) + \alpha_y(m) W_y(m)
 \end{aligned}
 \tag{5}$$

마지막으로 모든 반복 복호가 끝난 후에는 각 축의 외부 정보와 채널 정보를 모두 더한 내부 정보를 이용하여 경판정한 후에 복호를 완료한다.

III. 시뮬레이션 결과 및 하드웨어 구현

본 장에서는 X축에 (64,57,4) 확장 해밍 부호, Y축에 (32,26,4) 확장 해밍 부호, Z축에 (8,7,2) SPC를 기반으로 하는 IEEE 802.16a 표준 3차원 TPC를 구성한 후, 성능을 기존 방법과 비교 분석하였다. 또한 하드웨어 구현을 위한 설계 과정을 VHDL을 이용하여 나타내었다.

3.1 제안된 3차원 TPC의 부동소수점형 코딩 성능

3차원 전체의 복호에 Chase-Pyndiah 알고리즘만을 적용한 경우와 본 논문에서 제안한 Z축 한 차원에만 별도로 min\*-sum 알고리즘을 사용하고 직렬 반복 복호 방식을 변형하여 적용한 경우에 대해서 비트 오류 성능을 나타내면 그림 2와 같다. 이를 위해 신뢰도가 가장 낮은 비트수 p=4, 반복 복호 횟수는 4회로 고정하였다. 또한 무게 인자  $\alpha = \{0, 0.25, 0.3125, 0.5, 0.6875, 0.875, 1.0, 1.0\}$ 와 신뢰도 인자  $\beta = \{0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5\}$ 는 추후 구현에서 사용이 용이하도록 양자화하여 적용하였고, 채널 환경은 AWGN, 변조방식은 BPSK를 사용하였다.

그림 2에서와 같이 min\*-sum 알고리즘을 Z축에 적용한 제안 기법이 3차원을 모두 Chase-Pyndiah 알

표 2. 3차원 TPC의 반복 복호 기법  
Table 2. Iterative decoding scheme of 3D TPC

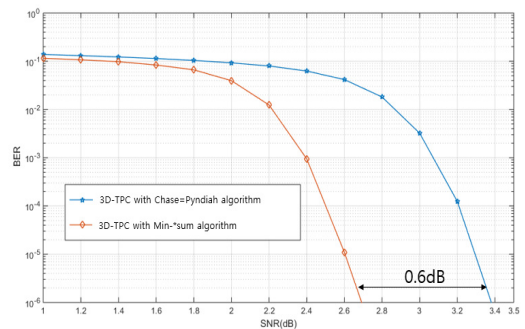
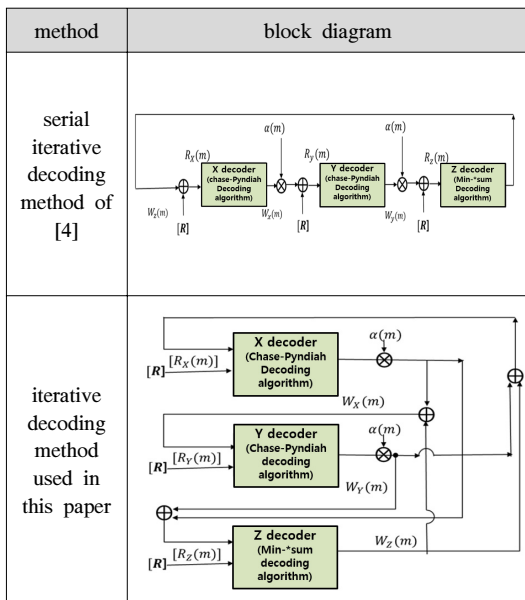


그림 2. Chase-Pyndiah 복호 기법과 min\*-sum 복호 기법을 적용한 3차원 TPC의 BER 성능 비교  
Fig. 2. Comparison between BER of 3D TPC with only Chase-Pyndiah decoding algorithm and with min\*-sum decoding algorithm

고리즘을 적용한 것에 비해 BER  $10^{-6}$  기준 0.6dB의 성능 개선이 있음을 알 수 있다.

### 3.2 하드웨어 구현을 위한 고정 소수점 형 코딩 및 성능

VHDL을 이용한 하드웨어 구현을 위해서는 실수 신호 값을 양자화 하는 과정이 필요하다. 실수 신호의 범위를 결정하기 위해 전체 사용 비트를 부호 1비트, 정수부 3비트로 고정하고 소수부를 1비트부터 부동소수점 코딩과 성능이 거의 차이가 없어지는 4비트까지 늘리면서 양자화 과정을 진행하였다. 여기서 정수부의 비트수 3은 반복 복호 4회, 신뢰도가 가장 낮은 값 4개를 기준으로 하여 정수부의 비트수를 늘리면서 시뮬레이션을 진행하여 결정하였다. 성능 결과를 확인한 결과 3비트까지는 정수 비트수를 늘리는 것에 대한 성능 개선이 발생하였으나 그 이상에서는 개선이 거의 없이 일정한 성능을 가짐을 확인할 수 있었다. 그 결과 그림 3과 같이 전체 사용 비트가 8비트 이상 일 경우에는 부동 소수점 형과 BER  $10^{-6}$  기준 0.02dB이하의 성능 차이가 발생하므로 8비트를 하드웨어 구현을 위한 비트 수로 사용하였다. 이 때, Chase -Pyndiah 복호 과정에서 결정되어야 하는 신뢰도가 가장 낮은 비트 수  $p$ 와 반복 복호 횟수는 부동 소수점 형 코딩과 동일한 값을 사용하였으며, 무계인자와 신뢰도 인자 또한 구현에 적합하도록 각각 다음의  $\alpha = \{0, 0.25, 0.3125, 0.5, 0.6875, 0.875, 1.0, 1.0\}$ ,  $\beta = \{0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5\}$ 를 이용하였다.

하지만 반복 복호 횟수가 증가할수록 복호 과정에서 사용되는 유클리드 거리 값이 수신 신호 R값의 증가와 제곱 연산으로 인해 급격히 커지는 현상이 발생하기 때문에 고정 소수점 형 코딩 시 이를 이진법으로 표현하기 위해서는 많은 비트 수가 요구되는 문제점이 발생한다. 이를 해결하기 위해 기존의 연구<sup>[8]</sup>에서

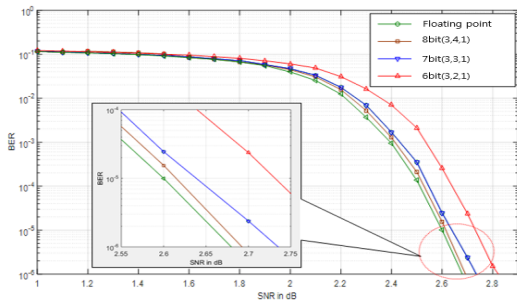


그림 3. 양자화 비트의 개수에 따른 BER 성능 비교  
Fig. 3. Comparison of BER performance with different number of quantized bits

는 제곱 연산을 제거하는 방식으로 유클리드 거리 값 연산을 진행하였으나 이 방법은 구현상의 복잡도를 줄일 수 있는 반면 성능 저하가 크게 발생하였다. 따라서 본 논문에서는 하드웨어 구현 시 성능 저하를 막기 위한 방법으로 유클리드 거리 계산 부분에서만 다른 과정과 달리 8비트 이상의 연산 범위가 필요하므로 이 부분의 값을 표현하기 위해 사용되는 비트를 14비트(정수부:12비트, 소수부:1비트, 부호:1비트)로 확장하였다. 여기서 사용된 14비트의 값은 반복 복호 4회에서 최대로 요구되는 유클리드 값을 모두 수용할 수 있는 값을 기준으로 설정하였다.

그림 4는 전체 TPC 복호기에서 가장 큰 영역을 차지하는 X축의 Chase-Pyndiah 복호기를 7비트와 8비트로 구성하였을 때의 합성결과를 비교한 결과이다. 전체 비트수가 1비트 증가할 때, 7비트 기준 복호기의 전체 Slice LUT는 4.5%, 메모리에 해당하는 Slice register는 6.5%의 증가하는 것을 확인할 수 있다. 따라서 적용하는 비트를 증가시키게 되면 메모리나 연산 영역에서 Slice 소모량이 커지므로 부동 소수점 형 코딩의 성능과 차이가 BER  $10^{-6}$  기준 0.1dB이하로 줄어드는 8비트를 하드웨어 구현에 적용하는 것이 적

1. Slice Logic Result of 7bit

Site Type	Used	Fixed	Available	UtiliX
Slice LUTs*	25018	0	254200	9.84
LUT as Logic	25018	0	254200	9.84
LUT as Memory	0	0	90600	0.00
Slice Registers	6797	0	508400	1.33
Register as Flip Flop	6797	0	508400	1.33
Register as Latch	0	0	508400	0.00
F7 Muxes	849	0	127100	0.66
F8 Muxes	227	0	63550	0.35

1. Slice Logic Result of 8bit

Site Type	Used	Fixed	Available	UtiliX
Slice LUTs*	26157	0	254200	10.28
LUT as Logic	26157	0	254200	10.28
LUT as Memory	0	0	90600	0.00
Slice Registers	7239	0	508400	1.42
Register as Flip Flop	7239	0	508400	1.42
Register as Latch	0	0	508400	0.00
F7 Muxes	1814	0	127100	1.42
F8 Muxes	231	0	63550	0.36

그림 4. 7bit, 8bit를 사용한 3D TPC의 X축 복호기 결과 비교  
Fig. 4. Comparison result of implementation performance with different number of quantized bits

절함을 알 수 있었다.

### 3.3 하드웨어 설계 및 결과

#### 3.3.1 VHDL로 구성된 3차원 TPC 복호기의 전체 구성

그림 5와 같이 반복 복호 과정에서는 세 축의 복호 과정을 우선 수행하며 그 결과인 외부 정보는 각각 X, Y, Z 출력 메모리에 저장된다. 그리고 나머지 반복 복호 과정에서는 나머지 두 축의 복호 결과가 저장되어 있는 값과 채널 값을 바탕으로 입력 부분의 메모리 저장 값을 갱신하여 나머지 반복 복호 과정을 진행한다. 총 4회의 반복 복호 과정을 진행한 후 최종 경 판정 값이 경 판정 출력 메모리에 저장되는 것을 마지막으로 복호 과정이 마무리 된다.

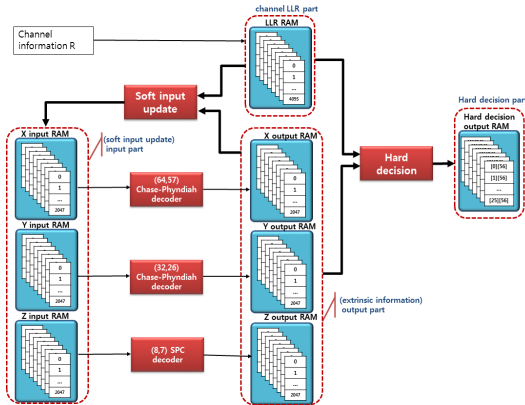


그림 5. 3차원 TPC의 전체 구성도  
Fig. 5. Block diagram of 3D TPC

#### 3.3.2 3차원 TPC의 각 축의 복호기 설계

그림 6은 X 및 Y축에 적용되는 Chase -Pyndiah 복호 과정에 대한 블록도이다. 우선 <1. Chase 복호기>에서는 경 판정 및 신뢰도를 구하는 절대 값 연산 부분, 신뢰도가 낮은 4개의 위치를 비교기를 통해 찾는 부분, 경 판정 결과와 테스트 패턴의 배타적 논리 합 연산 부분, 확장 해밍 부호의 복호 부분이 포함되어 있다. 뒤이어 <2.유클리드 거리 연산>에서는 16개의 부호어 후보군의 유클리드 거리 연산을 수행한다. 이때 결과 값은 14비트로 확장되어 램에 저장된다. 그리고 <3. 최소거리 부호어 D와 각 비트별 경쟁 부호어 C 결정>은 16개 부호어 후보군의 유클리드 거리를 비교기를 통해 토너먼트를 진행하여 가장 거리가 짧은 것을 D로 선택하고, 각 비트별로 16개의 후보군 중 D와 반대인 값을 가지는 것들 중 가장 거리가 짧은 것

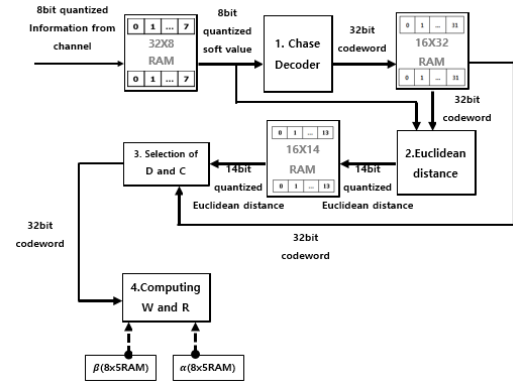


그림 6. (32,26) Chase-Pyndiah 복호기의 구조  
Fig. 6. Structure of (32,26) Chase-Pyndiah decoder

을 C로 선택한다. 마지막으로 <4. 외부 정보 W와 채널 갱신 값 R 연산 부>에서는 표 1의 식을 이용하여 TPC의 출력을 계산한다.

Z축에 적용되는 min\*-sum 복호 과정에서는 식(4)에서 정의된 LUT를 추가하는 부분의 구현이 우선 요구되며 그 식은 다음과 같다. 하지만 식

$LUT_g(a,b) = \log(1 + e^{-|a+b|}) - \log(1 + e^{-|a-b|})$ 의 구현 과정에서 로그 연산을 동일하게 구현하는 것은 불가능하기 때문에 a, b 값에 따른 LUT의 값을 구간별로 세분화하여 나눈 후 양자화 시키면 성능 저하를 최소화하는 구현이 가능하다.

LUT 연산이 완료되면 식(3)과 (4)의 나머지 부분에 해당하는 절대값 연산, 최소값 찾는 과정, 부호 결정 과정을 차례로 수행한 후 Z축에 해당하는 복호 과정이 완료된다.

#### 3.3.3 설계 결과

VHDL을 통해 구성된 3차원 (64,57,4)\* (32,26,4)\* (8,7,2) TPC 복호기의 최종 구현 결과는 다음의 표 3과 같다. 본 논문의 구현 결과 확인에 사용된 시뮬레이션 도구로는 Xilinx ISE 14.7을 이용하였으며, 목표로 하는 칩은 KIN TEX7 (XC7K410TFFG676-1)을 설정하였다.

본 논문에서 구현된 3차원 TPC 복호기의 최대 동작 주파수는 111.663MHz이며, 전체 소요 Slice 51812 중 X축과 Y축의 Chase -Pyndiah 복호기는 각각 54%(소요 Slice 크기:28328), 39%(소요 Slice 크기:20345)의 Slice LUT를 사용하여 전체 소요 Slice의 대부분을 차지하고 있으며, Z축의 복호기는 나머지 두 축에 비해 부호의 길이가 짧고 상대적으로 단순

표 3. 3차원 TPC 복호기의 구현 결과  
Table 3. Implementation result of 3D TPC decoder

	usage & size	use rate(%)
operating frequency	111.663MHz	-
used Slice	51812	20.38
used memory Slice	12754	2.5

한 형태의 min\*-sum 알고리즘을 사용하고 있기 때문에 사용되는 Slice LUT가 0.9%(소요 Slice 크기:468) 정도의 작은 영역을 차지한다. 이를 제외한 나머지 Slice는 조종부(Controller 영역)에서 차지하고 있다. 소요 메모리 Slice는 복호 과정에서 Chase-Pyndiah 복호 과정 및 min\*-sum 알고리즘에서 데이터 저장에 사용하는 면적을 의미하며 12754 Slice를 차지한다.

#### IV. 결 론

본 논문에서는 SPC 부호를 적용하여 3차원 TPC를 구성할 때 성능을 개선할 수 있는 방법을 제시하였다. 이를 위해 SPC 부호를 적용한 축에서 LDPC의 min\*-sum 알고리즘을 사용하여 복호 알고리즘을 구성하고, 각 축의 외부 정보와 내부 정보를 개선된 직렬 반복 복호 기법을 사용하여 구성함으로써 기존의 Chase-Pyndiah 알고리즘만을 이용하는 것에 비해 제안 기법이 0.6dB 가량의 성능 개선이 가능함을 그림 2에서 보였다. 또한 제안된 알고리즘을 VHDL을 통하여 하드웨어 구현을 진행하고, 이 과정에서 고려해야 할 사항을 양자화 과정과 복호 과정을 중심으로 살펴 보았다.

#### References

[1] R. M. Pyndiah, "Near-optimum decoding of product codes: Block turbo codes," *IEEE Trans. Commun.*, vol. 46, no. 8, pp. 1003-1010, Aug. 1998.

[2] D. Chase, "A class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inf. Theory*, vol. 18, no. 1, pp. 170-182, Jan. 1972.

[3] S. C. Ha, B. K. Ahn, and J. Heo, "Iterative decoding scheme considering the implementation of three dimensional turbo product codes with parity check code," in *Proc. KICS*

*Int. Conf. Commun.* 2016, pp. 147-148, Jung sun, Korea, Jan. 2016.

[4] X. Wu, Y. He, and G. Zhu, "Performance of improved three-dimensional turbo code decoder," *IEEE Int. Conf. Integration Technol.*, pp. 564-567, Shenzhen, China, Mar. 2007.

[5] K. W. Kim, T. H. Yoon, and E. K. Joo, "Erasure decoding by parity check equations of hamming codes," in *Proc. KICS Int. Conf. Commun.* 2014, pp. 852-853, Pyeongchang, Korea, Jan. 2014.

[6] J. R. Barry, "Low-density parity-check codes," *course notes, Georgia Institute of Technology*, Oct. 2001.

[7] H. H. Back, J. W. Kang, K. S. Kim, and H. N. Lee, "Introduction and performance analysis of approximate message passing(AMP) for compressed sensing signal recovery," *J. KICS*, vol. 38, no. 11, pp. 1029-1043, Nov. 2013.

[8] D. G. Choi, I. K. Lee, and J. W. Jung, "High speed turbo product code decoding algorithm," *J. KICS*, vol. 30 no. 6C, pp. 442-449, Jun. 2005.

#### 하 상 철 (Sang-chul Ha)



2015년 2월 : 고려대학교 전기 전자전파공학부 졸업  
2015년 3월~현재 : 고려대학교 전기전자공학과 석사과정  
<관심분야> 통신공학, 채널코딩, FPGA 설계

**안 병 규 (Byung-kyu Ahn)**



2011년 8월 : 고려대학교 전기  
전자전파공학부 졸업  
2011년 9월~현재 : 고려대학교  
전기전자공학과 석박사통합  
과정  
<관심분야> 양자오류정정부호,  
채널코딩, FPGA 설계

**오 지 명 (Ji-myung Oh)**



1997년 2월 : 고려대학교 전파  
학과 졸업  
1999년 2월 : 고려대학교 전파  
학과 석사  
2004년 10월~2009년 6월 : 포  
스데이터 책임연구원  
2009년 7월~현재 : LIG넥스원  
수석연구원

<관심분야> 전파공학, 통신공학, 무선이동통신

**김 도 경 (Do-kyoung Kim)**



1996년 2월 : 항공대학교 통신  
정보 공학과 졸업  
2014년 2월 : 고려대학교 국방  
기술경영 석사  
2009년 6월~현재 : LIG넥스원  
수석연구원  
<관심분야> 통신공학, 무선이  
동통신

**허 준 (Jun Heo)**



1989년 2월 : 서울대학교 전자  
공학과 학사 졸업  
1991년 2월 : 서울대학교 전자  
공학과 석사 졸업  
2002년 2월 : University of  
Southern California 박사 졸업  
1991년~1996년 : LG전자 영상  
미디어 연구소, 주임연구원  
1996년~2002년 : LG전자 중앙연구소, 선임연구원  
2002년~2003년 : Hynix 반도체 System IC Comp.  
책임연구원  
2003년~2007년 : 건국대학교 전자공학부 조교수  
2007년~2012년 : 고려대학교 전기전자공학부 부교수  
2012년~현재 : 고려대학교 전기전자공학부 정교수  
<관심분야> Channel Coding, Network Coding,  
Quantum Information Theory