ISSN : 1226-7244 (Print)
ISSN : 2288-243X (Online)
논문번호 19-04-25

j.inst.Korean.electr.electron.eng.Vol.23,No.4,1288~1294,December 2019
http://dx.doi.org/10.7471/ikeee.2019.23.4.1288

긴 극 부호를 위한 저 면적 부분 병렬 극 부호 부호기 설계

Area-Efficient Semi-Parallel Encoding Structure for Long Polar Codes

신 예 린*, 최 소 연*, 유 호 영*★

Yerin Shin*, Soyeon Choi*, and Hoyoung Yoo*★

Abstract

The channel-achieving property made the polar code show to advantage as an error-correcting code. However, sufficient error-correction performance shows the asymptotic property that is achieved when the length of the code is long. Therefore, efficient architecture is needed to realize the implementation of very-large-scale integration for the case of long input data. Although the most basic fully parallel encoder is intuitive and easy to implement, it is not suitable for long polar codes because of the high hardware complexity. Complementing this, a partially parallel encoder was proposed which has an excellent result in terms of hardware area. Nevertheless, this method has not been completely generalized and has the disadvantage that different architectures appear depending on the hardware designer. In this paper, we propose a hardware design scheme that applies the proposed systematic approach which is optimized for bit-dimension permutations. By applying this solution, it is possible to design a generalized partially parallel encoder for long polar codes with the same intuitive architecture as a fully parallel encoder.

요 약

Polar code의 채널용량 달성 특성은 polar code를 각광 받는 오류 정정 부호로 만들었다. 하지만 충분한 오류 정정 성능은 부호의 길이가 길어졌을 때 달성되는 점근적 속성을 보인다. 따라서 입력 데이터가 길어지는 경우에 대한 초대규모 집적회로 구현을 실현하기 위하여 효율적인 구조가 필요하게 되었다. 기존의 polar code 부호기 구조 중 가장 기본적인 완전 병렬 구조는 직관적이고 구현이 쉽지만 긴 polar code에 높은 하드웨어 복잡성을 보이므로 부적합하다. 그리고 이를 보완하여 제안된 부분 병렬 구조는 하드웨어 면적 측면에서 큰 성과를 얻었으나 그 방식이 일반화되어 있지 않아 설계자에 따라 구조에 변동이 발생할 수 있다. 본 논문에서는 이를 개선하고자 비트 차원의 치환을 위해 제안된 회로 설계법을 polar code에 적용하는 하드웨어 설계법을 제안한다. 제안하는 방법을 polar code의 부호기에 적용함으로써 완전 병렬 부호기만큼 직관적인 구조를 가짐과 동시에 일반화된 polar code 부분 병렬 부호기를 설계할 수 있다.

Key words : Area efficient, bit-permutation, partially parallel encoder, polar codes, hardware structure

* Dept. of Electronics Engineering, Chungnam national University
★ Corresponding author
E-mail : hyyoo@cnu.ac.kr, Tel : +82-42-821-6585
※ Acknowledgment
This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2019M3F3A1A01074449) and EDA tools were supported by IDEC, Korea.
Manuscript received Dec, 19, 2019; revised Dec. 21, 2019; accepted Dec. 26, 2019.

Ⅰ. 서론

Polar code는 기존의 채널 코딩 기법과 비교했을 때 상대적으로 낮은 복잡도로 채널용량을 달성할 수 있는 새로운 종류의 오류 정정 코드이다[1]. 그러나 polar code가 채널용량을 달성하기 위해서는 부호의 길이가 충분히 길어야 한다. 하지만 부호의 길이가 길어지는 경우 polar code의 부호화와 복호화 과정에 필요한 하드웨어의 복잡도가 증가하고 지연 시간 또한 길어진다. 따라서 부호의 길이가 긴 polar code를 효율적으로 처리할 수 있는 하드웨어 구조의 필요성이 대두되었다.

Polar code의 복호화를 효율적으로 수행하기 위한 연구는 연속-제거(Successive cancellation; SC) 복호 [5]-[6]와 신뢰-전파(belief propagation; BP) 복호 [7]-[8]를 기반으로 활발하게 진행되었다. 하지만 부호화를 효율적으로 수행하기 위한 연구는 활발하게 진행되지 않았다. 가장 전통적인 polar code 부호기로 모든 메시지 비트를 완전 병렬적 방식으로 처리하는 완전 병렬 부호기 구조가 제안되었다[1]. 이 방법은 직관적이고 구현이 쉽다는 장점이 있지만, 모든 부호를 동시에 처리하여 코드 워드의 길이가 길어질수록 요구되는 하드웨어의 면적이 급격히 증가해 면적 측면에서 적합한 방법이라 할 수 없다. 그리고 이어서 완전 병렬 구조의 단점을 보완하기 위하여 폴딩(Folding) 기법을 적용한 부분 병렬 구조가 제안되었다[9]. 이 구조는 완전 병렬 구조의 부호기에 비해 하드웨어 면적을 절약할 수 있다는 장점이 있지만, 동일한 병렬화 계수(parallel factor)를 가지더라도 설계자의 레지스터 할당 방법에 따라 하드웨어 구조가 달라질 수 있다. 본 논문에서는 설계자에 따라 구조의 차이가 발생할 수 있는 점을 극복하기 위하여 비트 차원의 치환(bit-dimension permutation)을 위해 제안된 하드웨어 설계법[10]을 polar code 부호기 구조에 적용하여 확실한 규칙성으로 일반화된 polar code 부호기 설계 기법을 제안한다.

Ⅱ. Polar code 부호화 (Polar Encoding)

Polar code는 N개의 비트-채널 집합으로 결합된 채널이 부호의 길이가 무한대에 가까워질수록 각 채널이 완전히 무결성을 보이거나 혹은 완전히 신뢰성을 잃어 노이즈 채널이 되는 채널 양극화 현상을 이용한다[1]. 각 비트-채널의 신뢰성은 선험적으로 알 수 있으므로 가장 신뢰할 수 있는 비트-채널 K개를 사용하여 정보비트를 전송하고 나머지 비트-채널은 고정 비트를 전송하여 polar code를 구성한다.

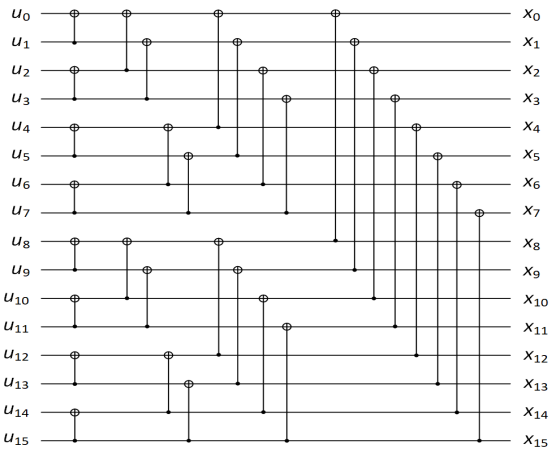

Fig. 1. Fully parallel architecture for encoding a 16-bit polar code.

그림 1. 16-bit polar code의 부호화를 위한 완전 병렬 구조

Polar code는 선형 블록 코드(linear block code)로 부호화 과정은 생성 행렬(generator matrix) $\mathbf{G}$에 의해 특징지어진다. 길이가 N 또는 2^n 인 부호에 대한 생성 행렬 $\mathbf{G}$는 커널 행렬 $\mathbf{F}=\begin{bmatrix}1&0\\1&1\end{bmatrix}$를 크로네커 거듭제곱하여 얻을 수 있다. 코드워드(codeword)는 $\boldsymbol{x}=\boldsymbol{u}\mathbf{G}_N$으로 구해지며, 이때 $\boldsymbol{u}$와 $\boldsymbol{x}$는 각각 입력 메시지 벡터와 코드워드를 나타낸다. Polar code의 부호화를 위한 완전 병렬 부호기 구조가 [1]에 제시되었고 길이 N의 polar code에 대해 $O(NlogN)$의 부호화 복잡도를 가지며 $N=2^n$ 일 때 n 단계로 나뉘어 부호화 과정이 진행된다.

그림 1에 N이 16일 때 4개의 단계에 걸쳐 부호화를 진행하며, 각 단계에서 $N/2$ 개의 XOR 게이트를 필요로 하는 완전 병렬 구조 부호기를 나타내었다. 완전 병렬 부호기는 생성 행렬 $\mathbf{G}$를 기반으로 직관적으로 설계되며 전체 부호화 과정이 한 사이클에 완료된다. 하지만 실제 구현에서 polar code의 부호화를 위해 완전 병렬 구조를 택하는 것은 부호 길이가 길어짐에 따라 XOR 게이트의 수가 증가하기 때문에 면적 측면에서 비효율적이다. 이러한 문제점을 개선하기 위하여 폴딩(Folding) 기법을 적

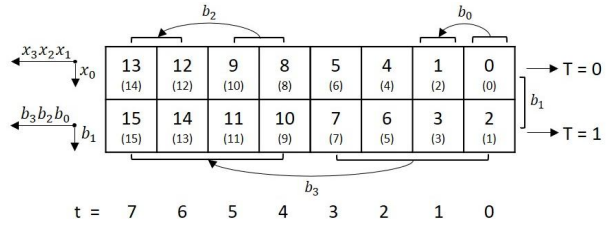

Fig. 2. Data flow representing $p=b_3b_2b_0|b_1$.
그림 2. $p=b_3b_2b_0|b_1$을 나타낸 데이터 흐름 모식도

용한 부분 병렬 구조가 제안되었다[9]. 폴딩 기법을 적용함으로써 하드웨어 면적을 줄일 수 있었으나, 구조를 유도하기 위한 과정이 복잡할 뿐 아니라 설계자의 의견이 설계 결과에 영향을 미쳐 하드웨어의 면적과 지연 시간 등 성능적인 측면에서 차이가 발생할 수 있다. 본 논문에서는 최적화된 비트 차원의 치환 하드웨어 설계 기법[10]을 polar code의 부분 병렬 부호기 설계에 적용하여 일관된 규칙성을 확보하고 완전 병렬 구조의 부호기 대비 복잡도 측면에서의 효율을 높이는 것을 목표로 하였다.

III. 비트 차원의 치환 회로 설계법을 적용한 부분 병렬 polar code 부호기

1. 비트 차원의 치환(Bit-dimension Permutations)

비트 차원의 치환이란 n차원의 공간 속에 있는 데이터 $x_{n-1}x_{n-2}\cdots x_0$를 치환 좌표에 따라서 재배열하는 것을 의미하며[11], 이를 수행하는 최적의 하드웨어 설계법이 제안되었다[10]. 제안된 방법의 기본 원리는 비트 치환을 기본적인 비트의 교환으로 분해하여 보는 것이다. 기본적인 비트의 교환(Elementary Bit Exchange, EBE)은 최종적으로 x_j와 x_k 두 차원 사이에서 일어난다.

비트 차원의 치환을 위하여 먼저 데이터의 흐름을 모델링 하는 과정이 필요하다. 데이터의 흐름은 관습적으로 왼쪽에서 오른쪽으로 진행되며 크게 직렬 차원 (serial dimension)과 병렬 차원 (parallel dimension)으로 구분하여 볼 수 있다.

직렬 차원에서는 서로 다른 시간에 같은 터미널을 통하여 데이터가 이동하고, 병렬 차원에서는 같은 시간에 서로 다른 터미널을 통해 데이터가 전달되게 된다. 결과적으로 데이터 흐름 모식도는 그림 2와 같이 2^{n-p}개의 직렬데이터, 2^p개의 병렬데이터로 구성된 직사각형 모양을 가진다. 이 직사각형의 공간 속에서 각 데이터의 위치 P가 정의된다. n 차원의 공간에서 각 데이터의 위치 값은 0에서 2^{n-1}까지의 번호로 나타나며, 식 (1)로 표현된다.

$$P=\sum_{i=0}^{n-1} x_i\,2^i \tag{1}$$

그리고 그림 2의 모식도에서 도착 시간 t와 데이터의 터미널 T를 어떠한 위치에서든 정의할 수 있다. 도착 시간 t는 회로 안의 주어진 지점에서 첫 번째 표본이 도착하는데 걸리는 시간을 의미하며 오로지 직렬 차원에만 의존적인 값이다.

$$t(P)=\sum_{i=0}^{n-1} x_i\,2^{i-p} \tag{2}$$

터미널 T는 식(3)을 통해 정의되며 도착 시간 t와는 반대로 오직 병렬 차원에만 의존적이다.

$$T(P)=\sum_{i=0}^{p-1} x_i\,2^i \tag{3}$$

도착 시간 t와 터미널 T를 사용하여 위치 P를 다시 식(4)로 나타낼 수 있다.

$$P=t\,|\,T \tag{4}$$

이때 바(bar) |는 직렬 차원과 병렬 차원을 구분하는 표시이다.

신호처리 알고리즘에서는 인덱스 된 데이터들로 수학 연산을 정의하기 때문에 데이터를 인덱스 화 하여야 하며, 인덱스 화는 식(5)을 통해 이루어진다. b_i는 인덱스의 10진수 값이고 2^i는 이진 표현의 비트이다.

$$I=\sum_{i=0}^{n-1} b_i\,2^i \tag{5}$$

P는 I 의 함수로 표현될 수 있으며, 인덱스화 된 데이터가 데이터 흐름의 각 위치에 할당되게 된다.

그림 2는 계산한 모든 값을 종합하여 하나의 병렬 차원과 세 개의 직렬 차원을 가지는 데이터 흐름을 나타낸 것이다. 연관된 도착 시간 t와 터미널 T를 표시하고 직사각형 공간 안의 괄호 안에는 위치 P값을, 괄호 밖에는 데이터의 인덱스 값을 표기하였다.

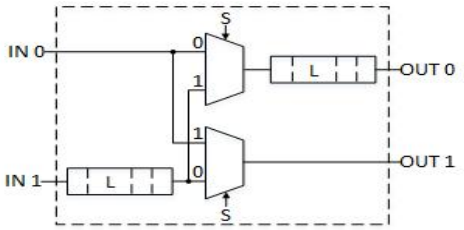

Fig. 3. Basic circuit for a serial-parallel EBE.
그림 3. 직렬-병렬 차원의 치환을 위한 기본적인 회로

2. 비트 차원의 치환을 위해 최적화된 회로 설계법

비트 차원의 치환을 위한 회로 설계는 비트 차원의 치환이 일련의 기본적인 비트의 교환으로 분해된다는 원칙을 기반으로 하며, 그에 따라 하드웨어 구조를 세 가지의 경우로 나눈다. 본 논문에서는 polar code에서 관찰될 수 있는 두 경우에 대해서만 간략히 살펴보도록 한다. 공통으로 데이터의 입력 순서를 p_0, 출력 순서를 p_1이라 하고 x_j와 x_k가 모두 병렬 차원에 존재하며 $p > j > k$라고 가정한다.

가. 두 병렬 차원 간의 기본적인 비트 교환

$$\begin{aligned} p_0 &\equiv u_{n-1} \cdots u_p \,|u_{p-1} \cdots u_j \cdots u_k \cdots u_0 \\ p_1 &\equiv u_{n-1} \cdots u_p \,|u_{p-1} \cdots u_k \cdots u_j \cdots u_0 \end{aligned} \tag{6}$$

두 병렬 차원 간의 비트 치환에는 직렬 차원이 포함되지 않으므로 같은 시간에 다른 터미널을 통해 데이터가 전달되며, 식 (6)은 치환의 결과를 식으로 나타낸 것이다. 결과적으로 단순히 입력 터미널과 출력 터미널을 상호 연결해 줌으로써 비트 치환을 완료할 수 있다.

나. 직렬, 병렬 차원 간의 기본적인 비트 교환

$$\begin{aligned} p_0 &\equiv u_{n-1} \cdots u_j \cdots u_p|u_{p-1} \cdots u_k \cdots u_0 \\ p_1 &\equiv u_{n-1} \cdots u_k \cdots u_p|u_{p-1} \cdots u_j \cdots u_0 \end{aligned} \tag{7}$$

직렬, 병렬 차원 간의 치환에는 치환되어야 하는 입력 데이터 쌍이 다른 터미널을 통해 전달되며, 치환 결과를 수식으로 표현하면 식 (7)과 같다. 다른 시간에 다른 터미널을 통해 치환되어야 하는 데이터가 입력되므로 데이터의 도착 소요 시간에 차이가 발생한다.

$$\triangle t = 2^{j-p} = L \tag{8}$$

이는 치환을 위해 요구되는 최소한의 지연 시간, L과 동일하며, 식(8)로 표현된다. 이 지연 시간의 영향을 반영하기 위해 L과 같은 길이의 버퍼 한 쌍과 치환을 위한 MUX 2개로 기본적인 셔플링 회로인 그림 3이 설계된다.

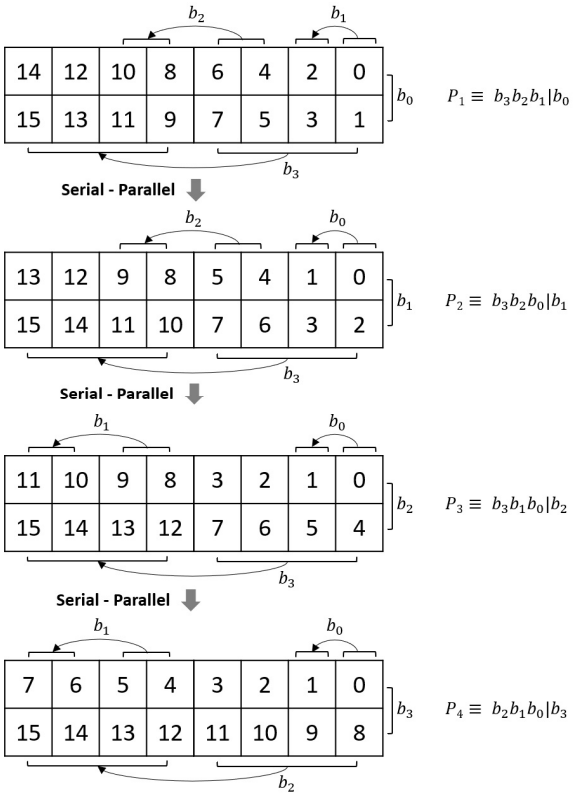

Fig. 4. Data flow of a polar encoder for each stage when *N=16*.
그림 4. *N=16* 일 때, 단계별 polar code 부호기의 데이터 흐름 모식도

두 MUX는 같은 컨트롤 시그널에 의해 통제되며 병렬 차원이 하나 이상일 경우, 그림 3의 회로가 2^{p-1}번 중복되어 결과 출력까지 식 (9)의 딜레이를 필요로 한다.

$$D(\sigma) = 2^{p-1} \cdot 2L = 2^j \tag{9}$$

이를 종합하여 전체 과정을 사이클로 구분하고 각 사이클마다 경우에 맞는 기초 회로를 적용해 비트 치환을 위해 최적화된 하드웨어 회로를 일반화하여 구할 수 있다.

3. Polar code를 위해 일반화된 부분 병렬 부호기

Polar code의 부호화를 위하여 거쳐야 하는 일련의 단계들 사이에는 치환이 요구된다. 부호화를 위해 치환이 필요한 부분에 비트 차원의 치환 회로 설계법을 적용하여 최종적으로 polar code를 위한

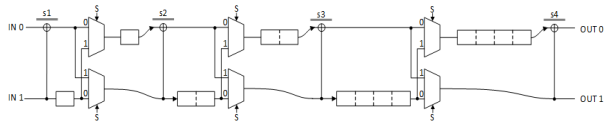

Fig. 5. Partially parallel encoder architecture of polar code using circuit design method based on bit-dimension permutations.

그림 5. 비트 차원의 치환을 기반으로 하는 회로 설계법을 사용한 polar code의 부분 병렬 부호기 구조

부분 병렬 부호기를 설계할 수 있다. 그림 4에 부호의 길이 N이 16이면서 병렬 차원이 1인 경우에 대한 데이터 흐름 모식도를 데이터 크기에 따라 4단계로 나타내었으며, 인덱스 화한 값과 단계가 전환될 때 이루어져야 하는 치환의 종류를 표기하였다. 그림 4의 데이터 흐름 모식도를 바탕으로 하여 polar code 부호기에 비트 차원의 치환 회로 설계법을 적용한 최종적인 회로를 그림 5에 제시하였다. 제안하는 polar code 부호기 설계법을 사용하면 부호의 길이가 늘어나더라도 직관적으로 데이터의 흐름을 모델링 할 수 있어 일반화되어 있는 회로를 적용하여 손쉽게 최적화된 polar code 부호기를 설계할 수 있다.

Ⅳ. 실험 결과

Polar code의 부호 길이 N을 1024로 고정하고 병렬 차원의 크기를 각각 4, 6, 8, 10으로 설정하여 비트 차원의 치환 회로 설계법을 기반으로 한 polar code 부분 병렬 부호기를 구현하였다. 합성은 CMOS 180nm 공정을 이용하여 동작 주파수를 200MHz로 가정하여 진행하였으며 그 결과를 표 1에 정리했다.

표 1에 나타난 합성 결과에 의하면 병렬 차원이 10인 경우는 완전 병렬 부호기와 동일한 구조를 보이며 병렬 차원이 4인 경우 대비 2.32배 더 큰 하드

Table 1. Synthesis results of a 1024-bit partially parallel polar encoder.

표 1. 1024-bit 부분 병렬 polar code 부호기 합성 결과

Parallel factor	16	64	256	1024
Gate Count	9,579	11,072	16,385	22,186
Latency (Clock cycles)	128	32	8	1
Throughput [Gbps]	1.6	6.4	25.6	204.8

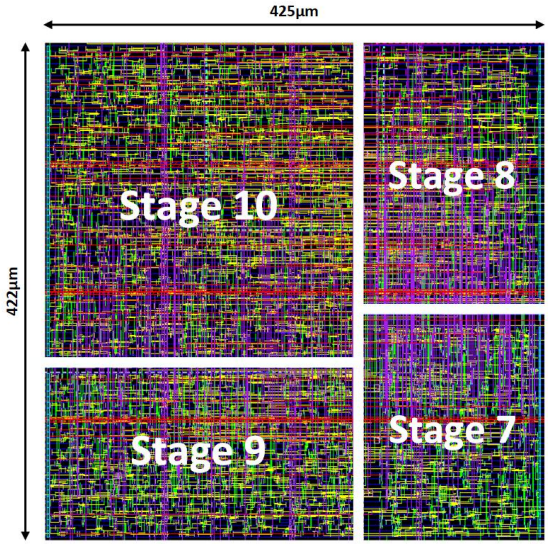

Fig. 6. When the size of the parallel dimension is 6, the layout of a 1024-bit partially parallel polar encoder.

그림 6. 병렬 차원의 크기가 6일 때, 1024-bit 부분 병렬 polar code 부호기 레이아웃

웨어 면적을 나타냄을 확인할 수 있다. 이를 통해 부분 병렬 polar code 부호기를 사용하는 것이 하드웨어의 면적 측면에서 더 효율적임을 알 수 있다. 하지만 최종 결과가 출력되기까지 소요되는 지연 시간은 병렬 차원이 늘어나 완전 병렬 구조에 가까워질수록 줄어든다. 즉, 부분 병렬 구조를 기반으로 polar code 부호기를 설계할 경우, 하드웨어 면적은 완전 병렬 구조 대비 줄어들어 이득을 볼 수 있지만 데이터의 처리량은 감소하게 된다.

그림 6에는 병렬 차원이 6일 경우에 대해 place& route를 진행하여 0.18㎟를 차지하는 레이아웃을 나타내었다. 병렬 차원이 6일 때에 대한 데이터 흐름 모델링 과정을 통해 총 10단계에 걸쳐 이루어지는 부호화 과정 중 6번째 단계까지는 두 병렬 차원 간의 치환이 이루어지며 그 이후 단계에서는 직렬-병렬 차원 간의 치환이 이루어짐을 알 수 있다. 두 병렬 차원 간의 치환 회로는 입력, 출력 터미널 간의 상호 연결만으로 구성되므로 전체 하드웨어 면적의 극히 일부를 구성하고 7-10단계를 위한 회로적 요소가 전체 레이아웃의 대부분을 이루게 된다. 이때, 단계가 증가함에 따라 필요한 버퍼의 수가 2배로 증가하므로 최종단계인 10단계에서 가장 큰 하드웨어 면적을 요구한다. 이를 나타내기 위해 각 단계별로 차지하는 면적을 구분하여 그림 6에 표기하였다.

Ⅴ. 결론

실험 결과를 통해 병렬 차원에 따른 부호기의 구조에 따라 데이터 처리량과 하드웨어 면적 간에 트레이드 오프가 존재함을 확인할 수 있다. 하지만 현재까지 polar code를 효율적으로 부호화하는 부호기 하드웨어 설계 방법에 관한 연구는 복호기에 비해 이루어지지 않았다. 본 논문에서 제안하는 비트 차원의 치환 회로 설계법을 적용한 polar code 부호기 설계 방식은 기존의 부분 병렬 polar code 부호기[9]에 적용된 방법과 비교해 보았을 때, 설계법이 더 직관적이며 확실한 규칙성을 보인다. 따라서 앞서 제안된 polar code 부호기의 단점을 보완하여 제안된 본 방법은 polar code 부호기를 위한 실용적인 솔루션을 제공한다.

References

[1] E. Arıkan, “Channel polarization : A method for constructing capacity achieving codes for symmetric binary-input memoryless channels,” *IEEE Trans. Inform. Theory*, vol.55, pp.3051-3073, 2009. DOI: 10.1109/TIT.2009.2021379

[2] SHRESTHA, Rahul; BANSAL, Pooja; SRINIVASAN, Srikant, “High-Throughput and High-Speed Polar-Decoder VLSI-Architecture for 5G New Radio,” *2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID),* IEEE, pp.329-334, 2019. DOI: 10.1109/VLSID.2019.00075

[3] WANG, Quanyv, et al. “Performance Analysis of Polar Codes for Wireless Sensor Networks,” *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC),* IEEE, pp.1-5, 2019.
DOI: 10.1109/ICEIEC.2019.8784608

[4] NGUYEN, Duc-Phuc, et al. “Performance Enhancement of Polar Codes in Multi-level Cell NAND Flash Memories using Systematic Encoding,” *2019 19th International Symposium on Communications and Information Technologies (ISCIT),* IEEE, pp.621-626, 2019.
DOI: 10.1109/ISCIT.2019.8905168

[5] B. Yuan and K. K. Parhi, “Low-Latency Successive-Cancellation Polar Decoder Architectures Using 2-Bit Decoding,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol.61, no.4, pp.1241-1254, 2014.
DOI: 10.1109/TCSI.2013.2283779

[6] WANG, Xiumin, et al, “Improved Adaptive Successive Cancellation List Decoding of Polar Codes,” *Entropy*, vol.21, NO.9, pp.899, 2019.
DOI: 10.3390/e21090899

[7] E. Arıkan, “Polar codes: A pipelined implementation,” *Proc. 4th Int. Symp. on Broad. Commun. ISBC 2010,* pp.11-14, 2010.

[8] GUO, Jing, et al. “Enhanced belief propagation decoding of polar codes through concatenation,” *2014 IEEE International Symposium on Information Theory,* IEEE, pp.2987-2991, 2014.
DOI: 10.1109/ISIT.2014.6875382

[9] H. Yoo and I. Park, “Partially Parallel Encoder Architecture for Long Polar Codes,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol.62, no.3, pp.306-310, 2015.
DOI: 10.1109/TCSII.2014.2369131

[10] M. Garrido, J. Grajal and O. Gustafsson, “Optimum Circuits for Bit-Dimension Permutations,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol.27, no.5, pp.1148-1160, 2019.
DOI: 10.1109/TVLSI.2019.2892322

[11] FRASER, Donald. “Array permutation by index-digit permutation,” *Journal of the ACM (JACM)*, vol. 23, NO. 2, p. 298-309, 1976.
DOI: 10.1145/321941.321949

BIOGRAPHY

Yerin Shin (Student Member)

2016~ : BS degree in Electronics Engineering, Chungnam National University.

Soyeon Choi (Student Member)

2018 : BS degree in Electronics Engineering, Chungnam National University.
2018~ : MS degree in Electronics Engineering, Chungnam National University.

Hoyong Yoo (Member)

2010 : BS degree in Electrical & Electronic Engineering, Yonsei University
2012 : MS degree in Electronics Engineering, KAIST
2016 : Ph.D. degree in Electronics Engineering, KAIST
2016 : Researcher, Samsung Electronics.
2016~ : Assistant Professor, Chungnam National University