

단일 비트 이하 오류 정정을 위한

극 부호용 선 처리 복호기법

Low-Latency Polar Decoding for Error-Free and Single-Error Cases

최 소 연*, 유 호 영*

Soyeon Choi*, Hoyoung Yoo*

Abstract

For the initial state of NAND flash memories, error-free and single-error cases are dominant due to a good channel environment on memory cells. It is important to deal with such cases, which affects the overall system performance. However, the conventional schemes for polar codes equally decode the codes even for the error-free and single-error cases since they cannot classify and decode separately. In this paper, a new pre-processing scheme for polar codes is proposed so as to improve the overall decoding latency by decoding the frequent error-free and single-error cases. Before the ordinary decoding process, the proposed scheme first decodes the frequent error-free and single-error cases. According to the experimental results, the proposed pre-processing scheme decreases the average decoding latency by 64% compared to the conventional scheme for (1024, 512) polar codes.

요 약

NAND 플래시 메모리를 포함하는 저장 매체의 경우 초기 상태에서 메모리 셀의 상태가 매우 우수하기 때문에, 오류가 없거나 단일 오류가 발생하는 경우가 지배적이다. 무-오류와 단일-오류에 대한 처리는 저장 매체의 시스템에 있어서 전체 시스템 성능에 영향을 미치는 중요한 요소가 된다. 전통적인 polar code 복호기법은 무-오류 또는 단일-오류에 대한 독립적인 검출이 불가능하기 때문에 일반적인 복호화 기법을 따르며 다른 오류 발생 경우와 동일한 수준의 지연 시간을 가진다. 본 논문에서는 저장 매체에서 빈번하게 발생하는 무-오류 또는 단일-오류에 대한 검출 및 정정을 일반적인 복호화 과정에 앞서 처리해줌으로써, 전체 복호화 과정에 필요한 평균적인 지연시간을 감소시키는 기법을 제안한다. 실험 결과에 따르면 무-오류 또는 단일-오류에 대한 제안하는 선 처리 (pre-processing) 기법을 적용할 경우 (1024, 512) 극 부호에 대하여 일반적인 복호화 기법 대비 평균 지연시간을 약 64% 줄일 수 있다.

Key words : Polar codes, pre-processing, bit-flipping, single-error correction, low-latency decoding

* 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 (NRF) grant funded by the Korea government (MSIP) (2017R1C1B5015962) and by the IC Design Education Center (IDEC).

Manuscript received Dec. 10, 2018; revised Dec. 17, 2018; accepted Dec. 17, 2018

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

I. 서론

채널 용량을 달성하는 우수한 오류정정능력 덕분에 polar code는 최근 가장 주목 받는 부호 중 하나이다 [1]. 차세대 통신 시스템과 저장 시스템에서 polar code를 오류정정부호로 도입하기 위하여 정보이론 측면과 하드웨어 구현 측면에서 활발하게 분석을 진행 중이다 [2]-[4]. Polar code는 채널 극화(channel polarization) 현상을 기반으로 하며, 전체 비트-채널을 완전 무결한 채널과 완전 노이즈 채널로 나눈다. 데이터 전송을 위하여 완전 무결한 채널을 통해 유의미한 정보를 전송하고 완전 노이즈 채널을 이용해 고정(frozen) 비트를 전송한다. Polar code의 복호화 과정으로는 연속 제거(successive cancellation : SC) 복호 기법 [1]과 연속제거 리스트(SC-List) 복호 기법 [5]-[6]이 널리 적용된다. 연속 제거 복호 [1]는 계산 복잡도가 작은 장점을 가지는 반면 오류정정 성능이 최신의 LDPC 부호나 터보부호에 비하여 우수하지 못하다. 연속제거 리스트 복호 기법 [5]-[6]에서는 여러 부호를 동시에 비교하여 연속 제거 복호 [1]의 오류정정 성능을 향상시킨다.

최근 5세대 무선통신시스템에 제어 채널용 오류정정부호로 polar code가 표준으로 채택되었으며 [7], 저장 시스템에서도 오류정정부호의 도입을 활발히 연구 중에 있다 [8]-[9]. 특히, 저장 시스템의 경우 전통적으로 BCH 부호나 LDPC 부호가 주로 적용 되었으나, 메모리 셀 당 저장하는 비트 수가 증가함에 따라 채널 환경이 악화되었고 이에 따라 더욱 강력한 오류정정능력을 보유한 polar code의 적용을 활발히 고려하고 있다. 저장 매체를 위한 메모리 셀의 경우 오류 분포를 분석해보면 초기 상태에는 무-오류 또는 단일-오류가 지배적이며, 프로그램과 소거를 반복할수록 다중 오류가 발생하는 경향을 보인다 [10]. 그림 1은 채널 환경이 매우 우수한 초기 상태를 가정해 두고 SNR이 10 dB일 때 발생하는 오류 패턴의 비율을 나타내며, 이 때 N 은 전체 부호 길이이다. 부호 길이가 1024인 경우 무-오류 또는 단일-오류의 경우가 전체 오류 패턴 중에 약 81%를 차지한다.

본 논문에서는 빈번하게 발생하는 무-오류 또는 단일-오류에 대한 검출 및 정정을 일반적인 복호화 과정에 앞서 처리해줌으로써, 전체 복호화 과정

에 필요한 평균적인 지연시간을 감소시키는 기법을 제안한다. 특히, 제안하는 알고리즘은 경판정(hard-decision)된 결과를 이용하여 복호를 진행하기 때문에 연판정(soft-decision)을 진행하는 일반적인 복호과정 대비 복잡도가 적고, 빈번하게 발생하는 무-오류 또는 단일-오류에 대한 선 처리가 가능하기 때문에 평균지연시간을 획기적으로 줄일 수 있다.

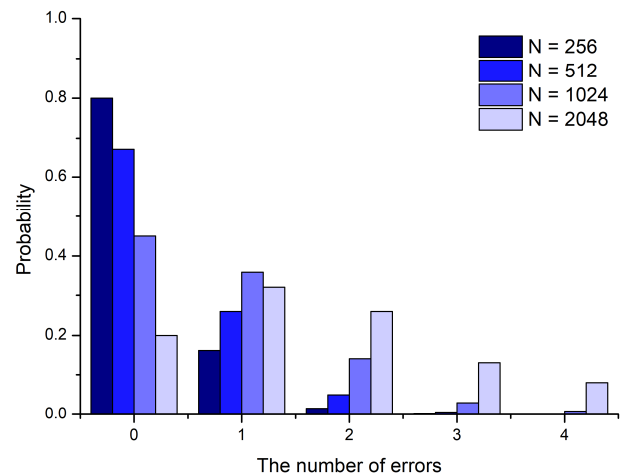


Fig. 1. The probability of the number of errors.

그림 1. 오류 패턴의 비율

II. 본론

1. Polar code

(N, K) polar code [1]를 가정했을 때, polar code는 K 개의 정보비트와 $N-K$ 개의 고정비트로 구성되며, N 은 전체 부호 길이를 나타내고 K 는 유의미한 정보비트 길이를 나타낸다. 전체 N 개의 비트채널 중 가장 신뢰도가 높은 K 개의 비트채널에 정보비트가 할당되고, 신뢰도가 낮은 $N-K$ 개의 비트채널에 고정 비트가 할당되어 부호를 전송한다. 이 때 전체 N 개의 인덱스 중에서 정보비트의 인덱스 집합을 A 라고 하며, 고정비트의 인덱스의 집합은 A^c 로 표현한다. Polar code는 선형 블록 부호에 속하기 때문에 부호의 모든 특징은 생성 행렬(Generator matrix)로 표현 가능하다. Polar code의 코드워드 $\mathbf{x}$ 는 입력 메시지 벡터 $\mathbf{u}$ 와 생성 행렬의 곱, $\mathbf{x}=\mathbf{u} \cdot \mathbf{G}_N$ 으로 생성된다. 생성 행렬 $\mathbf{G}_N$ 은 행렬 $\mathbf{F}$ 의 $n(n=\log_2 N)$ 크로네커 곱(Kronecker product)으로 만들어지며, $\mathbf{F}=\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$, $\mathbf{G}_N=\mathbf{F}^{\otimes n}$ 이다.

Polar code의 복호화 과정으로는 연속 제거 복호 기법 [1]과 연속제거 리스트 복호 기법 [5]-[6]이 널리 적용된다. 연속 제거 복호 기법 [1]과 연속제거 리스트 복호 기법 [5]-[6]에서 모두 하나의 비트 채널을 복호하기 위하여 선행된 비트 채널들의 복호가 우선 진행되어야 하기 때문에, 한 비트 씩 연속적으로 복호화를 진행한다. 이러한 연속적인 복호화 과정은 부득이하게 긴 지연 시간을 요구하며 결과적으로 전체 시스템 성능을 떨어뜨린다. 예를 들어 (N, K) polar code를 이용하여 하드웨어를 구현하였을 경우, tree 구조 [2]로 하드웨어를 구현하면 $2(N-1)$ 지연 시간을 가지며, 병렬화 계수 P 를 이용하여 semi-parallel 구조 [3]로 하드웨어를 구현하면 $2N + (N/P)\log_2(N/4P)$ 지연 시간을 가진다. 부호 길이가 짧은 경우 복호화를 위한 지연 시간이 길지 않지만, 저장 매체를 위한 polar code의 경우 부호 길이가 1024이상의 긴 부호 길이를 가지기 때문에 긴 지연 시간이 전체 시스템을 결정한다. 특히, 메모리 셀의 오류가 거의 발생하지 않는 초기 저장 매체의 경우 저 지연을 달성할 수 있는 기법이 반드시 필요하다.

III. 비트 반전 복호 기법

그림 1에 따르면 메모리 셀이 매우 양호한 초기 상태에서는 무-오류 또는 단일-오류의 경우가 전체 오류 패턴에 대하여 가장 빈번하게 일어난다. 일반적인 연속 제거 복호 기법 [1]과 연속제거 리스트 복호 기법 [5]-[6]에서 모두 무-오류 또는 단일-오류에 대한 복호가 가능하기 때문에, 해당 오류를 위한 복호화를 위해 활용할 수 있지만 연관성에 따른 계산 복잡도와 지연 시간 증가는 피할 수 없다. 따라서 무-오류 또는 단일-오류를 위해 특화된 검출 및 정정 방안이 필요하다. 본 논문에서는 LDPC 복호화를 위해 Gallager가 제안한 비트-반전 복호기법 [11]을 polar code에 적용하여 낮은 계산 복잡도와 짧은 지연 시간을 지니며 무-오류와 단일-오류에 대해 복호가 가능한 기법을 제안한다.

1. Polar code의 H-행렬

비트-반전 복호 기법 [11]은 신드롬 판단을 통해 오류 발생 유무를 판단하며, 비트-반전 복호 기법 [11]을 polar code에 적용하기 위하여 polar code의

$\mathbf{H}$ -행렬 [12]이 우선 정의되어야 한다. 패리티 검사 행렬인 $\mathbf{H}$ -행렬 [12]은 주어진 부호어가 유효한 코드워드인지 확인할 수 있는 행렬이다. 만약 주어진 코드워드 벡터 $\mathbf{x}$ 가 유효한 코드워드 라면 $\mathbf{x}\mathbf{H}^T=0$ 이어야 한다. 그렇지 않다면 전송과정에서 오류가 생긴 것을 의미한다. (N, K) polar code에서 $\mathbf{H}$ -행렬 [12]의 크기는 $(N-K) \times N$ 이다. $\mathbf{H}$ -행렬 [12]의 각 행은 $\mathbf{G}_N$ 행렬의 i 번째 열과 같으며, $i \in A^c$ 이다. 원소의 수가 N 개인 코드워드 벡터 $\mathbf{x}$ 는 $\mathbf{x}\mathbf{H}^T=0$ 을 만족하거나 그 역을 만족하면 유효한 코드워드라고 말할 수 있다.

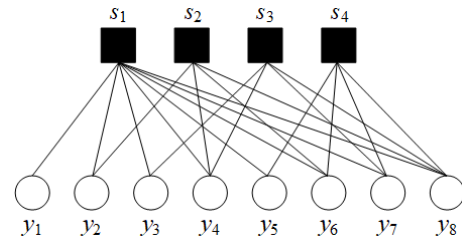


Fig. 2. Factor graph of bit-flipping decoding.

그림 2. 비트 반전 복호 기법의 factor graph

코드 길이가 8, 코드 비율이 1/2라고 했을 때, 생성행렬 $\mathbf{G}_N$ 은

$$\mathbf{G}_N = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (1)$$

이고, 생성행렬을 이용하여 $\mathbf{H}$ -행렬 [12]을 구하면 식 (2)의 행렬이 된다.

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (2)$$

2. 비트-반전 복호

신드롬 판단을 위해서는 확인노드(check node)와 변수노드(variable node)를 구별해야 한다. $\mathbf{H}$ -행렬 [12]의 c 번째 행의 v 번째 열에 있는 원소를 $\mathbf{H}(c, v)$ 라고 했을 때, c 번째 확인노드에 연결된 변수노드의 집합은

$$V(c) = \{v : \mathbf{H}(c, v) = 1\} \quad (3)$$

이고, v 번째 변수노드에 연결된 확인노드의 집합은

$$C(v) = \{c : \mathbf{H}(c, v) = 1\} \quad (4)$$

이다. (8, 4) polar code의 $\mathbf{H}$ -행렬 [12]에서 확인 할 수 있는 확인노드와 변수노드의 연결을 표현한 factor graph는 그림 2와 같이 나타낼 수 있다.

Proposed algorithm

```

Syndrome check:
  for  $c = 0$  step 1 until  $(N-K)$  begin
    calculate  $s_c$ 
  end for

if  $(\sum s_c = 0)$  then
  Error free case; break
else
  for  $v = 0$  step 1 until  $N$  begin
    calculate  $E_v$ 
     $k \leftarrow \text{argmax } E_v$ 
  end for
  flip  $h(y_k)$ 
  Syndrome check
  if  $(\sum s_c = 0)$  then
    Single error case; break
  else
    Multiple error case;
    Conventional decoding
  end if

```

Fig. 3. Proposed bit-flipping algorithm.

그림 3. 제안하는 비트-반전 알고리즘

그림 2의 검정색 네모는 확인노드를 나타내고, 하얀색 동그라미는 변수노드를 나타낸다.

비트-반전 복호 기법 [11]은 신드롬 판단을 근거로 반전시킬 비트 위치를 결정한다. 각 확인노드에서는 신드롬 계산을 위하여 식 (5)의 연산을 진행한다.

$$s_c = \sum_{i=1}^N h(y_i) \mathbf{H}(c, i) \quad (5)$$

이 때 경관정 함수 $h(y_i)$ 는 $y_i \geq 0$ 이면 0, $y_i < 0$ 은 1로 결정되며, y_i 는 코드워드 x_i 가 채널을 통과해 수신된 신호로 $y_i = x_i + n(t)$ 이고 $n(t)$ 는 AWGN을 의미한다. 식 (5)에 따르면 확인노드의 신드롬 판단 결과인 s_c 는 이진 필드에서 경관정 결과와 $\mathbf{H}$ -행렬 [12] 원소의 곱으로 계산하기 때문에 0 또는 1의 값을 가진다. 최종적으로 모든 확인 노드의 신드롬

판단이 0인 경우 해당 코드워드에 오류가 없음으로 판단을 내리고 복호화 과정을 종료하고, 그렇지 않다면 다음 과정인 판정 값 계산을 진행한다.

판정 값 계산은 가장 신뢰도가 낮은 비트 인덱스를 결정하기 위하여 필요하며, 모든 변수노드에서 식 (6)과 같이 계산된다.

$$E_v = \sum_{c \in C(v)} (2s_c - 1) \quad (6)$$

변수노드의 판정 값을 모두 계산한 다음, 판정 값이 가장 큰 비트 인덱스를 찾는다. 판정 값이 가장 큰 인덱스를 k 라고 할 때, k 번째 경관정 결과 $h(y_k)$ 를 반전한다. 그림 3은 제안하는 단일-비트 이하 오류 정정을 위한 비트-반전 복호 기법을 나타낸다.

제안하는 비트-반전 복호를 이용하면 단일-오류가 발생하였을 경우에는 완벽하게 오류정정이 가능하며 이를 수학적으로 증명할 수 있다. 단일-오류가 발생하였다고 가정하였을 때, 계산의 편의성을 위하여 입력 벡터 $\mathbf{u}$ 는 영 벡터라고 가정하고, 오류가 발생한 인덱스는 e 라고 가정한다. 오류가 발생한 인덱스가 e 이기 때문에 수신된 신호 벡터 $\mathbf{y}$ 의 원소 중 y_e 의 경관정 결과만 1이다. 따라서 신드롬 판단의 식 (5)은

$$s_c = \sum_{i=1}^N h(y_i) \mathbf{H}(c, i) = h(y_e) \mathbf{H}(c, e) \quad (7)$$

로 다시 표현할 수 있으며, 오류 인덱스의 s_c 만 1이다. 주어진 신드롬 결과를 이용한 판정 값을 식 (6)에 따라서 계산하면 다음 식과 같이 나타낼 수 있다.

$$E_v = \sum_{c \in C(v)} (2s_c - 1) = \begin{cases} \sum_{c \in C(v)} \mathbf{H}(c, e) & , v = e \\ \sum_{c \in C(v)} \mathbf{H}(c, v) \cdot (2s_c - 1) & , otherwise \end{cases} \quad (8)$$

식 (8)에서 오류 인덱스와 변수노드의 인덱스가 다를 때, 두 가지 경우를 고려해야한다. 먼저 $\mathbf{H}(c, v) = 0$ 이고, $\mathbf{H}(c, e) = 1$ 인 경우에는 s_c 의 값이 항상 1이지만 $\mathbf{H}(c, v)$ 가 0이기 때문에 $\mathbf{H}(c, v)(2s_c - 1)$ 은 0이다. $\mathbf{H}(c, v) = 1$ 이고, $\mathbf{H}(c, e) = 0$ 인 경우에는 s_c 의 값이 -1이기 때문에 $\mathbf{H}(c, v)(2s_c - 1)$ 은 -1이다. $\mathbf{H}$ -행렬 [12] 의 서로 다른 두 열 벡터는 서로 독립적이기 때문에 v 와 e 가 다른 경우에는 $\mathbf{H}(c, v) = 1$ 이고 $\mathbf{H}(c, e) =$

0인 경우가 1개 이상 존재하게 되어 이때의 판정 값은 항상 $v=e$ 인 변수노드의 판정 값보다 작다. 따라서 판정 값이 최대인 인덱스는 항상 오류가 발생한 인덱스 e 가 되며, $h(y_e)$ 의 결과를 반전시켜주면 오류는 완벽하게 정정된다.

3. 제안하는 복호 기법의 하드웨어 구조

제안하는 단일 비트 이하 오류정정을 위한 하드웨어 구조는 그림 4와 같다. 변수노드의 신드롬 계산을 진행하고, 이를 바탕으로 판정 값을 계산한 후, 가장 신뢰도가 낮은 비트 인덱스를 결정하여 이를 반전시킨다. 그림 4와 같이 신드롬 계산을 위하여 $\lceil (N/2)\log_2 N \rceil$ 의 XOR가 필요하고, 판정값 계산을 위해 $(N-K)$ 개의 매핑 회로와 $\lceil (N/2)\log_2 N \rceil$ 개의 덧셈기 $(N-1)$ 개의 비교기가 필요하다. 이 때 매핑회로는 신드롬 계산값 0과 1을 1과 -1 로 변환해 준다. 예를 들어 $(8, 4)$ polar code는 그림 4의 하드웨어 구조와 같이 XOR 10개, 매핑회로 4개, 덧셈기 7개, 비교기 7개가 필요하다. 제안하는 알고리즘은 경관정된 결과를 이용하여 복호를 진행하기 때문에 연판정을 진행하는 일반적인 복호과정 대비 복잡도가 적고 지연 시간 또한 매우 짧아, 일반적인 복호화 과정에 앞서 처리하기에 매우 용이하다.

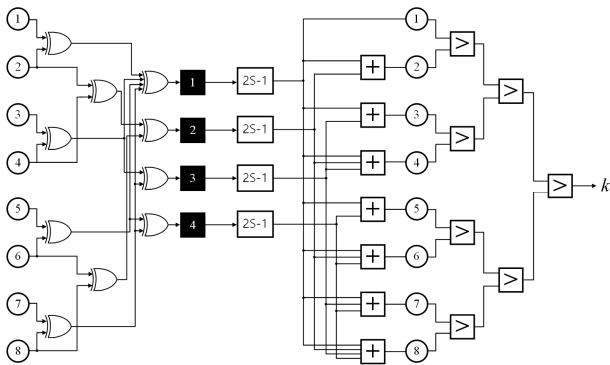


Fig. 4. Hardware architecture of BF algorithm.
그림 4. BF 알고리즘의 하드웨어 구조

IV. 분석 및 실험결과

제안하는 복호화 기법의 성능을 분석하기 위하여 연속 제거 복호 기법 [1]과 연속 제거 리스트 복호 기법 [5]-[6]에 가장 널리 적용되는 tree 구조 [2]와 semi-parallel 구조 [3]에 적용하였고, 일반적인 복

호화 기법과 제안하는 기법에 대한 평균적인 지연 시간을 비교하였다. Tree 구조 [3]의 경우 한 클럭 사이클 당 한 스테이지를 처리할 수 있으므로 $2(N-1)$ 지연시간을 가지며, 면적 최적화에 대한 반대급부로 semi-parallel 구조 [3]의 경우 $2N + (N/P)\log_2(N/4P)$ 지연 시간을 가지고 이 때 P 는 병렬화 계수이다. 일반적인 복호 기법을 적용하였을 경우 tree 구조 [2]와 semi-parallel 구조 [3] 모두 오류 패턴에 따른 지연시간의 변화는 없다. 각 하드웨어 구조에 제안하는 복호 기법을 적용하여 비트-반전 복호기를 우선적으로 처리할 경우 평균 지연 시간은 식 (9)와 같이 나눌 수 있다.

$$T_{avg} = (p_0 + p_1)T_{proposed} + (1 - p_0 - p_1)(T_{proposed} + T_{conv}) \quad (9)$$

p_0 와 p_1 은 무-오류와 단일-오류에 대한 확률을 나타내며, $T_{proposed}$ 는 제안하는 알고리즘의 지연 시간, T_{conv} 는 일반적인 복호 기법의 지연시간을 나타내며 tree구조 [2]를 적용할 경우 $2(N-1)$ 이고 semi-parallel 구조 [3]를 적용할 경우 $2N + (N/P)\log_2(N/4P)$ 이다.

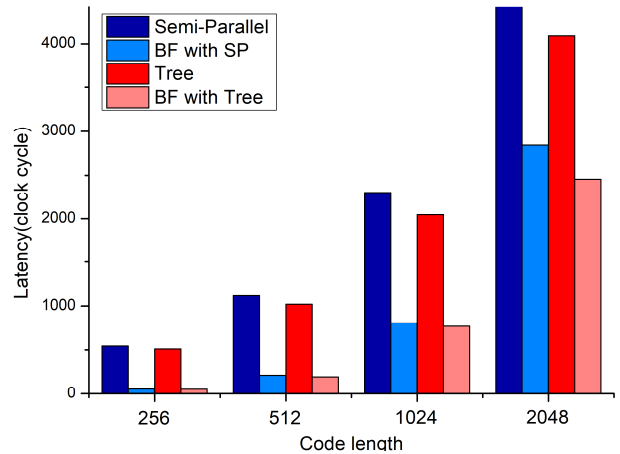


Fig. 5. Latency comparison according to code length.
그림 5. 전체 메시지 길이에 따른 지연시간 비교

그림 5은 다양한 부호 길이 N 와 병렬화 계수 P 는 16인 경우에 대하여 일반적인 복호 기법과 제안하는 복호기법을 적용한 tree 구조 [2]와 semi-parallel 구조 [3]의 평균 지연 시간을 비교한다. 저장 시스템의 초기 상태를 가정하고 채널 SNR을 10 dB로 지정하였다. 부호의 길이가 1024일 때 제안하는 복호 기법을 적용하면 일반적인 복호화 기법 대비 tree

구조 [2]와 semi-parallel 구조 [3]에서 64%의 지연 시간을 줄일 수 있다.

V. 결론

본 논문에서는 polar code를 이용하였을 때 무-오류 또는 단일-오류 비트를 검출 및 정정할 수 있는 효율적인 선 처리 기법을 제안하였다. 제안하는 비트-반전 복호 기법을 적용하면, 일반적인 복호 기법과는 달리 무-오류 또는 단일-비트에 대하여 저 복잡도 저 지연시간을 가지는 복호를 진행할 수 있으며, 다중-비트 오류에 대해서는 일반적인 복호 기법과 동일한 수준의 지연시간을 갖는다. 반복적으로 진행되는 선 처리 복호화 과정의 계산 량과 지연시간이 매우 짧기 때문에 다중-비트 오류의 경우에도 오버헤드가 크지 않다. 따라서 채널 환경이 매우 우수한 통신 시스템이나 초기 상태의 저장 시스템의 경우 제안하는 복호 기법을 활용할 경우 지연 시간을 획기적으로 감소시킬 수 있다. 또한, 제안하는 복호 기법은 선 처리 과정으로써 다양한 하드웨어 구조에 적용이 가능하기 때문에 활용성이 매우 높다.

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] C. Leroux, I. Tal, A. Vardy, and W. J. Gross, "Hardware architectures for successive cancellation decoding of polar codes," in *Acoustics, Speech and Signal Processing (ICASSP)*, 2011 IEEE International Conference on. IEEE, pp.1665-1668, 2011. DOI:10.1109/ICASSP.2011.5946819

[3] C. Leroux, A. J. Raymond, G. Sarkis, and W. J. Gross, "A semi-parallel successive cancellation decoder for polar codes," *IEEE Trans. Signal Process.*, vol.61, no.2, pp.289-299, 2013. DOI:10.1109/TSP.2012.2223693

[4] B. Yuan and K. K. Parhi, "Low-latency successive-cancellation polar decoder architectures using 2-bit decoding," *Circuits and Systems I: Regular Papers*, IEEE Transactions on, vol.61, no.4, pp.1241-1254, 2014. DOI:10.1109/TCSI.2013.2283779

[5] I. Tal and A. Vardy, "List decoding of polar codes," in *Proc. IEEE Symp. Inform. Theory*, Saint Petersburg, Russia, pp.1-5, 2011. DOI:10.1109/TIT.2015.2410251

[6] B. Li, H. Shen, and D. Tse, "An Adaptive Successive Cancellation List Decoder for Polar Codes with Cyclic Redundancy Check," in *IEEE Communications Letters*, vol.16, no.12, pp.2044-2047, 2012. DOI:10.1109/LCOMM.2012.111612.121898

[7] 3GPP, "5G: Study on new radio (NR) access technology," 3GPP TS 38.212 v.15.0.0, 2017.

[8] Guangyi Liu and Dajie Jiang, "5G: Vision and Requirements for Mobile Communication System towards Year 2020," *Chinese Journal of Engineering*, vol.2016, Article ID 5974586, 2016. DOI:155/2016/5974586

[9] G. Sarkis and W. J. Gross, "Polar codes for data storage applications," *2013 International Conference on Computing, Networking and Communications (ICNC)*, San Diego, CA, pp.840-844, 2013. DOI:10.1109/ICCNC.2013.6504198

[10] O. Afisiadis, A. Balatsoukas-Stimming, and A. Burg, "A low-complexity improved successive cancellation decoder for polar codes," in *Proc. 48th Asilomar Conf. Signals Syst. Comput.*, pp.2116-2120, 2014.

[11] R. Gallager, "Low-density parity-check codes," in *IRE Transactions on Information Theory*, vol.8, no.1, pp.21-28, 1962. DOI:10.1109/TIT.1962.1057683

[12] H. Yoo and I. -C. Park, "Efficient pruning for successive-cancellation decoding of polar codes," *IEEE Commun. Lett.*, vol.20, no.12, pp.2362-2365, 2016. DOI:10.1109/LCOMM.2016.2607167

BIOGRAPHY

Soyeon Choi (Student Member)

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

Hoyoung 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~ : Assistant Professor, Chungnam National University