

연속 제거 복호기반의 최신 극 부호 복호기법 비교

Comparison on Recent Decoding Methods for Polar Codes based on Successive-Cancellation Decoding

최 소 연*, 유 호 영*

Soyeon Choi*, Hoyoung Yoo*

Abstract

Successive cancellation (SC) decoding that is one of the decoding algorithms for polar codes has long decoding latency and low throughput because of the nature of successive decoding. To reduce the latency and increase the throughput, various decoding structures for polar codes are presented. In this paper, we compare the previous decoding structures and analyze them by dividing into two types, pruning and multi-path decoders. Decoders for applying pruning are representative of SSC (simplified SC), Fast-SSC and redundant-LLR structures, and decoders with multi-path are representative of 2-bit SC and redundant-LLR structures. All the previous structures are compared in terms decoding latency and hardware area, and according to the comparison, the syndrome check based decoder has the lowest latency and redundant-LLR decoder has the highest hardware efficiency.

요 약

Polar code의 복호 기법 중에 하나인 연속 제거 (successive cancellation; SC) 복호는 순차적으로 복호를 수행해야하는 특성으로 인해 지연시간이 길고, 복호를 위해 필요한 하드웨어 면적이 크다. 이를 극복하기 위하여 다수의 연구들이 진행되었으며, 본 논문에서는 연속 제거 복호를 기반으로 한 복호 기법을 가지치기 (pruning) 복호 기법들과 다중-경로 (multi-path) 복호기법들로 나누어 정리하였다. 가지치기 복호기법에는 SSC (simplified SC), fast-SSC, 신드롬 판단 기반 복호 등이 있으며, 다중-경로 복호 기법에는 2-비트 연속 제거 복호와 redundant-LLR 표현 기반의 복호가 있다. 본 논문에서는 SSC, fast-SSC, 신드롬 판단, 2-비트 연속 제거, 그리고 redundant-LLR 표현 기반의 복호 기법들을 지연시간과 하드웨어 면적 측면에서 비교했으며, 비교 결과 신드롬 판단 기반 복호기법이 지연시간이 가장 짧고, redundant-LLR 표현 기반의 복호가 하드웨어 면적이 가장 작은 복호 기법이다.

Key words : Polar codes, Successive-cancellation, polar decoder, pruning, multi-path

* 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 Brain Korea 21 Plus and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2019M3F3A1A01074449)

Manuscript received Jun. 1, 2016; revised Jun. 23, 2020; accepted Jun. 24, 2020.

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.

1. 서론

Polar code는 2008년 Arikan이 소개한 채널 코딩 기법으로, 메모리가 없는 채널(memoryless channel)에서 새년의 채널 용량을 달성할 수 있는 최초의 오류 정정 부호이다[1]. 새년의 채널 용량을 달성할 수 있다는 특성 때문에 polar code는 부호 이론에서 가장 중요한 오류 정정 부호 중 하나가 되었으며, 차세대 통신 및 데이터 저장 시스템에 오류 정정을 위해 적용할 수 있다[2], [3]. Polar code의 대표적인 복호 기법으로는 연속 제거(successive cancellation; SC) 복호 [1]가 있으며, 이를 기반으

로 연속 제거 리스트 (SC-List) 복호 [4], [5], 연속 제거 반전 (SC-Flip) 복호 [6], [7] 기법들이 개발되었다. 특히, 연속 제거 리스트 복호 기법은 짧은 길이에 polar code를 적용했을 때에도 의미 있는 오류 정정 성능을 보여주며 [4], [5], 5G 무선 통신에 적용되는 3GPP 표준으로 선정되었다[8]. 그러나 연속 제거 기반의 복호화 기법들 [1]-[7]은 순차적으로 복호가 진행되는 특성으로 인해 복호화 과정에 소요되는 지연시간이 길어지는 단점이 있으며, 이로 인해 실제 시스템에 적용하는 것에 한계가 존재한다[3].

최근 트리구조 하드웨어 [2]를 기반으로 한 연속 제거 복호를 효율적으로 수행하기 위한 가지치기 (pruning) 복호 기법 [9]-[13]과 다중-경로(multi-path) 복호 기법과 [14], [15]에 대한 연구가 진행되고 있다. 가지치기 기법을 사용하는 연속 제거 복호 기

반의 복호 기법 [9]-[13]은 복호 트리의 하위노드가 모두 고정비트 값을 갖거나 정보비트 값을 가지는 노드를 가지치기하는 SSC(simplified successive cancellation) 복호 기법 [9], [10], 복호 트리를 구성하는 노드의 종류를 세분화 하여 SSC 복호 보다 처리율을 높이는 fast-SSC 기법 [11], 신드롬 판단 (syndrome check)기법을 사용하여 가지치기를 수행하는 복호화 기법 [13] 등이 있다. 다중-경로 복호 기법 [14], [15]은 2-비트를 동시에 복호화 하는 2-비트 연속 제거 복호 기법 [14], 채널 LLR(log-likelihood ratio) 값의 수 표현을 달리하여 redundant-LLR 표현을 기반으로 하여 연속 제거 복호의 연산을 수행하는 PE (processing element) 회로를 개선한 하드웨어 [15] 등이 있다.

II. Polar code

Polar code의 입력 메시지 벡터 \mathbf{u} 는 고정 비트와 정보비트로 구성되며, 입력 메시지 벡터가 N -비트이고 정보비트가 K -비트라고 가정했을 때, 완전 노이즈(completely noisy) 채널에 고정비트 $N-K$ 개를 할당하고 완전무결(noise-free) 채널에 정보비트 K 개를 할당하여 부호화를 진행하며 (N, K) polar code로 표현한다[1]. Polar code의 부호화 (encoding)는 생성 행렬(generator matrix) \mathbf{G}_N 을 입력 메시지 벡터 \mathbf{u} 와 곱하여 부호 벡터 $\mathbf{x}=\mathbf{u}\mathbf{G}_N$ 를 생성한다. 생성 행렬 \mathbf{G}_N 은 행렬 \mathbf{F} 의 n ($n=\log_2N$) 크로네커 곱 (kronecker product) 으로 만들어지며, $\mathbf{F}=\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$, $\mathbf{G}_N=\mathbf{F}^{\otimes n}$ 로 표현한다.

Polar code로 부호화 된 메시지를 복호하는 대표적인 방법으로 연속 제거 복호가 있다 [1]. 연속 제거 복호 [1]는 수신 메시지 벡터 \mathbf{y} 로 채널 LLR $L_{j,i}$ 을 계산하는 과정을 순차적으로 진행하며, 부호 길이 N 이 8인 경우 복호과정을 그림으로 나타내면 그림 1 (a)와 같다. 가장 먼저 수신된 메시지 벡터 \mathbf{y} 를 사용하여 초기 채널 LLR 값 $L(u_i)=P(\mathbf{y}|x_i=0)/P(\mathbf{y}|x_i=1)$ 을 계산한다. 이후 각 복호 단계에서 채널 LLR $L_{j,i}$ 은 식 (1)로 계산한다.

$$L_{j,i} = \begin{cases} f(L_{j-1,i}, L_{j-1,i+2^{n-j}}), & i/2^j \text{ is even} \\ g(L_{j-1,i-2^{n-j}}, L_{j-1,i}, \hat{u}_j), & \text{otherwise} \end{cases}, \quad (1)$$

이때 $L_{j,i}$ 는 i 번째 복호 단계의 j 번째 노드의 채널

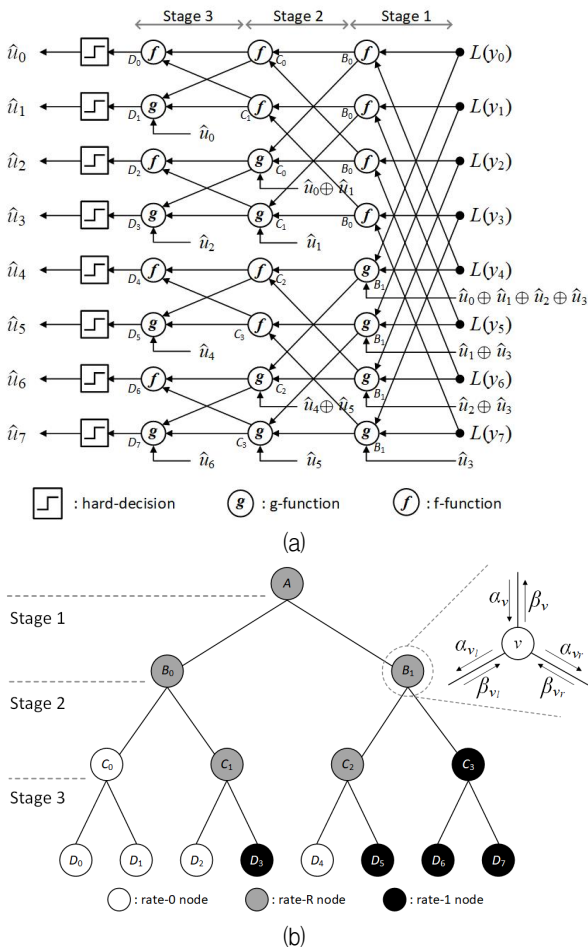


Fig. 1. (a) SC decoding procedure [1] and (b) decoding tree for (8, 4) polar code. 그림 1. (8, 4) polar code의 (a) 연속 제거 복호 과정 [1]과 (b) 복호 트리

LLR 값을 나타내며, $0 \leq i \leq N-1$, $1 \leq j \leq n$ 이고, \hat{u}_s 는 부분 합을 나타낸다. f 함수와 g 함수는 모두 입력으로 채널 LLR 값이 들어가며 g 함수는 부분 합 또한 입력으로 필요하다. f 함수와 g 함수를 식으로 나타내면 다음과 같다.

$$f(a,b) = 2 \tanh^{-1}(\tanh(a/2)\tanh(b/2)) \approx \text{sign}(a)\text{sign}(b)\min(|a|,|b|), \quad (2)$$

$$g(a,b,\hat{u}_s) = (1-\hat{u}_s)g^{(0)}(a,b) + (\hat{u}_s)g^{(1)}(a,b), \quad (3)$$

이때, a 와 b 는 채널 LLR 입력을 의미하고, \hat{u}_s 는 부분 합을 의미한다.

그림 1 (b)의 복호 트리는 연속 제거 복호 과정 [1]을 복호 트리로 나타낸 것으로, rate-0 노드는 하위노드가 모두 고정비트, rate-1 노드는 하위노드가 모두 정보비트인 노드, rate-R 노드는 하위노드가 정보비트와 고정비트로 구성된 노드를 의미한다. 그림 1 (b)에서 rate-0, rate-1, rate-R 노드는 각각 하얀색, 검은색, 회색으로 표현한다.

그림 1 (b)은 (8, 4) polar code는 정보비트의 인덱스를 3, 5, 6, 7로 가정하고 0, 1, 2, 4를 고정비트 인덱스로 가정했다. 그림 1 (b)의 α_v 는 채널 LLR을 의미하며, 노드 v 의 왼쪽 하위노드로 이동하는 채널 LLR은 $\alpha_v[i] = \alpha_v[2i] \oplus \alpha_v[2i+1]$ 이고, 오른쪽 하위노드로 이동하는 채널 LLR은 $\alpha_v[i] = \alpha_v[2i](1 - \beta_v[i]) + \alpha_v[2i+1]$ 이고, $0 \leq i \leq 2^{n-j}$ 이다. 노드 v 의 경관정 결과를 나타내는 β_v 는 $\beta_v[2i+1] = \beta_v[i]$ 와 $\beta_v[2i] = \beta_v[i] \oplus \beta_v[i]$ 로 결정된다. 그림 1 (b)의 복호 트리에서 B , C , D 노드에 해당하는 연산을 그림 1 (a)에도 나타내었다.

III. 연속 제거 복호기반의 복호기법 분석

연속 제거 복호의 대표적인 하드웨어 구조로 트리구조 하드웨어 [2]가 있으며, 부호 길이가 8인 polar code에 사용되는 트리구조 하드웨어는 그림 2 (a)와 같으며, 부호 길이 N 이 8인 경우이다. 복호 트리의 각 노드 연산에 필요한 PE(processing element)를 그림 2 (b)의 PE에 나타내었으며, 그림 2 (a)의 트리구조 하드웨어 [2]는 PE, 부분 합을 계산하기 위한 PSA(partial sum accumulating) 회로, 멀티플렉서, 경관정(hard-decision)을 위한 회로로 구성 된다. 트리구조 하드웨어 [2]를 사용한 복호

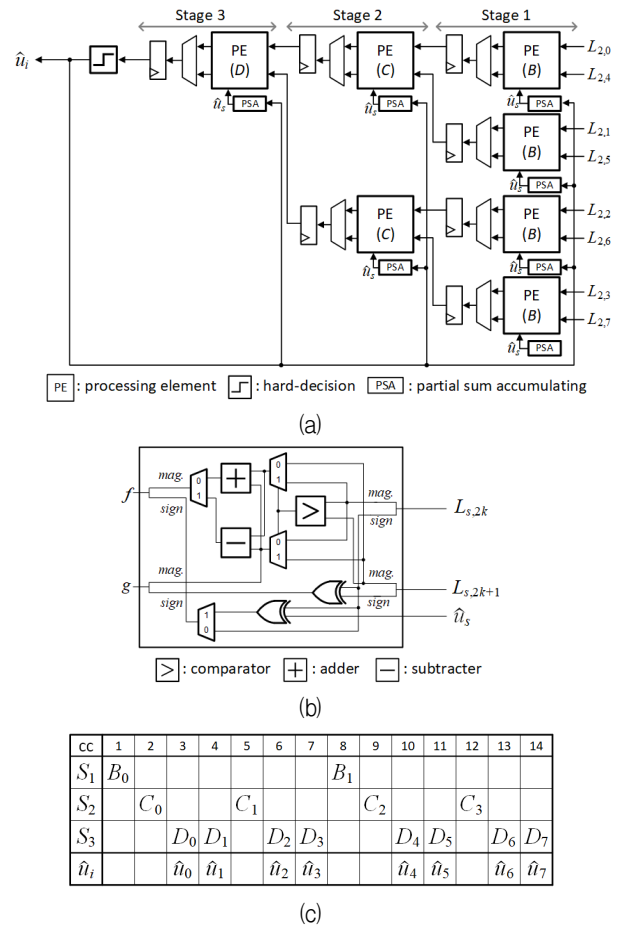


Fig. 2. (a) Tree architecture, (b) PE architecture, and (c) decoding schedule for polar SC decoder when $N = 8$ [2].

그림 2. $N = 8$ 일 때 polar code의 (a) 트리 구조 연속 제거 복호기, (b) 트리 구조 복호기의 PE, 그리고 (c) 복호 과정 [2]

과정은 그림 2 (b)와 같으며, 8-비트 복호에 14 클럭 사이클(clock cycle; cc)이 필요하다.

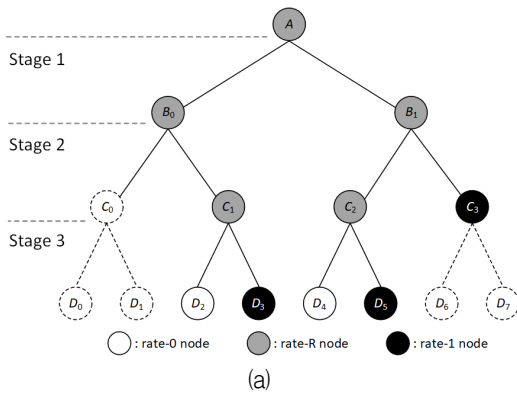
트리구조의 연속 제거 복호 [2]는 지연시간이 길고 하드웨어 면적이 크기 때문에, 이를 개선하기 위한 가지치기 혹은 다중-경로 복호 기법이 지속적으로 개발되고 있다[9]-[15].

1. 연속 제거 복호 기반의 가지치기 복호 기법

연속 제거 복호를 기반의 가지치기 복호 기법들이 다수 제안되었으며 [9]-[13], 대표적으로 SSC 복호 [9],[10], fast-SSC 복호 [11], 그리고 신드롬 판단 기반의 복호 [13]가 있다.

가. SSC 복호

트리 구조 기반의 SSC 복호 기법 [9], [10]은 부



cc	1	2	3	4	5	6	7	8
S_1	B_0			B_1				
S_2		C_1			C_2		C_3	
S_3			D_3			D_5		β_{C_3}, G_2
\hat{u}_i	\hat{u}_0, \hat{u}_1		\hat{u}_2, \hat{u}_3			\hat{u}_5		\hat{u}_6, \hat{u}_7

(b)

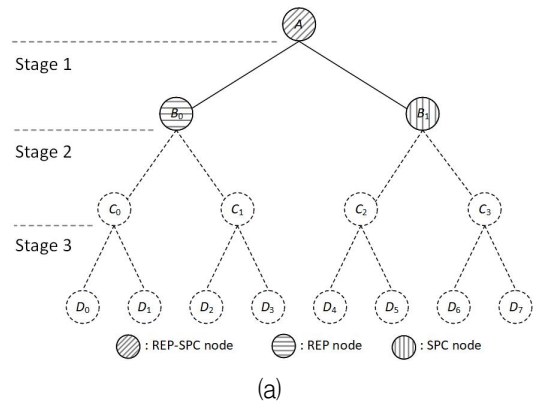
Fig. 3. (a) SSC decoder [9] and (b) decoding schedule [10] for (8, 4) polar code. 그림 3. (8, 4) polar code의 (a) SSC 복호기 [9]와 (b) 복호 과정 [10]

호 길이 N 이 8일 때 복호 트리로 나타내면 그림 3 (a)과 같다. SSC 복호기법은 rate-0 노드와 rate-1 노드를 가지치기하여 연속 제거 복호 과정의 지연 시간을 줄이며, 그림 3 (a)에서 점선으로 표현한 노드와 경로는 소거되는 노드와 경로를 표현하고 있다.

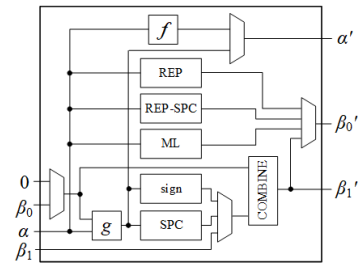
SSC 복호 [9]에서 노드가 소거되는 경우를 살펴보면 노드 v 가 rate-0 노드인 경우 하위노드가 모두 고정비트 이므로 노드 v 는 소거되며, 노드 v 가 rate-1 노드인 경우 하위노드가 모두 정보비트인 것이 명확하므로 β_v 를 계산하고, β_v 를 생성행렬 G_{N_j} 와 곱하여 $\{\hat{u}[\min A_v], \dots, \hat{u}[\max A_v]\}$ 를 결정하고 복호를 완료한다. 이때 N_j 는 j 스테이지에서 복호되는 비트 수를 의미한다. 노드 v 가 rate- R 노드이면 rate-0 혹은 rate-1 노드에 도달할 때까지 채널 LLR α_v 를 이용한 계산을 계속한다. 예를 들어, 그림 1 (b)와 같은 구성을 갖는 (8, 4) polar code의 복호를 진행하면 그림 3 (a)와 같이 스테이지 2의 C_0 노드와 C_3 의 하위노드는 소거되며, 복호 과정 [10]은 그림 3 (b)에 나타난 것과 같이 8 클럭 사이클에 8-비트 복호를 완료하여 트리구조 하드웨어 [2]보다 지연시간이 43% 줄어든다.

나. Fast-SSC 복호

SSC 복호 기법 [9], [10]를 사용했을 때 복호에 소



(a)



(b)

cc	1	2
S_1	B_0	B_1
\hat{u}_i	$\hat{u}_0, \hat{u}_1, \hat{u}_2, \hat{u}_3$	$\hat{u}_4, \hat{u}_5, \hat{u}_6, \hat{u}_7$

(c)

Fig. 4. (a) Fast-SSC decoder, (b) PE architecture, and (c) decoding schedule of fast-SSC for (8, 4) polar code [11].

그림 4. (8, 4) polar code의 (a) fast-SSC 복호 기법, (b) fast-SSC의 PE 구조 그리고 (c) 복호 과정 [11]

요되는 시간을 줄이기 위하여 fast-SSC 복호 [11]가 제안되었다. Fast-SSC 복호 [11]는 복호 트리를 구성하는 노드의 종류를 SSC 복호 [9], [10]보다 세분화하여 구분한 다음, 각 노드 종류에 따라 서로 다른 연산을 적용하여 복호를 진행한다[11]. 그림 4 (a)의 복호 트리도 그림 1 (b)와 같은 구성의 polar code를 가정했으며, rate-0 노드와 rate-1 노드는 SSC 복호 [9], [10]와 동일하나 rate- R 노드는 연산의 간소화를 위해 SPC(single-parity check) 노드와 REP(repetition) 노드, REP 노드와 SPC 노드가 각각 왼쪽 하위노드와 오른쪽 하위노드로 결합된 REP-SPC 노드로 다시 구분한다. SPC 노드는 연결된 하위노드들 가운데 하나의 노드만 고정비트이고 나머지 노드는 정보비트로 구성된 노드를 의미하며, REP 노드는 연결된 하위노드 가운데 최상위 비트만 정보비트이고 나머지는 고정비트로 구성된 노드를 의미한다.

Fast-SSC 복호 [11]를 나타내는 그림 4 (a)의 (8, 4) polar code의 복호 트리에서 0, 1, 2, 3번째 비트가 연결된 B_0 노드는 3번째 비트만 정보비트 이므로 REP 노드, 4, 5, 6, 7번째 비트가 연결된 B_1 노드는 4번째 비트만 고정비트 이므로 SPC 노드이고, B_0 노드와 B_1 노드를 하위노드로 갖는 A 노드는 REP-SPC 노드이다. 점선으로 표시된 노드의 소거를 통해 연산하지 않아도 되는 노드를 나타낸다. 그림 4 (a)와 그림 3 (a)를 비교하면 소거되는 노드의 수가 fast-SSC 복호 기법 [11]이 SSC 복호 기법 [9]보다 더 많은 것을 알 수 있다. 기본 복호과정은 SSC 복호 [9], [10]의 처리율보다 80% 향상된 처리율을 가지는 ML(maximum likelihood)-SSC [12]을 사용한다.

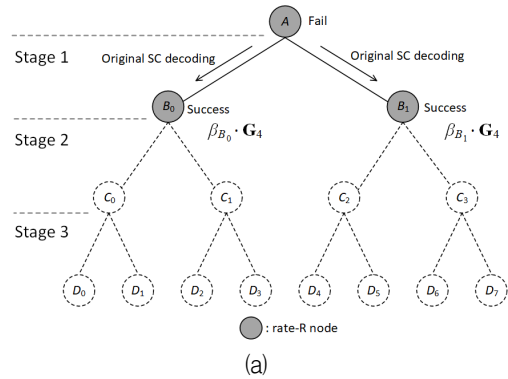
Fast-SSC 복호를 위한 전체 하드웨어는 트리구조 하드웨어 [2]와 동일하나 PE 내부 회로 구조는 그림 4 (b)와 같으며, f 함수와 g 함수를 위한 회로, 각 노드의 연산을 위한 회로, ML 연산을 위한 회로, 그리고 β_i 과 β_r 의 결합을 위한 COMBIME 회로로 구성된다[11].

그림 4 (c)는 fast-SSC 복호 [11]의 복호 과정을 나타내며, REP 노드인 B_0 노드의 연산을 수행하면 3번째 비트 값을 복호하고, SPC 노드인 B_1 노드의 연산을 수행하면 고정비트를 제외한 5, 6, 7번째 비트 값을 알 수 있다. 0, 1, 2, 4번째의 고정비트 값은 이미 결정되어 있으므로 복호 할 필요가 없다. Fast-SSC 복호 [11]는 2 클럭 사이클 만에 복호가 종료되어 트리구조 하드웨어 [2]의 복호과정과 비교하면 지연시간이 72% 줄어든다.

다. 신드롬 판단 기반의 복호

SSC 복호 [9],[10]와 마찬가지로 노드 가지치기를 통해 연속 제거 복호 [1]를 간소화하는 복호 기법으로 노드 가지치기를 위해 신드롬 판단을 사용하는 복호 기법이다[13]. 신드롬 판단 기반의 연속 제거 복호 [13]를 위한 기본 구조는 그림 5 (a)에서 보이는 바와 같이 SSC 복호 [9], [10]와 동일하며, (8, 4) polarcode를 예시로 그림 5 (a)에 묘사하였다. 신드롬 판단을 위해서 생성행렬 \mathbf{G} 의 역행렬인 \mathbf{H} -행렬이 필수적인데, \mathbf{G} 의 역행렬은 $\mathbf{G}_N^{-1}=\mathbf{G}^N$ 이다[13].

수신된 메시지 벡터 \mathbf{y} 로 계산한 초기 채널 LLR 값을 가지고 있는 최상위 노드 A 에서 N -비트 \mathbf{H} -



cc	1	2
S_1	B_0	B_1
\hat{u}_i	$\hat{u}_0, \hat{u}_1, \hat{u}_2, \hat{u}_3$	$\hat{u}_4, \hat{u}_5, \hat{u}_6, \hat{u}_7$

(b)

Fig. 5. (a) Syndrome check SC decoder and (b) decoding schedule for (8, 4) polar code [13]. 그림 5. (8, 4) polar code의 (a) 신드롬 판단 복호기와 (b) 복호 과정 [13]

행렬을 사용하여 신드롬 판단을 수행하여 신드롬 판단의 성공여부를 판단한다. 만약 A 노드에서 신드롬 판단이 성공하면, v 노드의 β_v 를 \mathbf{G}_v 과 곱하여 복호를 완료하고, 실패했다면 원래의 연속 제거 복호 [1]를 수행하여 스테이지 1로 이동한다. 모든 비트에 대한 복호를 완료할 때까지 스테이지 1부터 노드 A 와 동일한 과정을 반복하여 복호를 진행하며, 이때 사용하는 \mathbf{H} -행렬과 생성행렬 \mathbf{G} 는 N_j -비트에 대한 \mathbf{H} -행렬 \mathbf{G}_{N_j} 를 사용한다.

부호 길이 N 이 8이고 고정비트 인덱스를 그림 1 (b)와 같이 0, 1, 2, 4로 가정한 그림 5 (a)의 복호 트리에서 A 노드에서 신드롬 판단이 실패했다고 가정하면, 원래의 연속 제거 복호 [1]를 수행하여 B_0 노드와 B_1 노드로 이동하여 신드롬 판단을 한다. B_0 노드와 B_1 노드에서 신드롬 판단에 성공했다면, β_{B_0} 와 β_{B_1} 를 각각 \mathbf{G}_4 와 곱하여 복호를 완료한다. 그림 5 (b)는 복호과정을 나타내며, 8-비트 복호를 완료하는데 5 클럭 사이클이 소요되어 신드롬 판단 기반의 복호 [13]를 트리구조 하드웨어 [2]와 비교했을 때 지연시간이 72% 줄어든다.

2. 연속 제거 복호 기반의 다중-경로 복호 기법

Polar code의 전통적인 연속 제거 복호 [1] 기반의 다중-경로 복호 기법들이 제안되었으며 [14], [15], 대표적으로 2-비트 연속 제거 복호 [14]와 redundant-LLR 표현 기반의 복호 [15]가 있다.

가. 2-비트 연속 제거 복호

2-비트 연속 제거 복호는 전체 복호과정의 마지막 복호 단계인 스테이지 n 에서 2-비트 복호를 동시에 진행하는 복호 기법이다[14]. 스테이지 $n-1$ 까지는 기존 연속 제거 복호의 트리구조 하드웨어 [2]에 사용된 PE와 동일한 PE를 사용하여 복호한다. 그러나 스테이지 n 에서 기존 PE 대신에 2-비트를 동시에 경관정하는 모듈 P를 사용하여 복호과정의 지연시간을 단축시켰다.

스테이지 n 에서 복호하는 2-비트의 조합은 (고정비트, 고정비트), (정보비트, 고정비트), (고정비트, 정보비트), 그리고 (정보비트, 정보비트)이다. 두 개의 비트 모두 고정비트 인 경우, 두 개의 비트 모두

바로 복호가 가능하다. 만약 \hat{u}_{2i+1} 비트만 고정비트 라면, $L_{j-1,2i}+L_{j-1,2i+1} \geq 0$ 인 경우 \hat{u}_{2i} 가 0이고, 그렇지 않은 경우 1이다. 만약 \hat{u}_{2i} 만 고정비트 라면, $\text{sign}(L_{j-1,2i})\text{sign}(L_{j-1,2i+1}) \geq 0$ 라면, \hat{u}_{2i+1} 가 0이고, 그렇지 않은 경우 1이다. \hat{u}_{2i} 과 \hat{u}_{2i+1} 2-비트 모두 정보비트 라면, $\text{LLR}(\hat{u}_{2i}=0, \hat{u}_{2i+1}=0) \triangleq L_{j-1,2i} + L_{j-1,2i+1}$, $\text{LLR}(\hat{u}_{2i}=0, \hat{u}_{2i+1}=1) \triangleq 0$, $\text{LLR}(\hat{u}_{2i}=1, \hat{u}_{2i+1}=0) \triangleq 0$, $\text{LLR}(\hat{u}_{2i}=1, \hat{u}_{2i+1}=1) \triangleq L_{j-1,2i}$ 로 결정된다.

2-비트를 동시에 경관정하는 P 모듈은 트리구조 하드웨어 [2]를 비롯한 부분-병렬(semi-parallel) 구조 [3], 라인 (line) 구조 [2] 등 다양한 구조의 마지막 복호단계인 스테이지 n 의 PE에 적용 가능 하며, 그림 6 (a)는 (8, 4) polar code일 때 트리구조 하드웨어 [2]에 2-비트를 동시에 경관정 하는 모듈 P [14]를 적용한 하드웨어를 나타내고, 이때 PE는 그림 2 (b)의 트리구조 하드웨어 [2]의 PE와 동일하며, 최종 복호 단계의 P는 그림 6 (b)와 같이 구성할 수 있다. 2-비트 연속 제거 복호를 적용한 트리구조 하드웨어 [14]의 복호 과정은 그림 6 (c)와 같 으며, 8-비트 복호에 10 클럭 사이클이 소요되어 기존 트리구조 하드웨어 [2]를 사용한 경우보다 지연시간이 29% 감소한다.

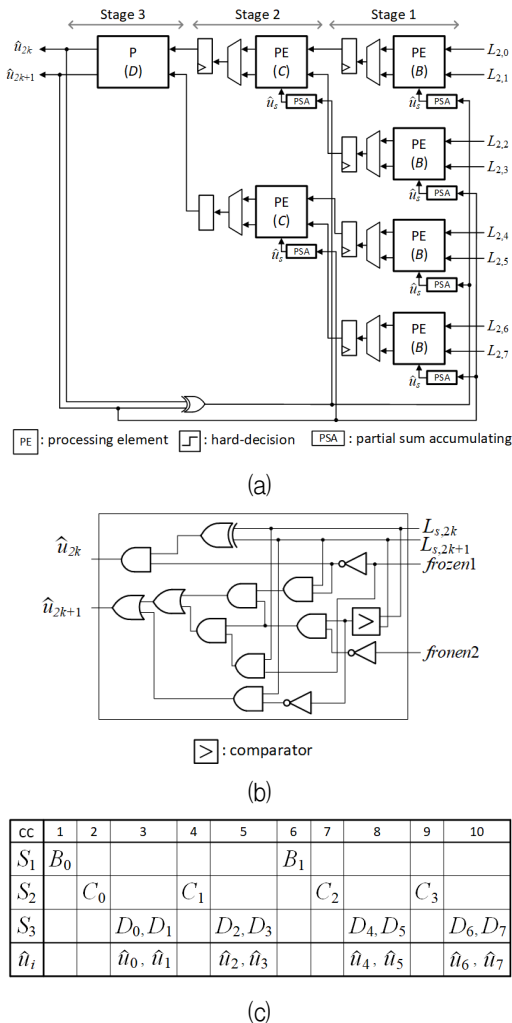


Fig. 6. 2-bit SC polar decoder (a) hardware architecture, P architecture, and (b) decoding scheduling when $N = 8$ [14].

그림 6. $N = 8$ 일 때, polar code의 2-비트 연속 제거 복호기의 (a) 하드웨어 구조, (b) P 구조, 그리고 (c) 복호 과정 [14]

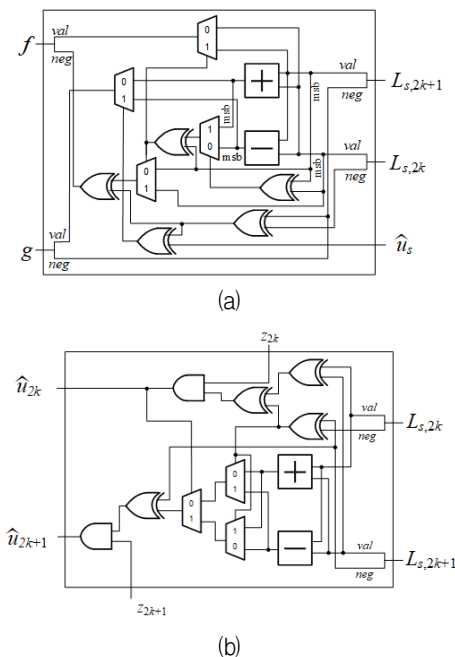


Fig. 7. (a) PE architecture and (b) P architecture of Redundant-LLR SC decoder for polar code when $N = 8$ [15]. 그림 7. $N = 8$ 일때 redundant-LLR 기반의 연속 제거 복호기의 (a) PE 구조와 (b) P 구조 [15]

나. Redundant-LLR 표현 기반의 복호

그림 2 (a)의 트리구조 하드웨어 [2]와 그림 6 (a)의 2-비트 연속 제거 복호 하드웨어 [14]의 PE는 부호화-절댓값으로 수 표현이 된 LLR 값을 사용하여 계산하였다. 그러나 g 함수를 계산할 때에는 부호화-절댓값으로 표현된 LLR 값을 사용하면 g 함수 계산에 소요되는 시간이 길어지며, 2의 보수(2's complement) 표현이 g 함수 계산에 소요되는 시간을 줄일 수 있다. 따라서 부호화-절댓값 표현을 사용하는 PE의 단점을 보완하기 위한 redundant-LLR 표현 기반의 복호기법 [15]은 2의 보수로 표현된 채널 LLR과 채널 LLR의 negating 비트를 사용하여 PE를 설계하였으며, redundant-LLR 표현 기반의 복호의 복호과정과 복호기의 구조는 2-비트 연속 제거 복호 [14]와 동일하다.

2의 보수 표현과 negating 비트를 사용한 LLR 계산에 필요한 PE와 P의 하드웨어는 각각 그림 7 (a)와 (b)로 구성할 수 있으며, 그림 7의 PE는 그림 2 (a)의 부호화-절댓값 표현을 위한 PE에 필요한 비교기(comparator) 대신에 XOR 게이트로 연산을 진행하여 하드웨어 복잡도가 감소한다.

부호 길이 N 이 1024인 경우 연속 제거 복호에 필요한 하드웨어 면적은 redundant-LLR 표현을 위한 PE를 적용한 경우 부호화-절댓값 표현을 위한 PE를 적용한 경우보다 25% 감소한다.

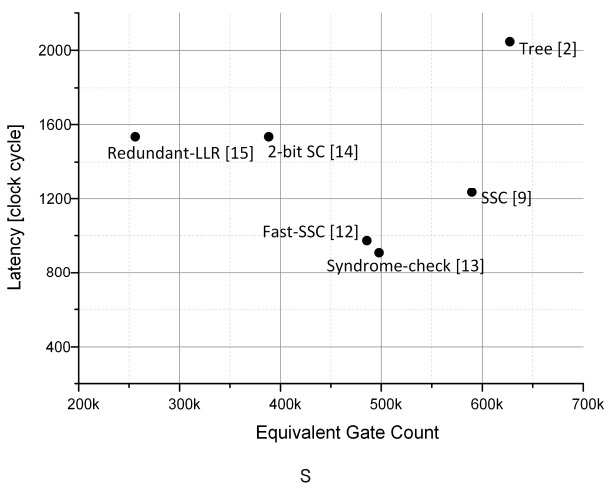


Fig. 8. Comparison chart of latency and equivalent gate count for different SC polar decoders when $N = 1024$.

그림 8. 부호 길이 $N = 1024$ 일 때, 연속 제거 복호기의 지연시간과 gate count 비교

Table 1. Comparison of the recent SC decoding.

표 1. 최신 극 부호 복호기의 비교

Algorithm	[9]	[12]	[13]	[14]	[15]
Type	Pruning	Pruning	Pruning	Multi-path	Multi-path
Pruning Ratio	Low	High	High	N.A.	N.A.
Hardware	Large	Moderate	Moderate	Moderate	Small
Latency	High	Low	Low	Moderate	Moderate

III. 결론

Polar code의 대표적인 복호 기법인 연쇄 제거 복호를 위한 여러 하드웨어 구조 중 트리구조 하드웨어[2]의 지연시간과 하드웨어 면적을 줄이고자 다양한 복호 기법들이 지속적으로 발표되었다 [9]-[15]. 본 논문에서는 발표된 연속 제거 복호 기반의 복호 기법들을 가지치기 복호 기법 [9]-[13]들과 다중-경로 복호 기법 [14], [15]들로 나누어 하드웨어 구조와 복호과정을 정리하였다.

표 1은 본 논문에서 비교한 연속 제거 복호 기반의 복호 기법 [9]-[15]들을 비교한 것으로 Fast-SSC [12]와 신드롬 판단 복호 [13]가 노드의 가지치기 비율이 가장 높고 지연시간이 짧으며, redundant-LLR [15]의 하드웨어 면적이 가장 작다. 그림 8은 트리 구조의 하드웨어 [2]와 본 논문에서 정리한 5가지 복호 기법들 [9]-[15]의 부호 길이 N 이 1024인 polar code의 연속 제거 복호를 위한 하드웨어의 면적과 지연시간을 비교하여 그래프로 나타낸 것으로 트리구조 하드웨어 [2]가 복호 하드웨어 면적과 지연시간이 가장 크며, 이를 제외하면 신드롬 판단 복호 [13]의 지연시간이 가장 짧고, redundant-LLR 표현 기반의 복호 [15]가 하드웨어가 가장 작은 복호 기법이다. 따라서 그림 8의 결과를 기반으로 하여 polar code가 사용될 환경의 제약조건에 따라 polar code의 연속 제거 복호기의 하드웨어 구조를 주어진 제약조건에 가장 적합한 것으로 선택할 수 있다.

References

[1] E. Arıkan, "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels,"

IEEE Transactions on Information Theory, vol.55, no.7, 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,” *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 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 Transactions on Signal Processing*, vol.61, no.2, pp.289–299, 2013.

DOI: 10.1109/TSP.2012.2223693.

[4] I. Tal and A. Vardy, “List Decoding of Polar Codes,” *IEEE Transactions on Information Theory*, vol.61, no.5, pp.2213–2226, 2015.

DOI: 10.1109/TIT.2015.2410251.

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

[6] O. Afisiadis, A. Balatsoukas-Stimming and A. Burg, “A low-complexity improved successive cancellation decoder for polar codes,” *2014 48th Asilomar Conference on Signals, Systems and Computers*, pp.2116–2120, 2014.

DOI: 10.1109/ACSSC.2014.7094848

[7] L. Chandesaris, V. Savin and D. Declercq, “An Improved SCFlip Decoder for Polar Codes,” *2016 IEEE Global Communications Conference (GLOBECOM)*, pp.1–6, 2016. DOI: 10.1109/GLOCOM.2016.7841594

[8] 3GPP, “5G: Study on new radio (NR) access technology,” 3GPP TS 38.212 v.15.0.0, 2017.

[9] A. Alamdar-Yazdi and F. R. Kschischang, “A Simplified Successive-Cancellation Decoder for Polar Codes,” *IEEE Communications Letters*, vol.15, no.12, pp.1378–1380, 2011.

DOI: 10.1109/LCOMM.2011.101811.111480.

[10] C. Zhang and K. K. Parhi, “Latency Analysis and Architecture Design of Simplified SC Polar Decoders,” *IEEE Transactions on Circuits and*

Systems II: Express Briefs, vol.61, no.2, pp.115–119, 2014. DOI: 10.1109/TCSII.2013.2291065.

[11] G. Sarkis, P. Giard, A. Vardy, C. Thibault and W. J. Gross, “Fast Polar Decoders: Algorithm and Implementation,” *IEEE Journal on Selected Areas in Communications*, vol.32, no.5, pp.946–957, 2014. DOI: 10.1109/JSAC.2014.140514.

[12] G. Sarkis and W. J. Gross, “Increasing the Throughput of Polar Decoders,” *IEEE Communications Letters*, vol.17, no.4, pp.725–728, 2013.

DOI: 10.1109/LCOMM.2013.021213.121633.

[13] H. Yoo and I. Park, “Efficient Pruning for Successive-Cancellation Decoding of Polar Codes,” *IEEE Communications Letters*, vol.20, no.12, pp.2362–2365, 2016.

DOI: 10.1109/LCOMM.2016.2607167.

[14] 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.

[15] H. Yoon and T. Kim, “Efficient Successive-Cancellation Polar Decoder Based on Redundant LLR Representation,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol.65, no.12, pp.1944–1948, 2018.

DOI: 10.1109/TCSII.2018.2811378.

BIOGRAPHY

Soyeon Choi (Member)



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

2018~ : Unified MS and Ph.D. 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 : Researcher, Samsung Electronics

2016~ : Assistant Professor, Chungnam National University