

## METHODS

# A Low-Complexity Sorting Network for a Fast List Polar Decoder

YONGJE LEE, (Graduate Student Member, IEEE), JAE HONG ROH,  
USEOK LEE<sup>id</sup>, (Member, IEEE), AND MYUNG HOON SUNWOO<sup>id</sup>, (Fellow, IEEE)

Department of Electrical and Computer Engineering, Ajou University, Suwon 16499, South Korea

Corresponding author: Myung Hoon Sunwoo (sunwoo@ajou.ac.kr)

This work was supported in part by the National Research and Development Program through the National Research Foundation of Korea (NRF) Funded by the Ministry of Science and Information and Communications Technology (MSIT) under Grant NRF-2021R1A2C2010228; in part by the MSIT, South Korea, through the Information Technology Research Center (ITRC) Support Program Supervised by the Institute for Information and Communications Technology Planning and Evaluation (IITP), under Grant IITP-2021-2020-0-01461; and in part by the EDA Tool was supported by the IC Design Education Center (IDEC), South Korea.

**ABSTRACT** Fast list decoding algorithms for polar codes have been proposed to achieve low latency and high error correction performance. In the Rate-1 and SPC nodes of fast list decoding, a high-complexity sorter is required if multiple bits are to be decoded simultaneously in a single clock cycle. This paper proposes a partitioned sorting network (PSN) that reduces the complexity of metric sorter in Rate-1 and single parity check (SPC) decoders for fast list decoding. The proposed PSN consists of two sorting networks. For the first sorting network, we analyze the order of the path metrics (PMs) and reduce the number of input candidate paths. Furthermore, for the second sorting network, we analyze the cumulative results of path selection and identify relatively frequently selected paths. The proposed PSN has up to 90% fewer compare-and-swap units (CASUs) and up to 137% higher operating frequencies than existing sorting networks for Rate-1 and SPC decoders with  $L = 8$ .

**INDEX TERMS** Polar codes, fast simplified successive cancellation list decoding, metric sorter.

## I. INTRODUCTION

Polar codes [1] were the first error correction codes that achieved channel capacity in a binary-input discrete memoryless channel. Polar codes have been adopted as the basis of a coding scheme for control channels in the 5G New Radio standard. However, the basic decoding algorithm for polar codes, called successive cancellation (SC) decoding, has two significant disadvantages. The first is the long latency due to the bit-sequential decoding process [2]. The second is to have lower error correction performance compared to other error-correcting codes, such as low-density parity-check (LDPC) and turbo codes, at the same code length.

Several decoding algorithms [3], [4], [5] have been proposed to overcome the first disadvantage, i.e., the long latency. The 2-b SC decoding algorithm in [3] can decode 2 bits simultaneously in the last stage of the polar decoding tree. In [4], the simplified SC (SSC) decoding algorithm

reduces the number of decoding clock cycles by using multi-bit decisions for Rate-0 and Rate-1 nodes. Furthermore, [5] proposed a fast SSC (FSSC) algorithm using two other node types: repetition (REP) nodes, with only an information bit in the last bit position, and single parity check (SPC) nodes, with only a frozen bit in the first-bit position.

Several algorithms with high error correction performance [6], [7] have been proposed to address the second disadvantage of the SC algorithm. In particular, the SC list (SCL) decoding algorithm [6] considers  $L$  decoding paths, where  $L$  is the list size, and each path is independently decoded. To prevent the number of paths from continuously growing, a path metric (PM) operation is performed for path pruning. Meanwhile, the list size  $L$  can be increased to improve the error correction performance of the SCL decoding algorithm [6]. However, the complexity of the metric sorter sharply increases with increasing  $L$  in hardware implementation.

A hardware architecture for SCL decoders is proposed in [8] and confirmed that the critical path occurs in the metric sorter. Thus, the metric sorter can determine the

The associate editor coordinating the review of this manuscript and approving it for publication was Oussama Habachi<sup>id</sup>.

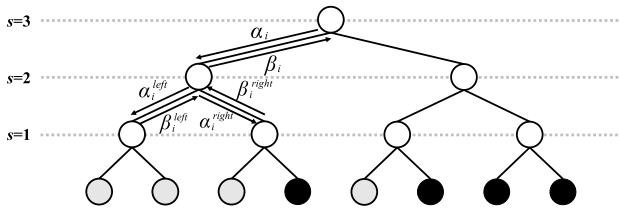


FIGURE 1. SC decoding tree for  $PC(8, 4)$ .

maximum operating frequency of an SCL decoder. Several sorting architectures for SCL decoders were proposed in [9], [10], [11], [12], and [13]. A pruned bitonic sorter (PBS) and a simplified bubble sorter (SBS) are proposed to sort PMs of the SCL decoder efficiently in [9]. An odd-even sorter (OES) is proposed to reduce unnecessary sorting operations in [10]. Various pairwise metric sorting (PMS) networks were proposed in [11]. Moreover, [12] and [13] proposed the pruned bitonic extractors (PBEs). The designs presented in [9], [10], [11], [12], and [13] can be applied in SCL decoders for  $2L$ -to- $L$  sorting.

A list-fast-SSC decoding algorithm is proposed to limit the number of candidate paths to  $M$  in [14], where  $M$  is the number of expanded candidate paths. Furthermore, [15] proposed a pruning algorithm that can reduce  $M \times L$  candidates according to  $M$  and its sorting network. The sorting network proposed in [15] is based on  $8L$ -to- $L$  path selection for list-fast-SSC decoding. However, in Rate-1 and SPC nodes for a list-fast-SSC decoding [14], error correction performance is not guaranteed compared to SCL decoding.

The SSC list (SSCL) and SSCL-SPC decoding algorithms [16], [17] achieve low latency and high error correction performance by combining the merits of the FSSC algorithm [5] and the SCL algorithm [6]. Furthermore, the Fast SSCL (FSSCL) and FSSCL-SPC algorithms [18], [19] effectively reduced the time steps by considering only bits with low reliability, not all bits in Rate-1 and SPC nodes, based on [18] and [19].

Reference [20] proposed a minimum-combinations set to speed up the splitting process of Rate-1 decoder for fast list decoding. The minimum-combination set optimizes the candidate paths in Rate-1 nodes by excluding unnecessary candidate paths that will not be selected. Fast list decoding algorithms with a minimum-combinations set [20] requires a single time step to decode a Rate-1 node.

In Rate-1 and SPC decoders for fast list decoding,  $2^\tau L$ -to- $L$  sorters are needed to decode  $\tau$  bits simultaneously in a single time step. However, this method dramatically increases the hardware complexity of the metric sorter and the propagation delay. Therefore, an improved metric sorter is needed for a faster decoder.

This paper proposes a PSN, where two kinds of sorting networks are used, for Rate-1 and SPC decoders of fast list decoding. Based on [20], the first sorting network is proposed for reducing the number of candidate paths in Rate-1 and SPC nodes. Furthermore, the second sorting network is proposed

through path selection ratio analysis. Finally, the proposed sorting network is implemented in hardware and compared with sorters for SCL-based decoders. The proposed PSN reduced CASU by up to 90% and increased operating frequency by up to 137% compared to existing sorting networks for  $L = 8$ .

The rest of this paper is organized as follows. Section II introduces polar codes and their decoding algorithms. Section III proposes a novel path selection method and a simplified sorting network. Section IV presents the hardware implementation results and comparisons, and Section V concludes the paper.

## II. PRELIMINARIES

### A. POLAR CODES AND SC DECODING

A polar code [1] of code rate  $R = K/N$  is represented by  $PC(N, K)$ , where  $N$  and  $K$  are the code length and the number of information bits, respectively.  $PC(N, K)$  is constructed using a generator matrix  $\mathbf{G}_N = \mathbf{F}_2^{\otimes n}$ , where  $n$  is  $\log_2 N$  and  $\mathbf{F}_2^{\otimes n}$  is the  $n^{\text{th}}$  Kronecker power of the matrix  $\mathbf{F}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ . The codeword  $\mathbf{x}$  obtained through the encoding process is expressed as  $\mathbf{x} = \mathbf{u} \cdot \mathbf{G}_N$ .

The SC decoding algorithm [1] can be represented through a decoding tree. Fig. 1 shows an SC decoding tree when  $N = 8$  and  $R = 1/2$ . In Fig. 1,  $s$  represents stages, and a node length  $N_v$  for each node is  $2^s$ . There are two kinds of messages,  $\alpha$  and  $\beta$ , in the decoding tree.  $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_{N_v}\}$  are LLR messages, which are passed from parent nodes to child nodes. The LLR messages  $\alpha_i^{\text{left}}$  and  $\alpha_i^{\text{right}}$  are calculated as

$$\alpha_i^{\text{left}} = \text{sgn}(\alpha_i) \text{sgn}(\alpha_{i+N_v/2}) \min(|\alpha_i|, |\alpha_{i+N_v/2}|). \quad (1)$$

$$\alpha_i^{\text{right}} = \alpha_{i+N_v/2} + (1 - 2\beta_i)\alpha_i. \quad (2)$$

$\beta = \{\beta_1, \beta_2, \dots, \beta_{N_v}\}$  are bit estimates and are determined by a partial sum operation of pre-decoded bits.  $\beta_i$  is passed from child nodes to parent nodes and is computed as

$$\beta_i = \begin{cases} \beta_i^{\text{left}} \oplus \beta_i^{\text{right}}, & \text{if } i < \frac{N_v}{2}, \\ \beta_{i-\frac{N_v}{2}}^{\text{right}}, & \text{otherwise.} \end{cases} \quad (3)$$

At a leaf node, a hard decision on the  $i$ -th bit is determined as

$$\hat{u}_i = \begin{cases} 0, & \text{if } i \in A^C \text{ or } \alpha_i \geq 0, \\ 1, & \text{otherwise,} \end{cases} \quad (4)$$

where  $A^C$  is the set of frozen bits.

### B. SCL DECODING

SCL decoding [6] was proposed to improve the performance of SC decoding by considering opposite decision results for information bits. Accordingly, each candidate codeword, called a path, is duplicated every time an information bit is decoded. However, each path requires an individual SC

decoding core, and it is unrealistic to decode all  $2^K$  possible paths simultaneously. To solve this problem, the SCL decoding algorithm limits its consideration to only  $L$  reliable paths; thus, a path selection process is needed for pruning the number of paths from  $2L$  to  $L$ .

In SCL decoding [6], every path has a PM. The PM is a non-negative real number used in the path selection process. Each path is duplicated in the SCL decoder, and correctly decoded bits and incorrectly decoded bits are stored separately. Two corresponding copied paths share their PM, but the absolute value of the corresponding log-likelihood ratio (LLR) is added as a penalty to the opposite path. All frozen bits are decoded as 0, and paths are not duplicated for frozen bits. In an LLR-based SCL decoder [8], the PM is calculated as given in (5). Here,  $PM_{i-1}^l$  is the PM of the  $l$ -th path before decoding, and  $PM_i^l$  denotes the PM of the  $i$ -th bit for the  $l$ -th path.

$$PM_i^l = \begin{cases} PM_{i-1}^l, & \text{if } \hat{u}_i^l = \frac{1}{2} (1 - \text{sign}(\alpha_i^l)), \\ PM_{i-1}^l + |\alpha_i^l|, & \text{otherwise.} \end{cases} \quad (5)$$

where  $\alpha_i^l$  is the  $i$ -th LLR of the node on the  $l$ -th path and  $\hat{u}_i^l$  is the bit decision for  $\alpha_i^l$ .

### C. FAST LIST DECODING

The FSSCL decoding algorithm [18], proposed based on the FSSC [5] and SCL [8] decoding algorithms, has low latency but similar error correction performance compared to SCL decoding. Various special nodes are distinguished by their arrangements of information bits and frozen bits: Rate-0, Rate-1, REP, and SPC nodes [5].

- 1) *Rate-0 Node*: Path splitting is not performed when a Rate-0 node is decoded because it has no information bits. Therefore, the number of paths remains the same. In Rate-0 nodes, the PM can be calculated as

$$PM_{N_v}^l = \frac{1}{2} \sum_{i=1}^{N_v} \text{sgn}(\alpha_i^l) \alpha_i^l - \alpha_i^l. \quad (6)$$

- 2) *REP Node*: The number of candidate paths increases to  $2L$  because an REP node has only one information bit. In REP nodes, the PM can be calculated in accordance with the corresponding bit estimate  $\beta_{N_v}^l$ , as follows:

$$PM_{N_v}^l = \frac{1}{2} \sum_{i=1}^{N_v} \text{sgn}(\alpha_i^l) \alpha_i^l - \beta_{N_v}^l \alpha_i^l. \quad (7)$$

The numbers of information bits for SPC and Rate-1 nodes are  $N_v - 1$  and  $N_v$ , respectively. Thus, the total numbers of candidate paths are at most  $2^{N_v - 1}$  and  $2^{N_v}$  in SPC and Rate-1 nodes, respectively. However, [18] and [19] proved that for SPC and Rate-1 nodes, it is necessary to consider at least  $\min(L, N_v)$  and  $\min(L - 1, N_v)$  less reliable bits, respectively, to ensure the error correction performance of the SCL decoder.

- 3) *SPC Node*: The single parity of the  $l$ -th path,  $\gamma^l$ , is calculated as shown in (8), and the PMs are initialized as shown in (9).

$$\gamma^l = \bigoplus_{i=1}^{N_v} \left( \frac{1}{2} (1 - \text{sign}(\alpha_i^l)) \right), \quad (8)$$

$$PM_0^l = \begin{cases} PM_{-1}^l + |\alpha_{i_{\min}}^l|, & \text{if } \gamma^l = 1, \\ PM_{-1}^l, & \text{otherwise.} \end{cases} \quad (9)$$

$i_{\min}$  is the least reliable bit index among the SPC nodes.  $PM_{-1}^l$  is the PM of the  $l$ -th path in the previous decoding step. Using (8) and (9), the PMs of the candidate paths are calculated as shown in (10).

$$PM_i^l = \begin{cases} PM_{i-1}^l + |\alpha_i^l| + (1 - 2\gamma^l) |\alpha_{i_{\min}}^l|, & \text{if } \beta_i^l \neq \frac{1 - \text{sgn}(\alpha_i^l)}{2}, \\ PM_{i-1}^l, & \text{otherwise.} \end{cases} \quad (10)$$

When the PM computation is finished, the least reliable bit is set by the even-parity constraint as follows:

$$\beta_{i_{\min}}^l = \bigoplus_{\substack{i=1 \\ i \neq i_{\min}}}^{N_v} \beta_i^l \quad (11)$$

- 4) *Rate-1 Node*: To decode a Rate-1 node, the PMs of the candidate paths are determined as shown in (12).

$$PM_i^l = \begin{cases} PM_{i-1}^l + |\alpha_i^l|, & \text{if } \beta_i^l \neq \frac{1 - \text{sgn}(\alpha_i^l)}{2}, \\ PM_{i-1}^l, & \text{otherwise.} \end{cases} \quad (12)$$

In [22], a parallel PM computing method is proposed to decode Rate-1 nodes in a single time step. Using a parallel PM computing method [22], in Rate-1, there are  $2^{\min(L-1, N_v)} L$  paths possibly considered as candidates. Thus, a  $2^{\min(L-1, N_v)} L$ -to- $L$  sorter is needed in Rate-1 to compute PMs in a single time step. This method can be applied to the PM calculation (10) of the SPC nodes. The SPC decoder in which the parallel PM computing method is applied, requires a  $2^{\min(L-1, N_v-1)} L$ -to- $L$  sorter.

### III. PROPOSED SORTING NETWORK

This section proposes methods to improve the PM sorting network for Rate-1 and SPC decoders in fast list decoding. The PM sorting network of Rate-1 and SPC decoders must consider a much larger number of path candidates than that of an SCL decoder. Therefore, the PM sorter of Rate-1 and SPC decoders can cause the critical path and decrease the operating frequency of the overall decoder. In addition, the area of the metric sorter and its computational complexity are drastically increased. To alleviate this problem, we propose a PSN considering the PMs provided as input to the sorter and the path selection ratio. Rate-1 and SPC decoders using the proposed PM sorting network shows an error correction performance similar to that of a conventional SCL decoder.

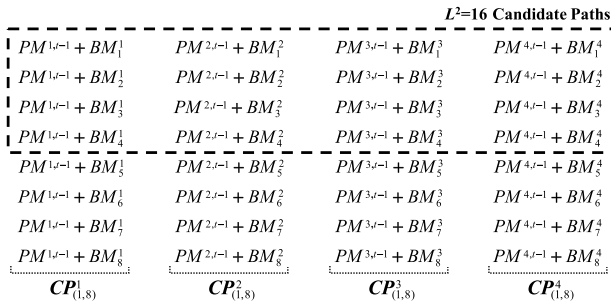


FIGURE 2. Candidate paths in Rate-1 and SPC Nodes with  $L = 4$ .

**A. PARTITIONED SORTING NETWORK**

Since a Rate-0 node has no information bits, path splitting is not performed. A REP node has only one information bit and requires only  $2L$  paths. Thus, metric sorters for Rate-0 nodes and REP nodes can be implemented using the existing methods [9], [10], [11], [12], [13]. However, when the Rate-1 and SPC nodes in fast list decoding are decoded, the maximum number of candidate paths for these two special types of nodes is  $2^\tau L$ , where  $\tau$  is the number of bits to be split in the nodes. For this reason, an effective  $2^\tau L$ -to- $L$  sorter is needed. In hardware implementation, the propagation delay of a sorter is directly affected by the number of sorter inputs. Therefore, we propose methods to efficiently reduce the number of sorter inputs.

Fig. 2 shows the PMs of the candidate paths in Rate-1 and SPC nodes with  $L = 4$ .  $PM_{-1}^l$  is the PM of the  $l$ -th path in the previous decoding step, and  $BM_t^l$  is the branch metric (BM) of each node. BM is the penalty added to the low-reliability path in (10) and (12). The PMs of the candidate paths (CPs) are determined as follows:

$$CP_t^l = PM_{-1}^l + BM_t^l \quad (13)$$

Suppose that the BMs are sorted in ascending order. The  $l$ -th column can be regarded as the CPs derived from the  $l$ -th path. In one column, the maximum number of candidate paths to be selected is  $L$ . Among the candidate paths in the first column,  $CP_{(1,4)}^1 = \{PM_{-1}^{1,t-1} + BM_t^1, \dots, PM_{-1}^{1,t-1} + BM_4^1\}$  can be selected, and it works the same for the other columns. Therefore, if the BMs from each path are sorted, Rate-1 and SPC decoders for fast list decoding would suffer no loss in error correction performance even when considering only  $L^2$  candidate paths. However, in practice, the BMs from a path are not sorted in ascending order. Taking this into account, we propose a PSN that considers the additional sorting of the CPs from each path. The proposed sorting network is shown in Fig. 3. The CPs branched out from each path are arranged by Sorter I. The  $m_i^l$  are the PMs pre-sorted by Sorter I, which are used as the input to Sorter II.

**B. CANDIDATE PATH EXCLUSION FOR SORTER I**

In Sorter I,  $2^\tau$  CPs are sorted to output  $L$  candidate paths. However, some PMs among these  $2^\tau$  candidates are not

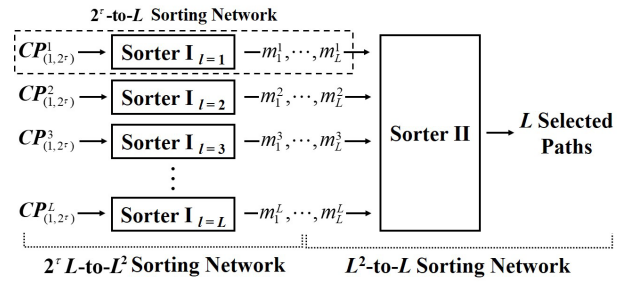


FIGURE 3. The proposed partitioned sorting network.

TABLE 1. The criteria for excluding candidate paths for each list size.

$L$	$CP(X)$	$NUM_{small}$	CP smaller than $CP(X)$
2	$CP(0)$	0	-
	$CP(1)$	1	$CP(0)$
4	$CP(2)$	2	$CP(0), CP(1)$
	$CP(3)$	3	$CP(0), CP(1), CP(2)$
	$CP(1,2)$	3	$CP(0), CP(1), CP(2)$
8	$CP(4)$	4	$CP(0), CP(1), CP(2), CP(3)$
	$CP(5)$	5	$CP(0), CP(1), CP(2), CP(3), CP(4)$
	$CP(1,3)$	5	$CP(0), CP(1), CP(2), CP(3), CP(1,2)$
	$CP(6)$	6	$CP(0), CP(1), CP(2), CP(3), CP(4), CP(5)$
	$CP(2,3)$	6	$CP(0), CP(1), CP(2), CP(3), CP(1,2), CP(1,3)$
	$CP(7)$	7	$CP(0), CP(1), CP(2), CP(3), CP(4), CP(5), CP(6)$
	$CP(1,4)$	7	$CP(0), CP(1), CP(2), CP(3), CP(4), CP(1,2), CP(1,3)$
$CP(1,2,3)$	7	$CP(0), CP(1), CP(2), CP(3), CP(1,2), CP(1,3), CP(2,3)$	

selected. Therefore, PMs that are not chosen should be excluded from the candidate paths. In [20], a minimum-combinations set is proposed for reducing the number of time steps in Rate-1 nodes. Based on [20], only candidate paths essential to the sorter input can be selectively extracted based on the order of the PMs that are branched out from one path.

$\alpha^*$  denote the node LLRs  $\alpha^l$  sorted in ascending order of absolute value ( $|\alpha_i^*| \leq |\alpha_{i+1}^*|, i = 1, 2, \dots, N_v - 1$ ).  $\beta^*$  denotes the bit estimates sorted in the same order as  $\alpha^*$ . In fast list decoding, using (10) and (12), the PMs of the candidate paths in Rate-1 and SPC nodes can be expressed as

$$CP(X) = PM_{-1} + \sum_{i \in X} p_i, \quad (14)$$

where  $X \subset \{1, 2, \dots, \tau\}$ .  $X$  is the set of bits which do not satisfy  $1 - 2\beta_i^* = \text{sgn}(\alpha_i^*)$ . The number of domains of  $CP$  is determined depending on the case which  $p_i$  is added. In a Rate-1 node,  $PM_{-1} = PM_{-1}^l$  and  $p_i = |\alpha_i^*|$ . In an SPC node, when  $\gamma = 1$ ,  $PM_{-1} = PM_{-1}^l + |\alpha_{\min}^*|$  and  $p_i = |\alpha_i^*| - |\alpha_{\min}^*|$ , and when  $\gamma = 0$ ,  $PM_{-1} = PM_{-1}^l$  and  $p_i = |\alpha_i^*| + |\alpha_{\min}^*|$ . In addition,  $CP(0) = PM_{-1}$ . According to the nature of the FSSCL-SPC decoding, the following properties are obtained.

- i) Since  $p_{i+1} \geq p_i$ ,  $CP(i + 1) \geq CP(i)$  is satisfied.
- ii) There is no apparent order between  $CP(i, j)$  and  $CP(k)$  ( $i < j < k$ ).
- iii) Among the PMs of all candidate paths,  $CP(0)$ ,  $CP(1)$ , and  $CP(2)$  are the smallest three PMs and satisfy  $CP(0) < CP(1) < CP(2)$ .

iv) For  $CP(i)$  to be selected as one of the  $L$  paths,  $CP(i - 1), \dots, CP(0)$ , which are smaller than  $CP(i)$ , must also be selected.

The number of PMs smaller than  $CP(X)$  is represented by  $NUM_{small}(CP(X))$ . If  $NUM_{small}(CP(X))$  is greater than or equal to  $L$ , then  $CP(X)$  cannot be selected as one of the  $L$  paths, and can be excluded from the input of Sorter I. In addition,  $CP(0)$ ,  $CP(1)$ , and  $CP(2)$  can be excluded from the input of Sorter I by iii). Therefore, Sorter I outputs  $L$  sorted paths. Table 1 shows the criteria for excluding candidate paths. As shown in Table 1, since  $NUM_{small}(CP(4)) = 4$ ,  $CP(4)$  is not considered as a candidate path when  $L = 4$ .

### C. CANDIDATE PATH SELECTION FOR SORTER II

Since PMs are pre-arranged by Sorter I, there are  $L^2$  candidate paths for inputs of Sorter II. For a more efficient sorting network, the number of inputs for Sorter II should be reduced. Therefore, we analyze the selection ratios for candidate paths. Figs. 4 and 5 show the cumulative sums of the  $L^2 = 16$  candidate path selections of Sorter II when decoding Rate-1 and SPC nodes of fast list decoding with  $PC(1024, 512)$  and  $L = 4$ .

In Figs. 4 and 5, specific paths account for most of the path selections, and there is a clear distinction between frequently chosen and infrequently chosen paths. In addition, although the selection frequency varies somewhat with the signal-to-noise ratio (SNR), the SNR does not significantly affect the ranking of the paths in terms of path selection frequency. Therefore, through efficient path metric sorting, the Rate-1 and SPC decoders have error correction performance close to that of SCL decoder without considering all  $L^2$  paths in Sorter II.

The existing list-fast-SSC decoding methods [14], [15] use a fixed  $M$ , where  $M$  is the number of candidate paths. In contrast, we set a different  $M$  value of Sorter II for each path based on our analysis of candidate paths. Accordingly, the proposed method requires setting parameters  $M^l$ , which denote the number of candidates  $m_{(1, L)}^l$  for each path. Let  $P$  represent the total number of candidate paths  $m_{(1, L)}^{(1, L)}$ , as shown in (15).

$$P = \sum_{l=1}^L M^l \tag{15}$$

To reduce the number of inputs of Sorter II, we can prune candidate paths that are selected less frequently than significant candidate paths. With this method,  $P$  can be reduced while guaranteeing performance and represented in terms of  $M^l$ . For example, in Fig. 4,  $M^l$  can be determined based on the selection frequency in descending order for  $L = 4$ , as

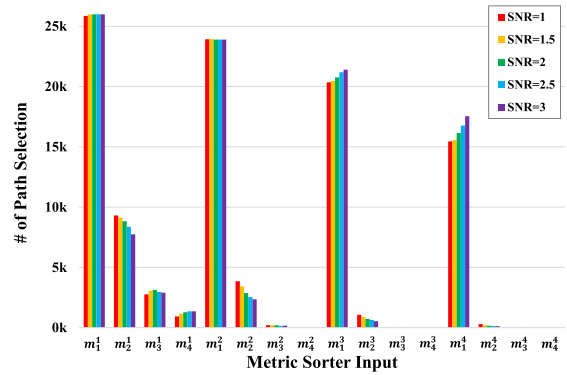


FIGURE 4. Candidate path selection of Sorter II when decoding SPC nodes with  $PC(1024, 512)$  and  $L = 4$  (1000 decoding trials).

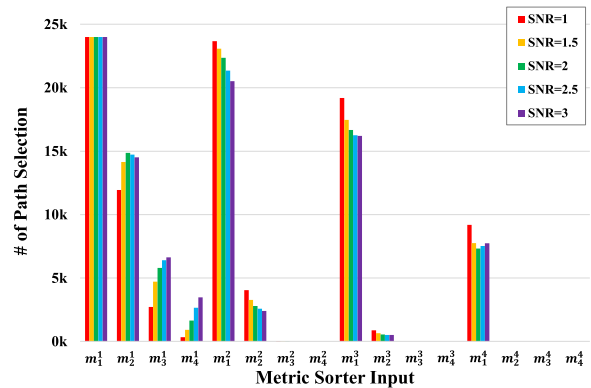


FIGURE 5. Candidate path selection of Sorter II when decoding Rate- 1 nodes with  $PC(1024, 512)$  and  $L = 4$  (2000 decoding trials).

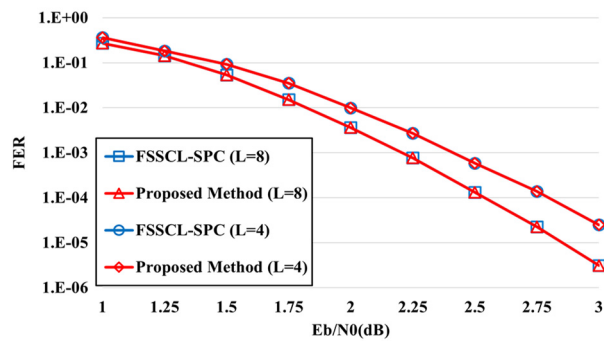


FIGURE 6. FER comparisons between the fast list decoding with proposed candidate path exclusion of Sorter I and the existing FSSCL-SPC for  $PC(1024,512)$ .

shown in (16).

$$M_{Proposed, L=4}^{(1, L)} = \{4, 2, 1, 1\}, \tag{16}$$

$$P_{Proposed} = L^2 / 2.$$

### D. ERROR CORRECTION PERFORMANCE ANALYSIS

Fig. 6 shows the FER comparisons among the fast list decoding with the candidate path exclusion for Sorter I and the

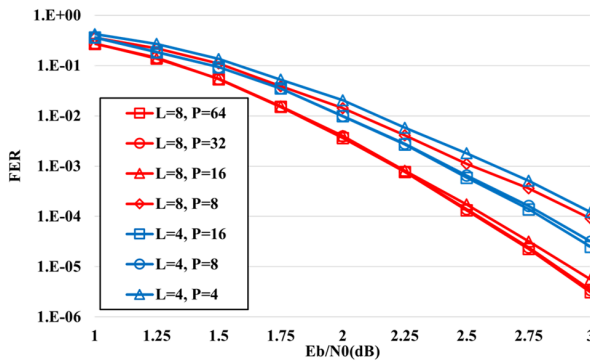


FIGURE 7. FER comparisons for fast list decoding with different numbers of candidates in Sorter II for PC(1024, 512).

TABLE 2. The numbers of candidates per path for sorter ii in Fig. 7.

$L$	$P$	$M^1$	$M^2$	$M^3$	$M^4$	$M^5$	$M^6$	$M^7$	$M^8$
4	16	4	4	4	4	-	-	-	-
	8	4	2	1	1	-	-	-	-
8	4	2	1	1	0	-	-	-	-
	64	8	8	8	8	8	8	8	8
	32	8	8	4	4	2	2	2	2
	16	5	3	2	2	1	1	1	1
	8	2	1	1	1	1	1	1	0

conventional FSSCL-SPC. First, the simulations were performed using binary phase-shift keying (BPSK) modulation and an additive white Gaussian noise (AWGN) channel. Then, the cyclic redundancy check (CRC-16) code was used. Since the fast list decoding with the candidate path exclusion for Sorter I selects the same  $L$  paths compared with the existing FSSCL-SPC, there is no error correction performance degradation.

Fig. 7 compares the FER performances achieved with different values of  $M^l$  and  $P$  in Sorter II. In addition, Table 2 shows the  $M^l$  values for various  $L$  and  $P$  values in Fig. 7. Since  $m_{(1,L)}^{(1,L)}$  are pre-sorted by Sorter I, the value of  $P$  in Sorter II is  $L^2$ . Therefore, the reduction ratio of the number of candidates can be expressed as  $1 - P/L^2$ . As shown in Fig. 7, the FER performance degrades as the reduction ratio increases. However, the FER performance with  $P = L^2/2$  is still similar to that of the conventional FSSCL-SPC. Therefore, in this paper, we set a reduction ratio of 50% as the threshold that guarantees the error correction performance. Accordingly, the proposed method can reduce the number of candidate paths by 50%, and the corresponding settings can be generalized as expressed in (17).

$$\begin{aligned}
 M^{(1,L/4)} &= L, \\
 M^{(L/4+1,L/2)} &= L/2, \\
 M^{(L/2+1,L)} &= L/4.
 \end{aligned} \tag{17}$$

Fig. 8 shows the results of FER comparisons among the SCL algorithm [6], the conventional FSSCL-SPC algorithm [19], and fast list decoding with the proposed PSN. Fast list decoding with the proposed PSN can achieve a

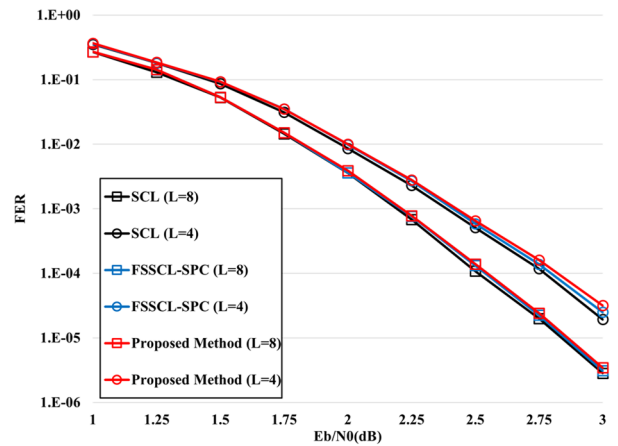


FIGURE 8. FER comparisons of the SCL, FSSCL-SPC, and proposed methods for PC(1024, 512).

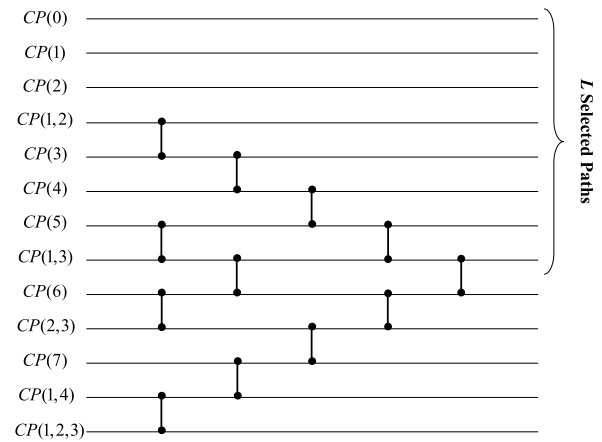


FIGURE 9. The proposed Sorter I for  $L = 8$  and  $N_v = 8$ .

similar FER performance as the SCL [6] and conventional FSSCL-SPC [19] algorithms. The FER performance of fast list decoding using the proposed method is degraded by less than 0.1 dB, with an FER of  $10^{-4}$ , for  $L = 4$ . Similarly, the FER is degraded by less than 0.05 dB, with an FER of  $10^{-4}$ , for  $L = 8$ .

### E. PROPOSED SORTING NETWORK

We propose a PSN to reduce the sorting complexity of the Rate-1 and SPC decoders for fast list decoding. In PSN, there are two sorters, Sorter I and Sorter II. Fig. 9 shows the proposed Sorter I for  $L = 8$ ,  $N_v \geq 8$ . Each horizontal line represents a PM with 8-bit quantization. Each vertical line represents a CASU. If the upper input of a CASU is larger than the lower input, the CASU swaps the two positions.

To design a more efficient Sorter I, the number of CASUs should be minimized. By the properties of section III-B, the inputs of Sorter I have CPs with a pre-determined order and CPs without a pre-determined order. CASU is only required for CPs whose order is not pre-determined. In addition,  $CP(0)$ ,  $CP(1)$ , and  $CP(2)$  are always the three smallest

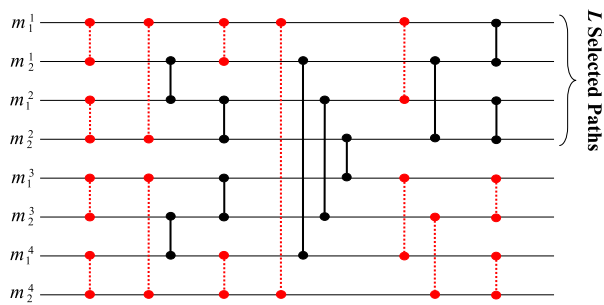


FIGURE 10. GBS [21] and PBS [9] for an SCL decoder when  $L = 4$ .

CPs and there is no need to compare them. Based on this, we designed the Sorter I with a minimized number of CASUs. Fig. 9 shows the proposed Sorter I for the PSN.

The proposed metric Sorter II is based on a bitonic sorter. Fig. 10 shows the general bitonic sorter (GBS) [21] and the PBS [9] for a conventional SCL decoder when  $L = 4$ . The red-dotted CASUs can be removed based on the properties of the PMs in the SCL decoder as expressed in (18) and (19). Therefore, these properties enable  $L$ -reliable path selection without degradation in error correction performance [9]. We applied the PBS's method of removing CASU for conventional SCL to Sorter II for Rate-1 and SPC decoders for fast list decoding.

$$m_i^l \leq m_{i+1}^{l+1} \tag{18}$$

$$m_i^l \leq m_{i+1}^l \tag{19}$$

According to Section III-C, the proposed Sorter II considers only  $L^2/2$  paths, representing the majority of the results of path selection. The proposed Sorter II can be generalized as follows: Paths  $m^{(1,L/4)}$  branched out from the first 1/4 paths satisfy  $M^{(1,L/4)} = L$ , paths  $m^{(L/4+1,L/2)}$  branched out from the next 1/4 paths satisfy  $M^{(L/4+1,L/2)} = L/2$ , and paths  $m^{(L/2+1,L)}$  branched out from the last 1/2 paths satisfy  $M^{(L/2+1,L)} = L/4$ . As a result, the  $L^2$  candidate paths are reduced to  $L^2/2$ . Therefore, we call this sorter the half-input bitonic sorter (HIBS). The input paths are simplified to  $m_{(1,4)}^1$ ,  $m_{(1,2)}^2$ , and  $m_1^{(3,4)}$  by (17) by the proposed path selection method. As a result, the  $L^2 = 16$  candidate paths are reduced to  $L^2/2=8$ .

Fig. 11 shows the proposed HIBS for Sorter II when  $L = 4$ . In the proposed HIBS, the input PMs  $m_{(1,L)}^{(1,L)}$  of Sorter II are pre-sorted by Sorter I. Therefore, (19) is satisfied. For this reason, comparing between  $m_{(1,L)}^l$  is unnecessary. However, in the PM initialization equations for SPC nodes (9), the parity  $\gamma^l$  has different values for each path. Therefore, the PM property expressed in (18) is not satisfied in an SPC decoder. For this reason, in Fig. 11, comparing between  $m_1^3$  and  $m_1^4$  is needed.

F. COMPLEXITY ANALYSIS

The complexity of a sorter is expressed in terms of the numbers of CASUs and CASU stages. For pair comparison

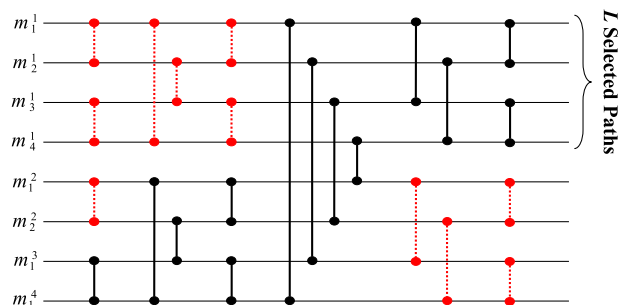


FIGURE 11. The proposed HIBS for Sorter II when  $L = 4$ .

with the proposed PSN, we adopted GBS [21] and PBS [9] for an SCL decoder to fast list decoding. Furthermore, for comparison of various cases, three sorters were used for Sorter II of the PSN: GBS, PBS, and HIBS.

Sorting networks for fast list decoding are designed based on GBS [21]. In addition, the complexities of the sorting networks were generalized using the complexity of GBS for the SCL. The complexity of GBS for the SCL algorithm is described in (20), where  $S_{SCL}^{GBS}(L)$  is the number of CASU stages in a  $2L$ -input GBS and  $C_{SCL}^{GBS}(L)$  is the number of CASUs in a  $2L$ -input GBS. Accordingly, the complexity in terms of  $L$  can be calculated, as shown in (20).

$$S_{SCL}^{GBS}(L) = \frac{1}{2}(\log(L) + 1)(\log(L) + 2),$$

$$C_{SCL}^{GBS}(L) = \frac{L}{2}(\log(L) + 1)(\log(L) + 2). \tag{20}$$

However, the number of inputs to the sorter for fast list decoding is determined not only by  $L$  but also by  $\tau$ . The complexity of a GBS for the FSSCL-SPC can be generalized as expressed in (21).

$$S_{fast\ list}^{GBS}(L, \tau) = \frac{1}{2}(\log(\frac{2^\tau L}{2}) + 1)(\log(\frac{2^\tau L}{2}) + 2),$$

$$C_{fast\ list}^{GBS}(L, \tau) = \frac{1}{2} \frac{2^\tau L}{2} (\log(\frac{2^\tau L}{2}) + 1)(\log(\frac{2^\tau L}{2}) + 2). \tag{21}$$

Based on (21), the complexity of PBS for the conventional FSSCL-SPC can be calculated. Thus, the complexity of the  $2^\tau L$ -input PBS for fast list decoding can be expressed as shown in (22).

$$S_{fast\ list}^{PBS}(L, \tau) = S_{fast\ list}^{GBS}(L, \tau),$$

$$C_{fast\ list}^{PBS}(L, \tau) = C_{fast\ list}^{GBS}(L, \tau) - L S_{fast\ list}^{GBS}(1, \tau) - \sum_{i=1}^{\log(2^\tau L)} \frac{2^\tau L}{2^{i+1}} \left[ \frac{2^\tau L - L}{2^{\log(2^\tau L) - i}} \right]. \tag{22}$$

The modified bitonic sorting network for fast list decoding is shown in Fig. 12.

The proposed PSN is consisting of two sorters: Sorter I and Sorter II. Table 3 shows the numbers of CASUs and CASU stages for Sorter I designed based on section III-B. Additionally, in the PSN, there are  $L$ -number of Sorter I.

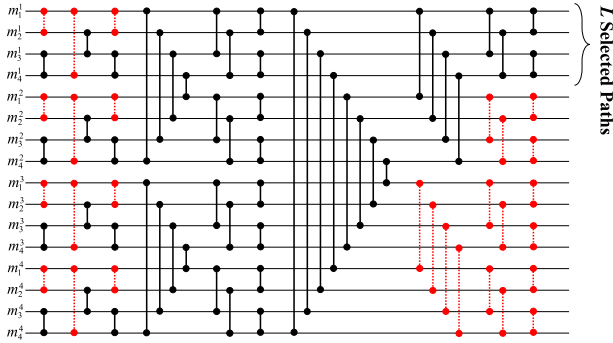


FIGURE 12. GBS and PBS for a Fast list decoder when  $L = 4$ .

TABLE 3. The numbers of CASUs and CASU stages for sorter I.

$L$	Rate-1 Node ( $N_v=4$ )		SPC Node ( $N_v=4$ )		Rate-1 Node ( $N_v=8$ )		SPC Node ( $N_v=8$ )	
	Stages	CASUs	Stages	CASUs	Stages	CASUs	Stages	CASUs
2	0	0	0	0	0	0	0	0
4	1	1	1	1	1	1	1	1
8	3	6	1	1	5	12	5	12

In Sorter II, since equations (21), (22) are for the  $2^r L$ -input, new equations for complexities of GBS, PBS, and HIBS are needed. These complexities are calculated by using (21). The complexity of GBS and PBS for Sorter II is computed as shown in (23) and (24).

$$\begin{aligned}
 S_{Sorter II}^{GBS}(L) &= S_{fast list}^{GBS}(L^2, 0), \\
 C_{Sorter II}^{GBS}(L) &= C_{fast list}^{GBS}(L^2, 0).
 \end{aligned}
 \tag{23}$$

$$\begin{aligned}
 S_{Sorter II}^{PBS}(L) &= S_{fast list}^{GBS}(L^2, 0) - S_{fast list}^{GBS}(L, 0), \\
 C_{Sorter II}^{PBS}(L) &= C_{fast list}^{GBS}(L^2, 0) - L C_{fast list}^{GBS}(\sqrt{L}, 0) \\
 &\quad - \sum_{i=1}^{\log(L^2)} \frac{L^2}{2^{i+1}} \left[ \frac{L^2 - L}{2^{\log(L^2)-i}} \right].
 \end{aligned}
 \tag{24}$$

Fig. 13 shows the GBS and PBS for Sorter II when  $L = 4$ . Likewise, the complexity of the proposed HIBS for Sorter II depicted in Fig. 11 is calculated as shown in (25).

$$\begin{aligned}
 S_{Sorter II}^{HIBS}(L) &= S_{fast list}^{GBS}(L^2/2, 0) - \max(0, S_{fast list}^{GBS}(L/4, 0)), \\
 C_{Sorter II}^{HIBS}(L) &= C_{fast list}^{GBS}(L^2/2, 0) - L/4 \times C_{fast list}^{GBS}(L, 0) \\
 &\quad - L/4 \times C_{fast list}^{GBS}(L/2, 0) - L/2 \\
 &\quad \times C_{fast list}^{GBS}(L/4, 0) \\
 &\quad - \sum_{i=1}^{\log(L^2/2)} \frac{L^2}{2^{i+2}} \left[ \frac{L^2/2 - L}{2^{\log(L^2/2)-i}} \right].
 \end{aligned}
 \tag{25}$$

Table 4 compares the numbers of CASUs and CASU stages between the GBS, the PBS, and the proposed PSNs when  $N_v = 4$ . Although the node type affects the number of CASUs, the comparison was performed targeting the Rate-1 node that uses the maximum CASU. The sorting networks are designed

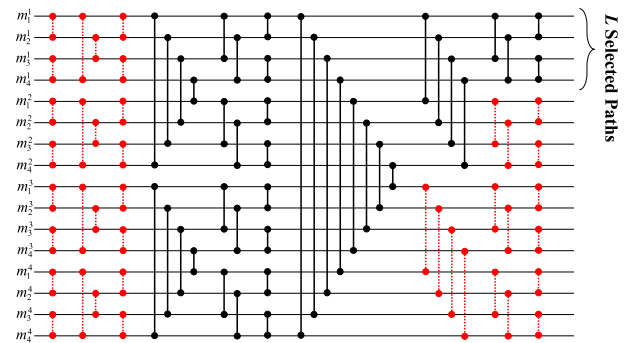


FIGURE 13. GBS, PBS for Sorter II when  $L = 4$ .

TABLE 4. Comparison of the numbers of CASUs and CASU stages.

$L$	GBS <sub>fast-list</sub>		PBS <sub>fast-list</sub>		Proposed PSN					
					GBS*		PBS*		HIBS*	
	Stages	CASUs	Stages	CASUs	Stages	CASUs	Stages	CASUs	Stages	CASUs
2	3	6	2	3	3	6	2	3	1	1
4	15	240	15	168	11	84	14	37	7	17
8	28	1792	28	1396	24	768	18	404	17	180

\* Type of network used in Sorter II

TABLE 5. Comparison of implementation results.

$L$	GBS <sub>fast-list</sub>		PBS <sub>fast-list</sub>		Proposed PSN					
					GBS*		PBS*		HIBS*	
	Freq. [MHz]	EGC	Freq. [MHz]	EGC	Freq. [MHz]	EGC	Freq. [MHz]	EGC	Freq. [MHz]	EGC
2	518	752	877	438	518	752	877	438	1351	259
4	110	17420	121	12973	188	5771	230	3493	261	1711
8	50	112514	62	87450	75	38409	101	26985	121	12344

\* Type of network used in Sorter II

for Rate-1 and SPC decoders for fast list decoding. The SBS [9], PMS [11], and PBE [12], [13] can be used for PM operation of SCL satisfying (18) and (19). However, PMs of fast list decoding do not satisfy (18) and (19). Therefore, these sorting networks [9], [11], [12], [13] are not suitable for fast list decoding and are not included in this comparison.

In the proposed sorting network design PSN<sub>HIBS</sub>, the number of CASU stages for  $L = 8$  is reduced by approximately 40% compared to GBS<sub>fast-list</sub> [21] and PBS<sub>fast-list</sub> [9]. In addition, the number of CASUs for  $L = 8$  is reduced by approximately 90% compared to GBS<sub>fast-list</sub> and by approximately 87% compared to PBS<sub>fast-list</sub>. Even in the cases of  $L = 4$  and 8, the proposed sorting network shows competitive results.

#### IV. IMPLEMENTATION RESULTS

Table 5 shows the implementation results regarding the equivalent gate count (EGC) and the maximum frequency for the sorting networks in Table 4. The sorting networks were synthesized using a standard cell library for Samsung 28 nm



technology, and the EGC was calculated based on 2-input NAND gates. The results in Table 5 may not be equivalent to those in Table 4 due to various optimization methods for ASIC implementation.

Table 5 shows that the proposed sorting networks have higher maximum frequencies and fewer EGCs than the others. For  $L = 4$ , the proposed design PSN<sub>HIBS</sub> can operate at approximately 137% and 115% higher frequencies than GBS<sub>fast-list</sub> and PBS<sub>fast-list</sub>, respectively. Furthermore, for  $L = 4$ , PSN<sub>HIBS</sub> requires about 90% and 86% fewer EGCs than GBS<sub>fast-list</sub> and PBS<sub>fast-list</sub>, respectively. In the proposed PSNs, PSN<sub>HIBS</sub> has 61% and 20% higher frequencies than PSN<sub>GBS</sub> and PSN<sub>PBS</sub>, respectively, for  $L = 8$ . PSN<sub>HIBS</sub> requires 68% and 54% fewer EGCs than PSN<sub>GBS</sub> and PSN<sub>PBS</sub>, respectively, for  $L = 8$ . The designs were not evaluated for  $L = 16$  since they are impractical to implement in these cases due to the exponential increase in the number of EGCs.

## V. CONCLUSION

This paper proposes the PSN, which is efficient for Rate-1 and SPC decoders. First, we analyzed the order of PMs. In fast list decoding, some paths are not selected for Rate-1, SPC nodes. Through this, we proposed candidate paths exclusion method for each list size. Second, the selection ratios for candidate paths were analyzed. It was shown that specific paths among all  $L^2$  possible candidates account for the majority of the selected paths. In addition, we compared the setting parameters  $M^l$  that minimized the FER performance degradation and determined that the number of inputs to the sorting network could be reduced by 50%.

Finally, we proposed low-complexity sorting networks of Rate-1 and SPC decoders for fast list decoding. The proposed sorting networks can guarantee error correction performance comparable to that of the SCL decoding algorithm. Furthermore, complexity comparisons based on the number of CASUs and ASIC synthesis results showed that the proposed sorting network design reduces the number of CASUs by down to 14% and permits up to 115% higher operating frequencies compared with sorting networks of fast list decoder. Therefore, the proposed method can be effectively applied for the hardware implementation of fast list decoders with  $L \leq 8$ .

## REFERENCES

- [1] E. Arkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.
- [2] 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, Jan. 2013.
- [3] B. Yuan and K. K. Parhi, "Low-latency successive-cancellation polar decoder architectures using 2-bit decoding," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 4, pp. 1241–1254, Apr. 2014.
- [4] A. Alamdar-Yazdi and F. R. Kschischang, "A simplified successive-cancellation decoder for polar codes," *IEEE Commun. Lett.*, vol. 15, no. 12, pp. 1378–1380, Dec. 2011.
- [5] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Fast polar decoders: Algorithm and implementation," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, pp. 946–957, May 2014.

- [6] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.
- [7] 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.*, Nov. 2014, pp. 2116–2120.
- [8] A. Balatsoukas-Stimming, M. B. Parizi, and A. Burg, "LLR-based successive cancellation list decoding of polar codes," *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5165–5179, Oct. 2015.
- [9] A. Balatsoukas-Stimming, M. B. Parizi, and A. Burg, "On metric sorting for successive cancellation list decoding of polar codes," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Lisbon, Portugal, May 2015, pp. 1993–1996.
- [10] B. Y. Kong, H. Yoo, and I. C. Park, "Efficient sorting architecture for successive-cancellation-list decoding of polar codes," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 63, no. 7, pp. 673–677, Jul. 2016.
- [11] H. Li, "Enhanced metric sorting for successive cancellation list decoding of polar codes," *IEEE Commun. Lett.*, vol. 22, no. 4, pp. 664–667, Apr. 2018.
- [12] V. Bioglio, F. Gabry, L. Godard, and I. Land, "Two-step metric sorting for parallel successive cancellation list decoding of polar codes," *IEEE Commun. Lett.*, vol. 21, no. 3, pp. 456–459, May 2017.
- [13] X. Wang, T. Wang, J. Li, L. Shan, H. Cao, and Z. Li, "Improved metric sorting for successive cancellation list decoding of polar codes," *IEEE Commun. Lett.*, vol. 23, no. 7, pp. 1123–1126, Jul. 2019.
- [14] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Fast list decoders for polar codes," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 318–328, Feb. 2016.
- [15] X. Zhang, R. Sui, J. Cui, D. Zhang, and Q. Zeng, "Efficient sorting architecture for list-fast-SSC decoding of polar codes," *IEEE Access*, vol. 6, pp. 61773–61781, 2018.
- [16] S. Ali Hashemi, C. Condo, and W. J. Gross, "Simplified successive-cancellation list decoding of polar codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 815–819.
- [17] S. A. Hashemi, C. Condo, and W. J. Gross, "A fast polar code list decoder architecture based on sphere decoding," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 12, pp. 2368–2380, Dec. 2016.
- [18] S. A. Hashemi, C. Condo, and W. J. Gross, "Fast simplified successive-cancellation list decoding of polar codes," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, San Francisco, CA, USA, Mar. 2017, pp. 1–6.
- [19] S. A. Hashemi, C. Condo, and W. J. Gross, "Fast and flexible successive-cancellation list decoders for polar codes," *IEEE Trans. Signal Process.*, vol. 65, no. 21, pp. 5756–5769, Nov. 2017.
- [20] Y. Zhao, Z. Yin, Z. Wu, and M. Xu, "Minimum-combinations set-based rate-1 decoder for fast list decoding of polar codes," *IEEE Commun. Lett.*, vol. 25, no. 10, pp. 3185–3189, Oct. 2021.
- [21] J. Lin and Z. Yan, "An efficient list decoder architecture for polar codes," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 11, pp. 2508–2518, Nov. 2015.
- [22] C. Kestel, L. Johannsen, O. Griebel, J. Jimenez, T. Vogt, T. Lehnigk-Emden, and N. Wehn, "A 506 Gbit/s polar successive cancellation list decoder with CRC," in *Proc. IEEE 31st Annu. Int. Symp. Pers., Indoor Mobile Radio Commun.*, Aug. 2020, pp. 1–7.



**YONGJE LEE** (Graduate Student Member, IEEE) received the B.S. degree in electrical and electronics engineering from Ajou University, Suwon, South Korea, in 2021, where he is currently pursuing the M.S. degree in electrical and electronic engineering. His current research interests include signal processing, channel coding, deep learning, and SoC design.



**JAE HONG ROH** received the B.S. and M.S. degrees in electrical and electronics engineering from Ajou University, Suwon, South Korea, in 2020 and 2022, respectively. He is currently working with LX Semicon, Seoul, South Korea. His current research interests include signal processing, channel coding, cryptography, and SoC design.



**USEOK LEE** (Member, IEEE) received the B.S. degree in electrical and electronics engineering from Ajou University, Suwon, South Korea, in 2019, where he is currently pursuing the combined M.S. and Ph.D. degrees in electrical and electronic engineering. His current research interests include signal processing, channel coding, cryptography, and SoC design.



**MYUNG HOON SUNWOO** (Fellow, IEEE) received the B.S. degree from Sogang University, in 1980, the M.S. degree in EE from KAIST, in 1982, and the Ph.D. degree in ECE from the University of Texas at Austin, Austin, TX, USA, in 1990. He worked at ETRI, Daejeon, South Korea, from 1982 to 1985; and at Digital Signal Processor Operations, Motorola, from 1990 to 1992. Since 1992, he has been with the School of Electrical and Computer Engineer-

ing, Ajou University, Suwon, South Korea, where he is currently a Professor. He has authored over 470 articles, holds more than 120 patents, and has won more than 55 awards. His current research interests include artificial intelligence circuits and systems, low-power algorithms and architectures, medical imaging diagnosis, and deep-learning-based channel coding. He is the Director of the Medical Image-Based Intelligent Diagnostic Solutions (MIIDS) Research Center. He has been a Distinguished Lecturer at IEEE Circuits and Systems Society (CASS) (2009–2010). He has served on the CASS Board of Directors (BoG) (2011–2016) and the IEEE CASS VP of Conferences (2018–2021). As the IEEE CASS VP of Conferences, he launched the IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), in 2019. Recently, he was elected IEEE CASS President-Election (2022–2023). He served as the General Chair of the International Symposium on Circuits and Systems (ISCAS) 2012 in Seoul, South Korea, and as the General Co-Chair of ISCAS 2021 in Daegu, South Korea. He has been an Associate Editor of IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS (2002–2003) and a Co-Editor of several books, including *Selected Topics in Biomedical Circuits and Systems* (River Publishers Series in Circuits and Systems, 2021).

...