

# An Efficient Adaptive Belief Propagation Decoder for Polar Codes

Zhongjun Yang<sup>†</sup>, Zuoxin Cai<sup>†</sup>, Li Chen<sup>†‡</sup>, and Huazi Zhang<sup>§</sup>

<sup>†</sup> School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China

<sup>‡</sup> Guangdong Province Key Laboratory of Information Security Technology, Guangzhou, China

<sup>§</sup> Hangzhou Research Center, Huawei Technologies Co. Ltd., Hangzhou, China

Email: yangzhj59@mail2.sysu.edu.cn, caizx7@mail2.sysu.edu.cn, chenli55@mail.sysu.edu.cn, zhanghuazi@huawei.com

**Abstract**—Due to the high parallelism of belief propagation (BP) decoding, it is considered as a promising solution for the decoding latency challenge of long polar codes. However, the error-correction performance of the classical BP decoding is inferior to that of the successive cancellation (SC) and the SC list (SCL) decoding. In this paper, an adaptive BP (ABP) decoding algorithm is proposed to bridge this performance discrepancy. It iteratively adjusts the *a priori* log-likelihood ratios (LLRs) of error-prone bits, which can be efficiently detected using the frozen and information processing elements (FIPEs). Moreover, a novel low-complexity FIPE-based early termination criterion (ETC) is proposed to further reduce the decoding complexity. It functions when all the frozen bits in the FIPEs are successfully decoded with stable LLR magnitudes. Our numerical results show that for the (1024, 512) polar code, the ABP decoding outperforms the classical BP decoding by 0.3 dB at the frame error rate (FER) of  $10^{-4}$  over the additive white Gaussian noise (AWGN) channel. It can also achieve up to 78.5% latency reduction over the fast simplified SC (FSSC) decoding, while maintaining the same performance. The proposed ETC also exhibits a lower hardware complexity over the existing G-matrix criterion.

**Index Terms**—Adjusted min-sum, belief propagation, early termination criterion, hardware, low-complexity, polar codes.

## I. INTRODUCTION

POLAR codes, invented by Arikan [1], can achieve channel capacity for binary-input discrete memoryless channels (B-DMCs), using the efficient successive cancellation (SC) decoding. However, for short-to-moderate codeword lengths, performance of the SC decoding is less favorable. To improve the decoding performance, the SC list (SCL) decoding [2] [3] and the cyclic redundancy check (CRC) aided SCL (CA-SCL) decoding [4] were proposed. To reduce decoding latency, the simplified SC (SSC) decoding [5] and the fast SSC (FSSC) decoding [6] were proposed. However, due to their sequential decoding feature, the above SC-based decoding algorithms still result in an unaffordable decoding latency for long polar codes.

As an alternative, belief propagation (BP) decoding [7] can be implemented in a parallel manner, yielding its low latency and high throughput advantages. However, there exists a severe performance gap between the classical BP and the SC-based decoding algorithms. To solve this problem, several attempts [8]–[13] have been proposed. By adding perturbations to the *a priori* log-likelihood ratios (LLRs) of the less reliable bits (LRBs), the post-processing methods [8] [9] can improve the BP decoding performance. Similar to [8], the BP correction

(BPC) decoding [10] corrects errors by resetting the *a priori* LLRs of the unreliable received symbols. However, the perturbations for the above methods are derived empirically. As a more radical method, the BP flip (BPF) decoding [11]–[13] sets the LLRs of the error bits to  $+\infty$  or  $-\infty$ . The BPF decoding can yield a similar performance as the SCL decoding, but with an unstable decoding complexity and latency. Another widely used solution is the BP list (BPL) decoding [14]–[19]. Its advantage lies in the low decoding latency but at the cost of a high decoding complexity.

Due to the slow convergence of BP decoding, a large number of iterations is needed. The early termination criterion (ETC) [20]–[23] can be effective in avoiding unnecessary iterations. In [20], the G-matrix criterion and the minimum LLR (minLLR) criterion were proposed. Compared with [20], the worst information bits (WIB) criterion [21] achieves the same termination performance with a lower complexity. Other efficient criteria include the frozen bit error rate (FBER) criterion [22] and the best frozen bits (BFB) criterion [23]. However, the FBER criterion requires division operations, which is computationally costly. For the BFB criterion, it is less effective at the low signal-to-noise ratio (SNR) region. To address above challenges, a more efficient ETC is needed for the BP decoding.

Recently, a special processing element (PE), the frozen and information PE (FIPE) [24], has been characterized for accurately identifying the erroneous bits. In this paper, we utilize the detection results of the FIPEs to construct the reliable index set (RIS). It contains the indices of the identified *reliable* information bits, which are considered being successfully estimated. Based on this, a novel adaptive BP (ABP) decoder for improving the decoding performance is proposed. In the ABP decoding, the *a priori* LLRs of LRBs within the RIS are iteratively adjusted, such that they can be correctly decoded. Our simulation results show that the proposed ABP decoder can achieve an improved performance over the classical BP decoder. Furthermore, it yields a similar performance as the SC decoder at the high SNR region. Moreover, by detecting the reliability and the LLR of the FIPEs, a new FIPE-based ETC is designed. It can further facilitate the ABP decoder to achieve a lower decoding complexity. Compared with the existing ETCs, such as the G-matrix criterion and the WIB criterion, the proposed criterion exhibits a lower hardware complexity.

## II. PRELIMINARIES

In this work, we use  $u_0^{N-1}$  to denote the vector  $(u_0, \dots, u_{N-1})$ . Let  $\hat{u}_i$  denote the estimation of bit  $u_i$ . Given an ordered set  $\mathcal{B}$ , its cardinality is  $|\mathcal{B}|$ , and  $\mathcal{B}[i]$  indicates the  $i$ -th element of  $\mathcal{B}$ , where  $0 \leq i < |\mathcal{B}|$ . The XOR operation is denoted as  $\oplus$ . The following provides the preliminary knowledge for polar codes and its BP decoding.

## A. Polar Codes

Polar codes are linear block codes characterized by its length  $N$  and dimension  $K$ , denoted as  $\mathcal{P}(N, K)$ . Based on the channel polarization, the indices of  $N$  polarized subchannels can be divided into two disjoint sets. The information set  $\mathcal{A}$  contains the indices of the  $K$  most reliable subchannels, which are used to transmit information bits. The remaining  $N - K$  indices form the frozen set  $\mathcal{A}_C$ , which are used to transmit the frozen bits (usually set as zero). These information bits and frozen bits constitute the input vector  $u_0^{N-1}$ . Codeword  $x_0^{N-1}$  of the  $\mathcal{P}(N, K)$  code is generated by  $x_0^{N-1} = u_0^{N-1} \mathbf{G}$ , where  $N = 2^n$  and  $\mathbf{G} = \mathbf{F}^{\otimes n}$  is the  $n$ -th Kronecker power of the kernel matrix  $\mathbf{F} = [(1, 0), (1, 1)]^T$ . Additionally, the rate of the  $\mathcal{P}(N, K)$  code is  $R = K/N$ .

## B. BP Decoding

The BP decoding can decode information bits parallelly using the given factor graph (FG) of a polar code. For the  $\mathcal{P}(N, K)$  code, the FG consists of  $(Nn)/2$  PEs, and each PE comprises four nodes. The FG and the PE architecture for the  $\mathcal{P}(8, 4)$  code are shown in Fig. 1 (a) and (b), respectively. Note that the  $j$ -th node in the  $i$ -th row of the FG is indexed by  $(i, j)$ , where  $0 \leq i < N$  and  $0 \leq j \leq n$ . Moreover, there are two types of messages in each node. Let  $L_{(i,j)}$  and  $R_{(i,j)}$  denote the left message and the right message of the node  $(i, j)$ , respectively. During BP decoding, messages are propagated and updated within the PEs, which satisfy

$$L_{(i,j)} = f(L_{(i,j+1)}, L_{(i+2^j,j+1)} + R_{(i+2^j,j)}), \quad (1a)$$

$$L_{(i+2^j,j)} = f(L_{(i,j+1)}, R_{(i,j)}) + L_{(i+2^j,j+1)}, \quad (1b)$$

$$R_{(i,j+1)} = f(L_{(i+2^j,j+1)} + R_{(i+2^j,j)}, R_{(i,j)}), \quad (1c)$$

$$R_{(i+2^j,j+1)} = f(L_{(i,j+1)}, R_{(i,j)}) + R_{(i+2^j,j)}, \quad (1d)$$

where  $f(a, b) = \alpha \text{sign}(a) \text{sign}(b) \min\{|a|, |b|\}$ , and  $\alpha$  is a scaling parameter that is optimized as 0.9375 [20] [25].

For a BP decoder,  $L_{(i,n)}$  represents the LLR of the  $i$ -th received symbol  $y_i$ . The  $R_{(i,0)}$  indicates the *a priori* LLR of bit  $u_i$ . If  $i \in \mathcal{A}$ ,  $R_{(i,0)} = 0$ ; otherwise,  $R_{(i,0)} = +\infty$ . Other messages are initialized to zero. Moreover, it is worth mentioning that  $L_{(i,0)}$  indicates the *extrinsic* LLR of bit  $u_i$ . During each iteration, the message propagation starts from right-to-left, updating the left messages. After that, it propagates from left-to-right, renewing the right messages. The BP decoding will terminate only it reaches the maximum number of iterations  $I_{\max}$  or an ETC [20]–[24] is satisfied. Finally, the decoder delivers the estimated bits  $\hat{u}_0^{N-1}$  based on

$$\hat{u}_i = \begin{cases} 0, & L_{(i,0)} + R_{(i,0)} \geq 0, \\ 1, & L_{(i,0)} + R_{(i,0)} < 0. \end{cases} \quad (2)$$

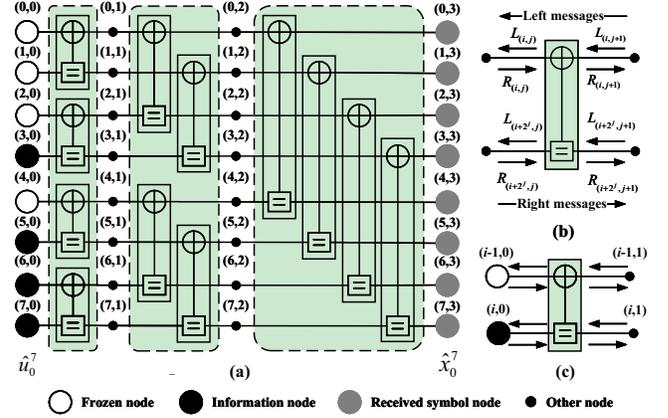


Fig. 1. Architecture of BP decoding for  $\mathcal{P}(8, 4)$  with  $\mathcal{A} = \{3, 5, 6, 7\}$ . (a) Factor graph; (b) General PE; (c) FIPE.

## C. The FIPEs

Fig. 1(c) illustrates the structure of the FIPEs. In a FIPE, there is a frozen node in the upper-left corner, and an information node in the bottom-left corner [24]. For all FIPEs, we use the ordered set  $\mathcal{B}$  to denote the index set of the corresponding information bits, where  $\mathcal{B}[i] < \mathcal{B}[j]$  if  $i < j$ . Thus, for any  $i \in \mathcal{B}$ , there exist  $R_{(i-1,0)} = +\infty$  and  $R_{(i,0)} = 0$ . For simplicity, let FIPE- $i$  denote a FIPE that consists of the frozen bit  $u_{i-1}$  and the information bit  $u_i$ .

## III. ADAPTIVE BP DECODING

In this section, the ABP decoding is proposed to enhance the error-correction performance. It adjusts the right message of the LRBs within the RIS iteratively. The detection results of the FIPEs will be utilized to identify the *reliable* information bits. They form the RIS. The RIS provides the indices of the LRBs to the ABP decoding.

## A. Reliable FIPEs

Let us consider the case that the frozen bit  $u_{i-1}$  in the FIPE- $i$  ( $i \in \mathcal{B}$ ) has been correctly decoded, i.e.,  $L_{(i-1,0)} \geq 0$ . In [24], such FIPE is classified as the *reliable* FIPE. The following Propositions 1 and 2 reveal the sufficient condition and the characteristics of the *reliable* FIPE, respectively.

**Proposition 1.** Given a FIPE- $i$ , if  $L_{(i-1,1)}L_{(i,1)} \geq 0$ , it is a *reliable* FIPE, where  $i \in \mathcal{B}$ .

*Proof:* Based on (1a),  $L_{(i-1,0)} = \alpha \text{sign}(L_{(i-1,1)}) \cdot \text{sign}(L_{(i,1)}) \min(|L_{(i-1,1)}|, |L_{(i,1)}|)$ . Given that  $L_{(i-1,1)} \cdot L_{(i,1)} \geq 0$ , we have  $L_{(i-1,0)} \geq 0$  and  $|L_{(i-1,0)}| = \alpha \min(|L_{(i-1,1)}|, |L_{(i,1)}|)$ . Based on (2), the frozen bit  $\hat{u}_{i-1}$  can be correctly decoded without the help of its  $R_{(i-1,0)}$ . Hence, the FIPE- $i$  is a *reliable* FIPE.  $\square$

**Proposition 2** ([18] [24]). For the *reliable* FIPE- $i$ , its information bit  $u_i$  can be considered successfully decoded and  $|L_{(i,0)}| \geq |L_{(i-1,0)}|$ , where  $i \in \mathcal{B}$ .

## B. Reliable Index Set

Let the ordered set  $\mathcal{B}^h \subseteq \mathcal{B}$  denote the index set of all the *reliable* FIPEs during the  $h$ -th BP iteration. Generally, the set  $\mathcal{B}^h$  can be constructed based on Proposition 1. However,

**Algorithm 1:** Constructing RIS ( $\mathcal{J}^h$ )

```

1 Subroutine ConstructRIS ( $\mathcal{B}$ ,  $\mathcal{S}$ ,  $h$ ):
2   Initialize:  $\mathcal{B}^h \leftarrow \emptyset$ ,  $\mathcal{J}^h \leftarrow \emptyset$ ;
   // Constructing  $\mathcal{B}^h$ 
3   For  $i \in \mathcal{B}$  do
4     Compute  $\gamma_i^h$  as in (3);
5     If  $\gamma_i^h = 0$  then
6        $\mathcal{B}^h \leftarrow \{i\} \cup \mathcal{B}^h$ ;
7   Construct  $\mathcal{J}^h$  as in (5);
8   Return  $\mathcal{J}^h$ 
    
```

this paper introduces a more hardware-friendly method. The detection equation is simplified as

$$\gamma_i^h = \text{sign}(L_{(i-1,1)}) \oplus \text{sign}(L_{(i,1)}), \quad i \in \mathcal{B}, \quad (3)$$

where  $\gamma_i^h$  denotes the checking result for the FIPE- $i$  in the  $h$ -th iteration. If  $\gamma_i^h = 0$ , the FIPE- $i$  is reliable. Otherwise, it is unreliable. Consequently, the set  $\mathcal{B}^h$  is constructed by

$$\mathcal{B}^h = \{i \in \mathcal{B} : \gamma_i^h = 0\}. \quad (4)$$

The critical set (CS), denoted as  $\mathcal{S}$ , is comprised of the index of the first bit in each of the rate-1 (R1) codes. According to [12] [26], the probability that the index of the first error bit is included in  $\mathcal{S}$  approaches one. Since a FIPE contains only a single information bit (i.e., size-1 R1 code), it is intuitive that  $\mathcal{B} \subseteq \mathcal{S}$ . Therefore, based on the detection results of the FIPES, e.g.,  $\mathcal{B}^h$ , the reliable estimations can be identified from  $\mathcal{S}$ .

Let  $\mathcal{J}^h$  denote the RIS obtained in the  $h$ -th iteration. It includes all the bit indices in  $\mathcal{S}$  that lie between every consecutive reliable FIPES  $\mathcal{B}[i]$  and  $\mathcal{B}[i+1]$ , where  $0 \leq i < |\mathcal{B}|$ . Hence, the set  $\mathcal{J}^h$  is constructed by

$$\mathcal{J}^h = \bigcup_{\{\mathcal{B}[i], \mathcal{B}[i+1]\} \in \mathcal{B}^h} \{j \in \mathcal{S} : \mathcal{B}[i] < j < \mathcal{B}[i+1]\}. \quad (5)$$

In particular, when  $i = |\mathcal{B}| - 1$ , we assume that  $\mathcal{B}[i+1] = N - 1$ . It should be pointed out that both the set  $\mathcal{B}$  and the set  $\mathcal{S}$  can be obtained offline. The above mentioned construction of the RIS is summarized as in Algorithm 1.

### C. ABP Decoding Algorithm

According to (1), in the regular PE (except for the FIPES), the values of  $R_{(i,1)}$  remain constant, i.e.,  $R_{(i,1)} \in \{0, +\infty\}$ . This is because during the classical BP decoding, the values of  $R_{(i,0)}$  are fixed and never updated. In [27], it has been shown that the above mentioned phenomenon would significantly slow down the convergence and weaken the BP decoding performance. However, the inefficient propagation can be improved when the *a priori* LLRs of the information bits ( $R_{(i,0)}$ ) are correctly updated [19] [28]. Due to the bidirectional propagation, for the BP decoding, the error bits can be corrected by enhancing the  $R_{(i,0)}$  of the reliable bits [8] [29] (not always being zero). Furthermore, it is worth mentioning that the information bits in the RIS possess the reliable  $L_{(i,0)}$ . Inspired by this, the ABP decoding is proposed. It utilizes the reliable  $L_{(i,0)}$  to renew the  $R_{(i,0)}$  in each BP iteration, instead of making  $R_{(i,0)}$  fixed.

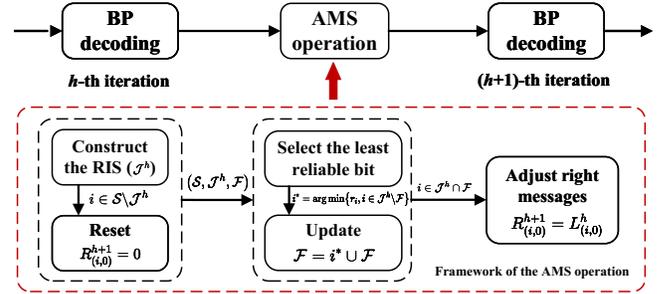


Fig. 2. Block diagram of the proposed ABP decoding.

The block diagram of the proposed ABP decoding is shown in Fig. 2. In each iteration, the ABP decoding updates  $L_{(i,j)}$  and  $R_{(i,j)}$  as in (1). It then invokes the adjusted min-sum (AMS) operation. Let  $R_{(i,j)}^h$  and  $L_{(i,j)}^h$  denote the  $R_{(i,j)}$  and  $L_{(i,j)}$  in the  $h$ -th iteration, respectively. Let us further define set  $\mathcal{F}$  as the indices of the LRBs, which have been adjusted in the previous iteration. Furthermore, the set  $\mathcal{J}^h \setminus \mathcal{F}$  represents the index set of unadjusted LRBs. Fig. 2 also illustrates the framework of the AMS operation, including four key steps:

- 1) Construct the RIS  $\mathcal{J}^h$  using Algorithm 1;
- 2) For each unreliable information bit indexed by  $i \in \mathcal{S} \setminus \mathcal{J}^h$ , reset its right messages to zero, i.e.,  $R_{(i,0)}^{h+1} = 0$ ;
- 3) Select the least reliable bit indexed by  $i^*$  from the set  $\mathcal{J}^h \setminus \mathcal{F}$ , and update  $\mathcal{F}$  as  $\mathcal{F} = \{i^*\} \cup \mathcal{F}$ ;
- 4) For each bit indexed by  $i \in \mathcal{J}^h \cap \mathcal{F}$ , adjust its right message  $R_{(i,0)}^{h+1}$  based on the updated  $L_{(i,0)}^h$ , as in (7).

It should be noted that when compared with other information bits, the LRBs in the RIS are more error-prone in the subsequent iterations [13] [29].

In the  $h$ -th iteration, if the bit indices from  $\mathcal{F}$  are no longer contained in  $\mathcal{J}^h$ , it indicates that the wrong *a priori* LLRs have been set for the LRBs. To rectify this, in Step 2), right messages of the unreliable bits are initialized to zero. In Step 3), the ABP decoder attempts to correct the least reliable bits that have not been adjusted in  $\mathcal{J}^h$ . Its selection can be performed based on the given reliabilities of subchannels  $r_0^{N-1}$ , i.e.,

$$i^* = \arg \min \{r_i, i \in \mathcal{J}^h \setminus \mathcal{F}\}. \quad (6)$$

After that, Step 4) rectifies those LRBs that are indexed in  $\mathcal{J}^h \cap \mathcal{F}$  by feeding back the reliable left messages, i.e.,  $L_{(i,0)}^h$ , into the decoder, and enhancing the right messages as

$$R_{(i,0)}^{h+1} = L_{(i,0)}^h. \quad (7)$$

As the number of iterations increases, the propagated messages become more accurate. Hence, the  $R_{(i,0)}$  of the information bits with lower reliabilities in  $\mathcal{J}^h$  are designed to be adjusted more frequently. For better understanding, the procedure of the ABP decoding is summarized as in Algorithm 2. Note that any efficient ETC can be used in the ABP decoding, and the early termination function is denoted as  $ETC(\cdot)$ .

## IV. THE FIPE BASED ETC

In this section, the characteristics of FIPES are further introduced. Meanwhile, a new ETC for BP decoding is proposed.

**Algorithm 2: The ABP Decoding**


---

**Input:**  $\mathcal{A}, \mathcal{A}_C, \mathcal{B}, \mathcal{S}, I_{\max}, r_0^{N-1}, y_0^{N-1}$   
**Output:**  $\hat{u}_0^{N-1}$

- 1 **Initialize:**  $\mathcal{J}^1 \leftarrow \emptyset, \mathcal{F} \leftarrow \emptyset;$
- 2 **For**  $h \leftarrow 1$  **to**  $I_{\max}$  **do**
- 3     Update  $L_{(i,j)}^h$  and  $R_{(i,j)}^h$  as in (1);
- 4     **If**  $ETC(\hat{u}_0^{N-1}) = true$  **then**
- 5         **Return**  $\hat{u}_0^{N-1};$      // Decoding successful
- 6     **Else**
- 7         // The AMS operation
- 7          $\mathcal{J}^h \leftarrow \text{ConstructRIS}(\mathcal{B}, \mathcal{S}, h);$      // Step 1)
- 8         **For**  $i \in \mathcal{S} \setminus \mathcal{J}^h$  **do**
- 9              $R_{(i,0)}^{h+1} \leftarrow 0;$      // Step 2)
- 10             $i^* \leftarrow \arg \min \{r_i, i \in \mathcal{J}^h \setminus \mathcal{F}\};$      // Step 3)
- 11             $\mathcal{F} \leftarrow \{i^*\} \cup \mathcal{F};$      // Update  $\mathcal{F}$
- 12            **For**  $i \in \mathcal{J}^h \cap \mathcal{F}$  **do**
- 13              $R_{(i,0)}^{h+1} \leftarrow L_{(i,0)}^h;$      // Adjust messages
- 14 **Return**  $\hat{u}_0^{N-1};$      // Decoding failure

---

The proposed FIPE-based ETC can be efficiently integrated with the proposed ABP decoding.

#### A. FIPE-Based ETC

It has been shown in [22] [23] that the characteristics of frozen bits can serve as a reliable indicator for detecting the BP decoding convergence. Based on the channel polarization theory [1], the error probabilities of the frozen bits in FIPEs are usually less than that of other frozen bits. Hence, it is reasonable to conjecture that monitoring the FIPEs can offer a new strategy for terminating the BP decoding.

Let  $H$  and  $D$  denote the indicators for describing the proposed FIPE-based ETC, which are defined as

$$H \triangleq \sum_{\tilde{h}=h-\varphi+1}^h \sum_{i \in \mathcal{B}} \gamma_i^{\tilde{h}} \quad (8)$$

and

$$D \triangleq \sum_{\tilde{h}=h-\varphi+1}^h \sum_{i \in \mathcal{B}} \left| L_{(i,0)}^{\tilde{h}} - L_{(i,0)}^{\tilde{h}-1} \right|, \quad (9)$$

respectively, where  $\varphi$  denotes the number of consecutive iterations. Note that  $H$  and  $D$  represent the reliability of FIPEs and the convergence properties of their LLRs, respectively.

In each BP iteration, the value of  $\gamma_i^h$  is evaluated by (3). When the above evaluations are completed, the FIPE-based ETC computes  $H$  and  $D$  according to (8) and (9), respectively. If  $H = 0$  and  $D = 0$ , it indicates that over the  $\varphi$  consecutive iterations, all the frozen bits in the FIPEs are successfully decoded with stable LLRs. In this case, the decoding can be terminated. Note that since  $\gamma_i^h$  has been computed in Algorithm 1, the proposed criterion is more suitable for the ABP decoder.

#### B. Complexity Evaluation

This subsection evaluates the complexity of the FIPE-based ETC by the number of hardware operations. For each FIPE,

TABLE I  
HARDWARE CONSUMPTIONS OF DIFFERENT ETCs FOR ONE ITERATION USING THE ABP DECODER

	Adder	OR	XOR	Comp
Proposed	$2 \mathcal{B}  + \varphi - 1$	$ \mathcal{B}  + \varphi - 2$	–	–
G-matrix	$2N$	–	$Nn/2$	$3N$
WIB [21]	$2N/8 + \varphi$	–	$N/8$	–
BFB [22]	–	$N/16 - 1$	–	$N/16$

it requires one 2-input XOR gate for computing (3). Thus, computing  $H$ , as in (8), requires  $|\mathcal{B}|$  XORs. However, please note that since  $\gamma_i^h$  is reused in the ABP decoder, implying the FIPE-based ETC does not require any XOR operation for obtaining  $H$ . Moreover, due to  $\gamma_i^h \in \{0, 1\}$ , the 2-input OR gate can perform the addition operations in (8). Thus, computing  $H$  requires  $|\mathcal{B}| + \varphi - 2$  ORs. Considering (9), it requires  $2|\mathcal{B}| + \varphi - 1$  Adders for computing  $D$ . Thus, the number of addition operations are  $2|\mathcal{B}| + \varphi - 1$ . In particular, the proposed criterion does not require any comparison (Comp) operation.

Table I shows the hardware complexity of different ETCs during one iteration using the ABP decoder. It can be seen that compared with other ETCs, the FIPE-based ETC does not require any XOR or Comp operations, yielding a lower hardware resource cost than those of others. Moreover, the hardware complexity of the proposed criterion increases linearly with  $|\mathcal{B}|$ . In order to reduce complexity, we can only detect a portion of the bits from the set  $\mathcal{B}$  with lower reliabilities.

## V. SIMULATION RESULTS

This section presents simulation results of the proposed ABP decoder and that with the FIPE-based ETC. The 5G standard polar codes [30] are considered. The length-2048 polar code is constructed by the Gaussian approximation (GA) [31] with a *design*-SNR of 2 dB. The polar codes with  $N \in \{512, 1024, 2048\}$  and  $R \in \{1/4, 1/2, 2/3, 3/4\}$  are considered. The codewords are modulated using the binary phase shift keying (BPSK) and transmitted over the additive white Gaussian noise (AWGN) channel. The maximum number of iterations  $I_{\max}$  for the ABP decoding is 100.

#### A. Performance of the ABP Decoder

Fig. 3 shows the frame error rate (FER) performance of the proposed ABP decoding against the state-of-the-art BP decoding [20] and the SC decoding. As shown in Fig. 3(a), the ABP decoding outperforms the BP decoding for various codeword lengths  $N \in \{512, 1024, 2048\}$  and rate  $R = 1/2$ . For the  $\mathcal{P}(1024, 512)$  code, the ABP decoding achieves a performance gain of 0.3 dB at the FER of  $10^{-4}$  over the BP decoding. Moreover, in high SNR regime (e.g.,  $\text{SNR} > 3$  dB), performance of the ABP decoding is similar to that of the SC decoding. Fig 3(b) compares the ABP decoding and BP decoding for rate  $R \in \{1/4, 2/3, 3/4\}$  and  $N = 1024$ . The ABP decoding still exhibits its performance advantage, yielding performance gains of at least 0.15 dB over the BP decoding.

In this paper, the decoding latency is measured by the average number of required clock cycles (CCs). The latency

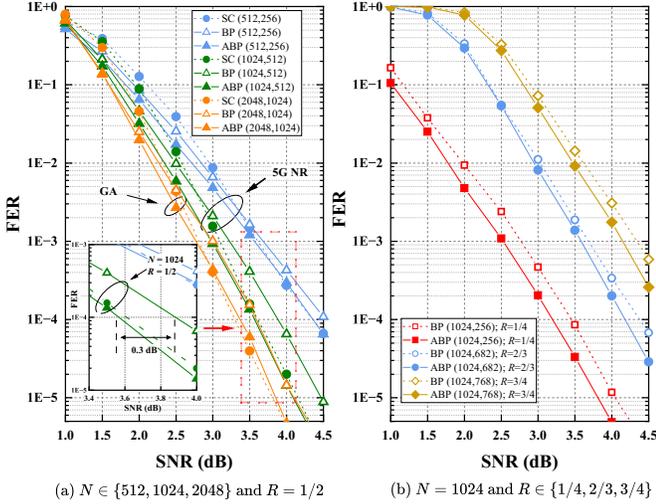


Fig. 3. Performance comparison between the BP, the SC and the proposed ABP decoding under various  $N$  and  $R$ . The ETC is the G-matrix criterion.

TABLE II  
AVERAGE NUMBER OF DECODING CCs FOR THE  $\mathcal{P}(1024, 512)$  CODE

SNR (dB)	2.5	3	3.5	4
FSSC [6]	427	427	427	427
BP (G-matrix) [20]	174	124	102	88
ABP (G-matrix)	172	128	106	92

of the FSSC decoder is evaluated based on [6]. Let  $I_{\text{avg}}$  denote the average number of iterations. Thus, the latency of the BP decoder is  $2I_{\text{avg}} \log_2 N$  [20]. In each iteration, the ABP decoder requires  $2 \log_2 N$  CCs to update messages. Since the AMS and the ETC can perform in parallel, additional one CC is needed. Overall, the average latency required by the ABP decoder is

$$\bar{\tau}_{\text{ABP}} = I_{\text{avg}} \cdot (2 \log_2 N + 1). \quad (10)$$

Table II presents the average latency for different polar decoders in decoding the  $\mathcal{P}(1024, 512)$  code. It is notable that the decoding latency of the ABP decoder is slightly higher than the BP decoder. They all require less than 200 CCs for the SNR range of 2.5 ~ 4 dB. As the SNR increases, the average number of iterations of the ABP decoder decreases rapidly, leading to a smaller number of required CCs. Therefore, its latency advantage becomes more significant in the high SNR region. In particular, at the SNR of 4 dB, the ABP decoder consumes only 92 CCs for decoding the  $\mathcal{P}(1024, 512)$  code, while the FSSC decoder requires 427 CCs. As a result, the ABP decoder realizes a 78.5% latency reduction over the FSSC decoder, while preserving the SC decoding performance.

### B. Performance of the FIPE-Based ETC

Fig. 4 shows the performance and the average iterations of the ABP decoder using different ETCs for the  $\mathcal{P}(1024, 512)$  code, where  $|\mathcal{B}| = 54$ . For the FIPE-based ETC, the number of detected bits in  $\mathcal{B}$  is 35, i.e.,  $N_{\text{FIPE}} = 35$ . Let  $N_{\text{WIB}}$  and  $N_{\text{BFB}}$  denote the number of detected bits for the WIB and the BFB criteria, respectively. In the simulation, it is set  $N_{\text{WIB}} = 128$  and  $N_{\text{BFB}} = 64$ . Note that the minimum threshold of LLR  $\theta$  in the BFB criterion is 13 [23]. Fig. 4(a) shows that the

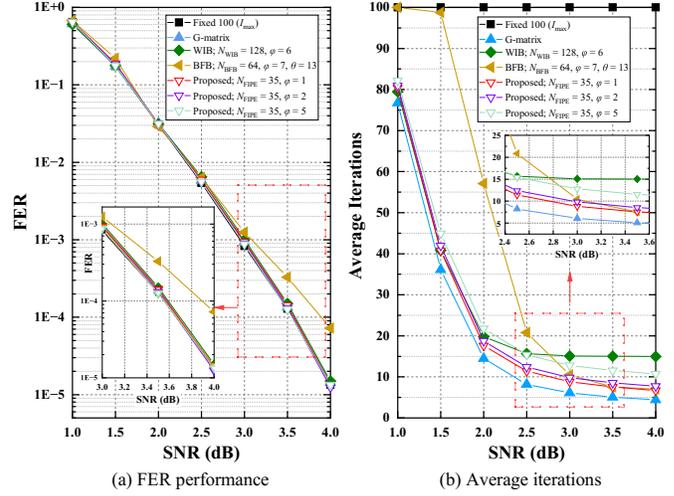


Fig. 4. Performance comparison under various ETCs using the ABP decoder for the  $\mathcal{P}(1024, 512)$  code.

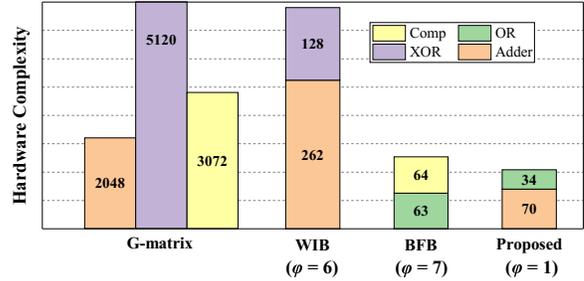


Fig. 5. Hardware complexity of the ABP decoder using various ETCs for the  $\mathcal{P}(1024, 512)$  code.

proposed criterion effectively terminates the decoding process with negligible FER degradation for  $\varphi \in \{1, 2, 5\}$ . Fig. 4(b) further shows that the proposed criterion achieves an iteration number slightly higher than that of the G-matrix criterion, but lower than that of the other ETCs. In particular, when  $\varphi = 1$ , the average iterations for the proposed criterion are lower than the WIB criterion ( $\varphi = 6$ ) by 49.2%. Although the BFB criterion ( $\varphi = 7$ ) yields a similar performance as the proposed criterion when  $\text{SNR} < 3$  dB, it requires a larger number of iterations. Furthermore, for the proposed criterion, the  $I_{\text{avg}}$  is mainly affected by the value of  $\varphi$ . Based on these analyses, the optimal setting for the FIPE-based criterion is  $\varphi = 1$ .

Fig. 5 shows the hardware complexity for different ETCs during a single iteration using the ABP decoder, where  $N = 1024$ . When  $N_{\text{FIPE}} = 35$  and  $\varphi = 1$ , the FIPE-based ETC requires 70 Adders and 34 ORs, which is lower than that of the G-matrix, the WIB ( $\varphi = 6$ ), and the BFB ( $\varphi = 7$ ) criteria. Fig. 4 and 5 show that the proposed criterion has the lowest hardware complexity compared with the existing ETCs, such as the G-matrix, the WIB, and the BFB criteria, while maintaining the error-correction performance.

### ACKNOWLEDGEMENT

This work is supported in part by the National Natural Science Foundation of China (NSFC) with project ID 62071498; and in part by the Natural Science Foundation of Guangdong Province with project ID 2024A1515010213.

## REFERENCES

- [1] E. Arıkan, "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] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.
- [3] K. Chen, K. Niu, and J. Lin, "List successive cancellation decoding of polar codes," *Electron. Lett.*, vol. 48, no. 9, pp. 500–501, Apr. 2012.
- [4] K. Niu and K. Chen, "CRC-aided decoding of polar codes," *IEEE Commun. Lett.*, vol. 16, no. 10, pp. 1668–1671, Oct. 2012.
- [5] 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.
- [6] G. Sarkis, P. Giard, A. Vardy and *et al.*, "Fast polar decoders: Algorithm and implementation," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, pp. 946–957, May 2014.
- [7] E. Arıkan, "A performance comparison of polar codes and reed-muller codes," *IEEE Commun. Lett.*, vol. 12, no. 6, pp. 447–449, Jun. 2008.
- [8] S. Sun, S.-G. Cho, and Z. Zhang, "Post-processing methods for improving coding gain in belief propagation decoding of polar codes," in *Proc. 2017 IEEE Glob. Commun. Conf. (GLOBECOM)*, Singapore, Dec. 2017.
- [9] B. Feng, R. Liu, and K. Tian, "A novel post-processing method for belief propagation list decoding of polar codes," *IEEE Commun. Lett.*, vol. 25, no. 8, pp. 2468–2471, Aug. 2021.
- [10] M. Zhang, Z. Li, and L. Xing, "An enhanced belief propagation decoder for polar codes," *IEEE Commun. Lett.*, vol. 25, no. 10, pp. 3161–3165, Oct. 2021.
- [11] H. Ji, Y. Shen, W. Song and *et al.*, "Hardware implementation for belief propagation flip decoding of polar codes," *IEEE Trans. Circuits Syst. I*, vol. 68, no. 3, pp. 1330–1341, Mar. 2021.
- [12] Y. Yu, Z. Pan, N. Liu and *et al.*, "Belief propagation bit-flip decoder for polar codes," *IEEE Access*, vol. 7, pp. 10 937–10 946, Jan. 2019.
- [13] Y. Shen, W. Song, H. Ji and *et al.*, "Improved belief propagation polar decoders with bit-flipping algorithms," *IEEE Trans. Commun.*, vol. 68, no. 11, pp. 6699–6713, Nov. 2020.
- [14] A. Elkelesh, M. Ebada, S. Cammerer and *et al.*, "Belief propagation list decoding of polar codes," *IEEE Commun. Lett.*, vol. 22, no. 8, pp. 1536–1539, Aug. 2018.
- [15] N. Doan, S. A. Hashemi, M. Mondelli and *et al.*, "On the decoding of polar codes on permuted factor graphs," in *Proc. 2018 IEEE Glob. Commun. Conf. (GLOBECOM)*, Abu Dhabi, United Arab Emirates, Dec. 2018.
- [16] A. C. Arlı and O. Gazi, "Noise-aided belief propagation list decoding of polar codes," *IEEE Commun. Lett.*, vol. 23, no. 8, pp. 1285–1288, Aug. 2019.
- [17] C. Wang, X. Wu, W. Wei and *et al.*, "Joint design of permutations and polar code under BPL decoding," *IEEE Commun. Lett.*, vol. 27, no. 1, pp. 41–45, Jan. 2023.
- [18] X. Wu, W. Wei, S. Zhang and *et al.*, "An efficient belief propagation list flip decoder for polar codes," *IEEE Commun. Lett.*, vol. 27, no. 9, pp. 2264–2268, Sep. 2023.
- [19] Y. Ren, W. Xu, Z. Zhang and *et al.*, "Efficient belief propagation list decoding of polar codes," in *Proc. 2019 IEEE Int. Conf. on ASIC (ASICON)*, Chongqing, China, Oct. 2019.
- [20] B. Yuan and K. K. Parhi, "Early stopping criteria for energy-efficient low-latency belief-propagation polar code decoders," *IEEE Trans. Signal Process.*, vol. 62, no. 24, pp. 6496–6506, Dec. 2014.
- [21] C. Simsek and K. Turk, "Simplified early stopping criterion for belief-propagation polar code decoders," *IEEE Commun. Lett.*, vol. 20, no. 8, pp. 1515–1518, Aug. 2016.
- [22] Q. Zhang, A. Liu, and X. Tong, "Early stopping criterion for belief propagation polar decoder based on frozen bits," *Electron. Lett.*, vol. 53, no. 24, pp. 1576–1578, Nov. 2017.
- [23] Y. Yan, X. Zhang, and B. Wu, "Simplified early stopping criterion for belief-propagation polar code decoder based on frozen bits," *IEEE Access*, vol. 7, pp. 134 691–134 696, Sep. 2019.
- [24] C. Chen, X. Zhang, Y. Liu and *et al.*, "An ultra-low complexity early stopping criterion for belief propagation polar code decoder," *IEEE Commun. Lett.*, vol. 26, no. 4, pp. 723–727, Apr. 2022.
- [25] B. Yuan and K. K. Parhi, "Architecture optimizations for BP polar decoders," in *Proc. 2013 IEEE Int. Conf. Acoustics, Speech Signal Proc. (ICASSP)*, Vancouver, Canada, May 2013.
- [26] Z. Zhang, K. Qin, L. Zhang and *et al.*, "Progressive bit-flipping decoding of polar codes over layered critical sets," in *Proc. 2017 IEEE Glob. Commun. Conf. (GLOBECOM)*, Singapore, Dec. 2017.
- [27] Y. Ren, Y. Shen, Z. Zhang and *et al.*, "Efficient belief propagation polar decoder with loop simplification based factor graphs," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5657–5660, May 2020.
- [28] S. Lin and D. J. Costello, *Error Control Coding*. Prentice Hall, 2004.
- [29] Y. Shen, W. Song, Y. Ren and *et al.*, "Enhanced belief propagation decoder for 5G polar codes with bit-flipping," *IEEE Trans. Circuits Syst. II*, vol. 67, no. 5, pp. 901–905, May 2020.
- [30] "Technical specification group radio access network," 3GPP, TS 38.212 version 15.2.0, Jul. 2018. [Online]. Available: [https://www.3gpp.org/ftp/Specs/archive/38\\_series/38.212/](https://www.3gpp.org/ftp/Specs/archive/38_series/38.212/)
- [31] P. Trifonov, "Efficient design and decoding of polar codes," *IEEE Trans. Commun.*, vol. 60, no. 11, pp. 3221–3227, Nov. 2012.