A Study on Polar Codes

Yue Fei 1003944146

I. Introduction

Polar codes emerged as the first class of error correction codes with proven capacity-achieving ability for binary-input discrete memoryless (symmetric) channels (B-DMC) in 2009 [1]. Although theoretically polar codes works beautifully with low encoding and decoding complexities, it suffers huge drawbacks especially decoding latency for practices.

In the past decade, as the ongoing 5^{th} generation wireless systems (5G) standardization process of the 3rd generation partnership project (3GPP) rolls on, polar code has now been widely adopted in uplink and downlink control channels [2]. Motivated by the alike interest in both academia and industries, I wish to obtain a fundamental understanding of the construction of polar code, the encoding process and various decoding algorithms with different performances.

A. Preliminaries

Before diving into polar code, we first introduce the notations used throughout this paper. We consider a generic B-DMC *bit* channel model denoted as: $W : X \rightarrow Y$ with input alphabet $\mathcal{X}$, output alphabet $\mathcal{Y}$, and transition probabilities $W(y|x)$, $x \in \mathcal{X}$, $y \in \mathcal{Y}$ as shown in Fig. 1. $W_X(x)$ denotes the input frequency. Code rate R is the ratio between the number of information bits K and the codeword length N. W^N indicates N independent uses of the bit channel W, which explicitly can be written as $W^N(y_i^N|x_i^N) = \Pi_{i=1}^{N} W(y_i|x_i)$ for $W^N : X^N \rightarrow Y^N$.

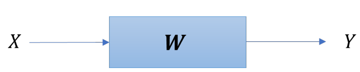


Fig. 1: General Channel Model

Now, if looking into the bit channels jointly, we can combine N copies of W into a vector channel $W_N : X^N \rightarrow Y^N$. Let $W_N^{(i)} : X_i \rightarrow Y^N \times X^{i-1}, 1 \leq i \leq N$ denotes the i^{th} synthetic channel, which associates input X_i with the output vector Y^N and all the previous inputs up to X^{i-1}, as depicted in Fig. 2.

In this paper, we are only working with binary field $GF(2)$, hence, the input alphabet $\mathcal{X}$ will always be $0, 1$, the output alphabet and the transition probabilities may be arbitrary depending on the channel definitions.

Given this setup, we introduce two primary channel parameters that are the essential building blocks of polar code: 1) symmetric capacity $I(W)$ as a measure of rate

$$I(W) \triangleq \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} \frac{1}{2} W(y|x) \log \frac{W(y|x)}{\frac{1}{2}W(y|0) + \frac{1}{2}W(y|1)} \quad (1)$$

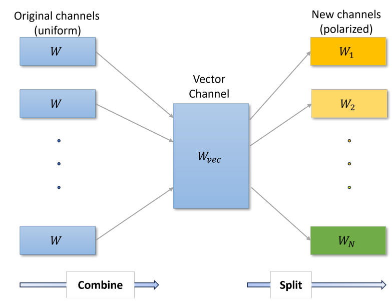


Fig. 2: Generic Channel Combining and Splitting [3]

2) the Bhattacharyya parameter $Z(W)$ to describe channel reliability

$$Z(W) \triangleq \sum_{y \in \mathcal{Y}} \sqrt{W(y|0)W(y|1)} \quad (2)$$

$I(W)$ is the highest rate for reliable communication across W with equal probable inputs. $Z(W)$ is the upper limit on the maximum-likelihood decision error probability of the binary input channel. Bhattacharyya parameter has its root in Bhattacharyya distance $Z_{x,x'}(W)$ in statistics, which measures the similarity of two probability distributions [4].

II. Channel Polarization

Arikan's polar code [1] achieves channel polarization by recursively invoking the simple 2-bit encoding kernel of for "channel combining" at the encoder.

$$\boldsymbol{G_2} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

The splitting process at the decoder. More specifically, two uses of the B-DMC are combined using a single eXclusive-OR (XOR) gate demonstrated in Fig 3b. The resulting compound channel at the encoder end is denoted by W^2, which has a capacity of

$$C(W^2) = I(u_1, u_2; y_1, y_2) = I(x_1, x_2; y_1, y_2) = 2 \times I(W) \quad (3)$$

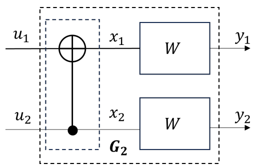
With chain rule of mutual information [5], we can express Eq.3 in terms of conditional mutual information:

$$\begin{aligned} C(W^2) &= I(u_1, u_2; y_1, y_2) && (4) \\ &= I(u_1; y_1, y_2) + I(u_2; y_1, y_2|u_1) && (5) \\ &= I(u_1; y_1, y_2) + I(u_2; y_1, y_2, u_1) && (6) \end{aligned}$$

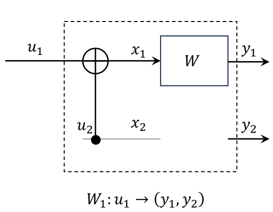
since u is independently and identically distributed. u_1 and u_2 are independent. Eq. **??** implies that we may decode the compound channel W^2 bit-by-bit as shown in Fig3b and Fig

3c. Hence, we have the channel capacity W_1 and W_2 as follows:

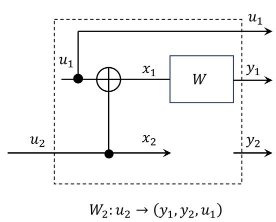
$$W_1 \triangleq u_1 \rightarrow (y_1, y_2) \tag{7}$$
$$W_2 \triangleq u_2 \rightarrow (y_1, y_2, u1)$$



(a) Channel combining



(b) Channel splitting: first bit channel



(c) Channel splitting: second bit channel

Fig. 3: Arikan's 2-bit polar code, relying on channel combining and channel splitting for channel polarization

For simplicity and later construction of polar code, we consider binary erasure channel (BEC) with erasure probability ϵ. The first bit-channel W_1 of Fig. 3b has no information about u_2. So, the receiver estimates u_1 based on the received bits y_1 and y_2 as follows:

$$\hat{u_1} = y_1 \oplus y_2 \tag{8}$$

, where $\oplus$ denotes modulo-2 addition. u_1 can be decoded only when neither y_1 nor y_2 is erased. Hence, the erasure probability of the first virtual channel W_1 is

$$\epsilon_1 = 1 - (1-\epsilon)^2 = 2\epsilon - \epsilon^2 \tag{9}$$

Then, W_1 is slightly worse than the original bit channel. We denote this channel as W^- with reduced capacity:

$$I(W^-) = 1 - \epsilon_1 = 1 - 2\epsilon + \epsilon^2 \tag{10}$$

When decoding the second bit channel of Fig. 3c, we see that there are three outputs. The additional output of u_1 is analogous to a side channel with full information of u_1. Jointly decoding with this additional side channel, we can reveal the value of u_2 when either y_1 or y_2 is not erased.

Assuming $\epsilon = 0.5$, we can see from Fig. 4, the first channel was driven more towards zero capacity and the last channel was polarized more towards one just with a second layer of XOR gate.

III. Construction of Polar Code

As shown in the previous section, the capacity-achieving ability of polar codes depend on a set of good and bad channels, so that information bits can be transmitted through

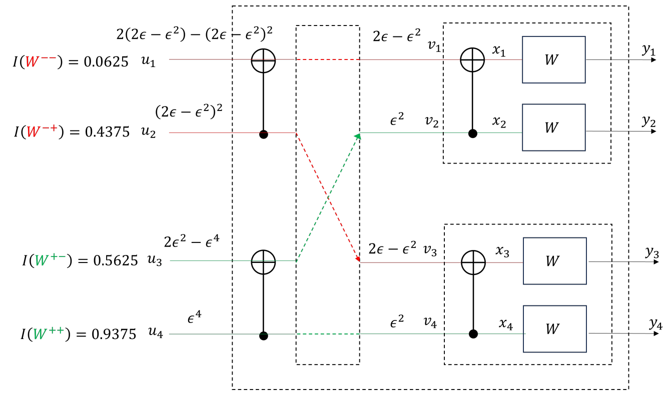


Fig. 4: 4-bit compound channel W^4 with natural indexing

the induced good channels, while the input to the induced bad channels are frozen or known to both the encoder and decoder. Hence, a polar encodes k information bits into N coded bits using $(N-k)$ redundant bits, which are called the "frozen bits". It can be characterized by the parameters (N, k, A_c), where $A_c \subset 1, 2, \ldots, N$ specifies the location of frozen bits. The location of frozen bits can be determined by using Bhattacharyya parameter. Computationally, it is equivalent to the recursively calculated channel capacity of BEC with erasure probability ϵ Eq. 11.

$$Z(W_{2i-1}^N) = 2 * Z(W_i^{N/2}) - Z(W_i^{N/2})^2 \tag{11}$$
$$Z(W_{2i}^N) = Z(W_i^{N/2})^2 \tag{12}$$

The polar encoding process may be done iteratively by combining two branches at a time, with an overall complexity of $\mathcal{O}(Nlog_2N)$. If we view the encoding operations as a full binary tree structure, we combine the sequences of left and right children recursively. The log_2N complexity comes from the depth of tree. N corresponds to the constant $N/2$ combinations at each tree depth.

IV. Formalization of the Successive Cancellation Decoder

The Successive Cancellation (SC) decoder described here follows the general idea due to Arikan [1], but recasted using Tal et al's [6] notation for programmability.

Let the polar code under consideration have length $N = 2^m$ and dimension k. Thus, the number of frozen bits is $n-k$. We denote the information bits vector (including the frozen bits) as $\mathbf{u} = (u_i)_{i=0}^{N-1} = u_0^{N-1}$, and the corresponding codedword to be transmitted as $\mathbf{c} = c_0^{N-1}$. At the receiver end, we obtain $\mathbf{y} = y_0^{N-1}$. A decoding algorithm is then applied to $\mathbf{y}$, resulting in a decoded codeword $\hat{c}$ having corresponding information bits $\hat{u}$.

A. An outline of Successive Cancellation

A high-level description of the SC decoding algorithm is given in Algorithm 1. In words, at each phase ϕ of the algorithm, we must first calculate the pair of probabilities $W_m^{\phi}(y_0^{N-1}, \hat{u}_0^{\phi-1}|0)$ and $W_m^{\phi}(y_0^{N-1}, \hat{u}_0^{\phi-1}|1)$, defined shortly.

Then, we must make a decision as to the value of $\hat{u}_\phi$ according to the pair of probabilities.

Note that to coordinate well with modern programming languages such C/C++, ϕ starts at 0 and goes up to $N-1$.

Algorithm 1: A high-level description of the SC decoder

Input: the received vector $\mathbf{y}$
Output: a decoded codeword $\hat{\mathbf{c}}$

for $\phi = 0, 1, \ldots, N-1$ **do**
  calculate $W_m^{(\phi)}(\mathbf{y}_0^{N-1}, \hat{\mathbf{u}}_0^{\phi-1}|0)$ and $W_m^{(\phi)}(\mathbf{y}_0^{N-1}, \hat{\mathbf{u}}_0^{\phi-1}|1)$
  if u_ϕ is frozen **then**
    set $\hat{u}_\phi$ to the *frozen* value of u_ϕ
  else
    if $W_m^{(\phi)}(\mathbf{y}_0^{N-1}, \hat{\mathbf{u}}_0^{\phi-1}|0) > W_m^{(\phi)}(\mathbf{y}_0^{N-1}, \hat{\mathbf{u}}_0^{\phi-1}|1)$ **then**
      set $\hat{u}_\phi \leftarrow 0$
    else
      set $\hat{u}_\phi \leftarrow 1$

return the codeword $\hat{\mathbf{c}}$ corresponding to $\hat{\mathbf{u}}$

Consider in a full binary tree setup, at each *layer* $\lambda \in [0, m]$, we have

$$\Lambda = 2^\lambda \tag{13}$$

number of nodes. The node index is $\phi \in [0, \ldots, \Lambda)$.

To achieve polarization, a block-wise channel combining and splitting is implemented, which transformed N independent bit channels into a vector channel W_N of size N. The ϕ^{th} bit of the new vector channel output is denoted as $W_N^{(\phi)}$. For traditional transistor circuit, there exists a one-to-one mapping $f : \mathcal{Y} \rightarrow \bar{\mathcal{Y}}$ [1, 7] (Further studies on quantum circuits are not in the scope of this work). The pair of channel output after a single-step transformation of two independent uses of a binary input channel $W : \mathcal{X} \rightarrow \mathcal{Y}$ are $W' : \mathcal{X} \rightarrow \bar{\mathcal{Y}}$ and $W'' : \mathcal{X} \rightarrow \bar{\mathcal{Y}} \times \mathcal{X}$. We rewrite the two mappings jointly

$$(W, W) \mapsto (W_2^{(0)}, W_2^{(1)})$$
$$(W, W) \mapsto (W', W'')$$

Then we can express the transition probabilities of the vector channel as

$$\begin{aligned} W_2^{(0)}(y_0^1|u_0) &\overset{Eq.(5)}{\triangleq} \sum_{u_1} \frac{1}{2} W_2(y_0^1|u_0^1) \\ &= \sum_{u_1} \frac{1}{2} W(y_0|u_0 \oplus u_1) W(y_1|u_1) \end{aligned} \tag{14}$$

$$\begin{aligned} W_2^{(1)}(y_0^1, u_0|u_1) &\triangleq \frac{1}{2} W_2(y_0^1|u_0^1) \\ &= \frac{1}{2} W(y_0|u_0 \oplus u_1) W(y_1|u_1), \end{aligned} \tag{15}$$

where (y_0^1, u_0) denotes the output of $W_2^{(1)}$ with its input u_1. The coefficient of $\frac{1}{2}$ is due to the equal probable input $u \in \mathcal{X}$. If a data vector $u_i^K \in \mathcal{X}^K$ is sent, then the input probability is a-priori uniform on $\mathcal{X}^K$, which becomes 2^{-K}.

Furthermore, we can prove such property can be extended recursively for $\phi = 0, 1, \ldots, N-1$, with the aid of channel splitting. The effective channel W_N^i seen by the ith decision element, has the following transition probabilities:

$$W_N^{(\phi)}(y_0^{N-1}, u_0^{\phi-1}|u_\phi) \triangleq \sum_{u_{\phi+1}^N \in \mathcal{X}^{N-\phi}} \frac{1}{2^{N-1}} W_N(y_0^{N-1}|u_0^{N-1}), \tag{16}$$

where $(y_0^{N-1}, u_0^{\phi-1})$ denotes the output of $W_N^{(\phi)}$ and its input u_ϕ. With the polarization effect introduced in section II, we assume that the ith decision element has full knowledge of the past channel input $u_0^{\phi-1}$. The uncertainty of the channel now is only contained in $\phi' \in [\phi+1, N]$.

In general, the earlier observed channel (*even* for *0-based* indexing) can be expressed as

$$\begin{aligned} &W_{2N}^{(2\phi)}(y_0^{2N-1}, u_0^{2\phi-1}|u_{2\phi}) \\ \text{Channel splitting} &\overset{Eq.(16)}{=} \sum_{u_{2\phi+1}^{2N-1}} \frac{1}{2^{2N-1}} W_{2N}(y_0^{2N-1}|u_0^{2N-1}) \\ \text{Channel combining} &\overset{Eq.(15)}{=} \sum_{u_{2\phi+1,even}^{2N-1}, u_{2\phi+1,odd}^{2N-1}} \frac{1}{2^{2N-1}} \\ &\cdot W_N(y_0^{N-1}|u_{0,even}^{2N-1} \oplus u_{0,odd}^{2N-1}) \\ &\cdot W_N(y_{N+1}^{2N-1}|u_{0,even}^{2N-1}) \\ &= \sum_{u_{2\phi+1}} \frac{1}{2} \overbrace{\sum_{u_{2\phi+2,odd}^{2N-1}} \frac{1}{2^{N-1}} W_N(y_{N+1}^{2N-1}|u_{0,odd}^{2N-1})}^{\text{branch 2}} \\ &\cdot \underbrace{\sum_{u_{2\phi+2,even}^{2N-1}} \frac{1}{2^{N-1}} W_N(y_0^{N-1}|u_{0,even}^{2N-1} \oplus u_{0,odd}^{2N-1})}_{\text{branch 1}} \end{aligned} \tag{17}$$

We can simplify *branch 1* using definition (16) and rewrite it as $W_N^{(\phi)}(y_0^{N-1}, u_{0,even}^{2\phi-1} \oplus u_{0,odd}^{2\phi-1}|u_{2\phi} \oplus u_{2\phi+1})$, since $u_{2\phi+2,even}^{2N-1}$ ranges over $\mathcal{X}^{N-\phi}$, $u_{2\phi+2,even}^{2N-1} \oplus u_{2\phi+2,odd}^{2N-1}$ also ranges over $\mathcal{X}^{\mathcal{N}-\prec}$. (β) term can be expressed as $W_N^{(\phi)}(y_{N+1}^{2N-1}, u_{0,odd}^{2\phi-1}|u_{2\phi+3})$. Altogether, the recursive definition of the vector channel is

$$\begin{aligned} &W_{2N}^{(2\phi)}(y_0^{2N-1}, u_0^{2\phi-1}|u_{2\phi}) \\ &= \sum_{u_{2\phi+1}} \frac{1}{2} W_N^{(\phi)}(y_0^{N-1}, u_{0,even}^{2\phi-1} \oplus u_{0,odd}^{2\phi-1}|u_{2\phi} \oplus u_{2\phi+1}) \\ &\cdot W_N^{(\phi)}(y_{N+1}^{2N-1}, u_{0,odd}^{2\phi-1}|u_{2\phi+1}) \end{aligned} \tag{18}$$

For latter observed channel (*odd* for *0-based* indexing) is expressed as in [1] Eq. (23).

$$\begin{aligned} &W_{2N}^{(2\phi+1)}(y_0^{2N-1}, u_0^{2\phi}|u_{2\phi+1}) \\ &= \sum_{u_{2\phi+2}^{2N-1}} \frac{1}{2^{2N-1}} W_{2N}(y_0^{2N-1}|u_0^{2N-1}) \\ &= \frac{1}{2} \overbrace{\sum_{u_{2\phi+2,odd}^{2N-1}} \frac{1}{2^{N-1}} W_N(y_{N+1}^{2N-1}|u_{0,odd}^{2N-1})}^{\text{branch 2}} \\ &\cdot \underbrace{\sum_{u_{2\phi+2,even}^{2N-1}} \frac{1}{2^{N-1}} W_N(y_0^{N-1}|u_{0,even}^{2N-1} \oplus u_{0,odd}^{2N-1})}_{\text{branch 1}} \end{aligned} \tag{19}$$

Similar arguments from above apply, we can reduce 19 to be

$$\begin{aligned}
&W_{2N}^{(2\phi)}(y_0^{2N-1}, u_0^{2\phi-1}|u_{2\phi}) \\
&= \frac{1}{2} W_N^{(\phi)}(y_0^{N-1}, u_{0,even}^{2\phi-1} \oplus u_{0,odd}^{2\phi-1}|u_{2\phi} \oplus u_{2\phi+1}) \\
&\cdot W_N^{(\phi)}(y_{N+1}^{2N-1}, u_{0,odd}^{2\phi-1}|u_{2\phi+1})
\end{aligned} \tag{20}$$

Rewrite 18 20 in terms of layer λ and the node index in each layer. Let $1 \leq \lambda \leq m$ and $0 \leq 2\psi < \Lambda$

$$\begin{aligned}
&\overbrace{W_\lambda^{(2\psi)}(y_0^{\Lambda-1}, u_0^{2\psi-1}|u_{2\psi})}^{\text{branch } \beta} \\
&= \sum_{u_{2\psi+1}} \frac{1}{2} \underbrace{W_{\lambda-1}^{(\psi)}(y_0^{\Lambda/2-1}, u_{0,even}^{2\psi-1} \oplus u_{0,odd}^{2\psi-1}|u_{2\psi} \oplus u_{2\psi+1})}_{\text{branch } 2\beta} \\
&\cdot \underbrace{W_{\lambda-1}^{(\psi)}(y_{\Lambda/2}^{\Lambda-1}, u_{0,odd}^{2\psi-1}|u_{2\psi+1})}_{\text{branch } 2\beta+1}
\end{aligned} \tag{21}$$

$$\begin{aligned}
&\overbrace{W_\lambda^{(2\psi)}(y_0^{\Lambda-1}, u_0^{2\psi-1}|u_{2\psi})}^{\text{branch } \beta} \\
&= \frac{1}{2} \underbrace{W_{\lambda-1}^{(\psi)}(y_0^{\Lambda/2-1}, u_{0,even}^{2\psi-1} \oplus u_{0,odd}^{2\psi-1}|u_{2\psi} \oplus u_{2\psi+1})}_{\text{branch } 2\beta} \\
&\cdot \underbrace{W_{\lambda-1}^{(\psi)}(y_{\Lambda/2}^{\Lambda-1}, u_{0,odd}^{2\psi-1}|u_{2\psi+1})}_{\text{branch } 2\beta+1}
\end{aligned} \tag{22}$$

B. Implementation Details

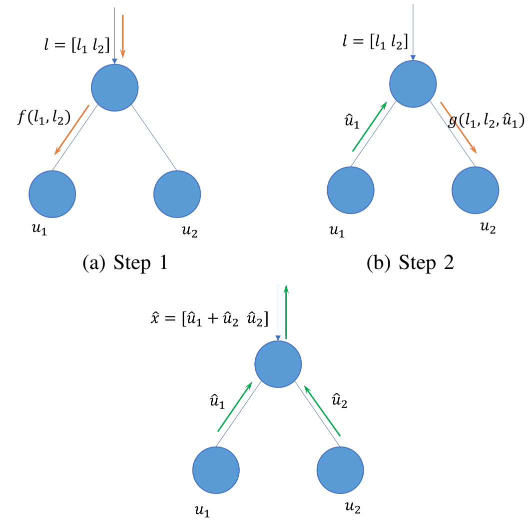
The naive SC decoder introduced in [6] *Algorithm 2* works theoretically. However, the probabilities may quickly vanish when we compute backward, i.e. from output end to input end.

Here we show an SC decoder in *"probability of 1" (P1)* domain for BEC [8].

Algorithm 2: SC Decoder Implementation

function POLARDECODEBEC
Input: **y** output bit APP from channel in output order, **P** input a prior probs in input order
Output: **x** output hard decision in output order, **u** input hard decision in input order

```
N = length(y)
if N == 1 then
    if P == 1/2 then // info bit
        x = y; u = x
    else // frozen bit
        x = P; u = P
else
    // left child
    LLR_1 ← fOp(y(1 : 2 : end), y(2 : 2 : end))
    û_{1,curr}, û_{1,prev} ← POLARDECODEBEC(LLR_1, P(1 : N/2))
    // right child
    LLR_2 ← gOp(fOp(û_{1,prev}, y(1 : 2 : end)), y(2 : 2 : end))
    û_{2,curr}, û_{2,prev} ← POLARDECODEBEC(LLR_2, P(N/2+1 : N))
    // parent node
    u ← [û_{1,curr}, û_{2,curr}]
    x ← reshape([fOp(û_{1,prev}, û_{2,prev}); û_{2,prev})], 1, [ ])
```



(a) Step 1 (b) Step 2 (c) Step 3

Fig. 5: Tree diagram of SC decoder for $N = 2$

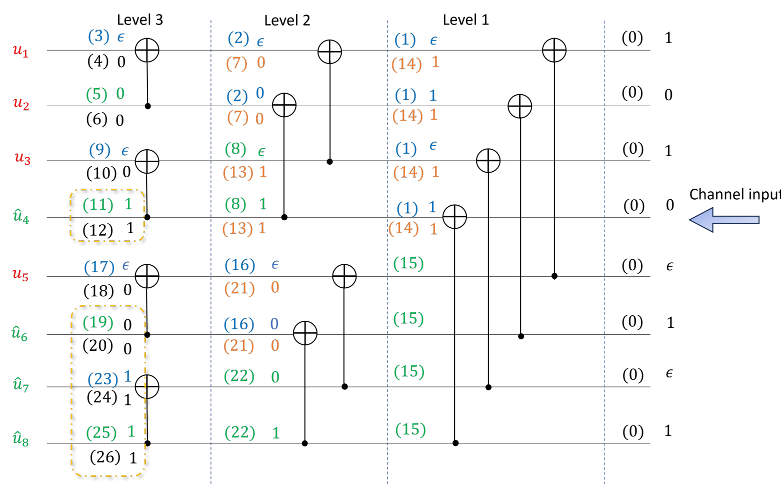


Fig. 6: Example of SC decoding process: An $N = 8$ polar code having $k = 4$, *frozen bit* indices $\mathcal{A} = 1, 2, 3, 5$, and the *frozen bits* $\mathcal{A}_\rfloor = (0000)$. A similar image showing the decoding process using *LLRs* can be found in [9] Fig. 15. View this figure jointly with Fig. 5, we can see that Op. (*) maps to $f(l_1, l_2) \triangleq XOR$ in *Step 1*; **Op. (*)** maps to the hard decision on $\hat{u}_1$; Op. (*) matches $g(l_1, l_2, \hat{u}_1)$; Op. (*) corresponds to *Step 3*. The final decoded output is the circled block of information bits.

Algorithm 3: f function

function FOP
Input: L_1, L_2
Output: f_{LLR}

```
for i = 1 : Length(L_1) do
    if L_{1,i} is ε or L_{2,i} is ε then
        f_{LLR,i} ← ε
    else
        f_{LLR,i} ← XOR(L_{1,i}, L_{2,i})
```

Algorithm 4: g function

function GOP
Input: L_1, L_2
Output: g_{LLR}
for $i = 1$: Length(L_1) **do**
  if $L_{1,i}$ is NOT ϵ **then**
    $g_{LLR,i} \leftarrow L_{1,i}$
  else if $L_{2,i}$ is NOT ϵ **then**
    $g_{LLR,i} \leftarrow L_{2,i}$
  else
    $g_{LLR,i} \leftarrow \epsilon$

C. Complexity Analysis

The memory complexity of this implementation is $\mathcal{O}(N \log N)$. The decoding procedure follows Fig. 5. We first compute left child, then right child and merge the result again at the parent node. f and g functions are described in *Algorithm 3* and *4* respectively. A step-by-step example of the decoding process is shown in Fig. 6.

V. Conclusion

Polar code is beautiful given its natural origin purely in channel capacity and mutual information Shannon's definition. Its recurisve encoding and decoding structures would also worth diving in depth into its hardware implementation. Extensive researches have been conducted on exploring more spatial and run-time efficient polar decoder. Some future work can be related to implementing the one exploiting *rate-one* node [10], list-decoding and list-decoding assisted with CRC [6].

References

[1] E. Arikan, "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.

[2] V. Bioglio, C. Condo, and I. Land, "Design of polar codes in 5g new radio," *IEEE Communications Surveys Tutorials*, vol. 23, no. 1, pp. 29–40, 2021.

[3] E. Arikan, "Polar coding tutorial." https://simons.berkeley.edu/sites/default/files/docs/2689/slidesarikan.pdf, 2015. Accessed: 2023–12-04.

[4] K. El-Abbasy, R. T. ElDin, S. Elramly, and B. Abdelhamid, "Polar codes bhattacharyya parameter generalisation," *IET Communications*, vol. 14, no. 11, pp. 1718–1729, 2020.

[5] A. Wyner, "A definition of conditional mutual information for arbitrary ensembles," *Information and Control*, vol. 38, no. 1, pp. 51–59, 1978.

[6] I. Tal and A. Vardy, "List decoding of polar codes," 2012.

[7] S. Kasi, J. Kaewell, S. Hamidi-Rad, and K. Jamieson, "Decoding polar codes via noisy quantum gates: Quantum circuits and insights," *ArXiv*, vol. abs/2210.10854, 2022.

[8] H. D. Pfister, "A brief introduction to polar codes notes for introduction to error-correcting codes," 2017.

[9] Z. Babar, Z. B. Kaykac Egilmez, L. Xiang, D. Chandra, R. G. Maunder, S. X. Ng, and L. Hanzo, "Polar codes and their quantum-domain counterparts," *IEEE Communications Surveys Tutorials*, vol. 22, no. 1, pp. 123–155, 2020.

[10] 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.