SCIENCE CHINA
Information Sciences

• Supplementary File •

Efficient Polar Coding Scheme and Implementation with Shared Information Bits

Wenyue ZHOU[1,2,3], Xiaofeng ZHOU[1], Yifei SHEN[1,2], Qiang LIU[1],
Liping LI[3*] & Chuan ZHANG[1,2*]

[1] *Laboratory of Efficient Architectures for Digital-communication and Signal-processing (LEADS), National Mobile Communications Research Laboratory, Southeast University, Nanjing 211100, China;*
[2] *Purple Mountain Laboratories, Nanjing 211111, China;*
[3] *Key Laboratory of Intelligent Computing and Signal Processing of the Ministry of Education of China, Anhui University, Hefei 230031, China*

Appendix A Preliminaries of Polar Codes

Appendix A.1 Polar Codes

Denote v_1^N as a vector $(v_1, ..., v_N)$ of length N. Let $W: \mathcal{X} \to \mathcal{Y}$ denote a symmetric B-DMC, with the input alphabet $\mathcal{X} = \{0, 1\}$, the output alphabet $\mathcal{Y}$, and the channel transition probability $W(y|x)$, $x \in \mathcal{X}$, $y \in \mathcal{Y}$. Let $N = 2^n$ $(n \geqslant 1)$ denote the block code length. The generator matrix of polar codes is G, which is given by $G = B_N F^{\otimes n}$. Here B_N denotes the bit-reversal permutation matrix, $F = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$, and $F^{\otimes n}$ represents the n-th Kronecker power of F over the binary field $\mathbb{F}_2$. The codewords x_1^N can be obtained by $x_1^N = u_1^N G$, where u_1^N is the source vector, consisting of K information bits and $N - K$ frozen bits (the fixed information in the source vector). The codeword x_1^N is transmitted over N independent copies of W, written as W^N, with a transition probability $W^N(y_1^N|x_1^N)$.

Channel polarization process has two parts: channel combining and channel splitting. Channel combining is a phase that combines copies of W in a recursive manner to produce a vector channel W_N, with $W_N(y_1^N|u_1^N) = W^N(y_1^N|x_1^N)$. Channel splitting is an operation splitting W_N back into a set of N binary-input channels $W_N^{(i)}$, $i \in \{1, 2, ..., N\}$. The i-th such channel is called bit channel i (meaning the channel that bit i virtually experiences). According to [1], $I(W_N^{(i)})$ (the capacity of bit channel i) converges to either 0 or 1 as N tends to infinity, and the fraction of the bit channels with capacity 1 approaches $I(W)$.

With finite block lengths, not all bit channels are fully polarized. The principle of polar codes is to choose the K most reliable bit channels among N bit channels to convey information bits. The other bits are called frozen bits which are fixed to be transmitted on the remaining channels. The good information set is denoted as $\mathcal{A}$ and complementary set is $\mathcal{A}_c$. Denote $u_{\mathcal{A}}$ as a subvector of the vector u_1^N that takes elements of it from the set $\mathcal{A}$.

The SC decoding was proposed in [1] and it recursively computes the log-likelihood ratio (LLR) of bit $\hat{u}_i$ by

$$\lambda^{(i)} \triangleq \ln \frac{P(y_1^N, \hat{u}_1^{i-1}|u_i = 0)}{P(y_1^N, \hat{u}_1^{i-1}|u_i = 1)}, \tag{A1}$$

where $\hat{u}_1^{i-1}$ denotes the previous decisions of all the $i-1$ bits. The SC decoder generates the estimate $\hat{u}_i$ of bit u_i $(i \in \mathcal{A})$ from

$$\hat{u}_i = \begin{cases} 0, \text{ if } \lambda^{(i)} \geqslant 0 \\ 1, \text{ otherwise.} \end{cases} \tag{A2}$$

For the frozen bit $i \in \mathcal{A}_c$, the decision $\hat{u}_i$ is simply set to the known frozen value u_i. The decoding complexity of SC is $\mathcal{O}(N \log N)$ [1].

Appendix A.2 Successive Cancellation List Decoding

The SCL decoding was proposed in [2–4]. It can search the polar codes decoding tree, enabling tracking L best paths concurrently. Each path contains a path metric (PM) value, which is a reliability measure for this path. For each bit to be estimated, the SCL decoder will fork the existing L paths according to the bit estimation hypothesises, and yield $2L$ paths. Then the PM values with respect to each path will be updated, and only L paths with the smallest PM values will be reserved as the current path lists. In the final decision stage, the path with the smallest PM value is determined as the decoder output. The PM is initialized as 0, and it can be updated by

$$\mathrm{PM}_l^i = \begin{cases} \mathrm{PM}_l^{i-1}, & \text{if } \hat{u}_i = \frac{1-\mathrm{sign}(\lambda^{(i)})}{2} \\ \mathrm{PM}_l^{i-1} + |\lambda^{(i)}|, & \text{otherwise.} \end{cases} \tag{A3}$$

[*] Corresponding author (email: liping_li@ahu.edu.cn, chzhang@seu.edu.cn)

where l is the path index $(0 < l \leqslant L)$ and i is index of the estimated bit.

The CRC aided SCL (CA-SCL) decoding was also proposed in [4]. In the encoding process of polar codes, a few CRC bits can be added to the information bits. At the final decision stage, the CA-SCL decoder will sequentially perform CRC detection for the L remaining paths according to the PM values. Once a path passing the CRC detection, the decoder will output the decisions of this path and the decoding is accomplished. If all of the paths are incorrect for the CRC detection, the decoder will output the estimated bits of the path with the smallest PM value.

Appendix A.3 Belief Propagation Decoding

The message passing of BP decoders for polar codes can be represented by factor graphs [5–7] based on processing elements (PEs). The LLR messages of the BP decoding can be divided into two types: the $\mathcal{L}$ messages flowing from the right side to the left side, and the $\mathcal{R}$ messages flowing in an inverse direction. BP decoder iteratively updates the $\mathcal{L}$ messages and the $\mathcal{R}$ messages by the following equations [7]:

$$\begin{cases} \mathcal{L}_{i,j} = f(\mathcal{L}_{i,j+1}, \mathcal{R}_{i+n/2^j,j} + \mathcal{L}_{i+n/2^j,j+1}), \\ \mathcal{L}_{i+n/2^j,j} = f(\mathcal{L}_{i,j+1}, \mathcal{R}_{i,j}) + \mathcal{L}_{i+n/2^j,j+1}, \\ \mathcal{R}_{i,j+1} = f(\mathcal{R}_{i,j}, \mathcal{R}_{i+n/2^j,j} + \mathcal{L}_{i+n/2^j,j+1}), \\ \mathcal{R}_{i+n/2^j,j+1} = f(\mathcal{R}_{i,j}, \mathcal{L}_{i,j+1}) + \mathcal{R}_{i+n/2^j,j}, \end{cases} \tag{A4}$$

where $f(a,b) = \lambda \times \text{sign}(a)\text{sign}(b)\text{min}(|a|,|b|)$, λ is the scale factor, $\mathcal{L}_{i,j}$ denotes the $\mathcal{L}$ messages of the i-th PE in stage j. The $\mathcal{L}$ messages in the right-most stage are the received LLRs, and the $\mathcal{R}$ messages in the left-most stage are set to ∞ for frozen bits and 0 for information bits. The iteration will be terminated when the maximum number of iteration $I_{\max}$ is achieved, then the estimated bits are decided by the following formula:

$$\hat{u}_i = \begin{cases} 0 \text{ if } \mathcal{L}_{i,1} \geqslant 0, \\ 1 \text{ if } \mathcal{L}_{i,1} < 0. \end{cases} \tag{A5}$$

Appendix B Supplementary Information of PCM Scheme with Two Underlying Blocks

In this section, the details of the decoding strategies, the principle for selecting the SIBs, the error performance analysis and the complexity analysis for the PCM scheme with two underlying blocks ($m = 2$) are discussed.

Appendix B.1 Decoding Strategy

The symbols of encoded code blocks are transmitted over a channel, and the noisy version of them are observed at the receiver side. The receiver first performs the SC decoding for both Block Odd and Block Even, and generates estimation bits for each block. Then the estimation bits will pass the CRC detection module, and detection results of each block will be returned. The possible detection results are:

- Case 1: Both Block Odd and Block Even are decoded correctly;
- Case 2: Block Odd is decoded correctly but Block Even is decoded incorrectly;
- Case 3: Block Odd is decoded incorrectly while Block Even is decoded correctly;
- Case 4: Both Blocks are decoded incorrectly.

For Case 1 and Case 2, since Block Odd is decoded correctly, the K_p estimations of the SIBs are stored in the memory for a possible application. For Case 1, since Block Even is also decoded successfully, there is no need for any more actions. For Case 2, Block Even is decoded incorrectly, a new-round SC decoding for Block Even can be carried out. For Case 3 and Case 4, since Block Odd is decoded incorrectly, the initial N LLR values of this block will be stored in the memory. For Case 3, since Block Even is successful, the estimations of SIBs will be stored to the memory and a new-round decoding can be performed for Block Odd. For Case 4, since Block Even is also decoded incorrectly, there is nothing the decoder can do for both blocks.

A more detailed description of the new-round decoding is as follows. The estimations of the SIBs in the memory can be exploited as extrinsic information. Take Case 2 as an example. Here the decoder of Block Even can reuse the computation of the SC decoding up to the first bit in $\mathcal{B}$. When it reaches to the first bit with the index $i \in \mathcal{B}$, then the decoder takes this bit as a frozen bit: no matter what the calculated LLR value is for u_i, it is assigned to the decision taken from the memory. The computation of the SC decoding goes on until the end, treating all of the bits in $\mathcal{B}$ as frozen bits. The new-round decoding for Block Odd in Case 3 is the same as that in Case 2.

The SC decoder can be perfectly replaced by the BP or CA-SCL decoder in the proposed PCM scheme. Here note that for the BP decoding, the best way to exploit the correct decisions of the SIBs in the new-round decoding is to treat them as frozen bits, instead of directly using their soft values. The reason is simple: by treating them as frozen bits, the initial LLR values of these bits are equivalently set to be infinity, which is definitely better than using finite soft LLR values. As for the PCM scheme with the CA-SCL decoder, the SIBs are treated as frozen bits directly. Therefore, even with the BP or CA-SCL decoder, the SIBs are utilized in the same way as the SC decoder in the PCM scheme.

Appendix B.2 Principles for Choosing Appropriate SIBs

The performance of the PCM scheme is directly effected by where to position the SIBs and how many SIBs are selected. Therefore, we will discuss the principles for choosing appropriate SIBs from the following two aspects.

Appendix B.2.1 *Principle for Choosing the Position of SIBs*

The positioning of SIBs is to find an optimal way in assigning them to the input of the two blocks (Block Odd and Block Even). Here "optimal" means the best system error performance. The exact formula is derived as follows. Let $u_\mathcal{B}$ represent the subvector which is comprised of the SIBs. And the size of set $\mathcal{B}$ is $|\mathcal{B}| = K_\text{p}$. Theoretically, there are $C_K^{K_\text{p}}$ ways to choose the set $\mathcal{B}$, but empirically speaking, the set $\mathcal{B}$ with respect to the worst channel quality will bring the best performance. The good performance of the PCM scheme relies on the rectification of the failed block. Only when the SIBs are with the worst channel reliability, the new-round decoding will exhibit the best rectification efficiency.

Now let us prove the empirical principle with theoretical derivation. We assume that the information set $\mathcal{A} = \{i_1, i_2, ..., i_K\}$ is organized in an ascending order with respect to the bit channel reliability. In other words, there exists the relationship of $P_e(W_N^{(i_1)}) \geqslant P_e(W_N^{(i_2)}) \geqslant ... \geqslant P_e(W_N^{(i_K)})$, where $P_e(W_N^{(i)})$ is the error probability of the i-th information bit. The following proposition states an optimal way to achieve the best union bound [8], which generates the best system performance.

Proposition 1. Supposing the information set $\mathcal{A} = \{i_1, i_2, ..., i_K\}$ is ordered in an ascending order with respect to the bit channel reliability, then the set $\mathcal{B}$ containing the first K_p elements of the set $\mathcal{A}$ as the SIBs indices can produce the minimum union bound.

Proof. Let $P_B(\mathcal{A})$ denote the error rate of the underlying blocks with the information set $\mathcal{A}$. Literature [8] indicates the union bound of $P_B(\mathcal{A})$ as

$$P_B(\mathcal{A}) \leqslant \sum_{i\in\mathcal{A}} P_e(W_N^{(i)}). \tag{B1}$$

Because the SIBs are utilized as the frozen bits in the new-round decoding, the underlying blocks enjoy a smaller information set represented as $\mathcal{A}' = \mathcal{A} \setminus \mathcal{B}$, thus yielding a better union bound:

$$P_B(\mathcal{A}') \leqslant \sum_{i\in\mathcal{A}'} P_e(W_N^{(i)}). \tag{B2}$$

Supposing set $\mathcal{B}'$ is any other type of SIB set, the information set of the underlying blocks in the new-round decoding becomes $\mathcal{A}'' = \mathcal{A} \setminus \mathcal{B}'$, leading to an union bound as

$$P_B(\mathcal{A}'') \leqslant \sum_{i\in\mathcal{A}''} P_e(W_N^{(i)}). \tag{B3}$$

Because set $\mathcal{B}$ contains the indices corresponding to the K_p largest error probabilities in $\mathcal{A}$, it is obvious that

$$\sum_{i\in\mathcal{B}} P_e(W_N^{(i)}) \geqslant \sum_{i\in\mathcal{B}'} P_e(W_N^{(i)}). \tag{B4}$$

Therefore we can obtain that

$$\sum_{i\in\mathcal{A}'} P_e(W_N^{(i)}) \leqslant \sum_{i\in\mathcal{A}''} P_e(W_N^{(i)}). \tag{B5}$$

It means that the union bound of $P_B(\mathcal{A}')$ is smaller than $P_B(\mathcal{A}'')$. Since $\mathcal{B}'$ is arbitrary, we can conclude that $P_B(\mathcal{A}')$ has the smallest union bound, thus yielding the best system performance.

Appendix B.2.2 *Principle for Choosing the Number of SIBs*

The performance of the PCM scheme is also affected by the number of the SIBs. In fact, a larger number of SIBs yields a better performance for the new-round decoding, but it also introduces a higher code rate loss for the PCM scheme. Therefore, trade-off should be considered in the PCM scheme. According to the simulation results, an empirically appropriate range of the number for the SIBs is $K_\mathrm{p} = 0.08NR \sim 0.16NR$.

Appendix B.3 Error Performance Analysis

In this section, we investigate the error performance of the proposed PCM scheme, then provide the FER formula of the PCM scheme. For simplicity, we omit the inside argument of $P_B(\mathcal{A})$, and use P_B to represent the FER of the underlying block with information set $\mathcal{A}$. The FER of the PCM scheme consists of two parts:

- Part 1: Block Odd and Block Even are both decoded incorrectly, corresponding to Case 4 in Section Appendix B.1.
- Part 2: The new-round decoding for Block Even (Case 2) or Block Odd (Case 3) fails.

For Part 1, the FER can be simply denoted as $P_1 = P_B^2$. For Part 2, the probability that there is only one underlying block decoded incorrectly is ${C_2}^1 P_B(1 - P_B)$. Since there is still one correct block, the FER of this part can be illustrated as

$$P_2 = \frac{1}{2} C_2^1 P_B(1 - P_B){P'}_B = P_B(1 - P_B){P'}_B, \tag{B6}$$

where ${P'}_B$ is the error probability of the new-round decoding. Together with P_1, the FER of the PCM scheme is:

$$P_\mathrm{new} = P_B^2 + P_B(1 - P_B)P'_B. \tag{B7}$$

With the optimal placement of the SIBs discussed in Section Appendix B.2, there must be some blocks which can be rectified in the new-round decoding. The error rate of the new-round decoding can be simplified with the following proposition.

Proposition 2. Let P_b denote the FER of the underlying block for polar codes with the information set $\mathcal{A}' = \mathcal{A}\setminus\mathcal{B} : P_b = P_B(\mathcal{A}')$, then P'_B can be derived as:

$$P'_B = \frac{P_b}{P_B}. \tag{B8}$$

Proof. Let ξ_1 denote the probability event that the received samples of an underlying block are decoded incorrectly with the information set $\mathcal{A}$, which has a probability of $P(\xi_1) = P_B$. Let ξ_2 denote the probability event that the received samples are decoded incorrectly with the information set $\mathcal{A}' = \mathcal{A} \setminus \mathcal{B}$, which corresponds to a probability of $P(\xi_2) = P_b$. The definition of P'_B indicates that it can be written as the following conditional probability:

$$P'_B = P(\xi_2|\xi_1) = \frac{P(\xi_1, \xi_2)}{P(\xi_1)}, \tag{B9}$$

where $P(\xi_1, \xi_2)$ denotes the joint probability of the event ξ_1 and ξ_2. According to the joint distribution of ξ_1 and ξ_2, we can obtain:

$$P(\xi_1, \xi_2) + P(\bar{\xi}_1, \xi_2) = P(\xi_2) = P_b, \tag{B10}$$

where $P(\bar{\xi}_1, \xi_2)$ denotes the joint probability that one of the received samples is decoded successfully with information set $\mathcal{A}$ but is decoded incorrectly with a smaller information set $\mathcal{A}'$, therefore we can obtain $P(\bar{\xi}_1, \xi_2) = 0$ and $P(\xi_1, \xi_2) = P_b$. The error probability P'_B can be illustrated as:

$$P'_B = \frac{P(\xi_1, \xi_2)}{P(\xi_1)} = \frac{P_b}{P_B}. \tag{B11}$$

According to Proposition 2, (B7) can be rewritten as:

$$P_{\text{new}} = P_B^2 + (1 - P_B)P_b. \tag{B12}$$

(B12) shows that the proposed PCM scheme can improve the performance of the conventional decoder for polar codes, because $P_b \leqslant P_B$ can always be guaranteed with the same noise condition.

Appendix B.4 Complexity Analysis

The memory consumption for the proposed PCM decoder is comprised of the memory for the conventional decoder and the memory for the new-round decoding. Taking the PCM-SC-2 decoder as an example. We assume that both the channel LLRs and the internal LLRs are quantized with Q bits, then the memory consumption of the conventional SC decoder can be formulated as [9]:

$$M_{\text{SC}} = (2N - 1)Q + N - 1 \qquad \text{[bits]}. \tag{B13}$$

The additional memory utilized for the new-round decoding can be illustrated by $NQ + K_{\text{p}}$ bits, in which NQ bits is used to store the channel LLRs for the failed blocks, and K_{p} bits is used to store the SIBs of the successful block. Therefore, the total memory consumption of the proposed PCM-SC-2 decoder is

$$M_{\text{PCM}} = (3N - 1)Q + N + K_{\text{p}} - 1 \qquad \text{[bits]}. \tag{B14}$$

In the decoding process, the PCM decoder does not constantly exchange soft information between two blocks. Instead, only when one block fails and the other succeeds, estimations of the SIBs are fed from the succeeded block to the failed block. The information pass can be considered as a sporadic procedure: the probability of performing the new-round decoding is only (denoted as P_a)

$$P_a = 2P_B(1 - P_B). \tag{B15}$$

The complexity of the conventional SC decoding is $\mathcal{O}(N \log N)$, therefore the complexity of the PCM-SC-2 decoding is $\mathcal{O}((1 + P_a)N \log N)$. The BP-decoding complexity is the same as the SC-decoding complexity, therefore, the decoding complexity of the PCM-BP-2 scheme is also $\mathcal{O}((1 + P_a)N \log N)$. Similarly, the SCL-decoding complexity is $\mathcal{O}(LN \log N)$, therefore the decoding complexity of the PCM-SCL-2 scheme is $\mathcal{O}((1+P_a)LN \log N)$. Note that this is a negligible increment of complexity, because P_a is usually a very small number in the realistic systems. With the negligible additional complexity, the PCM scheme can significantly improve the performance of the conventional decoder for polar codes as shown in (B12).

Appendix C PCM Scheme with Multiple Underlying Blocks

Appendix C.1 Direct Extended Version of PCM Scheme

The FER of the PCM scheme in Section Appendix B.3 is $P_B^2 + P_B(1-P_B)P'_B$. If we directly extend this scheme from two underlying blocks to m ($m > 2$) underlying blocks, in which each block contains the same K_{p} SIBs, the FER of the D-PCM scheme will consist of the following parts:

- Part 1: Only one block is decoded incorrectly in the first-round decoding, and it fails again in the new-round decoding;
- Part 2: Two blocks are decoded incorrectly in the first-round decoding, and at least one block fails again in the new-round decoding;
- ...
- Part m: All of the m blocks are decoded incorrectly in the first-round decoding.

The FER of Part 1 can be written as the following formula derived from (B6):

$$P_1 = \frac{1}{m} C_m^1 P_B (1 - P_B)^{m-1} P'_B. \tag{C1}$$

For Part 2, since there are two blocks decoded incorrectly in the first-round decoding, the erroneous results of the new-round decoding are comprised of the following cases: 1) only one of blocks is still decoded incorrectly, and 2) both the two blocks are decoded incorrectly. The FER of Part 2 can be illustrated by:

$$P_2 = \frac{2}{m} C_m^2 P_B^2 (1 - P_B)^{m-2} (\frac{1}{2} C_2^1 P'_B (1 - P'_B) + P_B^{\prime 2}). \tag{C2}$$

From (C1) and (C2) we can derive that the FER of the Part k ($1 \leqslant k \leqslant m$) is:

$$P_k = \frac{k}{m} C_m^k P_B^k (1 - P_B)^{m-k} (\frac{1}{k} C_k^1 P'_B (1 - P'_B)^{k-1} + \frac{2}{k} C_k^2 P_B^{\prime 2} (1 - P'_B)^{k-2} + ... + P_B^{\prime k}). \tag{C3}$$

For Part m, because all of the blocks are decoded incorrectly in the first round, the FER can be simply denoted as $P_m = P_B^m$. Accumulating the components of each FER part and simplifying the formula, the FER of D-PCM scheme with m underlying blocks is:

$$P_{\text{new}} = \sum_{k=1}^{m} P_k = P_B (1 - P_B)^{m-1} P'_B + \frac{2}{m} C_m^2 P_B^2 (1 - P_B)^{m-2} P'_B + ... + P_B^m. \tag{C4}$$

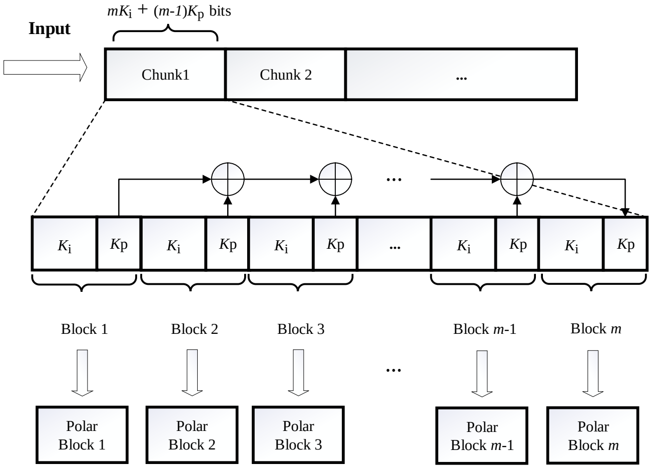

Figure C1 Configuration of the input bit stream for the proposed G-PCM scheme.

Replacing P'_B by P_b/P_B in (C4), we can obtain:

$$P_\mathrm{new} = (1-P_B)^{m-1}P_b + \frac{2}{m}C_m^2 P_B(1-P_B)^{m-2}P_b + \ldots + P_B^m, \tag{C5}$$

where the $(1-P_B)^{m-1}P_b$ denotes the FER for the Part 1, associated with case that only one underlying block is decoded incorrectly and the other $m-1$ underlying block are decoded correctly. It is observed that all of the FER for the other Parts have an order beyond $P_B \times P_b$. Therefore, the system FER performance of the D-PCM scheme is dominated by the FER of the Part 1. Based on this fact, we propose a G-PCM scheme with a high code rate efficiency, in which the new-round decoding only occurs in the case with respect to Part 1. The details of the G-PCM scheme will be discussed in the next section.

Appendix C.2 Generalized Extended Version of PCM

The code rate of the D-PCM scheme can be illustrated by:

$$R = \frac{m(K-K_\mathrm{crc})-(m-1)K_\mathrm{p}}{mN}, \tag{C6}$$

which suffers from a rate loss of $\frac{(m-1)K_\mathrm{p}}{mN}$ caused by the SIBs. The FER analysis of the D-PCM scheme indicates that we can design an encoding scheme with a more efficient code rate for the extended PCM. Figure C1 shows the configuration of the input bit stream for the proposed G-PCM scheme. Each chunk in Figure C1 contains $mK_\mathrm{i}+(m-1)K_\mathrm{p}$ bits. All of the preceding $m-1$ blocks are filled with $K_\mathrm{i}+K_\mathrm{p}$ information bits, while the last block is only comprised of the remaining K_i bits. Then, the SIBs of the preceding $m-1$ blocks are taken and added together in GF(2), and the resultant K_p bits are put as the SIBs for the last block. Therefore, the configuration set of the SIBs in the proposed G-PCM scheme can be written as:

$$u_\mathcal{B}^m = u_\mathcal{B}^1 \oplus u_\mathcal{B}^2 \oplus \ldots \oplus u_\mathcal{B}^{m-1}, \tag{C7}$$

where $u_\mathcal{B}^k$, $k \in (1,2,\ldots,m)$ denotes the K_p SIBs of block k. The arrangement of the SIBs in each block also follows Proposition 1, in order to get the best system performance.

In this way, the code rate of the G-PCM scheme is:

$$R = \frac{m(K-K_\mathrm{crc})-K_\mathrm{p}}{mN}. \tag{C8}$$

The rate loss caused by the SIBs for the G-PCM scheme is K_p/mN. Given a constant K_p, this rate loss will become negligible with a large block number m. However, a larger block number comes with a higher decoding complexity. Trade-off can always be made between a smaller rate loss and a lower decoding latency.

After the first-round decoding, if all other $m-1$ blocks are decoded successfully, the SIBs of the failed block can be recovered by:

$$u_\mathcal{B}^k = \sum_{i=1, i\neq k}^{m} u_\mathcal{B}^i. \tag{C9}$$

The FER of the G-PCM scheme can be derived from (C5), and can be written as:

$$P_\mathrm{new} = (1-P_B)^{m-1}P_b + \frac{2}{m}C_m^2 P_B^2(1-P_B)^{m-2} + \ldots + \frac{k}{m}C_m^k P_B^k(1-P_B)^{m-k} + \ldots + P_B^m. \tag{C10}$$

It seems that the G-PCM scheme suffers from a performance degradation compared with the D-PCM scheme. However, it is noted that (C10) enjoys a better P_B than (C5) given the same code rate, because the underlying blocks of the G-PCM scheme are always with a smaller information set compared with the D-PCM scheme.

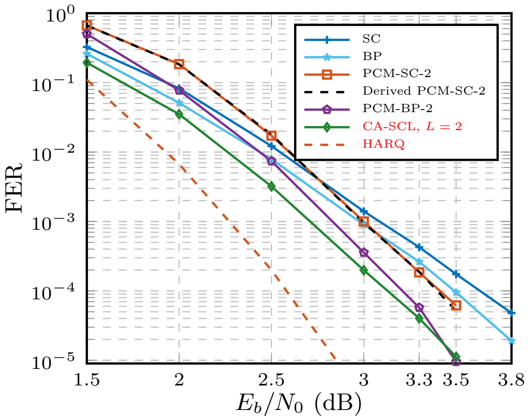

Figure D1 FER performance of the PCM-SC-2 scheme and the PCM-BP-2 scheme with $N = 1024$, $K = 565$, $K_{\mathrm{p}} = 74$, and $K_{\mathrm{crc}} = 16$. The overall code rate is $R = 0.5$. The HARQ scheme employs the SC decoder and the maximum number of the re-transmission is set as $\kappa = 1$.

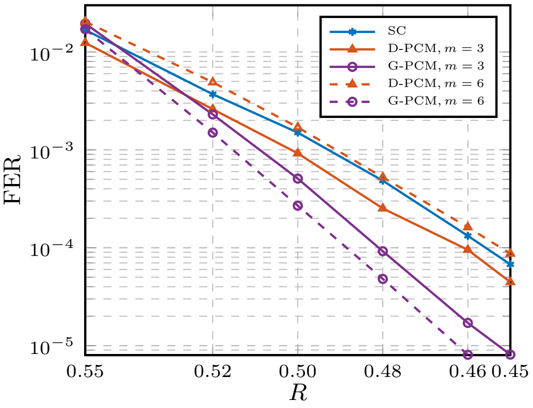

Figure D2 FER comparison results of the proposed G-PCM scheme and the D-PCM scheme with SC decoder at different code rate. For fair comparison, all of the curves are with $N = 1024$ and $K_{\mathrm{crc}} = 16$. $E_s/N_0 = 0$ dB is used to guarantee the same noise condition for all of the experiments.

Appendix D Numerical Results

In this section, we provide numerical results to validate the proposed PCM scheme. Additive white gaussian noise (AWGN) channel and binary phase-shift keying (BPSK) modulation are utilized for all of the simulations. The bit channel reliability sequence is generated according to [8] with a designed signal-to-noise ratio of 2 dB. A polynomial $g(x) = x^{16} + x^{12} + x^{5} + 1$ in the 5G standard is used for the CRC modules. The maximum iteration number of the BP decoder is set as $I_{\max} = 60$. The SIBs of the PCM schemes are chosen with the principles discussed in Section Appendix B.2.

Appendix D.1 Performance of the PCM Scheme with Two Underlying Blocks

Figure D1 reports the FER performance of PCM-SC-2 scheme and PCM-BP-2 scheme with $N = 1024$, $K = 565$, $K_{\mathrm{p}} = 74$. The overall code rate is $R = 0.5$. The conventional CA-SCL decoder is with the list size of $L = 2$. In addition, we offer the FER performance of the HARQ scheme for polar codes with the SC decoder. For fair comparison, the maximum number of the re-transmission is set as $\kappa = 1$, because the new-round of decoding for the failed underlying block is only performed once in the PCM scheme. It can be observed that the PCM-SC-2 scheme outperforms the conventional SC decoder by about 0.25 dB at a target FER of 10^{-4}. It is also observed that the HARQ scheme achieves great performance gain compared with the proposed PCM-SC-2 scheme. However, this performance gain is achieved at an expense of a higher decoding latency. The PCM-BP-2 scheme outperforms the conventional BP decoder by about 0.3 dB at the same target FER, and it achieves a comparable performance as the CA-SCL decoder with list size of $L = 2$ with $E_b/N_0 \geqslant 3.3$ dB. Figure D1 also shows the derived FER of PCM-SC-2 scheme from (B12). It can be seen that the experimental results are consistent with the derived results, demonstrating the performance analysis in Section Appendix B.3.

Appendix D.2 Comparison Results of the PCM Scheme with Multiple Underlying Blocks

Figure D2 shows the FER performance comparison results of the D-PCM scheme and the G-PCM scheme at different code rate with $N = 1024$ and $K_{\mathrm{crc}} = 16$. In order to demonstrate the influence of the rate loss to the FER performance, the signal-to-noise ratio with $E_s/N_0 = 0$ dB is used to guarantee the same noise condition. Several observations can be drawn from the Figure. First, with the same number of the underlying blocks, the G-PCM scheme outperforms the D-PCM scheme since it brings down the code rate loss. Second, with a larger number of the underlying blocks, the G-PCM scheme exhibits a better FER performance, which

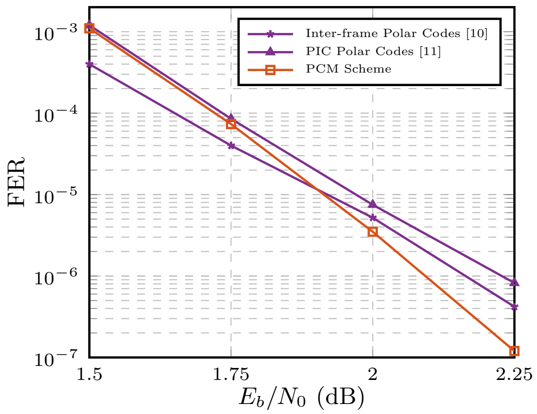

Figure D3 FER comparison results of the PCM-SCL-2 scheme, the inter-frame polar coding scheme [10], and the PIC scheme [11]. The code length is $N = 2048$ and the code rate is $R = 0.5$. All of the curves employ the same CA-SCL decoder with list of $L = 32$.

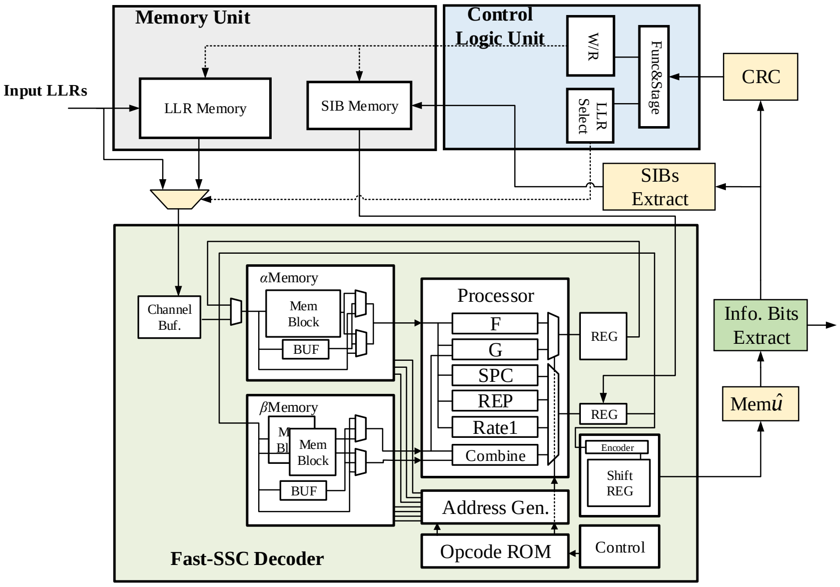

Figure E1 Overall architecture of the proposed PCM-FastSSC-2 decoder.

benefits from the further rate loss degradation. The comparison results are consistent with the theoretical analysis of the PCM scheme with multiple underlying blocks.

Appendix D.3 Comparison Results with the Related Works

The comparison results of the proposed PCM-SCL-2 scheme, the inter-frame polar coding scheme [10] and the PIC scheme [11] are shown in Figure D3. The code length is $N = 2048$, and all of the curves share the same parameters with code rate $R = 0.5$ and list size $L = 32$. It can be seen that the PCM-SCL-2 scheme outperforms the PIC scheme. In addition, for small E_b/N_0 (less than 1.8 dB) regions, the inter-frame polar coding scheme outperforms the PCM-SCL-2 scheme. However, for relatively large E_b/N_0 regions, the PCM-SCL-2 scheme enjoys a better performance than the inter-frame polar coding scheme.

Appendix E Hardware Architecture and Implementation Results of PCM Decoder

In this section, we present an overall architecture and time schedule of the PCM scheme with two underlying blocks. In order to get a high hardware efficiency, we exploit the Fast-SSC decoder [12] in our PCM scheme, therefore we call it PCM-FastSSC-2 decoder in this paper. We also provide comparison results of the proposed PCM-FastSSC-2 decoder to validate our hardware architecture.

Appendix E.1 Overall Architecture

The overall architecture of the proposed PCM-FastSSC-2 decoder is shown in Figure E1. The architecture is comprised of three parts: 1) a memory unit, 2) a control logic unit, and 3) a Fast-SSC decoder. The memory unit is utilized to store the LLR values and the estimations of the SIBs in the first-round decoding. The control logic unit is used to decide whether the LLR values or the SIBs should be reserved into the memory unit according to the results of the CRC detection. In addition, it also determines which LLRs should be chosen as the output of the MUX. The Fast-SSC decoder has the same architecture as that in [12], which is utilized to perform high-efficiency SC decoding for the underlying blocks. The detailed designs are discussed in the following sections.

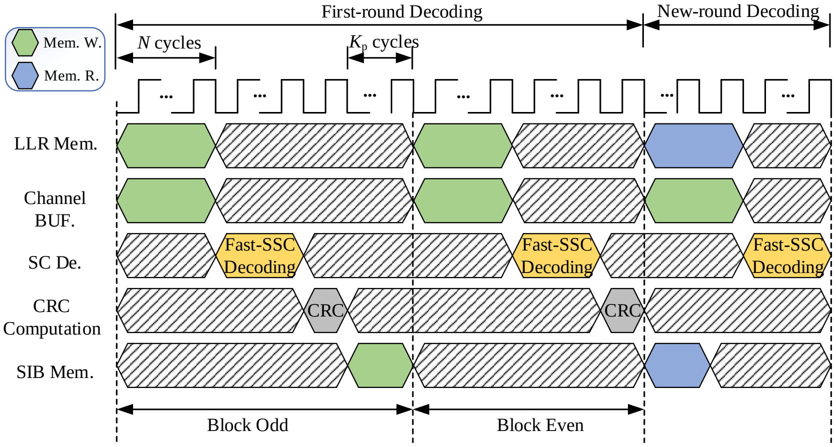

Figure E2 Latency analysis of the PCM-FastSSC-2 decoder with the Case 2 in Section Appendix B.1.

Appendix E.2 Control Logic Unit

The control logic unit determines the write or read (W/R) operations of the memory unit. It also decides which LLRs will be chosen as the output from the MUX. The detailed activities of the control logic unit are listed as follows:

1) W/R control of the LLR memory: When the decoding for each pair of PCM blocks begins, the input channel LLRs of Block Odd are stored into the LLR memory first. If Block Odd is decoded successfully in the first-round decoding, the channel LLRs of Block Even will be written into this memory. The reserved LLRs will be read as the channel LLRs when the new-round of decoding begins.

2) W/R control of the SIB memory: Once the estimation bits of Block Odd or Block Even pass the CRC detection in the first-round decoding, the correct estimated SIBs will be written to the SIB memory. In the new-round decoding, the SIBs in this memory will be read and transmitted into the register of the Fast-SSC decoder, and be exploited as frozen bits.

3) Output control of the MUX: In the first-round decoding, the MUX will choose the LLRs from channel as its output. However, in the new-round decoding, the LLRs read from the LLR memory will be selected as the output, in order to rectify the failed block.

Appendix E.3 Latency Analysis of the PCM-FastSSC-2 Decoder

The latency of the Fast-SSC decoder is the same as that in [12], which is denoted as τ clock cycles in this paper. Figure E2 shows the time schedule of the PCM-FastSSC-2 decoder with Case 2 in Section Appendix B.1. It is the worst case for the PCM-FastSSC-2 decoder in terms of decoding latency. In the first-round decoding of Block Odd, the input LLRs are written to the LLR memory and channel buffer simultaneously, which requires N clock cycles. Then the Fast-SSC decoding is performed, and it needs τ clock cycles. The CRC detection also requires one clock cycle. Because the estimation bits of Block Odd pass the CRC detection in this case, the estimated SIBs will be written to the register of the Fast-SSC decoder, which requires K_{p} clock cycles. Therefore, the latency with respect to Block Odd in the first-round decoding is $N + \tau + K_{\mathrm{p}} + 1$. The latency with respect to Block Even in the first-round decoding can be illustrated as $N + \tau + 1$ clock cycles, without the clock cycles of reserving the SIBs. In the new-round decoding, the LLR memory and the SIB memory can be read simultaneously, leading to a decoding latency of $N + \tau$ clock cycles. Therefore, the total latency of the PCM-FastSSC-2 decoder in Case 2 is $3(N+\tau)+K_{\mathrm{p}}+2$ clock cycles. The PCM-FastSSC decoder in Case 3 shares the same decoding latency as Case 2. The decoding latency in Case 4 (both blocks are decoded incorrectly in the first-round decoding) is $2(N+\tau+1)$ clock cycles, and the decoding latency in Case 1 is $2(N+\tau+1)+K_{\mathrm{p}}$ clock cycles, where the additional K_{p} cycles is caused by the storage of the SIBs.

Appendix E.4 Implementation Results

The PCM-FastSSC-2 decoder is implemented in Verilog and synthesized using SMIC 65 nm CMOS technology. Table E1 shows the synthesis results for the PCM-FastSSC-2 decoder, the Fast-SCF decoder [12], the Fast-SSCF decoder [13], the 2b-rSCL decoders [14], and the Fast-SSCL decoder [15]. For fair comparison, we set the maximum number of additional iterations as $T_{\max} = 1$ for the Fast-SSCF decoder, since new-round decoding is only performed once for the proposed PCM-FastSSC-2 decoder in the worst case. The average latency and throughput for the proposed decoder are calculated at the E_b/N_0 with a target FER of 10^{-3}. The quantization parameters for internal and channel LLRs are both set to $Q = 6$ bits.

It can be seen from Table E1 that compared with the Fast-SCF decoder [12], the proposed PCM-FastSSC-2 decoder has the same area consumption and 1.3× less throughput, thus leading to a 0.77× less area efficiency. Here note that the Fast-SCF decoder is also implemented based on the Fast-SSC decoder. Although the proposed PCM-FastSSC-2 decoder and the Fast-SCF decoder have the same area consumption, they requires different units in addition to the Fast-SSC decoder. Apart from the Fast-SSC decoder, the PCM-FastSSC-2 decoder requires a memory unit and a control logic unit while the Fast-SSCF decoder needs a CRC computation unit and a datapath for sorting flipping indices based on the decision LLRs. It is also observed that the worst case (W. C.) throughput of the proposed PCM-FastSSC-2 decoder is slightly better than that of the Fast-SCF decoder. In the worst case, the proposed PCM-FastSSC-2 decoder requires 1.5× decoding latency (only one of the two underlying blocks performs new-round decoding), while the Fast-SSCF requires double decoding latency (performs an additional SC decoding). Therefore, the W. C. throughput of the PCM-FastSSC-2 decoder can be calculated by dividing 1.5 by its average code throughput, while the W. C. throughput of the Fast-SSCF decoder can be calculated by dividing 2 by its average code throughput. Compared with the Fast-SSCF decoder [13], although the proposed PCM-FastSSC-2 decoder is 1.3×larger in terms of the area consumption, its throughput is 5.6×better than the Fast-SSCF decoder, achieving 4.3×more area efficiency.

Table E1 also shows the comparisons of the proposed PCM-FastSSC-2 decoder and the state-of-the-art SCL decoder. It is observed that the proposed decoder requires 1.7× less area and provides 3.9× more throughput compared to the 2b-rSCL decoder

Table E1 Synthesis Results with 65 NM SMIC COMS Technology for the Proposed PCM-FastSSC-2 Decoder and the SCF Decoders, the SCL Decoders.

Decoders	**This work**	**Fast-SCF** [12] [TCASI'20]	**Fast-SSCF** [13] [ISVLSI'19]	**2b-rSCL** [14] [VLSI'15]	**Fast-SSCL** [15] [TSP'17]
Technology [nm]	65	65	28	65	65
Parameters	–	$T_{\max} = 1$	$T_{\max} = 1$	$L = 2$	$L = 2$
$\mathbf{P_e}$	64	64	64	512	64
Area [mm^2]	**0.56**	0.56	**0.43***	0.97	1.05
Frequency [MHz]	333	430	303*	600	885
Latency† [μs]	0.88	0.68	–	3.41	0.55
W. C. Latency [μs]	1.32	1.36	–	3.41	0.55
Power [mW]	106.75	83.44	–	321	–
Power Density [mW/mm^2]	190.63	149.53	–	330.93	–
Code T/P† [Gbps]	**1.17**	**1.51**	0.21*	0.3	**1.86**
W. C. Code T/P [Gbps]	**0.78**	0.76	–	0.3	1.86
Area Efficiency [Gbps/mm^2]	**2.09**	**2.7**	0.49*	0.31	1.77

* Normalized for 65 nm CMOS technology, based on the scaling techniques from [16].

† Average value in target FER $= 10^{-3}$.

[14], respectively. Hence, the proposed decoder is up to 6.7× more area efficient. Compared to the Fast-SSCL decoder [15], the proposed decoder requires 1.9× less area consumption and yields 1.6× less throughput. As a result, the proposed decoder achieves 1.2× more area efficiency.

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 K. Niu and K. Chen. CRC-aided decoding of polar codes. IEEE Commun Lett, 2012, 16: 1668–1671

3 K. Chen, K. Niu, and J. Lin. Improved successive cancellation decoding of polar codes. IEEE Trans Commun, 2013, 61: 3100–3107

4 I. Tal and A. Vardy. List decoding of polar codes. IEEE Trans Inf Theory, 2015, 61: 2213–2226

5 E. Arıkan. A performance comparison of polar codes and reed-muller codes. IEEE Commun Lett, 2008, 12: 447–449

6 A. Eslami and H. Pishro-Nik. On bit error rate performance of polar codes in finite regime. In: Proceedings of IEEE Annual Allerton Conference on Communication, Control, and Computing, 2010. 188–194

7 B. Yuan and K. K. Parhi. Architecture optimizations for bp polar decoders. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, 2013. 2654–2658

8 I. Tal and A. Vardy. How to construct polar codes. IEEE Trans Inf Theory, 2013, 59: 6562–6582

9 S. A. Hashemi, C. Condo, F. Ercan, et al. Memory-efficient polar decoders. IEEE J Emerg Sel Topic Circuits Syst, 2017, 7: 604–615

10 H. Zheng, S. A. Hashemi, B. Chen, et al. Inter-frame polar coding with dynamic frozen bits. IEEE Commun Lett, 2019, 23: 1462–1465

11 X. Wu, L. Yang, Y. Xie, et al. Partially information coupled polar codes. IEEE Access, 2018, 6: 689–63 702

12 F. Ercan, T. Tonnellier, and W. J. Gross. Energy-efficient hardware architectures for fast polar decoders. IEEE Trans Circuits Syst I, Regul Papers, 2020, 67: 322–335

13 J. Zeng, Y. Zhou, J. Lin, et al. Hardware implementation of improved Fast-SSC-Flip decoder for polar codes. In: Proceedings of IEEE Computer Society Annual Symposium on VLSI, 2019. 580–585

14 B. Yuan and K. K. Parhi. Low-latency successive-cancellation list decoders for polar codes with multibit decision. IEEE Trans VLSI Syst, 2015, 23: 2268–2280

15 S. A. Hashemi, C. Condo, and W. J. Gross. Fast and flexible successive-cancellation list decoders for polar codes. IEEE Trans Signal Process, 2017, 65: 5756–5769

16 C. Wong and H. Chang. Reconfigurable turbo decoder with parallel architecture for 3GPP LTE system. IEEE Trans Circuits Syst II, Express Briefs, 2010,5: 566–570