**中等码长的极化码编译码方案研究与实现**

**摘要**

极化码以其优异的性能被当选为5G eMBB场景下控制信道的信道编码方案，从而又掀起了学术界的研究热潮。极化码最主要的优点是可以从数学理论的角度证明它是一种达到信道容量极限的编码方案。然而，在实际应用中，在中等码长条件下的极化码的性能会呈指数型衰减。因此提升极化码在中等码长条件下的译码性能一直是学术界研究的热点。受到串行抵消翻转（SCF）译码算法的启发，本文提出了附带记忆存储单元的极化码（Polar Codes with Memory，PCM）编译码方案。通过在两个连续的编码信息块中共享一定数量的位置相同的互信息比特（Mutual Information Bits），当接收端的某个码字译码错误时，就有可能从另一个码字的译码结果中获取正确的互信息比特，然后再译码一次。在新一轮的译码过程中，互信息比特会直接被当作冻结位信息处理，类似于对译码错误的码字中互信息比特的集体“翻转”，从而提升译码的性能。PCM编译码方案的底层译码器可以是SC译码器（PCM-SC），BP译码器（PCM-BP），或者SCL译码器（PCM-SCL）。仿真结果表明，PCM-SC在码长，码率，误组率PER条件下，相比于传统的SC译码算法的增益为0.27 dB。列表数为的PCM-SCL在同样条件下的PER性能可以达到传统的SCL译码器在列表数为时的PER性能。然后，本文又提出了基于多个编码信息块的PCM直接扩展方案和可以降低码率损失的一般型PCM扩展方案。最后，本文还设计了两种PCM译码器的硬件实现结构：顺序（In-serial, IS）结构译码器和低延时交织（Low-latency Interleaved）结构译码器。硬件仿真的综合结果表明，相比于IS结构的译码器和传统的SC译码器以及SCL译码器，LLI结构的译码器拥有最低的译码延时和最高的吞吐率。

**关键词：极化码；串行抵消译码；互信息比特；交织译码器；附带记忆存储功能的极化码**

**Abstract**

Polar codes have been selected as the channel coding scheme for controlling channel in 5G eMBB scenario due to the excellent performance, leading to a new round of research in academics. The major advantage of polar codes is that it can be mathematically proved to be a coding family which achieves the channel capacity. However, in practice, the performance of polar codes of the moderate code length will degrade exponentially. Therefore, it is always the hotspots of research that improving the performance of the polar codes of the moderate code length. In this paper, an innovative polar codes scheme, called polar codes with memory (PCM), is proposed, which is inspired by the successive cancellation flip (SCF) decoding algorithm. By containing a controlled number of mutual information bits in a pair of consecutive code blocks, the shared information bits of the succeeded block can help the failed block to recover in the decoding process. The underlying polar codes can employ any decoding scheme such as the successive cancellation (SC) decoding (PCM-SC), the belief propagation (BP) decoding (PCM-BP), and the successive cancellation list (SCL) decoding (PCM-SCL). Simulation results indicate that PCM-SC outperforms the stand-alone SC decoding by about 0.27 dB at PERfor the block length , code rate . The PER of PCM-SCL with list size of can match that of the stand-alone SCL decoding with list size of . In addition, this scheme can be directly extended to blocks (). Based on this, a general PCM is proposed in this paper, which reduces the rate loss while maintaining the same order of packet error rate (PER), comparing with the direct extension of PCM. Two hardware decoders for PCM are also implemented: the in-serial (IS) decoder and the low-latency interleaved (LLI) decoder. Synthesis results show that the LLI decoder has the lowest latency and the highest throughput, comped with the IS decoder, the conventional SC decoder and the SCL decoder.

**Keywords: polar codes; successive cancellation decoding; interleaved decoder; polar codes with memory**

**目录**

[第1章：绪论 4](#_Toc33718891)

[1.1 研究背景与意义 4](#_Toc33718892)

[1.2 极化码的发展历史与研究现状 6](#_Toc33718893)

[1.3 本文主要研究工作与结构安排 8](#_Toc33718894)

[第2章：极化码的原理概述 10](#_Toc33718895)

[2.1 B-DMC信道参数 10](#_Toc33718896)

[2.2 信道极化原理 11](#_Toc33718897)

[2.2.1 信道联合 11](#_Toc33718898)

[2.2.2 信道分裂 13](#_Toc33718899)

[2.2.3 信道极化 14](#_Toc33718900)

[2.3 极化码的编码 15](#_Toc33718901)

[2.4 常用译码算法 16](#_Toc33718902)

[2.4.1 SC译码算法 16](#_Toc33718903)

[2.4.2 SCL译码 17](#_Toc33718904)

[2.4.3 BP译码算法 19](#_Toc33718905)

[第3章：附带记忆存储单元的极化码系统设计 21](#_Toc33718906)

[3.1 系统模型与编译码原理 21](#_Toc33718907)

[3.1.1 PCM编码原理 21](#_Toc33718908)

[3.1.2 PCM译码策略 22](#_Toc33718909)

[3.2 系统的互信息比特位置选择准则 24](#_Toc33718910)

[3.3 系统的误码率分析 26](#_Toc33718911)

[3.4 基于BP或SCL译码器的PCM系统 28](#_Toc33718912)

[3.5 与Turbo码的比较 28](#_Toc33718913)

[3.6 仿真结果与分析 29](#_Toc33718914)

[第4章：基于多信息块编码的PCM系统 34](#_Toc33718915)

[4.1 直接型PCM扩展方案 34](#_Toc33718916)

[4.3 仿真结果分析 38](#_Toc33718917)

[第5章：附带记忆存储单元的极化码系统硬件实现 39](#_Toc33718918)

[5.1 基于顺序结构的PCM译码器 39](#_Toc33718919)

[5.2 基于低延迟交织译码结构的PCM译码器 40](#_Toc33718920)

[5.3 硬件仿真结果与分析 42](#_Toc33718921)

[第6章：总结与展望 44](#_Toc33718922)

[参考文献 46](#_Toc33718923)

# 第1章：绪论

## 1.1 研究背景与意义

1948年10月，Claude Shannon（克劳德.香农）在他的经典论文“A Mathematical Theory of Communication”（通信的数学理论）的引言部分中写道[1]：

“通信的基本问题就是在某一点精确或者近似地再生另一点选择的信息。”

为了解决这个问题，香农在论文中以数学证明的方式阐述了信息论的基本概念和方法，并且奠定了信息论的理论基础。信息论的基本原理显示，当发送端的信息速率与待传输信道的信道容量满足条件时，一定存在某种编码方案，可以实现信息的无差错传输；反之，当发送端以bit/s的速率发送二进制数据时，任何编码系统都无法实现无差错的传输。信道容量可以通过以下公式来计算：

公式（1-1）被称为香农公式，其中表示信道的带宽，为信号的平均功率，表示噪声平均功率，为信噪比。香农公式描述了在一个通信系统中，有限带宽，随机热噪声信道的最大传输速率与信道带宽，信噪比之间的关系，利用该公式可以计算出信道容量的极限。在实际传输系统中，寻找某种合适的信道编码方案，使得信道容量达到香农极限，一直以来都是通信领域的研究热点。

回顾信道编码的发展历史，可以看到通信学科的研究者们在不断创新的基础上，提出了一个又一个信道编码方案。1949年，R. Hamming在文章[2]中提出了实用性的差错控制编码方案——汉明码（Hamming Code）。随后，M. Golay针对汉明码的缺点，并且提出了改进的Golay码[3]。Golay码在1979~1981年间被用于美国国家航天局太空探测器Voyager与地面通信系统中的差错控制编码。随后，RM码[4;5]（Reed-muller Code）与循环冗余校验码[6]（CRC）也相继问世。RM码和CRC都属于分组码（Block Code），分组码虽然编译码简单，但是存在着两个主要缺点：1）在译码过程中，必须等待整个码字接收完毕，才能开始译码；2）需要非常精确的帧同步，因此导致译码的时延较大。此外，在1955年，Elias在文章[7]中提出了不同于分组码的另一类型的信道编码——卷积码（Convolutional Code）。卷积码与分组码的不同之处在于，它充分地利用了每个信息块的相关性。在卷积码的编码阶段，当前编码的码字不仅与当前信息块的信息比特有关，还与前若干个信息块中的信息比特相关。同时，在译码时，译码器不仅要从当前接收到的码字中提取译码信息，还需要利用前面时刻或以后接收到的码字，并从中提取相关译码信息。卷积码的译码是连续进行的，因此具有相对较小的译码延时，其缺点是计算的复杂度较高。在Elias工作的基础上，Viterbi提出了卷积码的Viterbi译码算法[8]，该译码算法很大程度上提升了卷积码的实用性，使得卷积码在通信系统中得到了广泛的应用，如GSM网络，IS-94 CDMA，卫星通信等。随后，法国工程师C. Berrou和A. Glavieux宣布他们发明了一种新的编码方法­——Turbo码[9;10]，并且该编码方案可以使得信道编码的效率接近香农极限。通过迭代译码的方式，Turbo码避开了计算复杂度的问题，并且获得了远超其他现有编码方案的性能，为信道编码的理论的研究方法带来了新的思路。但是，由于采用迭代译码的方式，Turbo码存在译码延时较高的问题。因此，在面临对实时性要求较高的场景时，Turbo码无法满足需求。

LDPC码（Low-density Parity-check Code）是一种在硬件实现复杂度和功耗方面均优于Turbo码的编码方案，它是由MIT的教授Robert Gallager在1962年提出的[11]。LDPC码是一种线性分组码，它基于高效的并行架构实现，并且译码性能可以逼近香农极限。此外，土耳其教授E. Arikan于2009年在文章[12]中提出了一种可以被理论证明达到香农极限的信道编码方案——极化码（Polar Codes）。极化码的编译码复杂度低，误码性能优于Turbo码，并且可以使用灵活的编码长度和方便的速率匹配方案。对比表明，极化码在很多方面的表现都优于Turbo码。由于LDPC码和极化码优异的性能，在面临5G应用场景下“超高速率，超低延时”的需求时，在3GPP规定的5G系列标准中，将LDPC码作为数据信道的编码方案，极化码作为5G广播和控制信道的编码方案。

极化码被选为5G通信系统中的信道编码方案，一方面表明它确实是一种综合性能优越的编码；另一方面也说明了对极化码的研究具有很高的实际意义。极化码在码长为无限长时，可以获得最佳的译码性能。但是在实际应用中，码长无限长的编码是不存在的，因此，极化码的实际应用场景一般为中等码长条件下，例如等。由于信道极化的不充分现象，中等码长条件下的极化码译码性能急剧恶化，无法满足应用的需求。因此，如何提升极化码在有限码长下的译码性能，一直以来都是国内外的研究热点。

## 1.2 极化码的发展历史与研究现状

上一小节已经提到，极化码是由土耳其教授E. Arikan于2009年在文章[12]中提出，并且已经被证明是一种能够在二进制离散无记忆信道（Binary-input Discrete Memoryless Channels, B-DMCs）条件下，达到信道容量极限的编码方案。Arikan还给出了一种极化码的译码算法——串行抵消（Successive Cancellation, SC）译码算法。SC译码算法具有很低的译码复杂度，仅为，但是其译码性能依赖于信道极化的程度。当极化码的编码长度受限时，就会出现信道极化不彻底的现象，从而导致SC译码算法的性能受到影响[13]。为了改善极化码在中等码长条件下的误码性能，文章[14;15;16]提出了串行抵消列表（Successive Cancellation List, SCL）译码算法。SCL算法是在SC算法的基础上，保留L条路径，每条路径对应一种译码结果，最后从这L条路径中挑选出一条最佳路径，作为译码的结果输出。文章[16]还提出了添加CRC校验的SCL译码算法（CA-SCL）。CA-SCL是对于SCL译码算法的改进，在编码阶段，在信息位中添加部分CRC校验比特，经过编码生成码字。然后在译码的最后阶段对L条路径的译码结果依次进行CRC校验，最先通过校验的路径即为CA-SCL译码算法的输出结果。CA-SCL译码算法的误码性能可以比肩（甚至优于）同等条件下的LDPC码的性能。文章[17;18;19;20]介绍了SC译码算法的硬件实现；文章[21;22;23;24]研究了SCL译码算法的硬件实现。通过对比SC译码算法与SCL译码算法的实现结果，可以看出SCL译码算法的优良误码性能是以相对较高的译码复杂度和较低的吞吐率为代价的。

文章[25;26]提出的置信传播（Belief Propagation, BP）译码是一种与SC译码不同类型的极化码译码算法。BP译码算法基于极化码的因子图，并且通过从接收端到发送端的反复迭代来得出最后的译码结果。BP 译码的性能优于SC译码，并且支持并行译码，但是其误码性能仍然低于SCL译码算法。

文章[27]提出的SCF（Successive Cancellation Flip, SCF）译码算法是基于SC译码算法的另一种改进算法。SCF译码算法的思路是，在SC译码算法出现错误时，找到其第一个错误估计的信息比特，并将其翻转（译码值取反），然后继续译码。它拥有比SCL更低的硬件实现复杂度，仿真结果表明，SCF能够达到列表数为2时的SCL译码算法的误码性能。文章[28]的作者研究了SC译码算法的第一个错误估计的比特的分布情况，并基于此提出了两种改进的SCF译码算法。此外，作者还研究了SC译码算法的错误传播规律，并且得出结论：如果能保证在SC译码过程中，最容易错误估计的信息比特的正确性，那么SC译码算法的性能将大幅度提升。

受SCF译码思路的影响，许多科研工作者开始研究如何在SC译码算法过程中，保证信道质量最差的信息位的正确译码，以提升SC译码算法的性能。文章[29;30]提出了基于动态冻结位（Dynamic Frozen Bits）的极化码编译码方案。所谓动态冻结位，即在极化码的信息位中，挑选出一部分（通常为信道质量较差的信息位部分），该部分信息位对应的信息比特与其他若干码字中的对应信息比特存在一定的相关性。因此，当相关码字译码正确时，可以直接计算出该部分信息比特，并在译码时将其视作冻结位；反之，当相关码字译码错误时，该部分信息比特被视作信息位，参与译码。通过相关编码，该部分信息位在译码阶段被动态地视作冻结位处理，因此称之为动态冻结位。文章[29]提出了部分信息位耦合（Partially Information Coupled, PIC）的极化码编译码方案。PIC方案采用了极化码的系统码编码，使得经过编码的码字，每个信息块的首尾分别与相邻的信息块共享一定数量的信息比特。在译码阶段，该部分共享的信息比特即作为动态冻结位处理。PIC方案还使用了删余（Puncture） 技术，以减小码率的损失。文章[30]提出了极化码的帧间辅助（Inter-frame Assisted, IFA）SCL译码算法（IFA-SCL）。在编码时，IFA的每个信息块中，一定数量最可靠的冻结位比特（Most Reliable Frozen Bits, MRFB）与前一个信息块中，相同数量最不可靠的非冻结比特（Most Unreliable Un-frozen Bits, MUUB）保持相同。在译码阶段，每个码字中的MRFB和MUUB都被视作动态冻结位处理。

PIC方案和IFA-SCL译码算法都可以在一定程度上带来极化码在中等码长条件下性能的提升，但是也存在以下缺点：1）编码结构复杂，不利于硬件实现；2）前后相关的编码结构使得在某些条件下的动态冻结位难以提前计算，因此影响了总体的误码性能；3）译码算法较为复杂，因此译码延时较高。

## 1.3 本文主要研究工作与结构安排

本文的主要研究工作为，在PIC和IFA方案的基础上，提出了一种新型的基于动态冻结位的极化码编译码方案。在该方案的编码阶段，一段固定长度的输入信息比特流被划分为两个小的信息比特集合，我们分别称为奇信息块（Block Odd）和偶信息块（Block Even）。然后将奇信息块中的一部分信息比特插入偶信息块，使得该部分信息比特在奇偶信息块中的位置相同，我们定义该部分信息比特为互信息比特（Mutual Information Bits）。上述奇偶信息块在经过极化的编码后，生成奇偶码字，再由信道传输至接收端进行译码。当奇偶码字的译码出现一个译码正确而另一个译码错误的结果时，译码错误的码字可以通过另一个正确码字的结果，获得正确的互信息比特，然后开始新一轮的译码。在第二轮译码中，这些互信息比特直接被当作冻结位处理，从而得到该码字正确的译码结果。由于互信息比特在可能出现的第二轮译码中，被视作冻结位，因此互信息比特就是该编码方案中的动态冻结位。

由于该方案的重点在于译码错误的码字可以获得存储在译码正确的码字中的互信息比特，并且开启新一轮的译码。所以该方案可以看作是相关的极化码码字之间存在着一些记忆存储功能，因此这种极化码的编译码方案被命名为附带记忆存储单元的极化码（Polar Codes with Memory, PCM）。除了上述两个信息块构成的编译码方案，PCM还适用于多个信息块之间共享互信息比特，并且形成动态冻结位的场景。基于此，本文还提出了两种PCM在多个信息块参与编码场景下的扩展方案：PCM的直接扩展方案和降低码率的PCM扩展方案。由于后一种扩展方案的码率损失更小，并且可以保持良好的误码性能，因此更具有实用性，普适性。所以，我们将降低码率的PCM扩展方案命名为一般型PCM方案（General PCM）。另外，我们还参考文章[19]，[31]的硬件结构，设计了两种PCM译码器的硬件实现结构：顺序（In-serial, IS）结构和低延时交织（Low-latency Interleaved, LLI）结构。我们还对上述PCM译码器的两种设计结构进行了硬件仿真，并且给出了硬件仿真的结果。

需要注意的是，PCM方案虽然也包含了动态冻结位，但是译码复杂度却远远低于PIC和IFA-SCL的译码复杂度。准确地说，PCM的译码复杂度约等于其底层译码器的复杂度，这个问题会在后续章节得到详细的讨论。PCM的底层译码器可以是SC译码器，BP译码器或者SCL译码器等常用的极化码译码器，为了方便描述，我们定义：包含个信息块，并且应用SC译码器的PCM方案为PCM-SC-；同理，PCM-BP-，PCM-SCL-分别表示包含个信息块的PCM方案，其底层译码器为BP，SCL译码器。

本文主要的贡献可以总结为以下三点：

1）提出了PCM编译码方案，系统地阐述了包含奇偶信息块的PCM系统模型，编译码策略，误码性能分析，并且通过仿真结果验证了PCM方案的性能。仿真结果表明，在PER级别下，PCM-SC-2的PER性能与列表数为2的SCL译码算法的性能相比，仅仅相差0.2 dB。PCM-BP-2可以达到列表数为2的SCL相同的性能。另外，PCM-SCL-2的在应用列表数为的SCL译码器时，其误码性能可以与列表数为的SCL的性能相匹配。

2）提出了两种包含多个信息块的PCM扩展方案，其中General PCM拥有更低的码率损失，更简单的编码结构。因此，若无特别说明，我们提到的多信息块PCM方案就是指General PCM扩展方案。

3）提出了两种PCM译码器的硬件设计结构，即顺序（In-serial, IS）结构和低延时交织（Low-latency Interleaved, LLI）结构。硬件实现的结果表明，在码长的条件下，与文章[19]中的SCL译码器相比，PCM-SC-2的LLI硬件结构具有较低的译码延迟以及更高的吞吐率。

文章剩余部分的结构安排如下：第2章为极化码的基本原理，编码方法以及常见的译码算法。第3章系统地阐述了包含两个信息块的PCM方案，并且给出了PCM的仿真结果。第4章提出了PCM的两种扩展方案。在第5章，我们设计了PCM的译码器的两种硬件结构，并且给出了硬件仿真结果。最后一章是总结与展望部分。

# 第2章：极化码的原理概述

## 2.1 B-DMC信道参数

在文献[12]中，Arikan用来表示一个一般类型的B-DMC，其中表示输入字符集，表示输出字符集，则信道转移概率可以表示为。输入字符集满足，输出字符集和转移概率可以是任意的。定义表示为对的次使用的信道，因此有：，并且转移概率为。

对于任意给定的B-DMC信道，存在两个信道参数：对称信道容量（Symmetric Capacity）

和巴氏参数（Bhattacharyya Parameter）

这两个参数可以用来衡量信道传输的信息速率和极化信道的可靠性。表示当信源等概率发送符号时的信道容量，也即在信息在对称信道中可靠传输所能达到的最大信息速率。是当信道仅传输二进制比特时，其采用最大似然（Maximum-likelihood, ML）译码算法的误码率上限。很显然，与的取值范围均在[0, 1]之间。事实上，对于一个B-DMC信道，其信道参数和满足下列关系：

根据公式（2-3），（2-4）可以得出，B-DMC的对称信道容量和巴氏参数的关系满足：当趋向于0时， 趋向于1；反之，当趋向于1时， 趋向于0。对于一个对称信道，当的值趋向于1时，说明该信道是一个完美的无噪声信道，适合传输有用的信息比特；反之，当趋向于0时，此时的信道是一个纯噪声信道，无法传输信息比特。

## 2.2 信道极化原理

为了便于说明，首先定义一些信道极化理论中常用的符号表示。定义表示一个长度为N的向量，用表示向量，即偶数下标，用表示向量，即奇数下标。在给定且集合的条件下，用来表示子向量。表示关于集合的补集。此外，表示集合中的元素个数。所有的编码译码计算过程都是在GF(2)范围内，也即二进制范围内，因此，符号表示模二加。如果无特殊说明，对数符号表示以2为底的对数。

信道极化是一个利用给定的B-DMC信道的个独立复制信道生成个极化信道的过程。信道极化带来的影响是，当非常大时，会趋于0或者1，信道因此也趋向于两个极端，完美的无噪声信道和纯噪声信道。信道极化过程可以被划分为信道联合（Channel Combining）阶段和信道分裂（Channel Splitting）阶段。

### 2.2.1 信道联合

信道联合指的是一个将给定的B-DMC信道的重独立复制信道以一种连续迭代的方式组合成一个信道向量，其中。信道联合开始于，此时只有信道本身，因此。当时，两个独立的结合生成，其转移概率为

图2.1展示了联合信道形成的过程。



图2.1：联合信道。

图2.2显示了接下来的信道联合迭代过程。两个独立的结合，形成了联合信道，其转移概率为。



图2.2：联合信道与信道和的关系。

图2.2中的是一个转置操作，将输入映射成。源信息比特经过联合信道，由信道输出码字，其映射关系可以表示为，其中。由此可得。



图2.3：联合信道的构造。

图2.3为信道联合的一般形式示意图，其中两个独立复制的信道联合生成了信道。信道的输入比特首先被转换成，并且对任意，均有，。图中所示的被称为反转变换操作，它将输入信号转换成，然后成为信道的输入信号。

在图中我们还可以看出，映射是一个在2进制数范围内的线性变换。同样可以推导出，联合信道的输入和底层信道的输入信号也满足线性变换关系，可以用矩阵来表示，即。被成为阶生成矩阵（Generator Matrix）。对于所有的，，信道到的转移概率为

生成矩阵满足关系式，其中是一个实现比特反转（Bit-reversal）的转置矩阵，。

### 2.2.2 信道分裂

在信道联合阶段，信道经过构造生成联合信道，信道极化的下一个阶段，信道分裂，即将生成的分裂，重新生成个并列的比特信道，。定义其转移概率为

其中表示信道的输出，表示其输入。表示当接收端的SC译码器在接收到后，前个信息比特译码正确的情况下，正在译码第个信息比特时的信道转移概率，该信道转移概率是一种抽象出来的数学上的概念。

公式（2-7）在计算码长较长时的分裂信道转移概率时，计算的复杂度很高。为了解决这个问题，需要用到迭代计算公式：

根据公式（2-8），信道转移概率可以一直迭代，直到出现，然后的概率可以通过合成基本信道的转移概率得到。经过这种线性迭代的方式，极化码原本呈指数型增长的计算复杂度转变为线性复杂度，具体的迭代公式如下：

公式（2-9）和（2-10）的证明可以参考文章[12]中附录部分定理3的证明。通过上述公式，我们就可以计算任意码长的极化码的信道转移概率，并且计算的复杂度是线性的。

### 2.2.3 信道极化

经过上述信道联合和信道分裂的两个小阶段，信道已经完成了极化过程，并且具备极化现象[12]：对于任意给定的B-DMC类型的信道，给定任意一个确定的值，当码长以形式趋向于无穷大时，极化信道，中，满足信道的信道容量会趋向于，满足信道的信道容量会趋向于。

可以通过下面两个迭代公式来计算：

其中。

图2.4展示了当信道为BEC信道，且擦除概率为时，信道极化的效果。从图中可以看出，在非常小的时候趋向于0，在非常大的时候趋向于1。根据信道极化现象，就可以将信道划分为“好”信道和“坏”信道。对于趋向于1的“好”信道，我们可以用来传输信息数据；对于“坏”信道，不传输信息。极化码的编译码过程就是基于上述极化思想。



图2.4：BEC信道极化后的对称容量，擦出概率为，码长为。

## 2.3 极化码的编码

极化码的编码即为利用信道极化的现象，构造出一种可以达到对称信道容量的编码过程。极化码编码的基本思想是，对于所有的独立的极化信道，只在那些信道参数接近0（即趋于1，由于更适合推导编码结果，后续用它来表示信道质量）的信道中传输数据。

对于任意码长，，且，极化码的编码公式可以表示为

为极化码的生成矩阵，为输入信息比特向量，前面已经简单介绍过。假设集合为集合的任意子集，公式（2-13）可以重写为

其中表示抽取集合中的元素对应的行向量形成的子矩阵。

通过公式（2-14）可以得到，对于确定的集合和向量，存在一个由源信息比特到码字的映射关系，该映射关系其实就是一个关于的陪集码，它由固定的向量决定。这种类型的陪集码被称为陪集码，可以用参数向量来表示，其中即为集合的元素个数。其中，比率被称之为码率（Code Rate）。定义集合为信息位集合，为冻结比特向量。以下是一个简单的例子：

码的编码公式可以表示为

，其中源信息比特为，编码得到的码字为。

极化码是一种特殊的陪集码，其信息位集合由信道参数决定。由于现实中的码长不可能是无限长的，所以存在部分比特信道不完全极化现象。因此，极化码在编码时，需要根据信道参数在个比特信道中挑选出个最可靠的信道，传输信息比特。固定的冻结位则在剩余的比特信道中传输。

（此处可以添加，极化码信道挑选方案）

## 2.4 常用译码算法

从极化码诞生之时，关于极化码译码算法的研究就一直是热点问题。对于一个译码算法，可以从两个方面来衡量其性能，即误码率和译码复杂度。类似于通信系统中的有效性和可靠性参数，误码率和译码复杂度也是互相制约的。一般情况下，误码率较低的译码算法，其复杂度相对较高；反之亦然。因此，极化码的译码算法必须综合考虑上述参数。常见的译码算法有SC译码算法，BP译码算法以及SCL译码算法等。

### 2.4.1 SC译码算法

SC译码算法由Arikan在文章[12]中随着极化码一起提出。它是一种顺序迭代的译码算法，只有当前个信息比特译码完成后，才会对第个信息比特进行译码。译码器观测到由信道传输的码字，并且已知冻结位集合和冻结位信息比特，假设是关于源信息比特的译码向量，那么表示对信息比特的估计值。对于信息比特的估计分为两种情况，如果，需要译码器给出估计值；如果，由于该信息比特是已知的，不需要译码，直接赋值。

对于每一个信息比特，定义其似然值（Likelihood Ratio, LR）如下：

文章[12]中还给出了LR值的迭代计算公式，分别用式（2-16）和（2-17）表示：

公式（2-16）用于计算蝶形图上半分支的LR值，公式（2-17）用于计算蝶形图下半分支的LR值。通过公式（2-15）（2-16）（2-17）的迭代计算，最后得出各个信息比特的似然值集合，并且根据下式进行判决：

### 2.4.2 SCL译码

文章[14;15;16]提出了极化码的SCL译码算法，SCL译码算法是对SC译码算法的扩展。SCL译码器可以在译码时最多保持条译码路径，每条路径都是一个完整的SC译码过程。当时，SCL译码算法退化成SC译码算法。SCL的每一条译码路径都存在一个名为路径度量（Path Metric, PM）值的参数，用以衡量该路径的可靠程度。当PM值越小的时候，对应的译码路径正确率越高。对于信息比特，SCL译码器在计算出其对应的似然值后，不是像SC译码器那样直接进行硬判决，而是对分别给出两种估计值和，这两个估计值分别对应两条扩展的路径。在经历一段时间的路径扩展后，路径数量超出最大值，此时需要对当前所有的路径按PM值进行排序，保留前条正确概率最高的路径。

PM的初始值为0，可以通过下面的公式来计算：

其中为PM值对应的第几条路径，且，为当前的判决比特位置。为了计算方便，我们使用对数似然值（Log-likelihood ratio, LLR）来代替LR值，公式（2-19）中的即为信息比特的LLR值，它由式（2-20）给出

SCL译码器重复上述路径扩展，排序，减枝的过程，直至完成最后一个信息比特的译码。当所有的译码结果都结束时，条保存的路径会被按照PM值进行排序，PM值最小的路径保存的译码结果被视作最终的译码结果输出。

图2.5显示了一个SCL译码器在码树中搜索路径并且根据PM值的大小最后得到最佳路径的译码示意图。其发送信号的参数向量为，源信息比特向量为。假设极化码的传输信道为AWGN信道，且接收端计算出的各个子信道的对数似然比为。由图可以看出，SCL译码器最终保留的路径，其译码输出结果为。需要注意对于此刻的译码，必须要做一次比特反转操作，才能得到正确的输出，这是由于极化码的在编码时是经过了比特反转的。



图2.5：SCL译码示例。

文章[16]还提出了添加CRC的SCL译码算法（CRC-aided SCL, CA-SCL）。CA-SCL的思路是，在极化码的编码阶段，将少量的CRC比特添加到信息位中。然后，在最后的判决输出阶段，SCL译码器将保留的条路径按照PM值递增的顺序排列，再依次进行CRC校验。第一个通过CRC校验的路径的译码结果将被作为结果输出。当所有的路径都没有通过CRC校验时，SCL译码器输出PM值最小的那条路径的译码结果。

### 2.4.3 BP译码算法

在极化码的BP译码过程中，消息的传递是基于文章[25;26;32]的极化码因子图的。图2.6显示了极化码因子图中，消息传递的基本模块，它也被称为处理部件（Process Element, PE）。对于一个码长为的极化码，其因子图可以划分为阶（从第0阶开始），并且每一阶包含个PE。所有的消息都是用LLR值来表示，并且被划分为两类：从左到右传递和从右到左传递。如图2.6所示，我们用，来表示一个经过节点的，从右边到左边传递的消息；同样的，我们用来表示一个经过节点且从左向右传递的消息。



图2.6：BP译码器的基本处理部件示意图。

对于每个节点，其消息传递可以通过下列迭代公式进行计算：

其中 ，是一个缩放因子。



图2.7：条件下的极化码BP译码示意图。

图2.7是一个码长的BP译码因子图。图中的和分别表示与的初始值向量，它们可以用以下公式来计算：

初始化LLR值后，BP译码器对于每一个节点，根据公式（2-21）（2-22），按照先从右向左的再从左向右的顺序，进行LLR的更新迭代。当迭代次数达到预先设定的最大次数时，根据最左侧的LLR值，通过下式输出判决结果：

在一定范围内，BP译码器的译码性能会随着最大迭代次数的增加而上升，但是其译码复杂度也会随之上升。BP译码算法的误码性能由于SC译码算法，但是却比不上SCL译码算法在列表数较大时的误码性能。

# 第3章：附带记忆存储单元的极化码系统设计

## 3.1 系统模型与编译码原理

PCM系统模型图如图3.1所示。我们用表示在每个信息块中添加的CRC比特的数量，并且这些CRC比特是包含在信息位中的。因此在每个信息块中有源信息比特。用表示互信息比特的数量，那么每个信息块中除了互信息比特还剩下个信息比特。



图3.1：PCM的系统模型图，其底层译码器为SC译码器。

### 3.1.1 PCM编码原理

从图3.1可以看出，在编码阶段，顺序输入的源信息比特流被划分为长度为的信息块，然后，每个信息块又被划分为两个小的信息块：奇信息块（Block odd）和偶信息块（Block even）。奇信息块包含个信息比特，偶信息块包含个信息比特。接着，偶信息会从奇信息块中获得数量为的信息比特，构成一个长度也为的源信息比特向量。通过这种方式，在奇偶信息块中均包含了长度为的互信息比特，并且这些互信息比特被放置在相同的位置，用集合来表示互信息比特的位置集合。具体的输入比特操作方式如图3.2所示。通过这种方式，就完成了奇偶信息块的构造，使得奇偶信息块可以包含互信息比特。（添加算法步骤流程描述）最后，奇偶信息块在控制开关的作用下，依次添加CRC校验比特，并且经过极化码的编码。图3.2表示的PCM奇偶信息块的构造过程可以**算法1**表示：



图3.2：PCM输入比特流的操作示意图。

**算法1：奇偶信息块构造算法**

步骤1：参数初始化，获取信息位集合，冻结位集合，以及互信息比特位置集合，并且有；

步骤2：生成源信息比特向量和，并且有，；

步骤3：计算集合中的每个元素在集合中的位置，存入向量；

步骤4：获取中下标为的元素比特，存入向量；

步骤5：将插入中对应位置，完成奇偶信息块的构造。

### 3.1.2 PCM译码策略

经过编码的码字在对称B-DMC信道中传输后，其添加噪声的码字在接收端被观测。接收端每次接收长度为的信息比特，并且划分为等长的奇偶码字。SC译码器对上述奇偶码字逐个译码，分别生成信息比特的估计向量，然后对其进行CRC校验。奇偶码字的译码结果经过CRC校验产生的可能性结果如下：

* 情况1：两个码字均译码正确；
* 情况2：奇码字译码正确，偶码字译码错误；
* 情况3：奇码字译码错误，偶码字译码正确；
* 情况4：两个码字均译码错误。

假设SC译码器先对奇码字进行译码，再对偶码字进行译码。对于情况1和情况2，由于奇码字译码正确，SC译码器将个正确的互信息比特存储在记忆单元中。当偶信码字译码结束时，存放的互信息比特就有可能会被使用。具体来说，对于情况1，由于偶码字也译码正确，所以不需要进行其他的操作，译码器直接输出正确的译码结果。对于情况2，偶码字译码错误，因此可以利用存储在记忆单元里的互信息比特，对该错误码字再执行一次SC译码过程。对于情况3和情况4，由于奇码字译码错误，因此先将接收码字对应的LLR值存储起来。当后续的偶码字译码正确时，会唤醒奇码字，进行再一次的SC译码。同样的，对于情况3，由于偶码字译码正确，所以它可以提供个正确的互信息比特，帮助奇码字进行新一轮的译码。对于情况4，由于偶码字也译码错误，SC译码器宣告译码失败，译码结束。

对于上述情况2和情况3发生的新一轮的SC译码，我们再详细地阐述一下执行过程中的细节。以情况2为例，首先，SC译码器获取存储在记忆单元中的正确的互信息比特。然后，根据首次译码得出的LLR值，开始按顺序译码。当译码器计算出信息比特，的LLR值时，译码器开始对做出判决，可能出现判决情况的有：1），则根据公式（2-18）进行硬判决；2），则直接等于相应的互信息比特；3），则直接有。很明显可以看出，在对错误码字的新一轮译码过程中，重点在于当判决比特满足时，译码器将当成冻结比特处理：即无论对应的LLR值计算出来为多少，直接被判决为从奇信息块中传递的互信息比特的相应值。SC译码器按顺序译码，将所有遇到的的信息比特视作冻结位，直至译码结束。情况3中奇码字的再次译码过程的原理和情况2是相同的，只不过初始条件有差别，即存储在记忆单元的是错误码字的LLR，而正确的互信息比特是可以直接利用的。上述错误码字的第二轮译码过程可以用算法2表示。

**算法2：新一轮SC译码算法**

步骤1：初始化译码参数，获取码字的接收端LLR值向量，互信息比特向量。并且已知信息位集合，冻结位集合，互信息比特位置集合；

步骤2：设置判决信息比特计数器，设置互信息比特计数器；

步骤3：根据译码迭代公式计算信息比特的LLR值；

步骤4：判断若是，则，进入步骤6；若否，则进入步骤5；

步骤5：判断若是，则根据公式（2-18）得出；若否，则是冻结比特，直接有；

步骤6：判断若是，则，进入步骤3；若否，则所有信息比特译码完毕，译码结束。

## 3.2 系统的互信息比特位置选择准则

在上述章节，我们阐述了PCM的编译码原理。PCM方案的设计思路主要是利用在编码信息块中共享的互信息比特，让第一次译码错误的码字可以再译码一次，并且在第二次译码中将互信息比特当作冻结位，因此可以提升正确译码的可能性，降低误码率。我们知道，SC译码的结果主要取决于接收码字的LLR值和冻结位集合。因为在新一轮的译码发生时，码字的LLR值是不变的，所以影响译码结果的主要是冻结位集合，准确地说，是被视作冻结位的互信息比特位置集合。因此，选择一个合适的互信息比特位置集合，是PCM编译码方案的重点问题。

PCM中每两个连续的编码信息块都包含了位共享的互信息比特。互信息比特的位置选择问题就是需要寻找将互信息比特放置在编码信息块中的最佳位置集合。这里我们所谓的“最佳”指的是使用该互信息比特位置集合可以获得最小的系统误码性能。下面用抽象化的公式来表示这个问题。

我们已经假设集合表示互信息比特位置的集合，并且有，子向量表示所有符合的信息比特组成的向量。从理论上分析，互信息比特的位置选择方式有种。我们在第1.2小节中提到过，文章[27;28]的作者研究了SC译码的错误规律，并且发现，如果可以保证SC译码过程中，最容易出错的信息比特的正确译码，那么SC译码的性能将大大提升。受此启发，互信息比特位置集合如果刚好是在SC译码过程中那些最容易出现错误估计的信息比特，那么在新一轮的译码中，就能保证该部分信息比特译码的正确性，从而提升译码正确率。假设信息位集合是以信道质量的升序来排序，即满足关系，其中表示第位信息比特对应的比特信道的错误概率。那么最容易出现错误译码的信息比特位置集合应为中的前个元素组成的子集合，即。以下命题阐述了能够达到最小误码率的的互信息比特位置集合。

**命题1：*假设信息位集合是以对应的比特信道的质量按升序排列的，那么当集合包含集合中的前个元素，作为互信息比特的位置集合时，可以达到最小系统误码率。***

**证明：**设极化码的SC译码的统计PER用表示，文章[33]推导出了该PER的上界，可以用下式表示

公式（3-1）表示极化码的联集上界（Union Bound）。由于在新一轮的译码中，译码器可以获取正确的互信息比特集合，并且将其当作冻结位处理，因此相当于新一轮译码对应的信息位集合为，因此新一轮译码的PER可以用表示。类似于公式（3-1），新一轮译码的PER上界可以表示为：

现在假设集合表示任意一个其他的互信息比特位置集合，即，那么新一轮译码过程对应的信息位集合可以表示为。用表示该新一轮译码的PER，类似的，我们可以得到

因为集合包含了集合中的前位元素，所以中元素对应的比特信道出错的概率最大，因此有

由式（3-4）我们可以很轻松的推导出

公式（3-5）说明了对应的联集上界小于对应的联集上界，也即互信息比特集合为时系统的PER小于互信息比特集合为时的系统PER。又因为集合是任意的，因此我们可以得出结论，拥有最小的系统PER上界。因此互信息比特位置集合包含信息位集合中前位元素，是最佳的互信息比特位置选择。证明结束。

## 3.3 系统的误码率分析

在本小节，我们开始分析PCM系统的误码性能。为了能够表示得更加方便一点，我们用符号来代替，它表示PCM的底层（Underlying）误码率。所谓底层误码率，指的是除去新一轮的译码过程，PCM每个码字独立统计的PER。**命题2**给出了PCM系统的PER计算公式。

**命题2：*当PCM的编码信息块数量为2时，其PER可以表示为***

（3-6）

***其中是第二轮译码的出错概率。***



图3.3：信息块数量为2时，PCM的传输结构。

**证明：**当PCM系统中，连续发送的编码信息块数量为2时，PCM的传输结构如图3.3所示。PCM的传输码字可以被划分为许多单元，每个单元包含一个奇码字和一个偶码字。对于每一个奇偶码字组成的单元，当信息块数量为时，其译码PER应该包含以下两个部分：

* 第一部分：奇偶码字均译码错误，对应本章第二小节的译码情况4；
* 第二部分：奇码字（情况3）或者偶码字（情况2）发生了新一轮的译码，且新一轮的译码仍然错误。

对于第一部分，由于不存在新一轮的译码，所以错误概率为。对于第二部分，两个码字中有一个出错的概率为。已知在新一轮的译码过程中，译码错误的概率为，那么，对于每一个码字，发生了新一轮的译码，且新一轮译码仍然错误的概率为。对于每个单元，有两个码字，因此，当其中一个出错时，其译码错误的概率为。考虑到第一部分的误码率，可以得到一个传输信息单元的总的误码率为 因此，问题的关键是每个传输信息单元由两个码字组成。由于这里的传输信息单元可以等效为传统的极化码传输过程中的独立信息块，因此每个信息传输单元的误码率即为PCM的误码率。证明结束。

在推导出PCM系统的误码率计算公式后，我们就可以根据公式来计算PCM系统的PER，再通过仿真来验证我们的推导结果。由公式（3-6）可以看出，PCM系统的误码率主要取决于系统的底层误码率和新一轮译码的误码率。通过仿真结果我们发现底层误码率和新一轮译码的误码率之间存在一定关系，因此可以通过仿真结果，经验地总结出与之间的关系，从而可以用代替，简化公式（3-6）。

我们假设仿真时已经根据**命题1**，将PCM系统中的互信息比特放置在合适的位置，并且获得了最佳的新一轮译码的误码性能。由于与的关系是基于仿真结果的，无法用确切的表达式来推导，因此，我们可以使用多项式来近似表示。值得注意的是，从仿真的结果来看PCM系统是没有数值错误溢出（Error Floor）的情况出现的，所以近似多项式是没有常数项的。在上述前提下，可以表示为

其中，，和分别为每一项的系数。

由于公式（3-7）的平方项以及更高次数项带入公式（3-6）会得到三次方及以上的高阶项，因此可以忽略不计。所以将公式（3-7）代入公式（3-6）可以得到

通过公式（3-8）我们可以看出，PCM系统的误码率可以减小到其底层误码率的平方级别。至此，我们分析了编码信息块数量为2时的PCM系统的误码性能，并且给出了理论推导公式。在后续章节，我们会通过仿真结果来验证PCM的误码率计算公式。

## 3.4 基于BP或SCL译码器的PCM系统

我们在上述章节对于PCM系统的分析是以底层译码器为SC译码器为例进行的，事实上，底层使用的SC译码器可以完美地替换为BP或者SCL译码器。对于PCM译码原理分析中的情况2或者情况3，当奇码字中有且仅有一个码字译码错误时，底层译码器可以获取正确的互信息比特的，并且对错误码字进行新一轮的译码。需要注意的是，对于使用BP译码器的PCM系统，在新一轮的译码中使用互信息比特的最佳方式是将其视作冻结比特，而不是使用其软判决的LLR值。这样做的原因是，作为已知的冻结位，互信息比特的LLR值直接被设置为无穷大，肯定会比使用其从正确信息块中传递过来的软判决LLR值的效果好。对于使用SCL译码器的PCM系统，互信息比特直接被当作冻结位处理即可。因此，在底层译码器为BP或者SCL译码器时，PCM的译码方式与使用SC译码器时是一样的。

## 3.5 与Turbo码的比较

PCM的编码是在两个紧密相连的信息块中，共享一定数量的信息比特。由于Turbo码也是拥有两个并联的相同结构编码器，所以我们可以将PCM与Turbo做比较。相比于Turbo码，PCM有以下几个不同点。

首先，对于Turbo码，所有的输入信息比特都会经过两个相同的编码器。然而PCM仅仅是在两个编码信息块中，共享一定数量的信息比特。因此PCM可以灵活的配置编码码率。第二点不同之处是，在译码阶段，PCM不会一直不断的在两个码字中交换软信息。仅仅当一个码字译码正确而另一个码字译码错误时，互信息比特的硬判决信息才会从正确的码字传递到译码错误的码字中。PCM中的信息传递可以看成是一种零星的，不定时发生的过程：所有码字发生的新一轮译码的次数总和占总的译码次数的百分比为（用表示）

从公式（3-9）可以看出，与传统的极化码相比，上述发生的可以使得PCM的误码率显著降低的新一轮译码，付出的代价仅仅是占很小百分比的额外次数的译码。仿真结果表明，当PCM系统的误码性能达到级别时，其底层误码率已经满足，即所有码字发生新一轮译码的次数占总的译码次数的比例仅仅为不到千分之一。根据这个事实，PCM系统因为发生新一轮译码而产生的额外复杂度在低误码率条件下是可以忽略的。因此我们可以得出结论，在对系统的误码率要求较高的条件下，PCM系统因为额外译码而产生的复杂度可以忽略不计，因此PCM的译码复杂度就是其使用的底层译码器的复杂度。

## 3.6 仿真结果与分析

本节的内容主要包括PCM方案的仿真结果和分析，以及PCM方案与相关的基于动态冻结位的极化码编译码方案的仿真结果比较。我们在第1.3节已经声明，用PCM-SC-来表示包含个码字，且使用SC译码器的PCM方案。同理，PCM-BP-和PCM-SCL-分别代表包含个码字，且使用BP和SCL译码器的PCM方案。

在PCM方案中，输入信息比特经过编码后，再经BPSK调制，在加性高斯白噪声信道（Additive White Gaussian Channel, AWGN）中传输。仿真的码长为，并且极化码的信道参数是根据文章[33]中的方法，在信噪比为2 dB的条件下仿真得到的。在PCM方案和传统的CA-SCL译码算法的极化码信息比特中，均添加了16位CRC校验比特，并且采用的是5G标准中的CRC生成多项式：。因此，PCM方案的等效码率可以表示为，所有与之相比较的传统的SC，BP以及SCL译码算法的都保持相同的码率。另外，对于PCM方案底层采用的BP译码器，以及单独的BP译码算法，其译码的最大迭代次数为。

图3.4展示了PCM方案在编码信息块数量为2时的PER性能。具体的仿真参数为码长，每个编码信息块的信息位数量为，其中包含了16比特CRC和位互信息比特，因此可以计算其等效码率为。对于传统的SCL译码算法，其列表数为。从图中可以看出，PCM-SC-2的性能，在PER时，优于传统的SC译码算法的性能0.27 dB，优于传统的BP算法0.17 dB。在同样的PER条件下，PCM-BP-2达到了与SCL译码算法在列表数为2时相似的性能（性能相差不到0.05 dB），当dB时，PCM-BP-2达到了与SCL译码算法在列表数等于2时相同的性能。



图3.4：编码信息块数量为2时的PCM方案误码率，信道为AWGN信道。仿真的码长为，每个信息块有信息比特，其中包含了16位CRC校验比特和位互信息比特，因此等效码率为。作为对比，传统的SC，BP以及SCL译码算法均采用相同的码率。



图3.5：底层为SCL译码器的PCM方案的PER性能图，极化码的信息位数量为，互信息比特的数量为，其他参数均与图3.4中的仿真参数相同。

图3.5展示了底层为SCL译码器的PCM方案的PER仿真结果图。其仿真参数为，，剩下的参数均与图3.4中的仿真参数相同，因此也可以得出其仿真的等效码率为。从图中可以看出，当底层的SCL译码器列表数为2时，PCM-SCL的性能与传统的SCL译码算法在列表数时的性能相近（在PER时相差大约0.15 dB）；底层SCL译码器的列表数为4时，PCM-SCL的性能在dB时达到了与列表数为8的传统的SCL译码算法相同性能；底层SCL译码器的列表数为8时，PCM-SCL的性能在PERdB标准时，优于传统的列表数为16的SCL译码算法大约0.07 dB。



图3.6：仿真得到的PCM-SC-2的PER性能与公式（3-8）推导出的PER对比图。PCM-SC-2的仿真参数与图3.4保持一致。曲线和表示由公式（3-8）导出的PCM系统的PER上下限，其对应的取值分别为和。

图3.6展示了PCM-SC-2的仿真结果和由公式（3-8）推导出来的PER性能的比较图。公式中的值是根据仿真的数据计算得到的，再带入公式，计算出PCM-SC-2的上限和下限。从图中可以看出，在仿真的区间内（），PCM-SC-2的仿真结果在小条件下（小于2 dB）符合其理论推导的下限，而在大时（大于3.5 dB）趋于其理论推导的上限。因此仿真的结果验证了公式（3-8）的正确性，也证明了第3.3小节关于PCM系统的误码率分析是正确的。

PCM系统良好的性能来自于其间或发生的新一轮的译码，公式（3-9）表示了新一轮译码的次数与总的译码次数所占的百分比。更准确地说，该百分比是由所有发生的新一轮的译码的次数与总共的译码次数的百分比。图3.7中的曲线表示了额外译码次数所占的百分比。曲线表示的是PCM系统的底层极化码的PER，另外，曲线表示的是额外发生的新一轮译码中，其译码正确的次数的百分比。从图中可以看出，当 dB时，曲线和曲线是相匹配的。需要注意的是，额外发生的新一轮的译码次数，确实受到PCM系统的底层误码率的影响：图中的曲线在大信噪比下，也与另外两条曲线相匹配。这也验证了公式（3-9）的推导：当底层的极化码的误码率非常小时（接近于1），额外发生的译码次数的百分比取决于的大小。因此PCM系统的PER也减小到了这个级别。



图3.7：公式（3-10）表示的，额外发生的译码次数所占的总译码次数的百分比（曲线表示）。仿真的参数与图3.4中的仿真参数相同。曲线表示底层极化码的PER，曲线表示额外发生且译码正确的新一轮的译码次数的百分比。



图3.8：PCM方案与PIC，IFA-SCL方案的仿真结果比较。所有比较的方案采用的底层译码器均为列表数为的SCL译码器，且仿真的等效码率均为。PCM的其他参数与图3.5的仿真参数一致。

图3.8展示了PCM方案与PIC，IFA-SCL极化码系统方案的仿真比较结果。仿真的码长均为，且码率为，底层译码器为列表数的SCL译码器。PCM 系统的仿真参数与图3.5的仿真参数相同，且PIC系统与IFA-SCL方案的仿真参数分别与文章[29]，[30]中的仿真参数保持一致。从图中可以看出，PCM系统的性能远远好于IFA-SCL系统，这是因为IFA-SCL方案对新一轮的译码发生的条件要求比较高，因此新一轮译码发生的次数比较少，纠错效率低于PCM系统。图中还可以看到，PCM系统的性能与PIC系统的性能是相互匹配的。需要注意的是，PIC系统的实现复杂度是远远高于PCM系统的，这是因为PIC系统采用了极化码的系统码编码，并且使用了删余技术来降低码率的损失。

# 第4章：基于多信息块编码的PCM系统

在上一章节，我们提出了包含两个编码信息块的PCM系统方案，并且通过仿真结果验证了理论分析。在此基础上，我们很自然地想到一个问题：可以扩展PCM系统的编码信息块数量，即使得编码信息块个数满足，并且保持甚至达到一个更好的误码性能吗？因此，本章首先提出了PCM系统从2个编码信息块到个编码信息块（）的直接扩展方案。然后，在此方案的基础上，提出了一种可以降低码率的多信息块编码的PCM方案，并且不会带来性能的损失。该方案被称之为一般型PCM（General PCM）方案。

## 4.1 直接型PCM扩展方案

我们在3.3节分析了PCM系统的误码性能，并且推导出每个传输信息单元包含两个编码信息块时的PER计算公式为。现在我们假设PCM系统的每个传输信息单元包含的编码信息块数量为，，并且每个信息块都包含了位互信息比特，那么此时的PCM系统的PER由对应以下几种情况：

* 第一部分：在个信息块中只有一个信息块译码错误，并且在新一轮译码再次出错；
* 第二部分：有两个信息块译码错误，并且在新一轮译码过程中，至少有一个信息块译码出错；
* **······**
* 第部分：所有的个信息块均译码错误。

对于第一部分，由于译码错误的信息块在新一轮译码时再次出错，所以在个信息块中最终只一个信息块译码出错，因此，这种情况下的PER可以表示为

对于第二部分，由于有两个信息块译码错误，因此最终在个信息块中出现错误信息块的情况可能有两种：1）两个出错的信息块，在新一轮译码时，其中一个译码正确，另一个译码错误；2）两个出错的信息块在新一轮译码时均在再次出错。因此，综合考虑这两种情况，我们可以得到第二部分的PER计算公式为

观察公式（4-1）（4-2），对于第部分的PER，我们可以推导出其PER计算公式为

对于第部分，由于所有的信息块都译码错误，所以对应的误码率为。将所有部分的PER累加起来，我们可以得到总体的多信息块编码的PCM系统，其误码率公式为

类似于在两个编码信息块组成的PCM系统中PER推导的处理方式，我们同样用来代替代入公式（4-4），因此，公式（4-4）可以重写为

由公式（4-5）可以看出，当相对较小时，PCM系统的PER取决于，也即为当PCM系统中只有一个信息块译码错误，而其他信息块都译码成功时的情况。在这种情况下PCM的误码率级别为的平方项，而其他情况下的PCM系统的误码率级别至少为的三次方。基于这个发现，我们将在下一小节提出一种能够降低码率损失的多信息块PCM编码方案，我们称之为一般型PCM扩展方案（General PCM）。在一般型PCM扩展方案中，当有且仅有一个信息块译码错误时，错误信息块才会开启新一轮的译码。对于上述提到的其他情况，译码器直接宣布译码失败。该方案可以保持与PCM的直接扩展方案同等级别的误码性能，但是提升了多信息块编码的PCM系统的码率。

**4.2 一般型PCM扩展方案**

一般型的PCM扩展方案是基于上一小节的PCM系统的直接扩展提出来的。在上一小节中，我们分析了PCM系统的直接扩展的误码组成。由公式（4-5）可以看出，直接扩展的 PCM系统的误码率也处于的平方级别，即系统的性能受到上一小节中关于PCM系统的直接扩展方案的PER组成第一部分的限制，而其他部分的PER远远小于这个级别，因此可以忽略不计。基于上述情况，我们提出一种一般型PCM扩展方案，即当有且仅有一个信息块译码错误时，该错误信息块才会开启新一轮的译码，当有两个或两个以上的信息块译码错误时，就不启动新一轮的译码，译码失败，直接宣布译码结束。相比于PCM的直接扩展方案，一般型PCM扩展方案可以降低码率的损失，并且不会影响译码的性能。下面详细介绍一般型PCM扩展方案的编译码原理。

图4.1展示了该方案的输入比特流构造编码信息块的示意图。从图中可以看出，每个大的信息块都包含个小的编码信息块，前个信息块均不包含互信息比特，第个信息块中包含的互信息比特是由前个信息块中相同位置的信息比特，模二加得到的。因此，一般型PCM扩展方案的输入信息比特构造编码信息块的过程可以用以下公式表示

其中，表示第个编码信息块中信息位集合对应的信息比特组成的向量。每一个信息块的互信息比特位置的选取遵循**命题1**：互信息比特被放置在对信息比特保护能力最差的信息位集合上。



图4.1：降低码率的扩展方案输入比特流构造编码信息块示意图。

对于直接扩展的多信息块编码PCM系统，其等效码率的计算公式为

其中因为互信息比特而损失的码率为，这是因为PCM的直接扩展方案的每一个信息块都包含了位互信息比特。对于一般型PCM扩展方案，其等效码率可以表示为

对于固定数量的，当越大时，由互信息比特造成的码率损失就越小。但是，连续编码的信息块数量越多，PCM系统的译码延时就越高。因此，在实际应用时，应当综合考虑，选择一个较为合适的编码信息块的数量。

下面我们来分析一般型PCM扩展方案的误码性能，并且从理论推导的角度证明一般型PCM扩展方案并不会带来性能上的损失。在一般型PCM系统扩展方案的译码阶段，当个信息块中有且仅有一个信息块译码错误时，译码出错的信息块中的互信息比特可以通过公式（4-9）被纠正过来，即

当个信息块中出现两个及以上的信息块译码错误时，采用该编码方案的PCM系统无法纠正译码错误的信息块，因为无法得到正确的互信息比特。

一般型PCM扩展方案的PER计算公式可以表示为：

与公式（4-4）对比，可以看出公式（4-10）中，只有项包含了新一轮译码的误码率，体现出一般型PCM扩展方案的译码策略：当译码错误的信息块有且仅有一个时，才进行新一轮的译码。类似地，我们可以将带入公式（4-10），可以将公式（4-10）重写为：

比较公式（4-5）和公式（4-11），其表示的PCM 系统误码率均处于级别。因此与直接扩展的多信息块PCM相比，一般型PCM扩展方案并没有带来性能的损失，但是它却降低了码率的损失。因此，在没有特别说明的条件下，本文默认使用的PCM扩展方案均为一般型PCM扩展方案。

## 4.3 仿真结果分析

图4.2是一般型PCM扩展方案的仿真结果图。仿真的码长为，信息位数量为，其中包含了位互信息比特。根据公式（4-8）可以算出仿真的PCM-SC-3的码率为0.5，因此图中出现的传统的SC，BP，以及SCL算法仿真的码率均为0.5，且码长与PCM-SC-3的信息块码长相同。PCM-SC-2的仿真参数与第3.6节中图3.4的仿真参数相同。

从图中可以看出，在PER级别为时，PCM-SC-3的性能比传统的SC算法的性能好0.4 dB，比传统的BP算法的性能好0.3 dB，该仿真结果验证了我们的分析，即一般型PCM扩展方案的PER性能并不会受到影响。从图中还可以看出PCM-SC-3的PER结果略优于PCM-SC-2，这是因为该扩展方案降低了系统损失的码率，因此其码率调整的噪声会变小，所以会带来性能的优化。



图4.2：降低码率的扩展方案系统仿真图。仿真的编码信息块数量为，码长为，等效码率为。

# 第5章：附带记忆存储单元的极化码系统硬件实现

本章提出了PCM系统基于SC译码器的两种硬件结构，并且给出了这两种结构的译码器与传统的SC译码器以及SCL译码器的硬件实现比较结果。这两种结构分别被称为顺序（In-serial, IS）译码结构和低延时交织（Low-latency interleaved, LLI）译码结构。IS译码结构基于文章[19]中提出的SC译码器结构，其中包括了被设计用来执行预计算的处理单元（Processing Element, PE）。这种IS结构可以用于SC译码，也可以在SC译码的基础上，进行PCM-SC系统的译码。LLI结构译码器是受到文章[31]中2层交织译码的SC译码器的启发后提出来的。与IS结构译码器相比，LLI译码器可以显著地降低译码延时，其代价是增加了少量的硬件资源消耗。

## 5.1 基于顺序结构的PCM译码器

图5.1展示了PCM-SC-2系统采用IS结构实现的译码器硬件结构图，从图中可以看出，该译码器结构主要包括5个模块：LLR存储模块，SC译码器模块，CRC校验模块，反馈（Feedback）模块，以及比特存储模块。与文章[19]中的SC译码器结构相比较，该译码器新增了LLR存储模块和比特存储模块。



图5.1：IS结构的PCM译码器。

LLR存储单元用于存储第3.2小节中，误码率分析的情况2和情况3 中的译码错误的信息块的LLR值。比特存储单元是该译码器硬件结构的重要单元，在传统的SC译码器的译码过程中，冻结位的位置和内容是提前设置好的，当比特存储单元接收到冻结比特时，会忽略该冻结比特，并且立即将其发送到反馈模块。在PCM的IS结构译码器的译码过程中，奇偶码字的首次译码过程与传统的SC译码器的译码过程是一样的。但是当奇偶码字中有一个通过了CRC校验时，比特存储单元会立即将译码正确的互信息比特存储起来。当出现上述情况2或者情况3时，译码错误的码字在进行新一轮译码的过程中，SC译码器会从比特存储单元中读取正确的互信息比特，并且将这些互信息比特视作冻结位处理。通过这种方式，译码正确信息块中的位互信息比特可以高效地通过比特存储单元传递至译码错误的信息块中，帮助其完成新一轮的译码。

## 5.2 基于低延迟交织译码结构的PCM译码器

对于上一小节中提出的IS结构的译码器，当发生新一轮的译码时，整个译码过程所需要的时间是传统SC译码器的1.5倍。这是由于IS结构译码器必须在等待首轮译码结束后，根据CRC校验结果，才能启动新一轮的译码。我们在前面章节提到过，在PCM系统的译码阶段，当发生新一轮的译码时，SC译码器并不会重新计算接收端码字对应的LLR值，而是使用原来的，译码错误的LLR。因此，新一轮译码与首次译码的区别就在于新一轮的译码过程中，多了一部分由互信息比特集合组成的冻结位。所以我们可以得出结论，新一轮的SC译码过程中，在译码进行到第一个被纠正的互信息比特之前时，所有信息比特判决结果和首轮译码是相同的。基于这个事实，我们提出了一种高效的低延迟交织译码器结构，该译码器结构可以节省系统译码的冗余时间，并且通过交织译码提升译码器的效率，从而降低译码的延时。

一般来说，传统的位（）SC译码器，可以划分为阶层（Stage）：从第1阶层到第阶层。传统的SC译码器在每个时钟周期（Clock Cycle）中有且仅有一个阶层是处于激活状态的，其余阶层处于闲置状态。文章[31]提出了能够进行交织译码的SC译码器，

该交织译码器的思路是，当第一个码字的译码进行到了译码器的第层时，译码器的前阶层处于闲置状态，因此可以用于第二个码字的译码，这样就提高了译码的效率。受到该种交织译码器的启发，我们提出了PCM系统的LLI结构译码器。该译码器的译码原则是当奇码字的译码进行到第层时，其前个处于闲置状态的阶层可以同时用于偶码字的译码。但是需要注意一点，当译码进行到最后一层，即第阶层时，上述奇偶码字的译码会出现冲突，这是因为译码过程中第阶段的计算过程需要两个时钟周期。为了解决这个问题，与IS结构的译码器相比，LLI结构的译码器采取了不同的硬件设计。



图5.2：LLI结构的译码器硬件设计图。

图5.2展示了LLI结构的译码器硬件设计图。从图中可以看出，LLI结构译码器的组成部分与IS结构译码大致相同，同样包括了SC译码器模块，比特存储模块，反馈模块，CRC校验模块。与IS结构译码器的不同之处主要有以下几点：1）LLI结构译码器的SC译码模块是2层交织的译码结构，具体表现在其第7阶层包含了额外的PE；2）LLI结构的译码器包含了两个比特存储模块和两个反馈模块，因此可以同时对奇偶码字进行译码操作；3）LLI结构的译码器没有LLR存储模块。除了以上三点，还需要注意的是LLI结构译码器中的PE不是IS结构译码器中传统的PE，它还添加了额外的寄存器和断点存储单元，以便保存断点的数据。当出现第3.1节中译码的情况2或者情况3时，译码器将计算出来的LLR立即存储在PE中的断点存储单元中，以便后续的计算。由于这种特殊的PE设计，LLI结构的译码器不存在LLR存储模块。图5.3是LLI结构译码器中的PE结构示意图。从图中可以看出，与IS结构译码器中的传统PE相比，它包含了额外的寄存器和断点存储单元。

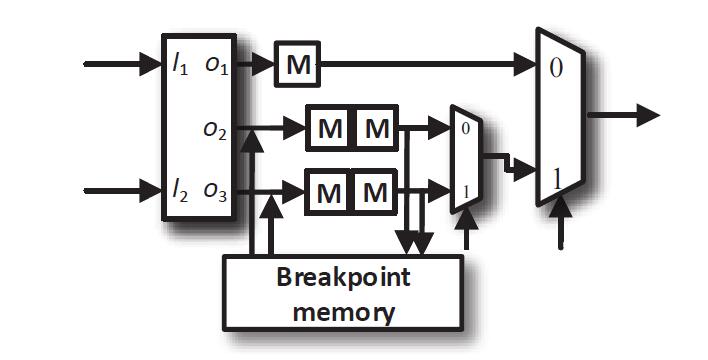


图5.3：LLI结构中的PE结构示意图。

在了解了LLI译码器的硬件结构之后，我们再来阐述一下其工作原理。假设奇码字的译码提前偶码字一个时钟周期，那么当偶码字的译码进行到互信息比特的判决阶段时，译码器会立即将该互信息比特的判决值与相同位置的，提前一个时钟周期判决完成的奇码字中的互信息比特相比较。若两个互信息比特的判决值相同，则继续译码下一个信息比特，直至完成最后一个互信息比特的判决。若两个互信息比特的判决值不同，译码器会立即中断译码过程，并将当前计算出的所有奇偶码字相关的LLR值送至断点存储单元，然后恢复译码。当奇偶码字整个译码结束，并且经过CRC校验后，出现了上述情况2或者情况3时，译码错误的码字将开启新一轮的译码。由于在第一个出错的信息互信息比特判决前，其LLR计算值已经保存在断点存储单元，因此可以直接读取LLR值，从当前出错的互信息比特开始译码。当使用图3.4的参数进行硬件仿真时，可以计算出互信息比特的位置从第128个信息比特开始，结束于第834个信息比特（非连续）。这意味着即使是在最坏的情况下，即第一个互信息比特就译码错误，新一轮的译码也只需要从判决第128个信息比特处开始计算LLR值。因此，通过这种方式，LLI结构的译码器可以节省新一轮译码中的重复的计算过程，从而节省译码的时间，降低译码延时。

## 5.3 硬件仿真结果与分析

我们在Xilinx ZNYQ-7000 可编程门阵列（FPGA）平台上对IS结构译码器，LLI结构译码器，传统的SC译码器，以及列表数为的SCL译码器进行了仿真实现。表5.1展示了上述不同类型的极化码译码器在码长时的硬件仿真结果。从表中我们可以看出，传统的SC译码器消耗的硬件资源（LUT和FF）最少。IS结构的译码器的硬件消耗比传统的SC译码器的硬件消耗多5.5%，这是因为它包含了额外的LLR存储模块和比特存储模块。此外，IS结构译码器的总的硬件消耗约为列表数为2的SCL译码器的硬件消耗的一半。LLI结构译码器消耗的硬件资源大约为IS结构译码消耗的硬件资源的1.9倍，这是因为LLI译码器的交织译码设计结构包含了额外的比特存储模块，反馈模块，以及更复杂的PE设计。值得注意的是，LLI结构的译码器消耗的硬件资源仍然小于列表数为2的SCL译码器消耗的硬件资源，两者约相差10.7%。

表5.1：极化码的不同译码器在码长条件下的实现综合结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 译码器 | PCM译码器 | | SC译码器[8] | SCL译码器[17] |
| **IS结构** | **LLI结构** |
| LUTs | 32668 | 42179 | 31833 | 82853 |
| FFs | 14022 | 47081 | 12418 | 17120 |
| Total | 46690 | 89260 | 44251 | 99973 |
| Max. Freq. [MHz] | 103.4 | 98.7 | 150.8 | 122.0 |
| Latency [us] | 20.17 | 10.59 | 13.57 | 33.56 |
| T/P [Mbps] | 101.45 | 193.37 | 150.95 | 61.03 |

表5.1还展示了各个极化码译码器的译码延时和吞吐率。从表中可以看出，LLI结构译码器的译码延时仅仅只有IS结构译码器译码延时的一半，但是前者的吞吐率却几乎是后者的两倍，这个仿真结果验证了我们在上一小节中的关于LLI结构译码器的性能分析。此外，LLI结构的PCM译码器与传统的SC译码器和SCL译码器相比，译码延时分别只有SC译码器和SCL译码器译码延时的78.4%和31.6%，但是LLI结构的PCM译码器的吞吐率却分别是传统SC译码器吞吐率的1.28倍，SCL译码器吞吐率的3.17倍。

通过硬件仿真的综合结果，我们可以得出，采用LLI结构的PCM译码器，与列表数为2的SCL译码器相比，拥有更少的硬件资源消耗，更低的译码延时，和更高的译码吞吐率。综合考虑3.6节中PER性能仿真结果，可以看出PCM方案是一个高效率并且低消耗的极化码系统的设计方案。

# 第6章：总结与展望

极化码是一种基于信道极化现象提出来的编码方式，它的编译码算法比较简单，并且在码长足够长时可以达到香农极限。极化码的良好性能使得它从提出以来一直都是信道编码方向的研究热点，如今更是被选用作为5G应用场景中控制信道的信道编码方案，具备重要的研究意义和广阔的应用前景。然而，当极化码的码长无法满足无限长的条件时，信道极化就会出现不彻底的现象，很大程度地影响了极化码的译码性能。因此，对极化码的研究的一个重要方向就是如何提升极化码在中等码长条件下的译码性能，这也是本文主要研究工作想要尝试解决的问题。

本文在比较分析了几种基于动态冻结位的极化码系统设计方案的基础上，提出了附带记忆存储单元的极化码编译码方案。我们首先在第三章详细地阐述了PCM方案在编码信息块数量为2时的编译码原理：在编码端，通过操作输入信息比特流，构成两个共享互信息比特的信息块，然后分别添加CRC校验比特，并参与极化码编码；在译码时，当CRC校验的结果出现一个码字译码正确而另一个码字译码错误的情况时，译码器就可以获取存储在记忆单元里的，互信息比特正确的硬判决信息，然后对译码错误的码字开启新一轮的译码，并且在新一轮的译码中将互信息比特当作冻结位处理。在描述了PCM编译码方案的原理后，我们证明了选择极化码的信息位集合中那些信道质量最差的信息位作为互信息比特位置集合，可以获得最佳的系统误码性能。在随后的第3.3节，我们从理论上分析了PCM的误码率性能，并且推导出误码率计算公式。PCM方案的底层译码器可以是任意的极化码传统译码器，使用不同的译码器并不会改变PCM方案的译码策略。我们同时还分析了PCM系统方案与同样存在若干个信息块相关编码的Turbo码的不同之处，并且简单地描述了PCM编译码方案的系统复杂度。在第三章的最后小节，我们通过仿真的结果验证了关于PCM编译码方案的理论分析，该方案确实可以带来极化码性能的提升。本文第四章的主要内容是多个信息块参与编码的PCM扩展方案，包括PCM的直接扩展方案和一般型PCM扩展方案。一般型PCM扩展方案不仅降低了系统的码率损失，还没有影响系统的误码性能，因此比PCM的直接扩展方案更加适用。第五章提出了PCM编译码方案的两种译码器结构的硬件设计：IS译码器结构和LLI译码器结构。IS结构的译码器设计简单，硬件资源消耗较少，但是译码的延时较高；LLI结构的译码器采用了交织译码结构，可以大大降低译码的延时，但是消耗的硬件资源高于IS结构的译码器。硬件仿真的结果也验证了我们对于两种硬件结构的译码器的分析。

PCM编译码方案是一套完整的极化码编译码方案，但是它还是存在着一些问题。比如在误码率分析的过程中，能否推导出新一轮译码的误码率与底层译码器的误码率的具体关系，而不是使用多项式来近似的表达。还有就是互信息比特的数量选择问题，不同数量的互信息比特会有不同的译码性能，能否找到一个合适的互信息比特数量，从而达到系统的最佳性能，也是一个值得探索的问题。在下一步的研究计划中，我们希望能够找到上述问题的答案，从而完善我们的附带记忆存储单元的极化码编译码方案。

# 参考文献

[1] Shannon C E. A mathematical theory of communication[J]. Bell System Technical Journal, 1948, 27(3): 379-423.

[2] Hamming R W. Error detecting and error correcting codes[J]. The Bell System Technical Journal, 1950, 29(2): 147-160.

[3] Golay M J. Notes on digital coding[J]. Proc. IEEE, 1949, 37: 657.

[4] Reed I S. A class of multiple-error-correcting codes and the decoding scheme[R]. Massachusetts Inst Of Tech Lexington Lincoln Lab, 1953.

[5] Muller D E. Application of boolean algebra to switching circuit design and to error detection[J]. Transactions of the IRE Professional Group on Electronic Computers, 1954(3): 6-12.

[6] Peterson W W, Brown D T. Cyclic codes for error detection[J]. Proceedings of the IRE, 1961, 49(1): 228-235.

[7] P.Elias. Coding for noisy channels[J]. IRE Convention Record, 1955: 37-46.

[8] Viterbi A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm[J]. IEEE transactions on Information Theory, 1967, 13(2): 260-269.

[9] Berrou C, Glavieux A, Thitimajshima P. Near Shannon limit error-correcting coding and decoding: Turbo-codes[C]. IEEE International Conference on Communications, 1993: 1064-1070.

[10] Berrou C, Glavieux A. Near optimum error correcting coding and decoding: Turbo-codes[J]. IEEE Transactions on Communications, 1996, 44(10): 1261-1271.

[11] Gallager R. Low-density parity-check codes[J]. IRE Transactions on Information Theory, 1962, 8(1): 21-28.

[12] Arikan E. Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels[J]. IEEE Transactions on Information Theory, 2009, 55(7): 3051-3073.

[13] Hussami N, Korada S B, Urbanke R. Performance of Polar Codes for Channel and Source Coding[C]. 2009 IEEE International Symposium on Information Theory, 2009: 1488-1492.

[14] Niu K, Chen K. CRC-Aided Decoding of Polar Codes[J]. IEEE Communications Letters, 2012, 16(10): 1668-1671.

[15] Chen K, Niu K, Lin J. Improved Successive Cancellation Decoding of Polar Codes[J]. IEEE Transactions on Communications, 2013, 61(8): 3100-3107.

[16] Tal I, Vardy A. List Decoding of Polar Codes[J]. IEEE Transactions on Information Theory, 2015, 61(5): 2213-2226.

[17] Leroux C, Tal I, Vardy A, et al. Hardware architectures for successive cancellation decoding of polar codes[C]. 2011 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2011: 1665-1668.

[18] Leroux C, Raymond A J, Sarkis G, et al. Hardware Implementation of Successive-Cancellation Decoders for Polar Codes[J]. Journal of Signal Processing Systems for Signal Image and Video Technology, 2012, 69(3): 305-315.

[19] Zhang C, Parhi K K. Low-Latency Sequential and Overlapped Architectures for Successive Cancellation Polar Decoder[J]. IEEE Transactions on Signal Processing, 2013, 61(10): 2429-2441.

[20] Giard P, Sarkis G, Thibeault C, et al. Multi-Mode Unrolled Architectures for Polar Decoders[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2016, 63(9): 1443-1453.

[21] Balatsoukas-Stimming A, Raymond A J, Gross W J, et al. Hardware Architecture for List Successive Cancellation Decoding of Polar Codes[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2014, 61(8): 609-613.

[22] Balatsoukas-Stimming A, Parizi M B, Burg A. LLR-Based Successive Cancellation List Decoding of Polar Codes[J]. IEEE Transactions on Signal Processing, 2015, 63(19): 5165-5179.

[23] Zhang C, You X, Sha J. Hardware Architecture for List Successive Cancellation Polar Decoder[C]. 2014 IEEE International Symposium on Circuits and Systems, 2014: 209-212.

[24] Hashemi S A, Condo C, Gross W J. Fast and Flexible Successive-Cancellation List Decoders for Polar Codes[J]. IEEE Transactions on Signal Processing, 2017, 65(21): 5756-5769.

[25] Arikan E. A performance comparison of polar codes and reed-muller codes[J]. IEEE Communications Letters, 2008, 12(6): 447-449.

[26] Eslami A, Pishro-Nik H. On bit error rate performance of polar codes in finite regime[C]. 2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton): 188-194.

[27] Afisiadis O, Balatsoukas-Stimming A, Burg A. A Low-Complexity Improved Successive Cancellation Decoder for Polar Codes[C]. Conference Record of the 2014 Forty-Eighth Asilomar Conference on Signals, Systems & Computers, 2014: 2116-2120.

[28] Condo C, Ercan F, Gross W J: Improved Successive Cancellation Flip Decoding of Polar Codes Based on Error Distribution, 2018 IEEE Wireless Communications and Networking Conference Workshops, 2018: 19-24.

[29] Wu X, Yang L, Xie Y, et al. Partially Information Coupled Polar Codes[J]. IEEE Access, 2018, 6: 63689-63702.

[30] Zheng H, Hashemi S A, Chen B, et al. Inter-Frame Polar Coding With Dynamic Frozen Bits[J]. IEEE Communications Letters, 2019, 23(9): 1462-1465.

[31] Zhang C, Parhi K K. Interleaved Successive Cancellation Polar Decoders[C]. 2014 IEEE International Symposium on Circuits and Systems, 2014: 401-404.

[32] Yuan B, Parhi K K. Architecture optimizations for bp polar decoders[C]. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013: 2654-2658.

[33] Tal I, Vardy A. How to Construct Polar Codes[J]. IEEE Transactions on Information Theory, 2013, 59(10): 6562-6582.