基于动态冻结位的极化码编译码方案设计与实现

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）[1]的引言部分中写道：

“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：

(bit/s) (1-1)

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[1]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。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[2]（Reed-muller Code）与循环冗余校验码[3]（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[4]中提出了不同于分组码的另一类型的信道编码——卷积码（Convolution 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[5]，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。1993年，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[6]，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[7]。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[8]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——Polar码。Polar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，Polar码在很多方面的表现都优于Turbo码。由于LDPC码和Polar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[9]，将LDPC码作为数据信道的编码方案，Polar码作为5G广播和控制信道的编码方案。

Polar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Polar码的研究具有很高的实际意义。Polar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，Polar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Polar码译码性能急剧恶化，无法满足应用的需求。因此，如何提升Polar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，Polar码是由土耳其教授E. Arikan于2009年在文章[8]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Polar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Polar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。为了改善Polar码在中等码长条件下的误码性能，文章[9;10;11]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[11]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[12;13;14;15]介绍了SC译码算法的硬件实现；文章[16;17;18;19]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[20;21]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[22]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[23]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。在文章[22;23]中，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[24;25]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Polar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[24]提出了部分信息位耦合（Partially Information Coupled, PIC）的Polar码编译码方案。PIC方案采用了Polar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[25]提出了Polar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Polar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）前后码字相关的编码结构使得译码错误的码字的动态冻结位难以计算，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

本文提出了一种极化码的新型编译码方案：在两块紧密相连的编码块中，共享一定数量的信息比特。系统的输入比特流被划分为奇信息比特和偶信息比特。在编码过程中，一组奇偶信息块共享一小块信息比特，并称之为互信息比特（Mutual information bits, MIB）。每个信息块还包含了CRC比特。在译码阶段，当一对信息块中只有一组译码失败时，译码正确的信息块可以给译码错误的信息块提供关于MIB的硬判决信息。合理地设计MIB的位置，可以使得译码错误的信息块开始新一轮的译码，并且被纠正过来。

由于是两个紧密相连的信息块共享MIB，所以这种极化码的编译码方案看起来就像是在编码阶段存在一些记忆存储单元。因此，这种方案被命名为有记忆的极化码（Polar codes with memory, PCM），用以区分传统的极化码编译码方案。另外，PCM还可以直接拓展到个信息块的场景。基于此，本文还提出了一种普适型PCM（General PCM, G-PCM）方案，与直接拓展PCM的方案相比，该方案具有更小的码率损失，并且能维持相似的PER。PCM的底层译码器可以是SC，BP或者SCL译码器，其译码复杂度极为底层译码器的复杂度。

本文的主要贡献如下：

1. 提出了PCM编译码方案，包括PCM的系统模型，编译码策略，误码性能分析。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的在使用列表数为的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. 拓展PCM至的场景，并且提出了G-PCM方案。与PCM的直接拓展方案相比，该方案可以实现更小的码率损失。PCM-SC-3的仿真结果表明，G-PCM方案没有损失误码性能。
3. 提出了两种实现PCM的硬件结构，即顺序（In-serial, IS）结构和低延迟交织（Low-latency interleaved, LLI）结构。硬件实现的结果表明，在码长的条件下，与文献[17]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。

文章剩余的结构安排如下：第2章简单介绍极化码的基本原理，编码方法以及常见的译码算法。第3章系统的介绍PCM方案。具体来说，第3章第1小节介绍PCM的编码方案；第2小节介绍PCM的译码策略；第3小节介绍MIB的最佳选择；第4小节分析了PCM的误码性能；采用BP和SCL译码的PCM出现在第5小节；在第6小节，我们还将PCM于Turbo码做了对比。第4章是关于PCM的拓展。第5章提供了关于PCM的仿真结果。在第6章，我们介绍了PCM的两种硬件实现。最后一章是总结部分。

第2章：极化码的相关工作

2.1 引言部分

Arikan在文献[7]中详细的介绍了极化码的极化原理。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。用表示一种对的次使用的信道；因此有：，并且转移概率为。

对于任意给定的B-DMC信道，存在两个性能参数：对称容量（Symmetric capacity）

(2-1)

和巴氏参数（Bhattacharyya parameter）

. (2-2)

这两个参数可以用来衡量信道传输的速率和可靠性。是在信息在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。

定义表示一个长度为N的向量。在给定且集合的条件下，用来表示子向量（）。表示关于的补集。此外，表示集合中的元素个数。

2.2 信道极化原理

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。信道极化分为信道联合（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

. (2-3)

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。



图2.3：联合信道的构造。

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

. (2-4)

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

(2-5)

其中表示信道的输出，表示其输入。

2.2.3 信道极化

信道极化定理：对于任意给定的B-DMC类型的，其极化信道满足，当趋向于无穷大时，其对应的信道对称容量趋向于1或者0.

该定理的证明已经在文献[7]中给出。图2.4展示了当信道为BEC信道，且擦出概率为时，信道极化的效果。可以通过下面两个迭代公式来计算：

(2-6)

(2-7)

其中。



图2.4：BEC信道极化后的对称容量，擦出概率为，。

从图2.4也可以看出，在非常小的时候趋向于0，在非常大的时候趋向于1。信道极化后，可以选择一个合理的值，来挑选出质量较好的信道，传输信息比特。

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）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

(2-8)

为极化码的生成矩阵，在前面已经简单介绍过。假设集合为集合的任意子集，公式（8）可以重写为

（2-9）

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（9）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。我们将这种类型的陪集码称为陪集码，可以用参数向量来表示，其中即为集合的元素个数。此为，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。

2.4 常用译码算法

极化码的译码算法主要有SC译码算法，BP译码算法以及SCL译码算法。我们将分别介绍这三种常用的译码算法。

2.4.1 SC译码

SC译码算法由Arikan在文献[7]中随着极化码一起提出。它是一种顺序译码的算法。对于每一个信息比特，其似然值（Likelihood ratio, LR）可以用下式计算

（2-10）

信息比特可以根据下式进行判决：

（2-11）

对于冻结比特，由于其在发送端和接收端都是已知的，所以当满足时，直接有。文献[7]中还提供了计算似然值的迭代公式：

（2-12）

（2-13）

2.4.2 SCL译码

文献[8;9;10]提出了SCL译码算法，SCL译码器可以在极化码译码树中搜索正确的译码路径，并且允许同时追踪条译码的最佳路径。对于每一条译码路径，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。SCL译码器从第一个判决比特开始，在每个信息比特的判决过程中，都会将当前保存的列表数翻倍，直至列表数已达到设置上限条。此时，SCL译码器会先将列表数翻倍为条，然后再对这条路径按照其PM值进行排序，并保留条PM值最小的路径。PM的初始值为0，通过下面的方程来计算：

（2-14）

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（14）中的即为信息比特的LLR值，它由下式给出

（2-15）

在最后的信息比特判决阶段，SCL译码器会找到PM值最小的路径对应的判决比特，作为译码器的输出。

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。

文献[10]还提出了添加CRC的SCL译码算法（CRC-aided SCL, CA-SCL）。在极化码的编码阶段，少量的CRC比特被添加到了信息位中。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。

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献[19;20;25]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，它也被称为流程元素（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PE。

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：

（2-16）

（2-17）

其中 ，是一个缩放因子。

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：

（2-18）

（2-19）

初始化LLR值后，BP译码器对于每一个节点，根据公式（2-16）（2-17），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：



图2.7：条件下的极化码BP译码示意图。

第3章：共享信息位的极化码系统

3.1 共享信息位的极化码系统编译码原理

共享信息位的极化码系统模型图如图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。用表示共享的互信息比特的数量，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 系统的编码原理阐述

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帧，然后，每个信息帧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）。其中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取信息比特，形成一个长度也为的输入信息比特向量，然后添加CRC比特。通过这种方式，在奇偶信息块中均包含了确切的个互信息比特，这些互信息比特被放置在相同的位置，我们用来表示互信息比特的位置集合。输入比特的操作方式如图3.2所示。上述奇偶信息块的编码可以顺序进行，也可以同时进行。如图3.1所示，奇偶信息块均需要添加CRC比特并且经过极化码的编码器，这个顺序是由开关控制的。

3.1.2 系统的译码原理

经过编码的码字在对称B-DMC信道中传输后，其添加噪声的码字符号（Symbols）在接收端被观测。接收端每次接收个比特作为一组：前个形成奇信息块的，后个为偶信息块。对于每个信息块，SC译码器产生对信息比特的估计向量，然后对其进行CRC校验。CRC校验产生的可能性结果如下：

* 情况1：奇偶信息块均译码错误；



图3.2：PCM输入比特流的操作示意图。

* 情况2：奇信息块译码正确，但是偶信息块译码失败；
* 情况3：奇信息块译码失败，但是偶信息块译码正确；
* 情况4：奇偶信息块均译码失败。

对于情况1和情况2，由于奇信息块译码正确，SC译码器将个正确的互信息比特存储在记忆单元中。当偶信息块译码结束时，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，由于偶信息块也译码正确，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，偶信息块译码错误，因此偶信息块会获取存储在记忆单元里的互信息比特，开始执行新一轮的SC译码。对于情况3和情况4，由于奇信息块译码错误，因此其初始化的个LLR的值会存放在记忆单元里面。当后续的偶信息块译码成功时，会唤醒奇信息块，进行新一轮的译码。同样的，对于情况3，由于偶信息块译码正确，所以它可以提供个正确的互信息比特，帮助奇信息块进行新一轮的译码。对于情况4，由于偶信息块也译码错误，SC译码器宣告译码失败，译码结束。

对于上述情况1和情况3发生的新一轮的SC译码，我们再来详细介绍一下其执行过程中的细节。以情况2为例，首先，译码正确的奇信息块中，个正确的互信息比特被传递到译码错误的偶信息块中。然后，偶信息块开始重复上一轮的SC译码，按顺序对计算各个信息比特的LLR值，并且产生判决比特，，直到译码进行到了第一个互信息比特，即。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中奇信息块的第二轮译码过程，与情况2类似。

3.2 系统的互信息比特位置选择准则

在上述章节，我们已经知道，PCM的每两个连续发送的信息块都包含了个共享的互信息比特。互信息比特的放置问题就是需要寻找一种最佳的方式将互信息比特放置在奇偶信息块中。这里我们所谓的“最佳”指的是该种放置方式能获得最小的系统误码性能。下面我们用抽象化的公式来表示这个问题。

我们已经设定集合表示互信息比特位置的集合，并且，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。假设信息位集合是以信道质量的升序来排序，也即有关系，其中表示第个信息比特对应的比特信道的错误概率。以下命题表示了能够达到最佳联集上界（Union bound）的互信息比特防止方式。

**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，可以达到最小的联集上界。***

**证明：**定义在信息位集合为时，其译码的PER为，则其联集上界可以使用文献[26]中的公式表示

（3-1）

由于PCM是有一对连续的信息块编码形成的，因此，当其中某一个信息块发生再次译码的时候，相当于该信息块本次译码对应的信息位集合为。这是因为当一个信息块译码正确而另一个信息块译码错误时，译码错误的信息块将获取正确的互信息比特，并在第二轮译码过程中，将互信息比特视作冻结位信息。

在上述条件下，该信息块第二轮译码正确的概率小于以下联集上界

（3-2）

现在假设集合表示任意一个其他的互信息比特位置集合，则错误信息块第二轮译码对应的信息位集合可以表示为，类似的，我们可以得到

（3-3）

因为集合中的元素是集合中个比特信道出错概率最大的元素，因此有

（3-4）

由式（23）我们可以很轻松的推导出

（3-5）

公式（24）说明了对应的联集上界小于对应的联集上界。又因为集合是任意的，因此我们可以得出结论，拥有最小的联集上界。因此互信息比特位置集合包含信息位集合中前个元素，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，我们开始分析共享信息位的极化码系统的误码性能。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，其信息位集合为。

**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的传输帧结构。

**证明：**当PCM系统中，连续的编码信息块数量为2时，PCM的传输帧结构如图3.3所示。PCM的编码信息块可以被划分为许多单元，每个单元包含一个奇信息块和一个偶信息块。对于每一个奇偶信息块组成的单元，当信息块数量为

时，其误码率包含以下两个部分：

* 第一部分：奇偶信息块均译码错误，对应本章第二小节的译码情况4；
* 第二部分：奇信息块（情况3）或者偶信息块（情况2）发生了第二轮译码，且第二轮译码仍然错误。

对于第一部分，其错误概率为。对于第二部分，两个信息块中有一个出错的概率为。已知在新一轮的译码过程中，信息块译码错误的概率为，那么，对于每一个信息块，当发生新一轮的译码，其仍然错误的概率为。对于每个单元，有两个信息块，因此，当其中一个出错时，其信息块的译码错误概率为。考虑到第一部分的误码率，可以得到一个信息单元的总的误码率为

（3-7）

因此，问题的关键是每个信息单元由两个信息块组成。由于这里的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，因此公式（26）中的即为PCM的误码率。证明结束。

根据**命题1**，将PCM系统中的互信息比特放置在合适的位置，可以获得最佳的错误纠正效果。对于PCM系统的第二轮译码的误码率，其大小其实是受影响的。由于与的关系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

（3-8）

其中，，和分别为每一项的系数。

由于公式（27）的平方项以及更高次数项带入公式（26）会得到三次方及以上的高阶项，因此可以忽略不计。所以公式（26）可以重写为

（3-9）

通过公式（28）我们可以看出，PCM系统的误码率可以减小到其底层的极化码误码率的平方级别。

3.4 应用BP或者SCL译码器的PCM

在我们提出的PCM系统中，其底层使用的SC译码器可以完美地替换为BP或者SCL译码器。对于本章第二小节分析中的情况2或者情况3，当奇偶信息块中有且仅有一个信息块译码错误时，译码正确的信息块可以提供互信息比特的正确硬判决信息，来供译码错误的信息块使用。需要注意的是，对于使用BP译码器的PCM系统，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做冻结位处理即可。因此，即使是使用BP或者SCL译码器，PCM的译码方式与使用SC译码器时，是保持一致的。

3.5 与Turbo码的比较

（此处可以插一下Turbo码的示意图）

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PCM可以与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均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，共享一小段的信息比特。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信息块中交换软信息。仅仅当一个信息块译码正确而另一个信息块译码错误时，互信息比特的估计信息才会从正确的信息块传递到译码错误的信息块中。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：所有信息块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）

（3-10）

从公式（25）可以看出，与单独的计划码相比，上述发生的新一轮的译码可以使得PCM的误码率显著降地，付出的代价仅仅是上述占很小百分比的额外次数的译码。

3.6 仿真结果与分析

本节的内容主要包括部分信息位共享的极化码系统的仿真结果和分析，以及部分信息位共享的极化码系统与绪论章节中提到的，与之相类似的工作的仿真结果比较与分析。为了描述方便，我们用PCM-SC-来表示拥有个共享互信息比特的信息块，且使用SC译码器的PCM方案。同理，PCM-BP-和PCM-SCL-分别代表使用BP以及SCL译码器的PCM方案。

在部分信息位共享的极化码系统中，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献[26]中的方法，在信噪比为2 dB的条件下仿真得到的。在底层为SCL译码器的部分信息位共享极化码系统，或者单独的CA-SCL译码算法中，均添加了16位CRC校验比特，其采用的是5G标准中的CRC生成多项式：。因此，该系统的等效码率可以表示为，所有与之相比较的单独的SC，BP以及SCL译码算法的均保持相同的码率。另外，对于系统底层采用的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：编码信息块数量为2时的部分信息位共享极化码系统，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，单独的SC，BP以及SCL译码算法均采用相同的码率。

图3.4展示了部分信息位共享的极化码系统在编码信息块数量为2时的PER性能。其仿真参数为码长，每个信息块的信息位数量为，其中包含了16比特CRC和个互信息比特，因此可以计算其等效码率为。对于单独的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，优于单独的SC译码算法的性能0.27 dB，优于单独的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3.5：底层为SCL译码器的部分信息位共享极化码系统的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译码器的部分信息位共享的极化码系统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的性能与单独的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的单独的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，优于单独的列表数为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（3-9）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（3-9）推导出的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（3-9）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（3-9）的正确性，也证明了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（3-10）表示了新一轮译码所占的百分比。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。需要注意的重点是，额外发生的，新一轮的译码次数，受到底层的极化码的误码率影响：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（3-10）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与PIC，Inter-frame极化码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与Inter-frame，PIC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Inter-frame与PIC系统的仿真参数分别与其参考文献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Inter-frame系统，这是因为Inter-frame系统对新一轮的译码发生的条件要求比较高，因此其发生的次数比较少，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，这是因为其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：多信息块编码的共享信息位极化码

4.1 引言

在上一章节，我们提出了在两个紧密相连的编码信息块中共享固定数量信息比特的有记忆的极化码系统。我们很自然的想到一个问题：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，并且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，提出了一种可以提升多信息块编码情形下PCM系统的码率的方案，并且不会造成性能的损失。该方案被称之为通用PCM（General PCM）方案。

4.2 共享信息位极化码系统的直接扩展

我们在3.3节分析了部分信息位共享的极化码系统的误码性能，并且推导出其在每个信息帧包含两个编码信息块时的PER计算公式为。我们假设PCM系统的每个信息帧包含的编码信息块数量为，，并且每个信息块都包含了个互信息比特，那么此时的PCM系统的PER由以下几个部分组成：

* 第一部分：在个信息块中只有一个信息块译码错误，并且其第二轮译码再次出错；
* 第二部分：有两个信息块译码错误，并且在第二轮译码过程中，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第二轮译码时再次出错，所以在个信息块中一共错了一个信息块，因此，这种情况下的PER可以表示为

（4-1）

对于第二部分，由于有两个信息块译码错误，因此最后在个信息块中出现错误信息块的可能有两种：1）两个出错的信息块，在第二轮译码时，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第二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

（4-2）

一般的，对于第部分的PER，我们可以推导出其PER计算公式如下：

（4-3）

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

（4-4）

类似于在两个编码块组成的PCM系统中PER推导的处理方式，我们同样用来代替，因此，公式（31）可以表示为

（4-5）

由公式（32）可以看出，当相对较小时，PCM系统的PER取决于，也即为当系统中只有一个信息块译码错误，而其他信息块均译码成功时的情形。在这种情况下PCM的误码率为的平方项，而其他情况下的PCM系统的误码率阶数至少为的三次方。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。该编码方案可以保持与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 降低码率的扩展方式

多信息块的PCM系统是基于上一小节的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。从公式（32）可以看出，多信息块编码的PCM系统的误码率仍然与两个信息块组成的PCM的误码率保持一致，即误码率被限制在的平方级别。且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

（4-6）

其中因为互信息比特而损失的码率为。为了减少多信息块编码的PCM系统的码率损失，并且不会降低误码性能，我们提出了一种降低码率的扩展方案。该方案的思路是当个信息块中，有且仅有一个信息块译码错误时，错误的信息块才会进行新一轮的译码；对于两个或以上的信息块译码错误的情况，PCM译码器直接宣布译码错误，不进行下一步操作。图4.1展示了该方案的输入比特流操作示意图，其中每个大的信息帧都包含个编码信息块。

从图4.1可以看出，每一个大的信息帧都包含了个编码信息块，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，多信息块的PCM的降低码率的扩展方案，输入信息比特流的编码方案可以用以下公式表示

（4-7）

其中，表示第个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上。



图4.1：降低码率的扩展方案输入比特流操作示意图

对于该降低码率的多信息块编码的PCM扩展方案，其等效码率为

（4-8）

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，但是其译码的延迟也会越来越大。因此，在实际应用时，应当综合考虑，选择最佳的参数。

在使用降低码率的PCM系统扩展方案进行编译码的条件下，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。

对于降低码率的PCM扩展方案，其PER计算公式可以表示为：

（4-9）

与公式（4-4）对比，可以看出公式（4-9）中，只有项包含了第二轮译码的误码率，体现出降低码率的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，PCM才进行第二轮译码。类似地，我们可以将带入公式（4-9），得到降低码率的多信息块PCM系统的近似误码率为

（4-10）

比较公式（4-5）和公式（4-10），其表示的误码率均处于级别。因此与直接扩展的多信息块PCM相比，降低码率的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。

4.3 仿真结果分析

图4.2是部分信息位共享的极化码系统采用降低码率的扩展方案仿真结果图。仿真的信息块码长为，信息位数量为，其中包含了个互信息比特。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章中图3.4的仿真参数相同。

从图4.2可以看出，在PER等级为时，PCM-SC-3的性能比单独的SC算法的性能好0.4 dB，比单独的BP算法的性能好0.3 dB，这说明了采用降低码率的扩展方案的极化码，其PER性能并不会受到影响，验证了我们在上一小节的理论分析。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1024，码率为0.5。

第5章：有记忆的极化码的硬件实现

本章提出部分信息位共享的极化码系统的两种硬件结构，和实现结果。这两种结构分别被称为顺序（In-serial, IS）译码结构和低延迟交织（Low-latency interleaved, LLI）译码结构。IS译码结构基于文献[13]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执行元素（Processing element, PE）。这种IS结构可以用于SC译码，也可以在SC姨妈的基础上，进行PCM-SC系统的译码。LLI译码器结构是受到文献[26]中，2层交织译码的SC译码器的影响，提出来的。与IS译码器结构相比，LLI译码器可以显著地降低译码延迟，并且在硬件资源的消耗上仅仅只有少量的增加。

5.1 基于顺序译码结构的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，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。与文献[13]中的SC译码器相比较，PCM-SC-2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3.1.2小节中的情况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的译码过程中，奇信息块和偶信息块的第一轮译码过程与传统的SC译码器的译码过程是一样的。当奇信息块或者偶信息块通过了CRC校验时，比特存储单元立即将该信息块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，译码错误的信息块在进行第二轮译码的过程中，比特存储单元会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

上一小节中提出的IS结构的PCM译码器，当存在新一轮的译码时，即上述的译码情况2或者情况3时，其译码所需要的时间是传统SC译码器的1.5倍。研究发现，当奇信息块或者偶信息块进行新一轮的译码时，其关于LLR值的计算，在进行到第一个出错的互信息比特之前，都是冗余的。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，该译码器结构可以实现交织译码，降低译码的延时。

一个传统的位（比特）SC译码器，包含个阶段（Stage）：从第1阶段到第阶段。对于每个时钟周期（Clock cycle），有且仅有一个阶段是处于激活状态的。文献[26]中的采用交织译码的SC译码器可以对两个极化码信息块同时译码，受到该种交织译码器的启发，LLI结构的PCM译码器被提出。LLI结构的译码器，核心思路是当奇信息块的译码进行到阶段时，其译码器的前阶段均处于闲置状态，因此可以用于偶信息块的同时译码。当译码进行到最后一个阶段，即第阶段时，上述奇偶信息块的译码会出现冲突，这是因为每个信息块在第阶段的计算过程需要两个时钟周期。因此与IS结构相比，LLI结构的译码器需要额外的执行单元（PE）。



图5.2：LLI结构的PCM译码器硬件结构图。

图5.2展示了采用LLI结构的PCM译码器硬件结构图。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，其中包括2层交织的SC译码器模块，比特存储模块，反馈模块，CRC校验模块。与IS结构译码器不同的是，LLI结构的SC译码模块在第７阶段包含了额外的PE。需要注意的是，此处的PE并不是普通的PE，与IS结构中，传统的PE相比，它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，因此LLI结构的译码器不需要LLR存储模块。另外， LLI结构的译码器还包含了两个比特存储模块和两个反馈模块，所以可以同时对奇偶信息块进行译码操作。图5.3是LLI结构的PE单元示意图。从图中可以看出，与IS结构的PCM译码器相比，它包含了额外的两个寄存器和断点存储单元。

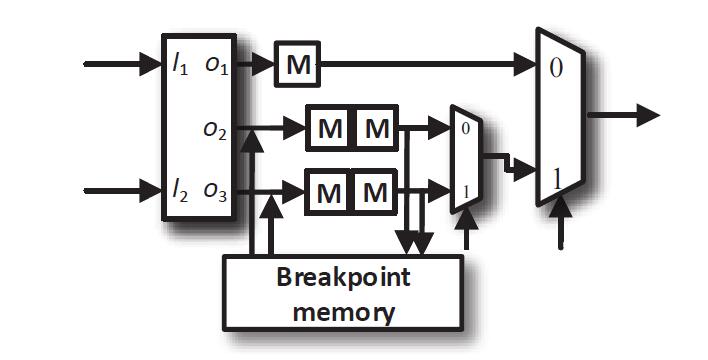


图5.3：LLI结构中的PE单元。

LLI译码器的工作原理是，当偶信息块的译码（假设奇信息块的译码提前偶信息块一个时钟周期）进行到互信息比特的判决时，该互信息比特会与相同位置的，已经判决奇信息块中的互信息比特相比较。如果两个互信息比特的判决值相同，则译码继续，直到判决完最后一个互信息比特。当两个互信息比特的判决值不同时，所有存储在寄存器里面的，两个信息块的当前LLR值，会被立即送至断点存储单元，然后译码继续进行。当后续的CRC判定结果显示，出现了情况2或者情况3时，译码错误的信息块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。当研究情况为图3.4的仿真参数时，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）。即使是在最坏的情况下，第一个互信息比特即译码错误，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，LLI结构的PCM译码器可以很大程度上降低译码的延迟。

5.3 硬件实现结果

本小节在Xilinx ZNYQ-7000 可编程门阵列（FPGA）平台上实现了IS结构与LLI结构的PCM译码器，传统的SC译码器，以及列表数为的SCL译码器。由于采用了奇偶信息块交织译码的硬件结构，并且新一轮的译码可以从第一个出错的互信息比特开始计算LLR的值，因此LLI结构的PCM译码器延时小于IS结构译码器的时延。

表5.1展示了不同类型的极化码译码器在码长时的硬件实现结果，其中包括了IS和LLI结构的PCM译码器，传统的SC译码器以及列表数为2的SCL译码器。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%，这是因为其包含了额外的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为IS结构译码的1.9倍，这是因为LLI译码器采用了交织译码的硬件结构。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理论分析。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，其译码延时分别只有SC译码器和SCL译码器的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译码器的1.28倍，SCL译码器的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3.7节的仿真结果中，PCM的PER性能，可以看出PCM系统是一个高效率并且低消耗的极化码编码设计和译码的方案。

第6章：总结与展望

6.1 全文主要工作总结

6.2 下一步研究方向