

高吞吐率的并行化极化码 CRC-SCL 译码器的 FPGA 实现

刘彦君^{1,2***}, 李岩^{1,2*}, 刘宇畅^{1,2}, 贾晓硕^{1,2}, 周洪航^{1,2}, 洪小斌^{1,2},
邱吉芳^{1,2}, 郭宏翔^{1,2}, 左勇^{1,2}, 李蔚^{1,2}, 伍剑^{1,2**}

¹北京邮电大学电子工程学院, 北京 100876;

²北京邮电大学信息光子学与光通信国家重点实验室, 北京 100876

摘要 极化码是通过严格的数学方法证明能达到信道容量极限的编码方案。为了使极化码应用于高速实时光通信系统中,采用完全展开式流水线架构,基于 Xilinx xcvu13p-flga2577-1-e 型号的现场可编程门阵列(FPGA)芯片,首次从硬件上实现了 256 码长适配多种码率的循环冗余校验码辅助的连续消除列表(CRC-SCL)译码器。该译码器在 156.25 MHz 的最大时钟频率下,总吞吐率达到 40 Gbit/s。同时进行了基于极化码的 56-Gbit/s(28-GBaud)正交相移键控背靠背实验。实验结果表明,所实现的 CRC-SCL 译码器能够在误码率为 10^{-3} 处获得 7.5 dB 的编码增益。此外,为了权衡纠错性能和资源量消耗,分析了量化位数对译码器性能的影响。

关键词 光通信; 相位调制; 光纤通信; 极化码; 现场可编程门阵列

中图分类号 TN929.11

文献标志码 A

doi: 10.3788/AOS202141.1706002

Implementation of FPGA Based on High Throughput Parallel CRC-SCL Decoder of Polar Codes

Liu Yanjun^{1,2***}, Li Yan^{1,2*}, Liu Yuyang^{1,2}, Jia Xiaoshuo^{1,2}, Zhou Honghang^{1,2},
Hong Xiaobin^{1,2}, Qiu Jifang^{1,2}, Guo Hongxiang^{1,2}, Zuo Yong^{1,2}, Li Wei^{1,2}, Wu Jian^{1,2**}

¹School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China;

²State Key Laboratory of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications, Beijing 100876, China

Abstract Polar codes belong to a coding scheme that is proved to reach the limit of channel capacity by strict mathematical methods. To apply polar codes to high-speed real-time optical communication systems, we adopt a fully-unrolled pipeline architecture in this paper. Based on Xilinx xcvu13p-flga2577-1-e field programmable gate array (FPGA) chips, we develop a successive cancellation list (CRC-SCL) decoder based on cyclic redundancy check codes with a code length of 256 for the first time in hardware, and this decoder can be adapted to various code rates. The total throughput of the decoder reaches 40 Gbit/s at the maximum clock frequency of 156.25 MHz. In addition, we carry out 56-Gbit/s (28-Gbaud) QPSK back-to-back experiments based on polar codes. Experimental results show that the CRC-SCL decoder implemented in this paper can obtain a coding gain of 7.5 dB when the bit error rate is equal to 10^{-3} . Furthermore, to balance the error correction performance and resource consumption, we also analyze the influence of quantization bits on the decoder performance.

收稿日期: 2021-02-06; 修回日期: 2021-03-04; 录用日期: 2021-03-23

基金项目: 国家重点研发计划课题(2019YFB1803601)、国家自然科学基金(62021005,61875019)、中央高校基本科研业务基金

通信作者: *liyan1980@bupt.edu.cn; **jianwu@bupt.edu.cn; ***yanjun_std@163.com

Key words optical communications; phase modulation; fiber-optic communication; polar code; field programmable gate array

OCIS codes 060.4510; 060.5060; 060.2330

1 引言

极化码是由 Arikan 教授^[1-2]于 2008 年提出的一种新型的前向纠错(FEC)技术,具有低复杂度、低时延、无错误平层、短码性能好等诸多显著优势。使用连续消除(SC)算法译码时,极化码可实现任意二进制输入离散无记忆对称信道的信道容量,因此引发了学术界和工业界的广泛关注。

译码算法的优劣是极化码纠错性能的主要决定因素。在码长足够大时,极化码通过 SC 译码算法能够达到信道容量的极限(即香农界)。然而在实际的系统中,编码码长较长导致系统复杂度过大,难以实现应用,且大多数的现场可编程门阵列(FPGA)、ASIC 芯片难以满足长码极化码所占用的硬件资源量。而中短码长的极化码在 SC 算法译码下的纠错性能却不够理想。改进的 SC 算法——连续消除列表(SCL)算法^[3]近年来受到了越来越多的关注,因为它显著提高了短至中等码长的极化码的纠错性能^[4]。且在基于级联循环冗余校验(CRC)码的 SCL 译码算法下^[5],中短码长的极化码在纠错性能方面可以超过 Turbo 码^[6]和 LDPC 码^[3]。近年来,多家机构均已从硬件上实现并验证了基于串行架构的 SCL 译码算法的可行性^[7-11],并获得了百兆量级的吞吐率。然而,光通信中的传输速率多为 10 Gbit/s 以上^[12],并且通信系统也在向高吞吐率、低时延的方向发展^[13]。上述的极化码译码器的吞吐率较低,达不到高速传输系统的要求。为了让极化码更适用于高速系统,一些研究提出了并行化极化码的高速硬件实现方案。比如,在文献[14]中,研究者利用流水线架构验证了 SC 译码器在 10 Gbit/s 速率以上的可行性,其吞吐率达到了 636 Gbit/s。Giard 等^[15]利用基于完全展开 Fast-SSC-List(Fast Simplified Successive Cancellation List)算法实现的译码器获得了 12 Gbit/s 的吞吐率,虽然 Fast-SSC-List 算法相对于 CRC-SCL 算法消耗较少的资源量,但其纠错性能较差。Fast-SSC-List 的架构采用极化子码的设计方案,因此在不同码长、码率的硬件实现上通用性不佳。总之,这两种译码器的吞吐率虽能达到 10 Gbit/s 以上,但纠错性能都不如 CRC-SCL 算法。而迄今为止,未见实现具有高灵

活、高吞吐率特性的 CRC-SCL 硬件译码器的相关报道。基于上述研究,为了提高译码器的纠错性能,同时尽可能降低硬件资源量的需求,以保持极化码在短码下的竞争优势,本文采用 Xilinx xcvu13p-flga2577-1-e 型号的 FPGA 平台,实现了 256 码长的并行化 CRC-SCL 译码器。结果表明,该译码器的总吞吐率达到了 40 Gbit/s,比目前最快的 CRC-SCL 译码器^[10]高一个数量级,且其能对多种码率的极化码进行译码,不失灵活性。与此同时,进行了基于极化码的 56-Gbit/s QPSK(polar-QPSK)背靠背实验,使用两个本文设计的 40 Gbit/s CRC-SCL 译码器进行 IQ 两路译码处理,完成了 56-Gbit/s 的 polar-QPSK 实验的译码。实验结果表明,本文实现的 CRC-SCL 译码器在误码率为 10^{-3} 时具有 7.5 dB 的编码增益。此外,本文还分析并优化了译码器的量化位数。当对数似然比量化为 8 bit,路径度量值为 12 bit 时,该译码器仍保持着优异的纠错性能,且对比原始方案,LUT、BRAM 等资源使用率分别降低了 6%、7%。

本文将介绍极化码 CRC-SCL 算法的原理,展示极化码 CRC-SCL 译码器的硬件设计方案,介绍极化码辅助的 QPSK 系统的背靠背实验,并展示了本文设计的 CRC-SCL 译码器的性能,最后给出结论。

2 原理

本文用 (N, K) 表示编码位数为 N 、信息位数为 K 的极化码,那么编码速率可表示为 $R = K/N$ 。用 $\mathbf{y} = (y_i)_{i=1}^N = \mathbf{y}_1^N$ 表示信道输出的接收信号,用 $\hat{\mathbf{u}} = (\hat{u}_i)_{i=1}^N = \hat{\mathbf{u}}_1^N$ 表示译码估计的信息比特向量。CRC-SCL 译码算法是一种贪婪算法,它使用信道输出信号 \mathbf{y} 和已经完成译码估计的前 $i-1$ 个比特 \hat{u}_1^{i-1} 来估计当前比特 \hat{u}_i 的值,按照发送顺序译码得到最可能的码字序列。图 1 概述了 CRC-SCL 算法的译码流程。如图 1(a)所示,CRC-SCL 算法分为两部分:软信息计算(模块①)和判决计算(模块②)。其中软信息计算部分包括对数似然比(LLR)和部分和项的计算,其流程大致如下:利用 CRC-SCL 译码算法计算信道输出信号的信道对数似然比(CLLR);接着根据译码蝶形图^[16],使用 CLLR 值计算蝶形图中各

层级的左上蝶形 LLR(上节点 LLR)和左下蝶形 LLR(下节点 LLR)。在计算下节点 LLR 时需要用到部分和项的计算结果,其中部分和项是对已经完成译码的估计比特 \hat{u}_1^{i-1} 进行异或计算得到的结果。判决计算部分包括路径度量值 (PM) 计算、排序 (sortor) 和 CRC 校验 (CRC-CHECK), 其计算流程

如下:随着译码的进行,算法在估计完每一个比特后会保留 L 条路径,并且为沿途的每条路径计算 PM 值;然而每个比特译码会扩展出 $2L$ 条路径,因此需要通过 PM 值进行排序筛选并保留其中的 L 条;最后通过 CRC 校验得到译码输出序列。下面分别对软信息计算部分和判决计算部分进行详细介绍。

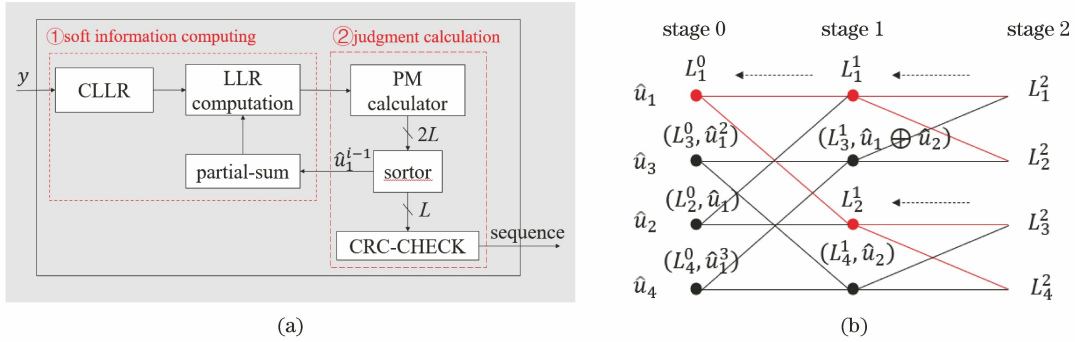


图 1 CRC-SCL 框图。(a) CRC-SCL 框架概述; (b) $N=4$ 极化码的译码蝶形图
Fig. 1 Block diagrams of CRC-SCL. (a) Overview of CRC-SCL framework;
(b) decoding butterfly graph of polar codes for $N=4$

2.1 软信息计算部分

在 CRC-SCL 译码算法中,LLR 值的计算起到了关键作用,因此在本节中主要介绍 LLR 等相关软信息的计算方法。译码中的软信息计算过程可以用蝶形图描述,图 1(b) 表示 $N=4$ (即码长为 4) 的译码蝶形图。将 $N(\log_2 N + 1)$ 个 LLR 节点分为 $\log_2 N + 1$ 个层级 (stage), 并基于蝶形图将它们成对地组合起来。将位于第 s 层的第 i 个 LLR 表示为 L_i^s , 其中 $i \in [1, N], s \in [0, \log_2 N]$ 。在图 1(b) 的右边, $L_i^{\log_2 N}$ 是从层级 $\log_2 N$ 输入到蝶形图的 CLLR。在图 1(b) 的左边, L_i^0 对应于 N 个叶节点的 LLR, \hat{u}_i 为译码估计比特。除了 CLLR 可直接使用之外,其余 LLR 值的计算从右往左按层级进行。以 $N=4$ 的极化码译码为例,LLR 具体的计算过程如图 2 所示:先将收到的 4 个 CLLR 值进行上节点 (或下节点) 的计算,得到 2 个新的 LLR 值,接着到下一层时,继续计算这 2 个新的 LLR 值的上节点或下节点,得到 1 个新 LLR 值,以此类推,每进行 1 次计算,得到的 LLR 个数减少为 $1/2$,最后一层仅得到一个 LLR 值。

在 高 斯 白 噪 声 (AWGN) 信 道 下, 噪 声 方 差 设 为 δ^2 , 调 制 格 式 选 择 BPSK, $y_i (i=1, 2, \dots, N)$ 为 信 道 输 出 信 号, 则 计 算 CLLR 的 公 式 可 以 简 化 为

$$L_i^{\log_2 N}(y_i) = \frac{2}{\delta^2} y_i \quad (1)$$

除了第 $\log_2 N$ 层的 LLR (即 CLLR) 是通过使用

(1) 式计算得到之外,其余层级的 LLR (即蝶形图中上下节点 LLR) 的原始计算公式^[16] 分别为

$$R_{LLR_N^{(2i-1)}}(y_1^N, \hat{u}_1^{2i-2}) = \frac{R_{LLR_{N/2}^{(i)}}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}) R_{LLR_{N/2}^{(i)}}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2}) + 1}{R_{LLR_{N/2}^{(i)}}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}) + R_{LLR_{N/2}^{(i)}}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2})} \quad (2)$$

$$R_{LLR_N^{(2i)}}(y_1^N, \hat{u}_1^{2i-1}) = [R_{LLR_{N/2}^{(i)}}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2})]^{1-2\hat{u}_1^{2i-1}} \cdot R_{LLR_{N/2}^{(i)}}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2}) \quad (3)$$

式中: $\hat{u}_{1,o}^{2i-2}$ 表示取估计比特 \hat{u}_1^{2i-2} 中的奇数序号的比特值, 同理 $\hat{u}_{1,e}^{2i-2}$ 表示取 \hat{u}_1^{2i-2} 中的偶数序号; $R_{LLR_N^{(2i-1)}}$ 为总码长为 N 的第 $2i-1$ 个 LLR, $R_{LLR_{N/2}^{(i)}}$ 为码长为 $N/2$ 的第 i 个 LLR。总码长为 N 的第 $2i-1$ 个 LLR 的计算被简化为码长为 $N/2$ 的两个 LLR 的计算。

为了简化它们在硬件中的实现,采用文献[17]提出的符号与幅值算法。对于蝶形图中的上节点 LLR 值,使用 f 函数进行计算,用 g 函数计算下节点 LLR 值。 f 函数可简化成函数 f_1 和 f_2 (f_1 表示上节点 LLR 的符号位, f_2 表示其幅值), 二者分别表示为

$$f_1(R_{LLR_N^{(2i-1)}}, R_{LLR_N^{(2i)}}) = \varphi(R_{LLR_N^{(2i-1)}}) \oplus \varphi(R_{LLR_N^{(2i)}}) \quad (4)$$

$$f_2(R_{\text{LLR}_N^{(2i-1)}}, R_{\text{LLR}_N^{(2i)}}) = \min(|R_{\text{LLR}_N^{(2i-1)}}|, |R_{\text{LLR}_N^{(2i)}}|), \quad (5)$$

其中 $\varphi(x)$ 函数的定义为

$$\varphi(x) = \begin{cases} 1, & x \geq 0 \\ -1, & \text{else} \end{cases}, \quad (6)$$

式中: \oplus 表示异或。

函数 g 可简化成函数 g_1 和 g_2 (g_1 表示上节点 LLR 的符号位, g_2 表示其幅值), 分别表示为

$$g_1(R_{\text{LLR}_N^{(2i-1)}}, R_{\text{LLR}_N^{(2i)}}) = \bar{\gamma}_R \cdot \varphi(R_{\text{LLR}_N^{(2i)}}) +$$

$$\gamma_R \cdot [\hat{s} \oplus \varphi(R_{\text{LLR}_N^{(2i-1)}})], \quad (7)$$

$$g_2(R_{\text{LLR}_N^{(2i-1)}}, R_{\text{LLR}_N^{(2i)}}) =$$

$$\max(|R_{\text{LLR}_N^{(2i-1)}}|, |R_{\text{LLR}_N^{(2i)}}|) +$$

$$(-1)^\lambda \min(|R_{\text{LLR}_N^{(2i-1)}}|, |R_{\text{LLR}_N^{(2i)}}|), \quad (8)$$

其中

$$\lambda = \hat{s} \oplus \varphi(R_{\text{LLR}_N^{(2i-1)}}) \oplus \varphi(R_{\text{LLR}_N^{(2i)}}), \quad (9)$$

比较因子 γ_{ab} 的定义为

$$\gamma_{ab} = \begin{cases} 1, & |a| > |b| \\ 0, & \text{else} \end{cases}, \quad (10)$$

式中: \hat{s} 代表部分和项。本设计根据文献[17]中的指示函数计算部分和项。那么部分和项的计算公式为

$$\hat{s}(s, i, z) = \overline{B(s, i)} \cdot \prod_{v=0}^{s-1} [B(s-v-1, z) + B(v, i)], \quad (11)$$

其中 B 函数的定义为

$$B(a, b) = (b/2^a) \bmod 2, \quad (12)$$

式中: s, i 分别表示第 s 层级和当前译码的第 i 个比特; z 表示每个层级中触发的索引下标。

2.2 判决计算部分

在 2.1 节中得到最后一层的对数似然比 LLR 后, 需要计算路径度量值 PM 来决定译码路径的选择, 公式如下:

$$V_{\text{PM}_l^{(i)}} = \begin{cases} V_{\text{PM}_l^{(i-1)}}, & \hat{u}_i[l] \notin \eta \text{ and } \hat{u}_i[l] = \delta[L_l^{(i)}(l)] \\ V_{\text{PM}_l^{(i-1)}} + |L_l^{(i)}[l]|, & \hat{u}_i[l] \notin \eta \text{ and } \hat{u}_i[l] \neq \delta[L_l^{(i)}(l)], \\ + \infty, & \hat{u}_i[l] \in \eta \end{cases}, \quad (13)$$

$$\delta(x) = \begin{cases} 0, & x \geq 0 \\ 1, & \text{else} \end{cases}, \quad (14)$$

式中: $V_{\text{PM}_l^{(i-1)}}$ 表示第 l 路径第 $i-1$ 个比特对应的路径度量值; $L_l^{(i)}$ 为第 l 条路径的第 i 个比特对数似然比值; $\hat{u}_i[l]$ 是第 l 条路径的当前译码比特; η 表示极化码编码时的固定比特且取值错误的集合。

每个比特估计时都会扩展出 $2L$ 条路径, 对这 $2L$ 条路径的 PM 值进行排序^[3], 并选择 PM 值最小的前 L 条所对应的路径, 即每次排序后都只保留 L 条路径。当译码结束后, 对保留下来的所有 L 条候选码字进行 CRC 校验。若这 L 条路径中存在通过 CRC 校验的路径, 则选择 PM 值最小的路径作为译码序列输出; 若 L 条路径均未能通过 CRC 校验, 则直接选择 PM 值最小的译码路径。

3 硬件实现

CRC-SCL 算法本质上是逐个比特判决的串行译码结构, 其存在很大的译码时延。因此无论采用传统的树状架构^[18] 还是重叠结构^[19] 硬件实现 CRC-SCL 译码器, 都难以使其吞吐率获得大幅度的

提升。而正如文献[20]所提到的, 展开式的流水线架构提供了极高的译码速度, 可以满足光通信系统高速传输的高吞吐率低延迟的要求, 因此需使用完全展开并利用深度流水线结构来实现高速率的 CRC-SCL 译码器。

3.1 并行路数的设计

吞吐率是评价译码器性能的重要指标。由于大多数的 FPGA 芯片所支持的时钟频率只能在百兆赫兹级, 而译码时延不足够小时, 译码吞吐率难以达到 10 Gbit/s 量级。事实上, 针对串行算法的硬件实现, 并行化的实现方案可以很大程度地降低时延。因此要实现并行化的极化码译码方案以达到高吞吐率, 首先需要确定并行路数, 而吞吐率和并行路数的关系为

$$T_p = \frac{p_r}{T} = \frac{p_r}{n \times t} = \frac{p_r \times f_{\text{clk}}}{n}, \quad (15)$$

式中: p_r 为并行路数; T 为总时延(包含 n 个时钟, 一个时钟的时延为 t); t 的倒数即为时钟频率 f_{clk} 。当 p_r 越大、 f_{clk} 越大、 n 越小时, 吞吐率越高。

在设计译码器时,应该采用尽可能大的并行路数以实现高吞吐量。同时,并行路数受限于码长的大小,且当码长等于并行路数的整数倍时,基于极化码编译码的蝶形结构的硬件实现会相对简单。因此,本文以并行路数等于码长为例对硬件设计方案进行介绍。

3.2 硬件设计

实际上,高吞吐率的译码器需要在每一个新时钟下接收新的输入帧数据,并且可以连续不断地输出新的译码序列。因此在硬件实现上,为了达到高吞吐率的目标,对 CRC-SCL 译码器采用完全展开式流水线架构,进行此操作主要有如下两个方面原因。一方面,基于逐个比特串行译码的 CRC-SCL 算法无法在一个时钟内得到译码序列(如同一帧下,计算完对数似然比 LLR 之后才能计算路径度量值 PM,接着进行 PM 值排序,最后进行路径筛选,因此至少需要 4 个时钟才能得到译码序列)。为了提高译码器的传输速度,需采用并行化流水线结构,即将输入译码器的每一帧数据分成 N 路进行并行处理,在流水线结构下,根据指令间的并行性,在第 n 帧

译码未完成前就开始第 $n+1$ 帧的译码。这样,译码器连续输入与输出数据,真正达到高速译码的目的。以码长 $N=256$ 的极化码为例,当并行路数等于码长时,即在每个时钟下输入 $N=256$ 个 CLLR 值,然后在经过 1025 个时钟周期后,每个时钟都有新的码字输出。因而并行化流水线结构的译码器在一个时钟下同时存在 1025 帧数据,这些数据帧或是进行不同序号的比特译码,或是处理相同序号的比特但进行 LLR、PM 等不同的译码操作。另一方面,并行化流水线结构使得译码器会同时存在多个帧的译码,为了保证各帧的译码数据不发生冲突,译码过程中的每一个操作都不共用(每个译码操作单独的模块实现),即采用完全展开结构。例如,CRC-SCL 译码器在译码每一个比特时,都需要进行 LLR 与 PM 计算、PM 排序、路径选择和 CRC 校验,那么 256 码长的译码器采用完全展开的架构,需要 LLR 计算、PM 计算、排序等模块的数量均为 256,这样才能够保证译码时无冲突。因此,基于完全展开式流水线架构的 $N=4$ 极化码译码流程如图 2 所示,其中列表数 $L=2$ 。

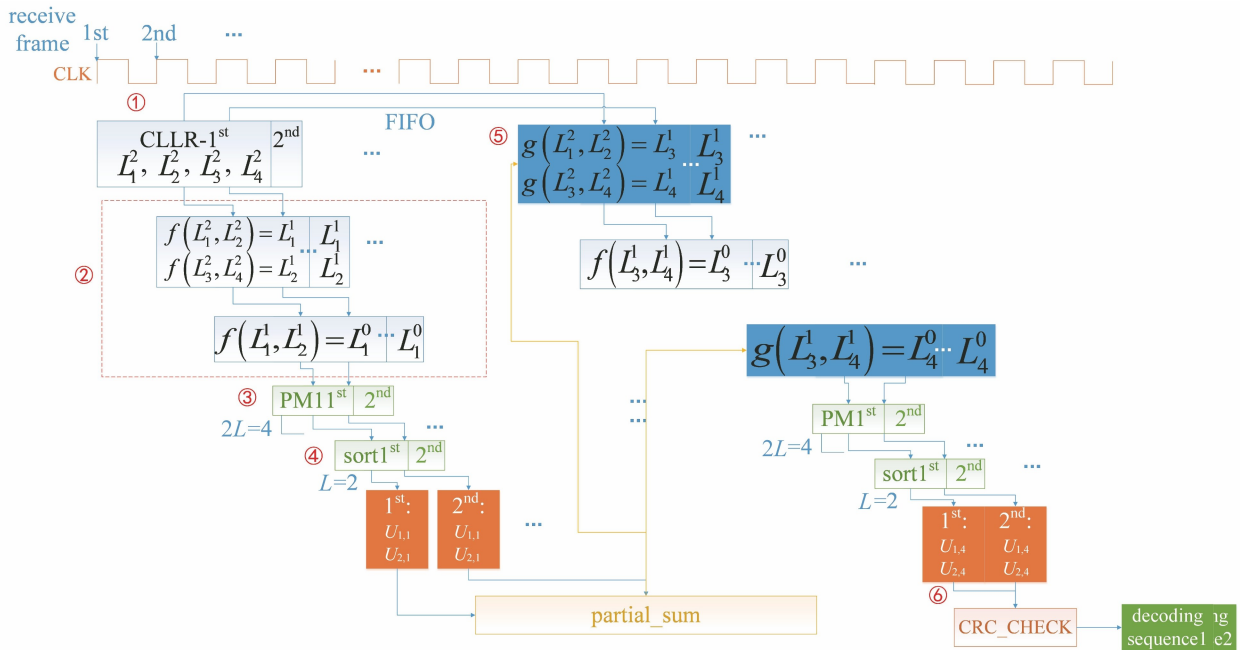


图 2 基于完全展开式流水线架构的 $N=4$ 极化码译码流程图

Fig. 2 Decoding flow chart of $N=4$ polar codes based on fully-unrolled pipeline architecture

由图 2 可知,极化码译码工作流程及设计步骤如下:

1) 在每个时钟的上升沿到来时,译码器接收到 N 个信道的对数似然比 $CLLR(L_1^2, L_2^2, L_3^2 \dots)$,并用 FIFO(First Input First Output)存储器将它们存储

起来。

2) 基于接收到的 CLLR 值,根据译码蝶形图及 f, g 函数,按层级顺序逐层计算各层的 LLR 值,那么有多少层的计算就需要消耗多少个时钟。本设计中预先计算 g 函数^[19],这样可以减少实例化模块的

数量,并缩短译码时间。

3) 接着在下一个时钟使用最后一层的 LLR 值来计算每一条路径对应的 PM 值。

4) 然后在新时钟的上升沿到来时对 PM 值进行排序,并根据排序结果来选择其中 PM 值最小的 L 条路径(图 2 示例中 $L=2$)。从扩展出的 $2L$ 条路径筛选出 L 条之后,如果原始的 L 条路径有些被删除,则保留下来的路径信息会覆盖被删除路径的存储空间,这些信息包括 LLR 值、译码比特序列、CRC 校验和等。使用寄存器存储译码比特序列和 CRC 校验和,二者所占用的存储空间较小,因此采用直接复制的方式,而 LLR 值占用的内存较大,因此选择文献[16]的方法,并使用指针寄存器进行 LLR 值资源的读取与存储。

5) 除了最后一层级的 LLR 值不用存取之外,其余层级的 LLR 值前后共使用两次,因此需要用 FIFO 存储,以达到流水线的目的。根据 FIFO 中读

取的 LLR 值,使用 g 函数计算下一层级的结果,然后再重复步骤 2)~4)中的计算。

6) 估计完 L 条路径的所有比特后,需要对它们进行 CRC 校验,如图 2 中的 CRC-CHECK 模块。从所有能通过 CRC 校验的路径中选择 PM 值最小的输出作为译码序列的输出,至此可获得一帧的译码比特序列。

4 实验及结果分析

基于极化码的编译码原理,将极化码应用于 QPSK 系统实验中,以设计合适的硬件译码器,并使用设计完成的硬件译码器处理实验数据,从而得到该系统的性能结果。下面介绍实验装置并分析实验结果。

4.1 实验装置介绍

图 3 展示了基于极化码的 QPSK 系统的实验装置。

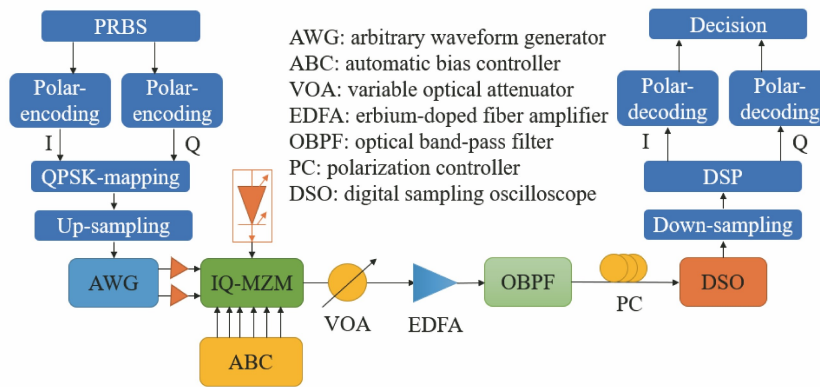


图 3 Polar-QPSK 系统实验装置

Fig. 3 Experimental setup for Polar-QPSK system

在发送端,对 IQ 两路的伪随机信息序列(PRBS)分别进行码长为 256、512 和 1024 的极化码编码,然后将编完码的 IQ 两路进行 QPSK 调制得到 QPSK 信号,这些过程在 MATLAB 中进行。接着将 QPSK 信号通过 64 GSa/s 的任意波形发生器(AWG,Keysight M8195A)产生 28-GBaud(56-Gbit/s)的 QPSK 电信号,使用电放大器(EA)放大信号,将信号送入 IQ 调制器产生光脉冲,然后使用可变光衰减器(VOA)来调节接收端的接收光功率(ROP)。接着光信号被掺铒光纤放大器(EDFA)和光通带滤波器(OBPF)放大,将放大信号送入采样率为 80 GSa/s 的实时数字采样示波器(DSO)中进行采样。最后,在 MATLAB 中对接收信号进行下采样后,使用一系列的数字信号处理算法(DSP)进行信号恢复及判决,这些算法包括正交不平衡补偿、时钟

恢复、频偏与载波相位恢复算法,其中接收端的译码部分分别采用 MATLAB 和 VIVADO 进行处理并得到最终的信号序列。

4.2 Polar-QPSK 系统实验的误码性能分析

接下来分析利用极化码在 QPSK 系统中进行背靠背实验的结果。本次实验中采用的极化码码长有 256、512 和 1024,码率分别有 1/2、2/3 和 4/5。在本节中,接收端的译码部分采用 MATLAB 伪代码进行处理,译码算法采用 SC 算法和级联 24 位 CRC 校验码的 CRC-SCL 算法,其中 CRC-SCL 中的列表数 L 设置为 2。图 4 为带有极化码编码的 QPSK 系统的性能曲线,横坐标为接收光功率 ROP,纵坐标是误码率(BER)。

图 4(a)展示了不同码长和码率的极化码通过 CRC-SCL($L=2$)算法译码后的系统性能。在 BER

为 10^{-3} 时,码率为 1/2 的 256、512 和 1024 码长的极化码相较于硬判决,编码增益分别可提高 7.5 dB、7.9 dB 和 8.5 dB,码率为 2/3 时分别获得了 4.5 dB、6.2 dB 和 5.9 dB 的编码增益。而这三种码长的极化码在码率为 4/5 时性能相差不大。由此可以得知,极化码辅助的 QPSK 系统的纠错性能得到了显著提升,且极化码码率越低纠错性能越好,

码长越长对系统性能的改善程度也越好。值得注意的是,SC 算法及其优化后的译码算法 CRC-SCL 是基于前后依赖的关系判决译码各个比特,正是因为这种串行的译码方式,在低 ROP(相当于低信噪比)时,SC 等译码算法容易产生错误决策,进而产生错误传播的现象^[21]。因此,4/5 与 2/3 码率的极化码在低 ROP 时的误码率会超过硬判决的误码率。

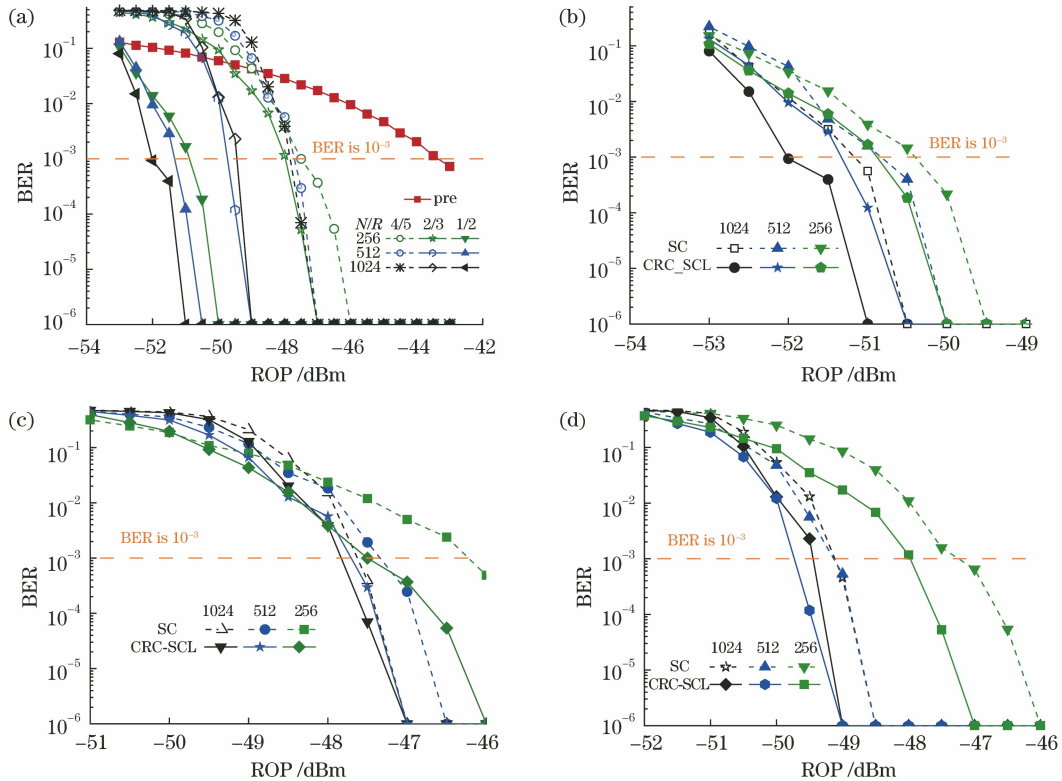


图 4 Polar-QPSK 系统背靠背实验的结果。(a)CRC-SCL 算法的误码率与接收光功率的关系;(b) $R=1/2$ 时 SC 与 CRC-SCL 的性能比较;(c) $R=4/5$ 时 SC 与 CRC-SCL 的性能比较;(d) $R=2/3$ 时 SC 与 CRC-SCL 的性能比较

Fig. 4 Results of back to back (B2B) experiments in polar-QPSK system. (a) Relationship between BER and ROP of CRC-SCL; (b) performance comparison of SC and CRC-SCL when $R=1/2$; (c) performance comparison of SC and CRC-SCL when $R=4/5$; (d) performance comparison of SC and CRC-SCL when $R=2/3$

图 4(b)~(d)中比较了码率(R)分别为 1/2、4/5 和 2/3 时,SC 算法和 CRC-SCL 算法的性能差异。实验结果表明,CRC-SCL 算法优于 SC 算法。如,在 BER 为 10^{-3} 时,1/2、4/5、2/3 码率的 256 码长极化码在 CRC-SCL 算法的译码下分别比同码长的 SC 算法多获得 0.5 dB、1.3 dB、0.7 dB 的编码增益。当 R 为 1/2、4/5 时,CRC-SCL 算法译码的 256 码长极化码的总体性能优于 SC 算法译码的 512 码长极化码,并且与 SC 译码的 1024 码长极化码译码性能相近。CRC-SCL 算法译码的 2/3 码率、256 码长极化码虽然并未采用最佳的码字构造方法(为了减小硬件实现的复杂度),但其性能仍好于相同码长的 SC 算法,得到的译码的 2/3 码率、256 码长极化

码性能仅稍低于 SC 算法译码的 1024 码长极化码的性能。综上所述,CRC-SCL 算法对 256 码长的极化码进行译码时,在多种码率下均能获得极具竞争力的性能优势。

由于码长越长,CRC-SCL 硬件译码器占用的资源量会越大,目前报道的单块 FPGA 芯片的性能不能满足较长码长的 CRC-SCL 译码器的资源量需求。虽然 CRC-SCL ($L=2$) 算法的复杂度稍高于 SC 算法,但短码长可降低消耗资源量。因此在本文中设计码长为 256 的 CRC-SCL 硬件译码器,以实现高速高性能的实时光通信系统的译码。

4.3 并行化 CRC-SCL 译码器的实现及性能分析

设计的 CRC-SCL 译码器的码长 $N=256$,列表

数 $L=2$, CRC 校验的比特数为 24。为了获得高吞吐率,码长较短的极化码译码器的并行路数应等于 N ,即将本文译码器的并行路数设计为 256。基于上述硬件设计思路,完成了基于 Xilinx xcvu13p-flga2577-1-e FPGA 芯片的 256 码长 CRC-SCL 译码器的 Verilog 编程。但译码器的性能除了受码长、码率的影响,还受量化位数的影响。定点量化会导致一定的纠错性能损失,通过有限的精度可减少计算数据的溢出,以保证 CRC-SCL 算法的性能优越性。由前文可知,CRC-SCL 译码器的运算数值主要是 LLR 和 PM,所以定点量化位数用 $Q(Q_C, Q_I, Q_{PM}, Q_r)$ 表示,其中 Q_C 表示 CLLR 值的位数, Q_I 为内部 LLR 值位数, Q_{PM} 是路径度量值 PM 位数, Q_r 为 Q_C 和 Q_{PM} 的小数位数。

将量化位数分别为 $Q(8, 13, 15, 2)$ 、 $Q(8, 12, 12, 2)$ 、 $Q(8, 8, 12, 2)$ 和 $Q(6, 6, 12, 1)$ 的并行化 CRC-SCL 译码算法的 Verilog 伪代码,经过集成设计环境 VIVADO 的综合和布局布线后,得到了它们占用 FPGA 板的硬件资源量以及可满足的最大时钟频率。表 1 列出了 4 种量化方案对应的译码器的资源消耗量。其中第一列为内部 LLR 和路径度

量值 PM 定点量化的位数,随着位数的减少,LUT 和 RAM 的数量都显著减少,而吞吐率保持不变。表中的 LUTs 和 Regs 分别表示查找表和寄存器的个数,而这两列对应的括号中数值分别表示译码器所使用的 LUTs 和 Regs 个数占总数目的百分比。从表中可知,这 4 种译码器可达到的最大时钟频率为 156.25 MHz。由于译码器的并行路数等于码长 N ,每个时钟都有新的译码帧输出(即 $n=1$),因此由(15)式得到这些译码器的总吞吐率(译码器输出的译码序列包括固定比特和信息比特)达到了 40 Gbit/s。而每次输出的信息比特数为 NR 与 CRC 的差值,因此当 $R=1/2$ 时信息吞吐率(输出的译码序列只包括信息比特)为 $(256 \times 0.5 - 24) \times 156.25 = 16.25$ Gbit/s。结果表明,完全展开式的并行化流水线架构可以改善串行算法带来的大延迟问题,从而使本文设计的 CRC-SCL 译码器的吞吐率获得了显著的提升。进一步地,若想继续提升吞吐率,需要使用更高级的芯片来支持更大的时钟频率。当然还可以对 CRC-SCL 这种串行译码算法向 Fast-SSC-List 算法进行优化,但需要解决算法的译码性能较差、通用性不佳的问题。

表 1 量化对 $N=256$ CRC-SCL 译码器硬件资源的影响

Table 1 Effect of quantization on hardware resource of CRC-SCL decoder with $N=256$

Bit number of LLR	Bit number of PM	LUTs	Regs	BRAM /bit	f /MHz	Throughput / (Gbit·s ⁻¹)
13	15	435390(25%)	628883(18%)	648(24%)	156.25	40
12	12	417975(24%)	606723(18%)	631(23%)	156.25	40
8	12	332720(19%)	562752(15%)	451(17%)	156.25	40
6	12	292256(17%)	537807(15%)	402(15%)	156.25	40

定点量化位数除了影响译码器的硬件资源消耗量以外,还会影响纠错性能。因此使用这 4 种不同量化方案的译码器来处理 polar-QPSK 系统实验的译码部分,以获取它们的纠错性能。使用两个本文设计的 CRC-SCL 译码器对实验的 IQ 两路分别进行译码处理,那么在 156.25 MHz 的最大时钟频率下,总吞吐率可达 80 Gbit/s,而 56-Gbit/s polar-QPSK 系统背靠背实验只需译码器的时钟频率为 109.375 MHz 即可。通过 VIVADO 和 MODELSIM 硬件仿真得到 4 种量化方案的译码性能对比。图 5 绘制了这 4 种方案的 BER 与 ROP 的曲线,从图中可以看出,前三种量化方案的纠错性能并没有相差很大。这是由于前三种方案对译码过程中发生的溢出采用截断处理,使得位宽不会逐级递增,大大简化了译码器的设计复杂度,且几乎没有性能损失。但

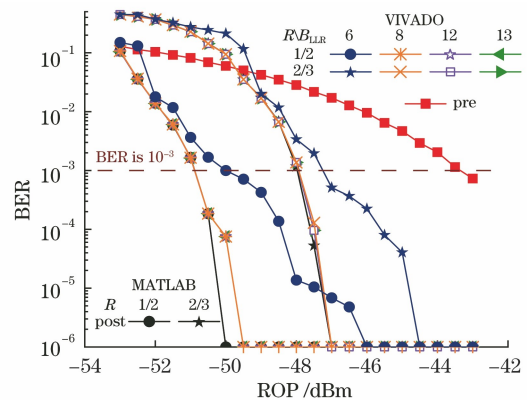


图 5 量化对 $N=256$ CRC-SCL 译码器纠错性能的影响

Fig. 5 Effect of quantization on error-correction performance of CRC-SCL decoder with $N=256$

当 LLR 值量化为 6 bit 时译码器的纠错性能下降,在 BER 为 10^{-3} 时,处理 1/2 码率的极化码的误码

率损失了近 1 dBm 的编码增益。图 5 中 B_{LLR} 表示 BLLR 值量化的位数。

4.4 与其他译码器的比较

在与其他译码器比较之前,通过 VIVADO 仿真,给出了本文设计的 CRC-SCL 译码器在高斯白噪声 (AWGN)信道下的误码率性能结果,如图 6 所示,横坐标为比特信噪比(E_b/N_0 ,其中 E_b 为信号强度, N_0 为噪声强度),纵坐标为 BER。该译码器可以对码率为 1/2、2/3 和 4/5 的极化码进行译码。当极化码码率为 1/2 时,在 BER 为 10^{-3} 处,译码器的比特信噪比为 2.8 dB。而当码率为 2/3 时,译码器的信息吞吐率为 22.8125 Gbit/s。同理,4/5 码率时译码器的信息吞吐率可达到 28.125 Gbit/s。由此可知,本文设计的 CRC-SCL 译码器可以适配不同码率,不失灵活性,当码率增大时虽纠错性能变差,但吞吐率有较大提升,并且该硬件架构具有普适性,适用于各种码长。

下面展示本文工作与当前报道的最快极化码译码器的比较。由于在文献中没有找到吞吐率超过 10 Gbit/s 的 CRC-SCL 译码器,因此在表 2 中,选择吞吐率能达到 Gbit/s 量级且比较有代表性的译码器进行比较。为公平起见,表中列出的吞吐率是信

表 2 高吞吐率的译码器的比较

Table 2 Comparison of high throughput polar decoders

Method	This work	Ref. [10]	Ref. [14]	Ref. [15]
Algorithm	CRC-SCL	CRC-SCL($L=2$)	SC	Fast-SSC-List
Length & Rate	256 & 0.5	1024 & 0.5	1024 & 0.5	512 & 0.83
IC type	FPGA	ASIC (40 nm)	ASIC (28 nm)	ASIC (28 nm)
Frequency /MHz	156.25	430	621	468
Throughput /($\text{Gbit}\cdot\text{s}^{-1}$)	16.25	3.25	318.00	12.00
$\frac{E_b}{N_0} / (10^{-3} \text{ dB})$	2.8	2.3	2.8	4.3

5 结 论

主要研究了高速并行化的 CRC-SCL 译码器,重点分析了译码器的纠错性能与硬件资源量的使用情况。在 Xilinx xcvu13p-flga2577-1-e 芯片上的结果表明,所提出的 CRC-SCL 译码器在 156.25 MHz 的时钟频率下,能够达到 40 Gbit/s 的总吞吐率,这是目前最快速的 CRC-SCL 极化码译码器。通过 polar-QPSK 背靠背实验展示了该译码器的纠错性能。在 256 码长下,对数似然比 LLR 量化为 8 bit、路径度量值 PM 量化为 12 bit 时,译码器能够在较低的硬件资源使用率(LUT 为 19%,BRAM 为 17%)

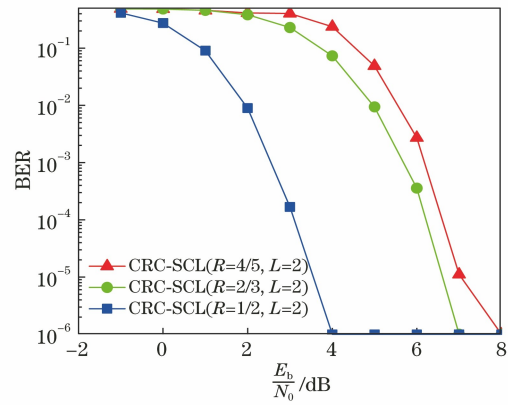


图 6 AWGN 信道上 $N=256$ 的 CRC-SCL 译码器的仿真结果

Fig. 6 Simulation results for CRC-SCL decoder with

$N=256$ over AWGN channel

息吞吐率。表 2 显示,相比于文献[10]中的 CSCL 译码器,本文方法的吞吐率提高了一个量级。相比于有相似吞吐率的 Fast-SSC-List 译码器,本文结构在比特信噪比方面提升了 1.5 dB。而相较于文献[14]中的 SC 译码器,本文方法与 CRC-SCL 算法结合的译码器可以达到中码长极化码的性能,虽其吞吐率低于 SC 译码器一个量级,但却能适配多种码率的极化码,因而无需设计多套架构即可实现不同码率的译码器,更具有灵活性。

上保持优异的纠错性能(误码率在 10^{-3} 处获得 7.5 dB 的编码增益)。与此同时,所设计的译码器可以对多种码率的极化码进行译码,不失灵活性。该研究验证了高速 CRC-SCL 译码器硬件实现的可行性,为高速光通信系统的实际应用提供了重要的参考。

参 考 文 献

[1] Arikan E. Channel polarization: a method for constructing capacity-achieving codes[C]//2008 IEEE International Symposium on Information Theory, July 6-11, 2008, Toronto, ON, Canada. New York: IEEE Press, 2008: 1173-1177.

- [2] Cao Y, Li Y, Li X H. Research on construction method of polarization code in wireless optical communication[J]. *Acta Optica Sinica*, 2020, 40(21): 2106003.
曹阳, 李岳, 李小红. 无线光通信中极化码构造方法研究[J]. *光学学报*, 2020, 40(21): 2106003.
- [3] Tal I, Vardy A. List decoding of polar codes [J]. *IEEE Transactions on Information Theory*, 2015, 61(5): 2213-2226.
- [4] 3GPP. "RAN1 Chairman's Notes" 3GPP TSG RAN WG1 Meeting 87[EB/OL]. (2016-11-01)[2021-07-18]. <https://www.3gpp.org>.
- [5] Niu K, Chen K. CRC-aided decoding of polar codes [J]. *IEEE Communications Letters*, 2012, 16(10): 1668-1671.
- [6] Niu K, Chen K, Lin J R. Beyond turbo codes: rate-compatible punctured polar codes [C]//2013 IEEE International Conference on Communications (ICC), June 9-13, 2013, Budapest, Hungary. New York: IEEE Press, 2013: 3423-3427.
- [7] 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.
- [8] 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.
- [9] Yuan B, Parhi K K. Low-latency successive-cancellation list decoders for polar codes with multibit decision[J]. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2015, 23(10): 2268-2280.
- [10] Tao Y Y, Cho S G, Zhang Z Y. A configurable successive-cancellation list polar decoder using split-tree architecture [J]. *IEEE Journal of Solid-State Circuits*, 2021, 56(2): 612-623.
- [11] Xia C Y, Fan Y Z, Tsui C Y. High throughput polar decoding using two-staged adaptive successive cancellation list decoding [EB/OL]. (2019-05-22) [2021-02-01]. <https://arxiv.org/abs/1905.09120>.
- [12] Chen W, Song Y X, Li Z X, et al. 50 Gbit/s NRZ IM-DD downstream transmission system based on 25G-class optical components [J]. *Acta Optica Sinica*, 2019, 39(6): 0606003.
陈炜, 宋英雄, 李正璇, 等. 基于 25G 级光器件的 50 Gbit/s NRZ IM-DD 下行传输系统[J]. *光学学报*, 2019, 39(6): 0606003.
- [13] Luo J W, Liu K, Wei Q, et al. Integrated chip for simultaneous transmission and reception in optical communication links [J]. *Acta Optica Sinica*, 2019, 39(8): 0806003.
罗俊伟, 刘凯, 位祺, 等. 光通信链路中集成芯片的收发一体工作[J]. *光学学报*, 2019, 39(8): 0806003.
- [14] Lehnigk-Emden T, Alles M, Kestel C, et al. Polar code decoder framework [C]//2019 Design, Automation & Test in Europe Conference & Exhibition (DATE), March 25-29, 2019, Florence, Italy. New York: IEEE Press, 2019: 1208-1209.
- [15] Giard P, Balatsoukas-Stimming A, Müller T C, et al. A multi-Gbps unrolled hardware list decoder for a systematic polar code [C]//2016 50th Asilomar Conference on Signals, Systems and Computers, November 6-9, 2016, Pacific Grove, CA, USA. New York: IEEE Press, 2016: 1194-1198.
- [16] 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.
- [17] Leroux C, Raymond A J, Sarkis G, et al. Asemi-parallel successive-cancellation decoder for polar codes [J]. *IEEE Transactions on Signal Processing*, 2013, 61(2): 289-299.
- [18] 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 (ICASSP), May 22-27, 2011, Prague, Czech Republic. New York: IEEE Press, 2011: 1665-1668.
- [19] Zhang C, Yuan B, Parhi K K. Reduced-latency SC polar decoder architectures [C]//2012 IEEE International Conference on Communications (ICC), June 10-15, 2012, Ottawa, ON, Canada. New York: IEEE Press, 2012: 3471-3475.
- [20] Kestel C, Weithoffer S, Wehn N. Polar code decoder exploration framework [J]. *Advances in Radio Science*, 2018, 16: 43-50.
- [21] Afisiadis O, Balatsoukas-Stimming A, Burg A. A low-complexity improved successive cancellation decoder for polar codes [C]//2014 48th Asilomar Conference on Signals, Systems and Computers, November 2-5, 2014, Pacific Grove, CA, USA. New York: IEEE Press, 2014: 2116-2120.