

说明书

一种数据处理方法和数据处理装置

本申请要求于 2023 年 03 月 31 日提交国家知识产权局、申请号为 202310386404.8、发明名称为“一种数据处理方法和数据处理装置”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本申请涉及光通信领域，尤其涉及一种数据处理方法和数据处理装置。

背景技术

在 5G、云计算、大数据和人工智能等持续推动下，光通信系统及光传输网（optical transport network, OTN）正朝着大容量和超高速方向发展。采用前向纠错编码（forward error correction, FEC）对传输的数据进行纠错，能够解决传输误码，从接收数据中恢复出发送端发送的原始数据。

当前提出有一种级联 FEC 的传输方案，发端设备和发端处理模块通过连接单元接口（attachment unit interface, AUI）连接。发端设备对待传输数据进行第一 FEC 编码，并将第一 FEC 编码后的数据发送至发端处理模块。发端处理模块对第一 FEC 编码后的数据再进行第二 FEC 编码，并将第二 FEC 编码后的数据通过信道传输至数据接收端。具体地，发端处理模块会收到多路数据流，要分别先对多路数据流先进行卷积交织，再对卷积交织后的每一路数据流进行第二 FEC 编码。然而，目前的方案中卷积交织需要较大的时延才可以达到较好的性能，在要求低时延的场景下应用效果不理想。

发明内容

本申请实施例提供了一种数据处理方法及数据处理装置，可以在采用较低时延的场景下获得较好的级联 FEC 方案性能。

第一方面，本申请实施例提供了一种数据处理方法，该方法包括如下步骤。首先，获取 m 条第一数据流， m 为大于 1 的整数。每条第一数据流由经过第一 FEC 编码后的 v 路第一码字交织得到，每条第一数据流包括连续的多个比特集合。每个比特集合包括 t 个比特子集， t 为大于 1 的整数。每个比特子集包括 d 个比特， $d=v \times s$ ， v 为大于 1 的整数， s 表示第一码字中每个符号包括的比特数， s 为大于或等于 1 的整数。每个比特子集中的比特来自 v 个符号， v 个符号分别来自 v 个第一码字， v 个第一码字分别来自 v 路第一码字。进而，根据 p 条延迟线对 1 条第一数据流进行延迟得到 1 条第二数据流，以得到共 m 条第二数据流。 p 为大于 1 的整数，每条延迟线包括的存储单元数量各不相同，存储单元数量最小的延迟线包括 0 个存储单元。每相邻两条延迟线的存储单元数量的差值为 Q ， Q 为大于或等于 1 的整数，每个存储单元用于存储 d 个比特。每条第一数据流中的比特按照 p 条延迟线的序号依次输入 p 条延迟线，每条延迟线单次输入 1 个比特子集中的 d 个比特且单次输出 d 个比特。

在该实施方式中，针对采用级联 FEC 的传输方案，在对第一数据流进行卷积交织之前需要先进行交织对准，以确定第一数据流中各比特集合的边界以及比特集合中各比特子集的边界。进而，以比特子集为粒度送入卷积交织操作的各存储单元，即每个比特子集包括 d 个比特，每个存储单元用于存储 d 个比特。其中，每个比特子集中的比特来自 v 个符号，且这 v 个符号分别来自 v 个第一码字。也就是说， v 个第一码字中每个第一码字各取 1 个符号构成比特子集。配合后续的卷积交织可以采用最少的交织延迟使得一个完整的外码码字符号映射到一个内码码字，在内码码字不可纠的情况下导致其关联的外码码字最多错一个符号，避免错误分散到外码码字的多个符号中，提升了级联码的性能。

在一些可能的实施方式中，每条第二数据流中连续的 $p \times d$ 个比特来自 $v \times p$ 个第一码字。后续对每条第二数据流中的每连续的 $p \times d$ 个比特进行内码编码得到一个内码码字，可以使得每个内码码字的比特映射到最大数目的外码码字，使得级联码的性能达到最优。

在一些可能的实施方式中， p 条延迟线中序号最大的延迟线包括 0 个存储单元， $(p \times Q + 1) \geq t$ 。提供了一种使级联码性能达到最优的具体实现方案，增强了本方案的可实现性。

在一些可能的实施方式中， p 条延迟线中序号最小的延迟线包括 0 个存储单元， $(p \times Q - 1) \geq t$ 。提供了另一种使级联码性能达到最优的具体实现方案，增强了本方案的灵活性。

在一些可能的实施方式中， Q 为 2 的倍数，便于将卷积交织器进行低频率时钟实现。

在一些可能的实施方式中， m 条第一数据流由 n 条通道数据流复用得到， n 为 m 的整数倍，每条通道数据流经过第一 FEC 编码。每条通道数据流包括多个对齐标识 (Alignment marker, AM)，每条第一数据流包括多个 AM 集合，每个 AM 集合包括 n/m 个 AM，每相邻两个 AM 集合的起始位置之间包括整数个比特集合。

5 在一些可能的实施方式中，获取 m 条第一数据流包括：根据至少一个 AM 集合对第一数据流进行交织对准，以确定第一数据流中每个比特集合的边界位置。也就是说，在第一数据流中通过识别 AM 集合来确定比特集合的边界位置，并且由于每个比特集合的长度是固定的，在确认了其中一个比特集合边界的基础上也就确定了其他每个比特集合的边界。进一步，基于确定位置的各比特集合，也就可以确定比特集合中每个比特子集的边界。

10 在一些可能的实施方式中，多个比特集合包括相邻的第一比特集合和第二比特集合，第一比特集合中的比特来自第一 FEC 编码后的其中 v 个第一码字，第二比特集合中的比特来自第一 FEC 编码后的另外 v 个码字。

在一些可能的实施方式中，比特子集中每连续的 v 个比特分别来自 v 个第一码字，或者，比特子集中每连续的 s 个比特来自同一个第一码字。

15 在一些可能的实施方式中， $m=8$ ， $v=4$ ， $t=68$ ， $s=10$ ， $p=3$ ， $Q=23$ 或 24；或者， $m=4$ ， $v=4$ ， $t=136$ ， $s=10$ ， $p=3$ ， $Q=46$ 。

在一些可能的实施方式中，得到共 m 条第二数据流之后，方法还包括：对 m 条第二数据流分别进行第二 FEC 编码得到 m 条第三数据流，第二 FEC 编码后的每个第二码字中的 K 个信息比特最多来自 $p \times v$ 个不同的第一码字， K 为 $p \times d$ 的整数倍。

20

第二方面，本申请实施例提供了一种数据处理装置，该数据处理装置包括：处理单元和卷积交织器。处理单元用于：获取 m 条第一数据流， m 为大于 1 的整数，每条第一数据流由经过第一 FEC 编码后的 v 路第一码字交织得到。每条第一数据流经过第一 FEC 编码，每条第一数据流包括连续的多个比特集合，每个比特集合包括 t 个比特子集， t 为大于 1 的整数。每个比特子集包括 d 个比特， $d=v \times s$ ， v 为大于 1 的整数， s 表示第一码字中每个符号包括的比特数， s 为大于或等于 1 的整数。每个比特子集中的比特来自 v 个符号， v 个符号分别来自 v 个第一码字， v 个第一码字分别来自 v 路第一码字。卷积交织器用于：根据 p 条延迟线对 1 条第一数据流进行延迟得到 1 条第二数据流，以得到共 m 条第二数据流。 p 为大于 1 的整数，每条延迟线包括的存储单元数量各不相同，存储单元数量最小的延迟线包括 0 个存储单元。每相邻两条延迟线的存储单元数量的差值为 Q ， Q 为大于或等于 1 的整数，每个存储单元用于存储 d 个比特。每条第一数据流中的比特按照 p 条延迟线的序号依次输入 p 条延迟线，每条延迟线单次输入 1 个比特子集中的 d 个比特且单次输出 d 个比特。

25

在一些可能的实施方式中，每条第二数据流中连续的 $p \times d$ 个比特来自 $v \times p$ 个第一码字。

在一些可能的实施方式中， p 条延迟线中序号最大的延迟线包括 0 个存储单元， $(p \times Q + 1) \geq t$ 。

在一些可能的实施方式中， p 条延迟线中序号最小的延迟线包括 0 个存储单元， $(p \times Q - 1) \geq t$ 。

35

在一些可能的实施方式中， Q 为 2 的倍数。

在一些可能的实施方式中， m 条第一数据流由 n 条通道数据流复用得到。 n 为 m 的整数倍，每条通道数据流经过第一 FEC 编码。每条通道数据流包括多个 AM，每条第一数据流包括多个 AM 集合，每个 AM 集合包括 n/m 个 AM，每相邻两个 AM 集合的起始位置之间包括整数个比特集合。

40

在一些可能的实施方式中，处理单元具体用于：根据至少一个 AM 集合对第一数据流进行交织对准，以确定第一数据流中每个比特集合的边界位置。

在一些可能的实施方式中，多个比特集合包括相邻的第一比特集合和第二比特集合，第一比特集合中的比特来自第一 FEC 编码后的其中 v 个第一码字，第二比特集合中的比特来自第一 FEC 编码后的另外 v 个码字。

45

在一些可能的实施方式中，比特子集中每连续的 v 个比特分别来自 v 个第一码字，或者，比特子集中每连续的 s 个比特来自同一个第一码字。

在一些可能的实施方式中， $m=8$ ， $v=4$ ， $t=68$ ， $s=10$ ， $p=3$ ， $Q=23$ 或 24；或者， $m=4$ ， $v=4$ ， $t=136$ ，

$s=10$, $p=3$, $Q=46$ 。

在一些可能的实施方式中, 数据处理装置还包括编码器, 编码器用于:

对 m 条第二数据流分别进行第二 FEC 编码得到 m 条第三数据流, 第二 FEC 编码后的每个第二码字中的 K 个信息比特最多来自 $p \times v$ 个不同的第一码字, K 为 $p \times d$ 的整数倍。

5 附图说明

- 图 1 为本申请实施例应用的一种通信系统示意图;
图 2 为图 1 所示通信系统中一种数据传输的过程示意图;
图 3 为本申请实施例中发端处理模块的一种数据处理示意图;
图 4 为本申请实施例提供的数据处理方法的一种流程示意图;
10 图 5 为本申请实施例中第一数据流的第一种结构示意图;
图 6 为本申请实施例中第一数据流的第二种结构示意图;
图 7 为本申请实施例中第一数据流的第三种结构示意图;
图 8 为本申请实施例中第一数据流的第四种结构示意图;
图 9 为本申请实施例中对 m 条第一数据流分别进行卷积交织的示意图;
15 图 10 (a) 为本申请实施例中卷积交织器的第一种结构示意图;
图 10 (b) 为本申请实施例中卷积交织器的第二种结构示意图;
图 11 为本申请实施例中对 m 条第二数据流进行第二 FEC 编码的示意图;
图 12 为发端设备采用 $1 \times 800G$ 接口的 32 条 PCS 通道数据流的示意图;
图 13 为本申请实施例中第一数据流的第五种结构示意图;
20 图 14 为本申请实施例中第一数据流的第六种结构示意图;
图 15 为本申请实施例中卷积交织器的第三种结构示意图;
图 16 为本申请实施例中卷积交织器的第四种结构示意图;
图 17 为本申请实施例中卷积交织器的第五种结构示意图;
图 18 为本申请实施例中第一数据流的第七种结构示意图;
25 图 19 为本申请实施例中卷积交织器的第六种结构示意图;
图 20 为 1600GbE 场景下 16 条 PCS 通道数据流的示意图;
图 21 为本申请实施例中第一数据流的第八种结构示意图;
图 22 为本申请实施例中发端处理模块的另一种数据处理示意图;
图 23 为本申请实施例中对第一数据流进行复用的示意图;
30 图 24 为本申请实施例中数据处理装置的一种结构示意图;
图 25 为本申请实施例中数据处理装置的另一种结构示意图。

具体实施方式

- 本申请实施例提供了一种数据处理方法及数据处理装置, 可以在采用较低时延的场景下获得较好的级联 FEC 方案性能。需要说明的是, 本申请说明书和权利要求书及上述附图中的术语“第一”、“第二”
35 等用于区别类似的对象, 而非限定特定的顺序或先后次序。应该理解, 上述术语在适当情况下可以互换, 以便在本申请描述的实施例能够以除了在本申请描述的内容以外的顺序实施。此外, 术语“包括”和“具有”以及他们的任何变形, 意图在于覆盖不排他的包含。例如, 包含了一系列步骤或单元的过程、方法、系统、产品或设备不必限于清楚地列出的那些步骤或单元, 而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步骤或单元。

- 40 图 1 为本申请实施例应用的一种通信系统示意图。如图 1 所示, 该通信系统包括发端设备 01、发端处理模块 02、信道传输媒介 03、收端处理模块 04 和收端设备 05。以该通信系统是数据中心网络为例, 发端设备 01 和收端设备 05 可以为交换机或路由器等设备, 且发端设备 01 也称为位于发端的客户侧芯片 (host chip), 收端设备 05 也称为位于收端的客户侧芯片, 信道传输媒介 03 可以为光纤。客户侧芯片有时也称为为客户侧设备 (host device)。其中, 发端设备 01 与发端处理模块 02 之间可以通过
45 连接单元接口 (attachment unit interface, AUI) 连接, 收端设备 05 与收端处理模块 04 之间可以通过 AUI 连接。发端处理模块 02 和收端处理模块 04 可以为光模块 (optical module)、电模块、连接器

(connector) 或其他在数据发送过程中对数据进行处理模块。例如, 例如, 该处理模块可以为 800GE FR 模块。并且, 该通信系统中的发端设备 01、发端处理模块 02、信道传输媒介 03、收端处理模块 04 和收端设备 05 均可以支持双向传输, 也可以支持单向传输, 具体此处不做限定。

图 2 为图 1 所示通信系统中一种数据传输的过程示意图。如图 2 所示, 在从发端设备 01 向收端设备 05 传输数据的过程中, 发端设备 01 用于对该数据进行外码编码, 然后向发端处理模块 02 传输经过外码编码的数据。发端处理模块 02 用于对经过外码编码的数据进行内码编码, 得到经过外码编码和内码编码的数据, 并将经过外码编码和内码编码的数据传输至信道传输媒介 03。信道传输媒介 03 用于将经过外码编码和内码编码的数据传输至收端处理模块 04。收端处理模块 04 用于对经过外码编码和内码编码的数据进行内码译码, 并向收端设备 05 传输经过内码译码的数据。收端设备 05 用于对经过内码译码的数据进行外码译码。

应理解, 内码中的“内”和外码中的“外”只是基于对数据进行操作的执行主体相对于信道传输媒介 03 的距离的远近来区分的。对内码进行操作的执行主体较靠近信道传输媒介, 对外码进行操作的执行主体较远离信道传输媒介。在本申请实施例, 由于数据从发端设备 01 发出后经过发端处理模块 02 传输至信道传输媒介 03, 然后从信道传输媒介 03 经过收端处理模块 04 传输至收端设备 05。经发端设备 01 编码的数据相对于经发端处理模块 02 编码的数据离信道传输媒介 03 较远, 经收端设备 05 译码的数据相对于经收端处理模块 04 译码的数据离信道传输媒介 03 较远。因此经发端设备 01 编码的数据称为经过外码编码的数据, 经发端处理模块 02 编码的数据称为经过内码编码的数据, 经收端设备 05 译码的数据称为经过外码译码的数据, 经收端处理模块 04 译码的数据称为经过内码译码的数据。在一种可能的实施方式中, 上述的内码编码和外码编码都是采用 FEC 编码的方式, 从而形成一种级联 FEC 的传输方案。例如, 发端设备 01 可以采用 RS 码进行外码编码, 发端处理模块 02 可以采用汉明 (Hamming) 码进行内码编码。又例如, 发端设备 01 可以采用 RS 码进行外码编码, 发端处理模块 02 可以采用博斯-查德胡里-霍昆格姆 (Bose - Chaudhuri - Hocquenghem, BCH) 码进行内码编码。

需要说明的是, 以上内容是对本申请实施例提供的数据处理方法的应用场景的示例性说明, 并不构成对于数据处理方法的应用场景的限定, 本领域普通技术人员可知, 随着业务需求的改变, 其应用场景可以根据应用需求进行调整, 本申请实施例对其不做一一列举。

对于上述采用级联 FEC 的传输方案, 本申请设计一种包括交织对准和卷积交织的数据处理方法, 使得该级联 FEC 有极低的时延和较优性能, 可以满足加多的传输场景, 尤其适用于对需要较低传输时延的传输场景, 如低时延数据中心内部互联场景。其中, 该数据处理方法是通过上述的发端处理模块 02 来实现的。

图 3 为本申请实施例中发端处理模块的一种数据处理示意图。如图 3 所示, 发端处理模块接收来自 AUI-m 接口的数据, 其中, AUI-m 接口包括 m 条数据通道, m 为大于 1 的正整数, m 条数据通道分别用于传输 m 条第一编码数据流。应理解, 每条第一编码数据流都经过外码编码。作为一个示例, 发送设备 01 进行外码编码得到 n 条物理编码子层 (Physical Coding Sublayer, PCS) 通道数据流或 FEC 通道数据流, 可以简称为 n 条通道数据流, 且每条通道数据流由 u 路第一 FEC 码字交织得到。对 n 条通道数据流进行交织或复用得到 m 条第一编码数据流, 且每条第一数据流由 v 路第一 FEC 码字交织得到, 且 v 能被 u 的整除。首先, 对 m 条第一编码数据流进行时钟恢复 (Clock and Data Recovery, CDR), 均衡以及解调等接收处理。之后, 对 m 条第一编码数据流再依次分别进行交织对准、卷积交织 (convolutional interleaving) 和内码编码得到 m 条第二编码数据流。进而, 经过内码编码的数据流进行数据处理后送入信道传输媒介进行传输, 该数据处理可包括调制映射 (mapping) 和信道交织 (channel interleaving) 等。

需要说明的是, 卷积交织操作具体将输入的比特打乱顺序后送入内码编码器进行内码编码。交织对准具体是在每条第一编码数据流中按照一定要求划分边界, 并基于这种划分方式将第一编码数据流中的比特分批次送入卷积交织器的存储单元, 以使得每批次送入卷积交织器中存储单元的比特序列满足设计要求, 下文会对该设计要求进行详细介绍。也就是说, 交织对准的操作可以看作是对卷积交织做的准备, 保证卷积交织是在本申请所设计的数据对准的前提下进行的。

图 4 为本申请实施例提供的数据处理方法的一种流程示意图,该数据处理方法具体可以由发端处理模块 02 执行。

101、获取 m 条第一数据流。

本实施例中, m 条第一数据流都经过第一 FEC 编码, 即外码编码, 其中, m 为大于 1 的整数。需要说明的是, m 条第一数据流由 n 条通道数据流复用得到, 每条通道数据流经过第一 FEC 编码, n 为 m 的整数倍, 第一 FEC 编码后形成的码字称之为第一 FEC 码字。在一种可能的实施方式中, 发端设备 01 对待发送的数据周期性插入对齐标识 (Alignment marker, AM), 并进行第一 FEC 编码。然后进行 u 路第一 FEC 码字交织后将第一 FEC 码字分配到 n 条 PCS 通道上, 以形成 n 条通道数据流。 n 条通道数据流经过物理媒体附加子层 (Physical Medium Attachment, PMA) 复用后得到 v 路第一 FEC 码字交织的 m 路第一数据流, 其通过 AUI- m 接口发送到数据发送处理模块 02, 其中, AUI- m 接口中的 m 条通道分别用于传输 m 条第一数据流。应理解, 每条第一数据流都是完成了交织对准的数据流, 下面对交织对准后的第一数据流的特征进行介绍。

图 5 为本申请实施例中第一数据流的第一种结构示意图。如图 5 所示, 每条第一数据流包括连续的多个比特集合, 例如, 比特集合 1、比特集合 2、…。每个比特集合包括 t 个比特子集, t 为大于 1 的整数。每个比特子集包括 d 个比特, $d=v \times s$, v 为大于 1 的整数。 s 表示第一 FEC 编码后的第一码字中每个符号包括的比特数, s 为大于或等于 1 的整数, 例如, 当第一 FEC 码字为 RS(544, 514) 时, $s=10$ 。应理解, 每个比特子集中的比特来自 v 个符号, 且这 v 个符号分别来自 v 个第一码字。也就是说, v 个第一码字中每个第一码字各取 1 个符号构成比特子集。应理解, 每条第一数据流由 v 路第一码字交织得到, 且每路第一码字包括多个第一码字。这里的 v 个第一码字分别来自 v 路第一码字, 相当于分别从 v 路第一码字中各取 1 个码字得到 v 个第一码字。因此, 比特集合中连续的 $L=v \times t \times s$ 个比特来自 v 个第一码字。应理解, 相当于来自同一个第一 FEC 码字中的 t 个符号分别位于 t 个比特子集中。还应理解, 连续的 2 个比特集合分别来自不同的第一 FEC 码字, 例如, 比特集合 1 中的比特来自第一 FEC 编码后的其中 v 个第一码字, 比特集合 2 中的比特来自第一 FEC 编码后的另外 v 个第一码字。

需要说明的是, 上述比特集合和比特子集只是为了便于描述而引入的概念, 在实际应用中, 每条第一数据流为一个整体并不存在划分。还需要说明的是, 在一些可能的场景中, 比特集合也可以称之为交织块, 比特子集也可以称之为交织子块, 此处不对其命名做具体约束。

图 6 为本申请实施例中第一数据流的第二种结构示意图。如图 6 所示, 以第一数据流中的比特集合 1 为例, 从比特集合 1 的起始边界开始放置有 AM 集合。应理解, 由于每条通道数据流包括多个 AM, 且 n 条通道数据流经过复用得到了 m 条第一数据流, 因此, 第一数据流的比特集合中的 AM 集合包括 n/m 个 AM, 其由 n/m 个 AM 复用/交织得到, 具体的数据可以由 n 条通道数据流复用为 m 条第一数据流时采用的具体的复用方式以及 AM 的具体的数据得出。也就是说, 本申请不限定 n/m 个 AM 在 AM 集合中的分布方式或排列方式。例如, $n/m=4$, n 条通道数据流采用符号复用的方式得到 m 条第一数据流, AM 集合中第 $4i$ 个符号来自 AM0, AM 集合中第 $4i+1$ 个符号来自 AM1, AM 集合中第 $4i+2$ 个符号来自 AM2, AM 集合中第 $4i+3$ 个符号来自 AM3, i 为整数, 如图 6 所示, 方框中不同图案表示符号来自不同的 AM。又例如, $n/m=4$, n 条通道数据流采用符号复用的方式得到 m 条第一数据流, AM 集合中第 $4i$ 个比特来自 AM0, AM 集合中第 $4i+1$ 个比特来自 AM1, AM 集合中第 $4i+2$ 个比特来自 AM2, AM 集合中第 $4i+3$ 个比特来自 AM3, i 为整数, 如图 6 所示, 方框中不同图案表示比特来自不同的 AM。

还应理解, 第一数据流包括多个 AM 集合, 每相邻两个 AM 集合的起始位置之间包括整数个比特集合。也就是说, 实际应用中, 并不一定是第一数据流中每个比特集合的起始位置都放置有 AM 集合, 可能是在比特集合 1 之后相隔多个比特集合再出现下一个 AM 集合。

需要说明的是, 本申请是以 AM 集合的起始位置来定义比特集合的边界。因此, 交织对准的操作可以理解为根据至少一个 AM 集合确定对应的比特集合的边界。相当于在第一数据流中通过识别 AM 集合来确定比特集合的边界位置, 并且由于每个比特集合的长度是固定的, 在确认了其中一个比特集合边界的基础上也就确定了其他每个比特集合的边界。进一步, 基于确定位置的各比特集合, 也就可以确定比特集合中每个比特子集的边界。

图 7 为本申请实施例中第一数据流的第三种结构示意图。如图 7 所示, 在一种可能的实施方式中,

以比特集合 1 的其中一个比特子集为例, 比特子集中每连续的 v 个比特分别来自 v 个第一码字。图 7 所示的比特子集中每个方框表示 1 个比特, 方框中的不同图案表示该比特来自不同的第一码字, 以图 7 为例 $v=4$ 。

图 8 为本申请实施例中第一数据流的第四种结构示意图。如图 8 所示, 在另一种可能的实施方式中, 以比特集合 1 的其中一个比特子集为例, 比特子集中每连续的 s 个比特来自同一个第一码字。图 7 所示的比特子集中每个方框表示 s 个比特, 即 1 个符号, 方框中的不同图案表示该符号来自不同的第一码字。

102、对 m 条第一数据流分别进行卷积交织得到 m 条第二数据流。

图 9 为本申请实施例中对 m 条第一数据流分别进行卷积交织的示意图。如图 9 所示, 可以通过 m 个卷积交织器分别对 m 条第一数据流进行卷积交织, 每条第一数据流经过卷积交织后可以得到数据顺序打乱的第二数据流。需要说明的是, 在本实施例中, 每个卷积交织器都采用类似的方式对输入的第一数据流进行卷积交织。具体地, 每个卷积交织器包括 p 条延迟线 (delay line), 每个卷积交织器根据 p 条延迟线对输入的第一数据流进行延迟得到第二数据流。其中, p 为大于 1 的整数, 每条延迟线包括的存储单元数量各不相同, 存储单元数量最小的延迟线包括 0 个存储单元, 每相邻两条延迟线的存储单元数量的差值为 Q , Q 为大于或等于 1 的整数。每个存储单元用于存储 d 个比特, 即每个存储单元用于存储 1 个比特子集中的比特。也就是说, p 条延迟线分别对应 p 个延迟取值, 延迟取值包括 0 个比特、 V 个比特、 $2V$ 个比特、 $\dots$ 、 $(p-1)V$ 个比特, 其中, $V=Q \times d$ 。延迟线的延迟取值所包括的比特数量越多, 就说明这条延迟线对数据流的延迟 (也称为时延) 越长。应理解, 当延迟线没有包含存储单元时, 延迟线的延迟为 0 个比特, 即为无延迟透传。

需要说明的是, 每条第一数据流中的比特按照 p 条延迟线的序号依次输入 p 条延迟线, 每条延迟线单次输入 1 个比特子集中的 d 个比特且单次输出 d 个比特。通过选取参数 Q , 可以使得每条第二数据流中连续的 $p \times d$ 个比特来自 $v \times p$ 个第一码字, 后续对每条第二数据流中的每连续的 $p \times d$ 个比特进行内码编码得到一个内码码字, 可以使得每个内码码字的比特映射到最大数目的外码码字, 使得级联码的性能达到最优。

下面结合附图对卷积交织器的具体结构进行介绍。

图 10 (a) 为本申请实施例中卷积交织器的第一种结构示意图。如图 10 (a) 所示, p 条延迟线中存储单元的数量按照 p 条延迟线的序号依次递减。即延迟线 0 具有 $(p-1) \times Q$ 个存储单元, 每一条延迟线依次减少 Q 个存储单元, 延迟线 $p-1$ 具有 0 个存储单元。在该实施方式中, 通过选取参数 p 、 Q 、 d 以满足 $(p \times Q + 1) \geq t$, 使得第二数据流中连续的 $p \times d$ 个比特来自 $v \times p$ 个第一码字。

图 10 (b) 为本申请实施例中卷积交织器的第二种结构示意图。如图 10 (b) 所示, p 条延迟线中存储单元的数量按照 p 条延迟线的序号依次递增。即延迟线 0 具有 0 个存储单元, 每一条延迟线依次增加 Q 个存储单元, 延迟线 $p-1$ 具有 $(p-1) \times Q$ 个存储单元。在该实施方式中, 通过选取参数 p 、 Q 、 d 以满足 $(p \times Q - 1) \geq t$, 使得第二数据流中连续的 $p \times d$ 个比特来自 $v \times p$ 个第一码字。

需要说明的是, 在同一个时刻, 卷积交织器输入和输出的开关 (switch) 位于同一条延迟线上, 待当前的延迟线单次输入 d 个比特并输出 d 个比特后, 再将开关切换到下一条延迟线上, 以保证每条第一数据流中的比特按照 p 条延迟线的序号依次输入到 p 条延迟线, 且第一数据流中连续的 $p \times d$ 个比特包括每条延迟线输出的 d 个比特。其具体数据读出写入操作如下: 从当前延迟线最靠近输出端口的存储单元读出 d 个比特。将当前延迟线中每个存储单元存储的 d 个比特转移至后一个存储单元。再将 d 个比特写入当前延迟线中最靠近输入端口的存储单元。之后, 切换到下一条延迟线并重复上述操作, 依次类推。

应理解, 当采用相同的参数 p 、 Q 和 d 时, 图 10 (a) 的卷积交织处理和图 10 (b) 的卷积交织处理相互是对方的逆操作。也就是说, 当发端处理模块采用图 10 (a) 所示的卷积交织结构, 其收端处理模块对应的卷积解交织采用如图 10 (b) 所示的结构。同理, 当发端处理模块采用图 10 (b) 所示的卷积交织结构, 其收端处理模块对应的卷积解交织采用如图 10 (a) 所示的结构。

还应理解, m 个卷积交织器中的任一个卷积交织器可以采用上述图 10 (a) 或图 10 (b) 中的一种。在实际应用中, m 个卷积交织器可以均采用图 10 (a) 所示的结构; 或者, m 个卷积交织器也可以均采用图 10 (b) 所示的结构; 又或者, 也可以部分的卷积交织器采用图 10 (a) 所示的结构, 剩余部分的卷积交织器采用图 10 (b) 所示的结构。

103、对 m 条第二数据流分别进行第二 FEC 编码得到 m 条第三数据流。

图 11 为本申请实施例中对 m 条第二数据流进行第二 FEC 编码的示意图。如图 11 所示，对 m 条第二数据流分别进行第二 FEC 编码，即上文提到的内码编码。具体地，将每条第二数据流中由 p 个延迟线轮询 w 次输出的连续 $w \times p \times d$ 个比特作为一个内码码字的信息数据，经过内码编码添加冗余数据 M 个比特，得到一个码长为 $N = K + M$ 的内码码字。第二 FEC 编码后的每个第二码字中的 $K = p \times w \times d$ 个信息比特的信息最多来自 $p \times v$ 个不同的第一码字，特别是当 $w=1$ 的时候，级联码字可以达到最好的性能。

需要说明的是，经过内码编码的数据流进行数据处理后送入信道传输媒介进行传输。该数据处理可包括调制映射(mapping)，信道交织(channel interleaving)，偏振划分(polarization distribution)，或 DSP 成帧处理(framing)等。比如内码编码数据流可通过交织后再进行传输，以提升系统抗突发错误的能力。

本申请实施例中，针对采用级联 FEC 的传输方案，在对第一数据流进行卷积交织之前需要先进行交织对准，以确定第一数据流中各比特集合的边界以及比特集合中各比特子集的边界。进而，以比特子集为粒度送入卷积交织操作的各存储单元，即每个比特子集包括 d 个比特，每个存储单元用于存储 d 个比特。其中，每个比特子集中的比特来自 v 个符号，且这 v 个符号分别来自 v 个第一码字。也就是说， v 个第一码字中每个第一码字各取 1 个符号构成比特子集。配合后续的卷积交织可以采用最少的交织延迟使得一个完整的外码码字符号映射到一个内码码字，在内码码字不可纠的情况下导致其关联的外码码字最多错一个符号，避免错误分散到外码码字的多个符号中，提升了级联码的性能。

下面结合一些具体的实施例对上述图 4 所描述的数据处理方法的流程进行进一步介绍。

实施例 1：应用于 $1 \times 800G$ 接口的场景， $m=8$ ， $v=4$ ， $t=68$ ， $s=10$ ， $p=3$ 。

图 12 为发端设备采用 $1 \times 800G$ 接口的 32 条 PCS 通道数据流的示意图。如图 12 所示，当发端设备 01 发送的 $1 \times 800GE$ 业务的时候，一种可能的实现是待发送数据流周期性的插入 AM，再进行 RS 编码（即外码编码或者第一 FEC 编码），然后进行 2 路 RS 交织将码字分配到 32 条 PCS 通道上，使得 32 条 PCS 通道上的 RS 码字排布如图 12 所示。PCS 通道数据流 0-15 由 2 路 RS 码字交织而成，PCS 通道数据流 16-31 由另外 2 路 RS 码字交织而成。例如图 12 所示，PCS 通道数据流 0-15 对应的第一个虚线框表示由 RS 码字 1 和 RS 码字 2 交织得到，PCS 通道数据流 0-15 对应的第二个虚线框表示由 RS 码字 3 和 RS 码字 4 交织得到，其中，RS 码字 1 和 RS 码字 3 来自其中一路 RS 码字，RS 码字 2 和 RS 码字 4 来自另一路 RS 码字，以此类推。同理，PCS 通道数据流 16-31 对应的第一个虚线框表示由 RS 码字 5 和 RS 码字 6 交织得到，PCS 通道数据流 16-31 对应的第二个虚线框表示由 RS 码字 7 和 RS 码字 8 交织得到，其中，RS 码字 5 和 RS 码字 7 来自其中一路 RS 码字，RS 码字 6 和 RS 码字 8 来自同另一路 RS 码字，以此类推。

因此，来自 PCS 通道数据流 0-15 中的任何一个符号与来自 PCS 通道数据流 16-31 中的任何一个符号不会来自同一个 RS 码字。进一步，每条 PCS 通道数据流中相邻 $a=2$ 个符号来自不同 RS 码字，且相邻两条 PCS 通道数据流的同个位置的 2 个符号来自不同 RS 码字，每条 PCS 通道上连续的 68 个符号来自两个不同的 RS 码字。发端设备 01 采用 32:8PMA 将 32 条 PCS 通道数据流复用为 8 条数据流，并通过 800GAUI-8 接口将数据发送到发端处理模块 02。发端处理模块 02 对来自 800GAUI-8 接口的 8 条数据流进行时钟恢复、均衡、解调等接收处理输出 8 条第一数据流。由于在发送设备 01 内每条通道数据流经过双向 RS 交织和 32:8 PMA 比特复用处理，则每条第一数据流被 $v=4$ 路 RS 码字交织。

图 13 为本申请实施例中第一数据流的第五种结构示意图。如图 13 所示，第一数据流中每两个 AM 集合的起始位置之间包括 $L_1=11141120$ 个比特，即 4096 个比特集合，每个比特集合包括 2720 个比特。AM 集合的长度为 PCS 通道数据流中 AM 的长度的 4 倍，以 AM 的长度为 120 比特为例，AM 集合的长度等于 $4 \times 120=480$ 比特。其中，每个比特集合中的比特由来自 4 个 RS 码字的数据交织而成，且连续的 2 个比特集合中的比特来自不同的 RS 码字。每个比特集合包含 $t=68$ 个比特子集，每个比特子集包括 40 个比特，且这 40 比特是从 4 个 RS 码字的分别获取 1 个 RS 符号交织而成，即每个比特子集包含 4 个 RS 符号，这 4 个 RS 符号分别来自不同的 RS 码字，且比特子集中每个连续的 4 比特分别来自 4 个不同的 RS 码字。

图 14 为本申请实施例中第一数据流的第六种结构示意图。如图 14 所示，在一种可能的实施方式中，

先对第一数据流中的 AM 集合进行锁定, 然后以 AM 集合为起始对第一数据流进行划分得到多个比特子集, 每个比特子集包括 $d=40$ 个比特。相当于以比特子集为粒度对准卷积交织器中每个存储单元, 因此称之为交织对准。

图 15 为本申请实施例中卷积交织器的第三种结构示意图。如图 15 所示, 其包含 $p=3$ 条延迟线(delay line)。这 $p=3$ 条延迟线分别包括 $2Q$ 个存储单元、 Q 个存储单元、 0 个存储单元, 每个存储单元用于存储一个比特子集($d=40$ 比特)。也就是延迟线 0 的延迟取值为 $80 \times Q$ 个比特, 延迟线 1 的延迟取值为 $40 \times Q$ 个比特, 延迟线 2 的延迟取值为 0 个比特即无延迟。

如图 15 所示的 $C_r(\cdot)$ 表示第一数据流 r ($0 \leq r \leq m-1$) 中的一个比特子集。例如, $C_r(3t)$ 表示第一数据流 r 当前输入给延迟线 0 的一个比特子集, $C_r(3t-6Q)$ 为延迟线 0 输出的比特子集。 $C_r(3t+1)$ 表示第一数据流 r 接下来输入给延迟线 1 的一个比特子集, $C_r(3t-3Q+1)$ 为延迟线 1 输出的一个比特子集。 $C_r(3t+2)$ 表示第一数据流 r 接下来输入给延迟线 2 的一个比特子集, $C_r(3t+2)$ 为延迟线 2 输出的一个比特子集。 $C_r(3t+3)$ 表示第一数据流 r 再随后输入给延迟线 0 的一个比特子集, $C_r(3t-6Q+3)$ 为延迟线 0 输出的一个比特子集, 依次类推。结合第一 FEC 编码数据流中 RS 分布规则, 第一数据流中连续的 2 个比特集合来自不同的 RS 码字, 且比特集合划分为 68 个比特子集, 即第一数据流中的比特子集 $C_r(A)$ 与比特子集 $C_r(A+68)$ 中的比特来自不同的 RS 码字。当 $p \times Q + 1 \geq 68$ 的时候, 即 $Q \geq 23$ 使可以使得卷积交织 3 个延迟线轮询输出的 $C_r(3t-6Q)$, $C_r(3t-3Q+1)$, $C_r(3t+2)$ 总共 120 个比特来自 12 个不同的 RS 码字。

图 16 为本申请实施例中卷积交织器的第四种结构示意图。如图 16 所示, 在一种可能的实施方式中, 选取 $Q = 23$, 其对应的交织和解交织总时延约为 $46 \times 40 \times 3 = 5520$ 比特, 其等效在 $1 \times 800\text{GE}$ 业务时交织和解交织总时延约为 52ns。

图 17 为本申请实施例中卷积交织器的第五种结构示意图。如图 17 所示, 为了便于将卷积交织器进行低频率时钟实现, 可进一步约束 Q 为 2 的倍数, 例如选取 $Q=24$ 。其对应的交织和解卷积的总时延为 $48 \times 40 \times 3 = 5520$ 比特, 其等效在 $1 \times 800\text{GE}$ 业务时交织和解交织总时延约为 54ns。对应硬件实现时此可以采用约 885Mhz 的时钟, 在每个时钟将 3 个连续的比特子集分别输入到 3 条延迟线存储位宽为 40 比特的存储单元, 然后从 3 条延迟线的存储单元分别读出一个比特子集。此等效为图 17 所示的卷积交织器的延迟线 0 的延迟单元采用位宽为 40 比特深度为 48 为 memory 实现, 而延迟线 0 的延迟单元采用位宽为 40 比特深度为 24 为 memory 实现。也可以采用约 442Mhz 时钟, 在每个时钟内将 6 个连续的比特子集中的第 i 个比特子集和第 $i+3$ 个比特子集输出到延迟线 i 上存储位宽为 80 比特的存储单元内, 同时从 3 条延迟线的存储单元分别读出 2 个比特子集。此等效为图 17 所示的卷积交织器的延迟线 0 的延迟单元采用位宽为 80 比特深度为 24 为 memory 实现, 而延迟线 0 的延迟单元采用位宽为 80 比特深度为 12 为 memory 实现。

对上述 8 条第二数据流分别进行内码编码, 其中内码编码信息位长度为 120 比特。具体地, 内码编码器分别将第二数据流中连续的 3 个比特子集总共 120 比特添加冗余, 得到第三数据流, 且此 120 个比特为卷积交织器的 3 条延迟线一次轮询输出的 3 个比特子集 $C_r(3t-6Q)$ 、 $C_r(3t-3Q+1)$ 和 $C_r(3t+2)$ 。采用此种方式可以便于在完成内码同步后进行解卷积交织的同步。在一种可能的实施方式中, 采用 Hamming(128, 120) 进行内码编码, 分别将每条第二数据流中连续的 120 比特添加 8 比特冗余得到 128 比特的码字。采用本实施例 1 的方案, 其 KP4 RS(544, 514) + Hamming(128, 120) 级联码在 AWGN 下, 为了达到纠后误比特率 BER 为 $1\text{E}-15$ 所对应的纠前 BER 约为 $4.8\text{E}-3$, 性能逼近该级联 FEC 方案的最优性能, 且交织器的时延仅为 54ns 以内。

本实施例 1 中, 对来自 800GAUI-8 接口的 8 个物理通道的数据流分别进行交织对准和卷积交织, 相比现有技术对 32 个 PCS 通道分别进行 AM 锁定和卷积交织, 其具有更低的复杂度。且直接利用 4 个 RS 交织后的第一数据流进行卷积交织相比现有技术中对 2 个 RS 交织后的 PCS 通道数据流进行卷积交织, 可以使得本方案中的卷积交织具有更低的时延, 使得该级联 FEC 传输方案能够应用于较多的传输场景, 尤其适用于对需要较低传输时延的传输场景, 如低时延数据中心互联场景。

实施例 2: 应用于 $1 \times 800\text{G}$ 接口的场景, $m=4$, $v=4$, $t=136$, $s=10$, $p=3$ 。

前端设备 01 采用 32:4 PMA 将 32 条 PCS 通道数据流复用为 4 条数据流, 并通过 800GAUI-4 接口将

数据发送到发端处理模块 02。发端处理模块 02 对来自 800GAUI-4 接口的 4 条数据流进行时钟恢复、均衡、解调等接收处理输出 4 条第一数据流。

图 18 为本申请实施例中第一数据流的第七种结构示意图。如图 18 所示, 第一数据流中每两个 AM 集合的起始位置之间包括 $L_1=22282240$ 个比特, 即 4096 个比特集合, 每个比特集合包括 5440 个比特。AM 集合的长度为 PCS 通道数据流中 AM 的长度的 8 倍, 以 AM 的长度为 120 比特为例, AM 集合的长度等于 $8 \times 120=960$ 比特。其中, 每个比特集合中的比特由来自 4 个 RS 码字的多个 RS 符号交织而成, 且连续的 2 个比特集合中的比特来自不同的 RS 码字。每个比特集合包含 136 个比特子集, 每个比特子集包括 40 个比特, 且这 40 比特是从 4 个 RS 码字的分别获取 1 个 RS 符号交织而成, 即每个比特子集包含 4 个 RS 符号, 这 4 个 RS 符号分别来自不同的 RS 码字, 且比特子集中连续的 10 比特来自同一个 RS 码字。

图 19 为本申请实施例中卷积交织器的第六种结构示意图。如图 19 所示, 根据第一编码流中的 RS 分布规则, 第一数据流中的比特子集 $C_r(A)$ 与比特子集 $C_r(A+136)$ 中的比特来自不同的 RS 码字。当 $p \times Q + 1 \geq 136$ 的时候, 即 $Q \geq 45$ 使得可以使得卷积交织 3 个延迟线轮询输出的 $C_r(3t-6Q)$, $C_r(3t-3Q+1)$, $C_r(3t+2)$ 总共 120 个比特来自 12 个不同的 RS 码字。且为了便于卷积交织器可以采用更低时钟实现, 可进一步 Q 应该为 2 或者 4 的倍数, 如取 $Q=46$ 。硬件实现时此可以采用约 1.77Ghz 的时钟, 在每个时钟将 3 个连续的比特子集分别输入到 3 条延迟线存储位宽为 40 比特的存储单元, 然后从 3 条延迟线的存储单元分别读出一个比特子集。此等效为图 19 所示的卷积交织器的延迟线 0 的延迟单元采用位宽为 40 比特深度为 92 为 memory 实现, 而延迟线 0 的延迟单元采用位宽为 40 比特深度为 46 为 memory 实现。也可以采用约 885Mhz 时钟, 在每个时钟内将 6 个连续的比特子集中的第 i 个比特子集和第 $i+3$ 个比特子集输出到延迟线 i 上存储位宽为 80 比特的存储单元内, 同时从 3 条延迟线的存储单元分别读出 2 个比特子集。此等效为图 19 所示的卷积交织器的延迟线 0 的延迟单元采用位宽为 80 比特深度为 46 为 memory 实现, 而延迟线 0 的延迟单元采用位宽为 80 比特深度为 23 为 memory 实现。

本实施例 2 中, 对来自 800GAUI-4 接口的 4 个物理通道的数据流分别进行交织对准和卷积交织, 相比现有技术对先从 800GAUI-4 接口的 4 个物理通道先恢复出 32 个 PCS 通道, 再分别进行 AM 锁定和卷积交织, 其具有更低的复杂度。且直接利用 4 个 RS 交织后的第一数据流进行卷积交织相比现有技术中对 2 个 RS 交织后的 PCS 通道数据流进行卷积交织, 可以使得本方案中的卷积交织具有更低的时延, 使得该级联 FEC 传输方案能够应用于较多的传输场景, 尤其适用于对需要较低传输时延的传输场景, 如低时延数据中心互联场景。

实施例 3: 应用于 1600GbE 场景, $m=8$, $v=4$, $t=68$, $s=10$, $p=3$ 。

图 20 为 1600GbE 场景下 16 条 PCS 通道数据流的示意图。如图 20 所示, 发端设备 01 对待发送数据流周期性的插入 AM, 再进行 RS 编码 (即外码编码或者第一 FEC 编码), 然后进行 4 路 RS 交织将码字分配到 16 条 PCS 通道上, 使得 16 条 PCS 通道上 RS 码字排布如图 20 所示。每条 PCS 通道数据流中相邻 $a=4$ 个符号来自不同 RS 码字, 且相邻两条 PCS 通道数据流的同个位置的 2 个符号来自不同 RS 码字, 每条 PCS 通道上连续的 136 个符号来自 4 个 RS 码字。发端设备 01 采用 16:8 PMA 对 16 条 PCS 通道数据流进行符号复用后通过 1600GAUI-8 接口将数据发送到发端处理模块 02。发端处理模块 02 对来自 1600GAUI-8 接口的 8 条数据流进行时钟恢复、均衡、解调等接收处理输出 8 条第一数据流。

图 21 为本申请实施例中第一数据流的第八种结构示意图。如图 21 所示, 第一数据流中每两个 AM 集合的起始位置之间包括 $L_1=22282240$ 个比特, 即 8192 个比特集合, 每个比特集合包括 2720 个比特。AM 集合的长度为 PCS 通道数据流中 AM 的长度的 2 倍, 以 AM 的长度为 120 比特为例, AM 集合的长度等于 $2 \times 120=240$ 比特。其中, 每个比特集合中的比特由来自 4 个 RS 码字的多个 RS 符号交织而成, 且连续的 2 个比特集合中的比特来自不同的 RS 码字。每个比特集合包含 $t=68$ 个比特子集, 每个比特子集包括 40 个比特, 且这 40 比特是从 4 个 RS 码字的分别获取 1 个 RS 符号交织而成, 即每个比特子集包含 4 个 RS 符号, 这 4 个 RS 符号分别来自不同的 RS 码字, 且比特子集中连续的 10 比特来自同一个 RS 码字。

实施例 3 可以采用上述图 15 所示的卷积交织器, 根据第一编码流中的 RS 分布规则, 第一数据流中的比特子集 $C_r(A)$ 与比特子集 $C_r(A+68)$ 中的比特来自不同的 RS 码字, 则当 $p \times Q + 1 \geq 68$ 是, 即 $Q \geq 23$ 时可以使使得卷积交织 3 个延迟线轮询输出的 $C_r(3t-6Q)$, $C_r(3t-3Q+1)$, $C_r(3t+2)$ 总共 120 个比特来自 12

个不同的 RS 码字。且为了便于卷积交织器可以采用更低时钟实现,可进一步选取 Q 为 2 或者 4 的倍数,如取 $Q=24$ 应卷积交织器具体实现如图 17 所示。其对应的交织和解卷积的总时延为 $48 \times 40 \times 3 = 5520$ 比特,其等效在 $1 \times 1600\text{GE}$ 业务时交织和解交织总时延约为 27ns 。

硬件实现时此可以采用约 1.77GHz 的时钟,在每个时钟将 3 个连续的比特子集分别输入到 3 条延迟线存储位宽为 40 比特的存储单元,然后从 3 条延迟线的存储单元分别读出一个比特子集。此等效为图 17 所示的卷积交织器的延迟线 0 的延迟单元采用位宽为 40 比特深度为 48 为 memory 实现,而延迟线 0 的延迟单元采用位宽为 40 比特深度为 24 为 memory 实现。也可以采用约 885Mhz 时钟,在每个时钟内将 6 个连续的比特子集中的第 i 个比特子集和第 $i+3$ 个比特子集输出到延迟线 i 上存储位宽为 80 比特的存储单元内,同时从 3 条延迟线的存储单元分别读出 2 个比特子集。此等效为图 17 所示的卷积交织器的延迟线 0 的延迟单元采用位宽为 80 比特深度为 24 为 memory 实现,而延迟线 0 的延迟单元采用位宽为 80 比特深度为 12 为 memory 实现。还可以采用约 442.5Mhz 时钟,在每个时钟内将 12 个连续的比特子集中的第 i 个比特子集、第 $i+3$ 个比特子集、第 $i+6$ 个比特子集、第 $i+9$ 个比特子集输出到延迟线 i 上存储位宽为 160 比特的存储单元内,同时从 3 条延迟线的存储单元分别读出 4 个比特子集。此等效为图 17 所示的卷积交织器的延迟线 0 的延迟单元采用位宽为 160 比特深度为 12 为 memory 实现,而延迟线 0 的延迟单元采用位宽为 160 比特深度为 6 为 memory 实现。

本实施例 3 中,对来自 16000GAUI-8 接口的 8 个物理通道的数据流分别进行交织对准和卷积交织,相比现有技术对先从 16000GAUI-8 接口的 8 个物理通道先恢复出 16 个 PCS 通道,再分别进行 AM 锁定和卷积交织,其具有更低的复杂度。

需要说明的是,在实际应用中,在完成交织对准后,发端处理模块还可以先对 m 条第一数据流先进行复用再进行卷积交织,以便于在当 AUI-m 接口的单物理通道进行速率提升后可以保持后续的卷积交织和编码不做任何改动。下面对这种实施方式进行介绍。

图 22 为本申请实施例中发端处理模块的另一种数据处理示意图。如图 22 所示,发端处理模块对 m 条第一编码数据流进行交织对准得到 m 条第一数据流之后,先对 m 条第一数据流进行复用得到 $m/2$ 条复用数据流,再对 $m/2$ 条复用数据流分别进行卷积交织得到 $m/2$ 条第二数据流。进而,对 $m/2$ 条第二数据流进行内码编码得到 $m/2$ 条第二编码数据流。

图 23 为本申请实施例中对第一数据流进行复用的示意图。如图 23 所示,复用操作是将每 2 条第一数据流复用为 1 条复用数据流。具体地,从 2 条第一数据流中轮询输出一个比特子集到一条复用数据流,即复用数据流中连续的 2 个比特子集分别来自第一数据流 $2i$ 和第一数据流 $2i+1$ 。

下面介绍本申请实施例提供的数据处理装置。

图 24 为本申请实施例中数据处理装置的一种结构示意图。如图 24 所示,该数据处理装置包括:处理单元 201、卷积交织器 202 和编码器 203。处理单元 201 用于执行上述步骤 101 的操作。卷积交织器 202 用于执行上述步骤 102 的操作。编码器 203 用于执行上述步骤 103 的操作。具体操作可以参照上述数据处理方法的相关介绍,此处不再赘述。

应理解,本申请提供的装置也可以通过其他方式实现。例如,上述装置中的单元划分仅仅是一种逻辑功能划分,实际实现时可以有另外的划分方式,例如,多个单元或组件可以结合或可以集成到另一个系统。另外,本申请各个实施例中的各功能单元可以集成在一个物理单元中,也可以是各个独立的物理单元,也可以是两个或两个以上个功能单元集成在一个物理单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

图 25 为本申请实施例中数据处理装置的另一种结构示意图。如图 25 所示,数据处理装置包括处理器 301、存储器 302 和收发器 303。该处理器 301、存储器 302 和收发器 303 通过线路相互连接。其中,存储器 302 用于存储程序指令和数据。具体地,收发器 303 用于执行数据收发的操作。处理器 301 用于执行上述数据处理方法的操作。在一种可能的实施方式中,处理器 301 可以包括上述图 24 所示的处理单元 201、卷积交织器 202 和编码器 203。

需要说明的是,上述图 24 中所示的处理器可以采用通用的中央处理器(Central Processing Unit,

CPU)，通用处理器、数字信号处理器（DSP）、专用集成电路（ASIC）、现场可编程门阵列（FPGA）或者其他可编程逻辑器件、晶体管逻辑器件、硬件部件或者其任意组合。上述图 24 中所示的存储器可以存储操作系统和其他应用程序。在通过软件或者固件来实现本申请实施例提供的技术方案时，用于实现本申请实施例提供的技术方案的程序代码保存在存储器中，并由处理器来执行。在一实施例中，处理器内部可以包括存储器。在另一实施例中，处理器和存储器是两个独立的结构。

所属领域的技术人员可以清楚地了解到，为描述的方便和简洁，上述描述的系统、装置和单元的具体工作过程，可以参考前述方法实施例中的对应过程，在此不再赘述。

本领域普通技术人员可以理解实现上述实施例的全部或部分步骤可以通过硬件来完成，也可以通过程序来指令相关的硬件完成，所述的程序可以存储于一种计算机可读存储介质中，上述提到的存储介质可以是只读存储器，随机接入存储器等。上述的这些功能究竟以硬件还是软件方式来执行，取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能，但是这种实现不应认为超出本申请的范围。

当使用软件实现时，上述实施例描述的方法步骤可以全部或部分地以计算机程序产品的形式实现。所述计算机程序产品包括一个或多个计算机指令。在计算机上加载和执行所述计算机程序指令时，全部或部分地产生按照本申请实施例所述的流程或功能。所述计算机可以是通用计算机、专用计算机、计算机网络、或者其他可编程装置。所述计算机指令可以存储在计算机可读存储介质中，或者从一个计算机可读存储介质向另一个计算机可读存储介质传输，例如，所述计算机指令可以从一个网站站点、计算机、服务器或数据中心通过有线（例如同轴电缆、光纤、数字用户线（DSL））或无线（例如红外、无线、微波等）方式向另一个网站站点、计算机、服务器或数据中心进行传输。所述计算机可读存储介质可以是计算机能够存取的任何可用介质或者是包含一个或多个可用介质集成的服务器、数据中心等数据存储设备。所述可用介质可以是磁性介质，（例如，软盘、硬盘、磁带）、光介质（例如，DVD）、或者半导体介质（例如固态硬盘 Solid State Disk（SSD））等。

权 利 要 求 书

1. 一种数据处理方法，其特征在于，包括：

获取 m 条第一数据流，所述 m 为大于 1 的整数，每条所述第一数据流由经过第一前向纠错 FEC 编码后的 v 路第一码字交织得到，每条所述第一数据流包括连续的多个比特集合，每个所述比特集合包括 t 个比特子集，所述 t 为大于 1 的整数，每个所述比特子集包括 d 个比特， $d=v \times s$ ，所述 v 为大于 1 的整数，所述 s 表示所述第一码字中每个符号包括的比特数，所述 s 为大于或等于 1 的整数，每个所述比特子集中的比特来自 v 个符号，所述 v 个符号分别来自 v 个第一码字，所述 v 个第一码字分别来自所述 v 路第一码字；

根据 p 条延迟线分别对所述 m 条第一数据流进行延迟得到 m 条第二数据流，所述 p 为大于 1 的整数，每条延迟线包括的存储单元数量各不相同，存储单元数量最小的延迟线包括 0 个存储单元，每相邻两条延迟线的存储单元数量的差值为 Q ，所述 Q 为大于或等于 1 的整数，每个存储单元用于存储 d 个比特，每条所述第一数据流中的比特按照所述 p 条延迟线的序号依次输入所述 p 条延迟线，每条延迟线单次输入 1 个所述比特子集中的 d 个比特且单次输出 d 个比特。

2. 根据权利要求 1 所述的方法，其特征在于，每条所述第二数据流中连续的 $p \times d$ 个比特来自 $v \times p$ 个第一码字。

3. 根据权利要求 1 或 2 所述的方法，其特征在于，所述 p 条延迟线中序号最大的延迟线包括 0 个存储单元， $(p \times Q + 1) \geq t$ 。

4. 根据权利要求 1 或 2 所述的方法，其特征在于，所述 p 条延迟线中序号最小的延迟线包括 0 个存储单元， $(p \times Q - 1) \geq t$ 。

5. 根据权利要求 1 至 4 中任一项所述的方法，其特征在于，所述 Q 为 2 的倍数。

6. 根据权利要求 1 至 5 中任一项所述的方法，其特征在于，所述 m 条第一数据流由 n 条通道数据流复用得到，所述 n 为所述 m 的整数倍，每条所述通道数据流经过所述第一 FEC 编码，每条所述通道数据流包括多个对齐标识 AM，每条所述第一数据流包括多个 AM 集合，每个所述 AM 集合包括 n/m 个所述 AM，每相邻两个所述 AM 集合的起始位置之间包括整数个所述比特集合。

7. 根据权利要求 6 所述的方法，其特征在于，获取所述 m 条第一数据流包括：

根据至少一个所述 AM 集合对所述第一数据流进行交织对准，以确定所述第一数据流中每个所述比特集合的边界位置。

8. 根据权利要求 1 至 7 中任一项所述的方法，其特征在于，所述多个比特集合包括相邻的第一比特集合和第二比特集合，所述第一比特集合中的比特来自所述第一 FEC 编码后的其中 v 个第一码字，所述第二比特集合中的比特来自所述第一 FEC 编码后的另外 v 个码字。

9. 根据权利要求 1 至 8 中任一项所述的方法，其特征在于，所述比特子集中每连续的 v 个比特分别来自 v 个第一码字，或者，所述比特子集中每连续的 s 个比特来自同一个第一码字。

10. 根据权利要求 1 至 9 中任一项所述的方法，其特征在于， $m=8$ ， $v=4$ ， $t=68$ ， $s=10$ ， $p=3$ ， $Q=23$ 或 24；或者， $m=4$ ， $v=4$ ， $t=136$ ， $s=10$ ， $p=3$ ， $Q=46$ 。

11. 根据权利要求 1 至 10 中任一项所述的方法，其特征在于，得到共 m 条第二数据流之后，所述方法还包括：

对所述 m 条第二数据流分别进行第二 FEC 编码得到 m 条第三数据流，所述第二 FEC 编码后的每个第二码字中的 K 个信息比特最多来自 $p \times v$ 个不同的第一码字，所述 K 为 $p \times d$ 的整数倍。

12. 一种数据处理装置，其特征在于，包括：处理单元和卷积交织器；

所述处理单元用于：获取 m 条第一数据流，所述 m 为大于 1 的整数，每条所述第一数据流由经过第一前向纠错 FEC 编码后的 v 路第一码字交织得到，每条所述第一数据流包括连续的多个比特集合，每个所述比特集合包括 t 个比特子集，所述 t 为大于 1 的整数，每个所述比特子集包括 d 个比特， $d=v \times s$ ，所述 v 为大于 1 的整数，所述 s 表示所述第一码字中每个符号包括的比特数，所述 s 为大于或等于 1 的整数，每个所述比特子集中的比特来自 v 个符号，所述 v 个符号分别来自 v 个第一码字，所述 v 个第一码字分别来自所述 v 路第一码字；

所述卷积交织器用于：根据 p 条延迟线分别对所述 m 条第一数据流进行延迟得到 m 条第二数据流，

所述 p 为大于 1 的整数，每条延迟线包括的存储单元数量各不相同，存储单元数量最小的延迟线包括 0 个存储单元，每相邻两条延迟线的存储单元数量的差值为 Q ，所述 Q 为大于或等于 1 的整数，每个存储单元用于存储 d 个比特，每条所述第一数据流中的比特按照所述 p 条延迟线的序号依次输入所述 p 条延迟线，每条延迟线单次输入 1 个所述比特子集中的 d 个比特且单次输出 d 个比特。

5 13. 根据权利要求 12 所述的数据处理装置，其特征在于，每条所述第二数据流中连续的 $p \times d$ 个比特来自 $v \times p$ 个第一码字。

14. 根据权利要求 12 或 13 所述的数据处理装置，其特征在于，所述 p 条延迟线中序号最大的延迟线包括 0 个存储单元， $(p \times Q + 1) \geq t$ 。

10 15. 根据权利要求 12 或 13 所述的数据处理装置，其特征在于，所述 p 条延迟线中序号最小的延迟线包括 0 个存储单元， $(p \times Q - 1) \geq t$ 。

16. 根据权利要求 12 至 15 中任一项所述的数据处理装置，其特征在于，所述 Q 为 2 的倍数。

17. 根据权利要求 12 至 16 中任一项所述的数据处理装置，其特征在于，所述 m 条第一数据流由 n 条通道数据流复用得到，所述 n 为所述 m 的整数倍，每条所述通道数据流经过所述第一 FEC 编码，每条所述通道数据流包括多个对齐标识 AM，每条所述第一数据流包括多个 AM 集合，每个所述 AM 集合包括
15 n/m 个所述 AM，每相邻两个所述 AM 集合的起始位置之间包括整数个所述比特集合。

18. 根据权利要求 17 所述的数据处理装置，其特征在于，所述处理单元具体用于：

根据至少一个所述 AM 集合对所述第一数据流进行交织对准，以确定所述第一数据流中每个所述比特集合的边界位置。

19. 根据权利要求 12 至 18 中任一项所述的数据处理装置，其特征在于，所述多个比特集合包括相邻的第一比特集合和第二比特集合，所述第一比特集合中的比特来自所述第一 FEC 编码后的其中 v 个第一码字，所述第二比特集合中的比特来自所述第一 FEC 编码后的另外 v 个码字。
20

20. 根据权利要求 12 至 19 中任一项所述的数据处理装置，其特征在于，所述比特子集中每连续的 v 个比特分别来自 v 个第一码字，或者，所述比特子集中每连续的 s 个比特来自同一个第一码字。

21. 根据权利要求 12 至 20 中任一项所述的数据处理装置，其特征在于， $m=8$ ， $v=4$ ， $t=68$ ， $s=10$ ， $p=3$ ， $Q=23$ 或 24；或者， $m=4$ ， $v=4$ ， $t=136$ ， $s=10$ ， $p=3$ ， $Q=46$ 。
25

22. 根据权利要求 12 至 21 中任一项所述的数据处理装置，其特征在于，所述数据处理装置还包括编码器，所述编码器用于：

对所述 m 条第二数据流分别进行第二 FEC 编码得到 m 条第三数据流，所述第二 FEC 编码后的每个第二码字中的 K 个信息比特最多来自 $p \times v$ 个不同的第一码字，所述 K 为 $p \times d$ 的整数倍。
30

35