



(12) 发明专利

(10) 授权公告号 CN 116318552 B

(45) 授权公告日 2023.09.22

(21) 申请号 202310277881.0

CN 101322317 A, 2008.12.10

(22) 申请日 2023.03.15

CN 103888224 A, 2014.06.25

(65) 同一申请的已公布的文献号

CN 104811211 A, 2015.07.29

申请公布号 CN 116318552 A

CN 111066252 A, 2020.04.24

(43) 申请公布日 2023.06.23

CN 114679185 A, 2022.06.28

(73) 专利权人 归芯科技(深圳)有限公司

CN 1529943 A, 2004.09.15

地址 518000 广东省深圳市南山区桃源街
道平山社区平山一路2号南山云谷创
业园二期11栋703

US 2010023844 A1, 2010.01.28

(72) 发明人 王翰林

US 2016149591 A1, 2016.05.26

(74) 专利代理机构 北京兰亭信通知识产权代理
有限公司 11667

US 2020212937 A1, 2020.07.02

专利代理人 赵永刚

Qualcomm Europe S.A.R.L..S4-090054 "

(51) Int.Cl.

TS 26.268 v1.1.0 eCall data transfer; In-

H04L 1/00 (2006.01)

band modem solution; ANSI-C reference

(56) 对比文件

code".3GPP tsg_sa\WG4_CODEC.2009, (第

CN 101335691 A, 2008.12.31
CN 108781124 A, 2018.11.09
CN 112152638 A, 2020.12.29
CN 101124731 A, 2008.02.13

TSGS4_52期), 全文.

ITRI.R1-070112 "IBP interleaver for

turbo coding and shortening position

assigning algorithm".3GPP tsg_ran\WG1_

RL1.2007, (第TSGR1_47bis期), 全文.

李海清; 杨莘元. 一种双螺旋Turbo码交织器

的设计. 应用科技. 2008, (第02期), 全文. (续)

审查员 牟茜茜

权利要求书3页 说明书10页 附图3页

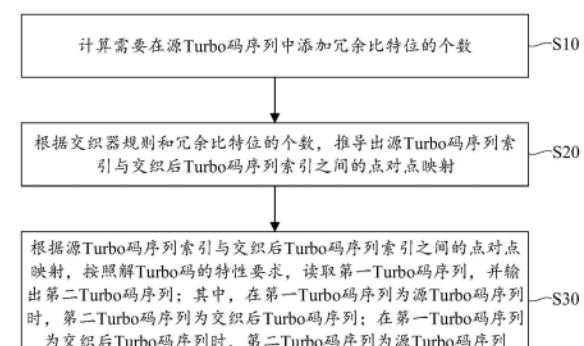
(54) 发明名称

分, 从而缩短处理时间。数据是紧密排列的, 不需要处理冗余比特位的时间。在资源只需要一块缓存即可, 从而减少交织或解交织过程中占用的处理资源。

Turbo码的交织或解交织方法及其器件、通信芯片和装置

(57) 摘要

本发明提供了一种Turbo码的交织或解交织方法及其器件、通信芯片和装置, 通过计算需要在源Turbo码序列中添加冗余比特位的个数; 之后根据交织器规则和冗余比特位的个数, 推导出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射; 再根据源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射, 按照解Turbo码的特性要求, 读取第一Turbo码序列, 并输出第二Turbo码序列, 进行由源Turbo码序列到交织后Turbo码序列的交织, 或由交织后Turbo码序列到源Turbo码序列的解交织。在处理时间上只需要做解交织部分, 而无需做交织部



[接上页]

(56) 对比文件

Marc Adrat. "Turbo DeCodulation using
Highly Redundant Index Assignments and

Multi-Dimensional Mappings Thorsten
Clevorn".《4th International Symposium on
Turbo Codes & Related Topics
》.2006,

1. 一种Turbo码的交织或解交织方法,其特征在于,包括:

计算需要在源Turbo码序列中添加冗余比特位的个数;

根据交织器规则和所述冗余比特位的个数,推导出所述源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射;

根据所述源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射,按照解Turbo码的特性要求,读取第一Turbo码序列,并输出第二Turbo码序列;其中,在所述第一Turbo码序列为所述源Turbo码序列时,所述第二Turbo码序列为所述交织后Turbo码序列;在所述第一Turbo码序列为所述交织后Turbo码序列时,所述第二Turbo码序列为所述源Turbo码序列;

其中,所述交织器规则包括:

根据所述源Turbo码序列索引和所述冗余比特位的个数N,获得添加冗余比特位后的第一过渡Turbo码序列索引;

根据所述交织器规则中的交织列数,计算所述第一过渡Turbo码序列索引所需的交织行数;

根据所述第一过渡Turbo码序列索引,按照先行后列的顺序填充源数据,生成交织表;

根据所述交织表,生成第二过渡Turbo码序列索引;

根据所述交织器规则中的列变换规则,生成列变换之后的第三过渡Turbo码序列索引;

删去所述第三过渡Turbo码序列索引中的冗余比特位,得到所述交织后Turbo码序列索引。

2. 如权利要求1所述的交织或解交织方法,其特征在于,所述根据交织器规则和所述冗余比特位的个数,推导出所述源Turbo码序列与交织后Turbo码序列索引之间的点对点映射,包括:

所述源Turbo码序列为 D_i ,其中,i表示所述源Turbo码序列中的索引号;

所述第一过渡Turbo码序列为 $D_{i'}$,其中, i' 表示所述第一过渡Turbo码序列中的索引号;

其中, $i' = i + N_{\text{dummy}}$, N_{dummy} 表示需要在所述源Turbo码序列中添加冗余比特位的个数。

3. 如权利要求2所述的交织或解交织方法,其特征在于,所述根据交织器规则和所述冗余比特位的个数,推导出所述源Turbo码序列与交织后Turbo码序列索引之间的点对点映射,还包括:

所述交织器规则中的交织列数为32;

计算所述第一过渡Turbo码序列索引所需的交织行数 $R = \text{ceil}(D/32)$,其中,D表示所述源Turbo码序列中的数据个数。

4. 如权利要求3所述的交织或解交织方法,其特征在于,所述根据交织器规则和所述冗余比特位的个数,推导出所述源Turbo码序列与交织后Turbo码序列索引之间的点对点映射,还包括:

所述第二过渡Turbo码序列为 D_j ,其中,j表示所述第二过渡Turbo码序列中的索引号;

计算所述第二过渡Turbo码序列中的行索引号 $\text{raw}_j = \text{floor}(i' / 32)$,其中, $\text{floor}(i' / 32)$ 表示 i' 除以32取商;

计算所述第二过渡Turbo码序列中的列索引号 $\text{col}_j = i' \% 32$,其中, $i' \% 32$ 表示 i' 除以32取余数。

5. 如权利要求4所述的交织或解交织方法,其特征在于,所述根据交织器规则和所述冗余比特位的个数,推导出所述源Turbo码序列与交织后Turbo码序列索引之间的点对点映射,还包括:

所述源Turbo码序列为系统比特码序列或第一校验码序列;

所述第三过渡Turbo码序列为 D_k ,其中,k表示所述第三过渡Turbo码序列中的索引号;

计算所述第三过渡Turbo码序列中的行索引号 $rawk=rawj$;

所述第三过渡Turbo码序列索引的列间变换矩阵为P;

按照如下公式,计算所述第三过渡Turbo码序列索引的列间变换矩阵P的反变换矩阵 P' :

$$P(x) = y, P'(y) = x$$

其中,x表示列变换前的列位置,y表示列变换后的列位置;

计算所述第三过渡Turbo码序列中的列索引号 $colk=P'(colj)$ 。

6. 如权利要求4所述的交织或解交织方法,其特征在于,所述根据交织器规则和所述冗余比特位的个数,推导出所述源Turbo码序列与交织后Turbo码序列索引之间的点对点映射,还包括:

所述源Turbo码序列为第二校验码序列;

所述交织器规则中,所述根据所述交织器规则中的列变换规则,生成列变换之后的第三过渡Turbo码序列索引,包括:根据所述交织器规则中的列变换规则,在列变换时循环右移了一列,生成列变换之后的第三过渡Turbo码序列索引;

所述第三过渡Turbo码序列为 D_k ,其中,k表示所述第三过渡Turbo码序列中的索引号;

所述第三过渡Turbo码序列索引的列间变换矩阵为P;

按照如下公式,计算所述第三过渡Turbo码序列索引的列间变换矩阵P的反变换矩阵 P' :

$$P(x) = y, P'(y) = x$$

其中,x表示列变换前的列位置,y表示列变换后的列位置;

如果所述第二过渡Turbo码序列中的列索引号 $colj=0$,在所述第二过渡Turbo码序列中的行索引号 $rawj=0$ 时,所述第三过渡Turbo码序列中的行索引号 $rawk=R-1$;在所述第二过渡Turbo码序列中的行索引号 $rawj\neq 0$ 时,所述第三过渡Turbo码序列中的行索引号 $rawk=rawj-1$;

如果所述第二过渡Turbo码序列中的列索引号 $colj\neq 0$,则所述第三过渡Turbo码序列中的行索引号 $rawk=rawj$;

计算所述第三过渡Turbo码序列中的列索引号 $colk=P'(colj)$ 。

7. 如权利要求5或6所述的交织或解交织方法,其特征在于,所述根据交织器规则和所述冗余比特位的个数,推导出所述源Turbo码序列与交织后Turbo码序列索引之间的点对点映射,还包括:

计算所述第三过渡Turbo码序列 D_k 中k之前的冗余比特位个数 nk ;

所述交织后Turbo码序列为 $D_{k'}$,其中, k' 表示所述交织后Turbo码序列中的索引号;

按照如下公式,计算 k' 与i之间的点对点映射:

$$k' = colk*R+rawk - nk。$$

8. 如权利要求7所述的交织或解交织方法,其特征在于,所述计算所述第三过渡Turbo码序列 D_k 中k之前的冗余比特位个数 n_k ,包括:

根据所述列变换规则,推算出每个冗余比特位所在的列位置;

根据每个冗余比特位所在的列位置,推算出所述第三过渡Turbo码序列 D_k 中k之前的冗余比特位个数 n_k 。

9. 一种Turbo码的交织或解交织器件,其特征在于,包括:

冗余比特位计算模块,用于计算需要在源Turbo码序列中添加冗余比特位的个数;

推导模块,用于根据交织器规则和所述冗余比特位的个数,推导出所述源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射;

交织或解交织模块,用于根据所述源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射,按照解Turbo码的特性要求,读取第一Turbo码序列,并输出第二Turbo码序列;其中,在所述第一Turbo码序列为所述源Turbo码序列时,所述第二Turbo码序列为所述交织后Turbo码序列;在所述第一Turbo码序列为所述交织后Turbo码序列时,所述第二Turbo码序列为所述源Turbo码序列;

其中,所述交织器规则包括:

根据所述源Turbo码序列索引和所述冗余比特位的个数N,获得添加冗余比特位后的第一过渡Turbo码序列索引;

根据所述交织器规则中的交织列数,计算所述第一过渡Turbo码序列索引所需的交织行数;

根据所述第一过渡Turbo码序列索引,按照先行后列的顺序填充源数据,生成交织表;

根据所述交织表,生成第二过渡Turbo码序列索引;

根据所述交织器规则中的列变换规则,生成列变换之后的第三过渡Turbo码序列索引;

删去所述第三过渡Turbo码序列索引中的冗余比特位,得到所述交织后Turbo码序列索引。

10. 如权利要求9所述的交织或解交织器件,其特征在于,还包括:

缓存模块,用于缓存所述第一Turbo码序列或第二Turbo码序列。

11. 一种通信芯片,其特征在于,包括:如权利要求9~10任一项所述的Turbo码的交织或解交织器件。

12. 一种通信装置,其特征在于,包括:如权利要求11所述的通信芯片。

Turbo码的交织或解交织方法及其器件、通信芯片和装置

技术领域

[0001] 本发明涉及通信技术领域,尤其涉及一种Turbo码的交织或解交织方法及其器件、通信芯片和装置。

背景技术

[0002] turbo编码作为一种信道编码方式,应用于LTE数据信道(PDSCH)中,其包括交织和解交织。其中,交织为把连续数据做离散处理,形成离散数据;解交织为把离散数据恢复成连续数据。由于PDSCH数据信道的数据处理量较大,所以对处理时间和面积都需要做到最优。现有的交织或解交织过程中,需要使用交织器和解交织器两个器件,每次解交织时,需要提前计算交织器的地址对应关系。解交织处理时间分两部分,主要包括计算交织器的时间和解交织的时间。并且还需要交织器buffer(缓存)和交织前数据缓存pre_buffer(缓存)两块缓存,这种方案是处理时间长,占用资源多。

发明内容

[0003] 本发明提供了一种Turbo码的交织或解交织方法及其器件、通信芯片和装置,缩短处理时间,减少交织或解交织过程中占用的处理资源。

[0004] 第一方面,本发明提供了一种Turbo码的交织或解交织方法,该Turbo码的交织或解交织方法包括:

[0005] 计算需要在源Turbo码序列中添加冗余比特位的个数;

[0006] 根据交织器规则和冗余比特位的个数,推导出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射;

[0007] 根据源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射,按照解Turbo码的特性要求,读取第一Turbo码序列,并输出第二Turbo码序列;其中,在第一Turbo码序列为源Turbo码序列时,第二Turbo码序列为交织后Turbo码序列;在第一Turbo码序列为交织后Turbo码序列时,第二Turbo码序列为源Turbo码序列。

[0008] 在上述的方案中,提供了一种新型快速交织或解交织的实现方案,通过计算需要在源Turbo码序列中添加冗余比特位的个数;之后根据交织器规则和冗余比特位的个数,推导出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射;再根据源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射,按照解Turbo码的特性要求,读取第一Turbo码序列,并输出第二Turbo码序列,进行由源Turbo码序列到交织后Turbo码序列的交织,或由交织后Turbo码序列到源Turbo码序列的解交织。在此过程中,不提前生成交织表,通过推导得到源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射。利用LTE交织中,如果添加的冗余比特位的个数确定了之后,就能够确定交织器规则中列变换之后索引k之前有几个冗余比特位就可以确定的特性,可以快速计算出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射。在处理时间上只需要做解交织部分,而无需做交织部分,从而缩短处理时间。数据是紧密排列的,不需要处理冗余比特位的时间,进一步

缩短处理时间。另外,在资源只需要一块缓存即可,从而减少交织或解交织过程中占用的处理资源。

[0009] 在一个具体的实施方式中,交织器规则包括:根据源Turbo码序列索引和冗余比特位的个数N,获得添加冗余比特位后的第一过渡Turbo码序列索引;根据交织器规则中的交织列数,计算第一过渡Turbo码序列索引所需的交织行数;根据第一过渡Turbo码序列索引,按照先行后列的顺序填充源数据,生成交织表;根据交织表,生成第二过渡Turbo码序列索引;根据交织器规则中的列变换规则,生成列变换之后的第三过渡Turbo码序列索引;删去第三过渡Turbo码序列索引中的冗余比特位,得到交织后Turbo码序列索引。便于快速且准确的推导得到源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射。

[0010] 在一个具体的实施方式中,根据交织器规则和冗余比特位的个数,推导出源Turbo码序列与交织后Turbo码序列索引之间的点对点映射,包括:源Turbo码序列为 D_i ,其中,i表示源Turbo码序列中的索引号;第一过渡Turbo码序列为 $D_{i'}$,其中, i' 表示第一过渡Turbo码序列中的索引号;其中, $i' = i + N_{\text{dummy}}$, N_{dummy} 表示需要在源Turbo码序列中添加冗余比特位的个数。便于快速计算得到需要在源Turbo码序列中添加冗余比特位的个数。

[0011] 在一个具体的实施方式中,根据交织器规则和冗余比特位的个数,推导出源Turbo码序列与交织后Turbo码序列索引之间的点对点映射,还包括:交织器规则中的交织列数为32;计算第一过渡Turbo码序列索引所需的交织行数 $R = \text{ceil}(D/32)$,其中,D表示源Turbo码序列中的数据个数。便于快速计算得到第一过渡Turbo码序列索引所需的交织行数。

[0012] 在一个具体的实施方式中,根据交织器规则和冗余比特位的个数,推导出源Turbo码序列与交织后Turbo码序列索引之间的点对点映射,还包括:第二过渡Turbo码序列为 D_j ,其中,j表示第二过渡Turbo码序列中的索引号;计算第二过渡Turbo码序列中的行索引号 $\text{raw}_j = \text{floor}(i' / 32)$,其中, $\text{floor}(i' / 32)$ 表示 i' 除以32取商;计算第二过渡Turbo码序列中的列索引号 $\text{col}_j = i' \% 32$,其中, $i' \% 32$ 表示 i' 除以32取余数。便于快速计算得到第二过渡Turbo码序列中的行索引号和列索引号。

[0013] 在一个具体的实施方式中,根据交织器规则和冗余比特位的个数,推导出源Turbo码序列与交织后Turbo码序列索引之间的点对点映射,还包括:源Turbo码序列为系统比特码序列或第一校验码序列;第三过渡Turbo码序列为 D_k ,其中,k表示第三过渡Turbo码序列中的索引号;计算第三过渡Turbo码序列中的行索引号 $\text{raw}_k = \text{raw}_j$;第三过渡Turbo码序列索引的列间变换矩阵为P;按照如下公式,计算第三过渡Turbo码序列索引的列间变换矩阵P的反变换矩阵 P' :

[0014] $P(x) = y, P'(y) = x$

[0015] 其中,x表示列变换前的列位置,y表示列变换后的列位置;

[0016] 计算第三过渡Turbo码序列中的列索引号 $\text{col}_k = P'(\text{col}_j)$ 。在源Turbo码序列为系统比特码序列或第一校验码序列时,便于快速计算得到第三过渡Turbo码序列中的行索引号和列索引号。

[0017] 在一个具体的实施方式中,根据交织器规则和冗余比特位的个数,推导出源Turbo码序列与交织后Turbo码序列索引之间的点对点映射,还包括:源Turbo码序列为第二校验码序列;交织器规则中,根据交织器规则中的列变换规则,生成列变换之后的第三过渡Turbo码序列索引,包括:根据交织器规则中的列变换规则,在列变换时循环右移了一列,生

成列变换之后的第三过渡Turbo码序列索引;第三过渡Turbo码序列为 D_k ,其中,k表示第三过渡Turbo码序列中的索引号;第三过渡Turbo码序列索引的列间变换矩阵为P;按照如下公式,计算第三过渡Turbo码序列索引的列间变换矩阵P的反变换矩阵P' :

[0018] $P(x) = y, P'(y) = x$

[0019] 其中,x表示列变换前的列位置,y表示列变换后的列位置;

[0020] 如果第二过渡Turbo码序列中的列索引号 $colj=0$,在第二过渡Turbo码序列中的行索引号 $rawj=0$ 时,第三过渡Turbo码序列中的行索引号 $rawk=R-1$;在第二过渡Turbo码序列中的行索引号 $rawj \neq 0$ 时,第三过渡Turbo码序列中的行索引号 $rawk=rawj-1$;

[0021] 如果第二过渡Turbo码序列中的列索引号 $colj \neq 0$,则第三过渡Turbo码序列中的行索引号 $rawk=rawj$;

[0022] 计算第三过渡Turbo码序列中的列索引号 $colk=P'(colj)$ 。在源Turbo码序列为第二校验码序列时,便于快速计算得到第三过渡Turbo码序列中的行索引号和列索引号。

[0023] 在一个具体的实施方式中,根据交织器规则和冗余比特位的个数,推导出源Turbo码序列与交织后Turbo码序列索引之间的点对点映射,还包括:计算第三过渡Turbo码序列 D_k 中k之前的冗余比特位个数nk;交织后Turbo码序列为 $D_{k'}$,其中,k'表示交织后Turbo码序列中的索引号;按照如下公式,计算k'与i之间的点对点映射:

[0024] $k' = colk * R + rawk - nk$

[0025] 便于快速得到第三过渡Turbo码序列 D_k 中k之前的冗余比特位个数nk,进而快速且准确的推导得到源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射。

[0026] 在一个具体的实施方式中,计算第三过渡Turbo码序列 D_k 中k之前的冗余比特位个数nk,包括:根据列变换规则,推算出每个冗余比特位所在的列位置;根据每个冗余比特位所在的列位置,推算出第三过渡Turbo码序列 D_k 中k之前的冗余比特位个数nk。便于快速且准确的得到第三过渡Turbo码序列 D_k 中k之前的冗余比特位个数nk。

[0027] 第二方面,本发明还提供了一种Turbo码的交织或解交织器件,该Turbo码的交织或解交织器件包括:冗余比特位计算模块、推导模块、交织或解交织模块。其中,冗余比特位计算模块用于计算需要在源Turbo码序列中添加冗余比特位的个数。推导模块用于根据交织器规则和冗余比特位的个数,推导出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射。交织或解交织模块用于根据源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射,按照解Turbo码的特性要求,读取第一Turbo码序列,并输出第二Turbo码序列;其中,在第一Turbo码序列为源Turbo码序列时,第二Turbo码序列为交织后Turbo码序列;在第一Turbo码序列为交织后Turbo码序列时,第二Turbo码序列为源Turbo码序列。

[0028] 在上述的方案中,提供了一种新型快速交织或解交织的实现方案,通过计算需要在源Turbo码序列中添加冗余比特位的个数;之后根据交织器规则和冗余比特位的个数,推导出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射;再根据源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射,按照解Turbo码的特性要求,读取第一Turbo码序列,并输出第二Turbo码序列,进行由源Turbo码序列到交织后Turbo码序列的交织,或由交织后Turbo码序列到源Turbo码序列的解交织。在此过程中,不提前生成交织表,通过推导得到源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射。利用LTE交织中,如果添加的冗余比特位的个数确定了之后,就能够确定交织器规则中列变换之

后索引k之前有几个冗余比特位就可以确定的特性,可以快速计算出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射。在处理时间上只需要做解交织部分,而无需做交织部分,从而缩短处理时间。数据是紧密排列的,不需要处理冗余比特位的时间,进一步缩短处理时间。另外,在资源只需要一块缓存即可,从而减少交织或解交织过程中占用的处理资源。

[0029] 在一个具体的实施方式中,交织或解交织器件还包括缓存模块,缓存模块用于缓存第一Turbo码序列或第二Turbo码序列。在资源只需要一块缓存即可,从而减少交织或解交织过程中占用的处理资源。

[0030] 第三方面,本发明还提供了一种通信芯片,该通信芯片包括上述任意一种Turbo码的交织或解交织器件。提供了一种新型快速交织或解交织的实现方案,通过计算需要在源Turbo码序列中添加冗余比特位的个数;之后根据交织器规则和冗余比特位的个数,推导出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射;再根据源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射,按照解Turbo码的特性要求,读取第一Turbo码序列,并输出第二Turbo码序列,进行由源Turbo码序列到交织后Turbo码序列的交织,或由交织后Turbo码序列到源Turbo码序列的解交织。在此过程中,不提前生成交织表,通过推导得到源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射。利用LTE交织中,如果添加的冗余比特位的个数确定了之后,就能够确定交织器规则中列变换之后索引k之前有几个冗余比特位就可以确定的特性,可以快速计算出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射。在处理时间上只需要做解交织部分,而无需做交织部分,从而缩短处理时间。数据是紧密排列的,不需要处理冗余比特位的时间,进一步缩短处理时间。另外,在资源只需要一块缓存即可,从而减少交织或解交织过程中占用的处理资源。

[0031] 第四方面,本发明还提供了一种通信装置,该通信装置包括上述任意一种通信芯片。提供了一种新型快速交织或解交织的实现方案,通过计算需要在源Turbo码序列中添加冗余比特位的个数;之后根据交织器规则和冗余比特位的个数,推导出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射;再根据源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射,按照解Turbo码的特性要求,读取第一Turbo码序列,并输出第二Turbo码序列,进行由源Turbo码序列到交织后Turbo码序列的交织,或由交织后Turbo码序列到源Turbo码序列的解交织。在此过程中,不提前生成交织表,通过推导得到源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射。利用LTE交织中,如果添加的冗余比特位的个数确定了之后,就能够确定交织器规则中列变换之后索引k之前有几个冗余比特位就可以确定的特性,可以快速计算出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射。在处理时间上只需要做解交织部分,而无需做交织部分,从而缩短处理时间。数据是紧密排列的,不需要处理冗余比特位的时间,进一步缩短处理时间。另外,在资源只需要一块缓存即可,从而减少交织或解交织过程中占用的处理资源。

附图说明

[0032] 图1为本发明实施例提供的一种Turbo码的交织或解交织方法的流程图;

[0033] 图2为本发明实施例提供的一种交织器规则的交织计算流程图;

- [0034] 图3为本发明实施例提供的一种Turbo编码交织时的流程图；
- [0035] 图4为本发明实施例提供的一种Turbo码的交织或解交织器件的示意框图；
- [0036] 图5为本发明实施例提供的另一种Turbo码的交织或解交织器件的示意框图。

具体实施方式

[0037] 为使本发明实施例的目的、技术方案和优点更加清楚，下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0038] 为了方便理解本发明实施例提供的Turbo码的交织或解交织方法，下面首先说明一下本发明实施例提供的Turbo码的交织或解交织方法的应用场景，该Turbo码的交织或解交织方法应用于LTE数据信道中，用于进行Turbo码的交织或解交织。下面结合附图对该Turbo码的交织或解交织方法进行详细的叙述。

[0039] 参考图1，本发明实施例的Turbo码的交织或解交织方法包括：

[0040] Step10：计算需要在源Turbo码序列中添加冗余比特位的个数；

[0041] Step20：根据交织器规则和冗余比特位的个数，推导出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射；

[0042] Step30：根据源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射，按照解Turbo码的特性要求，读取第一Turbo码序列，并输出第二Turbo码序列；其中，在第一Turbo码序列为源Turbo码序列时，第二Turbo码序列为交织后Turbo码序列；在第一Turbo码序列为交织后Turbo码序列时，第二Turbo码序列为源Turbo码序列。

[0043] 在上述的方案中，提供了一种新型快速交织或解交织的实现方案，通过计算需要在源Turbo码序列中添加冗余比特位的个数；之后根据交织器规则和冗余比特位的个数，推导出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射；再根据源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射，按照解Turbo码的特性要求，读取第一Turbo码序列，并输出第二Turbo码序列，进行由源Turbo码序列到交织后Turbo码序列的交织，或由交织后Turbo码序列到源Turbo码序列的解交织。在此过程中，不提前生成交织表，通过推导得到源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射。利用LTE交织中，如果添加的冗余比特位的个数确定了之后，就能够确定交织器规则中列变换之后索引k之前有几个冗余比特位就可以确定的特性，可以快速计算出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射。在处理时间上只需要做解交织部分，而无需做交织部分，从而缩短处理时间。数据是紧密排列的，不需要处理冗余比特位的时间，进一步缩短处理时间。另外，在资源只需要一块缓存即可，从而减少交织或解交织过程中占用的处理资源。下面结合附图对上述各个步骤进行详细的介绍。

[0044] 其中，交织主要目的是找到对应的索引号。上述的交织器规则可以包括如下的交织步骤。参考图2和图3，首先，根据源Turbo码序列索引和冗余比特位的个数N，获得添加冗余比特位后的第一过渡Turbo码序列索引。之后，根据交织器规则中的交织列数，计算第一过渡Turbo码序列索引所需的交织行数。之后，根据第一过渡Turbo码序列索引，按照先行后列的顺序填充源数据，生成交织表。接下来，根据交织表，生成第二过渡Turbo码序列索引。

之后,根据交织器规则中的列变换规则,生成列变换之后的第三过渡Turbo码序列索引。之后,删去第三过渡Turbo码序列索引中的冗余比特位,得到交织后Turbo码序列索引。便于快速且准确的推导得到源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射。

[0045] 上述Step10步骤中,在计算需要在源Turbo码序列中添加冗余比特位的个数时,示例性的,假设源Turbo码序列为 D_i ,其中,i表示源Turbo码序列中的索引号。第一过渡Turbo码序列为 $D_{i'}$,其中, i' 表示第一过渡Turbo码序列中的索引号。则, $i' = i + N_{\text{dummy}}$, N_{dummy} 表示需要在源Turbo码序列中添加冗余比特位的个数。便于快速计算得到需要在源Turbo码序列中添加冗余比特位的个数。上述根据源Turbo码序列索引和冗余比特位的个数N,获得添加冗余比特位后的第一过渡Turbo码序列索引时,计算冗余比特位的个数 N_{dummy} (可以分别为 n_0 , n_1 , n_2 , \dots , n_{N-1} ,所有数据都是NULL),在源Turbo码序列前填充 N_{dummy} 个<NULL>,生成添加冗余比特位后的第一过渡Turbo码序列索引。源Turbo码序列中的个数D是已知条件,其他参数都是未知条件。

[0046] 上述步骤Step20步骤中,在根据交织器规则和冗余比特位的个数,推导出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射时,示例性的,交织器规则中的交织列数可以为32。此时,在计算第一过渡Turbo码序列索引所需的交织行数时,计算第一过渡Turbo码序列索引所需的交织行数 $R = \text{ceil}(D/32)$,其中,D表示源Turbo码序列中的数据个数。便于快速计算得到第一过渡Turbo码序列索引所需的交织行数。

[0047] 上述步骤Step20步骤中,在根据交织器规则和冗余比特位的个数,推导出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射时,示例性的,可以假设第二过渡Turbo码序列为 D_j ,其中,j表示第二过渡Turbo码序列中的索引号。此时,在计算第二过渡Turbo码序列中的行索引号和列索引号时,可以采用如下的方式。计算第二过渡Turbo码序列中的行索引号 $\text{raw}_j = \text{floor}(i' / 32)$,其中, $\text{floor}(i' / 32)$ 表示 i' 除以32取商。计算第二过渡Turbo码序列中的列索引号 $\text{col}_j = i' \% 32$,其中, $i' \% 32$ 表示 i' 除以32取余数。便于快速计算得到第二过渡Turbo码序列中的行索引号和列索引号。

[0048] 上述步骤Step20步骤中,在根据交织器规则和冗余比特位的个数,推导出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射时,示例性的,上述的源Turbo码序列可以为系统比特码序列或第一校验码序列。此时,可以假设第三过渡Turbo码序列为 D_k ,其中,k表示第三过渡Turbo码序列中的索引号。在计算第三过渡Turbo码序列中的行索引号和列索引号时,可以采用如下的计算方式。计算第三过渡Turbo码序列中的行索引号 $\text{raw}_k = \text{raw}_j$ 。可以假设第三过渡Turbo码序列索引的列间变换矩阵为P。之后,可以按照如下公式,计算第三过渡Turbo码序列索引的列间变换矩阵P的反变换矩阵P' :

[0049] $P(x) = y, P'(y) = x$

[0050] 其中,x表示列变换前的列位置,y表示列变换后的列位置;

[0051] 再计算第三过渡Turbo码序列中的列索引号 $\text{col}_k = P'(\text{col}_j)$ 。在源Turbo码序列为系统比特码序列或第一校验码序列时,便于快速计算得到第三过渡Turbo码序列中的行索引号和列索引号。

[0052] 其中,上述系统比特码序列、第一校验码序列在交织过程中的列间置换模式反变换可以采用如下表1所示出的方式。

[0053] 表1-系统比特码序列、第一校验码序列在交织过程中的列间置换

	列数 $C_{subblock}^{CC}$	列间置换模式 $\langle P'(0), P'(1), \dots, P'(C_{subblock}^{CC} - 1) \rangle$
[0054]	32	[0,16,8,24,4,20,12,28,2,18,10,26,6,22,14,30,1, 17,9,25,5,21,13,29,3,19,11,27,7,23,15,31]

[0055] 上述步骤Step20步骤中,在根据交织器规则和冗余比特位的个数,推导出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射时,示例性的,上述的源Turbo码序列还可以为第二校验码序列。此时,交织器规则中,根据交织器规则中的列变换规则,生成列变换之后的第三过渡Turbo码序列索引时,需要根据交织器规则中的列变换规则,在列变换时循环右移了一列,生成列变换之后的第三过渡Turbo码序列索引。此时,第三过渡Turbo码序列的计算方式,与前述示出的源Turbo码序列可以为系统比特码序列或第一校验码序列时的计算方式存在差异。具体的,此时在计算第三过渡Turbo码序列中的行索引号和列索引号时,可以采用如下的计算方式。同样可以第三过渡Turbo码序列为 D_k ,其中,k表示第三过渡Turbo码序列中的索引号。同样可以计算第三过渡Turbo码序列索引的列间变换矩阵为P。之后,按照如下公式,计算第三过渡Turbo码序列索引的列间变换矩阵P的反变换矩阵 P' :

$$P(x) = y, P'(y) = x$$

[0057] 其中,x表示列变换前的列位置,y表示列变换后的列位置。

[0058] 示例性的,第二校验码序列由于编码做列变换时循环右移了一列,所以计算步骤有点区别。如下表2所示出的第二校验码序列的列间置换模式反变换。

[0059] 表2-第二校验码序列在交织过程中的列间置换模式

	列数 $C_{subblock}^{CC}$	列间置换模式 $\langle P'(0), P'(1), \dots, P'(C_{subblock}^{CC} - 1) \rangle$
[0060]	32	[31,0,16,8,24,4,20,12,28,2,18,10,26,6,22,14,3 0,1,17,9,25,5,21,13,29,3,19,11,27,7,23,15]

[0061] 之后,需要根据第二过渡Turbo码序列中的列索引号的不同,采用不同的推导方式计算得到第三过渡Turbo码序列中的行索引号,具体如下:

[0062] 判断第二过渡Turbo码序列中的列索引号 col_j 是否等于0。如果第二过渡Turbo码序列中的列索引号 $col_j=0$,则进一步判断第二过渡Turbo码序列中的行索引号 raw_j 是否等于0。在第二过渡Turbo码序列中的列索引号 $col_j=0$,且第二过渡Turbo码序列中的行索引号 $raw_j=0$ 时,第三过渡Turbo码序列中的行索引号 $raw_k=R-1$ 。在第二过渡Turbo码序列中的列索引号 $col_j=0$,且第二过渡Turbo码序列中的行索引号 $raw_j \neq 0$ 时,第三过渡Turbo码序列中的行索引号 $raw_k=raw_{j-1}$ 。

[0063] 如果第二过渡Turbo码序列中的列索引号 col_j 不等于0,即第二过渡Turbo码序列中的列索引号 $col_j \neq 0$,则第三过渡Turbo码序列中的行索引号 $raw_k=raw_j$ 。

[0064] 在三过渡Turbo码序列中的列索引号时,源Turbo码序列为第二校验码序列时的计算方式,与源Turbo码序列为系统比特码序列或第一校验码序列时的计算方式相同。具体的,都是计算第三过渡Turbo码序列中的列索引号 $col_k=P'(col_j)$ 。在源Turbo码序列为第二校验码序列时,便于快速计算得到第三过渡Turbo码序列中的行索引号和列索引号。

[0065] 上述步骤Step20步骤中,在根据交织器规则和冗余比特位的个数,推导出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射时,示例性的,还可以进一步计算第三过渡Turbo码序列 D_k 中 k 之前的冗余比特位个数 n_k 。假设交织后Turbo码序列为 $D_{k'}$,其中, k' 表示交织后Turbo码序列中的索引号。之后,可以按照如下公式,计算 k' 与 i 之间的点对点映射:

[0066] $k' = \text{col}_k * R + \text{raw}_k - n_k$

[0067] 便于快速得到第三过渡Turbo码序列 D_k 中 k 之前的冗余比特位个数 n_k ,进而快速且准确的推导得到源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射。

[0068] 其中,上述在计算第三过渡Turbo码序列 D_k 中 k 之前的冗余比特位个数 n_k 时,可以根据列变换规则,推算出每个冗余比特位所在的列位置。之后,根据每个冗余比特位所在的列位置,推算出第三过渡Turbo码序列 D_k 中 k 之前的冗余比特位个数 n_k 。便于快速且准确的得到第三过渡Turbo码序列 D_k 中 k 之前的冗余比特位个数 n_k 。具体的,可以利用枚举和列变换 P 提前生成32个32bit的向量,每个向量有32bit数据代表32列,其中1表示当前位置是冗余比特位。根据 N_{dummy} 选取其中一个向量,再根据 col_k ,计算向量 $0 \sim \text{col}_k$ 列有几个冗余比特位,累加计算出 n_k 。

[0069] 接下来,参考图1,在Step30中,根据源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射,按照解Turbo码的特性要求,读取第一Turbo码序列,并输出第二Turbo码序列。其中,在第一Turbo码序列为源Turbo码序列时,第二Turbo码序列为交织后Turbo码序列,此时,所进行的是交织过程。在第一Turbo码序列为交织后Turbo码序列时,第二Turbo码序列为源Turbo码序列,此时所进行的是解交织过程。如图4所示,在上述的第一Turbo码序列为乱序的交织后Turbo码序列时,可以将该第一Turbo码序列存储到`pre_buffer`缓存模块中,采用本申请示出的解交织方法解交织出的顺序的源Turbo码序列之后,上述的第二Turbo码序列为经过解交织得到的源Turbo码序列,此时,可以直接输入到解Turbo码模块中,而无需使用额外的缓存模块。参考图5,上述的第一Turbo码序列可以来自于数据流,而无需使用额外的缓存模块进行缓存,此时,只需采用一个`after_buffer`缓存模块,来缓存交织后得到的第二Turbo码序列即可。可以看出,本申请中在应用时,只需使用一个缓存模块即可,同时,不需要处理交织的时间,也不需要处理冗余比特位的时间,能够缩短处理时间,减少在此过程中所占用的处理资源。

[0070] 在上述示出的各种实施方式中,提供了一种新型快速交织或解交织的实现方案,通过计算需要在源Turbo码序列中添加冗余比特位的个数;之后根据交织器规则和冗余比特位的个数,推导出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射;再根据源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射,按照解Turbo码的特性要求,读取第一Turbo码序列,并输出第二Turbo码序列,进行由源Turbo码序列到交织后Turbo码序列的交织,或由交织后Turbo码序列到源Turbo码序列的解交织。在此过程中,不提前生成交织表,通过推导得到源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射。利用LTE交织中,如果添加的冗余比特位的个数确定了之后,就能够确定交织器规则中列变换之后索引 k 之前有几个冗余比特位就可以确定的特性,可以快速计算出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射。在处理时间上只需要做解交织部分,而无需做交织部分,从而缩短处理时间。数据是紧密排列的,不需要处理冗余比

特位的时间,进一步缩短处理时间。另外,在资源只需要一块缓存即可,从而减少交织或解交织过程中占用的处理资源。

[0071] 另外,本发明实施例还提供了一种Turbo码的交织或解交织器件,该Turbo码的交织或解交织器件包括:冗余比特位计算模块、推导模块、交织或解交织模块。其中,冗余比特位计算模块用于计算需要在源Turbo码序列中添加冗余比特位的个数。推导模块用于根据交织器规则和冗余比特位的个数,推导出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射。交织或解交织模块用于根据源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射,按照解Turbo码的特性要求,读取第一Turbo码序列,并输出第二Turbo码序列;其中,在第一Turbo码序列为源Turbo码序列时,第二Turbo码序列为交织后Turbo码序列;在第一Turbo码序列为交织后Turbo码序列时,第二Turbo码序列为源Turbo码序列。

[0072] 在上述的方案中,提供了一种新型快速交织或解交织的实现方案,通过计算需要在源Turbo码序列中添加冗余比特位的个数;之后根据交织器规则和冗余比特位的个数,推导出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射;再根据源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射,按照解Turbo码的特性要求,读取第一Turbo码序列,并输出第二Turbo码序列,进行由源Turbo码序列到交织后Turbo码序列的交织,或由交织后Turbo码序列到源Turbo码序列的解交织。在此过程中,不提前生成交织表,通过推导得到源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射。利用LTE交织中,如果添加的冗余比特位的个数确定了之后,就能够确定交织器规则中列变换之后索引k之前有几个冗余比特位就可以确定的特性,可以快速计算出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射。在处理时间上只需要做解交织部分,而无需做交织部分,从而缩短处理时间。数据是紧密排列的,不需要处理冗余比特位的时间,进一步缩短处理时间。另外,在资源只需要一块缓存即可,从而减少交织或解交织过程中占用的处理资源。

[0073] 另外,上述的交织或解交织器件还可以包括缓存模块,缓存模块用于缓存第一Turbo码序列或第二Turbo码序列。在资源只需要一块缓存即可,从而减少交织或解交织过程中占用的处理资源。

[0074] 再者,本发明实施例还提供了一种通信芯片,该通信芯片包括上述任意一种Turbo码的交织或解交织器件。提供了一种新型快速交织或解交织的实现方案,通过计算需要在源Turbo码序列中添加冗余比特位的个数;之后根据交织器规则和冗余比特位的个数,推导出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射;再根据源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射,按照解Turbo码的特性要求,读取第一Turbo码序列,并输出第二Turbo码序列,进行由源Turbo码序列到交织后Turbo码序列的交织,或由交织后Turbo码序列到源Turbo码序列的解交织。在此过程中,不提前生成交织表,通过推导得到源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射。利用LTE交织中,如果添加的冗余比特位的个数确定了之后,就能够确定交织器规则中列变换之后索引k之前有几个冗余比特位就可以确定的特性,可以快速计算出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射。在处理时间上只需要做解交织部分,而无需做交织部分,从而缩短处理时间。数据是紧密排列的,不需要处理冗余比特位的时间,进一步

缩短处理时间。另外,在资源只需要一块缓存即可,从而减少交织或解交织过程中占用的处理资源。

[0075] 此外,本发明实施例还提供了一种通信装置,该通信装置包括上述任意一种通信芯片。提供了一种新型快速交织或解交织的实现方案,通过计算需要在源Turbo码序列中添加冗余比特位的个数;之后根据交织器规则和冗余比特位的个数,推导出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射;再根据源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射,按照解Turbo码的特性要求,读取第一Turbo码序列,并输出第二Turbo码序列,进行由源Turbo码序列到交织后Turbo码序列的交织,或由交织后Turbo码序列到源Turbo码序列的解交织。在此过程中,不提前生成交织表,通过推导得到源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射。利用LTE交织中,如果添加的冗余比特位的个数确定了之后,就能够确定交织器规则中列变换之后索引k之前有几个冗余比特位就可以确定的特性,可以快速计算出源Turbo码序列索引与交织后Turbo码序列索引之间的点对点映射。在处理时间上只需要做解交织部分,而无需做交织部分,从而缩短处理时间。数据是紧密排列的,不需要处理冗余比特位的时间,进一步缩短处理时间。另外,在资源只需要一块缓存即可,从而减少交织或解交织过程中占用的处理资源。

[0076] 以上所述,仅为本发明的具体实施方式,但本发明的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本发明揭露的技术范围内,可轻易想到的变化或替换,都应涵盖在本发明的保护范围之内。因此,本发明的保护范围应该以权利要求的保护范围为准。

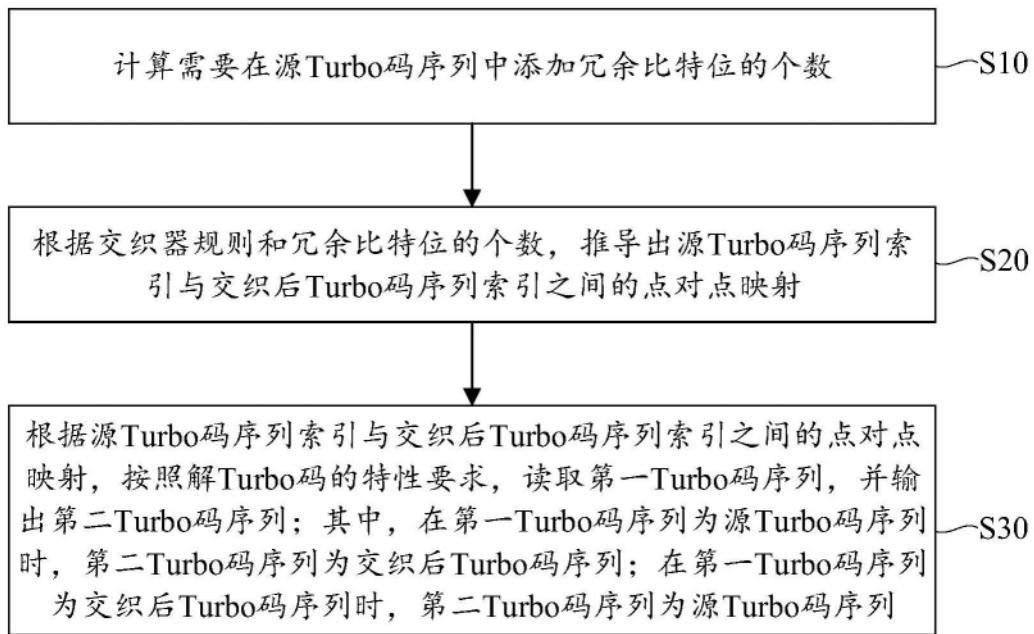


图1

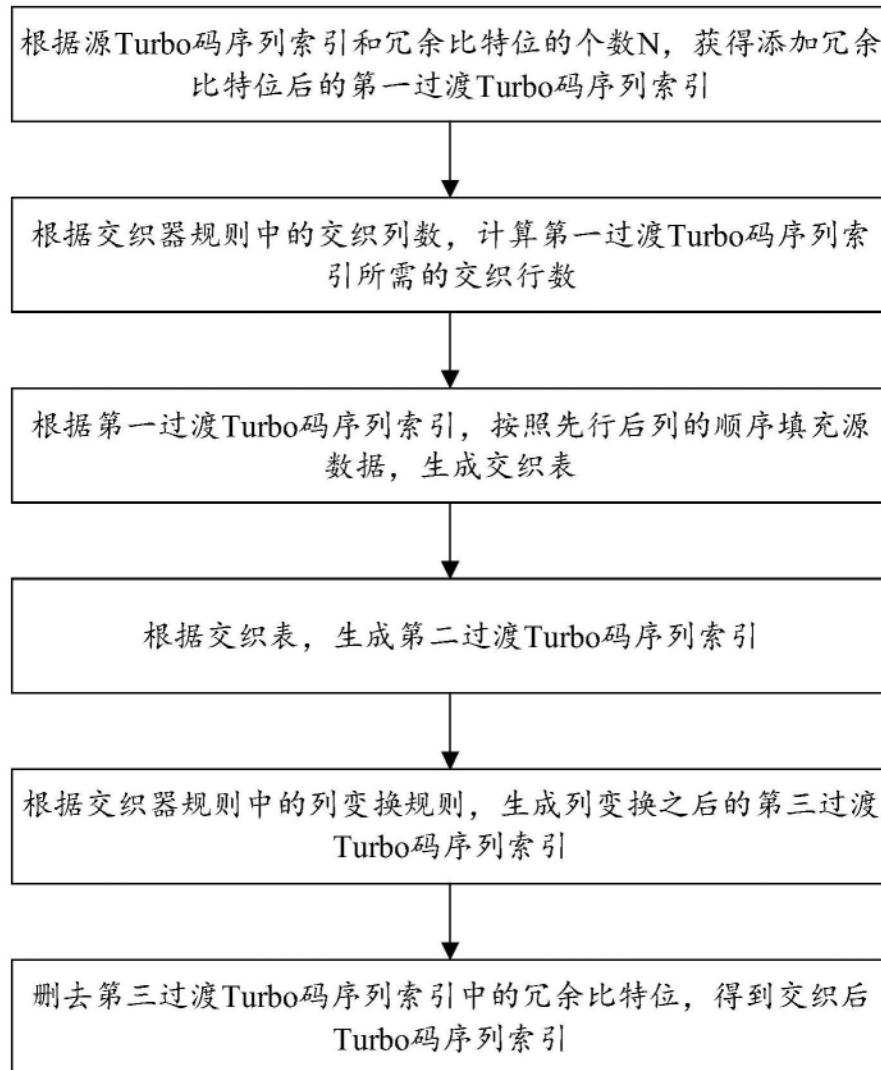


图2



图3

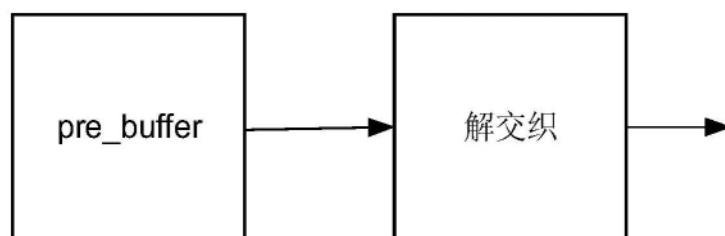


图4

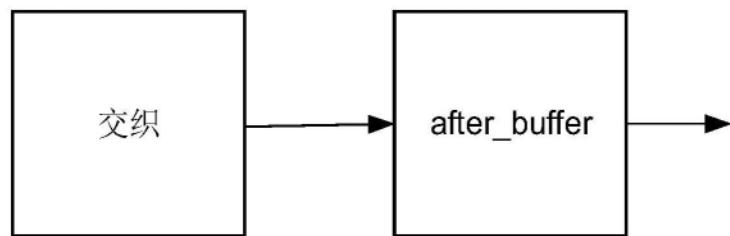


图5