

[19] 中华人民共和国国家知识产权局

[51] Int. Cl⁷

H03M 13/29

H03M 13/27 H04L 1/00



[12] 发明专利申请公开说明书

[21] 申请号 02805651.5

[43] 公开日 2004 年 5 月 5 日

[11] 公开号 CN 1494770A

[22] 申请日 2002.2.26 [21] 申请号 02805651.5

[30] 优先权

[32] 2001. 2. 28 [33] US [31] 60/272,123

[32] 2001. 5. 10 [33] US [31] 09/853,332

[86] 国际申请 PCT/US2002/006030 2002.2.26

[87] 国际公布 WO02/069504 英 2002.9.6

[85] 进入国家阶段日期 2003.8.28

[71] 申请人 高通股份有限公司

地址 美国加利福尼亚州

[72] 发明人 I·姚 D·-S·修

[74] 专利代理机构 上海专利商标事务所

代理人 李家麟

权利要求书 4 页 说明书 25 页 附图 10 页

[54] 发明名称 适于 TURBO 解码器的交错器

[57] 摘要

使用多个查表来为 Turbo 码交错器有效地产生存储器地址的技术。交错器包括一个存储单元、表单组和地址产生器。存储单元在表示 $R \times C$ 数组的位置存储数据分组的 K 个元素，这些元素以第一次序(例如线性)被存储，并以第二次序(例如交错)被提供。第一组表单存储用于执行此阵列行变换的序列(如行间变换序列 P_A 、 P_B 、 P_C 和 P_D)，以从第一个次序映射到第二个次序。第二组表单存储用于执行列变换的序列(如行内基数序列和素数序列)。基于存储在表单中的序列，地址产生器接收适于第一个次序的第一个地址，并产生相应的适于第二个次序的第二个地址。

I S S N 1 0 0 8 - 4 2 7 4

1、利用多个已给定 K 和已知 R 、 C 和 $pind$ 的表单的 Turbo 交错器，派生出 W-CDMA 兼容的 turbo 交错器地址，此交错器包括：

- 一个包含多个行间变换表单的行间表单；
- 一个素数表单；
- 可运行以产生素数下标信息的第一个单元；
- 一个行内基数序列表单（INTRABASE）；
- 一个行内基数序列存取表单（INTRADDR）；
- 一个 (q) 序列表单；及

为产生 W-CDMA 兼容的 turbo 交错器地址，与第一个单元和每个表单连接的交错器硬件逻辑。

2、适于联接卷积（Turbo）码的交错器包括：

一个存储单元，配置成在代表二维数组的位置存储数据分组的多个 (K) 元素，所述元素以第一个次序被存储到存储单元中且以第二个次序从存储单元提供；

包含至少一个表单的第一组表单，配置成存储用于实现二维数组的行变换以从第一个次序映射到第二个次序的值的的第一组序列；

包含至少一个表单的第二组表单，配置成存储用于实现二维数组的列变换以从第一个次序映射到第二个次序的值的第二组序列；及

一个和存储单元和第一和第二组表单联接的地址产生器，配置成部分地基于在第一和第二组表单中存储的序列，以第一个次序接收第一个地址，以第二个次序产生相应的第二个地址。

3、如权利要求 2 所述的交错器，其特征在于，第一个次序是线性的，且对应于在数据分组中线性位置上的 K 个元素的存取，第二个次序是交错的，且对应于在数据分组中交错位置上的 K 个元素的存取。

4、如权利要求 2 所述的交错器，其特征在于，第一个和第二个次序都是由 W-CDMA 标准定义的。

5、如权利要求 2 所述的交错器，其特征在于，进一步包括：

包含至少一个表单的第三组表单，配置成存储用于以第二个次序产生第二个地址的参数值。

6、如权利要求 5 所述的交错器，其特征在于，所述存储在包含至少一个表单的第三组表单的参数值，包括为用于派生第二个次序定义的一组素数。

7、如权利要求 2 所述的交错器，其特征在于，包含至少一个表单的第一组表单包括配置成存储多个第一个系列值的第一个（INTROW）表单，其中每个第一个序列被用于第一个次序中的行数映射到第二个次序中相应的行数。

8、如权利要求 2 所述的交错器，其特征在于，包含至少一个表单的第二组表单包括配置成存储多个第二个系列值的第二个（INTRABASE）表单，其中每个第二个序列将在第一个次序中的列下标映射到第二个次序中相应的列数。

9、如权利要求 8 所述的交错器，其特征在于，所述多个第二个序列对应于多个素数，且所述一个第二个序列是基于 K 值被选出用于数据分组。

10、如权利要求 8 所述的交错器，其特征在于，所述包含至少一个表单的第二组表单进一步包括一个配置成存储多个第三个序列值的第三个（Y）表单，其每个第三个序列包括用于派生列下标的值。

11、如权利要求 2 所述的交错器，其特征在于，所述地址产生器可配置成派生对应于第一个地址的输入行数和输入列数，并基于第一和第二组表单的一部分，进一步分别派生对应于输入行和列数的交错行数和交错列数。

12、如权利要求 2 所述的交错器，其特征在于，所述地址产生器可配置成为每个时钟周期提供至少一个有效的第二个地址。

13、如权利要求 2 所述的交错器，其特征在于，所述地址产生器包括

至少二个地址发生单元，每个单元配置成接收各自的第一个地址，并产生对应于接收到的第一个地址的第二个地址。

14、如权利要求 13 所述的交错器，其特征在于，所述连续的第一个地址被提供给至少二个地址发生单元。

15、如权利要求 2 所述的交错器，其特征在于，多个元素是来自 Turbo 解码过程的中间结果。

16、地址产生器可运行以为配置成存储联接卷积 (Turbo) 码的数据的存储单元产生地址，其特征在于，所述数据分组的多个 (K) 元素被存储在表示二维数组存储单元中的位置，且所述元素被以第一个次序存储到存储单元中，以第二个次序从存储单元提供，地址产生器包括：

第一个单元，配置成对应第一个次序第一个地址，为二维数组提供一个输入行数和一个输入列数；

一个联接第一个单元的行映射单元，配置成接收至少输入行数，并提供相应变换后的行数；

一个联接第一个单元的列映射单元，配置成接收至少输入列数，并提供相应变换后的列数；

一个与行和列映射单元联接的组合单元，配置成接收和组合变换后的行和列数，以为第二个次序产生第二个地址；和

包含至少一个表单的第一组表单，配置成存储用于实现二维数组的行变换以从第一个次序映射到第二个次序和第二组的值的第一组序列；和

包含至少一个表单的第二组表单，配置成存储实现二维数组的列变换以从第一个次序映射到第二个次序的值的第二组序列；和

其中基于从表单的第一和第二组接收到的值派生出所述变换后的行和列数。

17、如权利要求 16 所述的地址产生器，其特征在于，所述第一个和第二个次序是由 W-CDMA 标准定义的。

18、Turbo 解码器包括：

一个组成解码器，配置成接收和依据特殊组成码元解码已编码的比特，以产生中间结果；

一个和组成解码器联接的存储单元，配置成在表示二维数组的位置存储中间结果，所述中间结果以第一个次序被存储到存储单元中，并以第二个次序从存储单元提供；和

一个和存储单元联接的地址产生器，配置成部分地基于存储在第一组表单中值的序列，以第一个次序接收第一个地址，并以第二个次序产生第二个地址，所述序列被用于实现二维数组的行和列变换，以从第一个次序映射到第二个次序。

19、如权利要求 18 所述的 Turbo 解码器，其特征在于，第一个和第二个次序是由 W-CDMA 标准定义的。

20、如权利要求 18 所述的 Turbo 解码器，其特征在于，地址产生器被配置成为每个时钟周期提供至少一个有效第二个地址。

21、为联接的卷积（Turbo）码产生地址的方法，其特征在于，数据分组的多个（K）元素被存储表示在二维数组的存储单元中的位置，且元素被以第一个次序存储到存储单元中，以第二个次序从存储单元提供，所述方法包括：

符合第一个次序的第一个地址，为二维数组判决输入行数和输入列数；

存取包含至少一个表单的第一组表单，以检索用于实现二维数组行变换以从第一个次序映射到第二个次序的包含至少一个序列的第一组序列的值；

基于包含至少一个序列的检索到的第一组，映射输入行数为变换行数；

存取包含至少一个表单的第二组表单，以检索用于实现二维数组列变换以从第一个次序映射到第二个次序的包含至少一个序列的第二组序列的值；

基于包含至少一个序列的检索到的第二组，映射输入列数为变换列数；

组合变换后的行和列数来以第二个次序产生第二个地址。

适于 TURBO 解码器的交错器

相关申请的交叉引用

此申请请求 2001 年 2 月 28 日提交的美国临时专利申请号 60/272,123 的权益。

背景

领域

本发明涉及数据通信。更特别地，本发明涉及新颖并且改良的适于存储 Turbo 解码器中间结果的交错器。

背景

无线通信系统被广泛用于提供各种类型的通信，例如语音、数据等等。这些系统可以基于码分多址（CDMA）、时分多址（TDMA），或某些其他的多址技术。CDMA 系统具有一些比其它类型的系统优良的优点，包括增加系统容量。

CDMA 系统可设计成遵循一个或多个 CDMA 标准，例如 IS-95、cdma2000 和 W-CDMA 这些在业内众所周知的标准。每个标准具体地定义了先于前向和反向链路传输的数据处理过程。例如，将语音信息以一个特殊的数据速率编码，格式化成已定义的帧格式，并依据一个特殊处理机制（例如为纠错和/或检波、交错等等）处理。由特殊标准（如 cdma2000 标准）定义的帧格式和处理过程可能不同于其他标准（如 W-CDMA 标准）。

W-CDMA 标准采用并行联接的卷积编码器（常称为 Turbo 编码器），Turbo 编码器可被选择用于在传输之前对码元段（即数据分组）编码。Turbo 编码器采用二个组成编码器，其并行运行，并与一个码元交错器联接。码元交错器根据特别定义的交错机制，混洗（即交错）在此分组中的信息比特。一个编码器编码在此分组中的信息比特，以产生第一个奇偶校验比特序列，另一个编码器编码混洗后的信息比特，以产生第二个奇偶校验比特序列。此信息比特和所有的或某些在第一和第二序列中的奇偶校验比特被发送。

互补（和计算密集的）Turbo 解码过程在接收单元执行。对于每个 Turbo

编码的分组，接收到的比特首先被存储到一个缓存器中。然后从存储器中取得第一个编码器的信息和奇偶校验比特，并基于第一个组成码元解码，以提供表示信息比特检测值中可信度的调节的“非本征”信息。然后将包括来自第一个解码器的非本征信息的中间结果以匹配在发送单元使用的码元交错的交错次序存储到存储单元。

然后来自第二个编码器的中间结果和奇偶校验比特，被从它们相应的信息来源检索出来，并基于第二个组成码元解码，以提供表示信息比特检测值的可信度的进一步调节的非本征信息。然后将包含来自第二个解码器非本征信息的中间结果以与在传输单元中使用的码元交错互补的解交错次序，存储到存储单元。此中间结果被第一个组成解码器的下一个循环使用。由第一个和第二个组成解码器执行的解码过程被重复多次以产生最后结果。

对于被解码的每个信息比特，存取存储单元来检索由先前解码过程为此比特产生的中间结果（如果有的话）。为每个解码后的比特产生的中间结果也被存储回存储单元以在后续解码过程中使用。这样当分组中的比特被解码时，存储单元被连续地存取。对于每次存储器的存取，需要产生正确的地址，以使得正确的中间结果被从存储单元检索（为了读取），或存储到正确位置（为了写入）。

如上所示，有效地产生存储器存取地址对有效的 Turbo 解码过程是非常需要的，特别是按照由 W-CDMA 标准定义的复杂交错机制。

概要

本发明从各方面提供技术，以有效地产生执行由 W-CDMA 标准定义的 Turbo 码元交错过程所需的存储器地址。一个方面，为了加速地址产生，提供多个查表（LUT_s）来存储用于产生交错地址的各种值的序列。这些表单的使用加速地址计算，并允许所需地址在更少的时间内被产生。在另一个方面，提供技术以基于这些表单，有效地产生交错地址。交错地址产生技术可以用于 Turbo 解码过程，特别有利于计算密集的 Turbo 解码过程。快速的地址产生对有效的 Turbo 解码过程是至关重要的，特别是在支持高数据速率，并按照 Turbo 解码过程的循环本质的时候。

本发明一个特殊实施例为联接的卷积（Turbo）码提供一个交错器。交错器包括一个存储单元，至少一个表单的第一和第二组和一个地址产生器。存储

单元为一个数据分组，在表示二维 ($R_x C$) 数组的位置，存储多个 (K 个) 元素 (如 Turbo 解码过程的中间结果)，元素以第一个次序 (如线性) 被存储到存储单元中，并以第二个次序 (如交错) 被从存储单元提供。第一组表单存储用于执行 $R_x C$ 数组的行变换，以从第一个次序映射到第二个次序的值的的第一组序列。对于 W-CDMA 标准来说，这些序列可以包括行间变换序列 P_A 、 P_B 、 P_C 和 P_D 。第二组表单存储用于执行 $R_x C$ 数组列变换的值的第二组序列。对于 W-CDMA 标准来说，这些序列可以包括行内基数序列 $c(i)$ 和素数序列 q_j 或变换后的素数序列 P_j ，其中所有的序列都用于执行列变换。地址产生器以第一个次序接收第一个地址，并基于在第一个和第二组表单中存储的序列的一部分，以第二个次序产生相应的第二个地址。

本发明的各个方面、实施例和特点将在下文更详细地描述。

附图简述

结合附图，通过下面给出的详细描述，本发明特点、本质和优点将变得更加明显，图中相同的参考号在整个说明中对应一致：

图 1 是能够实现本发明各种方面的通信系统的简化方框图；

图 2A 和 2B 是分别在发送单元和接收单元的信号处理过程图，适于遵循 W-CDMA 标准的下行链路数据传输；

图 3 是由 W-CDMA 标准定义的 Turbo 编码器图；

图 4 是特殊 Turbo 解码器设计方框图；

图 5 是特殊 Turbo 解码器实现方框图；

图 6A 是显示次序的图表，在此次序中，码元段中的比特被写入 $R_x C$ 数组，适于由 W-CDMA 标准定义的 Turbo 交错机制；

图 6B 是显示码元段大小 K 是 40 的例子的交错过程图；

图 7 显示可以用于加速地址产生的表单，适于由 W-CDMA 标准定义的交错机制；

图 8 是能够依据本发明的一个实施例，提供交错地址的地址产生器的方框图；

图 9 是能够产生顺序输入地址的序列的输入行和列数的序列的单元图；和

详细描述

图 1 是能够实现本发明各个方面的通信系统 100 的一个简化方框图。在一个特殊实施例中，通信系统 100 是遵循 W-CDMA 标准的 CDMA 系统。在发送单元 110，典型的是在方框中，数据从数据源 112 发送到格式化、编码和处理数据以产生一个或多个模拟信号的发送 (TX) 数据处理器 114。模拟信号然后被提供给 (正交) 调制、滤波、放大及上变频信号以产生调制信号的发送器 (TMTR) 116。调制信号然后通过一个或多个天线 118 (在图 1 中只显示一个) 发送到一个或多个接收单元。

在接收单元 130，发送的信号由一个或多个天线 132 接收下来 (在此只显示一个)，并提供给一个接收器 (RCVR) 134。在接收器 134 中，接收到的信号被放大、过滤、下变频、(正交) 解调及数字化以生成样本。样本然后被接收 (RX) 数据处理器 136 处理和解码以恢复此发送的数据。在接收器单元 136 的处理过程和解码过程，以与在发送单元 110 执行的处理过程和编码过程互补的方式执行。恢复的数据然后被提供给数据汇 138。

上述信号处理过程支持单向的语音、视频、分组数据、消息和其他类型的通信。双向通信系统支持双向数据传输。然而，简单起见，图 1 没有显示另一个方向的信号处理过程。

图 2A 是依据 W-CDMA 标准，为下行链路数据传输在发送单元 110 的信号处理过程图。下行链路指从基站到用户终端 (或用户设备 (UE)) 的传输，上行链路指从用户终端到基站的传输。图 2A 中显示的信号处理过程一般由图 1 中的发送数据处理器 114 执行。W-CDMA 系统的上方信号发送层支持许多传送信道的并发传输，每个传送信道能够负载特殊通信 (如语音、视频、数据等等) 的数据。在亦称为传送程序块的程序块中，每个传送信道的数据提供给各自的传送信道处理部分 210。

在传送信道处理部分 210 中，在程序块 212 每个传送程序块被用于计算循环冗余码校验 (CRC) 比特。CRC 比特被附加到传送程序块上，并在接收单元被用来做错误检测。多个 CRC 编码程序块然后在程序块 214 中被串联地联接在一起。如果联接后的比特总数大于编码数据块的最大大小，则比特被分解成许多 (相同大小的) 码元数据块。每个码元数据块然后在程序块 216 中以特定的编码机制编码 (如卷积编码)，或者根本不编码。

然后在程序块 218，对码元比特执行速率匹配。速率匹配是依据由更高信号层指定的速率匹配属性来执行的。在上行链路中，比特被重复或忽略 (即删

除)，如此被发送的比特数和有效的比特数相匹配。在下行链路中，未使用的比特位置在程序块 220 用不连续的传输（DTX）比特来填充。DTX 比特表示何时传输应该关闭，且不被实际发送。

然后在程序块 222，依据特定的交错机制比特被交错，以提供时间差异。依据 W-CDMA 标准，执行交错过程的时间间隔可以从一组可能的时间间隔（即 10 兆秒、20 兆秒、40 兆秒或者 80 兆秒）中选出。交错时间间隔同样称为传输时间间隔（TTI）。TTI 与每个传送信道的属性相关，并且依据 W-CDMA 标准，在一个通信会话的期间内不改变。如在此所使用的，“通信量”包括在特定传送信道的一个 TTI 中的比特。

当所选的 TTI 大于 10 兆秒时，在 224 程序块，通信量被分解并映射到连续的传送信道无线电帧。每个传送信道无线电帧对应一个在无线电帧期间（10 兆秒）的传输。依据 W-CDMA 标准，通信量可能在 1、2、4 或 8 个无线电帧期间被交错。

来自所有活动传送信道处理部分 210 的无线电帧然后在 232 程序块串连地复用到一个编码复合传送信道（CCTrCH）中。在程序块 234，DTX 比特然后可被插入到复用的无线电帧，如此发送的比特数和用于数据传输的物理信道上有效的比特数相匹配。如果使用超过一个的物理信道，在程序块 236 中，比特在物理信道间被分割。特殊的物理信道能负载不同 TTI 的传送信道。每个物理信道的每个无线电帧期间的比特然后在程序块 238 被交错，以提供额外的时差。交错的物理信道无线电帧然后在程序块 240，映射到它们各自的物理信道。产生适合传输到用户终端的调制信号的后续信号处理过程在业内众所周知，在此不再描述。

图 2B 是依据 W-CDMA 标准，为下行链路数据传输在接收单元 130 的信号处理过程图。图 2B 中所示的信号处理过程与图 2A 中的互补，一般由图 1 中的接收数据处理器 136 执行。最初，已调制的信号被接收、均衡、数字化及处理，来为用于数据传输的物理信道提供码元。每个码元有一个特定的分辨率（如 4 比特或更多），并对应一个发送比特。每个物理信道每个无线电帧中的码元在程序块 252 中解交错，来自所有物理信道的解交错码元在程序块 254 被联接。对于下行链路传输，没有发送的比特在程序块 256 中被检测和移除。码元然后在程序块 258 中解复用到各种传送信道中。每个传送信道的无线电帧然后提供给各自传送信道处理部分 260。

在传送信道处理部分 260 中，传送信道无线电帧在程序块 262 被联接成通信量。每个通信量包括一个或多个传送信道无线电帧，并且对应用于发送单元的特殊 TTI。每个通信量内的码元在程序块 264 被解交错，未发送的码元在程序块 266 被移除。然后在程序块 268 执行反向速率匹配以积累重复码元，为已击穿的码元插入“无须关注”。然后在程序块 270 通信量中每个已编码的数据块被解码。解码后数据块然后在程序块 272 中被联接，并分解到各自的传送程序块中。每个传送程序块然后在程序块 274 使用 CRC 比特检查误码。

图 3 是依据由 W-CDMA 标准定义的并联卷积编码器 300（也称为 Turbo 编码器）的图。Turbo 编码器 300 可用于在图 2A 程序块 216 中执行信道编码过程。Turbo 编码器 300 包括一对组成编码器 312a 和 312b，码元交错器 314 和击穿器和复用器 316。码元交错器 314 依据由 W-CDMA 标准定义的特殊交错机制来接收并交错码元段中的信息比特（即数据分组），下面将进一步详细描述。

每个组成编码器 312 接收线性次序或交错的信息比特，用已定义的组成码元来编码接收到的信息比特，并提供一个奇偶校验比特序列。击穿和复用器 316 来自编码器 312a 和 312b 的信息比特和奇偶校验比特，击穿（即删除）零个或多个奇偶校验比特以获取期望数量的比特，复用未击穿的信息比特和奇偶校验比特成一个已编码比特序列。

每个组成编码器 312 包括 3 个串联延迟单元 322，4 个模-2 加法器 324 和一个开关 326。最初，延迟单元 322 的状态被设定在 0，开关 326 在上方的位置。然后，对数据分组中的每个信息比特，加法器 324a 把信息比特 x 和来自加法器 324d 的输出比特做模-2 加法，并把结果提供给延迟单元 322a。加法器 324b 和 324c 接收，并把来自加法器 324a 和延迟单元 322a 和 322c 的比特做模-2 加法，并提供奇偶校验比特 y 。加法器 324d 对来自延迟单元 322b 和 322c 的比特做模-2 加法。

当数据分组中的所有 N 信息比特编码后，开关 326 移到向下位置，且 3 个零（“0”）末尾比特提供给组成编码器 312a。组成编码器 312a 然后编码这 3 个末尾比特并提供 3 个末尾奇偶校验比特。

对于每个 N 信息比特的分组，组成编码器 312a 提供 N 奇偶校验比特 y 和最前 6 个末尾奇偶校验比特，组成编码器 312b 提供 N 奇偶校验比特 z 和最后 6 个末尾奇偶校验比特。对于每个分组，击穿和复用器 316 接收 N 信息比特，来自编码器 312a 的 $N+6$ 奇偶校验比特和来自编码器 312b 的 $N+6$ 奇偶校验比特。

击穿和复用器 316 可以击穿许多奇偶校验比特来提供所需的由未击穿信息和奇偶校验比特组成的编码比特数量。

图 4 是 Turbo 解码器 400 的设计方框图。在这个设计中，Turbo 解码器 400 包括 2 个组成解码器 410a 和 410b、码元交错器 412、码元解交错器 414 和检测器 416。每个解码器 410 典型地作为软输入/软输出 (SISO) 最大后验 (MAP) 解码器来实现。然而，其他类型的解码器同样可以使用，诸如实现软输出 Viterbi 算法的解码器 (SOVA)。编码器的设计典型地依赖于在发送器单元使用的特殊 Turbo 编码机制。

对应于发送编码比特的接收到的 (soft) 比特由信道解交错器 (没有在图 4 中显示) 解交错，以撤回在发送器单元 (图 2A 中程序块 222 和 238) 所执行的第一和第二个交错。对于每个要解码的数据分组，信道解交错比特根据需要被提供给解码器 410a 和 410b。

在图 4 所示的实施例中，累加器 408a 接收并累加 $LLR(APP^0)$ ， $LLR(x')$ ，和来自解交错器 414 的非本征信息，以为解码器 410a 提供一个先验概率 (APP)。 $LLR(APP^0)$ 是源自信息比特的一个根本假设得出的对数可能比。如果数据分组中的每个信息比特假设同等地可能为零 (“0”) 或一 (“1”)，则对于分组中所有接收到的比特， $LLR(APP^0)$ 等于 0，且任何涉及 $LLR(APP^0)$ 的部分被忽略。来自解交错器 414 的非本征信息在第一个解码循环中设为 0。 $LLR(x')$ 是接收到的信息比特 x 的对数可能比。每个接收到的信息和奇偶校验比特的 LLR ， b_m ，可以计算为：

$$LLR(b_m) = \log \{ P(b_m = 0) / P(b_m = 1) \}$$

接收比特的 LLR 是比特为零的概率和比特为一的概率之比的对数。对每个接收比特，概率 $P(b_m = 0)$ 和 $P(b_m = 1)$ 典型地基于那个比特的 soft 数值。一个消除的 LLR (即击穿比特) 是对比特是零或一的相同可能的预示 (即 $LLR=0$)。

解码器 410a 从累加器 408a 接收 APP，从第一组成编码器接收到奇偶校验比特的 LLR ， $LLR(y')$ 。 $LLR(y')$ 包括去除击穿 (即未发送的) 奇偶校验比特，如果有的话。解码器 410a 然后依照 MAP 算法解码 APP 和 $LLR(y')$ ，来产生一个后验概率。APP 然后通过累加器 408b 被从后验概率中减去，以提供非本征信息， e_1 ，它表示对由接收奇偶校验比特 y' 投送的接收信息比特 x' 的值的可信度的校

正/调整。

来自累加器 408b 的非本征信息 e_1 和信息比特 LLRs、 $LLR(x')$ 相加，中间结果（即下一个解码器的 APP）被存储到码元交错器 412。码元交错器 412 实现和用于 Turbo 编码器相同的码元交错机制（如图 3 中用于码元交错器 314 的相同机制）。

解码器 410b 接收来自交错器 412 的交错后的 APP，和来自第二组成编码器的接收奇偶校验比特的 LLR， $LLR(z')$ 。解码器 410b 然后依照 MAP 算法解码 APP 和 $LLR(z')$ ，以产生后验概率。APP 然后通过累加器 408b 被从后验概率中减去，以提供非本征信息 e_2 ，其表示由接收奇偶校验比特 z' 投送的接收信息比特 x' 的值的可信度的进一步校正/调整。非本征信息 e_2 包含来自解码器 410b 的存储到码元解交错器 414 的中间结果。解交错器 414 实现和用在交错器 412 的交错机制互补的解交错机制。

信息比特 LLR 的解码过程循环多次（如 6、8、10 或可能更多次）。每次循环，信息比特检测到的值就获得更大的可信度。当所有的解码过程循环完毕，最终的 LLR 被提供给检测器 418，检测器 418 基于它们的 LLR 为接收信息比特提供硬判决值（即“0s”和“1s”）。

图 5 是依据本发明的实施例的 Turbo 解码器 500 的方框图。Turbo 解码器 500 是图 4 中 Turbo 解码器 400 的一个特殊实现。在这个实施例中，Turbo 解码器 500 包括输入接口单元 506，本地存储/组合器 508，SISO MAP 解码器 510，检测器 512，能量公制计算器 514，存储单元 520 和地址产生器 530。回述到图 4，2 个组成解码器 410a 和 410b 串行运行，来自一个解码器的非本征信息提供给另一个解码器作为输入。因为 2 个组成解码器的串行运行，仅仅一个（物理）解码器就可用于实现全部组成解码器。

输入接口单元 506 提供信道解交错器 502 和 MAP 解码器 510 间的接口。对于某些系统来说，如果击穿过程在发送器单元执行的话，输入接口单元 506 可设计为提供编码比特的解击穿过程。解击穿过程通过对每个击穿比特的去除来实现，去除预示着比特是零或一有同等可能。

依靠已执行的特殊解码通过（即对第一或第二组成编码器解码），信息和奇偶校验比特的正确序列从信道解交错器 502 提供给存储器/组成器 508。来自先前循环获得的 APP 同样从存储单元 520 提供给存储器/组成器 508。在一个实

施例中，存储器/组成器 508 被设计成 (1) 接收和为接收(soft)信息比特 x' 计算 LLR，(2) 把步骤 (1) 中计算好的 LLR(x')和相应的非本征信息结合在一起产生 APP，(3) 接收和为接收(soft)信息比特 y' 或 z' 计算 LLR。

在一个实施例中，存储器/组成器 508 用临时存储如信息和奇偶校验比特的 4 个窗口宽的滑动窗口架构 (SWA) 来实现。当信息和奇偶校验比特的 3 个窗口在解码器 510 中的 3 个状态公制计算器上运行时，第四个窗口由来自信道解交错器 502 和存储单元 520 的值所更新。在一个实施例中，每个窗口有 32 比特的大小，即每个窗口容纳 32 个 x' 码元和 32 个 y' (或 z') 码元。其他窗口长度和/或不同数量的窗口同样可以使用。

在一个实施例中，解码器 510 实现执行对数-MAP 算法的 SISO 解码器。任何基于 MAP 的解码算法 (如最大对数-MAP 算法或最大*对数-MAP 算法，两者都在业内众所周知) 同样可以使用。在一个实施例中，为实现 MAP 算法，解码器 510 包括一个前向状态公制计算器和二个反向状态公制计算器。每个前向 (反向) 状态公制计算器在一个给定的当前时间实例内，基于 (1) 在先前 (或未来) 时间实例内状态的概率，(2) 在先前 (或未来) 状态和现行时间状态之间的路径的概率，来计算格子中每个 2^{k-1} 状态的概率的对数，其中 K 是组成编码器的约束长度。这些前向和反向状态公制 (分别地， α 和 β) 然后用于计算信息比特的后验概率。前向和反向状态公制计算和 Turbo 解码过程由 Steven S. Pietrobon 所著的题为 “Turbo/MAP 解码器的实现和性能” 1998，卷 16，航天通讯国际杂志，中的页 23-46 中详述，其在此引用结合参考。

解码器 510 的输出存储在存储单元 520 中。在一个实施例中，存储单元 520 以交错的次序 (对应于同样可以使用的线性次序) 操作来存储来自解码器 510 的 APP 码元。以交错次序存储中间结果可以简化存储器的划分为多个存储体，从而易于存取，并进一步允许 Turbo 编码器和解码器使用相同的交错地址产生器。在一个实施例中，存储单元 520 进一步运行，以存储来自第一个组成解码过程的 APP 码元和来自第二个组成解码过程的非本征信息。

对于第一个组成解码过程来说，来自先前第二个组成解码过程的非本征信息被以解交错次序从存储单元 520 检索出来，解码过程产生的 APP 码元以交错的次序存储在存储单元 520。相应地，对于第二个组成解码过程来说，来自先

前第一个组成解码过程的 APP 码元以线性次序从存储单元 520 中检索出来，解码过程产生的非本征信息以线性的次序存储在存储单元 520。

在一个实施例中，存储单元 520 被实现划分成许多存储体。存储体以避免 APP 数据的双重缓冲的方式被分配和运行。每个存储体可以实现成为能从其他存储体分开地和独立地访问。这可通过提供每个存储体他们自身的地址集和数据线来达到。

地址产生器 530 为存储单元 520 提供写和读地址。象征性地显示在图 5 中的复用器 532a，指示 APP 码元/非本征信息可以以线性或交错的次序写入存储单元 520，象征性地显示的复用器 532b 指示 APP 码元/非本征信息可以以线性或解交错的次序从存储单元 520 检索出。

检测器 512 在最后解码循环后检索出 APP 码元，为接收信息提供硬判决。能量公制计算器 514 为信息比特（或它们的 LLR）提供能量公制。能量公制可以用作检测信息比特可信度的另一个指示。

如图 5 所示，控制器 540 可以控制 Turbo 解码器 500 的运行，能进一步提供 Turbo 解码器 500 所需的各种参数（如码元段的大小 K）。

对于图 5 中的 Turbo 解码器设计，存储单元 520 存储来自第一个组成解码器的 APP 数据和来自第二个组成解码器的非本征信息。APP 数据和非本征信息是来自组成解码器的中间结果的二个不同的形式。如在此所述，中间结果能包含任何从一个组成解码器通到后续解码器的信息，并可以为任意形式。典型地，来自任何组成解码器被存储的中间结果的特殊形式依赖于 Turbo 解码器的特殊设计。

码元交错过程是 Turbo 编码器和解码器的重要和应有的部分。无论在 Turbo 编码器为码元交错过程选择怎样的机制，同样的机制用于存储/检索来自第一个组成解码过程的 APP 码元，且互补的机制用于为第二个组成解码过程存储/检索非本征信息。

W-CDMA 为 Turbo 编码器定义了一个专用的交错机制。这个交错机制可以划分成 3 个阶段：（1）把在码元段（即数据分组）中的信息比特一行接着一行地写入一个 $R \times C$ 数组中，（2）在每行中重排元素（即行内交换），和（3）行交换过程（即行间交换）。其后在 $R \times C$ 数组中，从 $R \times C$ 数组的最左上方元素开始，一列接着一列的读出比特。这 3 个阶段将在下面进一步详细描述，其后提

供了一个例子以便更好地理解交错机制。

在第一个阶段，每个码元段中的比特被写入 $R \times C$ 数组。W-CDMA 标准支持各种大小从 40 到 5114 比特范围的码元段。最初，阵列中的行数 R 是由码元段的大小 K 所决定的，如下：

$R = 5$ ，假如 $40 \leq K \leq 159$ ；

$R = 10$ ，假如 $160 \leq K \leq 200$ 或者 $481 \leq K \leq 530$ ；或者

$R = 20$ ，对于所有其他的 K ；

接下来数组中的列数 C 是由 R 和 K 所决定的，如下：

$C = 53$ ，假如 $481 \leq K \leq 530$ ；否则

选择一个素数 p 使得 $(p+1) \cdot R \leq K$ ，然后

选择 $C = \min\{p-1, p, p+1\}$ 使得 $R \cdot C \leq K$

一旦 R 和 C 由给定的 K 所确定，码元段中的比特被一行行地写入 $R \times C$ 数组。由于 $K \leq R \cdot C$ ，在数组的底部可能有空单元（即一行或多行，或其中的一部分，可能没有包括任何比特）。

图 6A 是显示码元段比特写入 $R \times C$ 数组的次序图。从数组的左上角开始，阵列每次填充一行。如果码元段中的比特数量不等于数组的大小（即 $K \leq R \cdot C$ ），则在数组的底部行会有空单元，如阴影单元所示。

在第二个阶段，每行的元素被交换（即混洗），基于由那行所特别定义的行间交换序列 $c_j(i)$ 。行间交换可用一系列的步骤来执行。在第一个步骤，产生长度为 p 的一个基数序列 $c(i)$ 。对于每个在第一阶段决定的可能的素数 p ，如 W-CDMA 标准所定义并显示在表 1 中，以据此素数有一个基本根 g_0 。基数序列 $c(i)$ 的元素能如此得出：

$$c(i) = [g_0 \cdot c(i-1)] \text{ modulo } (p) \quad \text{for } i = 1, 2, \dots, (p-1) \text{ 等式 (1)}$$

其中 $c(0) = 1$

表 1

p	g ₀	P	g ₀	p	g ₀	p	g ₀	p	g ₀
7	3	47	5	101	2	157	5	223	3
11	2	53	2	103	5	163	2	227	2
13	2	59	2	107	2	167	5	229	6
17	3	61	2	109	6	173	2	233	3
19	2	67	2	113	3	179	2	239	7
23	5	71	2	127	3	181	2	241	7
29	2	73	7	131	2	191	19	251	6
31	3	79	3	137	3	193	5	257	3
37	2	83	2	139	2	197	2		
41	6	89	3	149	2	199	3		
43	3	97	5	151	6	211	2		

在第二阶段的第二步骤，R 个素数的一个序列， q_j 被创建。选择了这个素数序列的元素，使得下面的标准得以满足：

$$\text{g. c. d. } \{ q_j \cdot p-1 \} = 1;$$

$$q_j > 6 ; \text{ 和 等式 (2)}$$

$$q_j > q_{j-1}$$

其中 g. c. d. 是最大公因数，且 $g_0 = 1$

素数序列 q_j 是排除了是 $p-1$ 因子的素数的递增的最小素数的一个序列。这个素数序列 q_j 中的 R 个元素分别和数组中的 R 行关联。如下面将进一步详细描述，序列 q_j 中 R 个元素中的每一个在而后用于计算相关行的行间交换序列 $c_j(i)$ 。

由于素数序列 q_j 中下标从 0 到 R-1 的元素分别和 0 到 R-1 行相关联，且由于行间交换后，行随后和行间交换序列 P_x 相交换，素数序列 q_j 的元素仍旧用相同的行间交换序列 P_x 交换。如下面将进一步详细描述，为该码元段，序列 P_x 从 4 个可能的序列 P_A, P_B, P_C, P_D 中选出。已交换的素数序列 q_j 的元素由如下所决定：

$$P_j = P_{p_{x(j)}} = q_j, \quad \text{for } j = 0, 1, \dots, R-1 \quad \text{等式 (3)}$$

在第二阶段的第三个步骤，每行的行间交换序列 $c_j(i)$ 由如下所决定：

$$c_j(i) = c([i \cdot p_j] \text{ modulo } (p-1)) \quad \text{for } i = 0, 1, \dots, (p-2) \quad \text{等式 (1)}$$

其中 $c_j(p-1) = 0$ ， j 是行间交换后的行下标， $c(x)$ 是由上面等式 (1) 得出的基数序列 $c(x)$ 的元素 x^{th} ， $c_j(i)$ 是 j^{th} 行交换后 i^{th} 输出的输入比特位置。每行 j 的行内交换序列 $c_j(i)$ 因此依靠基数序列 $c(x)$ 和关联于该行的交换素数序列中的素数 p_j 得出。每行中的元素被交换，由此位于交换行的 i^{th} 单元存储了位于原先行中 $c_j(i)^{\text{th}}$ 单元中存储的元素。

如上面所注释的， C 能等于 $p-1$ ， p ， $p+1$ 。因此，行间交换序列 $c_j(i)$ 按如下所使用：

假如 $C = p-1$ ，使用 $c_j(i)-1$ 对 $i = 0, 1, 2, \dots (p-2)$ ；等式 (5)

其它假如 $C = p$ ，使用 $c_j(i)$ 对 $i = 0, 1, 2, \dots (p-2)$ ，和 $c_j(p-1)=0$ ；和

其它假如 $C = p+1$ ，使用 $c_j(i)$ 对 $i = 0, 1, 2, \dots (p-2)$ ，和 $c_j(p-1)=0$ ， $c_j(p)=p$ ；和

假如 $R \cdot C = K$ ，则把 $c_{R-1}(p)$ 和 $c_{R-1}(0)$ 交换。

在第三个阶段，基于行间变换序列 P_X ，数组中的 R 行被变换， P_X 是由 W-CDMA 标准定义的四个可能序列 P_A 、 P_B 、 P_C 和 P_D 中选择一个，如下：

$$P_A = \{19, 9, 1, 4, 0, 2, 5, 7, 12, 18, 10, 8, 13, 17, 3, 1, 16, 6, 15, 11\},$$

$$P_B = \{19, 9, 14, 1, 0, 2, 5, 7, 12, 18, 16, 13, 17, 15, 3, 1, 6, 11, 8, 10\},$$

$$P_C = \{9, 8, 7, 6, 5, 4, 3, 2, 1, 0\}, \quad \text{和}$$

$$P_D = \{4, 3, 2, 1, 0\}.$$

基于下面条件选择一个特殊行间变换序列以用于码元段：

假如 K 属于 $[201, 480]$ ， $[531, 2280]$ ， $[2481, 3160]$ ，或 $[3211, 5114]$ 比特，则

选择 P_A 。

假如 K 属于 $[2281, 2480]$ 或 $[3161, 3210]$ ，则选择 P_B 。

假如 K 属于 $[160, 200]$ 或 $[481, 530]$ (即 $K=10$)，则选择 P_C 。

假如 K 属于 $[40, 155]$ (即任何时候 $R=5$ 使用 P_D)，则选择 P_D 。

行间变换就这样被实现，原始数组中的 j^{th} 行被移动到变换后数组的 $P_x(j)$ 行位置。

行间变换后，比特被从 $R \times C$ 数组一列一列的地从顶部到底部 (即从 0 行到 $R-1$ 行) 读出。如上面所注释的，因为 $K > R \cdot C$ ，所以数组中的某些单元不能包含有效数据，当读出数据时这些单元被跳过。

为明确起见，下面为上述交错机制提供了一个例子。在此例子中， $K=40$ 并导致 $R=5$ 选择。然后素数 p 被判决为 $p=7$ ，列数 C 被判决为 $C=8$ 。对于此素数 p ，相关基本根 $g_0=3$ 。变换可被总结为如下：

$$K=40, R=5, C=8, p=7 \text{ 和 } g_0=3$$

使用等式 (1)，基数序列 $\underline{c}(i)$ 被判决为：

$$\underline{c}(i) = \{1, 3, 2, 6, 4, 5\}$$

素数序列 q_j 接下来从等式组 (2) 派生出

$$q_j = \{1, 7, 11, 13, 17\}。$$

对于 $K=40$ 来说，选择行间变换序列 P_D 。基于等式 $P_{PA(j)}=q_j$ ，从素数序列 q_j 中产生变换后素数序列 \underline{P}_j 。以提供下面的：

$$\underline{P}_j = \{17, 13, 11, 7, 1\}。$$

然后基于变换后素数 \underline{P}_j ，为行和基数序列 $\underline{c}(i)$ 判决每个行 j 的行间变换序列 $\underline{c}_j(i)$ 。对于第一行 ($j=0$)，行的变换后素数是 $P_0=17$ ，且行间变换序列 $\underline{c}_0(i)$ 被判决为：

$$\begin{aligned}\underline{c}_0(i) &= c([i \cdot P_0] \text{ modulo } (p-1)), \quad i=0, 1, \dots, 5, \\ &= c([i \cdot 17] \text{ modulo } 6) \\ &= \{1, 5, 4, 6, 2, 3\}.\end{aligned}$$

由于 $C=p+1$, $c_j(6)=0$ 和 $c_j(7)=7$ 。所以第五行的行间变换序列可以表示为:

$$\begin{aligned}\underline{c}_0(i) &= \{1, 5, 4, 6, 2, 3, 0, 7\}, \\ \underline{c}_1(i) &= \{1, 3, 2, 6, 4, 5, 0, 7\}, \\ \underline{c}_2(i) &= \{1, 5, 4, 6, 2, 3, 0, 7\}, \\ \underline{c}_3(i) &= \{1, 3, 2, 6, 4, 5, 0, 7\}, \text{ 和} \\ \underline{c}_4(i) &= \{1, 3, 2, 6, 4, 5, 0, 7\}.\end{aligned}$$

图 6B 是显示 $K=40$ 上面例子的交错过程图。在第一个阶段, 参数被判决为如下: $K=40$, $R=5$, $C=8$, $p=7$ 和 $g_0=3$ 。然后码元段内的比特被一行一行的写入 $R \times C(5 \times 8)$ 数组 710, 从 0 行的 0 列开始, 到 4 行的 7 列结束, 如图 6B 所示。

对于第二个阶段来说, 基数序列 $\underline{c}(i)$ 被首先派生出, 然后素数序列 \underline{q}_j 被判决和变换, 以派生出变换后的素数序列 \underline{p}_j 。然后 5 行中的每一个的行间变换序列 $\underline{c}_j(i)$ 按如上所示判决。接着基于与行相关的行间变换序列 $\underline{c}_j(i)$, 混洗 $R \times C$ 数组的每行中的元素。例如, 对于 0 行, 行间变换后的数组 712 中在行下标 $\{0, 1, 2, 3, 4, 5, 6, 7\}$ 的单元用初始数组 710 中来自行下标 $\{1, 5, 4, 6, 2, 3, 0, 7\}$ 的单元值来填充。同样对于 1 行来说, 数组 712 中在行下标 $\{0, 1, 2, 3, 4, 5, 6, 7\}$ 的单元用数组 710 中来自行下标 $\{1, 3, 2, 6, 4, 5, 0, 7\}$ 的单元值填充。同样, 因为 $K=R \cdot C$, 在行下标 $c_{R-1}(7)$ 的单元和在行下标 $c_{R-1}(0)$ 的单元交换, 在此 $c_{R-1}(0)$ 和 $c_{R-1}(7)$ 是行间变换后的元素。

对于第三个阶段来说, 基于行间变换序列 \underline{p}_D , 混洗在数组 712 中的行, 以派生出最后交错的数组 714。然后以下面的次序 $\{34, 26, 18, 10, 8, 36, 28, \dots, 16, 2\}$ 读出数组 714 中的元素。

如上所示, 由 W-CDMA 标准定义的交错过程是一个复杂的过程。在实际实现过程中, 为了达到交错过程, 码元段的比特以特殊次序 (例如线性或交错) 被存储到存储器单元中, 并以互补的次序 (即交错或线性) 从存储器单元被检

索。

本发明从各方面提供技术，以有效产生实现由 W-CDMA 标准定义的 Turbo 码交错过程所需的存储地址。在一个方面中，为了加速地址的产生，许多查表 (LUT_S) 被提供，以存储用于产生交错后地址值的各种序列。这些表的使用加速了地址计算，并允许所需地址在更少的时间内被产生。因此地址产生可以对于 Turbo 解码过程不是瓶颈。这些表单中的一部分和其中存储的序列将在下面描述。

在另一个方面，在此提供以基于这些表单有效产生交错地址的技术。交错地址产生技术可以用于 Turbo 解码过程，特别有利于计算密集的 Turbo 解码过程。快速的地址产生对有效的 Turbo 解码过程是至关重要的，特别是在支持高数据速率，并按照 Turbo 解码过程的循环本质的时候。

PRIME 表单存储所有可以使用的素数 p 。对于所有 W-CDMA 标准支持的可能码元段大小，总共有 52 个素数 p 。因此 PRIME 表单包括 52 个条目，对于此 52 个素数，下标从 0 到 51，如表 2 所示。因此最大的素数是 257，每个表单条目可以用 9 个比特实现。对于 PRIME 表单，通过合适的下标 p_{ind} 可以检索出特殊的素数。

表 2

index	p	index	p	index	p	index	p	index
0	7	11	47	22	101	33	157	44
1	11	12	53	23	103	34	163	45
2	13	13	59	24	107	35	167	46
3	17	14	61	25	109	36	173	47
4	19	15	67	26	113	37	179	48
5	23	16	71	27	127	38	181	49
6	29	17	73	28	131	39	191	50
7	31	18	79	29	137	40	193	51
8	37	19	83	30	139	41	197	
9	41	20	89	31	149	42	199	
10	43	21	97	32	151	43	211	

INTRROW 表单存储四个行间变换序列 P_A 、 P_B 、 P_C 和 P_D 。 P_A 、 P_B 、 P_C 和 P_D 分别包括 20, 20, 10 和 5 个条目, 并分别在 INTRROW 表单中地址 0, 20, 40 和 50 开始存储。

INTRABASE 表单存储在 PRIME 表单中列出的 52 个素数 p 的 52 个行内基数序列 $\underline{c}(i)$ 。如等式 (1) 中所示, 每个素数 p 与各自具有 $p-1$ 长度的行内基数序列 $\underline{c}(i)$ 相关。 $\underline{c}(i)$ 元素取值范围是 1 到 256, $\underline{c}(i)-1$ 而不是 $\underline{c}(i)$ 可以使用 8 个比特存储在 INTRABASE 表单中。当 $C=p-1$ 时, 对于行内基数序列 $\underline{c}(i)$ 的 $\underline{c}(i)-1$ 的存储值可以被直接使用。当 $C=p$ 或 $C=p+1$ 时, $\underline{c}(i)-1$ 的存储值加 1 以获得 $\underline{c}(i)$, 然后使用 $\underline{c}(i)$ 。

INTRADDR 表单存储在 INTRABASE 表单中存储的 52 个行内基数序列 $\underline{c}(i)$ 的开始地址。因为每个基数序列 $\underline{c}(i)$ 有可变的长度 (即 $p-1$), 所以 INTRADDR 表单用于快速指明 INTRABASE 表单中每个基数序列 $\underline{c}(i)$ 的开始地址。这个开始地址作为一个偏移使用以检索所需基数序列的单个元素。

Y 表单为 PRIME 表单中列出的 52 个素数 p 存储 52 个素数序列 q_j 。如等式组 (2) 中所示, 每个素数 p 与各自具有长度 R 的素数序列 q_j 相关, 在此 R 是 5, 10 或 20, 最多是 20。在一个替换实施例中, 变换后的素数序列 \underline{p}_j 可以替换的被存储在 Y 表单中。每个素数序列 q_j 的 R 元素被相关行间变换序列 P_x 变换, 其根据码元段的大小 K 可以是 P_A 、 P_B 、 P_C 或 P_D 。因此, 每个素数序列 q_j 可以基于与行间变换序列 P_x 相关的序列被变换, 并作为变换后素数序列 \underline{p}_j 存储。

变换后素数序列 \underline{p}_j 中的 R 个元素中的每一个被用于产生各自行内变换序列 $\underline{c}_j(i)$, 如等式 (4) 中所示。派生出每行的行内变换序列 $\underline{c}_j(i)$ 的元素:

$$\begin{aligned}\underline{c}_j(i) &= c([i \cdot p_j] \text{ modulo } (p-1)) \\ &= c(x),\end{aligned}$$

在此 $x = [i \cdot p_j] \text{ modulo } (p-1)$ 。基数序列 $\underline{c}(x)$ 的参数 x 也可以表示为:

$$x = [i \cdot \{ p_j \text{ modulo } (p-1) \}] \text{ modulo } (p-1)。$$

为了减少所需存储的数量, 简化随后基于变换后的素数序列 p_j 的计算, 素数序列 q_j 的元素可以在模 $(p-1)$ 除法后存储, 其可表示为:

$$q_{jp} = q_j \text{ modulo } (p-1)$$

因此 $q_j \text{ modulo } (p-1)$ 被存储在 Y 表单，而不是 q_j 。可以显示的是所有 52 个序列的最大数字是 39，因此每个元素可以使用 7 个比特存储。

在一个实施例中，为简化 Y 表单的存取过程，52 个序列 q_j 的每个使用 20 个条目存储在 Y 表单中，每个序列在各自开始地址开始存储。每个序列中的条目可以使用公式存取： $p_{ind} * 20 + \text{偏移}$ ，在此 p_{ind} 是与序列相关的素数 p 的下标，偏移是在序列中所需元素的偏移。

图 7 显示了可以用于为由 W-CDMA 标准定义的交错机制加速地址产生的表单。PRIME 表单包括 52 个素数 p 的 52 个条目。INTRROW 表单包括 4 个行间变换序列 P_A 、 P_B 、 P_C 和 P_D 的 55 个条目。INTRABASE 表单近似包括在 52 个不同开始地址的 52 个素数 p 的 52 个行内基数序列 $c(i)$ 的 6K 个条目。INTRADDR 表单包括存储在 INTRABASE 表单中 52 个行内基数序列 $c(i)$ 的开始地址的 52 个条目。Y 表单包括对应于 52 个素数 p 的 52 个素数序列 q_j 的 1040 个条目。

使用上面定义的表单，可以派生出为已给出码元段大小 K 和已知 R 和 C 产生交错地址的程序。这个程序可以使用下面显示的伪码来表示。接下来同样提供伪码的描述。

```

10  p = PRIME(pind);
20  intra_st = INTRADDR(pind);
30  r_in = 0;
40  c_in = 0;
50  for (i=0; i<k; i++) {
60  LOOP;
70r_new = INTRROW(introw_st+r_in);
80pj = Y[pind*20+r_in];
90if (c_in==0) c_idx(r_in) = 0;
100  elseif (c_in<p-1) {
110c_idx[r_in] = c_idx[r_in]+pj;
120if (c_idx(r_in) >?? p-1)

```

```

130  c_idx[r_in] = c_idx[r_in]-p+1;
140  }
150  if (C==p-1) c_new=INTRABASE[intra_st+c_idx[r_in]];
160  else {
170  elseif (c_in<p-1) {
180  c_new = INTRABASE[intra_st+c_idx[r_in]]+1;
190  if (K==R*C && C==p+1 && r_new==R-1 && c_in==0) {
200  last = c_new;
210  c_new = P;
220  }
230}
240elseif (c_in==p-1) c_new=0;
250elseif (c_in==p) {
260 if (K==R*C && C==p+1 && r_new==R-1) c_new=last;
270 else c_new=P;
280}
290  }
300  addr_new = r_new*C+c_new;
310  r_in++;
320  if (r_in=R) {c_in++; r_in=0; }
330  if (addr_new>K-1) goto LOOP;
340  }

```

上面的伪码产生接下来的交错地址。码元段中的 K 个元素被假设以上述和图 6A（即一行接着一行的）次序写入 $R \times C$ 数组。接下来的码元段参数同样假设为已知，先于执行伪码： K 、 R 、 C 、 p_{ind} 和 $introw_st$ ，在此 p_{ind} 是 PRIME 表单的下标， $introw_st$ 是 INTROW 表单中所需行间变换序列 P_x 的开始地址。这些参数可由具有处理过的码元段的大小信息的控制器（例如图 5 中的控制器 540）提供。

最初，来自使用下标 p_{ind} （行 10）的 PRIME 表单的素数 p 被判决，且来自 INTRADDR 表单（行 20）的相关基数序列 $c(i)$ 的开始地址被判决。当前行和

列数使用的可变量 r_in 和 c_in 分别被初始化为 0 (行 30 和 40)。

然后执行 FOR 循环, 来为码元段 (行 50 和 60) 中 K 个元素的每一个产生交错地址。对于在码元段中的每个元素, 交错过程后, $R_x C$ 数组中元素的位置的行和列数 r_new 和 c_new 由程序产生。首先, 基于从 INTROW 表单 (行 70) 检索到的行间变换序列 P_x , 确定对应于当前行数 r_in 的变换后行数 r_new 。

接下来判决变换后的列数 c_new 。这要通过首先检索来自 Y 表单 (行 80) 的对应于当前行 r_in 的素数序列 q_j 元素来达到。然后表达式

$$c_j(i) = c([i \cdot q_j] \text{ modulo } (p-1)) = c(x)$$

由第一个求值的参数 x 来判决。这可以显示参数 x 可以循环的使用一些步骤来求值。首先,

$$\text{假如 } x_{i-1} = [(i-1) \cdot q_j] \text{ modulo } (p-1), i > 1, 2, \dots, p-2,$$

$$\text{则 } x_i = [x_{i-1} + q_j] \text{ modulo } (p-1), \text{ 在此 } x_0 = 0$$

因此, 对 $[(i-1) \cdot q_j]$ 的乘法操作可以用对 $[x_{i-1} + q_j]$ 的加法操作代替。第二, 因为被 $(p-1)$ 模以后, 从而存储 $q_j \pmod{(p-1)}$, 因此 $x_i < 2 \cdot (p-1)$, 模 $(p-1)$ 操作以产生 x_i 可以通过简单从 x_i 减去 $(p-1)$ 的方法达到, 假如 x_i 大于 $(p-1)$ 。假如参数 x 将被递归求解, 临时数组 $c_idx[]$ 被用于存储当前 x (即 x_{i-1}) 的值, 其实用于计算下一个 x (即 x_{i+1}) 值的后面一个。

当前行 r_in 的临时数组 $c_idx[]$ 中的元素的第一列 (行 90) 被设为 0。对于每个后续的列来说, 与当前行 r_in 相关的值 q_j 是从 Y 表单中检索出来的, 并加到先前的参数值 (即 x_{i-1}), 其在 $c_idx[r_in]$ (行 100 和 110) 临时数组中存储。假如结果只等于或大于 $(p-1)$, 被 $(p-1)$ 模可以通过被 $(p-1)$ 减去结果值, 以为当前列 (行 120 和 130) 产生参数 x 来实现。

然后通过查询基数序列 $c(i)$ 中的 x^{th} 元素来获得 $c_j(i)$ 的值, 其存储在 INTRABASE 表单中开始于偏移量 $intra_st$ 。假如列数 C 等于 $(p-1)$, 则直接使用从 INTRABASE 表单获得的 $c_j(i)-1$ 值, 如通过等式组 (5) (行 150) 中的第一个等式所显示的。否则, 从 INTRABASE 表单获得的 $c_j(i)-1$ 值被增加 1, 以获得 $c_j(i)$, 然后其被用于输入列数 0 到 $(P-2)$ (行 160 到 180)。

假如 $K=R \cdot C$, $C=p+1$, 则 $c_{R-1}(p)$ 与 $c_{R-1}(0)$ 交换, 如通过等式组 (5) 中最后一个等式所示。这可以通过当 $r_in=R-1$ 且 $c_in=0$ (行 190 和 200) 时, 保留

$c_{R-1}(0)$ 到变量 “last”，设置 c_{new} 到 $c_{R-1}(p)$ ，当 $C=p+1$ 时，其等于 p 来实现，如由等式组(5)(行 210)中第三个等式所示。(在行 180 c_{new} 初始设为 $c_{R-1}(0)$ ，且在行 200 “last” 设为 c_{new} ，从而存储 $c_{R-1}(0)$ 到 “last”。)随后当 $r_{new}=R-1$ ， $c_{in}=p$ 时， c_{new} 被设为 $c_{R-1}(0)$ ，其先前被存储在变量 “last” 中(行 250 和 260)。

假如 $c_{in}=p-1$ ，则 c_{new} 被设为 0，因为 $c_j(p-1)=0$ ，如在等式组(5)中第二个和第三个等式所示(行 240)。当 $c_{in}=p$ 时(除了当 $K=R \cdot C$ ， $C=p+1$ 和 $r_{in}=R-1$ 时，在这种情况下 $c_{R-1}(p)$ 与 $c_{R-1}(0)$ 交换)， c_{new} 被设为 p ，因为 $c_j(p-1)=p$ ，如等式组(5)中等式三和等式四所示(行 250 和 270)。

变量 r_{new} 和 c_{new} 表示交错过程后 R_xC 数组中的行和列数。因为码元段中的元素是存储在一维存储单元(例如在 0 到 $K-1$ 地址段)中的，变换后的行和列数 r_{new} 和 c_{new} 被变换成一个交错后地址 $addr_{new}$ (行 300)。因此，在 i 的存储器地址的元素映射到交错后 $addr_{new}$ 的存储器地址。

然后输入行和列数 r_{in} 和 c_{in} 增加(行 310 和 320)。因为在交错后数组中的元素被一一列的读出，行数首先增加，仅仅假如行计数器达到 R_xC 数组的末端时列数增加(行 320)，在此情况行数重新设为 0。

产生的交错地址 $addr_{new}$ 可以超过码元段大小 $K-1$ 。如图 6A 所示，码元段大小 K 可以小于 $R \cdot C$ ，在这种情况下 R_xC 数组中一个或多个单元不被使用。因此，当 $addr_{new} \geq K$ 时，这种情况表示没有已产生地址的有效数据，并且计算一个新的交错地址。在伪码(行 330)中的最后 “If” 语句处理这种情况，并通过回到 LOOP(在行 60)的开始来计算另一个交错地址。

上述交错地址产生技术可以在软件、硬件或他们的结合体中实现。对于某些应用(例如高速率 Turbo 解码过程)，交错地址需要以高速率产生。对于这些应用，在带查表的结合体中的硬件电路可用于快速产生所需的交错地址。

在一个实施例中，为了确保为每个时钟周期产生一个有效交错地址，提供二个地址发生单元以并发产生 2 个新的地址。如上面所注释的，计算出的交错地址 $addr_{new}$ 可能大于码元段大小 K ，因此不再有效。假如第一个交错地址在有效范围内($addr_{new} < K$)，则此地址被使用，且第二个交错地址被丢弃。然而，假如第一个交错地址在有效范围之外(即 $addr_{new} \geq K$)，则此地址被丢弃，且第二个交错地址被使用。二个地址产生单元是足够的，因为 Turbo 交错器的一个属性：没有二个连续交错过程地址将是无效的。

图 8 是依据本发明的一个实施例，能够提供交错地址的地址产生器 800 的方框图，地址产生器 800 近似地实现上述的伪码，包括二个能够同时产生二个交错地址的地址产生单元 810a 和 810b。假设对于每个时钟周期，一个正确的交错地址被 Turbo 解码器所需，那么二个地址发生单元 810 可以用于确保为每个时钟周期产生至少一个正确的交错地址。

如图 8 所示，一个（线性）输入地址通过一个加法器 812 与来自计数器 816 的坏地址计数值、bad_addr_cnt 相加，以派生第一个地址。最初，在码元段的开始，计数器 816 被设为 0。然后第一个地址被提供给地址发生单元 810a。随后，输入地址有规律地增加（例如每个时钟周期加 1）。每次一个“坏地址”被注册，计数器 816 在此线性地址（输入地址）被由一个计数“提高”。计数器 816 计数为输入地址所需的“提高”数字，并作为来自此输入地址的适当偏移。例如，假设当遇到坏地址时输入地址等于 2，则 bad_addr_cnt 增加到 1，提供给地址产生器的第一个地址是 3 而不是 2。假设当输入地址等于 5 时，遇到另一个坏地址，则坏地址增加到 2，提供给地址产生器的第一个地址是 7 而不是 5。

在地址发生单元 810a 中，第一个地址被提供给除法器单元 822，此单元用 R 除第一个地址，并在 $R \times C$ 数组中对应于第一个地址，提供一个表示输入行 r_{in} 的商，和一个表示输入列 c_{in} 的提醒信号。然后输入行数 r_{in} ，通过一个使用 INTROW 表单执行行间变换的行变址单元，映射到一个变换行数 r_{new} 。通过一个乘法器 826，此变换行数和列数 C 相乘，且产生结果 r_{os} 表示在变换行 r_{new} 的存储单元中的开始地址。

输入行数 r_{in} 同样提供给依据此行数提供素数 q_j 的单元 832，用 p_{ind} 表示。单元 832 包括或涉及 Y 表单以提供 q_j 。素数 q_j 和输入列数 c_{in} 被提供给模数加法器 834，以产生一个列系数 c_{idx} ，针对行内变换（即 $c_{idx}=[i \cdot q_j] \text{ modulo } (p-1)$ ，如等式（4）中所示）。此列系数 c_{idx} 被提供给列映射单元 836，其依据此输入地址提供一个变换列数 c_{new} 。单元 836 包括或引用 INRABASE 表单，以派生此变换列数 c_{new} 。然后加法器 828 接收并将此变换列数 c_{new} 和行开始地址 r_{os} 相加，以产生第一个交错地址 int_addr1 。

通过加法器 814，第一个地址同样即可以增加也可以减少（依据相应的输入地址是否被增加或减少），来为地址产生单元 810b 提供第二个地址。此第二个地址与第一个地址相邻，并在假如第一个地址导致合法的交错地址时使

用。单元 810b 以和单元 810a 一样的方式运行，并依据此第二地址提供第二个交错地址 int_addr2 。

检波器 838 接收第一个交错地址 int_addr1 ，并判决是否此地址是正确的（即是否 $int_addr1 < K$ ）。假如由地址产生单元 810a 产生的第一个交错地址是正确的，则此地址被作为交错地址 int_addr 从乘法器 840 提供，且由单元 810b 产生的第二个交错地址被丢弃。但是，假如第一个交错地址不正确（即超过范围，或 $int_addr1 > K$ ），则此地址被丢弃，且第二个交错地址被作为交错地址从乘法器 840，计数器 816 增加。从而乘法器 840 接收第一个和第二个交错地址 int_addr1 和 int_addr2 ，并基于来自检波器 838 的控制信号，提供这些地址中的一个作为输出交错地址 int_addr 。

计数器 816 计数为码元段产生的坏地址数量。来自计数器 816 的坏地址计数 bad_addr_cnt 被通过加法器 812 与输入地址相加，以产生一个校正的输入地址（即第一个地址），适于交错地址计算。

图 9 是能够依据本发明一个实施例为顺序输入地址序列，产生输入行和列数 r_in 和 c_in 序列的单元 822a 图。单元 822a 是图 8 中除法器单元 822 的一个实现。假设此码元段的元素已经被以线性次序存储，并将被以交错次序检索。

在单元 822a 中，计数器 910 被用来倒计数码元段中元素的数量。通过控制信号“开始”，计数器 910 被初始加载为值 $K-1$ ，其后倒计数降到 0。当计数器 910 到 0 时，寄存器 912 重新设置到低，且没有任何行和列数被产生，直到开始信号设置寄存器。寄存器 912 提供一个启动码元段行和列数的 K 系列产生的控制信号“结束”。

单元 822a 使用二个计数器来产生行和列数 r_in 和 c_in ，而不是执行除法操作，因为输入地址是连续的。计数器 922 产生行数 r_in ，因为 $R \times C$ 数组是以交错地址次序一列接着一列存取的，所以计数器首先是增加的。计数器 924 产生列数 c_in ，且如由比较器 926 判决的，行数达到 $R-1$ 后第二个增加。因为坏地址如上所述是偶尔产生的，当坏地址被检测到时，计数器 922 跳过一个计数（即计数 2）。此机制自动跳过坏地址。在此注释的是当行计数器达到 $R-1$ 后跳跃发生时，一个（‘1’）值载入行计数器 922。当“ bad_addr ”是真时，与门 932、寄存器 934 和或门 936 允许计数器 922 增加 2。

假如输入地址是递减而不是递增的，则行和列计数器计数下降而不是上升（图 9 中没有显示）。递减交错地址可以为某些 Turbo 解码器设计要求所需。

图 10 是单元 1000 的一个实施例图，其映射输入行和列数 r_{in} 和 c_{in} 为变换后的行和列数 r_{new} 和 c_{new} 。单元 1000 可以用于实现除了图 8 中地址发生单元 810 的除法器单元 822 以外的任何元素。

符合码元段大小 K 的素数下标 p_{ind} 可以被提供给 PRIME 表单和 INTADDR 表单，其分别为 INTRABASE 表单中所需行内基数序列 $c(i)$ 提供素数 p 和开始地址 $intra_{st}$ 。素数 p 可以由一个加法器一个接一个的递增，以提供值 $p+1$ ，并同样由另一个加法器一个接一个的递减，以提供值 $p-1$ 。这些值可被此单元中各种比较器使用。

被使用的特殊行间变换序列 P_x 的标示（在此是 P_A 、 P_B 、 P_C 或 P_D ）可以被提供给 INTROW START 表单，其为 INTROW 表单中所需序列 P_x 提供开始地址 $interow_{st}$ 。输入行数 r_{in} 可以通过加法器和开始地址 $interow_{st}$ 相加，来为 INTROW 表单产生地址，其然后符合输入行数 r_{in} 和 $inter_{row}$ 变换序列 P_x 提供变换的行数 r_{new} 。

素数下标 p_{ind} 同样被乘法器乘以 20，结果值由加法器和输入行数 r_{in} 相加，从而为 Y 表单产生一个地址。Y 表单可以提供与输入行和处理过的素数对应的素数 q_j 。

然后加法器可以将此素数 q_j 和先前的列下标 c_{idx} 相加，存储到锁存器以产生中间列下标。假如中间列下标小于 $(p-1)$ ，如比较器判决的，则从乘法器提供此下标。否则，中间列下标被加法器减去 $(p-1)$ ，结果从乘法器提供。锁存器、比较器、加法器和乘法器使用如上所述的递归技术有效地实现函数 $x_i = [i \cdot q_j] \text{ modulo } (p-1)$ 。例如，假设如第二个比较器判决的， $c_{in}=0$ ，则列下标 c_{idx} 通过第二个乘法器设为 0。否则，列下标被从第一个乘法器设为输出。从而第二个乘法器提供列下标 c_{idx} 。

此外，为所需的行内基数序列 $c(i)$ ，加法器可以用于将列下标 c_{idx} 和开始地址 $intra_{st}$ 相加，以产生适于 INTRABASE 表单的地址。然后 INTRABASE 表单可以提供对应于接收到的列下标（在此 $x=c_{idx}$ ）的值 $c_j(x)-1$ 。另一个加法器可以用于将值 $c_j(x)-1$ 和 1 相加以提供值 $c_j(x)$ 。第三个乘法器可以用于接收 $c_j(x)$ 和 $c_j(x)-1$ ，并在假设如由另一个比较器判决 $C=p-1$ 的话，提供 (1) 值 $c_j(x)-1$ 作为 c_{temp} ，否则 (2) 值 $c_j(x)$ 作为 c_{temp} 。

应理解对于各种条件，更多的逻辑可以被用于产生变换的列数。

在此描述的地址产生技术，可用于在 W-CDMA 系统中，和为上行链路 Turbo

码一样，为下行链路产生交错地址。

为清楚起见，已经描述了适于由 W-CDMA 标准定义的特殊 Turbo 码交错机制的地址产生器。每个 CDMA 标准可以定义一个不同于其他 CDMA 标准，包括 W-CDMA 标准定义的码元交错机制。例如，cdma2000 标准定义一个交错机制，由此行依据比特翻转次序被变换，例如行 1 (00001) 行 16 (10000) 和交换，行 3 (00011) 和行 24 (11000) 交换等等。对于这些不同的码元交错机制，在表单中存储的序列可能不同于由 W-CDMA 标准定义的上述交错机制的序列。

在此描述的地址产生技术可以用软件、硬件或他们的结合体实现。对于用硬件的实现，地址产生器可以用一个或多个数字信号处理器 (DSP)、专用集成电路 (ASIC)、处理器、微处理器、控制器、微控制器、现场可编程门数组 (FPGA)、可编程逻辑器件、其他电子单元、或它们的任意结合体来实现。地址产生器可以作为一个独立单元来实现、集成在控制器或存储单元中、在也包括其他处理元素的 ASIC 中，或通过某些其他的设计实现。对于软件实现来说，交错地址可以由在处理器 (如图 5 中的控制器 540) 上执行的程序代码来产生。上面描述了可以用于产生交错地址的一个伪代码例子，在本发明范围内许多其它的实现也是可能的。

表单和存储单元也可以用各种存储器技术如随机存储器 (RAM)、动态 RAM (DRAM)、闪存和其他技术来实现。表单和存储单元的各种结构和实现在本发明范围内是可能的。

所提供的首选实施例在前面的描述，使业内技术熟练人士实现和使用本发明。对这些实施例的各种修改将很容易的显现给业内的那些技术熟练人士，且无需使用创新能力，在此定义的一般法则可应用于其他实施例。因此，本发明并不意味着仅限于在此显示的实施例，而是与符合在此揭示的法则和新颖特点的最广阔范围相一致。

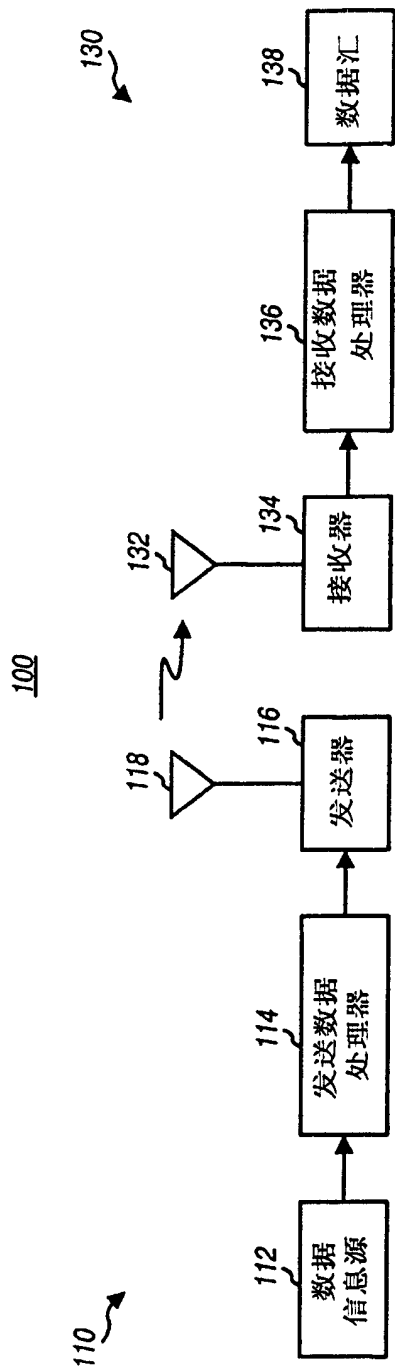


图 1

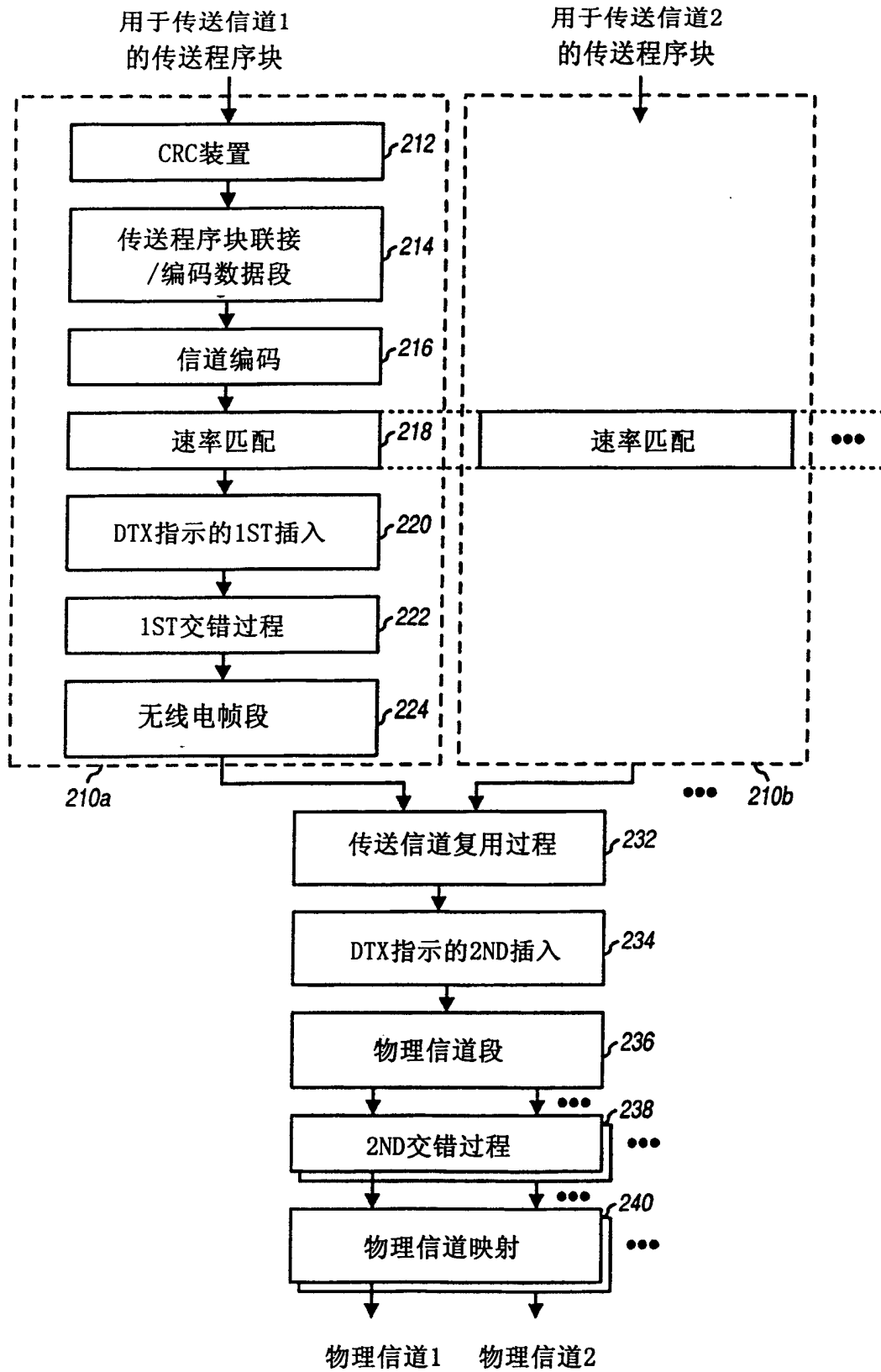


图 2A

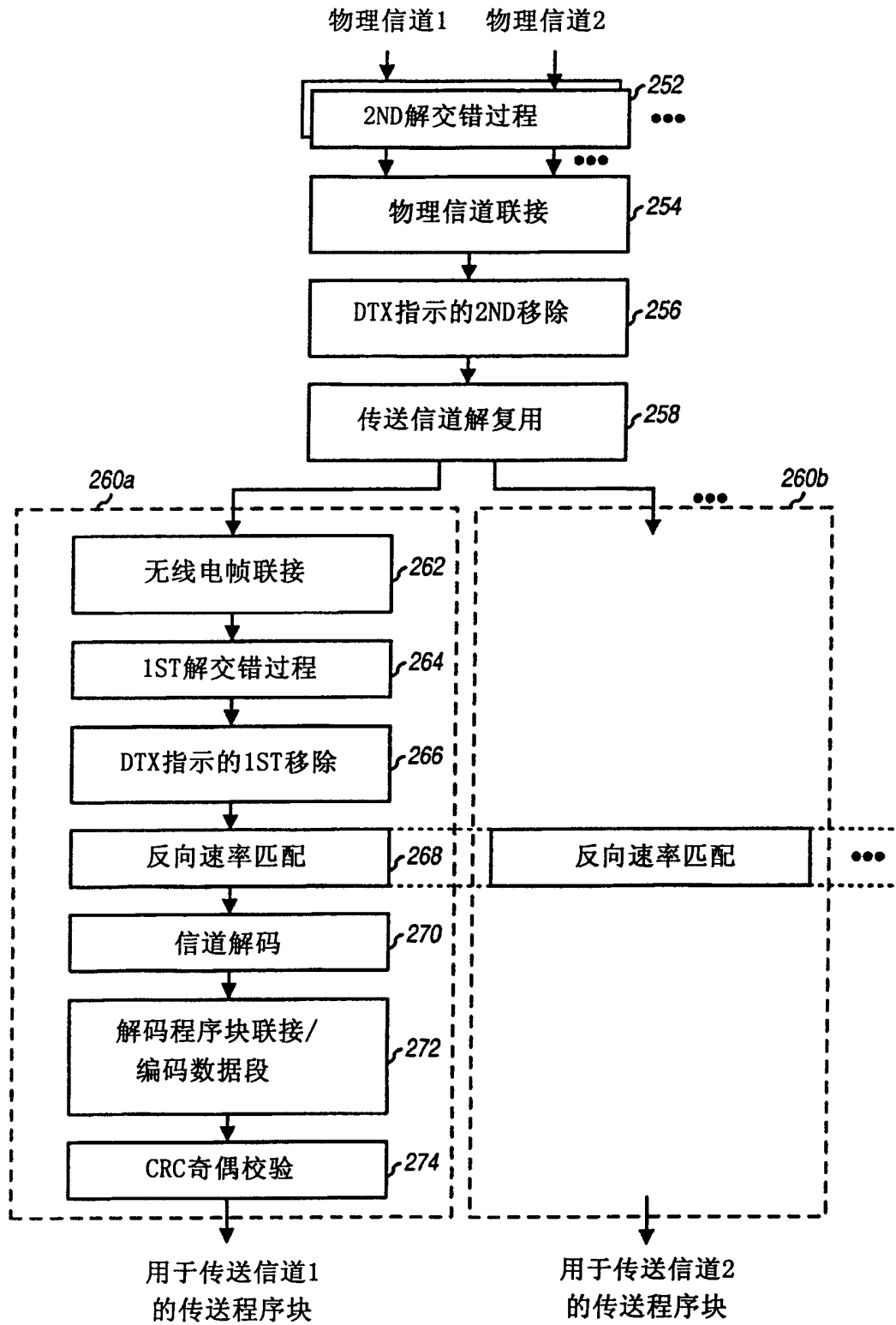


图 2B

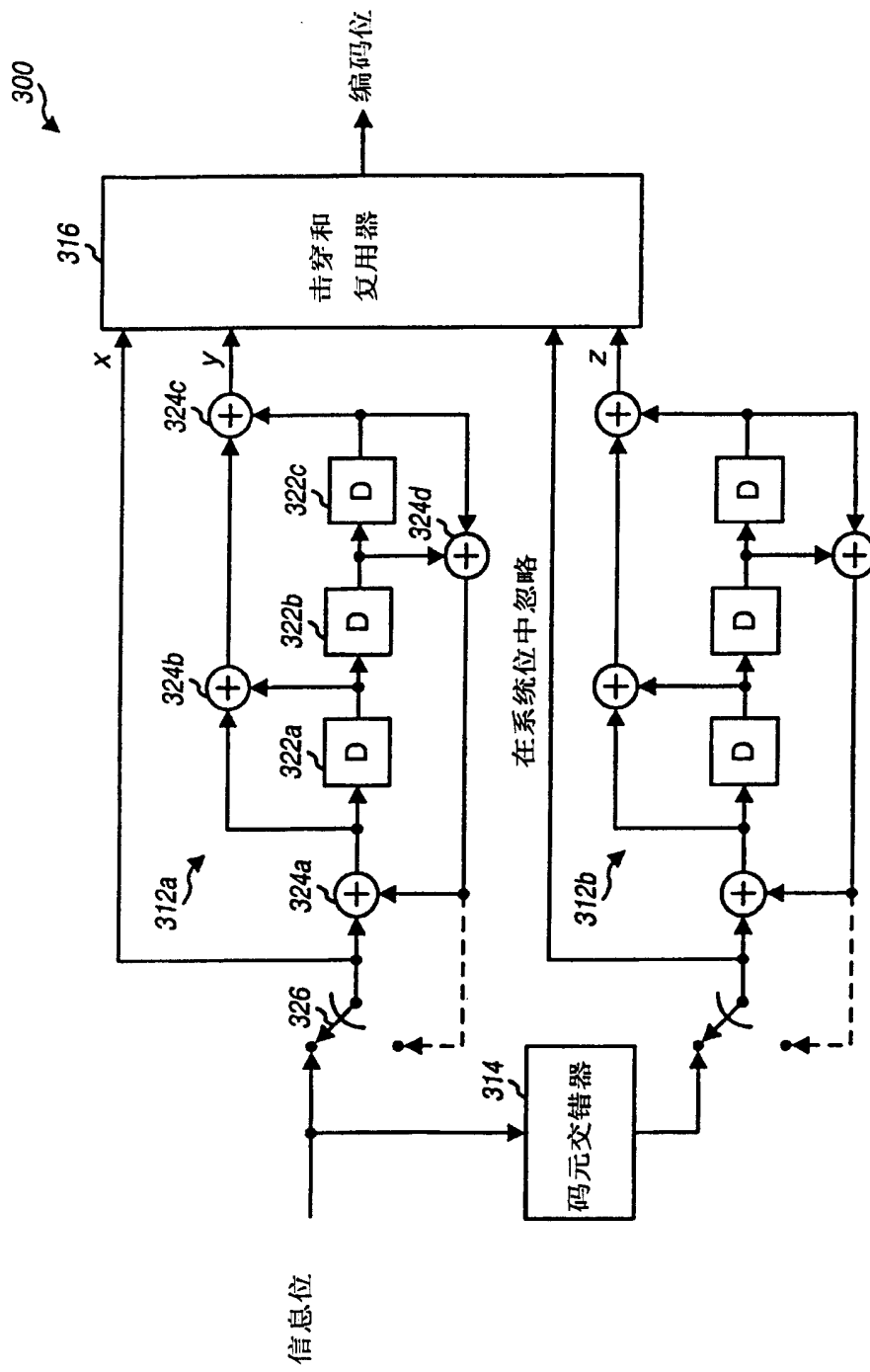


图 3

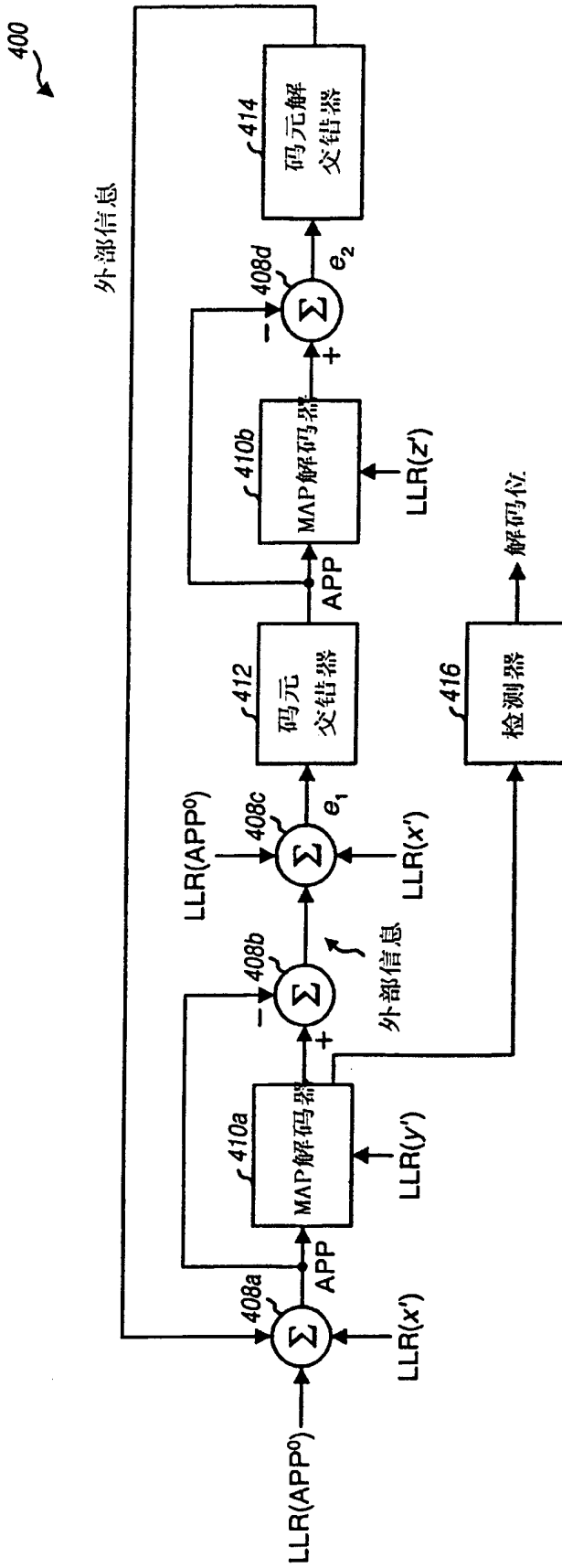


图 4

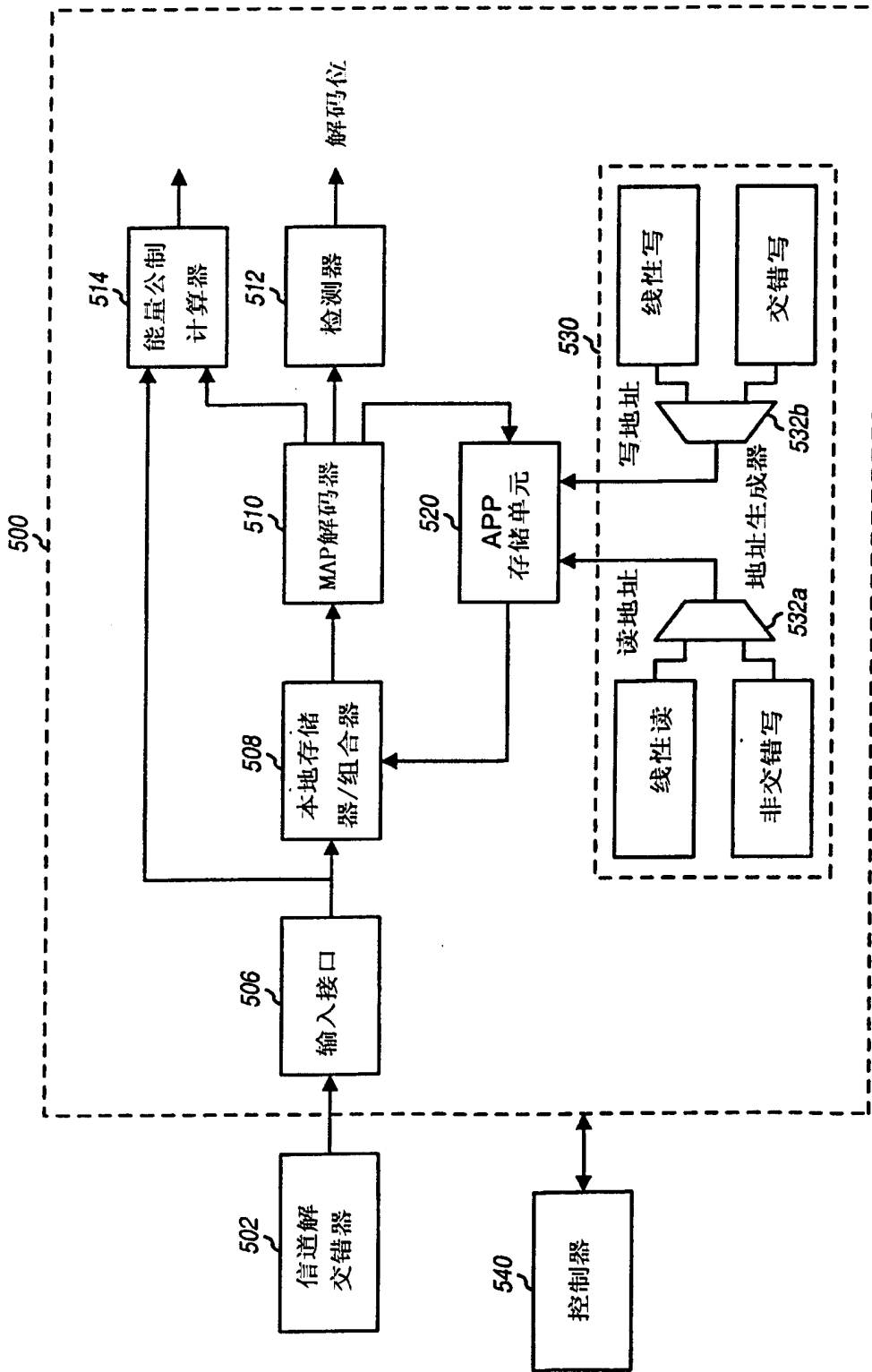


图 5

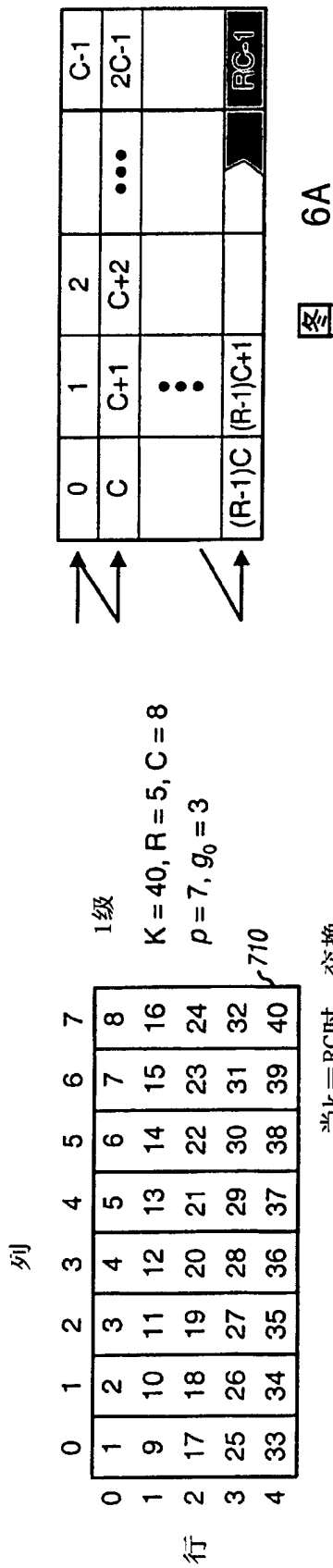


图 6A

2级一行内交换

$g_i = \{1, 3, 2, 6, 4, 5\}$
 $g_j = \{1, 7, 11, 13, 17\}$
 $P_D = \{4, 3, 2, 1, 0\}$
 $P_j = \{17, 13, 11, 7, 1\}$

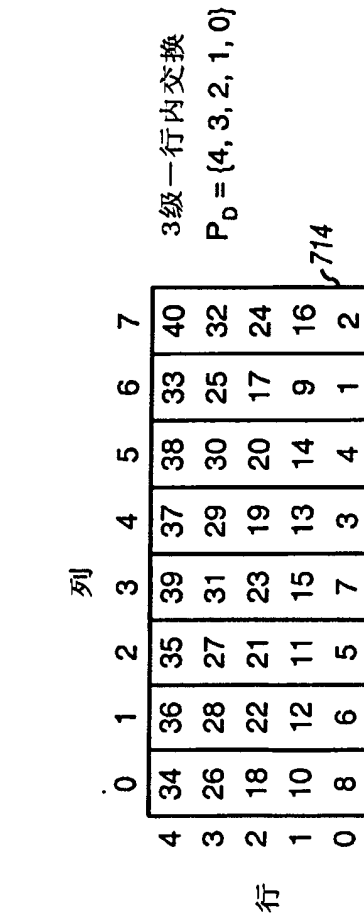
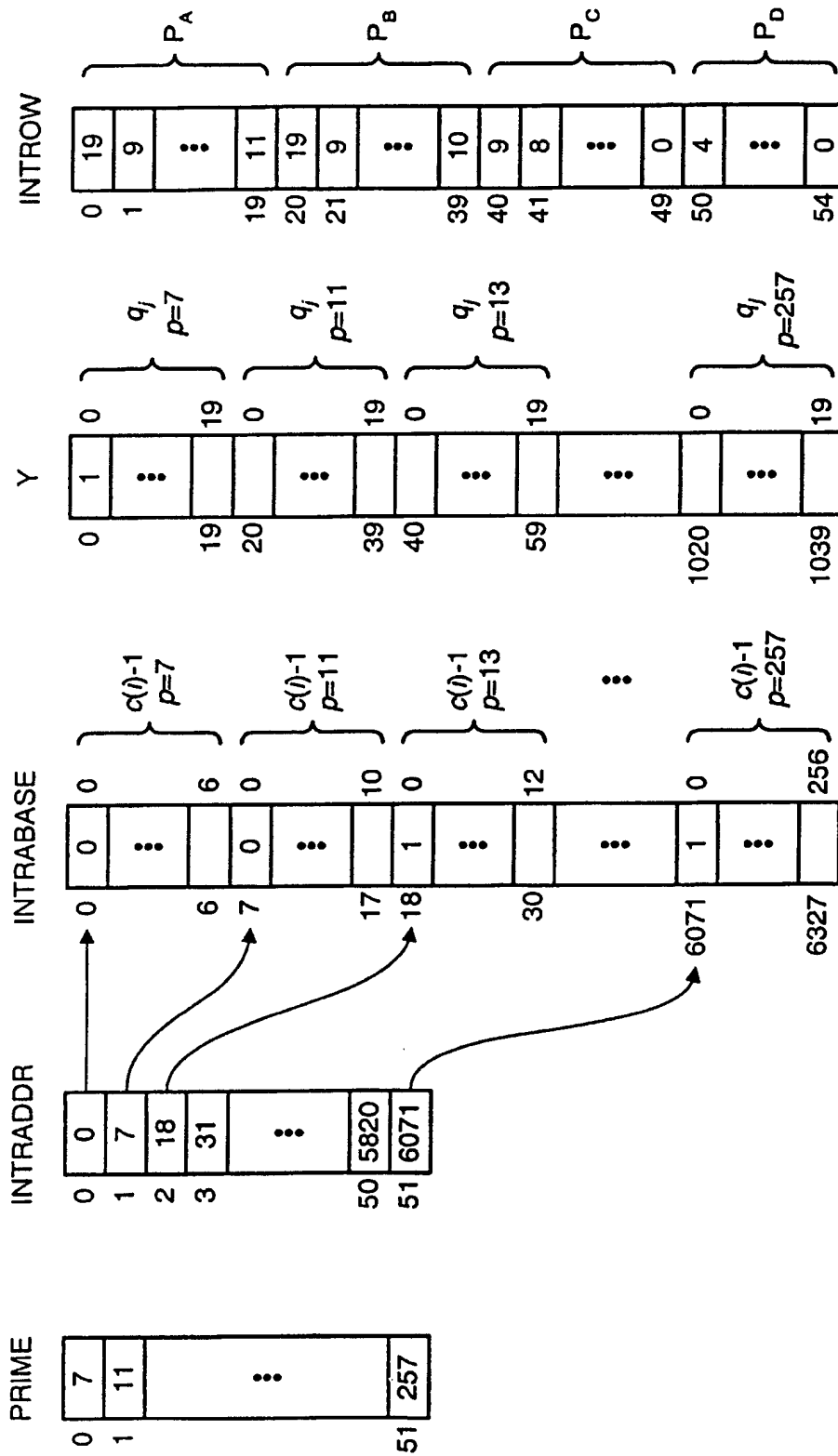


图 6B



7

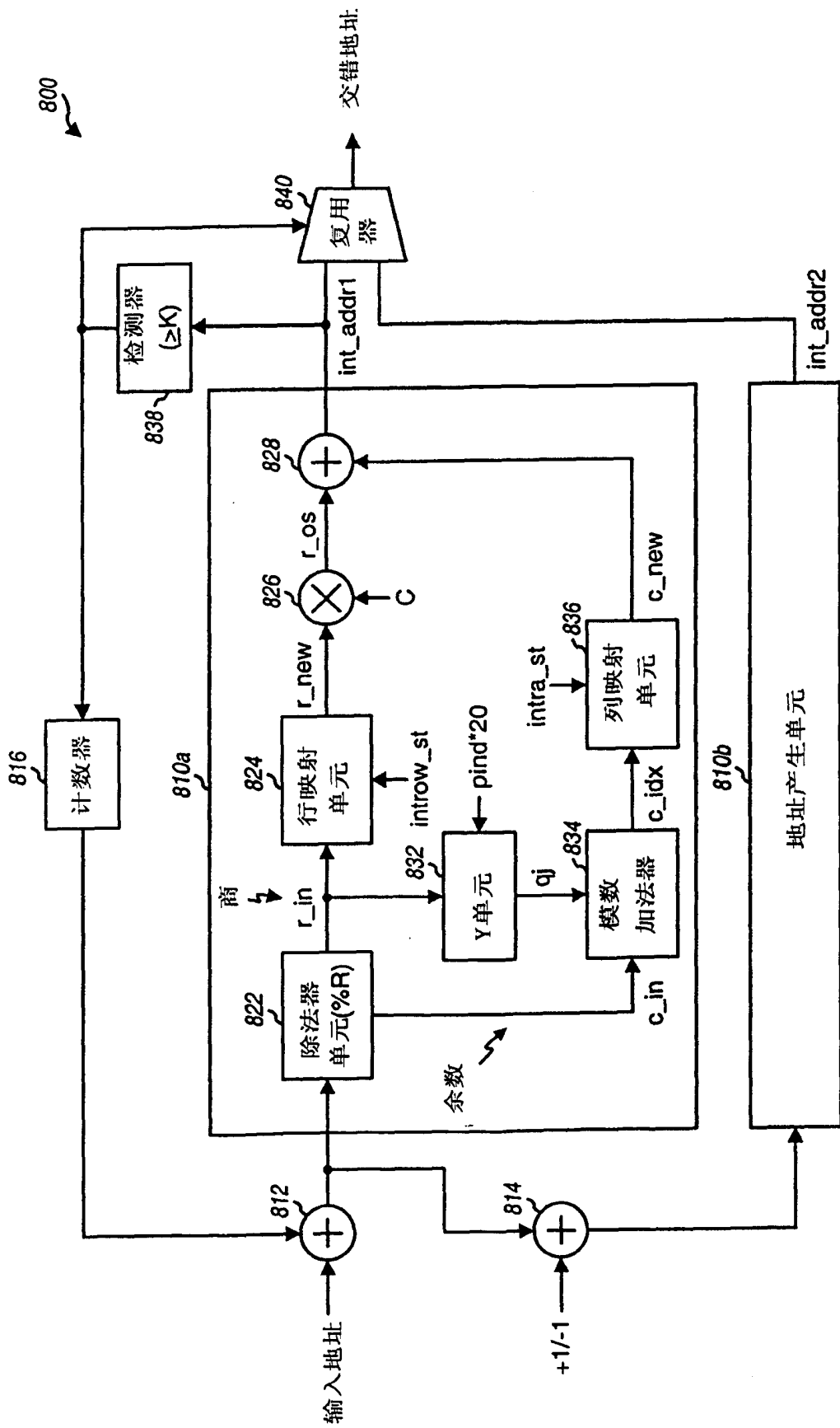


图 8

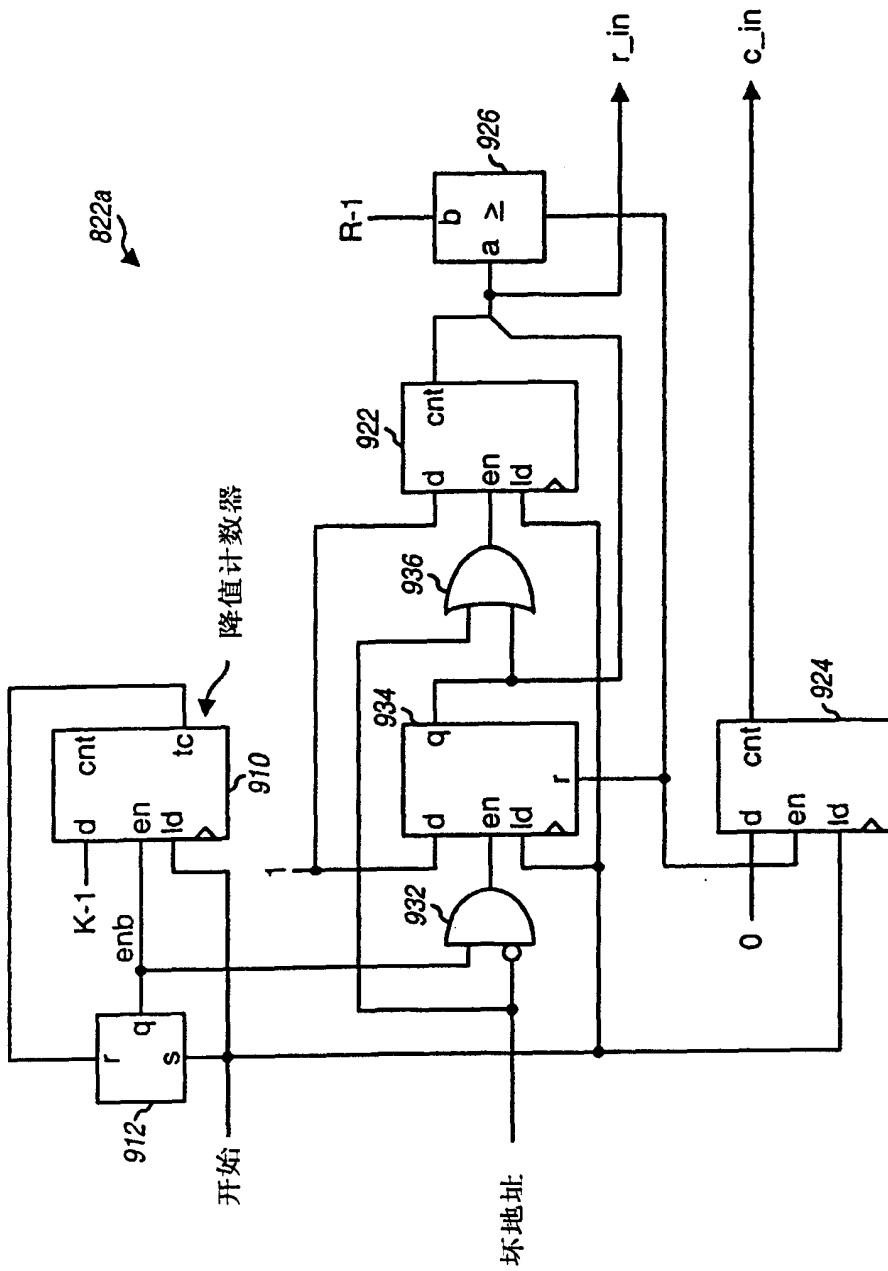


图 9