

1. 一种无线通信设备,包括:

第一 CRC 代码器,所述第一 CRC 代码器被配置成对传送块生成第一块 CRC 奇偶比特,所述第一块 CRC 奇偶比特是基于第一生成器多项式,所述第一 CRC 代码器将所述第一块 CRC 奇偶比特与所述传送块关联;

分割实体,所述分割实体具有耦合到所述第一 CRC 代码器的输入,所述分割实体被配置成在关联后将所述传送块分割成多个代码块;

第二代码器,所述第二代码器被配置成对每个代码块生成第二块 CRC 奇偶比特,所述第二块 CRC 奇偶比特中的每个是基于第二生成器多项式,第二 CRC 代码器将所述第二块 CRC 奇偶比特与每个代码块相关联,与每个代码块相关联的所述第二块 CRC 奇偶比特是基于对应代码块而生成的所述第二块 CRC 奇偶比特;

信道编码器,所述信道编码器被配置成对包括关联的第二块 CRC 奇偶比特的每个所述代码块进行编码;

从下述生成器多项式的组中选择所述第一生成器多项式和第二生成器多项式,该组包括:

$$D^{24}+D^{23}+D^6+D^5+D+1;$$

$$D^{24}+D^{21}+D^{20}+D^{17}+D^{15}+D^{11}+D^9+D^8+D^6+D^5+D+1;和$$

$$D^{24}+D^{23}+D^{18}+D^{17}+D^{14}+D^{11}+D^{10}+D^7+D^6+D^5+D^4+D^3+D+1。$$

2. 如权利要求 1 所述的设备,其中所述第一生成器多项式是 $D^{24}+D^{23}+D^6+D^5+D+1$ 。

3. 如权利要求 1 所述的设备,其中所述第二生成器多项式是 $D^{24}+D^{23}+D^6+D^5+D+1$ 。

4. 一种在无线通信发射机中的方法,该方法包括:

对传送块生成第一块 CRC 奇偶比特,所述第一块 CRC 奇偶比特是基于第一生成器多项式;

将所述第一块 CRC 奇偶比特与所述传送块关联;

在关联后将所述传送块分割成多个代码块;

对每个代码块生成第二块 CRC 奇偶比特,所述第二块 CRC 奇偶比特的每个是基于第二生成器多项式;

将所述第二块 CRC 奇偶比特与每个代码块关联,与每个代码块关联的所述第二块 CRC 奇偶比特是基于对应代码块而生成的所述第二块 CRC 奇偶比特;

对包括关联的第二块 CRC 奇偶比特的每个所述代码块进行信道编码;

在信道编码后连接所述代码块,

从下述生成器多项式的组中选择所述第一生成器多项式和第二生成器多项式,该组包括:

$$D^{24}+D^{23}+D^6+D^5+D+1;$$

$$D^{24}+D^{21}+D^{20}+D^{17}+D^{15}+D^{11}+D^9+D^8+D^6+D^5+D+1;和$$

$$D^{24}+D^{23}+D^{18}+D^{17}+D^{14}+D^{11}+D^{10}+D^7+D^6+D^5+D^4+D^3+D+1。$$

5. 一种在无线通信接收机中的方法,该方法包括:

使与多个接收代码块中的每一个关联的第二块 CRC 奇偶比特解除关联,基于第二生成器多项式并且基于所述第二块 CRC 奇偶比特所关联的对应代码块来生成所述第二块 CRC 奇偶比特;

通过在移除关联的第二块 CRC 奇偶比特之后连接所述代码块,来形成估计传送块,所述估计传送块具有与之关联的第一块 CRC 奇偶比特,其中与所述传送块关联的所述第一块 CRC 奇偶比特是基于第一生成器多项式,

基于所述第一生成器多项式来对所述估计传送块执行 CRC 校验,

从下述生成器多项式的组中选择所述第一生成器多项式和第二生成器多项式,该组包括:

$$D^{24}+D^{23}+D^6+D^5+D+1;$$

$$D^{24}+D^{21}+D^{20}+D^{17}+D^{15}+D^{11}+D^9+D^8+D^6+D^5+D+1; \text{和}$$

$$D^{24}+D^{23}+D^{18}+D^{17}+D^{14}+D^{11}+D^{10}+D^7+D^6+D^5+D^4+D^3+D+1。$$

6. 如权利要求 5 所述的方法,其中在对所述估计传送块执行所述 CRC 校验之前,对所述多个代码块执行 CRC 校验。

7. 如权利要求 6 所述的方法,其中只有在对所述多个代码块的所述 CRC 校验没有检测到任何错误时,才执行对所述估计传送块的所述 CRC 校验。

8. 一种在无线通信发射机中的方法,该方法包括:

对传送块生成第一块 CRC 奇偶比特,所述第一块 CRC 奇偶比特是基于第一生成器多项式生成的;

将所述第一块 CRC 奇偶比特与所述传送块关联;

在关联后交织所述传送块;

将所交织的传送块分割成多个代码块;

对每个代码块生成第二块 CRC 奇偶比特,每个所述第二块 CRC 奇偶比特是基于第二生成器多项式而生成的;

将第二块 CRC 奇偶比特与每个代码块关联,与每个代码块关联的所述第二块 CRC 奇偶比特是基于对应代码块而生成的所述第二块 CRC 奇偶比特;

对包括关联的第二块 CRC 奇偶比特的代码块中的每个进行信道编码;

在信道编码后连接所述代码块,

从下述生成器多项式的组中选择所述第一生成器多项式和第二生成器多项式,该组包括:

$$D^{24}+D^{23}+D^6+D^5+D+1;$$

$$D^{24}+D^{21}+D^{20}+D^{17}+D^{15}+D^{11}+D^9+D^8+D^6+D^5+D+1; \text{和}$$

$$D^{24}+D^{23}+D^{18}+D^{17}+D^{14}+D^{11}+D^{10}+D^7+D^6+D^5+D^4+D^3+D+1。$$

9. 如权利要求 8 所述的方法,其中在比特级执行交织所述传送块的步骤。

10. 如权利要求 8 所述的方法,其中通过置换组来执行交织所述传送块的步骤,其中每个组包含多个比特。

11. 如权利要求 8 所述的方法,其中

在关联所述第一块 CRC 奇偶比特后交织所述传送块以及在关联所述第一块 CRC 奇偶比特后将所交织的传送块分割成多个代码块的步骤

相当于

在关联所述第一块 CRC 奇偶比特之后将所述传送块分割成多个代码块,并且单独地交织所述代码块。

无线通信系统中的多层循环冗余校验码

技术领域

[0001] 本公开总体上涉及无线通信,更具体地涉及使用循环冗余校验(CRC)码的数据代码化。

背景技术

[0002] CRC 代码化普遍地用于检测在无线通信系统中发射的数据中的错误。例如,在演进的 3GPP LTE 的规范中,建议是,基于整个传送块(TB)产生 24 个 CRC 奇偶比特。然后,将该 24 个 CRC 奇偶比特附加到 TB,之后,将 TB 分割成多个代码块(CB)。在 LTE 提议中,也基于每个代码块(CB)计算 24 个 CRC 奇偶比特,并且之后将 24 个 CRC 奇偶比特附加到对应的 CB。在 LTE 提议中,在基于传送块生成 CRC 奇偶比特和基于代码块生成 CRC 奇偶比特时使用同样的生成器多项式。第一传送 CRC 编码有助于接收设备检测残留误差。建议是,可以由接收设备使用该代码块的 CRC 编码以减少 turbo 解码处理的次数,或减少 turbo 解码迭代的次数,或减少 turbo 解码器存储器的使用率。然后代码块的传输之前,例如,使用 turbo 码对其进行信道编码。

[0003] 通过对以下伴随附图的具体实施方式的仔细考虑,本公开的各个方面、特点和优势,对于本领域的技术人员将是显而易见的。为了简明,这些图已经被简化了,并且不一定按比例绘制。

附图说明

[0004] 图 1 图解无线通信系统。

[0005] 图 2 是包括发射机的无线通信设备的示意框图。

[0006] 图 3 是包括接收机的无线通信设备的示意框图。

具体实施方式

[0007] 在图 1 中,无线通信系统 100 包括一个或多个固定基本基础设施单元,其形成在地理区域上分布的网络。基本单元也可被称为接入点、接入终端、节点 B、eNode-B 或本领域使用的其他术语。一个或多个基本单元 101 和 102 服务于服务区域内(例如,小区或小区扇区内)的多个远程单元 103 和 110。远程单元可以是固定单元或移动终端。远程单元也可以被称为订户单元、移动站、用户、终端、订户站、用户设备(UE),或本领域使用的其他术语。

[0008] 一般来说,基本单元 101 和 102 在时域和/或频域中向服务的远程单元发射下行链路通信信号 104 和 105。经由上行链路通信信号 106 和 113,远程单元 103 和 110 与一个或多个基本单元 101 和 102 通信。一个或多个基本单元可以包括服务远程单元的一个或多个发射机和一个或多个接收机。远程单元也可以包括一个或多个发射机和一个或多个接收机。

[0009] 在一个实施例中,通信系统利用 OFDMA 或用于上行链路传输的下一代基于单载波(SC)的 FDMA 架构,诸如交织 FDMA(IFDMA)、本地 FDMA(LFDMA)、具有 IFDMA 或 LFDMA 的 DFT

扩展 OFDM(DFT-SOFDM)。在基于 OFDM 的系统中,无线电资源包括 OFDM 码元,其可被划分为时隙,时隙是子载波的分组。作为示例的基于 OFDM 的协议是正在开发的 3GPP LTE 协议。

[0010] 通过循环冗余校验(CRC)对例如传送块的协议数据单元提供错误检测。图 2 是无线通信设备 200,它或它其中的一部分被配置成在无线通信系统中进行传输的 CRC 代码数据。在图 1 中,在基站 101 和远程单元 103 之间发射该数据。在 3GPP LTE 的实现中,数据或协议数据单元是传送块。CRC 代码化一般发生在基本单元和远程单元的发射机中。在图 2 中,发射机包括第一 CRC 代码器实体 210,其被配置成对传送块 202 生成第一块 CRC 奇偶比特。

[0011] 整个传送块通常用于计算或生成 CRC 奇偶比特。将传送块中要递送到层 1 的比特标记为 $a_0, a_1, a_2, a_3, \dots, a_{A-1}$,将奇偶比特标记为 $p_0, p_1, p_2, p_3, \dots, p_{L-1}$ 。A 是传送块规模,而 L 是奇偶比特的数目。在一种 3GPP LTE 的实施方式中,第一块包括 24 个 CRC 奇偶比特,即 L 被设置为 24 比特,但更普遍的是,块可能包括某个其他数目的奇偶比特。奇偶比特的计算基于第一 CRC 生成器多项式 212。通常将第一块 CRC 奇偶比特与传送块关联。在图 2 中,第一块 CRC 奇偶比特 204 被附加到传送块 202。在其他的实施例中,第一块 CRC 奇偶比特被附加到传送块的某其他部分。

[0012] 在图 2 中,无线通信设备 200 还包括分割实体 214。附加了第一 CRC 奇偶比特的传送块被递送给该分割实体。输入到代码块分割的输入比特序列被标记为 $b_0, b_1, b_2, b_3, \dots, b_{B-1}$,其中 $B > 0$ 。分割实体将具有相关的第一 CRC 奇偶比特块的传送块 202 分割为多个代码块 206、207、208。第二 CRC 代码器实体 216 被配置成对多个代码块 206、207、208 中的每一个生成第二块 CRC 奇偶比特。第二块 CRC 奇偶比特中的每个基于第二生成器多项式 218。在一个 3GPP LTE 的实施方式中,第二块 CRC 奇偶比特也包括 24 个 CRC 奇偶比特。然后第二 CRC 奇偶比特块中的每个被关联到该第二块 CRC 奇偶比特所基于的对应代码块。在图 2 中,第二块 CRC 奇偶比特 230、232 和 234 被附加到对应的代码块 206、207 和 208。可以对每个分割的代码块连续地实施该处理。在一些实施方式中,分割是有条件的。例如,如果 B 大于最大代码块的规模,例如 $Z = 6144$,则执行对输入比特序列的分割,并且额外的 $L = 24$ 比特的 CRC 序列被附加到每个代码块,其中 CRC 比特是基于第二 CRC 生成器多项式被计算的。如果 B 小于或等于最大代码块规模,则代码块分割 214 是透明的,且不需要第二块 CRC 奇偶比特。

[0013] 发明人已经认识到,对图 2 中的第一和第二 CRC 代码器 210 和 216 使用相同的生成器多项式会导致在一级或两级的 CRC 校验中仍有错误事件未被检测到。如果在两级 CRC 校验中仍有错误事件未被检测到,则接收机将接受不正确块作为正确块。因此,最好是减少可能在两级 CRC 校验中仍未被检测的错误事件。对于 CRC 码,等同于非零码字的错误事件不能被 CRC 解码器检测到(还应指出,码字的非循环移位版本仍然是码字)。因此,当对于第一和第二 CRC 代码器选择了相同的生成器多项式时,代码块的系统部分中的不能检测的错误事件在两级 CRC 校验中将仍未被检测到,并且接收机可能会接受不正确的块。

[0014] 发明人已经进一步认识到,通过对于 CRC 编码传送块和分割的代码块使用不同的生成器多项式,可以提高两级 CRC 的错误检测能力。因此,在一些实施例中,第一和第二生成器多项式是不同的。在一个实施例中,例如,第一和第二生成器多项式具有至少一个不同的因子。在另一个实施例中,第一和第二生成器多项式不共享共同的因子。在另一个实施

例中,第一和第二生成器多项式具有不同多项式系数集合。在其他的实施例中,第一和第二生成器多项式由其它特征来区别。更一般地,第一和第二多项式可由这些和 / 或其他特征的组合来区别。在一个实施例中,第一和第二生成器多项式共享因子 $(D+1)$ 和 / 或共同的阶数。然而,在其他的实施例中,第一和第二生成器多项式如下进一步所述。

[0015] 在一个实施方式中,从包括至多共享 $(D+1)$ 因子的以下 24 次 CRC 生成器多项式的组中选择第一和第二阶数的生成器多项式:

[0016] $g_{\text{CRC24,a}}(D) = D^{24} + D^{23} + D^6 + D^5 + D + 1$ 。该生成器多项式可以被分解成以下形式: $(D+1)(D^{23} + D^5 + 1)$;

[0017] $g_{\text{CRC24,b}}(D) = D^{24} + D^{21} + D^{20} + D^{17} + D^{15} + D^{11} + D^9 + D^8 + D^6 + D^5 + D + 1$ 。该生成器多项式可以被分解成以下形式:

[0018] $g_{\text{CRC24,b}}(D) = (D+1)(D^{23} + D^{22} + D^{21} + D^{19} + D^{18} + D^{17} + D^{14} + D^{13} + D^{12} + D^{11} + D^8 + D^5 + 1)$;

[0019] $g_{\text{CRC24,c}}(D) = D^{24} + D^{23} + D^{18} + D^{17} + D^{14} + D^{11} + D^{10} + D^7 + D^6 + D^5 + D^4 + D^3 + D + 1$ 。该生成器多项式可以被分解成以下形式: $(D+1)(D^{23} + D^{17} + D^{13} + D^{12} + D^{11} + D^9 + D^8 + D^7 + D^5 + D^3 + 1)$;

[0020] $g_{\text{CRC24,d}}(D) = D^{24} + D^{23} + D^{14} + D^{12} + D^8 + 1$ 。该生成器多项式可以被分解成以下形式: $= (D+1)(D^3 + D^2 + 1)(D^{10} + D^8 + D^7 + D^6 + D^5 + D^4 + D^3 + D + 1)(D^{10} + D^9 + D^6 + D^4 + 1)$;

[0021] $g_{\text{CRC24,e}}(D) = D^{24} + D^{21} + D^{20} + D^{16} + D^{15} + D^{14} + D^{13} + D^{12} + D^{11} + D^{10} + D^9 + D^8 + D^4 + D^3 + 1$;

[0022] $g_{\text{CRC24,f}}(D) = D^{24} + D^{22} + D^{20} + D^{19} + D^{18} + D^{16} + D^{14} + D^{13} + D^{11} + D^{10} + D^8 + D^7 + D^6 + D^3 + D + 1$ 。该生成器多项式可以被分解成以下形式: $(D+1)^2(D^{11} + D^9 + D^8 + D^7 + D^6 + D^3 + 1)(D^{11} + D^9 + D^8 + D^7 + D^5 + D^3 + D^2 + D + 1)$;

[0023] $g_{\text{CRC24,g}}(D) = D^{24} + D^{22} + D^{21} + D^{20} + D^{19} + D^{17} + D^{16} + D^8 + D^7 + D^5 + D^4 + D^3 + D^2 + 1$ 。该生成器多项式可以被分解成以下形式:

[0024] $(D+1)^2(D^{22} + D^{19} + D^{18} + D^{16} + D^{15} + D^{13} + D^{11} + D^9 + D^7 + D^6 + D^4 + D^3 + 1)$;

[0025] $g_{\text{CRC24,h}}(D) = D^{24} + D^{21} + D^{20} + D^{17} + D^{13} + D^{12} + D^3 + 1$ 。该生成器多项式可以被分解成以下形式: $(D+1)^2(D^{11} + D^{10} + D^9 + D^8 + D^7 + D^6 + D^5 + D^2 + 1)(D^{11} + D^{10} + D^9 + D^7 + D^6 + D^5 + D^4 + D^3 + 1)$;

[0026] $g_{\text{CRC24,i}}(D) = D^{24} + D^{22} + D^{12} + D^{10} + D^9 + D^2 + D + 1$ 。该生成器多项式可以被分解成以下形式: $(D+1)^2(D^{11} + D^9 + 1)(D^{11} + D^9 + D^7 + D^5 + D^3 + D + 1)$; 和

[0027] $g_{\text{CRC24,j}}(D) = D^{24} + D^{22} + D^{20} + D^{19} + D^{17} + D^{16} + D^{15} + D^{14} + D^{10} + D^7 + D^6 + D^5 + D^4 + D^2 + 1$ 。该生成器多项式可以被分解成以下形式: $(D^{12} + D^{11} + D^7 + D^4 + D^2 + D + 1)(D^{12} + D^{11} + D^8 + D^7 + D^5 + D^4 + D^2 + D + 1)$ 。

[0028] 在另一个实施方式中,第一和第二阶数的生成器多项式选自包括上述 24 次 CRC 生成器多项式之一和上述 24 次 CRC 生成器多项式之一的互反 (reciprocal) 的组。 $n-k$ 次的 $g(D)$ 的互反多项式是 $D^{n-k}g(D^{-1})$ 。例如, $g_{\text{CRC24a}}(D)$ 的互反是 $1 + D + D^{18} + D^{19} + D^{23} + D^{24} = (D+1)(D^{23} + D^{18} + 1)$ 。在一个更具体的实施方式中,第一和第二阶数的生成器多项式是选自 $g_{\text{CRC24a}}(D)$ 和 $g_{\text{CRC24a}}(D)$ 的互反的组。

[0029] 在另一个实施方式中,第一和第二生成器多项式选自包括下列生成器多项式的生成器多项式组: $D^{24} + D^{23} + D^6 + D^5 + D + 1$; $D^{24} + D^{21} + D^{20} + D^{17} + D^{15} + D^{11} + D^9 + D^8 + D^6 + D^5 + D + 1$ 和 $D^{24} + D^{23} + D^{18} + D^{17} + D^{14} + D^{11} + D^{10} + D^7 + D^6 + D^5 + D^4 + D^3 + D + 1$ 。在另一个实施方式中,第一和第二生成器多项式至少之一是 $D^{24} + D^{23} + D^6 + D^5 + D + 1$ 。

[0030] 可使用如下多项式算法实现 L 比特 CRC 代码器。在 CRC 计算中,使用 $a_0, a_1, a_2, a_3, \dots, a_{A-1}$ 标记针对 CRC 计算的输入比特并且使用 $p_0, p_1, p_2, p_3, \dots, p_{L-1}$ 标记校验比特。 A 是输入序列的规模,而 L 是校验比特的数目。通过循环生成器多项式或 CRC 生成器多项式

($g_{CRC}(D)$) 生成具有 L 个 CRC 奇偶比特的校验比特。以系统的形式执行编码,这意味着在 $GF(2)$ 中,该多项式:

$$[0031] \quad a_0 D^{A+L-1} + a_1 D^{A+L-2} + \dots + a_{A-1} D^L + p_0 D^{L-1} + p_1 D^{L-2} + \dots + p_{L-2} D + p_{L-1}$$

[0032] 当该多项式被 $g_{CRC}(D)$ 除时,该多项式产生了等于 0 的余数。由 b_0 、 b_1 、 b_2 、 b_3 …… b_{B-1} 来标记 CRC 附加之后的比特,其中 $B = A+L$ 。 a_k 和 b_k 之间的关系是:

$$[0033] \quad b_k = a_k, \text{ 当 } k = 0, 1, 2, \dots, A-1$$

$$[0034] \quad b_k = p_{(L-1-(k-A))}, \text{ 当 } k = A, A+1, A+2, \dots, A+L-1。$$

[0035] 在另一种方法中, a_k 和 b_k 之间的关系如下:

$$[0036] \quad b_k = a_k, \text{ 当 } k = 0, 1, 2, \dots, A-1$$

$$[0037] \quad b_k = p_{(k-A)}, \text{ 当 } k = A, A+1, A+2, \dots, A+L-1。$$

[0038] 在图 2 中,无线通信设备 200 还包括被配置成对包括相关的第二块 CRC 奇偶比特的每个代码块进行编码的信道编码实体 222。信道编码实体可实现为多个不同形式,包括但不限于 turbo 编码器,或卷积编码器等其它信道编码器。发射机还包括被配置成在信道编码后连接代码块的连接实体 224。该连接实体的输出耦合到用于传输的功率放大器。连接实体 224 可以执行一系列的一个或多个其它操作来准备用于传输的代码块,例如速率匹配、HARQ 冗余版本选择、信道交织、比特加扰、映射到物理信道资源、比特到码元映射、IFFT、DFT 扩展等。

[0039] 图 3 是无线通信设备 300,它或它其中的一部分被配置成接收和解码使用 CRC 码编码的数据。该设备 300 接收代码块 206、207 和 208,其中每一个代码块分别与 CRC 奇偶比特的相应块 230、232 和 234 关联。这些 CRC 代码化的代码块对应于由图 2 中的发射机 200 发射的代码块。该设备 300 包括 CRC 移除实体 310,其被配置成使与多个接收的代码块的每一个关联的第二块 CRC 奇偶比特解除关联,从而剩余代码块 206、207 和 208。第二块 CRC 奇偶比特的移除基于第二 CRC 生成器多项式 312。由图 3 中的 CRC 移除实体执行的功能本质上反转了由图 2 中的 CRC 编码器实体 216 执行的处理。因此,图 3 中的 CRC 移除实体 310 使用的第二 CRC 生成器多项式 312 与图 2 中第二 CRC 编码器实体 216 使用的用以生成第二块 CRC 奇偶比特并且将其与代码块关联的第二 CRC 生成器多项式 218 是相同的。

[0040] 在图 3 中,设备 300 包括连接器实体 314,其被配置成形成估计的传送块 205,该估计的传送块 205 具有与之关联的第一块 CRC 奇偶比特 204。在由 CRC 移除实体 310 移除相关的第二块 CRC 奇偶比特后,该连接器实体 314 连接代码块 206、207 和 208。图 3 中的连接器实体 314 本质上反转了由图 2 中的分割实体 214 执行的处理。因此,在图 3 中,第一块 CRC 奇偶比特 204 实质上图 2 中的与传送块 202 关联的第一块 CRC 奇偶比特 204 对应。

[0041] 在图 3 中,设备 300 包括 CRC 解码器实体 316,其被配置成基于第一生成器多项式 318 对估计的传送块 202 执行 CRC 校验。应当注意,图 3 中的第一生成器多项式 318 对应于图 2 中的第一生成器多项式 212。CRC 校验确定由接收机恢复的估计的传送块 205 是否对应于所发射的传送块,例如图 2 中的传送块 202。一旦通过 CRC 校验检测到错误,则将估计的传送块视为与所发射的传送块不对应,可以请求重传。如果没有检测到错误,则估计的传送块被视为与所发射的传送块相对应,并且被递送到更高的层。众所周知,CRC 校验具有不能检测到错误的特定概率,这是对 CRC 码性能的度量。

[0042] 在一些实施例中,该设备 300 包括第二 CRC 解码器实体 320,其被配置成对在接收

机接收的多个代码块 206、207 和 208 执行 CRC 校验。在代码块被连接以形成估计的传送块之前,因此在对估计的传送块 205 执行 CRC 校验之前,CRC 解码器实体 320 对代码块执行校验。在一些实施例中,与估计的传送块 205 相关的第一块 CRC 奇偶比特基于第一生成器多项式 318,第一生成器多项式 318 不同于形成与代码块相关的第二块 CRC 奇偶比特的基础的第二生成器多项式 312。然而,在其他的实施例中,第一和第二生成器多项式与下面进一步讨论的相同。在一些实施例中,可由接收设备使用代码块的 CRC 编码来减少 turbo 解码处理的次数,或减少 turbo 解码迭代的次数,或减少 turbo 解码器存储器的使用率。

[0043] 在包括第二 CRC 解码器实体 320 的一些实施例中,对估计的传送块 205 执行 CRC 校验是有条件的。在一个实施方式中,只有在对多个代码块 206、207 和 208 执行的 CRC 校验未检测到任何错误时,才对估计的传送块 205 执行 CRC 校验。在图 3 中,条件控制器 322 基于在代码块 206、207 和 208 上是否检测到错误来提供信号,该信号控制 CRC 解码器 316 是否对估计的传送块 205 执行 CRC 校验。在一些实施方式中,如果第二 CRC 解码器实体 320 在代码块上检测到错误,则代码块被重传。在某些实施方式中,如果检测到错误,则传送块被重传。

[0044] 在图 2 的替代实施例中,第一和第二生成器多项式 212 和 218 用于生成共享至少一个因子的第一和第二块 CRC 奇偶比特。在一个实施方式中,第一和第二生成器多项式是相同的。在该替代的实施例中,在关联第一块 CRC 奇偶比特之后在传送块 202 中执行交织操作。由交织实体 240 执行交织。在一个实施例中,在分割之前执行交织,因此交织实体 240 位于第一 CRC 编码实体和分割实体之间。在替代的实施例中,在分割后,但在代码块的 CRC 编码之前,执行交织。在该替代的实施例中,交织实体 240 位于分割实体和第二 CRC 编码实体 216 之间。交织模式可以下列方式定义:在关联第一块 CRC 奇偶比特 204 之后交织传送块 202 并且将交织的传送块分割成多个代码块相当于在关联第一块 CRC 奇偶比特之后将传送块 202 分割成多个代码块并且单独交织代码块 206、207 和 208。这种等效可以是概念的,其中在分割前执行交织。替代地,等效可以是实际的,其中通过在分割之后执行多个子交织来实现交织器。在一个实施方式中,在比特级上执行传送块 202 的交织。在另一个实施方式中,通过置换组来执行传送块交织,其中每一个组包含多个比特。

[0045] 在某些实施方式中,图 2 中的交织器置换不允许相同的不能被检测的错误事件被保持在第一级 CRC 校验和第二级的 CRC 校验之间,从而使错误检测属性得以改善。虽然交织是一种选择,但重新排序(或交织)之外的其它转换也是可行的,该其它转换将不允许相同的不能被检测的错误被保持在第一级 CRC 校验和第二级 CRC 校验之间。在传送块和代码块之间引入的交织一次可交织一比特或一个字节(或其他规模比特的组)。如果交织是在传送块级执行的,则在关联第一块 CRC 奇偶比特之后需要一个与传送块关联的交织器。替代地,交织可以在代码块级执行(即其中不同片段的比特不混合的交织)。为了代码块级别的交织,可能需要总共 C 个子交织器,其中 C 是消息片段的数目。第 i 个子交织器与第 i 个代码块关联。传送块级的或代码块级的交织器可以有简单的格式,诸如反转,即从尾到头读取比特;或循环移位;比特反转等。交织可能在接收机上需要额外的等待时间或电路,尽管通过合适地选择交织器可以减少其确切数量。

[0046] 在图 3 中,在第一和第二生成器多项式相同并且由发射设备将传送块或代码块交织的实施方式中,接收设备包括解交织器实体 328。如果在发射设备中对传送块没有发生交

织,则解交织实体 328 位于图 3 中示出的接收机设备中的连接实体 314 之后。如果在发射设备中对代码块发生交织,则解交织实体位于接收设备中的连接实体之前。

[0047] 与不进行交织的在第一和第二 CRC 代码器中使用相同的生成器多项式相比,通常需要更多的电路和 / 或存储器,以使用用于第一和第二 CRC 代码器的不同生成器多项式并且还在将第一块 CRC 奇偶比特关联到传送块之后实现交织。然而,错误检测的增强性能可能超过与增加的复杂性相关的成本。

[0048] 虽然已经以建立所有权并且使得本领域技术人员能够制造和使用本公开及其最佳实施方式的方式对其进行了描述,但是应当理解和认识到,在不背离本发明的范围和精神的情况下,存在与此公开的示范性的实施例等同的实施例,并且可对其作修改和变化,它们不由示范性的实施例限定,而是体现在附加的权利要求中。

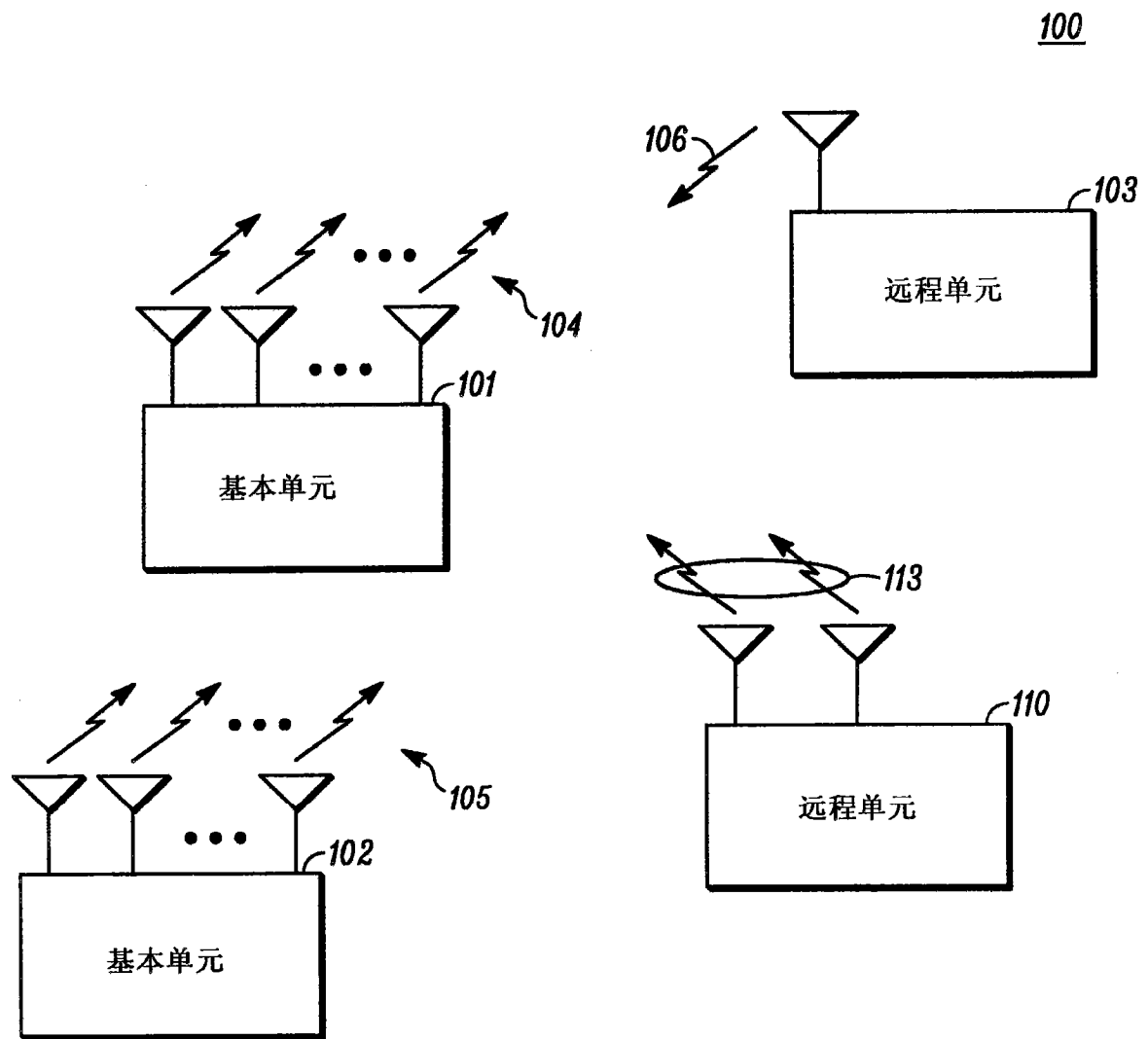


图 1

200

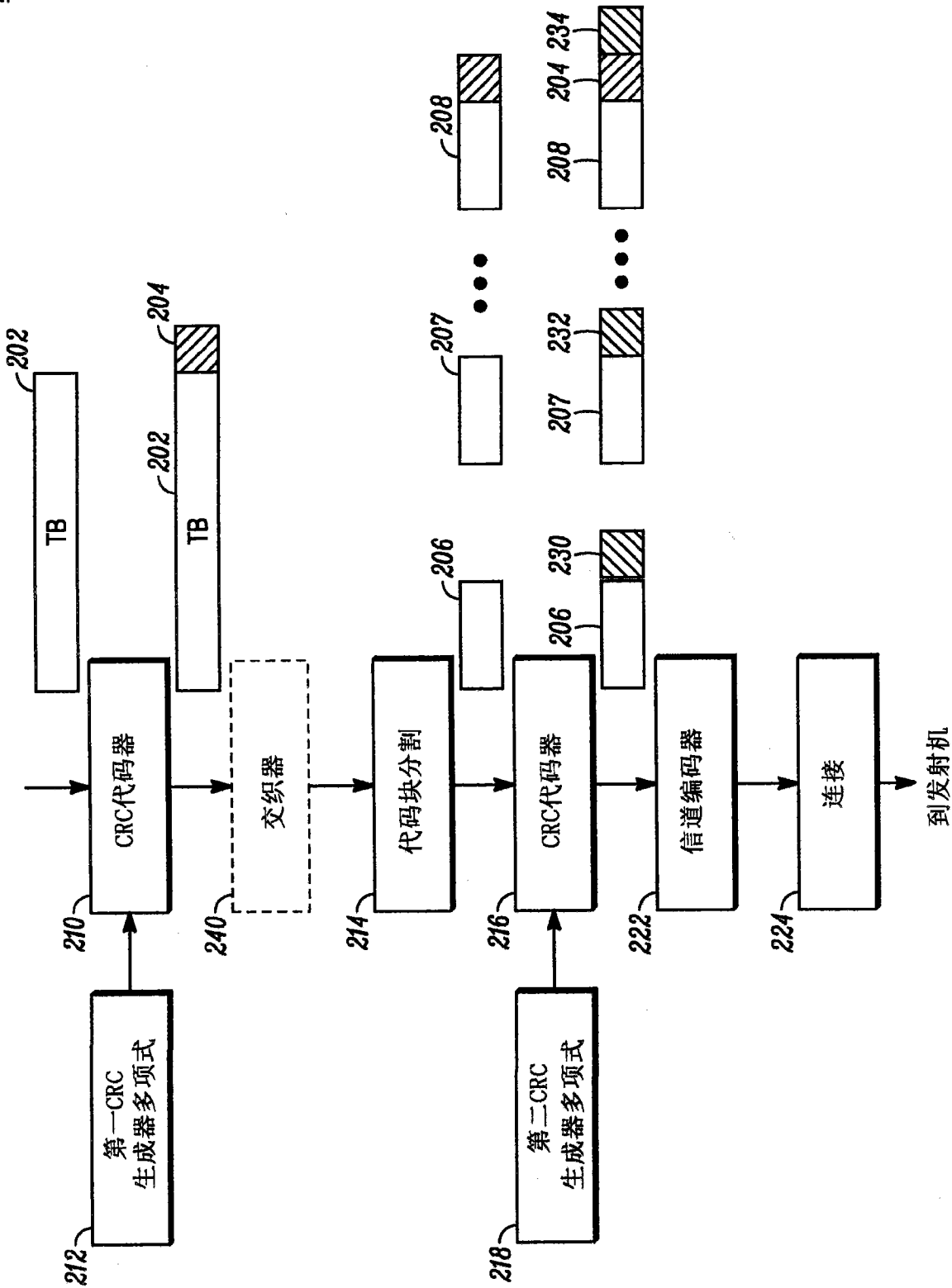


图 2

300

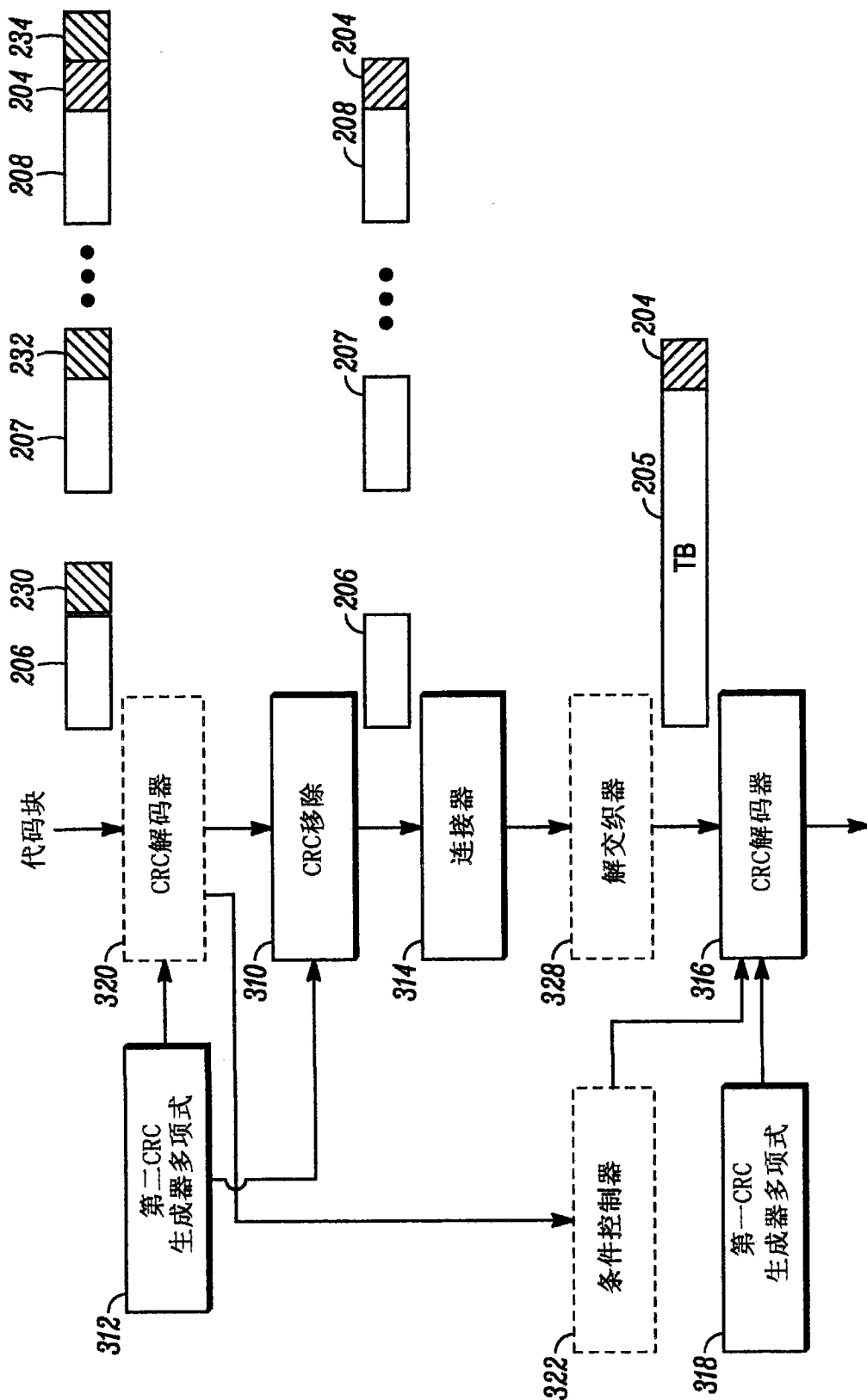


图 3