(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 103562995 B
(45)授权公告日 2017.02.15

(21)申请号 201280026298.2

(22)申请日 2012.05.25

(65)同一申请的已公布的文献号
申请公布号 CN 103562995 A

(43)申请公布日 2014.02.05

(30)优先权数据
13/149,905 2011.06.01 US

(85)PCT国际申请进入国家阶段日
2013.11.29

(86)PCT国际申请的申请数据
PCT/CN2012/076089 2012.05.25

(87)PCT国际申请的公布数据
WO2012/163250 EN 2012.12.06

(73)专利权人 国际商业机器公司
地址 美国纽约

(72)发明人 R·D·塞德斯亚恩
T·米特尔洛尔泽 P·J·西格
田中启介

(74)专利代理机构 中国国际贸易促进委员会专利商标事务所 11038
代理人 曾琳

(51)Int.Cl.
G11B 5/008(2006.01)

(56)对比文件
CN 1276886 A,2000.12.13,

审查员 姚宗妮

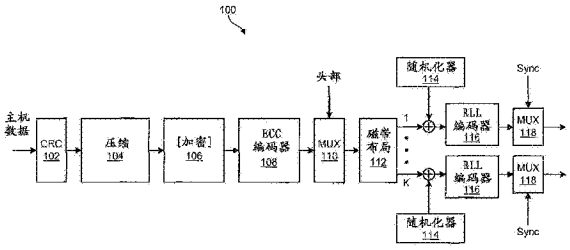
权利要求书2页 说明书8页 附图10页

(54)发明名称

减轻虚假VFO检测的轨道相关数据随机化

(57)摘要

描述了一种用于使数据随机化以减轻虚假VFO检测的方法。在一个实施例中,这种方法包括同时接收多个输入数据流。每个输入数据流与磁带介质上的不同轨道相关联。同时对输入数据流加扰以产生多个随机化数据流。对输入数据流加扰,以使得即使在输入数据流中的相应比特模式相同的情况下也在随机化数据流中产生不同的比特模式。将随机化数据流同时写到它们在磁带介质上相关联的数据轨道。还描述了相应的装置。



CN 103562995 B

1.一种用于使数据随机化以减轻虚假VFO检测的方法,所述方法包括:

同时接收多个不同的输入数据流,其中每个输入数据流与磁带介质上的不同轨道相关联;

同时对输入数据流加扰以产生多个随机化数据流,其中加扰包括即使在输入数据流中的相应比特模式相同的情况下也在随机化数据流中产生不同的比特模式;以及

将随机化数据流同时写到它们在磁带介质上相关联的数据轨道,使得即使在输入数据流中的相应比特模式相同的情况下也向每个轨道写不同的数据比特模式。

2.如权利要求1所述的方法,其中,加扰包括使用多个数据随机化器来对进入的数据流加扰。

3.如权利要求2所述的方法,还包括在对进入的数据流加扰之前不同地初始化所述多个数据随机化器中的每一个。

4.如权利要求2所述的方法,其中,所述多个数据随机化器中的每一个包括线性反馈移位寄存器。

5.如权利要求4所述的方法,其中,所述线性反馈移位寄存器是线性反馈移位寄存器的斐波那契实现。

6.如权利要求4所述的方法,还包括在对进入的数据流加扰之前用不同的预设值来初始化所述多个数据随机化器的线性反馈移位寄存器。

7.如权利要求6所述的方法,其中,所述多个数据随机化器中的每一个产生重复的伪随机序列。

8.如权利要求7所述的方法,其中,所述重复的伪随机序列包含2^L-1个比特,其中L是每个线性反馈移位寄存器中的比特的数目。

9.如权利要求7所述的方法,其中,初始化所述多个数据随机化器的线性反馈移位寄存器包括用来自所述重复的伪随机序列的不同的预设值来初始化线性反馈移位寄存器。

10.如权利要求9所述的方法,还包括基本上使重复的序列内所述多个数据随机化器的预设值之间的最小距离最大化。

11.一种用于使数据随机化以减轻虚假VFO检测的装置,所述装置包括:

多个数据随机化器,并行接收多个不同的输入数据流,其中每个输入数据流和每个数据随机化器与磁带介质上的一特定轨道相关联;以及

所述多个数据随机化器还被配置为对输入数据流加扰以产生多个随机化数据流,使得即使在输入数据流中的相应比特模式相同的情况下数据随机化器也在随机化数据流中产生不同的比特模式,以及将随机化数据流同时写到它们在磁带介质上相关联的数据轨道,使得即使在输入数据流中的相应比特模式相同的情况下也向每个轨道写不同的数据比特模式。

12.如权利要求11所述的装置,其中,所述多个数据随机化器中的每一个在结构上是相同的。

13.如权利要求11所述的装置,其中,在对进入的数据流加扰之前所述多个数据随机化器中的每一个被不同地初始化。

14.如权利要求11所述的装置,其中,所述多个数据随机化器中的每一个包含线性反馈移位寄存器。

15.如权利要求14所述的装置，其中，每个线性反馈移位寄存器是线性反馈移位寄存器的斐波那契实现。

16.如权利要求14所述的装置，其中，在对进入的数据流加扰之前每个线性反馈移位寄存器用不同的预设值被初始化。

17.如权利要求16所述的装置，其中，每个线性反馈移位寄存器产生重复的伪随机序列。

18.如权利要求17所述的装置，其中，所述重复的伪随机序列包含2^L-1个比特，其中L是每个线性反馈移位寄存器中的比特的数目。

19.如权利要求17所述的装置，其中，每个线性反馈移位寄存器用来自所述重复的伪随机序列的不同的预设值被初始化。

20.如权利要求19所述的装置，其中，在重复的序列内基本上使不同的预设值之间的最小距离最大化。

减轻虚假VFO检测的轨道相关数据随机化

技术领域

[0001] 本发明涉及磁带记录，更特别地，涉及用于消除虚假VFO模式检测或使虚假VFO模式检测减到最少的装置和方法。

背景技术

[0002] 在当前的线性磁带开放(linear tape open，LTO)和企业级磁带驱动器中，可变频率振荡器(variable-frequency oscillator，VFO)模式(pattern)是数据记录和同步的基本成分。这样的VFO模式可用于对齐时钟频率和比特位置。它们还可用于执行主复位，从而使得电路和/或定时被复位到初始状况。如果存在于磁带上的VFO模式未被检测到，或者不是VFO模式的数据被错误地确定为VFO模式，那么当尝试从磁带读取数据时可发生严重的系统问题。在一些情况下，磁带上的数据可能不可恢复。

[0003] 虽然意欲使VFO模式相对于普通数据中遇到的模式而言是独一无二的，但在普通的记录数据中仍然可能出现匹配模式(以下称为“虚假VFO模式”)。避免此问题的一种方法是要求多个轨道同时包含VFO模式。此方法在每个轨道上的数据独立的情况下工作良好，因为每个轨道上的数据不太可能同时包含相同的虚假VFO模式。然而，在多个轨道具有相同或非常相似的数据的情况下，比如在某些测试数据模式被写入到磁带的情况下，此方法可能失败。

[0004] 另一种避免虚假VFO检测的方法是使用较长的VFO模式并且要求所有比特匹配该VFO模式。此方法的一个缺点是它增大了实际VFO模式将被错过的可能性，因为任何差错或介质缺陷可能导致较长的VFO模式未被检测到。此方法的另一个缺点是较长的VFO模式降低了存储格式效率，因为较长的VFO模式消耗更多的存储空间。

[0005] 另一种避免虚假VFO检测的方法是调整VFO检测窗口的大小。然而，调整VFO检测窗口的大小提供了折衷。如果VFO检测窗口大，则检测到实际VFO模式的可能性降低，因为任何差错或介质缺陷可能导致实际VFO模式未被检测到。如果VFO检测窗口小，则检测到虚假VFO模式的可能性增大。

[0006] 鉴于前述情况，所需要的是避免虚假VFO检测或使虚假VFO检测减到最少的改进的装置和方法。理想情况下，这样的装置和方法将使某些类型的数据(比如测试数据或其他重复的数据模式)将引起VFO被虚假地检测到的机率最小化。还需要的是在不降低存储格式效率的情况下提供上述益处的装置和方法。

发明内容

[0007] 本发明是响应于当前的技术发展水平、特别是响应于本领域中尚未被当前可用的装置和方法完全解决的问题和需求而开发出来的。因此，本发明被开发来提供用于减轻磁带驱动器中的虚假VFO检测的装置和方法。本发明的特征和优点将通过以下描述和所附权利要求而变得更完全清楚，或者可通过实践如以下所述的发明而获知。

[0008] 与前述相符的，这里公开了一种用于使数据随机化以减轻虚假VFO检测的方法。在

一个实施例中，这种方法包括同时接收多个输入数据流。每个输入数据流与磁带介质上的不同轨道相关联。同时对输入数据流加扰以产生多个随机化数据流。对输入数据流加扰，以使得即使在输入数据流中的相应比特模式相同的情况下也在随机化数据流中产生不同的比特模式。将随机化数据流同时写到它们在磁带介质上相关联的数据轨道。

[0009] 在本发明的另一方面中，一种用于使数据随机化以减轻虚假VFO检测的装置包括多个数据随机化器，这些数据随机化器并行接收多个输入数据流。每个输入数据流和每个数据随机化器与磁带介质上的一特定轨道相关联。多个数据随机化器被配置为对输入数据流加扰以产生多个随机化数据流。数据随机化器被配置为即使在输入数据流中的相应比特模式相同的情况下也在随机化数据流中产生不同的比特模式。

附图说明

[0010] 为了能更容易理解本发明的优点，将通过参照附图中示出的特定实施例来给出对以上简要描述的发明的更具体描述。要理解的是，这些附图只描绘了本发明的典型实施例并且因此不应被认为是对其范围的限制，将通过使用附图来更加具体且详细地描述和说明本发明，在附图中：

[0011] 图1是示出磁带驱动器的数据流的一个示例的高级别框图；

[0012] 图2是示出在磁带上出现的实际VFO模式和未对齐的虚假VFO模式的一个示例的高级别框图；

[0013] 图3示出了可以如何使用表决逻辑来对图2中所示的未对齐的虚假VFO模式避免虚假VFO检测；

[0014] 图4是示出在磁带上出现的实际VFO模式和对齐的虚假VFO模式的示例的高级别框图；

[0015] 图5示出了表决逻辑如何不能对图4中所示的对齐的虚假VFO模式避免虚假VFO检测；

[0016] 图6示出了用于用不同预设值初始化的每个轨道的数据随机化器，从而即使在每个轨道的输入数据流相同的情况下也对每个轨道产生不同的随机化数据流；

[0017] 图7是示出线性反馈移位寄存器的一个示例的高级别框图，在此示例中是线性反馈移位寄存器的斐波那契(Fibonacci)实现；

[0018] 图8示出了包括用预设值初始化的线性反馈移位寄存器的数据随机化器的一个示例；

[0019] 图9是示出可用于对K=32个轨道初始化数据随机化器的预设值的第一示例的表格；以及

[0020] 图10是示出可用于对K=32个轨道初始化数据随机化器的预设值的第二示例的表格。

具体实施方式

[0021] 将容易理解，如这里概括描述并在附图中示出的本发明的组件可按多种不同配置来布置和设计。从而，以下对附图中表示的本发明的实施例的更详细描述并不想要限制要求保护的本发明的范围，而只是代表了根据本发明的当前设想的实施例的某些示例。通过

参照附图将最好地理解当前描述的实施例，附图中相似的部件始终由相似的标号来标注。

[0022] 本领域技术人员将会明白，本发明可实现为装置、系统、方法或计算机程序产品。另外，本发明可采取如下形式：硬件实施例、被配置为操作硬件的软件实施例(包括固件、驻留软件、微代码等等)、或者组合软件要素和硬件要素两者的实施例。这些实施例中的每一个可由一个或多个模块或方框来表示。另外，本发明可采取在存储有计算机可用程序代码的任何有形表达介质中实现的计算机可用存储介质的形式。

[0023] 一个或多个计算机可用或计算机可读存储介质的任何组合可被利用来存储计算机程序产品。计算机可用或计算机可读存储介质可以例如是——但不限于——电的、磁的、光的、电磁的、红外的或半导体的系统、装置或设备。计算机可读存储介质的更具体示例(非穷举列表)可包括以下：具有一条或多条导线的电连接、便携式计算机盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦除可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑盘只读存储器(CDROM)、光存储设备、或者磁存储设备。在本文档的上下文中，计算机可用或计算机可读存储介质可以是任何可包含、存储或传输程序供指令执行系统、装置或者设备使用或与之结合使用的介质。

[0024] 可以以一种或多种编程语言的任何组合来编写用于执行本发明的操作的计算机程序代码，所述编程语言包括面向对象的编程语言(比如Java、Smalltalk、C++等)以及传统的过程式编程语言(比如“C”编程语言或类似的编程语言)。用于实现本发明的计算机程序代码也可用诸如汇编语言之类的低级别编程语言来编写。

[0025] 下面可参照根据本发明的实施例的方法、装置、系统和计算机程序产品的流程图和/或框图描述本发明。要理解，流程图和/或框图的每个方框以及流程图和/或框图中的各方框的组合可由计算机程序指令或代码来实现。计算机程序指令可以被提供给通用计算机、专用计算机或其他可编程数据处理装置的处理器以产生一种机器，使得经由计算机或其他可编程数据处理装置的处理器执行的这些指令创建用于实现流程图和/或框图的一个或多个方框中指定的功能/动作的装置。

[0026] 也可以把计算机程序指令存储在可指引计算机或其他可编程数据处理装置以特定方式工作的计算机可读存储介质中，从而存储在计算机可读存储介质中的指令产生出包括实现流程图和/或框图的一个或多个方框中指定的功能/动作的指令装置的制造品(article of manufacture)。计算机程序指令也可被加载到计算机或其他可编程数据处理装置上，以使得一系列操作步骤在该计算机或其他可编程装置上被执行来产生由计算机实现的过程，从而在该计算机或其他可编程装置上执行的指令提供用于实现流程图和/或框图的一个或多个方框中指定的功能/动作的过程。

[0027] 参考图1，示出了一高级别框图，其示出了磁带驱动器(比如LTO或企业级磁带驱动器)的数据流100。如图所示，循环冗余校验(CRC)模块102从主机设备接收可变长度数据块。CRC模块102可向这些块添加CRC信息。压缩模块104随后可压缩这些块，并且加密模块106可以可选地对这些块加密。这些数据块随后可被分割成固定大小的数据集合，这些数据集合进而又可被分割成固定大小的子数据集合(SDS)。每个SDS可被组织成二维数据阵列。每个SDS数据阵列随后可被传递到ECC编码器108。ECC编码器108可为数据阵列中的每一行和数据阵列中的每一列生成ECC奇偶校验。该行ECC奇偶校验和列ECC奇偶校验可被附加到该阵列。

[0028] 一旦ECC奇偶校验被附加到阵列，复用器110就可向数据阵列(包括列ECC奇偶校验和行ECC奇偶校验)中的行添加头部。在某些实施例中，头部包含差错检测码，比如CRC。在图示实施例中，头部没有受到上述的行或列ECC奇偶校验的保护。在其他实施例中，头部可在计算行ECC奇偶校验之前被添加到数据阵列。行ECC奇偶校验随后可被计算为包括头部。这将确保头部受到至少一维ECC奇偶校验的保护。在其他实施例中，头部可在已计算行ECC奇偶校验之后被添加到数据阵列。行ECC奇偶校验随后可被重新计算来包括头部。

[0029] 磁带布局模块112可将扩展的数据阵列(包括ECC奇偶校验和头部)在K个不同轨道上且按不同顺序分布在磁带上。可利用随机化器114来进一步处理由磁带布局模块112产生的数据序列以对数据执行额外的信号处理。更具体而言，随机化器114可用于对进入的数据进行变换以创建输出，该输出是“1”和“0”的伪随机序列。这从进入的数据中尽可能地去除了周期性。游程长度受限(RLL)编码器116可处理经随机化的数据流以防止在数据中出现不想要的模式(例如，长串的“1”或“0”或其他不期望的模式)。复用器118可将诸如可变频率振荡器(VFO)模式、同步字符等等之类的同步信息复用到信息中以使其在被读取时能够被同步。

[0030] 所得到的数据随后可被发送到写驱动器(未示出)，这些写驱动器可使得电流流经记录头元件以生成磁通量并从而将数据写到磁记录介质的轨道。通常，复用器110右侧的每个方框或模块对数据执行不同的变换以使其更适合于磁记录。

[0031] 参考图2，如前所述，VFO模式可被插入到记录数据中以用于同步和其他目的。在当前的LTO和企业级磁带驱动器中，VFO模式是由具有4比特周期的2T磁记录模式构成的，其中T对应于记录的比特。采取NRZI记法的VFO模式是1 0 1 0 1 0 1 0……类型的序列，其中“0”表示在记录磁化方向上没有变化，“1”表示在记录磁化方向上反转。

[0032] RLL编码器116被设计为限制记录数据内的类似VFO(VFO-like)的2T模式的最大长度，以防止在读回过程期间在数据模式内错误地检测到VFO模式。这保证了在数据中将不会检测到类似VFO模式，除非读回数据由于诸如噪声或介质缺陷之类的差错而损坏。然而，在存在噪声时，如果噪声出现在VFO检测窗口内，则可能发生虚假VFO检测。

[0033] 图2示出了在普通随机数据的情况下如何检测VFO模式。可以观察到，在记录在磁带500的轨道上的数据504中可出现类似VFO模式502。因为每个轨道上的数据504是独立的，所以可在沿轨道方向508上随机地分布类似VFO模式502。这与实际VFO模式506形成对比，实际VFO模式506是故意地在沿轨道方向508上对齐的。虽然在特定轨道中可能检测到虚假VFO模式，但磁带驱动器可包括表决逻辑电路来在N≤K个轨道的群组中检测VFO模式。该表决逻辑电路可以当且仅当在至少N’个轨道中检测到VFO模式时升起VFO检测标志，其中1<N’≤N。例如，在K=8、16或32个轨道的情况下，轨道的总数可被划分成四个轨道的群组，并且表决逻辑电路可以仅在如下情况下升起VFO检测标志：在每个四轨道群组中的四个轨道之中的三个轨道中在相同的沿轨道位置检测到VFO模式。此概念在图3中示出。

[0034] 参考图3，同时继续大体参考图2，虽然在一四个轨道群组中的每一个中的多个位置处检测到了类似VFO模式502，但仅当在四个轨道中同时出现指定数目的类似VFO模式502时才升起VFO检测标志。由于在任何给定时间仅出现单个类似VFO模式502，所以VFO表决电路不会升起VFO检测标志，如图3中所示。这与在若干个轨道上对齐的实际VFO模式506形成对比。如图3中所示，由于在相同的沿轨道位置检测到指定数目的VFO模式506，因此表决逻

辑电路升起VFO检测标志。

[0035] 参考图4，虽然表决逻辑电路在每上轨道上的数据独立的情况下工作良好，但是在类似VFO模式502在磁带500上同时出现的情况下它可能就不那么起作用了。对于测试数据或其他重复的数据模式可能发生这样的情形。一个这种场景在图4中示出。如图4中所示，类似VFO模式502(即，虚假VFO模式502)在多个轨道上同时出现。因为这些类似VFO模式502出现在相同的沿轨道位置，所以表决逻辑电路升起VFO检测标志(如图5中所示)以指示检测到了VFO模式。在此情况下，VFO模式被虚假地检测到。此示例表明VFO表决逻辑电路对于在相同的沿轨道位置出现的重复模式可能不能避免虚假VFO检测。

[0036] 参考图6，为了减轻如上所述的虚假VFO检测，以下公开一种新颖且改进的用于将数据写到磁带的技术。此技术不仅在随机数据的普通情况下而且在重复数据(比如重复的测试数据)的情况下都减少或防止虚假VFO检测。如下文将更详细描述的，该技术即使在对于每个轨道的进入的主机数据相同时也向每个轨道写不同的数据比特模式。这是通过使用不同的预设值(即，初始值)来初始化每个轨道的数据随机化器114而实现的。如下文将更详细说明的，在一个实施例中，该技术利用K个不同的预设值来初始化K个数据随机化器114，如图6中所示。此技术对于普通用户数据以及重复的测试数据都避免虚假VFO检测，同时确保实际VFO模式将被正确检测到。

[0037] 图7示出了线性反馈移位寄存器700的斐波那契实现。这种线性反馈移位寄存器700被用在当前的LTO和企业级磁带驱动器的随机化器114中，并且可用在用于实现本发明的随机化器114中。在线性反馈移位寄存器700的此实现方式中，所有的加法和乘法都是在具有两个元素的伽罗瓦(Galois)域(也称为有限域)GF(2)上执行的。具有元素{0,1}的GF(2)上的加法运算“+”对应于模2加法(0+0=0,0+1=1,1+0=1，并且1+1=0)，而GF(2)上的乘法运算“×”对应于整数乘法(0×0=0,0×1=0,1×0=0，并且1×1=1)。

[0038] 图7中的L位移位寄存器700的初始设置(即，种子、预设值或初始值)由二进制值$a_0,a_1,\ldots,a_{L-1}$表示。GF(2)乘法器的系数由二进制值$g_1,g_2,\ldots,g_{L-1}$表示。由线性反馈移位寄存器700生成的二进制序列是最大长度序列(m序列)，如果所生成的序列的周期对于包含至少一个非零成分的任何非零初始设置是2L-1比特的话。如果多项式$1+g_1x+\ldots+g_{L-1}x^{L-1}+x^L$是本原多项式，则线性反馈移位寄存器700生成m序列。图7中所示的线性反馈移位寄存器700生成的序列满足递归$a_n=a_{n-1}g_{L-1}+\ldots+a_{n-L+1}g_1+a_{n-L}$。

[0039] 图8示出了当前LTO驱动器中使用的数据随机化器114的示例。这种数据随机化器114可用于实现这里公开的改进的数据写技术。LTO磁带驱动器中的数据随机化是基于也被称为加性加扰的同步加扰的，同步加扰利用模2加法器(异或(XOR)逻辑门)把由线性反馈移位寄存器电路700(比如图7中所示的电路700)生成的伪随机二进制序列(m序列)添加到二进制输入数据流800。在LTO中，使用15位移位寄存器700，其将伪随机二进制序列的周期限制到$2^{15}-1=32767$比特。LTO中用来定义线性反馈移位寄存器700的序列的本原多项式由$g(x)=1+x+x^{15}$给出。

[0040] 在传统的LTO磁带驱动器中，所有K个轨道中使用的数据随机化器114在码字交织(CWI)之前的头部开始时被预设到相同的值。换言之，数据随机化器114中的线性反馈移位寄存器700的初始状态不依赖于数据被指派到的逻辑或物理轨道。在LTO中，所有K个轨道的数据随机化器114在CWI之前的每一个头部开始时被预设到相同的15比特值

(100000000000000)。所有K个轨道上的头部被同时记录,意味着它们出现在磁带500上的大致相同的沿轨道位置。

[0041] LTO数据随机化器114的初始设置可由具有15个整数成分的列向量a=[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]'来描述。在m个时钟周期(m次移位)之后LTO随机化器114的寄存器内容可利用如下的MATLAB语言来描述:

[0042] a=[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]';%LTO预设值

[0043] ag=gf(a,1);

[0044] A=horzcat(zeros(14,1),diag(ones(1,14)));

[0045] B=horzcat([1 1],zeros(1,13));

[0046] Q=vertcat(A,B);

[0047] Qg=gf(Q,1);

[0048] bg=(Qg^m)*ag;

[0049] 列向量bg包含m个时钟周期之后随机化器114的内容。上述MATLAB代码指定了在由m个时钟周期表征的任何期望时间处LTO数据随机化器114的移位寄存器内容。由于bg的第一成分表示数据随机化器114的输出比特804,所以图8中的随机化器114的输出804完全由MATLAB代码指定。在一个实施例中(比如结合图9论述的那个),初始预设值被选择为a=[0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]'。以上所示的MATLAB代码的第一行可由预设值a替换以描述m个时钟周期之后随机化器114的寄存器内容。

[0050] 在某些实施例中,可能希望使伪随机二进制序列中的任两个预设值之间的最小"距离"最大化,其中"距离"对应于将移位寄存器的内容从一个预设值移到下一个所需的时钟周期的数目。这可确保在任两个轨道中生成的伪随机序列之间的M个时钟周期的最大的最小移位。此最大的最小移位M可利用如下的MATLAB语言来计算:

[0051] M=max(min(mod(c-circshift(c,[01]),2^L-1)))

[0052] where

[0053] c=sort(mod([0:F:(K-1)*F],(2^L-1)))

[0054] 其中F是相邻轨道的预设值之间的时钟周期的数目,最大化发生在F的所有可能值上,并且c是以相对于第一轨道(轨道0)的时钟偏移给出的所有预设值的有序集。对于K=32轨道以及L=15寄存器,最大的最小移位M是1023个时钟周期。

[0055] 示例1

[0056] 考虑第一示例,其中K=32轨道并且L=15寄存器。利用这些值,数据随机化器114将以重复方式生成伪随机模式的32767比特序列——即,序列的周期是32767比特。在此示例中,用于初始化轨道0的数据随机化器114a的预设值是a=[0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]'。轨道1的预设值相对于轨道0的预设值移位了996字节(即,996×8比特)。这对应于相对于轨道0的预设值有F=7968个时钟周期的偏移。

[0057] 剩余轨道的预设值可按类似的方式计算,其中轨道t的预设值相对于轨道t-1偏移7968个时钟周期,其中t=1,2,3,…,31。在此实施例中,相对于轨道t-1偏移轨道t的预设值所需的时钟周期的数目正好对应于传统的CWI-4(包括头部在内具有四个交织的码字的码字交织)中包含的比特数。如果CWI-4的头部包含12字节并且CWI-4的数据包括四个246字节里德所罗门(Reed-Solomon)码字,则CWI-4将包含总共(12+4×246)×8=7968个比特。在这

种实施例中，非有序集c(即，以相对于第一轨道(轨道0)的时钟偏移表达的所有32个轨道的预设值的非有序集)可如下计算：

[0058] c=(mod([0:7968:31*7968],(2^15-1)))={0,7968,15936,23904,31872,7073,15041,23009,30977,6178,14146,22114,30082,5283,13251,21219,29187,4388,12356,20324,28292,3493,11461,19429,27397,2598,10566,18534,26502,1703,9671,17639}

[0059] 如上所示，预设值一旦到达32767比特就折回。有序集c(即，按时钟偏移的数目组织的所有32个轨道的预设值的有序集)可如下计算：

[0060] c=sort(mod([0:7968:31*7968],(2^15-1)))={0,1703,2598,3493,4388,5283,6178,7073,7968,9671,10566,11461,12356,13251,14146,15041,15936,17639,18534,19429,20324,21219,22114,23009,23904,26502,27397,28292,29187,30082,30977,31872}

[0061] 如上所示，在连续轨道中生成的伪随机模式的块的串接产生m序列。在此情况下生成的两个伪随机序列之间的时钟周期的最小偏移可如下计算：

[0062] min(mod(c-circshift(c,[01]),2^15-1))=895

[0063] 从而，在每个预设值之间存在至少895个时钟周期的距离。该值接近最大可能值M=1023个时钟周期。所有32个轨道的预设值在图9中以二进制形式示出。如图所示，每个预设值具有15比特，对应于图8中所示的线性反馈移位寄存器700的大小。

[0064] 示例2

[0065] 考虑第二示例，其中K=32轨道并且L=15寄存器。利用这些值，数据随机化器114将以重复方式生成伪随机模式的32767比特序列。在此示例中，用于初始化轨道0的数据随机化器114a的预设值是a=[0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]'。轨道1的预设值相对于轨道0的预设值移位了996比特。这对应于相对于轨道0的预设值有F=996个时钟周期的偏移。在这种实施例中，非有序集c是如下计算的：

[0066] c=(mod([0:996:31*996],(2^15-1)))={0,996,1992,2988,3984,4980,5976,6972,7968,8964,9960,10956,11952,12948,13944,14940,15936,16932,17928,18924,19920,20916,21912,22908,23904,24900,25896,26892,27888,28884,29880,30876}

[0067] 从以上可以观察到，由于F的小得多的值，预设值不会折回。从而，在没有任何排序的情况下，预设值的顺序从最小到最大。所有32个轨道的预设值在图10中以二进制形式示出。

[0068] 前述示例中的值和硬件/软件配置只是作为示例给出的，而并不想要是限制性的。这些值和/或配置在不同实施例中可变化。例如，线性反馈移位寄存器700不限于15位线性反馈移位寄存器，而是可包括不同大小的线性反馈移位寄存器。线性反馈移位寄存器的实现方式也可变化，这进而又可改变所生成的伪随机序列。于是，预设值以及预设值中的比特的数目也可变化。伪随机二进制序列中的任两个预设值之间的“距离”在不同实施例中也可变化。从而，虽然在上述示例中最小距离被最大化，但这并不是必要的，并且可能并非总是希望的。从而，可以使用预设值之间的其他距离或间距。其他值，例如轨道和在磁带驱动器中实现的相关联的数据随机化器的数目，也可变化。

[0069] 还应当认识到，并不是所有的K个数据随机化器114都必需用不同的预设值被编程，虽然在一些实施例(比如上述实施例)中可能是这样。在其他实施例中，K个数据随机化

器114中的某数目n的数据随机化器用相同的预设值被编程，其中n是小于K的整数。以上描述和所附权利要求意欲涵盖多个(即，至少两个)(并行操作的)数据随机化器114用不同预设值被编程的实施例。磁带驱动器的实现方式以及检测或虚假检测VFO模式的方式可指示需要多少个不同的预设值以及哪些随机化器114用相同或不同的预设值被编程。

[0070] 附图中的流程图和框图示出了根据本发明的各种实施例的系统、方法和计算机可用存储介质的可能实现的体系架构、功能和操作。在这点上，流程图或框图中的每个方框可以代表模块、程序段或代码的一部分，所述模块、程序段或代码的一部分包括一个或多个用于实现指定的(一个或多个)逻辑功能的可执行指令。还应当注意，在一些替代实现方式中，方框中所标注的功能可以以不同于附图中标注的顺序发生。例如，两个连续示出的方框实际上可以基本上并行地执行，或者这些方框有时可以按相反的顺序执行，这依赖于所涉及的功能。还要注意，框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合，可以用执行指定的功能或动作的专用的基于硬件的系统来实现，或者可以用专用硬件与计算机指令的组合来实现。

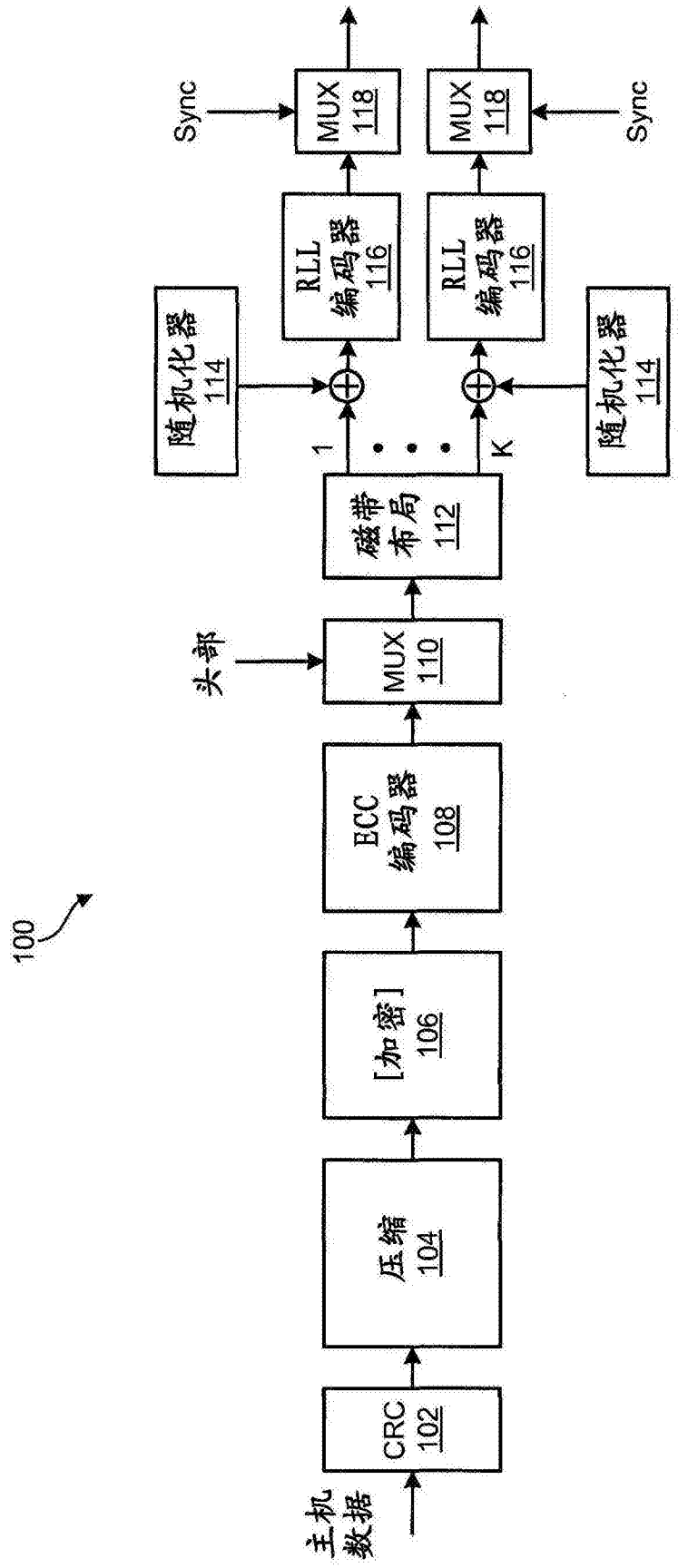

主机数据
CRC 102
压缩 104
[加密] 106
ECC 编码器 108
MUX 110
头部
磁带布局 112
1
K
随机化器 114
随机化器 114
RLL 编码器 116
RLL 编码器 116
MUX 118
MUX 118
Sync
Sync
100

图1

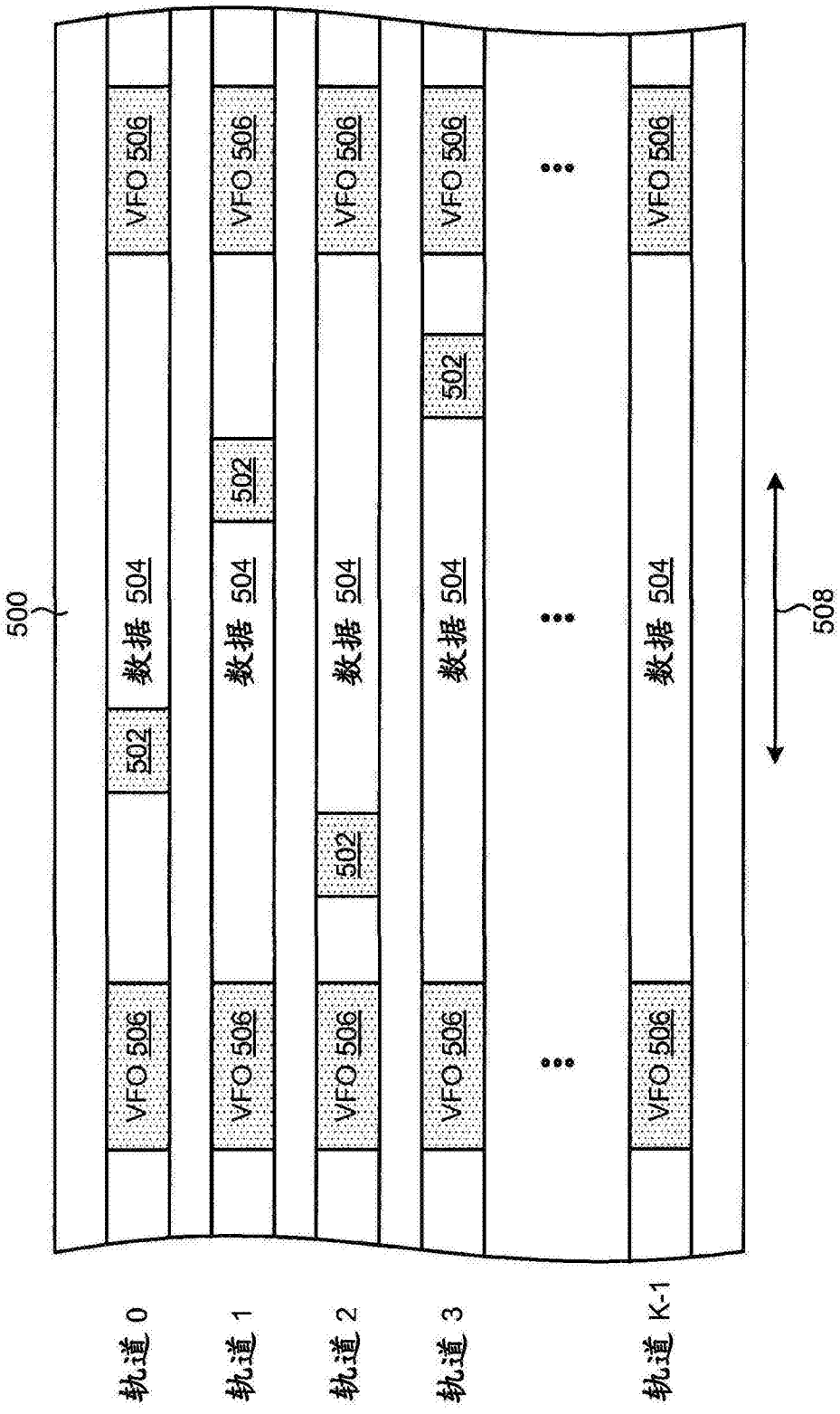


图2

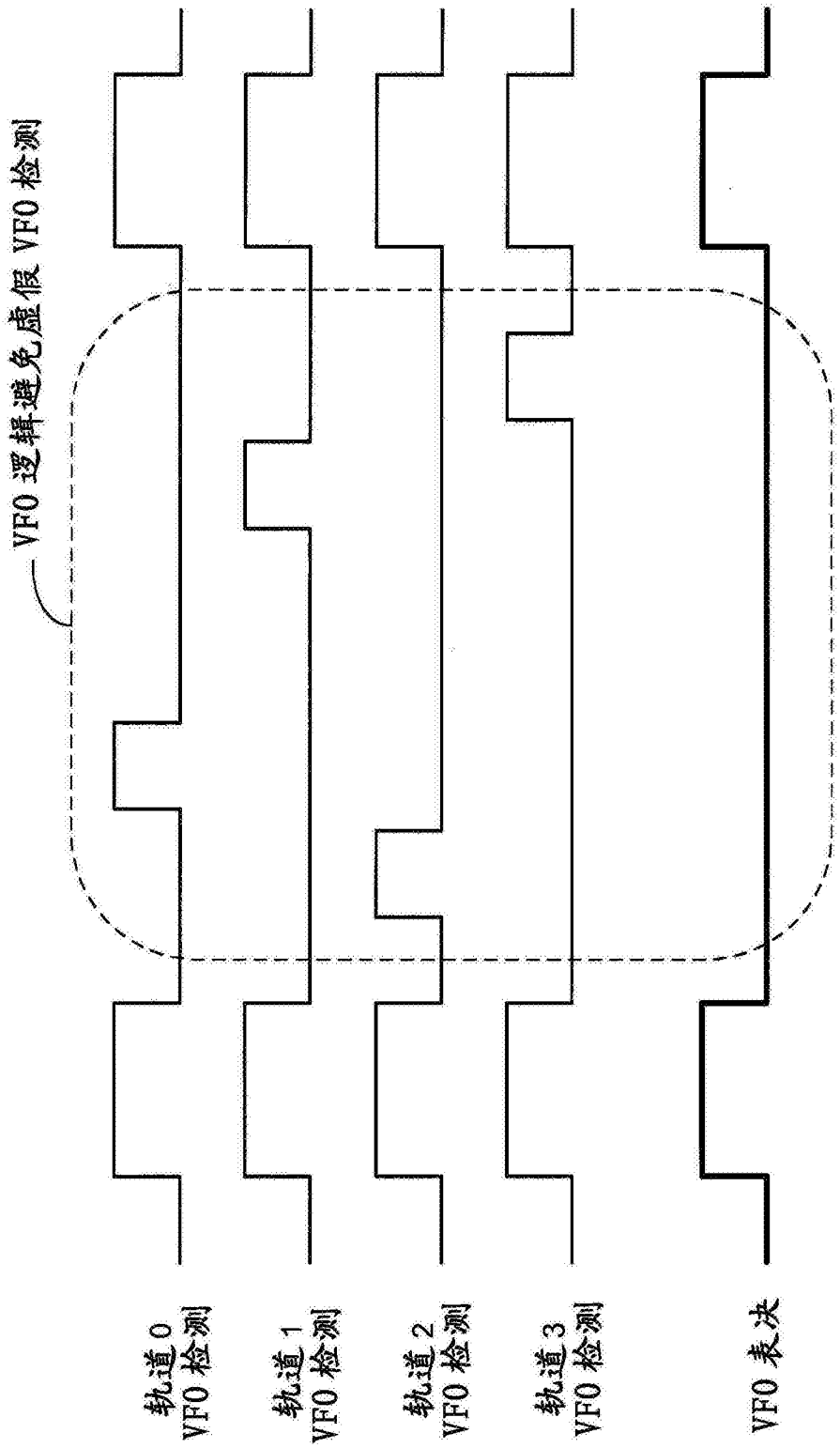

VFO 逻辑避免虚假 VFO 检测
轨道 0 VFO 检测
轨道 1 VFO 检测
轨道 2 VFO 检测
轨道 3 VFO 检测
VFO 表决

图3

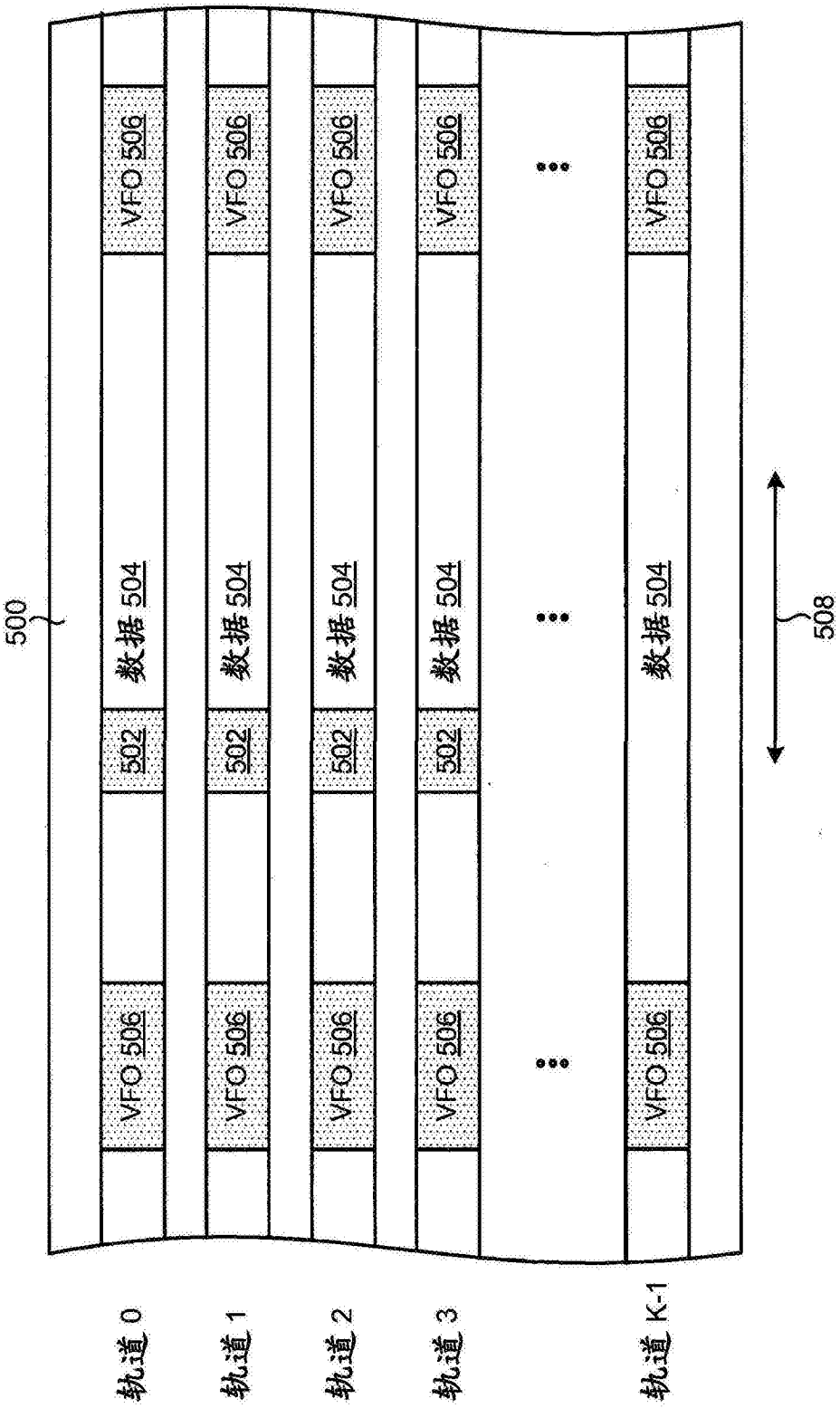


图4

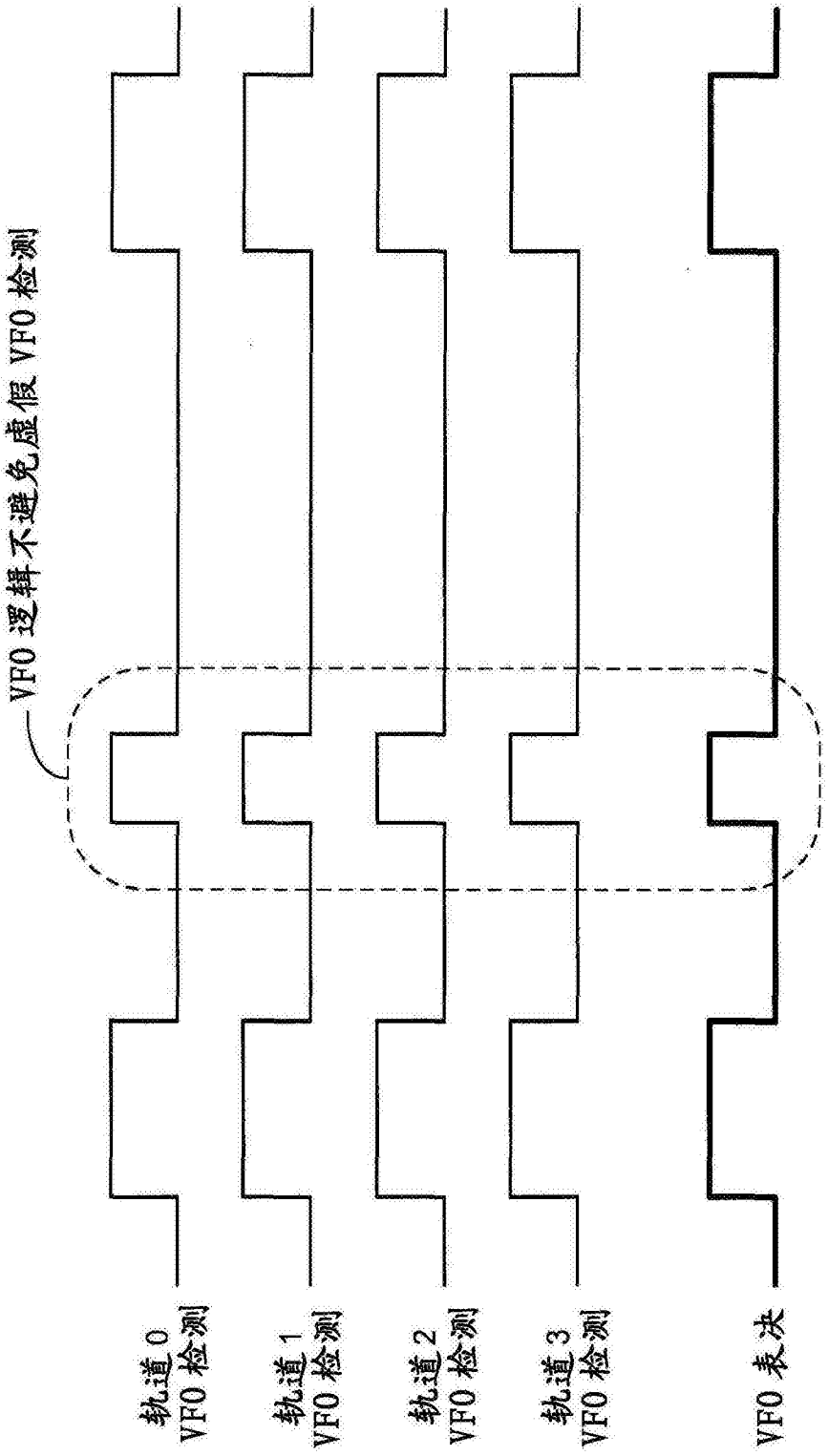

VFO 逻辑不避免虚假 VFO 检测
轨道 0 VFO 检测
轨道 1 VFO 检测
轨道 2 VFO 检测
轨道 3 VFO 检测
VFO 表决

图5

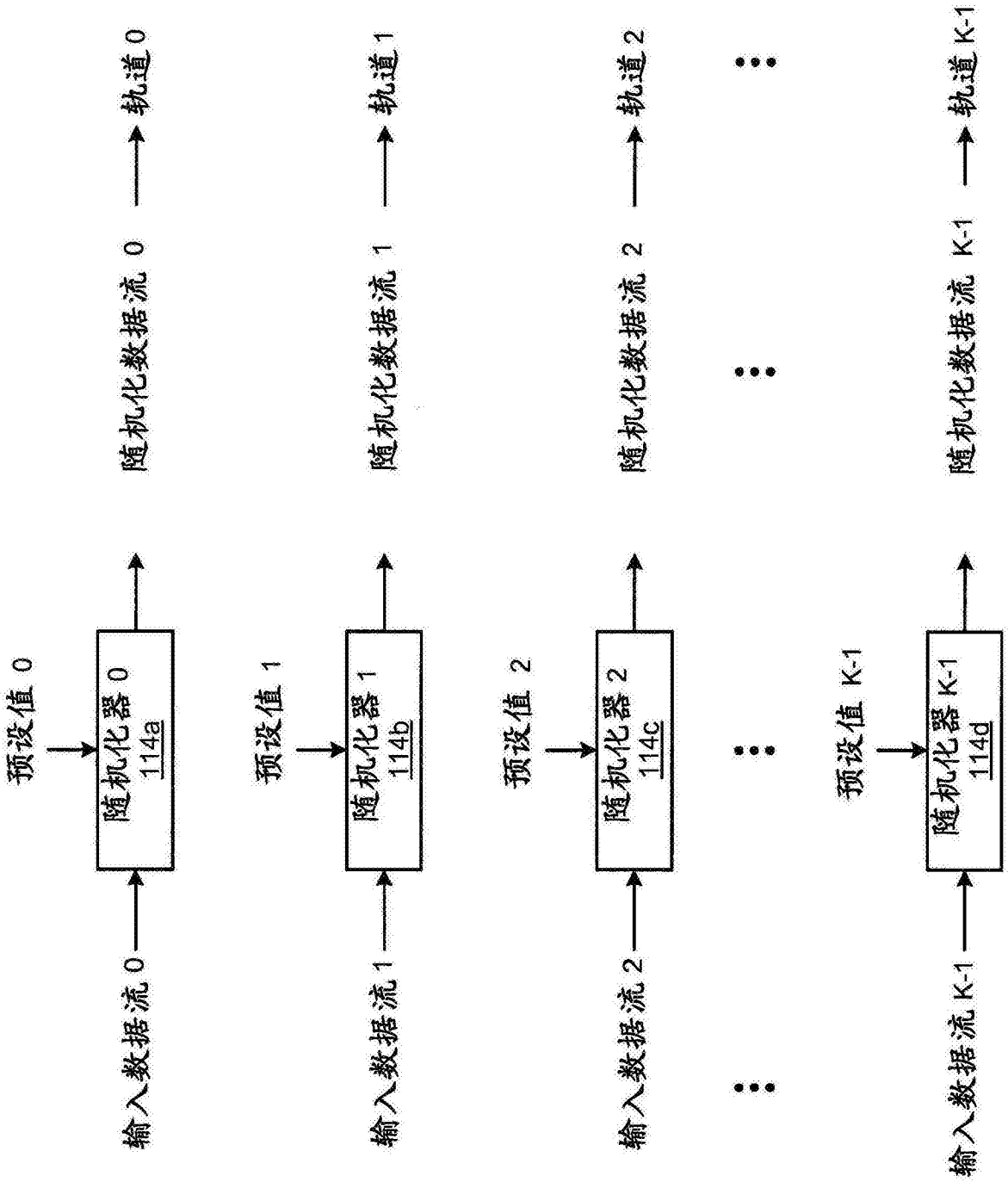

预设值 0
随机化器 0
114a
输入数据流 0
随机化数据流 0
轨道 0
预设值 1
随机化器 1
114b
输入数据流 1
随机化数据流 1
轨道 1
预设值 2
随机化器 2
114c
输入数据流 2
随机化数据流 2
轨道 2
预设值 K-1
随机化器 K-1
114d
输入数据流 K-1
随机化数据流 K-1
轨道 K-1

图6

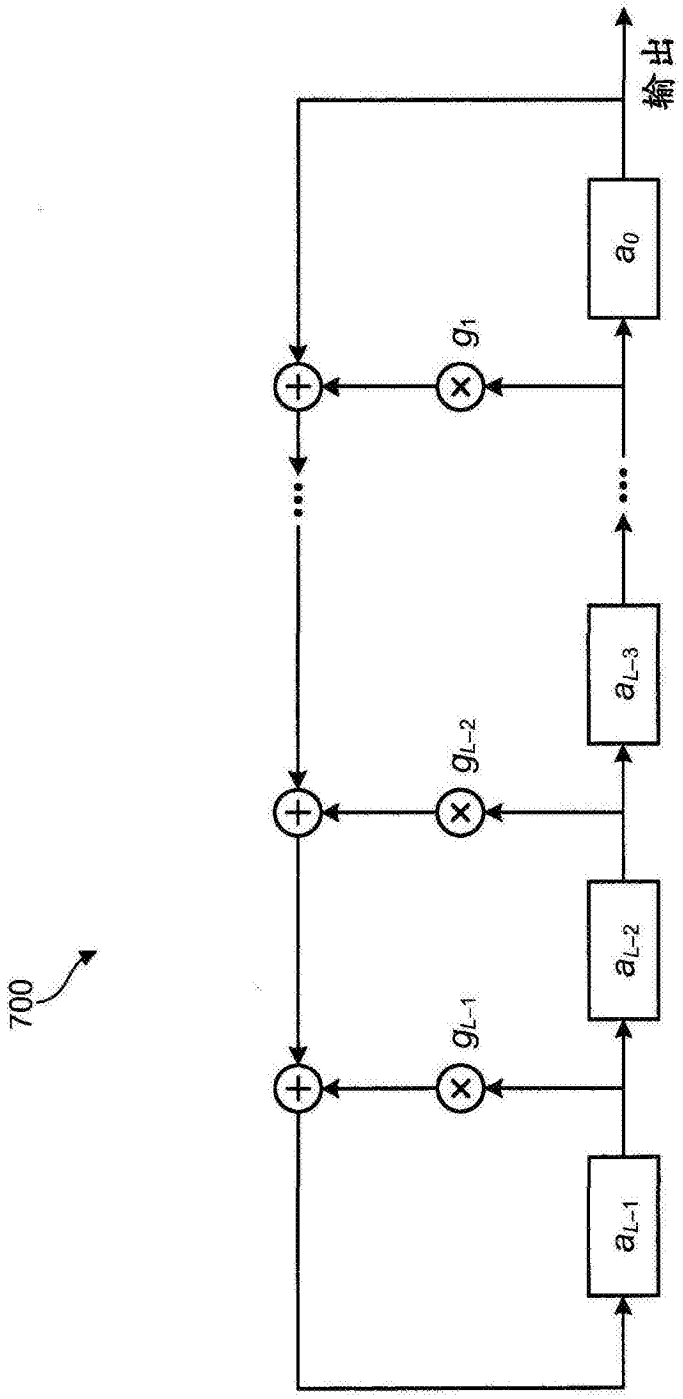


图7

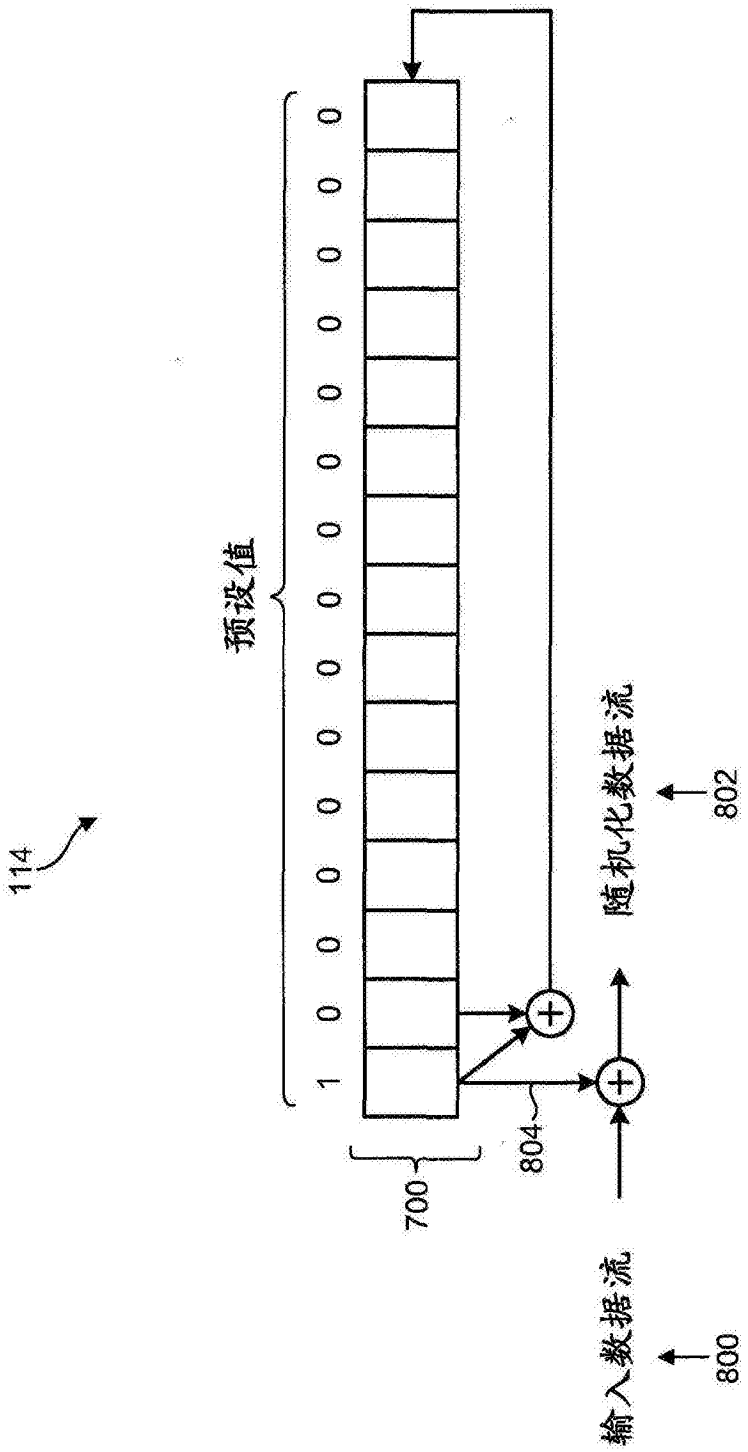


图8

轨道号	预设值
0	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
1	1 1 0 0 1 1 1 1 0 1 1 0 1 0 1
2	1 0 1 1 0 1 0 0 1 1 1 1 0 1 0
3	1 1 1 1 1 0 0 0 1 1 0 1 0 0 1
4	1 0 0 1 0 1 1 1 1 1 0 0 1 0 1
5	0 0 0 1 1 1 0 1 1 1 1 1 1 0 1
6	1 1 0 0 0 0 0 1 1 0 1 0 1 1 1
7	1 0 0 0 1 0 1 1 1 0 0 0 1 1 0
8	1 1 1 1 0 1 0 0 1 1 0 1 0 1 1
9	0 1 1 0 0 0 1 0 0 0 0 0 1 0 0
10	1 1 0 0 1 1 1 1 0 1 0 1 0 0 1
11	1 0 0 0 0 1 1 1 1 0 0 1 0 1 0
12	0 0 0 1 1 0 0 0 1 0 1 0 0 1 0
13	1 1 0 0 0 0 0 0 1 0 0 1 1 0 1
14	1 0 1 0 0 0 1 0 1 1 0 0 0 0 0
15	0 1 0 0 1 0 1 1 0 0 1 0 0 0 0
16	0 1 1 0 1 0 0 1 1 0 1 0 1 1 1
17	0 1 1 1 0 0 0 1 0 1 0 0 0 0 1
18	1 1 1 0 0 1 0 0 0 1 1 1 1 0 0
19	0 1 1 0 0 1 1 0 0 0 1 0 0 0 1
20	1 1 0 1 0 1 0 0 1 1 1 1 0 0 0
21	0 1 1 1 1 1 0 0 0 0 0 1 0 0 1
22	1 0 0 1 0 0 1 0 1 1 0 0 0 0 0
23	0 1 0 1 0 0 0 0 1 1 0 1 1 1 1
24	1 0 1 1 1 0 0 0 0 1 0 0 0 0 1
25	1 1 1 0 0 1 1 0 0 0 1 0 0 0 0
26	0 1 1 0 0 0 0 1 1 0 1 0 0 0 0
27	0 0 0 0 0 1 1 0 1 1 0 0 1 1 1
28	1 1 1 1 1 0 1 1 1 1 0 0 1 0 0
29	1 0 1 1 0 1 0 1 0 1 1 0 0 1 1
30	0 1 0 0 0 1 0 1 0 1 1 0 0 1 0
31	0 0 1 1 0 0 0 1 0 1 1 0 0 0 1

图9

轨道号	预设值
0	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
1	0 1 0 0 0 1 1 1 1 1 1 1 1 1 1
2	0 1 1 1 0 0 1 1 0 0 1 1 0 1 0
3	0 0 0 1 1 1 1 0 0 1 1 1 0 0 0
4	0 1 1 1 1 1 0 1 0 1 1 1 1 0 1
5	1 0 1 1 0 1 0 1 1 1 0 1 1 1 0
6	1 1 0 1 0 1 0 0 0 0 0 1 0 0 1
7	0 0 1 1 0 1 0 0 0 1 0 1 0 1 0
8	1 1 0 0 1 1 1 1 0 1 1 0 1 0 1
9	1 0 1 1 1 0 1 1 0 0 0 1 0 0 0
10	0 1 1 1 0 1 1 0 1 1 0 0 1 1 1
11	0 1 0 0 1 0 0 0 0 1 0 0 0 0 0
12	1 0 0 1 0 0 0 1 1 1 1 0 0 1 1
13	1 1 0 0 1 0 0 0 0 0 1 0 1 1 1
14	0 1 1 0 1 0 0 0 0 0 0 0 1 1 1
15	0 1 1 0 1 0 1 0 0 1 0 0 0 1 0
16	1 0 1 1 0 1 0 0 1 1 1 1 0 1 0
17	1 0 0 1 0 0 0 1 0 0 1 0 1 1 0
18	0 0 1 1 1 0 1 0 1 0 0 1 1 1 1
19	1 0 0 1 0 0 1 0 0 0 1 0 0 1 1
20	1 1 1 1 0 0 1 0 0 0 1 0 1 1 1
21	0 1 0 1 1 0 0 0 0 0 0 0 0 1 1
22	0 1 0 0 0 0 0 0 0 1 1 1 0 0 0
23	0 1 0 1 0 0 1 0 1 0 0 1 1 0 1
24	1 1 1 1 1 0 0 0 1 1 0 1 0 0 1
25	0 0 0 1 1 0 0 1 1 0 1 0 0 1 0
26	0 1 0 1 1 0 0 1 1 1 1 0 0 1 0
27	1 1 1 1 0 1 1 1 1 1 1 1 1 1 1
28	1 0 1 0 0 1 1 0 0 1 1 1 1 0 0
29	0 0 1 1 1 1 0 1 1 1 1 1 0 1 0
30	1 0 1 1 1 1 0 0 0 1 1 0 1 0 0
31	1 0 1 0 0 1 1 0 0 1 1 1 0 0 0

图10