



(45)授权公告日 2018.01.09

权利要求书3页 说明书12页 附图8页

1. 一种数据处理系统,所述数据处理系统包括:
数据解码系统,可操作用于:
利用第一修改算法修改对应于码字的解码器输入以得到第一修改输出;
利用第二修改算法修改对应于所述码字的所述解码器输入以得到第二修改输出,其中所述第一修改算法与所述第二修改算法不同;
由数据解码器电路对第一修改输出应用数据解码算法以得到第一解码输出;
由数据解码器电路对第二修改输出应用数据解码算法以得到第二解码输出;以及
至少部分地基于第一解码输出的第一特征和第二解码输出的第二特征选择第一解码输出和第二解码输出中的一个作为选定的解码结果。
2. 如权利要求1所述的数据处理系统,其中第一特征是对应于第一解码输出的不满意校验的第一数量,其中第二特征是对应于第二解码输出的不满意校验的第二数量,并且其中数据解码系统进一步可操作用于:
确定对应于第一解码输出的不满意校验的第一数量;以及
确定对应于第二解码输出的不满意校验的第二数量。
3. 如权利要求2所述的数据处理系统,其中选择第一解码输出和第二解码输出中的一个包括:
在不满意校验的第一数量小于不满意校验的第二数量时选择第一解码输出作为选定的解码结果;以及
在不满意校验的第二数量小于不满意校验的第一数量时选择第二解码输出作为选定的解码结果。
4. 如权利要求1所述的数据处理系统,其中数据解码系统进一步可操作用于:
由选定的解码结果引导对第一修改输出重新应用数据解码算法以得到第三解码输出;
以及
由选定的解码结果引导对第二修改输出重新应用数据解码算法以得到第四解码输出。
5. 如权利要求1所述的数据处理系统,其中解码器输入包括多个非二进制符号。
6. 如权利要求5所述的数据处理系统,其中每一个非二进制符号都包括四位。
7. 如权利要求5所述的数据处理系统,其中非二进制符号中的位被成列设置,并且其中第一修改输出内的每一个非二进制符号都包括来自连续列的位。
8. 如权利要求7所述的数据处理系统,其中第二修改输出内至少一个非二进制符号包括来自非连续列的位。
9. 如权利要求1所述的数据处理系统,其中第一修改算法是直通算法,以使第一修改输出与解码器输入相同。
10. 如权利要求9所述的数据处理系统,其中第二修改算法是轮换算法,以使第二修改输出是解码器输入的轮换版本。
11. 如权利要求1所述的数据处理系统,其中所述系统进一步包括:
数据检测器电路,可操作用于对样本集应用数据检测算法以得到检测输出,其中解码器输入源于检测输出。
12. 如权利要求11所述的数据处理系统,其中数据检测器电路选自包括Viterbi算法数据检测器电路和最大后验概率数据检测器电路的组。

13. 如权利要求1所述的数据处理系统,其中数据解码算法是低密度奇偶校验算法。

14. 如权利要求1所述的数据处理系统,其中数据处理系统被实施为从包括存储设备和接收设备的组中选出的设备的一部分。

15. 如权利要求1所述的数据处理系统,其中数据处理系统被实施为集成电路的一部分。

16. 一种数据处理方法,所述方法包括:

利用修改算法修改解码器输入以得到修改输出;

通过数据解码器电路对解码器输入应用数据解码算法以得到第一解码输出;

通过数据解码器电路对修改输出应用数据解码算法以得到第二解码输出;以及

至少部分地基于第一解码输出中不满意校验的第一数量和第二解码输出中不满意校验的第二数量选择第一解码输出和第二解码输出中的一个作为选定的解码结果。

17. 如权利要求16所述的方法,其中所述方法进一步包括:

由选定的解码结果引导对第一修改输出重新应用数据解码算法以得到第三解码输出;
以及

由选定的解码结果引导对第二修改输出重新应用数据解码算法以得到第四解码输出。

18. 如权利要求16所述的方法,所述方法进一步包括:

确定对应于第一解码输出的不满意校验的第一数量;以及

确定对应于第二解码输出的不满意校验的第二数量。

19. 如权利要求16所述的方法,其中解码器输入包括多个非二进制符号,其中非二进制符号内的位被成列设置,并且其中修改输出内的每一个非二进制符号都包括来自非连续列的位。

20. 一种存储设备,所述存储设备包括:

存储介质;

磁头组件,针对存储介质设置并且可操作用于提供与存储介质上的信息相对应的感测信号;

读通道电路,包括:

模拟前端电路,可操作用于提供与感测信号相对应的模拟信号;

模数转换器电路,可操作用于采样模拟信号以得到一系列数字样本;

均衡器电路,可操作用于均衡数字样本以得到样本集;

样本缓存,可操作用于保留样本集;

数据解码器电路,可操作用于:

利用第一修改算法修改解码器输入以得到第一修改输出,其中解码器输入源于样本集;

利用第二修改算法修改解码器输入以得到第二修改输出;

由数据解码器电路对第一修改输出应用数据解码算法以得到第一解码输出;

由数据解码器电路对第二修改输出应用数据解码算法以得到第二解码输出;

至少部分地基于第一解码输出的第一特征和第二解码输出的第二特征选择第一解码输出和第二解码输出中的一个作为选定的解码结果;

由选定的解码结果引导对第一修改输出重新应用数据解码算法以得到第三解码输出;

以及

由选定的解码结果引导对第二修改输出重新应用数据解码算法以得到第四解码输出。

用于符号重组解码处理的系统和方法

技术领域

[0001] 本发明涉及用于执行数据处理的方法和系统，并且更具体地涉及用于对解码器输入的不同轮换应用数据解码算法以作为数据处理一部分的方法和系统。

背景技术

[0002] 数据处理电路经常包括数据检测器电路和数据解码器电路。在某些情况下设有通过数据检测器电路和数据解码器电路的多条通路以尝试恢复原始写入的数据。每一次通过数据检测器电路和数据解码器电路都可能包括通过数据解码器电路的多次迭代。在某些情况下，允许通过数据解码器电路和数据检测器电路多次迭代可能无法得到正确的结果。

[0003] 因此，至少出于上述理由，在本领域内存在对用于数据处理的先进系统和方法的需求。

发明内容

[0004] 本发明涉及用于执行数据处理的方法和系统，并且更具体地涉及用于对解码器输入的不同轮换应用数据解码算法以作为数据处理一部分的方法和系统。

[0005] 本发明的各种实施例提供了包括数据解码系统的数据处理系统。数据解码系统可操作用于：利用第一修改算法修改解码器输入以得到第一修改输出；利用第二修改算法修改解码器输入以得到第二修改输出；由数据解码器电路对第一修改输出应用数据解码算法以得到第一解码输出；由数据解码器电路对第二修改输出应用数据解码算法以得到第二解码输出；然后至少部分地基于第二解码输出的第一特征和第二解码输出的第二特征选择一个第一解码输出作为选定的解码结果。在本发明的某些实施例中，数据解码算法是低密度奇偶校验算法。在某些情况下，数据处理系统被实施为存储设备或接收设备的一部分。在一种或多种情况下，数据处理系统被实施为集成电路的一部分。

[0006] 在上述实施例的某些实例中，第一特征是对应于第一解码输出的不满意校验的第一数量，而第二特征是对应于第二解码输出的不满意校验的第二数量。在某些这样的实例中，数据解码系统进一步可操作用于：确定对应于第一解码输出的不满意校验的第一数量；以及确定对应于第二解码输出的不满意校验的第二数量。在上述实例的某些情况下，选择第一解码输出和第二解码输出中的一个包括：在不满意校验的第一数量小于不满意校验的第二数量时选择第一解码输出作为选定的解码结果；以及在不满意校验的第二数量小于不满意校验的第一数量时选择第二解码输出作为选定的解码结果。

[0007] 在上述实施例的各种实例中，数据解码系统进一步可操作用于：由选定的解码结果引导对第一修改输出重新应用数据解码算法以得到第三解码输出；以及由选定的解码结果引导对第二修改输出重新应用数据解码算法以得到第四解码输出。

[0008] 在上述实施例的各种实例中，解码器输入包括多个非二进制符号。在特定情况下，每一个非二进制符号都包括四位。在各种情况下，非二进制符号中的位被成列设置，并且第一修改输出中的每一个非二进制符号都包括来自连续列的位。在一种或多种情况下，第二

修改输出中至少有一个非二进制符号包括来自非连续列的位。在上述实施例的一个或多个实例中,第一修改算法是直通算法以使第一修改输出与解码器输入相同。在某些这样的实例中,第二修改算法是轮换算法以使第二修改输出是解码器输入的轮换版本。

[0009] 本发明的另一些实施例提供的方法包括:利用修改算法修改解码器输入以得到修改输出;通过数据解码器电路对解码器输入应用数据解码算法以得到第一解码输出;通过数据解码器电路对修改输出应用数据解码算法以得到第二解码输出;然后至少部分地基于第一解码输出中不满意校验的第一数量和第二解码输出中不满意校验的第二数量选择一个第一解码输出作为选定的解码结果。在上述实施例的某些实例中,所述方法进一步包括:由选定的解码结果引导对第一修改输出重新应用数据解码算法以得到第三解码输出;以及由选定的解码结果引导对第二修改输出重新应用数据解码算法以得到第四解码输出。在上述实施例的各种实例中,所述方法进一步包括:确定对应于第一解码输出的不满意校验的第一数量;以及确定对应于第二解码输出的不满意校验的第二数量。在上述实施例的某些特定实例中,解码器输入包括多个非二进制符号,并且非二进制符号中的位被成列设置。在某些这样的实例中,修改输出中的每一个非二进制符号都包括来自非连续列的位。

[0010] 该发明内容部分仅提供了本发明某些实施例的概述。本发明多种其他的目标、特征、优点和另外的实施例将根据以下的详细说明、所附权利要求和附图而变得更加显而易见。

附图说明

[0011] 通过参照在说明书下余部分介绍的附图即可实现对本发明各种实施例更进一步的理解。在附图中,相似的附图标记在若干附图中始终被用于表示类似的组件。在某些实例中,小写字母构成的下标与附图标记相关联以表示多个类似的组件之一。在引用未注明存在下标的附图标记进行说明时,应该认为是指代所有这些多个类似的组件。

[0012] 图1示出了根据本发明的某些实施例的具有解码器输入轮换电路的数据处理电路;

[0013] 图2-3示出了可以针对本发明的各种实施例使用的解码器输入的两种不同轮换的示例;

[0014] 图4a-4c是示出了根据本发明的某些实施例的用于轮换输入解码处理的方法流程图;

[0015] 图5示出了根据本发明的一个或多个实施例的一种存储设备,其中包括具有解码器输入轮换电路的读通道;以及

[0016] 图6示出了根据本发明的某些实施例的一种数据传输设备,其中包括具有解码器输入轮换电路的接收器。

具体实施方式

[0017] 本发明涉及用于执行数据处理的系统和方法,并且更具体地涉及用于对解码器输入的不同轮换应用数据解码算法以作为数据处理一部分的系统和方法。

[0018] 本发明的各种实施例提供用于在需要延迟处理时修改数据解码处理。这样的延迟处理在数据处理电路的处理要求有所减少时进行,由此允许应用否则就不使用的资源以用

于先前非收敛的数据集。数据处理电路处理要求的这种减少可以例如由于在从存储介质访问数据时改变磁道或者数据传输设备的传输结束导致引入数据解码过程的数据量减少而产生。根据本文中提供的公开内容,本领域普通技术人员应该意识到可以有多种情况导致数据处理电路的不充分使用。还应该注意到在本申请中讨论的修改解码过程也可以适用于未涉及不充分使用的情况。

[0019] 在本发明的特定实施例中,数据处理被应用于指定的数据集,直到处理收敛(也就是获得正确结果)或满足超时条件为止。在满足超时条件时,数据集可以被保留,直到有另外的处理时间和/或资源可用的随后时间处。该另外的处理时间可以是但不局限于磁道的改变。一旦有另外的处理时间可用,数据解码算法即被应用于第一轮换的解码器输入以得到第一解码输出。数据解码算法还被应用于第二轮换的解码器输入以得到第二解码输出。作为一个示例,第一轮换的解码器输入可以是非二进制解码器输入,其中每一个非二进制符号都包括由以下公式描述的元素:

[0020] $\text{symbol}_i = \{\text{Element}_{\text{row}+i, \text{column}}, \text{Element}_{\text{row}+i+1, \text{column}}, \text{Element}_{\text{row}+i, \text{column}+1}, \text{Element}_{\text{row}+i+1, \text{column}+1}\}$ 其中row和column表示符号中的元素相对于在其中设置码字的阵列的位置。类似地,第二轮换的解码器输入是与以上所述相同的解码器输入,只是分组有所不同。例如,第二轮换的解码器输入可以是非二进制解码器输入,其中每一个非二进制符号都包括由以下公式描述的元素:

[0021] $\text{Symbol}_i = \{\text{Element}_{\text{row}+i+1, \text{column}}, \text{Element}_{\text{row}+i+2, \text{column}}, \text{Element}_{\text{row}+i+1, \text{column}+1}, \text{Element}_{\text{row}+i+2, \text{column}+1}\}$ 应该注意的是尽管上述公式表示的是四位符号,但是也可以将其他的位数用于符号尺寸。将第一解码器输出中剩余的不满意校验数量与第二解码器输出中剩余的不满意校验数量进行比较。当第二解码器输出中剩余的不满意校验数量大于第一解码器输出中剩余的不满意校验数量时,选择第一解码器输出用于后续处理。否则就选择第二解码器输出。如本文中所用的术语“轮换”被用于广义地表示数据集的任何重排。作为一个示例,这样的轮换包括重组符号。在一种特定的情况下,使用两种不同的符号分组:一种是使用跨连续列的标准分组,而另一种是其中的第一符号包括来自一列的若干位和来自非连续符号的若干位。根据本文中提供的公开内容,本领域普通技术人员应该意识到有多种修改均可被认为是本申请范围内的轮换。

[0022] 转至图1,示出了根据本发明的某些实施例的具有解码器输入轮换电路的数据处理电路100。数据处理电路100包括接收模拟输入108的模拟前端电路110。模拟前端电路110处理模拟输入108并将经过处理的模拟信号112提供给模数转换器电路115。模拟前端电路110可以包括但不限于本领域已知的模拟滤波器和放大器电路。根据本文中提供的公开内容,本领域普通技术人员应该意识到可以包括多种电路作为模拟前端电路110的一部分。在某些情况下,模拟输入108源自相对于存储介质(未示出)设置的读/写磁头组件(未示出)。在另一些情况下,模拟输入108源自可操作用于从传输介质(未示出)接收信号的接收器电路(未示出)。传输介质可以是有线或无线的。根据本文中提供的公开内容,本领域普通技术人员应该意识到可以有多种得到模拟输入308的来源。

[0023] 模数转换器电路115将经过处理的模拟信号112转换为对应的一系列数字样本117。模数转换器电路115可以是本领域已知的能够生成与模拟输入信号相对应的数字样本的任何电路。根据本文中提供的公开内容,本领域普通技术人员应该意识到对于本发明的

不同实施例可以使用多种模数转换器电路。数字样本117被提供给均衡器电路120。均衡器电路120对数字样本117应用均衡算法以得到均衡输出122。在本发明的某些实施例中,均衡器电路120是本领域已知的数字有限脉冲响应滤波器电路。

[0024] 均衡输出122被提供给数据检测器电路125和样本缓存电路175。样本缓存电路175将均衡输出122存储为缓存数据177以供在通过数据检测器电路125进行后续迭代时使用。数据检测器电路125可以是本领域已知的能够生成检测输出127的任何数据检测器电路。作为一些示例,数据检测器电路125可以是但不局限于本领域已知的Viterbi算法检测器电路或最大后验概率检测器电路。值得注意的是通用短语“Viterbi数据检测算法”或“Viterbi算法数据检测器电路”被用于广义地表示Viterbi检测算法或Viterbi算法检测器电路或其变形,其中包括但不限于双向Viterbi检测算法或双向Viterbi算法检测器电路。同样地,通用短语“最大后验概率数据检测算法”或“最大后验概率数据检测器电路”被用于广义地表示任何最大后验概率检测算法或检测器电路或其变形,其中包括但不限于简化的最大后验概率数据检测算法和最大对数后验概率数据检测算法或对应的检测器电路。根据本文中提供的公开内容,本领域普通技术人员应该意识到对于本发明的不同实施例可以使用多种数据检测器电路。检测输出127可以包括硬决策和软决策。术语“硬决策”和“软决策”均以其广义使用。具体地,“硬决策”是表示预期原始输入值(例如二进制的“1”或“0”,或者非二进制的数值)的输出,而“软决策”则表示对应硬决策正确的可能性。根据本文中提供的公开内容,本领域普通技术人员应该意识到对于本发明的不同实施例可以使用多种硬决策和软决策。

[0025] 检测输出127被提供给操作于缓存在数据检测器电路125和数据解码系统189之间传输的数据的中央队列存储器电路160。在某些情况下,中央队列存储器电路160包括本领域已知的交错(也就是数据重排)电路和反交错(也就是数据反重排)电路。当数据解码系统189可用时,数据解码系统189通过中央队列存储器电路160访问检测输出127以作为解码器输入156。数据解码系统189向分组电路194和分组电路197提供解码器输入156。

[0026] 分组电路194提供第一轮换解码器输入183,其是解码器输入156的第一轮换。在本发明的一个特定实施例中,第一轮换是不轮换。在此情况下,分组电路194是提供解码器输入156作为第一轮换解码器输入183的直通电路。图2a-2c中介绍了未轮换的解码器输入的一个示例,其中根据以下的公式描述送往数据解码器电路的符号:

[0027] $\text{Symbol}_i = \{\text{Element}_{\text{row}+i, \text{column}}, \text{Element}_{\text{row}+i+1, \text{column}}, \text{Element}_{\text{row}+i, \text{column}+1}, \text{Element}_{\text{row}+i+1, \text{column}+1}\}$ 其中row和column表示符号中的元素相对于在其中设置码字的阵列的位置。同样地,应该注意的是尽管上述公式表示的是四位符号,但是也可以将其他的位数用于符号尺寸。这种未轮换的解码器输入的特征在于一行内的第一符号包括来自连续列的元素。

[0028] 分组电路197提供第二轮换解码器输入181,其构成解码器输入156的第二轮换。在本发明的一个特定实施例中,第二轮换是移位轮换(offset rotation)。图3a-3d中介绍了具有这种移位轮换的解码器输入的一个示例,其中根据以下的公式描述送往数据解码器电路的符号:

[0029] $\text{Symbol}_i = \{\text{Element}_{\text{row}+i+1, \text{column}}, \text{Element}_{\text{row}+i+2, \text{column}}, \text{Element}_{\text{row}+i+1, \text{column}+1}, \text{Element}_{\text{row}+i+2, \text{column}+1}\}$ 同样地,应该注意的是尽管上述公式表示的是四位符号,但是也可以

将其他的位数用于符号尺寸。这种轮换解码器输入的特征在于一行内的第一符号包括来自由轮换或移位得出的非连续列的元素。这种移位可跨越指定行内的后续符号继续。

[0030] 在通过使延迟处理信号195无效而指明的标准处理期间,选组器电路198选择第一轮换解码器输入183作为解码器输入187。数据处理控制器电路190可操作用于只要是延迟处理输入192无效就使延迟处理信号195无效,以及在延迟处理输入192有效时使延迟处理信号195有效。

[0031] 数据解码器电路196可操作用于对解码器输入187应用数据解码算法以得到解码输出152。类似于检测输出127,解码输出152可以包括硬决策和软决策。例如,数据解码器电路196可以是本领域已知的能够对接收的输入应用解码算法的任何数据解码器电路。数据解码器电路196可以是但不局限于本领域已知的低密度奇偶校验(LDPC)解码器电路或Reed Solomon解码器电路。根据本文中提供的公开内容,本领域普通技术人员应该意识到对于本发明的不同实施例可以使用多种数据解码器电路。在原始数据得以恢复(也就是数据解码算法收敛)时或者在出现超时状态时,解码输出152被存储到硬决策输出电路180内包含的存储器中。相应地,硬决策输出电路180将收敛的解码输出152作为数据输出184提供给接收者(未示出)。接收者例如可以是能够操作用于接收经过处理的数据集的接口电路。根据本文中提供的公开内容,本领域普通技术人员应该意识到对于本发明的不同实施例可以使用多种接收者。在原始数据未能在超时状态之前恢复(也就是数据解码算法无法收敛)时,解码输出152如以下更加详细介绍的那样指明数据不可用,并且数据输出184被类似地标注为不可用。

[0032] 通过数据检测器电路125和数据解码器电路196的组合可以进行一次或多次迭代以努力使原始写入的数据集收敛。通过数据检测器电路和数据解码器电路进行的处理因此被称为“全局迭代”。相比之下,每一次通过数据解码器电路都被称为“局部迭代”。对于第一次全局迭代,数据检测器电路125无需来自解码输出的引导就向均衡输出122应用数据检测算法。对于随后的全局迭代,数据检测器电路125在解码输出152的引导下向缓存数据177应用数据检测算法。解码输出152被存储至中央队列存储器电路160作为解码器输出154,并且由中央队列存储器电路160提供作为检测器输入129。

[0033] 在每一次全局迭代期间,数据解码器电路196均可进行包括对解码器输入156应用数据解码算法的一次或多次局部迭代。对于第一次局部迭代,数据解码器电路196无需来自解码输出152的引导就应用数据解码器算法。对于随后的局部迭代,数据解码器电路196在先前解码输出152的引导下向解码器输入156应用数据解码算法。允许的局部迭代次数例如可以是十次。根据本文中提供的公开内容,本领域普通技术人员应该意识到对于本发明的不同实施例可以允许有多种不同的局部迭代次数。在通过数据解码器电路196的局部迭代次数超过允许次数但是确定在数据集的标准处理期间仍允许至少一次另外的全局迭代时,解码输出152被送回中央队列存储器电路160作为解码输出154。解码输出154被保留在中央队列存储器电路160内,直到数据检测器电路125变为可用于执行另外的处理为止。

[0034] 相比之下,在通过数据解码器电路196的局部迭代次数超过允许次数并且确定已经超过用于数据集的全局迭代可允许次数和/或超时或存储器的使用要求终止处理特定数据集时,数据集的标准处理即告结束,同时解码输出被提供作为数据输出152并指明该输出不可用。在数据集的标准处理已经结束的这种情况下,未收敛(也就是提供作为数据输出

152)的数据集被存储在硬决策输出电路180的存储器内,并且对应于未收敛数据集(也就是样本缓存电路175内保留)的样本数据被保留在样本缓存电路175内至少达到规定的时段,以等待通过使延迟处理输入192有效而做出的延迟处理指示。另外,未收敛的数据集和对应的样本数据根据保持信号191有效时的指示而被识别用于延迟处理。同样地,通过使延迟处理输入192有效而发出信号进行延迟处理。

[0035] 在根据延迟处理输入192有效时的指示进行延迟处理期间,根据来自数据处理控制器电路190的控制输出176的指示而访问与保留在样本缓存电路175内的先前未收敛数据集相对应的其中一个样本数据集。另外,数据处理控制器电路190使延迟处理信号195对数据解码系统189和硬决策输入电路180有效。数据检测器电路125对识别用于延迟处理且根据控制输出176的指示访问的样本集重新应用数据检测算法以得到检测输出127。检测输出127被存储至中央队列存储器电路160。

[0036] 一旦数据解码系统189变为可用,数据解码系统189就通过中央队列存储器电路160访问检测输出127以作为解码器输入156。另外,数据解码系统189还访问先前存储到硬决策输出电路180的存储器中的对应解码输出182。在第一次通过时,选组器电路198选择第一轮换解码器输入183作为解码器输入187,并且数据解码器电路196对解码器输入187应用数据解码算法以得到第一解码输出,第一解码输出被存储回中央存储器队列160以作为解码器输入154。在第二次通过时,选组器电路198选择第二轮换解码器输入181作为解码器输入187,并且数据解码器电路196对解码器输入187应用数据解码算法以得到第二解码输出,第二解码输出被存储回中央存储器队列160以作为解码器输入154。

[0037] 数据解码系统189随后确定第一解码输出中不满意校验的第一数量和第二解码输出中不满意校验的第二数量。数据解码系统189随后比较不满意校验的第一数量和不满意校验的第二数量。当跟未轮换的解码器输入183相关联的不满意校验的第一数量小于跟轮换的解码器输入181相关联的不满意校验的第二数量时,保留第一解码输出以供引导未来的数据检测和/或数据解码使用。否则,保留第二解码输出以供引导未来的数据检测和/或数据解码使用。

[0038] 在允许通过数据解码器电路196进行另一次局部迭代时,对未轮换的解码器输入183和轮换的解码器输入181应用数据解码算法的上述过程就由先前保留的数据解码结果引导。重复该过程,直到第一解码输出或第二解码输出之一收敛、针对当前的全局迭代完成最大次数的局部迭代或者满足超时条件为止。在满足超时条件时报告出错。出错的条件可以在用完最大次数的全局迭代时得到满足。根据本文中提供的公开内容,本领域普通技术人员应该意识到有多种其他的情况均可被认为是满足了超时条件。在已经完成了最大次数的局部迭代后,由保留的结果引导对来自样本缓存电路175的数据输入应用数据检测算法,并且解码过程再次开始。在第一解码输出或第二解码输出收敛时,提供收敛结果以作为数据输出。

[0039] 转至图2a-2c,其中示出了可以针对本发明的各种实施例使用的未轮换的解码器输入示例。图2a以图表示出了对应于源自检测输出的示例性数据集的阵列200。如图所示,阵列200是拥有十(10)行和十八(18)列的二维阵列。如图所示,在表示未轮换的解码器输入的阵列210中,阵列200中表示的数据集被组织为由列号(SymA-SymI)和行号(Sym1-Sym5)表示的四位符号。每一个四位符号都根据以下的公式描述:

[0040] $\text{Symbol}_i = \{\text{Element}_{\text{row}+i, \text{column}}, \text{Element}_{\text{row}+i+1, \text{column}}, \text{Element}_{\text{row}+i, \text{column}+1}, \text{Element}_{\text{row}+i+1, \text{column}+1}\}$ 其中row和column表示符号中的元素相对于在其中设置码字的阵列的位置。应该注意的是尽管上述公式和附图涉及的是四位符号,但是也可以将其他的位数用于符号尺寸。这种未轮换的解码器输入的特征在于一行内的第一符号包括来自连续列(例如标记为SymB的列A,B)的元素。这种排列可以跨行继续,其中下一个符号从下一个连续列中得出(例如标记为SymB的列C,D;标记为SymC的列E,F;标记为SymD的列G,H;标记为SymE的列I,J;标记为SymF的列K,L;标记为SymG的列M,N;标记为SymH的列O,P;以及标记为SymI的列Q,R)。这些符号如图2c所示作为解码器输入220被串行地提供给数据解码器电路。

[0041] 转至图3a-3d,其中示出了可以针对本发明的各种实施例使用的轮换的解码器输入示例。图3a以图表示出了与源自与上述在图2a中所示相同的检测输出的示例性数据集相对应的阵列200。同样地,阵列200是拥有十(10)行和十八(18)列的二维阵列。如图所示,在表示轮换的解码器输入的阵列310中,阵列200中表示的数据集被组织为由列号(SymA-SymI和X)以及行号(Sym1-Sym5)表示的四位符号。每一个四位符号都根据以下的公式描述:

[0042] $\text{Symbol}_i = \{\text{Element}_{\text{row}+i+1, \text{column}}, \text{Element}_{\text{row}+i+2, \text{column}}, \text{Element}_{\text{row}+i+1, \text{column}+1}, \text{Element}_{\text{row}+i+2, \text{column}+1}\}$

[0043] 应该注意的是尽管上述公式表示的是四位符号,但是也可以将其他的位数用于符号尺寸。这种轮换的解码器输入的特征在于一行内的第一符号包括来自非连续列(例如标记为SymA的列R,A)的元素。具体地,标注为X的最后一列被用作由箭头315表示的第一列,并且所有其他的列都向右移动一位。这种排列可以跨行继续,其中下一个符号从下一个连续列中得出(例如标记为SymB的列B,C;标记为SymC的列D,E;标记为SymD的列F,G;标记为SymE的列H,I;标记为SymF的列J,K;标记为SymG的列L,M;标记为SymH的列N,O;以及标记为SymI的列P,Q)。图3c示出了将最后一列轮换至SymA中包含的第一列的数据集320。如数据集320内组合而成的符号就表示轮换的解码器输入。数据集320内的符号如图3d所示作为解码器输入330被串行地提供给数据解码器电路。

[0044] 转至图4a-4c,流程图400,445,470示出了根据本发明的某些实施例的用于双路二进制和非二进制的解码过程的方法。转至图4a和所附的流程图400,接收模拟输入(模块405)。模拟输入可以源于例如存储介质或数据传输通道。根据本文中提供的公开内容,本领域普通技术人员应该能够想到多种模拟输入的来源。将模拟输入转换为一系列数字样本(模块410)。这种转换可以利用本领域已知的模数转换器电路或系统完成。应该注意的是本领域已知的能够将模拟信号转换为表示所接收模拟信号的一系列数字值的任何电路均可使用。将得到的数字样本均衡化以得到均衡输出(模块415)。在本发明的某些实施例中,均衡化可利用本领域已知的数字有限脉冲响应电路完成。根据本文中提供的公开内容,本领域普通技术人员应该意识到根据本发明的不同实施例可以使用多种均衡器电路代替这样的数字有限脉冲响应电路来执行均衡化。缓存均衡输出(模块420)。

[0045] 判定是否需要延迟处理(模块425)。在数据处理电路的处理要求有所减少时选择这样的延迟处理。处理要求的这种减少例如可以因为在读取存储介质时改变磁道或者因为数据传输设备的传输结束而产生。根据本文中提供的公开内容,本领域普通技术人员应该意识到可以有多种情况导致数据处理电路的不充分使用。

[0046] 在不期望延迟处理时(模块425),应用标准处理。该标准处理包括从缓存中选择下

一个均衡输出用于处理 (模块430)。这种选择可以根据本领域已知的任何数据处理电路选择算法完成。对选定的均衡输出应用数据检测算法以得到检测输出 (模块435), 并且将检测输出 (或其导出值) 存储至中央存储器电路 (模块440)。随后即可从中央存储器访问该存储的数据以用于执行标准处理 (流程图445)。该标准处理应根据图4b中的流程图445来完成。

[0047] 在需要延迟处理时 (模块425), 应用延迟处理。该延迟处理包括从缓存中选择下一个延迟处理均衡输出用于处理 (模块450)。延迟处理均衡输出选自在标准处理期间无法收敛的一个或多个数据集。将可用于当前处理数据集的剩余全局迭代次数设定为等于延迟全局最大值 (模块480)。在本发明的某些实施例中, 延迟全局最大值为二十 (20)。对选定的延迟处理均衡输出应用数据检测算法以得到检测输出 (模块485), 并且将检测输出 (或根据其导出的数据集) 存储至中央存储器电路 (模块490)。在某些情况下, 存储的检测输出导出值是检测输出的交错或重排版本。随后即可从中央存储器访问这些存储的数据以用于执行延迟处理 (图4c中的流程图495)。

[0048] 转至图4b, 流程图445示出了上述标准处理的实现。根据流程图445, 判定解码器电路是否可用于处理先前存储的检测输出 (模块401)。在解码器电路可用时 (模块401), 选择下一个检测输出的导出值用于处理并从中央存储器电路中访问 (模块406)。通过数据解码器电路对选定的检测输出应用数据解码算法的第一次局部迭代以得到解码输出 (模块411)。在本发明的某些实施例中, 选定的检测输出被无任何轮换地处理。图2a-2c中介绍了未轮换的解码器输入的一个示例, 其中根据以下的公式描述送往数据解码器电路的符号:

[0049] $\text{Symbol}_i = \{\text{Element}_{\text{row}+i, \text{column}}, \text{Element}_{\text{row}+i+1, \text{column}}, \text{Element}_{\text{row}+i, \text{column}+1}, \text{Element}_{\text{row}+i+1, \text{column}+1}\}$ 其中row和column表示符号中的元素相对于在其中设置码字的阵列的位置。同样地, 应该注意的是尽管上述公式表示的是四位符号, 但是也可以将其他的位数用于符号尺寸。这种未轮换的解码器输入的特征在于一行内的第一符号包括来自连续列的元素。

[0050] 然后判定解码输出是否收敛 (也就是得到了正确结果) (模块416)。在解码输出收敛时 (模块416), 将解码输出提供给硬决策输出缓存 (模块421)。然后判定硬决策输出缓存是否已准备好进行卸载 (模块456)。在某些情况下, 硬决策输出缓存在最近完成的解码输出正是在先前提供作为数据输出之后的下一个解码输出时就准备好进行卸载。当硬决策输出缓存准备好进行卸载时 (模块456), 将硬决策输出缓存中保留的所有连续解码输出作为数据输出提供给接收设备 (模块461)。根据本文中提供的公开内容, 本领域普通技术人员应该意识到对于本发明的不同实施例可以使用多种接收设备。

[0051] 可选地, 在非二进制的解码输出无法收敛时 (模块416), 判定局部迭代次数是否已超过局部迭代限制 (模块426)。该局部迭代限制例如可以是十 (10) 次局部迭代。在局部迭代次数尚未超过时 (模块426), 由先前生成的解码输出引导对用于后续局部迭代的当前处理数据集重新应用数据解码算法以得到更新的解码输出 (模块431)。随后重复开始于模块416的上述过程。

[0052] 可选地, 在用于当前进行的全局迭代的局部迭代次数已经超限时 (模块426), 判定是否已经对当前处理的数据集执行了最大次数的全局迭代 (模块436)。全局迭代的次数例如可以在出现超时状态或者已经超过存储器使用限制时完成。在全局迭代尚未完成时 (模块436), 将解码输出存储至中央存储器作为标准输出, 在该情况下其等待在后续的全局迭

代中的处理(模块441)。可选地,在全局迭代已完成(模块436)时,标识当前处理的数据集以用于延迟处理(也就是在延迟处理时段期间的处理)(模块446),并且提供非二进制的解码输出以作为标记成不可用的数据输出(模块451)。

[0053] 转至图4c,流程图495示出了上述延迟处理的实现。根据流程图495,判定解码器电路是否可用于处理先前存储的检测输出(模块402)。在解码器电路可用时(模块402),选择下一个生成用于延迟处理(也就是在模块446中标识用于延迟处理)的检测输出导出值以用于处理并从中央存储器电路中访问(模块407)。根据第一分组将访问的检测输出导出值分组以得出第一分组的解码器输入(模块412)。在本发明的某些实施例中,第一分组是未轮换的分组。图2a-2c中介绍了未轮换的解码器输入的一个示例,其中根据以下的公式描述送往数据解码器电路的符号:

[0054] $Symbol_i = \{Element_{row+i,column}, Element_{row+i+1,column}, Element_{row+i,column+1}, Element_{row+i+1,column+1}\}$ 其中row和column表示符号中的元素相对于在其中设置码字的阵列的位置。同样地,应该注意的是尽管上述公式表示的是四位符号,但是也可以将其他的位数用于符号尺寸。这种未轮换的解码器输入的特征在于一行内的第一符号包括来自连续列的元素。随后由先前选定的解码结果引导对第一分组解码器输入应用数据解码算法,以得到第一解码输出(模块417)。对于延迟处理的第一次局部迭代,先前选定的解码结果是来自模块451的输出缓存中存储的解码输出。对于随后的局部迭代,先前选定的解码结果是从根据第一分组轮换的解码器输入得出的解码输出(模块482)或根据第二分组轮换的解码器输入得出的解码输出(模块477)中选定的一个。

[0055] 判定第一解码输出是否收敛(也就是得到了正确结果)(模块422)。在第一解码输出收敛时(模块422),第一解码输出被提供作为数据输出并且对后续的数据集开始运行另一次全局迭代(模块432)。否则,在第一解码输出无法收敛时(模块422),存储第一解码输出以供将来使用(模块427)。第一解码输出例如可以被存储在输出缓存、中央存储器缓存或输入缓存的未使用部分内。另外,确定第一解码输出中不满意校验(例如仍未满足的奇偶校验方程)的数量(模块437)。这可以通过计量第一解码输出中仍未满足的奇偶校验方程的数量来完成。

[0056] 接下来,根据第二分组将访问的检测输出导出值分组以得出第二分组的解码器输入(模块442)。在本发明的某些实施例中,第二分组是经过轮换的分组。图3a-3d中介绍了具有轮换的解码器输入的一个示例,其中根据以下的公式描述送往数据解码器电路的符号:

[0057] $Symbol_i = \{Element_{row+i+1,column}, Element_{row+i+2,column}, Element_{row+i+1,column+1}, Element_{row+i+2,column+1}\}$ 同样地,应该注意的是尽管上述公式表示的是四位符号,但是也可以将其他的位数用于符号尺寸。这种轮换解码器输入的特征在于一行内的第一符号包括来自自由轮换或移位得出的非连续列的元素。这种移位可跨越指定行内的后续符号继续。随后由先前选定的解码结果引导对第二分组解码器输入应用数据解码算法以得到第二解码输出(模块447)。对于延迟处理的第二次局部迭代,先前选定的解码结果是来自模块451的输出缓存中存储的解码输出。对于随后的局部迭代,先前选定的解码结果是从根据第一分组轮换的解码器输入得出的解码输出(模块482)或根据第二分组轮换的解码器输入得出的解码输出(模块477)中选定的一个。

[0058] 判定第二解码输出是否收敛(也就是得到了正确结果)(模块452)。在第二解码输

出收敛时(模块452),第二解码输出被提供作为数据输出并且对后续的数据集开始运行另一次全局迭代(模块462)。否则,在第二解码输出无法收敛时(模块452),第二解码输出被存储起来以供将来使用(模块457)。第二解码输出例如可以被存储在输出缓存、中央存储器缓存或输入缓存的未使用部分内。另外,确定第二解码输出中不满意校验(例如仍未满足的奇偶校验方程)的数量(模块437)。这可以通过计量第二解码输出中仍未满足的奇偶校验方程的数量来完成。

[0059] 判定不满意校验的第二数量是否大于不满意校验的第一数量(模块472)。在不满意校验的第二数量大于不满意校验的第一数量时(模块472),选择来自模块417的第一解码输出作为选定的解码结果(模块482)。否则,在不满意校验的第二数量不大于不满意校验的第一数量时(模块472),选择来自模块447的第二解码输出作为选定的解码结果(模块477)。

[0060] 然后判定是否允许另一次局部迭代(模块487)。在允许另一次局部迭代时(模块487)的情况下,利用新选定的解码结果重复开始于模块412的过程。否则,在不允许另一次局部迭代时(模块487)的情况下,判定是否允许另一次全局迭代(模块492)。在不允许另一次全局迭代时(模块492)的情况下,报错并开始处理后续的数据集(模块499)。否则,在允许另一次全局迭代时(模块492),由选定的解码结果引导对选定的延迟处理均衡输出应用数据检测算法,以得到更新的检测输出(模块497),并且针对相同的数据集重新启动开始于模块402的过程。

[0061] 应该注意的是以上应用中讨论的各种模块可以在集成电路中与其他功能一起实施。这样的集成电路可以包括指定模块、系统或电路或者仅仅所述模块、系统或电路的子集的所有功能。而且,模块、系统或电路中的元素可以跨越多个集成电路实施。这样的集成电路可以是本领域已知的任意类型的集成电路,包括但不限于单片集成电路、倒装芯片集成电路、多芯片模块集成电路和/或混合信号集成电路。还应该注意的是本文中所述模块、系统或电路的各种功能可以用软件或固件实施。在某些这样的情况下,整个系统、模块或电路可以利用其等价的软件或固件实施。在另一些情况下,指定系统、模块或电路的一部分可以用软件或固件实施,而其他部分则用硬件实施。

[0062] 转至图5,示出了根据本发明的某些实施例的一种存储系统500,其中包括具有解码器输入轮换电路的读通道电路510。存储系统500例如可以是硬盘驱动器。存储系统500还包括前置放大器570、接口控制器520、硬盘控制器566、电机控制器568、主轴电机572、盘片578和读/写磁头576。接口控制器520控制盘片578输入输出数据的寻址和定时。盘片578上的数据由在读/写磁头组件576准确定位到盘片578上时可通过读/写磁头组件检测到的多组磁信号构成。在一个实施例中,盘片578包括根据纵向或垂直记录方案记录的磁信号。

[0063] 在典型的读操作中,读/写磁头组件576由电机控制器568准确定位在盘片578的期望数据磁道上。电机控制器568通过在硬盘控制器566的引导下将读/写磁头组件移动至盘片578上的正确数据磁道来相对于盘片578定位读/写磁头组件576并驱动主轴电机572。主轴电机572以预定转速(RPM)旋转盘片578。在通过主轴电机572旋转盘片578时,一旦读/写磁头组件578已定位在正确的数据磁道附近,表示盘片576上数据的磁信号即可由读/写磁头组件578测量。测量的磁信号被提供作为表示盘片578上磁性数据的连续微弱模拟信号。该微弱模拟信号从读/写磁头组件576经前置放大器570传输至读通道电路510。前置放大器570可操作用于放大从盘片578得到的微弱模拟信号。接下来,读通道电路510解码并数字化

接收的模拟信号以重建最初写入盘片578的信息。该数据作为读取数据503被提供给接收电路。写操作基本上与以上的读操作相反,并且写入数据501被提供给读通道电路510。这些数据随后被解码并写入盘片578。

[0064] 在操作期间,从盘片578感测数据并利用标准处理进行处理。这样的标准处理可以利用编码器输入的一种轮换完成。在某些情况下,利用标准处理来处理的一个或多个数据集无法收敛。在这样的情况下,未收敛的数据集被提供作为指示未收敛数据集不可用的输出,并且未收敛数据集以及对应的未检测/编码数据集被保留在数据处理系统内以用于在延迟处理时段期间进行另外的处理。该延迟处理时段例如可以是在读/写磁头组件576从当前磁道移向另一条磁道时的时间段。在该延迟处理时段期间,访问未收敛数据集和对应的未检测/编码数据集的组合以供重新处理。重新处理包括对原始轮换的解码器输入执行数据解码以得到第一解码输出,以及对另一种轮换的解码器输入执行数据解码以得到第二解码输出。将第一解码器输出中剩余的不满意校验数量与第二解码器输出中剩余的不满意校验数量进行比较。当第二解码器输出中剩余的不满意校验数量大于第一解码器输出中剩余的不满意校验数量时,选择第一解码器输出用于后续处理。否则就选择第二解码器输出。在本发明的某些实施例中,可以使用类似于以上针对图1介绍的数据处理电路,和/或可以完成类似于以上针对图4a-4c介绍的处理。

[0065] 应该注意的是存储系统500可以集成到更大的存储系统例如基于RAID(廉价磁盘冗余阵列或独立磁盘冗余阵列)的存储系统内。这种将多块磁盘组合为一个逻辑单元的RAID存储系统通过冗余性来增强稳定性和可靠性。数据可以根据多种算法分布在RAID存储系统内包括的多块磁盘上并由操作系统像单块磁盘一样地进行访问。例如,数据可以镜像至RAID存储系统内的多块磁盘,或者可以用多种技术切分或分配到多块磁盘上。如果RAID存储系统内有少量的磁盘故障或变得不可用,那么可以使用纠错技术根据来自RAID存储系统内其他磁盘的其余数据部分重建丢失的数据。RAID存储系统内的磁盘可以是但不限于单独的存储系统例如存储系统500,并且可以设置为彼此靠近或者为了增加安全性而更加广泛地分布。在写操作中,写入数据被提供给控制器,其通过例如镜像或拆分写入数据将写入数据跨磁盘存储。在读操作中,控制器从磁盘中检索数据。控制器随后就像RAID存储系统是单块磁盘一样地生成得到的读取数据。

[0066] 针对读通道电路510使用的数据解码器电路可以是但不局限于本领域已知的低密度奇偶校验(LDPC)解码器电路。这样的低密度奇偶校验技术可应用于在任何虚拟通道内传输信息或者在任何虚拟介质内存储信息。传输应用包括但不限于光纤、射频通道、有线或无线局域网、数字用户专线技术、无线蜂窝网络、任何介质例如铜缆或光纤上的以太网、有线信道例如有线电势以及地球卫星通信。存储应用包括但不限于硬盘驱动器、光盘、数字视频盘、磁带和各种存储设备例如DRAM、NAND闪存、NOR闪存、其他的非易失性存储器和固态驱动器。

[0067] 转至图6,根据本发明的某些实施例示出了一种数据传输设备600,其中包括具有解码器输入轮换电路的接收器620。数据传输设备600包括发送器610,发送器610可操作用于通过现有技术已知的传输介质630发送编码信息。通过接收器620从传输介质630接收编码数据。

[0068] 在操作期间,数据由接收器620通过传输介质630接收并利用标准处理进行处理。

这样的标准处理可以利用编码器输入的一种轮换完成。在某些情况下,利用标准处理来处理的一个或多个数据集无法收敛。在这样的情况下,未收敛的数据集被提供作为指示未收敛数据集不可用的输出,并且未收敛数据集以及对应的未检测/编码数据集被保留在数据处理系统内以用于在延迟处理时段期间进行另外的处理。该延迟处理时段例如可以是未通过传输介质630进行任何传输时的时间段。在该延迟处理时段期间,访问未收敛数据集和对应的未检测/编码数据集的组合以供重新处理。重新处理包括对原始轮换的解码器输入执行数据解码以得到第一解码输出,以及对另一种轮换的解码器输入执行数据解码以得到第二解码输出。将第一解码器输出中剩余的不满意校验数量与第二解码器输出中剩余的不满意校验数量进行比较。当第二解码器输出中剩余的不满意校验数量大于第一解码器输出中剩余的不满意校验数量时,选择第一解码器输出用于后续处理。否则就选择第二解码器输出。在本发明的某些实施例中可以使用类似于以上针对图1介绍的数据处理电路,和/或可以完成类似于以上针对图4a-4c介绍的处理。

[0069] 总之,本发明提供了用于数据处理的新型系统、设备、方法和装置。尽管以上已经给出了本发明一个或多个实施例的详细说明,但是各种可选方案、修改和等价形式对于本领域技术人员来说显而易见且并不背离本发明的实质。因此,上述说明内容不应被理解为限制了本发明由所附权利要求界定的保护范围。

R/C	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
0	0	1	0	0	0	0	0	1	1	0	0	0	1	0	1	1	1	1
1	0	0	1	1	0	0	1	1	1	1	1	1	0	0	0	1	1	0
2	0	0	0	0	1	0	1	0	1	0	0	1	1	0	0	0	1	1
3	0	1	1	1	1	1	1	0	0	0	1	0	0	1	1	0	0	1
4	0	0	1	0	1	1	0	1	0	1	0	1	1	1	1	1	0	0
5	1	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	1
6	0	1	0	0	0	1	1	0	1	1	1	1	1	1	0	1	0	1
7	0	1	0	0	0	0	0	0	1	1	1	0	0	1	0	1	1	1
8	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0
9	1	0	0	0	1	1	0	1	0	0	1	1	0	1	0	0	0	0

200

图2a

	Sym A		Sym B		Sym C		Sym D		Sym E		Sym F		Sym G		Sym H		Sym I	
Sym 1	0	1	0	0	0	0	0	1	1	0	0	0	1	0	1	1	1	1
	0	0	1	1	0	0	1	1	1	1	1	1	0	0	0	1	1	0
Sym 2	0	0	0	0	1	0	1	0	1	0	0	1	1	0	0	0	1	1
	0	1	1	1	1	1	1	0	0	0	1	0	0	1	1	0	0	1
Sym 3	0	0	1	0	1	1	0	1	0	1	0	1	1	1	1	1	0	0
	1	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	1
Sym 4	0	1	0	0	0	1	1	0	1	1	1	1	1	1	0	1	0	1
	0	1	0	0	0	0	0	0	1	1	1	0	0	1	0	1	1	1
Sym 5	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0
	1	0	0	0	1	1	0	1	0	0	1	1	0	1	0	0	0	0

210

图2b

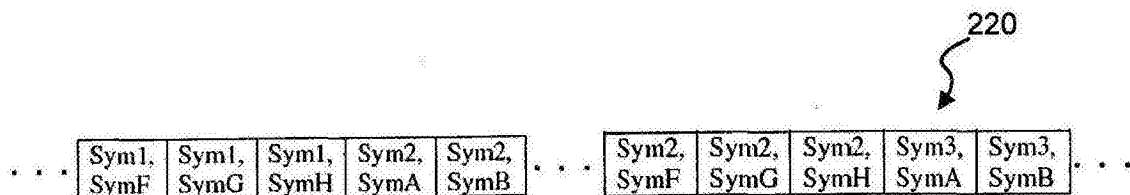


图2c

R/C	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
0	0	1	0	0	0	0	0	1	1	0	0	0	1	0	1	1	1	1
1	0	0	1	1	0	0	1	1	1	1	1	1	0	0	0	1	1	0
2	0	0	0	0	1	0	1	0	1	0	0	1	1	0	0	0	1	1
3	0	1	1	1	1	1	1	0	0	0	1	0	0	1	1	0	0	1
4	0	0	1	0	1	1	0	1	0	1	0	1	1	1	1	1	0	0
5	1	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	1
6	0	1	0	0	0	1	1	0	1	1	1	1	1	1	0	1	0	1
7	0	1	0	0	0	0	0	0	1	1	1	0	0	1	0	1	1	1
8	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0
9	1	0	0	0	1	1	0	1	0	0	1	1	0	1	0	0	0	0

200

图3a

	Sym A		Sym B		Sym C		Sym D		Sym E		Sym F		Sym G		Sym H		Sym I		X
Sym 1	0	1	0	0	0	0	0	0	1	1	0	0	0	1	0	1	1	1	1
	0	0	1	1	0	0	1	1	1	1	1	1	1	0	0	0	1	1	0
Sym 2	0	0	0	0	1	0	1	0	1	0	0	1	1	0	0	0	1	1	1
	0	1	1	1	1	1	1	0	0	0	1	0	0	1	1	0	0	1	1
Sym 3	0	0	1	0	1	1	0	1	0	1	0	1	1	1	1	1	1	0	0
	1	1	1	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1
Sym 4	0	1	0	0	0	1	1	0	1	1	1	1	1	1	1	0	1	0	1
	0	1	0	0	0	0	0	0	1	1	1	1	0	0	1	0	1	1	1
Sym 5	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0
	1	0	0	0	1	1	0	1	0	0	1	1	0	1	0	0	0	0	0

310

图3b

	Sym A		Sym B		Sym C		Sym D		Sym E		Sym F		Sym G		Sym H		Sym I		
Sym 1	1	0	1	0	0	0	0	0	1	1	0	0	0	1	0	1	1	1	1
	0	0	0	1	1	0	0	1	1	1	1	1	1	0	0	0	1	1	0
Sym 2	1	0	0	0	0	1	0	1	0	1	0	0	1	1	0	0	0	1	1
	1	0	1	1	1	1	1	1	0	0	0	1	0	0	1	1	0	0	1
Sym 3	0	0	0	1	0	1	1	0	1	0	1	0	1	1	1	1	1	0	0
	1	1	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	1
Sym 4	1	0	1	0	0	0	1	1	0	1	1	1	1	1	1	0	1	0	1
	1	0	1	0	0	0	0	0	0	1	1	1	0	0	1	0	1	1	1
Sym 5	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0
	0	1	0	0	0	1	1	0	1	0	0	1	1	0	1	0	0	0	0

320

图 3c

330

...	Sym1, SymF	Sym1, SymG	Sym1, SymH	Sym2, SymA	Sym2, SymB	...	Sym2, SymF	Sym2, SymG	Sym2, SymH	Sym3, SymA	Sym3, SymB	...
-----	---------------	---------------	---------------	---------------	---------------	-----	---------------	---------------	---------------	---------------	---------------	-----

图 3d

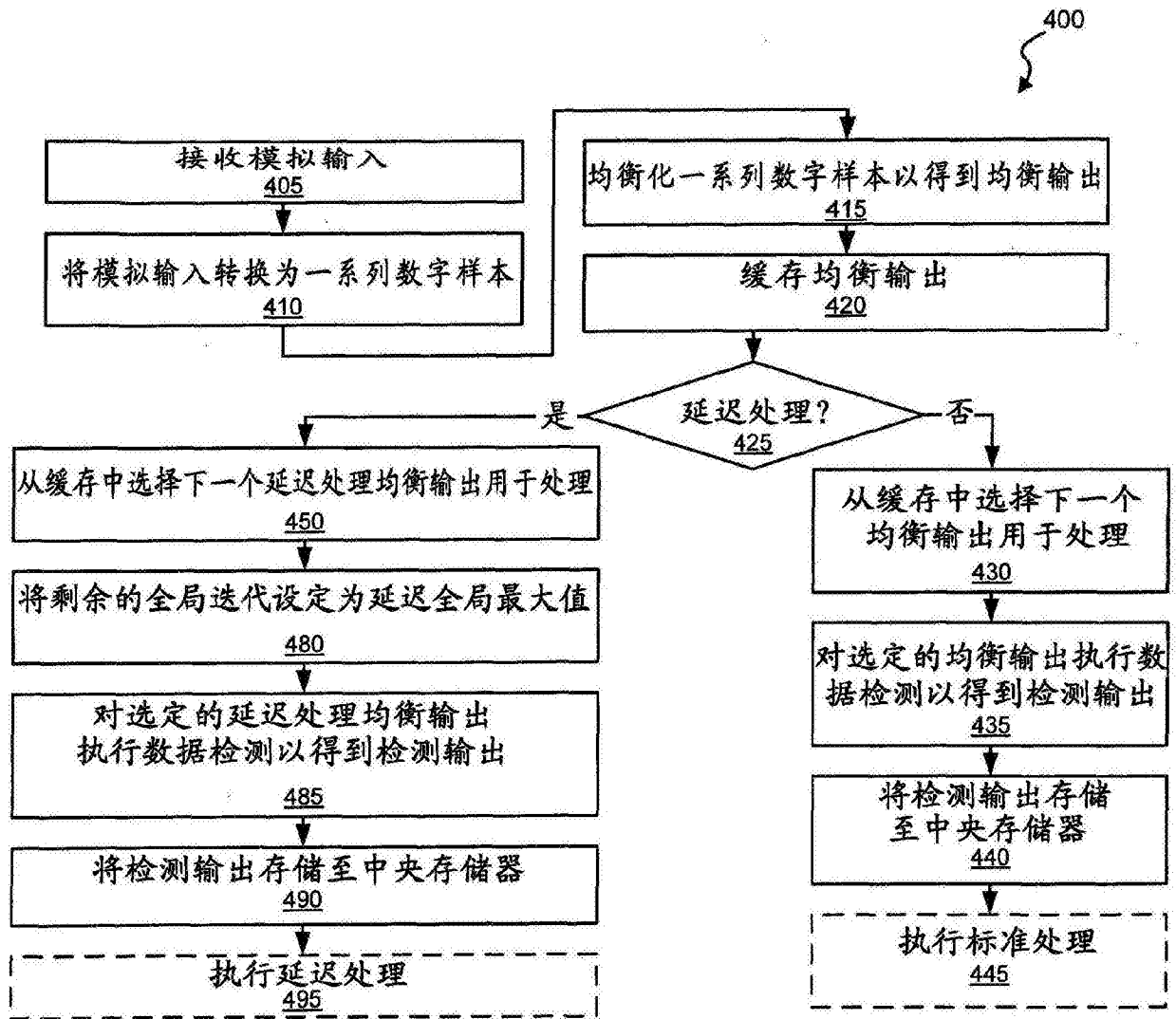


图4a

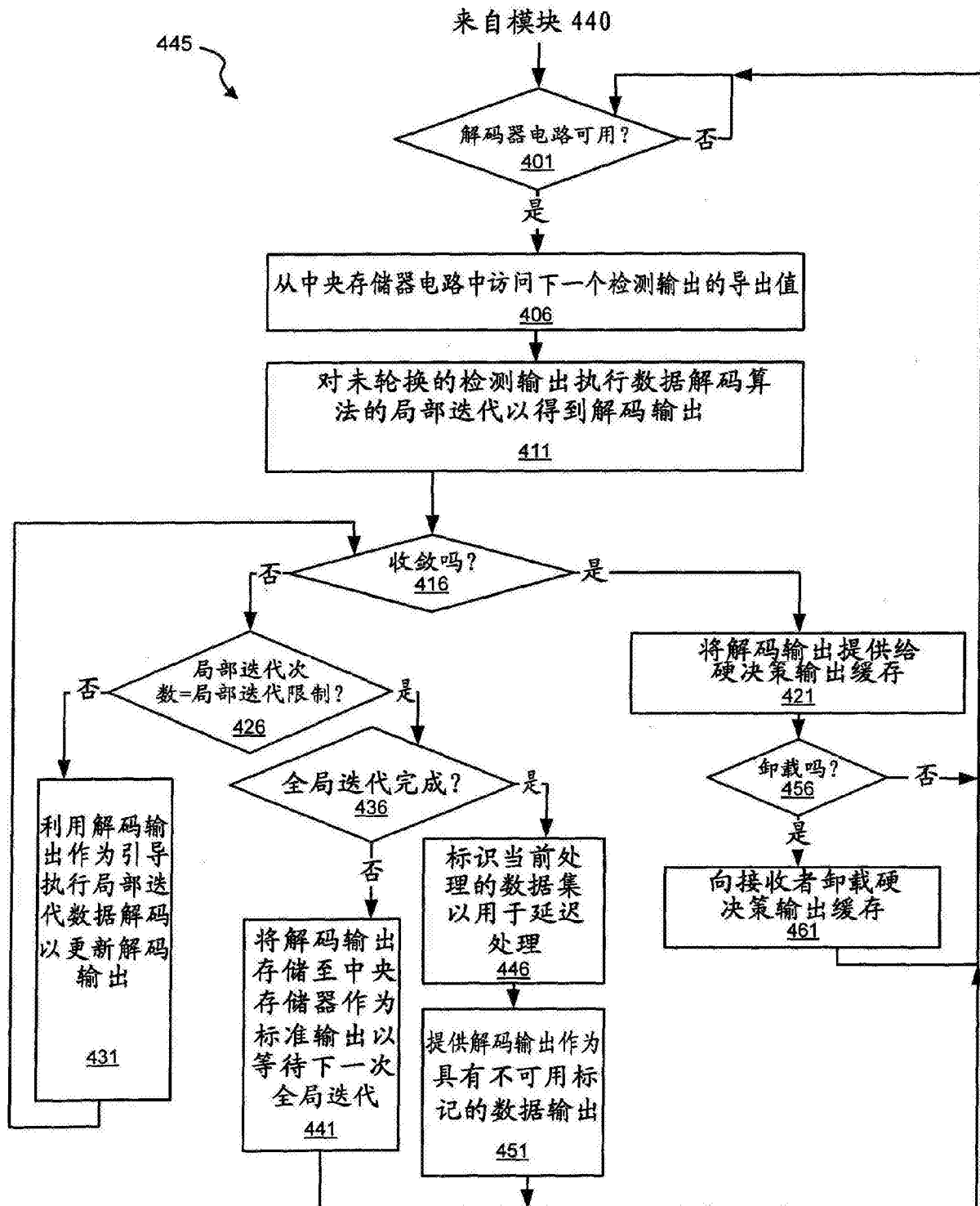


图4b

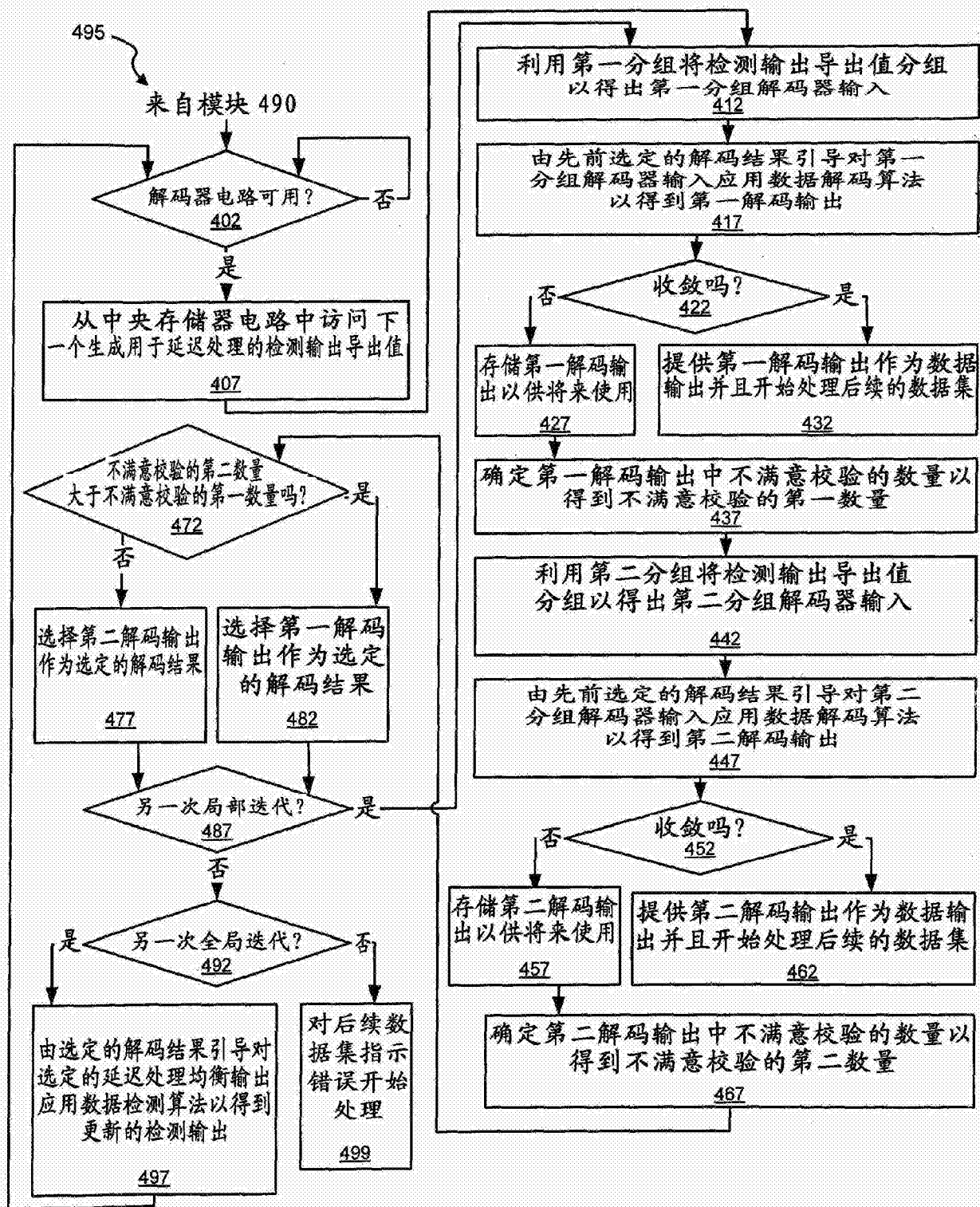


图4c

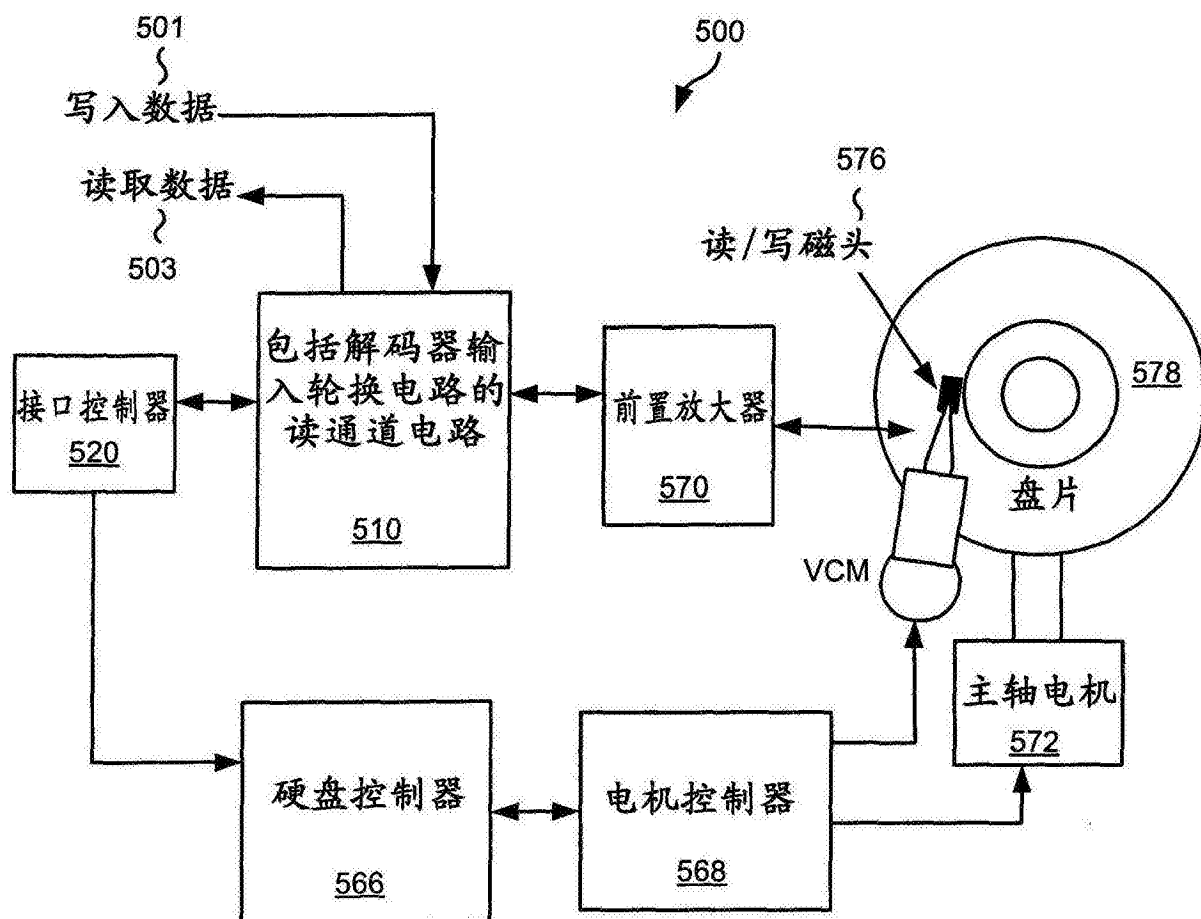


图5

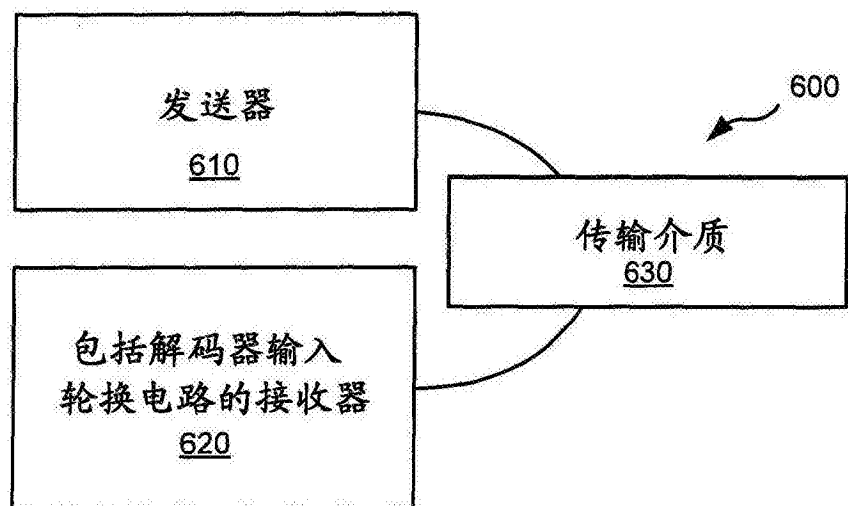


图6