



(12)发明专利申请

(10)申请公布号 CN 105900340 A

(43)申请公布日 2016.08.24

(21)申请号 201480055750.7

(74)专利代理机构 上海专利商标事务所有限公司 31100

(22)申请日 2014.10.09

代理人 唐杰敏

(30)优先权数据

61/889,030 2013.10.09 US

61/946,647 2014.02.28 US

(51)Int.Cl.

H03M 13/09(2006.01)

(85)PCT国际申请进入国家阶段日

2016.04.08

(86)PCT国际申请的申请数据

PCT/US2014/059981 2014.10.09

(87)PCT国际申请的公布数据

W02015/054548 EN 2015.04.16

(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 S·森戈库

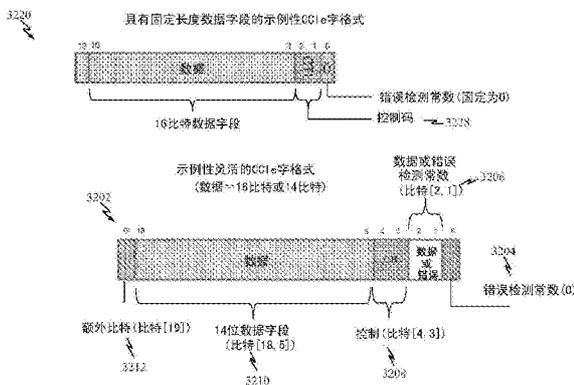
权利要求书2页 说明书18页 附图38页

(54)发明名称

CC1e协议上的错误检测能力

(57)摘要

提供了一种设备,包括共享总线、从设备和主设备。从设备可耦合至共享总线。主设备可耦合至控制数据总线并且被适配成管理共享总线上的通信。共享总线上的传输是被编码到三进制数中的多个比特,这些三进制数随后被转码成码元以供传输,并且该多个比特中的3个最低有效位或者最低有效位被用于传输的错误检测。



1. 一种设备,包括:
共享总线;
耦合至所述共享总线的从设备;
耦合至所述控制数据总线并且被适配成管理所述共享总线上的通信的主设备;并且
其中所述共享总线上的传输是被编码到三进制数中的多个比特,所述三进制数随后被转码成码元以供传输,并且所述多个比特中的3个最低有效位或者最低有效位被用于所述传输的错误检测。
2. 如权利要求1所述的设备,其特征在于,所述共享总线是双线总线,并且所述双线总线中的两条线均被用于传输所述码元。
3. 如权利要求1所述的设备,其特征在于,时钟信号被嵌入在码元到码元转变内。
4. 如权利要求3所述的设备,其特征在于,所述从设备实现状态机逻辑电路,所述状态机逻辑电路检测所述时钟信号中的错误的时钟脉冲缺失和错误的额外时钟脉冲。
5. 如权利要求1所述的设备,其特征在于,所述多个比特是20比特序列。
6. 如权利要求5所述的设备,其特征在于,将所述3个最低有效位用于错误检测保证对整个20比特序列中的错误的检测。
7. 如权利要求1所述的设备,其特征在于,第二和第三最低有效位的使用被灵活地分配给数据传输或错误检测。
8. 如权利要求7所述的设备,其特征在于,将所述最低有效位用于错误检测仅在约50%的时间保证对整个20比特序列中的单码元错误的检测。
9. 如权利要求1所述的设备,其特征在于,在仅所述最低有效位被用于错误检测时,所述最低有效位的值被设置成恒定的二进制值“1”或“0”。
10. 如权利要求6所述的设备,其特征在于,在所述3个最低有效位被用于错误检测时,所述3个最低有效位的值被设置成恒定的二进制值“000”或“111”。
11. 一种在发射机设备上操作的方法,包括:
获得要在共享总线上传送的多个比特,其中所述多个比特中的3个最低有效位或最低有效位被用于错误检测;
将所述多个比特转换成三进制数;
将所述三进制数的各个数位转换成顺序码元;以及
将多个码元组合成字以供传输。
12. 如权利要求11所述的方法,其特征在于,进一步包括:
在所述共享总线上向接收方设备传送所述字。
13. 如权利要求11所述的方法,其特征在于,进一步包括:
计算所传送比特的校验和;以及
向所述接收方设备传送所传送比特的所述校验和。
14. 如权利要求11所述的方法,其特征在于,所述校验和被插入在任何的任意字内以供传送给所述接收方设备。
15. 如权利要求1所述的方法,其特征在于,时钟被嵌入在码元到码元转变内。
16. 如权利要求11所述的方法,其特征在于,所述共享总线是双线总线,并且所述双线总线中的两条线均被用于传输所述码元。

17. 一种发射机设备,包括:
用于耦合至共享总线的总线接口;
处理电路,其耦合到所述总线接口并且被配置成:
获得要在共享总线上传送的多个比特,其中所述多个比特中的3个最低有效位或最低有效位被用于错误检测;
将所述多个比特转换成三进制数;
将所述三进制数的各个数位转换成顺序码元;以及
将多个码元组合成字以供传输。
18. 如权利要求17所述的设备,其特征在于,所述处理电路被进一步配置成:在所述共享总线上向接收方设备传送所述字。
19. 如权利要求17所述的设备,其特征在于,所述处理电路被进一步配置成:
计算所传送比特的校验和;以及
向所述接收方设备传送所传送比特的所述校验和。
20. 如权利要求17所述的设备,其特征在于,所述校验和被插入在任何的任意字内以供传送给所述接收方设备。
21. 如权利要求17所述的设备,其特征在于,时钟被嵌入在码元到码元转变内。
22. 如权利要求17所述的设备,其特征在于,所述共享总线是双线总线,并且所述双线总线中的两条线均被用于传输所述码元。
23. 一种接收机设备,包括:
用于耦合至共享总线的总线接口;
处理电路,其耦合到所述总线接口并且被配置成:
在所述共享总线上接收多个码元;
将所述多个码元转换成各个数位;
将所述各个数位组合成三进制数;以及
将所述三进制数转换成比特,其中所述多个比特中的3个最低有效位或最低有效位被用于错误检测。
24. 如权利要求23所述的接收机设备,其特征在于,所述处理电路被进一步配置成:
基于所述多个比特中的所述3个最低有效位或所述最低有效位来查明所接收到的多个比特内的错误。
25. 如权利要求23所述的接收机设备,其特征在于,时钟被嵌入在所述多个收到码元的码元到码元转变内。
26. 如权利要求23所述的接收机设备,其特征在于,进一步包括:
状态机逻辑电路,所述状态机逻辑电路检测所述时钟信号中的错误的时钟脉冲缺失和错误的额外时钟脉冲。
27. 如权利要求23所述的接收机设备,其特征在于,所述共享总线是双线总线,并且所述双线总线中的两条线均被用于传输所述码元。

CCl_e协议上的错误检测能力

[0001] 相关申请的交叉引用

[0002] 本专利申请要求于2013年10月9日提交的题为“Error Correction Capability Over CCl_e Bus(CCl_e总线上的纠错能力)”的美国临时申请申请号61/889,030,以及于2014年2月28日提交的题为“Bit Allocation Over A Camera Control Interface Extended Bus(相机控制接口扩展总线上的比特分配)”的美国临时申请申请号No.61/946,647的优先权,这两篇申请均被转让给本申请受让人并且通过援引被明确纳入于此。

[0003] 领域

[0004] 本公开涉及实现共享总线上的高效操作,尤其涉及促成共享相机控制接口扩展(CCl_e)总线上的错误检测。

背景技术

[0005] I²C(也称为I²C)是被用于将低速外围设备附连至母板、嵌入式系统、蜂窝电话或其他电子设备的多主控串行单端总线。I²C总线包括具有7比特寻址的时钟(SCL)和数据(SDA)线。该总线具有两个设备角色:主设备和从设备。主设备是生成时钟并发起与从设备的通信的设备。从设备是接收时钟并在被主设备寻址时进行响应的设备。I²C总线是多主控总线,这意味着可存在任何数目的主设备。另外,主角色和从角色可以在消息之间(在STOP被发送之后)改变。I²C定义了基本类型的消息,其中每种消息始于开始(START)并结束于停止(STOP)。

[0006] 在相机实现的此上下文中,单向传输可被用于从传感器捕捉图像并向基带处理器中的存储器传送此类图像数据,而控制数据可在该基带处理器与传感器以及其他外围设备之间交换。在一个示例中,相机控制接口(CCI)协议可被用于基带处理器与图像传感器(和/或一个或多个从设备)之间的此类控制数据。在一个示例中,CCI协议可在图像传感器与基带处理器之间的I²C串行总线上实现。

[0007] 在共享总线的诸设备之间的传输中提供错误检测将是期望的。

[0008] 概述

[0009] 提供了一种设备,包括共享总线、从设备和主设备。从设备可耦合至共享总线。主设备可耦合至控制数据总线并且被适配成管理共享总线上的通信。共享总线上的传输是被编码到三进制数中的多个比特,这些三进制数随后被转码成码元以供传输,并且该多个比特中的3个最低有效位或者最低有效位被用于传输的错误检测。

[0010] 共享总线可以是双线总线,并且该双线总线中的两条线均被用于传输码元。时钟信号可被嵌入在码元到码元转变内。从设备可实现状态机逻辑电路,该状态机逻辑电路检测时钟信号中的错误的时钟脉冲缺失和错误的额外时钟脉冲。

[0011] 在一个示例中,该多个比特可以是20比特序列。将3个最低有效位用于错误检测保证了对整个20比特序列中的错误的检测。另外,第二和第三最低有效位的使用可以被灵活地分配给数据传输或错误检测。在3个最低有效位被用于错误检测时,这3个最低有效位的值可以被设置成恒定的二进制值“000”或“111”。

[0012] 将最低有效位用于错误检测仅在约50%的时间保证对整个20比特序列中的单码元错误的检测。在一个示例中,在仅最低有效位被用于错误检测时,该最低有效位的值被设置成恒定的二进制值“1”或“0”。

[0013] 根据一个方面,提供了一种包括总线接口和处理电路的发射机设备。总线接口可用于将发射机设备耦合至共享总线(接收方设备或从设备耦合至该共享总线)。发射机设备可管理或控制共享总线上的通信。该处理电路可被配置成:(a)获得要在共享总线上传送的多个比特,其中该多个比特中的3个最低有效位或最低有效位被用于错误检测;(b)将该多个比特转换成三进制数;(c)将该三进制数的各个数位转换成顺序码元;(d)将多个码元组合成字以供传输;和/或(e)在共享总线上向接收方设备传送该字。时钟可被嵌入在码元到码元转变内。共享总线可以是双线总线,并且该双线总线中的两条线均被用于传输码元。该处理电路可以被进一步配置成:(a)计算所传送比特的校验和;和/或(b)向接收方设备传送所传送比特的校验和。该校验和可被插入在任何的任意字内以供传送给接收方设备。

[0014] 根据另一方面,提供了一种接收机设备,包括:总线接口和处理电路。总线接口可用于耦合至主设备在其上管理通信的共享总线。该处理电路可被配置成:(a)在共享总线上接收多个码元;(b)将该多个码元转换成数位;(c)将这些数位组合成三进制数;(d)将该三进制数转换成比特,其中该多个比特中的3个最低有效位或最低有效位被用于错误检测;和/或(e)基于该多个比特中的3个最低有效位或最低有效位来查明所接收到的多个比特内的错误。时钟可被嵌入在多个收到码元的码元到码元转变内。共享总线可以是双线总线,并且该双线总线中的两条线均被用于传输码元。接收机设备可实现状态机逻辑电路,该状态机逻辑电路检测时钟信号中的错误的时钟脉冲缺失和错误的额外时钟脉冲。

[0015] 附图

[0016] 在结合附图理解下面阐述的详细描述时,各种特征、本质、和优点会变得明显,在附图中,相像的附图标记贯穿始终作相应标识。

[0017] 图1是解说具有基带处理器和图像传感器并且实现图像数据总线和多模式控制数据总线的设备的框图。

[0018] 图2解说了时钟可如何在CCIe模式中被嵌入在码元到码元转变内,由此允许将I2C总线中的两条线(即,SDA线和SCL线)用于数据传输。

[0019] 图3是解说用于在发射机处将数据比特转码成经转码码元以在这些经转码码元内嵌入时钟信号的示例性方法的框图。

[0020] 图4解说了转变数与顺序码元之间的示例性转换。

[0021] 图5解说了转变数与顺序码元之间的转换。

[0022] 图6解说了用于从最高有效位到最低有效位将二进制比特转换成三进制数的方法。

[0023] 图7解说了用于从最高有效位到最低有效位将二进制比特转换成三进制数的发射机侧逻辑电路。

[0024] 图8解说了用于从最高有效位到最低有效位将三进制数转换成二进制比特的方法。

[0025] 图9解说了用于将12位三进制数转换成20比特的接收机侧逻辑电路。

[0026] 图10概念性地解说了比特19(即,在比特计数开始于为比特0的第一比特时的第20

比特)大多数情况下在CCIE协议中不被使用并且可被用于共享总线上的设备之间的命令。

[0027] 图11解说了对CCIE模式进入指示符的示例性通用调用,该CCIE模式进入指示符可由主设备在共享总线上发送以向从设备指示共享总线正在从I2C操作模式切换到CCIE操作模式。

[0028] 图12解说了可由CCIE主设备(例如,图1中的处于I2C模式的主设备)发布以向所有具有CCIE能力的设备指示从CCIE模式转变到I2C模式的示例性CCIE调用。

[0029] 图13解说了示例性CCIE从标识符(SID)字格式。

[0030] 图14解说了示例性CCIE地址字格式。

[0031] 图15解说了示例性写数据字格式。

[0032] 图16解说了示例性读规范字格式。

[0033] 图17解说了示例性读数据字格式。

[0034] 图18解说了I2C一个字节写数据操作的示例性时序图。

[0035] 图19解说了其中数据比特已被转码成12个码元以供在SDA线和SCL线上传送的示例性CCIE传输。

[0036] 图20解说了从图2-10中解说的编码方案得到的第20比特(比特19)的示例性映射。

[0037] 图21解说了图20的第20比特(比特19)区域的示例性映射内的子区域的细节。

[0038] 图22解说了可能发生的各种码元错误状况。

[0039] 图23解说了示出所传送的码元序列0321_0321_0321(其转译成二进制序列0000_0000_0000_0000和三进制数0000_0000_0000₃)中的可能错误的表以及可如何在3个最低有效位内检测此类错误。

[0040] 图24解说了示出所传送的码元序列2301_2301_2301(其转译成二进制序列0100_0000_1101_1111_1000和三进制数1111_1111_1111₃)中的可能错误的表以及可如何在3个最低有效位内检测此类错误。

[0041] 图25解说了示出所传送的码元序列3131_3131_3131(其转译成二进制序列1000_0001_1011_1111_0000和三进制数2222_2222_2222₃)中的可能错误的表以及可如何在3个最低有效位内检测此类错误。

[0042] 图26解说了示出所传送的码元序列0132_3101_3231中的可能错误的表以及可如何在3个最低有效位内检测此类错误。

[0043] 图27解说了示出所传送的码元序列2030_2120_3021中的可能错误的表以及可如何在3个最低有效位内检测此类错误。

[0044] 图28解说了示出所传送的码元序列3231_0132_3101中的可能错误的表以及可如何在3个最低有效位内检测此类错误。

[0045] 图29解说了CCIE字可如何将3个最低有效位或最低有效位用于错误检测。

[0046] 图30解说了可如何在CCIE字内实现8比特校验和。

[0047] 图31解说了8比特校验和生成器的示例性实现。

[0048] 图32解说了促成错误检测的示例性CCIE字格式。

[0049] 图33是解说根据本公开提供的主/从设备的示例性组件的框图。

[0050] 图34解说了根据协议(例如,CCIE协议)的一个或多个错误检测特征来促成通信的示例性方法。

[0051] 图35解说了不具有时钟错误的示例性CCIE字传输。

[0052] 图36解说了具有单个时钟缺失错误的示例性CCIE字传输。

[0053] 图37解说了具有两个时钟缺失错误的示例性CCIE字传输。

[0054] 图38解说了具有额外时钟错误的示例性CCIE字传输。

[0055] 详细描述

[0056] 在以下描述中,给出了具体细节以提供对诸实施例的透彻理解。然而,本领域普通技术人员将理解,没有这些具体细节也可实践这些实施例。例如,电路可能用框图示出以免使这些实施例混淆在不必要的细节中。在其他实例中,公知的电路、结构、和技术可能不被详细示出以免使这些实施例不明朗。

[0057] 总览

[0058] 已开发出被称为CCIE(相机控制接口扩展)的对CCI的扩展,该CCIE将二进制数转换成三进制数,该三进制数随后被转码成嵌入有时钟的码元以供在双线I2C总线上传送,从而实现比之前更高的速度。二进制数首先被转换成三进制数(即,基数为3的数)。三进制数的每一数位随后被转换成码元,其中没有两个顺序码元重复并且仅使用有限数目的码元(例如,4个码元)。这些码元随后在总线上传送。

[0059] 使用三进制数空间并且转换成码元导致额外比特变得可用。在一个示例中,此额外比特可以是最高有效位,从而三进制数的一区域变得可用以支持原本不可用的其他功能性。例如,由于可被包括在此额外比特中的额外信息,因而可以促成错误检测和校验和。

[0060] 根据一个方面,每个所传送字(例如,20比特字)的最低有效位可被用于检测可能在传输期间出现的约一半错误。

[0061] 根据第二方面,灵活的校验和字被添加到通信协议以检测传输中的错误。该校验和字可被插入到传输中的任何的任意点处。

[0062] 根据第三方面,字内的各个比特被安排成使得3个最低有效位可用于整个字上的错误检测。

[0063] 根据第四方面,接收机设备可维护状态机逻辑电路,该状态机逻辑电路帮助检测嵌入在CCIE字的码元到码元转变中的时钟信号内的缺失的时钟脉冲和额外的时钟脉冲以检测错误。

[0064] 示例性操作环境

[0065] 图1是解说具有基带处理器104和图像传感器106并且实现图像数据总线116和多模式控制数据总线108的设备102的框图。虽然图1解说了相机设备内的多模式控制数据总线108,但是应当清楚,该控制数据总线108可在各种不同设备和/或系统中实现。可在图像数据总线116(例如,高速差分DPHY链路)上将图像数据从图像传感器106发送给基带处理器104。

[0066] 在一个示例中,控制数据总线108可以是包括两条线路(时钟线(SCL)和串行数据线(SDA))的I2C总线。时钟线SCL可被用来发送被用于同步I2C总线(控制数据总线108)上的所有数据传输的时钟。数据线SDA和时钟线SCL耦合至I2C总线(控制数据总线108)上的所有设备112、114、和118。在该示例中,可以经由控制数据总线108在基带处理器104与图像传感器106以及其他外围设备118、122和/或124之间交换控制数据。I2C的标准时钟(SCK)速度最高达100KHz。I2C快速模式中的标准时钟SCL速度最高达400KHz,并且在I2C快速模式+(Fm+)

中最高达1MHz。在用于相机应用时，I2C总线上的这些操作模式可被称为相机控制接口(CCI)模式。

[0067] 根据一个方面，可以在多模式控制数据总线108上实现改进的操作模式(即，大于1MHz的控制数据总线传输频率)以支持相机操作。I2C总线上的这种改进的操作模式在用于相机应用时可被称为相机控制接口扩展(CCIe)模式。在CCIe模式中，SCL线和SDA线两者均可用于传送数据，而时钟被嵌入在这两条线上的码元到码元转变内。在该示例中，基带处理器104包括主设备112且图像传感器106包括从设备114，主设备112和从设备114两者可根据相机控制接口扩展(CCIe)模式在控制数据总线108上操作，而不影响耦合至控制数据总线108的其他旧式I2C设备的正确操作。根据一个方面，控制数据总线108上的这种改进模式可在CCIe设备与旧式I2C从设备之间没有任何桥接设备的情况下实现。

[0068] 提供了准许兼容I2C的设备和兼容CCIe的设备并发地耦合至共享控制数据总线108的协议。控制数据总线108可动态地在根据不同通信协议(例如，I2C模式和CCIe模式)操作之间切换。如先前提及的，多模式主设备112管理对共享控制数据总线108的通信和/或访问。主设备传送进入调用以指示控制数据总线108要将其通信协议从第一协议模式(例如，I2C模式)切换到第二协议模式(例如，CCIe模式)。类似地，主设备传送退出调用以指示控制数据总线108要将其通信协议从第二协议模式(例如，CCIe模式)切换到第一协议模式(例如，I2C模式)。耦合至共享总线108的从设备监视这些进入和退出调用以查明它们何时可在共享总线108上操作。

[0069] 示例性CCIe编码技术

[0070] 图2解说了时钟可如何在CCIe模式中被嵌入在码元到码元转变内，由此允许将I2C总线中的两条线(即，SDA线和SCL线)用于数据传输。在一个示例中，时钟的这种嵌入可以通过转变时钟转码来达成。例如，要在物理链路(线路)上传送的数据204被转码，从而保证所传送码元在所传送码元206的每个码元循环或转变处改变状态。在一个示例中，比特序列被转换成三进制数，并且该三进制数的每一位被转换成码元以供传输。即使在三进制数的两个顺序数位相同的情况下，也保证顺序码元是不同的。因此，原始时钟202可被嵌入在每个码元循环的码元状态的改变中。接收机从(所传送码元206中的)每个码元处的状态转变恢复时钟信息208并且随后倒转对所传送码元206的转码以获得原始数据210。在一个示例中，每个码元被转换成一数位，多个数位构成一三进制数，其中该三进制数随后被转换成多个比特。因此，原始时钟202可被嵌入在每个码元循环的码元状态的改变中。这允许I2C总线的两条线路(图1中的控制数据总线108，SDA线和SCL线)被用于发送数据信息。另外，码元速率可以加倍，因为不再需要在时钟信号与数据信号之间具有建立时间和保持时间。

[0071] 图3是解说用于在发射机处将数据比特转码成经转码码元以在这些经转码码元内嵌入时钟信号的示例性方法的框图。在发射机302处，数据比特序列304被转换成三进制(基数为3)数(即，“转变数”)，并且这些三进制数随后被转换成在时钟线SCL 312和数据线SDA 314上传送的(顺序)码元。

[0072] 在一个示例中，原始的20比特二进制数据被输入到比特至转变数转换器块308以转换成12位三进制数。该12位三进制数的每一位表示“转变数”。两个连贯转变数可具有相同的数字(即，三进制数的连贯数位可以是相同的)。每个转变数在转变至码元块310处被转换成顺序码元以使得任何两个连贯的顺序码元不具有相同值。由于在每个顺序码元处保

证有转变,因而此类顺序码元转变可用于嵌入时钟信号。每个顺序码元316随后在双线物理链路(例如,包括SCL线312和SDA线314的I2C总线)上被发送。

[0073] 图4解说了转变数402与顺序码元404之间的示例性转换。三进制数(基数为3的数)的个体数位(也被称为转变数)可具有三(3)个可能数字或状态0、1或2之一。虽然在三进制数的两个连贯数位中可能出现相同数字,但任何两个连贯的顺序码元不具有相同值。转变数和顺序码元之间的转换保证顺序码元即使在连贯的转变数是相同的情况下也总是改变(从顺序码元到顺序码元)。

[0074] 在图5中解说性地阐述转换功能。在发射机侧(TX:T到S)502,转变数(T)可被转换成顺序码元(S)。例如,当前的顺序码元(C_s)可基于先前的顺序码元(P_s)和作为当前转变数(T)的函数的临时转变数(T_{tmp})来获得。临时转变数(T_{tmp})可通过将当前转变数T与0进行比较来获得,并且当 $T=0$ 时,临时转变数(T_{tmp})变成等于3,否则(当 $T \neq 0$ 时) T_{tmp} 变成等于T(即, $T_{tmp}=T=0?3:T$)。当前顺序码元可以作为当前顺序码元(C_s)加先前顺序码元(P_s)加临时转变数(T_{tmp})的总和(即, $C_s=P_s+T_{tmp}$)来获得。

[0075] 在接收机侧(RX:S到T)504,转换操作被倒转以从当前顺序码元(C_s)和先前顺序码元(P_s)获得转变数。临时转变数(T_{tmp})可以作为当前顺序码元(C_s)加4减去先前顺序码元(P_s)的总和(即, $T_{tmp}=C_s+4-P_s$)来获得。当前转变数(T)等于临时转变数(T_{tmp}),但是临时转变数(T_{tmp})与三(3)进行比较,并且当 $T_{tmp}=3$,临时转变数(T_{tmp})变成等于零(0),否则(当 $T_{tmp} \neq 3$ 时)T变成等于 T_{tmp} (即, $T=T_{tmp}=3?0:T$)。

[0076] 表506解说了转变数与顺序码元之间的转换。

[0077] 再次参照图4,本文解说了转变数与顺序码元之间的转换的示例。例如,在第一循环406中,当前转变数(T_a)为2,所以 T_{tmp} 也为2,并且在先前顺序码元 P_s 为1的情况下,新的当前顺序码元 C_s 现在为3。

[0078] 在第二循环408中,转变数(T_b)为1。由于转变数(T_b)不等于0,因而临时转变数 T_{tmp} 等于转变数(T_b)值1。通过将先前顺序码元(P_s)值3与临时转变数 T_{tmp} 1相加来获得当前顺序码元(C_s)。由于该加法运算的结果等于4,大于3,因而翻转数0变成当前顺序码元(C_s)。

[0079] 在第三循环410中,当前转变数(T)为1。因为转变数T为1,所以临时转变数 T_{tmp} 也为1。通过将先前顺序码元(P_s)值0与临时转变数 T_{tmp} 1相加来获得当前顺序码元(C_s)。由于加法运算的结果等于1,不大于3,因而当前码元(C_s)等于1。

[0080] 在第四循环412中,当前转变数(T)为0。因为转变数T为0,所以临时转变数 T_{tmp} 为3。

[0081] 通过将先前顺序码元(P_s)值1与临时转变数 T_{tmp} 3相加来获得当前顺序码元(C_s)。由于该加法运算的结果为4,大于3,因而翻转数0变成当前顺序码元(C_s)。

[0082] 注意,即使两个连贯的三进制位 T_b 和 T_c 具有相同数字,该转换也保证了两个连贯顺序码元具有不同状态值。正因如此,顺序码元404中的经保证转变可用于嵌入时钟信号,由此释放I2C总线中的时钟线SCL以用于数据传输。

[0083] 再次参照图3,在接收机320处,该过程被倒转以将经转码码元转换回比特,并且在该过程中从码元转变中提取时钟信号。接收机320在双线物理链路(例如,包括SCL线324和SDA线326的I2C总线)上接收顺序码元序列322。所接收的顺序码元322被输入时钟数据恢复(CDR)块328以恢复时钟定时并采样经转码码元(S)。码元至转变数转换器块330随后将经转码(顺序)码元转换成转变数(即,一个三进制数位数字)。接着,转变数至比特转换器332转

换12个转变数以从12位三进制数恢复20比特原始数据。

[0084] 图3和4中针对双线系统和12个转变数解说的示例可被一般化为n线系统和m个转变数。如果每一个T(T₀到T_{m-1})存在r个可能的码元转变状态,那么m个转变可发送 r^m 个不同状态(即, $r=2^n-1$)。因此,转变T₀...T_{m-1}包含可具有 $(2^n-1)^m$ 个不同状态的数据。

[0085] 本文解说的技术可被用来增加控制总线108(图1)的链路速率,超出I2C标准总线提供的链路速率,并且于此被称为CCIe模式。在一个示例中,耦合至控制数据总线108的主设备和/或从设备可实现在码元传输内嵌入时钟信号的发射机和/或接收机(如图2、3、4和5中所解说的),以在相同控制数据总线上达成比使用标准I2C总线可能达成的更高比特率。

[0086] 图6解说了用于从最高有效位到最低有效位将二进制比特转换成三进制数的方法。三进制数的每一数位可被转码(转换)成传送给接收方设备的码元。对于12位三进制数602,其中T₀,T₁...T₁₁表示该三进制数,T₀表示 3^0 数位(并且是最低有效位),而T₁₁表示 3^{11} 位(并且是最高有效位)。从收到比特(例如,20比特序列)开始,首先获得三进制数602的最高有效位T₁₁。随后,接下来获得下一最高有效位T₁₀。此过程继续直至获得最低有效位T₀。三进制数602的每一数位也可被称为“转变数”。

[0087] 图7解说了用于从最高有效位到最低有效位将二进制比特转换成三进制数的发射机侧逻辑电路。图6和7解说了以T₁₁、T₁₀、T₉、...、T₀的次序发送的12位三进制数602。通过首先获得和发送最高有效位,所涉及的逻辑和电路系统的复杂度被简化。在图6和7中的办法中,最高有效顺序码元被首先传送给接收方设备,并且因此被称为先MSS(首先最高有效码元)。如本文所使用的,“最低有效码元”指代与三进制数602的最低有效位相对应的经转码码元。例如并且参照图4和5,当T₀被转码成顺序码元时,该码元是最低有效码元,因为它源自最低有效三进制位。类似地,如本文所使用的,“最高有效码元”指代与三进制数602的最高有效位相对应的经转码码元。例如并且参照图4和5,当T₁₁被转码成顺序码元时,该码元是最高有效码元,因为它源自最高有效三进制位。并且在码元到转变数转换器块330(图3)随后接收并且将经转码的(顺序)码元转换成转变数(即,三进制数的数位)时,其将首先是最高有效位T₁₁并且最后是最低有效位T₀。

[0088] 回头参考3,20比特原始数据被按倒序转换成三进制数(即,最高有效位被首先提供给转换器),随后三进制数的每一数位(例如,每个转变数)被按倒序转换(即,转码)成顺序码元,并且这些经转码码元按倒序(即,首先最高有效码元)在总线上传送。

[0089] 图8解说了用于从最高有效位到最低有效位将三进制数转换成二进制比特的方法。即,此接收机侧转换倒转图6和7中所解说的发射机侧转换中执行的操作。接收方设备(例如,从设备)接收倒序传输并且执行时钟恢复和码元采样以将经转码码元转换回三进制数,该三进制数随后以倒序被提供给图9中的将该三进制数转换回20比特二进制原始数据的逻辑电路。图7解说了具有耦合至去往逻辑设备的单个输出端的12个输入端的复用器。

[0090] 图9解说了用于将12位三进制数转换成20比特的接收机侧逻辑电路。

[0091] 图10概念性地解说了比特19(即,在比特计数开始于为比特0的第一比特时的第20比特)大多数情况下在CCIe协议中不被使用并且可被用于共享总线上的设备之间的命令。即,作为图3-9中解说的编码方案的结果,所传送码元中的额外比特(即,比特19)现在可用。更具体地,图10解说了比特19(即,第20比特)。换言之,如计算机科学中典型的,从零开始逐比特计数,并且比特19是第20比特。此处,在三进制数范围0000_0000_0000₃到2221_2201_

2001₃内表示比特0-18。范围2221_2201_2002₃到2222_2222_2222₃中的三进制数未被使用。因此,三进制数范围2221_2201_2002₃到2222_2222_2222₃可被用于表示比特19(即,第20比特)。换言之,三进制2221,2201,2002₃是二进制10,000,000,000,000,000,000(十六进制0x80000),并且三进制2222_2222_2222₃(0x81BF0)是可能的最大12位三进制数。

[0092] 用于CCIE模式的示例性协议

[0093] 图11解说了对CCIE模式进入指示符的示例性通用调用,该CCIE模式进入指示符可由主设备在共享总线上发送以向从设备指示共享总线正在从I2C操作模式切换到CCIE操作模式。通用调用1102可由I2C主设备在共享总线(例如,图1中的处于I2C模式的主设备112在SDA线和SCL线)上发布以向所有兼容I2C的设备指示从I2C模式到CCIE模式的转变。

[0094] 在I2C模式中,CCIE主设备发布具有“CCIE模式”字节或指示符1104的此I2C通用调用1102。兼容CCIE的从设备确认对通用调用1102的接收。兼容CCIE的从设备可通过在通用调用期间保持(控制数据总线108的)SCL线为低来插入等待循环(若必要)。

[0095] 一旦处于CCIE模式,所有兼容CCIE的设备能够对来自CCIE主设备的请求作出响应。不支持CCIE模式的兼容I2C的旧式从设备在共享控制数据总线上的操作状态或任何功能性不受任何CCIE事务的影响。

[0096] 图12解说了可由CCIE主设备(例如,图1中的处于I2C模式的主设备112)发布以向所有具有CCIE能力的设备指示从CCIE模式到I2C模式的转变的示例性CCIE调用1202。CCIE主设备可发布此退出调用1202来代替CCIE SID。

[0097] 在CCIE模式中,在CCIE模式中的之后跟随着S的最末数据之后,CCIE主设备发送特殊的CCIE SID码(“退出”码/指示符1204)以指示(例如,向兼容CCIE的设备)CCIE模式的结束并且转变回到I2C模式。另外,在“退出”码/指示符1204之后,CCIE主设备根据I2C协议来发送之后跟随着“通用调用”1206的S(开始比特),其中“退出”码1208在I2C协议内的第二个字节处。所有具有CCIE能力的从设备都必须确收通用调用1204。

[0098] 图13解说了示例性CCIE从标识符(SID)字格式。其解说了将16比特从标识符(SID)1304用作CCIE SID字格式1302的一部分。此类SID字格式将在该字被置于控制数据总线上时被用于标识特定的从设备。

[0099] 图14解说了示例性CCIE地址字格式1402。其解说了每个地址字1406包括16比特地址1404。地址字1406还包括2比特控制码1408和1比特错误检测常数1410。表1412解说了控制码的各种可能值。

[0100] 多个地址字可被顺序发送。如果当前控制字是‘00’,则这表示地址字将跟在后面。如果控制码是‘01’,则下一数据字为写数据字。如果控制码是‘01’,则下一数据字为一字读数据字。控制码‘11’被禁止。

[0101] 图15解说了示例性写数据字格式1502。其解说了每个数据字1500包括16比特写数据部分1502。写数据字1500还包括2比特控制码1504和1比特错误检测常数1510。表1514解说了控制码的各种可能值。

[0102] 多个写数据字可被顺序发送。如果当前写字的控制码是‘00’(码元C0),则数据要被写入到先前地址。如果当前写字的控制码是‘01’(码元C1),则数据要被写入到先前地址+1。如果控制码是‘10’(码元E),则下一字将是SID或退出码。

[0103] 图16解说了示例性读规范字格式1600。读规范数据字1600可包括16比特读数据值

部分1604、2比特控制码1608、和3比特错误检测常数1610。

[0104] 在最后一个地址字1607之后, 跟随着“读规范”(RS)字1612。读规范(RS)字1612指定了后面跟随的读数据字的数目。如表1616中解说的, 控制码‘00’被用于指示来自相同地址的读字。控制码‘01’被用于指示来自递增地址的读字。(数据正被从中读取的)从设备将不发送比由“读规范”(RS)字1604指定的数据字更多的数据字(不包括CHK字)。从设备将发送至少一个读字(不包括CHK字)。从设备可在发送由“读规范”(RS)1604字指定的字数目之前结束读取传输。

[0105] 图17解说了示例性读数据字格式1702。读数据字1702可包括16比特读数据值部分1704、2比特控制码1706、和1比特错误检测常数1708。由SID 1707寻址的从设备确定要返回给请求方主设备的字数目。如表1716中解说的, 如果读字继续来自相同地址, 则控制码是“00”(码元R0)。如果读字继续来自递增地址, 则控制码是“01”(码元R1)。如果该字是最后一个读字并且在该字之后没有CHK, 则控制码是“10”(码元E)。控制码“00”被禁止。

[0106] 共享总线上的示例性I2C传输相对于CCIe传输

[0107] 图18解说I2C一个字节写数据操作的示例性时序图。在该示例中, 共享控制数据总线108(图1)包括串行数据线SDA 1802和串行时钟线SCL 1804。图18中解说的传输方案可被称为“I2C模式”。SCL线1804被用于从主设备向所有从设备发送时钟, 而SDA线1802传送数据比特。I2C主设备在SDA线1802中发送7比特从ID 1808以指示主设备希望访问I2C总线上的哪一个从设备, 然后发送指示写操作的1比特。只有其ID与7比特从ID 1808匹配的从设备才能导致预期动作。为了使I2C从设备检测其自己的ID, 主设备必须在SDA线上发送至少8比特(或者在SCL线2204上发送8个时钟脉冲)。

[0108] I2C标准要求所有兼容I2C的从设备在接收到START(开始)状况1806(例如, 在SCL线上为高时由SDA线上的高到低转变指示)之际使其总线逻辑复位。

[0109] CCIe协议将SDA线1802和SCL线1804两者用于数据传输, 而同时将时钟信号嵌入在数据传输内。例如, 数据比特可被转码成随后在诸条线上传送的多个码元。通过将时钟信号(图18中的I2C总线的SCL线)嵌入在码元转变内, SDA线1802和SCL线1804两者均可被用于数据传输。

[0110] 图19解说了其中数据比特已被转码成12个码元以用于在SDA线1902和SCL线1904上进行传输的示例性CCIe传输。图19中解说的传输方案可被称为“CCIe模式”。CCIe模式是源同步的, 由推挽驱动器驱动。在共享控制数据总线上发出数据的任何设备也发出嵌入在数据(例如, 嵌入在码元到码元转变内)的时钟信息。因此, 仅控制数据总线上的一个设备被允许在任一时间驱动共享控制数据总线。

[0111] 为了在同一条总线上支持旧式I2C设备和CCIe设备两者, CCIe模式操作使用相同的START状况1906、1908、1910, 这防止旧式I2C从设备对任何CCIe操作作出反应(例如, CCIe模式期间的START状况使旧式I2C从设备复位)。在此示例中, 在传送全部从ID(即, 全部7比特)之前检测START状况1906、1908、1910(即, 在SCL线1904为高时由SDA线1902上的高到低转变指示), 因此这是不完整的从ID(少于7比特)。如果主设备发送6个SCL脉冲随后发布START状况1906、1908、1910, 则所有旧式I2C从设备在它们将该数据识别为I2C从ID之前使其总线逻辑复位。由于这些6比特序列(例如, 对应于每两个码元)是在两个START状况1906、1908、1910之间发送的, 因而这些6比特序列不被任何I2C从设备解码为有效的从ID。因此,

旧式I2C从设备将不会对不完整的从ID采取动作。

[0112] 在此系统中,主设备控制对总线的访问。因此,希望在控制数据总线上进行传送的任何设备必须例如通过发布中断请求来向主设备请求此类访问。用于发布中断的现有技术机制依赖于专用的中断线或专用的中断总线。然而,此类专用的中断线或中断总线意味着设备必须包括至少一个附加管脚以容适此类中断线或中断总线。为了消除对此类专用的中断管脚和中断线/总线的需要,需要一种用于CCIE内的带内中断的机制。

[0113] 带内中断的使用还应当避免总线争用或冲突。例如,为了避免冲突,当主设备正驱动控制数据总线时,从设备不应被允许驱动控制数据总线(例如,SDA线1802或SCL线1904)以断言IRQ。

[0114] 示例性比特19区域和校验和

[0115] 图20解说了从图2-10中解说的编码方案得到的第20比特(比特19)的示例性映射。如能够领会的,可用的三进制数可用于扩展主设备与从设备之间的特征和能力。例如,比特19内可用的此三进制数空间(即,其比特19为“1”的数据区域)可用于促成或指示:从设备到从设备传输,(b)传输的校验和,(c)至从设备的主操作切换,(d)心跳时钟等。

[0116] 图21解说了图20的第20比特(比特19)区域的示例性映射内的子区域的细节。

[0117] 图22解说了可能发生的各种码元错误状况。时序图2202解说了控制数据总线(SDA线和SCL线)和接收机时钟(RXCLK)上的正确传输。

[0118] 解说了时钟缺失2204,其中接收机时钟(RXCLK)缺少两个循环2212和2214,以使得数据比特2210被不正确地检测。如果在相同传输方向上有更多后续字,则在后续字中很可能检测到字数据错误。同步(SYNC)丢失也可能被检测到。如果在最后一个字上出现错误,则主设备需要超时检测功能性。

[0119] 解说了额外时钟2206,其中接收机时钟(RXCLK)具有在额外时钟循环2220处检测到的额外码元‘01’2216和2218。这种错误很可能在该字或后续字中检测到。同步丢失也可能被检测到。

[0120] 解说了码元错误2208,其中没有接收机时钟(RXCLK)缺失但是出现单个码元错误2222。这种错误很可能在该字或后续字中检测到。校验和错误很可能被检测到。

[0121] 所传送码元内的示例性错误检测

[0122] 图23-28解说了对于各种CCIE字而言可能出现的各种码元错误状况(即,不具有码元滑动的单个码元错误)。如所示出的,这些错误可以通过使用3个比特(比特0、1和2)来检测,如以下进一步讨论的。这些示例将三(3)个最低有效位(比特[2:0])用于错误检测。

[0123] 图23解说了示出所传送的码元序列0321_0321_0321中的可能错误的表2300以及可如何在3个最低有效位内检测此类错误。通过使用图3、4、5、6、7和8中解说的方法来将20比特序列(比特[19:0])0000_0000_0000_0000_00002302转换成三进制数(T11...T0)0000_0000_0000₃ 2304,该三进制数随后被转换成顺序码元(S11...S0)0321_0321_0321 2306。出于此示例的目的,该3个最低有效位2308均为0(000)。如果在传输期间在原始顺序码元0321_0321_0321 2306中的任何码元处引入了错误,则这将导致错误的码元2310。例如,如果最后一个码元“1”被改变成“0”,则这会导致3个最低有效位从“000”改变成“010”。如果最后一个码元“1”被改变成“3”,则这会导致3个最低有效位从“000”改变成“001。”如果第一个码元“0”被改变成“2”,则这会导致3个最低有效位从“000”改变成“100”。表2300解说了只要

3个最低有效位是已知常数(例如,固定常数“000”)则可如何通过这3个最低有效位来检测任何单个码元的改变的各种其他示例。

[0124] 图24解说了示出所传送的码元序列2301_2301_2301中的可能错误的表2400以及可如何在3个最低有效位内检测此类错误。通过使用图3、4、5、6、7和8中解说的方法来将20比特序列(比特[19:0])0100_0000_1101_1111_10002402转换成三进制数($T_{11}\cdots T_0$)1111_1111_1111₃ 2404,该三进制数随后被转换成顺序码元($S_{11}\cdots S_0$)2301_2301_2301 2406。出于此示例的目的,该3个最低有效位2408均为0(000)。如果在传输期间在原始顺序码元2301_2301_2301 2406中的任何码元处引入了错误,则这将导致错误的码元2410。例如,如果最后一个码元“1”被改变成“3”,则这会导致3个最低有效位从“000”改变成“111。”如果最后一个码元“1”被改变成“2”,则这会导致3个最低有效位从“000”改变成“001”。如果第一个码元“2”被改变成“0”,则这会导致3个最低有效位从“000”改变成“100”。表2400解说了只要3个最低有效位是已知常数(例如,固定常数“000”)则可如何通过这3个最低有效位来检测任何单个码元的改变的各种其他示例。

[0125] 图25解说了示出所传送的码元序列3131_3131_3131中的可能错误的表2500以及可如何在3个最低有效位内检测此类错误。通过使用图3、4、5、6、7和8中解说的方法来将20比特序列(比特[19:0])1000_0001_1011_1111_00002502转换成三进制数($T_{11}\cdots T_0$)2222_2222_2222₃ 2504,该三进制数随后被转换成顺序码元($S_{11}\cdots S_0$)3131_3131_3131 2506。出于此示例的目的,该3个最低有效位2508均为0(000)。如果在传输期间在原始顺序码元3131_3131_3131 2406中的任何码元处引入了错误,则这将导致错误的码元2510。例如,如果最后一个码元“1”被改变成“0”,则这会导致3个最低有效位从“000”改变成“111”。如果最后一个码元“1”被改变成“2”,则这会导致3个最低有效位从“000”改变成“100”。如果第一个码元“3”被改变成“0”,则这会导致3个最低有效位从“000”改变成“001”。表2500解说了只要3个最低有效位是已知常数(例如,固定常数“000”)则可如何通过这3个最低有效位来检测任何单个码元的改变的各种其他示例。

[0126] 图26解说了示出所传送的码元序列0132_3101_3231中的可能错误的表2600以及可如何在3个最低有效位内检测此类错误。通过使用图3、4、5、6、7和8中解说的方法来将20比特序列(比特[19:0])0001_1000_1111_0011_10002602转换成三进制数($T_{11}\cdots T_0$)0120_1201_2012₃ 2604,该三进制数随后被转换成顺序码元($S_{11}\cdots S_0$)0132_3101_3231 2606。出于此示例的目的,该3个最低有效位2608均为0(000)。如果在传输期间在原始顺序码元0132_3101_3231 2606中的任何码元处引入了错误,则这将导致错误的码元2610。例如,如果最后一个码元“1”被改变成“0”,则这会导致3个最低有效位从“000”改变成“111”。如果最后一个码元“1”被改变成“2”,则这会导致3个最低有效位从“000”改变成“110”。如果第一个码元“0”被改变成“3”,则这会导致3个最低有效位从“000”改变成“111”。表2600解说了只要3个最低有效位是已知常数(例如,固定常数“000”)则可如何通过这3个最低有效位来检测任何单个码元的改变的各种其他示例。

[0127] 图27解说了示出所传送的码元序列2030_2120_3021中的可能错误的表2700以及可如何在3个最低有效位内检测此类错误。通过使用图3、4、5、6、7和8中解说的方法来将20比特序列(比特[19:0])0100_1010_1101_1010_10002702转换成三进制数($T_{11}\cdots T_0$)1201_2012_0120₃ 2704,该三进制数随后被转换成顺序码元($S_{11}\cdots S_0$)2030_2120_3021 2706。出

于此示例的目的,该3个最低有效位2708均为0(000)。如果在传输期间在原始顺序码元2030_2120_3021_2706中的任何码元处引入了错误,则这将导致错误的码元2710。例如,如果最后一个码元“1”被改变成“0”,则这会导致3个最低有效位从“000”改变成“010”。如果第一个码元“2”被改变成“0”,则这会导致3个最低有效位从“000”改变成“011”。表2700解说了只要3个最低有效位是已知常数(例如,固定常数“000”)则可如何通过这3个最低有效位来检测任何单个码元的改变的各种其他示例。

[0128] 图28解说了示出所传送的码元序列3231_0132_3101中的可能错误的表2800以及可如何在3个最低有效位内检测此类错误。通过使用图3、4、5、6、7和8中解说的方法来将20比特序列(比特[19:0])0101_1110_1101_0000_10002802转换成三进制数(T11...T0)2012_0120_1201₃ 2804,该三进制数随后被转换成顺序码元(S11...S0)3231_0132_3101_2806。出于此示例的目的,该3个最低有效位2808均为0(000)。如果在传输期间在原始顺序码元3231_0132_3101_2806中的任何码元处引入了错误,则这将导致错误的码元2810。例如,如果最后一个码元“1”被改变成“3”,则这会导致3个最低有效位从“000”改变成“111”。如果第一个码元“3”被改变成“0”,则这会导致3个最低有效位从“000”改变成“100”。表2800解说了只要3个最低有效位是已知常数(例如,固定常数“000”)则可如何通过这3个最低有效位来检测任何单个码元的改变的各种其他示例。

[0129] 用于错误检测的第一示例性办法

[0130] 如图23-28中所提及的,使用20比特CCIE字的3个最低有效位来检测不具有时钟缺失或码元滑动的任何单个码元错误是可能的。

[0131] 图29解说了CCIE字2902可如何将3个最低有效位2903或最低有效位2905用于错误检测。对于第一CCIE字2902(例如,20比特序列),3个最低有效位2903可被设置成固定常数(例如,“000”),并且如果所接收到的20比特序列具有不同于该固定常数(例如,“000”)的3个最低有效位,则在接收方设备处检测到错误。通过使用这种办法不会遗漏CCIE字中(例如,该字中的12个码元中的任何一个码元中或者20比特(从十六进制0x00000到0x81BF0)的比特[19:0])中的任何比特中)的不具有码元滑动的单个码元错误,由此提供对任何此类单个码元错误的100%检测。然而,将二十(20)比特中的三(3)个比特用于错误检测对于许多实现而言可能过于昂贵。

[0132] 替换办法提供了将单个比特(例如,最低有效位0)而不是3个最低有效位用于错误检测。在约一半情形中,可以使用固定的比特0(例如,固定为“0”)来帮助检测码元错误(不具有时钟缺失或码元滑动)。除了使用固定的比特0之外,顺序CCIE字的比特1和2的校验和可被计算(参见图31)并且被发送给接收方设备以查明是否在所传送的CCIE字中已出现错误。

[0133] 对于第二CCIE字2904(例如,20比特序列),最低有效位2905可被设置成固定常数(例如,“0”)。如从图23-28可领会的,当在相应的码元序列中存在码元错误时,最低有效位2905在约一半的时间发生改变。因此,对于在最低有效位处反应的错误,如果序列中的最低有效位不是固定常数(例如,“0”),则接收方设备可成功地查明在所接收到的20比特序列中存在单个码元错误(不具有时钟缺失或码元滑动)。

[0134] 除了将最低有效位2905用于检测或者结合将最低有效位2905用于检测,校验和消息可被用于提供对CCIE字序列的较高程度的错误检测。此类校验和消息未检测到错误的唯

一情景为：在计算校验和时两个码元错误的效果抵消。

[0135] 图30解说了可如何在CCIE字内实现8比特校验和。在此示例中，校验和字3005可（在20比特序列内）通过将比特19设置成‘1’并且将比特12设置成‘0’而同时在CCIE地址字3004a、写字3004b和/或读字3004c的CCIE帧内放置校验和值来标示（例如，每个字的比特4-11可被用于校验和）。8比特校验和值3003可以在若干CCIE字传输上被计算。例如，此校验和字3005可以在CCIE字传输序列的末尾处被发送。

[0136] 由于比特19，此校验和字可以在可用的附加空间内定义（参见图20）。校验和字3005可被插入在写命令3006和/或读命令3010的地址或数据部分内。例如，在写命令3006中，一个或多个地址字3008a、3008b和/或一个或多个写字3008c和3008d可被用于传达校验和值（例如，从主设备向从设备）。类似地，在读命令3010中，一个或多个地址字3012a、3012b和/或一个或多个读字3012c和3012d可被用于传达校验和值。

[0137] 与其他校验和办法形成对比，可以由发送方在传输内的任何地方和任何时间插入当前CCIE校验和字3005。

[0138] 图31解说了8比特校验和生成器的示例性实现。校验和生成器3102可包括用于校验和3104的8个比特中的每一个比特的1比特寄存器。第一CCIE字N 3110的比特2和1与校验和3104中的最低有效位异或(XOR)。随后，第二CCIE字N+1 3112的比特2和1与校验和3104中的下一最低有效位异或(XOR)。随后，第三CCIE字N+2 3114的比特2和1与校验和3104中的下一最低有效位异或(XOR)。随后，第四CCIE字N+3 3116的比特2和1与校验和3104中的最高有效位异或(XOR)。在达到校验和值3104的最高有效位之后，在下一CCIE字N+4上，校验和生成器可在校验和3104的两个最低有效位处再次启动。

[0139] 以此方式，来自不同CCIE字的贡献跨8比特校验和传播其使用。这些1比特寄存器可在传输的每个开始处（例如，在SID之前和/或在读字的开始之前）被初始化。为了实现CCIE校验和字，传送方设备可在发送每个CCIE字时计算校验和值。在接收机设备处，也以相同的方式为每个接收到的CCIE字计算校验和值。随后，发射机设备可向接收机设备发送具有最新近校验和值的CCIE校验和字，接收机设备可将该校验和值与其自己的校验和值进行比较以查明是否已发生任何码元错误。

[0140] 校验和字3005可跟随在包括以下各项的任何CCIE字之后：(a)CHK字（重复CHK字）和/或(b)心跳/SYNC字。CCIE发射机和接收机可支持校验和功能性。例如，CCIE发射机可发送CCIE接收机随后使用的校验和字3005。

[0141] 如果接收机设备检测到校验和(CHK)错误，则：(a)该接收机设备可挂起CCIE动作（例如，寄存器写）直至SYNC字或WAKEUP（苏醒）事件，(b)该接收机设备可经由IRQ报告CHK错误，(c)该接收机设备可在状态寄存器中设置CHK错误事件，和/或(d)该接收机设备可忽略校验和错误。

[0142] 校验和(CHK)字可被插入在任何字边界处。这利用了可在不同码元/协议内容容易携带的校验和以及可缩放的数据质量（例如，在较不可靠的总线上使用较多CHK）。

[0143] 在一些实现中，校验和(CHK)支持可以不是强制性的。这为低等级产品创建了较少负担并且还在不需要使用校验和的情况下允许平滑折旧。

[0144] 用于错误检测的第二示例性办法

[0145] 检测具有图29-31中解说的格式的字中的码元错误并不始终可靠并且对于一些系

统而言可能是有问题的。在要求高数据完整性的总线系统(例如,诸如医疗、汽车、航天或军事使用)中,往往希望立即检测所有码元错误。为了解决这个问题,公开了一种替换CCIE字格式,其中可达到100%码元错误检测(即,在不具有码元滑动的情况下每字单个码元错误)。为此,首先应当注意,本文公开的各方面部分地依赖于图23-28中的观察,其中可以在3个最低有效位(LSB)均固定具有已知常数的情况下传送经编码字时达到100%码元错误检测。然而,利用3个LSB来表示错误比特检测常数要求这3个LSB是灵活的,以使得它们的比特值可被控制。

[0146] 例如,在图30中的CCIE校验和字3005中,比特1和2被分配给CCIE字的控制码。此类控制码标识字包含读操作还是写操作并且不能在不影响CCIE字的操作的情况下被修改。

[0147] 一种解决方案是修改图30中解说的CCIE字格式,以使得3个LSB可表示3比特错误检测常数。此类修改的示例在图13-15中提供,其实现了可按需优化数据吞吐量或错误检测的灵活的比特分配方案。

[0148] 图32解说了促成错误检测的示例性灵活CCIE字格式3203。在此20比特字3202中,比特[0]被分配给错误检测常数3204;比特[2,1]被分配给数据或者扩展错误检测常数3206;比特[4,3]被分配给控制比特3208;比特[18,5]被分配给14比特数据字段3210;并且比特[19]被分配给先前提及的“额外”比特3212。即,相对于其中比特[2,1]可被用于不能被用于错误检测的控制码3228的替换CCIE字格式3220,灵活的CCIE字3202可将这两个控制比特从比特[2,1]移走,这提供了将比特[2,1]用于数据以优化数据吞吐量(即,通过经由比特[18,5]和比特[2,1]来为数据分配16比特)或者将比特[2,1]用于错误检测常数扩展以优化错误检测(即,通过为错误检测常数分配3个LSB以达到前面提及的100%错误检测)的灵活性。用户可由此按需在以非100%码元错误检测发送具有16比特数据的经编码字或者以100%码元错误检测发送具有14比特数据的经编码字之间进行选择。

[0149] 应当注意,各种比特分配方案中的任一者均被构想落在本文所公开的各方面的范围内。例如,尽管图32解说了将比特[4,3]分配给控制比特以使它们对于此特定示例而言更靠近LSB(即,而不是将它们分配得更靠近较高值的MSB),但是其他比特分配方案可包括将这些控制比特放置在比特[18,3]内的任何比特组合中。还应当注意,尽管图32解说了20比特CCIE字格式,但是比特分配方案被类似地构想用于任何比特长度的字,其中分配给数据或错误检测常数扩展的最低有效位的固定数目可以适当地增加/减少。

[0150] 此外,本文公开的各方面可促成第一比特分配方案与第二比特分配方案之间的动态切换,其中第一比特分配方案涉及数据优化(即,为数据分别16比特),并且其中第二比特分配方案涉及错误检测优化(即,为错误检测常数分配3个LSB)。

[0151] 示例性时钟/码元滑动错误检测

[0152] 由时钟缺失或额外时钟造成的码元滑动错误可能不被错误检测常数和/或校验和检测到。然而,这些类型的错误中的大多数可在下一码字处检测和/或使用接收机设备处的状态机来检测。

[0153] 图35解说了不具有时钟错误的示例性CCIE字传输。使用包括第一线(SDA)3502和第二线(SCL)3504的共享总线,在开始状况3508和3510之间传送12个码元3512(构成一CCIE字)。此类开始状况3508和3510在第二线(SCL)3504为高时在第一线(SDA)3502上的高到低转变上发生。所有遵从CCIE的设备将此理解为在可由接收机设备的状态机跟踪的预期CCIE

字边界处的开始状况(即,在第二线(SCL)3504为高时的第一线(SDA)3502上的高到低转变)。另外,还解说了嵌入在码元到码元转变内的时钟3506。

[0154] 图36解说了具有单个时钟缺失错误的示例性CCIE字传输。即,在此示例中,时钟脉冲3602未被接收机设备检测到,这导致相应的码元“2”被忽略。此状况3604可被称为码元滑动。因为接收机设备错过时钟,所以该接收机设备将继续从第一和第二线3502和3504读取码元并且将第二开始状况3510解读为码元,以及将预期下一码元3602为开始状况。此时,时钟缺失错误由接收机设备检测到,因为下一码元3602将决不会是开始状况。接收机设备可随后假定同步丢失并且使其接收机逻辑复位。

[0155] 图37解说了具有两个缺失时钟错误的示例性CCIE字传输。即,在此示例中,两个时钟脉冲3702未被接收机设备检测到,这导致码元“2”和“3”3704被忽略。此状况可被称为双码元滑动。因为接收机设备错过两个时钟,所以该接收机设备将继续从第一和第二线3502和3504读取码元并且将第二开始状况3510解读为码元。对于9种情形中的8种情形,接收机设备逻辑将识别出预期的开始状况缺失(在下一字3702的第一和第二码元之间),由此检测到双时钟缺失。接收机设备可包括用于跟踪何时预期开始状况的状态机逻辑。对于下一字3702的第一和第二码元之间的转变将被检测为开始状况(即,3到1码元转变)的1/9几率,接收机设备可在下一字上检测到此双时钟错误(即,在下一CCIE字中的某一点处,第一码元与第二码元之间的转变将不是开始状况并且错误被接收机设备检测到)。

[0156] 图38解说了具有额外时钟错误的示例性CCIE字传输。即,在此示例中,额外时钟脉冲3802被接收机设备检测到,这导致额外的码元“1”被读取。在此状况3804下,接收机设备将在最后一个有效码元转变3806期间预期下一开始状况。在9种情形中的8种情形中,CCIE字的最后两个码元中的码元组合将不会导致开始状况,所以接收机设备将识别出同步错误。在9种情形中的1种情形中,CCIE字的最后两个码元将被接收机设备识别为开始状况(即,3到1码元转变)。对于该情况,接收机设备可依赖当前和/或下一CCIE字的错误检测常数来检测这个错误。

[0157] 具有灵活错误检测的示例性设备

[0158] 图33是解说根据本公开提供的主/从设备的示例性组件的框图。如所解说的,主/从设备3314经由控制数据总线3350耦合至主/从设备3360。此处,构想主/从设备3314或主/从设备3360可根据本文公开的前述方面作为主设备或从设备来操作,并且主/从设备3314和主/从设备3360可具有基本上相似的组件。

[0159] 在该示例中,主/从设备3314可被实现成具有由总线3302一般化地表示的内部总线架构。取决于主/从设备3314的具体应用和整体设计约束,总线3302可包括任何数目的互连总线和桥接器。总线3302将包括一个或多个处理器(一般地由处理电路3304表示)、存储器3305和计算机可读介质(一般地由计算机可读介质3306表示)的各种电路链接在一起。总线3302还可链接各种其它电路,诸如定时源、外围设备、稳压器和功率管理电路,这些电路在本领域中是众所周知的,且因此将不再进一步描述。总线接口/电路3308提供控制数据总线3350与主/从设备3314之间的接口。

[0160] 在本公开的一方面,计算机可读介质3306被配置成包括各种指令3306a、3306b和/或3306c以促成如本文公开的CCIE协议通信。在类似方面,此类通信可取而代之经由硬件通过将处理电路3304耦合至电路3320、3330和/或3340中的任一者来实现,如所示出的。替换

地,处理电路3304可包括和/或实现电路3320、3330和/或3340中的任一者。此外,构想了CCIE通信可通过指令3306a、3306b和/或3306c的任何组合以及电路3320、3330和/或3340的任何组合来促成。

[0161] 例如,编码器/解码器指令3306a和编码器/解码器电路3320涉及根据CCIE协议来编码/解码各个字。如先前提及的(参见图2-21),此类编码/解码可包括在逐数位基础上将三进制数转换成多个码元(例如,12位三进制数导致12个码元)以产生前述“额外比特”(比特19)。

[0162] 在本公开的另一方面,错误检测指令3306b和错误检测电路3330涉及执行如图23-32中描述和解说错误检测和/或根据期望的CCIE字格式来执行错误检测。为此,如图31-32中那样,本文公开的所构想的CCIE字格式包括经编码的20比特字,其中3个最低有效位被分配用于数据或错误检测。此外,构想了错误检测指令3306b和/或错误检测电路3330中任一者可被构造成促成灵活的比特分配,其中最低有效位被分配用于错误检测,并且其中第二最低有效位和第三最低有效位中的每一者被分配用于数据或错误检测。如图23-30和35-38中那样,进一步构想了错误检测指令3306b和/或错误检测电路3330中的任一者可被配置成通过使用固定的最低有效位和/或结合校验和字被添加到协议,和/或通过使用检测缺失的开始状况的接收机设备逻辑来促成错误检测。

[0163] 在本公开的另一方面,校验和指令3306c和/或校验和电路3340可被配置成为从总线3350接收和/或传送的字生成校验和。具体而言,校验和指令3306c和/或校验和电路3340中的任一者可被配置成促成校验和计算以支持CCIE校验和字。

[0164] 应当领会,处理电路3304负责管理总线3302(即,当设备3314是主设备时)和一般处理,包括存储在计算机可读介质3306上的软件的执行。软件在由处理电路3304执行时使用主/从设备3314执行以下针对任何特定装置描述的各种功能。计算机可读介质3306还可被用于存储由处理电路3304在执行软件时操纵的数据。

[0165] 处理电路3304中的一个或多个处理器可执行软件。软件应当被宽泛地解释成意为指令、指令集、代码、代码段、程序代码、程序、子程序、软件模块、应用、软件应用、软件包、例程、子例程、对象、可执行件、执行的线程、规程、函数等,无论其是用软件、固件、中间件、微代码、硬件描述语言、还是其他术语来述及皆是如此。软件可驻留在计算机可读介质3306上。计算机可读介质3306可以是非瞬态计算机可读介质。作为示例,非瞬态计算机可读介质包括:磁存储设备(例如,硬盘、软盘、磁条)、光盘(例如,压缩碟(CD)或数字多功能碟(DVD))、智能卡、闪存设备(例如,记忆卡、记忆棒、或钥匙驱动器)、随机存取存储器(RAM)、只读存储器(ROM)、可编程ROM(PROM)、可擦式PROM(EPROM)、电可擦式PROM(EEPROM)、寄存器、可移动盘、以及任何其他用于存储可由计算机访问和读取的软件和/或指令的合适介质。作为示例,计算机可读介质还可包括承载、传输线、与任何其他用于传送可由计算机访问与读取的软件与/或指令的合适介质。计算机可读介质3306可以驻留在主/从设备3314中、在主/从设备3314外部、或跨包括该主/从设备3314在内的多个实体分布。计算机可读介质3306可以实施在计算机程序产品中。作为示例,计算机程序产品可包括封装材料中的计算机可读介质。本领域技术人员将认识到如何取决于具体应用和加诸于整体系统上的总体设计约束来最佳地实现本公开中通篇给出的所描述的功能性。

[0166] 图34解说了根据协议(例如,CCIE协议)的一个或多个错误检测特征来促成通信的

示例性方法。获得要在共享总线上传送的多个比特,其中该多个比特中的3个最低有效位或最低有效位可被用于错误检测,3402。该多个比特被转换成三进制数,3404。三进制数的各个数位被转换成顺序码元,其中时钟被嵌入在码元到码元转变内,3406。多个码元被组合成字以供传输,3408。该字可随后在共享总线上传送给接收方设备,3410。另外,可为所传送比特计算校验和,3412。所传送比特的校验和可被发送给接收方设备,3414。

[0167] 根据一个方面,提供了一种包括总线接口和处理电路的发射机设备。总线接口可用于将发射机设备耦合至共享总线(接收方设备或从设备耦合至该共享总线)。发射机设备可管理或控制共享总线上的通信。该处理电路可被配置成:(a)获得要在共享总线上传送的多个比特,其中该多个比特中的3个最低有效位或最低有效位被用于错误检测;(b)将该多个比特转换成三进制数;(c)将该三进制数的各个数位转换成顺序码元;(d)将多个码元组合成字以供传输;和/或(e)在共享总线上向接收方设备传送该字。时钟可被嵌入在码元到码元转变内。共享总线可以是双线总线,并且该双线总线中的两条线均被用于传输码元。该处理电路可以被进一步配置成:(a)计算所传送比特的校验和;和/或(b)向接收方设备传送所传送比特的校验和。该校验和可被插入在任何的任意字内以供传送给接收方设备。

[0168] 根据另一方面,提供了一种接收机设备,包括:总线接口和处理电路。总线接口可用于耦合至主设备在其上管理通信的共享总线。该处理电路可被配置成:(a)在共享总线上接收多个码元;(b)将该多个码元转换成各个数位;(c)将这些数位组合成三进制数;(d)将该三进制数转换成比特,其中该多个比特中的3个最低有效位或最低有效位被用于错误检测;和/或(e)基于该多个比特中的3个最低有效位或最低有效位来查明所接收到的多个比特内的错误。时钟可被嵌入在多个收到码元的码元到码元转变内。共享总线可以是双线总线,并且该双线总线中的两条线均被用于传输码元。接收机设备可实现状态机逻辑电路,该状态机逻辑电路检测时钟信号中的错误的时钟脉冲缺失和错误的额外时钟脉冲。

[0169] 附图中解说的组件、步骤、特征、和/或功能之中的一个或多个可以被重新编排和/或组合成单个组件、步骤、特征、或功能,或可以实施在数个组件、步骤或功能中。还可添加附加的元件、组件、步骤、和/或功能而不会脱离本文中所公开的新颖特征。附图中所图解的装置、设备和/或组件可以被配置成执行在这些附图中所描述的方法、特征、或步骤中的一个或多个。本文中描述的新颖算法还可以高效地实现在软件中和/或嵌入在硬件中。

[0170] 另外应注意,这些实施例可能是作为被描绘为流程图、流图、结构图、或框图的过程来描述的。尽管流程图可能会把诸操作描述为顺序过程,但是这些操作中有许多操作能够并行或并发地执行。另外,这些操作的次序可以被重新安排。过程在其操作完成时终止。过程可对应于方法、函数、规程、子例程、子程序等。当过程对应于函数时,它的终止对应于该函数返回调用方函数或主函数。

[0171] 此外,存储介质可以代表用于存储数据的一个或多个设备,包括只读存储器(ROM)、随机存取存储器(RAM)、磁盘存储介质、光学存储介质、闪存设备、和/或其他用于存储信息的机器可读介质。术语“机器可读介质”包括,但不被限定于,便携或固定的存储设备、光学存储设备、无线信道以及能够存储、包含或承载指令和/或数据的各种其它介质。

[0172] 此外,诸实施例可以由硬件、软件、固件、中间件、微代码、或其任何组合来实现。当在软件、固件、中间件、或微码中实现时,执行必要任务的程序代码或代码段可被存储在诸如存储介质之类的机器可读介质或其它存储中。处理器可以执行这些必要的任务。代码段

可表示规程、函数、子程序、程序、例程、子例程、模块、软件包、类,或是指令、数据结构、或程序语句的任何组合。通过传递和/或接收信息、数据、自变量、参数、或存储器内容,一代码段可被耦合到另一代码段或硬件电路。信息、自变量、参数、数据等可以经由包括存储器共享、消息传递、令牌传递、网络传输等的任何合适的手段被传递、转发、或传输。

[0173] 结合本文中公开的示例描述的各个解说性逻辑块、模块、电路、元件和/或组件可用设计成执行本文中描述的功能的通用处理器、数字信号处理器(DSP)、专用集成电路(ASIC)、现场可编程门阵列(FPGA)或其他可编程逻辑组件、分立的门或晶体管逻辑、分立的硬件组件、或其任何组合来实现或执行。通用处理器可以是微处理器,但在替换方案中,该处理器可以是任何常规的处理器、控制器、微控制器、或状态机。处理器还可以实现为计算组件的组合,例如DSP与微处理器的组合、数个微处理器、与DSP核心协作的一个或多个微处理器、或任何其他此类配置。

[0174] 结合本文中公开的示例描述的方法或算法可直接在硬件中、在能由处理器执行的软件模块中、或在这两者的组合中以处理单元、编程指令、或其他指示的形式实施,并且可包含在单个设备中或跨多个设备分布。软件模块可驻留在RAM存储器、闪存、ROM存储器、EPROM存储器、EEPROM存储器、寄存器、硬盘、可移动盘、CD-ROM、或本领域中所知的任何其他形式的存储介质中。存储介质可耦合到处理器以使得该处理器能从/向该存储介质读写信息。替换地,存储介质可以被整合到处理器。

[0175] 本领域技术人员将可进一步领会,结合本文中公开的实施例描述的各种解说性逻辑块、模块、电路、和算法步骤可被实现为电子硬件、计算机软件、或两者的组合。为清楚地解说硬件与软件的这一可互换性,各种解说性组件、块、模块、电路、和步骤在上面是以其功能性的形式作一般化描述的。此类功能性是被实现为硬件还是软件取决于具体应用和施加于整体系统的设计约束。

[0176] 本文所述的本发明的各种特征可实现于不同系统中而不脱离本发明。应注意,以上实施例仅是示例,且不应被解释成限定本发明。这些实施例的描述旨在是说明性的,而非旨在限定权利要求的范围。由此,本发明的教导可以现成地应用于其他类型的装置,并且许多替换、修改和变形对于本领域技术人员将是显而易见的。

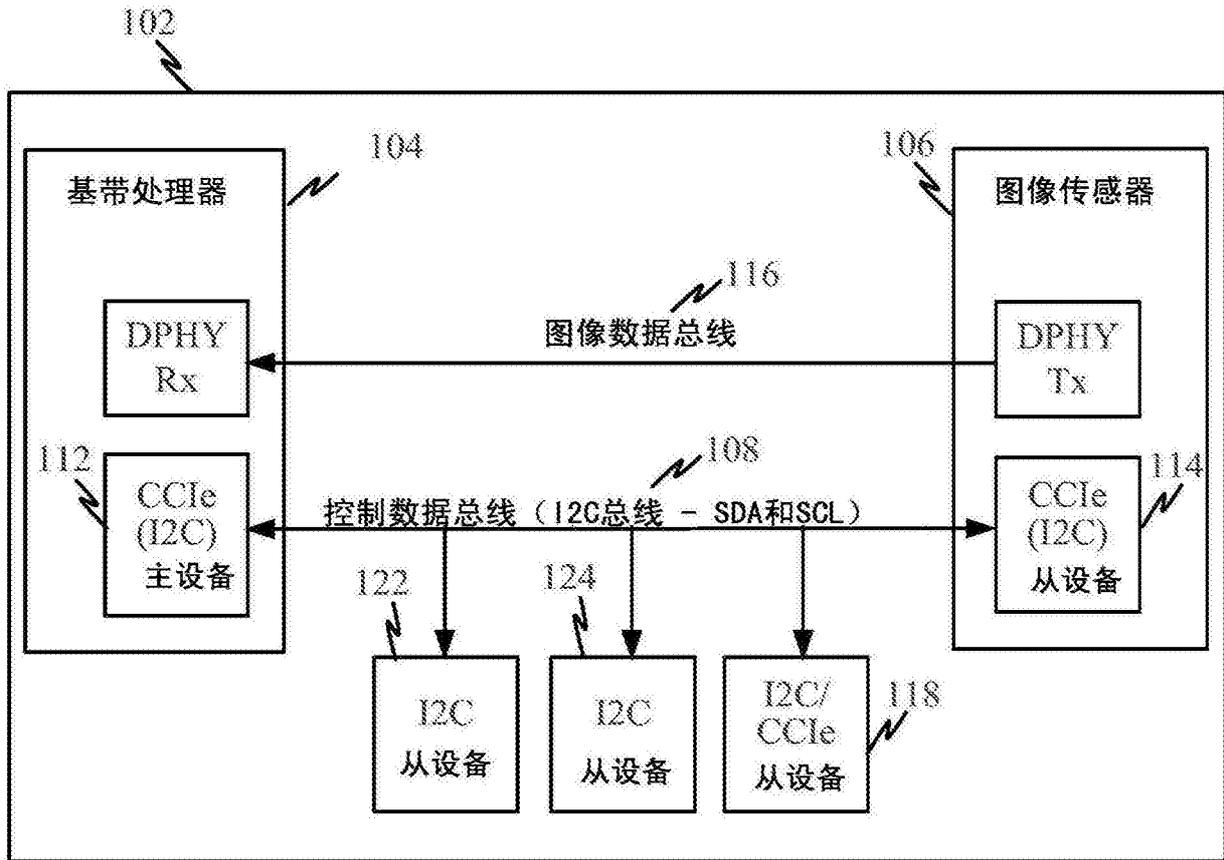


图1

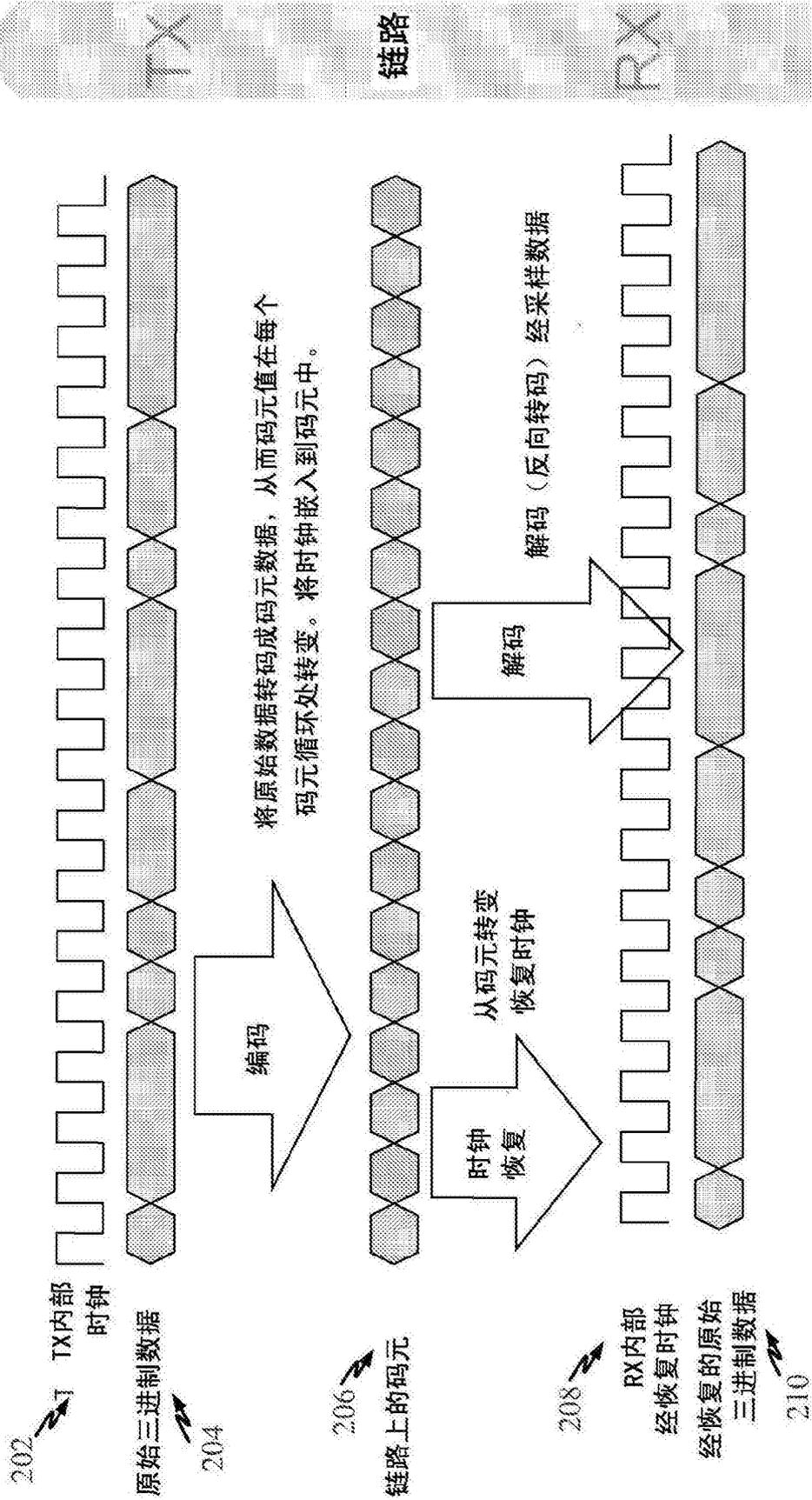


图2

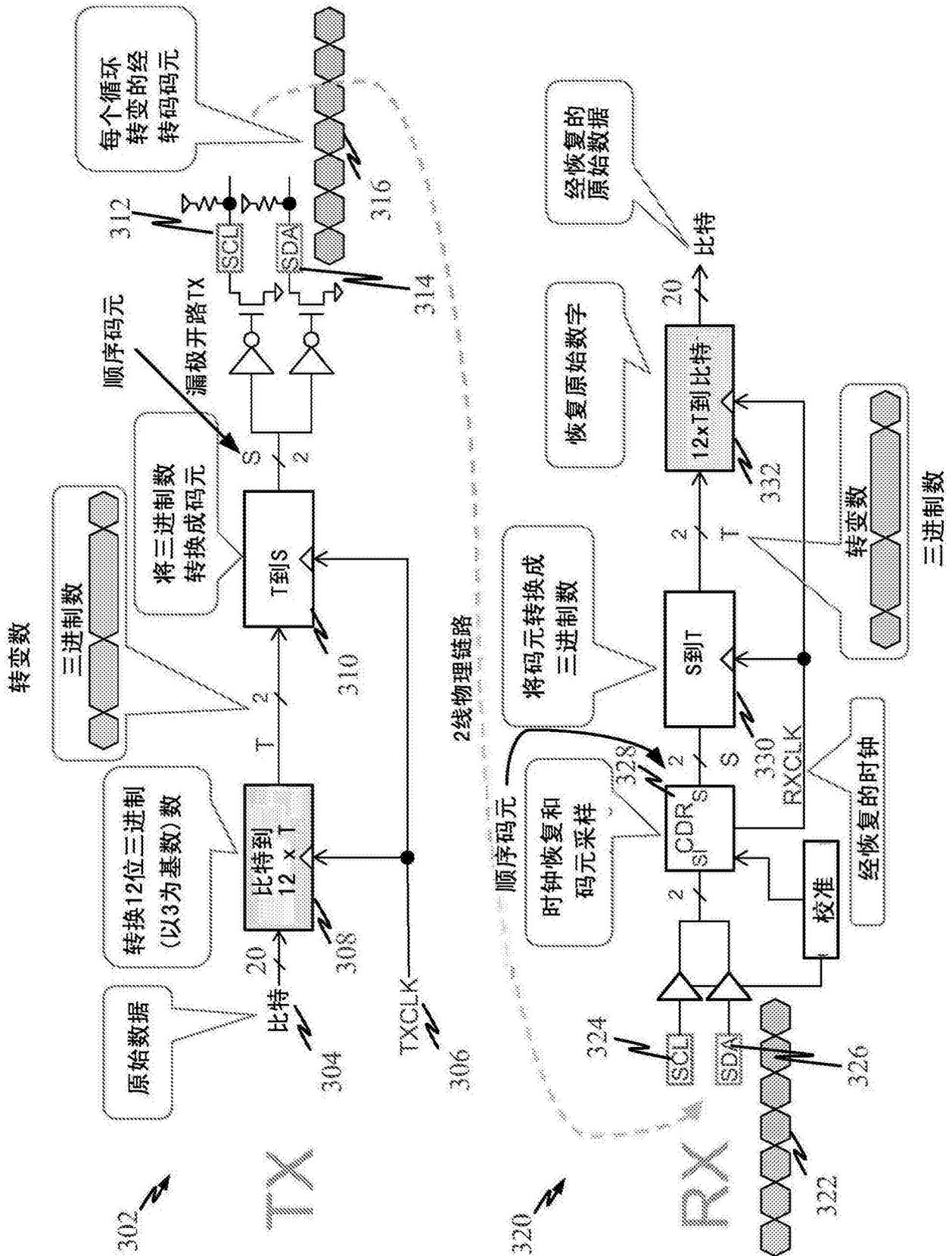


图3

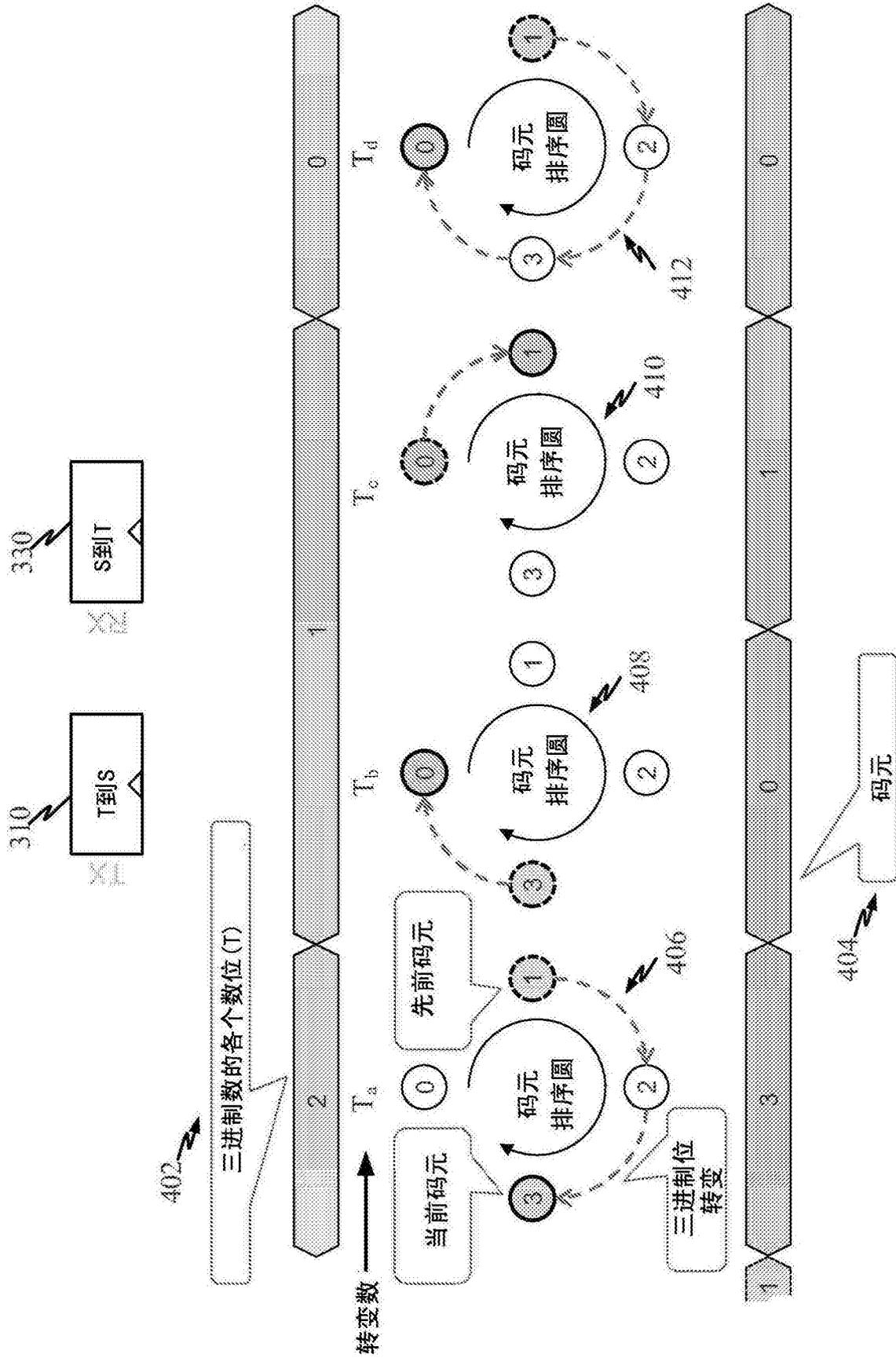


图4

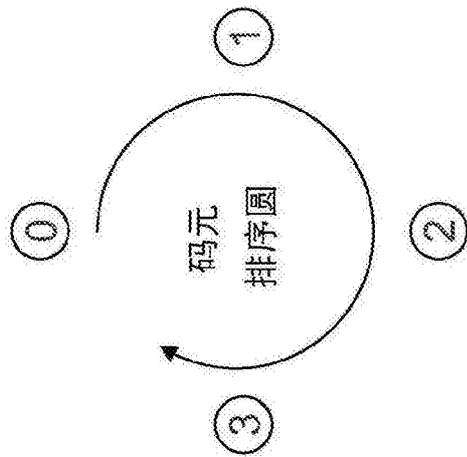
506 ↗

给每个码元转变指派数字: T

在CCle系统中, 存在4个可能的原始码元: S=0, 1, 2和3
 在两个原始码元的任何序列中, 不可能出现相同的码元

→ 每个转变有3种可能情形
 将顺序索引指派给每个转变

先前码元 Ps	当前码元 Cs	转变数 T
0	1	1
	2	2
	3	0
1	2	1
	3	2
	0	0
2	3	1
	0	2
	1	0
3	1	1
	2	2
	0	0



502 ↗ TX: T 到 S

$$T_{tmp} = T == 0 ? 3 : T$$

$$Cs = Ps + T_{tmp}$$

504 ↗ RX: S 到 T

$$T_{tmp} = 4 + Cs - Ps$$

$$T = T_{tmp} == 3 ? 0 : T_{tmp}$$

图5

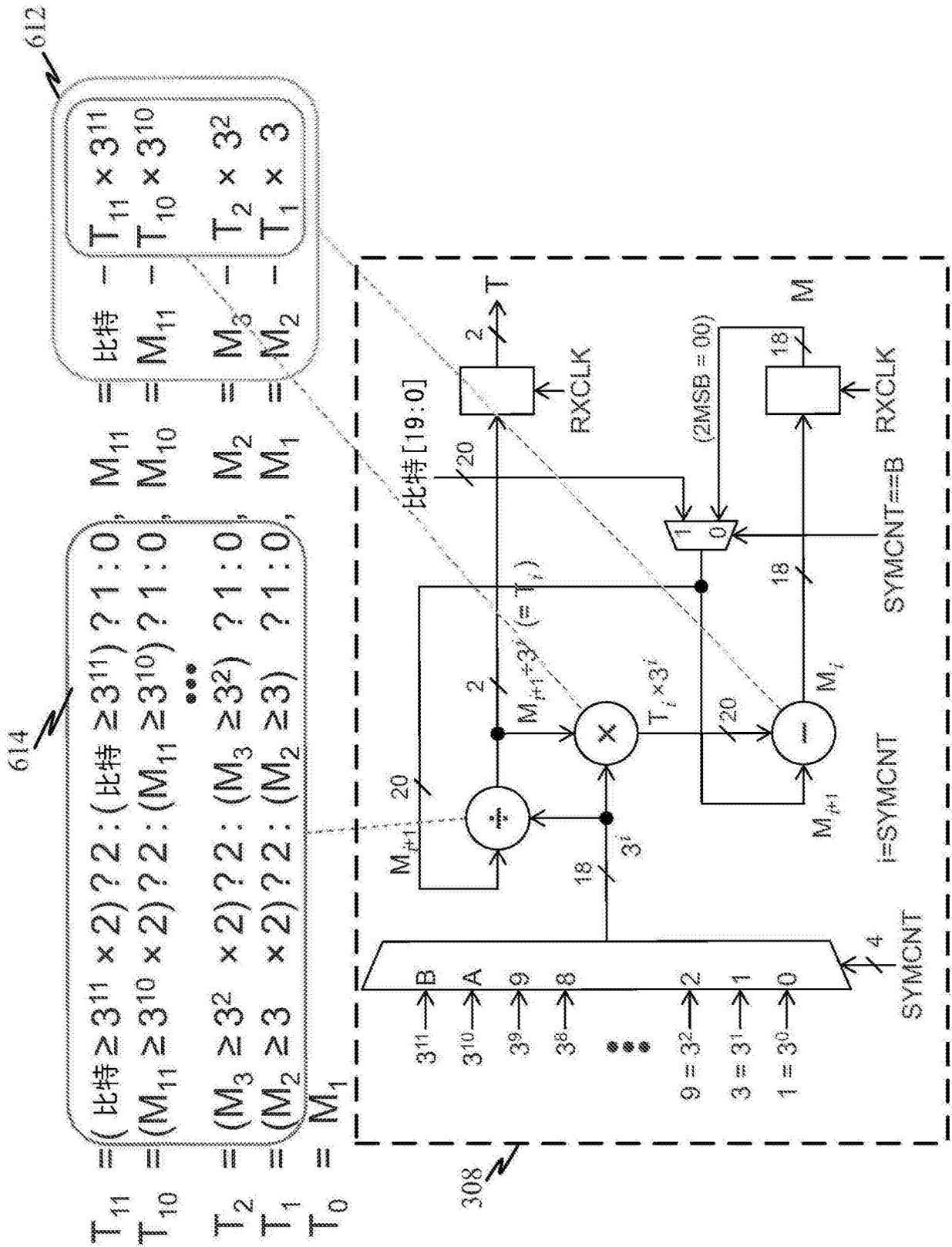
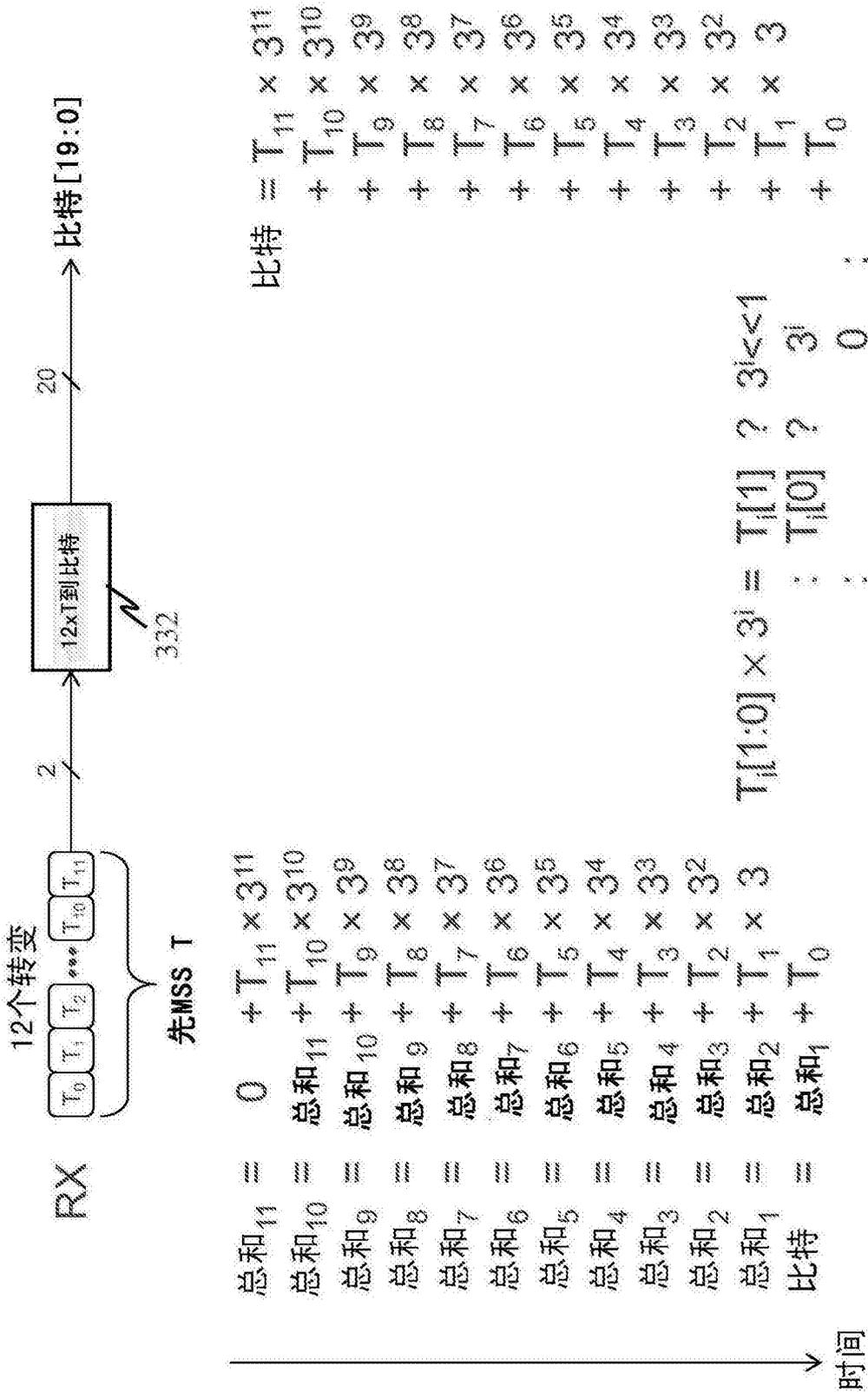


图7



T_i是2比特三进制数位，并且唯一可能的值为00, 01或10(不是11)

T_i × 3ⁱ不需要任何加法器(仅MUX操作)。

图8

先MSS CClc: 12xT到比特电路

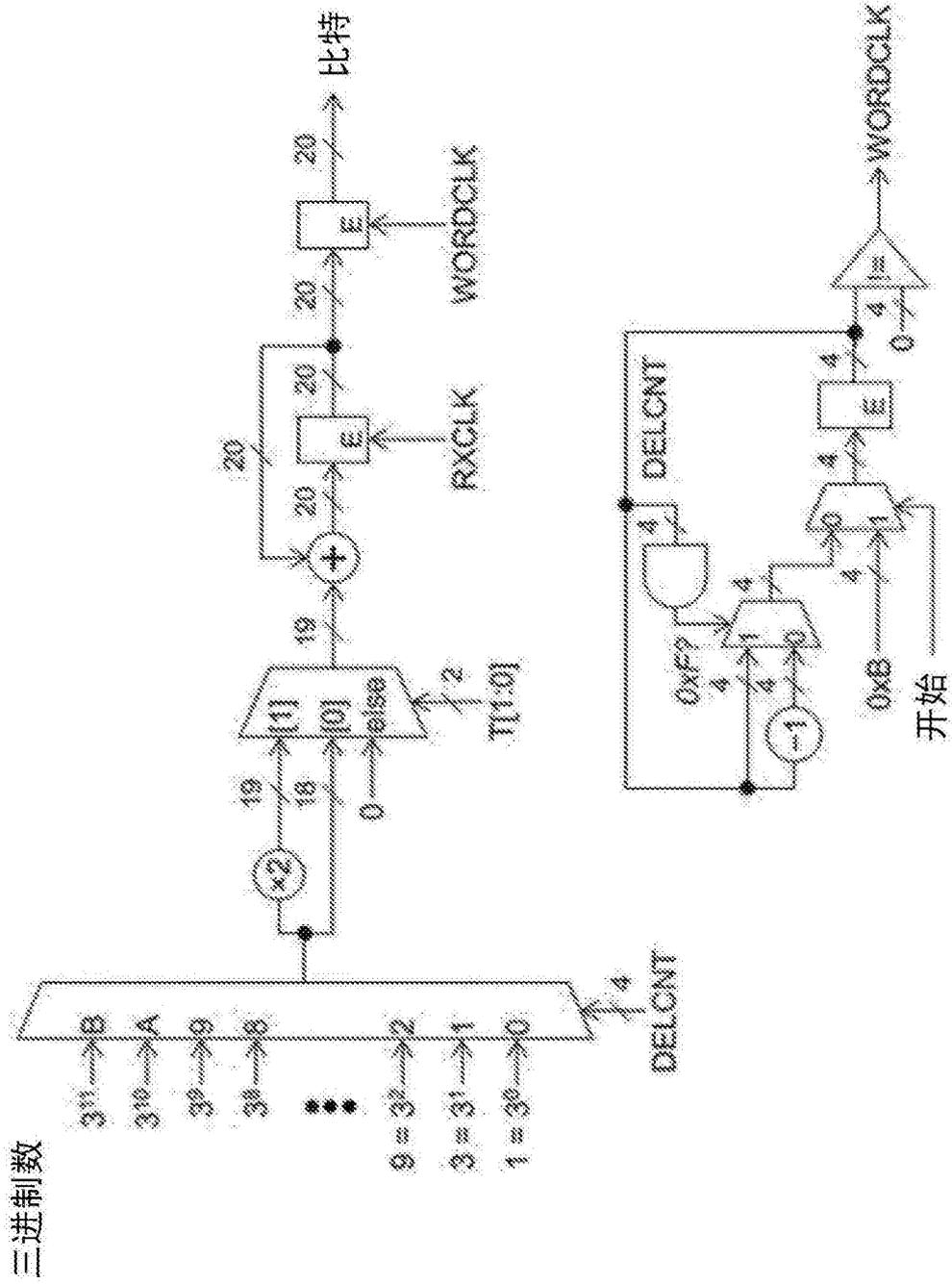


图9

比特19 (第二十比特)

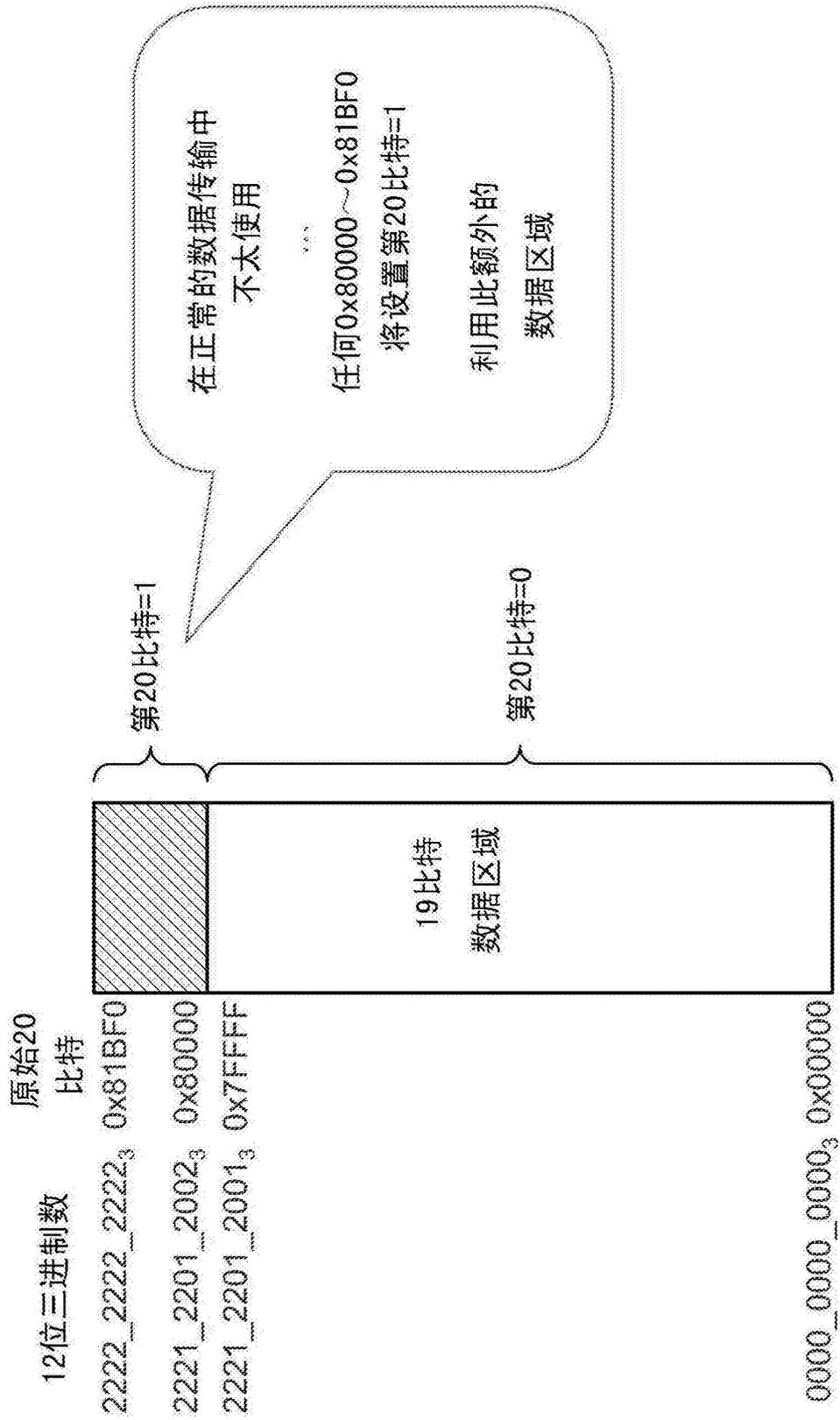
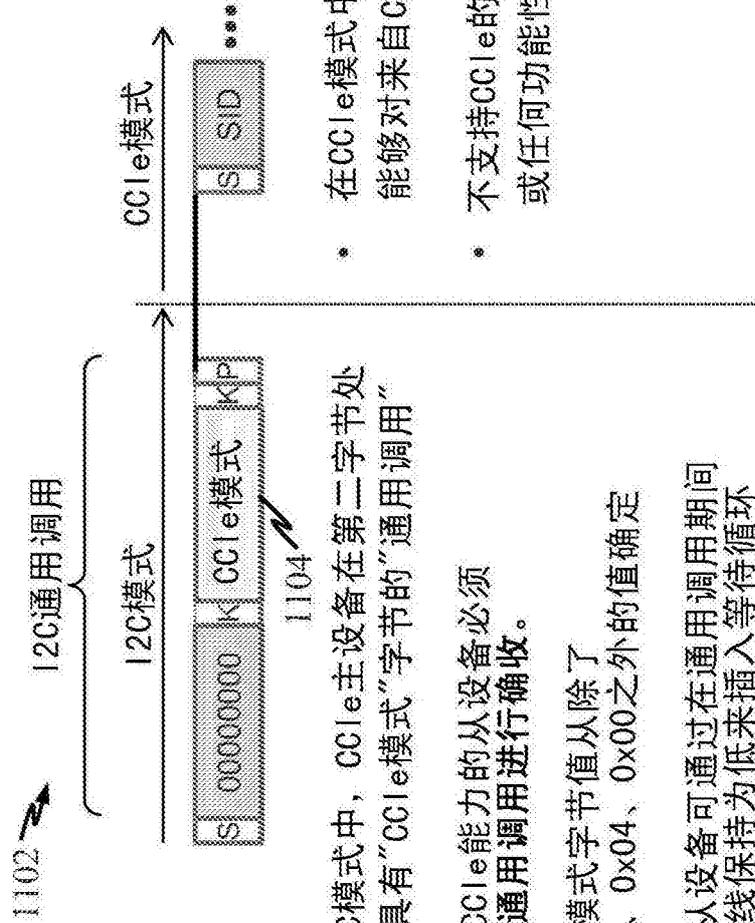


图10

来自CCle模式的示例性进入指示符

具有CCle能力的主设备发布具有第二字节0x?? (将在稍后确定的码)的I2C“通用调用” (SID=0x00)以向所有具有CCle能力的从设备指示从I2C模式到CCle模式的转变



- 在I2C模式中，CCle主设备在第二字节处发布具有“CCle模式”字节的“通用调用”
- 具有CCle能力的从设备必须对该通用调用进行确收。
- CCle模式字节值从除了0x06、0x04、0x00之外的值确定
- CCle从设备可通过在通用调用期间将SCL线保持为低来插入等待循环(若必要)
- 在CCle模式中，所有CCle设备必须能够对来自CCle主设备的请求作出响应
- 不支持CCle的旧式I2C从设备的操作状态或任何功能性应当不受任何CCle事务影响

图11

来自CC1e模式的示例性退出指示符

CC1e主设备发布“退出”码替代CC1e SID

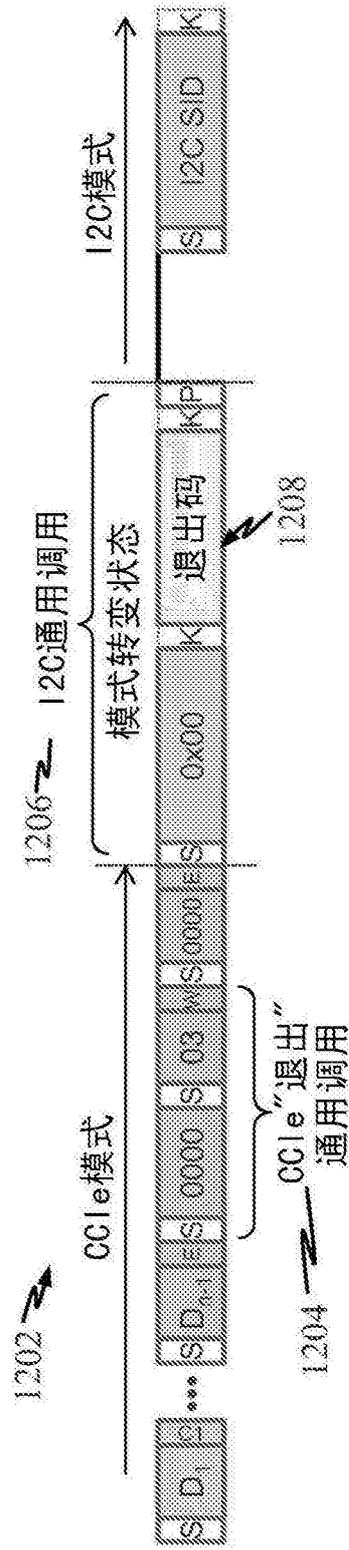
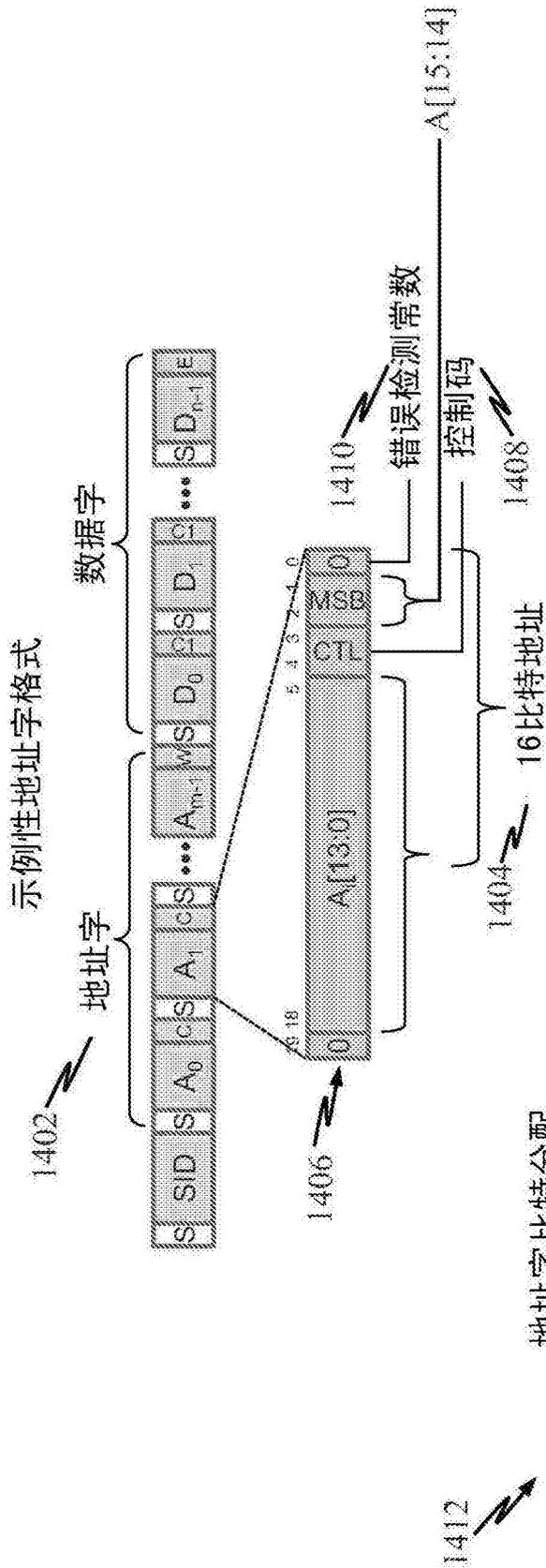


图12

- 在CC1e模式中，在CC1e模式中后面跟随S的最后一个数据之后，CC1e主设备发送CC1e“退出”通用调用以指示CC1e模式的结束并且转变回到I2C模式
- 在CC1e“退出”通用调用之后，CC1e主设备发送后面跟随I2C通用调用的S(开始状况)，其中该I2C通用调用在第二字节处具有“退出”码
- 所有具有CC1e能力的从设备必须对I2C通用调用进行确收



- 地址字由16比特地址值、2比特控制码、和1比特错误检测常数(0)构成

- 多个地址字可顺序发送。在当前地址字的控制码为00时将有序后续地址字

- 如果控制码是01, 则下一数据字为写数据

- 如果控制码是'01', 则下一数据字为1字读规范(RS)

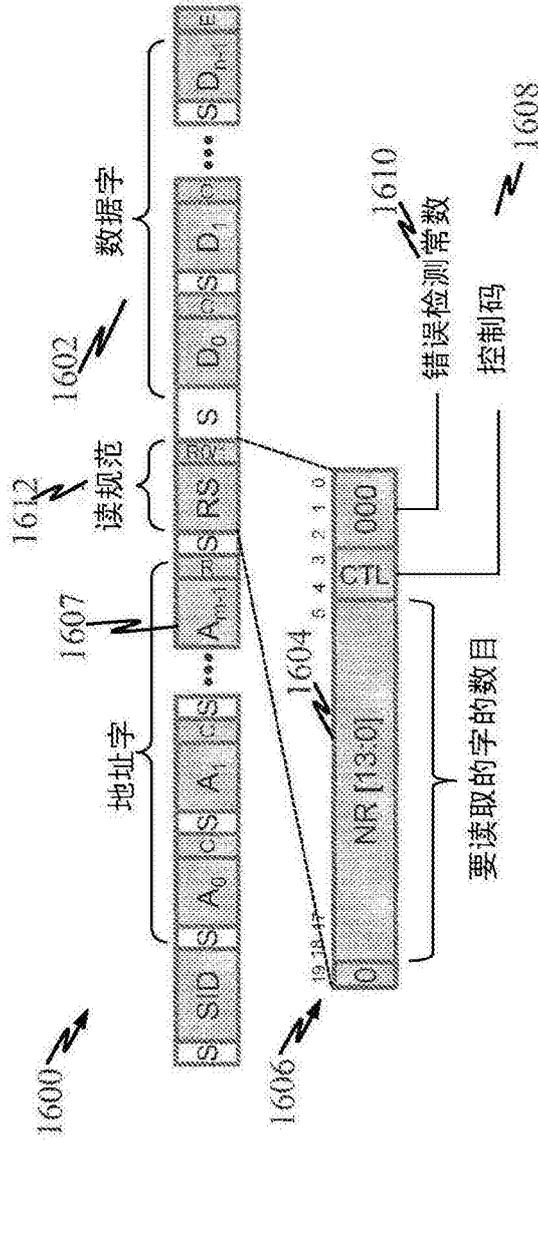
- 控制码11被禁止(保留用于SID)

CTL[1:0] 比特分配

B[2:1]	码元	描述
00	C	继续到下一地址字
01	W	写
10	RS	后面跟随读规范字
11	-	禁止

图14

示例性读规范格式



RS字比特分配

比特字段	值	描述
19	0	
18:5	NR[13:0]	读取的字的数目
4:3	CTL[1:0]	控制码(参见以下)
2:0	000	错误检测常数

CTL[1:0] 比特分配

CTL[1:0] 码元	描述
00	从相同地址读取字
01	从递增地址读取字
10	保留(或继续?)
11	禁止

- RS(读规范)字定义后面跟随的读数据字序列的性质
 - NR [13:0] 要读取的字的数目
 - 其他性质?
- 当NR[13:0]=0x0000时, 读字数目是无限的, 并且读取继续直至从设备以CTL=10(E)结束读取
- 当控制码是"00"时, 从相同地址A读取字
- 当控制码是"01"时, 从A开始的递增地址读取字

图16

示例性兼容 I2C 协议的传输

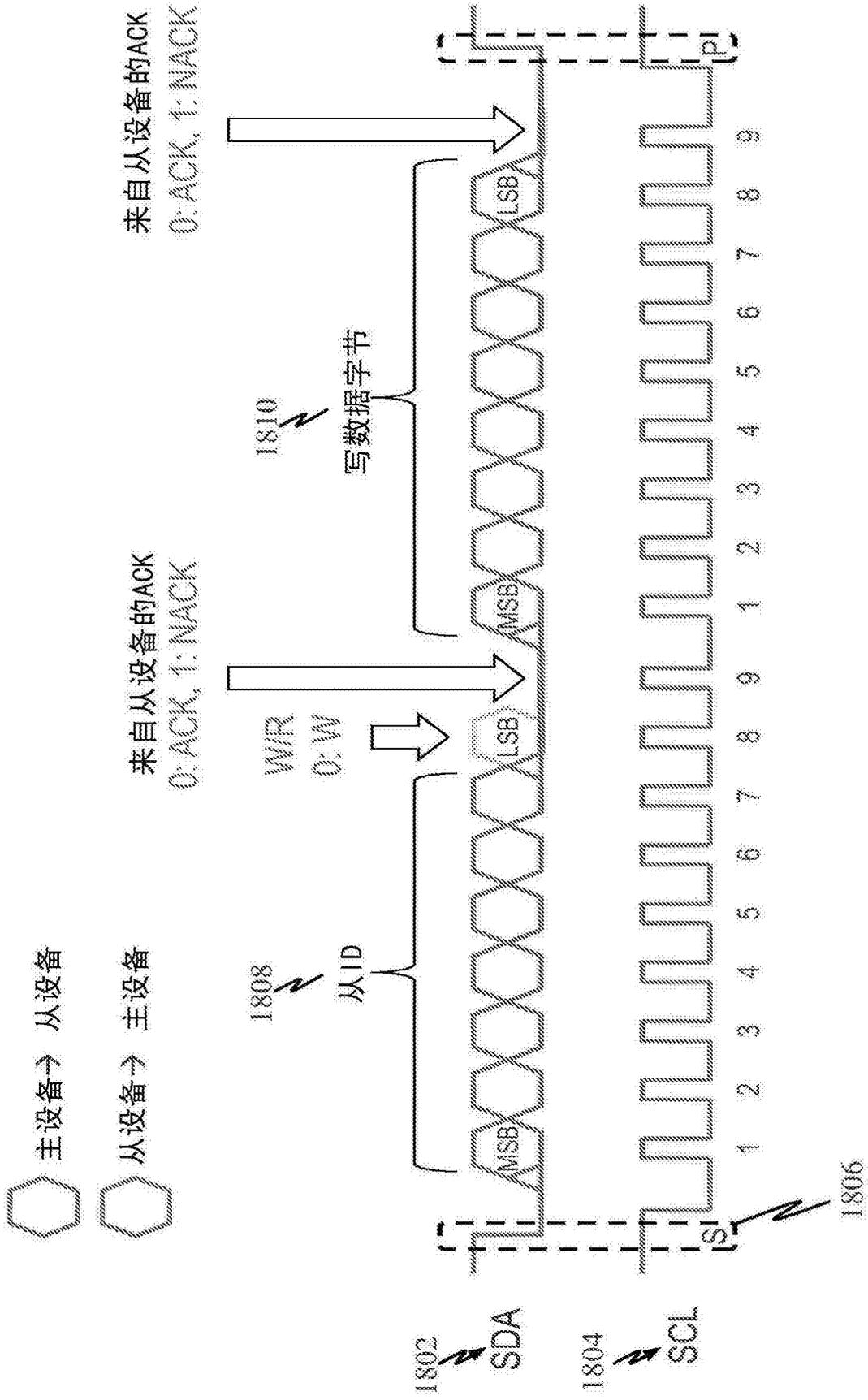


图18

示例性兼容CC1e协议的传输

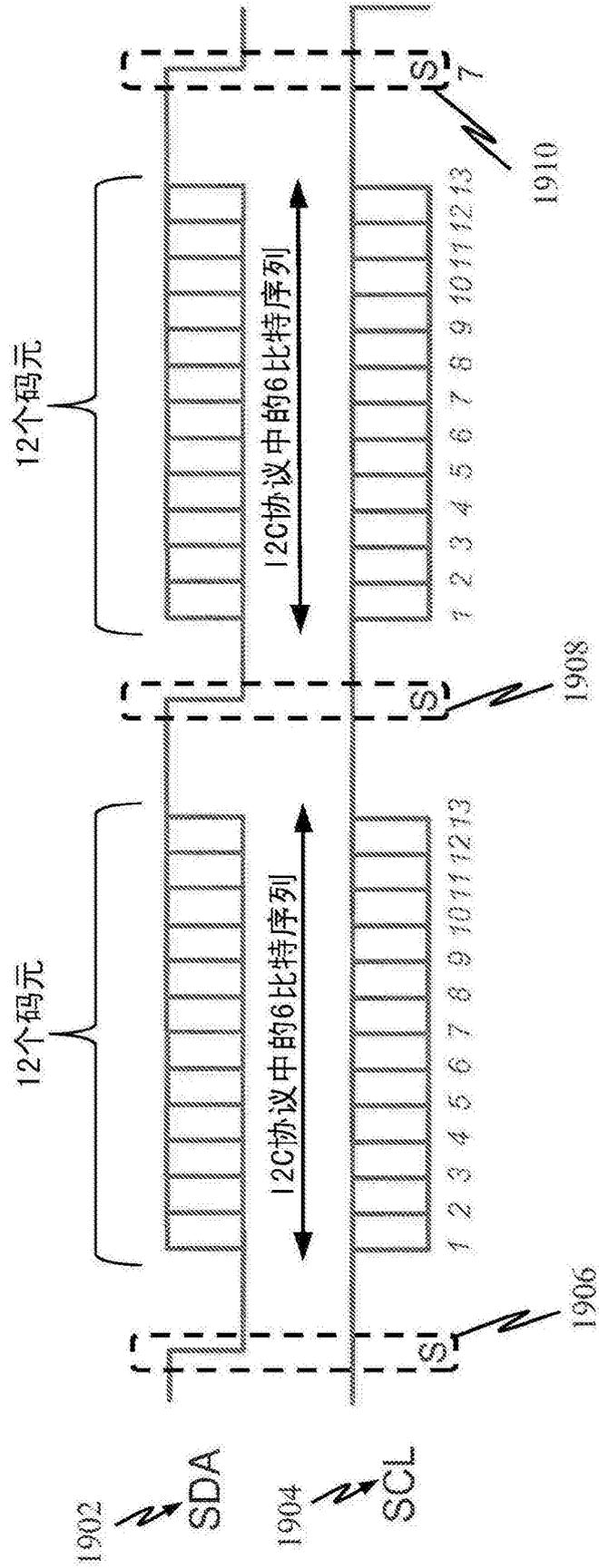


图19

CCle 比特19映射

三进制 比特[19:0]	地址	写	读	比特[i]																		
				19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3		
2222_2222_2222 ₃ 0x81BF0 0x31 (49) ↓	参见图21			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
2222_2222_1012 ₃ 0x81BC0				1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2222_2222_1011 ₃ 0x81BBF [6比特] 0x40 (64) ↓				1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2222_2221_1211 ₃ 0x81B80				1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2222_2221_1210 ₃ 0x81B7F [7比特] 0x80 (128) ↓				保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	
2222_2220_0002 ₃ 0x81B00				保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	
2222_2220_0001 ₃ 0x81AFF [8比特] 0x100 (256) ↓				保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	
2222_2202_2121 ₃ 0x81A00				保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	
2222_2202_2120 ₃ 0x819FF [9比特] 0x200 (512) ↓				保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	
2222_2112_1122 ₃ 0x81800				保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	
2222_2112_1121 ₃ 0x817FF [11比特] 0x800 (2048) ↓	CCle 寄存器地址	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留	保留				
2222_1121_0210 ₃ 0x81000	8比特 CHK	8比特 CHK	8比特CHK	1	0	x	x	x	x	x	x	x	x	x	x	x	x	x				
2222_1121_0202 ₃ 0x80FFF [12比特] 0x1000 (4096) ↓	8比特 CHK	8比特 CHK	8比特CHK	1	0	x	x	x	x	x	x	x	x	x	x	x	x	x				
2222_2201_2002 ₃ 0x80000	8比特 CHK	8比特CHK	8比特CHK	1	0	x	x	x	x	x	x	x	x	x	x	x	x	x				
2221_2201_2001 ₃ 0x7FFFF 0x80000 (524288) ↓	19比特数据区域			0	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
0000_0000_0000 ₃ 0x00000				0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		

i=[6]~19

图20

CCle比特19映射继续

三进制	比特[19:0]	地址	写	读	位[i]																		
					19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
2222	2222_2222 ₃	0x81BF0	禁止	禁止	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0		
2222	2222_2221 ₃	0x81BEF	禁止	禁止	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0		
	0x2A (42) ↓		禁止	禁止																			
2222	2222_1102 ₃	0x81BC6	禁止	禁止																			
2222	2222_1101 ₃	0x81BC5	禁止	禁止	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0		
2222	2222_1100 ₃	0x81BC4	禁止	禁止																			
	0x2A (42) ↓		禁止	禁止																			
2222	2222_1000 ₃	0x81BBE	保留	保留	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0		
2222	2222_0222 ₃	0x81BBA	保留	保留	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0		
2222	2222_0221 ₃	0x81BB9	保留	保留	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0		
2222	2222_0220 ₃	0x81BB8	保留	保留	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0		
2222	2222_0212 ₃	0x81BB7	保留	保留	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0		
	[3比特] 0x8 (8) ↓		保留	保留																			
2222	2222_0121 ₃	0x81BB0	保留	保留	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0		
2222	2222_0120 ₃	0x81BAF	保留	保留	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0		
	[4比特] 0x10 (16) ↓		保留	保留																			
2222	2222_0000 ₃	0x81BA0	保留	保留	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0		
2222	2221_2222 ₃	0x81B9F	禁止	禁止																			
	0x10 (16) ↓		禁止	禁止																			
2222	2221_2102 ₃	0x81B90	禁止	禁止	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0		
2222	2221_2101 ₃	0x81B8F	禁止	禁止	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0		
2222	2222_2100 ₃	0x81B8E	禁止	禁止																			
	0xA (10) ↓		禁止	禁止																			
2222	2221_2000 ₃	0x81B85	保留	保留	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0		
2222	2221_1222 ₃	0x81B84	保留	保留	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0		
2222	2221_1221 ₃	0x81B83	保留	保留	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0		
	[2比特] 0x4 (4) ↓		保留	保留																			
2222	2221_1211 ₃	0x81B80	保留	保留	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0		

图21

导致错误类型

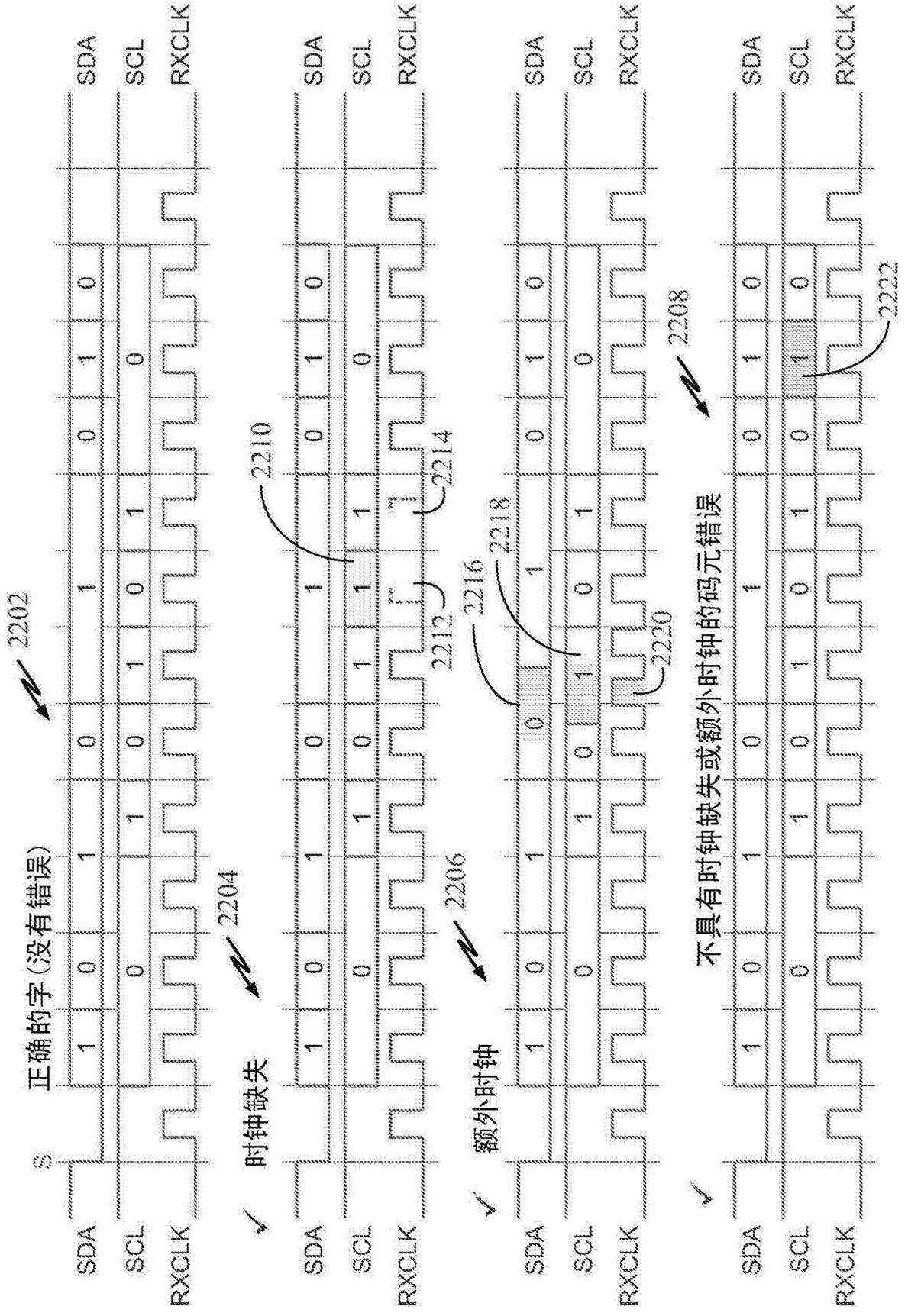


图22

码元错误学习

当单个码元错误在CCle字
=0000_0000_00003 (0x00000) 上出现时

2300

2306

2304

2302 2308

错误位置	S11..S0(码元)	T11..T0 (转变)	十六进制	比特[19:0]	比特[2:0]
良好	0321 0321 0321	0000 0000 0000	000000	0000 0000 0000 0000 0000	000
T00^1	0321 0321 0321	0000 0000 0002	000002	0000 0000 0000 0000 0010	010
T00^2	0321 0321 0321	0000 0000 0001	000001	0000 0000 0000 0000 0001	001
T01^2	0321 0321 0301	0000 0000 0011	000004	0000 0000 0000 0000 0100	100
T02^2	0321 0321 0121	0000 0000 0110	00000c	0000 0000 0000 0000 1100	110
T03^2	0321 0321 2321	0000 0000 1100	000024	0000 0000 0000 0010 0100	100
T04^2	0321 0321 0321	0000 0001 1000	00006c	0000 0000 0000 0110 1100	100
T05^2	0321 0301 0321	0000 0011 0000	001044	0000 0000 0001 0100 0100	100
T06^2	0321 0121 0321	0000 0110 0000	0030cc	0000 0000 0011 1100 1100	100
T07^2	0321 2321 0321	0000 1100 0000	00b64	0000 0000 1011 0110 0100	100
T08^2	0321 0321 0321	0001 1000 0000	02222c	0000 0011 0011 0011 1100	100
T09^2	0301 0321 0321	0011 0000 0000	06684	0000 0110 0110 1000 0100	100
T10^2	0121 0321 0321	0110 0000 0000	1338c	0001 0011 0011 1000 1100	100
T11^2	2321 0321 0321	1100 0000 0000	39aa4	0011 1001 1010 1010 0100	100

2310

不导致时钟缺失的
单线错误

码元错误的阶数越高，
则突发错误越多

3个LSB检测所有
单个码元错误

图23

不具有时钟缺失(不具有码元滑动)的示例性码元错误
 当单个码元错误在 $001e_3 = 1111_1111_1111_3$ (0x40DF8) 上出现时

错误位置	S11..S0 (码元)	T11..T0 (转变)	十六进制	比特 [19:0]	比特 [2:0]
良好	2301 2301 2301	1111 1111 1111	40df8	0100 0000 1101 1111 1000	000
T00^2	2301 2301 2301	1111 1111 1110	40df7	0100 0000 1101 1111 0111	111
T00^3	2301 2301 2301	1111 1111 1112	40df9	0100 0000 1101 1111 1001	001
T01^2	2301 2301 2321	1111 1111 1100	40df4	0100 0000 1101 1111 0100	100
T02^2	2301 2301 2101	1111 1111 1001	40dec	0100 0000 1101 1110 1100	100
T03^2	2301 2301 0301	1111 1111 0011	40dd4	0100 0000 1101 1101 0100	100
T04^2	2301 2301 2301	1111 1110 0111	40d8c	0100 0000 1101 1000 1100	100
T05^2	2301 2321 2301	1111 1100 1111	40cb4	0100 0000 1100 1011 0100	100
T06^2	2301 2101 2301	1111 1001 1111	40a2c	0100 0000 1010 0010 1100	100
T07^2	2301 0301 2301	1111 0011 1111	40294	0100 0000 0010 1001 0100	100
T08^2	2301 2301 2301	1110 0111 1111	3ebcc	0011 1110 1011 1100 1100	100
T09^2	2321 2301 2301	1100 1111 1111	3a774	0011 1010 0111 0111 0100	100
T10^2	2101 2301 2301	1001 1111 1111	2da6c	0010 1101 1010 0110 1100	100
T11^2	0301 2301 2301	0011 1111 1111	07354	0000 0111 0011 0101 0100	100

图24

不具有时钟缺失(不具有码元滑动)的示例性码元错误
 当无时钟缺失的单个码元错误在0C1e字= 2222_2222_2222₃ (0x81BF0) 上出现时

错误位置	S11..S0 (码元)	T11..T0 (转变)	十六进制	比特 [19:0]	比特 [2:0]
良好	3131 3131 3131	2222 2222 2222	81bf0	1000 0001 1011 1111 0000	000
T00^1	3131 3131 3131	2222 2222 2221	81bef	1000 0001 1011 1110 2221	221
T00^3	3131 3131 3131	2222 2222 2220	81bee	1000 0001 1011 1110 1110	110
T01^1	3131 3131 3121	2222 2222 2210	81beb	1000 0001 1011 1110 1011	011
T01^3	3131 3131 3101	2222 2222 2201	81be9	1000 0001 1011 1110 1001	001
T02^1	3131 3131 3031	2222 2222 2102	81be1	1000 0001 1011 1110 0002	002
T02^3	3131 3131 3231	2222 2222 2012	81bdb	1000 0001 1011 1100 2012	012
T03^1	3131 3131 2131	2222 2222 1022	81bc3	1000 0001 1011 1100 0012	012
T03^3	3131 3131 0131	2222 2222 0122	81bb1	1000 0001 1011 1011 0001	001
T04^1	3131 3130 3131	2222 2221 0222	81b69	1000 0001 1011 0110 1002	002
T04^3	3131 3132 3131	2222 2220 1222	81b33	1000 0001 1011 0211 0011	011
T05^1	3131 3121 3131	2222 2210 2222	81a5b	1000 0001 1010 0101 1011	011
T05^3	3131 3101 3131	2222 2201 2222	819b9	1000 0001 1001 1011 1001	001
T06^1	3131 3031 3131	2222 2102 2222	81731	1000 0001 0111 0211 0001	001
T06^3	3131 3231 3131	2222 2012 2222	8154b	1000 0001 0101 0100 0012	012
T07^1	3131 2131 3131	2222 1022 2222	80db3	1000 0000 1101 1011 0011	011
T07^3	3131 0131 3131	2222 0122 2222	80801	1000 0000 1000 0000 0001	001
T08^1	3130 3131 3131	2221 0222 2222	7ff139	0121 1111 0001 0011 2001	001
T08^3	313 3131 3131	2220 1222 2222	7e023	0111 1110 0002 0010 0011	011
T09^1	3121 3131 3131	2210 2222 2222	79bcb	0111 1001 1011 1100 1011	011
T09^3	3101 3131 3131	2201 2222 2222	76889	0111 0110 1000 1000 1001	001
T10^1	3031 3131 3131	2102 2222 2222	69b81	0100 1001 1011 1000 0001	001
T10^3	3031 3131 3131	2012 2222 2222	601bb	0110 0000 0001 1011 1011	011
T11^1	2131 3131 3131	1022 2222 2222	39aa3	0011 1001 1010 1010 0011	011
T11^3	0131 3131 3131	0122 2222 2222	1cd51	0001 1100 1101 0101 0001	001

图25

2600

不具有时钟缺失(不具有码元滑动)的示例性码元错误

当单个码元错误在001e字= 0120_1201_2012₃ (0x18F38)上出现时

2606

2604

2602 2608

错误位置	S11..S0(码元)	T11..T0(转变)	十六进制	比特[19:0]	比特[2:0]
良好	0132 3101 3231	0120 1201 2012	18f38	0001 1000 1111 0011 1000	000
T00^1	0132 3101 3231	0120 1201 2011	18f37	0001 1000 1111 0011 0111	111
T00^3	0132 3101 3231	0120 1201 2010	18f36	0001 1000 1111 0011 0110	110
T01^3	0132 3101 3201	0120 1201 2021	18f3a	0001 1000 1111 0011 1010	010
T02^2	0132 3101 3031	0120 1201 2102	18f3e	0001 1000 1111 0011 1110	110
T02^3	0132 3101 3131	0120 1201 2222	18f4d	0001 1000 1111 0100 1101	101
T03^3	0132 3101 0231	0120 1201 0212	18f14	0001 1000 1111 0001 0100	100
T04^3	0132 3102 3231	0120 1202 1012	18f6e	0001 1000 1111 0110 1110	110
T05^2	0132 3121 3231	0120 1210 2012	18fda	0001 1000 1111 1101 1010	010
T05^3	0132 3131 3231	0120 1222 2012	1916f	0001 1001 0001 0110 1111	111
T06^3	0132 3201 3231	0120 1021 2012	18b6c	0001 1000 1011 0110 1100	100
T07^3	0132 0101 3231	0120 2101 2012	194ea	0001 1001 0100 1110 1010	010
T08^2	0130 3101 3231	0121 0201 2012	1a04e	0001 1010 0000 0100 1110	110
T08^3	0131 3101 3231	0122 2201 2012	1cb05	0001 1100 1011 0000 0101	101
T09^3	0102 3101 3231	0102 1201 2012	128b4	0001 0010 1000 1011 0100	100
T10^3	0232 3101 3231	0210 1201 2012	228fe	0010 0010 1000 1111 1110	110
T11^2	2132 3101 3231	1020 1201 2012	35c8a	0011 0101 1100 1000 1010	010
T11^3	3132 3101 3231	2220 1201 2012	7dcd7	0111 1101 1101 1101 0111	111

2610

图26

不具有时钟缺失 (不具有码元滑动) 的示例性码元错误
 当单个码元错误在0Cle字= 1201_2012_0120₃ (0x4ADA8) 上出现时

错误位置	S11..S0 (码元)	T11..T0 (转变)	十六进制	比特[19:0]	比特[2:0]
良好	2030 2120 3021	1201 2012 0120	4ada8	0100 1010 1101 1010 1000	000
T00^1	2030 2120 3020	1201 2012 0122	4adaa	0100 1010 1101 1010 1000	010
T00^2	2030 2120 3021	1201 2012 0121	4ada9	0100 1010 1101 1010 1001	001
T01^1	2030 2120 3031	1201 2012 0102	4ada4	0100 1010 1101 1010 0100	100
T02^1	2030 2120 3121	1201 2012 0210	4adae	0100 1010 1101 1010 1110	110
T03^1	2030 2120 2021	1201 2012 2220	4ade7	0100 1010 1101 1110 0111	111
T03^2	2030 2120 1021	1201 2012 1020	4adba	0100 1010 1101 1011 1010	010
T04^1	2030 2120 3021	1201 2010 2120	4ad3c	0100 1010 1101 0010 1100	100
T05^1	2030 2110 3021	1201 2021 0120	4ae4a	0100 1010 1110 0100 1010	010
T06^1	2030 2020 3021	1201 2222 0120	4b44d	0100 1011 0100 0100 1101	101
T06^2	2030 2120 3021	1201 2102 0120	4af8e	0100 1010 1111 1000 1110	110
T07^1	2030 3120 3021	1201 0212 0120	4a244	0100 1010 0010 0100 0100	100
T08^1	2030 2120 3021	1202 1012 0120	4bebe	0100 1011 1110 1011 1110	110
T09^1	2020 2120 3021	1222 2012 0120	5610f	0101 0110 0001 0000 1111	111
T09^2	2010 2120 3021	1210 2012 0120	4e0ea	0100 1110 0000 1110 1010	010
T10^1	2030 2120 3021	1021 2012 0120	37a1c	0011 0111 1010 0001 1100	100
T11^1	3030 2120 3021	2101 2012 0120	67afa	0110 0111 1010 1111 1010	010
T11^2	0030 2120 3021	0001 2012 0120	02c5b	0000 0010 1100 0101 1011	011

图27

不具有时钟缺失 (不具有码元滑动) 的示例性码元错误
当单个码元错误在CC|e字= 2012_0120_1201₃ (0x5ED08) 上出现时

错误位置	S11..S0 (码元)	T11..T0 (转变)	十六进制	比特 [19:0]	比特 [2:0]
良好	3231 0132 3101	2012 0120 1201	5ed08	0101 1110 1101 0000 1000	000
T00^2	3231 0132 3103	2012 0120 1200	5ed07	0101 1110 1101 0000 0213	213
T00^3	3231 0132 3102	2012 0120 1202	5ed09	0101 1110 1101 0000 1001	001
T01^2	3231 0132 3121	2012 0120 1210	5ed0a	0101 1110 1101 0000 1010	010
T01^3	3231 0132 3131	2012 0120 1222	5ed0f	0101 1110 1101 0000 1111	111
T02^3	3231 0132 3201	2012 0120 1021	5edcf	0101 1110 1100 1110 1200	100
T03^3	3231 0132 0101	2012 0120 2101	5ed1a	0101 1110 1101 0001 1010	010
T04^2	3231 0130 3101	2012 0121 0201	5ed3e	0101 1110 1101 0011 1210	210
T04^3	3231 0131 3101	2012 0122 2201	5edc5	0101 1110 1101 1100 0301	101
T05^3	3231 0102 3101	2012 0102 1201	5ebc4	0101 1110 1011 1100 0100	100
T06^3	3231 0032 3101	2012 0210 1201	5eeee	0101 1110 1110 1110 1210	210
T07^2	3231 2132 3101	2012 1020 1201	5f2ba	0101 1111 0010 1011 1010	010
T07^3	3231 3132 3101	2012 2220 1201	600f7	0110 0000 0000 1111 0111	111
T08^3	3232 0132 3101	2010 2120 1201	5eadc	0101 1100 1010 1101 1200	200
T09^3	3201 0132 3101	2021 0120 1201	6204a	0110 0010 0000 0100 1010	010
T10^2	3031 0132 3101	2102 0120 1201	686ce	0110 1000 0110 1100 1110	110
T10^3	3131 0132 3101	2222 0120 1201	8073d	1000 0000 0111 0011 1201	201
T11^3	0231 0132 3101	0212 0120 1201	25264	0010 0101 0010 0110 0100	100

图28

错误检测—使用固定的常数比特

- 比特查位 [19:0] 的3个LSB检测任何单个码元错误; 不会遗漏从0x00000到0x81BF0的任何比特 [19:0] 中的错误
 - 发送任何已知值 (诸如比特 [2:0] 的 '000') 将提供对此类错误的100%检测
- 然而, 这将花费19比特信息中的3比特 (~19.02)
 - 折衷: 将比特 [0] 固定为0, 针对比特 [2:1] 取校验和

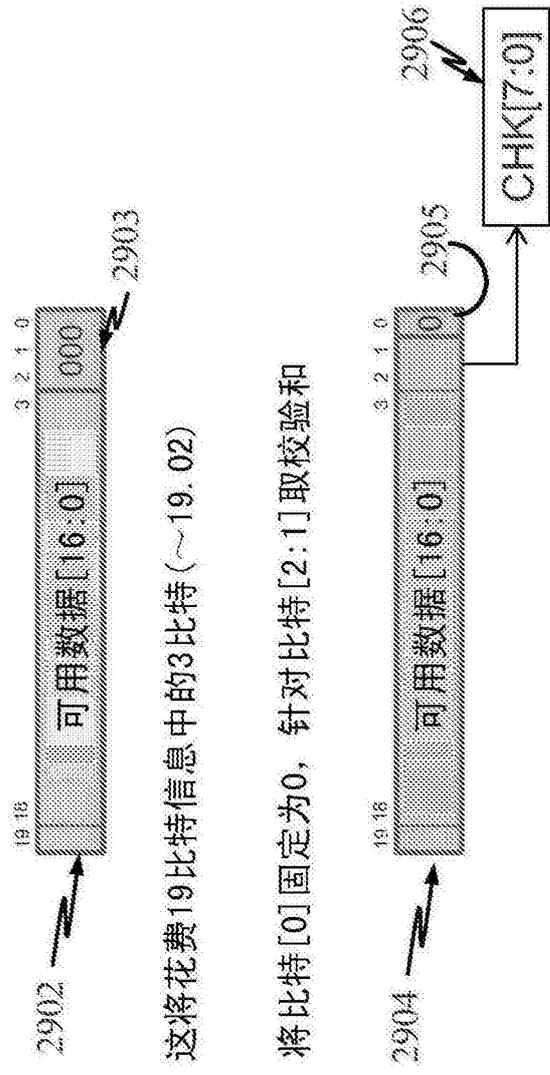


图29

兼容CC1e协议的传输中的示例性循环冗余校验消息

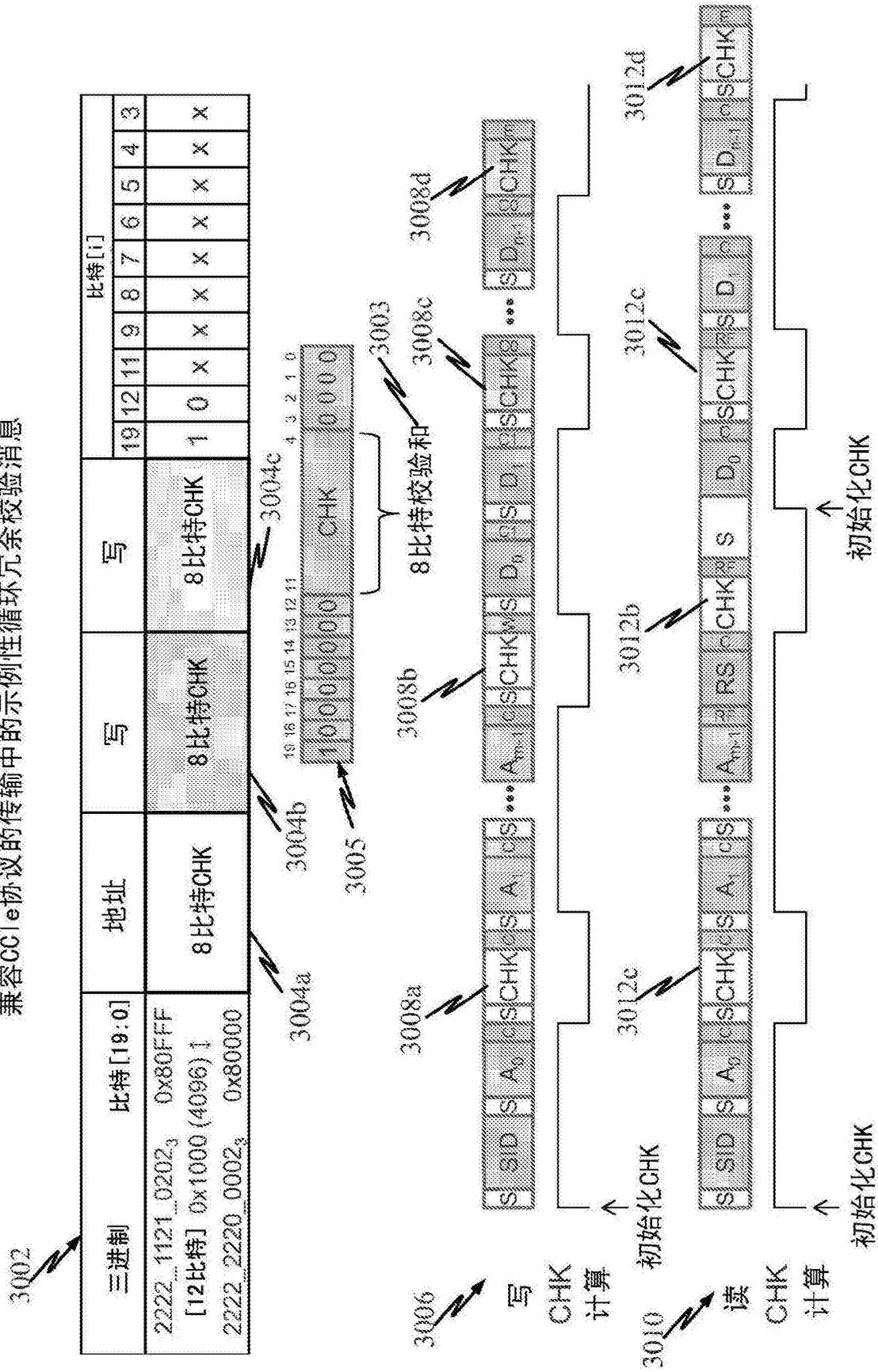
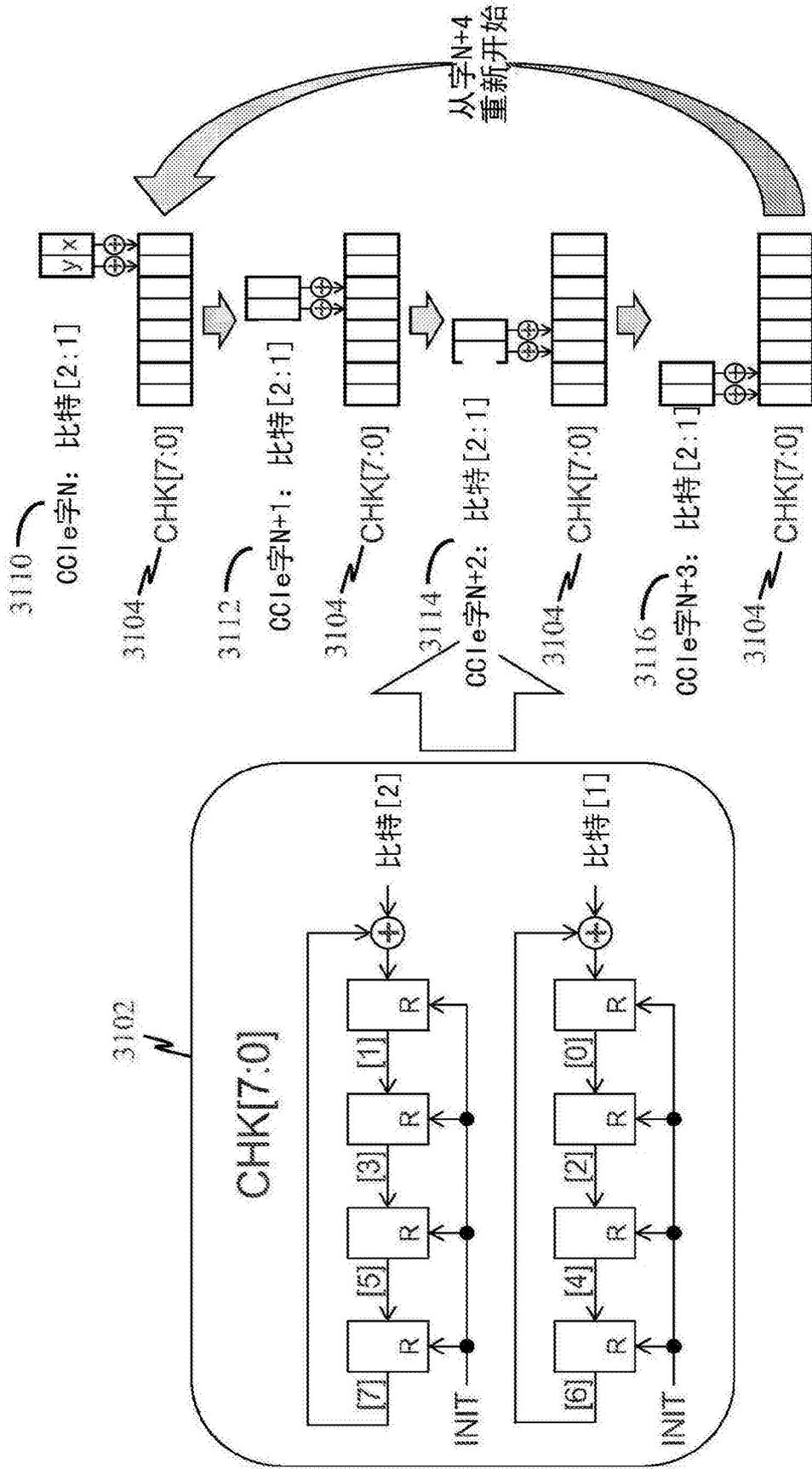


图30

示例性校验和生成器



30

图31

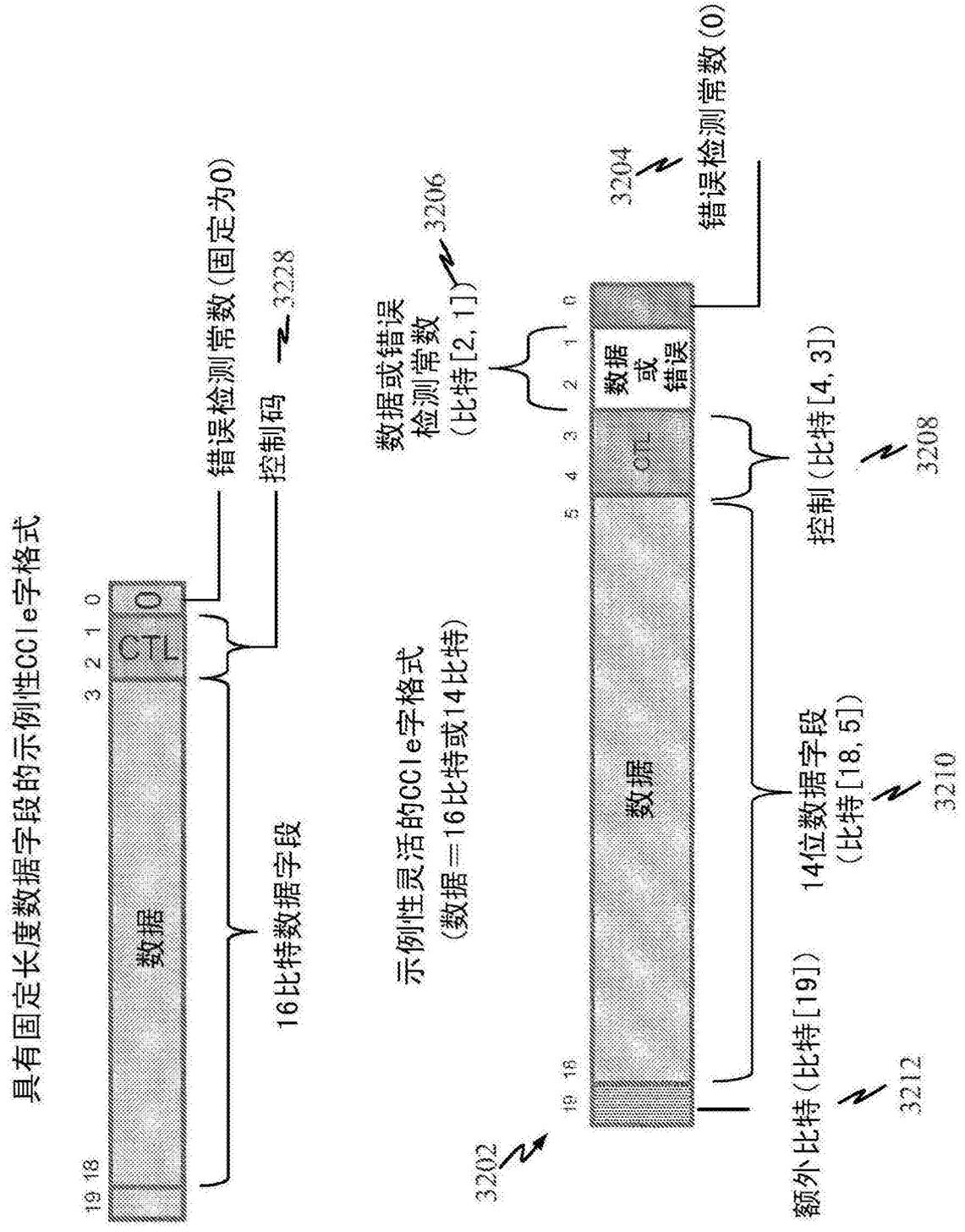


图32

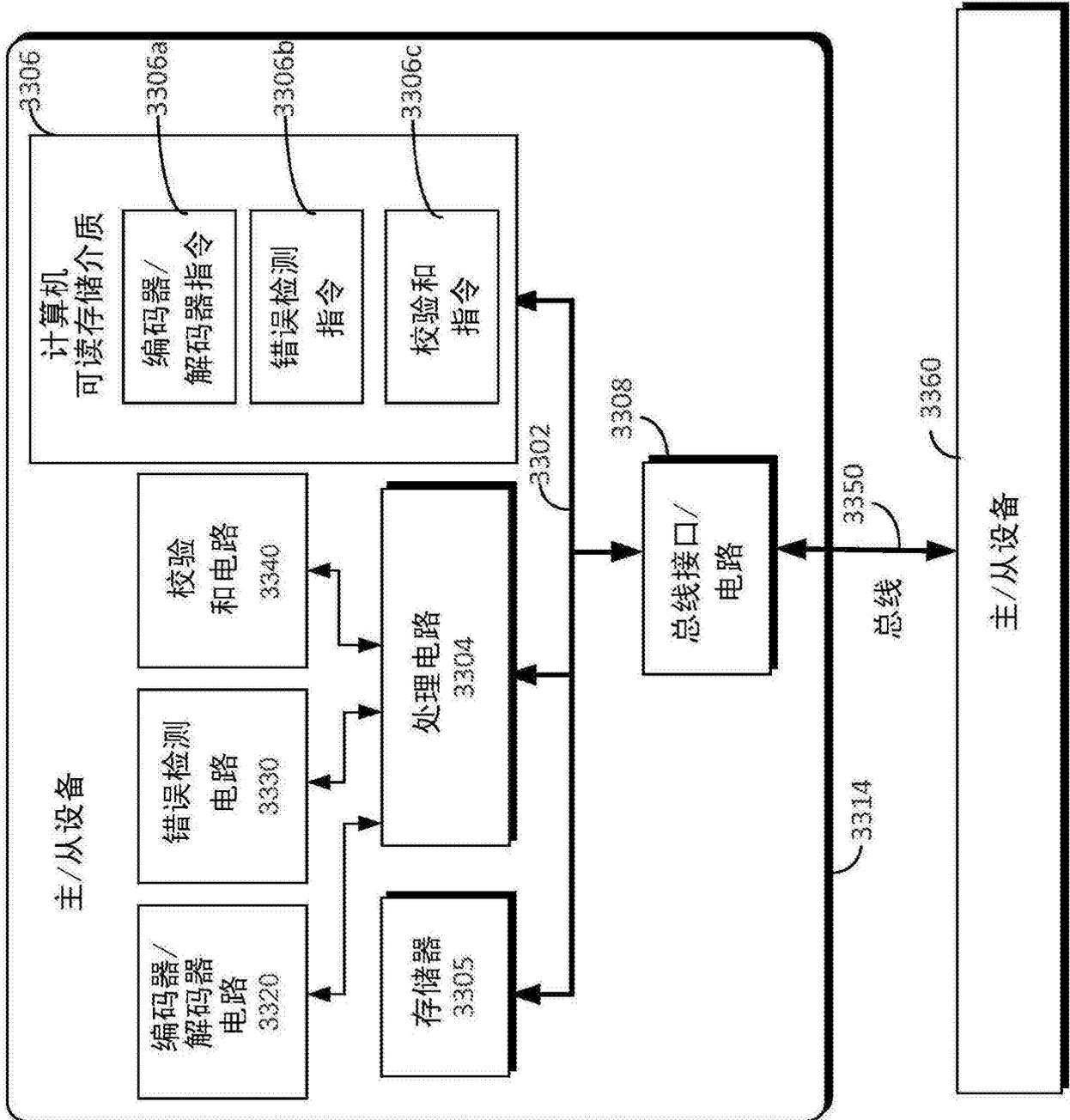
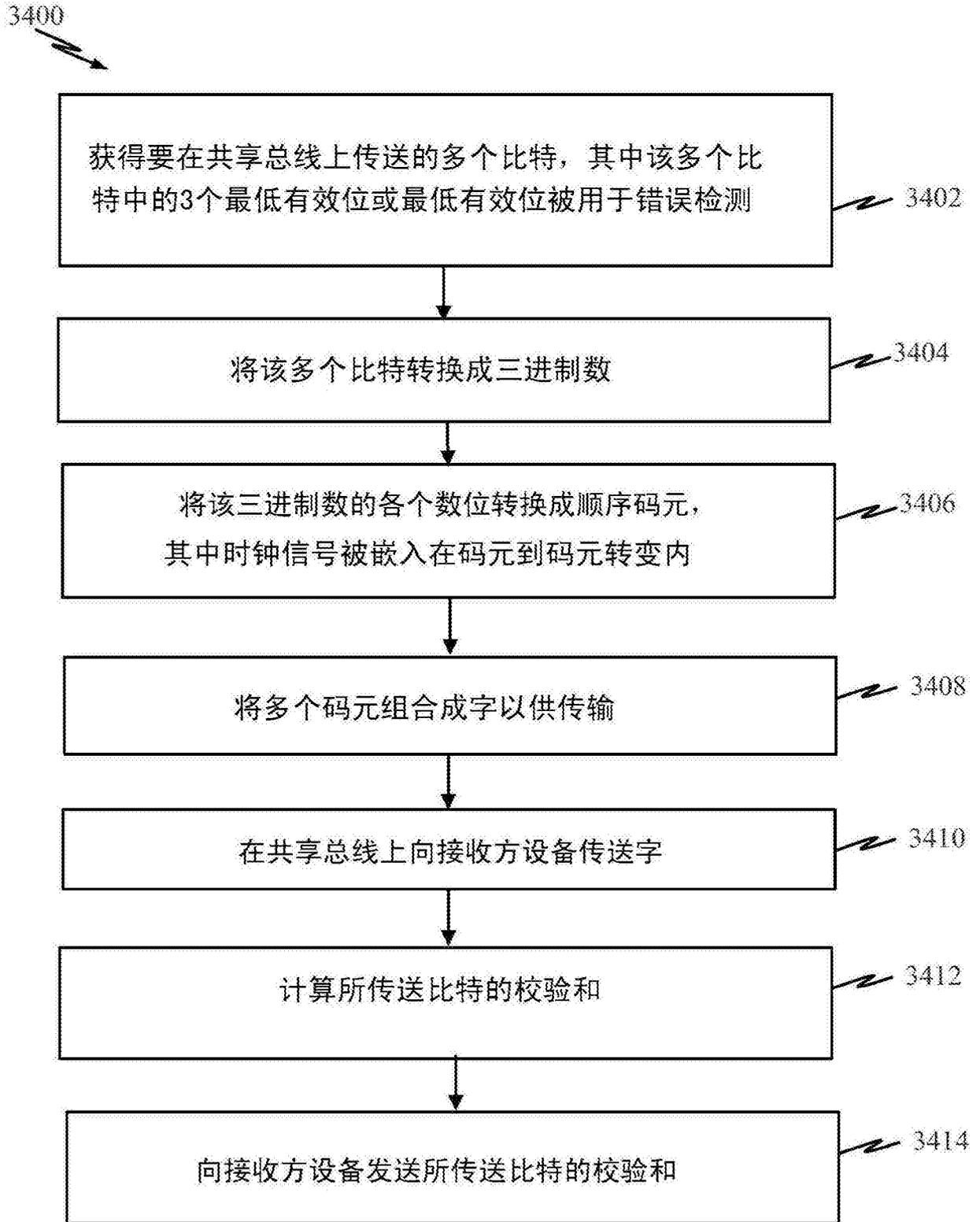


图33



用于错误检测的方法

图34

不具有时钟错误的示例性CCle传输

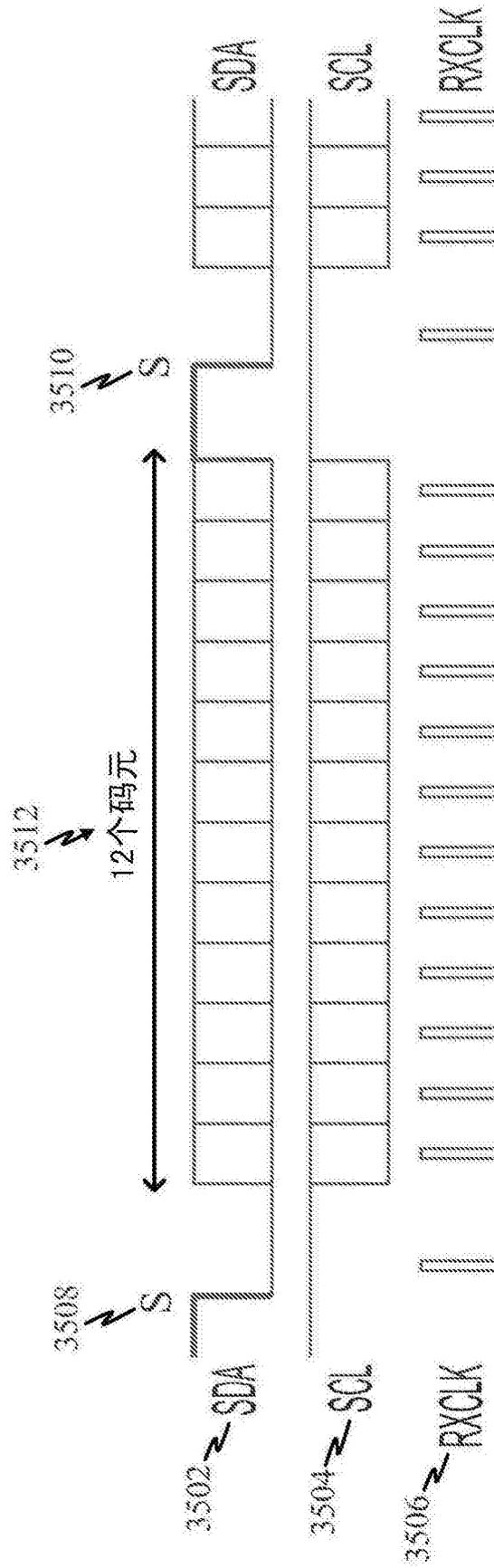


图35

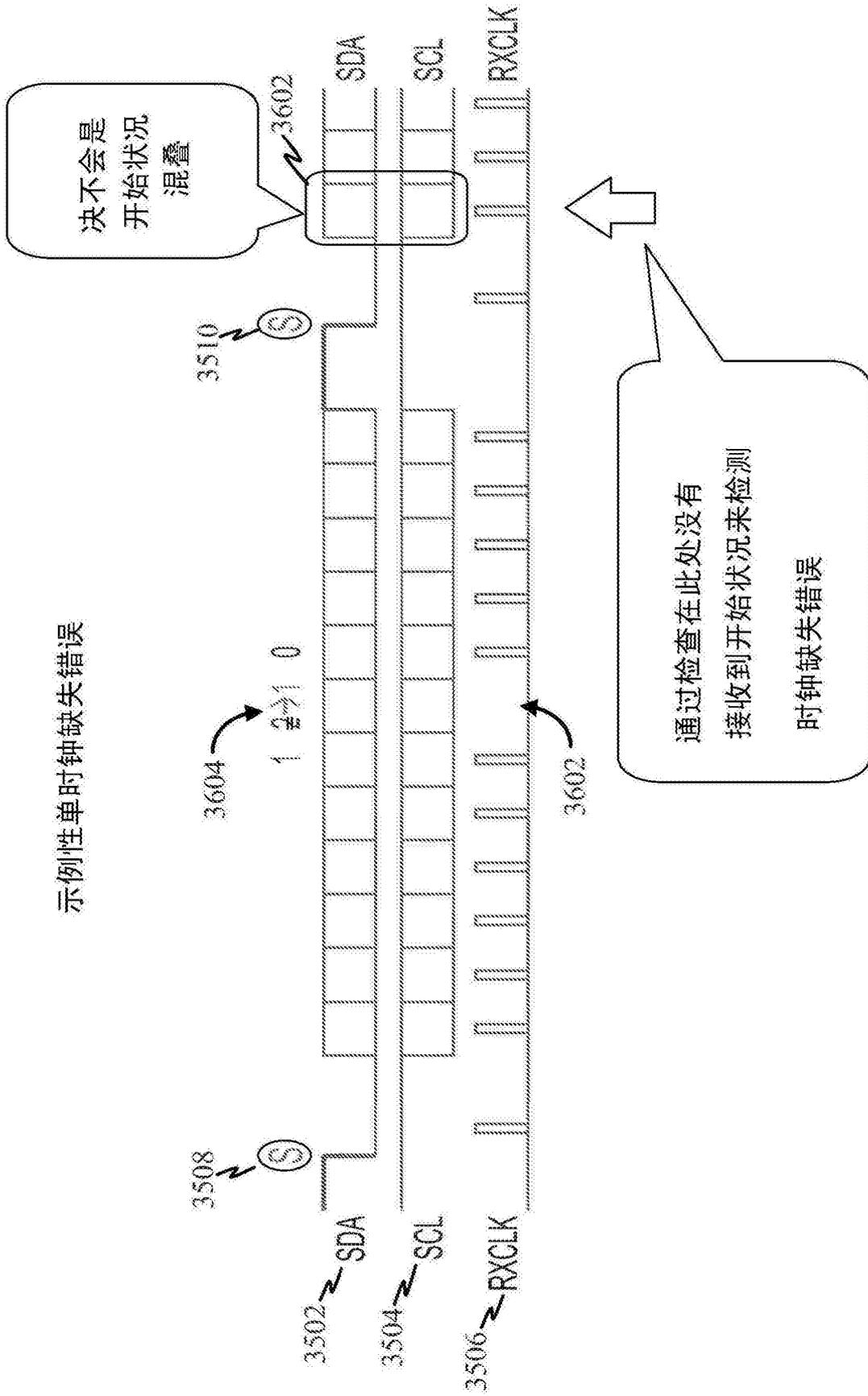


图36

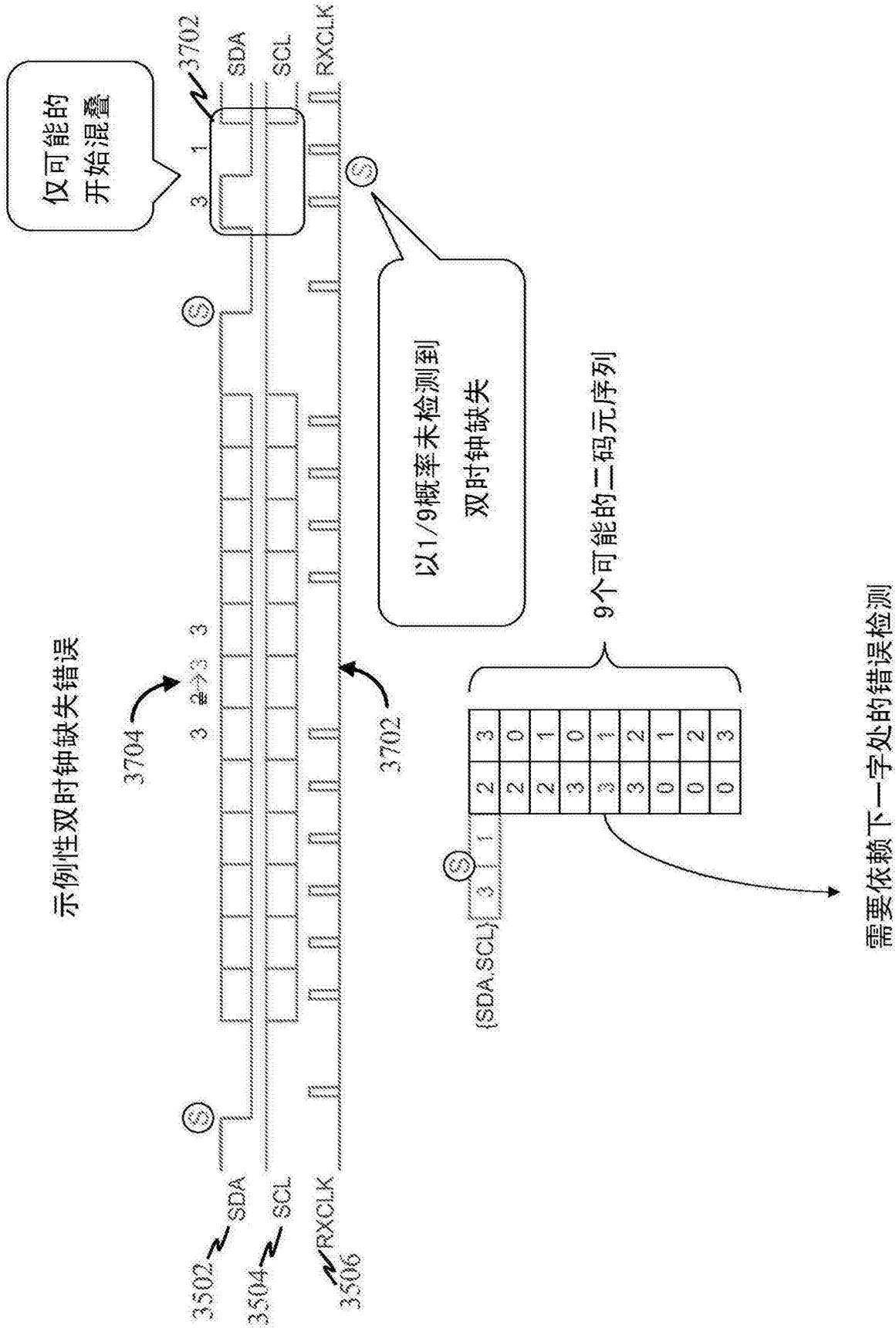


图37

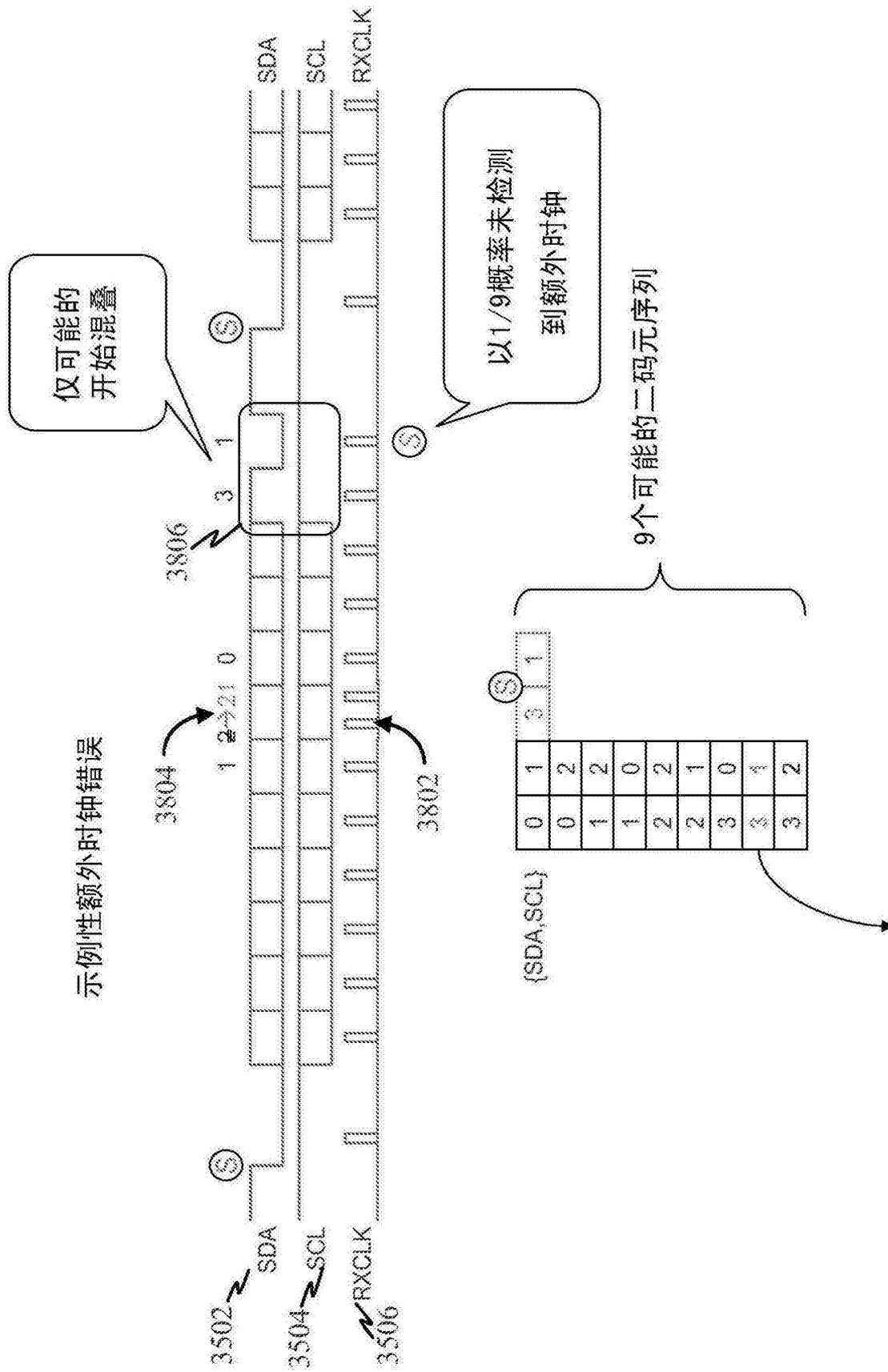


图38