



## (12)发明专利

(10)授权公告号 CN 105103454 B

(45)授权公告日 2019.03.29

(21)申请号 201480015715.2

(22)申请日 2014.03.14

(65)同一申请的已公布的文献号  
申请公布号 CN 105103454 A

(43)申请公布日 2015.11.25

(30)优先权数据  
13/841,574 2013.03.15 US

(85)PCT国际申请进入国家阶段日  
2015.09.15

(86)PCT国际申请的申请数据  
PCT/US2014/029554 2014.03.14

(87)PCT国际申请的公布数据  
WO2014/144941 EN 2014.09.18

(73)专利权人 吉林克斯公司  
地址 美国加利福尼亚州

(72)发明人 蒋蔚荣 高登·J·布莱诺  
马克·B·卡尔森

(74)专利代理机构 北京银龙知识产权代理有限公司 11243

代理人 许静 安利霞

(51)Int.Cl.  
H03M 13/09(2006.01)

(56)对比文件  
US 2009282322 A1,2009.11.12,  
CN 1431594 A,2003.07.23,  
CN 101296053 A,2008.10.29,  
CN 102571266 A,2012.07.11,  
JP 2001285076 A,2001.10.12,  
KR 100645388 B1,2006.11.14,

审查员 赵丽敬

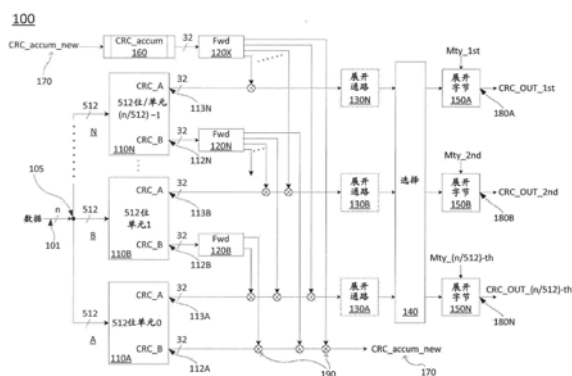
权利要求书2页 说明书12页 附图8页

### (54)发明名称

模块化且可扩展的循环冗余校验计算电路

### (57)摘要

本发明揭示用于执行循环冗余校验的装置以及方法。例如,装置(100)具有用于将数据字分解成多个路径的分解器(105)。所述装置还具有多个循环冗余校验单元(110A-110N)。所述单元中的每个单元用于处理所述路径中的相应一个路径。另外,所述单元(110A-110N)中的每个单元包含用于输出在所述单元内结束的数据包的循环冗余校验值的第一输出端口(113A-113N),以及用于输出在所述单元内起始或持续的数据包的循环冗余校验值的第二输出端口(112A-112N)。



1. 一种用于将循环冗余校验 (CRC) 附加到数据包的装置, 其包括:

分解器, 其用于将数据字分解成多个路径;

多个循环冗余校验单元, 其中所述循环冗余校验单元中的每个循环冗余校验单元用于处理所述路径中的相应一个路径, 其中所述循环冗余校验单元中的每个循环冗余校验单元包括:

第一输出端口, 其用于输出在所述循环冗余校验单元内结束的数据包的循环冗余校验值; 以及

第二输出端口, 其用于输出在所述循环冗余校验单元内起始或持续的数据包的循环冗余校验值。

2. 根据权利要求1所述的装置, 其进一步包括:

至少一个合并器, 其用于合并在所述多个循环冗余校验单元中的第一循环冗余校验单元内起始或持续的数据包的第一循环冗余校验值与在所述多个循环冗余校验单元中的第二循环冗余校验单元内结束的数据包的第二循环冗余校验值。

3. 根据权利要求2所述的装置, 其中在所述多个循环冗余校验单元中的所述第一循环冗余校验单元内起始或持续的所述数据包和在所述多个循环冗余校验单元中的所述第二循环冗余校验单元内结束的所述数据包是同一数据包。

4. 根据权利要求2或权利要求3所述的装置, 其进一步包括:

至少一个校验电路, 其用于校验至少一个数据包是否在所述多个循环冗余校验单元中的至少两个循环冗余校验单元之间连续, 所述至少一个数据包由所述多个循环冗余校验单元中的所述至少两个循环冗余校验单元处理。

5. 根据权利要求4所述的装置, 其中当在与所述两个循环冗余校验单元中的第一循环冗余校验单元相关联的第一路径中存在数据包起始信号或所述数据包在所述第一路径中持续时, 以及当在与所述两个循环冗余校验单元中的所述第二循环冗余校验单元相关联的第二路径中存在数据包结束信号时, 所述校验电路确定所述数据包在所述两个循环冗余校验单元之间是连续的。

6. 根据权利要求1到3中任一权利要求所述的装置, 其中所述循环冗余校验单元中的每个循环冗余校验单元进一步包括:

分解器, 其用于将所述路径中的一个路径分解成多个通路; 以及

多个循环冗余校验模块, 其用于计算所述多个通路中的相应一个通路的位的部分循环冗余校验值。

7. 根据权利要求1到3中任一权利要求所述的装置, 其中所述循环冗余校验单元中的每个循环冗余校验单元进一步包括:

选择模块, 其用于选择循环冗余校验值以用于作为在所述循环冗余校验单元内结束的所述数据包的所述循环冗余校验值来输出。

8. 根据权利要求6所述的装置, 其中所述循环冗余校验单元中的每个循环冗余校验单元进一步包括:

选择模块, 其用于选择循环冗余校验值以用于作为在所述循环冗余校验单元内起始或持续的所述数据包的所述循环冗余校验值来输出, 其中所述选择模块用于根据所述多个通路的一个通路中的数据包起始信号或根据若在所述多个通路的任何通路中不存在数据包

起始信号时的默认条件而选择所述循环冗余校验值,以用于作为在所述循环冗余校验单元内起始或持续的所述数据包的所述循环冗余校验值来输出。

9. 根据权利要求6所述的装置,其中所述循环冗余校验单元中的每个循环冗余校验单元进一步包括:

至少一个合并器,其用于合并来自至少第一循环冗余校验模块的部分循环冗余校验值与来自至少第二循环冗余校验模块的部分循环冗余校验值,以输出所述多个通路中与所述第二循环冗余校验模块相关联的相应一个通路的循环冗余校验值。

10. 根据权利要求1到3中任一权利要求所述的装置,其中所述循环冗余校验单元中的每个循环冗余校验单元进一步包括:

至少一个推进模块,其用于根据在所述至少一个推进模块的通路和目的地通路之间的位的数目而对至少一个部分循环冗余校验值进行位推进。

11. 根据权利要求1到3中任一权利要求所述的装置,其进一步包括:

输入端口,其用于接收来自前一数据字的累计循环冗余校验值;或

输出端口,其用于输出累计循环冗余校验值以用于下一数据字。

12. 根据权利要求11所述的装置,其进一步包括:

至少一个合并器,其用于合并所述累计循环冗余校验值与在所述循环冗余校验单元的至少一个单元内结束的数据包的至少一个循环冗余校验值;并且

其中由所述多个循环冗余校验单元处理的数据包的大小至少是所述路径中的每个路径的路径大小。

13. 根据权利要求1到3中任一权利要求所述的装置,其进一步包括:

至少一个推进单元,其用于根据在所述至少一个推进单元的相应路径与目的地路径之间的位的数目而对至少一个部分循环冗余校验值进行位推进。

14. 根据权利要求1到3中任一权利要求所述的装置,其进一步包括:

至少一个展开单元,其用于根据在数据包的结尾与通路的结尾之间或在所述数据包的所述结尾与路径的结尾之间的空位的数目,来执行至少一个循环冗余校验值的矩阵变换。

15. 一种用于将循环冗余校验(CRC)附加到数据包的方法,其包括:

将数据字分解成多个路径;以及

以并行方式处理所述路径中的每个路径,其中对于每个路径,所述处理包括:

计算在所述路径内结束的数据包的循环冗余校验值;以及

计算在所述路径内起始或持续的数据包的循环冗余校验值。

## 模块化且可扩展的循环冗余校验计算电路

### 技术领域

[0001] 本发明大体上涉及用于对高数据速率下的数据包执行循环冗余校验计算的装置(例如,现场可编程门阵列等集成电路)以及方法。

### 背景技术

[0002] 循环冗余码(CRCs)广泛应用于电信和网络中以确保数据完整性。例如,用附加的CRC-32码(32位CRC码)传输每个以太网数据包。已经使用移位寄存器串联地实施用于执行循环冗余校验的电路。然而,此方法不足以跟上现行的数据速率。另外,虽然对于多达100Gb/s的数据速率存在若干CRC电路设计,但是这些设计通常包括使用相对狭窄的数据总线,其具有不超过512位的输入宽度。

### 发明内容

[0003] 在一个实施例中,揭示一种用于执行循环冗余校验的装置。例如,所述装置包括用于将数据字分解成多个路径的分解器。所述装置还包括多个循环冗余校验单元。所述单元中的每个单元用于处理所述路径中的相应一个路径。另外,所述单元中的每个单元包括用于输出在所述单元内结束的数据包的循环冗余校验值的第一输出端口以及用于输出在所述单元内起始或持续的数据包的循环冗余校验值的第二输出端口。

[0004] 在各种实施例中:通过多个循环冗余校验单元处理的数据包的大小是路径中的每个路径的至少路径大小;所述装置进一步包括至少一个合并器,其用于合并多个循环冗余校验单元中的第一单元内起始或持续的数据包的第一循环冗余校验值与在多个循环冗余校验单元中的第二单元内结束的数据包的第二循环冗余校验值;所述至少一个合并器用于对第一循环冗余校验值和第二循环冗余校验值执行异或运算;在多个循环冗余校验单元中的第一单元内起始或持续的数据包和在多个循环冗余校验单元中的第二单元内结束的数据包是同一数据包;所述装置进一步包括至少一个校验电路,其用于校验通过多个循环冗余校验单元中的至少两个单元处理的至少一个数据包是否在多个循环冗余校验单元中的所述至少两个单元之间连续;当在与所述两个单元中的第一单元相关联的第一路径中存在数据包起始信号或数据包在第一路径中持续时,以及当在与所述两个单元中的第二单元相关联的第二路径中存在数据包结束信号时,校验电路确定数据包在所述两个单元之间连续;循环冗余校验单元中的每个单元进一步包括用于将路径中的一个路径分解成多个通路的分解器以及用于计算多个通路中的相应一个通路的位的部分循环冗余校验值的循环冗余校验模块;M个循环冗余校验单元中的每个单元进一步包括选择模块,其用于选择循环冗余校验值以用于作为在循环冗余校验单元内结束的数据包的循环冗余校验值来输出;选择单元用于根据通路中的一个通路中的数据包结束信号或若在所述通路中的任何通路中不存在数据包结束信号则根据默认条件选择中间循环冗余校验值,以用于作为在循环冗余校验单元内结束的数据包的循环冗余校验值来输出;循环冗余校验单元中的每个单元进一步包括选择模块,其用于选择循环冗余校验值以用于作为在循环冗余校验单元内起始或持续

的数据包的循环冗余校验值来输出；所述选择模块用于根据通路中的一个通路中的数据包起始信号或若在所述通路中的任何通路中不存在数据包起始信号则根据默认条件选择循环冗余校验值，以用于作为在循环冗余校验单元内起始或持续的数据包的循环冗余校验值来输出；循环冗余校验单元中的每个单元进一步包括至少一个合并器，其用于合并来自至少第一循环冗余校验模块的部分循环冗余校验值与来自至少第二循环冗余校验模块的部分循环冗余校验值，以输出与第二循环冗余校验模块相关联的通路中的相应一个通路的循环冗余校验值；循环冗余校验单元中的每个单元进一步包括至少一个推进单元，其用于根据在至少一个推进单元的通路和目的地通路之间的位的数目对至少一个部分循环冗余校验值进行位推进；所述装置进一步包括用于接收来自前一数据字的累计循环冗余校验值的输入端口和/或用于输出累计循环冗余校验值以用于下一数据字的输出端口；所述装置进一步包括至少一个合并器，其用于合并累计循环冗余校验值与在循环冗余校验单元中的至少一个单元内结束的数据包的至少一个循环冗余校验值；所述装置进一步包括至少一个推进单元，其用于根据在至少一个推进单元的路径和目的地路径之间的位的数目对至少一个部分循环冗余校验值进行位推进；所述装置进一步包括至少一个展开单元，其用于根据在数据包的结尾与通路的结尾之间或在数据包的结尾与路径的结尾之间的空位数展开至少一个循环冗余校验值。

[0005] 在另一实施例中，揭示用于执行循环冗余校验的另一装置。例如，所述装置包括用于处理数据字的多个循环冗余校验单元。在一个实施例中，所述循环冗余校验单元中的每个单元用于处理数据字的一部分且用于输出相应循环冗余校验值。所述装置还包含选择单元，其用于选择多个相应循环冗余校验值以用于作为相应多个数据包的最终循环冗余校验值来输出。数据字大于最小数据包大小。

[0006] 在另一实施例中，揭示一种用于执行循环冗余校验的方法。例如，所述方法包括将数据字分解成多个路径以及处理所述路径中的每个路径。在一个实施例中，对于所述路径中的每个路径，所述处理包括计算在路径内结束的数据包的循环冗余校验值以及计算在路径内起始或持续的数据包的循环冗余校验值。

## 附图说明

[0007] 附图示出根据本发明的一个或多个方面的示例性实施例；然而，附图不应被视作限制所示出的实施例的揭示内容，而是仅用于说明和理解。

[0008] 图1示出了CRC电路/装置实施方案的方块图；

[0009] 图2示出了CRC单元的方块图；

[0010] 图3示出了CRC单元的替代实施例的方块图；

[0011] 图4示出了在路径上和/或通过CRC单元处理的位的示例性情境的时序图；

[0012] 图5示出了用于校验数据包是否在两个CRC单元之间连续的校验电路的方块图；

[0013] 图6示出了在路径上和/或通过CRC单元处理的位的其它示例性情境的时序图；

[0014] 图7示出了用于执行循环冗余校验的方法的方块图；以及

[0015] 图8示出了适用于执行本文中所描述的功能的通用计算机或计算装置的高阶方块图。

## 具体实施方式

[0016] 本发明大体上涉及用于执行循环冗余校验的装置以及方法。例如,本发明可以在集成电路(IC)内实施,例如,可编程逻辑装置(PLD)、专用IC、现场可编程门阵列(FPGA)等,从而对在高位速率下以及在宽数据总线上接收到的数据包执行循环冗余校验运算。本发明的实施例可以使用具有包含1024位以及超过1024位的输入宽度的宽数据总线支持超过100Gb/s的数据传输速率,例如,从400Gb/s往上。电路架构是模块化且可扩展的,从而允许提供资源性能的权衡。

[0017] 循环冗余码(CRC)广泛应用于电信和网络以及许多其它应用中以确保数据完整性。例如,用也称为帧校验序列(FCS)的附加的CRC-32码(32位CRC码)传输以太网数据包。已经使用移位寄存器串联地实施用于执行循环冗余校验的电路。然而,此方法不足以跟上现行的数据速率。另外,虽然对于多达100Gb/s的数据速率存在若干CRC电路设计,但是这些设计通常包括使用具有不超过512位的输入宽度/字长的相对狭窄的数据总线。这限制了未来的吞吐量扩展,因为变得更难以获得时脉频率的显著提高。为了实现更高的吞吐量,例如400Gb/s,可以使用更宽的数据总线,例如,CRC电路可用来处理每时钟周期超过1024位的更宽的数据总线。然而,因为网络数据包/以太网帧的最小大小是512位,所以在此宽数据总线上可以同时呈现多个数据包。数据包可以在数据总线内的任何位置处起始和结束。因此,随着每字的数据包的数目增加,由此考虑到不同数据包位置来说变得更复杂。

[0018] 因此,本发明的实施例描述可扩展的CRC架构,例如,在512位CRC单元上建构的可扩展的CRC架构,例如,优化为512位输入数据宽度的可扩展的CRC架构。本发明的实施例提供随着吞吐量增加时的资源使用的线性扩展,而传统的方法包括超线性扩展。应注意,尽管本发明的实施例结合512位的倍数和分数来描述,但是本发明不因此而受限制。即,示例性架构以512位路径大小/CRC单元大小为特征,所述大小对应于现行的最小以太网帧大小,且因此较适用于说明性目的。然而,所属领域的技术人员将了解,可以根据本发明采用其它数据总线宽度、帧大小、路径大小和行大小。因此,本发明的其它实施例可以适应不同通信技术、针对不同网络层优化、以及针对不同数据包/单元大小调整,例如,帧、数据包、数据段、数据报、协议数据单元(PDU)、传输数据单元(TDU)、会话数据单元(SDU)等。

[0019] 循环冗余校验是在数据传输应用中广泛使用的根据奇偶校验位的差错检测方案且根据多项式算法。将被传输的数据的位是多项式的系数。作为实例,位流1101011011具有10位,表示10项多项式:

[0020]  $M(x) = 1 \cdot x^9 + 1 \cdot x^8 + 0 \cdot x^7 + 1 \cdot x^6 + 0 \cdot x^5 + 1 \cdot x^4 + 1 \cdot x^3 + 0 \cdot x^2 + 1 \cdot x^1 + 1 \cdot x^0 = x^9 + x^8 + x^6 + x^4 + x^3 + x^1 + 1$ 。

[0021] 为了计算消息的CRC,选择称为生成多项式的另一多项式 $G(x)$ 。 $G(x)$ 应具有大于零的次数且在 $x^0$ 项中具有非零系数。这产生了生成多项式的若干可能选项,因此需要标准化。CRC-16是一种此类标准,其使用以下生成多项式:

[0022]  $G(x) = x^{16} + x^{15} + x^2 + 1$ 。

[0023] CRC-16检测所有单一和双重差错、具有奇数位数的所有差错、长度16或更短的所有突发差错、以及针对更长突发的大多数差错。另一标准CRC是CRC-32,其使用以下生成多项式:

[0024]  $G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ 。

[0025] 一般来说,通过将数据流表示为多项式 $M(x)$ ,将 $M(x)$ 乘以 $x^n$ (其中 $n$ 是多项式 $G(x)$ 的次数),并且将结果除以生成多项式 $G(x)$ 来计算 $n$ 位CRC。所产生的余数被附加到数据流并进行传输。接着将完整的所传输多项式除以在接收器端的同一生成多项式。如果此除法运算的结果没有余数,则不存在传输差错。在数学上,这可以表示为:

[0026]  $CRC = [(M(x) \cdot x^n) / G(x)]$  的余数。

[0027] IEEE 802.3将多项式 $M(x)$ 界定为目的地址、源地址、长度/类型、以及帧的数据,其中对前面的32位进行补充(complemented)。对来自以上CRC计算的余数进行补充,并且结果是IEEE 802.3的32位帧校验序列(FCS)字段。FCS被附加到以太网帧的结尾,并且以最高阶位为第一的顺序( $x^{31}$ 、 $x^{30}$ 、...、 $x^1$ 、 $x^0$ )进行传输。

[0028] 在一个实施例中,本发明根据生成多项式对传入数据的各个部分采用矩阵运算,以生成部分、中间和最终CRC。在本发明的实施例中使用循环冗余码(CRC)的若干适用属性,例如:

[0029] 属性1:  $CRC(A \text{ 异或 } B) = CRC(A) \text{ 异或 } CRC(B)$ ;

[0030] 属性2:  $CRC(\{X, 0^P\}) = H^P \cdot CRC(X)$ ;

[0031] 其中 $0^P$ 是 $P$ 位零,  $\{X, 0^P\} = X$ 具有附加的 $P$ 个零,  $H^P$ 是从生成多项式导出的矩阵;并且

[0032] 属性3:  $CRC(\{0^P, Y\}) = CRC(Y)$ 。

[0033] 作为实例,传入数据字 $D$ 可以包括1024位。为了计算 $D$ 的循环冗余码(CRC),数据可以描述为 $D = [\text{位}1023 \dots \text{位}0] = E[D1 \mid 0^{512}] \text{ 异或 } F[0^{512} \mid D2]$ ,其中 $D1 = [\text{位}1023 \dots \text{位}512]$ 且 $D2 = [\text{位}511 \dots \text{位}0]$ 。因此,根据属性1,  $CRC(D) = CRC(E) \text{ 异或 } CRC(F)$ 。 $CRC(F)$ 可以通过根据属性3简单地计算 $D2$ 的CRC来计算。换句话说,可以关于 $D2$ 的仅512位而不是对 $F$ 的所有1024位来执行 $CRC(F)$ 计算。可以根据如下属性2计算 $CRC(E)$ 。首先,针对 $D1$ 计算CRC。其次,将 $CRC(D1)$ 乘以从生成函数 $H^P$ 导出的矩阵,其中 $CRC(E)$ 为结果。一旦已知 $CRC(E)$ 和 $CRC(F)$ ,就能计算出 $CRC(D)$ 为 $CRC(D) = CRC(E) \text{ 异或 } CRC(F)$ 。

[0034] 下文结合各个组件和特征论述这些各个属性和变换的实用性。确切地说,本发明的实施例以采用并联CRC单元和并联路径的CRC电路为特征。每个路径包含CRC单元,通过CRC单元各自处理传入数据字的一部分以生成两个CRC输出,所述输出接着可与来自其它单元中的一个或多个单元的CRC输出合并以产生在数据字内结束的一个或多个数据包的最终CRC值。

[0035] 图1示出了根据本发明的实施例的装置或CRC电路100的方块图。如图1中所图示,CRC电路100被划分成 $N$ 个路径,其中每个路径包含512位宽的计算区块(概括地, $n$ 位宽计算区块),例如,每个路径分别具有512位CRC单元110A、110B、...、110N,在各种实施例中,每个512位CRC单元包括四个通路,每个通路具有128位CRC计算模块,其分别在图2和3中以更多细节图示出。应注意,示出的四个128位通路仅出于说明性目的。因此,本发明不限于任何数目的通路或任何特定的通路大小。另外,在一个实施例中,可能使用不同类型的CRC单元架构,例如,根据宽(多达549位)异或门的更为整体式的区块。因此,CRC单元110A到110N中的任何一个或多个单元的此类替代性架构可并入到CRC电路100中而不改变其基本功能。

[0036] 在一个实施例中,CRC电路100包括 $N$ 个数目的512位路径A、B和N(图1中仅图示了三个路径)。因此,输入端口101上的传入数据可包括1536位宽的数据总线。分解器105将数据总线上的信号分解成相应的512位路径A、B和N。在一个实施例中,串行数据在进入CRC电路

100之前转化成并行数据字。因此,在一个实施例中,“前面”的位可以从电路的顶部朝向电路的底部放置在总线上,“后面”的位在电路的底部处经过处理。

[0037] 根据本发明的实施例,数据字的宽度是N位。因为最小大小的以太网数据包是64字节(即,512位),所以存在多达 $(N/512)$ 个数据包在同一字中结束。因此,对于每个字存在多达 $(N/512)$ 个CRC输出。因此,在图1中,例如在输出端口180A、180B和180N上,可能存在三个CRC输出,因为可能有多达三个512位以太网帧在同一1536位数据字内结束。例如,第n个CRC输出180N是针对同一N位输入数据字上的第n个完成的数据包( $n=N/512$ )。

[0038] 每个512位CRC单元110A到110N生成两个循环冗余码:一个针对在单元/路径(例如,CRC输出端口113A到113N上的CRC\_A)内结束的数据包,且另一个针对在单元/路径(例如,CRC输出端口112A到112N上的CRC\_B)内起始或持续的数据包。寄存器160存储跨越多个字的数据包的累计CRC值(CRC\_accum\_new 170)。

[0039] 在并行CRC计算中,可以将零附加到数据以用于计算部分CRC。这些附加的零称为“空位”。因此,各种实施例还将在数据包的结尾处的另外累计的零校正为128位边界(例如,使用如图2中所示的CRC电路)或者512位边界(例如,使用如图3中所示的CRC电路)。这些方法中的每一个还适用于不需要(例如,若原始数据包在开始处补零)将其结尾与128位或512位边界对齐的结尾调整上。在一个实施例中,也可以在CRC单元110A到110N中的每个单元中进行处理之后实施考虑空位和/或补零的进一步运算。例如,“Fwd”区块(即,推进单元120B、120N和120X)将CRC值推进了不同数目的空位。通路展开单元130A到130N以及字节展开单元150A到150N将CRC展开了不同数目的空字节。下文在论述图2和3的示例性CRC单元200和300之后将更详细地描述“推进”和“展开”功能。

[0040] 在一个实施例中,N位数据总线上的传入数据(例如,以太网帧,其在本文中也可以被称为数据包)包含 $(N/L)$ 位数据包起始(SoP)信号和数据包结束(EoP)信号。在一个实施例中,SoP信号和EoP信号的大小是N/L位。在整个CRC电路100内存在N/L个通路。因此,在一个实施例中,SoP信号或EoP信号指示数据包(或帧、PDU等)是否在对应通路处起始或结束以及帧在通路内的何处结束。如果针对通路检测到SoP信号或EoP信号,则设置对应于所述通路的SoP或EoP标志,例如,单一位设置成“1”以为特定通路指示检测到SoP信号,并且如果在所述通路中的数据上未检测到SoP信号,则SoP标志可以保持零“0”。在一个实施例中,假定数据包的起始必须与通路的起始对齐。在一个实施例中,数据包的结尾可以不与通路的结尾对齐。然而,在一个实施例中,数据包的结尾假定将与字节的结尾对齐。因此,通路含有L/8个字节并且数据包可以在结束通路内的任何字节处结束。在一个实施例中,计算出Mty(“空”)值以指示通路内的空字节的数目。在一个实施例中,根据检测在特定字节内的EoP信号以及确定在通路内的低于检测到所述EoP信号的所述字节的位置的额外字节的数目来计算Mty值。存储Mty值所需的位数通过以下公式给出: $\text{size\_Mty} = (N/512) * \text{Log}_2(L/8)$ 位。因为最小大小的以太网数据包是512位,所以在同一字中可以含有多达 $(n=N/512)$ 个数据包。因此,第n个Mty是针对第n个完成的数据包。

[0041] 图2示出512位CRC单元200的示例性实施例,其中 $L=128$ (即,其中通路宽度是128位)。例如,CRC单元200可实施为图1中的CRC单元110A到110N中的每个单元。如图2中所示,CRC单元200包含四个CRC模块210A到210D。所述模块中的每个模块对每个L位通路A到D上的数据计算CRC值。每个L位通路上的CRC值在本文中被称为部分CRC值。推进单元220B、220C和



220D包含用于将部分CRC值推进“i”个通路的“Fxi”模块。例如,如图2所示,推进单元220C包含Fx2模块,其将通路C的部分CRC值推进了两个通路,即,到通路A。如图2所示,推进后的部分CRC值中的每一个与目的地通路的部分CRC值合并。例如,图2中示出几个合并器290,所述合并器中的每一个用于对其采用作为输入的部分CRC值执行(例如)异或运算。

[0042] 除推进部分CRC值之外,在推进/发送部分CRC值到特定目的地通路之前,推进单元220B到220D中的每一个根据对应于目的地通路的位数“推进”部分CRC值。应注意,进行位推进的运算是CRC值的数据处理/运算,与将所产生的CRC值“推进”(例如,发送和/或输出)到其它通路和/或到选择模块(例如,多路复用器230和240)大不相同。例如,通路D中的推进单元220D的Fx2模块将根据256位(对应于两个通路的位数)来推进部分CRC值并且将所产生的部分CRC值发送到通路B,在通路B处,所产生的部分CRC在合并器290处经由异或运算与通路B的部分CRC值合并。在一个实施例中,推进包括根据生成函数和到目的地通路的距离(位数)对输入CRC值进行的矩阵变换。在特定通路的最后一个合并器之后的CRC值输出包括到多路复用器230的输入中的一个。推进单元220B到220D中的每个单元的推进模块中的每个模块以类似方式运算。包括来自各个通路的部分CRC值的合并的CRC值包括到多路复用器230和240的相应输入。

[0043] “EoP\_Index”信号270指定其中数据包结束的通路的索引(例如,其中对应EoP位/标志设置成一“1”以指示数据包在此通路中结束的通路)。EoP\_Index信号270控制到多路复用器230的输入中的哪些输入要在输出端口260上作为CRC\_A(在通过CRC模块200处理的512位内结束的数据包的CRC值)输出。换句话说,多路复用器230根据EoP\_Index的值选择将哪些输入CRC值作为CRC\_A输出。类似地,“SoP\_Index”信号280指定其中数据包起始的通路的索引(例如,针对所述通路的对应SoP位/标志设置成一“1”)。EoP\_index和SoP\_index的默认值分别是0和(512/L-1)。SoP\_Index信号280控制到多路复用器240的输入CRC值中的哪些输入CRC值要在输出端口250上作为CRC\_B(在CRC单元200内起始或持续的数据包的CRC值)输出。换句话说,多路复用器240根据SoP\_Index的值选择将哪些输入CRC值作为CRC\_B输出。

[0044] 返回图1的论述,现在将描述CRC电路100的其它组件和特征。如上所述,假定数据包在通路起始处开始。然而,在一个实施例中,数据包可以在路径内的任何通路处并且在通路内的任何位置处结束(假定在字节的结尾处结束)。因此,在一个实施例中,在CRC电路100中包含字节展开单元150A到150N以考虑在通路的结尾并未结束的数据包。如上所述,在一个实施例中,计算出Mty(“空”)值以指示通路内的空字节的数目。可以根据检测在特定字节内的EoP信号以及确定在通路内的低于检测到EoP信号的字节的位置的额外字节的数目来计算Mty值。存储Mty值所需的位数通过以下公式给出: $\text{size\_Mty} = (N/512) * \text{Log}_2(L/8)$  位。因为最小大小的以太网数据包是512位,所以在同一字中可以含有多达( $n = N/512$ )个数据包。因此,第n个Mty是针对第n个完成的数据包。如图1中所示,Mty<sub>1<sup>st</sup></sub>、Mty<sub>2<sup>nd</sup></sub>.....Mty<sub>(n/512)<sup>th</sup></sub>分别输入到字节展开单元150A到150N。作为实例,Mty<sub>1<sup>st</sup></sub>可以具有值7。这表示在通路的字节9处结束的数据包,以及表示所述通路的最后七个字节应处理为零/空字节。

[0045] 在一个实施例中,为了根据Mty值“展开”或“解开”CRC,执行根据生成函数和根据Mty的值(位数)对输入CRC值进行的矩阵变换。在另一实施例中,可以在字节展开单元150A到150N中的任何一个或多个中实施流水线式展开运算。在一个实施例中,流水线包括i个阶段,其中第i个阶段用于展开 $2^i$ 字节。因此,在一个实施例中,需要四个流水线阶段来展开至

多16个字节。在另一实施例中,执行迭代函数。例如,在一个实施例中,移去CRC值的最低有效位。如果所述位是一“1”,则将除数(即,生成多项式/生成函数)增加到所得CRC值。然而,如果所移去的位是零“0”,则将零增加到所得CRC值。针对如通过Mty值指示的位数迭代此运算。举例来说,如果Mty<sub>1<sup>st</sup></sub>的值是2(2字节或16位),则执行16次展开运算以计算最终CRC值。在另一实施例中,可以采用不同展开算法。换句话说,本发明不限于任何一个特定的展开算法。

[0046] 当CRC电路100的CRC单元110A到110N中的每个单元包括例如图2的单元200的CRC单元时,那么可从电路100省略通路展开单元130A到130N。然而,在一个实施例中,这些通路展开单元包含于电路100中,其中CRC单元维持数据包对齐到512位路径的结束。图3中示出对齐到路径的结束的示例性CRC单元。

[0047] 图3示出根据一个实施例的CRC单元300。举例来说,图1的CRC单元110A到110N中的任何一个或多个可包括CRC单元300。如图3中所展示,CRC单元300与图2的CRC单元200共享许多类似性。例如,CRC单元300具有512位输入端口/总线以及用于将512位分解成四个128位通路A到D的分解器305。通路A到D中的每个通路分别包含CRC模块310A到310D。然而,和图2不同的是,图3的推进单元320B到320D在输出/推进CRC值之前执行仅单个位推进的运算。例如,推进单元320D可根据384位推进CRC值中的位(三个通路的推进),假定路通A的目的地通路为路径的最终通路。应注意,尽管数据包实际上可在通路A上结束,但是不必这样。例如,数据包仍可在通路B上结束。类似地,推进单元320C可根据256位“推进”通过CRC单元310C输出的部分CRC的位,也假定目的地通路A为路径的最终通路。以此方式进行的位推进基本上将数据包与路径的结尾/CRC单元300的结尾对齐。然而,由于此假设可能不正确,因此可在图1中的CRC电路100的通路展开单元130A到130N中校正数据包到路径的结尾的对齐。CRC单元300的另外的组件以与图2中的CRC单元200的那些组件类似的方式作用。例如,选择模块/多路复用器330根据数据包结束EoP\_Index值370选择将作为CRC\_A在输出端口360上输出的输入部分CRC值。类似地,选择模块/多路复用器340根据数据包起始SoP\_Index值380选择将作为CRC\_B在输出端口350上输出的输入CRC值。另外,合并器390中的每一个具有与图2中的合并器290相同的功能。

[0048] 返回到图1的CRC电路100,当使用对齐到512位路径的结尾的CRC单元时,可以包含通路展开单元130A到130N。例如,假定图1中的CRC单元110B包括图3的CRC单元300。如果数据包在CRC单元300的通路C(第二通路)上结束,则多路复用器330将选择输入2(来自通路C)作为输出CRC\_A。然而,推进单元320D和320C将基本上已经将其相应CRC值“过度推进”了两个(2)多余的通路。因此,通路展开单元130B可以将其作为输入接收的中间CRC值展开两个通路(256位)。通路展开单元可以基于如先前描述的字节展开单元150A到150N相同的或类似的设计。例如,展开运算可以包括根据生成函数和对应于空通路的数目的空位数对输入CRC值进行的矩阵处理。在一个实施例中,通路展开单元确定通路的数目以根据对应通路的EoP\_Index值来展开。举例来说,根据以上同一实例,如果数据包在通路B上结束,则EoP\_index=2。因此,通路展开单元将根据对应于两个(2)通路(256位)的位数来展开。应注意,由于数据包仍然可能已经在通路B的中间结束,所以仍然应将CRC值展开,展开程度为在字节展开单元150A到150N的一个单元中的结束通路内空字节的数目。

[0049] CRC电路100的其它功能,具体来说,推进单元120B、120N和选择单元140的功能,可

以使用图4的实例来说明。确切地说,图4示出了具有可以通过根据本发明实施例的CRC模块处理的512位输入的各个情境的时序图400。例如,五个实例A到E中的每个实例表示可以在任何一个时钟周期中接收到的/处理的512位。应注意,在这些实例中,位511是在数据流中“前面”的位且位0是在数据流中“后面”的位。例如,图1的电路100等CRC电路可以包括仅收发器这一个组件。在一个实施例中,收发器传输和/或接收串行数据,所述串行数据接着转化成并行数据以用于通过CRC电路和收发器的其它组件进行的更快处理。因此,随着串行地接收到数据,其可以放入数据字(其接着可以通过CRC电路分解成若干路径以用于通过若干512位CRC单元进行处理,如上文所描述)的连续的并行位位置中。在实例A中,在路径中(即,在通过512位CRC单元处理的512位内)不存在接收到的SoP或EoP信号。这可能归因于两个原因:在路径上不存在数据包/数据,或者是数据包持续通过路径(例如,当数据包大于512位时,数据包可以在通过其它512位CRC单元处理的不同路径上开始和结束)。在任何情况下,由于不存在在路径内结束的数据包,因此Valid\_A设置成零“0”以指示在路径内结束的数据包的CRC\_A输出不是有效输出。另一方面,Valid\_B设置成一“1”以指示在路径内起始或继续的数据包的CRC\_B输出是有效输出。

[0050] 在实例B中,在512位内存在SoP信号,但是仍然不存在EoP信号。因此,存在在路径中开始的数据包,但是没有数据包在路径内结束。由此,Valid\_A设置成零“0”且Valid\_B设置成一“1”。在实例C中,在512位内存在EoP信号,但是仍然不存在SoP信号。因此,Valid\_A设置成一“1”以指示在路径内结束的数据包的CRC\_A输出是有效的(因为在路径内结束的数据包由EoP信号指示)。然而,Valid\_B设置成零“0”以指示在路径内起始或继续的数据包的CRC\_B输出是无效输出。

[0051] 在实例D中,在512位内存在SoP信号和EoP信号两者。尽管在512位内存在SoP信号和EoP信号两者,但是仅Valid\_A设置成一“1”。这是因为EoP存在于SoP之后,从而清楚地指示EoP在以SoP信号开始的同一数据包中结束。由此,输出CRC\_B是无效的,并且因此不应与来自任何其它CRC单元的任何其它CRC值合并,因为数据包在所述路径内已经结束。在实例E中,在512位中再次存在SoP信号和EoP信号两者。然而,EoP信号比SoP信号更在前面,这表示EoP信号是针对从另一CRC单元/路径继续但是在所述电流路径内结束的数据包,而SoP信号是针对新数据包的起始。因此,Valid\_A设置成一“1”以指示CRC\_A(即在路径内结束的数据包的CRC值)是有效的。同样地,Valid\_B设置成一“1”以指示CRC\_B(即通过所述单元起始或连续的数据包的CRC值)也是有效的。

[0052] 现结合图4参考图1,电路100的数据总线可以包括1536位的字,所述字被分解为三个512位路径。作为实例,最前面的路径(路径N)包括如图4的实例B中所示的512位,中间路径(路径B)包括实例E的512位,且最后一个路径(路径A)包括实例C的512位。由此,路径N的CRC单元110N处理在所述数据路径上的位以生成CRC\_A和CRC\_B。另外,Valid\_A设置成零“0”且Valid\_B设置成“1”。因此,不应使用通过CRC单元110N输出的CRC\_A。选择单元140的用途是选择将作为一个或多个最终CRC值以在输出端口180A到180N上输出的有效CRC值。因此,基于针对路径N的Valid\_A信号/标志为零“0”,即,无效,选择单元140可以忽略或不选择将作为CRC单元110N的CRC\_A输出的CRC值,以作为最终CRC值来输出。推进单元120B和120N的用途是推进在电流路径内未结束的数据包的中间CRC值,以与来自其中数据包结束的另一路径/CRC单元的中间CRC值合并。因此,由于Valid\_B设置成“1”,即,有效,推进单元120N可

以输出/发送CRC\_B以与其它CRC单元的输出合并。

[0053] 应注意,推进单元120N以及推进单元120B和120X以图2和3中示出的推进单元类似的方式起作用。因此,举例来说,当推进单元120N将CRC\_B推进了一个通路(例如,推进到通路B)时,其将基于512位(到目的地通路的距离,以位为单位)来推进,且如果推进两个通路(例如,推进到通路A),则其将基于1024位来推进。

[0054] 在路径N上进行处理的同时,路径B中的CRC单元110B如图4的实例E中所示处理512位。CRC单元110B生成CRC值以用于作为中间CRC值CRC\_A和CRC\_B来输出。另外,存在EoP信号随后是SoP信号。由此Valid\_A和Valid\_B均设置成一“1”。值得注意的是,推进单元120N将从路径N到路径B输出/发送CRC\_B(如上文所描述,对于CRC单元110N/路径N,Valid\_B=1)。因此,来自CRC单元110N的CRC\_B在(例如)执行异或运算的合并器190处与来自CRC单元110B的CRC\_A合并。合并后的CRC值接着通过展开单元130B和选择单元140接收。另外,选择单元140将选择路径B上的合并后的CRC值以作为最终CRC值来输出。例如,选择单元140可以参考路径B的或沿着路径B的Valid\_A信号,其设置成“1”以指示数据包已经在当前CRC单元110B内结束。

[0055] 另外,新的数据包也在路径B上的数据内开始。由于Valid\_B也设置成一“1”,推进单元120B沿着图1中所指示的连接来推进CRC\_B。

[0056] 在CRC单元110A中,处理图4的实例C的512位并且输出CRC\_A和CRC\_B。在路径A上的数据内存在EoP信号。因此,Valid\_A=1而Valid\_B=0。CRC单元110A的CRC\_A的值在(例如)执行异或运算的合并器190处与来自CRC单元110B的CRC\_B合并。合并后的CRC值接着通过展开单元130A和选择单元140接收。由于Valid\_A=1,因此选择单元140将允许沿着路径A的合并后的CRC值作为最终CRC值来输出。值得注意的是,在此实例中,针对单个1536位的字,通过选择单元140并行输出两个CRC值以作为最终CRC值。

[0057] 尽管以上实例示出了本发明的实施例的若干特征和优点,但是还应指出,在一些情况下数据包可以跨越多个字。为了考虑此情形,CRC电路中的最后一个路径(例如,图1的CRC电路100中的路径A)的输出用作下一数据字的种子CRC值以在后续循环中进行处理。如图1中所示,值CRC\_accum\_new170包括到CRC\_accum寄存器160的输入。因此,例如,对于在CRC电路100的路径B中起始且在下一数据字内结束的数据包,推进单元120B将推进来自CRC单元110B的CRC\_B值以与CRC单元110A的CRC\_A和CRC\_B两者合并。来自上述路径B的CRC\_B与单元110A的CRC\_A合并。然而,由于数据包不在路径A上结束,因此CRC单元110A的CRC\_A将是无效(Valid\_A=0)且选择单元140简单地忽略了此合并后的CRC值。另一方面,来自单元110B的CRC\_B与来自单元110A的CRC\_B合并(例如,通过异或)并且此合并后的CRC值是有效的且是用于下一数据字的种子CRC值(CRC\_accum\_new 170)。

[0058] 还应注意,当对于特定CRC单元/路径的Valid\_B=1时,路径的推进单元可以将CRC\_B值推进到超过一个其它路径。例如,推进单元120N可以将CRC单元110N的CRC\_B值推进到路径B和A两者。然而,在路径N上起始或继续的数据包可以在路径B上结束而不在路径A上结束。因此,路径N的CRC\_B值仅应与路径B的单元110B的输出CRC\_A值合并并且应在路径A上被忽略。为了实现此用途,在一个实施例中,CRC电路100可进一步包含连续校验电路。

[0059] 图5示出了用于校验数据包是否在相邻路径/CRC单元之间连续的校验电路500的实例。如图5中所示,在路径A和B中分别存在两个相邻的CRC单元510A和510B。校验单元530

位于推进单元520B的输出与目的地通路A之间。在一个实施例中,校验单元包括多路复用器。在一个实施例中,到校验单元530的输入中的一个输入包括推进单元520B的输出且另一输入包括接地(例如,零“0”)。另外,在一个实施例中,校验单元530通过“数据包连续”的控制信号Pkt\_cont(j-1,i+1)来控制。

[0060] 如果数据包在单元i的上方结束,则推进后的CRC\_accum或CRC\_B可以不与单元i的CRC\_A合并。例如,如果数据包在单元(j-1)与单元(i+1)之间连续,则来自单元j的推进后的CRC\_B将与单元i的CRC\_A合并。另外,如果数据包在顶部单元与单元(i+1)之间连续,则推进后的CRC\_accum将与单元i的CRC\_A合并。类似地,如果数据包在当前数据字内结束,则推进后的CRC\_accum或CRC\_B可不用以生成新的CRC\_accum。例如,如果数据包在单元(j-1)与单元0之间连续,则来自单元j的推进后的CRC\_B将用于累计。如果数据包在顶部单元与单元0之间连续,则推进后的CRC\_accum(来自上一数据字)也将用于累计。在一个实施例中,通过查看在单元之间的SoP和EoP位确定数据包是否在所述单元之间连续。因此,如图5中所示,校验单元530确定是否应传送推进后的CRC(CRC\_B),其中Pkt\_cont(j-1,i+1)是基于SoP和EoP信号生成的信号以指示数据包是否在单元(j-1)与(i+1)之间(例如,分别在单元510A与单元510B之间)连续。应注意,多个类似的校验单元可以位于不同路径之间从而以类似方式执行不同的连续性校验。

[0061] 除决定是否应将推进后的CRC传送到单元,在一个实施例中,还必需决定单元是否想要接受推进后的CRC。就数据包在同一单元内起始和结束的情况下,此单元的CRC\_A不需要与任何推进后的CRC合并。图6示出在路径上和/或通过CRC单元处理的位的几种示例性情境的时序图600及关于是否应接受推进后的CRC值的确定。在一个实施例中,可能在路径内呈现的SoP和EoP信号可确定单元/路径是否应接受来自更高层单元和来自CRC\_accum寄存器的推进后的CRC。此处“\*”意味“不考虑”,因为单元的CRC\_A输出在这些情况下是无效的。在一个实施例中,“接受推进”Accept\_Fwd信号包括单个位,其可用作到图5的校验单元530的额外控制信号。在另一实施例中,单独的控制逻辑可定位于同一路径中以完成根据Accept\_Fwd信号的推进或阻挡。

[0062] 图7示出用于执行循环冗余校验的方法700的方块图。例如,方法700可以通过根据上述实施例中的任何一个或多个实施例的电路/装置来执行。在一个实施例中,根据本发明的计算机可读媒体存储一系列指令,所述指令使得可编程装置经编程以执行方法700的步骤、功能或操作。所述方法在步骤702中开始并前进到步骤710。

[0063] 在步骤710中,方法700接收输入数据字。例如,所述方法可以经由1024位宽或更宽的数据总线接收并行数据。

[0064] 在步骤720中,方法700将输入数据字分解成多个路径。例如,所述方法可以将数据字分解成若干相同大小的路径以用于并行处理。在一个实施例中,路径中的每个路径是512位宽。

[0065] 在步骤730中,方法700(例如)以并行方式处理路径中的每个路径。在一个实施例中,处理路径中的每个路径包含计算在路径内结束的数据包的循环冗余校验(CRC)值,以及计算在路径内起始或持续的数据包的循环冗余校验值。

[0066] 在一个实施例中,针对每个路径,所述处理包括将路径分解成多个通路。举例来说,通路中的每个通路可以包括来自所述路径的相同数目的有顺序字节。在一个实施例中,

针对每个通路,所述方法计算部分CRC值并且将所述部分CRC值推进到“下面”的通路(概括地,其它通路)。在一个实施例中,推进包括根据从当前通路到目的地通路的距离对部分CRC值进行的矩阵变换。在一个实施例中,针对每个通路,所述方法(例如)使用异或运算进一步合并针对所述通路计算出的CRC值与从任何“上面”的通路(概括地,其它通路)推进到所述通路的任何CRC值。最后,所述方法可以根据在特定通路中可检测到的数据包信号的结束而从所述通路中的一个通路选择合并后的CRC值,以作为在所述路径内结束的数据包的CRC值来输出。同样地,所述方法可以根据默认条件(例如,其中在所述通路的任何通路中未检测到数据包信号的起始,其中假定数据包未在所述路径中开始并且在上面的一些路径中或在前一数据字中起始)或根据所述通路中的一个通路内的数据包信号的起始的检测,选择通过所述路径起始或持续的数据包的合并后的CRC值来输出。

[0067] 在步骤740中,所述方法将CRC值从至少第一路径推进到至少第二路径。例如,数据包可以在第一路径中起始或持续。因此,在所述路径内起始或持续的数据包的部分CRC值与“下面”的一个或多个路径(例如,其中数据包结束的路径)的其它部分CRC值合并。在一个实施例中,推进包括根据目的地路径对部分CRC值进行的矩阵变换。举例来说,如果目的地路径是电流路径前方的两个路径,则矩阵变换可以是基于每路径的位数的两倍。应注意,在一个实施例中,步骤740包括将CRC值从多个路径推进到多个不同的目的地路径。另外,在一个实施例中,在步骤740处来自前一数据字的累计CRC值也可以推进到路径中的一个或多个路径。

[0068] 在步骤750中,方法700合并至少在第一路径内起始或持续的数据包的至少第一CRC值与在至少第二路径内结束的数据包的至少第二CRC值。在一个实施例中,合并包括对两个CRC值执行异或运算。在一个实施例中,步骤750包含首先校验数据包是否在两个路径之间连续。例如,同一数据包可以在第一路径上起始并且在第二路径上结束。因此,合并来自所述两个路径的CRC值。然而,如果不存在同一数据包跨越两个路径的连续性,则不应合并CRC值。替代地或另外,在一个实施例中,步骤750可以包括合并来自前一数据字的累计CRC值与来自当前数据字的一个或多个路径的一个或多个CRC值。

[0069] 在任意的步骤760处,方法700输出累计CRC值。例如,如果数据包在数据字的上一路径中持续或连续且所述数据包在当前数据字内未结束,则应输出并存储累计CRC值以用于方法700关于下一数据字(其中包数据可能结束)的后续迭代。

[0070] 在步骤770处,方法700展开至少一个CRC值。例如,数据包可以在通路内的任何位置处结束。因此,可能需要调整CRC值以考虑在数据包的结尾与通路的结尾和/或路径的结尾之间的任何空位。例如,在一个实施例中,展开包括根据生成函数/多项式和空位的数目对CRC值进行的矩阵变换。在一个实施例中,步骤700包括展开两个或更多个CRC值。举例来说,在本发明的实施例中,两个或更多个数据包可能在同一数据字内结束。因此,在步骤770处可以根据相应的空位数关于CRC值中的每个值执行单独的展开运算。

[0071] 在步骤780处,所述方法选择一个或多个CRC值并且将所述CRC值作为最终CRC值输出。举例来说,一个、两个或若干个数据包可能全部在同一数据字内结束。例如,如果字长是N,路径大小是512且最小包大小也是512,则在同一字内可能结束的数据包的数目是 $n=N/512$ 。如果 $N=1536$ ,则存在三个路径且可能有三个数据包在所述字内结束。然而,尽管路径中的每个路径可以具有CRC输出,但是并非所有输出都是必定有效的最终CRC值。举例来说,

数据包可以在第一路径内和在最后一个路径内结束,但是不在中间路径中结束。因此,在步骤780处,方法700可以仅选择有效的那些CRC值以作为最终CRC值来输出。

[0072] 在步骤780之后,方法700前进到步骤795,所述方法结束。

[0073] 应注意,尽管未具体地指定,但是方法700的一个或多个步骤或操作可包含针对特定应用所需要的存储、显示和/或输出步骤。换句话说,在所述方法中讨论的任何数据、记录、字段和/或中间结果可针对特定应用所需而存储、显示和/或输出到另一装置。此外,列举确定操作或涉及决策的图7中的步骤、操作或区块不一定需要对确定操作的两个分支都进行实践。换句话说,确定操作的分支中的一个分支可被认为是可选步骤。

[0074] 图8描绘适用于执行本文所述的功能中的一些或全部的通用计算机、计算装置或计算系统800的高阶方块图。如图8中所描绘,系统800包括处理器元件802(例如,CPU)、存储器804(例如,随机存取存储器(RAM)和/或只读存储器(ROM))、用于执行循环冗余检验的模块805、及各种输入/输出装置806(例如,存储装置,包括但不限于存储器装置、磁带驱动器、软驱、硬盘驱动器或光盘驱动器、接收器、发射器、扬声器、显示器、语音合成器、输出端口,及用户输入装置(例如键盘、小键盘、鼠标等))。

[0075] 本发明的实施例可以软件和硬件的组合来实施,例如,使用专用集成电路(ASIC)、例如FPGA等PLD、通用计算机或任何其它硬件等效物(例如微处理器)。在一个实施例中,用于执行循环冗余校验的本发明的模块或过程可以加载到存储器804中并且通过处理器802执行,以实施如上文所论述的功能。由此,本发明的实施例的用于执行循环冗余校验(包含相关联的数据结构)的本发明的模块或过程805可以存储在非暂时性计算机可读存储媒体上,例如,RAM存储器、磁性或光盘驱动器或磁盘等。

[0076] 本发明的实施例也可以完全或部分地通过可调IC来实施,例如,PLD和/或FPGA。更确切地说,可编程逻辑装置(PLD)是可以由用户编程以实施各种所选择功能的通用装置。一种类型的PLD是现场可编程门阵列(FPGA),其通常包含可配置逻辑块(CLB)阵列和多个输入/输出块(IOB)。CLB是可个别地编程的并且可经配置以对若干输入信号执行各种逻辑功能。IOB可经配置以将来自CLB的输出信号驱动到FPGA的外部识别码和/或接收来自外部FPGA识别码的输入信号。FPGA还包含可编程互连结构,其可经编程以选择性地在各个CLB和IOB之中路由信号以产生许多输入信号的更多复杂功能。通过将配置数据(例如,来自通用计算机800)加载到相关联的存储器单元(其控制CLB、IOB和互连结构内的各个交换机和多路复用器)中对CLB、IOB和可编程互连结构进行编程,以实施所述配置数据所指定的逻辑和路由功能从而在FPGA中实施用户设计。FPGA还可包含其它可编程和非可编程资源。由此,以上结合图1到3示出的电路可以在多个CLB中实施,所述多个CLB执行图1到3的电路中的任何一个或多个电路的任何一个或多个组件的等效逻辑运算。例如,用于对经由1536位宽的数据总线接收到的数据包执行CRC的本发明的实施例已经在使用大致15,000个查找表(LUT)、3,000个寄存器以及4,500个芯片(slice)的FPGA上实施。

[0077] 虽然上文根据本发明的一个或多个方面描述了示例性实施例,但是可以不脱离本发明范围的情况下设计根据本发明的一个或多个方面的其它和进一步实施例,所述范围通过所附权利要求书及其等效物确定。权利要求书列出的步骤并不暗示步骤的任何次序。商标是它们的对应拥有者的财产。

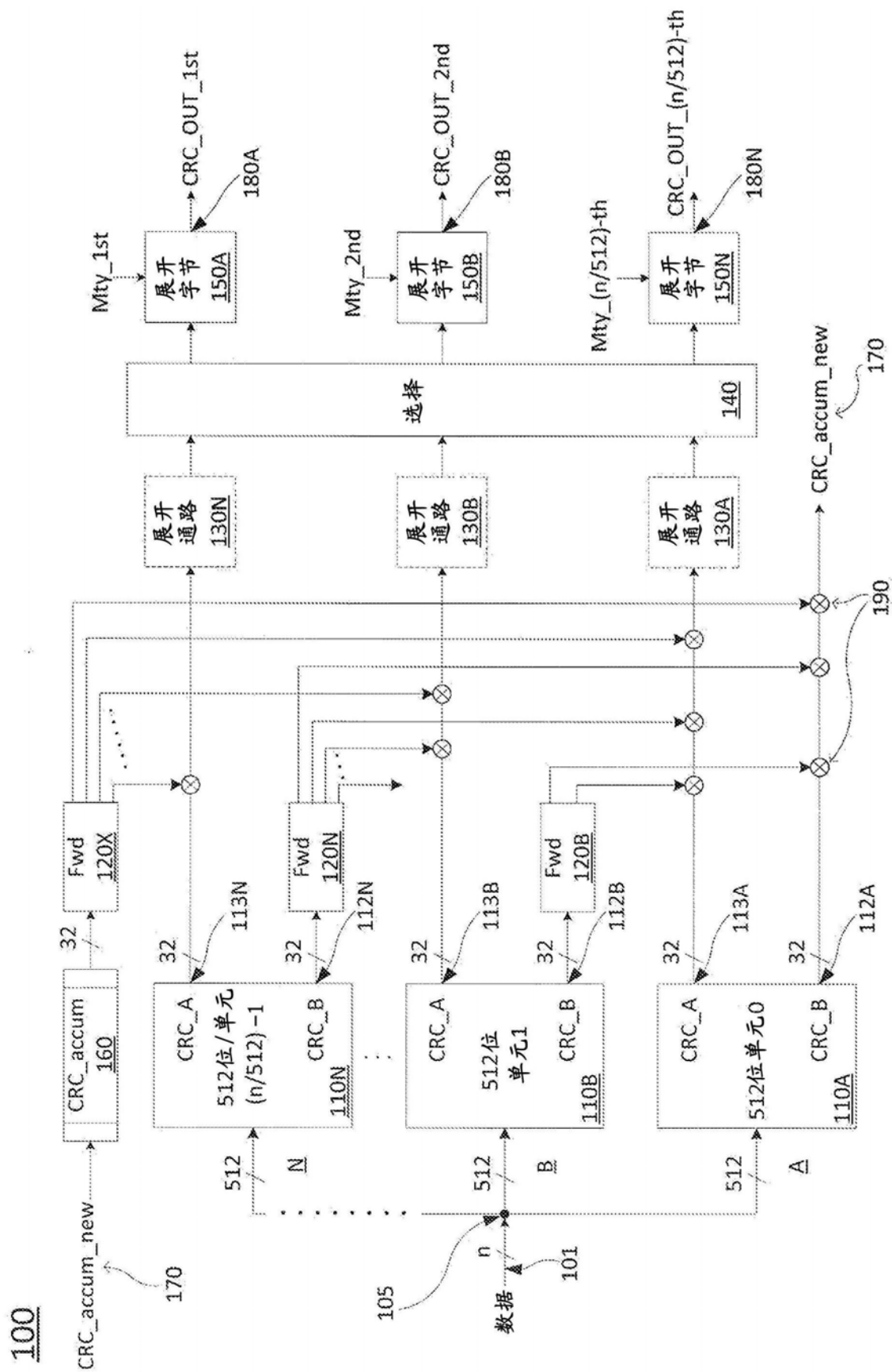


图1



200

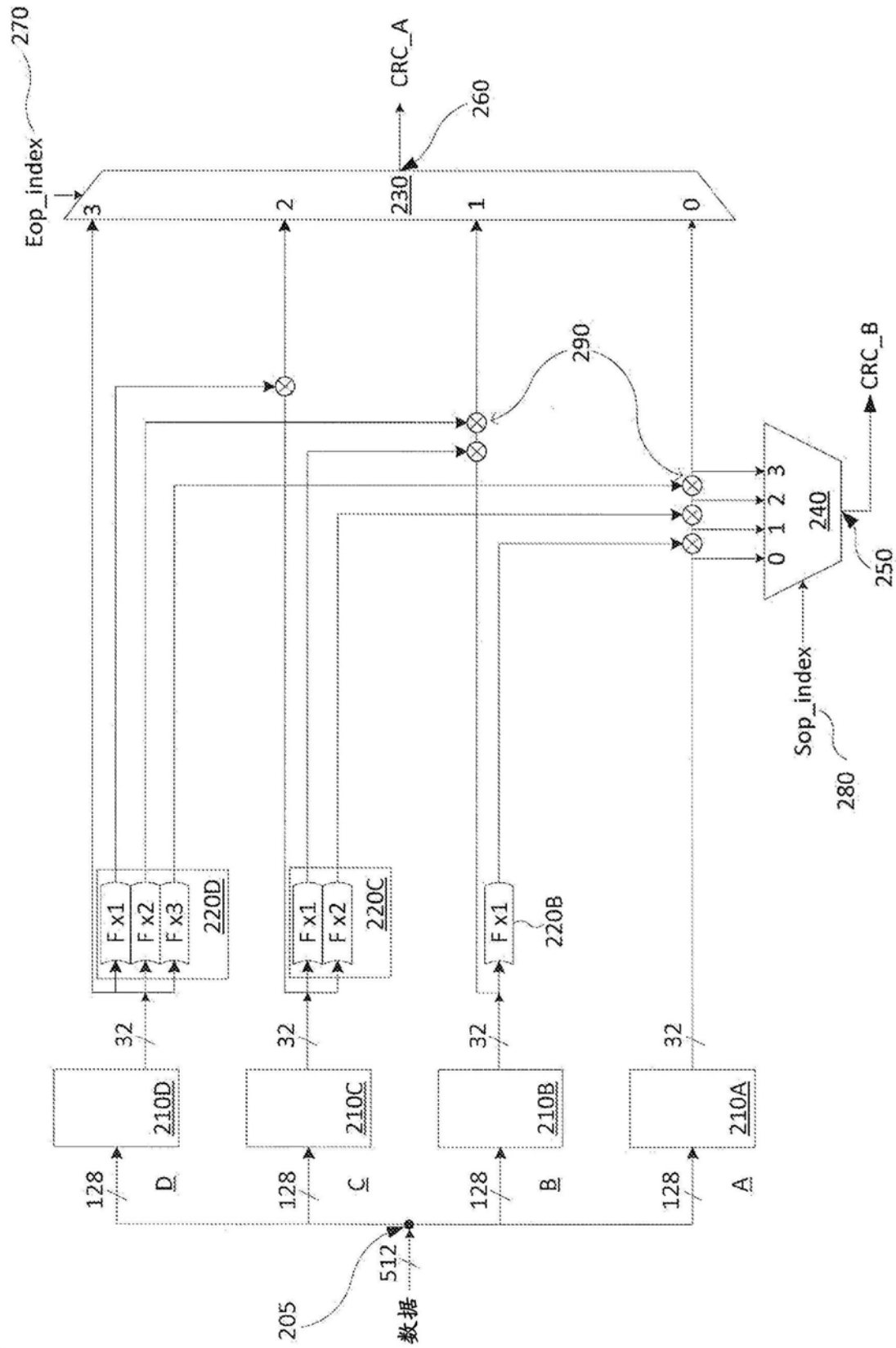


图2

300

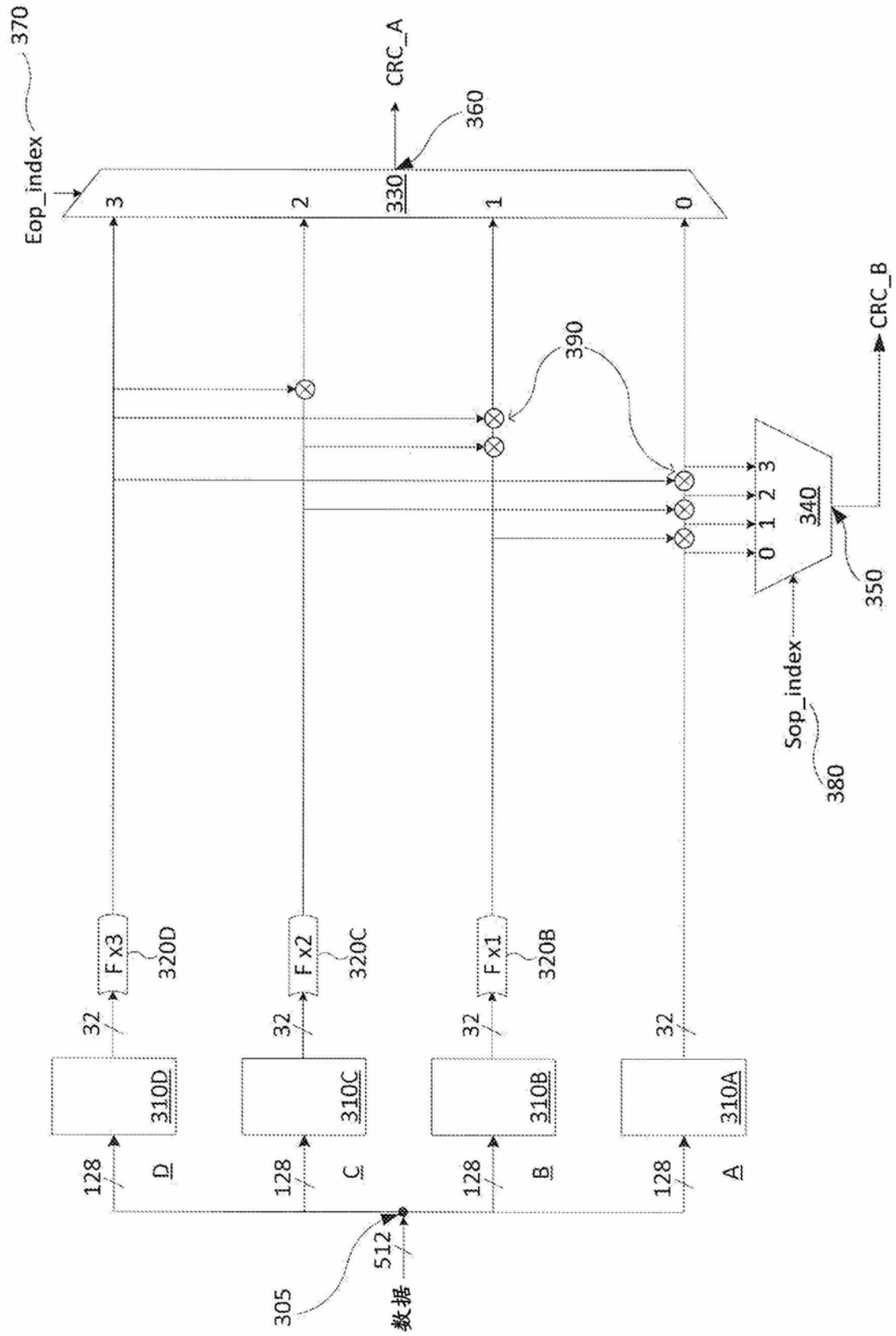


图3

400

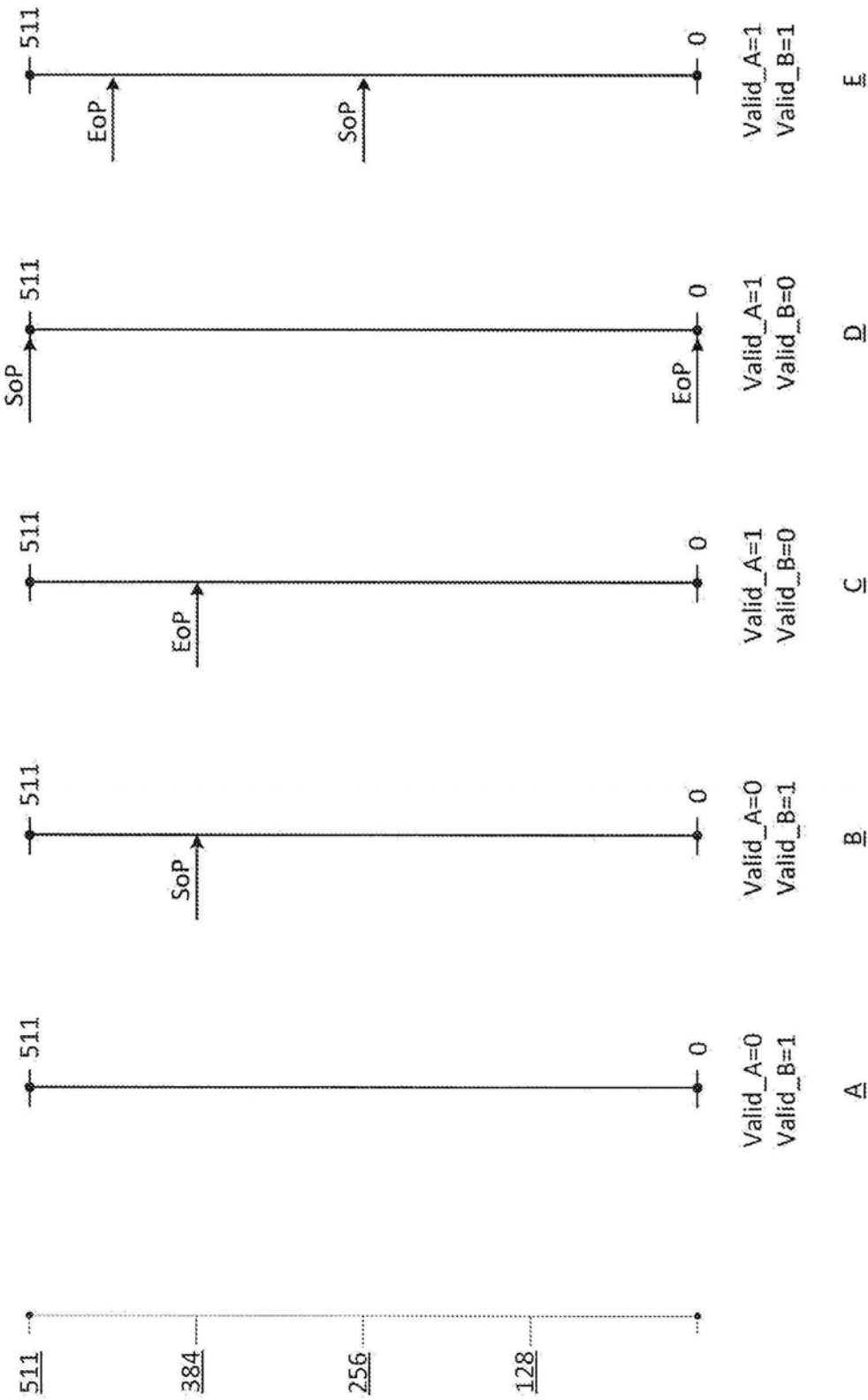


图4

500

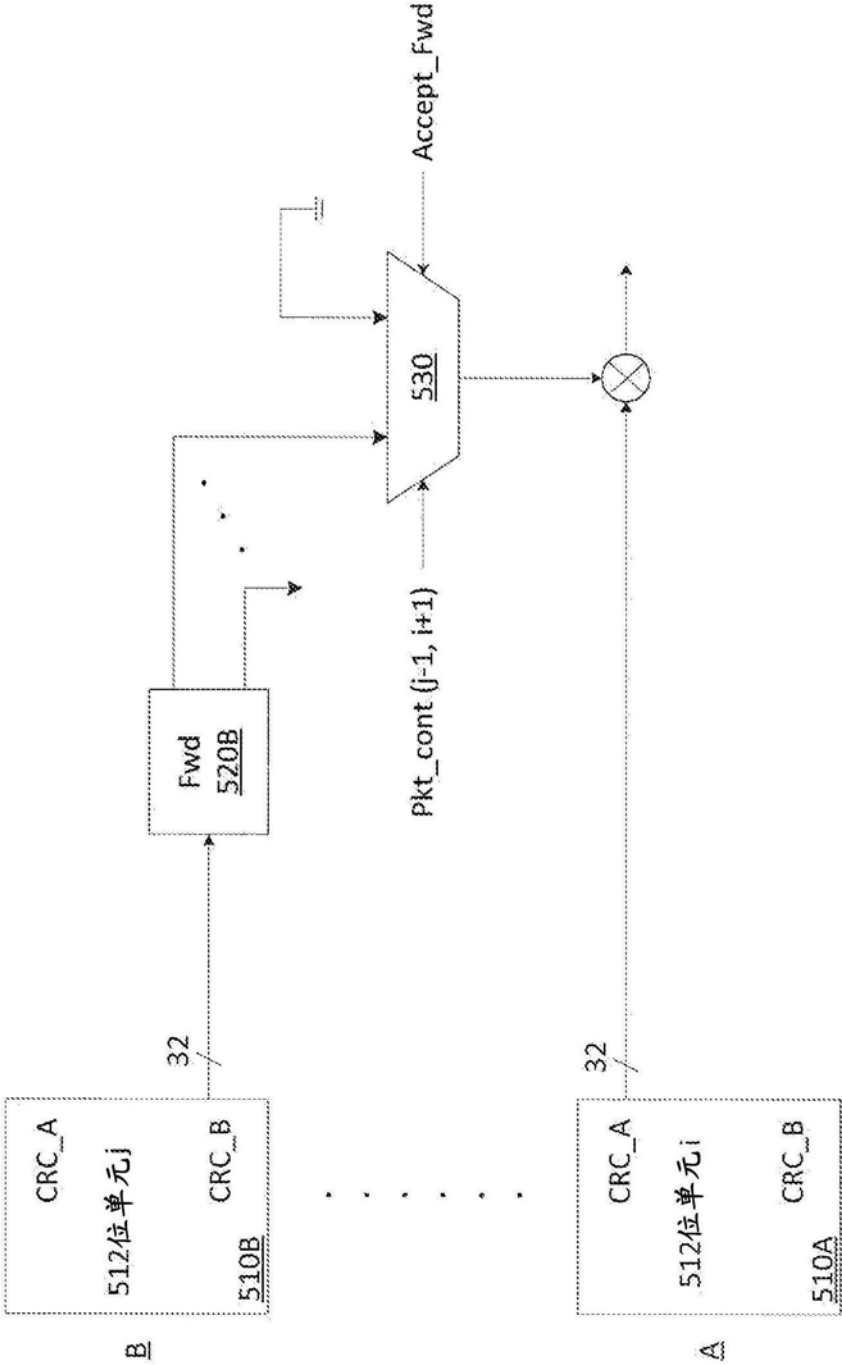
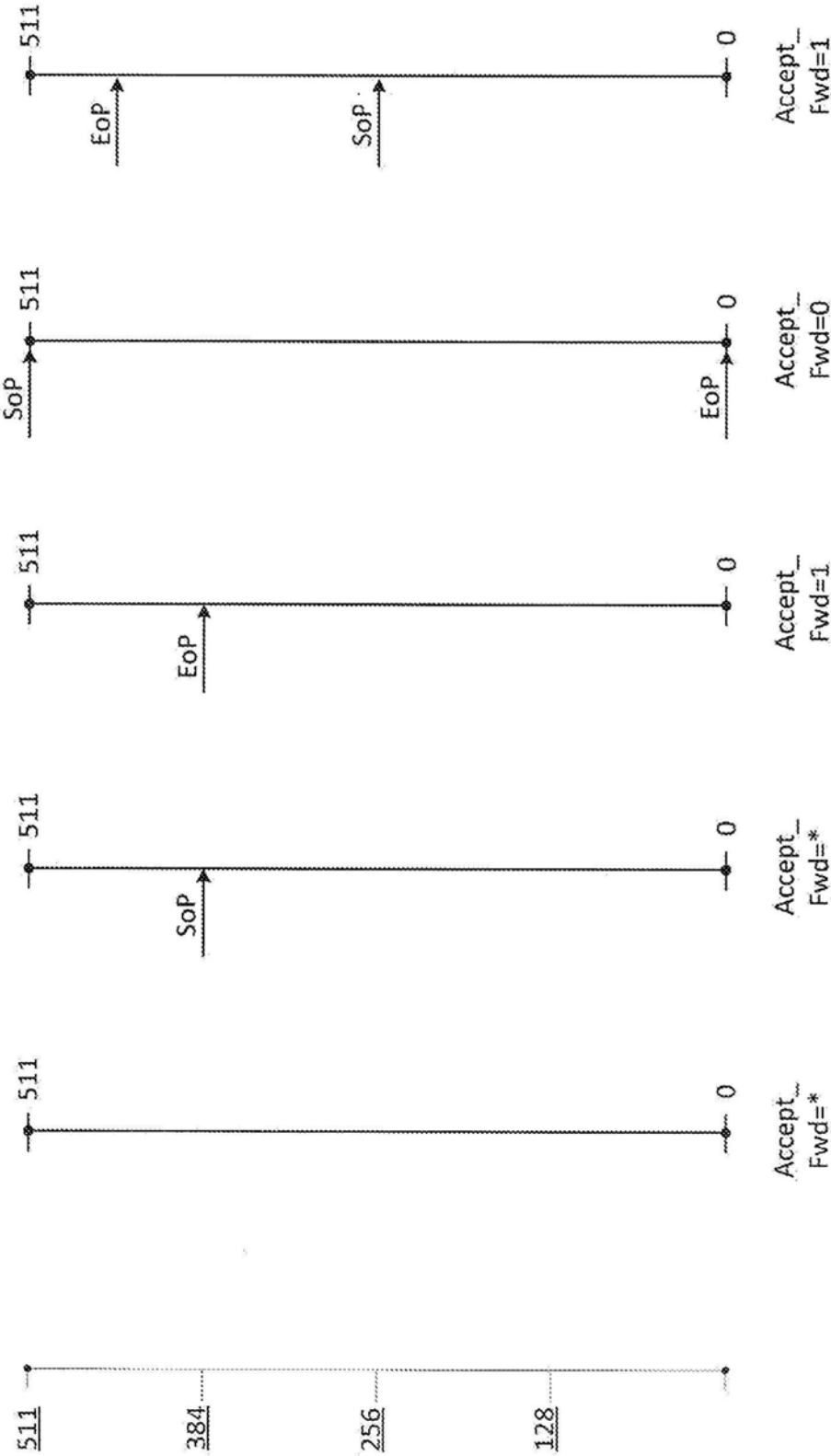


图5



600

图6

700

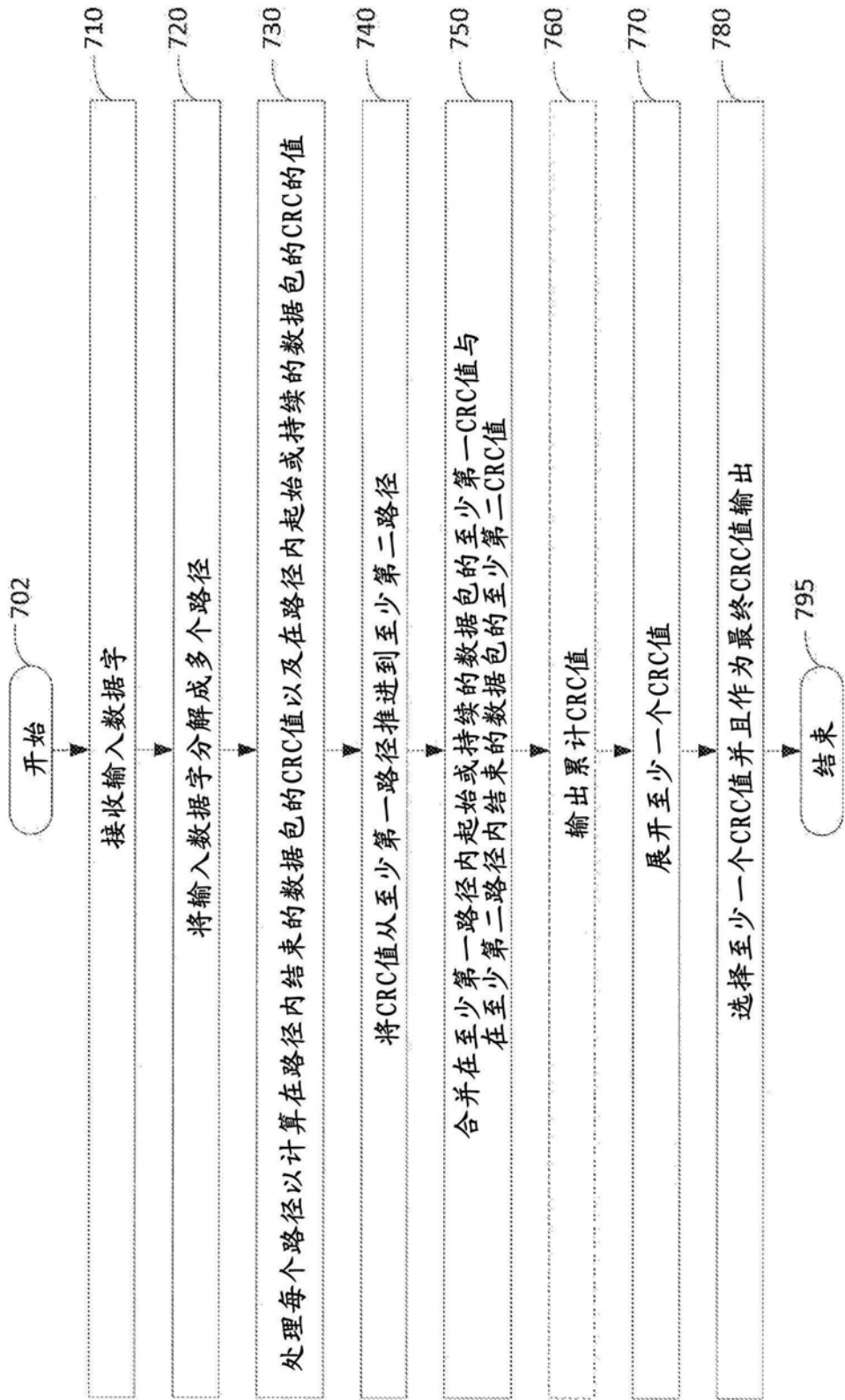


图7

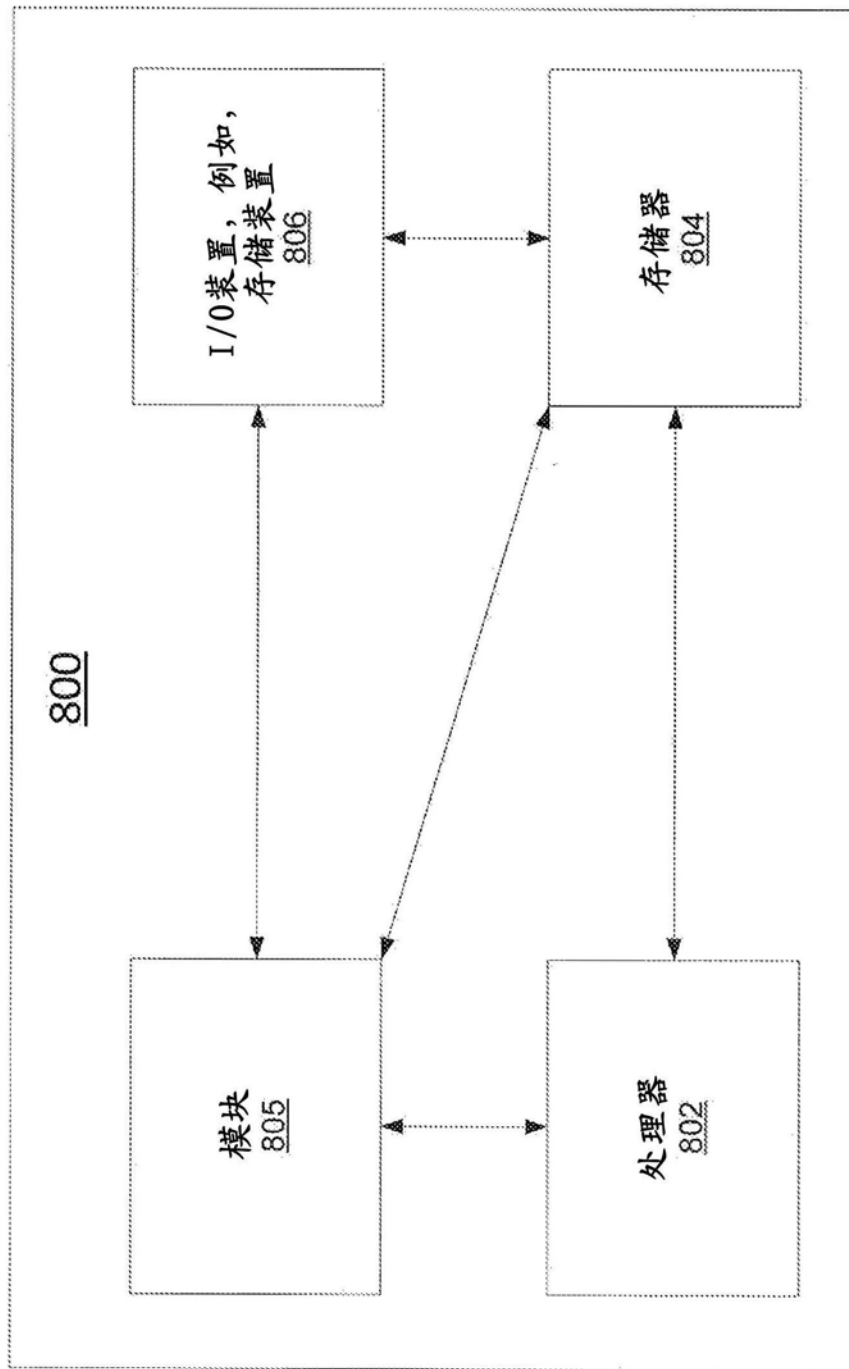


图8