(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 104285378 B
(45)授权公告日 2017.09.22

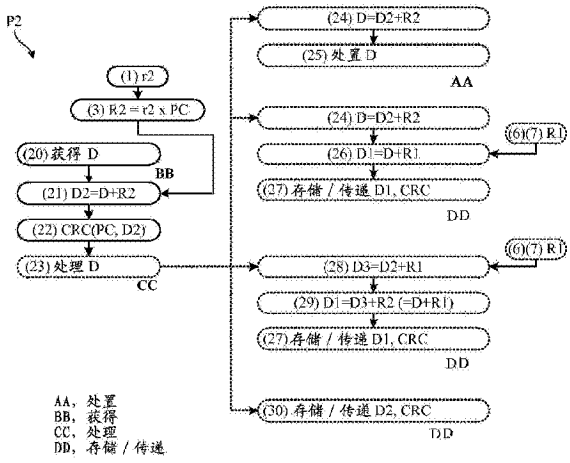
(21)申请号 201380022194.9
(22)申请日 2013.03.26
(65)同一申请的已公布的文献号
申请公布号 CN 104285378 A
(43)申请公布日 2015.01.14
(30)优先权数据
1253804 2012.04.25 FR
(85)PCT国际申请进入国家阶段日
2014.10.27
(86)PCT国际申请的申请数据
PCT/FR2013/050646 2013.03.26
(87)PCT国际申请的公布数据
WO2013/160575 FR 2013.10.31
(73)专利权人 英赛瑟库尔公司
地址 法国梅吕埃
(72)发明人 M·罗瑟莱特 V·文森特
(74)专利代理机构 北京市中咨律师事务所 11247
代理人 杨晓光 于静
(51)Int.Cl.
*H03M 13/09*(2006.01)
*G06F 21/55*(2006.01)
*H04L 9/00*(2006.01)
(56)对比文件
US 5377270 A,1994.12.27,全文.
US 2004264468 A1,2004.12.30,全文.
US 2010077225 A1,2010.03.25,全文.
US 2005154960 A1,2005.07.14,全文.
US 7523305 B2,2009.04.21,全文.
审查员 刘蕾蕾
权利要求书3页 说明书10页 附图5页

(54)发明名称

具有免受侧信道攻击保护的循环冗余校验方法

(57)摘要

本发明涉及一种用于处理二进制数据项(D)的方法,包括借助于生成多项式(PC)来计算针对所述数据项的循环冗余校验码的步骤,其中所述计算所述循环冗余校验码的步骤包括以下步骤:使用随机的二进制掩码(R2)对所述数据项进行掩盖,所述随机的二进制掩码(R2)是所述生成多项式的倍数,以及从所述掩盖的数据项(D2)来生成针对所述数据项的所述循环冗余校验码,掩盖过程在于使所述随机二进制掩码(R2)与所述二进制数据项相加。




CN 104285378 B

1.一种用于在电子设备中处理二进制数据项的方法,所述方法包括:

接收所述二进制数据项;以及

使用生成多项式来计算针对所述二进制数据项的循环冗余校验码,其中计算所述循环冗余校验码包括:

-使用多项式加法,使用二进制掩码对所述二进制数据项进行掩盖,所述二进制掩码是所述生成多项式的随机倍数,以及

-使用所述生成多项式,从所掩盖的二进制数据项来生成针对所述二进制数据项的所述循环冗余校验码。

2.根据权利要求1所述的方法,还包括:

-将所述二进制数据项分成至少两个部分,

-使用第一二进制掩码对所述二进制数据项的第一部分进行掩盖,所述第一二进制掩码是所述生成多项式和第一随机数的相乘,

-生成针对第一掩盖的部分的第一循环冗余校验码,

-将所述第一循环冗余校验码与所述二进制数据项的下一个部分进行连接,以形成中间的数据项,

-使用第二二进制掩码对所述中间的数据项进行掩盖,所述第二二进制掩码是所述生成多项式和第二随机数的相乘,以及

-生成针对掩盖的中间的数据项的第二循环冗余校验码。

3.根据权利要求1所述的方法,还包括生成所述二进制掩码,所述生成所述二进制掩码包括:

-生成随机数,以及

-使所述随机数与所述生成多项式相乘。

4.根据权利要求1所述的方法,其中所述二进制数据项包括N个比特,以及所述生成多项式包括P个比特,以及随机数包括至少N-P个比特。

5.根据权利要求1所述的方法,还包括生成所述二进制掩码,所述生成所述二进制掩码包括:

-生成多个随机数,

-使所述多个随机数中的每个随机数与所述生成多项式相乘以获得多个各自的掩码部分,以及

-将所述多个各自的掩码部分进行连接。

6.根据权利要求5所述的方法,其中所述二进制数据项包括N个比特,以及所述生成多项式包括P个比特,以及所述多个随机数中的每个随机数包括等于(N-nP)/n的比特数目,n是被包含在所述多个随机数中的随机数的数目。

7.根据权利要求1至6中的一项所述的方法,其中所述二进制数据项包括使用存储或传递掩码所掩盖的二进制数据项,所述存储或传递掩码是随机二进制掩码,在使用所述生成多项式来计算针对所述二进制数据项的循环冗余校验码之前,所述方法还包括以下步骤:

-与使用存储或传递掩码所掩盖的二进制数据项一起,接收针对使用存储或传递掩码所掩盖的二进制数据项的循环冗余校验码,

-从使用存储或传递掩码所掩盖的二进制数据项移除存储或传递掩码以获得所述二进

制数据项，以及

-在使用所述生成多项式生成针对所述二进制数据项的所述循环冗余校验码后，将针对使用存储或传递掩码所掩盖的二进制数据项的循环冗余校验码与使用所述生成多项式所生成的针对所述二进制数据项的所述循环冗余校验码进行比较。

8.根据权利要求1至6中的一项所述的方法，其中所述二进制数据项包括使用存储或传递掩码所掩盖的二进制数据项，所述存储或传递掩码是随机二进制掩码，所述方法还包括以下步骤：

-在使用所述生成多项式来计算针对所述二进制数据项的循环冗余校验码之前，与使用存储或传递掩码所掩盖的二进制数据项一起，接收针对使用存储或传递掩码所掩盖的二进制数据项的循环冗余校验码，

-在使用多项式加法，使用二进制掩码对使用存储或传递掩码所掩盖的二进制数据项进行掩盖之后，并且在使用所述生成多项式，从使用所述二进制掩码所掩盖的二进制数据项来生成针对所述二进制数据项的循环冗余校验码之前，从使用存储或传递掩码和所述二进制掩码两者所掩盖的二进制数据项移除存储或传递掩码，其中所述二进制掩码是所述生成多项式和随机数的相乘，以及

-将从使用二进制掩码所掩盖的二进制数据项生成的针对所述二进制数据项的循环冗余校验码与针对使用存储或传递掩码所掩盖的二进制数据项的循环冗余校验码进行比较，其中所述二进制掩码是所述生成多项式和随机数的相乘。

9.根据权利要求1所述的方法，其中从所掩盖的二进制数据项生成的针对所述二进制数据项的循环冗余校验码是第一循环冗余校验码，以及所述二进制数据项包括使用存储或传递掩码所掩盖的二进制数据项，所述存储或传递掩码是随机二进制掩码，所述方法还包括以下步骤：

-在使用所述生成多项式，计算针对所述二进制数据项的所述循环冗余校验码之前，与使用存储或传递掩码所掩盖的二进制数据项一起，接收第二循环冗余校验码，所述第二循环冗余校验码是针对使用存储或传递掩码所掩盖的二进制数据项的循环冗余校验码，以及

将所述第一循环冗余校验码和所述第二循环冗余校验码进行比较。

10.一种用于处理二进制数据项的电子设备，其中所述电子设备被配置为实现根据权利要求1至9中的一项的所述方法。

11.一种便携式物品，包括根据权利要求10的电子设备，其中所述电子设备被制造在半导体芯片上。

12.一种用于存储或传递数据项的方法，包括：

接收所述数据项，

使用多项式加法，使用二进制掩码对所述数据项进行掩盖，所述二进制掩码是用于生成循环冗余校验码的生成多项式的随机倍数，

-使用所述生成多项式，从所掩盖的数据项来生成循环冗余校验码，以及

在电子设备中存储所掩盖的数据项和所述循环冗余校验码以对所掩盖的数据项进行解码，或者向电子设备传递所掩盖的数据项和所述循环冗余校验码以对所掩盖的数据项进行解码。

13.根据权利要求12所述的方法，还包括通过以下来生成所述二进制掩码：

-生成随机数，以及

-使所述随机数与所述生成多项式相乘。

14.根据权利要求12所述的方法，还包括通过以下来生成所述二进制掩码：

-生成多个随机数，

-使所述多个随机数中的每个随机数与所述生成多项式相乘以获得多个各自的掩码部分，以及

-将所述多个各自的掩码部分进行连接。

15.一种电子设备，包括电路，所述电路被配置为根据权利要求12至14任一所述方法来存储或传递数据项。

# 具有免受侧信道攻击保护的循环冗余校验方法

[0001] 本发明涉及用于处理二进制数据项的方法，该方法包括借助于生成多项式来计算循环冗余校验码的步骤。本发明还涉及保护装备有半导体的集成电路免受侧信道攻击。

[0002] 在受保护的应用中使用的半导体芯片上的集成电路受到各种攻击，特别是基于观察它们的电流消耗或它们的磁场或电磁场辐射的侧信道攻击。此类攻击目的在于发现它们处理的敏感数据，例如，加密密钥、应用数据、中间的计算变量等。

[0003] 最频繁的侧信道攻击实现统计分析方法，诸如DPA(“差分功率分析”)或CPA(“相关功率分析”)分析。这些分析技术使得通过获取许多电路的消耗曲线能够发现加密算法的密钥。DPA由根据关于寻找的密钥的假设的电流消耗曲线的统计分类组成。CPA基于电流消耗模型，以及涉及计算一方面所测量的消耗点与另一方面估计的消耗值之间的相关系数，从消耗模型以及关于由电路执行的操作的假设来计算。

[0004] 一般实现各种硬件和/或软件对策以对抗此类攻击。特别地，一般以掩盖的形式来存储或传递敏感数据，即在与被认为由攻击者未知的二进制掩码结合后，以及一般仅在它们被充分保护免受攻击时才不掩盖敏感数据。

[0005] 其次，由集成电路处理的数据一般被保护免受数据损坏，数据损坏可能是偶然的或由故意的动作而导致。数据损坏可以例如在存储单元(在存储单元中存储数据)的阈值电压改变期间发生，或在当在数据总线上运送数据时的电磁干扰期间发生，或在通过错误注入的攻击期间发生。在它的存储或传递期间，因此数据伴随有被称为“校验和”的错误检测码。错误检测码可以例如是奇偶校验位、汉明码、CRC码(“循环冗余校验”)等。

[0006] 图1示意性地示出了用于处理借助于码CRC来结合掩盖和错误检测的数据项D的常规方法P1。该方法包括处理数据项的初始阶段以及随后的处理阶段。初始阶段包括：

[0007] -步骤10，获取数据项D，

[0008] -步骤11，借助于生成多项式PC来计算数据项D的码CRC，

[0009] -步骤12，通过将数据项D与掩码R1相加，使数据项与掩码R1掩盖，以获得掩盖的数据项D1，以及

[0010] -步骤13，处理数据项D。

[0011] 在步骤11中，通过数据项除以多项式PC的多项式除法来计算码CRC，在GF(2)[X](即多项式域)中进行，多项式的系数属于有限域GF(2)(伽罗瓦域)，其形成已知的最小的有限域。在步骤12中执行的加法是在布尔代数中对应于掩码R1与数据项D的逐位异或的GF(2)[X]中的多项式加法。最后，处理步骤13可以在于在存储器中存储数据项，或将它传送给除了已经执行步骤10至12的实体之外的实体。在这种情况下，以它的掩盖的形式D1以及伴随有码CRC来存储和传送该数据项。

[0012] 随后的处理阶段包括：

[0013] -步骤14，获取掩盖的数据项D1以及它的码CRC(接收或在存储器中读取)，

[0014] -步骤15，去除数据项D的掩盖，即通过使数据项D1与掩码R1相加，移除掩码R1，

[0015] -步骤16，借助于多项式PC，计算数据项D的码CRC’，

[0016] -步骤17，比较码CRC和CRC’

[0017] -如果码CRC和码CRC’不同，则错误处理步骤18，以及

[0018] -如果码CRC和码CRC’相同，则处理数据项D的步骤19。

[0019] 本发明的一些实施例基于在这个数据处理方法中的安全故障的发现。这种安全故障涉及从未掩盖的数据项执行的计算码CRC的步骤11或步骤16，在没有未掩盖的数据项的情况下，码CRC将是无效的。现在，我们已经发现了这个计算步骤易于受到某些类型的攻击，该攻击能够使得数据项D的值被发现。特别地，“模板”类型的攻击是可能的。该攻击涉及根据数据项D的值形成执行码CRC的计算的电路的电流消耗的所有刨面图的数据库。于是，在码CRC的计算期间的电路的消耗刨面图的观察使得在该数据库中能够找到数据项D的对应值。在某些应用中，在码CRC的计算期间的DPA类型的攻击也是可能的。

[0020] 因此，期望的是改进由包含码CRC计算的数据处理方法提供的安全性。

[0021] 本发明的一些实施例涉及用于处理二进制数据项的方法，该方法包括借助于生成多项式来计算针对数据项的循环冗余校验码的步骤，其中计算循环冗余校验码的步骤包括以下步骤：使用随机二进制掩码对该数据项进行掩盖，随机二进制掩码是该生成多项式的倍数，以及从被掩盖的数据项生成针对该数据项的循环冗余校验码。

[0022] 根据一个实施例，该方法包括以下步骤：将数据项分成至少两部分，使用第一随机二进制掩码对该数据项的第一部分进行掩盖，该第一随机二进制掩码是生成多项式的倍数，生成针对第一掩盖的部分的第一循环冗余校验码，将第一循环冗余校验码与该数据项的下一个部分进行连接，以形成中间的数据项，使用第二随机二进制掩码对中间的数据项进行掩盖，该第二随机二进制掩码是生成多项式的倍数，以及生成针对掩盖的中间数据项的第二循环冗余校验码。

[0023] 根据一个实施例，该方法包括生成随机二进制掩码的步骤，生成随机二进制掩码的步骤包括生成任何随机数的步骤，以及使该任何随机数与生成多项式相乘。

[0024] 根据一个实施例，数据项包括N比特，以及生成多项式包括P比特，以及生成的任何随机数包括至少N-P比特。

[0025] 根据一个实施例，该方法包括生成随机二进制掩码的步骤，生成随机二进制掩码的步骤包括以下步骤：生成第一任何随机数，使该第一任何随机数与生成多项式相乘以获得第一掩码部分，生成至少一个第二任何随机数，使该第二任何随机数与生成多项式相乘以获得至少一个第二掩码部分，以及将第一掩码部分与第二掩码部分连接。

[0026] 根据一个实施例，数据项包括N比特，以及生成多项式包括P比特，以及每个生成的任何随机数包括等于(N-nP)/n的比特数目，n是生成的多个任何随机数的数目。

[0027] 根据一个实施例，该方法包括以下步骤：获得以使用形成存储或传递掩码的第一任何随机的二进制掩码进行掩盖的形式的数据项，连同第一循环冗余校验码，从该数据项移除第一掩码，使用第二随机的二进制掩码对该数据项进行掩盖，该第二随机二进制掩码是生成多项式的倍数，从使用第二掩码进行掩盖的数据项来生成第二循环冗余校验码，以及将第一循环冗余校验码与第二循环冗余校验码进行比较。

[0028] 根据一个实施例，该方法包括以下步骤：获得以由形成存储或传递掩码的第一任何随机二进制掩码进行掩盖的形式的数据项，连同第一循环冗余校验码，使用第二随机的二进制掩码对该数据项进行第二次掩盖，该第二随机二进制掩码是生成多项式的倍数，从掩盖了两次的数据项移除第一掩码，从使用第二掩码进行掩盖的数据项来生成第二循环冗

余校验码，以及将第一循环冗余校验码与第二循环冗余校验码进行比较。

[0029] 根据一个实施例，该方法包括以下步骤：获得以使用形成存储或传递掩码的第一二进制掩码进行掩盖的形式的数据项，该第一二进制掩码是生成多项式的倍数，从使用第一掩码进行掩盖的数据项来生成第二循环冗余校验码，以及将第一循环冗余校验码与第二循环冗余校验码进行比较。

[0030] 本发明的一些实施例还涉及电子设备，该电子设备包括用于处理数据项的构件，该构件被配置为实现依照上述方法的处理步骤。

[0031] 本发明的一些实施例还涉及便携式物品，该便携式物品包括此类电子设备。

[0032] 本发明的一些实施例还涉及用于存储或传递数据项的方法，该方法包括使用随机二进制掩码对数据项进行掩盖的在先步骤，其中存储或传递以使用随机二进制掩码进行掩盖的形式的该数据项，该随机二进制掩码是用于生成循环冗余校验码的生成多项式的倍数。

[0033] 根据一个实施例，该方法包括生成随机二进制掩码的步骤，生成随机二进制掩码的步骤包括以下步骤：生成任何随机数，以及使该任何随机数与生成多项式相乘。

[0034] 根据一个实施例，该方法包括生成随机二进制掩码的步骤，生成随机二进制掩码的步骤包括以下步骤：生成第一任何随机数，以及使该第一任何随机数与生成多项式相乘以获得第一掩码部分，生成至少一个第二任何随机数，使该第二任何随机数与生成多项式相乘以获得至少一个第二掩码部分，以及将第一掩码部分与第二掩码部分连接。

[0035] 本发明的一些实施例还涉及电子设备，该电子设备包括用于存储和传递数据项的构件，该构件被配置为依照上述的存储或传递方法来存储或传递数据项。

[0036] 在关于但不限于附图的以下描述中，将更详细地描述根据本发明的数据处理方法和电路的一些实施例，其中：

[0037] -图1描述了常规数据处理方法的以上示出的步骤，

[0038] -图2示出了根据本发明的数据处理方法的第一阶段的步骤，

[0039] -图3示出了根据本发明的数据处理方法的第二阶段的步骤，

[0040] -图4示出了本方法的第二阶段的可替代实施例，

[0041] -图5示出了本方法的第二阶段的另一个可替代实施例，

[0042] -图6示出了根据本发明的数据处理方法的某些步骤的第一可替代方案，

[0043] -图7示出了根据本发明的数据处理设备，

[0044] -图8示出了根据本发明的数据处理设备的另一个示例，

[0045] -图9示出了根据本发明的码CRC计算电路的实施例的示例，以及

[0046] -图10示出了根据本发明的数据处理方法的某些步骤的第二可替代方案。

[0047] 本发明的一些实施例基于以下发现，能够从掩盖的数据项来有效地生成数据项的码CRC，假如借助于掩码(该掩盖是用于生成码CRC的生成多项式的倍数)对该数据项进行掩盖。通过将随机数与生成多项式相乘，能够从随机数生成具有这个属性的随机掩码。我们将考虑以下情况作为数字示例：

[0048] 1)使D为32比特的任何数据项，例如，$D=a120b721_h$(“h”表示以16为基数的数据项的写法)。该数据项还能够被写成为31阶的多项式：

[0049] $D=X^{31}+X^{29}+X^{24}+X^{21}+X^{15}+X^{13}+X^{12}+X^{10}+X^{9}+X^{8}+X^{5}+1$

[0050] 2)使PC为16阶的生成多项式,例如已知的多项式CRC-16-DECT:

[0051] $PC=X^{16}+X^{10}+X^{8}+X^{7}+X^{3}+1$,

[0052] 形成等于$10589_h$的17位字。

[0053] 应当注意的是,数据项D的码CRC的常规计算(通过在有限域GF(2)[X]中将数据项D除以多项式PC)给出15阶的多项式,即,这里:

[0054] $CRC=X^{15}+X^{12}+X^{8}+X^{7}+X^{5}+X^{3}+X+1$,

[0055] 也就是说,等于$91ab_h$的16位码。

[0056] 3)替代使用数据项D来计算码CRC,我们将考虑15比特的任何随机数r2,例如:

[0057] $r2=2e03_h$,即

[0058] $r2=X^{13}+X^{11}+X^{10}+X^{9}+X+1$

[0059] 4)我们将r2与生成多项式相乘,以获得32比特的掩码R2:

[0060] $R2=r2xPC=2e83509b_h$

[0061] 5)我们使用掩码R2对数据项D进行掩盖(在GF(2)[X]中的多项式加法,对应于逐位异或操作),以获得掩盖的数据项D2,即,这里:

[0062] $D2=D+R2=8fa3e7ba_h$

[0063] 6)用于计算数据项D的码CRC,我们将在GF(2)[X]中执行掩盖的数据项D2除以多项式PC的多项式除法,即以2为模的多项式除法。获得以下:

[0064] $CRC=91ab_h$

[0065] 因此,表现出的是,从掩盖的数据项D2计算的码CRC等于从未掩盖的数据项D计算的码CRC。能够示出的是,这个规则适用于任何的掩码R2(R2是PC的倍数)。

[0066] 根据优选的实施例,掩码R2具有与将被掩盖的数据项D相同的长度,以便掩盖该数据项的所有比特。在这种情况下,观察到以下规则:如果数据项D包括N比特,以及多项式PC包括P比特,则任何随机数r2包括Q比特,其中Q等于N-P比特。在以上示例中,N=32,P=17,Q=15。此外,获得的码CRC具有P-1比特,即此处16比特。

[0067] 图2描述了根据本发明的数据处理方法的一个实施例的初始阶段P2。该初始阶段包括:

[0068] -步骤1,生成随机数r2,

[0069] -步骤3,通过r2乘以PC,生成随机掩码R2,

[0070] -步骤20,获得数据项D,

[0071] -步骤21,通过将该数据项与掩码R2相加,使用掩码R2对该数据项进行掩盖,以获得掩盖的数据项D2,以及

[0072] -步骤22,从多项式PC和掩盖的数据项D2,计算数据项D的码CRC,使得计算步骤被保护尤其是免受“模板”类型的攻击,以及

[0073] -步骤23,处理该数据项。

[0074] 在图2上示出了处理步骤23的实施例的各种示例。假设,在计算了码CRC后立即执行步骤23,以及对于执行这个步骤的单元或实体可以获得掩码R2。

[0075] 示例1:处理步骤23包括步骤24,步骤24通过使用掩码R2对数据项D2进行掩盖的GF(2)[X]中的多项式加法,移除掩码R2,跟随的步骤25处置未掩盖的数据项D。优选地,由保护免受攻击的电路或电路的一部分来执行这些步骤。利用步骤25包括例如密码计算的执行,

该数据项是将被编码的消息，用于编码消息的密钥或子密钥。

[0076] 示例2：处理步骤23包括步骤24，步骤24移除掩码R2，跟随的是步骤26，步骤26使用任何随机掩码R1对未掩盖的数据项D进行掩盖(即，R1未必是PC的倍数)以获得掩盖的数据项D1。在步骤27期间，掩盖的数据项D1连同它的码CRC被存储在存储器中或被传递给除了执行上述步骤的单元或实体之外的单元或实体。

[0077] 示例3：处理步骤23包括步骤28，步骤28通过使用任何随机掩码R1对掩盖的数据项D2进行掩盖，以获得掩盖两次的数据项D3。跟随在步骤28后的步骤29通过将该掩码加到数据项D3来移除掩码R2，以及导致与上述步骤26相同的结果，即数据项D1由掩码R1进行掩盖。上述步骤27跟随在步骤29之后。

[0078] 示例4：在这个示例中，掩盖的数据项D2连同它的码CRC仅被存储在存储器中或传递给除了执行上述步骤的单元或实体之外的单元或实体(步骤30)。

[0079] 将注意的是，生成掩码R1的步骤6或获得掩码R1的步骤7在示例2的掩盖步骤26或示例3的掩盖步骤28的前面。

[0080] 还将注意的是，示例3是示例2的优选可替代方案，示例2在步骤24和步骤26之间具有使得数据项处于明语的缺点，如果在不安全的环境中移除掩码，则明语的数据项可能是不期望的。

[0081] 最后，处理步骤23可以包括刚刚已经描述的各种步骤24至30的不同组合。

[0082] 初始阶段P2可以由在图3、图4和图5上示出的随后的处理阶段P3、P4、P5跟随，在随后的处理阶段P3、P4、P5期间，码CRC用于检查掩盖的数据项D1或D2的完整性。

[0083] 在图3上的阶段P3由步骤31开始，步骤31获得掩盖的数据项D1连同它的码CRC。这个步骤出现在例如上述步骤27之后。步骤31由步骤32跟随，步骤32通过将掩码R1与掩盖的数据项D1相加来移除掩码R1，接着，通过上述步骤21，通过将R2与数据项相加，使数据项与掩码R2进行掩盖，以获得掩盖的数据项D2。步骤21由步骤33跟随，步骤33从数据项D的掩盖值D2和多项式PC来计算数据项D的码CRC'。步骤33由步骤34跟随，步骤34将码CRC'与接收的码CRC进行比较，如果码CRC与CRC'不同，则通过错误处理步骤35，否则通过步骤23来处理数据项D，以上已经描述了这些步骤的各种示例(处理、存储或传递)。

[0084] 取决于正在执行的操作的性质，错误处理步骤35可以包括各种动作，从发送简单的错误信号到执行安全动作，诸如中断正在执行的操作或者甚至破坏敏感数据或重新设置执行这个操作的电路。

[0085] 生成掩码R1的步骤6或获得掩码的R1的步骤7在去除掩盖的步骤32之前。因为已经生成了掩码R1，因此如果这种生成是可重复的，例如借助于已知的随机种子和确定性的生成函数，则步骤6是唯一可能的。否则，必须提供获取步骤7。步骤7涉及例如读取在预定的或经由命令接收的地址处的寄存器中或存储器中的掩码。

[0086] 在掩盖步骤21之前是生成掩码R2的步骤3，以及步骤3包含R2乘以PC的乘法，或掩盖步骤21之前是获得掩码R2的步骤4。在适用的情况下，步骤3之前是生成随机数r2的步骤1或获得数r2的步骤2。此处，针对在计算步骤33期间进行掩盖，以及针对在步骤1中生成的数r2与在图2中的步骤1中生成的数r2相同不是必要的。

[0087] 在图4上的阶段P4是阶段P3的优选的替代方案，阶段3在步骤32和步骤21之间具有使得数据项处于明语的缺点。阶段P4也由获得掩盖的数据项D1连同它的码CRC的步骤31开

始。步骤31由步骤36跟随，步骤36使用掩码R2对掩盖的数据项D1进行掩盖，以获得掩盖两次的数据项D3。步骤36由步骤37跟随，步骤37通过掩码R1与数据项D3相加，移除掩码R1，以获得使用掩码R2进行掩盖的数据项D2。步骤37由上述步骤34跟随，步骤34将码CRC'与接收的码CRC进行比较，如果码CRC和码CRC'不同，则通过上述错误处理步骤35，否则通过处理数据项D的上述步骤23。

[0088] 在掩盖步骤36之前是生成掩码R2的步骤3或获得掩码R2的步骤4。在适用的情况下，步骤3之前是随机地生成数r2的步骤1或获得数r2的步骤2。此外，去除掩盖的步骤37之前是生成掩码R1的步骤6或获得掩码R1的步骤7。

[0089] 在图5上的阶段P5是根据本发明的处理阶段的一种简单和有利的实施例。在这个实施例中，掩码R2不仅在计算码CRC期间使用，而且被用作用于存储或传递数据项D的存储或传递掩码。因此，阶段P5由步骤38开始，步骤38获得掩盖的数据项D2连同码CRC，步骤38发生在例如上述步骤30之后。获得步骤直接由步骤33和步骤34跟随，步骤33计算码CRC'，以及步骤34将CRC'与原始的码CRC进行比较，而不需要移除掩码R2。如上所述，步骤33、步骤34由错误处理步骤35或由处理数据项D的步骤23跟随。

[0090] 处理步骤23可以预先地要求执行生成掩码R2的步骤3或获得掩码R2的步骤4，例如如果它包含步骤24和步骤25(图2)。如上所指示的，于是，步骤3之前是步骤1或步骤2。在这种情况下，因为已经预先地生成了掩码R2，所以如果R2是已知的，则步骤3是唯一可能的。于是，步骤之前是生成数r2的步骤1或获得数r2的步骤2。因为，已经预先地生成了数r2，因此如果数r2是从已知的随机种子和已知的生成函数生成的，则步骤1是唯一可能的。否则，必须执行获得步骤2。获得步骤2涉及例如在预定的或经由命令接收的地址处读取寄存器中或存储器中的数r2。在这种情况下，可以优选地替代执行获得步骤4，获取步骤4涉及例如在预定的或经由命令接收的地址处读取存储器中或寄存器中的掩码R2。

[0091] 图6示出了用于通过将掩码的段进行连接来生成掩码R2的方法，该方法能够在我们具有使得不能生成Q比特的数r2的随机数生成器时使用，其中Q等于N-P比特。该方法包括生成第一随机数r2a的步骤1a或获得第一随机数r2a的步骤2a，以及生成第二随机数r2b的步骤1b或获得第二随机数r2b的步骤2b。步骤1a或步骤2a由步骤3a跟随，步骤3a通过r2a与多项式PC相乘来生成掩码的第一段R2a。类似地，步骤1b或步骤2b由步骤3b跟随，步骤3b通过r2b与多项式PC相乘来生成该掩码的第二段R2b。步骤3a和步骤3b由步骤5跟随，步骤5连接两个段R2a、R2b以获得掩码R2。可替代地，以及尤其在上述阶段P3或P4期间，步骤3a能够由获得段R2a的步骤4a来替换，以及步骤3b能够由获得段R2b的步骤4b来替换。

[0092] 还能够使用更大数目的段r2i来生成掩码。一般地，如果数据项包括N比特，以及生成多项式包括P比特，则段r2i中的每个段具有等于(N-nP)/n的比特数目，n表示生成的段r2i的数目。

[0093] 图7示出了电子设备DV1、DV2，其被配置为实现根据本发明的数据处理方法。设备DV1、DV2是例如半导体芯片上的集成电路，每个设备被布置在介质CD1、CD2上。设备DV1形成例如接触和/或无接触的智能卡、无接触的电子标签、SD或微SD卡或任何其它便携式电子设备。设备DV2形成例如与设备DV1兼容的智能卡阅读器、标签阅读器、SD卡阅读器或任何其它类型的终端。

[0094] 设备DV1、DV2中的每个设备包括：处理器PROC1，PROC2，存储器MEM1，MEM2，随机生

成器RGEN1，RGEN2以及有线、无线或无接触的通信接口电路ICCT1，ICCT2。设备中的每个设备装备有软件或硬件加密构件(未示出)，共享的加密密钥K以及共享的生成多项式PC，密钥K和多项式PC例如被存储在它们各自的存储器中。

[0095] 作为操作的示例，此处，考虑存储和传递顺序，在存储和传递顺序期间，设备DV1在它的存储器MEM1中存储数据项D，接着，将它传递给设备DV2，DV2将它存储在它的存储器MEM2中。在对应于上述阶段P2(图2)的初始处理阶段期间，处理器PROC1生成或接收数据项D，计算它的码CRC，接着将它以它的掩盖的形式D1连同码CRC存储在存储器MEM1中(图2，步骤27)。用于这个目的的掩码R1由生成器RGEN1提供，以及被保持在处理器的寄存器中。随后，处理器PROC1通过电路ICCT1、ICCT2与设备DV2建立数据链路。接着，处理器PROC1读取掩盖的数据项D1以及码CRC，以及执行阶段P4(参照图4)以检查码CRC。用于这个目的的掩码R2由处理器PROC1从由生成器RGEN1提供的随机数r2来计算，或直接由生成器RGEN1提供。当检查码CRC时，处理器PROC1执行保护免受攻击的处理顺序，该处理顺序包括移除掩码R1，以及借助于密钥K将数据项D转换到编码的数据项$D4_{(k)}$。接着，处理器向设备DV2发送数据项$D4_{(k)}$以及码CRC。处理器PROC2借助于它自己的密钥K解码数据项D，将使用从由生成器RGEN2提供的随机数r2'计算的或由生成器RGEN2直接提供的掩码R2'对它进行掩盖，以获得数据项D2'，检测它的码CRC(图3，步骤33和步骤34)，将任何掩码R1'应用于它以获得数据项D3'(图2，步骤28)，将掩码R1'存储在寄存器中，移除掩码R2'(图2，步骤29)以获得数据项D1'，接着将数据项D1'连同码CRC存储在它的存储器MEM2中(图2，步骤27)。

[0096] 在一个可替代的实施例中，掩码R2、R2'都用作存储掩码以及用作用于计算码CRC的掩码。接着，替代数据项D1、D1'，数据项D2、D2'随同码CRC被存储在存储器MEM1、MEM2中。

[0097] 在不涉及任何加密方法来传递数据项D的另一种可替代方案中，掩码R2在耦合设备DV1、DV2的阶段期间被确定，以及被保持在它们各自的存储器中。该数据项以它的掩盖的形式D2被传送给设备DV2，因此此处掩码R2还用作传递掩码。

[0098] 图8示出了电子设备DV3的另一个示例，电子设备DV3被配置为实现根据本发明的数据处理方法。设备DV3是例如上述类型的集成电路，被安装便携式介质、信号解码集成电路(例如TV解码器)、支付终端集成电路等上。设备DV3包括：上述类型的处理器PROC以及通信接口电路ICCT，ICCT直接耦合到处理器PROC。它还包括：具有密钥K的加密协处理器CPROC、存储器MEM、随机生成器RGEN以及专用于码CRC计算的电路CRCCT。这些单元通过数据总线DB耦合到处理器。出于简洁的原因，没有示出这些单元之间的其它链路，尤其是地址总线或控制总线。还提供了一组特定的链路SLi，以在不通过数据总线DB的情况下来交换掩码或随机变量。这些特定的链路SLi特别地使得生成器RGEN能够：

[0099] -向协处理器CPROC提供用于实现对策(例如对抗DPA类型的攻击)的随机掩码Mi

[0100] -向电路CRCCT、处理器PROC以及协处理器CPROC提供任何随机掩码R1以存储或传递数据，

[0101] -向电路CRCCT提供任何随机数r2，此处，CRCCT被配置为从数r2提供随机掩码R2，R2是多项式PC的倍数。

[0102] 作为示例，考虑数据处理顺序，在数据处理顺序期间，处理器PROC必须对经由接口电路ICCT接收的消息M进行解码。编码的消息M具有类型$F_K[D,CRC]$，以及含有数据项D以及它的码CRC，已经借助于加密函数F和密钥K将它们连接和编码在一起。

[0103] 处理器将消息M应用于数据总线DB，以及要求协处理器CPROC来解码它。CPROC将消息M加载到内部寄存器(未示出)中，要求生成器RGEN向它提供一个或多个对策掩码Mi，以及存储和传递掩码R1。掩码R1还被提供给电路CRCCT以及协处理器CPROC，它们中的每一个将掩码R1存储在内部寄存器(未示出)中。在保护免受攻击的顺序期间，协处理器借助于密钥K来解码该消息，以获得数据项D和它的码CRC，将使用掩码R1对数据项D进行掩盖，以获得掩盖的数据项D1，将数据项D1应用于数据总线，要求生成器RGEN向电路CRCCT提供随机数r2，以及要求电路CRCCT计算它的码CRC。电路CRCCT生成掩码R2，在总线上读取数据项D1，使用R2对该数据项进行掩盖，以获得掩盖的数据项D3(图4，步骤36)，移除掩码R1以获得掩盖的数据项D2(图4，步骤37)，计算码CRC'以及将它应用于总线DB。协处理器将与数据项一起接收的码CRC与由电路CRCCT计算的码CRC'进行比较，以及如果这两个码是相同的，则向处理器PROC指示数据项已经被正确地解码。处理器从数据总线取走数据项D1和码CRC，以及执行一个或多个处理步骤(参照图2)。

[0104] 在一种可替代方案中，替代掩码R1，掩码R2用作存储和传递掩码。在这种情况下，掩码R2不再由电路CRCCT生成，而是由生成器RGEN来提供，以替代掩码R1。接着，生成器RGEN装备有用于接收多项式PC的寄存器和乘法器。由电路CRCCT执行的双重掩盖、移除掩码R1和移除掩码R2的步骤不再需要。

[0105] 图9表示使得能够实现刚刚已经描述的数据处理方法的示例的电路CRCCT的一个实施例。电路CRCCT包括：输入寄存器REG1、REG2、REG3，缓冲寄存器REG4、REG5，输出寄存器REGS，乘法器MLT、多项式加法器AD1(例如，具有两个输入的一系列的异或门)，循环冗余校验码生成电路CRCORE以及复用单元MUX。这些各种组件由控制单元CTU进行控制，CTU接收由上述处理器或协处理器发送的命令CMD。

[0106] 寄存器REG1包括连接到数据总线DB的输入，以及用于顺序地接收多项式PC以及数据项(该数据项的码CRC必须被计算)，例如数据项D，掩盖的数据项D1或掩盖的数据项D2。寄存器REG1的输出链接到寄存器REG4的输入，寄存器REG4的输入用于接收多项式PC，以及链接到复用器MUX的输入E1。寄存器REG4的输出链接到乘法器MLT的第一输入，以及链接到电路CRCORE的输入E1。寄存器REG2具有链接到特定链路SLi的输入，以及用于接收任何掩码R1。寄存器REG2包括链接到复用器的输入E2的输出以向加法器传递掩码R1。寄存器REG3具有链接到特定链路SLi的输入，以及用于接收随机数r2。寄存器REG3包括链接到乘法器MLT的第二输入的输出，乘法器包括链接到复用器MUX的输入E3的输出。寄存器REG 5具有链接到加法器的输出的输入，以及链接到复用器MUX的输入E4的输出。复用器包括连接到加法器的输入的两个输出S2、S3，以及链接到电路CRCORE的输入E2的输出S1，该输出S1还链接到加法器的输出。CRCORE向寄存器REGS提供码CRC，REGS的输出链接到总线DB。单元CTU控制复用器MUX中的数据路径以将输入E1至E4连接到输出S1至S3，以及执行操作，以下示意性描述关于该操作的示例。这些操作之前是以下步骤：将生成多项式加载到寄存器REG1，接着将该多项式传递到寄存器REG4中。因此，多项式PC被应用于乘法器MLT的第一输入，以及电路CRCORE的输入E1。

[0107] 1)计算CRC(PC,D1)：

[0108] 1a)载入数据：将D1载入到REG1，将R1载入到REG2，将r2载入到REG3，将PC和r2应用于乘法器MLT，MLT提供R2。

[0109] 1b)计算数据项D3:通过复用器的输出S2,将R2应用于加法器AD1的输入E1。图片没干过复用器的输出S3将D1应用于加法器的输入E2,以及将输出S1置为高阻抗。加法器提供D3。

[0110] 1c)移除掩码R1:将D3载入REG5,通过复用器的输出S3,将D3应用于加法器的输入E2,通过复用器的输出S2,将R1应用于加法器的输入E1,以及将输出S1置为高阻抗。加法器提供数据项D2。

[0111] 1d)计算码CRC:激活电路CRCORE,电路CRCORE接收PC和D2。电路CRCORE向REGS的输入提供码CRC。通过REGS,将码CRC传递到总线DB上。

[0112] 2)计算CRC(PC,D2)

[0113] 2a)将D2载入到REG1中,通过复用器的输出S1,将D2应用于电路CRCORE的输入E2,以及将加法器的输出置为高阻抗。

[0114] 2b)计算码CRC,如上所述。

[0115] 3)计算CRC(PC,D):

[0116] 3a)将D载入到REG1中,将r2载入到REG3中,将PC和r2应用于乘法器MLT,MLT提供R2。

[0117] 3b)通过复用器的输出S2,将R2应用于加法器AD1的输入E1。通过输出S3将D应用于加法器的输入E2。加法器提供D2。

[0118] 3c)计算码CRC,如上所述。

[0119] 示例3涉及未掩盖的数据项D的第一码CRC的计算。然而,可能期望的是从不将未掩盖的数据项D应用于数据总线。出于这个目的,处理器能够被配置为从由生成器RGEN提供的掩码R1或从数r2,自己生成掩盖的数据项D1或D2,如上所述,以及向电路CRCCT提供掩盖的数据项D1或D2。然而,能够在具有不是非常敏感的数据的某些应用中,或在安全地点中执行对设备进行定制的步骤期间来执行这个操作。

[0120] 本领域的技术人员将理解的是,电路CRCCT容许各种其他可替代实施例。用于根据本发明来计算码CRC的方法也容许各种其他可替代方案。特别地,能够以从数据项D的F段连续运行的方式来计算码CRC,例如当电路CRCCT的输入寄存器REG1具有不充足的大小以接收数据项D的N个比特时。图10示出了用于以从数据项的掩盖的段连续运行的方式来计算码CRC的方法的示例。

[0121] 该方法包括:生成第一随机数r2ab的步骤1ab或获得第一随机数r2ab的步骤2ab,生成第二随机数r2bc的步骤1bc或获得第二随机数r2bc的步骤2bc,生成第三随机数r2cd的步骤1cd或获得第三随机数r2cd的步骤2cd。步骤1ab或2ab由通过r2ab与PC相乘来生成第一段的掩码R2ab的步骤3ab跟随。步骤1bc或2bc由通过r2bc与PC相乘来生成第二段的掩码R2bc的步骤3bc跟随。步骤1cd或2cd由通过r2cd与PC相乘来生成第三段的掩码R2cd的步骤3cd跟随。可替代地,以及尤其是在上述阶段P3或P4期间,步骤3ab能够用获得掩码的段R2ab的步骤4ab来替换,步骤3bc能够用获得掩码的段R2bc的步骤4bc来替换,以及步骤3cd能够用获得掩码的段R2cd的步骤4cd来替换。

[0122] 接着,该方法包括将数据项分成三段Dab、Dc和Dd的步骤40。段Dab,或最重要的段,具有两倍于两个其它段Dc、Dd的长度的长度(即比特的数目)。例如,如果数据项是128比特,则段Dab是64比特,以及两个其它的段是32比特。在步骤41期间,借助于掩码的段R2ab对段

Dab进行掩盖，以获得掩盖的数据项段D2ab。在步骤42期间，从掩盖的数据项的段D2ab和从多项式PC来计算中间的循环冗余校验码CRCab。在步骤43期间，中间的码CRCab与数据项的段Dc进行连接，作为最重要的数据项的段，以形成数据项的段Dbc＝CRCab|Dc(“|”表示连接操作)。在步骤44，借助于掩码的段R2bc对段Dbc进行掩盖，以获得掩盖的数据项的段D2bc。在步骤45期间，从段D2bc和从多项式PC来计算中间的循环冗余校验码CRCbc。在步骤46期间，对中间的码CRCbc与数据项的段Dd进行连接，作为最重要的数据项的段，以形成数据项的段Dcd＝CRCbc|Dd。在步骤47期间，借助于掩码的段R2cd对段Dcd进行掩盖，以获得掩盖的中间的数据项的段D2cd。在步骤48期间，从掩盖的中间的数据项的段D2cd和从多项式PC来计算循环冗余校验码CRCcd。这个码既是中间的段Dcd的码CRC也是数据项D的码CRC。

[0123] 取决于数据项的长度以及用于执行上述步骤的寄存器的大小，数据项的段的数目F可以不同于3，例如F＝2或F>3。在这种情况下，计算的中间的码CRC的数量等于F-1。

[0124] 通过增加在虚线中示出的元件，即中间寄存器REG6(它的输入链接到电路CRCORE的输出)，连接寄存器CREG(具有连接到寄存器REG6的输出的输入，以及连接到寄存器REG1的输出的输入，以及连接到复用器的输入E5的输出)，可以由图9中的电路CRCCT来实现这个方法。在这种情况下，寄存器REG1首先接收段Dab，电路CRCORE提供码CRCab，码CRCab被传递到REG6中。接着，寄存器REG1接收段Dc，在寄存器CREG中，段Dc与CRCab连接，以及电路CRCORE提供码CRCbc，CRCbc被传递到REG6中。接着，寄存器REG1接收段Dd，在CREG中，段Dd与CRCbc连接，接着，电路CRCORE提供搜索的码CRC。

[0125] 根据一个实施例，寄存器REG1包括用于存储段Dab、Dc和Dd的若干位置。根据一个实施例，随机数r2ab、r2bc和r2cd是相同的。在另一个实施例中，数r2ab、r2bc和r2cd被提供给电路CRCCT作为计算进展。在又一个实施例中，在图9上示出的，寄存器REG3包括用于同时存储r2ab、r2bc和r2cd，或甚至其它数，一直到r2n的若干位置。在这种情况下，当计算无分段的码CRC时，能够通过单元CTU随机地选择若干数r2。在又一个实施例中，随机生成器被集成到电路CRCCT中。

[0126] 本领域的技术人员将理解的是，根据本发明的数据处理方法容许各种其它可替代实施例。根据本发明的数据处理方法能够特别是借助于不同于CRC-16-DECT的各种生成多项式来实现，例如以下已知的生成多项式：CRC-1,CRC-4-ITU,CRC-5-EPC,CRC-5-ITU,CRC-5-USB,CRC-6-ITU,CRC-7,CRC-8-CCITT,CRC-8-Dallas/Maxim,CRC-8,CRC-8-SAEJ1850,CRC-8-WCDMA,CRC-10,CRC-11,CRC-12,CRC-15-CAN,CRC-16-IBM,CRC-16-CCITT,CRC-16-T10-DIF,CRC-16-DNP,CRC-16-ARINC,CRC-16-Fletcher,CRC-24,CRC-24-Radix-64CRC-30,CRC-32-Adler,CRC-32,CRC-32C(卡斯塔尼奥利),CRC-32K(考夫曼),CRC-32Q,CRC-40-GSM,CRC-64-ISO,CRC-64-ECMA-182。

[0127] 最后，将注意的是，在本说明书中以及在权利要求书中，词语“随机”可以意味着“伪随机”。此外，词语“随机”可以仅意味着“那是攻击者不知道的”，以及可以从已知的种子和从确定性的函数来生成掩码R1或R2，如上所述。

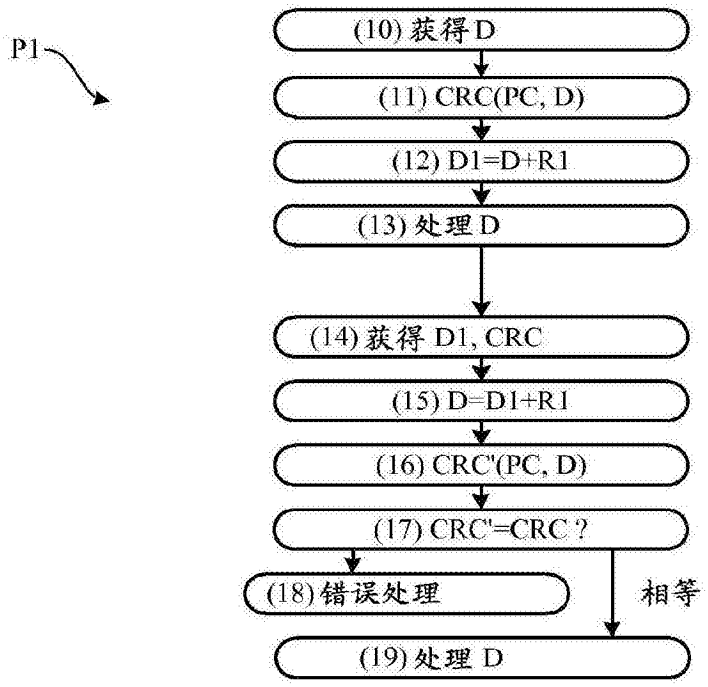

P1
(10) 获得 D
(11) CRC(PC, D)
(12) D1=D+R1
(13) 处理 D
(14) 获得 D1, CRC
(15) D=D1+R1
(16) CRC'(PC, D)
(17) CRC'=CRC ?
(18) 错误处理
相等
(19) 处理 D

图1 (现有技术)

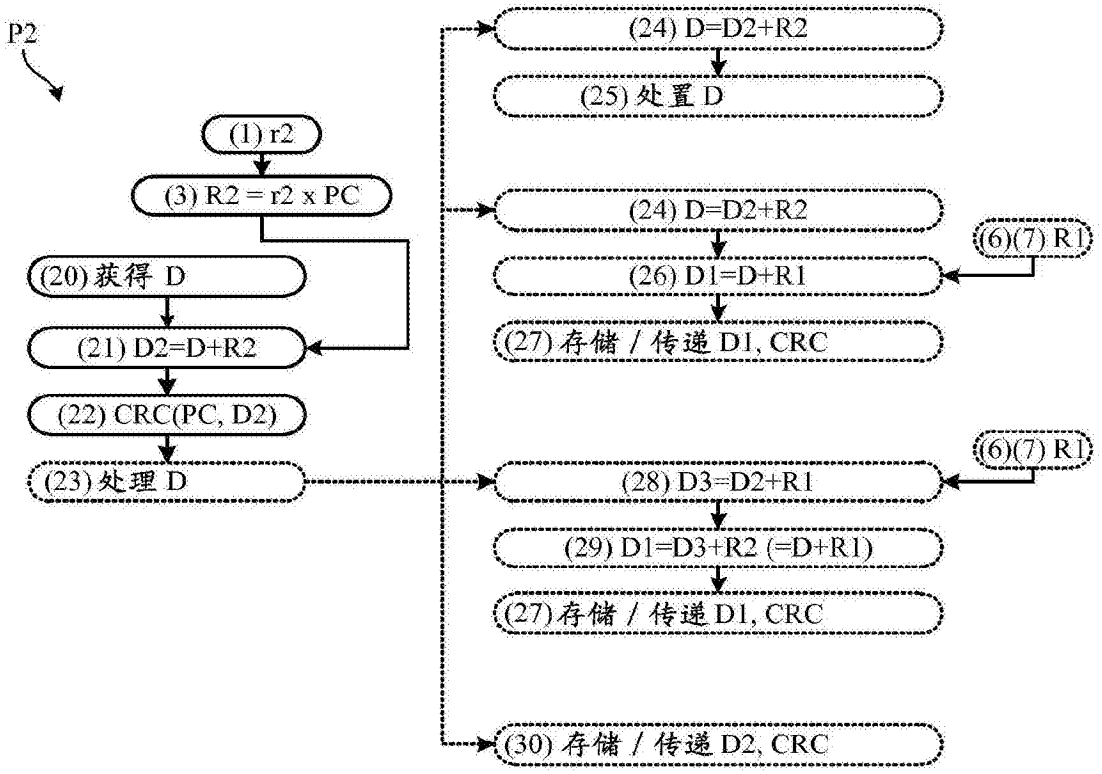

P2
(1) r2
(3) R2 = r2 x PC
(20) 获得 D
(21) D2=D+R2
(22) CRC(PC, D2)
(23) 处理 D
(24) D=D2+R2
(25) 处置 D
(24) D=D2+R2
(6)(7) R1
(26) D1=D+R1
(27) 存储 / 传递 D1, CRC
(6)(7) R1
(28) D3=D2+R1
(29) D1=D3+R2 (=D+R1)
(27) 存储 / 传递 D1, CRC
(30) 存储 / 传递 D2, CRC

图2

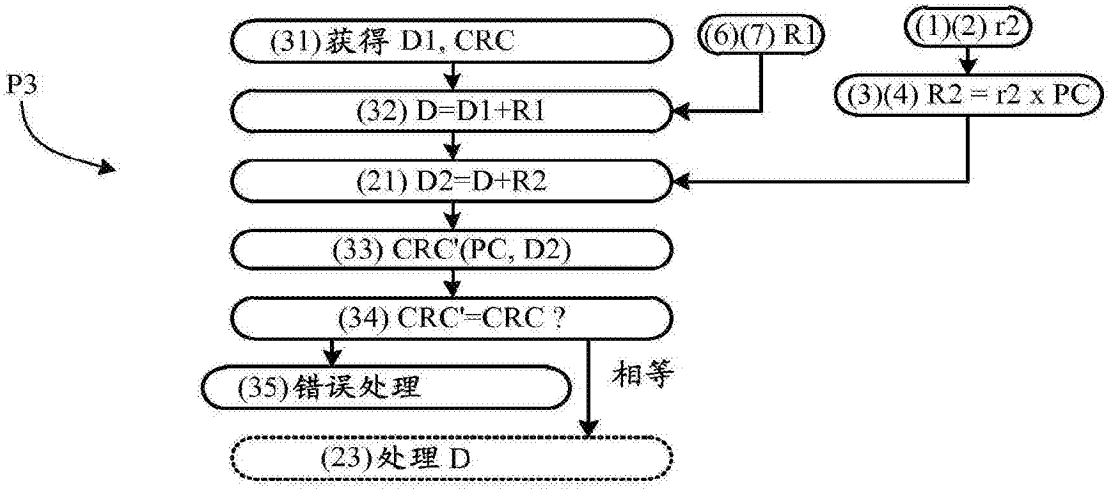

P3
(31)获得 D1, CRC
(6)(7) R1
(1)(2) r2
(32) D=D1+R1
(3)(4) R2 = r2 x PC
(21) D2=D+R2
(33) CRC'(PC, D2)
(34) CRC'=CRC ?
相等
(35)错误处理
(23)处理 D

图3

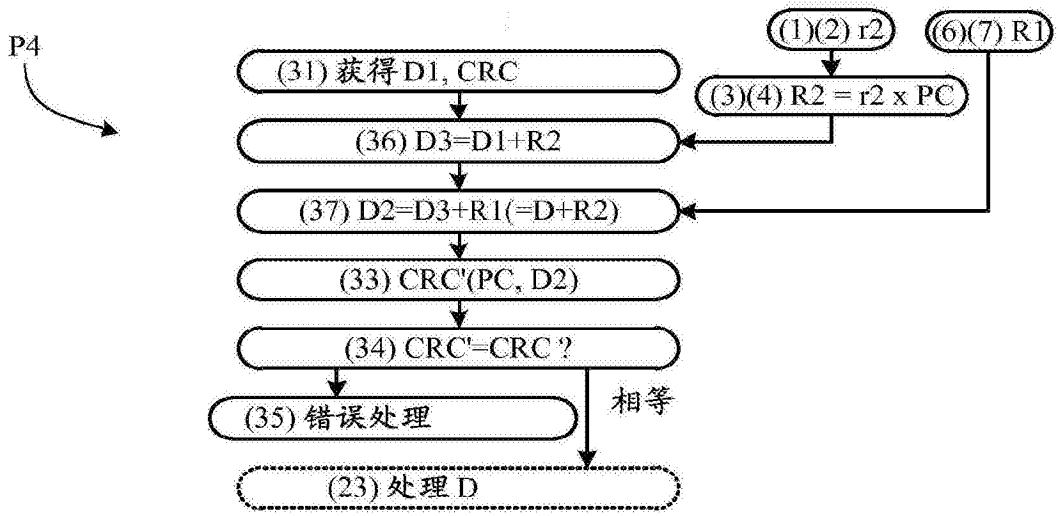

P4
(1)(2) r2
(6)(7) R1
(31) 获得 D1, CRC
(3)(4) R2 = r2 x PC
(36) D3=D1+R2
(37) D2=D3+R1(=D+R2)
(33) CRC'(PC, D2)
(34) CRC'=CRC ?
相等
(35) 错误处理
(23) 处理 D

图4

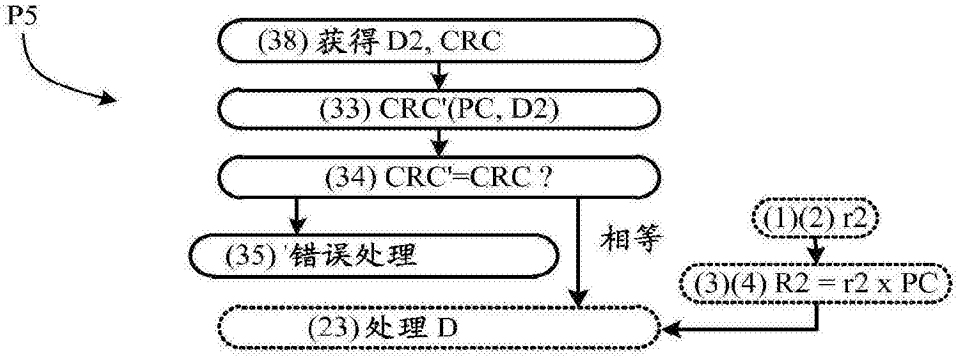

P5
(38) 获得 D2, CRC
(33) CRC'(PC, D2)
(34) CRC'=CRC ?
相等
(35) '错误处理
(1)(2) r2
(3)(4) R2 = r2 x PC
(23)处理 D

图5

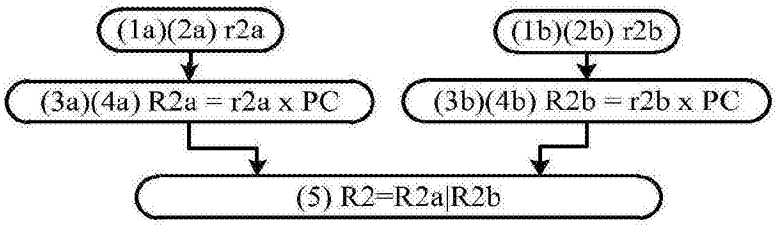

(1a)(2a) r2a
(1b)(2b) r2b
(3a)(4a) R2a = r2a x PC
(3b)(4b) R2b = r2b x PC
(5) R2=R2a|R2b

图6

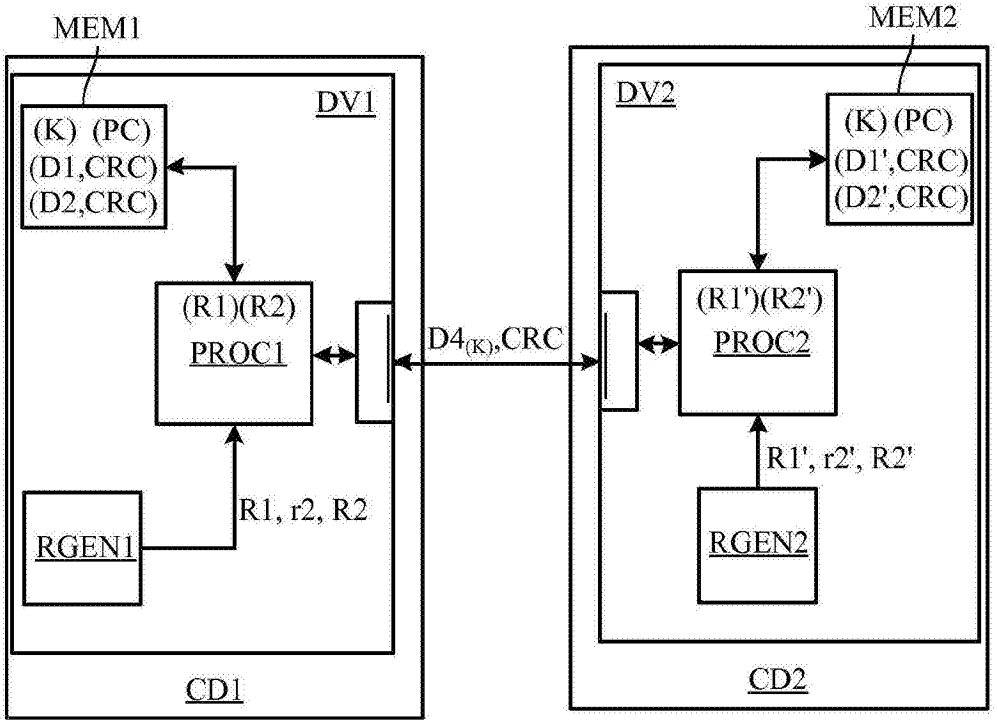

MEM1
MEM2
DV1
DV2
(K) (PC)
(D1,CRC)
(D2,CRC)
(K) (PC)
(D1',CRC)
(D2',CRC)
(R1)(R2)
PROC1
D4(K),CRC
(R1')(R2')
PROC2
R1, r2, R2
R1', r2', R2'
RGEN1
RGEN2
CD1
CD2

图7

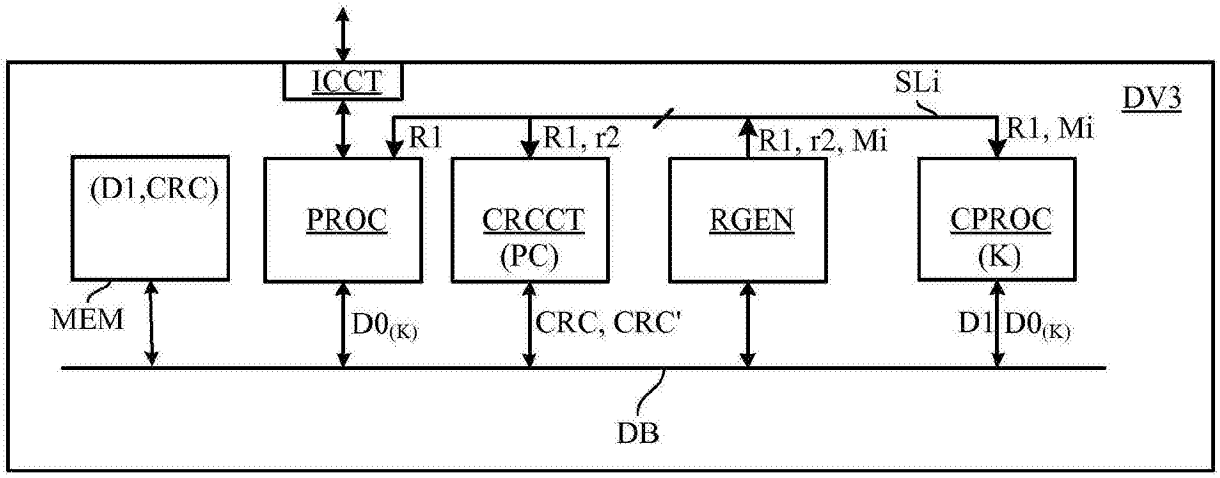

ICCT
SLi
DV3
R1
R1, r2
R1, r2, Mi
R1, Mi
(D1,CRC)
PROC
CRCCT
(PC)
RGEN
CPROC
(K)
MEM
D0(K)
CRC, CRC'
D1
D0(K)
DB

图8

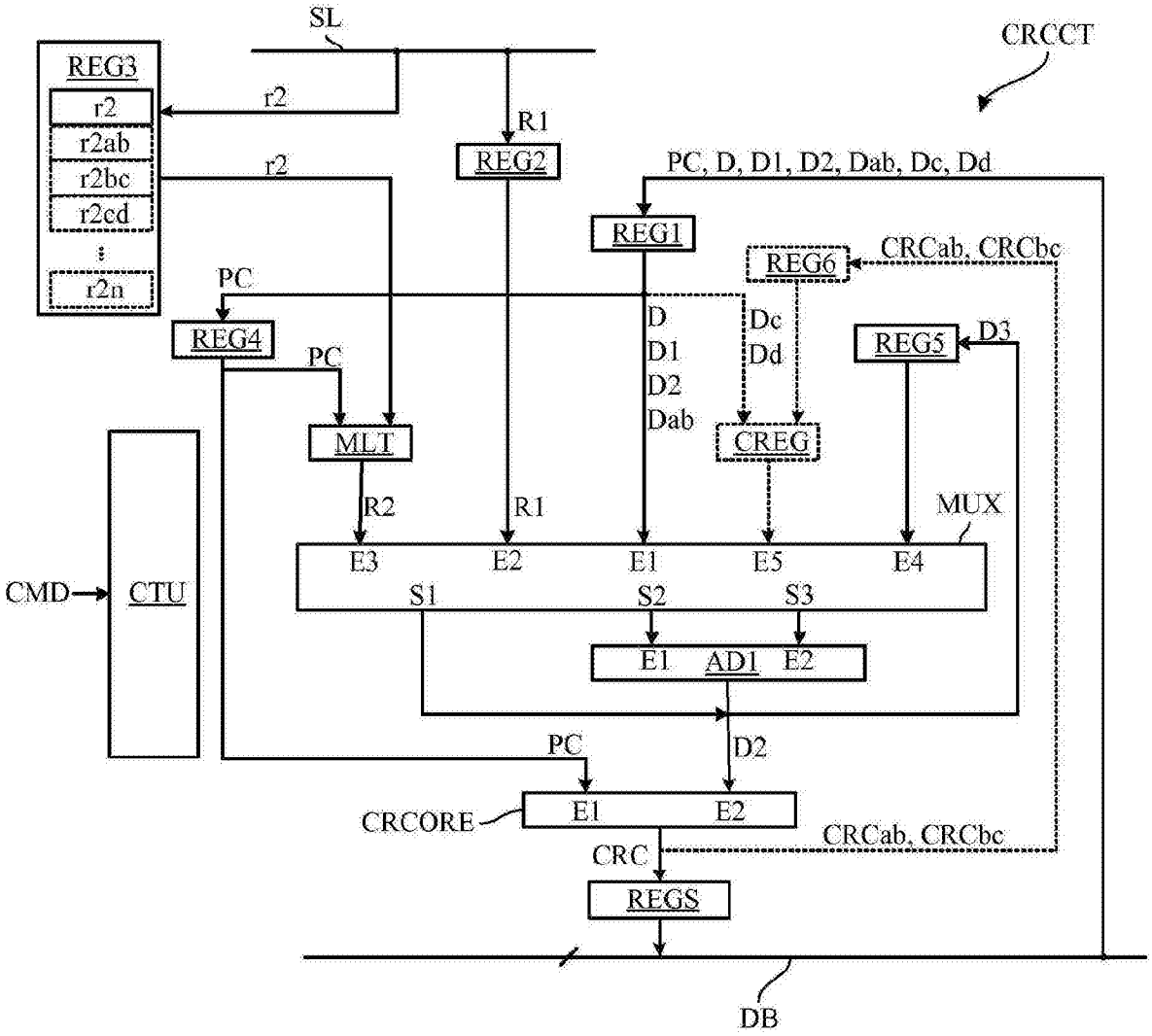

SL
CRCCT
REG3
r2
r2ab
r2bc
r2cd
r2n
r2
R1
REG2
r2
PC, D, D1, D2, Dab, Dc, Dd
REG1
REG6
CRCab, CRCbc
PC
D
D1
D2
Dab
Dc
Dd
REG5
D3
REG4
PC
MLT
CREG
R2
R1
MUX
E3
E2
E1
E5
E4
CMD
CTU
S1
S2
S3
E1
AD1
E2
PC
D2
CRCORE
E1
E2
CRCab, CRCbc
CRC
REGS
DB

图9

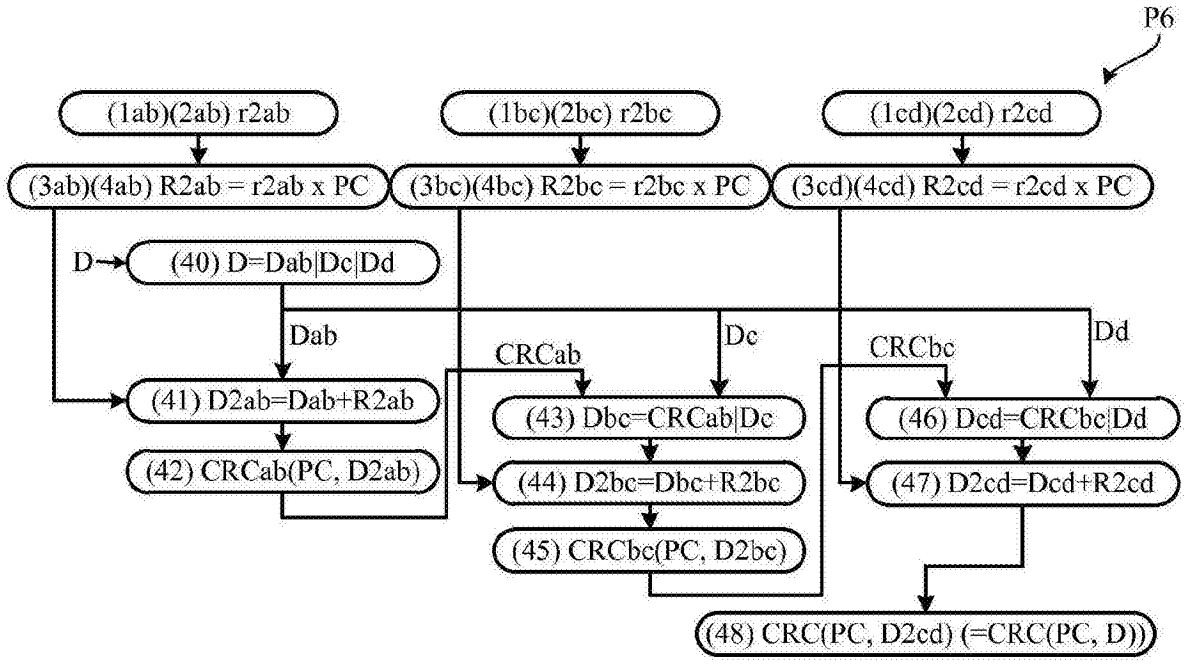

P6
(1ab)(2ab) r2ab
(1bc)(2bc) r2bc
(1cd)(2cd) r2cd
(3ab)(4ab) R2ab = r2ab x PC
(3bc)(4bc) R2bc = r2bc x PC
(3cd)(4cd) R2cd = r2cd x PC
D
(40) D=Dab|Dc|Dd
Dab
CRCab
Dc
CRCbc
Dd
(41) D2ab=Dab+R2ab
(43) Dbc=CRCab|Dc
(46) Dcd=CRCbc|Dd
(42) CRCab(PC, D2ab)
(44) D2bc=Dbc+R2bc
(47) D2cd=Dcd+R2cd
(45) CRCbc(PC, D2bc)
(48) CRC(PC, D2cd) (=CRC(PC, D))

图10