[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
H03M 13/09 (2006.01)
H04L 1/00 (2006.01)
G06F 11/10 (2006.01)

[12] 发明专利说明书

专利号 ZL 200480007468.8

[45] 授权公告日 2009 年 2 月 11 日

[11] 授权公告号 CN 100461637C

[22] 申请日 2004.2.27
[21] 申请号 200480007468.8
[30] 优先权
[32] 2003.3.28 [33] US [31] 10/249,290
[86] 国际申请 PCT/EP2004/003036 2004.2.27
[87] 国际公布 WO2004/086633 英 2004.10.7
[85] 进入国家阶段日期 2005.9.19
[73] 专利权人 国际商业机器公司
地址 美国纽约
[72] 发明人 M-I·M·林 D·R·施陶费尔
[56] 参考文献
US4703485A 1987.10.27
WO02082664A2 2002.10.17
CN1324517A 2001.11.28
US2002174399A1 2002.11.21

Parallel checksum calculation accelerator. IBM technical disclosure bulletin, Vol. 36 No. 12. 1993

审查员 邓隽
[74] 专利代理机构 北京市中咨律师事务所
代理人 于静 李峥

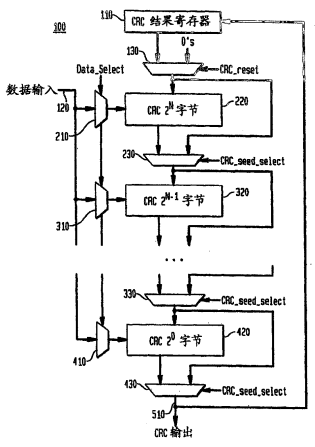
权利要求书 5 页 说明书 14 页 附图 5 页

[54] 发明名称

用于可变宽度并行循环冗余校验计算的迭代电路和方法

[57] 摘要

本申请公开了一种级联的迭代 CRC 计算电路，借此 CRC 计算可被细分到具有可选择的总线宽度的块。其优势是为任意数量字节的总线宽度即为可变的总线宽度提供并行 CRC 计算。一个实施例是一种用于生成与宽度最大为 w 字节的数据相关联的 CRC 代码字的系统，其包括第一多个串行连接的代码生成块，每个块都被构造成接收字节宽度从 $2^N + M$ 到 $2^{N-L} + M$ 的数据输入，其中 $N = \log_2(w)$，M 是偏移值，且 L 是基于最大传播延时标准的整数；第二多个并行连接的代码生成块，每个块被构造成接收字节宽度从 $2^{N-L} - 1 + M$ 到 2^0 的数据输入；以及用于选择将被包含在 CRC 计算中的、第一和第二多个块中的特定 CRC 代码生成块的装置。



1. 一种用于生成附加在这样的数据包上的 CRC 代码字的装置，该数据包具有给定的字节宽度并将在具有 2^n 字节的并行数据宽度的通信线路上传输，该装置包括多个代码生成块，每个代码生成块具有对数地选择的数据路径宽度，并级联在一起以便为最高达 2^n 字节数据的可选择并行数据宽度提供 CRC 处理，所述多个代码生成块的第一代码生成块生成针对 2^n 字节的代码，且所述多个代码生成块的第二代码生成块生成针对 2^{n-1} 字节的代码，其中来自所述第一代码生成块的所述代码被选择性地连接到所述第二代码生成块，并且对于给定的数据包，根据所述给定数据包的所述给定字节宽度，选择性地启用所述多个代码生成块中的所述第一代码生成块、第二代码生成块和其余代码生成块中的一个或多个，以便生成所述 CRC 代码字。

2. 一种用于生成附加在这样的数据包上的 CRC 代码字的装置，该数据包具有给定的字节宽度并将在具有 2^n 字节的并行数据宽度的通信线路上传输，该装置包括：

第一多个串行连接的代码生成块，每个代码生成块具有对数地选择的数据路径宽度以便为最高达 2^n 字节数据的可选择并行数据宽度提供 CRC 处理，所述第一多个串行连接的代码生成块的第一代码生成块生成针对 2^n 字节的所述代码，且所述第一多个串行连接的代码生成块的第二代码生成块生成针对 2^{n-1} 字节的所述代码，其中来自所述第一代码生成块的所述代码被选择性地连接到所述第二代码生成块；以及

第二多个并行连接的代码生成块，所述第二多个并行连接的代码生成块的第一代码生成块生成针对 2^x 字节的所述代码，且所述第二多个并行连接的代码生成块的第二代码生成块生成针对 2^y 字节的所述代码，x 和 y 与 n 和 n－1 不同；

其中，对于给定的数据包，根据所述给定数据包的所述给定字节宽度，选择性地启用所述第一多个串行连接的代码生成块中的一个或多个、而不启用所述第二多个并行连接的代码生成块或启用其中之一。

3. 一种用于生成与将在通信信道上传输的数据字节相关联的 CRC 代码字的电路，所述通信信道能够传输宽度最大为 w 个字节的数据，所述电路包括：

第一多个串行连接的代码生成块，每个该块用于根据所述块的数据输入生成 CRC 值，所述第一多个串行连接的代码生成块的各个块被构造成接收相应字节宽度在 $2^N + M$，$2^{N-1} + M$，...，$2^{N-L} + M$ 的范围内的数据输入，其中 $w = 2^N + M$，M 是偏移值，且 L 是基于在所述第一多个串行连接的代码生成块中计算 CRC 值的最大传播延时标准的整数；

第二多个并行连接的代码生成块，每个该块用于根据所述块的数据输入生成 CRC 值，所述第二多个代码生成块的各个块被构造成接收相应字节宽度在 $2^{N-L} - 1 + M$，$2^{N-L} - 2 + M$，...，2^0 的范围内的数据输入；以及

多路复用器装置，该装置可控制用于根据所述数据输入来选择将包含在 CRC 计算中的、所述第一多个串行连接的代码生成块和第二多个并行连接的代码生成块中的特定 CRC 代码生成块；

其中，通过选择性地包含或绕过 CRC 代码生成块，可处理可被 2^{N-L} 除尽的任意数量的数据输入字节，以用于相应的 CRC 代码生成。

4. 根据权利要求 3 的 CRC 代码生成电路，其中，$M \geq 0$。

5. 根据权利要求 3 的 CRC 代码生成电路，其中，N 等于 $\log_2(w)$。

6. 一种用于生成与将在通信信道上传输的数据字节相关联的 CRC 代码字的电路，所述通信信道能够传输宽度最大为 w 个字

节的数据，所述电路包括：

第一多个串行连接的代码生成块，每个该块用于根据所述块的数据输入生成 CRC 值，所述第一多个串行连接的代码生成块的各个块被构造成接收相应字节宽度在 2^N 到 2^{N-L} 的范围内的为 2 的幂的数据输入，其中，N 等于 $\log_2(w)$，且 L 是基于在所述第一多个串行连接的代码生成块中计算 CRC 值的最大传播延时标准的整数；

第二多个并行连接的代码生成块，每个该块用于根据所述块的数据输入生成 CRC 值，所述第二多个并行连接的代码生成块的各个块被构造成接收相应字节宽度在 $2^{N-L}-1$ 到 2^0 的范围内的数据输入；以及

多路复用器装置，该装置可控制用于根据所述数据输入来选择将包含在 CRC 计算中的、所述第一多个串行连接的代码生成块和第二多个并行连接的代码生成块中的特定 CRC 代码生成块；

其中，通过选择性地包含或绕过 CRC 代码生成块，可处理可被 2^{N-L} 除尽的任意数量的数据输入字节，以用于相应的 CRC 代码生成。

7. 根据权利要求 6 的 CRC 代码生成电路，其中，所述第一多个串行连接的代码生成块包括的数量为 L + 1，所述第二多个并行连接的代码生成块包括的数量为 $2^{N-L}-1$。

8. 根据权利要求 6 的 CRC 代码生成电路，其中，在所述第一多个串行连接的代码生成块内计算 CRC 值的所述第一最大传播延时标准包括时钟周期的最大延时 d_{max}，其中所述整数 $L = d_{max} - 1$。

9. 一种用于生成与将在通信信道上传输的数据字节相关联的 CRC 代码字的电路，所述通信信道能够传输宽度最大为 w 个字节的数据，所述电路包括：

第一多个串行连接的代码生成块，每个该块用于根据所述块

的数据输入生成 CRC 值，所述第一多个串行连接的代码生成块的各个块被构造成接收字节宽度为 S 个字节的数据输入；

第二多个并行连接的代码生成块，每个该块用于根据所述块的数据输入生成 CRC 值，所述第二多个并行连接的代码生成块的各个块被构造成接收相应字节宽度在 y 字节到 1 字节的范围内的数据输入；以及

多路复用器装置，该装置可控制用于根据所述数据输入来选择将包含在 CRC 计算中的、所述第一多个串行连接的代码生成块和第二多个并行连接的代码生成块中的特定 CRC 代码生成块；

其中，在所述第一多个串行连接的代码生成块内计算特定 CRC 代码值的时间与在所述第二多个并行连接的代码生成块内计算特定 CRC 代码值的时间之和小于或等于一个时钟周期。

10. 根据权利要求 9 的 CRC 代码生成电路，其中，$y = S - 1$，所述第二多个并行连接的代码生成块的所述字节宽度为连续的 S－1，S-2，...，1 个字节。

11. 根据权利要求 9 的 CRC 代码生成电路，其中，$y>S$。

12. 根据权利要求 9 的 CRC 代码生成电路，其中，所述第一多个串行连接的代码生成块包括 k 个块，每个块能够处理 S 个字节，其中（k + 1）*S>w，且 $k*S \leq w$。

13. 一种用于生成与将在通信信道上传输的数据字节相关联的 CRC 代码字的电路，所述通信信道能够传输宽度最大为 w 个字节的数据，所述电路包括：

第一多个串行连接的代码生成块，每个该块用于根据所述块的数据输入生成 CRC 值，所述第一多个串行连接的代码生成块的各个块被构造成接收字节宽度分别为 F_x，F_{x-1}，...，F_1 的数据输入，其中 $F_x \geq F_{x-1} \geq \ldots \geq F_1 \geq 0$ 字节；

第二多个并行连接的代码生成块，每个该块用于根据所述块的数据输入生成 CRC 值，该第二多个并行连接的代码生成块的各

个块被构造成接收相应字节宽度为 G_{y-i} 的数据输入，其中对于 $i=0$ 到 $y-1$，$G_{y-i}=G_{y-i-1}+1$，而 $y \geq F_1-1$；以及

多路复用器装置，该装置可控制用于根据所述数据输入来选择将包含在 CRC 计算中的、所述第一多个串行连接的代码生成块和第二多个并行连接的代码生成块中的特定 CRC 代码生成块；

其中，通过根据以下关系：$A_x*F_x+A_{x-1}*F_{x-1}+\ldots+A_1*F_1+B_y*G_y+\ldots+B_1*G_1$，其中 A_x，A_{x-1}，…，A_0，B_y，…，$B_1=0$ 或 1，选择性地包含或绕过 CRC 代码生成块，可处理任意数量的数据输入字节，以用于相应的 CRC 代码生成。

14. 根据权利要求 13 的 CRC 代码生成电路，其中，所述 $F_x=2^N$，$F_{x-1}=2^{N-1}$，…，$F_1=2^0$；且 $y=F_1-1=0$。

15. 根据权利要求 13 的 CRC 代码生成电路，其中，所述 $F_x=2^N$，$F_{x-1}=2^{N-1}$，…，$F_1=2^{N-L}$，$G_y \geq 2^{N-L}-1$，…$G_1=1$，且 L 是基于在所述第一多个串行连接的代码生成块中计算 CRC 值的最大传播延时标准的整数。

16. 根据权利要求 13 的 CRC 代码生成电路，其中，所述 $F_x=2^N+M$，$F_{x-1}=2^{N-1}+M$，…，$F_1=2^{N-L}+M$，$G_y \geq 2^{N-L}+M-1$，…$G_1=1$。

17. 根据权利要求 13 的 CRC 代码生成电路，其中，所述 $F_x=F_{x-1}=\ldots=F_1=S$；且 $G_y=y-1$，…，$G_1=1$，其中 $y \geq S-1$。

用于可变宽度并行循环冗余校验计算的迭代电路和方法

技术领域

本发明一般涉及在通信系统中基于包的循环冗余校验的实现，并尤其涉及用于在通信系统中执行循环冗余校验计算以及对其进行时间优化的迭代电路。

背景技术

许多基于包的通信协议使用附加在包传输中的代码字来检查在通信信道中引入的错误的存在。一种常用的生成这种代码字的方法是循环冗余校验（CRC）。发射机在包的结尾附加 CRC 代码字，而接收机对于整个包包括代码字重新计算 CRC。几种 CRC 方法是常用的；不同的方法为计算使用不同的多项式，并且得到的代码字长度也不同。

对于在串行数据流中传送的包，在发射机或接收机中计算 CRC 代码字所需的逻辑电路是公知、且非常有效的。带有实现目标多项式所需的异或门的线性反馈移位寄存器是一种足够的实现。移位寄存器的每个状态是基于当前的串行位以及移位寄存器的前一个状态计算的。因此，对于串行数据流，n 个锁存器（n 是多项式的阶数）以及一些异或门就是所需要的电路的范围。

但是，高速串行数据接口（例如 10Gbps、40Gbps 或更高的接口）经常需要更昂贵的技术（例如 SiGe（锗硅））来实现串行波特率的数据信号。这种接口使用高速模拟电路来实现高速接口，并通常将去往/来自该串行接口的数据多路复用/解复用到较慢的并行数据路径中，以便在 CMOS 芯片内进行处理。因此，CRC 计算电路更通常地在并行数据总线上操作。如果数据总线的宽度是“w”个字节，则 CRC 计算必须同时处理 w 个字节以

确定CRC计算的下一个状态。此外，由于CRC计算的下一个状态依赖于该计算的前一个状态，因此该计算不适于流水线操作。

当包数据不能保证是整数个w字节，和/或不能保证在并行数据总线上的对准位置开始/结束时，会引入进一步的复杂性。例如，给定32字节宽的数据总线，则CRC计算电路必须能够处理任何一个可能产生的计算宽度：w=1，2，3，4，...，31，32字节。这使得用于CRC计算的下一个状态解码变得非常复杂。所产生的逻辑电路会需要非常大的芯片面积。此外，由于该芯片面积主要由具有大的扇出连接的组合逻辑占用，会导致可布线性和时间方面的问题。

为了满足系统要求，CRC计算逻辑必须通常包括多个不同宽度的CRC计算块，且数据引导装置将数据选入将在任何给定周期上使用的每个块内。一种现有技术的实现是实现w字节宽的数据总线，并因此使用大小为1字节、2字节、3字节等直到w字节的“w”个CRC计算块，以实现功能。在此结构中，数据被并行地馈入所有这些块。在任何给定的时钟周期上，仅使用一个CRC计算块的输出。即，在此并行方法中，在每个周期内选择一个且仅一个CRC计算块，从而组合传播延时将总是等于一个CRC计算块的延时。

非常希望为CRC计算电路提供一种结构化的迭代方法，借此该CRC计算可被细分到总线宽度可选择的块，该些块可级联以为任意字节数量的总线宽度提供计算。

非常希望为CRC计算电路提供一种结构化的迭代方法，以便对于给定的目标传播延时使电路面积的减小最大。

发明内容

本发明是一种基于以下认识而优化CRC计算的方法，该认识即CRC计算块的大小与计算的宽度成正比，并且与减少用于窄计算宽度的块的数量相比减少用于宽计算宽度的块的数量可提供更大的节约。

因此，本发明的一个目标是为CRC计算电路提供一种结构化的迭代

方法，借此 CRC 计算可被细分到具有可选择的总线宽度的块，该些块可级联在一起以为任意字节数量的并行总线宽度提供计算。

本发明的另一个目标是为 CRC 计算电路提供一种结构化的迭代方法，借此 CRC 计算可被细分到具有可选择的总线宽度的块，该些总线宽度为二（2）的幂例如 2^N，例如 N = 0，1，…，X 个字节，该些块可级联在一起以提供对任意字节数量的总线宽度的计算。

本发明的另一个目标是为 CRC 计算电路提供一种结构化的对数迭代方法，借此 CRC 计算可被细分到允许可选择的总线宽度的块，该些总线宽度的值不是 2 的幂。

该 CRC 计算的结构化方法由迭代电路执行，借此根据优选实施例，提供了一种用于生成与将在通信信道上传送的宽度最大为 w 个字节的数据相关联的 CRC 代码字的系统，该系统包括：

第一多个串行连接的代码生成块，每个该块用于根据每个块的数据输入生成一 CRC 值，该第一多个块的各个块构造成接收字节宽度分别在 $2^N + M$ 到 $2^{N-L} + M$ 的范围内的数据输入，其中 $2^N + M = w$，M 是偏移值，L 是基于最大传播延时标准的整数；

第二多个并行连接的代码生成块，每个该块用于根据数据输入生成一 CRC 值，该第二多个块的各个块构造成接收字节宽度分别在 $2^{N-L} - 1 + M$ 到 2^0 的范围内的数据输入；以及

用于根据数据输入来选择将包含在 CRC 计算中的第一和第二多个块中的特定 CRC 代码生成块；从而可处理任意宽度的数据输入字节。

根据本发明的原理，基于以下认识优化了该结构化的迭代方法中的 CRC 计算处理时间，该认识即 CRC 计算块的大小与计算的宽度成正比，并且与减少用于窄计算宽度的块的数量相比减少用于宽计算宽度的块的数量可提供更大的节约。有利地是，对于宽的数据总线宽度，该结构化的、对数迭代方法大大减少了执行计算所需的逻辑的数量。

附图说明

考虑到下文结合附图进行的详细说明，本发明的目标、特征和优点对于本技术领域内的技术人员将很明显，在该些附图中：

图 1 是根据本发明的第一实施例形成的整个系统体系结构的框图，本发明可在该系统体系结构内操作；

图 2 是根据本发明的第二实施例形成的整个系统体系结构的框图，本发明可在该系统体系结构内操作；

图 3 是根据本发明的第三实施例形成的整个系统体系结构的框图，本发明可在该系统体系结构内操作；

图 4 是根据本发明的第四实施例形成的整个系统体系结构的框图，本发明可在该系统体系结构内操作；

图 5 是根据本发明的一般实施例形成的整个系统体系结构的框图，本发明可在该系统体系结构内操作；

具体实施方式

参照图 1 可更全面地了解本发明，该附图示出根据本发明的第一实施例的整个系统体系结构。

该第一实施例是针对一种对于 CRC 计算电路的结构化的对数迭代方法，该方法提供了 CRC 计算块的级联，级联的块的数量依赖于计算所期望的计算宽度。

图 1 是根据本发明的第一变型的 CRC 计算电路 100 的框图。根据此第一变型，CRC 计算电路 100 可细分为一些块，该些块具有被对数地选择的总线宽度 1 字节、2 字节、4 字节、8 字节等（即 2 的幂字节）。这些块可级联在一起来为为任意字节数量的总线宽度提供计算。

如图 1 中所示，逻辑电路 100 包括一些 CRC 计算块（220、320、340），每个块根据来自种子（seed）多路复用器（130、230、330、430）的种子输入以及根据数据输入（120）来计算 CRC 值。应理解，w 字节的传入数据输入路径宽度应等于 2^N。即对于 w 个字节的数据总线，$N = \log_2(w)$，并且该电路包括 N + 1 个针对字节宽度为 2^N，2^{N-1}，…，2^1，2^0 的 CRC 计

算块。当进行计算时，CRC_seed_select 信号控制种子多路复用器（230、330、430），以选择在该计算中是包括相应的 CRC 计算块，还是绕过该计算块。通过选择性地包括或绕过这些计算块，可处理任意数量的字节。例如，要处理 w 个字节，选择 2^N 块，并绕过所有其他的块；要处理 w－1 个字节，则绕过 2^N 块，而选择块 2^{N-1}，…，2^0；要处理 w－2 个字节，则绕过 2^N 和 2^0 块，而选择块 2^{N-1}，…，2^1，等等。应理解，每个 CRC 块都是一组合 XOR 树，其精确设计依赖于实施的 CRC 多项式。

在开始时，电路以 CRC_reset 信号有效开始，从而多路复用器 130 选择为 0 的初始种子值。然后，由被种子多路复用器（230、330、430）选择或绕过的 CRC 计算块（220、320、420）执行选择的 CRC 计算。在多路复用器 430 处的 CRC 计算输出被保存在 CRC 结果寄存器 110 中。根据数据输入（120）在每个周期内更新 CRC 寄存器的值。数据引导多路复用器（210、310、410）根据选择了哪些块，来选择将用于每个 CRC 计算块的数据。在包的最后一个周期上，CRC 输出 510 向下游逻辑提供该包的计算出的 CRC 值。

在本发明的一个示例性实施例中，提供了一个 32 字节宽的数据输入。因此，此 N＝5 系统具有宽度分别为 32 字节、16 字节、8 字节、4 字节、2 字节和 1 字节的 6 个 CRC 计算块。名义上，包数据占用全部数据宽度，并选择 32 字节宽的 CRC 计算块。但是，在包传输开始和结束时包可能仅占用该宽度的一部分。在这些情况下，根据总线上的预期数据对准来生成 CRC_select 和 Data_select 控制信号。

对于宽的数据总线宽度，根据第一变型的方法大大减少了执行计算所需的逻辑的数量。（对于 w＝32 字节，可实现 88%的逻辑大小减小）。对于 w 字节宽的数据总线，其中级联的块的数量依赖于计算的期望宽度，最坏情况的传播延时在计算宽度为 w－1 时发生，在此时 $\log_2$（w）个 CRC 计算块级联。

根据此实施例，将在几个时钟周期上计算长于 w 字节的传入数据包的 CRC。假设 R 代表必须在给定的计算时钟周期内处理的字节数。尽管在任

何时钟周期上 R 可采用任意值（$R \leqslant w$），但是 CRC 计算需要一定数量的其间 $R = w$ 的周期，以及一个其间将处理任何剩余的字节的周期。

因此，第一实施例的控制级可处理 R 字节的数据（$R \leqslant w$），并且其中计算电路包括大小为 2^N，2^{N-1}，2^{N-2}，...，4（$=2^2$），2（$=2^1$），1（$=2^0$）字节的 CRC 计算块。如果 $w = 32$，则存在 32 字节、16 字节、8 字节、4 字节、2 字节、1 字节的块。该控制逻辑使控制信号 A_{n-1}，A_{n-2}，...，A_0 有效，从而 $R \leqslant 32$；$R = (A_{n-1}) *2^{(N-1)} + (A_{n-2}) *2^{(N-2)} + ... + (A_0) *1$。控制信号 A_{n-1}，A_{n-2}，...，$A_0 = 1$ 或 0 如下所述地选择块。

在 R = 32 时　使用 32_字节或 16_字节 + 16_字节　CRC

在 R = 31 时　使用 16_字节 + 8_字节 + 4_字节 + 2_字节 + 1_字节　CRC

当 R = 30 时　使用 16_字节 + 8_字节 + 4_字节 + 2_字节　CRC

当 R = 29 时　使用 16_字节 + 8_字节 + 4_字节 + 1_字节　CRC

当 R = 28 时　使用 16_字节 + 8_字节 + 4_字节　CRC

当 R = 27 时　使用 16_字节 + 8_字节 + 2_字节 + 1_字节　CRC

当 R = 26 时　使用 16_字节 + 8_字节 + 2_字节　CRC

当 R = 25 时　使用 16_字节 + 8_字节 + 1_字节　CRC

当 R = 24 时　使用 16_字节 + 8_字节　CRC

当 R = 23 时　使用 16_字节 + 4_字节 + 2_字节 + 1_字节　CRC

当 R = 22 时　使用 16_字节 + 4_字节 + 2_字节　CRC

当 R = 21 时　使用 16_字节 + 4_字节 + 1_字节　CRC

当 R = 20 时　使用 16_字节 + 4_字节　CRC

当 R = 19 时　使用 16_字节 + 2_字节 + 1_字节　CRC

当 R = 18 时　使用 16_字节 + 2_字节　CRC

当 R = 17 时　使用 16_字节 + 1_字节　CRC

当 R = 16 时　使用 16_字节　CRC

当 R = 15 时	使用 8_字节 + 4_字节 + 2_字节 + 1_字节	CRC
当 R = 14 时	使用 8_字节 + 4_字节 + 2_字节	CRC
当 R = 13 时	使用 8_字节 + 4_字节 + 1_字节	CRC
当 R = 12 时	使用 8_字节 + 4_字节	CRC
当 R = 11 时	使用 8_字节 + 2_字节 + 1_字节	CRC
当 R = 10 时	使用 8_字节 + 2_字节	CRC
当 R = 9 时	使用 8_字节 + 1_字节	CRC
当 R = 8 时	使用 8_字节	CRC
当 R = 7 时	使用 4_字节 + 2_字节 + 1_字节	CRC
当 R = 6 时	使用 4_字节 + 2_字节	CRC
当 R = 5 时	使用 4_字节 + 1_字节	CRC
当 R = 4 时	使用 4_字节	CRC
当 R = 3 时	使用 2_字节 + 1_字节	CRC
当 R = 2 时	使用 2_字节	CRC
当 R = 1 时	使用 1_字节	CRC

对于每个 32 字节，...，1 字节的 CRC 模块，将相应字节长度的数据发送给该模块。例如：当 R = 10 时，则使用 8_字节 + 2_字节模块，然后使控制信号 'Data_select' 有效（图 1 中所示），以选择数据的前 8 个字节发送给 8_字节 CRC 计算块，并将数据的后两个 2 字节发送给 2_字节 CRC 计算块。

本发明的第二变型在现有技术方法和第一变型即对数迭代方法之间提供了优化。根据此实施例，对于给定的目标传播延时，使面积减小最大。这是通过注意到 CRC 计算块的大小与计算的宽度成正比来实现的。因此，与减少用于窄计算宽度的块的数量相比，减少用于宽的计算宽度的 CRC 计算块的数量可实现更大的节约。同时，对数迭代方法中的级联块中的传播延时主要是通过窄宽度的 CRC 计算块的延时。因此，通过对较宽的计算宽度使用对数迭代方法，而对较小的计算宽度使用并行方法，对于该电

路提供了时间相对于面积的优化。

假设一系统具有宽度为 w 字节的数据总线。假设 $N = \log_2(w)$ 和 d_{max} = 允许的最大延时（以 CRC 计算块延时为单位），则 $L = d_{max} - 1$。

于是，通过对于宽度为 2^{N-L} 或更大的 CRC 计算块使用对数迭代方法，而对于宽度小于 2^{N-L} 的 CRC 计算块使用并行方法，来构造根据第二变型的 CRC 计算系统。所得的系统包括用于 2^N，2^{N-1}，…，$2^{N-L}+1$，2^{N-L}，$2^{N-L}-1$，$2^{N-L}-2$，…，2^1，2^0 字节宽度的 CRC 计算块。所得的系统在该系统的对数迭代部分内包括 $L+1$ 个 CRC 计算块，而在电路的并行部分内包括 $2^{N-L}-1$ 个 CRC 计算块。这种系统的最坏情况的延时发生在计算在 2^{N-1} 到 $2^{N-1}+1$ 范围内的字节宽度时。在此范围内，存在 L 个级联的迭代 CRC 计算块，和一个并行的 CRC 计算块，这些块是传播必须通过的。

本发明的第二变型假设 CRC 计算的目标字节宽度（w）为 2 的幂。

图 2 是根据所公开的本发明的第二变型的 CRC 计算电路 200 的框图。如图 2 中所示，逻辑电路 200 包括一些 CRC 计算块（220、320、420、520、521），每个计算块根据来自种子多路复用器（130、230、330、430）的种子输入以及根据数据输入（120）来计算 CRC 值。对于具有 w 字节的数据总线和最大延时 d_{max} 的系统，$N = \log_2(w)$，且 $L = d_{max} - 1$。该系统的对数迭代部分包括 $L+1$ 个针对字节宽度 2^N，2^{N-1}，…，2^{N-L} 的 CRC 计算块（220、320、420）。该系统的并行部分包括 2^{N-L} 个针对字节宽度 $2^{N-L}-1$，…，2^0 的 CRC 计算块（520、521）。当进行计算时，CRC_seed_select 信号控制种子多路复用器（230、330、430），以选择每个迭代 CRC 计算块是被包括在该计算中还是被绕过。通过选择性地包括或绕过这些计算块，可处理可被 2^{N-L} 除尽的任何数量的字节。另外，多路复用器 530 选择，如果有的话将选择哪个并行 CRC 计算块（520、521）来提供输出。这使处理能力扩展到任意数量的字节。

例如，要处理 w 个字节，则选择 2^N 块，绕过所有其他的迭代块，并且多路复用器 530 选择旁路输入；要处理 $w-1$ 个字节，则绕过 2^N 块，选择所有其他迭代块（即 2^{N-1}，…，$2^{N-L}+1$，2^{N-L}），并且多路复用器 530

选择来自 $2^{N-L}-1$ 块的输入；等等。每个 CRC 块都是一组合 XOR 树，其精确设计依赖于实施的 CRC 多项式。

在开始时，电路以 CRC_reset 信号有效开始，从而多路复用器 130 选择为 0 的初始种子值。然后，由被种子多路复用器（230、330、430）选择或绕过的迭代 CRC 计算块（220、320、420），以及由被输出多路复用器 530 选择或绕过的并行 CRC 计算块（520、521）执行选择的 CRC 计算。在多路复用器 530 处的 CRC 计算输出被保存在 CRC 结果寄存器 110 中。根据数据输入（120）在每个周期内更新 CRC 寄存器的值。数据引导多路复用器（210、310、410）根据选择了哪些块，来选择将用于每个迭代 CRC 计算块的数据。数据引导多路复用器 510 选择将被所选择的并行 CRC 计算块使用的数据。在包的最后一个周期上，CRC 输出 610 向下游逻辑提供该包的计算出的 CRC 值。

在实现本发明的第二变型的示例性实施例中，对于一个 32 字节宽的数据总线输入，$d_{max}=3$。这会得到一个 N=5、L=2 的系统，该系统具有宽度为 32 字节、16 字节和 8 字节的三（3）个迭代 CRC 计算块，和宽度从 7 字节降至 1 字节的 7 个并行 CRC 计算块。名义上，包数据占用全部数据宽度，并选择该 32 字节宽的 CRC 计算块。但是，在包传输开始和结束时包可能仅占用该宽度的一部分。在这些情况下，根据总线上的预期数据对准来生成 CRC_select 和 Data_select 控制信号。

在根据图 2 的第二变型的时间优化方案中，与图 1 相比，关键路径中的级数较少以便实现更好的用时。与先前实施例的情况相同，在多个时钟周期上执行长度大于 w 字节的数据包的 CRC 计算。假设，R 代表必须在给定的计算时钟周期内处理的字节数。

这样，第二实施例的控制级可处理 R 字节的数据（$R \leqslant w$），且其中计算电路包括大小为 2^N，$2^{(N-1)}$，$2^{(N-2)}$，…，$2^{(N-L)}$，$2^{(N-1)}-1$，$2^{(N-1)}-2$，…，和 1 字节的 CRC 计算块。如果 w=32，N=5 和 L=2，则存在 32 字节、16 字节、8 字节、7 字节、6 字节、5 字节、4 字节、3 字节、2 字节、1 字节 CRC 级。控制逻辑使控制信号 A_{n-1}，A_{n-2}，…，A_{n-L}，B_k

$_{-1}$，...，B_0有效，以选择对R个字节的处理，以使$R \leq 32$，且$R = (A_{n-1}) *2^{(N-1)} + (A_{n-2}) *2^{(N-2)} + ... + (A_0) *2^{(N-L)} + (B_{k-1}) * (2^{(N-L)} - 1) + ... + (B_0) *1$，其中$k = 2^{N-L}$。控制信号$A_{n-1}$，$A_{n-2}$，...，$A_{n-L}$，$B_{k-1}$，...，$B_0 = 1$或0如下所述地选择块。

在 R = 32 时	使用 32_字节或 16_字节 + 16_字节	CRC
在 R = 31 时	使用 16_字节 + 8_字节 + 7_字节	CRC
当 R = 30 时	使用 16_字节 + 8_字节 + 6_字节	CRC
当 R = 29 时	使用 16_字节 + 8_字节 + 5_字节	CRC
当 R = 28 时	使用 16_字节 + 8_字节 + 4_字节	CRC
当 R = 27 时	使用 16_字节 + 8_字节 + 3_字节	CRC
当 R = 26 时	使用 16_字节 + 8_字节 + 2_字节	CRC
当 R = 25 时	使用 16_字节 + 8_字节 + 1_字节	CRC
当 R = 24 时	使用 16_字节 + 8_字节	CRC
当 R = 23 时	使用 16_字节 + 7_字节	CRC
当 R = 22 时	使用 16_字节 + 6_字节	CRC
当 R = 21 时	使用 16_字节 + 5_字节	CRC
当 R = 20 时	使用 16_字节 + 4_字节	CRC
当 R = 19 时	使用 16_字节 + 3_字节	CRC
当 R = 18 时	使用 16_字节 + 2_字节	CRC
当 R = 17 时	使用 16_字节 + 1_字节	CRC
当 R = 16 时	使用 16_字节	CRC
当 R = 15 时	使用 8_字节 + 7_字节	CRC
当 R = 14 时	使用 8_字节 + 6_字节	CRC
当 R = 13 时	使用 8_字节 + 5_字节	CRC
当 R = 12 时	使用 8_字节 + 4_字节	CRC
当 R = 11 时	使用 8_字节 + 3_字节	CRC
当 R = 10 时	使用 8_字节 + 2_字节	CRC

当 R = 9 时	使用 8_字节 + 1_字节	CRC
当 R = 8 时	使用 8_字节	CRC
当 R = 7 时	使用 7_字节	CRC
当 R = 6 时	使用 6_字节	CRC
当 R = 5 时	使用 5_字节	CRC
当 R = 4 时	使用 4_字节	CRC
当 R = 3 时	使用 3_字节	CRC
当 R = 2 时	使用 2_字节	CRC
当 R = 1 时	使用 1_字节	CRC

用于执行循环冗余校验计算及对其进行时间优化的电路的第三实施例涉及一种结构化的对数迭代方法，该方法更为一般，允许不是 2 的幂的“w”值。

图 3 是根据本发明的第三变型的 CRC 计算电路 300 的框图。根据此第三变型，系统的逻辑迭代部分包括 L + 1 个字节宽度已被更一般地指定为 $2^N + M$，$2^{N-1} + M$，...，$2^{N-L} + M$ 的 CRC 计算块（220、320、420），其中 M 是一包含任意常数的正的偏移值（即大于或等于 0）。根据第三变型的这个更一般的表示使得该系统的目标字节宽度（w）可以是除 2 的幂之外的值。相应地，该系统的并行部分包括针对字节宽度 $2^{N-L} - 1 + M$，...，2^0 的 $2^{N-L} + M$ 个 CRC 计算块（520、521）。与前文所述的此发明的变型相同，当执行计算时，CRC_seed_select 信号控制种子多路复用器（230、330、430），以选择每个迭代 CRC 计算块是被包含在该计算中还是被绕过。通过选择性地包含或绕过这些计算块，可处理可被 2^{N-L} 除尽的任意数量的字节。另外，多路复用器 530 选择，如果有的话将选择哪个并行 CRC 计算块（520、521）来提供输出。

图 4 是根据本发明的第四实施例的 CRC 计算电路 400 的框图，其中每个 CRC 计算块（220、320、420 等）能够处理 S 个字节，其中 S 是任意的正整数。电路的迭代部分包括“k”个块，每个块能够处理 S 个字节，

其中（k+1）*S>w，以及 k*S≤w。系统的并行部分包括针对字节宽度 S－1，S－2，…，1 的 B－1 个计算块。当执行计算时，CRC_seed_select 信号控制种子多路复用器（130、230、330、430），以选择将被处理的 S 个字节的倍数。另外，多路复用器 530 选择，如果有的话将选择哪个并行 CRC 计算块（520、521）来提供输出。这使该处理能力扩展到任意数量的字节。

与使用第二和第三实施例的实现相比第四实施例减小了逻辑，同时与使用第一实施例的实现相比第四实施例仍优化了时间。假设 d 代表电路的延时（以 CRC 计算块为单位）。于是，通过此电路的 d 的最坏情况的值发生在计算在 k*S 到 w 字节范围内的 x 个字节时，此时选择 k 个块来在前面 k*S 个字节上执行计算，并选择一个并行块来在剩余字节上执行计算，其中 d=k+1。对于 w=33 字节、L=2、M=1、k=4 的情况，使用第三实施例的实现的最坏情况的延时 d=2 个块，而此实施例的最坏情况的延时 d=5 个块。但是，由于与第二实施例相比宽度较大的 CRC 计算块的数量减少了，因此此实施例减小了逻辑。根据此实施例，通常并行构造的 CRC 计算块的数量等于 S－1，但是对于更积极的时间方案，并行构造的 CRC 计算块的数量可超过 S。

因此，第四实施例的控制级可处理 R（R≤w）字节的数据，且其中计算电路包括 k 个 S 字节的块以及 S－1 个字节为 S－1，S－2，…，1 的并行块。例如，根据图 4 中所示的本发明的第四变型，其中 w=34 和 S=5，控制逻辑使控制信号 A_{k-1}，A_{k-2}，…，A_0，B_{s-1}，…，B_0 有效，以选择对 R 个字节的处理，以使 R≤34；且 $R=(A_{k-1})*S+(A_{k-2})*S+\ldots+(A_0)*S+(B_{s-1})*(S-1)+\ldots+(B_0)*1$。控制信号 A_{k-1}，A_{k-2}，…，A_0，B_{s-1}，…，B_0=1 或 0 如下所述（其中 X 代表在给定时钟周期期间选择的 S 字节的块的数量，其选择在其他方面是任意的）。

34 字节 CRC=S_字节*6+4_字节　　其中 X=6（整条路径）

33 字节 CRC=S_字节*6+3_字节　　其中 X=6

32 字节 CRC = S_字节*6 + 2_字节 其中 X = 6

31 字节 CRC = S_字节*6 + 1_字节 其中 X = 6

30 字节 CRC = S_字节*6 其中 X = 6

29 字节 CRC = S_字节*5 + 4_字节 其中 X = 5

25 字节 CRC = S_字节*5 其中 X = 5

24 字节 CRC = S_字节*4 + 4_字节 其中 X = 4

23 字节 CRC = S_字节*4 + 3_字节 其中 X = 4

22 字节 CRC = S_字节*4 + 2_字节 其中 X = 4

21 字节 CRC = S_字节*4 + 1_字节 其中 X = 4

20 字节 CRC = S_字节*4 其中 X = 4

19 字节 CRC = S_字节*3 + 4_字节 其中 X = 3

18 字节 CRC = S_字节*3 + 3_字节 其中 X = 3

17 字节 CRC = S_字节*3 + 2_字节 其中 X = 3

16 字节 CRC = S_字节*3 + 1_字节 其中 X = 3

15 字节 CRC = S_字节*3 其中 X = 3

14 字节 CRC = S_字节*2 + 4_字节 其中 X = 2

13 字节 CRC = S_字节*2 + 3_字节 其中 X = 2

12 字节 CRC = S_字节*2 + 2_字节 其中 X = 2

11 字节 CRC = S_字节*2 + 1_字节 其中 X = 2

10 字节 CRC = S_字节*2 其中 X = 2

9 字节 CRC = S_字节 + 4_字节 其中 X = 1

8 字节 CRC = S_字节 + 3_字节 其中 X = 1

7 字节 CRC = S_字节 + 2_字节 其中 X = 1

6 字节 CRC = S_字节 + 1_字节 其中 X = 1

5 字节 CRC = S_字节 其中 X = 1

4 字节 CRC = 4_字节 其中 X = 0

3 字节 CRC = 3_字节 其中 X = 0

2 字节 CRC = 2_字节 其中 X = 0

1 字节 CRC = 1_字节 其中 X = 0

图 5 示出包含根据图 1 - 4 说明的本发明的全部四个变型的 CRC 计算电路 450 的一般形式，其中对于串行级联的块（220、320、420 等）的迭代部分，CRC 计算块的字节宽度的大小在字节 F_x，F_{x-1}，...，F_1 的范围内，其中 $F_x \geqslant F_{x-1} \geqslant \ldots \geqslant F_1 \geqslant 0$；而对于并联块（520，521 等），CRC 计算块的字节宽度的大小为 G_{y-i} 字节，其中对于 i = 0 到 y - 1，$G_{y-i} = G_{y-i-1} + 1$，而 $y \geqslant F_1 - 1$。根据此一般形式，CRC 计算中包含的字节的数量可由以下关系表示：$R = A_x*F_x + A_{x-1}* F_{x-1} + \ldots + A_1*F_1 + B_y*G_y + \ldots + B_1*G_1$，其中 x 是 F_x 块的号码，y 是 G_x 块的号码，且如四个实施例之一中所述的那样，A_{x-1}，A_{x-1}，...，A_0，B_y，...，$B_1 = 0$ 或 1。

图 5 示出本发明的所有变型的一般形式。对于第一实施例，$F_x = 2^N$，$F_{x-1} = 2^{N-1}$，...，$F_1 = 2^0$；$y = F_1 - 1 = 0$，从而在电路的并行部分内没有 CRC 计算块。对于第二实施例，$F_x = 2^N$，$F_{x-1} = 2^{N-1}$，...，$F_1 = 2^{N-L}$，$G_y = 2^{N-L} - 1$，...$G_1 = 1$。对于第三实施例，$F_x = 2^N + M$，$F_{x-1} = 2^{N-1} + M$，...，$F_1 = 2^{N-L} + M$，$G_y = 2^{N-L} + M - 1$，...$G_1 = 1$。对于第四实施例，$F_x = F_{x-1} = \ldots = F_1 = S$；而 $G_y = y - 1$，...，$G_1 = 1$，其中 $y \geqslant S - 1$。

尽管已参照本发明的示例性的和优选的实施例具体地示出和说明了本发明，但是本技术领域内的技术人员应理解，可在形式和细节上进行前述的以及其他改变而不会背离本发明的精神和范围，本发明的精神和范围仅由所附权利要求的范围限定。

图 1

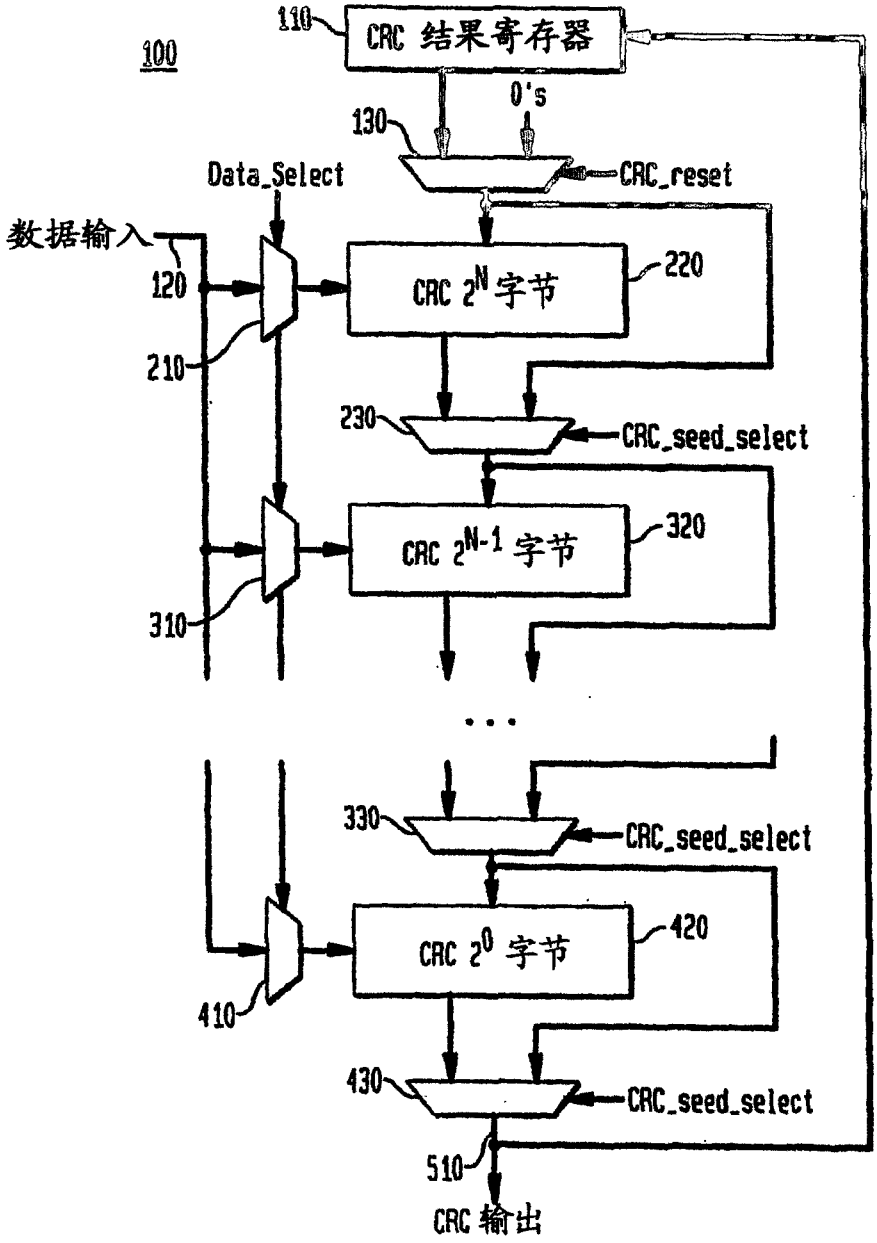

100
110
CRC 结果寄存器
0's
130
Data_Select
CRC_reset
数据输入
120
CRC 2^N 字节
220
210
230
CRC_seed_select
CRC 2^{N-1} 字节
320
310
330
CRC_seed_select
CRC 2^0 字节
420
410
430
CRC_seed_select
510
CRC 输出

图 2

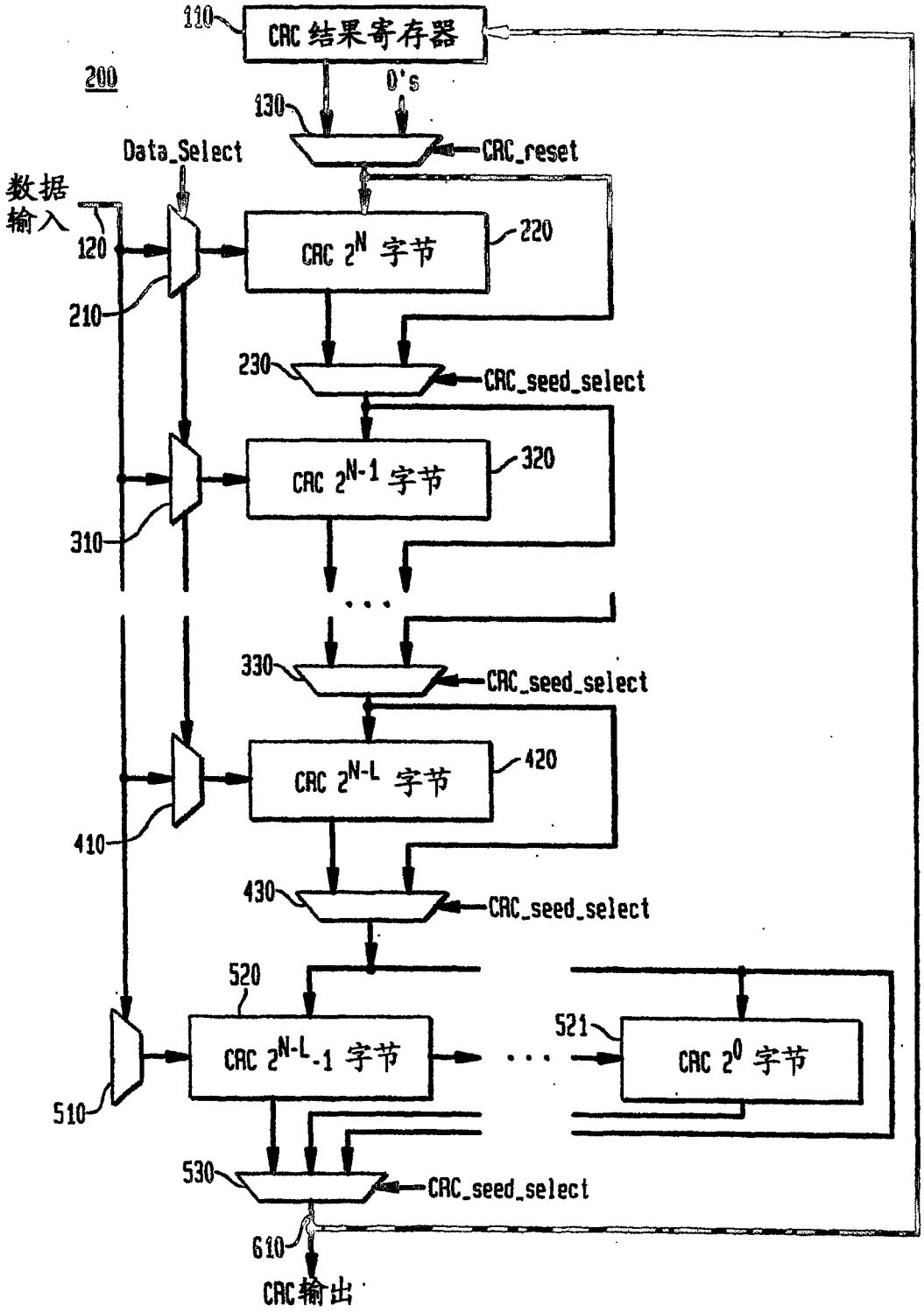

110
CRC 结果寄存器
200
0's
130
CRC_reset
Data_Select
数据输入
120
CRC 2^N 字节
220
210
230
CRC_seed_select
CRC 2^{N-1} 字节
320
310
330
CRC_seed_select
CRC 2^{N-L} 字节
420
410
430
CRC_seed_select
520
521
CRC 2^{N-L-1} 字节
CRC 2^0 字节
510
530
CRC_seed_select
610
CRC输出

图 3

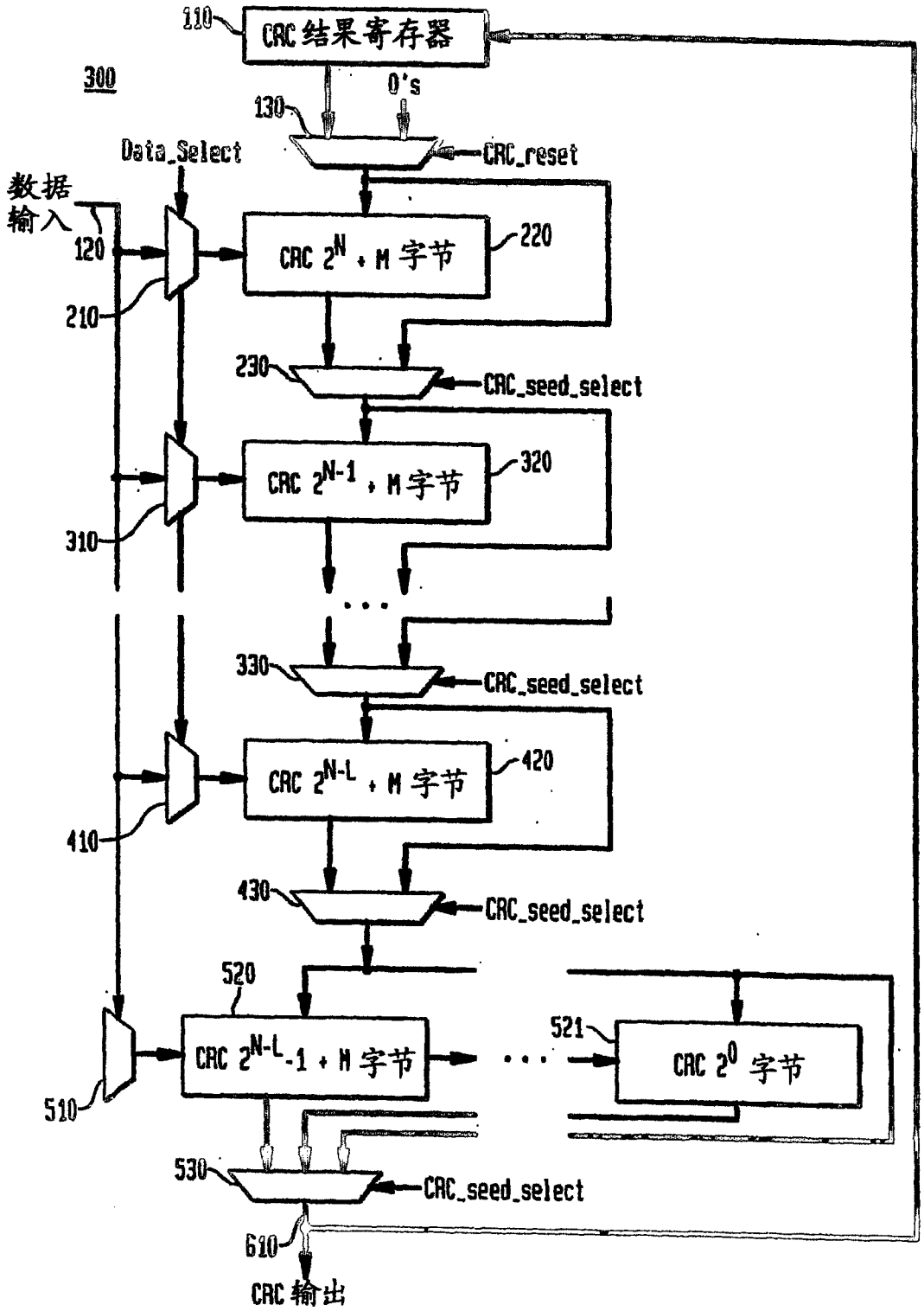

110
CRC结果寄存器
300
0's
130
Data_Select
CRC_reset
数据
输入
120
CRC 2^N + M 字节
220
210
230
CRC_seed_select
CRC 2^{N-1} + M 字节
320
310
330
CRC_seed_select
CRC 2^{N-L} + M 字节
420
410
430
CRC_seed_select
520
CRC 2^{N-L}-1 + M 字节
521
CRC 2^0 字节
510
530
CRC_seed_select
610
CRC 输出

图 4

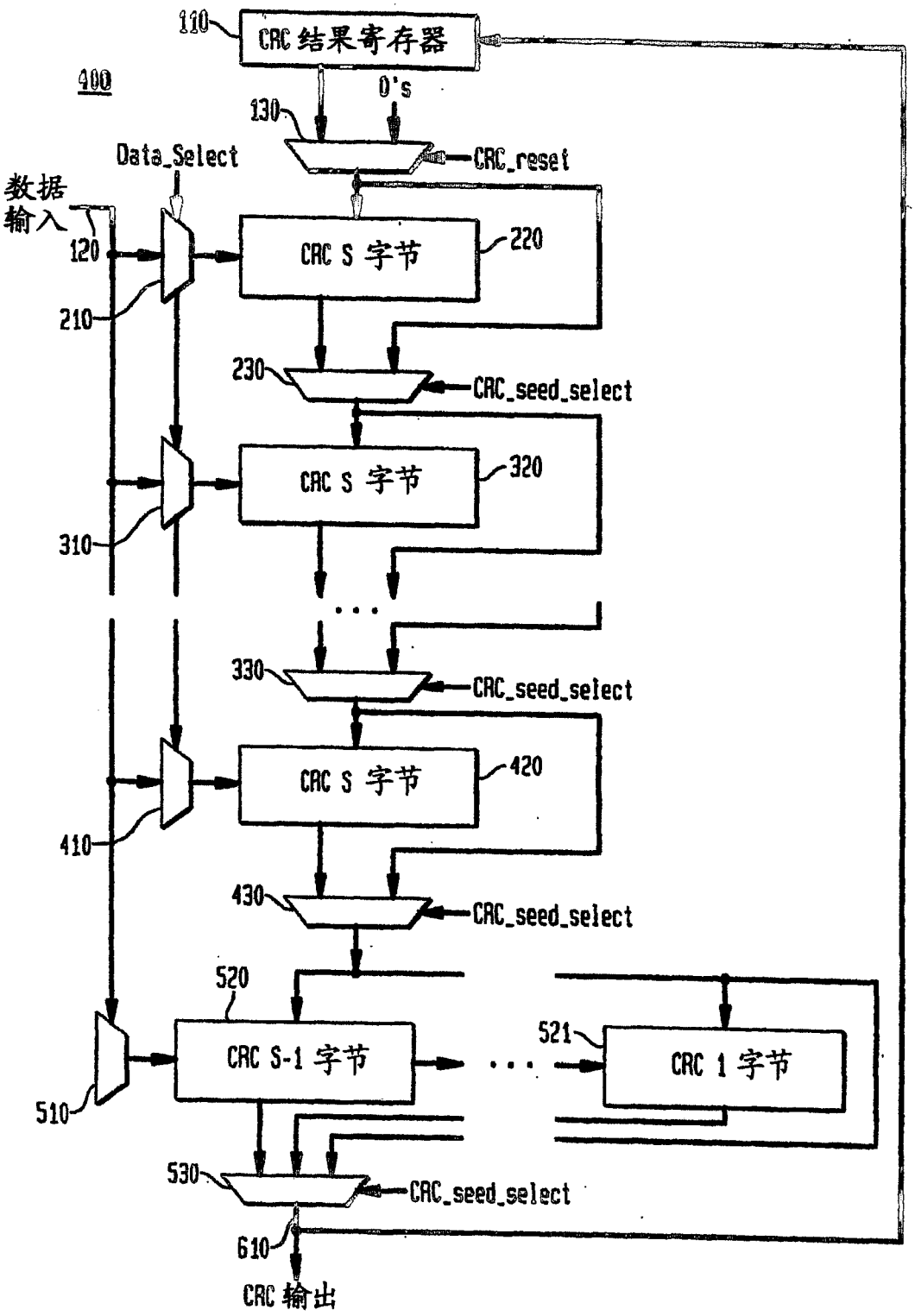
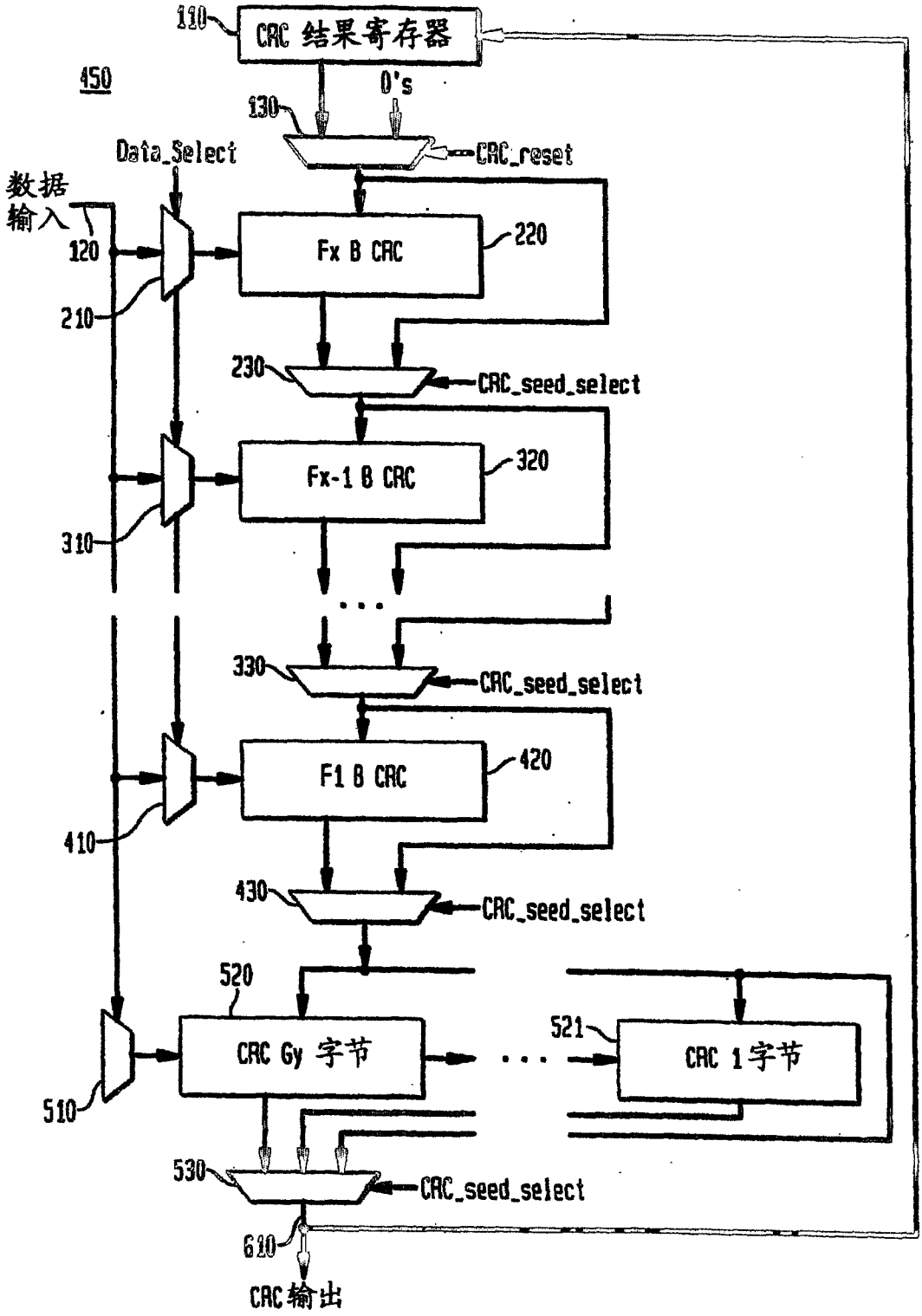

110
CRC 结果寄存器
400
0's
130
Data_Select
CRC_reset
数据
输入
120
CRC S 字节
220
210
230
CRC_seed_select
CRC S 字节
320
310
330
CRC_seed_select
CRC S 字节
420
410
430
CRC_seed_select
520
CRC S-1 字节
521
CRC 1 字节
510
530
CRC_seed_select
610
CRC 输出

图 5


110
CRC 结果寄存器
450
0's
130
Data_Select
CRC_reset
数据
输入
120
Fx B CRC
220
210
230
CRC_seed_select
Fx-1 B CRC
320
310
330
CRC_seed_select
F1 B CRC
420
410
430
CRC_seed_select
520
CRC Gy 字节
521
CRC 1字节
510
530
CRC_seed_select
610
CRC输出