



[12] 发明专利申请公开说明书

[21] 申请号 96198005.2

[43]公开日 1998 年 12 月 9 日

[11] 公开号 CN 1201536A

[22]申请日 96.10.29

[30]优先权

[32]95.10.30[33]FI[31]955187

[86]国际申请 PCT / FI96 / 00572 96.10.29

[87]国际公布 WO97 / 16786 英 97.5.9

[85]进入国家阶段日期 98.4.30

[71]申请人 诺基亚电信公司

地址 芬兰埃斯波

[72]发明人 贾莫·林尼

凯里·帕桑宁

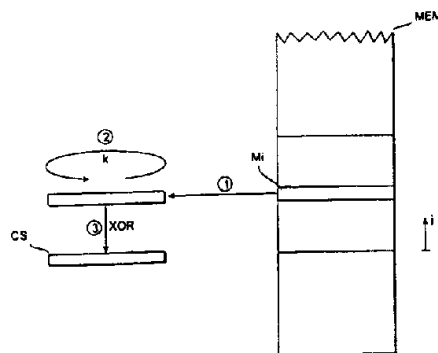
[74]专利代理机构 中国国际贸易促进委员会专利商标
事务所
代理人 蒋世迅

权利要求书 2 页 说明书 8 页 附图页数 2 页

[54]发明名称 检测计算机中的存储器问题

[57]摘要

在控制计算机的过程中，如开关交换，可以证明解决故障花费很大。解决软件缺陷尤其困难。如果问题发生，重要的是尽快直接维修正确的目标。破坏者和黑客构成了威胁。如果从别处用于维修的选项合并于计算机中，那么误用的危险将相应地增加。用于发现存储器问题的现有技术不能必要地揭示由计算方法引起的以有规律的间隔发生的缺陷。举例来说，如果存储器电路的一根地址线出错，存储器缺陷将以有规律的间隔发生。现有技术的第二个缺点是即使只是一个存储器位置变化也不得不重新计算校验和。根据本发明发现存储器问题的方法是“有规律地无规律”，使得它能以可靠的方式发现在任何存储区域（MA）内的未受控制的变化。此外，根据本方法计算的校验和（CS）可以根据存储区域（MA）内单元的变化更新，使得与在被监视的整个存储区域（MA）内计算校验和相比，更新需要较少的计算。本方法适用于发现硬件和软件缺陷以及加速计算机的重新启动。





权利要求书

1.一种方法，用于在包括存储器单元(M_i)的一个存储区域(MA)计算一个M位校验和(CS)，其特征在于以下步骤：

-在计算之前给该校验和(CS)一个预定的初始值，

-在存储区域(MA)的每个存储器单元(M_i)处计算校验和(CS)，作为一个逐位求和(异或)，逐位求和的第一因子在该单元(M_i)的基础上形成，逐位求和的第二因子在前一个单元(M_{i-1})处计算的校验和(CS)值的基础上形成；

-在计算和之前，将第二因子的位循环一个预定的数。

2.如权利要求1所述的方法，其特征在于，

-第二因子本身是校验和的先前值，

-第一因子通过向右循环单元(M_i)的位K位(向最低位的方向)形成，其中

$$k = \sum_{j=0}^{\infty} INT(i / M^j)$$

其中INT表示一个整数，保证循环位数K的正确。

3.如权利要求1所述的方法，其特征在于在方法中

-第一因子本身是单元(M_i)；以及

-第二因子通过循环校验和(CS)的先前值形成，循环是由

-在保证0 < Mⁿ < i+1的每个M的整数次幂Mⁿ处，确定单元(M_i)的序号加1(i+1)是否可以被该M的整数次幂Mⁿ整除；以及

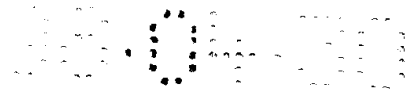
-响应单元(M_i)序号加1(i+1)被该M的整数次幂Mⁿ整除，向左循环校验和(CS)一位；以及

-一旦存储区域(MA)的所有的存储单元(M_i)都进行一遍，向右循环校验和(CS)，循环位数与计算校验和(CS)的过程中向左循环的总位数相同。

4.如权利要求1所述的方法，用于根据存储区域(MA)的单元(M_i)中的变化更新校验和(CS)，其特征在于该方法包括以下步骤：

-将变化的单元(M_i)的新数据与该同一单元(M_i)的旧数据逐位求和；

-在变化的单元(M_i)的序号(i)的基础上，推断，在计算原始校验和(CS)时该单元(M_i)共循环了多少位；



-向右循环校验和，循环位数与计算原始校验和时在该单元(M_i)向左循环的位数相同；

-对循环的校验和(CS)求和，由此获得原始校验和(CS)。

5.如上述任一权利要求所述的方法，其特征在于该方法进一步包括以下步骤：

-在至少一个存储区域计算一个校验和(CS)；

-比较在该存储区域(MA)计算的校验和(CS)和在以前阶段同一存储区域(MA)计算的校验和；

-响应在同一存储区域(MA)不同时刻计算的校验和之间的失配，产生指示该差别的一个信号。

6.如权利要求 5 所述的方法，其特征在于，在该存储区域(MA)计算的校验和与在以前阶段同一存储区域(MA)计算的校验和(CS)以预定的次数周期性地进行比较。

7.一种方法，用于包括一个盘驱动器的计算机的快速重新启动，其特征在于以下步骤：

-在至少一个存储区域计算一个校验和(CS)；

-重新启动计算机；

-在至少一个这样的存储区域(MA)计算一个校验和(CS)，在重新启动之前已经在该存储区域计算了一个校验和(CS)；

-比较重新启动后计算的校验和(CS)和重新启动之前在同一存储区域(MA)计算的校验和(CS)；

-响应重新启动后计算的校验和与重新启动之前计算的校验和之间的失配，从盘驱动器中装载与该存储区域对应的数据。

8.如权利要求 7 所述的方法，其特征在于，重新启动之前计算的校验和(CS)的值存储在一个主存储器中。

9.如权利要求 8 所述的方法，其特征在于，重新启动之前计算的校验和(CS)的值存储在一个写保护的主存储器中。

10.如权利要求 7 所述的方法，其特征在于，重新启动之前计算的校验和(CS)的值存储在一个硬盘中。



说明书

检测计算机中的存储器问题

本发明涉及检查计算机中存储器内容的完整。

图 1 显示一台计算机中与本发明有关的基本的部分。在初始阶段，在电源开启之后，主存储器 RAM 的内容处于一个不明确的状态。常规计算机的初始装载发生，使得电源开启产生一个初始信号 INIT，它使得，例如，一个处理器 CPU 跳进一个预定的地址。在一个只读存储器 ROM 中，这个存储器具有一个初始装载的程序，它的执行包括从一个 HD 磁盘驱动器中的一个预定位置装载操作系统（OS）的基本部分。这样装载的 OS 基本部分包括更多高级的装载程序，不仅从 ROM 中的为 HD 预定的位置装载一部分 OS，还从其它地方也装载一部分 OS。这些更高级的装载程序依次装载其它软件模块等。

这些软件模块不必位于 HD 磁盘驱动器的相邻区域。在读 HD 时，通常情况是位于 HD 不同区域的信息必须接连读取。一个建议的指标值为可能从磁盘驱动器以约 1MB/s 的速度读相邻区域的数据。但实际上读的速度只是这个速度的一半。例如，装载 30MB 的软件可能花大约一分钟的时间。

在控制计算机的过程中，如开关交换，可以证明解决故障花费很大。解决软件缺陷尤其困难。硬件缺陷常常可以通过与另一个已知处于正常工作状态的仪器比较而发现。软件缺陷通常无法用这种方式发现，因为通常不存在已知处于正常工作的参考软件。多数情况下，通常遇到的进一步的缺点是根据征兆通常不可能迅速推断出缺陷是在硬件还是软件。在特定存储区域的未受控制的变化，可能由两种原因中的任意一个造成。如果问题发生，重要的是尽快直接维修正确的目标。

破坏者和“黑客”构成了威胁。如果用于远程维修的选项合并于计算机中，那么误用的危险将相应地增加。

为了发现存储器问题，已经开发了几个不同的解决方案。现有技术已知一个所谓的校验位，依靠它可能在存储位置有偶数个一位（或者，如果



需要，奇数个)。正如众所周知，通过在存储位置的偶数个位，使用校验不能发现状态的同时变化。

另一个一般使用的工具是循环冗余校验 CRC。CRC 和是发现串行模式数据，如位于磁盘驱动器中和数据链的数据，中的错误的一个有用的工具。关于矩阵数据，CRC 和面临两个问题。第一是 CRC 和不能用于可靠地发现在由计算 CRC 和的方法所引起的间隔时发生的缺陷。第二个问题与如果仅一个存储位置的内容改变就必须重新计算 CRC 和这个事实有关。

由以上可得出结论，开发一种方法和仪器，利用它们，计算机的存储器问题能被发现并且上述的问题能够被解决，这是本发明的一个目的。一方面，本发明是基于一种方法，根据最新收集的经验，该方法对于发现存储器中所有可能的未受控制的变化是足够广泛的，而且另一方面，本发明是基于在计算机的各种操作中均可使用本方法。

首先，根据本发明的方法和系统有可靠地发现存储器问题的优点。这因此导致第二个优点，即通过省略在现有技术系统中执行的“仅仅是以防万一”的一些指令，加速计算机的重新启动。

以下，将参考附图，利用本发明首选的实施例更详尽地描述本发明，其中：

图 1 显示一台计算机中与本发明有关的基本的部分；

图 2 说明使用校验和监视存储器内容；

图 3 是说明保持校验和的一个流程图；以及

图 4 说明根据本发明计算校验和。

关于图 2 的流程图，该校验和在一个特定存储区域计算，它与先前计算的校验和比较，如果这两个校验和不同，就会发出一条错误消息。在图 2 的流程图中，重要的问题是正在计算的校验和被初始化的方法。根据本发明的一个实施例，这种计算发生在有规律的间隔。根据另一个实施例，是结合特定的事件，例如，打开计算机、修改程序等，来计算校验和。

图 3 的流程图说明写入存储器是如何发生的。一旦一个存储区域的内容变化，将通过 (i) 从先前的校验和中删除变化的存储区域的部分，以及 (ii) 向这样变化了的校验和增加与变化了的存储区域的新内容对应的部分，来修正校验和。



图 3 中的流程图的简单使得计算校验和的现有技术的方法无法用于从一个计算过的校验和中删除变化的存储区域的先前部分，并且用一个与该相同存储区域的新内容对应的部分来代替。下面，参考图 4，描述用于计算校验和的根据本发明的一种新方法。将在计算机的整个存储器 MEM 中的被监视的一个存储区域 MA 内计算校验和。校验和 (CS) 的宽度 M 是任意的，但是宽度 M 的合适值等于计算机的字长 W。在计算之前，给校验和一个初始值，例如 0。进行计算，使得在一个存储区域 MA 的具有 M 位的每个单元 M_i (其中 i 表示单元在存储区域内的序号) 与校验和 CS 逐位求和 (通过一个异或运算)，使得在求和之前，单元 M_i 向右循环 k 位 (向最低位的方向)。循环的位数 k 由下式获得：

$$k = \sum_{j=0}^{\infty} INT(i/M^j) \quad (1)$$

其中 INT 表示一个整数。实际上，求和不必永远继续，只要达到当 M^j 小于或等于单元序号 i 时的 j 的最大值 (或可能的最大的序号值)。

利用以下的快速方法能够得到同样的结果。存储区域 MA 的单元与校验和 CS 进行逐位求和 (异或)。每次求和以后，校验和向左 (向最高位的方向) 循环一位。如果 $i+1$ 被 M 整除则重复循环，然后如果 $i+1$ 被 M^2 整除则同样循环，等等，直到 M 的所有可能的整数次幂都进行一遍。一旦所有的单元都求和完毕，校验和向右循环，位数与求和时校验和向左循环的位数相同。

本方法值得一提的优点是这样计算的校验和可能容易被更新。这种更新如以下发生：

- (1) 变化的单元的数据与该单元的旧数据求和。
- (2) 根据变化的单元的序号，推断，当计算原始校验和时在该单元共循环了多少位。
- (3) 将和向右循环相同程度。
- (4) 将这样得到的循环的和与原始的校验和求和。

这样得到的结果是根据变化的存储区域更新的新的校验和。如果一个具有任意尺寸的存储区域发生变化，将通过每个单元仅计算两个和来更新



校验和。这样，新的校验和不必在整个存储区域计算，那将是循环冗余校验（CRC）的事。

通过单元序号 i 除以 M 的可能的整数次幂，然后根据方程（1）对这些商取整求和，可以计算出在变化的单元循环的位数。

参考表 1，现在将描述根据本发明的方法的校验和的一个示范计算。为简化起见，假定存储区域包括 16 个能够通过一个 4 位序号寻址的位置。在本例中，存储位置的数据也具有 4 位的宽度。同样，校验和的初始值置为 0。作所有这些假定都是为了说明本发明，它们并不以任何方式限定本发明。

存储位置 0 的内容 1001 与初始值 0000 逐位求和，然后因为 $0+1$ 不能被校验和的宽度（4）整除，所以和向左循环一位。得到的新的校验和是 0011。在这个和上加上存储位置 1 的内容 1011，这样得到的和向左循环一位，因为 $1+1$ 不能被校验和的宽度整除。这样以后，新的校验和为 0001。这个过程以这种方式继续，直到整个存储区域进行一遍。在存储位置 3、7 和 11，校验和循环两位，因为这些序号加 1 后能够被校验和的宽度整除。在存储位置 15，将循环三位，因为 $15+1$ 既可以被校验和的宽度整除，也可以被校验和的宽度的平方整除。在所有其它位置，校验和仅循环一位。最后，校验和向相反方向，即向右循环 21 位。最终得到的校验和为 0111。计算的逐步过程如表 1 所示。



表 1

地址	数据	旧 CS	位和	新 CS	循环
0000	1001	0000	1001	0011	1
0001	1011	0011	1000	0001	1
0010	1101	0001	1100	1001	1
0011	0010	1001	1011	1110	2
0100	1011	1110	0101	1010	1
0101	1010	1010	0000	0000	1
0110	0111	0000	0111	1110	1
0111	0010	1110	1100	0011	2
1000	0000	0011	0011	0110	1
1001	0001	0110	0111	1110	1
1010	0111	1110	1001	0011	1
1011	0101	0011	0110	1001	2
1100	1111	1001	0110	1100	1
1101	0011	1100	1111	1111	1
1110	1110	1111	0001	0010	1
1111	0101	0010	0111	1011	3
		1011		1101	-21

下面，假定存储位置例如 0101 的内容从 1010 变为 0110。新旧内容的逐位求和将为 1100。根据序号 0101 (=5)，逐位求和 1100 向右循环 $5+1=6$ 位，由此得到的结果为 0011。当它加到旧的校验和上时，得到的新的校验和为 1110。通过从开始就重复结合表 1 讨论的步骤，可以得到相同的结果，但是存储位置 0101 的内容具有新的值，在这种情况下为 0110。

校验和可以通过计算机方便地计算。为此，可以利用以下类似 Pascal 语言的伪代码形式的程序。



```
SUM:=initial value;  
for I:=0 to N-1 do  
SUM:=SUM xor X(I);  
SUM:=SUM rol 1;  
if (I+1) mod M = 0 then  
SUM:=SUM rol 1;  
if (I+1) mod (M*M) = 0 then  
SUM:=SUM rol 1;  
if (I+1) mod (M*M*M) = 0 then  
SUM:=SUM rol 1;  
...  
end;  
end;  
end;  
end;  
end;  
end;  
N1:=N+N/M+N/(M*M)+N/(M*M*M)+N...;  
SUM:=SUM ror N1;  
end;
```

在上述伪代码中，N 表示被监视的存储区域的尺寸，xor 表示逐位求和（异或），mod 表示模运算符（余数），rol 表示左循环，ror 对应地表示右循环。

以下伪代码可以用于更新校验和：

```
SUM:=initial value;  
for I:=0 to N-1 do  
I1:=I+I/M+I/(M*M)+I/(M*M*M)+I...;  
SUM:=SUM xor (X(I) ror I1);  
end;
```



显然，没有一个校验和可以发现所有的存储器问题。一个显而易见的原因是：因为校验和 CS 的信息内容比计算校验和时所处的存储位置 MA 的信息内容小得多，所以存储位置不可避免地具有一些产生相同校验和 CS 的可能内容。然而，实际上，缺陷能被可靠地发现。该存储位置有一个未受控制的变化，而校验和无并发变化将要求存储区域的几个地址的几个位根据一个特定的数学规律性同时变化。尽管实际的存储器问题通常确实表现数学的规律性，但是这个规律性没有这种性质，即使用本发明的方法，存储器问题将一直不被发现。在这个方面，可以说根据本发明计算校验和是“有规律地无规律”。

用于计算校验和的本发明的方法可以如下被应用：一个有贡献的校验和被分配到被监视的每个存储区域。随着存储区域的内容的改变，该校验和也相应改变。校验和的完整性被例如在特定间隔或结合特定事件的时刻监视。如果发现校验和不匹配，缺陷至少被报告并且其它要求的测量加之进行。

利用本发明的校验和计算方法的第二个方面是加速计算机的重新启动。常规的计算机在这种情况下从一个磁盘驱动器，一个网络等装载整个软件。通过利用根据本发明的校验和，可能指示哪个存储区域关于其内容已经不存在缺陷，导致其内容不必重新装载。结果，与现有技术相比，计算机的启动可以加速几十分钟。在重新启动之前，在被监视的一个或多个存储区域 MA 可以计算校验和 CS，并且这些校验和可以存储在主存储器中。一些系统允许保护一部分存储器以后不被覆盖。在这种类型的存储器中，相应于不同程序模块的校验和可以被存储，因为一旦该程序模块被装载进一个存储器，其内容就不应被改变。或者，一个预定的校验和可以与存储在磁盘驱动器的每个程序模块结合。在装载程序模块的过程中，该校验和可以首先从磁盘驱动器读出，并与在相应的存储区域计算的校验和比较。如果校验和一致，程序模块将不必从磁盘驱动器重新装载。包括多个部分的软件的不同模块的校验和可以存在一个共有的文件中，读该文件可迅速区分哪些模块已经无缺陷地启动以及哪些模块应从磁盘驱动器重新装载。

尽管本发明是以计算机为例解释的，但本发明并不局限于计算机的一



个狭义范围，而可以应用于使用电子存储器的所有情况。这种应用的例子包括计算机外围设备和通讯系统、图象和声音存储器、数据记录系统、控制和处理自动控制系统以及各种机器人。同样显然，对于熟练的技术人员，随着技术的进步，本发明的基本原理可以以不同方式应用。本发明以及其中的实施例并不局限于上述的例子，而是在权利要求书的范围内可以变化

。

说明书附图

图 1

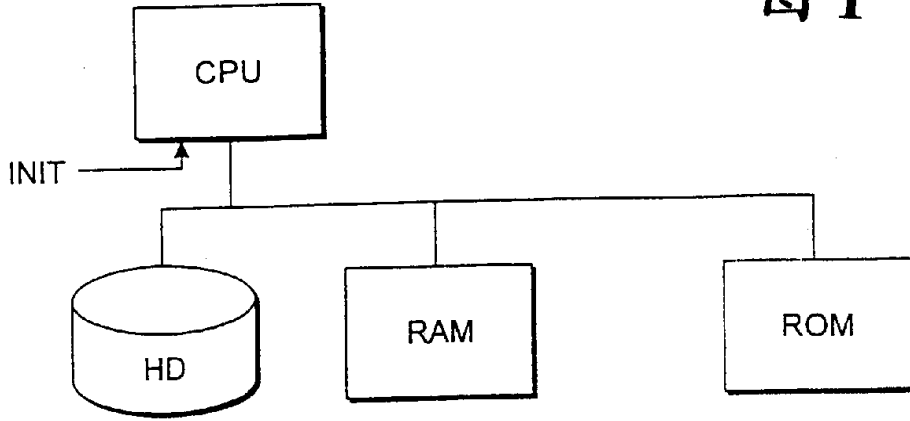


图 4

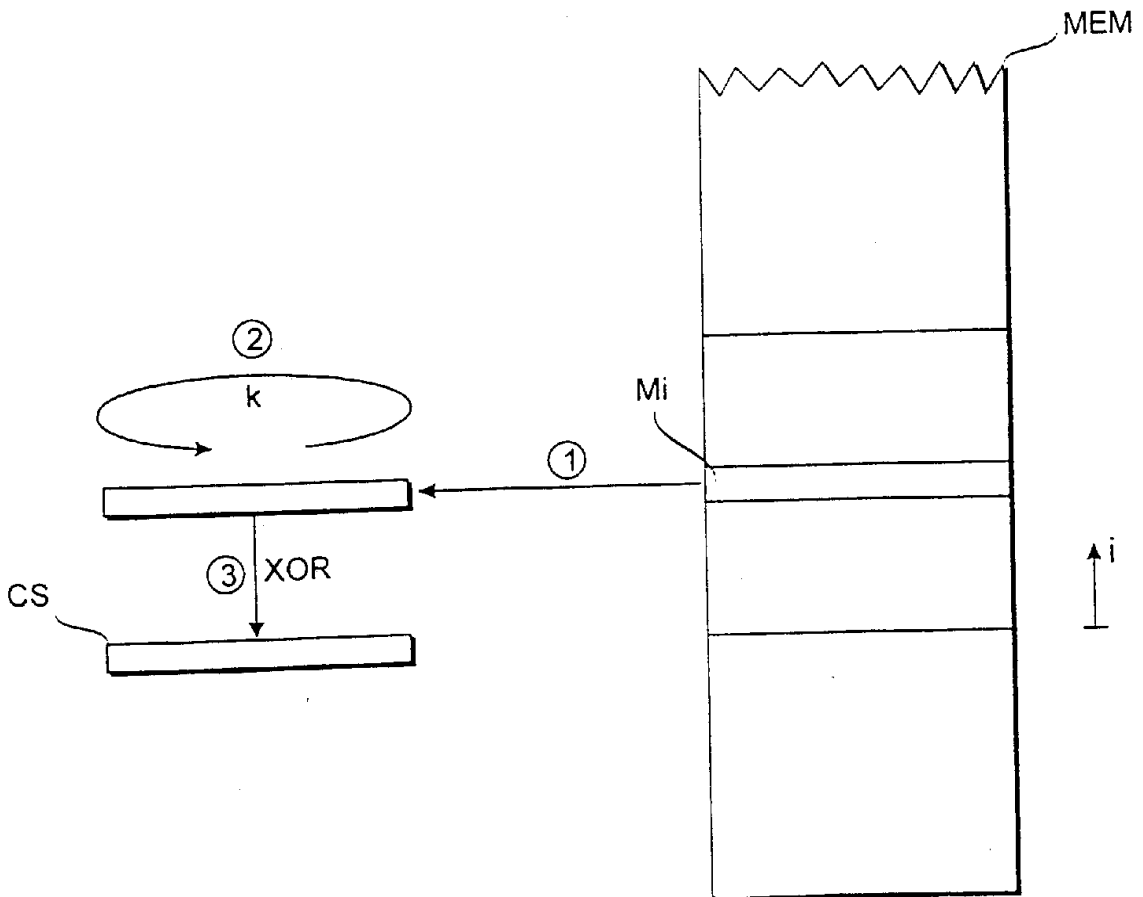




图 2

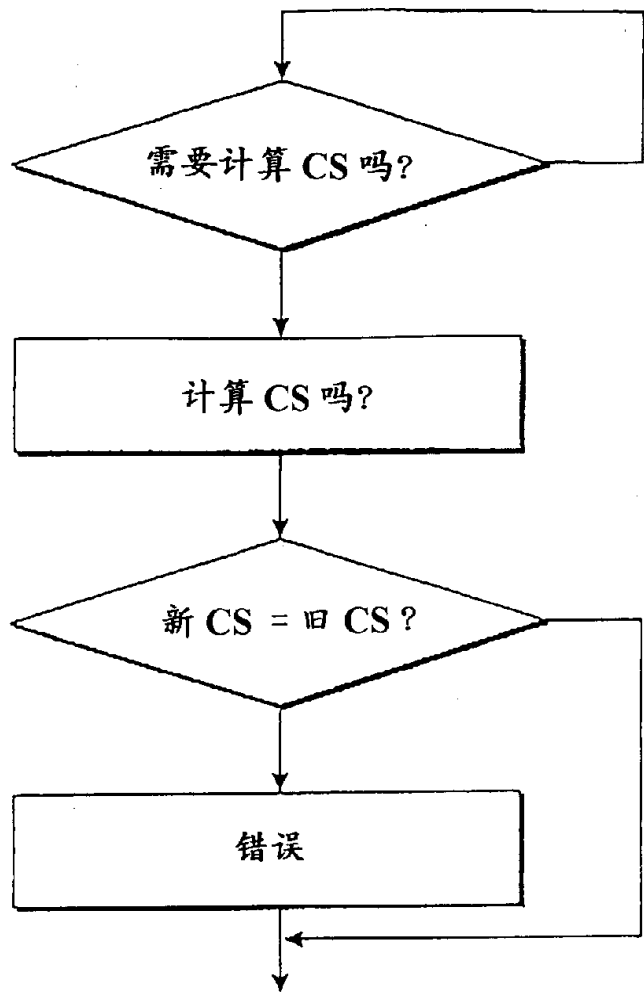


图 3

