[19] 中华人民共和国国家知识产权局

[51] Int. Cl[7]
G06F 11/00

[12] 发明专利说明书

[21] ZL 专利号 00806634.5

[45] 授权公告日 2004 年 10 月 27 日

[11] 授权公告号 CN 1173264C

[22] 申请日 2000.4.19 [21] 申请号 00806634.5
[30] 优先权
[32] 1999. 4.23 [33] DE [31] 19918620.0
[86] 国际申请 PCT/EP2000/003530 2000.4.19
[87] 国际公布 WO2000/065442 德 2000.11.2
[85] 进入国家阶段日期 2001.10.23
[71] 专利权人 德国捷德有限公司
地址 德国慕尼黑
[72] 发明人 迈克尔·鲍迪施韦勒
审查员 龙 安

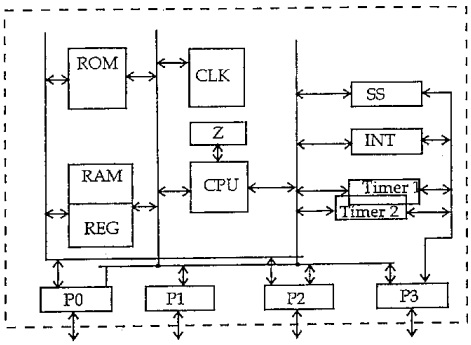
[74] 专利代理机构 北京市柳沈律师事务所
代理人 黄小临 王志森

权利要求书 2 页 说明书 4 页 附图 1 页

[54] 发明名称 保护计算机的核心免受外部窜改的方法

[57] 摘要

本发明涉及一种方法，用于保护计算机的中央处理器，尤其是芯片卡。 在 CPU 执行一条指令之后，多个单独的敏感的寄存器被逻辑组合以形成一个校验和。 所述的校验和被存储并用在下一条指令开始处理之前适当形成的校验和进行比较。 如果被比较的校验和不能匹配，这表示在两条被处理的指令之间的时间期间 CPU 的寄存器内容的窜改。在这样一个情形下，对应的错误消息被给出而且处理器停止或卡被没收。




ISSN1008-4274

1.一种用于保护带有中央处理器(CPU)的计算机避免外部的窜改的方法，其特征在于最后校验和通过根据在 CPU 的指令执行结束时产生的 CPU 的寄存器内容而形成并被存储，并且初始校验和与在 CPU 的下一条指令执行的开始时产生的寄存器内容相关地形成，如果初始校验和与最后校验和不匹配，则产生错误消息。

2. 如权利要求 1 所述的方法，其特征在于在装入指令时，计数器被启动用于对执行该指令必要的时钟周期计数并当预定时钟周期超出或不足时输出一个错误消息。

3. 如权利要求 2 所述的方法，其特征在于该错误信号触发一个中断或导致时钟信号供应的终止。

4. 如权利要求 2 所述的方法，其特征在于执行一指令必需的时钟周期的数量被通过一个逻辑电路从指令操作码获得。

5. 如权利要求 1 所述的方法，其特征在于所述数学组合通过对寄存器内容的异或组合而进行。

6. 如权利要求 1 至 5 中任一项所述的方法，其特征在于该方法的启动由随机或预定义的事件所触发。

7. 如权利要求 6 所述的方法，其特征在于该方法被以依赖于时间的形式触发。

8. 如权利要求 6 所述的方法，其特征在于该方法当 CPU 的一个或多个寄存器的内容对应于预定码型时触发。

9. 如权利要求 6 所述的方法，其特征在于该方法在每个情况下预定数量的指令被处理后被触发。

10. 一种计算机用的中央处理器（CPU），用于执行按照权利要求 1 到 6 的任一个的方法，包括:

- 通过逻辑元件的 CPU 的几个寄存器的组合以形成一个校验和，
- 一个校验和存储器，用于存储逻辑元件形成的第一校验和，
- 一个比较器，用于比较逻辑元件形成的第二校验和与存储在存储器中的第一校验和，
- 一个控制器件，用于控制在校验和存储器中的第一校验和的存储并用

于控制比较器。

11. 按照权利要求 10 的一个中央处理器，其特征在于包括一个计数器，用于对指令执行要求的时钟周期进行计数。

12. 按照权利要求 10 或 11 的中央处理器，其特征在于包括一个逻辑电路，用于从指令的操作码确定执行指令必要的时钟周期。

13. 一个包括按照权利要求 10 或 12 的中央处理器的计算机。

14. 一个包括按照权利要求 10 或 12 的任一个的中央处理器的智能卡。

# 保护计算机的核心免受外部窜改的方法

## 技术领域

本发明涉及保护计算机免受外部窜改(manipulation)，特别是在计算机核心或中央处理器（CPU）中的数据的保护。本发明特别应用于智能卡，因为它们必须要特别被保护免受外部的窜改。

## 背景技术

众所周知，通过诸如总线加密、存储区加密等方法保护计算机的存储区免受窜改。DE 37 09 524 C2 公开了例如一个测试例程，用于检查一个程序存储器的存储单元的内容。通过在程序运行开始或程序运行过程中形成对存储单元内容的一个校验和并将它与以前存储在程序存储器中的校验和相比较，人们可以检测在原始存储单元内容中的变化以及仅在操作过程中发生的变化，这导致了错误消息。

## 发明内容

本发明的问题是提出一个更好保护计算机免受外部窜改的方法。

此问题的解决是按照本发明通过根据独立权利要求的特征的一个方法、一个用于执行所述方法的中央处理器、一个带有这样的中央处理器的计算机和智能卡来进行。

按照本发明一个方面，提供了一种用于保护带有中央处理器(CPU)的计算机避免外部的窜改的方法，其特征在于最后校验和通过根据在 CPU 的指令执行结束时产生的 CPU 的寄存器内容而形成并被存储，并且初始校验和与在 CPU 的下一条指令执行的开始时产生的寄存器内容相关地形成，如果初始校验和与最后校验和不匹配，则产生错误消息。

按照本发明另一个方面，提供了一种计算机用的中央处理器（CPU），用于执行上述方法，该中央处理器包括：通过逻辑元件的 CPU 的几个寄存器的组合以形成一个校验和；一个校验和存储器，用于存储逻辑元件形成的第一校验和；一个比较器，用于比较逻辑元件形成的第二校验和与存储在存储器

中的第一校验和；一个控制器件，用于控制在校验和存储器中的第一校验和的存储并用于控制比较器。

本发明开始于通过保护计算机核心、即计算机的中央处理器中出现的数据免受外部窜改来增加计算机的安全性的想法，因为存在于计算机核心中的数据是未加密的形式并因此易于被窜改。

为了识别这种窜改，人们在一个指令已经被CPU处理后，通过数学组合，例如通过异或操作（XOR操作），来从计算机的几个寄存器内容确定校验和，并将其存储在存储器中作为最后的校验和。在下一个指令被计算机处理之前，又形成一个校验和，即初始的校验和。通过比较初始校验和和最后校验和——它们必须相匹配，人们可以确定是否CPU的寄存器内容在上一个指令被处理之后被窜改。人们可以使用CPU的可以取非零状态的那些区域的内容作为寄存器内容，如在8051型处理器中的累加器(accu)、B-累加器、数据指针（DPTR、DPL、DPH）、寄存器组的寄存器（R0 至 R7）程序状态字（PSW）、栈指针（SP）、专用功能寄存器等。

为了进一步增加安全性，人们可以当装载一条指令时，额外启动一个计数器用于对执行指令必要的时钟周期进行计数。该计数器最好是以硬件形式构成。一逻辑单元从指令操作码导出执行必需的时钟周期数并将其转换成计数器值。该计数器随后与被执行指令并行运行。检查是否要执行的该指令在规定的时钟周期内被执行。如果该指令在预定时间周期内未被执行，例如使时钟的提供被停止以便指令的进一步执行不再可能。作为另一选择，复位可以被触发，中央处理器因此复位。如果该指令过早地被执行，即指令计数器的限定值还没有被达到而一个新操作码已经被识别时，也可以采取同样的步骤。

安全相关的寄存器的逻辑组合可以通过硬件或软件实现。在两个连续指令之间的校验和的形成可以基于在例如随机或预定义的事件来实现或一直实现。

附图说明

下面参照附图详细说明本发明，其中：

图1示意了通过8051处理器的示例而给出的微控制器的结构；

图2示意了用于组合中央处理器的几个区域的逻辑。

具体实施方式

图 1 示意了一个 8051 处理器的结构，即一个 8 位处理器。当数据被通过已知的加密方法中的总线加密或存储器加密保护免受窜改的同时，数据以未加密的形式存在于计算机的核心，即中央处理器或 CPU 中。本发明的方法现在判定是否 CPU 的一个或多个寄存器已经被窜改。

图 2 通过示例示意了可能被窜改的 CPU 的安全相关区域，即栈指针（SP）、累加器 AC、B-累加器 BAC、寄存器 R0 至 R7、用于内部 RAM 的上下区域的数据指针 DPL 和 DPH。所述的寄存器被逻辑组合以形成一个校验和。在图 2 中，两个 8 位寄存器在每种情况下通过一个异或门（XOR）被组合。因此，寄存器 R0 和 R2 的异或产生了一个新的 8 位码型(pattern)，它又与从寄存器 R1 和 R7 的异或产生的 8 位码型相异或。所产生的 8 位码型的异或最后产生了一个 8 位的码型作为一个校验和并被指定为图 2 中的“初始校验和”。不通过异或，人们当然可以选择其他实施例以形成校验和，尽管采用异或在这方面是有好处的。

如果寄存器的组合被通过逻辑元件以硬件形式执行，当寄存器的内容改变时，校验和立刻改变。即在 CPU 中处理的一条指令的执行过程中校验和可能多次改变。然而，对于执行该方法关键的仅有的校验和是指令执行以后的那一个和下一条指令执行之前的那一个，因为这两个校验和（一条指令的最后校验和和下一指令的初始校验和）在一比较器中比较。

比较执行如下。在第一条指令执行结尾产生的校验和被存储在 CPU 的一个存储器中作为最后校验和。为了确定对 CPU 的窜改是否在所述第一指令执行后和在下一条、第二条指令装入 CPU 前已经发生，初始校验和如上所述地与所述第二条指令的装载并行地形成。在第一步骤 a)中，初始校验和通过一个比较器与来自以前被执行的第一指令的存储在存储器中的最后校验和相比较。如在 CPU 上没有执行窜改，初始和最后的校验和相匹配而且比较的结果值是 0。比较器输出一个信号，在此基础上，当前可用的校验和被存储在存储器中作为第二条指令执行后的步骤 b)中的新的最后校验和。即，第二条指令的执行在此情形下未被中断。然而，如果初始校验和和最后校验和的比较产生了一个非零值，可推断一定有对 CPU 的窜改。比较器的输出随后不是进到第二步骤 b)而是进到错误消息 c)，它在图 2 所示的情形下引起指令处理的

终止。例如，处理器可以被停止、安全传感器被激活或在智能卡的情况下智能卡被终端扣留。

上述安全机制也可以以软件形式被严格实现，这是通过一方面，在一条指令执行的结尾确定校验和，另一方面，在下一条指令执行的开始处确定校验和，并比较它们。对应的程序可以被存储在例如处理器的 ROM 或 EPROM 中，最后的校验和被存储在以位寻址的处理器的 RAM 中。

所述的方法不需要在要执行的每条指令之前被使用。本发明的一个实施例提供了该方法的执行依赖于随机或预定义的事件。按照第一实施例，该方法可以以依赖时间的形式被触发。

按照本发明的另一个实施例，该方法可以通过与预定码型相对应的 CPU 的一个或多个寄存器的内容被触发。

本发明的又一个实施例提供了在每个情况下在预定数量的指令处理后要被触发的方法。

一个优选实施例是这样的，该方法只在下列情况被触发，即如果在一个指令与下一个指令的开始执行的初始校验和之间有一个较长的、预定义的时间间隔的情况，所述指令执行后的校验和被存储在存储器中作为最后检验和。这节省了在带有多条指令的程序的执行中宝贵的计算机的运算能力。假设 CPU 的窜改，尤其是对于智能卡，在程序运行中并未发生，但是在智能卡被从智能卡终端拿出时发生，对 CPU 的窜改仍然可以通过该后一种实施例的方法可靠地检测。

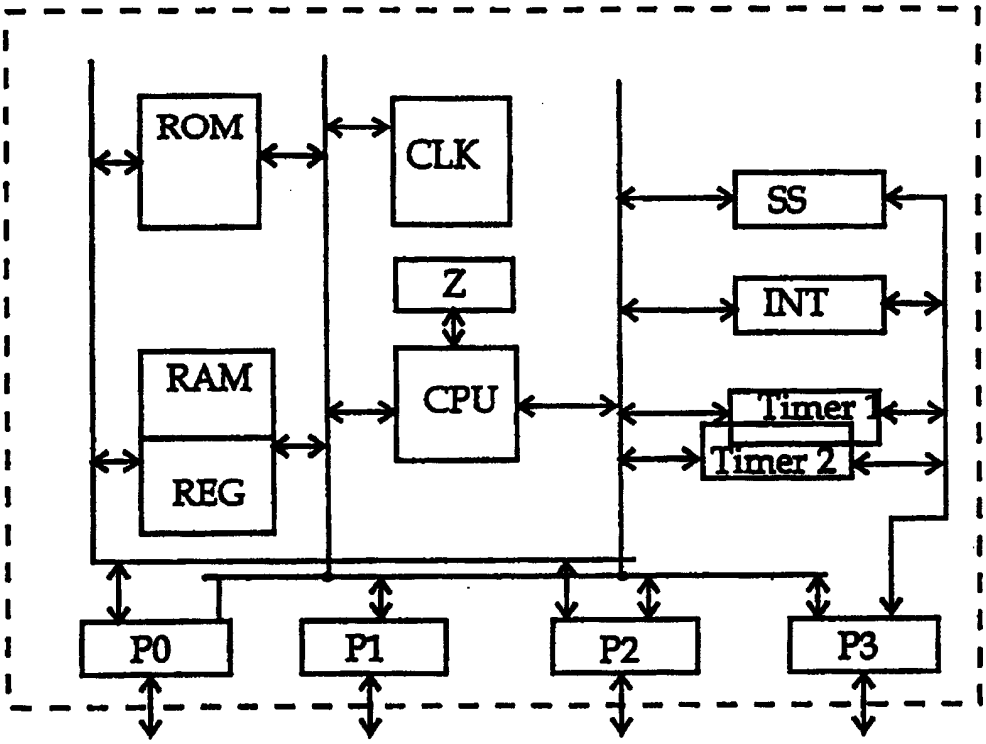

ROM
CLK
SS
Z
INT
RAM
CPU
Timer 1
Timer 2
REG
P0
P1
P2
P3

图 1

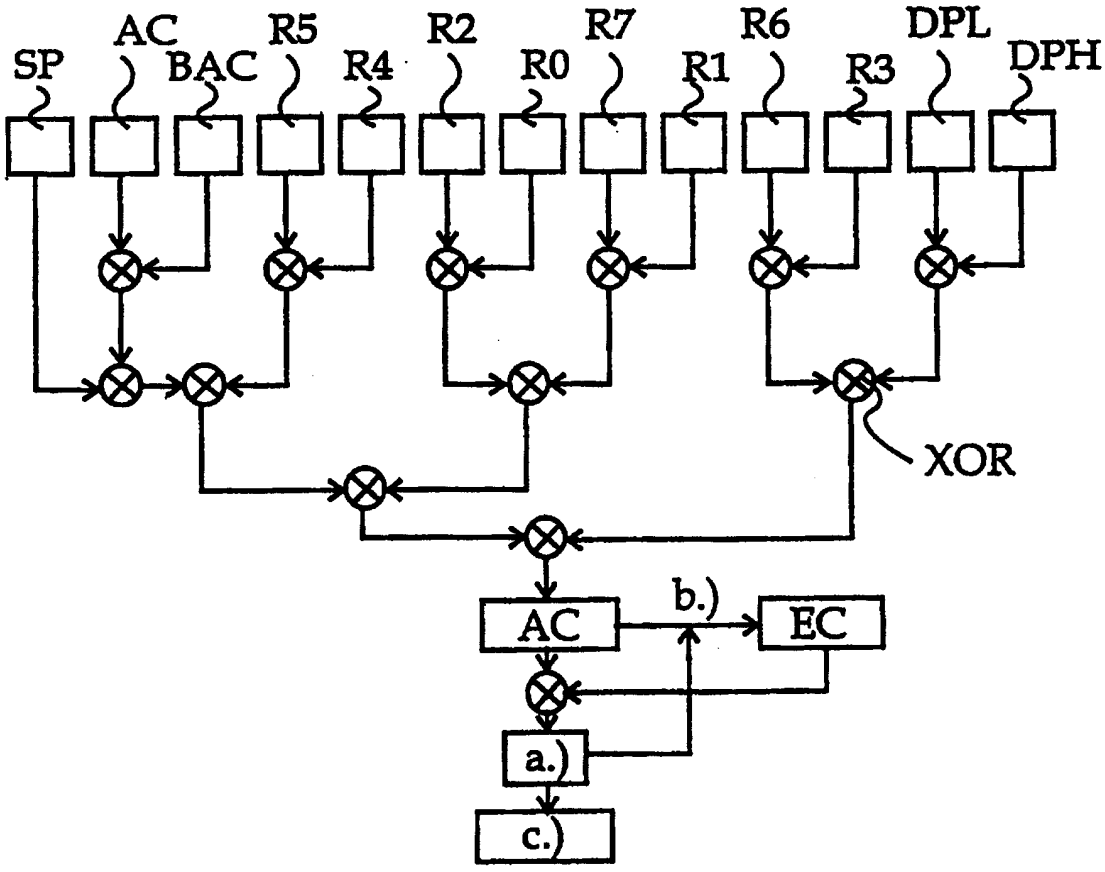

SP
AC
BAC
R5
R4
R2
R0
R7
R1
R6
R3
DPL
DPH
XOR
AC
b.)
EC
a.)
c.)

图 2