



(10) **AT 513533 A2 2014-05-15**

(12) **Österreichische Patentanmeldung**

(21) Anmeldenummer: A 791/2013  
(22) Anmeldetag: 14.10.2013  
(43) Veröffentlicht am: 15.05.2014

(51) Int. Cl.: **G06F 11/10** (2006.01)

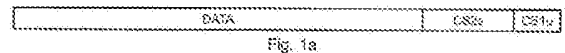
(30) **Priorität:**  
18.10.2012 DE 102012020442.6 beansprucht.

(71) **Patentanmelder:**  
ROBERT BOSCH GMBH  
70442 STUTTGART (DE)

(74) **Vertreter:**  
PATENTANWÄLTE PUCHBERGER, BERGER  
& PARTNER  
WIEN

(54) **Verfahren zum Überprüfen von Daten mittels wenigstens zweier Prüfsummen**

(57) Die Erfindung betrifft ein Verfahren zum Überprüfen von Daten in einer Recheneinheit zur Laufzeit eines Programms, wobei die Daten (DATA) mit einer Prüfsumme erster Art (CS2s) und mit einer Prüfsumme zweiter Art (CS1u) überprüft werden, indem die Prüfsumme erster Art und die Prüfsumme zweiter Art zur Laufzeit des Programms berechnet und mit jeweils mit einer gespeicherten Prüfsumme (CS1u, CS2s) verglichen werden, wobei zur Laufzeit des Programms der Rechenzeitbedarf zum Berechnen der Prüfsumme zweiter Art (CS1u) geringer als der Rechenzeitbedarf zum Berechnen der Prüfsumme erster Art (CS2s) ist.



AT 513533 A2 2014-05-15

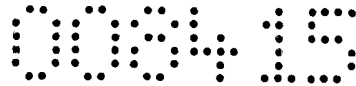
Zusammenfassung

5

Die Erfindung betrifft ein Verfahren zum Überprüfen von Daten in einer Recheneinheit zur Laufzeit eines Programms, wobei die Daten (DATA) mit einer Prüfsumme erster Art (CS2s) und mit einer Prüfsumme zweiter Art (CS1u) überprüft werden, indem die Prüfsumme erster Art und die Prüfsumme zweiter Art zur Laufzeit des Programms berechnet und mit jeweils

10 mit einer gespeicherten Prüfsumme (CS1u, CS2s) verglichen werden, wobei zur Laufzeit des Programms der Rechenzeitbedarf zum Berechnen der Prüfsumme zweiter Art (CS1u) geringer als der Rechenzeitbedarf zum Berechnen der Prüfsumme erster Art (CS2s) ist.

15 (Figur 1a)



Robert Bosch GmbH, Stuttgart

5

## **Verfahren zum Überprüfen von Daten mittels wenigstens zweier Prüfsummen**

### Beschreibung

10

Die vorliegende Erfindung betrifft ein Verfahren zum Überprüfen von Daten mittels wenigstens zweier Prüfsummen sowie eine Recheneinheit zu dessen Durchführung.

### Stand der Technik

15

Die Erfindung befasst sich mit sicherheitsgerichteter Software, bei der Daten (insbesondere Programmcode und/oder andere statische (konstante) Daten) in einer Speichereinheit liegen. Die Datenintegrität dieser Daten muss dabei sichergestellt werden, um sicherzustellen, dass die Daten während der Laufzeit bzw. Lebenszeit des Gerätes nicht verändert werden.

20

Solche Anwendungen findet man beispielsweise bei Sicherheits-SPSen.

Es kann vorgesehen sein, die Daten zyklisch zu überprüfen. Hierbei soll ein bestimmter Diagnosedeckungsgrad "DC" pro Prüfungsdurchlauf erreicht werden und die Prüfungsdurchläufe finden innerhalb einer vorgegebenen Zeit statt (z.B. mindestens alle 8 Stunden). Die Sicherheit wird dabei dadurch erreicht, dass die zyklische Prüfung bei einer vorgegebenen Zykluszeit mit einer vorgegebenen Wahrscheinlichkeit aufgetretene Datenkorruption erkennt. Beispielsweise soll eine Prüfung alle 8 Stunden mit einer Diagnosedeckung von 99% erreicht werden. D.h. bei jedem Durchlauf soll mit 99%iger Wahrscheinlichkeit eine Datenkorruption erkannt werden (DC = 99%).

30

Die Datenintegritätsprüfung kann dabei mittels Redundanzinformation stattfinden. Dies kann in Geräten mit sicherheitsgerichteter Funktionalität durch Redundanzinformationen (beispielsweise Prüfsummen), die zu den zu sichernden Daten hinzugefügt werden, in Verbindung mit einer zyklischen Überprüfung erreicht werden. Dabei wird wiederum zyklisch (z.B. alle 8 Stunden) überprüft, ob eine berechnete Prüfsumme noch mit einer gespeicherten

35

Prüfsumme der Daten übereinstimmt. Ist dies der Fall, wird die sicherheitsgerichtete Funktion fortgesetzt, ist dies nicht der Fall, wird das System in einen sicheren Zustand versetzt. Damit soll z.B. verhindert werden, dass bei korruptem Programmcode Gefahr für Leib und Leben entstehen kann, indem z.B. Maschinenbewegungen ausgelöst werden, die nicht ausgelöst werden sollen.

Bei sicherheitsgerichteter Software muss die Prüfung der Datenintegrität immer erfolgen. Sie ist besonders dadurch notwendig, da der Programmcode bzw. statische Daten beim Start des sicherheitsgerichteten Gerätes meist aus einem nichtflüchtigen Speicher (z.B. Flash-Speicher) in einen flüchtigen Speicher (RAM) umkopiert wird und dieser Programmcode dann aus dem schnelleren RAM heraus ausgeführt bzw. mit den statischen Daten aus dem RAM heraus gearbeitet wird. Durch Fehler in Programmen kann dabei der flüchtige Speicher (RAM) korruptiert werden, d.h. es werden durch ungewollte RAM-Speicherveränderungen aufgrund von Softwarefehlern Programmcodeinformationen bzw. Werte der eigentlich statischen Daten verändert. Neben Programmcodeinformationen sind somit auch statische Daten, mit denen das Programm arbeitet (z.B. konstante Variablen) zu sichern.

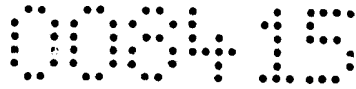
Insbesondere die zyklische Überprüfung bereitet jedoch Probleme, da sie zur Laufzeit der sicherheitsgerichteten Software durchgeführt wird und daher Rechenleistung der ausführenden Recheneinheit, insbesondere SPS, bzw. deren CPU beansprucht, die dann nicht für andere Funktionen zur Verfügung steht.

Es ist daher wünschenswert, die Überprüfung von Daten in einer Recheneinheit zu vereinfachen.

Offenbarung der Erfindung

Erfindungsgemäß wird ein Verfahren zum Überprüfen von Daten mittels wenigstens zweier Prüfsummen mit den Merkmalen des Patentanspruchs 1 vorgeschlagen. Vorteilhafte Ausgestaltungen sind Gegenstand der Unteransprüche sowie der nachfolgenden Beschreibung.

Die Erfindung basiert auf der Maßnahme, die zu sichernden Daten mit wenigstens zwei Prüfsummen unterschiedlicher Art abzusichern, wobei eine Prüfsumme einer ersten Art eine hohe Sicherheit bei ggf. auch erhöhtem Rechenleistungsbedarf und eine Prüfsumme zweiter Art einen geringen Rechenleistungsbedarf bei ggf. auch verringerter Sicherheit bietet. Die



Prüfsumme erster Art ist daher auf hohe Sicherheit ausgelegt, die Prüfsumme zweiter Art auf geringe Rechenleistung. Die Prüfsumme erster Art wird vorzugsweise bei der Initialisierung bzw. beim Start der Software verwendet, die Prüfsumme zweiter Art bei der zyklischen Überprüfung während der Laufzeit.

5

#### Vorteile der Erfindung

10 Bisher werden zur zyklischen Prüfung bei sicherheitsgerichteten Geräten aufwändige Prüfsummen (meist CRCs) auch bei großen Datenmengen verwendet, obwohl diese zur Erreichung eines geforderten Diagnosedeckungsgrades nicht notwendig sind. Dies liegt insbesondere an möglichen Forderungen (z.B. Hammingabstände deutlich größer 3) an die Sicherheit bei der Initialisierung bzw. beim Start der Software, die mit einfachen Prüfsummen nicht zu erfüllen sind. Die Erfindung verknüpft nun eine sehr sichere Prüfsummenprüfung beim Programmstart bzw. bei der Initialisierung und eine deutlich einfachere Prüfsummenprüfung (regelmäßig bzw. zyklisch) während des Betriebs.

15

Durch die Erfindung besitzt man den Vorteil, dass die erstmalige Datenintegritätsprüfung mit deutlich besserer Fehleraufdeckungswahrscheinlichkeit als bisher ausgeführt werden kann, da ein Rechenzeitbedarf zu diesem Zeitpunkt nicht relevant ist. Die spätere zyklische Datenintegritätsprüfung wird hingegen rechenzeitoptimal durchgeführt.

20

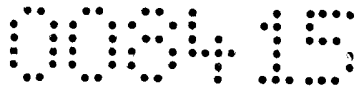
Vorzugsweise handelt es sich bei der Prüfsumme erster Art um eine CRC-Prüfsumme (zyklische Redundanzprüfung, englisch cyclic redundancy check). Eine CRC-Prüfsumme basiert auf einer Polynomdivision. Eine solche Prüfsumme kann eine höhere Fehlererkennung liefern, ist jedoch auch aufwändiger zu berechnen. Eine CRC-Prüfsumme besitzt die Eigenschaft, dass bei kürzeren Daten eine höhere Erkennbarkeit der Fehler vorhanden ist. Hierbei spricht man vom sog. Hammingabstand, dies ist die Mindestanzahl an Bits, in denen sich zwei Datenfelder (incl. CRC) unterscheiden und somit auch die Anzahl an beliebig verteilten Fehlern, die nicht mehr in jedem Falle erkannt werden können. Beispielsweise wird durch eine CRC-Prüfsumme nach IEC 802.3 (CRC32) bei großen Datenmengen eine Diagnosedeckung von ca. 99,99999998 % erreicht.

25

30

CRC-Prüfsummen besitzen die Eigenschaft, dass der Vorteil der höheren Erkennbarkeit von Fehlern bei kürzeren Daten bei Anwendung auf große Datenmengen nicht mehr gegeben ist und sie im Grenzfall nur eine minimal bessere Fehlererkennung besitzen als einfache Additionsprüfsummen.

35



Vorzugsweise handelt es sich bei der Prüfsumme zweiter Art um eine Additionsprüfsumme. Unter Additionsprüfsummen versteht man Prüfsummen, die durch die einfache Addition der Daten entstehen. Eine solche Prüfsumme ist sehr einfach zu berechnen und benötigt nur  
5 wenig Rechenkapazität. Man kann abschätzen, dass eine einfache Additionsprüfsumme ca. nur 10% des Rechenzeitbedarfs einer CRC32 besitzt. Mittels Additionsprüfsummen wird jeder Ein-Bit-Fehler erkannt. Zwei-Bit-Fehler werden nicht mehr sicher erkannt. Allgemein wird eine beliebige Datenverfälschung der Daten bei M Prüfsummenbits mit einer Wahrscheinlichkeit von  $1-1/2^M$  erkannt. Will man beispielsweise eine Diagnosedeckung von 99% erreichen,  
10 reicht dafür eine 7 Bit Additionsprüfsumme.

Vorzugsweise entspricht die Länge der Additionsprüfsumme der internen Speicherzugriffsdatenbreite oder der CPU-internen Registerbreite (z.B. 16, 32, 64... Bit) der die Software ausführenden CPU (z.B. Sicherheits-CPU). Sie ist dann zwar meist länger als durch die einzu-  
15 haltende Diagnosedeckung gefordert, jedoch kann der Speicherzugriff dadurch beschleunigt und der gesamte Rechenzeitbedarf der Prüfsummenberechnung optimiert werden.

Gemäß einer anderen Ausgestaltung handelt es sich bei der Prüfsumme zweiter Art um eine CRC-Prüfsumme, deren Berechnung jedoch weniger Rechenzeit benötigt als die Berechnung der Prüfsumme erster Art. Beispielsweise kann sie kürzer sein (z.B. nur 8 Bit). Gemäß  
20 einer weiteren bevorzugten Ausführungsform handelt es sich bei der Prüfsumme zweiter Art um eine 8Bit-CRC-Prüfsumme, die über einfache Tabellenzugriffe (Tabelle mit 256 Bytewerten) den jeweiligen Folge-Bytewert anhand des Datenbytewertes berechnet, beispielsweise nach dem Schema:

```
25     for (byte = 0; byte < nBytes; ++byte)
        {
            checksum = crcTable[data];
        }
```

30 Dem Fachmann ist aus dem Softwarebereich bekannt, wie die Tabelle (crcTable[]) bei der Initialisierung vorbesetzt werden muss.

Ein solches Verfahren ist besonders für kurze Prüfsummen geeignet, da die Tabelle  $2^M$  Einträge besitzen muss, wobei M die Anzahl der Prüfsummenbits ist.  
35

Im Vergleich dazu schneidet eine 16bittige oder 32bittige Additionsprüfsumme mit vergleichbarem Rechenaufwand wie eine 8bittige CRC bei langen Datenfeldern deutlich besser ab: Die Diagnoseabdeckung beträgt bei langen Datenfeldern ungefähr:

- 5
- bei einer 8Bit-Prüfsumme nur 99,6% (CRC- oder Additionsprüfsumme)
  - bei einer 16Bit-Prüfsumme 99,998% (CRC- oder Additionsprüfsumme)
  - bei einer 23Bit-Prüfsumme 99,99999998 % (CRC- oder Additionsprüfsumme)

10 Nimmt man an, dass die geforderte Diagnoseabdeckung der zyklischen Integritätsprüfung 99% betragen soll, reicht eine 7bittige (in der Praxis 8 Bit) Additionsprüfsumme aus. Aufgrund der typischen Organisation von Speichern bzw. Registern in Computern in Vielfachen von 8 Bit ist jedoch eine 8bittige Berechnung die kürzeste Prüfsumme, die normalerweise implementiert wird.

15 Vorzugsweise sind die Daten und die Prüfsumme zweiter Art durch die Prüfsumme erster Art gesichert. Hierdurch wird erreicht, dass bei Programmstart auch eine Korrumpierung der einfacheren Prüfsumme sicher erkannt wird.

20 Eine erfindungsgemäße Recheneinheit, z.B. eine SPS, ist, insbesondere programmtechnisch, dazu eingerichtet, ein erfindungsgemäßes Verfahren durchzuführen.

Auch die Implementierung der Erfindung in Form von Software ist vorteilhaft, da dies besonders geringe Kosten ermöglicht, insbesondere wenn eine ausführende Recheneinheit noch für weitere Aufgaben genutzt wird und daher ohnehin vorhanden ist. Geeignete Datenträger zur Bereitstellung des Computerprogramms sind insbesondere Disketten, Festplatten, Flash-Speicher, EEPROMs, CD-ROMs, DVDs u.a.m. Auch ein Download eines Programms über Computernetze (Internet, Intranet usw.) ist möglich.

25

Weitere Vorteile und Ausgestaltungen der Erfindung ergeben sich aus der Beschreibung und der beiliegenden Zeichnung.

30

Es versteht sich, dass die vorstehend genannten und die nachfolgend noch zu erläuternden Merkmale nicht nur in der jeweils angegebenen Kombination, sondern auch in anderen Kombinationen oder in Alleinstellung verwendbar sind, ohne den Rahmen der vorliegenden Erfindung zu verlassen.

35



Die Erfindung ist anhand von Ausführungsbeispielen in der Zeichnung schematisch dargestellt und wird im Folgenden unter Bezugnahme auf die Zeichnung ausführlich beschrieben.

5

#### Figurenbeschreibung

Figuren 1a und 1b zeigen jeweils eine bevorzugte Ausführungsform einer Anordnung aus Daten und Prüfsummen in einer Speichereinheit, wobei jeweils eine Prüfsumme erster Art und eine Prüfsumme zweiter Art vorgesehen sind.

10

Figuren 2a bis 2d zeigen jeweils eine bevorzugte Ausführungsform einer Anordnung aus Daten und Prüfsummen in einer Speichereinheit, wobei die Prüfsumme erster Art jeweils aus drei Teilprüfsummen besteht.

15

Figuren 3a und 3b zeigen jeweils eine bevorzugte Ausführungsform einer Anordnung aus Daten und Prüfsummen in einer Speichereinheit, wobei die Prüfsumme erster Art jeweils aus drei Teilprüfsummen und einer zusätzlichen Teilprüfsumme besteht.

20

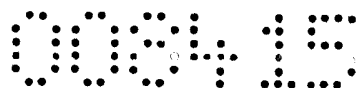
Figuren 4a und 4b zeigen jeweils eine bevorzugte Ausführungsform einer Anordnung aus Daten und Prüfsummen in einer Speichereinheit, wobei die Prüfsumme erster Art und die Prüfsumme zweiter Art jeweils aus drei Teilprüfsummen bestehen.

#### Detaillierte Beschreibung der Zeichnung

25

In den Figuren werden unterschiedliche bevorzugte Ausgestaltungen der Erfindung am Beispiel von zu sichernden Daten und Prüfsummen als Anordnung in einer Speichereinheit dargestellt. Die zu sichernden Daten sind mit DATA bezeichnet. Die Prüfsumme erster Art ("sichere Prüfsumme") ist mit CS\_s, die Prüfsumme zweiter Art ("unsichere Prüfsumme") ist mit CS\_u bezeichnet. Die Prüfsumme erster Art wird vorzugsweise nur einmalig, insbesondere beim Programmstart, zur Überprüfung verwendet, wohingegen die Prüfsumme zweiter Art vorzugsweise regelmäßig während des Programmbetriebs zur Überprüfung verwendet wird. Beide Prüfsummen sind vorhanden und werden verwendet. Die sichere Prüfsumme CS\_s ist aufwändig und rechenzeitintensiv. Dies ist jedoch beim Programmstart (Initialisierung des Gerätes) meist von untergeordneter Bedeutung. Die einfache Prüfsumme CS\_u ist rechen-

35



zeitoptimal, d.h. sie belastet die Sicherheits-CPU während ihrer eigentlichen sicherheitsgerichteten Arbeit nicht.

5 In Figur 1a ist dargestellt, dass eine sichere Prüfsumme CS2s (z.B. 128 Bit CRC) mit einer unsicheren Prüfsumme CS1u (z.B. 8 Bit Addition) verknüpft wird. Vorzugsweise sind die Daten und die Prüfsumme zweiter Art durch die Prüfsumme erster Art gesichert. Dies ist in Figur 1b illustriert. Hierdurch wird erreicht, dass beim erstmaligen Prüflauf (Programmstart) auch eine Korruption der einfacheren Prüfsumme sicher erkannt wird.

10 Es kann vorgesehen sein, dass die Prüfsumme erster Art und/oder die Prüfsumme zweiter Art jeweils mehrere Teilprüfsummen umfassen. Dementsprechend können die Daten in mehrere Teildatenbereiche unterteilt werden. Vorzugsweise wird jedem Teildatenbereich eine Teilprüfsumme zugeordnet. Dies erhöht insbesondere bei CRC-Prüfsummen erster Art die Sicherheit, da Fehler in kleineren Datenmengen mit größerer Sicherheit erkannt werden  
15 können.

In den Figuren 2a und 2b ist gezeigt, dass drei Teildatenbereiche DATA1, DATA2 und DATA3 sowie drei zugehörige sichere Teilprüfsummen CS1s, CS2s und CS3s vorgesehen sind. Die unsichere Prüfsumme CS4u ist hier nicht unterteilt, sondern auf die gesamten Daten DATA gerichtet. Figur 2b unterscheidet sich von Figur 2a nur durch die Platzierung der  
20 Teildatenbereiche und Teilprüfsummen in der Speichereinheit. Die Platzierung gemäß Figur 2b, bei der die Teildatenbereiche unmittelbar aneinander stehen, ist im Hinblick auf die Generierung der Daten vorteilhaft, da keine Unterbrechungen im Code gemacht werden müssen, sondern die Prüfsummen CS1s, CS2s, CS3s hinten angehängt werden können (ebenso  
25 Figuren 2d, 3b, 4a, 4b). In Figur 2c ist illustriert, dass die Daten DATA3 und die unsichere Prüfsumme CS4u durch die sichere Teilprüfsumme CS3s gesichert sind. In Figur 2d ist illustriert, dass jeder Teildatenbereich DATA1, DATA2 und DATA3 zusammen mit der unsicheren Prüfsumme CS4u durch die zugehörige sichere Teilprüfsumme CS1s, CS2s bzw. CS3s gesichert ist (beispielsweise kann die unsichere Prüfsumme CS4u gemeinsam mit dem Datenfeld DATA3 durch die sichere Teilprüfsumme CS3s gesichert werden).  
30

In den Figuren 3a und 3b ist dargestellt, dass die mehreren, jeweils einem Teildatenbereich zugeordneten Teilprüfsummen CS1s, CS2s, CS3s zusammen mit einer weiteren Teilprüfsumme CS5s die Prüfsumme erster Art bilden. Soll eine sehr hohe Sicherheit erreicht werden, so sind lange CRC-Prüfsummen vonnöten (z.B. 128 Bit). Dies kann jedoch vereinfacht  
35



werden, indem mehrere kürzere CRC-Teilprüfsummen verwendet werden (z.B. 32 Bit), die relativ einfach berechnet werden können, und diese mit einer langen weiteren Prüfsumme gesichert werden (z.B. 128 Bit). Vorzugsweise wird die zusätzliche Teilprüfsumme CS5s nur über die Teilprüfsummen CS1s, CS2s, CS3s (und vorzugsweise CS4u) gebildet. Es kann jedoch auch vorgesehen sein, dass die zusätzliche Teilprüfsumme CS5s über alle Teildatenbereiche DATA1, DATA2, DATA3 und alle Teilprüfsummen CS1s, CS2s, CS3s (und vorzugsweise CS4u) gebildet wird.

In den Figuren 4a und 4b ist gezeigt, dass drei Teildatenbereiche DATA1, DATA2 und DATA3 sowie drei zugehörige sichere Teilprüfsummen CS1s, CS2s und CS3s und drei zugehörige unsichere Teilprüfsummen CS4u, CS5u und CS6u vorgesehen sind. Der Gesamtaufwand einer Prüfsummenprüfung ist nahezu unabhängig davon, in wie viele Teildatenfelder eine Datenmenge zerteilt wird. Lediglich fällt für jedes Teildatenfeld eine Initialisierung bzw. Prüfung der Prüfsumme zusätzlich an. Dies kann bei relativ großen Teildatenfeldern im Vergleich zur Bildung der jeweiligen Prüfsumme jedoch vernachlässigt werden. Aus diesem Grunde kann optional auch die einfachere Prüfsummenprüfung auf mehrere kleinere Teildatenfelder zerlegt werden. Hierdurch ergibt sich zwar keine Erhöhung der Diagnoseabdeckung, wenn man als Annahme einer Additionsprüfsumme davon ausgeht, dass für das Teildatenfeld auch nur eine Erkennung von  $1-1/2^M$  erreicht wird. Jedoch kann dadurch die Wahrscheinlichkeit, dass aufgetretene Datenkorruption unerkannt bleibt, ohne komplexe und aufwändige Algorithmen signifikant verringert werden. Da hierzu alle Prüfsummen der k Teildatenfelder CS4u, ..., CS6u unerkannt verfälscht werden müssten, potenziert sich diese Wahrscheinlichkeit mit der Anzahl k der Prüfsummen auf  $1-1/2^{(kM)}$ .

Patent-  
Ansprüche

5

1. Verfahren zum Überprüfen von Daten in einer Recheneinheit zur Laufzeit eines Programms, wobei die Daten (DATA) mit einer Prüfsumme erster Art (CS2s) und mit einer Prüfsumme zweiter Art (CS1u) überprüft werden, indem die Prüfsumme erster Art und die Prüfsumme zweiter Art zur Laufzeit des Programms berechnet und mit jeweils mit einer gespeicherten Prüfsumme (CS1u, CS2s) verglichen werden, wobei zur Laufzeit des Programms der Rechenzeitbedarf zum Berechnen der Prüfsumme zweiter Art (CS1u) geringer als der Rechenzeitbedarf zum Berechnen der Prüfsumme erster Art (CS2s) ist.

10

15

2. Verfahren nach Anspruch 1, wobei zur Laufzeit des Programms die Prüfsumme erster Art (CS2s) weniger häufig berechnet wird als die Prüfsumme zweiter Art (CS1u).

20

3. Verfahren nach Anspruch 1 oder 2, wobei die Prüfsumme erster Art (CS2s) bei der Initialisierung des Programms berechnet und mit der gespeicherten Prüfsumme verglichen wird.

25

4. Verfahren nach einem der vorstehenden Ansprüche, wobei zur Laufzeit des Programms die Prüfsumme zweiter Art (CS1u) mehrmals, insbesondere in regelmäßigen und/oder vorgebbaren Zeitabständen, berechnet und mit der gespeicherten Prüfsumme verglichen wird.

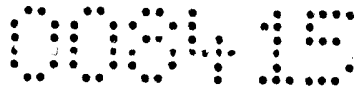
30

5. Verfahren nach einem der vorstehenden Ansprüche, wobei die Prüfsumme erster Art (CS2s) eine CRC-Prüfsumme ist.

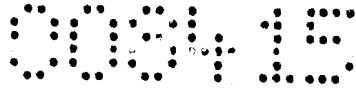
6. Verfahren nach Anspruch 5, wobei die Prüfsumme zweiter Art (CS1u) eine CRC-Prüfsumme ist, deren Länge kürzer als die der Prüfsumme erster Art (CS2s) ist.

35

7. Verfahren nach einem der Ansprüche 1 bis 5, wobei die Prüfsumme zweiter Art (CS1u) eine Additionsprüfsumme, deren Länge vorzugsweise der internen Speicherzugriffsdatenbreite oder einer Registerbreite eines das Programm ausführenden Prozessors der Recheneinheit entspricht, ist.



8. Verfahren nach einem der vorstehenden Ansprüche, wobei die Prüfsumme erster Art (CS2s) über die Daten (DATA) und über die Prüfsumme zweiter Art (CS1u) berechnet wird.
- 5 9. Verfahren nach einem der vorstehenden Ansprüche, wobei die Daten (DATA) in mehrere Teildatenbereiche (DATA1, DATA2, DATA3) unterteilt werden und für jeden der mehreren Teildatenbereiche (DATA1, DATA2, DATA3) jeweils eine Teilprüfsumme erster Art (CS1s, CS2s, CS3s) und/oder jeweils eine Teilprüfsumme zweiter Art (CS4u, CS5u, CS6u) berechnet werden.
- 10 10. Verfahren nach Anspruch 9, wobei eine weitere Teilprüfsumme erster Art (CS5s) über die jeweils eine Teilprüfsumme erster Art (CS1s, CS2s, CS3s) für jeden der mehreren Teildatenbereiche (DATA1, DATA2, DATA3) berechnet wird.
- 15 11. Verfahren nach einem der vorstehenden Ansprüche, wobei die zu überprüfenden Daten vor der Überprüfung aus einem nichtflüchtigen Speicher in einen flüchtigen Speicher der Recheneinheit umkopiert werden und die Daten in dem flüchtigen Speicher überprüft werden.
- 20 12. Verfahren nach einem der vorstehenden Ansprüche, wobei bei einer durch das Vergleichen festgestellten Abweichung einer berechneten Prüfsumme von einer gespeicherten Prüfsumme eine Fehlerreaktion eingeleitet wird.
- 25 13. Verfahren nach einem der vorstehenden Ansprüche, wobei eine von der Prüfsumme erster Art (CS2s) erzielbare Diagnosedeckung höher als eine von der Prüfsumme zweiter Art (CS1u) erzielbare Diagnosedeckung ist.
- 30 14. Verfahren nach einem der vorstehenden Ansprüche, wobei von der Prüfsumme erster Art (CS2s) und/oder von der Prüfsumme zweiter Art (CS1u) eine vorbestimmte Diagnosedeckung erzielbar ist.
15. Recheneinheit, die dazu eingerichtet ist, ein Verfahren nach einem der vorstehenden Ansprüche durchzuführen.



16. Recheneinheit nach Anspruch 15, die dazu eingerichtet ist, sicherheitsgerichtete oder sicherheitskritische Anwendungen durchzuführen.
17. Recheneinheit nach Anspruch 15 oder 16, die Teil einer Sicherheits-SPS ist.
- 5
18. Computerprogramm mit Programmcodemitteln, die eine Recheneinheit veranlassen, ein Verfahren nach einem der Ansprüche 1 bis 14 durchzuführen, wenn sie auf der Recheneinheit, insbesondere nach Anspruch 15, 16 oder 17 ausgeführt werden.
- 10
19. Maschinenlesbares Speichermedium mit einem darauf gespeicherten Computerprogramm nach Anspruch 18.

**14. Okt. 2013**

PATENTANWÄLTE  
PUCHBERGER, BERGER & PARTNER  
A-1010 Wien Reichsratsstrasse 13  
Telefon 512 23 02 Telefax 513 37 09

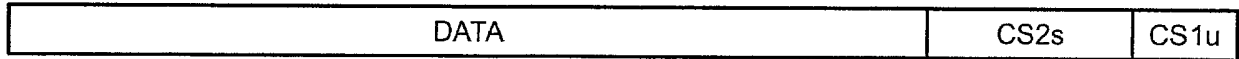


Fig. 1a



Fig. 1b



Fig. 2a

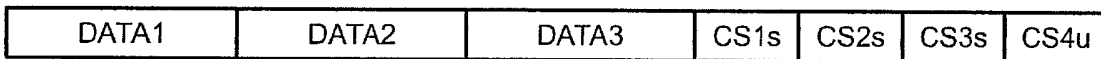


Fig. 2b



Fig. 2c

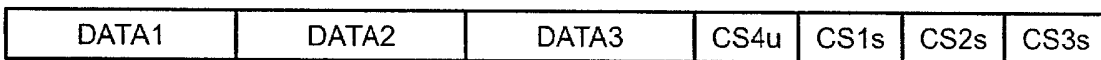


Fig. 2d

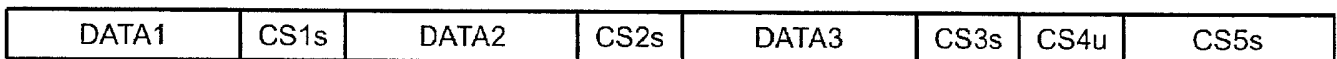


Fig. 3a

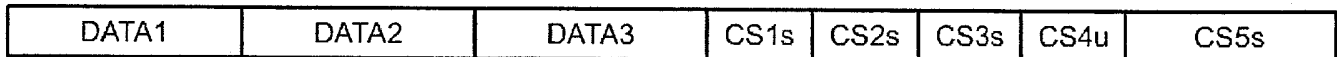


Fig. 3b

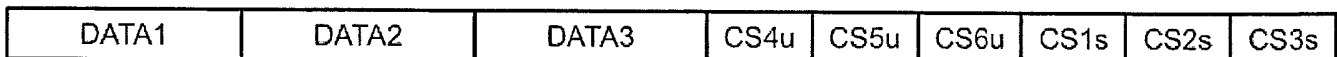


Fig. 4a



Fig. 4b