(12) NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG

(19) Weltorganisation für geistiges Eigentum
Internationales Büro

(43) Internationales Veröffentlichungsdatum
19. August 2004 (19.08.2004)

PCT

(10) Internationale Veröffentlichungsnummer
WO 2004/070496 A2

(51) Internationale Patentklassifikation[7]: G06F

(21) Internationales Aktenzeichen: PCT/EP2004/000519

(22) Internationales Anmeldedatum:
22. Januar 2004 (22.01.2004)

(25) Einreichungssprache: Deutsch

(26) Veröffentlichungssprache: Deutsch

(30) Angaben zur Priorität:
103 04 900.2 6. Februar 2003 (06.02.2003) DE

(71) Anmelder *(für alle Bestimmungsstaaten mit Ausnahme von US)*: **INFINEON TECHNOLOGIES AG** [DE/DE]; St.-Martin-Str. 53, 81669 München (DE).

(72) Erfinder; und

(75) Erfinder/Anmelder *(nur für US)*: **SCHEPERS, Jörg** [DE/DE]; Ennemoserstr. 8, 83700 Rottach-Egern (DE).

(74) Anwälte: **ZIMMERMANN, Tankred** usw.; Postfach 246, 82043 Pullach bei München (DE).

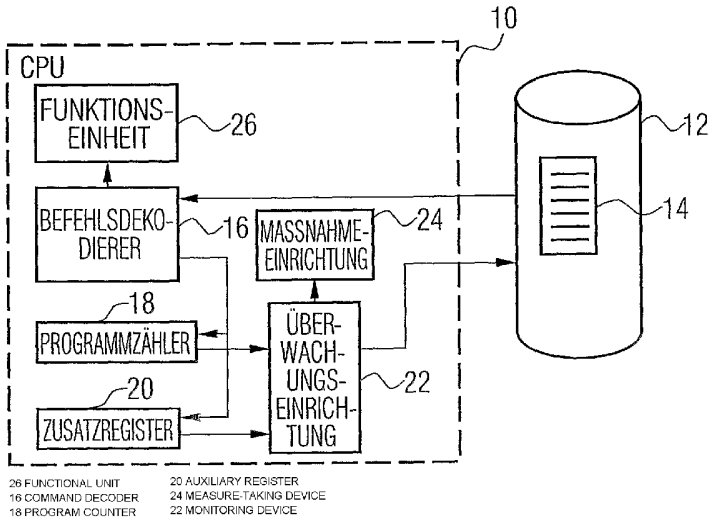
(81) Bestimmungsstaaten *(soweit nicht anders angegeben, für jede verfügbare nationale Schutzrechtsart)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Bestimmungsstaaten *(soweit nicht anders angegeben, für jede verfügbare regionale Schutzrechtsart)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), eurasisches (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), europäisches (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT,

[Fortsetzung auf der nächsten Seite]

(54) Title: CIRCUIT HAVING PROTECTION AGAINST MANIPULATIVE ATTACKS AND METHOD

(54) Bezeichnung: SCHALTUNG MIT SCHUTZ VOR MANIPULATIVEN ANGRIFFEN UND VERFAHREN



(57) **Abstract:** An inventive circuit comprises a first operation unit (18) for carrying out an operation, whereby the first operation unit, when carrying out the operation, changes its state in a first manner. The circuit also comprises a second operation unit (20) for carrying out the operation, whereby the second operation unit, when carrying out the operation, changes its state in second first manner that differs from the first. The circuit additionally comprises a monitoring device (22) for verifying the state of the first operation unit (18) and the state of the second operation unit (20) and for signaling an alarm when the state of the first operation unit and the state of the second operation unit do not have a predetermined relationship with one another. Lastly, the circuit comprises a device (24) for taking a measure in response to a signaling of the alarm. The improvement consists of being able to detect manipulative attacks more effectively or with a higher probability.

[Fortsetzung auf der nächsten Seite]

WO 2004/070496 A2

RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Veröffentlicht:

— *ohne internationalen Recherchenbericht und erneut zu veröffentlichen nach Erhalt des Berichts*

Zur Erklärung der Zweibuchstaben-Codes und der anderen Abkürzungen wird auf die Erklärungen ("Guidance Notes on Codes and Abbreviations") am Anfang jeder regulären Ausgabe der PCT-Gazette verwiesen.

(57) Zusammenfassung: Eine erfindungsgemässe Schaltung umfasst eine erste Operationseinheit (18) zum Durchführen einer Operation, wobei die erste Operationseinheit bei Durchführung der Operation ihren Zustand auf eine erste Weise verändert, eine zweite Operationseinheit (20) zum Durchführen der Operation, wobei die zweite Operationseinheit bei Durchführung der Operation ihren Zustand auf eine zweite Weise verändert, die sich von der ersten Weise unterscheidet, eine Überwachungseinrichtung (22) zum Untersuchen des Zustands der ersten Operationseinheit (18) und des Zustands der zweiten Operationseinheit (20) und zum Signalisieren eines Alarms, wenn der Zustand der ersten Operationseinheit und der Zustand der zweiten Operationseinheit nicht in einer vorbestimmten Beziehung zueinander stehen, und eine Einrichtung (24) zum Ergreifen einer Massnahme ansprechend auf eine Signalisierung des Alarms hin. Die Verbesserung besteht darin, dass manipulative Angriffe effektiver oder mit höherer Wahrscheinlichkeit erfasst werden können.

Beschreibung

Schaltung mit Schutz vor manipulativen Angriffen und Verfahren

Die vorliegende Erfindung bezieht sich auf den Schutz von Schaltungen, wie z.B. in Chipkarten, vor manipulativen Angriffen, wie z.B. DFA- (Differential Fault Analysis = Differentielle Fehleranalyse) Angriffen oder Angriffen mit dem Ziel, den ordnungsgemäßen Programmfluss der Schaltung zu stören, um Sicherheitsabfragen oder dergleichen zu umgehen.

Der ordnungsgemäße Betrieb eines Mikrocontrollers kann durch eine Vielzahl externer Störungen beeinflusst werden. Beispiele externer Störungen umfassen das Beschicken des Mikrocontrollers mit Strompulsen, das Aussetzen des Mikrocontrollers gegenüber Lichtpulsen, elektromagnetischer Einstrahlung oder Temperaturänderungen oder dergleichen. Die Beeinflussung oder Manipulation des Mikrocontrollers kann dazu führen, dass die Programmausführung unterbrochen oder dahingehend gestört wird, dass der programmierte Kontrollfluss des Programms umgangen wird, um beispielsweise Sicherheitsabfragen zu umgehen oder Kontrollparameter zu modifizieren. Ziel solcher Manipulationen kann neben der Umgehung des programmierten Kontrollflusses auch darin bestehen, den Mikrocontroller oder Coprozessoren desselben zu einem fehlerhaften Betrieb zu provozieren, so dass derselbe ein fehlerhaftes Ergebnis ausgibt. Basierend auf einem solchen fehlerhaften Ergebnis können gemäß einem DFA-Angriff Rückschlüsse auf beispielsweise einen geheimen Schlüssel oder andere sicherheitsrelevante Daten gezogen werden.

Heutzutage werden derartige externe Störungen zumeist über die Detektion bzw. Erfassen des auslösenden Effekts bzw. der externen Beeinflussung erkannt. Hierzu werden Sensoren vorgesehen, die sensitiv auf die physischen Beeinflussungen, wie Stromimpulse, Lichtpulse oder ähnliches (siehe oben), reagie-

ren und daraufhin einen Sicherheitsalarm auslösen. Diese Vorgehensweise ist darin nachteilhaft, dass es teilweise recht schwierig ist, die Erfassung der Beeinflussungen zuverlässig durchzuführen. Zudem definieren die vorhandenen Sensoren eine abschließende Menge von erkennbaren Manipulationen, die nicht unbedingt alle möglichen Arten von Manipulationen umfassen muss.

Eine andere Strategie sieht vor, die fehlerhafte Funktionsweise der Schaltung an sich zu erkennen. So ist es beispielsweise durch zweimaliges Berechnen einer Funktion, wie z.B. einer Modulo-Berechnung im Rahmen eines RSA-Algorithmus, hintereinander möglich, durch Vergleich der sich bei den beiden Malen ergebenden Ergebnissen und Feststellen, ob die Ergebnisse identisch sind, auf eine Manipulation rückzuschließen, indem angenommen wird, dass eine Manipulation zu keinen deterministischen, sondern einem zufälligen Ergebnis führt. Ein Vergleich der beiden Ergebnisse lässt somit mit einer hohen Wahrscheinlichkeit für dessen Richtigkeit den Schluss zu, dass keine Manipulation der Schaltung vorliegt und das Funktionsergebnis richtig ist. Vorgehensweisen dieser Art sind nachteilhaft, da sie aufgrund der doppelten Durchführung der Funktion die Ausführungszeit und den Energieverbrauch verdoppeln, was insbesondere bei batterielosen Anwendungen und im Bereich von Kontaktloschipkarten von besonderem Nachteil ist.

Der Nachteil der längeren Verarbeitungszeit ließe sich durch Verdopplung der entsprechenden die Funktion durchführenden Einheiten ausräumen. Das Ergebnis solcher auf die gleiche Weise arbeitenden Funktionseinheiten könnte miteinander verglichen werden, um bei Identität das Fehlen manipulativer Angriffe anzunehmen. Der Nachteil des doppelten Energieverbrauchs bleibt jedoch erhalten und die hohe Geschwindigkeit wird mit einer verdoppelten Chipfläche erkauft, was insbesondere bei Massenprodukten, wie Chipkarten, zu enormen Kostennachteilen führt.

Demgegenüber besteht die Aufgabe der vorliegenden Erfindung darin, eine Schaltung und ein Verfahren zur Steuerung derselben zu schaffen, so dass die Sicherheit der Schaltung vor manipulativen Angriffen erhöht werden kann.

Diese Aufgabe wird durch eine Schaltung gemäß Anspruch 1 und ein Verfahren gemäß Anspruch 10 gelöst.

Eine erfindungsgemäße Schaltung umfasst eine erste Operationseinheit zum Durchführen einer Operation, wobei die erste Operationseinheit bei Durchführung der Operation ihren Zustand auf eine erste Weise verändert, eine zweite Operationseinheit zum Durchführen der Operation, wobei die zweite Operationseinheit bei Durchführung der Operation ihren Zustand auf eine zweite Weise verändert, die sich von der ersten Weise unterscheidet, eine Überwachungseinrichtung zum Untersuchen des Zustands der ersten Operationseinheit und des Zustands der zweiten Operationseinheit und zum Signalisieren eines Alarms, wenn der Zustand der ersten Operationseinheit und der Zustand der zweiten Operationseinheit nicht in einer vorbestimmten Beziehung zueinander stehen, und eine Einrichtung zum Ergreifen einer Maßnahme ansprechend auf eine Signalisierung des Alarms hin.

Der Kerngedanke der vorliegenden Erfindung besteht darin, dass manipulative Angriffe effektiver oder mit höherer Wahrscheinlichkeit erfasst werden können, wenn Schaltungsteile bzw. Operationseinheiten einer Schaltung nicht einfach verdoppelt werden, sondern dass denselben redundante Schaltungsteile bzw. Operationseinheiten nebengeordnet werden, die dieselbe Operation durchführen, aber bei Durchführung der Operation ihren (Ergebnis-) Zustand auf eine andere Weise ändern. Durch Untersuchung der Zustände beider Operationseinheiten ist es dann möglich, festzustellen, ob die beiden Zustände in einer vorbestimmten Beziehung zueinander stehen, und, falls dies nicht der Fall ist, hieraus auf einen manipu-

lativen Angriff zu schließen. Im Vergleich zur reinen Verdopplung von Schaltungsteilen bzw. Operationseinheiten trägt die erfindungsgemäße Vorgehensweise dem Umstand Rechnung, dass manipulative Störungen im allgemeinen die Modifikation von Registerinhalten bzw. Zuständen von solchen Operationseinheiten in eine Vorzugsrichtung bewirken, wie z.B. alle Bits eines Registers auf den Zustand 0 oder alle Bits eines Registers auf den Zustand 1. Aufgrund der Tatsache, dass beide Operationseinheiten bei Durchführung der Operation ihren Zustand auf verschiedene Weise ändern, wirken sich somit auch die Modifikationen bzw. Störungen in verschiedener Weise aus und lassen beim Vergleich der beiden Zustände die Manipulation erkennen.

Gemäß einem speziellen Ausführungsbeispiel der vorliegenden Erfindung ist es zum Schutz gegen die Manipulation des Programmkontrollflusses in einem Prozessor vorgesehen, dem Programmzähler des Prozessors, also der Einheit, die das Register, welches die nächste auszuführende Instruktionsadresse für den Prozessor enthält, verwaltet, um einen weiteren Programmzähler zu ergänzen, welcher in seinem Zusatzregister in dem Fall keiner Manipulation stets eine Darstellung der nächsten auszuführenden Instruktionsadresse in einer bitweise invertierten Darstellung enthält. Programmzähleraktualisierungseinrichtung und Zusatzregisteraktualisierungseinrichtung der beiden Programmzähler arbeiten komplementär zueinander, d.h. beim Laden einer neuen Adresse, wie es beispielsweise bei einem Programmsprung der Fall ist, lädt die Programmzähleraktualisierungseinrichtung die neue Adresse in das Programmzählerregister, während das Zusatzregister mit der invertierten neuen Adresse geladen wird. Auf ähnliche Weise dekrementiert die Zusatzregisteraktualisierungseinrichtung den Registerinhalt des Zusatzregisters, wenn der Programmzählerwert durch die Programmzähleraktualisierungseinrichtung inkrementiert wird. Auf diese Weise bleiben, keine Manipulation der Schaltung vorausgesetzt, die Bitwerte der Registerinhalte beider Register immer in einer derartigen Beziehung

zueinander, dass der Registerinhalt des Zusatzregisters eine invertierte Darstellung des Programmzählerwerts in dem Programmzählerregister ist. Ein Wegfall dieser Beziehung deutet auf einen manipulativen Angriff hin, der bei Erfolg zu einer Störung des Programmkontrollflusses bzw. einem Verspringen des Prozessors, wie z.B. einer CPU, führen könnte, und damit zur Umgehung bestimmter Sicherheitsabfragen oder ähnliches.

Bevorzugte Ausführungsbeispiele der vorliegenden Erfindung werden nachfolgend Bezug nehmend auf die beiliegenden Zeichnungen näher erläutert. Es zeigen:

Fig. 1 ein schematisches Blockdiagramm einer Schaltung gemäß einem Ausführungsbeispiel der vorliegenden Erfindung, bei dem neben dem Programmzähler einer CPU ein weiterer Programmzähler mit invertierter Darstellung vorgesehen ist; und

Fig. 2 ein schematisches Blockdiagramm, das eine Realisierung des Programmzählers und des invertierten Programmzählers der Schaltung von Fig. 1 gemäß einem Ausführungsbeispiel der vorliegenden Erfindung detaillierter darstellt.

Zunächst wird darauf hingewiesen, dass gleiche Elemente in den Fig. 1 und 2 mit gleichen Bezugszeichen versehen sind und dass eine wiederholte Beschreibung der Elemente weggelassen wird.

Fig. 1 zeigt eine CPU bzw. zentrale Verarbeitungseinheit 10, die ein in einem Speicher 12 gespeichertes Programm 14 ausführt. Die CPU 10 weist einen Befehlsdecodierer 16, einen Programmzähler 18, ein Zusatzregister 20, eine Überwachungseinrichtung 22, eine Maßnahmeeinrichtung 24 und weitere Funktionseinheiten, die allgemein mit dem Bezugszeichen 26 versehen sind, auf.

Die in Fig. 1 gezeigte Anordnung ist beispielsweise ein Controller einer Chipkarte, die zur sicheren Kommunikation mit einem Terminal, wie z.B. einem Bankautomaten, vorgesehen ist. Das Programm 14 ist dann beispielsweise eine Terminalapplikation mit Unterprogrammen zur Durchführung spezieller Vorgänge während einer Transaktion mit dem Terminal, wie z.B. der Auf/Abbuchung eines Geldbetrages auf ein auf der Chipkarte gespeichertes Guthaben, der Überprüfung einer PIN (persönlichen Identifikationsnummer), die ein Benutzer am Terminal eingegeben hat, der Überprüfung der Authentifikation des Terminals bzw. Kommunikationspartners oder dergleichen.

Der Befehlsdecodierer 16 ist mit dem Speicher 12 verbunden, um den nächsten auszuführenden Befehl bzw. die nächste auszuführende Instruktion aus dem Programm 14 zu erhalten. Die auszuführenden Instruktionen können aus mehreren Bytes bestehen und beispielsweise aus Operationscode einerseits und Operanden andererseits zusammengesetzt sein, wie z.B. aus Sprungbefehlscode und zugeordneter Sprungadresse. Der Befehlsdecodierer 16 decodiert die ankommende Instruktion und weist entsprechende Funktionseinheiten 26 an, die Instruktion auszuführen.

Die Instruktionen in dem Programmcode 14 weisen ihrer Befehlsart bzw. ihrem Reparationscode entsprechend, die bzw. der einer bzw. einem aus einem Befehlssatz der CPU entspricht, eine unterschiedliche Anzahl von Bytes auf. Der Befehlsdecodierer 16 signalisiert dem Programmzähler 18 (program counter = PC), einen in dem selben gespeicherten und von demselben verwalteten Programmzählerwert um einen dieser Anzahl entsprechenden Wert zu inkrementieren. Der Programmzählerwert zeigt somit zwischen zwei aufeinanderfolgenden Befehlszyklen der CPU 10 immer auf die nächste zu verarbeitende Instruktion in dem Programmcode 14. Sobald also in einer Ausführungsphase bzw. einem Befehlszyklus der CPU 10 eine Instruktion ausgeführt worden ist, wird in dem darauffolgenden Befehlszyklus unter Verwendung des Programmzähler-

werts in dem Programmzähler 18 auf den Speicher 12 zugegriffen, um die nächste zu verarbeitende Instruktion in den Befehlsdecodierer 16 auszulesen, um somit die nächste Ausführungsphase zu beginnen.

Um jedoch zu verhindern, dass durch externe Beeinflussungen, wie z.B. durch Strompulse, Lichtpulse, elektromagnetische Einstrahlung, Temperaturänderung oder dergleichen, mit einem manipulierten Programmzählerwert in dem Programmzähler 18 weitergearbeitet wird, und somit eventuell Programmteile des Programms 14 übersprungen werden, wie z.B. Sicherheitsabfragen, PIN-Abfragen oder dergleichen, wird der Wert des Programmzählers 18 nicht ohne eine vorherige Überprüfung durch die Überwachungseinrichtung 22 an den Speicher 12 ausgegeben, um mit demselben auf die nächsten auszuführende Instruktion zuzugreifen.

Um eine Manipulation des Programmzählerwerts in dem Programmzähler 18 erkennbar zu machen, ist in der CPU 10 ein invertierter Programmzähler 20 mit einem Zusatzregister (in Fig. 1 nicht gezeigt) vorgesehen. Der invertierte Programmzähler 20 enthält in dem Zusatzregister eine redundante, invertierte Darstellung des Programmzählerwerts, wie er in dem Register des Programmzählers 18 gespeichert ist. Die invertierte Darstellung kann ein Einerkomplement oder ein Zweierkomplement der nicht invertierten Darstellung sein.

Ebenso wie der Programmzähler 18 ist der invertierte Programmzähler 20 mit dem Befehlsdecodierer 16 verbunden, um von demselben Signale bezüglich der Aktualisierung des Programmzählerwerts zu erhalten. Die Ausgänge des Programmzählers 18 sowie des invertierten Programmzählers 20 sind mit der Überwachungseinrichtung 22 verbunden.

Da bei Nichtvorliegen einer Manipulation Programmzählerwert im Programmzähler 18 und der Bitwert in dem invertierten Programmzähler 20 bitweise zueinander invertiert sein soll-

ten, kann die Überwachungseinrichtung 22 auf der Grundlage eines bitweisen Vergleichs der beiden Werte und Überprüfung, ob die Bits derselben immer unterschiedlich bzw. komplementär zueinander sind, entscheiden, ob eine äußere Beeinflussung bzw. ein manipulativer Angriff auf die CPU 10 vorliegt oder nicht. Sind die Programmzählerwerte im Programmzähler 18 und der Wert in dem invertierten Programmzähler 20 zueinander bitweise invers, gibt die Überwachungseinrichtung 22 den Programmzählerwert des Programmzählers 18 an den Speicher 12 weiter. Andernfalls signalisiert die Überwachungseinrichtung 22 der Maßnahmeeinrichtung 24, dass dieselbe geeignete Maßnahmen treffen soll, um Sicherheitsrisiken auszuschalten, die sich durch Störung des durch das Programm 14 festgelegten Programmkontrollflusses ergäben.

Die Maßnahmeeinrichtung 24 sorgt auf das Alarmsignal der Überwachungseinrichtung 22 hin beispielsweise dafür, dass jedwede weitere Programmausführung des laufenden Programms unterbunden wird, oder jedenfalls jegliche Ausgabe eines Ergebnisses, wie es sich nach der Ausführung des Programms 14 ergibt, unterbleibt, beispielsweise durch Ausschalten der CPU, Beenden der Abarbeitung des Programms 14 und Fortsetzen bei einem Grundzustand oder einer Grundroutine dergleichen.

Alternativ oder zusätzlich ist die Maßnahmeeinrichtung ausgebildet, um einen Interrupt-Sprung zu einer Interrupt-Routine zu bewirken und/oder ein durch ein Betriebssystem abfragbares Alarmbit zu setzen.

Nachdem im vorhergehenden Bezug nehmend auf Fig. 1 der Aufbau der CPU 10 und die generelle Funktionsweise derselben im Zusammenhang mit dem Programmzähler 18 sowie der Zweck des zusätzlich vorgesehenen invertierten Programmzählers 20 beschrieben worden ist, wird im folgenden der genaue Aufbau des Programmzählers 18 und des zusätzlich vorgesehenen invertierten Programmzählers 20 beschrieben.

Fig. 2 zeigt mit gestrichelten Kästchen den Programmzähler 18 und den zusätzlichen invertierten Programmzähler 20. Beide umfassen ein Register 18a bzw. ein Zusatzregister 20a. In dem Register 18a ist der Programmzählerwert gespeichert, der die nächste auszuführende Instruktionsadresse bzw. den Programmzählerwert in normaler binärer Darstellung enthält. In dem Register 20a ist die invertierte Darstellung der nächsten auszuführenden Instruktionsadresse gespeichert.

Sowohl Programmzähler 18 als auch invertierter Programmzähler 20 weisen eine Veränderungseinrichtung 18b bzw. 20b auf. Die Veränderungseinrichtung 18b des Programmzählers 18 inkrementiert auf ein Inkrementsignal von dem Befehlsdecodierer 16 hin den in dem Register 18a enthaltenen Programmzählerwert. Die Veränderungseinrichtung 18b fungiert folglich als Inkrementierer. Auf ähnliche Weise enthält die Veränderungseinrichtung 20b dasselbe Inkrementsignal von dem Befehlsdecodierer 16, fungiert jedoch als Dekrementierer 20b, um auf das Inkrementsignal hin den in dem Register 20a gespeicherten Wert zu dekrementieren. Auf diese Weise ist gewährleistet, dass auch bei Aktualisierung des Programmzählerwerts in dem Register 18a während linearer Abarbeitung des Programms 14 in dem Register 20a stets eine invertierte Darstellung des Programmzählerwerts erhalten bleibt, die sich von der Bitdarstellung des Programmzählerwerts in dem Register 18a dadurch unterscheidet, dass jedes Bit des Programmzählerwerts in dem Register 18a zu dem entsprechenden Bit in dem Register 20a invertiert ist.

Auf ähnliche Weise ist sowohl bei dem Programmzähler 18 als auch bei dem invertierten Programmzähler 20 ein Eingang 18c bzw. 20c vorgesehen, der mit dem Befehlsdecodierer 16 verbunden ist, um in dem Fall eines Sprungsbefehles einen neuen Wert zu erhalten, auf den der Registerinhalt des Registers 18b bzw. 20a eingestellt werden soll. Im Unterschied zu dem Programmzähler 18 ist jedoch bei dem invertierten Programmzähler 20 der Eingang 20c nicht derart mit dem Register 20a

verbunden, dass der neu einzutragende, an dem Eingang 20c anliegende Wert direkt in das Register 20a übernommen wird. Vielmehr ist zwischen Eingang 20c und Register 20a ein Invertierer 20d geschaltet, welcher den in das Register 20a einzutragenden Wert vor seiner Eintragung bitweise invertiert. Wenn folglich der Befehlsdecodierer 16 eine Sprungadresse an den Programmzähler 18 ausgibt, wird die Sprungadresse bei dem Programmzähler 18 an dem Eingang 18c empfangen und direkt in das Register 18a eingetragen, während bei dem invertierten Programmzähler 20 die Sprungadresse nach Empfang an dem Eingang 20c zunächst invertiert und erst dann in das Register 20a eingetragen wird.

Durch Vorsehen von Inkrementierer 18b, Dekrementierer 20b und Invertierer 20d ist folglich sichergestellt, dass während des gesamten Betriebs der CPU 10 der Registerinhalt des Registers 20a stets die invertierte Darstellung des Programmzählerwerts in dem Register 18a widerspiegelt, und zwar sowohl in dem Fall der linearen Programmabarbeitung ohne Sprünge und in dem Fall von Programmsprüngen.

Die Manipulationen werden beim obigen Ausführungsbeispiel folglich dadurch erkennbar gemacht, dass Programmzähler und invertierter Programmzähler durch entsprechende verschiedene Steuerungsmechanismen, d.h. Dekrementierer und Inkrementierer, die auch Teil der CPU sein können, um nebenher auch noch andere Aufgaben zu erfüllen, die neuen Werte bestimmen.

Die Ausgänge der Register 18a und 20a sind mit der Überwachungseinrichtung 22 verbunden, die somit anhand einer Störung dieser inversen Beziehung zwischen Programmzählerwert in dem Register 18a und invertierter Darstellung in den Register 20a auf einen manipulativen Angriff schließen kann.

Bei dem Bezug nehmend auf die Fig. 1 und 2 beschriebenen Ausführungsbeispiel wurden externe Störungen bzw. Beeinflussungen von der Überwachungseinrichtung bzw. Testschaltung 22

zu Beginn jedes Befehlszyklus erfasst, indem überprüft wurde, ob die Darstellungen in den Registern 18a und 20a konsistent bzw. bitweise invers zueinander sind, um, falls dies nicht der Fall war, einen Alarm auszulösen. Es ist jedoch ferner möglich, die Untersuchung durch die Überwachungseinrichtung nach der Befehlsdekodierung vor jeder Ausführungsphase durchzuführen, oder die Untersuchung nicht bei jedem Befehlszyklus auszuführen sondern nur an bestimmten, gegebenenfalls zufälligen, Zeitpunkten, was freilich von der erwünschten Sicherheit vor Manipulationen abhängt.

Vorteile der in Fig. 1 und 2 gezeigten Schaltung bestehen jedenfalls darin, dass sie einfach zu implementieren ist, nicht viel Platz kostet und in einem Designentwurf gut versteckt werden kann. Sie bietet ferner Schutz vor einer Vielzahl von Angriffen, wobei sie deren Wirkung, d.h. die Abweichung der Funktionsweise der Schaltung von der Sollfunktionsweise, und nicht den Auslöser selbst, d.h. die äußeren physischen Beeinflussungen der Schaltung, erfasst, was oftmals wesentlich schwieriger ist.

Im Vergleich zu dem Vorsehen redundanter Schaltungsteile bzw. Programmzähler hat die Schaltung von Fig. 1 und 2 den Vorteil, dass sie bei einer größeren Anzahl von Angriffen wirksam ist, nämlich auch bei solchen, bei denen die Störungen im allgemeinen die Modifikation von Registerinhalten in eine Vorzugsrichtung bewirken, nämlich aller Bits in einem logisch niedrigen oder einem logisch hohen Zustand, denn in einem solchen Fall ginge die Beziehung zwischen den Darstellung, invertiert zu nicht-invertiert, verloren.

Das obige Ausführungsbeispiel von Fig. 1 und 2 bezog sich auf die Ergänzung eines Programmzählers um einen invertierten Programmzähler, wobei die Veränderungseinrichtung zur Veränderung des Registerinhalts bei dem invertierten Programmzähler verglichen zu der Veränderungseinrichtung bei dem Programmzähler entsprechend umgestaltet wurde. Es ist jedoch

ferner möglich, das gleiche Konzept auch auf andere Schaltungsteile einer Schaltung, eines Mikrocontrollers oder einer zentralen Verarbeitungseinheit oder dergleichen zu übertragen, bei denen konsistente Informationen in inversen Darstellungen durch unabhängige Schaltungsteile abgeleitet werden können. Beispielsweise könnte eine Operationseinheit zur Handhabung eines Programmstatusbits, wie z.B. eines Übertragsbits oder eines Bits für Zugriffsrechte in einem Prozessor, um eine Operationseinheit ergänzt werden, die ein Register aufweist, welches einen zu dem Programmstatusbitregister inversen Eintrag aufweist, und welches zudem eine Veränderungseinrichtung aufweist, die den Registerinhalt immer genau invers verändert, nämlich von 0 auf 1, wenn das Programmstatusbit von 1 auf 0 verändert wird, und umgekehrt.

Bezugszeichenliste

10	CPU
12	Speicher
14	Programm
16	Befehlsdecodierer
18	Programmzähler
18a	Register
18b	Inkrementierer
18c	Eingang
20	invertierter Programmzähler
20a	Register
20b	Dekrementierer
20c	Eingang
20d	Invertierer
22	Überwachungseinrichtung
24	Maßnahmeeinrichtung
26	Funktionseinheiten

Patentansprüche

1. Schaltung mit

einer ersten Operationseinheit (18) zum Durchführen einer Operation, wobei die erste Operationseinheit bei Durchführung der Operation ihren Zustand auf eine erste Weise verändert;

einer zweiten Operationseinheit (20) zum Durchführen der Operation, wobei die zweite Operationseinheit bei Durchführung der Operation ihren Zustand auf eine zweite Weise verändert, die sich von der ersten Weise unterscheidet;

einer Überwachungseinrichtung (22) zum Untersuchen des Zustands der ersten Operationseinheit (18) und des Zustands der zweiten Operationseinheit (20) und zum Signalisieren eines Alarms, wenn der Zustand der ersten Operationseinheit und der Zustand der zweiten Operationseinheit nicht in einer vorbestimmten Beziehung zueinander stehen; und

einer Einrichtung (24) zum Ergreifen einer Maßnahme ansprechend auf eine Signalisierung des Alarms.

2. Schaltung gemäß Anspruch 1, bei der die erste Operationseinheit (18) ein erstes Register (18a) und eine erste Veränderungseinrichtung (18b) aufweist, und die zweite Operationseinheit (20) ein zweites Register (20a) und eine zweite Veränderungseinrichtung (20b) aufweist, wobei der Registerinhalt des ersten Registers (18a) den Zustand der ersten Operationseinheit (18) und der Registerinhalt des zweiten Registers (20a) den Zustand der zweiten Operationseinheit (20) definiert, und wobei die erste Veränderungseinrichtung (18b) den Registerinhalt des ersten Registers (18a) abhängig von einem Signal auf die erste Weise und die zweite Veränderungseinrichtung (20b) den Registerinhalt des zweiten Registers (20a) abhängig von demselben Signal auf die zweite Weise verändert.

3. Schaltung gemäß Anspruch 1 oder 2, bei der die erste Operationseinheit (18) und die zweite Operationseinheit (20) auf dasselbe Ereignis ansprechen, um die Operation durchzuführen, und derart ausgebildet sind, um bei Durchführung der Operation einen jeweiligen Bitwert zu ändern, der den jeweiligen Zustand derselben zumindest teilweise definiert, und zwischen aufeinanderfolgenden Operationsdurchführungen den jeweiligen geänderten Bitwert beizubehalten.

4. Schaltung gemäß Anspruch 3, bei der sich die erste Weise von der zweiten Weise derart unterscheidet, dass sich bei Durchführung der Operation der Bitwert der zweiten Operationseinheit (20) um einen Wert verringert, um den sich der Bitwert der ersten Operationseinheit (18) erhöht oder umgekehrt.

5. Schaltung gemäß Anspruch 3 oder 4, bei der die vorbestimmte Beziehung darin besteht, dass die Bitwerte der Operationseinheiten (18, 20) invertiert zueinander sind, und die erste und die zweite Operationseinheit (18, 20) derart ausgebildet sind, dass nach Durchführung der Operation der Bitwert der ersten Operationseinheit (18) und der Bitwert der zweiten Operationseinheit (20) invertiert zueinander bleiben.

6. Schaltung nach Anspruch 5, bei der eine Biteinheit einer Operationseinheit ein Einerkomplement oder ein Zweierkomplement zu einer Biteinheit der anderen Operationseinheit ist.

7. Schaltung gemäß einem der Ansprüche 1 bis 5, bei der die erste oder die zweite Operationseinheit derart ausgebildet ist, dass ihr Zustand anderen Teilen der Schaltung zur Verfügung steht, während der Zustand der anderen Operationseinheit lediglich der Überwachungseinrichtung (22) zur Verfügung steht.

8. Schaltung gemäß einem der Ansprüche 1 bis 6, bei der die Schaltung ein Prozessor ist und die erste oder die zweite Operationseinheit einen Programmzähler oder ein Statusbitregister des Prozessors umfasst.

9. Schaltung gemäß einem der Ansprüche 1 bis 7, bei der die Maßnahme geeignet ist, um zu verhindern, dass der Prozessor eine laufende Programmausführung fortsetzt, dass der Prozessor einen Interrupt-Sprung zu einer Interrupt-Routine ausführt und/oder dass der Prozessor ein durch ein Betriebssystem abfragbares Alarmbit setzt.

10. Schaltung gemäß einem der Ansprüche 1 bis 8, bei der die Überwachungseinrichtung (22) ausgebildet ist, um die Untersuchung vor jedem Mal vorzunehmen, da der Rest der Schaltung den Zustand der ersten Operationseinheit oder der zweiten Operationseinheit verwendet.

11. Verfahren zum Steuern einer Schaltung mit einer ersten Operationseinheit zum Durchführen einer Operation, die bei Durchführung der Operation ihren Zustand auf eine erste Weise verändert, und einer zweiten Operationseinheit zum Durchführen der Operation, die bei Durchführung der Operation ihren Zustand auf eine zweite Weise verändert, die sich von der ersten Weise unterscheidet, mit folgenden Schritten:

Durchführen der Operation mittels der ersten Operationseinheit;

Durchführen der Operation mittels der zweiten Operationseinheit;

nach den Schritten des Durchführens der Operation durch die erste und die zweite Operationseinheit, Untersuchen des Zustands der ersten Operationseinheit und des unterschiedlichen Zustands der zweiten Operationseinheit;

Signalisieren eines Alarms, wenn der Zustand der ersten Operationseinheit und der Zustand der zweiten Operationseinheit nicht in einer vorbestimmten Beziehung zueinander stehen; und

Ergreifen einer Maßnahme ansprechend auf eine Signalisierung des Alarms.

FIG 1

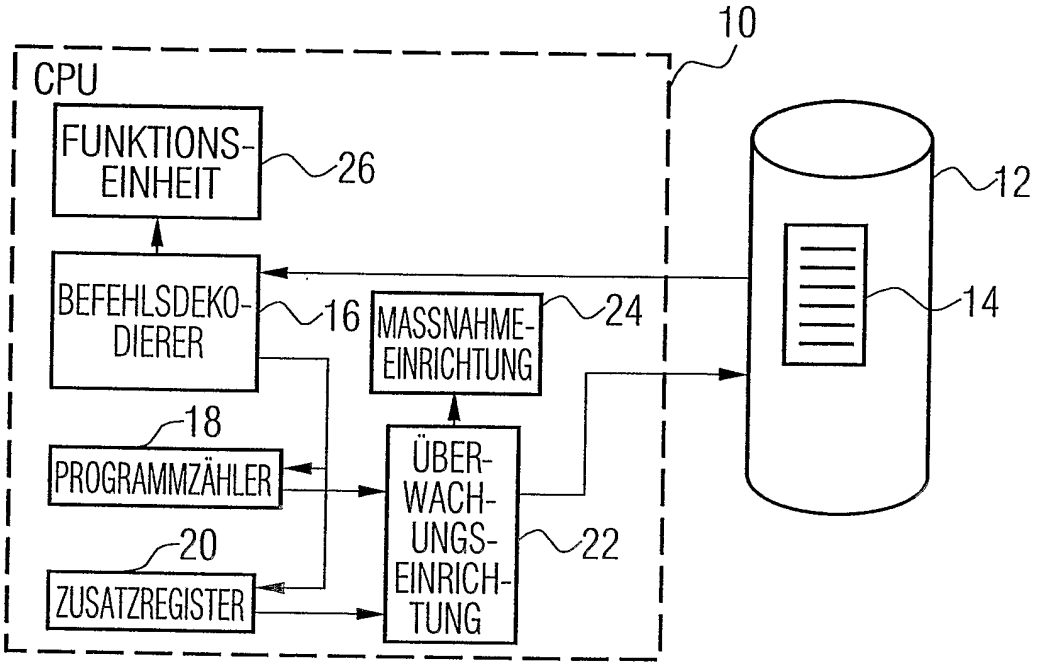
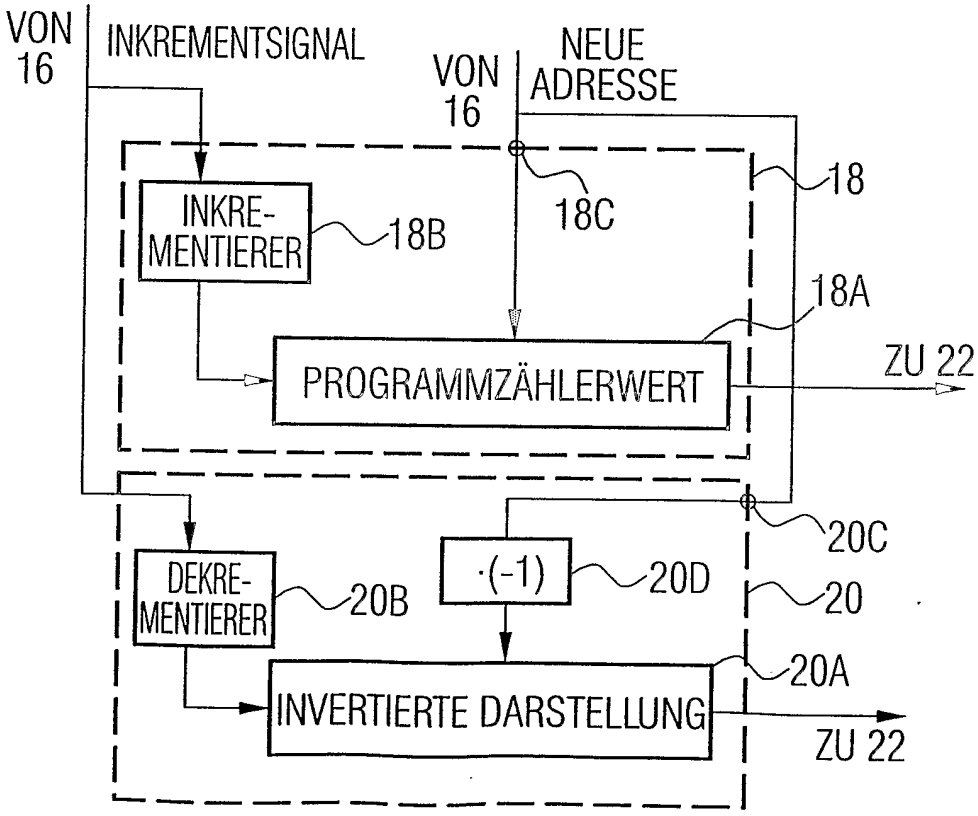

10
CPU
FUNKTIONS-EINHEIT
26
12
BEFEHLSDEKO-DIERER
16
MASSNAHME-EINRICHTUNG
24
14
18
PROGRAMMZÄHLER
ÜBER-WACH-UNGS-EINRICH-TUNG
22
20
ZUSATZREGISTER

FIG 2


VON 16
INKREMENTSIGNAL
VON 16
NEUE ADRESSE
18
INKRE-MENTIERER
18B
18C
18A
PROGRAMMZÄHLERWERT
ZU 22
20C
DEKRE-MENTIERER
20B
·(-1)
20D
20
20A
INVERTIERTE DARSTELLUNG
ZU 22