(19) **Bundesrepublik Deutschland**
Deutsches Patent- und Markenamt

(10) **DE 603 03 126 T2** 2006.07.20

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 1 499 906 B1**
(21) Deutsches Aktenzeichen: **603 03 126.9**
(86) PCT-Aktenzeichen: **PCT/US03/11399**
(96) Europäisches Aktenzeichen: **03 721 656.1**
(87) PCT-Veröffentlichungs-Nr.: **WO 2004/051294**
(86) PCT-Anmeldetag: **14.04.2003**
(87) Veröffentlichungstag der PCT-Anmeldung: **17.06.2004**
(97) Erstveröffentlichung durch das EPA: **26.01.2005**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **04.01.2006**
(47) Veröffentlichungstag im Patentblatt: **20.07.2006**

(51) Int Cl.[8]: ***G01R 31/3185*** (2006.01)
G06F 1/00 (2006.01)

(30) Unionspriorität:
135877 **30.04.2002** **US**

(73) Patentinhaber:
Freescale Semiconductor, Inc., Austin, Tex., US

(74) Vertreter:
SCHUMACHER & WILLSAU, Patentanwaltssozietät, 80335 München

(84) Benannte Vertragsstaaten:
DE, FR, GB, IT, NL

(72) Erfinder:
TKACIK, Thomas, Phoenix, AZ 85044, US; SPITTAL, E., John, Waddell, AZ 85355, US; LUTZ, Jonathan, Kitchener, Ontario N2M 5E7, CA; CASE, Lawrence, Foutain Hills, AZ 85268, US; HARDY, Douglas, Scottsdale, AZ 85259, US; REDMAN, Mark, Gilbert, AZ 85296, US; SCHMIDT, Gregory, Chandler, AZ 85226, US; TUGENBERG, Steven, Scottsdale, AZ 85258, US; FITZSIMMONS, D., Michael, Austin, TX 78731, US; CARDER, L., Darrell, Dripping Springs, TX 78620, US

(54) Bezeichnung: **VERFAHREN UND VORRICHTUNG ZUM SICHEREN SCAN-TESTEN**

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

Beschreibung

[0001] Die vorliegende Offenbarung betrifft im Allgemeinen das Prozessor-Scan-Testen und insbesondere Sicherheitsvorrichtungen für das Scan-Testen.

Hintergrund

[0002] In seiner grundlegendsten Form ist eine Scan-Kette eine Reihe von Elementen, die so miteinander verbunden sind, dass der Ausgang eines Elementes an den Eingang des nächsten Elementes in der Serie verbunden ist, das wiederum einen Ausgang besitzt, der an den Eingang eines nachfolgenden Elementes verbunden ist, und sofort. Mitunter verwenden Schaltungsdesigner Scan-Ketten, um auf interne Elemente eines Prozessors, die anderweitig nicht zugreifbar sind, einen Testzugriff zur Verfügung zu stellen. Durch das Verwenden einer Scan-Kette kann ein Prüfingenieur sequenzielle Daten unter Verwendung eines einzelnen Eingangsanschlusses Daten in den Prozessor schieben. Der Prozessor arbeitet mit den Daten und die Ergebnisse der Operationen werden dann sequenziell unter Verwendung eines einzelnen Ausganganschlusses ausgelesen. Auf diese Weise kann eine maximale Anzahl interner Schaltungen mit einem Minimum an zusätzlicher Komplexität getestet werden.

[0003] Diese Testfreundlichkeit jedoch führt zu Problemen beim Datenzugriff, die in Erwägung gezogen werden müssen, insbesondere vor dem Hintergrund der Verschlüsselungs- und Sicherheitsanforderungen im Software-, Telekommunikations-, Unterhaltungs- und weiteren Bereichen. Beispielsweise besteht in der Telekommunikationsindustrie das Erfordernis, sichere Codes in einigen der Halbleiterchips zu speichern, die zur Verarbeitung von Informationen in Mobiltelefonen, Pagern und ähnlichem verwendet werden. Diese sicheren Codes können als Teil von proprietären Datenverarbeitungsverfahren zur Hardwareidentifikation und -authentifikation eingesetzt werden, um einen sicheren Zustand zu spezifizieren, oder für eine beliebige Anzahl weiterer Verwendungsmöglichkeiten. Wenn jedoch die Schaltung, die für die Handhabung dieser Codes verantwortlich ist, über eine Scan-Kette zugänglich ist, kann es Mitbewerbern möglich sein, die Scan-Kette dazu auszunützen, um Zugriff auf die sicheren Codes zu erlangen, die in dem Chip gespeichert sind, oder um in einen sicheren Zustand einzutreten.

[0004] Um das Problem, die Sicherheitskette dazu auszunützen, um einen Zugriff auf sichere Informationen zu erlangen, die in dem Chip gespeichert sind oder dem Chip vorzugaukeln, er wäre in einem sicheren Zustand, anzugehen, haben die Hersteller im Allgemeinen die Schaltungen von der Sicherheitskette entfernt, die dazu verwendet werden, sichere Informationen zu verarbeiten. Indem diese Schaltungen von der Sicherheitskette entfernt wurden, ist es für den nicht autorisierten Benutzer schwieriger, Zugriff auf die sicheren Codes zu erlangen. Mit dieser Lösung jedoch ist ein wesentlicher Teil des Chips für ein komplettes Testen unzugänglich.

[0005] Die US 5,898,776 A offenbart ein Sicherheits-Antifuse, das ein Auslesen von sensiblen Daten während des Scan-Testens verhindert. Die US 5,530,749 A offenbart ein Verfahren zum sicheren Konfigurieren einer Hardware unter Verwendung eines verschlüsselten Codes.

[0006] Wie aus der obigen Diskussion ersichtlich sein sollte, sind die momentan verfügbaren Testverfahren nicht ideal, da sie einen Designer dazu zwingen, entweder einen Testzugriff mit einer reduzierten Datensicherheit oder eine Datensicherheit ohne einen Testzugriff für wesentliche Teile eines Datenprozessors zu wählen. Es wird demnach ein Weg benötigt, einen Testzugriff auf Teile eines Prozessors zu erlauben, die sichere Informationen verarbeiten, aber gleichzeitig die Geheimhaltung jeglicher sicherer Informationen in dem Prozessor aufrechtzuerhalten. Eine Lösung gemäß der vorliegenden Erfindung ist in dem kennzeichneten Abschnitt der unabhängigen Ansprüche offenbart.

Kurze Beschreibung der Zeichnungen

[0007] Verschiedene Vorteile, Eigenschaften und Charakteristiken der vorliegenden Offenbarung sowie Verfahren, Betriebs- und Funktionsweise von verbunden Strukturelementen sowie die Kombination von Teilen und Herstellungsvorteilen werden bei der Betrachtung der folgenden Beschreibung und in den Ansprüchen unter Bezugnahme auf die begleitenden Zeichnungen offensichtlich werden, von denen alle einen Teil dieser Beschreibung bilden.

[0008] Fig. 1 ist ein Blockdiagramm eines Prozessors, der einen Scan-Controller gemäß einer Ausführungsform der vorliegenden Offenbarung verwendet;

[0009] Fig. 2 ist ein Flussdiagramm, das ein Verfahren zum Scan-Testen veranschaulicht, das das Löschen von sensiblen Daten vor dem Erlauben eines Zugriffs auf scan-beobachtbare Teile eines Prozessors umfasst, gemäß einer Ausführungsform der vorliegenden Offenbarung;

[0010] Fig. 3-Fig. 5 sind Logikdiagramme, die Scan-Controller zum Steuern des Zugriffs auf eine Scan-Kette veranschaulichen, gemäß verschiedener Ausführungsformen der vorliegenden Offenbarung;

[0011] Fig. 6 ist ein beispielhaftes Zeitdiagramm, das die Zeitgebung des Logikdiagramms, das in Fig. 3 dargestellt ist, in Verbindung mit dem Eintreten in einen Testmodus veranschaulicht;

[0012] Fig. 7 ist ein beispielhaftes Zeitdiagramm, das die Zeitgebung des in Fig. 3 gezeigten Logikdiagramms in Verbindung mit einem Verlassen eines Testmodus veranschaulicht;

Detaillierte Beschreibung der Figuren

[0013] In der folgenden detaillierten Beschreibung der Figuren werden die Begriffe "bestätigen" und "negieren" (oder "ent-bestätigen") verwendet, wenn auf das Setzen eines Signals, eines Statusbits oder ähnlicher Vorrichtungen, auf dessen logischen Wahr- beziehungsweise logischen Falschzustand Bezug genommen wird. Wenn der logische Wahrzustand ein logischer Pegel eins ist, ist der logische Falschzustand ein Pegel null. Wenn der logische Wahrzustand ein logischer Pegel null ist, dann ist der logische Falschzustand ein logischer Pegel eins.

[0014] Demnach kann jedes Signal, das hierin beschrieben wird, als logisch positiv oder negativ ausgelegt sein, wobei logisch negativ mittels eines Balken über dem Signalnamen oder mittels eines Sterns (*), das dem Namen folgt, angezeigt werden. In dem Fall eines logisch negativen Signals ist das Signal Active Low, wenn der logische Wahrzustand einem logischen Pegel null entspricht. In dem Fall eines logisch positiven Signals ist das Signal Active High, wenn der logische Wahrzustand einem logischen Pegel eins entspricht.

[0015] Die Fig. 1–Fig. 7 veranschaulichen, wie ein Scan-Controller in einem Informationsprozessor verwendet werden kann, um einen Schutz vor einem elektronischen Eindringen durch das Verhindern eines Zugangs auf sensible Informationen über Prozessortestmodi zur Verfügung stellt. Die Weise, auf die die Sicherheit aufrechterhalten wird, ermöglicht eine vergrößerte Testabdeckung für die Komponente, ohne die Sicherheit von sensiblen Informationen zu opfern. Diese erhöhte Testabdeckung sollte wiederum ein weniger teueres Produkttesten und einen schnelleren Marktzugang ermöglichen.

[0016] Um die Sicherheit sensibler Daten zu erhalten, löscht eine hierin beschriebene Ausführungsform lesesensible Sicherheitsdaten von scan-beobachtbaren Abschnitten des Prozessor vom dem Ermöglichen eines Zugriffs auf die Scan-Kette und löscht schreibsensible Sicherheitsdaten vor einem Verlassen des Testmodus und dem Aufnehmen des normalen Betriebs. Das Löschen sensibler Abschnitte der Scan-Kette zu diesen Zeitpunkten verhindert, dass nicht autorisiertes Personal auf einfache Weise sichere Daten abscannen und Elemente auf der Scan-Kette vor dem normalen Betrieb vorladen kann, um sensible Statusinformationen einzustellen.

[0017] Unter anfänglicher Bezugnahme auf Fig. 1 wird ein Prozessor erläutert, der einen Scan-Controller gemäß den hier im Folgenden dargelegten Lehren einsetzt und der im Allgemeinen als Prozessor **100** bezeichnet wird. Der Prozessor **100** umfasst eine Reihe von Signalspeichern **128** - **188** und einen Zustandsautomaten **150**, die auf der Scan-Kette **180** liegen; verschiedene Quellen von sensiblen Informationen, wie etwa ein sicherer Schlüssel **130** und ein sicherer RAM ("RAM = Random Access Memory"/Speicher mit wahlfreiem Zugriff) **140**; sowie einen Scan-Controller **120**, der den Zugriff auf die Scan-Kette **180** steuert und zurücksetzt und/oder Moduskonfigurationsignale zur Verfügung stellt. Der Prozessor **100** umfasst auch einen Verschlüsselungsblock **110** zum Verschlüsseln von Benutzerdaten und einen RAM-Reset **170**, um Informationen von dem sicheren RAM **140** zu löschen. Es sei bemerkt, dass lediglich bestimmte Elemente als auf der Scan-Kette **180** liegend gezeigt sind und dass beliebige Elemente, für die ein Scan-Testen erwünscht ist, beispielsweise der Verschlüsselungsblock **110**, ebenfalls auf der Scan-Kette **180** angeordnet werden können.

[0018] Die vor einem Zugriff zu schützende Information kann Informationen umfassen, die in den Prozessor **100** während der Herstellung gespeichert wurden, wie etwa fest verdrahtete Identifikationsschlüssel und proprietäre hardware/firmwareimplementierte Algorithmen oder sensible Informationen, die nach der Herstellung gespeichert wurden. Beispielsweise kann der sichere Schlüssel **130** ein Hardwareidentifikationsschlüssel sein, der zum Identifizieren einer bestimmten Vorrichtung zur mobilen Kommunikation verwendet wird und der Zustandsautomat **150** kann eine Serie logischer Elemente sein, die von dem Prozessor **130** dazu verwendet wird, zu bestimmen, ob der Prozessor sich in einem sicheren Betriebsmodus befindet. In jedem dieser Fälle muss die in den Prozessor **100** eingebaute Information sicher bewahrt werden, um von einem Fälschen abzuschrecken oder um Reverse Engineering durch Mitbewerber schwieriger zu gestalten.

[0019] Obwohl der sichere Schlüssel **130** während der Herstellung implementiert werden kann, ist der sichere RAM **140** ein Weg, um die Speicherung sicherer Informationen in den Prozessor **100** nach dem Abschluss des Herstellungsprozesses zu implementieren. Es sei beispielsweise angenommen, dass der Prozessor **100** ein Graphikprozessor ist, der in einer drahtlosen Internetanwendung verwendet wird. Wenn ein bestimmter Serviceprovider einen proprietären Graphikkompressionsalgorithmus besitzt, kann der Provider den verschlüsselten Algorithmus in den Prozessor **100** über den Benutzerdateneingang **109** laden. Der Prozessor **100** würde dann den Algorithmus unter Verwendung des Verschlüsselungsblocks **100** entschlüsseln und die entschlüsselten Daten zur Speicherung in den sicheren RAM **140** weiterleiten. Es sollte klar sein, dass geeignete Verfahren zum

Speichern von Informationen in dem Prozessor **100** zusätzlich oder anstatt derjenigen, die in Fig. 1 veranschaulicht sind, von dem Fachmann eingesetzt werden können, ohne von den Lehren, die hierin dargelegt werden, abzuweichen.

[0020] Die Signalspeicher **182**, **184**, **186** und **188** sind in der Lage, sowohl in einem normalen Modus als auch in einem Testmodus zu funktionieren. Im normalen Modus halten die Signalspeicher **182** und **184** sowie der Zustandsautomat **150** sensible Informationen zur Verwendung anderer Abschnitte des Prozessors **100**. Beispielsweise kann der Zwischenspeicher **182** einer aus einer Anzahl von Zwischenspeichern sein, die dazu verwendet werden, auf den sicheren Schlüssel **130** zuzugreifen und den sicheren Schlüssel **130** an einen Authentifizierungsabschnitt (nicht abgebildet) des Prozessors **100** zu liefern. Als ein weiteres Beispiel kann eine verschlüsselte Softwareunterroutine von dem sicheren RAM **140** an eine zentrale Verarbeitungseinheit über einen Signalspeicher **184** geleitet werden. Wenn die Zwischenspeicher **182** oder **184** Informationen enthalten, auf die nicht ohne eine geeignete Befugnis zugegriffen werden kann, spricht man davon, dass die Signalspeicher schreibsensible Informationen enthalten.

[0021] Der Zustandsautomat **150** kann Daten enthalten, die den Prozessor **100** in einen nicht sicheren Modus bringen. Wenn die Zustandsdaten in dem Zustandautomaten **150** gerade vor dem Verlassen eines Scanmodus geändert werden könnten, könnte der Prozessor dahingehend überlistet werden, dass er annimmt, dass er sich in einem nicht sicheren Zustand befindet, dabei könnte möglicherweise der sichere Betrieb beeinträchtigt werden. Daten, die davor geschützt werden müssen, nach einem Scanmodusbetrieb gespeichert zu werden, werden als schreibsensible Daten bezeichnet. Andere Signalspeicher (nicht abgebildet) können zum Speichern von Ausgaben anderer Zustandsautomaten (nicht abgebildet) verwendet werden, die lese- oder schreibsensible Informationen enthalten. In jedem dieser Beispiele könnte die Sicherheit der Daten beeinträchtigt werden, wenn der Zugriff auf die Scan-Kette nicht geschützt wird.

[0022] Im Testmodus sind die Signalspeicher **182**, **184**, **186** und die mit dem Zustandsautomaten **150** in Bezug stehenden Signalspeicher außerhalb des Prozessors **100** über die Scan-Kette **180** beobachtbar. Der Zugriff auf die Scan-Kette **180** wird über einen Scan-Eingangsanschluss **181** und über einen Scan-Ausgangsanschluss **189** zur Verfügung gestellt. Die Daten werden in den Signalspeicher **182**, dem ersten scan-beobachtbaren Signalspeicher der Signalkette **180** über den Scan-Eingangsanschluss **181** getaktet. Jedes Mal, wenn Daten in den Signalspeicher **182** getaktet werden, werden an dem Signalspeicher **182** Ausgangsdaten an den Eingang des Signalspeichers **148** gesendet. Jedes Mal, wenn die Ausgangsdaten des Signalspeichers **182** an den Eingang des Signalspeichers **184** gesendet werden, werden die Ausgangsdaten am Signalspeicher **184** an den Eingang des Signalspeichers **186** gesendet, und so weiter, bis die Daten den ganzen Weg durch die Kette zu dem Scan-Ausgangsanschluss **198** gewandert sind. Es sei beispielsweise angenommen, dass in der veranschaulichten Scan-Kette **180** eine logische 1 in den Signalspeicher **182** während des ersten Taktzyklus getaktet wird. Während des zweiten Taktzyklus wird die in dem Signalspeicher **182** gespeicherte logische 1 an den Signalspeicher **184** geliefert. Während des dritten Taktzyklus würde die gleiche logische 1 an den Signalspeicher **168** gesendet. Der Vorgang würde sich fortsetzen, bis schließlich die logische 1 an den Auslesesignalspeicher **188** übertragen ist und zum Auslesen an dem Scan-Ausgansanschluss **189** während des vierten Taktzyklus zur Verfügung steht. Der Fachmann wird erkennen, dass dieses einfache Beispiel lediglich veranschaulichend ist, und dass Daten, die in einem bestimmten Signalspeicher geschoben wurden, auf verschiede Weisen manipuliert werden können, bevor sie durch den verbleibenden Rest der Scan-Kette **180** geschickt werden.

[0023] In der veranschaulichten Ausführungsform enthält der Auslesesignalspeicher **188** im Gegensatz zu den Signalspeichern **182**, **184** und dem Zustandsautomaten **150** während eines normalen Modus keine sensiblen Daten. Stattdessen blockiert der Auslesesignalspeicher **188** unter der Steuerung des Scan-Controllers **120** das Auslesen von Daten aus der Scan-Kette, außer gewisse vorherbestimmte Bedingungen liegen vor. Es sollte klar sein, dass, obwohl nicht veranschaulicht, ein Signalspeicher, der auf eine Weise ähnlich der des Auslesesignalspeichers **188** gesteuert wird, dazu verwendet werden könnte, zu blockieren, dass an dem Eingang der Scan-Kette **180** beliebige Daten eingescannt werden. Es sollte ebenfalls klar sein, dass in anderen Ausführungsformen, wie etwa in verschiedenen Ausführungsformen, die im Folgenden erläutert werden, der Auslesesignalspeicher **188** nicht verwendet wird.

[0024] Der Scan-Controller **120** steuert den Zugriff auf die Scan-Kette **180** und steuert folglich den Zugriff auf beliebige sensible Informationen, die in dem Signalspeicher **182**, **184** und dem Zustandsautomaten **150** gespeichert sein können. Bei zumindest einer Ausführungsform empfängt der Scan-Controller **120** als Eingang ein TEST-MODE-Signal, ein SCAN-ENABLE-Signal, ein RESET-Signal und ein EVENT-TRIGGER-Signal. Unter Verwendung dieser Eingangssignale erzeugt der Scan-Controller **120** ein SCAN-ENABLE-(INTERNAL)-Signal und ein SCAN-DATA-ENABLE-Signal, die dazu verwendet werden, die Signalspeicher **182**–**188** und den Zu-

standsautomaten **150** zu konfigurieren, um ein Scan-Testen zu ermöglichen. Beispielsweise platziert ein bestätigtes SCAN-EANBLE-(INTERNAL) jeden Scan-Signalspeicher in einem Scanmodus, während ein bestätigtes SCAN-DATA-ENABLE es Daten ermöglicht, zu dem Scan-Ausgangsanschluss **189** gescannt zu werden. Der Scan-Controller **120** erzeugt ebenfalls SCAN-EXIT- und SCAN-RESET-Signale, die dazu verwendet werden, Elemente nach Bedarf in der Scan-Kette **180** zurückzusetzen.

[0025] In der veranschaulichten Ausführungsform steuert der Scan-Controller **120** die meisten Zurücksetzungssequenzen, die mit der Scan-Kette **180** in Beziehung stehen, wodurch sichergestellt wird, dass jeder der Signalspeicher **182**–**188** und der Zustandsautomat **150** nach Bedarf richtig zurückgesetzt werden. Es sei bemerkt, dass in der veranschaulichten Ausführungsform die Zwischenspeicher **186** und **188**, die nicht dazu verwendet werden, sensible Informationen zu speichern, nicht zurückgesetzt werden müssen, um sensible Informationen zu schützen. Da es jedoch wünschenswert sein kann, die Zwischenspeicher **186** und **188** während eines "harten" Zurücksetzens oder während anderer Zeitpunkte zurückzusetzen, wird der RESET-Eingang dem Scan-Controller **120** zur Verfügung gestellt, um die Signalspeicher **186**, **188** zurückzusetzen. In weiteren Ausführungsformen kann das RESET-Signal dazu zur Verfügung gestellt werden, um Scan-Kettenelemente, beispielsweise den Zustandsautomaten **150**, zusätzlich zu dem SCAN-RESET-Signal zu sichern. Wenn es unter vielen Umständen wünschenswert sein kann, jedes Element in der Scan-Kette zurückzusetzen, können nicht sensible Elemente durch die Ausgaben der Scan-Controllers **120** nicht zurückgesetzt verbleiben, ohne dass von den hier dargelegten Lehren abgewichen wird.

[0026] Der RAM-Reset **170** wird in einer Ausführungsform dazu verwendet, um Informationen von einem sicheren RAM **140** als Antwort auf bestimmte Ereignisse zu löschen. Der RAM-Reset **170** kann durch einen separaten Reset-Zustandsautomaten (nicht abgebildet) direkt über den Scan-Controller **120** oder auf andere Weise gesteuert werden. Der RAM-Reset **170** kann auch ein EVENT-TRIGGER-Signal zur Verfügung stellen, das anzeigt, dass Daten in dem sicheren RAM **140** erfolgreich gelöscht wurden. Dieses Ausgangssignal könnte als EVENT-TRIGGER-Eingang für den Scan-Controller **120** verwendet werden. Die Verwendung des EVENT-TRIGGER-Signals kann insbesondere dann nützlich sein, wenn die Zeit, die dazu benötigt wird, um den sicheren RAM **140** zurückzusetzen, unbestimmt ist. Es sollte klar sein, dass obwohl der RAM-Reset **170** in der veranschaulichten Ausführungsform verwendet wird, dies nicht für jede Ausführungsform notwendig ist. In zumindest einer Ausführungsform wird die Erzeugung eines SCAN-DATA-ENABLE-Signals, eines SCAN-ENABLE-(INTERNAL)-Signals und eines EVENT-TRIGGER-Signals oder andere ähnliche Signale zumindest teilweise von einem Signal (nicht in Fig. 1 gezeigt) gesteuert, dass anzeigt, ob Daten, die in den Elementen der Scan-Kette **180** enthalten sind, mittels des Durchführens eines Resets oder auf andere Weise gesichert wurden. Eine Ausführungsform eines derartigen Signals ist das UNSECURE*-Signal, das nachfolgend unter Bezugnahme auf Fig. 5 erläutert wird.

[0027] Unter nachfolgender Bezugnahme auf Fig. 2 wird ein Verfahren zum Scan-Testen eines Prozessors wie etwa eines Prozessors **100** (Fig. 1) gemäß einer Ausführungsform der vorliegenden Offenbarung erläutert. Das Verfahren beginnt mit dem Schritt **210**, in dem der Prozessor **100** in einem normalen oder Nicht-Test-Modus arbeitet. Im normalen Modus werden Elemente der Scan-Kette **180** beim Durchführen gewöhnlicher Verarbeitungsaufgaben verwendet. Während sich die Elemente der Scan-Kette **180** in einem normalen Modus befinden, kann auf sie über den SCAN-IN-Anschluss **188** oder über den SCAN-OUT-Anschluss **189** nicht zugegriffen werden, da die Signalspeicher **182**–**188** und der Zustandsautomat **150** nicht dahingehend konfiguriert sind, dass sie Informationen über ihre Scan-Ketten-Anschlüsse empfangen oder senden können. Im normalen Modus können die Signalspeicher **182**, **184** und der Zustandsautomat **150** sensible Daten oder Zustandsinformationen enthalten, sodass, wenn die Elemente der Scan-Kette **180** zu einem Scan-Ketten-Zugriff während normaler Operationen freigegeben werden könnten, beliebige Informationen, die in den Elementen der Scan-Kette enthalten sind, aus dem Scan-Ausgangsanschluss **189** ausgelesen werden könnten, wodurch möglicherweise die Sicherheit der Informationen beeinträchtigt werden könnte.

[0028] Das Verfahren fährt mit Schritt **220**, in dem die Scan-Kette zum Testen darauf vorbereitet wird, dass sensible Daten in den Scan-Ketten-Signalspeicher **182**, **184** und dem Zustandsautomaten **150** als Antwort auf eine gewünschte Eingabe oder eine Kombination von Eingaben zurückgesetzt oder auf andere Weise modifiziert werden. Beispielsweise bewirkt in einer Ausführungsform der Empfang eines bestätigten TEST-MODE-Signals und eines bestätigten SCAN-ENABLE-Signals, dass der Scan-Controller **120** ein bestätigtes SCAN-RESET-Signal erzeugt, das direkt auf die Reset-Anschlüsse der Signalspeicher **182**, **184** und des Zustandsautomaten **150** gelegt werden kann. Alternativ könnte ein geeigneter Hardware-, Software- oder Firmwarecontroller die Daten in dem Signalspeicher **182**, **184** und dem Zustandsautomaten **150** zufällig oder auf eine andere Weise modifizieren, um sicherzustellen, dass keine sicheren Daten aus dem Signalspeicher ausgelesen werden können.

[0029] In Schritt **230** prüft der Scan-Controller, ob irgendwelche sensiblen Daten gelöscht wurden oder auf andere Weise modifiziert wurden. In Schritt **230** kann das Vorhandensein eines bestätigten EVENT-TRIGGER-Signal an dem Eingang zu dem Scan-Controller **120** überprüft werden und, wenn das EVENT-TRIGGER-Signal nicht bestätigt ist, dann das SCAN-ENABLE-(INTERNAL)-Signal nicht bestätigt werden. Wenn es beispielsweise erwünscht ist, den sicheren RAM **140** zurückzusetzen, bevor ein Zugang auf die Scan-Kette **180** erlaubt wird, könnte der Scan-Controller **120** auf ein Signal von dem RAM-Reset **170** warten, das anzeigt, dass ein Reset des sicheren RAM **140** abgeschlossen wurde. In anderen Ausführungsformen wird kein EVENT-TRIGGER-Signal benötigt, da die Zeitgebung für die Modifikation von Daten in Scan-Ketten-Elementen deterministisch ist, und Schritt **230** wird dadurch ausgeführt, dass einfach eine Anzahl von Taktzyklen gewartet wird, die ausreicht, um ein Zurücksetzen der Signalspeicher **182**–**184** zu ermöglichen.

[0030] Wenn in Schritt **230** die sensiblen Daten einmal modifiziert wurden, erlaubt der Scan-Controller **120** Zugriff auf die Scan-Kette **180** in Schritt **240**. Während des Schritts **240** können die dem Fachmann bekannten Scan-Testprozeduren durchgeführt werden, ohne die Sicherheit von sensiblen Informationen zu opfern, die vorhergehend in einem beliebigen der scan-beobachtbaren Elemente der Scan-Kette **180** gespeichert wurden. Es können Daten in den SCAN-IN-Anschluss **181** eingescannt werden und von dem SCAN-OUT-Anschluss **189** zum Test der Funktionalität von verschiedenen internen Abschnitten des Prozessors **100** ausgelesen werden.

[0031] Wenn das Scan-Testen abgeschlossen ist, setzt sich das Verfahren der Fig. 2 von Schritt **240** zu Schritt **250** fort. In Schritt **240** werden Vorkehrungen getroffen, um den Scan-Test-Modus zu verlassen und den normalen Modus wieder einzunehmen. In einer Ausführungsform wird über den Schritt **250** der Zugriff auf die Scan-Kette **180** blockiert und beliebige Daten in den Signalspeicher **182**, **184** und in dem Zustandsautomaten **150** modifiziert oder zurückgesetzt. Die Scan-Kette **150** kann durch das Benachrichtigen des Scan-Controllers **120** (Fig. 1) blockiert werden, um einen normalen Modus durch das TEST-MODE-Signal einzunehmen. Als Antwort auf das Ent-Bestätigen des TEST-MODE-Signals kann ein SCAN-RESET-SIGNAL bestätigt werden, um die Zwischenspeicher **182**, **184** und den Zustandsautomaten **150** (Fig. 1) und beliebige andere Elemente der Scan-Kette **180** bestätigt werden, die sensible Informationen lesen oder schreiben. Zusätzlich können Elemente des beobachtbaren Abschnitts der Scan-Kette neu konfiguriert werden, um zu verhindern, dass Daten ausgescannt werden. Die Signale, die dazu verwendet werden, um die Scan-Kette für einen normalen Betrieb vorzubereiten, werden detaillierter unter Bezugnahme auf Fig. 7 beschrieben.

[0032] Das Löschen von Informationen von scan-beobachtbaren Abschnitten des Prozessors **100** vor dem Verlassen eines Nicht-Testzustandes vermeidet, dass jemand "Keim"-Informationen während eines Scan-Tests einscannt und dann die Ausgänge des Prozessors **100** überwacht, um zu bestimmen, welche Operationen auf den Keiminformationen durchgeführt wurden. Das Löschen der Informationen an diesem Punkt vermeidet ebenso, dass jemand einen Zustandsautomaten, beispielsweise den Zustandsautomaten **150** auf einen bestimmten Zustand setzt, was beispielsweise den Prozessor dahingehend "überlisten" könnte, dass er annimmt, er würde in einem sicheren Zustand arbeiten, was tatsächlich nicht der Fall ist. In zumindest einer Ausführungsform kann ein SCRN-EXIT-Signal, das während des Schritts **250** erzeugt wurde, als Eingabe für verschiedene Zustandsmaschinen verwendet werden, um anzuzeigen, dass der augenblickliche Zustand nicht richtig sein kann, anstatt oder zusätzlich zum Modifizieren/Zurücksetzen der Daten, wie in dem vorhergehenden Absatz erläutert. Die Zustandsmaschinen können dann selbst in einen bekannten Zustand übergehen, auch wenn die Zustandsbits nicht beim Verlassen des Scans gelöscht sind.

[0033] In Schritt **260** überprüft der Scan-Controller **120**, um sicherzustellen, dass die Daten von beliebigen notwendigen scan-beobachtbaren Abschnitten des Prozessor **100** unter Verwendung der gleichen oder ähnlichen Techniken gelöscht werden, wie diejenigen, die in Bezug auf Schritt **230** erläutert wurden. Beispielsweise kann in einer Ausführungsform der Scan-Controller **120** auf eine Bestätigung eines EVENT-TRIGGER-Signals warten, um anzuzeigen, dass ein Zurücksetzten abgeschlossen wurde, bevor er ein SCAN-DATA-ENABLE-Signal bestätigt, das dazu verwendet werden kann, um ein Verschieben von Daten in oder aus der Scan-Kette **180** zu erlauben oder zu verhindern. In einer anderen Ausführungsform, die detaillierter nachfolgend unter Bezugnahme auf Fig. 5 erläutert werden wird, kann die Ent-Bestätigung eines UNSECURE*-Signals auf die Bestätigung eines EVENT-TRIGGER-Signals gegründet werden.

[0034] Das UNSECURE*-Signal, das unter Bezugnahme auf Fig. 5 erläutert werden wird, kann dazu verwendet werden, ein oder mehrere Elemente auf der Scan-Kette **180** zu steuern, um eine Konfiguration verschiedener Elemente der Scan-Kette **180** für normale Operationen zu verhindern oder zu erlauben. Beispielsweise kann UNSECURE* anstelle eines Steuersignals, wie etwa das in Fig. 1 veranschaulichte SCAN-DATA-ENABLE-Signal verwendet werden, um zu verhindern, dass Daten ausgegeben werden. Alternativ kann UNSECURE* als eine Eingabe für eine logische Schaltung verwendet werden, die

dazu verwendet wird, eines oder mehrere Signale zu erzeugen, beispielsweise SCAN-DATA-ENABLE oder SCAN-ENABLE-(INTERNAL). In zumindest einer Ausführungsform kombiniert das UNSECURE*-Signal (Fig. 5) die Funktionalität sowohl des SCAN-RESET- und SCAN-EXIT-Signals (Fig. 3–Fig. 4). Wenn die Daten und/oder Statusinformationen einmal gelöscht sind, können der Scan-Controller **120** und andere geeignete Hardware-, Software- oder Firmwareelemente den Prozessor **100** in den normalen Modus zurückführen.

[0035] Es sollte klar sein, dass die verschiedenen Schritte des Verfahrens der Fig. 2 gleichzeitig oder in einer unterschiedlichen Reihenfolge implementiert werden können, ohne von den hierin dargelegten Lehren abzuweichen. Beispielsweise kann das Überprüfen, um nachzusehen, dass sensible Daten von scan-beobachtbaren Elementen gelöscht wurden, wie in Schritt **230**, während des normalen Modus **210** durchgeführt werden, zusätzlich zu einem Durchführen nach dem Schritt **230**. Alternativ kann das Überprüfen, ob sensible Daten gelöscht wurden, kontinuierlich durchgeführt werden. Zusätzlich können in Abhängigkeit von verschiedenen Design-, Marketing-, Kosten-, Sicherheits- oder anderen Faktoren bestimmte Abschnitte des in Fig. 2 beschriebenen Verfahrens ausschließlich in andere Abschnitte implementiert werden. Beispielsweise können beim Eintreten in einen Testmodus ausschließlich Daten in bestimmten Signalspeichern gelöscht werden, wie in Schritt **220**, und die Schritte **250** und **260** können lediglich dazu verwendet werden, um Statusinformationen von sicheren Zustandsautomaten zu verlassen und für ein Eintreten in einen normalen Zustand verwendet werden.

[0036] Unter Bezugnahme auf Fig. 3 wird eine bestimmte Ausführungsform eines Abschnittes des Scan-Controllers **120** erläutert. In der veranschaulichten Ausführungsform weist der Abschnitt des Scan-Controllers **120** drei Eingänge auf: TEST-MODE, RESET und SCAN-ENABLE. Diese drei Eingänge werden in Kombination verwendet, um drei Ausgaben zu erzeugen: SCAN-ENABLE-(INTERNAL), SCAN-RESET und SCAN-EXIT. TEST-MODE ist ein Signal, das dazu verwendet wird, Schaltungen funktional zu modifizieren, um sie leichter testbar zu machen. Es wird bestätigt, bevor das Scan-Testen beginnt. SCAN-ENABLE ist ein Signal, das dazu verwendet wird, um Daten in die Scan-Kette zu schieben und RESET ist ein Signal, das anzeigt, dass interne Daten gelöscht, auf einen bekannten Zustand gesetzt und/oder auf andere Weise modifiziert werden müssen. SCAN-ENABLE-(INTERNAL) ist eine Gatter-Version von SCAN-ENABLE, die von internen Schaltungen anstelle von SCAN-ENABLE verwendet wird, um Daten durch die Scan-Kette zu schieben. SCAN-RESET kann, wenn es bestätigt ist, dazu verwendet werden, um sensible Daten von scan-beobachtbaren Elementen der Scan-Kette zu löschen. SCAN-EXIT ist, wenn es bestätigt ist, ein Signal, das anzeigt, dass der Testmodus, wie er von dem TEST-MODE-Signal angezeigt wird, ent-bestätigt wurde und kann als ein Eingabe für einen Zustandsautomaten verwendet werden, um einen Zustandsübergang in einen bekannten Zustand zu erzwingen.

[0037] Ein Abschnitt des Scan-Controllers **120** stellt sicher, dass keine Daten in die Scan-Kette geschoben werden können, bis Informationen in einer sensiblen Schaltung gelöscht wurden. Beispielsweise können die Daten nicht in die Scan-Kette geschoben werden, bis das TEST-MODE-Signal bestätigt ist, was anzeigt, dass die kontrollierte Schaltung sich in einem Testmodus befindet. Darüber hinaus verzögert ein Abschnitt des Scan-Controllers **120** das Verschieben von Informationen in die Scan-Kette, bis zwei Taktzyklen nach dem TEST-MODE-Signal bestätigt wurden. Dies stellt sicher, dass der Scan-Controller **120** Zeit gefunden hat, den SCAN-RESET-Puls zu erzeugen, der automatisch auf die Bestätigung des TEST-MODE-Signals erzeugt wird. Wenn der Testmodus verlassen wird, die durch die Ent-Bestätigung des TEST-MODE-Signals angezeigt wird, wird das SCAN-EXIT-Signal bestätigt. Die Zeitgebung der Signale wird detaillierter nachfolgend unter Bezugnahme auf die Fig. 6 und Fig. 7 erläutert.

[0038] Unter nachfolgender Bezugnahme auf Fig. 4 wird eine weitere Ausführungsform eines Abschnitts des Scan-Controllers **120** gezeigt. Die in der veranschaulichten Ausführungsform gezeigte Logik ist im Wesentlich die gleiche, wie sie unter Bezugnahme auf Fig. 3 beschrieben wurde, mit der Ausnahme, dass die Bestätigung des SCAN-ENABLE-(INTERNAL) jetzt auf der Bestätigung eines EVENT-TRIGGER-Signals gründet. Es sollte klar, dass der in Figur gezeigte Scan-Controller dahingehend modifiziert werden kann, um die durch den in Fig. 4 gezeigten Scan-Controller bereit gestellte Funktionalität wie gewünscht zu enthalten. Das EVENT-TRIGGER-Signal, wie unter Bezugnahme auf Fig. 1 erläutert, wird als zusätzliche Steuerung verwendet, um sicherzustellen, dass Informationen in der kontrolliert werdenden Schaltung modifiziert, zurückgesetzt, gelöscht, etc. wurden und dass alle oder bestimmte Elemente der Scan-Kette gelöscht, eingestellt oder auf andere Weise bereitgemacht wurden, um in den Testmodus einzutreten. Lediglich beim Empfang des EVENT-TRIGGER-Signals wird das Ausgangssignal SCAN-ENABLE-(INTERNAL) erzeugt. Das SCAN-ENABLE-(INTERNAL)-Signal kann, wie in Fig. 3 erläutert, unter anderem dazu verwendet werden, ein Eingangs- oder Ausgangsgatter, ein Flip-Flop, etc. zu steuern, um zu verhindern, dass Daten in die oder aus der Scan-Kette geschoben werden.

[0039] Unter nachfolgender Bezugnahme auf Fig. 6

wird eine Ausführungsform betrachtet, die einen Abschnitt des Scan-Controllers **120** umfasst, der asynchrone Flip-Flops einsetzt. Die in Fig. 5 veranschaulichte Ausführungsform arbeitet so, dass im Wesentlichen die gleichen Auswirkungen wie die der in den Fig. 3 und Fig. 4 veranschaulichten Ausführungsformen erreicht werden, allerdings unter Verwendung einer leicht unterschiedlichen Logikkonfiguration. Zusätzlich veranschaulicht Fig. 5 eine Logik, die die Gatter **510** und **520** enthält, die Teil eines modifizierten Scan-Controllers oder einer weiteren Schaltung in Fig. 1 sein kann.

[0040] Unter Bezugnahme auf die Veranschaulichung der Fig. 5 wird das TEST-MODE-Signal in den Takteingang des Flip-Flops **540** und in den invertierten Takteingang des Flip-Flops **550**, sodass die ansteigende Flanke des TEST-MODE-Signals bewirkt, dass der Ausgang des Flip-Flops **540** auf High und die fallende Flanke des TEST-MODE-Signals bewirkt, dass der Ausgang des Flip-Flops **540** auf High gesetzt wird.

[0041] Demnach bewirkt ein beliebiger Übergang von dem TEST-MODE weg, unter der Annahme, dass die RESET-Eingänge der Flip-Flops ent-bestätigt sind, dass UNSECURE* bestätigt wird. Wenn UNSECURE* bestätigt wird, wird SCAN-ENABLE-INTERNAL ent-bestätigt, was den Betrieb der Scan-Kette verhindert. Die zurückgesetzten Eingänge der Flip-Flops **540** und **550**, die asynchron sind, werden mit dem invertierenden Ausgang des Flip-Flops **560** verbunden, sodass die Flip-Flops **540** und **550** als Antwort auf ein SECURE-RESET-Signal zurückgesetzt werden. SECURE-RESET kann als Teil eines Systemresets erzeugt werden, als Antwort auf eine Benutzeraktion, oder auf andere Weise. In zumindest einer Ausführungsform ist SECURE-RESET ein bestimmtes Beispiel des in Fig. 1 veranschaulichten RESET-Signals. Da die Signaleingänge des Flip-Flops **540** und **550** auf eine hohe Referenzspannung bezogen sind, wird wenn SECURE-RESET-Signal für zumindest zwei Taktzyklen bestätigt wird, das logische Active-Low-Signal UNSECURE* ent-bestätigt (d. h. ein logisch hoher Wert), was anzeigt, dass die sensiblen Daten sicher sind.

[0042] In der veranschaulichten Ausführungsform wird TEST-MODE bestätigt, bevor SECURE-RESET-Signal bestätigt wird, und das UNSECURE*-Signal wird auf den Empfang des bestätigten SECURE-RESET-Signals ent-bestätigt, was anzeigt, dass die sensiblen Daten sicher sind und UNSECURE* nichtbestätigt verbleibt, auch nachdem das SECURE-RESET-Signal ent-bestätigt wurde. Wenn jedoch das TEST-MODE-Signal den Zustand verändert, nachdem das SECURE-RESET-Signal entbestätigt wurde, wird das UNSECURE*-Signal bestätigt, was anzeigt, dass die Daten in der Scan-Kette nicht sicher sind. Um beim Verständnis der Funktion des UNSECURE*-Signals zu helfen, soll im folgenden Absatz ein Beispiel betrachtet werden.

[0043] Beim Betrachten des folgenden Beispiels sei vermerkt, dass SCAN-IN-(INTERNAL) **507**, SCAN-OUT-(INTERNAL) **517** und SCAN-ENABLE-(INTERNAL) Gatter-Versionen von SCAN-IN **181**, SCAN-OUT **189** und SCAN-ENABLE (Fig. 1) sind, die alle dazu verwendet werden können, um einen Zugang von außen auf die Scan-Kette **180** einzuschränken. Es sei bemerkt, dass SCAN-IN-(INTERNAL) und SCAN-OUT-(INTERNAL) nicht ausdrücklich in Fig. 1 veranschaulicht sind, sie würden jedoch, wenn sie implementiert wären, die SCAN-IN- und SCAN-OUT-Signale, die in Fig. 1 veranschaulicht sind, steuern. Es sei beispielsweise angenommen, dass die Scan-Kette **180** (Fig. 1) sich momentan in einem Test-Modus befindet, wobei das TEST-MODE-Signal bestätigt und das UNSECURE*-Signal nicht bestätigt sind. Um den Scan-Testmodus zu verlassen, wird das TEST-MODE-Signal ent-bestätigt. Das Flip-Flop **550** wird durch die fallende Flanke des TEST-MODE-Signals getriggert, was wiederum bewirkt, dass UNSECURE* bestätigt wird. Das bestätigte UNSECURE*-Signal zeigt an, dass der Scan-Test-Modus verlassen wird und dass die in der Scan-Kette **180** (Fig. 1) befindlichen Daten zu löschen sind. In der veranschaulichten Ausführungsform wird UNSECURE* als Eingabe für das Logikgatter **510** verwendet, um die Daten SCAN-IN **581** vor einem Weitergeben durch das AND-Gatter **510** und vor einem Umwandeln in SCAN-IN-(INTERNAL)-Daten **507** zu blockieren, die in die Scan-Kette **180** (Fig. 1) eingescannt werden können, und die Daten SCAN-OUT-(INTERNAL) **517** davor zu blockieren, dass sie aus der Scan-Kette **180** (Fig. 1) ausgelesen werden. Zusätzlich wird UNSECURE* als Eingabe für das Logikgatter **512** verwendet, um SCAN-ENABLE-(INTERNAL) davor zu blockieren, als Antwort auf SCAN-ENABLE bestätigt zu werden, wenn die Daten nicht sicher sind. Es sollte klar sein, dass ein beliebiges dieser Verfahren dazu verwendet werden kann, um zu verhindern, dass Daten aus der Vorrichtung **100** ausgescannt werden.

[0044] Zusätzlich zu der Verwendung als Eingabe für die Logikgatter **510**, **512** und **520** kann das UNSECURE*-Signal dazu verwendet werden, beispielsweise einen zentralen Prozessor zu benachrichtigen, ein SECURE-RESET-Signal zu bestätigen oder auf andere Weise in der Scan-Kette zu löschen. Es kann auch dazu verwendet werden, ein Sicher/Unsicher-Register (nicht abgebildet) 10 einzustellen, dass es sich auf verschiedene Firmware oder Software beziehen kann, um den Status der Scan-Kette zu bestimmen. Alternativ könnte das UNSECURE*-Signal als direkte Eingabe verwendet werden, um die Konfiguration eines oder mehrerer Signalspeicher oder Zustandsautomaten zu steuern, wie in Fig. 1 veranschaulicht.

[0045] Schließlich verwendet die in Fig. 5 gezeigte Ausführungsform ein SECURE-RESET-Signal zur Bestätigung eines bestätigten CLEAR/RESET-Signal, um Daten zu modifizieren, die in scan-beobachtbaren Elementen der Scan-Kette gespeichert sind. Es sei bemerkt, dass CLEAR/RESET analog zu SCAN-RESET der Fig. 1 sein kann. Das SECURE-RESET-Signal kann automatisch mittels einer weiteren Schaltung innerhalb eines Prozessors erzeugt werden, der einen Scan-Controller **120** einsetzt oder ein System kann so konfiguriert werden, das es ein SECURE-RESET-Signal lediglich dann erzeugt, wenn ein Bediener physikalisch einen Reset initiiert.

[0046] Unter nachfolgender Bezugnahme auf die Fig. 3 und Fig. 6 wird die Zeitgebungsbeziehung zwischen den Signalen, die in einer Ausführungsform eines Scan-Controllers verwendet werden, erläutert. Fig. 6 veranschaulicht die Zeitgebungsbeziehung beim Eintreten in einen Testmodus. Die gesamte Zeitgebung in der folgenden Erläuterung bezieht sich auf den Takt **610** und insbesondere auf die erste ansteigende Flanke des Taktzyklus C1. Vor dem Beginnen des Taktzyklus C1 sind alle Signale nicht bestätigt, was einen Betrieb in einem normalen, das heißt Nicht-Testmodus anzeigt. Während der ersten Hälfte des ersten Taktzyklus C1 versucht ein Benutzer, das Scannen durch das Bestätigen von SCAN-ENABLE **630** zu aktivieren. Da der Prozessor, der den Scan-Controller **120** verwendet, immer noch in einem normalen Modus arbeitet, bewirkt das Bestätigen von SCAN-ENABLE **630** nicht, das SCAN-ENABLE-(INTERNAL) **640** auf High geht.

[0047] Bei der ersten ansteigenden Flanke des zweiten Taktzyklus C2 jedoch wird TEST-MODE **620** normal weiter bestätigt. Als Antwort auf die Bestätigung von TEST-MODE **620** geht SCAN-RESET **660** auf High. SCAN-RESET **660** ist ein Puls, der die Modifikation, den Reset oder das Löschen von scanbeobachtbaren Elementen über die Scan-Kette (siehe in Fig. 1) triggert. Mit der ersten ansteigenden Flanke des Taktzyklus C4, zwei ansteigende Taktflanken nach dem Bestätigen von TEST-MODE **620** geht SCAN-ENABLE-(INTERNAL) **640** auf High, als Antwort darauf, dass sowohl SCAN-ENABLE **630** als auch TEST-MODE **620** auf High stehen. Das Verzögern der Bestätigung von SCAN-ENABLE-(INTERNAL) **640** für zwei Taktzyklen ermöglicht es, dass ein Reset durchgeführt wird, bevor der Zugriff auf die Scan-Kette ermöglicht wird, wodurch sensible Informationen beim Eintreten in einen Testmodus geschützt werden. Es sei bemerkt, dass RESET **650** und SCAN-EXIT **670** während des Eintretens in den Testmodus nicht bestätigt sind. Einige Zeit, nachdem SCAN-ENABLE-(INTERNAL) bei Zyklus C4 als Antwort darauf, dass TEST-MODE **620** bei Zyklus C1 bestätigt wurde, können Daten in die und von der Scan-Kette geschoben werden, um das Testen von internen Prozessorkomponenten zu erleichtern, aber an dieser Stelle wurden jegliche sichere Informationen gelöscht.

[0048] Unter nachfolgender Bezugnahme auf die Fig. 3 und Fig. 7 wird die Zeitgebungsbeziehung zwischen den Signalen, die in einer Ausführungsform eines Scan-Controllers verwendet wurden, weiterhin erläutert, insbesondere die Zeitgebungsbeziehung beim Verlassen des Testmodus. Die Zeitgebungsbeziehungen der in Fig. 7 gezeigten Signale werden unter Bezugnahme auf die erste ansteigende Flanke des Taktzyklus C1 erläutert. Am Anfang des Taktzyklus C1 werden TEST-MODE **720**, SCAN-ENABLE **730** und SCAN-ENABLE-(INTERNAL) **740** bestätigt, während alle anderen Signale negiert sind. Dies entspricht einem Testmodus, in dem Daten frei in die und von der Scan-Kette geschoben werden können, ohne das die Gefahr einer Beeinträchtigung sensibler Daten besteht. Bei der fallenden Flanke des Testzyklus C1 wird TEST-MODE **720** negiert, was das Ende des Testzyklus und den Eintritt in den normalen Modus signalisiert. Zu der gleichen Zeit, zu der TEST-MODE **720** negiert wird, wird SCAN-EXIT **770** bestätigt, um zu signalisieren, dass seit dem letzten Mal, zu dem RESET **750** bestätigt wurde, in den Testmodus gewechselt wurde und dieser verlassen wurde. SCAN-EXIT **770** kann als Eingabe für einen Zustandsautomaten verwendet werden, um einen Statusübergang zu einem bekannten Status zu erzwingen, kann als Eingabe für einen Signalspeicher verwendet werden, um den Betriebszustand des Signalspeichers zu steuern, kann an einen Prozessor gekoppelt werden, um anzuzeigen, dass Daten in der Scan-Kette zurückgesetzt werden müssen, bevor es ermöglicht wird, dass sie in normalen Operation verwendet werden, oder kann in verschiedenen anderen ähnlichen Weisen verwendet werden, um anzuzeigen, das ein Scan-Testmodus verlassen wurde.

[0049] An der fallenden Flanke des zweiten Taktzyklus C2 wird RESET **750** bestätigt, was bewirkt, dass SCAN-RESET **760** bestätigt wird. SCAN-RESET **760** wird in zumindest einer Ausführungsform als Reset-Eingabe für Datensignalspeicher und Zustandsautomaten verwendet, die die Scan-Kette bilden. Wenn es auf diese Weise verwendet wird, löscht SCAN-RESET **760** sensible Daten von der Scan-Kette beim Verlassen eines Testmodus. SCAN-ENABLE-(INTERNAL) **740** geht auf Low zur gleicher Zeit, zu der SCAN-EXIT **770** weiter negiert wird, wodurch verhindert wird, dass Daten aus der Scan-Kette nach dem Verlassen des Scan-Testmodus gescannt werden. Schließlich wird SCAN-ENABLE **730** ent-bestätigt, sodass keine zusätzlichen Daten in die Scan-Kette gescannt werden können. Es sei bemerkt, dass in der Ausführungsform des Scan-Controllers **120**, dessen Zeitgebung hier erläutert wird, SCAN-RESET **760** nicht automatisch bestätigt wird, wenn TEST-MODE **720** negiert wird. Stattdessen sollte RESET **750** als Antwort auf eine Benutzerakti-

on oder auf andere Weise bestätigt werden, wodurch bewirkt wird, dass SCAN-RESET **760** bestätigt wird, um die Scan-Kette vor dem Eintreten in einen normalen Modus zu löschen. Weitere Ausführungsformen können einen Resetpuls ähnlich dem RESET **750** automatisch beim Verlassen eines Testmodus erzeugen.

[0050] Zusammenfassend sollte aus einem Überblick über die vorangegangene Offenbarung klar sein, dass das Modifizieren von Informationen, die in einem scan-beobachtbaren Bereich eines Prozessors gespeichert sind, vor und nach dem Erlauben eines Zugriffs auf die Scan-Kette ein Prozessor, der einen Scan-Controller verwendet, der gemäß den hierin dargelegten Lehren konstruiert ist, eine erhöhte Datensicherheit zur Verfügung stellen kann, ohne die Testfähigkeit zu opfern. Eine erhöhte Datensicherheit kann dazu beitragen, mögliche Nachahmer vor dem Auswerten sensibler Daten durch das Begrenzen deren Zugriff auf die Daten abzuschrecken. Gleichzeitig kann eine größere Testfähigkeit dazu dienen, die Herstellungskosten zu reduzieren und die Einführung eines Produktes auf dem Markt zu beschleunigen.

[0051] In der vorhergehenden detaillierten Beschreibung der Figuren wurde Bezug auf die begleitenden Zeichnungen genommen, die einen Teil derselben bilden und in denen zur Veranschaulichungszwecken spezifische Ausführungsformen gezeigt wurden, in denen die Erfindung ausgeführt ist. Diese Ausführungsformen werden in ausreichendem Detail beschrieben, um dem Fachmann das Ausführen der Erfindung zu ermöglichen, und es sollte klar sein, dass andere Ausführungsformen verwendet werden können und dass logische, mechanische, chemische und elektrische Veränderungen durchgeführt werden können, ohne vom Geltungsbereich der Erfindung abzuweichen.

[0052] Die vorliegende Offenbarung ist nicht dazu gedacht, durch die spezifische Form, die hierin dargelegt wurde, begrenzt zu sein, sondern ist im Gegenteil dazu gedacht, solche Alternativen, Modifikationen und Äquivalente abzudecken, die vernünftigerweise innerhalb des Geltungsbereichs der Erfindung enthalten sind, wie dieser lediglich durch die angehängten Ansprüche definiert wird.

Patentansprüche

1. Verfahren, das die Schritte umfasst:
Empfangen eines Konfigurationssignals, um eine Scan-Kette (**180**) zum Testen vorzubereiten; wobei das Verfahren durch die Schritte gekennzeichnet ist:
Modifizieren von Informationen in einem scan-beobachtbaren Abschnitt eines Datenprozessors als Antwort auf das Konfigurationssignal; und
Aktivieren des Scan-Testens des scan-beobachtbaren Bereichs nach dem Schritt des Modifizierens.

2. Verfahren nach Anspruch 1, wobei:
der Schritt des Modifizierens das Zurücksetzen des scan-beobachtbaren Bereichs als Antwort auf das Konfigurationssignal umfasst; und wobei
der Schritt des Aktivierens das Erzeugen eines Aktiviere-Scan-Signals umfasst, um eine Scan-Logik innerhalb des scan-beobachtbaren Bereichs zu aktivieren.

3. Verfahren nach Anspruch 1, das weiterhin die Schritte umfasst:
Verhindern, das Informationen in den scan-beobachtbaren Bereich vor dem Schritt des Aktivierens gescannt werden; und
Verhindern, dass Informationen aus dem scan-beobachtbaren Bereich vor dem Schritt des Aktivierens gescannt werden.

4. Verfahren, das die Schritte umfasst:
Empfangen eines Konfigurationsindikators, um eine Scan-Kette (**180**) für einen normalen Betrieb vorzubereiten, wobei das Verfahren durch die Schritte gekennzeichnet ist:
Modifizieren von Informationen in einem scan-beobachtbaren Bereich eines Datenprozessors (**100**) als Antwort auf den Konfigurationsindikator; und
Aktivieren des normalen Betriebs des Datenprozessors nach dem Schritt des Modifizierens.

5. Verfahren nach Anspruch 4, wobei der Schritt des Modifizierens das Zurücksetzen des scan-beobachtbaren Bereichs umfasst.

6. Verfahren nach Anspruch 4, das weiterhin die Schritte des Verhinderns, dass Informationen in den scan-beobachtbaren Bereich, nachfolgend dem Schritt des Aktivierens, gescannt werden und des Verhinderns, dass Informationen aus dem scan-beobachtbaren Bereich, nachfolgend dem Schritt des Aktivierens, gescannt werden, enthält.

7. Scan-Controller (**120**), der eine Logik umfasst, um ein Scan-Zurücksetz-Signal vor dem Scan-Testen zur Verfügung zu stellen, wobei der Scan-Controller (**120**) dadurch gekennzeichnet ist, dass das Scan-Zurücksetz-Signal Informationen in einem scan-beobachtbaren Bereich eines Datenprozessors modifiziert.

8. Prozessor, umfassend:
einen funktionalen Abschnitt, um sichere Informationen während eines normalen Modus zu verarbeiten, wobei der funktionale Abschnitt während eines Testmodus beobachtbar ist;
wobei der Prozessor gekennzeichnet ist durch:
einen Teststeuerabschnitt, um einen Zugriff auf die sicheren Informationen durch das Modifizieren von Informationen in dem funktionalen Abschnitt vor dem Testen des funktionalen Abschnitts zu verhindern.

9. Prozessor nach Anspruch 8, wobei der Teststeuerabschnitt eine Logik umfasst, um ein Scan-Zurücksetz-Signal vor dem Scan-Testen zur Verfügung zu stellen, wobei das Scan-Zurücksetz-Signal Informationen in dem funktionalen Abschnitt des Prozessors modifiziert.

10. Prozessor nach Anspruch 8, wobei der Teststeuerabschnitt weiterhin eine Logik umfasst, um ein Scan-Zurücksetz-Signal, nachfolgend dem Scan-Testen, zur Verfügung zu stellen.

Es folgen 4 Blatt Zeichnungen

Anhängende Zeichnungen

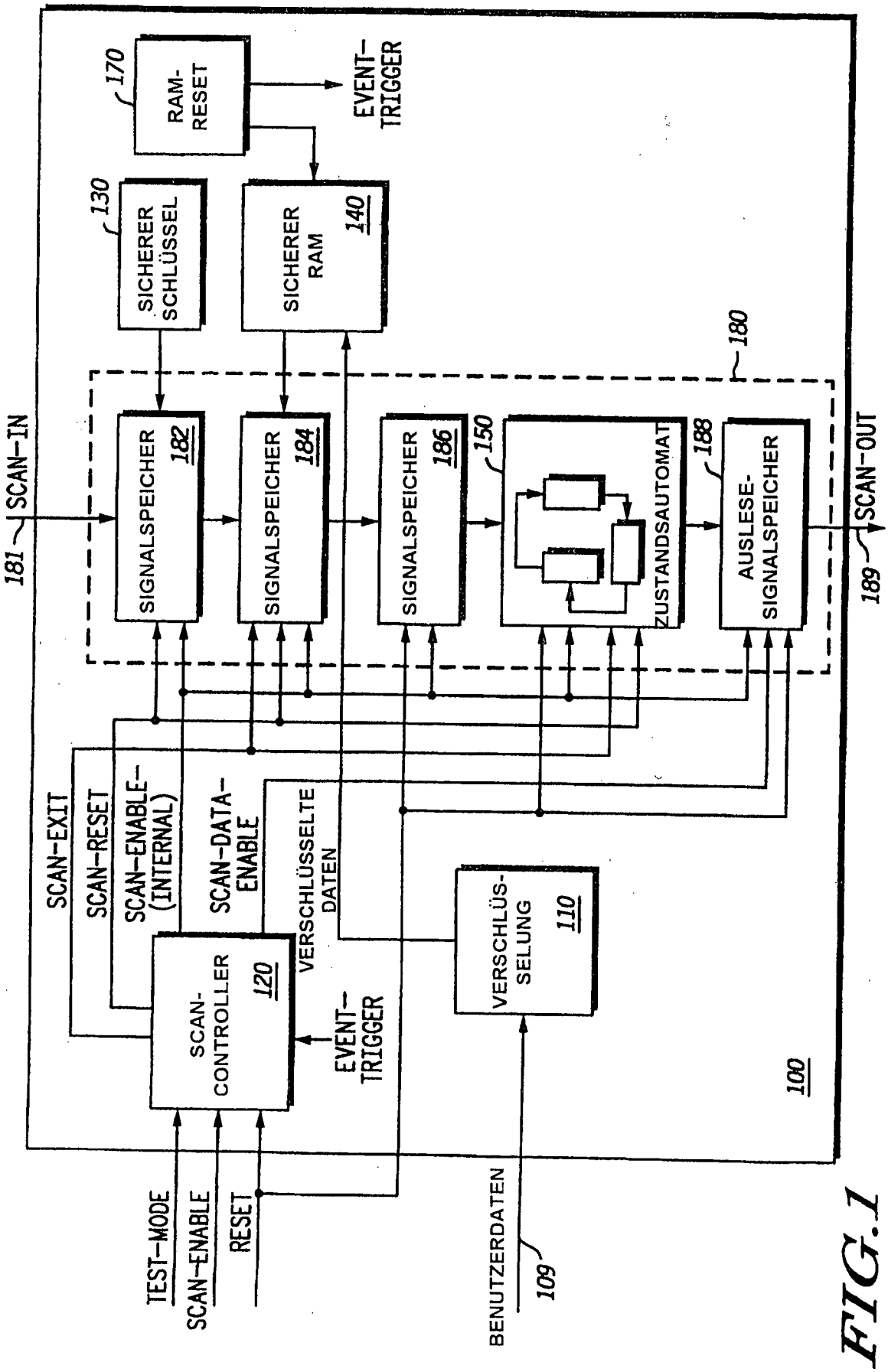

FIG. 1

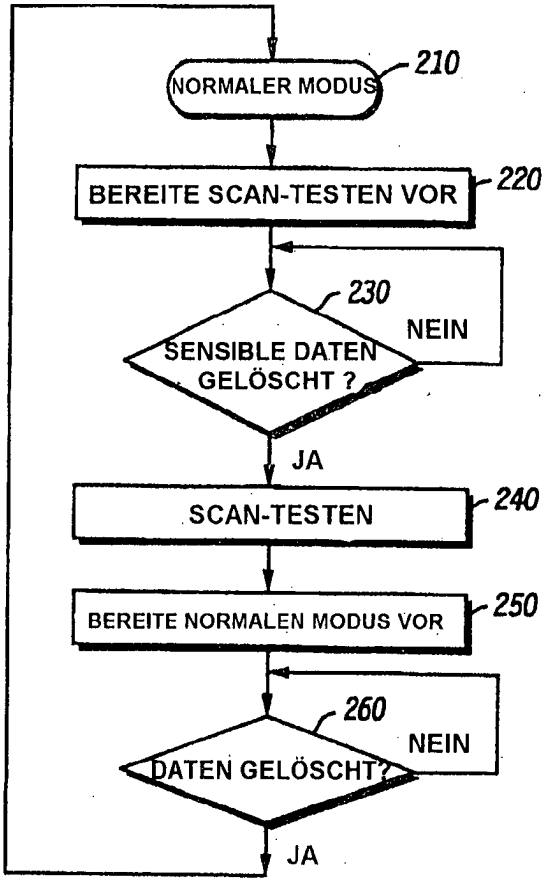
NORMALER MODUS
210
BEREITE SCAN-TESTEN VOR
220
230
NEIN
SENSIBLE DATEN GELÖSCHT ?
JA
SCAN-TESTEN
240
BEREITE NORMALEN MODUS VOR
250
260
NEIN
DATEN GELÖSCHT?
JA

FIG.2

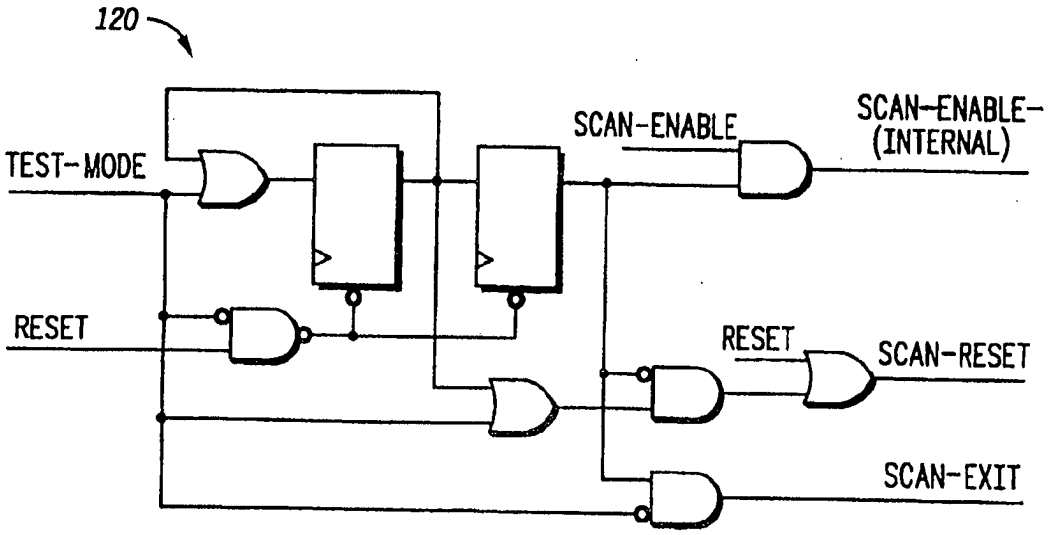
120
TEST-MODE
RESET
SCAN-ENABLE
SCAN-ENABLE-(INTERNAL)
RESET
SCAN-RESET
SCAN-EXIT

FIG.3

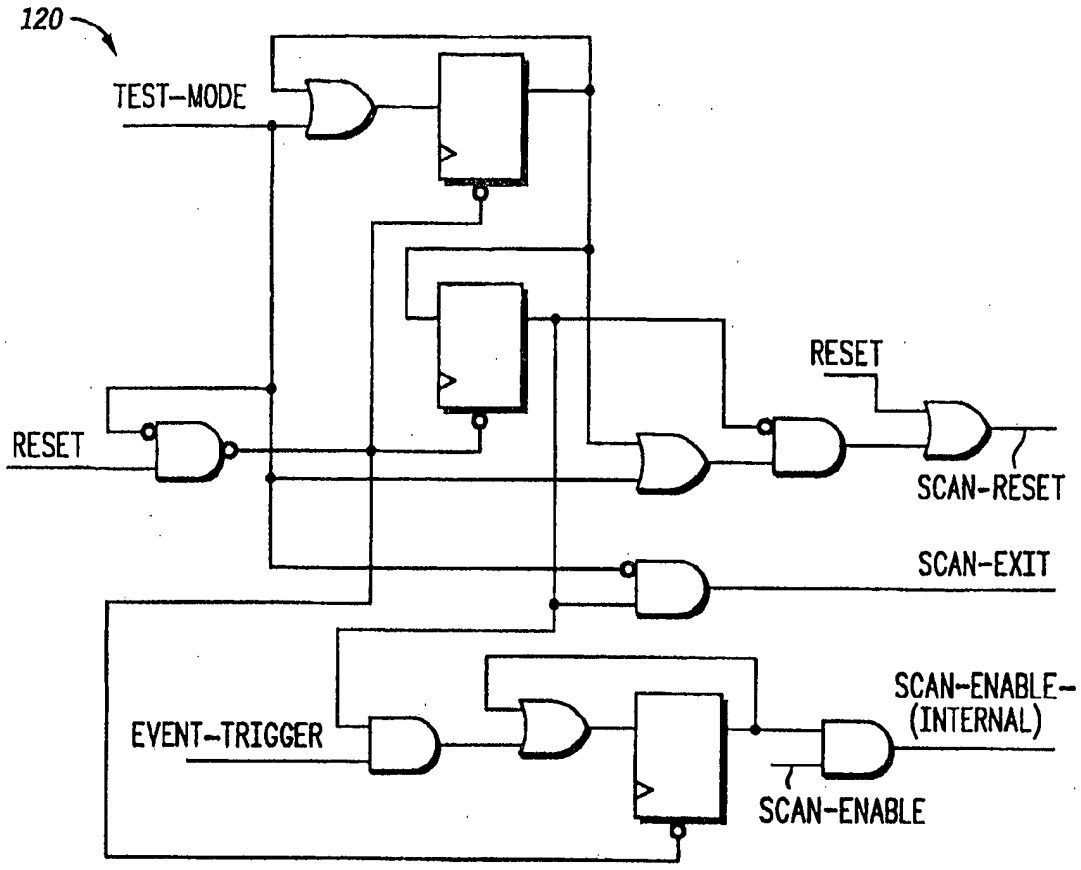
120
TEST-MODE
RESET
RESET
SCAN-RESET
SCAN-EXIT
EVENT-TRIGGER
SCAN-ENABLE-(INTERNAL)
SCAN-ENABLE

FIG.4

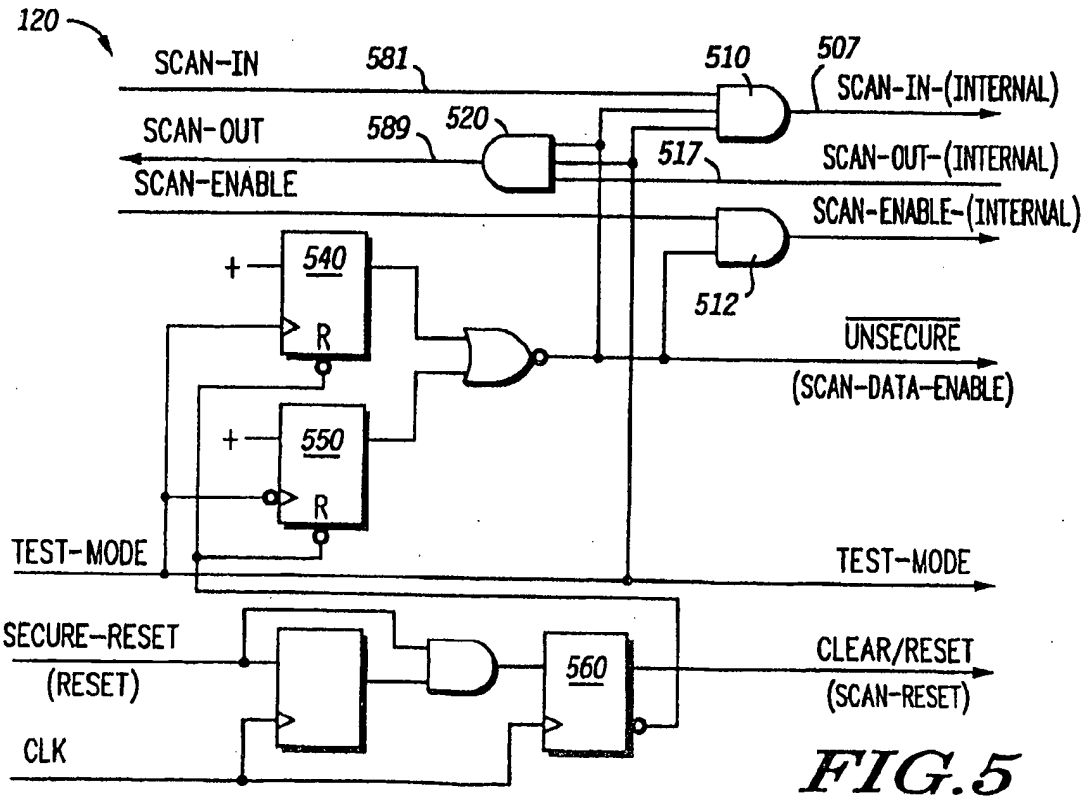
120
SCAN-IN
581
510
507
SCAN-IN-(INTERNAL)
SCAN-OUT
589
520
517
SCAN-OUT-(INTERNAL)
SCAN-ENABLE
SCAN-ENABLE-(INTERNAL)
540
R
512
UNSECURE
(SCAN-DATA-ENABLE)
550
R
TEST-MODE
TEST-MODE
SECURE-RESET
(RESET)
560
CLEAR/RESET
(SCAN-RESET)
CLK

FIG.5

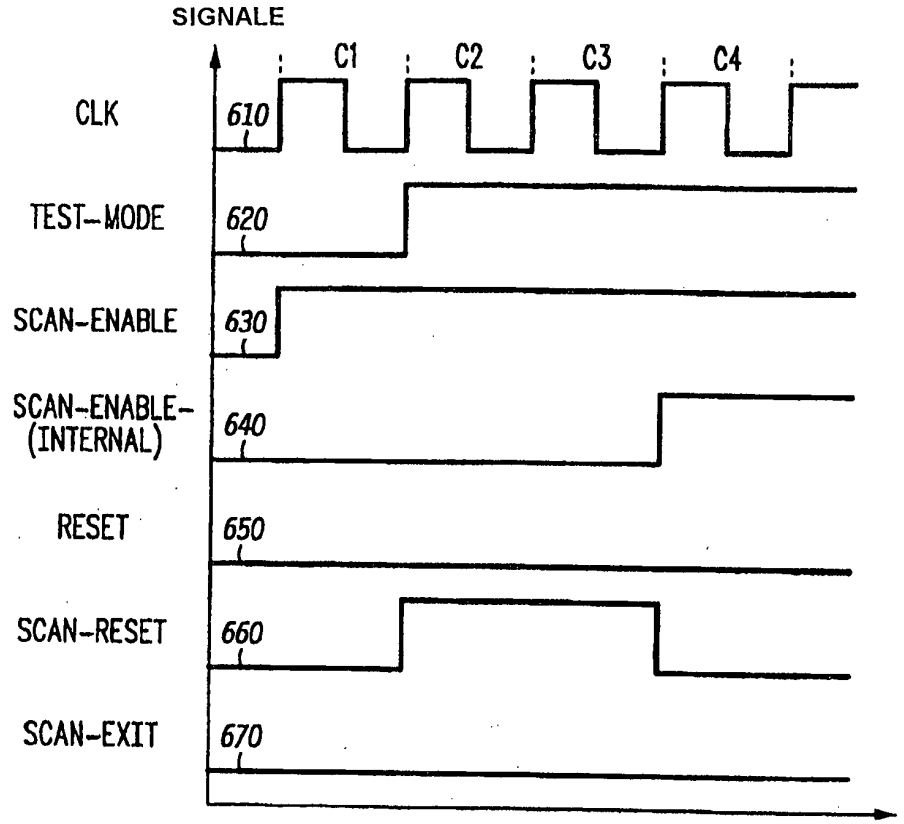
SIGNALE
C1
C2
C3
C4
CLK
610
TEST-MODE
620
SCAN-ENABLE
630
SCAN-ENABLE-
(INTERNAL)
640
RESET
650
SCAN-RESET
660
SCAN-EXIT
670
ZEIT

FIG.6

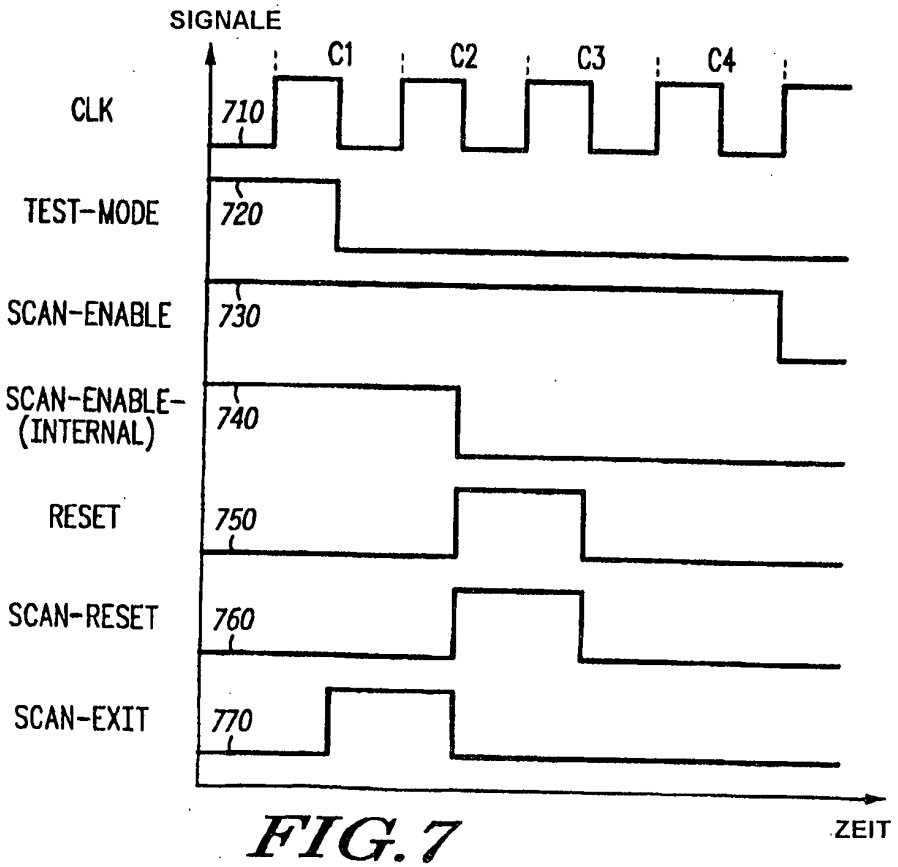
SIGNALE
C1
C2
C3
C4
CLK
710
TEST-MODE
720
SCAN-ENABLE
730
SCAN-ENABLE-
(INTERNAL)
740
RESET
750
SCAN-RESET
760
SCAN-EXIT
770
ZEIT

FIG.7