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

(10) **DE 600 31 101 T2** 2007.05.10

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 1 166 492 B1**
(21) Deutsches Aktenzeichen: **600 31 101.5**
(86) PCT-Aktenzeichen: **PCT/US00/06916**
(96) Europäisches Aktenzeichen: **00 930 090.6**
(87) PCT-Veröffentlichungs-Nr.: **WO 2000/059153**
(86) PCT-Anmeldetag: **16.03.2000**
(87) Veröffentlichungstag der PCT-Anmeldung: **05.10.2000**
(97) Erstveröffentlichung durch das EPA: **02.01.2002**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **04.10.2006**
(47) Veröffentlichungstag im Patentblatt: **10.05.2007**

(51) Int Cl.[8]: ***H04L 9/22*** (2006.01)

(30) Unionspriorität:
283096 **31.03.1999** **US**

(84) Benannte Vertragsstaaten:
DE, FI, FR, GB, SE

(73) Patentinhaber:
Intel Corporation, Santa Clara, Calif., US

(74) Vertreter:
Hauck Patent- und Rechtsanwälte, 80339 München

(72) Erfinder:
WELLS, E., Steven, El Dorado Hills, CA 95762, US;
WARD, A., David, Sacramento, CA 95826, US

(54) Bezeichnung: **TASTVERHÄLTNISKORREKTUR FÜR EINEN ZUFALLSZAHLENGENERATOR**

Beschreibung

GEBIET DER ERFINDUNG

[0001] Die vorliegende Erfindung betrifft allgemein die Computersicherheit und im Besonderen das Erzeugen eines ungefähr einheitlichen Arbeitszyklus in einem Zufallszahlengenerator.

STAND DER TECHNIK

[0002] Zufallszahlengeneratorschaltungen werden in einer Vielzahl von elektronischen Anwendungen eingesetzt. Eine wichtige Anwendung für Zufallszahlengeneratoren betrifft das Gebiet der Computersicherheit, wenn Nachrichten- bzw. Mitteilungsdaten verschlüsselt und entschlüsselt werden. Die Kryptographie umfasst die Transformation von Daten in eine codierte Nachricht, die an einen vorgesehenen Empfänger übermittelt und von diesem decodiert werden soll. Die meisten kryptographischen Techniken verwenden Schlüssel, die von dem Sender verwendet werden, um die Nachricht zu codieren, und wobei sie von dem Empfänger verwendet werden, um die codierte Nachricht zu decodieren. Herkömmliche Verschlüsselungssysteme verwenden entweder einen einzigen Schlüssel, d.h. einen Schlüssel zum Codieren und Decodieren einer Nachricht, oder zwei Schlüssel, einen zum Codieren der Nachricht und einen zum Decodieren der Nachricht.

[0003] Die Schlüssel, die zum Verschlüsseln und Entschlüsseln von Nachrichten verwendet werden, sind im Wesentlichen binäre Datenmuster, in Bezug auf welche eine Nachricht verarbeitet oder gefiltert wird. Effektive Verschlüsselungssysteme erfordern den Einsatz von Schlüsseln mit einer ausreichend hohen Anzahl von Bits, um eine Replikation eines Schlüssels nahezu unmöglich zu machen. Ferner müssen die Datenmuster, welche die Schlüssel umfassen, ausreichend zufällig bzw. willkürlich sein, so dass ihr Muster oder die Muster in der durch den Schlüssel codierten Nachricht nicht vorhergesehen werden können. Effektive kryptographische Systeme erfordern somit den Einsatz hochwertiger Zufallszahlengeneratoren, um zu gewährleisten, dass die binären Daten in einer Nachricht auf eine vollständig unvorhersehbare Art und Weise transformiert werden. Im Allgemeinen erzeugt jeglicher Mangel der Willkürlichkeit bzw. der Zufälligkeit in einem Verschlüsselungssystem ein gewisses Maß der Korrelation zwischen den codierten und den nicht codierten Daten. Diese Korrelation kann dann dazu eingesetzt werden, den Code durch Techniken zu knacken, wie etwa durch iterative empirische Versuchsprädiktionen möglicher Ausgabemuster auf der Basis einer codierten Nachricht.

[0004] Ein wünschenswertes Merkmal eines binären Zufallszahlengenerators ist es, dass er Bits von Einsen und Nullen in einer absolut zufälligen Anordnung ausgibt. Der Wert des Ausgangsbits sollte somit jederzeit vollständig unvorhersehbar sein. Es ist wünschenswert, dass der Arbeitszyklus des Ausgangs des Zufallszahlengenerators ungefähr fünfzig Prozent über einer unendlichen Probengröße liegt, so dass die Wahrscheinlichkeit, dass eine Ausgabe logisch niedrig (Null) ist, gleich der Wahrscheinlichkeit ist, dass es sich bei der Ausgabe um einen logisch hohen Wert (Eins) handelt. Ferner ist es wünschenswert, dass ein Zufallszahlengenerator eine niedrige Korrelation (z.B. eine Korrelation von ungefähr Null) zwischen jedem Bit und jedem anderen Bit aufweist sowie eine flache Fourier-Verteilung zwischen den Ausgangsbits.

[0005] Das U.S. Patent US-A-5.781.458 offenbart eine Vorrichtung zum Erzeugen von Zufallszahlen zur Verwendung in einem kryptographischen Schlüsselgenerator. Die Vorrichtung extrahiert Entropie (mittleren Informationsgehalt) aus dem Ausgangssignal einer RC-Oszillationsschaltung, indem die Periode eines ersten Signals mit der Periode eines zweiten Signals verglichen wird, wobei die erste und die zweite Periode durch zwei Zyklen getrennt sind. Wenn die Perioden gleich sind, werden die Daten verworfen. Wenn die vorherige Periode größer ist als die spätere Periode, so wird einem Schieberegister eine „1" zugeordnet, wobei im anderen Fall dem Schieberegister eine „0" zugeordnet wird. Eine derartige Vorrichtung ist nicht in der Lage eine einheitliche Arbeitszyklusausgabe von 1en und 0en zu erzeugen, was eine wünschenswerte Aufgabe der vorliegenden Erfindung ist.

[0006] Aktuell bekannte Zufallszahlengeneratoren neigen jedoch dazu, eine uneinheitliche Anzahl von Nullen oder Einsen über eine statistisch signifikante Probengröße zu erzeugen. Ein typischer Grund dafür, dass dem Stand der Technik entsprechende Zufallszahlengeneratoren einen ungleichmäßigen Arbeitszyklus aufweisen ist es, dass die Latches, welche den Zufallszahlengenerator umfassen, für gewöhnlich einen der beiden Zustände bevorzugen, wenn Daten während einer unzulässigen Einricht-/Haltezeit zwischengespeichert werden. Ein kennzeichnendes aktuelles Verfahrne zur Reduzierung der Schwankungen von Arbeitszyklen in Zufallszahlengeneratoren umfasst den Einsatz eines Linear Feedback Shift Registers (LFSR bzw. eines linear rückgekoppelten Schieberegisters) auf der Ausgangsstufe einer Zufallsbitquelle bzw. einer wahlfreien Bitquelle.

[0007] Die Abbildung aus Fig. 1 veranschaulicht einen dem Stand der Technik entsprechenden Zufallszahlengenerator, der ein linear rückgekoppeltes Schieberegister **104** verwendet, das mit dem Ausgang einer Zufallsbitquelle **102** gekoppelt ist. Das LFSR **104** umfasst eine Reihe von Latches **105** und

Gattern **106**, durch welche die Ausgangsbit von der Zufallsbitquelle **102** verteilt werden. Die Zustände der Ausgangsbits werden durch die Gatter **106** zufällig invertiert, und die Reihenfolge der Bits wird durch Rückkopplung der Bits durch Latches **105** weiter gemischt.

[0008] Im Allgemeinen weisen linear rückgekoppelte Schieberegister, wie das in der Abbildung aus Fig. 1 veranschaulichte Register, bestimmte Nachteile auf und berichtigen nicht in vollem Umfang uneinheitliche Arbeitzykluseigenschaften, die kennzeichnende Zufallsbitquellen aufweisen. Wie dies durch das LFSR **104** dargestellt ist, umfasst ein kennzeichnendes LFSR selbst eine Reihe von Latches und Gattern. Diese Latches und Gatter neigen dazu, unter bestimmten Umständen die gleiche Eigenschaft zum Zwischenspeichern einer Null oder einer Eins aufzuweisen wie die Latches in der Zufallsbitquelle **102**. Ein kennzeichnendes LFSR erzeugt somit selbst keine einheitliche Arbeitszyklusausgabe von Einsen und Nullen und kann somit etwaige Arbeitszyklusschwankungen in einer Zufallsbitquelle nicht vollständig berichtigen.

[0009] Ein weiterer Nachteil der linear rückgekoppelten Schieberegister ist die Anforderung in Bezug auf eine große Anzahl von Latches und Gattern. Zum Beispiel erfordert ein 32-Bit-LFSR, wie etwa gemäß der Abbildung aus Fig. 1, 32 D-Typ-Latches sowie eine Reihe kombinatorischer Gatter. Dies erhöht signifikant die erforderliche Siliziumfläche für eine Zufallszahlengeneratorschaltung, die ein LFSR verwendet.

ZUSAMMENFASSUNG DER ERFINDUNG

[0010] Vorgesehen ist gemäß einem ersten Aspekt der vorliegenden Erfindung ein Verfahren gemäß dem gegenständlichen Anspruch 1.

[0011] Vorgesehen ist gemäß einem zweiten Aspekt der vorliegenden Erfindung eine Arbeitszyklus-Berichtigungsschaltung gemäß dem gegenständlichen Anspruch 8.

[0012] Weitere Merkmale und Vorteile der vorliegenden Erfindung werden aus den beigefügten Zeichnungen sowie aus der folgenden genauen Beschreibung der Erfindung deutlich.

KURZE BESCHREIBUNG DER ZEICHNUNGEN

[0013] Die vorliegende Erfindung ist in den Abbildungen der beigefügten Zeichnungen beispielhaft und ohne einzuschränken veranschaulicht, wobei die gleichen Elemente darin mit den gleichen Bezugszeichen bezeichnet sind. Es zeigen:

[0014] Fig. 1 einen herkömmlichen Zufallszahlengenerator unter Verwendung eines linearen rückgekoppelten Schieberegisters;

[0015] Fig. 2 ein Blockdiagramm einer Zufallsbitquelle und eines Ausführungsbeispiels der Arbeitszyklus-Berichtigungseinrichtung;

[0016] Fig. 3 ein Logikdiagramm eines Ausführungsbeispiels der Arbeitszyklus-Berichtigungseinrichtung aus Fig. 2;

[0017] Fig. 4 ein Flussdiagramm des Betriebs der Arbeitszyklus-Berichtigungseinrichtung aus Fig. 2;

[0018] Fig. 5 ein Beispiel für ein berichtigtes Bitmuster, das durch die Arbeitszyklus-Berichtigungseinrichtung aus Fig. 3 erzeugt worden ist;

[0019] Fig. 6 ein Logikdiagramm eines weiteren Ausführungsbeispiels der Arbeitszyklus-Berichtigungseinrichtung aus Fig. 2;

[0020] Fig. 7 ein Flussdiagramm, das den Betrieb der Arbeitszyklus-Berichtigungseinrichtung aus Fig. 6 veranschaulicht;

[0021] Fig. 8 ein Beispiel des durch die Arbeitszyklus-Berichtigungseinrichtung aus Fig. 6 erzeugten berichtigten Bitmusters;

[0022] Fig. 9 ein Logikdiagramm eines Ausführungsbeispiels der Arbeitszyklus-Berichtigungseinrichtung aus Fig. 2;

[0023] Fig. 10 ein Flussdiagramm des Betriebs der Arbeitszyklus-Berichtigungseinrichtung aus Fig. 9;

[0024] Fig. 11 ein Beispiel für das durch die Arbeitszyklus-Berichtigungseinrichtung aus Fig. 9 erzeugte berichtigte Bitmuster;

[0025] Fig. 12 ein Logikdiagramm eines Ausführungsbeispiels der Arbeitszyklus-Berichtigungseinrichtung aus Fig. 2;

[0026] Fig. 13 ein Flussdiagramm des Betriebs der Arbeitszyklus-Berichtigungseinrichtung aus Fig. 12;

[0027] Fig. 14 ein Beispiel des durch die Arbeitszyklus-Berichtigungseinrichtung aus Fig. 12 erzeugten berichtigten Bitmusters; und

[0028] Fig. 15 ein Blockdiagramm eines Computernetzwerks, das ein Bitpaarbildungssystems zur Datenverschlüsselung/Datenentschlüsselung gemäß einem Ausführungsbeispiel der vorliegenden Erfindung.

GENAUE BESCHREIBUNG DER ERFINDUNG

[0029] Beschrieben wird eine Arbeitszyklus-Berichtigungseinrichtung zur Verwendung in einem Zufallszahlengenerator. In einem Ausführungsbeispiel werden sequentielle Par von Bitausgaben der Zufallsbitquelle bzw. der wahlfreien Bitquelle durch die Arbeitszyklus-Berichtigungseinrichtung verarbeitet. Wenn beide Bits in einem Bitpaar identisch sind, verwirft die Arbeitszyklus-Berichtigungseinrichtung das Bitpaar oder gibt das Bitpaar nicht aus. Wenn sich die Bits in einem Bitpaar unterscheiden, so gibt die Arbeitszyklus-Berichtigungseinrichtung eines der Bits aus dem Bitpaar aus.

[0030] Es ist ein beabsichtigter Vorteil der Ausführungsbeispiele der Erfindung, eine Schaltung bereitzustellen, die einen ungefähr einheitlichen bzw. gleichmäßigen Arbeitszyklus für die Ausgabe einer wahlfreien Bitquelle bereitstellt. Ein weiterer beabsichtigter Vorteil der Ausführungsbeispiele der Erfindung ist es, einen Zufallszahlengenerator bereitzustellen, der eine kleinere Siliziumfläche erfordert, wenn er in einem integrierten Schaltungsbaustein implementiert ist.

[0031] Eine Zufallsbitquelle ist eine digitale Schaltung, die eine Reihe binärer Stellen in scheinbar willkürlicher Reihenfolge ausgibt. Bei einer idealen Zufallsbitquelle ist die Wahrscheinlichkeit, dass ein bestimmtes Ausgangsbit gleich Null ist, identisch mit der Wahrscheinlichkeit, dass es einer Eins entspricht. Das heißt, der Arbeitszyklus der Ausgangskurvenform der Zufallsbitquelle entspricht einheitlich fünfzig Prozent über eine statistisch signifikante Probengröße. Die meisten wahlfreien Bitquellen bzw. Zufallsbitquellen weisen jedoch eine gewisse Schwankung des Arbeitzyklus auf, aufgrund einer Tendenz der Latches und Gatter in der Zufallsbitquelle, einen bestimmten Logikwert zwischenzuspeichern, wenn Daten während einer unzulässigen Halte- oder Einrichtzeit zwischengespeichert werden.

[0032] Die Wahrscheinlichkeit, dass ein bestimmtes Bit zu einem bestimmten Zeitpunkt durch eine wahlfreie Bitquelle ausgegeben wird, kann durch bestimmte mathematische Beziehungen ausgedrückt werden. Wenn die Wahrscheinlichkeit, dass die Ausgabe eine Null ist (P(0)), gleich p ist, so ist die Wahrscheinlichkeit, dass es sich bei der Ausgabe um eine Eins (P(1)) handelt, gleich 1 – p. Das heißt:
Wahrscheinlichkeit für die Erzeugung einer Null: $P(0) = p$
Wahrscheinlichkeit für die Erzeugung einer Eins: $P(1) = 1 – p$

[0033] Für eine ideale Zufallsbitquelle ist p gleich 50 Prozent. Bei einer nicht idealen Zufallsbitquelle kann p deutlich größer oder kleiner sein als 50 Prozent.

[0034] Wenn sequentielle Ausgangsbits der Zufallsbitquelle in Paaren berücksichtigt werden, so werden die Wahrscheinlichkeiten zu:
Wahrscheinlichkeit für die Erzeugung von Null, Null: $P(00) = P(0)P(0) = p^2$
Wahrscheinlichkeit für die Erzeugung von Null, Eins: $P(01) = P(0)P(1) = p(1 – p)$
Wahrscheinlichkeit für die Erzeugung von Eins, Null: $P(10) = P(1)P(0) = (1 – p)p$
Wahrscheinlichkeit für die Erzeugung von Eins, Eins: $P(11) = P(1)P(1) = (1 – p)^2$

[0035] Mathematisch sind die Wahrscheinlichkeiten für die Erzeugung eines Ausgangspaares von 0,1 oder eines Ausgangspaares von 1,0 identisch, wie dies aus den oben genannten Wahrscheinlichkeitsgleichungen hervorgeht. Das heißt, da $p(1 – p) = (1 – p)p$, gilt $P(01) = P(10)$.

[0036] Diese Eigenschaft gilt unabhängig von der Wahrscheinlichkeit der Zufallsbitquelle für die Erzeugung einer Eins oder Null für jede gegebene Ausgabe. Selbst wenn p somit für eine bestimmte Zufallsbitquelle ungleich 50 Prozent ist, so entspricht die Wahrscheinlichkeit, dass die Zufallsbitquelle ein Ausgabepaar Null-Eins erzeugt, gleich der Wahrscheinlichkeit, dass sie ein Ausgabepaar Eins-Null erzeugt. In einem Ausführungsbeispiel der vorliegenden Erfindung wird dieses Prinzip eingesetzt für die Berichtigung der Ausgabe einer Zufallsbitquelle, die einen uneinheitlichen Arbeitszyklus aufweist und eine ungleichmäßige Verteilung von Nullen und Einsen in einem bestimmten Ausgangsbitstrom erzeugt.

[0037] Gemäß einem Verfahren der vorliegenden Erfindung verarbeitet die Arbeitszyklus-Berichtigungseinrichtung ausgegebene Bitpaare von einer Zufallsbitquelle, um einen berichtigten, im Wesentlichen einheitlichen Bitstrom zu bestimmen. Wenn in einem Ausführungsbeispiel beide Bits in einem Paar identisch sind, so wird das Paar verworfen, und es wird nicht durch die Arbeitszyklus-Berichtigungseinrichtung als Teil des berichtigten Bitstroms ausgegeben. Wenn somit beide Bits in einem Ausgangsbitpaar eine Null darstellen, so wird dieses Paar verworfen. Wenn beide Bits in einem Ausgangsbitpaar im anderen Fall eine Eins darstellen, so wird das Paar ebenfalls verworfen. Wenn sich die Bits in einem Ausgangsbitpaar hingegen unterscheiden, so gibt die Arbeitszyklus-Berichtigungseinrichtung eines der Bits in dem Paar als ein Bit in dem berichtigten Bitstrom aus. In einem Ausführungsbeispiel gibt die Arbeitszyklus-Berichtigungseinrichtung das erste Bit in einem ungleichen Paar von Bits als das berichtigte Bit aus. Wenn somit in Bezug auf dieses Ausführungsbeispiel das Ausgabepaar Null-Eins lautet, so wird das berichtigte Bit auf Null gesetzt, und wenn das Ausgabepaar Eins-Null lautet, so wird das berichtigte Bit auf Eins gesetzt. Die berichtigten Bitwerte, die den verschiedenen paarweisen Fällen ent-

sprechen, können durch die folgenden Beziehungen dargestellt werden:

$P(00) = P(0)P(0) = p^2$	Zurückweisung
$P(01) = P(0)P(1) = p(1 - p)$	Ausgabe logische 0
$P(10) = P(1)P(0) = (1 - p)p$	Ausgabe logische 1
$P(11) = P(1)P(1) = (1 - p)p^2$	Zurückweisung

[0038] In einem alternativen Ausführungsbeispiel gibt die Arbeitszyklus-Berichtigungseinrichtung das zweite Bit in einem ungleichen Paar als das berichtigte Bit aus. Wenn in diesem Ausführungsbeispiel das Ausgabepaar Null-Eins lautet, wird das berichtigte Bit auf Eins gesetzt; und wenn das Ausgabepaar Eins-Null lautet, wird das berichtigte Bit auf Null gesetzt. Die berichtigten Bitwerte, die den verschiedenen Paaren entsprechen, können für dieses alternative Ausführungsbeispiel durch die folgenden mathematischen Beziehungen dargestellt:

$P(00) = P(0)P(0) = p^2$	Zurückweisung
$P(01) = P(0)P(1) = p(1 - p)$	Ausgabe logische 1
$P(10) = P(1)P(0) = (1 - p)p$	Ausgabe logische 0
$P(11) = P(1)P(1) = (1 - p)p^2$	Zurückweisung

[0039] Die Abbildung aus Fig. 2 zeigt ein Ausführungsbeispiel einer Arbeitszyklus-Berichtigungseinrichtung **200**, das die vorstehend beschriebenen Ausführungsbeispiele implementiert, um einen im Wesentlichen einheitlichen Bitstrom von einer Zufallsbitquelle **202** zu erzeugen, wobei weniger Gatter als in früheren LFSR-Schaltungen verwendet werden. Bei der Zufallsbitquelle **202** kann es sich um jede Zufallsbitquelle handeln, die einen Zufallsbitstrom auf der Signalleitung **222** ausgibt. In einem Ausführungsbeispiel der vorliegenden Erfindung ist die Zufallsbitquelle **202** als eine Latch-Schaltung implementiert, die ein zufällig schwankendes Taktsignal mit niedriger Geschwindigkeit verwendet, um periodisch ein Hochgeschwindigkeits-Oszillationssignal zwischenzuspeichern. Der Wert Des Ausgangsbits aus dem Latch der Zufallsbitquelle ist abhängig von dem Spannungswert des Hochgeschwindigkeitssignals, wenn dieses durch das Niedergeschwindigkeitssignal zwischengespeichert wird. Die Zufallsbitquelle **202** kann auch ein Takt- oder Strobe-Signal CLK auf der Signalleitung **216** erzeugen, wie dies im Fach allgemein bekannt ist.

[0040] Die Arbeitszyklus-Berichtigungseinrichtung **200** weist die Speicherelemente **204** und **206** auf, die Vergleichsschaltung **208**, die Validierungslogik **210** und eine Ausgangsschaltung **212**. Die Speicherschaltungen **204** und **206** speichern Paar aufeinander folgender Bits in dem Zufallsbitstromausgang aus der Zufallsbitquelle **202** für einen Vergleich durch die Vergleichsschaltung **208**. In einem ersten Taktzyklus von CLK wird das erste Bit in einem Bitpaar in der Speicherschaltung **204** gespeichert. In einem späteren Taktzyklus wird das erste Bit in der Speicherschaltung **206** gespeichert, während die nächste Bitausgabe der Zufallsbitquelle **202** in der Speicherschaltung **204** gespeichert wird. Die Speicherschaltungen **204** und **206** können jede Art von Speicherelementen darstellen, einschließlich Latches, Register, flüchtige oder nichtflüchtige Speicherzellen und dergleichen.

[0041] Die in den Speicherschaltungen **204** und **206** gespeicherten Bits werden durch die Vergleichsschaltung **208** verglichen. Bei der Vergleichsschaltung **208** kann es sich um jede Art von Vergleichsschaltung handeln, einschließlich eines XOR-Gatters oder eines Komparators bzw. einer Vergleichseinrichtung. Wenn beide Bits identisch sind, aktiviert die Vergleichsschaltung **208** das Signal ACCEPT auf der Signalleitung **220** in einen hohen Logikzustand. Wenn beide Bits nicht identisch sind, deaktiviert die Vergleichsschaltung **208** das Signal ACCEPT in einen niedrigen Logikzustand. Somit zeigt das Signal ACCEPT an, ob das Paar der von der Zufallsbitquelle ausgegebenen Bits ein berichtigtes Bit in dem Bitstrom erzeugt, der durch die Arbeitszyklus-Berichtigungseinrichtung **200** ausgegeben wird.

[0042] Das Signal ACCEPT wird gemeinsam mit CLK an die Validierungslogik **210** bereitgestellt. Die Vergleichsschaltung **208** aktiviert ACCEPT, wenn es sich bei zwei beliebigen aufeinander folgenden Bits in den Speicherschaltungen **204** und **206** um unterschiedliche Bits handelt. Es ist jedoch vorteilhaft, sich nicht überschneidende Paar aus der Zufallsbitquelle **202** ausgegebenen Bits zu vergleichen, um einen einheitlichen Arbeitszyklus-Ausgangsstrom der Arbeitszyklus-Berichtigungseinrichtung **200** zu erzeugen. Die Validierungslogik **210** führt diese Funktion aus. Die Validierungslogik **210** aktiviert das Strobe-Signal STB auf einen hohen Logikzustand auf der Signalleitung **218**, wenn ACCEPT aktiviert wird, und wobei das gewünschte Bitpaar in den Seicherschaltungen **204** und **206** gespeichert wird. Wenn STB aktiviert wird, speichert die Ausgangsschaltung **212** das Bit in der Speicherschaltung **206**. Die Ausgangsschaltung **212** kann jedes Speicherelement darstellen, dazu zählen ein Register, ein Latch, eines oder mehrere flüchtige oder nichtflüchtige Speicherelemente, ein AND-Gatter oder eine andere Logik. Das durch die Ausgangsschaltung **212** gespeicherte Bit wird an die Signalleitung **214** ausgegeben, wenn das berichtigte Bit den Bits entspricht, die in den Speicherschaltungen **204** und **206** gespeichert sind.

[0043] In Bezug auf ein alternatives Ausführungsbeispiel kann die Ausgabe der Speicherschaltung **204** an Stelle der Ausgabe der Speicherschaltung

206 an die Ausgangsschaltung **212** bereitgestellt werden. Wenn in Bezug auf das vorliegende Ausführungsbeispiel STB aktiviert wird, speichert die Ausgangsschaltung **212** das Bit in der Speicherschaltung **204** und gibt das Bit als das berichtigte Bit aus, das den Bits entspricht, die in den Speicherschaltungen **204** und **206** gespeichert werden.

[0044] Die Abbildung aus Fig. 3 zeigt eine Arbeitszyklus-Berichtigungseinrichtung **300**, die ein Ausführungsbeispiel der Arbeitszyklus-Berichtigungseinrichtung **200** darstellt. Die Arbeitszyklus-Berichtigungseinrichtung **300** weist die Latches **304** und **306** auf, die ein Ausführungsbeispiel der entsprechenden Speicherschaltungen **204** und **206** darstellen; das XOR-Gatter **310**, das ein Ausführungsbeispiel der Vergleichsschaltung **210** darstellt; das transparente Latch **308** und das AND-Gatter **309**, die gemeinsam ein Ausführungsbeispiel der Validierungslogik **210** umfassen; und das Latch **312**, das ein Ausführungsbeispiel der Ausgangsschaltung **212** darstellt.

[0045] Die Latches **304** und **306** speichern Paare aufeinander folgender Bits von der Zufallsbitquelle **202** für einen Vergleich durch das XOR-Gatter **310**. Das erste Bit in einem Paar von Bits wird durch CLK in dem Latch **304** zwischengespeichert. Bei dem nächsten Taktimpuls von CLK wird das erste Bit in dem Latch **306** zwischengespeichert, und das nächste aus der Zufallsbitquelle **202** ausgegebene Bit, wird in dem Latch **304** zwischengespeichert. Wenn beide Bits in einem Bitpaar identisch sind, deaktiviert das XOR-Gatter **310** ACCEPT auf eine Null; wenn beide Bits in einem Bitpaar nicht identisch sind, aktiviert das XOR-Gatter **310** ACCEPT auf Eins.

[0046] Das transparente Latch **308** speichert CLK von der Zufallsbitquelle **302** zwischen, um das UND-Gatter **309** zu takten, so dass STB nur dann aktiviert wird, wenn ACCEPT aktiviert wird, und ein sich nicht überschneidendes Bitpaar in den Latches **304** und **306** gespeichert wird. Wenn STB aktiviert wird, speichert das Latch **312** das von dem Latch **306** ausgegebene Bit als die berichtigte Bitausgabe auf der Signalleitung **214** zwischen. In einem alternativen Ausführungsbeispiel kann die Ausgabe des Latches **304** an das Latch **312** bereitgestellt werden. Das Bit kann als Reaktion auf STB durch das Latch **312** zwischengespeichert werden.

[0047] Die Abbildung aus Fig. 4 zeigt ein Flussdiagramm des Betriebs bzw. der Funktionsweise der Arbeitszyklus-Berichtigungseinrichtung **300**. In dem Schritt **400** wird ein erstes Bitpaar von der Zufallsbitquelle **202** in den Latches **304** und **306** zwischengespeichert, wobei das erste Bit in dem Latch **306** gespeichert wird und das zweite Bit in dem Latch **304**. In dem Schritt **402** bestimmt das XOR-Gatter **310**, ob die beiden Bits in dem Paar identisch sind. Wenn das Bitpaar in dem Schritt **204** übereinstimmt, wird ACCEPT deaktiviert, STB deaktiviert und das Bitpaar zurückgewiesen bzw. verworfen. In dem Schritt **402** wird keines der Bits durch das Latch **312** zwischengespeichert und an die Signalleitung **214** ausgegeben. Wenn hingegen in dem Schritt **402** bestimmt wird, dass sich die beiden Bits voneinander unterscheiden, wird in dem Schritt **406** das erste Bit als das Ausgabebit verwendet.

[0048] In dem nächsten Schritt **408** wird das nächste sich überschneidende Bitpaar von der Zufallsbitquelle **202** gesammelt und dar Ablauf wiederholt sich ab Schritt **402**. Der Ablauf wird wiederholt, bis alle Ausgangbitpaar von der Zufallsbitquelle verarbeitet worden sind. Alle Ausgangsbits von der Zufallsbitquelle, die nicht zu Paaren zusammengefasst sind, können nicht verarbeitet werden und werden somit zurückgewiesen. Hiermit wird festgestellt, dass das erste Bit des Paars aus dem Latch **306** zwar als das berichtigte Bit vorgesehen wird, in einem alternativen Verfahren das zweite Bit des Paares aus dem Latch **304** als das berichtigte Bit bereitgestellt wird.

[0049] Die Abbildung aus Fig. 5 zeigt eine weitere Darstellung des Betriebs der Arbeitszyklus-Berichtigungseinrichtung **300**. Die Abbildung aus Fig. 5 zeigt, dass das zweite Bitpaar (Bits 2 und 3) und das dritte Bitpaar (Bits 4 und 5) berichtigte Bits erzeugen, während das erste Bitpaar (Bits 0 und 1) und das vierte Bitpaar (Bits 6 und 7) dies nicht machen.

[0050] Die Zufallsbitquelle **202** kann Bits ausgeben, die eine Autokorrelation erster Ordnung zwischen nacheinander erzeugten Bits aufgrund der Beschaffenheit der Latches, der Logikgatter und anderer Schaltungselemente aufweisen. Obgleich eine Arbeitszyklus-Berichtigungseinrichtung wie etwa gemäß den Abbildungen der Fig. 2 und Fig. 3 ein im Wesentlichen zufälliges bzw. wahlfreies Bitmuster mit einem im Wesentlichen einheitlichen Arbeitszyklus ausgeben kann, nimmt die Wahrscheinlichkeit dafür zu, dass die Ausgabe der Arbeitszyklus-Berichtigungseinrichtung näher an einem einheitlichen Arbeitszyklus ist, wenn die Autokorrelation zwischen den von der Zufallsbitquelle ausgegebenen Bits niedriger ist. Das heißt, die Wahrscheinlichkeit, dass die Ausgabe der Arbeitszyklus-Berichtigungseinrichtung näher an einem einheitlichen Arbeitszyklus liegt, nimmt zu, wenn die Bits in dem Zufallsbitstrom zueinander in keinem Verhältnis stehen.

[0051] Wie dies in der Abbildung aus Fig. 5 dargestellt ist, kann die Arbeitszyklus-Berichtigungseinrichtung **300** aus Fig. 3 beeinflusst werden durch die Autokorrelation erster Ordnung zwischen den Bits in dem Zufallsbitstrom, der durch die Zufallsbitquelle **202** ausgegeben wird, wenn die Arbeitszyklus-Berichtigungseinrichtung **300** konsekutive Bitpaare bearbeitet. Die Abbildung aus Fig. 6 zeigt ein weiteres Ausführungsbeispiel der Arbeitszyklus-Berichti-

gungseinrichtung **600**, das die Auswirkung der Autokorrelation erster Ordnung zwischen den Bitpaarausgaben durch die Zufallsbitquelle **202** reduziert. Die Arbeitszyklus-Berichtigungseinrichtung **600** reduziert die Autokorrelation erster Ordnung, indem ein Bit von der Zufallsbitquelle **202** verworfen wird, sobald ein Bitpaar detektiert wird, das ein berichtigtes Bit erzeugt.

[0052] Die Arbeitszyklus-Berichtigungseinrichtung **600** entspricht der Arbeitszyklus-Berichtigungseinrichtung **300**, mit der Ausnahme, dass das transparente Latch **308** der Validierungslogik durch einen Modulo-2-Zähler **602** ersetzt wird. Die Funktionsweise der Arbeitszyklus-Berichtigungseinrichtung **600** ist in der Abbildung aus Fig. 7 veranschaulicht. In dem Schritt **700** wird ein erstes Bitpaar von der Zufallsbitquelle **202** in den Latches **304** und **306** bei den Zählwerten 0 und 1 des Modul-2-Zählers **602** zwischengespeichert. In dem Schritt **702** bestimmt das XOR-Gatter **310**, ob die beiden Bits des Paares übereinstimmen. Wenn die Bits des Paares identisch sind, wird in dem Schritt **704** ACCEPT deaktiviert, STB deaktiviert und das Bitpaar zurückgewiesen oder verworfen. In dem Schritt **704** wird kein Bit durch das Latch **312** zwischengespeichert und auf die Signalleitung **214** ausgegeben. Wenn die Bits in dem Schritt **702** nicht übereinstimmen, wird in dem Schritt **704** ACCEPT beim Zählwert 1 aktiviert, STB wird aktiviert und das erste Bit (oder alternativ das zweite Bit) wird durch das Latch **312** ausgegeben. STB wird ferner in den Modulo-2-Zähler **602** zurückgeführt, so dass bei einer Aktivierung von STB der Modulo-2-Zähler **602** einen Zählwert überspringt und die nächsten beiden Taktzyklen niedrig hält (d.h. beide mit einem Zählwert 0). Dies bewirkt, dass die Arbeitszyklus-Berichtigungseinrichtung **600** in dem Schritt **807** das nächste Bit in dem Zufallsbitstrom von der Zufallsbitquelle **202** verwirft. Dies erfolgt, da es, obwohl das nächste Bit in das Latch **304** geladen wird, durch das Latch **306** getaktet wird, bevor der Modulo-2-Zähler **602** zum nächsten Mal den Zählwert 1 an das AND-Gatter **309** ausgibt. In dem Schritt **708** wird das nächste sich überschneidende Bitpaar der Zufallsbitquelle **202** erfasst bzw. gesammelt, und wobei sich der Ablauf ab Schritt **702** wiederholt. Der Ablauf wiederholt sich, bis alle Ausgangsbitpaare von der Zufallsbitquelle verarbeitet worden sind.

[0053] Die Abbildung aus Fig. 8 zeigt eine weitere Darstellung des Betriebs der Arbeitszyklus-Berichtigungseinrichtung **600**. Wenn die Bits des ersten Bitpaares (Bits 0 und 1) übereinstimmen, werden die Bits verworfen, und für das erste Bitpaar wird kein berichtigtes Bit erzeugt. Darüber hinaus wird STB nicht aktiviert und der Modulo-2-Zähler **602** überspringt keinen Zählwert. Die Bits des zweiten Bitpaares (Bits 2 und 3) stimmen nicht überein, und die Arbeitszyklus-Berichtigungseinrichtung **600** gibt eine Null aus, und wobei sie danach bewirkt, dass das nächste Bit, das Bit 4, verworfen wird, da der Modulo-2-Zähler **600** einen Zählwert überspringt. Die Bits des dritten Bitpaares (Bits 5 und 6) stimmen ebenfalls nicht überein. Die Arbeitszyklus-Berichtigungseinrichtung **600** gibt eine Eins aus und bewirkt, dass das nächste Bit, das Bit 7, verworfen wird, da der Modulo-2-Zähler **602** einen Zählwert überspringt. Die Bits des letzten Bitpaares (Bits 8 und 9) stimmen überein und werden verworfen.

[0054] Das Verwerfen der Bits 4 und 7 reduziert die Autokorrelation erster Ordnung zwischen dem zweiten Bitpaar (Bits 2 und 3) und dem dritten Bitpaar (Bits 5 und 6) und zwischen dem dritten Bitpaar und dem vierten Bitpaar (Bits 8 und 9). Der berichtigte Bitstrom wird beeinflusst durch eine weniger signifikante Autokorrelation zweiter Ordnung zwischen dem zweiten Bitpaar und dem dritten Bitpaar und zwischen dem dritten Bitpaar und dem vierten Bitpaar, die durch die Zufallsbitquelle **202** ausgegeben werden.

[0055] Die Arbeitszyklus-Berichtigungseinrichtung **600** reduziert die Autokorrelation erster Ordnung zwischen den durch die Zufallsbitquelle erzeugten Bits. Sie kann jedoch eine gewisse Uneinheitlichkeit in den Arbeitszyklus des berichtigten Bitstroms einfügen, da die einzelnen verworfenen Bits (z.B. die Bits 4 und 7 aus Fig. 8) eine uneinheitliche Verteilung in dem Zufallsbitstrom aufweisen können.

[0056] Die Abbildung aus Fig. 9 zeigt ein weiteres Ausführungsbeispiel einer Arbeitszyklus-Berichtigungseinrichtung **900**, welche die Auswirkungen der Autokorrelation erster Ordnung zwischen den durch die Zufallsbitquelle **202** ausgegebenen Bitpaaren reduziert. Die Arbeitszyklus-Berichtigungseinrichtung **900** reduziert die Autokorrelation erster Ordnung, indem Bits von der Zufallsbitquelle **202** verworfen werden, die bei anderen speziellen Zählwerten eines Modulo-5-Zählers verschoben werden.

[0057] Die Arbeitszyklus-Berichtigungseinrichtung **900** entspricht der Arbeitszyklus-Berichtigungseinrichtung **300**, mit der Ausnahme, dass das transparente Latch **308** der Validierungslogik ersetzt wird durch den Modulo-5-Zähler **902**, die Inverter **904**, **906** und **908**, die AND-Gatter **910** und **912** und das NOR-Gatter **914**. Der Modulo-5-Zähler **902** weist drei binäre Ausgabebits C0, C1 und C2 auf. Das AND-Gatter **910** ist ein AND-Gatter mit drei Eingängen, das einen ersten Eingang aufweist, der über den Inverter **904** mit C2 gekoppelt ist, mit einem zweiten Eingang, der mit C1 gekoppelt ist, und mit einem dritten Eingang, der über den Inverter **908** mit C0 gekoppelt ist. Das AND-Gatter **912** ist ein AND-Gatter mit drei Eingängen, das einen ersten Eingang aufweist, der mit C2 gekoppelt ist, mit einem zweiten Eingang, der über den Inverter **906** mit C1 gekoppelt ist, und mit einem dritten Eingang, der über den Inverter **908**

mit C0 gekoppelt ist. Das NOR-Gatter **914** empfängt die Ausgänge der UND-Gatter **910** und **912** und steuert einen Eingang des UND-Gatters **309**.

[0058] Der Betrieb der Arbeitszyklus-Berichtigungseinrichtung **900** ist in der Abbildung aus Fig. 10 veranschaulicht. In dem Schritt **1000** und bei dem Zählwert 0 des Modulo-5-Zählers **902** wird ein erstes Bit in das Latch **304** geladen. Das Bit wird verworfen, da das Signal auf der Signalleitung **915** für die Zählwerte 0 und 1 nicht aktiviert bzw. behauptet wird. In dem Schritt **1002** wird ein erstes Bitpaar von der Zufallsbitquelle **202** bei den Zählwerten 1 und 2 in den Latches **304** und **306** zwischengespeichert. Dies bewirkt, dass das erste Bit aus der Berichtigungseinrichtung **1000** verworfen wird. In dem Schritt **1004** und bei dem Zählwert 2 bestimmt das XOR-Gatter **310**, ob die beiden Bits des Paars übereinstimmen. Wenn die Bits des Paares übereinstimmen, wird in dem Schritt ACCEPT deaktiviert, STB deaktiviert und das Bitpaar zurückgewiesen bzw. verworfen. In dem Schritt **1006** wird kein Bit durch das Latch **312** zwischengespeichert und an die Signalleitung **214** ausgegeben. Wenn die Bits in dem Schritt **1004** und bei dem Zählwert 2 nicht übereinstimmen, so wird ACCEPT in dem Schritt **1008** aktiviert, STB aktiviert und das erste Bit (oder alternativ das zweite Bit) durch das Latch **312** ausgegeben.

[0059] In dem Schritt **1010** wird ein zweites Bitpaar von der Zufallsbitquelle **202** bei den Zählwerten 3 und 4 in den Latches **304** und **306** zwischengespeichert. In dem Schritt **1012** und bei dem Zählwert 4 bestimmt das XOR-Gatter **310**, ob die beiden Bits des Paares übereinstimmen. Wenn die Bits des Paares übereinstimmen, so wird in dem Schritt **1014** ACCEPT deaktiviert, STB deaktiviert und das Bitpaar zurückgewiesen bzw. verworfen. Wenn die Bits in dem Schritt **1012** und bei dem Zählwert 4 nicht übereinstimmen, wird in dem Schritt **1016** ACCEPT aktiviert, STB aktiviert und das erste Bit (oder alternativ das zweite Bit) durch das Latch **312** ausgegeben. Dieser Vorgang wird wiederholt, bis alle Ausgangsbitpaare der Zufallsbitquelle verarbeitet worden sind.

[0060] Die Abbildung aus Fig. 11 zeigt eine weitere Darstellung des Betriebs der Arbeitszyklus-Berichtigungseinrichtung **900**. Das erste Bit, das Bit 0, wird in die Berichtigungseinrichtung **900** geladen, wobei es verworfen wird, wenn das erste Bitpaar geladen wird. Wenn die Bits des ersten Paares (Bits 1 und 2) übereinstimmen, werden die Bits verworfen, und es wird für dieses Bitpaar kein berichtigtes Bit erzeugt. Die Bits des zweiten Bitpaares (Bits 3 und 4) stimmen nicht überein, und die Arbeitszyklus-Berichtigungseinrichtung **900** gibt eine Null aus. Der Modulo-5-Zähler **902** gibt danach den Zählwert 0 zurück, so dass das Bit 5 schließlich verworfen wird. Die Bits des dritten Bitpaares (Bits 6 und 7) stimmen nicht überein, und die Arbeitszyklus-Berichtigungseinrichtung **900** gibt eine Eins aus. Die Bits des letzten Bitpaares (Bits 8 und 9) stimmen überein und werden verworfen.

[0061] Das Verwerfen der Bits 0 und 5 reduziert die Autokorrelation der ersten Ordnung zwischen dem zweiten Bitpaar (Bits 3 und 4) und dem dritten Bitpaar (Bits 6 und 7). Der berichtigte Bitstrom wird beeinflusst durch eine weniger signifikante Autokorrelation zweiter Ordnung zwischen den zweiten und dritten Bitpaaren. Die verworfenen Bits des Zählwertes 0 (Zählwert 5, Zählwert 10, etc.) sind einheitlich bzw. gleichmäßig in dem Zufallsbitstrom verteilt, und somit kann deren Ausschluss zu einem ungefähr einheitlichen Arbeitszyklus für den berichtigten Bitstrom führen.

[0062] Die Abbildung aus Fig. 12 zeigt ein weiteres Beispiel der Arbeitszyklus-Berichtigungseinrichtung **1200**, welche die Auswirkungen der Autokorrelation erster Ordnung zwischen durch die Zufallsbitquelle **202** ausgegebenen Bitpaaren reduziert. Die Arbeitszyklus-Berichtigungseinrichtung **1200** reduziert die Autokorrelation erster Ordnung, indem Bits von der Zufallsbitquelle **202** verworfen werden, die sich zwischen Bitpaaren befinden, die verglichen werden.

[0063] Die Arbeitszyklus-Berichtigungseinrichtung **1200** entspricht der Arbeitszyklus-Berichtigungseinrichtung **300**, mit der Ausnahme, dass das transparente Latch **308** der Validierungslogik durch den Modulo-3-Zähler **1202** ersetzt wird, der ein hohes Logiksignal auf der Signalleitung **1204** nu bei dem Zählwert 2 ausgibt.

[0064] Der Betrieb der Arbeitszyklus-Berichtigungseinrichtung **1200** ist in der Abbildung aus Fig. 13 veranschaulicht. In dem Schritt **1300** und bei dem Zählwert 0 des Modulo-3-Zählers **902** wird ein erstes Bit in das Latch **304** geladen. Das Bit wird verworfen, da das Signal auf der Signalleitung **1204** für die Zählwerte 0 und 1 nicht aktiviert wird. In dem Schritt **1302** wird ein erstes Bitpaar der Zufallsbitquelle **202** bei den Zählwerten 1 und 2 in den Latches **304** und **306** zwischengespeichert. Dies bewirkt, dass das erste Bit aus der Berichtigungseinrichtung **1200** verworfen wird. In dem Schritt **1304** und bei dem Zählwert 2 bestimmt das XOR-Gatter **310**, ob die beiden Bits des Paares übereinstimmen. Wenn die Bits des Paares übereinstimmen, so wird in dem Schritt **1306** ACCEPT deaktiviert, STB deaktiviert und das Bitpaar zurückgewiesen bzw. verworfen. In dem Schritt **1306** wird kein Bit durch das Latch **312** zwischengespeichert und an die Signalleitung **214** ausgegeben. Wenn die Bits in dem Schritt **1304** und bei dem Zählwert 2 nicht übereinstimmen, so wird in dem Schritt **1308** ACCEPT aktiviert, STB aktiviert und das erste Bit (oder alternativ das zweite Bit) durch Ausgabe durch das Latch **312** ausgegeben. Der Ablauf wiederholt sich, bis alle Ausgabebitpaare aus der Zufallsbit-

quelle verarbeitet worden sind.

[0065] Die Abbildung aus Fig. 14 zeigt eine weitere Darstellung des Betriebs der Arbeitszyklus-Berichtigungseinrichtung **1200**. Das erste Bit, das Bit 0, wird in die Berichtigungseinrichtung **1200** geladen, jedoch verworfen, wenn das erste Bitpaar geladen wird. Wenn die Bits des ersten Bitpaares (Bits 1 und 2) übereinstimmen, werden die Bits verworfen, und es wird für dieses Bitpaar kein berichtigtes Bit erzeugt. Das vierte Bit, Bit 3, wird in die Berichtigungseinrichtung **1200** geladen, wobei es jedoch verworfen wird, wenn das zweite Bitpaar geladen wird. Die Bits des zweiten Bitpaars (Bits 4 und 5) stimmen nicht überein, und die Arbeitszyklus-Berichtigungseinrichtung **1200** gibt eine Null aus. Das siebte Bit, Bit 6, wird in die Berichtigungseinrichtung **1200** geladen, wobei es jedoch verworfen wird, wenn das dritte Bitpaar geladen wird. Die Bits des dritten Bitpaares (Bits 7 und 8) stimmen nicht überein, und die Arbeitszyklus-Berichtigungseinrichtung **1200** gibt eine Null aus. Das zehnte Bit, Bit 9, wird in die Berichtigungseinrichtung **1200** geladen, jedoch verworfen, wenn das vierte Bitpaar geladen wird. Die Bits des vierten Bitpaares (Bits 10 und 11) sind identisch und werden verworfen.

[0066] Das Verwerfen der Bits 0, 3, 6, 9, etc. reduziert die Autokorrelation erster Ordnung zwischen den verglichenen Bitpaaren. Der berichtigte Bitstrom wird durch eine weniger signifikante Autokorrelation zweiter Ordnung zwischen den verglichenen Bitpaaren beeinflusst. Die verworfenen Bits der Zählwerte 0, 3, 6, 9, etc. des Modulo-3-Zählers **1202** sind in dem Zufallsbitstrom gleichmäßig bzw. einheitlich verteilt, und deren Ausschluss kann zu einem ungefähr einheitlichen Arbeitszyklus des berichtigten Bitstroms führen.

[0067] Für die vorstehend veranschaulichten Arbeitszyklus-Berichtigungseinrichtungen wurde zwar beschrieben, dass sie zwei konsekutive Bits in dem Zufallsbitstrom einer Zufallsbitquelle vergleichen, wobei aber in alternativen Ausführungsbeispielen auch nicht konsekutive Bits verglichen werden können. Zum Beispiel können zusätzliche Speicherschaltungen zwischen den Speicherschaltungen **204** und **206** eingefügt werden, oder jede Speicherschaltung kann durch andere Takte oder unterschiedliche Taktflanken getaktet werden.

[0068] Darüber hinaus ist die Ausgangsschaltung **212** so dargestellt, dass sie entweder die Ausgabe der Speicherschaltung **204** oder **206** empfängt. In einem alternativen Ausführungsbeispiel kann die Vergleichslogik **208** eine Logik aufweisen, welche die an die Ausgangsschaltung **212** bereitgestellten Daten als Reaktion auf die Daten bestimmt, die in den Speicherschaltungen **204** und **206** gespeichert sind.

[0069] Die zur Erzeugung einer im Wesentlichen einheitlichen Verteilung von Einsen und Nullen von einer Zufallsbitquelle beschriebenen Arbeitszyklus-Berichtigungseinrichtungen können in Verbindung mit einem Zufallszahlengenerator zum Codieren und Decodieren von Nachrichten eingesetzt werden, die über ein Computernetzwerk übermittelt werden. Die Abbildung aus Fig. 15 zeigt ein Blockdiagramm eines Computernetzwerks zur Übertragung verschlüsselter Nachrichten unter Verwendung eines beliebigen der vorstehend beschriebenen Ausführungsbeispiele. Das Netzwerk **1500** weist einen sendenden Host-Computer **1502** auf, der über ein Netzwerk mit einem empfangenden Host-Computer **1504** gekoppelt ist. Sowohl der sendende Host-Computer als auch der empfangende Host-Computer weisen Netzwerkschnittstellenvorrichtungen auf, welche die physikalischen und logischen Verbindungen zwischen den Host-Computersystemen und dem Netzwerkmedium bereitstellen. Beide Host-Computer weisen ferner Verschlüsselungs-/Entschlüsselungsschaltungen auf, die verschiedene kryptographische Funktionen ausführen, um Datenübertragungen zu schützen. Der sendende Host **1502** weist eine Verschlüsselungs-/Entschlüsselungsschaltung **1506** auf, und der empfangende Host **1504** weist eine Verschlüsselungs-/Entschlüsselungsschaltung **1507** auf. Die Verschlüsselungs-/Entschlüsselungsschaltungen **1506** und **1507** weisen beide entsprechende Zufallszahlengeneratoren **1508** und **1509** auf, die ein beliebiges der Ausführungsbeispiele aus den Fig. 2, Fig. 3, Fig. 6, Fig. 9 oder Fig. 12 einsetzen. Die Zufallszahlengeneratoren werden eingesetzt, um Paare aus öffentlichen/privaten Schlüsseln in Systemen mit öffentlichen/privaten Schlüsseln zu erzeugen.

[0070] Verschiedene Verfahren der Datenverschlüsselung können in dem Netzwerk **1500** eingesetzt werden, um Kommunikationen zwischen dem sendenden Host **1502** und dem empfangenden Host **1504** zu schützen bzw. zu sichern. In einem Ausführungsbeispiel verwendet das Netzwerk **1500** ein kryptographisches System mit öffentlichem Schlüssel (asymmetrisch). In einem System mit öffentlichem Schlüssel werden zwei unterschiedliche Schlüssel eingesetzt. Ein Schlüssel wird von dem Sender verwendet, um eine Nachricht zu codieren, und der andere Schlüssel wird von dem Empfänger verwendet, um die codierte Nachricht zu entschlüsseln. Bei diesem System kann der (öffentliche) Verschlüsselungsschlüssel weit verbreitet veröffentlicht werden, während der (private) Entschlüsselungsschlüssel geheim gehalten werden muss, so dass nur der vorgesehene Empfänger die Nachricht decodieren kann. Die öffentlichen und privaten Schlüssel werden für gewöhnlich gemeinsam von sehr großen Prim- und Zufallszahlen abgeleitet. Effektive Zufallszahlengeneratoren sind somit erforderlich, um wirklich zufällig bzw. wahlfreie Schlüsselpaare zu erzeugen.

[0071] Bei einem Beispiel einer Datenübertragung

unter Verwendung eines öffentlichen Schlüsselsystems setzt der sendende Host **1502** eine Nachricht M für die Übertragung an einen empfangenden Host **1504** zusammen. Die beiden für die Übermittlung verwendeten Schlüssel umfassen den öffentlichen Schlüssel des Empfängers (PuK_R) und den privaten Schlüssel des Empfängers (PrK_R). Der Empfänger wählt für gewöhnlich einen öffentlichen Schlüssel aus dem öffentlich verfügbaren Register von Schlüsseln aus und leitet den privaten Schlüssel aus dem öffentlichen Schlüssel über ein nur dem Empfänger bekanntes Transformationsverfahren ab. Die Korrelation zwischen dem öffentlichen Schlüssel und dem privaten Schlüssel ist somit allgemein geheim und sicher. Unter Verwendung des öffentlichen Schlüssels codiert der sendende Host **1502** die Nachricht über die Verschlüsselungs-/Entschlüsselungsschaltung **1506**, so dass eine codierte Nachricht M' erzeugt wird. Nach der Codierung kann nur der entsprechende private Schlüssel die Nachricht decodieren. Nach dem Empfang der Nachricht decodiert der empfangende Host **1504** die Nachricht M' mit dem privaten Schlüssel, um die ursprüngliche Nachricht wiederherzustellen.

[0072] In einem Ausführungsbeispiel weist die Verschlüsselungs-/Entschlüsselungsschaltung **1507** in dem empfangenden Host **1504** einen Zufallszahlengenerator **1509** auf, der ein beliebiges der Ausführungsbeispiele aus den Fig. 2, Fig. 3, Fig. 6, Fig. 9 oder Fig. 12 einsetzt. Diese Technik stellt sicher, dass die Bitverteilung von dem Zufallszahlengenerator **1509** ausreichend einheitlich und wahlfrei ist, so dass keine einheitliche Korrelation zwischen den durch den empfangenen Host **1504** erzeugten privaten Schlüssels und öffentlichen Schlüsseln existiert. Wie dies in dem Netzwerk **1500** dargestellt ist, weist die Verschlüsselungs-/Entschlüsselungsschaltung **1506** in dem sendenden Host **1502** ebenfalls einen Zufallszahlengenerator **1508** auf, der ein beliebiges der Ausführungsbeispiele der Fig. 2, Fig. 3, Fig. 6, Fig. 9 oder Fig. 12 einsetzt. Dies ermöglicht es, dass der sendende Host **1502** sichere private Schlüssel und öffentliche Schlüssel erzeugt, wenn er eine Übermittlung eines öffentlichen Schlüssels einsetzt. Ein hohes Maß der Zufälligkeit bzw. Wahlfreiheit ist erforderlich, um das Schlüsselpaar zu erzeugen, um eine nicht umfassende Suche nach privaten Schlüsseln übermäßig schwierig zu gestalten.

[0073] In einem alternativen Ausführungsbeispiel verwendet das Netzwerk **1500** ein System mit einem einzigen Schlüssel (symmetrisch) für die Ausführung kryptographischer Funktionen. Bei einem System mit einem Schlüssel wird ein Schlüssel sowohl von dem Sender zum Verschlüsseln der Nachricht als auch von dem Empfänger zum entschlüsseln der codierten Nachricht verwendet. Das System basiert auf der Geheimhaltung des Schlüssels. Somit ist ein sicherer Prozess erforderlich, damit der Schlüssel nur zwischen dem Sender und dem Empfänger und keinem Dritten offenbart wird. Bei diesem Ausführungsbeispiel werden für gewöhnlich unterschiedliche Schlüssel für verschiedene Nachrichtentransaktionen verwendet. Somit erfordert die Erzeugung verschiedener Schlüssel einen wahlfreien Prozess, um sicherzustellen, dass ein Schlüssel, der für eine Nachrichtentransaktion verwendet wird, aus keinem Schlüssel bestimmt werden kann, der für eine andere Nachrichtentransaktion verwendet wird. Bei diesem System werden die Zufallsgeneratoren in den Verschlüsselungs-/Entschlüsselungsschaltungen in jedem der Host-Computer des Netzwerks **1500** verwendet, um die zufälligen Schlüsselmuster zum Codieren und Decodieren der zwischen den Host-Computern übertragenen Nachrichtendaten zu erzeugen.

[0074] Hiermit wird festgestellt, dass neben den beschriebenen Ausführungsbeispielen der vorliegenden Erfindung in Bezug auf Verschlüsselungssysteme mit einem Schlüssel und öffentlichem/privatem Schlüssel auch Ausführungsbeispiele für die Zufallszahlenerzeugung in andersartigen kryptographischen Systemen für sichere Computernetzwerkanwendungen verwendet werden können. Ferner können die in der Abbildung aus Fig. 15 veranschaulichten Verschlüsselungs-/Entschlüsselungsschaltungen in Systemen zur sicheren Datenübertragung für die Ausführung verschiedener kryptographischer Funktionen eingesetzt werden, wie etwa der Codierung und Decodierung von Nachrichten, der Authentifizierung von übermittelten Nachrichten, der Verifizierung digitaler Signaturen und anderer ähnlicher Funktionen.

[0075] Vorstehend wurde eine Schaltung zum Erzeugen eines Zufallszahlengenerators mit einheitlichem Arbeitszyklus beschrieben. Die vorliegende Erfindung wurde vorstehend in Bezug auf bestimmte Ausführungsbeispiele beschrieben, wobei jedoch ersichtlich ist, dass verschiedene Modifikationen und Abänderungen in Bezug auf diese Ausführungsbeispiele möglich sind, ohne dabei vom weiteren Umfang der Erfindung abzuweichen, der in den Ansprüchen ausgeführt ist. Folglich dienen die Beschreibung und die Zeichnungen Zwecken der Veranschaulichung und haben keine einschränkende Funktion.

Patentansprüche

1. Verfahren zum Erzeugen eines Bitstroms mit einer Korrelation von ungefähr Null aus einem wahlfreien Bitstromausgang einer wahlfreien Bitquelle, wobei das Verfahren folgendes umfasst:
das Vergleichen (**208**) der Bits jedes Paares der Mehrzahl von Bitpaaren, welche miteinander den wahlfreien Bitstrom bilden;
das Verwerfen (**210**) jedes Paars der genannten Mehrzahl von Bitpaaren, wenn die Bitpaare identisch sind;

das Ausgeben (**212**) eines Bits jedes Paars der genannten Mehrzahl von Bitpaaren, wenn die Bitpaare nicht identisch sind; und
das Verwerfen (**708**) jedes X-ten Bits, das durch eine Modulo-X-Zählung des Taktsignals der wahlfreien Bitquelle bestimmt wird, aus dem Ausgangs-Bitstrom, um die Autokorrelation zwischen den Bits zu reduzieren.

2. Verfahren nach Anspruch 1, wobei das Ausgeben eines Bits jedes Paars der Mehrzahl von Bitpaaren das Ausgeben des ersten Bits des Bitpaares umfasst.

3. Verfahren nach Anspruch 1, wobei das Ausgeben eines Bits jedes Paars der Mehrzahl von Bitpaaren das Ausgeben des zweiten Bits des Bitpaares umfasst.

4. Verfahren nach Anspruch 1, wobei dieses ferner das synchrone Verriegeln jedes Paars der Mehrzahl von Bitpaaren in einem Paar seriell gekoppelter Latches (**304**, **306**) umfasst.

5. Verfahren nach Anspruch 1, wobei das Verfahren ferner folgendes umfasst:
das Vergleichen von Bits eines anderen folgenden Bitpaares; und
das Verwerfen eines Bits in einem wahlfreien Bitstrom zwischen den beiden Paaren verglichener Bits.

6. Verfahren nach Anspruch 1, wobei das Verfahren folgendes umfasst:
das Verwerfen des anderen Bits in dem wahlfreien Bitstrom, unabhängig davon, ob das Bitpaar identisch oder nicht identisch ist.

7. Verfahren nach einem der vorstehenden Ansprüche beim Einsatz in einem Zufallszahlengenerator, der so betrieben werden kann, dass binäre Zufallszahlen zur Verwendung in einem kryptographischen System erzeugt werden, für sichere Kommunikationen zwischen einer Mehrzahl von Computern in einem Netzwerk.

8. Arbeitszyklus-Berichtigungsschaltung (**200**) zum Erzeugen eines Bitstroms mit einer Korrelation von ungefähr Null aus einem wahlfreien Bitstromausgang einer wahlfreien Bitquelle (**202**), wobei die Arbeitszyklus-Berichtigungsschaltung folgendes umfasst:
eine erste Speicherschaltung (**204**), die so angeordnet ist, dass sie ein erstes Bit jedes Paars einer Mehrzahl von Bitpaaren, welche den wahlfreien Bitstrom bilden, empfängt und speichert;
eine zweite Speicherschaltung (**206**), die mit der ersten Speicherschaltung gekoppelt und so angeordnet ist, dass sie ein zweites Bit jedes Paars der genannten Mehrzahl von Bitpaaren empfängt und speichert;
eine Vergleichsschaltung (**208**), die mit der ersten Speicherschaltung und der zweiten Speicherschaltung gekoppelt und so angeordnet ist, dass die Bits jedes Paars der genannten Mehrzahl von Bitpaaren in dem wahlfreien Bitstrom miteinander verglichen werden;
eine Validierungslogik (**210**), die mit der ersten Speicherschaltung, der zweiten Speicherschaltung und der Vergleichsschaltung gekoppelt ist, wobei die Validierungslogik so angeordnet ist, dass sie jedes der Paare der genannten Mehrzahl von Bitpaaren verwirft, wenn die Bitpaare identisch sind, und wobei ein Bit jedes Paars der genannten Mehrzahl von Bitpaaren ausgegeben wird, wenn die Bitpaare nicht identisch sind; und
eine Zeitsteuerungsschaltung, die mit der Validierungslogik gekoppelt ist, um jedes X-te Bit aus dem durch einen Modulo-X-Zähler (**602**), der mit einem Taktausgang der wahlfreien Bitquelle gekoppelt ist, bestimmten Ausgangsbitstrom zu verwerfen, um die Autokorrelation zwischen den Bits zu reduzieren.

9. Arbeitszyklus-Berichtigungsschaltung nach Anspruch 8, wobei die erste Speicherschaltung und die zweite Speicherschaltung Latches (**304**, **306**) umfassen.

10. Arbeitszyklus-Berichtigungsschaltung nach Anspruch 8, wobei die Vergleichsschaltung ein XOR-Gatter (**310**) umfasst.

11. Arbeitszyklus-Berichtigungsschaltung nach Anspruch 8, wobei diese ferner eine Ausgangsschaltung (**212**) umfasst, die so gekoppelt ist, dass sie den Signalausgang durch die Validierungslogik und das in der ersten Speicherschaltung oder in der zweiten Speicherschaltung gespeicherte Bit empfängt, wobei die Ausgangsschaltung so angeordnet ist, dass sie das Bit mit einer Korrelation von ungefähr Null in dem Bitstrom ausgibt.

12. Arbeitszyklus-Berichtigungsschaltung nach Anspruch 11, wobei die Ausgangsschaltung eine Speicherschaltung (**312**) umfasst.

13. Arbeitszyklus-Berichtigungsschaltung nach Anspruch 8, wobei die Validierungsschaltung folgendes umfasst:
ein transparentes Latch (**308**), das mit einem Taktausgang der wahlfreien Bitquelle gekoppelt ist; und
ein erstes AND-Gatter (**309**) mit einem ersten Eingang, der mit einem Ausgang des transparenten Latches gekoppelt ist, und mit einem zweiten Eingang, der mit einem ersten Ausgang der Vergleichsschaltung gekoppelt ist.

14. Arbeitszyklus-Berichtigungsschaltung nach Anspruch 8, wobei die Validierungslogik ferner folgendes umfasst:
den Modulo-X-Zähler (**602**), der mit dem Taktausgang der wahlfreien Bitquelle gekoppelt ist, wobei X

eine ganze Zahl größer als 1 ist; und
ein zweites AND-Gatter (**309**) mit einem ersten Eingang, der mit einem Ausgang des Modulo-X-Zählers gekoppelt ist, und mit einem zweiten Eingang, der mit einem zweiten Ausgang der Vergleichsschaltung gekoppelt ist.

15. Computernetzwerk (**1500**), das folgendes umfasst:
einen Computer (**1502**);
eine Netzwerkschnittstellenvorrichtung, die so betrieben werden kann, dass sie Nachrichten zwischen dem Computer und einem Netzwerkmedium sendet und empfängt; und
eine Verschlüsselungs-/Entschlüsselungsschaltung (**1506**), die so angeordnet ist, dass sie von dem Computer übertragene Nachrichten codiert und decodiert, wobei die Verschlüsselungs-/Entschlüsselungsschaltung einen Zufallszahlengenerator aufweist, der so angeordnet ist, dass er den wahlfreien Bitstrom erzeugt und an die Arbeitszyklus-Berichtigungsschaltung gemäß der Definition in einem der Ansprüche 8 bis 14 ausgibt.

16. Schaltung nach Anspruch 15, wobei die Verschlüsselungs-/Entschlüsselungsschaltung ferner so angeordnet ist, dass sie durch den Computer übertragene und empfangene Nachrichten unter Verwendung eines schlüsselbasierten kryptographischen Verfahrens codiert und decodiert.

17. Schaltung nach Anspruch 16, wobei das schlüsselbasierte kryptographische Verfahren ein System mit einem Schlüssel darstellt.

18. Schaltung nach Anspruch 17, wobei das schlüsselbasierte kryptographische Verfahren ein System mit einem öffentlichen Schlüssel/einem privaten Schlüssel darstellt.

Es folgen 12 Blatt Zeichnungen

Anhängende Zeichnungen

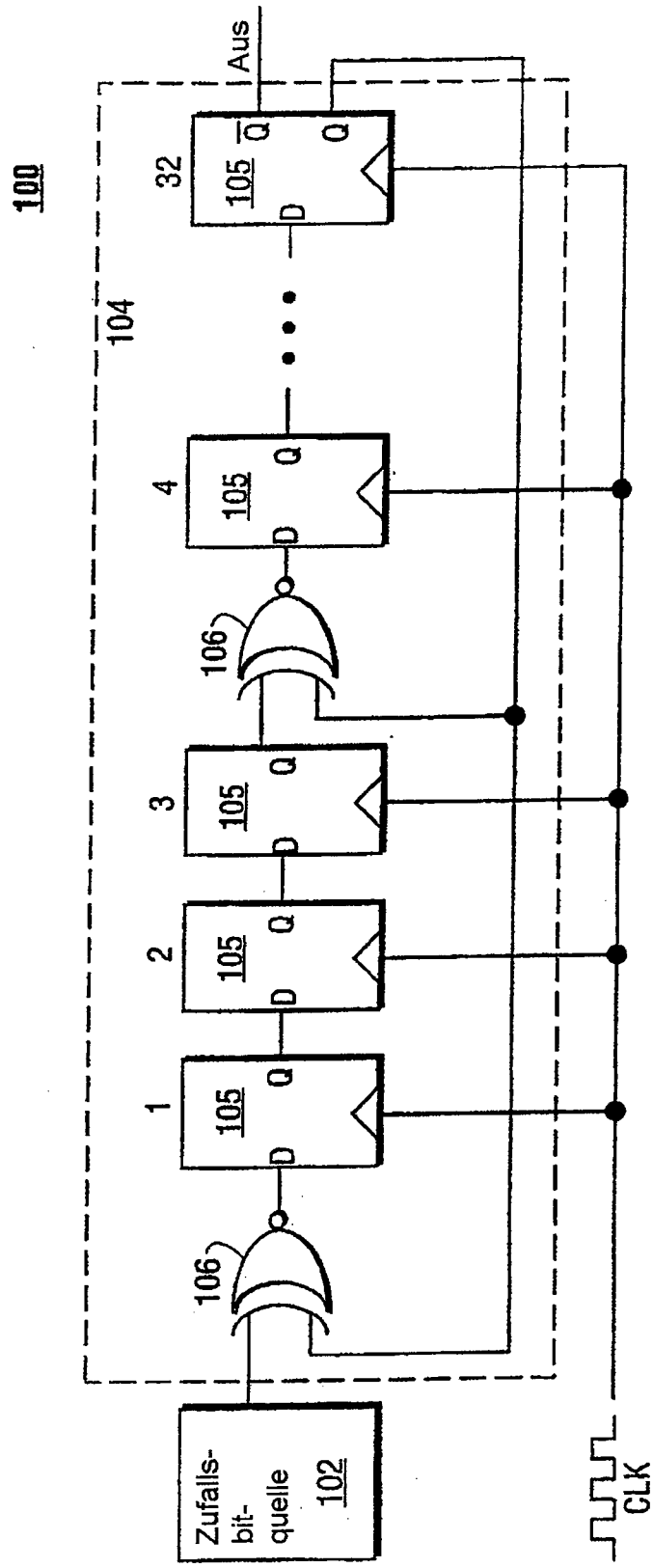

FIG. 1

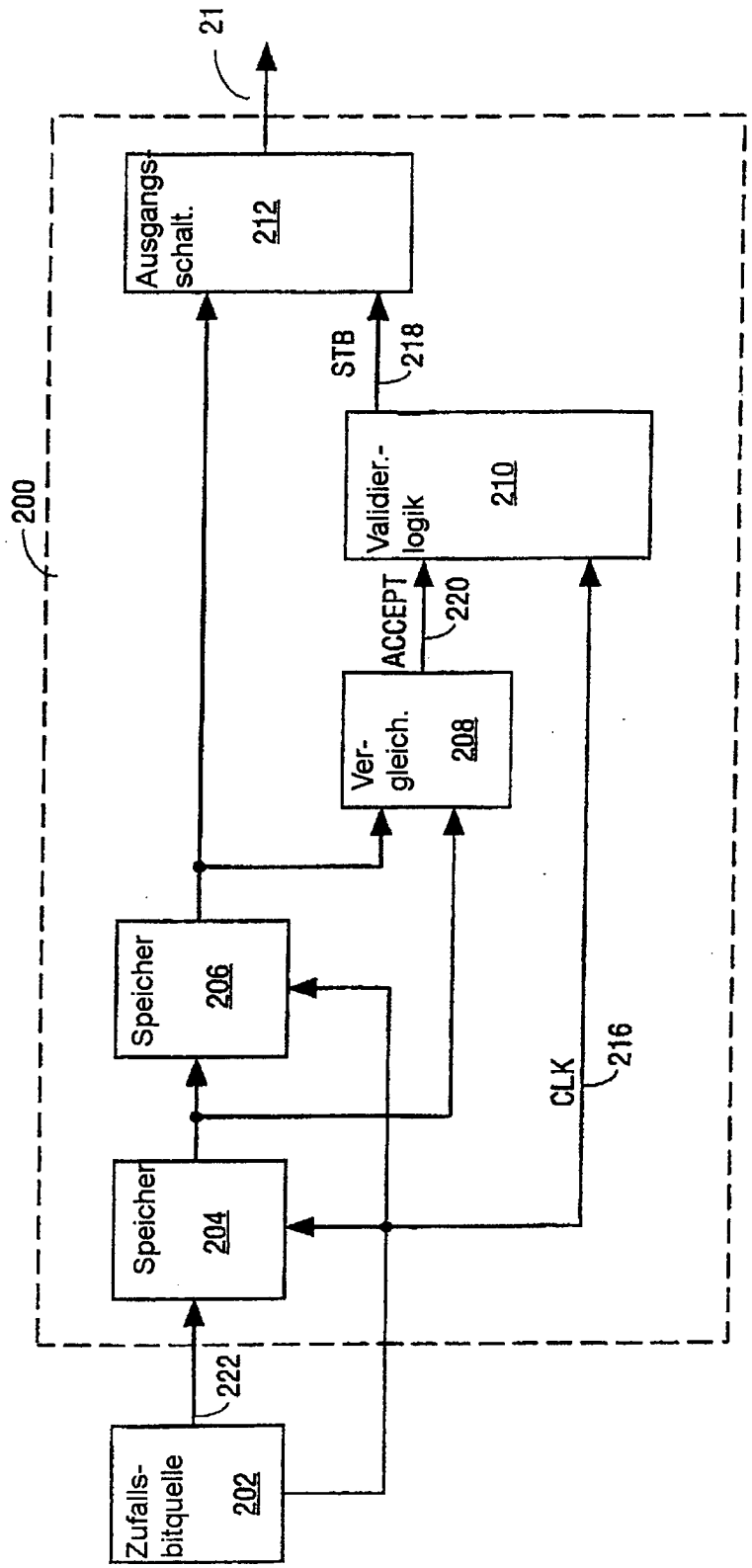
21
Ausgangs-schalt. 212
STB 218
Validier.-logik 210
200
ACCEPT 220
Ver-gleich. 208
Speicher 206
CLK 216
Speicher 204
222
Zufalls-bitquelle 202

FIG. 2

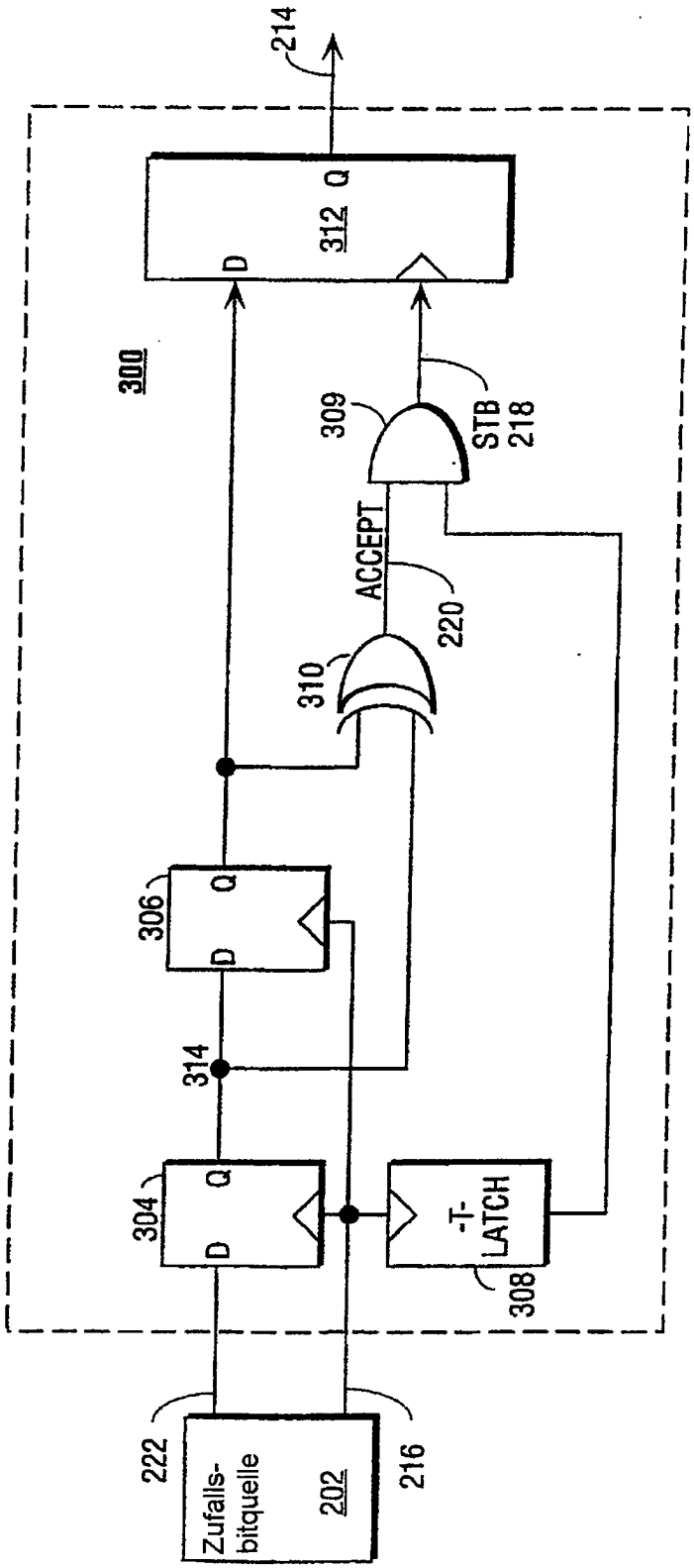
214
312
Q
D
300
309
STB
218
ACCEPT
220
310
306
Q
D
314
304
Q
D
-T-
LATCH
308
222
Zufalls-
bitquelle
202
216

FIG. 3

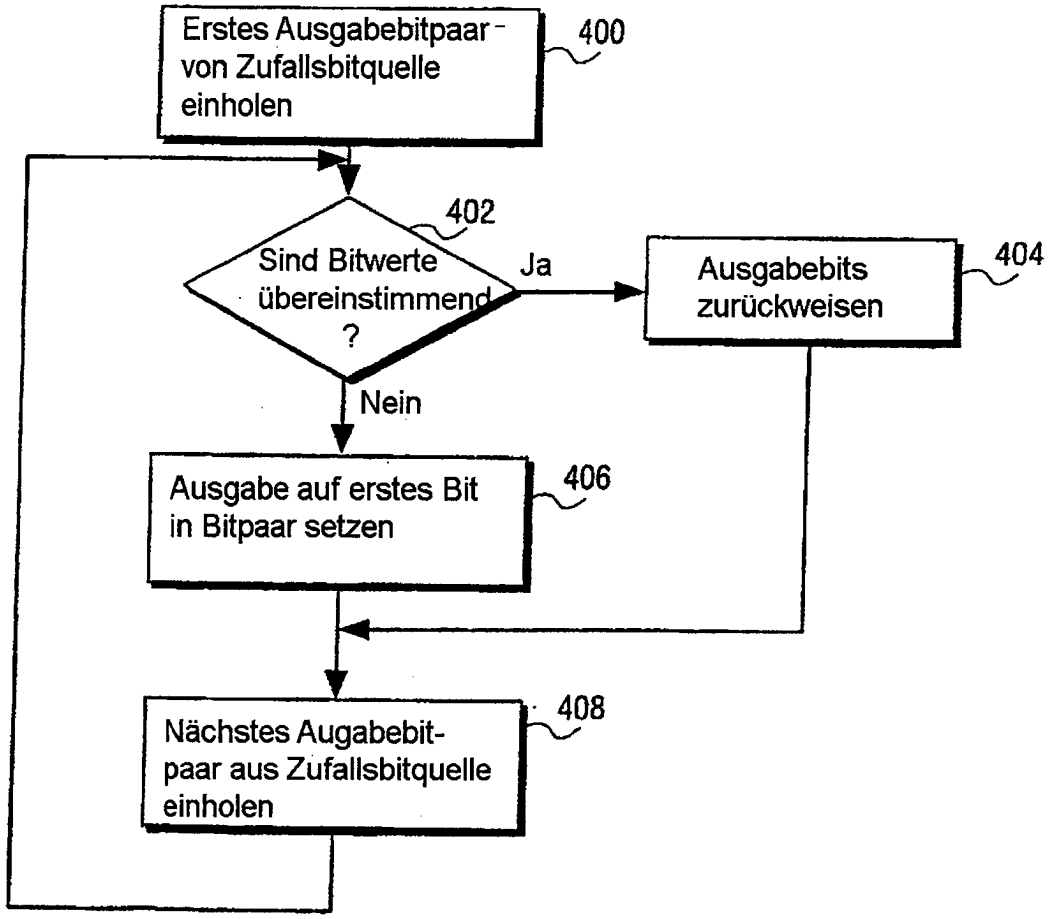
Erstes Ausgabebitpaar von Zufallsbitquelle einholen
400
402
Sind Bitwerte übereinstimmend ?
Ja
Ausgabebits zurückweisen
404
Nein
Ausgabe auf erstes Bit in Bitpaar setzen
406
Nächstes Augabebit-paar aus Zufallsbitquelle einholen
408

FIG. 4

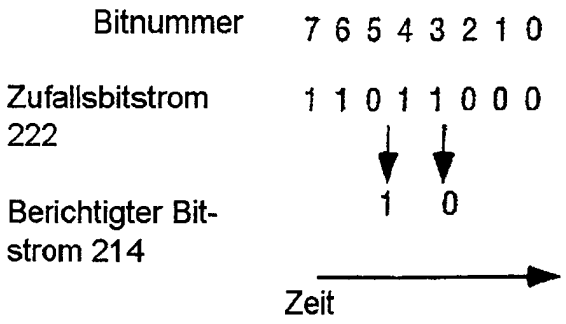

FIG. 5

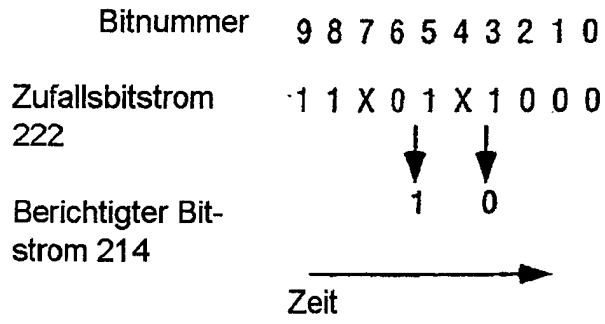

FIG. 8

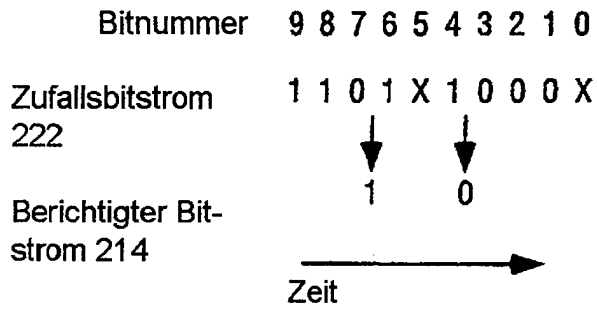

FIG. 11

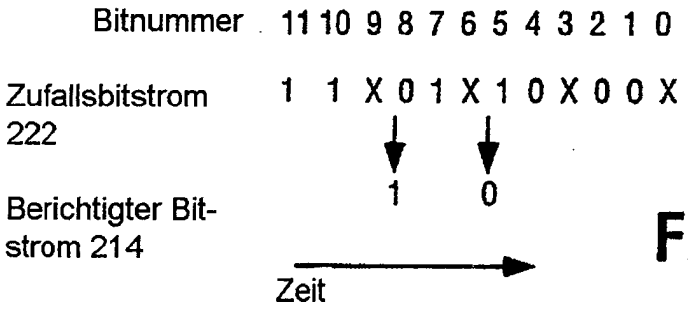

FIG. 14

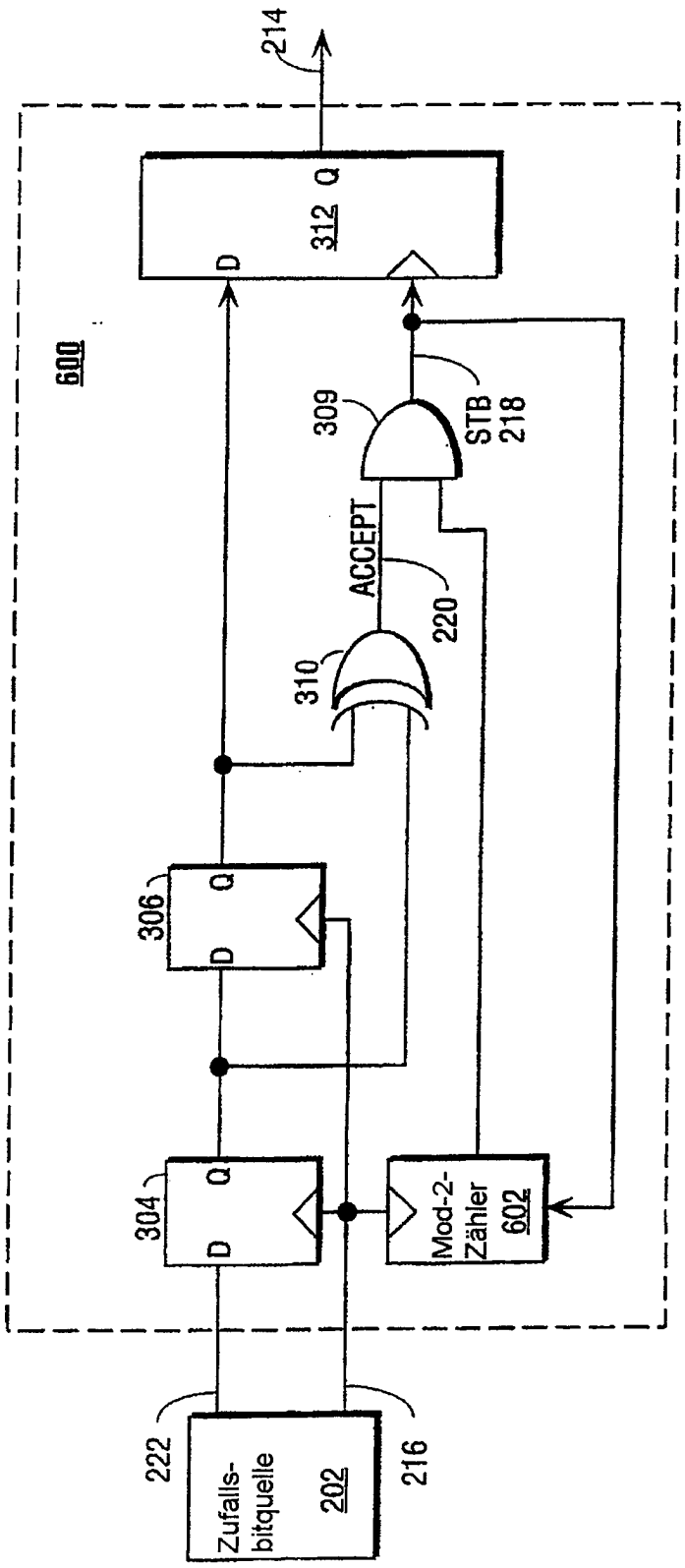
214
312
Q
D
600
309
STB
218
ACCEPT
220
310
306
Q
D
304
Q
D
Mod-2-
Zähler
602
222
Zufalls-
bitquelle
202
216

FIG. 6

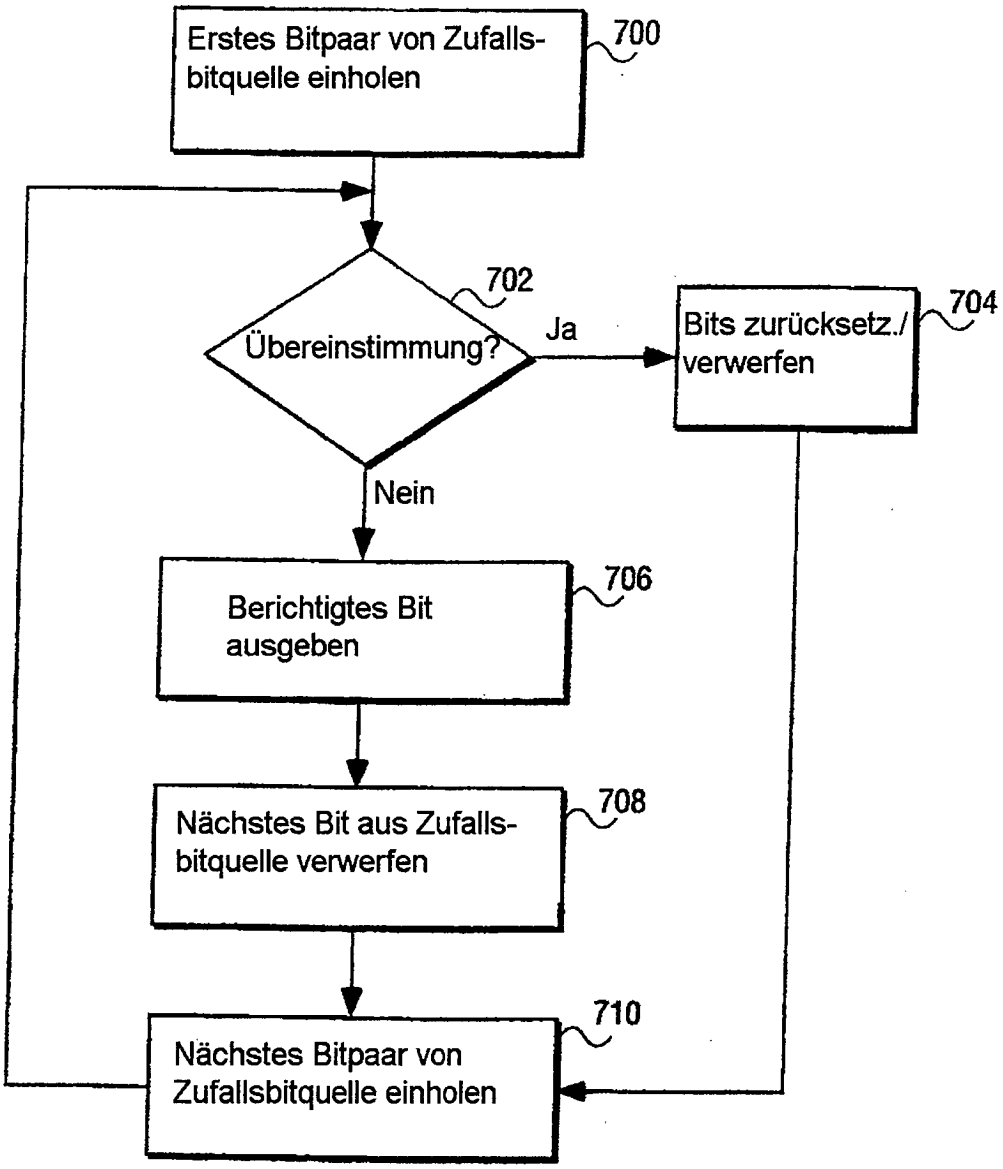
Erstes Bitpaar von Zufalls-bitquelle einholen
700
Übereinstimmung?
702
Ja
Bits zurücksetz./ verwerfen
704
Nein
Berichtigtes Bit ausgeben
706
Nächstes Bit aus Zufalls-bitquelle verwerfen
708
Nächstes Bitpaar von Zufallsbitquelle einholen
710

FIG. 7

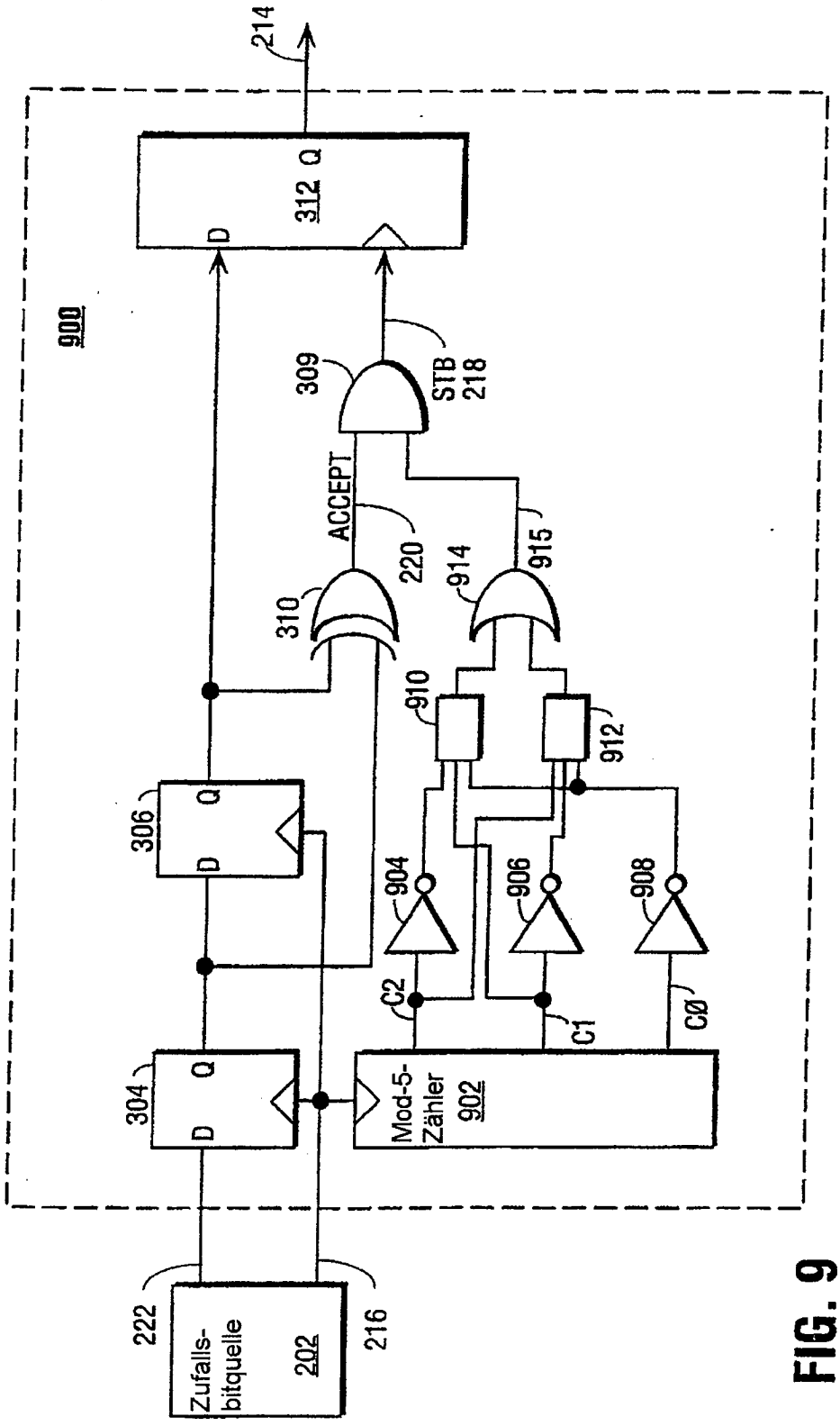
214
900
312
Q
D
309
STB
218
ACCEPT
220
310
914
915
910
912
306
Q
D
904
906
908
C2
C1
C0
304
Q
D
Mod-5-
Zähler
902
222
Zufalls-
bitquelle
202
216

FIG. 9

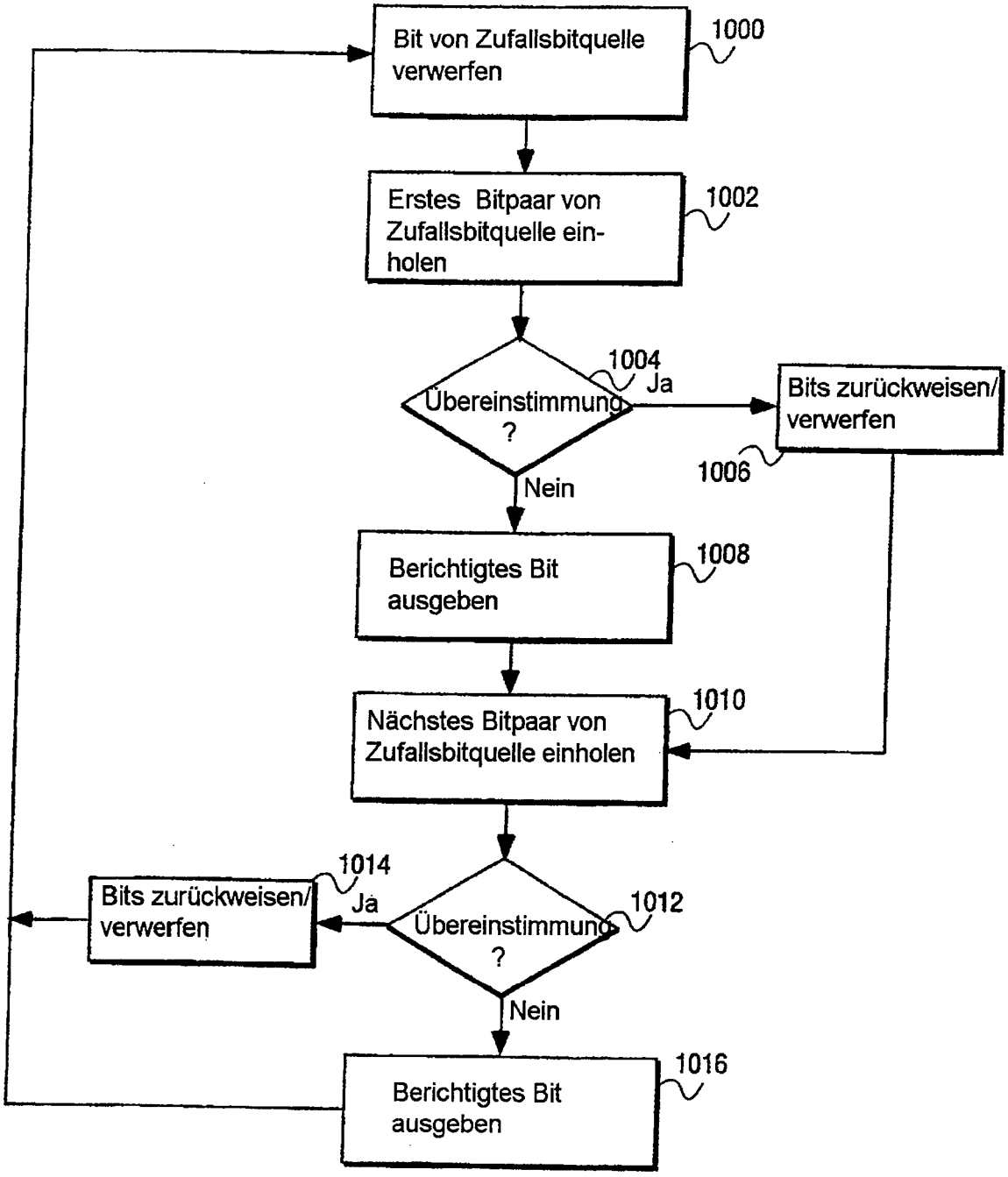
Bit von Zufallsbitquelle verwerfen
1000
Erstes Bitpaar von Zufallsbitquelle einholen
1002
Übereinstimmung ?
1004
Ja
Bits zurückweisen/ verwerfen
1006
Nein
Berichtigtes Bit ausgeben
1008
Nächstes Bitpaar von Zufallsbitquelle einholen
1010
Übereinstimmung ?
1012
Ja
Bits zurückweisen/ verwerfen
1014
Nein
Berichtigtes Bit ausgeben
1016

FIG. 10

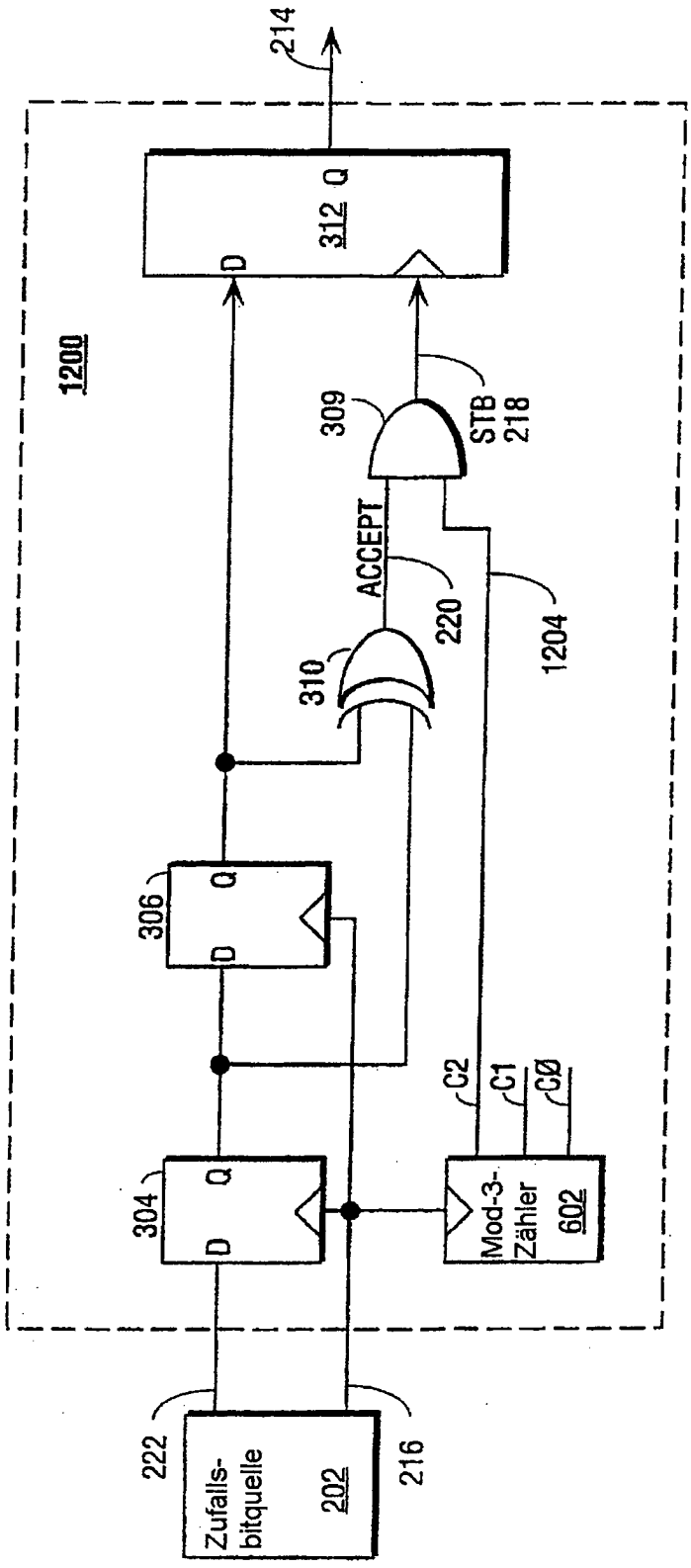
214
1200
312
Q
D
309
STB
218
ACCEPT
220
310
1204
306
Q
D
304
Q
D
C2
C1
C0
Mod-3-
Zähler
602
222
Zufalls-
bitquelle
202
216

FIG. 12

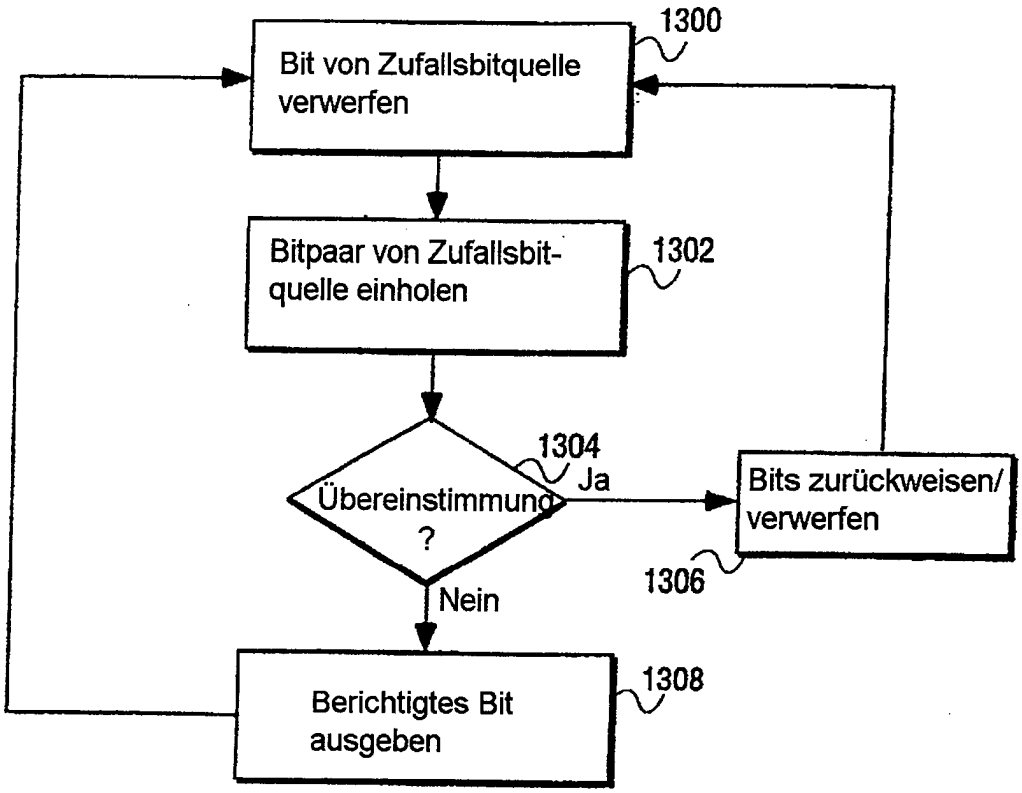
1300
Bit von Zufallsbitquelle verwerfen
Bitpaar von Zufallsbit-quelle einholen
1302
1304
Übereinstimmung ?
Ja
Bits zurückweisen/ verwerfen
1306
Nein
Berichtigtes Bit ausgeben
1308

FIG. 13

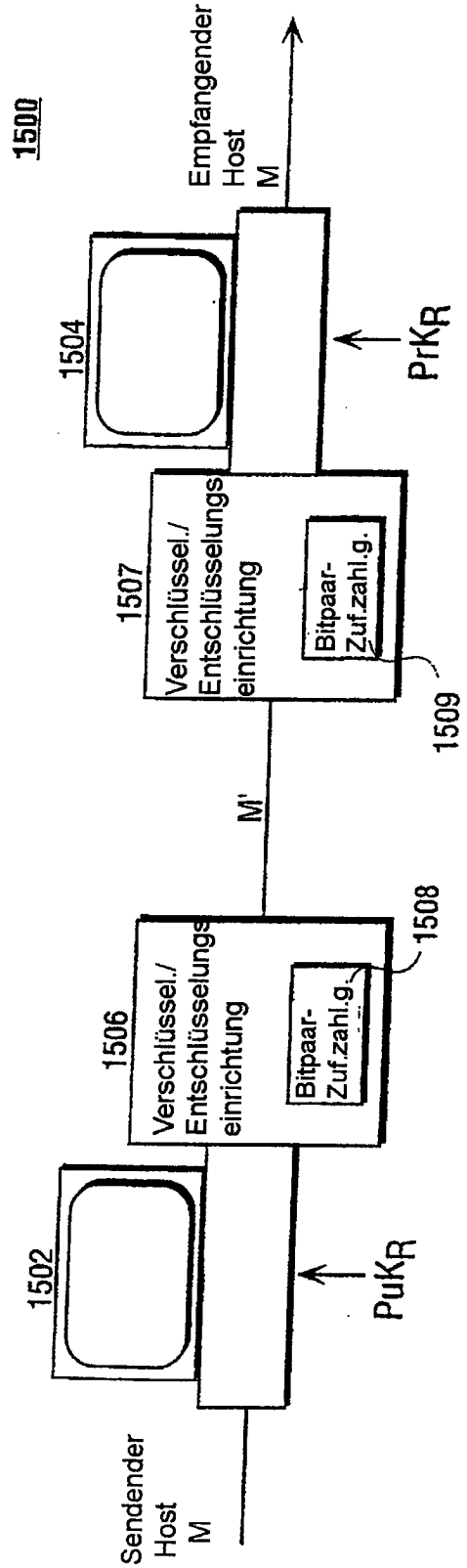
1500
Empfangender Host M
1504
PrK_R
1507
Verschlüssel./ Entschlüsselungs einrichtung
Bitpaar-Zuf.zahl.g.
1509
M'
1508
1506
Verschlüssel./ Entschlüsselungs einrichtung
Bitpaar-Zuf.zahl.g.
1502
PuK_R
Sendender Host M

FIG. 15