



(19)  
Bundesrepublik Deutschland  
Deutsches Patent- und Markenamt

(10) **DE 603 14 530 T2** 2008.02.14

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 546 947 B1**

(51) Int Cl.<sup>8</sup>: **G06F 17/50** (2006.01)

(21) Deutsches Aktenzeichen: **603 14 530.2**

(86) PCT-Aktenzeichen: **PCT/US03/24601**

(96) Europäisches Aktenzeichen: **03 784 944.5**

(87) PCT-Veröffentlichungs-Nr.: **WO 2004/015596**

(86) PCT-Anmeldetag: **05.08.2003**

(87) Veröffentlichungstag  
der PCT-Anmeldung: **19.02.2004**

(97) Erstveröffentlichung durch das EPA: **29.06.2005**

(97) Veröffentlichungstag  
der Patenterteilung beim EPA: **20.06.2007**

(47) Veröffentlichungstag im Patentblatt: **14.02.2008**

(30) Unionspriorität:  
**215869 09.08.2002 US**

(84) Benannte Vertragsstaaten:  
**AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB,  
GR, HU, IE, IT, LI, LU, MC, NL, PT, RO, SE, SI, SK,  
TR**

(73) Patentinhaber:  
**Synplicity, Sunnyvale, US**

(72) Erfinder:  
**NG, Chun Kit, Portland, OR 97229, US; MCELVAIN,  
Kenneth S., Los Altos, CA 94024, US**

(74) Vertreter:  
**Samson & Partner, Patentanwälte, 80538 München**

(54) Bezeichnung: **VERFAHREN UND SYSTEM ZUM DEBUGGEN UNTER VERWENDUNG DUPLIZIERTER LOGIK**

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****KURZFASSUNG****HINTERGRUND DER ERFINDUNG****1. Gebiet der Erfindung**

**[0001]** Die Erfindung bezieht sich auf das Gebiet des Debuggen integrierter Schaltungen, und insbesondere auf das Debuggen integrierter Schaltungen unter Verwendung replizierter Logik.

**2. Hintergrund der Erfindung und Beschreibung des verwandten Stands der Technik**

**[0002]** Für den Entwurf digitaler Schaltungen verwenden Designer computergestützte Techniken. Standardsprachen, wie zum Beispiel Hardwarebeschreibungssprachen (HDLs), wurden entwickelt, um digitale Schaltungen zu beschreiben, um den Entwurf und die Simulation komplexer digitaler Schaltungen zu unterstützen. Da die Gerätetechnologie weiterhin fortschreitet, wurden verschiedene Produktentwurfswerkzeuge entwickelt, um HDLs zur Verwendung mit neueren Geräten und Entwurfsarten anzupassen.

**[0003]** Nachdem der HDL-Code geschrieben und kompiliert wurde, muss der Entwurf einer integrierten Schaltung (IC) oder eines Systems, das mehrere ICs umfasst, auf Korrektheit hin verifiziert werden. Kontinuierlich fortschreitende Verarbeitungstechnologien und die dazugehörige Explosion in der Entwurfsgröße und Komplexität haben zu Verifikationsproblemen für komplexe Schaltungsentwürfe geführt, wie zum Beispiel anwendungsspezifische integrierte Schaltungen (ASICs), die schwierig zu lösen sind, wenn man traditionelle Simulationswerkzeuge und -techniken verwendet.

**[0004]** Demzufolge erstellen einige Designer Prototypplatinen unter Verwendung mehrerer ICs, wie zum Beispiel field-programmable Gate-Arrays (FPGAs), um ihre ASIC-Entwürfe zu verifizieren. Jedoch gibt es immer noch Probleme beim Debuggen des Hardwareentwurfs. Wenn ein Fehler während des Debuggens detektiert wird, können Designer versuchen, Signale von Interesse aus der Schaltung abzugreifen und einen Logikanalysator zu verwenden, um die Ursache des Fehlers zu bestimmen. Dies ist jedoch ein schwieriger Prozess und oft nicht effektiv, insbesondere im Fall von intermittierenden Fehlern. Fehler, die bereits aufgetreten sind, sind oft schwierig zu wiederholen und zu rekonstruieren.

**[0005]** US 6,269,467 B1 beschreibt ein Verfahren zum Entwerfen eines Schaltungssystems. EP 1 168 206 A2 beschreibt ein Verfahren zum Analysieren des Leistungsverlustes einer elektronischen Schaltung.

**[0006]** Die Erfindung offenbart Verfahren und Vorrichtungen zum Debuggen unter Verwendung replizierter Logik. Gemäß einem Aspekt der Erfindung wird eine Textdarstellung einer Schaltung kompiliert, um eine erste Registertransferebene (RTL)-Netzliste zu erzeugen. Die erste RTL-Netzliste kann auf eine Zielarchitektur abgebildet werden, wie zum Beispiel ein field-programmable Gate-Array (FPGA), um eine Prototypplatine zum Debuggen der Schaltung zu erzeugen. Nach dem Debuggen kann ein Designer einen Problembereich in der Schaltung finden. Der Designer kann diesen Teil der Schaltung auswählen, um ihn zur weiteren Analyse zu replizieren. Der ausgewählte Teil der Schaltung wird repliziert. Eine Verzögerungslogik wird eingefügt, um Eingänge in den replizierten Teil der Schaltung zu verzögern. Eine Taktsteuerungslogik kann eingefügt werden, um zu erlauben, dass der Takt an den replizierten Teil pausiert wird. Die Textdarstellung der Schaltung wird rekompiliert, um eine zweite RTL-Netzliste zu erzeugen. Die zweite RTL-Netzliste kann auf eine Zielarchitektur abgebildet werden, wie zum Beispiel einen FPGA oder eine anwendungsspezifische, integrierte Schaltung (ASIC).

**[0007]** Gemäß einem Aspekt der Erfindung werden Vorrichtungen vorgesehen, um die obigen und weitere Verfahren auszuführen.

**KURZE BESCHREIBUNG DER ZEICHNUNGEN**

**[0008]** Die Erfindung wird beispielhaft und nicht einschränkend in den Figuren der beigefügten Zeichnungen veranschaulicht, in denen gleiche Bezugszeichen sich auf gleichartige Elemente beziehen.

**[0009]** [Fig. 1](#) veranschaulicht ein Blockdiagramm eines Computersystems, das verwendet werden kann, um die Ausführungsformen der Erfindung zu implementieren.

**[0010]** [Fig. 2](#) ist ein Ablaufdiagramm, welches eine Ausführungsform eines Verfahrens der Erfindung veranschaulicht.

**[0011]** [Fig. 3](#) veranschaulicht ein Beispiel eines Schaltungsabschnitts, der eine Ausführungsform der Erfindung implementiert.

**[0012]** [Fig. 4](#) veranschaulicht ein Beispiel einer Taktsteuerungslogik gemäß einer Ausführungsform der Erfindung.

**DETAILLIERTE BESCHREIBUNG DER ERFINDUNG**

**[0013]** Ausführungsformen eines Verfahrens und Vorrichtung zum Debuggen unter Verwendung replizierter Logik.

zierter Logik werden beschrieben. In der folgenden Beschreibung werden zahlreiche spezielle Details für ein gründliches Verständnis von Ausführungsformen der Erfindung bereitgestellt. Ein im einschlägigen Fachgebiet kundiger Fachmann wird jedoch erkennen, dass die Erfindung ohne ein oder mehrere der speziellen Details oder mit anderen Verfahren, Komponenten, Materialien, etc. ausgeführt werden kann. In weiteren Fällen werden wohlbekannte Strukturen, Materialien oder Operationen nicht detailliert gezeigt oder beschrieben, um ein Verschleiern von Aspekten der Erfindung zu vermeiden.

**[0014]** Der Bezug in dieser Spezifikation auf "eine Ausführungsform" bedeutet, dass ein spezielles Merkmal, Struktur oder Kennzeichen, das in Verbindung mit der Ausführungsform beschrieben ist, in wenigstens einer Ausführungsform der Erfindung umfasst ist. Daher beziehen sich die Vorkommnisse der Phrasen "in einer Ausführungsform" an verschiedenen Stellen in dieser Spezifikation nicht notwendigerweise alle auf dieselbe Ausführungsform. Außerdem können die speziellen Merkmale, Strukturen oder Kennzeichen auf beliebige, geeignete Weise in einer oder mehreren Ausführungsformen kombiniert werden.

**[0015]** [Fig. 1](#) veranschaulicht ein Blockdiagramm eines Computersystems **100**, das verwendet werden kann, um eine Ausführungsform der Erfindung zu implementieren. Das Computersystem **100** umfasst einen Prozessor **102**, der über einen Bus **110** mit einem Direktzugriffsspeicher (RAM) **104**, einem Festwertspeicher (ROM) **106** und einem Massenspeichergerät **108** gekoppelt ist. Das Massenspeichergerät **108** stellt ein persistentes Datenspeichergerät, wie zum Beispiel ein Diskettenlaufwerk, Festplattenlaufwerk (z. B. magnetisch, optisch, magneto-optisch oder dergleichen), oder ein Streaming-Bandlaufwerk dar. Der Prozessor **102** kann in einem Allzweckprozessor, einem Spezialzweckprozessor oder einem speziell programmierten Logikgerät verkörpert sein.

**[0016]** Ein Anzeigegerät **112** ist an den Prozessor **102** durch den Bus **110** gekoppelt und stellt eine grafische Ausgabe für das Computersystem **100** bereit. Eine Tastatur **114** und eine Cursorsteuerungseinheit **116** sind an den Bus **110** zum Kommunizieren von Information und Befehlsauswahlen an Prozessor **102** gekoppelt. Auch an den Prozessor **102** durch einen Bus **110** gekoppelt ist eine Eingabe/Ausgabe (I/O)-Schnittstelle **117**, die verwendet werden kann, um Daten zu steuern und an elektronische Geräte (Drucker, andere Komponenten, etc.) zu übertragen, die mit dem Computersystem **100** verbunden sind.

**[0017]** Es sei darauf hingewiesen, dass die Architektur von [Fig. 1](#) nur zu Zwecken der Veranschaulichung bereitgestellt wird, und dass ein Computer, der in Verbindung mit Ausführungsformen der Erfindung

verwendet wird, nicht auf diese spezielle Architektur beschränkt ist.

**[0018]** Wie von Fachleuten geschätzt wird, kann der Inhalt zum Implementieren einer Ausführungsform eines Verfahrens der Erfindung, zum Beispiel Computerprogrammbefehle, durch ein beliebiges maschinenlesbares Medium bereitgestellt werden, das Daten speichern kann, die durch das System **100** zugänglich sind, als Teil von oder zusätzlich zum Speicher, der umfasst aber nicht beschränkt ist auf Karten, Magnetkassetten, Flash-Speicherkarten, digitale Videoscheiben, Direktzugriffsspeicher (RAM), Festwertspeicher (ROMs) und dergleichen. In dieser Hinsicht ist das System **100** ausgestattet, um mit einem solchen maschinenlesbaren Medium auf eine im Stand der Technik wohlbekannte Weise zu kommunizieren.

**[0019]** Es wird weiterhin von Fachleuten geschätzt werden, dass der Inhalt zum Implementieren einer Ausführungsform des Verfahrens der Erfindung dem System **100** aus einem beliebigen Gerät bereitgestellt werden kann, das den Inhalt speichern kann und den Inhalt an das System **100** kommunizieren kann. Zum Beispiel kann in einer Ausführungsform das System **100** mit einem Netz verbunden sein, und der Inhalt kann auf einem beliebigen Gerät im Netz gespeichert sein.

**[0020]** [Fig. 2](#) ist ein Ablaufdiagramm, welches eine Ausführungsform eines Verfahrens der Erfindung veranschaulicht. Bei **200** wird eine Textdarstellung einer Schaltung kompiliert, um eine erste Registertransfer (RTL)-Netzliste zu erzeugen. Eine Schaltung wird typischerweise durch eine Textdarstellung beschrieben, indem Hardware Beschreibungssprache (HDL)-Quellcode Beschreibungen der Elemente der Schaltung geschrieben werden. Das Schreiben von HDL-Quellcode ist in der Literatur gut beschrieben. Siehe, zum Beispiel, Verilog HDL: a Guide to Digital Design and Synthesis, Samir Palnithar, SunSoft Press, 1996; siehe auch, A VHDL Synthesis Primer, J. Bhasher, Star Galaxy Publishing, 1996.

**[0021]** Die Textdarstellung wird dann in einen Compiler eingegeben. Ein Beispiel eines Compilers ist ein Logiksynthese-Compiler, der typischerweise ein Computerprogramm ist, das auf einem Altzweck-Computersystem arbeitet, obwohl in einigen Ausführungsformen das Computersystem ein dediziertes, Spezialzweck-Computersystem sein kann. Ein Beispiel eines Logiksynthese-Compilers ist das Programm "Synplify<sup>TM</sup>" von Synplify, Inc. aus Sunnyvale, Kalifornien. Nach einer Kompilierung wird eine RTL-Netzliste erzeugt. Die RTL-Netzliste zeigt gewöhnlich Register und andere verbundene Logik, um den Datenfluss durch eine Schaltung zu zeigen, die in der Textdarstellung beschrieben wurde.

**[0022]** In einer Ausführungsform der Erfindung wird die RTL-Netzliste auf eine Zielarchitektur abgebildet. Die Zielarchitektur wird typischerweise durch einen Lieferanten der integrierten Schaltung (IC) bestimmt. Beispiele von Zielarchitekturen umfassen field-programmable Gate-Arrays (FPGAs) und komplexe programmierbare Logikgerät von Anbietern wie zum Beispiel Alters, Lucent Technologies, Advanced Micro Devices (AMD) und Lattice Semiconductor. Die Abbildungsoperation konvertiert die RTL-Ebenenbeschreibung der gewünschten Schaltung in die äquivalente Schaltung, die unter Verwendung von Baublöcken der Zielarchitektur implementiert wird. Eine technologiespezifische Netzliste wird erzeugt. Herkömmliche Platzierungs- und Leitungsführungsoperationen-Softwarewerkzeuge können dann verwendet werden, um einen Schaltungsentwurf in der Zielarchitektur zu schaffen.

**[0023]** Zu Debugzwecken können IC-Designer Prototypplatinen unter Verwendung mehrerer ICs, wie zum Beispiel FPGAs, erstellen, um ihre Entwürfe zu verifizieren. Zum Beispiel kann nach der Kompilierung, dem Abbilden, und Platzierungs- und Leitungsführungsoperationen die Schaltung in FPGAs programmiert werden, um einen Prototyp des Entwurfs zu schaffen. Die FPGAs können dann getestet werden, um Problembereiche im Entwurf zu bestimmen.

**[0024]** Wenn ein Problembereich im Entwurf gefunden wird, kann der Designer den Teil der Schaltung auswählen, um ihn zu replizieren und das Problem weiter zu analysieren. Bei **202** wird ein Teil der Schaltung ausgewählt. Diese Replikation kann eine Replikation der Logikelemente, der Eingangssignale und der Ausgangssignale des ausgewählten Teils der Schaltung umfassen.

**[0025]** In einer Ausführungsform der Erfindung werden auch Taktsignale repliziert. Taktsteuerungslogik wird eingefügt, um die Taktsignale zu steuern. Die Taktsteuerungslogik ermöglicht, dass der Takt an den replizierten Logikblock pausiert wird, um die replizierte Logik von der Ausführung abzuhalten, wenn bestimmte Zustände vorliegen und ein Einzelschrittdurchgehen durch die replizierte Logik zu erlauben, um einen Fehler zu analysieren. Der Designer kann dann einen Stopppunkt auswählen, um den Takt an den replizierten Teil der Schaltung zu pausieren, wenn bestimmte Zustände vorliegen. Zum Beispiel kann der Designer Werte für die Ausgänge und Eingänge auswählen, die den Takt pausieren. Dies ermöglicht dem Designer, die ausgewählte Logik sorgfältiger zu analysieren, wenn bestimmte Problemzustände vorliegen.

**[0026]** Bei **206** wird eine Verzögerungslogik eingefügt, um Eingänge in den replizierten Teil der Schaltung zu verzögern. Die Länge der Verzögerung kann vom Schaltungsdesigner ausgewählt werden. Die

Verzögerungslogik erlaubt es einen im ausgewählten Teil der Schaltung beobachteten Fehler zu analysieren, nachdem ein Auftreten des Fehlers beobachtet wird, da der Fehler im replizierten Teil der Schaltung zu einem späteren Zeitpunkt wieder auftreten wird.

**[0027]** Bei **208** wird die Textdarstellung der Schaltung rekompiliert, um eine zweite RTL-Netzliste zu erzeugen. Dann können die Abbildungs- und Platzierungs- und Leitungsführungsoperation unter Verwendung der zweiten RTL-Netzliste durchgeführt werden, um die Schaltung in einer Zielarchitektur, wie zum Beispiel einen FPGA, zu implementieren. In einer Ausführungsform der Erfindung wird eine Syntheseoperation durchgeführt, um eine anwendungsspezifische integrierte Schaltung (ASIC) aus der zweiten RTL-Netzliste zu generieren. Eine Schaltung mit replizierter Logik wird hergestellt, die einem Schaltungsdesigner erlaubt, einen Problembereich im Design zu analysieren.

**[0028]** [Fig. 3](#) veranschaulicht ein Beispiel eines Abschnitts einer Schaltung **300**, die eine Ausführungsform der Erfindung implementiert. Ein Logikblock **302** ist ein Teil der Schaltung im ursprünglichen IC-Entwurf. Das Debuggen des ursprünglichen IC-Entwurfs enthüllte ein Problem mit dem Logikblock **302**. Deshalb wurde der ursprüngliche Logikblock **302** ausgewählt und repliziert, um eine weitere Analyse des Problems zu ermöglichen. Der ursprüngliche Logikblock **302** wird repliziert, um einen replizierten Logikblock **304** herzustellen. Ausgänge **308** aus dem ursprünglichen Logikblock **302** werden repliziert, um replizierte Ausgänge **310** herzustellen. Eingänge **306** können auch repliziert werden.

**[0029]** Eine Verzögerungslogik **312** wird eingefügt, um die Eingänge **306** in den replizierten Logikblock **304** zu verzögern. Die Verzögerungslogik umfasst typische Schaltungslogik und -elemente, wie zum Beispiel Inverter, die veranlassen, dass die Eingänge **306** am replizierten Logikblock **304** zeitlich später ankommen als die Eingänge am ursprünglichen Logikblock **302** ankommen. Auf diese Weise kann ein Fehler analysiert werden, nachdem sein Auftreten im ursprünglichen Logikblock wahrgenommen wird, da der Fehler im replizierten Logikblock zu einem späteren Zeitpunkt erscheinen wird.

**[0030]** Eine Taktsteuerungslogik **314** wird eingefügt, um die Taktsignale **322** an den replizierten Logikblock **304** zu steuern. Die Taktsteuerungslogik **314** enthält typische Logik- und Schaltungselemente, die erlauben, dass der Takt **322** an den replizierten Logikblock **304** pausiert wird, um die replizierte Logik von der Ausführung abzuhalten, wenn bestimmte Bedingungen vorliegen. Die Taktsteuerungslogik **314** kann auch ein Einzelschrittdurchgehen durch die replizierte Logik auf einer Takt-per-Takt-Grundlage, um einen Fehler zu analysieren. Ein Haltepunkt **318** kann ge-

setzt werden, um den Takt zu pausieren, wenn bestimmte Zustände vorliegen.

**[0031]** [Fig. 4](#) veranschaulicht ein Beispiel der Taktsteuerungslogik **314** gemäß einer Ausführungsform der Erfindung. Während des Normalbetriebs taktet der Systemtakt **316** die Schaltungsflüsse durch das Latch **400** und agiert wie der Takt **322** an den replizierten Logikblock **304**. Der Haltepunkt **318** schaltet den Takt **322** auf eine gelatchte Version des Systemtakts **316** um, der dann durch Taktsteuerungssignale **320** gesteuert werden kann, um zu erlauben, dass der Takt **322** pausiert und im Einzelschritt auf einer Zyklus-per-Zyklus-Grundlage durchlaufen wird.

**[0032]** Es wurden also Ausführungsformen eines Verfahrens und einer Vorrichtung zum Debuggen unter Verwendung replizierter Logik beschrieben. Die obige Beschreibung veranschaulicht Ausführungsformen der Erfindung, einschließlich dessen, was in der Zusammenfassung beschrieben ist, soll nicht erschöpfend sein oder die Erfindung auf die präzisen, offenbaren Formen beschränken. Während spezielle Ausführungsformen von der, und Beispiele für die, Erfindung hierin zu Zwecken der Veranschaulichung beschrieben werden, sind verschiedene äquivalente Modifikationen innerhalb des Schutzbereichs der Erfindung möglich, wie Fachleute aus dem einschlägigen Fachgebiet erkennen werden. Diese Modifikationen können an der Erfindung im Lichte der obigen detaillierten Beschreibung durchgeführt werden. Die in den folgenden Ansprüchen verwendeten Begriffe sollten nicht dahingehend ausgelegt werden, dass sie die Erfindung auf die speziellen in der Spezifikation und den Ansprüchen offenbaren Ausführungsformen beschränken. Stattdessen ist der Schutzbereich der Erfindung gänzlich durch die folgenden Ansprüche zu bestimmen.

#### Legende zu den Figuren

#### Bezugszeichenliste

#### [Fig. 1](#)

<b>100</b>	Computersystem
<b>102</b>	Prozessor
<b>104</b>	Direktzugriffsspeicher
<b>106</b>	Festwertspeicher
<b>108</b>	Massenspeichergerät
<b>110</b>	Bus
<b>112</b>	Anzeige
<b>114</b>	Tastatur
<b>116</b>	Cursorsteuerung
<b>118</b>	Ein/Ausgabe-Schnittstelle

#### [Fig. 2](#)

<b>200</b>	Kompilieren eine Textdarstellung einer Schaltung
<b>202</b>	Auswählen eines Teils der Textdarstellung der Schaltung
<b>204</b>	Replizieren des ausgewählten Teils der Schaltung
<b>206</b>	Einfügen von Verzögerungslogik, um Eingänge an den replizierten Teil der Schaltung zu verzögern
<b>208</b>	Rekompilieren der Textdarstellung der Schaltung

#### [Fig. 3](#)

<b>300</b>	Schaltungsteil
<b>302</b>	ursprünglicher Logikblock
<b>304</b>	replizierter Logikblock
<b>306</b>	Eingänge
<b>308</b>	Ausgänge
<b>310</b>	replizierte Ausgänge
<b>312</b>	Verzögerungslogik
<b>314</b>	Taktsteuerungslogik
<b>316</b>	Systemtakt
<b>318</b>	Haltepunkt
<b>320</b>	Taktsteuerungen
<b>322</b>	gesteuerter Takt

#### [Fig. 4](#)

<b>314</b>	Taktsteuerungslogik
<b>316</b>	Systemtakt
<b>318</b>	Haltepunkt
<b>320</b>	Taktsteuerungen
<b>322</b>	gesteuerter Takt
<b>400</b>	Latch

#### Patentansprüche

1. Debugverfahren, umfassend:  
 Kompilieren (**200**) einer Textrepräsentation einer Schaltung, um eine erste Registertransferebene-Netzliste zu erzeugen;  
 Auswählen (**202**) eines Teils (**302**) der Textrepräsentation der Schaltung;  
 Replizieren (**204**) des ausgewählten Teils (**302**) der Textrepräsentation der Schaltung;  
 Einfügen (**206**) von Verzögerungslogik (**312**), um Eingänge zum replizierten Teil der Textrepräsentation der Schaltung (**304**) zu verzögern; und  
 Rekompilieren (**208**) der Textrepräsentation der Schaltung, um eine zweite Registertransferebene-Netzliste zu erzeugen.

2. Verfahren gemäß Anspruch 1, wobei die Textrepräsentation in einer Hardwarebeschreibungssprache geschrieben ist.

3. Verfahren gemäß Anspruch 1, wobei das Rep-



lizieren des ausgewählten Teils der Schaltung ein Replizieren von Taktsignalen an den ausgewählten Teil der Schaltung **(304)** umfasst.

4. Verfahren gemäß Anspruch 3, welches außerdem ein Einfügen einer Logik **(314)** umfasst, um die Taktsignale **(316)** an den replizierten Teil der Schaltung **(304)** zu steuern, um zu ermöglichen, dass eine Ausführung des replizierten Teils der Schaltung **(304)** pausiert wird.

5. Verfahren gemäß Anspruch 1, welches außerdem ein Abbilden der ersten Registertransferebene-Netzliste auf eine ausgewählte Technologiearchitektur umfasst.

6. Verfahren gemäß Anspruch 5, welches außerdem ein Durchführen einer Platzierungs- und Leitungsführungsoperation umfasst, um die Schaltung in der ausgewählten Technologiearchitektur zu implementieren.

7. Verfahren gemäß Anspruch 6, welches außerdem ein Programmieren der ersten Registertransferebene-Netzliste in eine programmierbare Hardwarevorrichtung umfasst.

8. Verfahren gemäß Anspruch 7, welches außerdem ein Testen eines Prototyps der Schaltung mit der programmierbaren Hardwarevorrichtung umfasst.

9. Verfahren gemäß Anspruch 7, wobei die programmierbare Hardwarevorrichtung ein field-programmable Gate-Array ist.

10. Verfahren gemäß Anspruch 1, welches außerdem ein Programmieren der zweiten Registertransferebene-Netzliste in eine programmierbare Hardwarevorrichtung umfasst.

11. Verfahren gemäß Anspruch 1, welches außerdem ein Erzeugen einer anwendungsspezifischen integrierten Schaltung aus der zweiten Registertransferebene-Netzliste umfasst.

12. Verfahren gemäß Anspruch 1, welches außerdem ein Erzeugen einer integrierten Schaltung **(300)** aus der zweiten Transferebene-Netzliste umfasst, wobei die integrierte Schaltung **(330)** folgendes umfasst:  
mehrere Logikelemente **(302)**;  
eine Replikation **(304)** eines oder mehrerer der mehreren Logikelemente;  
Verzögerungslogik **(312)**, die an den replizierten Teil der Schaltung **(304)** gekoppelt ist, um Eingänge **(306)** in den replizierten Teil der Schaltung **(304)** zu verzögern; und  
Taktsteuerungslogik **(314)**, die an den replizierten Teil der Schaltung **(304)** gekoppelt ist, um zu ermöglichen, dass die Ausführung des replizierten Teils der

Schaltung **(304)** pausiert wird.

13. Verfahren gemäß Anspruch 12, wobei die Taktsteuerungslogik **(314)** einen Stopppunkt **(318)** umfasst, um den replizierten Teil der Schaltung **(304)** zu pausieren.

14. Verfahren gemäß Anspruch 12, wobei die Taktsteuerungslogik **(314)** außerdem Logik umfasst, um zu ermöglichen, dass der replizierte Teil der Schaltung **(304)** auf einer Takt-für-Takt-Grundlage ausgeführt wird.

15. Verfahren gemäß Anspruch 12, wobei die Taktsteuerungslogik ein Latch **(400)** umfasst.

16. Maschinenlesbares Medium, das mehrere ausführbare Befehle enthält, die, wenn sie auf einem digitalen Verarbeitungssystem ausgeführt werden, das digitale Verarbeitungssystem veranlassen, das Debugverfahren gemäß Anspruch 1 durchzuführen.

17. Maschinenlesbares Medium gemäß Anspruch 16, wobei das Verfahren außerdem ein Abbilden der ersten Registertransferebene-Netzliste auf eine ausgewählte Technologiearchitektur umfasst.

18. Maschinenlesbares Medium gemäß Anspruch 17, wobei das Verfahren außerdem ein Durchführen einer Platzierungs- und Leitungsführungsoperation umfasst, um die Schaltung in der ausgewählten Technologiearchitektur zu implementieren.

19. Maschinenlesbares Medium gemäß Anspruch 16, wobei das Verfahren außerdem ein Programmieren der ersten Registertransferebene-Netzliste in eine programmierbare Hardwarevorrichtung umfasst.

20. Maschinenlesbares Medium gemäß Anspruch 16, wobei das Verfahren außerdem ein Programmieren der zweiten Registertransferebene-Netzliste in eine programmierbare Hardwarevorrichtung umfasst.

21. Maschinenlesbares Medium gemäß Anspruch 16, wobei das Verfahren außerdem ein Erzeugen einer anwendungsspezifischen integrierten Schaltung aus der zweiten Registertransferebene-Netzliste umfasst.

22. Computersystem **(100)**, umfassend:  
einen Bus **(110)**;  
eine Datenspeichervorrichtung **(108)**, die an den Bus gekoppelt ist; und  
einen Prozessor, der an die Datenspeichervorrichtung gekoppelt ist, wobei der Computer **(100)** dazu programmiert ist, das Debugverfahren gemäß Anspruch 1 auszuführen.

23. Computersystem gemäß Anspruch 22, wobei das Verfahren außerdem ein Abbilden der ersten und zweiten Registertransferebene-Netzliste auf eine ausgewählte Technologiearchitektur umfasst.

24. Computersystem gemäß Anspruch 23, wobei das Verfahren außerdem ein Durchführen eines Platzierungs- und Leitungsführungsoperation umfasst, um die Schaltung in der ausgewählten Technologiearchitektur zu implementieren.

25. Computersystem gemäß Anspruch 22, wobei das Verfahren außerdem ein Programmieren der ersten Registertransferebene-Netzliste in eine programmierbare Hardwarevorrichtung umfasst.

26. Computersystem gemäß Anspruch 22, wobei das Verfahren außerdem ein Programmieren der zweiten Registertransferebene-Netzliste in eine programmierbare Hardwarevorrichtung umfasst.

27. Computersystem gemäß Anspruch 22, wobei das Verfahren außerdem ein Erzeugen einer anwendungsspezifischen integrierten Schaltung aus der zweiten Registertransferebene-Netzliste umfasst.

Es folgen 4 Blatt Zeichnungen

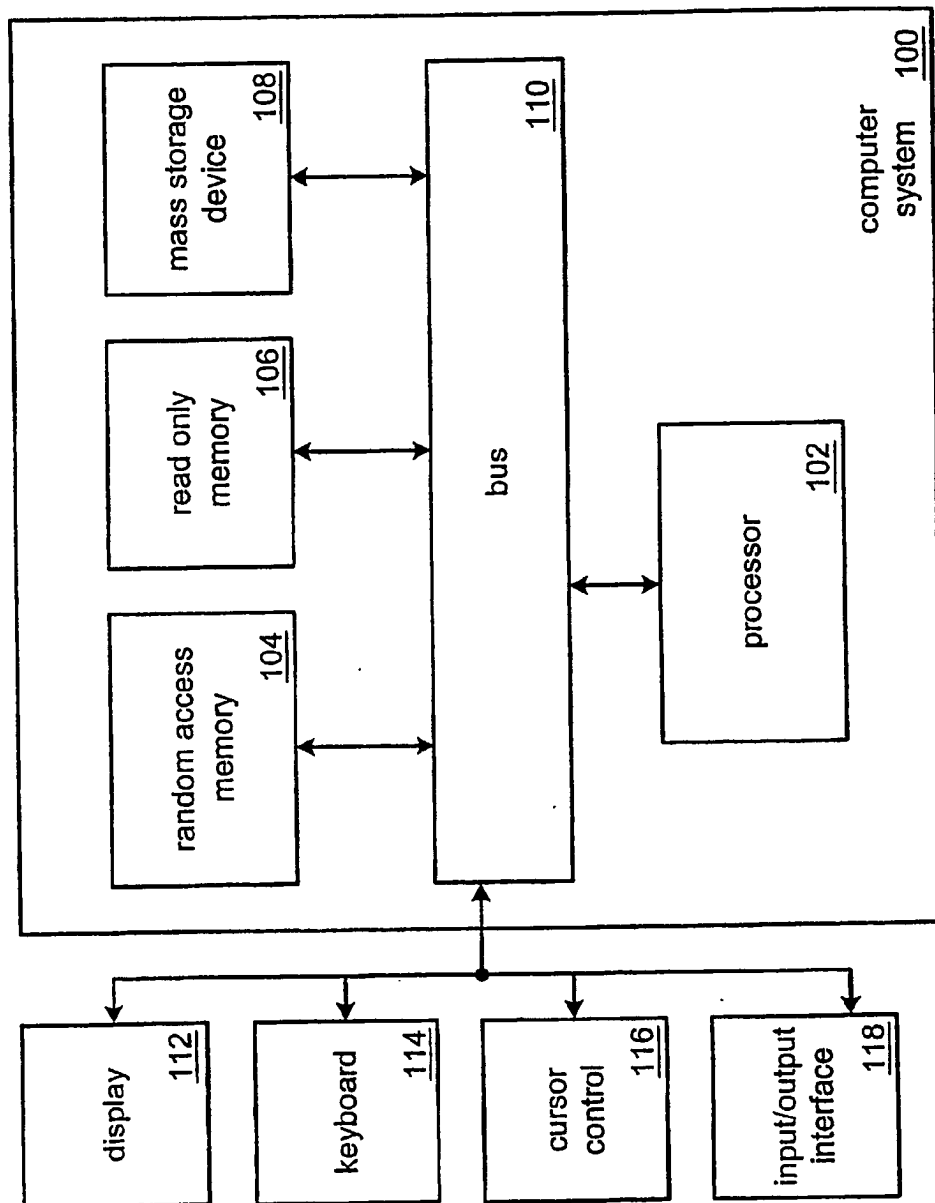
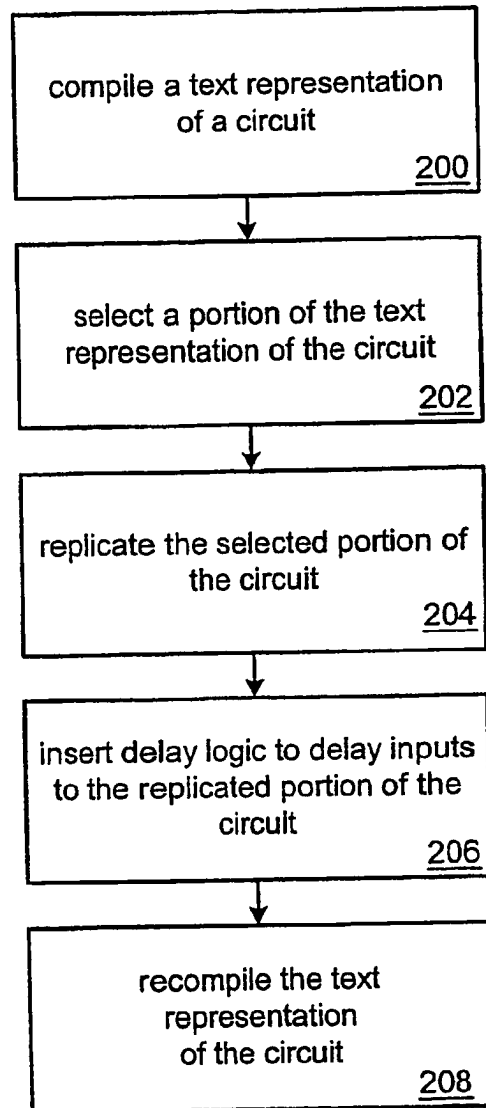


FIG. 1





**FIG. 2**

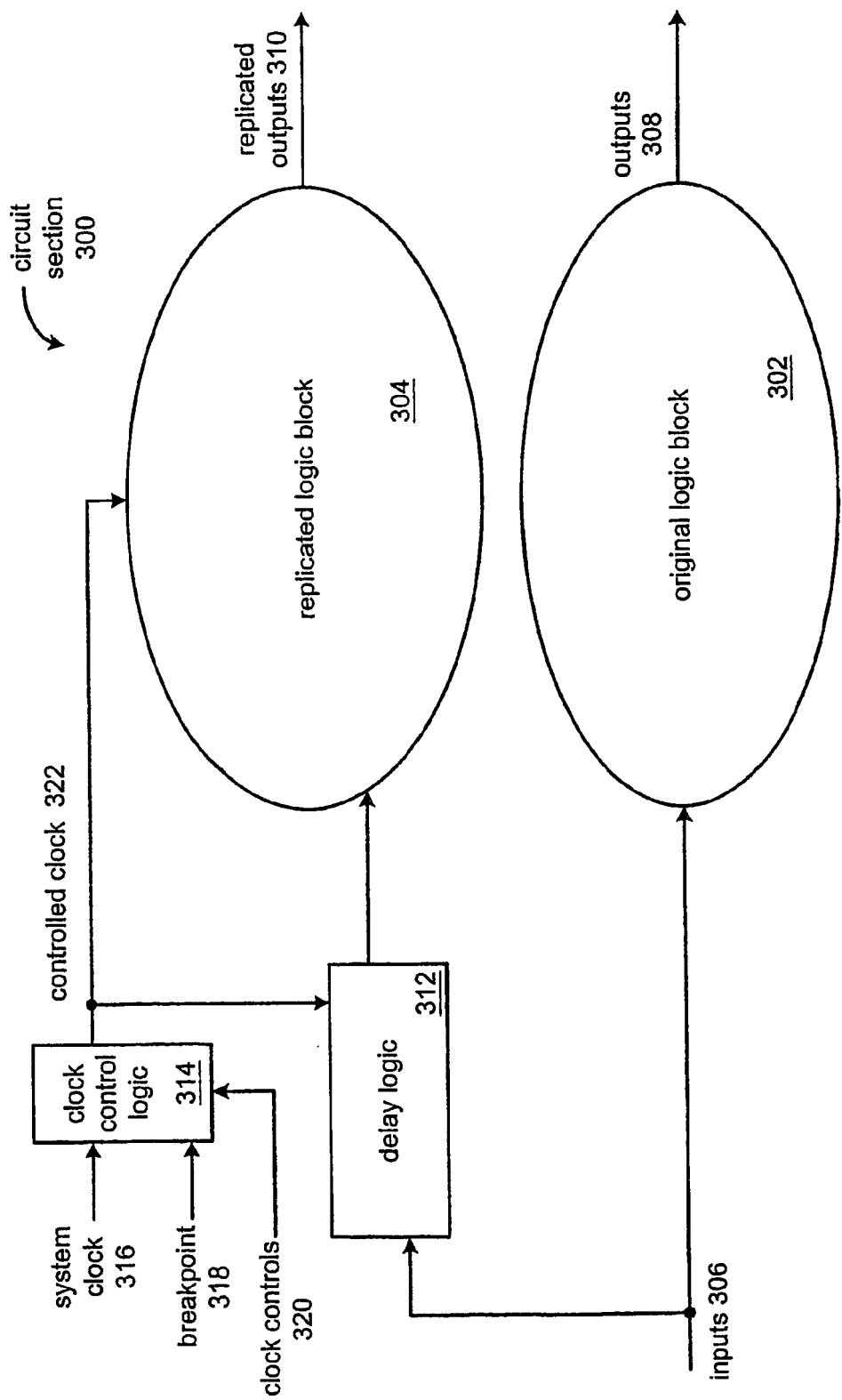


FIG. 3

