

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

(19) Weltorganisation für geistiges Eigentum
Internationales Büro



(43) Internationales Veröffentlichungsdatum
9. Februar 2006 (09.02.2006)

PCT

(10) Internationale Veröffentlichungsnummer
WO 2006/012875 A2

(51) Internationale Patentklassifikation⁷: **G06F 17/50**

(21) Internationales Aktenzeichen: PCT/DE2005/001349

(22) Internationales Anmeldedatum:
28. Juli 2005 (28.07.2005)

(25) Einreichungssprache: Deutsch

(26) Veröffentlichungssprache: Deutsch

(30) Angaben zur Priorität:
10 2004 037 402.3 30. Juli 2004 (30.07.2004) DE

(71) Anmelder (für alle Bestimmungsstaaten mit Ausnahme von
US): **CERTESS S.A.** [—/FR]; 347, rue Aristide Bergès,
Parc d'Activités de Centr'Alp, F-38340 Voreppe (FR).

(72) Erfinder; und

(75) Erfinder/Anmelder (nur für US): **GROSSE, Jörg**
[DE/DE]; Marienallee 1, 01099 Dresden (DE). **HAMP-
TON, Mark** [NZ/FR]; Les Roussets, F-38420 Revel (FR).

(74) Anwälte: **RIEHEL MANN, Jens** usw.; PAe Riechel-
mann & Carlsohn, Wiener Strasse 91, 01219 Dresden (DE).

(81) Bestimmungsstaaten (soweit nicht anders angegeben, für
jede verfügbare nationale Schutzrechtsart): AE, AG, AL,

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

(84) Bestimmungsstaaten (soweit nicht anders angegeben, für
jede verfügbare regionale Schutzrechtsart): ARIPO (BW,
GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG,
ZM, ZW), eurasisches (AM, AZ, BY, KG, KZ, MD, RU,
TJ, TM), europäisches (AT, BE, BG, CH, CY, CZ, DE, DK,
EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC,
NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG,
CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Veröffentlicht:

— ohne internationalen Recherchenbericht und erneut zu ver-
öffentlichen nach Erhalt des Berichts

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

(54) Title: METHOD FOR EVALUATING A TEST PROGRAM QUALITY

(54) Bezeichnung: VERFAHREN ZUR BEWERTUNG DER GÜTE EINES TESTPROGRAMMS

(57) Abstract: The invention relates to a method for evaluating an integrated circuit test program simulated with the aid of a com-
puter and consisting (a) in providing a first file describing an integrated circuit, (b) in simulating a mutant integrated circuit obtainable
by introducing mutations in the integrated circuit described at the first stage, (c) in supplying input values to the mutant integrated
circuit and detecting output values produced by the mutant integrated circuit for said input values, (d) in comparing the output val-
ues produced by the mutant integrated circuit with expected values produced by the test program, wherein said expected values are
generated in a reference system and (e) in evaluating the test program quality by means of comparison results.

(57) Zusammenfassung: Die Erfindung betrifft ein Verfahren zur Bewertung der Güte eines Testprogramms für mittels eines Com-
puters simulierte integrierte Schaltungen, umfassend (a) das Bereitstellen einer ersten Datei, die eine integrierte Schaltung beschreibt;
(b) das Simulieren einer mutierten integrierten Schaltung, die durch Einbau von Mutationen in die in der ersten Datei beschriebene
integrierte Schaltung erhalten wird; (c) das Zuführen von Eingangswerte zu der mutierten integrierten Schaltung und Erfassen der
für diese Eingangswerte von der mutierten integrierten Schaltung erzeugten Ausgangswerte; (d) den Vergleich der von der mutierten
integrierten Schaltung erzeugten Ausgangswerte mit Erwartungswerten, die von dem Testprogramm bereitgestellt werden, wobei
die Erwartungswerte in einem Referenzsystem generiert worden sind; und (e) die Bewertung der Güte des Testprogramms anhand
der Vergleichsergebnisse.

WO 2006/012875 A2

Beschreibung

Verfahren zur Bewertung der Güte eines Testprogramms

- 5 Die Erfindung betrifft ein Verfahren zur Bewertung der Güte eines Testprogramms für mittels eines Computers simulierte integrierte Schaltungen.

Die Größe und Komplexität von integrierten Schaltungen nimmt
10 ständig zu. Damit ist jedoch auch die Zunahme von Fehlern verbunden.

In der Regel werden integrierte Schaltungen mittels spezieller Computersprachen, sogenannten High Level Description Languages (HDLs), beschrieben. Übliche HDLs sind VHDL und Verilog.
15 Der HDL-File wird verwendet, um mittels spezieller Computerprogramme automatisch das Layout der Schaltungsanordnung und die erforderlichen photolithographischen Masken zu erzeugen, mit denen die tatsächliche Schaltungsanordnung hergestellt werden kann.
20

Zur Vermeidung von Fehlern in den integrierten Schaltungen kann es jedoch erforderlich sein, die mittels des HDL-Files beschriebenen Schaltungsanordnungen zu testen. Zu diesem
25 Zwecke werden Programme verwendet, die auf Grundlage des HDL-Files die integrierte Schaltung simulieren. Derartige Programme werden als Simulatoren bezeichnet. Ein solcher Simulator wirkt mit einer als Prüfbank ("Test-Bench") bezeichneten Einheit, dem Testprogramm (d. h. Computercode) zusammen, das
30 die Eingangswerte ("Input") bereitstellt, die die integrierte Schaltung verarbeiten soll, und die Ergebnisse der Verarbeitung (Ausgangswerte) in der simulierten Schaltung mit den er-

warteten Ergebnissen vergleicht. Das Testprogramm beschreibt also (1) die Eingangswerte (ggf. in Form von Signalwellenformen), (2) die Leistungs- und Richtigkeitskriterien, die nach der Verarbeitung der Eingangswerte in der simulierten Schaltung erhalten werden, und (3) die Handlungen, die der Simulator ausführen soll, wenn die Ergebnisse der Simulation den Leistungs- und Richtigkeitskriterien entsprechen oder wenn dies nicht der Fall ist.

Das Testergebnis hängt allerdings wesentlich von der Qualität und der Vollständigkeit des Testprogramms ab. Im Hinblick auf die hohe Zahl elektronischer Komponenten, die in dem HDL-File beschrieben werden, ist es sehr schwierig, Prüfbanken zu schaffen, die der Komplexität der zu testenden simulierten integrierten Schaltung gerecht werden. Fehler in der simulierten Schaltung können nur dann gefunden werden, wenn das Testprogramm tatsächlich Tests vorsieht, die den fehlerhaften Bereich der simulierten integrierten Schaltung betreffen. Anderenfalls kann ein positives Testergebnis unter Umständen lediglich auf eine unzureichende Testung (bzw. Verifizierung) der integrierten Schaltung zurückzuführen sein. Eine Überprüfung der Qualität und der Vollständigkeit des Testprogramms ist daher erforderlich. Die ohnehin geringe Geschwindigkeit der Softwaresimulation einer integrierten Schaltung wird durch Tests, die die Qualität und die Vollständigkeit des Testprogramms überprüfen sollen, allerdings weiter verringert.

DE 199 59 157 A1 beschreibt ein Verfahren zum funktionsmäßigen Simulieren und Überprüfen der Korrektheit eines Computerprogramms, bei dem mutationsbehaftete Computerprogramme erzeugt werden. Das Computerprogramm kann Quellcodemodule aus

Funktionsmodellen einer integrierten Schaltung aufweisen. Das Verfahren simuliert ein Computerprogramm, in das eine syntaktische Mutation eingebaut wurde, um festzustellen, ob diese Mutation im Kontext des Computerprogramms eine feine oder grobe Mutation ist. Ist die Mutation eine grobe Mutation so wird die Simulation beendet; ist die Mutation dagegen eine feine Mutation so wird versucht, diese Mutation innerhalb eines vorbestimmten Zeitraums zu erfassen. Das Verfahren bewertet somit die Güte eines Computerprogramms, indem es versucht, den Einfluß der Mutation im Kontext des Computerprogramms zu bestimmen. Das offenbarte System vergleicht dazu das mutationsbehaftete Computerprogramm mit dem nicht mutierten Computerprogramm. Es ist daher nur zur Bestimmung der Güte der Stimulation der Mutationen durch die Testumgebung geeignet. Auf die Güte des eigentlich zu testenden Computerprogramms kann daraus nur indirekt geschlossen werden. Fehler, die bereits in der nicht mutierten Schaltung enthalten sind, können nur dann erkannt werden, wenn die Mutation tatsächlich Auswirkungen auf die Fehler hat. Dies ist jedoch nicht zwingend der Fall. Die Güte des Computerprogramms kann somit mit dem vorgeschlagenen System nur unzureichend bestimmt werden.

Aufgabe der Erfindung ist es, die Nachteile nach dem Stand der Technik zu beseitigen. Insbesondere soll ein Verfahren angegeben werden, daß die Bewertung der Güte eines Testprogramms für mittels eines Computers simulierte integrierte Schaltungen mit höherer Qualität ermöglicht.

Diese Aufgabe wird durch die Merkmale des Anspruchs 1 gelöst. Zweckmäßige Ausgestaltungen der Erfindungen ergeben sich aus den Merkmalen der Ansprüche 2 bis 14.

Nach Maßgabe der Erfindung ist ein Verfahren zur Bewertung der Güte eines Testprogramms für mittels eines Computers simulierte integrierte Schaltungen vorgesehen, das folgendes umfaßt:

5

Verfahren zur Bewertung der Güte eines Testprogramms für mittels eines Computers simulierte integrierte Schaltungen, umfassend

10 (a) das Bereitstellen einer ersten Datei, die eine integrierte Schaltung beschreibt;

(b) das Simulieren einer mutierten integrierten Schaltung, die durch Einbau von Mutationen in die in der ersten Datei
15 beschriebene integrierte Schaltung erhalten wird;

(c) das Zuführen von Eingangswerten zu der mutierten integrierten Schaltung und Erfassen der für diese Eingangswerte von der mutierten integrierten Schaltung erzeugten Ausgangswerte;
20

(d) den Vergleich der von der mutierten integrierten Schaltung erzeugten Ausgangswerte mit Erwartungswerten, die von dem Testprogramm bereitgestellt werden, wobei die Erwartungswerte in einem Referenzsystem generiert worden sind; und
25

(e) die Bewertung der Güte des Testprogramms anhand der Vergleichsergebnisse.

30 Das erfindungsgemäße Verfahren vergleicht somit die von der mutierten integrierten Schaltung genierten Ausgangswerte mit Erwartungswerten, die aus einer unabhängigen Quelle, d. h.

dem Referenzsystem stammen. Die Erwartungswerte sollen in einem Referenzsystem generiert werden, das unabhängig von der in der ersten Datei beschriebenen integrierten Schaltung ist. Unter "unabhängig" wird dabei verstanden, daß das Referenzsystem nicht auf der ersten Datei basiert, sondern funktionelle Unterschiede gegenüber der integrierten Schaltung, die in der ersten Datei beschrieben wird, aufweist. Vorteilhafterweise kann das Referenzsystem in einer anderen Programmiersprache und/oder durch einen zweiten Programmierer programmiert sein.

10

In einer ersten Ausführungsform umfaßt Schritt (b) folgende Schritte:

(b1) das Erzeugen einer zweiten Datei, die die mutierte integrierte Schaltung beschreibt, aus der ersten Datei durch den Einbau der Mutationen in die erste Datei; und

(b2) das Simulieren der mutierten integrierten Schaltung.

Alternativ kann Schritt (b) in einer zweiten Ausführungsform folgende Schritte umfassen:

(b1) das Einlesen der ersten Datei in das Simulationsprogramm des Computers;

25

(b2) das Abbilden der in der ersten Datei beschriebenen integrierten Schaltung in dem Simulationsprogramm;

(b3) den Einbau der Mutationen in die in dem Simulationsprogramm abgebildete integrierte Schaltung, wodurch die mutierte integrierte Schaltung erhalten wird; und

30

(b4) das Simulieren der mutierten integrierten Schaltung.

Die erste Datei kann aus einer Vielzahl von Unterdateien bestehen, die dann zusammen die integrierte mutierte Schaltung
5 beschreiben.

Das Simulationsprogramm des Computers ist der Teil des Computers, indem die Simulation der mutierten integrierten Schaltung durchgeführt wird.
10

Der Einbau der Mutationen in die integrierte Schaltung umfaßt in beiden Ausführungsformen zweckmäßigerweise folgende Schritte:

15 (i) Analyse der ersten Datei, um die Stellen in der ersten Datei aufzufinden, an denen Mutationen eingebaut werden können, und für jede aufgefundene Stelle den Typ der Mutation zu bestimmen, der an dieser Stelle eine Änderung des Verhaltens der integrierten Schaltung bewirkt;

20 (ii) Analyse der Tests, die das Testprogramm ausführen kann, um die Tests auszuwählen, die Eingangswerte bereitstellen, die eine Mutation ansprechen können; und

25 (iii) Auswahl der Mutationen, die an einer gemäß Schritt (i) aufgefundenen Stelle eine Änderung des Verhaltens der integrierten Schaltung bewirken und die durch einen der Tests angesprochen werden können, und Erzeugen der mutierten integrierten Schaltung.

30 Unter dem Ausdruck "eine Mutation ansprechen" ist im Sinne der vorliegenden Erfindung zu verstehen, daß eine aktiv ge-

schaltete Mutation an der Verarbeitung der Eingangswerte beteiligt und an der Generierung der Ausgangswerte mitwirken kann. Ob sie tatsächlich beteiligt ist oder mitwirkt, d. h. eine Veränderung der Ausgangswerte bewirkt, wird durch den
5 Vergleich der Ausgangswerte mit den Erwartungswerten überprüft.

Unter einer aktivierten, aktiven oder aktiv geschalteten Mutation ist folgendes zu verstehen: Erfindungsgemäß weist die
10 mutierte integrierte Schaltung Mutationen gegenüber der in der ersten Datei beschriebenen integrierten Schaltung auf. Diese Mutationen sind zunächst inaktiviert, d. h. die in der ersten Datei beschriebene integrierte Schaltung liefert genau dieselben Ausgangswerte wie die mutierte integrierte Schal-
15 tung, da keine der Mutationen an der Verarbeitung der Eingangswerte teilnimmt. Im Lauf des Verfahrens wird jedoch nacheinander immer genau eine der eingebauten Mutationen aktiv geschaltet, d. h. die aktiv geschaltete Mutation kann an der Verarbeitung der Eingangswerte teilnehmen. Dabei wurden
20 die Tests, die die Eingangswerte liefern, so ausgewählt, daß sie der aktiv geschaltete Mutation ansprechen.

Tests, die eine der ausgewählten Mutationen zumindest einmal ansprechen können, werden zweckmäßigerweise zu einem Testset
25 zusammengefaßt. Tests, die eine ausgewählte Mutation zumindest einmal ansprechen können, werden im folgenden auch als dieser Mutation zugeordnete Tests bezeichnet. Testsets, die diese Tests umfassen, werden auch als dieser Mutation zugeordnete Testsets bezeichnet.

30

Unter einer Mutation wird im Sinne der vorliegenden Erfindung eine Veränderung der ersten Datei verstanden, die eine funk-

tionelle Veränderung der elektronischen Schaltung, die auf Basis dieser Datei generiert wird, bewirkt. Die Mutation kann in einer Anweisung, die in der Datei beschrieben ist, positioniert werden oder über mehrere Testzeilen der Datei verteilt sein. Im Sinne der vorliegenden Erfindung sind Mutationen funktionelle Mutationen, die in die RTL-Ebene (RTL = Register Transfer Level) oder eine höhere Ebene (Systemebene oder algorithmische Ebene) der integrierten Schaltung eingebaut werden. Im folgenden ist unter einer Mutation eine funktionelle Mutation zu verstehen, sofern nichts anderes angegeben ist.

Geeignete Mutationstypen sind beispielsweise logische Negationsfehler, Auslassen logischer Faktoren, unkorrekte logische Faktoren, Auslassungen von logischen Ausdrücken, unkorrekte logische Ausdrücke, unkorrekte numerische Werte und Fallauslassungen. Die Mutationen im Sinne der vorliegenden Erfindung sind willentliche Veränderungen. Mutationen, die in der ersten Datei bereits enthalten sind, stellen Änderungen gegenüber dem vom Programmierer tatsächlich beabsichtigten Programmierzziel, das in der Datei abgebildet wird, dar. Solche nicht-willentlichen Mutationen sollen in Tests, während denen die in der ersten Datei beschriebene integrierte Schaltung simuliert wird, erkannt werden. Dies kann jedoch nur dann mit hinreichendem Erfolg der Fall sein, wenn das Testprogramm eine ausreichende Güte aufweist, d. h. die nicht-willentlichen Mutationen tatsächlich von dem Testprogramm erkannt werden können. Diese Fähigkeit, d. h. die Fähigkeit des Testprogramms, Mutationen zu erkennen, wird durch den willentlichen Einbau von Mutationen in die erste Datei und Simulieren der erhaltenen mutierten integrierten Schaltung überprüft. Erkennt das Testprogramm die willentlich eingebauten Mutationen

nicht, so ist nicht zu erwarten, daß es nicht-willentlich eingebaute Mutationen erkennen kann, da das Testprogramm in diesem Fall keine Tests ausführt, die Mutationen dieses Typs an der Stelle, an der die Mutation willentlich eingebaut worden ist, auffinden könnten. Das erfindungsgemäße Verfahren, das es ermöglicht, zunächst das Testprogramm auf seine Eignung zu testen, die integrierte Schaltung zu testen, die in der ersten Datei beschrieben wird, stellt somit eine erhebliche qualitative Verbesserung gegenüber den bisherigen Testverfahren für integrierte Schaltungen dar, da das Testergebnis, das beim Testen der ersten Datei durch Simulation der darin beschriebenen integrierten Schaltung erhalten wird, entscheidend von der Güte des Testprogramms abhängt. Ist die Güte des Testprogramms gering (dies ist der Fall, wenn es keine oder keine hinreichende Zahl von Tests bereitstellt, die die in der (ersten) Datei enthaltenen nicht-willentlichen Mutationen auffinden können), so wird bei Ausführung dieses Testprogramms ein positives Testergebnis erhalten, das den Tatsachen aufgrund der Mängel des Testprogramms nicht entspricht. Die aufgrund dieses Testergebnisses hergestellte integrierte Schaltung ist tatsächlich nicht hinreichend getestet worden und kann erhebliche Fehler, die aus den nicht-willentlichen Mutationen resultieren, aufweisen. Demgegenüber bietet das erfindungsgemäße Verfahren aufgrund der Überprüfung des Testprogramms eine hohe Gewähr dafür, daß die integrierte Schaltung tatsächlich hinreichend überprüft worden ist, da Testprogramme mit geringer Güte, d. h. Testprogramme, die keine oder keine hinreichende Zahl von Tests bereitstellen, die die in der (ersten) Datei enthaltenen nicht-willentlichen Mutationen auffinden können, zum Testen der ersten Datei nicht eingesetzt werden oder so verändert werden,

daß sie Tests bereitstellen, die die nicht-willentlichen Mutationen auffinden können.

Im folgenden wird unter einer Mutation eine willentliche Mutation verstanden.

Die Güte des Testprogramms wird, wie bereits erläutert, durch den Vergleich der generierten Ausgangswerte, die von der simulierten integrierten Schaltung generiert werden, mit den Erwartungswerten, die in dem Testprogramm enthalten sind, bestimmt. Das Ergebnis dieses Vergleiches kann als negativ angesehen werden, wenn der für eine Mutation erhaltene Ausgangswert dem für den Ausgangswert vom Referenzsystem generierten Erwartungswert gleicht. In diesem Fall ist die Mutation, die ja eine Veränderung des Ausgangswertes hätte bewirken müssen, von den Tests, die von dem Testprogramm ausgeführt worden sind, nicht erkannt worden.

Im Gegensatz dazu kann das Ergebnis des Vergleiches als positiv angesehen werden, wenn der für eine Mutation erhaltene Ausgangswert ungleich dem für den Ausgangswert vom Referenzsystem erzeugten Erwartungswert ist. In diesem Fall hat die Mutation eine Veränderung des Ausgangswertes bewirkt (oder die Veränderung des Ausgangswertes beruht auf einer nicht-willentlichen Mutation), was zeigt, daß der Test, der von dem Testprogramm ausgeführt wird, für das Testen der simulierten Schaltung nützlich ist, da er eine Veränderung, d.h. die Abweichung der von der simulierten Schaltung generierten Ausgangswerte von den Erwartungswerten, erkennt.

30

Die Anforderungen, die an die Güte des Testprogramms gestellt werden, hängen natürlich von den Vorgaben ab, die von dem An-

wender an die Güte der integrierten Schaltung, die in der ersten Datei beschrieben wird, gestellt werden. In einer bevorzugten Ausführungsform wird die Güte des Testprogramms aus dem Verhältnis der Anzahl der positiven Vergleichsergebnisse zur Anzahl der negativen Vergleichsergebnisse ermittelt. Dann kann die Güte des Testprogramms mit gut bewertet werden, wenn für jede eingebaute Mutation ein positives Vergleichsergebnis erhalten wird. Alternativ kann die Güte des Testprogramms mit gut bewertet werden, wenn die Anzahl der Mutationen, für die ein positives Vergleichsergebnis erhalten wird, einen vorgegebenen Wert übersteigt.

Ist das Ergebnis des Vergleiches für eine Mutation negativ, so kann das Verfahren nach einer der folgenden Alternativen fortgeführt werden:

Variante 1: Der Erwartungswert für diese Mutation wird verändert und anschließend die Schritte (b) bis (e) wiederholt werden. Dies ist insbesondere dann sinnvoll, wenn zweifelhaft ist, ob die Erwartungswerte, die von dem Referenzsystem geliefert werden, tatsächlich den vom Programmier der ersten Datei beabsichtigten Ausgangswerten entsprechen.

Variante 2: Die erste Datei wird verändert, und anschließend werden die Schritte (b) bis (e) wiederholt. Dies ist insbesondere dann sinnvoll, wenn das negative Ergebnis möglicherweise auf nicht-willentliche Mutationen in der ersten Datei zurückzuführen ist, die die willentlichen Mutationen derart beeinflussen, daß die von letzteren bewirkten Veränderungen sich nicht auf die Ausgangswerte auswirken.

Variante 3: Es werden Tests ausgewählt, die dieser Mutation ansprechen und gegenüber den bisherigen Tests veränderte Eingangswerte bereitstellen, und anschließend die Schritte (b) bis (e) wiederholt. Dies ist insbesondere dann sinnvoll, wenn
5 vermutet werden kann, daß weder dem Referenzsystem noch der simulierten integrierten Schaltung Eingangswerte zugeführt worden sind, die zu Erwartungswerten bzw. Ausgangswerten führten, die im Hinblick auf den beabsichtigten Verwendungszweck der integrierten Schaltung sinnvoll sind.

10

Die mutierte integrierte Schaltung kann mittels eines Hardwarebeschleunigers realisiert werden. Eine bevorzugter Hardwarebeschleuniger ist eine feldprogrammierbare Gateanordnung (FPGA).

15

Das erfindungsgemäße Verfahren ist nicht auf die Bewertung der Güte eines Testprogramms für mittels eines Computers simulierte integrierte Schaltungen beschränkt. Das Verfahren kann ebenso zur Bewertung der Güte von Computerprogrammen
20 verwendet werden. In diesem Fall ist unter dem Begriff "integrierte Schaltung" ein Computerprogramm und unter dem Begriff "mutierte integrierte Schaltung" ein mutiertes Computerprogramm zu verstehen. Ferner ist der Begriff "Simulieren" durch "Ausführen" zu ersetzen.

25

In diesem Fall ist somit ein Verfahren zur Bewertung der Güte eines Testprogramms für ein Computerprogramm, umfassend

(a) das Bereitstellen einer ersten Datei, die ein Computerprogramm beschreibt;
30

(b) das Ausführen eines mutierten Computerprogramms, das durch Einbau von Mutationen in das in der ersten Datei beschriebene Computerprogramm erhalten wird;

5 (c) das Zuführen von Eingangswerten zu dem mutierten Computerprogramm und Erfassen der für diese Eingangswerte von dem mutierten Computerprogramm erzeugten Ausgangswerte;

(d) den Vergleich der von dem mutierten Computerprogramm erzeugten Ausgangswerte mit Erwartungswerten, die von dem Testprogramm bereitgestellt werden, wobei die Erwartungswerte in einem Referenzsystem generiert worden sind; und

10

(e) die Bewertung der Güte des Testprogramms anhand der Vergleichsergebnisse.

15

Die Erfindung wird nachstehend anhand von Zeichnungen näher erläutert. Dabei zeigt

20 Fig. 1 ein Flußdiagramm zur Veranschaulichung einer ersten Ausführungsform des erfindungsgemäßen Verfahrens;

Fig. 2 die erste Datei in Form eines HDL-Files einer beispielhaften unmutierten integrierten Schaltung; und

25

Fig. 3 die aus der ersten Datei erhaltene zweite Datei in Form eines HDL-Files einer mutierten integrierten Schaltung.

30 Fig. 4 ein Flußdiagramm zur Veranschaulichung einer zweiten Ausführungsform des erfindungsgemäßen Verfahrens; und

Fig. 5 eine schematische Darstellung des der Erfindung zugrundeliegenden Prinzips.

5 Nach Fig. 1, die eine erste Ausführungsform des erfindungsge-
mäßigen Verfahrens zeigt, ist die erste Datei ein HDL-File 1.
Der HDL-File 1 beschreibt eine integrierte Schaltung, die
nicht-willentliche Mutationen enthalten kann. Die erste Datei
wird in den Computer, der durch eine gestrichelte Linie dar-
10 gestellt ist, überführt. In dem Computer wird aus dem HDL-
File 1 ein mutierter HDL-File erzeugt, der die zweite Datei
darstellt und als HDL'-File 2 bezeichnet ist. Dazu werden
funktionelle Mutationen 3, die von dem Testprogramm bereitge-
stellt werden, in der beschriebenen Weise in den HDL-File 1
15 eingebaut. Der erhaltene HDL'-File 2 beschreibt eine mutierte
integrierte Schaltung, die auf der in dem HDL-File 1 be-
schriebenen integrierten Schaltung basiert, jedoch zusätzlich
Mutationen aufweist.

20 Die im HDL'-File 2 beschriebene mutierte integrierte Schal-
tung wird von dem Simulationsprogramm simuliert. Während der
Simulation 4 werden von einem Steuerprogramm 52 (siehe dazu
Fig. 5) die eingebauten Mutationen nacheinander und - sofern
dies zweckmäßig ist - in einer vorgegebenen Reihenfolge akti-
25 viert, so daß jeweils nur eine Mutation aktiviert ist. Das
Testprogramm 70 führt die Tests aus, die die jeweils akti-
vierte Mutation ansprechen können. Diese Tests sind in Test-
sets zusammengefaßt. Das Testprogramm 70 generiert in Abhän-
gigkeit von dem ausgewählten Test Eingangswerte EgW, die der
30 simulierten mutierten integrierten Schaltung 61 zugeführt
werden. Diese Eingangswerte EgW werden von der simulierten
mutierten integrierten Schaltung 61 verarbeitet und dabei in

Ausgangswerte AW überführt. Die Ausgangswerte AW werden dann von dem Testprogramm 70 mit Erwartungswerten EW verglichen (Bezugszeichen 6 in Fig. 1), die von dem Testprogramm 70 mittels eines Referenzsystems 5 (in Fig. 5 Bezugszeichen 72) generiert werden. Die in dem Testset zusammengefaßten Tests werden solange nacheinander ausgeführt, bis ein Test zu Ausgangswerten AW führt, die sich von den Erwartungswerten EW unterscheiden. Anschließend wird die aktivierte Mutation inaktiv geschaltet und - sofern noch nicht alle eingebauten Mutationen getestet worden sind - die nächste Mutation aktiviert, der wiederum ein Testset mit Tests, die diese nun aktivierte Mutation ansprechen, zugeordnet ist.

Es ist somit nicht erforderlich, alle Tests, die in einem Testset zusammengefaßt sind, auszuführen. Die Ausführung der Tests eines Testsets für eine Mutation kann abgebrochen werden, sobald die Ausgangswerte, die bei der Ausführung eines Tests erhalten werden, sich von den Erwartungswerten unterscheiden. Wird festgestellt, daß ein Test in einem Testset die zugehörige Mutation tatsächlich anspricht (was sich in der fehlenden Übereinstimmung der Ausgangswerte mit den Erwartungswerten widerspiegelt), so kann aus dieser Feststellung gefolgert werden, daß das Testprogramm zumindest einen Test bereitstellt, der die zugehörige Mutation testen kann. Dieser Fall ist ein positives Testergebnis, da das Testprogramm die Mutation erkennt. Die Güte des Testprogramms ist somit in bezug auf die eine Mutation gut. Werden für keinen Test des Testsets Ausgangswerte erhalten, die sich von den Erwartungswerten unterscheiden, so hat die aktive Mutation keinen Einfluß auf die Verarbeitung der Eingangswerte zu Ausgangswerten gehabt. Dieser Fall ist ein negatives Testergebnis, da das Testprogramm die Mutation nicht erkennt. Die Güte

des Testprogramms ist somit in bezug auf die eine Mutation schlecht.

Aus der Anzahl der auf diese Weise vom Testprogramm erkannten
5 Mutationen kann die Güte des Testprogramms ermittelt werden
(Bezugszeichen 7 in Fig. 1). Wird jede Mutation erkannt, so
ist die Güte des Testprogramms gut. Werden nicht alle Muta-
tionen erkannt, so ist dieses Ergebnis ein schlechteres Er-
gebnis, wobei allerdings aus dem Verhältnis der erkannten Mu-
10 tationen zu den nicht erkannten Mutationen ein Wert für die
Güte berechnet werden kann, der wiederum mit einem vorgegebe-
nen Wert verglichen wird. Ist der berechnete Wert für die Gü-
te höher als der vorgegebene Wert, so kann das Testprogramm
als gut bewertet werden.

15

In Fig. 2 wird ein Beispiel einer ersten Datei in Form eines
HDL-File 1 gezeigt, aus dem, wie in Fig. 3 gezeigt wird, ein
mutierter HDL'-File 2 erhalten wurde, indem Mutationen in den
HDL'-File 1 eingebaut worden sind.

20

Der in Fig. 2 dargestellte HDL-File enthält den vollständigen
HDL-Code für einen 4-Bit-Vorwärts/Rückwärts-Zähler, der die
Datenausgänge C0, C1, C2 und C3 aufweist und von 0 bis 15
oder in entgegengesetzter Richtung zählt. Er umfaßt folgende
25 Dateneingänge:

Reset: der Zähler wird bei Reset = 0 auf den Wert 0 zu-
rückgesetzt, wenn eine steigende Clock-Flanke vor-liegt (d.
h. Clock von 0 auf 1 geht);

30

Clock: Takteingang - steigende Flanke (auch als "Clock-
Flanke" bezeichnet)

UpDown: der Zähler addiert bei UpDown = 1; er subtrahiert bei UpDown = 0

5 Count: der Wert des Zählers wird mit der steigenden Clock-Flanke um eins erhöht, wenn Count = 1, Reset = 1 und UpDown = 1; der Wert des Zählers wird mit der steigenden Clock-Flanke um eins verringert, wenn Count = 1, Reset = 1 und UpDown = 0; der Wert des Zählers wird nicht verändert, wenn Count = 0 und
10 Reset = 1.

Der in Fig. 3 dargestellte HDL'-File enthält funktionelle Mutationen, die in den in Fig. 2 dargestellten HDL-File eingefügt worden sind. Der HDL'-Code beschreibt einen 4-Bit-
15 Vorwärts/Rückwärts-Zähler, der die Datenausgänge C0, C1, C2 und C3 aufweist und von 0 bis 15 oder in entgegengesetzter Richtung zählt. Er umfaßt folgende Dateneingänge:

Reset: der Zähler wird bei Reset = 0 auf den Wert 0 zurückgesetzt, wenn eine steigende Clock-Flanke vorliegt (d.
20 h. Clock von 0 auf 1 geht);

Clock: Takteingang - steigende Flanke

25 UpDown: der Zähler addiert bei UpDown = 1; er subtrahiert bei UpDown = 0

Count: der Wert des Zählers wird mit der steigenden Clock-Flanke um eins erhöht, wenn Count = 1, Reset = 1 und UpDown =
30 1; der Wert des Zählers wird mit der steigenden Clock-Flanke um eins verringert, wenn Count = 1, Reset = 1 und UpDown = 0;

der Wert des Zählers wird nicht verändert, wenn Count = 0 und Reset = 1.

Em ermöglicht es, eine Mutation emi der in den HDL'-File
5 eingefügten Mutationen emn (n 1 bis 13) aktiv zu schalten, während alle anderen Mutationen inaktiv geschaltet sind.

Die einzelnen Mutationen sind in dem HDL'-File 2 binär als em
= "0011" (em1), em = 2"0100" (em2) ... em = "1110" (em13) co-
10 diert.

Nach Fig. 4, die eine zweite Ausführungsform des erfindungs-
gemäßen Verfahrens zeigt, ist die erste Datei ein HDL-
File 101. Der HDL-File 101 beschreibt eine integrierte Schal-
15 tung, die nicht-willentliche Mutationen enthalten kann. Die erste Datei wird in den Computer, der durch eine gestrichelte Linie dargestellt ist, überführt, in die Simulationseinheit des Computers eingelesen (Bezugszeichen 102a) und dort die integrierte Schaltung abgebildet (Bezugszeichen 102b). Aus
20 der abgebildeten integrierten Schaltung wird eine mutierte integrierte Schaltung erzeugt, indem funktionelle Mutationen 103, die von dem Testprogramm bereitgestellt werden, in der beschriebenen Weise eingebaut werden. Es wird eine mutierte integrierte Schaltung erhalten, die auf der in dem
25 HDL-File 1 beschriebenen integrierten Schaltung basiert, jedoch zusätzlich Mutationen aufweist.

Die mutierte integrierte Schaltung wird von der Simulationseinheit simuliert (Bezugszeichen 104). Während der Simulati-
30 on 104 werden von einem Steuerprogramm 52 (siehe dazu Fig. 5) die eingebauten Mutationen nacheinander und - sofern dies zweckmäßig ist - in einer vorgegebenen Reihenfolge aktiviert,

so daß jeweils nur eine Mutation aktiviert ist. Das Testprogramm führt die Tests aus, die die jeweils aktivierte Mutation ansprechen können. Diese Tests sind in Testsets zusammengefaßt. Das Testprogramm 70 generiert in Abhängigkeit von dem
5 ausgewählten Test Eingangswerte EgW, die der simulierten mutierten integrierten Schaltung 61 zugeführt werden. Diese Eingangswerte EgW werden von der simulierten mutierten integrierten Schaltung 61 verarbeitet und dabei in Ausgangswerte AW überführt. Die Ausgangswerte AW werden dann von dem Test-
10 programm 70 mit Erwartungswerten EW verglichen (Bezugszeichen 106 in Fig. 4), die von dem Testprogramm mittels eines Referenzsystems 105 (in Fig. 5 Bezugszeichen 72) generiert werden. Die in dem Testset zusammengefaßten Tests werden solange nacheinander ausgeführt, bis ein Test zu Ausgangswerten
15 führt, die sich von den Erwartungswerten unterscheiden. Anschließend wird die aktivierte Mutation inaktiv geschaltet und - sofern noch nicht alle eingebauten Mutationen getestet worden sind - die nächste Mutation aktiviert, der wiederum ein Testset mit Tests, die diese nun aktivierte Mutation ansprechen, zugeordnet ist.
20

Es ist somit nicht erforderlich, alle Tests, die in einem Testset zusammengefaßt sind, auszuführen. Die Ausführung der Tests eines Testsets für eine Mutation kann abgebrochen werden, sobald die Ausgangswerte, die bei der Ausführung eines
25 Tests erhalten werden, sich von den Erwartungswerten unterscheiden. Wird festgestellt, daß ein Test in einem Testset die zugehörige Mutation tatsächlich anspricht (was sich in der fehlenden Übereinstimmung der Ausgangswerte mit den Erwartungswerten widerspiegelt), so kann aus dieser Feststellung gefolgert werden, daß das Testprogramm zumindest einen
30 Test bereitstellt, der die zugehörige Mutation testen kann.

Dieser Fall ist ein positives Testergebnis, da das Testprogramm die Mutation erkennt. Die Güte des Testprogramms ist somit in bezug auf die eine Mutation gut. Werden für keinen Test des Testsets Ausgangswerte erhalten, die sich von den Erwartungswerten unterscheiden, so hat die aktive Mutation keinen Einfluß auf die Verarbeitung der Eingangswerte zu Ausgangswerten gehabt. Dieser Fall ist ein negatives Testergebnis, da das Testprogramm die Mutation nicht erkennt. Die Güte des Testprogramms ist somit in bezug auf die eine Mutation schlecht.

Aus der Anzahl der auf diese Weise vom Testprogramm erkannten Mutationen kann die Güte des Testprogramms ermittelt werden (Bezugszeichen 107 in Fig. 1). Wird jede Mutation erkannt, so ist die Güte des Testprogramms gut. Werden nicht alle Mutationen erkannt, so ist dieses Ergebnis ein schlechteres Ergebnis, wobei allerdings aus dem Verhältnis der erkannten Mutationen zu den nicht erkannten Mutationen ein Wert für die Güte berechnet werden kann, der wiederum mit einem vorgegebenen Wert verglichen wird. Ist der berechnete Wert für die Güte höher als der vorgegebene Wert, so kann das Testprogramm als gut bewertet werden.

Fig. 5 zeigt in schematischer Darstellung einen Computer 51, der ein Simulationsprogramm (nicht gezeigt), ein Steuerprogramm 52, ein Programm 53 zur Bewertung der Güte des Testprogramms auf Basis der Anzahl der erkannten und der Anzahl der nicht-erkannten Mutationen sowie eine Einrichtung 54 zur Ausgabe des Bewertungsergebnisses umfaßt. Das Simulationsprogramm simuliert die mutierte integrierte Schaltung 61, die gemäß den in Fig. 1 und Fig. 4 dargestellten Ausführungsformen erhalten werden kann. Das Simulationsprogramm führt fer-

ner das Testprogramm 70 aus. Das Ausführen des Testprogramms 70 und die Simulation der mutierten integrierten Schaltung 61 sind als Simulation 60 zusammengefaßt. Das Testprogramm 70 generiert 71 die Eingangswerte EgW für jeden auszuführenden Test und generiert 73 die Erwartungswerte EW mittels eines Referenzsystems 72. Es kann zusätzlich den Vergleich 74 der von der mutierten integrierten Schaltung generierten Ausgangswerte AW ausführen, wobei der Vergleich auch außerhalb des Testprogramms 70 von dem Computer 51 vorgenommen werden kann.

Das Steuerprogramm 52 steuert die Simulation 60 (Pfeil 53) und das Testprogramm 70 (Pfeil 54). Es schaltet ferner die Mutationen, die der mutierte integrierten Schaltung 61 enthalten sind, aktiv und inaktiv (Pfeil 55). Es weist das Testprogramm an, die Tests auszuführen, die der aktiv geschalteten Mutation zugeordnet sind.

Das Testprogramm 70 führt nach den Vorgaben des Steuerprogramms 52 die Tests für die von dem Steuerprogramm 52 aktiv geschalteten Mutationen aus, wozu es die mutierte integrierte Schaltung 61 mit Eingangswerten EgW versorgt. Die von der mutierten integrierten Schaltung 61 für die Eingangswerte EgW erhaltenen Ausgangswerte AW werden dem Testprogramm 70 wiederzugeführt, sofern der Vergleich 74 von dem Testprogramm ausgeführt werden soll und nicht an anderer Stelle von dem Computer 51 durchgeführt wird. Werden die Ausgangswerte AW dem Testprogramm 70 zugeführt, so führt es einen Vergleich 74 zwischen den für die aktiv geschaltete Mutation erhaltenen Ausgangswerten AW und Erwartungswerten EW durch. Anhand der Vergleichsergebnisse wird die Bewertung 53 der Güte des Testprogramms 70 vorgenommen und ein Bericht über ein geeignetes

Mittel 54, beispielsweise einen Monitor oder einen Drucker, ausgegeben.

Das Ergebnis des Vergleichs wird ferner dem Steuerprogramm 52
5 übermittelt, daß, wenn die für eine aktiv geschaltete Mutation erhaltenen Ausgangswerte AW sich von den Erwartungswerten EW unterscheiden, die Ausführung der Tests, die dieser Mutation zugeordnet sind, abbricht, die bisher aktiv geschaltete Mutation inaktiv schaltet und die nächste Mutation aktiv
10 schalten kann (gestrichelter Pfeil 56).

Patentansprüche

1. Verfahren zur Bewertung der Güte eines Testprogramms für
5 mittels eines Computers simulierte integrierte Schaltungen,
umfassend
- (a) das Bereitstellen einer ersten Datei, die eine integrierte
Schaltung beschreibt;
- 10 (b) das Simulieren einer mutierten integrierten Schaltung,
die durch Einbau von Mutationen in die in der ersten Datei
beschriebene integrierte Schaltung erhalten wird;
- 15 (c) das Zuführen von Eingangswerten zu der mutierten integrierten
Schaltung und Erfassen der für diese Eingangswerte von der
mutierten integrierten Schaltung erzeugten Ausgangswerte;
- 20 (d) den Vergleich der von der mutierten integrierten Schaltung
erzeugten Ausgangswerte mit Erwartungswerten, die von dem
Testprogramm bereitgestellt werden, wobei die Erwartungswerte
in einem Referenzsystem generiert worden sind; und
- 25 (e) die Bewertung der Güte des Testprogramms anhand der
Vergleichsergebnisse.
- 30 2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß
Schritt (b) folgende Schritte umfaßt:

(b1) das Erzeugen einer zweiten Datei, die die mutierte integrierte Schaltung beschreibt, aus der ersten Datei durch den Einbau der Mutationen in die erste Datei; und

5

(b2) das Simulieren der mutierten integrierten Schaltung.

3. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß
10 Schritt (b) folgende Schritte umfaßt:

(b1) das Einlesen der ersten Datei in das Simulationsprogramm des Computers;

15 (b2) das Abbilden der in der ersten Datei beschriebenen integrierten Schaltung in der Simulationseinheit;

(b3) den Einbau der Mutationen in die in der Simulationseinheit abgebildete integrierte Schaltung, wodurch die mutierte integrierte Schaltung erhalten wird; und

20

(b4) das Simulieren der mutierten integrierten Schaltung.

25 4. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß die erste Datei aus einer Vielzahl von Unterdateien besteht, die zusammen die integrierte mutierte Schaltung beschreiben.

5. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß der Einbau der Mutationen in die integrierte Schaltung folgende Schritte umfaßt:

5 (i) Analyse der ersten Datei, um die Stellen in der ersten Datei aufzufinden, an denen Mutationen eingebaut werden können, und für jede aufgefundene Stelle den Typ der Mutation zu bestimmen, der an dieser Stelle eine Änderung des Verhaltens der integrierten Schaltung bewirkt;

10

(ii) Analyse der Tests, die das Testprogramm ausführen kann, um die Tests auszuwählen, die Eingangswerte bereitstellen, die eine Mutation ansprechen können; und

15 (iii) Auswahl der Mutationen, die an einer gemäß Schritt (i) aufgefundenen Stelle eine Änderung des Verhaltens der integrierten Schaltung bewirken und die durch einen der Tests angesprochen werden können, und Erzeugen der mutierten integrierten Schaltung.

20

6. Verfahren nach Anspruch 5, dadurch gekennzeichnet, daß es ferner das Zusammenstellen der Tests, die eine der ausgewählten Mutation zumindest einmal ansprechen können, zu einem Testset umfaßt.
- 25

7. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß das Ergebnis des Vergleiches negativ ist, wenn der für eine Mutation erhaltene Ausgangswert dem für den Ausgangswert generierten Erwartungswert gleicht.
- 30

8. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß das Ergebnis des Vergleiches positiv ist, wenn der für eine Mutation erhaltene Ausgangswert ungleich dem für den Ausgangswert generierten Erwartungswert ist.
9. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß die Güte des Testprogramms aus dem Verhältnis der Anzahl der positiven Vergleichsergebnisse zur Anzahl der negativen Vergleichsergebnisse ermittelt wird.
10. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß die Güte des Testprogramms mit gut bewertet wird, wenn für jede eingebaute Mutation ein positives Vergleichsergebnis erhalten wird.
11. Verfahren nach einem der Ansprüche 1 bis 9, dadurch gekennzeichnet, daß die Güte des Testprogramms mit gut bewertet wird, wenn die Anzahl der Mutationen, für die ein positives Vergleichsergebnis erhalten wird, einen vorgegebenen Wert übersteigt.
12. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß, wenn das Ergebnis des Vergleiches für eine Mutation negativ ist, der Erwartungswert für diese Mutation verändert wird und anschließend die Schritte (b) bis (e) wiederholt werden.
13. Verfahren nach einem der Ansprüche 1 bis 11, dadurch gekennzeichnet, daß, wenn das Ergebnis des Vergleiches für

eine Mutation negativ ist, die erste Datei verändert wird und anschließend die Schritte (b) bis (e) wiederholt werden.

- 5 14. Verfahren nach einem der Ansprüche 1 bis 11, dadurch gekennzeichnet, daß, wenn das Ergebnis des Vergleiches für eine Mutation negativ ist, Tests ausgewählt werden, die diese Mutation ansprechen und gegenüber den bisherigen Tests veränderte Eingangswerte bereitstellen, und an-
- 10 schließend die Schritte (b) bis (e) wiederholt werden.
- 15 15. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß die mutierte integrierte Schaltung mittels eines Hardwarebeschleunigers realisiert wird.

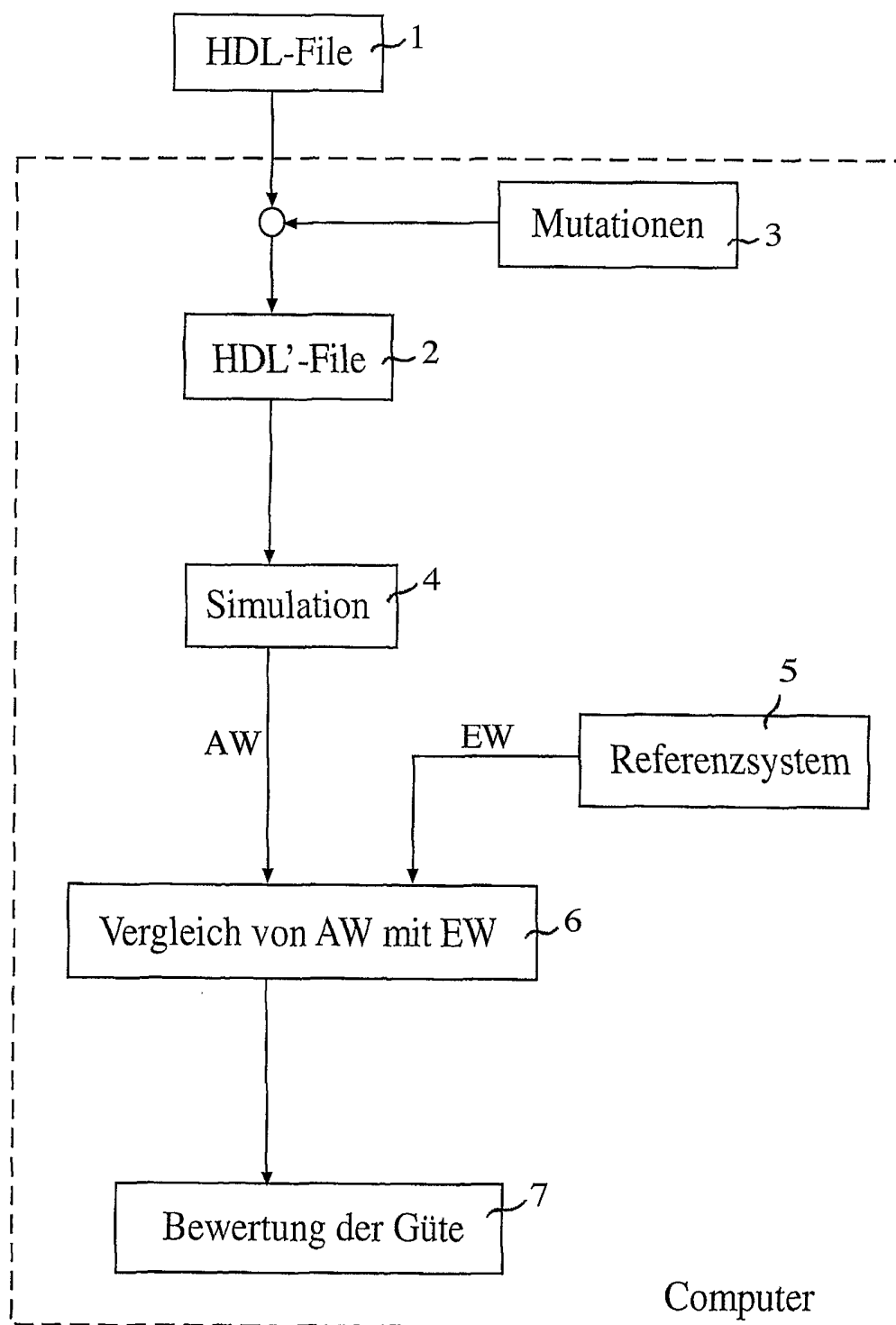


Fig. 1

Fig. 2

```

library IEEE;
use IEEE.std_logic_1164.all; -- import std_logic types
use IEEE.std_logic_arith.all; -- import add/sub of std_logic_vector

ENTITY counter_updown is
PORT ( -- Counter Control Signals
      clock      : IN  std_logic;  -- counter clock
      reset      : IN  std_logic;  -- counter reset
      updown     : IN  std_logic;  -- count direction up/down
      count      : IN  std_logic;  -- count enable
      -- Counter Output Signals
      c0, c1, c2, c3 : OUT std_logic); -- counter bit 0..3
END counter_updown;

ARCHITECTURE rtl OF counter_updown IS
  -- internal signal declaration
  SIGNAL counter : std_logic_vector(3 DOWNTO 0);
  BEGIN

    count_process : PROCESS
    -- This process builds the counter
    BEGIN
      -- wait for rising edge of clock (0->1)
      WAIT UNTIL clock'EVENT AND clock = '1';
      IF reset = '0' THEN
        -- set all counter bits to zero
        counter <= (others => '0');
      ELSE
        IF (count = '1') AND (updown = '1') THEN
          -- count up: n, n+1, n+2, ..., 15, 0, 1, ...
          counter <= CONV_STD_LOGIC_VECTOR(UNSIGNED(counter) + 1, 4);
        ELSEIF (count = '1') AND (updown = '0') THEN
          -- count down: n, n-1, n-2, ..., 0, 15, 14, ...
          counter <= CONV_STD_LOGIC_VECTOR(UNSIGNED(counter) - 1, 4);
        ELSE
          -- hold counter in the same value as before
          counter <= counter;
        END IF;
      END IF;
    END PROCESS;

    -- Connect internal counter to outputs
    c0 <= counter(0);
    c1 <= counter(1);
    c2 <= counter(2);
    c3 <= counter(3);

  END rtl;

```

Fig. 3

```

library std,work;
library IEEE;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use std.standard.all;

entity counter_updown is
  port (
    -- em is the control port to enable a mutant
    em      : in std_logic_vector (3 downto 0);
    clock   : in std_logic;
    reset    : in std_logic;
    updown   : in std_logic;
    count    : in std_logic;
    c0,c1,c2,c3 : out std_logic) ;
end counter_updown ;

library std,work;
architecture rtl of counter_updown is
  signal counter : std_logic_vector(3 downto 0);
begin
  count_process : process
  begin
    wait until (clock'event and (clock = '1'));
    -- conditional mutants
    if ((em="0011" and false) or
        (em="0100" and true) or
        (em="0101" and (not(reset = '0'))))
        or (em/="0011" and em/="0100" and em/="0101" and (reset =
'0')))) then
      -- dead assign mutant
      if em="0110" then
        else
          counter <= (others => '0');
        end if;
      else
        -- conditional mutants
        if ((em="0111" and false) or
            (em="1000" and true) or
            (em="1001" and (not((count = '1') and (updown = '1'))))
            or(em/="0111" and em/="1000" and em/="1001" and ((count =
'1') and (updown = '1')))) then
          -- dead assign mutant
          if em="1010" then
            else
              counter <= conv_std_logic_vector((unsigned(counter) +
1),4);
            end if;
          -- conditional mutants
          elsif ((em="1011" and false) or
                (em="1100" and true) or
                (em="1101" and (not((count = '1') and (updown = '0'))))
or
                (em/="1011" and em/="1100" and em/="1101" and ((count =
'1') and (updown = '0')))) then
            -- dead assign mutant

```

Fig. 3 (Fortsetzung)

```
        if em="1110" then
        else
            counter <= conv_std_logic_vector((unsigned(counter) -
1),4);
        end if;
    else
        -- dead assign mutant
        if em="1111" then
        else
            counter <= counter;
        end if;
    end if;
end if;
end process ;
c0 <= counter(0);
c1 <= counter(1);
c2 <= counter(2);
c3 <= counter(3);
end rtl ;
```

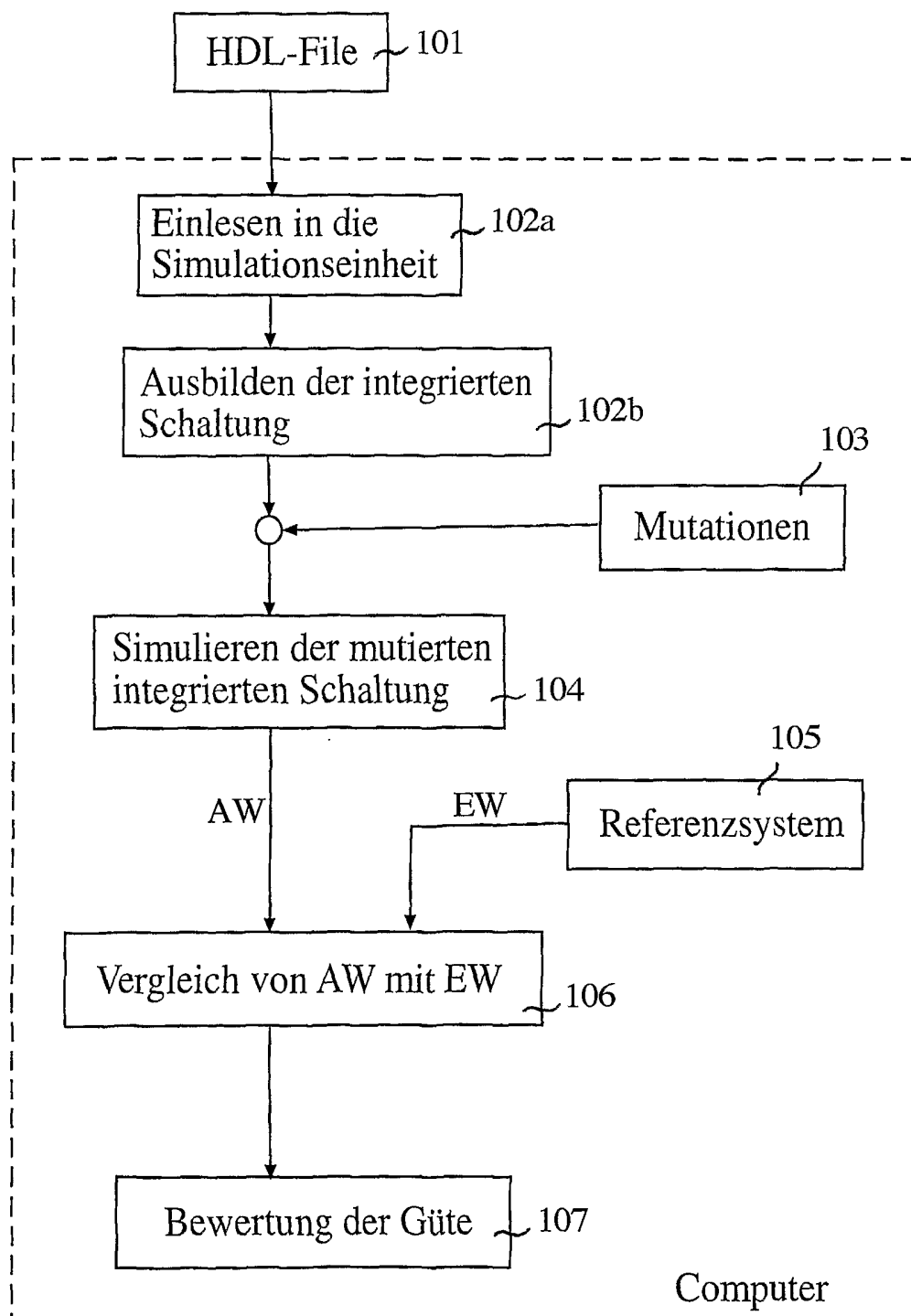


Fig. 4

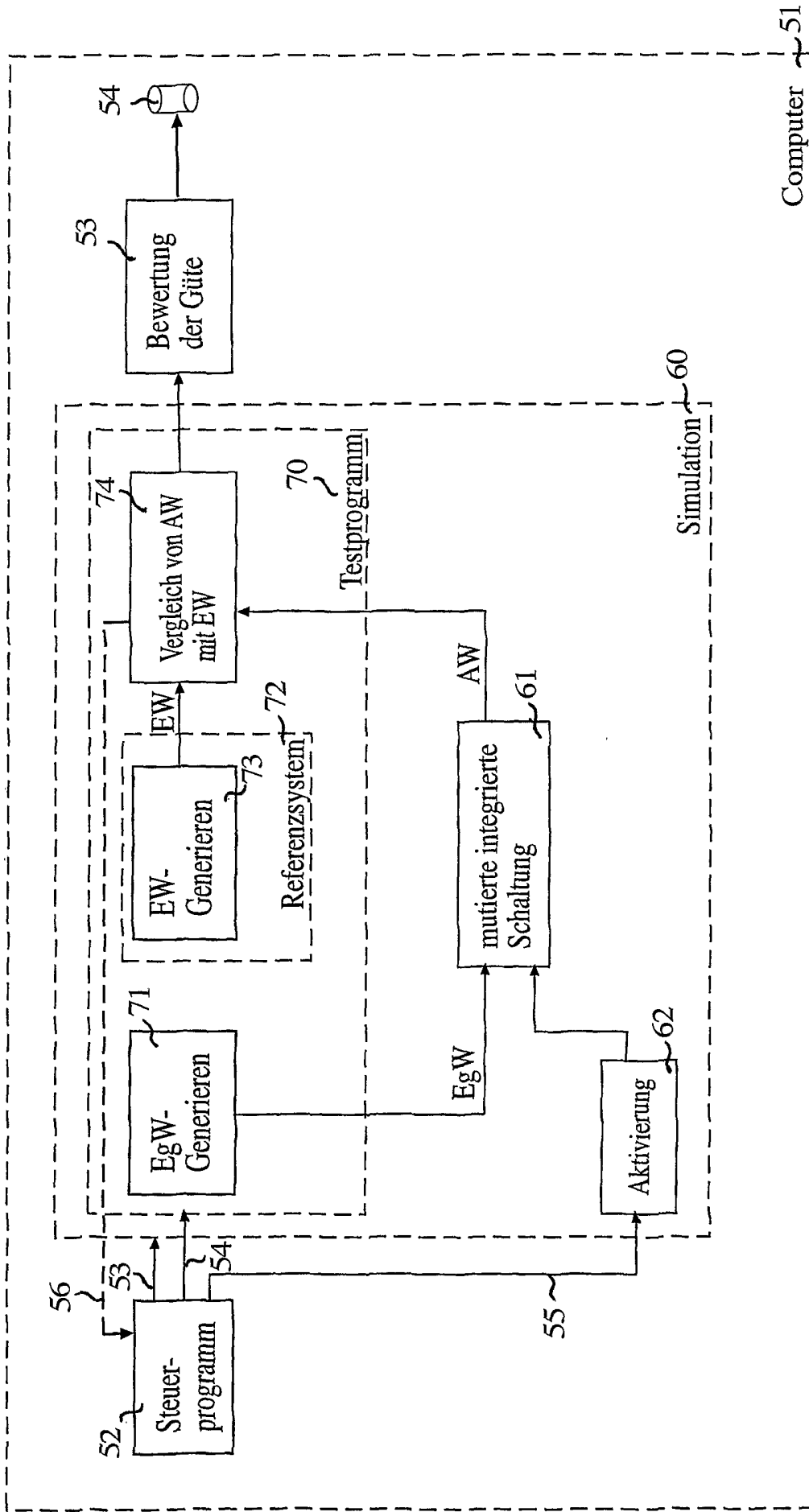


Fig. 5