

(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
4. Mai 2006 (04.05.2006)

PCT

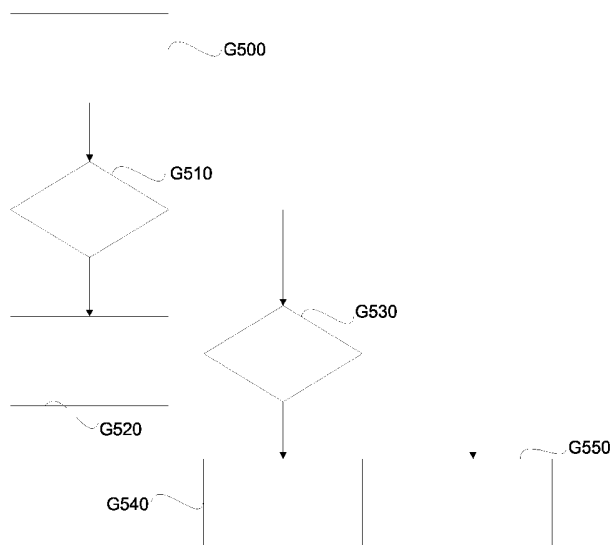
(10) Internationale Veröffentlichungsnummer
WO 2006/045780 A1

- (51) Internationale Patentklassifikation:
G06F 11/16 (2006.01) 10 2004 051 952.8
25. Oktober 2004 (25.10.2004) DE
10 2005 037 212.0 8. August 2005 (08.08.2005) DE
- (21) Internationales Aktenzeichen: PCT/EP2005/055507
- (22) Internationales Anmeldedatum:
25. Oktober 2005 (25.10.2005)
- (25) Einreichungssprache: Deutsch
- (26) Veröffentlichungssprache: Deutsch
- (30) Angaben zur Priorität:
10 2004 051 950.1 25. Oktober 2004 (25.10.2004) DE
10 2004 051 992.7 25. Oktober 2004 (25.10.2004) DE
10 2004 051 964.1 25. Oktober 2004 (25.10.2004) DE
10 2004 051 937.4 25. Oktober 2004 (25.10.2004) DE
- (71) Anmelder (für alle Bestimmungsstaaten mit Ausnahme von US): **ROBERT BOSCH GMBH** [DE/DE]; Postfach 30 02 20, 70442 Stuttgart (DE).
- (72) Erfinder; und
(75) Erfinder/Anmelder (nur für US): **ANGERBAUER, Ralf** [DE/DE]; Clara-Schumann-Strasse 4, 71701 Schwieberdingen (DE). **COLLANI, Yorck** [DE/DE]; Lisztweg 9, 71717 Beilstein (DE). **GMEHLICH, Rainer** [DE/DE]; Hoehenweg 2, 71254 Ditzingen (DE). **BOEHL, Eberhard** [DE/DE]; Heimbühlstrasse 36, 72768 Reutlingen (DE).
- (74) Gemeinsamer Vertreter: **ROBERT BOSCH GMBH**; Postfach 30 02 20, 70442 Stuttgart (DE).
- (81) Bestimmungsstaaten (soweit nicht anders angegeben, für jede verfügbare nationale Schutzrechtsart): AE, AG, AL,

[Fortsetzung auf der nächsten Seite]

(54) Title: METHOD AND DEVICE FOR SEPARATING THE PROCESSING OF PROGRAM CODES IN A COMPUTER SYSTEM COMPRISING AT LEAST TWO EXECUTION UNITS

(54) Bezeichnung: VERFAHREN UND VORRICHTUNG ZUR TRENNUNG DER ABARBEITUNG VON PROGRAMMCODE BEI EINEM RECHNERSYSTEM MIT WENIGSTENS ZWEI AUSFÜHRUNGSEINHEITEN



(57) Abstract: The invention relates to a method and a device for separating the processing of program codes in a computer system comprising at least two execution units. It is possible to switch between at least two operating modes, the first operating mode corresponding to a comparison mode and the second operating mode corresponding to a performance mode. The at least two execution units process the same program code in the comparison mode. The invention is characterised in that a separation in the program code is carried out during the switching from the comparison mode to the performance mode, by associating a characteristic with each execution unit, the association being carried out according to the characteristic of at least two execution units of different program codes.

[Fortsetzung auf der nächsten Seite]

WO 2006/045780 A1



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, LY, 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.

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:

- mit internationalem Recherchenbericht
- vor Ablauf der für Änderungen der Ansprüche geltenden Frist; Veröffentlichung wird wiederholt, falls Änderungen eintreffen

(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,

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

(57) Zusammenfassung: Verfahren und Vorrichtung zur Trennung der Abarbeitung von Programmcode bei einem Rechnersystem mit wenigstens zwei Ausführungseinheiten, wobei zwischen wenigstens zwei Betriebsmodi umgeschaltet wird und ein erster Betriebsmodus einem Vergleichsmodus und ein zweiter Betriebsmodus einem Performanzmodus entspricht und die wenigstens zwei Ausführungseinheiten im Vergleichsmodus den gleichen Programmcode abarbeiten dadurch gekennzeichnet, dass bei der Umschaltung vom Vergleichsmodus in den Performanzmodus eine Trennung im Programmcode dadurch erfolgt, dass jeder Ausführungseinheit eine Kennung zugeordnet ist und abhängig von der Kennung mindestens zwei Ausführungseinheiten unterschiedlicher Programmcode zugeordnet wird.

10 Verfahren und Vorrichtung zur Trennung der Abarbeitung von Programmcode bei einem Rechner-System mit wenigstens zwei Ausführungseinheiten

Stand der Technik

15 Transiente Fehler, ausgelöst durch Alpha-Teilchen oder kosmische Strahlung, werden zunehmend ein Problem für integrierte Halbleiterschaltungen. Durch abnehmende Strukturbreiten, sinkende Spannungen und höhere Taktfrequenzen nimmt die Wahrscheinlichkeit zu, dass eine Spannungsspitze, hervorgerufen durch ein Alpha-Teilchen oder kosmische Strahlung, einen logischen Wert in einer integrierten Schaltung verfälscht. Ein falsches Berechnungsergebnis kann die Folge sein. In sicherheitsrelevanten Systemen, insbesondere im Kraftfahrzeug, müssen solche Fehler daher zuverlässig detektiert werden.

20 Bei sicherheitsrelevanten Systemen, wie z.B. einem ABS-Regelsystem in einem Kraftfahrzeug, in denen Fehlfunktionen der Elektronik sicher detektiert werden müssen, werden bei den entsprechenden Steuereinrichtungen solcher Systeme üblicherweise Redundanzen zur Fehlererkennung eingesetzt. So ist beispielsweise in bekannten ABS-Systemen jeweils der komplette Mikrocontroller dupliziert, wobei die gesamten ABS-Funktionen redundant berechnet und auf Übereinstimmung geprüft werden. Tritt eine Diskrepanz der Ergebnisse auf, so wird das ABS-System abgeschaltet.

25 Wesentliche Komponenten eines Mikrocontroller sind einerseits aus Speichermodulen (z.B. RAM, ROM, Cache), den I/Os und die Ein- /Ausgangs-Schnittstellen, die so genannten Peripherals (z.B. A/D-Wandler, CAN-Schnittstelle). Da Speicherelemente mit Prüfcodes (Parity oder ECC) effektiv überwacht werden können, und Peripherals oft anwendungsspezifisch als

- 2 -

Teil eines Sensor- oder Aktor-Signalfades überwacht werden, besteht ein weiterer Redundanzansatz in der alleinigen Verdopplung der Cores eines Mikrocontrollers.

Solche Mikrocontroller mit zwei integrierten Cores sind auch als Dual-Core Architekturen bekannt. Beide Cores führen redundant und takt synchron (Lockstep-Modus) das gleiche Programmsegment aus, die Ergebnisse der beiden Cores werden verglichen, und ein Fehler wird dann bei dem Vergleich auf Übereinstimmung erkannt werden. Diese Konfiguration eines Dual-Core Systems kann als ein Vergleichsmodus bezeichnet werden.

Dual-Core Architekturen werden in anderen Anwendungen auch zur Leistungssteigerung, also zu einer Performanz-Steigerung eingesetzt. Beide Cores führen unterschiedliche Programme, Programmsegmente und Befehle aus, wodurch sich eine Leistungssteigerung erzielen lässt, weshalb diese Konfiguration eines Dual-Core Systems als ein Performanzmodus bezeichnet werden kann. Dieses System wird auch als ein symmetrisches Multiprozessorsystem (SMP) bezeichnet.

Eine Erweiterung dieser Systeme ist eine Umschaltung durch Software zwischen diesen beiden Modi mittel eines Zugriffs auf eine spezielle Adresse und spezialisierter Hardware-Vorrichtungen. Im Vergleichsmodus werden die Ausgangssignale der Cores miteinander verglichen. Im Performanzmodus arbeiten die beiden Cores als ein symmetrisches Mehrprozessorsystem (SMP) und führen unterschiedliche Programme, Programmsegmente oder Befehle aus.

Bei einem solchen System ist es ein Problem, beim Übergang von einem Vergleichs- in einen Performanzmodus die Programmflüsse geordnet zu trennen. Es ist deshalb Aufgabe der Erfindung Verfahren und Mittel zur Verfügung zu stellen, die eine solche Trennung in einfacher Form ermöglichen.

Vorteile der Erfindung

Vorteilhafterweise verwendet man ein Verfahren zur Trennung der Abarbeitung von Programmcode bei einem Rechnersystem mit wenigstens zwei Ausführungseinheiten, wobei zwischen wenigstens zwei Betriebsmodi umgeschaltet wird und ein erster Betriebsmodus einem Vergleichsmodus und ein zweiter Betriebsmodus einem Performanzmodus entspricht und die wenigstens zwei Ausführungseinheiten im Vergleichsmodus den gleichen Programmcode abarbeiten, dadurch gekennzeichnet, dass bei der Umschaltung vom

- 3 -

Vergleichsmodus in den Performanzmodus eine Trennung im Programmcode dadurch erfolgt, dass jeder Ausführungseinheit eine Kennung zugeordnet ist und abhängig von der Kennung wenigstens zwei Ausführungseinheiten unterschiedlicher Programmcode zugeordnet wird.

5 Vorteilhafterweise verwendet man ein Verfahren, bei dem die Kennung jeweils in einem Speicher, insbesondere einem Register, einer Ausführungseinheit enthalten ist.

Vorteilhafterweise verwendet man ein Verfahren, bei dem die Kennung jeweils in einem Statusregister einer Ausführungseinheit enthalten ist

10

Vorteilhafterweise verwendet man ein Verfahren, bei dem die Kennung jeweils in einem Interrupt-Status-Register einer Ausführungseinheit enthalten ist

15 Vorteilhafterweise verwendet man ein Verfahren, bei dem die Kennung jeweils in einem Interrupt-Maskierungs-Register einer Ausführungseinheit enthalten ist

Vorteilhafterweise verwendet man ein Verfahren, bei dem abhängig von der Kennung für jede Ausführungseinheit auf eine vorgebbare Adresse verwiesen wird, wobei die Ausführungseinheit den bei dieser Adresse beginnenden Programmcode abarbeitet.

20

Vorteilhafterweise verwendet man ein Verfahren, bei dem der jeweilige Verweis auf eine vorgebbare Adresse durch einen Adresszeiger erfolgt.

25 Vorteilhafterweise verwendet man ein Verfahren, bei dem der jeweilige Verweis auf eine vorgebbare Adresse durch einen Sprungbefehl erfolgt.

Vorteilhafterweise verwendet man ein Verfahren, bei dem jede Kennung gegen wenigstens eine vorgegebene Kennung verglichen wird und bei Gleichheit ein vorgegebener Programmcode zugeordnet wird.

30

Vorteilhafterweise verwendet man ein Verfahren, bei dem jede Kennung gegen wenigstens eine vorgegebene Kennung verglichen wird und bei Ungleichheit aller Kennungen ein

- 4 -

Ereignis erkannt wird, das zu einem undefinierten Zustand führen würde.

Vorteilhafterweise verwendet man eine Vorrichtung zur Trennung der Abarbeitung von
Programmcodes bei einem Rechnersystem mit wenigstens zwei Ausführungseinheiten, wobei
5 zwischen wenigstens zwei Betriebsmodi umgeschaltet wird und ein erster Betriebsmodus
einem Vergleichsmodus und ein zweiter Betriebsmodus einem Performanzmodus entspricht
und die wenigstens zwei Ausführungseinheiten im Vergleichsmodus den gleichen
Programmcodes abarbeiten dadurch gekennzeichnet, dass die Vorrichtung derart ausgebildet
ist, dass bei der Umschaltung vom Vergleichsmodus in den Performanzmodus eine Trennung
10 im Programmcodes dadurch erfolgt, dass jeder Ausführungseinheit eine Kennung zugeordnet
ist und abhängig von der Kennung mindestens zwei Ausführungseinheiten unterschiedlicher
Programmcodes zugeordnet wird.

Vorteilhafterweise verwendet man eine Vorrichtung, die einen Speicher, insbesondere ein
15 Register enthält, in dem die Kennung jeder Ausführungseinheit abgespeichert ist.

Vorteilhafterweise verwendet man eine Vorrichtung, bei der jede Ausführungseinheit ein
Statusregister enthält, in dem jeweils die Kennung der Ausführungseinheit abgespeichert ist.

Vorteilhafterweise verwendet man eine Vorrichtung, die wenigstens ein Interrupt-Status-
20 Register enthält, in dem die Kennungen der Ausführungseinheiten abgespeichert sind.

Vorteilhafterweise verwendet man eine Vorrichtung, die wenigstens ein Interrupt-
Maskierungs-Register enthält, in dem die Kennungen der Ausführungseinheiten
25 abgespeichert sind.

Vorteilhafterweise verwendet man eine Ausführungseinheit eines Rechnersystems mit einer
Vorrichtung zur Trennung der Abarbeitung von Programmcodes bei einem Rechnersystem mit
wenigstens zwei Ausführungseinheiten, wobei zwischen wenigstens zwei Betriebsmodi
30 umgeschaltet wird und ein erster Betriebsmodus einem Vergleichsmodus und ein zweiter
Betriebsmodus einem Performanzmodus entspricht und die wenigstens zwei
Ausführungseinheiten im Vergleichsmodus den gleichen Programmcodes abarbeiten dadurch
gekennzeichnet, dass die Vorrichtung derart ausgebildet ist, dass bei der Umschaltung vom

Vergleichsmodus in den Performanzmodus eine Trennung im Programmcode dadurch erfolgt, dass jeder Ausführungseinheit eine Kennung zugeordnet ist und abhängig von der Kennung mindestens zwei Ausführungseinheiten unterschiedlicher Programmcode zugeordnet wird.

5 Vorteilhafterweise verwendet man ein Rechnersystem mit zwei Ausführungseinheiten und einer Vorrichtung zur Trennung der Abarbeitung von Programmcode bei einem Rechnersystem mit wenigstens zwei Ausführungseinheiten, wobei zwischen wenigstens zwei Betriebsmodi umgeschaltet wird und ein erster Betriebsmodus einem Vergleichsmodus und ein zweiter Betriebsmodus einem Performanzmodus entspricht und die wenigstens zwei
10 Ausführungseinheiten im Vergleichsmodus den gleichen Programmcode abarbeiten dadurch gekennzeichnet, dass die Vorrichtung derart ausgebildet ist, dass bei der Umschaltung vom Vergleichsmodus in den Performanzmodus eine Trennung im Programmcode dadurch erfolgt, dass jeder Ausführungseinheit eine Kennung zugeordnet ist und abhängig von der Kennung mindestens zwei Ausführungseinheiten unterschiedlicher Programmcode zugeordnet wird.

15 Weitere Vorteile und vorteilhafte Ausgestaltungen ergeben sich aus den Merkmalen der Ansprüche sowie der Beschreibung.

Figuren

20 In der Figur 1 ist ein Multiprozessorsystem G60 mit zwei Ausführungseinheiten G10a, G10b einer Vergleichseinheit G20, einer Umschalteinheit G50 und einer Einheit zur Umschaltwunscherkennung G40 dargestellt.

25 In Figur 2 ist ein Multiprozessorsystem G60 mit zwei Ausführungseinheiten G10a, G10b einer kombinierten Vergleichs- und Umschalteinheit G70 bestehend aus einer Vergleichseinheit G20 und einer Umschalteinheit G50 sowie einer Einheit zur Umschaltwunscherkennung G40 dargestellt.

30 In Figur 3 ist ein Multiprozessorsystem G60 mit zwei Ausführungseinheiten G10a, G10b einer kombinierten Umschaltwunscherkennung, Vergleichs und Umschalteinheit G80 bestehend aus einer Vergleichseinheit G20 und einer Umschalteinheit G50 und einer Einheit zur Umschaltwunscherkennung G40, dargestellt.

In Figur 4 ist ein Multiprozessorsystem G200 mit zwei Ausführungseinheiten G210a, G210b einer Umschalt und Vergleichseinheit G260 dargestellt.

5 In der Figur 5 ist in einem Flussdiagramm ein Verfahren dargestellt, das innerhalb einer speziellen Pipelinestufe G230a, G230b eine spezielle undefinierte Bitkombination mit einer NOP oder anderen neutralen Bitkombination austauscht.

10 In Figur 6 ist ein Multiprozessorsystem H200 mit zwei Ausführungseinheiten H210a, H210b und einer Umschalt und Vergleichseinheit H260 dargestellt.

15 In der Figur 7 ist in einem Flussdiagramm ein Verfahren dargestellt das zeigt, wie mit Hilfe der Einheiten-ID der Programmfluss beim Wechsel von einem Vergleichsmodus in einen Performanzmodus in einem Multiprozessorsystem mit 2 Ausführungseinheiten getrennt werden kann.

20 In Figur 8 ist ein mögliches Verfahren dargestellt, wie mit Hilfe der Einheiten-ID der Programmfluss beim Wechsel von einem Vergleichsmodus in einen Performanzmodus in einem Multiprozessorsystem mit 3 Ausführungseinheiten getrennt werden kann.

In Figur 9 ist in einem Flussdiagramm ein Verfahren dargestellt, das die Ausführungseinheiten beim Umschalten vom Performanzmodus in den Vergleichsmodus synchronisiert.

25 In Figur 10 ist ein Zustandsautomat dargestellt, der die Umschaltung zwischen einem Performanz und einem Vergleichsmodus darstellt.

30 In Figur 11 ist ein Multiprozessorsystem G400 mit zwei Ausführungseinheiten sowie zwei Interruptcontrollern G420a, G420b inklusive darin enthaltenen Interrupt Maskierungsregistern G430a, G430b und verschiedenen Interruptquellen G440a bis G440n dargestellt.

In Figur 12 wird ein Multiprozessorsystem mit zwei Ausführungseinheiten, einer Umschalt- und Vergleichseinheit und einem Interruptcontroller mit drei Registersätzen dargestellt.

- 7 -

In Figur 13 ist die einfachste Form eines Vergleichers dargestellt.

Figur 14 zeigt einen Vergleich mit einer Einheit um einen Phasenversatz zu auszugleichen.

5 In Figur 15 ist das prinzipielle Verhalten der bevorzugten Komponente M700 (Umschalt- und Vergleichseinheit) im Vergleichsmodus beschrieben.

In Figur 16 ist das prinzipielle Verhalten der bevorzugten Komponente M700 (Umschalt- und Vergleichseinheit) im Performanzmodus beschrieben.

10

In Figur 17 ist eine Ausführungsform der Umschalt und Vergleichseinheit dargestellt.

In Figur 18 ist eine weitere Ausführungsform der Umschalt und Vergleichseinheit dargestellt.

15

In Figur 19 ist eine Umschalt und Vergleichseinheit, die ein Modus Signal erzeugt, dargestellt.

In Figur 20 ist eine allgemeine Darstellung einer Umschalt- und Vergleichseinheit dargestellt.

20

Figur 21 zeigt eine allgemeine Darstellung einer Umschalt- und Vergleichseinheit, die ein allgemeines Modus und ein allgemeines Fehlersignal erzeugt.

In Figur 22 ist die Frage Antwort Kommunikation mit einer externen Einheit dargestellt.

25

In Figur 23 ist die Kommunikation mit einem intelligenten Aktor dargestellt.

Beschreibung der Ausführungsbeispiele

30

Als Ausführungseinheit kann im Folgenden sowohl ein Prozessor, ein Core, eine CPU, als auch eine FPU (Floating Point Unit), ein DSP (Digitaler Signalprozessor), ein Coprozessor oder eine ALU (Arithmetic logical Unit) bezeichnet werden.

- 8 -

In der Figur 1 ist ein Multiprozessorsystem G60 mit zwei Ausführungseinheiten G10a, G10b einer Vergleichseinheit G20, einer Umschalteinheit G50 und einer Einheit zur Umschaltwunscherkennung G40 dargestellt.

5 Die Erfindung bezieht sich auf ein Multiprozessorsystem G60 dargestellt in Figur 1, Figur 2, Figur 3 mit wenigstens zwei Ausführungseinheiten G10a, G10b, einer Vergleichseinheit G20, einer Umschalteinheit G50 und einer Einheit zur Umschaltwunscherkennung G40. Die Umschalteinheit G50 hat wenigstens zwei Ausgänge zu wenigstens zwei Systemschnittstellen G30a, G30b. Über diese Schnittstellen können Register, Speicher oder Peripherals wie Digitale
10 Ausgänge, D/A-Wandler, Kommunikationscontroller angesteuert werden. Dieses Multiprozessorsystem kann in wenigstens zwei Betriebsmodi betrieben werden, einem Vergleichsmodus (VM) und einem Performanzmodus (PM).

15 Im Performanzmodus werden in den unterschiedlichen Ausführungseinheiten unterschiedliche Befehle, Programmsegmente oder Programme parallel ausgeführt. In diesem Betriebsmodus ist die Vergleichseinheit G20 deaktiviert. Die Umschalteinheit G50 ist in diesem Betriebsmodus so konfiguriert, dass jede Ausführungseinheit G10a, G10b mit einer Systemschnittstelle G30a, G30b verbunden ist. Dabei ist die Ausführungseinheit G10a mit der Systemschnittstelle G30a und die Ausführungseinheit G10b mit der Systemschnittstelle G30b verbunden.

20 Im Vergleichsmodus werden in beiden Ausführungseinheiten G10a, G10b gleiche oder gleichartige Befehle, Programmsegmente oder Programme abgearbeitet. Günstigerweise werden diese Befehle takt synchron abgearbeitet, es ist aber auch eine Abarbeitung mit Asynchronität oder einem definiertem Taktversatz denkbar. Die Ausgangssignale der Ausführungseinheiten
25 G10a, G10b werden in der Vergleichseinheit G20 verglichen. Bei einem Unterschied wird auf einen Fehler erkannt und es können entsprechende Maßnahmen ergriffen werden. Diese Maßnahmen können ein Fehlersignal auslösen, eine Fehlerbehandlung einleiten, Schalter betätigen oder eine Kombination von diesen und weiteren denkbaren Maßnahmen sein. Die Umschalteinheit G50 ist in einer Variation so konfiguriert, dass nur ein Signal zu den
30 Systemschnittstellen G30a, G30b verbunden ist. In einer anderen Konfiguration bewirkt die Umschalteinheit nur, dass die verglichenen und damit gleichen Signale an die Systemschnittstellen G30a, G30b verbunden werden.

Die Umschaltwunscherkennung G40 detektiert unabhängig vom gerade aktiven Modus einen Umschaltwunsch in einen anderen Modus.

5 In Figur 2 ist ein Multiprozessorsystem G60 mit zwei Ausführungseinheiten G10a, G10b einer kombinierten Vergleichs- und Umschaltseinheit G70 bestehend aus einer Vergleichseinheit G20 und einer Umschaltseinheit G50 sowie einer Einheit zur Umschaltwunscherkennung G40 dargestellt.

10 In einer Ausführungsform des oben beschriebenen Sachverhaltes, kann die Umschaltseinheit G50 und die Vergleichseinheit G20 zu einer gemeinsamen Umschalt- und Vergleichseinheit (UVE) G70 zusammengefasst sein, wie in Figur 2 dargestellt. Diese gemeinsame Komponente G70 übernimmt dann die Aufgaben der Einzelkomponenten G50, G20. In Figur 15, Figur 16, Figur 17, Figur 18 und Figur 19 sind Ausführungsvarianten der UVE G70 dargestellt.

15 In einer weiteren Ausführungsform, wie in Figur 3 dargestellt, kann die Einheit zur Umschaltwunscherkennung G40, der Vergleichs G20 und die Umschaltseinheit G50 in einer gemeinsamen Komponente G80 zusammengefasst sein. In einer weiteren Ausführungsform, die nicht in einer Figur dargestellt ist, kann die Umschaltwunscherkennung G40 und der Vergleichs G20 in einer gemeinsamen Komponente zusammengefasst sein. Ebenfalls denkbar
20 ist eine Zusammenfassung Umschaltwunscherkennung G40 mit dem Umschalter G50 in einer gemeinsamen Komponente.

Im weiteren Text wird, falls nicht anders angegeben, davon ausgegangen, dass eine
25 Umschaltwunscherkennung G40 und eine kombinierte Umschalt und Vergleichseinheit G70, vorliegen.

Ein allgemeiner Fall der Umschalt- und Vergleichskomponente, auch für den Verwendung für
mehr als zwei Ausführungseinheiten, ist in Figur 20 gezeigt. Von den n zu berücksichtigenden
Ausführungseinheiten gehen n Signale N140,..., N14n an die Umschalt- und
30 Vergleichskomponente N100. Diese kann bis zu n Ausgangssignale N160,..., N16n aus diesen
Eingangssignalen erzeugen. Im einfachsten Fall, dem „reinen Performanzmodus“, werden alle
Signale N14i auf die entsprechenden Ausgangssignale N16i geleitet. Im entgegen gesetzten

- 10 -

Grenzfall, dem „reinen Vergleichsmodus“ werden alle Signale N140,..., N14n nur auf genau eines der Ausgangssignale N16i geleitet.

An dieser Figur lässt sich darlegen, wie die verschiedenen denkbaren Modi entstehen können. Dazu ist in dieser Figur die logische Komponente einer Schaltlogik N110 enthalten. Diese Komponente muss nicht als eigene Komponente vorhanden sein. Entscheidend ist, dass die beschriebenen Funktionen im System realisiert sind. Die Schaltlogik N110 legt zunächst fest, wie viele Ausgangssignale es überhaupt gibt. Weiter legt sie fest, welche der Eingangssignale zu welchem der Ausgangssignale beitragen. Dabei kann ein Eingangssignal zu genau einem Ausgangssignal beitragen. In mathematischer Form anders formuliert ist also durch die Schaltlogik eine Funktion definiert, die jedem Element der Menge {N140,..., N14n} ein Element der Menge {N160,..., N16n} zuordnet.

Die Verarbeitungslogik N120 legt dann zu jedem der Ausgänge N16i fest, in welcher Form die Eingänge zu diesem Ausgangssignal beitragen. Auch diese Komponente muss nicht als eigene Komponente vorhanden sein. Entscheidend ist wieder, dass die beschriebenen Funktionen im System realisiert sind. Um beispielhaft die verschiedenen Variationsmöglichkeiten zu beschreiben, sei ohne Beschränkung der Allgemeinheit angenommen, dass der Ausgang N160 durch die Signale N141, ..., N14m erzeugt wird. Falls $m = 1$ entspricht dies einfach einer Durchschaltung des Signals, falls $m = 2$ dann werden die Signale N141, N142 verglichen, wie beispielsweise im Vergleich in Figur 13, Figur 14 beschrieben. Dieser Vergleich kann synchron oder asynchron durchgeführt werden, er kann bitweise oder nur auf signifikante Bits oder auch mit einem Toleranzband durchgeführt werden.

Falls $m \geq 3$ gibt es mehrere Möglichkeiten.

Eine erste Möglichkeit besteht darin alle Signale zu vergleichen und bei Vorhandensein mindestens zweier verschiedener Werte einen Fehler zu detektieren, den man optional signalisieren kann.

Eine zweite Möglichkeit besteht darin, dass man eine k aus m –Auswahl vornimmt ($k > m/2$).

Diese kann durch Verwendung von Vergleichern realisiert werden. Optional kann ein Fehlersignal generiert werden, wenn eines der Signale als abweichend erkannt wird. Ein möglicherweise davon verschiedenes Fehlersignal kann generiert werden, wenn alle drei Signale verschieden sind.

Eine dritte Möglichkeit besteht darin, diese Werte einem Algorithmus zuzuführen. Dies kann beispielsweise die Bildung eines Mittelwerts, eines Medianwert, oder die Verwendung eines fehlertoleranten Algorithmus (FTA) darstellen. Ein solcher FTA beruht darauf, Extremwerte der Eingangswerte weg zu streichen und eine Art der Mittelung über die restlichen Werte vorzunehmen. Diese Mittelung kann über die gesamte Menge der restlichen Werte oder vorzugsweise über eine in HW leicht zu bildenden Teilmenge vorgenommen werden. In diesem Fall ist es nicht immer notwendig, die Werte tatsächlich zu vergleichen. Bei der Mittelwertbildung muss beispielsweise nur addiert und dividiert werden, FTM, FTA oder Median erfordern eine teilweise Sortierung. Gegebenenfalls kann auch hier bei hinreichend großen Extremwerten optional ein Fehlersignal ausgegeben werden.

Diese verschiedenen genannten Möglichkeiten der Verarbeitung mehrerer Signale zu einem Signal werden der Kürze wegen als Vergleichsoperationen bezeichnet.

Die Aufgabe der Verarbeitungslogik ist es also, die genaue Gestalt der Vergleichsoperation für jedes Ausgangssignal – und damit auch für die zugehörigen Eingangssignale – festzulegen. Die Kombination der Information der Schaltlogik N110 (d.h. die o.g. Funktion) und der Verarbeitungslogik (d.h. die Festlegung der Vergleichsoperation pro Ausgangssignal, d.h. pro Funktionswert) ist die Modusinformatio und diese legt den Modus fest. Diese Information ist im allgemeinen Fall natürlich mehrwertig, d.h. nicht nur über ein logisches Bit darstellbar. Nicht alle theoretisch denkbaren Modi sind in einer gegebenen Implementierung sinnvoll, man wird vorzugsweise die Zahl der erlaubten Modi einschränken. Zu betonen ist, dass im Fall von nur zwei Ausführungseinheiten, wo es nur einen Vergleichsmodus gibt, die gesamte Information auf nur ein logisches Bit kondensiert werden kann.

Eine Umschaltung von einem Performanz- in einen Vergleichsmodus ist im allgemeinen Fall dadurch charakterisiert, dass Ausführungseinheiten, die im Performanzmodus auf verschiedene Ausgänge hin abgebildet werden, im Vergleichsmodus auf den gleichen Ausgang hin abgebildet werden. Vorzugsweise ist dies dadurch realisiert, dass es ein Teilsystem von Ausführungseinheiten gibt, bei dem im Performanzmodus alle Eingangssignale N14i, die im Teilsystem zu berücksichtigen sind, direkt auf korrespondierende Ausgangssignale N16i geschaltet werden, während sie im Vergleichsmodus alle auf einen Ausgang hin abgebildet sind. Alternativ kann eine solche Umschaltung auch dadurch realisiert werden, dass Paarungen geändert werden. Es ist dadurch erläutert, dass man im allgemeinen Fall nicht von dem einen Performanzmodus und dem einen Vergleichsmodus sprechen kann, obwohl man in einer

gegebenen Ausprägung der Erfindung die Menge der erlaubten Modi so einschränken kann, dass dies der Fall ist. Man kann aber immer von einer Umschaltung von einem Performanz- in einen Vergleichsmodus (und umgekehrt) sprechen.

Zwischen diesen Modi kann, über Software gesteuert, dynamisch im Betrieb umgeschaltet werden. Ausgelöst wird die Umschaltung dabei entweder über die Ausführung von speziellen Umschaltinstruktionen, speziellen Instruktionssequenzen, explizit gekennzeichneten Instruktionen oder durch den Zugriff auf bestimmte Adressen durch wenigstens eine der Ausführungseinheiten des Multiprozessorsystems.

Die Fehlerschaltungslogik N130 sammelt die Fehlersignale, die beispielsweise von den Vergleichen generiert werden, und kann optional die Ausgänge N16i passiv schalten, indem sie diese beispielsweise über einen Schalter unterbricht.

Die folgenden Beispiele konzentrieren sich aber zumeist auf den Fall von zwei Ausführungseinheiten, an dem sich die meisten Konzepte einfacher darstellen lassen

Die Umschaltung zwischen den Modi kann durch verschiedene Verfahren kodiert werden. In einem möglichen Verfahren sind spezielle Umschaltbefehle zu verwenden, die von der Einheit zur Umschaltwunscherkennung G40 detektiert werden. Ein weiteres mögliches Verfahren zur Kodierung der Umschaltung ist durch den Zugriff auf einen speziellen Speicherbereich definiert den wieder die Einheit zur Umschaltwunscherkennung G40 detektiert. Ein weiteres Verfahren wertet in der Einheit zur Umschaltwunscherkennung G40 ein externes Signal, das eine Umschaltung signalisiert, aus. Im Folgenden wird ein Verfahren beschrieben, das nicht benutzte Bitkombinationen im bestehenden Befehlssatz des Prozessors verwendet. Besonderer Vorteil dieses Verfahrens ist es, dass bestehende Entwicklungsumgebungen (Assembler, Compiler, Linker, Debugger) weiter verwendet werden kann.

In Figur 4 ist ein Multiprozessorsystem G200 mit zwei Ausführungseinheiten G210a, G210b und einer Umschalt und Vergleichseinheit G260 dargestellt. Zur Umschaltung zwischen einem Vergleichsmodus und einem Performanzmodus (und umgekehrt) werden im Assembler nicht definierte Bitkombinationen der mindestens zwei Ausführungseinheiten G210a, G210b benutzt. Als nicht definierte oder undefinierte Bitkombinationen in diesem Sinne sind alle Bitkombinationen zu verstehen, die in der Beschreibung des Befehlssatz als undefiniert oder

illegal spezifiziert sind. Dies sind z.B. Illegal Operand, Illegal Instruktion, Illegal Operation. Allgemeines Kennzeichen dieser undefinierten Bitkombinationen ist es, dass eine normale Ausführungseinheit bei der Ausführung einer solchen Bitkombination entweder ein Fehlersignal erzeugt oder ein nicht definiertes Verhalten zeigt. Diese Bitkombinationen werden also nicht benötigt um die Semantik eines gewöhnlichen Programms darzustellen.

Für die Softwareentwicklung kann damit die bisherige Entwicklungsumgebung benutzt werden, wie sie für Einprozessorsysteme vorliegt. Dies kann beispielsweise dadurch realisiert werden, dass man ein Makro „SWITCH MODE TO PM“ und ein Makro „SWITCH MODE TO VM“ definiert, das an geeigneter Stelle im Code entsprechende im oben definierten Sinn undefinierte Bitkombinationen einfügt.

Die Verwendung dieser Kombination wird dann als allgemeines „SWITCH“ Makro definiert. Dieses bewirkt dann einen Wechsel abhängig vom derzeitigen Modus in den jeweils anderen. Falls im System mehr als zwei unterschiedliche Modi vorliegen, müssen zur Verwendung dieses Verfahrens mehr solcher Kombinationen vorliegen, vorzugsweise kann dann eine pro Modus zur Umschaltkennung verwendet werden.

Erfindungsgemäß wird der Umschaltwunsch durch eine im Befehlssatz nicht definierte Bitkombination kodiert. Diese dürfen innerhalb einer Ausführungseinheit G210a, G210b nicht in der üblichen Weise verarbeitet werden. Aus diesem Grund wird eine zusätzliche Pipelinestufe (REPLACE Stufe) G230a, G230b vorgeschlagen, die die entsprechenden Bitkombinationen erkennt und diese zur weiteren Verarbeitung durch neutrale Bitkombinationen ersetzt. Günstigerweise wird dazu der „NOP“ (No Operation) Befehl benutzt. Ein NOP Befehl zeichnet sich dadurch aus, dass er den internen Zustand der Ausführungseinheit, bis auf den Instruktionszeiger nicht ändert. Die REPLACE Stufe G230a, G230b wird dabei nach der üblicherweise ersten Stufe, der FETCH Stufe G220a G220b und vor den restlichen Pipelinestufen G240a, G240b, werden im Assembler nicht definierte Bitkombinationen, die hier in einer Einheit zusammengefasst sind, eingefügt.

Erfindungsgemäß wird die hier dargestellte Implementierung einer Einheit zur Umschaltwunscherkennung G40 als spezielle Pipelinestufe G230a, G230b in einer Pipelineeinheit G215a, G215b, ein zusätzliche Signale G250a, G250b erzeugen, wenn eine

- 14 -

entsprechende Bitkombination zur Umschaltung erkannt wurde, das einer separaten Umschaltseinheit und Vergleichseinheit G260 signalisiert, dass ein Wechsel des Verarbeitungsmodus durchzuführen ist.

5 Die REP Stufen G230a, G230b sind vorzugsweise zwischen den FET G220a, G220b und den restlichen Pipelinestufen G240a, G240b in den Pipelineeinheiten G215a, G215b der Ausführungseinheiten G210a, G210b angeordnet. Die REP Stufen G230a, G230b erkennen dabei die entsprechenden Bitkombinationen und leiten in diesem Fall NOP Befehle an die übrigen Stufen G240a, G240b weiter. Gleichzeitig wird das jeweilige Signal G250a oder
10 G250b aktiviert. In allen anderen Fällen verhalten sich die REP Stufen G230a, G230b neutral, d.h. alle anderen Befehle werden unverändert an die übrigen Stufen G240a, G240b weitergereicht.

In der Figur 5 ist in einem Flussdiagramm ein Verfahren dargestellt, das innerhalb einer
15 speziellen Pipelinestufe G230a, G230b eine spezielle undefinierte Bitkombination mit einer NOP oder anderen neutralen Bitkombination austauscht. In der FETCH Stufe G300 wird ein Befehl d.h. eine Bitkombination aus dem Speicher geholt. Danach wird im Block G310 unterschieden ob die geholte Bitkombination der speziellen undefinierten Bitkombination, die eine Umschaltung kodiert, entspricht. Wenn dies nicht der Fall ist, wird im nächsten Schritt
20 G320 die Bitkombination ohne Veränderung an die übrigen Pipelinestufen G340 zur weiteren Verarbeitung übergeben. Wenn die spezielle Bitkombination, die eine Umschaltung kodiert, im Schritt G310 erkannt wurde, wird im Schritt G330 diese durch die NOP Bitkombination ersetzt und diese dann zur weiteren Verarbeitung an die weiteren Pipelinestufen G340 übergeben. In einer vorteilhaften Ausführungsform stellen die Blöcke G310, G320, G330 die Funktionalität
25 einer erfindungsgemäßen REPLACE Stufe G230a, G230b dar, wobei diese auch weitere Funktionalität enthalten können.

In Figur 6 ist ein Multiprozessorsystem H200 mit zwei Ausführungseinheiten H210a, H210b und einer Umschalt und Vergleichseinheit H260 dargestellt. Die Komponenten H220a, H220b,
30 H240a, H240b haben die gleich Bedeutung wie G220a, G220b, G240a, G240b. In einer alternativen Ausführung der Einheit zur Umschaltwunscherkennung G40, hier beschrieben durch die speziellen Pipelinestufen H230a, H230b, besitzt diese neben den Signalen H250a, H250b, die eine Umschaltung signalisieren, weitere Signale. Damit die Ausführungseinheiten

- 15 -

H210a, H210b beim Wechsel vom Performanzmodus in den Vergleichsmodus synchronisiert werden können, besitzen die Pipelineeinheiten H215a, H215b der Ausführungseinheiten H210a, H210b jeweils einen Signaleingang H280a, H280b, mit dem die Verarbeitung gestoppt werden kann. Dieses Signal wird von der Umschalt- und Vergleichseinheit H260 für diejenige
5 Pipelineeinheit H215a oder H215b gesetzt, die zuerst einen Umschaltbefehl erkannt hat und damit das Signal H250a bzw. GH50b aktiviert hat. Erst wenn beide Pipelineeinheiten H215a, H215b der Ausführungseinheiten H210a, H210b den Umschaltbefehl erkannt haben und durch Software oder weitere Hardwaremaßnahmen ihre internen Zustände synchronisiert haben, wird dieses Signal H280a, H280b wieder zurückgenommen. Beim Wechsel vom Vergleichsmodus in
10 den Performanzmodus werden die H280a, H280b nicht benötigt, da keine Synchronisation notwendig ist.

Vorraussetzung für den hier beschriebenen Vorschlag ist eine Einheit (als ID-Einheit bezeichnet) oder Verfahren, über den jede Ausführungseinheit ihre individuelle Nummer oder
15 Einheiten-ID ermitteln kann. Bei einem System mit zwei Ausführungseinheiten kann beispielsweise eine Ausführungseinheit für sich die Nummer 0, die andere die Nummer 1 ermitteln. Bei einem System mit mehr als 2 Ausführungseinheiten werden die Nummern entsprechend vergeben bzw. ermittelt. Diese ID unterscheidet nicht zwischen einem Vergleichsmodus und einem Performanzmodus sondern bezeichnet eine Ausführungseinheit
20 eindeutig. Die ID-Einheit kann in den jeweiligen Ausführungseinheiten enthalten sein, zum Beispiel implementiert als Bit oder Bitkombination im Prozessorstatusregister oder als eigenes Register oder als einzelnes Bit oder als zu den Ausführungseinheiten externe Einheit die auf Anfrage eine entsprechende ID liefert.

Nachdem die Ausführungseinheiten gemäß einem Umschaltwunsch die Umschaltung in den Performanzmodus ausgeführt haben, ist zwar die Vergleichseinheit nicht mehr aktiv, aber die Ausführungseinheiten führen immer noch dieselben Befehle aus. Das liegt daran, dass die Instruktionszeiger, die die Stelle im Programm kennzeichnen, an denen eine Ausführungsarbeit
25 im nächsten Schritt arbeitet bzw. zur Zeit arbeitet, von der Umschaltung nicht beeinflusst werden. Damit die Ausführungseinheiten anschließend verschiedene SW-Module ausführen können, muss der Programmablauf der Ausführungseinheiten getrennt werden. Der Sache entsprechend haben deswegen die Instruktionszeiger im Performanzmodus in der Regel
30 verschiedene Werte, da ja erfindungsgemäß unabhängige Befehle, Programmsegmente oder

Programme verarbeitet werden. Die Trennung der Programmflüsse geschieht im hier beschriebenen Vorschlag durch das Ermitteln der jeweiligen Ausführungseinheiten-Nummer. Je nachdem welche ID eine Ausführungseinheit besitzt, führt die Ausführungseinheit ein bestimmtes Softwaremodul aus. Da jede Ausführungseinheit eine individuelle Nummer oder ID besitzt, kann hierdurch der Programmfluss der beteiligten Ausführungseinheiten zuverlässig getrennt werden.

In der Figur 7 ist in einem Flussdiagramm ein Verfahren dargestellt das zeigt, wie mit Hilfe der Einheiten-ID der Programmfluss beim Wechsel von einem Vergleichsmodus in einen Performanzmodus in einem Multiprozessorsystem mit 2 Ausführungseinheiten getrennt werden kann. Nach dem Ausführen der Umschaltung von einem Vergleichs- in einen Performanzmodus G500 erfolgt eine Abfrage der Einheiten ID oder Ausführungseinheiten-Nummer G510 durch beide Ausführungseinheiten. Erfindungsgemäß wird dabei die Ausführungseinheit 0 die Ausführungseinheiten-Nummer 0 erhalten, die Ausführungseinheit 1 die Ausführungseinheiten-Nummer 1. In G510 erfolgt ein Vergleich der ermittelten Ausführungseinheiten-Nummer mit der Nummer 0. Falls diese gleich sind, fährt im Schritt G520 die Ausführungseinheit, für die dieser Vergleich erfolgreich war, mit dem Code für Ausführungseinheit 0 fort. Die Ausführungseinheit, für die dieser Vergleich nicht erfolgreich war, fährt in G530 mit dem Vergleich mit der Nummer 1 fort. Falls dieser Vergleich erfolgreich ist, wird mit dem Code für Ausführungseinheit 1 in G540 fortgefahren. Falls dieser Vergleich nicht erfolgreich ist, wurde für die entsprechende Ausführungseinheit damit eine Ausführungseinheiten-Nummer ungleich 0 und 1 ermittelt. Dieses stellt einen Fehlerfall dar und es wird mit G550 fortgefahren.

In Figur 8 ist ein mögliches Verfahren für 3 Ausführungseinheiten beschreiben. Nach dem Ausführen der Umschaltung von einem Vergleichs- in einen Performanzmodus H500 erfolgt eine Abfrage der Einheiten ID oder Ausführungseinheiten-Nummer H510 durch die Ausführungseinheiten. Erfindungsgemäß wird dabei zum Beispiel die Ausführungseinheit 0 die Ausführungseinheiten-Nummer 0 erhalten, die Ausführungseinheit 1 die Ausführungseinheiten-Nummer 1 und Ausführungseinheit 2 die Ausführungseinheiten-Nummer 2. In H510 erfolgt ein Vergleich der ermittelten Ausführungseinheiten-Nummer mit der Nummer 0. Falls diese gleich sind, fährt im Schritt H520 die Ausführungseinheit, für die dieser Vergleich erfolgreich war, mit dem Code für Ausführungseinheit 0 fort. Die

- 17 -

Ausführungseinheiten, für die dieser Vergleich nicht erfolgreich war, fahren mit dem Vergleich mit der Nummer 1 in H530 fort. In der Ausführungseinheit, für die dieser Vergleich erfolgreich ist, wird mit dem Code für Ausführungseinheit 1 in H540 fort gefahren. Die Ausführungseinheiten, für die dieser Vergleich nicht erfolgreich war, fahren mit dem Vergleich mit der Nummer 2 in H535 fort. Die Ausführungseinheit, für die dieser Vergleich erfolgreich ist, wird mit dem Code für Ausführungseinheit 2 in H536 fort gefahren. Falls dieser Vergleich nicht erfolgreich war, wurde für die entsprechende Ausführungseinheit damit eine Ausführungseinheiten-Nummer ungleich 0,1 und 2 ermittelt. Dieses stellt einen Fehlerfall dar und es wird mit H550 fortgefahren. Alternativ zu dem Vergleich mit einer Nummer kann die ermittelte Ausführungseinheiten-Nummer auch direkt als Index in eine Sprungtabelle verwendet werden.

Entsprechend dieser Beschreibung kann dieses Verfahren auch für Multiprozessorsysteme mit mehr als 3 Ausführungseinheiten angewendet werden.

Wenn vom Performanzmodus in den Vergleichsmodus geschaltet wird, dann müssen mehrere Dinge beachtet werden. Bei der Umschaltung vom Performanzmodus in den Vergleichsmodus muss sichergestellt werden, dass die internen Zustände der Ausführungseinheiten nach dem Umschalten gleichartig sind, ansonsten würde im Vergleichsmodus eventuell auf einen Fehler erkannt werden, falls die unterschiedlichen Startzustände zu unterschiedlichen Ausgaben führen. Dies kann per Hardware, per Software, per Firmware oder in einer Kombination aller drei durchgeführt werden. Voraussetzung hierfür ist, dass alle Ausführungseinheiten gleiche oder gleichartige Befehle, Programme oder Programmsegmente nach der Umschaltung in den Vergleichsmodus ausführen. Im weiteren ist ein Synchronisationsverfahren beschrieben, das anwendbar ist, wenn der Vergleichsmodus sich dadurch auszeichnet, dass identische Befehle verarbeitet werden und ein bitgenauer Vergleich stattfindet.

In Figur 9 ist in einem Flussdiagramm ein Verfahren dargestellt, das die Ausführungseinheiten beim Umschalten von einem Performanzmodus in einen Vergleichsmodus synchronisiert. Im Schritt G600 werden vorzugsweise alle Interrupts gesperrt. Dies ist nicht nur wichtig, weil die Interrupt-Controller für den Vergleichsmodus entsprechend umprogrammiert werden müssen. Auch durch Software soll der interne Zustand der Ausführungseinheiten angeglichen werden.

Wenn aber während der Vorbereitung zum Umschalten in den Vergleichsmodus ein Interrupt ausgelöst wird, dann ist eine Angleichung nicht mehr ohne weiteren Aufwand möglich.

5 Schritt G610: Besitzen die beiden Ausführungseinheiten separate Caches, so muss auch der Inhalt der Caches vor der Umschaltung angeglichen werden, um zu verhindern, dass im Vergleichsmodus für eine Adresse für die eine Ausführungseinheit ein Cache-Hit und für eine andere Ausführungseinheiten ein Cache-Miss auftritt. Wird dies durch die Cache-Hardware nicht selbstständig durchgeführt, ist dies zum Beispiel durch Markieren aller Cachelines als ungültig zu bewerkstelligen. Es muss solange gewartet werden, bis der Cache (oder die Caches) 10 komplett ungültig sind. Dies ist bei Bedarf durch eine Warteschleife im Programmcode sicherzustellen. Dies kann auch durch andere Mittel erreicht werden, entscheidend ist, dass nach diesem Schritt die Caches im gleichen Zustand sind.

15 Im Schritt G620 werden die Schreib-Puffer der Ausführungseinheiten geleert, damit nach der Umschaltung keine Aktivitäten der Ausführungseinheiten stattfinden, die noch aus dem Performanzmodus herrühren.

20 Im Schritt G630 wird der Zustand der Pipelinestufen der Ausführungseinheiten synchronisiert. Hierfür führt man beispielsweise eine geeignete Anzahl von NOP (No operation) Befehlen vor der Umschaltsequenz/Umschaltbefehl aus. Die Anzahl der NOP-Befehle richtet sich nach der Anzahl der Pipelinestufen und sind damit abhängig von der jeweiligen Architektur. Welcher Befehl sich als NOP-Befehl eignet, ist ebenfalls architekturabhängig. Besitzen die Ausführungseinheiten einen Instruktionscache, so ist dabei sicherzustellen, dass diese Befehlssequenz an den Grenzen einer Cacheline ausgerichtet ist (Alignment). Da der 25 Instruktionscache vor der Ausführung dieser NOP's als ungültig markiert worden ist, müssen diese NOP's erst in den Cache geladen werden. Beginnt diese Befehlssequenz an einer Cachelinegrenze so ist der Datentransfer vom Speicher (z.B. RAM/ROM/Flash) zum Cache abgeschlossen, bevor der Befehl zum Umschalten erfolgt. Auch diese muss bei der Bestimmung der notwendigen Anzahl der NOP's berücksichtigt werden.

30 Im Schritt G640 wird der Befehlschritt zur Umschaltung in den Vergleichsmodus tatsächlich durchgeführt.

Im Schritt G650 wird der Inhalt der jeweiligen Registerfiles jeder Ausführungseinheit angeglichen. Hierfür sind die Register vor oder nach der Umschaltung mit identischen Inhalten zu laden. Wichtig ist hierbei, dass nach der Umschaltung der Inhalt eines Registers in den Ausführungseinheiten identisch ist, bevor der Registerinhalt nach extern transferiert und damit von der Vergleichseinheit verglichen wird.

Im Schritt G660 werden die Interrupt-Controller umprogrammiert, so dass ein externes Interruptsignal bei allen zusammengeschalteten Ausführungseinheiten den gleichen Interrupt auslöst.

Im Schritt G670 werden die Interrupts wieder freigegeben.

Ist es vom Programmablauf nicht eindeutig, wann in den Vergleichsmodus umgeschaltet werden soll, so müssen die beteiligten Ausführungseinheiten über die beabsichtigte Umschaltung informiert werden. Hierfür wird vorzugsweise in den zu den jeweiligen Ausführungseinheiten gehörenden Interrupt-Controller z.B. per SW ein Interrupt initiiert. Die Interruptbehandlung veranlasst dann die Ausführung der oben beschriebenen Sequenz zur Zusammenschaltung.

In Figur 10 ist ein Zustandsautomat dargestellt, der die Umschaltung zwischen einem Performanz und einem Vergleichsmodus (und umgekehrt) darstellt. Beim Start des Systems, verursacht durch „Power On“ oder auch Reset (Software oder Hardware), wird das System über den Übergang G800 in den Zustand G700 versetzt. Allgemein gilt, dass das System nach einem undefinierten Ereignis, das in der Lage ist, einen Reset auszulösen, immer im Zustand G700 anfängt zu arbeiten. Beispielhafte Ereignisse, die einen Reset auslösen können, sind externe Signale, Probleme in der Spannungsversorgung oder interne Fehlerereignisse, die ein Weiterarbeiten nicht mehr sinnvoll machen. Der Zustand G700 der Umschalt- und Vergleichseinheit G70 und auch des Multiprozessorsystems G60, in dem im Performanzmodus gearbeitet wird, ist damit der Default Zustand des Systems. In allen Fällen in den ein sonst undefinierter Zustand eingenommen würde, wird der Default Zustand G700 eingenommen. Diese Default Stellung des Zustand G700 wird dabei durch Hardwaremaßnahmen sichergestellt. Beispielsweise kann der Systemzustand bzw. der Zustand der Umschalt- und Vergleichseinheit

G60 in einem Register, in einem Bit einem Register, durch eine Bitkombination in einem Register oder durch ein Flip-Flop, kodiert sein.

Per Hardware wird dann sichergestellt, dass nach einem Reset oder Power on immer der Zustand G700 eingenommen wird. Dies wird dadurch sichergestellt, dass z.B. das Reset Signal bzw. das „Power On“ Signal auf den Reset Eingang oder den Set Eingang des Flip-Flops oder des Registers geführt wird.

Im Zustand G700 arbeitet das System in einem Performanzmodus. Die Ausführungseinheiten G10a, G10b arbeiten damit unterschiedliche Befehle, Programme oder Programmstücke ab.

Ein Umschaltwunsch kann beispielsweise dadurch erkannt werden, dass eine Ausführungseinheit G10a, G10b einen speziellen Umschaltbefehl ausführt. Andere Möglichkeiten sind eine Erkennung durch den Zugriff auf eine spezielle Speicheradresse, durch ein internes Signal oder auch durch ein externes Signal. Solange kein Umschaltwunsch vorliegt, verbleibt das Multiprozessorsystem G60 und damit auch die Umschalt und Vergleichseinheit G70 im Zustand G700. Im Weiteren wird mit dem Umschaltwunsch das Erkennen einer Umschaltbedingung bezeichnet, die so gekennzeichnet ist wie ein Umschaltwunsch in diesem speziellen System gekennzeichnet wird.

Das Verbleiben im Zustand G700 ist über den Übergang G810 dargestellt. Wird von der Ausführungseinheit G10a ein Umschaltwunsch erkannt, dann wird über den Übergang G820 die Umschalt und Vergleichseinheit G70 in den Zustand G710 überführt. Der Zustand G710 bezeichnet damit die Situation, dass die Ausführungseinheit G10a einen Umschaltwunsch erkannt hat und wartet, bis die Ausführungseinheit G10b ebenfalls einen Umschaltwunsch erkennt. Solange dies nicht der Fall ist, verbleibt die Umschalt und Vergleichseinheit G70 im Zustand G710, was mit dem Übergang G830 dargestellt ist.

Der Übergang G840 findet statt, wenn im Zustand G710 die Ausführungseinheit G10b ebenfalls einen Umschaltwunsch erkennt. Die Umschalt und Vergleichseinheit G70 nimmt damit den Zustand G730 ein. Dieser Zustand bezeichnet die Situation, wenn beide Ausführungseinheiten G10a, G10b einen Umschaltwunsch erkannt haben. In Zustand G730 finden die Synchronisationsverfahren statt, mit denen die zwei Ausführungseinheiten G10a, G10b zueinander synchronisiert werden, um anschließend im Vergleichsmodus zu arbeiten. Während dieses Vorgangs verbleibt die Umschalt und Vergleichseinheit G70 im Zustand G730, was mit dem Übergang G890 dargestellt ist.

Falls im Zustand G700 zuerst von der Ausführungseinheit G10b ein Umschaltwunsch erkannt wird, dann wird über den Übergang G860 in den Zustand G720 geschaltet. Der Zustand G720 bezeichnet damit die Situation, dass die Ausführungseinheit G10b einen Umschaltwunsch erkannt hat und wartet bis die Ausführungseinheit G10a ebenfalls einen Umschaltwunsch
5 erkennt. Solange dies nicht der Fall ist, verbleibt die Umschalt und Vergleichseinheit G70 im Zustand G720, was mit dem Übergang G870 dargestellt ist. Der Übergang G880 findet statt wenn im Zustand G720 die Ausführungseinheit G10 a ebenfalls einen Umschaltwunsch erkennt. Die Umschalt und Vergleichseinheit nimmt damit den Zustand G730 ein.

Falls im Zustand G700 beide Ausführungseinheiten G10a, G10b gleichzeitig einen
10 Umschaltwunsch erkennen, wird sofort in den Zustand G730 übergegangen. Dieser Fall stellt den Übergang G850 dar.

Wenn die Umschalt und Vergleichseinheit G70 im Zustand G730 ist, haben beide Ausführungseinheiten G10a, G10b einen Umschaltwunsch erkannt. In diesem Zustand werden die internen Zustände der Ausführungseinheiten G10a, G10b synchronisiert, um nach
15 Beendigung dieser Synchronisationsvorgänge im Vergleichsmodus zu arbeiten. Mit Beendigung dieser Synchronisationsarbeiten findet der Übergang G900 statt. Dieser Übergang zeigt das Ende der Synchronisation an. Im Zustand G740 arbeiten die Ausführungseinheiten G10a, G10b im Vergleichsmodus. Die Beendigung der Synchronisationsarbeiten kann von den Ausführungseinheiten G10a, G10b selbst signalisiert werden. Dies bedeutet, dass der Übergang
20 G900 stattfindet, wenn beide Ausführungseinheiten G10a, G10b signalisiert haben, dass sie bereit sind im Vergleichsmodus zu arbeiten. Die Beendigung kann auch über eine fest eingestellte Zeit signalisiert werden. Dies bedeutet, dass in der Umschalt und Vergleichseinheit G70 fest codiert ist, wie lange im Zustand G730 verblieben wird. Diese Zeit wird so eingestellt, dass sicher beide Ausführungseinheiten G10a, G10b ihre Synchronisationsarbeiten beendet
25 haben. Nach Ablauf dieser Zeit wird dann der Übergang G900 initiiert. In einer weiteren Variation kann die Umschalt und Vergleichseinheit G70 die Zustände der Ausführungseinheiten G10a, G10b überwachen und selbst erkennen, wenn beide Ausführungseinheiten G10a, G10b ihre Synchronisationsarbeiten beendet haben. Nach dem Erkennen wird dann der Übergang G900 eingeleitet.

30 Solange kein Umschaltwunsch erkannt wird, verbleibt das Multiprozessorsystem G60 im Vergleichsmodus, dargestellt durch den Übergang G910. Wenn im Zustand G740 ein Umschaltwunsch erkannt wird, wird die Umschalt und Vergleichseinheit über den Übergang

- 22 -

G920 in den Zustand G700 versetzt. Wie schon beschrieben, arbeitet das System im Zustand G700 im Performanzmodus. Die Trennung der Programmflüsse beim Übergang vom Zustand G740 in den Zustand G700, kann dann wie im beschriebenen Verfahren durchgeführt werden.

5 In Figur 11 ist ein Multiprozessorsystem G400 mit zwei Ausführungseinheiten G410a, G410b sowie zwei Interruptcontrollern G420a, G420b inklusive darin enthaltener Interrupt Maskierungsregister G430a, G430b und verschiedener Interruptquellen G440a bis G440n dargestellt. Weiter dargestellt ist eine Umschalt- und Vergleichseinheit G450 mit einem speziellen Interruptmaskierungsregister G460.

10

Vorteilhafterweise besitzt jede Ausführungseinheit G410a, G410b ihren eigenen Interruptcontroller G420a, G420b, um im Performanzmodus gleichzeitig zwei Interrupts behandeln zu können.

Dies ist besonders in Systemen vorteilhaft, in denen die Interruptbehandlung einen Flaschenhals

in der Systemperformanz darstellt. Die Interruptquellen G440a bis G440n werden dabei

15

vorteilhafterweise an beide Interruptcontroller G420a, G420b jeweils gleich angeschlossen.

Diese Anschlussart bewirkt, dass ohne weitere Maßnahmen an beiden Ausführungseinheiten

G410a, G410b der gleiche Interrupt ausgelöst wird. Im Performanzmodus werden die Interrupt Controller G420a, G420b so programmiert, dass die entsprechenden Interruptquellen G440a bis

G440n auf die verschiedenen Ausführungseinheiten G410a, G410b je nach Anwendung

20

geeignet aufgeteilt werden. Dies geschieht mittels einer geeigneten Programmierung der

Interrupt Maskierungsregister G430a, G430b. Die Maskierungsregister sehen für jede

Interruptquelle G440a bis G440n ein Bit im Register vor. Falls dieses Bit gesetzt ist, wird der

Interrupt gesperrt, er wird also nicht an die angeschlossene Ausführungseinheit G410a, G410b

weitergeleitet. Günstigerweise wird in einem Performanzmodus eine gegebene Interruptquelle

25

G440a bis G440n von genau einer Ausführungseinheit G410a oder G410b bearbeitet.

Vorteilhafterweise gilt dies zumindest für einige der Interruptquellen. Damit kann erreicht

werden, dass mehrere Interruptquellen G440a bis G440n gleichzeitig bearbeitet werden können,

ohne dass ein Interrupt-Nesting (eine Interrupt Bearbeitung wird durch einen zweiten Interurupt

unterbrochen) oder Interrupt-Pending (die Bearbeitung des Zweiten wird verschoben bis die

30

Bearbeitung des ersten beendet ist) stattfindet.

Im Vergleichsmodus muss sichergestellt werden, dass die Interruptcontroller G420a, G420b an allen Ausführungseinheiten G410a, G410b gleichzeitig den gleichen Interrupt auslösen,

ansonsten würde, einem Vergleichsmodus entsprechend, auf einen Fehler erkannt werden. Dies bedeutet, dass in der Synchronisierungsphase beim Umschalten vom Performanzmodus in den Vergleichsmodus sichergestellt werden muss, dass die Interrupt Maskierungsregister G430a, G430b identisch sind. Diese Synchronisation ist in der Figur 9 in Schritt G660 beschrieben.

5 Diese Synchronisation kann per Software durchgeführt werden, indem beide Interruptmaskierungsregister G430a, G430b entsprechend mit dem gleichen Wert programmiert werden. Es wird vorgeschlagen ein spezielles Register G460 zu verwenden, um den Umschaltvorgang zu beschleunigen. In einer Ausführungsform ist dieses Register G460 in der Umschalt- und Vergleichseinheit G460 angeordnet, es kann aber auch in der
10 Umschaltwunscherkennung G40, in einer kombinierten Umschaltwunscherkennung, im Vergleich, in der Umschalteinheit G80, sowie in allen Kombinationen enthalten sein. Ebenso denkbar ist es, dass dieses Register außerhalb dieser drei Komponenten an einer anderen geeigneten Stelle angeordnet ist. Das Register G460 enthält die Interruptmaskierung, die im Vergleichsmodus gelten soll. Die Umschalt- und Vergleichseinheit G450 erhält von der
15 Umschaltwunscherkennung G40 ein Signal zur Umschaltung von einem Performanz in einen Vergleichsmodus. Nachdem die Interrupts im Schritt G600 gesperrt wurden können, die Interruptmaskierungsregister G430a, G430b der Interruptcontroller G420a, G420b umprogrammiert werden. Dies wird nun per Hardware von der Umschalt und Vergleichseinheit G450 parallel zu den übrigen Synchronisationsschritten durchgeführt, nachdem das
20 Umschaltsignal erhalten wurde und die Interruptcontroller G420a, G420b gesperrt wurden. Günstigerweise werden die Interruptmaskierungsregister G430a, G430b im Vergleichsmodus nicht einzeln umprogrammiert, sondern immer das zentrale Register G460. Dieses wird dann synchron per Hardware auf die beiden Interruptmaskierungsregister G430a, G430b übertragen. Das Verfahren, das hier für ein Interruptmaskierungsregister beschrieben ist, kann in gleicher
25 Weise auf alle Interruptstatusregister, die in einem Interruptcontroller angeordnet sind, übertragen werden. Natürlich ist es auch denkbar, anstelle eines Registers G460 ein anderes Speichermedium zu verwenden, aus dem möglichst schnell auf die Interruptmaskierungsregister G430a, G430b übertragen werden kann.

30 In Figur 12 wird ein Multiprozessorsystem G1000 mit zwei Ausführungseinheiten G1010a, G1010b, einer Umschalt- und Vergleichseinheit G1020, sowie einem Interruptcontroller G1030 mit drei verschiedenen Registersätzen G1040a, G1040b, G1050 vorgeschlagen. Als Alternative zu der oben beschriebenen Lösung wird ein spezieller

Interruptcontroller G1030 vorgeschlagen, wie in Figur 12 dargestellt. Dieser wird in einem Multiprozessorsystem G1000 verwendet, das im Beispiel mit zwei Ausführungseinheiten G1010a, G1010b, sowie einer Umschalt- und Vergleichseinheit G1020, die zwischen einem Vergleichs und einem Performanzmodus umschalten kann, dargestellt ist.

5 Im Performanzmodus werden dabei die Registersätze G1040a, G1040b verwendet. In diesem Fall arbeitet der Interruptcontroller G1030 genau so wie zwei Interruptcontroller G420a, G420b. Dieses Verhalten ist in Figur 11 dargestellt und beschrieben. Der Registersatz G1040a ist dabei der Ausführungseinheit G1010a zugeordnet und der Registersatz G1040b der Ausführungseinheit G1010b. Die Interruptquellen G1060a bis G1060n werden per Maskierung
10 auf die Ausführungseinheiten G1010a, G1010b geeignet aufgeteilt. Beim Umschalten von einem Performanzmodus in einen Vergleichsmodus erzeugt die Umschalt- und Vergleichseinheit G1020 ein Signal G1070. Dieses signalisiert dem Interruptcontroller G1030, dass in den Vergleichsmodus umgeschaltet wird bzw., dass das System ab diesem Zeitpunkt im Vergleichsmodus arbeitet. Der Interruptcontroller G1030 verwendet danach den Registersatz
15 G1050. Damit ist sichergestellt, dass an beiden Ausführungseinheiten G1010a, G1010b die gleichen Interruptsignale entstehen. Mit einem Wechsel vom Vergleichsmodus in den Performanzmodus, den die Umschalt- und Vergleichseinheit G1020 wieder über das Signal G1070 dem Interruptcontroller G1030 signalisiert, wird auf die Registersätze G1040a, G1040b wieder umgeschaltet. Günstigerweise kann damit auch ein Schutz der entsprechenden
20 Registersätze erreicht werden, indem im Performanzmodus nur ein Schreiben auf die Registersätze G1040a, G1040b erlaubt ist und ein Schreiben auf den Registersatz G1050, der dem Vergleichsmodus vorbehalten ist, per Hardware unterbunden wird. Gleiches ist auch in der anderen Richtung möglich, dass im Vergleichsmodus nur ein Schreiben auf den Registersatz G1050 erlaubt ist und ein Schreiben auf die Registersätze G1040a, G1040b unterbunden wird.

25 In Figur 13 ist die einfachste Form eines Vergleichers M500, G20 dargestellt. Eine wesentliche Komponente in einem Multiprozessorsystem G60 mit wenigstens zwei Ausführungseinheiten G10a, G10b mit einer Umschaltung zwischen einem Performanzmodus und einem Vergleichsmodus ist der Vergleichs M500. In der einfachsten Form ist er in Figur 13
30 dargestellt. Die Vergleichskomponente M500 kann zwei Inputsignale M510 und M511 aufnehmen. Sie vergleicht diese dann auf Gleichheit, im hier dargestellten Kontext vorzugsweise im Sinne einer Bit-weisen Gleichheit. Im Gleichheitsfalle wird der Wert der Inputsignale M510, M511 auf das Outputsignal M520 gegeben und das Fehlersignal M530 wird

nicht aktiv, d.h. es signalisiert den „Gut“-Zustand. Falls sie Ungleichheit detektiert, wird das Fehlersignal M530 aktiviert. Das Signal M520 kann dann optional deaktiviert werden. Dies hat den Vorteil, dass der Fehler aus dem entsprechenden System nicht herauskommt („fault containment“). D.h. andere Komponenten, die außerhalb der Ausführungseinheiten liegen,
5 werden durch das potenziell fehlerhafte Signal nicht korrumpiert. Es gibt jedoch auch Systeme, in denen das Signal M520 nicht deaktiviert werden muss. Dies ist z.B. dann der Fall, wenn auf Systemebene nur Fail-silence gefordert ist. Dann kann das Fehlersignal beispielsweise nach extern geführt werden.

10 Von diesem Basis-System ausgehend sind eine Vielzahl von Ausführungsformen denkbar. Zunächst kann die Komponente M500 als so genannte TSC- Komponente (totally self checking) ausgeführt werden. In diesem Fall wird das Fehlersignal M530 auf mindestens zwei Leitungen („dual rail“) nach außen geführt, und es ist durch interne Design- und Fehlerentdeckungsmaßnahmen sichergestellt, dass in jedem möglichen Fehlerfall der
15 Vergleichskomponente dieses Signal korrekt oder erkennbar unkorrekt vorliegt. Ein dual rail Signal stellt dabei ein binäres Signal über zwei Leitungen zur Verfügung, vorzugsweise so, dass die beiden Leitungen im fehlerfreien Fall zueinander invertiert sind. Eine bevorzugte Variante in der Benutzung des erfindungsgemäßen Systems ist es, einen solchen TSC- Vergleichs- zu verwenden.

20 Eine zweite Klasse von Ausführungsformen kann dahingehend unterschieden werden, welchen Grad der Synchronität die beiden Inputs M510, M511 (bzw. M610, M611) haben müssen. Eine möglich Ausführungsform ist durch taktweise Synchronität gekennzeichnet, d.h. der Vergleich der Daten kann in einem Takt durchgeführt werden.

25 Eine leichte Änderung entsteht dadurch, dass bei einem festen Phasenversatz zwischen den Inputs ein synchrones Verzögerungselement verwendet wird, das die entsprechenden Signale beispielsweise um halbzahlige oder ganzzahlige Taktperioden verzögert. Ein solcher Phasenversatz ist nützlich, um Common Cause Fehler zu vermeiden, d.h. solche Fehlerursachen, die mehrere Verarbeitungseinheiten gleichzeitig und gleichartig beeinflussen
30 können.

Figur 14 beschreibt deshalb eine weitere Ausführungsform. Die Komponenten und Signale M600, M610, M611, M620, M630 in haben die gleiche Bedeutung wie die entsprechenden

Komponenten und Signale M500, M510, M511, M520, M530 aus Figur 13. In Figur 14 ist daher über diese Komponenten hinaus die Komponente M640 eingefügt, die den zeitlich früheren Input um den Phasenversatz verzögert. Vorzugsweise ist dieses Verzögerungselement im Vergleich untergebracht, um es nur im Vergleichsmodus zu verwenden.

5 Alternativ oder ergänzend kann man Zwischenpuffer M650, M651 in die Inputkette legen, um solche Asynchronitäten, die sich nicht als reiner Takt- oder Phasenversatz darstellen, ebenfalls tolerieren zu können. Vorzugsweise werden diese Zwischenpuffer als FIFO-Speicher (first-in, first out) ausgelegt. Ein solcher Speicher hat einen Eingang und einen Ausgang und kann mehrere Speicherworte speichern. Ein einkommendes Speicherwort wird bei Eintreffen eines
10 neuen Speicherworts in seiner Stelle verschoben. Nach der letzten Stelle (der Tiefe des Puffers) wird es „aus dem Speicher hinaus“ verschoben. Falls ein solcher Puffer vorliegt, kann man auch Asynchronitäten bis zur maximalen Tiefe des Puffers tolerieren. In diesem Fall muss ein Fehlersignal auch dann ausgegeben werden, wenn der Puffer überläuft.

15 Weiter kann man im Vergleich Ausführungsformen danach unterscheiden, wie das Signal M520 (oder M620) generiert wird. Eine bevorzugte Ausführungsform ist es, die Inputsignale M510, M511 (bzw. M610, M611) auf den Ausgang zu legen und die Verbindung durch Schalter unterbrechbar zu machen. Der besondere Vorteil dieser Ausführungsform ist es, dass zur Umschaltung zwischen Performanzmodus und möglichen verschiedenen Vergleichsmodi
20 dieselben Schalter verwendet werden können. Alternativ können die Signale auch aus vergleichererinternen Zwischenspeichern generiert werden.

Eine letzte Klasse von Ausführungsformen kann dahingehend unterschieden werden, wie viele
25 Inputs am Vergleich vorliegen und wie der Vergleich reagieren soll. Bei drei Inputs kann ein Majoritätsvoting, ein Vergleich von allen drei oder ein Vergleich von nur zwei Signalen vorgenommen werden. Bei vier oder mehr Inputs sind entsprechend mehr Ausführungsformen denkbar. Eine ausführliche Beschreibung der möglichen Ausführungsformen ist in der Beschreibung von Figur 20 enthalten.

30 Die genaue Wahl der Ausführungsformen ist vorzugsweise mit den verschiedenen Betriebsmodi des Gesamtsystems zu koppeln. D.h., wenn es mehrere verschiedene Performanz- oder Vergleichsmodi gibt, dann sind diese vorzugsweise mit dem entsprechenden Modus des Vergleichers gekoppelt.

An einigen Stellen in dieser Erfindung ist es notwendig oder vorteilhaft, einen Vergleicher oder ein allgemeineres Voting-/Verarbeitungs-/Sortierelement zu deaktivieren oder passiv zu machen (im Folgenden der Einfachheit halber immer als Vergleicher bezeichnet). Dazu gibt es viele
5 Möglichkeiten. Zum einen kann man an den Vergleicher ein Signal führen, mit er aktiviert oder deaktiviert wird. Dazu ist im Vergleicher eine zusätzliche Logik einzufügen, die dies durchführen kann. Eine weitere Möglichkeit ist es, dem Vergleicher keine zu vergleichenden Daten zuzuführen. Eine dritte Möglichkeit ist es, auf Systemebene das Fehlersignal des Vergleichers zu ignorieren. Weiter kann man auch das Fehlersignal selbst unterbrechen. Allen
10 Möglichkeiten ist gemeinsam, dass es im System keine Rolle spielt, dass zwei oder mehr Daten, die potenziell verglichen werden, verschieden sind. Falls dies der Fall ist, wird der Vergleicher als passiv oder deaktiviert betrachtet.

Im Weiteren wird eine Implementierung eines Umschalters in Verbindung mit einem
15 Vergleicher, also einer Umschalt- und Vergleichseinheit G70 betrachtet. Diese Implementierung ist besonders günstig, falls diese zusammen mit den Ausführungseinheiten G10a, G10b innerhalb eines Chips ausgeführt sind.

Durch Zusammenfassung der Komponenten Vergleicher und Umschalter entsteht nur ein sehr
20 kleiner Hardwareoverhead bei Implementierung innerhalb eines Chips. Eine bevorzugte Variante der Implementierung ist es deshalb, diese beiden Teile in einer Komponente zusammenzufassen. Dies ist eine Komponente mit mindestens den Eingangssignalen (Output Ausführungseinheit 1, Output Ausführungseinheit 2), mindestens den Ausgangssignalen (Output 1, Output 2), einem logischen Outputsignal „Output gesamt“ (kann physikalisch mit
25 Output 1 oder Output 2 übereinstimmen) und einem Vergleicher. Die Komponente hat die Fähigkeit den Modus umzuschalten, im Performanzmodus alle Signale durchlassen, und in einem Vergleichs-Modus mehrere Signale zu vergleichen und ggf. eines durchlassen. Zusätzlich sind noch weitere Eingangs- und Ausgangssignale vorteilhaft: Ein Fehlersignal zur
30 Signalisierung eines detektierten Fehlers, ein Modussignal um den Modus zu signalisieren, in dem diese Komponente sich befindet, und Steuersignale von und zu der Komponente.

In einem bevorzugten Ausführungsbeispiel sind die zwei oder mehr Ausführungseinheiten im Performanzmodus als Master an einen prozessorinternen Bus angeschlossen. Die

Vergleichseinheit ist deaktiviert oder das Fehlersignal, das bei einem unterschiedlichen Verhalten der Ausführungseinheiten in einem der denkbaren Vergleichsmodi erzeugt wird, ist maskiert. Dies bedeutet, dass die Umschalt und Vergleichseinheit für die Software transparent ist. In dem betrachteten Vergleichsmodus werden die zu vergleichenden physikalischen Ausführungseinheiten als eine logische Ausführungseinheit am Bus behandelt, d.h. es tritt nur ein Master am Bus auf. Das Fehlersignal des Vergleichers ist aktiviert. Die Umschalt und Vergleichseinheit trennt dazu alle bis auf eine Ausführungseinheit über Schalter vom prozessorinternen Bus ab, dupliziert die Eingaben der einen logischen Ausführungseinheit und stellt diese allen am Vergleichsmodus beteiligten Ausführungseinheiten zur Verfügung. Beim Schreiben auf den Bus werden die Ausgaben in der Vergleichseinheit verglichen und bei Gleichheit werden diese Daten über den einen vorhandenen Zugang zum Bus geschrieben.

In Figur 15 und Figur 16 ist das prinzipielle Verhalten der bevorzugten Komponente M700 (Umschalt- und Vergleichseinheit, entspricht G70) beschrieben. Der Einfachheit halber ist diese Figur nur für zwei Ausführungseinheiten gezeichnet. Figur 15 zeigt dabei den Status der Komponente im Vergleichsmodus, Figur 16 im Performanzmodus. Die verschiedenen Schalterstellungen in diesen Modi werden von M700 durch die Ansteuerung M760 realisiert. Die beiden Ausführungseinheiten M730, M731 können zunächst im Performanzmodus auf den Daten- und Adressbus M710 schreiben, wenn die Schalter M750 und M751 geschlossen sind, wie in Figur 16 gezeigt. Es wird vorausgesetzt, dass eventuelle Schreibkonflikte entweder über das Busprotokoll oder durch weitere, nicht eingezeichnete Komponenten aufgelöst werden. Im Vergleichsmodus ist das Verhalten zumindest aus logischer Sicht ein anderes. Wie in Figur 15 gezeigt, sind dann die Schalter M750, M751 geöffnet und somit die direkten Zugriffsmöglichkeiten unterbrochen. Im Gegensatz zu Figur 16 sind in Figur 15 dann allerdings die Schalter M752, M753 geschlossen. Die Signale M740, M741 der Ausführungseinheiten M730, M731 werden an die Vergleichskomponente M720 geleitet. Diese ist mindestens so aufgebaut, wie in Figur 13 gezeichnet, sie kann aber auch Erweiterungen, wie in Figur 14 beschrieben, beinhalten. Auf eine Darstellung des Fehlersignals oder auch weiterer Signale der Vergleichskomponente M720 wird in Figur 15, und Figur 16 aber verzichtet. Falls die beiden Signale übereinstimmen, wird der Schalter M754 geschlossen, und eines der beiden übereinstimmenden Signale dann auf den Adress/Datenbus M710 weitergeleitet. In Summe ist es dazu notwendig, dass die Umschalt- und Vergleichseinheit M700 die Schalter M750-M754 beeinflussen kann. Die jeweilige Schalterstellung ist abhängig vom Modus und von der

Fehlererkennung. Varianten, in denen der Schalter M754 immer geschlossen ist, und eine geeignete Systemreaktion durch das Fehlersignal erzeugt wird, sind hiermit auch abgedeckt.

In Figur 17 ist eine Variante der Umschalt und Vergleichseinheit dargestellt. Auch schon für ein
5 einfaches System mit nur zwei Ausführungseinheiten G10a, G10b gibt es viele Varianten der Implementierung einer Umschalt- und Vergleichseinheit. Eine weitere, die besonders vorteilhaft ist, wenn keine Puffer im Vergleich zu verwenden sind, ist in Figur 17 gezeigt. Wie in Figur 15, Figur 16 gibt es die Signale M840, M841 der Ausführungseinheiten. Letztere sind in dieser Figur nicht gezeichnet. In der erfindungsgemäßen Komponente M800 gibt es eine Moduslogik
10 M810, die den Modus der Komponente vorgibt. Im Performanzmodus schließt sie den Schalter M831, im Vergleichsmodus öffnet sie ihn. Weiter gibt sie das Modussignal an den Vergleichs M820. Dieser vergleicht in dieser Implementierung immer, benutzt aber das Ergebnis des Vergleichs und das Modussignal, um den Schalter M830 anzusteuern. Im Performanzmodus ist der Schalter immer geschlossen, im Vergleichsmodus immer dann, wenn kein Fehler vorliegt.
15 Natürlich kann auch, falls einmal ein Fehler festgestellt wurde, der Schalter weiterhin geöffnet bleiben, bis ein entsprechender Reset kommt.

In Figur 18 ist eine weitere Ausführungsform der Umschalt und Vergleichseinheit dargestellt. Diese Alternative, verfügt zwar über mehr Schalter, lässt dafür aber den Vergleich im
20 Performanzmodus inaktiv und kann deshalb auch leichter mit Asynchronitäten umgehen. Wieder gibt es die beiden Signale M940, M941 der Ausführungseinheiten. Letztere sind wieder in diesem Bild nicht gezeichnet. In der erfindungsgemäßen Komponente M900 gibt es eine Moduslogik M910, die den Modus der Komponente vorgibt. Im Performanzmodus schließt sie den Schalter M931 und öffnet die Schalter M932, M933. Damit wird die Vergleichskomponente
25 M920 in diesem Modus nicht mit Daten beschickt. Dies erlaubt bei Asynchronitäten längere Pufferzeiten, bzw. in einer Implementierung geringere Puffertiefen. Im Performanzmodus ist der Schalter M930 immer geschlossen. Im Vergleichsmodus schließt die Komponente M910 die Schalter M932, M933 und unterbricht den direkten Zugriff auf den Bus durch Öffnen des Schalters M931. Optional kann die Moduslogik M910 dem Vergleichs M920 den Modus noch
30 mitteilen. Im Vergleichsmodus ist im fehlerfreien Fall der Schalter M930 geschlossen. Im Fehlerfall unterbricht die Vergleichskomponente M920 die Weiterleitung des Signals M940 auf den Bus durch Öffnen des Schalters M930.

In den beschriebenen Zeichnungen ist es ohne Aufwand möglich, die Modus oder Fehlersignale nach außen zu leiten. Es ist weiter, insbesondere zur Generierung des internen Moduszustands, problemlos möglich, dass weitere Signale an die Komponente gehen.

5 Zusammenfassend ist eine bevorzugte Implementierung dieser Komponente also dadurch gekennzeichnet, dass es mehrere Verarbeitungseinheiten gibt, die Ausgangssignale auf den Bus (z.B. Adress/Daten-Bus) schreiben können. Wesentlich ist, dass die Komponente mindestens zwei der Ausgangssignale der Ausführungseinheiten verarbeiten kann (z.B. vergleichen, aber
10 eventuell auch voten oder sortieren) und dass die Komponente mindestens einen Schalter beeinflussen kann, mit dem mindestens einer der direkten Buszugriffe unterbrochen wird. Besonders nützlich ist dies, wenn die Ausführungseinheiten Rechnerkerne sind. Weiterhin ist es vorteilhaft, wenn der Zustand der beeinflussbaren Schalter den Betriebsmodus der Recheneinheit charakterisiert.

15 Die Systemeigenschaften, insbesondere die möglichen Vergleichsmodi, werden dann besonders gut umgesetzt, wenn die Komponente ein Signal auf den Adress-Datenbus legen kann. Vorteilhafterweise ist dieses eine Durchschaltung eines der Ausgangssignale einer der Ausführungseinheiten. Alternativ kann dies aus der Verarbeitung verschiedener Ausgangssignale der verschiedenen Ausführungseinheiten entstehen.

20 Wie beispielsweise schon in den Beschreibungen zu den Figuren Figur 17, Figur 18 deutlich wurde, kann man im System und - je nach Einteilung in die Komponenten - auch in einer der Komponenten eine Modus-Information identifizieren. Diese Modusinformation kann je nach Implementierung sogar explizit in einer Teilkomponente vorliegen. In einer bevorzugten
25 Implementierung kann dieses Signal auch aus der Komponente herausgeführt werden und anderen Teilen des Systems zur Verfügung gestellt werden.

Im allgemeinen Fall kann man das erfindungsgemäße Verhalten an Figur 21 erläutern. Die Signale und Komponenten N100, N110, N120, N130, N140, N141, N142, N143, N14n, N160,
30 N161, N162, N163, N16n haben die gleiche Bedeutung wie in Figur 20. Darüber hinaus ist das Modussignal N150 und das Fehlersignal N170 in dieser Figur eingezeichnet. Das optionale Fehlersignal wird von der Fehlerschaltungslogik N130, die die Fehlersignale sammelt, generiert und ist entweder eine direkte Weiterleitung der Einzelfehlersignale oder eine Bündelung der

darin enthaltenen Fehlerinformation. Das Modussignal N150 ist optional, seine Verwendung außerhalb dieser Komponente kann aber an vielen Stellen vorteilhaft sein. Die Kombination der Information der Schaltlogik N110 (d.h. die in der Beschreibung von Figur 20 beschriebene Funktion) und der Verarbeitungslogik (d.h. die Festlegung der Vergleichsoperation pro Ausgangssignal, d.h. pro Funktionswert) ist die Modusinformation und diese legt den Modus fest. Diese Information ist im allgemeinen Fall natürlich mehrwertig, d.h. nicht nur über ein logisches Bit darstellbar. Nicht alle theoretisch denkbaren Modi sind in einer gegebenen Implementierung sinnvoll, man wird vorzugsweise die Zahl der erlaubten Modi einschränken. Das Modussignal bringt dann die relevante Modusinformation nach außen. Eine HW-Implementierung ist vorzugsweise so dargestellt, dass das extern sichtbare Modussignal konfiguriert werden kann. Vorzugsweise sind ebenfalls die Verarbeitungslogik und die Schaltlogik konfigurierbar gestaltet. Vorzugsweise sind diese Konfigurationen aufeinander abgestimmt. Alternativ kann man auch nur oder ergänzend Änderungen des Modussignals nach außen geben. Dies hat insbesondere in einer Zweierkonfiguration Vorteile.

Vorzugsweise wird dieses Modussignal abgesichert. Eine Implementierung im Zweier-System wird, beispielsweise basierend auf der Implementierung, die in Figur 17 dargestellt ist, in Figur 19 dargestellt. Dort wird das Signal M850 aus der Umschalt und Vergleichseinheit herausgeführt. In einem Zweier-System ist diese Information logisch über ein Bit darstellbar. Eine Absicherung kann dann vorzugsweise über ein dual-rail-Signal dargestellt werden. Im allgemeinen Fall kann man das Signal ebenfalls über eine Verdoppelung, die optional invertiert ist absichern. Alternativ kann man auch ein Parity generieren, das vorzugsweise intern eigensicher generiert wird, oder einen CRC (cyclic redundancy check) oder ECC (error correcting code) verwenden.

Das Modussignal kann außerhalb der Komponente verwendet werden. Zunächst kann es zur Eigenüberwachung des Betriebssystems verwendet werden. Dieses ist aus SW-Sicht für einen Umschaltung verantwortlich und sollte immer wissen, in welchem Modus sich das System befindet und das System auch in diesen Modus bringen. Eine Überprüfung dieses Signals kann damit zur Absicherung verwendet werden. Dies kann zunächst direkt geschehen. Eine alternative Möglichkeit ist es aber auch, über Timer oder andere „unabhängige“ Einheiten eine Anfrage beim Betriebssystem mit diesem Signal zu plausibilisieren.

Allgemein kann dieses Signal optional auch in anderen Datensenken eines μC (oder allgemeineren Recheneinheit) verwendet werden. Beispielsweise kann eine MPU (memory protection unit) so programmiert sein, dass sie bestimmte Speicherzugriffe (von bestimmten Ausführungseinheiten) nur in bestimmten Modi erlaubt. Eine MPU ist dabei eine Einheit, die sicherstellen kann, dass nur erlaubte Zugriffe auf den Daten/Adressbus durchgeführt werden, beispielsweise indem sie für bestimmte Programmteile den Zugriff auf bestimmte Adressräume unterbindet. Durch eine Hinführung des Modussignals an die MPU, entsprechende Konfiguration und Programmierung dieser MPU, und Auswertung dieser Konfigurationsdaten und des Modussignals, kann eine zusätzliche Absicherung erfolgen. Unter Umständen vereinfacht das sogar die Programmierung, falls das Modussignal schon eine ausreichende Information zur Überprüfung ist. Dann reicht eine quasi-statische Programmierung zur Initialisierungszeit des μC aus. Entsprechendes kann für Peripherieeinheiten gelten. Auch hier gibt es Anwendungen, bei denen ein Zugriff auf ein entsprechendes Peripherieelement nur in bestimmten Modi erlaubt ist. Durch eine Hinführung des Modussignals an das Peripherieelement, entsprechende Konfiguration und Programmierung des Peripherieelements, und Auswertung dieser Konfigurationsdaten und des Modussignals, kann eine zusätzliche Absicherung erfolgen. Unter Umständen vereinfacht das sogar die Programmierung, falls das Modussignal schon eine ausreichende Information zur Überprüfung ist. Dann reicht eine quasi-statische Programmierung zur Initialisierungszeit des μC aus. Analog kann die Auswertung dieses Signals auch am Interruptcontroller verwendet werden. Solche Überwachungen können dann die Basis oder einen wesentlichen Bestandteil des Sicherheitskonzepts ausmachen. Durch geeignete Ausführung und SW-Strukturierung kann es möglich sein, das Sicherheitskonzept für eine ganze Fehlerklasse in der betrachteten Anwendung auf diesem Modus-Signal aufzubauen. Dies ist insbesondere dann vorteilhaft, wenn das Modus-Signal in einer geeigneten Form, wie oben beschrieben, eigensicher ist. Es ist in diesem Fall dann weiter vorteilhaft, wenn die betrachtete Komponente die Möglichkeit hat, ein Fehlersignal zu senden oder einen Abschaltpfad zu betätigen, falls sie eine Unstimmigkeit zwischen dem Modus-Signal und dem Zugriff auf sich selbst detektiert.

Ein wesentlicher weiterer Verwendungszweck ist die Auswertung des Modus-Signals außerhalb der Recheneinheit. Eine direkte Anwendung ist die Auswertung in einem dekrementierenden Watchdog. Ein solcher „Watch-Dog“ besteht aus mindestens einem (Zähler-)Register, welches durch den Mikroprozessor auf einen Integer Wert gesetzt werden kann. Nach dem Setzen dieses

Registers dekrementiert der „Watch-Dog“ selbstständig den Wert des Registers mit einer festen Periode. Ist der Wert des Registers Null oder tritt ein Überlauf auf, erzeugt der „Watch-Dog“ ein Fehlersignal. Soll das Fehlersignal nicht erzeugt werden, so muss der Mikroprozessor den Wert des Registers rechtzeitig wieder zurücksetzen. Damit kann (in Grenzen) überprüft werden, ob der Mikroprozessor die Software korrekt ausführt. Führt der Mikroprozessor die Software nicht mehr korrekt aus, wird angenommen, dass in diesem Fall auch der „Watch-Dog“ nicht mehr korrekt bedient wird und somit ein Fehlersignal vom „Watch-Dog“ erzeugt wird. Die Integrität der Hardware und der Datenstrukturen kann zuverlässig in einem Vergleichsmodus überprüft werden, dazu muss aber sichergestellt sein, dass der Mikroprozessor regelmäßig wieder in diesen zurückschaltet. Aufgabe des hier beschriebenen „Watch-Dog“ ist es daher, nicht nur ein Fehlersignal zu erzeugen, wenn er innerhalb eines definierten Zeitraumes nicht mehr zurückgesetzt wird, sondern auch, wenn der Mikroprozessor innerhalb eines definierten Zeitraumes nicht mehr in den definierten Vergleichsmodus zurückschaltet. Beispielsweise kann der „Watch-Dog“ nur dann zurückgesetzt werden, wenn das Modus Signal den definierten Vergleichsmodus der Recheneinheit anzeigt. Damit ist sichergestellt, dass die Recheneinheit regelmäßig in diesen Modus zurückschaltet. Alternativ oder ergänzend wird der Wert im Register des „Watch-Dog“ erst dann dekrementiert, wenn bestimmte Interrupts beim Mikroprozessor ausgelöst werden. Dazu müssen die externen Interruptsignale des μC auch an den Watchdog gekoppelt werden. Im Watchdog ist gespeichert, welche Interrupts den μC in den definierten Vergleichsmodus schalten. Der Watchdog wird „aufgezogen“, sobald ein solcher Interrupt kommt, er wird zurückgesetzt durch das Vorhandensein des korrekten Modus Signals.

Ganz allgemein ist es nützlich, insbesondere in der Anwendung auf ein Sicherheitskonzept, das Modus-Signal in einer μC -externen Quelle auszuwerten. Ein wesentlicher Punkt in der Absicherung des korrekten Ablaufs der Software auf einem Rechner, wie er in der Erfindung beschrieben ist, ist der korrekte Wechsel zwischen den verschiedenen erlaubten Modi. Zunächst soll die Wechselfähigkeit selbst überprüft werden, vorzugsweise auch noch das korrekte Wechseln. Wie oben beschrieben, kann man auch daran interessiert sein, dass ein spezieller Modus regelmäßig angenommen wird. Besonders vorteilhaft ist ein solches Verfahren immer dann, wenn das Modus-Signal selbst eigensicher ausgestaltet ist.

Eine Möglichkeit ist es das Modus-Signal an einen ASIC oder einen anderen μC zu leiten. Dieser kann über Timer und einfache Logik unter Verwendung dieses Signals mindestens folgende Punkte überprüfen:

Kommt die Recheneinheit hinreichend oft (z.B. spätestens alle 1000 μ s) in einen oder mehrere definierte Modi?

Wird beim Wechsel in einen Modus immer ein bestimmtes Signal ausgegeben?

5 Geht die Recheneinheit regelmäßig aus einem Modus heraus?

Sind bestimmte einfache Muster der Abfolge der Modi gültig?

Ist ein allgemeines zeitliches Muster gültig (z.B. im Durchschnitt <70% im Modus 1 und <50% im modus 2)

10 Irgendeine Kombination von logischen, zeitlichen Eigenschaften des Modus-signals, gegebenenfalls ergänzt durch Verwertung zusätzlicher Signale.

In Figur 22 ist dabei die Basiskonfiguration für einen darüber hinaus gehenden Vorschlag beschrieben. In dem zwischen einem solchen Partner-ASIC oder - μ C und der betrachteten Recheneinheit, die über diese Erfindung verfügt, ein spezielles Frage-Antwort-Spiel durchgeführt wird. N300 ist eine Recheneinheit, die ein solches Modus-Signal aussenden kann. Dies kann beispielsweise ein μ C sein mit mehreren Ausführungseinheiten und einer anderen Komponente, die dieses Modussignal generieren kann. Beispielsweise kann diese andere Komponente wie in Figur 19 oder Figur 21 realisiert sein. N300 gibt dieses Signal N310 an den Partner (z.B. andere Recheneinheit, anderer μ C oder ASIC) N330. Dieser kann über das Signal N320 Fragen an N300 stellen, die N300 über N321 zu beantworten hat. Eine solche Frage kann eine Rechenaufgabe sein, deren korrektes Ergebnis über N321 von N300 innerhalb eines definierten Zeitintervalls zu liefern ist. N330 kann die Korrektheit dieses Ergebnisses unabhängig von N300 prüfen. Beispielsweise sind die Ergebnisse in N330 gespeichert oder N330 kann sie selbst ausrechnen. Bei Detektion eines nicht korrekten Werts wird auf einen Fehler erkannt. Das Besondere an der vorgeschlagenen Frage-Antwort-Kommunikation ist es, dass ein parallel zur Antwort das Modus-Signal beobachtet wird. Vorzugsweise sind die Fragen so zu stellen, dass zur Beantwortung durch N300, diese bestimmte Modi einnehmen muss. Damit kann zuverlässig überprüft werden, dass alle Modus-Wechsel funktionsfähig sind, und dass im Programmablauf vorgesehene Moduswechsel auch durchgeführt werden. Insbesondere bei der Initialisierung eines Systems, aber auch im Betrieb, kann dies als wesentlicher Baustein eines Sicherheitskonzepts dienen.

15

20

25

30

- 35 -

Eine weitere Anwendung dieser Idee ist die Auswertung des Modus-Signals in einer Aktoransteuerung. In vielen Anwendungen im Automobilbereich ist heute ein Trend zu so genannten intelligenten Aktoren. Dies sind Aktoren mit einem minimalen Elektronikumfang, der ausreicht, ein Aktor-Stellkommando zu empfangen, und den Aktor dann so anzusteuern, dass dieses Stellkommando dann auch ausgeführt wird.

5

Die Basisidee ist in Figur 23 dargestellt. Eine Recheneinheit N400, die über die Erfindung verfügt, gibt an einen (intelligenten) Aktor oder eine Aktoransteuerung N430 ein Stellkommando über die Verbindung N420. Parallel gibt sie das Modus-Signal an diesen Aktor über die Verbindung N410. Der Aktor N430 prüft anhand des Modus-Signals, ob die Ansteuerung erlaubt ist und gibt optional über das Signal N440 einen Fehlerstatus zurück. Bei fehlerhafter Ansteuerung nimmt er den im System unkritischen Fail-silence Zustand ein.

10

5

10 Patentansprüche

1. Verfahren zur Trennung der Abarbeitung von Programmcode bei einem Rechnersystem mit wenigstens zwei Ausführungseinheiten, wobei zwischen wenigstens zwei Betriebsmodi umgeschaltet wird und ein erster Betriebsmodus einem Vergleichsmodus und ein zweiter Betriebsmodus einem Performanzmodus entspricht und die wenigstens zwei Ausführungseinheiten im Vergleichsmodus den gleichen Programmcode abarbeiten, dadurch gekennzeichnet, dass bei der Umschaltung vom Vergleichsmodus in den Performanzmodus eine Trennung im Programmcode dadurch erfolgt, dass jeder Ausführungseinheit eine Kennung zugeordnet ist und abhängig von der Kennung wenigstens zwei Ausführungseinheiten unterschiedlicher Programmcode zugeordnet wird.
2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die Kennung jeweils in einem Speicher, insbesondere einem Register, einer Ausführungseinheit enthalten ist.
3. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die Kennung jeweils in einem Statusregister einer Ausführungseinheit enthalten ist
4. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die Kennung jeweils in einem Interrupt-Status-Register einer Ausführungseinheit enthalten ist
5. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die Kennung jeweils in einem Interrupt-Maskierungs-Register einer Ausführungseinheit enthalten ist

30

6. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass abhängig von der Kennung für jede Ausführungseinheit auf eine vorgebbare Adresse verwiesen wird, wobei die Ausführungseinheit den bei dieser Adresse beginnenden Programmcode abarbeitet.

5 7. Verfahren nach Anspruch 6, dadurch gekennzeichnet, dass der jeweilige Verweis auf eine vorgebbare Adresse durch einen Adresszeiger erfolgt.

8. Verfahren nach Anspruch 6, dadurch gekennzeichnet, dass der jeweilige Verweis auf eine vorgebbare Adresse durch einen Sprungbefehl erfolgt.

10

9. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass jede Kennung gegen wenigstens eine vorgegebene Kennung verglichen wird und bei Gleichheit ein vorgegebener Programmcode zugeordnet wird.

15

10. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass jede Kennung gegen wenigstens eine vorgegebene Kennung verglichen wird und bei Ungleichheit aller Kennungen ein Ereignis erkannt wird, das zu einem undefinierten Zustand führen würde.

20

11. Vorrichtung zur Trennung der Abarbeitung von Programmcode bei einem Rechnersystem mit wenigstens zwei Ausführungseinheiten, wobei zwischen wenigstens zwei Betriebsmodi umgeschaltet wird und ein erster Betriebsmodus einem Vergleichsmodus und ein zweiter Betriebsmodus einem Performanzmodus entspricht und die wenigstens zwei Ausführungseinheiten im Vergleichsmodus den gleichen Programmcode abarbeiten dadurch gekennzeichnet, dass die Vorrichtung derart ausgebildet ist, dass bei der Umschaltung vom Vergleichsmodus in den Performanzmodus eine Trennung im Programmcode dadurch erfolgt, dass jeder Ausführungseinheit eine Kennung zugeordnet ist und abhängig von der Kennung mindestens zwei Ausführungseinheiten unterschiedlicher Programmcode zugeordnet wird.

25

30

12. Vorrichtung nach Anspruch 11, dadurch gekennzeichnet, dass die Vorrichtung einen Speicher, insbesondere ein Register enthält, in dem die Kennung jeder Ausführungseinheit abgespeichert ist.

- 38 -

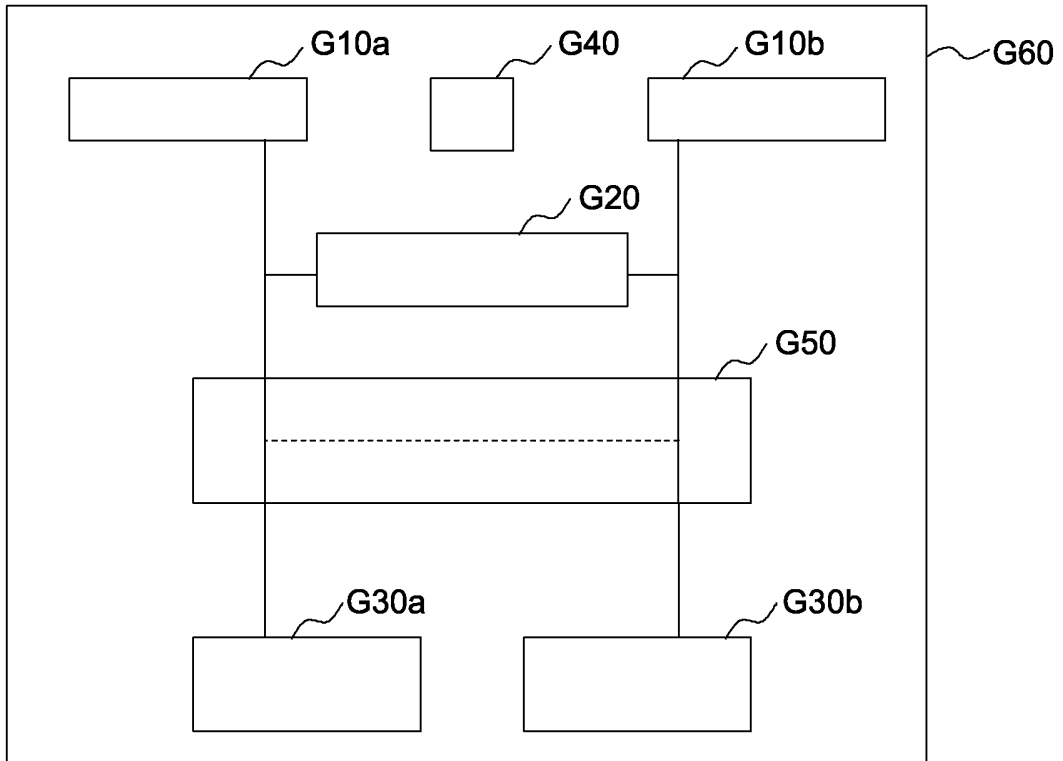
13. Vorrichtung nach Anspruch 11, dadurch gekennzeichnet, dass jede Ausführungseinheit ein Statusregister enthält, in dem jeweils die Kennung der Ausführungseinheit abgespeichert ist.

5 14. Vorrichtung nach Anspruch 11, dadurch gekennzeichnet, dass die Vorrichtung wenigstens ein Interrupt-Status-Register enthält, in dem die Kennungen der Ausführungseinheiten abgespeichert sind.

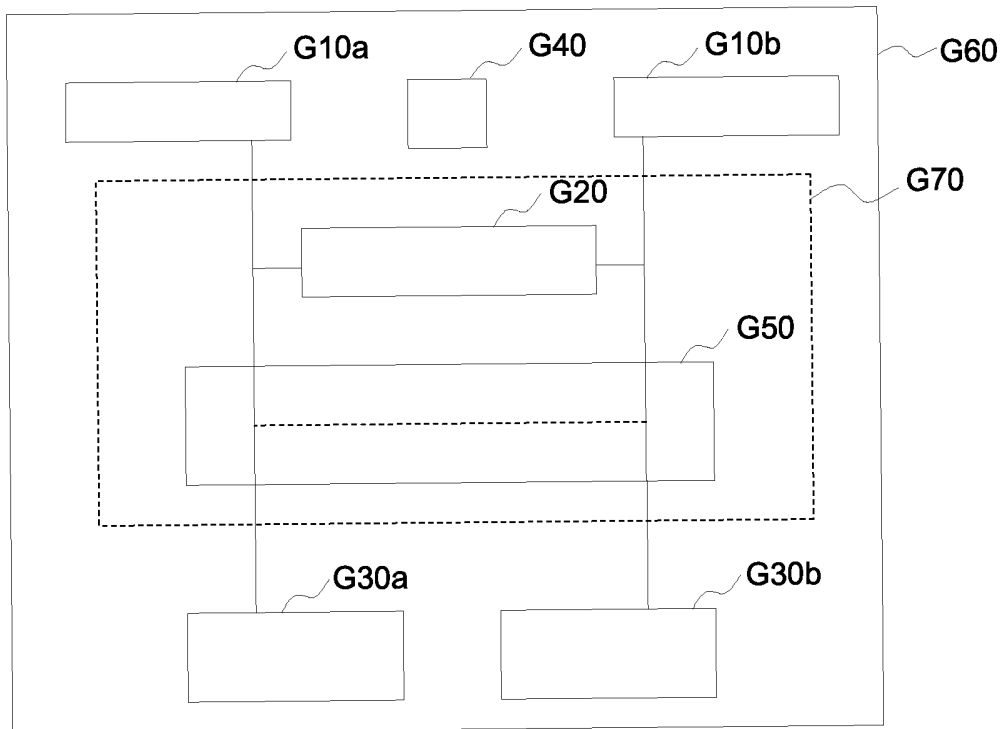
10 15. Vorrichtung nach Anspruch 11, dadurch gekennzeichnet, dass die Vorrichtung wenigstens ein Interrupt-Maskierungs-Register enthält, in dem die Kennungen der Ausführungseinheiten abgespeichert sind.

16. Ausführungseinheit eines Rechnersystems mit einer Vorrichtung nach Anspruch 11.

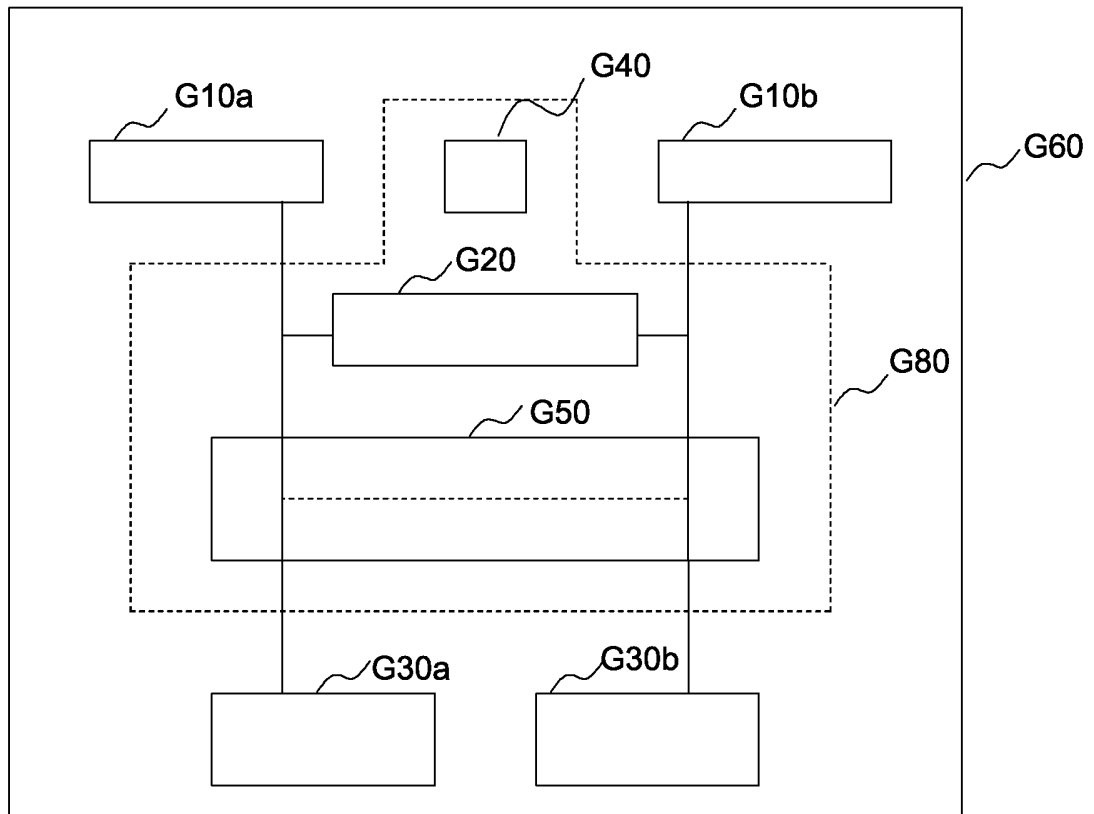
15 17. Rechnersystem mit zwei Ausführungseinheiten und einer Vorrichtung nach Anspruch 11.



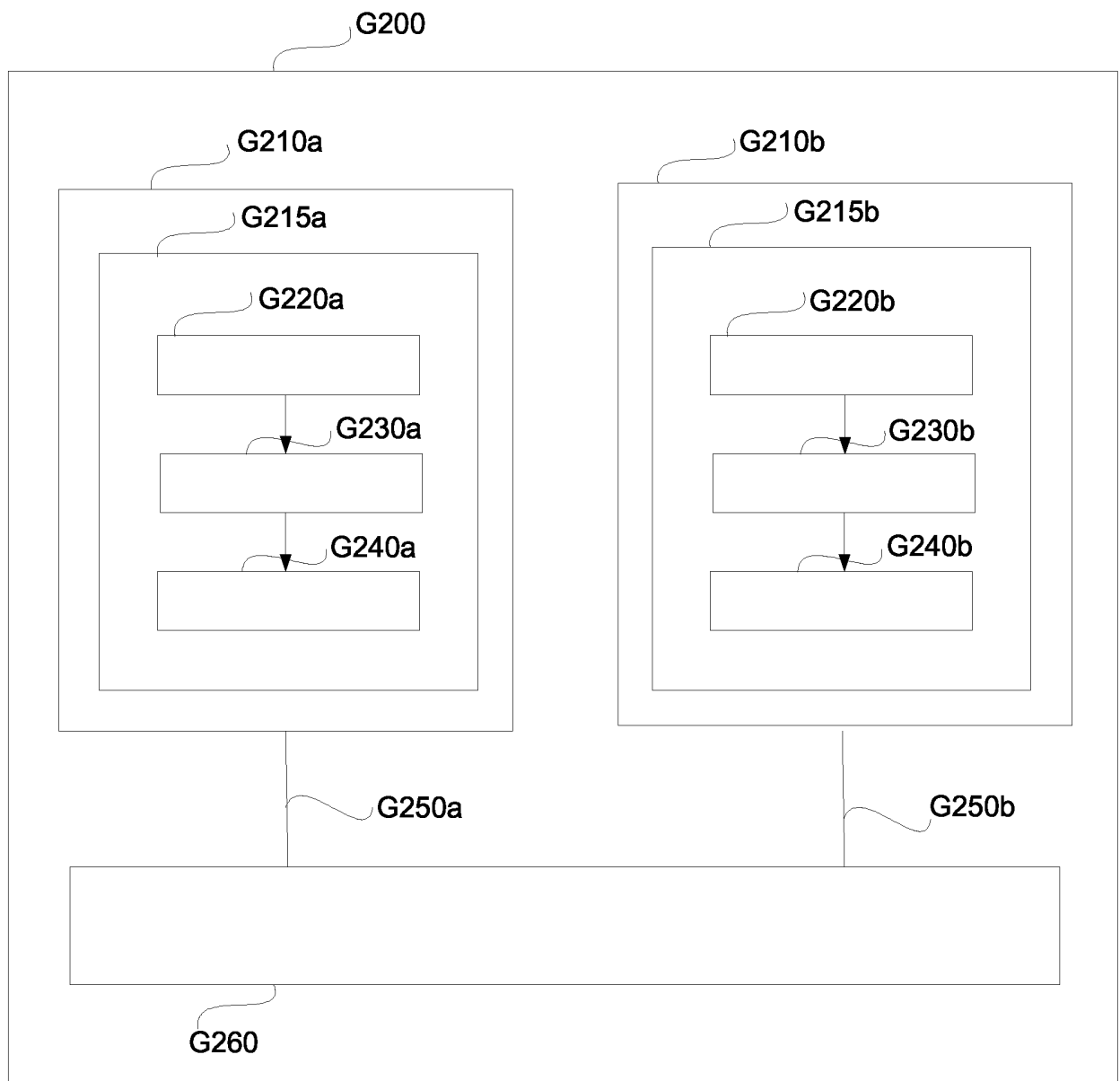
Figur 1



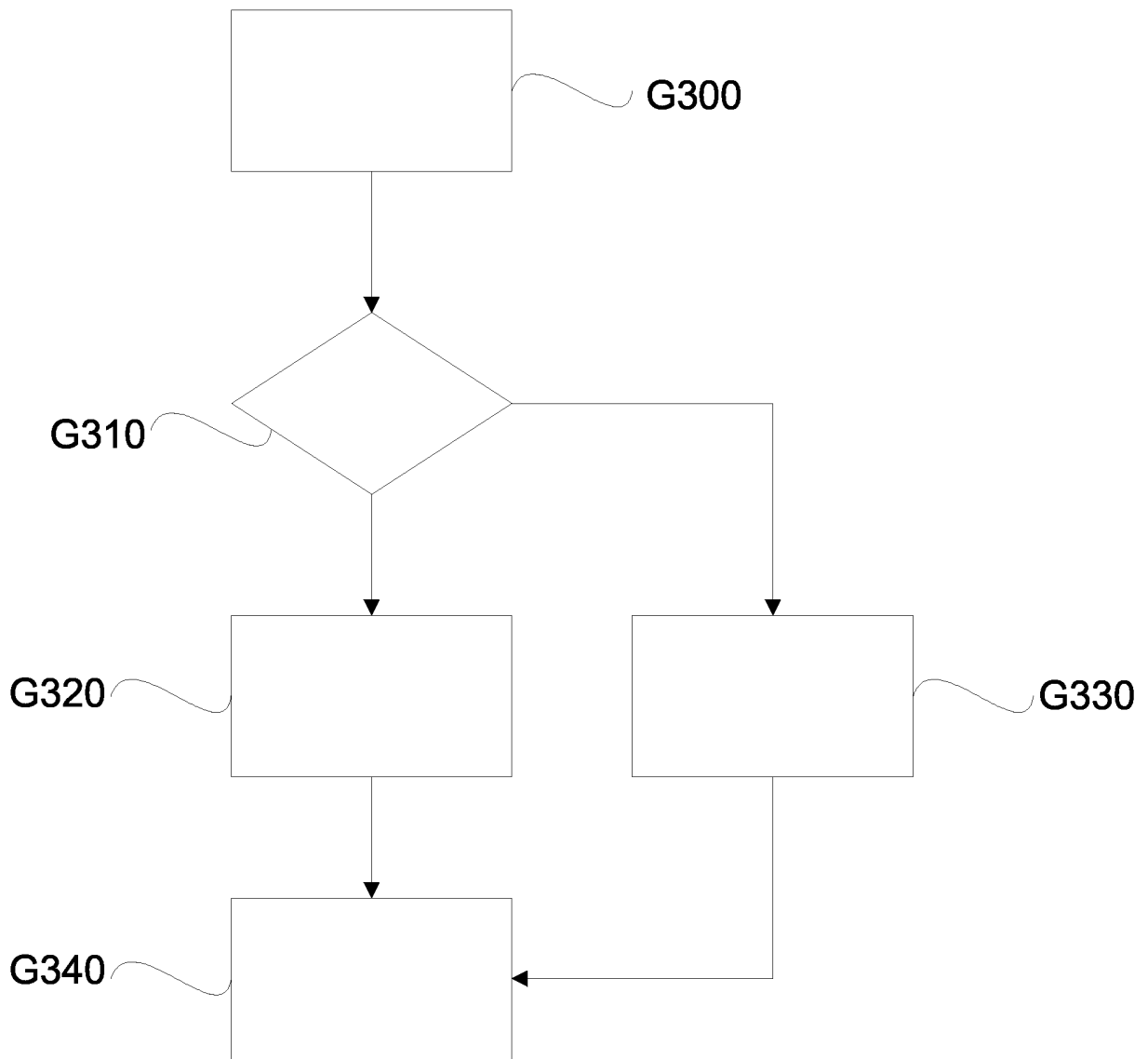
Figur 2



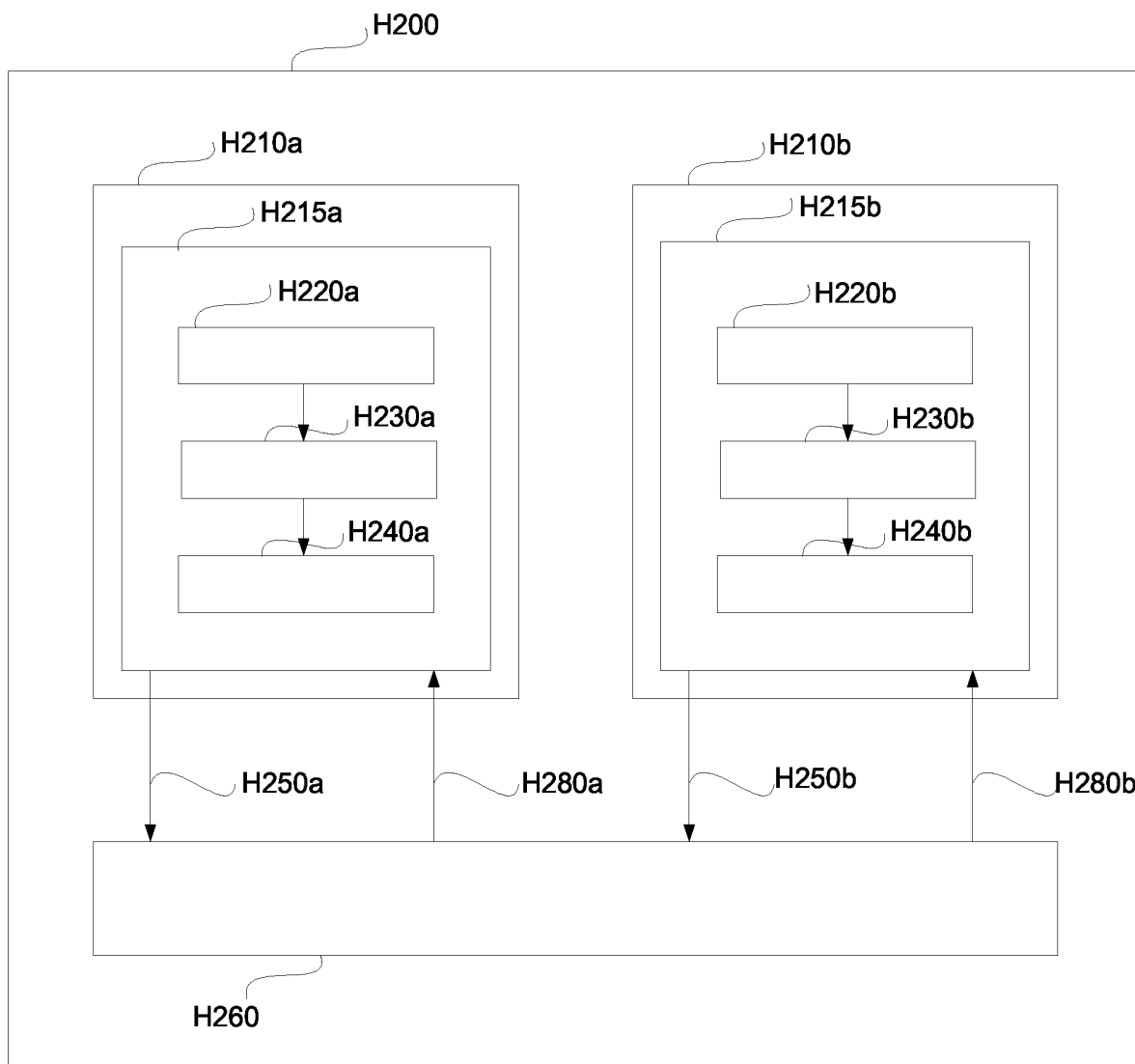
Figur 3



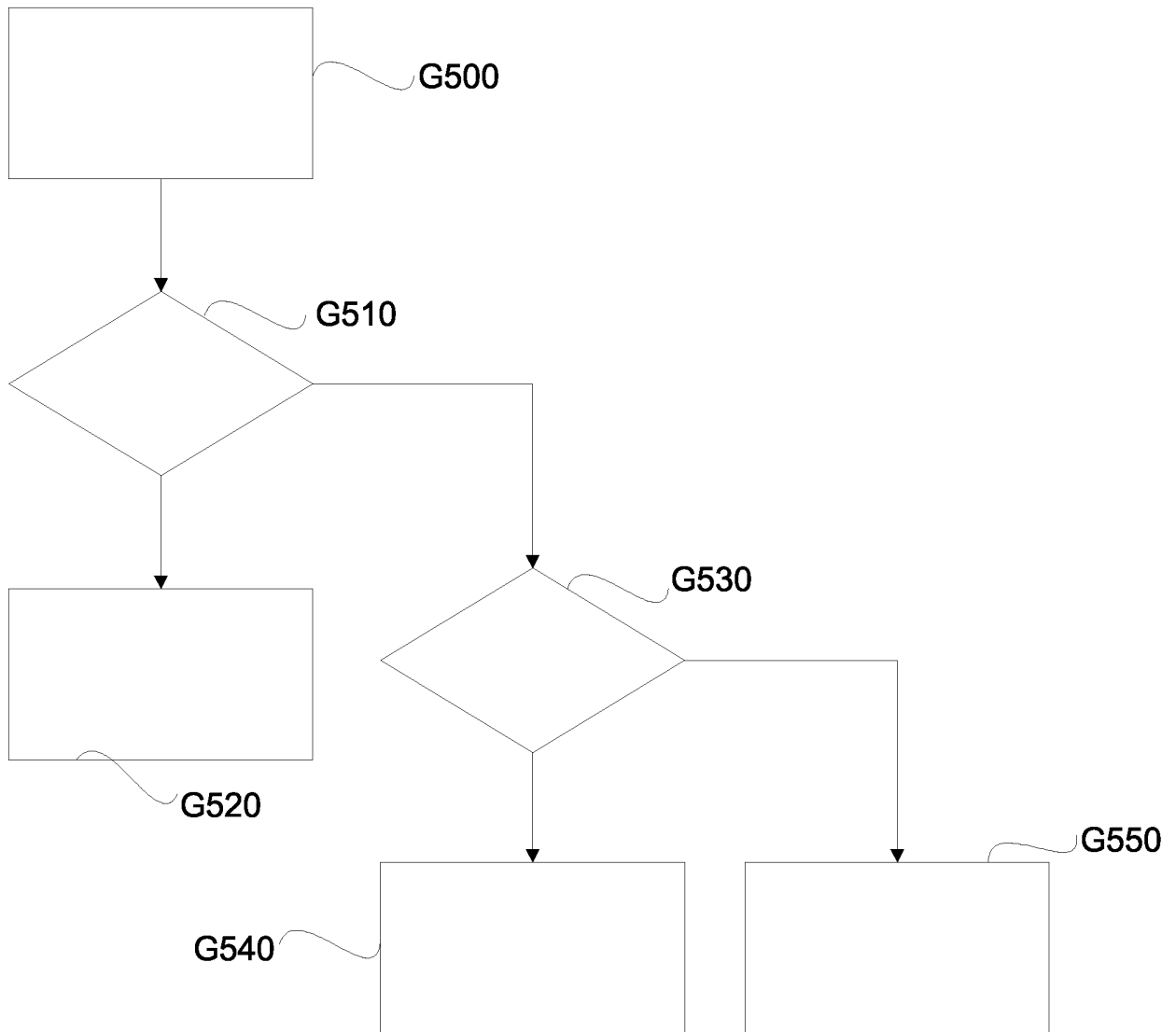
Figur 4



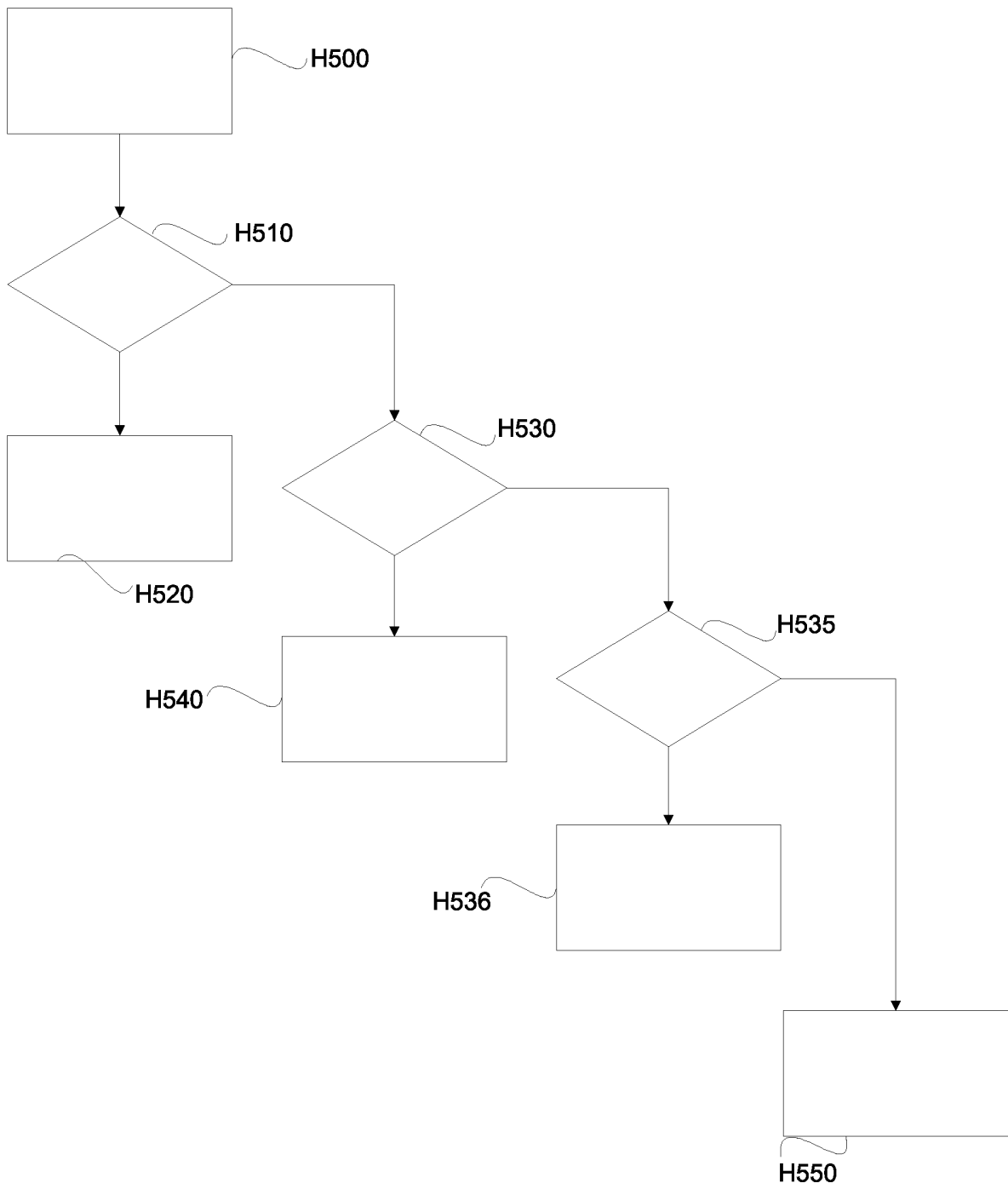
Figur 5



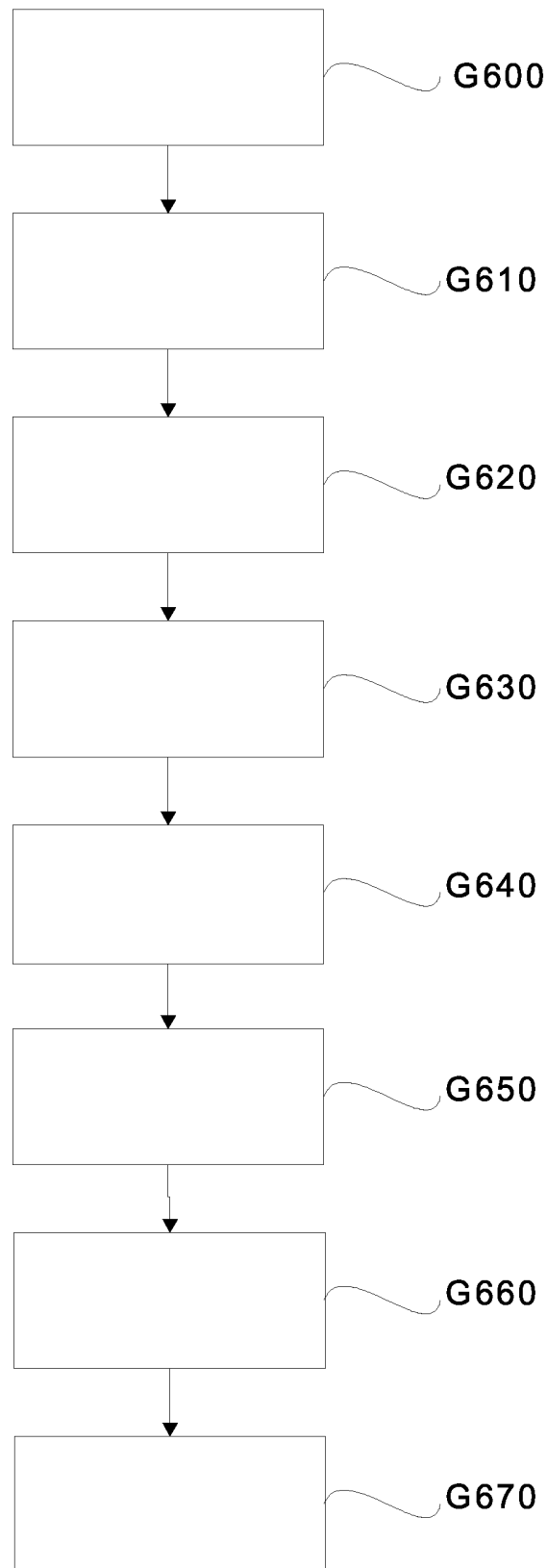
Figur 6



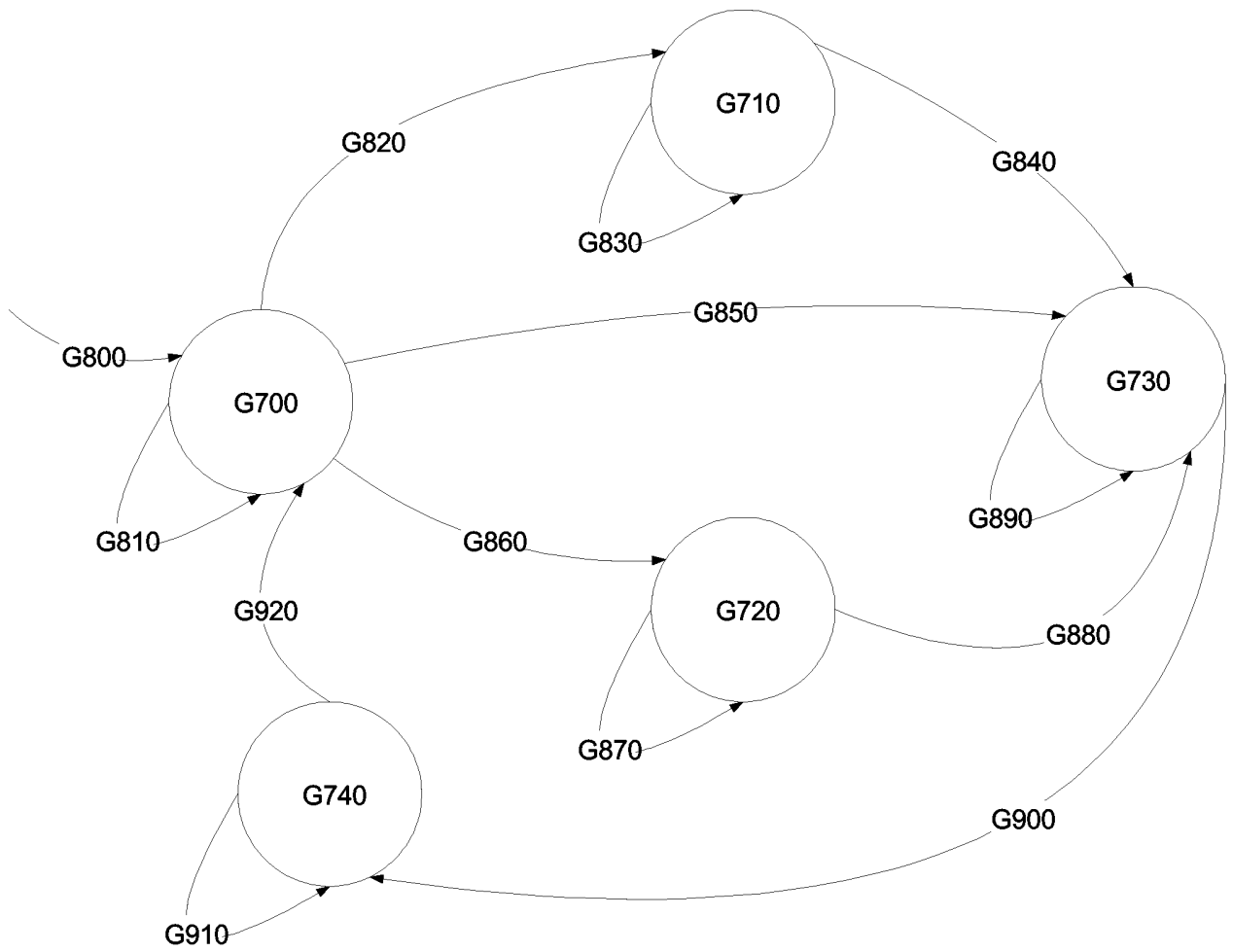
Figur 7



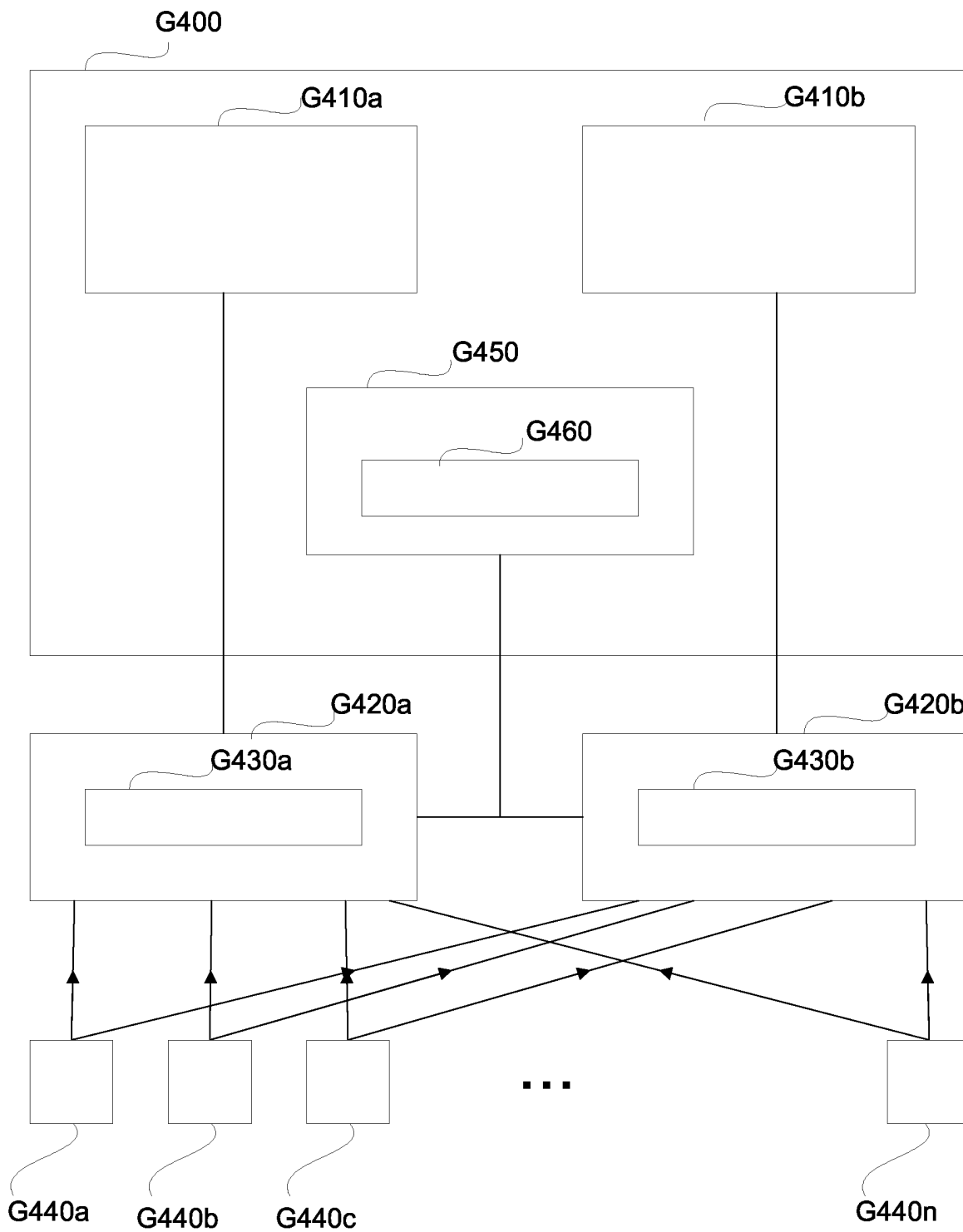
Figur 8



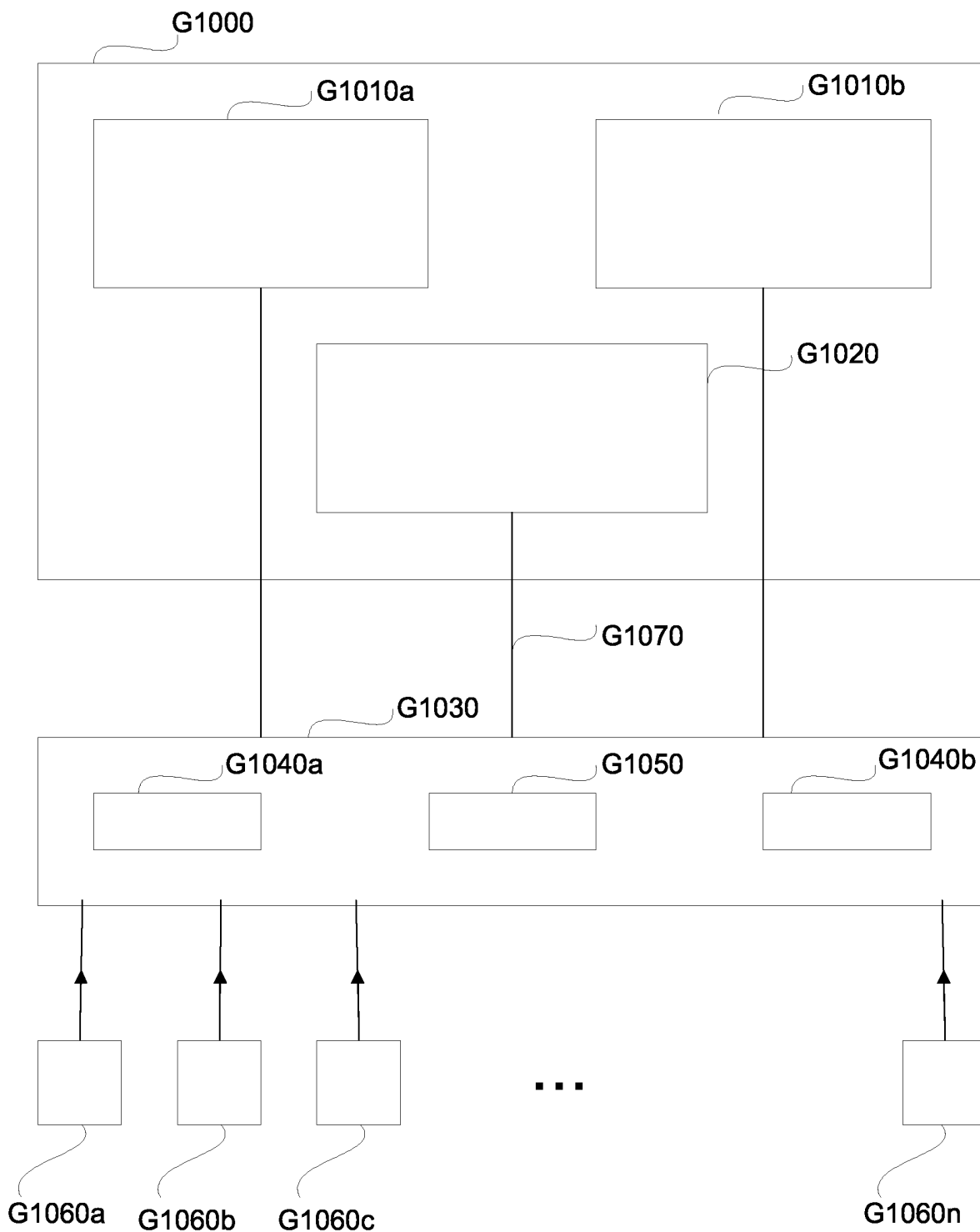
Figur 9



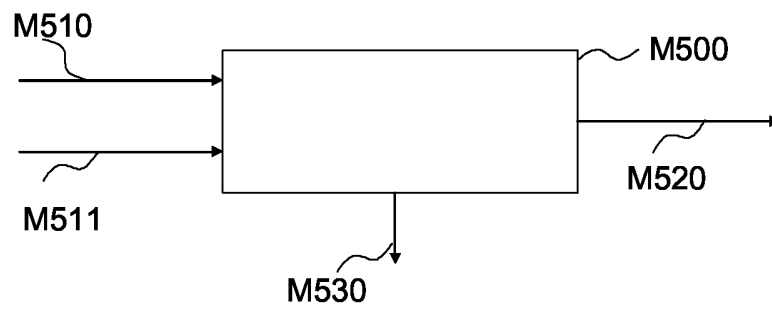
Figur 10



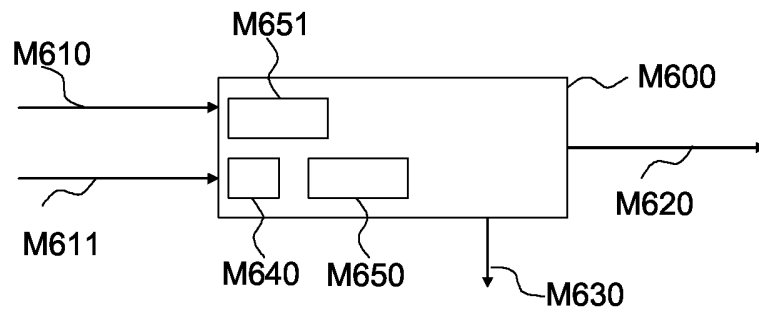
Figur 11



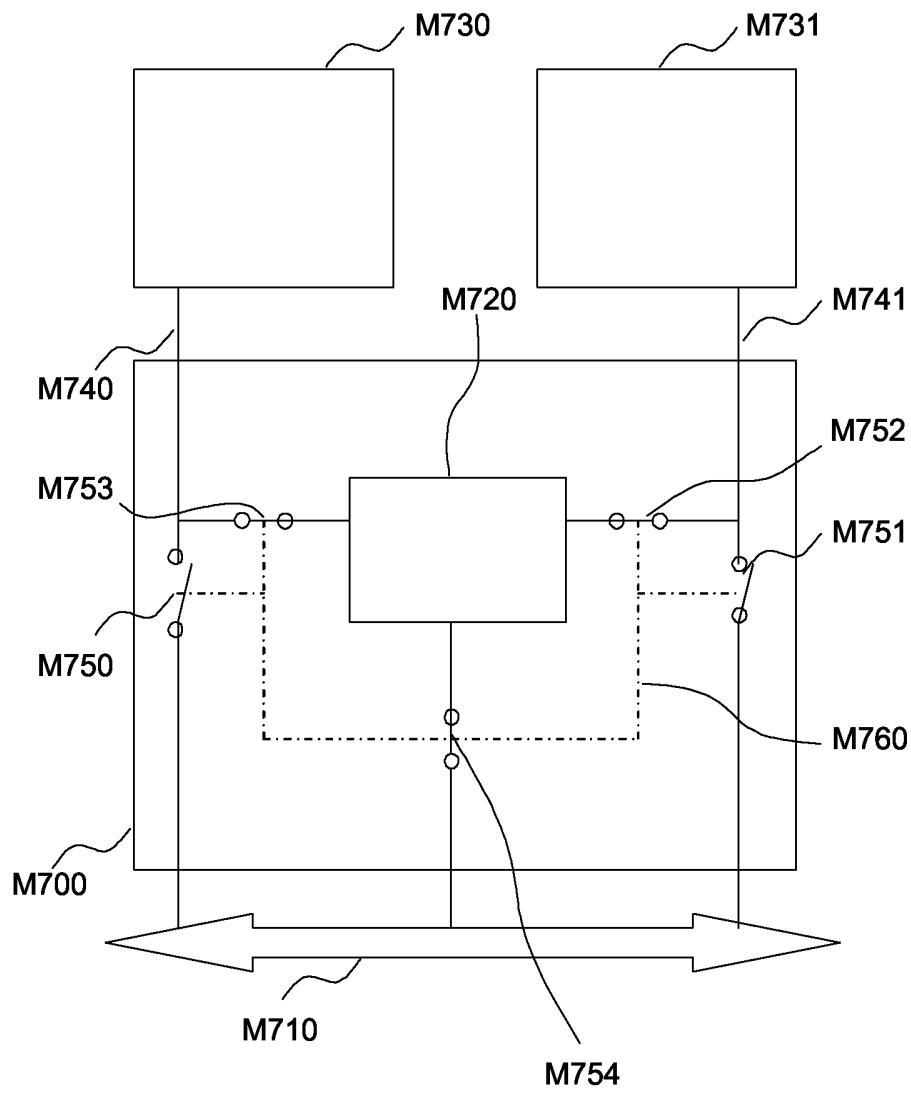
Figur 12



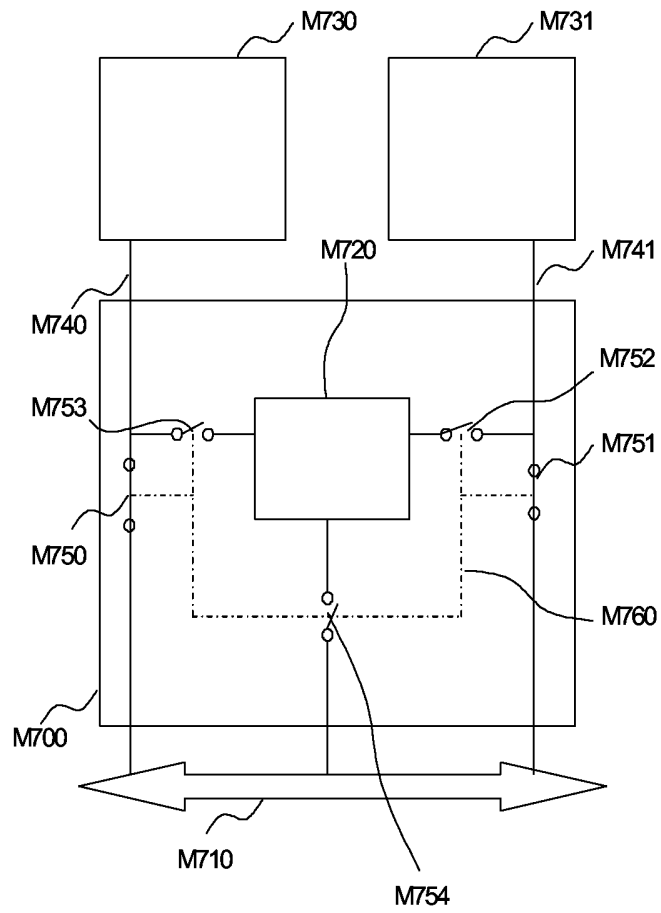
Figur 13



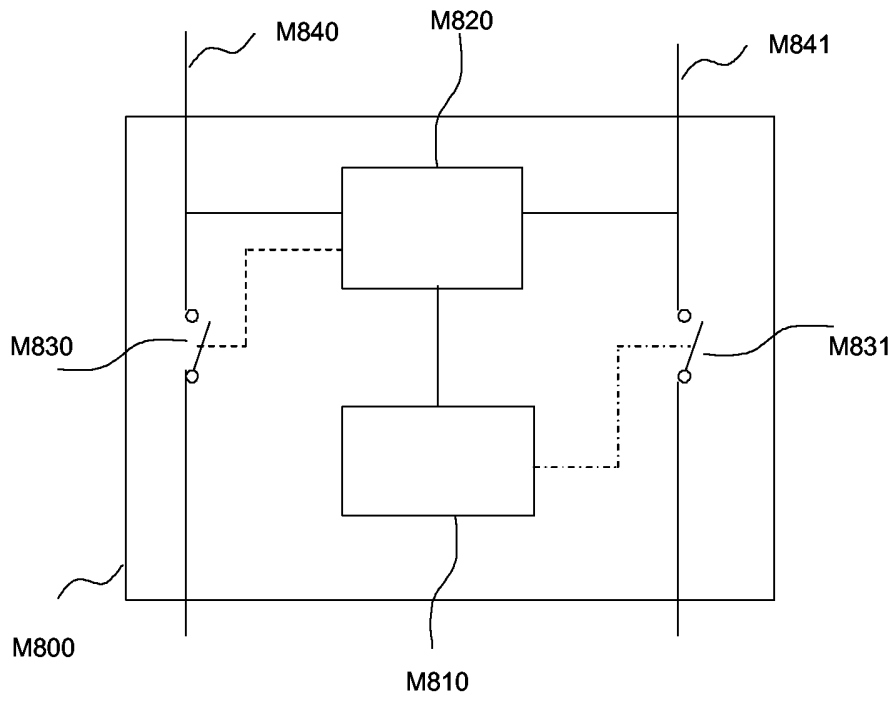
Figur 14



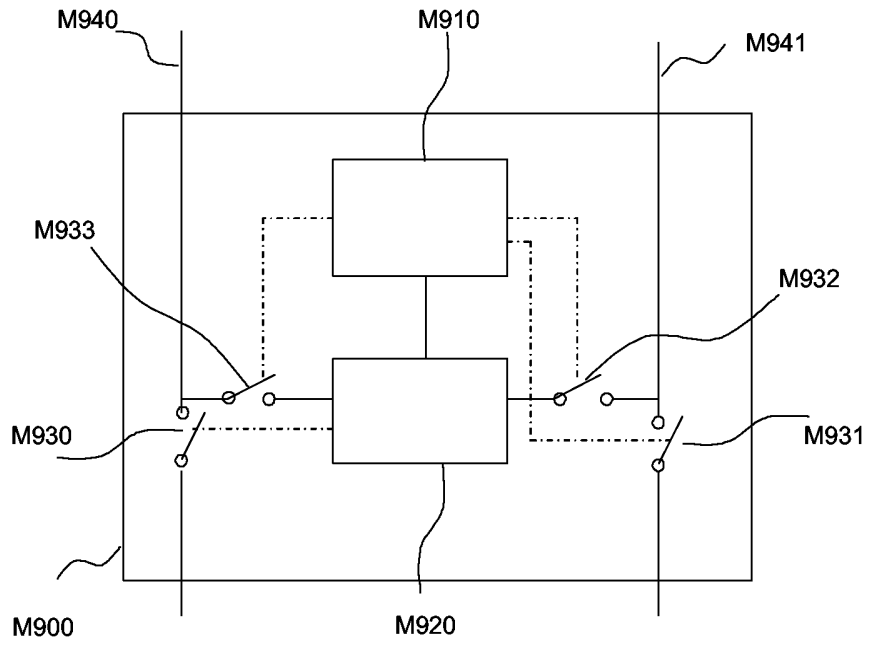
Figur 15



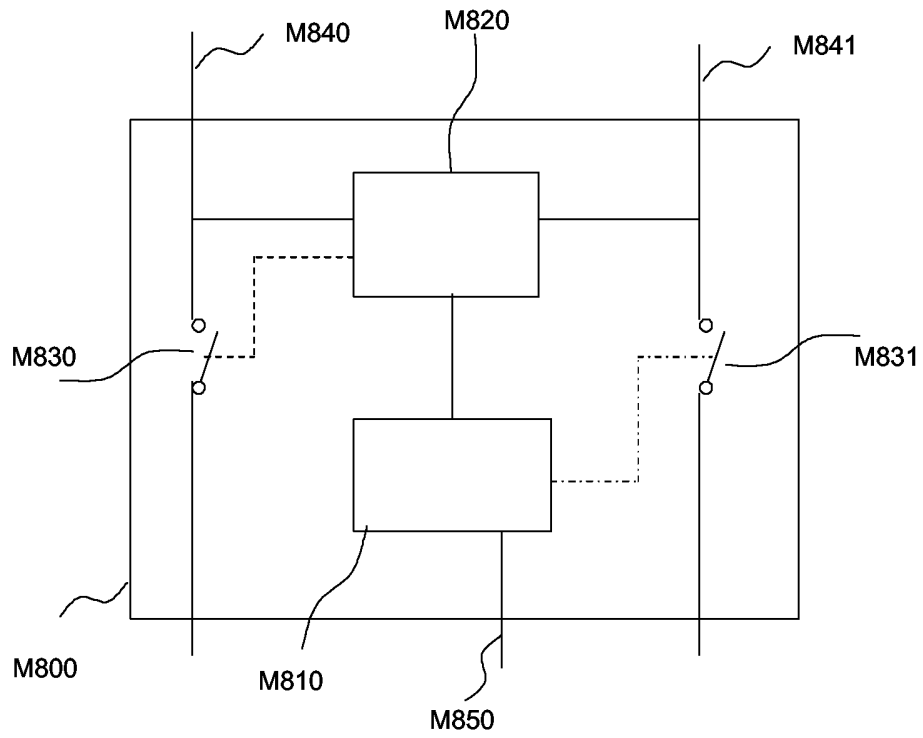
Figur 16



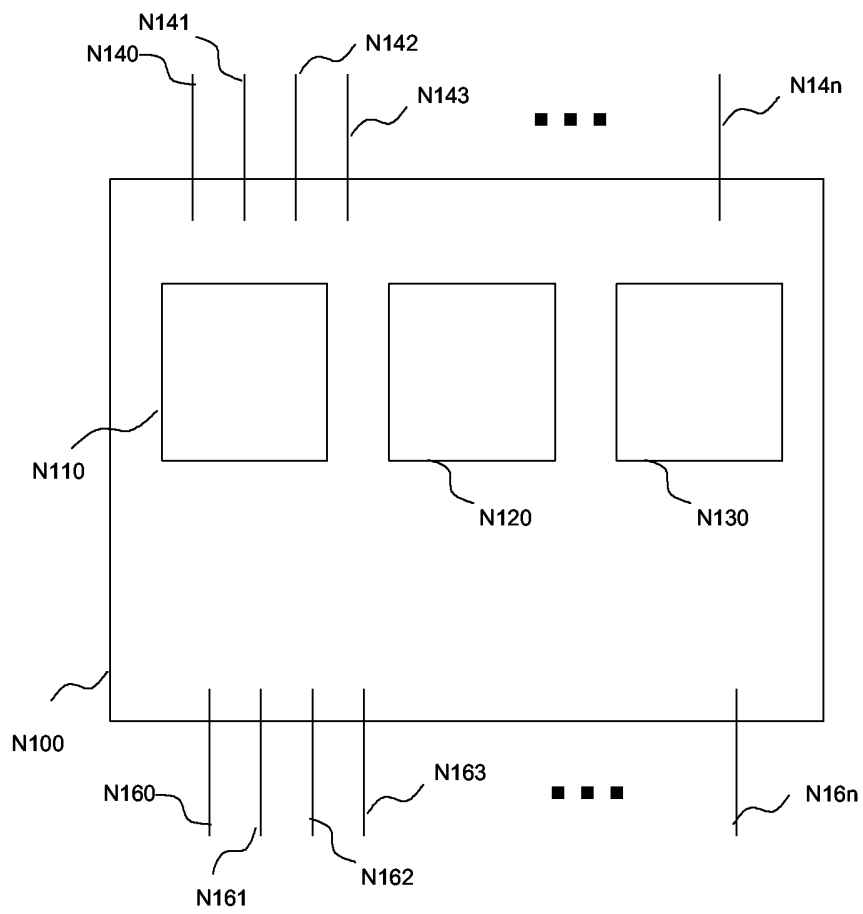
Figur 17



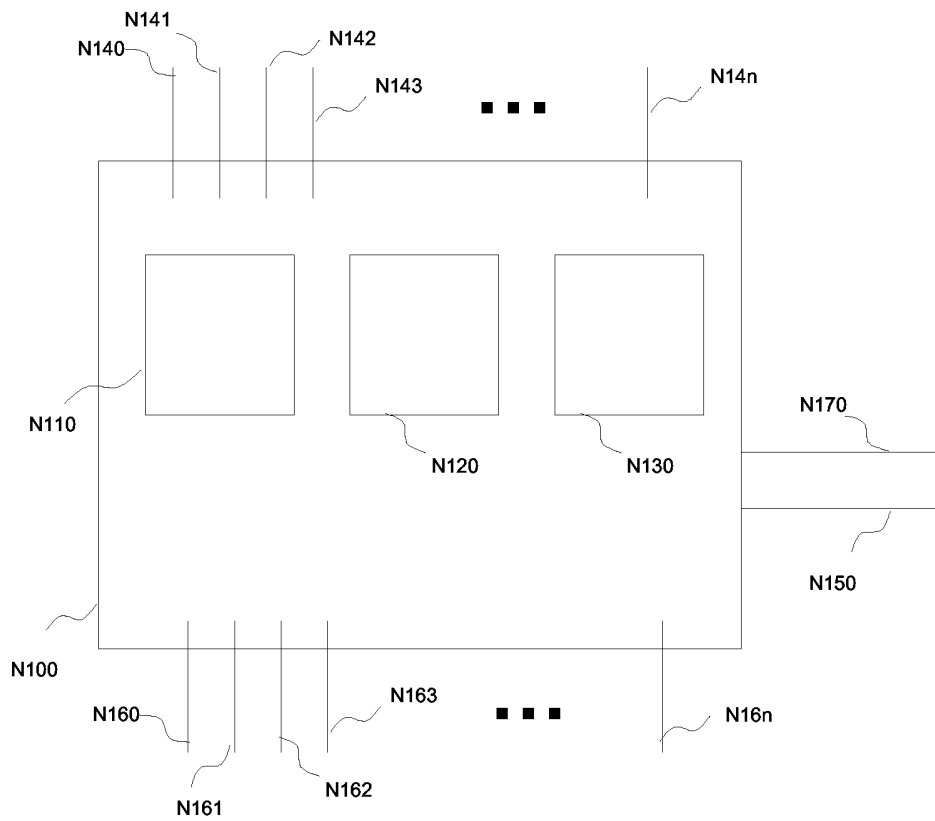
Figur 18



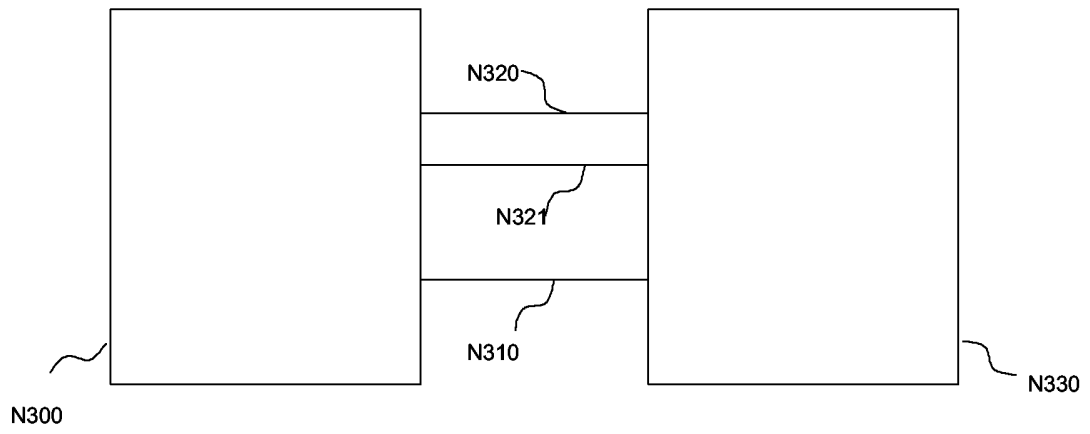
Figur 19



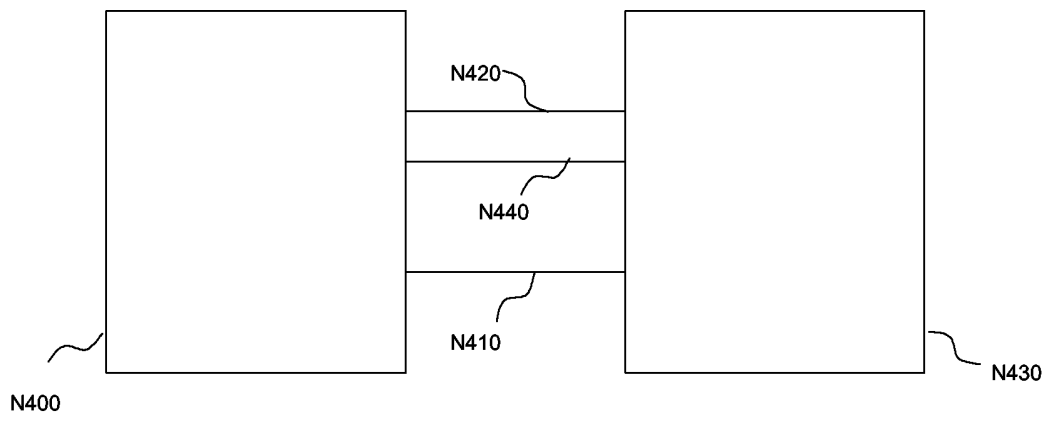
Figur 20



Figur 21



Figur 22



Figur 23

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2005/055507

A. CLASSIFICATION OF SUBJECT MATTER
G06F11/16

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 6 615 366 B1 (GROCHOWSKI EDWARD T ET AL) 2 September 2003 (2003-09-02) abstract figures 1,3a,3b column 1, line 62 - column 2, line 53 column 4, line 1 - column 6, line 11 column 7, line 1 - column 10, line 3 column 11, line 1 - column 12, line 61	1-17
Y	US 5 794 066 A (DREYER ET AL) 11 August 1998 (1998-08-11) abstract figures 1,2,5 column 1, line 28 - column 2, line 27 column 3, lines 22-27 column 5, line 40 - column 7, line 51 column 11, line 45 - column 12, line 12 ----- -/--	1-17

Further documents are listed in the continuation of Box C.

See patent family annex.

* Special categories of cited documents :

A document defining the general state of the art which is not considered to be of particular relevance

E earlier document but published on or after the international filing date

L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

O document referring to an oral disclosure, use, exhibition or other means

P document published prior to the international filing date but later than the priority date claimed

T later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

X document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

Y document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

8 document member of the same patent family

Date of the actual completion of the international search

27 February 2006

Date of mailing of the international search report

07/03/2006

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Weber, V

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2005/055507

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 6 256 658 B1 (MOUREY JEAN-CHARLES D ET AL) 3 July 2001 (2001-07-03) abstract column 1, lines 30-38 column 2, lines 47-64 column 3, lines 30-38 column 7, lines 7-62 column 8, line 66 - column 9, line 41 column 9, lines 26-34 -----	1-17
A	US 2004/186979 A1 (JANKE MARCUS ET AL) 23 September 2004 (2004-09-23) paragraphs '0008! - '0010!, '0017!, '0018!, '0020!, '0031!, '0032!, '0034! -----	1-17
P, X	WO 2005/045664 A (ROBERT BOSCH GMBH; WEIBERLE, REINHARD; KOTTKE, THOMAS; STEININGER, AND) 19 May 2005 (2005-05-19) pages 1-3 page 6, lines 4-9 page 8, lines 8-18 page 11, line 19 - page 13 -----	1-17

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2005/055507

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6615366	B1	02-09-2003	NONE
US 5794066	A	11-08-1998	NONE
US 6256658	B1	03-07-2001	AT 171554 T 15-10-1998 AU 5801294 A 19-07-1994 CA 2152041 A1 07-07-1994 DE 69321255 D1 29-10-1998 DE 69321255 T2 27-05-1999 EP 0674783 A1 04-10-1995 JP 3508775 B2 22-03-2004 JP 8506195 T 02-07-1996 JP 3705547 B2 12-10-2005 JP 2004046881 A 12-02-2004 JP 2005149532 A 09-06-2005 WO 9415283 A1 07-07-1994 US 5452456 A 19-09-1995
US 2004186979	A1	23-09-2004	AT 295975 T 15-06-2005 DE 10136335 A1 13-02-2003 WO 03010638 A1 06-02-2003 EP 1410151 A1 21-04-2004
WO 2005045664	A	19-05-2005	DE 10349581 A1 25-05-2005

INTERNATIONALER RECHERCHENBERICHT

Internationales Aktenzeichen
PCT/EP2005/055507

A. KLASSIFIZIERUNG DES ANMELDUNGSGEGENSTANDES
G06F11/16

Nach der Internationalen Patentklassifikation (IPC) oder nach der nationalen Klassifikation und der IPC

B. RECHERCHIERTE GEBIETE

Recherchierter Mindestprüfstoff (Klassifikationssystem und Klassifikationssymbole)
G06F

Recherchierte, aber nicht zum Mindestprüfstoff gehörende Veröffentlichungen, soweit diese unter die recherchierten Gebiete fallen

Während der internationalen Recherche konsultierte elektronische Datenbank (Name der Datenbank und evtl. verwendete Suchbegriffe)

EPO-Internal, WPI Data

C. ALS WESENTLICH ANGESEHENE UNTERLAGEN

Kategorie*	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
Y	US 6 615 366 B1 (GROCHOWSKI EDWARD T ET AL) 2. September 2003 (2003-09-02) Zusammenfassung Abbildungen 1, 3a, 3b Spalte 1, Zeile 62 - Spalte 2, Zeile 53 Spalte 4, Zeile 1 - Spalte 6, Zeile 11 Spalte 7, Zeile 1 - Spalte 10, Zeile 3 Spalte 11, Zeile 1 - Spalte 12, Zeile 61	1-17
Y	US 5 794 066 A (DREYER ET AL) 11. August 1998 (1998-08-11) Zusammenfassung Abbildungen 1, 2, 5 Spalte 1, Zeile 28 - Spalte 2, Zeile 27 Spalte 3, Zeilen 22-27 Spalte 5, Zeile 40 - Spalte 7, Zeile 51 Spalte 11, Zeile 45 - Spalte 12, Zeile 12	1-17
	----- -/--	



Weitere Veröffentlichungen sind der Fortsetzung von Feld C zu entnehmen



Siehe Anhang Patentfamilie

- * Besondere Kategorien von angegebenen Veröffentlichungen :
- *A* Veröffentlichung, die den allgemeinen Stand der Technik definiert, aber nicht als besonders bedeutsam anzusehen ist
- *E* älteres Dokument, das jedoch erst am oder nach dem internationalen Anmeldedatum veröffentlicht worden ist
- *L* Veröffentlichung, die geeignet ist, einen Prioritätsanspruch zweifelhaft erscheinen zu lassen, oder durch die das Veröffentlichungsdatum einer anderen im Recherchenbericht genannten Veröffentlichung belegt werden soll oder die aus einem anderen besonderen Grund angegeben ist (wie ausgeführt)
- *O* Veröffentlichung, die sich auf eine mündliche Offenbarung, eine Benutzung, eine Ausstellung oder andere Maßnahmen bezieht
- *P* Veröffentlichung, die vor dem internationalen Anmeldedatum, aber nach dem beanspruchten Prioritätsdatum veröffentlicht worden ist
- *T* Spätere Veröffentlichung, die nach dem internationalen Anmeldedatum oder dem Prioritätsdatum veröffentlicht worden ist und mit der Anmeldung nicht kollidiert, sondern nur zum Verständnis des der Erfindung zugrundeliegenden Prinzips oder der ihr zugrundeliegenden Theorie angegeben ist
- *X* Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann allein aufgrund dieser Veröffentlichung nicht als neu oder auf erfinderischer Tätigkeit beruhend betrachtet werden
- *Y* Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann nicht als auf erfinderischer Tätigkeit beruhend betrachtet werden, wenn die Veröffentlichung mit einer oder mehreren anderen Veröffentlichungen dieser Kategorie in Verbindung gebracht wird und diese Verbindung für einen Fachmann naheliegend ist
- * & * Veröffentlichung, die Mitglied derselben Patentfamilie ist

Datum des Abschlusses der internationalen Recherche	Absenddatum des internationalen Recherchenberichts
27. Februar 2006	07/03/2006

Name und Postanschrift der Internationalen Recherchenbehörde Europäisches Patentamt, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Bevollmächtigter Bediensteter Weber, V
---	---

INTERNATIONALER RECHERCHENBERICHT

Internationales Aktenzeichen
PCT/EP2005/055507

C. (Fortsetzung) ALS WESENTLICH ANGESEHENE UNTERLAGEN		
Kategorie*	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
A	US 6 256 658 B1 (MOUREY JEAN-CHARLES D ET AL) 3. Juli 2001 (2001-07-03) Zusammenfassung Spalte 1, Zeilen 30-38 Spalte 2, Zeilen 47-64 Spalte 3, Zeilen 30-38 Spalte 7, Zeilen 7-62 Spalte 8, Zeile 66 - Spalte 9, Zeile 41 Spalte 9, Zeilen 26-34 -----	1-17
A	US 2004/186979 A1 (JANKE MARCUS ET AL) 23. September 2004 (2004-09-23) Absätze '0008! - '0010!, '0017!, '0018!, '0020!, '0031!, '0032!, '0034! -----	1-17
P,X	WO 2005/045664 A (ROBERT BOSCH GMBH; WEIBERLE, REINHARD; KOTTKE, THOMAS; STEININGER, AND) 19. Mai 2005 (2005-05-19) Seiten 1-3 Seite 6, Zeilen 4-9 Seite 8, Zeilen 8-18 Seite 11, Zeile 19 - Seite 13 -----	1-17

INTERNATIONALER RECHERCHENBERICHT

Angaben zu Veröffentlichungen, die zur selben Patentfamilie gehören

Internationales Aktenzeichen

PCT/EP2005/055507

Im Recherchenbericht angeführtes Patentdokument	Datum der Veröffentlichung	Mitglied(er) der Patentfamilie	Datum der Veröffentlichung
US 6615366	B1	02-09-2003	KEINE
US 5794066	A	11-08-1998	KEINE
US 6256658	B1	03-07-2001	AT 171554 T 15-10-1998 AU 5801294 A 19-07-1994 CA 2152041 A1 07-07-1994 DE 69321255 D1 29-10-1998 DE 69321255 T2 27-05-1999 EP 0674783 A1 04-10-1995 JP 3508775 B2 22-03-2004 JP 8506195 T 02-07-1996 JP 3705547 B2 12-10-2005 JP 2004046881 A 12-02-2004 JP 2005149532 A 09-06-2005 WO 9415283 A1 07-07-1994 US 5452456 A 19-09-1995
US 2004186979	A1	23-09-2004	AT 295975 T 15-06-2005 DE 10136335 A1 13-02-2003 WO 03010638 A1 06-02-2003 EP 1410151 A1 21-04-2004
WO 2005045664	A	19-05-2005	DE 10349581 A1 25-05-2005