(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 2006/045802 A2

(51) Internationale Patentklassifikation:
G06F 9/318 (2006.01)

(21) Internationales Aktenzeichen: PCT/EP2005/055539

(22) Internationales Anmeldedatum:
25. Oktober 2005 (25.10.2005)

(25) Einreichungssprache: Deutsch

(26) Veröffentlichungssprache: Deutsch

(30) Angaben zur Priorität:
10 2004 051 952.8
25. Oktober 2004 (25.10.2004) DE
10 2004 051 964.1
25. Oktober 2004 (25.10.2004) DE
10 2004 051 950.1
25. Oktober 2004 (25.10.2004) DE
10 2004 051 937.4
25. Oktober 2004 (25.10.2004) DE
10 2004 051 992.7
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)*: **KOTTKE, Thomas** [DE/DE]; Leimentalstrasse 13/1, 71139 Ehningen (DE).

(74) Gemeinsamer Vertreter: **ROBERT BOSCH GMBH**; Postfach 30 02 20, 70442 Stuttgart (DE).

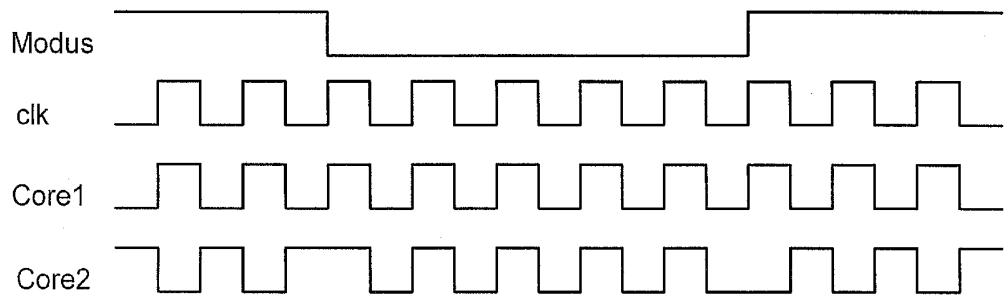
(81) Bestimmungsstaaten *(soweit nicht anders angegeben, für jede verfügbare nationale Schutzrechtsart)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, 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.

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

[Fortsetzung auf der nächsten Seite]

(54) **Title:** METHOD AND DEVICE FOR CARRYING OUT CLOCK CHANGEOVER IN A MULTIPROCESSOR SYSTEM

(54) **Bezeichnung:** VERFAHREN UND VORRICHTUNG ZUR TAKTUMSCHALTUNG BEI EINEM MEHRPROZESSORSYSTEM



(57) **Abstract:** The invention relates to a unit and method for carrying out clock changeover in a system comprising at least two arithmetic units. To this end, changeover means are contained in the unit and enable switching between at least two different operating modes of the system. In at least one arithmetic unit, a clock changeover can be activated or deactivated during a switching between the operating modes.

(57) **Zusammenfassung:** Einheit und Verfahren zur Taktumschaltung in einem System mit wenigstens zwei Recheneinheiten, wobei Umschaltmittel enthalten sind durch welche zwischen wenigstens zwei Betriebsmodi des Systems umgeschalten werden kann, wobei bei wenigstens einer Recheneinheit bei einer Umschaltung des Betriebsmodus eine Taktumschaltung aktivierbar oder deaktivierbar ist.

WO 2006/045802 A2

Veröffentlicht:

— *ohne internationalen Recherchenbericht und erneut zu veröffentlichen nach Erhalt des Berichts*

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.

Verfahren und Vorrichtung zur Taktumschaltung bei einem Mehrprozessorsystem

Stand der Technik

In technischen Anwendungen, wie insbesondere im Kraftfahrzeug oder im Industriegüterbereich also z.B. Maschinenbereich und in der Automatisierung werden ständig mehr und mehr mikroprozessor- oder rechnerbasierte Steuerungs- und Regelungssysteme für sicherheitskritische Anwendungen eingesetzt. Dabei sind Zweirechnersysteme oder Zweiprozessorsysteme (Dual Cores) heutzutage gängige Rechnersysteme für sicherheitskritische Anwendungen, insbesondere im Fahrzeug wie beispielsweise für Antiblockiersysteme, das Elektronische Stabilitätsprogramm (ESP), X-by-Wire-Systeme wie Drive-by-Wire oder Steer-by-Wire sowie Break-by-Wire, usw. oder auch bei sonstigen vernetzten Systemen. Um diese hohen Sicherheitsansprüche in zukünftigen Anwendungen zu befriedigen, sind mächtige Fehlerentdeckungsmechanismen und Fehlerbehandlungsmechanismen erforderlich, insbesondere um transienten Fehler, die beispielsweise bei Verkleinerung der Halbleiterstrukturen der Rechnersysteme entstehen, zu begegnen. Dabei ist es relativ schwierig den Core selbst, also den Prozessor zu schützen. Eine Lösung hierfür ist wie erwähnt die Verwendung eines Zweirechnersystems oder Dual Core-Systems zur Fehlerdetektion.

Solche Prozessoreinheiten mit wenigstens zwei integrierten Ausführungseinheiten sind somit als Dual-Core- oder Multi-Core-Architekturen bekannt. Solche Dual-Core- oder Multi-Core-Architekturen werden nach heutigem Stand der Technik hauptsächlich aus zwei Gründen vorgeschlagen:

Zum Einen kann damit eine Leistungssteigerung, also eine Performance-Steigerung erreicht werden, indem die beiden Ausführungseinheiten oder Cores als zwei Recheneinheiten auf einem Halbleiterbaustein betrachtet und behandelt werden. In dieser Konfiguration bearbeiten die zwei Ausführungseinheiten oder Cores unterschiedliche Programme respektive Tasks. Dadurch lässt sich eine Leistungssteigerung erzielen, weshalb diese Konfiguration als Leistungsmodus oder Performance-Mode bezeichnet wird.

Der zweite Grund, eine Dual-Core- oder Multi-Core-Architektur zu realisieren, ist eine Sicherheitssteigerung, indem die beiden Ausführungseinheiten redundant das gleiche Programm abarbeiten. Die Ergebnisse der beiden Ausführungseinheiten oder CPUs, also Cores werden verglichen und ein Fehler kann bei dem Vergleich auf Übereinstimmung erkannt werden. Im Folgenden wird diese Konfiguration als Sicherheitsmodus oder Safety-Mode oder auch Fehlererkennungsmodus bezeichnet.

Heutzutage gibt es somit einerseits Zwei- oder Mehrprozessorsysteme die zur Erkennung von Hardware-Fehlern redundant arbeiten (siehe Dual-Core oder Master-Checker-Systeme) und anderseits Zwei- oder Mehrprozessorsysteme, die auf ihren Prozessoren unterschiedliche Daten abarbeiten. Kombiniert man nun diese beiden Betriebsarten in einem Zwei- oder Mehrprozessorsystem (der Einfachheit halber wird nun nur noch von einem Zweiprozessorsystem gesprochen, die nachfolgende Erfindung ist aber genauso auf Mehrprozessorsystemen anwendbar), so müssen die beiden Prozessoren im Performance-Modus unterschiedliche Daten erhalten und im Fehlererkennungsmodus die gleichen Daten.

Die Aufgabe der Erfindung ist nun eine Einheit und eine Verfahren vorzustellen, welche die Phase der Takte bzw. auch den Takt der beiden Prozessoren zueinander umschaltet in Abhängigkeit von dem Prozessormodus als Schutz gegen Common Cause Fehler.
Solch eine Einheit ist bis jetzt noch nicht bekannt. Sie ermöglicht den effektiven sicheren Betrieb eines Zweiprozessorsystems, so dass in den beiden Modi Sicherheit und Performance im Betrieb umgeschaltet werden kann. Dabei wird im weiteren von Prozessoren gesprochen, was aber ebenso Cores bzw. Recheneinheiten begrifflich einschließt.

Weiterhin ist es Aufgabe der Erfindung ein Verfahren und eine Vorrichtung anzugeben, durch welche eine Optimierung der Funktion im Rahmen der Umschaltung zwischen den Betriebsmodi ermöglicht wird.

Beschreibung der Ausführungsbeispiele und Vorteile der Erfindung

In einem Zweirechnersystem gibt es 2 Prozessoren die dieselben oder verschiedene Aufgaben abarbeiten können. Diese beiden Prozessoren des Zweirechnersystems können diese Aufgaben taktsynchron oder taktversetzt abarbeiten. Wird ein Zweiprozessorsystem zur Fehlerentdeckung aufgebaut, ist es vorteilhaft zur Vermeidung von Common-Mode Fehlern, dass diese beide Prozessoren mit einem Taktversatz arbeiten. Am effektivsten ist diese Methode wenn ein nicht ganzzahliger Taktversatz gewählt wird. Arbeiten die beiden Prozessoren verschiedene Aufgaben ab, ist es vorteilhafter sie taktflankensynchron laufen zu lassen, da die externen Komponenten wie Speicher nur mit dem Takt eines Prozessors angesteuert werden können. Soll nun ein zwischen diesen beiden Modi umschaltbares Zweiprozessorsystem eingesetzt werden, ist es somit nur auf einen Betriebsmodi optimiert.

Erfindungsgemäß wird dies dadurch kompensiert, dass in dem Zweiprozessorsystem (bzw. Mehrprozessorsystem), welches umschaltbar ist zwischen 2 Modi wie Sicherheit und Performance, die beiden Prozessoren im Modus Sicherheit mit einem Taktversatz arbeiten und im Modus Performance ohne Taktversatz. Im Modus Performance ist kein Taktversatz vorteilhaft, da die externen Komponenten wie Speicher meistens mit einer niedrigeren Taktfrequenz betrieben werden und von der Taktflanke nur auf einen Prozessor passend ausgelegt sind. Der zweite taktversetzte Prozessor hätte sonst bei jedem Speicherzugriff einen Wartezyklus, da er die externe Komponente um einen halben Takt zu spät ansteuert.

Durch eine Taktumschaltung für ein Zweiprozessorsystem wird im Modus Sicherheit das Optimum bei der Fehlererkennung herausgeholt und im Modus Performance das Maximum an der Performance.

Somit geht die Erfindung vorteilhafter Weise von einer Einheit zur Taktumschaltung in einem System mit wenigstens zwei Recheneinheiten aus, sowie einem entsprechenden System mit einer solchen Einheit, wobei Umschaltmittel (ModeSwitch) enthalten sind

durch welche zwischen wenigstens zwei Betriebsmodi des Systems umgeschalten werden kann, wobei die Einheit derart ausgestaltet ist, dass bei wenigstens einer Recheneinheit bei einer Umschaltung des Betriebsmodus eine Taktumschaltung erfolgt.

Ebenso ist ein Verfahren zur Taktumschaltung in einem System mit wenigstens zwei Recheneinheiten gezeigt, wobei Umschaltmittel enthalten sind durch welche zwischen wenigstens zwei Betriebsmodi des Systems umgeschalten werden kann, wobei bei wenigstens einer Recheneinheit bei einer Umschaltung des Betriebsmodus eine Taktumschaltung erfolgt.

In einem Modus arbeiten die beiden Prozessoren in einem Taktversatz. Dieser kann sowohl um ganze Takte als auch um Teile des Taktes gegeneinander verschoben sein. Eine weitere Variante ist, dass in den beiden Modi eine unterschiedliche Taktfrequenz verwendet wird. Im sicherheitskritischen Modus kann zur Störungsunterdrückung z.B. ein niedrigerer Takt verwendet werden als im Performancemodus. Dabei können diese beiden Varianten auch miteinander kombiniert werden.

Dabei entspricht der erste Betriebsmodus einem Sicherheitsmodus, bei dem die zwei Recheneinheiten gleiche Programme und/oder Daten abarbeiten und Vergleichsmittel vorgesehen sind, welche die bei der Abarbeitung der gleichen Programme entstehenden Zustände auf Übereinstimmung vergleichen.

Die erfindungsgemäße Einheit bzw. das erfindungsgemäße Verfahren ermöglicht die optimierte Implementierung der beiden Modi in einem Zweiprozessorsystem.

Arbeiten die beiden Prozessoren im Fehlererkennungsmodus (F-Modus), so erhalten die beiden Prozessoren die gleichen Daten/Instruktionen und arbeiten sie im Performancemodus (P-Modus), so kann jeder Prozessor auf den Speicher zugreifen. Dann verwaltet diese Einheit die Zugriffe auf den nur einfach vorhandenen Speicher oder Peripherie.

Im F-Modus übernimmt die Einheit die Daten/Adressen eines Prozessors (hier Master genannt) und leitet diese an die Komponenten wie Speicher, Bus, usw. weiter. Der zweite Prozessor (hier Slave) möchte den gleichen Zugriff machen. Die Datenverteilungseinheit nimmt dies an einem zweiten Port entgegen, aber leitet die Anfrage nicht an die weiteren

Komponenten weiter. Die Datenverteilungseinheit übergibt dem Slave die gleichen Daten wie dem Master und vergleicht die Daten der beiden Prozessoren. Sind diese unterschiedlich, so zeigt dies die Datenverteilungseinheit (hier DVE) durch ein Fehlersignal an. Es arbeitet somit nur der Master auf den Bus/Speicher und der Slave bekommt die selben Daten (Funktionsweise wie bei einem Dual-Core System).

Im P-Modus arbeiten die beiden Prozessoren unterschiedliche Programmteile ab. Die Speicherzugriffe sind somit auch unterschiedlich. Die DVE nimmt somit die Anforderung der Prozessoren entgegen und gibt die Ergebnisse/angeforderte Daten an den Prozessor zurück, der sie angefordert hat. Möchten nun beide Prozessoren gleichzeitig auf eine Komponenten zugreifen, so wird ein Prozessor in einen Wartezustand versetzt, bis der andere bedient wurde.

Die Umschaltung zwischen den beiden Modi und somit der unterschiedlichen Arbeitsweise der Datenverteilungseinheit erfolgt durch ein Steuersignal. Dies kann entweder von einem der beiden Prozessoren generiert werden oder extern.

Wird das Zweiprozessorsystem im F-Modus mit einem Taktversatz betrieben und im P-Modus nicht, so verzögert die DVE—Einheit die Daten für den Slave entsprechend, bzw. speichert die Ausgangsdaten des Master solange, bis sie mit den Ausgangsdaten des Slave zur Fehlererkennung verglichen werden können.

Der Taktversatz wird anhand der Figur 1 näher erläutert:

Figur 1 zeigt ein Zweirechnersystem mit einem ersten Rechner 100, insbesondere einem Masterrechner und einem zweiten Rechner 101, insbesondere einem Slave-Rechner. Das gesamte System wird dabei mit einem vorgebbaren Takt bzw. in vorgebbaren Taktzyklen (clock cycle) CLK betrieben. Über den Takteingang CLK1 des Rechners 100 sowie über den Takteingang CLK2 des Rechners 101 wird diesem der Takt zugeführt. Bei diesem Zweirechnersystem ist darüber hinaus beispielhaft ein spezielles Merkmal zur Fehlererkennung enthalten, in dem nämlich der erste Rechner 100 sowie der zweite Rechner 101 mit einem Zeitversatz, insbesondere einem vorgebbaren Zeitversatz bzw. einem vorgebbaren Taktversatz arbeiten. Dabei ist jede beliebige Zeit für einen Zeitversatz vorgebbar und auch jeder beliebige Takt bezüglich eines Versatzes der Taktzyklen. Dies kann ein ganzzahliger Versatz des Taktzyklus (clock cycle) sein, aber

eben auch wie in diesem Beispiel dargestellt, beispielsweise ein Versatz von 1,5 Taktzyklen, wobei hier der erste Rechner 100 eben 1,5 Taktzyklen vor dem zweiten Rechner 101 arbeitet respektive betrieben wird. Durch diesen Versatz kann vermieden werden, dass Gleichtaktfehler, sogenannte common mode failures, die Rechner oder Prozessoren, also die Cores des Dual Cores Systems, gleichartig stören und damit unerkannt bleiben. D.h. solche Gleichtaktfehler betreffen durch den Versatz die Rechner zu unterschiedlichen Zeitpunkten im Programmablauf und bewirken demnach unterschiedliche Effekte bezüglich der beiden Rechner wodurch Fehler erkennbar werden. Gleichartige Fehlerwirkungen ohne Taktversatz wären u.U. in einem Vergleich nicht erkennbar, dies wird dadurch vermieden. Um diesen Versatz bezüglich der Zeit oder des Taktes, hier insbesondere 1,5 Taktzyklen im Zweirechnersystem zum implementieren sind die Versatzbausteine 112 bis 115 implementiert.

Um die genannten Gleichtaktfehler zu erkennen ist dieses System eben beispielsweise dazu ausgelegt in einem vorgegebenen Zeitversatz oder Taktzyklenversatz zu arbeiten, insbesondere hier 1,5 Taktzyklen, d.h. während der eine Rechner, z. B. Rechner 100 direkt die Komponenten, insbesondere die externen Komponenten 103 und 104 anspricht, arbeitet der zweite Rechner 101 mit einer Verzögerung von genau 1,5 Taktzyklen dazu. Um in diesem Fall die gewünschte Eineinhalbzyklusverzögerung, also von 1,5 Taktzyklen zu erzeugen wird Rechner 101 mit der invertierten Clock, also dem invertierten Takt am Takteingang CLK2 gespeist. Dadurch müssen aber auch die vorgenannten Anschlüsse des Rechners also seine Daten bzw. Befehle über die Busse um die genannten Taktzyklen, also hier insbesondere 1,5 Taktzyklen verzögert werden, wozu eben wie gesagt die Versatz- oder Verzögerungsbausteine 112 bis 115 vorgesehen sind. Neben den beiden Rechnern oder Prozessoren 100 und 101 sind Komponenten 103 und 104 vorgesehen, die über Busse 116, bestehend aus den Busleitungen 116A und 116B und 116C sowie 117, bestehend aus den Busleitungen 117A und 117B mit den beiden Rechnern 100 und 101 in Verbindung stehen. 117 ist dabei ein Befehlsbus, bei welchem mit 117A ein Befehlsadressbus und mit 117B der Teil-Befehls(daten)bus bezeichnet ist. Der Adressbus 117A ist über einen Befehlsadressanschluss IA1 (Instruction Adress 1) mit Rechner 100 und über einen Befehlsadressanschluss IA2 (Instruction Adress 2) mit Rechner 101 verbunden. Die Befehle selbst werden über den Teil-Befehlsbus 117B übertragen, der über einen Befehlsanschluss I1 (Instruction 1) mit Rechner 100 und über einen Befehlsanschluss I2 (Instruction 2) mit Rechner 101 verbunden ist. In diesem

Befehlsbus 117 bestehend aus 117A und 117B ist eine Komponente 103 z. B. ein Befehlsspeicher, insbesondere ein sicherer Befehlsspeicher oder dergleichen zwischengeschaltet. Auch diese Komponente, insbesondere als Befehlsspeicher wird in diesem Beispiel mit dem Takt CLK betrieben. Daneben ist mit 116 ein Datenbus dargestellt, welcher einen Datenadressbus oder eine Datenadressleitung 116A und einen Datenbus oder eine Datenleitung 116B enthält. Dabei ist 116A, also die Datenadressleitung, über einen Datenadressanschluss DA1 (Data Adress 1) mit dem Rechner 100 und über einen Datenadressanschluss DA2 (Data Adress 2) mit Rechner 101 verbunden. Ebenso ist der Datenbus oder die Datenleitung 116B über einen Datenanschluss DO1 (Data Out 1) und einen Datenanschluss DO2 (Data Out 2) mit Rechner 100 bzw. Rechner 101 verbunden. Weiterhin zu Datenbus 116 gehört die Datenbusleitung 116C, welche über einen Datenanschluss DI1 (Data In 1) und einen Datenanschluss DI2 (Data In 2) jeweils mit Rechner 100 bzw. Rechner 101 verbunden ist. In diesem Datenbus 116 bestehend aus den Leitungen 116A, 116B und 116C ist eine Komponente 104 zwischengeschaltet, beispielsweise ein Datenspeicher, insbesondere ein sicherer Datenspeicher o. ä . Auch diese Komponente 104 wird in diesem Beispiel mit dem Takt CLK versorgt.

Dabei stehen die Komponenten 103 und 104 stellvertretend für beliebige Komponenten die über einen Datenbus und/oder Befehlsbus mit den Rechnern des Zweirechnersystems verbunden sind und entsprechend der Zugriffe über Daten und/oder Befehle des Zweirechnersystems bezüglich Schreiboperationen und/oder Leseoperationen fehlerhafte Daten und/oder Befehle erhalten oder abgeben können. Zur Fehlervermeidung sind zwar Fehlerkennungsgeneratoren 105, 106 und 107 vorgesehen welche eine Fehlerkennung erzeugen wie beispielsweise ein Parity-Bit oder auch einen anderen Fehlercode wie beispielsweise einen Error- Correction-Code, also ECC, o. ä.. Dazu vorgesehen sind dann auch die entsprechenden Fehlerkennungsprüfeinrichtungen oder Check-Einrichtungen 108 und 109 zur Überprüfung der jeweiligen Fehlerkennung also beispielsweise des Parity-Bit oder eines anderen Fehlercodes wie ECC.

Der Vergleich der Daten und/oder Befehle bezüglich der redundanten Ausführung im Zweirechnersystem erfolgt in den Vergleichern oder Komparatoren 110 und 111 wie in Figur 1 dargestellt. Existiert nun aber ein Zeitversatz, insbesondere ein Takt- oder Taktzyklusversatz zwischen den Rechnern 100 und 101, entweder hervorgerufen durch

ein nichtsynchrones Zweiprozessorsystem oder bei einem synchronen Zweiprozessorsystem durch Fehler in der Synchronisierung oder auch wie in diesem speziellen Beispiel durch einen zur Fehlererkennung gewünschten Zeit- bzw. Taktzyklusversatz, insbesondere hier von 1,5 Taktzyklen, so kann in diesem Zeit- oder Taktversatz ein Rechner hier insbesondere Rechner 100 fehlerhafte Daten und/oder Befehle in Komponenten, insbesondere externe Komponenten wie z. B. hier insbesondere die Speicher 103 oder 104, aber auch bezüglich anderen Teilnehmern oder Aktuatoren oder Sensoren schreiben oder lesen. So kann er auch in fehlerhafter Weise einen Schreibzugriff anstatt eines vorgesehenen Lesezugriffs durch diesen Taktversatz durchführen. Diese Szenarien führen selbstverständlich zu Fehlern im gesamten System, insbesondere ohne klare Anzeigemöglichkeit welche Daten und/oder Befehle gerade fehlerhaft geändert wurden, wodurch auch die Recovery-Problematik entsteht.

Um diese Problematik zu lösen wird nun eine Verzögerungseinheit 102 wie dargestellt in die Leitungen des Datenbusses und/oder in den Befehlsbus geschaltet. Aus Gründen der Übersichtlichkeit ist nur die Einschaltung in den Datenbus dargestellt. Bezüglich des Befehlsbusses ist dies natürlich genauso möglich und denkbar. Diese Verzögerungseinheit 102 oder die Delay Unit verzögert die Zugriffe, hier insbesondere die Speicherzugriffe so, dass ein möglicher Zeit- oder Taktversatz kompensiert wird, insbesondere bei einer Fehlererkennung beispielsweise über die Komparatoren 110 und 111 z.B. mindestens solange, bis das Fehlersignal im Zweirechnersystem erzeugt ist, also die Fehlererkennung im Zweirechnersystem durchgeführt ist. Dabei können verschiedene Varianten implementiert sein:
Verzögerung der Schreib- und Leseoperationen, Verzögerung nur der Schreiboperationen oder auch, wenn auch nicht bevorzugt, eine Verzögerung der Leseoperationen. Dabei kann durch ein Änderungssignal, insbesondere das Fehlersignal, eine verzögerte Schreiboperation in eine Leseoperation gewandelt werden um fehlerhaftes Schreiben zu unterbinden.

Nachfolgend anhand Figur 2 nun eine beispielhafte Implementierung bezüglich der Datenverteilungseinheit (DVE), die sich vorzugsweise aus einer Vorrichtung zur Detektierung des Umschaltwunsches (durch IIIOPDetect), der Mode-Switch-Einheit sowie dem Iram- und Dram-Control-Baustein zusammensetzt:

IllOpDetect: Die Umschaltung zwischen den beiden Modi wird durch die Einheiten "`Switch-Detect'" erkannt. Diese Einheit liegt zwischen dem Cache und dem Prozessor auf dem Instruktionsbus und schaut ob der Befehl IllOp in den Prozessor geladen wird. Wird der Befehl detektiert, so wird dieses Ereignis der Modeswitch Einheit mitgeteilt. Die "`Switch-Detect'" Einheit ist für jeden Prozessor einzeln vorhanden. Die Einheit "`Switch-Detect'" muss nicht fehlertolerant ausgeführt sein, da sie doppelt und somit redundant vorhanden ist. Andererseits ist es denkbar diese Einheit fehlertolerant und damit singulär auszuführen, bevorzugt ist aber die redundante Ausführung.

ModeSwitch: Die Umschaltung zwischen den beiden Modi wird durch die "`Switch-Detect'" Einheit getriggert. Soll eine Umschaltung vom Lock in den Split Modus erfolgen, detektieren beide "`Switch-Detect'" Einheiten die Umschaltung, da beide Prozessoren den gleichen Programmcode im Lock Modus abarbeiten. Die "`Switch-Detect'" Einheit des Prozessor 1 erkennt dies 1,5 Takte vor der "`Switch-Detect'" Einheit des Prozessors 2. Die "`Modeswitch'" Einheit hält mit Hilfe des Wait Signals den Prozessor 1 um 2 Takte an. Der Prozessor 2 wird 1,5 Takte später ebenfalls angehalten, aber nur um einen halben Takt, damit er zum Systemtakt synchronisiert wird. Anschließend wird das Status-Signal auf Split geschaltet für die weiteren Komponenten und die beiden Prozessoren arbeiten weiter. Damit die beiden Prozessoren nun unterschiedliche Tasks ausführen, müssen sie im Programmcode auseinanderlaufen. Dies erfolgt, indem direkt nach Umschalten in den Split-Modus ein Lesezugriff auf die Prozessor-ID erfolgt. Diese ausgelesene Prozessor-ID ist für jeden der beiden Prozessoren unterschiedlich. Wird nun auf eine Soll-Prozessor-ID verglichen, kann anschließend mit einem Conditional Jump Befehl der entsprechende Prozessor an eine andere Programmstelle gebracht werden. Bei einer Umschaltung vom Split-Modus in den Lock-Modus wird dies ein Prozessor bemerken, bzw. einer der beiden zuerst. Dieser Prozessor wird Programmcode ausführen, in dem der Umschaltbefehl enthalten ist. Dies wird nun durch die "`Switch-Detect'" Einheit registriert und teilt dies der Modeswitch Einheit mit. Diese hält den entsprechenden Prozessor an und teilt dem zweiten den Wunsch der Synchronisation durch einen Interrupt mit. Der zweite Prozessor erhält einen Interrupt und kann nun eine Softwareroutine zur Beendigung seines Tasks ausführen. Nun springt er ebenfalls an die Programmstelle, in der sich der Befehl zur Umschaltung befindet. Seine "`Switch-Detect'" Einheit signalisiert nun ebenfalls den Wunsch zum Moduswechsel an die Modeswitch Einheit. Zur nächsten steigenden Systemtaktflanke

wird nun das Wait Signal für den Prozessor 1 deaktiviert und 1,5 Takte später für den Prozessor 2. Nun arbeiten beide wieder mit einem Taktversatz von 1,5 Takten synchron.

Befinden sich das System im Lock Modus, so müssen beide "`Switch-Detect"' Einheiten der Modeswitch Einheit mitteilen, dass sie in den Split Modus wollen. Erfolgt der Umschaltwunsch nur von einer Einheit, so wird der Fehler von den Vergleichseinheiten erkannt, da diese von einem der beiden Prozessoren weiterhin Daten geliefert bekommen und diese nicht mit dem angehaltenen Prozessoren übereinstimmen.

Sind die beiden Prozessoren im Split Modus und einer schaltet nicht zurück in den Lock-Modus, so kann dies durch einen externen Watchdog erkannt werden. Bei einem Triggersignal für jeden Prozessor bemerkt der Watchdog dass der wartende Prozessor sich nicht mehr meldet. Ist nur ein Watchdogsignal für das Prozessorsystem vorhanden, so darf die Triggerung des Watchdogs nur im Lock-Modus erfolgen. Somit würde der Watchdog erkennen, dass die Modusumschaltung nicht erfolgte. Das Modussignal liegt als Dual-Rail Signal vor. Dabei steht "`10"' für den Lock-Modus und "`01"' für den Split-Modus. Bei "`00"' und "`11"' sind Fehler
aufgetreten.

IramControl: Der Zugriff auf den Befehlsspeicher der beiden Prozessoren wird über die IRAM Control gesteuert. Diese muss sicher ausgelegt sein, da sie ein Single Point of Failure ist. Sie besteht aus zwei Zustandsautomaten für jeden Prozessor: als je einen taktsynchronen iram1clkreset und einen asynchronen readiram1. Im sicherheitskritischen Modus überwachen sich die Zustandsautomaten der beiden Prozessoren gegenseitig und im Performancemodus arbeiten sie getrennt.

Das Nachladen der beiden Caches der Prozessoren werden durch 2 Zustandsautomaten gesteuert. Einem synchronen Zustandsautomaten iramclkreset und einem asynchronen readiram. Durch diese beiden Zustandsautomaten werden auch die Speicherzugriffe im Split-Modus verteilt. Hierbei hat Prozessor 1 die höhere Priorität. Nach einem Zugrif auf den Hauptspeicher durch Prozessor 1 bekommt nun -- wenn
beide Prozessoren wieder auf den Hauptspeicher zugreifen wollen -- Prozessor2 die Speicherzugriffserlaubnis zugeteilt. Diese beiden Zustandsautomaten sind für jeden Prozessor implementiert. Im Lock-Modus werden die Ausgangssignale der Automaten verglichen um auftretende Fehler erkennen zu können.

Die Daten zum Aktualisieren des Cache 2 im Lock-Modus werden in der IRAM-Control Einheit um 1,5 Takte verzögert.

In Bit 5 im Register 0 der SysControl wird codiert um welchen Core es sich handelt. Core 1 ist das Bit 0 und bei Core 2 ist es High. Dieses Register ist in den Speicherbereich mit der Adresse 65528 gespiegelt.

Bei einem Speicherzugriff von Core 2 wird erst überprüft in welchem Modus sich der Rechner befindet. Ist er im Lock-Modus so wird sein Speicherzugriff unterdrückt. Dieses Signal liegt als Common-Rail Signal vor, da es sicherheitskritisch ist.

Der Programmcounter des Prozessors 1 wird um 1,5 Takte verzögert um im Lock-Modus mit dem Programmcounter des Prozessors 2 verglichen werden zu können.

Im Split Modus können die Caches der beiden Prozessoren unterschiedlich nachgeladen werden. Wenn nun in den Lock-Modus umgeschaltet wird, sind die beiden Caches nicht kohärent zueinander. Dadurch können die beiden Prozessoren auseinanderlaufen und die Vergleicher signalisieren folglich einen Fehler. Um dies zu vermeiden, ist in der IRAM Control eine Flag Tabelle aufgebaut. In dieser wird vermerkt, ob eine Cachezeile im Lock- oder im Split-Modus geschrieben wurde. Im Lock-Modus wird der für die Cachezeile entsprechende Eintrag bei einer Cachezeilennachladung auf 0 gesetzt und im Split-Modus -- auch bei einer Cacheaktualisierung der Cachezeile von nur einem Cache -- auf 1. Führt der Prozessor nun im Lock-Modus einen Speicherzugriff aus, so wird überprüft, ob diese Cachezeile im Lock-Modus aktualisiert wurde, d.h. in beiden Caches gleich ist.

ImSplit-Modus kann der Prozessor immer auf die Cachezeile zugreifen, unabhängig wie der Flag_Vector ist. Diese Tabelle muss nur einmal vorhanden sein, da bei einem Fehler die beiden Prozessoren auseinanderlaufen und somit an den Vergleichern dieser Fehler sicher erkannt wird. Da die Zugriffszeiten auf der zentralen Tabelle relativ hoch sind, kann diese Tabelle auch zu jedem Cache kopiert werden.

DramControl: In dieser Komponente werden für die Adress-, Daten- und Speichersteuersignale von jedem Prozessor das Parity gebildet.

Es gibt einen Prozess für beide Prozessor zum Sperren des Speichers. Dieser Prozess muss nicht sicher implementiert sein, da im Lock-Modus fehlerhafte Speicherzugriffe durch die Vergleicher erkannt werden und im Split-Modus keine sicherheitsrelevanten Anwendungen ausgeführt werden. Hierin wird überprüft, ob der Prozessor den Speicher für den anderen Prozessor sperren möchte. Dieses Sperren des Datenspeichers erfolgt durch einen Zugriff auf die Speicheradresse $FBFF$=64511. Dieses Signal soll genau ein Takt lang anliegen, auch wenn am Prozessor zum Zeitpunkt des Aufrufens ein wait-command anliegt. Der Zustandsautomat zur Verwaltung der Datenspeicherzugriffe besteht aus 2 Hauptzuständen:

- Prozessorstatus Lock: Die beiden Prozessoren arbeiten im Lock-Modus. D.h. die Funktionalität des Datenspeicherlocking ist nicht notwendig. Prozessor 1 koordiniert die Speicherzugriffe.
- Prozessorstatus Split: Nun ist eine Zugriffskonfliktauflösung auf den Datenspeicher nötig und ein Speichersperren muss erfolgen können.

Der Zustand im Split-Modus ist wiederum in 7 Zustände untergliedert, die die Zugriffskonflikte auflösen und den
Datenspeicher für jeweils den anderen Prozessor sperren können. Bei gleichzeitigem Wunsch der beiden Prozessoren bei einem Zugriff, stellt die aufgeführte Reihenfolge gleichzeitig die Priorisierung dar.

- Core1_Lock: Prozessor 1 hat den Datenspeicher gesperrt. Möchte in diesem Zustand Prozessor 2 auf den Speicher zugreifen, so wird er durch ein Wartesignal angehalten, bis Prozessor 1 den Datenspeicher wieder freigibt. \
- Core2_Lock: Ist der gleiche Zustand wie der vorige nur dass nun Prozessor 2 den Datenspeicher gesperrt hat und Prozessor 1 bei Datenspeicheroperationen angehalten wird.
- lock1_wait: Der Datenspeicher war durch den Prozessor 2 gesperrt als Prozessor 1 ihn ebenfalls für sich reservieren wollte. Prozessor 1 ist somit für die nächste Speichersperrung vorgemerkt.
- nex: Das gleiche für Prozessor 2. Der Datenspeicher war während des Sperrversuchs durch Prozessor 1 gesperrt. Prozessor 2 bekommt den Speicher vorreserviert. Bei normalen Speicherzugriff ohne Sperren kann hier Prozessor 2 vor Prozessor 1 zugreifen wenn davor Prozessor 1 dran war.

- Speicherzugriff von Prozessor 1: Der Speicher ist in diesem Fall nicht gesperrt. Prozessor 1 darf auf den Datenspeicher zugreifen. Falls er ihn sperren möchte, kann er dies in diesem Zustand vornehmen.
- Speicherzugriff durch Prozessor 2. Im selben Takt wollte Prozessor 1 nicht auf den Speicher zugreifen somit ist der Speicher frei für den Prozessor 2.
- kein Prozessor möchte auf den Datenspeicher zugreifen

Die DVE setzt sich wie erwähnt zusammen aus dem Detektierung des Umschaltwunsches (IllOPDetect) der ModeSwitch-Einheit und der Iram- und DramControl.

In Figur 3 ist nun die Taktumschaltung an einem Beispiel dargestellt, so dass bezüglich des einen Modus im Vergleich zum anderen Modus eine Taktumschaltung erfolgt. Dabei sind die beiden Modi, der Takt clk und die beiden Prozessor- oder Coretakte gezeigt.

In einem Modus arbeiten die beiden Prozessoren in einem Taktversatz. Dieser kann sowohl um ganze Takte als auch um Teile des Taktes gegeneinander verschoben sein. Eine weitere Variante ist, dass in den beiden Modi eine unterschiedliche Taktfrequenz verwendet wird. Im sicherheitskritischen Modus kann zur Störungsunterdrückung z.B. ein niedrigerer Takt verwendet werden als im Performancemodus. Dabei können diese beiden Varianten auch miteinander kombiniert werden.

Kern der Erfindung ist somit die modusabhängige Taktumschaltung.

Daneben löst aber auch die dargestellte spezielle Implementierung die Eingangs genannten Aufgaben.

Bei den Implementierungen von insbesondere Zweiprozessorsystemen (Dual-Core) wird für jeden Prozessor ein Cache vorgesehen wie nochmals schematisch in Figur 4 gezeigt. Ein Cache ist normalerweise nicht ausreichend, da dieser Cache räumlich gesehen zwischen den beiden Prozessoren angeordnet werden muss. Aufgrund der langen Laufzeit zwischen dem Cache und den beiden Prozessoren könnten folglich die beiden Prozessoren nur mit einer begrenzten Taktfrequenz arbeiten.

Caches dienen als schneller Zwischenspeicher, damit der Prozessor die Daten nicht immer aus dem langsamen Hauptspeicher holen muss. Um dies zu ermöglichen, muss bei

der Implementierung von Cache stark auf dessen Zugriffsdauer geachtet werden. Diese setzt sich aus der eigentlichen Zugriffszeit um die Daten aus dem Cache zu holen und aus der Zeit um die Daten an den Prozessor weiterzureichen zusammen. Ist der Cache nun räumlich weit entfernt vom Prozessor platziert, so dauert die Übermittlung der Daten sehr lange und der Prozessor kann nicht mehr mit seinem vollen Takt arbeiten. Aufgrund dieses Timingproblems kann bei diesen beispielhaften Zweiprozessorsystemen für jeden Prozessor und/oder Core ein eigener Cache vorgesehen sein.

Wenn diese beiden Prozessoren nun mit einem Taktversatz betrieben werden, kann nun mit dem in Figur 5 vorgeschlagenen Verfahren auf den zweiten Cache für den Slave-Prozessor verzichtet werden. Ein Cache benötigt viel Chipfläche und auch viel Strom. Dadurch produziert er auch viel Abwärme, die abgeführt werden muss. Kann nun auf einen Cache verzichtet werden, so lässt sich ein Zweiprozessorsystem deutlich kostengünstiger implementieren.

Eine modusabhängige Taktumschaltung der Prozessoren bzw. Cores wie in der vorher beschriebenen Ausführungsform ist auch bei einem System mit einem Cache möglich.

Bei dem hier als zusätzliches Ausführungsbeispiel vorgestellten Zweirechnersystem ist wie gesagt ein Prozessor der Master und ein Prozessor der Slave. Der Master arbeitet als erstes die Daten ab und steuert folglich auch die Peripheriekomponenten wie Speicher, Cache, DMA-Kontroller usw. an. Der Slave arbeitet die gleichen Daten mit einem Taktversatz von hier beispielhaft 1,5 Takte ab. Das bedeutet auch, dass er die Daten aus dem gemeinsamen Speicher und von den externen Komponenten ebenfalls um diese Zeitdauer später erhält. Die Ausgangsdaten der beiden Prozessoren wie Speicheradresse, Daten, usw. werden miteinander verglichen. Um die Daten miteinander vergleichen zu können, müssen die Ergebnisse des Masters ebenfalls 1,5 Takte zwischengespeichert werden. Ein solches Beispielsystem ist unten abgebildet.

Um gemäß Figur 5 nun ein Cache für beide Prozessoren verwenden zu können, werden nun der Befehls- und Datencache direkt am Master angeordnet wie bei einem Single-Prozessor. Der Master muss somit keine Performanceeinbußen bezüglich der Laufzeiten zwischen Cache und Prozessor hinnehmen. Da der Slave die Daten erst 1,5 Takte später abarbeitet, kann man diese Zeit nun benutzen um die Daten an den zweiten nun räumlich weiter vom Cache entfernten Prozessor zu führen.

Dazu können bei einem beispielhaften Taktversatz von 1,5 Takten zwei Flip-Flops benutzt werden, wie dies in Figur 6 dargestellt ist. Das Erste wird mit dem Takt des Masters angesteuert, das Zweite mit dem Takt des Slaves. Das erste Flip-Flop wird direkt am Ausgang der Quelle positioniert. Das Zweite wird nun entsprechend der Länge, die das Signal in der Differenz zwischen den beiden Takten zurücklegen kann, entsprechend näher am Slave positioniert. Dies entspricht bei 1,5 Takte Zeitversatz der Laufzeitlänge in einem halben Takt und bei einem Taktversatz von 2 Takte der Laufzeitlänge von einem Takt. Dann übernimmt das zweite Flip-Flop das Signal. Nun kann noch einmal die Strecke, die das Signal während eines ganzen Taktes zurücklegen kann, überbrückt werden. In der Abbildung ist dies durch 1.) die nahe Anordnung an der Senke dargestellt, 2.) entspricht der Länge die in der Taktdifferenz zurückgelegt werden kann und 3.) ist die Länge die in einem Takt nach dem zweiten Flip-Flop zurückgelegt werden kann.

Ansprüche

1. Verfahren zur Taktumschaltung in einem System mit wenigstens zwei Recheneinheiten, wobei Umschaltmittel enthalten sind durch welche zwischen wenigstens zwei Betriebsmodi des Systems umgeschaltet werden kann, wobei bei wenigstens einer Recheneinheit bei einer Umschaltung des Betriebsmodus eine Taktumschaltung aktivierbar oder deaktivierbar ist.

2. Verfahren zur Taktumschaltung nach Anspruch 1, dadurch gekennzeichnet, dass die Taktumschaltung derart erfolgt, dass die Recheneinheiten mit einem vorgebbaren Phasenversatz arbeiten.

3. Verfahren zur Taktumschschaltung nach Anspruch 2, dadurch gekennzeichnet, dass der Phasenversatz halbzahlig vorgegeben wird.

4. Verfahren zur Taktumschaltung nach Anspruch 3, dadurch gekennzeichnet, dass der Pasenversatz zu 1,5 Takten vorgegeben wird.

5. Verfahren zur Taktumschaltung nach Anspruch 2, dadurch gekennzeichnet, dass der Phasenversatz bei einer Umsachaltung in den Sicherheitsmodus vorgegeben wird.

6. Verfahren zur Taktumschaltung nach Anspruch 2, dadurch gekennzeichnet, dass bei einer Umschaltung in den Performanzmodus der Phasenversatz ausgeglichen wird, so dass die Recheneinheiten im Performanzmodus gleichphasig arbeiten.

7. Verfahren zur Taktumschaltung nach Anspruch 1, dadurch gekennzeichnet, dass bei Aktivierung eine Umschaltung der Taktfrequenz derart erfolgt, so dass eine Recheneinheit mit einer höherer Taktfrequenz arbeitet als die andere.

8. Verfahren zur Taktumschaltung nach Anspruch 2, dadurch gekennzeichnet, dass der Phasenversatz so vorgegeben wird, dass gleiche Instruktionen mit einem Zeitversatz ungleich Null auf den wenigstens zwei recheneinheiten abgearbeitet werden.

9. Einheit zur Taktumschaltung in einem System mit wenigstens zwei Recheneinheiten, wobei Umschaltmittel enthalten sind durch welche zwischen wenigstens zwei Betriebsmodi des Systems umgeschalten werden kann, wobei die Einheit derart ausgestaltet ist, dass bei wenigstens einer Recheneinheit bei einer Umschaltung des Betriebsmodus eine Taktumschaltung aktivierbar oder deaktivierbar ist.

10. Einheit zur Taktumschaltung nach Anspruch 9, dadurch gekennzeichnet, dass die Einheit derart ausgestaltet ist, dass die Taktumschaltung derart erfolgt, dass die Recheneinheiten mit einem vorgebbaren Phasenversatz arbeiten.

11. Einheit zur Taktumschschaltung nach Anspruch 10, dadurch gekennzeichnet, dass die Einheit derart ausgestaltet ist, dass der Phasenversatz halbzahlig vorgegeben wird.

12. Einheit zur Taktumschaltung nach Anspruch 10, dadurch gekennzeichnet, dass die Einheit derart ausgestaltet ist, dass der Phasenversatz ganzzahlig vorgegeben wird.

13. Einheit zur Taktumschaltung nach Anspruch 11, dadurch gekennzeichnet, dass die Einheit derart ausgestaltet ist, dass der Pasenversatz zu 1,5 Takten vorgegeben wird.

14. Einheit zur Taktumschaltung nach Anspruch 10, dadurch gekennzeichnet, dass die Einheit derart ausgestaltet ist, dass der Phasenversatz bei einer Umsachaltung in den Sicherheitsmodus vorgegeben wird.

15. Einheit zur Taktumschaltung nach Anspruch 10, dadurch gekennzeichnet, dass die Einheit derart ausgestaltet ist, dass bei einer Umschaltung in den Performanzmodus der Phasenversatz ausgeglichen wird, so dass die Recheneinheiten im Performanzmodus gleichphasig arbeiten.

16. Einheit zur Taktumschaltung nach Anspruch 9, dadurch gekennzeichnet, dass die Einheit derart ausgestaltet ist, dass bei Aktivierung eine Umschaltung der

Taktfrequenz derart erfolgt, so dass eine Recheneinheit mit einer höherer Taktfrequenz arbeitet als die andere.

17. Einheit zur Taktumschaltung nach Anspruch 10, dadurch gekennzeichnet, dass die Einheit derart ausgestaltet ist, dass der Phasenversatz so vorgegeben wird, dass gleiche Instruktionen mit einem Zeitversatz ungleich Null auf den wenigstens zwei Recheneinheiten abgearbeitet werden.

18. Einheit zur Taktumschaltung nach Anspruch 9, dadurch gekennzeichnet, dass die Einheit derart ausgestaltet ist, dass die Taktumschaltung durch ein Modussignal, welches einen Betriebsmodus angibt, ausgelöst wird.

19. Einheit zur Taktumschaltung nach Anspruch 9, dadurch gekennzeichnet, dass die Einheit derart ausgestaltet ist, dass die Taktumschaltung durch Zuschalten oder Wegschalten von wenigstens einer Kippschaltung ausgelöst wird.

20. Einheit zur Taktumschaltung nach Anspruch 10, dadurch gekennzeichnet, dass die Einheit derart ausgestaltet ist, dass der Phasenversatz durch Zuschalten oder Wegschalten von wenigstens einer Kippschaltung ausgelöst wird.

21. System mit einer Einheit zur Taktumschaltung nach einem der Ansprüche 9 bis 17.

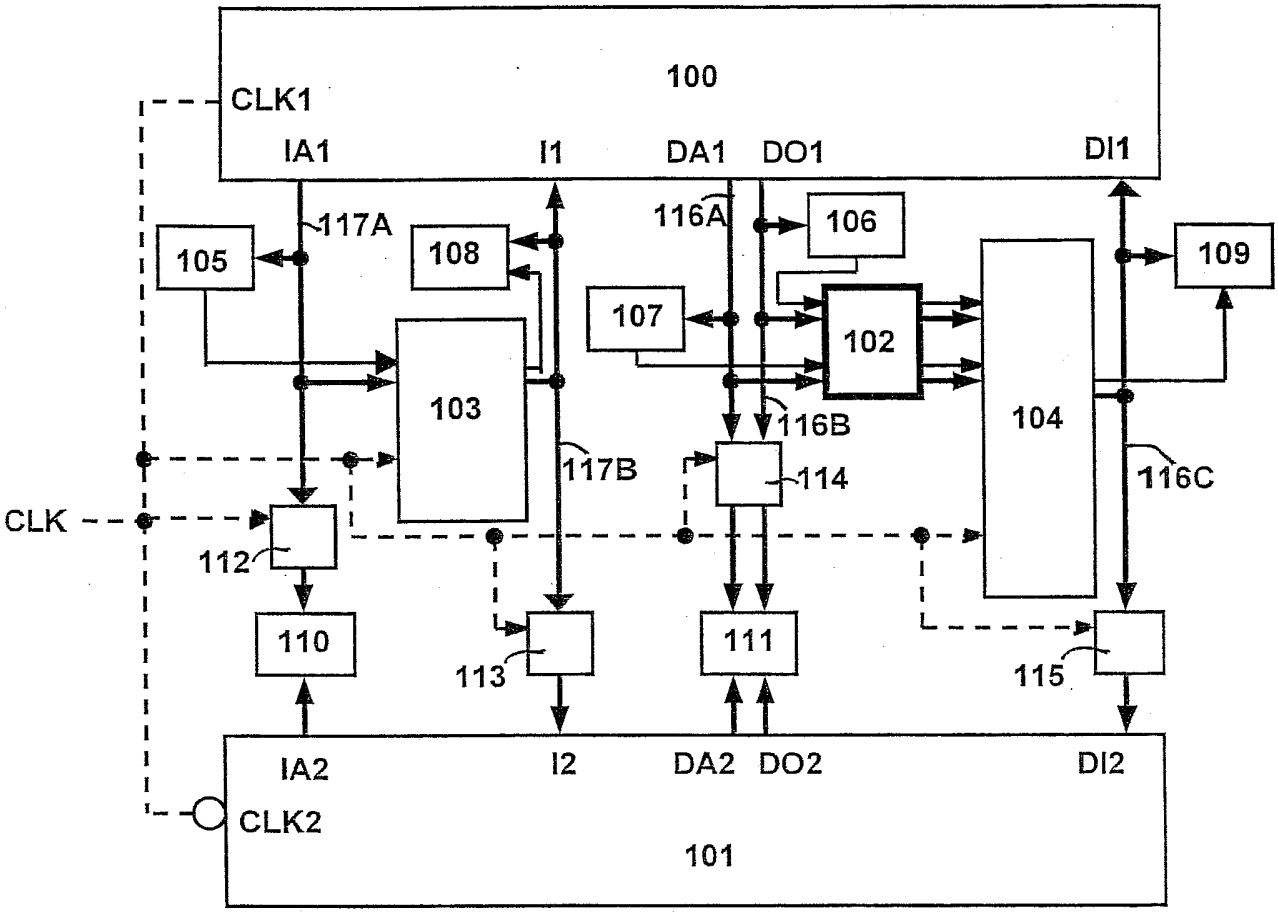

100
CLK1
IA1
I1
DA1
DO1
DI1
117A
116A
105
108
106
109
107
102
103
104
116B
117B
114
116C
CLK
112
110
113
111
115
IA2
I2
DA2
DO2
DI2
CLK2
101

Fig. 1

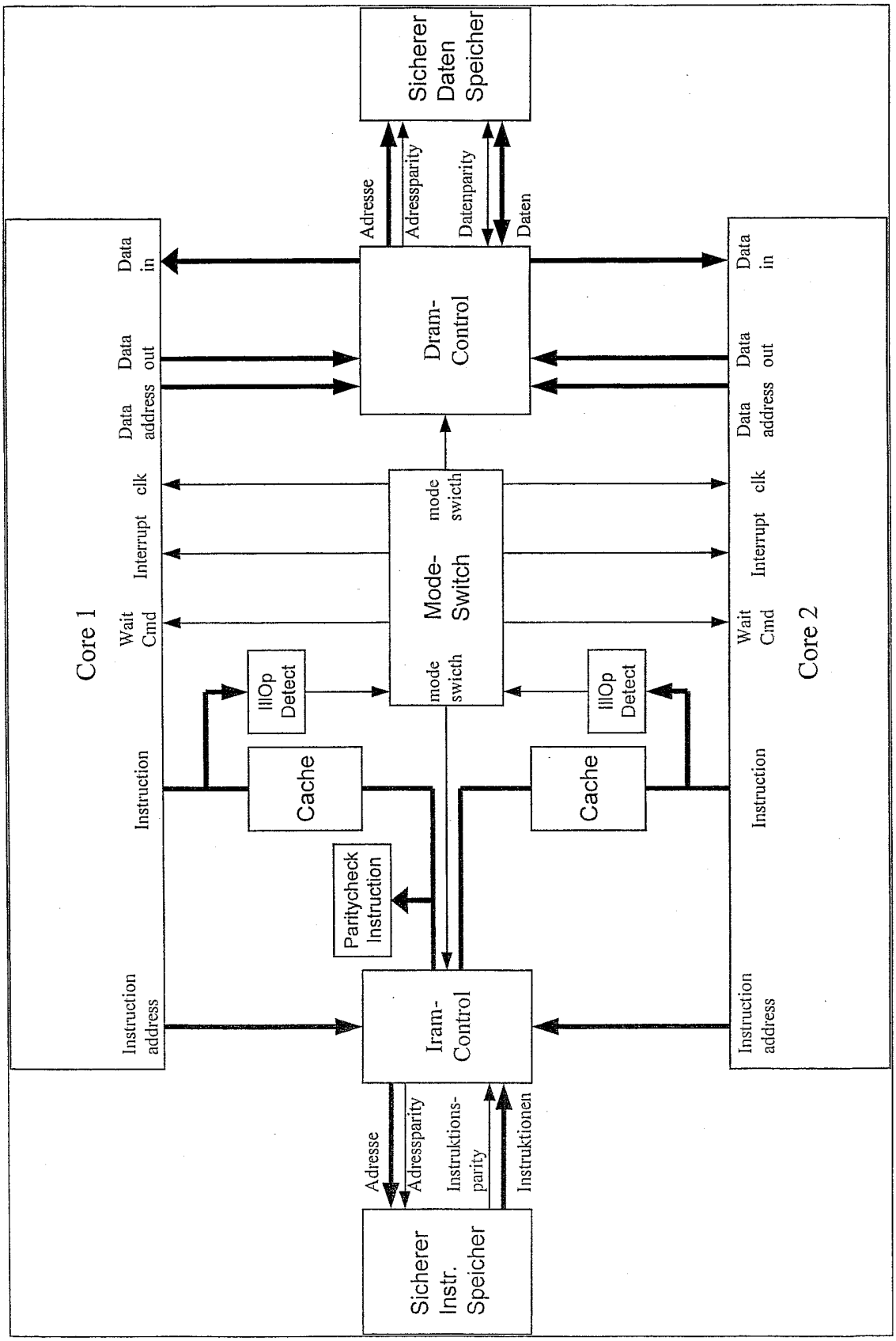

Sicherer Daten Speicher
Adresse
Adressparity
Datenparity
Daten
Dram-Control
Data in
Data out
Data address
Core 1
Core 2
Wait Cmd
Interrupt
clk
Mode-Switch
mode swicth
IllOp Detect
Cache
Instruction
Paritycheck Instruction
Instruction address
Iram-Control
Instruktions-parity
Instruktionen
Sicherer Instr. Speicher

Fig. 2

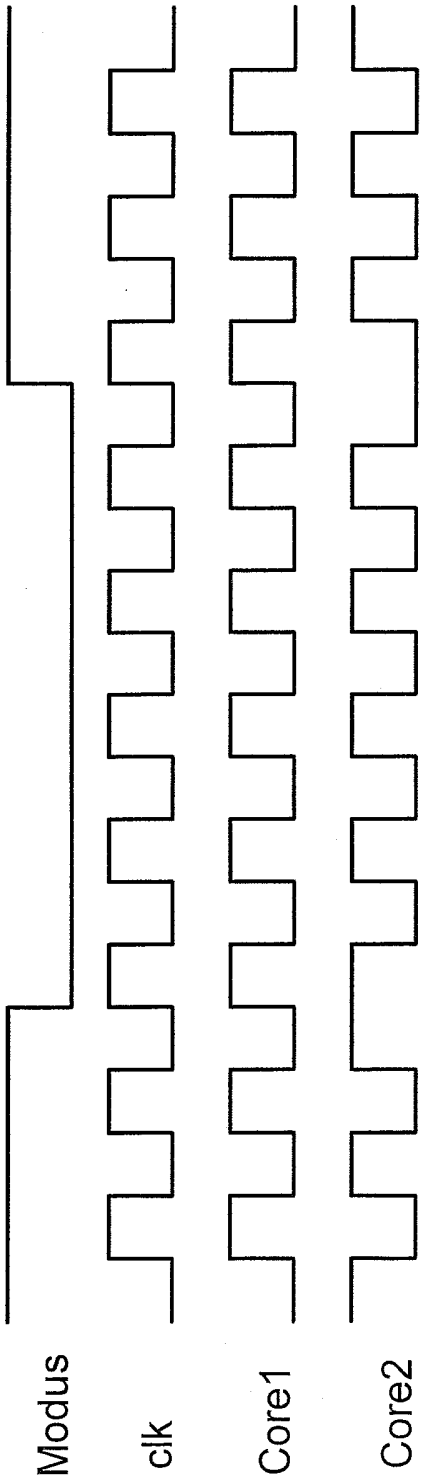


Fig. 3

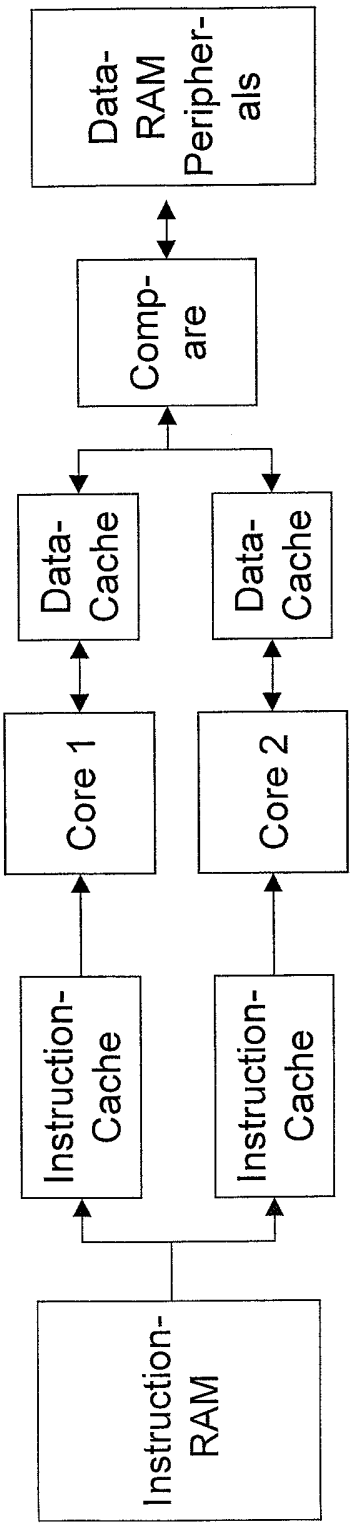

Data-RAM Peripher-als
Comp-are
Data-Cache
Data-Cache
Core 1
Core 2
Instruction-Cache
Instruction-Cache
Instruction-RAM

Fig. 4

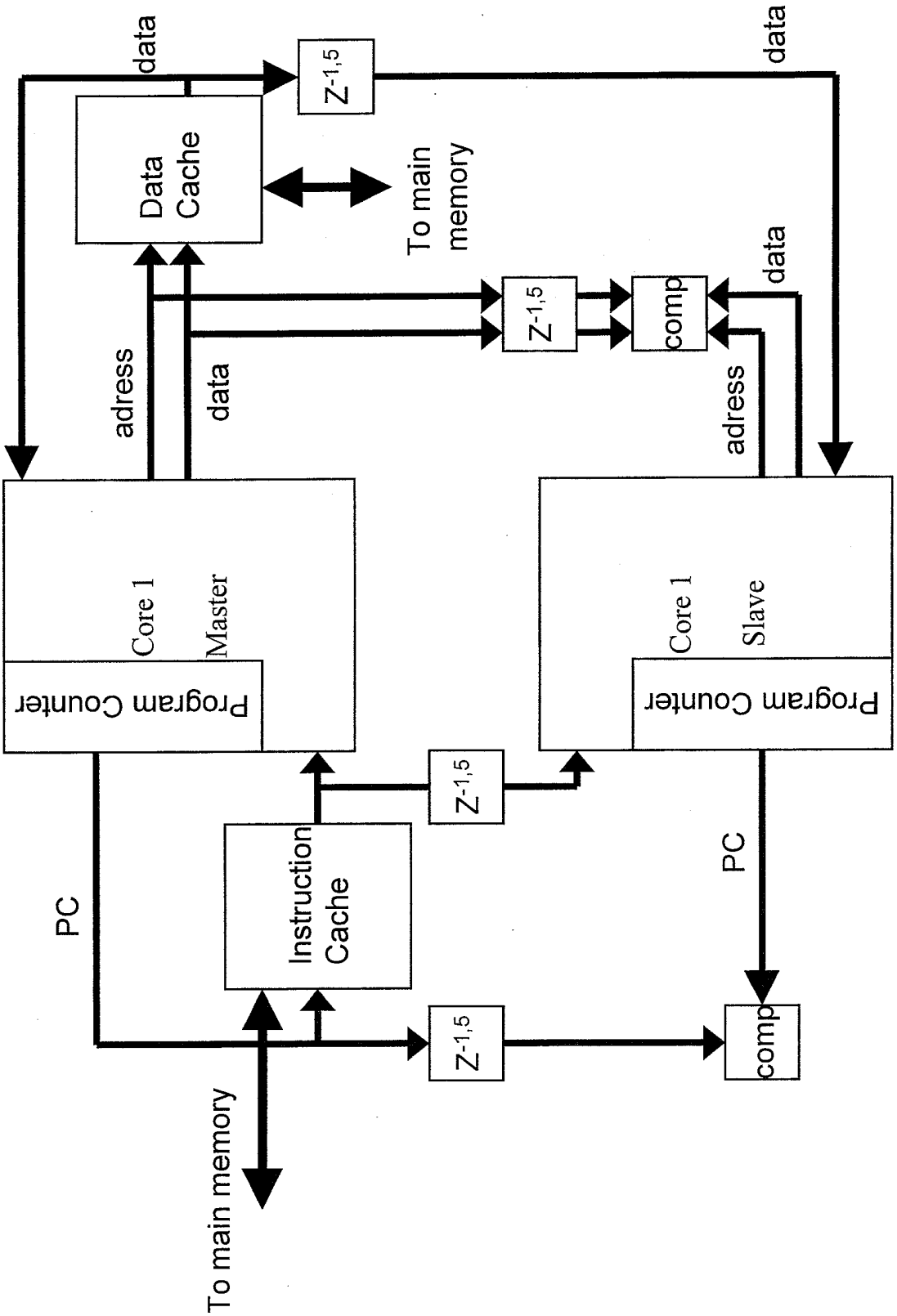

data
Z-1,5
data
Data Cache
To main memory
Z-1,5
comp
data
adress
data
adress
Core 1
Master
Program Counter
Core 1
Slave
Program Counter
Z-1,5
Instruction Cache
PC
PC
Z-1,5
comp
To main memory

Fig. 5

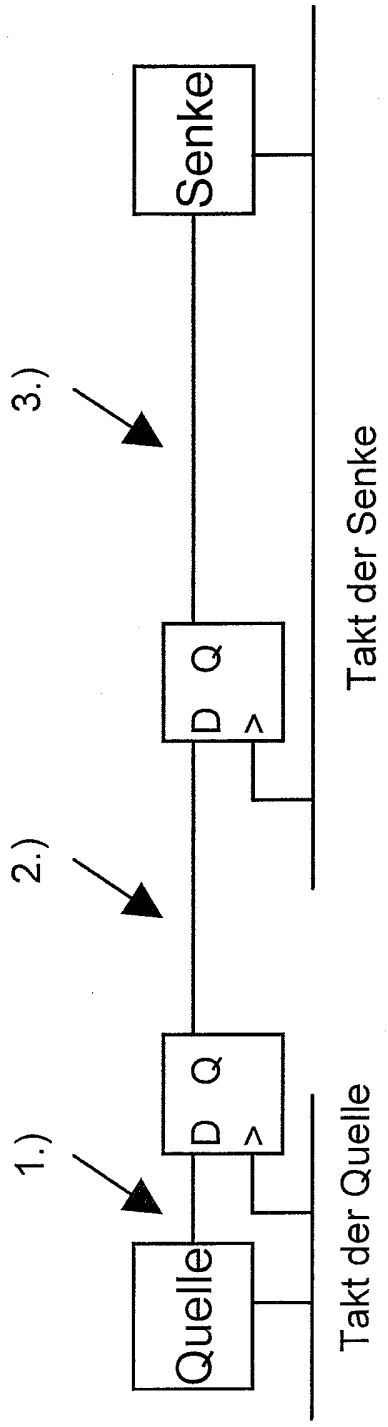


Fig. 6