

(19)

österreichisches  
patentamt

(10)

AT 413 308 B 2006-01-15

(12)

## Patentschrift

- (21) Anmeldenummer: A 1429/2003 (51) Int. Cl.<sup>7</sup>: G06F 13/14  
(22) Anmeldetag: 2003-09-10 G06F 13/00  
(42) Beginn der Patentdauer: 2005-06-15  
(45) Ausgabetag: 2006-01-15

(56) Entgegenhaltungen:

AT 408383B EP 0991216A2  
GB 2383434A US 5577075A  
US 6157957A US 637015981

(73) Patentinhaber:

FTS COMPUTERTECHNIK GES.M.B.H  
A-2500 BADEN-SIEGENFELD,  
NIEDERÖSTERREICH (AT).

(54) VERFAHREN UND APPARAT ZUR KALIBRIERUNG DES UHRENGANGS IN EINEM  
VERTEILTEM ECHTZEITSYSTEM

(57) Die vorliegende Erfindung hat zum Ziel, in einem verteilten Echtzeitcomputersystem die Qualität der globalen Zeitbasis wesentlich zu verbessern und eine einfache externe Uhrensynchronisation zu realisieren. Um dieses Ziel zu erreichen wird in einem System mit einer verteilten fehlertoleranten Uhrenstandsynchronisation erfindungsgemäß eine fehlertolerante zentrale Uhrenganganpassung vorgenommen. Die zentrale Uhrenganganpassung unterstützt auch die externe Synchronisation von Multi-cluster Systemen. Die Kombination von verteilter Uhrenstandsynchronisation und zentraler Uhrenganganpassung führt zu einer präzisen fehlertoleranten Zeitbasis, selbst bei Verwendung von ungenauen - und daher billigen - Oszillatoren.

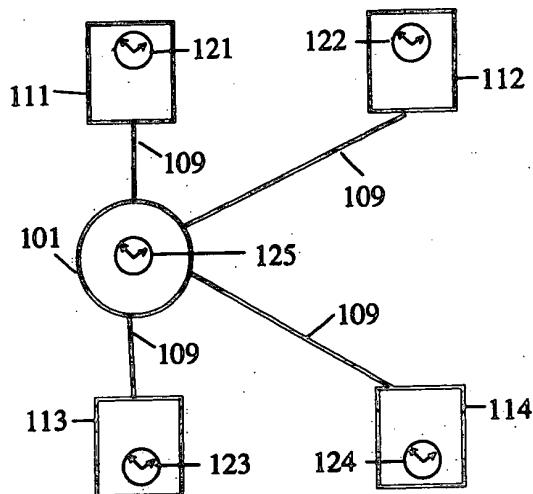


FIG. 1

Zitierte Patente:

[1] US 5694542 issued on Dec. 12, 1989: A loosely coupled distributed computer system with node synchronization for precision in real time.

5 [2] EP 0 658 257 vom 18.12.1996: Kommunikationskontrolleinheit und Verfahren zur Übermittlung von Nachrichten.

10 [3] US 5887143 issued on March 23, 1999: Time-Triggered Communcation Control Unit and Communication.

[4] AT 407 582 vom 15.6.2000: Nachrichtenverteilereinheit mit integriertem Guardian zur Verhinderung von Babbling Idiot Fehlern.

15 [5] AT 408 383 vom 15.3.2001: Verfahren und Kommunikationskontrolleinheit zur Multimasteruhrensynchronisation in einem verteilten Echtzeitcomputersystem.

[6] AT Patentanmeldung 1723/2001 vom 10. 10. 2000: Verfahren zur Tolerierung von slightly-off-specification Fehlern in einem verteilten fehlertoleranten Echtzeitsystem.

20 [7] AT Patentanmeldung 429/2001 vom 19.3.2001: Kommunikationsverfahren zur Realisierung von Ereigniskanälen in einem zeitgesteuerten Kommunikationssystem.

[8] GB 2 383 434 vom 25.6.2003 Fault-Tolerant Clock Synchronization

25 [9] EP 0 991 216 vom 10.6.99 Organization of Time Synchronization in Distributed Systems

[10] US 6 370 159 vom 9.4.2002 System Application Techniques using Time Synchronization

30 [11] US 6 157 957 vom 5.12.2000 Clock Synchronization System and method using a continuous conversion function for a communication network

[12] US 5 577 075 vom 19.11.1996 Distributed clocking system

35 [13] US 5907 685 vom 25.5.1999 System and Method for synchronizing clocks in distributed computer nodes

Sonstige Literatur:

40 [14] TTP/C Specification Document, 2002, Available from [www.tttech.com](http://www.tttech.com)

[15] Kopetz, H, Bauer, G, The Time-Triggered Architecture, *Proc. of the IEEE*, Jan. 2003, p. 112-126

45 [16] Kopetz, H. (1997). *Real-Time Systems, Design Principles for Distributed Embedded Applications*; ISBN: 0-7923-9894-7. Boston. Kluwer Academic Publishers.

TECHNISCHES UMFELD

50 Diese Erfindung betrifft die Verbesserung der Uhrensynchronisation in einer verteilten zeitgesteuerten Computerarchitektur für die Echtzeitdatenverarbeitung.

HINTERGRUND DIESER ERFINDUNG

55 Zeitgesteuerte verteilte Echtzeitsysteme stoßen aufgrund ihres vorhersehbaren Zeitverhaltens

auf großes industrielles Interesse. Ein zeitgesteuertes verteiltes Echtzeitsystem besteht aus einer Anzahl von Knotenrechnern, die über ein Kommunikationssystem zum Austausch von Nachrichten verbunden sind. In einem zeitgesteuerten Echtzeitsystem ist die Qualität der verfügbaren Zeitbasis, ausgedrückt durch die Präzision und die Fehlertoleranz der Uhrensynchronisation, von zentraler Bedeutung.

Entsprechend [16, S. 63] ergibt sich die *Präzision P* bei einer fehlertoleranten Uhrenstandsynchronisation mit dem FTA Algorithmus zu

$$P = (e + x) \frac{N - 2k}{N - 3k}$$

wobei  $e$  den *Jitter* (Differenz zwischen maximaler und minimaler Transportzeit) des Kommunikationssystems und  $x$  den *Driftoffset* der Uhren am Ende des Uhrenstandresynchronisationsintervalls darstellen.  $N$  entspricht der Anzahl der an der Synchronisation beteiligten Uhren und  $k$  ist die Anzahl der Uhren, die einen beliebigen Fehler haben können. Um einen beliebigen Fehler zu tolerieren benötigt man daher mindestens vier Uhren.

Der *Driftoffset x* errechnet sich aus

$$x = (2 \times \rho \times r)$$

wo  $\rho$  die durch die durch physikalischen Aufbau gegebene maximale Driftrate der Oszillatoren und  $r$  die Länge des Uhrenstandresynchronisationsintervalls darstellen.

## ZIEL DER ERFINDUNG

Die vorliegende Erfindung hat zum Ziel in einem Ensemble von kommunizierenden Echtzeitcomputeruhren, die über einen bekannten verteilten fehlertoleranten Uhrenstandsalgorithmus periodisch resynchronisiert werden, durch die Einführung eines Uhrengangmasters, der Zugriff auf eine externe Zeitreferenz hat, folgendes zu erreichen

1. den Stand aller Uhren an diese externe Zeitreferenz heranzuführen, um die Uhren des Ensembles extern zu synchronisieren, und
2. den Uhrengang aller Uhren des Ensembles an den Uhrengang des Uhrengangmasters anzugeleichen, um den *Driftoffset x* und damit die *Präzision P* des Ensembles signifikant zu verbessern.

Es ist ein wesentlicher Aspekt dieser Erfindung, dass bei Ausfall des Uhrengangmasters die Uhrenstandsynchronisation des bekannten verteilten fehlertoleranten Uhrenstandsynchronisationsalgorithmus aufrechterhalten bleibt.

Die Kombination einer verteilten fehlertoleranten Uhrenstandsynchronisation mit einer zentralen fehlertoleranten Uhrenganganpassung führt zu einem sehr kostengünstigen Uhrensynchronisationssystem für die externe und fehlertolerante interne Uhrensynchronisation von hoher Genauigkeit.

## STAND DER TECHNIK

Das Gebiet der Uhrensynchronisation in verteilten Echtzeitcomputersystem wird in der Patentliteratur mehrfach behandelt. Die Patente lassen sich wie folgt systematisch einteilen:

1. *Uhrenstandsynchronisation*: Die Uhrenstände der lokalen Uhren werden periodisch angepasst, um einen system-globale Zeitbegriff aufzubauen. Der Uhrengang wird nicht verändert.

1.1 zentrale Uhrenstandsynchronisation: Es gibt eine zentrale Referenzuhr, die statisch oder dynamisch ausgewählt wird: [8], [9], [10], [11], [12]

5 1.2 verteilte fehlertolerante Uhrenstandsynchronisation: Ein verteilter Algorithmus baut aus den lokalen Zeiten der Uhren eine globale Zeit auf. Eine festgelegte Anzahl von fehlerhaften Uhren wird toleriert [1], [2], [3], [5], [8], [12]

10 2. *Uhrengangssynchronisation*: Bei der Uhrengangssynchronisation wird die *die Drift (Geschwindigkeit) einer Uhr* verändert (siehe [[16], p.48]), ähnlich wie ein Uhrmacher die mechanische Frequenz eines *Unruheschwingsystems* einer Armbanduhr verändert. [10], [13]

15 3. *Externe Uhrensynchronisation*: Bei der externen Uhrensynchronisation wird der Stand der Uhren des Computersystems an den Stand einer externen Referenzuhr herangeführt [10], [5].

20 15 Die wesentliche Innovation der vorliegenden Erfindung ist die Kombination einer verteilten fehlertoleranten Uhrenstandsynchronisation mit einer zentralen Uhrengangssynchronisation, um eine mit einer Referenzzeit synchronisierte fehlertolerante globale Zeitbasis hoher Genauigkeit kostengünstig aufzubauen.

## 25 20 ZUSAMMENFASSUNG

25 Die vorliegende Erfindung hat zum Ziel, in einem verteilten Echtzeitcomputersystem die Qualität der globalen Zeitbasis wesentlich zu verbessern und eine einfache externe Uhrensynchronisation zu realisieren. Um dieses Ziel zu erreichen wird in einem System mit einer verteilten fehlertoleranten Uhrenstandsynchronisation erfindungsgemäß eine fehlertolerante zentrale Uhrenganganpassung vorgenommen. Die zentrale Uhrenganganpassung unterstützt auch die externe Synchronisation von Multi-cluster Systemen. Die Kombination von verteilter Uhrenstandsynchronisation und zentraler Uhrenganganpassung führt zu einer präzisen fehlertoleranten Zeitbasis, selbst bei Verwendung von ungenauen - und daher billigen - Oszillatoren.

## 30 30 KURZE BESCHREIBUNG DER ABBILDUNGEN

35 Das vorab beschriebene Ziel und andere neue Eigenschaften der vorliegenden Erfindung werden in den angeführten Abbildungen erläutert.

35 Fig. 1 zeigt die Struktur eines verteilten Computersystems.

Fig. 2. zeigt die Korrekturintervalle zu einem Resynchronisationszeitpunkt.

## 40 BESCHREIBUNG EINER REALISIERUNG

40 Im folgenden Abschnitt wird eine Realisierung des neuen Verfahrens an einem möglichen Beispiel mit vier Knotenrechnern, die über einen oder zwei replizierte Sternkoppler verbunden sind, gezeigt.

45 45 Fig. 1 zeigt ein verteiltes Computersystem mit einem Kommunikationssystem. Es besteht aus den vier Knotenrechner 111, 112, 113, und 114, die über je einen Kommunikationskontroller mit je einem bidirektionalen Kommunikationskanalanschluß zum Kommunikationskanal 109 verfügen. Über den Kommunikationskanal 109 werden periodisch Nachrichten ausgetauscht. In dem Kommunikationskanal kann sich ein intelligenter Sternkoppler 101 zur zentralen Steuerung der Kommunikation befinden. Jeder Knotenrechner 111, 112, 113, und 114, und der Sternkoppler verfügen über eine Echtzeituhr 121, 122, 123, 124 und 125. In einer anderen Ausprägung können die Kommunikationskanäle 109 über einen Bus verbunden sein. Die Kommunikationskanäle 109 können auch *wireless*, d.h. drahtlos sein. In diesem Fall findet sich in jedem Knotenrechner ein Kommunikationskontroller der das drahtlose Senden und Empfangen von Nachrichten unterstützt. Fig. 1 geht davon aus, dass die vier Knotenrechner 111, 112, 113, und 114 *time-*

keeping Knotenrechner sind, d.h., Knotenrechner die unter Anwendung eines verteilten Algorithmus, wie er in [10, S.60], beschrieben ist, eine fehlertoleranten Zeitbasis aufbauen. Weitere Knotenrechner, die an dem Aufbau einer fehlertoleranten Zeitbasis nicht aktiv beteiligt sind, sondern sich auf die von den *time-keeping* Knotenrechnern aufgebaute Zeitbasis synchronisieren ohne die Zeitbasis zu beeinflussen, sind in Fig. 1 nicht dargestellt.

Fig. 2 zeigt die Folge der Zeitpunkte am Ende eines periodischen Uhrenstandresynchronisationsintervalls, zu denen die lokale Repräsentation der globalen Zeit in den vier *time-keeping* Uhren den nächsten periodischen Synchronisationszeitwert erreicht. Auf der Abzisse 200 von

10 Fig. 2 ist Fortschreiten der Zeit dargestellt. Die Periode des Uhrenstandresynchronisationsintervalls sei  $r$  Zeiteinheiten lang. Da die vier Uhren einen unterschiedlichen Uhrentgang haben, erreicht die Uhr 121 den Zeitwert bereits zum Zeitpunkt 221. Die Uhr 122 erreicht den gleichen Zeitwert zum Zeitpunkt 222, die Uhr 123 zum Zeitpunkt 223 und die Uhr 124 zum Zeitpunkt 224. Die vier Knotenrechner errechnen mittels eines bekannten fehlertoleranten Synchronisationsal-

15 gorithmus (z.B. dem FTA Algorithmus in [10, S.60]) den gemeinsamen Mittelwert des Uhrenstandes 230. Entsprechend diesem fehlertoleranten Algorithmus müssen die Uhren ihren lokalen Uhrenstand verändern, und zwar erhöht die Uhr 222 ihren lokalen Uhrenstand um den Wert 231 und die Uhr 224 erniedrigt ihren lokalen Uhrenstand um den Wert 232. Die beiden anderen Uhren 221 und 223 verfahren entsprechend. Erfindungsgemäß sei nun *a priori* festgelegt, dass

20 die Uhr 224 die Rolle eines Uhrentgangmasters einnimmt, d.h., alle anderen *time-keeping* Uhren müssen ihren Uhrentgang periodisch innerhalb eines Uhrentgangresynchronisationsintervalls (UGRSI) an den Uhrentgang des Gangmasters 224 anpassen. Die Länge von UGRSI sei ein Vielfaches vom Uhrenstandresynchronisationsintervall  $r$ . Entsprechend dem Algorithmus verändert der Gangmaster seinen Uhrentgang nicht. Im folgenden sei die Uhrentgangsanpassung aus

25 der Sicht der Uhr 222 beschrieben. Um die Uhrentgangdifferenz zwischen der Uhr 222 und dem Gangmaster 224 während eines UGRSI zu bestimmen, muss die Uhr 222 in einem Zähler alle Korrekturwerte 231 der Uhr 222 und 232 der Uhr 224 summieren. Dividiert man diese Summe durch die Länge von UGRSI, so erhält man die Uhrentgangdifferenz zwischen der Uhr 222 und dem Uhrentgangmaster 224 während des letzten UGRSI.

30 Um ein fehlerhaftes Verhalten des Gangmasters zu erkennen, wird der Wert der errechneten Uhrentgangdifferenz und die erste Ableitung der Uhrentgangdifferenz überprüft ehe die Uhr 222 ihren Uhrentgang korrigiert. Der Wert der Uhrentgangdifferenz muss innerhalb eines Intervalls liegen, dass von den physikalischen Parametern der verwendeten Oszillatoren abhängt. Ange-

35 nommen, der Hersteller der Oszillatoren garantiert eine Abweichung des Oszillators von der physikalischen Zeit von besser als  $10^{-5}$  sec/sec, dann muss der maximale Wert der Uhrentgangdifferenz von noch „guten“ Oszillatoren kleiner sein als  $2 \times 10^{-5}$  sec/sec. Wenn die gemessene Uhrentgangdifferenz größer ist als dieser Maximalwert, dann ist entweder der Oszillator des Uhrentgangmasters 224 fehlerhaft oder der Oszillator des Rechnerknotens 222 ist fehlerhaft. In

40 beiden Fällen wird vom Rechnerknoten 222 keine Anpassung der Uhrentgangdifferenz durchgeführt und eine Diagnosemeldung an einen Diagnoserechner versandt. In weiterer Folge wird aufgrund dieses Fehlers der FTA Algorithmus zur Uhrenstandsynchronisation die fehlerhafte Uhr nicht weiter berücksichtigt.

45 Analog wird auch die Änderungsgeschwindigkeit der Uhrentgangdifferenz überprüft. Aufgrund von physikalischen Gesetzmäßigkeiten kann sich der Gang einer Uhr nur innerhalb bekannter Grenzen ändern. Wenn die Änderungsgeschwindigkeit einen vorgegebenen technologiebestimmten Maximalwert überschreitet, so muss auch ein Fehler vorliegen.

50 Wenn der Uhrentgangmaster 114 fehlerhaft wird, so kann *a priori* ein *Shadow Master* bestimmt werden, der die Uhrentgangkalibrierung übernimmt. Z.B. kann der Rechnerknoten 113 diese Rolle übernehmen, falls der Rechnerknoten 114 ausfällt.

55 Wenn nun die Uhr 222 festgestellt hat, dass die Uhrentgangdifferenz zwischen dem Uhrentgangmaster und der Uhr 222 innerhalb der erlaubten Grenzwerte von Uhrentgang und erster

Ableitung des Uhrengangs liegt, dann ändert der Rechnerknoten 112 den lokalen Parameter in seiner Uhrensynchronisationshardware, der das Verhältnis der Mikroticklänge zur Makroticklänge bestimmt (beschrieben in [8]) derart, dass von nun an der Uhrengang der Uhr 222 dem Uhrengang des Uhrengangmasters 224 entspricht. Die Funktion des fehlertoleranten Uhren-

5 standalgorithmus wird durch diese Anpassung nicht verändert. Alle anderen *time-keeping* Knoten, die nicht die Funktion des Uhrengangmasters ausführen, verfahren analog.

Wenn der Uhrengangmaster 114 keinen Zugriff auf eine Referenzzeit hat, so bestimmt der Uhrengang des Uhrengangmasters somit den Uhrengang des gesamten Ensembles von Rech-

10 nerknoten. Es ist deshalb sinnvoll, im Uhrengangmaster einen besonders genauen Oszillator zu installieren.

Wenn hingegen der Uhrengangmaster 114 Zugriff auf eine externe Referenzzeit, z.B. auf einen GPS Empfänger [10, S. 65] hat, so wird er seinen Uhrengang dem Uhrengang der Referenzzeit

15 anpassen. Sinnvollerweise wird der Uhrengangmaster in diesem Fall auch den Uhrenstand an den Uhrenstand der Referenzzeit anpassen. Dies bezeichnet man externe Synchronisation [10, S. 65]. Die externe Synchronisation kann in zwei Phasen geschehen. Die erste Phase betrifft

20 den Systemstart. Nach dem Einschalten wird der Uhrengangmaster zuerst die Referenzzeit übernehmen und dann eine Initialisierungsnachricht mit dieser Referenzzeit an alle anderen Knoten des Ensembles senden. Die zweite Phase betrifft den laufenden Betrieb. Wenn im laufenden Betrieb der Uhrenstand des Ensembles von der Referenzzeit abweichen sollte, dann wird der Uhrengangmaster 114 den Uhrengang des gesamten Ensembles langsam (innerhalb

der beschriebenen Schranken) so beeinflussen, dass der Uhrenstand des Ensembles sich dem Uhrenstand der Referenzzeit anpasst.

25

Wenn der Uhrengangmaster 114 ausgefallen ist, so kann der Shadow Master 113 den System-

start übernehmen. In diesem Fall muss der Shadow Master 113 auch Zugriff auf die externe Referenzzeit haben.

30 Wenn die externe Synchronisation fehlertolerant sein soll, so müssen mindestens drei Uhren vorhanden sein, die voneinander unabhängig Zugriff auf die Referenzzeit haben. Diese drei Uhren übernehmen dann den Systemstart und akzeptieren einen Zeitwert nur, wenn er von mindestens zwei der drei Uhren angeboten wird. Da zum Zeitpunkt des Systemstarts diese drei Uhren bereits über eine synchronisierte Zeitbasis (die Referenzzeit) verfügen, kann der System-

35 start bereits nach einem *a priori* festgelegten Zeitplan erfolgen.

40 Die nicht *time-keeping* Uhren, die keinen Einfluss auf die Bildung der globalen Zeit haben, können ihren Uhrenstand und ihren Uhrengang an den durch den fehlertoleranten Algorithmus errechneten Mittelwert der *time-keeping* Uhren anpassen.

45 Da das beschriebene Verfahren zur Uhrengangkalibrierung vollkommen unabhängig ist von der fehlertoleranten Uhrenstandsynchronisation können in einem Ensemble auch unterschiedliche Rechnerknoten eingesetzt werden, d.h. solche die eine Uhrengangkalibrierung unterstützen und solche die eine Uhrengangkalibrierung nicht unterstützen. Dem Systemingenieur stehen somit

50 zwei Alternativen offen: er kann zwischen einem aufwendigere Knotenrechner (mit Uhrengang-anpassung) und einem billigen (ungenauen) Oszillator oder einem einfacheren Knotenrechner (ohne Uhrenganganpassung) und einem teureren (genauerem) Oszillator entscheiden, um eine Zeitbasis der gewünschten Genauigkeit aufzubauen. Bei kleinen Stückzahlen einer Serie wird die erste Alternative sinnvoll sein, bei großen Stückzahlen die zweite.

55

Aufgrund der Struktur des vorgestellten Verfahrens ist es möglich nachzuweisen, dass dieses Verfahren stabil ist und zu keinen Schwingungen im Uhrensynchronisationssystem führen kann.

Es sei weiters bemerkt, dass das vorab beschriebene Verfahren zur Kalibrierung des Uhren-

55 gangs auch direkt in der Hardware eines Rechnerknotens realisiert werden kann.

Abschließend sei festgehalten dass sich durch das hier offengelegte Verfahren zur Kalibrierung des Uhrengangs folgende wirtschaftliche Vorteile ergeben:

- Das Verfahren führt zu einer wesentlichen Verbesserung der Qualität der globalen Zeit in einem verteilten Echtzeitcomputersystem, selbst bei Verwendung von billigen Oszillatoren.
- Das Verfahren unterstützt die nahtlose Integration von interner und externer Uhrensynchronisation.
- Das Verfahren ermöglicht den Aufbau von verteilten Systemen mit gemischten Rechnerknoten, d.h. solche die die Uhrengangskalibrierung unterstützen und solche die sie nicht unterstützen.

### Patentansprüche:

1. Verfahren zur Synchronisation der Uhren in einem verteilten Echtzeitsystem, in dem  $n$  Knotenrechnern mit je mindestens einer Echtzeituhr über ein Kommunikationssystem bestehend aus einem oder mehreren Kommunikationskanälen periodisch Nachrichten austauschen, und wo  $k$  der  $n$  Knotenrechner ( $k \leq n$ ), die  $k$  *time-keeping* Uhren, mit einem verteilten fehlertoleranten Algorithmus eine verteilte globale Zeitbasis aufbauen durch gekennzeichnet, dass eine der  $k$  *time-keeping* Uhren *a priori* als Uhrengangmaster bestimmt wird und die anderen  $k-1$  *time-keeping* Uhren ihren Uhrengang an den Gang des Uhrengangmasters anpassen und wo der Uhrengangmaster, der Zugriff auf eine externe Zeitreferenz hat, bei Abweichung des Uhrenstandes der verteilten globalen Zeit vom Uhrenstand der externen Zeitreferenz seinen Uhrengang vorübergehend so ändert, dass die Uhrenstände der *time-keeping* Uhren in der Folge den Uhrenstand der externen Zeitreferenz erreichen.
2. Verfahren nach Anspruch 1 dadurch gekennzeichnet, dass bei einem permanenten Ausfall der *time-keeping* Uhr, die die Uhrengangmasterfunktion ausführt, eine *a priori* festgelegte andere *time-keeping* Uhr (shadow master) die Rolle des Uhrengangmasters übernimmt.
3. Verfahren nach Anspruch 1 dadurch gekennzeichnet, dass jede *time-keeping* Uhr, die nicht die Rolle des Uhrengangmasters ausführt, während eines *a priori* festgelegten periodischen Uhrengangresynchronisationsintervalls (UGRSI) die durch den fehlertoleranten Uhrensynchonisationsalgorithmus errechneten Korrekturwerte der eigenen Uhr und die Korrekturwerte des aktuellen Uhrengangmasters addiert und unmittelbar nach dem Ende des UGRSI diese Summe durch die Länge des UGRSI dividiert, um die Uhrengangdifferenz zwischen dem Gang der eigenen Uhr und dem Gang des Uhrengangmaster zu bestimmen.
4. Verfahren nach einem oder mehreren der Ansprüche 1 bis 3, dadurch gekennzeichnet, dass jede *time-keeping* Uhr, die nicht die Rolle des Uhrengangmasters ausführt, unmittelbar nach Berechnung der aktuellen Uhrengangdifferenz überprüft, ob der Absolutwert der Uhrengangdifferenz innerhalb eines Intervalls liegt, das sich mit den technischen Daten der verwendeten Oszillatoren verträgt und dass bei negativem Ausgang dieser Überprüfung keine Gangkorrektur der eigenen Uhr vorgenommen wird.
5. Verfahren nach einem oder mehreren der Ansprüche 1 bis 4, dadurch gekennzeichnet, dass jede *time-keeping* Uhr, die nicht die Rolle des Uhrengangmasters ausführt, unmittelbar nach Berechnung der aktuellen Uhrengangdifferenz überprüft, ob der Absolutwert der ersten Ableitung der Uhrengangdifferenz bezogen auf die Zeit kleiner ist als eine *a priori* vorgegebene Schranke und dass bei negativem Ausgang dieser Überprüfung keine Gangkorrektur der eigenen Uhr vorgenommen wird.
6. Verfahren nach einem oder mehreren der Ansprüche 1 bis 5, dadurch gekennzeichnet, dass jede *time-keeping* Uhr, die nicht die Rolle des Uhrengangmasters ausführt, die Uhren-

gangkorrektur lokal dadurch realisiert, dass der Makrotick/Microtick Konversionsfaktor in der lokalen Uhrensynchrosynchronisationshardware entsprechend verändert wird.

7. Verfahren nach einem oder mehreren der Ansprüche 1 bis 6 *dadurch gekennzeichnet*, dass der Uhrentaktmaster das System initialisiert und falls er Zugriff auf eine Referenzzeit hat beim *Startup* den aktuellen Uhrenstand der globalen Zeit auf den Uhrenstand der Referenzzeit setzt.
8. Verfahren nach einem oder mehreren der Ansprüche 1 bis 7 *dadurch gekennzeichnet*, dass dass bei Ausfall des Uhrentaktmasters der *Shadow Master* das System initialisiert und falls er Zugriff auf eine Referenzzeit hat beim *Startup* den aktuellen Uhrenstand der globalen Zeit auf den Uhrenstand der Referenzzeit setzt.
9. Verfahren nach einem oder mehreren der Ansprüche 1 bis 8 *dadurch gekennzeichnet*, dass mindestens drei der *time-keeping* Uhren Zugriff auf die externe Referenzzeit haben und jede dieser drei *time-keeping* Uhren nach dem *Power-up* eine Initialisierungsnachricht erst sendet wenn ein in einem *a priori* bekannten Zeitplan spezifizierter Sendezeitpunkt erreicht wird.
10. Verfahren nach einem oder mehreren der Ansprüche 1 bis 9, wobei in den Kommunikationskanälen ein oder mehrere Sternkoppler angeordnet sind, *dadurch gekennzeichnet*, dass die Überprüfungen des Uhrentaktes entsprechend den Ansprüchen 4 und 5 in den Sternkopplern durchgeführt werden.
11. Verfahren nach einem oder mehreren der Ansprüche 1 bis 10, *dadurch gekennzeichnet*, dass die  $(n-k)$  *nicht time-keeping* Uhren ihre Uhrentaktanpassung an den durch den fehlertoleranten verteilten Uhrenstandalgorithmus ermittelten Mittelwert der  $k$  *time-keeping* Uhren anpassen.
12. Verfahren nach einem oder mehreren der Ansprüche 1 bis 11, *dadurch gekennzeichnet*, dass in einem Ensemble Rechnerknoten mit einer Uhrentaktkalibrierung und Rechnerknoten ohne Uhrentaktkalibrierung nebeneinander eingesetzt werden können.
13. Knotenrechner nach Anspruch 1 *dadurch gekennzeichnet*, dass ein oder mehrere der in Ansprüchen 2 bis 10 beschriebenen Verfahrensschritte teilweise oder vollkommen in der Hardware des Knotenrechners realisiert sind.

## Hiezu 1 Blatt Zeichnungen

40

45

50

55

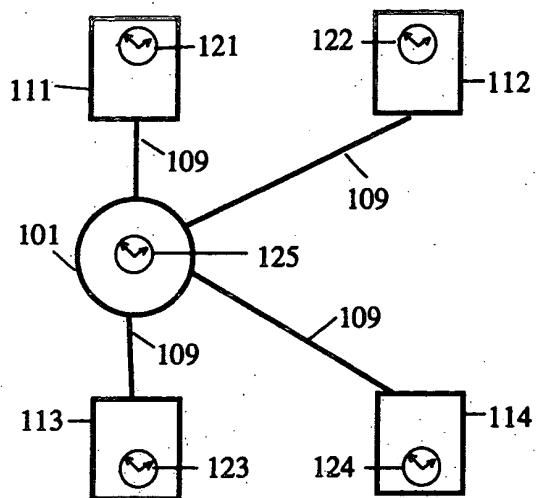


FIG. 1

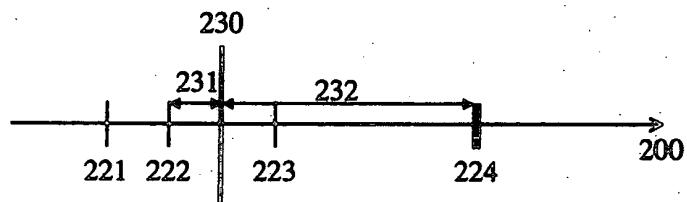


FIG. 2