DEUTSCHE DEMOKRATISCHE REPUBLIK

PATENTSCHRIFT

(12) Wirtschaftspatent

(19) DD (11) 275 545 A1

Erteilt gemäß § 17 Absatz 1 Patentgesetz

4(51) G 06 F 11/28

AMT FÜR ERFINDUNGS- UND PATENTWESEN

In der vom Anmelder eingereichten Fassung veröffentlicht

(21)	WP G 06 F / 319 880 8	(22)	16.09.88	(44)	24.01.90

(71) Akademie der Wissenschaften, Zentralinstitut für Kybernetik und Informationsprozesse, Kurstraße 33, Berlin, 1086, DD

(72) Stopp, Andreas, Dr.-Ing., DD

(54) Verfahren und Anordnung eines fehlertoleranten Mehrrechnersystems

(55) Mehrrechnersystem, Strukturredundanz, Signaturverfahren, Voting-System

(57) Das Ziel der Erfindung ist die Schaffung eines Rechnersystems, das einen hohen Fehlertoleranzgrad gestattet, das flexibel einsetzbar ist, das eine einfache Reparatur zuläßt und das eine effektive Auslastung gestattet. Erfindungsgemäß wird die Aufgabe dadurch gelöst, daß in einem fehlertoleranten Mehrrechnersystems mit untereinander kommunizierenden Rechnern N-Rechner zur Realisierung eines 2-aus-N-Systems parallel arbeiten. In N Rechnern werden identische Datenfolgen zu einem Signaturvektor komprimiert. Der Signaturvektor sowie das Ergebnis der Datenfolgen bzw. die Signatur der Ergebnisse der Datenfolge jedes Rechners werden über das Kommunikationsnetzwerk an alle N-1 anderen Rechner zur dezentralen Bewertung gesendet. Eine Selbstbewertung erfolgt durch Vergleich eines eigenen Signaturvektors mit dem als korrekt ermittelten Signaturvektor. Zur Fehlertoleranz gegen Entwurfsfehler wird mindestens ein zum Algorithmus des 2 aus N-Systems diversitärer Algorithmus auf mindestens einem Rechner abgearbeitet. Die Schaltungsanordnung zeichnet sich dadurch aus, daß jeder der M-Rechner einen fehlerunabhängigen Signaturschaltkreis und einen Signatur-RAM besitzt.

ISSN 0433-6461

10 Seiten

Patentansprüche:

1. Verfahren eines fehlertoleranten Mehrrechnersystems mit untereinander kommunizierenden Rechnern zur Realisierung eines 2-aus-N-Systems, wobei die Rechner parallel arbeiten, **dadurch gekennzeichnet,** daß identische Datenfolgen in N-Rechnern im gleichen Zeitfenster, aber nicht notwendigerweise zeitsynchron abgearbeitet werden, daß die Datenfolgen dabei in jedem Rechner zu einem Signaturvektor komprimiert werden und daß der Signaturvektor sowie das Ergebnis der Datenfolgen bzw. die Signatur der Ergebnisse der Datenfolge jedes Rechners über das Kommunikationsnetzwerk an alle N-1 anderen Rechner gesendet wird und daß jeder Rechner die beiden Vektoren der anderen Rechner in seinem Signatur-RAM-Bereich speichert und als dezentraler Signaturvoter arbeitet, indem jeder Rechner seinen eigenen Signaturvektor und die maximal N-1 erhaltenen vergleicht, wobei bei mindestens zwei identischen Signaturvektoren die zugeordneten Rechner, Datenfolgen und Ergebnisse in diesem Zeitraster als korrekt angenommen werden und daß eine Selbstbewertung durch Vergleich des eigenen Signaturvektors mit dem als korrekt ermittelten Signaturvektor erfolgt.
2. Verfahren nach Anspruch 1, **dadurch gekennzeichnet,** daß mindestens ein zum Algorithmus des 2-aus-N-Systems diversitärer Algorithmus auf mindestens einem Rechner abgearbeitet wird, über den ebenfalls die Prozeßsignatur und die Ergebnissignatur gebildet wird und daß ein zweiter Vergleich der Ergebnisse bzw. Ergebnissignaturen der diversitären Algorithmen stattfindet und daß das Vergleichsergebnis ein Test ist, der über die Wiederholung desselben oder eines diversitären Algorithmus entscheidet.
3. Verfahren nach Anspruch 2, **dadurch gekennzeichnet,** daß bezüglich des Vergleichs der Ergebnissignaturen von 1 diversitären Algorithmen ein 2-aus-1-System organisiert wird.
4. Verfahren nach Anspruch 2 oder 3, **dadurch gekennzeichnet,** daß mindestens ein diversitärer Algorithmus auf mindestens einem Rechner im gleichen Zeitfenster zum 2-aus-N-System abgearbeitet wird.
5. Verfahren nach Anspruch 2 oder 3, **dadurch gekennzeichnet,** daß mindestens ein diversitärer Algorithmus in mindestens einem nachfolgenden Zeitfenster auf mindestens einem der N-Rechner der 2-aus-N-System abgearbeitet wird.
6. Schaltungsanordnung zur Durchführung des Verfahrens nach Anspruch 1 bis 5 eines fehlertoleranten Mehrrechnersystems bestehend aus mindestens gruppenweise gleichen Rechnern, wobei jeder Rechner mindestens eine CPU, einen ROM, einen RAM, einen Timer, ein I/O-Interface und ein Kommunikationsinterface besitzt, **dadurch gekennzeichnet,** daß jeder der M-Rechner einen fehlerunabhängigen Signaturschaltkreis (10) besitzt, der mit dem hierarchisch höchsten Prozessor (1) des Rechners über den a-Bit-Datenvektor (D), den Steuersignalvektor (S) und den Adreßivektor (A) verbunden ist, daß in jedem Rechner ein Signatur-RAM (7) existiert, der die eigenen Prozeß- und Ergebnissignaturen sowie die von den anderen Rechnern gesendeten speichert.
7. Schaltungsanordnung nach Anspruch 6 unter Verwendung von bidirektionalen Verstärkern, Eingabespeicher, Signaturspeicher, Signaturkomparatoren, Timern, Anzeigesteuerung und einer testobjektabhängigen Adaptierschaltung, über die die Anordnung an den Prozessor angeschaltet ist, **dadurch gekennzeichnet,** daß ein a-Bit-Datenvektor (D) des Prozessors (1) über mindestens einen bidirektionalen Treiber des Signaturschaltkreises (10) und mindestens einen Eingabespeicher an eine 1-bit-Signaturverarbeitungslogik geschaltet sind, deren Ausgänge mit dem 1-bit-Signaturspeicher verbunden sind und daß sein nichtnegierter 1-Bit-Signaturvektor auf mindestens einen bidirektionalen Treiber, auf die 1-Bit-Signaturverarbeitungslogik als Zustandsinformation und auf den ersten 1-Bit-Signaturkomparator und daß sein negierter 1-Bit-Signaturvektor auf eine Anzeigesteuerung und einen zweiten 1-Bit-Signaturkomparator geschaltet sind, daß die Signaturspeicher bezogen auf den a-Bit Datenvektor (D) vollständig steuerbar und vollständig beobachtbar sind, daß durch Vergleichssteuersignale getrennt wählbare fehlersichere Festsollsignaturen steuerbar sind, die von einem ersten Festsignaturspeicher nichtnegiert an den ersten 1-Bit-Signaturkomparator und von einem zweiten Festsignaturspeicher negiert an den zweiten 1-Bit-Signaturkomparator geschaltet sind und daß die Ausgänge der beiden 1-Bit-Signaturkomparatoren getrennt auf eine Auswerteschaltung gegeben werden, die unabhängig zwei Retrigger- (RTG 1/2) und zwei Interruptsignale (INT 1/2) zur doppelten Weiterverarbeitung und/oder zum Anschluß zweier unabhängiger Timer (13, 14) liefert und daß aus der Rückmeldung

der Ausgangssignale der Timer (13, 14) ein nichtnegiertes Inhibitsignal (INH) und ein negiertes Inhibitsignal (NINH) zur redundanten Summenfehlermeldung geschaltet wird und daß beide Signale an alle Kommunikationsinterface (11) und an alle I/O-Interface (12) geschaltet sind.

8. Schaltungsanordnung nach Anspruch 6 oder 7, **dadurch gekennzeichnet,** daß jeder Rechner eine eigene von anderen Rechnern fehlerunabhängige Stromversorgung (8) und eine vorgeschaltete abschaltbare Rechnerschnittstelle (15) besitzt und daß diese derart abschaltbar ist, daß ein rückwirkungsfreies Entfernen und Replazieren einer Rechnerbaugruppe ermöglicht wird.
9. Schaltungsanordnung nach Anspruch 6 bis 8 zur fehlertoleranten Meßgrößenerfassung, **dadurch gekennzeichnet,** daß eine Meßgröße von mehr als zwei unabhängigen Meßwertaufnehmern erfaßt und getrennt an mindestens zwei fehlerunabhängige Rechner verschaltet wird.
10. Schaltungsanordnung nach Anspruch 6 oder 9 zur fehlertoleranten Stellgrößenschaltung, **dadurch gekennzeichnet,** daß die fehlerunabhängigen Ausgabekanäle unabhängiger Rechner getrennt auf jeweils mehrere Stellgrößenschalter einer Stellgröße geschaltet werden.

Hierzu 3 Seiten Zeichnungen

Anwendungsgebiet der Erfindung

Anwendungsgebiete der Erfindung sind Rechnersysteme, die ständig oder zeitweise Aufgaben mit hoher Zuverlässigkeit bearbeiten müssen, insbesondere für die Prozeßautomatisierung oder zur Bearbeitung nichtreproduzierbarer Vorgänge, beispielsweise der Bildverarbeitung.

Charakteristik des bekannten Standes der Technik

Bekannt sind die Anwendung von Signaturverfahren zur Datenkompression. In (Voelkel, Lutz; Pliquett, Jürgen: Signaturanalyse. Akademie-Verlag Berlin 1988) wird das gegenwärtige Wissen auf diesem Fachgebiet zusammengefaßt.
Votingprinzipien wie Mehrheitsentscheid, Schwellenentscheid und 2-aus-N-Systeme sind aus Realisierungen der hybriden Redundanz als Vorschläge in der Literatur bekannt. Der praktische Nachteil ist, daß sie nur sehr aufwendig realisierbar sind, insbesondere der Voting- und Abschaltvorgang.
Bekannt sind Prinzipien der Fehlertoleranz auf der Grundlage statischer und dynamischer Redundanz. Verfahren der dynamischen Redundanz umfassen Selbstdiagnose und anschließendes Recovery (Rekonfiguration und Restart), die eine hochzuverlässige Selbstdiagnose und einen hochzuverlässigen Umschaltmechanismus von der defekten auf eine intakte Einheit voraussetzen. Das ist nur sehr aufwendig realisierbar.
Die Literaturstellen (Lala. Parag K.: Fault tolerant and fault testable hardware design. Prentice Hall, London, 1985) und (Hedtke, Rolf.: Mikroprozessorsysteme: Zuverlässigkeit, Testverfahren, Fehlertoleranz. Springer-Verlag, Berlin Heidelberg N.Y., 1984) sowie die weiter unten zitierte Literaturstelle beschrieben diese Prinzipien.
Die statische Redundanz benötigt hochzuverlässige Voter-Bausteine und ist auf Grund der aufwendigen Instrumentierung, der für viele Anwendungsfälle sogar ungenügenden Zuverlässigkeitswerte für die vorliegende Aufgabenstellung ungeeignet. Auch das in den Zuverlässigkeitswerten vorteilhafte M-aus-N-System der hybriden Redundanz vereinigt die technischen Probleme der dynamischen und statischen Redundanz und besitzt teilweise deren Nachteile. Aus der Literatur sind inbesondere die Voting-Prinzipien des Mehrheitsentscheids (majority voting) und des Schwellenentscheids (threshold voting) bekannt, die eine Bewertung aber lediglich durch Vergleich der Ergebnisdaten zulassen.
In der Literaturstelle (Pradhan, D. K.: Fault-Tolerant Computing – Theory and Techniques. Prentice Hall, 1986) wird der SIFT-Computer für die Luftfahrt beschrieben, der mit unabhängigen Rechnerknoten arbeitet. Hierbei werden bezüglich der Input-Daten softwareimplementierte Mehrheitsvoter eingesetzt, deren Nachteile bereits genannt wurden.

Ziel der Erfindung

Das Ziel der Erfindung ist die Schaffung eines Rechnersystems, das einen hohen Fehlertoleranzgrad gestattet, das flexibel einsetzbar ist, das eine einfache Reparatur zuläßt und das eine effektive Auslastung gestattet.

Darlegung des Wesens der Erfindung

Aufgabe der Erfindung ist es, ein Verfahren und eine Schaltungsanordnung eines fehlertoleranten Mehrrechnersystems zu schaffen, das zentrale Umschaltmechanismen, zentrale Mehrheits- oder Schwellenentscheid-Mechanismen sowie starre Voting- und Redundanzprinzipien vermeidet.

Erfindungsgemäß wird die Aufgabe dadurch gelöst, daß in einem Mehrrechnersystem mit untereinander kommunizierenden und parallel arbeitenden Rechnern zur Realisierung eines 2-aus-N-Systems identische Datenfolgen im gleichen Zeitfenster aber nicht notwendigerweise zeitsynchron abgearbeitet werden. Die Datenfolgen werden dabei in jedem Rechner zu einem Signaturvektor komprimiert. Der Signaturvektor sowie das Ergebnis der Datenfolgen bzw. die Signatur der Ergebnisse der Datenfolge jedes Rechners werden über das Kommunikationsnetzwerk an alle N-1 anderen Rechner gesendet. Jeder Rechner speichert die beiden Vektoren der anderen Rechner in seinem Signatur-RAM-Bereich und arbeitet als dezentraler Signaturvoter, indem er seinen eigenen Signaturvektor und die maximal N-1 erhaltenen vergleicht, wobei bei mindestens zwei identischen Signaturvektoren die zugeordneten Rechner, Datenfolgen und Ergebnisse in diesem Zeitraster als korrekt angenommen werden. Eine Selbstbewertung erfolgt durch Vergleich des eigenen Signaturvektors mit dem als korrekt ermittelten Signaturvektor.
Zur Fehlertoleranz gegen Entwurfsfehler wird mindestens ein zum Algorithmus des 2-aus-N-Systems diversitärer Algorithmus auf mindestens einem Rechner abgearbeitet, über den ebenfalls die Prozeßsignatur und die Ergebnissignatur gebildet wird. Danach findet ein zweiter Vergleich der Ergebnisse bzw. Ergebnissignaturen der diversitären Algorithmen statt, wobei das Vergleichsergebnis ein Test ist, der über die Wiederholung desselben oder eines diversitären Algorithmus entscheidet.
Beim Vergleich der Ergebnissignaturen von 1 diversitären Algorithmen kann ein 2-aus-1-System organisiert werden.
Eine Variante ist, daß mindestens ein diversitärer Algorithmus auf mindestens einem Rechner im gleichen Zeitfenster zum 2-aus-N-System abgearbeitet wird.
Eine weitere Variante ist, daß mindestens ein diversitärer Algorithmus in mindestens einem nachfolgenden Zeitfenster auf mindestens einem der N-Rechner der 2-aus-N-System abgearbeitet wird.
Die erforderliche Schaltungsanordnung des fehlertoleranten Mehrrechnersystems besteht aus mindestens gruppenweise gleichen Rechnern. Jeder Rechner besitzt mindestens eine CPU, einen ROM, einen RAM, einen Timer, eine I/O-Schnittstelle und eine Kommunikationsschnittstelle. Die Schaltungsanordnung zeichnet sich dadurch aus, daß jeder der M-Rechner einen fehlerunabhängigen Signaturschaltkreis besitzt, der mit dem hierarchisch höchsten Prozessor des Rechners über den a-Bit-Datenvektor, den Steuersignalvektor und den Adreßvektor verbunden ist und daß in jedem Rechner ein Signatur-RAM existiert, der die eigenen Prozeß- und Ergebnissignaturen sowie die von den anderen Rechnern gesendeten speichert.
Der Signaturschaltkreis wird realisiert unter Verwendung von bidirektionalen Verstärkern, Eingabespeicher, Signaturspeicher, Signaturkomparatoren, Timern, Anzeigesteuerung und einer testobjektabhängigen Adaptierschaltung über die die Anordnung an den Prozessor angeschaltet ist. Der Signaturschaltkreis zeichnet sich dadurch aus, daß ein a-Bit-Datenvektor des Prozessors über mindestens einen bidirektionalen Treiber und mindestens einen Eingabespeicher an eine 1-bit-Signaturverarbeitungslogik geschaltet sind. Deren Ausgänge sind mit dem 1-bit-Signaturspeicher verbunden und sein nichtnegierter 1-Bit-Signaturvektor ist auf mindestens einen bidirektionalen Treiber, auf die 1-Bit-Signaturverarbeitungslogik als Zustandsinformation und auf den ersten 1-Bit-Signaturkomparator geschaltet und sein negierter 1-Bit-Signaturvektor ist auf eine Anzeigesteuerung und einen zweiten 1-Bit-Signaturkomparator geschaltet. Die Signaturspeicher müssen bezogen auf den a-Bit Datenvektor vollständig steuerbar und vollständig beobachtbar sein. Durch Vergleichssteuersignale sind getrennt wählbare fehlersichere Festsollsignaturen steuerbar, die von einem ersten Festsignaturspeicher nichtnegiert an den ersten 1-Bit-Signaturkomparator und von einem zweiten Festsignaturspeicher negiert an den zweiten 1-Bit-Signaturkomparator geschaltet werden. Die Ausgänge der beiden 1-Bit-Signaturkomparatoren werden getrennt auf eine Auswerteschaltung gegeben, die unabhängig zwei Retrigger- und zwei Interruptsignale zur doppelten Weiterverarbeitung und/oder zum Anschluß zweier unabhängiger Timer liefert. Aus der Rückmeldung der Ausgangssignale der Timer wird ein nichtnegiertes Inhibitsignal und ein negiertes Inhibitsignal zur redundanten Summenfehlermeldung erzeugt. Beide Signale werden zum Sperren der aktiven Kommunikation redundant an alle Kommunikationsinterface und an alle I/O-Interface geschaltet.
Zur Verbesserung der Fehlerunabhängigkeit ist vorgesehen, daß jeder Rechner eine eigene von anderen Rechnern fehlerunabhängige Stromversorgung und eine vorgeschaltete abschaltbare Rechnerschnittstelle besitzt. Diese muß derart abschaltbar sein, daß ein rückwirkungsfreies Entfernen und Replazieren einer Rechnerbaugruppe ermöglicht wird.
Zur fehlertoleranten Meßgrößenerfassung wird eine Meßgröße von mehr als zwei unabhängigen Meßwertaufnehmern erfaßt und getrennt an mindestens zwei fehlerunabhängige Rechner verschaltet.
Zur fehlertoleranten Stellgrößenschaltung werden die fehlerunabhängigen Ausgabekanäle unabhängiger Rechner getrennt auf jeweils mehre Stellgrößenschalter einer Stellgröße geschaltet.

Ausführungsbeispiel

Es zeigen:

Fig. 1: die erfindungsgemäße Anordnung
Fig. 2: den Ablauf einer ersten Verfahrensvariante
Fig. 3: den Ablauf einer zweiten Verfahrensvariante
Fig. 4: den Ablauf einer dritten Verfahrensvariante.

Das erfindungsgemäße Rechnersystem muß mindestens gruppenweise homogen kann auch vollständig homogen sein. Das entspricht der Forderung und Zukunftsprognose der Hardwareentwicklung.
Das Mehrrechnersystem umfaßt M weitgehend fehlerunabhängige und lose gekoppelte Rechner, die auf redundantem Wege kommunizieren können, so daß fehlerhafte, selbstisolierte oder entfernte Rechner sowie fehlerhafte Kommunikationsverbindungen bezüglich der Kommunikation umgangen werden können.
Die Wahl der Verbindungstopologie hängt vor allem vom beabsichtigten Wert M also der Anzahl der Rechner und der Anzahl der Link- bzw. Busports pro Rechner ab.

Anwendbar sind Multibus, Array, Hypertorus, Hyperkubus, Cube-Connected-Cycle u.a. Verbindungsstrukturen. Damit ist gewährleistet, daß jeder Rechner mit jedem anderen auf direktem oder indirektem Wege (über andere) kommunizieren kann und daß jeder über Broadcast alle anderen benachrichtigen kann. Das bedeutet, daß virtuell jeder Rechner mit jedem anderen verbunden ist. Die erforderliche Schaltungsanordnung des fehlertoleranten Mehrrechnersystems besteht aus mindestens gruppenweise gleichen Rechnern. Jeder Rechner besitzt mindestens einen Prozessor 1, einen ROM 4, eine RAM 5, einen Timer 6, ein I/O-Interface 12 und ein Kommunikationsinterface 11. Letztere sollten um Redundanz und Erweiterbarkeit zu ermöglichen mindestens 3 Link- bzw. 2 Busports umfassen.
Der Timer 6 steuert bzw. überwacht die Prozeßfenster bei der taskparallelen Bearbeitung des 2-aus-N-System.
Die Schaltungsanordnung zeichnet sich dadurch aus, daß jeder der M-Rechner einen fehlerunabhängigen Signaturschaltkreis 10 besitzt, der mit dem hierarchisch höchsten Prozessor 1 des Rechners über den a-Bit-Datenvektor D, den Steuersignalvektor S und den Adreßvektor A verbunden ist und daß in jedem Rechner ein Signatur-RAM 7 existiert, der die eigenen Prozeß- und Ergebnissignaturen sowie die von den anderen Rechnern gesendeten speichert. An den hierarchisch höchsten Prozessor können beispielsweise über Dual-Port-RAM 3 weitere Arbeits- und Coprozessoren 2 angeschlossen sein.
Der Signaturschaltkreis 10 dient zur parallelen Datenkompression der Daten des Datenbusses des Prozessors 1 entsprechend der Signaturanalyse, wobei gesteuert durch den Prozessor 1 sowohl Verarbeitungsprozesse bzw. ausgewählte Zustände von Prozessen zu einer Prozeßsignatur als auch Speicherinhalte z.B. Ergebnisdatenblöcke, die auf Verarbeitungsprozesse rückgeführt werden, zu einer Ergebnisdatensignatur komprimiert werden.
Zwei identische Prozesse bzw. Ergebnisdatenmengen müssen im Intaktfalle d.h. bei Identität folglich identische Signaturen aufweisen. Der Signaturschaltkreis 10 dient außerdem zur Selbstüberwachung des Rechners, zum Selbsttest und im festgestellten Fehlerfall zur Selbstisolierung, wobei dann die aktive Kommunikation über zwei Steuerleitungen das nichtnegierte Inhibitsignal INH und das negierte Inhibitsignal NINH redundant und fehlererkennbar (two-rail-codiert) unterbrochen wird. Dieses Doppelsignal kann auch an die direkten Nachbarn des jeweiligen Rechners zur beschleunigten Nachbarschaftsdiagnose verschaltet werden.
Der Signaturschaltkreis wird vom Prozessor initialisiert, gestartet, gestoppt, der Signaturvektor korrigiert, gelesen oder fehlersicher geladen.

Der Signaturschaltkreis läßt sich durch folgende Befehle steuern:

Funktion	Befehl	Adresse	Daten	Kommentar
RESI	OUT	XXX0	–	RESET-Befehl
START	OUT	XXX1	–	Freigabe Takttorung
STOP	OUT	XXX2	–	Sperren Takttorung
SELECT	OUT	XXX3	E	Taktauswahl
COMP 1	OUT	XXX4	–	Vergleich Grundtest
COMP 2	OUT	XXX5	–	Vergleich Gesamttest
RDO	IN	XXX6	S0	Istsignatur lesen (Bit 0–15)
RD 1	IN	XXX7	S1	Istsignatur lesen (Bit 16–21)
CORR 0	OUT	XXX6	C0	Puffer schreiben (Bit 0–15)
CORR 1	OUT	XXX7	C1	Puffer schreiben (Bit 16–21) und Signaturbildung

XXX = verdrahtungsprogrammierbare Chip-Select-Adresse

Der Signaturschaltkreis 10 wird realisiert unter Verwendung von bidirektionalen Verstärkern, Eingabespeicher, Signaturspeicher, Signaturkomparatoren, Timern, Anzeigesteuerung und einer testobjektabhängigen Adaptierschaltung über die die Anordnung an den Prozessor angeschaltet ist. Der Signaturschaltkreis 10 zeichnet sich dadurch aus, daß ein a-Bit-Datenvektor D des Prozessors 1 über mindestens einen bidirektionalen Treiber und mindestens einen Eingabespeicher an eine 1-bit-Signaturverarbeitungslogik geschaltet sind. Deren Ausgänge sind mit dem 1-bit-Signaturspeicher verbunden und sein nichtnegierter 1-Bit-Signaturvektor ist auf mindestens einen bidirektionalen Treiber, auf die 1-Bit-Signaturverarbeitungslogik als Zustandsinformation und auf den ersten 1-Bit-Signaturkomparator geschaltet und sein negierter 1-Bit-Signaturvektor ist auf eine Anzeigesteuerung und einen zweiten 1-Bit-Signaturkomparator geschaltet. Die Signaturspeicher müssen bezogen auf den a-Bit-Datenvektor D vollständig steuerbar und vollständig beobachtbar sein. Durch Vergleichssteuersignale sind getrennt wählbare fehlersichere Festsollsignaturen steuerbar, die von einem ersten Festsignaturspeicher nichtnegiert an den ersten 1-Bit-Signaturkomparator und von einem zweiten Festsignaturspeicher negiert an den zweiten 1-Bit-Signaturkomparator geschaltet werden. Die Ausgänge der beiden 1-Bit-Signaturkomparatoren werden getrennt auf eine Auswerteschaltung gegeben, die unabhängig je zwei Retrigger- RTG 1/2 und zwei Interruptsignale INT 1/2 zur doppelten Weiterverarbeitung und/oder zum Anschluß zweier unabhängiger Timer 13, 14 liefert. Aus deren Rückmeldung wird ein nichtnegiertes Inhibitsignal INH und ein negiertes Inhibitsignal NINH zur redundanten Summenfehlermeldung erzeugt. Beide Signale werden zum fehlererkennenden Sperren der aktiven Kommunikation redundant an alle Kommunikationsinterface 11 und an alle I/O-Interface 12 geschaltet. Das Senden wird schaltungstechnisch nur dann erlaubt, wenn INH = Low und NINH = High ist. Bei den restlichen drei Kombinationen der Signale wird gesperrt.

Das Entfernen eines Rechnerknotens aus dem Gesamtsystem muß ohne Beeinflussung des Restsystems, d.h. rückwirkungsfrei durchführbar sein.
Zur Verbesserung der Fehlerunabhängigkeit ist vorgesehen, daß jeder Rechner eine eigene von anderen Rechnern fehlerunabhängige Stromversorgung 8 und eine vorgeschaltete abschaltbare Rechnerschnittstelle 15 besitzt. Diese muß derart abschaltbar sein, daß ein rückwirkungsfreies Abschalten und Entfernen einer defekten Rechnerbaugruppe sowie das Replazieren und Zuschalten einer reparierten Rechnerbaugruppe ermöglicht wird. Das verbleibende Rechnersystem muß während dieser Zeit arbeitsfähig bleiben. Ein galvanisches Auftrennen des Abschaltsignals AS bewirkt das rückwirkungsfreie Isolieren der abschaltbaren Rechnerschnittstelle 15. Zur Darstellung des Arbeitszustandes und zur Darstellung von Fehlermitteilungen ist an den Signaturschaltkreis 10 optional eine Signatur- und Statusanzeige anschließbar.
In jedem Rechner existiert physisch ein Signatur-RAM, der ein reservierter RAM-Bereich sein kann. In diesem Signatur-RAM werden die Prozeßsignaturen und die Ergebnisse bzw. die Ergebnissignaturen sowohl die des eigenen Rechners als auch die N-1 gesendeten der anderen Rechner bis zum Signaturvotingprozeß abgelegt.
Auf N intakten der insgesamt M-Rechner kann das erfindungsgemäße 2-aus-N-System mittels dezentralem verteiltem Signaturvoting und diversitärer Taskbearbeitung installiert werden.
Wichtig ist, daß der Vergleich von zwei Signaturen als Gut-Schlecht-Test und der Vergleich von N Signaturen bei N > 2 als 2-aus-N-Vergleich genutzt wird.
Soll beispielsweise in einem Mehrrechnersystem mit 9 Rechnern ein 2 aus 7 System für eine bestimmte Task installiert werden, so muß in jedem der 7 Rechner diese Task gespeichert sein.
Die taskparallele Bearbeitung erfolgt nicht zeitsynchron wohl aber innerhalb eines begrenzten Zeitfensters. Das gestattet das für die Fehlertoleranz vorteilhafte Entkoppeln der Taktsysteme der Rechner.
In jedem Rechner wird die Taskbearbeitung mittels Signaturschaltkreis, gesteuert durch den Prozessor, gleichzeitig zur Bearbeitung bezüglich der repräsentativen Daten auf dem Prozessordatenbus und in Abhängigkeit programmselektierter Steuersignale zur Prozeßsignatur der Task komprimiert. Außerdem werden anschließend die Ergebnisdaten durch Lesen des Prozessors in gleicher Weise zu einer Ergebnisdatensignatur komprimiert.
Beide Signaturen werden im eigenen Signatur-RAM abgelegt und außerdem werden die beiden Signaturen an alle anderen beteiligten N-1-Rechner abgesendet.
Nach einer durch das Zeitfenster bestimmten Zeit beginnt der erste Votingprozeß, wobei in jedem Rechner also verteilt der Signaturvergleich der abgelegten Signaturen stattfindet. Unter der Annahme, daß keine gleichen Fehler zum gleichen Zeitpunkt in fehlerunabhängigen Rechnern auftreten, kann davon ausgegangen werden, daß zwei Prozesse mit übereinstimmenden Prozeßsignaturen bezüglich permanenten und transienten Hardwarefehlern intakt sind, daß also N-2-fehlerhafte-Prozesse toleriert werden können. Zusätzlich können die, zu den korrekten Prozeßsignaturen gehörenden, Ergebnisdatensignaturen verglichen werden, die außerdem als Schutz der Ausgabedaten beim Senden an die Prozeßperipherie bzw. beim Weiterverarbeiten als Schutz der Eingabedaten der nächsten Task dienen.
Jeder Rechner kann die korrekten Prozeß- und Ergebnisdatensignaturen (2 aus N) im eigenen Votingprozeß ermitteln und vergleicht diese mit den eigenen zwecks Selbstbewertung und im wiederholten Fehlerfall zwecks Selbstisolierung im Sinne der selbstreinigenden Redundanz aber auch um festzustellen ob die eigenen Ergebnisdaten weiterverarbeitet werden können oder nicht. Signaturverfahren besitzen eine sehr hohe Fehlererkennungswahrscheinlichkeit und sind vergleichsweise einfach zu instrumentieren.
Die eigenen Ergebnisdaten eines Rechners können dann zur Ausgabe oder zur Weiterverarbeitung als korrekt angenommen werden, wenn die Signatur über diese Ergebnisdaten mit mindestens einer Ergebnisdatensignatur eines anderen intakten Rechners übereinstimmt und wenn dieser auch die korrekte Prozeßsignatur aufweist. Der Vorteil ist, daß die Rechner untereinander nur die Signaturen austauschen müssens um den Prozeß und die Daten zu bewerten. Lediglich zum Schluß können die korrekten Ergebnisse zur Weiterverarbeitung z.B. über Broadcast verteilt werden. Weiterhin wird auch die Signatur zur Überprüfung der Datenübertragung eingesetzt. Das ist erforderlich, falls die Anzahl der teilnehmenden Rechner also der Redundanzgrad vergrößert wird, falls eine Rekonfiguration also Umverlagerung der Bearbeitung im Mehrrechnersystem stattfindet oder einfach nur wenn ein oder mehrere Rechner inkorrekte Ergebnisse hatten aber weiterhin im Verband bleiben und mit korrekten Ergebnissen die nachfolgende Task beginnen sollen.
Vorteil und Problem ist, daß nur vollkommen identische Prozeßabläufe bzw. Ergebnisdaten zu identischen Signaturen führen. Das widerspricht einer erforderlichen Diversität (Design-, Programmdiversität) zum Erkennen bzw. Tolerieren von Entwurfsfehlern.
Andererseits kann bei künftigen homogenen Rechnerstrukturen auf Grund der begrenzten Komplexität des einzelnen Rechners, der hohen Herstellungsstückzahlen und der damit verbundenen Perfektionierung des Entwurfs und der Herstellung davon ausgegangen werden, daß Hardware-Entwurfsfehler vermieden werden können. Da aber demgegenüber die Vielfalt der Softwareimplementierungen stark zunehmen wird, werden sich auch die Entwurfsfehler der Software häufen.
Im Normalfall d.h. bei einem ausgetesteten und die Regeln beachtenden Programmentwurf, der die Voraussetzung für den Einsatz in fehlertolerierenden Rechnersystemen ist, kann geschätzt werden, daß die Fehlerrate der Entwurfsfehler wesentlich geringer als die Fehlerrate der permanenten und transienten Hardwarefehler ist, die auf Alterserscheinungen, Herstellungsfehler, Umwelteinflüsse zurückzuführen und nicht vollständig vermeidbar sind.
Wenn nicht nur permanente und transiente Hardwarefehler und davon verursachte Softwarefehler toleriert werden sollen, sondern auch Softwareentwurfsfehler, so wird eine zweite Taskbearbeitung mit einem diversitären Programm durchgeführt. So ist diese Task dann in jedem der 7 Rechner in einer ersten Taskvariante und einer diversitären zweiten Taskvariante vorhanden. Die Diversität besteht dabei Algorithmus und der Logik ggf. durch unabhängige Programmierung.
Nachfolgend werden drei Beispielvarianten der diversitären Taskbearbeitung beschrieben.
Eine erste Verfahrensvariante (vgl. Fig. 2) beruht darauf, daß die Bearbeitung der zweiten Taskvariante auf der gleichen Menge Rechner und mit einem anschließender, zweiten Prozeßsignaturvoting der Rechner untereinander und anschließendem Vergleichstest der zwei repräsentativen korrekten Ergebnisdatensignaturen erfolgt. Merkmal dieser Variante ist geringe Hardwareredundanz auf Kosten von Zeitredudanz. Im Differenzfall muß auf einen Softwarentwurfsfehler in einer der beiden Programmvarianten ausgegangen werden, der mit sehr hoher Fehlererkennungswahrscheinlichkeit entdeckt wird.

Von der Aufgabenstellung hängt es ab, wie weiterverfahren wird. Entweder die Ergebnisse können durch Akzeptanztest beispielsweise logische Bewertung in korrekte und unkorrekte unterschieden werden, oder es muß eine dritte diversitäre Programmversion vorhanden sein, die für diesen selten auftretenden Fall u.U. nachgeladen und in beschriebener Weise abgearbeitet wird und deren Ergebnisdatensignatur dann im 2-aus-3-System wiederum verteilt bewertet wird, wodurch dann dieser Entwurfsfehler toleriert wird.
Eine zweite Verfahrensvariante (vgl. Fig. 3) beruht darauf, daß die diversitäre Bearbeitung auch parallelisiert werden kann. Merkmal dieses Verfahrens ist reduzierte Zeitredundanz auf Kosten von Hardwareredundanz. Dazu werden zur Entwurfsfehlererkennung zwei, zur Entwurfsfehlertoleranz mindestens drei annähernd gleichgroße aber nicht notwendigerweise gleichgroße Gruppen von Rechnern dynamisch festgelegt, wobei in der ersten Gruppe die erste Programmversion, in der zweiten Gruppe die zweite Programmversion usw. im gleichen Zeitfenster bearbeitet wird. In den Gruppen wird 2-aus-N-Prozeßsignaturvoting durchgeführt und sofort anschließend über die repräsentativen korrekten Ergebnissignaturen das 2-aus-3-Ergebnissignaturvoting vorgenommen.
Eine weitere vorteilhafte Verfahrensvariante (vgl. Fig. 4) ist, daß die erste Version beispielsweise 2 aus 7 bearbeitet wird, daß die zweite Version nur auf einem Rechner bearbeitet wird, daß anschließend der Vergleich der Ergebnissignaturen der beiden Versionen durchgeführt wird. Bei Übereinstimmung ist der Prozeß abgeschlossen. Bei Differenz bearbeiten die 7 Rechner die zweite Version und ein Rechner u.U. bereits die dritte Version usw.
Mit nachfolgendem 2-aus-3-Voting über die Ergebnisdatensignaturen wird ein Verfahren geschaffen, das mit wenigen Rechnern und mit geringer Zeitredundanz eine extrem hohe Fehlersicherheit bietet.
Aus diesen drei repräsentativen Beispielen sind weitere Varianten ableitbar.
Es existiert eine Grundkonfiguration mit M-Rechnern, in der sich als Untermenge die intakte Konfiguration (Menge aller arbeitsfähigen Rechner) mit K Rechnern zum Zeitpunkt t befindet.
In dieser Menge können je nach gefordertem Fehlertoleranzgrad ein oder mehrere Votingmengen (Votingkonfiguration) mit beispielsweise je annähernd N-Rechner organisiert werden.
Die Implementierung des Fehlertoleranzgrades kann als flexibler, dynamisch veränderbarer Prozeß aufgefaßt werden, wodurch eine optimale Auslastung der Ressourcen gewährleistet wird.
In einem Rechnersystem mit M-Rechnern seien zum Zeitpunkt t genau K-Rechner intakt. M-K-Rechner sind selbstisoliert und warten auf die Reparatur bzw. sind in Reparatur. Es stehen K-Rechner zur Verfügung.
Normalerweise können alle K-Rechner auch K verschiedene Tasks zur gleichen Zeit bearbeiten. Für hochzuverlässige Aufgaben können dagegen im Extremfall alle K-Rechner die gleiche Task bearbeiten und so ein 2-aus-K-System bilden.
Nimmt man beispielsweise ein Mehrrechnersystem mit 256 Rechnern an, so ist es vollkommen ausreichend wenn 8 Rechner, die verteilt im System liegen, eine hochzuverlässige Task bearbeiten und die restlichen mit gleichem, höherem oder niedrigem Redundanzgrad verteilt arbeiten.
Diese Verteilung der Zusammenarbeit kann in der nächsten Zeitscheibe vollkommen anders sein. Dieser hohe Grad an Flexibilität durch dynamische Anpassung ist sehr vorteilhaft für das Überleben eines Rechnersystems d.h. für seine hohe Verfügbarkeit.
Die räumliche Verteilung der Taskkopien ist dabei eine ergänzende Maßnahme zur Verbesserung der Fehlertoleranz.
Ganz gleich welche der 256 Rechner ausfallen, der erforderliche Redundanzgrad des 2-aus-8-Systems ist trotz einer hohen Fehlerzahl dynamisch softwaremäßig rekonfigurierbar.
Natürlich sollte die Anzahl fehlerhafter Rechner durch ereignisabhängige, prophylaktische (bei registrierten Unzuverlässigkeiten) oder durch die Missionszeit festgelegte Reparatur so gering wie möglich gehalten werden.
Bei Anwendung fehlertoleranter Rechnersysteme für Prozeßsteuerungen ist es außerdem sinnvoll, Meßwerte redundant zu erfassen, unter den Rechnern auszutauschen und zu vergleichen, um nur korrekte und identische aber fehlertolerant erfaßte Eingabewerte zu verwenden. Das ist eine sinnvolle Ergänzung zum Prozeßsignaturvoting.
Zur fehlertoleranten Meßgrößenerfassung wird eine Meßgröße von mehr als zwei unabhängigen Meßwertaufnehmern erfaßt und getrennt an mindestens zwei fehlerunabhängige Rechner verschaltet.
Die Ausgabedaten, die durch die Ergebnisdatensignatur gesichert werden, sollten auf fehlerunabhängige Ausgabekanäle unabhängiger Rechner und getrennt auf jeweils mehrere Stellgrößenschalter einer Stellgröße geschaltet werden, wobei beispielsweise für failsafe-eingeschaltet eine Parallelschaltung und für failsafe-ausgeschaltet eine Reihenschaltung zu verwenden ist.

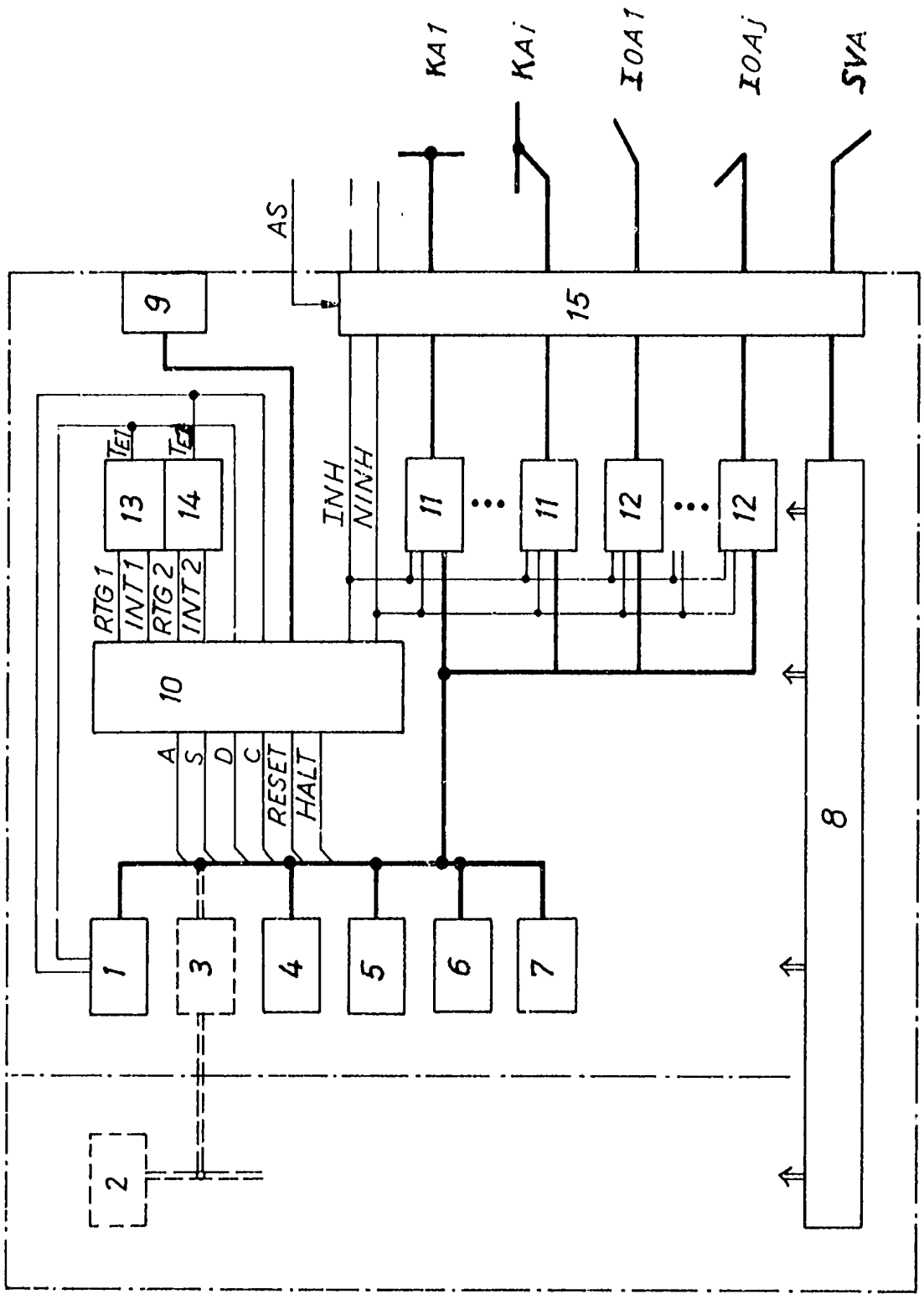
KA1
KAi
IOA1
IOAj
SVA
AS
9
15
IEI
IEA
13
14
INH
NINH
11
11
12
12
RTG 1
INT 1
RTG 2
INT 2
10
A
S
D
C
RESET
HALT
8
1
3
4
5
6
7
2

Fig. 1

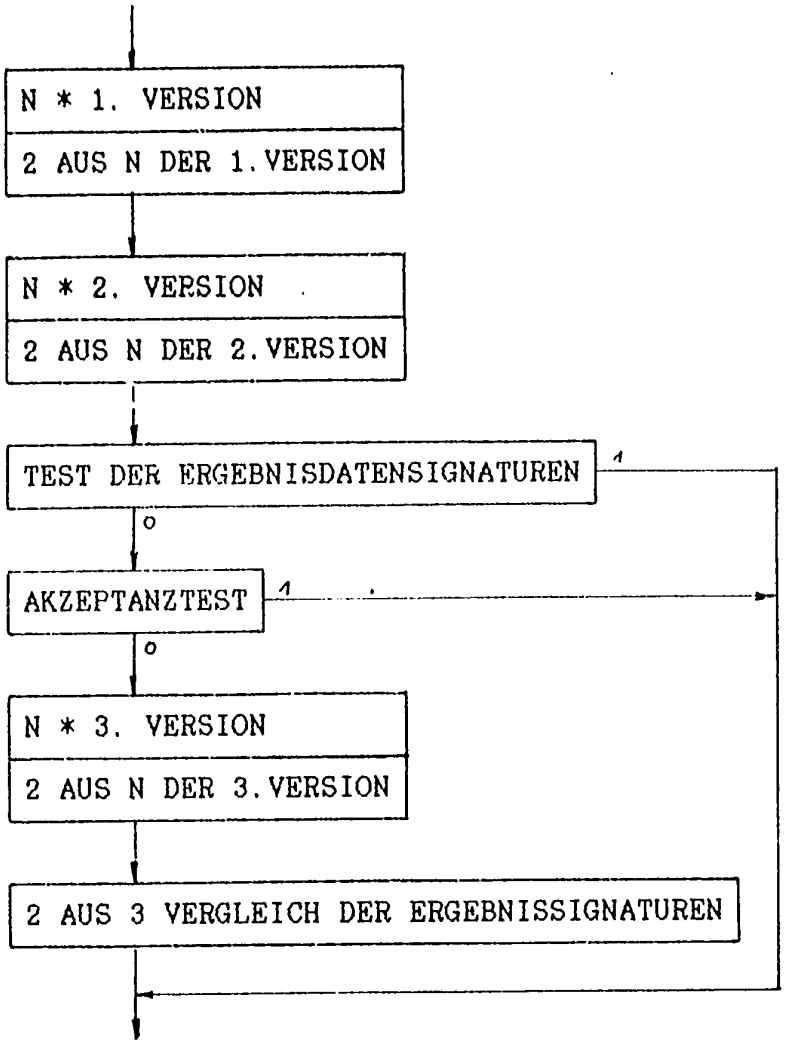
N * 1. VERSION
2 AUS N DER 1.VERSION
N * 2. VERSION
2 AUS N DER 2.VERSION
TEST DER ERGEBNISDATENSIGNATUREN
1
0
AKZEPTANZTEST
1
0
N * 3. VERSION
2 AUS N DER 3.VERSION
2 AUS 3 VERGLEICH DER ERGEBNISSIGNATUREN

FIG. 2

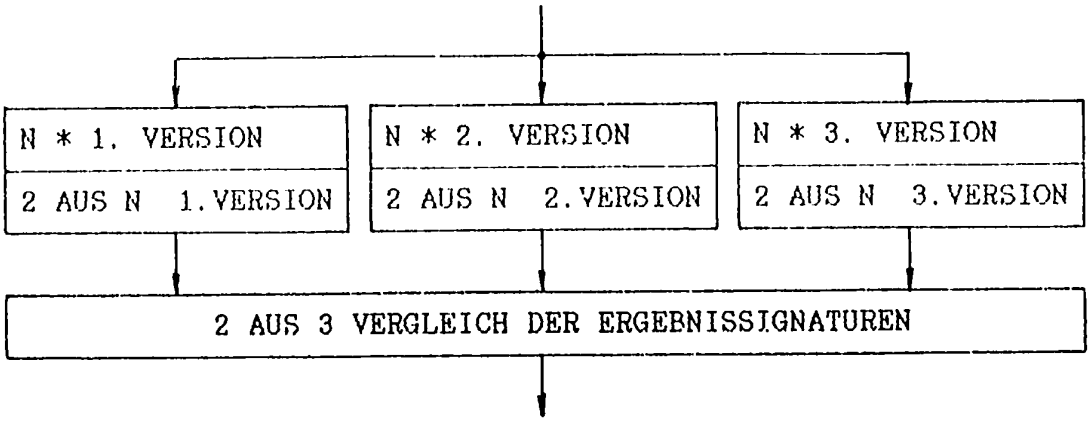
N * 1. VERSION
2 AUS N 1.VERSION
N * 2. VERSION
2 AUS N 2.VERSION
N * 3. VERSION
2 AUS N 3.VERSION
2 AUS 3 VERGLEICH DER ERGEBNISSIGNATUREN

FIG. 3

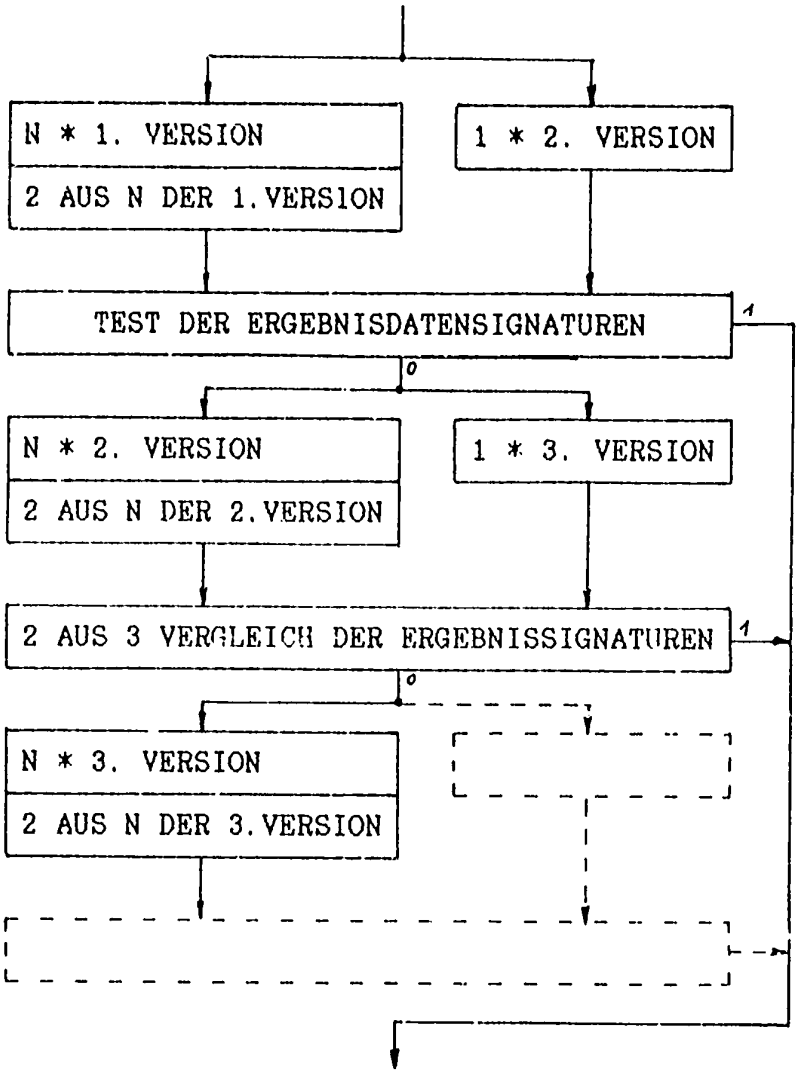
N * 1. VERSION
2 AUS N DER 1.VERSION
1 * 2. VERSION
TEST DER ERGEBNISDATENSIGNATUREN
1
0
N * 2. VERSION
2 AUS N DER 2.VERSION
1 * 3. VERSION
2 AUS 3 VERGLEICH DER ERGEBNISSIGNATUREN
1
0
N * 3. VERSION
2 AUS N DER 3.VERSION

FIG. 4