



(10) **DE 10 2017 130 552 B3** 2019.03.14

(12)

Patentschrift

(21) Aktenzeichen: **10 2017 130 552.1**

(22) Anmeldetag: **19.12.2017**

(43) Offenlegungstag: –

(45) Veröffentlichungstag
der Patenterteilung: **14.03.2019**

(51) Int Cl.: **G06F 9/50 (2006.01)**

Innerhalb von neun Monaten nach Veröffentlichung der Patenterteilung kann nach § 59 Patentgesetz gegen das Patent Einspruch erhoben werden. Der Einspruch ist schriftlich zu erklären und zu begründen. Innerhalb der Einspruchsfrist ist eine Einspruchsgebühr in Höhe von 200 Euro zu entrichten (§ 6 Patentkostengesetz in Verbindung mit der Anlage zu § 2 Abs. 1 Patentkostengesetz).

(73) Patentinhaber:
Beckhoff Automation GmbH, 33415 Verl, DE

(74) Vertreter:
Wilhelm & Beck, 80639 München, DE

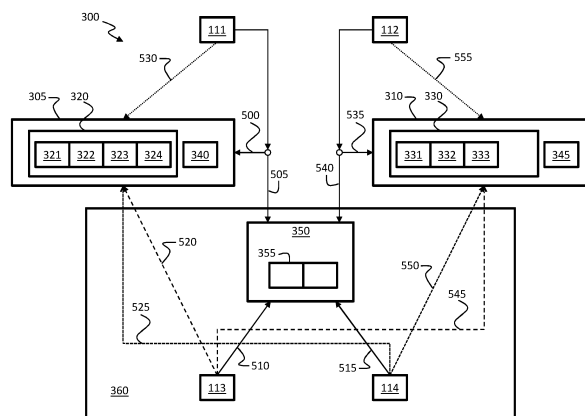
(72) Erfinder:
Vogt, Robin, 33415 Verl, DE

(56) Ermittelter Stand der Technik:

Robert I. Davis and Alan Burns. 2011. A survey of hard real-time scheduling for multiprocessor systems. ACM Comput. Surv. 43, 4, Article 35 (October 2011), 44 pages. DOI: <https://doi.org/10.1145/1978802.1978814>

(54) Bezeichnung: **Verfahren zur Datenverarbeitung und speicherprogrammierbare Steuerung**

(57) Zusammenfassung: Die Erfindung betrifft ein Verfahren 300 zur Datenverarbeitung auf einer speicherprogrammierbaren Steuerung (SPS). Wenigstens einem parallelen Bearbeitungsabschnitt eines Programms eines Hauptprozessorkerns 111, 112 einer Steuerungsaufgabe ist eine Priorität mit einer vorgegebenen Prioritätsstufe 340, 345 zugewiesen. Die jeweiligen Prioritätsstufen 340, 345 werden in eine Datenstruktur eingefügt, sobald der jeweilige Hauptprozessorkern 111, 112 im Programm an dem parallelen Bearbeitungsabschnitt angelangt ist. Wenigstens ein Parallelprozessorkern 113, 114 prüft, ob in der Datenstruktur Einträge vorhanden sind und sofern Einträge vorhanden sind, arbeitet er Teilaufgaben 321-324, 331-333 aus einem Arbeitspaket 305, 310 des Hauptprozessorkerns 111, 112 ab, dessen Prioritätsstufe 340, 345 an erster Stelle der Einträge in der Datenstruktur steht. Eine Echtzeitbedingung der Steuerungsaufgabe wird eingehalten, indem Ausführungszeiten der Programme für die jeweiligen Hauptprozessorkerne 111, 112 so festgesetzt sind, dass die mehreren Hauptprozessorkerne 111, 112 in der Lage sind, die Teilaufgaben 321-324, 331-333 aus den Arbeitspaketen 305, 310 ohne die Unterstützung des wenigstens einen Parallelprozessorkerns 113, 114 abzuarbeiten. Die mehreren Hauptprozessorkerne 111, 112 arbeiten ferner Teilaufgaben 321-324, 331-333 ab, die nicht von dem wenigstens einen Parallelprozessorkern 113, 114 abgearbeitet werden.



Beschreibung

[0001] Die vorliegende Erfindung betrifft ein Verfahren zur Datenverarbeitung auf einer speicherprogrammierbaren Steuerung (SPS). Die Erfindung betrifft des Weiteren eine speicherprogrammierbare Steuerung, die insbesondere zur Regelung oder Steuerung einer Maschine oder einer Anlage eingesetzt wird.

[0002] Maschinen oder Anlagen eines Automatisierungssystems werden häufig mithilfe speicherprogrammierbarer Steuerungen (SPS) gesteuert. Die SPS kann dabei als externes Gerät oder als Software-SPS vorliegen. Zur Steuerung oder Regelung der Aktoren und Sensoren der Maschine oder der Anlage nutzt die SPS meist eine Kommunikationsschnittstelle, die in Form eines Feldbussystems verwirklicht sein kann. Die Aktoren und Sensoren der Maschine oder der Anlage können dann über das Feldbussystem miteinander vernetzt sein.

[0003] Durch das Auslesen der Messdaten der Sensoren und/oder der aktuellen Ist-Daten der Aktoren, die mit den Eingängen der speicherprogrammierbaren Steuerung verbunden sind, erhält die SPS eine Information über den Status der Maschine oder der Anlage. Die Aktoren sind an die Ausgänge der speicherprogrammierbaren Steuerung angeschlossen und ermöglichen die Steuerung der Maschine oder der Anlage. Zur dynamischen Ansteuerung der Aktoren generiert die SPS auf Basis der Ist-Daten und/oder auf Basis der Messdaten der Sensoren die Ausgabedaten für die Aktoren, wobei sich die genannten Daten um einzelne Werte oder Wertegruppen handeln können. Ebenso kann die Ansteuerung der Aktoren aufgrund von Vorgaben, zum Beispiel anhand von Bewegungsprofilen erfolgen. Um die gewünschte Betriebsweise der Maschine bzw. der Anlage bereitstellen zu können, legt die Steuerungsaufgabe der SPS fest, welche seitens der SPS, in Abhängigkeit von entsprechenden Eingangsdaten, erzeugte Ausgangsdaten den Aktoren zugeführt werden. Die Verarbeitung der Daten durch die SPS erfolgt meist zyklisch und umfasst drei Schritte: Bereitstellen von aktuellen Eingabedaten (beispielsweise Ist-Daten der Aktoren und/oder Messdaten der Sensoren), Verarbeitung der Eingabedaten zu Ausgabedaten, und Ausgabe der Ausgabedaten für die Bewegungssteuerung (beispielsweise Zielpositionen, etc.).

[0004] Für die Steuerung des Automatisierungssystems ist es zwingend erforderlich, dass die Verarbeitung der Eingabedaten spätestens zu dem Zeitpunkt beendet ist, zu dem die Ausgabedaten für die Aktoren benötigt werden. Dieser Zeitpunkt bezeichnet die Deadline und entspricht gewöhnlich dem Ende eines Programmzyklus der SPS. Für die speicherprogrammierbare Steuerung wird in der Regel eine harte Echtzeitfähigkeit gefordert, was bedeutet, dass

Deadlines konsequent eingehalten und niemals überschritten werden dürfen. Zusätzlich muss zum Zeitpunkt der Deadline ein valides Ergebnis für die Aktoren vorliegen. Dies ist vor allem dann relevant, wenn das Überschreiten der Deadline zu Personen- oder Sachschäden führen kann, beispielsweise wenn ein Roboterarm nicht rechtzeitig abgebremst wird.

[0005] Die Programmausführung der SPS ist beispielsweise in der Norm IEC 61131-3 festgelegt. Die SPS besitzt eine Steuerungsaufgabe und diese setzt sich meist aus einem oder mehreren im Allgemeinen zyklisch auszuführenden Programmen mit Aufgaben, den sogenannten „Tasks“ zusammen. Im einfachsten Fall weist die speicherprogrammierbare Steuerung nur einen einzelnen Prozessorkern für mehrere unabhängige Programme mit Aufgaben (Tasks) auf. Deshalb muss die verfügbare Rechenzeit derart aufgeteilt werden, dass alle Programme mit entsprechenden Aufgaben ihre Deadlines einhalten können. Die verschiedenen Zykluszeiten der Programme in der SPS können sich beispielsweise je nach Anwendung in einem Bereich von 100 µs bis 20 ms oder bei komplexen Aufgaben in einem Bereich von 50 ms bis 100 ms befinden. Aufgrund der verschiedenen Zykluszeiten der Programme gilt es zu verhindern, dass Programme mit Aufgaben mit einer längeren Zykluszeit den Start von Programmen mit Aufgaben mit einer kürzeren Zykluszeit verzögern, so dass diese ihre Deadlines nicht mehr einhalten können. Häufig kann dies nur erreicht werden, wenn die Ausführung eines solchen langsamen Programms unterbrochen und später fortgesetzt wird. Aus diesem Grund wird jedem Programm mit entsprechenden Aufgaben gewöhnlich eine vorgegebene Prioritätsstufe zugeteilt. Diese kann aus den Abhängigkeiten der jeweiligen Programme zu den anderen Programmen ermittelt werden, wenn die Programme beispielsweise auf die Ergebnisse der anderen Programme angewiesen sind.

[0006] Da die meisten SPS Prozessoren mehrere Prozessorkerne und/oder mehrere Prozessoren umfassen, kann die SPS Programme mit entsprechenden Aufgaben auf mehrere Prozessorkerne oder Prozessoren aufteilen, um die Gesamtverarbeitungszeit der Programme mit den entsprechenden Aufgaben zu reduzieren. Aus der DE 10 2012 216 568 B4 ist ein Verfahren zum Ausführen von Programmen mit entsprechenden Aufgaben auf mehreren Prozessorkernen bekannt, wobei die Programme in Programm-Gruppen organisiert sind und der jeweiligen Programmgruppe eine Prioritätsstufe zugeteilt sein kann. Innerhalb einer Programmgruppe ist die Prioritätsstufe der Programme gleich, das heißt es werden keine individuellen Prioritätsstufen für die einzelnen Programme berücksichtigt. Die Programme können zugunsten anderer Programme mit einer höheren Prioritätsstufe auf den parallel arbeitenden Prozessorkernen unterbrochen werden und später auf

den parallel arbeitenden Prozessorkernen fortgesetzt werden.

[0007] Aus Robert I. Davis and Alan Burns. 2011. A survey of hard real-time scheduling for multiprocessor systems. ACM Comput. Surv. 43, 4, Article 35 (October 2011) ist ein Verfahren zur Datenverarbeitung auf einer speicherprogrammierbaren Steuerung, die mehrere Prozessorkerne umfasst. Eine Steuerungsaufgabe kann dabei Teilaufgaben aufweisen, die parallel von den Prozessorkernen ausgeführt werden, wobei eine Echtzeitbedingung der Steuerungsaufgabe eingehalten wird.

[0008] Die Aufgabe der vorliegenden Erfindung besteht darin ein Verfahren zur Datenverarbeitung weiterzuentwickeln, das bei Echtzeitsystemen eingesetzt werden kann und die benötigte Rechenzeit reduziert sowie die vorhandenen Ressourcen optimal nutzt. Es ist weiterhin Aufgabe der Erfindung, eine verbesserte speicherprogrammierbare Steuerung anzugeben.

[0009] Diese Aufgabe wird durch die Merkmale der unabhängigen Ansprüche gelöst. Weitere vorteilhafte Ausführungsformen der Erfindung sind in den abhängigen Ansprüchen angegeben.

[0010] Erfindungsgemäß wird ein Verfahren zur Datenverarbeitung auf einer speicherprogrammierbaren Steuerung vorgeschlagen. Die SPS umfasst mehrere Hauptprozessorkerne und wenigstens einen Parallelprozessorkern, wobei den Hauptprozessorkernen jeweils ein ausführbares Programm einer Steuerungsaufgabe zugeordnet ist. Jedes Programm weist jeweils wenigstens einen parallelen Bearbeitungsabschnitt mit einem Arbeitspaket auf, das mehrere Teilaufgaben umfasst. Den parallelen Bearbeitungsabschnitten in den jeweiligen Programmen ist eine Priorität mit einer vorgegebenen Prioritätsstufe zugewiesen. Die jeweiligen Prioritätsstufen werden in eine Datenstruktur eingefügt, sobald der jeweilige Hauptprozessorkern im Programm an dem parallelen Bearbeitungsabschnitt angelangt ist. Der wenigstens eine Parallelprozessorkern prüft, ob in der Datenstruktur Einträge vorhanden sind. Sofern Einträge in der Datenstruktur vorhanden sind, arbeitet der wenigstens eine Parallelprozessorkern Teilaufgaben aus dem Arbeitspaket des Hauptprozessorkerns ab, dessen Prioritätsstufe an erster Stelle der Einträge in der Datenstruktur steht. Eine Echtzeitbedingung der Steuerungsaufgabe wird eingehalten, indem Ausführungszeiten der Programme für die jeweiligen Hauptprozessorkerne so festgesetzt sind, dass die mehreren Hauptprozessorkerne in der Lage sind, die Teilaufgaben aus den Arbeitspaketen ohne die Unterstützung des wenigstens einen Parallelprozessorkerns abzuarbeiten. Die mehreren Hauptprozessorkerne arbeiten ferner Teilaufgaben ab, die nicht von dem wenigstens

einen Parallelprozessorkern abgearbeitet werden.

[0011] Durch die Verteilung einzelner Aufgaben der entsprechenden Programme der Hauptprozessorkerne auf die mehreren Hauptprozessorkerne und den wenigstens einen Parallelprozessorkern kann die Gesamtbearbeitungszeit der Aufgaben und die Ausführungszeit der jeweiligen Programme reduziert werden. Die Aufgaben selbst können zur Reduzierung der Ausführungszeit in Arbeitspaketen mit unabhängigen Teilaufgaben untergliedert sein und im parallelen Bearbeitungsabschnitt des ausführbaren Programms der Steuerungsaufgabe auf dem entsprechenden Hauptprozessorkern und dem wenigstens einen Parallelprozessorkern abgearbeitet werden. Somit ist es beispielsweise möglich, die freie Rechenkapazität für größere oder komplexere Aufgaben zu nutzen, um die Deadlines einhalten zu können. Die mehreren Hauptprozessorkerne können jeweils nur die Teilaufgaben ihrer eigenen Arbeitspakete abarbeiten und nicht die Teilaufgaben von Programmen auf den anderen Hauptprozessorkernen, wohingegen der wenigstens eine Parallelprozessorkern die Teilaufgaben von allen Hauptprozessorkernen abarbeiten kann. Demnach bietet eine solche Ausgestaltung den Vorteil, dass die vorhandenen Ressourcen optimal genutzt werden, da die mehreren Hauptprozessorkerne und der wenigstens eine Parallelprozessorkern zusammen an der Abarbeitung der Teilaufgaben mitwirken.

[0012] Die Verteilung der Teilaufgaben der jeweiligen Programme der Hauptprozessorkerne auf die mehreren Hauptprozessorkerne und den wenigstens einen Parallelprozessorkern kann die Datenverarbeitung der SPS weiterhin echtzeitfähig gestalten. Denn durch die Aufgabenverteilung können die Deadlines der jeweiligen Programme optimal eingehalten werden. Von Vorteil ist, dass es durch die Parallelisierung für die am höchsten priorisierten Programme der mehreren Hauptprozessorkerne möglich ist, die kürzere Ausführungszeit der Programme durch die Parallelisierung einzuplanen, und auf diese Weise eine Deadline, die kleiner als eine rein sequentielle Ausführungszeit der Programme ist, einzuhalten. Beispielsweise kann eine Maschine oder Anlage eines Automatisierungssystems dadurch schneller arbeiten.

[0013] In einer weiteren Ausführungsform erfolgt die Verteilung der Teilaufgaben dynamisch. Die Hauptprozessorkerne und der wenigstens eine Parallelprozessorkern greifen auf unbearbeitete Teilaufgaben der Arbeitspakete zur Abarbeitung zu, wobei die Teilaufgaben, die dem wenigstens einen Parallelprozessorkern für die Bearbeitung zugeteilt werden, auch auf dem entsprechenden Hauptprozessorkern ausführbar sind. Die dynamische Verteilung der Teilaufgaben bietet die Möglichkeit, die verfügbare Rechen-

kapazität noch besser zu nutzen, als im Fall der statischen Verteilung. Denn die Hauptprozessorkerne und der wenigstens eine Parallelprozessorkern können selbst aktiv beim Arbeitspaket nach freien Teilaufgaben anfragen und den Zugriff auf unbearbeitete Teilaufgaben vom Arbeitspaket erhalten. Ist eine Teilaufgabe vom zugeordneten Hauptprozessorkern fertig abgearbeitet worden und der wenigstens eine Parallelprozessorkern beispielsweise gerade mit einer anderen Teilaufgabe ausgelastet, so kann (und soll) der entsprechende Hauptprozessorkern vom Arbeitspaket eine neue, unbearbeitete Teilaufgabe anfordern und diese abarbeiten, unabhängig davon, was der wenigstens eine Parallelprozessorkern gerade abarbeitet. Auf diese Weise kann die ungenutzte Rechenkapazität für eine frühzeitige Abarbeitung der Teilaufgaben genutzt werden, um die Gesamtbearbeitungszeit des jeweiligen Programms der Steuerungsaufgabe zu reduzieren und die Deadlines einzuhalten.

[0014] In einer weiteren Ausführungsform besteht nur für den die entsprechende Teilaufgabe aktuell abarbeitenden Hauptprozessorkern oder den die entsprechende Teilaufgabe aktuell abarbeitenden wenigstens einen Parallelprozessorkern ein Zugriff auf die entsprechende Teilaufgabe aus einem der Arbeitspakete. Die Teilaufgaben aus den Arbeitspaketen werden einzeln und nacheinander an die Hauptprozessorkerne und den wenigstens einen Parallelprozessorkern verteilt. Hat nur der die entsprechende Teilaufgabe aktuell abarbeitende Hauptprozessorkern oder der die entsprechende Teilaufgabe aktuell abarbeitende wenigstens eine Parallelprozessorkern Zugriff auf die entsprechende Teilaufgabe, so kann eine unnötige Mehrfachabarbeitung derselben Teilaufgabe vermieden werden. Zudem wird eine Mehrfachverteilung der Teilaufgaben unterbunden, indem die Verteilung der Teilaufgaben einzeln und nacheinander erfolgt.

[0015] In einer weiteren Ausführungsform erfolgt die Verteilung der Teilaufgaben aus den jeweiligen Arbeitspaketen statisch. Dazu wird den Hauptprozessorkernen und dem wenigstens einen Parallelprozessorkern jeweils eine bestimmte Anzahl an Teilaufgaben aus dem gleichen Arbeitspaket zur parallelen Abarbeitung fest zugeteilt. In dieser Ausgestaltung entfällt für den wenigstens einen Parallelprozessorkern die zusätzliche Rechenzeit zum Anfragen der Teilaufgaben aus den Arbeitspaketen. Durch die entsprechende feste Zuteilung, die beispielsweise zu Beginn der Programmausführung oder zu Beginn des parallelen Bearbeitungsabschnitts des Programms erfolgen kann, ist es also möglich den Verwaltungsaufwand zu reduzieren.

[0016] In einer weiteren Ausführungsform werden dem wenigstens einen Parallelprozessorkern Änderungen in der Reihenfolge der Einträge der Prioritätsstufen in der Datenstruktur mitgeteilt.

In der aktiven Ausgestaltung wird der wenigstens eine Parallelprozessorkern über Änderungen in der Reihenfolge der Einträge der Prioritätsstufen an der ersten Stelle in der Datenstruktur informiert, damit die aktuell bearbeitete Teilaufgabe unterbrochen wird und nach erfolgter Anfrage einer neuen Teilaufgabe aus dem Arbeitspaket des Hauptprozessorkerns mit höherer Prioritätsstufe mit deren Abarbeitung begonnen werden kann. Alternativ dazu ist auch eine passive Ausgestaltung denkbar, in der der wenigstens eine Parallelprozessorkern selbst anfragt, ob sich die Reihenfolge der Einträge der Prioritätsstufen an der ersten Stelle der Einträge in der Datenstruktur geändert hat. Der wenigstens eine Parallelprozessorkern kann gegebenenfalls die Berechnungen, also die Abarbeitung der aktuellen Teilaufgabe, unterbrechen.

[0017] In einer weiteren Ausführungsform werden die Berechnungen aktueller Teilaufgaben im parallelen Bearbeitungsabschnitt auf dem wenigstens einen Parallelprozessorkern unterbrochen, wenn ein neuer Eintrag einer Prioritätsstufe an die erste Stelle der Datenstruktur gesetzt wird. Durch die Unterbrechung der Berechnungen von aktuellen Teilaufgaben auf dem wenigstens einen Parallelprozessorkern können die Deadlines der Programme eingehalten und die Echtzeitdatenverarbeitung ermöglicht werden. Ferner kann durch eine solche Ausgestaltung die Rechenkapazität besser ausgenutzt werden, denn es ist beispielsweise möglich, die vom wenigstens einen Parallelprozessorkern unterbrochene Teilaufgabe auf dem entsprechenden Hauptprozessorkern ab dem Unterbrechungspunkt weiter zu berechnen, während der wenigstens eine Parallelprozessorkern mit der Abarbeitung der neuen Teilaufgabe beginnt. Alternativ dazu kann die unterbrochene Teilaufgabe auch nach Fertigstellung der Teilaufgaben aus dem Arbeitspaket mit höherer Prioritätsstufe des parallelen Bearbeitungsabschnitts auf dem wenigstens einen Parallelprozessorkern weiter abgearbeitet werden.

[0018] In einer weiteren Ausführungsform werden die aktuellen Teilaufgaben von dem wenigstens einen Parallelprozessorkern bis zu einem vorgegebenen Unterbrechungspunkt weiter bearbeitet und dann unterbrochen. Eine solche Ausgestaltung bietet den Vorteil, dass weniger Daten gesichert werden müssen, da bekannt ist, welche Daten bei einer Fortführung der unterbrochenen Teilaufgabe benötigt werden. Auch erfordert die Implementierung der Ausgestaltung einen geringeren Eingriff in das System, als er beispielsweise bei einem sofortigen Unterbrechen der Teilaufgabe auf dem wenigstens einen Parallelprozessorkern und dem anschließenden Weiterausführen auf dem jeweiligen Hauptprozessorkern erforderlich sein kann.

[0019] In einer weiteren Ausführungsform ist der Unterbrechungspunkt insbesondere in einer Kontrollstruktur in den Teilaufgaben der Arbeitspakete in den entsprechenden Programmen der Hauptprozessorkerne festgelegt. Der Unterbrechungspunkt kann zum Beispiel zu Beginn oder am Ende eines Durchlaufs der Kontrollstruktur gesetzt sein. Die Kontrollstruktur kann als Schleife verwirklicht sein, da bestimmte Instruktionen der entsprechenden Programme der Hauptprozessorkerne mit verschiedenen Variablenwerten mehrmals iterativ ausgeführt werden können. Folglich gestaltet das Setzen des Unterbrechungspunkts in der Kontrollstruktur in den Teilaufgaben der Arbeitspakete die Datensicherung einfach und kompakt, da beispielsweise nur die aktuelle Iteration und dazu gegebenenfalls vorhandene Zwischenergebnisse der Berechnungen gespeichert werden müssen.

[0020] In einer weiteren Ausführungsform wird ein Zwischenergebnis der Berechnungen der aktuellen Teilaufgaben der Arbeitspakete gesichert. Wird das Zwischenergebnis der Berechnungen der Teilaufgaben gesichert, so müssen unterbrochene Teilaufgaben nicht mehr von Beginn an auf dem jeweiligen Hauptprozessorkern oder dem wenigstens einen Parallelprozessorkern berechnet werden, da das bisherige Zwischenergebnis nicht verloren ist. In der Regel ist es jedoch nicht bestimmbar, welche Daten für eine Fortführung der Teilaufgabe aus dem entsprechenden Arbeitspaket des zugeordneten Hauptprozessorkerns nach der Unterbrechung relevant sind, da die Teilaufgabe an einer beliebigen Stelle im Programm des Hauptprozessorkerns während der Ausführungszeit des Programms unterbrochen wird. Insofern kann diese Ausgestaltung die Sicherung sämtlicher Daten erfordern, die bei der Berechnung der Teilaufgabe im entsprechenden Programm des zugeordneten Hauptprozessorkerns erstellt bzw. bearbeitet wurden, wenn kein entsprechender Unterbrechungspunkt in den Teilaufgaben gesetzt wurde. Denn mithilfe des vorgegebenen Unterbrechungspunkts, der beispielsweise in der Kontrollstruktur in der Teilaufgabe gesetzt sein kann, ist bekannt, welche Daten bei der Fortführung der unterbrochenen Teilaufgabe benötigt werden. Im Fall des gesetzten Unterbrechungspunkts müssen nicht sämtliche Daten gesichert werden, sondern es ist ausreichend, die aktuelle Iteration der Kontrollstruktur, die sich um eine Schleife handeln kann, und gegebenenfalls vorhandene Zwischenergebnisse der Berechnungen zu speichern.

[0021] Eine weitere Ausführungsform sieht vor, dass die Berechnungen der Teilaufgaben auf dem wenigstens einen Parallelprozessorkern umgehend abgebrochen werden und dabei keine Zwischenergebnisse der Berechnungen gesichert werden. Folglich muss eine abgebrochene Teilaufgabe wieder von Beginn an neu berechnet werden. Von Vorteil ist dabei,

dass mithilfe der Ausgestaltung eine geringste mögliche Latenz erzielt werden kann, wobei die Latenz die Zeit zwischen dem Abbruch der Berechnungen der aktuellen Teilaufgabe auf dem wenigstens einen Parallelprozessorkern und dem Beginn der Berechnungen der anderen Teilaufgabe auf dem wenigstens einen Parallelprozessorkern, die mit einer höheren Prioritätsstufe verknüpft ist, angibt.

[0022] Erfindungsgemäß wird des Weiteren eine speicherprogrammierbare Steuerung (SPS) vorgeschlagen. Die SPS weist eine Kommunikationsschnittstelle, zum Einlesen von Sensordaten und zum Ausgeben von Aktordaten auf. Weitere Bestandteile der SPS sind eine Datenverarbeitungseinrichtung, die mehrere Hauptprozessorkerne und wenigstens einen Parallelprozessorkern zum Ausführen einer Steuerungsaufgabe umfasst, um aus den Sensordaten Aktordaten zu erzeugen, und eine Steuerungseinrichtung, die einen Speicher zum Speichern einer Datenstruktur mit Prioritätsstufen und einen Prioritätenverwalter zum Verwalten der Prioritätsstufen in der Datenstruktur aufweist. Den Hauptprozessorkernen ist jeweils ein ausführbares Programm der Steuerungsaufgabe zugeordnet, wobei jedes Programm jeweils einen parallelen Bearbeitungsabschnitt mit einem Arbeitspaket aufweist, und das Arbeitspaket mehrere Teilaufgaben umfasst. Den parallelen Bearbeitungsabschnitten in den jeweiligen Programmen ist die vorgegebene Prioritätsstufe zugewiesen. Der Prioritätenverwalter ist ausgebildet, die jeweiligen Prioritätsstufen in die Datenstruktur im Speicher einzufügen, sobald der jeweilige Hauptprozessorkern im Programm an dem parallelen Bearbeitungsabschnitt angelangt ist und dabei den Eintrag mit der höchsten Prioritätsstufe an die erste Stelle der Datenstruktur im Speicher zu setzen. Der wenigstens eine Parallelprozessorkern ist dazu ausgelegt, die Datenstruktur im Speicher auf Einträge der Prioritätsstufen zu prüfen. Der wenigstens eine Parallelprozessorkern ist, sofern Einträge in der Datenstruktur vorhanden sind, ferner ausgebildet, die Teilaufgaben aus dem Arbeitspaket des Hauptprozessorkerns abzuarbeiten, dessen Prioritätsstufe an erster Stelle der Einträge in der Datenstruktur steht. Die speicherprogrammierbare Steuerung ist des Weiteren ausgebildet, eine Echtzeitbedingung der Steuerungsaufgabe einzuhalten, indem Ausführungszeiten der Programme für die jeweiligen Hauptprozessorkerne so festgesetzt sind, dass die mehreren Hauptprozessorkerne in der Lage sind, die Teilaufgaben aus den Arbeitspaketen ohne die Unterstützung des wenigstens einen Parallelprozessorkerns abzuarbeiten. Weiterhin sind die mehreren Hauptprozessorkerne dazu ausgelegt, Teilaufgaben abzuarbeiten, die nicht von dem wenigstens einen Parallelprozessorkern abgearbeitet werden.

[0023] Speicherprogrammierbare Steuerungen bilden häufig die Grundlage eines Automatisierungs-

systems und verfügen derzeit über komplexe Rechen- und Regelfunktionen, die eine leistungsstarke Hardware und effiziente Software zur Umsetzung erfordern. Die hohe Rechenleistung kann dabei auf der gleichzeitigen Nutzung von mehreren Hauptprozessorkernen und dem wenigstens einen Parallelprozessorkern basieren, um das System effizient zu betreiben und die vorhandenen Ressourcen bestmöglich auszulasten. Die so konzipierte SPS ermöglicht es, die Deadlines der Programme einzuhalten und sie kann ferner die Echtzeitdatenverarbeitung gewähren.

[0024] In einer weiteren Ausführungsform ist die Steuerungseinrichtung auf mehreren Hauptprozessorkernen verteilt ausgeführt. Diese Ausgestaltung bietet die Möglichkeit die Rechenkapazität flexibel zu nutzen und anderen Anwendungen dabei weiterhin Zugriff auf die Hauptprozessorkerne zu gewähren. Der Prioritätenverwalter, der Bestandteil der Steuerungseinrichtung ist, kann beispielsweise mehrere Module umfassen, wobei die einzelnen Module des Prioritätenverwalters auf den mehreren Hauptprozessorkernen ausgeführt werden können. Denkbar ist weiterhin, dass der Prioritätenverwalter nur ein einziges zentrales Modul aufweist, das auf einem der mehreren Hauptprozessorkerne ausgeführt wird.

[0025] In einer weiteren Ausführungsform ist der Prioritätenverwalter ausgelegt, den ersten Eintrag der Prioritätsstufe der mehreren Hauptprozessorkerne der Datenstruktur durch einen neuen Eintrag zu ersetzen, wenn dieser eine höhere Prioritätsstufe aufweist. Der Prioritätenverwalter ist ferner ausgebildet, den wenigstens einen Parallelprozessorkern über die Änderung der Einträge der Datenstruktur zu informieren, um aktuelle Berechnungen von Teilaufgaben aus dem entsprechenden Arbeitspaket zu unterbrechen. Der Prioritätenverwalter gestaltet somit die Koordination der Einträge der Prioritätsstufen in der Datenstruktur und die Kommunikation mit dem wenigstens einen Parallelprozessorkern einfach.

[0026] Eine weitere Ausführungsform der Erfindung sieht vor, dass der wenigstens eine Parallelprozessorkern ausgelegt ist, die Berechnungen der aktuellen Teilaufgaben aus dem entsprechenden Arbeitspaket zu unterbrechen, wenn ein neuer Eintrag einer Prioritätsstufe an die erste Stelle der Einträge in der Datenstruktur im Speicher gesetzt wird und die Zwischenergebnisse der Berechnungen an die Steuerungseinrichtung zu übergeben. Die Steuerungseinrichtung ist dazu ausgebildet, die Zwischenergebnisse der Berechnungen der aktuellen Teilaufgaben aus den jeweiligen Arbeitspaketen auf einer Speichereinheit zu sichern, auf die die mehreren Hauptprozessorkerne und der wenigstens eine Parallelprozessorkern zugreifen. Durch die Steuerungseinrichtung wird die Sicherung der Zwischenergebnisse der Berechnungen übersichtlich gestaltet, da die Daten zentral auf einer Speichereinheit gesichert werden, auf die

sowohl die Hauptprozessorkerne und der wenigstens eine Parallelprozessorkern zugreifen können.

[0027] In einer weiteren Ausführungsform ist der Prioritätenverwalter ausgelegt, die Einträge der Prioritätsstufen der mehreren Hauptprozessorkerne aus der Datenstruktur zu entfernen, wenn die Teilaufgaben aus den entsprechenden Arbeitspaketen vollständig abgearbeitet sind. Durch eine solche Ausgestaltung des Prioritätenverwalters kann der Speicher der Steuerungseinrichtung einfach aktualisiert werden.

[0028] Die vorstehend erläuterten und/oder in den Unteransprüchen wiedergegebenen vorteilhaften Aus- und Weiterbildungen der Erfindung können - außer zum Beispiel in Fällen eindeutiger Abhängigkeiten oder unvereinbarer Alternativen - einzeln oder aber auch in beliebiger Kombination miteinander zur Anwendung kommen.

[0029] Die oben beschriebenen Eigenschaften, Merkmale und Vorteile dieser Erfindung, sowie die Art und Weise, wie diese erreicht werden, werden klarer und deutlicher verständlich in Zusammenhang mit der folgenden Beschreibung von Ausführungsbeispielen, die im Zusammenhang mit den schematischen Zeichnungen näher erläutert werden. Es zeigen:

Fig. 1 eine schematische Darstellung einer erfindungsgemäßen Ausführungsform eines Automatisierungssystems mit einer speicherprogrammierbaren Steuerung (SPS);

Fig. 2 ein Schema eines Verfahrens zur Datenverarbeitung auf der SPS gemäß **Fig. 1**;

Fig. 3 ein zeitliches Ablaufschema eines Verfahrens zur Datenverarbeitung auf der SPS gemäß **Fig. 1**;

Fig. 4 ein Programmablaufdiagramm für einen Hauptprozessorkern in der in **Fig. 1** gezeigten SPS; und

Fig. 5 ein Programmablaufdiagramm für einen Parallelprozessorkern in der in **Fig. 1** gezeigten SPS.

[0030] Anhand der folgenden Figuren wird ein Ausführungsbeispiel eines Verfahrens zur Datenverarbeitung auf einer speicherprogrammierbaren Steuerung (SPS) beschrieben. Es wird darauf hingewiesen, dass die Figuren lediglich schematischer Natur und nicht maßstabsgetreu sind. In diesem Sinne können in den Figuren gezeigte Komponenten und Bauteile zum besseren Verständnis übertrieben groß oder verkleinert dargestellt sein. Ferner wird darauf hingewiesen, dass die Bezugszeichen in den Figuren unverändert gewählt worden sind, wenn es sich um gleich ausgebildete Bauteile und/oder Komponenten und/oder Größen handelt.

[0031] Speicherprogrammierbare Steuerungen (SPS) bilden häufig die Grundlage eines Automatisierungssystems und dienen zur Steuerung oder Regelung einer Anlage oder einer Maschine. Die Ansteuerung der SPS bzw. der Anlage oder der Maschine erfolgt dabei über Sensoren und Aktoren. Die Sensoren und Aktoren können über eine Kommunikationsschnittstelle mit der Maschine oder der Anlage vernetzt sein, wobei die SPS diese Kommunikationsschnittstelle nutzt, um mit den Sensoren und Aktoren zu interagieren. Die Datenverarbeitung der SPS ist meist zyklisch und erfordert von der SPS, genauer gesagt von der Steuerungsaufgabe, dass die Deadlines konsequent eingehalten werden. Es darf in diesem Zusammenhang zum Beispiel nicht vorkommen, dass zu dem Zeitpunkt zu dem die Ausgabedaten für die Aktoren benötigt werden, die Verarbeitung der Eingabedaten noch nicht abgeschlossen ist. Die Steuerungsaufgabe muss also die Echtzeitfähigkeit des Automatisierungssystems gewährleisten können.

[0032] Meist setzt sich die Steuerungsaufgabe der SPS aus einer oder mehreren zyklisch auszuführenden Programmen mit Aufgaben, den „Tasks“ zusammen. Da die meisten SPS Prozessoren mehrere Prozessorkerne, die zum Beispiel als Haupt- und Parallelprozessorkerne verwirklicht sein können, und/oder mehrere Prozessoren umfassen, kann die SPS die Programme mit Aufgaben (Tasks) auf mehrere Prozessorkerne und/oder mehrere Prozessoren aufteilen, um die Gesamtverarbeitungszeit der Programme mit Aufgaben zu reduzieren und die Einhaltung der Deadlines der Programme zu sichern. Werden die Aufgaben der Programme von mehreren Prozessorkernen (Hauptprozessorkerne und Parallelprozessorkerne) parallel abgearbeitet, so ist es sinnvoll, die Aufgaben der Programme für die Abarbeitung einer vorgegebenen Prioritätsstufe entsprechend klassifizieren zu können. Aus diesem Grund wird jedem Programm mit entsprechenden Aufgaben gewöhnlich eine vorgegebene Prioritätsstufe zugeteilt. Diese kann aus den Abhängigkeiten der jeweiligen Programme zu den anderen Programmen ermittelt werden, wenn die Programme beispielsweise auf die Ergebnisse der anderen Programme angewiesen sind.

[0033] Die Prioritätsstufe des entsprechenden Programms mit Aufgaben wird von einem Prioritätenverwalter in einer Datenstruktur organisiert. Der Prioritätenverwalter legt den Eintrag der Prioritätsstufe in der Datenstruktur ab. Sofern kein weiterer Eintrag einer Prioritätsstufe in der Datenstruktur vorhanden ist, setzt der Prioritätenverwalter die genannte Prioritätsstufe an die erste Stelle in der Datenstruktur. Der Prioritätenverwalter kann ferner dazu ausgebildet sein, die parallel arbeitenden Prozessorkerne darüber in Kenntnis zu setzen, wenn sich die Reihenfolge der Einträge der Prioritätsstufen in der Da-

tenstruktur ändert. Beispielsweise kann sich die Reihenfolge ändern, wenn ein neuer Eintrag einer Prioritätsstufe in der Datenstruktur hinzukommt, wobei die Prioritätsstufe dann einen höheren Wert als die Prioritätsstufe des bisherigen Eintrags haben kann. In einem solchen Fall können die Parallelprozessorkerne die Abarbeitung der aktuellen Aufgaben des entsprechenden Programms unterbrechen und mit der Abarbeitung der Aufgaben beginnen, die mit einer höheren Prioritätsstufe des entsprechenden Programms korrespondieren, um dessen Deadlines einzuhalten. Die Aufgaben eines entsprechenden Programms können in mehrere Arbeitspakete mit Teilaufgaben untergliedert sein.

[0034] Auf den Hauptprozessorkernen wird das jeweilige Programm der Steuerungsaufgabe der SPS ausgeführt, wobei die Ausführungszeiten der Programme so festgelegt sind, dass die Hauptprozessorkerne die Teilaufgaben ohne die Unterstützung der Parallelprozessorkerne abarbeiten können. Zur Reduzierung der Ausführungszeiten der Programme können die Parallelprozessorkerne für die Abarbeitung der Teilaufgaben jedoch mit genutzt werden. Der Kern der Erfindung liegt in der Möglichkeit der parallelen Abarbeitung der Teilaufgaben durch die Hauptprozessorkerne und die Parallelprozessorkerne, um die Deadlines der jeweiligen Programme der Hauptprozessorkerne einhalten und die Echtzeitfähigkeit des Automatisierungssystems gewähren zu können. Ferner kann eine optimale Ausnutzung der Ressourcen bereitgestellt werden, indem die Hauptprozessorkerne und die Parallelprozessorkerne gemeinsam Teilaufgaben abarbeiten.

[0035] Fig. 1 zeigt eine speicherprogrammierbare Steuerung (SPS) **100** zur Steuerung einer Maschine bzw. Anlage **200**. Die SPS **100** weist eine Kommunikationsschnittstelle **130** sowie eine Datenverarbeitungseinrichtung **110** auf. Über die Kommunikationsschnittstelle **130**, die beispielsweise in Form eines hardware- und/oder softwaremäßig ausgebildeten Feldbusmasters eines Feldbussystems verwirklicht sein kann, kann die SPS **100** die entsprechenden Aktoren **210** und Sensoren **220** der Maschine bzw. der Anlage **200** des Automatisierungssystems steuern, wobei die Aktoren **210** und Sensoren **220** mithilfe des Feldbussystems miteinander vernetzt sein können.

[0036] Um die gewünschte Betriebsweise der Maschine bzw. der Anlage **200** bereitstellen zu können, legt die Steuerungsaufgabe der SPS **100** fest, welche seitens der SPS **100** in Abhängigkeit von entsprechenden Eingangsdaten erzeugte Ausgangsdaten den Aktoren **210** zugeführt werden. Die SPS **100** erhält eine Information über den Zustand der Maschine bzw. der Anlage **200**, indem sie die Messdaten der Sensoren **220** und/oder die Ist-Daten der Aktoren **210** ausliest, die entsprechend mit den Eingängen

der SPS **100** verbunden sind. Die Sensoren **220** können ein elektrisches Signal zur Erfassung einer Messgröße erzeugen oder die Messgröße selbst analog oder digital aufnehmen. Die Aktoren **210** sind an die Ausgänge der SPS **100** angeschlossen und setzen die elektrischen Signale der Ausgangsdaten der SPS **100** in mechanische Bewegung oder andere physikalische Größen (zum Beispiel Temperatur, Druck, etc.) um.

[0037] Die dynamische Ansteuerung der Aktoren **210** kann zunächst auf dem Auslesen der aktuellen Ist-Daten der Aktoren **210** beruhen. Auf Basis der Ist-Daten der Aktoren **210** und der Messdaten der Sensoren **220** können aus den Eingabedaten die Ausgabedaten für die Aktoren **210** generiert werden. Alternativ dazu kann die Ansteuerung der Aktoren **210** aufgrund von Vorgaben, zum Beispiel anhand von Bewegungsprofilen, erfolgen.

[0038] Die Datenverarbeitungseinrichtung **110** umfasst mehrere Prozessorkerne, wobei in **Fig. 1** beispielsweise ein erster Hauptprozessorkern **111** und ein zweiter Hauptprozessorkern **112** sowie ein erster Parallelprozessorkern **113** und ein zweiter Parallelprozessorkern **114** gezeigt sind. Ebenso denkbar ist eine Ausgestaltung, in der die Datenverarbeitungseinrichtung **110** eine dazu abweichende Anzahl an Hauptprozessorkernen und/oder Parallelprozessorkernen aufweist. Der Vorteil an der Ausgestaltung der Datenverarbeitungseinrichtung **110** mit einem ersten Hauptprozessorkern **111** und einem zweiten Hauptprozessorkern **112** sowie einem ersten Parallelprozessorkern **113** und einem zweiten Parallelprozessorkern **114** liegt darin, dass die abzuarbeitenden Teilaufgaben der jeweiligen Programme der Steuerungsaufgabe der SPS **100** gemäß der folgenden Beschreibung auf die beiden Hauptprozessorkerne **111**, **112** und die beiden Parallelprozessorkerne **113**, **114** verteilt werden können. Dadurch ist es möglich die Rechenzeit zu reduzieren und zusätzlich eine optimale Auslastung der Ressourcen zu gewährleisten.

[0039] Dem ersten Hauptprozessorkern **111** und dem zweiten Hauptprozessorkern **112** ist jeweils das ausführbare Programm der Steuerungsaufgabe zugeordnet, wobei der erste Parallelprozessorkern **113** und der zweite Parallelprozessorkern **114** wie oben genannt unterstützend bei der Abarbeitung der Teilaufgaben, in den entsprechenden Programmen des ersten und zweiten Hauptprozessorkerns **111**, **112**, mitwirken können. Die Steuerungsaufgabe kann beispielsweise aus einem innerhalb der SPS **100** ausgeführten Programmzyklus bestehen, wobei der Programmzyklus den Empfang der Eingabedaten (die Messdaten der Sensoren **220** und die aktuellen Ist-Daten der Aktoren **210**), die Verarbeitung der Eingabedaten zu Ausgabedaten für die Aktoren **210** und die Ausgabe der Ausgabedaten für die Aktoren **210**

umfasst. Nach der erfolgten Verarbeitung, also dem Ende des Programmzyklus, beginnt der Programmzyklus der Steuerungsaufgabe von neuem.

[0040] Im Unterschied zum erfindungsgemäßen Programmzyklus der Steuerungsaufgabe, umfasst ein Feldbuszyklus neben den oben genannten Schritten des Programmzyklus, des Weiteren die Übermittlung der Eingabedaten der Sensoren **220** bzw. Aktoren **210** über den Feldbus an die SPS **100**. Die Ausgabedaten, die im Programmzyklus innerhalb der SPS **100** generiert wurden, werden in dem Feldbuszyklus ferner über den Feldbus an die Aktoren **210** übertragen, damit die Aktoren **210** entsprechend der empfangenen Ausgabedaten handeln können.

[0041] Werden die Teilaufgaben von einem Hauptprozessorkern und mehreren Parallelprozessorkernen parallel abgearbeitet, so können die Teilaufgaben einem parallelen Bearbeitungsabschnitt in dem Programm des ersten Hauptprozessorkerns **111** bzw. in dem Programm des zweiten Hauptprozessorkerns **112** zugeordnet werden. Den parallelen Bearbeitungsabschnitten in den entsprechenden Programmen des ersten Hauptprozessorkerns **111** und des zweiten Hauptprozessorkerns **112** kann eine Priorität mit einer vorgegebenen ersten Prioritätsstufe und einer vorgegebenen zweiten Prioritätsstufe zugewiesen sein, wobei die erste Prioritätsstufe und die zweite Prioritätsstufe von einem Prioritätenverwalter **350** einer Steuerungseinrichtung **120** organisiert werden. Der Prioritätenverwalter **350** dient in erster Linie zum Verwalten der ersten und zweiten Prioritätsstufe in der Datenstruktur. Ferner kann der Prioritätenverwalter **350** dazu ausgelegt sein, den ersten und zweiten Parallelprozessorkern **113**, **114** über eine Änderung in der Reihenfolge der Einträge der Prioritätsstufen in der Datenstruktur zu informieren.

[0042] Der Prioritätenverwalter **350** kann auf dem ersten Hauptprozessorkern **111** und/oder dem zweiten Hauptprozessorkern **112** ausgeführt werden. Ebenfalls denkbar ist es, den Prioritätenverwalter **350** als eigenständiges Modul zu verwirklichen, auf den der erste Parallelprozessorkern **113** und der zweite Parallelprozessorkern **114** zugreifen können, um den ersten Hauptprozessorkern **111** und den zweiten Hauptprozessorkern **112** bei der Abarbeitung der Teilaufgaben zu unterstützen.

[0043] In den parallelen Bearbeitungsabschnitten in den Programmen des ersten Hauptprozessorkerns **111** und des zweiten Hauptprozessorkerns **112** werden zudem ein erstes Arbeitspaket und ein zweites Arbeitspaket erstellt. Das erste Arbeitspaket umfasst eine erste Menge an Teilaufgaben und das zweite Arbeitspaket umfasst eine zweite Menge an Teilaufgaben. Die erste Menge an Teilaufgaben und die zweite Menge an Teilaufgaben können von dem ersten Hauptprozessorkern **111** und dem zweiten Hauptpro-

zessorkern **112** bzw. dem ersten Parallelprozessorkern **113** und dem zweiten Parallelprozessorkern **114** entsprechend der ersten und zweiten Prioritätsstufe des zugehörigen parallelen Bearbeitungsabschnitts abgearbeitet werden.

[0044] Weiterhin denkbar ist, dass mehrere Programme der Steuerungsaufgabe auf einem einzigen Hauptprozessorkern **111**, **112** ausgeführt werden können. Die Teilaufgaben aus den entsprechenden Arbeitspaketen der jeweiligen Programme werden einer sequentiellen Prioritätsstufe entsprechend auf dem einen Hauptprozessorkern **111**, **112** abgearbeitet. Die sequentielle Prioritätsstufe der Programme kann sich von der Prioritätsstufe der den jeweiligen Programmen zugeordneten parallelen Bearbeitungsabschnitte unterscheiden. Beispielsweise ist es dadurch möglich, dass Teilaufgaben eines Arbeitspakets eines entsprechenden Programms auf den Parallelprozessorkernen **113**, **114** ausgeführt werden und gleichzeitig können Teilaufgaben aus diesem Arbeitspaket von dem einen Hauptprozessorkern **111**, **112** sequentiell abgearbeitet werden. Falls die sequentielle Abarbeitung der Teilaufgaben des entsprechenden Arbeitspakets durch ein höher priorisiertes Programm auf dem einen Hauptprozessorkern **111**, **112** unterbrochen wird, können die Teilaufgaben des entsprechenden Arbeitspakets dennoch weiter parallel von den Parallelprozessorkernen **113**, **114** abgearbeitet werden, da den parallelen Bearbeitungsabschnitten in den entsprechenden Programmen eine zur sequentiellen Prioritätsstufe separate Prioritätsstufe zugewiesen sein kann. Folglich kann eine Verdrängung der aktuellen Teilaufgaben durch höher priorisierte andere Teilaufgaben nicht nur im Fall der parallelen Abarbeitung der Teilaufgaben auf den Parallelprozessorkernen **113**, **114**, sondern auch im Fall einer sequentiellen Abarbeitung der Teilaufgaben auf dem einen Hauptprozessorkern **111**, **112** erfolgen.

[0045] Fig. 2 zeigt eine schematische Struktur eines Verfahrens **300** zur Datenverarbeitung auf der SPS mit der in Fig. 1 gezeigten Ausführungsform. Die von dem ersten Hauptprozessorkern **111** und dem zweiten Hauptprozessorkern **112** auszuführenden Programme der Steuerungsaufgabe weisen jeweils den parallelen Bearbeitungsabschnitt auf. Der Pfeil mit dem Bezugszeichen **500** stellt das Erstellen des ersten Arbeitspakets **305** zu Beginn des parallelen Bearbeitungsabschnitts des ersten Hauptprozessorkerns **111** dar. Das erste Arbeitspaket **305** umfasst die erste Menge an Teilaufgaben **320**, die im dargestellten Beispiel aus einer ersten Teilaufgabe **321**, einer zweiten Teilaufgabe **322**, einer dritten Teilaufgabe **323** und einer vierten Teilaufgabe **324** besteht. Das erste Arbeitspaket **305** umfasst ferner die Referenz auf die erste Prioritätsstufe **340** des parallelen Bearbeitungsabschnitts des ersten Hauptprozessorkerns **111**. Das Übergeben der Referenz auf die erste Prioritätsstufe **340** des parallelen Bearbeitungsabschnitts des ers-

ten Hauptprozessorkerns **111** an den Prioritätenverwalter **350** wird mithilfe des Pfeils mit dem Bezugszeichen **505** verdeutlicht. Der Prioritätenverwalter **350** fügt die erste Prioritätsstufe **340** in eine Datenstruktur **355** ein und speichert die Datenstruktur **355** im Speicher der Steuerungseinrichtung, der im Folgenden Datenstrukturspeicher genannt wird und in den Figuren nicht gezeigt ist. Besitzt die Datenstruktur **355** keine Einträge, so fügt der Prioritätenverwalter **350** die Referenz auf die erste Prioritätsstufe **340** an die erste Stelle der Einträge der Prioritätsstufen in der Datenstruktur **355** ein.

[0046] Die Pfeile mit den Bezugszeichen **510**, **515** zeigen das Anfragen des ersten und des zweiten Parallelprozessorkerns **113**, **114** beim Prioritätenverwalter **350** an, welcher Eintrag der Prioritätsstufen an der ersten Stelle der Einträge in der Datenstruktur **355** steht. Beispielsweise kann der erste Eintrag der Prioritätsstufe in der Datenstruktur **355** die Referenz auf die erste Prioritätsstufe **340** des ersten Hauptprozessorkerns **111** sein. Da sich die erste Prioritätsstufe **340** auf den parallelen Bearbeitungsabschnitt des ersten Hauptprozessorkerns **111** bezieht, greifen der erste Parallelprozessorkern **113** und der zweite Parallelprozessorkern **114** auf das dazu korrespondierende erste Arbeitspaket **305** zu. Der Zugriff des ersten und zweiten Parallelprozessorkerns **113**, **114** auf das erste Arbeitspaket **305** wird mit den Pfeilen mit den Bezugszeichen **520**, **525** angezeigt. Der erste Parallelprozessorkern **113** kann beispielsweise beim ersten Arbeitspaket **305** anfragen, ob es eine freie erste bis vierte Teilaufgabe **321-324** gibt, die noch nicht anderweitig verteilt wurde. Die Anfrage des ersten Parallelprozessorkerns wird mithilfe des Pfeils mit dem Bezugszeichen **520** angezeigt. Beispielsweise kann dies die erste Teilaufgabe **321** sein, die der erste Parallelprozessorkern **113** zur Abarbeitung zugeteilt bekommt. Die erste Teilaufgabe **321** steht dann für die Verteilung vorerst nicht mehr zur Verfügung, um eine Mehrfachvergabe derselben Teilaufgabe vermeiden zu können. Dem zweiten Parallelprozessorkern **114** kann auf diese Weise zum Beispiel die zweite Teilaufgabe **322** zugewiesen werden. Der erste Hauptprozessorkern **111** kann während des parallelen Bearbeitungsabschnitts im Programm auch sequentiell Teilaufgaben abarbeiten, demnach fragt er ebenfalls beim ersten Arbeitspaket **305** nach einer freien ersten bis vierten Teilaufgabe **321-324** zur Abarbeitung an. Die Anfrage des ersten Hauptprozessorkerns **111** wird über den Pfeil mit dem Bezugszeichen **530** dargestellt. Beispielsweise kann ihm die dritte Teilaufgabe **323** zugeteilt werden.

[0047] Sobald der erste Hauptprozessorkern **111** oder einer der beiden Parallelprozessorkerne **113**, **114** ihre erste bis dritte Teilaufgabe **321-323** vollständig abgearbeitet haben, kann die verbleibende vierte Teilaufgabe **324** wie oben beschrieben zugeteilt werden. Es wurde ein Ausführungsbeispiel für die Vertei-

lung der ersten bis vierten Teilaufgabe **321-324** auf den ersten Hauptprozessorkern **111** und den ersten und zweiten Parallelprozessorkern **113, 114** gegeben. Ebenfalls denkbar ist es die erste bis vierte Teilaufgabe **321-324** anderweitig auf den ersten Hauptprozessorkern **111** und den ersten und zweiten Parallelprozessorkern **113, 114** aufzuteilen.

[0048] Gelangt der zweite Hauptprozessorkern **112** an seinen parallelen Bearbeitungsabschnitt in seinem Programm, so erstellt er das zweite Arbeitspaket **310** mit der zweiten Menge an Teilaufgaben **330**, die im dargestellten Beispiel aus einer fünften Teilaufgabe **331**, einer sechsten Teilaufgabe **332** und einer siebten Teilaufgabe **333** besteht. Das Erstellen des zweiten Arbeitspakets **310** vom zweiten Hauptprozessorkern **112** ist mithilfe des Pfeils mit dem Bezugszeichen **535** visualisiert. Ferner umfasst das zweite Arbeitspaket **310** die Referenz auf die zweite Prioritätsstufe **345** des parallelen Bearbeitungsabschnitts im Programm des zweiten Hauptprozessorkerns **112**. Die Referenz auf die zweite Prioritätsstufe **345** übergibt er zweite Hauptprozessorkern **112** ferner dem Prioritätenverwalter **350**. Das Übergeben der Referenz auf die zweite Prioritätsstufe wird mit dem Pfeil mit dem Bezugszeichen **540** dargestellt. Der Prioritätenverwalter **350** ordnet die Referenz auf die zweite Prioritätsstufe **345** ihrem Wert entsprechend an die zweite Stelle der Einträge der Prioritätsstufen in die Datenstruktur **355** oder an die erste Stelle der Einträge der Prioritätsstufen in die Datenstruktur **355** ein. Befand sich vorher bereits ein anderer Eintrag an der ersten Stelle der Einträge der Prioritätsstufen in der Datenstruktur **355**, so wird dieser, für den Fall, dass die erste Prioritätsstufe **340** niedriger ist als die zweite Prioritätsstufe **345** des neuen Eintrags, vom Prioritätenverwalter **350** an die zweite Stelle der Einträge der Prioritätsstufen in der Datenstruktur **355** gesetzt und die Einträge der Datenstruktur **355** im Datenstruktur-speicher gesichert.

[0049] Ändert sich die Reihenfolge der Einträge der Prioritätsstufen in der Datenstruktur **355**, so informiert der Prioritätenverwalter **350** den ersten und zweiten Parallelprozessorkern **113, 114** darüber. Der Vorgang geht mit dem Unterbrechen der aktuell von dem ersten Parallelprozessorkern **113** und/oder dem zweiten Parallelprozessorkern **114** bearbeiteten ersten, zweiten und gegebenenfalls vierten Teilaufgabe **321, 322, 324** aus der ersten Menge an Teilaufgaben **320** einher. Die Zwischenergebnisse der Berechnungen können an die in **Fig. 1** dargestellte Steuerungseinrichtung **120** übergeben werden und auf einer nicht dargestellten Speichereinheit, die im Folgenden als Zwischenergebnisspeichereinheit bezeichnet wird, gesichert werden. Auf die Zwischenergebnisspeichereinheit können sowohl der erste und zweite Hauptprozessorkern **111, 112** als auch der erste und zweite Parallelprozessorkern **113, 114** zugreifen, falls die Bearbeitung der unterbrochenen ersten,

zweiten und gegebenenfalls vierten Teilaufgabe **321, 322, 324** auf den Parallelprozessorkernen **113, 114** fortgeführt werden soll.

[0050] Während der erste Hauptprozessorkern **111** die dritte Teilaufgabe **323** aus der ersten Menge an Teilaufgaben **320** des ersten Arbeitspakets **305** weiter sequentiell abarbeitet, so fragen der erste Parallelprozessorkern **113** und der zweite Parallelprozessorkern **114** sowie der zweite Hauptprozessorkern **112** beim zweiten Arbeitspaket **310** nach der fünften bis siebten Teilaufgaben **331-333** aus der zweiten Menge an Teilaufgaben **330** an. Die Anfragen der jeweiligen Prozessorkerne sind mithilfe der drei Pfeile mit den Bezugszeichen **545, 550, 555** dargestellt. Beispielsweise kann die fünfte Teilaufgabe **331** dem ersten Parallelprozessorkern **113** und die siebte Teilaufgabe **333** dem zweiten Parallelprozessorkern **114** zugeteilt werden. Die sechste Teilaufgabe **332** kann vom zweiten Hauptprozessorkern **112** währenddessen sequentiell abgearbeitet werden. Für die Verteilung der fünften bis siebten Teilaufgabe **331-333** auf den ersten und zweiten Parallelprozessorkern **113, 114** und den zweiten Hauptprozessorkern **112** ist ebenso eine vom dargestellten Beispiel abweichende Aufteilung der fünften bis siebten Teilaufgabe **331-333** denkbar.

[0051] Ist die fünfte, sechste und siebte Teilaufgabe **331, 332, 333** von den oben beschriebenen Prozessorkernen aus der zweiten Menge an Teilaufgaben **330** gänzlich erledigt, so meldet das entsprechende zweite Arbeitspaket **310** dies an den Prioritätenverwalter **350**, damit der Prioritätenverwalter **350** die entsprechende Referenz auf die zweite Prioritätsstufe **345** aus den Einträgen der Prioritätsstufen in der Datenstruktur **355** entfernt. Nachdem die fünfte und/oder sechste und/oder siebte Teilaufgabe **331, 332, 333** vollständig von den beiden Parallelprozessorkernen **113, 114** und dem zweiten Hauptprozessorkern **112** abgearbeitet wurden, können die unterbrochene erste und/oder zweite Teilaufgabe **321, 322** auf den beiden Parallelprozessorkernen **113, 114** weiter abgearbeitet werden. Die Verteilung der ersten und zweiten Teilaufgabe **321, 322** auf den ersten und zweiten Parallelprozessorkern **113, 114** kann dabei beispielsweise analog zum beschriebenen Ausführungsbeispiel erfolgen, d.h. der erste Parallelprozessorkern **113** bekommt die erste Teilaufgabe **321** und der zweite Parallelprozessorkern **114** die Teilaufgabe **322** vom ersten Arbeitspaket **305** zugeteilt. Die beiden Parallelprozessorkerne **113, 114** greifen auf die Zwischenergebnisspeichereinheit zu, um die Zwischenergebnisse der ersten und zweiten Teilaufgabe **321, 322** für die weitere Abarbeitung der genannten Teilaufgaben nutzen zu können.

[0052] Da der erste Hauptprozessorkern **111** die dritte Teilaufgabe **323** des ersten Arbeitspakets **305** weiter abgearbeitet hat, während der erste Parallelpro-

zessorkern **113** die fünfte Teilaufgabe **331** und der zweite Parallelprozessorkern **114** die siebte Teilaufgabe **333** aus dem zweiten Arbeitspaket **310** abgearbeitet hat, ist es möglich, dass der erste Hauptprozessorkern **111** in der Zeit, in der die Parallelprozessorkerne **113**, **114** ihre zugeteilte fünfte und siebte Teilaufgabe **331**, **333** gänzlich abarbeiten, auch der erste Hauptprozessorkern **111** die Abarbeitung der dritten Teilaufgabe **323** vollständig abschließt. Danach kann der erste Hauptprozessorkern **111** beim ersten Arbeitspaket **305** nach der verbleibenden vierten Teilaufgabe **324** anfragen und diese abarbeiten, während die Parallelprozessorkerne **113**, **114** die Abarbeitung der ersten und zweiten Teilaufgabe **321**, **322** wieder aufnehmen. Die Anfrage des ersten Hauptprozessorkerns **111** beim ersten Arbeitspaket **305** wird über den Pfeil mit dem Bezugszeichen **520** angezeigt.

[0053] Ist die erste, zweite und vierte Teilaufgabe **321**, **322**, **324** vollständig von dem ersten und zweiten Parallelprozessorkern **113**, **114** sowie dem ersten Hauptprozessorkern **111** abgearbeitet worden (sowie die dritte Teilaufgabe **323** vom ersten Hauptprozessorkern **111** fertiggestellt worden), so meldet das zugehörige erste Arbeitspaket **305** die Erledigung der ersten, zweiten und vierten Teilaufgabe **321**, **322**, **324** (sowie dritte Teilaufgabe **323**) an den Prioritätenverwalter **350**, damit dieser die Referenz auf die erste Prioritätsstufe **340** aus den Einträgen der Prioritätsstufen in der Datenstruktur **355** entfernt. Denkbar ist weiterhin, dass der Prioritätenverwalter **350** selbst bei dem ersten und zweiten Arbeitspaket **305**, **310** anfragt, ob noch erste bis siebte Teilaufgaben **321-324**, **331-333** aus der ersten und zweiten Menge an Teilaufgaben **320**, **330** zur Erledigung bereitstehen, und dann gegebenenfalls die entsprechende Referenz auf die erste und zweite Prioritätsstufe **340**, **345** aus der Datenstruktur **355** im Datenstrukturspeicher entfernt.

[0054] Durch die Verteilung einzelner erster bis siebter Teilaufgaben **321-324**, **331-333** auf den ersten und zweiten Hauptprozessorkern **111**, **112** und den ersten und zweiten Parallelprozessorkern **113**, **114** kann die Gesamtbearbeitungszeit der ersten bis siebten Teilaufgaben **321-324**, **331-333** reduziert und die Rechenkapazität optimal ausgenutzt werden. Ferner können Deadlines der Programme eingehalten und die Echtzeitfähigkeit der Steuerungsaufgabe gewährleistet werden, indem die erste und zweite Prioritätsstufe **340**, **345** der jeweiligen parallelen Bearbeitungsabschnitte der Programme des ersten und zweiten Hauptprozessorkerns **111**, **112** berücksichtigt werden. Denn mithilfe der ersten und zweiten Prioritätsstufe **340**, **345** kann die Abarbeitung der ersten bis vierten Teilaufgabe **321-324**, aus der ersten Menge an Teilaufgaben **320** zugunsten anderer fünfter bis siebter Teilaufgaben **331-333** aus der zweiten Menge an Teilaufgaben **330** unterbrochen werden, falls

diesen eine höhere zweite Prioritätsstufe **345** zugeordnet ist. Ebenso denkbar ist, dass die Abarbeitung der fünften bis siebten Teilaufgabe **331-333** aus der zweiten Menge an Teilaufgaben mit der zweiten Prioritätsstufe **345** aufgrund der höher priorisierten ersten Menge an Teilaufgaben **320** mit der ersten bis vierten Teilaufgabe **321-324** und der ersten Prioritätsstufe unterbrochen wird. Unterbrochene erste bis siebte Teilaufgaben **321-324**, **331-333** aus der ersten und zweiten Menge an Teilaufgaben **320**, **330** können weiter abgearbeitet werden. Denn die Referenz auf den ersten und zweiten Prioritätsstufeneintrag **340**, **345** bleibt in der Datenstruktur **355** im Datenstrukturspeicher erhalten, so lange die entsprechenden ersten bis siebten Teilaufgaben **321-324**, **331-333** aus der ersten und zweiten Menge an Teilaufgaben **320**, **330** des ersten und zweiten Arbeitspakets **305**, **310** nicht vollständig abgearbeitet worden sind.

[0055] In einer Ausgestaltung kann der Prioritätenverwalter **350** den ersten und zweiten Parallelprozessorkern **113**, **114** aktiv über die Änderung des ersten Eintrags der ersten oder zweiten Prioritätsstufe **340**, **345** in der Datenstruktur **355** informieren.

[0056] Damit können der erste und der zweite Parallelprozessorkern **113**, **114** jeweils direkt beim entsprechenden ersten oder zweiten Arbeitspaket **305**, **310** nach einer freien ersten bis siebten Teilaufgabe **321-324**, **331-333** anfragen und frühzeitig mit deren Abarbeitung beginnen. Die Anfragen der jeweiligen Prozessorkerne sind mithilfe der Bezugszeichen **520**, **525**, **545**, **550** dargestellt. Auch kann der Prioritätenverwalter **350** dahingehend ausgestaltet sein, dass der erste und zweite Parallelprozessorkern **113**, **114** selbst aktiv beim Prioritätenverwalter **350** anfragen, ob es eine Änderung in der Reihenfolge der Einträge der ersten und zweiten Prioritätsstufe **340**, **345** gibt. Das Anfragen des ersten und zweiten Parallelprozessorkerns **113**, **114** ist mithilfe der beiden Pfeile mit den Bezugszeichen **510**, **515** angedeutet.

[0057] Denkbar ist weiterhin, dass die Anfragen des ersten und zweiten Hauptprozessorkerns **111**, **112** und des ersten und zweiten Parallelprozessorkerns **113**, **114** an das erste oder zweite Arbeitspaket **305**, **310** von einer zentralen Einrichtung gesteuert werden. Die Anfragen der jeweiligen Prozessorkerne sind mithilfe der Pfeile mit den Bezugszeichen **520**, **525**, **530**, **545**, **550**, **555** dargestellt. Die zentrale Einrichtung kann des Weiteren die Aufgabenverteilung der ersten bis siebten Teilaufgaben **321-324**, **331-333** aus der ersten und zweiten Menge an Teilaufgaben **320**, **330** des ersten und zweiten Arbeitspakets **305**, **310** steuern. Ferner kann die zentrale Einrichtung dafür Sorge tragen, dass die Meldung der Erledigung aller ersten bis siebten Teilaufgaben **321-324**, **331-333** aus der ersten und zweiten Menge an Teilaufgaben **320**, **330** des ersten und zweiten Arbeits-

pakets **305**, **310** an den Prioritätenverwalter **350** geleitet wird.

[0058] Haben die erste und zweite Prioritätsstufe **340**, **345** der entsprechenden parallelen Bearbeitungsabschnitte in den Programmen der ersten und zweiten Hauptprozessorkerne **111**, **112** den gleichen Wert, so können sich der erste und zweite Hauptprozessorkern **111**, **112** den ersten und zweiten Parallelprozessorkern **113**, **114** für die parallelisierte Ausführung **360** der ersten bis siebten Teilaufgaben **321-324**, **331-333** teilen.

[0059] Fig. 3 zeigt ein zeitliches Ablaufschema eines Verfahrens **300** gemäß Fig. 2. Fig. 3 veranschaulicht die Echtzeitdatenverarbeitung der in Fig. 1 dargestellten SPS **100** auf dem ersten und zweiten Hauptprozessorkern **111**, **112** und dem ersten und zweiten Parallelprozessorkern **113**, **114**. Die Echtzeitdatenverarbeitung stellt den Kern der Erfindung dar und wird über die vier Zeitachsen visualisiert, die die Ausführung der jeweiligen Programme der Steuerungsaufgabe auf dem ersten und zweiten Hauptprozessorkern **111**, **112** und dem ersten und zweiten Parallelprozessorkern **113**, **114** veranschaulichen.

[0060] Der Grundgedanke der Erfindung liegt darin, dass die Ausführungszeiten der Programme des ersten und zweiten Hauptprozessorkerns **111**, **112** so festgelegt sind, dass der erste und zweite Hauptprozessorkern **111**, **112** in der Lage sind, die ersten bis siebten Teilaufgaben **321-324**, **331-333** aus dem ersten und zweiten Arbeitspaket **305**, **310** ohne die Unterstützung des ersten und/oder zweiten Parallelprozessorkerns **113**, **114** sequentiell abzuarbeiten. Damit die vorhandenen Ressourcen jedoch optimal ausgenutzt und die Gesamtausführungszeit der ersten bis siebten Teilaufgaben **321-324**, **331-333** in den jeweiligen Programmen der Steuerungsaufgabe reduziert werden, können die ersten bis siebten Teilaufgaben **321-324**, **331-333** ebenfalls auf den ersten und/oder zweiten Parallelprozessorkern **113**, **114** aufgeteilt werden, wobei der erste und zweite Hauptprozessorkern **111**, **112** weiterhin sequentiell an der Abarbeitung der ersten bis siebten Teilaufgaben **321-324**, **331-333** beteiligt sind. Zudem ist es mithilfe der Parallelisierung für die am höchsten priorisierten Programme der jeweiligen Hauptprozessorkerne **111**, **112** möglich, die kürzere Ausführungszeit der entsprechenden Programme durch die Parallelisierung einzuplanen, und auf diese Weise eine Deadline, die kleiner als eine rein sequentielle Ausführungszeit der Programme ist, einzuhalten.

[0061] Die beiden vertikalen Balken zu Beginn der oberen beiden Zeitachsen in Fig. 3 markieren den Start **301** der jeweiligen Ausführung der Programme auf dem ersten Hauptprozessorkern **111** und dem zweiten Hauptprozessorkern **112**. Diese Balken sind zur besseren Vergleichbarkeit in die unteren beiden

Zeitachsen für den ersten Parallelprozessorkern **113** und den zweiten Parallelprozessorkern **114** ebenfalls mit eingezeichnet, um den Start **301** der Ausführung der Programme anzuzeigen. Der erste Hauptprozessorkern **111** und der zweite Hauptprozessorkern **112** arbeiten nach dem entsprechenden Start **301** der Ausführung der Programme beispielsweise jeweils Aufgaben sequentiell **365** ab, während der erste Parallelprozessorkern **113** und der zweite Parallelprozessorkern **114** zunächst keine Berechnungen durchführen (gestrichelte horizontale Linie in den unteren beiden Zeitachsen).

[0062] Der zweite vertikale Balken mit dem Bezugszeichen **304** in der obersten Zeitachse in Fig. 3 zeigt an, dass das Programm des ersten Hauptprozessorkerns **111** nun in den parallelen Bearbeitungsabschnitt **303** gelangt. Daraufhin erstellt der erste Hauptprozessorkern **111** das erste Arbeitspaket **305**, das die erste Menge an Teilaufgaben **320** enthält und die Referenz auf die erste Prioritätsstufe **340** des parallelen Bearbeitungsabschnitts **303** des ersten Hauptprozessorkerns **111** beinhaltet. Das Erstellen des ersten Arbeitspakets **305** vom ersten Hauptprozessorkern **111** wird mithilfe des Pfeils mit dem Bezugszeichen **500** dargestellt. Der erste Hauptprozessorkern **111** kann die Referenz auf die erste Prioritätsstufe **340** des parallelen Bearbeitungsabschnitts **303** des ersten Hauptprozessorkerns **111** über den in Fig. 2 dargestellten Prioritätenverwalter **350** an den ersten und zweiten Parallelprozessorkern **113**, **114** übergeben. Die Übergabe der Referenz der ersten Prioritätsstufe **340** an den ersten und zweiten Parallelprozessorkern **113**, **114** wird anhand der Pfeile mit den Bezugszeichen **560** visualisiert. Der Prioritätenverwalter **350** trägt die Referenz der ersten Prioritätsstufe **340** entsprechend ihrer Prioritätsstufe als Eintrag in die Datenstruktur im Datenstrukturspeicher ein. Direkt unterhalb der beiden oberen Zeitachsen für die jeweiligen Ausführungen der Programme auf dem ersten Hauptprozessorkern **111** und dem zweiten Hauptprozessorkern **112** in Fig. 3 sind die entsprechenden Einträge der ersten Prioritätsstufe **340** der Datenstruktur im Zeitverlauf dargestellt.

[0063] Der erste Parallelprozessorkern **113** und der zweite Parallelprozessorkern **114** fragen beim ersten Arbeitspaket **305** jeweils nach einer freien ersten bis vierten Teilaufgabe **321-324** an, wie bereits oben im Zusammenhang mit Fig. 2 beschrieben. Die Anfrage des ersten und zweiten Parallelprozessorkerns **113**, **114** wird mithilfe der beiden Pfeile mit den Bezugszeichen **520**, **525** angezeigt. Beispielsweise kann der erste Parallelprozessorkern **113** die erste Teilaufgabe **321** für die Abarbeitung zugeteilt bekommen. Die erste Teilaufgabe **321** steht dann für die Verteilung vorerst nicht mehr zur Verfügung, um eine Mehrfachvergabe derselben Teilaufgabe vermeiden zu können. Dem zweiten Parallelprozessorkern **114** kann auf diese Weise zum Beispiel die zweite Teilaufgabe

be **322** zugewiesen werden. Der erste Hauptprozessorkern **111** arbeitet währenddessen sequentiell **365** und kann beispielsweise die dritte Teilaufgabe **323** zur Abarbeitung zugeteilt erhalten. Der zweite Hauptprozessorkern **112** kann in der Zeit weiterhin sequentiell **365** Aufgaben abarbeiten, die keine Parallelisierung erfordern und folglich nicht in dem ersten oder dem zweiten Arbeitspaket **305**, **310** enthalten sind.

[0064] Der dritte vertikale Balken mit dem Bezugszeichen **306** der zweiten Zeitachse in **Fig. 3** indiziert den Beginn des parallelen Bearbeitungsabschnitts **303** des zweiten Hauptprozessorkerns **112**. Mit einem durchgezogenen Pfeil **535** ist das vom zweiten Hauptprozessorkern **112** erstellte zweite Arbeitspaket **310** angedeutet. Dieses weist die zweite Menge an Teilaufgaben **330** auf und beinhaltet die Referenz auf die zweite Prioritätsstufe **345** des parallelen Bearbeitungsabschnitts **303** des zweiten Hauptprozessorkerns **112**. Wie oben beschrieben, kann die Referenz auf die zweite Prioritätsstufe **345** über den in **Fig. 2** dargestellten Prioritätenverwalter **350** an den ersten Parallelprozessorkern **113** und den zweiten Parallelprozessorkern **114** übergeben und in der Datenstruktur **355** im Datenstrukturspeicher gesichert werden. Das Übergeben der Referenz auf die zweite Prioritätsstufe **345** an den ersten und zweiten Parallelprozessorkern **113**, **114** wird mit den beiden Pfeilen mit den Bezugszeichen **565** dargestellt. Beispielsweise kann die zweite Prioritätsstufe **345** des parallelen Bearbeitungsabschnitts **303** des zweiten Hauptprozessorkerns **112** höher sein, als die erste Prioritätsstufe **340** des parallelen Bearbeitungsabschnitts **303** im Programm des ersten Hauptprozessorkerns **111**. Der Prioritätenverwalter kann daraufhin die Reihenfolge der Einträge in der Datenstruktur ändern, so dass die zweite Prioritätsstufe **345** an die erste Stelle der Einträge der Prioritätsstufen gesetzt wird und kann weiterhin den ersten Parallelprozessorkern **113** und den zweiten Parallelprozessorkern **114** über die Änderung der Reihenfolge der Einträge der Prioritätsstufen informieren. Die Einträge der zweiten Prioritätsstufe **345** und der ersten Prioritätsstufe **340** der Datenstruktur sind unterhalb der oberen beiden Zeitachsen in **Fig. 3** gezeigt.

[0065] Die zuvor von dem ersten Parallelprozessorkern **113** und dem zweiten Parallelprozessorkern **114** bearbeiteten ersten und zweiten Teilaufgaben **321**, **322** werden aus diesem Grund unterbrochen. Die Zwischenergebnisse der Berechnungen können an die in **Fig. 1** dargestellte Steuerungseinrichtung **120** übergeben werden und auf der Zwischenergebnisspeichereinheit gesichert werden, falls die unterbrochenen ersten und zweiten Teilaufgaben **321**, **322** mit den bisherigen Zwischenergebnissen weiter bearbeitet werden sollen. Ferner besteht die Möglichkeit, die Zwischenergebnisse nicht zu sichern. Dies erfordert jedoch eine erneute Berechnung der ersten und zweiten Teilaufgabe **321**, **322** nach deren

Abbruch. Weiterhin ist denkbar, die erste und zweite Teilaufgaben **321**, **322** bis zu einem vorgegebenen Unterbrechungspunkt weiter zu bearbeiten und die Abarbeitung dann zu unterbrechen. Beispielsweise kann der Unterbrechungspunkt in einer Kontrollstruktur in der ersten und zweiten Teilaufgabe **321**, **322** des ersten Arbeitspakets **305** gesetzt sein. Der Unterbrechungspunkt kann zum Beispiel zu Beginn oder am Ende eines Durchlaufs der Kontrollstruktur gesetzt sein. Die Festlegung hat den Vorteil, dass nur die aktuelle Iteration der Kontrollstruktur, bei der es sich beispielsweise um eine Schleife handeln kann, und gegebenenfalls vorhandene Zwischenergebnisse gesichert werden müssen.

[0066] Der erste Parallelprozessorkern **113** und der zweite Parallelprozessorkern **114** fragen nun wie oben beschrieben beim zweiten Arbeitspaket **310** jeweils nach einer der unbearbeiteten fünften bis siebten Teilaufgaben **331-333** an. Die Anfrage des ersten und zweiten Parallelprozessorkerns **113**, **114** ist mithilfe der beiden Pfeile mit den Bezugszeichen **545**, **550** dargestellt. Beispielsweise wird dem ersten Parallelprozessorkern **113** die fünfte Teilaufgabe **331** und dem zweiten Parallelprozessorkern **114** die siebte Teilaufgabe **333** zugeteilt. Währenddessen kann zum Beispiel der erste Hauptprozessorkern **111** weiterhin die dritte Teilaufgabe **323** abarbeiten und der zweite Hauptprozessorkern **112** kann auf diese Weise die sechste Teilaufgabe **332** zur sequentiellen Abarbeitung **365** erhalten. Sobald der erste Hauptprozessorkern **111** die dritte Teilaufgabe **323** vollständig abgearbeitet hat, kann er beispielsweise mit der Abarbeitung der vierten Teilaufgabe **324** beginnen, unabhängig davon, dass die zugehörige erste Prioritätsstufe **340** nicht an erster Stelle der Einträge der Prioritätsstufen in der Datenstruktur steht, da diese Reihenfolge der Einträge der Prioritätsstufen nur für die beiden Parallelprozessorkerne **113**, **114** relevant ist. Alternativ dazu kann der erste Hauptprozessorkern **111** nach der Fertigstellung der Bearbeitung der dritten Teilaufgabe **323** gegebenenfalls auch eine der zuvor unterbrochenen ersten oder zweiten Teilaufgabe **321**, **322** abarbeiten.

Ist die fünfte bis siebte Teilaufgabe **331-333** des zweiten Arbeitspakets **310** am Ende des parallelen Bearbeitungsabschnitts **303** im Programm des zweiten Hauptprozessorkerns **112** vollständig von den oben genannten Prozessorkernen **112**, **113**, **114** abgearbeitet, so entfernt der Prioritätenverwalter den Eintrag der zweiten Prioritätsstufe **345** aus der Datenstruktur im Datenstrukturspeicher. Gleichzeitig setzt der Prioritätenverwalter den Eintrag der ersten Prioritätsstufe **340** wieder zurück an die erste Stelle in der Datenstruktur und informiert den ersten und zweiten Parallelprozessorkern **113**, **114** über die Änderung der Einträge der Prioritätsstufen in der Datenstruktur. Die Einträge der zweiten Prioritätsstufe **345** und/oder der ersten Prioritätsstufe **340** der Datenstruktur sind, wie

bereits oben beschrieben, unterhalb der oberen beiden Zeitachsen in **Fig. 3** zu sehen.

[0067] Der vierte vertikale Balken mit dem Bezugszeichen **307** der Zeitachse des zweiten Hauptprozessorkerns **112** in **Fig. 3** zeigt beispielsweise das Ende des parallelen Bearbeitungsabschnitts **303** im Programm des zweiten Hauptprozessorkerns **112** an, d.h. alle Teilaufgaben des zweiten Arbeitspakets **310** wurden abgearbeitet und die zweite Prioritätsstufe **345** wird aus der Datenstruktur entfernt. Dadurch ist die erste Prioritätsstufe **340** des ersten Arbeitspakets **305** wieder an erster Stelle der Einträge in der Datenstruktur, weshalb der erste und zweite Parallelprozessorkern **113, 114** erneut auf das erste Arbeitspaket **305** und die darin enthaltenen ersten bis vierten Teilaufgaben **321-324** zugreifen. Der Zugriff des ersten und zweiten Parallelprozessorkerns **113, 114** auf das erste Arbeitspaket **305** ist durch die beiden Pfeile mit den Bezugszeichen **570, 575** dargestellt. Beispielsweise kann die Zuordnung der Teilaufgaben auf die beiden Parallelprozessorkerne **113, 114** unverändert sein, wie oben beschrieben. Das heißt, die erste Teilaufgabe **321** kann für den ersten Parallelprozessorkern **113**, die zweite Teilaufgabe **322** für den zweiten Parallelprozessorkern **114**, sowie die vierte Teilaufgabe **324** für den ersten Hauptprozessorkern **111** zugeteilt sein. Die Zuteilung der vierten Teilaufgabe **324** für den ersten Hauptprozessorkern ist mit dem Pfeil mit dem Bezugszeichen **580** angedeutet. Die dritte Teilaufgabe **323** kann währenddessen bereits vollständig vom ersten Hauptprozessorkern **111** abgearbeitet worden sein. Beispielsweise können die Zwischenergebnisse der Berechnungen bei der Unterbrechung der ersten und zweiten Teilaufgabe **321, 322** auf der Zwischenergebnisspeichereinheit gesichert worden sein. Da der erste Hauptprozessorkern **111** und der erste und zweite Parallelprozessorkern **113, 114** auf die Zwischenergebnisspeichereinheit zugreifen, können die genannten ersten und zweiten Teilaufgaben **321, 322** direkt ab dem Unterbrechungspunkt weiter bearbeitet werden.

[0068] Sobald der zweite Hauptprozessorkern **112** im parallelen Bearbeitungsabschnitt **303** eine weitere Teilaufgabe von seinem Arbeitspaket **310** anfordert, aber keine unbearbeitete Teilaufgaben mehr verfügbar ist, wartet der zweite Hauptprozessorkern **112** bis die Parallelprozessorkerne **113, 114** die aktuell noch ausgeführten Teilaufgaben zu Ende abgearbeitet haben. Dann wird der parallele Bearbeitungsabschnitt **303** des zweiten Hauptprozessorkerns **112** verlassen und das zweite Arbeitspaket **310** gelöscht. Danach führt der zweite Hauptprozessorkern **112** beliebige weitere (nicht parallelisierte) Berechnungen aus, z.B. auf den Ergebnissen des vorherigen parallelen Bearbeitungsabschnitts **303** aufbauend. Das Ende **302** des Programms des zweiten Hauptprozessorkerns **112** wird mithilfe des fünften vertikalen Balkens angezeigt. Das Programm des zweiten Hauptprozessor-

kerns **112** hat dann sein finales Ergebnis berechnet und ist am Ende **302** angelangt.

[0069] Beispielsweise kann die Ausführung des Programms des zweiten Hauptprozessorkerns **112** in **Fig. 3** vor der Ausführung des Programms des ersten Hauptprozessorkerns **111** enden **302**. Das Ende **302** der Ausführung des Programms des zweiten Hauptprozessorkerns **112** ist durch den fünften vertikalen Balken der zweiten Zeitachse in **Fig. 3** gekennzeichnet. Im gezeigten Ausführungsbeispiel führen die Parallelprozessorkerne **113, 114** bis zum Ende **302** der Ausführung des Programms des ersten Hauptprozessorkerns **111** Berechnungen durch, allerdings nur, weil der erste Hauptprozessorkern **111** bis zum Ende **302** der Ausführung seines Programms einen parallelen Bearbeitungsabschnitt **303** aufweist. Daher wurde das Ende **302** der Ausführung des Programms des ersten Hauptprozessorkerns **111** mit in die beiden Zeitachsen der beiden Parallelprozessorkerne **113, 114** eingezeichnet. Da die Parallelprozessorkerne **113, 114** nur parallelisierte Berechnungen durchführen können, das heißt Teilaufgaben aus dem ersten und/oder zweiten Arbeitspaket **305, 310** abarbeiten, führen der erste und zweite Parallelprozessorkern **113, 114** nur solange Berechnungen aus, wie noch Teilaufgaben in den Arbeitspaketen **305, 310** verfügbar sind. Solange mindestens einer der beiden Hauptprozessorkerne **111, 112** aktiv ist, können aber zu einem späteren Zeitpunkt noch weitere neue Arbeitspakete mit Teilaufgaben erstellt werden, die die Parallelprozessorkerne **113, 114** abarbeiten können. Demnach kann das Ende **302** des Programms des zugeordneten Hauptprozessorkerns **111, 112** auch auf einen späteren Zeitpunkt fallen.

[0070] Sobald der erste Hauptprozessorkern **111** im parallelen Bearbeitungsabschnitt **303** eine weitere Teilaufgabe von seinem Arbeitspaket **305** anfordert, aber keine unbearbeitete mehr verfügbar ist, wartet der erste Hauptprozessorkern **111** bis die Parallelprozessorkerne **113, 114** die aktuell noch ausgeführten Teilaufgaben zu Ende abgearbeitet haben. Dann wird der parallele Bearbeitungsabschnitt **303** des ersten Hauptprozessorkerns **111** verlassen und das erste Arbeitspaket **305** gelöscht. Das Ende **302** des Programms des ersten Hauptprozessorkerns **111** wird mithilfe des fünften vertikalen Balkens angezeigt. Das Programm des ersten Hauptprozessorkerns **111** hat dann sein finales Ergebnis berechnet und ist am Ende **302** angelangt. Ebenfalls denkbar ist, dass der erste Hauptprozessorkern **111** nach dem parallelen Bearbeitungsabschnitt **303** analog zum zweiten Hauptprozessorkern **112** beliebige weitere (nicht parallelisierte) Berechnungen ausführen kann, z.B. auf den Ergebnissen des vorherigen parallelen Bearbeitungsabschnitts **303** aufbauend. Dies ist in **Fig. 3** nicht dargestellt.

[0071] In einer Ausgestaltung wird das erste und zweite Arbeitspaket **305, 310** mit der ersten und zweiten Menge an Teilaufgaben **320, 330**, sowie der Referenz auf die erste und zweite Prioritätsstufe **340, 345** zum Start **301** der Ausführung des jeweiligen Programms von dem ersten und zweiten Hauptprozessorkern **111, 112** erstellt. Ebenfalls ist denkbar, dass am Anfang eines weiteren parallelen Bearbeitungsabschnitts **303** im Programm des ersten Hauptprozessorkerns **111** und/oder des zweiten Hauptprozessorkerns **112** jeweils ein neues Arbeitspaket erstellt wird.

[0072] Statt der Unterbrechung der aktuell ausgeführten ersten bis vierten Teilaufgabe **321-324**, (fünftens bis siebten Teilaufgabe **331-333**) ist es ferner denkbar, die erste bis vierte Teilaufgabe **321-324** (fünfte bis siebte Teilaufgabe **331-333**) bis zum Ende abzuarbeiten und eine neue fünfte bis siebte Teilaufgabe **331-333** (erste bis vierte Teilaufgabe **321-324**) mit höherer Prioritätsstufe des jeweiligen parallelen Bearbeitungsabschnitts **303**, geeignet auf den ersten und zweiten Parallelprozessorkern **113, 114** und den zweiten Hauptprozessorkern **112** (ersten Hauptprozessorkern **111**) aufzuteilen. Die Parallelprozessorkerne **113, 114** werden auf diese Weise nicht gezwungen, ihre Berechnungen umgehend abzubrechen oder sie zu unterbrechen, falls der neuen fünften bis siebten Teilaufgabe **331-333** (ersten bis vierten Teilaufgabe **321-324**) eine höhere Priorität zugeordnet ist, als der aktuellen ersten bis vierten Teilaufgabe **321-324** (fünftens bis siebten Teilaufgabe **331-333**). Beispielsweise kann die Ausgestaltung zum Einsatz kommen, wenn die ersten bis siebten Teilaufgaben **321-324, 331-333** in dem ersten Arbeitspaket **305** und dem zweiten Arbeitspaket **310** relativ klein sind und die Abarbeitung von den Prozessorkernen demnach zügig erfolgen kann. Der Vorteil an dieser Ausgestaltung ist, dass der Rechendurchsatz ohne Abbruch oder Unterbrechung der ersten bis siebten Teilaufgabe **321-324, 331-333** auf den beiden Parallelprozessorkernen **113, 114** am höchsten ist.

[0073] Der sofortige Abbruch der aktuell ausgeführten ersten bis siebten Teilaufgaben **321-324, 331-333** auf den beiden Parallelprozessorkernen **113, 114** ohne Sicherung der Daten kann beispielsweise angewendet werden, wenn der erste oder zweite Hauptprozessorkern **111, 112** einen parallelen Bearbeitungsabschnitt **303** mit einer sehr kleinen Zykluszeit aufweisen und die Parallelisierung erforderlich ist, um die Deadlines einhalten zu können. Von Vorteil ist dabei, dass mithilfe der Ausgestaltung eine geringste mögliche Latenz erzielt werden kann, wobei die Latenz die Zeit zwischen dem Abbruch der Berechnungen der aktuellen ersten bis vierten Teilaufgabe **321-324** (fünftens bis siebten Teilaufgabe **331-333**) auf den Parallelprozessorkernen **113, 114** und dem Beginn der Berechnungen der neuen fünften bis siebten Teilaufgabe **331-333** (ersten bis vierten Teil-

aufgabe **321-324**) auf den Parallelprozessorkernen **113, 114**, die mit einer höheren Prioritätsstufe verknüpft ist, angibt. Eine solche Ausführungsform setzt das Erreichen einer möglichst geringen Latenz zum Ziel, daher können Zwischenergebnisse der ersten bis siebten Teilaufgabe **321-324, 331-333** mit einer niedrigen Prioritätsstufe verworfen werden.

[0074] Die Ausgestaltung, bei der die aktuell bearbeiteten ersten bis siebten Teilaufgaben **321-324, 331-333** unterbrochen und deren Zwischenergebnisse auf der Zwischenergebnisspeichereinheit gesichert werden, kann einem guten Kompromiss zwischen Latenz und Rechendurchsatz entsprechen. Es ist möglich, die oben beschriebenen verschiedenen Ausgestaltungen der Unterbrechung, des sofortigen Abbruchs oder der passenden Einsortierung der ersten bis siebten Teilaufgabe miteinander zu kombinieren und sie situationsbedingt geeignet umzusetzen.

[0075] Des Weiteren können die ersten bis siebten Teilaufgaben **321-324, 331-333** aus dem ersten und zweiten Arbeitspaket **305, 310** zu Beginn des jeweiligen parallelen Bearbeitungsabschnitts **303** im Programm des ersten und zweiten Hauptprozessorkerns **111, 112** verteilt werden. Ferner können die genannten ersten bis siebten Teilaufgaben **321-324, 331-333** zum Start **301** der Programmausführung fest an den ersten und zweiten Hauptprozessorkern **111, 112** und den ersten und zweiten Parallelprozessorkern **113, 114** verteilt werden. Durch die feste Zuteilung der ersten bis siebten Teilaufgaben **321-324, 331-333** aus dem ersten Arbeitspaket **305** und dem zweiten Arbeitspaket **310** kann der Verwaltungsaufwand reduziert werden, da die Anfrage des ersten und zweiten Parallelprozessorkerns **113, 114** nach einer freien ersten bis siebten Teilaufgabe **321-324, 331-333** entfällt. Auch können der erste Hauptprozessorkern **111** und/oder der zweite Hauptprozessorkern **112** nach erfolgter Abarbeitung der eigenen Teilaufgaben prüfen, ob noch freie erste bis siebte Teilaufgaben **321-324, 331-333** im ersten und/oder zweiten Arbeitspaket **305, 310** vorhanden sind. Diese noch freien ersten bis siebten Teilaufgaben **321-324, 331-333** können von dem ersten Hauptprozessorkern **111** und/oder dem zweiten Hauptprozessorkern **112** selbst bearbeitet werden, damit die Deadlines der jeweiligen Programme eingehalten werden können.

[0076] Werden mehr als die in den **Fig. 1** bis **Fig. 3** dargestellten zwei Hauptprozessorkerne **111, 112** eingesetzt, so können sich die Anzahl der Arbeitspakete und die Anzahl der Teilaufgaben von der in den **Fig. 2** und **Fig. 3** gezeigten und entsprechend beschriebenen Anzahl unterscheiden. In einem solchen Fall können die Anzahl der Arbeitspakete und die Anzahl der Teilaufgaben steigen, da mehr Rechenkapazität zur Verfügung gestellt werden kann. Die Rechenkapazität kann mit mehr Arbeitspaketen und mehr Teilaufgaben bestmöglich ausgenutzt wer-

den. Für die Ausgestaltungen in den **Fig. 2** und **Fig. 3** wurde jeweils ein erstes Arbeitspaket **305** und ein zweites Arbeitspaket **310**, sowie vier erste bis vierte Teilaufgaben **321-324** des ersten Arbeitspakets **305** und drei fünfte bis siebte Teilaufgaben **331-333** des zweiten Arbeitspakets **310** angenommen. In weiteren Ausgestaltungen können die Anzahl der Arbeitspakete und die Anzahl der Teilaufgaben auch bei unveränderter Zahl an Hauptprozessorkernen zu der in den **Fig. 2** und **Fig. 3** dargestellten und beschriebenen Ausführungsform abweichen.

[0077] In **Fig. 4** ist ein Programmablaufdiagramm **370** für den ersten und den zweiten Hauptprozessorkern **111, 112** nach den Ausgestaltungen der **Fig. 1** bis **Fig. 3** gezeigt. Im ersten Schritt **375** wird die Ausführung des jeweiligen Programms auf dem ersten und zweiten Hauptprozessorkern **111, 112** der **Fig. 2** und **Fig. 3** gestartet. Anschließend folgt eine Verzweigung in **Fig. 4**, in der in einem zweiten Schritt **366** geprüft wird, ob eine sequentielle Ausführung **365** des jeweiligen Programms, gemäß **Fig. 3** vorgesehen ist. Ist dies der Fall, so werden Aufgaben, die keine Parallelisierung erfordern von dem ersten und/oder zweiten Hauptprozessorkern **111, 112** gemäß den **Fig. 2** und **Fig. 3** sequentiell **365** bearbeitet. Ist im Programm keine sequentielle Bearbeitung **365** der Teilaufgaben vorgesehen, wird in einer zweiten Verzweigung in einem dritten Schritt **361** in **Fig. 4** geprüft, ob eine parallele Ausführung **360** der Teilaufgaben, gemäß **Fig. 3** möglich ist. Wenn dies der Fall ist, so können die Teilaufgaben vom ersten und zweiten Hauptprozessorkern **111, 112** und dem ersten und zweiten Parallelprozessorkern **113, 114** gemäß **Fig. 2** und **Fig. 3** bearbeitet werden. Dazu erstellen der erste und zweite Hauptprozessorkern **111, 112** zu Beginn des parallelen Bearbeitungsabschnitts **303** gemäß **Fig. 3**, wie oben beschrieben, das erste und zweite Arbeitspaket **305, 310**, das die erste und zweite Menge an Teilaufgaben **320, 330** und die Referenz auf die erste und zweite Prioritätsstufe **340, 345** des parallelen Bearbeitungsabschnitts **303** enthält. Zusätzlich wird die Referenz auf die erste und zweite Prioritätsstufe **340, 345** an den in **Fig. 2** dargestellten Prioritätenverwalter **350** übergeben und dieser trägt sie in die Datenstruktur **355** im Datenstrukturspeicher ein. Die Referenz auf die erste und zweite Prioritätsstufe **340, 345** wird vom Prioritätenverwalter **350** des Weiteren an den ersten und zweiten Parallelprozessorkern **113, 114** weitergegeben.

[0078] In einer dritten Verzweigung wird in einem vierten Schritt **380** im Programm geprüft, ob freie erste bis siebte Teilaufgaben **321-324, 331-333** gemäß der **Fig. 2** und **Fig. 3** im entsprechenden ersten und zweiten Arbeitspaket **305, 310** vorhanden sind. Dies geschieht durch die Anfrage des ersten und zweiten Hauptprozessorkerns **111, 112** sowie des ersten und zweiten Parallelprozessorkerns **113, 114** beim ersten und zweiten Arbeitspaket **305, 310**, sofern die ers-

ten und zweiten Hauptprozessorkerne **111, 112** bzw. ersten und zweiten Parallelprozessorkerne **113, 114** gerade keine Berechnungen ausführen. Nach erfolgreicher Zuteilung der ersten bis siebten Teilaufgaben **321-324, 331-333** an den ersten und zweiten Hauptprozessorkern **111, 112** sowie den ersten und zweiten Parallelprozessorkern **113, 114** führen der erste und zweite Hauptprozessorkern **111, 112** bzw. der erste und zweite Parallelprozessorkern **113, 114** die entsprechenden ersten bis siebten Teilaufgaben **321-324, 331-333** so lange aus, bis sie vollendet sind bzw. bis die ersten und zweiten Hauptprozessorkerne **111, 112** bzw. ersten und zweiten Parallelprozessorkerne **113, 114** vom Prioritätenverwalter **350** informiert werden, dass sich die Reihenfolge der Einträge der Prioritätsstufen in der Datenstruktur **355** geändert hat. Bei der Änderung der Reihenfolge der Einträge der ersten und zweiten Prioritätsstufe **340, 345** in der Datenstruktur **355** werden die bearbeiteten ersten bis siebten Teilaufgaben **321-324, 331-333** auf den beiden Parallelprozessorkernen **113, 114** unterbrochen und mit der Berechnung der ersten bis siebten Teilaufgabe **331-333, 321-324** des anderen, ersten oder zweiten Arbeitspakets **305, 310** begonnen.

[0079] Die ersten bis siebten Teilaufgaben **321-324, 331-333** gemäß der **Fig. 2** und **Fig. 3** können sofort unterbrochen werden oder bis zu einem vorgegebenen Unterbrechungspunkt weiter abgearbeitet und dann unterbrochen werden, wie oben beschrieben. Die Zwischenergebnisse können auf der Zwischenergebnisspeichereinheit gesichert werden, was eine Wiederaufnahme der Berechnungen der ersten bis siebten Teilaufgaben **321-324, 331-333** ab dem Unterbrechungspunkt schnell und einfach gestaltet. Nachdem eine erste bis siebte Teilaufgabe **321-324, 331-333** vollständig von dem ersten oder zweiten Parallelprozessorkern **113, 114** oder dem ersten oder zweiten Hauptprozessorkern **111, 112** abgearbeitet worden ist, fragen die ersten und/oder zweiten Hauptprozessorkerne **111, 112** bzw. ersten und/oder zweiten Parallelprozessorkerne **113, 114** erneut beim ersten oder zweiten Arbeitspaket **305, 310** an, ob es weitere unbearbeitete bzw. freie erste bis siebte Teilaufgaben **321-324, 331-333** gibt. Dieser vierte Schritt **380** in **Fig. 4** wird so lange wiederholt, bis keine freien ersten bis siebten Teilaufgaben **321-324, 331-333** im zugehörigen ersten und zweiten Arbeitspaket **305, 310** mehr vorhanden sind.

[0080] Die beiden Hauptprozessorkerne **111, 112** müssen warten, bis die Parallelprozessorkerne **113, 114** die dem jeweiligen Hauptprozessorkern **111, 112** zugehörigen aktuellen ersten bis siebten Teilaufgaben **321-324, 331-333** abgeschlossen haben, da es möglich ist, dass die Hauptprozessorkerne **111, 112** zwar keine freie erste bis siebte Teilaufgabe **321-324, 331-333** zugeteilt erhalten, die Parallelprozessorkerne **113, 114** können jedoch noch mit der Abarbeitung der ersten bis siebten Teilaufgabe **321-324, 331-333**

beschäftigt sein. Demnach kann erst mit der sequentiellen Ausführung **365** fortgefahren werden, wenn alle zugehörigen ersten bis siebten Teilaufgaben **321-324**, **331-333** bis zum Ende fertig abgearbeitet wurden.

[0081] In **Fig. 4** wird danach in dem zweiten Schritt **366** geprüft, ob im jeweiligen Programm des ersten und/oder zweiten Hauptprozessorkerns **111**, **112** eine sequentielle Ausführung **365** vorgesehen ist. Die weiteren Prüfungsschritte der Verzweigungen in **Fig. 4** werden wie oben beschrieben fortgeführt. Ergibt die Prüfung gemäß **Fig. 4** hinsichtlich der Art der Abarbeitung der ersten bis siebten Teilaufgaben **321-324**, **331-333** weder eine sequentielle Ausführung **365**, noch eine parallele Ausführung **360** der Berechnungen, so wird die Ausführung des Programms auf dem ersten Hauptprozessorkern **111** und/oder dem zweiten Hauptprozessorkern **112** im fünften Schritt **385** gemäß **Fig. 4** beendet.

[0082] **Fig. 5** zeigt das Programmblaufdiagramm **390** für den ersten und zweiten Parallelprozessorkern **113**, **114** nach den Ausgestaltungen der **Fig. 1** bis **Fig. 4**. In einem ersten Schritt **391** wird die Ausführung des jeweiligen Programms gestartet. Danach folgt eine Verzweigung, in der in einem zweiten Schritt **392** geprüft wird, ob in der in **Fig. 2** dargestellten Datenstruktur **355** ein erster oder zweiter Prioritätsstufeneintrag **340**, **345** vorhanden ist. Ist dies der Fall, so fragen der erste und zweite Parallelprozessorkern **113**, **114** gemäß den **Fig. 2** und **Fig. 3** in einem dritten Schritt **393** beim entsprechenden ersten oder zweiten Arbeitspaket **305**, **310** an, ob es freie erste bis siebte Teilaufgaben **321-324**, **331-333** zur Bearbeitung gibt. Dieser dritte Schritt **393** zur Prüfung in **Fig. 5** ist mit der zweiten Verzweigung angedeutet. Stehen freie erste bis siebte Teilaufgaben **321-324**, **331-333** zur Verfügung, so arbeiten der erste und zweite Parallelprozessorkern **113**, **114** diese ab. Auch hier kann es zu einer Unterbrechung und Sicherung der ersten bis siebten Teilaufgaben **321-324**, **331-333** bzw. einer Weiterberechnung der ersten bis siebten Teilaufgaben **321-324**, **331-333** bis zu einem vorgegebenen Unterbrechungspunkt und Sicherung der Zwischenergebnisse kommen, wie im Zusammenhang mit den **Fig. 2** und **Fig. 3** beschrieben worden ist. Nachdem eine erste bis siebte Teilaufgabe **321-324**, **331-333** vollständig von dem ersten oder zweiten Parallelprozessorkern **113**, **114** abgearbeitet worden ist, erfolgt im entsprechenden Programm des ersten und zweiten Parallelprozessorkerns **113**, **114** ein Rücksprung zur ersten Verzweigung, dem zweiten Schritt **392** in **Fig. 5**, in dem geprüft wird, ob ein erster oder zweiter Eintrag der Prioritätsstufe **340**, **345** in der Datenstruktur **355** vorhanden ist. Befindet sich aktuell kein Eintrag der Prioritätsstufe **340**, **345** in der Datenstruktur **355**, so erfolgt nach kurzer Wartezeit eine erneute Prüfung der Einträge in der Datenstruktur **355** entsprechend dem zweiten Schritt **392**.

[0083] Führt die Prüfung der zweiten Verzweigung im dritten Schritt **393** gemäß **Fig. 5** dazu, dass im entsprechenden ersten und/oder zweiten Arbeitspaket **305**, **310** derzeit keine freie erste bis siebte Teilaufgabe **321-324**, **331-333** vorhanden ist, so wird in der dritten Verzweigung in **Fig. 5** in einem vierten Schritt **394** geprüft, ob die Programme des ersten und zweiten Hauptprozessors **111**, **112** gemäß der Beschreibung zu **Fig. 4** bereits zu Ende ausgeführt worden sind. Nur für den Fall, dass die Ausführungen der Programme des ersten und des zweiten Hauptprozessorkerns **111**, **112** bereits beendet sind, endet auch das Programm des ersten und zweiten Parallelprozessorkerns **113**, **114** in fünften Schritt **395** in **Fig. 5**. Sind die Programme des ersten und zweiten Hauptprozessorkerns **111**, **112** hingegen nicht beendet, so erfolgt im Programm des ersten und zweiten Parallelprozessorkerns **113**, **114** ein Rücksprung in **Fig. 5** zu dem zweiten Schritt **392**.

[0084] Die Erfindung wurde im Detail durch bevorzugte Ausführungsbeispiele beschrieben. Sie ist jedoch nicht durch die offenbarten Beispiele eingeschränkt, da vom Fachmann andere Variationen daraus abgeleitet werden können, ohne dabei den Schutzbereich der Erfindung zu verlassen.

Bezugszeichenliste

100	Speicherprogrammierbare Steuerung
110	Datenverarbeitungseinrichtung
111	erster Hauptprozessorkern
112	zweiter Hauptprozessorkern
113	erster Parallelprozessorkern
114	zweiter Parallelprozessorkern
120	Steuerungseinrichtung
130	Kommunikationsschnittstelle
200	Maschine/Anlage
210	Aktoren
220	Sensoren
300	Verfahren zur Datenverarbeitung
301	Beginn Ausführung Programm
302	Ende Ausführung Programm
303	paralleler Bearbeitungsabschnitt
304	Beginn erster paralleler Bearbeitungsabschnitt
305	erstes Arbeitspaket
306	Beginn zweiter paralleler Bearbeitungsabschnitt
307	Beginn dritter paralleler Bearbeitungsabschnitt

310 zweites Arbeitspaket
320 erste Menge an Teilaufgaben
330 zweite Menge an Teilaufgaben
321 erste Teilaufgabe
322 zweite Teilaufgabe
323 dritte Teilaufgabe
324 vierte Teilaufgabe
331 fünfte Teilaufgabe
332 sechste Teilaufgabe
333 siebte Teilaufgabe
340 erste Prioritätsstufe
345 zweite Prioritätsstufe
347 erste sequentielle Prioritätsstufe
348 zweite sequentielle Prioritätsstufe
350 Prioritätenverwalter
355 Datenstruktur
360 Parallele Ausführung
365 Sequentielle Ausführung
370 Programmablaufdiagramm eines Hauptprozessorkerns
375 Start
380 Freie Teilaufgaben vorhanden im entsprechenden Arbeitspaket?
385 Ende
390 Programmablaufdiagramm eines Parallelprozessorkerns
391 Start
392 Prioritätsstufeneintrag in Datenstruktur vorhanden?
393 Freie Teilaufgaben vorhanden im entsprechenden Arbeitspaket?
394 Alle Programme ausgeführt?
395 Ende
500 erster Pfeil
505 zweiter Pfeil
510 dritter Pfeil
515 vierter Pfeil
520 fünfter Pfeil
525 sechster Pfeil
530 siebter Pfeil
535 achter Pfeil
540 neunter Pfeil

545 zehnter Pfeil
550 elfter Pfeil
555 zwölfter Pfeil
560 dreizehnter Pfeil
565 vierzehnter Pfeil
570 fünfzehnter Pfeil
575 sechzehnter Pfeil
580 siebzehnter Pfeil

Patentansprüche

1. Verfahren (300) zur Datenverarbeitung auf einer speicherprogrammierbaren Steuerung (100) zur Steuerung von Maschinen oder Anlagen eines Automatisierungssystems, die mehrere Hauptprozessorkerne (111, 112) und wenigstens einen Parallelprozessorkern (113, 114) umfasst, wobei den Hauptprozessorkernen (111, 112) jeweils ein ausführbares Programm einer Steuerungsaufgabe zugeordnet ist, wobei jedes Programm jeweils wenigstens einen parallelen Bearbeitungsabschnitt (303) mit einem Arbeitspaket (305, 310) aufweist, das mehrere Teilaufgaben (321-324, 331-333) umfasst, wobei den parallelen Bearbeitungsabschnitten (303) in den jeweiligen Programmen eine Priorität mit einer vorgegebenen Prioritätsstufe (340, 345) zugewiesen ist, wobei die jeweiligen Prioritätsstufen (340, 345) in eine Datenstruktur (355) eingefügt werden, sobald der jeweilige Hauptprozessorkern (111, 112) im Programm an dem parallelen Bearbeitungsabschnitt (303) angelangt ist, wobei der wenigstens eine Parallelprozessorkern (113, 114) prüft, ob in der Datenstruktur (355) Einträge vorhanden sind und wobei, sofern Einträge vorhanden sind, Teilaufgaben (321-324, 331-333) aus dem Arbeitspaket (305, 310) des Hauptprozessorkerns (111, 112), dessen Prioritätsstufe (340, 345) an erster Stelle der Einträge in der Datenstruktur (355) steht, von dem wenigstens einen Parallelprozessorkern (113, 114) abgearbeitet werden, wobei eine Echtzeitbedingung der Steuerungsaufgabe eingehalten wird, indem Ausführungszeiten der Programme für die jeweiligen Hauptprozessorkerne (111, 112) so festgesetzt sind, dass die mehreren Hauptprozessorkerne (111, 112) in der Lage sind, die Teilaufgaben (321-324, 331-333) aus den Arbeitspaketen (305, 310) ohne die Unterstützung des wenigstens einen Parallelprozessorkerns (113, 114) abzuarbeiten, wobei die mehreren Hauptprozessorkerne (111, 112) Teilaufgaben (321-324, 331-333) abarbeiten, die nicht von dem wenigstens einen Parallelprozessorkern (113, 114) abgearbeitet werden, und wobei es durch die Parallelisierung für die am höchsten priorisierten Programme der mehreren Hauptprozessorkerne möglich ist, die kürzere Ausführungszeit

der Programme durch die Parallelisierung einzuplanen, und auf diese Weise eine Deadline, die kleiner als eine rein sequentielle Ausführungszeit der Programme ist, einzuhalten.

2. Verfahren nach Anspruch 1, wobei die Verteilung der Teilaufgaben (321-324, 331-333) aus den jeweiligen Arbeitspaketen (305, 310) dynamisch erfolgt, wobei die Hauptprozessorkerne (111, 112) und der wenigstens eine Parallelprozessorkern (113, 114) auf unbearbeitete Teilaufgaben (321-324, 331-333) der Arbeitspakete (305, 310) zur Abarbeitung zugreifen, und wobei die Teilaufgaben (321-324, 331-333), die dem wenigstens einen Parallelprozessorkern (113, 114) für die Abarbeitung zugeteilt werden, auch auf dem entsprechenden Hauptprozessorkern (111, 112) ausführbar sind.

3. Verfahren nach Anspruch 2, wobei nur für den die entsprechende Teilaufgabe (321-324, 331-333) aktuell abarbeitenden Hauptprozessorkern (111, 112) oder den die entsprechende Teilaufgabe (321-324, 331-333) aktuell abarbeitenden wenigstens einen Parallelprozessorkern (113, 114) ein Zugriff auf die entsprechende Teilaufgabe (321-324, 331-333) aus einem der Arbeitspakete (305, 310) besteht, und wobei die Teilaufgaben (321-324, 331-333) aus den Arbeitspaketen (305, 310) einzeln und nacheinander an die Hauptprozessorkerne (111, 112) und den wenigstens einen Parallelprozessorkern (113, 114) verteilt werden.

4. Verfahren nach Anspruch 1, wobei die Verteilung der Teilaufgaben (321-324, 331-333) aus den jeweiligen Arbeitspaketen (305, 310) statisch erfolgt, wobei den mehreren Hauptprozessorkernen (111, 112) und dem wenigstens einen Parallelprozessorkern (113, 114) dazu jeweils eine bestimmte Anzahl an Teilaufgaben (321-324, 331-333) aus dem gleichen Arbeitspaket (305, 310) zur parallelen Abarbeitung (360) fest zugeteilt wird.

5. Verfahren nach einem der Ansprüche 1 bis 4, wobei dem wenigstens einen Parallelprozessorkern (113, 114) Änderungen in der Reihenfolge der Einträge der Prioritätsstufen (340, 345) in der Datenstruktur (355) mitgeteilt werden.

6. Verfahren nach Anspruch 5, wobei die Berechnungen aktueller Teilaufgaben (321-324, 331-333) im parallelen Bearbeitungsabschnitt (303) auf dem wenigstens einen Parallelprozessorkern (113, 114) unterbrochen werden, wenn ein neuer Eintrag der Prioritätsstufe (340, 345) an die erste Stelle der Datenstruktur (355) gesetzt wird.

7. Verfahren nach Anspruch 5, wobei die aktuellen Teilaufgaben (321-324, 331-333) von dem wenigstens einen Parallelprozessorkern (113, 114) bis zu ei-

nem vorgegebenen Unterbrechungspunkt weiter bearbeitet und dann unterbrochen werden.

8. Verfahren nach Anspruch 7, wobei der Unterbrechungspunkt insbesondere in einer Kontrollstruktur in den Teilaufgaben (321-324, 331-333) der Arbeitspakete (305, 310) festgelegt ist.

9. Verfahren nach Anspruch 7 oder 8, wobei ein Zwischenergebnis der Berechnungen der aktuellen Teilaufgaben (321-324, 331-333) der Arbeitspakete (305, 310) gesichert wird.

10. Speicherprogrammierbare Steuerung (SPS) (100) zur Steuerung von Maschinen oder Anlagen eines Automatisierungssystems, umfassend: eine Kommunikationsschnittstelle (130), zum Einlesen von Sensordaten und zum Ausgeben von Aktordaten, eine Datenverarbeitungseinrichtung (110), die mehrere Hauptprozessorkerne (111, 112) und wenigstens einen Parallelprozessorkern (113, 114) zum Ausführen einer Steuerungsaufgabe umfasst, um aus den Sensordaten Aktordaten zu erzeugen, eine Steuerungseinrichtung (120), die einen Speicher zum Speichern einer Datenstruktur (355) mit Prioritätsstufen (340, 345) und einen Prioritätenverwalter (350) zum Verwalten der Prioritätsstufen (340, 345) in der Datenstruktur (355) aufweist, wobei den Hauptprozessorkernen (111, 112) jeweils ein ausführbares Programm der Steuerungsaufgabe zugeordnet ist, wobei jedes Programm jeweils wenigstens einen parallelen Bearbeitungsabschnitt (303) mit einem Arbeitspaket (305, 310) aufweist, und das Arbeitspaket (305, 310) mehrere Teilaufgaben (321-324, 331-333) umfasst, wobei den parallelen Bearbeitungsabschnitten (303) in den jeweiligen Programmen die vorgegebene Prioritätsstufe (340, 345) zugewiesen ist, wobei der Prioritätenverwalter (350) ausgebildet ist, die jeweiligen Prioritätsstufen (340, 345) in die Datenstruktur (355) im Speicher einzufügen, sobald der jeweilige Hauptprozessorkern (111, 112) im Programm an dem parallelen Bearbeitungsabschnitt (303) angelangt ist und dabei den Eintrag mit der höchsten Prioritätsstufe (340, 345) an die erste Stelle der Einträge in der Datenstruktur (355) im Speicher zu setzen, wobei der wenigstens eine Parallelprozessorkern (113, 114) ausgebildet ist, die Datenstruktur (355) im Speicher auf Einträge der Prioritätsstufen (340, 345) zu prüfen, und sofern Einträge vorhanden sind, Teilaufgaben (321-324, 331-333) aus dem Arbeitspaket (305, 310) des Hauptprozessorkerns (111, 112) abzuarbeiten, dessen Prioritätsstufe (340, 345) an erster Stelle der Einträge in der Datenstruktur (355) steht, wobei die speicherprogrammierbare Steuerung (100) ausgebildet ist, eine Echtzeitbedingung der Steuerungsaufgabe einzuhalten, indem Ausführungszeiten

der Programme für die jeweiligen Hauptprozessorkerne (111, 112) so festgesetzt sind, dass die mehreren Hauptprozessorkerne (111, 112) in der Lage sind, die Teilaufgaben (321-324, 331-333) aus den Arbeitspaketen (305, 310) ohne die Unterstützung des wenigstens einen Parallelprozessorkerns (113, 114) abzuarbeiten, und wobei die mehreren Hauptprozessorkerne (111, 112) ausgelegt sind, Teilaufgaben (321-324, 331-333) abzuarbeiten, die nicht von dem wenigstens einen Parallelprozessorkern (113, 114) abgearbeitet werden, und und wobei durch die Parallelisierung für die am höchsten priorisierten Programme der mehreren Hauptprozessorkerne möglich ist, die kürzere Ausführungszeit der Programme durch die Parallelisierung einzuplanen, und auf diese Weise eine Deadline, die kleiner als eine rein sequentielle Ausführungszeit der Programme ist, einzuhalten.

11. Speicherprogrammierbare Steuerung nach Anspruch 10, wobei die Steuerungseinrichtung (120) auf mehreren Hauptprozessorkernen (111, 112) verteilt ausgeführt ist.

12. Speicherprogrammierbare Steuerung nach Anspruch 10 oder 11, wobei der Prioritätenverwalter (350) ausgelegt ist, den ersten Eintrag der Prioritätsstufen (340, 345) der mehreren Hauptprozessorkerne (111, 112) der Datenstruktur (355) durch einen neuen Eintrag zu ersetzen, wenn dieser eine höhere Prioritätsstufe (340, 345) aufweist, und wobei der Prioritätenverwalter (350) ausgebildet ist, den wenigstens einen Parallelprozessorkern (113, 114) über die Änderung der Einträge der Prioritätsstufen (340, 345) der mehreren Hauptprozessorkerne (111, 112) in der Datenstruktur (355) zu informieren, um aktuelle Berechnungen von Teilaufgaben (321-324, 331-333) aus dem entsprechenden Arbeitspaket (305, 310) zu unterbrechen.

13. Speicherprogrammierbare Steuerung nach Anspruch 12, wobei der wenigstens eine Parallelprozessorkern (113, 114) ausgelegt ist, die Berechnungen der aktuellen Teilaufgaben (321-324, 331-333) aus dem entsprechenden Arbeitspaket (305, 310) zu unterbrechen, wenn ein neuer Eintrag der Prioritätsstufen (340, 345) an die erste Stelle der Einträge in der Datenstruktur (355) im Speicher gesetzt wird und die Zwischenergebnisse der Berechnungen an die Steuerungseinrichtung (120) zu übergeben, und wobei die Steuerungseinrichtung (120) ausgebildet ist, die Zwischenergebnisse der Berechnungen der aktuellen Teilaufgaben (321-324, 331-333) aus den jeweiligen Arbeitspaketen (320, 330) auf einer Speichereinheit zu sichern, auf die die mehreren Hauptprozessorkerne (111, 112) und der wenigstens eine Parallelprozessorkern (113, 114) zugreifen.

14. Speicherprogrammierbare Steuerung nach einem der Ansprüche 10 bis 13, wobei der Prioritätenverwalter (350) ausgelegt ist, die Einträge der Prioritätsstufen (340, 345) der mehreren Hauptprozessorkerne (111, 112) aus der Datenstruktur (355) zu entfernen, wenn die Teilaufgaben (321-324, 331-333) aus den entsprechenden Arbeitspaketen (305, 310) vollständig abgearbeitet sind.

Es folgen 5 Seiten Zeichnungen

Anhängende Zeichnungen

Fig. 1

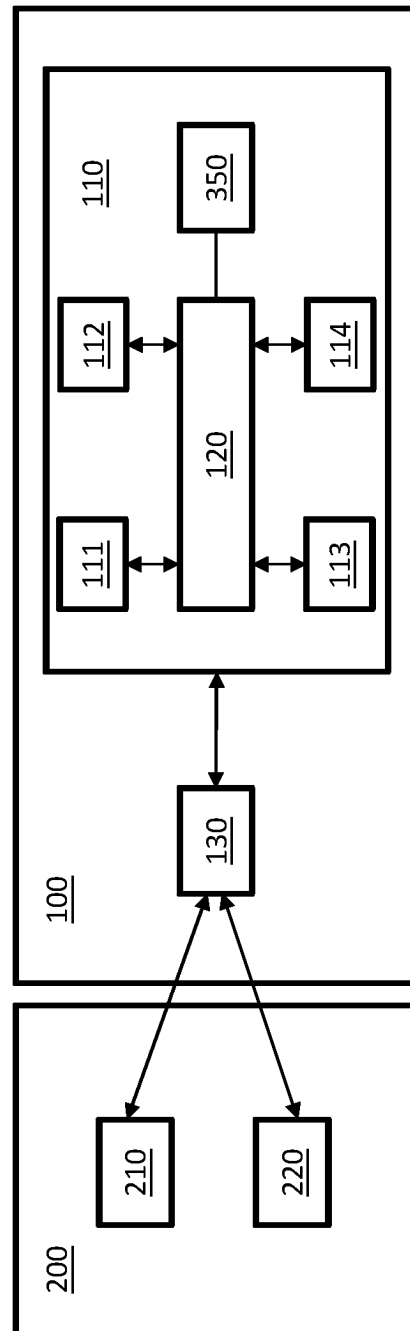


Fig. 2

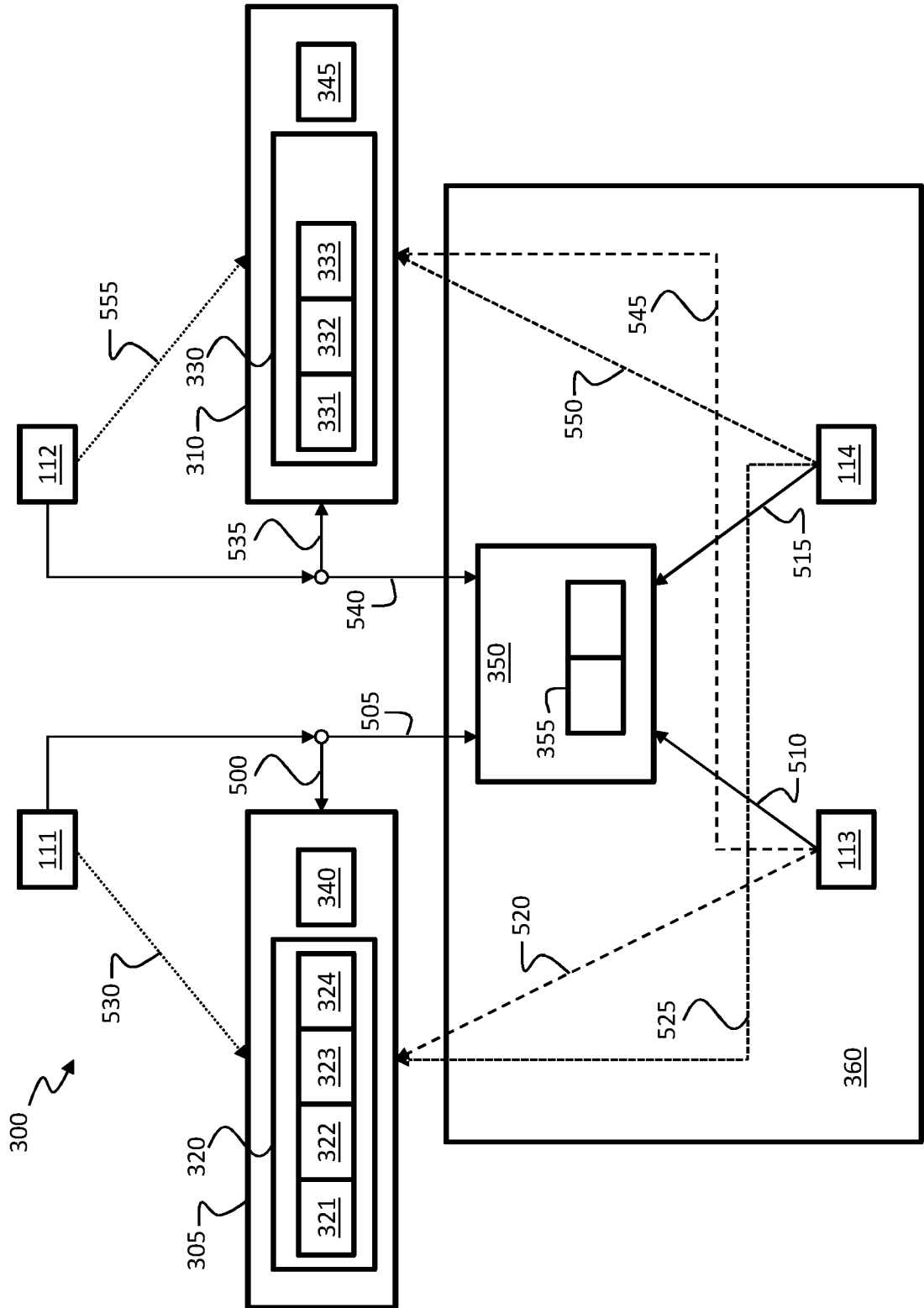
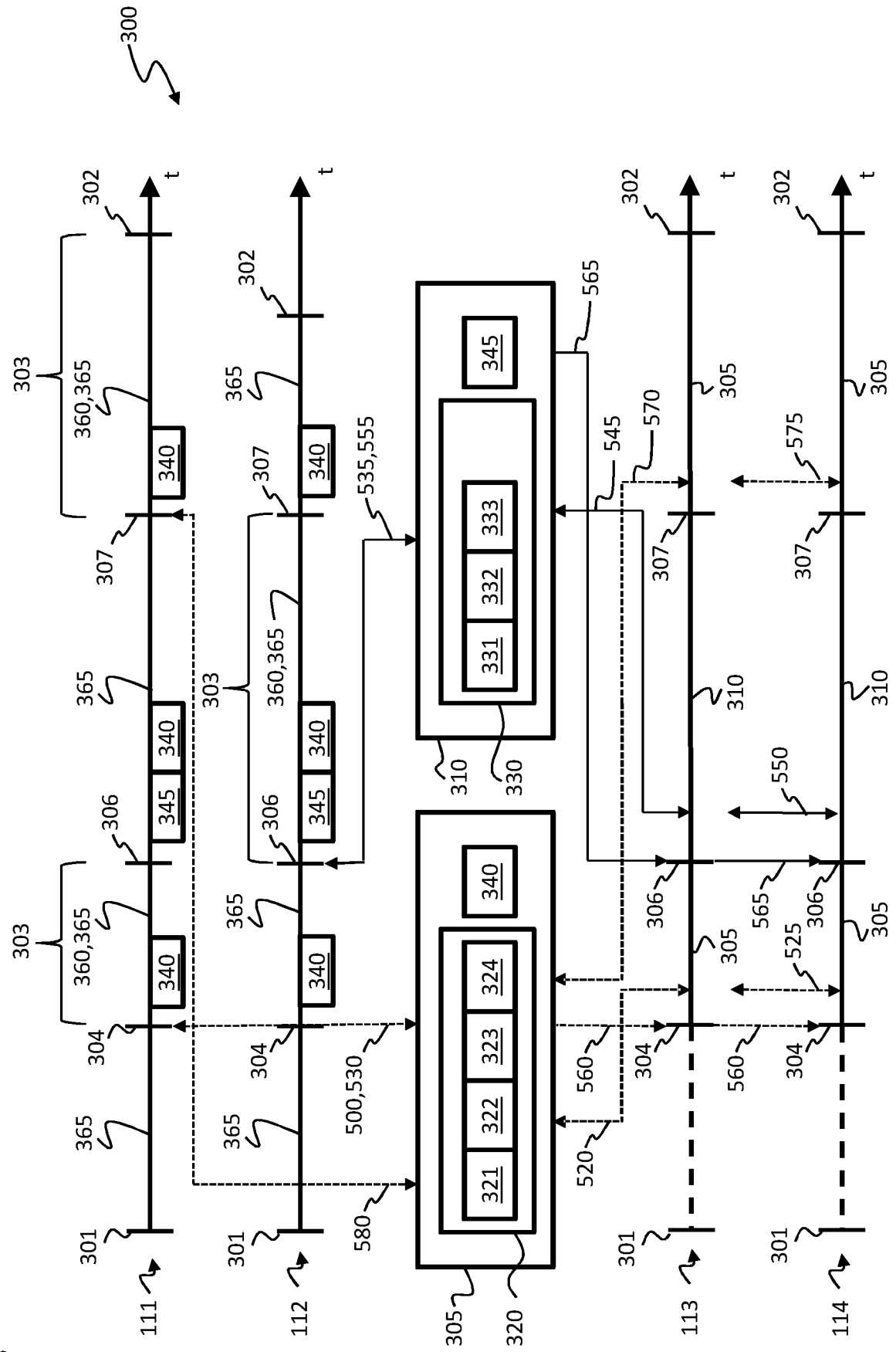


Fig. 3



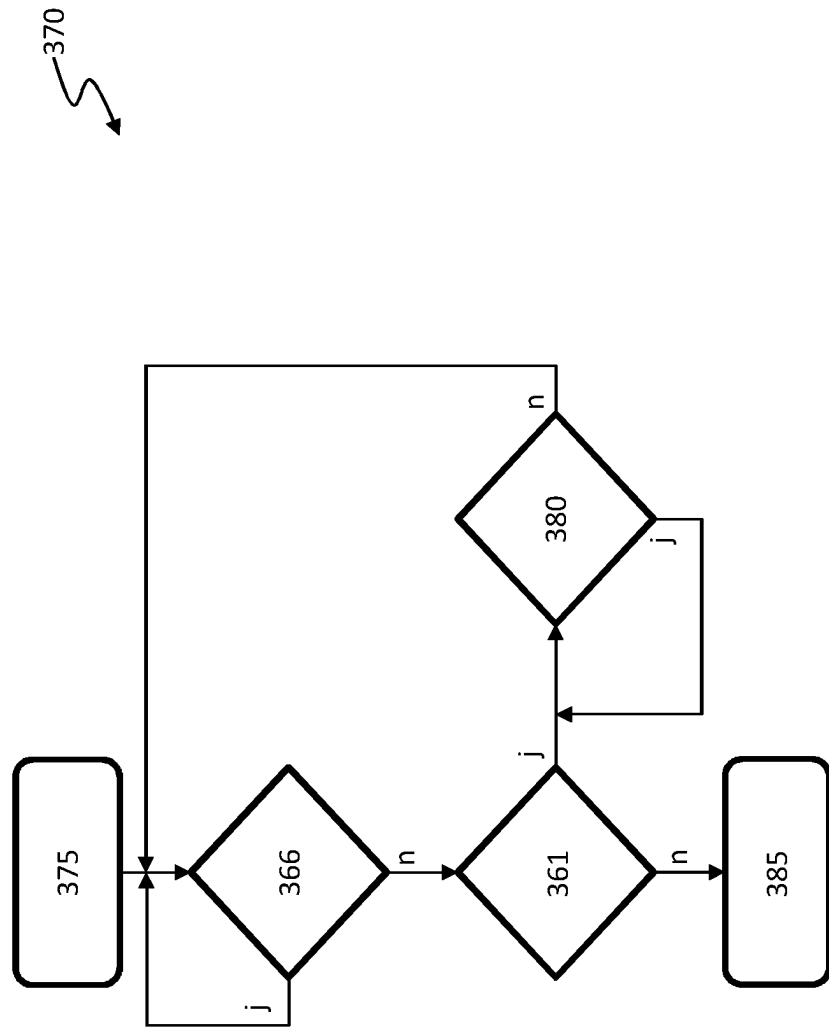


Fig. 4

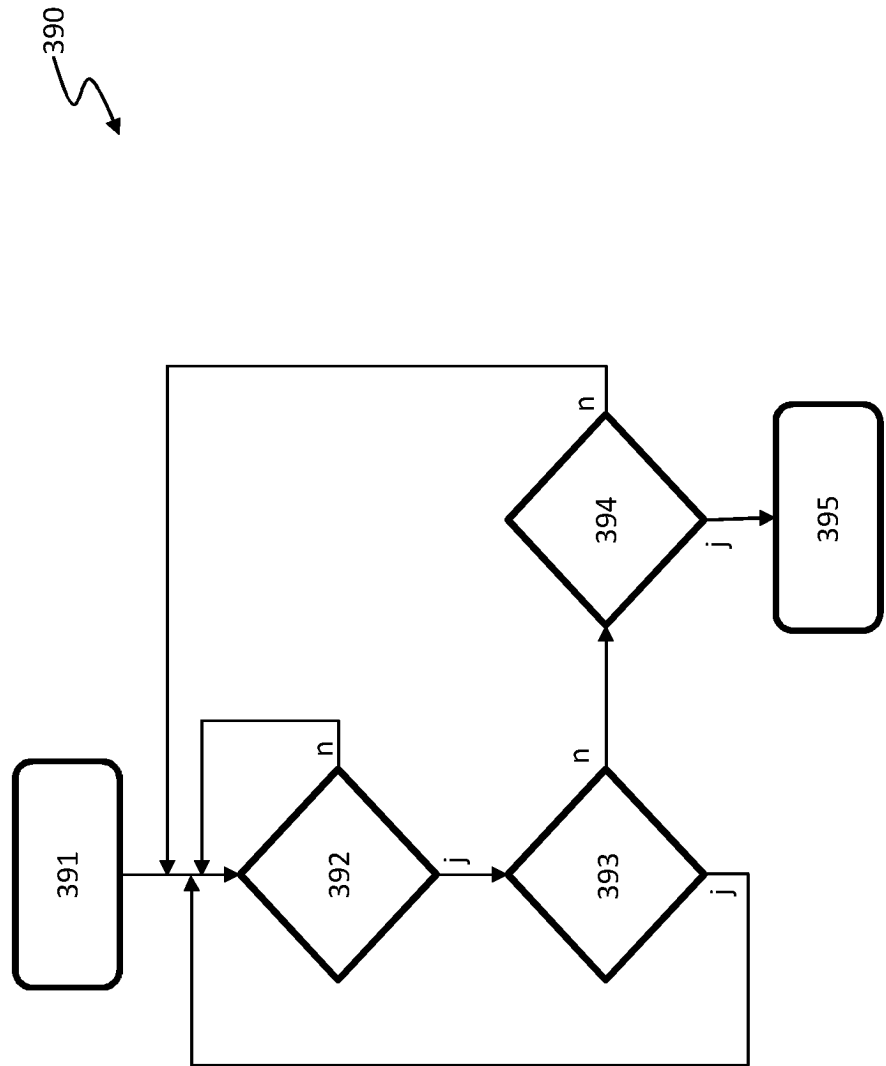


Fig. 5