



(12) **Offenlegungsschrift**

(21) Aktenzeichen: **10 2018 125 090.8**

(22) Anmeldetag: **10.10.2018**

(43) Offenlegungstag: **16.04.2020**

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

(71) Anmelder:
Beckhoff Automation GmbH, 33415 Verl, DE

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

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

(56) Ermittelte Stand der Technik:

Mohanty, R. et al.: Priority Based Dynamic Round Robin (PBDRR) Algorithm with Intelligent

Time Slice for Soft Real Time Systems. In: (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 2, No.2, February 2011. S. 46 - 50

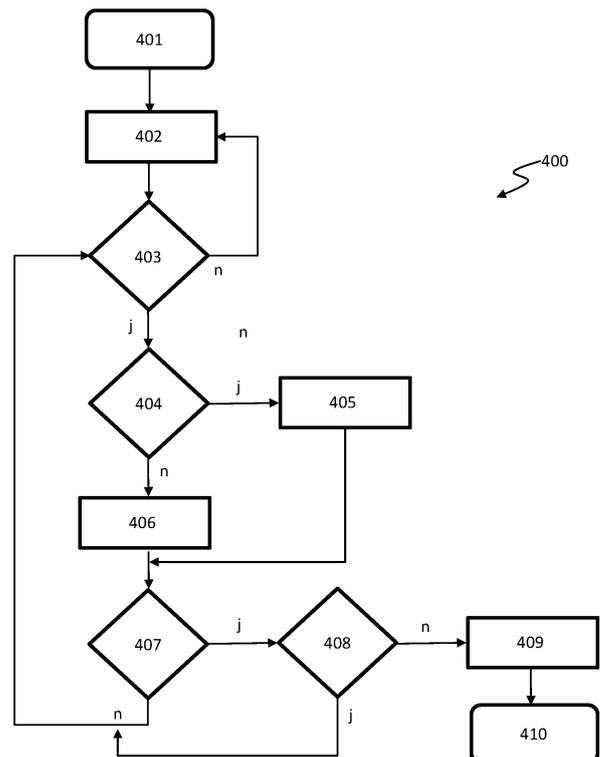
Rajput, I. S. et al.: A Priority based Round Robin CPU Scheduling Algorithm for Real Time Systems. In: International Journal of Innovations in Engineering and Technology (IJET), Vol. 1, Oct. 2012. S. 1 - 11

Prüfungsantrag gemäß § 44 PatG ist gestellt.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

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

(57) Zusammenfassung: Die Erfindung betrifft ein Verfahren zum Optimieren einer Datenverarbeitung auf einer speicherprogrammierbaren Steuerung (SPS). Die SPS weist eine Steuerungsaufgabe auf, die mehrere ausführbare Programme umfasst, wobei wenigstens zwei Programme der Steuerungsaufgabe jeweils wenigstens einen parallelen Bearbeitungsabschnitt mit einem Arbeitspaket mit mehreren Teilaufgaben aufweisen. Den parallelen Bearbeitungsabschnitten in den jeweiligen Programmen ist eine Priorität mit einer vorgegebenen Prioritätsstufe zugewiesen, wobei die jeweiligen Prioritätsstufen in eine Datenstruktur eingefügt werden, sobald ein Ausführen des Programms an dem parallelen Bearbeitungsabschnitt angelangt ist. Wenigstens ein Parallelprozessorkern prüft, ob in der Datenstruktur Einträge vorhanden sind und sofern Einträge vorhanden sind, arbeitet er Teilaufgaben aus dem Arbeitspaket des Programms ab, dessen Prioritätsstufe an erster Stelle der Einträge in der Datenstruktur steht. Während eines Programmzyklus wird ein zu erwartender Rechendurchsatz der Steuerungsaufgabe laufend bewertet und wenigstens eine der Prioritätsstufen der parallelen Bearbeitungsabschnitte der jeweiligen Programme dynamisch angepasst, wenn sich der Rechendurchsatz der Steuerungsaufgabe dadurch erhöht.



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 steuert oder regelt die Aktoren und Sensoren der Maschine oder der Anlage per Kommunikationsschnittstelle. Diese kann beispielsweise als Feldbusystem verwirklicht sein, wobei die Aktoren und Sensoren der Maschine oder der Anlage über das Feldbussystem vernetzt sein können.

[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 es sich bei den genannten Daten um einzelne Werte oder Wertegruppen handeln kann. 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 Zeit-

punkt 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 Programmzyklus-Zeiten in der SPS können sich beispielsweise je nach Anwendung in einem Bereich von 100 μ s bis 20 ms oder bei komplexen Aufgaben in einem Bereich von 50 ms bis 100 ms befinden. Aufgrund der verschiedenen Programmzyklus-Zeiten 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 der Deadline und/oder den Abhängigkeiten der jeweiligen Programme zu den anderen Programmen ermittelt werden, wenn die Programme beispielsweise auf die Ergebnisse der anderen Programme angewiesen sind. Die Prioritätsstufe kann ferner einem ganzzahligen Wert entsprechen.

[0006] Da viele SPS Prozessoren mit mehreren Prozessorkernen 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 durch eine zumindest teilweise parallele Ausführung der entsprechenden Aufgaben zu reduzieren. Dabei kann sich ein sogenannter Rechendurchsatz der Steuerungsaufgabe aus einer zeitlichen Unterbrechungsdauer der Programme, einer Anzahl an Unterbrechungen der Programme bzw. einer Rechenzeit der jeweiligen Programme ergeben. Ferner kann der Rechendurchsatz der Steuerungsaufgabe von einer Auslastung der Prozessorkerne abhängen. Werden die Programme und Aufgaben auf mehrere Prozessorkerne aufgeteilt, so kann es aufgrund der zugeordneten Prioritätsstufen vorkommen, dass Aufgaben, die gerade parallel ausgeführt werden von

anderen Aufgaben, die mit einer höheren Prioritätsstufe im entsprechenden Programm verknüpft sind, auf den parallel arbeitenden Prozessorkernen unterbrochen werden. Unterbrechungen von Aufgaben erhöhen den sogenannten Verwaltungsaufwand und können zeitlich voneinander variieren. Der Verwaltungsaufwand umfasst die Aktivitäten, die beim Unterbrechen der Abarbeitung der Aufgaben in den Programmen erforderlich sind. Die Aktivitäten können beispielsweise im Übergeben und Sichern von Zwischenergebnissen der unterbrochenen Aufgaben liegen - andernfalls müssen die Aufgaben der Programme wieder von Neuem abgearbeitet werden. Folglich kann der Rechendurchsatz bei einem hohen Verwaltungsaufwand sinken, da während der Unterbrechungsdauer der Programme in der Regel keine Berechnungen durchgeführt werden. Ebenso kann die Rechenzeit der Programme einen Einfluss auf den Rechendurchsatz nehmen, da Programme mit kurzer Programmzyklus-Zeit gegebenenfalls ohne Parallelisierung schneller rein sequentiell ausgeführt werden können, als auf den parallel arbeitenden Prozessorkernen. Denn die Programme auf den parallel arbeitenden Prozessorkernen können unter Umständen durch höher priorisierte andere Programme unterbrochen werden, wodurch sich die Gesamtausführungszeit der Programme verlängern kann. Werden die Prozessorkerne nicht vollständig ausgelastet, z.B. indem die Programme sequentiell ausgeführt werden, statt parallel von den dafür vorgesehenen Prozessorkernen, so ist es möglich, dass die Rechenleistung der für die Parallelisierung vorgesehenen Prozessorkerne nicht vollständig ausgenutzt wird. Dies kann den Rechendurchsatz der Steuerungsaufgabe ebenfalls reduzieren.

[0007] Die Aufgabe der vorliegenden Erfindung besteht darin ein Verfahren zum Optimieren einer Datenverarbeitung anzugeben, das die vorhandenen Ressourcen bestmöglich nutzt. Es ist weiterhin Aufgabe der Erfindung, eine verbesserte speicherprogrammierbare Steuerung anzugeben.

[0008] Diese Aufgabe wird durch die Merkmale der unabhängigen Ansprüche gelöst. Weitere vorteilhafte Ausführungsformen der Erfindung sind in den abhängigen Ansprüchen angegeben.

[0009] Erfindungsgemäß wird ein Verfahren zum Optimieren einer Datenverarbeitung auf einer speicherprogrammierbaren Steuerung (SPS) vorgeschlagen, wobei die SPS wenigstens einen Parallelprozessorkern umfasst. Die SPS weist ferner eine Steuerungsaufgabe auf, die mehrere ausführbare Programme umfasst. Wenigstens zwei Programme der Steuerungsaufgabe weisen jeweils wenigstens einen parallelen Bearbeitungsabschnitt mit einem Arbeitspaket auf, und das Arbeitspaket umfasst mehrere Teilaufgaben. Den parallelen Bearbeitungsabschnitten in den jeweiligen Programmen ist eine Priorität

mit einer vorgegebenen Prioritätsstufe zugewiesen, wobei die jeweiligen Prioritätsstufen in eine Datenstruktur eingefügt werden, sobald ein Ausführen des Programms an dem parallelen Bearbeitungsabschnitt angelangt ist. Der wenigstens eine Parallelprozessorkern prüft, ob in der Datenstruktur Einträge vorhanden sind und sofern Einträge vorhanden sind, arbeitet der wenigstens eine Parallelprozessorkern Teilaufgaben aus dem Arbeitspaket des Programms ab, dessen Prioritätsstufe an erster Stelle der Einträge in der Datenstruktur steht. Während eines Programmzyklus wird ein zu erwartender Rechendurchsatz der Steuerungsaufgabe laufend bewertet und wenigstens eine der Prioritätsstufen der parallelen Bearbeitungsabschnitte der jeweiligen Programme dynamisch angepasst, wenn sich der Rechendurchsatz der Steuerungsaufgabe dadurch erhöht.

[0010] Ein Parallelprozessorkern bezeichnet einen Prozessorkern der SPS, der für die parallele Abarbeitung von Teilaufgaben vorgesehen ist. Zusätzlich kann die SPS weitere Prozessorkerne, sogenannte Hauptprozessorkerne, aufweisen. Die Hauptprozessorkerne können jeweils ein oder mehrere Programme der Steuerungsaufgabe ausführen. Die mehreren Programme der Steuerungsaufgabe können Programme aufweisen, die rein sequentiell ausgeführt werden, wobei wenigstens zwei Programme der Steuerungsaufgabe wenigstens einen parallelen Bearbeitungsabschnitt umfassen. Diese parallelen Bearbeitungsabschnitte der Programme können mit der Unterstützung der Parallelprozessorkerne ausgeführt werden.

[0011] Der Rechendurchsatz der Steuerungsaufgabe kann über die zeitliche Unterbrechungsdauer und die Anzahl der Unterbrechungen der parallelen Bearbeitungsabschnitte der Programme bewertet werden. Denn während der Unterbrechungsdauer der Abarbeitung der Teilaufgaben der parallelen Bearbeitungsabschnitte der Programme werden keine Berechnungen durchgeführt, sondern Zwischenergebnisse übergeben und gesichert, die Einträge der Prioritätsstufen in der Datenstruktur werden neu sortiert, der wenigstens eine Parallelprozessorkern muss vor der Abarbeitung der Teilaufgaben des Arbeitspakets eines weiteren Programms mit parallelem Bearbeitungsabschnitt erst nach freien Teilaufgaben für die Abarbeitung anfragen oder bei der Wiederaufnahme der Abarbeitung der unterbrochenen Teilaufgaben zunächst auf den Zwischenergebnisspeicher zugreifen, um die bereits berechneten Zwischenergebnisse als Ausgangspunkt für die weitere Abarbeitung der unterbrochenen Teilaufgaben nutzen zu können. Insofern wirken sich die zeitliche Unterbrechungsdauer, sowie die Anzahl der Unterbrechungen der Programme mit parallelen Bearbeitungsabschnitten auf den Rechendurchsatz der Steuerungsaufgabe aus, da sie die Gesamtausführungszeit der Programme verlängern. Demnach könnte ein maximaler Re-

chendurchsatz der Steuerungsaufgabe ohne Unterbrechungen der Programme mit parallelen Bearbeitungsabschnitten erzielt werden.

[0012] Die Anzahl der Unterbrechungen der Programme mit parallelen Bearbeitungsabschnitten kann mithilfe der dynamischen Anpassung des Werts der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts der Programme reduziert werden, da der Wert der wenigstens einen Prioritätsstufe gezielt an die jeweilige Situation angepasst werden kann, um den Verwaltungsaufwand der Steuerungsaufgabe zu reduzieren. Werden Teilaufgaben eines Arbeitspakets gerade parallel ausgeführt und von anderen Teilaufgaben eines weiteren Arbeitspakets mit höherer Prioritätsstufe des entsprechenden parallelen Bearbeitungsabschnitts des jeweiligen Programms unterbrochen, so kann der Wert der wenigstens einen Prioritätsstufe der Teilaufgaben direkt innerhalb des Programmzyklus verändert werden, um eine weitere Unterbrechung der Teilaufgaben der entsprechenden parallelen Bearbeitungsabschnitte und zusätzlichen Verwaltungsaufwand zu vermeiden. Die Anpassung des Werts der wenigstens einen Prioritätsstufe kann ebenso für einen neuen Programmzyklus erfolgen, in dem das jeweilige Programm wieder von Anfang an ausgeführt wird.

[0013] Damit der Wert der wenigstens einen Prioritätsstufe nicht für jeden Programmzyklus verringert wird, kann über eine zusätzliche Bedingung, zum Beispiel in der Form, dass der Wert der wenigstens einen Prioritätsstufe für den neuen Programmzyklus mindestens dem halben Wert der vorgegebenen wenigstens einen Prioritätsstufe entsprechen muss, sichergestellt werden, dass der Wert der wenigstens einen Prioritätsstufe der Programme nicht ungewollt auf einen minimalen Wert gesetzt wird. Denn der minimale Wert der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts der Programme bedeutet, dass das jeweilige Programm sequentiell auf dem entsprechenden Hauptprozessorkern ausgeführt wird. Auf diese Weise kann die Rechenleistung des wenigstens einen Parallelprozessorkerns gegebenenfalls nicht vollständig ausgenutzt sein, wobei die nicht genutzte Rechenleistung den Rechendurchsatz der Steuerungsaufgabe reduziert. Demnach kann die Auslastung der Prozessorkerne vorteilhaft über die genannte zusätzliche Bedingung der Untergrenze für die Anpassung des Werts der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts der Programme berücksichtigt werden.

[0014] Des Weiteren kann der Rechendurchsatz der Steuerungsaufgabe mithilfe der Rechenzeit der parallelen Bearbeitungsabschnitte bewertet werden. Programme mit Teilaufgaben, die eine kurze Programmzyklus-Zeit aufweisen, beispielsweise 1 ms, können zwar durchaus in dem parallelen Bearbei-

tungsabschnitt des Programms auf den wenigstens einen Parallelprozessorkern zur parallelen Abarbeitung der Teilaufgaben verteilt werden. Dies kann gegebenenfalls aber dazu führen, dass der parallele Bearbeitungsabschnitt des Programms mit den jeweiligen Teilaufgaben für einen anderen parallelen Bearbeitungsabschnitt mit höherer Prioritätsstufe unterbrochen wird. In dem erläuterten Beispiel kann die Parallelisierung des Programms mit kurzer Programmzyklus-Zeit dazu führen, dass der Verwaltungsaufwand erhöht wird und die Rechenzeit für den parallelen Bearbeitungsabschnitt dadurch ansteigt. Die gesamte Ausführungszeit des Programms kann aufgrund der Parallelisierung langsamer sein, als für den Fall, in dem das Programm rein sequentiell und unterbrechungsfrei ausgeführt wird. Da die sequentielle Rechenzeit des Programms mit kurzer Programmzyklus-Zeit gegebenenfalls kürzer sein kann, als die Rechenzeit des parallelen Bearbeitungsabschnitts, ist es vorteilhaft, die jeweilige benötigte Rechenzeit für die Ausführung des Programms bei der Bewertung des Rechendurchsatzes (neben der zeitlichen Dauer und der Anzahl der Unterbrechungen der parallelen Bearbeitungsabschnitte sowie der Auslastung der Prozessorkerne) mit zu berücksichtigen, um den Rechendurchsatz der Steuerungsaufgabe zu verbessern.

[0015] Die dynamische Anpassung des Werts der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts der Programme erfolgt während der Programmlaufzeit, wenn der Rechendurchsatz der Steuerungsaufgabe dadurch erhöht werden kann. Ohne die dynamische Anpassung des Werts der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts in den entsprechenden Programmen, kann es sonst passieren, dass die gesamte Ausführungszeit der jeweiligen Programme aufgrund der Unterbrechungsdauer und des damit verbundenen zusätzlichen Verwaltungsaufwands steigt und die Programme der Steuerungsaufgabe insgesamt langsamer sind, als für den Fall, in dem die Aufgaben der Programme ohne Unterbrechung rein sequentiell in den zugeordneten Programmen abgearbeitet werden.

[0016] Neben der dynamischen Anpassung des Werts der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts der Programme während der Programmlaufzeit, ist es ferner denkbar, den bestmöglichen Wert der wenigstens einen Prioritätsstufe der parallelen Bearbeitungsabschnitte der Programme mithilfe einer Simulation vor dem Programmstart der Steuerungsaufgabe der SPS zu ermitteln. Der auf diese Weise erhaltene optimale Wert der wenigstens einen Prioritätsstufe der parallelen Bearbeitungsabschnitte der Programme kann als Startwert für die Programmausführung genutzt werden, wenn sich mehrere Programme der Steuerungsaufgabe der SPS einen Hauptprozessorkern teilen bzw.

auch für den Fall, dass auf einem Hauptprozessorkern jeweils nur ein Programm ausgeführt wird.

[0017] In einer weiteren Ausführungsform erfolgt die Anpassung der wenigstens einen Prioritätsstufe der parallelen Bearbeitungsabschnitte der jeweiligen Programme bei einer Unterbrechung der Abarbeitung der Teilaufgaben des Programms. Der Rechendurchsatz der Steuerungsaufgabe ist von der zeitlichen Unterbrechungsdauer der Abarbeitung der Teilaufgaben abhängig, da in dieser Zeit keine Teilaufgaben der parallelen Bearbeitungsabschnitte der Programme abgearbeitet werden, sondern Zwischenergebnisse der Berechnungen der Teilaufgaben gegebenenfalls übergeben und gesichert werden. Zudem kann sich jede Unterbrechung zeitlich unterscheiden. Die Unterbrechungsdauer kann folglich die Gesamtausführungszeit der Programme verlängern, da die zeitliche Unterbrechungsdauer zusätzlich zur Rechenzeit der Programme berücksichtigt werden muss. Es ist ferner ersichtlich, dass häufige Unterbrechungen der Abarbeitung der Teilaufgaben der parallelen Bearbeitungsabschnitte zu einer Verminderung des Rechendurchsatzes führen, da mit der Häufigkeit der Unterbrechungen der Berechnungen auch der Verwaltungsaufwand durch das vermehrte Übergeben und Sichern der Zwischenergebnisse der Berechnungen der Teilaufgaben steigt. Daher sollen die Anzahl und die Dauer der Unterbrechungen möglichst gering gehalten werden, um auf diese Weise einen hohen Rechendurchsatz der Steuerungsaufgabe erzielen zu können. Dies kann mithilfe der dynamischen Anpassung der wenigstens einen Prioritätsstufe der parallelen Bearbeitungsabschnitte der Programme bei der Unterbrechung der Abarbeitung der Teilaufgaben der parallelen Bearbeitungsabschnitte der Programme umgesetzt werden.

[0018] Eine weitere Ausführungsform sieht vor, dass die Anpassung der wenigstens einen Prioritätsstufe der parallelen Bearbeitungsabschnitte der jeweiligen Programme erfolgt, wenn die Teilaufgaben des Programms aufgrund von anderen Teilaufgaben eines weiteren Programms mit höherer Prioritätsstufe des parallelen Bearbeitungsabschnitts auf dem wenigstens einen Parallelprozessorkern unterbrochen werden. Werden die Teilaufgaben eines Arbeitspakets aufgrund von anderen Teilaufgaben eines weiteren Arbeitspakets, das mit einem höheren Wert der Prioritätsstufe des parallelen Bearbeitungsabschnitts im zugeordneten Programm verknüpft ist, auf dem wenigstens einen Parallelprozessorkern unterbrochen, so kann der Wert der wenigstens einen Prioritätsstufe der unterbrochenen Teilaufgaben direkt innerhalb des Programmzyklus verändert werden, um möglichst wenig Verwaltungsaufwand für die noch ausstehenden Teilaufgaben zu erzeugen.

[0019] Darüber hinaus, ist es ebenfalls möglich, den Wert der wenigstens einen Prioritätsstufe der un-

terbrochenen Teilaufgaben des parallelen Bearbeitungsabschnitts des jeweiligen Programms für den nächsten Programmzyklus entsprechend anzupassen, wie oben bereits beschrieben wurde. Auch kann mit der oben genannten, zusätzlichen Bedingung für den Startwert der wenigstens einen Prioritätsstufe, für den nächsten Programmzyklus sichergestellt werden, dass der Wert der wenigstens einen Prioritätsstufe nicht ungewollt unterhalb einer Untergrenze sinkt.

[0020] Der Rechendurchsatz der Steuerungsaufgabe ergibt sich aus der Dauer und der Anzahl der Unterbrechungen der Programme mit Aufgaben, also dem Verwaltungsaufwand. Der Rechendurchsatz kann erhöht werden, indem der Verwaltungsaufwand, das heißt die Anzahl der Unterbrechungen der Programme reduziert wird. Weiterhin ist es möglich den Rechendurchsatz zu erhöhen, in dem die Rechenzeit der jeweiligen Programme berücksichtigt wird. Kann ein Programm mit Teilaufgaben mit kurzer Programmzyklus-Zeit rein sequentiell schneller, das heißt mit kürzerer sequentieller Rechenzeit unterbrechungsfrei ausgeführt werden, als dies im parallelen Bearbeitungsabschnitt des Programms gegebenenfalls mit Unterbrechungen möglich ist, so kann der Rechendurchsatz der Steuerungsaufgabe durch eine Berücksichtigung der entsprechenden Rechenzeit im Programm ebenfalls erhöht werden. Auch kann der Rechendurchsatz der Steuerungsaufgabe erhöht werden, indem die Rechenleistung des wenigstens einen Parallelprozessorkerns vollständig ausgenutzt und dafür gesorgt wird, dass parallel abzuarbeitende Teilaufgaben des parallelen Bearbeitungsabschnitts des jeweiligen Programms von dem wenigstens einen Parallelprozessorkern abgearbeitet werden, wobei sequentiell auszuführende Programme auf dem jeweiligen Hauptprozessorkern ausgeführt werden. Ein erhöhter Rechendurchsatz kann ferner dazu beitragen, dass die Gesamtausführungszeit der Programme und Teilaufgaben reduziert wird, da die Steuerungsaufgabe auf diese Weise mehr Rechenzeit erhält.

[0021] In einer weiteren Ausführungsform wird bei der Anpassung der wenigstens einen Prioritätsstufe der parallelen Bearbeitungsabschnitte der jeweiligen Programme eine zusätzliche Bedingung, insbesondere eine Echtzeitbedingung der Steuerungsaufgabe, berücksichtigt. Die dynamische Anpassung des Werts der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts der jeweiligen Programme während der Programmlaufzeit kann weiterhin eine Echtzeitfähigkeit des Automatisierungssystems garantieren. Die Deadlines der Programme können durch die Anpassung des Werts der Prioritätsstufe in jedem Fall eingehalten werden. Für die Anpassung der wenigstens einen Prioritätsstufe der parallelen Bearbeitungsabschnitte der Programme kann neben der Echtzeitbedingung noch die oben genann-

te mögliche Bedingung der Untergrenze für den Wert der wenigstens einen Prioritätsstufe berücksichtigt werden.

[0022] In einer weiteren Ausführungsform wird die wenigstens eine Prioritätsstufe der parallelen Bearbeitungsabschnitte der jeweiligen Programme um einen Faktor reduziert, wenn die Teilaufgaben des Programms aufgrund von anderen Teilaufgaben eines weiteren Programms mit höherer Prioritätsstufe des parallelen Bearbeitungsabschnitts auf dem wenigstens einen Parallelprozessorkern unterbrochen werden. Eine Reduktion des Werts der wenigstens einen Prioritätsstufe der parallelen Bearbeitungsabschnitte der jeweiligen Programme um einen Faktor kann dazu beitragen, dass der Verwaltungsaufwand reduziert wird. Denn der Wert der wenigstens einen Prioritätsstufe des jeweiligen parallelen Bearbeitungsabschnitts im entsprechenden Programm, der mit den Teilaufgaben verbunden ist, die unterbrochen worden sind, kann zur Vermeidung einer weiteren Unterbrechung der parallelen Bearbeitungsabschnitte mit den Teilaufgaben direkt innerhalb des Programmzyklus herabgesetzt werden. Bei der Unterbrechung der Berechnungen der Teilaufgaben in den parallelen Bearbeitungsabschnitten der Programme kann es zudem vorkommen, dass aktuelle Zwischenergebnisse übergeben und Daten gesichert werden müssen. Häufige Unterbrechungen der Teilaufgaben können daher einen enormen Aufwand erzeugen, den es zu minimieren gilt. Die Anpassung des Werts der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts des jeweiligen Programms erfolgt stets unter der Bedingung, dass die Echtzeitfähigkeit des Automatisierungssystems gewährleistet werden kann, das heißt, dass die Verarbeitung der Eingabedaten spätestens zu dem Zeitpunkt beendet sein muss, zu dem die Ausgabedaten für die Akteure benötigt werden. Folglich muss garantiert werden können, dass die Deadlines der Programme trotz einer Reduktion des Werts der wenigstens einen Prioritätsstufe der parallelen Bearbeitungsabschnitte der entsprechenden Programme weiterhin eingehalten werden können.

[0023] Des Weiteren kann der Wert der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts des Programms der unterbrochenen Teilaufgaben unmittelbar nach der Unterbrechung der Abarbeitung der Teilaufgaben erhöht werden. Die Erhöhung des Werts der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts der Programme kann dazu beitragen, dass die zugeordneten Teilaufgaben nicht mehr von dem anderen parallelen Bearbeitungsabschnitt des weiteren Programms unterbrochen werden. Neben der dynamischen Anpassung des Werts der wenigstens einen Prioritätsstufe der unterbrochenen Teilaufgaben der parallelen Bearbeitungsabschnitte der Programme ist es weiterhin denkbar, den Wert der Prioritätsstufe des Programms

anzupassen, dass die Unterbrechung der Abarbeitung der Teilaufgaben des parallelen Bearbeitungsabschnitts des anderen Programms verursacht hat. So kann vermieden werden, dass die aktuell abgearbeiteten Teilaufgaben des parallelen Bearbeitungsabschnitts des einen Programms nicht erneut von dem anderen parallelen Bearbeitungsabschnitt des anderen Programms unterbrochen werden.

[0024] Alternativ zur Reduktion der Dauer und der Anzahl der Unterbrechungen des parallelen Bearbeitungsabschnitts in dem jeweiligen Programm durch die Anpassung des Werts der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts ist es ferner denkbar, die Rechenzeit des parallelen Bearbeitungsabschnitts des jeweiligen Programms bei der Bewertung des Rechendurchsatzes mit zu berücksichtigen. Das heißt es kann individuell entschieden werden, ob der Wert der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts des jeweiligen Programms angepasst und/oder die Teilaufgaben des Programms gegebenenfalls rein sequentiell abgearbeitet werden sollen, falls die sequentielle Rechenzeit für die Teilaufgaben des jeweiligen Programms kürzer ist, als die Rechenzeit des parallelen Bearbeitungsabschnitts des jeweiligen Programms. Dies gilt für die nachfolgenden Ausführungsformen in gleicher Weise, daher wird diese alternative Möglichkeit für die nachfolgenden Ausführungsformen nicht mehr wiederholt.

[0025] Eine weitere Ausführungsform sieht vor, dass die wenigstens eine Prioritätsstufe der parallelen Bearbeitungsabschnitte der jeweiligen Programme auf einen minimalen Wert gesetzt wird, wenn die Teilaufgaben der Programme sequentiell in den entsprechenden Programmen ausführbar sind. Im Extremfall ist es möglich, den Wert der wenigstens einen Prioritätsstufe der parallelen Bearbeitungsabschnitte der mehreren Programme der Steuerungsaufgabe auf einen minimalen Wert zu setzen. Insbesondere ist diese Ausgestaltung denkbar, wenn die Teilaufgaben keine zwingende parallele Ausführung erfordern, sondern ebenfalls sequentiell in dem zugeordneten Programm abgearbeitet werden können.

[0026] In einer weiteren Ausführungsform wird die wenigstens eine Prioritätsstufe der parallelen Bearbeitungsabschnitte der jeweiligen Programme zu Beginn eines neuen Programmzyklus auf den vorgegebenen Wert zurückgesetzt. Von Vorteil bei der Anpassung des Werts der Prioritätsstufe der parallelen Bearbeitungsabschnitte der jeweiligen Programme ist, dass diese vorübergehend erfolgen kann. Die dynamische Anpassung des Werts der wenigstens einen Prioritätsstufe kann direkt innerhalb der Programmlaufzeit erfolgen und muss nicht für die gesamte Laufzeit des Programms in gleicher Weise stattfinden. Der Wert der Prioritätsstufe kann beispielsweise für einen neuen Programmzyklus wieder auf den vorge-

gebenen Wert zurückgesetzt werden und kann während des Programmzyklus und/oder für einen weiteren Programmzyklus erneut abgeändert werden.

[0027] Eine weitere Ausführungsform sieht vor, dass die wenigstens eine Prioritätsstufe der parallelen Bearbeitungsabschnitte der jeweiligen Programme zu Beginn eines neuen Programmzyklus erhöht wird, wenn die Teilaufgaben vorher sequentiell in den zugeordneten Programmen ausgeführt wurden. Statt der Reduktion des Werts der wenigstens einen Prioritätsstufe der parallelen Bearbeitungsabschnitte der jeweiligen Programme kann der Wert der wenigstens einen Prioritätsstufe auch erhöht werden. Beispielsweise kann die Anpassung des Werts der wenigstens einen Prioritätsstufe zu Beginn des neuen Programmzyklus erfolgen. Diese Ausgestaltung ist denkbar, wenn die Teilaufgaben mit entsprechender Prioritätsstufe in einem vorherigen Programmzyklus rein sequentiell in dem zugeordneten Programm abgearbeitet wurden. Mithilfe dieser Ausgestaltung kann vermieden werden, dass zu viele Programme der Steuerungsaufgabe irgendwann einen minimalen Wert der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts der Programme haben und die Programme rein sequentiell ausgeführt werden. Werden die Programme rein sequentiell auf den Hauptprozessorkernen ausgeführt, kann dies gegebenenfalls zur Folge haben, dass die verfügbare Rechenleistung der Parallelprozessorkerne nicht mehr vollständig ausgenutzt wird, was dann in einem verminderten gesamten Rechendurchsatz der Steuerungsaufgabe resultiert.

[0028] Erfindungsgemäß wird des Weiteren eine speicherprogrammierbare Steuerung (SPS) vorgeschlagen. Die SPS umfasst eine Kommunikationsschnittstelle, zum Einlesen von Sensordaten und zum Ausgeben von Aktordaten. Ferner umfasst die SPS eine Datenverarbeitungseinrichtung, die wenigstens einen Parallelprozessorkern zum Ausführen einer Steuerungsaufgabe umfasst, um aus den Sensordaten Aktordaten zu erzeugen, sowie eine Steuerungseinrichtung, die einen Datenstrukturspeicher zum Speichern einer Datenstruktur mit Prioritätsstufen und einen Prioritätenverwalter zum Verwalten der Prioritätsstufen in der Datenstruktur aufweist. Die Steuerungseinrichtung umfasst einen Zwischenergebnisspeicher, auf dem Zwischenergebnisse von Berechnungen gesichert werden. Die Steuerungsaufgabe umfasst weiterhin mehrere ausführbare Programme, wobei wenigstens zwei Programme der Steuerungsaufgabe jeweils wenigstens einen parallelen Bearbeitungsabschnitt mit einem Arbeitspaket aufweisen, und das Arbeitspaket mehrere Teilaufgaben umfasst. Den parallelen Bearbeitungsabschnitten in den jeweiligen Programmen ist eine Priorität mit der vorgegebenen Prioritätsstufe zugewiesen. Der Prioritätenverwalter ist ausgebildet, die jeweiligen Prioritätsstufen in die Datenstruktur im Da-

tenstrukturspeicher einzufügen, sobald ein Ausführen des Programms an dem parallelen Bearbeitungsabschnitt angelangt ist und dabei den Eintrag mit der höchsten Prioritätsstufe an die erste Stelle der Einträge in der Datenstruktur zu setzen. Der wenigstens eine Parallelprozessorkern ist ausgebildet, die Datenstruktur im Datenstrukturspeicher auf Einträge der Prioritätsstufen zu prüfen, und sofern Einträge vorhanden sind, Teilaufgaben aus dem Arbeitspaket des entsprechenden Programms abzuarbeiten, dessen Prioritätsstufe an erster Stelle der Einträge in der Datenstruktur steht. Die Steuerungseinrichtung ist ausgebildet, während eines Programmzyklus einen zu erwartenden Rechendurchsatz der Steuerungsaufgabe laufend zu bewerten. Ferner ist der Prioritätenverwalter ausgelegt, wenigstens eine der Prioritätsstufen der parallelen Bearbeitungsabschnitte der jeweiligen Programme dynamisch anzupassen, wenn sich der Rechendurchsatz der Steuerungsaufgabe dadurch erhöht.

[0029] Speicherprogrammierbare Steuerungen bilden häufig die Grundlage eines Automatisierungssystems 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 Prozessorkernen basieren, um das System effizient zu betreiben und die vorhandenen Ressourcen bestmöglich auszulasten. Die so konzipierte SPS ermöglicht die Deadlines der Programme einzuhalten und kann ferner die Echtzeitdatenverarbeitung gewähren. Das Verfahren zur Optimierung der Datenverarbeitung auf einer SPS kann eine solche effiziente Software für die SPS darstellen, um die Rechenleistung der vorhandenen Prozessorkerne bestmöglich auszunutzen, in dem die Programme der Steuerungsaufgabe auf mehrere Prozessorkerne verteilt ausgeführt werden können. Werden die Programme parallelisiert auf dem wenigstens einen Parallelprozessorkern ausgeführt, so kann es aufgrund der den jeweiligen parallelen Bearbeitungsabschnitten der Programme zugeordneten Prioritätsstufen vorkommen, dass die Programme von anderen Programmen mit höherer Prioritätsstufe des parallelen Bearbeitungsabschnitts unterbrochen werden. Unterbrechungen der Programme der Steuerungsaufgabe der SPS sind in der Regel nicht gewünscht, da sie den Rechendurchsatz der Steuerungsaufgabe reduzieren. Sie verlängern zudem die Gesamtausführungszeit der Programme, da sich die zeitliche Unterbrechungsdauer der Programme additiv auf die reine Rechenzeit der Programme auswirkt. Insofern ist es günstig, den Wert der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts des Programms vom Prioritätenverwalter der Steuerungseinrichtung der SPS dynamisch während des Programmzyklus bei einer Unterbrechung des Programms mit parallelem Bearbeitungsabschnitts anpassen zu können. Dabei dient

die Steuerungseinrichtung der SPS zur Ermittlung und Bewertung des zu erwartenden Rechendurchsatzes der Steuerungsaufgabe. Die Steuerungseinrichtung kann den Prioritätenverwalter der SPS zudem zur Anpassung des Werts der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts des jeweiligen Programms anstoßen, wenn sich der Rechendurchsatz durch die mögliche Anpassung des Werts der wenigstens einen Prioritätsstufe der parallelen Bearbeitungsabschnitte der Programme auf diese Weise erhöht. Gegebenenfalls kann die Steuerungseinrichtung der SPS bei einer kurzen Programmzyklus-Zeit entscheiden, dass die Teilaufgaben des Programms rein sequentiell ausgeführt werden sollen, statt in dem parallelen Bearbeitungsabschnitt des Programms, der unter Umständen mit Unterbrechungen der Berechnungen der Teilaufgaben verbunden sein kann. Es kann andernfalls möglich sein, dass die gesamte Ausführungszeit des Programms durch die Parallelisierung langsamer ist, als für den Fall, in dem die Teilaufgaben des Programms ohne Unterbrechung sequentiell abgearbeitet werden.

[0030] Zudem kann die Steuerungseinrichtung bei der Bewertung des Rechendurchsatzes die Ausnutzung der Rechenleistung des wenigstens einen Parallelprozessorkerns mit berücksichtigen und dafür sorgen, dass parallel auszuführende Programme, die nicht sequentiell auf dem jeweiligen Hauptprozessorkern ausgeführt werden sollen, von dem wenigstens einen Parallelprozessorkern ausgeführt werden. Die Steuerungseinrichtung der SPS teilt dem Prioritätenverwalter dazu mit, dass er den Wert der wenigstens einen Prioritätsstufe der parallelen Bearbeitungsabschnitte der Programme dahingehend anpassen soll, beispielsweise erhöhen falls er vorher minimal war.

[0031] Die speicherprogrammierbare Steuerung mit dem Verfahren zum Optimieren einer Datenverarbeitung ist flexibel und universell für die verschiedensten Aufgaben des Automatisierungssystems einsetzbar. Beispielsweise kann die SPS aus den Sensordaten Daten erzeugen, die für die Qualitätssicherung/-kontrolle des Automatisierungssystems relevant sein können. Ist der Sensor zum Beispiel in Form einer Kamera verwirklicht, so kann mithilfe dieser Kamera die Qualität von hergestellten Bauteilen bewertet werden. In diesem Fall erzeugt die SPS dann keine Aktordaten als Ergebnisse, sondern bewertet die Qualität der Bauteile beispielsweise als fehlerhaft oder als gut. Ferner kann die SPS mit dem Verfahren zum Optimieren einer Datenverarbeitung in einem Transportsystem umgesetzt werden, in dem es erforderlich ist, Transportelemente auf modularen Führungsschienen mit Motormodulen ansteuern und positionieren zu können. Da die Transportelemente individuell positioniert werden können, ist es möglich die Berechnungen für die Transportelemente zu parallelisieren und auf mehrere Prozessorkerne verteilt aus-

zuführen. Die SPS mit dem oben genannten Verfahren kann dabei zu einer effizienten Nutzung der Rechenleistung der Prozessorkerne beitragen, mithilfe der Berechnungen eine hohe Zuverlässigkeit für die Positionierung der Transportelemente bereitstellen und verursacht, durch die flexible Nutzungsmöglichkeit der SPS, geringe Kosten in der Umsetzung.

[0032] In einer weiteren Ausführungsform ist der Prioritätenverwalter ausgebildet, die Anpassung der wenigstens einen Prioritätsstufe der parallelen Bearbeitungsabschnitte der jeweiligen Programme bei einer Unterbrechung der Abarbeitung der Teilaufgaben des Programms vorzunehmen. Der Prioritätenverwalter kann den Wert der wenigstens einen Prioritätsstufe direkt bei der Unterbrechung der Abarbeitung der Teilaufgaben des Programms innerhalb des Programmzyklus anpassen, um eine weitere Unterbrechung des Programms für die noch ausstehenden Teilaufgaben zu vermeiden. Gleichermaßen ist der Prioritätenverwalter in der Lage den Wert der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts des Programms für den nächsten Programmzyklus anzupassen.

[0033] In einer weiteren Ausführungsform ist der Prioritätenverwalter ausgelegt, die Anpassung der wenigstens einen Prioritätsstufe der parallelen Bearbeitungsabschnitte der jeweiligen Programme vorzunehmen, wenn die Teilaufgaben aufgrund von anderen Teilaufgaben eines weiteren Programms mit höherer Prioritätsstufe auf dem wenigstens einen Parallelprozessorkern unterbrochen werden. Der Prioritätenverwalter kann den Wert der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts des entsprechenden Programms für die unterbrochenen Teilaufgaben auf dem wenigstens einen Parallelprozessorkern anpassen, um einer weiteren Unterbrechung der Teilaufgaben auf dem wenigstens einen Parallelprozessorkern vorzubeugen und in diesem Zusammenhang für möglichst wenig Verwaltungsaufwand für die noch ausstehenden Teilaufgaben zu sorgen. Denn bei jeder Unterbrechung der Teilaufgaben müssen die vom wenigstens einen Parallelprozessorkern berechneten Zwischenergebnisse übergeben und auf dem Zwischenergebnisspeicher gesichert werden. Bei jeder Unterbrechung sortiert der Prioritätenverwalter zudem die Einträge der Prioritätsstufen in der Datenstruktur neu. Bei jeder Wiederaufnahme der unterbrochenen Teilaufgaben muss der wenigstens eine Parallelprozessorkern erneut auf den Zwischenergebnisspeicher zugreifen und die gesicherten Zwischenergebnisse als Ausgangspunkt für die weitere Abarbeitung der Teilaufgaben nutzen. Aus diesem Grund sollen die Teilaufgaben möglichst selten unterbrochen werden, um einen höheren Rechendurchsatz der Steuerungsaufgabe zu erzeugen und die gesamte Ausführungszeit der Programme und Teilaufgaben auf diese Weise zu reduzieren.

[0034] Eine weitere Ausführungsform sieht vor, dass der Prioritätenverwalter ausgebildet ist, bei der Anpassung der wenigstens einen Prioritätsstufe der parallelen Bearbeitungsabschnitte der jeweiligen Programme eine zusätzliche Bedingung, insbesondere eine Echtzeitbedingung der Steuerungsaufgabe, zu berücksichtigen. Der Prioritätenverwalter gestaltet die Koordination sowie die dynamische Anpassung der Werte der Einträge der Prioritätsstufen in der Datenstruktur und die Kommunikation mit dem wenigstens einen Parallelprozessorkern einfach. Ferner kann mithilfe des Prioritätenverwalters die Echtzeitfähigkeit des Automatisierungssystems gewährleistet werden, da der Prioritätenverwalter die Anpassung der Werte der Prioritätsstufen der jeweiligen parallelen Bearbeitungsabschnitte der entsprechenden Programme in der Weise vornimmt, dass die Deadlines der Programme in jedem Fall eingehalten werden. Der Prioritätenverwalter passt den Wert von wenigstens einer Prioritätsstufe an, wenn sich der Rechendurchsatz der Steuerungsaufgabe dadurch erhöht. Die Steuerungseinrichtung kann in diesem Zusammenhang den Rechendurchsatz der Steuerungsaufgabe während eines Programmzyklus laufend bewerten und den Prioritätenverwalter anhand der Bewertung des Rechendurchsatzes zur Anpassung des Werts der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts im jeweiligen Programm anstoßen.

[0035] Eine weitere Ausführungsform sieht vor, dass der Prioritätenverwalter ausgebildet ist, den Wert der wenigstens einen Prioritätsstufe der parallelen Bearbeitungsabschnitte der jeweiligen Programme um einen Faktor zu reduzieren, wenn die Teilaufgaben des Programms aufgrund von anderen Teilaufgaben eines weiteren Programms mit höherer Prioritätsstufe auf dem wenigstens einen Parallelprozessorkern unterbrochen werden. Der Prioritätenverwalter kann dazu ausgelegt sein, den Wert der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts des jeweiligen Programms um einen Faktor zu reduzieren. Beispielsweise kann der Wert der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts des jeweiligen Programms bei jeder Unterbrechung der Teilaufgaben des entsprechenden Programms direkt während des Programmzyklus um die Hälfte reduziert werden. Alternativ dazu ist ebenso eine Anpassung des Werts der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts des jeweiligen Programms um einen geringeren Faktor denkbar. Ferner kann die Anpassung des Werts der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts des entsprechenden Programms für jeden Programmzyklus auf andere Weise erfolgen. Beispielsweise muss der angepasste Wert der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts des jeweiligen Programms nicht für jeden Programmzyklus beibehalten werden. Der Prioritätenverwalter kann die An-

passung des Werts der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts des entsprechenden Programms nach der Bewertung des Rechendurchsatzes der durch die Steuerungseinrichtung erfolgt, vornehmen. Da die Steuerungseinrichtung den Rechendurchsatz während eines Programmzyklus laufend bewertet, kann der Prioritätenverwalter den Wert der Prioritätsstufe innerhalb eines jeden Programmzyklus und für jeden neuen Programmzyklus dynamisch anpassen.

[0036] In einer weiteren Ausführungsform ist der Prioritätenverwalter ausgelegt, die wenigstens eine Prioritätsstufe der parallelen Bearbeitungsabschnitte der jeweiligen Programme auf einen minimalen Wert zu setzen, wenn die Teilaufgaben der Programme sequentiell in den entsprechenden Programmen ausführbar sind. Der Prioritätenverwalter ist in der Lage den Wert der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts des entsprechenden Programms auf einen minimalen Wert zu setzen. Setzt der Prioritätenverwalter den Wert der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts des jeweiligen Programms auf den geringsten möglichen Wert, so werden die zugehörigen Teilaufgaben in dem zugeordneten Programm sequentiell abgearbeitet. Denn der minimale Wert der Prioritätsstufe bedeutet, dass die Teilaufgaben keine zwingende parallele Abarbeitung erfordern um die Deadlines der Programme einhalten zu können, selbst dann nicht, wenn der wenigstens eine Parallelprozessorkern gerade keine anderen Teilaufgaben abarbeitet, sondern die Teilaufgaben können ebenso sequentiell vom zugeordneten Programm abgearbeitet werden. Diese Ausführungsform ist ebenfalls denkbar bei einer häufigen Unterbrechung der Berechnungen der Teilaufgaben in den parallelen Bearbeitungsabschnitten, da der Prioritätenverwalter den Wert der Prioritätsstufe des parallelen Bearbeitungsabschnitts des Programms gleich soweit herabsetzen kann, dass die Teilaufgaben nach der Unterbrechung der Berechnungen auf dem wenigstens einen Parallelprozessorkern anschließend in dem zugeordneten Programm weiter abgearbeitet werden. Auf diese Weise ist es ferner möglich, den Verwaltungsaufwand zu reduzieren.

[0037] In der oben dargestellten Ausführungsform wird der Wert der wenigstens einen Prioritätsstufe der parallelen Bearbeitungsabschnitte der Programme vom Prioritätenverwalter gezielt auf den minimalen Wert gesetzt, damit die Teilaufgaben der Programme sequentiell im zugeordneten Programm abgearbeitet werden, da sie keine zwingende Parallelisierung benötigen. Es kann jedoch möglich sein, dass der Wert der wenigstens einen Prioritätsstufe der parallelen Bearbeitungsabschnitte der Programme direkt nach der Unterbrechung der Abarbeitung der Teilaufgaben der parallelen Bearbeitungsabschnitte der Programme während des Programmzyklus vom Prio-

ritätenverwalter reduziert wurde. Unabhängig davon kann in einem solchen Fall eine weitere Reduktion des Werts der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts der Programme für einen neuen Programmzyklus von Vorteil sein, wenn die Abarbeitung der Teilaufgaben der parallelen Bearbeitungsabschnitte der Programme im vorherigen Programmzyklus unterbrochen wurde. Damit der Wert der wenigstens einen Prioritätsstufe jedoch nicht für jeden Programmzyklus verringert wird, kann über eine zusätzliche Bedingung, die zum Beispiel in der Form implementiert sein kann, dass der Wert der wenigstens einen Prioritätsstufe für den neuen Programmzyklus mindestens dem halben Wert der vorgegebenen wenigstens einen Prioritätsstufe entspricht, sichergestellt werden, dass der Wert der wenigstens einen Prioritätsstufe der Programme nicht ungewollt auf den minimalen Wert gesetzt wird.

[0038] Eine weitere Ausführungsform sieht vor, dass der Prioritätenverwalter ausgebildet ist, die wenigstens eine Prioritätsstufe der parallelen Bearbeitungsabschnitte der jeweiligen Programme zu Beginn eines neuen Programmzyklus auf den vorgegebenen Wert zurückzusetzen. Der Prioritätenverwalter kann, wie bereits oben erläutert, den Wert der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts des entsprechenden Programms dynamisch mit jedem Programmzyklus anpassen. In diesem Zusammenhang ist es demnach möglich, dass der Prioritätenverwalter den Wert der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts des jeweiligen Programms für den neuen Programmzyklus wieder auf den vorgegebenen Wert der Prioritätsstufe zurücksetzt. Die Anpassung des Werts der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts des jeweiligen Programms durch den Prioritätenverwalter hat den Vorteil, dass sie vorübergehend erfolgen und wieder rückgängig gemacht werden kann.

[0039] In einer weiteren Ausführungsform ist der Prioritätenverwalter ausgelegt, die wenigstens eine Prioritätsstufe der parallelen Bearbeitungsabschnitte der jeweiligen Programme zu Beginn eines neuen Programmzyklus zu erhöhen, wenn die Teilaufgaben der Programme vorher sequentiell in den zugeordneten Programmen ausgeführt wurden. Der Prioritätenverwalter ist ebenso in der Lage den Wert der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts des jeweiligen Programms der Steuerungsaufgabe zu erhöhen, damit die zugehörigen Teilaufgaben vor anderen Teilaufgaben eines anderen Arbeitspakets und Programms auf dem wenigstens einen Parallelprozessorkern abgearbeitet werden können. Diese Ausführungsform ist denkbar, wenn die Teilaufgaben vorher im zugeordneten Programm sequentiell abgearbeitet worden sind. Mithilfe dieser Ausgestaltung kann vermieden werden, dass zu viele Programme der Steuerungsaufgabe

be irgendwann einen minimalen Wert der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts der Programme haben und die Programme rein sequentiell ausgeführt werden. Werden die Programme rein sequentiell auf den Hauptprozessorkernen ausgeführt, kann dies gegebenenfalls zur Folge haben, dass die verfügbare Rechenleistung der Parallelprozessorkerne nicht mehr vollständig ausgenutzt wird, was dann in einem verminderten gesamten Rechendurchsatz der Steuerungsaufgabe resultiert.

[0040] Eine weitere Ausführungsform sieht vor, dass der Prioritätenverwalter ausgebildet ist, den Eintrag der Prioritätsstufe in der Datenstruktur im Datenstrukturspeicher gemäß der vorgenommenen Anpassung der wenigstens einen Prioritätsstufe abzuändern. Ferner ist der Prioritätenverwalter ausgebildet, den wenigstens einen Parallelprozessorkern über die Änderung des Eintrags der Prioritätsstufe in der Datenstruktur zu informieren. Der Prioritätenverwalter übernimmt einerseits die dynamische Anpassung des Werts der wenigstens einen Prioritätsstufe des entsprechenden parallelen Bearbeitungsabschnitts des jeweiligen Programms und kann andererseits weiterhin den wenigstens einen Parallelprozessorkern über die Anpassung des Werts der wenigstens einen Prioritätsstufe in der Datenstruktur informieren.

[0041] Eine weitere Ausführungsform sieht vor, dass der Prioritätenverwalter ausgebildet ist, auf mehreren Programmen der Steuerungsaufgabe verteilt ausführbar zu sein. Der Prioritätenverwalter, der Bestandteil der Steuerungseinrichtung ist, kann beispielsweise mehrere Module (beispielsweise Software-Module) umfassen, wobei die einzelnen Module (beispielsweise Software-Module) des Prioritätenverwalters in den mehreren Programmen der Steuerungsaufgabe ausgeführt werden können. Denkbar ist weiterhin, dass der Prioritätenverwalter nur ein einziges zentrales Modul (beispielsweise ein Software-Modul) aufweist, das in einem der mehreren Programme der Steuerungsaufgabe ausgeführt wird. Zudem ist denkbar, den Prioritätenverwalter als eigenständiges Programm zu verwirklichen oder als ein oder mehrere externe Hardware-Module.

[0042] 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.

[0043] 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 zur Erstellung von Arbeitspaketen mit Teilaufgaben und Verwaltung von Prioritätsstufen innerhalb der SPS nach **Fig. 1**;

Fig. 3 ein Schema eines Verfahrens zum Optimieren einer Datenverarbeitung auf der SPS nach den **Fig. 1** und **Fig. 2**; und

Fig. 4 ein Ablaufdiagramm zur dynamischen Anpassung einer Prioritätsstufe eines parallelen Bearbeitungsabschnitts eines Programms nach **Fig. 3**.

[0044] Anhand der folgenden Figuren wird ein Ausführungsbeispiel eines Verfahrens zum Optimieren einer 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.

[0045] 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.

[0046] Meist setzt sich die Steuerungsaufgabe der SPS aus einer oder mehreren zyklisch auszuführenden Programmen mit Aufgaben, den „Tasks“ zusam-

men. Da viele SPS Prozessoren mit mehreren Prozessorkernen, 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 der Deadline und/oder den Abhängigkeiten der jeweiligen Programme zu den anderen Programmen ermittelt werden, wenn die Programme beispielsweise auf die Ergebnisse der anderen Programme angewiesen sind.

[0047] 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 Datenstruktur 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.

[0048] Während eines Programmzyklus wird ein zu erwartender Rechendurchsatz der Steuerungsaufgabe laufend bewertet. Anhand dieser Bewertung kann entschieden werden, dass wenigstens eine der Prioritätsstufen des parallelen Bearbeitungsabschnitts des jeweiligen Programms der Steuerungsaufgabe dynamisch angepasst wird, wenn sich der Rechendurchsatz der Steuerungsaufgabe durch die Anpassung des Werts der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts erhöht. Der zu er-

wartende Rechendurchsatz der Steuerungsaufgabe ist von einer Anzahl an Unterbrechungen der parallelen Bearbeitungsabschnitte der Programme abhängig, da die Anzahl der Unterbrechungen der parallelen Bearbeitungsabschnitte der Programme in einem Verwaltungsaufwand resultiert. Der Verwaltungsaufwand beschreibt die Aktivitäten, die beim Unterbrechen der Abarbeitung der Teilaufgaben in den parallelen Bearbeitungsabschnitten in den Programmen der Steuerungsaufgabe erforderlich sind, wie zum Beispiel das Übergeben und Sichern von Zwischenergebnissen. Häufige Unterbrechungen der Abarbeitung der parallelen Bearbeitungsabschnitte führen demnach zu einem höheren Verwaltungsaufwand. Auch die zeitliche Unterbrechungsdauer der Programme ist für den Rechendurchsatz relevant, da während der zeitlichen Dauer der Unterbrechung der Abarbeitung der Teilaufgaben der parallelen Bearbeitungsabschnitte der Programme keine Berechnungen durchgeführt, sondern die oben beschriebenen Aktivitäten ausgeführt werden. Die zeitliche Unterbrechungsdauer verlängert demnach die Ausführungszeit des Programms, da sie additiv auf die eigentliche Rechenzeit des Programms wirkt und diese ist in der Regel vorgegeben. Zudem kann die Unterbrechungsdauer bei der Unterbrechung der Berechnungen der Programme mit parallelen Bearbeitungsabschnitten zeitlich variieren. Das Ziel ist also den Verwaltungsaufwand möglichst gering zu halten und den Wert der wenigstens einen Prioritätsstufe dynamisch anzupassen, um die Häufigkeit der Unterbrechungen der Berechnungen der Programme zu senken und den Rechendurchsatz damit zu erhöhen.

[0049] Ferner ist der zu erwartende Rechendurchsatz der Steuerungsaufgabe von der genutzten Rechenleistung des wenigstens einen Parallelprozessorkerns abhängig. Werden die Programme rein sequentiell auf den entsprechenden Hauptprozessorkernen ausgeführt, zum Beispiel in dem sie den minimalen Wert der wenigstens einen Prioritätsstufe der parallelen Bearbeitungsabschnitte der Programme zugewiesen bekamen, so kann es passieren, dass der wenigstens eine Parallelprozessorkern gerade keine aktiven Berechnungen durchführt oder nur eingeschränkt aktiv ist. Dies vermindert den Rechendurchsatz der Steuerungsaufgabe, da die Rechenleistung des wenigstens einen Parallelprozessorkerns nicht oder nicht vollständig ausgenutzt wird. In einem solchen Fall kann der Wert der wenigstens einen Prioritätsstufe der parallelen Bearbeitungsabschnitte der Programme angepasst, zum Beispiel erhöht werden, damit die Programme parallel auf dem wenigstens einen Parallelprozessorkern ausgeführt werden und die Rechenleistung der vorhandenen Prozessorkerne bestmöglich genutzt wird.

[0050] Des Weiteren kann bei der Bewertung des zu erwartenden Rechendurchsatzes der Steuerungsaufgabe die Rechenzeit der parallelen Bearbeitungs-

abschnitte der Programme mit berücksichtigt werden. Denn Programme mit Teilaufgaben mit kurzer Programmzyklus-Zeit können gegebenenfalls ohne Parallelisierung schneller rein sequentiell ausgeführt werden, als im jeweiligen parallelen Bearbeitungsabschnitt im Programm. Der parallele Bearbeitungsabschnitt des jeweiligen Programms kann unterbrochen werden für einen anderen parallelen Bearbeitungsabschnitt eines weiteren Programms mit höherer zugeordneter Prioritätsstufe. Da die Unterbrechungen des parallelen Bearbeitungsabschnitts beliebig häufig durch andere Programme mit parallelen Bearbeitungsabschnitten ausgelöst werden können, kann es unter Umständen vorkommen, dass die gesamte Ausführungszeit des Programms, das durch andere parallele Bearbeitungsabschnitte weiterer Programme unterbrochen wird, länger ist, als ohne Parallelisierung. In einem solchen Fall ist es also denkbar, das Programm mit Teilaufgaben rein sequentiell auszuführen, um auf diese Weise eine kürzere Ausführungszeit des entsprechenden Programms zu erzielen und den Rechendurchsatz zu erhöhen.

[0051] Der Kern der Erfindung liegt in der Möglichkeit der dynamischen Anpassung des Werts der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts des jeweiligen Programms der Steuerungsaufgabe. Denn durch die dynamische Anpassung des Werts der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts des entsprechenden Programms kann flexibel entschieden werden, ob die Teilaufgaben des jeweiligen Programms sequentiell auf dem zugeordneten Hauptprozessorkern oder im jeweiligen parallelen Bearbeitungsabschnitt des Programms zusätzlich auf dem wenigstens einen Parallelprozessorkern abgearbeitet werden sollen. Auf diese Weise kann eine optimale Ausnutzung der Ressourcen bereitgestellt werden, indem die Hauptprozessorkerne und die Parallelprozessorkerne gemeinsam Teilaufgaben des jeweiligen Programms abarbeiten. Die Echtzeitfähigkeit der Steuerungsaufgabe kann durch die Anpassung des Werts der Prioritätsstufe des parallelen Bearbeitungsabschnitts des entsprechenden Programms der Steuerungsaufgabe weiterhin garantiert werden, indem die Anpassung des Werts der Prioritätsstufe stets so erfolgt, dass die Echtzeitbedingung der Steuerungsaufgabe und die Deadlines der Programme eingehalten werden können.

[0052] 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 beispielweise 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 Maschi-

ne bzw. der Anlage **200** des Automatisierungssystems steuern, wobei die Aktoren **210** und Sensoren **220** mithilfe des Feldbussystems miteinander vernetzt sein können.

[0053] 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.

[0054] 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.

[0055] Die Datenverarbeitungseinrichtung **110** kann mehrere Prozessorkerne umfassen, 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.

[0056] Die mehreren ausführbaren Programme der Steuerungsaufgabe, von denen wenigstens zwei Programme einen parallelen Bearbeitungsabschnitt aufweisen, können beispielsweise auf dem ersten und zweiten Hauptprozessorkern **111**, **112** ausgeführt werden, wobei der erste Parallelprozessorkern **113** und der zweite Parallelprozessorkern **114** wie oben genannt, unterstützend bei der Abarbeitung der Teilaufgaben in den jeweiligen parallelen Abschnitten der Programme der Steuerungsaufgabe 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.

[0057] 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.

[0058] Die Teilaufgaben in den Programmen, die eine Parallelisierung erfordern und von mehreren Prozessorkernen abgearbeitet werden, können jeweils wenigstens 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 und zur Vornahme der Anpassung des Werts der jeweiligen Prioritätsstufe des zugeordneten parallelen Bearbeitungsabschnitts des Programms. Die Datenstruktur kann des Weiteren Methoden aufweisen, welche von den Programmen auf den beiden Hauptprozessorkernen **111**, **112** und den beiden Parallelprozessorkernen **113**, **114** aufgerufen werden können. Wenn der erste Hauptprozessorkern **111** beispielsweise an einen parallelen Be-

arbeitsabschnitt im Programm angelangt, so ruft er eine „Füge meine Teilaufgaben hinzu“-Methode von der Datenstruktur des Prioritätenverwalters **350** auf. In der Methode können die Teilaufgaben dann in die Datenstruktur eingeordnet werden und der Prioritätenverwalter **350** kann dazu ausgelegt sein, den ersten und zweiten Parallelprozessorkern **113**, **114** über einen neuen, höheren Eintrag oder eine geänderte Reihenfolge der Einträge der Prioritätsstufen in der Datenstruktur zu informieren, falls die Teilaufgaben der Methode mit einer höheren Prioritätsstufe verknüpft sind, wobei die Methode auf dem ersten Hauptprozessorkern **111** ausgeführt wird. Die beiden Parallelprozessorkerne unterbrechen daraufhin ihre Abarbeitung der aktuellen Teilaufgaben, organisieren die Zwischenergebnisse und rufen dann eine „Sichere meine Zwischenergebnisse“-Methode aus der Datenstruktur des Prioritätenverwalters **350** auf. In dieser Methode kann dann bei Bedarf auch die Prioritätsstufe der zugehörigen Teilaufgaben in der Datenstruktur angepasst werden. Das heißt diese Methode wird jeweils auf dem ersten und zweiten Parallelprozessorkern **113**, **114** ausgeführt.

[0059] Es ist im Allgemeinen nicht notwendig, dass der Prioritätenverwalter **350** den ersten und zweiten Hauptprozessorkern **111**, **112** bzw. das auf den beiden Hauptprozessorkernen **111**, **112** auszuführende Programm über eine Änderung des Werts der wenigstens einen Prioritätsstufe informiert. Denn der erste und zweite Hauptprozessorkern **111**, **112** arbeiten die ihm zugehörigen Teilaufgaben stets ab, auch wenn sie nicht an erster Stelle der Einträge der Prioritätsstufen, die mit den Teilaufgaben verknüpft sind, in der Datenstruktur stehen. Das heißt, die beiden Hauptprozessorkerne **111**, **112** arbeiten die ihnen zugehörigen Teilaufgaben unabhängig von der Unterstützung der beiden Parallelprozessorkerne **113**, **114** ab. Hingegen stellt die rein sequentielle Abarbeitung der Teilaufgaben auf dem ersten und zweiten Hauptprozessorkern **111**, **112** einen Unterschied für den ersten und zweiten Parallelprozessorkern **113**, **114** dar. Denn wenn der erste und zweite Parallelprozessorkern **113**, **114** eine „Gib mir eine neue Teilaufgabe“-Methode aus der Datenstruktur des Prioritätenverwalters **350** aufrufen und nur noch Teilaufgaben, die mit dem minimalen Wert der Prioritätsstufe der parallelen Bearbeitungsabschnitte der Programme verknüpft sind, vorhanden sind und folglich für die rein sequentielle Abarbeitung auf dem ersten und zweiten Hauptprozessorkern **111**, **112** vorgesehen sind, so erhalten die beiden Parallelprozessorkerne **113**, **114** „Es sind keine Teilaufgaben verfügbar“ als Antwort. Der Einfachheit halber wird für die Beschreibung im Folgenden auf die einzelnen Methoden der Datenstruktur des Prioritätenverwalters **350** verzichtet. Gleichwohl können die oben genannten Methoden sowie weitere Methoden der Datenstruktur für die Implementierung des erfindungsgemäßen

Verfahrens in den nachfolgenden Figuren eingesetzt werden.

[0060] Der Prioritätenverwalter **350** kann auf mehreren Programmen in Form von mehreren Modulen (beispielsweise Software-Module) verteilt ausgeführt werden. Ebenfalls denkbar ist es, den Prioritätenverwalter **350** als eigenständiges Modul (beispielsweise als Software-Modul) in einem Programm 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. Zudem ist denkbar, den Prioritätenverwalter **350** als eigenständiges Programm zu verwirklichen oder als ein oder mehrere externe Hardware-Module.

[0061] 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 Hauptprozessorkern **112** bzw. dem ersten Parallelprozessorkern **113** und dem zweiten Parallelprozessorkern **114** entsprechend der ersten und zweiten Prioritätsstufe des zugehörigen parallelen Bearbeitungsabschnitts abgearbeitet werden.

[0062] Es ist möglich, dass ein Programm mehrere sequentielle Bearbeitungsabschnitte sowie mehrere parallele Bearbeitungsabschnitte umfasst, wobei die sequentiellen Bearbeitungsabschnitte der Programme im Unterschied zu den parallelen Bearbeitungsabschnitten nicht in Arbeitspaketen mit Teilaufgaben aufgeteilt sein und vom Prioritätenverwalter verwaltet werden müssen. Denn die sequentiellen Bearbeitungsabschnitte der Programme - mit Aufgaben, die keine Parallelisierung erfordern - können auf dem entsprechenden Hauptprozessorkern **111**, **112** ohne die Unterstützung des ersten und zweiten Parallelprozessorkerns **113**, **114** ausgeführt werden. Die sequentiellen und parallelen Bearbeitungsabschnitte eines Programms können alternierend auftreten. Zudem können Programme nur rein sequentiell ausgeführt werden. Die sequentiellen Bearbeitungsabschnitte bzw. die sequentiell auszuführenden Programme können eine sequentielle Prioritätsstufe aufweisen, die für den sequentiellen Bearbeitungsabschnitt bzw. das ganze Programm der Steuerungsaufgabe gilt. Des Weiteren ist die sequentielle Prioritätsstufe unabhängig davon, ob die Programme der Steuerungsaufgabe einen parallelen Bearbeitungsabschnitt aufweisen oder nicht. Die Aufgaben können dann rein sequentiell im sequentiellen Bearbeitungs-

abschnitt des Programms bzw. dem sequentiell auszuführenden Programm des entsprechenden Hauptprozessorkerns **111**, **112** abgearbeitet werden.

[0063] In diesem Zusammenhang ist es denkbar, dass mehrere Programme der Steuerungsaufgabe auf einem einzigen Hauptprozessorkern **111**, **112** ausgeführt werden können, wobei die sequentielle Prioritätsstufe hierbei relevant ist. 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 des parallelen Bearbeitungsabschnitts eines entsprechenden Programms auf den Parallelprozessorkernen **113**, **114** ausgeführt werden und gleichzeitig können Aufgaben, die keine Parallelisierung erfordern von dem einen Hauptprozessorkern **111**, **112** sequentiell abgearbeitet werden. Falls die sequentielle Abarbeitung der Aufgaben ohne Parallelisierung durch ein höher priorisiertes sequentiell auszuführendes Programm (mit Aufgaben ohne Parallelisierung) bzw. einen sequentiellen Bearbeitungsabschnitt eines Programms (mit Aufgaben ohne Parallelisierung) auf dem einen Hauptprozessorkern **111**, **112** unterbrochen wird, kann die parallele Abarbeitung der Teilaufgaben des entsprechenden Arbeitspakets auf dem ersten und zweiten Parallelprozessorkern **113**, **114** weiterhin unterbrechungsfrei erfolgen, da den parallelen Bearbeitungsabschnitten in den entsprechenden Programmen eine zur sequentiellen Prioritätsstufe separate Prioritätsstufe zugewiesen sein kann.

[0064] Folglich kann eine Unterbrechung 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 Aufgaben ohne Parallelisierung auf dem einen Hauptprozessorkern **111**, **112** erfolgen. Der Wert der Prioritätsstufe des parallelen Bearbeitungsabschnitts des jeweiligen Programms der Steuerungsaufgabe kann auch in der beschriebenen Ausführungsform mit einem einzigen Hauptprozessorkern **111**, **112** vom Prioritätenverwalter **350** während des Programmzyklus oder für einen neuen Programmzyklus angepasst werden.

[0065] Fig. 2 zeigt ein Schema **600** zur Erstellung von Arbeitspaketen mit Teilaufgaben und Verwaltung der Prioritätsstufen innerhalb der SPS nach Fig. 1. Die mehreren ausführbaren Programme der Steuerungsaufgabe der SPS können beispielsweise auf dem ersten und zweiten Hauptprozessorkern **111**, **112** ausgeführt werden, wobei wenigstens zwei Programme der Steuerungsaufgabe jeweils den wenigstens einen parallelen Bearbeitungsabschnitt aufweisen. 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 ersten 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 Datenstrukturspeicher der Steuerungseinrichtung, der 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.

[0066] 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.

[0067] 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 **113** 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. Die parallele Ausführung der ersten und zweiten Teilaufgabe **321**, **322** von dem ersten und zweiten Parallelprozessorkern **113**, **114** ist mithilfe des Bezugszeichens **360** und dem Kasten um den ersten und zweiten Parallelprozessorkern **113**,

114 herum angedeutet. 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.

[0068] 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 Verteilung 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.

[0069] 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 der 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.

[0070] Ändert sich die Reihenfolge der Einträge der Prioritätsstufen in der Datenstruktur **355**, so infor-

miert 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 einem nicht dargestellten Zwischenergebnisspeicher gesichert werden. Auf den Zwischenergebnisspeicher können sowohl der erste und zweite Hauptprozessorkern **111, 112** als auch der erste und zweite Parallelprozessorkern **113, 114** zugreifen, falls die Abarbeitung der unterbrochenen ersten, zweiten und gegebenenfalls vierten Teilaufgabe **321, 322, 324** auf den Parallelprozessorkern **113, 114** fortgeführt werden soll.

[0071] 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 den 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.

[0072] Die beiden Parallelprozessorkerne **113, 114** müssen nicht zwingend warten, bis alle fünften bis siebten Teilaufgaben **331-333** aus dem höher priorisierten zweiten Arbeitspaket **310** des parallelen Bearbeitungsabschnitts des Programms des zweiten Hauptprozessorkerns **112** fertig abgearbeitet worden sind. Hat der erste Parallelprozessorkern **113** zum Beispiel seine fünfte Teilaufgabe **331** bereits vollständig abgearbeitet, während der zweite Parallelprozessorkern **114** noch mit der Abarbeitung der siebten Teilaufgabe **333** und der zweite Hauptprozessorkern **112** mit der Abarbeitung der sechsten Teilaufgabe **332** des zweiten Arbeitspakets **310** beschäftigt sind, so kann der erste Parallelprozessorkern **113** bereits mit der Abarbeitung einer der zuvor unterbrochenen ersten bis vierten Teilaufgabe **321-324** des ersten Ar-

beitspakets **305** beginnen, die mit einer niedrigeren Prioritätsstufe verknüpft sind. Dies ist möglich, falls der erste Parallelprozessorkern **113** nach der Abarbeitung seiner fünften Teilaufgabe **331** beim zweiten Arbeitspaket **310** nach einer freien Teilaufgabe anfragt, es jedoch gerade keine freie Teilaufgabe an erster Stelle des zweiten Arbeitspakets mehr gibt, wobei die beiden Arbeitspakete **305**, **310** ebenfalls Teil der Datenstruktur **355** sein können. Dann kann der erste Parallelprozessorkern **113** mit der Abarbeitung einer der ersten bis vierten Teilaufgabe **321-324** aus dem ersten Arbeitspaket **305** beginnen, damit die Rechenleistung des ersten Parallelprozessorkerns **113** optimal ausgenutzt werden kann. Das heißt, unnötige Wartezeiten für den Beginn der Abarbeitung der ersten bis siebten Teilaufgaben **321-324**, **331-333** von den beiden Parallelprozessorkernen **113**, **114** können vermieden werden.

[0073] Bei der Wiederaufnahme der Abarbeitung der ersten und zweiten Teilaufgabe **321**, **322** des ersten Arbeitspakets **305** durch den ersten und zweiten Parallelprozessorkern **113**, **114** kann die Verteilung der ersten und zweiten Teilaufgabe **321**, **322** aus dem ersten Arbeitspaket **305** auf den ersten und zweiten Parallelprozessorkern **113**, **114** beispielsweise analog zum beschriebenen Ausführungsbeispiel erfolgen, d.h. der erste Parallelprozessorkern **113** bekommt die erste Teilaufgabe **321** und der zweite Parallelprozessorkern **114** die zweite Teilaufgabe **322** vom ersten Arbeitspaket **305** zugeteilt. Die beiden Parallelprozessorkerne **113**, **114** greifen auf den Zwischenergebnisspeicher zu, um die Zwischenergebnisse der ersten und zweiten Teilaufgabe **321**, **322** für die weitere Abarbeitung der genannten Teilaufgaben nutzen zu können.

[0074] Sind 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.

[0075] Da der erste Hauptprozessorkern **111** die dritte Teilaufgabe **323** des ersten Arbeitspakets **305** weiter abgearbeitet hat, während der erste Parallelprozessorkern **113** die fünfte Teilaufgabe **331** und der zweite Parallelprozessorkern **114** die siebte Teilaufgabe **333** aus dem zweiten Arbeitspaket **310** abgearbeitet haben, 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 **530** angezeigt.

[0076] Sind 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, und ferner die dritte Teilaufgabe **323** von dem ersten Hauptprozessorkern **111** fertiggestellt worden, so meldet das zugehörige erste Arbeitspaket **305** die Erledigung der ersten bis vierten Teilaufgabe **321-324** 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.

[0077] Sobald der erste und zweite Hauptprozessorkern **111**, **112** im parallelen Bearbeitungsabschnitt eine weitere erste bis siebte Teilaufgabe **321-324**, **331-333** von seinem jeweiligen Arbeitspaket **305**, **310** anfordert, aber keine unbearbeitete erste bis siebte Teilaufgabe **321-324**, **331-333** mehr verfügbar ist, wartet der erste und zweite Hauptprozessorkern **111**, **112** bis die Parallelprozessorkerne **113**, **114** die aktuell noch ausgeführten ersten bis siebten Teilaufgaben **321-324**, **331-333** zu Ende abgearbeitet haben. Dann wird der entsprechende parallele Bearbeitungsabschnitt des ersten und zweiten Hauptprozessorkerns **111**, **112** verlassen und das erste und zweite Arbeitspaket **305**, **310** gelöscht. Danach können der erste und zweite Hauptprozessorkern **111**, **112** beliebige weitere (nicht parallelisierte) Berechnungen ausführen, z.B. auf den Ergebnissen des vorherigen parallelen Bearbeitungsabschnitts aufbauend.

[0078] 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 Bear-

beitungsabschnitte 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.

[0079] In einer Ausführungsform 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** zu Beginn der Ausführung des jeweiligen Programms von dem ersten und zweiten Hauptprozessorkern **111**, **112** erstellt. Ebenfalls ist denkbar, dass am Anfang eines weiteren parallelen Bearbeitungsabschnitts im Programm des ersten Hauptprozessorkerns **111** und/oder des zweiten Hauptprozessorkerns **112** jeweils ein neues Arbeitspaket erstellt wird.

[0080] Bei der Unterbrechung der Abarbeitung der ersten bis siebten Teilaufgabe **321-324**, **331-333** in dem parallelen Bearbeitungsabschnitt des jeweiligen Programms ist in der Regel jedoch nicht bestimmbar, welche Daten für eine Fortführung der genannten Teilaufgaben im jeweiligen Programm nach der Unterbrechung relevant sind, da die erste bis siebte Teilaufgabe **321-324**, **331-333** an einer beliebigen Stelle im entsprechenden Programm während der Ausführungszeit des Programms unterbrochen werden kann. Insofern kann diese Ausführungsform die Sicherung sämtlicher Daten erfordern, die bei der Abarbeitung der ersten bis siebten Teilaufgaben **321-324**, **331-333** im parallelen Bearbeitungsabschnitt im entsprechenden Programm erstellt bzw. bearbeitet wurden, wenn kein entsprechender Unterbrechungspunkt in den ersten bis siebten Teilaufgaben **321-324**, **331-333** gesetzt wurde, bis zu dem die Abarbeitung der ersten bis siebten Teilaufgaben **321-324**, **331-333** fortgeführt und dann unterbrochen wird. Denn mithilfe des vorgegebenen Un-

terbrechungspunkts, der beispielsweise in einer Kontrollstruktur in den ersten bis siebten Teilaufgaben **321-324**, **331-333** gesetzt sein kann, ist bekannt, welche Daten bei der Fortführung der unterbrochenen ersten bis siebten Teilaufgaben **321-324**, **331-333** 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.

[0081] Eine weitere Ausführungsform sieht vor, dass die Abarbeitung der ersten bis siebten Teilaufgabe **321-324**, **331-333** in dem parallelen Bearbeitungsabschnitt des jeweiligen Programms auf dem ersten und zweiten Parallelprozessorkern **113**, **114** umgehend abgebrochen wird und dabei keine Zwischenergebnisse der Berechnungen gesichert werden. Folglich muss eine abgebrochene erste bis siebte Teilaufgabe **321-324**, **331-333** wieder von Beginn an neu berechnet werden. 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 mit einer sehr kleinen Programmzyklus-Zeit 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** (fünften 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 Teilaufgabe **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.

[0082] Statt der Unterbrechung der aktuell ausgeführten ersten bis vierten Teilaufgabe **321-324** ist es ferner denkbar, die erste bis vierte Teilaufgabe **321-324** bis zum Ende abzuarbeiten und eine neue fünfte bis siebte Teilaufgabe **331-333** mit höherer Prioritätsstufe des jeweiligen parallelen Bearbeitungsabschnitts, geeignet in die Datenstruktur **355** des Prioritätenverwalters **350** einzusortieren, jedoch ohne die Abarbeitung der aktuellen ersten bis vierten Teilaufgaben **321-324** dabei zu unterbrechen. Die Parallelprozessorkerne **113**, **114** werden auf diese Weise nicht gezwungen, ihre Berechnungen umge-

hend abubrechen oder sie zu unterbrechen, falls der neuen fünften bis siebten Teilaufgabe **331-333** eine höhere Priorität zugeordnet ist, als der aktuellen ersten bis vierten Teilaufgabe **321-324**. Denn sobald die beiden Parallelprozessorkerne **113, 114** die Abarbeitung ihrer aktuellen ersten bis vierten Teilaufgaben **321-324** mit niedrigerer Prioritätsstufe abgeschlossen haben, bekommen sie bei der Anfrage nach der nächsten Teilaufgabe über den herkömmlichen Weg eine der neuen, höher priorisierten fünften bis siebten Teilaufgaben **331-333** zugeteilt. 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.

[0083] Die Ausführungsform, bei der die aktuell bearbeiteten ersten bis siebten Teilaufgaben **321-324, 331-333** unterbrochen und deren Zwischenergebnisse auf dem Zwischenergebnisspeicher 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.

[0084] In einer Ausführungsform 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. 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 Parallelprozessorkerne **113, 114** 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.

[0085] 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 Arbeitspakets **305, 310** an den Prioritätenverwalter **350** geleitet wird.

[0086] 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 parallele Ausführung **360** der ersten bis siebten Teilaufgaben **321-324, 331-333** teilen.

[0087] Die bisherige Beschreibung hat sich darauf bezogen, dass der Wert der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts des entsprechenden Programms unverändert während eines Programmzyklus bleibt, um die Erläuterung der Erstellung der Arbeitspakete und des Zugriffs auf die Arbeitspakete sowie die Verwaltung der Prioritätsstufen der parallelen Bearbeitungsabschnitte der Programme zu vereinfachen. Der Wert der wenigstens einen Prioritätsstufe der jeweiligen parallelen Bearbeitungsabschnitte der Programme der Steuerungsaufgabe kann für die in den **Fig. 1** und **Fig. 2** gezeigten Ausführungsformen vom Prioritätenverwalter ebenso angepasst werden, wie es in den nachfolgenden Figuren, im Fall der Unterbrechung des parallelen Bearbeitungsabschnitts der Programme beschrieben wird. Um den Rechendurchsatz der Steuerungsaufgabe zu erhöhen, ist es bei der Bewertung des Rechendurchsatzes für die **Fig. 1** und **Fig. 2** (sowie die nachfolgenden Figuren) ferner möglich, die Rechenzeit des jeweiligen parallelen Bearbeitungsabschnitts der Programme der Steuerungsaufgabe mit zu berücksichtigen und Teilaufgaben der entsprechenden Programme, die keine zwingende parallele Ausführung erfordern rein sequentiell auf dem zugeordneten Hauptprozessorkern abzarbeiten. Ferner ist es günstig, bei der Bewertung des Rechendurchsatzes der Steuerungsaufgabe die Rechenleistung der beiden Parallelprozessorkerne mit zu berücksichtigen, und ihnen im Fall der nicht vollständigen Ausnutzung ihrer Rechenleistung Teilaufgaben für die parallele Abarbeitung zuzuteilen. Dazu kann es erforderlich sein, dass der Prioritätenverwalter den

Wert der wenigstens einen Prioritätsstufe der parallelen Bearbeitungsabschnitte der Programme geeignet anpasst.

[0088] Fig. 3 zeigt ein Schema eines Verfahrens **300** zur Datenverarbeitung auf der SPS nach den Fig. 1 und Fig. 2. Als Ausgangspunkt dient ebenso wie in den Fig. 1 und Fig. 2 gezeigt wurde, der erste und zweite Hauptprozessoren **111**, **112** sowie der erste und zweite Parallelprozessoren **113**, **114**. Es ist hingegen auch eine dazu abweichende Anzahl an Haupt- und/oder Parallelprozessoren denkbar. Jedoch ist wenigstens ein Parallelprozessor für eine parallele Ausführung **375**, **380**, **385**, **390** der Teilaufgaben der jeweiligen parallelen Bearbeitungsabschnitte **303**, **304** der Programme der Steuerungsaufgabe notwendig. Wird eine zu den in den Fig. 1 bis Fig. 3 gezeigte, abweichende Anzahl an Hauptprozessoren eingesetzt, so können sich die Anzahl der Arbeitspakete und die Anzahl der Teilaufgaben folglich unterscheiden. In dem in Fig. 3 gezeigten Ausführungsbeispiel gelangt zunächst der erste Hauptprozessor **111** im Programm an seinen parallelen Bearbeitungsabschnitt **303**. Der parallele Bearbeitungsabschnitt ist mithilfe des obersten gestreiften horizontalen Balkens und der geschweiften Klammer mit dem Bezugszeichen **303** angedeutet. Zu Beginn seines parallelen Bearbeitungsabschnitts **303** erstellt der erste Hauptprozessor **111** das erste Arbeitspaket mit der ersten bis vierten Teilaufgabe und der Referenz auf die erste Prioritätsstufe des zugeordneten parallelen Bearbeitungsabschnitts **303**, wie oben beschrieben. An dieser Stelle wird auf eine explizite Beschreibung der Zuteilung der jeweiligen Teilaufgaben verzichtet, da sie in gleicher Weise erfolgen kann, wie bereits in Fig. 2 beschrieben wurde.

[0089] Die erste bis vierte Teilaufgabe kann im parallelen Bearbeitungsabschnitt **303** des entsprechenden Programms auf dem ersten Hauptprozessor **111** sequentiell ausgeführt werden, was mithilfe des Bezugszeichens **365** verdeutlicht wird. Im parallelen Bearbeitungsabschnitt **303** des ersten Hauptprozessors **111** kann die erste bis vierte Teilaufgabe zusätzlich zur sequentiellen Ausführung **365** von dem ersten und zweiten Parallelprozessoren **113**, **114** auch parallel ausgeführt werden. Die parallele Ausführung der ersten bis vierten Teilaufgabe ist über die beiden unteren gestreiften Balken mit den Bezugszeichen **375** und **380** dargestellt. Zuerst sind nur die erste bis vierte Teilaufgabe des ersten Arbeitspakets des parallelen Bearbeitungsabschnitts **303** des ersten Hauptprozessors **111** verfügbar, welche daher konkurrenzlos und ohne vorherige Unterbrechung von dem ersten und zweiten Parallelprozessoren **113**, **114** abgearbeitet werden können.

[0090] Gelangt jedoch der zweite Hauptprozessor **112** im Programm an seinen parallelen Bear-

beitungsabschnitt **304**, so werden die Berechnungen der zuvor aktiv abgearbeiteten ersten bis vierten Teilaufgaben auf dem ersten und zweiten Parallelprozessoren **113**, **114** unterbrochen. Die Unterbrechung der Abarbeitung der ersten bis vierten Teilaufgabe des parallelen Bearbeitungsabschnitts **303** des Programms, das auf dem ersten Hauptprozessor **111** ausgeführt wird, ist mithilfe des Bezugszeichens **366** dargestellt. Die Unterbrechung der Abarbeitung der ersten bis vierten Teilaufgabe des parallelen Bearbeitungsabschnitts **303** des entsprechenden Programms kann nach einer der im Zusammenhang mit Fig. 2 erläuterten Möglichkeiten erfolgen. Gegebenenfalls können ermittelte Zwischenergebnisse der Berechnungen an den Zwischenergebnisspeicher übergeben und dort gesichert werden. Dieser Verwaltungsaufwand führt dazu, dass der erste und zweite Parallelprozessoren **113**, **114** nach dem Zeitpunkt der Unterbrechung der Abarbeitung der ersten bis vierten Teilaufgabe des parallelen Bearbeitungsabschnitts **303** des Programms, das auf dem ersten Hauptprozessor **111** ausgeführt wird, nicht umgehend mit der Abarbeitung der Teilaufgaben des parallelen Bearbeitungsabschnitts **304** des entsprechenden Programms, das auf dem zweiten Hauptprozessor **112** ausgeführt wird, beginnen können. Aufgrund des Verwaltungsaufwands kann die Abarbeitung der Teilaufgaben des parallelen Bearbeitungsabschnitts **304** des Programms, das auf dem zweiten Hauptprozessor **112** ausgeführt wird, erst mit entsprechender zeitlicher Verzögerung erfolgen.

[0091] Der zweite Hauptprozessor **112** erstellt zu Beginn seines parallelen Bearbeitungsabschnitts **304** im entsprechenden Programm das zweite Arbeitspaket **310** mit der fünften bis siebten Teilaufgabe und der Referenz auf die Prioritätsstufe des zugeordneten parallelen Bearbeitungsabschnitts **304**. Die fünfte bis siebte Teilaufgabe kann im parallelen Bearbeitungsabschnitt **304** des entsprechenden Programms auf dem zweiten Hauptprozessor **112** sequentiell ausgeführt werden, was mithilfe des Bezugszeichens **370** verdeutlicht wird. Im parallelen Bearbeitungsabschnitt **304** des zweiten Hauptprozessors **112** kann die fünfte bis siebte Teilaufgabe zusätzlich zur sequentiellen Ausführung **370** von dem ersten und zweiten Parallelprozessoren **113**, **114** auch parallel ausgeführt werden. Die parallele Ausführung der fünften bis siebten Teilaufgabe ist über die beiden unteren gepunkteten Balken mit den Bezugszeichen **385**, **390** dargestellt. In Fig. 3 wurde angenommen, dass der Wert der zweiten Prioritätsstufe des parallelen Bearbeitungsabschnitts **304** des entsprechenden Programms, das auf dem zweiten Hauptprozessor **112** ausgeführt wird, höher ist, als der Wert der ersten Prioritätsstufe des entsprechenden parallelen Bearbeitungsabschnitts **303** des anderen Programms, das auf dem ersten Hauptprozessor **111** ausgeführt wird. Diese Annah-

me führt dazu, dass die Abarbeitung der ersten bis vierten Teilaufgabe des parallelen Bearbeitungsabschnitts **303** des entsprechenden Programms auf dem ersten und zweiten Parallelprozessorkern **113**, **114** unterbrochen wird.

[0092] Weitet man das dargestellte Ausführungsbeispiel in **Fig. 3** auf weitere Haupt- und Parallelprozessorkerne aus, so ist ersichtlich, dass die Anzahl der Unterbrechungen bei der Abarbeitung der entsprechenden Teilaufgaben der entsprechenden parallelen Bearbeitungsabschnitte der Programme auf den Parallelprozessorkernen zunehmen kann. Falls die Teilaufgaben der jeweiligen parallelen Bearbeitungsabschnitte **303**, **304** der Programme hingegen eine nicht zwingende parallele Ausführung **375**, **380**, **385**, **390** auf dem ersten und zweiten Parallelprozessorkern **113**, **114** erfordern, so kann der dem jeweiligen parallelen Bearbeitungsabschnitt **303**, **304** zugeordnete Wert der Prioritätsstufe dynamisch während des Programmzyklus angepasst werden. Im konkreten Beispiel kann der Wert der zweiten Prioritätsstufe vom Prioritätenverwalter um einen Faktor reduziert bzw. sofort auf einen minimalen Wert gesetzt werden, damit die fünfte bis siebte Teilaufgabe des parallelen Bearbeitungsabschnitts **304** des entsprechenden Programms direkt sequentiell vom zweiten Hauptprozessorkern **112** abgearbeitet werden können, ohne dass die Abarbeitung der ersten bis vierten Teilaufgaben **321-324** auf den beiden Parallelprozessorkernen **113**, **114** unterbrochen werden muss. Der Vorteil an der Verringerung des Werts der Prioritätsstufe des parallelen Bearbeitungsabschnitts **304** des Programms, das auf dem zweiten Hauptprozessorkern **112** ausgeführt wird, ist, dass die Unterbrechung der Abarbeitung der ersten bis vierten Teilaufgabe des parallelen Bearbeitungsabschnitts **303** des zugeordneten Programms, das auf dem ersten Hauptprozessorkern **111** ausgeführt wird, dadurch vermieden werden kann. Somit ist es möglich den Verwaltungsaufwand zu minimieren, wobei die dynamische Anpassung des Werts der Prioritätsstufe des parallelen Bearbeitungsabschnitts der Programme vom Prioritätenverwalter erfolgt, wenn der Rechendurchsatz der Steuerungsaufgabe dadurch erhöht werden kann. Die Erhöhung des Rechendurchsatzes der Steuerungsaufgabe kann sich insbesondere positiv auf Programme mit einem hohen Wert der Prioritätsstufe des parallelen Bearbeitungsabschnitts **303**, **304** des Programms auswirken und kann zum Beispiel kürzere Deadlines ermöglichen. Auf diese Weise kann es möglich sein, die Arbeitsgeschwindigkeit der Maschine oder Anlage des Automatisierungssystems zu erhöhen.

[0093] Ferner kann mithilfe des Prioritätenverwalters die Echtzeitfähigkeit des Automatisierungssystems gewährleistet werden, da der Prioritätenverwalter die Anpassung der Werte der Prioritätsstufen der jeweiligen parallelen Bearbeitungsabschnitte **303** der

entsprechenden Programme in der Weise vornimmt, dass die Deadlines der Programme in jedem Fall eingehalten werden.

[0094] Der Wert der wenigstens einen Prioritätsstufe kann beliebig verringert werden. Beispielsweise kann der Wert der Prioritätsstufe bei der ersten Unterbrechung des jeweiligen parallelen Bearbeitungsabschnitts **303**, **304** des Programms vom Prioritätenverwalter unverändert bleiben. Nach der zweiten Unterbrechung des entsprechenden parallelen Bearbeitungsabschnitts **303**, **304** des Programms (nicht dargestellt) kann der Prioritätenverwalter den Wert der Prioritätsstufe des zugeordneten parallelen Bearbeitungsabschnitts **303**, **304** um einen Faktor reduzieren. Dabei kann der Wert der Prioritätsstufe des entsprechenden parallelen Bearbeitungsabschnitts **303**, **304** des Programms zum Beispiel umso stärker reduziert werden, je häufiger der jeweilige parallele Bearbeitungsabschnitt **303**, **304** mit den enthaltenen Teilaufgaben des Programms unterbrochen wird.

[0095] Der Prioritätenverwalter kann die Anpassung des Werts der Prioritätsstufe des parallelen Bearbeitungsabschnitts **303**, **304** dann vornehmen, wenn die Teilaufgaben aufgrund von anderen Teilaufgaben eines weiteren Programms mit höherer Prioritätsstufe auf den beiden Parallelprozessorkern **113**, **114** unterbrochen werden. Der Prioritätenverwalter kann den Wert der Prioritätsstufe des parallelen Bearbeitungsabschnitts **303**, **304** des entsprechenden Programms für die unterbrochenen Teilaufgaben auf den beiden Parallelprozessorkern **113**, **114** anpassen, um einer weiteren Unterbrechung der Teilaufgaben auf den beiden Parallelprozessorkern **113**, **114** vorzubeugen und in diesem Zusammenhang für möglichst wenig Verwaltungsaufwand für die noch ausstehenden Teilaufgaben in dem jeweiligen Arbeitspaket zu sorgen.

[0096] Ferner kann der Prioritätenverwalter auch den Wert der Prioritätsstufe der parallelen Bearbeitungsabschnitte **303**, **304** der Programme anpassen, der für die Unterbrechung der Abarbeitung der Teilaufgaben der parallelen Bearbeitungsabschnitte der Programme verantwortlich ist. Alternativ dazu kann der Prioritätenverwalter den Wert der Prioritätsstufe der parallelen Bearbeitungsabschnitte **303**, **304** der entsprechenden Programme dann anpassen, wenn sich die jeweilige Prioritätsstufe als Eintrag in der Datenstruktur im Datenstrukturspeicher befindet und gerade aktiv keine Teilaufgaben aus dem zugeordneten parallelen Bearbeitungsabschnitt **303**, **304** von den beiden Haupt- und Parallelprozessorkernen **111-114** abgearbeitet werden. Ferner kann der Prioritätenverwalter den Wert der Prioritätsstufe der parallelen Bearbeitungsabschnitte **303**, **304** der entsprechenden Programme dann anpassen, wenn Teilaufgaben im jeweiligen parallelen Bearbeitungsabschnitt **303**, **304** des Programms aktiv von den Prozessorkernen abgearbeitet werden und im zugehörigen Ar-

beitspaket freibleibende, unbearbeitete Teilaufgaben vorhanden sind. Im diesem Fall betrifft die Anpassung des Werts der Prioritätsstufe des parallelen Bearbeitungsabschnitts **303, 304** die freibleibenden Teilaufgaben des Arbeitspakets.

[0097] Von Vorteil bei der Anpassung des Werts der Prioritätsstufe der parallelen Bearbeitungsabschnitte **303, 304** der jeweiligen Programme ist, dass diese vorübergehend erfolgen kann. Die dynamische Anpassung des Werts der wenigstens einen Prioritätsstufe muss nicht für den gesamten Programmzyklus in gleicher Weise erfolgen. Der Wert der Prioritätsstufe kann beispielsweise für einen neuen Programmzyklus wieder auf den vorgegebenen Wert zurückgesetzt werden und für einen weiteren Programmzyklus erneut abgeändert werden. Auch innerhalb des Programmzyklus kann der Wert der wenigstens einen Prioritätsstufe der parallelen Bearbeitungsabschnitte der Programme vom Prioritätenverwalter beliebig oft angepasst werden. In diesem Zusammenhang ist es weiterhin möglich, den Wert der wenigstens einen Prioritätsstufe der parallelen Bearbeitungsabschnitte **303, 304** der jeweiligen Programme für den nächsten Programmzyklus zu erhöhen, falls die Teilaufgaben im vorangehenden Programmzyklus rein sequentiell in den jeweiligen Programmen abgearbeitet wurden, damit die Deadlines der Programme eingehalten werden können oder um dafür Sorge zu tragen, dass die erste bis siebte Teilaufgabe der jeweiligen parallelen Bearbeitungsabschnitte der Programme für den jeweiligen nächsten Programmzyklus parallel ausgeführt werden. Gleichzeitig kann so sichergestellt werden, dass die Rechenleistung der beiden Parallelprozessorkerne **113, 114** bestmöglich ausgenutzt wird.

[0098] Ebenso denkbar ist eine Ausführungsform, in der ein Programm, welches umgehend ausgeführt werden muss (zum Beispiel aufgrund einer kurzen Programmzyklus-Zeit und Deadline), mit einem sehr hohen Wert der Prioritätsstufe des parallelen Bearbeitungsabschnitts des Programms vom Prioritätenverwalter versehen wird, und dieser Wert der Prioritätsstufe vom Prioritätenverwalter trotz einer möglichen Unterbrechung des parallelen Bearbeitungsabschnitts **303, 304** mit entsprechender Prioritätsstufe für den nächsten Programmzyklus weiterhin unverändert bleibt. Auch kann der Wert der Prioritätsstufe des jeweiligen parallelen Bearbeitungsabschnitts **303, 304** des Programms mit der entsprechenden ersten bis siebten Teilaufgabe bei einer Unterbrechung des jeweiligen parallelen Bearbeitungsabschnitts erhöht werden, damit der entsprechende parallele Bearbeitungsabschnitt **303, 304** mit der jeweiligen ersten bis siebten Teilaufgabe des Programms im nächsten Programmzyklus nicht mehr unterbrochen wird.

[0099] Die genannten Möglichkeiten für die Anpassung des Werts der Prioritätsstufe des parallelen

Bearbeitungsabschnitts **303, 304** des Programms bezogen sich jeweils darauf, den Verwaltungsaufwand durch eine geringe Anzahl an Unterbrechungen der parallelen Bearbeitungsabschnitte **303, 304** der Programme zu reduzieren und eine bestmögliche Ausnutzung der Rechenleistung der Prozessorkerne bereitzustellen. Die Anzahl der Unterbrechungen der parallelen Bearbeitungsabschnitte **303, 304** und ihre zeitliche Unterbrechungsdauer sowie die genutzte Rechenleistung der Prozessorkerne kann von der Steuerungseinrichtung der SPS ermittelt werden, wenn der zu erwartende Rechendurchsatz der Steuerungseinrichtung bewertet werden soll. Ebenso kann der zu erwartende Rechendurchsatz auch anhand der Rechenzeit der parallelen Bearbeitungsabschnitte **303, 304** der Programme der Steuerungsaufgabe ermittelt werden. Das heißt Teilaufgaben der entsprechenden Programme, die keine zwingende parallele Ausführung **375, 380, 385, 390** erfordern, können ebenso rein sequentiell auf dem zugeordneten Hauptprozessorkern abgearbeitet werden, wenn die sequentielle Rechenzeit kürzer ist, als die Rechenzeit der Teilaufgaben des parallelen Bearbeitungsabschnitts **303, 304** mit möglichen Unterbrechungen des parallelen Bearbeitungsabschnitts **303, 304**. Dazu kann der Prioritätenverwalter den Wert der wenigstens einen Prioritätsstufe der parallelen Bearbeitungsabschnitte **303, 304** der Programme reduzieren bzw. minimieren, damit die Teilaufgaben der parallelen Bearbeitungsabschnitte **303, 304** gegebenenfalls sequentiell abgearbeitet werden, falls sie nicht notwendigerweise parallel ausgeführt werden müssen. Die Steuerungsaufgabe muss zur Bewertung des Rechendurchsatzes im genannten Fall zudem prüfen, ob die Rechenleistung der beiden Parallelprozessorkerne **113, 114** durch eine sequentielle Abarbeitung der Teilaufgaben der parallelen Bearbeitungsabschnitte **303, 304** weiterhin bestmöglich genutzt wird.

[0100] Fig. 4 zeigt ein Ablaufdiagramm zur dynamischen Anpassung wenigstens einer Prioritätsstufe eines parallelen Bearbeitungsabschnitts eines Programms nach Fig. 3, wobei die Anpassung des Werts der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts des entsprechenden Programms der Steuerungsaufgabe vom Prioritätenverwalter **350** in den Fig. 1 und Fig. 2 erfolgt. Ein erster Schritt **401** zeigt den Start der dynamischen Anpassung des Werts der wenigstens einen Prioritätsstufe des parallelen Bearbeitungsabschnitts des entsprechenden Programms an. In einem zweiten Schritt **402** werden die Startwerte der ersten bis k-ten Prioritätsstufe der parallelen Bearbeitungsabschnitte der Programme vom Prioritätenverwalter für den jeweiligen Programmzyklus festgelegt, wobei keine ganze, positive Zahl echt größer eins sein kann. Die Werte der Prioritätsstufen der jeweiligen parallelen Bearbeitungsabschnitte der Programme können sich um ganzzahlige Werte handeln.

[0101] In einem nachfolgenden dritten Schritt **403** wird in einer Verzweigung geprüft, ob die Abarbeitung der ersten bis vierten Teilaufgabe des ersten Arbeitspakets im entsprechenden parallelen Bearbeitungsabschnitt (mit zugeordneter erster Prioritätsstufe) des Programms aufgrund von Teilaufgaben des k-ten Arbeitspakets im jeweiligen parallelen Bearbeitungsabschnitt (mit zugeordneter k-ter Prioritätsstufe) unterbrochen wurden, wobei der Wert der k-ten Prioritätsstufe höher ist, als der Wert der ersten Prioritätsstufe. Wenn dies nicht der Fall ist, erfolgt ein Rücksprung zu dem zweiten Schritt **402**, also dem Festlegen der jeweiligen Startwerte der Prioritätsstufen der parallelen Bearbeitungsabschnitte der Programme durch den Prioritätenverwalter. Ergibt die Prüfung auf Unterbrechung der Abarbeitung der Teilaufgaben im dritten Schritt **403** ein positives Ergebnis, so wird in einem vierten Schritt **404** ferner geprüft, ob die unterbrochenen ersten bis vierten Teilaufgaben des ersten Arbeitspakets des parallelen Bearbeitungsabschnitts des entsprechenden Programms der Steuerungsaufgabe sequentiell ausgeführt werden sollen.

[0102] Beispielsweise kann der Prioritätenverwalter bei Programmen mit sehr kurzer Programmzyklus-Zeit den Wert der Prioritätsstufe der parallelen Bearbeitungsabschnitte der Programme direkt bei der ersten Unterbrechung der entsprechenden Teilaufgaben auf den minimalen Wert setzen, damit die Teilaufgaben sequentiell vom jeweiligen Hauptprozessorkern abgearbeitet werden. Setzt der Prioritätenverwalter hingegen bei der ersten Unterbrechung der Abarbeitung der Teilaufgaben der parallelen Bearbeitungsabschnitte der Programme den Wert jeglicher Prioritätsstufen auf den minimalen Wert, so sind die Parallelprozessorkerne gegebenenfalls nicht mehr voll ausgelastet. Das heißt, es ist freie Rechenzeit verfügbar, wenn die höher priorisierten parallelen Bearbeitungsabschnitte mit Teilaufgaben fertig abgearbeitet und noch keine weiteren höher priorisierten Teilaufgaben eines weiteren Programms mit parallelem Bearbeitungsabschnitt für die Abarbeitung auf den Parallelprozessorkernen vorhanden sind. Diese verfügbare Zeit sollte von den beiden Parallelprozessorkernen im Normalfall für die Abarbeitung von niedriger priorisierten parallelen Bearbeitungsabschnitten mit Teilaufgaben verwendet werden. Erst wenn die Steuerungseinheit bei der Bewertung des Rechendurchsatzes merkt, dass die niedriger priorisierten parallelen Bearbeitungsabschnitte mit Teilaufgaben eines Arbeitspakets trotz mehrfacher Verringerung des Werts der Prioritätsstufe häufig unterbrochen wurden, kann sie den Prioritätenverwalter dazu anstoßen, für die unterbrochenen Teilaufgaben der parallelen Bearbeitungsabschnitte diese minimale Prioritätsstufe zu setzen.

[0103] Sind die ersten bis vierten Teilaufgaben des ersten Arbeitspakets des parallelen Bearbeitungsabschnitts des Programms auf die Parallelisierung an-

gewiesen, um die Deadline des Programms einzuhalten, so passt der Prioritätenverwalter den Wert der ersten Prioritätsstufe (und/oder gegebenenfalls der k-ten Prioritätsstufe) des parallelen Bearbeitungsabschnitts des entsprechenden Programms der Steuerungsaufgabe in einem fünften Schritt **406** entsprechend an. Die Anpassung des Werts der ersten und/oder k-ten Prioritätsstufe des parallelen Bearbeitungsabschnitts des entsprechenden Programms von dem Prioritätenverwalter kann in einem Verringern oder Erhöhen des Werts der Prioritätsstufe liegen, analog zu der Beschreibung in **Fig. 3**, um weiteren möglichen Unterbrechungen des parallelen Bearbeitungsabschnitts des Programms mit der ersten bis vierten Teilaufgabe vorzubeugen bzw. deren Anzahl zu reduzieren, um den Verwaltungsaufwand zu minimieren und den Rechendurchsatz der Steuerungsaufgabe zu erhöhen.

[0104] Wird die Prüfung im vierten Schritt **404** hingegen bejaht, so setzt der Prioritätenverwalter in einem sechsten Schritt **405** den Wert der ersten Prioritätsstufe des parallelen Bearbeitungsabschnitts des Programms auf einen minimalen Wert, damit die Teilaufgaben des parallelen Bearbeitungsabschnitts des Programms für den aktuellen Programmzyklus direkt rein sequentiell auf dem entsprechenden Hauptprozessorkern ausgeführt werden können. Die Verzweigung im vierten Schritt **404** führt für die beiden Möglichkeiten, die oben beschrieben wurden, auf einen siebten Schritt **407**, einer weiteren Verzweigung. In der Verzweigung im siebten Schritt **407** prüft der Prioritätenverwalter, ob der parallele Bearbeitungsabschnitt des jeweiligen Programms beendet ist. Ist dies nicht der Fall, das heißt, dauert die Abarbeitung der ersten bis vierten Teilaufgaben des parallelen Bearbeitungsabschnitts des Programms noch an, so erfolgt ein Rücksprung zum dritten Schritt **403**, in dem geprüft wird, ob die erste bis vierte Teilaufgabe des ersten Arbeitspakets des parallelen Bearbeitungsabschnitts des Programms aufgrund von Teilaufgaben des k-ten Arbeitspakets mit höherem Wert der k-ten Prioritätsstufe unterbrochen wurden. Die nachfolgenden Schritte der Verzweigung des dritten Schritts **403** erfolgen dann wie bereits beschrieben.

[0105] Ergibt die Prüfung im siebten Schritt **407**, dass der parallele Bearbeitungsabschnitt beendet ist, so führt die Verzweigung direkt auf einen achten Schritt **408**. Im achten Schritt **408** prüft der Prioritätenverwalter, ob der Wert der angepassten ersten Prioritätsstufe des parallelen Bearbeitungsabschnitts des Programms für den nächsten Programmzyklus beibehalten werden soll. Falls die Prüfung des Prioritätenverwalters im achten Schritt **408** zu einem positiven Ergebnis führt, so wird der Wert der angepassten ersten Prioritätsstufe des parallelen Bearbeitungsabschnitts des Programms für den nächsten Programmzyklus beibehalten und der achte Schritt **408** zurückgeführt auf die Verzweigung des dritten Schritts **403**.

Im dritten Schritt **403** erfolgt dann die Prüfung, ob die Abarbeitung der ersten bis vierten Teilaufgabe des ersten Arbeitspakets des parallelen Bearbeitungsabschnitts des Programms aufgrund von Teilaufgaben des k-ten Arbeitspakets des entsprechenden parallelen Bearbeitungsabschnitts mit höherer k-ter Prioritätsstufe unterbrochen wurde. Die nachfolgenden Prüfungsschritte können in der Weise wie bereits beschrieben erfolgen.

[0106] Soll der Wert der angepassten ersten Prioritätsstufe des parallelen Bearbeitungsabschnitts des Programms für den nächsten Programmzyklus nicht beibehalten werden, so setzt der Prioritätenverwalter den Wert der ersten Prioritätsstufe im einem neunten Schritt **409** auf den vorgegebenen Wert, also den Startwert zurück. Die dynamische Anpassung des Werts der Prioritätsstufe des parallelen Bearbeitungsabschnitts des Programms ist in einem zehnten Schritt **410** beendet. Beispielsweise kann dies der Fall sein, wenn die entsprechenden Arbeitspakete nach erfolgter Abarbeitung der Teilaufgaben gelöscht und die jeweiligen Einträge der Prioritätsstufen der parallelen Bearbeitungsabschnitte aus der Datenstruktur entfernt wurden.

[0107] Die einzelnen ersten bis zehnten Schritte **401** bis **410** können vom Prioritätenverwalter und gegebenenfalls der Steuerungseinrichtung geprüft werden, da die Steuerungseinrichtung den zu erwartenden Rechendurchsatz der Steuerungsaufgabe laufend bewertet und den Prioritätenverwalter gegebenenfalls zur Anpassung des Werts der Prioritätsstufe des parallelen Bearbeitungsabschnitts des jeweiligen Programms anstößt.

[0108] 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 Schutzzumfang der Erfindung zu verlassen.

Bezugszeichenliste

100	Speicherprogrammierbare Steuerung	220	Sensoren
110	Datenverarbeitungseinrichtung	300	Verfahren zur Datenverarbeitung auf der SPS
111	erster Hauptprozessorkern	303	paralleler Bearbeitungsabschnitt des ersten Hauptprozessorkerns
112	zweiter Hauptprozessorkern	304	paralleler Bearbeitungsabschnitt des zweiten Hauptprozessorkerns
113	erster Parallelprozessorkern	310	zweites Arbeitspaket
114	zweiter Parallelprozessorkern	320	erste Menge an Teilaufgaben
120	Steuerungseinrichtung	330	zweite Menge an Teilaufgaben
130	Kommunikationsschnittstelle	321	erste Teilaufgabe
200	Maschine/Anlage	322	zweite Teilaufgabe
210	Aktoren	323	dritte Teilaufgabe
		324	vierte Teilaufgabe
		331	fünfte Teilaufgabe
		332	sechste Teilaufgabe
		333	siebte Teilaufgabe
		340	erste Prioritätsstufe
		345	zweite Prioritätsstufe
		350	Prioritätenverwalter
		355	Datenstruktur
		360	Parallele Ausführung
		365	Sequentielle Ausführung des ersten Hauptprozessorkerns
		366	Unterbrechung
		370	Sequentielle Ausführung des zweiten Hauptprozessorkerns
		375	erste parallele Ausführung des ersten Parallelprozessorkerns
		380	erste parallele Ausführung des zweiten Parallelprozessorkerns
		385	zweite parallele Ausführung des ersten Parallelprozessorkerns
		390	zweite parallele Ausführung des zweiten Parallelprozessorkerns
		400	Ablaufdiagramm zur dynamischen Anpassung einer Prioritätsstufe
		401	Start
		402	Startwerte erste bis k-te Prioritätsstufe
		403	Teilaufgaben des ersten Arbeitspakets aufgrund von Teilaufgaben des k-ten Arbeitspakets mit einer höheren Prioritätsstufe unterbrochen?
		404	Teilaufgaben des ersten Arbeitspakets auch sequentiell ausführbar?

- 405** Setze jeweilige Prioritätsstufe auf einen minimalen Wert
- 406** Reduziere oder Erhöhe den Wert der ersten und/oder k-ten Prioritätsstufe
- 407** paralleler Bearbeitungsabschnitt beendet?
- 408** Wert der jeweiligen angepassten Prioritätsstufe für den nächsten Programmzyklus beibehalten?
- 409** Setze Wert der jeweiligen Prioritätsstufe auf den entsprechenden Startwert
- 410** 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
- 600** Erstellung von Arbeitspaketen mit Teilaufgaben und Verwaltung der Prioritätsstufen

Patentansprüche

1. Verfahren zum Optimieren einer Datenverarbeitung (300) auf einer speicherprogrammierbaren Steuerung (100), die wenigstens einen Parallelprozessorkern (113, 114) umfasst, wobei eine Steuerungsaufgabe mehrere ausführbare Programme umfasst, wobei wenigstens zwei Programme der Steuerungsaufgabe jeweils wenigstens einen parallelen Bearbeitungsabschnitt (303) mit einem Arbeitspaket (305, 310) aufweisen, und das Arbeitspaket (305, 310) 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 ein Ausführen des Programms 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 Programms, dessen Prioritätsstufe (340, 345) an erster Stelle der Einträge in der Datenstruktur (355) steht, von dem wenigstens einen Parallelprozessorkern (113, 114) abgearbeitet werden, wobei während eines Programmzyklus ein zu erwartender Rechendurchsatz der Steuerungsaufgabe laufend bewertet wird, und wobei wenigstens eine der Prioritätsstufen (340, 345) der parallelen Bearbeitungsabschnitte (340, 345) der jeweiligen Programme dynamisch angepasst wird, wenn sich der Rechendurchsatz der Steuerungsaufgabe dadurch erhöht.

2. Verfahren nach Anspruch 1, wobei die Anpassung der wenigstens einen Prioritätsstufe (340, 345) der parallelen Bearbeitungsabschnitte (303) der jeweiligen Programme bei einer Unterbrechung (366) der Abarbeitung der Teilaufgaben (321-324, 331-333) des Programms erfolgt.

3. Verfahren nach Anspruch 2, wobei die Anpassung der wenigstens einen Prioritätsstufe (340, 345) der parallelen Bearbeitungsabschnitte (303) der jeweiligen Programme erfolgt, wenn die Teilaufgaben (321-324, 331-333) des Programms aufgrund von anderen Teilaufgaben eines weiteren Programms (321-324, 331-333) mit höherer Prioritätsstufe (340, 345) des parallelen Bearbeitungsabschnitts (303) auf dem wenigstens einen Parallelprozessorkern (113, 114) unterbrochen werden.

4. Verfahren nach Anspruch 1 bis 3, wobei bei der Anpassung der wenigstens einen Prioritätsstufe (340, 345) der parallelen Bearbeitungsabschnitte (303) der jeweiligen Programme eine zusätzliche Bedingung, insbesondere eine Echtzeitbedingung der Steuerungsaufgabe, berücksichtigt wird.

5. Verfahren nach Anspruch 3 oder 4, wobei die wenigstens eine Prioritätsstufe (340, 345) der parallelen Bearbeitungsabschnitte (303) der jeweiligen Programme um einen Faktor reduziert wird, wenn die Teilaufgaben (321-324, 331-333) des Programms aufgrund von anderen Teilaufgaben (321-324, 331-333) eines weiteren Programms mit höherer Prioritätsstufe (340, 345) des parallelen Bearbeitungsabschnitts (303) auf dem wenigstens einen Parallelprozessorkern (113, 114) unterbrochen werden.

6. Verfahren nach einem der Ansprüche 3 bis 5, wobei die wenigstens eine Prioritätsstufe (113, 114) der parallelen Bearbeitungsabschnitte (303) der jeweiligen Programme auf einen minimalen Wert gesetzt wird, wenn die Teilaufgaben (321-324, 331-333) der Programme sequentiell (365) in den entsprechenden Programmen ausführbar sind.

7. Verfahren nach Anspruch 1 bis 6, wobei die wenigstens eine Prioritätsstufe (340, 345) der parallelen Bearbeitungsabschnitte (303) der jeweiligen Programme zu Beginn eines neuen Programmzyklus auf den vorgegebenen Wert zurückgesetzt wird.

8. Verfahren nach Anspruch 6, wobei die wenigstens eine Prioritätsstufe (340, 345) der parallelen Bearbeitungsabschnitte (303) der jeweiligen Programme zu Beginn eines neuen Programmzyklus erhöht wird, wenn die Teilaufgaben (321-324, 331-333) vorher sequentiell (365) in den zugeordneten Programmen ausgeführt wurden.

9. Speicherprogrammierbare Steuerung (100), umfassend:

eine Kommunikationsschnittstelle (130), zum Einlesen von Sensordaten und zum Ausgeben von Aktordaten,

eine Datenverarbeitungseinrichtung (110), die wenigstens einen Parallelprozessorkern (113, 114) zum Ausführen einer Steuerungsaufgabe umfasst, um aus den Sensordaten Aktordaten zu erzeugen,

eine Steuerungseinrichtung (120), die einen Datenstrukturspeicher 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 die Steuerungseinrichtung (120) einen Zwischenergebnisspeicher umfasst, auf dem Zwischenergebnisse von Berechnungen gesichert werden, wobei die Steuerungsaufgabe mehrere ausführbare Programme umfasst,

wobei wenigstens zwei Programme der Steuerungsaufgabe jeweils wenigstens einen parallelen Bearbeitungsabschnitt (303) mit einem Arbeitspaket (305, 310) aufweisen, und das Arbeitspaket (305, 310) mehrere Teilaufgaben (321-324, 331-333) umfasst, wobei den parallelen Bearbeitungsabschnitten (303) in den jeweiligen Programmen eine Priorität mit der vorgegebenen Prioritätsstufe (340, 345) zugewiesen ist,

wobei der Prioritätenverwalter (350) ausgebildet ist, die jeweiligen Prioritätsstufen (340, 345) in die Datenstruktur (355) im Datenstrukturspeicher einzufügen, sobald ein Ausführen des Programms 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) zu setzen,

wobei der wenigstens eine Parallelprozessorkern (113, 114) ausgebildet ist, die Datenstruktur (355) im Datenstrukturspeicher auf Einträge der Prioritätsstufen (340, 345) zu prüfen, und sofern Einträge vorhanden sind, Teilaufgaben (321-324, 331-333) aus dem Arbeitspaket (305, 310) des entsprechenden Programms abzuarbeiten, dessen Prioritätsstufe (340, 345) an erster Stelle der Einträge in der Datenstruktur (355) steht,

wobei die Steuerungseinrichtung (120) ausgebildet ist, während eines Programmzyklus einen zu erwartenden Rechendurchsatz der Steuerungsaufgabe laufend zu bewerten, und

wobei der Prioritätenverwalter (350) ausgelegt ist, wenigstens eine der Prioritätsstufen (340, 345) der parallelen Bearbeitungsabschnitte (303) der jeweiligen Programme dynamisch anzupassen, wenn sich der Rechendurchsatz der Steuerungsaufgabe dadurch erhöht.

10. Speicherprogrammierbare Steuerung nach Anspruch 9, wobei der Prioritätenverwalter ausgebildet ist, die Anpassung der wenigstens einen Prioritätsstufe (340, 345) der parallelen Bearbeitungsabschnitte (303) der jeweiligen Programme bei einer Unterbrechung (366) der Abarbeitung der Teilaufgaben (321-324, 331-333) des Programms vorzunehmen.

11. Speicherprogrammierbare Steuerung nach Anspruch 10, wobei der Prioritätenverwalter (350) ausgelegt ist, die Anpassung der wenigstens einen Prioritätsstufe (340, 345) der parallelen Bearbeitungsabschnitte (303) der jeweiligen Programme vorzunehmen, wenn die Teilaufgaben (321-324, 331-333) aufgrund von anderen Teilaufgaben (321-324, 331-333) eines weiteren Programms mit höherer Prioritätsstufe (340, 345) auf dem wenigstens einen Parallelprozessorkern (113, 114) unterbrochen werden.

12. Speicherprogrammierbare Steuerung nach Anspruch 9 bis 11, wobei der Prioritätenverwalter (350) ausgebildet ist, bei der Anpassung der wenigstens einen Prioritätsstufe (340, 345) der parallelen Bearbeitungsabschnitte (303) der jeweiligen Programme eine zusätzliche Bedingung, insbesondere eine Echtzeitbedingung der Steuerungsaufgabe, zu berücksichtigen.

13. Speicherprogrammierbare Steuerung nach Anspruch 10 bis 12, wobei der Prioritätenverwalter (350) ausgebildet ist, die wenigstens eine Prioritätsstufe (340, 345) der parallelen Bearbeitungsabschnitte (303) der jeweiligen Programme um einen Faktor zu reduzieren, wenn die Teilaufgaben (321-324, 331-333) des Programms aufgrund von anderen Teilaufgaben (321-324, 331-333) eines weiteren Programms mit höherer Prioritätsstufe (340, 345) auf dem wenigstens einen Parallelprozessorkern (113, 114) unterbrochen werden.

14. Speicherprogrammierbare Steuerung nach Anspruch 10 bis 13, wobei der Prioritätenverwalter (350) ausgelegt ist, die wenigstens eine Prioritätsstufe (340, 345) der parallelen Bearbeitungsabschnitte (303) der jeweiligen Programme auf einen minimalen Wert zu setzen, wenn die Teilaufgaben (321-324, 331-333) der Programme sequentiell (365) in den entsprechenden Programmen ausführbar sind.

15. Speicherprogrammierbare Steuerung nach Anspruch 9 bis 14, wobei der Prioritätenverwalter (350) ausgebildet ist, die wenigstens eine Prioritätsstufe (340, 345) der parallelen Bearbeitungsabschnitte (303) der jeweiligen Programme zu Beginn eines neuen Programmzyklus auf den vorgegebenen Wert zurückzusetzen.

16. Speicherprogrammierbare Steuerung nach Anspruch 14, wobei der Prioritätenverwalter (350) ausgelegt ist, die wenigstens eine Prioritätsstufe (340, 345) der parallelen Bearbeitungsabschnitte (303) der jeweiligen Programme zu Beginn eines neuen Programmzyklus zu erhöhen, wenn die Teilaufgaben (321-324, 331-333) der Programme vorher sequentiell (365) in den zugeordneten Programmen ausgeführt wurden.

17. Speicherprogrammierbare Steuerung nach Anspruch 9 bis 16, wobei der Prioritätenverwalter (350) ausgebildet ist, den Eintrag der Prioritätsstufe (340, 345) in der Datenstruktur (350) im Datenstrukturspeicher gemäß der vorgenommenen Anpassung der wenigstens einen Prioritätsstufe (340, 345) abzuändern, und wobei der Prioritätenverwalter (350) ausgebildet ist den wenigstens einen Parallelprozessorkern (113, 114) über die Änderung des Eintrags der Prioritätsstufe (340, 345) in der Datenstruktur (355) zu informieren.

18. Speicherprogrammierbare Steuerung nach Anspruch 9 bis 17, wobei der Prioritätenverwalter (350) ausgebildet ist, auf mehreren Programmen der Steuerungsaufgabe verteilt ausführbar zu sein.

Es folgen 4 Seiten Zeichnungen

Anhängende Zeichnungen

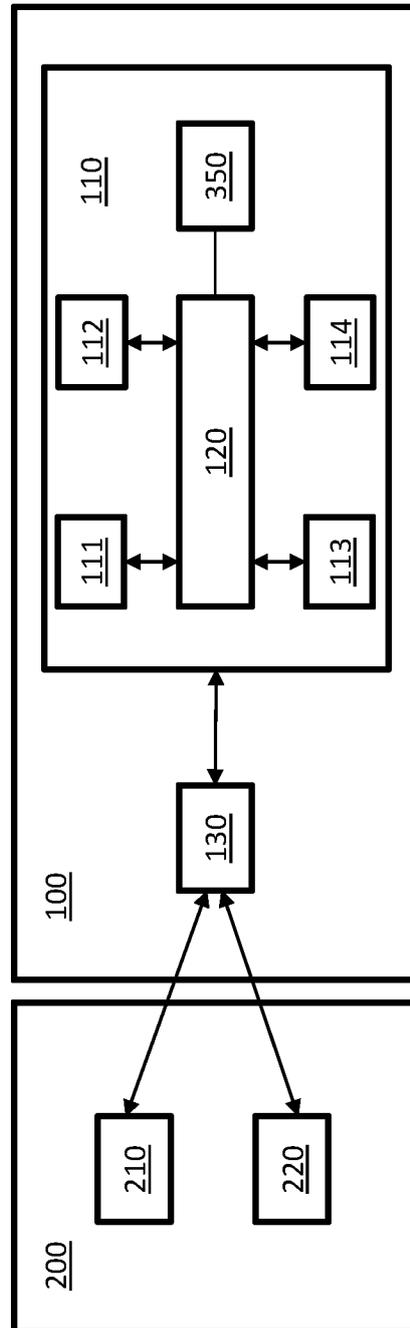


Fig. 1

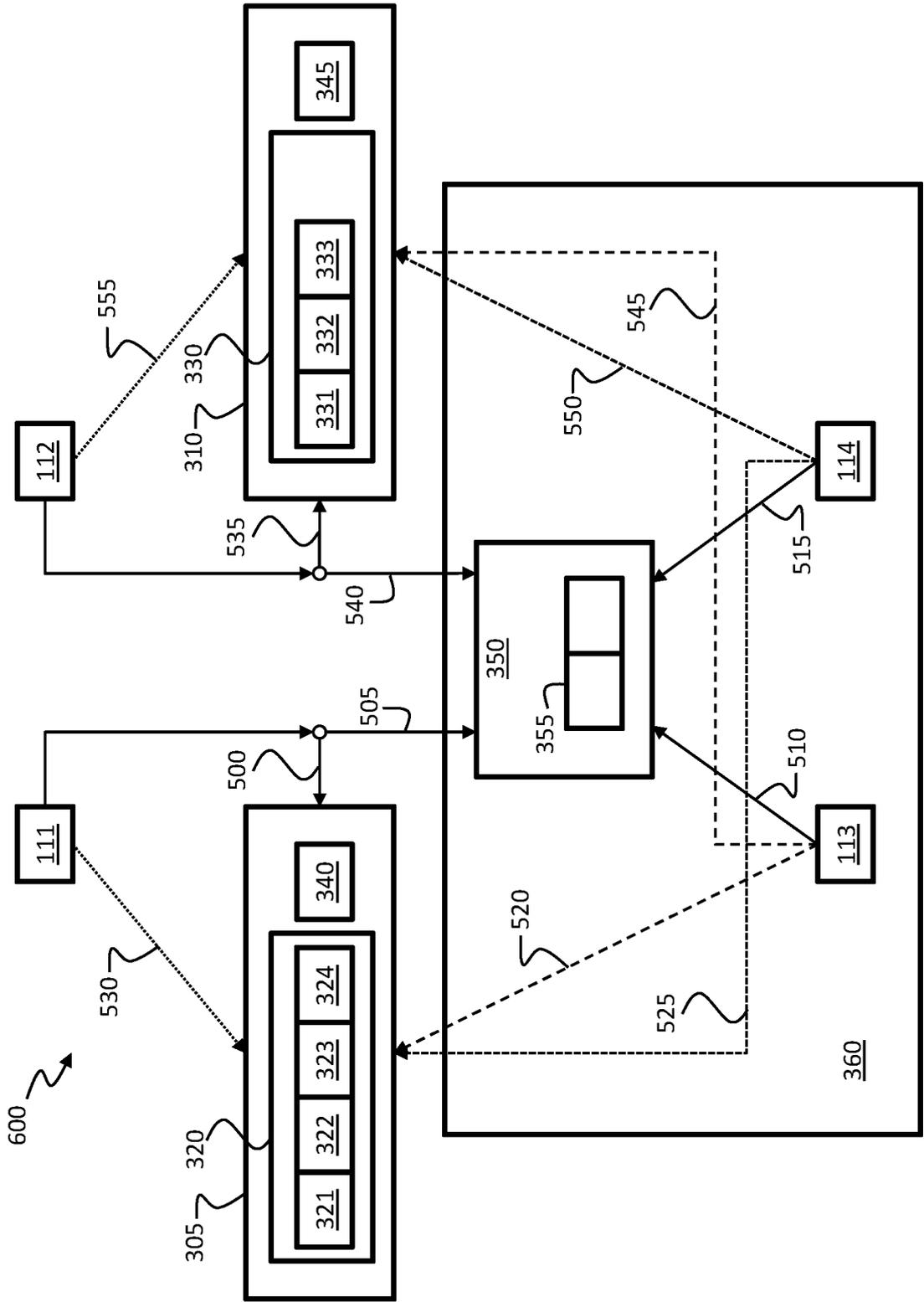


Fig. 2

Fig. 3

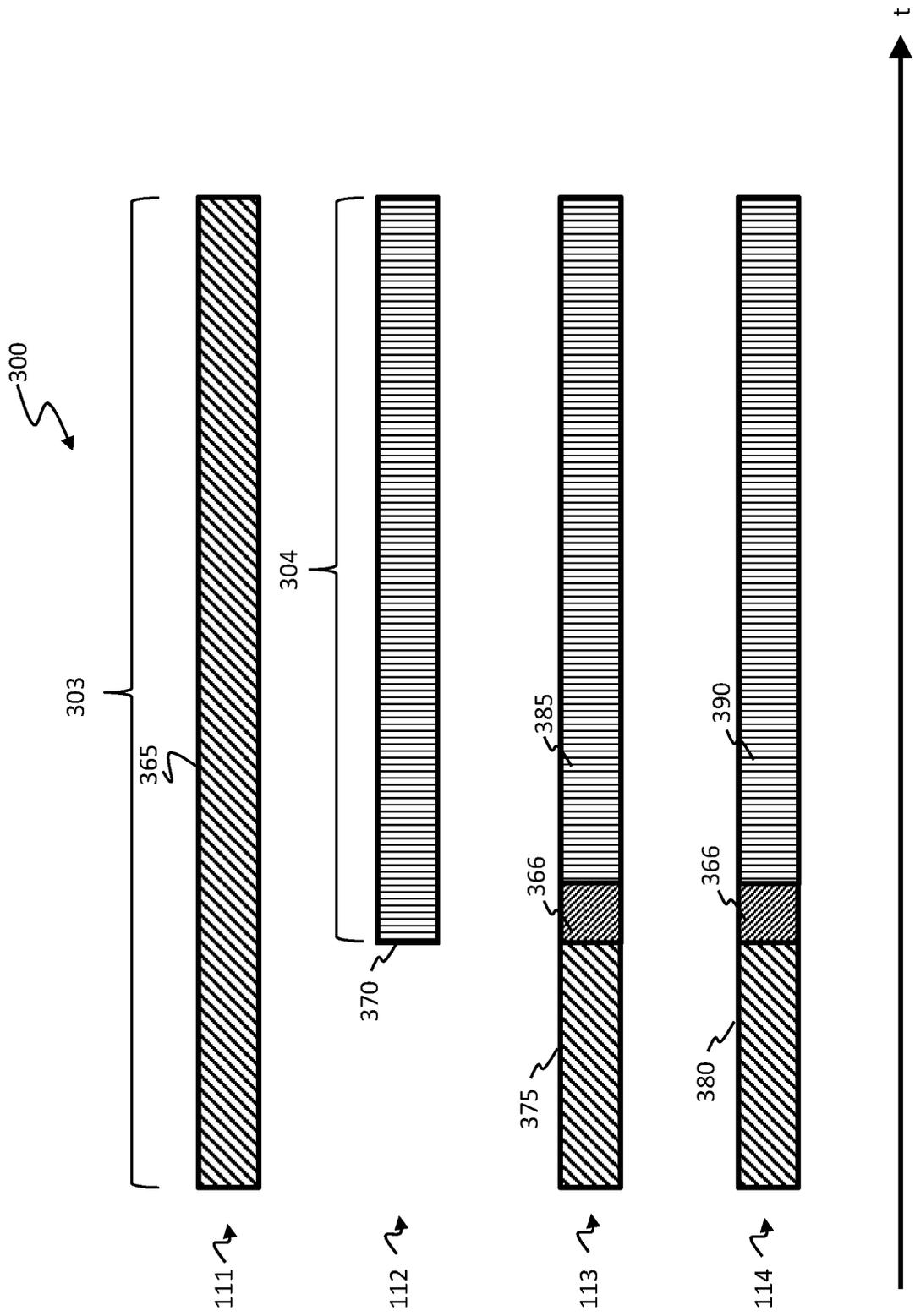


Fig. 4

