(19) Deutsches Patent- und Markenamt

(10) **DE 10 2023 208 085 A1** 2025.02.27

(12) # Offenlegungsschrift

(21) Aktenzeichen: **10 2023 208 085.0**
(22) Anmeldetag: **24.08.2023**
(43) Offenlegungstag: **27.02.2025**

(51) Int Cl.: ***H04L 47/6295*** (2022.01)
G06F 12/1072 (2016.01)
H04L 49/103 (2022.01)
H04L 49/901 (2022.01)
H04L 49/9005 (2022.01)
G06F 12/14 (2006.01)
G06F 12/12 (2016.01)
G06F 12/0891 (2016.01)

(71) Anmelder:
Robert Bosch Gesellschaft mit beschränkter Haftung, 70469 Stuttgart, DE

(72) Erfinder:
Le Meunier, Stephan, 74357 Bönnigheim, DE;
Mueller, Christoph, 17489 Greifswald, DE

(56) Ermittelter Stand der Technik:

US	**2003 / 0 037 178**	**A1**
US	**2011 / 0 239 219**	**A1**
US	**2023 / 0 138 522**	**A1**

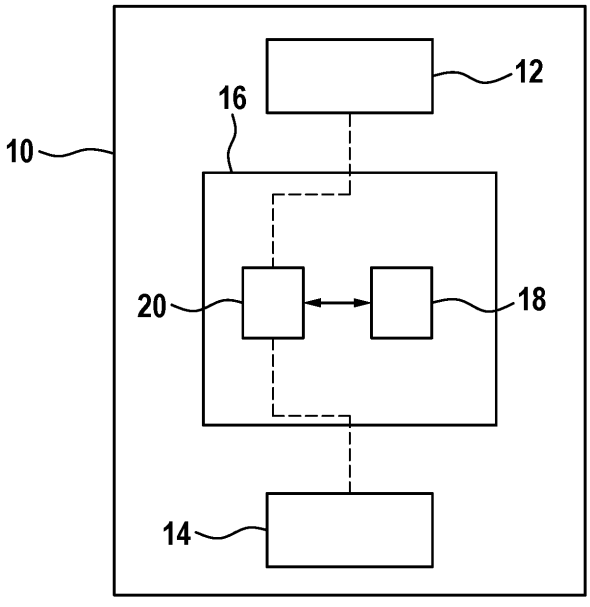
Prüfungsantrag gemäß § 44 PatG ist gestellt.

Rechercheantrag gemäß § 43 PatG ist gestellt.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **Verfahren zum Übertragen einer Nachricht mittels eines von zumindest zwei Recheneinheiten verwendeten Arbeitsspeichers**

(57) Zusammenfassung: Die Erfindung betrifft ein Verfahren zum Übertragen einer Nachricht mittels eines von zumindest zwei Recheneinheiten (12, 14a, 14b, 14c, 26, 28) verwendeten Arbeitsspeichers (16a, 16b, 16c),
wobei jede Warteschlange (20a, 20b, 20c) mehrere Plätze für Indizes umfasst,
wobei die Indizes auf Nachrichten in dem Nachrichtenpool (18) verweisen,
wobei eine erste Recheneinheit (12) und eine zweite Recheneinheit (14a, 14b und 14c) der zumindest zwei Recheneinheiten (12, 14a, 14b, 14c, 26, 28) auf den Arbeitsspeicher (16a, 16b und 16c) zugreifen,
wobei das Verfahren die folgenden Schritte umfasst:
- Allozieren eines Speicherplatzes (S10);
- Schreiben einer Nachricht (S12);
- Indizieren der Nachricht und Schreiben eines Indexes (S14) in einen Platz in der Warteschlange (20a, 20b, 20c);
- Auslesen des Indexes (S16);
- Ermitteln der Nachricht (S18) unter Verwendung des Indexes;
- Verwenden der Nachricht (S20) durch die zweite Recheneinheit (14a, 14b, 14c).



Beschreibung

[0001] Die Erfindung liegt auf dem Gebiet der Interprozessorkommunikation.

[0002] Interprozessorkommunikation (IPC) bezieht sich auf den Austausch von Informationen, Daten und Signalen zwischen verschiedenen Prozessen oder Prozessoren in einem Computersystem.
Sie ermöglicht die Zusammenarbeit, Koordination und den Datenaustausch zwischen den laufenden Prozessen, die auf verschiedenen Prozessoren oder Kernen eines Mehrprozessorsystems oder eines verteilten Systems ausgeführt werden.

[0003] IPC ist unabhängig davon, ob sie auf demselben System oder auf verschiedenen Systemen ausgeführt werden. Die IPC kann grundsätzlich sowohl synchron (blockierend) als auch asynchron (nicht blockierend) sein.

[0004] Bekannt sind verschiedene Mechanismen und Methoden zur Interprozessorkommunikation.

[0005] Bei einem Shared Memory (gemeinsamer Speicher) Mechanismus teilen sich die beteiligten Prozesse einen gemeinsamen Speicherbereich, auf den sie lesend und schreibend zugreifen können. Durch das Schreiben und Lesen von Daten in den gemeinsamen Speicher können die Prozesse miteinander kommunizieren und Informationen austauschen.

[0006] Bei einem Mechanismus mittels Nachrichtenübermittlung kommunizieren die Prozesse, indem sie Nachrichten senden und empfangen. Es gibt zwei grundlegende Ansätze: synchrones Senden, bei dem der sendende Prozess blockiert, bis die Nachricht empfangen wurde, und asynchrones Senden, bei dem der sendende Prozess die Nachricht sendet und seine Ausführung ohne Blockierung fortsetzt.

[0007] Pipes (Rohre) und FIFOs (First-In-First-Out) sind ebenfalls Mechanismen zur Kommunikation zwischen Prozessen. Pipes sind unidirektional und ermöglichen die Datenübertragung zwischen einem schreibenden und einem lesenden Prozess. FIFOs sind ähnlich, ermöglichen jedoch die Kommunikation zwischen mehreren Prozessen. Pipes sind dabei Linux/Unix-spezifische Softwareimplementierungen einer FIFO, um IPC zwischen zwei Prozesse zu ermöglichen. Eine FIFO für IPC kann jedoch auch im Dateisystem in Form einer Datei angelegt werden.

[0008] Die Erfindung betrifft im Besonderen eine IPC mit Shared Memory. Bekannte Ausführungsformen sind jedoch zumeist auf ein Betriebssystem beschränkt. Ein heterogenes Gesamtsystem mit unterschiedlichen Betriebssystemen und Hardware-Architekturen wird in der Regel nicht unterstützt. Ferner benötigen derartige IPC einen Synchronisationsmechanismus, um die Datenintegrität zu gewährleisten. Dies erfolgt üblicherweise mit Mechanismen, wie Spinlock, Semaphore und Mutex, aber auch über das Scheduling des Betriebssystems.

[0009] In jedem Fall werden die Daten während der IPC aufwendig kopiert. Ferner kann dabei keine 1:n-Übertragung unterstützt werden. 1:n-Übertragung bedeutet, dass ein Sender mehreren Empfängern parallel seine Daten zur Verfügung stellt. Beschreiben Sender und Empfänger denselben Speicher, kann das Freedom-From-Interference bei mixed ASIL-Anwendungen erschwert sein. Ferner kann durch die Synchronisierung ein Teilnehmer den anderen in seiner Laufzeit ausbremsen, was eine unerwünschte Beeinträchtigung entsprechen kann.

[0010] Aufgabe der Erfindung ist es daher, einen Mechanismus vorzuschlagen, der kopierfrei und ohne Synchronisationsmechanismus eine Interprozessorkommunikation zwischen verschiedenen Prozessen und/oder Prozessoren ermöglicht.

[0011] Die Aufgabe wird gelöst durch die Gegenstände der unabhängigen Ansprüche.

Offenbarung der Erfindung

[0012] Gemäß einem ersten Aspekt der Erfindung wird diese Aufgabe gelöst durch ein, insbesondere computerimplementiertes, Verfahren zum Übertragen einer Nachricht mittels eines von zumindest zwei Recheneinheiten verwendeten Arbeitsspeichers. Der Arbeitsspeicher umfasst zumindest einen als Nachrichtenpool eingerichteten Speicherbereich. Jedem Nachrichtenpool ist zumindest eine Warteschlange zugeordnet. Jede Warteschlange umfasst wiederum mehrere Plätze für Indizes, die auf Nachrichten in dem Nachrichtenpool verweisen. Eine erste Recheneinheit und eine zweite Recheneinheit der zumindest zwei Recheneinheiten greifen auf den Arbeitsspeicher zu. Dazu umfasst das Verfahren die folgenden Schritte:

- Allozieren eines Speicherplatzes im Nachrichtenpool durch die erste Recheneinheit;
- Schreiben einer Nachricht in den allozierten Speicherplatz durch die erste Recheneinheit;
- Indizieren der Nachricht und Schreiben eines Indexes der Nachricht in einen Platz in der Warteschlange durch die erste Recheneinheit;
- Auslesen des Indexes aus der Warteschlange durch die zweite Recheneinheit;
- Ermitteln der Nachricht unter Verwendung des Indexes durch die zweite Recheneinheit;
- Verwenden der Nachricht durch die zweite Recheneinheit; und

- Prüfen auf eine Freigabe der Nachricht und Markieren des Platzes in der Warteschlange zur Wiederverwendung durch die erste Recheneinheit.

[0013] Im Kontext dieser Erfindung kann eine Recheneinheit eine physische oder eine virtuelle Recheneinheit sein. Insbesondere können zwei oder mehr Recheneinheiten Kerne einer CPU umfassen. Ferner können Recheneinheiten durch virtuelle Maschinen, GPUs, Mikroprozessoren oder Mikrocontroller realisiert sein.

[0014] Jede Warteschlange verbindet in jedem Fall zwei Recheneinheiten kommunikativ miteinander. Dabei ist eine der Recheneinheiten ein Sender und die andere Recheneinheit ist ein Empfänger der zu übertragenden Nachricht. Die zumindest eine Warteschlange ist damit ein Teil der Speicherverwaltung und jede Warteschlange ist mit einem Nachrichtenpool gekoppelt.

[0015] Jeder Nachrichtenpool umfasst eine Menge an Nachrichten oder Speicherplätzen, in die Nachrichten hineingeschrieben werden können. Eine Nachricht und ein Speicherplatz für eine Nachricht sind dabei zwei Seiten derselben Sache. Die Nachricht ist eine Information, die aus Bits besteht. Der Speicherplatz ist die physische Repräsentation ebendieser Nachricht, ein Bereich auf einer Hardware, in dem Bits volatil oder nicht-volatil erhalten bleiben. Die Begriffe Speicherplatz und Nachricht sind daher austauschbar.

[0016] Wenn die Recheneinheiten unterschiedliche Hardwarearchitekturen und/oder Betriebssysteme aufweisen, kann es vorkommen, dass der Arbeitsspeicher von den Recheneinheiten unterschiedlich angesprochen wird. Dieses Hindernis wird umgangen, indem die Übertragung der Nachricht von einer Recheneinheit in die andere über einen Index erfolgt.

[0017] Der Index verweist auf eine Position innerhalb einer Datenstruktur, insbesondere auf eine Position in einer Tabelle. Erfindungsgemäß ist an der Position, auf die der Index verweist, unter Verwendung einer Look-up-Table, ein Zeiger hinterlegt, der auf den Speicherort eines bestimmten Datenbereichs im Arbeitsspeicher zeigt, oder, unter Verwendung einer Konvertierungsfunktion, wird ein Zeiger errechnet, der ebenfalls auf den Speicherort eines bestimmten Datenbereichs im Arbeitsspeicher zeigt.

[0018] Der Zeiger ermöglicht den direkten Zugriff auf den Wert oder die Daten, die an einer bestimmten Speicheradresse im Arbeitsspeicher gespeichert sind. Er enthält insbesondere die Information über die Position oder die Adresse des Speicherbereichs, auf den er verweist.

[0019] Wenn die zweite Recheneinheit den Index nicht direkt verwenden kann, eben weil sie den Arbeitsspeicher anders ansteuert als die erste Recheneinheit, dann kann der Schritt des Ermittelns der Nachricht beispielsweise ein Zuordnen oder Übersetzen des Indexes unter Verwendung einer Look-up-Table oder einer Konvertierungsfunktion erfolgen, die die Position des mit dem Index gekennzeichneten Speicherplatzes für die zweite Recheneinheit übersetzt.

[0020] Gemäß der vorliegenden Erfindung wird der Index verwendet, um den Zugriff auf gemeinsam genutzte Ressourcen oder Datenstrukturen zu ermöglichen. Durch die Weitergabe des Indexes zwischen den Recheneinheiten können diese auf den gleichen Speicherbereich zugreifen und Daten austauschen, ohne dass eine Kopie der Daten erstellt werden muss. Im Zusammenhang mit dem Arbeitsspeicher ermöglicht die Verwendung des Indexes in einer Warteschlange die dynamische Allokation und Verwaltung von Speicherressourcen daher kopierfrei. Das Übertragen der Nachricht von der ersten an die zweite Recheneinheit umfasst also zumindest ein Bereitstellen der Nachricht und eines entsprechenden Indexes durch die erste Recheneinheit und ein Auslesen des Indexes sowie ein Ermitteln der bereitgestellten, indizierten Nachricht durch die zweite Recheneinheit. Das Verwenden der Nachricht kann bspw. ein Auslesen von Daten aus der Nachricht und/oder ein Weiterleiten der Nachricht umfassen. Insgesamt löst der vorgeschlagene Mechanismus damit die Aufgabe der Erfindung.

[0021] In einer Ausführungsform gehen dem Prüfen auf die Freigabe der Nachricht die folgenden Schritte voraus:

- Vergeben eines Nutzungswerts für den Index beim Schreiben des Indexes durch die erste Recheneinheit;

- Vergeben eines Freigabewerts für den Index nach der Nutzung der Nachricht durch die zweite Recheneinheit; und

- Übertragen des Freigabewerts von der zweiten Recheneinheit an die erste Recheneinheit.

[0022] Das Prüfen auf die Freigabe umfasst dabei einen Vergleich des Freigabewerts und des Nutzungswerts.

[0023] Für jede Nachricht kann vorzugsweise ein spezieller Speicherbereich für einen Nutzungsindikator reserviert sein, der angibt, ob die Nachricht noch benötigt wird, oder ob sie durch eine andere Nachricht ersetzt werden kann. Der Nutzungsindikator kann vorzugsweise ein Bit und damit die Werte „0“ für nicht genutzt" und „1“ für „genutzt“ umfassen. Die erste Recheneinheit gibt den indizierten Platz im Nachrichtenpool frei, indem sie den Nutzungsindi-

kator auf „nicht genutzt“ oder „0“ setzt. Gemäß dieser Ausführungsform kann sie dies insbesondere dann tun, wenn der Freigabewert und der Nutzungswert eines Indexes übereinstimmen. Nur nicht genutzte Nachrichten können für weitere Übertragungen alloziert und mit neuen Informationen überschrieben werden.

[0024] Wird eine Nachricht alloziert, wird ein Nutzungswert vergeben. Dieser kann in Ausführungsformen beispielsweise inkrementiert werden, sodass der Nutzungswert angibt, wie oft ein Index zur Nutzung übergeben wurde. Wenn ein Index beispielsweise an eine weitere Recheneinheit übergeben wurde, wird der Nutzungswert von 1 vergeben, oder der Nutzungswert von 0 wird inkrementiert auf 1.

[0025] Nach der Nutzung vergibt die zweite Recheneinheit den Freigabewert. Ein Freigabewert von 1 bedeutet, dass ein Nutzer der Nachricht (die zweite Recheneinheit) die Nachricht nicht länger verwendet. In einer Ausführungsform kann die zweite Recheneinheit Nutzungswert als Freigabewert übernehmen, sodass diese identisch sind und die Nachricht freigegeben wird.

[0026] Wenn der Nutzungswert und der Freigabewert übereinstimmen, bedeutet das also, dass kein Nutzer die Nachricht mehr verwendet. Sie kann daher als „nicht genutzt“ markiert und zur Allokation freigeben werden.

[0027] Das Übertragen des Freigabewerts von der zweiten Recheneinheit an die erste Recheneinheit kann insbesondere über den gemeinsam geteilten Arbeitsspeicher erfolgen.

[0028] In vorteilhafter Weise verhindert das Abgleichen des Nutzungswerts mit dem Freigabewert, dass es zu einem Mangel an Nachrichten kommt, da die erste Recheneinheit regelmäßig nach nicht länger genutzten Nachrichten suchen und diese für eine erneute Allokation freigeben kann.

[0029] In einer alternativen Ausführungsform kann die zweite Recheneinheit die Nachricht ihrerseits an eine weitere Recheneinheit weitergeben. Dazu wird das Verfahren um die folgenden Schritte vor dem Freigeben der Nachricht erweitert:

- Vergeben eines Nutzungswerts für den Index bei der Allokation der Nachricht durch die erste Recheneinheit;

- Indizieren der Nachricht und Schreiben des Indexes der Nachricht in einen Platz in einer weiteren Warteschlange durch die zweite Recheneinheit nach dem Verwenden der Nachricht durch die zweite Recheneinheit;

- Auslesen des Indexes aus der weiteren Warteschlange durch die weitere Recheneinheit;

- Ermitteln der Nachricht anhand des Indexes durch die weitere Recheneinheit;

- Verwenden der Nachricht durch die weitere Recheneinheit;

- Vergeben eines Freigabewerts für den Index nach der Nutzung der Nachricht durch die weitere Recheneinheit; und

- Übertragen des Freigabewerts von der weiteren Recheneinheit an die erste Recheneinheit.

[0030] Das Prüfen auf die Freigabe umfasst dabei einen Vergleich des Freigabewerts und des Nutzungswerts.

[0031] Im Grundsatz ist diese Ausführungsform ähnlich zu der Ausführungsform mit zwei Recheneinheiten. Sie unterscheiden sich dadurch voneinander, dass in dieser Ausführungsform der Freigabewert von der weiteren Recheneinheit vergeben wird.

[0032] Grundsätzlich könnte diese Ausführungsform auch mit beliebig vielen Recheneinheiten durchgeführt werden. Das Prinzip ist daher erweiterbar, indem der Index von einer Recheneinheit an die nächste Recheneinheit weitergegeben wird, die dann wiederum die Nachricht anhand des erhaltenen Indexes ermittelt und sie nutzt. Solange der Freigabewert nicht vergeben wird, gilt die Nachricht als genutzt und kann von beliebig vielen Recheneinheiten nacheinander verwendet werden. Erst die letzte Recheneinheit in einer Arbeitskette von Recheneinheiten vergibt den Freigabewert, sodass die Nachricht dann von der ersten Recheneinheit wieder als nicht-genutzt gekennzeichnet und für die Übertragung weiterer Nachrichten alloziert werden kann.

[0033] Das Übertragen des Freigabewerts von der weiteren Recheneinheit an die erste Recheneinheit kann insbesondere über den gemeinsam geteilten Arbeitsspeicher erfolgen.

[0034] In einer weiteren Ausführungsform umfasst das Indizieren der Nachricht ein Hinzufügen eines Zeitstempels zu der Nachricht. Das Prüfen auf die Freigabe kann dann ein Überprüfen des Zeitstempels der Nachricht umfassen, sodass die erste Recheneinheit den indizierten Platz in der Warteschlange freigibt, wenn das Überprüfen des Zeitstempels das Erreichen eines definierten Alters der Nachricht anzeigt.

[0035] Es ist möglich, dass Prozesse unterbrochen werden oder unerwartet stoppen. Insbesondere dann, wenn die erste Recheneinheit auf einen Freigabewert einer Nachricht wartet, würde dieser unter Umständen nicht vergeben werden, sodass eine Nachricht ohne die Vergabe des Freigabewerts nicht als „nicht genutzt“ gekennzeichnet werden könnte. Passiert das zu häufig, würde ein Datenleck

entstehen, bei dem der Arbeitsspeicher irgendwann vollständig belegt wäre, ohne dass eine neue Nachricht in den Nachrichtenpool geschrieben werden kann.

[0036] Das Vergeben und nachfolgende Überprüfen des Zeitstempels einer Nachricht löst dieses Problem, da nach dem Ablauf einer bestimmten Zeit, also wenn die Nachricht ein bestimmtes Alter erreicht hat, die Nachricht unabhängig von der Verwendung durch die zweite oder eine weitere Recheneinheit neu alloziert werden kann.

[0037] In einer weiteren Ausführungsform greift zumindest eine weitere zweite Recheneinheit auf den Arbeitsspeicher zu. Das Verfahren zwischen der ersten Recheneinheit und der zumindest einen weiteren zweiten Recheneinheit wird in dieser Ausführungsform jeweils mit einer eigenen Warteschlange durchgeführt.

[0038] Die zumindest eine weitere zweite Recheneinheit wird dabei genauso wie die „erste“ zweite Recheneinheit behandelt. Das bedeutet, alle zweiten Recheneinheiten sind nebengeordnet und empfangen einen Index von der ersten Recheneinheit. In jeder Paarung aus erster Recheneinheit und zweiter Recheneinheit wird eine Warteschlange verwendet, um die Indizes zu übertragen. Dementsprechend werden so viele Warteschlangen benötigt, wie zweite Recheneinheiten Indizes empfangen können.

[0039] In einer weiteren Ausführungsform ist jede Warteschlange demselben Nachrichtenpool zugeordnet.

[0040] Die Zuordnung gleicher Nachrichtenpools bewirkt in vorteilhafter Weise, dass auch später erst zugeschaltete Recheneinheiten mit einem im Voraus zugewiesenen Speicherbereich im mit der ersten Recheneinheit geteilten Arbeitsspeicher Nachrichten von der ersten Recheneinheit verwenden können. Dies kann insbesondere auch als late joining service bezeichnet werden.

[0041] Alternativ kann diese Ausführungsform auch dazu genutzt werden, die Arbeitslast einer Recheneinheit auf andere Recheneinheiten auszulagern. Hat beispielsweise eine Recheneinheit eine zu hohe Auslastung, kann sich eine weitere zweite Recheneinheit hinzuschalten, die anstelle der eigentlich vorgesehenen Recheneinheit die Nachrichten der ersten Recheneinheit verwendet.

[0042] Als weitere Alternative ermöglicht diese Ausführungsform ein kopierfreies Verteilen einer Nachricht an mehrere Recheneinheiten parallel. Der Prozess wird dann auch als Multiplexen oder Multicasting bezeichnet.

[0043] Wenn mehrere Recheneinheiten dieselbe Nachricht erwarten, kann diese Ausführungsform verwendet werden, um die Übertragung der Nachrichten an mehrere Recheneinheiten zu beschleunigen. Da der Prozess kopierfrei abläuft, wird die Zeit zwischen dem Schreiben der Nachricht und ihrer Verwendung durch die empfangende Recheneinheit minimiert.

[0044] In dieser Ausführungsform kann insbesondere das Vergeben des Freigabewerts von jeder zweiten Recheneinheit vorgenommen werden. Die erste Recheneinheit prüft dann, ob die Summe der Freigabewerte dem Nutzungswert entspricht. Wenn beispielsweise die erste Recheneinheit einen Index an drei zweite Recheneinheiten verteilt hat, kann sie den Nutzungswert auf 3 setzen. Insbesondere wird der Nutzungswert nach der Allokation der Nachricht und vor dem Versand an die zweiten Recheneinheiten gesetzt werden.

[0045] Die drei zweiten Recheneinheiten vergeben nach dem Verwenden der Nachricht den Freigabewert, sodass die erste Recheneinheit die vergebenen Freigabewerte zählen kann. Wenn die erste Recheneinheit drei Freigabewerte zählt, stimmen Nutzungswert und Freigabewerte überein, sodass die erste Recheneinheit erkennt, dass die Nachricht von keiner Recheneinheit mehr verwendet wird. Nun kann sie die Nachricht als „nicht genutzt“ kennzeichnen und so zur Allokation freigeben.

[0046] In vorteilhafter Weise kann diese Ausführungsform genutzt werden, um eine Nachricht kopierfrei an mehrere Recheneinheiten parallel zu übertragen.

[0047] In einer Ausführungsform kann ein System mehrere Recheneinheiten umfassen, die teilweise parallel und teilweise in Reihe angeordnet sind. Beispielsweise kann eine erste Recheneinheit eine Nachricht für drei zweite Recheneinheiten bereitstellen. Von diesen drei zweiten Recheneinheiten können zwei die Nachricht an weitere Recheneinheiten weitergeben und so fort. Durch das vorgeschlagene Verfahren lassen sich beliebig viele Recheneinheiten parallel und/oder in Reihe anordnen, von denen jede auf einen geteilten Arbeitsspeicher zugreift.

[0048] In einer Ausführungsform werden die Recheneinheiten mit unterschiedlichen Betriebssystemen betrieben.

[0049] Diese Ausführungsform ermöglicht das Betreiben von sogenannten System-Ona-Chip-Anwendungen (SoCs). Bei SoCs werden unterschiedliche Recheneinheiten in einem Chip verbaut, die sich dann gemeinsame Ressourcen, darunter insbesondere der Arbeitsspeicher, teilen. Beispielsweise kann eine Recheneinheit ein Mikroprozessor

und eine andere Recheneinheit ein Mikrocontroller sein.

[0050] Ein Mikrocontroller und ein Mikroprozessor sind zwei unterschiedliche Arten von integrierten Schaltkreisen, die in elektronischen Geräten zum Einsatz kommen. Obwohl sie ähnliche Aufgaben erfüllen können, gibt es grundlegende Unterschiede in ihrer Struktur, Architektur und Funktionalität.

[0051] Ein Mikrocontroller ist ein einzelner integrierter Schaltkreis, der einen Prozessor (CPU), Speicher (ROM, RAM) und verschiedene Ein- und Ausgabeschnittstellen vereint. Es handelt sich also um eine vollständige Einheit, die speziell für eine bestimmte Anwendung oder einen bestimmten Aufgabenbereich entwickelt wurde. Ein Mikrocontroller ist in der Regel darauf ausgelegt, eigenständig Aufgaben auszuführen. Er enthält den Programmcode und die Daten, die beispielsweise zur Steuerung einer bestimmten Anwendung oder eines Systems erforderlich sind. Mikrocontroller finden sich häufig in eingebetteten Systemen, wie in Haushaltsgeräten, Automobilen, industriellen Steuerungen usw. Sie bieten eine einfache Programmierbarkeit und sind darauf spezialisiert, spezifische Aufgaben effizient und deterministisch zu erledigen.

[0052] Im Gegensatz dazu besteht ein Mikroprozessor nur aus der CPU und verfügt nicht über die integrierten Speicher oder Peripheriegeräte eines Mikrocontrollers. Mikroprozessoren sind so konzipiert, dass sie in Verbindung mit externen Komponenten verwendet werden, um ein vollständiges System zu bilden. Ein Mikroprozessor ist damit flexibler und kann in unterschiedlichen Anwendungen eingesetzt werden. Mikroprozessoren werden häufig in allgemeinen Computern, Laptops, Smartphones und anderen Geräten verwendet, die eine hohe Rechenleistung und Flexibilität erfordern. Sie können vielfältige Aufgaben ausführen, sind aber möglicherweise nicht so effizient in spezifischen Anwendungen wie Mikrocontroller.

[0053] In SoCs werden die Vorteile beider Recheneinheiten in ein einem Chipdesign vereint, was jedoch eine Handhabung des gemeinsamen Arbeitsspeichers durch unterschiedliche Betriebssysteme nötig macht.

[0054] In einer Ausführungsform entspricht der Abstand der Speicherplätze für die Indizes in der Warteschlange mindestens der größten Datenbusweite der auf den Arbeitsspeicher zugreifenden Recheneinheiten.

[0055] Die in dieser Ausführungsform gesetzte untere Grenze für den Abstand der Speicherplätze bewirkt in vorteilhafter Weise, dass die Recheneinheiten die Indizes trotz unterschiedlicher Datenbusbreiten aus dem Arbeitsspeicher konsistent auslesen können, da die Auslesebreite (Speicherausrichtung) an die größte Datenbusbreite angepasst wird. Bei kleineren Speicherauslesebreiten könnten Recheneinheiten mit größeren Datenbusbreiten die Indizes ansonsten gegebenenfalls nicht konsistent auslesen.

[0056] Wenn beispielsweise die Datenbusweite einer Recheneinheit 64 Bit beträgt und der Abstand der Speicherplätze im Arbeitsspeicher jedoch 32 Bit ist, dann würde die Recheneinheit auf den Index überlappend zugreifen, da sich in den ausgelesenen Bits Teile zweier Indizes befänden.

[0057] Wenn beispielsweise ein 64-Bit System auf einen 32-Bit Index mit einer 32-Bit Speicherausrichtung zugreift, dann werden bei einem Speicherzugriff zwei Indizes gelesen oder geschrieben. Die richtige Lokalisierung des Index im Arbeitsspeicher wird von der Hardware gewährleistet. Da aber auf Hardwareebene bei einem 64-Bit System immer nur 64-Bit gelesen oder geschrieben werden können, würde immer auf zwei 32-Bit Indexe zugegriffen. Weil bei einem Schreibvorgang aber nur ein Index geschrieben werden soll, entsteht für den zweiten Index eine potenzielle Wettlaufsituation mit einer anderen Recheneinheit, die auf den zweiten Index zugreift.

[0058] In einer weiteren Ausführungsform entspricht der allozierte Speicherplatz für einen Index höchstens der kleinsten Datenbusweite der auf den Arbeitsspeicher zugreifenden Recheneinheiten.

[0059] Diese Ausführungsform bewirkt in vorteilhafter Weise, dass auch Recheneinheiten im System mit geringer Datenbusbreite die Indizes konsistent auslesen können.

[0060] Wenn beispielsweise eine Recheneinheit mit einer Datenbusbreite von 32 Bit einen Index mit 64 Bit auslesen würde, müsste sie dies in zwei 32-Bit-Schritten tun. Da Indizes jedoch immer als Ganzes und konsistent ausgelesen werden sollten, ist es vorteilhaft, die Größe der Indizes von vornherein an die kleinste Datenbusbreite anzupassen.

[0061] Insgesamt stellt es kein Problem für das Auslesen der Indizes dar, wenn die Abstände zwischen den Indizes größer sind als die Indizes selbst. 32 Bit große Indizes können beispielsweise in einem 64 Bit großen Abstand gespeichert sein. Eine Recheneinheit mit 64 Bit Datenbusbreite würde einen Index mit 32 Bit und 32 weitere leere oder ungenutzte Bit auslesen. Dies beeinträchtigt den Auslesevorgang nicht, führt jedoch gegebenenfalls zu einem höheren Speicherbedarf für die Nachrichten. Eine potenzielle Wettlaufsituation mit anderen Recheneinheiten, die auf den gleichen oder auch auf den daneben liegenden Wert zugreifen, wird dadurch unterbunden.

[0062] In einer Ausführungsform ist die Warteschlange als Ringpuffer ausgebildet.

[0063] Ein Ringpuffer ist eine Datenstruktur, die zur Verwaltung von Daten in einer festen Größe verwendet wird. Er basiert auf dem Konzept einer zyklischen Anordnung von Speicherplätzen, die wie ein Ring organisiert sind. Der Ringpuffer ist besonders nützlich in Situationen, in denen eine kontinuierliche Erfassung oder Verarbeitung von Daten erfolgen soll, während ältere Daten verworfen werden.

[0064] Die Funktion eines Ringpuffers besteht darin, Daten zu speichern, zu lesen und zu verwalten. Dabei wird ein festgelegter Speicherbereich zyklisch genutzt, wodurch ein kontinuierlicher Datenfluss ermöglicht wird. Der Ringpuffer hat eine feste Kapazität, die im Voraus definiert wird.

[0065] Der Ringpuffer verwendet zwei Indizes, den Leseindex (auch als „head“ bezeichnet) und den Schreibindex (auch als „tail“ bezeichnet), um den Zustand der Datenstruktur zu verfolgen. Der Schreibindex zeigt auf die Position, an der neue Daten eingefügt werden, während der Leseindex auf die Position zeigt, an der die Daten gelesen werden.

[0066] In einer Ausführungsform entspricht der Abstand der Lese- und Schreibindizes mindestens der größten Datenbusweite der auf den Arbeitsspeicher zugreifenden Recheneinheit. Ferner entspricht die Datenbusweite der kleinsten Datenbusweite der auf den Arbeitsspeicher zugreifenden Recheneinheit.

[0067] In vorteilhafter Weise wird dadurch eine potenzielle Wettlaufsituation mit einer anderen Recheneinheit unterbunden, die auf der gleichen oder auch auf dem daneben liegenden Wert zugreift.

[0068] Neue Daten werden an der aktuellen Position des Schreibindex eingefügt. Der Schreibindex wird dann auf die nächste Position im Ringpuffer bewegt. Wenn der Schreibindex das Ende des Puffers erreicht, wird er auf den Anfang zurückgesetzt, wodurch die zyklische Natur des Puffers ermöglicht wird. Wenn der Puffer voll ist, das bedeutet, wenn der Schreibindex den Leseindex eingeholt hat, wird der Schreibvorgang ausgesetzt, bis wieder Platz für neue Daten vorhanden ist.

[0069] Wenn die Daten bzw. der Nachrichtenindex überschrieben werden, kann dieser Nachrichtenindex von zweiten Recheneinheiten auch nicht mehr ausgelesen werden. In Folge kann diese Nachricht von der zweiten Recheneinheiten auch nicht mehr freigegeben werden. Dies entspricht einem Speicherleck bei den Nachrichten. Die Anzahl der Nachrichten im Nachrichtenpool und die Anzahl der Einträge im Ringpuffer können grundsätzlich jedoch unterschiedlich sein.

[0070] Die Daten werden an der aktuellen Position des Leseindexes abgerufen und gelesen. Der Leseindex wird dann auf die nächste Position im Ringpuffer bewegt. Wenn der Leseindex das Ende des Puffers erreicht, wird er ebenfalls auf den Anfang zurückgesetzt. Hat der Leseindex den Schreibindex eingeholt, ist der Puffer leer.

[0071] Ein wichtiger Vorteil eines Ringpuffers besteht darin, dass er wenig Speicherplatz benötigt. Da der Puffer eine feste Größe hat und ältere Daten überschrieben werden können, ist er besonders gut geeignet, wenn die aktuellen Daten wichtiger sind als die älteren Daten. Der Ringpuffer wird vorzugsweise in verschiedenen Anwendungen eingesetzt, wie in der Audioverarbeitung, der Sensorik und vielen anderen Situationen, in denen eine kontinuierliche Erfassung und Verarbeitung von Daten erforderlich ist.

[0072] In einer Ausführungsform ist die Nachricht ein Parameter eines Systems, insbesondere eines Kraftfahrzeugs. Der Parameter kann beispielsweise, ohne darauf beschränkt zu sein, die aktuelle Geschwindigkeit, der Lenkwinkel, der Ladezustand einer Fahrzeugbatterie oder eine ähnliche Größe sein.

[0073] Eine konkrete Anwendung könnte beispielsweise darin bestehen, dass die Geschwindigkeit mittels eines Mikrocontrollers kontinuierlich erfasst und in einen Arbeitsspeicher geschrieben wird. Mikroprozessoren entscheiden dann, abhängig von der jeweiligen Geschwindigkeit, ob und wenn ja, wie der Messwert weiterverarbeitet wird. Das Weiterverarbeiten kann beispielsweise ein Abspeichern der Geschwindigkeit in einem persistenten Speicher umfassen, sodass eine Historie der gefahrenen Geschwindigkeit entsteht.

[0074] In einem weiteren Aspekt betrifft die Erfindung ein Computerprogramm mit Programmcode, um ein computerimplementiertes Verfahren zum Übertragen einer Nachricht mittels eines von zumindest zwei Recheneinheiten verwendeten Arbeitsspeichers wie vorangehend beschrieben auszuführen, wenn das Computerprogramm auf einem Computer ausgeführt wird.

[0075] In einem weiteren Aspekt betrifft die Erfindung einen computerlesbaren Datenträger mit Programmcode eines Computerprogramms, um ein Verfahren zum Übertragen einer Nachricht mittels eines von zumindest zwei Recheneinheiten verwendeten Arbeitsspeichers wie voranstehend beschrieben auszuführen, wenn das Computerprogramm auf einem Computer ausgeführt wird.

[0076] In einem weiteren Aspekt betrifft die Erfindung ein System zum Übertragen einer Nachricht mittels eines von zumindest zwei Recheneinheiten verwendeten Arbeitsspeichers, wobei das System umfasst:

- einen Arbeitsspeicher,
- eine erste Recheneinheit, und
- eine zweite Recheneinheit,

wobei das System Mittel umfasst, um das Verfahren wie vorangehend beschrieben auszuführen.

[0077] Insgesamt werden somit ein Verfahren, ein Computerprogramm, ein computerlesbarer Datenträger sowie eine Vorrichtung angegeben.

[0078] Die beschriebenen Ausgestaltungen und Weiterbildungen lassen sich beliebig miteinander kombinieren.

[0079] Weitere mögliche Ausgestaltungen, Weiterbildungen und Implementierungen der Erfindung umfassen auch nicht explizit genannte Kombinationen von zuvor oder im Folgenden bezüglich der Ausführungsbeispiele beschriebenen Merkmalen der Erfindung.

Kurze Beschreibung der Zeichnungen

[0080] Die beiliegenden Zeichnungen sollen ein weiteres Verständnis der Ausführungsformen der Erfindung vermitteln. Sie veranschaulichen Ausführungsformen und dienen im Zusammenhang mit der Beschreibung der Erklärung von Prinzipien und Konzepten der Erfindung.

[0081] Andere Ausführungsformen und viele der genannten Vorteile ergeben sich im Hinblick auf die Zeichnungen. Die dargestellten Elemente der Zeichnungen sind nicht notwendigerweise maßstabsgetreu zueinander gezeigt.

[0082] Es zeigen:

Fig. 1 eine schematische Darstellung des vorgeschlagenen Verfahrens gemäß einer Ausführungsform;

Fig. 2 zeigt schematisch den Aufbau eines Systems umfassend zwei Recheneinheiten gemäß einer Ausführungsform;

Fig. 3 zeigt schematisch drei in nebeneinander positionierten Speicherbereichen gespeicherte Nachrichten;

Fig. 4 zeigt schematisch eine weitere Ausführungsform mit einem System mit vier Recheneinheiten, die über drei Arbeitsspeicher miteinander nebenläufig verbunden sind;

Fig. 5 zeigt schematisch eine weitere Ausführungsform mit einem System mit vier Recheneinheiten, die über einen gemeinsamen Arbeitsspeicher miteinander nebenläufig verbunden sind;

Fig. 6 zeigt schematisch eine weitere Ausführungsform mit einem System mit vier Recheneinheiten, die über einen gemeinsamen Arbeitsspeicher miteinander sequenziell verbunden sind.

[0083] In den Figuren der Zeichnungen bezeichnen gleiche Bezugszeichen gleiche oder funktionsgleiche Elemente, Bauteile oder Komponenten, soweit nichts Gegenteiliges angegeben ist.

[0084] **Fig.** 1 zeigt schematisch den Ablauf eines Verfahrens gemäß einer Ausführungsform. In einem ersten Schritt S10 wird von einer ersten Recheneinheit ein Speicherplatz in einem Nachrichtenpool alloziert. Die erste Recheneinheit schreibt dann eine Nachricht in den allozierten Speicherplatz in Schritt S12 und beschreibt diese mit Information.

[0085] Im nächsten Schritt S14 wird die Nachricht durch die erste Recheneinheit indiziert. Indizieren bedeutet, dass die Speicheradresse oder auch der darauf verweisende Zeiger oder Pointer der Nachricht in einen Index umgerechnet wird. Dieser Index ist von den zweiten Recheneinheit(en) in seinem Adressraum als Speicheradresse der Nachricht rückrechenbar.

[0086] Den Index der Nachricht schreibt die erste Recheneinheit in einen Platz in einer Warteschlange. Der Platz in der Warteschlange kann dazu leer sein oder einen Index einer nicht länger verwendeten Nachricht beinhalten.

[0087] Eine zweite Recheneinheit kann auf den Arbeitsspeicher zugreifen und liest einen Platz aus der Warteschlange bzw. den darin gespeicherten Index in Schritt S16 aus. Anhand des Indexes ermittelt die zweite Recheneinheit in Schritt S18 die Nachricht aus dem Arbeitsspeicher, um sie in Schritt S20 zu verwenden. Die Nachricht kann beispielsweise für eine Rechenoperation verwendet, auf eine Peripherie (z.B. CAN, Ethernet) weitergeleitet oder in einen persistenten Speicher übertragen werden.

[0088] Eine nicht mehr benötigte Nachricht kann von der zweiten Recheneinheit freigegeben werden.

[0089] Im letzten Schritt S22 überprüft die erste Recheneinheit, ob die Nachricht freigegeben werden kann. Ist dies der Fall, kann der Speicherplatz, in dem die Nachricht liegt, wieder alloziert und der Index in der Warteschlange neu vergeben werden.

[0090] **Fig.** 2 zeigt schematisch den Aufbau eines Systems 10 mit einer ersten Recheneinheit 12 und einer zweiten Recheneinheit 14. Die Recheneinheiten 12 und 14 teilen sich einen gemeinsamen Arbeitsspeicher 16, auf den sie beide zugreifen können.

[0091] In einer Ausführungsform sind die Zugriffsrechte der Recheneinheiten 12, 14 auf den Arbeitsspeicher 16 beschränkt oder teilweise beschränkt. Beispielsweise kann die erste Recheneinheit 12 so ausgebildet sein, dass sie die Nachrichten schreiben, aber nicht lesen kann. Ferner kann die zweite Recheneinheit 14 beispielsweise so konfiguriert sein, dass sie den Arbeitsspeicher 16 nur auslesen, nicht aber beschreiben kann. Vorzugsweise ist die zweite Recheneinheit 14 mit dem Recht ausgestattet, zumindest in einen Teil des Arbeitsspeichers 16, im Besonderen einen Freigabewert, zu schreiben. Der Freigabewert zeigt an, ob eine betreffende Nachricht noch von der zweiten Recheneinheit 14 verwendet wird oder nicht.

[0092] Im Arbeitsspeicher 16 ist Speicher reserviert für einen Nachrichtenpool 18, in dem die Nachrichten gespeichert werden. Dem Nachrichtenpool 18 ist eine Warteschlange 20 umfassend mehrere Plätze für Indizes der Nachrichten aus dem Nachrichtenpool 18 zugeordnet. Die Indizes indizieren damit den genauen Platz, in dem die Nachrichten im Arbeitsspeicher 16 gespeichert sind.

[0093] Grundsätzlich könnten auch direkt ganze Nachrichten von der ersten Recheneinheit 12 an die zweite Recheneinheit 14 über die Warteschlange übertragen werden. Dies würde jedoch deutlich mehr Zeit, gemessen in Taktzyklen, in Anspruch nehmen, da die Indizes, sehr viel kleiner als die eigentlichen Nachrichten sind.

[0094] In **Fig.** 3 ist gemäß einer Ausführungsform dargestellt, in welchen Abständen 22 Indizes 24 in einer Warteschlange oder die Werte in einem Nachrichtenpool angeordnet sein können.

[0095] Die Indizes 24 oder Werte besitzen eine Länge I. Die Länge I ist so gewählt, dass sie der höchstens der kleinsten Datenbusweite der auf den Nachrichtenpool und die Warteschlange zugreifenden Recheneinheiten entspricht. Dadurch wird sichergestellt, dass auch Recheneinheiten mit kleinen Datenbusweiten die Indizes vollständig, ununterbrechbar und in einem Zuge auslesen können.

[0096] Die Indizes 24 sind in einem Abstand der Länge L angeordnet. Die Länge L entspricht mindestens der größten Datenbusweite der auf den Arbeitsspeicher zugreifenden Recheneinheiten. Recheneinheiten mit einer zu kleiner Datenbusbreite könnten insbesondere bei einem Schreibvorgang auch die benachbarten Indizes 24 mit beschreiben. Dies sollte jedoch in jedem Fall vermieden werden.

[0097] Vorzugsweise haben alle auf den Arbeitsspeicher zugreifenden Recheneinheiten dieselbe Datenbusweite, sodass die Längen L und I identisch ausgestaltet sein können. In diesem Fall würde der Arbeitsspeicher, der als Abstandhalter eingesetzt wird, minimiert werden.

[0098] **Fig.** 4 zeigt eine Ausführungsform eines Systems 10 mit einer ersten Recheneinheit 12 und drei zweiten Recheneinheiten 14a, 14b und 14c. Die erste Recheneinheit 12 kommuniziert mit den zweiten Recheneinheiten 14a, 14b und 14c jeweils über einen eigenen Arbeitsspeicher 16a, 16b und 16c. Dabei kann sie ihre Dienste den zweiten Recheneinheiten 14a, 14b und 14c unabhängig voneinander anbieten und in einer Verteilfunktion arbeiten.

[0099] Jede zweite Recheneinheit 14a, 14b und 14c hat dabei grundsätzlich nur Zugriff auf die in dem jeweils gemeinsam mit der ersten Recheneinheit 12 geteilten Arbeitsspeicher 16a, 16b und 16c. Die Nachrichten der anderen zweiten Recheneinheiten 14a, 14b und 14c sind dabei jenseits des Zugriffshorizonts. Aufgrund dieser Trennung der Arbeitsspeicher 16a, 16b und 16c eignet sich diese Ausführungsform für Anwendungen mit unterschiedlichen Sicherheitsanforderungen an die zweiten Recheneinheiten 14a, 14b und 14c. Ferner können zu einem späteren Zeitpunkt weitere zweite Recheneinheiten 14a, 14b und 14c mit jeweils eigenem Arbeitsspeicher 16a, 16b und 16c hinzugeschaltet werden.

[0100] In **Fig.** 5 ist eine weitere Ausführungsform schematisch dargestellt. In dieser Ausführungsform umfasst das System 10 eine erste Recheneinheit 12, einen Arbeitsspeicher 16 und drei zweite Recheneinheiten 14a, 14b und 14c. Jede der zweiten Recheneinheiten 14a, 14b und 14c kommuniziert mit der ersten Recheneinheit 12 über den gemeinsam genutzten Arbeitsspeicher 16 und einen gemeinsam genutzten Nachrichtenpool 18. Allerdings ist für jede zweite Recheneinheit 14a, 14b und 14c eine eigene Warteschlange 20a, 20b und 20c vorgesehen.

[0101] In dieser Ausführungsform stammen zwar alle Nachrichten von einem Sender, der ersten Recheneinheit 12, können jedoch einen oder mehrere Empfänger, die zweiten Recheneinheiten 14a, 14b und 14c haben. Einzelne Nachrichten können somit kopierfrei an mehrere Empfänger gleichzeitig versendet werden.

[0102] Diese Ausführungsform ist beispielsweise geeignet für das Routen von Nachrichten, wenn der Speicherplatz der Nachrichten für den Datenempfang zur Verfügung gestellt werden muss, bevor der eigentliche Empfänger bekannt ist. Bei recheninten-

siven Aufgaben kann eine weitere, derart aufgebaute Ausführungsform genutzt werden, um dynamisch Ressourcen in Form von zweiten Recheneinheiten 14a, 14b und 14c hinzu- oder abzuschalten.

[0103] In dieser Ausführungsform kann eine Nachricht mehrfach verwendet werden. Umso wichtiger ist es, dass die erste Recheneinheit 12 einen Nutzungswert verwendet, der die Anzahl der Indizes pro Nachricht nachhält.

[0104] Wenn ein Index einer Nachricht beispielsweise über zwei Warteschlangen an zwei der zweiten Recheneinheiten 14a, 14b und 14c weitergegeben wurde, muss die erste Recheneinheit 12 abwarten, bis keine der Recheneinheiten 14a, 14b und 14c die Nachricht mehr verwendet. Um dies zu ermöglichen, kann jede zweite Recheneinheit 14a, 14b und 14c einen Freigabewert vergeben. Die Summe aller Freigabewerte wird dann von der ersten Recheneinheit 12 mit dem Nutzungswert verglichen. Sind die Werte gleich, haben so viel zweite Recheneinheiten 14a, 14b und 14c die Nachricht freigeben, wie sie verteilt wurde. Damit ist sicher, dass die Nachricht nicht mehr verwendet wird. Die erste Recheneinheit 12 kann die Nachricht dann freigeben.

[0105] **Fig.** 6 zeigt eine Ausführungsform eines Systems 10 mit vier Recheneinheiten 12, 14, 26 und 28, die „in Reihe geschaltet“ sind. Alle Recheneinheiten 12, 14, 26 und 28 teilen sich den gemeinsamen Arbeitsspeicher 16 und greifen auf einen gemeinsam genutzten Nachrichtenpool 18 zu.

[0106] Im Gegensatz zu der Ausführungsform aus **Fig.** 5 sind hier die Recheneinheiten 12, 14, 26 und 28 nacheinander über die Warteschlangen 20a, 20b und 20c verbunden. Die erste Recheneinheit 12 kann beispielsweise eine Nachricht des den Nachrichtenpools 18 beschreiben, sodass die zweite Recheneinheit 14 diese Nachricht nach dem Erhalt verarbeiten kann. Anstatt die Nachricht nach dem Verwenden freizugeben, kann die Recheneinheit 14 die Nachricht wiederum an die nächste Recheneinheit 26 weiterreichen, indem sie in die Warteschlange 20b einen entsprechenden Index schreibt. Die Recheneinheit 26 kann die Nachricht dann ebenfalls aus dem Nachrichtenpool 18 abrufen, wobei die Nachricht selbst nicht kopiert werden muss.

[0107] In der dargestellten Ausführungsform gibt es eine weitere Recheneinheit 28, die die Nachricht ein letztes Mal verwendet. Dazu erhält sie von der Recheneinheit 26 einen Index über die Warteschlange 20c.

[0108] Die letzte Recheneinheit 14, 26 und 28, die die Nachricht verwendet und diese nicht indiziert an die nächste Recheneinheit weitergibt, markiert die Nachricht als freigegeben, sodass die erste Recheneinheit 12 die Nachricht als nicht genutzt markieren und den Speicherplatz neu allozieren kann.

Patentansprüche

1. Verfahren zum Übertragen einer Nachricht mittels eines von zumindest zwei Recheneinheiten (12, 14a, 14b, 14c, 26, 28) verwendeten Arbeitsspeichers (16a, 16b, 16c),
wobei der Arbeitsspeicher (16a, 16b und 16c) zumindest einen als Nachrichtenpool 18 eingerichteten Speicherbereich umfasst,
wobei jedem Nachrichtenpool 18 zumindest eine Warteschlange (20a, 20b, 20c) zugeordnet ist,
wobei jede Warteschlange (20a, 20b, 20c) mehrere Plätze für Indizes umfasst, wobei die Indizes auf Nachrichten in dem Nachrichtenpool (18) verweisen,
wobei eine erste Recheneinheit (12) und eine zweite Recheneinheit (14a, 14b und 14c) der zumindest zwei Recheneinheiten (12, 14a, 14b, 14c, 26, 28) auf den Arbeitsspeicher (16a, 16b und 16c) zugreifen,
wobei das Verfahren die folgenden Schritte umfasst:
- Allozieren eines Speicherplatzes (S10) im Nachrichtenpool (18) durch die erste Recheneinheit (12);
- Schreiben einer Nachricht (S12) in den allozierten Speicherplatz durch die erste Recheneinheit (12);
- Indizieren der Nachricht und Schreiben eines Indexes (S14, 24) der Nachricht in einen Platz in der Warteschlange (20a, 20b, 20c) durch die erste Recheneinheit (12);
- Auslesen des Indexes (S16, 24) aus der Warteschlange (20a, 20b, 20c) durch die zweite Recheneinheit (14a, 14b, 14c);
- Ermitteln der Nachricht (S18) unter Verwendung des Indexes (24) durch die zweite Recheneinheit (14a, 14b, 14c);
- Verwenden der Nachricht (S20) durch die zweite Recheneinheit (14a, 14b, 14c); und
- Prüfen auf eine Freigabe (S22) der Nachricht und Markieren des Platzes in der Warteschlange (20a, 20b, 20c) zur Wiederverwendung durch die erste Recheneinheit (12).

2. Verfahren nach Anspruch 1, wobei dem Prüfen auf die Freigabe (S22) der Nachricht die folgenden Schritte vorausgehen:
- Vergeben eines Nutzungswerts für den Index (24) durch die erste Recheneinheit (12);
- Vergeben eines Freigabewerts für den Index (24) nach der Nutzung der Nachricht durch die zweite Recheneinheit (14a, 14b, 14c); und
- Übertragen des Freigabewerts von der zweiten Recheneinheit (14a, 14b, 14c) an die erste Recheneinheit (12), wobei das Prüfen auf die Freigabe (S22) einen Vergleich des Freigabewerts und des Nutzungswerts umfasst.

3. Verfahren nach Anspruch 1, wobei eine weitere Recheneinheit (26, 28) auf den Arbeitsspeicher (16a, 16b, 16c) zugreift, wobei dem Prüfen auf die Freigabe (S22) der Nachricht die folgenden Schritte vorausgehen:
- Vergeben eines Nutzungswerts für den Index (24) beim Schreiben des Indexes durch die erste Recheneinheit (12);
- Indizieren der Nachricht und Schreiben des Indexes (24) der Nachricht in einen Platz in einer weiteren Warteschlange (20a, 20b, 20c) durch die zweite Recheneinheit (14a, 14b, 14c) nach dem Verwenden der Nachricht durch die zweite Recheneinheit (14a, 14b, 14c);
- Auslesen des Indexes (24) aus der weiteren Warteschlange (20a, 20b, 20c) durch die weitere Recheneinheit (26, 28);
- Ermitteln der Nachricht anhand des Indexes (24) durch die weitere Recheneinheit (26, 28);
- Verwenden der Nachricht durch die weitere Recheneinheit (26, 28);
- Vergeben eines Freigabewerts für den Index (24) nach der Nutzung der Nachricht durch die weitere Recheneinheit (26, 28); und
- Übertragen des Freigabewerts von der weiteren Recheneinheit (26, 28) an die erste Recheneinheit (12), wobei das Prüfen auf die Freigabe einen Vergleich des Freigabewerts und des Nutzungswerts umfasst.

4. Verfahren nach einem der vorangegangenen Ansprüche, wobei das Indizieren der Nachricht (S14) ein Hinzufügen eines Zeitstempels zu der Nachricht umfasst und
wobei das Prüfen der Freigabe (S22) ein Überprüfen des Zeitstempels der Nachricht umfasst,
wobei die erste Recheneinheit (12) den indizierten Platz in der Warteschlange (20a, 20b, 20c) freigibt, wenn das Überprüfen des Zeitstempels das Erreichen eines definierten Alters der Nachricht anzeigt.

5. Verfahren nach einem der vorangegangenen Ansprüche, wobei zumindest eine weitere zweite Recheneinheit (14a, 14b, 14c) auf den Arbeitsspeicher zugreift, wobei das Verfahren zwischen der ersten Recheneinheit (12) und der zumindest einen weiteren zweiten Recheneinheit (14a, 14b, 14c) jeweils mit einer eigenen Warteschlange (20a, 20b, 20c) durchgeführt wird.

6. Verfahren nach Anspruch 5, wobei jede Warteschlange (20a, 20b, 20c) demselben Nachrichtenpool (18) zugeordnet ist.

7. Verfahren nach einem der vorangegangenen Ansprüche, wobei die Recheneinheiten (12, 14a, 14b, 14c, 26, 28) mit unterschiedlichen Betriebssystemen betrieben werden.

8. Verfahren nach einem der vorangegangenen Ansprüche, wobei der Abstand der Speicherplätze für die Indizes (24) in der Warteschlange (20) mindestens der größten Datenbusweite der auf den Arbeitsspeicher (16a, 16b, 16c) zugreifenden Recheneinheiten (12, 14a, 14b, 14c, 26, 28) entspricht.

9. Verfahren nach einem der vorangegangenen Ansprüche, wobei der Speicherplatz für einen Index (24) in der Warteschlange (20) höchstens der kleinsten Datenbusweite der auf den Arbeitsspeicher (16a, 16b, 16c) zugreifenden Recheneinheiten (12, 14a, 14b, 14c, 26, 28) entspricht.

10. Verfahren nach einem der vorangegangenen Ansprüche, wobei die Warteschlange (20a, 20b, 20c) als Ringpuffer eingerichtet ist.

11. Computerprogramm mit Programmcode, um ein computerimplementiertes Verfahren zum Übertragen einer Nachricht mittels eines von zumindest zwei Recheneinheiten (12, 14a, 14b, 14c, 26, 28) verwendeten Arbeitsspeichers (16a, 16b, 16c) nach einem der vorangegangenen Ansprüche auszuführen, wenn das Computerprogramm auf einem Computer ausgeführt wird.

12. Computerlesbarer Datenträger mit Programmcode eines Computerprogramms, um ein Verfahren zum Übertragen einer Nachricht mittels eines von zumindest zwei Recheneinheiten (12, 14a, 14b, 14c, 26, 28) verwendeten Arbeitsspeichers (16a, 16b, 16c) nach einem der Ansprüche 1 bis 10 auszuführen, wenn das Computerprogramm auf einem Computer ausgeführt wird.

13. System zum Übertragen einer Nachricht mittels eines von zumindest zwei Recheneinheiten (12, 14a, 14b, 14c, 26, 28) verwendeten Arbeitsspeichers (16a, 16b, 16c), wobei das System umfasst:
- einen Arbeitsspeicher (16a, 16b, 16c),
- eine erste Recheneinheit (12), und
- eine zweite Recheneinheit (14a, 14b, 14c), wobei das System Mittel umfasst, um das Verfahren nach einem der Ansprüche 1 bis 10 auszuführen.

14. System nach 13, wobei
- die erste Recheneinheit (12) ein Mikroprozessor und die zweite Recheneinheit (14a, 14b, 14c) ein Mikrocontroller oder
- die erste Recheneinheit (12) ein Mikrocontroller und die zweite Recheneinheit (14a, 14b, 14c) ein Mikroprozessor ist.

Es folgen 4 Seiten Zeichnungen

Anhängende Zeichnungen

Fig. 1

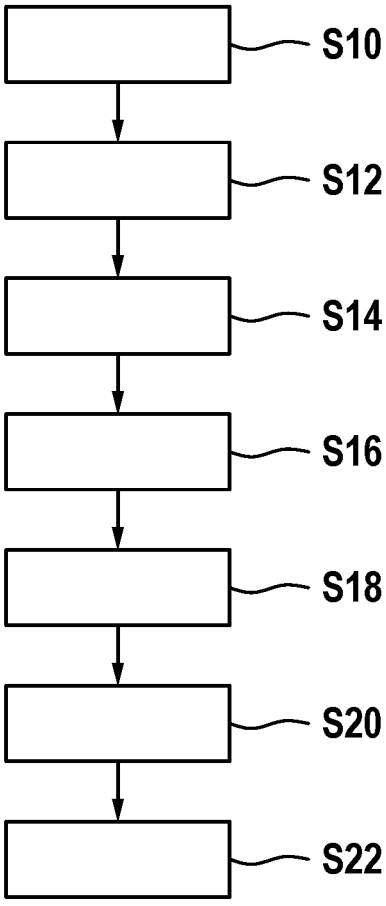


Fig. 2

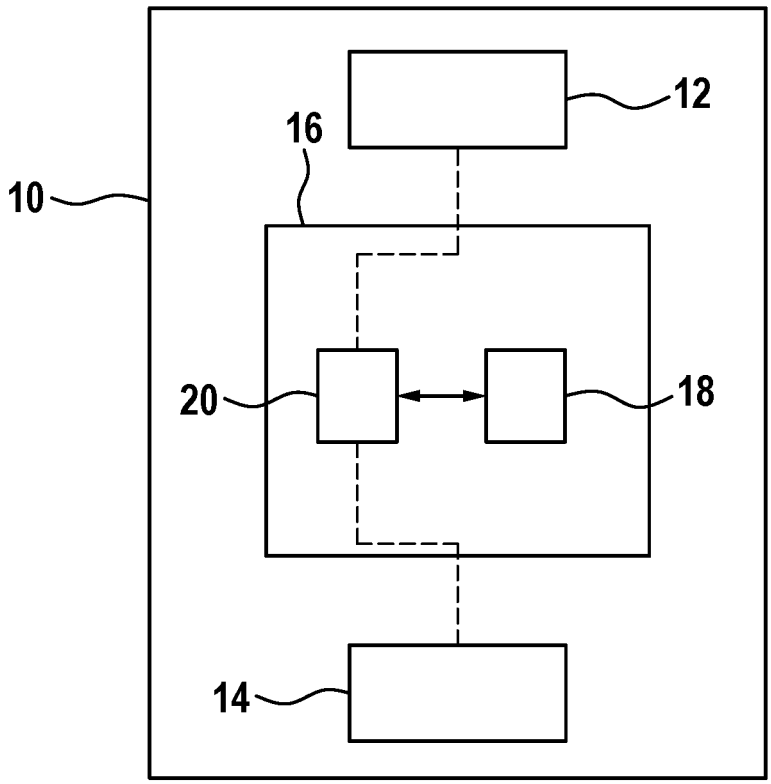


Fig. 3

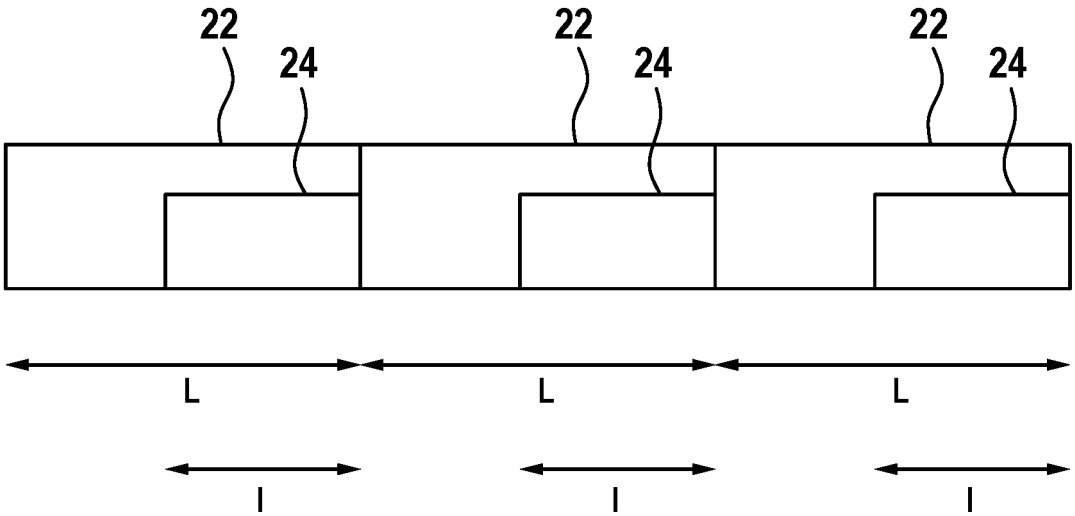


Fig. 4

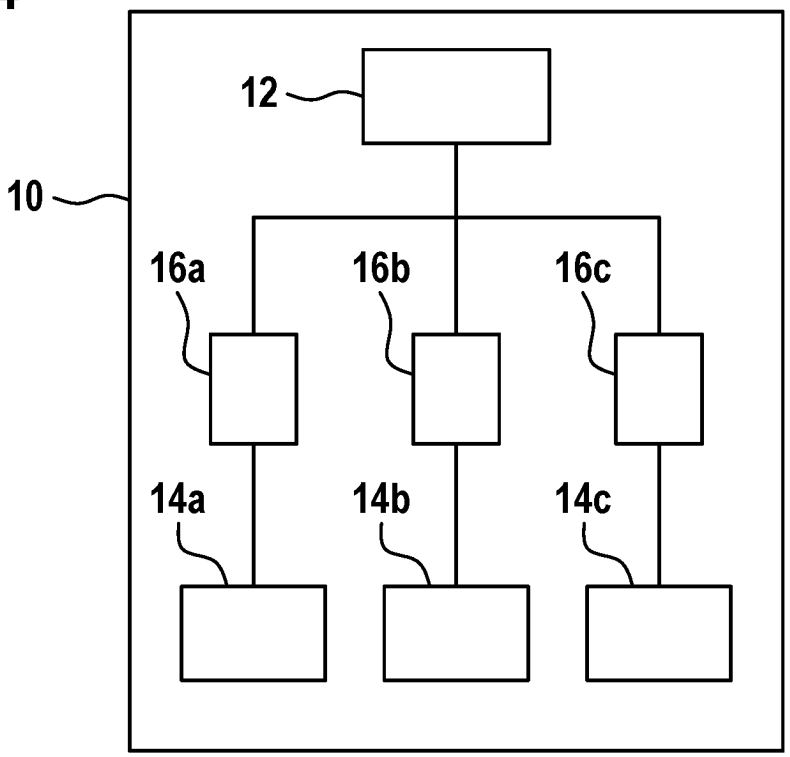


Fig. 5

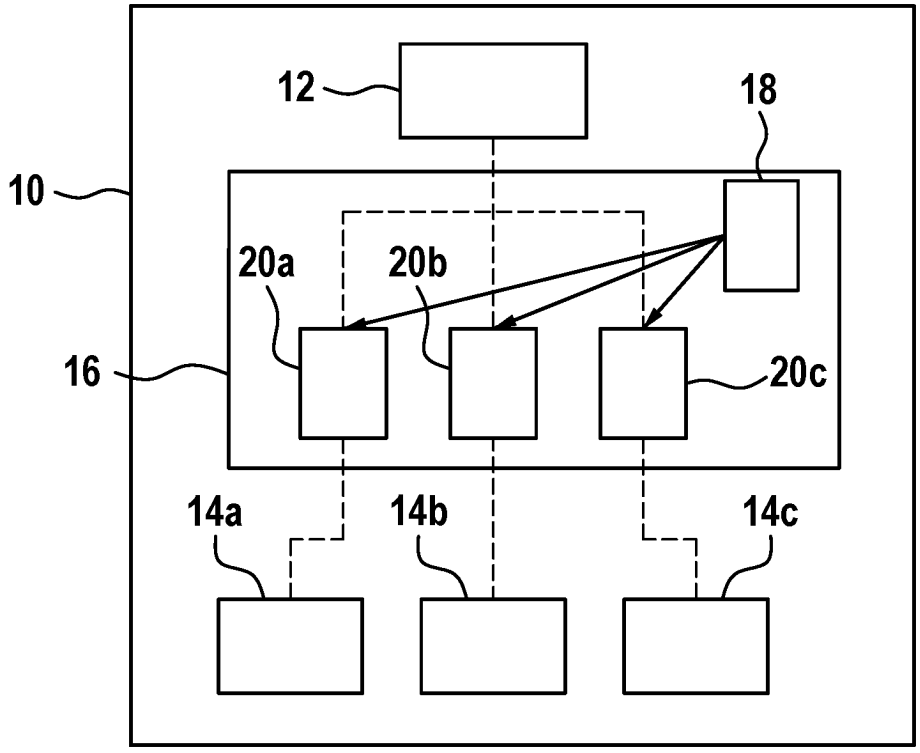
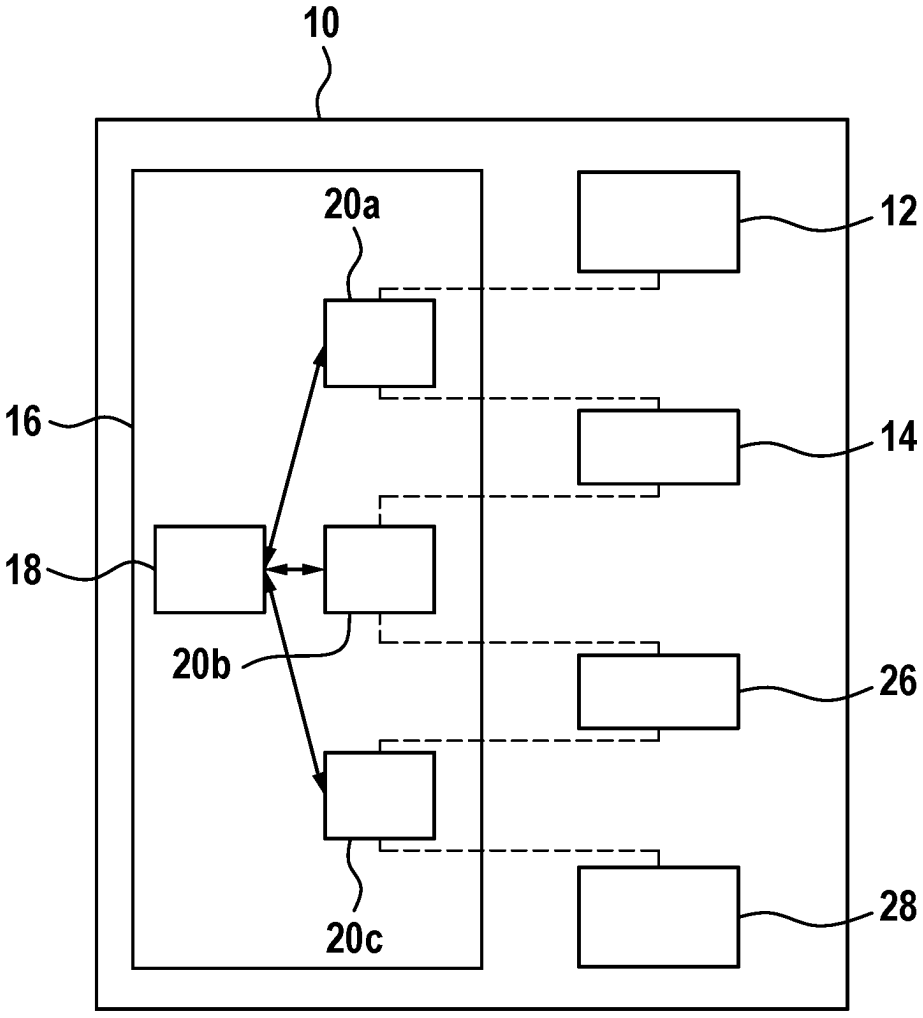


Fig. 6



 Das Dokument wurde durch die Firma Luminess hergestellt.