



(10) **DE 10 2019 210 552 B4** 2022.12.22

(12)

Patentschrift

(21) Aktenzeichen: **10 2019 210 552.1**
(22) Anmeldetag: **17.07.2019**
(43) Offenlegungstag: **21.01.2021**
(45) Veröffentlichungstag
der Patenterteilung: **22.12.2022**

(51) Int Cl.: **H04L 69/22 (2022.01)**
G06F 9/455 (2006.01)
G06F 13/00 (2006.01)
H04L 49/111 (2022.01)
H04L 49/901 (2022.01)

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

(73) Patentinhaber:
AUDI AG, 85057 Ingolstadt, DE

(74) Vertreter:
**Hofstetter, Schurack & Partner - Patent- und
Rechtsanwaltskanzlei, PartG mbB, 81541
München, DE**

(72) Erfinder:
**Dalke, Christoph, 85110 Kipfenberg, DE; Lerzer,
Jürgen, 92318 Neumarkt, DE; Paffen, Matthijs,
85123 Karlskron, DE; Sander, Christian, 85276
Pfaffenhofen, DE**

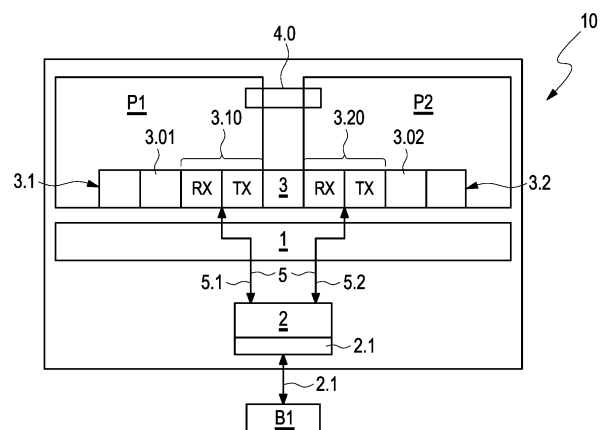
(56) Ermittelter Stand der Technik:
siehe Folgeseiten

(54) Bezeichnung: **Fahrzeugelektronikeinheit mit einer physikalischen Netzwerkschnittstelle und mehreren, virtuelle Netzwerkschnittstellen aufweisenden virtuellen Maschinen sowie Datenkommunikationsverfahren**

(57) Hauptanspruch: Fahrzeugelektronikeinheit (10) umfassend

- wenigstens zwei virtuellen Maschinen (P1, P2) mit jeweils einer virtuellen Netzwerkschnittstelle (5.1, 5.2),
- einen Hypervisor (1), welcher zur Trennung der virtuellen Maschinen (P1, P2) ausgebildet ist,
- einen Speicher (3) mit jeweils einer der virtuellen Maschinen (P1, P2) exklusiv zugeordneten Speicherabschnitten (3.1, 3.2), wobei jeder Speicherabschnitt (3.1, 3.2) direkte Speicherbereiche (3.10, 3.20) für die zugeordnete virtuelle Netzwerkschnittstelle (5.1, 5.2) aufweist, und
- eine physikalische Netzwerkschnittstelle (2.1) zur Herstellung der jeweiligen Kommunikationsverbindung zwischen der jeweiligen virtuellen Netzwerkschnittstelle (5.1, 5.2) der virtuellen Maschinen (P1, P2) einerseits und einem Fahrzeugnetzwerk (B1) andererseits, dadurch gekennzeichnet, dass
- die physikalische Netzwerkschnittstelle (2.1) zur Herstellung der jeweiligen Kommunikationsverbindung dazu eingerichtet ist, für aus dem Fahrzeugnetzwerk (B1) eintreffende Datennachrichten jeweils einen eine Header-Information aufweisenden Header der Datennachricht mittels einer Filter- und Routingeinheit (2.3) hinsichtlich der virtuellen Maschine (P1, P2), welcher ein Datenpaket der Datennachricht zuzuweisen ist, zu analysieren, und das Datenpaket mittels einer direkten Speicherzugriffs-Funktion, DMA-Funktion, in den direkten Speicherbereich (3.10, 3.20) der der Header-Information entsprechenden virtuellen Maschine (P1, P2) zu kopieren, sodass es für die virtuelle Netzwerkschnittstelle (5.1, 5.2) dieser virtuellen Maschine (P1, P2) verfügbar ist, und dass

- die Filter- und Routingeinheit (2.3) dazu eingerichtet ist, einen Nachrichtensignalisierung-Interrupt dem Hypervisor (1) und/oder einem Interrupt-Controller zur Verfügung zu stellen, wobei die Filter- und Routingeinheit (2.3) in Abhängigkeit davon, welcher virtuellen Maschine (P1, P2) das Datenpaket zuzuweisen ist, einen anderen Nachrichtensignalisierung-Interrupt erzeugt, sodass in dem Nachrichtensignalisierung-Interrupt selbst kodiert ist, welche virtuelle Maschine zu benachrichtigen ist und die Nachrichtensignalisierung-Interrupts sich in einer Interrupt-Kennung oder in einer Kanalangabe oder Quellenangabe, aber nicht in der Angabe einer der virtuellen Maschine unterscheiden, sondern allein durch die Tatsache, dass pro virtueller Maschine ein anderer Interrupt-Typ verwendet wird, signalisiert ...



(56) Ermittelter Stand der Technik:

DE	10 2018 200 555	A1
US	2018 / 0 167 364	A1
US	2019 / 0 087 245	A1

SHAHER, J. [et al.]: Concurrent direct network access for virtual machine monitors. In: 2007 IEEE 13th International Symposium on High Performance Computer Architecture. IEEE, 2007. S. 306-317

Beschreibung

[0001] Die Erfindung betrifft eine Fahrzeugelektronikeinheit mit einer physikalischen Netzwerkschnittstelle und wenigstens zwei virtuellen Maschinen sowie ein Datenkommunikationsverfahren zwischen den virtuellen Maschinen und der Netzwerkschnittstelle zu einem lokalen Fahrzeugnetzwerk eines Fahrzeugs.

[0002] In modernen Fahrzeugen werden Informationen wie Fahrzeugzustandsinformationen und Unterhaltung (Entertainment) in einem gemeinsamen System aus Bedien- und Anzeigeelementen zugänglich gemacht und als Cockpit- und Infotainmentsystem bezeichnet. Derartige Systeme bieten bspw. Zugriff auf Betriebs- und Fahrzeugdatenanzeigen, Navigation, Servicefunktionen, Mobiltelefon und Internet sowie Audio/Video usw.

[0003] Um die Applikationen für die unterschiedlichen Komponenten zu realisieren, werden Mehrprozessorsysteme für solche Cockpit und Infotainmentsystem eingesetzt und die entsprechenden Funktionen in einem solchen Mehrprozessorsystem zusammengefasst. Um Nutzdaten und Steuerdaten zwischen den unterschiedlichen Komponenten austauschen zu können, wird das Konzept der Visualisierung mit mehreren Betriebssystemen (OS) durch einen Hypervisor (VMM, virtual machine monitor) realisiert. Mittels solcher Hypervisoren wird eine virtuelle Umgebung mit Hardwareressourcen, wie bspw. Prozessor, Speicher und Peripherie definiert, die unabhängig von der tatsächlich vorhandenen Hardware als Basis für die Installation der Betriebssysteme (auch Gastssysteme genannt) dient. Damit können mehrere Betriebssysteme (auch Partitionen) parallel auf eine gemeinsame Hardware betrieben werden. Der Hypervisor verwaltet hierbei die Ressourcenzuteilung für die einzelnen Partitionen, so dass für jedes Betriebssystem die benötigten Ressourcen verfügbar sind.

[0004] Es ist bekannt, dass zur Übertragung von Daten innerhalb einer Hypervisor-Architektur mit einer ersten und zweiten Partition ein Datenübertragungsverfahren verwendet wird, welche zu einer unerwünschten erhöhten CPU-Last führt und nachteilige Eigenschaften hinsichtlich von Sicherheitsaspekten aufweist. Bei diesem Datenübertragungsverfahren werden Ethernet-Datenpakete zwischen einer Partition und einer Netzwerkschnittstelle eines Fahrzeugnetzwerkes übertragen, wobei Daten der zweiten Partition nicht direkt an die Netzwerkschnittstelle übergeben werden, sondern zunächst über ein virtuelles Netzwerk zur ersten Partition und erst anschließend zur Netzwerkschnittstelle übergeben werden.

[0005] Aus der US 2009/0307436 A1 ist ein Datenverarbeitungssystem mit einem Hypervisor mit einer geteilten Speicherpartition zur Seitenfehlerverarbeitung bekannt, bei welchem der Hypervisor Daten von mehreren Partitionen auf der geteilten Speicherpartition verwaltet.

[0006] Die US 2012/0084517A1 beschreibt ein Verfahren zur Bereitstellung eines gemeinsamen Speicherplatzes zwischen Knoten einer Haupt- und Subpartition eines Hypervisors.

[0007] Ein Datenübertragungsverfahren zwischen einer vertrauenswürdigen Domain (trusted Domain) und einer Gast-Domain beschreibt die US 2014/0330868 A1, bei welchem ein geteilter Speicher von den Domains über eine sende- und Empfangseinheit genutzt wird und diese Domains mit einem Hypervisor verbunden sind.

[0008] Aus einer wissenschaftlichen Veröffentlichung von P. Willmann et al. (Paul Willmann, Jeffrey Shafer, David Carr, Aravind Menon, Scott Rixner, Alan L. Cox, Willy Zwaenepoel; „Concurrent Direct Network Access for Virtual Machine Monitors“, 2007 IEEE 13th International Symposium on High Performance Computer Architecture) ist bekannt, dass in einem Computersystem mehrere virtuelle Maschinen direkt auf dieselbe Netzwerkkarte zugreifen und diese für eigenen Netzwerkverkehr nutzen können. Die Netzwerkkarte muss hierfür allerdings in nachteiliger Weise mit einer besonderen CDNA-Logik (CDNA - Concurrent Direct Network Access) ausgestattet sein, was die Anschaffungskosten für eine solche Netzwerkkarte erhöht. Zudem ist die Anzahl der versorgbaren virtuellen Maschinen aufgrund eines limitierten Kontext-Speichers der Netzwerkkarte begrenzt. Damit nach der Ankunft eines Datenpakets ein Hypervisor die richtige virtuelle Maschine per Interrupt benachrichtigen kann, ist zudem zwischen der physikalischen Netzwerkschnittstelle und dem Hypervisor eine aufwändige Kommunikation notwendig, die zunächst die Übertragung eines Bitvektors und anschließend die Auslösung eines physikalischen Interrupts vorsieht, der von einer einzelnen Interrupt-Dienstroutine des Hypervisors behandelt wird.

[0009] Die Internetseite „etherealmind.com“ beschreibt eine Filtereinheit TCAM (Ternary Content Address Memory) für Netzwerkkarten (<https://etherealmind.com/basics-what-is-ternary-content-address-memorytcam/>), mit der für Datenpakete aus einem Datennetzwerk eine Filterregel vorgegeben werden kann. Trifft ein Datenpaket ein, das auf eine hinterlegte Filterregel passt, so kann für das Datenpaket durch die Netzwerkkarte eine Verarbeitungsprozedur angestoßen werden. Eine weitere hardware-basierte Filtereinheit CAM ist auf derselben Internetseite unter <https://etherealmind.com/basics->

what-is-content-addressable-memory-cam/ beschrieben.

[0010] Aus der nach dem Anmeldetag der vorliegenden Anmeldung veröffentlichten DE 10 2018 200 555 A1 ist bekannt, Datennachrichten, die in einem Computersystem über eine Netzwerkkarte eintreffen, auf unterschiedliche virtuelle Maschinen aufzuteilen, indem die Netzwerkkarte mittels eines Interrupts das Weiterleiten auslöst und dann zum Ermitteln der adressierten virtuellen Maschine Header-Informationen der Datennachricht ausgewertet werden.

[0011] Aus der US 2019/0087245 A1 ist bekannt, Nachrichten, die in einem Computersystem über eine Netzwerkkarte eintreffen, an unterschiedliche virtuelle Maschinen weiterzuleiten. An welche virtuelle Maschine eine Nachricht jeweils weitergeleitet werden soll, ergibt sich anhand der MAC-Adressen, von denen die Netzwerkkarte mehreren nutzen kann. Jede virtuelle Maschine kann direkt auf die Netzwerkkarte durchgreifen, sodass eine Vermittlung durch einen Hypervisor nicht notwendig ist.

[0012] Aus der US 2018/0167364 A1 ist bekannt, dass eine Netzwerkkarte NIC und eine CPU über Interrupts miteinander kommunizieren können. Falls auf der CPU mehrere virtuelle Maschinen laufen, wird eine direkte Kommunikation zwischen den virtuellen Maschinen auf der CPU und „virtuellen Funktionen“ VF auf der Netzwerkkarte hergestellt. Eine Kommunikation zwischen der Netzwerkkarte und den virtuellen Maschinen läuft hierbei über ein Interrupt-Mapping, durch welches ein realer Interrupt ohne Zwischenschaltung von virtuellen Interrupts von den virtuellen Maschinen verarbeitet werden kann.

[0013] Die Aufgabe der Erfindung besteht darin, eine Fahrzeugelektronikeinheit mit wenigstens zwei virtuellen Maschinen anzugeben, mit welcher eine leistungsstarke, sichere und zuverlässige Datenkommunikation innerhalb der Architektur der Fahrzeugelektronikeinheit erzielt wird. Ferner ist es Aufgabe ein entsprechendes Datenkommunikationsverfahren anzugeben.

[0014] Die erstgenannte Aufgabe wird durch eine erfindungsgemäße Fahrzeugelektronikeinheit gelöst.

[0015] Eine solche Fahrzeugelektronikeinheit umfasst

- wenigstens zwei virtuellen Maschinen mit jeweils einer virtuellen Netzwerkschnittstelle,
- einen Hypervisor, welcher zur Trennung der virtuellen Maschinen ausgebildet ist,

- einen Speicher mit jeweils einer der virtuellen Maschinen exklusiv zugeordneten Speicherabschnitten, wobei jeder Speicherabschnitt direkte Speicherbereiche für die zugeordnete virtuelle Netzwerkschnittstelle aufweist, und

- eine physikalischen Netzwerkschnittstelle zur Herstellung einer jeweiligen Kommunikationsverbindung zwischen der jeweiligen virtuellen Netzwerkschnittstelle der virtuellen Maschinen einerseits und einem Fahrzeugnetzwerk andererseits.

[0016] Diese Ausgestaltung einer Fahrzeugelektronikeinheit ist an sich aus dem Stand der Technik bekannt. Um nun eine die besagte Kommunikationsverbindung zwischen der physikalischen Netzwerkschnittstelle und den einzelnen virtuellen Netzwerkschnittstellen der virtuellen Maschinen mit geringem schaltungstechnischen Aufwand für die physikalischen Netzwerkschnittstelle zu realisieren, ist erfindungsgemäß Folgendes vorgesehen.

[0017] Die physikalische Netzwerkschnittstelle ist zur Herstellung der jeweiligen Kommunikationsverbindung dazu eingerichtet, für Datennachrichten, die aus dem Fahrzeugnetzwerk eintreffen, jeweils einen eine Header-Information aufweisenden Header der Datennachricht mittels einer Filter- und Routingeinheit hinsichtlich derjenigen virtuellen Maschine, welcher das Datenpaket der Datennachricht zuzuweisen ist, zu analysieren, und das Datenpaket mittels einer direkten Speicherzugriffs-Funktion, DMA-Funktion (DMA - Direct Memory Access), in den direkten Speicherbereich der der Header-Information entsprechenden virtuellen Maschine zu kopieren, sodass es für die virtuelle Netzwerkschnittstelle dieser virtuellen Maschine verfügbar ist. Die jeweilige virtuelle Netzwerkschnittstelle kann also in dem jeweiligen direkten Speicherbereich auf das Datenpaket in derselben Weise zugreifen, wie eine Netzwerkschnittstelle ansonsten auf den Speicher einer physikalischen Netzwerkschnittstelle zugreifen würde, wenn keine virtuelle Maschine vorhanden wäre.

[0018] Nun ist noch eine Benachrichtigung der virtuellen Maschine darüber notwendig, dass neue RX-Daten (RX - Receive; Empfang) vorliegen, nämlich ein neu empfangenes Datenpaket. Hierzu ist die Filter- und Routingeinheit dazu eingerichtet, einen Nachrichtensignalisierung-Interrupt dem Hypervisor zur Verfügung zu stellen, wobei die Filter- und Routingeinheit in Abhängigkeit davon, welcher virtuellen Maschine das Datenpaket zuzuweisen ist, einen anderen Nachrichtensignalisierung-Interrupt erzeugt. Mit anderen Worten ist also schon in dem Nachrichtensignalisierung-Interrupt selbst kodiert, welche virtuelle Maschine zu benachrichtigen ist. Die Nachrichtensignalisierung-Interrupts unterscheiden sich insbesondere in einer Interrupt-Kennung

oder in einer Kanalangabe oder Quellenangabe, also nicht in der Angabe einer der virtuellen Maschine, da in einem Interrupt solche Informationen nicht übermittelt werden können, weil ein Interrupt nur begrenzt mit zusätzlichen Informationen versehen werden kann. Aber allein die Tatsache, dass pro virtueller Maschine ein anderer Interrupt-Typ verwendet wird, wird hier ausgenutzt, zu signalisieren, welche virtuelle Maschine zu benachrichtigen ist.

[0019] Der Hypervisor ist dazu eingerichtet, einen virtuellen Interrupt für die der Header-Information entsprechenden virtuellen Maschine bereitzustellen, wobei der Hypervisor den virtuellen Interrupt gemäß dem oben beschriebenen Ansatz in Abhängigkeit davon erzeugt, welcher Nachrichtensignalisierung-Interrupt eintrifft. Der Hypervisor ist also nicht darauf angewiesen, zusätzliche Informationen betreffend die zu benachrichtigende virtuelle Maschine auszuwerten. Insbesondere erzeugt er den virtuellen Interrupt unabhängig von der Header-Information und unabhängig von dem Datenpaket, da er nur ausgewertet, welcher Nachrichtensignalisierung-Interrupt aus mehreren möglichen Nachrichtensignalisierung-Interrupts eingetroffen ist.

[0020] Zusätzlich oder alternativ zum Hypervisor kann auch ein Interrupt-Controller bereitgestellt sein, der dazu eingerichtet ist, einen virtuellen Interrupt für die der Header-Information entsprechenden virtuellen Maschine bereitzustellen, wobei der Interrupt-Controller den virtuellen Interrupt in Abhängigkeit davon erzeugt, welcher Nachrichtensignalisierung-Interrupt eintrifft. Falls ein solcher Interrupt-Controller zur Verfügung steht, kann durch dessen entsprechende Programmierung oder Konfiguration die Erzeugung von virtuellen Interrupts durchgeführt und damit vermieden werden, dass der Hypervisor Rechenressourcen für die Weiterleitung der Interrupts bereitstellen muss. Ein Interrupt-Controller ist eine dedizierte Hardware, beispielsweise ein integrierter Schaltkreis (IC - integrated circuit), der auf die Behandlung von Interrupts spezialisiert ist. Er kann durch Konfigurieren von Registern für die besagte Funktion angepasst werden. Er kann zum Erzeugen von virtuellen Interrupts mit einer Schnittstelle für virtuelle Maschinen ausgestattet sein. Ein solcher Interrupt-Controller ist beispielsweise unter dem Produktnamen „GIC 400“ ® von dem Unternehmen ARM ® beziehbar.

[0021] Zusätzlich oder alternativ zur Erzeugung eines virtuellen Interrupts kann vorgesehen sein, dass der Hypervisor dazu eingerichtet ist, in einem jeweiligen Signalisierungsbereich des Speicherabschnitts der der Header-Information entsprechenden virtuellen Maschine abzuspeichern oder zu signalisieren, dass neue RX-Daten vorliegen. Dies kann durch Hinterlegen von vorbestimmten digitalen Werten erfolgen.

[0022] Bei der erfindungsgemäßen Fahrzeugelektronikeinheit wird der gemeinsame Speicher in jeweils einer virtuellen Maschine exklusiv zugeordneten Speicherabschnitte aufgeteilt, wobei jeder Speicherabschnitt direkte Speicherbereiche für die Kommunikationsverbindung der virtuellen Maschine über deren zugeordneten virtuellen Netzwerkschnittstelle aufweist. Eine solche virtuelle Netzwerkschnittstelle kann insbesondere als ein sogenannten Protokollstack, z.B. ein TCP/IP-Stack, realisiert sein (TCP - Transmission Control Protocol; IP - Internet Protocol). Ein exklusiver Speicherabschnitt ist nur von demselben zugeordneten virtuellen Maschine zugreifbar. Dies kann mittels eines sogenannten Memory-Mapping (Speichereinblendung) realisiert sein. Die besagte Kommunikationsverbindung führt von der virtuellen Netzwerkschnittstelle zu der physikalischen Netzwerkschnittstelle. Dort laufen die Kommunikationsverbindungen aller virtuellen Netzwerkschnittstelle zusammen, ohne dass sich die Kommunikationsverbindungen gegenseitig beeinflussen können.

[0023] Die Vorteile einer solchen Fahrzeugelektronik-Architektur bestehen in einer klaren Trennung der zwischen jeweils einer virtuellen Maschine und der physikalischen Netzwerkschnittstelle des lokalen Fahrzeugnetzwerkes zu kommunizierenden Datenpaketen und damit können weder negativen Auswirkungen auf die Sicherheit dieser Architektur noch auf die Leistungsfähigkeit und Zuverlässigkeit eintreten.

[0024] Die Fahrzeugelektronikeinheit ist mit einem Hypervisor ausgebildet, welcher zur Trennung der mindestens zwei virtuellen Maschinen ausgebildet ist.

[0025] Um zu signalisieren, für welche virtuelle Maschine ein Datenpaket (neue RX-Daten) eingetroffen ist, wird von der physikalischen Netzwerkschnittstelle aus mehreren möglichen Nachrichtensignalisierung-Interrupts derjenige ausgewählt, welcher der entsprechenden virtuellen Maschine zugeordnet ist. Somit genügt bereits das Senden dieses Nachrichtensignalisierung-Interrupts allein, damit der Hypervisor erkennt, welche virtuellen Maschine zu benachrichtigen ist. In dem Hypervisor kann hierzu eine Zuordnungstabelle gespeichert sein, durch welche jedem möglichen Nachrichtensignalisierung-Interrupt genau eine der virtuellen Maschinen zugeordnet ist.

[0026] Die Erfindung umfasst auch Ausführungsformen, durch die sich zusätzliche Vorteile ergeben.

[0027] Das „Filtern“ und das „Routing“ der Filter- und Routingeinheit besteht also darin, in einer Daten- nachricht vorbestimmte Identifikationsdaten zum Identifizieren derjenigen virtuellen Maschine, für welche die Datennachricht bestimmt ist, zu erkennen

(Filtern) und dann die Übertragung des in der Daten-
nachricht enthaltenen Datenpakets zu dieser virtuel-
len Maschine zu veranlassen (Routing). Die verwen-
deten Identifikationsdaten können beispielsweise in
einen bestimmten Nachrichtenabschnitt der Daten-
nachricht (beispielsweise X Bytes ab dem N-ten
Byte der Datennachricht) enthalten sein. Es kann
sich bei den Identifikationsdaten um eine IP-Adresse
und/oder einen Servicetyp (Kamera-Datenstrom,
Internetverkehr, Sprachsignal) handeln. Eine weitere
wichtige Erkenntnis der Erfindung besteht in diesem
Zusammenhang darin, dass die besagte Filter- und
Routingereinheit nicht durch zusätzliche Hardware in
der physikalischen Netzwerkschnittstelle bereitge-
stellt werden muss, sondern dass auf bereits in
einer gängigen physikalischen Netzwerkschnittstelle
bereitgestellte Hardware zurückgegriffen werden
kann.

[0028] Hierzu ist gemäß einer Ausführungsform die
Filter- und Routingereinheit für das Filtern auf der
Grundlage eines Content Addressable Memory,
CAM, insbesondere eines Ternary Content Adres-
sable Memory, TCAM, realisiert, durch welchen meh-
reren möglichen Speicherinhalten des Headers
jeweils eine der virtuellen Maschinen zugeordnet
ist, sodass für den Fall, dass der Header einen der
möglichen Speicherinhalten enthält, der CAM die
zugeordnete virtuelle Maschine signalisiert. Ein sol-
cher CAM ist auf gängiger Hardware einer physikali-
schen Netzwerkschnittstelle bereits verfügbar und
kann mit geringem Aufwand an die beschriebene
Funktion der Filter- und Routingereinheit angepasst
werden.

[0029] Hierzu sieht eine Ausführungsform vor, dass
die Filter- und Routingereinheit auf der Grundlage einer
Switching/Routing-Einheit der physikalischen Netz-
werkschnittstelle für ein Layer 2 und/oder Layer 3
und/oder Layer 4 Switching/Routing gebildet ist,
wobei durch einen Netzwerkprozessor der physikali-
schen Netzwerkschnittstelle eine Umprogrammierung
oder Umkonfiguration der Switching/Routing-
Einheit für das Erzeugen des Nachrichtensignalisie-
rung-Interrupts bewirkt ist. Da eine gängige physika-
lischen Netzwerkschnittstelle (NIC) eine solche Switch-
ing/Routing-Einheit und zudem einen
Netzwerkprozessor (programmierbare Rechenein-
richtung, beispielsweise zumindest ein Mikroprozes-
sor) aufweist, kann also mit einer gängigen physikali-
schen Netzwerkschnittstelle unter Nutzung von
deren Switching/RoutingEinheit die benötigte Filter-
und Routingereinheit realisiert werden. Denn auch für
ein Switching und/oder Routing ist das Filtern (Detek-
tion von vorbestimmten Identifikationsdaten in einer
Datennachricht) und das Weiterleiten eines Daten-
pakets bei Erkennen der vorbestimmten Identifika-
tionsdaten in einer Datennachricht (Routing)
ebenfalls vorgesehen. Der Unterschied besteht
darin, dass das Weiterleiten beim herkömmlichen

Routing in das Fahrzeugnetzwerk gerichtet ist, wäh-
rend erfindungsgemäß per DMA-Funktion zu einer
der virtuellen Maschinen weitergeleitet wird, was
vom Fachmann durch eine entsprechende Umkonfi-
guration im Sinne der Erfindungsidee erreicht wer-
den kann. Somit ist aber keine hardwaretechnische
Umgestaltung zur Realisierung der Erfindung not-
wendig. Hierzu kann der Netzwerkprozessor beim
Betriebsstart entsprechende Programminstruktionen
und/oder Konfigurationsdaten in die Switching/Rou-
ting-Einheit speichern. So kann z.B. eine Zuordnung
von den besagten Identifikationsdaten hin zu einem
jeweils zu verwendenden Nachrichtensignalisierung-
Interrupt erfolgen, sodass dann durch den Netzwerk-
prozessor lediglich der ausgegebene Nachrichten-
signalisierung-Interrupt ausgelöst werden muss.
Dies kann z.B. über eine PCIe-Schnittstelle oder
eine Interrupt-Auslöseroutine in bekannter Weise
erfolgen.

[0030] Die Kommunikation zwischen der physikali-
schen Netzwerkschnittstelle und dem Hypervisor
läuft über den beschriebenen Nachrichtensignalisie-
rung-Interrupt. Dieser kann als physikalischer Inter-
rupt (Legacy Interrupt) realisiert sein, was aber
bedeutet, dass jeder der mehreren möglichen Nach-
richtensignalisierung-Interrupts eine eigene elektri-
sche Leitung benötigt. Gemäß einer Ausführungs-
form ist stattdessen die physikalische
Netzwerkschnittstelle dazu eingerichtet ist, als den
Nachrichtensignalisierung-Interrupt einen Message-
Signaled-Interrupt, MSI, zu erzeugen. Somit können
mehrere verschiedene Nachrichtensignalisierung-
Interrupts über dieselben elektrischen Leitungen
übertragen werden (In-Band-Signalisierung). Damit
ist die Anzahl der benötigten elektrischen Leitungen
unabhängig von der Anzahl der virtuellen Maschi-
nen.

[0031] Nach einer bevorzugten Weiterbildung der
Erfindung ist vorgesehen, dass bei der Fahrzeugele-
ktronikeinheit

- ein erster Speicherabschnitt einer ersten der
virtuellen Maschinen (P1) in ihrem direkten
Speicherbereich einen RX-Puffer zum Spei-
chern von empfangenen Daten, und einen TX-
Puffer zum Speichern von zu sendenden Daten
(TX - Transmit; Senden) aufweist, und
- ein zweiter Speicherabschnitt einer zweiten
der virtuellen Maschinen in ihrem direkten Spei-
cherbereich einen RX-Puffer zum Speichern
von empfangenen Daten, und einen TX-Puffer
zum Speichern von zu sendenden Daten auf-
weist, wobei in der ersten und der zweiten vir-
tuellen Maschine deren jeweilige virtuelle Netz-
werkschnittstelle als ein Protokollstack
ausgestaltet ist, dessen unterste Layer (Ebene)
dazu eingerichtet ist, direkt auf den jeweiligen
RX-Puffer und TX-Puffer der virtuellen

Maschine zuzugreifen. Als unterste Layer kann die Layer 1 oder Layer 2 oder Layer 3 oder Layer 4 (gemäß dem an sich bekannten ISO/-OSI-Referenzmodell) vorgesehen sein.

[0032] Hiernach sind die beiden Speicherabschnitte jeweils weiter unterteilt in Speicherabschnitte nämlich in jeweils einen Signalisierungsbereich, einen RX-Puffer sowie einen TX-Puffer. In einem solchen Speicherabschnitt ist ein Software-Treiber installiert, bspw. für den IP-Datenverkehr oder für die Bearbeitung von Video- und/oder Grafikdaten. Von dem jeweiligen Protokollstack kann der RX-Puffer und der TX-Puffer jeweils in derselben Weise genutzt werden, als wäre der Protokollstack nicht in einer virtuellen Maschine implementiert, sondern als würde er direkt mit einer physikalischen Netzwerkschnittstelle kommunizieren. Somit muss also auch in der virtuellen Maschine keine aufwändige Anpassung an die vorgesehene individuelle Kommunikationsverbindung zwischen jeder virtuellen Maschine einerseits und der physikalischen Netzwerkschnittstelle andererseits vorgenommen werden.

[0033] Der Austausch von Daten mit dem Fahrzeugnetzwerk kann unterschiedliche Datentypen betreffen. So kann der erste Speicherabschnitt einen Speicherbereich für IP-Steuerdaten, und einen Speicherbereich für Nutzdaten, bspw. Video- und/oder Grafikdaten aufweisen und ebenso der zweite Speicherabschnitt mit einem Speicherbereich für IP-Steuerdaten und einen Speicherbereich für Nutzdaten, bspw. Video- und/oder Grafikdaten ausgebildet werden.

[0034] Nach einer weiteren bevorzugten Ausgestaltung der Erfindung ist wenigstens eine virtuelle Maschine für die DMA-Funktion mit einer PCIe-Schnittstelle zum Empfangen von in dem zugeordneten Speicherabschnitt zu speichernden Daten und Senden von Daten aus dem zugeordneten Speicherabschnitt konfiguriert.

[0035] Eine besonders vorteilhafte Weiterbildung der Erfindung sieht vor, dass wenigstens zwei der virtuellen Maschinen

- auf wenigstens zwei physikalischen Prozessoren ausgeführt sind, sodass jeder der beiden virtuellen Maschinen ein eigener physikalischer Prozessor zur Verfügung steht,
- den beiden Prozessoren jeweils wenigstens ein eigener physikalischer Speicher (z.B. RAM - Random Access Memory) zugeordnet ist, in welchem der jeweilige der virtuellen Maschine zugeordnete Speicherabschnitt angeordnet ist, und
- die beiden physikalischen Speicher jeweils für die DMA-Funktion eine PCIe-Schnittstelle zum Empfangen von zu speichernden Daten aus

der physikalischen Netzwerkschnittstelle und zum Senden von Daten über die physikalische Netzwerkschnittstelle aufweisen. Die Kombination aus physikalischen Prozessor und zugehörigem eigenem physikalischen Speicher kann beispielsweise als sogenanntes SOC (System on Chip) realisiert sein. Die Speicher der unterschiedlichen Prozessoren sind gemäß der Ausführungsform durch eine PCIe-Hardware miteinander gekoppelt, über welche die Übertragung von Daten (Datenpakete und Nachrichtensignalisierung-Interrupts) auch zwischen baulich getrennten physikalischen Speichern beispielsweise per DMA-Funktion (für Datenpakete) und MSI (für Nachrichtensignalisierung-Interrupts) möglich ist. Somit ist auch eine Nutzung einer einzigen gemeinsamen physikalischen Netzwerkschnittstelle durch mehrere getrennte Prozessoren ermöglicht.

[0036] Die erfindungsgemäße Fahrzeugelektronikeinheit kann als ein Steuergerät (ECU - Electronic Control Unit) für ein Kraftfahrzeug ausgestaltet sein. Die physikalische Netzwerkschnittstelle kann als eine sogenannte Netzwerkkarte oder NIC (Network Interface Controller) ausgestaltet sein. Die virtuellen Maschinen können jeweils durch eine sogenannte Partition (Software-Paket mit eigenem Dateisystem) bereitgestellt sein. Die virtuelle Maschinen können auf einem einzelnen gemeinsamen Prozessor (mit einem oder mehreren Prozessorkernen) oder auf einem Mehrprozessorsystem oder auch auf voneinander baulich getrennten Prozessoren (SOCs) betrieben werden.

[0037] Die zweitgenannte Aufgabe wird durch ein erfindungsgemäßes Verfahren gelöst.

[0038] Nach der erstgenannten Lösung wird das Datenkommunikationsverfahren zwischen den virtuellen Maschinen der erfindungsgemäßen Fahrzeugelektronikeinheit und der Netzwerkschnittstelle zum Empfangen wenigstens einer ein Datenpaket mit einem Header aufweisenden Datennachricht durchgeführt.

[0039] Das erfindungsgemäße Datenkommunikationsverfahren dient zur Bereitstellung der besagten Kommunikationsverbindung zwischen den virtuellen Maschinen der Fahrzeugelektronikeinheit einerseits und der physikalischen Netzwerkschnittstelle der Fahrzeugelektronikeinheit andererseits, wobei zum Empfangen wenigstens einer ein Datenpaket mit einem Header aufweisenden Datennachricht entsprechend einem Netzwerkprotokoll folgende Verfahrensschritte durchgeführt werden:

- Speichern des Datenpaketes der Datennachricht in einem RX-Puffer der physikalischen Netzwerkschnittstelle,

- Auswerten des Headers der Datennachricht mittels einer Filter- und Routingeinheit hinsichtlich einer das Datenpaket zuzuweisenden Partition einer der virtuellen Maschinen,

- Kopieren des Datenpaketes mittels einer DMA-Funktion in den direkten Speicherbereich des Speicherabschnitts der der Header-Information entsprechenden virtuellen Maschine, und

- Erzeugen eines Nachrichtensignalisierung-Interrupts für einen Hypervisor, wobei in Abhängigkeit davon, welcher virtuellen Maschine das Datenpaket zuzuweisen ist, ein anderer Nachrichtensignalisierung-Interrupt erzeugt wird,

- Bereitstellen eines virtuellen Interrupts (z.B. mittels einer MSI-Interrupt-Steuerungsfunktion) für die der Header-Information entsprechenden virtuellen Maschine, wobei der virtuelle Interrupt unmittelbar in Abhängigkeit davon erzeugt wird, welcher Nachrichtensignalisierung-Interrupt eintrifft, und

- Verarbeiten des Datenpaketes durch die der Header-Information entsprechenden virtuellen Maschine mittels deren virtueller Netzwerkschnittstelle, die hierzu direkt auf den direkten Speicherbereich (insbesondere einen dort bereitgestellten RX-Puffer) zugreift.

[0040] Bei diesem erfindungsgemäßen Datenkommunikationsverfahren zum Empfangen einer Datennachricht aus dem Fahrzeugnetzwerk mittels der Netzwerkschnittstelle übernimmt bspw. eine Netzwerkkarte oder ein Netzwerkprozessor die Aufgabe, die von der Netzwerkschnittstelle empfangenen Datenpakete einer Datennachricht in denjenigen Speicherabschnitt der der Header-Information entsprechenden virtuellen Maschine zu schreiben. Hierzu wird das Datenpaket in einen RX-Puffer der Netzwerkschnittstelle geschrieben, der Header der Datennachricht ausgewertet und mittels eines MSI-Interrupts die entsprechende Partition informiert.

[0041] Wie bereits ausgeführt, wird gemäß einer Ausführungsform die DMA-Funktion durch eine PCIe-Schnittstelle bereitgestellt. In einem Speicherabschnitt einer virtuellen Maschine können unterschiedliche Applikationen implementiert werden, wie bspw. die Verarbeitung von Steuerdaten für den IP-Datenverkehr und die Verarbeitung von Video- und/oder Grafikdaten. Es können die Datenpakete entsprechend der Header-Informationen jeweils dem richtigen Speicherabschnitt einer virtuellen Maschine zugewiesen werden.

[0042] Nach der zweitgenannten Lösung wird das Datenkommunikationsverfahren zwischen den virtuellen Maschinen der erfindungsgemäßen Fahrzeugelektronikeinheit und der Netzwerkschnittstelle zum Senden wenigstens einer ein Datenpaket mit einem

Header aufweisende Datennachricht durchgeführt. Bisher wurde nur das Empfangen einer Datennachricht beschrieben. Eine Ausführungsform des Datenkommunikationsverfahrens sieht vor, dass zum Senden wenigstens einer ein Datenpaket mit einem Header aufweisenden Datennachricht entsprechend einem Netzwerkprotokoll folgende Verfahrensschritte durchgeführt werden:

- Erzeugen des an die physikalische Netzwerkschnittstelle weiterzuleitenden Datenpaketes durch eine der virtuellen Maschinen und Speichern des Datenpaketes in dem zugehörigen direkten Speicherbereich des der virtuellen Maschine zugeordneten Speicherabschnitts,

- Erzeugen einer die Bereitstellung des weiterzuleitenden Datenpaketes anzeigenden Nachricht für die DMA-Funktion,

- Kopieren des Datenpaketes in einen TX-Puffer der physikalischen Netzwerkschnittstelle mittels der DMA-Funktion,

- Ergänzen des Datenpaketes mit einer Header-Information entsprechenden Information mittels der physikalischen Netzwerkschnittstelle,

- Erstellen der Datennachricht aus dem Datenpaket mit Header entsprechend dem Netzwerkprotokoll mittels der physikalischen Netzwerkschnittstelle, und

- Senden der Datennachricht mittels der physikalischen Netzwerkschnittstelle.

[0043] Bei diesem erfindungsgemäßen Datenkommunikationsverfahren zum Übermitteln einer Datennachricht aus einer der virtuellen Maschinen an die Netzwerkschnittstelle des Fahrzeugnetzwerkes übernimmt bspw. ebenso eine Netzwerkkarte oder ein Netzwerkprozessor die Aufgabe, die an die Netzwerkschnittstelle weiterzuleitenden Datenpakete aus einem Speicherabschnitt der jeweiligen virtuellen Maschine auszulesen und in einen TX-Puffer der Netzwerkschnittstelle zu kopieren. Mittels der Netzwerkschnittstelle wird das Datenpaket mittels eines Headers zu einer Datennachricht entsprechend des Netzwerkprotokolls ergänzt und in das Fahrzeugnetzwerk gesendet.

[0044] Die Erfindung umfasst auch Weiterbildungen des erfindungsgemäßen Verfahrens, die Merkmale aufweisen, wie sie bereits in Zusammenhang mit den Weiterbildungen der erfindungsgemäßen Fahrzeugelektronikeinheit beschrieben wurden. Aus diesem Grund sind die entsprechenden Weiterbildungen des Erfinders gemessen Verfahrens hier nicht noch einmal beschrieben.

[0045] Die Erfindung umfasst auch ein Kraftfahrzeug mit einer Ausführungsform der erfindungsgemäßen Fahrzeugelektronikeinheit. Das erfindungs-

gemäße Kraftfahrzeug ist bevorzugt als Kraftwagen, insbesondere als Personenkraftwagen oder Lastkraftwagen, oder als Personenbus oder Motorrad ausgestaltet.

[0046] Weitere Vorteile, Merkmale und Einzelheiten der Erfindung ergeben sich aus den Ansprüchen, der nachfolgenden Beschreibung bevorzugter Ausführungsformen sowie anhand der Zeichnungen. Dabei zeigen:

Fig. 1 ein Blockschaltbild einer Fahrzeugelektronikeinheit mit einem Hypervisor und zwei virtuellen Maschinen,

Fig. 2 ein Blockschaltbild einer weiteren Fahrzeugelektronikeinheit mit einem Hypervisor und zwei virtuellen Maschinen mit jeweils mehreren Speicherabschnitten,

Fig. 3 ein detailliertes Blockschaltbild einer Fahrzeugelektronikeinheit gemäß **Fig. 1**,

Fig. 4 ein Blockschaltbild zur Darstellung einer Datenkommunikation einer über ein Fahrzeugnetzwerk empfangenen Datennachricht innerhalb der Fahrzeugelektronikeinheit nach **Fig. 2**, und

Fig. 5 ein Blockschaltbild zur Darstellung einer Datenkommunikation einer über ein Fahrzeugnetzwerk zu sendenden Datennachricht innerhalb der Fahrzeugelektronikeinheit nach **Fig. 2**.

[0047] Bei den im Folgenden erläuterten Ausführungsbeispielen handelt es sich um bevorzugte Ausführungsformen der Erfindung. Bei den Ausführungsbeispielen stellen die beschriebenen Komponenten der Ausführungsformen jeweils einzelne, unabhängig voneinander zu betrachtende Merkmale der Erfindung dar, welche die Erfindung jeweils auch unabhängig voneinander weiterbilden. Daher soll die Offenbarung auch andere als die dargestellten Kombinationen der Merkmale der Ausführungsformen umfassen. Des Weiteren sind die beschriebenen Ausführungsformen auch durch weitere der bereits beschriebenen Merkmale der Erfindung ergänzbar.

[0048] In den Figuren bezeichnen gleiche Bezugszeichen jeweils funktionsgleiche Elemente.

[0049] Die Fahrzeugelektronikeinheit 10 nach **Fig. 1** und **Fig. 2** zeigt zwei virtualisierte Steuergeräte als erste virtuelle Maschine P1 und zweite virtuelle Maschine P2 mit jeweils einem Betriebssystem (OS), wobei die beiden virtuellen Maschinen P1 und P2 mittels eines Hypervisors 1 (auch Virtual Machine Monitor oder VMM genannt) getrennt werden. Ein solcher Hypervisor 1 ist eine Software, mit welcher für jedes Betriebssystem der beiden virtuellen Maschinen P1 und P2 eine Systemumgebung mit virtuellen Ressourcen einschließlich virtuellen Prozess-

oren zur Realisierung und Steuerung der Betriebssysteme erzeugt werden. Zur Realisierung einer solchen Systemumgebung durch den Hypervisor 1 werden entsprechende Hardwarekomponenten, wie ein Mikroprozessor 4.0 und ein Speicher 3 der Fahrzeugelektronikeinheit 10 verwendet, wobei der Speicher 3 als gemeinsamer Speicher von beiden virtuellen Maschinen P1 und P2 benutzt wird.

[0050] Nach **Fig. 1** wird dieser gemeinsame Speicher 3 in zwei Teile aufgeteilt, nämlich in einen ersten Speicherabschnitt 3.1, auf welchen nur von der virtuellen Maschine P1 zugegriffen werden kann, und einen zweiten Speicherabschnitt 3.2, auf welche nur von der virtuellen Maschine P2 zugegriffen werden kann. Für eine von einer virtuellen Maschine P1 oder P2 ausgeführte Applikation, wie bspw. Steuerung des IP-Datenverkehrs oder Bearbeitung von Video- und/oder Grafikdaten wird ein RX-Pufferbereich sowie ein TX-Pufferbereich als Speicherbereich 3.10 des ersten Speicherabschnittes 3.1 und ein RX-Pufferbereich sowie ein TX-Pufferbereich als Speicherabschnitt 3.20 des zweiten Speicherabschnittes 3.2 und jeweils ein Signalisierungsbereich 3.01 und 3.02 zum Signalisieren, ob RX-Daten oder TX-Daten empfangen werden, bereitgestellt. Die Funktion dieser Signalisierungsbereiche 3.01 und 3.02 wird anhand des anhand von **Fig. 4** beschriebenen Datenkommunikationsverfahrens erläutert. Diese Signalisierungsbereiche 3.01 und 3.02 können auch dadurch ersetzt werden, dass definierte Interrupts durchgeführt werden.

[0051] Eine Datenkommunikation zwischen jeweils einer virtuellen Maschine P1 oder P2 und einer Netzwerkschnittstelle 2.1 zu einem Fahrzeugnetzwerk B1, bspw. einem CAN-Bus erfolgt mittels eines von dem Hypervisor 1 realisierten virtuellen Netzwerks 5, welches in **Fig. 1** und **Fig. 2** als virtuelle Netzwerkschnittstellen 5.1 und 5.2 dargestellt ist, während die Netzwerkschnittstelle 2.1 von einer Netzwerkkarte 2 realisiert wird.

[0052] Bei dieser Datenkommunikation werden Datenpakete von Datennachrichten aus dem Fahrzeugnetzwerk B1 mittels des virtuellen Netzwerks 5. von einem RX-Puffer der Netzwerkkarte 2 direkt in den ersten Speicherabschnitt 3.1 der ersten virtuellen Maschine P1 oder in den zweiten Speicherabschnitt 3.2 der zweiten virtuellen Maschine P2 in Abhängigkeit einer Header-Information der empfangenen Datennachricht geschrieben. Auch umgekehrt wird ein Datenpaket einer zu sendenden Datennachricht aus einer der beiden Speicherabschnitte 3.1 oder 3.2 direkt in einen TX-Puffer der Netzwerkkarte 2 geschrieben und anschließend als Datennachricht entsprechend dem Netzwerkprotokoll über die Netzwerkschnittstelle 2.1 gesendet.

[0053] Nach **Fig. 2** wird der gemeinsame Speicher 3 ebenso in zwei Teile aufgeteilt, nämlich in einen ersten Speicherabschnitt 3.1, auf welchen nur von der virtuellen Maschine P1 zugegriffen werden kann, und einen zweiten Speicherabschnitt 3.2, auf welchen nur von der virtuellen Maschine P2 zugegriffen werden kann. Ferner sind für jeden Speicherabschnitt 3.1 und 3.2 jeweils zwei Speicherbereiche vorgesehen, nämlich Speicherbereiche 3.10 und 3.11 für den ersten Speicherabschnitt 3.1 und Speicherbereiche 3.20 und 3.21 für den zweiten Speicherabschnitt 3.2, um damit unterschiedliche Applikationen zu realisieren. Auch bei dieser Fahrzeugelektronikeinheit 10 nach **Fig. 2** weist jeder Speicherabschnitt 3.1 und 3.2 jeweils ein Signalisierungsbereich 3.01 bzw. 3.02 auf.

So dient bspw. der Speicherbereich 3.10 des ersten Speicherabschnittes 3.1 und der Speicherbereich 3.20 des zweiten Speicherabschnittes 3.2 zur Steuerung des IP-Datenverkehrs. Für den entsprechenden Software-Treiber ist ein RX-Pufferbereich sowie ein TX-Pufferbereich als Speicherbereich 3.10 bzw. 3.20 eingerichtet.

[0054] Der Speicherbereich 3.11 des ersten Speicherabschnittes 3.1 und der Speicherbereich 3.21 des zweiten Speicherabschnittes 3.2 dient bspw. zur Bearbeitung von Video- und/oder Grafikdaten. Für den entsprechenden Software-Treiber ist ein RX-Pufferbereich sowie ein TX-Pufferbereich als Speicherabschnitt 3.11 bzw. 3.21 eingerichtet.

[0055] Der Datenverkehr zwischen den Speicherabschnitten 3.10 und 3.11 bzw. 3.20 und 3.21 des ersten oder zweiten Speicherabschnittes 3.1 bzw. 3.2 einerseits und der Netzwerkschnittstelle 2.1 andererseits erfolgt entsprechend der Hypervisor-Architektur nach **Fig. 1** bzw. **Fig. 2** über das von dem Hypervisor 1 realisierte virtuelle Netzwerk 5, und zwar direkt zwischen einem Speicherbereich 3.10, 3.11, 3.20 bzw. 3.21 über einen RX- bzw. TX-Puffer der Netzwerkkarte 2 und der Netzwerkschnittstelle 2.1.

[0056] Bei diesen Hypervisor-Architekturen gemäß den **Fig. 1** und **Fig. 2** kann eine virtuelle Maschine P1 bzw. P2 nur auf den ihr zugeordneten Speicherabschnitt 3.1 bzw. 3.2 zugreifen. Dies führt zu einer klaren Trennung der zu kommunizierenden Datenpakete bezüglich deren Zugehörigkeit zur virtuellen Maschine P1 oder virtuellen Maschine P2, wodurch negative Einflüsse hinsichtlich Sicherheit des Datenverkehrs vermieden werden. Auch treten keine negativen Einflüsse auf die Leistung und die Funktionssicherheit solcher Hypervisor-Architekturen auf.

[0057] Die **Fig. 3** zeigt eine detaillierte Realisierung einer Fahrzeugelektronikeinheit 10 mit einer Hypervisor-Architektur gemäß **Fig. 2**. Diese Fahrzeugelektronikeinheit 10 besteht aus einer MMX (Multi Media Extension)-Steckkarte 10.1 und einer RSC

(Radio Sound Control)-Steckkarte, die auf einem Board eines Zentralrechners angeordnet sind.

[0058] Mittels der MMX-Steckkarte 10.1 wird ein Hypervisor 1 realisiert, welcher eine erste virtuelle Maschine P1 und eine zweite virtuelle Maschine P2 trennt. Zur Realisierung der beiden virtuellen Maschinen P1 und P2 ist ein Mikroprozessor 4.0 sowie ein gemeinsamen Speicher 3 vorgesehen, welcher in zwei Abschnitte geteilt ist, nämlich in einen ersten Speicherabschnitt 3.1 und in einen zweiten Speicherabschnitt 3.2, wobei auf den ersten Speicherabschnitt 3.1 nur die erste virtuelle Maschine P1 und auf den zweiten Speicherabschnitt 3.2 nur die zweite virtuelle Maschine P2 zugreifen kann.

[0059] Innerhalb dieser Speicherabschnitte 3.1 und 3.2 ist eine weitere Aufteilung in Speicherbereiche 3.10 und 3.11 bzw. 3.20, 3.21 und 3.22 vorgenommen, nämlich für Steuerdaten des IP-Datenverkehrs und Video- und/oder Grafikdaten.

[0060] Der Speicherbereich 3.10 des ersten Speicherabschnittes 3.1 und die Speicherbereiche 3.20 und 3.22 des zweiten Speicherabschnittes 3.2 dienen zur Steuerung des IP-Datenverkehrs. Für die entsprechenden Software-Treiber ist ein RX-Pufferbereich sowie ein TX-Pufferbereich als Speicherabschnitte 3.10 bzw. 3.20 und 3.22 eingerichtet. Für jeden dieser Speicherabschnitte 3.10, 3.20 und 3.22 ist ein Software-Treiber mit einem zugehörigen Protokoll-Stack 3.100, 3.200 und 3.220 mittels des Hypervisors 1 realisiert.

[0061] Der Speicherbereich 3.11 des ersten Speicherabschnittes 3.1 und Speicherbereich 3.21 des zweiten Speicherabschnittes 3.2 dienen zur Bearbeitung von Video- und/oder Grafikdaten. Für den entsprechenden Software-Treiber ist ein RX-Pufferbereich sowie ein TX-Pufferbereich als Speicherabschnitt 3.11 bzw. 3.21 eingerichtet. Für jeden dieser Speicherbereiche 3.11 und 3.21 ist ein Software-Treiber mit einem zugehörigen Protokoll-Stack 3.110 und 3.210 mittels des Hypervisors 1 realisiert.

[0062] Jeder der beiden Speicherabschnitte 3.1 und 3.2 weist natürlich auch Signalisierungsbereiche entsprechend der Darstellung nach **Fig. 2** auf, die jedoch zugunsten einer besseren Übersichtlichkeit in **Fig. 3** nicht dargestellt sind.

[0063] Die RSC-Steckkarte 10.2 besteht aus einer Netzwerkkarte 2 mit einem Mikroprozessor 4.1 und einer Filter- und Routingeinheit 2.3, mit welcher eine Ethernet-Schnittstelle als Netzwerkschnittstelle 2.1 über DMA (Direkt Memory Access)-Ausgängen 2.30 mit einer PCIe-Schnittstelle 7 und mit Rechenknoten 10.20 und 10.21 verbunden wird. Die beiden

Rechenknoten 10.20 und 10.21 weisen jeweils einen Mikroprozessor 4.2 bzw. 4.3 auf.

[0064] Die Netzwerkschnittstelle 2.1 ist über einen Gateway/Ethernet-Switch 11.1 eines Gateways 11 einerseits mit einem Kombiinstrument (FPK), einem Head-up-Display (HUD), einer Rückfahrkamera (RVC) und einer C-Box usw. verbunden und andererseits über das Gateway 11 mit einem Gateway eines CAN-Busses als Fahrzeugnetzwerk B1.

[0065] Ein DMA-Ausgang der Filter- und Routingeinheit 2.3 wird über die PCIe-Schnittstelle 7 und einen PCI-Treiber 6 mit dem Speicherbereich 3.20 des zweiten Speicherabschnittes 3.2 verbunden, ein weiterer DMA-Ausgang ist ebenso über die PCIe-Schnittstelle 7 und den PCI-Treiber 6 mit dem Speicherbereich 3.22 des zweiten Speicherabschnittes 3.2 jeweils als virtuelle Netzwerkschnittstelle 5.2 verbunden.

[0066] Ein weiterer DMA-Ausgang der Filter- und Routingeinheit 2.3 führt auf ein Softwaremodul 4.10 der Netzwerkkarte 2 für Videodaten und einen DMA-Speicher, dessen verarbeitete Videodaten über einen MPEG-TS-Treiber 4.11 mittels der PCIe-Schnittstelle 7 und des PCIe-Treibers 6 mit der Speichersektion 3.11 des ersten Speicherabschnittes 3.1 als virtuelle Netzwerkschnittstelle 5.1 verbunden ist.

[0067] Ein weiterer DMA-Ausgang ist mit einem Softwaremodul 4.20 des Rechenknotens 10.20 verbunden, welches eine Bedien- und Anzeigesoftware sowie einem zugehörigen DMA-Speicher ist.

[0068] Schließlich ist ein weiteres Softwaremodul 4.12 für das Internet Netzwerk Management einen weiteren DMA-Ausgang der Filter- und Routingeinheit 2.3 angeschlossen.

[0069] Der Rechenknoten 10.21 weist ein Softwaremodul 4.30 mit einer Ethernet-Software, einem zugehörigen DMA-Speicher zur Verarbeitung von IP-Steuerdaten sowie mit einem Protokoll-Stack 4.31 für den IP-Datenverkehr auf, wobei dieser Protokoll-Stack 4.31 Ausgänge 4.32 für weitere Softwaremodule aufweist.

[0070] Eine MOST-Schnittstelle 4.4 verbindet ein MOST-Netzwerk als weiteres Fahrzeugnetzwerk B2 über ein Softwaremodul 4.34 mit einem MPEG-Treiber zur Übertragung von Video- und/oder Grafikdaten über die PCIe-Schnittstelle 7 und den PCIe-Treiber 6 der ersten virtuellen Maschine P1 mit der Speichersektion 3.21 des zweiten Speicherabschnittes 3.2 zur Realisierung einer weiteren virtuellen Netzwerkschnittstelle 5.2.

[0071] Schließlich ist ein Softwaremodul 4.33 mit einer Ethernet-Datenkommunikationssoftware und

einem zugehörigen Protokoll-Stack des Rechenknotens 10.21 mit der Speichersektion 3.10 des ersten Speicherabschnittes 3.1 der ersten virtuellen Maschine P1 zur Realisierung einer weiteren virtuellen Netzwerkschnittstelle 5.1 verbunden.

[0072] Im Folgenden wird der interne Datenverkehr der Fahrzeugelektronikeinheit 10 gemäß **Fig. 3** anhand der **Fig. 4** und **Fig. 5** beschrieben.

[0073] Bei dieser Fahrzeugelektronikeinheit 10 gemäß den **Fig. 4** und **Fig. 5** weist diese einen Hypervisor 1 neben einer ersten und zweiten virtuellen Maschine P1 und P2 weitere virtuelle Maschinen bis zur virtuellen Maschine Px auf. Jeder dieser virtuellen Maschinen P1, ... Px ist ein Speicherabschnitt 3.1, 3.2, ... 3.x eines gemeinsamen Speichers 3 zugeordnet.

[0074] Die Fahrzeugelektronikeinheit 10 gemäß den **Fig. 4** und **Fig. 5** umfasst einen Rechenknoten als MMX-Steckkarte 10.1 sowie eine Netzwerkkarte 2 entsprechend der Struktur nach **Fig. 3**.

[0075] Nach **Fig. 4** empfängt die Fahrzeugelektronikeinheit 10 aus einem Fahrzeugnetzwerk B1 über eine Netzwerkschnittstelle 2.1 der Netzwerkkarte 2 eine Datennachricht, die neben einem Datenpaket einen Header und ein Nachrichtensignalisierungs-Interrupt (MSI = Message Signaled Interrupt) entsprechend dem Netzwerkprotokoll, bspw. dem Ethernet-Protokoll des an die Netzwerkschnittstelle 2.1 angeschlossenen Fahrzeugnetzwerkes B1, bspw. einem CAN-Bus umfasst.

[0076] Das Datenpaket (oder auch mehrere) wird von der Netzwerkschnittstelle 2.1 in einen RX-Puffer 2.2 der Netzwerkkarte 2 geschrieben, wobei gleichzeitig von der Netzwerkschnittstelle 2.1 eine Eingangsmeldung einem Ethernet-Treiber 2.5 übermittelt wird, welcher ein Triggersignal an die Filter- und Routingeinheit 2.3 sendet. Der Header wird zur Analyse von dessen Header-Information von dem RX-Puffer 2.2 an die Filter- und Routingeinheit 2.3 gesendet und hinsichtlich der der Header-Information entsprechenden Zuordnung zu einer virtuellen Maschine P1, ... Px, insbesondere zu einem Speicherabschnitt der Speicherabschnitte 3.1, ... 3.x ausgewertet. Die Filter- und Routing-Regeln werden von dem Hypervisor 1 der Filter- und Routingeinheit 2.3 bereitgestellt.

[0077] Ein Triggersignal der Filter- und Routingeinheit 2.3 veranlasst einen PCIe-Treiber 2.4 der Netzwerkkarte 2 mittels einer Speicherdirektzugriffs (DMA)-Funktion das Datenpaket aus dem RX-Puffer 2.2 über die DMA-Ausgänge 2.30 der Filter- und Routingeinheit 2.3 und die PCIe-Schnittstelle 7 in den der Header-Information entsprechenden Speicherbereich 3.1, ... oder 3.x zu kopieren. Ferner

erhält ein Signalisierungsbereich 3.0 von dem PCIe-Treiber 2.4 der Netzwerkkarte 2 das Nachrichtensignalisierungs-Interrupt und leitet es weiter an den PCIe-Treiber 6 der MMX-Steckkarte 10.1. Mittels des Signalisierungsbereichs 3.0 wird signalisiert, ob RX- oder TX-Daten übermittelt werden. Der Signalisierungsbereich 3.0 umfasst die Signalisierungsbereiche 3.01, 3.02, ... 3.0x der Speicherabschnitte 3.1, 3.2, ... 3.x. Dieser Signalisierungsbereich 3.0 kann auch dadurch ersetzt werden, dass definierte Interrupt durchgeführt werden.

[0078] Eine MSI-Interrupt-Steuerungsfunktion wird von dem Hypervisor 1 als MSI-Handler realisiert, welchem das Nachrichtensignalisierungs-Interrupt von dem PCIe-Treiber 6 zugeführt wird. Von diesem MSI-Handler wird das Nachrichtensignalisierungs-Interrupt verarbeitet und die der Header-Information entsprechende virtuelle Maschine P1, ... oder Px mittels eines virtuellen Interrupts informiert. Anschließend wird von dieser virtuellen Maschine P1, ... oder Px auf deren zugeordneten Speicherbereich 3.1, ... oder 3.x zugegriffen und das Datenpaket verarbeitet.

[0079] Datenpakete, die als Datennachricht über die Netzwerkschnittstelle 2.1 in das Fahrzeugnetzwerk B1 zu senden sind, werden von einer Applikation in einer der virtuellen Maschinen P1, ... oder Px erzeugt und in dem zugehörigen Speicherabschnitt 3.1, ... oder 3.x des gemeinsamen Speichers 3 gespeichert. Auf die anderen Speicherabschnitte kann diese eine virtuelle Maschine nicht zugreifen.

[0080] Gleichzeitig wird von dieser virtuellen Maschine, welche das an die Netzwerkschnittstelle 2.1 zu übermittelnde Datenpaket erzeugt und in den dieser virtuellen Maschine zugeordneten Speicherbereich abgelegt hat, ein TX-Data-Ready-Signal an den PCIe-Treiber 6 gesendet, um anzuzeigen, dass ein Datenpaket zur Übernahme in den entsprechenden Speicherbereich bereit steht. Diese TX-Data-Ready-Signale werden von allen virtuellen Maschinen P1, ... Px zusammengefasst und über die PCIe-Schnittstelle 7 dem PCIe-Treiber 2.4 der Netzwerkkarte 2 zugeführt, wodurch mittels der Speicherdirektzugriffs (DMA)-Funktion der DMA-Ausgänge 2.30 mittels der PCIe-Schnittstelle 7 das Datenpaket in einen TX-Puffer 2.30 der Netzwerkkarte 2 kopiert wird. Sind Datenpakete von mehreren virtuellen Maschinen P1, ... Px in den jeweiligen Speicherabschnitten 3.1, ... 3.x abgelegt, werden alle Datenpakete in den TX-Puffer 2.20 kopiert.

[0081] Ein von dem PCIe-Treiber 2.4 erzeugtes Sende-Trigger-Signal wird an den Ethernet-Treiber 2.5 gesendet, welcher die Netzwerkschnittstelle 2.1 dazu veranlasst, für das Datenpaket, insbesondere das Ethernet-Datenpaket einen Header entsprechend den Informationen des PCIe-Treibers 6 zu

erzeugen, eine Datennachricht aus dem Datenpaket und dem Header entsprechend dem Netzwerkprotokoll, bspw. dem Ethernet-Protokoll zu erstellen und diese derart erstellte Datennachricht zu senden.

[0082] In der Fahrzeugelektronikeinheit 10 gemäß Fig. 3 werden die beiden virtuellen Maschinen P1 und P2 mittels eines einzigen Mikroprozessors 4.0 und einem einzigen Speicher 3 durchgeführt, wobei diesem Speicher 3 nur eine einzige PCIe-Schnittstelle 7 zugeordnet ist.

[0083] Es ist auch möglich die beiden virtuellen Maschinen P1 und P2 mit wenigstens zwei unterschiedlichen Prozessoren zu realisieren, denen jeweils ein Speicher zugeordnet ist. Damit stehen wenigstens zwei unterschiedliche Speicher zur Verfügung, denen jeweils PCIe-Schnittstellen zum Empfangen von zu speichernden Daten und zum Senden von Daten über die physikalische Netzwerkschnittstelle 2.1 zugeordnet sind.

[0084] Die zugrunde liegende Idee ist somit, den PCIe / Shared Memory Bereich und den Hypervisor zu nutzen. Dazu wird der PCIe / Shared Memory in zwei Teile aufgeteilt. Ein Speicherbereich ist nur von der Partition 1 (virtuelle Maschine P1) zugreifbar, der andere Speicherbereich nur von der Partition 2 (virtuelle Maschine P2). Innerhalb dieser Speicherbereiche kann eine zusätzliche Unterteilung in Kontrolldaten (z.B. IP-Traffic) und Videodaten erfolgen. Auf jede dieser Unterteilungen kann ein Treiber zugreifen, sodass dieser Treiber einen eigenen RX/TX-Konfigurationsbereich definiert erhält. Die Netzwerkkarte oder ein Netzwerkprozessor kann die Aufgabe übernehmen, die Daten vom Netzwerk-Interface (Fahrzeugnetzwerk) in den entsprechenden Speicherbereich in der jeweiligen Partition zu kopieren. Dazu schreibt er Daten in den RX-Puffer und liest Daten aus dem TX-Puffer. Die verschiedenen Treiber oberhalb der Puffer werden über Interrupts informiert. Durch die Nutzung unterschiedlicher Interrupts (sowohl bei MSI Interrupts als auch bei Legacy Interrupts) kann der Hypervisor unterscheiden, welche Gast-Partition (virtuelle Maschine) einen entsprechenden virtuellen Interrupt erhalten muss. Das Erzeugen der virtuellen Interrupts in Abhängigkeit von den eintreffenden Nachrichtensignalisierungs-Interrupts kann (anstelle des Hypervisors) durch einen Interrupt-Controller durchgeführt werden, der hierzu beispielsweise durch den Hypervisor konfiguriert werden kann.

[0085] Darüber hinaus kann der Netzwerkkarten-Controller verschiedene Perioden/Zeitabstände des Sendens von Interrupts für unterschiedliche der virtuellen Maschinen verwenden, sodass die Leistung auf einer Partition optimiert werden kann (z.B. 1 Interrupt für 100 Datenpakete für Partition 2), wäh-

rend die andere Partition Interrupts pro Datenpaket empfängt, um eine geringe Latenz zu gewährleisten.

Bezugszeichenliste

1	Hypervisor
1.1	MSI-Handler
2	Netzwerkkarte
2.1	Netzwerkschnittstelle
2.2	RX-Puffer der Netzwerkkarte 2
2.20	TX-Puffer der Netzwerkkarte 2
2.3	Filter- und Routingeinheit
2.30	DMA-Ausgänge der Filter-und Routingeinheit 2.3
2.4	PCIe-Treiber
2.5	Ethernet-Treiber
3	gemeinsamer Speicher
3.0	Signalisierungsbereich
3.01	Signalisierungsbereich
3.02	Signalisierungsbereich
3.0x	die Finanzierungsbereich
3.1	erste Speicherabschnitt der ersten virtuellen Maschine P1
3.10	Speicherbereich des ersten Speicherabschnittes 3.1
3.100	Protokoll-Stack
3.11	Speicherbereich des ersten Speicherabschnittes 3.1
3.110	Protokoll-Stack
3.2	zweiter Speicherabschnitt der zweiten virtuellen Maschine P2
3.20	Speicherbereich des zweiten Speicherabschnittes 3.2
3.200	Protokoll-Stack
3.21	Speicherbereich des zweiten Speicherabschnittes 3.2
3.210	Protokoll-Stack
3.22	Speicherbereich des zweiten Speicherabschnittes 3.2
3.220	Protokoll-Stack
3.x	Speicherabschnitt der virtuellen Maschine Px
4.0	Mikroprozessor
4.1	Mikroprozessor
4.10	Softwaremodul

4.11	MPEG-TS-Treiber
4.2	Mikroprozessor
4.20	Softwaremodule
4.3	Mikroprozessor
4.30	Softwaremodule
4.31	Protokoll-Stack
4.32	Ausgänge des Protokoll-Stacks 4.31
4.33	Softwaremodule
4.34	Softwaremodul
4.4	MOST-Schnittstelle
5	virtuelles Netzwerk
5.1	virtuelle Netzwerkschnittstelle(n) des virtuellen Netzwerks 5
5.2	virtuelle Netzwerkschnittstelle(n) des virtuellen Netzwerks 5
6	PCIe-Treiber
7	PCIe-Schnittstelle
10	Fahrzeugelektronikeinheit
10.1	MMX-Steckkarte
10.2	RSC-Steckkarte
10.20	Rechenknoten
10.21	Rechenknoten
11	Gateway
11.1	Gateway/Ethernet-Switch
B1	Fahrzeugnetzwerk
B2	Fahrzeugnetzwerk
P1	erste virtuelle Maschine
P2	zweite zweite virtuellen Maschine
Px	virtuelle Maschine

Patentansprüche

1. Fahrzeugelektronikeinheit (10) umfassend
 - wenigstens zwei virtuellen Maschinen (P1, P2) mit jeweils einer virtuellen Netzwerkschnittstelle (5.1, 5.2),
 - einen Hypervisor (1), welcher zur Trennung der virtuellen Maschinen (P1, P2) ausgebildet ist,
 - einen Speicher (3) mit jeweils einer der virtuellen Maschinen (P1, P2) exklusiv zugeordneten Speicherabschnitten (3.1, 3.2), wobei jeder Speicherabschnitt (3.1, 3.2) direkte Speicherbereiche (3.10, 3.20) für die zugeordnete virtuelle Netzwerkschnittstelle (5.1, 5.2) aufweist, und
 - eine physikalische Netzwerkschnittstelle (2.1) zur Herstellung der jeweiligen Kommunikationsverbindung zwischen der jeweiligen virtuellen Netzwerk-

schnittstelle (5.1, 5.2) der virtuellen Maschinen (P1, P2) einerseits und einem Fahrzeugnetzwerk (B1) andererseits, **dadurch gekennzeichnet**, dass

- die physikalische Netzwerkschnittstelle (2.1) zur Herstellung der jeweiligen Kommunikationsverbindung dazu eingerichtet ist, für aus dem Fahrzeugnetzwerk (B1) eintreffende Datennachrichten jeweils einen eine Header-Information aufweisenden Header der Datennachricht mittels einer Filter- und Routingeinheit (2.3) hinsichtlich der virtuellen Maschine (P1, P2), welcher ein Datenpaket der Datennachricht zuzuweisen ist, zu analysieren, und das Datenpaket mittels einer direkten Speicherzugriffs-Funktion, DMA-Funktion, in den direkten Speicherbereich (3.10, 3.20) der der Header-Information entsprechenden virtuellen Maschine (P1, P2) zu kopieren, sodass es für die virtuelle Netzwerkschnittstelle (5.1, 5.2) dieser virtuellen Maschine (P1, P2) verfügbar ist, und dass

- die Filter- und Routingeinheit (2.3) dazu eingerichtet ist, einen Nachrichtensignalisierung-Interrupt dem Hypervisor (1) und/oder einem Interrupt-Controller zur Verfügung zu stellen, wobei die Filter- und Routingeinheit (2.3) in Abhängigkeit davon, welcher virtuellen Maschine (P1, P2) das Datenpaket zuzuweisen ist, einen anderen Nachrichtensignalisierung-Interrupt erzeugt, sodass in dem Nachrichtensignalisierung-Interrupt selbst kodiert ist, welche virtuelle Maschine zu benachrichtigen ist und die Nachrichtensignalisierung-Interrupts sich in einer Interrupt-Kennung oder in einer Kanalangabe oder Quellenangabe, aber nicht in der Angabe einer der virtuellen Maschine unterscheiden, sondern allein durch die Tatsache, dass pro virtueller Maschine ein anderer Interrupt-Typ verwendet wird, signalisiert ist, welche virtuelle Maschine zu benachrichtigen ist, und dass

- der Hypervisor (1) und/oder der Interrupt-Controller jeweils dazu eingerichtet ist,

- a) einen virtuellen Interrupt für die der Header-Information entsprechenden virtuellen Maschine (P1, P2) bereitzustellen, wobei der virtuelle Interrupt in Abhängigkeit davon erzeugt wird, welcher Nachrichtensignalisierung-Interrupt eintrifft, oder

- b) in einem jeweiligen Signalisierungsbereich (3.01) der der Header-Information entsprechenden virtuellen Maschine (P1, P2) zu signalisieren, dass neue RX-Daten vorliegen.

2. Fahrzeugelektronikeinheit (10) nach Anspruch 1, wobei die Filter- und Routingeinheit (2.3) für das Filtern einen Content Addressable Memory, CAM, insbesondere einen Ternary Content Addressable Memory, TCAM, aufweist, durch welchen mehreren möglichen Speicherinhalten des Headers jeweils eine der virtuellen Maschinen zugeordnet ist, sodass für den Fall, dass der Header einen der möglichen Speicherinhalte enthält, der CAM die zugeordnete virtuelle Maschine signalisiert.

3. Fahrzeugelektronikeinheit (10) nach einem der vorhergehenden Ansprüche, wobei die Filter- und Routingeinheit (2.3) auf der Grundlage einer Switching/Routing-Einheit der physikalischen Netzwerkschnittstelle (2.1) für ein Layer 2 und/oder Layer 3 und/oder Layer 4 Switching/Routing gebildet ist, wobei durch einen Netzwerkprozessor der physikalischen Netzwerkschnittstelle (2.1) eine Umkonfiguration der Switching/Routing-Einheit für das Erzeugen des Nachrichtensignalisierung-Interrupts bewirkt ist.

4. Fahrzeugelektronikeinheit (10) nach einem der vorhergehenden Ansprüche, wobei die physikalische Netzwerkschnittstelle (2.1) dazu eingerichtet ist, als den Nachrichtensignalisierung-Interrupt einen Message-Signaled-Interrupt, MSI, zu erzeugen.

5. Fahrzeugelektronikeinheit (10) nach einem der vorhergehenden Ansprüche, bei welcher

- ein erster Speicherabschnitt (3.1) einer ersten der virtuellen Maschinen (P1) in dem direkten Speicherbereich (3.10)

- einen RX-Puffer zum Speichern von empfangenen Daten, und

- einen TX-Puffer zum Speichern von zu sendenden Daten aufweist, und

- ein zweiter Speicherabschnitt (3.2) einer zweiten der virtuellen Maschinen (P2) in dem direkten Speicherbereich (3.20)

- einen RX-Puffer zum Speichern von empfangenen Daten, und

- einen TX-Puffer zum Speichern von zu sendenden Daten aufweist, wobei in der ersten und der zweiten virtuellen Maschine (P1, P2) deren jeweilige virtuelle Netzwerkschnittstelle (5.1, 5.2) als ein Protokollstapel ausgestaltet ist, dessen unterste Layer dazu eingerichtet ist, direkt auf den jeweiligen RX-Puffer und TX-Puffer der jeweiligen virtuellen Maschine (P1, P2) zuzugreifen.

6. Fahrzeugelektronikeinheit (10) nach einem der vorhergehenden Ansprüche, bei welcher jeweils ein Speicherabschnitt (3.1, 3.2) einen Speicherbereich (3.10, 3.20) für IP-Steuerdaten und einen Speicherbereich (3.11, 3.21) für Video- und/oder Grafikdaten aufweist,

7. Fahrzeugelektronikeinheit (10) nach einem der vorhergehenden Ansprüche, bei welcher wenigstens eine der virtuellen Maschinen (P1, P2) für die DMA-Funktion mit einer PCIe-Schnittstelle (6) zum Empfangen von in dem zugeordneten Speicherabschnitt (3.1, 3.2) zu speichernden Daten und Senden von Daten aus dem zugeordneten Speicherabschnitt (3.1, 3.2) konfiguriert ist.

8. Fahrzeugelektronikeinheit (10) nach einem der vorhergehenden Ansprüche, bei welcher

wenigstens zwei der virtuellen Maschinen (P1, P2)

- auf wenigstens zwei physikalischen Prozessoren ausgeführt sind, sodass jeder der beiden virtuellen Maschinen ein eigener physikalischer Prozessor zur Verfügung steht,
- den beiden Prozessoren jeweils wenigstens ein eigener physikalischer Speicher zugeordnet ist, in welchem der jeweilige der virtuellen Maschine zugeordnete Speicherabschnitt (3.1, 3.2) angeordnet ist, und
- die beiden physikalischen Speicher jeweils für die DMA-Funktion eine PCIe-Schnittstelle zum Empfangen von zu speichernden Daten aus der physikalischen Netzwerkschnittstelle (2.1) und zum Senden von Daten über die physikalische Netzwerkschnittstelle (2.1) aufweisen.

9. Datenkommunikationsverfahren zwischen den virtuellen Maschinen (P1, P2) der Fahrzeugelektronikeinheit (10) nach einem der vorhergehenden Ansprüche einerseits und der physikalischen Netzwerkschnittstelle (2.1) der Fahrzeugelektronikeinheit (10) andererseits, wobei zum Empfangen wenigstens einer ein Datenpaket mit einem Header aufweisenden Datennachricht entsprechend einem Netzwerkprotokoll folgende Verfahrensschritte durchgeführt werden:

- Speichern des Datenpaketes der Datennachricht in einem RX-Puffer (2.2) der physikalischen Netzwerkschnittstelle (2.1),
- Auswerten des Headers der Datennachricht mittels einer Filter- und Routingeinheit (2.3) hinsichtlich einer das Datenpaket zuzuweisenden Partition einer der virtuellen Maschinen (P1, P2),
- Kopieren des Datenpaketes mittels einer DMA-Funktion in den direkten Speicherbereich (3.10, 3.20) des Speicherabschnitts (3.1, 3.2) der der Header-Information entsprechenden virtuellen Maschine (P1, P2), und
- Erzeugen eines Nachrichtensignalisierung-Interrupts für einen Hypervisor (1), wobei in Abhängigkeit davon, welcher virtuellen Maschine (P1, P2) das Datenpaket zuzuweisen ist, ein anderer Nachrichtensignalisierung-Interrupt erzeugt wird, wobei in dem Nachrichtensignalisierung-Interrupt selbst kodiert ist, welche virtuelle Maschine zu benachrichtigen ist und die Nachrichtensignalisierung-Interrupts sich in einer Interrupt-Kennung oder in einer Kanalangabe oder Quellenangabe, aber nicht in der Angabe einer der virtuellen Maschine unterscheiden, sondern allein durch die Tatsache, dass pro virtueller Maschine ein anderer Interrupt-Typ verwendet wird, signalisiert ist, welche virtuelle Maschine zu benachrichtigen ist,
- Bereitstellen eines virtuellen Interrupts für die der Header-Information entsprechenden virtuellen Maschine (P1, P2), wobei der virtuelle Interrupt unmittelbar in Abhängigkeit davon erzeugt wird, welcher Nachrichtensignalisierung-Interrupt eintrifft, und

- Verarbeiten des Datenpaketes durch die der Header-Information entsprechende virtuelle Maschine (P1, P2) mittels deren virtueller Netzwerkschnittstelle, die hierzu direkt auf den direkten Speicherbereich (3.10, 3.20) zugreift.

10. Datenkommunikationsverfahren nach Anspruch 9, wobei die DMA-Funktion durch eine PCIe-Schnittstelle (7) bereitgestellt wird.

11. Datenkommunikationsverfahren nach Anspruch 9 oder 10, wobei zum Senden wenigstens einer ein Datenpaket mit einem Header aufweisenden Datennachricht entsprechend einem Netzwerkprotokoll folgende Verfahrensschritte durchgeführt werden:

- Erzeugen des an die physikalische Netzwerkschnittstelle (2.1) weiterzuleitenden Datenpaketes durch eine der virtuellen Maschinen (P1, P2) und Speichern des Datenpaketes in dem zugehörigen direkten Speicherbereich (3.10, 3.20) des der virtuellen Maschine (P1, P2) zugeordneten Speicherabschnitts (3.1, 3.2),
- Erzeugen einer die Bereitstellung des weiterzuleitenden Datenpaketes anzeigenden Nachricht für die DMA-Funktion,
- Kopieren des Datenpaketes in einen TX-Puffer (2.20) der physikalischen Netzwerkschnittstelle (2.1) mittels der DMA-Funktion,
- Ergänzen des Datenpaketes mit einer Header-Information mittels der physikalischen Netzwerkschnittstelle (2.1),
- Erstellen der Datennachricht aus dem Datenpaket mit Header entsprechend dem Netzwerkprotokoll mittels der physikalischen Netzwerkschnittstelle (2.1), und
- Senden der Datennachricht mittels der physikalischen Netzwerkschnittstelle (2.1).

12. Kraftfahrzeug mit einer Fahrzeugelektronikeinheit (10) nach einem der Ansprüche 1 bis 8.

Es folgen 4 Seiten Zeichnungen

Anhängende Zeichnungen

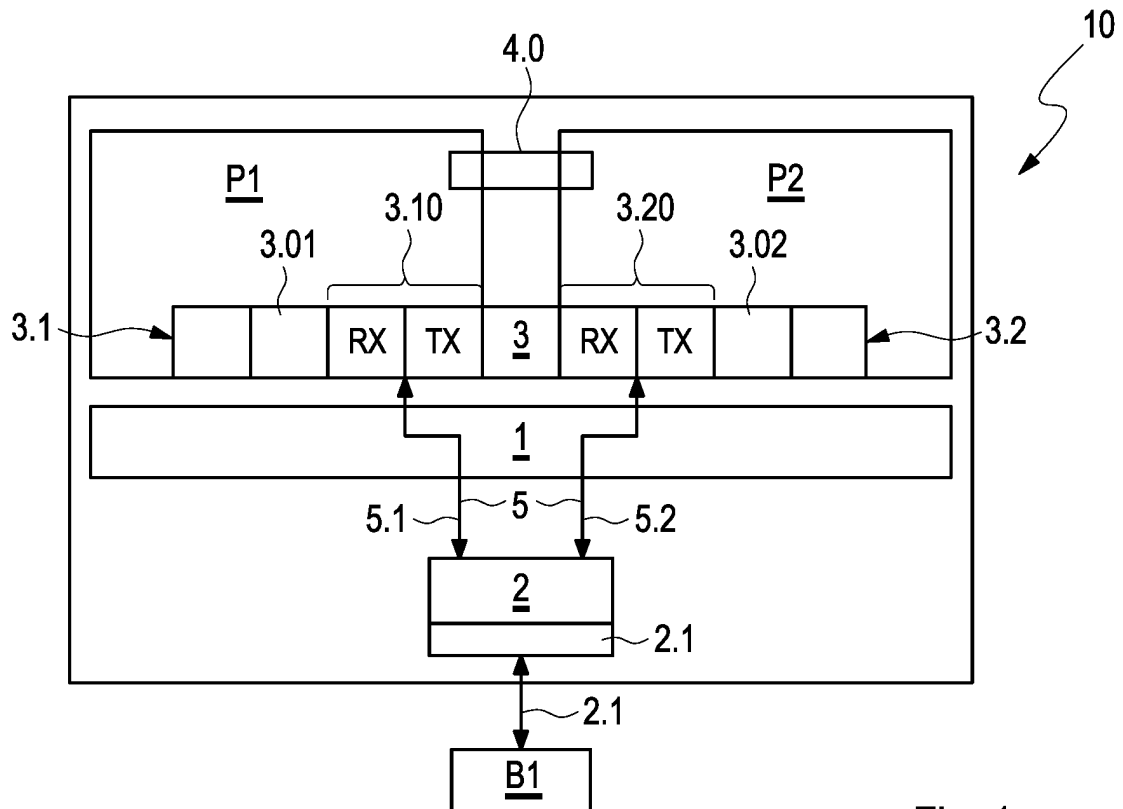


Fig. 1

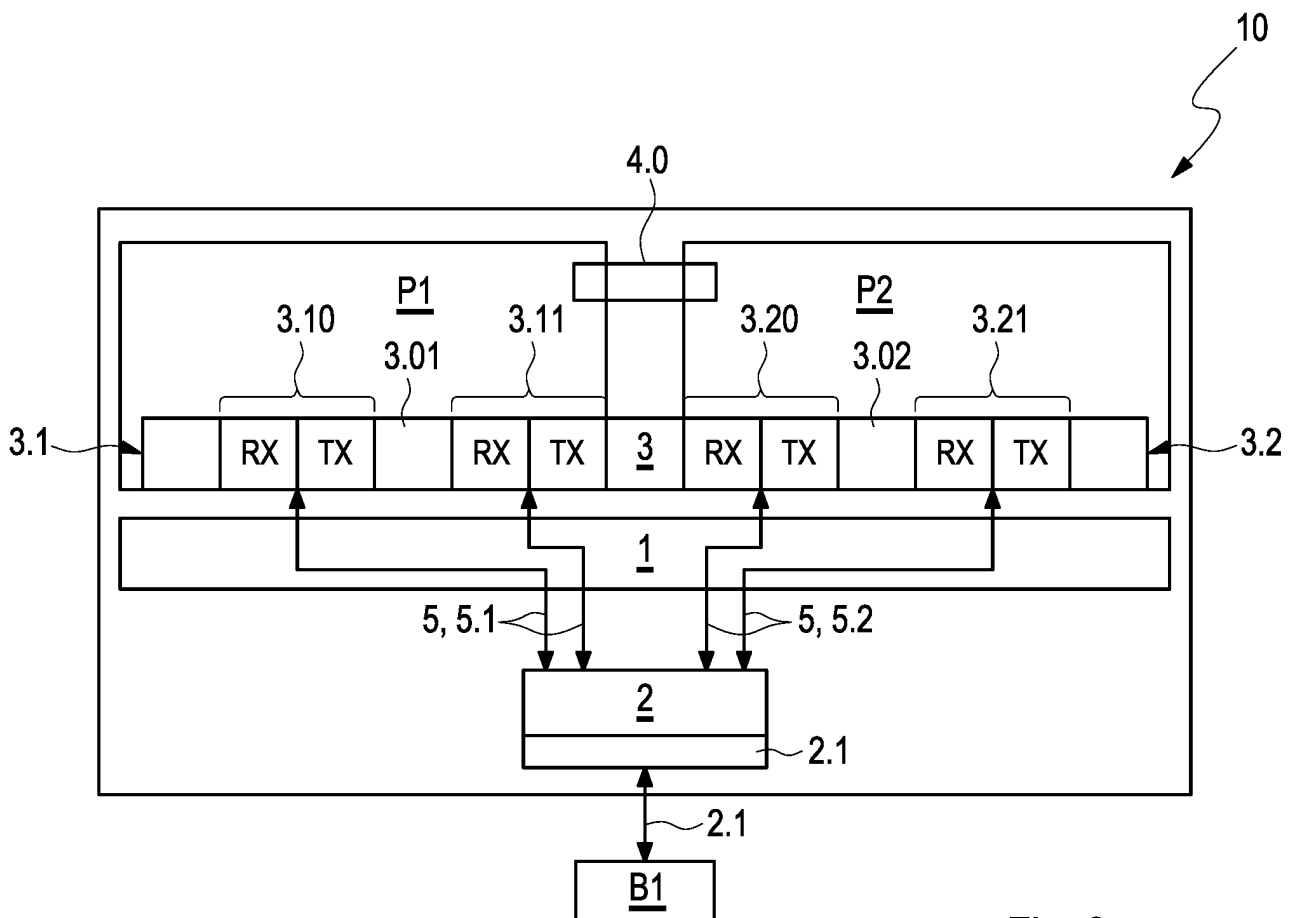
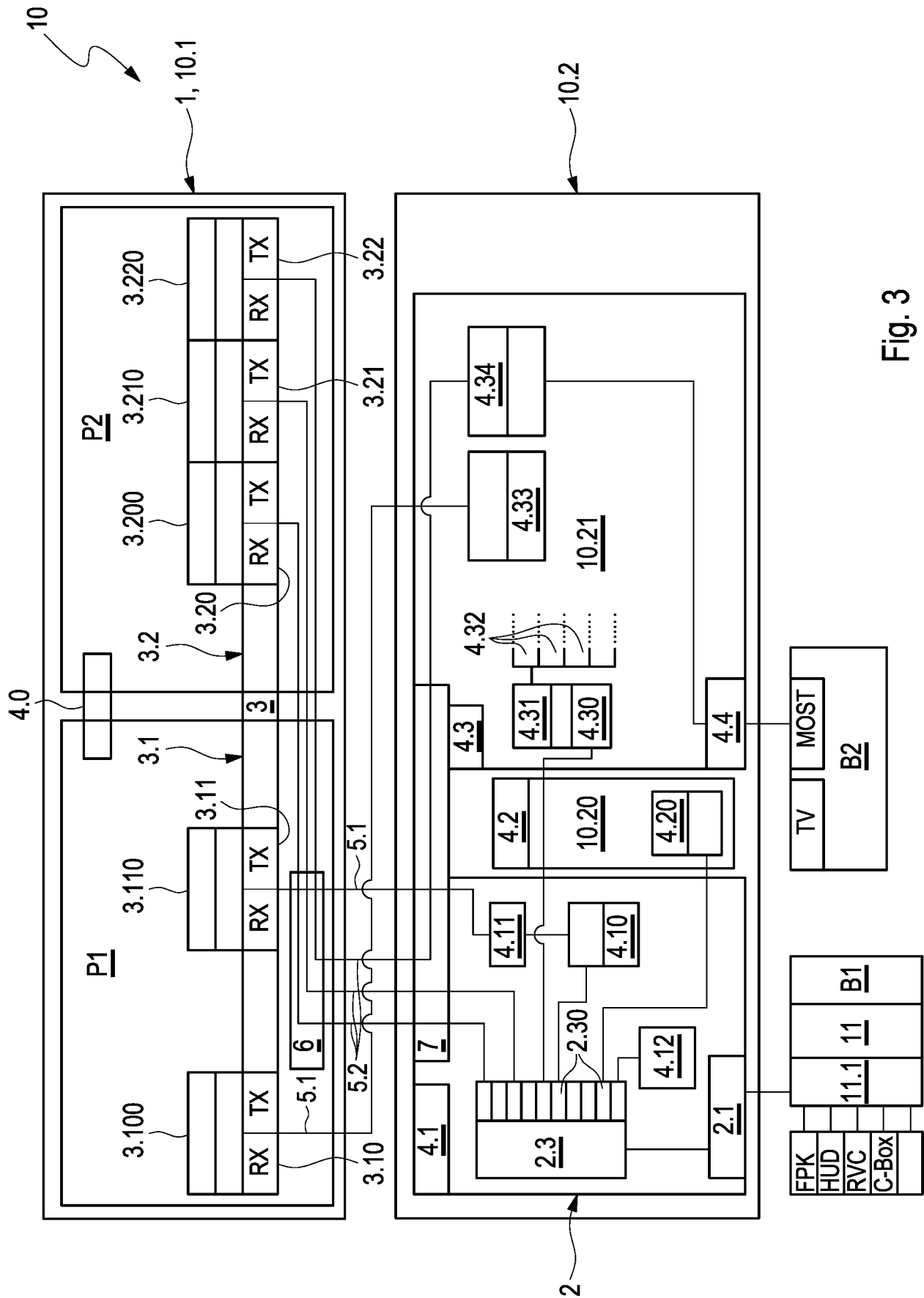


Fig. 2



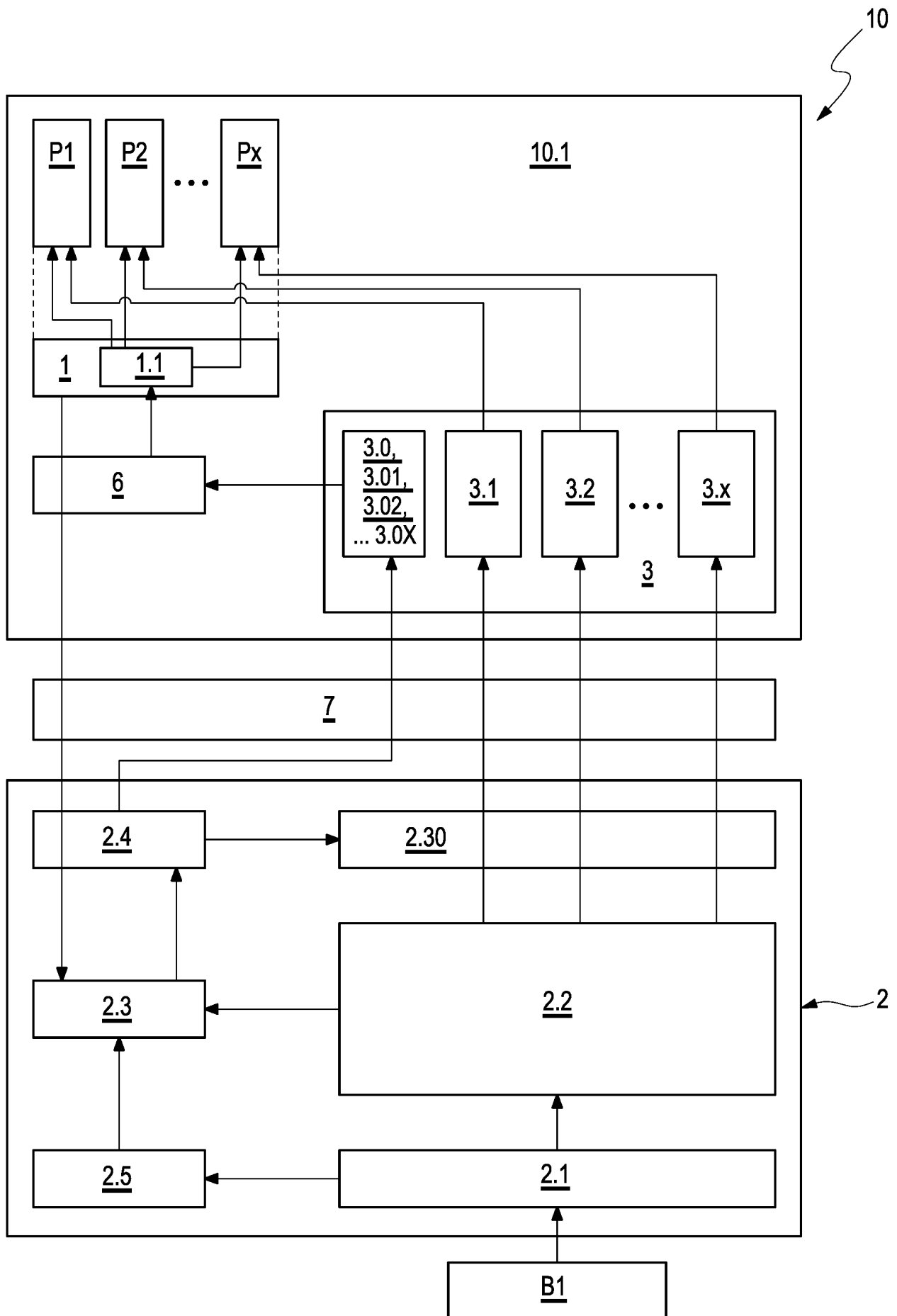


Fig. 4

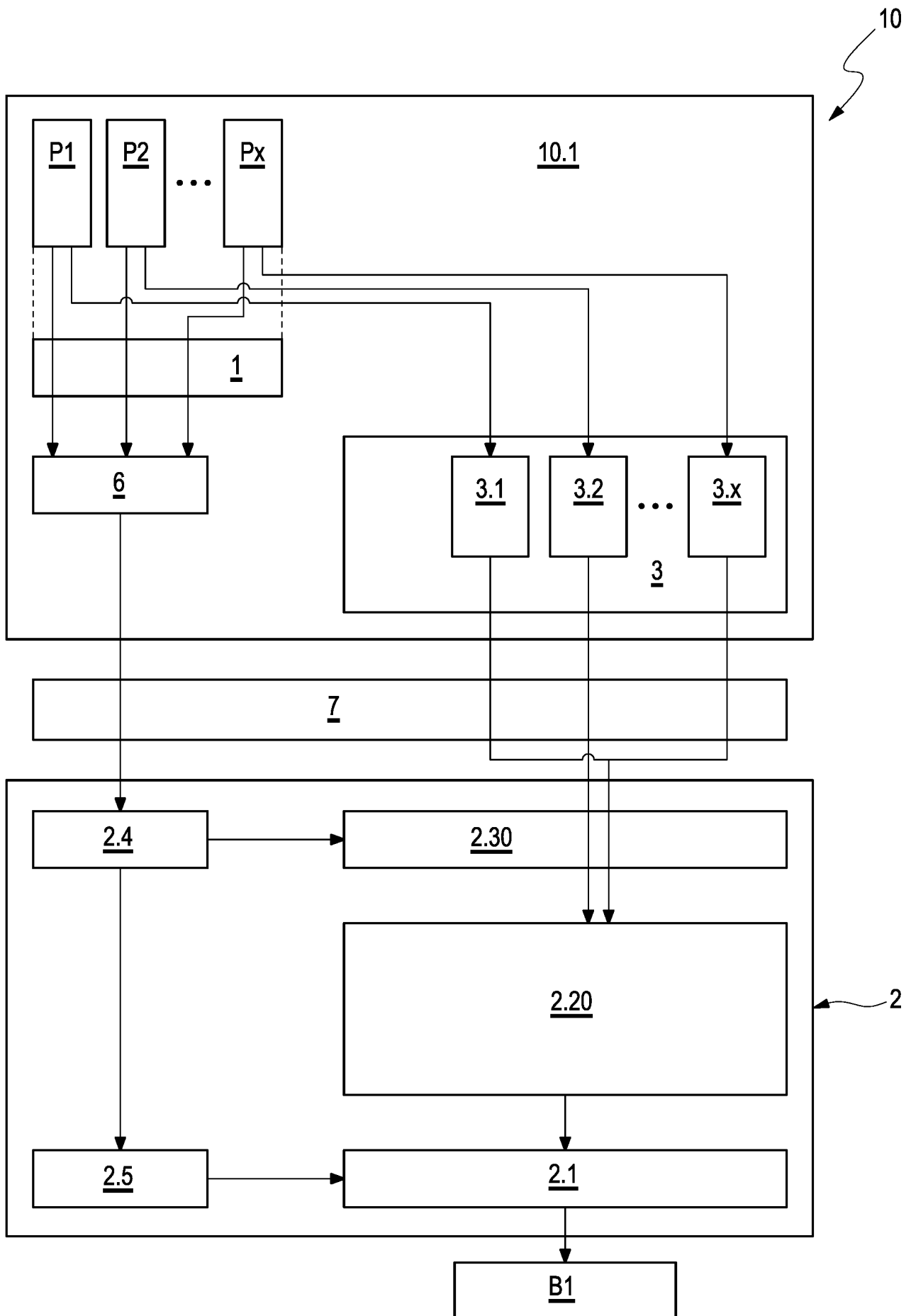


Fig. 5