



19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA

11 Número de publicación: **2 358 609**

51 Int. Cl.:  
**H04L 12/56** (2006.01)  
**G06F 13/28** (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Número de solicitud europea: **06806964 .0**  
96 Fecha de presentación : **04.10.2006**  
97 Número de publicación de la solicitud: **1941674**  
97 Fecha de publicación de la solicitud: **09.07.2008**

54 Título: **Participante y controlador de comunicaciones de un sistema de comunicaciones y método para la implementación de una funcionalidad de la puerta de enlace en un participante de un sistema de comunicaciones.**

30 Prioridad: **06.10.2005 DE 10 2005 048 585**

45 Fecha de publicación de la mención BOPI:  
**12.05.2011**

45 Fecha de la publicación del folleto de la patente:  
**12.05.2011**

73 Titular/es: **ROBERT BOSCH GmbH**  
**Postfach 30 02 20**  
**70442 Stuttgart, DE**

72 Inventor/es: **Ihle, Markus;**  
**Taube, Jan y**  
**Lorenz, Tobias**

74 Agente: **Carvajal y Urquijo, Isabel**

ES 2 358 609 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

## DESCRIPCIÓN

Participante y controlador de comunicaciones de un sistema de comunicaciones y método para la implementación de una funcionalidad de la puerta de enlace en un participante de un sistema de comunicaciones.

5 La presente invención hace referencia a un participante (el así llamado procesador central) de un sistema de comunicaciones. El participante presenta un microprocesador, al menos dos controladores de comunicaciones y un bus periférico. Mediante el bus periférico, el microprocesador se encuentra en contacto con los controladores de comunicaciones y, mediante los controladores de comunicaciones, respectivamente, se encuentra conectado a un enlace de comunicaciones del sistema de comunicaciones, mediante el cual son transmitidos mensajes.

10 Asimismo, la presente invención hace referencia también a un controlador de comunicaciones (el así llamado Communication Controller; CC) de un participante (el así llamado procesador central) de un sistema de comunicaciones. El participante presenta un microprocesador, los controladores de comunicaciones, de forma opcional otros controladores de comunicaciones y un bus periférico. Por un lado, el controlador de comunicaciones se encuentra en contacto con el microprocesador mediante el bus periférico y, por otro lado, se encuentra conectado a un enlace de comunicaciones del sistema de comunicaciones, mediante el cual son transmitidos mensajes.

15 Por último, la presente invención hace referencia también a un método para la implementación de una funcionalidad de una puerta de enlace dentro de un participante (llamado procesador central) de un sistema de comunicaciones. El participante presenta un microprocesador, al menos dos controladores de comunicaciones y un bus periférico. Mediante el bus periférico, los controladores de comunicaciones, por un lado, se encuentran en contacto con el microprocesador y, por otro lado, respectivamente, se encuentran conectados a un enlace de comunicaciones del sistema de comunicaciones, mediante el cual son transmitidos mensajes.

### Estado del arte

25 La conexión en red de dispositivos de control, sensores y actuadores con ayuda de un sistema de comunicaciones y de un enlace de comunicaciones conformado como un sistema ómnibus ha aumentado en forma drástica en los últimos años, tanto en la construcción de vehículos modernos como también en la construcción de máquinas, especialmente en el sector de las máquinas-herramienta, y en la automatización. A través de la distribución de funciones en varios dispositivos de control pueden obtenerse efectos sinérgicos. Se habla así de sistemas distribuidos. La comunicación entre diferentes estaciones tiene lugar, frecuentemente, cada vez más a través de un enlace de comunicaciones conformado como un sistema ómnibus, o mediante un enlace de comunicaciones. La circulación de la información en el sistema ómnibus, los mecanismos de entrada y de salida, así como también el control de errores, están regidos por un protocolo.

35 Dentro de esta área es conocido el protocolo FlexRay, donde la especificación v2.0 del protocolo FlexRay se constituye actualmente como una base. El protocolo Flexray es un sistema ómnibus rápido, determinista y tolerante a fallos, en especial para la utilización en un vehículo. La transmisión de datos, conforme al protocolo FlexRay, tiene lugar de acuerdo al método de acceso múltiple por división de tiempo (TDMA, por sus siglas en inglés), en el cual los componentes - las partes- de los mensajes - las partes- de los mensajes son adjudicados a ranuras de tiempo, donde éstas tienen una entrada exclusiva al enlace de comunicación.

40 La transmisión de datos mediante el enlace de comunicaciones tiene lugar en ciclos de transmisión que se repiten de forma regular, los cuales, respectivamente, se encuentran subdivididos en varias tramas de datos, conocidas también como ranuras de tiempo. Los participantes, así como los mensajes a ser transmitidos, son adjudicados a ranuras de tiempo determinadas, donde éstas tienen un acceso exclusivo al enlace de comunicaciones. Las ranuras de tiempo se repiten en los ciclos de transmisión determinados, de manera que puede predecirse con exactitud el instante en el que se transmite el mensaje mediante el bus, y el acceso al bus se produce determinísticamente.

45 Para aprovechar óptimamente el ancho de la banda para la transmisión del mensaje en el sistema ómnibus, FlexRay subdivide el ciclo de transmisión, también llamado ciclo del bus, en una parte dinámica y una parte estática. Las ranuras de tiempo determinadas se encuentran así en la parte estática al comienzo de un ciclo de bus. En la parte dinámica, las ranuras de tiempo se adjudican dinámicamente. De esta forma se posibilita entonces el exclusivo acceso al bus cada vez sólo por un corto tiempo, conocido como minislots (miniranuras de tiempo). Sólo cuando un acceso al bus tiene lugar dentro de una miniranura, se alarga la ranura durante el tiempo necesario. De esta manera, sólo se utiliza el ancho de banda cuando efectivamente se le necesita.

FlexRay comunica mediante dos líneas separadas físicamente del enlace de comunicaciones, con una velocidad máxima de 10 megabits (10 MBaud). De este modo, a los 5 ms, en algunos sistemas de comunicaciones incluso a 1

m o a 2,5 ms, es cerrado un ciclo del bus. Ambos canales corresponden a la capa física, en especial al modelo de capas del OSI (Open System Architecture). Estos canales sirven efectivamente para lograr una transmisión de mensajes redundante y tolerante a fallos, por lo cual se duplicaría entonces la velocidad. FlexRay puede, sin embargo, ser activado también con velocidades más bajas.

5 Para realizar funciones sincrónicas y optimizar el ancho de banda mediante pequeñas distancias entre los mensajes, los participantes, así como los componentes distribuidos en la red de comunicaciones, requieren de una base de tiempo común, conocida como tiempo global. Para lograr la sincronización de tiempos se transmiten mensajes de sincronización a la parte estática del ciclo, de manera tal que con la ayuda de un algoritmo especial correspondiente a la especificación FlexRay, el tiempo local de un participante es corregido, logrando así que todos los relojes marchen sincrónicamente en una secuencia de tiempo global.

10 Un participante FlexRay, el cual puede ser denominado también como nodo de red FlexRay o como procesador central, contiene un procesador del participante o un procesador central, un controlador FlexRay o controlador de comunicaciones, así como un control del bus contiene un controlador del bus. Con ello, el procesador del participante o el procesador central envía y procesa los datos transmitidos por el controlador de comunicaciones FlexRay. Para la comunicación en una red FlexRay pueden configurarse mensajes u objetos de mensaje, con por ejemplo, hasta 254 bytes de datos.

15 Para el acoplamiento de un enlace de comunicaciones FlexRay, mediante el cual son transmitidos mensajes a un participante FlexRay, de acuerdo a la solicitud DE 10 2005 034 744, la cual aún no había sido publicada el día de la solicitud de la presente invención, se emplea un módulo de comunicaciones FlexRay que mediante una interfaz del participante se encuentra conectado al participante y, mediante otra conexión, al enlace de comunicaciones. De este modo, para la transmisión de los mensajes entre el participante y el enlace de comunicaciones, se proporciona en el módulo de comunicaciones una disposición para el almacenamiento de los mensajes. La transmisión es controlada a través de una máquina de estado. El módulo de comunicaciones FlexRay puede estar conformado como un componente integral del controlador de comunicaciones FlexRay o como un componente separado.

20 En el módulo de comunicaciones se proporciona un módulo de la interfaz que se encuentra compuesto por dos partes, donde una de las partes del módulo es independiente del participante y la otra parte del módulo es específica del participante. La parte del módulo específica del participante o parte del módulo específica del cliente, denominada también como Customer CPU Interface (CIF), conecta un participante específico del cliente, en forma de un procesador CPU específico del participante, con el módulo de comunicaciones FlexRay. La parte del módulo independiente del participante, la cual es denominada también como Generis CPU Interface (GIF), representa una interfaz del CPU genérica, es decir común, mediante la cual, a través de las partes del módulo de comunicaciones específicas del cliente correspondientes, o sea Customer CPU Interfaces (CIFs), pueden ser conectados diferentes procesadores CPU específicos del cliente con el módulo de comunicaciones FlexRay. Debido a esto es posible una adaptación sin dificultades del módulo de comunicaciones a participantes diferentes, ya que en función del participante sólo debe variar la parte del módulo específica del participante, mientras que la parte del módulo de comunicaciones específica del cliente y el resto del módulo de comunicaciones siempre pueden estar conformados de la misma forma. Por tanto, mediante la ayuda del módulo de comunicaciones resulta una interfaz estándar para la conexión de los participantes FlexRay deseados a un enlace de comunicaciones FlexRay, donde, a través de una variación simple de la parte del módulo específica del participante, la interfaz puede ser adaptada de forma flexible a los participantes que se encuentran conformados o diseñados del modo deseado. De este modo, las partes del módulo pueden ser realizadas dentro del módulo de la interfaz, respectivamente en el software, es decir que cada parte del módulo puede ser realizada como una función del software.

25 La máquina de estado, en el módulo de comunicaciones FlexRay, puede encontrarse estructurada de forma fija en el hardware. De forma alternativa, la máquina de estado en el módulo de comunicaciones puede ser también programada libremente, mediante la interfaz, a través del participante.

30 De acuerdo al estado del arte, la memoria de mensajes del módulo de comunicaciones FlexRay, preferentemente, se encuentra ejecutada como una single-ported RAM (RAM de acceso sencillo, no simultáneo de lectura y escritura - Random Access Memory). Esta memoria RAM almacena los mensajes, así como los objetos de mensaje, es decir los datos útiles propiamente dichos, junto con los datos de estado y los datos de configuración. La estructura precisa de la memoria de mensajes del módulo de comunicaciones conocido puede ser observada en la solicitud DE 10 2005 034 744, mencionada anteriormente.

35 De acuerdo al estado del arte, dentro del participante, el microcontrolador, el cual comprende el microprocesador (así llamado procesador CPU), una memoria (por ejemplo Random Access Memory; RAM) y un bus core entre el microprocesador y la memoria, se encuentra conectado, como un así llamado maestro, al bus periférico mediante una interfaz pasiva y una activa. Mediante la interfaz pasiva, el microcontrolador sólo puede recibir órdenes y datos de otros participantes del bus periférico. Mediante la interfaz activa, el microcontrolador mismo puede enviar datos y órdenes a otros participantes del bus periférico. Una conexión de otro participante al bus periférico, mediante una

interfaz pasiva, equivale a una conexión del participante como un así llamado esclavo. Una conexión de un participante mediante una interfaz activa corresponde a una conexión del participante como un así llamado maestro. En el sistema de comunicaciones conocido, un controlador de comunicaciones, mediante una interfaz pasiva, como un así llamado esclavo, se encuentra conectado al bus periférico.

5 Los controladores de comunicaciones representan la conexión de las participantes hacia uno o más enlaces de comunicaciones (por ejemplo CAN, TTCAN, MOST, FlexRay, ByteFlight, etc.). Estos, respectivamente, poseen una memoria de mensajes donde son depositados los nuevos mensajes recibidos desde el enlace de comunicaciones y son leídos los mensajes a ser enviados en el enlace de comunicaciones. El microprocesador (Procesador CPU) puede acceder a los objetos de mensaje almacenados mediante la interfaz pasiva del controlador de comunicaciones.

10 El microprocesador configura, controla y dirige el controlador de comunicaciones. El microprocesador lee los mensajes recibidos, los evalúa, calcula nuevos mensajes y se ocupa de la escritura de los mensajes para el envío mediante el enlace de comunicaciones. Para la transmisión de datos dentro del participante, el microprocesador transmite los datos en forma de palabras desde el controlador de comunicaciones hacia la memoria del microcontrolador. De este modo, en la frecuencia de los ciclos actualmente usual del microprocesador se producen más ciclos de espera, durante los cuales el microprocesador espera la finalización de la transmisión de datos y no puede ocuparse de otras tareas.

15 En las operaciones sencillas de la puerta de enlace, frecuentemente sólo es necesario leer los datos recibidos desde un controlador de comunicaciones y escribir para el envío hacia uno o varios controladores de comunicaciones del participante. Si no se utiliza ningún controlador DMA (Direct Memory Access), entonces el microprocesador (el procesador CPU) transfiere los datos en forma de palabras desde los controladores de comunicaciones hacia un elemento de memoria asociado al microprocesador o hacia un elemento de memoria interno del CPU, para, dado el caso, procesarlos y, finalmente, copiarlos en los controladores de comunicaciones correspondientes. En la frecuencia de los ciclos actualmente usual del microprocesador (del procesador CPU), se producen varios ciclos de espera, durante los cuales el microprocesador se encuentra bloqueado y no puede realizar ninguna otra tarea.

20 El microprocesador configura, controla y dirige al menos un controlador de comunicaciones, así como la unidad lógica allí contenida y los datos de configuración de la interfaz activa. El microprocesador, ya en la memoria de mensajes del controlador de comunicaciones, lee los objetos de mensaje recibidos y copiados de forma automática en el elemento de memoria del microprocesador, los evalúa y procesa, calcula nuevos objetos de mensaje y los deposita en el elemento de memoria del microprocesador. Además, es transmitida información a los controladores de comunicaciones correspondientes, para que estos puedan transferir los datos actuales desde el elemento de memoria del microprocesador hacia su propia memoria de mensajes.

25 Del mismo modo, es conocido el hecho de, mediante una interfaz pasiva y una activa, conectar un controlador DMA al bus periférico, como un así llamado maestro, para descargar el microprocesador del microcontrolador de un participante del sistema de comunicaciones. El controlador DMA puede realizar la transferencia de datos entre el elemento de memoria del microcontrolador y el controlador de comunicaciones. Para ello, es configurado por el microprocesador (Procesador CPU) e iniciado. Seguidamente, el controlador DMA transmite datos en forma de palabras desde el controlador de comunicaciones en el elemento de memoria del microcontrolador o –siempre que esta función sea sustentada – directamente entre los controladores de comunicaciones. La finalización de la transmisión de datos es comunicada al microprocesador mediante una interrupción, de manera que el microprocesador inicia nuevamente el proceso para el siguiente mensaje. La operación de interrupción produce una gran cantidad de órdenes del CPU a través de las cuales se encuentra ligada una gran parte de los recursos de cálculo y de memoria del microprocesador. Asimismo, a través de las interrupciones frecuentes se incrementa la variabilidad temporal (indeterminación temporal con respecto a la duración de ejecución) de las tareas interrumpidas del software.

30 Por tanto, es objeto de la presente invención el optimizar la funcionalidad de la puerta de enlace dentro de un participante de un sistema de comunicaciones, en particular durante el enrutamiento de mensajes entre los controladores de comunicaciones de un participante deben ser reducidas la cantidad de ciclos de espera en el microprocesador del participante y la carga de interrupciones.

35 Para alcanzar este objeto, tomando como base el participante de la clase mencionada en la introducción, se sugiere que al menos uno de los controladores de comunicaciones presente una interfaz activa, mediante la cual el controlador de comunicaciones se encuentre en contacto con el bus periférico, y una unidad lógica para la implementación autónoma de una funcionalidad de la puerta de enlace.

Ventajas de la invención

De acuerdo a la invención, se sugiere conectar el controlador de comunicaciones al bus periférico no como un así llamado esclavo, sino como un maestro. De forma adicional, el controlador de comunicaciones obtiene una inteligencia en forma de una unidad lógica, la cual le posibilita coordinar de forma autónoma y controlar una funcionalidad de la puerta de enlace entre una memoria de mensajes del controlador de comunicaciones y una memoria del microcontrolador, así como entre memorias de mensajes de varios controladores de comunicaciones. A través de la unidad lógica puede ser proporcionada al controlador de comunicaciones una cierta funcionalidad DMA.

Por tanto, de acuerdo a la invención, la transferencia de datos entre el microcontrolador (Procesador CPU) del participante y una memoria de mensajes de un controlador de comunicaciones, así como entre memorias de mensajes de varios controladores de comunicaciones, es optimizada a través de la incorporación de una interfaz activa. De este modo, pueden ser sustentadas en particular funciones específicas de la puerta de enlace, como por ejemplo la transferencia de datos controlada mediante tiempo o controladas por sucesos. La transferencia de los mensajes a través del controlador de comunicaciones, de acuerdo a la configuración correspondiente, es desarrollada ampliamente de forma autónoma a través del microprocesador (procesador CPU) del participante. El método conforme a la invención reduce así como evita la hasta el momento elevada carga del microprocesador (procesador CPU) del participante a través de ciclos de espera e interrupciones, puesto que los mensajes pueden ser transferidos directamente mediante la interfaz activa.

La presente invención puede ser utilizada en cualquier participante de cualquier sistema de comunicaciones. A continuación, la invención es descrita a modo de ejemplo mediante un sistema de comunicaciones FlexRay. Esta referencia al protocolo FlexRay, sin embargo, no debe comprenderse como restrictiva. Naturalmente, la presente invención puede ser utilizada también en participantes de un sistema de comunicaciones MOST (Media Oriented Systems Transport), CAN (Controller Area Network), TTCAN (Time Triggered CAN), LIN (Local Interconnect Network) o de cualquier otro sistema de comunicaciones conocido.

De forma adicional con respecto a la interfaz pasiva, el controlador de comunicaciones conforme a la invención dispone de una interfaz activa mediante la cual una unidad lógica incorporada al controlador de comunicaciones, de forma autónoma, puede tanto leer los objetos de mensaje desde la memoria de mensajes del controlador de comunicaciones y escribirlos en una memoria de un destino correspondiente (elemento de memoria del microprocesador o memoria de mensajes de un controlador de comunicaciones), como también leer objetos de mensaje desde una fuente (elemento de memoria del microprocesador o memoria de mensajes de un controlador de comunicaciones) y escribirlos en las memorias de mensajes del controlador de comunicaciones conforme a la invención. Naturalmente, es posible pensar que se conformen no sólo uno, sino varios controladores de comunicaciones de un participante del modo acorde a la invención. Para la presente invención no se requiere que todos los módulos (controladores de comunicaciones) posean una interfaz activa. La interfaz activa se encuentra diseñada de modo tal que puede peticionar datos de otros módulos de forma autónoma (sin una activación a través de un microcontrolador de control o de otras unidades de control).

Una funcionalidad preferente de la puerta de enlace, la cual es implementada por el controlador de comunicaciones conforme a la invención, así como por su unidad lógica, es por ejemplo el enrutamiento (retransmisión) de mensajes que ingresan dentro de la red (la cual se compone del bus periférico, el microprocesador, los controladores de comunicaciones y, eventualmente, de otros módulos). En principio puede pensarse en diferentes métodos de enrutamiento, de los cuales algunos son explicados en detalle a continuación.

Enrutamiento estático:

Este método no es adaptable, es muy sencillo y, por tanto, es muy utilizado. Cada nodo (controlador de comunicaciones) mantiene una tabla con una hilera para cada nodo de destino posible. Una hilera contiene n inscripciones, con el mejor, el segundo mejor, etc., trayecto de transmisión para este destino, junto con una ponderación. Para la retransmisión de un paquete de datos, la inscripción correspondiente es seleccionada de la tabla y es transmitida a una de las líneas posibles. La ponderación refleja aquí la probabilidad de que esta línea sea seleccionada.

Enrutamiento centralizado:

El enrutamiento centralizado representa un método adaptable. En la red existe un centro de control de enrutamiento (RCC, por sus siglas en inglés), al cual cada nodo envía periódicamente informaciones de estado. Las informaciones de estado, a modo de ejemplo, se refieren a una lista de todos los nodos contiguos activos, a la longitud de la cola, a la extensión del tráfico desde la última comunicación, etc. El RCC reúne las informaciones de estado y, en base a este conocimiento, calcula las longitudes óptimas de las trayectos entre todos los nodos a lo largo de toda la red. A continuación, el RCC transmite a cada nodo una tabla de enrutamiento mediante la cual el nodo toma sus decisiones

de enrutamiento. La función del RCC puede ser realizada por el controlador de comunicaciones conforme a la invención, así como por su unidad lógica.

Ventajas:

- el RCC, teóricamente, posee una visión de conjunto completa y, de este modo, puede tomar decisiones "perfectas";
- 5 - los nodos no deben ejecutar cálculos costosos;
- alivio del microprocesador del participante, puesto que la función del RCC es realizada, al menos de forma parcial, por uno o varios controladores de comunicaciones, así como por sus unidades lógicas.

Desventajas:

- para redes de gran tamaño el cálculo puede eventualmente ser muy largo;
- 10 - la falla del RCC paraliza la red en su totalidad (siempre que no se encuentre presente o no haya sido definido un calculador de reserva);
- son posibles inconsistencias globales, ya que los nodos que se encuentran más próximos al RCC obtienen esencialmente antes las nuevas tablas de enrutamiento en comparación con los nodos que se encuentran más alejados;
- 15 - intensa carga del RCC a través de la función central.

Enrutamiento aislado:

En este método de enrutamiento cada nodo decide sólo en base a la informaciones que el mismo reúne, así como ha reunido. No tiene lugar ningún intercambio de informaciones de enrutamiento entre los nodos. La adecuación a las modificaciones del tráfico o a la topología de la red (por ejemplo a través de la suspensión o la incorporación de nodos), puede aquí sólo efectuarse en una extensión restringida, pero es posible, en principio, sin una gran inversión. Entre los métodos de enrutamiento aislados cuentan:

- el enrutamiento de multidifusión
- el enrutamiento de "patata caliente" (Hot Potato)
- el enrutamiento de aprendizaje hacia atrás (Backward Learning)
- 25 - el enrutamiento delta (Delta- Routing)

Enrutamiento de multidifusión

En el enrutamiento de multidifusión es enviado un paquete a todos los nodos. Se diferencian aquí dos variantes: se produce una vez un paquete separado para cada nodo y, por otra parte, el flujo, en el cual es transmitido el mismo paquete a cada nodo. El flujo es el método más sencillo y no es adaptable. Cada paquete que ingresa es transmitido en cada línea de transmisión, excepto en aquellas en las cuales se realiza. En este caso pueden también tomarse medidas para el encauzamiento del flujo, como:

- reconocimiento de duplicados de paquetes, a través de la numeración de los paquetes
- control de la vida útil de los paquetes a través del conteo de los trayectos parciales recorridos (saltos)
- flujo selectivo (retransmisión no a todas sino sólo a algunas líneas)
- 35 - camino aleatorio (Random Walk) (selección aleatoria de una línea)

Enrutamiento de "patata caliente" (Hot Potato)

Cada nodo intenta retransmitir los paquetes que ingresan tan pronto como sea posible. (Los nodos manipulan el paquete como una patata caliente, de allí proviene la denominación). De este modo, la línea de transmisión es

seleccionada con la cola más breve. Existen también combinaciones de este método con el método del enrutamiento estático:

- selección de la mejor línea de transmisión de acuerdo al método estático, mientras que su longitud de la cola permanezca por debajo de un umbral determinado.

5 - selección de la línea de transmisión con la cola más breve, en caso de que su carga sea demasiado reducida.

(Véase el enrutamiento estático, explicado anteriormente)

#### Enrutamiento de aprendizaje hacia atrás (Backward Learning)

En este método debe ser almacenada en el paquete la siguiente información:

- identificación del nodo fuente;

10 - contador que aumenta en uno con cada trayecto parcial (salto) recorrido. Cuando un nodo sólo obtiene un paquete, puede reconocer la cantidad de saltos y saber mediante qué entrada lo ha recibido. De este modo, cada nodo puede inferir a partir del paquete obtenido mediante qué camino éste puede alcanzar los otros nodos con la cantidad mínima de saltos. Es reemplazada una inscripción en la tabla de enrutamiento cuando un paquete alcanza los nodos con una cantidad menor de saltos que la que se encuentra registrada en la tabla. Sin embargo, las inscripciones son actualizadas también cuando a lo largo de un cierto tiempo no fue obtenido ningún paquete más con una cantidad de saltos determinada desde el respectivo nodo. Por tanto, se admiten períodos de aprendizaje a intervalos fijos, en los cuales son sobreescritas inscripciones mejores con inscripciones peores, cuando éstas son viejas durante un cierto tiempo. De este modo, se parte de la consideración de que ya no existe más el mejor enlace y es seleccionado el siguiente considerado mejor. De ello resultan los siguientes problemas:

20 - durante el período de aprendizaje el enrutamiento no es óptimo;

- en el caso de períodos de aprendizaje breves (las inscripciones son actualizadas más rápido y de forma deficiente), muchos paquetes toman caminos de una calidad no identificada;

- en el caso de períodos de aprendizaje largos se produce un mal comportamiento en cuanto a la adaptación a la situación en la red.

#### 25 Enrutamiento delta

Este método representa una combinación entre el enrutamiento centralizado y el enrutamiento aislado. En este caso, cada nodo asume los costos de forma periódica (por ejemplo, en el sentido de retraso, descarga, capacidad, etc.) de cada trayecto de transmisión y envía esta información al RCC. El RCC calcula entonces los  $k$  mejores trayectos de nodos  $i$  hacia nodos  $j$  (a para todos los nodos  $i, j$ ), donde sólo son considerados trayectos que se diferencian en su línea inicial. El RCC envía a cada nodo la lista de todos los trayectos equivalentes para todos los puntos de destino. Para el enrutamiento actual, un nodo puede seleccionar de forma aleatoria un trayecto equivalente o decidir en base a los costes medidos actualmente. El término delta que otorga la denominación proviene aquí de la función mediante la cual se transmite si dos trayectos pueden considerarse como equivalentes.

30

#### Enrutamiento adaptativo distribuido

35 En este método cada nodo, de forma periódica, intercambia informaciones de enrutamiento con cada uno de los nodos vecinos. También en este caso cada nodo obtiene una tabla de enrutamiento que contiene una inscripción en la red para cada uno de los otros nodos. En esta tabla se encuentra la línea de transmisión preferida para estos nodos, así como una estimación respectiva al tiempo o a la distancia en relación a estos nodos:

- cantidad de saltos;

40 - retraso estimado en milisegundos;

- cantidad total estimada de paquetes que esperan a lo largo del trayecto.

Estas estimaciones se obtienen a partir del tiempo/distancia con respecto a los nodos vecinos (por ejemplo mediante paquetes ECHO con una marca de tiempo) y/o estimaciones de los nodos vecinos. Un intercambio de información

de enrutamiento puede tener lugar de forma sincrónica en intervalos de actualización determinados o de forma asincrónica en el caso de variaciones significativas. A estos métodos, entre otros, pertenecen:

- el enrutamiento por vector de distancia;
- el enrutamiento por estado de enlace.

5 Enrutamiento por vector de distancia

10 Éste consiste en un enrutamiento distribuido, adaptable, el cual fue utilizado anteriormente en Internet como un RIP (Routing Information Protocol). En este caso, cada enrutador almacena una tabla con la mejor distancia (por ejemplo cantidad de saltos, retraso, etc.) con respecto a cada destino y a la salida correspondiente. En la práctica, este método presenta una convergencia demasiado lenta en relación a un estado consistente para muchos enrutadores, debido a la problemática del "Count-to-infinity".

Enrutamiento por estado de enlace

Éste consiste en un enrutamiento adaptable, distribuido, el cual es utilizado en Internet como un OSPF (Open Shortest Path First) y un IS-IS (Intermediate System to intermediate System). En estos casos se encuentra una aplicación para el siguiente algoritmo:

- 15 - descubrimiento de nuevos nodos vecinos mediante un así llamado paquete HELLO;
- medición del retraso, así como de los costes con respecto a cada nodo vecino, mediante un así llamado paquete ECHO;
- 20 - elaboración de un así llamado paquete de LINK-STATE (estado de enlace) con todos los datos aprendidos (emisor, lista de los nodos vecinos con retraso, antigüedad, etc.) el cual es generado periódicamente o controlado de acuerdo a los sucesos (por ejemplo nuevos nodos, suspensión de un nodo, etc.);
- envío de este paquete a todos los nodos vecinos (principalmente mediante flujos, pero con refinamiento: eliminación de las duplicaciones, destrucción de la información de acuerdo a cierta antigüedad, etc.);
- cálculo de la ruta más corta con respecto a todos los otros enrutadores (por ejemplo conforme al algoritmo de Dijkstra - algoritmo de caminos mínimos).
- 25 Este método implica una gran cantidad de cálculos, pero existen optimizaciones de este método, las cuales pertenecen por tanto a la respectiva topología de la red.

Enrutamiento jerárquico

30 El principio del enrutamiento jerárquico es el fraccionamiento de redes grandes en regiones. Los nodos de una región poseen sólo información de enrutamiento sobre sus propias regiones. En cada región existe al menos un nodo que se distingue, el cual sirve como interfaz con respecto a las otras regiones. En el caso de redes muy grandes son posibles otras jerarquías debido al tamaño creciente de las redes (regiones, cluster, zonas, grupos,...).

35 Conforme a un perfeccionamiento ventajoso de la presente invención, se sugiere que la unidad lógica implemente la funcionalidad de un motor de enrutamiento. La función de un motor de enrutamiento de un controlador de comunicaciones consiste, principalmente, en retransmitir mensajes que ingresan a la red (comprendiendo el bus periférico, el microprocesador, el controlador de comunicaciones y, eventualmente, otros módulos). La funcionalidad a ser implementada de un motor de enrutamiento hace referencia en particular al control y a la coordinación de los métodos de enrutamiento explicados anteriormente.

40 De acuerdo a una forma de ejecución preferente de la invención, se sugiere que los controladores de comunicaciones, respectivamente, presenten una memoria de mensajes para el almacenamiento temporal de mensajes desde el enlace de comunicaciones asociado al controlador de comunicaciones o para el enlace de comunicaciones, donde la unidad lógica coordina y controla un enrutamiento autónomo de datos entre la memoria de mensajes de un controlador de comunicaciones y el microprocesador (procesador CPU) o entre las memorias de mensajes de los controladores de comunicaciones.

45 Conforme a un perfeccionamiento especialmente ventajoso de la presente invención, se sugiere que al menos uno de los controladores de comunicaciones presente un módulo de comunicaciones, el cual comprende una memoria

de mensajes para el almacenamiento temporal de mensajes desde el enlace de comunicaciones asociado al controlador de comunicaciones o para el enlace de comunicaciones, donde la unidad lógica coordina y controla un enrutamiento autónomo de datos entre la memoria de mensajes de un controlador de comunicaciones y/o la memoria de mensajes de al menos un módulo de comunicaciones.

- 5 De acuerdo a una forma de ejecución preferente de la invención, se sugiere que cada módulo de comunicaciones presente una memoria búfer dispuesta entre la memoria de mensajes y el bus periférico, preferentemente al menos una memoria búfer de entrada y al menos una memoria búfer de salida, donde la unidad lógica coordina y controla un enrutamiento autónomo de datos entre la memoria de mensajes de un controlador de comunicaciones o entre al menos una memoria búfer de la memoria de mensajes de al menos un módulo de comunicaciones y el
- 10 microprocesador o entre las memorias de mensajes del controlador de comunicaciones y/o al menos una memoria búfer de la memoria de mensajes de al menos un módulo de comunicaciones.

La unidad lógica en el controlador de comunicaciones comprende, de forma conveniente, una máquina de estado, preferentemente una máquina de estado DMA. De forma preferente, la máquina de estado se encuentra implementada de acuerdo al hardware y se encuentra estructurada de forma fija en el controlador de

15 comunicaciones.

De acuerdo a otra forma de ejecución de la invención, se sugiere que cada controlador de comunicaciones, el cual posee una interfaz activa y una unidad lógica, presente un árbitro, el cual arbitre las operaciones de acceso concurrentes del microprocesador y de la unidad lógica (por ejemplo de la máquina de estado) en una memoria de mensajes del controlador de comunicaciones o en una memoria de mensajes de un módulo de comunicaciones o en

20 al menos una memoria búfer de una memoria de mensajes de un módulo de comunicaciones.

De forma ventajosa, cada controlador de comunicaciones, el cual posee una interfaz activa y una unidad lógica, presenta un registro de control y/o de estado, al cual tiene acceso el microprocesador para la configuración, la dirección y/o el monitoreo de la funcionalidad de la puerta de enlace.

Asimismo, se sugiere que cada controlador de comunicaciones, el cual posee una interfaz activa y una unidad

25 lógica, presente una memoria de enrutamiento, a la cual tenga acceso el microprocesador para la configuración, la dirección y/o el monitoreo de la funcionalidad de la puerta de enlace.

De forma preferente, el participante forma parte de un sistema de comunicaciones FlexRay, en el cual tiene lugar una transmisión de datos entre el participante y otros participantes FlexRay que se encuentran conectados al enlace de comunicaciones FlexRay, en base al protocolo FlexRay.

30 Como otra solución con respecto al objeto de la presente invención, tomando como base el controlador de comunicaciones de la clase mencionada en la introducción, se sugiere que el controlador de comunicaciones presente una interfaz activa, mediante la cual el controlador de comunicaciones se encuentre en contacto con el bus periférico, y una unidad lógica para la implementación autónoma de una funcionalidad de la puerta de enlace.

Como otra solución con respecto al objeto de la presente invención, tomando como base el método de la clase mencionada en la introducción, se sugiere que al menos uno de los controladores de comunicaciones se encuentre conectado al bus periférico mediante una interfaz activa y que la funcionalidad de la puerta de enlace sea implementada a través de una unidad lógica en al menos un controlador de comunicaciones.

35

De acuerdo a un perfeccionamiento ventajoso de la presente invención, se sugiere que en el marco de la funcionalidad de la puerta de enlace sean iniciadas y procesadas interrupciones, sean fragmentados y defragmentados mensajes y sean intercambiados mensajes entre controladores de comunicaciones del participante y/o son intercambiados mensajes entre uno de los controladores de comunicaciones y la memoria de mensajes del microprocesador.

40

Asimismo, se sugiere que el microprocesador almacene información de control y de estado en un registro de control y/o de estado de al menos un controlador de comunicaciones y, de este modo, la funcionalidad de la puerta de enlace sea configurada, controlada y/o monitoreada. A su vez, se sugiere que el microprocesador almacene información de enrutamiento en una memoria de enrutamiento de al menos un controlador de comunicaciones y, de este modo, la funcionalidad de la puerta de enlace sea controlada y/o monitoreada.

45

De acuerdo a una forma de ejecución preferente de la invención, se sugiere que la unidad lógica, en al menos un controlador de comunicaciones, comprenda una máquina de estado, donde sean arbitradas las operaciones de acceso concurrentes del microprocesador y de la máquina de estado en una memoria de mensajes del controlador de comunicaciones o en una memoria de mensajes de un módulo de comunicaciones o en al menos una memoria búfer de una memoria de mensajes de un módulo de comunicaciones.

50

Para el copiado activo de datos en una memoria de mensajes de otro controlador de comunicaciones se sugiere que los siguientes pasos sean ordenados y controlados de forma autónoma por la unidad lógica de al menos un controlador de comunicaciones:

- 5 - representación de forma visible de los datos recibidos en una memoria búfer de al menos un controlador de comunicaciones;
- creación de una copia de los datos recibidos en un rango de direcciones que puede ser configurado, de un elemento de memoria asociado al microprocesador (procesador CPU) o en una memoria búfer de otro controlador de comunicaciones, a través de operaciones de acceso de escritura mediante la interfaz activa; y
- 10 - cuando una copia de datos es creada en la memoria búfer del otro controlador de comunicaciones, inicio de una transferencia de los datos almacenados desde la memoria búfer hacia una memoria de mensajes del otro controlador de comunicaciones.

Para el copiado activo de datos desde una memoria de mensajes de otro controlador de comunicaciones se sugiere que los siguientes pasos sean ordenados y controlados de forma autónoma por la unidad lógica de al menos un controlador de comunicaciones:

- 15 - representación de forma visible de los datos en una memoria búfer del otro controlador de comunicaciones;
- creación de una copia del contenido de un elemento de memoria asociado al microprocesador o de la memoria búfer del otro controlador de comunicaciones en un rango de direcciones que puede ser configurado, de al menos una memoria búfer del controlador de comunicaciones, a través de operaciones de acceso de lectura mediante la interfaz activa; e
- 20 - inicio de una transferencia de los datos almacenados desde al menos una memoria búfer hacia una memoria de mensajes del módulo de comunicaciones del controlador de comunicaciones.

- 25 Por último, se sugiere que el método para el copiado activo de datos en una y/o desde una memoria de mensajes de otro controlador de comunicaciones sea iniciado controlado mediante tiempo, a través de la recepción de datos, en particular de un mensaje, a través de activación mediante al menos un controlador de comunicaciones o a través de una activación mediante el microprocesador. La activación, por lo general, comprende la transmisión de una orden especial de inicio hacia la unidad lógica.

### Dibujos

A continuación, mediante las figuras, se explican en detalle otras características y ventajas de la presente invención. Los dibujos muestran:

- 30 **Figura 1**: un módulo de comunicaciones y su conexión a un enlace de comunicaciones y un participante de comunicaciones o procesador de un sistema de comunicaciones Flex-Ray en una representación esquemática;

**Figura 2**: una forma de ejecución especial del módulo de comunicaciones de la figura1, así como su conexión en detalle;

**Figura 3**: la estructura de una memoria de mensajes del módulo de comunicaciones de la figura 2;

- 35 **Figuras 4 a 6**: la arquitectura y el proceso de un acceso a datos en la dirección del participante hacia la memoria de mensajes, en una representación esquemática;

**Figuras 7 a 9**: la arquitectura y el proceso de un acceso a datos en la dirección de la memoria de mensajes hacia el participante;

- 40 **Figura 10**: la estructura de un gestor de mensajes y de las máquinas de estado allí contenidas, en una representación esquemática;

**Figura 11**: componentes del módulo de comunicaciones de las figuras 1 y 2, así como los participantes y las rutas de datos correspondientes controladas por el gestor de mensajes, en una representación esquemática;

**Figura 12**: la distribución del acceso a la memoria de mensajes en relación a las rutas de datos de la figura 11;

**Figura 13:** un participante conforme a la invención de un sistema de comunicaciones;

**Figura 14:** posibles trayectos de transmisión para datos enrutados en un participante conforme a la invención;

**Figura 15:** un controlador de comunicaciones conforme a la invención del participante de la figura 13 en detalle;

**Figura 16:** una interfaz del participante del controlador de comunicaciones conforme a la invención de la figura 15; y

5 **Figura 17:** un participante de un sistema de comunicaciones, conocido por el estado del arte.

#### Descripción de los ejemplos de ejecución

La figura 1, de forma esquemática, muestra un módulo de comunicaciones FlexRay 100 para la conexión de un participante o procesador central 102 a un enlace de comunicaciones FlexRay 100, es decir a la capa física de FlexRay. Éste, por ejemplo, se encuentra conformado como un bus de datos FlexRay, el cual, de forma preferente, dispone de dos líneas de transmisión. Para ello, el módulo de comunicaciones FlexRay 100, mediante una conexión 107, se encuentra conectado al participante, así como al procesador del participante 102 y, mediante una conexión 106, se encuentra conectado al enlace de comunicaciones 101. Para una conexión sin dificultades, por un lado en relación a los tiempos de transmisión y, por otro lado, a la integridad de los datos, se diferencian esencialmente tres disposiciones en el módulo de comunicaciones FlexRay. De este modo, una primera disposición 105 sirve para el almacenamiento, en particular para el portapapeles, de al menos una parte de los mensajes a ser transmitidos. Entre el participante 102 y esta primera disposición 105 se encuentra conectada una segunda disposición 104 mediante las conexiones 107 y 108. Del mismo modo, entre el enlace de comunicaciones 101 y la primera disposición 105, mediante las conexiones 106 y 109, se encuentra conectada una tercera disposición, debido a lo cual puede obtenerse una entrada y una salida de datos muy flexible, como parte de mensajes, en particular mensajes FlexRay en, así como desde la primera disposición 105, asegurando la integridad de los datos a una velocidad óptima.

En la figura 2, este módulo de comunicaciones 100, en una forma de ejecución preferente, se encuentra representado nuevamente, más en detalle. Las respectivas conexiones 106 a 109 se encuentran representadas también de forma detallada. Para la conexión del enlace de comunicaciones FlexRay 100 al participante FlexRay 102, así como al procesador central, la segunda disposición 104 contiene una memoria búfer de entrada o una memoria búfer de ingreso 201 (Input Buffer IBF), una memoria búfer de salida o de devolución 202 (Output Buffer OBF), así como un módulo de la interfaz compuesto por dos partes 203 y 204, donde una parte del módulo 203 es independiente del participante y la segunda parte del módulo 204 es específica del participante. La parte del módulo 204 específica del participante (Customer CPU Interface CIF) conecta un procesador CPU 102, es decir un participante específico del cliente 102, al módulo de comunicaciones FlexRay 100. Para ello se proporciona una línea de datos 216 bidireccional, una línea de dirección 217, así como una entrada de control 218. Asimismo, se proporciona una salida de interrupción o de corte 219. La parte del módulo 204 específica del participante se encuentra en contacto con una parte del módulo 203 independiente del participante (Generic CPU Interface, GIF), es decir que el módulo de comunicaciones FlexRay o el módulo IP FlexRay dispone de una interfaz del CPU 203 genérica común a la cual puede conectarse una gran cantidad de diferentes CPUs procesadores centrales 102 específicos del cliente mediante partes del módulo específicas del participante correspondientes. De este modo, en función del participante 102 sólo debe variar la parte del módulo 204, lo cual significa que se realiza una inversión considerablemente reducida. La interfaz del CPU 203 y el resto del módulo de comunicaciones pueden ser aceptados sin modificaciones.

La memoria búfer de entrada o memoria búfer de ingreso 201 y la memoria búfer de salida o memoria búfer de devolución 202 pueden estar conformadas en un módulo de memoria común o, de lo contrario, en módulos de memorias separados. De esta manera, la memoria búfer de entrada 201 sirve para el almacenamiento temporal de mensajes para la transmisión hacia una memoria de mensajes 300. Con ello, el módulo búfer de entrada 201, de forma preferente, se encuentra conformado de modo tal que puede almacenar dos mensajes completos, los cuales se componen, respectivamente, de un segmento de cabecera o segmento Header, en especial con datos de configuración, y un segmento de datos o segmento Payload (segmento de carga útil de datos de información y no de datos de control). Para ello, la memoria búfer de entrada 201 consta de dos partes (sub memoria búfer y memoria espejo), a través de las cuales, mediante una escritura alternada de ambas partes de la memoria búfer de entrada, así como a través del cambio de acceso, se puede acelerar la transmisión entre el procesador CPU del participante 102 y la memoria de mensajes 300. Del mismo modo, la memoria búfer de salida o memoria búfer de devolución (Output Búfer OBF) sirve para el almacenamiento temporal de mensajes en la transmisión de la memoria de mensajes 300 al CPU del participante 102. Para ello, la memoria búfer de salida 202 se encuentra a su vez conformada de manera tal, que pueden ser almacenados dos mensajes completos compuestos por un segmento de cabecera, en especial con datos de configuración y por un segmento de datos o segmento Payload. También aquí la memoria búfer de salida 202 se encuentra dividida en dos partes, en una sub memoria búfer y en una memoria espejo, a través de las cuales, mediante una escritura alternada de ambas partes, así como a través del cambio de acceso, se puede acelerar la transmisión entre el procesador CPU del participante 102 y la memoria de mensajes

300. Esta segunda disposición 104, compuesta por los bloques 201 a 204, se encuentra, tal como se representa, conectada a la primera disposición 105.

La primera disposición 105 está compuesta por un gestor de mensajes 200 (Message Handler MHD) y una memoria de mensajes 300 (Message RAM). El gestor de mensajes controla, por tanto dirige la transferencia de datos entre la memoria búfer de entrada 201, así como también entre la memoria búfer de salida 202 y la memoria de mensajes 300. En igual medida controla, por tanto dirige la transferencia de datos en dirección inversa a través de la disposición 103. La memoria se encuentra, preferentemente, constituida como single-ported RAM (RAM de acceso sencillo, no simultáneo de lectura y escritura). Esta memoria RAM almacena los mensajes u objetos del mensaje, o sea los mensajes propiamente dichos, conjuntamente con datos de configuración y datos de estado. La estructura exacta de la memoria 300 es representada con detalle en la figura 3.

La tercera disposición 103 está compuesta por los bloques 205 a 208. En correspondencia con los dos canales de la capa física de FlexRay, esta disposición 103 se encuentra dividida en dos rutas de datos con dos direcciones de datos en cada una. Esto se pone de manifiesto claramente mediante las conexiones 213 y 214, en las cuales se presentan ambas direcciones para el canal A, RxA y TxA para recepción (RxA) y transmisión (TxA), así como para el canal B, RxB y TxB. La conexión 215 se refiere a una óptima entrada de control bidireccional. El enlace de la tercera disposición búfer 103 se produce mediante una primera memoria búfer 205 para el canal B y una segunda memoria búfer 206 para el canal A. Ambas memorias búfer (búfer no residente RAM's: RAM A y RAM B) sirven como memorias intermedias para la transmisión de datos desde y hacia la primera disposición 105. En correspondencia con los dos canales, estas dos memorias búfer 205 y 206 se encuentran conectadas respectivamente a interfaces del módulo 207 y 208, las cuales contienen el controlador del protocolo FlexRay o controlador protocolo bus, el cual consta de un registro de desplazamiento de recepción y transmisión y del protocolo FlexRay para máquinas de estado finito. Ambas memorias búfer 205 y 206 sirven como memorias intermedias para la transmisión de datos entre los registros de desplazamiento de la interfaz del módulo o del controlador del protocolo FlexRay 207 y 208 y la memoria de mensajes 300. También aquí se almacenan de manera ventajosa, a través de cada memoria búfer 205 ó 206, los campos de datos, por tanto el segmento Payload o el segmento de datos de información carga útil de dos mensajes FlexRay.

Posteriormente representada como 209, en el módulo de comunicaciones 100, se encuentra la unidad de tiempo global (Global Time Unit GTU), la cual es competente para la representación del intervalo de tiempo global en FlexRay, por tanto de los microcilos  $\mu T$  y de los macrocilos MT. Del mismo modo se encuentra reglamentada, mediante la unidad de tiempo global 209, la sincronización del tiempo tolerante a fallos del contador de ciclos (Cycle Counter) y el control del ritmo de ciclos en el segmento estático y dinámico de FlexRay. Como bloque 210 se encuentra representado el sistema de control universal (System Universal Control SUC), a través del cual se controlan y dirigen los modos de operación del controlador de comunicaciones FlexRay. Entre ellos se cuentan el wakeup (la activación), el startup (arranque), la reintegración o integración, la operación normal (normal operation) y la operación pasiva (passive operation).

El bloque 211 muestra la red y la gestión de fallos (Network- und Error Management NEM), tal como se describe en el protocolo de especificación v2.0 FlexRay. Por último, el bloque 212 muestra el control de interrupción (Interrupt Control INT), el cual gestiona las marcas de la interrupción con respecto a fallos y al estado (status and error interrupt flags) y controla, por tanto dirige las salidas de interrupción 219 hacia el CPU participante 102. El bloque 212 contiene además un contador o temporizador para generar las interrupciones de tiempo o las interrupciones del contador.

Para la comunicación dentro de una red FlexRay pueden configurarse objetos de mensaje, o sea mensajes (Mensaje Búfer) de hasta 254 bytes de datos. La memoria de mensajes 300, especialmente, consiste en una memoria RAM de mensajes (Message RAM) que puede almacenar, por ejemplo, hasta un máximo de 128 objetos de mensaje. Todas las funciones que atañen al tratamiento, así como a la gestión de los mensajes en sí mismos, se implementan en el gestor de mensajes o administrador de mensajes. Éstas son, por ejemplo, el filtrado de aceptación, la transferencia de mensajes entre ambos bloques del controlador del protocolo FlexRay 207 y 208 y la memoria de mensajes 300, por tanto de la memoria RAM, así como el control de la secuencia de emisión y la provisión de datos de configuración, por tanto de datos de estado.

Un CPU externo, o sea un procesador externo del participante 102, puede acceder en forma directa, mediante la interfaz del participante 107, con la parte específica del participante 204, al registro del módulo de comunicaciones FlexRay100. De esta manera se utiliza una gran cantidad de registros. Estos registros se constituyen para configurar y dirigir el controlador del protocolo FlexRay, por tanto las interfaces del módulo 207 y 208 del gestor de mensajes (manejador de mensajes MHD) 200, la unidad de tiempo global 209 (Global Time Unit GTU), el sistema de control universal (System Universal Control SUC) 210, la red y la gestión de fallos (Network and Error Management NEM) 211, el control de interrupción (Interrupt Control INT) 212, así como el acceso a la RAM de mensajes, por tanto a la de memoria de mensajes 300 y de este modo consignar el correspondiente estado. Partes de este registro, como mínimo, se abordan en detalle en las figuras 4 a 6 y 7 a 9. Un módulo de configuración FlexRay 100 como el

descrito, conforme a la invención, posibilita la conversión simple del protocolo de especificación v2.0 FlexRay, mediante el cual fácilmente se puede generar un ASIC (Circuito Integrado de Aplicaciones Específicas) o un microcontrolador con la funcionalidad FlexRay correspondiente integrada.

5 A través del módulo de comunicaciones FlexRay 100 descrito puede ser sustentada completamente la especificación del protocolo FlexRay, en particular la especificación v2.0 y, con ello, pueden ser configurados, por ejemplo, hasta 128 mensajes, así como objetos de mensaje. De este modo, resulta una memoria de mensajes que puede ser configurada de forma flexible para el almacenamiento de una cantidad diferente de objetos de mensaje en función del tamaño del respectivo campo de datos, así como del área de datos del mensaje. Con ello, de forma ventajosa, pueden ser configurados mensajes u objetos de mensaje que poseen campos de datos con una extensión diferente. La memoria de mensajes 300 se encuentra diseñada de forma ventajosa como FIFO (first in-first out -primero en entrar, primero en salir), de modo que resulta una recepción FIFO configurable. Cada mensaje, así como cada objeto de mensaje en la memoria puede ser configurado como un objeto de memoria de recepción (Receive-Buffer), como un objeto de memoria de transmisión o como parte de la recepción FIFO configurable. Del mismo modo, es posible un filtrado de aceptación en la trama de identificación (Frame-ID), canal de identificación (Channel-ID) y un contador de ciclos (Cycle-Counter) en la red FlexRay. De manera ventajosa, asimismo, se prevén enmascaramientos de interrupción del módulo.

En la figura 3 se describe detalladamente la división de la memoria de mensajes. Para lograr la funcionalidad exigida de un controlador de comunicaciones FlexRay se precisa una memoria de mensajes para la relocalización de los búfers de mensajes a transmitir (Transmit Búfer Tx) así como el almacenamiento de mensajes recibidos sin fallos (Receive Búfer Rx). Un protocolo FlexRay permite mensajes con un área de datos, por tanto de un área Payload de 0 a 254 bytes. De esta forma, en la figura 2 se representa la memoria de mensajes como parte del módulo de comunicaciones FlexRay 100. El método que se describe a continuación, así como la correspondiente memoria de mensajes 300, describen el almacenamiento de mensajes a transmitir, así como de mensajes recibidos, especialmente mediante el uso de una Random Access Memory (RAM), con lo cual es posible, a través del mecanismo conforme a la invención, almacenar una cantidad variable de mensajes en una memoria de mensajes del tamaño previsto. Al mismo tiempo, la cantidad de mensajes almacenables depende del tamaño de las áreas de datos de cada mensaje, por lo cual en primer lugar se minimiza el tamaño precisado para el almacenamiento sin reducir el tamaño de las áreas de datos y en segundo lugar se aprovecha la memoria en forma óptima. A continuación se describe en detalle esta división variable de una memoria de mensajes basada en RAM en especial para un controlador de comunicaciones FlexRay.

Para la implementación se considerará, a modo de ejemplo, una memoria de mensajes con una longitud de palabra predeterminada de  $n$  bits, por ejemplo 8, 16, 32, etc.; así como una profundidad de almacenamiento de palabras  $m$  ( $m, n$  equivalen a números naturales). De esta manera, la memoria de mensajes 300 se divide en dos segmentos, un segmento Header (de cabecera) o segmento de inicio HS y un segmento de datos DT (Payload Section, Payload Segment). Por mensaje se delinea un área inicial HB y un área de datos DB. Para los mensajes de  $0,1$  bits  $k$  ( $k$  como número natural), se delimitan segmentos Header (de cabecera) o segmentos de inicio HB0, HB1 a HB $k$  y áreas de datos DB0, DB1 a DB $k$ . Dentro de un mensaje, se diferencia así entre datos primarios y datos secundarios, de modo que los datos primarios, datos de configuración o datos de estado se corresponden con respecto al mensaje FlexRay y se clasifican respectivamente en un área de inicio HB (HB0, HB1, ..., HB $k$ ). Los datos secundarios, los cuales equivalen a los datos útiles propiamente dichos que deben ser transmitidos, se clasifican respectivamente en un área de inicio DB (DB0, DB1, ..., DB $k$ ). De este modo surge, para los datos primarios por cada mensaje, una primera amplitud de datos (medida en bits, bytes o por palabras almacenadas) y, para los datos secundarios de un mensaje, una segunda amplitud de datos (también medida en bits, bytes o por palabras almacenadas), donde la segunda amplitud de datos por mensaje puede ser diferente. La división entre el segmento de inicio HS y el segmento de datos DS es variable en la memoria de mensajes 300, esto significa que no existe ningún límite previsto entre las áreas. La división entre el segmento de inicio HS y el segmento de datos DS depende de la cantidad  $k$  de mensajes, así como también de la segunda amplitud de datos, por tanto de la amplitud de los datos útiles propiamente dichos, de un mensaje, por tanto de todos los mensajes  $k$  en conjunto. A los datos de configuración KDO, KD1 hasta KD $k$  del respectivo mensaje, se le adjunta en forma directa un elemento indicador o puntero de datos DP0, DP1 hasta DP $k$ . En la conformación especial se le adjunta a cada área de inicio de HB0, HB1a HB $k$  una cantidad fija de palabras almacenadas, en este caso dos, de manera que siempre un dato de configuración KD (KD0, KD1, ..., KD $k$ ) y un elemento indicador DP (DP0, DP1, ..., DP $k$ ) se clasifica conjuntamente en un área de inicio HB. A este segmento de inicio HS con el área de inicio HB, cuyo tamaño, por tanto amplitud de datos, depende de la cantidad  $k$  de mensajes a almacenar, se conecta un segmento de datos DS para el almacenamiento de los mensajes propiamente dichos D0, D1 a D $k$ . Este segmento de datos (o sección de datos) DS depende de los datos de mensaje clasificados, en este caso, por ejemplo, en DB0 seis palabras, en DB1 una palabra, y en DB $k$  dos palabras. Los respectivos elementos indicadores DPO, DP1 a DP $k$ , de este modo, se muestran siempre al comienzo, o sea, en la dirección inicial del área de datos correspondiente DB0, DB1 a DB $k$ , en los cuales los datos D0, D1 a D $k$  del respectivo mensaje se clasifican de 0, 1 a  $k$ . Debido a ello, la división de la memoria de mensajes 300 entre el segmento de inicio HS y el segmento de datos DS es variable y depende de la cantidad  $k$  de mensajes en sí mismos, así como de la respectiva amplitud de los datos de un mensaje, por tanto de

la amplitud total de la segunda amplitud de datos. Si se configuran pocos mensajes, se disminuye el segmento de cabecera HS y al área que se libera en la memoria de mensajes 300 puede ser utilizada como suplemento del segmento de datos DS para el almacenamiento de datos. Mediante esta variabilidad se puede garantizar un óptimo aprovechamiento, de manera que también es posible la utilización de memorias más pequeñas. El segmento de datos FDS, en especial su tamaño, igualmente dependiente de la cantidad k de mensajes almacenados y de la respectiva amplitud de datos de los mensajes, es por tanto mínimo y puede incluso ser igual a 0.

Junto a la utilización de elementos indicadores es también posible clasificar los datos primarios y secundarios, por tanto los datos de configuración KD (KDO, KD1, ..., KDk) y los datos propiamente dichos D (D=, D1, ... ,Dk) en una secuencia prevista, de manera tal que la secuencia del área de inicio HBO hasta HBk en el segmento de cabecera HS y la secuencia del área de datos DBO hasta DBk en el segmento de datos DS respectivamente, son idénticas. Así se podría incluso, eventualmente, renunciar a un elemento indicador.

En una conformación especial, se le adjunta a la memoria de mensajes un generador corrector, en especial un elemento generador de bits de paridad y un verificador corrector, en especial un elemento verificador de bits de paridad, para garantizar la corrección de los datos almacenados en HS y DS, pudiendo clasificarse, por palabra almacenada o por área (HB o DB), por una suma de control, en especial como una verificación de bits de paridad. También se puede pensar en otros indicadores de control, por ejemplo, en CRC (Cyclic redundancy check – control de redundancia cíclica-) o en indicadores de una mayor potencia, como ECC (Error Code Correction – código corrector de errores-). Comparada con una división fija de la memoria de mensajes, se presentan las siguientes ventajas:

El usuario puede decidir durante la programación si desea utilizar una gran cantidad de mensajes con un campo de datos pequeño o una pequeña cantidad de mensajes con un gran campo de datos. Durante la configuración de mensajes con áreas de datos DB de diferentes tamaños se aprovecha el espacio de memoria presente de forma óptima. El usuario tiene la posibilidad de utilizar el área de almacenamiento de datos en común para diferentes mensajes.

Durante la implementación del controlador de comunicaciones a una conexión integrada, mediante una adaptación de la profundidad de memoria (cantidad m de palabras) de la memoria utilizada, se puede adecuar el tamaño de la memoria de mensajes 300 a las necesidades de la aplicación, sin modificar otras funciones del controlador de comunicaciones.

A continuación se describe en detalle, mediante las figuras 4 a 6 y 7 a 9, el acceso del procesador CPU, por tanto de la escritura y la lectura de datos de configuración, o sea, de datos de estado y de los datos propiamente dichos, a través de la disposición de la memoria búfer 201 y 202. Por este medio se intenta lograr un desacoplamiento con respecto a la transmisión de datos, el cual pueda asegurar la integridad de los datos y al mismo tiempo, garantizar una alta velocidad de transmisión. El control de estas operaciones tiene lugar mediante el gestor de mensajes 200, el cual será descrito más adelante en las figuras 10, 11 y 12.

En las figuras 4, 5 y 6 se explican en detalle, en primer lugar, los accesos de escritura de la memoria de mensajes 300 a través del procesador CPU del participante 102 mediante la memoria de acceso 201. La figura 4 muestra además nuevamente el módulo de comunicaciones 100, donde, para mayor claridad, sólo se muestran las partes aquí relevantes del módulo de comunicaciones 100. Este módulo es, en primer lugar, el gestor de mensajes 200, responsable del control de los resultados, así como dos registros de control 403 y 404, los cuales pueden ser colocados, tal como se representa, por fuera del gestor de mensajes 200 en el módulo de comunicaciones 100, pero que también pueden estar contenidos dentro del gestor de mensajes 200. 403 representa el registro de peticiones de entrada (Input Búfer Command Request Register) y 404 el registro de enmascaramiento de entrada (Input Buffer Command Mask Register). Los accesos de escritura del procesador CPU 102 se producen en la memoria de mensajes 300 por medio de una memoria de acceso interconectada 201 (Input Búfer). Esta memoria de acceso interconectada 201 se exhibe dividida, por tanto duplicada, a saber en una sub memoria búfer 400 y una memoria espejo 401 perteneciente a la sub memoria búfer. Con esto, puede tener lugar, como se describe a continuación, un acceso continuo del controlador CPU 102 a los mensajes, por tanto a los objetos de mensaje y a los respectivos datos de la memoria de mensajes, y puede asegurarse la integridad de los datos y una transmisión rápida.

El control de los accesos se produce mediante el registro de peticiones de entrada 403 y el registro de enmascaramiento de entrada 404. En el registro 403, en la figura 5, se representan con los números 0 a 31, aquí a modo de ejemplo, los respectivos espacios bit en 403 para un ancho de 32 bits. Esto mismo es válido para el registro 404 y los espacios bit 0 a 31 en el registro de enmascaramiento de entrada 404 de la figura 6.

A modo de ejemplo, los espacios bit 0 a 5, 15, 16 a 21 y 31 del registro 403 mantienen, con respecto al control de desarrollo, una función especial. De esta manera se puede inscribir en los espacios bits 0 a 5 del registro 403 un identificador IBRH (Input Búfer Request Host) como identificador de mensaje. De la misma manera, se puede inscribir un identificador IBRS (Input Buffer Request Shadow) en los espacios bit del registro 403. Del mismo modo

se pueden inscribir en el espacio de registro 15 de 403, IBSYH y en el espacio de registro 31 de 403, IBSYS como identificadores de acceso. Los espacios 0 a 2 del registro 404 se encuentran también marcados, donde en 0 y 1, junto con LSHH (Load Header Section Host – segmento de cabecera de la carga útil de datos-) y LDSH (Load Data Section Host -segmento de la carga útil de datos-) se inscriben otros identificadores como identificadores de datos. Estos identificadores de datos se encuentran aquí conformados de la manera más simple, o sea conformados como un bit. En el espacio bit 2 del registro 404 se inscribe un identificador inicial con STXRH (Set Transmission X Request Host – configuración de transmisión a petición del sistema principal-).A continuación se describe el desarrollo del acceso de escritura a la memoria de mensajes 300 mediante el búfer de acceso 201.

El procesador CPU 102 escribe los datos de los mensajes a transferir en la memoria búfer de acceso 201. De este modo, el procesador CPU 102 puede escribir sólo los datos de configuración e inicio KD de un mensaje para el segmento de cabecera HS de la memoria de mensajes 300 o sólo los datos propiamente dichos, los datos a transmitir de un mensaje D para el segmento de datos DS de la memoria de mensajes 300 o escribir ambos. Qué parte de un mensaje, por tanto qué datos de configuración deben ser transmitidos, se determina por medio de identificadores de datos LSHH y LDSH en el registro de enmascaramiento de entrada 404. Así se determina, mediante LSHH, si los datos de inicio, por tanto las configuraciones de datos KD, deben ser transmitidas o si los datos D deben ser los transmitidos. Debido a que la memoria búfer de acceso 201 se encuentra conformada por dos partes, o sea, por una parte de la memoria búfer 400 y una memoria espejo 401 perteneciente a la memoria búfer 400 y a que debe producirse un acceso alternado, se proporcionan, como opuestos a LSHH y LDSH, otros dos identificadores de área, relativos a la memoria espejo 401. Estos identificadores de área, en los espacios bit 16 y 17 del registro 404, se caracterizan como LHSS (Load Header Section Shadow – espejado del segmento de cabecera de la carga útil de datos-) y LDSS (Load Data Section Shadow – espejado del segmento de la carga útil de datos). A través de estos se dirige el proceso de transmisión con respecto a la memoria espejo 401.

Si se dispone del bit de inicio, por tanto del identificador de inicio STX-RH (Set Transmission X Request Host) en el espacio bit 2 del registro de enmascaramiento de entrada 404, se asienta automáticamente, de acuerdo a la transferencia efectuada de los respectivos datos de configuración y/o de los datos propiamente dichos a transmitir en la memoria de mensajes 300, una petición de transmisión (Transmission Request) para el objeto de mensaje correspondiente. Esto significa, que mediante este identificador de inicio STXRH, se dirige y en especial se da inicio, a la transmisión automática de un objeto de mensaje a ser transmitido.

El opuesto correspondiente aquí en la memoria espejo 401 es el identificador de inicio STXRS (SetTransmission X Request Shadow – espejado de la configuración de transmisión a petición del sistema principal), el cual, a modo de ejemplo, está contenido en el espacio bit 18 del registro de enmascaramiento de entrada 404 y también aquí, de la manera más simple, se constituye como un bit. La función de STXRS es análoga a la función de STXRH, sólo que con respecto a la memoria espejo 401.

Cuando el procesador CPU 102 intercambia el identificador de mensajes, en especial el número de los objetos de mensaje, en la memoria 300, en la cual se deben transmitir los datos de la memoria búfer de acceso 201, en los espacios bit 0 a 5 del registro de peticiones de acceso 403, por consiguiente escritos conforme a IBRH, se intercambian la sub memoria búfer 400 de la memoria búfer de entrada 201 y de la respectiva memoria espejo 401, tal como se indica mediante la flecha semicircular. De este modo se da inicio, por ejemplo, a la transmisión de datos hacia la memoria de mensajes 300. La transmisión de datos hacia la memoria de mensajes 300 se produce en sí misma desde la memoria espejo 401. Al mismo tiempo, se intercambian las áreas de registro IBRH e IBRS. De la misma forma se intercambian LSHH y LDSH en oposición a LHSS y LDSS. STXRH con STXRS se intercambian en igual medida. IBRS indica con esto la identificación del mensaje, de este modo, el número del objeto de mensaje para una transmisión. Por tanto se inicia una transferencia desde la memoria espejo 401. De este modo, la transferencia se refiere a qué área dentro de una memoria de mensajes 300 se obtiene como último dato (KD y/o D) de la memoria espejo 401. Mediante el identificador (1 bit nuevamente a modo de ejemplo) IBSYS (Input Búfer Busy Shadow – entrada de búfer espejado ocupado), en el espacio bit 31 del registro de peticiones de acceso 403, se indica si en ese momento se está produciendo una transmisión con participación de la memoria 401. De esta manera, se transmite desde la memoria espejo 401, a modo de ejemplo, en IBSYS=1 y en IBSYS=0. Este bit IBSYS se dispone, a modo de ejemplo, mediante la escritura de IBRH como espacio bit 0 a 5 en el registro 403, para señalar que se ha iniciado una transferencia entre la memoria espejo 401 y la memoria de mensajes 300. Al finalizar dicha transmisión de datos hacia la memoria de mensajes 300, IBSYS se coloca nuevamente en su lugar.

Mientras se está realizando la transferencia desde la memoria espejo 401, el procesador CPU 102 puede escribir los siguientes datos a transferir en la memoria de acceso, por tanto en la sub memoria búfer 400. Mediante la ayuda de otro identificador de acceso, por ejemplo, IBSYH (Input Búfer Busy Host – entrada de búfer espejado ocupado del sistema principal), en el espacio de bit 15 del registro 403, se puede perfeccionar aún más la identificación. Si el procesador CPU 102 escribe IBRH, o sea los espacios de bits 0 a 5 del registro 403 durante la realización de una transmisión entre la memoria espejo 401 y la memoria de mensajes 300, entonces IBSYS=1, de esta manera IBSYH se coloca en el registro de peticiones de acceso 403. Tan pronto como concluye la transferencia, por tanto la transmisión que se está realizando, se da inicio a la transferencia requerida y el bit IBSYH se coloca nuevamente en

su lugar. El bit IBSYS permanece todo el tiempo marcado, para indicar que los datos son transferidos a la memoria de mensajes. Todos los bits utilizados como ejemplos de ejecución pueden conformarse con más de un bit como indicadores. La separación de un bit es ventajosa ya que permite una economización de recursos durante el procesamiento.

- 5 El mecanismo así descrito le permite al procesador CPU 102 transferir datos en forma continua, a los objetos de mensaje compuestos por un área de inicio HB y un área de datos DB que se encuentran en la memoria de mensajes 300, siempre y cuando la velocidad de acceso del procesador CPU 102 en la memoria búfer de entrada 201 sea menor o igual al índice de transferencia de datos interno del módulo IP FlexRay del módulo de comunicaciones 100.

10 En las figuras 7, 8 y 9 se explican en detalle los accesos de lectura en la memoria de mensajes 300 a través del procesador CPU 102 o participante CPU 102, mediante la de memoria búfer de salida o la memoria búfer de devolución 202. La figura 7 muestra, a su vez, nuevamente el módulo de comunicaciones 100, donde, para mayor claridad, sólo se muestran las partes aquí relevantes del módulo de comunicaciones 100. Este módulo es, en primer lugar, el gestor de mensajes 200, responsable del control de los resultados, así como dos registros de control 703 y 704, los cuales pueden ser posicionados, tal como se representa, por fuera del gestor de mensajes 200 en el módulo de comunicaciones 100, pero que también pueden estar contenidos dentro del gestor de mensajes 200. 703 representa el registro de peticiones de salida (Output Búfer Command Request Register) y 704 el registro de enmascaramiento de salida (Output Búfer Command Mask Register). Los accesos de lectura del procesador CPU 102 se producen en la memoria de mensajes 300 por medio de una memoria búfer de salida 202 (Output Búfer). Esta memoria búfer de salida 202 se exhibe dividida, por tanto duplicada, a saber en una sub memoria búfer 701 y una memoria espejo 700 perteneciente a la sub memoria búfer. Con esto, puede tener lugar, como se describe a continuación, un acceso continuo del controlador CPU 102 a los mensajes, por tanto a los objetos del mensaje y a los respectivos datos de la memoria de mensajes, y puede asegurarse la integridad de los datos y una transmisión rápida en la dirección contraria de la memoria de mensajes hacia el procesador. El control de los accesos se produce mediante el registro de peticiones de salida 703 y el registro de enmascaramiento de salida 704. En el registro 703 se representan con los números 0 a 31, aquí a modo de ejemplo, los respectivos espacios bit en 703 para un ancho de 32 bits (véase la figura 8). Esto mismo es válido para el registro 704 y los espacios bit 0 a 31 en 704 (véase la figura 9).

A modo de ejemplo, los espacios bit 0 a 5, 8 y 9, 15, 16 a 21 del registro 703 mantienen, con respecto al control de desarrollo del acceso de lectura, una función especial. De esta manera se puede inscribir en los espacios bits 0 a 5 del registro 703 un identificador OBRS (Output Búfer Request Shadow) como identificador de mensaje. De la misma manera, se puede inscribir un identificador del sistema principal de salida del búfer OBRH (Output Buffer Request Host) en los espacios bit 16 a 21 del registro 703. Los espacios 0 a 1 del registro de salida 704 se encuentran también marcados, donde en 0 y 1, y junto con RDSS (Read Data Section Shadow- espejado del segmento de lectura de datos-) y RHSS (Read Header Section Shadow – espejado del segmento de lectura de la cabecera-) se inscriben otros identificadores como identificadores de datos. Otros identificadores de datos, con RDSH y RSHS se proporcionan en los espacios bit 16 y 17. Estos indicadores de datos se conforman de un bit respectivamente, aquí también en la forma más simple, a modo de ejemplo. En el espacio bit 9 del registro 703 se inscribe un indicador de inicio REQ. Posteriormente se proporciona un identificador conmutable VIEW, el cual se inscribe, a modo de ejemplo, en el espacio bit 8 del registro 703.

40 El procesador CPU 102 requiere los datos de un objeto de mensajes de la memoria de mensajes 300, escribiendo la identificación del mensaje deseado, en especial del número del objeto de mensaje deseado, conforme a OBRS, en los espacios bit 0 a 5 del registro 703. De este modo, también aquí el procesador CPU 102 puede, como en la dirección contraria, leer sólo los datos de configuración e inicio KD de un mensaje de un área de inicio o sólo los datos propiamente dichos de un mensaje D del área de datos o leer ambos. Qué parte de los datos del área de inicio o del área de datos puede ser transmitida, se determina, en forma comparable a la dirección contraria, a través de RHSS y RDSS. Esto significa que RHSS indica si los datos de inicio deben ser leídos y RDSS indica si los datos propiamente dichos deben ser leídos.

Un indicador de inicio sirve para dar comienzo a la transmisión desde la memoria de mensajes hacia la memoria espejo 700. Esto significa que se utiliza como indicador, en el caso más simple, un bit, y, mediante la disposición del bit REQ en el espacio bit 9 en el registro de peticiones de salida 703, se da inicio a la transmisión desde la memoria de mensajes 300 hacia la memoria espejo 700. La transmisión que se está realizando es mostrada en el registro 703, mediante un bit OBSYS, aquí nuevamente tratándose del caso más simple. Para evitar colisiones es ventajoso que se coloque el bit REQ sólo cuando el bit OBSYS no esté colocado, o sea, cuando no se esté realizando una transmisión. Aquí se produce entonces también la transferencia de mensajes entre la memoria de mensajes 300 y la memoria espejo 700. La secuencia propiamente dicha podría ser controlada, en forma comparable a la dirección contraria, tal como se describe en las figuras 4, 5, y 6 (registro complementario de ocupación) y efectuarse en una variación a través de un identificador complementario, o sea un identificador conmutable VIEW en el espacio bit 8 del registro 703. Esto significa que al finalizar la transmisión se coloca nuevamente el bit OBSYS y, mediante la colocación del bit VIEW en el registro de peticiones de salida 703, se intercambian la sub memoria búfer 701 con su

equivalente memoria espejo 700, así como los accesos se intercambian y el procesador CPU 102 puede leer el objeto del mensaje requerido por la memoria de mensajes, o sea, el mensaje correspondiente. De esta manera se intercambian las celdas de registro OBRS y OBRH, aquí también en forma comparable a la dirección de transmisión contraria de las figuras 4 a 6. En igual medida se intercambian RHSS y RDSS en oposición a RSH y RDSH. Como mecanismo de protección puede aquí también preverse, que el bit VIEW sólo se coloque cuando OBSYS ya no se encuentre posicionado, por tanto cuando no tenga lugar ninguna transmisión.

De este modo, se producen accesos de lectura del procesador CPU 102 a la memoria de mensajes 300 mediante una memoria búfer de salida 202. Esta memoria búfer de salida, al igual que la memoria búfer de entrada, se encuentra diseñada dividida, por tanto duplicada, para garantizar un acceso continuo del controlador CPU 102 a los mensajes, por tanto a los objetos de mensaje y a los respectivos datos de la memoria de mensajes, logrando así mantener la integridad de los datos y una transmisión rápida.

Mediante la utilización de las memorias búfer de entrada y de salida 201, 202 descritas, se asegura que un controlador CPU pueda acceder sin interrupciones a la memoria de mensajes 300, pese a los tiempos de latencia internos del módulo.

Para asegurar la integridad de los datos, se efectúa la transmisión de los datos, en especial la transmisión en el módulo de comunicaciones 100, mediante el gestor de mensajes 200 (Message Handler MHD). Con este fin se representa en la figura 10 el gestor de mensajes 200. El gestor de mensajes puede ser representado en su funcionalidad, mediante varias máquinas de estado o máquinas automáticas de estado, o sea, mediante las llamadas máquinas de estado finito (FSM). Para esto se proporcionan como mínimo tres máquinas de estado y, en una forma de ejecución especial, cuatro máquinas de estado finito. Una primera máquina de estado finito es la IOBF-FSM, designada con el número 501 (Input/ Output Búfer State Machine – búfer de máquina de estado infinito de entrada/salida-). Esta IOBF-FSM podría estar dividida, de acuerdo a cada dirección de transmisión con respecto a la memoria búfer de entrada 201 o a la memoria búfer de salida 202, en dos máquinas de estado finito IBF- FSM (Input Búfer FSM) y OBF- FSM (Output Búfer FSM), de modo que se podría pensar en como máximo cinco máquinas de estado automáticas (IBF- FSM, OBF- FSM, TBF1- FSM, TBF2- FSM, AFSM). Proporcionar una IOBF- FSM común es, sin embargo, preferente. Al menos una segunda máquina de estado finito se divide, durante el desarrollo del ejemplo de ejecución preferente, en dos bloques 502 y 503 y sirve a los dos canales A y B con respecto a las memorias 205 y 206, tal como se describe en la Fig. 2. Para ello puede proporcionarse una máquina de estado finito para servir a los dos canales A y B o, como en la forma preferente, una máquina de estado finito TBF1- FSM, (Transient Búfer 1- – búfer transitorio 1- (206, RAM A) State Machine) denominada como 502 para el canal A y para el canal B una TBF2- FSM (Transient Búfer – búfer transitorio 2- (205, RAM B) State Machine – de máquina de estado-), denominada como 503.

En el ejemplo de ejecución preferente, para controlar el acceso de las tres máquinas de estado finito 501-503, se emplea una máquina de estado finito árbitro, conocida como AFSM, denominada como 500. Los datos (KD y/o D) se transmiten en el módulo de comunicaciones 100, mediante un temporizador, adaptado o generado a través de un medio temporizador, como, por ejemplo, un VCO (Voltage controlled oscillator – voltaje controlado por oscilador), un resonador de cuarzo, etc. El temporizador T puede ser generado en el módulo o desde afuera, por ejemplo, ser determinado como temporizador bus. Esta máquina de estado finito árbitro AFSM 500 permite a las tres máquinas de estado finito 501-503 acceder, en forma alterna, a la memoria de mensajes 300, en especial respectivamente durante un período elemental T. Esto significa que el tiempo disponible se divide en correspondencia con las peticiones de acceso de cada máquina de estado automática 501, 502, 503. Si se verifica una petición de acceso de una sola máquina de estado finito, ésta obtiene el 100% del tiempo de acceso, o sea, todos los temporizadores T. Si se verifica una petición de acceso de dos máquinas de estado automáticas, cada máquina de estado finito obtiene 50% del tiempo de acceso. Por último, si se verifica una petición de acceso de tres máquinas de estado automáticas, cada una de las máquinas obtiene 1/3 del tiempo de acceso. De esta forma se aprovecha en forma óptima el ancho de banda que se encuentra disponible.

La primera máquina de estado finito 501, o sea IOBF- FSM, realiza en caso necesario las siguientes acciones:

- transferencia de datos desde la memoria de entrada 201 hacia el objeto de mensaje seleccionado en la memoria de mensajes 300.

- transferencia del objeto de mensaje seleccionado en la memoria de mensajes 300 hacia la memoria búfer de salida 202.

La máquina de estado para el canal A 502, o sea TBF1FSM, realiza las siguientes acciones:

- transferencia del objeto de mensaje seleccionado en la memoria de mensajes 300 hacia la memoria búfer 206 del canal A.

- transferencia de datos desde la memoria búfer 206 hacia el objeto de mensaje seleccionado en la memoria de mensajes 300.

5 - búsqueda de los objetos de mensaje adecuados en la memoria de mensajes 300, donde se busca, durante la recepción del objeto de mensaje (Recibe Búfer) hasta el almacenamiento, un mensaje recibido en el canal A en el marco de un filtrado de aceptación y durante la transmisión, el primer objeto de mensaje a transmitir en el canal A (Transmit Búfer).

10 La acción de TBF2- FSM es análoga a la de la anterior, o sea a la de la máquina de estado finita para el canal A en el bloque 53. Ésta dirige la transferencia de datos del objeto de mensaje seleccionado en la memoria de mensajes 300 hacia la memoria búfer 205 por el canal B y de los datos de transferencia de la memoria búfer 205 hacia el objeto de mensaje seleccionado en la memoria de mensajes 300. También la función de búsqueda es análoga a TBF1- FSM, tras un objeto de mensaje adecuado en la memoria de mensajes 300, donde durante la recepción, el objeto de mensaje (Receive Búfer – búfer de recepción) es buscado para el almacenamiento de un mensaje recibido en el canal B en el marco de un filtrado de aceptación y, durante la transmisión, es buscado el primer objeto de mensaje a transmitir u objeto de mensaje (Transmit Búfer) en el canal B.

15 En la figura 11 se representan nuevamente el desarrollo y las rutas de transmisión. Las tres máquinas de estado 501-503 dirigen las respectivas transmisiones de datos entre cada parte. De este modo se denomina nuevamente al procesador CPU como 102, a la memoria búfer de entrada como 201 y a la memoria búfer de salida como 202. Como 300 se denomina a la memoria de mensajes y ambas memorias búfer para el canal A y el canal B como 206 y 205. Los elementos de la interfaz 207 y 208 son asimismo representados. La primera máquina de estado automática IOBF- FSM 501 dirige la transferencia de datos Z1A y Z1B, desde la memoria de acceso 201 a la memoria de mensajes 300 y desde la memoria de mensajes 300 hacia la memoria búfer de salida 202. La transmisión de datos se produce mediante buses de datos con una amplitud de palabra de, por ejemplo, 32 bits, donde también cualquier otra cantidad de bits es posible. Esto mismo es válido para la transmisión Z2 entre la memoria de mensajes y la memoria búfer 206. Esta transmisión de datos se dirige a través de TBF1- FSM, o sea de la máquina de estado 502 para el canal A. La transmisión Z3 entre la memoria de mensajes 300 y la memoria búfer 205 se dirige a través de la máquina de estado automática TBF2- FSM 503. También en este caso la transmisión de datos se produce mediante buses de datos con una amplitud de palabra de, por ejemplo, 32 bits, donde también cualquier otra cantidad de bits es posible. Por lo general, la transferencia de un objeto de mensaje completo por medio de las llamadas rutas de datos requiere varios períodos del temporizador T. Por consiguiente, se produce una división del tiempo de transmisión con respecto a los períodos del temporizador T a través del árbitro AFSM 500. En la figura 11 se representan las rutas de datos entre los componentes de almacenamiento controlados por el gestor de mensajes 200. Para garantizar la integridad de los objetos de mensaje almacenados en la memoria de mensajes 300, se deben intercambiar al mismo tiempo, de manera ventajosa, simultáneamente, datos que se encuentran en una de las rutas representadas Z1A, Z1B, así como Z2 y Z3 al mismo tiempo.

35 En la figura 12 se muestra mediante un ejemplo, cómo se dividen los temporizadores de sistema T que se encuentran a disposición, en tres máquinas de estado automáticas requeridas. En la fase 1 (I) se verifican peticiones de acceso de la máquina de estado automática 501 y la máquina de estado automática 502, lo que significa que el tiempo total se divide respectivamente a la mitad en ambas máquinas de estado automáticas. Con respecto a los períodos del temporizador en la fase 1, esto significa que la máquina de estado automática 501 en los períodos del temporizador T1 y T3 logra el acceso, mientras que la máquina de estado automática 502 lo logra en los períodos del temporizador T2 y T4. En la fase 2 (II) se verifica el acceso sólo a través de la máquina de estado automática 501, de manera que los tres períodos, o sea el 100% del tiempo de acceso de T5 a T7 en IOBF- FSM se suprime. En la fase 3 (III) se verifican peticiones de acceso de las tres máquinas de estado automáticas 501 a 503, de manera que se produce una tripartición del tiempo total de acceso. El árbitro AFSM divide el tiempo de acceso, por ejemplo de manera tal que logren el acceso, la máquina de estado automática 501 en los períodos del temporizador T8 y T11, la máquina de estado automática 502 en los períodos del temporizador T9 y T12 y la máquina de estado automática 503 en los períodos del temporizador T10 y T13. En la fase 4 (IV), por último, se produce el acceso a través de dos máquinas de estado automáticas, 502 y 503, en los dos canales A y B del módulo de comunicaciones, de manera que se produce una división de acceso de los períodos del temporizador T14 y T16 en la máquina de estado automática 502 y de los períodos del temporizador T15 y T17 en la máquina de estado automática 503.

55 La máquina de estado automática árbitro AFSM 500 se ocupa de dividir el acceso de las máquinas de estado automáticas, en forma de períodos alternados, dado el caso, de modo que más de una de las tres máquinas de estado automáticas presente una petición para el acceso a la memoria de mensajes 300. Esta forma de procedimiento asegura la integridad de los objetos de mensaje que se encuentran en la memoria de mensajes, por tanto la integridad de los datos. Si por ejemplo, el procesador CPU 102 desea leer un objeto de mensaje mediante el la memoria búfer de salida, mientras que en ese mismo momento se escribe un mensaje recibido en ese objeto de mensaje, se leerá o el estado antiguo o el estado nuevo, dependiendo de cual petición se presentó en primer lugar, de manera tal que no se produzca una colisión de los accesos en la misma memoria de mensajes 300.

5 El procedimiento descrito le permite al procesador CPU 102 en actividad, leer o escribir cualquier objeto de mensaje en la memoria de mensajes, sin que el objeto de mensaje seleccionado para la duración del acceso del procesador CPU 102 sea cerrado (Búfer Locking –cierre de búfer-) por la participación en el intercambio de datos en los dos canales del bus FlexRay. Al mismo tiempo, se asegura la integridad de los datos depositados en la memoria de mensajes mediante la jerarquización de los accesos, y, a su vez se incrementa la velocidad de transmisión mediante el aprovechamiento del ancho de banda en su totalidad.

10 Hasta el momento, tanto el participante como el microprocesador (el procesador central) del participante fue indicado mediante el signo de referencia 102 y se representó de forma equivalente en la descripción. No obstante, para la siguiente descripción de la invención, es necesaria una diferenciación. Por lo tanto, a continuación se utilizará el signo de referencia 900 para el participante FlexRay en su totalidad, mientras que el signo de referencia 102 sólo hará referencia al microprocesador (el procesador central) del participante. Para una explicación detallada debe remitirse a la figura 17, donde se representa un participante 900 conocido por el estado del arte.

15 El participante 900 conocido comprende un microcontrolador 810, un bus periférico 820 y varios, en el ejemplo de ejecución representado tres, controladores de comunicaciones FlexRay 750a, 750b, 750c. El bus periférico 820 puede estar diseñado como cualquier bus de datos interno. Como buses periféricos se utilizan frecuentemente buses de datos propios, ya que el componente denominado con 900 en su totalidad es realizado, generalmente, por el mismo y por un sólo fabricante de semiconductores. Por tanto, sólo los elementos constitutivos internos 800, 810 y 750 deben comunicarse en el componente 900 mediante el bus periférico 820.

20 La utilización de un controlador DMA no es obligatoria. Se conocen también participantes 900 en los cuales la transmisión de datos entre el microcontrolador 800 y un controlador de comunicaciones 750, así como entre los controladores de comunicaciones 750, funciona sin un controlador DMA 810.

25 El microcontrolador 800 comprende el microprocesador 102 (procesador CPU), un elemento de memoria 802, así como un bus core 804. El elemento de memoria 802, a modo de ejemplo, puede estar realizado como una "dual ported tightly coupled memory" (TCM). El microcontrolador 800 se encuentra conectado al bus periférico 820 mediante una interfaz activa "a" del microprocesador 102 y una interfaz pasiva "p" del elemento de memoria 802. Los controladores de comunicaciones 750, de este modo, se encuentran conectados sólo mediante una interfaz pasiva "p" al bus periférico 820 en el participante 900 conocido. Expresado de otro modo, entonces, el microcontrolador 800 y el controlador DMA 810 se encuentran conectados como maestros al bus periférico 820, mientras que el controlador de comunicaciones 750 se encuentra conectado al bus 820 sólo como un esclavo.

30 El microprocesador 102 configura, controla y dirige los controladores de comunicaciones 750 por separado. El microprocesador 102 lee los mensajes recibidos, los procesa y los evalúa, calcula y genera nuevos mensajes, y se ocupa de la escritura de los mensajes para el envío mediante el enlace de comunicaciones 101. De este modo, por ejemplo en operaciones sencillas de la puerta de enlace, es necesario con frecuencia leer los datos recibidos desde un controlador de comunicaciones 750 y escribirlos en uno o en otros varios controladores de comunicaciones 750 para el envío. Para la implementación de una funcionalidad de la puerta de enlace dentro del participante 900 sin la utilización de un controlador DMA 810, el microprocesador 102 transmite los datos en forma de palabras desde los controladores de comunicaciones 750 (línea punteada a) hacia el elemento de memoria 802 del microcontrolador 800 (línea punteada f) o en una memoria interna del CPU (no representada), para, dado el caso, procesarlos y a continuación copiarlos en los controladores de comunicaciones 750 correspondientes (línea punteada g). De esta manera, la frecuencia de los ciclos actualmente usual del microprocesador 102 produce más ciclos de espera, durante los cuales el microprocesador 102 espera la finalización de la transmisión de datos, de modo que se encuentra bloqueado, y no puede ocuparse de otras tareas.

45 En caso de utilizarse un controlador DMA 810, éste puede realizar la transferencia de datos entre el elemento de memoria 802 del microcontrolador 800 y los controladores de comunicaciones 750. Para ello, éste es configurado e iniciado por el microprocesador 102 (línea de puntos b). Seguidamente, el controlador DMA 810 transmite datos en forma de palabras desde los controladores de comunicaciones 750 (línea punteada c) hacia la memoria 802 del microcontrolador 800 (línea punteada d) o de forma directa entre los controladores de comunicaciones 750 (línea de puntos h). La finalización de la transmisión de datos es comunicada al microprocesador 102 mediante una interrupción (línea punteada e), de manera que el microprocesador 102 inicia nuevamente el proceso para el mensaje siguiente. El procesamiento de las interrupciones genera una gran cantidad de órdenes del procesador central, a través de las cuales se encuentran conectados una gran parte de los recursos de cálculos y de los recursos de memoria del microprocesador 102. Asimismo, a través de las interrupciones frecuentes se incrementa la variabilidad temporal (indeterminación temporal con respecto a la duración de ejecución) de las tareas interrumpidas del software.

55 En la figura 13 se representa un participante 900 conforme a la invención de un sistema de comunicaciones FlexRay. Puede observarse claramente que dos de los controladores de comunicaciones 750, ciertamente el controlador de comunicaciones 750a y el controlador de comunicaciones 750c, se encuentran conectados al bus

periférico 820 tanto mediante una interfaz pasiva "p", como mediante una interfaz activa "a". Esto significa que los controladores de comunicaciones 750a, 750c no se encuentran conectados al bus periférico 820 como esclavos, sino como maestros. Con ello, los controladores de comunicaciones 750a, 750c no sólo pueden recibir y procesar órdenes y datos de otros módulos que se encuentran conectados al bus periférico 820 (por ejemplo el microcontrolador 800 u otros controladores de comunicaciones 750), sino evaluar por sí mismos los mensajes recibidos (datos y órdenes), calcular nuevos mensajes y enviarlos hacia el microcontrolador 800 o hacia otro controlador de comunicaciones 750, mediante el bus periférico 820.

El microprocesador 102 configura y controla los controladores de comunicaciones 750, así como los datos de configuración de las interfaces activas "a" (línea punteada a). El microprocesador 102 lee los objetos de mensaje ya recibidos y copiados de forma automática en el elemento de memoria 802 (línea punteada b), los evalúa y los procesa (línea punteada c), calcula y genera nuevos objetos de mensaje y los deposita en el elemento de memoria 802 (línea punteada c). Además, es proporcionada información a los controladores de comunicaciones 750 correspondientes, de manera que estos pueden transferir los datos actuales desde el elemento de memoria 802 hacia su propia memoria de mensajes 300 (línea punteada a). La transferencia es ejecutada de forma automatizada por los controladores de comunicaciones 750a, 750c. La memoria de mensajes 300 almacena los objetos de mensaje (los así llamados mensajes búfer), junto con datos de configuración y de estado.

De forma adicional con respecto a las implementaciones conocidas por el estado del arte, los controladores de comunicaciones 750a, 750c, respectivamente, poseen una interfaz activa "a", mediante la cual la unidad lógica que se encuentra en el controlador 750, de forma autónoma, puede leer objetos de mensaje desde la memoria de mensajes 300 y escribirlos en un destino correspondiente (en el elemento de memoria 802 o en la memoria de mensajes 300 de otro controlador de comunicaciones 750), así como también leer objetos de mensaje desde una fuente (desde el elemento de memoria 802 o desde la memoria de mensajes 300 de otro controlador de comunicaciones 750) y escribirlos en la memoria de mensajes 300. El enrutamiento en y desde el elemento de memoria 802 se muestra mediante la línea punteada b y el enrutamiento entre los controladores de comunicaciones mediante la línea punteada d. La interfaz activa "a" de los controladores de comunicaciones 750a, 750c se encuentra diseñada de tal modo que ésta, de forma autónoma (incluso sin una activación a través de un microcontrolador de control 800 u otra unidad de control) puede peticionar datos desde otros módulos (por ejemplo desde el controlador de comunicaciones 750b). Para la presente invención, no es necesario que todos los módulos presenten una interfaz activa "a".

Para iniciar la transmisión de datos, los controladores de comunicaciones 750a, 750c con una interfaz activa "a" pueden verificar de forma regular si nuevos datos han ingresado al elemento de memoria 802 del microcontrolador 800 o a la memoria de mensajes 300 de uno de los controladores de comunicaciones 750. En particular, se verifica si un nuevo objeto de mensaje fue depositado en el elemento de memoria 802 y/o si un registro de control 703, 704 muestra un nuevo objeto de mensajes para una memoria búfer de salida (Output- Buffer; OBF) 202 de la memoria de mensajes 300 de uno de los controladores de comunicaciones 750.

El método conforme a la invención para el copiado activo de datos desde la memoria de mensajes del controlador de comunicaciones 750a hacia una/o desde una memoria de mensajes de otro controlador de comunicaciones 750b, 750c; controlado mediante tiempo (por ejemplo de forma periódica), puede ser iniciado/activado a través de la recepción de datos, en particular de un mensajes, a través de una modificación de los datos recibidos, a través de la activación mediante uno de los otros controladores de comunicaciones 750b, 750c o a través de la activación mediante el microprocesador 102 del controlador de comunicaciones 750a. La activación comprende, por lo general, la transmisión de una orden especial de inicio hacia la unidad lógica.

En la figura 14 se representan posibles trayectos de transmisión para datos enrutados en el participante 900 conforme a la invención. Tal como se ha descrito anteriormente, los controladores de comunicaciones 750 comprenden una memoria de mensajes 300 y pueden comprender memorias búfer, en particular al menos una memoria búfer de entrada 201 y al menos una memoria búfer de salida 202. En función de la conformación concreta de los controladores de comunicaciones 750, por ejemplo la unidad lógica del controlador de comunicaciones 750a con una interfaz activa "a" puede enrutar datos en los trayectos de transmisión explicados a continuación:

- entre la memoria de mensajes 300a y el elemento de memoria 802 del microcontrolador 800 (por lo general, sólo durante la iniciación del módulo de comunicaciones 100, de lo contrario se requiere un diseño especial del hardware),

- desde el elemento de memoria 802 hacia la memoria búfer de entrada 201a,

- desde la memoria búfer de salida 202c del otro controlador de comunicaciones 750c hacia la memoria búfer de entrada 201a,

- desde la memoria búfer de salida 202a hacia el elemento de memoria 802,
  - desde la memoria búfer de salida 202a hacia la memoria búfer de entrada 201c del otro controlador de comunicaciones 750c,
  - desde las memorias de mensajes 300a y 300c hacia la memoria búfer de salida 202a, así como 202c, y
- 5 - desde las memorias búfer de entrada 201a y 201c hacia las memorias de mensajes 300a, así como 300c.

De manera correspondiente, la unidad lógica del otro controlador de comunicaciones 750c, con una interfaz activa "a", puede enrutar datos en los siguientes trayectos de transmisión:

- entre la memoria de mensajes 300c y el elemento de memoria 802 del microcontrolador 800 (por lo general, sólo durante la iniciación del módulo de comunicaciones 100),
- 10 - desde el elemento de memoria 802 hacia la memoria búfer de entrada 201c,
- desde la memoria búfer de salida 202a del controlador de comunicaciones 750a hacia la memoria búfer de entrada 201c,
  - desde la memoria búfer de salida 202c hacia el elemento de memoria 802,
- 15 - desde la memoria búfer de salida 202c hacia la memoria búfer de entrada 201a del controlador de comunicaciones 750a,
- desde las memorias de mensajes 300a y 300c hacia la memoria búfer de salida 202a, así como 202c, y
  - desde las memorias búfer de entrada 201a y 201c hacia las memorias de mensajes 300a, así como 300c.

En la figura 15 se representa un diagrama de bloques de un controlador de comunicaciones 750 conforme a la invención con una interfaz activa "a". La interfaz 204 específica del participante, así como del cliente, se representa en detalle en la figura 16. En conjunto, la figura 16 muestra un controlador de comunicaciones 750 conforme a la invención, tal como, por ejemplo, es utilizado en un participante 900 de acuerdo a la figura 13. El controlador de comunicaciones 750 comprende un módulo de comunicaciones FlexRay 100, tal como éste fue descrito anteriormente en detalle. El módulo de comunicaciones 100 – tal como se ha mencionado- se encuentra subdividido en una parte genérica, la cual se encuentra denominada con el signo de referencia 840 y la cual puede estar conformada siempre del mismo modo, en forma independiente del microcontrolador 800 conectado y en una parte 204 específica del cliente o del participante. La parte genérica 840 es denominada también como controlador de comunicaciones core. De la parte genérica 840 del módulo de comunicaciones 100, se encuentran representadas en la figura 16, a modo de ejemplo, sólo la interfaz genérica 203 (GIF), así como las memorias búfer 201, 202, situadas aguas arriba con respecto a la memoria de mensajes 300. La memoria búfer de entrada 201 sirve para el almacenamiento temporal de mensajes para la transferencia hacia la memoria de mensajes 300. La memoria búfer de salida 202 sirve para el almacenamiento temporal de mensajes para la transferencia desde la memoria de mensajes 300 hacia el microcontrolador c 800 o hacia una memoria de mensajes 300 de otro controlador de comunicaciones 750 del participante 900. Naturalmente, el módulo de comunicaciones 100 de la figura 16 puede contener todos o sólo algunos de los elementos representados en la figura 2.

35 La parte genérica 840 del módulo de comunicaciones 100, mediante la interfaz genérica 203 (GIF), se encuentra conectada a la interfaz 204 (CIF) específica del participante de la parte específica del participante del módulo de comunicaciones 100. La interfaz genérica 203 puede ser conectada a diferentes procesadores CPU 102 específicos del cliente, mediante la interfaz 204 específica del participante. La unidad lógica del controlador de comunicaciones 750 se encuentra conformada en la interfaz 204 (CIF) específica del participante en forma de una así llamada máquina de enrutamiento 830 (motor de enrutamiento; RE). La interfaz 204 (CIF) específica del participante conecta el bus periférico 820 del microprocesador 102 a la interfaz genérica 203 (GIF) del controlador de comunicaciones core FlexRay 840. Además, se proporciona un registro de control y de estado 838, donde es depositada información de estado y de control, así como una memoria de enrutamiento 842, donde es depositada información de enrutamiento. El registro 838 y la memoria 842 pueden estar diseñados como áreas de memoria determinadas de un elemento de memoria o como Flip- Flops.

Los signos de referencia 832 y 834 denominan la interfaz pasiva "p", así como la interfaz activa "a" del controlador de comunicaciones 750. La interfaz pasiva 832 conecta el controlador de comunicaciones 750 a una interfaz activa "a" de otro módulo (por ejemplo del microcontrolador 800 o de uno de los controladores de comunicaciones 750a, 750c a una interfaz activa 834). Ésta no puede iniciar por sí misma una transferencia y, de acuerdo a ello, siempre

debe ser activada a través de otro módulo. Mediante la interfaz pasiva 832, el módulo activo en el controlador de comunicaciones 750 puede acceder a la información de control y de estado en el registro de control y de estado 838 y a la información de enrutamiento en la memoria de enrutamiento 842. La interfaz pasiva 832 es dependiente del bus periférico 820 on - chip utilizado.

5 La interfaz activa 834 puede iniciar de forma autónoma la transferencia de datos. Para ello, puede petitionar por sí misma datos desde otros módulos (por ejemplo desde el microcontrolador 800 o desde uno de los controladores de comunicaciones 750) o escribir datos en otras áreas de almacenamiento (por ejemplo en el elemento de memoria 802 o en una memoria de mensajes 300 de otro controlador de comunicaciones 750). No es de importancia si la estación secundaria consiste en un módulo activo o en uno pasivo. La interfaz 834 se encuentra bajo el control exclusivo del motor de enrutamiento 830. La configuración de la interfaz activa 834 tiene lugar mediante el registro de control y de estado 838, a través del microprocesador 102 (procesador CPU). La interfaz activa 834 es independiente del bus periférico 820 on - chip utilizado.

15 El motor de enrutamiento 830 comprende una máquina de estado que puede iniciar de forma autónoma la transferencia de un mensaje. El inicio de una transferencia de datos tiene lugar, a modo de ejemplo, controlado mediante tiempo y a través de un temporizador integrado 844 o a través de la recepción de un mensaje o de la activación mediante otro módulo con una interfaz activa "a" (por ejemplo a través del microprocesador 102 después del procesamiento de un mensaje). Éste controla la carga/almacenamiento del mensaje en el controlador de comunicaciones 750, así como la transferencia hacia/desde un módulo. El motor de enrutamiento 830, a través de las operaciones correspondientes de copiado y de desplazamiento, se encuentra en condiciones de agrupar nuevos mensajes que se depositan o que son depositados en una memoria de mensajes 300 de otro controlador de comunicaciones 750.

20 Para el control de la transferencia de datos, durante el inicio, es escrita información de enrutamiento correspondiente en un área de la memoria 842, a través del microprocesador 102 (procesador central). Con esta información, el motor de enrutamiento puede iniciar una transferencia de datos al recibir un mensaje o en caso de un control mediante tiempo. De forma adicional, otros módulos pueden enviar instrucciones para el proceso de un mensaje que después son ejecutadas por el motor de enrutamiento 830. Para el control, la (des-) activación y la indicación de propiedades/estados se encuentra presente información de control y de estado.

25 Un árbitro 836 controla las peticiones concurrentes de acuerdo a los accesos del microprocesador 102 (mediante la interfaz pasiva 832) y el motor de enrutamiento 830 en la interfaz genérica 203 (GIF) del control de comunicaciones core 840. A través de la información depositada en cada módulo activo y de la unidad lógica implementada para la transferencia de datos autónoma, puede hablarse aquí de un enrutamiento distribuido, puesto que no hay ningún nodo central y ningún controlador de comunicaciones 750 que posea información sobre la red en su totalidad. A través de la distribución de la inteligencia de enrutamiento en varios módulos de la red puede ser mejorada la disponibilidad del participante, ya que la suspensión de un módulo, necesariamente, no conduce a un colapso de todas las actividades de enrutamiento en el participante.

30 Para aumentar el paso de datos entre los módulos por separado puede ser utilizada una así llamada red de interconexión de barras cruzadas (Crossbar Switch Interconnection Network). Ésta permite la apertura simultánea de diferentes canales de comunicaciones. De este modo, dos o más (en función de la configuración del sistema) maestros (microprocesador 102 o controlador de comunicaciones 750a, 750c con interfaz activa "a") pueden realizar una transferencia de datos, respectivamente hacia otro módulo (maestro o esclavo) y transmitir los datos. Los accesos simultáneos a un mismo módulo son priorizados/arbitrados y deben ser procesados de forma secuencial. La señalización de una tarea ininterrumpida puede tener lugar mediante banderas o semáforos. El ancho de banda posible del sistema en su totalidad, de este modo, aumenta de forma múltiple. Los tiempos de latencia del microprocesador 102 y la inestabilidad de las tareas del software interrumpidas durante el tiempo del enrutamiento, asimismo, son reducidos.

35 Para el copiado activo después de la recepción de un mensaje, controlada mediante tiempo, (después de descender por debajo de un valor predeterminado del temporizador 844) o después de la activación a través de otro controlador de comunicaciones (por ejemplo el controlador de comunicaciones 750c) con la interfaz activa "a" o a través del microprocesador 102 (procesador CPU), el motor de enrutamiento 830 de un controlador de comunicaciones determinado (por ejemplo el controlador de comunicaciones 750a) con una interfaz activa "a", controla de forma autónoma los siguientes pasos, para transferir un mensaje desde la memoria de mensajes 300a del controlador de comunicaciones 750a hacia una memoria de mensajes 300b ó 300c de otro controlador de comunicaciones 750b ó 750c o hacia el elemento de memoria 802 del microcontrolador 800:

40 - representación de forma visible de los objetos de mensaje recibidos en la memoria búfer de entrada 202a del controlador de comunicaciones 750a;

- creación de una copia del objeto de mensajes recibido en la memoria búfer de entrada 201b del otro controlador de comunicaciones 750b ó 750c o en un rango de direcciones que puede ser configurado, del elemento de memoria 802 asociado al microprocesador 102, a través de operaciones de acceso de escritura mediante la interfaz activa 834; y

- 5 - en el caso de una transferencia hacia un segundo controlador de comunicaciones 750b ó 750c: inicio de la transferencia del objeto de mensaje desde la memoria búfer de entrada 201b ó 201c hacia la memoria de mensajes 300b ó 300c del segundo controlador de comunicaciones 750b ó 750c.

10 Después de la recepción de un mensaje, controlada mediante tiempo, (después de descender por debajo de un valor predeterminado del temporizador 844) o después de la activación a través de otro controlador de comunicaciones (por ejemplo el controlador de comunicaciones 750c) con la interfaz activa "a" o a través del microprocesador 102 (la activación tiene lugar a través de la escritura de los datos de configuración, como el área de almacenamiento o la cantidad de datos), el motor de enrutamiento 830 de un controlador de comunicaciones determinado (por ejemplo el controlador de comunicaciones 750a) con una interfaz activa "a", controla de forma autónoma los siguientes pasos, para transferir un mensaje desde la memoria de mensajes 300b ó 300c de otro controlador de comunicaciones 750b ó 750c o desde el elemento de memoria 802 del microcontrolador 800 hacia la propia memoria de mensajes 300a:

- representación de forma visible de los datos (del objeto de mensaje) en la memoria búfer de salida 202b ó 202c del otro controlador de comunicaciones 750b ó 750c, siempre que deban ser transmitidos datos desde otro controlador de comunicaciones 750b ó 750c hacia el controlador de comunicaciones 750a;

- 20 - creación de una copia del contenido del elemento de memoria 802, así como de la memoria búfer de salida 202b ó 202c del otro controlador de comunicaciones 750b ó 750c en la propia memoria búfer de entrada 201a a través de operaciones de acceso de lectura mediante la interfaz activa "a" en un rango de direcciones que puede ser configurado; e

- 25 - inicio de la transmisión de los datos desde la memoria búfer 201a hacia la memoria de mensajes 300a del controlador de comunicaciones (750a) del módulo de comunicaciones (100a).

A través de la utilización de la interfaz activa 834 descrita, junto con la unidad lógica en forma del motor de enrutamiento 830, resultan las siguientes ventajas:

- el microprocesador 102, en comparación con accesos mediante un controlador DMA externo (véase el signo de referencia 810 en la figura 5) se encuentra liberado de una carga de interrupciones elevada.

- 30 - una menor carga de interrupciones posibilita una inestabilidad reducida de las tareas y, con ello, una mejor previsibilidad del sistema en su totalidad.

- a través del acceso a copias de los mensajes en el elemento de memoria 802 es posible un mayor ancho de banda de acceso.

- 35 - a través del ancho de banda de acceso de mayor tamaño, el microprocesador 102 dispone de más tiempo de cálculo para otras tareas (los tiempos de latencia del procesador 102 son reducidos).

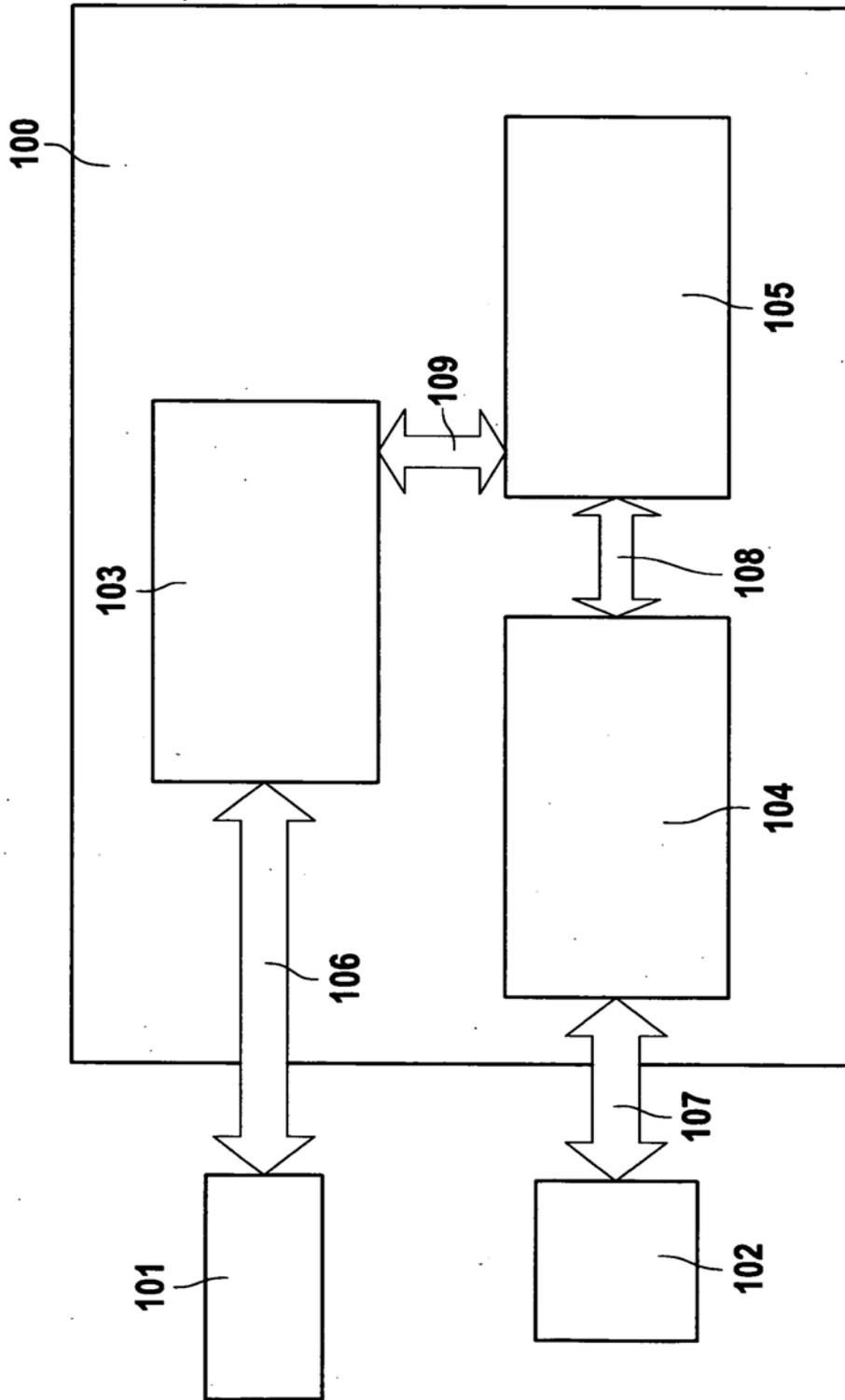
## REIVINDICACIONES

- 5 1. Participante (900) de un sistema de comunicaciones, donde el participante (900) presenta un microprocesador (102), al menos dos controladores de comunicaciones (750a, 750b, 750c) y un bus periférico (820), y donde el microprocesador (102), mediante el bus periférico (820), se encuentra en contacto con los controladores de comunicaciones (750a, 750b, 750c) y, mediante los controladores de comunicaciones (750a, 750b, 750c) se encuentra conectado, respectivamente, a un enlace de comunicaciones (101a, 101b, 101c) del sistema de comunicaciones, mediante el cual son transmitidos mensajes, **caracterizado p orque** al menos uno de los controladores de comunicaciones (750a, 750c) presenta una interfaz activa (834), mediante la cual el controlador de comunicaciones (750a, 750c) se encuentra en contacto con el bus periférico (820) y mediante la cual el controlador de comunicaciones (750a, 750c) puede enviar datos e instrucciones a otro participante del bus periférico (820), y una unidad lógica para la implementación autónoma de una funcionalidad de una puerta de enlace, **y porque** los controladores de comunicaciones (750a, 750b, 750c) presentan respectivamente una memoria de mensajes (300a, 300b, 300c) para el almacenamiento temporal de mensajes desde el enlace de comunicaciones (101a, 101b, 101c) asociado al controlador de comunicaciones (750a, 750b, 750c), o para el enlace de comunicaciones (101a, 101b, 101c), donde la unidad lógica coordina y controla un enrutamiento autónomo de datos entre la memoria de mensajes (300a, 300b, 300c) de un controlador de comunicaciones (750a, 750b, 750c) y el microprocesador (102) o entre las memorias de mensajes (300a, 300b, 300c) del controlador de comunicaciones (750a, 750b, 750c).
- 20 2. Participante (900) conforme a la reivindicación 1, **caracterizado porque** al menos unos de los controladores de comunicaciones (750a, 750b, 750c) presenta un módulo de comunicaciones (100a, 100b, 100c), el cual comprende una memoria de mensajes (300a, 300b, 300c) para el almacenamiento temporal de mensajes desde el enlace de comunicaciones (101a, 101b, 101c) asociado al controlador de comunicaciones (750a, 750b, 750c) o para el enlace de comunicaciones (101a, 101b, 101c), donde la unidad lógica coordina y controla un enrutamiento autónomo de datos entre la memoria de mensajes de un controlador de comunicaciones o entre la memoria de mensajes (300a, 300b, 300c) de al menos un módulo de comunicaciones (100a, 100b, 100c) y el microprocesador (102) o entre las memorias de mensajes (300a, 300b, 300c) de los controladores de comunicaciones (750a, 750b, 750c) y/o la memoria de mensajes (300a, 300b, 300c) de al menos un módulo de comunicaciones (100a, 100b, 100c).
- 30 3. Participante (900) conforme a la reivindicación 2, **caracterizado porque** cada módulo de comunicaciones (100a, 100b, 100c) presenta al menos una memoria búfer dispuesta entre la memoria de mensajes (300a, 300b, 300c) y el bus periférico (820), preferentemente al menos una memoria búfer de entrada (201a, 201b, 201c) y al menos una memoria búfer de salida (202a, 202b, 202c), donde la unidad lógica coordina y controla un enrutamiento autónomo de datos entre la memoria de mensajes (300a, 300b, 300c) de un controlador de comunicaciones (750a, 750b, 750c) o entre al menos una memoria búfer (201a, 201b, 201c; 202a, 202b, 202c) de la memoria de mensajes (300a, 300b, 300c) de al menos un módulo de comunicaciones (100a, 100b, 100c) y el microprocesador (102), o entre las memorias de mensajes (300a, 300b, 300c) de los controladores de comunicaciones (750a, 750b, 750c) y/o al menos una memoria búfer (201a, 201b, 201c; 202a, 202b, 202c) de la memoria de mensajes (300a, 300b, 300c) de al menos un módulo de comunicaciones (100a, 100b, 100c).
- 35 4. Participante (900) conforme a una de las reivindicaciones 1 a 3, **caracterizado porque** la unidad lógica en el controlador de comunicaciones (750a, 750c) comprende una máquina de estado.
- 40 5. Participante (900) conforme a la reivindicación 4, **caracterizado po rque** la máquina de estado se encuentra estructurada de forma fija.
- 45 6. Participante (900) conforme a la reivindicación 4 ó 5, **caracterizada porque** cada controlador de comunicaciones (750a, 750c), el cual presenta una interfaz activa (834) y una unidad lógica, presenta un árbitro (836), el cual arbitra las operaciones de acceso concurrentes del microprocesador (102) y del motor de enrutamiento (830) en una memoria de mensajes (300a, 300c) del controlador de comunicaciones (750a, 750c) o en una memoria de mensajes (300a, 300c) de un módulo de comunicaciones (100a, 100c) o en al menos una memoria búfer (201a, 201c; 202a, 202c) de una memoria de mensajes (300a, 300c) de un módulo de comunicaciones (100a, 100c).
- 50 7. Participante (900) conforme a una de las reivindicaciones 1 a 6, **caracterizado porque** cada controlador de comunicaciones (750a, 750c), el cual posee una interfaz activa (834) y una unidad lógica, presenta un registro de control y/o de estado (838), al cual tiene acceso el microprocesador (102) para la configuración, la dirección y/o el monitoreo de la funcionalidad de la puerta de enlace.
8. Participante (900) conforme a una de las reivindicaciones 1 a 7, **caracterizado porque** cada controlador de comunicaciones (750a, 750c), el cual posee una interfaz activa (834) y una unidad lógica, presenta una memoria de enrutamiento (844), a la cual tiene acceso el microprocesador (102) para la configuración, la activación y/o el monitoreo de la funcionalidad de la puerta de enlace.

9. Participante (900) conforme a una de las reivindicaciones 1 a 8, **caracterizado porque** el participante (900) forma parte de un sistema de comunicaciones FlexRay, en el cual tiene lugar una transmisión de datos entre el participante (900) y otros participantes FlexRay que se encuentran conectados al enlace de comunicaciones FlexRay (101), en base al protocolo FlexRay.
- 5 10. Método para la implementación de una funcionalidad de la puerta de enlace dentro de un participante (900) de un sistema de comunicaciones, donde el participante (900) presenta un microprocesador (102), al menos dos controladores de comunicaciones (750a, 750b, 750c) y un bus periférico (820), y donde los controladores de comunicaciones (750a, 750b, 750c), por un lado, se encuentran en contacto con el microprocesador (102) mediante el bus periférico (820) y, por otro lado, se encuentran conectados respectivamente a un enlace de comunicaciones (101a, 101b, 101c) del sistema de comunicaciones, mediante el cual son transmitidos mensajes, **caracterizado**  
10 **porque** al menos uno de los controladores de comunicaciones (750a, 750c) presenta una interfaz activa (834), mediante la cual el controlador de comunicaciones (750a, 750c) se encuentra en contacto con el bus periférico (820) y mediante la cual el controlador de comunicaciones (750a, 750c) puede enviar datos e instrucciones a otro participante del bus periférico (820) **y porque** la funcionalidad de la puerta de enlace es implementada a través de una unidad lógica en al menos un controlador de comunicaciones (750a, 750c), donde la unidad lógica coordina y controla un enrutamiento autónomo de datos entre una memoria de mensajes (300a, 300b, 300c) del controlador de comunicaciones (750a, 750b, 750c) y el microprocesador (102) o entre memorias de mensajes (300a, 300b, 300c) del controlador de comunicaciones (750a, 750b, 750c).
- 15 11. Método conforme a la reivindicación 10, **caracterizado porque** en el marco de la funcionalidad de la puerta de enlace son iniciadas y procesadas interrupciones, son fragmentados y defragmentados mensajes y son intercambiados mensajes entre controladores de comunicaciones (750a, 750b, 750c) del participante (900) y/o son intercambiados mensajes entre uno de los controladores de comunicaciones (750a, 750b, 750c) y el microprocesador (102).
- 20 12. Método conforme a la reivindicación 10 u 11, **caracterizado porque** el microprocesador (102) almacena información de control y de estado en un registro de control y/o de estado (838) de al menos un controlador de comunicaciones (750a, 750c) y, de este modo, la funcionalidad de la puerta de enlace es configurada, controlada y/o monitoreada.
- 25 13. Método conforme a una de las reivindicaciones 10 a 12, **caracterizado porque** el microprocesador (102) almacena información de enrutamiento en una memoria de enrutamiento (842) de al menos un controlador de comunicaciones (750a, 750c) y, de este modo, la funcionalidad de la puerta de enlace es controlada y/o monitoreada.
- 30 14. Método conforme a una de las reivindicaciones 10 a 13, **caracterizado porque** la unidad lógica, en al menos un controlador de comunicaciones (750a, 750c), comprende un motor de enrutamiento (830), donde son arbitradas las operaciones de acceso concurrentes del microprocesador (102) y del motor de enrutamiento (830) en la memoria de mensajes (300a, 300c) del controlador de comunicaciones (750a, 750c) o en una memoria de mensajes (300a, 300c) de un módulo de comunicaciones (100a, 100c) o en al menos una memoria búfer (201a, 201c; 202a, 202c) de la memoria de mensajes (300a, 300c) del módulo de comunicaciones (100a, 100c).
- 35 15. Método conforme a una de las reivindicaciones 10 a 14, **caracterizado porque** para el copiado activo de datos en una memoria de mensajes (300b, 300c) de otro controlador de comunicaciones (750b, 750c), los siguientes pasos son ordenados y controlados de forma autónoma por la unidad lógica de al menos un controlador de comunicaciones (750a): - representación de forma visible de los datos recibidos en una memoria búfer (202a) de al menos un controlador de comunicaciones (750a); - creación de una copia de los datos recibidos en un rango de direcciones que puede ser configurado, de un elemento de memoria (802) asociado al microprocesador (102) o en una memoria búfer (201b, 201c) de otro controlador de comunicaciones (750b, 750c), a través de operaciones de acceso de escritura mediante la interfaz activa (834); y - cuando una copia de datos es creada en la memoria búfer (201b, 201c) del otro controlador de comunicaciones (750b, 750c), inicio de una transferencia de los datos almacenados desde la memoria búfer (201b, 201c) hacia una memoria de mensajes (300b, 300c) del otro controlador de comunicaciones (750b, 750c).
- 40 45 16. Método conforme a una de las reivindicaciones 10 a 15, **caracterizado porque** para el copiado activo de datos desde una memoria de mensajes (300b, 300c) de otro controlador de comunicaciones (750b, 750c), los siguientes pasos son ordenados y controlados de forma autónoma por la unidad lógica de al menos un controlador de comunicaciones (750a): - representación de forma visible de los datos en una memoria búfer (202b, 202c) del otro controlador de comunicaciones (750b, 750c); creación de una copia del contenido de un elemento de memoria (802) asociado al microprocesador (102) o de la memoria búfer (202b, 202c) del otro controlador de comunicaciones (750b, 750c) en un rango de direcciones que puede ser configurado, de al menos una memoria búfer (201a) del controlador de comunicaciones (750a), a través de operaciones de acceso de lectura mediante la interfaz activa
- 50 55

(834); e -inicio de una transferencia de los datos almacenados desde al menos una memoria búfer (201a) hacia una memoria de mensajes (300a) del módulo de comunicaciones (100a) del controlador de comunicaciones (750a).

5 **17.** Método conforme a la reivindicación 15 ó 16, **caracterizado porque** el método es iniciado y controlado mediante tiempo, a través de la recepción de datos, en particular de un mensaje, mediante una activación a través de un controlador de comunicaciones (750a, 750c) con una interfaz activa (834), o mediante una activación a través del microprocesador (102).



**Fig. 1**

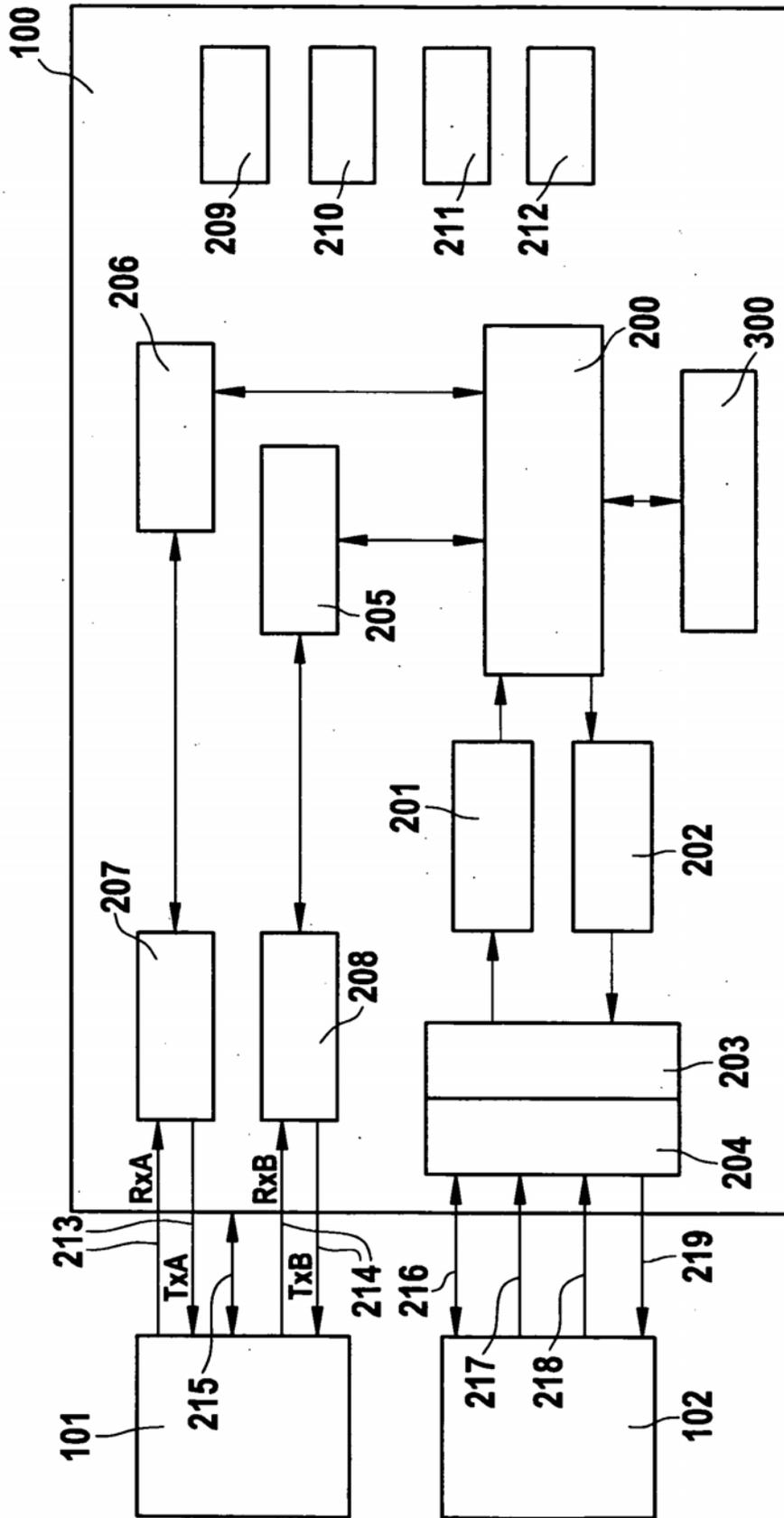


Fig. 2

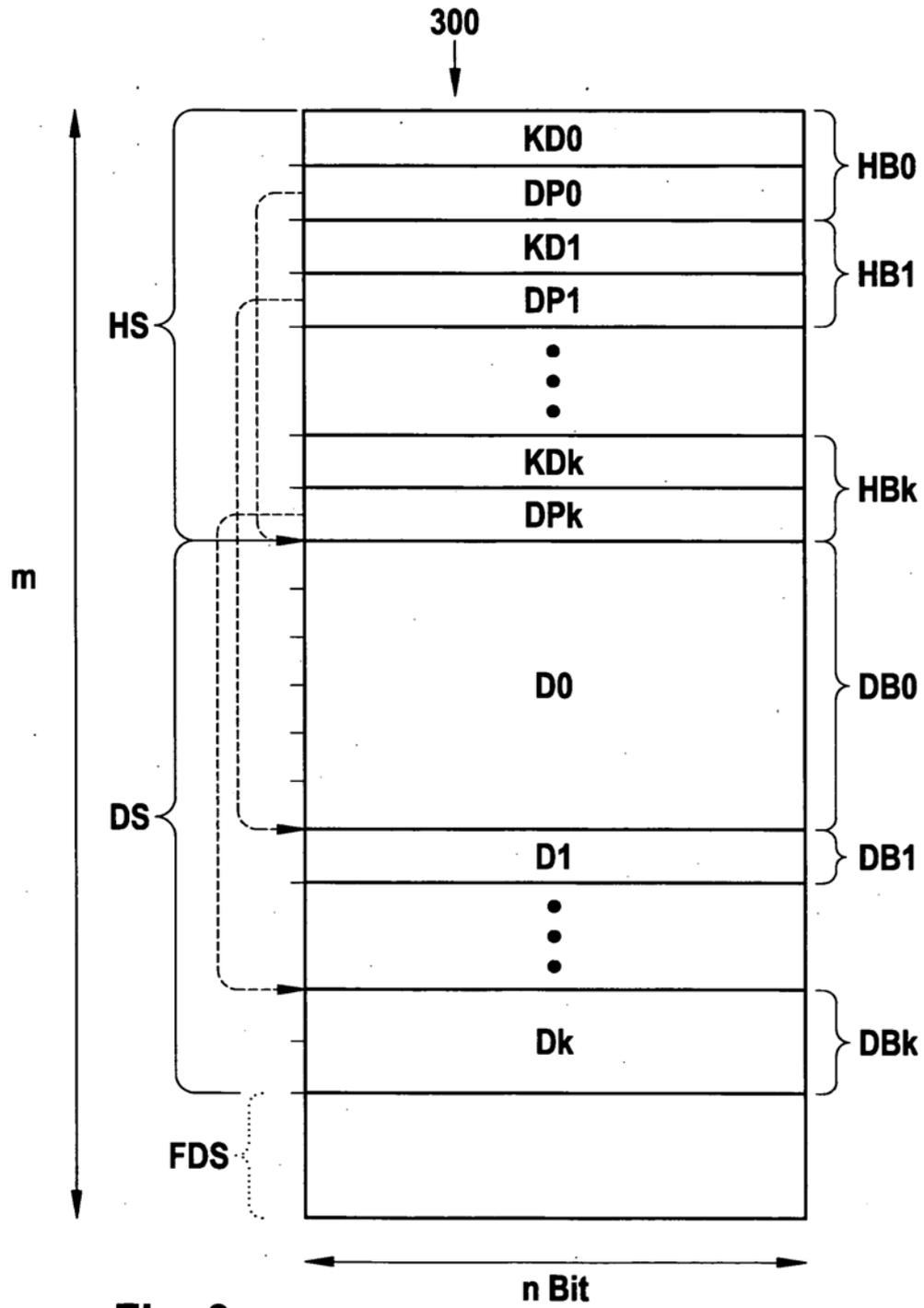


Fig. 3

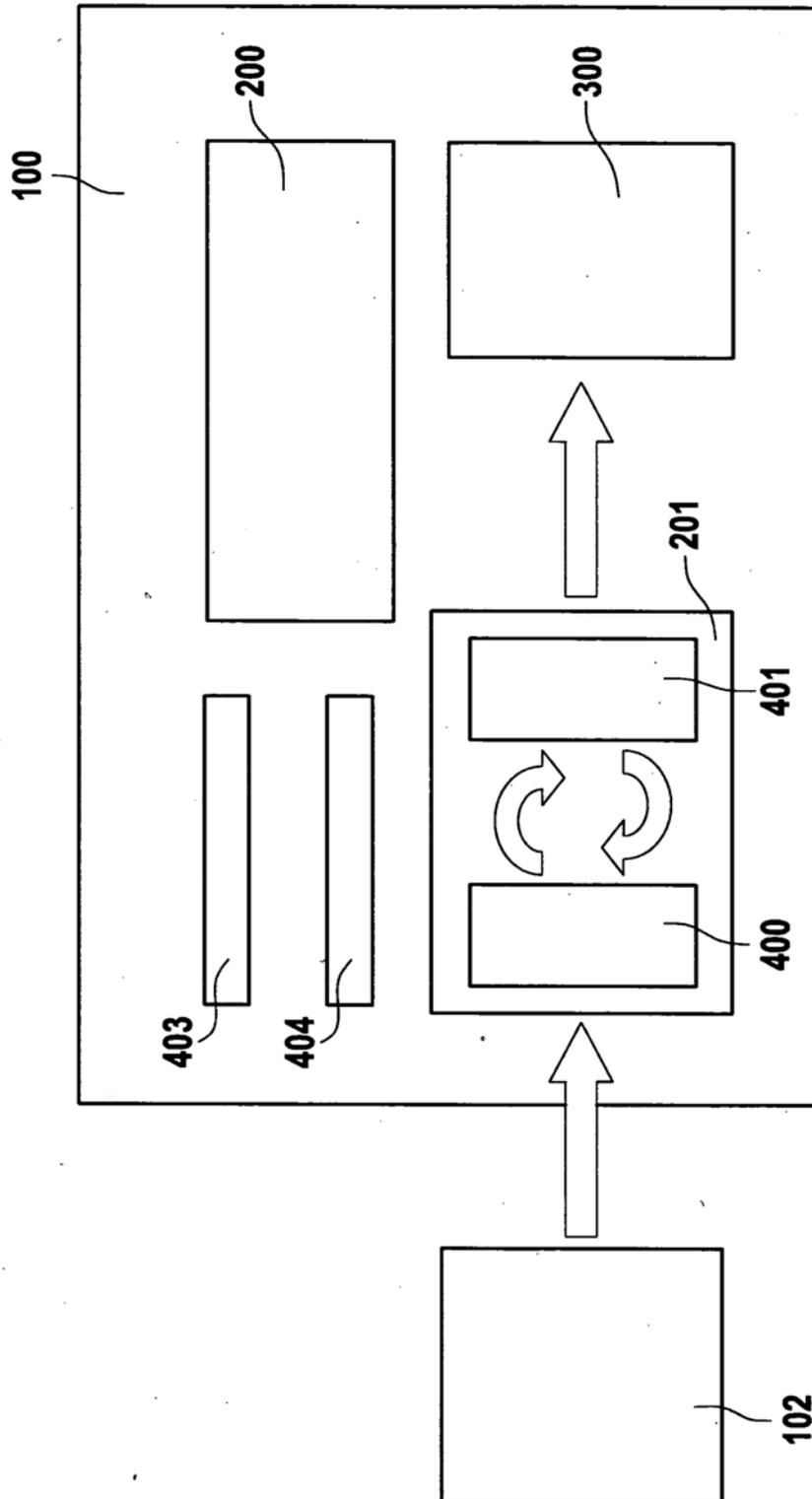


Fig. 4

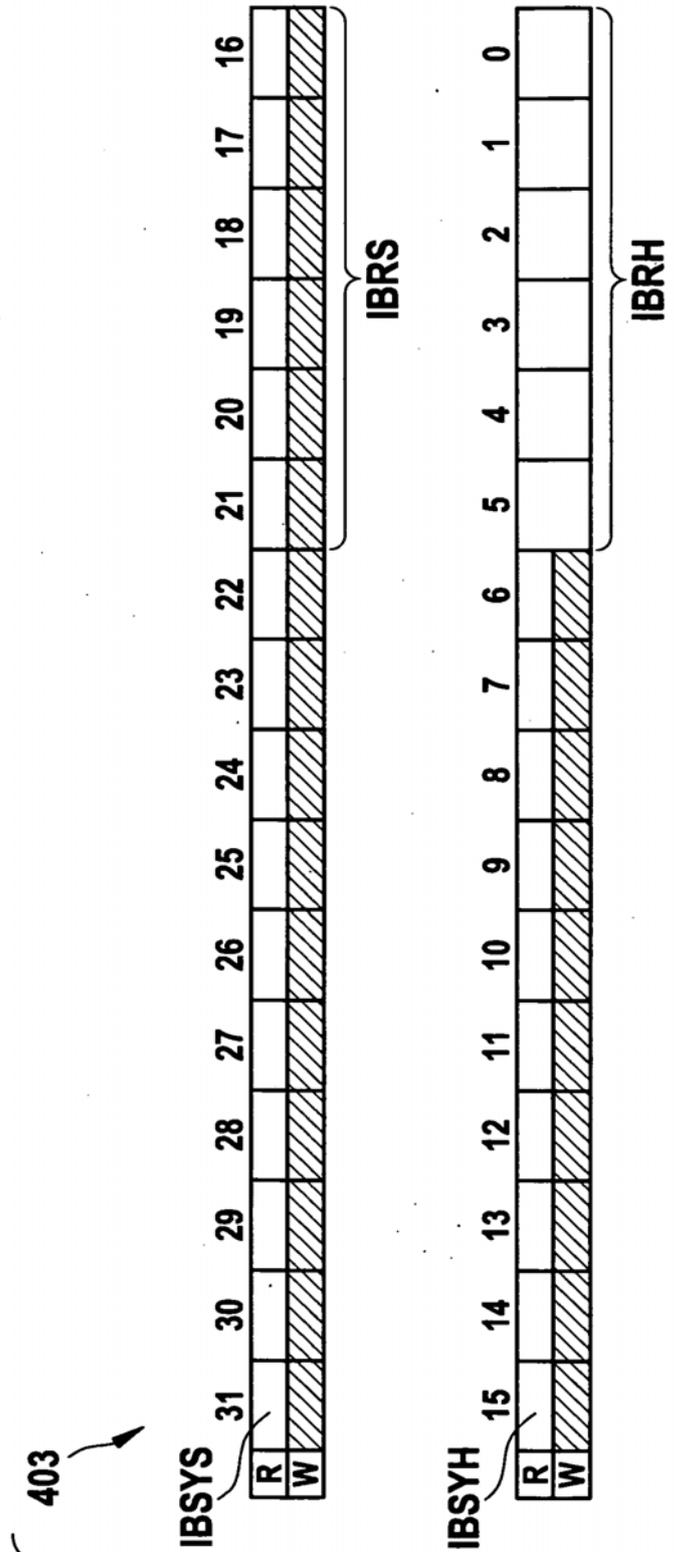


Fig. 5

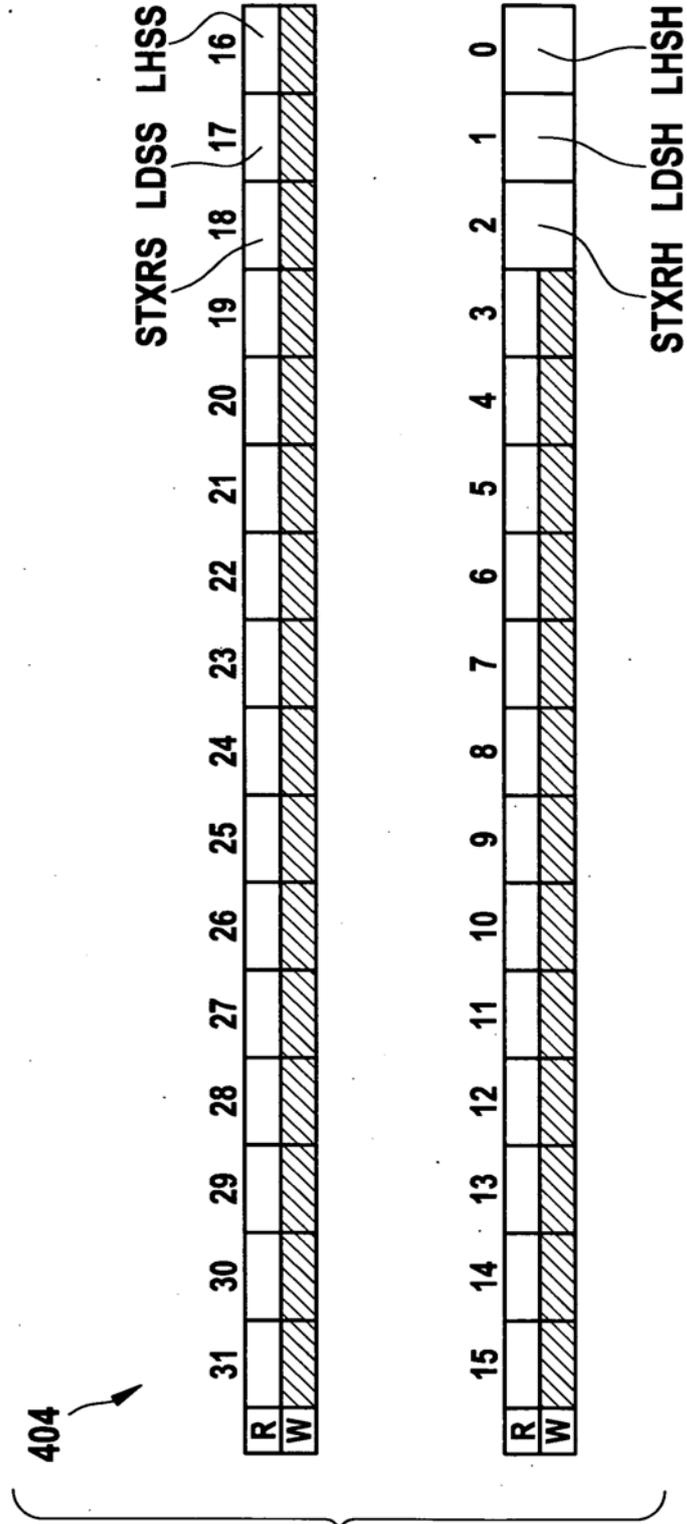


Fig. 6

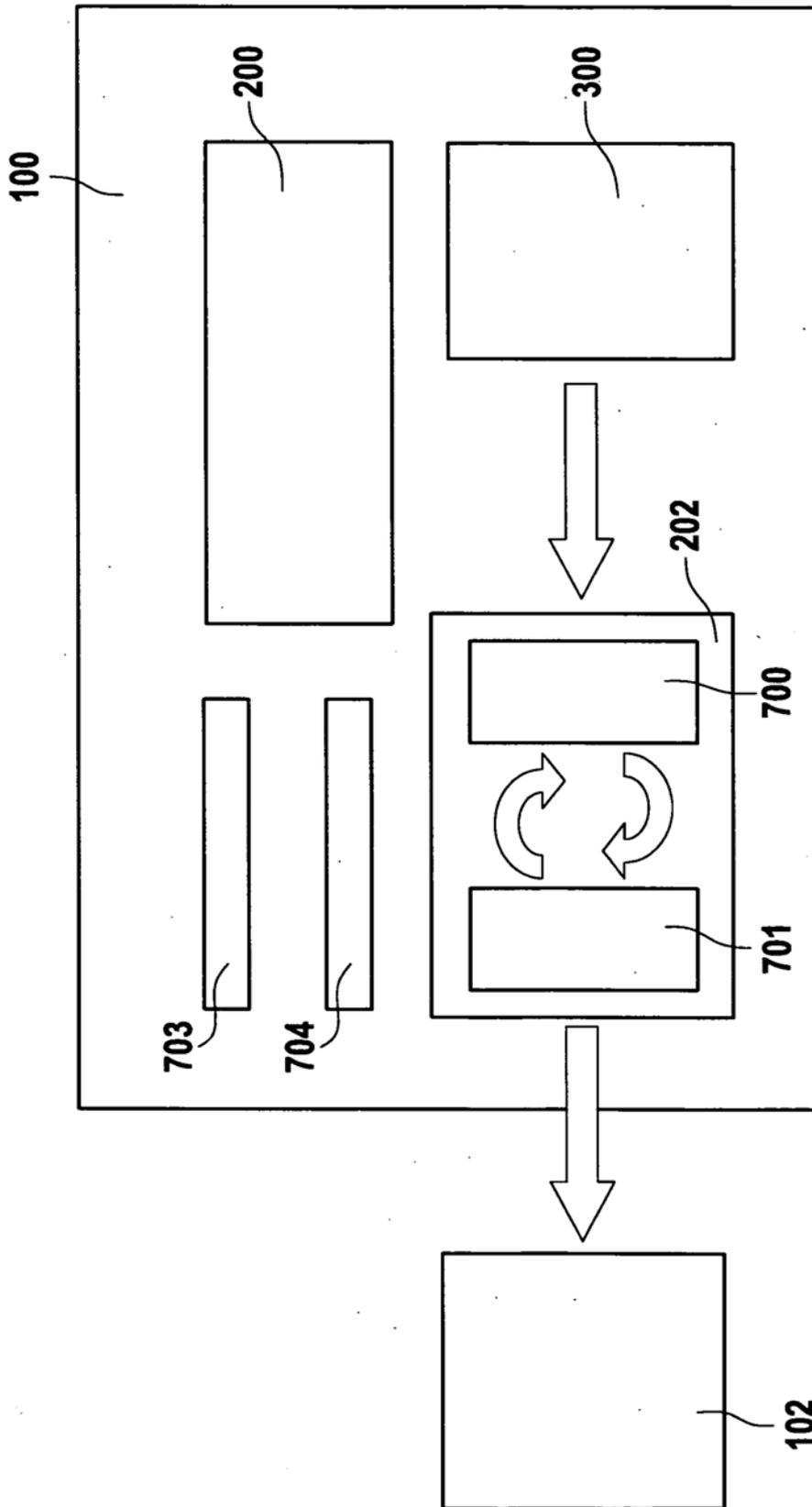


Fig. 7

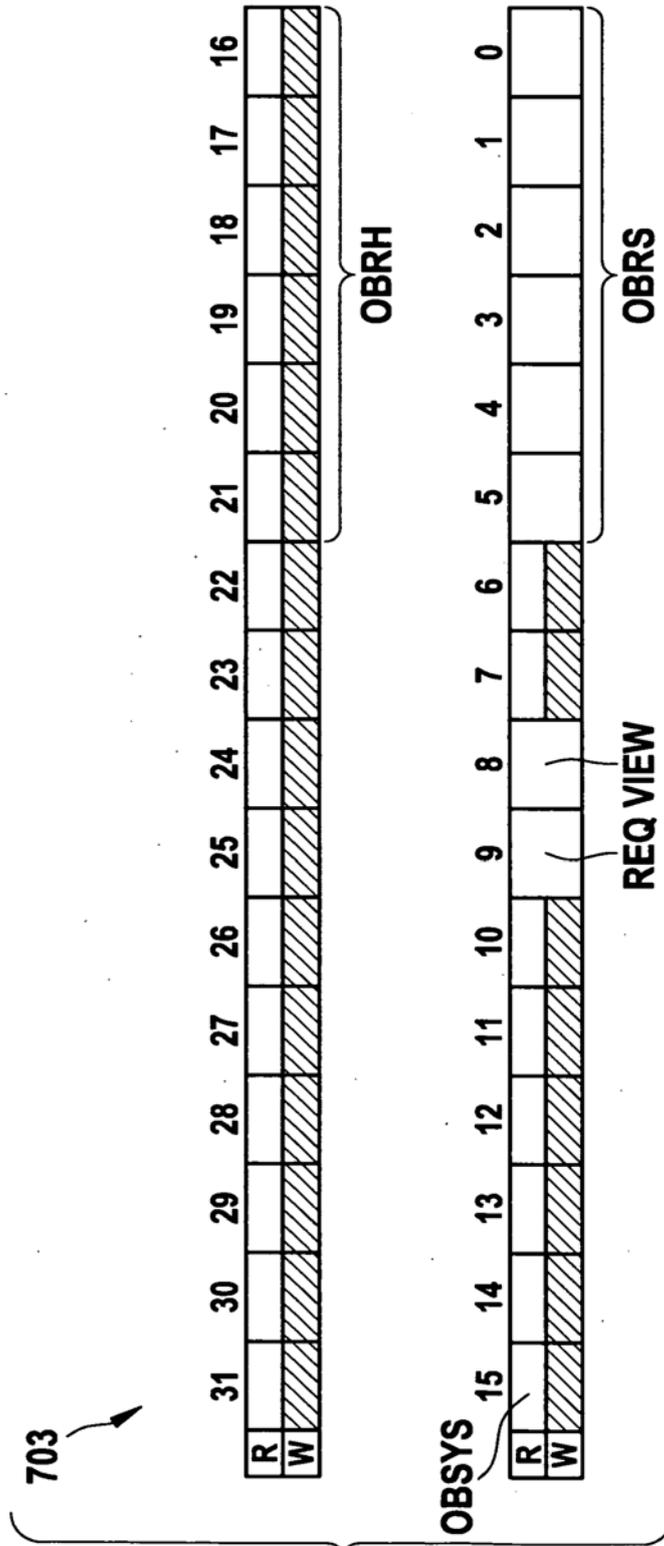


Fig. 8

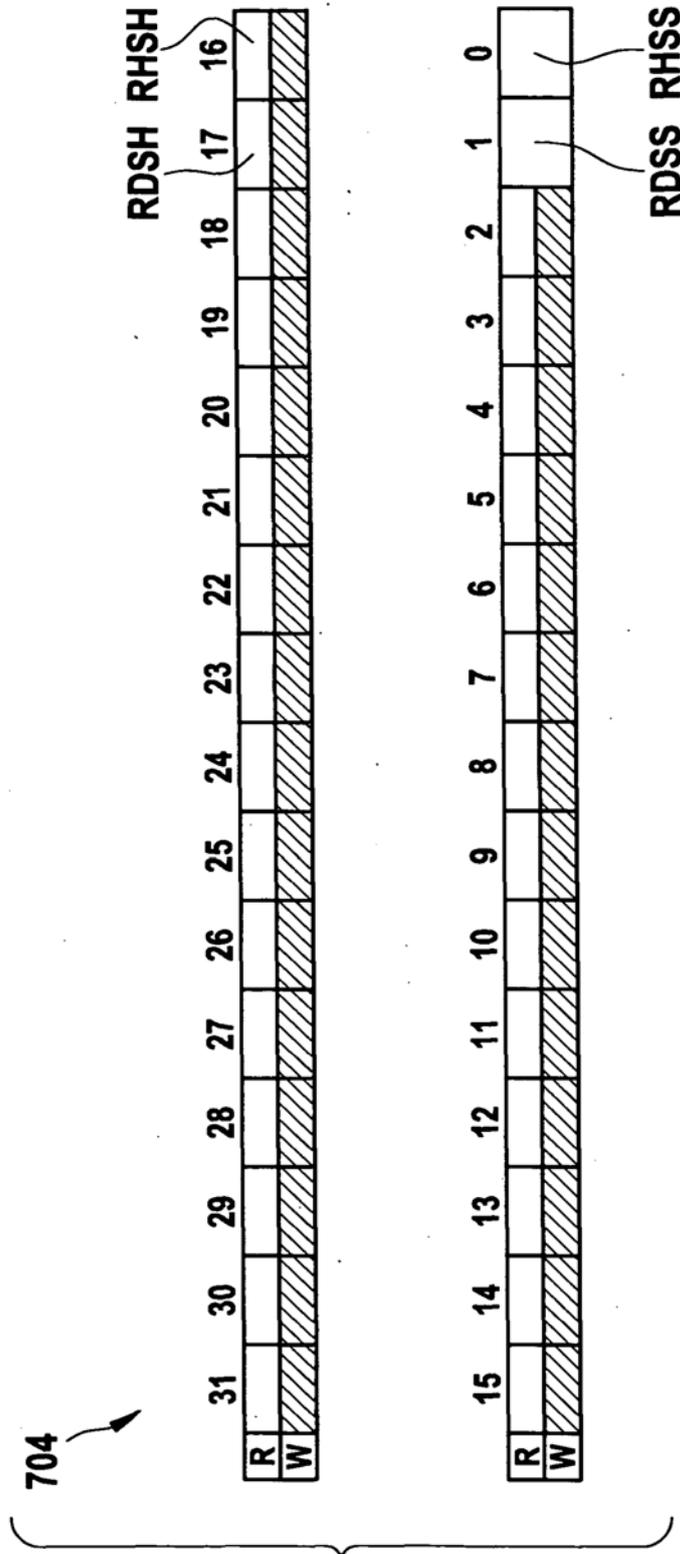
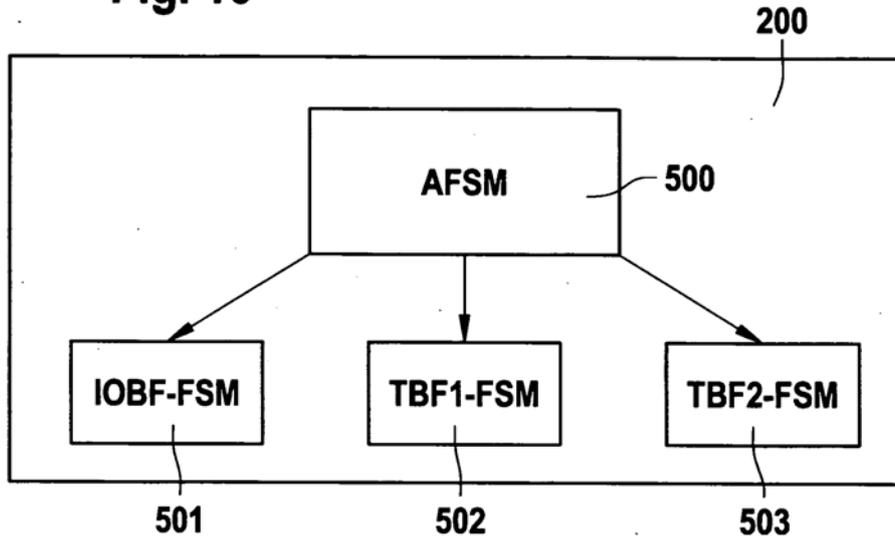
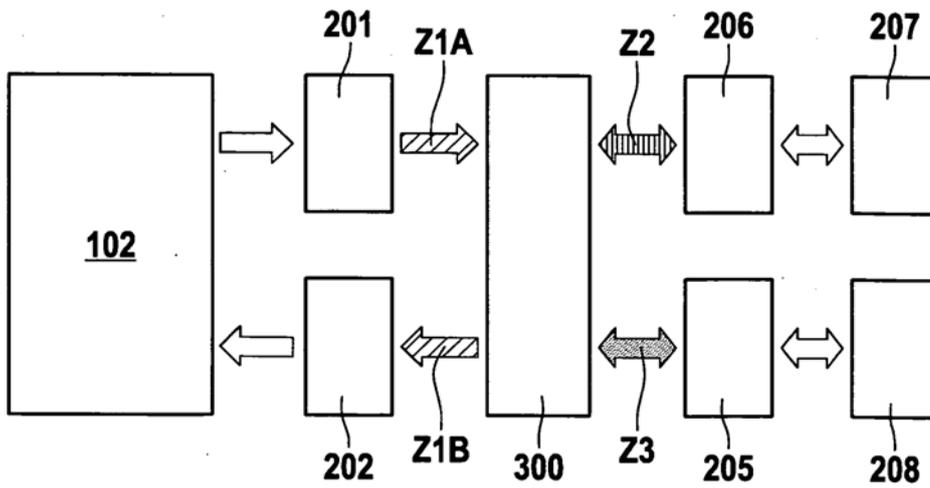


Fig. 9

**Fig. 10**



**Fig. 11**



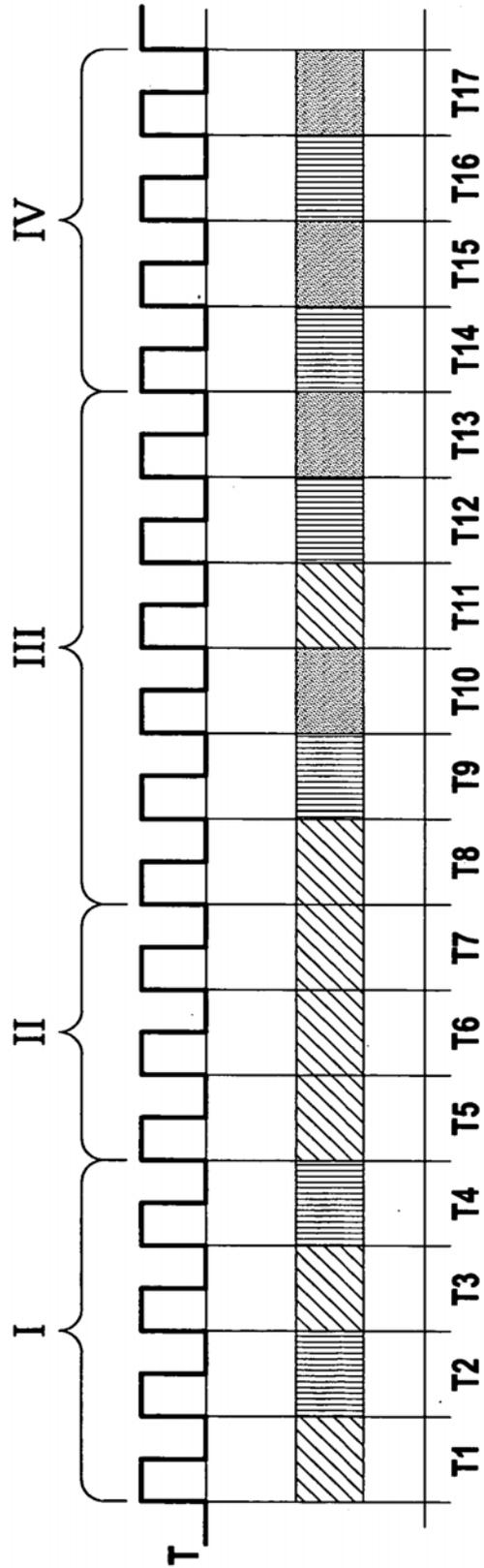


Fig. 12

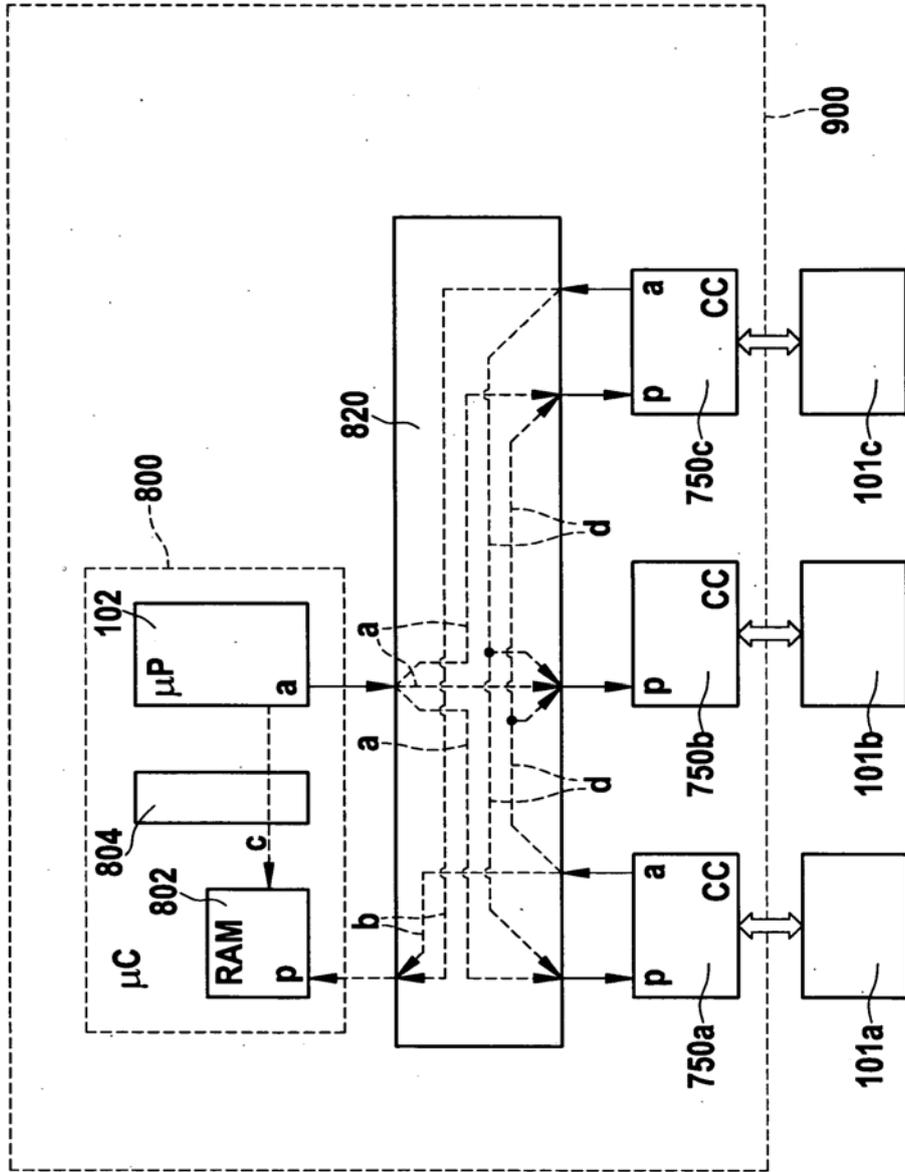


Fig. 13

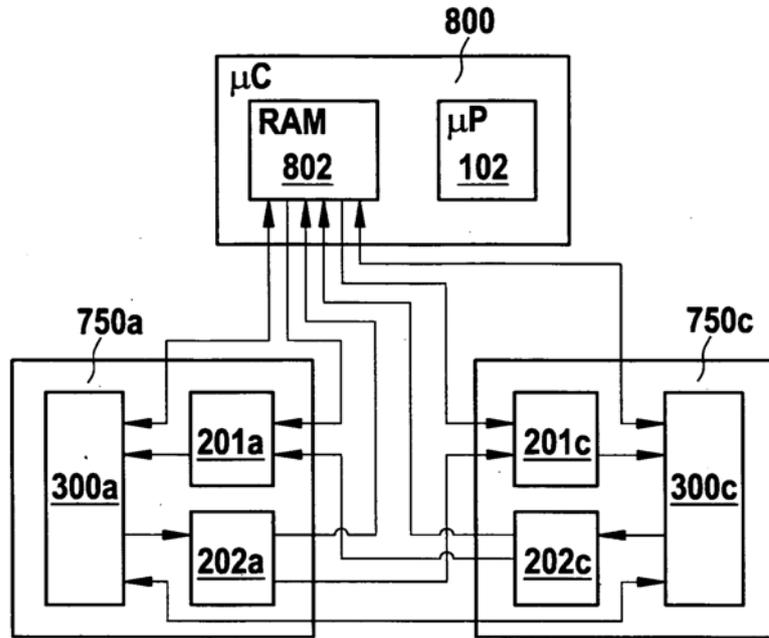


Fig. 14

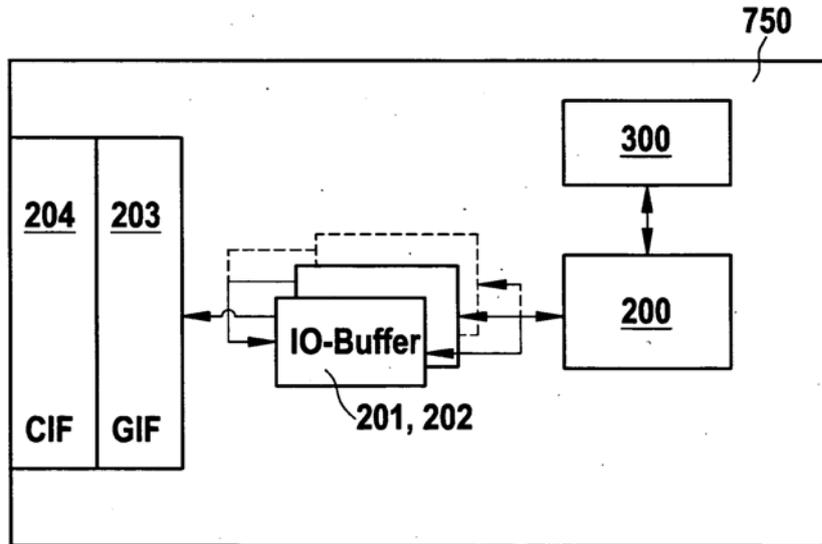


Fig. 15

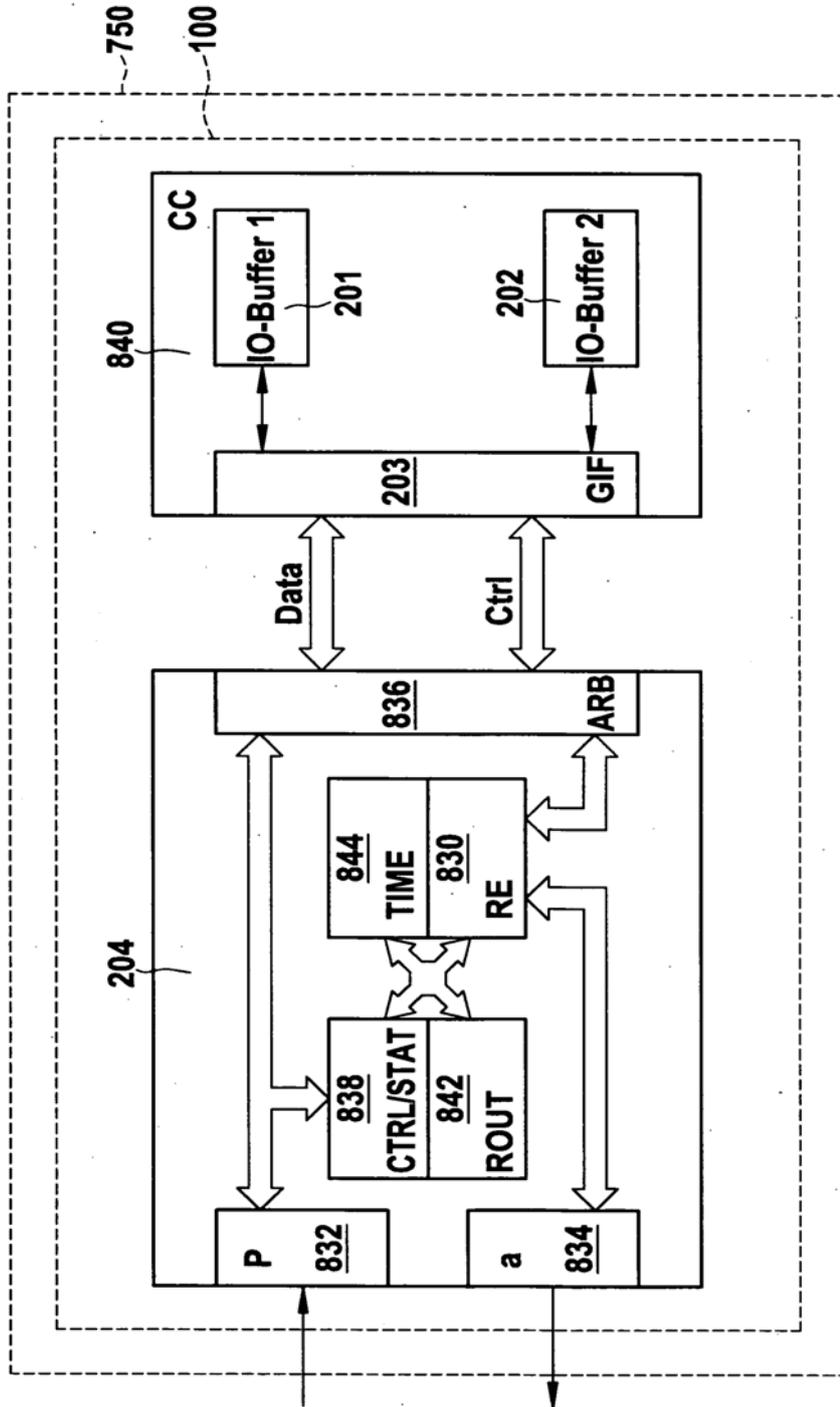


Fig. 16

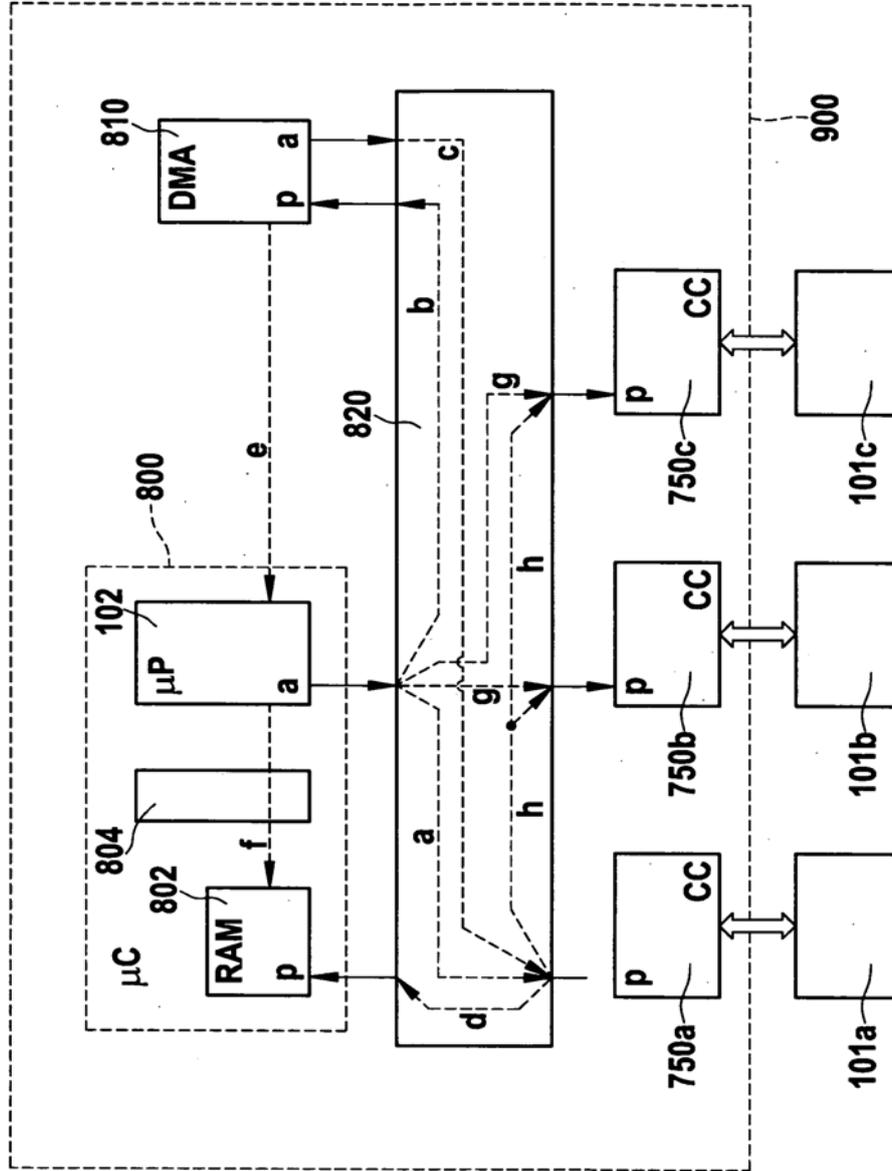


Fig. 17