



19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

11 Número de publicación: **2 266 579**

51 Int. Cl.:
H04L 29/06 (2006.01)
H04L 12/40 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Número de solicitud europea: **02774365 .7**
86 Fecha de presentación : **13.09.2002**
87 Número de publicación de la solicitud: **1430690**
87 Fecha de publicación de la solicitud: **23.06.2004**

54 Título: **Procedimiento para el acceso a una unidad de instrucción para una red de datos.**

30 Prioridad: **26.09.2001 DE 101 47 411**
14.08.2002 DE 102 37 350

45 Fecha de publicación de la mención BOPI:
01.03.2007

45 Fecha de la publicación del folleto de la patente:
01.03.2007

73 Titular/es: **SIEMENS AKTIENGESELLSCHAFT**
Wittelsbacherplatz 2
80333 München, DE

72 Inventor/es: **Brückner, Dieter;**
Götz, Franz-Josef;
Klotz, Dieter;
Krause, Karl-Heinz y
Schimmer, Jürgen

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

ES 2 266 579 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

Procedimiento para el acceso a una unidad de instrucción para una red de datos.

La invención se refiere a un procedimiento para el acceso a una unidad de instrucción para una red de datos, especialmente una Ethernet en tiempo real, a un producto de programa de ordenador así como a un usuario con una unidad de instrucción de este tipo y un sistema de comunicación.

Por un sistema de comunicación sincronizado síncrono con propiedades de equidistancia se entiende un sistema que está constituido por al menos dos usuarios, que están conectados entre sí a través de una red de datos con el objeto del intercambio mutuo de datos o bien de la transmisión mutua de datos. En este caso, el intercambio de datos se lleva a cabo de una manera cíclica en ciclos de comunicación equidistantes, que se predeterminan a través del pulso de reloj de comunicación utilizado por el sistema. Los usuarios son, por ejemplo, aparatos de automatización central, aparatos de programación, aparatos de proyección o aparatos de mando, aparatos periféricos como, por ejemplo, grupos estructurales de entrada/salida, accionamientos, actuadores, sensores, controles programables con memoria (SPS) otras unidades de control, ordenadores o máquinas, que intercambian datos electrónicos con otras máquinas, especialmente procesan datos de otras máquinas. Los usuarios se llaman también nodos de red o nodos. Por unidades de control se entienden a continuación unidades de regulación o unidades de control, pero también, por ejemplo, conmutadores y/o controladores de conmutación. Como redes de datos se utilizan, por ejemplo, sistemas de bus, como por ejemplo bus de campo, Profibus, Ethernet, Ethernet Industrial, FireWire o también sistemas de bus internos de PC (PCI), etc., pero especialmente también Ethernet en tiempo real isócrona.

Las redes de datos posibilitan la comunicación entre varios usuarios a través de la reticulación, es decir, la comunicación de usuarios individuales entre sí. La comunicación significa en este caso la transmisión de datos entre los usuarios. Los datos a transmitir se envían en este caso como telegramas de datos, es decir, que los datos son agrupados en varios paquetes y son enviados en esta forma a través de la red de datos al receptor correspondiente. Por lo tanto, se habla también de paquetes de datos. El concepto transmisión de datos se utiliza en este caso en este documento totalmente como sinónimo de la transmisión mencionada anteriormente de telegramas de datos o paquetes de datos.

En los sistemas de automatización distribuidos, por ejemplo en el campo de la técnica de accionamiento, determinados datos deben entrar en instantes determinados en los usuarios para los que están destinados y deben ser procesados por los receptores. Se habla en este caso de datos críticos en tiempo real bien de tráfico de datos, puesto que una entrada inoportuna de los datos en el lugar de destino conduce a resultados no deseados en el usuario, en oposición a la comunicación de datos no crítica en tiempo real, por ejemplo basada en Internet o bien en Intranet. De acuerdo con el documento IEC 61491, EN61491 SERCOS Interface - Breve Descripción Técnica (http://www.sercos.de/deutch/index_deutsch.htm) se puede garantizar un tráfico de datos crítico en tiempo real con éxito del tipo mencionado en sistemas de

automatización distribuidos.

Los componentes de automatización (por ejemplo, controles, accionamientos, ...) disponen actualmente, en general, de una interfaz con un sistema de comunicación sincronizado cíclicamente. Un plano de ejecución del componente de automatización (ciclo rápido) (por ejemplo, regulación de la posición en un control, regulación del par motor de un accionamiento) está sincronizado al ciclo de comunicación. De esta manera, se establece el ciclo de la comunicación. Otros algoritmos de bajo rendimiento (ciclo lento) (por ejemplo, regulaciones de la temperatura) de los componentes de automatización se pueden comunicar de la misma manera sólo a través de este pulso de reloj de comunicación con otros componentes (por ejemplo, conmutadores binarios para ventiladores, bombas, ...), aunque sería suficiente un ciclo más lento. A través de la utilización de un solo pulso de reloj de comunicación para la transmisión de todas las informaciones en el sistema se plantean altos requerimientos a la anchura de banda del trayecto de transmisión.

Durante el funcionamiento de la unidad de instrucción para el accionamiento en la red de datos (interfaz de comando) en un sistema multi-maestro, varias aplicaciones pueden acceder al mismo tiempo o de una manera consecutiva, pero en cualquier caso coordinada, a la unidad de instrucción. Una coordinación de las aplicaciones individuales a la interfaz de comando es necesaria en este caso para garantizar la transferencia y el procesamiento de las instrucciones en la interfaz de comando. Para la implementación de las aplicaciones pueden estar previstos uno o varios procesadores (maestros) del usuario.

La coordinación ha sido realizada hasta ahora en el plano del software (circuito de excitación), siendo bloqueadas las aplicaciones a través de bloqueos de interrupción durante el periodo de tiempo del procesamiento. En el plano del hardware, a través de un bloqueo del bus se podría impedir un acceso adicional a la interfaz de comando. De esta manera, se detiene igualmente el software durante el periodo de tiempo del procesamiento. Inconveniente:

- Aunque una aplicación no quiera realizar ningún acceso a la interfaz de comando, se interrumpe el procesamiento del software en virtud de los mecanismos de bloqueo del bus.
- A través del bloqueo de la interrupción, durante el periodo de tiempo del procesamiento de la instrucción no se puede procesar inmediatamente ya un acontecimiento de interrupción producido, sino que la rutina de interrupción es ejecutada posteriormente. El tiempo de bloqueo de interrupción más largo determina el tiempo de latencia de la interrupción del sistema.

Por lo tanto, la invención tiene el cometido de crear un procedimiento mejorado para el acceso a una unidad de instrucción para una red de datos. La invención tiene, además, el cometido de crear un programa de ordenador mejorado para el acceso de una aplicación a una unidad de instrucción de este tipo así como un usuario para un sistema de comunicación.

Los cometidos en los que se basa la invención se solucionan en cada caso con las características de

las reivindicaciones de patente independientes correspondientes. Las formas de realización preferidas de la invención se indican en las reivindicaciones dependientes de la patente.

De acuerdo con la invención, la transferencia de la instrucción a la unidad de instrucción no se realiza directa, sino indirectamente, siendo transmitido solamente un indicador a la zona de direcciones de la estructura de instrucciones en la memoria del usuario en el registro de entrada de la unidad de instrucción. En este caso es especialmente ventajoso que la transferencia de la instrucción a la unidad de instrucción se pueda realizar como acceso de escritura "atómico" al registro de entrada, que solamente necesita, por ejemplo, un pulso de reloj del bus.

De acuerdo con una forma de realización preferida de la invención, la unidad de instrucción sirve para la realización de determinadas operaciones básicas a través de la red de datos. La unidad de instrucción se utiliza por las diferentes aplicaciones del usuario a través de una interfaz común. En este caso, todas las aplicaciones son tratadas como equivalentes. Una unidad de arbitraje regula el acceso a un bus de datos interno del usuario para el acceso a las aplicaciones a la unidad de instrucción.

Es especialmente ventajoso que en virtud del acceso de escritura "atómico" al registro de entrada de la unidad de instrucción no se requieren mecanismos de bloqueo del bus o bloqueos de interrupción. De acuerdo con la invención, se deposita en primer lugar una estructura de instrucción en la memoria del usuario, especialmente en la memoria de comunicación, antes de que esta estructura de instrucción sea escrita a través de la escritura de un puntero en la zona de la dirección de la estructura de la instrucción en la memoria en el registro de entrada. Tanto el procesador como también la unidad de instrucción pueden acceder a la memoria de comunicación.

La unidad de instrucción accede entonces a la estructura de la instrucción y la procesa. Esto tiene, en comparación con la transferencia directa de los parámetros de la instrucción a la unidad de instrucción, la ventaja de que se pueden procesar estructuras de instrucciones de más de 32 bits. Además, tampoco está limitada la duración de la realización para la ejecución de una estructura de la instrucción y puede suministrar de retorno mucha información diferente.

Otra ventaja es que se pueden emitir instrucciones de varias aplicaciones, que no tienen que estar necesariamente coordinadas entre sí. Además, se pueden escribir también instrucciones iguales directamente unas detrás de otras en el registro de entrada, sin tener que esperar a la realización de cada instrucción individual.

De acuerdo con una forma de realización preferida de la invención, se lleva a cabo una confirmación de instrucciones recibidas a través del registro de entrada por medio de la unidad de instrucción, porque la unidad de instrucción registra la confirmación en un campo de confirmación en la estructura de la instrucción en la memoria del usuario.

De acuerdo con otra forma de realización preferida de la invención, no se plantean requerimientos de ningún tipo en el bus de datos con respecto al bloqueo, es decir, que el bus no es bloqueado nunca más allá de una operación de lectura o de escritura individual. Además, la cantidad de información transmitida entre las aplicaciones y la unidad de instrucción no

está limitada hacia arriba. No es necesaria una coordinación, como por ejemplo bloqueos de interrupción entre las aplicaciones que llaman.

De acuerdo con otra forma de realización preferida de la invención, se transfieren todas las instrucciones depositadas por las aplicaciones en un ciclo de escritura al registro de entrada (interfaz de comando), para evitar tiempos de bloqueo del bus. A la interfaz de comando solamente se transfiere un puntero (dirección) a una zona de la memoria, en la que se encuentra la estructura de la instrucción depositada por la aplicación con anterioridad en la memoria. De esta manera se evitan accesos de escritura múltiples a la interfaz de comando, que son necesarias en otro caso con instrucciones con varios operandos.

A partir de la zona de direcciones en la memoria se leen entonces los datos de instrucciones desde la estructura de instrucciones definida por la interfaz de comando, se interpreta y se transfiere a la unidad de ejecución correspondiente para el procesamiento.

Los usuarios del bus activos de bus de datos, es decir, las aplicaciones activas, pueden realizar accesos de escritura independientes entre sí a la interfaz de comando. De esta manera, se soportan también las tareas múltiples a través de accesos (mixtos) no ordenados de las diferentes aplicaciones sobre la interfaz de comando. Para no bloquear mutuamente las tareas individuales de una aplicación a través de bloqueos de interrupción, se implementan en la interfaz de comando de una manera preferida unos mecanismos, que permiten un acceso libre opcional de todas las aplicaciones en diferente secuencia y en diferentes instantes a la Interfaz.

De una manera preferida, las aplicaciones reconocen si han sido recibidos sus punteros a una estructura de instrucciones, que fue escrita en la Interfaz de comando. Si la dirección leída y escrita son idénticas, entonces se acepta el comando en la interfaz de comando. Si son diferentes, entonces pueden ser responsables de ello dos posibilidades:

- El comando no ha sido aceptado porque no había concluido todavía la recepción de un comando anterior.
- El comando ya ha sido recibido, pero entre la escritura y la lectura ha sido recibido otro comando. Esta situación puede resultar tanto con una aplicación (Kernel con multitarea premitiva, cuando ésta no es administrada por el sistema operativo) como también en sistemas con varios usuarios físicos. Por lo tanto, se soportan por el hardware de una manera preferida dos mecanismos, que deben ser utilizados ambos por el software para el reconocimiento unívoco:
 - o relectura de los datos recién leídos (dirección sobre la estructura de instrucciones) desde la interfaz de comando. En este caso, se reconoce si los datos de una aplicación han sido recibidos en la interfaz de comando. Si éste es el caso, entonces se ejecuta la instrucción a través de la interfaz de comando; en caso negativo, entonces se inscriben ya previamente los datos de una segunda aplicación en la

interfaz y no se conmuta todavía desde el hardware a la lista de comandos, de manera que no podría aceptarse un comando de la primera aplicación,

- o confirmación de los datos recibidos. Se realizan entre escritura y relectura de la dirección las siguientes acciones en la interfaz de comando, como por ejemplo:

- recepción de la instrucción a través de la interfaz de comando,
- confirmación de la recepción a través de un campo de "reconocimiento" en la estructura de la instrucción también a través de la interfaz de comando,
- descripción de la interfaz de comando con nuevos datos de direcciones a través de un segundo usuario,

de manera que no coinciden ya los datos escritos y releídos. La recepción de una instrucción solamente puede ser reconocida ahora de una manera en el campo de "reconocimiento" de la estructura de instrucción.

Para excluir una carrera crítica, se coloca de una manera preferida solamente el campo de "reconocimiento", antes de que se pueda escribir de nuevo la interfaz de usuario. La recepción de un comando se termina, en general, con la concatenación de la estructura de la instrucción en una lista de comandos, es decir, que se lleva a cabo de una manera rápida. Una optimización en un tiempo de recepción lo más corto posible y garantizado tiene prioridad para no soportar tiempos de espera innecesarios en el procesamiento del software en el caso de un nuevo acceso de escritura.

De acuerdo con otra forma de realización preferida de la invención se transfiere una instrucción o bien secuencias de instrucciones de las aplicaciones sobre un registro de entrada común utilizado (registro de comandos) de la unidad de instrucción.

En este caso, se deposita en el registro de comando la dirección (puntero) sobre una estructura de instrucción. La estructura propiamente dicha contiene todos los datos necesarios para el procesamiento de la instrucción. Para no bloquear, en el caso de procesamiento largo de la instrucción, el registro de comandos para toda la duración del procesamiento de la instrucción, se concatena la estructura de instrucción transferida en una lista de comandos. La estructura de la instrucción contiene a tal fin un puntero "siguiente", con cuya ayuda se pueden concatenar todavía estructuras no procesadas. El puntero "siguiente" contiene la dirección de la siguiente estructura de instrucción no procesada. A través de esta concatenación se consigue una memorización intermedia de las estructuras de instrucción en la lista de comando. De este modo se consigue un desacoplamiento entre procesamiento de la instrucción y transferencia de la instrucción.

Si se lleva a cabo un acceso de escritura desde uno de los usuarios al registro, entonces se impide la recepción de otros datos en el registro de comandos hasta que la instrucción transferida está concatenada en la lista de comandos. Solamente después de que la interfaz de comandos ha confirmado la recepción de la instrucción a través de la colocación del campo de

"reconocimiento" correspondiente, se acepta un nuevo acceso de escritura al registro de comandos.

Después de que las instrucciones transferidas a través del registro de comandos han sido procesadas, se retornan las estructuras de instrucciones correspondientes al usuario. En este caso, para cada aplicación está presente un registro de retorno separado, mencionado a continuación también como registro de salida, a través del cual se transfieren las estructuras de instrucción. Para no tener que garantizar ninguna recogida progresiva de las estructuras de instrucciones en el registro de retorno a través de los usuarios, se concatenan las estructuras procesadas en listas de retorno separadas, en función de los usuarios. El retorno de cada estructura procesada es conocido por el usuario respectivo. Las estructuras de instrucciones concatenadas son retornadas en este caso a través del registro de retorno.

De acuerdo con una forma de realización preferida de la invención, una aplicación, una aplicación transfiere una estructura de instrucción a través de un acceso de escritura al registro de comandos en la interfaz de comando.

Como consecuencia del proceso de escritura en el registro de comando no se aceptan datos de accesos de escritura siguientes hasta que se concatena la estructura transferida desde la interfaz de comando en la lista común de comandos y se confirma en el campo de "reconocimiento" de la estructura la recepción. Sin embargo, los accesos de lectura al registro de comandos están permitidos en adelante. Para controlar si el valor recién registrado ha sido introducido también en el registro de comandos, se releen de nuevo los datos del registro. Si los datos de escritura y los datos de lectura coinciden, entonces se ejecuta el ciclo de escritura en el registro de comandos.

Sin embargo, si existen diferencias entre los datos escritos y los datos leídos, entonces pueden ser responsables de ello dos posibilidades:

- En el periodo de tiempo entre el ciclo de escritura y el ciclo de lectura de una aplicación 1, la estructura de la instrucción registrada en el registro de comandos ya ha sido recibida en la lista de comandos y la confirmación ha sido colocada en el campo de "reconocimiento" de la estructura. La inscripción de nuevos datos en el registro de comandos ha sido liberada a continuación. Todavía antes de que la aplicación 1 haya realizado su ciclo de lectura, la aplicación 2 podría depositar un acceso de escritura al registro de comandos.
- Antes de la escritura del registro de comandos a través de una aplicación 1 ha sido depositado ya un acceso de escritura por la aplicación 2. De esta manera, se impide la entrada de los datos siguientes de la dirección de la aplicación.

Si aparecen diferencias entre los datos de escritura y los datos de lectura, entonces la aplicación debe analizar el campo de "reconocimiento" de la estructura de la instrucción. Si la confirmación está colocada en el campo de "reconocimiento" de la estructura, entonces la estructura de la instrucción ha sido recibida y ha sido concatenada en la lista de comandos. La recepción no confirmada de la estructura conduce a

un acceso de escritura cíclico al registro de comando (Polling).

Además, es especialmente ventajoso que los procedimientos publicados se pueden aplicar o bien emplearen sistemas de automatización, especialmente en máquinas de envase, prensas, máquinas de inyección de plástico, máquinas textiles, máquinas de imprenta, máquinas herramientas, robots, sistemas de manipulación, máquinas de procesamiento de madera, máquinas de procesamiento de vidrio, máquinas de procesamiento de cerámicas así como productos semiacabados.

Otra ventaja es que la invención se puede emplear tanto para aplicaciones en comunicaciones como también para otras aplicaciones, como por ejemplo interfaces de comando de otros subsistemas inteligentes, especialmente sistemas gráficos.

Por lo demás, se explican en detalle formas de realización preferidas de la invención con referencia a los dibujos. En este caso:

La figura 1 muestra un diagrama de bloques de una forma de realización de un usuario de una red de datos de acuerdo con la invención.

La figura 2 muestra una representación de principio de la interfaz de comando.

La figura 3 muestra un ejemplo de estructuras de instrucciones concatenadas en la memoria del usuario.

La figura 4 muestra un diagrama de flujo de una forma de realización del procedimiento de acuerdo con la invención.

La figura 1 muestra un usuario 100 de una red de datos 102. En la red de datos 102 se puede tratar, por ejemplo, de una Ethernet en tiempo real para aplicaciones en la técnica de automatización. En una red de datos 102 de este tipo se conectan típicamente varios usuarios, que están constituidos, en principio, de la misma forma que el usuario 100. De este modo se crea un sistema de comunicación.

El usuario 100 tiene varias aplicaciones 104, 106, 108, ..., que pueden acceder a un bus de datos 110 del usuario 100. El acceso de las aplicaciones 104, 106, 108, ... al bus de datos se regula a través de un árbitro 112.

Además, el usuario 100 tiene una memoria 114 y una unidad de instrucciones 116. La memoria 114 y la unidad de instrucciones 116 están acopladas de la misma manera en el bus de datos 110.

Cada una de las aplicaciones 104, 106, 108 puede describir a través del bus de datos 110 la memoria 114 con una estructura de instrucciones 118. En la forma de realización de la figura 1, la estructura de instrucciones 118 está constituida por una instrucción 120, que puede ser ejecutada por la unidad de instrucciones, y por un campo de confirmación 122. La estructura de instrucciones 118 es memorizada en una zona de direcciones de la memoria 114, a la que apunta el puntero 124.

La unidad de instrucciones 116 está acoplada tanto con el bus de datos 110 como también con la red de datos 102. La unidad de instrucciones 116 sirve para la realización de diferentes operaciones básicas, que se refieren a la red de datos 102 con relación a las aplicaciones 104, 106, 108, ... La unidad de instrucciones 116 contiene una interfaz para las aplicaciones 104, 106, 108, ... que presente un registro de comandos 126 y varios registros de retorno 128.

El registro de comandos 126 sirve como registro

de entrada para la memorización de punteros 124. Al registro de comandos 126 puede acceder cualquiera de las aplicaciones 104, 106, 108, ... a través del bus de datos 110. En cambio, cada uno de los registros de retorno 128 está asociado a una aplicación determinada. Por ejemplo, el registro de retorno 130 está asociado a la aplicación 104, el registro de retorno 132 está asociado a la aplicación 106 y el registro de retorno 134 está asociado a la aplicación 108, etc.

La unidad de instrucciones 116 contiene, además, una lista de comandos 116, que se designa también, por decirlo así, como pila. En la lista de comandos 136 se encuentran las instrucciones aceptadas y a procesar.

Además, la unidad de instrucción 116 tiene un circuito lógico 138 para el procesamiento de las instrucciones.

En el funcionamiento, una de las aplicaciones del usuario 100, por ejemplo la aplicación 104 accede a través del bus de datos 110 a la memoria 114, con el fin de depositar allí una estructura de instrucciones 118. A continuación, la aplicación 104 accede a través del bus de datos 110 al registro de comandos 126 con un acceso de escritura, con el fin de escribir el puntero 24 sobre la estructura de instrucciones 118 en el registro de comandos 126. La unidad de instrucción 116 recibe entonces la estructura de la instrucción 118 desde la memoria 114 en la lista de comandos 136 y confirma la recepción a través de una entrada correspondiente en el campo de confirmación 122 de la estructura de instrucciones 118.

Después del procesamiento de la estructura de la instrucción 118, la unidad de instrucciones 116 escribe el puntero 124 en el registro de retorno 130, que está asociado a la aplicación 104. La aplicación 104 puede consultar el registro de retorno 130 a través de un acceso de lectura a través del bus de datos 110, para verificar si la estructura de la instrucción 118 ya ha sido procesada.

La figura 2 muestra un desarrollo de la interfaz de ordenador de la figura 1.

El registro de comandos 126 se puede describir en este caso por diferentes aplicaciones 104, 106, 108, ... de forma no coordinada. A partir de las estructuras de las instrucciones 118 recibidas resulta la lista de comandos 136.

Además, a cada uno de los registros de retorno 130, 132, 134, ... está asociada una lista de retorno 140, ..., al registro de retorno 130, que está asociada de nuevo a la aplicación 104. Con la ayuda de la lista de retorno 140 se pueden tamponar la salida de punteros 124.

La figura 3 muestra un ejemplo de realización para una estructura de instrucciones en la memoria 114. En este ejemplo de realización, la estructura de instrucciones 118 tiene un campo 142 para la memorización de una o varias instrucciones 120 (ver la figura 1), un campo 144 para la memorización de una confirmación, que corresponde al campo de confirmación 122 de la figura 1 así como un campo 146 para la memorización de un puntero 148 sobre otra estructura de la instrucción 118, que está constituida, en principio, igual. La otra estructura de la instrucción 118 tiene un puntero 150 sobre otra estructura de la instrucción 118, etc. La última estructura de la instrucción 118 de la cadena no tiene otros punteros, con lo que está identificado el último miembro de la estructura de la instrucción concatenada.

Las estructuras de la instrucción 118 tienen, además, campos 152 para la memorización de parámetros, datos útiles u operandos para la realización de la instrucción 120 respectiva.

Si se transfiere el puntero 124 en esta forma de realización al registro de comandos 126 (ver la figura 1) y se confirma la transferencia desde la unidad de instrucciones 116, entonces se procesa toda la cadena de estructuras de instrucciones 118 en adelante desde la unidad de instrucciones 116. Con un único acceso de escritura "atómica" al registro de comandos 126, por ejemplo, dentro de un pulso de reloj del bus, se puede iniciar el procesamiento de una secuencia de instrucciones a través de la unidad de instrucción 116 por una aplicación.

La figura 4 muestra un diagrama de flujo de un procedimiento para el funcionamiento del sistema de la figura 1. El procedimiento se divide en un proceso de software 154 y un proceso de hardware 156. Después del comienzo del proceso de software 154 en la etapa 200, la aplicación, a la que pertenece el proceso de software 154, ejecuta un acceso de escritura al registro de comando en un ciclo de bus. Esto se lleva a cabo en la etapa 202. En el ciclo de escritura, la aplicación 104 transfiere el puntero 124 a una estructura de instrucción 118 o bien a una concatenación de estructuras de instrucciones 118. De esta manera se inicia el proceso de hardware 156.

En la etapa 204 del proceso de hardware 156 se verifica si el registro de comandos se puede describir. Si éste no es el caso, el ciclo termina en la etapa 206. En este caso, la aplicación debe comenzar de nuevo el proceso de software 154 con la etapa 200.

No obstante, cuando el registro de comandos se puede describir, entonces se registra el puntero 124 transferido por la aplicación 104 en el registro de co-

mandos, lo que se lleva a cabo en la etapa 206. A continuación, se bloquea la descripción del registro de comandos en la etapa 208, para que otras aplicaciones no puedan sobrescribir el puntero que se encuentra en el registro.

En la etapa 210 se recibe la estructura de la instrucción o bien la estructura de la instrucción concatenada, es decir, que se lleva a cabo una concatenación de las instrucciones a procesar en la lista de comandos y una confirmación de la recepción en el campo de confirmación de la estructura de la instrucción.

A continuación se libera de nuevo la descripción del registro de comandos en la etapa 212 y se termina la ejecución del proceso de hardware 156 con la etapa 206.

Después del ciclo de escritura en la etapa 202 se lleva a cabo en el proceso de software 154 un ciclo de lectura en la etapa 214 en el registro de comandos en la etapa 214. En la etapa 216 se verifica si los datos escritos anteriormente en la etapa 202 en el registro de comandos, es decir, el puntero 124, están todavía en el registro de comandos.

Si éste es el caso, entonces esto significa que se lleva a cabo una recepción de la estructura de la instrucción en la etapa 210, de manera que el proceso de software termina con la etapa 218. En caso contrario, esto puede significar que se lleva a cabo una recepción y ya otra aplicación ha descrito el registro de comandos con otro puntero, o que no se realiza una recepción por parte del proceso de hardware 156. En este caso, se verifica en la etapa 220 el campo de confirmación en la estructura de la instrucción. Cuando allí ha sido inscrita una confirmación, entonces se puede terminar el proceso de software 154 de nuevo con la etapa 218. Si éste no es el caso, entonces el control del desarrollo pasa a la etapa 202.

REIVINDICACIONES

1. Procedimiento para el acceso a una unidad de instrucción (116) para una red de datos (102) con las siguientes etapas:

- funcionamiento de varias aplicaciones (104, 106, 108, ...) en un usuario (100) de la red de datos (102), en el que las aplicaciones acceden a un bus de datos (110) del usuario (100),
- escritura de al menos una estructura de la instrucción (118) en una zona de direcciones de una memoria del usuario (100) a través de una primera de las aplicaciones a través del bus de datos (110),

caracterizado por las otras etapas:

- escritura de un puntero (124) sobre la zona de direcciones en un registro de entrada (126) de la unidad de instrucciones (116) a través de la primera aplicación por medio del bus de datos (110),
- acceso de la unidad de instrucciones (116) a la zona de direcciones a través del bus de datos (110) y procesamiento de la estructura de instrucciones (118),
- escritura del puntero (124) en un registro de salida (136, 130) asociado a la primera aplicación después del procesamiento de la estructura de instrucciones (118) a través del usuario (100).

2. Procedimiento de acuerdo con la reivindicación 1, en el que el acceso al bus de datos (110) se regula a través de una unidad de arbitraje (112), en el que se permite el acceso para un número predeterminado de pulsos de reloj del bus y el número predeterminado de pulsos de reloj del bus es suficiente para la escritura del puntero (124) en el registro de entrada (126).

3. Procedimiento de acuerdo con la reivindicación 1 ó 2, en el que la estructura de la instrucción (118) contiene un campo de confirmación, con las siguientes etapas adicionales:

- bloqueo del registro de entrada (126) después de la escritura del puntero (124) a través del usuario (100),
- escritura de una confirmación en el campo de confirmación (122) a través del usuario (100),
- liberación del registro de entrada (126) a través del usuario (100) después de la confirmación.

4. Procedimiento de acuerdo con una de las reivindicaciones anteriores 1 a 3, con las siguientes etapas adicionales:

- lectura del registro de entrada (126) a través de la primera aplicación después de la escritura del puntero (124),
- verificación a través de la primera aplicación de si el registro de entrada (126) contiene el puntero (124),
- si éste no es el caso, verificación de si ha sido memorizada una confirmación en el

campo de confirmación (122).

5. Procedimiento de acuerdo con una de las reivindicaciones anteriores 1 a 4, en el que la estructura de la instrucción (118) contiene instrucciones ejecutable y datos útiles.

6. Procedimiento de acuerdo con una de las reivindicaciones anteriores 1 a 5, en el que a través de la primera aplicación se escriben varias estructuras de la instrucción concatenadas entre sí en la memoria y el puntero (124) apunta a la zona de direcciones de la primera estructura de la instrucción (118) de la cadena.

7. Producto de programa de ordenador para una aplicación (104, 106, 108, ...) de un usuario (100) de una red de datos (102), en el que la aplicación puede acceder a un bus de datos (110) para varias aplicaciones del usuario (100), y el usuario (100) presenta un registro de entrada (126) y un registro de salida (136, 130) asociado a la aplicación, con medios de programa para la realización de las siguientes etapas:

- escritura de una estructura de la instrucción (118) en una zona de direcciones de una memoria del usuario (100) a través del bus de datos (110),

caracterizado por las otras etapas:

- escritura de un puntero (124) en la zona de direcciones en el registro de entrada (126) de una unidad de instrucciones (116) a través del bus de datos (110),
- lectura del registro de entrada (126), para verificar que la estructura de la instrucción (118) ha sido confirmada a través de la unidad de instrucciones (116).

8. Producto de programa de ordenador de acuerdo con la reivindicación 7, en el que se lleva a cabo una verificación de un campo de confirmación (122) en la estructura de la instrucción (118), cuando durante la lectura del registro de entrada (126), el puntero (124) no está ya en el registro de entrada (126).

9. Producto de programa de ordenador de acuerdo con la reivindicación 7 u 8, en el que varias estructuras de la instrucción concatenadas entre sí son escritas en la memoria del usuario (100), y el puntero (124) apunta a la zona de direcciones de la primera estructura de la instrucción (118) de la cadena.

10. Producto de programa de ordenador de acuerdo con la reivindicación 7, 8 ó 9, en el que se lee el registro de salida (136, 130) asociado a la aplicación, con el fin de verificar si la estructura de la instrucción (118) ha sido procesada por la unidad de instrucciones (116).

11. Aparato de usuario (100) de una red de datos (102) con

- una unidad de instrucciones (116) para el acceso a la red de datos (102),
- medios para el accionamiento de varias aplicaciones (104, 106, 108, ...), en el que las aplicaciones pueden acceder a un bus de datos (110) del usuario (100) y
- medios para la escritura al menos de una estructura de la instrucción (118) en una zona de direcciones de una memoria del usuario (100) a través de una primera de

las aplicaciones a través del bus de datos (110),

caracterizado por

- medios para la escritura de un puntero (124) sobre la zona de direcciones en un registro de entrada (126) de la unidad de instrucciones (116) a través de la primera aplicación sobre el bus de datos (110), 5
- medios para el acceso de la unidad de instrucciones (116) a la zona de direcciones a través del bus de datos (110) y procesamiento de la estructura de la instrucción (118) y 10
- medios para la escritura del puntero (124) en un primer registro de salida (136, 130) que está asociado a la primera aplicación después del procesamiento de la estructura de la instrucción (118) a través del usuario (100). 15 20

12. Aparato de usuario (100) de acuerdo con la reivindicación 11 con una unidad de arbitraje (112) para la regulación del acceso al bus de datos (110), en el que se permite un acceso durante un número predeterminado de pulsos de reloj del bus para la escritura del puntero (124) en el registro de entrada (126). 25

13. Aparato de usuario (100) de acuerdo con la reivindicación 11 ó 12, en el que un campo de confirmación (122) está contenido en la estructura de la instrucción (118) y con 30

- medios para el bloqueo del registro de entrada (126) después de la escritura del pun- 35

tero (124) a través del usuario (100),

- medios para la escritura de una confirmación en el campo de confirmación (122) a través del usuario (100),
- medios para la liberación del registro de entrada (126) a través del usuario (100) después de la confirmación.

14. Aparato de usuario (100) de acuerdo con la reivindicación 11, 12 ó 13 con

- medios para la lectura del registro de entrada (126) a través de la primera aplicación después de la escritura del puntero (124),
- medios para la verificación a través de la primera aplicación de si el registro de entrada (126) contiene el puntero (124) y, si éste no es el caso, para la verificación de si ha sido memorizada una confirmación en el campo de confirmación (122). 15 20

15. Aparato de usuario (100) de acuerdo con una de las reivindicaciones anteriores 11 a 14, en el que la estructura de la instrucción (118) contiene instrucciones ejecutables y datos útiles. 25

16. Aparato de usuario (100) de acuerdo con una de las reivindicaciones anteriores 11 a 15, con varias estructuras de instrucciones concatenadas entre sí en la memoria, en el que el puntero (124) apunta a la zona de direcciones de la primera estructura de la instrucción (118) de la cadena. 30

17. Sistema de comunicación con una red de datos (102) y varios aparatos de usuarios de acuerdo con una de las reivindicaciones anteriores 11 a 16. 35

40

45

50

55

60

65





