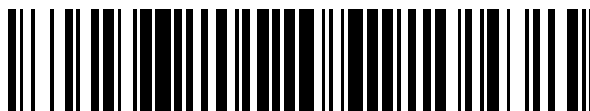


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 523 878**

51 Int. Cl.:

G06F 9/312 (2006.01)

G06F 13/38 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **08.11.2010** **E 10776345 (0)**

97 Fecha y número de publicación de la concesión europea: **22.10.2014** **EP 2430524**

54 Título: **Instrucciones de almacenamiento/almacenamiento de bloque para la comunicación con adaptadores**

30 Prioridad:

23.06.2010 US 821194

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
02.12.2014

73 Titular/es:

**INTERNATIONAL BUSINESS MACHINES
CORPORATION (100.0%)
New Orchard Road
Armonk, NY 10504, US**

72 Inventor/es:

**GREINER, DAN;
CRADDOCK, DAVID;
GREGG, THOMAS y
FARRELL, MARK**

74 Agente/Representante:

DE ELZABURU MÁRQUEZ, Alberto

ES 2 523 878 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

Instrucciones de almacenamiento/almacenamiento de bloque para la comunicación con adaptadores

Antecedentes

Esta invención se refiere, en general, al procesamiento de entradas/salidas de un entorno informático y, en particular, a facilitar la comunicación con los adaptadores del entorno informático.

Un entorno informático puede incluir uno o más tipos de dispositivos de entrada/salida, incluyendo diversos tipos de adaptadores. Un tipo de adaptador es un adaptador de tipo interconexión de componentes periféricos (Peripheral Component Interconnect, PCI) o de tipo interconexión de componentes periféricos express (Peripheral Component Interconnect Express, PCIe). Este adaptador incluye uno o más espacios de direcciones usados en la comunicación de datos entre el adaptador y el sistema al cual está conectado el adaptador. La especificación PCI está disponible en la World Wide Web en www.pcisig.com/home.

La patente US N° 6.704.831, presentada el 9 de Marzo de 2004, Avery, "Method and Apparatus for Converting Address Information Between PCI Bus Protocol and a Message-Passing Queue-Oriented Bus Protocol", describe que las operaciones de carga/almacenamiento PCI y las operaciones DMA se implementan a través de pares de colas de trabajo en una arquitectura de bus orientada a colas, con intercambio de mensajes. El espacio de direcciones PCI se divide en segmentos y cada segmento, a su vez, se divide en regiones. A cada segmento se le asigna una cola de trabajo independiente. Una primera parte de una dirección PCI es comparada con los rangos de direcciones representados por los segmentos y usados para seleccionar un segmento de memoria y su cola de trabajo correspondiente. Una entrada en la cola de trabajo contiene una segunda parte de la dirección PCI que especifica una región dentro del segmento seleccionado que es asignada a un dispositivo PCI específico. En una realización, las operaciones de carga/almacenamiento PIO se implementan seleccionando una cola de trabajo asignada a las operaciones PIO y creando una entrada de cola de trabajo con la dirección PCI de un registro en un dispositivo PCI y un puntero a los datos PIO. La entrada en la cola de trabajo es enviada a un puente PCI donde la dirección PCI es extraída y usada para programar el registro de dispositivo apropiado con los datos usando el puntero de datos. Las transferencias DMA son implementadas también seleccionando una cola de trabajo por medio de una comparación de una parte de la dirección PCI generada por el dispositivo PCI con una tabla de rangos de direcciones y seleccionando una cola de trabajo que proporciona servicio al rango de direcciones. Se crea una entrada de cola de trabajo con el resto de la dirección PCI y un puntero a los datos DMA. Se usa una operación RDMA para realizar la transferencia DMA. Los datos de región y página son usados en conexión con una tabla de protección de traducción en el adaptador de canal de servidor para acceder a la memoria física y realizar la transferencia DMA.

La patente US N° 7.613.847, presentada el 3 de Noviembre de 2009, Kjos et al, "Partially Virtualizing an E/S Device for Use by Virtual Machines", describe un sistema de ordenador que comprende un ordenador físico y un monitor de máquina virtual ejecutable en el ordenador físico y configurado para crear una emulación de al menos un sistema operativo huésped adaptado para controlar el ordenador físico. El sistema de ordenador comprende además un servidor ejecutable en el ordenador físico que gestiona los recursos físicos acoplados al ordenador físico en nombre del monitor de máquina virtual y el al menos un sistema operativo huésped. El servidor está adaptado para virtualizar un espacio de direcciones de a configuración de interconexión de componentes periféricos (PCI) de manera que el al menos un sistema operativo huésped controla los dispositivos de entrada/salida (E/S) PCI directamente y en ausencia de emulación E/S.

En algunos sistemas, tales como los descritos en la publicación de patente internacional WO 02/41157 A2, una parte de un espacio de direcciones de la unidad central de procesamiento (CPU) acoplada al adaptador es asignada a un espacio de direcciones del adaptador permitiendo que las instrucciones de CPU que acceden al almacenamiento manipulen directamente los datos en el espacio de direcciones del adaptador.

Breve resumen

Según un aspecto de la presente invención, se proporciona una capacidad para facilitar la comunicación con los adaptadores, tales como adaptadores PCI o PCIe. Se proporcionan y usan para la comunicación instrucciones de control diseñadas específicamente para la comunicación de datos desde y hacia los adaptadores.

La provisión de un producto de programa de ordenador para ejecutar una instrucción de almacenamiento para almacenar datos en un adaptador supera los inconvenientes de la técnica anterior y proporciona una serie de ventajas. El producto de programa de ordenador incluye un medio de almacenamiento legible por ordenador, legible por un circuito de procesamiento, e instrucciones de almacenamiento para su ejecución por el circuito de procesamiento para realizar un procedimiento. El procedimiento incluye, por ejemplo, obtener una instrucción

máquina para su ejecución, en el que la instrucción máquina está definida para su ejecución por un ordenador según una arquitectura de ordenador, en el que la instrucción máquina incluye, por ejemplo, un campo de código de operación que identifica una instrucción de almacenamiento en adaptador; un primer campo que identifica una primera ubicación que incluye los datos a ser almacenados en un adaptador; un segundo campo que identifica una segunda ubicación, cuyos contenidos incluyen un descriptor de función que identifica el adaptador, una designación de un espacio de direcciones dentro del adaptador en el que deben almacenarse los datos, y un desplazamiento dentro del espacio de direcciones; y ejecutar la instrucción máquina, en el que la ejecución incluye usar el descriptor de función para obtener una entrada de la tabla de funciones asociada con el adaptador; obtener una dirección de datos del adaptador usando al menos uno de entre la información en la entrada de la tabla de funciones y el desplazamiento; y almacenar datos desde la primera ubicación en una ubicación específica en el espacio de direcciones identificado por la designación del espacio de direcciones, la ubicación específica identificada por la dirección de datos del adaptador.

Además, se proporciona un producto de programa de ordenador para ejecutar una instrucción de almacenamiento de bloque para almacenar datos en un adaptador. El producto de programa de ordenador incluye un medio de almacenamiento legible por ordenador, legible por un circuito de procesamiento, e instrucciones de almacenamiento para su ejecución por el circuito de procesamiento para realizar un procedimiento. El procedimiento incluye, por ejemplo, obtener una instrucción máquina para su ejecución, en el que la instrucción máquina está definida para su ejecución por un ordenador según una arquitectura de ordenador, en el que la instrucción máquina incluye, por ejemplo, un campo de código de operación que identifica una instrucción de almacenamiento de bloque en el adaptador; en el que un primer campo identifica una primera ubicación, cuyos contenidos incluyen un descriptor de función que identifica un adaptador, y una designación de un espacio de direcciones dentro del adaptador en el que deben almacenarse los datos; un segundo campo que identifica una segunda ubicación que incluye un desplazamiento dentro del espacio de direcciones; un tercer campo que identifica una tercera ubicación que incluye una dirección en la memoria que incluye los datos a ser almacenados en el adaptador; y ejecutar la instrucción máquina, en el que la ejecución incluye usar el descriptor de función para obtener una entrada de la tabla de funciones asociada con el adaptador; obtener una dirección de datos del adaptador usando la información en la entrada de la tabla de funciones y el desplazamiento; y almacenar los datos obtenidos desde la memoria en la dirección en el tercer campo en una ubicación específica en el espacio de direcciones identificado por la designación del espacio de direcciones, la ubicación específica identificada por la dirección de datos del adaptador.

También se describen y reivindican en la presente memoria procedimientos y sistemas relacionados con uno o más aspectos de la presente invención.

Las técnicas de la presente invención permiten conseguir características y ventajas adicionales. Otras realizaciones y aspectos de la invención se describen en detalle en la presente memoria, y se consideran parte de la invención reivindicada.

Breve descripción de las diversas vistas de los dibujos

Uno o más aspectos de la presente invención se resaltan particularmente y se reivindican marcadamente como ejemplos en las reivindicaciones al final de la presente especificación. Lo indicado anteriormente y otros objetos, características y ventajas de la invención son evidentes a partir de la descripción detallada siguiente, tomada en conjunción con los dibujos adjuntos, en los que:

La Fig. 1A representa una realización de un entorno informático para incorporar y usar uno o más aspectos de la presente invención;

La Fig. 1B representa una realización de una entrada de la tabla de dispositivos situada en el concentrador de E/S de la Fig. 1A y usada según un aspecto de la presente invención;

La Fig. 1C representa otra realización de un entorno informático para incorporar y usar uno o más aspectos de la presente invención;

La Fig. 2 representa un ejemplo de espacios de direcciones de una función de adaptador, según un aspecto de la presente invención;

La Fig. 3A representa un ejemplo de una entrada de la tabla de funciones usada según un aspecto de la presente invención;

La Fig. 3B representa una realización de un descriptor de función usado según un aspecto de la presente invención;

La Fig. 4A representa una realización de una instrucción de carga PCI usada según un aspecto de la presente invención;

La Fig. 4B representa una realización de un campo usado por la instrucción de carga PCI de la Fig. 4A, según un aspecto de la presente invención;

5 La Fig. 4C representa una realización de otro campo usado por la instrucción de carga PCI de la Fig. 4A, según un aspecto de la presente invención;

Las Figs. 5A-5B representan una realización de la lógica para realizar una operación de carga PCI, según un aspecto de la presente invención;

10 La Fig. 6A representa una realización de una instrucción de almacenamiento PCI usada según un aspecto de la presente invención;

La Fig. 6B representa una realización de un campo usado por la instrucción de almacenamiento PCI de la Fig. 6A, según un aspecto de la presente invención;

La Fig. 6C representa una realización de otro campo usado por la instrucción de almacenamiento PCI de la Fig. 6A, según un aspecto de la presente invención;

15 Las Figs. 7A-7B representan una realización de la lógica para realizar una operación de almacenamiento PCI, según un aspecto de la presente invención;

La Fig. 8A representa una realización de una instrucción de almacenamiento de bloque PCI usada según un aspecto de la presente invención;

20 La Fig. 8B representa una realización de un campo usado por la instrucción de almacenamiento de bloque PCI de la Fig. 8A, según un aspecto de la presente invención;

La Fig. 8C representa una realización de otro campo usado por la instrucción de almacenamiento de bloque PCI de la Fig. 8A, según un aspecto de la presente invención;

La Fig. 8D representa una realización de todavía otro campo usado por la instrucción de almacenamiento de bloque PCI de la Fig. 8A, según un aspecto de la presente invención;

25 Las Figs. 9A-9B representan una realización de la lógica para realizar una operación de almacenamiento de bloque PCI, según un aspecto de la presente invención;

La Fig. 10 representa una realización de un producto de programa de ordenador que incorpora uno o más aspectos de la presente invención;

30 La Fig. 11 representa una realización de un sistema de ordenador servidor para incorporar y usar uno o más aspectos de la presente invención;

La Fig. 12 representa un ejemplo adicional de un sistema de ordenador para incorporar y usar uno o más aspectos de la presente invención;

La Fig. 13 representa otro ejemplo de un sistema de ordenador que comprende una red de ordenadores para incorporar y usar uno o más aspectos de la presente invención;

35 La Fig. 14 representa una realización de diversos elementos de un sistema de ordenador para incorporar y usar uno o más aspectos de la presente invención;

La Fig. 15A representa una realización de la unidad de ejecución del sistema de ordenador de la Fig. 14 para incorporar y usar uno o más aspectos de la presente invención;

40 La Fig. 15B representa una realización de la unidad de bifurcación del sistema de ordenador de la Fig. 14 para incorporar y usar uno o más aspectos de la presente invención;

La Fig. 15C representa una realización de la unidad de carga/almacenamiento del sistema de ordenador de la Fig. 14 para incorporar y usar uno o más aspectos de la presente invención; y

La Fig. 16 representa una realización de un sistema de ordenador servidor emulado para incorporar y usar uno o más aspectos de la presente invención.

Descripción detallada

Según un aspecto de la presente invención, se proporcionan una o más instrucciones de control para facilitar la comunicación con los adaptadores de un entorno informático. Las instrucciones de control están diseñadas específicamente para comunicar datos hacia y desde los espacios de direcciones de los adaptadores.

5 Tal como se usa en la presente memoria, el término adaptador incluye cualquier tipo de adaptador (por ejemplo, adaptador de almacenamiento, adaptador de red, adaptador de procesamiento, adaptador PCI, adaptador criptográfico, otro tipo de adaptadores de entrada/salida, etc.). En una realización, un adaptador incluye una función de adaptador. Sin embargo, en otras realizaciones, un adaptador puede incluir una pluralidad de funciones de adaptador. Uno o más aspectos de la presente invención son aplicables independientemente de si un adaptador
10 incluye una función de adaptador o una pluralidad de funciones de adaptador. En una realización, si un adaptador incluye una pluralidad de funciones de adaptador, entonces cada función puede ser comunicada según un aspecto de la presente invención. Además, en los ejemplos presentados en la presente memoria, el adaptador se usa de manera intercambiable con la función de adaptador (por ejemplo, función PCI) a menos que se indique lo contrario.

Una realización de un entorno informático para incorporar y usar uno o más aspectos de la presente invención se describe con referencia a la Fig. 1 A. En un ejemplo, un entorno 100 informático es un servidor System z[®] ofrecido por International Business Machines Corporation. El System z[®] está basado en la arquitectura z/Architecture[®] ofrecida por International Business Machines Corporation. Los detalles relativos a la arquitectura z/Architecture[®] se describen en una publicación de IBM[®] titulada, "z/Architecture Principles of Operation", Publicación IBM N° SA22-7832-07, Febrero de 2009. IBM[®], System z[®] y z/Architecture[®] son marcas registradas de International Business
15 Machines Corporation, Armonk, Nueva York. Otros nombres usados en la presente memoria pueden ser marcas comerciales registradas, marcas comerciales o nombres de productos de International Business Machines Corporation o de otras empresas.

En un ejemplo, el entorno 100 informático incluye una o más unidades de procesamiento central (CPUs) 102 acopladas a una memoria 104 de sistema (conocida también como memoria principal) a través de un controlador 106 de memoria. Para acceder a la memoria 104 de sistema, una unidad de procesamiento central 102 emite una solicitud de lectura o escritura que incluye una dirección usada para acceder a la memoria del sistema. Típicamente, la dirección incluida en la solicitud no es usable directamente para acceder a la memoria de sistema y, por lo tanto, es traducida a una dirección que es usable directamente para acceder a la memoria de sistema. La dirección es traducida por medio de un mecanismo de traducción (XLATE) 108. Por ejemplo, la dirección es
20 traducida de una dirección virtual a una dirección real o absoluta usando, por ejemplo, traducción de direcciones dinámicas (Dynamic Address Translation, DAT).

La solicitud, que incluye la dirección (traducida, si es necesario), es recibida por el controlador 106 de memoria. En un ejemplo, el controlador 106 de memoria está compuesto de hardware y es usado para arbitrar el acceso a la memoria del sistema y para mantener la coherencia de la memoria. Este arbitraje es realizado para las solicitudes recibidas desde las CPUs 102, así como para las solicitudes recibidas desde uno o más adaptadores 110. Al igual que las unidades de procesamiento central, los adaptadores emiten solicitudes a la memoria 104 de sistema para acceder a la memoria de sistema.

En un ejemplo, el adaptador 110 es un adaptador de tipo interconexión de componentes periféricos (PCI) o PCI express (PCIe) que incluye una o más funciones PCI. Una función PCI emite una solicitud que es enrutada a un concentrador 112 de entradas/salidas (por ejemplo, un concentrador PCI) a través de uno o más conmutadores 114 (por ejemplo, conmutadores PCIe). En un ejemplo, el concentrador de entradas/salidas comprende hardware, incluyendo una o más máquinas de estado, y está acoplado al controlador 106 de memoria a través de un bus 120 E/S-a-memoria.

El concentrador de entradas/salidas incluye, por ejemplo, un complejo 116 raíz que recibe la solicitud desde un conmutador. La solicitud incluye una dirección de entrada/salida que es proporcionada a una unidad 118 de traducción y protección de direcciones que accede a la información usada para la solicitud. Como ejemplos, la solicitud puede incluir una dirección de entrada/salida usada para realizar una operación de acceso directo a memoria (Direct Memory Access, DMA) o para solicitar una interrupción señalizada por mensaje (Message Signaled Interrupt, MSI). La unidad 118 de traducción y protección de direcciones accede a la información usada para la solicitud MSI o DMA. Como un ejemplo particular, para una operación de DMA, la información puede ser obtenida para traducir la dirección. A continuación, la dirección traducida es reenviada al controlador de memoria para acceder a la memoria de sistema.

En un ejemplo, tal como se describe con referencia a la Fig. 1B, la información usada para la solicitud MSI o DMA emitida por un adaptador es obtenida desde una entrada 130 de la tabla de dispositivos de una tabla 132 de dispositivos situada en el concentrador de E/S (por ejemplo, en la unidad de traducción y protección de
55 dispositivos).

direcciones). La entrada de la tabla de dispositivos incluye información para el adaptador, y cada adaptador tiene al menos una entrada de la tabla de dispositivos asociada con el mismo. Por ejemplo, hay una entrada en la tabla de dispositivos por cada espacio de direcciones (en la memoria del sistema) asignado al adaptador. Para una solicitud emitida desde un adaptador (por ejemplo, función 138 PCI), una entrada en la tabla de dispositivos es localizada usando un ID de solicitante proporcionado en la solicitud.

Con referencia ahora a la FIG. 1C, en una realización adicional de un entorno informático, además de o en lugar de una o más CPUs 102, un complejo de procesamiento central está acoplado al controlador 106 de memoria. En este ejemplo, un complejo 150 de procesamiento central incluye, por ejemplo, una o más particiones o zonas 152 (por ejemplo, particiones lógicas LP1-LPn), uno o más procesadores centrales (por ejemplo, CP1-CPm) 154, y un hipervisor 156 (por ejemplo, un gestor de particiones lógicas), cada uno de los cuales se describe a continuación.

Cada partición 152 lógica es capaz de funcionar como un sistema independiente. Es decir, cada partición lógica puede ser restablecida de manera independiente, cargada inicialmente con un sistema operativo o un hipervisor (tal como z/VM[®] proporcionado por International Business Machines Corporation, Armonk, Nueva York), si se desea, y funciona con diferentes programas. Un sistema operativo, un hipervisor o un programa de aplicación que se está ejecutando en una partición lógica parece tener acceso a un sistema total y completo, pero sólo está disponible una parte del mismo. Una combinación de hardware y código interno bajo licencia (Licensed Internal code) (conocido también como microcódigo o milicódigo) previene que un programa en una partición lógica interfiera con el programa en una partición lógica diferente. Esto permite que diversas particiones lógicas diferentes operen en un único procesador físico o múltiples procesadores físicos con una estrategia de división de tiempo. En este ejemplo particular, cada partición lógica tiene un sistema operativo 158 residente, que puede diferir para una o más particiones lógicas. En una realización, el sistema operativo 158 es un sistema operativo z/OS[®] o zLinux, proporcionado por International Business Machines Corporation, Armonk, Nueva York. z/OS[®] y z/VM[®] son marcas comerciales registradas de International Business Machines Corporation, Armonk, Nueva York.

Los procesadores 154 centrales son recursos de procesadores físicos que se asignan a las particiones lógicas. Por ejemplo, una partición 152 lógica incluye uno o más procesadores lógicos, cada uno de los cuales representa la totalidad o una parte del recurso 154 de procesador físico asignado a la partición. El recurso de procesador subyacente puede estar dedicado a esa partición o puede ser compartido con otra partición.

Las particiones 152 lógicas están gestionadas por un hipervisor 156 implementado mediante firmware que se ejecuta en los procesadores 154. Las particiones 152 lógicas y el hipervisor 156 comprenden, cada uno, uno o más programas residentes en partes respectivas del almacenamiento central asociado con los procesadores centrales. Un ejemplo de hipervisor 156 es el recurso de procesador/gestor de sistemas (Processor Resource/Systems Manager, PR/SM), proporcionado por International Business Machines Corporation, Armonk, Nueva York.

Tal como se usa en la presente memoria, el firmware incluye, por ejemplo, microcódigo, milicódigo y/o macrocódigo del procesador. Incluye, por ejemplo, instrucciones a nivel hardware y/o estructuras de datos usadas en la implementación de código de máquina de nivel superior. En una realización, incluye, por ejemplo, código propietario que típicamente es suministrado como microcódigo que incluye software de confianza o microcódigo específico del hardware subyacente y controla el acceso del sistema operativo al hardware del sistema.

Aunque en este ejemplo se describe un complejo de procesamiento central que tiene particiones lógicas, uno o más aspectos de la presente invención pueden ser incorporados en y usados por otras unidades de procesamiento, incluyendo unidades de procesamiento de un único procesador o de múltiples procesadores que no están particionados, entre otros ejemplos. El complejo central de procesamiento descrito en la presente memoria es sólo un ejemplo.

Tal como se ha descrito anteriormente, los adaptadores pueden emitir solicitudes a los procesadores solicitando diversas operaciones, tales como accesos directos a memoria, interrupciones señalizadas por mensajes, etc. Además, los procesadores pueden emitir solicitudes a los adaptadores. Por ejemplo, con relación a la Fig. 1B, un procesador 102 puede emitir una solicitud para acceder a una función 138 de adaptador. La solicitud es enrutada desde el procesador a la función de adaptador a través del concentrador 112 de E/S y uno o más conmutadores 114. En esta realización, no se muestra el controlador de memoria. Sin embargo, el concentrador de E/S puede estar acoplado al procesador directamente o a través de un controlador de memoria.

Como ejemplo, un sistema operativo 140 que se está ejecutando en el procesador emite una instrucción a la función de adaptador que solicita una operación particular. En este ejemplo, las instrucciones emitidas por el sistema operativo son específicas de la infraestructura de E/S. Es decir, debido a que la infraestructura de E/S está basada en PCI o PCIe (a los que, en la presente memoria, se hace referencia como PCI, a menos que se indique lo contrario), las instrucciones son instrucciones PCI. Las instrucciones PCI ejemplares incluyen carga PCI, almacenamiento PCI y almacenamiento de bloque PCI, por nombrar unas pocas. Aunque, en este ejemplo, la

infraestructura de E/S y las instrucciones están basadas en PCI, en otras realizaciones, pueden usarse otras infraestructuras e instrucciones correspondientes.

En un ejemplo particular, las instrucciones se dirigen a una ubicación específica dentro de un espacio de direcciones de la función de adaptador. Por ejemplo, tal como se muestra en la Fig. 2, una función 138 de adaptador incluye un almacenamiento 200, que está definido como una pluralidad de espacios de direcciones, incluyendo, por ejemplo: un espacio 202 de configuración (por ejemplo, el espacio de configuración PCI para una función PCI); un espacio 204 de E/S (por ejemplo, espacio PCI de E/S); y uno o más espacios 206 de memoria (por ejemplo, espacio PCI de memoria). En otras realizaciones, pueden proporcionarse más, menos espacios de direcciones o diferentes espacios de direcciones. Las instrucciones están dirigidas a un espacio de direcciones particular y una ubicación particular dentro del espacio de direcciones. Esto asegura que la configuración (por ejemplo, el sistema operativo, LPAR, procesador, huésped, etc.) que emite la instrucción está autorizada a acceder a la función de adaptador.

Para facilitar el procesamiento de las instrucciones, se usa la información almacenada en una o más estructuras de datos. Una de dichas estructuras de datos que incluye información relacionada con los adaptadores es una tabla 300 de funciones almacenada, por ejemplo, en una memoria segura. Tal como se muestra en la Fig. 3A, en un ejemplo, una tabla 300 de funciones incluye una o más entradas 302 de la tabla de funciones (Function Table Entries, FTEs). En un ejemplo, hay una entrada de la tabla de funciones por cada función de adaptador. Cada entrada 302 de la tabla de funciones incluye información a ser usada en el procesamiento asociado a su función de adaptador. En un ejemplo, la entrada 302 de la tabla de funciones incluye, por ejemplo:

Número 308 de instancia: Este campo indica una instancia particular del descriptor de función asociado con la entrada de la tabla de funciones;

Índice 310 de la entrada de tabla de dispositivos (Device Table Entry, DTE), 1 ... n: Puede haber uno o más índices de la tabla de dispositivos, y cada índice es un índice a una tabla de dispositivos para localizar una entrada de la tabla de dispositivos (DTE). Hay una o más entradas de la tabla de dispositivos por cada función de adaptador, y cada entrada incluye información asociada con su función de adaptador, incluyendo la información usada para procesar las solicitudes de la función de adaptador (por ejemplo, solicitudes DMA, solicitudes MSI) y la información relacionada con las solicitudes a la función de adaptador (por ejemplo, instrucciones PCI). Cada entrada de la tabla de dispositivos está asociada con un espacio de direcciones dentro de la memoria del sistema asignada a la función de adaptador. Una función de adaptador puede tener uno o más espacios de direcciones dentro de la memoria del sistema asignada a la función de adaptador.

Indicador 312 de ocupado: Este campo indica si la función de adaptador está ocupada;

Indicador 314 de estado de error permanente: Este campo indica si la función de adaptador está en un estado de error permanente;

Indicador 316 de recuperación iniciada: Este campo indica si se ha iniciado una recuperación para la función de adaptador;

Indicador 318 de permiso: Este campo indica si el sistema operativo que está tratando de habilitar la función de adaptador tiene autorización para hacerlo;

Indicador 320 de habilitación: Este campo indica si la función de adaptador está habilitada (por ejemplo, 1 = habilitada, 0 = deshabilitada);

Identificador (RID) 322 de solicitante: Este es un identificador de la función de adaptador y puede incluir, por ejemplo, un número de bus, un número de dispositivo y un número de función. Este campo se usa, por ejemplo, para los accesos de un espacio de configuración de la función de adaptador.

Por ejemplo, el espacio de configuración puede ser accedido especificando el espacio de configuración en una instrucción emitida por el sistema operativo (u otra configuración) para la función de adaptador. En la instrucción se especifican un desplazamiento en el espacio de configuración y un descriptor de función usado para localizar la entrada de la tabla de funciones apropiada que incluye el RID. El firmware recibe la instrucción y determina que es para un espacio de configuración. Por lo tanto, usa el RID para generar una solicitud al concentrador de E/S, y el concentrador de E/S crea una solicitud para acceder al adaptador. La localización de la función de adaptador se basa en el RID, y el desplazamiento especifica un desplazamiento en el espacio de configuración de la función de adaptador. Por ejemplo, el desplazamiento especifica un número de registro en el espacio de configuración.

Registro 324 de dirección base (Base Address Register, BAR) (1 a n): Este campo incluye una pluralidad de enteros sin signo, designados como BAR0 - BARn, que están asociados con la función de adaptador especificada

originalmente, y cuyos valores se almacenan también en los registros de dirección base asociados con la función de adaptador. Cada BAR especifica la dirección de inicio de un espacio de memoria o espacio de E/S dentro de la función de adaptador, e indica también el tipo de espacio de direcciones, es decir, si se trata de un espacio de memoria de 64 o 32 bits, o un espacio de E/S de 32 bits, por ejemplo;

- 5 En un ejemplo, se usa para accesos a un espacio de memoria y/o un espacio de E/S de la función de adaptador. Por ejemplo, un desplazamiento proporcionado en una instrucción para acceder a la función de adaptador es sumado al valor en el registro de dirección base asociado con el espacio de direcciones designado en la instrucción para obtener la dirección a ser usada para acceder a la función de adaptador. El identificador de espacio de direcciones en la instrucción identifica el espacio de direcciones dentro de la función de adaptador a ser accedida y el BAR correspondiente a ser usado;

Tamaño 326 (1 n): Este campo incluye una pluralidad de enteros sin signo, designados como SIZE0 - SIZE_n. El valor de un campo Tamaño, cuando es diferente de cero, representa el tamaño de cada espacio de direcciones en el que cada entrada corresponde a un BAR descrito anteriormente.

A continuación, se describen detalles adicionales acerca de BAR y Tamaño.

- 15 1. Cuando un BAR no está implementado para una función de adaptador, el campo BAR y su campo de tamaño correspondiente se almacenan ambos como ceros.
2. Cuando un campo BAR representa un espacio de direcciones de E/S o un espacio de direcciones de memoria de 32 bits, el campo de tamaño correspondiente es diferente de cero y representa el tamaño del espacio de direcciones.
- 20 3. Cuando un campo BAR representa un espacio de direcciones de memoria de 64 bits,
- a. El campo BAR_n representa los bits menos significativos de la dirección.
- b. El próximo campo BAR_n + 1 consecutivo representa los bits más significativos de la dirección.
- c. El campo SIZE_n correspondiente es diferente de cero y representa el tamaño del espacio de direcciones.
- 25 d. El correspondiente campo SIZE_n + 1 no es significativo y se almacena como cero.

Información 328 de enrutamiento interno: Esta información se usa para realizar un enrutamiento particular al adaptador. Incluye, por ejemplo, información de nodo, de chip procesador y de direccionamiento de concentrador de E/S, como ejemplos.

- 30 Indicador 330 de estado: Proporciona una indicación de si las operaciones de carga/almacenamiento están bloqueadas o no, así como otras indicaciones.

- En un ejemplo, el indicador de ocupado, el indicador de estado de error permanente y el indicador de recuperación iniciada se establecen en base a una supervisión llevada a cabo por el firmware. Además, el indicador de permiso se establece, por ejemplo, en base a una política. La información de BAR se basa en información de configuración descubierta durante el uso del bus por el procesador (por ejemplo, el firmware del procesador). Otros campos
- 35 pueden ser establecidos en base a la configuración, la inicialización y/o los eventos. En otras realizaciones, la entrada de la tabla de funciones puede incluir más, menos información o información diferente. La información incluida puede depender de las operaciones soportadas por o habilitadas para la función de adaptador.

- Para localizar una entrada de la tabla de funciones en una tabla de funciones que incluye una o más entradas, en una realización se usa un descriptor de función. Por ejemplo, uno o más bits del descriptor de función son usados como un índice para la tabla de funciones para localizar una entrada particular en la tabla de funciones.

- Con referencia a la Fig. 3B, en la misma se describen detalles adicionales con respecto a un descriptor de función. En un ejemplo, un descriptor 350 de funciones incluye un indicador 352 de habilitación que indica si el descriptor de función PCI está habilitado; un número 354 de función PCI que identifica la función (este es un identificador estático y, en una realización, es un índice en la tabla de funciones); y un número 356 de instancia que indica la
- 45 instancia particular de este descriptor de función. Por ejemplo, cada vez que se habilita la función, se incrementa el número de instancia para proporcionar un nuevo número de instancia.

Según un aspecto de la presente invención, para acceder a una función de adaptador, una configuración emite una solicitud a la función de adaptador, que es ejecutada por un procesador. En los ejemplos de la presente memoria, la configuración es un sistema operativo pero, en otros ejemplos, puede ser un sistema, un procesador, una

partición lógica, un huésped, etc. Estas solicitudes se realizan a través de instrucciones específicas, que acceden al adaptador. Las instrucciones ejemplares incluyen instrucciones de carga PCI, de almacenamiento PCI y de almacenamiento de bloque PCI. Estas instrucciones son específicas de la arquitectura del adaptador (por ejemplo, PCI). A continuación, se describen detalles adicionales acerca de estas instrucciones. Por ejemplo, una realización de una instrucción de carga PCI se describe con referencia a las Figs. 4A-5B; una realización de una instrucción de almacenamiento PCI se describe con referencia a las Figs. 6A-7B; y una realización de una instrucción de almacenamiento de bloque PCI se describe con referencia a las Figs. 8A-9B.

Con referencia inicialmente a la Fig. 4A, en la misma se representa una realización de una instrucción de carga PCI. Tal como se muestra, una instrucción 400 de carga PCI incluye, por ejemplo, un código 402 de operación que indica la instrucción de carga PCI; un primer campo 404 que especifica una ubicación en la que serán cargados los datos recuperados desde una función de adaptador; y un segundo campo 406 que especifica una ubicación en la que se incluye diversa información acerca de la función de adaptador desde la que deben cargarse los datos. A continuación, se describen adicionalmente los contenidos de las ubicaciones designadas por los campos 1 y 2.

En un ejemplo, el Campo 1 designa un registro general y, tal como se representa en la Fig. 4B, los contenidos de ese registro 404 incluyen un rango contiguo de uno o más bytes cargados desde la ubicación de la función de adaptador especificada en la instrucción. En un ejemplo, los datos se cargan en las posiciones de byte situadas más a la derecha del registro.

En una realización, el Campo 2 designa un par de registros generales que incluyen diversa información. Tal como se muestra en la Fig. 4B, los contenidos de los registros incluyen, por ejemplo:

Descriptor 410 habilitado: Este campo es un descriptor de función habilitada de la función de adaptador desde la cual deben cargarse los datos;

Espacio 412 de direcciones: Este campo identifica el espacio de direcciones dentro de la función de adaptador desde el cual deben cargarse los datos;

Desplazamiento 414 dentro del espacio de direcciones: Este campo especifica el desplazamiento dentro del espacio de direcciones especificado desde el cual deben cargarse los datos;

Campo 416 de longitud: Este campo especifica la longitud de la operación de carga (por ejemplo, el número de bytes a cargar); y

Campo 418 de estado: Este campo proporciona un código de estado que es aplicable cuando la instrucción se completa con un código de condición predefinido.

En una realización, los bytes cargados desde la función de adaptador deberán ser contenidos dentro de un límite integral en el espacio de direcciones PCI designado de la función de adaptador. Cuando el campo de espacio de direcciones designa un espacio de direcciones de memoria, el tamaño del límite integral es, por ejemplo, una palabra doble. Cuando el campo de espacio de direcciones designa un espacio de direcciones de E/S o un espacio de direcciones de configuración, el tamaño del límite integral es, por ejemplo, una palabra.

Una realización de la lógica asociada con una instrucción de carga PCI se describe con referencia a las Figs. 5A-5B. En un ejemplo, la instrucción es emitida por un sistema operativo (u otra configuración) y ejecutada por el procesador (por ejemplo, firmware) que ejecuta el sistema operativo. En los ejemplos en la presente memoria, las instrucciones y funciones de adaptador están basadas en PCI. Sin embargo, en otros ejemplos, pueden usarse una arquitectura de adaptador diferente e instrucciones correspondientes.

Para emitir la instrucción, el sistema operativo proporciona los siguientes operandos a la instrucción (por ejemplo, en uno o más registros designados por la instrucción): el descriptor de función PCI, el espacio de direcciones PCI (PCIAS), el desplazamiento en el espacio de direcciones PCI y la longitud de los datos a cargar. Tras completar con éxito la instrucción de carga PCI, los datos se cargan en la ubicación (por ejemplo, registro) designada por la instrucción.

Con referencia a la Fig. 5A, inicialmente, se realiza una determinación de si está instalada o no la función que permite una instrucción de carga PCI, PREGUNTA 500. Esta determinación se realiza, por ejemplo, comprobando un indicador almacenado, por ejemplo, en un bloque de control. Si la función no está instalada, se proporciona una condición de excepción, ETAPA 502. De lo contrario, se realiza una determinación de si los operandos están alineados, PREGUNTA 504. Por ejemplo, si ciertos operandos necesitan estar en pares de registro pares/impares, se realiza una determinación de si se cumplen o no los requisitos. Si los operandos no están alineados, entonces se proporciona una excepción, ETAPA 506. De lo contrario, si la función está instalada y los operandos están alineados, se realiza una determinación de si el descriptor proporcionado en los operandos de la instrucción de

carga PCI está habilitado o no, PREGUNTA 508. En un ejemplo, esta determinación se realiza comprobando un indicador de habilitación en el descriptor. Si el descriptor no está habilitado, entonces se proporciona una condición de excepción, ETAPA 510.

5 Si el descriptor está habilitado, el descriptor es usado para localizar una entrada en la tabla de funciones, ETAPA 512. Es decir, al menos una parte del descriptor es usada como un índice a la tabla de funciones para localizar la entrada en la tabla de funciones correspondiente a la función de adaptador desde la que se cargan los datos.

10 A continuación, si la configuración que emite la instrucción es un huésped, se realiza una determinación de si la función está configurada para su uso por un huésped, PREGUNTA 514. Si no está autorizada, entonces se proporciona una condición de excepción, ETAPA 516. Esta pregunta puede ser ignorada si la configuración no es un huésped o pueden comprobarse otras autorizaciones, si han sido designadas. (En un ejemplo, en la arquitectura z/Architecture[®], se ejecuta interpretativamente un huésped paginable a través de la instrucción inicio de ejecución interpretativa (Start Interpretive Execution, SIE), en el nivel 2 de interpretación. Por ejemplo, el hipervisor de partición lógica (Logical Partition, LPAR) ejecuta la instrucción SIE para iniciar la partición lógica en una memoria física fija. Si z/VM[®] es el sistema operativo en esa partición lógica, emite la instrucción SIE para ejecutar sus máquinas huéspedes (virtuales) en su almacenamiento V = V (virtual). Por lo tanto, el hipervisor LPAR usa SIE nivel- 1, y el hipervisor z/VM[®] usa SIE nivel-2).

A continuación, se realiza una determinación de si la función está habilitada, PREGUNTA 518. En un ejemplo, esta determinación se realiza comprobando un indicador de habilitación en la entrada de la tabla de funciones. Si no está habilitado, entonces se proporciona una condición de excepción, ETAPA 520.

20 Si la función está habilitada, entonces se realiza una determinación de si el espacio de direcciones es válido o no, PREGUNTA 522. Por ejemplo, si el espacio de direcciones especificado es un espacio de direcciones designado de la función de adaptador y uno que sea apropiado para esta instrucción. Si el espacio de direcciones no es válido, entonces se proporciona una condición de excepción, ETAPA 524. De lo contrario, se realiza una determinación de si la carga/almacenamiento está bloqueada, PREGUNTA 526. En un ejemplo, esta determinación se realiza comprobando el indicador de estado en la entrada de la tabla de funciones. Si la carga/almacenamiento está bloqueada, entonces se proporciona una condición de excepción, ETAPA 528.

30 Sin embargo, si la carga/almacenamiento no está bloqueada, se realiza una determinación de si la recuperación está activa, PREGUNTA 530. En un ejemplo, esta determinación se realiza comprobando el indicador de recuperación iniciada en la entrada de la tabla de funciones. Si la recuperación está activa, entonces se proporciona una condición de excepción, ETAPA 532. De lo contrario, se realiza una determinación de si la función está ocupada, PREGUNTA 534. Esta determinación se realiza comprobando el indicador de ocupado en la entrada de la tabla de funciones. Si la función está ocupada, entonces se proporciona una condición de ocupado, ETAPA 536. Con una condición de ocupado, la instrucción puede intentarse de nuevo, en lugar de ser ignorada.

35 Si la función no está ocupada, entonces se realiza una determinación adicional de si el desplazamiento especificado en la instrucción es válido, PREGUNTA 538. Es decir, está el desplazamiento en combinación con la longitud de la operación dentro de la base y la longitud del espacio de direcciones, tal como se especifica en la entrada de la tabla de funciones. Si no, entonces se proporciona una condición de excepción, ETAPA 540. Sin embargo, si el desplazamiento es válido, entonces se realiza una determinación de si la longitud es válida, PREGUNTA 542. Es decir, en función del tipo de espacio de direcciones, el desplazamiento dentro del espacio de direcciones y un tamaño de límite integral, si la longitud es válida. Si no, entonces se proporciona una condición de excepción, ETAPA 544. De lo contrario, el procesamiento continúa con la instrucción de carga. (En una realización, el firmware realiza las comprobaciones anteriores).

45 Continuando con la Fig. 5B, el firmware realiza una determinación de si la carga es para un espacio de direcciones de configuración de la función de adaptador, PREGUNTA 550. Es decir, en base a la configuración de la memoria de la función de adaptador, es el espacio de direcciones especificado proporcionado en la instrucción de un espacio de configuración. Si es así, entonces el firmware realiza diversos procesamientos para proporcionar la solicitud a un concentrador acoplado a la función de adaptador; a continuación, el concentrador enruta la solicitud a la función, ETAPA 552.

50 Por ejemplo, el firmware obtiene el ID del solicitante a partir de la entrada de la tabla de funciones a la que apunta el descriptor de función proporcionado en los operandos de la instrucción. Además, el firmware determina, en base a la información en la entrada de la tabla de funciones (por ejemplo, la información de enrutamiento interno) del concentrador para recibir esta solicitud. Es decir, un entorno puede tener uno o más concentradores y el firmware determina el concentrador acoplado a la función de adaptador. A continuación, reenvía la solicitud al concentrador. El concentrador genera un paquete de solicitud de lectura de configuración que fluye en el bus PCI a la función de adaptador identificada por el RID, en la entrada de la tabla de funciones. La solicitud de lectura de configuración

incluye el RID y el desplazamiento (es decir, la dirección de datos) que se usan para recuperar los datos, tal como se describe a continuación.

Volviendo a la PREGUNTA 550, si el espacio de direcciones designado no es un espacio de configuración, entonces, una vez más, el firmware realiza diversos procesamiento para proporcionar la solicitud al concentrador, ETAPA 554. El firmware usa el descriptor para seleccionar una entrada de la tabla de funciones y, a partir de esa entrada, obtiene información para localizar el concentrador apropiado. También calcula una dirección de datos a ser usada en la operación de carga. Esta dirección se calcula sumando la dirección de inicio BAR (en el que el BAR es aquel asociado con el identificador de espacio de direcciones proporcionado en la instrucción) obtenida desde la entrada de la tabla de funciones al desplazamiento proporcionado en la instrucción. Esta dirección de datos calculada es proporcionada al concentrador. A continuación, el concentrador toma esa dirección y la incluye en un paquete de solicitud, tal como un paquete de solicitud de lectura DMA, que fluye a través del bus PCI a la función de adaptador.

En respuesta a la recepción de la solicitud a través de la ETAPA 552 o la ETAPA 554, la función de adaptador recupera los datos solicitados desde la ubicación especificada (es decir, en la dirección de datos) y devuelve esos datos en respuesta a la solicitud, ETAPA 556. La respuesta es reenviada desde la función de adaptador al concentrador de E/S. En respuesta a la recepción de la respuesta, el concentrador reenvía la respuesta al procesador iniciador. A continuación, el procesador iniciador toma los datos desde el paquete de respuesta y los carga en la ubicación designada especificada en la instrucción (por ejemplo, el campo 1 404). La operación de carga PCI concluye con una indicación de éxito (por ejemplo, estableciendo un código de condición de cero).

Además de una instrucción de carga que recupera datos desde una función de adaptador y los almacena en una ubicación designada, otra instrucción que puede ser ejecutada es una instrucción de almacenamiento. La instrucción de almacenamiento almacena datos en una ubicación especificada en la función de adaptador. Una realización de una instrucción de almacenamiento PCI se describe con referencia a la Fig. 6A. Tal como se muestra, una instrucción 600 de almacenamiento PCI incluye, por ejemplo, un código 602 de operación que indica la instrucción de almacenamiento PCI; un primer campo 604 que especifica una ubicación que incluye los datos a ser almacenados en una función de adaptador; y un segundo campo 606 que especifica una ubicación en la que se incluye diversa información relativa a la función de adaptador en la que deben almacenarse los datos. Los contenidos de las ubicaciones designadas por los Campos 1 y 2 se describen adicionalmente a continuación.

En un ejemplo, el Campo 1 designa un registro general y, tal como se representa en la Fig. 6B, los contenidos 604 de ese registro incluyen un rango contiguo de uno o más bytes de datos a ser almacenados en la ubicación especificada de una función de adaptador. En un ejemplo, se almacenan los datos en las posiciones de byte más a la derecha del registro.

En una realización, el Campo 2 designa un par de registros generales que incluyen diversa información. Tal como se muestra en la Fig. 6B, los contenidos del registro incluyen, por ejemplo:

Descriptor 610 activado: Este campo es un descriptor de función habilitado de la función de adaptador en el cual deben almacenarse los datos;

Espacio 612 de direcciones: Este campo identifica el espacio de direcciones dentro de la función de adaptador en el cual deben almacenarse los datos;

Desplazamiento 614 dentro del espacio de direcciones: Este campo especifica el desplazamiento dentro del espacio de direcciones especificado en el cual deben almacenarse los datos;

Campo 616 de longitud: Este campo especifica la longitud de la operación de almacenamiento (por ejemplo, el número de bytes a almacenar); y

Campo 618 de estado: Este campo proporciona un código de estado que es aplicable cuando la instrucción se completa con un código de condición predefinido.

Una realización de la lógica asociada con una instrucción de almacenamiento PCI se describe con referencia a las Figs. 7A-7B. En un ejemplo, la instrucción es emitida por un sistema operativo, y es ejecutada por el procesador (por ejemplo, firmware) que ejecuta el sistema operativo.

Para emitir la instrucción, el sistema operativo proporciona los siguientes operandos a la instrucción (por ejemplo, en uno o varios registros designados por la instrucción): el descriptor de función PCI, el espacio de direcciones PCI (PCIAS), el desplazamiento en el espacio de direcciones PCI, la longitud de los datos a almacenar y un puntero a los datos a almacenar. Tras completar con éxito la instrucción de almacenamiento PCI, los datos se almacenan en la ubicación designada por la instrucción.

Con referencia a la Fig. 7A, inicialmente, se realiza una determinación de si está instalada la función que permite una instrucción de almacenamiento PCI, PREGUNTA 700. Esta determinación se realiza, por ejemplo, comprobando un indicador almacenado, por ejemplo, en un bloque de control. Si la función no está instalada, se proporciona una condición de excepción, ETAPA 702. De lo contrario, se realiza una determinación de si los operandos están alineados, PREGUNTA 704. Por ejemplo, si ciertos operandos tienen que estar en pares de registros pares/impares, se realiza una determinación de si se cumplen o no esos requisitos. Si los operandos no están alineados, entonces se proporciona una excepción, ETAPA 706. De lo contrario, si la función está instalada y los operandos están alineados, se realiza una determinación de si el descriptor proporcionado en los operandos de la instrucción de almacenamiento PCI está habilitado o no, PREGUNTA 708. En un ejemplo, esta determinación se realiza comprobando un indicador de habilitación en el descriptor. Si el descriptor no está habilitado, entonces se proporciona una condición de excepción, ETAPA 710.

Si el descriptor está habilitado, entonces el descriptor es usado para localizar una entrada de la tabla de funciones, ETAPA 712. Es decir, al menos una parte del descriptor es usada como un índice en la tabla de funciones para localizar la entrada de la tabla de funciones correspondiente a la función de adaptador en la que deben almacenarse los datos.

A continuación, si la configuración que emite la instrucción es un huésped, se realiza una determinación de si la función está configurada para su uso por un huésped, PREGUNTA 714. Si no está autorizada, entonces se proporciona una condición de excepción, ETAPA 716. Esta pregunta puede ser ignorada si la configuración no es un huésped o pueden comprobarse otras autorizaciones, si han sido designadas.

A continuación, se realiza una determinación de si la función está habilitada, PREGUNTA 718. En un ejemplo, esta determinación se realiza comprobando un indicador de habilitación en la entrada de la tabla de funciones. Si no está habilitada, entonces se proporciona una condición de excepción, ETAPA 720.

Si la función está habilitada, entonces se realiza una determinación de si el espacio de direcciones es válido, PREGUNTA 722. Por ejemplo, si el espacio de direcciones especificado es un espacio de direcciones designado de la función de adaptador y uno que es apropiado para esta instrucción. Si el espacio de direcciones no es válido, se proporciona una condición de excepción, ETAPA 724. De lo contrario, se realiza una determinación de si la carga/almacenamiento está bloqueada, PREGUNTA 726. En un ejemplo, esta determinación se realiza comprobando el indicador de estado en la entrada de la tabla de funciones. Si la carga/almacenamiento está bloqueada, entonces se proporciona una condición de excepción, ETAPA 728.

Sin embargo, si la carga/almacenamiento no está bloqueada, se realiza una determinación de si la recuperación está activa, PREGUNTA 730. En un ejemplo, esta determinación se realiza comprobando el indicador de recuperación iniciada en la entrada de la tabla de funciones. Si la recuperación está activa, entonces se proporciona una condición de excepción, ETAPA 732. De lo contrario, se realiza una determinación de si la función está ocupada, PREGUNTA 734. Esta determinación se realiza comprobando el indicador de ocupado en la entrada de la tabla de funciones. Si la función está ocupada, entonces se proporciona una condición de ocupado, ETAPA 736. Con una condición de ocupado, la instrucción puede ser intentada de nuevo, en lugar de ser ignorada.

Si la función no está ocupada, entonces se realiza una determinación adicional de si el desplazamiento especificado en la instrucción es válido o no, PREGUNTA 738. Es decir, está el desplazamiento en combinación con la longitud de la operación dentro de la base y la longitud del espacio de direcciones, tal como se especifica en la entrada de la tabla de funciones. Si no, entonces se proporciona una condición de excepción, ETAPA 740. Sin embargo, si el desplazamiento es válido, entonces se realiza una determinación de si la longitud es válida, PREGUNTA 742. Es decir, en función del tipo de espacio de direcciones, el desplazamiento dentro del espacio de direcciones, y un tamaño de límite integral es la longitud válida. Si no, entonces se proporciona una condición de excepción, ETAPA 744. De lo contrario, el procesamiento continúa con la instrucción de almacenamiento. (En una realización, el firmware realiza las comprobaciones anteriores).

Continuando con la Fig. 7B, el firmware realiza una determinación de si el almacenamiento es para un espacio de direcciones de configuración de la función de adaptador, PREGUNTA 750. Es decir, en base a la configuración de la memoria de la función de adaptador, es el espacio de direcciones especificado en la instrucción de un espacio de configuración. Si es así, entonces el firmware realiza diversos procesamiento para proporcionar la solicitud a un concentrador acoplado a la función de adaptador; a continuación, el concentrador enruta la solicitud a la función, ETAPA 752.

Por ejemplo, el firmware obtiene el ID de solicitante a partir de la entrada de la tabla de funciones a la que apunta el descriptor de función proporcionado en los operandos de la instrucción. Además, el firmware determina, en base a la información en la entrada de la tabla de funciones (por ejemplo, la información de enrutamiento interno) el concentrador para recibir esta solicitud. Es decir, un entorno puede tener uno o más concentradores y el firmware

determina el concentrador acoplado a la función de adaptador. A continuación, reenvía la solicitud al concentrador. El concentrador genera un paquete de solicitud de escritura de configuración que fluye en el bus PCI a la función de adaptador identificada por el RID en la entrada de la tabla de funciones. La solicitud de escritura de configuración incluye el RID y el desplazamiento (es decir, dirección de datos) que se usan para almacenar los datos, tal como se describe a continuación.

Volviendo a la PREGUNTA 750, si el espacio de direcciones designado no es un espacio de configuración, entonces, una vez más, el firmware realiza diversos procesamiento para proporcionar la solicitud al concentrador, ETAPA 754. El firmware usa el descriptor para seleccionar una entrada de la tabla de funciones y a partir de esa entrada obtiene información para localizar el concentrador correspondiente. También calcula una dirección de datos a ser usada en la operación de almacenamiento. Esta dirección se calcula sumando la dirección de inicio BAR obtenida a partir de la entrada de la tabla de funciones al desplazamiento proporcionado en la instrucción. Esta dirección de datos calculada es proporcionada al concentrador. A continuación, el concentrador toma esa dirección y la incluye en un paquete de solicitud, tal como un paquete de solicitud de escritura DMA, que fluye a través del bus PCI a la función de adaptador.

En respuesta a la recepción de la solicitud, a través de la ETAPA 752 o la ETAPA 754, la función de adaptador almacena los datos solicitados en la ubicación especificada (es decir, en la dirección de los datos), ETAPA 756. La operación de almacenamiento PCI concluye con una indicación de éxito (por ejemplo, estableciendo un código de condición de cero).

Además de las instrucciones de carga y almacenamiento, que típicamente cargan o almacenan un máximo de, por ejemplo, 8 bytes, otra instrucción que puede ser ejecutada es una instrucción de almacenamiento de bloque. La instrucción de almacenamiento de bloque almacena grandes bloques de datos (por ejemplo, 16, 32, 64, 128 o 256 bytes) en una ubicación especificada en la función de adaptador; los tamaños de bloque no están limitados necesariamente a tamaños que son potencias de dos. En un ejemplo, la ubicación especificada está en un espacio de memoria de la función de adaptador (no un espacio de E/S o de configuración).

Una realización de una instrucción de almacenamiento de bloque PCI se describe con referencia a la Fig. 8A. Tal como se muestra, una instrucción 800 de almacenamiento de bloque PCI incluye, por ejemplo, un código 802 de operación que indica la instrucción de almacenamiento de bloque PCI; un primer campo 804 que especifica una ubicación en la que se incluye diversa información acerca de la función de adaptador en la que deben almacenarse los datos; un segundo campo 806 que especifica una ubicación que incluye un desplazamiento dentro del espacio de direcciones especificado en el que deben almacenarse los datos; y un tercer campo 808 que especifica una ubicación que incluye una dirección en la memoria del sistema de los datos a ser almacenados en la función de adaptador. Los contenidos de las ubicaciones designadas por los Campos 1, 2 y 3 se describen adicionalmente a continuación.

En una realización, el Campo 1 designa un registro general que incluye diversa información. Tal como se muestra en la Fig. 8B, los contenidos del registro incluyen, por ejemplo:

Descriptor 810 habilitado: Este campo es un descriptor de función habilitado de la función de adaptador en la cual deben almacenarse los datos;

Espacio 812 de direcciones: Este campo identifica el espacio de direcciones dentro de la función de adaptador en el cual deben almacenarse los datos;

Campo 814 de longitud: Este campo especifica la longitud de la operación de almacenamiento (por ejemplo, el número de bytes a almacenar); y

Campo 816 de estado: Este campo proporciona un código de estado que es aplicable cuando la instrucción se completa con un código de condición predefinido.

En un ejemplo, el Campo 2 designa un registro general y, tal como se representa en la Fig. 8C, los contenidos del registro incluyen un valor (por ejemplo, un entero de 64 bits sin signo) que especifica el desplazamiento dentro del espacio de direcciones especificado en el cual deben almacenarse los datos.

En un ejemplo, el Campo 3, tal como se representa en la Fig. 8D, incluye la dirección lógica en la memoria del sistema del primer byte de datos 822 a ser almacenado en la función de adaptador.

Una realización de la lógica asociada con una instrucción de almacenamiento de bloque PCI se describe con referencia a las Figs. 9A-9B. En un ejemplo, la instrucción es emitida por un sistema operativo, y es ejecutada por el procesador (por ejemplo, firmware) que ejecuta el sistema operativo.

Para emitir la instrucción, el sistema operativo proporciona los siguientes operandos a la instrucción (por ejemplo, en uno o más registros designados por la instrucción): el descriptor de función PCI, el espacio de direcciones PCI (PCIAS), el desplazamiento en el espacio de direcciones PCI, la longitud de los datos a almacenar y un puntero a los datos a almacenar. El operando puntero puede comprender un registro y un desplazamiento con o sin signo.

5 Tras completar con éxito la instrucción de almacenamiento de bloque PCI, los datos se almacenan en la ubicación en el adaptador designado por la instrucción.

10 Con referencia a la Fig. 9A, inicialmente, se realiza una determinación de si la función que permite una instrucción de almacenamiento de bloque PCI está instalada o no, PREGUNTA 900. Esta determinación se realiza, por ejemplo, comprobando un indicador almacenado, por ejemplo, en un bloque de control. Si la función no está instalada, se proporciona una condición de excepción, ETAPA 902. De lo contrario, si la función está instalada, se realiza una determinación de si el descriptor proporcionado en los operandos de la instrucción de almacenamiento de bloque PCI está habilitado, PREGUNTA 904. En un ejemplo, esta determinación se realiza comprobando un indicador de habilitación en el descriptor. Si el descriptor no está habilitado, entonces se proporciona una condición de excepción, ETAPA 906.

15 Si el descriptor está habilitado, el descriptor es usado para localizar una entrada de la tabla de funciones, ETAPA 912. Es decir, al menos una parte del descriptor es usada como un índice en la tabla de funciones para localizar la entrada de la tabla de funciones correspondiente a la función de adaptador en que deben almacenarse los datos.

20 A continuación, si la configuración que emite la instrucción es un huésped, se realiza una determinación de si la función está configurada para su uso por un huésped, PREGUNTA 914. Si no está autorizada, entonces se proporciona una condición de excepción, ETAPA 916. Esta pregunta puede ser ignorada si la configuración no es un huésped o pueden comprobarse otras autorizaciones, si han sido designadas.

A continuación, se realiza una determinación de si la función está habilitada, PREGUNTA 918. En un ejemplo, esta determinación se realiza comprobando un indicador de habilitación en la entrada de la tabla de funciones. Si no está habilitada, entonces se proporciona una condición de excepción, ETAPA 920.

25 Si la función está activada, entonces se realiza una determinación de si el espacio de direcciones es válido, PREGUNTA 922. Por ejemplo, si el espacio de direcciones especificado es un espacio de direcciones designado de la función de adaptador y uno que sea apropiado para esta instrucción (es decir, un espacio de memoria). Si el espacio de direcciones no es válido, se proporciona una condición de excepción, ETAPA 924. De lo contrario, se realiza una determinación de si la carga/almacenamiento está bloqueada, PREGUNTA 926. En un ejemplo, esta determinación se realiza comprobando el indicador de estado en la entrada de la tabla de funciones. Si la carga/almacenamiento está bloqueada, entonces se proporciona una condición de excepción, ETAPA 928.

30 Sin embargo, si la carga/almacenamiento no está bloqueada, se realiza una determinación de si la recuperación está activada, PREGUNTA 930. En un ejemplo, esta determinación se realiza comprobando el indicador de recuperación iniciada en la entrada de la tabla de funciones. Si la recuperación está activa, entonces se proporciona una condición de excepción, ETAPA 932. De lo contrario, se realiza una determinación de si la función está ocupada, PREGUNTA 934. Esta determinación se realiza comprobando el indicador de ocupado en la entrada de la tabla de funciones. Si la función está ocupada, entonces se proporciona una condición de ocupado, ETAPA 936. Con una condición de ocupado, la instrucción puede ser intentada de nuevo, en lugar de ser ignorada.

40 Si la función no está ocupada, entonces se realiza una determinación adicional de si el desplazamiento especificado en la instrucción es válido, PREGUNTA 938. Es decir, está el desplazamiento en combinación con la longitud de la operación dentro de la base y la longitud del espacio de direcciones, tal como se especifica en la entrada de la tabla de funciones. Si no, entonces se proporciona una condición de excepción, ETAPA 940. Sin embargo, si el desplazamiento es válido, entonces se realiza una determinación de si la longitud es válida, PREGUNTA 942. Es decir, en función del tipo de espacio de direcciones, desplazamiento dentro del espacio de direcciones, y un tamaño límite integral es la longitud válida. Si no, entonces se proporciona una condición de excepción, ETAPA 944. De lo contrario, el procesamiento continúa con la instrucción de almacenamiento de bloque (en una realización, el firmware realiza las comprobaciones anteriores).

45 Continuando con la Fig. 9B, el firmware realiza una determinación de si el almacenamiento que incluye los datos a ser almacenados es accesible, PREGUNTA 950. Si no, se proporciona una condición de excepción, ETAPA 952. Si es así, entonces el firmware realiza diversos procesamiento para proporcionar la solicitud a un concentrador acoplado a la función de adaptador; a continuación, el concentrador enruta la solicitud a la función, ETAPA 954.

Por ejemplo, el firmware usa el descriptor para seleccionar una entrada de la tabla de funciones y desde esa entrada obtiene información para localizar el concentrador apropiado. También calcula una dirección de datos a ser usada en la operación de almacenamiento de bloque. Esta dirección se calcula sumando la dirección de inicio

BAR (en el que el BAR es identificado por el identificador de espacio de direcciones), obtenida a partir de la entrada de la tabla de funciones, al desplazamiento proporcionado en la instrucción. Esta dirección de datos calculada es proporcionada al concentrador. Además, los datos referenciados por la dirección proporcionada en la instrucción son recuperados de la memoria del sistema y proporcionados al concentrador de E/S. A continuación, el concentrador toma esa dirección y los datos y los incluye en un paquete de solicitud, tal como un paquete de solicitud de escritura DMA, que fluye a través del bus PCI a la función de adaptador.

En respuesta a la recepción de la solicitud, la función de adaptador almacena los datos solicitados en la ubicación especificada (es decir, en la dirección de datos), ETAPA 956. La operación de almacenamiento de bloque PCI concluye con una indicación de éxito (por ejemplo, estableciendo un código de condición de cero).

Se ha descrito detalladamente una capacidad de comunicación con los adaptadores de un entorno informático usando instrucciones de control diseñadas específicamente para dicha comunicación. La comunicación es realizada sin usar E/S mapeadas en memoria y no se limita a los registros de control en la función de adaptador. Las instrucciones aseguran que la configuración que emite la instrucción está autorizada a acceder a la función de adaptador. Además, para la instrucción de almacenamiento de bloque, asegura que la ubicación de almacenamiento principal especificada está dentro de la memoria de la configuración.

En las realizaciones descritas en la presente memoria, los adaptadores son adaptadores PCI. PCI, tal como se usa en la presente memoria, se refiere a cualquier adaptador implementado según una especificación basada en PCI según se define por el grupo de interés especial en interconexión de componentes periféricos (Peripheral Component Interconnect Special Interest Group, PCI-SIG), incluyendo pero sin limitarse a, PCI o PCIe. En un ejemplo particular, interconexión de componentes periféricos express (PCIe) es un estándar de interconexión a nivel de componente que define un protocolo de comunicación bidireccional para las transacciones entre los adaptadores de E/S y los sistemas servidor. Las comunicaciones PCIe se encapsulan en paquetes según el estándar PCIe para su transmisión en un bus PCIe. Las transacciones originadas en los adaptadores de E/S y dirigidas a los sistemas servidor se denominan transacciones salientes "upbound". Las transacciones originadas en los sistemas servidor y dirigidas a los adaptadores de E/S se denominan transacciones entrantes "downbound". La topología PCIe está basada en enlaces unidireccionales punto a punto que están emparejados (por ejemplo, un enlace "upbound", un enlace "downbound") para formar el bus PCIe. El estándar PCIe es mantenido y publicado por el PCI-SIG, tal como se ha indicado anteriormente en la sección de antecedentes.

Tal como apreciará una persona con conocimientos en la materia, los aspectos de la presente invención pueden realizarse como un sistema, procedimiento o producto de programa de ordenador. Por consiguiente, los aspectos de la presente invención pueden adoptar la forma de una realización totalmente en hardware, una realización totalmente en software (incluyendo firmware, software residente, microcódigo, etc.) o una realización que combina aspectos de software y hardware, todos los cuales pueden ser denominados, en general, en la presente memoria, como un "circuito", "módulo" o "sistema". Además, los aspectos de la presente invención pueden adoptar la forma de un producto de programa de ordenador realizado en un medio o medios legibles por uno o más ordenadores, que tienen un código de programa legible materializado en los mismos.

Puede utilizarse cualquier combinación de medio o medios legibles por uno o más ordenadores. El medio legible por ordenador puede ser un medio de almacenamiento legible por ordenador. Un medio de almacenamiento legible por ordenador puede ser, por ejemplo, pero sin limitarse a, un sistema, aparato o dispositivo electrónico, magnético, óptico, electromagnético, infrarrojo o semiconductor, o cualquier combinación adecuada de los anteriores. Los ejemplos más específicos (una lista no exhaustiva) del medio de almacenamiento legible por ordenador incluyen los siguientes: una conexión eléctrica que tiene uno o más cables, un disquete de ordenador portátil, un disco duro, una memoria de acceso aleatorio (RAM), una memoria de sólo lectura (ROM), una memoria de sólo lectura programable y borrable (EPROM o memoria Flash), una fibra óptica, una memoria de sólo lectura de disco compacto portátil (CD-ROM), un dispositivo de almacenamiento óptico, un dispositivo de almacenamiento magnético o cualquier combinación adecuada de los anteriores. En el contexto de la presente memoria, un medio de almacenamiento legible por ordenador puede ser cualquier medio tangible que pueda contener o almacenar un programa para su uso por o en conexión con un sistema, aparato o dispositivo de ejecución de instrucciones.

Con referencia ahora a la Fig. 10, en un ejemplo, un producto 1000 de programa de ordenador incluye, por ejemplo, uno o más medios 1002 de almacenamiento legibles por ordenador para almacenar medios o lógica 1004 de código de programa legible ordenador para proporcionar y facilitar uno o más aspectos de la presente invención.

El código de programa incorporado en un medio legible por ordenador puede ser transmitido usando un medio apropiado, incluyendo pero sin limitarse a tecnología inalámbrica, cable, cable de fibra óptica, RF, etc., o cualquier combinación adecuada de los anteriores.

El código de programa de ordenador para realizar las operaciones para los aspectos de la presente invención puede estar escrito en cualquier combinación de uno o más lenguajes de programación, incluyendo un lenguaje de programación orientado a objetos, tal como Java, Smalltalk, C++ o similares, y lenguajes de programación de procedimientos convencionales, tales como el lenguaje de programación "C", ensamblador o lenguajes de programación similares. El código de programa puede ser ejecutado por completo en el ordenador del usuario, en parte en el equipo del usuario, como un paquete de software independiente, en parte en el ordenador del usuario y en parte en un ordenador remoto o totalmente en el ordenador remoto o servidor. En el último escenario, el ordenador remoto puede estar conectado al ordenador del usuario a través de cualquier tipo de red, incluyendo una red de área local (LAN) o una red de área amplia (WAN), o la conexión puede realizarse a un ordenador externo (por ejemplo, a través de Internet usando un proveedor de servicios de internet).

En la presente memoria, los aspectos de la presente invención se describen con referencia a ilustraciones de diagramas de flujo y/o diagramas de bloques de procedimientos, aparatos (sistemas) y productos de programa de ordenador según las realizaciones de la invención. Se entenderá que cada bloque de las ilustraciones de diagramas de flujo y/o diagramas de bloques, y las combinaciones de bloques en las ilustraciones de diagramas de flujo y/o diagramas de bloques, pueden ser implementados por instrucciones de programa de ordenador. Estas instrucciones de programa de ordenador pueden ser proporcionadas a un procesador de un ordenador de propósito general, un ordenador de propósito especial u otro aparato de procesamiento de datos programable para producir una máquina, de manera que las instrucciones, que son ejecutadas por el procesador del ordenador u otro aparato programable de procesamiento de datos, crean medios para implementar las funciones/acciones especificadas en el diagrama de flujo y/o un bloque o bloques del diagrama de bloques.

Estas instrucciones de programa informático pueden ser almacenadas también en un medio legible por ordenador que puede dirigir un ordenador, otros aparatos programables de procesamiento de datos u otros dispositivos para que funcionen de una manera particular, de manera que las instrucciones almacenadas en el medio legible por ordenador produzcan un artículo de fabricación que incluye instrucciones que implementan la función/acción especificada en el diagrama de flujo y/o bloque o bloques del diagrama de bloques.

Las instrucciones de programa de ordenador pueden ser cargadas también en un ordenador, otro aparato de procesamiento de datos programable u otros dispositivos para causar una serie de etapas operacionales a ser realizadas en el ordenador, otro aparato programable u otros dispositivos para producir un procedimiento implementado por ordenador, de manera que las instrucciones que se ejecutan en el ordenador u otro aparato programable proporcionan procedimientos para implementar las funciones/acciones especificadas en el diagrama de flujo y/o bloque o bloques del diagrama de bloques.

El diagrama de flujo y los diagramas de bloques en las figuras ilustran la arquitectura, la funcionalidad y el funcionamiento de posibles implementaciones de sistemas, procedimientos y productos de programa de ordenador según diversas realizaciones de la presente invención. En este sentido, cada bloque en el diagrama de flujo o en los diagramas de bloques puede representar un módulo, segmento o parte de código, que comprende una o más instrucciones ejecutables para implementar la función lógica o funciones lógicas especificadas. Cabe señalar también que, en algunas implementaciones alternativas, las funciones indicadas en el bloque pueden realizarse en un orden diferente al indicado en las figuras. Por ejemplo, dos bloques mostrados en sucesión pueden ser ejecutados, de hecho, de manera sustancialmente simultánea, o algunas veces, los bloques pueden ser ejecutados en orden inverso, dependiendo de la funcionalidad implicada. Se observará también que cada bloque de los diagramas de bloques y/o la ilustración de diagrama de flujo, y las combinaciones de bloques en los diagramas de bloques y/o la ilustración de diagrama de flujo, puede ser implementado por sistemas basados en hardware, de propósito especial, que realizan las funciones o acciones especificadas, o combinaciones de hardware de propósito especial e instrucciones de ordenador.

Además de lo indicado anteriormente, uno o más aspectos de la presente invención pueden ser proporcionados, ofrecidos, desplegados, gestionados, servidos, etc., por un proveedor de servicios que ofrece la gestión de los entornos de cliente. Por ejemplo, el proveedor de servicios puede crear, mantener, soportar, etc., código de ordenador y/o una infraestructura de ordenador que realiza uno o más aspectos de la presente invención para uno o más clientes. A cambio, el proveedor de servicios puede recibir el pago del cliente en virtud de un acuerdo de suscripción y/o tarifa, como ejemplos. De manera adicional o alternativa, el proveedor de servicios puede recibir el pago por la venta de contenidos publicitarios a una o más terceras partes.

En un aspecto de la presente invención, puede desplegarse una aplicación para realizar uno o más aspectos de la presente invención. Como un ejemplo, el despliegue de una aplicación comprende proporcionar una infraestructura de ordenador operable para realizar uno o más aspectos de la presente invención.

Como un aspecto adicional de la presente invención, puede desplegarse una infraestructura de computación que comprende la integración de código legible por ordenador en un sistema de ordenador, en el que el código en

combinación con el sistema de ordenador es capaz de realizar uno o más aspectos de la presente invención.

Como todavía un aspecto adicional de la presente invención, puede proporcionarse un procedimiento para la integración de una infraestructura de computación que comprende la integración de código legible por ordenador en un sistema de ordenador. El sistema de ordenador comprende un medio legible por ordenador, en el que el medio de ordenador comprende uno o más aspectos de la presente invención. El código, en combinación con el sistema de ordenador, es capaz de realizar uno o más aspectos de la presente invención.

Aunque anteriormente se han descrito diversas realizaciones, estas son sólo ejemplos. Por ejemplo, los entornos informáticos de otras arquitecturas pueden incorporar y usar uno o más aspectos de la presente invención. Como ejemplos, servidores distintos de los servidores System z[®], tales como los servidores Power Systems u otros servidores ofrecidos por International Business Machines Corporation, o servidores de otras compañías, pueden incluir, usar y/o beneficiarse de uno o más aspectos de la presente invención. Además, aunque en los ejemplos de la presente memoria, los adaptadores y el concentrador PCI se consideran como una parte del servidor, en otras realizaciones, no tienen que ser considerados necesariamente como una parte del servidor, sino que puede considerarse que están simplemente acoplados a la memoria del sistema y/u otros componentes de un entorno informático. El entorno informático no necesita ser un servidor. Además, aunque se describen tablas, puede usarse cualquier estructura de datos y el término tabla incluye la totalidad de dichas estructuras de datos. Además, aunque los adaptadores están basados en PCI, uno o más aspectos de la presente invención pueden ser usados con otros adaptadores u otros componentes de E/S. El adaptador y el adaptador PCI son sólo ejemplos. Además, el FTE o los parámetros del FTE pueden estar situados y mantenidos en una ubicación diferente a una memoria segura, incluyendo, por ejemplo, en hardware (por ejemplo, función PCI hardware). El DTE, el FTE y/o el descriptor pueden incluir más, menos información o diferente información, así como cualquiera de las instrucciones o los campos de instrucción. Muchas otras variaciones son posibles.

Además, otros tipos de entornos informáticos pueden beneficiarse de uno o más aspectos de la presente invención. Como ejemplo, puede usarse un sistema de procesamiento de datos adecuado para almacenar y/o ejecutar código de programa, que incluye al menos dos procesadores acoplados directa o indirectamente a elementos de memoria a través de un bus del sistema. Los elementos de memoria incluyen, por ejemplo, la memoria local empleada durante la ejecución real del código de programa, el almacenamiento principal y la memoria caché que proporciona almacenamiento temporal de al menos parte del código de programa con el fin de reducir el número de veces que el código debe ser recuperado desde el almacenamiento principal durante la ejecución.

Las entradas/salidas o los dispositivos de E/S (incluyendo, pero sin limitarse a, teclados, monitores, dispositivos de señalamiento, DASD, cinta, CD, DVD, memorias USB y otros medios de memoria, etc.) pueden ser acoplados al sistema, directamente o a través de controladores de E/S intermedios. También pueden acoplarse adaptadores de red al sistema para permitir que el sistema de procesamiento de datos sea acoplado a otros sistemas de procesamiento de datos o impresoras o dispositivos de almacenamiento remotos a través de redes privadas o públicas intermedias. Los módems, los cable módems y las tarjetas de red son sólo algunos de los tipos de adaptadores de red disponibles.

Con referencia a la Fig. 11, en la misma se representan los componentes representativos de un sistema informático servidor para implementar uno o más aspectos de la presente invención. El ordenador servidor representativo comprende una o más CPUs 5001 en comunicación con una memoria 5002 de ordenador (es decir, el almacenamiento central), así como interfaces de E/S a dispositivos 5011 de medios de almacenamiento y redes 5010 para comunicarse con otros ordenadores o SAN y similares. La CPU 5001 es compatible con una arquitectura que tiene un conjunto de instrucciones diseñado y funcionalidad diseñada. La CPU 5001 puede disponer de traducción dinámica de direcciones (DAT) 5003 para transformar las direcciones de programa (direcciones virtuales) en direcciones reales de memoria. Típicamente, una DAT incluye una memoria intermedia de traducción de direcciones de memoria (Translation Lookaside Buffer, TLB) 5007 para almacenar en caché las traducciones para que más tarde los accesos al bloque de la memoria 5002 de ordenador no requieran del retraso de la traducción de direcciones. Típicamente, se emplea una caché 5009 entre la memoria 5002 de ordenador y el procesador 5001. La caché 5009 puede ser jerárquica, con un gran caché disponible para más de una CPU y cachés más pequeñas, más rápidas (de nivel inferior) entre la gran caché y cada CPU. En algunas implementaciones, las memorias caché de nivel inferior se dividen para proporcionar cachés de bajo nivel separadas para recuperación de instrucciones y accesos a datos. En una realización, una instrucción es recuperada desde la memoria 5002 por una unidad 5004 de extracción de instrucciones a través de una memoria caché 5009. La instrucción es decodificada en una unidad 5006 de decodificación de instrucciones y es enviada (en algunas realizaciones con otras instrucciones) a la unidad o unidades 5008 de ejecución de instrucciones. Típicamente, se emplean varias unidades 5008 de ejecución, por ejemplo, una unidad de ejecución aritmética, una unidad de ejecución en coma flotante y una unidad de ejecución de instrucciones de bifurcación. La instrucción es

ejecutada por la unidad de ejecución, accediendo a los operandos en los registros o la memoria especificados por la instrucción, según sea necesario. Si un operando debe ser accedido (cargado o almacenado) en la memoria 5002, típicamente, una unidad 5005 de carga/almacenamiento gestiona el acceso bajo el control de la instrucción que está siendo ejecutada. Las instrucciones pueden ser ejecutadas en circuitos de hardware o en microcódigo interno (firmware) o una combinación de ambos.

Tal como se ha indicado, un sistema de ordenador incluye información en el almacenamiento local (o principal), así como un registro de direccionamiento, protección y referencia y cambio. Algunos aspectos del direccionamiento incluyen el formato de las direcciones, el concepto de espacios de direcciones, los diversos tipos de direcciones y la manera en la que un tipo de dirección es traducida a otro tipo de dirección. Parte del almacenamiento principal incluye ubicaciones de almacenamiento asignadas de manera permanente. El almacenamiento principal proporciona al sistema un almacenamiento de datos de acceso rápido, direccionable directamente. Tanto los datos como los programas deben ser cargados en el almacenamiento principal (desde dispositivos de entrada) antes de que puedan ser procesados.

El almacenamiento principal puede incluir uno o más almacenamientos temporales más pequeños, más rápidos, a veces denominados memorias cachés. Típicamente, una memoria caché está asociada físicamente con una CPU o un procesador de E/S. Los efectos, excepto en el rendimiento, de la construcción física y el uso de los distintos medios de almacenamiento no son observables, en general, por el programa.

Pueden mantenerse memorias cachés separadas para instrucciones y para operandos de datos. La información dentro de una memoria caché es mantenida en bytes contiguos en un límite integral denominado un bloque de caché o línea de caché (o línea, para abreviar). Un modelo puede proporcionar una instrucción EXTRACT CACHE ATTRIBUTE que devuelve el tamaño en bytes de una línea de caché. Un modelo puede proporcionar también instrucciones PREFETCH DATA y PREFETCH DATA RELATIVE LONG que llevan a cabo la recuperación previa del almacenamiento a la memoria caché de datos o de instrucciones o la liberación de los datos de la memoria caché.

El almacenamiento es visto como una larga cadena horizontal de bits. Para la mayoría de las operaciones, los accesos al almacenamiento se realizan en una secuencia de izquierda a derecha. La cadena de bits es subdividida en unidades de ocho bits. Una unidad de ocho bits se denomina un byte, que es el bloque de construcción básico de todos los formatos de información. Cada posición de byte en el almacenamiento se identifica mediante un número entero no negativo único, que es la dirección de esa ubicación de byte o, simplemente, la dirección de byte. Las ubicaciones de bytes contiguas tienen direcciones consecutivas, comenzando de 0 a la izquierda y continuando en una secuencia de izquierda a derecha. Las direcciones son números enteros binarios sin signo y son de 24, 31 o 64 bits.

La información es transmitida entre el almacenamiento y una CPU o un subsistema de canal un byte, o un grupo de bytes, cada vez. A menos que se especifique lo contrario, por ejemplo, en la arquitectura z/Architecture®, un grupo de bytes en el almacenamiento es direccionado por el byte más a la izquierda del grupo. El número de bytes en el grupo está especificado implícita o explícitamente por la operación a realizar. Cuando se usa en una operación de la CPU, un grupo de bytes se denomina campo. Dentro de cada grupo de bytes, por ejemplo, en la arquitectura z/Architecture®, los bits están numerados en una secuencia de izquierda a derecha. En la arquitectura z/Architecture®, a veces los bits más a la izquierda se denominan "bits de orden alto" y los bits a la derecha "bits de orden bajo". Sin embargo, los números de bits no son direcciones de almacenamiento. Sólo pueden direccionarse bytes. Para operar sobre los bits individuales de un byte en el almacenamiento, se accede a todo el byte. Los bits en un byte están numerados de 0 a 7, de izquierda a derecha (por ejemplo, en la arquitectura z/Architecture®). Los bits en una dirección pueden estar numerados como 8-31 o 40-63 para las direcciones de 24 bits, o 1-31 o 33-63 para direcciones de 31 bits; se numeran 0-63 para las direcciones de 64 bits. Dentro de cualquier otro formato de longitud fija de múltiples bytes, los bits que componen el formato se numeran consecutivamente empezando desde 0. Para fines de detección de errores y, preferiblemente, para la corrección, uno o más bits de comprobación pueden ser transmitidos con cada byte o con un grupo de bytes. Dichos bits de comprobación son generados automáticamente por la máquina y no pueden ser controlados directamente por el programa. Las capacidades de almacenamiento se expresan en número de bytes. Cuando la longitud de un campo de operando de almacenamiento está implícita en el código de operación de una instrucción, se dice que el campo tiene una longitud fija, que puede ser de uno, dos, cuatro, ocho o dieciséis bytes. Los campos más grandes pueden estar implícitos en algunas instrucciones. Cuando la longitud de un campo de operando de almacenamiento no está implícita, pero se indica explícitamente, se dice que el campo tiene una longitud variable. Los operandos de longitud variable pueden tener una longitud variable con incrementos de un byte (o con algunas instrucciones, en múltiplos de dos bytes u otros múltiplos). Cuando la información es colocada en el almacenamiento, solo se reemplazan los contenidos de aquellas ubicaciones de bytes que están incluidas en el campo designado, a pesar de que la anchura de la ruta física al almacenamiento puede ser mayor que la longitud del campo que está siendo

almacenado.

Ciertas unidades de información deben estar en un límite integral en el almacenamiento. Un límite se denomina integral para una unidad de información cuando su dirección de almacenamiento es un múltiplo de la longitud de la unidad en bytes. Se proporcionan nombres especiales a los campos de 2, 4, 8, y 16 bytes en un límite integral. Una media palabra es un grupo de dos bytes consecutivos en un límite de dos bytes y es el bloque de construcción básico de las instrucciones. Una palabra es un grupo de cuatro bytes consecutivos en un límite de cuatro bytes. Una palabra doble es un grupo de ocho bytes consecutivos en un límite de ocho bytes. Una palabra cuádruple es un grupo de 16 bytes consecutivos en un límite de 16 bytes. Cuando las direcciones de almacenamiento designan medias palabras, palabras, palabras dobles y palabras cuádruples, la representación binaria de la dirección contiene uno, dos, tres o cuatro bits cero a la derecha, respectivamente. Las instrucciones deben estar en límites integrales de dos bytes. Los operandos de almacenamiento de la mayoría de las instrucciones no tienen requisitos de límite-alineación.

En los dispositivos que implementan cachés separadas para instrucciones y operandos de datos, puede experimentarse un retraso considerable si el programa se almacena en una línea de caché desde la cual las instrucciones son recuperadas posteriormente, independientemente de si el almacenamiento altera las instrucciones que son recuperadas posteriormente.

En una realización, la invención puede ponerse en práctica mediante software (a veces denominado código interno con licencia, firmware, microcódigo, milicódigo, picocódigo y similares, cualquiera de los cuales sería coherente con la presente invención). Con referencia a la Fig. 8, el código de programa de software que incorpora la presente invención es accedido, típicamente, por el procesador 5001 del sistema 5000 servidor desde los dispositivos 5011 de medios de almacenamiento permanente, tales como una unidad de CD-ROM, una unidad de cinta o un disco duro. El código de programa de software puede ser incorporado en cualquiera de entre una diversidad de medios conocidos para su uso con un sistema de procesamiento de datos, tal como un disquete, un disco duro o un CD-ROM. El código puede ser distribuido en dichos medios, o puede ser distribuido a los usuarios desde la memoria 5002 del ordenador o el almacenamiento de un sistema de ordenador sobre una red 5010 a otros sistemas de ordenador para su uso por los usuarios de dichos otros sistemas.

El código de programa de software incluye un sistema operativo que controla la función y la interacción de los diversos componentes del ordenador y uno o más programas de aplicación. Normalmente, el código de programa es paginado desde el dispositivo 5011 de medios de almacenamiento al almacenamiento 5002 de ordenador, de velocidad relativamente más alta, donde está disponible para su procesamiento por el procesador 5001. Las técnicas y los procedimientos para incorporar el código de programa de software en memoria, en medios físicos y/o distribuir código de software a través de redes son bien conocidos y no se describirán adicionalmente en la presente memoria. El código del programa, cuando es creado y almacenado en un soporte material (incluyendo, pero sin limitarse a módulos electrónicos de memoria (RAM), memoria flash, discos compactos (CDs), DVDs, cintas magnéticas y similares), se denomina frecuentemente "producto de programa informático". Típicamente, el medio del producto de programa informático es legible por un circuito de procesamiento preferiblemente en un sistema de ordenador para su ejecución por el circuito de procesamiento.

La Fig. 12 ilustra una estación de trabajo o un sistema de hardware servidor representativo en el que se puede practicar la presente invención. El sistema 5020 de la Fig. 12 comprende un sistema 5021 de ordenador base representativo, tal como un ordenador personal, una estación de trabajo o un servidor, que incluye dispositivos periféricos opcionales. El sistema 5021 de ordenador base incluye uno o más procesadores 5026 y un bus empleado para conectar y habilitar la comunicación entre el procesador o procesadores 5026 y los otros componentes del sistema 5021, según técnicas conocidas. El bus conecta el procesador 5026 con la memoria 5025 y el almacenamiento 5027 permanente que puede incluir una unidad de disco duro (incluyendo cualquiera de entre soportes magnéticos, CD, DVD y memoria flash, por ejemplo) o una unidad de cinta, por ejemplo. El sistema 5021 podría incluir también un adaptador de interfaz de usuario, que conecta el microprocesador 5026 a través del bus a uno o más dispositivos de interfaz, tales como un teclado 5024, un ratón 5023, una impresora/escáner 5030 y/u otros dispositivos de interfaz, que pueden ser cualquier dispositivo de interfaz de usuario, tal como una pantalla táctil, teclado de entrada digitalizado, etc. El bus conecta también un dispositivo 5022 de visualización, tal como una pantalla LCD o monitor, al microprocesador 5026 a través de un adaptador de pantalla.

El sistema 5021 puede comunicarse con otros ordenadores o redes de ordenadores por medio de un adaptador de red capaz de comunicarse 5028 con una red 5029. Los adaptadores de red ejemplares son canales de comunicación, Token Ring, Ethernet o módems. De manera alternativa, el sistema 5021 puede comunicarse usando una interfaz inalámbrica, tal como una tarjeta CDPD (datos por paquetes digitales celulares, Cellular Digital Packet Data). El sistema 5021 puede estar asociado con dichos otros ordenadores en una red de área local (LAN) o una red de área amplia (WAN), o el sistema 5021 puede ser un cliente en un acuerdo cliente/servidor con otro

ordenador, etc. Todas estas configuraciones, así como el hardware y el software de comunicaciones apropiado, son conocidas en la técnica.

La Fig. 13 ilustra una red 5040 de procesamiento de datos en la que puede ponerse en práctica la presente invención. La red 5040 de procesamiento de datos puede incluir una pluralidad de redes individuales, tales como una red inalámbrica y una red cableada, cada una de las cuales puede incluir una pluralidad de estaciones 5041, 5042, 5043, 5044 de trabajo individuales. Además, tal como apreciarán las personas con conocimientos en la materia, pueden incluirse una o más LANs, donde una LAN puede comprender una pluralidad de estaciones de trabajo inteligentes acopladas a un procesador servidor.

Todavía con referencia a la Fig. 13, las redes pueden incluir también ordenadores o servidores "mainframe", tales como un ordenador pasarela (servidor de cliente 5046) o servidor de aplicaciones (servidor 5048 remoto que puede acceder a un repositorio de datos y puede ser accedido también directamente desde una estación 5045 de trabajo). Un ordenador 5046 pasarela sirve como un punto de entrada a cada red individual. Se necesita una pasarela cuando se conecta un protocolo de red a otro. La pasarela 5046 puede estar acoplada preferiblemente a otra red (Internet 5047, por ejemplo) por medio de un enlace de comunicaciones. La pasarela 5046 puede estar acoplada también directamente a una o más estaciones 5041, 5042, 5043, 5044 de trabajo usando un enlace de comunicaciones. El ordenador pasarela puede estar implementado usando un servidor IBM eServer™ System z® disponible en International Business Machines Corporation.

Con referencia al mismo tiempo a la Fig. 12 y la FIG. 13, el código de programación de software que puede materializar la presente invención puede ser accedido por el procesador 5026 del sistema 5020 desde los medios 5027 de almacenamiento permanente, tales como una unidad de CD-ROM o disco duro. El código de programación de software puede estar materializado en cualquiera de entre una diversidad de medios conocidos para su uso con un sistema de procesamiento de datos, tales como un disquete, disco duro o CD-ROM. El código puede ser distribuido en dichos medios, o puede ser distribuido a los usuarios 5050, 5051 desde la memoria o el almacenamiento de un sistema de ordenador a través de una red a otros sistemas de ordenador para uso por los usuarios de dichos otros sistemas.

De manera alternativa, el código de programación puede ser materializado en la memoria 5025, y puede ser accedido por el procesador 5026 usando el bus del procesador. Dicho código de programación incluye un sistema operativo que controla la función y la interacción de los diversos componentes informáticos y uno o más programas 5032 de aplicación. Normalmente, el código del programa es paginado desde los medios 5027 de almacenamiento a la memoria 5025 de alta velocidad, donde está disponible para su procesamiento por el procesador 5026. Las técnicas y los procedimientos para materializar un código de programación de software en una memoria, en medios físicos y/o para distribuir código de software a través de redes son bien conocidos y no se describirán adicionalmente en la presente memoria. El código del programa, cuando se crea y se almacena en un soporte material (incluyendo, pero sin limitarse a módulos electrónicos de memoria (RAM), memoria flash, discos compactos (CDs), DVDs, cintas magnéticas y similares), se denomina frecuentemente "producto de programa informático". Típicamente, el medio de producto de programa informático es legible por un circuito de procesamiento, preferiblemente en un sistema de ordenador para su ejecución por el circuito de procesamiento.

La memoria caché que está más fácilmente disponible para el procesador (normalmente más rápida y más pequeña que otras memorias caché del procesador) es la memoria caché más baja (L1 o nivel uno) y el almacenamiento principal (memoria principal) es la memoria caché de nivel más alto (L3, si hay 3 niveles). Frecuentemente, la memoria caché de nivel más bajo se divide en una memoria caché de instrucciones (I-Caché) que contiene instrucciones máquina a ser ejecutadas y una memoria caché de datos (D-Caché) que contiene operandos de datos.

Con referencia a la Fig. 14, en la misma se representa una realización ejemplar de un procesador para el procesador 5026. Típicamente, se emplean uno o más niveles de caché 5053 para almacenar temporalmente bloques de memoria con el fin de mejorar el rendimiento del procesador. La memoria 5053 caché es una memoria intermedia de alta velocidad que contiene líneas de caché de datos de memoria que tienen una alta probabilidad de ser usadas. Las líneas de caché típicas son 64, 128 o 256 bytes de datos de memoria. Frecuentemente, se emplean cachés separadas para almacenar/recuperar las instrucciones y para almacenar/recuperar los datos. La coherencia de caché (sincronización de las copias de las líneas en la memoria y en las cachés) es proporcionada frecuentemente por varios algoritmos "snoop" bien conocidos en la técnica. El almacenamiento 5025 de la memoria principal de un sistema de ordenador se denomina frecuentemente memoria caché. En un sistema de ordenador que tiene 4 niveles de caché 5053, el almacenamiento 5025 principal se denomina a veces caché de nivel 5 (L5) ya que, típicamente, es más rápido y sólo mantiene una parte de la memoria no volátil (DASD, cinta, etc.) que está disponible para un sistema de ordenador. El almacenamiento 5025 principal almacena en "caché" páginas de datos paginados a y desde la memoria 5025 principal por el sistema operativo.

Un contador 5061 de programa (contador de instrucciones) realiza un seguimiento de la dirección de la instrucción actual a ser ejecutada. Un contador de programa en un procesador z/Architecture[®] es de 64 bits y puede ser truncado a 31 o 24 bits para soportar los límites de direccionamiento anteriores. Típicamente, un contador de programa es materializado en una palabra de estado de programa PSW (Program Status Word) de un ordenador de manera que persiste durante el cambio de contexto. De esta manera, un programa en curso, que tiene un valor de contador de programa, puede ser interrumpido, por ejemplo, por el sistema operativo (cambio de contexto desde el entorno de programa al entorno del sistema operativo). La PSW del programa mantiene el valor de contador de programa mientras que el programa no está activo, y se usa el contador de programa (en la PSW) del sistema operativo mientras se está ejecutando el sistema operativo. Típicamente, el contador de programa se incrementa en una cantidad igual al número de bytes de la instrucción en curso. Típicamente, las instrucciones RISC (computación de conjunto reducido de instrucciones, Reduced Instruction Set Computing) son de longitud fija, mientras que típicamente las instrucciones CISC (computación de conjunto de instrucciones complejas, Complex Instruction Set Computing) son de longitud variable. Las instrucciones de arquitectura IBM z/Architecture[®] son instrucciones CISC que tienen una longitud de 2, 4 o 6 bytes. El contador 5061 de programa es modificado por una operación de cambio de contexto o una operación de bifurcación tomada de una instrucción de bifurcación, por ejemplo. En una operación de cambio de contexto, el valor actual del contador de programa es salvado en la palabra de estado de programa, junto con otra información de estado acerca del programa que se está ejecutando (por ejemplo, los códigos de condición), y se carga un nuevo valor del contador de programa que apunta a una instrucción de un nuevo módulo de programa a ser ejecutado. Una operación de bifurcación tomada se realiza con el fin de permitir que el programa tome decisiones o realice bucles dentro del programa cargando el resultado de la instrucción de bifurcación en el contador 5061 de programa.

Típicamente, se emplea una unidad 5055 de recuperación de instrucciones para recuperar instrucciones en nombre del procesador 5026. La unidad de recuperación recupera "las siguientes instrucciones secuenciales", instrucciones de destino de las instrucciones de bifurcación tomada, o las primeras instrucciones de un programa después de un cambio de contexto. Las unidades modernas de recuperación de instrucciones emplean frecuentemente técnicas de recuperación previa para recuperar instrucciones, de manera especulativa, en base a la probabilidad de uso de las instrucciones recuperadas previamente. Por ejemplo, una unidad de recuperación puede recuperar 16 bytes de instrucción que incluye la siguiente instrucción secuencial y bytes adicionales de instrucciones secuenciales adicionales.

A continuación, las instrucciones recuperadas son ejecutadas por el procesador 5026. En una realización, la instrucción recuperada o las instrucciones recuperadas son pasadas a una unidad 5056 de envío de la unidad de recuperación. La unidad de envío decodifica la instrucción o las instrucciones y reenvía información acerca de la instrucción decodificada o instrucciones decodificadas a las unidades 5057, 5058, 5060 apropiadas. Típicamente, una unidad 5057 de ejecución recibirá información acerca de las instrucciones aritméticas decodificadas desde la unidad 5055 de recuperación de instrucciones y realizará operaciones aritméticas sobre los operandos según el código de operación de la instrucción. Preferiblemente, los operandos son proporcionados a la unidad 5057 de ejecución desde la memoria 5025, registros 5059 diseñados o desde un campo inmediato de la instrucción que está siendo ejecutada. Los resultados de la ejecución, cuando se almacenan, se almacenan en la memoria 5025, los registros 5059 o en otro hardware de la máquina (por ejemplo, registros de control, registros de PSW y similares).

Típicamente, un procesador 5026 tiene una o más unidades 5057, 5058, 5060 para ejecutar la función de la instrucción. Con referencia a la Fig. 15A, una unidad 5057 de ejecución puede comunicarse con registros 5059 generales diseñados, una unidad 5056 de decodificación/envío, una unidad 5060 de almacenamiento y carga, y otras unidades 5065 procesadoras por medio de una lógica 5071 de interfaz. Una unidad 5057 de ejecución puede emplear diversos circuitos 5067, 5068, 5069 de registro para contener información con la que operará la unidad lógica aritmética (ALU) 5066. La ALU realiza operaciones aritméticas, tales como suma, resta, multiplicación y división, así como funciones lógicas, tales como AND, OR y OR-exclusivo (XOR), rotación y desplazamiento. Preferiblemente, la ALU soporta operaciones especializadas que dependen del diseño. Otros circuitos pueden proporcionar otras funciones 5072 de diseño, incluyendo códigos de condición y lógica de soporte de recuperación, por ejemplo. Típicamente, el resultado de una operación ALU se mantiene en un circuito 5070 de registro de salida que puede reenviar el resultado a una diversidad de otras funciones de procesamiento. Hay muchas disposiciones de unidades de procesador, la presente descripción sólo pretende proporcionar una comprensión representativa de una realización.

Una instrucción ADD, por ejemplo, sería ejecutada en una unidad 5057 de ejecución que tiene la funcionalidad aritmética y lógica, mientras que una instrucción de coma flotante, por ejemplo, sería ejecutada en una unidad de ejecución en coma flotante con capacidades de coma flotante especializadas. Preferiblemente, una unidad de ejecución opera sobre operandos identificados por una instrucción realizando una función definida por el código de operación sobre los operandos. Por ejemplo, una instrucción ADD puede ser ejecutada por una unidad 5057 de

ejecución sobre operandos que se encuentran en dos registros 5059 identificados por los campos de registro de la instrucción.

La unidad 5057 de ejecución realiza la suma aritmética sobre dos operandos y almacena el resultado en un tercer operando, en el que el tercer operando puede ser un tercer registro o uno de los dos registros de origen. Preferiblemente, la unidad de ejecución usa una unidad lógica aritmética (ALU) 5066 que es capaz de realizar una diversidad de funciones lógicas, tales como desplazamiento, rotación, AND, OR y XOR, así como una diversidad de funciones algebraicas, incluyendo cualquiera de entre suma, resta, multiplicación, división. Algunas ALU 5066 están diseñadas para operaciones escalares y algunas para operaciones en coma flotante. Los datos pueden ser Big Endian (donde el byte menos significativo está en la dirección de byte más alta) o Little Endian (donde el byte menos significativo está en la dirección de byte más baja) dependiendo de la arquitectura. La arquitectura IBM z/Architecture® es Big Endian. Los campos con signo pueden ser signo y magnitud, complemento a 1 o complemento a 2 en función de la arquitectura. Un número complemento a 2 es ventajoso en el sentido de que la ALU no necesita disponer de una capacidad de resta ya que un valor negativo o un valor positivo en complemento a 2 sólo requiere una adición dentro de la ALU. Los números se describen comúnmente en forma abreviada, donde un campo de 12 bits define una dirección de un bloque de 4096 bytes y se describe comúnmente como un bloque de 4 Kbytes (Kilobyte), por ejemplo.

Con referencia a la Fig. 15B, la información de instrucción de bifurcación para la ejecución de una instrucción de bifurcación es enviada típicamente a una unidad 5058 de bifurcación que emplea frecuentemente un algoritmo de predicción de bifurcación, tal como una tabla 5082 de historial de bifurcaciones para predecir el resultado de la bifurcación antes de que se hayan completado otras operaciones condicionales. El objetivo de la instrucción de bifurcación actual será recuperado y será ejecutado, de manera especulativa, antes de que las operaciones condicionales estén completadas. Una vez completadas las operaciones condicionales, las instrucciones de bifurcación ejecutadas de manera especulativa son completadas o desechadas en base a las condiciones de la operación condicional y el resultado especulado. Una instrucción de bifurcación típica puede comprobar los códigos de condición y saltar a una dirección de destino si los códigos de condición cumplen el requisito de bifurcación de la instrucción de bifurcación, una dirección de destino puede ser calculada en base a diversos números, incluyendo los que se encuentran en los campos de registro o un campo inmediato de la instrucción, por ejemplo. La unidad 5058 de bifurcación puede emplear una ALU 5074 que tiene una pluralidad de circuitos 5075, 5076, 5077 de registro de entrada y un circuito 5080 de registro de salida. La unidad 5058 de bifurcación puede comunicarse con los registros 5059 generales, la unidad 5056 de envío y decodificación u otros circuitos 5073, por ejemplo.

La ejecución de un grupo de instrucciones puede ser interrumpida por una diversidad de razones, incluyendo un cambio de contexto iniciado por un sistema operativo, una excepción de programa o un error que causa un cambio de contexto, una señal interrupción de E/S que causa un cambio de contexto o actividad multi-hilo de una pluralidad de programas (en un entorno multi-hilo), por ejemplo. Preferiblemente una acción de cambio de contexto salva la información de estado acerca de un programa que se está ejecutando actualmente y, a continuación, carga la información de estado acerca de otro programa que está siendo invocado. La información de estado puede ser salvada en registros de hardware o en memoria, por ejemplo. Preferiblemente, la información de estado comprende un valor de contador de programa que apunta a una próxima instrucción a ejecutar, códigos de condición, información de traducción de memoria y contenido del registro diseñado. Una actividad de cambio de contexto puede ser realizada por circuitos de hardware, programas de aplicación, programas del sistema operativo o código de firmware (microcódigo, picocódigo o código interno bajo licencia (LIC)) solos o en combinación.

Un procesador accede a los operandos según los procedimientos definidos en la instrucción. La instrucción puede proporcionar un operando inmediato usando el valor de una parte de la instrucción, puede proporcionar uno o más campos de registro que señalan explícitamente a registros de propósito general o a registros de propósito especial (registros de coma flotante, por ejemplo). La instrucción puede utilizar registros implícitos identificados por un campo de código de operación como operandos. La instrucción puede utilizar posiciones de memoria para operandos. Una ubicación de memoria de un operando puede ser proporcionada por un registro, un campo inmediato, o una combinación de registros y campo inmediato, tal como ejemplifica la función de desplazamiento largo de la arquitectura z/Architecture®, en la que la instrucción define un registro base, un registro índice y un campo inmediato (campo de desplazamiento) que se suman para proporcionar la dirección del operando en la memoria, por ejemplo. En la presente memoria, la ubicación implica típicamente una ubicación en la memoria principal (memoria principal), a menos que se indique lo contrario.

Con referencia a la Fig. 15C, un procesador accede al almacenamiento usando una unidad 5060 de carga/almacenamiento. La unidad 5060 de carga/almacenamiento puede realizar una operación de carga obteniendo la dirección del operando de destino en la memoria 5053 y cargando el operando en un registro 5059 u otra ubicación 5053 de memoria, o puede realizar una operación de almacenamiento obteniendo la dirección del

operando de destino en la memoria 5053 y almacenamiento los datos obtenidos desde un registro 5059 u otra ubicación 5053 de memoria en la ubicación del operando de destino en la memoria 5053. La unidad 5060 de carga/almacenamiento puede ser especulativa y puede acceder a la memoria en una secuencia con un orden diferente con respecto a la secuencia de instrucciones, sin embargo la unidad 5060 de carga/almacenamiento debe hacer creer a los programas que las instrucciones se ejecutan en orden. Una unidad 5060 de carga/almacenamiento puede comunicarse con registros 5059 generales, la unidad 5056 de decodificación/envío, la interfaz 5053 cache/memoria u otros elementos 5083 y comprende diversos circuitos de registro, ALUs 5085 y lógica 5090 de control para calcular direcciones de almacenamiento y proporcionar secuenciación "pipeline" para mantener el orden de las operaciones. Algunas operaciones pueden estar desordenadas, pero la unidad de carga/almacenamiento proporciona una funcionalidad para hacer que el programa crea que las operaciones han sido realizadas en orden, tal como es bien conocido en la técnica.

Preferiblemente, las direcciones que un programa de aplicación "ve" se denominan frecuentemente direcciones virtuales. A veces, las direcciones virtuales se denominan "direcciones lógicas" y "direcciones efectivas". Estas direcciones virtuales son virtuales en el sentido de que son redirigidas a una ubicación de memoria física por una de entre una diversidad de tecnologías de traducción dinámica de direcciones (DAT), incluyendo, pero sin limitarse a, anteponiendo simplemente a una dirección virtual un valor de desplazamiento, traduciendo la dirección virtual con una o más tablas de traducción, en el que las tablas de traducción comprenden preferiblemente al menos una tabla de segmentos y una tabla de páginas, solas o en combinación, en el que, preferiblemente, la tabla de segmentos tiene una entrada que apunta a la tabla de páginas. En la arquitectura z/Architecture®, se proporciona una jerarquía de traducción que incluye una primera tabla de regiones, una segunda tabla de regiones, una tercera tabla de regiones, una tabla de segmentos y una tabla de páginas opcional. El rendimiento de la traducción de direcciones se mejora frecuentemente utilizando una memoria temporal de traducción de direcciones (TLB) que comprende entradas que asignan una dirección virtual a una ubicación de memoria física asociada. Las entradas se crean cuando el DAT traduce una dirección virtual usando las tablas de traducción. El uso posterior de la dirección virtual puede utilizar, a continuación, la entrada de la TLB rápida en lugar de los accesos secuenciales lentos a la tabla de traducción. El contenido TLB puede ser gestionado por una diversidad de algoritmos de sustitución, incluyendo LRU (Least Recently Used, usado menos recientemente).

En el caso en el que el procesador es un procesador de un sistema multi-procesador, cada procesador tiene la responsabilidad de mantener los recursos compartidos, tales como E/S, memorias caché, TLBs y memoria, interbloqueados para asegurar la coherencia. Típicamente, se utilizarán tecnologías "snoop" para mantener la coherencia de la caché. En un entorno "snoop", cada línea de caché puede ser marcada como en uno cualquiera de entre un estado compartido, un estado exclusivo, un estado cambiado, un estado no válido y similares, con el fin de facilitar la compartición.

Las unidades 5054 de E/S (Fig. 14) proporcionan al procesador medios para conectarse a los dispositivos periféricos, incluyendo cintas, discos, impresoras, pantallas y redes, por ejemplo. Las unidades de E/S son presentadas frecuentemente al programa de ordenador por medio de controladores de software. En "mainframes", tales como el sistema System z® de IBM®, los adaptadores de canal y los adaptadores de sistemas abiertos son unidades de E/S del "mainframe" que proporcionan las comunicaciones entre el sistema operativo y los dispositivos periféricos.

Además, otros tipos de entornos informáticos pueden beneficiarse de uno o más aspectos de la presente invención. Como un ejemplo, un entorno puede incluir un emulador (por ejemplo, software u otros mecanismos de emulación), en el que una arquitectura particular (incluyendo, por ejemplo, ejecución de instrucciones, funciones diseñadas, tales como traducción de direcciones, y registros diseñados) o un subconjunto de la misma es emulada (por ejemplo, en un sistema de ordenador nativo que tiene un procesador y memoria). En dicho entorno, una o más funciones de emulación del emulador pueden implementar uno o más aspectos de la presente invención, incluso si un ordenador que ejecuta el emulador puede tener una arquitectura diferente a las capacidades que se emulan. Como un ejemplo, en modo emulación, la instrucción u operación específica que se está emulado es decodificada, y una función de emulación apropiada es construida para implementar la instrucción u operación individual.

En un entorno de emulación, un ordenador central incluye, por ejemplo, una memoria para almacenar instrucciones y datos; una unidad de recuperación de instrucciones para recuperar instrucciones desde la memoria y, opcionalmente, para proporcionar un almacenamiento temporal local para la instrucción recuperada; una unidad de decodificación de instrucciones para recibir las instrucciones recuperadas y para determinar el tipo de instrucciones que han sido recuperadas; y una unidad de ejecución de instrucciones para ejecutar las instrucciones. La ejecución puede incluir cargar datos en un registro desde memoria; almacenar datos de nuevo en la memoria desde un registro; o realizar algún tipo de operación aritmética o lógica, según lo determina la unidad de decodificación. En un ejemplo, cada unidad se implementa en software. Por ejemplo, las operaciones realizadas

por las unidades se implementan como una o más subrutinas dentro del software emulador.

Más particularmente, en un "mainframe", los programadores, normalmente programadores en "C" actuales, usan instrucciones máquina diseñadas, frecuentemente por medio de una aplicación de compilador. Estas instrucciones almacenadas en el medio de almacenamiento pueden ser ejecutadas de manera nativa en una arquitectura z/Architecture® IBM® Server o, de manera alternativa, en máquinas que ejecutan otras arquitecturas. Pueden ser emuladas en los servidores mainframe IBM® existentes y futuros y en otros equipos de IBM® (por ejemplo, servidores Power Systems y System x® Servers). Pueden ser ejecutadas en equipos que ejecutan Linux en una amplia diversidad de máquinas que usan hardware fabricado por IBM®, Intel®, AMD™, y otros. Además de la ejecución en ese hardware bajo arquitectura z/Architecture®, puede usarse Linux así como máquinas que usan emulación por Hércules o FSI (Fundamental Software, Inc), donde generalmente la ejecución es en un modo de emulación. En el modo de emulación, el software de emulación es ejecutado por un procesador nativo para emular la arquitectura de un procesador emulado. La información acerca de los productos de emulador indicados anteriormente está disponible en la World Wide Web, respectivamente, en www.hercules-390.org y www.funsoft.com.

Típicamente, el procesador nativo ejecuta un software de emulación que comprende firmware o un sistema operativo nativo para realizar la emulación del procesador emulado. El software de emulación es responsable de recuperar y ejecutar las instrucciones de la arquitectura del procesador emulado. El software de emulación mantiene un contador de programa emulado para realizar un seguimiento de los límites de las instrucciones. El software de emulación puede recuperar una o más instrucciones máquina emuladas a la vez y convertir la una o más instrucciones máquina emuladas a un grupo correspondiente de instrucciones máquina nativas para su ejecución por el procesador nativo. Estas instrucciones convertidas pueden ser almacenadas en caché de manera que pueda conseguirse una conversión más rápida. No obstante, el software de emulación debe mantener las reglas de arquitectura de la arquitectura del procesador emulado para asegurar que los sistemas operativos y las aplicaciones escritas para el procesador emulado funcionan correctamente. Además, el software de emulación debe proporcionar los recursos identificados por la arquitectura del procesador emulado incluyendo, pero sin limitarse a, registros de control, registros de propósito general, registros de coma flotante, función dinámica de traducción de direcciones que incluyen tablas de segmentos y tablas de páginas, por ejemplo, mecanismos de interrupción, mecanismos de cambio de contexto, los relojes hora del día (TOD, Time Of Day) e interfaces diseñados a los subsistemas de E/S de manera que un sistema operativo o un programa de aplicación, diseñado para ejecutarse en el procesador emulado, pueda ser ejecutado en el procesador nativo que tiene el software de emulación.

Una instrucción específica que está siendo emulada es decodificada, y una subrutina es llamada para realizar la función de la instrucción individual. Una función de software de emulación que emula una función de un procesador emulado es implementada, por ejemplo, en una subrutina "C" o controlador, o algún otro procedimiento para proporcionar un controlador para el hardware específico, tal como estará incluido en las capacidades de las personas con conocimientos en la materia tras comprender la descripción de la realización preferida. Diversas patentes de emulación de software y hardware incluyendo, pero sin limitarse a la patente US N° 5.551.013, titulada "Multiprocessor for Hardware Emulation", de Beausoleil et al.; y la patente US N° 6.009.261, titulada "Preprocessing of Stored Target Routines for Emulating Incompatible Instructions on a Target Processor", de Scalzi et al; y la patente US N° 5.574.873, titulada "Decoding Guest Instruction to Directly Access Emulation Routines that Emulate the Guest Instructions", de Davidian et al; y la patente US N° 6.308.255, titulada "Symmetrical Multiprocessing Bus and Chipset Used for Coprocessor Support Allowing Non-Native Code to Run in a System", de Gorishek et al; y la patente US N° 6.463.582, titulada "Dynamic Optimizing Object Code Translator for Architecture Emulation and Dynamic Optimizing Object Code Translation Method", de Lethin et al; y la patente US N° 5.790.825, titulada "Method for Emulating Guest Instructions on a Host Computer Through Dynamic Recompile of Host Instructions", de Eric Traut, y muchas otras, ilustran una diversidad de maneras conocidas para conseguir la emulación de un formato de instrucción diseñado para una máquina diferente para una máquina objetivo disponible para las personas con conocimientos en la materia.

En la Fig. 16, se proporciona un ejemplo de un sistema 5092 de ordenador servidor emulado que emula un sistema 5000' de ordenador servidor de una arquitectura servidor. En el sistema 5092 de ordenador servidor emulado, el procesador central (CPU) 5091 es un procesador servidor emulado (o procesador servidor virtual) y comprende un procesador 5093 de emulación que tiene una arquitectura de conjunto de instrucciones nativas diferente a la del procesador 5091 del ordenador 5000' servidor. El sistema 5092 de ordenador servidor emulado tiene una memoria 5094 accesible por el procesador 5093 de emulación. En la realización ejemplar, la memoria 5094 es dividida en una parte de memoria 5096 del ordenador servidor y una parte 5097 de rutinas de emulación. La memoria 5096 del ordenador servidor está disponible para los programas del ordenador 5092 servidor emulado según la arquitectura de ordenador servidor. El procesador 5093 de emulación ejecuta instrucciones nativas de un conjunto de instrucciones diseñadas de una arquitectura distinta de la del procesador 5091 emulado, las

instrucciones nativas obtenidas desde la memoria 5097 de rutinas de emulación, y pueden acceder a una instrucción de servidor para su ejecución desde un programa en la memoria 5096 del ordenador servidor empleando una o más instrucciones obtenidas en una rutina de secuencia y acceso/decodificación que puede decodificar la instrucción o instrucciones de servidor accedidas para determinar una rutina de ejecución de instrucción nativa para emular la función de la instrucción de servidor accedida. Otras funciones que se definen para la arquitectura de sistema 5000' de ordenador servidor pueden ser emuladas por las rutinas de funciones diseñadas, incluyendo funciones tales como registros de propósito general, registros de control, traducción dinámica de direcciones y soporte de subsistema de E/S y caché del procesador, por ejemplo. Las rutinas de emulación pueden aprovechar también las funciones disponibles en el procesador 5093 de emulación (tales como los registros generales y la traducción dinámica de direcciones virtuales) para mejorar el rendimiento de las rutinas de emulación. También pueden proporcionarse hardware especial y motores de reducción de carga para ayudar al procesador 5093 a emular la función del ordenador 5000' servidor.

La terminología usada en la presente memoria tiene el propósito de describir realizaciones particulares solamente y no pretende ser limitativa de la invención. Tal como se usan en la presente memoria, las formas singulares "un", "una" y "el", "la" pretenden incluir también las formas plurales, a menos que el contexto indique claramente lo contrario. Se entenderá además que los términos "comprende" y/o "que comprende", cuando se usan en la presente memoria descriptiva, especifican la presencia de características, números enteros, etapas, operaciones, elementos y/o componentes indicados, pero no excluyen la presencia o adición de una o más de otras características, números enteros, etapas, operaciones, elementos, componentes y/o grupos de los mismos.

Las correspondientes estructuras, materiales, acciones y equivalentes de todos los medios o etapas más elementos de función en las reivindicaciones siguientes, en su caso, pretenden incluir cualquier estructura, material o acción para realizar la función en combinación con otros elementos reivindicados tal como se reivindica específicamente. La descripción de la presente invención ha sido presentada con propósitos de ilustración y descripción, pero no pretende ser exhaustiva o limitarse a la invención en la forma descrita. Muchas modificaciones y variaciones serán evidentes para las personas con conocimientos en la materia sin apartarse del alcance de la invención. La realización fue elegida y descrita con el fin de explicar de la mejor manera los principios de la invención y la aplicación práctica, y para permitir que otras personas con conocimientos ordinarios en la materia comprendan la invención para diversas realizaciones con diversas modificaciones que sean adecuadas al uso particular contemplado.

REIVINDICACIONES

1. Un procedimiento para ejecutar dentro de un circuito de procesamiento, que comprende un medio de almacenamiento legible por ordenador, legible por dicho circuito de procesamiento e instrucciones de almacenamiento para su ejecución por dicho circuito de procesamiento, una instrucción de almacenamiento que comprende un campo de código de operación que identifica una instrucción de almacenamiento en adaptador, un primer campo que identifica una primera ubicación que incluye los datos a ser almacenados en un adaptador, un segundo campo que identifica una segunda ubicación, cuyos contenidos incluyen un descriptor de función que identifica el adaptador, una designación de un espacio de direcciones dentro del adaptador en el que deben almacenarse los datos, y un desplazamiento dentro del espacio de direcciones, para almacenar datos en un adaptador, en el que el procedimiento comprende las etapas de:
 - 5 obtener una instrucción máquina a ser ejecutada, en el que la instrucción máquina está definida para su ejecución por un ordenador según una arquitectura de ordenador; y ejecutar la instrucción máquina, en el que la ejecución comprende:
 - 15 usar el descriptor de función para obtener una entrada de la tabla de funciones asociada con el adaptador;
 - obtener una dirección de datos del adaptador usando al menos uno de entre la información en la entrada de la tabla de funciones y el desplazamiento; y
 - 20 almacenar los datos desde la primera ubicación en una ubicación específica en el espacio de direcciones identificado por la designación del espacio de direcciones, en el que la ubicación específica es identificada por la dirección de datos del adaptador.
 2. Procedimiento según la reivindicación 1, en el que el espacio de direcciones a acceder es un espacio de memoria o un espacio de E/S, y en el que la obtención de la dirección de datos comprende el uso de uno o más parámetros de la entrada de la tabla de funciones para obtener la dirección de datos.
 3. Procedimiento según la reivindicación 2, en el que el uso de uno o más parámetros comprende sumar un valor de un registro de dirección base de la entrada de la tabla de funciones al desplazamiento para obtener la dirección de datos.
 4. Procedimiento según la reivindicación 1, en el que los contenidos de la segunda ubicación incluyen una cantidad de datos a almacenar.
 5. Un sistema de ordenador para ejecutar una instrucción de almacenamiento para almacenar datos en un adaptador, en el que dicho sistema de ordenador comprende:
 - 30 una memoria;
 - un procesador en comunicación con la memoria;
 - el procesador operable para obtener una instrucción máquina para su ejecución, en el que la instrucción máquina está definida para su ejecución en un ordenador según una arquitectura de ordenador, en el que la instrucción máquina comprende:
 - 35 un campo de código de operación que identifica una instrucción de almacenamiento en adaptador;
 - un primer campo que identifica una primera ubicación que incluye los datos a ser almacenados en un adaptador;
 - un segundo campo que identifica una segunda ubicación, cuyos contenidos incluyen un descriptor de función que identifica el adaptador, una designación de un espacio de direcciones dentro del adaptador en el que deben almacenarse los datos, y un desplazamiento dentro del espacio de direcciones; y
 - 40 el procesador operable para ejecutar la instrucción máquina, en el que la ejecución comprende:
 - usar el descriptor de función para obtener una entrada de la tabla de funciones asociada con el adaptador;
 - 45 obtener una dirección de datos del adaptador usando al menos uno de entre la información en la entrada de la tabla de funciones y el desplazamiento; y

almacenar los datos desde la primera ubicación en una ubicación específica en el espacio de direcciones identificado por la designación del espacio de direcciones, en el que la ubicación específica es identificada por la dirección de datos del adaptador.

- 5 6. Sistema de ordenador según la reivindicación 5, en el que el espacio de direcciones a ser accedido es uno de entre un espacio de memoria o un espacio de E/S, y en el que la obtención de la dirección de datos comprende el uso de uno o más parámetros de la entrada de la tabla de funciones para obtener la dirección de datos.
7. Sistema de ordenador según la reivindicación 6, en el que el uso de uno o más parámetros comprende sumar un valor de un registro de dirección base de la entrada de la tabla de funciones al desplazamiento para obtener la dirección de datos.
- 10 8. Sistema de ordenador según la reivindicación 5, en el que los contenidos de la segunda ubicación incluyen una cantidad de datos a almacenar.
9. Un programa de ordenador cargado en un sistema de ordenador según la reivindicación 5 para realizar todas las etapas del procedimiento según una cualquiera de las reivindicaciones 1 a 4.

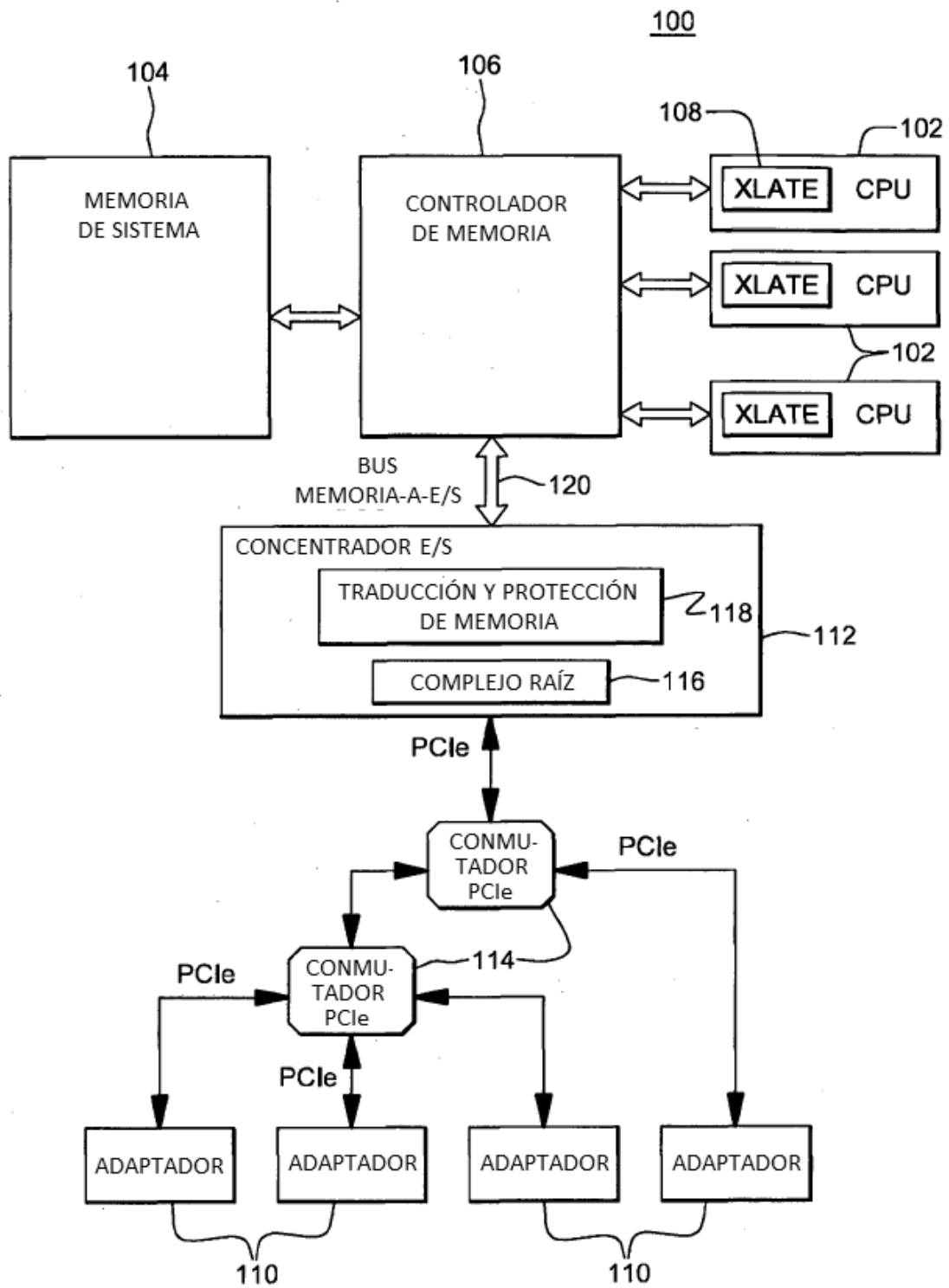


FIG. 1A

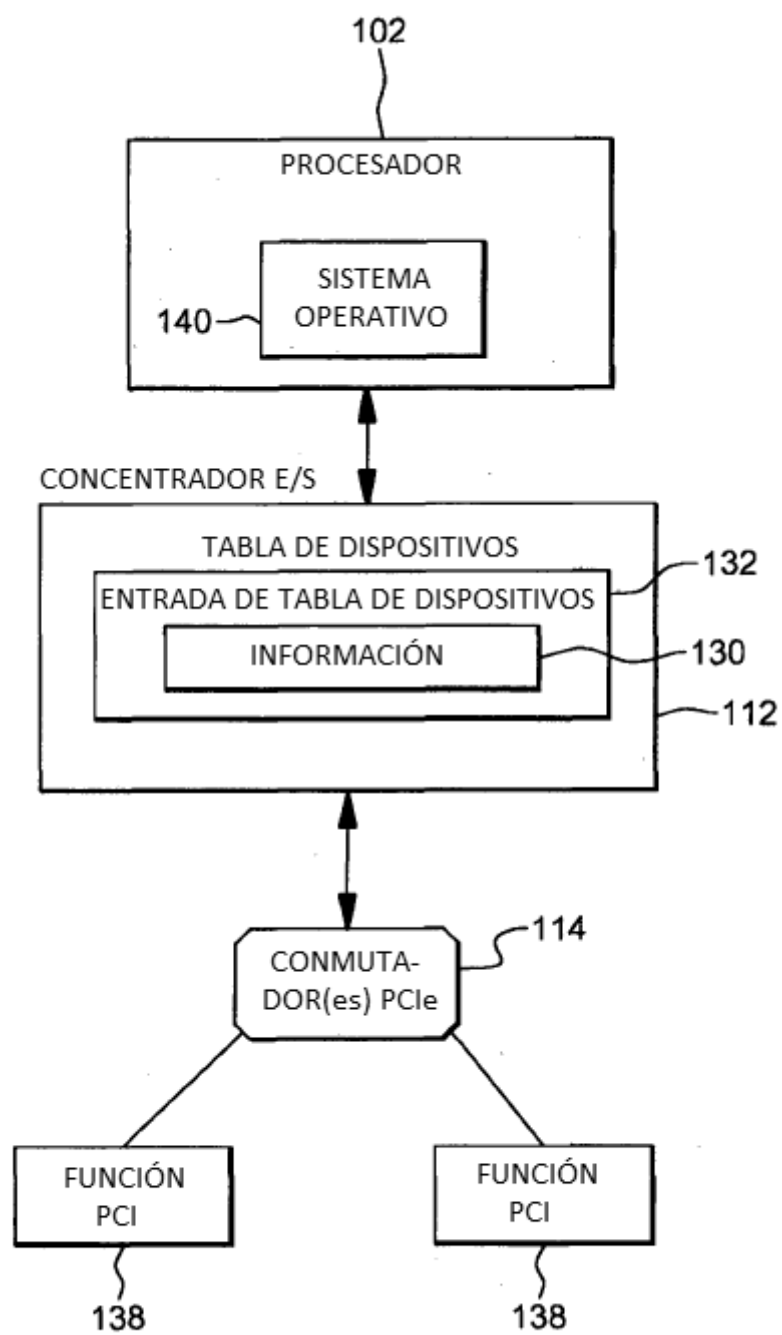


FIG. 1B

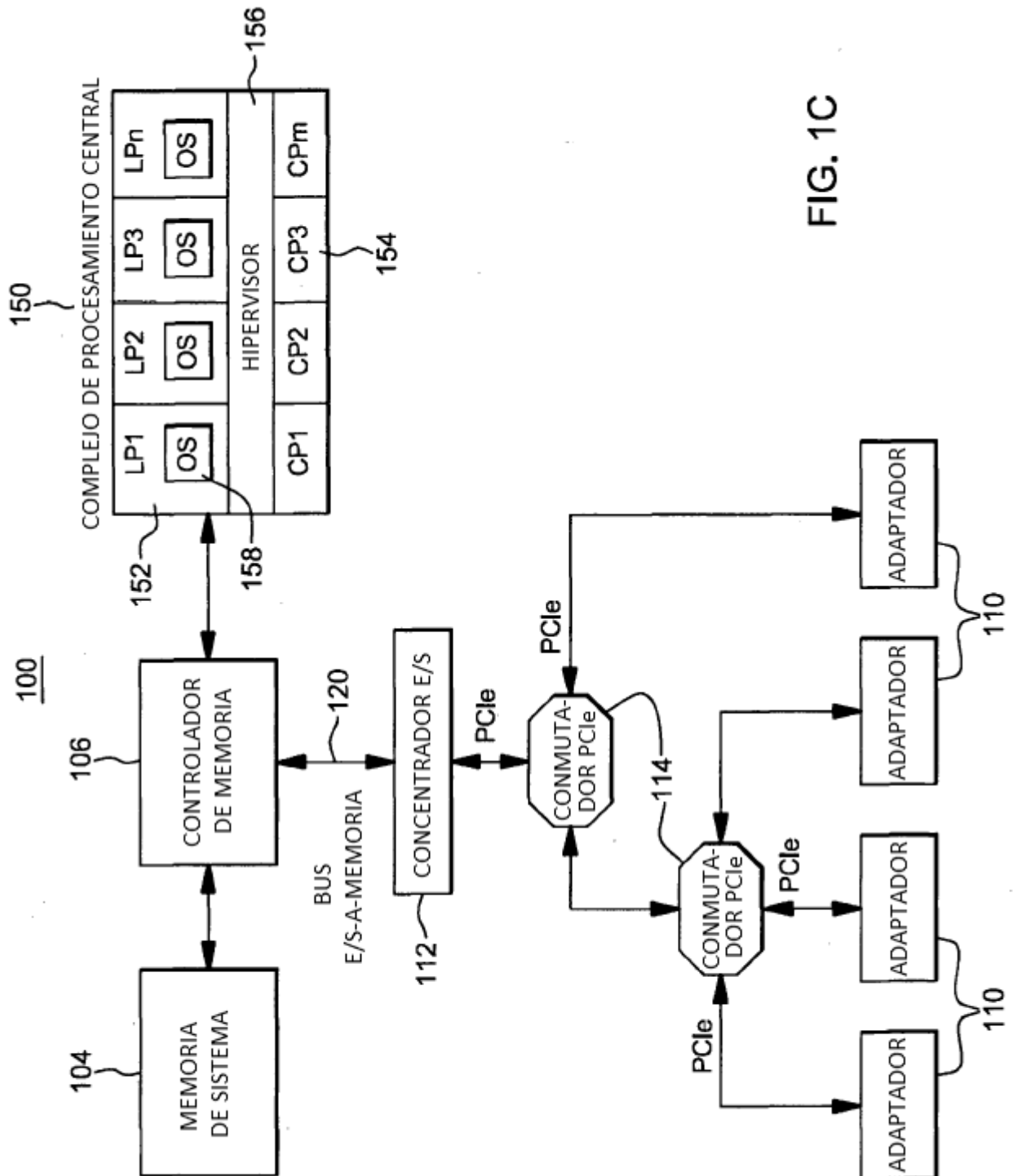


FIG. 1C

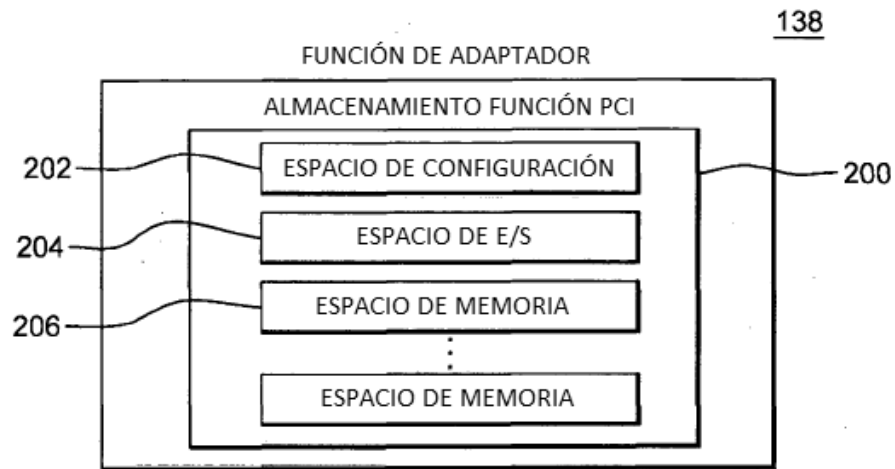


FIG. 2

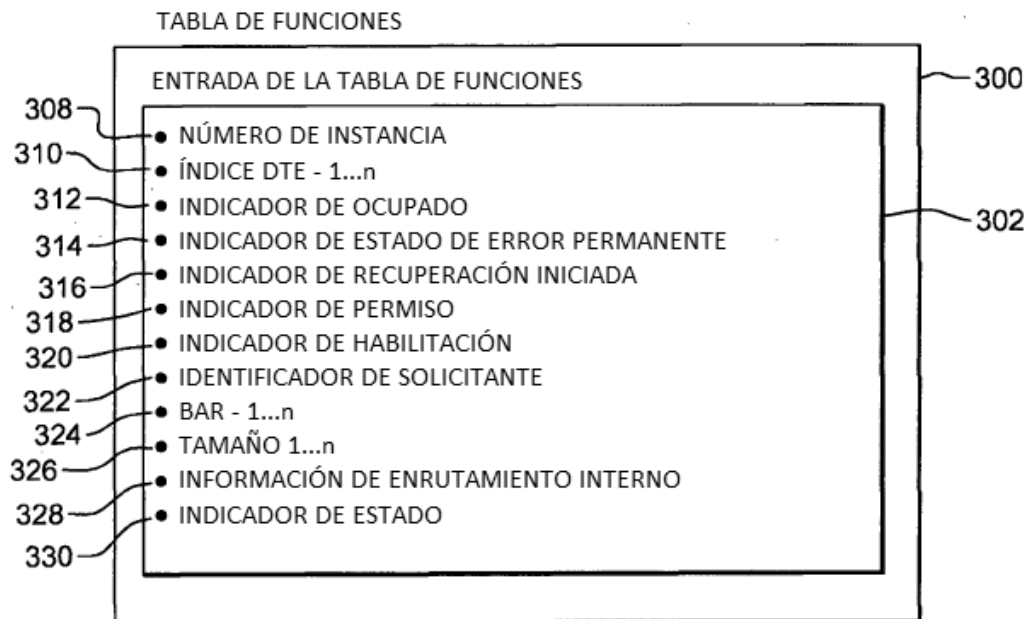


FIG. 3A

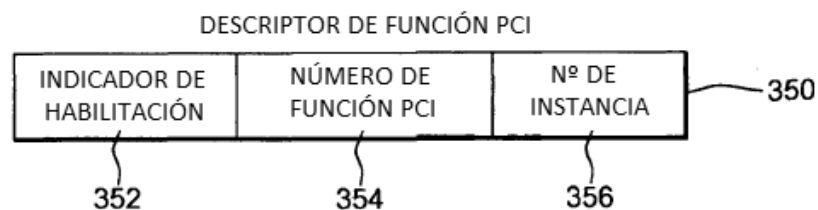


FIG. 3B

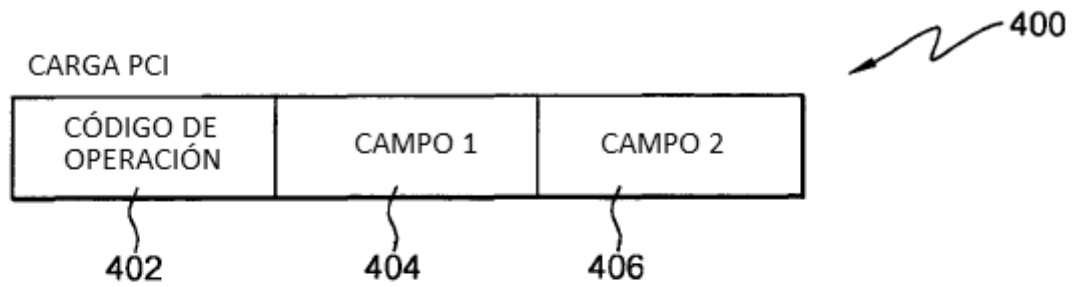


FIG. 4A

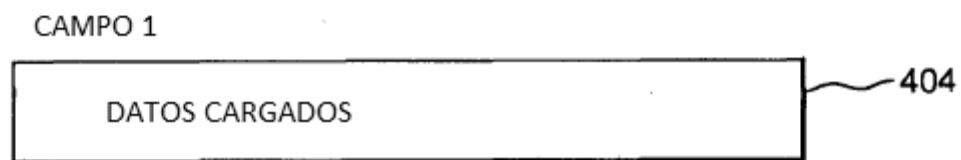


FIG. 4B

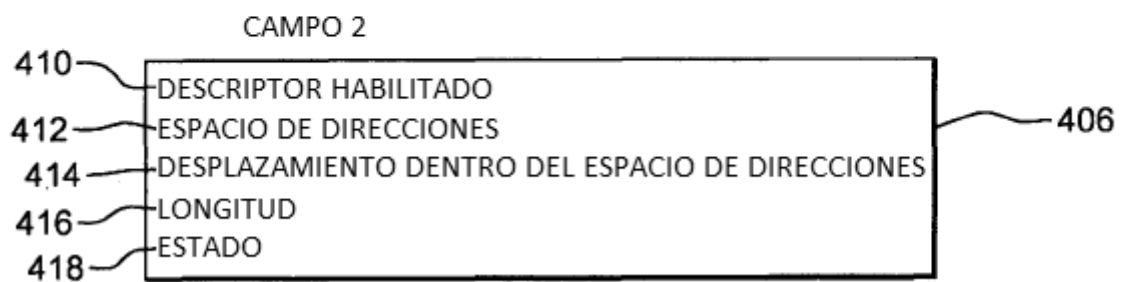


FIG. 4C

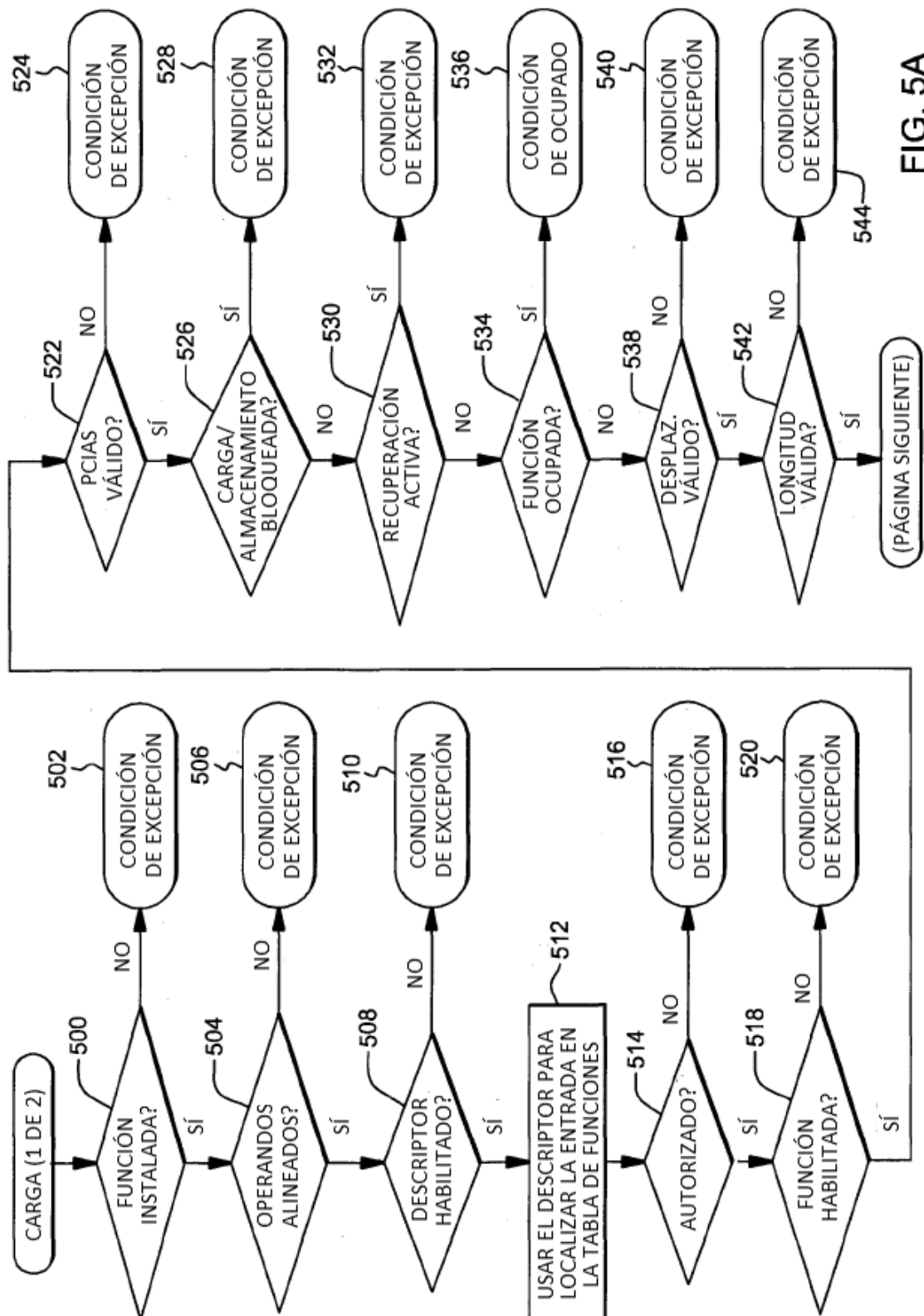


FIG. 5A

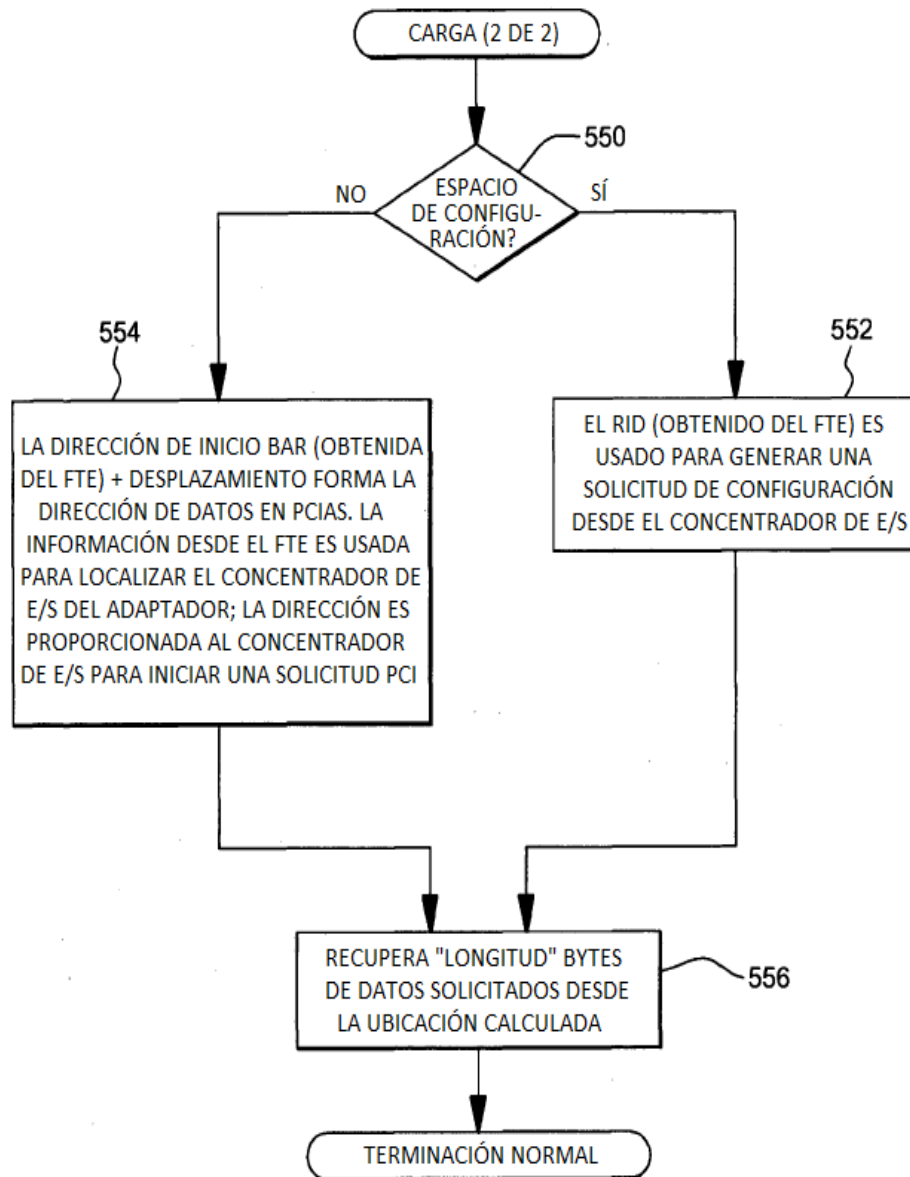


FIG. 5B

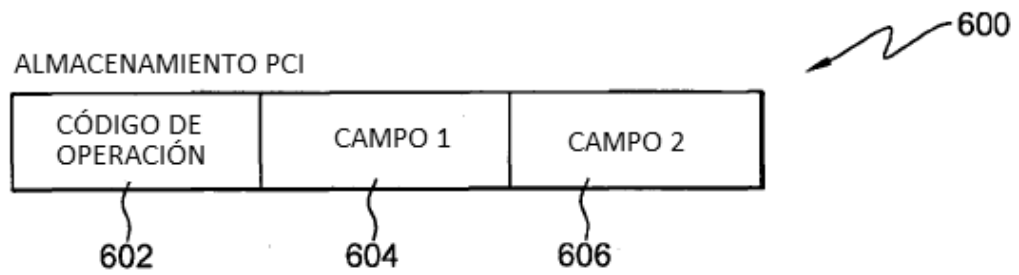


FIG. 6A

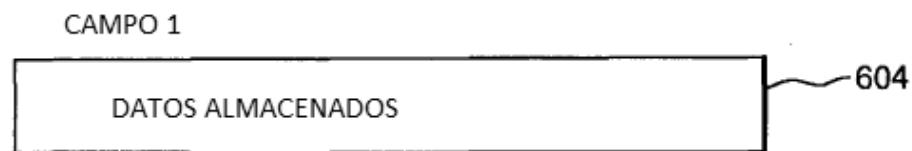


FIG. 6B

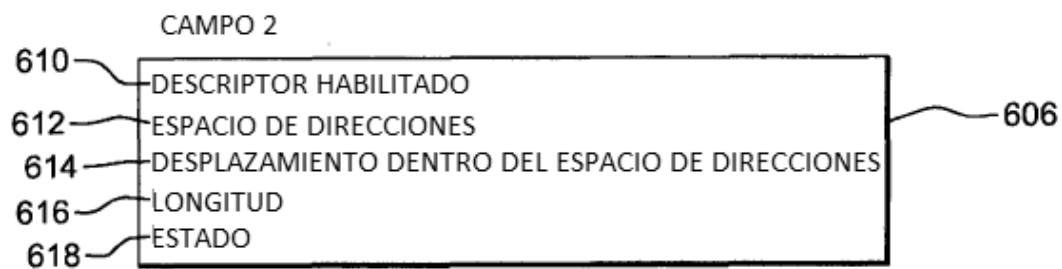


FIG. 6C

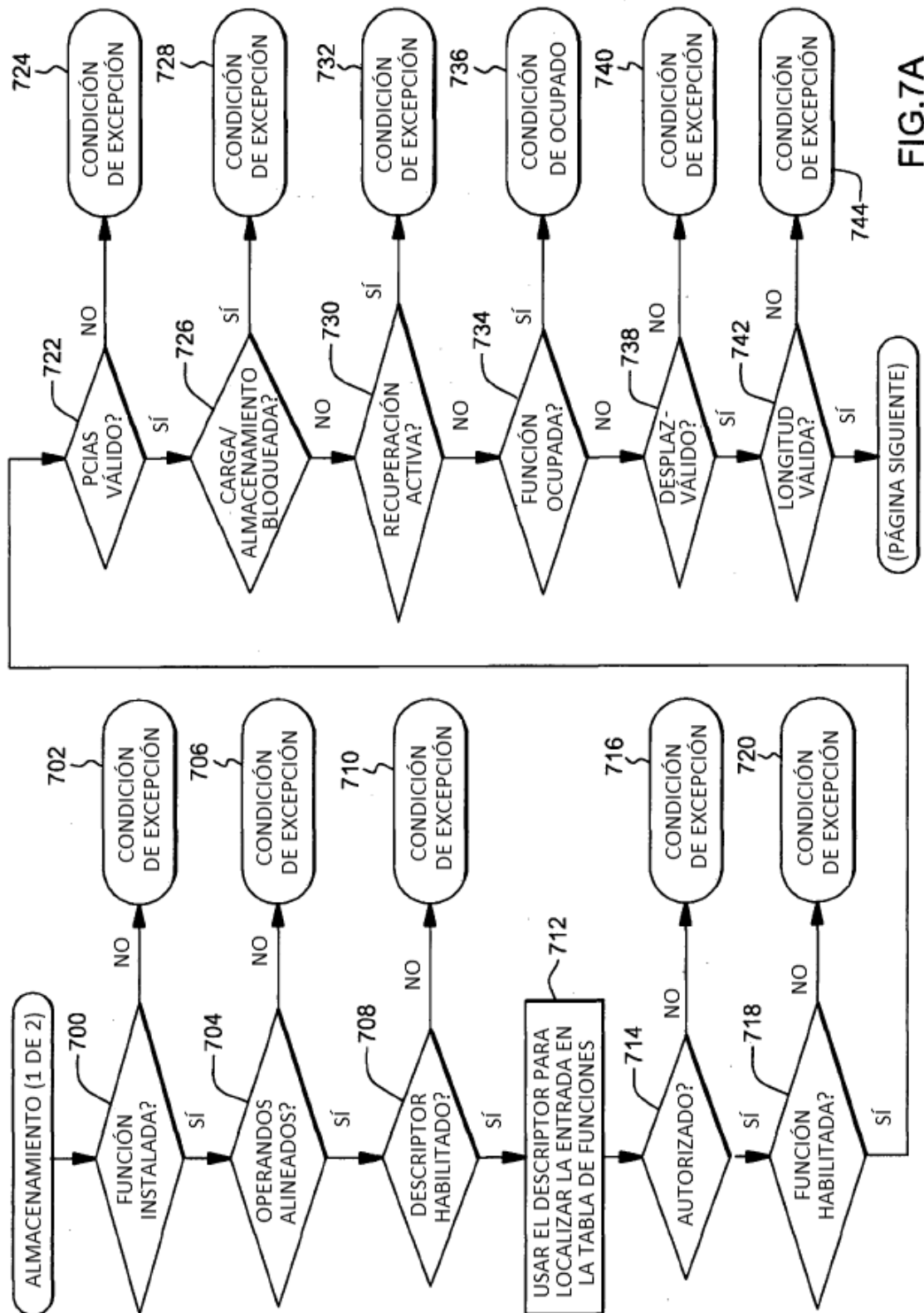


FIG.7A

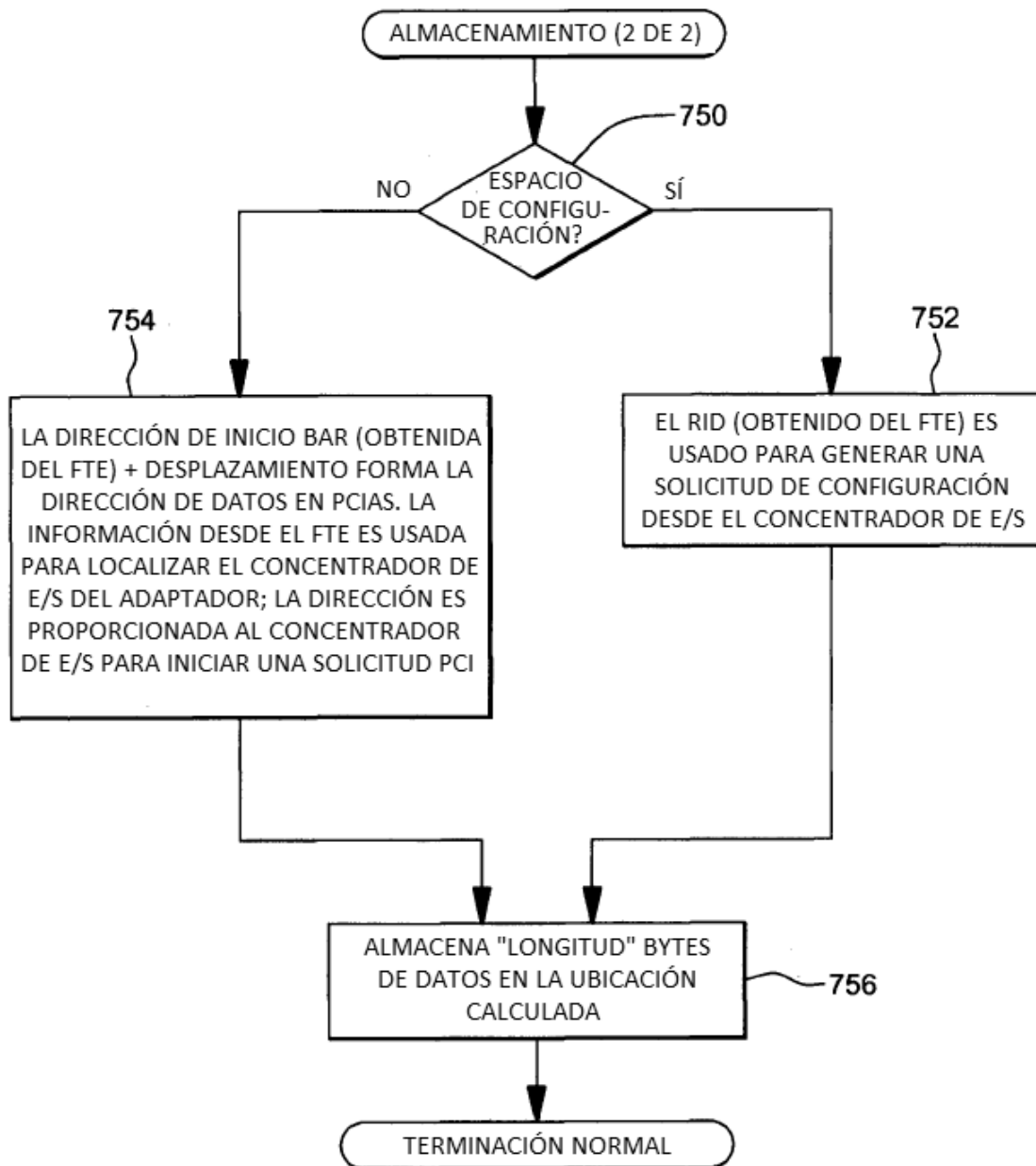


FIG. 7B

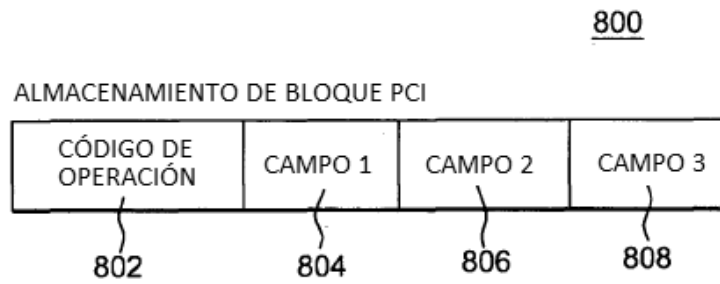


FIG. 8A

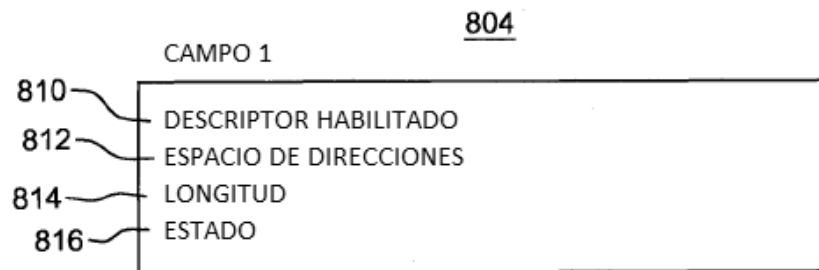


FIG. 8B

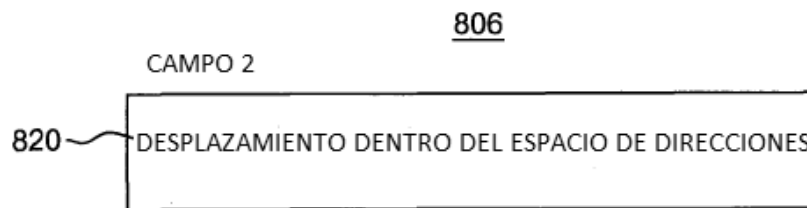


FIG. 8C

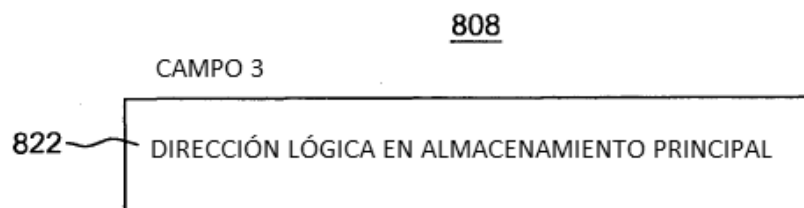


FIG. 8D

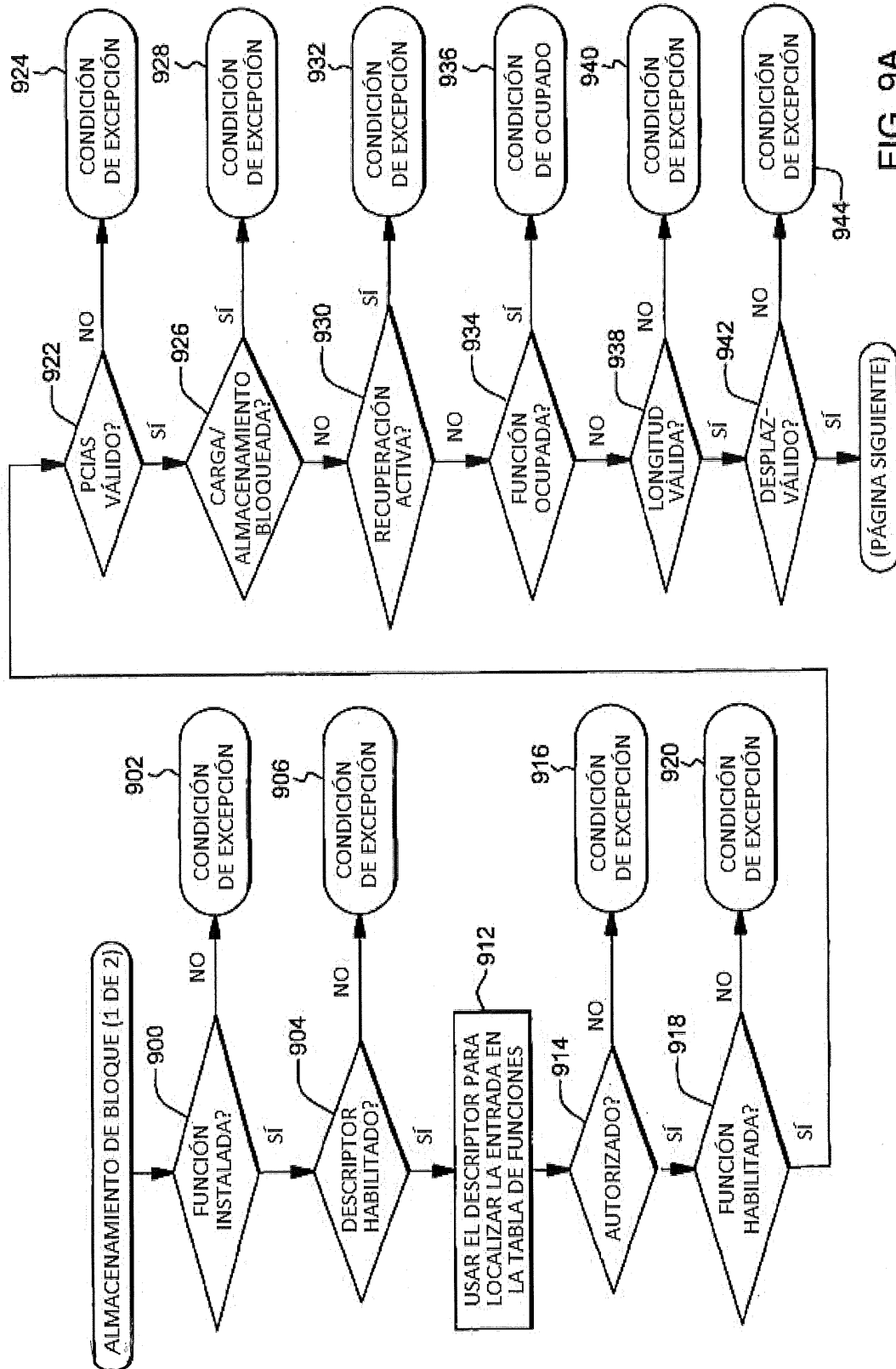


FIG. 9A

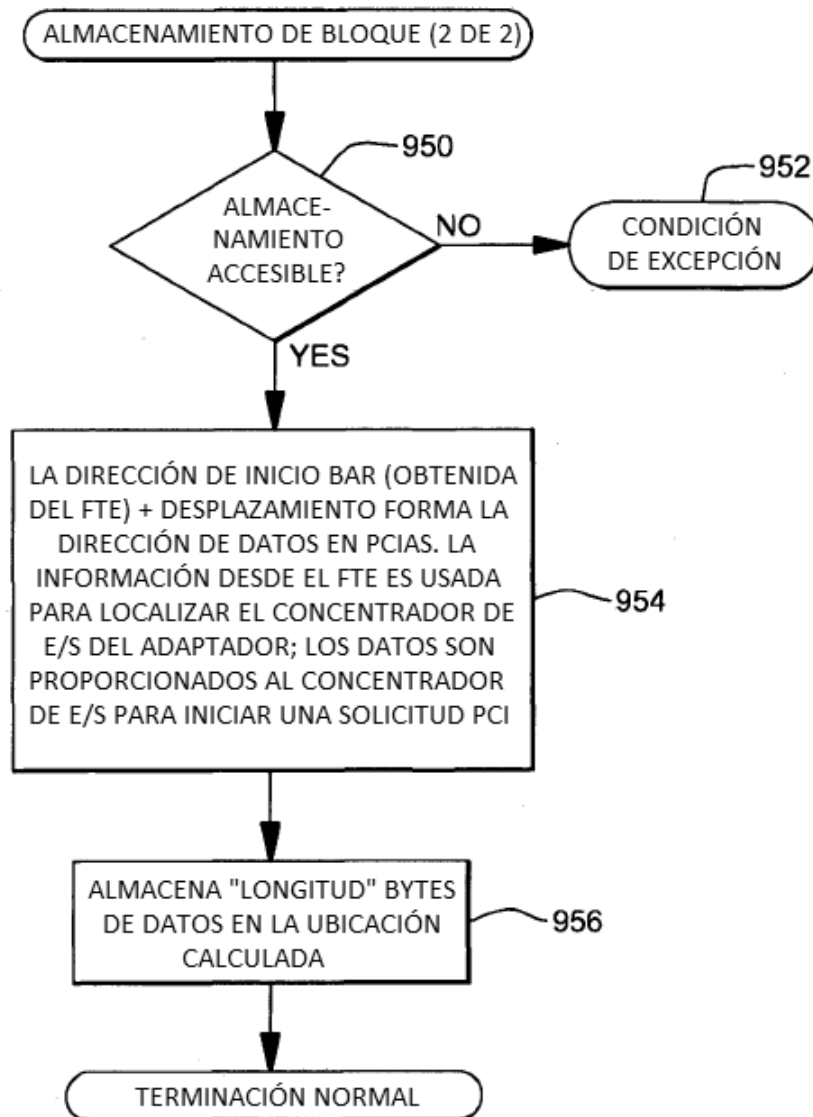


FIG. 9B

PRODUCTO DE
PROGRAMA DE
ORDENADOR
1000



FIG. 10

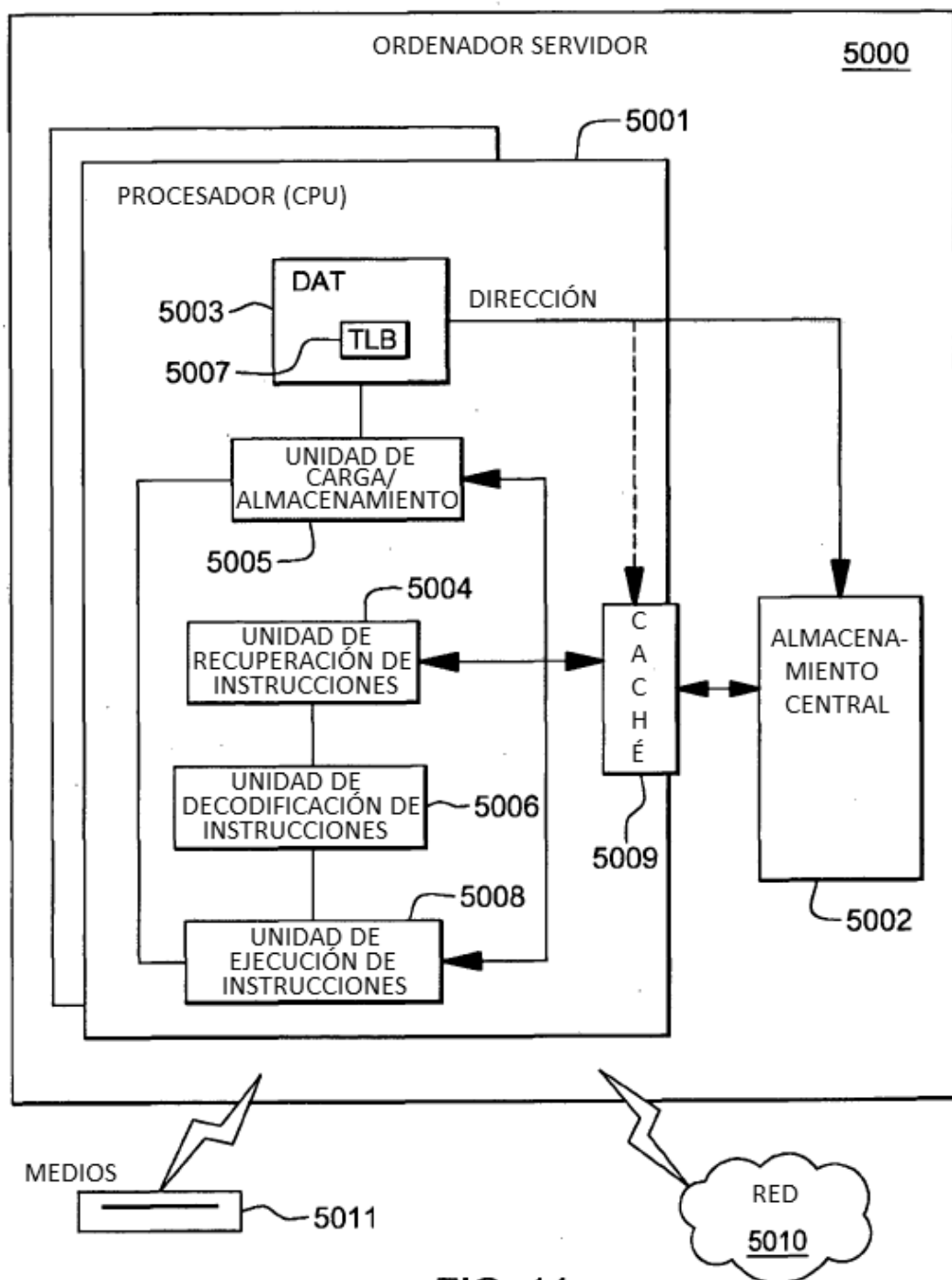


FIG. 11

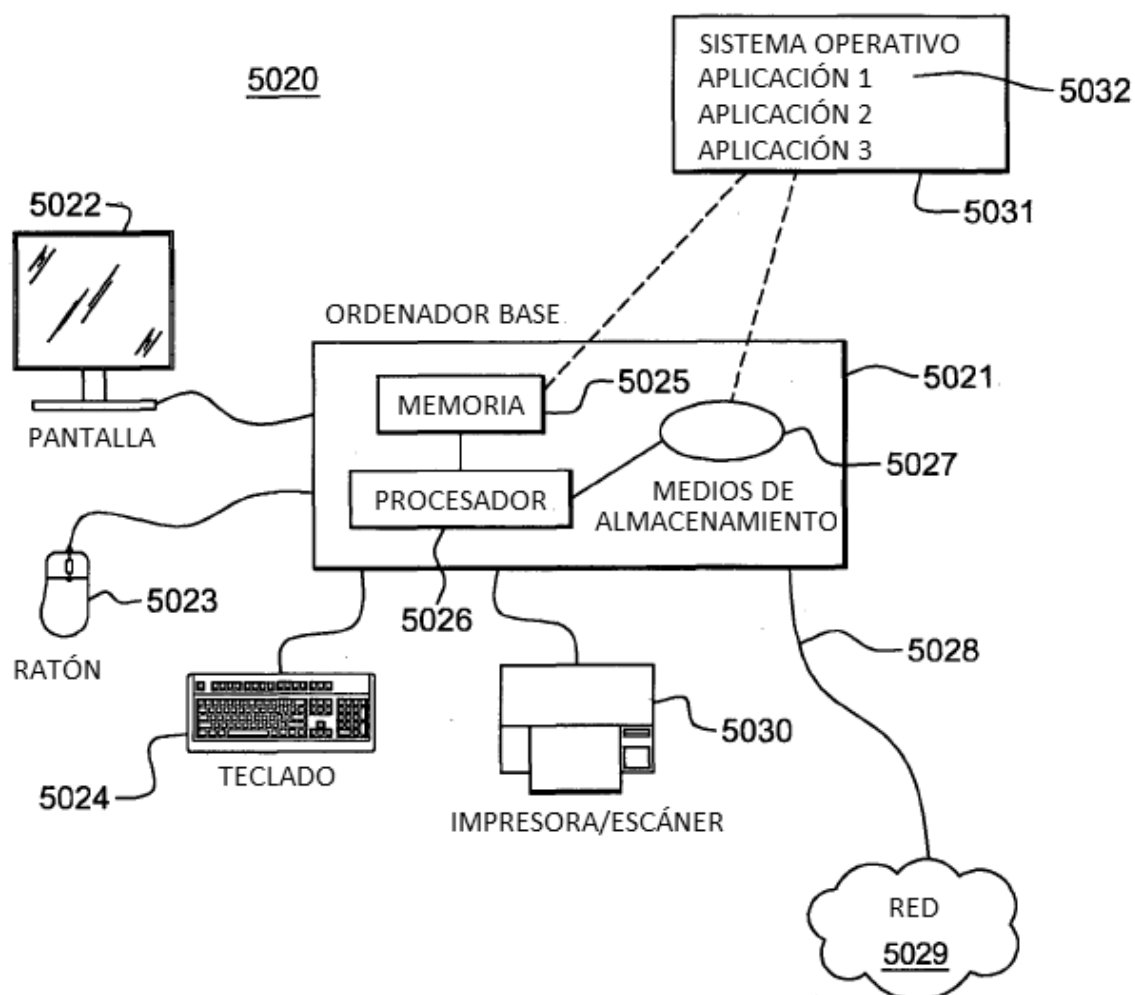


FIG. 12

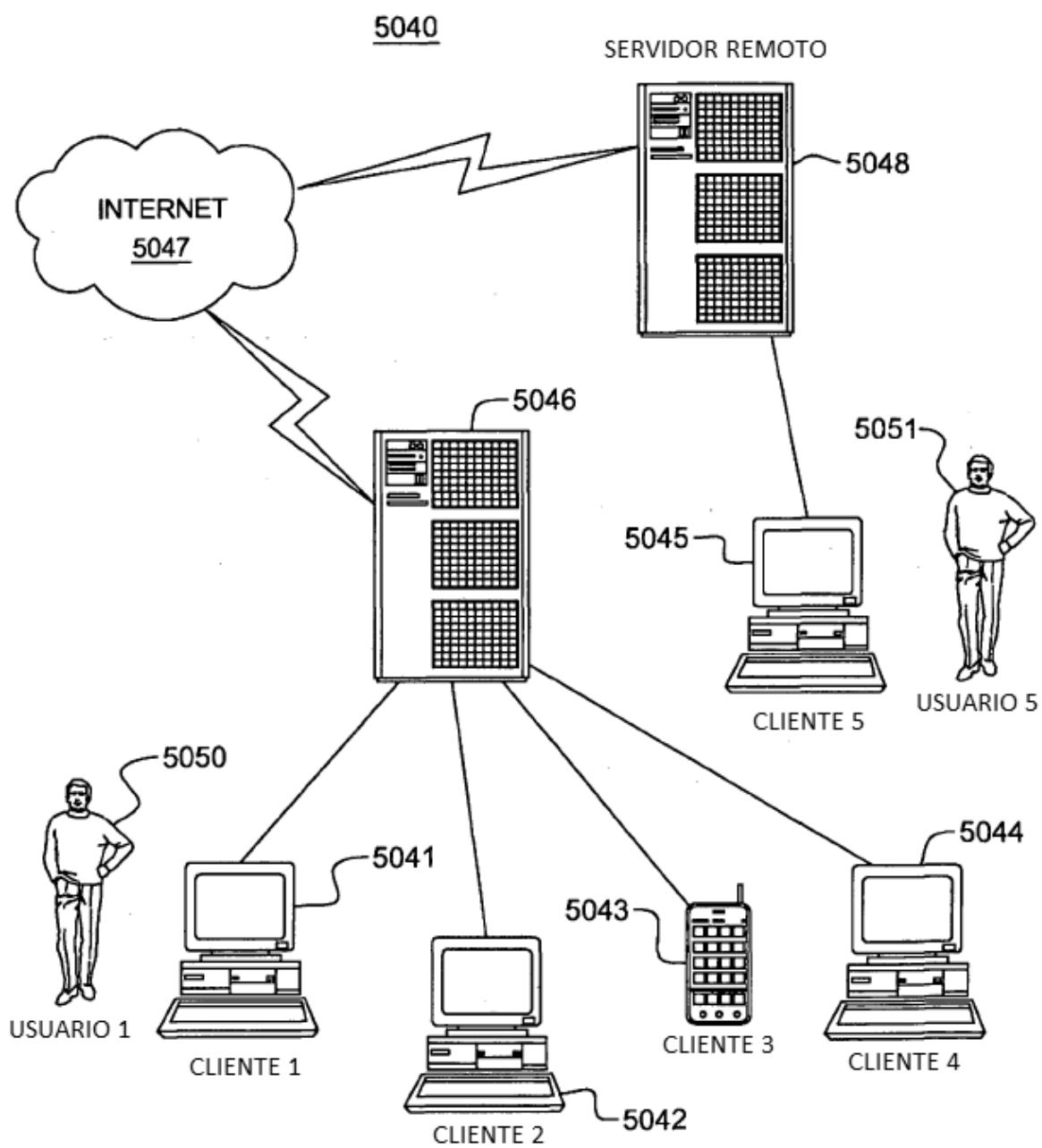


FIG. 13

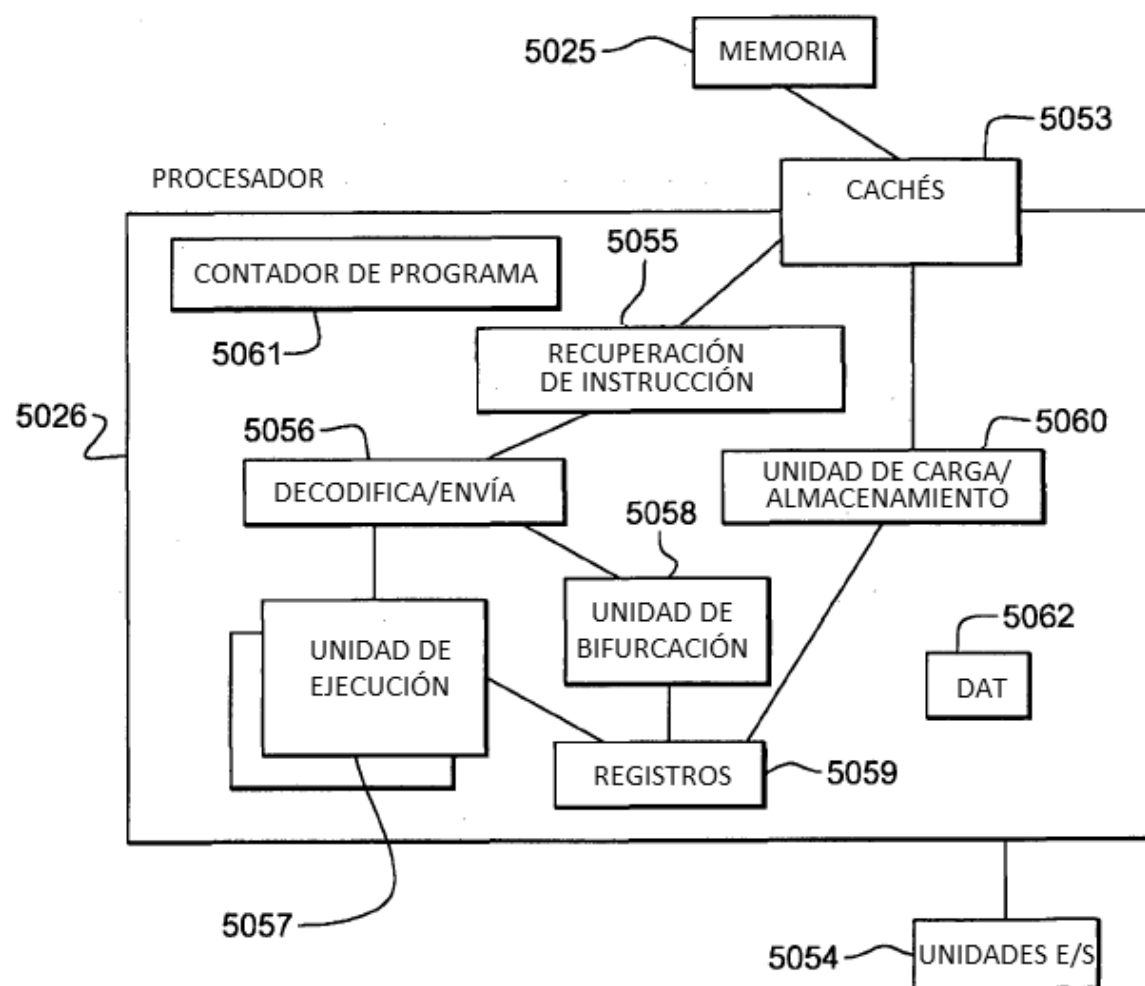


FIG. 14

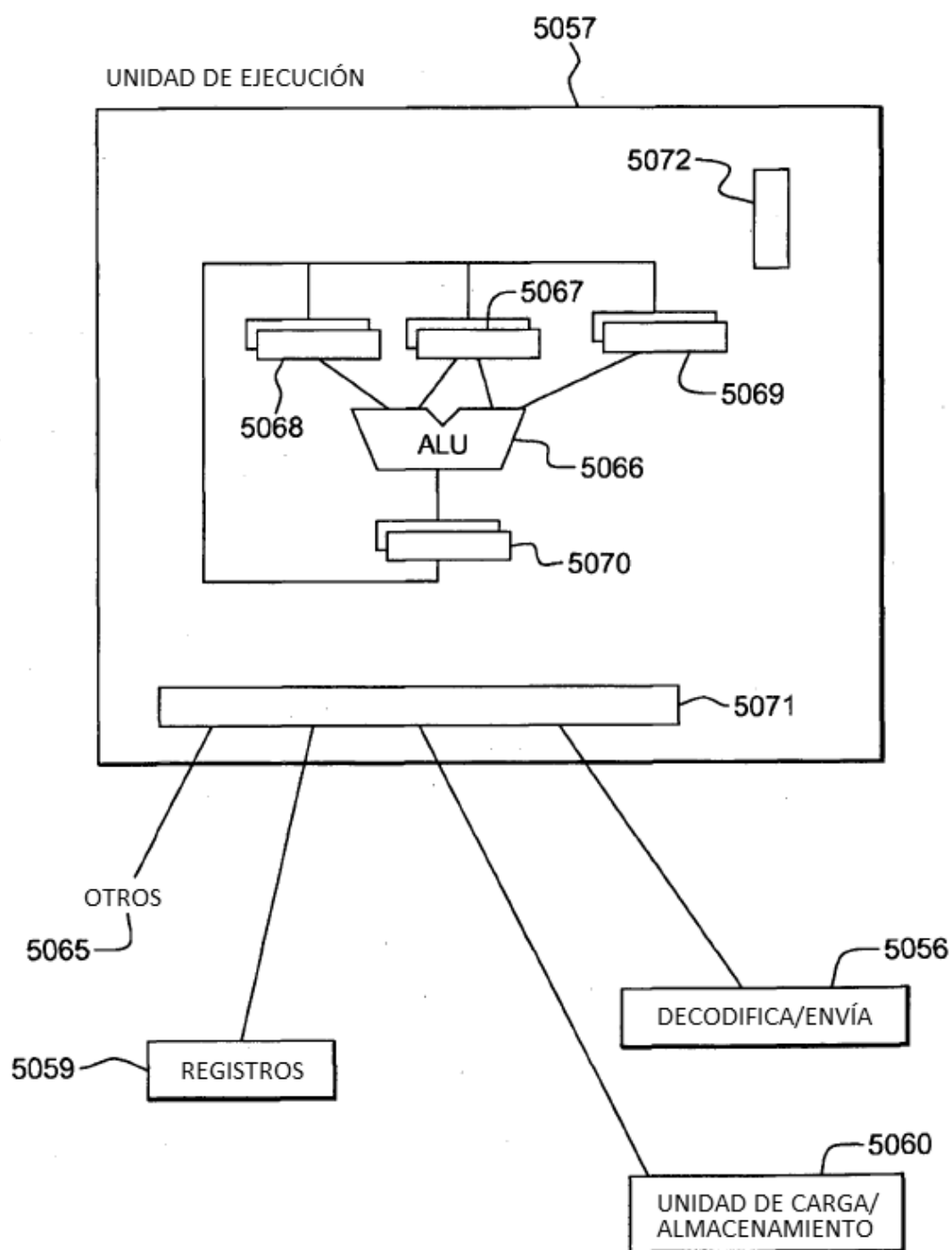


FIG. 15A

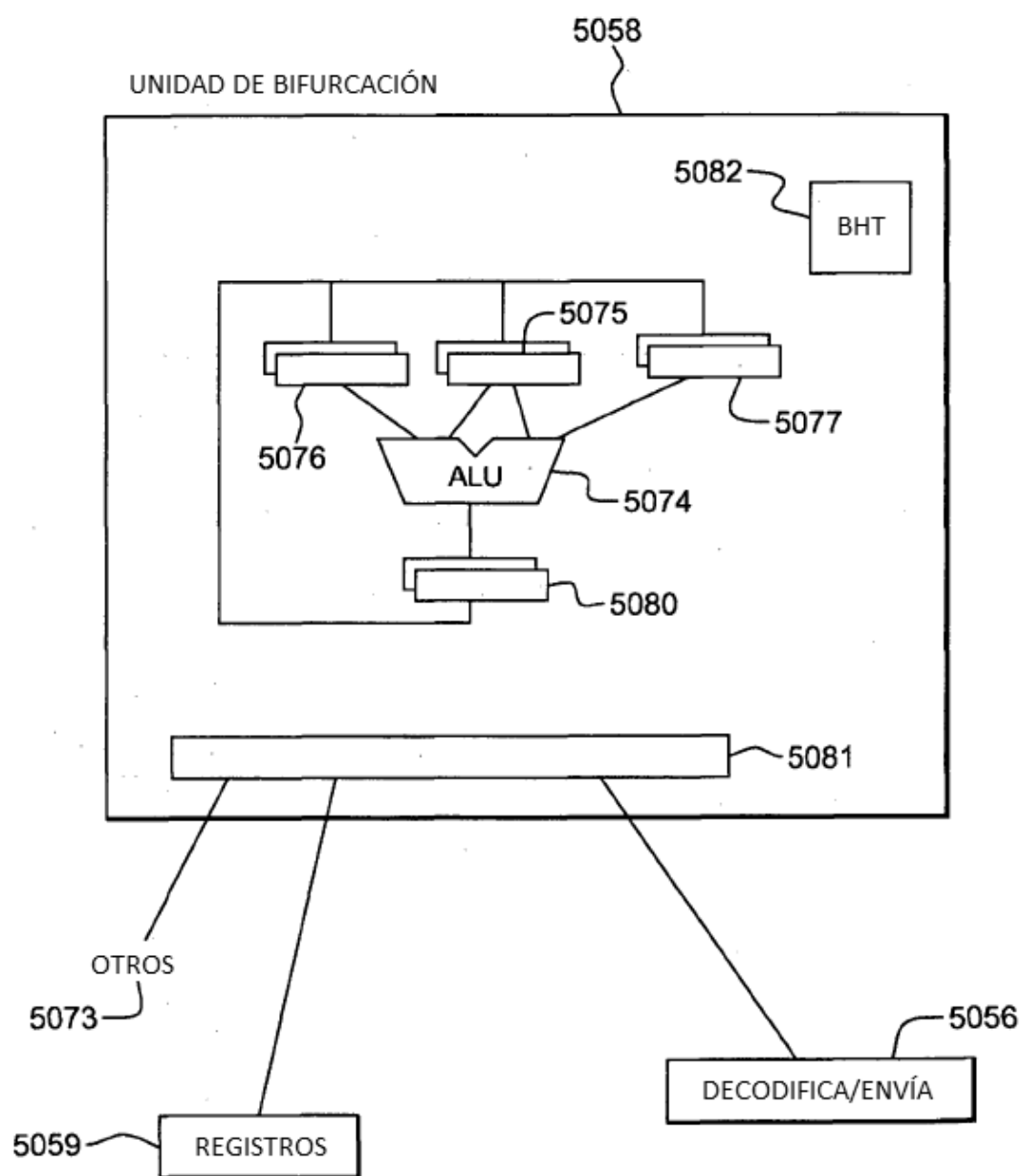


FIG. 15B

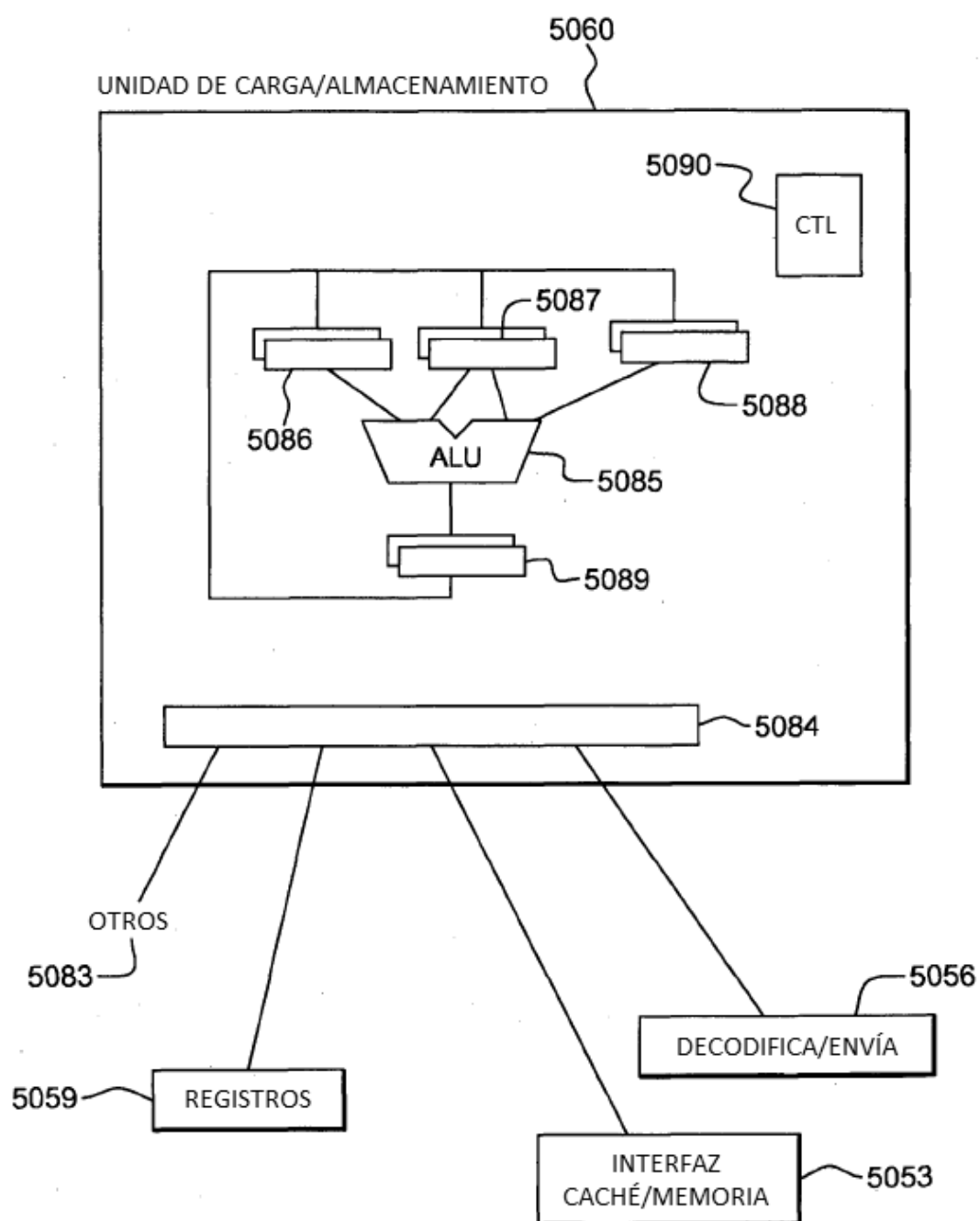


FIG. 15C

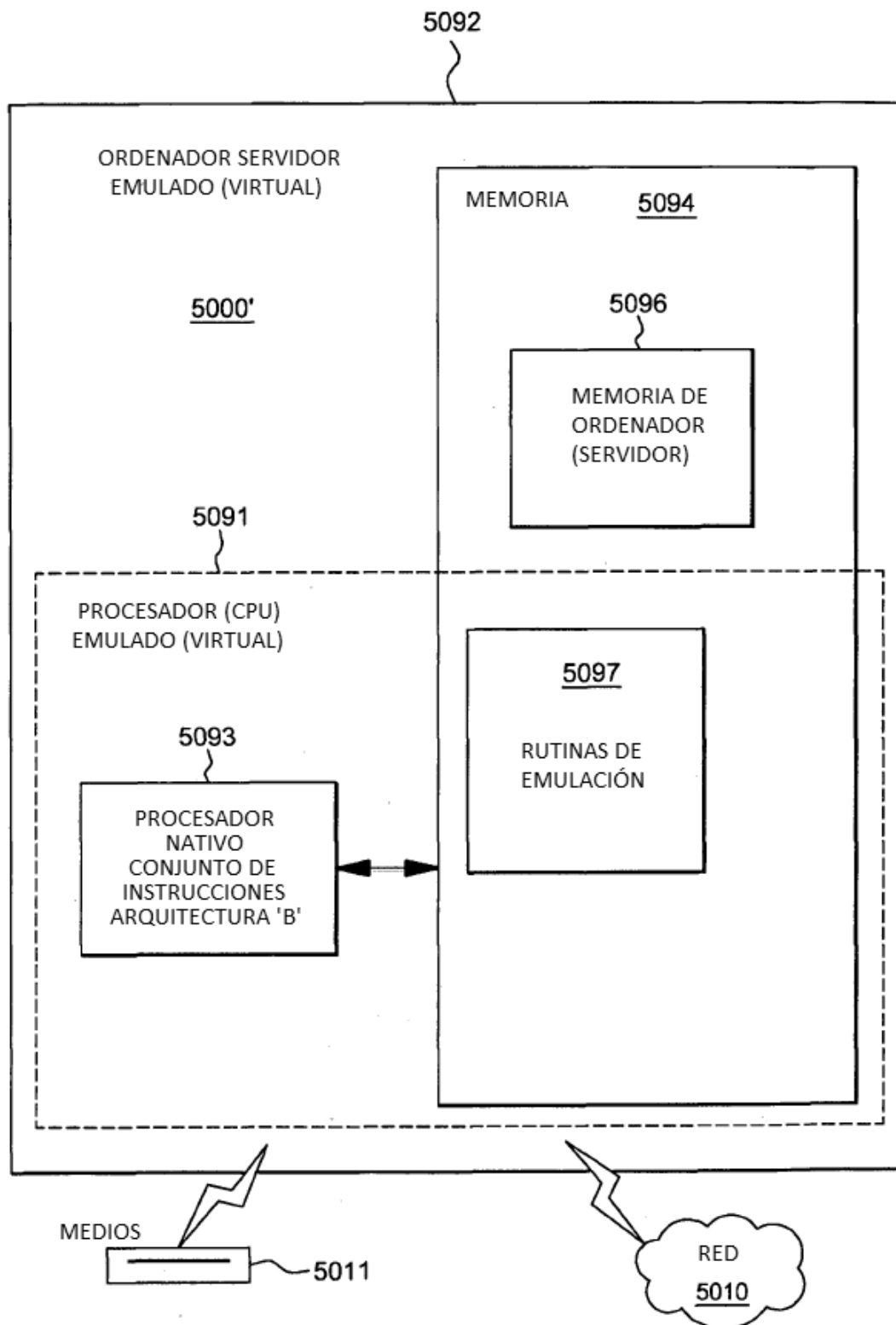


FIG. 16