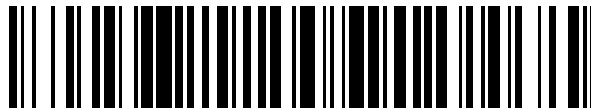


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 452 871**

51 Int. Cl.:

G06F 13/28 (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 10779302 (8)**

97 Fecha y número de publicación de la concesión europea: **12.02.2014 EP 2430556**

54 Título: **Adaptadores de habilitación/deshabilitación de un entorno de ordenadores**

30 Prioridad:

23.06.2010 US 821187

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
03.04.2014

73 Titular/es:

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

72 Inventor/es:

**GREINER, DAN;
GAINEY, CHARLES;
CRADDOCK, DAVID;
CONESKI, ANTHONY;
GLENDENING, BETH;
FARRELL, MARK;
GREGG, THOMAS y
NJOKU-CHARLES, UGOCHUKWU**

74 Agente/Representante:

DE ELZABURU MÁRQUEZ, Alberto

ES 2 452 871 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

Adaptadores de habilitación/deshabilitación de un entorno de ordenadores

Antecedentes

5 Esta invención se refiere, en general, al procesamiento de entrada/salida de un entorno de ordenadores, y en particular, a adaptadores de habilitación/deshabilitación del entorno de ordenadores.

Hoy en día los entornos de ordenadores tienen diversas configuraciones y utilizan diversos tipos de dispositivos de entrada/salida (I/O) ("input/output"). Para utilizar un dispositivo I/O, éste se habilita, y después, una vez que se ha completado la utilización, se deshabilita. La forma en la que un dispositivo I/O se habilita/deshabilita depende del dispositivo.

10 En z/Architecture® y sus procesadores ofrecidos por International Business Machines Corporation, la habilitación y deshabilitación de dispositivos I/O se ha realizado tradicionalmente en un recorrido de canal, unidad de control y base de subcanal. Las distintas funciones de una instrucción de Llamada de Subsistema de Canal proporcionan interfaces mediante las cuales los sistemas pueden manejar los distintos recursos I/O.

15 Sin embargo, se pueden utilizar otros tipos de dispositivos I/O que no incluyen canales ni subcanales. Por ejemplo, los adaptadores de interconexión de componente periférico (PCI) utilizan paradigmas de unión y comunicación que son diferentes de los de los dispositivos I/O tradicionales. La especificación PCI está disponible en la página web www.pcisig.com/home.

20 La Publicación de Estados Unidos N° 2004/0117534 A1, publicada en 17 de Junio de 2004, Parry et al., "Apparatus and Methods for Dynamically Enabling and Disabling Interrupt Coalescing in Data Processing System" describe un aparato y un método para habilitar y deshabilitar dinámicamente la coalescencia de interrupción en un sistema de procesamiento de datos. La presente invención implica monitorizar constantemente las cargas I/O sobre una IOP de un adaptador IO. El firmware en el adaptador IO puede tener una variable global que almacena contadores para el registro de función PCI. El contador se incrementa si se recibe un nuevo IO y disminuye después de enviar el mensaje de vuelta al OS. Una interrupción de temporizador se genera periódicamente de manera que un ISR se puede realizar periódicamente. En el ISR, es almacenado el máximo valor almacenado de cada contador visto desde la última interrupción de temporizador. Cuando el máximo valor almacenado es mayor que un valor umbral predeterminado, la coalescencia de interrupción se habilita.

30 La Publicación de Estados Unidos N° 2010/0005234 A1, publicada el 7 de enero de 2010, Ganga et al., "Enabling Functional Dependency in a Multi-Function Device", describe en una realización, la presente invención incluye un método para leer la información de configuración de un dispositivo multi-función (MFD), que construye un árbol de dependencia de una dependencia funcional de funciones realizadas por el MFD en base a la información de configuración, que indica que el MFD es capaz de realizar al menos un función dependiente de otra función, y el software de carga asociado con las funciones en orden basado al menos en parte en la dependencia funcional indicada. Otras realizaciones están descritas y reivindicadas.

35 La publicación de Estados Unidos N° 2004/0199700 A1, publicada el 7 de octubre de 2004, Shawn Adam Clayton "Virtual Peripheral Component Interconnect Multiple-Function Device" describe un dispositivo de interconexión de componente periférico (PCI) que comprende una interfaz de bus acoplada a un bus de interconexión de componente, una pluralidad de conjuntos de registros de espacio de configuración, y lógica virtual de funciones múltiples. Cada conjunto de registros de espacio de configuración está asociado con una función. La lógica virtual de funciones múltiples está conectada a la interfaz de bus y los conjuntos de registro de espacio de configuración. La lógica virtual de funciones múltiples proporciona acceso a una pluralidad de registros de espacio de configuración para una pluralidad de funciones. La lógica virtual de funciones múltiples también habilita una pluralidad de funciones para compartir con la interfaz de bus y otra lógica interna.

45 El documento US2006195617 expone un modo principal con un adaptador de PCI. Un número de adaptadores virtuales están definidos sobre un adaptador físico.

Breve resumen

50 De acuerdo con un aspecto de la presente invención, se proporciona una capacidad para la habilitación/deshabilitación de adaptadores, tales como adaptadores PCI. En un ejemplo, la capacidad, cuando aparece en el sistema operativo, es común a través de los adaptadores, y por tanto, se considera independiente del dispositivo.

55 Las deficiencias de la técnica anterior son superadas y se proporcionan ventajas mediante la provisión de un producto de programa de ordenador para habilitar adaptadores en un entorno de ordenadores. El producto de programa de ordenador incluye un medio de almacenamiento leíble por ordenador, leíble por un circuito de procesamiento a instrucciones de procesamiento para la ejecución del circuito de procesamiento para realizar un método. El método incluye, por ejemplo, responder a una instrucción de Procesador Lógico de Llamada (CLP) para

5 habilitar un adaptador, comprendiendo la instrucción CLP un identificador o "handle" de función que identifica el adaptador y que tiene un adaptador un indicador no habilitado, solicitando la instrucción CLP un número de espacios de dirección DMA, habilitando la ejecución uno o más espacios de dirección DMA comprendiendo a) y b): a) habilitar el adaptador, en donde la habilitación incluye el registro para la traducción de dirección e interrupciones para soportar el acceso de memoria directo a interrupciones señalizadas con mensaje para el adaptador, y b) retornar el identificador de función que tiene un indicador de adaptador habilitado.

Los métodos y sistema relacionados con uno o más aspectos de la presente invención también se describen y reivindican aquí.

10 Ventajas y características adicionales se comprenden a través de las técnicas de la presente invención. Otras realizaciones y aspectos de la invención se describen con detalle aquí y se consideran parte de la invención reivindicada.

Breve descripción de las distintas vistas de dibujos

Una realización preferida de la presente invención se describirá a continuación, sólo a modo de ejemplo, con referencia a los dibujos adjuntos, en los que:

15 la Fig. 1A muestra una realización de un entorno de ordenador para incorporar y utilizar uno o más aspectos de la presente invención;

la Fig. 1B muestra otra realización de un entorno de ordenador para incorporar y utilizar uno o más aspectos de la presente invención;

20 la Fig. 2 muestra una realización de detalles adicionales de la memoria del sistema y el concentrador de I/O de las Figs. 1A y 1B, de acuerdo con un aspecto de la presente invención;

la Fig. 3A muestra un ejemplo de una entrada de tabla de función utilizada de acuerdo con un aspecto de la presente invención;

la Fig. 3B muestra una realización de un identificador de función utilizado de acuerdo con un aspecto de la presente invención;

25 la Fig. 4A muestra una realización de una instrucción de Procesador Lógico de Llamada utilizada de acuerdo con un aspecto de la presente invención;

la Fig. 4B muestra una realización de un bloque de solicitud utilizado por la instrucción de Procesador Lógico de Llamada de la Fig. 4A, de acuerdo con un aspecto de la presente invención;

30 la Fig. 4C muestra una realización de un bloque de respuesta proporcionado por la instrucción de Procesador Lógico de Llamada de la Fig. 4A, de acuerdo con un aspecto de la presente invención;

la Fig. 5 muestra una realización de la lógica para habilitar una función PCI, de acuerdo con un aspecto de la presente invención;

la Fig. 6 muestra una realización de la lógica para deshabilitar una función PCI, de acuerdo con un aspecto de la presente invención;

35 la Fig. 7 muestra una realización de un producto de programa de ordenador que incorpora uno o más aspectos de la presente invención;

la Fig. 8 muestra una realización de un sistema de ordenador principal que incorpora y utiliza uno o más aspectos de la presente invención;

40 la Fig. 9 muestra un ejemplo más de un sistema de ordenador para incorporar y utilizar uno o más aspectos de la presente invención;

la Fig. 10 muestra otro ejemplo de un sistema de ordenador que comprende una red de ordenadores para incorporar y utilizar uno o más aspectos de la presente invención;

la Fig. 11 muestra una realización de varios elementos de un sistema de ordenador para incorporar y utilizar uno o más aspectos de la presente invención;

45 la Fig. 12A muestra una realización de la unidad de ejecución del sistema de ordenador de la Fig. 11 para incorporar y utilizar uno o más aspectos de la presente invención;

la Fig. 12B muestra una realización de la unidad de ramificación del sistema de ordenador de la Fig. 11 para incorporar y utilizar uno o más aspectos de la presente invención;

la Fig. 12C muestra una realización de la unidad de carga/almacenamiento del sistema de ordenador de la Fig. 11 para incorporar y utilizar uno o más aspectos de la presente invención; y

la Fig. 13 muestra una realización de un sistema de ordenador principal emulado para incorporar y utilizar uno o más aspectos de la presente invención.

5 Descripción detallada

De acuerdo con un aspecto de la presente invención, se proporciona capacidad para la habilitación/deshabilitación de adaptadores de un entorno de ordenadores. La capacidad es independiente del dispositivo desde el punto de vista de un sistema operativo. Es decir, el sistema operativo realiza la misma lógica independientemente del tipo de adaptador.

10 Como se ha utilizado aquí, el firmware incluye, por ejemplo, el microcódigo, milicódigo y macrocódigo del procesador. Incluye, por ejemplo, las instrucciones de nivel de hardware y/o las estructuras de datos utilizadas en la implementación del código de máquina de nivel más alto. En una realización, incluye, por ejemplo, el código patentado que es típicamente suministrado como microcódigo que incluye software de confianza o microcódigo específico para el hardware subyacente y controla el acceso del sistema operativo al hardware del sistema.

15 Además, el término adaptador incluye cualquier tipo de adaptador (por ejemplo, adaptador de almacenamiento, adaptador de procesamiento, adaptador PCI, otros tipos de adaptadores de entrada/salida, etc.). Además, en los ejemplos presentados aquí, el adaptador se utiliza de forma intercambiable con la función de adaptador (por ejemplo, la función PCI). 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 si un adaptador 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 se puede habilitar/deshabilitar de acuerdo con un aspecto de la presente invención.

20 Una realización de un entorno de ordenadores para incorporar y utilizar uno o más aspectos de la presente invención se describen con referencia a la Fig. 1A. En un ejemplo, un entorno de ordenadores 100 es un servidor System z® ofrecido por International Business Machines Corporation. System z® está basado en z/Architecture® ofrecida por International Business Machines Corporation. Detalles referentes a z/Architecture® se describen en una publicación de IBM® titulada, "z/Architecture Principles of Operation", Publicación de IBM N° SA22-7832-07, febrero de 2009. IBM®, System z® y z/Architecture® son marcas registradas de International Business Machines Corporation, Armonk, Nueva York. Otros nombres sutilizados aquí pueden ser marcas registradas, marcas o nombres de productos de International Business Machines Corporation u otras compañías.

25 En un ejemplo, el entorno de ordenadores 100 incluye una o más unidades de procesamiento central (CPUs) 102 conectadas a una memoria de sistema 104 (a.k.a., memoria principal) a través de un controlador de memoria 106. Para acceder a la memoria del sistema 104, una unidad de procesamiento central 102 expide una solicitud leída o escrita que incluye una dirección utilizada para acceder a la memoria del sistema. La dirección incluida en la solicitud típicamente no es directamente utilizable para acceder a la memoria del sistema, y por tanto, es traducida a una dirección que es directamente utilizable para acceder a la memoria del sistema. La dirección es traducida a través de un mecanismo de traducción (XLATE) 108. Por ejemplo, la dirección es traducida desde una dirección virtual a una dirección real o absoluta utilizando, por ejemplo, la traducción de dirección dinámica (DAT).

35 La solicitud, incluyendo la dirección (traducida si es necesario), es recibida por el controlador de memoria 106. En un ejemplo, el controlador de memoria 106 está compuesto por un hardware y es utilizado para arbitrar el acceso a la memoria del sistema y para mantener la consistencia de memoria. Este arbitraje se realiza para solicitudes recibidas desde las CPUs 102, así como para solicitudes recibidas desde uno o más adaptadores 110. Al igual que las unidades de procesamiento central, los adaptadores expiden solicitudes a la memoria al sistema 104 para obtener acceso a la memoria del sistema.

40 En un ejemplo, el adaptador 110 es un adaptador de Interconexión de Componente Periférico (PCI) o PCI express (PCIe) que incluye una o más funciones PCI. Una función PCI expide una solicitud que es enviada a un concentrador de entrada/salida 112 (por ejemplo un concentrador PCI) a través de uno o más conmutadores (por ejemplo, conmutadores PCIe) 114. En un ejemplo, el concentrador de entrada/salida está compuesto por un hardware, que incluye una o más máquinas de estado, y está conectado a un controlador de memoria 106 a través de un bus I/O-a-memoria 120.

45 El concentrador de entrada/salida incluye, por ejemplo un complejo de raíz 116 que recibe la solicitud desde un conmutador. La solicitud incluye una dirección de entrada/salida que es proporcionada a una traducción de dirección y unidad de protección 118 que accede a la información utilizada para la solicitud. Como ejemplos, la solicitud puede incluir una dirección de entrada/salida utilizada para realizar una operación de acceso directo a memoria (DMA) o para solicitar una interrupción señalizada por mensaje (MSI). La unidad de traducción de dirección y de protección 110 accede a la información utilizada por la solicitud de DMA y MSI. Como ejemplo particular, para una operación DMA, se puede obtener información para traducir la dirección. La dirección traducida es entonces enviada al

controlador de memoria para acceder a la memoria del sistema.

En una realización adicional del entorno de ordenadores, además de, o en lugar de, una o más CPUs 102, un complejo de procesamiento central está conectado al controlador de memoria 106 como se muestra en la Fig. 1B. En este ejemplo, el complejo de procesamiento central 150 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 partición lógica) cada uno de los cuales se describe más adelante.

Cada partición lógica 152 es capaz de funcionar como un sistema separado. Esto es, cada partición lógica puede ser restablecida independientemente, cargada inicialmente con un sistema operativo o un hipervisor (tal como z/VM® ofrecido por International Business Corporation Armonk, Nueva York), si se desea, y funcionar con diferentes programas. Un sistema operativo, un hipervisor, o un programa de aplicación que se ejecuta en una partición lógica parece tener acceso a un sistema total o completo, pero sólo una parte de ésta está disponible. Una combinación de hardware y Código Interno Autorizado (también referido como microcódigo o milicódigo) evita que un programa en una partición lógica interfiera con el programa en una partición lógica diferente. Esto permite que varias particiones lógicas diferentes funcionen como un único o múltiple procesador físico en una forma de fracción de tiempo. En este ejemplo particular, cada partición lógica tiene un sistema operativo residente 158, que puede diferir para una o más particiones lógicas. En una realización, el sistema operativo 158 es un sistema operativo z/OS® ó zLinux, ofrecido por International Business Machines Corporation, Armonk, Nueva York. Z/OS® y z/VM® son marcas registradas de International Business Machines Corporation, Armonk, Nueva York.

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

Las particiones lógicas 152 son gestionadas por el hipervisor 156 implementado por el firmware que se ejecuta en los procesadores 154. Las particiones lógicas 152 y el hipervisor 156 cada uno comprende uno o más programas que residen en las respectivas partes del almacenamiento central asociado con los procesadores centrales. Un ejemplo de hipervisor 156 es el Processor Resource/Systems Management (PR/SM) ofrecido por International Business Machines Corporation, Armonk, Nueva York.

Aunque en este ejemplo, se describe un complejo de procesamiento central que tiene particiones lógicas, uno o más aspectos de la presente invención se pueden incorporar en, y utilizar por, otras unidades de procesamiento incluyendo unidades de procesamiento de único o múltiples procesadores que no están particionadas, entre otras. El complejo de procesamiento central descrito aquí sólo es un ejemplo.

Detalles adicionales relacionados con la memoria del sistema y el concentrador de entrada/salida se describen con referencia a la Fig. 2. En este ejemplo, el controlador de memoria no se muestra, pero se puede utilizar. El concentrador I/O puede estar conectado con la memoria del sistema 104 y/o al procesador 204 directamente o a través de un controlador de memoria.

Haciendo referencia a la Fig. 2, en un ejemplo, la memoria del sistema 104 incluye uno o más espacios de dirección 200. Un espacio de dirección es una parte particular del sistema de memoria que ha sido asignada a un componente particular del entorno de ordenadores, tal como un adaptador particular. En un ejemplo, el espacio de dirección es accesible mediante el acceso directo a memoria (DMA) iniciado por el adaptador, y por tanto, un espacio de memoria está referido, en los ejemplos mostrados aquí, como un espacio de dirección DMA. Sin embargo, en otros ejemplos, el acceso de memoria directo no se utiliza para acceder al espacio de dirección.

En un ejemplo, hay un sistema operativo 202 que se ejecuta dentro de un procesador 204 (por ejemplo, la CPU 102 o un CP 154 asignados a un LP 152) que asigna un espacio de dirección DMA a un adaptador particular. Esta asignación se realiza a través de un proceso de registro, que causa una inicialización (a través de, por, ejemplo un software de confianza) de una entrada de tabla de dispositivo 210 para ese adaptador. Hay una entrada de tabla de dispositivo por espacio de dirección asignado y esta entrada de tabla de dispositivo está asociada con un único adaptador. La entrada de tabla del dispositivo está situada en una tabla de dispositivo 212 situada en un concentrador I/O 112. Por ejemplo, la tabla de dispositivo 212 está situada dentro de la traducción de dirección y la unidad de protección del concentrador de I/O.

En un ejemplo, la entrada de tabla de dispositivo 210 incluye información utilizable en proporcionar diversos servicios para el adaptador. Por ejemplo, la entrada de tabla de dispositivo incluye un indicador de habilitación 214 que indica si la entrada de tabla de dispositivo está habilitada para un adaptador particular. La entrada de tabla de dispositivo puede incluir más, menos o diferente información para las operaciones de habilitación/deshabilitación, así como para otros servicios proporcionados, tales como traducción de dirección, gestión de interrupción, etc.

En una realización, la entrada de tabla de dispositivo para ser utilizada por un adaptador particular que expide una solicitud es localizada utilizando un identificador de solicitante (RID) (y/o una parte de la dirección) situada en una solicitud expedida por una función PCI 220 asociada con un adaptador. La ID de solicitante (por ejemplo, un valor de

16 bits que especifica, por ejemplo un número de bus, un número de dispositivo y un número de función) es incluido en la solicitud así como una dirección I/O para ser utilizada. La solicitud, que incluye el RID y la dirección I/O, son proporcionadas, por ejemplo, una memoria direccionable de contenidos (CAM) 230 a través de, por ejemplo, un conmutador 114, que se utiliza para proporcionar un valor de índice. Por ejemplo, la CAM incluye múltiples entradas, correspondiendo cada entrada a un índice en la tabla del dispositivo. Cada entrada CAM incluye un valor de un RID. Si, por ejemplo, el RID recibido coincide con el valor contenido en una entrada en la CAM, se utiliza un correspondiente índice de tabla de dispositivo para localizar la entrada de tabla de dispositivo. Esto es, la salida de la CAM se utiliza para indexar en la tabla de dispositivo 212 para localizar la entrada de tabla de dispositivo 210. Si no hay coincidencia, el paquete recibido es descartado. (En otras realizaciones, una CAM u otra consulta no es necesaria y el RID recibido se utiliza como índice.)

Además de una entrada de tabla de dispositivo, otra estructura está también asociada con un adaptador, que incluye información referente al adaptador. En los ejemplos particulares descritos aquí, el adaptador es una función PCI, y por tanto, la estructura de datos está referida como entrada de tabla de función (FTE). Aunque los ejemplos aquí se refieren a funciones PCI, en otras realizaciones, otras funciones de adaptador o adaptadores se pueden habilitar/deshabilitar, de acuerdo con un aspecto de la presente invención.

Como se muestra en la Fig. 3A, en un ejemplo, una entrada de tabla de función 300 es una entrada en una tabla de función 302 almacenada, por ejemplo en la memoria segura. Cada entrada de tabla de función 300 incluye información para ser utilizada en el procesamiento asociado con su adaptador. En un ejemplo, la entrada de tabla de función 300 incluye un número de caso 308 que indica un caso particular de la función de adaptador asociada con la entrada de tabla de función; uno o más índices de entrada de tabla de dispositivo 310, cada uno de los cuales se utiliza como un índice en la tabla de dispositivo para localizar su entrada de tabla de dispositivo correspondiente (una función PCI puede tener una pluralidad de espacios de dirección asignados a la misma, y por tanto, una pluralidad de DTEs); un indicador ocupado 312 que indica si la función PCI está ocupada; un indicador de estado de error permanente 314 que indica si la función está en un estado de error permanente; un indicador de recuperación iniciada 316 que indica si se ha iniciado la recuperación para la función; un indicador de permiso 318 que indica si el sistema operativo que intenta habilitar la función PCI tiene autoridad para hacerlo; y un indicador de habilitación 320 que indica si la función está habilitada (por ejemplo, 1 = habilitada, 0 = deshabilitada).

En un ejemplo, el indicador ocupado, el indicador de estado de error permanente, y el indicador de recuperación iniciada son establecidos en base a un control realizado por el firmware. Además, el indicador de permiso se establece, por ejemplo, en base a la política. En otras realizaciones, la entrada de tabla de función puede incluir, más, menos, o diferente información.

En una realización, para localizar la entrada de tabla de función en una tabla de función que incluye una o más entradas, se utiliza un identificador de función, tal como un "handle" de función. Por ejemplo, se utilizan uno o más bits del identificador de función como índice en la tabla de función para localizar la entrada de tabla de función particular.

Haciendo referencia a la Fig. 3B, se describen detalles adicionales referentes a un identificador de función. En un ejemplo, un identificador de función 350 incluye un indicador de habilitación 352 que indica si el identificador de función PCI está habilitado; un número de función PCI 354 identifica la función (este es un identificador estático) y, en una realización, es un índice en la tabla de función, y un número de caso 356 que indica el caso particular de este identificador de función. Por ejemplo, cada vez que la función es habilitada, el número de caso se incrementa para proporcionar un nuevo número de caso.

Para utilizar la función PCI, tiene que estar habilitada. Por ejemplo, el sistema operativo al que le gustaría utilizar una función PCI realiza una consulta para determinar la una o más funciones que se pueden elegir para utilizar (en base a la configuración I/O) y selecciona una de esas funciones para ser habilitada. En un ejemplo, la función es habilitada utilizando un comando de función PCI de ajuste de una instrucción de Procesador Lógico de Llamada. Una realización de esta instrucción se muestra en la Fig. 4A. Como se muestra, en un ejemplo, una instrucción de Procesador Lógico de Llamada 400 incluye un código de operación 402 que indica que esta es una instrucción de Procesador Lógico de Llamada; y una indicación para un comando 404. En un ejemplo, esta indicación es una dirección de un bloque de solicitud que describe el comando que a va ser realizado. Una realización de tal bloque de solicitud se muestra en la Fig. 4B.

Como se muestra en la Fig. 4B, en un ejemplo, un bloque de solicitud 420 incluye un cierto número de parámetros, tales como, por ejemplo, un campo de longitud 422 que indica la longitud del bloque de solicitud; un campo de comando 424 que indica el comando de función PCI de ajuste; un identificador de función PCI 426, que es el identificador que va ser proporcionado o bien a la función de habitación o bien a la función de deshabilitación; un código de operación 428, que se utiliza para designar o bien una operación de habilitación o bien de deshabilitación; y un número de espacios de acceso DMA (DMAAS) 430, que indica el número solicitado de espacios de dirección que va a ser asociado con la función PCI particular. Se puede incluir más, menos o diferentes información en otras realizaciones.

Por ejemplo, en un entorno virtual en el que la instrucción es expedida por un ordenador principal de un huésped de

modo de almacenamiento paginable se proporciona una identidad de huésped. También son posibles otras variaciones. En un ejemplo, en z/Architecture®, un huésped paginable es ejecutado interpretativamente a través de la instrucción de Ejecución Interpretativa de Inicio (SIE), en el nivel 2 de interpretación. Por ejemplo, el hipervisor de partición lógica (LPAR) ejecuta la instrucción SIE para empezar la partición lógica en memoria física, fijada. Si z/VM® es el sistema operativo en esa partición lógica, expide la instrucción SIE para ejecutar sus máquinas (virtuales) de huésped en su almacenamiento (virtual) V=V. Por lo tanto, el hipervisor LPAR utiliza SIE de nivel-1, el hipervisor z/VM® utiliza SIE de nivel-2.

Como respuesta a la expedición y procesamiento de la instrucción de Procesador Lógico de Llamada, un bloque de respuesta en devuelta y la información incluida en el bloque de respuesta es dependiente de la operación que se va a realizar. Una realización del bloque de respuesta se muestra en la Fig. 4C. En un ejemplo, el bloque de respuesta 450 incluye un campo de longitud 452 que indica la longitud del bloque de respuesta; indicando un código de respuesta a un estado del comando, y un identificador de función PCI 456 que identifica la función PCI. Como respuestas al comando de habilitación, el identificador de función de PCI es un identificador habilitado de la función PCI. Además, después de completar la operación de deshabilitación, el identificador de función PCI es un identificador general que se puede habilitar en el futuro mediante una función de habilitación.

Una realización de la lógica para habilitar una función PCI se describe con referencia a la Fig. 5. En un ejemplo, esta lógica es iniciada como respuesta a la expedición de una instrucción de Procesador Lógico de Llamada en el que el comando es establecido al comando de función de PCI de establecimiento y el código de operación es establecido para habilitar la función. Esta lógica se realiza mediante, por ejemplo un procesador como respuesta al sistema operativo o a un controlador de dispositivo del sistema operativo autorizado para realizar esta lógica expidiendo la instrucción. En otra realización, la lógica se puede realizar con el uso de la instrucción de procesador Lógico de Llamada.

Haciendo referencia a la Fig. 5, inicialmente, se hace una determinación de si un identificador dispuesto en el bloque de solicitud de la instrucción de Procesador Lógico de Llamada es un identificador válido, INVESTIGACIÓN 500. Esto es, ¿el identificador apunta a una entrada válida en la tabla de función o está fuera del rango de entradas válidas (por ejemplo, designa la parte de número de función del identificador una función instalada?). Si el identificador no es conocido, entonces un código de respuesta correspondiente se proporciona indicando que el identificador no se reconoce. Sin embargo, si el identificador es conocido, entonces se realiza una investigación adicional para averiguar si el identificador está habilitado, INVESTIGACIÓN 504. Esta determinación se hace comprobando el indicador de habilitación en el identificador de función PCI. Si la indicación es establecida indicando que el identificador está habilitado, entonces se devuelve un código de respuesta indicándolo, ETAPA 506.

Sin embargo, si el identificador es conocido y no está habilitado (es decir, válido para la habilitación), entonces se hace una determinación adicional de si el número de espacios de dirección solicitado para ser asignados a la función PCI es mayor que un valor máximo, INVESTIGACIÓN 508. Para realizar esta determinación, el número de espacios de dirección DMA es especificado cuando el bloque de solicitud es comparado con un valor máximo (provisto en base a la política, en un ejemplo). Si el número de espacios de dirección es mayor que el máximo valor, entonces se proporciona un código de respuesta que indica un valor válido para los espacios de dirección DMA, ETAPA 510. De lo contrario se realiza una determinación de si el número de espacios de dirección solicitado está disponible, INVESTIGACIÓN 512. Esta determinación se hace comprobando si existen entradas de tabla de dispositivo disponibles para el número solicitado de espacios de dirección. Si el número de espacios de dirección solicitado no está disponible, entonces se devuelve un código de respuesta que indica que hay insuficientes recursos, ETAPA 514. En caso contrario, el procesamiento continúa para habilitar la función PCI.

El identificador provisto se utiliza para localizar una entrada de tabla de función, ETAPA 516. Por ejemplo, se utilizan uno o más bits designados del identificador como índice en la tabla de función para localizar la entrada de tabla de función particular. Como respuesta a la localización de la entrada de tabla de función apropiada, se hace una determinación de si la función está habilitada, INVESTIGACIÓN 518. Esta determinación se hace comprobando el indicador de habilitación en la entrada de tabla de función. Si la función ya está habilitada (es decir, el indicador está establecido a uno), entonces se devuelve un código de respuesta que indica que la función de PCI ya está en el estado solicitado, ETAPA 520.

Si la función no está todavía habilitada, el procesamiento continúa con la determinación de si la función está en un estado de error permanente, INVESTIGACIÓN 522. Si el indicador de estado de error permanente en la entrada de tabla de función indica que está en un estado de error permanente, entonces se devuelve un código de respuesta indicándolo, ETAPA 524. Sin embargo, si la función no está en un estado de error permanente, se hace una determinación adicional de si la recuperación del error se ha iniciado para la función, INVESTIGACIÓN 526. Si el indicador de recuperación iniciada en la entrada de tabla de función está establecido, entonces se proporciona un código de respuesta que indica que la recuperación se ha iniciado, ETAPA 528. En caso contrario, se hace una investigación de si la función PCI está ocupada, INVESTIGACIÓN 530. De nuevo, si una comprobación del indicador ocupado en la entrada de tabla de función indica que la función PCI está ocupada, entonces se proporciona tal indicación, ETAPA 532. Sin embargo, si la función PCI no está en el estado de error permanente, la recuperación no está iniciada y no está ocupada, entonces se realiza una investigación adicional de si se permite al sistema operativo habilitar esta función PCI, ETAPA 534. Si no se le permite en base al indicador de permiso de entrada de tabla de

función, entonces se proporciona un código de respuesta que indica una acción no autorizada, ETAPA 536. Sin embargo, si todas las pruebas se han superado con éxito, entonces se realiza una determinación adicional de si hay algún DTEs disponible para esta función PCI, INVESTIGACIÓN 538. Como ejemplos, la determinación de que los DTEs están disponibles se pueden basar en que los DTEs no están actualmente habilitados en el concentrador de I/O. Adicionalmente, se podrá aplicar la política para limitar más el número de DTEs disponibles para un sistema operativo dado o una partición lógica. Cualquier DTE que esté accesible al adaptador puede ser asignado. Si no hay DTEs disponibles, entonces se devuelve un código de respuesta que indica que uno o más DTEs solicitados no están disponibles, ETAPA 540.

Si los DTEs están disponibles, entonces el número de DTEs correspondiente al número solicitado de espacios de dirección son asignados y habilitados ETAPA 542. En un ejemplo, la habilitación incluye establecer el indicador de habilitación en cada DTE que va a ser habilitado. Además, la habilitación incluye, en este ejemplo, establecer la CAM para proporcionar un índice a cada DTE. Por ejemplo para cada DTE, se carga una entrada en la CAM con el índice.

Además, los DTEs son asociados con la entrada de tabla de función, ETAPA 544. Esto incluye, por ejemplo, incluir cada índice DTE en la entrada de tabla de función. La función es entonces marcada como habilitada estableciendo el indicador de habilitación en la entrada de tabla de función, ETAPA 546. Además, se establece el bit de habilitación en el identificador, y el número de caso es actualizado, ETAPA 548. Este identificador habilitado es entonces devuelto, ETAPA 550, permitiendo el uso del adaptador PCI. Por ejemplo, como respuesta a la habilitación de la función, se pueden realizar el registro de las traducciones de dirección e interrupciones, se pueden realizar las operaciones DMA por la función PCI; se pueden solicitar interrupciones por la función; y/o instrucciones de controles de función de carga, almacenamiento, bloque de almacenamiento y/o modificación (por ejemplo, Carga PCI, Bloque de Almacenamiento PCI, Controles de Función PCI de Modificación) se pueden emitir a la función.

Una realización de la lógica para deshabilitar una función PCI se describe con referencia a la Fig. 6. En este ejemplo, el comando de función PCI establecido es solicitado a través de la instrucción de Procesador Lógico de Llamada en la que el código de operación está establecida para deshabilitar; sin embargo, en otras realizaciones, dicha instrucción no se utiliza. En un ejemplo es el sistema operativo o un controlador de dispositivo del sistema operativo en que realiza esta lógica.

Haciendo referencia a la Fig. 6, inicialmente, se hace una determinación de si el identificador proporcionado en el bloque de solicitud para la instrucción de Procesador Lógico de Llamada es un identificador conocido. INVESTIGACIÓN 600. Por ejemplo, se hace una comprobación de si el identificador apunta a una entrada válida en la tabla de función. Si el identificador apunta a una entrada válida, entonces el identificador es un identificador conocido. Si no, entonces se proporciona un código de respuesta que indica un identificador desconocido, ETAPA 602. Sin embargo, si el identificador es conocido, entonces se realiza una determinación adicional de si el identificador ya está deshabilitado, INVESTIGACIÓN 604. Si el indicador de habilitación del identificador indica que el identificador ya está deshabilitado, entonces se proporciona un código de respuesta indicándolo. Por el contrario, si el identificador es conocido y está habilitado, el identificador es válido para una operación de deshabilitación y se utiliza para localizar la entrada de tabla de función, ETAPA 608.

Como respuesta a la obtención de la entrada de tabla de función, se hace una determinación de si la función ya está deshabilitada como se indica mediante el indicador de habilitación en la entrada de tabla de función, ETAPA 610. Si el indicador no está establecido (es decir, el indicador de habilitación = 0) entonces se proporciona un código de respuesta que indica que la función ya está deshabilitada, ETAPA 612.

Si el indicador está establecido (por ejemplo, habilitación = 1), entonces se realiza una determinación de si la función está en el estado de error permanente, INVESTIGACIÓN 614. Si está en un estado de error permanente, entonces se proporciona un código de respuesta que indica error, ETAPA 616. Si la recuperación de error está iniciada, entonces se proporciona un código de respuesta indicándolo, ETAPA 620. Si la recuperación de error no está iniciada, entonces se realiza una determinación de si la función PCI está ocupada, INVESTIGACIÓN 62. Si está ocupada, entonces se proporciona un código de respuesta promocionando esto, ETAPA 624. En caso contrario, se realiza una determinación de si el sistema operativo está autorizado para expedir este comando de deshabilitación, ETAPA 626. Esta determinación se hace, por ejemplo, comprobando el indicador de permiso en la tabla de entrada de función, así como comparando el número de caso en el identificador con el número de caso en la entrada de tabla de función. Si no son iguales, entonces se realiza una solicitud para deshabilitar un caso diferente de la función que estaba habilitada. Si el indicador de permiso indica no permitido o los números de caso no son iguales, el sistema operativo o está autorizado y se proporciona un código de respuesta indicando no autorización, ETAPA 628. Sin embargo, si el indicador de permiso especifica permitido y los números de caso son iguales, el sistema operativo está autorizado.

Si todas las comprobaciones son exitosas, entonces la función es deshabilitada, ETAPA 630. En un ejemplo, esto incluye establecer el indicador de habilitación en la entrada de tabla de función a cero (o por el contrario a un estado desactivado). Después, se borran los parámetros de registro en los DTEs asociados con esta función PCI, ETAPA 632, y los DTEs son liberados para ser utilizados por otras funciones PCI, ETAPA 634. Por ejemplo, el bit de habilitación en el DTE se borra y la entrada de CAM asociada con el DTE se renueva. Además, el indicador de

habilitación en el identificador es restablecido a cero (o algún otro valor que indica deshabilitado o desconectado), ETAPA 636, y el identificador deshabilitado es devuelto, ETAPA 638.

En una realización adicional, si uno o más de las pruebas en las INVESTIGACIONES 614, 618 y 622 fallan, entonces la deshabilitación continúa y se puede proporcionar un código de respuesta indicándolo.

5 Lo que se ha descrito con detalle anteriormente es una capacidad de habilitación/deshabilitación de una función PCI. Esta capacidad es independiente del dispositivo desde el punto de vista del sistema operativo, y proporciona un alto nivel de detalle de control, en la que un sistema operativo es capaz de habilitar o deshabilitar una función PCI. Como respuesta a la deshabilitación de la función, otros sistemas operativos pueden habilitar la función. Esto permite que múltiples sistemas operativos (por ejemplo un entorno particionado lógicamente) compartan funciones de adaptador.

10 En las realizaciones descritas aquí, los adaptadores son adaptadores PCI. PCI, como se ha utilizado aquí se refiere a cualquier adaptador implementado de acuerdo con la especificación basada en PCI como está definida por el "Peripheral Component Interconnect Special Interest Group (PCI-SIG)", incluyendo pero no limitado a PCI o PCIe. En un ejemplo particular, la Interconexión de Componente Periférico Express (PCIe) es una norma de interconexión de nivel de componente que define un protocolo de comunicación bidireccional para transacciones entre adaptadores I/O y sistemas principales. Las comunicaciones PCIe están encapsuladas en paquetes de acuerdo con la norma PCIe para transmisión en un bus PCIe. Las transacciones que originan un sistema principal y que terminan en adaptadores I/O se denominan transacciones de límite inferior. La tipología PCIe está basada en enlaces unidireccionales punto a punto que están emparejados (por ejemplo el enlace de límite superior, un enlace de límite inferior) para formar un bus PCIe.

20 La norma PCIe se mantiene y expide por el PCI-SIG, como se ha mencionado en la sección de Antecedentes.

Como apreciarán los expertos en la técnica, los aspectos de la presente invención se pueden implementar como un sistema, método o producir de programa de ordenador. Por consiguiente, los aspectos de la presente invención pueden adoptar la forma de un realización de hardware enteramente, una realización de software enteramente (incluyendo firmware, software residente, microcódigo, etc.) o una realización que combine aspectos de software y hardware que pueden todos estar generalmente referidos aquí como un "circuito", "módulo" o "sistema". Además, los aspectos de la presente invención pueden adoptar la forma de un producto de programa de ordenador implementado en uno o más medios leíbles con ordenador que tengan un código de programa leíble implementados en los mismos.

30 Cualquier combinación de uno o más medios leíbles de ordenador se pueden utilizar. El medio leíble por ordenador puede ser un medio de almacenamiento leíble con ordenador. Un medio de almacenamiento leíble con ordenador puede ser, por ejemplo, pero no se limita a, sistema, aparato o dispositivo electrónico, magnético, óptico, electromagnético, de infrarrojos o semiconductor, o cualquier combinación adecuada de los mismos. Ejemplos más específicos (una lista no exhaustiva) de medios de almacenamiento leíbles con ordenador incluyen los siguientes: una conexión eléctrica que tenga 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, borrable, (EPROM o memoria temporal), una fibra óptica, una memoria de solo lectura de disco compacto portátil (CD-ROM), un dispositivo de almacenamiento óptico, un dispositivo de almacenamiento magnético, o cualquier combinación adecuada de los mismos. En el contexto de este documento, un medio de almacenamiento leíble con ordenador puede ser un medio tangible que puede contener o almacenar un programa para utilizar por o en combinación con un sistema, aparato, o dispositivo de ejecución de instrucción.

Haciendo ahora referencia a la Fig. 7, en un ejemplo, un producto de programa de ordenador 700 incluye, por ejemplo, uno o más medios de almacenamiento leíbles con ordenador 702 para almacenar medios de código de programa leíble con ordenador o lógica 704 en el mismo para proporcionar y facilitar uno o más aspectos de la presente invención.

45 El código de programa implementado en un medios leíble con ordenador se puede transmitir utilizando un medio apropiado, que incluye, pero no se limita a, inalámbrico, línea de cable, cable de fibra óptica, RF, etc., o cualquier combinación apropiada de los mismos.

50 El código de programa de ordenador para realizar las operaciones para aspectos de la presente invención se puede escribir en cualquier combinación de uno o más lenguajes de programación, que incluyen un lenguaje de programación orientado a un objetivo, tal como Java, Smalltalk, C++ o similar, y lenguajes de programación de procedimiento convencional, tales como el lenguaje de programación "C", ensamblador o lenguajes de programación similares. El código de programa se puede ejecutar totalmente en el ordenador del usuario, parcialmente en el ordenador del usuario, como un único paquete de software, parcialmente en el ordenador del usuario y parcialmente en un ordenador remoto o totalmente en el ordenador remoto o servidor. En este 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) una red de área ancha (WAN), o la conexión se puede hacer con un ordenador externo (por ejemplo, a través de Internet utilizando un Proveedor de Servicio de Internet).

Los aspectos de la invención se han descrito aquí como referencia a las ilustraciones de la carta de flujo y/o diagramas de bloques de los métodos, aparatos (sistemas) y productos de programa de ordenador de acuerdo con las realizaciones de la invención. Se entenderá que cada bloque de ilustraciones de carta de flujo y/o diagrama de bloques, y combinaciones de bloques en las ilustraciones de carta de flujo y/o diagramas de bloques, se pueden implementar mediante instrucciones de programa de ordenador. Estas instrucciones de programa de ordenador se pueden proporcionar a un procesador de un ordenador de finalidad general, ordenador de finalidad especial, o otro aparato de procesamiento de datos programable para producir una máquina, de manera que las instrucciones que se ejecutan a través del procesador del ordenador u otro aparato de procesamiento de datos programable, creen medios para implementar las funciones/actos especificados en la carta de flujo y/o bloque o bloques de diagrama de bloques.

Estas instrucciones de programa de ordenador también se pueden almacenar en un medio leíble con ordenador que puede dirigir un ordenador, otros aparatos de procesamiento de datos programables, u otros dispositivos para que funciones de una manera particular, de manera que las instrucciones almacenadas en el medio leíble con ordenador producen un artículo de fabricación que incluye instrucciones que implementan la función/acto especificado en la carta de flujo y/o bloque o bloques de diagrama de bloques.

Las instrucciones de programa de ordenador también se pueden cargar en un ordenador, en otro aparato de procesamiento de datos programable, o en otro dispositivo para hacer que una serie de etapas en el ordenador sean realizadas en el ordenador, otro aparato programable u otro dispositivo para producir un proceso implementado de ordenador de manera que las instrucciones que se ejecutan sobre el ordenador u otro aparato programable proporcionan procesos para implementar las funciones/actos especificados en la carta de flujo y/o bloque o bloques del diagrama de bloques.

La carta de flujo y los diagramas de bloques de las figuras ilustran la arquitectura, funcionalidad, y funcionamiento de posibles implementaciones de sistemas, métodos y productos de programa de ordenador de acuerdo con las distintas realizaciones de la presente invención. En este sentido, cada bloque en la carta de flujo de diagramas de flujo pueden representar un módulo, segmento, o parte de código, que comprende una o más instrucciones ejecutables para implementar la(s) función(es) lógicas(s) especificada(s). Se ha de observar que, en algunas implementaciones alternativas, las funciones anotadas en el bloque pueden ocurrir fuera del orden anotado en las figuras. Por ejemplo, dos bloques mostrados en sucesión pueden, de hecho, ser ejecutados de forma sustancialmente concurrente, o los bloques pueden ser a veces ejecutados en orden inverso, dependiendo de la funcionalidad implicada. Se observará también que cada bloque de los diagramas de bloques y/o ilustración de carta de flujo, y combinaciones de bloques en los diagramas de bloques y/o ilustración de carta de flujo, se puede implementar mediante sistemas basados en hardware de finalidad espacial que realizan las funciones o actos especificados, o combinaciones de hardware de finalidad especial e instrucciones de ordenador.

Además de lo anterior, se pueden proporcionar, ofrecer, desplegar, gestionar, etc. uno o más aspectos de la presente invención por un proveedor de servicio que ofrece gestión de los entornos del consumidor. Por ejemplo, el proveedor de servicios puede crear, mantener, dar soporte, etc., al código de ordenador y/o estructura de ordenadores que realiza uno o más aspectos de la presente invención para uno o más clientes. Como contrapartida, el proveedor de servicios puede recibir pago del consumidor bajo una inscripción y/o acuerdo libre, como ejemplos. Adicional o alternativamente, el proveedor de servicios puede recibir el pago de la venta del contenido de publicidad a una o más terceras partes.

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

Como un aspecto más de la presente invención, se puede desplegar una infraestructura de ordenador que comprende la integración del código leíble con 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 más de la presente invención, se puede proporcionar un proceso para integrar la infraestructura de ordenador que comprende integrar el código leíble con ordenador en un sistema de ordenador. El sistema de ordenador comprende un medio leíble con 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 se han descrito anteriormente distintas realizaciones, éstas son sólo ejemplos. Por ejemplo, las realizaciones de ordenador de otras arquitecturas pueden incorporar y utilizar uno o más aspectos de la presente invención. Como ejemplos, otros servidores distintos de servidores Systems z®, tales como servidores Power Systems u otros servidores ofrecidos por International Business Machines Corporation, o servidores de otras compañías pueden incluir, utilizar y/o beneficiarse de uno o más aspectos de la presente invención. Además, aunque en el ejemplo presente, los adaptadores y concentrador PCI son considerados una parte del servidor, en otras realizaciones, no tiene que ser necesariamente considerados como parte del servidor, sino que simplemente pueden ser considerados conectados a la memoria del sistema y/o a otros componentes de un entorno de

ordenadores. El entorno de ordenadores no necesita ser un servidor. Además, aunque se han descrito tablas, cualquier estructura de datos se puede utilizar y el término tabla es para incluir todas esas estructuras de datos. Aún más, aunque los adaptadores están basados en PCI, uno o más aspectos de la presente invención se pueden utilizar con otros adaptadores u otros componentes I/O. El adaptador y el adaptador PCI son sólo ejemplo. Además, la FTE o los parámetros de la FTE se puede localizar y mantener en otra memoria segura incluyendo, por ejemplo en el hardware (por ejemplo el hardware de función PCI). Las DTE, FTE y/o identificador pueden incluir más, menos o diferente información, así como la solicitud y/o bloque de respuesta. Adicionalmente, la instrucción de Procesador Lógico de Llamada puede incluir más, menos o diferentes campos. Son posibles muchas otras variaciones.

Además, otros tipos de entornos de ordenador pueden beneficiarse de uno o más aspectos de la presente invención. Como ejemplo, se puede utilizar un sistema de procesamiento de datos adecuado para almacenar y/o ejecutar un código de programa que incluye al menos dos procesadores conectados directamente o indirectamente a los elementos de memoria a través de un bus de sistema. Los elementos de memoria incluyen, por ejemplo, memoria local empleada durante la ejecución real del código de programa, almacenamiento de grandes cantidades, y memoria caché que proporcionan almacenamiento temporal de al menos algún código de programa con el fin de reducir el número de veces que el código debe ser recuperado del almacenamiento de grandes cantidades durante la ejecución.

Los dispositivos de Entrada/Salida o dispositivos I/O (incluyendo, pero no limitándose a, teclados, pantallas, dispositivos de apuntar, DASD, cinta, CDs, DVDs, dispositivos de pulgar y otros medios de memoria, etc.) se pueden conectar al sistema o bien directamente o bien a través de controladores I/O de intervención. Los adaptadores de red también se pueden acoplar al sistema para habilitar el sistema de procesamiento de datos para conectarlo a otros sistemas de procesamiento de datos o impresoras remotas o dispositivos de almacenamiento a través de redes de intervención privadas o públicas. Modems, modems de cable, y tarjetas de Ethernet son sólo unos pocos de los tipos disponibles de adaptadores de red.

Haciendo referencia a la Fig. 8, están representados los componentes representativos del sistema de Ordenador Principal 5000 para implementar uno o más aspectos de la presente invención. El ordenador principal representativo 5000 comprende una o más CPUs 5001 en comunicación con la memoria del ordenador (es decir el almacenamiento central) 5002, así como interfaces I/O para dispositivos de medios de almacenamiento 5011 y redes 5010 para comunicar con otros ordenadores o SANs y similares. La CPU 5001 cumple con la arquitectura teniendo un conjunto de instrucciones diseñadas y funcionalidad diseñada. La CPU 5001 puede tener traducción de dirección dinámica (DAT) 5003 para transformar direcciones de programa (direcciones virtuales) en direcciones reales de memoria. Una DAT típicamente incluye un búfer de traducción anticipada (TLB) 5007 para capturar traducciones de manera que el último acceso al bloque de memoria de ordenador 5002 no requiere el retraso de la traducción de dirección. Típicamente una caché 5009 se emplea entre la memoria de ordenador 5002 y el procesador 5001. La caché 5009 puede ser jerárquico teniendo una caché disponible para más de una CPU y cachés más pequeñas, más rápidas (nivel inferior) entre la caché grande y cada CPU. En algunas implementaciones, las cachés de nivel inferior están divididas para proporcionar cachés de nivel bajo separadas para recoger la traducción y acceso de datos. En una realización, una instrucción es recogida de la memoria 5002 mediante una unidad de recogida de instrucción 5004 a través de una caché 5009. La instrucción es decodificada en una unidad de descodificación de instrucción 5006 y enviada (con otras instrucciones en algunas realizaciones) a la unidad o unidades de ejecución de instrucción 5008. Típicamente se emplean varias unidades de ejecución 5008, por ejemplo una unidad de ejecución aritmética, una unidad de ejecución de punto flotante y una unidad de ejecución de instrucción de ramal. La instrucción es ejecutada por la unidad de ejecución, operandos de acceso de los registros específicos de instrucción o memoria según se necesite. Si se va a acceder a un operando (cargado o almacenado) desde la memoria 5002, una unidad de carga/almacenamiento 5005 típicamente maneja en 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 mediante una combinación de ambos.

Como se observa, un sistema de ordenador incluye información en almacenamiento local (o principal), así como direccionamiento, protección y referencia y cambio de registro. Algunos aspectos del direccionamiento incluyen el formato de dirección, el concepto de espacios de dirección, los distintos tipos de direcciones, y la manera en la que un tipo de dirección es traducido a otro tipo de dirección. Algún almacenamiento principal incluye localizaciones de almacenamiento permanentemente asignadas. El almacenamiento principal proporciona los sistemas con almacenamiento de datos directamente direccionable de acceso rápido. Tanto los datos como los programas van a ser cargados en el almacenamiento principal (desde los dispositivos de entrada) antes de puedan ser procesados.

El almacenamiento principal incluye uno o más almacenamiento de búfer más pequeños de acceso rápido a veces llamadas cachés. Una caché está típicamente asociada físicamente con una CPU o un procesador I/O. Los efectos, excepto en el rendimiento, de la construcción física y uso de los distintos medios de almacenamiento generalmente no son observables por el programa.

Se pueden mantener cachés separadas para instrucciones y operandos de datos. La información dentro de la caché se mantiene en bytes contiguos en una delimitación integral llamada bloque de caché o línea de caché (o línea para abreviar). Un moldeo puede proporcionar una instrucción de ATRIBUTO DE CACHÉ DE EXTRACCIÓN que devuelve el tamaño de una línea de caché de bytes. Un modelo también puede proporcionar instrucciones de

DATOS DE PRERECOGIDA e instrucciones de DATOS DE PRERECOGIDA RELATIVAMENTE LARGAS que efectúan la prerecogida del almacenamiento en los datos o caché de instrucción o liberar los datos de la caché.

El almacenamiento de ve como una cadena horizontal larga de bits. Para la mayoría de las operaciones, el acceso de almacenamiento avanza en una secuencia de izquierda a derecha. La cadena de bits está subdividida en unidades de ocho bits. Una unidad de ocho bits se denomina byte, que es el bloque de construcción básico de todos los formatos de información. Cada localización de byte en el almacenamiento está identificada por un único número entero no negativo, que es la dirección de esa localización de byte o, simplemente la dirección de byte. Las localizaciones de byte adyacentes tienen direcciones consecutivas, empezando por 0 en la izquierda y continuando en una secuencia de izquierda a derecha. Las direcciones son números enteros binarios sin signo y son 24, 31 ó 64 bits.

La información es transmitida entre el almacenamiento y la CPU o un subsistema de canal un byte, o un grupo de bytes, de una vez. A menos que se especifique lo contrario, en, por ejemplo la z/Architecture®, un grupo de bytes en el almacenamiento es dirigido por el byte situada más a la izquierda del grupo. El número de bytes en el grupo está o bien implicado o bien especificado explícitamente por la operación que se va a realizar. Cuando se utiliza en una operación de CPU, un grupo de bytes se denomina campo. Dentro de cada grupo de bytes en, por ejemplo la z/Architecture®, los bits están numerados en una secuencia de izquierda a derecha. En la z/Architecture®, los bits situados más a la izquierda son a veces denominados como bits de "alto orden" y los bits situados más a la derecha como bits de "bajo orden". Sin embargo los números de bit no son direcciones de almacenamiento.

Sólo los bytes pueden ser direccionados. Para operar en los bits individuales de un byte en el almacenamiento, se accede a todo el byte. Los bits de un byte están numerados de 0 a 7, de izquierda a derecha (en, por ejemplo la z/Architecture®). Los bits en una dirección pueden estar numerados de 8-31 ó 40-63 para direcciones de 24 bit, ó 1-31 o 33-63 para direcciones de 31 bits; están numerados de 0-63 para direcciones de 64 bits. Dentro de cualquier otro formato de longitud fija de múltiples bytes, los bits que forman el formato están numerados consecutivamente empezando por 0. Para fines de detección de errores, y preferiblemente para corrección, uno o más bits pueden ser transmitidos con cada byte o con cada grupo de bytes. Tales bits de comprobación son generados automáticamente por la máquina y no pueden ser controlados directamente por el programa. Las capacidades de almacenamiento están expresadas en número de bytes. Cuando la longitud de un campo de operando de almacenamiento es implicada por el código de operación de una instrucción, se dice que el campo tiene longitud fija, la cual puede ser uno, dos, cuatro, ocho o dieciséis bytes. Campos más grandes pueden ser implicados para algunas instrucciones. Cuando la longitud de un campo de operando de almacenamiento no está implicada pero está establecida explícitamente, se dice que el campo tiene una longitud variable. Los operandos de longitud variable pueden variar de longitud en incrementos de un byte (o con algunas instrucciones, en múltiplos de dos bytes u otros múltiplos). Cuando la información es almacenada, los contenidos de sólo son remplazadas aquellas localizaciones de byte que están incluidas en el campo designado, incluso aunque la anchura de la trayectoria física para el almacenamiento sea mayor que la longitud del campo que está siendo almacenado.

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

En dispositivos que implementan cachés separadas para instrucciones y operandos de datos, se puede experimentar un retraso significativo si el programa almacena una línea caché desde la que las instrucciones son posteriormente recogidas, independientemente de si el almacenamiento alerta a las instrucciones que son posteriormente recogidas.

En una realización, la invención se puede llevar a la práctica mediante un software (a veces denominado código interno autorizado, firmware, micro-código, mili-código, pico-código y similar, cualquiera de los cuales estaría de acuerdo con la presente invención). Haciendo referencia a la Fig. 8, al código de programa de software que encarna la presente invención se accede típicamente mediante un procesador 5001 del sistema principal 5000 desde dispositivos de medios de almacenamiento de larga duración 5011, tales como un accionador de CD-ROM, un accionador de cinta o un disco duro. El código de programa de software se puede llevar a la práctica en cualquier variedad de medios conocidos para utilizar con un sistema de procesamiento de datos, tales como un disquete, disco duro o CD-ROM. El código puede ser distribuido en tales medios, o puede ser distribuido a los usuarios desde la memoria de ordenador 5002 el almacenamiento de un sistema de ordenados sobre una red 5010 a otros sistemas de ordenador para utilizar por los usuarios de tales otros sistemas.

El código de programa de software incluye un sistema operativo que controla la función e interacción de varios componentes de ordenados y uno o más programas de aplicación. El código de programa normalmente está paginado desde el dispositivo de medio de almacenamiento 5011 hasta el almacenamiento de ordenador de relativamente alta velocidad 5002 en donde está disponible para el procesamiento por el procesador 5001. Las técnicas y métodos para incorporar el código de programa de software en la memoria, en el medio físico, y/o distribuir el código de software a través de redes son bien conocidos y no serán descritos adicionalmente aquí. El código de programa, cuando se crea y alacena en un medio tangible (que incluye pero no se limita a módulos de memoria electrónica (RAM), memoria temporal, Discos Compactos (CDs), DVDs, Cintas magnéticas y similares a menudo es denominado como "producto de programa de ordenador". El medio de producto de programa de ordenador es típicamente leíble por una unidad de procesamiento preferiblemente en un sistema para la ejecución del circuito de procesamiento.

La Fig. 9 ilustra una estación de trabajo representativa en un sistema de hardware de servidor en el que se pueden llevar a la práctica la presente invención. El sistema 5020 de la Fig. 9 comprende un sistema de ordenador de base representativo 5021, tal como un ordenador personal, una estación de trabajo o un servidor que incluyen dispositivos periféricos opcionales. El sistema de ordenador base 5021 incluye uno o más procesadores 5026 y un bus empleado en conectar y habilitar la comunicación entre el(los) procesador(es) 5026 y los demás componentes del sistema 5021 de acuerdo con técnicas conocidas. El bus conecta el procesador 5026 a la memoria 5025 y al almacenamiento de larga duración 5027 que puede incluir un disco duro (que incluye cualquier medio magnético, CD, DVD y memoria temporal, por ejemplo) o un accionador de cinta, por ejemplo. El sistema 5021 también podría incluir una 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/o otros dispositivos de interfaz, que pueden ser cualquier dispositivo de interfaz de usuario, tal como una pantalla táctil, un teclado de entrada digitalizado, etc. El bus también conecta un dispositivo de presentación 5022, tal como una pantalla o monitor LCD, al microprocesador 5026 a través de un adaptador de presentación.

El sistema 5021 puede comunicarse con otros ordenadores y redes de ordenadores a través de un adaptador de red capaz de comunicarse 5028 con una red 5029. Adaptadores de red a modo de ejemplo son canales de comunicación, redes en anillo, Ethernet o módems. Alternativamente, el sistema 5021 puede comunicarse utilizando una interfaz inalámbrica, tal como una tarjeta CDPD (datos de paquete digital celular). El sistema 5021 puede estar asociado con tales otros ordenadores en la Red de Área Local (LAN) o una Red de Área Ancha (WAN), o el sistema 5021 puede ser un cliente en una configuración de cliente/servidor con otro componente, etc. Todas estas configuraciones, así como el hardware o software de comunicaciones apropiado se conocen en la técnica.

La Fig. 10 ilustra una red de procesamiento de datos 5040 en la que se puede llevar a la práctica la presente invención. La red de procesamiento de datos 5040 puede incluir una pluralidad de redes individuales, tales como una red inalámbrica y una red con cable, cada una de las cuales puede incluir una pluralidad de estaciones de trabajo individuales 5041, 5042, 5043, 5044. Adicionalmente, los expertos en la técnica apreciarán que se pueden incluir una o más LANs, en donde una LAN puede comprender una pluralidad de estaciones de trabajo inteligentes conectadas a un procesador principal.

Haciendo todavía referencia a la Fig. 10, las redes también pueden incluir ordenadores centrales de grandes prestaciones ("mainframe") o servidores, tales como un ordenador de pasarela (servidor de cliente 5046) o servidor de aplicación (servidor remoto 5048 que puede acceder a un depósito de datos y también se puede acceder a el directamente desde una estación de trabajo 5045). Un ordenador de pasarela 5046 sirve como punto de entrada en cada red individual. Se necesita una pasarela cuando se conecta un protocolo de red a otro. La pasarela 5046 puede estar preferiblemente conectada a otra red (Internet 5047, por ejemplo) por medio de un enlace de comunicaciones. La pasarela 5046 también puede estar directamente conectada a una o más estaciones de trabajo 5041, 5042, 5043, 5044 utilizando un enlace de comunicaciones. El ordenador de pasarela también puede ser implementado utilizando un servidor IBM eServer™ System z® disponible de International Business Machines Corporation.

Haciendo referencia simultáneamente a la Fig. 9 y a la Fig. 10, se puede acceder al código de programación de software que puede encarnar la presente invención mediante el procesador 5026 del sistema 5020 desde los medios de almacenamiento de larga dirección 5027, tales como accionador de CD-ROM o disco duro. El código de programación de software se puede llevar a la incorporar en cualquiera de una variedad de medios conocidos para utilizar con un sistema de procesamiento de datos, tales como un disquete, disco duro, o CD-ROM. El código se puede distribuir sobre tal medio, o se puede distribuir a los usuarios 5050, 5051 desde la memoria o almacenamiento de un sistema de ordenador en una red a otros sistemas de ordenador para utilizar por los usuarios de dichos otros sistemas.

Alternativamente, el código de programación se puede incorporar en la memoria 5025, y se puede acceder a él mediante el procesador 5026 utilizando el bus de procesador. Tal código de programación incluye un sistema operativo que controla la función de interacción de los distintos componentes de ordenador y uno o más programas de ordenador 5032. El código de programa esta normalmente paginado desde los medios de almacenamiento 5027 hasta la memoria de alta velocidad 5025 en la que está disponible para el procesamiento por el procesador 5026. Las técnicas y métodos para incorporar el código de programación de software en la memoria, o en el medio físico, y/o distribuir el código de software a través de las redes son bien conocidas y no serán descritas adicionalmente

aquí. El código de programa, cuando se creó y almacenó en un medio tangible (que incluye pero no se limita a módulos de memoria electrónicos (RAM), memoria temporal, Discos Compactos (CDs), DVDs, Cinta magnética o similares es a menudo denominado "producto de programa de ordenador". El medio de producto de programa de ordenador típicamente es leíble mediante un circuito de procesamiento preferiblemente en un sistema de ordenador para la ejecución por el circuito de procesamiento.

La caché que está más fácilmente disponible para el procesador (normalmente más rápida y más pequeña que otras cachés del procesador) en la caché más baja (L1 o nivel 1) y el almacenamiento principal (memoria principal) es la caché de nivel más alto (L3 si hay 3 niveles). La caché de nivel más inferior a menudo está dividida en una caché de instrucción (I-Caché) que lleva las instrucciones de la máquina que van a ser ejecutadas y una caché de datos (D-caché) que lleva los operandos de datos.

Haciendo referencia a la Fig. 11, una realización de procesador a modo de ejemplo se muestra para el procesador 5026. Típicamente uno o más niveles de caché 5033 se emplean para almacenar en búfer los bloques de memoria con el fin de mejorar el rendimiento del procesador. La caché 5053 un búfer de alta velocidad que lleva datos líneas de caché de memoria que probablemente van a ser utilizados. Las líneas de caché típicas son 64, 128 ó 256 bytes de datos de memoria. A menudo se emplean cachés separadas para almacenar temporalmente instrucciones en lugar de para almacenar temporalmente datos. La coherencia de caché (sincronización de copias de líneas en memoria y las cachés) a menudo es provisto por varios algoritmos de "vigilancia" bien conocidos en la técnica. El almacenamiento de memoria principal 5025 de un sistema de procesador es a menudo denominado caché. En un sistema procesador que tiene 4 niveles de caché 5053, almacenamiento principal 5025 a veces se denomina como la caché de nivel % (L5) dado que es típicamente más rápido y solo lleva una parte del almacenamiento no volátil (DASD, cinta, etc.) que está disponible para un sistema de ordenador. El almacenamiento principal 5025 "cachés" páginas de datos paginadas dentro y fuera del almacenamiento principal 5025 mediante el sistema operativo.

Un contador de programa (contador de instrucción) 5061 mantiene la pista de la dirección de la instrucción actual que va a ser ejecutada. Un contador de programa en un procesador z/Architecture® es de 64 bits y puede estar truncado a 31 ó 24 bits para soportar los límites de direccionamiento previos. Un contador de programa está típicamente incorporado en una PSW (palabras de estado de programa) de un ordenador de manera que persiste durante la conmutación del contexto. De este modo, un programa en proceso, que tiene un valor de contador de programa, puede ser interrumpido por, por ejemplo, el sistema operativo (conmutador de contexto desde el entorno de programa al entorno de sistema operativo). La PSW del programa mantiene el valor de contador de programa mientras que el programa no está activo, y el contador de programa (en la PSW) del sistema operativo es utilizado mientras el sistema operativo se está ejecutando. Típicamente, el contador de programa es incrementado en una cantidad igual al número de bytes de la instrucción actual. Las instrucciones RISC (Computación de Ajuste de Instrucción reducida) son típicamente longitudes fijas mientras que las instrucciones CISC (Computación de Ajuste de Instrucción Compleja) típicamente son de longitud variable. Las instrucciones del IBM z/Architecture® son instrucciones CISC que tienen una longitud de 2, 4 ó 6 bytes. El contador de programa 5061 es modificado o bien por una operación de conmutación de contexto o bien por una operación de toma de ramificación de una instrucción de ramificación, por ejemplo. En un contexto de operación de conmutación, el valor de contador de programa actual es guardado en la palabra de estado de programa junto con otra información de estado acerca del programa que está siendo ejecutado (tal como códigos de condición), y un nuevo valor de contador de programa es cargado señalando a una instrucción de un nuevo módulo de programa que va ser ejecutado. Una operación de toma de ramificación 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 ramificación en el contador de programa 5061.

Típicamente, una unidad de recogida de instrucción 5055 se emplea para recoger las instrucciones en nombre del procesador 5026. La unidad de recogida o bien recoge "instrucciones secuenciales siguientes", instrucciones objetivo de instrucciones de toma de ramificación, o bien primero instrucciones de un programa seguid de un conmutador de contexto. Las unidades de recogida de instrucción modernas a menudo emplean técnicas de prerecogida para especulativamente prerecoger instrucciones en base a la probabilidad de que las instrucciones prerecogidas puedan ser utilizadas. Por ejemplo, una unidad de recogida puede recoger 16 bytes de instrucción lo que incluye la siguiente instrucción secuencial y bytes adicionales a las instrucciones secuenciales adicionales.

Las instrucciones recogidas son entonces ejecutadas por el procesador 5026. En una realización, la(s) instrucción(es) recogidas son hechas pasar a una unidad de envío 5056 de la unidad de recogida. La unidad de envío descodifica la(s) instrucción(es) y envía información acerca de la(s) instrucción(es) decodificada a las unidades apropiadas 5057, 5057, 5060. Una unidad de ejecución 5057 típicamente recibirá información acerca de las instrucciones aritméticas descodificadas procedente de la unidad de recogida de instrucción 5055 y realizará operaciones aritméticas en los operandos de acuerdo con el código de operación ("opcode") de la instrucción. Los operandos son proporcionados a la unidad de ejecución 5057 preferiblemente o bien desde la memoria 5025, registros diseñados 5059 o desde un campo inmediato de la instrucción que está siendo ejecutada. Los resultados de la ejecución, cuando se almacenan, son almacenados o bien en la memoria 5025, registros 5059 o bien en el hardware de la máquina (tal como los registros de controla registros PSW y similares).

Un procesador 5026 típicamente tiene una o más unidades 5057, 5058, 5060 para ejecutar la función de la instrucción. Haciendo referencia a la Fig. 12A, una unidad de ejecución 5057 puede comunicar con los registros

generales diseñados 5059, una unidad de decodificación/envío 5056, una unidad de almacenamiento de carga 5060, y otras unidades procesadores 5065 por medio de la lógica de interfaz 5071. Una unidad de ejecución 5057 puede utilizar varios circuitos registros 5067, 5068, 5069 para guardar información sobre la que la unidad de lógica aritmética (ALU) 5066 operará. La ALU realiza operaciones aritméticas tales como sumas, restas, multiplicaciones y divisiones así como funciones lógicas tales como y, o, y exclusivo-o (XOR) girar y desplazar. Preferiblemente la ALU soporta operaciones especializadas que dependen del diseño. Otros circuitos pueden proporcionar otras prestaciones de diseño 5072 incluyendo códigos de condición y lógica de soporte de recuperación, por ejemplo. Típicamente el resultado de una operación de ALU es mantenida en un circuito registro de salida 5079 que puede enviar el resultado a una variedad de otras funciones de procesamiento. Hay muchas configuraciones de unidades procesadoras, la presente descripción sólo está destinada a proporcionar un entendimiento representativo de una realización.

Una instrucción ADD por ejemplo sería ejecutada en una unidad de ejecución 5057 que tienen funcionalidad aritmética y lógica mientras que la instrucción de punto flotante por ejemplo sería ejecutada en una ejecución de punto flotante que tiene capacidad de punto flotante especializada. Preferiblemente, una unidad de ejecución opera en operandos identificados por una instrucción mediante la realización de una función definida de código de operación en los operandos. Por ejemplo, una instrucción ADD puede ser ejecutada por una unidad de ejecución 5057 en los operandos encontrados en dos registros 5059 identificados por los campos registros de la instrucción.

La unidad de ejecución 5057 realiza la adición aritmética en dos operandos y almacena el resultado en un tercer operando en donde el tercer operando puede ser un tercer registro o uno de los dos registros de fuente. la unidad de ejecución preferiblemente utiliza una Unidad Lógica Aritmética (ALU) 5056 que es capaz de realizar una variedad de funciones lógicas tales como Desplazamiento, Rotación, Y, O, XOR así como una variedad de funciones algebraicas incluyendo cualquiera de suma, resta, multiplicación, división. Algunas ALUs están diseñadas para operaciones escalares y algunas para punto flotante. Los datos pueden ser leídos en Big Endian (en donde el byte menos significativo está en la dirección de byte más elevada) o en Little Endian (en donde el byte menos significativo está en la dirección de byte más baja) dependiendo de la arquitectura. La IBM z/Architecture® es Big Endian. Los campos con signo pueden ser signo y magnitud complemento de 1 ó complemento de 2 dependiendo de la arquitectura. Un número de complemento de 2 es ventajoso ya que la ALU no necesita diseñar una capacidad de sustracción ya que o bien un valor negativo o bien un valor positivo en complemento de 2 requiere sólo una adición dentro de la ALU. Los números están comúnmente descritos en taquigrafía, en donde un campo de 12 bits define una dirección de un bloque de 4.096 y comúnmente descrito como un bloque de 4 Kbyte (Kilobyte), por ejemplo.

Haciendo referencia a la Fig. 12B, la información de instrucción de ramificación para ejecutar una instrucción de ramificación es típicamente enviada a una unidad de ramificación 5058 que a menudo emplea un algoritmo de predicción de ramificación tal como una tabla de historia de ramificación 5082 para predecir el resultado de la ramificación antes de que otras operaciones adicionales se completen. El objetivo de la instrucción de ramificación en curso será recogido y especulativamente ejecutado antes de que las operaciones condicionales se completen. Cuando las operaciones condicionales están completadas, las instrucciones de ramificación ejecutadas especulativamente so o bien completadas o bien desechadas en base a las condiciones de la operación condicional y el resultado especulado. Una instrucción de ramificación típica puede examinar los códigos de condición y ramificación a una dirección objetivo son los códigos de condición cumplen los requisitos de ramificación de la instrucción de ramificación, una dirección objetivo se puede calcular en base a varios números que incluyen los encontrados en los campos de registro o un campo inmediato de la instrucción por ejemplo. La unidad de ramificación 5058 puede emplear una ALU 5074 que tiene una pluralidad de circuitos de registro de entrada 5075, 5076, 5077 y un circuito de registro de salida 5080. La unidad de ramificación 5058 puede comunicar con los registros generales 5059, decodificar la unidad de envío 5056 u otros circuitos 5073, por ejemplo.

La ejecución de un grupo de instrucciones se puede interrumpir por una variedad de razones que incluyen una conmutación de contexto iniciado por un sistema operativo, una excepción o error de programa que produce una conmutación de contexto, una señal de interrupción de I/O que produce una conmutación de contexto o actividad multi-hilo de una pluralidad de programas (en un entorno multi-hilo), por ejemplo. Preferiblemente una acción de conmutación de contexto guarda la información de estado acerca de un programa que se está ejecutando actualmente y después carga la información de estado acerca de otro programa que está siendo llamado. La información de estado puede ser guardada en registros de hardware o en la memoria por ejemplo. La información de estado preferiblemente comprende un valor de contador de programa que apunta a una siguiente instrucción que va a ser ejecutada, códigos de condición, información de traducción de memoria y contenido de registro diseñado. Una actividad de conmutación de contexto se puede ejercer por circuitos de hardware, programas de aplicación, programas de sistema operativo o código de firmware (microcódigo, pico-código o código interno con licencia (LIC)) solos o en combinación.

Un procesador accede a los operandos de acuerdo con los métodos definidos de la instrucción. La instrucción puede proporcionar un operando intermedio que utiliza el valor de una parte de la instrucción, puede proporcionar uno o más campos de registro apuntando explícitamente o bien a los registros de finalidad general o bien a los registros de finalidad especial (registros de punto flotante, por ejemplo). La instrucción puede utilizar registros implicados identificados por un campo de código de operación como operandos. La instrucción puede utilizar

posiciones de memoria para los operandos. Una posición de memoria para un operando puede ser proporcionada por un registro, un campo inmediato, o una combinación de registros y campos inmediatos como se ejemplifica mediante la prestación de desplazamiento largo de *z/Architecture*® en donde la instrucción define un registro base, un registro de índice y un campo intermedio (campo de desplazamiento) que son añadidos para proporcionar la dirección del operando en la memoria, por ejemplo. La posición aquí típicamente implica una localización en la memoria principal (almacenamiento principal) a menos que se indique lo contrario.

Haciendo referencia a la Fig. 12C, un procesador accede al almacenamiento utilizando una unidad de carga/almacenamiento 5060. La unidad de carga/almacenamiento 5060 puede realizar una operación de carga obteniendo la dirección del operando objetivo en la memoria 5053 y cargando el operando en un registro 5059 o otra posición de memoria 5053, o puede realizar una operación de almacenamiento obteniendo la dirección del operando objetivo en la memoria 5053 y almacenar los datos obtenidos de un registro 5059 u otra posición de memoria 5053 en la localización del operando objetivo en la memoria 5053. La unidad de carga/almacenamiento 5060 puede ser especulativa o puede acceder a la memoria en una secuencia que está fuera de orden con relación a la secuencia de instrucción, sin embargo la unidad de carga/almacenamiento 5060 es para mantener la apariencia para los programas cuyas instrucciones fueron ejecutadas en orden. Una unidad de carga/almacenamiento 5060 puede comunicarse con los registros generales 5059, la unidad de decodificación/envío 5056, la interfaz de caché/memoria 5053 u otros elementos 5083 y comprende varios circuitos de registro, ALUs 5085 y lógica de control 5090 para calcular la dirección de almacenamiento y para proporcionar una secuencia de cadena de procesos tubería para mantener las operaciones en orden. Algunas operaciones pueden estar fuera de orden pero la unidad de carga/almacenamiento proporciona funcionalidad para hacer que las operaciones fuera de orden aparezcan para el programa que han sido realizadas en orden, como es bien conocido en la técnica.

Preferiblemente, las direcciones que un programa de aplicación “ve” a menudo están referidas como direcciones virtuales. Las direcciones virtuales a veces están denominadas como “direcciones lógicas” y “direcciones efectivas”. Las direcciones virtuales con virtuales ya que son redireccionadas a una posición de memoria física por una o una variedad de tecnologías de traducción de direcciones dinámicas (DAT) que incluyen, pero no se limitan a, simplemente prefijar una dirección virtual con un valor de desplazamiento, traducir la dirección virtual a través de una o más tablas de traducción, comprendiendo las tablas de traducción preferiblemente al menos una tabla de segmento y una tabla de página sola o en combinación, preferiblemente, teniendo la tabla de segmento una entrada que apunta a la tabla de página. En *z/Architecture*®, se promociona una jerarquía de traducción que incluye una tabla de primera región, una tabla de segunda región, una tabla de tercera región, una tabla de segmento y una tabla de página opcional. El rendimiento de la traducción de dirección es a menudo mejorado utilizando un búfer de traducción anticipada (TLB) que comprende entradas que mapean una dirección virtual para una localización de memoria física asociada. Las entradas son creadas cuando la DAT traduce una dirección virtual utilizando las tablas de traducción. El uso posterior de la dirección virtual puede entonces utilizar la entrada del TLB rápida en lugar de los accesos de tabla de traducción secuencial lentos. El contenido de TLB puede ser gestionado por una variedad de algoritmos de sustitución que incluyen LRU (Utilizado Menos Recientemente).

En el caso en el que el procesador sea un procesador de un sistema de múltiples procesadores, cada procesador tiene la responsabilidad de mantener los recursos compartidos, tales como I/O, cachés, TLBs y memoria, interbloqueados por coherencia. Típicamente, las tecnologías “de vigilancia” serán utilizadas en el mantenimiento de la coherencia de caché. En un entorno de vigilancia, cada línea de caché puede ser marcada como estando el cualquier de un estado compartido, un estado exclusivo, un estado cambiado, un estado inválido y similar con el fin de facilitar que se pueda compartir.

Las unidades I/O 5054 (Fig. 11) proporcionan al procesador medios para conectarse a dispositivos periféricos que incluyen cintas, discos, impresoras, pantallas y redes, por ejemplo. Las unidades I/O son a menudo presentadas al programa de ordenador mediante controladores (“drivers”) de software. En los ordenadores centrales de grandes prestaciones, tales como System *z*® de IBM, los adaptadores de canal y adaptadores de sistema abiertos son unidades I/O del ordenador central de grandes prestaciones que proporcionan las comunicaciones entre el sistema operativo y los dispositivos periféricos.

Además, otros tipos de entorno de ordenador se pueden beneficiar de uno o más aspectos de la presente invención. Como ejemplo, un entorno puede incluir un emulador (por ejemplo, un software u otros mecanismos de emulación), en el que una arquitectura particular (que incluye, por ejemplo, ejecución de instrucción, funciones diseñadas, tales como traducción de dirección, y registros diseñados) o un subconjunto de los mismos es emulada (por ejemplo en un sistema de ordenador nativo que tiene un procesador y una memoria). En tal entorno, una o más funciones de emulación del emulador pueden implementar uno o más aspectos de la presente invención, incluso aunque un ordenador que ejecute un emulador pueda tener diferente arquitectura que las capacidades que están siendo emuladas. Como ejemplo, en el modo de emulación, la instrucción u operación específicas que están siendo emuladas son descodificadas, y una función de emulación apropiada es construida para implementar la instrucción u operación individual.

En un entorno de emulación, un ordenador principal incluye, por ejemplo una memoria para almacenar instrucciones y datos; una unidad de recogida de instrucción para recoger instrucciones en la memoria y para opcionalmente, proporcionar almacenamiento en búfer local para la instrucción recogida; una unidad de decodificación de

instrucción para recibir las instrucciones recogidas y para determinar el tipo de instrucciones que han sido recogidas; y una unidad de ejecución de instrucción para ejecutar las instrucciones. La ejecución puede incluir cargar datos en un registro desde la memoria; almacenar datos de nuevo en la memoria desde un registro; o realizar algún tipo de operación aritmética o lógica; cuando se determina por la unidad de decodificación. En un ejemplo, cada unidad está implementada en software. Por ejemplo, las operaciones que están siendo realizadas por las unidades son implementadas como una o más subrutinas dentro del software emulador.

Más concretamente, en un ordenador central de grandes prestaciones, las instrucciones de máquina diseñadas son utilizadas programas, normalmente hoy programas "C", a menudo por medio de una aplicación compiladora. Estas instrucciones almacenadas en el medio de almacenamiento se pueden ejecutar nativamente en un Servidor z/Architecture® IBM, o alternativamente en máquinas que ejecutan otras arquitecturas. Pueden ser emuladas en servidores existente y en futuros servidores de ordenador central de grandes prestaciones IBM® y en otras máquinas de IBM® (por ejemplo servidores de Sistemas de Potencia y Servidores System x®). Pueden ser ejecutadas en máquinas que funcionan con Linux o una amplia variedad de máquinas que utilizan hardware fabricado por IBM®, Intel®, AMD™, y otros. Además de la ejecución en ese hardware bajo z/Architecture®, se puede utilizar Linux asó como máquinas que utilicen emulación por Hercules o FSI (Fundamental Software, Inc.), en 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 a arquitectura de un procesador emulador. Información acerca de los productos emuladores anteriormente mencionados está disponible respectivamente en las páginas web www.hercules-390.org y www.funsoft.com.

El procesador nativo típicamente ejecuta el software de emulación que comprende o bien firmware o bien un sistema operativo nativo para la recogida y ejecución de instrucciones de la arquitectura de procesador emulado. El software de emulación mantiene un contador de programa emulador para mantener la trayectoria de las delimitaciones de instrucción. Este software de emulación puede recoger una o más instrucciones de máquina emulada de una vez y convertir la una o más instrucciones de máquina emulada a un grupo correspondiente de instrucciones de máquina nativa para la ejecución por el procesador nativo. Estas instrucciones convertidas pueden ser almacenadas temporalmente de manera que se puede realizar una conversión más rápida. No obstante, el software de emulación es para mantener las reglas de arquitectura de la arquitectura del procesador emulado de manera que se garantice que los sistemas operativos y las aplicaciones escritas para el procesador emulado operan correctamente. Además, el software de emulación es para proporcionar recursos identificados por la arquitectura de procesador emulada, pero no se limita a, registraros de control, registros de finalidad general, registros de punto flotante, función de traducción de dirección dinámica que incluyen tablas de segmentos y tablas de página por ejemplo, mecanismo de interrupción, mecanismo de conmutación de contexto, relojes de Hora del Día (TOD) e interfaces diseñadas para subsistemas I/O de manera que un sistema operativo o un programa de aplicación diseñado para ejecutarse en el procesador emulador, se puede ejecutar en el procesador nativo que tiene 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 emulador está implementada, por ejemplo en una subrutina o controlador "C", o algún otro método de proporciona un controlador para el hardware específico como entenderán los expertos en la técnica a partir de la descripción de la realización preferida. distintas patentes de emulación de software y hardware, que incluye, pero no se limitan a el documento de Patente de Estados Unidos Nº 5.551.013, titulado "Microprocessor for Hardware Emulation" concedida a Beausleil et al.; y el Documento de Patente de Estados Unidos Nº 6.009.261, titulado "Processing of Stored Target Routines for Emulating Incompatible Instructions on a Target Processor", concedida a Scalzi et al.; y el Documento de Patente de Estados Unidos Nº 5.574.873 titulado "Decoding Guest Instruction to Directly Access Emulation Routines that Emulate the Guest Instructions" concedida a Davidian et al.; y el Documento de Patente de Estados Unidos Nº 6.308.255, titulado "Symmetrical Microprocessing Bus and Chipset Used for Coprocessor Support Allowing Non-Native Code to Run in a System", concedida a Gorishek et al; y el Documento de Patente Nº 6.463.582, titulado "Dynamic Optimizing Code Translator for Architecture Emulation Dynamic Optimizing Object Code Translation Method" concedida a Lethin et al; el Documento de Patente Nº 5.790.825, titulado "Method for Emulating Guest Instructions on a Host Computer Through Dynamic Recompilation of Host Instructions", concedido a Eric Traut; y muchos otros, ilustran una variedad de máquinas diferentes para una máquina objetivo disponible para los expertos en la técnica.

En la Fig. 13, se proporciona un ejemplo de un sistema de ordenador principal emulador 5092 que emula un sistema de ordenador principal 5000' de una arquitectura principal. En el sistema de ordenador principal emulado 5092, el procesador principal (CPU) 5091 es un procesador principal emulado (o procesador principal virtual) y comprende un procesador de emulación 5093 que tiene una arquitectura de conjunto de instrucción nativa diferente que la del procesador 5091 del ordenador principal 5000'. El sistema de ordenador principal emulado 5092 tiene la memoria 5094 accesible al procesador de emulación 5093. En la realización ejemplo, la memoria 5094 está situada en una parte de memoria de ordenador principal 5096 y una parte de rutinas de emulación 5097. La memoria de ordenador principal 5096 está disponible para programas del ordenador principal emulado 5092 de acuerdo con la arquitectura de ordenador principal. El procesador de emulación 5093 ejecuta las instrucciones nativas de un conjunto de instrucciones diseñadas de una arquitectura distinta de la del procesador emulado 5091, las instrucciones nativas obtenidas de la memoria de rutinas de emulación 5097, y puede acceder a una instrucción principal para la

5 ejecución desde un programa en la memoria de ordenador principal 5096 empleando una o más instrucciones (s) obtenidas en una rutina de secundaria y acceso/decodificación que puede decodificar las instrucciones principales a las que se ha accedido para determinar una rutina de ejecución de instrucción nativa para emular la función de la instrucción principal a la que se ha accedido. Otras prestaciones que están definidas por la arquitectura del sistema de ordenador principal 5000' se pueden emular mediante las rutinas de prestaciones diseñadas incluyendo tales prestaciones como registro de finalidad general, registros de control, traducción de dirección dinámica y soporte de subsistemas I/O y caché de procesador, por ejemplo. Las rutinas de emulación pueden también sacar partido de las funciones disponibles en el procesador de emulación 5093 (tal como los registros generales y traducciones dinámicas de direcciones virtuales) para mejorar el rendimiento de las rutinas de emulación. Se pueden proporcionar hardware especial y "offload engines" para ayudar al procesador 5093 en la emulación de la función del ordenador principal 5000'.

10

REIVINDICACIONES

1. Un método para habilitar adaptadores en un entorno de ordenadores, que comprende las etapas de:

5 responder para ejecutar una instrucción de Procesador Lógico de Llamada CLP (400) expedida por un sistema operativo para habilitar un adaptador seleccionado por el sistema operativo, comprendiendo la instrucción CLP un identificador de función (426) que identifica el adaptador y que tiene un indicador de adaptador no habilitado (352), solicitando la instrucción CLP un número de espacios de direcciones de acceso directo a memoria DMA (430) para ser asignados al adaptador, habilitando la ejecución uno o más espacios de dirección DMA y que comprende:

 - 10 a) habilitar el adaptador, en donde la habilitación comprende habilitar el registro para la traducción e interrupción de dirección para el soporte de accesos de memoria directos e interrupciones señalizadas con mensaje para el adaptador, y en donde la habilitación comprende determinar (512) que el número de de espacios de dirección DMA solicitados está disponible comprobando que hay entradas de tabla de dispositivo para el número de espacios de dirección DMA solicitado, y asignar (542) un número de entradas de tabla de dispositivo, correspondiente al número de espacios de dirección de DMA solicitado, al adaptador; y
 - 15 b) devolver (550) el identificador de función que tiene un indicador de adaptador habilitado.
2. El método de la reivindicación 1, en el que la habilitación comprende habilitar una o más instrucciones para ser expedidas al adaptador.
3. El método de la reivindicación 1, en el que el número de espacios de dirección para ser asignados está indicado en un bloque de solicitud de la instrucción CLP.
- 20 4. El método de la reivindicación 1, en el que la habilitación comprende además:

utilizar el identificador de función del adaptador para localizar una entrada de tabla de función asociada con el adaptador; y

25 emplear información en la entrada de tabla de función para determinar si el adaptador va a ser habilitado, y en la que la habilitación asigna una o mas entradas de tabla de dispositivo al adaptador como respuesta a la determinación de que el adaptador va a ser habilitado.
5. El método de la reivindicación 4, en el que el identificador de función está asociado con la entrada de tabla de función e incluye un número de función y un número de caso, y en donde el método comprende además determinar la validez del identificador, comprendiendo la determinación:

30 comprobar que el indicador no habilitado indica no habilitado; y

comprobar que el número de función designa una función instalada, en la que se realiza la utilización como respuesta a la determinación de un identificador válido.
6. El método de la reivindicación 4, en el que la etapa de emplear comprende comprobar al menos uno de, el indicador no habilitado, un indicador de estado permanente, un indicador de recuperación iniciada, un indicador de ocupado o un indicador de permiso en la entrada de tabla de función para determinar si el adaptador va a ser habilitado.
7. El método de la reivindicación 1, en el que la habilitación comprende además asociar una o más entradas de tabla de dispositivo con una entrada de tabla de función asociada con el adaptador, proporcionando la entrada de tabla de función información referente al adaptador.
- 40 8. El método de la reivindicación 7, en el que la entrada de tabla de función está asociada con el identificador de función, y en el que la habilitación comprende además realizar al menos una de las siguientes acciones:

establecer un indicador de habilitación de función en la entrada de tabla de función para indicar habilitado;

establecer uno o más indicadores de habilitación de dispositivo en una o más entradas de tabla de dispositivo para indicar habilitado;

45 incluir en una memoria direccionable de contenidos uno o más índices para una o más entradas de tabla de dispositivo, siendo la memoria direccionable de contenidos utilizable para localizar una entrada de tabla de dispositivo como respuesta a una solicitud procedente del adaptador;

establecer el indicador de adaptador habilitado en el identificador de función para indicar habilitado; y actualizar un número de caso del identificador de función.

9. El método de la reivindicación 1, en el que el método comprende además la etapa de deshabilitar el adaptador.
10. El método de la reivindicación 9, en el que la deshabilitación comprende:
- 5 utilizar el identificador de función para localizar una entrada de tabla de función asociada con el adaptador; y
- utilizar información en la entrada de tabla de función para determinar si el adaptador va a ser deshabilitado, y continuar con la deshabilitación en respuesta a la determinación de que el adaptador va a ser deshabilitado.
11. El método de la reivindicación 10, en el que la etapa de proceder con la deshabilitación incluye al menos una de las siguientes acciones:
- 10 establecer un indicador de habilitación de función en la entrada de tabla de función en deshabilitado; borrar y liberar una o más entradas de tabla de dispositivo asociadas con el adaptador; y establecer el indicador no habilitado del identificador de función para indicar deshabilitado.
12. El método de la reivindicación 10, en el que la deshabilitación comprende además determinar la validez del identificador de función, comprendiendo la determinación:
- 15 comprobar que el indicador de adaptador habilitado está establecido en habilitado; y comprobar que el identificador apunta a una entrada válida en la tabla de función, en donde la utilización se realiza como respuesta a la determinación de que el identificador de función es válido.
13. El método de la reivindicación 12, en el que la etapa de emplear comprende comparar un número de caso en el identificador de función con un número de caso en la entrada de tabla de función, en donde la acción de proceder con la deshabilitación se produce como respuesta a la comparación que indica igualdad.
14. El método de la reivindicación 1, en el que el adaptador comprende una función de Interconexión de Componente Periférico (PCI).
15. Un programa de ordenador que comprende un código de programa de ordenador almacenado en un medio leíble por ordenador para, cuando está cargado en un sistema de ordenador y se ejecuta en el mismo, hacer que dicho sistema de ordenador realice todas las etapas de un método de acuerdo con cualquiera de las reivindicaciones 1 a 14.
16. Un sistema de ordenador para habilitar adaptadores en un entorno de ordenadores, comprendiendo dicho sistema:
- 30 una memoria (104)
- un procesador (102) en comunicación con la memoria;
- un elemento solicitante como respuesta a la ejecución de una instrucción de Procesador Lógico de Llamada CLP expedida por un sistema operativo para habilitar un adaptador (110) seleccionado por el sistema operativo, comprendiendo la instrucción CLP un identificador de función que identifica el adaptador y que
- 35 tiene un indicador de adaptador no habilitado, solicitando la instrucción CLP un número de espacios de dirección de acceso directo a memoria DMA para ser asignado al adaptador, habilitando la ejecución uno o más espacios de dirección DMA, y que comprende:
- a) un habilitador para habilitar el adaptador, en donde el habilitador comprende registro de habilitación para traducciones e interrupciones de dirección para soportar accesos a memoria directos e interrupciones señalizadas con mensaje para el adaptador, y en donde la habilitación comprende
- 40 determinar que el número solicitado de espacios de dirección DMA está disponible comprobando que hay entradas de tabla de dispositivo para el número solicitado de espacios de dirección DMA, y asignar un número de entradas de tabla de dispositivo, correspondiente al número solicitado de espacios de dirección DMA, al adaptador; y
- b) un componente de devolución para devolver el identificador de función que tiene un indicador de adaptador habilitado.
17. El sistema de ordenador de la reivindicación 16, en el que la habilitación comprende habilitar una o más instrucciones para ser expedidas al adaptador.

18. El sistema de ordenador de la reivindicación 16, en el que la habilitación comprende además:
- utilizar el identificador de función del adaptador para localizar una entrada de tabla de función asociada con el adaptador; y
- 5 emplear información en la entrada de tabla de función para determinar si el adaptador va a ser habilitado, y en donde la habilitación asigna una o más entradas de tabla de dispositivo al adaptador como respuesta a la determinación que el adaptador va a ser habilitado.
19. El sistema de ordenador de la reivindicación 16, en el que la habilitación comprende además asociar una o más entradas de tabla de dispositivo con una entrada de tabla de función asociada con el adaptador, proporcionado la entrada de tabla de función información referente al adaptador.
- 10 20. El sistema de ordenador de la reivindicación 19, en el que la entrada de tabla de función está asociada con el identificador de función, y en el que la habilitación comprende además realizar al menos una de las dos acciones siguientes:
- establecer un indicador de habilitación de función en la entrada de tabla de función para indicar habilitado;
- 15 establecer uno o más indicadores de habilitación de dispositivo en una o más entradas de tabla de dispositivo para indicar habilitado;
- incluir en una memoria direccionable de contenidos uno o más índices para una o más entradas de tabla de dispositivo, siendo la memoria direccionable de contenidos utilizable para localizar una entrada de tabla de dispositivo como respuesta a una solicitud procedente del adaptador;
- establecer el indicador de adaptador habilitado en el identificador de función para indicar habilitado; y
- 20 actualizar un número de caso del identificador de función.
21. El sistema de ordenador de la reivindicación 16, en el que el sistema comprende además un deshabilitador para deshabilitar el adaptador, y en el que la deshabilitación comprende:
- utilizar el identificador de función para localizar una entrada de tabla de función asociada con el adaptador; y
- 25 emplear información en la entrada de tabla de función para determinar si el adaptador va a ser deshabilitado, y proceder con la deshabilitación como respuesta a la determinación de que el adaptador va a ser deshabilitado.
22. El sistema de ordenador de la reivindicación 21, en el que la deshabilitación comprende al menos una de las siguientes acciones:
- 30 establecer un indicador de habilitación de función en una entrada de tabla de función en deshabilitado;
- borrar y liberar una o más entradas de tabla de dispositivo asociadas con el adaptador; y
- establecer el indicador no habilitado del identificador de función para indicar deshabilitado.

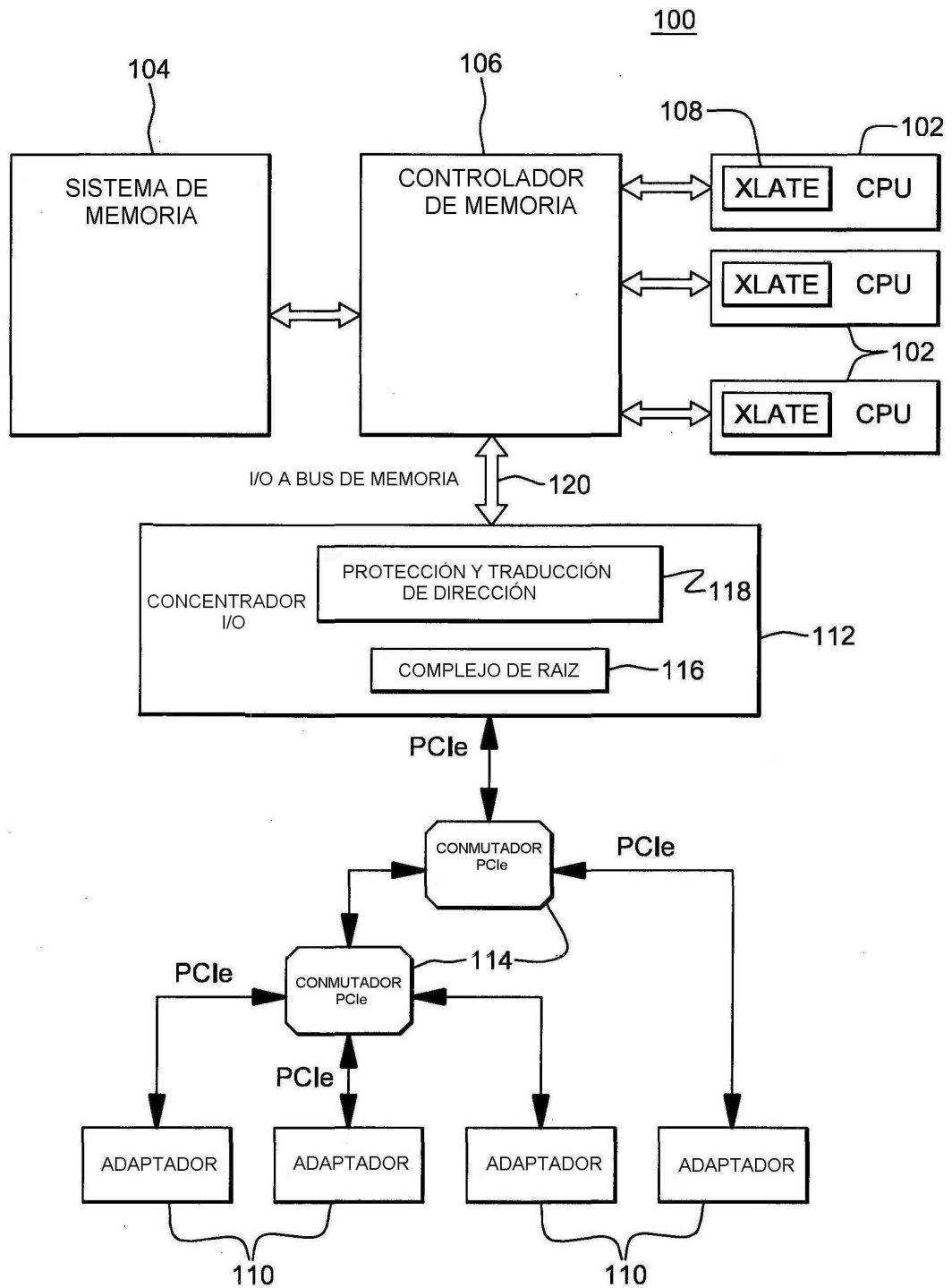


FIG. 1A

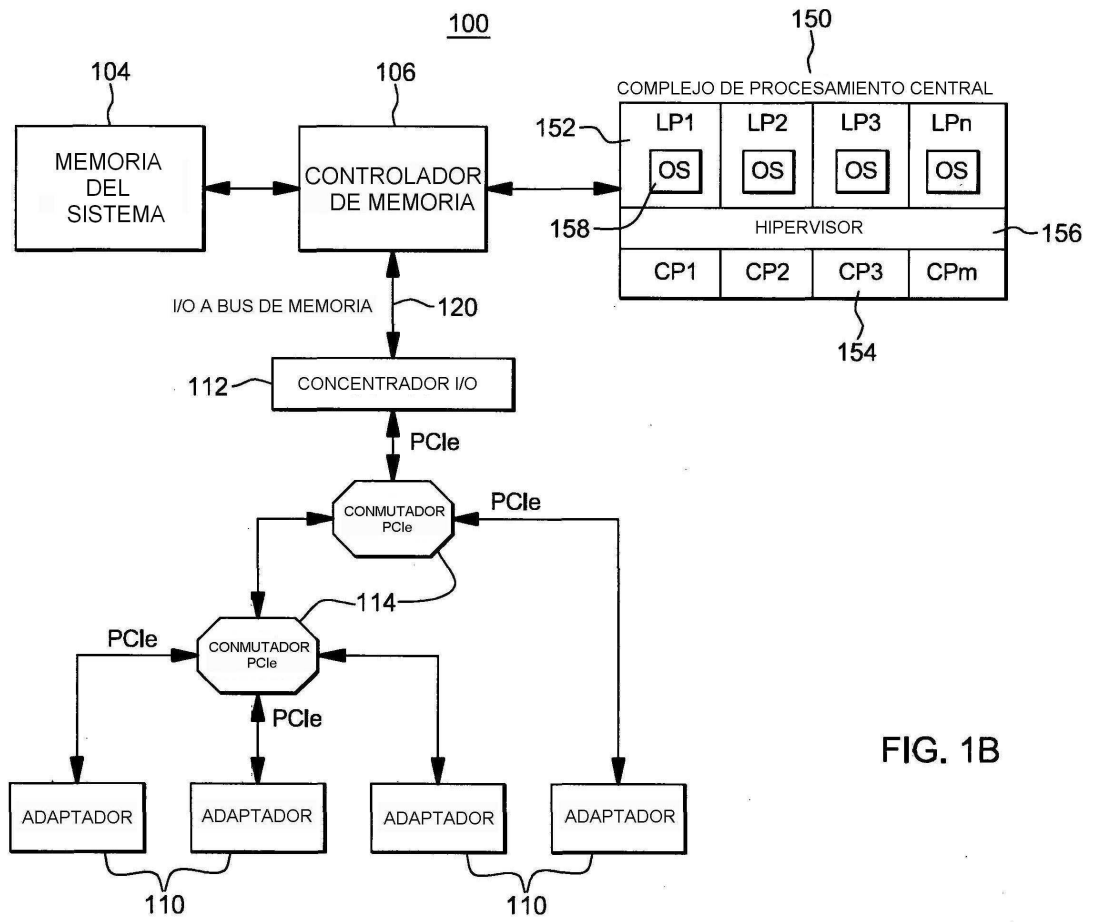


FIG. 1B

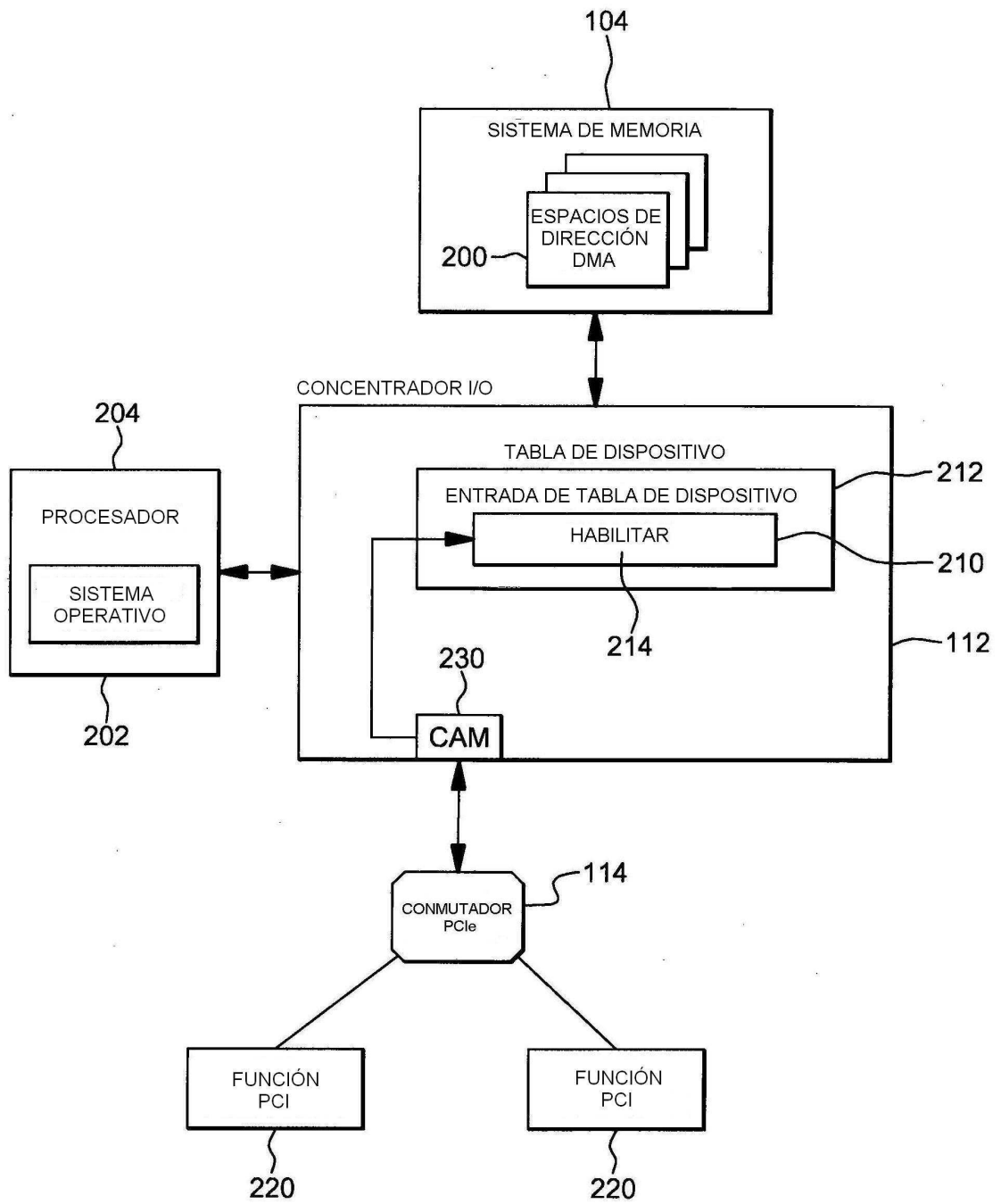


FIG. 2

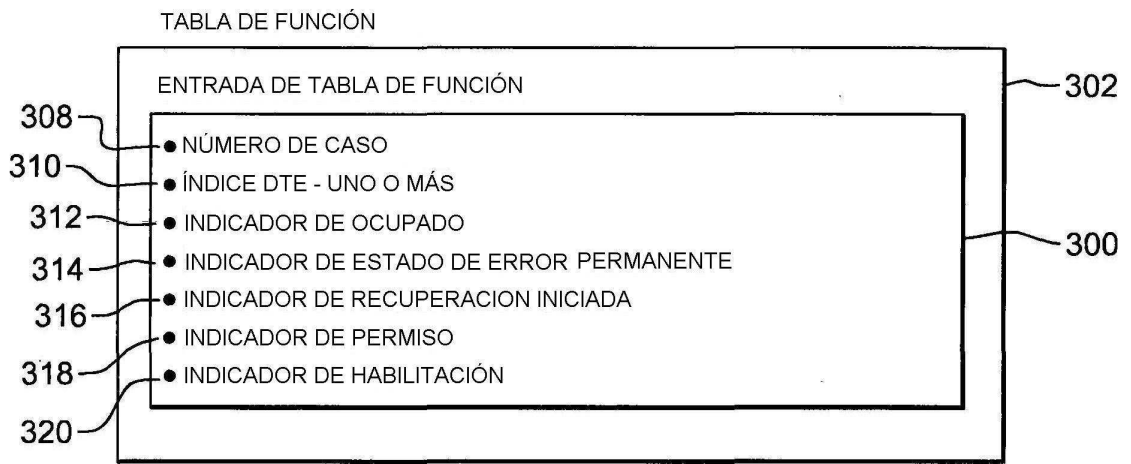


FIG. 3A

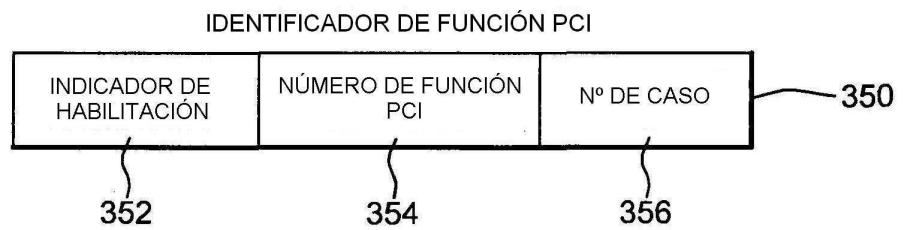


FIG. 3B

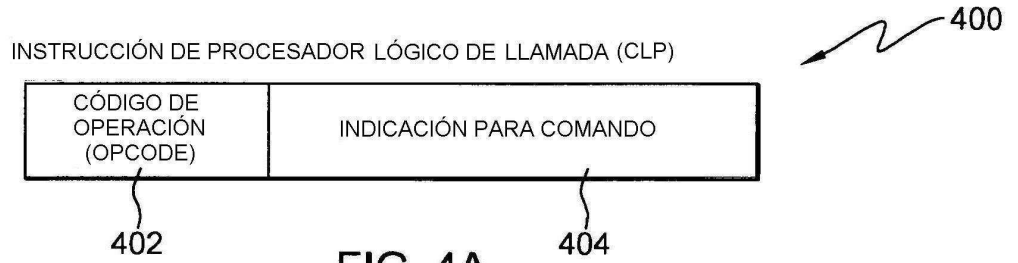


FIG. 4A

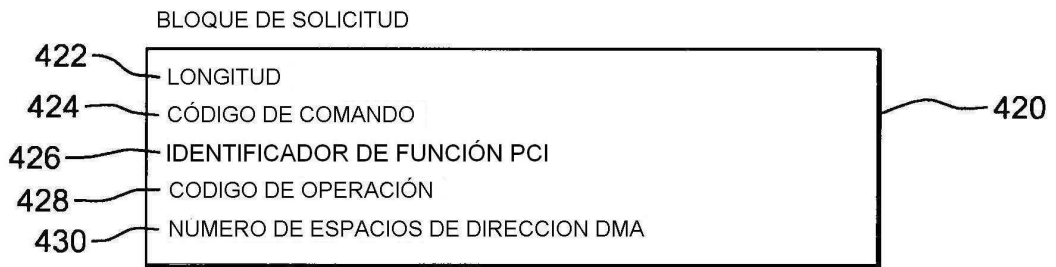


FIG. 4B

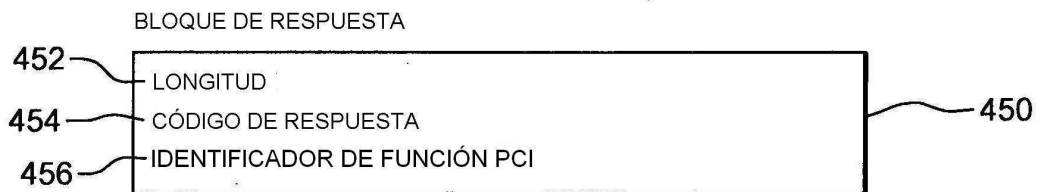


FIG. 4C

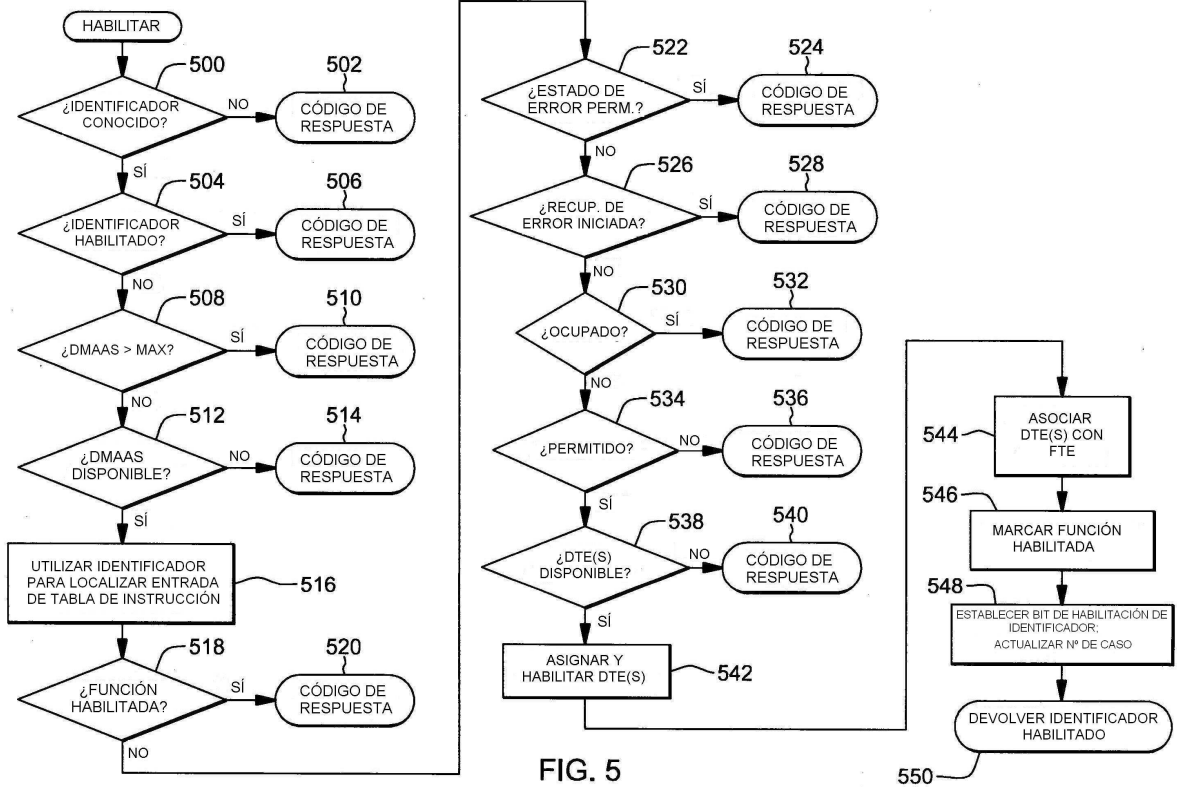


FIG. 5

550

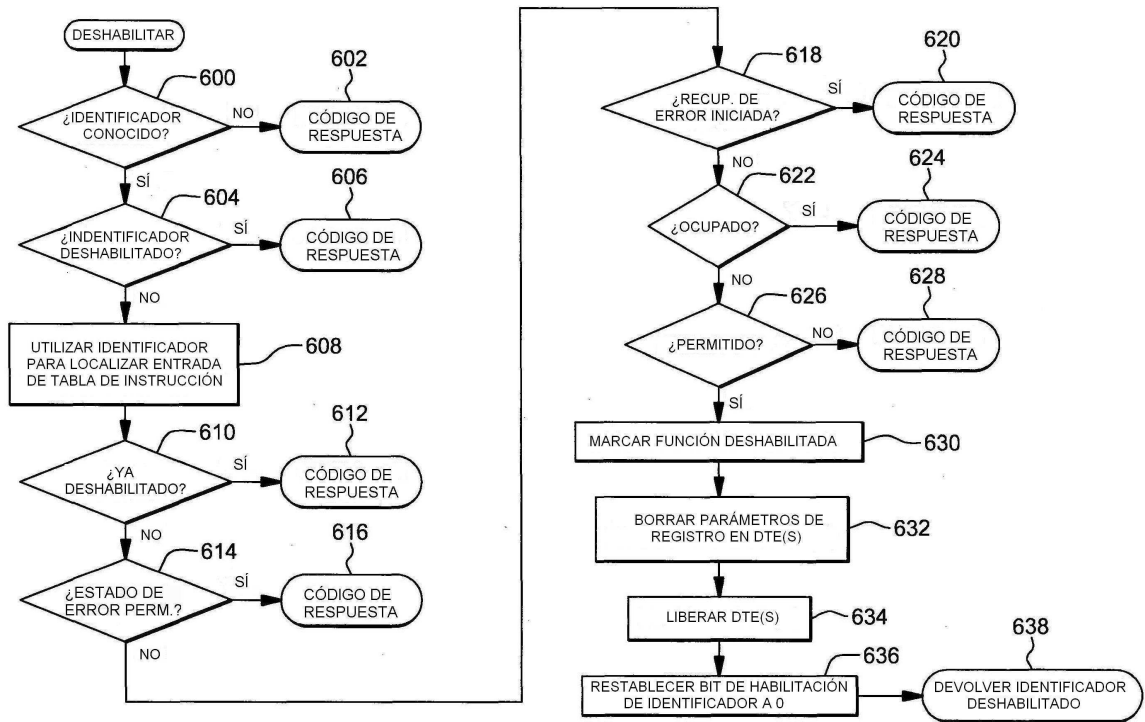


FIG. 6

PRODUCTO DE PROGRAMA
DE ORDENADOR

700

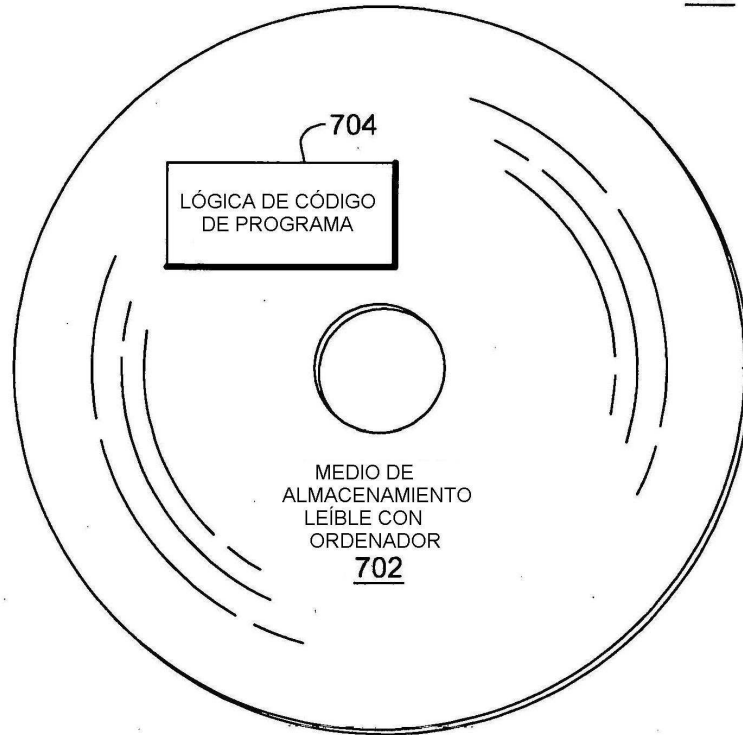


FIG. 7

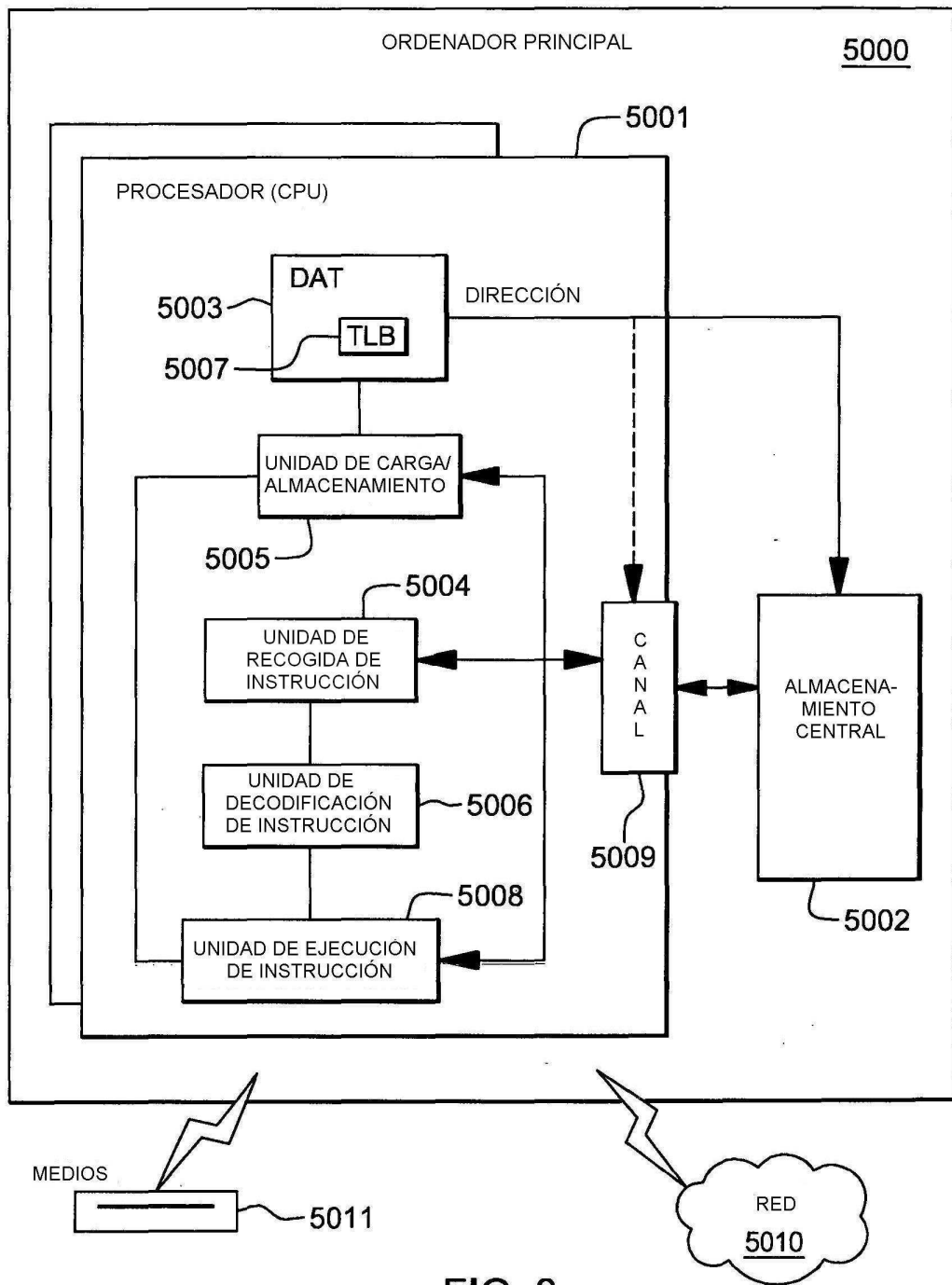


FIG. 8

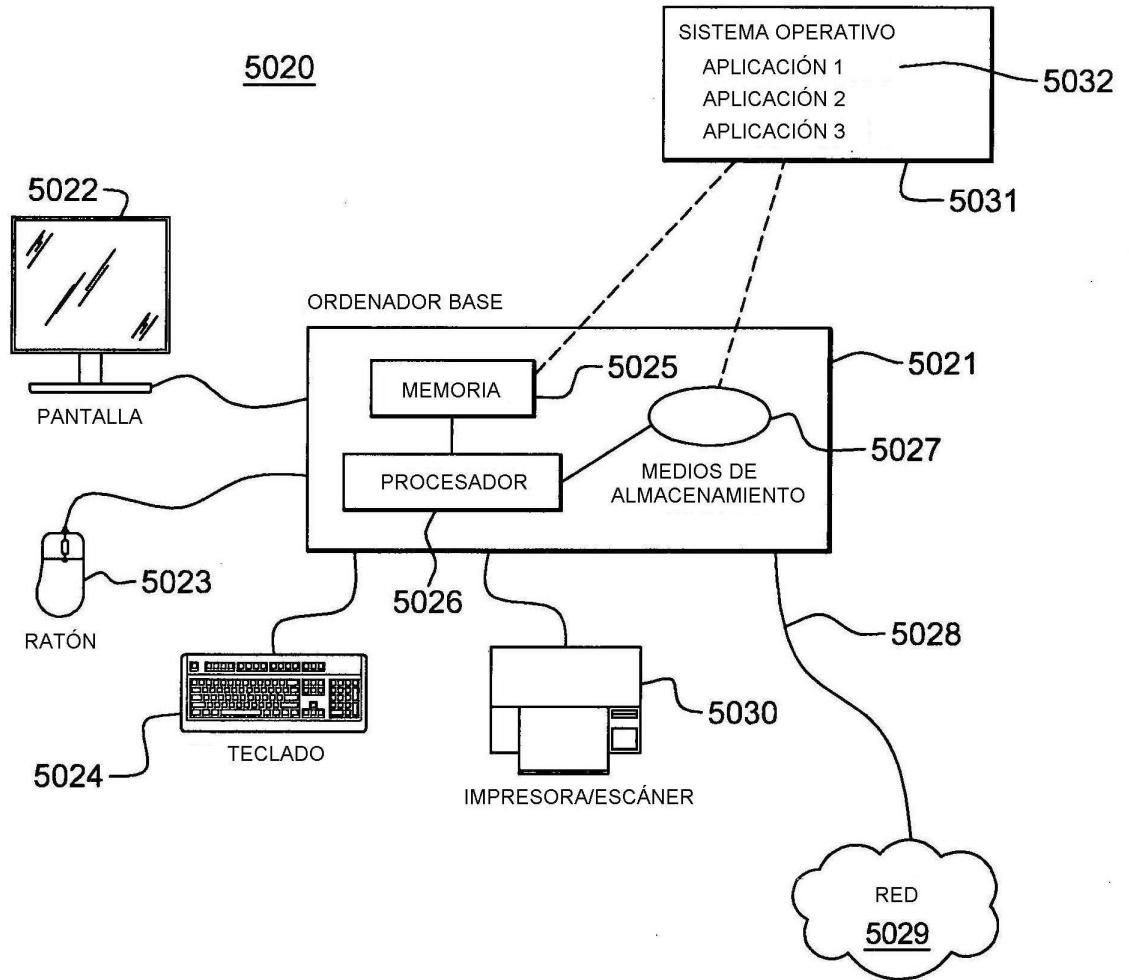


FIG. 9

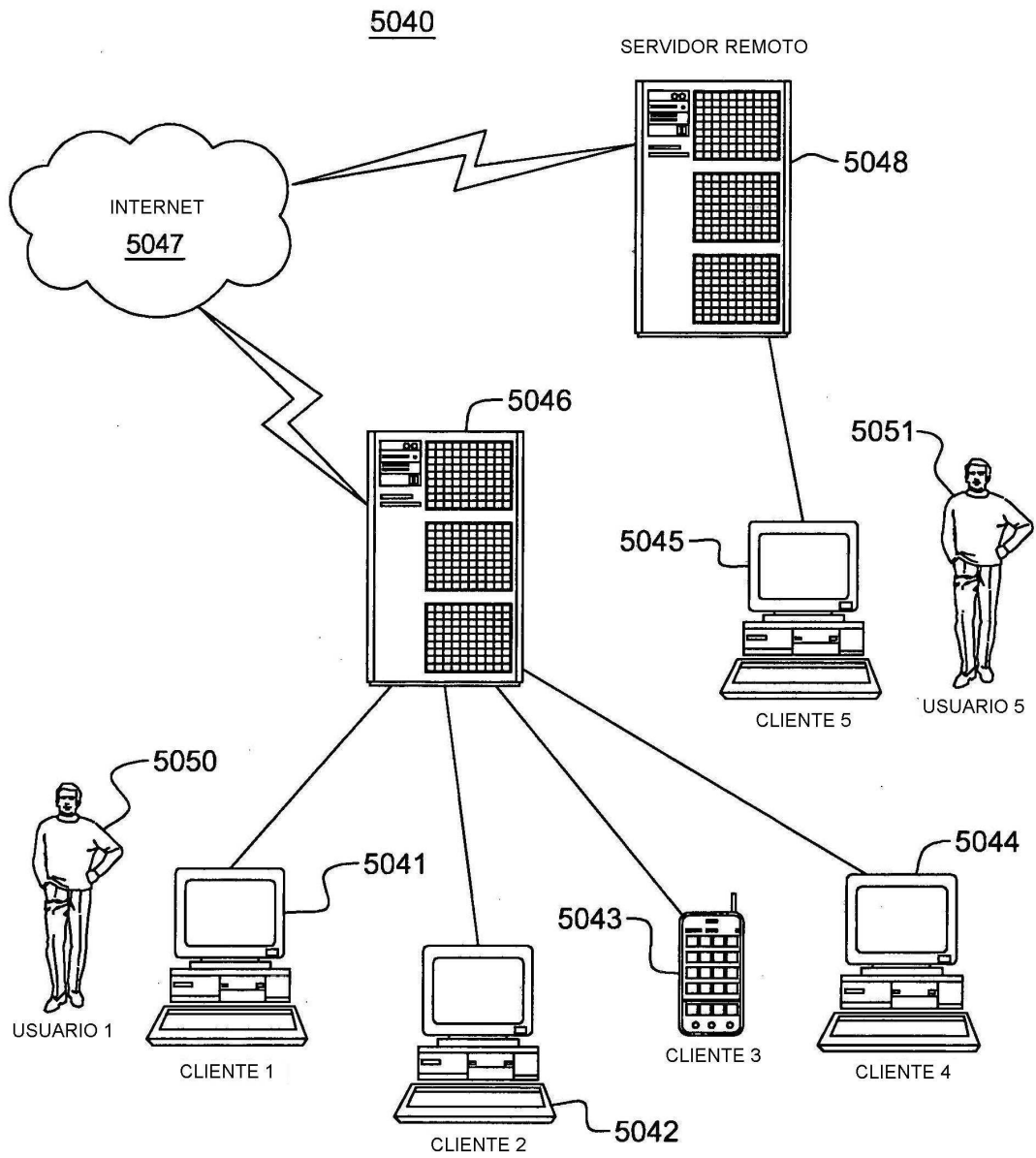


FIG. 10

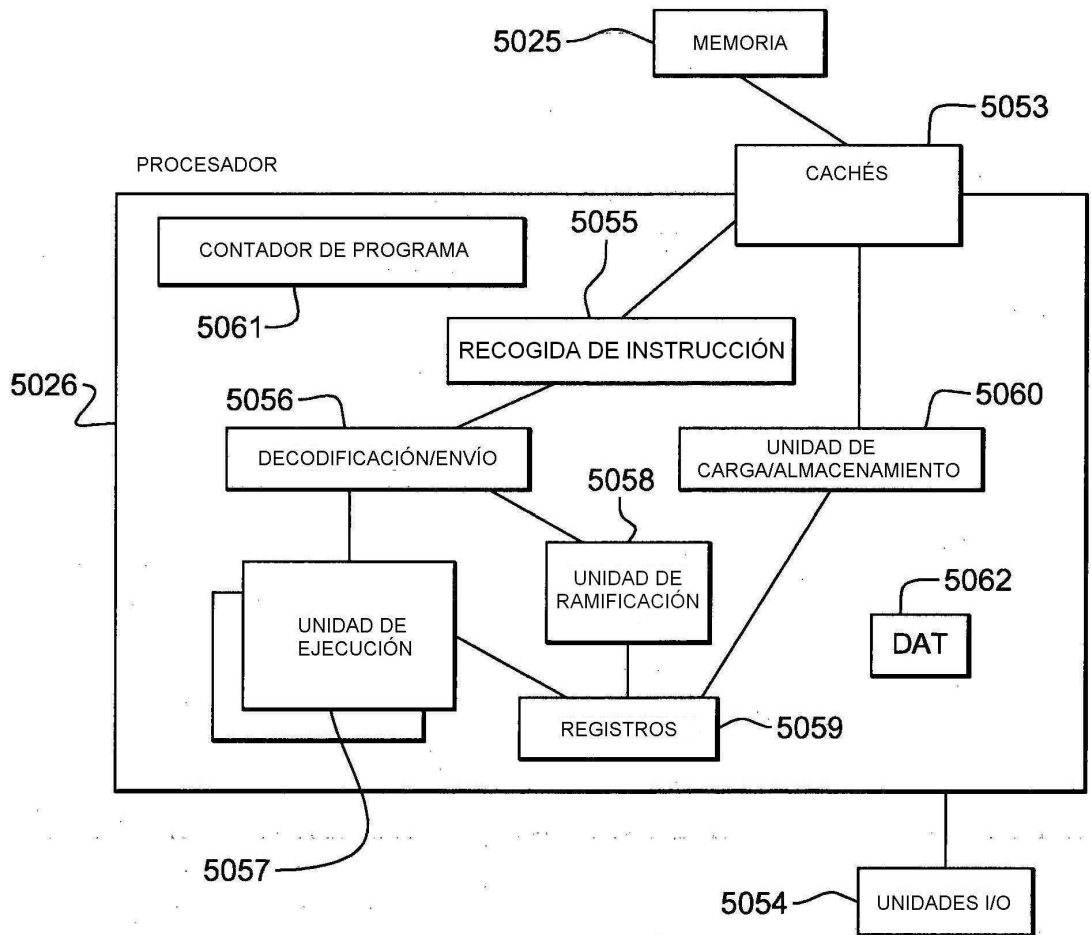


FIG. 11

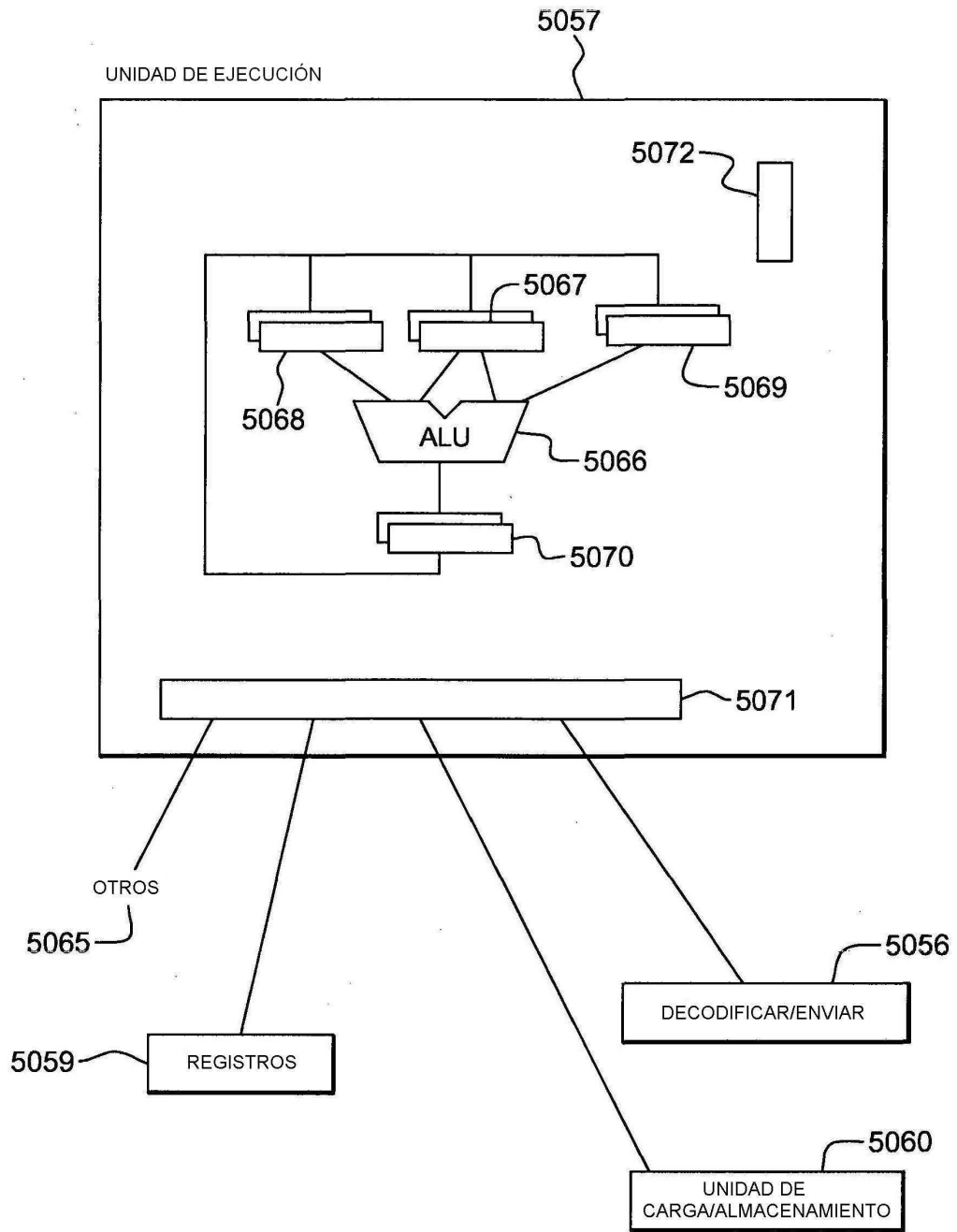


FIG. 12A

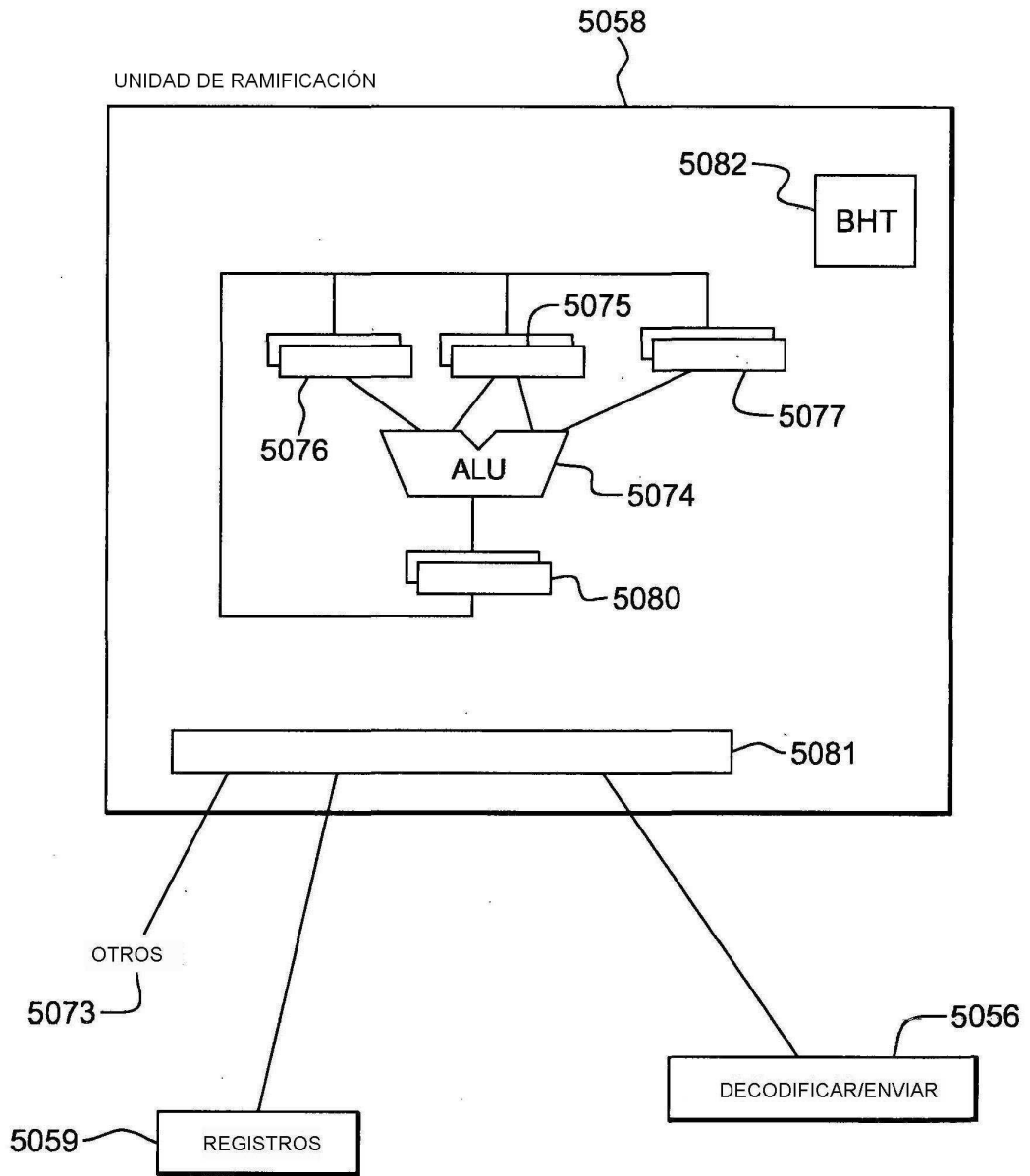


FIG. 12B

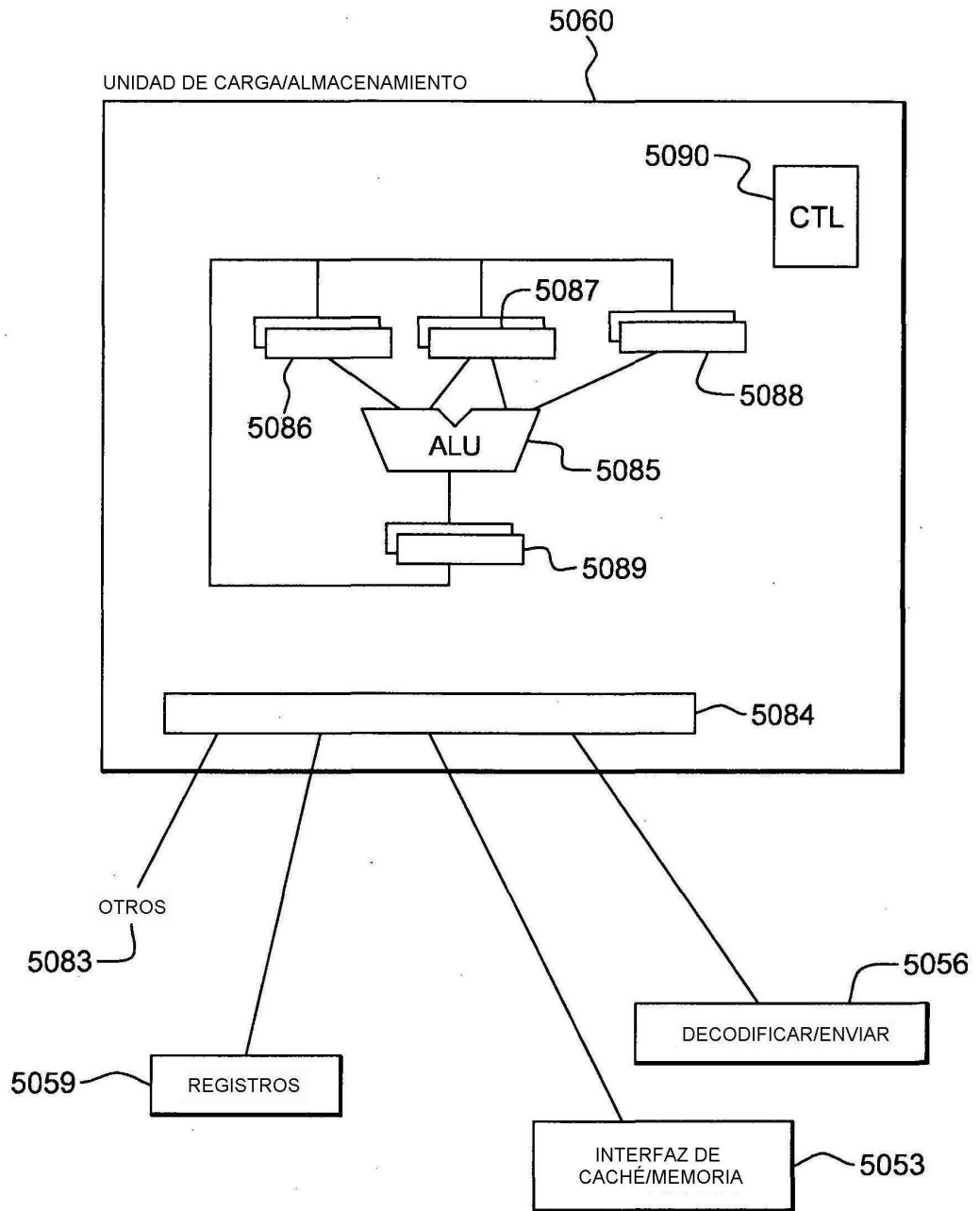


FIG. 12C

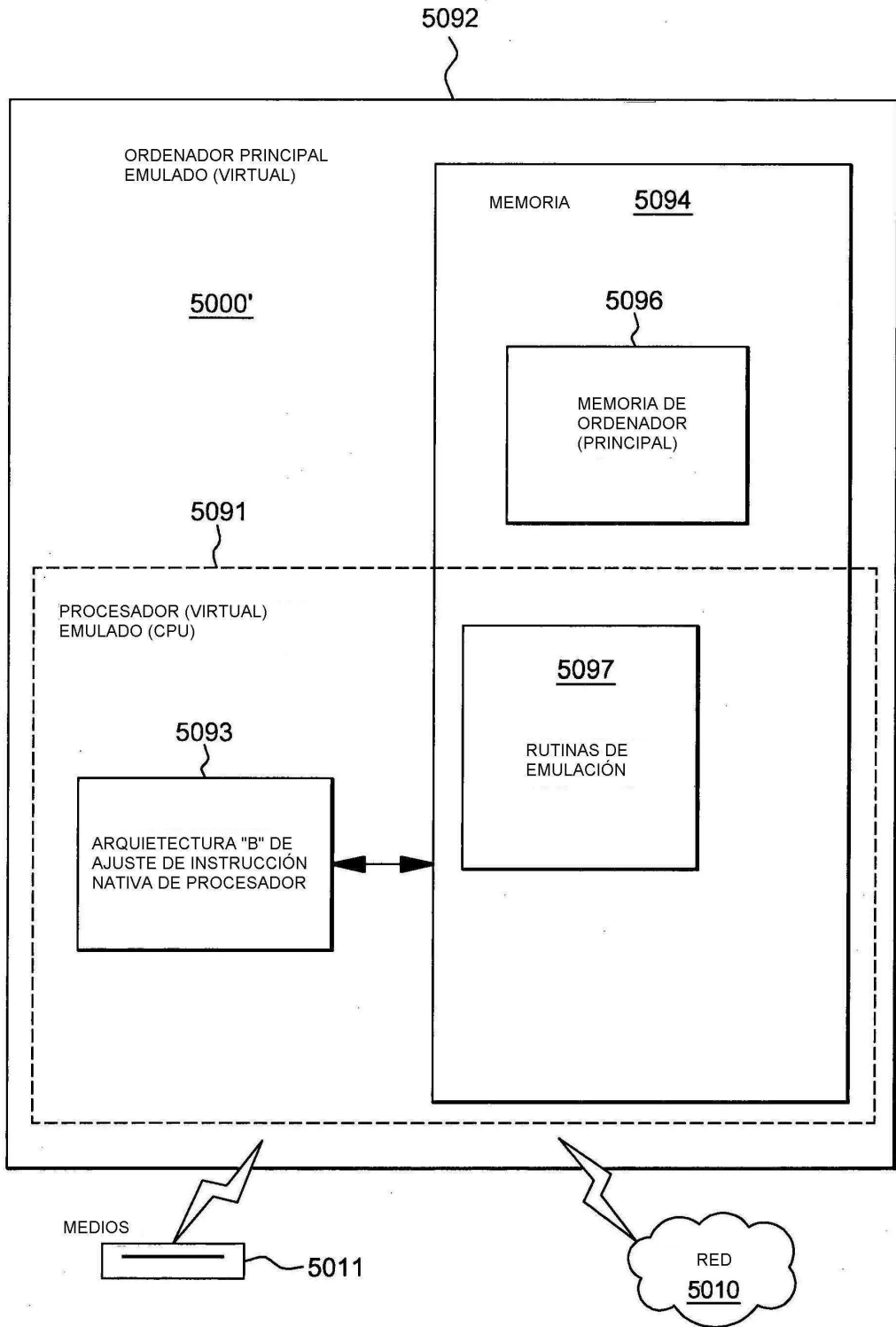


FIG. 13