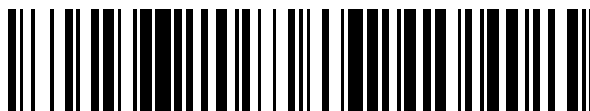


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 535 333**

51 Int. Cl.:

G06F 13/38 (2006.01)

G06F 13/24 (2006.01)

G06F 9/48 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **08.11.2010 E 10781635 (7)**

97 Fecha y número de publicación de la concesión europea: **25.03.2015 EP 2430558**

54 Título: **Convertir una interrupción señalada por mensaje en una notificación de evento de adaptador de I/O**

30 Prioridad:

23.06.2010 US 821175

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

08.05.2015

73 Titular/es:

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

72 Inventor/es:

**SITTMANN III, GUSTAV;;
CRADDOCK, DAVID;
GREGG, THOMAS;
FARRELL, MARK;
EASTON, JANET y
LAIS, ERIC NORMAN**

74 Agente/Representante:

DE ELZABURU MÁRQUEZ, Alberto

ES 2 535 333 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

Convertir una interrupción señalada por mensaje en una notificación de evento de adaptador de I/O

Antecedentes

5 La invención se refiere, en general, a procesamiento de interrupciones dentro de un entorno informático y, en particular, a manejar interrupciones generadas por adaptadores del entorno informático.

Una interrupción señalada por mensaje (MSI) es una manera para que una función de adaptador, tal como una función de Interconexión de Componente Periférico (PCI), genere una interrupción de la unidad central de proceso (CPU) para informar al sistema operativo de la aparición de un evento o la presencia de algún estado. MSI es una alternativa a tener un pin de interrupción dedicado en cada dispositivo. Cuando una función de adaptador se configura para usar una MSI, la función solicita una interrupción realizando una operación de escritura de MSI de un número especificado de bytes de datos en una dirección especial. La combinación de esta dirección especial y un valor de datos único se denomina un vector de MSI.

10 Algunas funciones de adaptador soportan solamente un vector de MSI; otras funciones de adaptador soportan múltiples vectores de MSI. Para funciones que soportan múltiples vectores de MSI, la misma dirección especial se usa con diferentes valores de datos.

En muchas plataformas informáticas, un controlador de dispositivo se configura a sí mismo como el gestor de interrupciones asociado con un vector de MSI. Este asocia eficazmente un vector de MSI con una entrada en un vector de interrupción de CPU. Por lo tanto, cuando una función de adaptador soporta múltiples vectores de MSI y está configurada para usar múltiples vectores de MSI, consume un número correspondiente de entradas en el vector de interrupción de CPU.

20 La Publicación de EE.UU. N° 2007/0271559 A1, publicada el 22 de noviembre de 2007, Easton et al., "Virtualization of Infiniband Host Channel Adapter Interruptions," describe un método, sistema, producto de programa y estructura de datos de ordenador para proporcionar dos niveles de virtualización de servidor. Un primer hipervisor habilita múltiples particiones lógicas para compartir un conjunto de recursos y proporciona un primer nivel de virtualización. Un segundo hipervisor habilita múltiples máquinas virtuales independientes para compartir los recursos que se asignan a una partición lógica única y proporciona un segundo nivel de virtualización. Todos los eventos para todas las máquinas virtuales dentro de dicha partición lógica única se agrupan en una cola única de evento propiedad de la partición para recibir notificaciones de evento desde los recursos compartidos para esa partición lógica única. Una petición de una interrupción se señala para los eventos agrupados desde la cola de evento propiedad de la partición para la demultiplexación de eventos agrupados, por la máquina, desde la cola de evento propiedad de la partición en colas de evento individuales, virtualizadas que se asignan sobre una base por máquina virtual.

25 La Publicación de EE.UU. N° 2005/0289271 A1, publicada el 29 de diciembre de 2005, Martínez et al., "Circuitry to Selectively Produce MSI Signals," describe, en algunas realizaciones, que las invenciones incluyen un circuito integrado que tiene un circuito de registro de estado acoplado a conductores para recibir señales de evento de interrupción para proporcionar señales fuente que corresponden a las señales de evento de interrupción. El circuito integrado también incluye un circuito de registro de control para proporcionar señales de habilitación fuente para las selectivas de las fuentes de interrupción y un circuito de lógica de rearme acoplado a los conductores para recibir las señales de evento de interrupción y proporcionar una señal de rearme. El circuito integrado además incluye un primer circuito lógico para recibir las señales fuente, las señales de habilitación fuente y la señal de rearme para proporcionar una señal de interrupción inicial y la lógica de generación de pulso de señal de interrupción señalada por mensaje (MSI) para recibir la señal de interrupción inicial y proporcionar una señal de MSI en respuesta a la misma. Otras realizaciones se describen y reivindican.

35 La Patente de EE.UU. N° 7.562.366, expedida el 14 de julio de 2009, Pope et al., "Transmit Completion Event Batching," describe un método para gestionar una cola de transmisión de datos, para uso con un ordenador principal y un dispositivo de interfaz de red. Descrita a grandes rasgos, el ordenador principal escribe descriptores de almacenador temporal de datos en una cola de descriptor de transmisión y el dispositivo de interfaz de red escribe eventos para notificar al ordenador principal cuándo ha completado el procesamiento de un almacenador temporal de datos de transmisión. Cada uno de los descriptores de evento de terminación de transmisión informa al ordenador principal de la terminación de una pluralidad de los almacenadores temporales de datos de transmisión.

40 "Chapter 6: Configuration Space" en: PCI-SIG: "PCI Local Bus Specification Revision 3.0", 3 de febrero de 2004, XP002638677, páginas 213-254, describe un dispositivo de PCI que tiene una memoria que almacena un espacio de configuración.

45 La US 2006/195617A1 (Arndt Richard L [US] et al), 31 de agosto de 2006 describe el número de Función de Canal Principal-Dispositivo (BDF) bien conocido.

55

Breve compendio

Según un aspecto de la presente invención, se proporciona una capacidad para facilitar la gestión de peticiones de interrupción desde adaptadores.

5 Las deficiencias de la técnica anterior se superan y se proporcionan ventajas a través del suministro de un método según la reivindicación 1 y el sistema correspondiente y producto de programa de ordenador para gestionar peticiones de interrupción en un entorno informático.

Breve descripción de las diversas vistas de los dibujos

10 Uno o más aspectos de la presente invención se indican particularmente y reivindican distintivamente como ejemplos en las reivindicaciones en la conclusión de la especificación. Los precedentes y otros objetos, rasgos y ventajas de la invención serán evidentes a partir de la siguiente descripción detallada tomada en conjunto con los dibujos anexos en los cuales:

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

La FIG. 2 representa una realización de detalles adicionales de una memoria del sistema y el centro de I/O de la FIG. 1, según un aspecto de la presente invención;

15 Las FIG. 3A-3B representan ejemplos de asignaciones de vectores de bits de interrupción de adaptador, según un aspecto de la presente invención;

Las FIG. 3C-3D representan ejemplos de asignaciones de bits de resumen de interrupción de adaptador, según un aspecto de la presente invención;

20 La FIG. 4 representa una realización de una vista general de la lógica a ser realizada durante la inicialización para configurar una función de adaptador para notificación de evento de adaptador de I/O, según un aspecto de la presente invención;

La FIG. 5 representa una realización de la lógica para realizar un registro para permitir la conversión de una interrupción señalada por mensaje (MSI) en una notificación de evento de adaptador de I/O, según un aspecto de la presente invención;

25 La FIG. 6A representa una realización de la lógica para convertir una petición de MSI a una notificación de evento de adaptador de I/O, según un aspecto de la presente invención;

La FIG. 6B representa una realización de la lógica para presentar la notificación de evento de adaptador de I/O a un sistema operativo, según un aspecto de la presente invención;

30 La FIG. 7A representa una realización de una instrucción Modificar los Controles de Función de PCI usada según un aspecto de la presente invención;

La FIG. 7B representa una realización de un campo usado por la instrucción Modificar los Controles de Función de PCI de la FIG. 7A, según un aspecto de la presente invención;

La FIG. 7C representa una realización de otro campo usado por la instrucción Modificar los Controles de Función de PCI de la FIG. 7A, según un aspecto de la presente invención;

35 La FIG. 7D representa una realización de los contenidos de un bloque de información de función (FIB) usado según un aspecto de la presente invención;

La FIG. 8 representa una realización de una vista general de la lógica de la instrucción Modificar los Controles de Función de PCI, según un aspecto de la presente invención;

40 La FIG. 9 representa una realización de la lógica asociada con una operación de interrupciones de adaptador de registro que se puede especificar por la instrucción Modificar los Controles de Función de PCI, según un aspecto de la presente invención;

La FIG. 10 representa una realización de la lógica asociada con una operación de interrupciones de adaptador de no registro que se puede especificar por la instrucción Modificar los Controles de Función de PCI, según un aspecto de la presente invención;

45 La FIG. 11A representa una realización de una instrucción Llamar al Procesador Lógico usada según un aspecto de la presente invención;

La FIG. 11B representa una realización de un bloque de petición usado por la instrucción Llamar al Procesador Lógico de la FIG. 11A para una operación de lista, según un aspecto de la presente invención;

La FIG. 11C representa una realización de un bloque de respuesta para la operación de lista de la FIG. 11B, según un aspecto de la presente invención;

La FIG. 11D representa una realización de una entrada de lista de función usada según un aspecto de la presente invención;

5 La FIG. 12A representa una realización de un bloque de petición usado por la instrucción Llamar al Procesador Lógico de la FIG. 11A para una operación de consultar la función, según un aspecto de la presente invención;

La FIG. 12B representa una realización de un bloque de respuesta para la operación de consultar la función de la FIG. 12A, según un aspecto de la presente invención;

10 La FIG. 13A representa una realización de un bloque de petición usado por la instrucción Llamar al Procesador Lógico de la FIG. 11A para una operación de consultar el grupo, según un aspecto de la presente invención;

La FIG. 13B representa una realización de un bloque de respuesta para la operación de consultar el grupo de la FIG. 13A, según un aspecto de la presente invención;

La FIG. 14 representa una realización de un producto de programa de ordenador que incorpora uno o más aspectos de la presente invención;

15 La FIG. 15 representa una realización de un sistema informático de ordenador principal para incorporar y usar uno o más aspectos de la presente invención;

La FIG. 16 representa un ejemplo adicional de un sistema informático para incorporar y usar uno o más aspectos de la presente invención;

20 La FIG. 17 representa otro ejemplo de un sistema informático que comprende una red informática para incorporar y usar uno o más aspectos de la presente invención;

La FIG. 18 representa una realización de diversos elementos de un sistema informático para incorporar y usar uno o más aspectos de la presente invención;

La FIG. 19A representa una realización de la unidad de ejecución del sistema informático de la FIG. 18 para incorporar y usar uno o más aspectos de la presente invención;

25 La FIG. 19B representa una realización de la unidad de ramal del sistema informático de la FIG. 18 para incorporar y usar uno o más aspectos de la presente invención;

La FIG. 19C representa una realización de la unidad de carga/almacenamiento del sistema informático de la FIG. 18 para incorporar y usar uno o más aspectos de la presente invención; y

30 La FIG. 20 representa una realización de un sistema informático de ordenador principal emulado para incorporar y usar uno o más aspectos de la presente invención.

Descripción detallada

35 Según un aspecto de la presente invención, se proporciona una capacidad para convertir una petición de interrupción señalada por mensaje (MSI) en una notificación de evento de adaptador de entrada/salida (I/O). La MSI se solicita por un adaptador y convierte a una notificación de evento de adaptador, en la que se fijan uno o más indicadores específicos y se hace una petición de que una interrupción sea presentada a un sistema operativo (u otro software, tal como otros programas, etc. Como se usa en la presente memoria, el término sistema operativo incluye controladores de dispositivo del sistema operativo). En un ejemplo particular, cada petición de MSI no provoca una petición de interrupción al sistema operativo, sino que en su lugar, una petición de interrupción abarca una pluralidad de peticiones de MSI.

40 Como se usa en la presente memoria, el término "adaptador" incluye cualquier tipo de adaptador (por ejemplo, adaptador de almacenamiento, adaptador de red, adaptador de procesamiento, adaptador criptográfico, adaptador de PCI, otro tipo de adaptador de entrada/salida, etc.). En una realización, un adaptador incluye una función de adaptador. No obstante, 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. Además, en los ejemplos presentados en la presente memoria, se usa adaptador intercambiamente con función de adaptador (por ejemplo, función de PCI) a menos que se señale de otro modo.

50 Una realización de un entorno informático para incorporar y usar uno o más aspectos de la presente invención se describe con referencia a la FIG. 1. En un ejemplo, un entorno informático 100 es un servidor System z[®] ofrecido por International Business Machines Corporation. System z[®] se basa en la z/Architecture[®] ofrecida por International Business Machines Corporation. Detalles con respecto a la 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 comerciales registradas de International Business Machines Corporation, Armonk, Nueva York. Otros nombres usados en la presente memoria pueden ser marcas comerciales registradas, marcas comerciales o nombres de productos de International Business Machines Corporation u otras compañías.

5 En un ejemplo, el entorno informático 100 incluye una o más unidades centrales de proceso (CPU) 102 acopladas a una memoria del sistema 104 (también conocida como memoria principal) a través de un controlador de memoria 106. Para acceder a la memoria del sistema 104, una unidad central de proceso 102 emite una petición de lectura o escritura que incluye una dirección usada para acceder a la memoria del sistema. La dirección incluida en la petición típicamente no es directamente utilizable para acceder a la memoria del sistema y, por lo tanto, se traduce a una
10 dirección que es directamente utilizable en el acceso a la memoria del sistema. La dirección se traduce a través de un mecanismo de traducción (XLATE) 108. Por ejemplo, la dirección se traduce desde una dirección virtual a una dirección real o absoluta usando, por ejemplo, traducción de direcciones dinámica (DAT).

La petición, que incluye la dirección (traducida, si es necesario), se recibe por el controlador de memoria 106. En un ejemplo, el controlador de memoria 106 se compone de hardware y se usa para arbitrar acceso a la memoria del sistema y para mantener la consistencia de la memoria. Este arbitraje se realiza para peticiones recibidas desde las CPU 102, así como para peticiones recibidas desde uno o más adaptadores 110. Como las unidades centrales de proceso, los adaptadores emiten peticiones a la memoria del sistema 104 para obtener acceso a la memoria del sistema.
15

En un ejemplo, el adaptador 110 es un adaptador de Interconexión de Componente Periférico (PCI) o PCI Rápido (PCIe) que incluye una o más funciones de PCI. Una función de PCI emite una petición que se encamina a un centro de entrada/salida 112 (por ejemplo, un centro de PCI) a través de uno o más conmutadores (por ejemplo, conmutadores de PCIe) 114. En un ejemplo, el centro de entrada/salida se compone de hardware, que incluye una o más máquinas de estado.
20

El centro de entrada/salida incluye, por ejemplo, un complejo raíz 116 que recibe la petición desde un conmutador. La petición incluye una dirección de entrada/salida que se usa para realizar un acceso directo a memoria (DMA) o para solicitar una interrupción señalada por mensaje (MSI), como ejemplos. Esta dirección se proporciona a una unidad de traducción y protección de direcciones 118 que accede a información usada o bien por el DMA o la petición de MSI.
25

Para una operación de DMA, la unidad de traducción y protección de direcciones 118 puede traducir la dirección a una dirección utilizable para acceder a la memoria del sistema. Entonces, la petición iniciada desde el adaptador, que incluye la dirección traducida, se proporciona al controlador de memoria 106 a través, por ejemplo, de un canal principal 120 de I/O a memoria. El controlador de memoria realiza su arbitraje y reenvía la petición con la dirección traducida a la memoria del sistema en el momento adecuado.
30

Para una petición de MSI, se obtiene información en la unidad de traducción y protección de direcciones 118 para facilitar la conversión de la petición de MSI a una notificación de evento de adaptador de I/O. Dado que las realizaciones descritas en la presente memoria se refieren a procesamiento de interrupción, detalles adicionales con respecto al centro de I/O y la memoria del sistema que se refieren a procesamiento de interrupción se describen con referencia a la FIG. 2. En la FIG. 2, el controlador de memoria no se muestra, pero se puede usar. El centro de I/O se puede acoplar a la memoria del sistema 104 y/o al procesador 254 directamente o a través del controlador de memoria.
35
40

Con referencia a la FIG. 2, en un ejemplo, la memoria del sistema 104 incluye una o más estructuras de datos utilizables en facilitar procesamiento de interrupción. En este ejemplo, la memoria del sistema 104 incluye un vector de bit de interrupción de adaptador (AIBV) 200 y un bit de resumen de interrupción de adaptador (AISB) 202 asociados con un adaptador particular. Puede haber un AIBV y un AISB correspondiente para cada adaptador.

45 En un ejemplo, el vector de bit de interrupción de adaptador 200 es una formación de dimensión única de uno o más bits en el almacenamiento principal que están asociados con un adaptador (por ejemplo, una función de PCI). Los bits en el vector de bit de interrupción de adaptador representan números de vectores de MSI. Un bit que se fija a uno en un AIBV indica una condición o tipo de evento para el adaptador asociado. En el ejemplo de una función de PCI, cada bit en el AIBV asociado corresponde a un vector de MSI. Por lo tanto, si una función de PCI soporta solamente un vector de MSI, su AIBV incluye un único bit; si una función de PCI soporta múltiples vectores de PCI, su AIBV incluye un bit por vector de MSI. En el ejemplo representado en la FIG. 2, la función de PCI soporta múltiples vectores de MSI (por ejemplo, 3) y, por lo tanto, hay múltiples bits (por ejemplo, 3) en el AIBV 200. Cada bit corresponde a un evento particular, por ejemplo, el bit 0 del AIBV, cuando se fija a uno, indica una operación completada; el bit 1 del AIBV, cuando se fija a uno, corresponde a un evento de error; etc. Como se muestra, el bit 1 se fija en este ejemplo.
50
55

En un ejemplo particular, un comando (por ejemplo, un comando Modificar los Controles de Función de PCI) se usa para designar un AIBV para una función de PCI. Específicamente, el comando se emite por el sistema operativo y especifica la identidad de la función de PCI, la ubicación de almacenamiento principal del área que incluye el AIBV,

5 el desplazamiento desde esa ubicación al primer bit del AIBV y el número de bits que comprenden el AIBV. En particular, usando este comando, los parámetros de interrupción de adaptador se copian desde un bloque de información de función que almacena tal información (por ejemplo, obtenida a partir de la inicialización y/o configuración) en la entrada de tabla de dispositivo del adaptador (descrita más adelante) y/o entrada de tabla de función (descrita más adelante).

10 La identidad de la función de PCI, en un ejemplo, es un gestor de función. Un gestor de función incluye, por ejemplo, un indicador de habilitación que indica si el gestor de función de PCI está habilitado; un número de función de PCI que identifica la función (esta es un identificador estático); y un número de caso que indica el caso particular de este gestor de función. Por ejemplo, cada vez que se habilita el gestor de función, el número de caso se aumenta para proporcionar un nuevo número de caso. El gestor de función se usa para localizar una entrada de tabla de función en una tabla de función que incluye una o más entradas. Por ejemplo, uno o más bits del gestor de función se usan como un índice en la tabla de función para localizar una entrada de tabla de función particular. La entrada de tabla de función incluye información con respecto a su función de PCI asociada. Por ejemplo, puede incluir varios indicadores con respecto al estado de su función de adaptador asociada y puede incluir uno o más índices de entrada de tabla de dispositivo usados para localizar las entradas de tabla de dispositivo, para esta función de adaptador. (Para el sistema operativo, en una realización, el gestor es simplemente un identificador opaco del adaptador).

20 Un AIBV se puede asignar en cualquier límite de byte o cualquier límite de bit. Esto permite al sistema operativo la flexibilidad de empaquetar los AIBV de múltiples adaptadores en un intervalo contiguo de bits y bytes. Por ejemplo, como se muestra en la FIG. 3A, en un ejemplo, el sistema operativo ha designado un área de almacenamiento común en la ubicación X para incluir cinco AIBV contiguos. El adaptador asociado con cada AIBV se identifica por las letras A-E. El evento que cada bit de AIBV representa para un adaptador se identifica además por los números 0-n. Los bits no asignados se identifican por la letra minúscula "u".

25 Un ejemplo adicional se representa en la FIG. 3B. En este ejemplo, el sistema operativo ha designado tres áreas de almacenamiento únicas, en las ubicaciones X, Y y Z para incluir los AIBV para cinco adaptadores de I/O. El almacenamiento en la ubicación X incluye los AIBV para los adaptadores A y B, el almacenamiento en la ubicación Y incluye el AIBV solamente para el adaptador C y el almacenamiento en la ubicación Z incluye los AIBV para los adaptadores D y E. El evento que representa cada bit de AIBV para un adaptador de I/O se identifica además por los números 0-n. Los bits no asignados se identifican por la letra "u".

30 Volviendo a la FIG. 2, además del AIBV, en este ejemplo, hay un AISB 202 para el adaptador, que incluye un único bit asociado con el adaptador. Un AISB que es uno indica que uno o más bits se han fijado a uno en un AIBV asociado con el AISB. El AISB es opcional y puede haber uno para cada adaptador, uno para cada adaptador seleccionado o uno para un grupo de adaptadores.

35 En una implementación particular para funciones de PCI, un comando (por ejemplo, un comando Modificar los Controles de Función de PCI) se usa para designar un AISB para una función de PCI. Específicamente, el comando se emite por el sistema operativo y especifica la identidad de la función de PCI (por ejemplo, el gestor), la ubicación del almacenamiento principal del área que incluye el AISB, el desplazamiento desde esa ubicación al AISB y un control de habilitación de notificación de resumen de interrupción de adaptador que indica que hay un bit de resumen.

40 Un AISB se puede asignar en cualquier límite de byte o cualquier límite de bit. Esto permite al sistema operativo la flexibilidad de empaquetar los AISB de múltiples adaptadores en un intervalo contiguo de bits o bytes. En un ejemplo, como se representa en la FIG. 3C, el sistema operativo ha designado un área de almacenamiento común, en la ubicación X, para incluir nueve AISB contiguos. El adaptador asociado con cada AISB se identifica por las letras A-I. Los bits no asignados se identifican por la letra minúscula "u".

45 Un ejemplo de asignación adicional se representa en la FIG. 3D, donde el sistema operativo ha designado tres ubicaciones de almacenamiento de AISB únicas, en las ubicaciones X, Y y Z para incluir los AISB de cada uno de los tres adaptadores. Los adaptadores asociados con cada AISB se identifican con las letras A-C. Los bits no asignados se identifican por la letra minúscula "u".

50 Además, el programa también puede asignar un AISB único a múltiples funciones de PCI. Este asocia múltiples AIBV con un bit resumen único. Por lo tanto, tal AISB que es uno indica que el sistema operativo debería explorar múltiples AIBV.

Volviendo a la FIG. 2, en un ejemplo, el AIBV y el AISB se indican por direcciones localizadas en una entrada de tabla de dispositivo 206 de una tabla de dispositivo 208 situada en un centro de I/O 112. En un ejemplo, la tabla de dispositivo 208 está localizada dentro de la unidad de traducción protección de dirección del centro de I/O.

55 La tabla de dispositivo 208 incluye una o más entradas 206, cada una de las cuales se asigna a una función de adaptador particular 210. Una entrada de tabla de dispositivo 206 incluye un número de campos, que se pueden rellenar usando, por ejemplo, los comandos mencionados anteriormente. Los valores de uno o más de los campos se basan en la política y/o configuración. Ejemplos de campos incluyen:

Subclase de Interrupción (ISC) 214: Indica una subclase de interrupción para la interrupción. La ISC identifica una clase enmascarable de interrupciones de adaptador que se pueden asociar con una prioridad con la que el sistema operativo procesará la interrupción;

5 Dirección de AIBV (@) 216: Proporciona, por ejemplo, una dirección absoluta del comienzo de la ubicación del almacenamiento que incluye el AIBV para la función de adaptador particular asignada a esta entrada de tabla de dispositivo;

Desplazamiento de AIBV 218: Un desplazamiento en la ubicación de almacenamiento principal al comienzo del AIBV;

10 Dirección de AISB (@) 220: Proporciona, por ejemplo, una dirección absoluta del comienzo de la ubicación del almacenamiento que incluye el AISB para esta función de PCI, si el sistema operativo ha designado un AISB;

Desplazamiento de AISB 222: Un desplazamiento en la ubicación de almacenamiento principal para el AISB;

Control de Habilitación de Notificación de Resumen de Interrupción de Adaptador (Habilitar) 224: Este control indica si hay un AISB;

15 Número de Interrupciones (NOI) 226: Indica el número máximo de vectores de MSI permitidos para esta función de PCI, con cero que indica ninguno permitido.

En otras realizaciones, la DTE puede incluir más, menos o diferente información.

20 En una realización, la entrada de tabla de dispositivo a ser usada para una petición de interrupción particular por un adaptador se localiza usando, por ejemplo, un identificador de solicitante (RID) (y/o una porción de la dirección) localizado en una petición emitida por el adaptador (por ejemplo, la función de PCI 210). El ID de solicitante (por ejemplo, un valor de 16 bits que especifica, por ejemplo, un número de canal principal, número de dispositivo y número de función) se incluye en la petición, así como una dirección a ser usada para la interrupción. La petición, que incluye el RID y la dirección, se proporcionan a, por ejemplo, una memoria direccionable de contenidos (CAM 230) a través, por ejemplo, de un conmutador y la memoria direccionable de contenidos se usa para proporcionar un valor de índice. Por ejemplo, la CAM incluye múltiples entradas, con cada entrada correspondiente a un índice dentro de la tabla de dispositivo. Cada entrada de CAM incluye el valor de un RID. Si, por ejemplo, el RID recibido coincide con el valor contenido en una entrada en la CAM, el índice de tabla de dispositivo correspondiente se usa para localizar la entrada de tabla de dispositivo. Es decir, la salida de la CAM se usa para indexar dentro de la tabla de dispositivo 208. Si no hay coincidencia, el paquete recibido se descarta. (En otras realizaciones, una CAM u otra búsqueda no es necesaria y el RID se usa como el índice). La DTE localizada se usa en el procesamiento de una petición de interrupción, como se describe en la presente memoria.

35 Para solicitar una interrupción, la función de adaptador 210 envía un paquete al centro de I/O. Este paquete tiene una dirección de MSI 232 y unos datos asociados 234. El centro de I/O compara al menos una parte de la dirección recibida con un valor en un registro de comparación de MSI 250. Si hay una coincidencia, entonces una interrupción (por ejemplo, MSI) está siendo solicitada, en contraposición a una operación de DMA. La razón para la petición (es decir, el tipo de evento que ha ocurrido) se indica en unos datos asociados 234. Por ejemplo, uno o más de los bits de orden bajo de los datos se usan para especificar un vector de interrupción particular (es decir, un vector de MSI) que indica la razón (evento).

40 Según un aspecto de la presente invención, la petición de interrupción recibida desde el adaptador se convierte en una notificación de evento de adaptador de I/O. Es decir, se fijan uno o más indicadores (por ejemplo, uno o más AIBV y opcionalmente un AISB) y se solicita una interrupción al sistema operativo, si una no está ya pendiente. En una realización, múltiples peticiones de interrupción (por ejemplo, MSI) desde uno o más adaptadores se fusionan en una única interrupción al sistema operativo pero con indicaciones de AIBV y AISB respectivas. Por ejemplo, si el centro de I/O ya ha recibido una petición de MSI, ha proporcionado, a su vez, una petición de interrupción a un procesador y esa interrupción está aún pendiente (por ejemplo, por una razón u otra, la interrupción no se ha presentado al sistema operativo (por ejemplo, las interrupciones están deshabilitadas)), entonces si el centro recibe una o más otras MSI, no solicita interrupciones adicionales. La interrupción sustituye y representa la pluralidad de peticiones de MSI. No obstante, se fija uno o más AIBV y opcionalmente uno o más AISB.

50 Detalles adicionales con respecto a convertir una MSI (u otra petición de interrupción de adaptador) a una notificación de evento de adaptador de I/O se describen más adelante con referencia a las FIG. 4-6B. Particularmente, la FIG. 4 describe diversas inicializaciones a ser realizadas; la FIG. 5 describe un proceso de registro; la FIG. 6A describe una lógica para convertir una MSI a una notificación de evento de adaptador; y la FIG. 6B describe una lógica para presentar la notificación de evento de adaptador de I/O al sistema operativo.

55 Con referencia a la FIG. 4, en un ejemplo, para convertir una petición de MSI a una notificación de evento de adaptador de I/O se realiza cierta inicialización. Durante la inicialización, el sistema operativo realiza una serie de pasos para configurar un adaptador para notificación de evento de adaptador a través de una petición de MSI. En

este ejemplo, es una función de PCI la que se configura; aunque, en otras realizaciones, pueden ser otros adaptadores, incluyendo otros tipos de funciones de adaptador.

Inicialmente, en una realización, se toma una determinación en cuanto a las funciones de PCI en la configuración, PASO 400. En un ejemplo, un comando (por ejemplo, un comando Consultar Lista) emitido por el sistema operativo se usa para obtener una lista de las funciones de PCI asignadas a la configuración solicitante (por ejemplo, asignada a un sistema operativo particular). Esta información se obtiene a partir de una estructura de datos de configuración que mantiene esta información.

A continuación, se selecciona una de las funciones de PCI en la lista, PASO 402 y se toma una determinación en cuanto a la dirección de MSI a ser usada para la función de PCI y el número de vectores de MSI soportados por la función de PCI. La dirección de MSI se determina en base a las características del centro de I/O y el sistema en el que se instala. El número de vectores de MSI soportado se basa en la política y es configurable.

Adicionalmente, el AIBV se asigna, así como el AISB, en su caso, PASO 410. En un ejemplo, el sistema operativo determina la ubicación del AIBV para permitir un procesamiento eficiente de uno o más adaptadores, típicamente en base a la clase de adaptador. Por ejemplo, los AIBV para adaptadores de almacenamiento se pueden localizar adyacentes unos de otros. El AIBV y el AISB se asignan y limpian a ceros y se especifica una operación de interrupción de adaptador de registro (por ejemplo, usando una instrucción Modificar los Controles de Función de PCI). Esta operación registra el AIBV, el AISB, la ISC, el número de interrupciones (vectores de MSI) y el control de habilitación de notificación de resumen de interrupción de adaptador, como se describe en más detalle más adelante, PASO 412. A partir de entonces, se lee/escrbe el espacio de configuración de la función de PCI, PASO 414. Específicamente, la dirección de MSI y el recuento de vectores de MSI se escriben coherentes con el registro previo.

A partir de entonces, se toma una determinación en cuanto a si hay funciones adicionales en la lista, PREGUNTA 416. Si es así, el proceso continúa con el PASO 402. De otro modo, se completa el proceso de inicialización.

Detalles adicionales con respecto al registro de diversos parámetros se describen con referencia a la FIG. 5. Inicialmente, se selecciona la entrada de tabla de dispositivo (DTE) para corresponder con la función de PCI para la cual está siendo realizada la inicialización. Esta selección se realiza, por ejemplo, por el microprograma de gestión que selecciona una DTE disponible a partir de la tabla de dispositivo. A partir de entonces, los diversos parámetros se almacenan en la entrada de tabla de dispositivo, PASO 502. Por ejemplo, la ISC, la dirección de AIBV, el desplazamiento de AIBV, la dirección de AISB, el desplazamiento de AISB, el control de habilitación y el número de interrupciones (NOI) se fijan a valores obtenidos a partir de la configuración de la función. Esto completa el proceso de registro.

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

Durante la operación, cuando una función de PCI desea generar una MSI, típicamente pone a disposición alguna información para el sistema operativo que describe la condición. Esto hace que ocurra uno o más pasos a fin de convertir la petición de MSI de la función de PCI a una notificación de evento de adaptador de I/O para el sistema operativo. Esto se describe con referencia a la FIG. 6A.

Con referencia a la FIG. 6A, inicialmente, se registra una descripción del evento para el cual se solicita la interrupción, PASO 600. Por ejemplo, la función de PCI registra una descripción del evento en una o más estructuras almacenadas de registro de descripción de evento específicas del adaptador, por ejemplo, en la memoria del sistema. Esto puede incluir registrar el tipo de evento, así como registrar información adicional. Adicionalmente, una petición se inicia por la función de PCI especificando la dirección de MSI y el número de vector de MSI, así como un ID de solicitante, PASO 601. Esta petición se recibe por el centro de I/O y en respuesta a recibir la petición, el ID de solicitante en la petición se usa para localizar la entrada de tabla de dispositivo para la función de PCI, PASO 602. El centro de I/O compara al menos una parte de la dirección en la petición con el valor en el registro de comparación de MSI, PREGUNTA 603. Si no son iguales, no está siendo solicitada una MSI. No obstante, si son iguales, entonces se ha especificado una dirección de MSI y de esta manera, se ha solicitado una MSI, en lugar de una operación de acceso a memoria directa.

A partir de entonces, se toma una determinación en cuanto a si el número de vector de MSI especificado en la petición es menor o igual que el número de interrupciones (NOI) permitido para esta función, PREGUNTA 604. Si el número de vector de MSI es mayor que el NOI, se indica un error. De otro modo, el centro de I/O emite una función de bit establecida para fijar el bit de AIBV adecuado en el almacenamiento. El bit adecuado se determina añadiendo el número de vector de MSI al desplazamiento de AIBV especificado en la entrada de tabla de dispositivo y desplazando este número de bits desde la dirección de AIBV especificada en la entrada de tabla de dispositivo,

PASO 605. Además, si se ha designado un AISB, el centro de I/O usa una función de bit establecida para fijar el AISB, usando la dirección de AISB y el desplazamiento de AISB en la entrada de tabla de dispositivo, PASO 606.

5 A continuación, en una realización, se toma una determinación (por ejemplo, por la CPU o el centro de I/O) en cuanto a si ya está pendiente una petición de interrupción. Para hacer esta determinación, se usa un indicador pendiente. Por ejemplo, se comprueba un indicador pendiente 252 (FIG. 2) almacenado en la memoria de un procesador 254, que es accesible a procesadores del entorno informático que pueden procesar la interrupción (por ejemplo, las CPU 102 de la FIG. 1), PREGUNTA 608. Si no está fijado, entonces se fija (por ejemplo, a 1), PASO 610. Si ya está fijado, se completa el procesamiento y no se requiere otra petición de interrupción. Por lo tanto, las peticiones de interrupción posteriores se abarcan por la petición ya pendiente.

10 En un ejemplo particular, puede haber un indicador pendiente por subclase de interrupción y, por lo tanto, el indicador pendiente de la subclase de interrupción asignada a la función de petición es el indicador que se comprueba.

15 Asíncronamente, como se representa en la FIG. 6B, uno o más procesadores comprueban el indicador pendiente, PREGUNTA 640. En particular, cada procesador habilitado para la ISC (y zona en otra realización) sondea el indicador para la ISC cuando, por ejemplo, se habilitan interrupciones para ese procesador (es decir, para su sistema operativo). Si uno de los procesadores determina que el indicador está fijado, arbitra con los otros procesadores habilitados para la misma ISC (y zona en otra realización) para presentar la interrupción, PASO 642. Volviendo a la PREGUNTA 640, si el indicador pendiente no está fijado, los procesadores habilitados para la ISC continúan sondeando un indicador fijado.

20 En respuesta al sistema operativo que se presenta con la interrupción, PASO 642, el sistema operativo determina si están registrados cualesquiera AISB, PREGUNTA 643. Si no están, el sistema operativo procesa los AIBV fijados, como se describe más adelante, PASO 645. De otro modo, el sistema operativo procesa cualesquiera AISB y AIBV fijados, PASOS 644, 645. Por ejemplo, comprueba si están fijados cualesquiera AISB. Si es así, usa el AISB para determinar la ubicación de uno o más AIBV. Por ejemplo, el sistema operativo recuerda las ubicaciones de los AISB y AIBV. Por otra parte, recuerda qué adaptador representa cada AISB y AIBV. Por lo tanto, puede mantener una forma de un bloque de control u otra estructura de datos que incluya las ubicaciones de AISB y AIBV y la asociación entre AISB, AIBV y el ID de adaptador. Usa este bloque de control para facilitar la ubicación de un AIBV en base a su AISB asociado. En una realización adicional, no se usa un AISB. En esa situación, el bloque de control se usa para localizar el AIBV particular.

30 En respuesta a localizar el uno o más AIBV, el sistema operativo explora los AIBV y procesa cualquier AIBV fijado. Procesa la interrupción de una manera coherente con el evento presentado (por ejemplo, proporciona el estado). Por ejemplo, con un adaptador de almacenamiento, un evento puede indicar que se ha completado una operación. Esto provoca al sistema operativo comprobar el estado almacenado por el adaptador para ver si la operación se completó con éxito y también los detalles de la operación. En el caso de una lectura de almacenamiento, esta es una indicación de que los datos leídos desde el adaptador están disponibles ahora en la memoria del sistema y se pueden procesar.

En una realización, si durante la operación de la conversión, se detecta un error, se genera una atención para el microprograma del sistema, en lugar de convertir la petición de MSI a una notificación de evento de adaptador.

40 Se describen en la presente memoria detalles adicionales con respecto a la instrucción Modificar los Controles de Función de PCI usada para registrar interrupciones de adaptador. Con referencia a la FIG. 7A, una instrucción Modificar los Controles de Función de PCI 700 incluye, por ejemplo, un código operativo 702 que indica la instrucción Modificar los Controles de Función de PCI; un primer campo 704 que especifica una ubicación en la que se incluye diversa información con respecto a la función de adaptador para la que los parámetros de operación están siendo establecidos; y un segundo campo 706 que especifica una ubicación para la cual se trae un bloque de información de función (FIB) de PCI. Los contenidos de las ubicaciones designadas por los Campos 1 y 2 se describen además más adelante.

50 En una realización, el Campo 1 designa un registro general que incluye diversa información. Como se muestra en la FIG. 7B, los contenidos del registro incluyen, por ejemplo, un gestor de función 710 que identifica el gestor de la función de adaptador en nombre del cual está siendo realizada la instrucción modificar; un espacio de direcciones 712 que designa un espacio de direcciones en la memoria del sistema asociado con la función de adaptador designada por el gestor de función; un control de operación 714 que especifica la operación a ser realizada por la función de adaptador; y un estado 716 que proporciona el estado con respecto a la instrucción cuando la instrucción se completa con un código predefinido.

55 En una realización, el gestor de función incluye, por ejemplo, un indicador de habilitación que indica si está habilitado el gestor, un número de función que identifica una función de adaptador (este es un identificador estático y se puede usar para indexar en una tabla de función); y un número de caso que especifica el caso particular de este gestor de función. Hay un gestor de función para cada función de adaptador y se usa para localizar una entrada de tabla de

función (FTE) dentro de la tabla de función. Cada entrada de tabla de función incluye parámetros de operación y/u otra información asociada con su función de adaptador. Como ejemplo, una entrada de tabla de función incluye:

Número de Caso: Este campo indica un caso particular del gestor de función de adaptador asociado con la entrada de tabla de función;

5 Índice de Entrada de Tabla de Dispositivo (DTE) 1...n: Puede haber uno o más índices de tabla de dispositivo y cada índice es un índice en una tabla de dispositivo para localizar una entrada de tabla de dispositivo (DTE). Hay una o más entradas de tabla de dispositivo por función de adaptador y cada entrada incluye información asociada con su función de adaptador, incluyendo información usada para procesar peticiones de la función de adaptador (por ejemplo, peticiones de DMA, peticiones de MSI) e información relativa a peticiones asociadas con la función de adaptador (por ejemplo, instrucciones de PCI). Cada entrada de tabla de dispositivo se asocia con un espacio de direcciones dentro de la memoria del sistema asignada a la función de adaptador. Una función de adaptador puede tener uno o más espacios de direcciones dentro de la memoria del sistema asignada a la función de adaptador.

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

15 Indicador de Estado de Error Permanente: Este campo indica si la función de adaptador está en un estado de error permanente;

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

20 Indicador de Permiso: Este campo indica si el sistema operativo que intenta controlar la función de adaptador tiene autoridad para hacerlo así;

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

Identificador de Solicitante (RID): Este es un identificador de la función de adaptador e incluye, por ejemplo, un número de canal principal, un número de dispositivo y un número de función.

25 En un ejemplo, este campo se usa para accesos de un espacio de configuración de la función de adaptador. (La memoria de un adaptador se puede definir como espacios de direcciones, que incluyen, por ejemplo, un espacio de configuración, un espacio de I/O y/o uno o más espacios de memoria). En un ejemplo, se puede acceder al espacio de configuración especificando el espacio de configuración en una instrucción emitida por el sistema operativo (u otra configuración) para la función de adaptador. Especificado en la instrucción está un desplazamiento dentro del espacio de configuración y un gestor de función usado para situar la entrada de tabla de función adecuada que incluye el RID. El microprograma recibe la instrucción y determina que es para un espacio de configuración. Por lo tanto, usa el RID para generar una petición al centro de I/O y el centro de I/O crea una petición para acceder al adaptador. La ubicación de la función de adaptador se basa en el RID y el desplazamiento especifica un desplazamiento dentro del espacio de configuración de la función de adaptador.

30 35 Registro de Dirección Base (BAR) (1 a n): Este campo incluye una pluralidad de enteros sin signo, designados como BAR₀ – BAR_n, que están asociados con la función de adaptador especificada originalmente y cuyos valores también están almacenados en los registros de direcciones base asociados con la función de adaptador. Cada BAR especifica la dirección de inicio de un espacio de memoria o espacio de I/O dentro de la función de adaptador y también indica el tipo de espacio de direcciones, que es si es un espacio de memoria de 64 o 32 bits o un espacio de I/O de 32 bits, como ejemplos.

40 45 En un ejemplo, se usa para accesos al espacio de memoria y/o el espacio de I/O de la función de adaptador. Por ejemplo, un desplazamiento proporcionado en una instrucción para acceder a la función de adaptador se añade al valor en el registro de dirección base asociado con el espacio de direcciones designado en la instrucción para obtener la dirección a ser usada para acceder a la función de adaptador. El identificador de espacio de direcciones proporcionado en la instrucción identifica el espacio de direcciones dentro de la función de adaptador a ser accedida y el BAR correspondiente a ser usado;

Tamaño 1...n: Este campo incluye una pluralidad de enteros sin signo, designados como TAMAÑO₁ – TAMAÑO_n. El valor de un campo Tamaño, cuando no es cero, representa el tamaño de cada espacio de direcciones con cada entrada que corresponde a un BAR descrito previamente.

50 Se describen más adelante detalles adicionales con respecto al BAR y Tamaño.

1. Cuando un BAR no se implementa para una función de adaptador, el campo de BAR y su campo de tamaño correspondiente se almacenan ambos como ceros.

2. Cuando un campo de BAR representa o bien un espacio de direcciones de I/O o bien un espacio de direcciones de memoria de 32 bit, el campo de tamaño correspondiente no es cero y representa el tamaño del espacio de direcciones.

3. Cuando un campo de BAR representa un espacio de direcciones de memoria de 64 bits,

- 5
- a. El campo de BAR_n representa los bits de dirección menos significativos,
 - b. El siguiente campo de BAR_{n+1} consecutivo representa los bits de dirección más significativos.
 - c. El campo de $TAMAÑO_n$ correspondiente es no cero y representa el tamaño del espacio de direcciones.
 - d. El campo de $TAMAÑO_{n+1}$ correspondiente no es significativo y se almacena como cero.

10 Información de Encaminamiento Interno: Esta información se usa para realizar un encaminamiento particular al adaptador. Incluye, por ejemplo, un nodo, circuito integrado de procesador e información de direccionamiento de centro, como ejemplos.

Indicación de Estado: Esta proporciona una indicación de, por ejemplo, si se bloquean operaciones de carga/almacenamiento o el adaptador está en el estado de error, así como otras indicaciones.

15 En un ejemplo, el indicador de ocupado, indicador de estado de error permanente y el indicador de recuperación iniciada se fijan en base a la monitorización realizada por el microprograma. Además, el indicador de permiso se fija, por ejemplo, en base a la política; y la información de BAR se basa en la información de configuración descubierta durante un paseo por el canal principal por el procesador (por ejemplo, el microprograma del procesador). Otros campos se pueden fijar en base a configuración, inicialización y/o eventos. En otras realizaciones, la entrada de tabla de función puede incluir más, menos o diferente información. La información
20 incluida puede depender de las operaciones soportadas por o habilitadas para la función de adaptador.

Con referencia a la FIG. 7C, en un ejemplo, el Campo 2 designa una dirección lógica 720 de un bloque de información de función (FIB) de PCI, que incluye información con respecto a una función de adaptador asociada. El bloque de información de función se usa para actualizar una entrada de tabla de dispositivo y/o entrada de tabla de función (u otra ubicación) asociada con la función de adaptador. La información se almacena en el FIB durante la
25 inicialización y/o configuración del adaptador y/o en respuesta a eventos particulares.

Detalles adicionales con respecto a un bloque de información de función (FIB) se describen con referencia a la FIG. 7D. En una realización, un bloque de información de función 750 incluye los siguientes campos:

Formato 751: Este campo especifica el formato del FIB.

30 Control de Interceptación 752: Este campo se usa para indicar si la ejecución invitada de instrucciones específicas por un invitado de modo paginable provoca una interceptación de instrucción;

Indicación de error 754: Este campo incluye la indicación de estado de error para acceso directo a memoria e interrupciones de adaptador. Cuando el bit está fijado (por ejemplo, a 1), se han detectado uno o más errores mientras que se realiza un acceso directo a memoria o una interrupción de adaptador para la función de adaptador;

35 Carga/Almacenamiento Bloqueado 756: Este campo indica si están bloqueadas las operaciones de carga/almacenamiento;

Función de PCI Válida 758: Este campo incluye un control de habilitación para la función de adaptador. Cuando el bit está fijado (por ejemplo, a 1), la función de adaptador se considera que está habilitada para operaciones de I/O;

40 Espacio de Direcciones Registrado 760: Este campo incluye un control de habilitación de acceso directo a memoria para una función de adaptador. Cuando el campo está fijado (por ejemplo, a 1) está habilitado el acceso directo a memoria;

Tamaño de Página 761: Este campo indica el tamaño de la página u otra unidad de memoria a ser accedida por un acceso a memoria DMA;

45 Dirección Base de PCI (PBA) 762: Este campo es una dirección base para un espacio de direcciones en la memoria del sistema asignado a la función de adaptador. Representa la dirección virtual más baja que se permite usar a una función de adaptador para acceso directo a memoria al espacio de direcciones de DMA especificado;

Límite de Direcciones de PCI (PAL) 764: Este campo representa la dirección virtual más alta que se permite acceder a una función de adaptador dentro del espacio de direcciones de DMA especificado;

Puntero de Traducción de Direcciones de Entrada/Salida (IOAT) 766: El puntero de traducción de direcciones de entrada/salida designa la primera de cualquier tabla de traducción usada por una traducción de direcciones virtual de PCI o puede designar directamente la dirección absoluta de una trama de almacenamiento que es el resultado de la traducción;

5 Subclase de Interrupción (ISC) 768: Este campo incluye la subclase de interrupción usada para presentar interrupciones de adaptador para la función de adaptador;

Número de Interrupciones (NOI) 770: Este campo designa el número de códigos de interrupción distintos aceptados para una función de adaptador. Este campo también define el tamaño, en bits, del vector de bit de interrupción de adaptador designado por unos campos de dirección de vector de bit de interrupción de adaptador y desplazamiento de vector de bit de interrupción de adaptador;

10 Dirección de Vector de Bit de Interrupción de Adaptador (AIBV) 772: Este campo especifica una dirección del vector de bit de interrupción de adaptador para la función de adaptador. Este vector se usa en el proceso de interrupción;

15 Desplazamiento de Vector de Bit de Interrupción de Adaptador 774: Este campo especifica el desplazamiento del primer bit de vector de bit de interrupción de adaptador para la función de adaptador;

Dirección de Bit de Resumen de Interrupción de Adaptador (AISB) 776: Este campo proporciona una dirección que designa el bit de resumen de interrupción de adaptador, que se usa opcionalmente en el proceso de interrupción;

20 Desplazamiento de Bit de Resumen de Interrupción de Adaptador 778: Este campo proporciona el desplazamiento en el vector de bit de resumen de interrupción de adaptador;

Dirección de Bloque de Medición de Función (FMB) 780: Este campo proporciona una dirección de un bloque de medición de función usado para recopilar mediciones con respecto a la función de adaptador;

Clave de Bloque de Medición de Función 782: Este campo incluye una clave de acceso para acceder al bloque de medición de función;

25 Control de Notificación de Bit de Resumen 784: Este campo indica si hay un vector de bit de resumen que se usa;

Testigo de Autorización de Instrucción 786: Este campo se usa para determinar si se autoriza a un invitado de modo de almacenamiento paginable para ejecutar instrucciones de PCI sin intervención del ordenador principal.

30 En un ejemplo, en la z/Architecture[®], un invitado paginable se ejecuta interpretativamente a través de la instrucción Ejecución Interpretativa de Inicio (SIE), en el nivel 2 de interpretación. Por ejemplo, el hipervisor de partición lógica (LPAR) ejecuta la instrucción de SIE para comenzar la partición lógica en la memoria física, fija. Si z/VM[®] es el sistema operativo en esa partición lógica, emite la instrucción de SIE para ejecutar sus máquinas invitadas (virtuales) en su almacenamiento V=V (virtual). Por lo tanto, el hipervisor de LPAR usa SIE de nivel 1 y el hipervisor de z/VM[®] usa SIE de nivel 2; y

35 Formato de Traducción de Direcciones 787: Este campo indica un formato seleccionado para traducción de direcciones de la tabla de traducción del nivel más alto a ser usada en la traducción (por ejemplo, la tabla de segmentos, 3ª región, etc.).

40 El bloque de información de función designado en la instrucción Modificar los Controles de Función de PCI se usa para modificar una entrada de tabla de dispositivo seleccionada, una entrada de tabla de función y/u otros controles de microprograma asociados con la función de adaptador designada en la instrucción. Modificando la entrada de tabla de dispositivo, la entrada de tabla de función y/u otros controles de microprograma, se proporcionan ciertos servicios para el adaptador. Estos servicios incluyen, por ejemplo, interrupciones de adaptador; traducciones de direcciones; reiniciar estado de error; reiniciar carga/almacenamiento bloqueado; fijar parámetros de medición de función y fijar control de interceptación.

45 Una realización de la lógica asociada con la instrucción Modificar los Controles de Función de PCI se describe con referencia a la FIG. 8. En un ejemplo, la instrucción se emite por un sistema operativo (u otra configuración) y ejecuta por el procesador (por ejemplo, microprograma) que ejecuta el sistema operativo. En los ejemplos en la presente memoria, las funciones de adaptador e instrucción se basan en PCI. No obstante, en otros ejemplos, se puede usar una arquitectura de adaptador diferente y las instrucciones correspondientes.

50 En un ejemplo, el sistema operativo proporciona los siguientes operandos a la instrucción (por ejemplo, en uno o más registros designados por la instrucción): el gestor de función de PCI; el identificador de espacio de direcciones de DMA; un control de operación; y una dirección del bloque de información de función.

Con referencia a la FIG. 8, inicialmente, se toma una determinación en cuanto a si está instalada la facilidad que permite una instrucción Modificar los Controles de Función de PCI, PREGUNTA 800. Esta determinación se hace, por ejemplo, comprobando un indicador almacenado, por ejemplo, en un bloque de control. Si no está instalada la facilidad, se proporciona una condición de excepción, PASO 802. De otro modo, se toma una determinación en cuanto a si la instrucción fue emitida por un invitado de modo de almacenamiento paginable (u otro invitado), PREGUNTA 804. Si es sí, el sistema operativo del ordenador principal emulará la operación para ese invitado, PASO 806.

De otro modo, se toma una determinación en cuanto a si están alineados uno o más operandos, PREGUNTA 808. Por ejemplo, se toma una determinación en cuanto a si la dirección del bloque de información de función está en un límite de palabra doble. En un ejemplo, esto es opcional. Si los operandos no están alineados, entonces se proporciona una condición de excepción, PASO 810.

De otro modo, se toma una determinación en cuanto a si está accesible el bloque de información de función, PREGUNTA 812. Si no está, entonces se proporciona una condición de excepción, PASO 814. De otro modo, se toma una determinación en cuanto a si está habilitado el gestor proporcionado en los operandos de la instrucción Modificar los Controles de Función de PCI, PREGUNTA 816. En un ejemplo, se toma esta determinación comprobando un indicador de habilitación en el gestor. Si el gestor no está habilitado, entonces se proporciona una condición de excepción, PASO 818.

Si el gestor está habilitado, entonces se usa el gestor para localizar una entrada de tabla de función, PASO 820. Es decir, al menos una parte del gestor se usa como un índice en la tabla de función para localizar la entrada de tabla de función que corresponde a la función de adaptador para la cual van a ser establecidos los parámetros de operación.

Se toma una determinación en cuanto a si se encontró la entrada de tabla de función, PREGUNTA 822. Si es no, entonces se proporciona una condición de excepción, PASO 824. De otro modo, si la configuración que emite la instrucción es una invitación, PREGUNTA 826, entonces se proporciona una condición de excepción (por ejemplo, una interceptación al ordenador principal), PASO 828. Esta pregunta se puede ignorar si la configuración no es una invitación o se pueden comprobar otras autorizaciones, si se designa.

Se toma entonces una determinación en cuanto a si está habilitada la función, PREGUNTA 830. En un ejemplo, se toma esta determinación comprobando un indicador de habilitación en la entrada de tabla de función. Si no está habilitada, entonces se proporciona una condición de excepción, PASO 832.

Si la función está habilitada, entonces se toma una determinación en cuanto a si está activa una recuperación, PREGUNTA 834. Si la recuperación está activa como se determina por un indicador de recuperación en la entrada de tabla de función, entonces se proporciona una condición de excepción, PASO 836. No obstante, si la recuperación no está activa, entonces se toma una determinación adicional en cuanto a si la función está ocupada, PREGUNTA 838. Esta determinación se toma comprobando el indicador de ocupado en la entrada de tabla de función. Si la función está ocupada, entonces se proporciona una condición de ocupada, PASO 840. Con la condición de ocupada, se puede volver a intentar la instrucción, en lugar de descartar.

Si la función no está ocupada, entonces se toma una determinación adicional en cuanto a si es válido el formato de bloque de información de función, PREGUNTA 842. Por ejemplo, se comprueba el campo de formato del FIB para determinar si este formato está soportado por el sistema. Si es inválido, entonces se proporciona una condición de excepción, PASO 844. Si el formato de bloque de información de función es válido, entonces se toma una determinación adicional en cuanto a si es válido el control de operación especificado en los operandos de la instrucción, PREGUNTA 846. Es decir, es el control de operación uno de los controles de operación especificados para esta instrucción. Si no es válido, entonces se proporciona una condición de excepción, PASO 848. No obstante, si el control de operación es válido, entonces el proceso continúa con el control de operación específico que se especifica.

En un ejemplo, el control de operación es una operación de interrupciones de adaptador de registro, que se usa para controlar las interrupciones de adaptador. En respuesta a este control de operación, los parámetros de función de adaptador relevantes a interrupciones de adaptador se fijan en la entrada de tabla de dispositivo en base a los contenidos adecuados del bloque de información de función.

Una realización de la lógica asociada con esta operación se describe con referencia a la FIG. 9. Como ejemplo, los operandos para esta operación, que se obtienen a partir del bloque de información de función, incluyen por ejemplo: una subclase de interrupción (ISC); el número de interrupciones (NOI) permitidas; un desplazamiento de vector de bit de interrupción de adaptador (AIBVO); una notificación de resumen (S); un desplazamiento de vector de bit de resumen de interrupción de adaptador (ABVSO); una dirección de vector de bit de interrupción de adaptador (AIBV); y una dirección de vector de bit de resumen de interrupción de adaptador (AISB).

Con referencia a la FIG. 9, inicialmente, se toma una determinación en cuanto a si el número de interrupciones (NOI) especificadas en el FIB es mayor que un máximo dependiente del modelo, PREGUNTA 900. Si es así, entonces se proporciona una condición de excepción, PASO 902. No obstante, si el número de interrupciones no es mayor que el

- máximo dependiente del modelo, entonces se toma una determinación en cuanto a si el número de interrupciones sumadas al desplazamiento de vector de bit de interrupción de adaptador (NOI + AIBVO) es mayor que el máximo dependiente del modelo, PREGUNTA 904. Si es así, entonces se proporciona una condición de excepción, PASO 906. Si el NOI más el AIBVO no es mayor que un máximo dependiente del modelo, entonces se toma una
- 5 determinación en cuanto a si la dirección de AIBV más el NOI abarca un límite de 4k, PREGUNTA 908. Si abarca el límite de 4k, entonces se proporciona una condición de excepción, PASO 910. De otro modo, se toma una determinación en cuanto a si están disponibles recursos suficientes para cualesquiera recursos necesarios, PASO 912. Si no hay suficientes recursos, entonces se proporciona una condición de excepción, PASO 914.
- De otro modo, se toma una determinación en cuanto a si ya están registradas las interrupciones de adaptador para esta función, PASO 916. En una realización, esto se determinaría comprobando uno o más de los parámetros (por ejemplo, en la DTE/FTE). En particular, se comprueban los parámetros asociados con las interrupciones, tales como el NOI. Si los campos están rellenos, entonces el adaptador se registra para las interrupciones. Si el adaptador ya está registrado, entonces se proporciona una condición de excepción, PASO 918. De otro modo, se obtienen los parámetros de interrupción desde el FIB y se colocan en la entrada de tabla de dispositivo y, opcionalmente, en la
- 10 entrada de tabla de función (FTE) correspondiente (u otra ubicación especificada). También, se fija un indicador de habilitación de MSI en la DTE, PASO 920. Es decir, los parámetros de función de PCI relevantes a las interrupciones de adaptador se fijan en la DTE y, opcionalmente, en la FTE en base a la información recuperada desde el bloque de información de función. Estos parámetros incluyen, por ejemplo, la ISC, el NOI, el AIBVO, S, AIBVSO, la dirección de AIBV y la dirección de AISB.
- 15 Además de lo anterior, otra operación de control que se puede especificar es una operación de interrupciones de adaptador no registrado, un ejemplo de la cual se describe con referencia a la FIG. 10. Con esta operación, se reinician los parámetros de función de adaptador relevantes a la interrupción de adaptador.
- Con referencia a la FIG. 10, inicialmente, se toma una determinación en cuanto a si está registrado el adaptador especificado por el gestor de función para interrupciones, PREGUNTA 1000. Si no está, entonces, se proporciona
- 20 una condición de excepción, PASO 1002. De otro modo, los parámetros de interrupción en la entrada de tabla de función (u otra ubicación) y la entrada de tabla de dispositivo correspondiente se fijan a ceros, PREGUNTA 1004. En un ejemplo, estos parámetros incluyen la ISC, el NOI, AIBVO, S, AIBSO, la dirección AIBV y la dirección AISB.
- Como se describió anteriormente, en una realización, para obtener la información con respecto a una función de adaptador, se usa una instrucción Llamar al Procesador Lógico. Una realización de esta instrucción se representa en la FIG. 11A. Como se muestra, en un ejemplo, una instrucción Llamar al Procesador Lógico (CLP) 1100 incluye un código de operación 1102 que indica que es la instrucción Llamar al Procesador Lógico; y una indicación para un comando 1104. En un ejemplo, esta indicación es una dirección de un bloque de petición que describe el comando a ser realizado y la información en el bloque de petición es dependiente del comando. Ejemplos de bloques de peticiones y bloques de respuesta correspondientes para diversos comandos se describen con referencia a las FIG.
- 30 11B-13B.
- Con referencia inicialmente a la FIG. 11B, se proporciona un bloque de petición para un comando de funciones de PCI. El comando lista de funciones de PCI se usa para obtener una lista de las funciones de PCI que están asignadas a la configuración solicitante (por ejemplo, el sistema operativo solicitante). Un bloque de petición 1120 incluye un número de parámetros, tales como, por ejemplo:
- 35
- 40 Campo de longitud 1122: Este campo indica la longitud del bloque de petición;
- Código de Comando 1124: Este campo indica el comando lista de funciones de PCI; y
- Reanudar Testigo 1126: Este campo es un entero que se usa o bien para iniciar un comando nueva lista de funciones de PCI o reanudar un comando de lista previa de funciones de PCI, como se describe en más detalle más adelante.
- 45 Cuando el campo reanudar testigo en el bloque de petición de comando incluye, por ejemplo, un valor de cero, se solicita una nueva lista de funciones de PCI. Cuando el campo reanudar testigo incluye, por ejemplo, un valor no cero, que fue devuelto desde un comando de lista previa de funciones de PCI, se solicita una continuación de una lista previa de funciones de PCI.
- En respuesta a emitir y procesar la instrucción Llamar al Procesador Lógico para un comando de lista de funciones de PCI, se devuelve un bloque de respuesta. Una realización del bloque de respuesta se representa en la FIG. 11C. En un ejemplo, un bloque de respuesta 1150 para un comando de lista de funciones de PCI incluye:
- 50
- Campo de longitud 1152: Este campo indica la longitud del bloque de respuesta;
- Código de Respuesta 1154: Este campo indica un estado del comando;
- 55 Lista de Funciones de PCI 1156: Este campo indica una lista de una o más funciones de PCI disponibles para el sistema operativo solicitante;

- Reanudar Testigo 1158: Este campo indica si se requiere una continuación de una lista previa de funciones de PCI. En un ejemplo, cuando la reanudación del testigo en el bloque de petición y la reanudación del testigo en el bloque de respuesta son cero, todas las funciones de PCI asignadas a la configuración solicitante se representan en la lista de función de PCI; si la reanudación del testigo en el bloque de petición es cero y la reanudación del testigo en el bloque de respuesta no es cero, pueden existir funciones de PCI adicionales asignadas a la configuración de petición que no se han representado en la lista; si la reanudación del testigo en el bloque de respuesta es cero, desde el punto de la reanudación, se representan en la lista las funciones de PCI restantes asignadas a la configuración solicitante; cuando tanto la reanudación de testigos en el bloque de petición como en el de respuesta no son cero desde el punto de la reanudación, pueden existir funciones de PCI adicionales asignadas a la configuración solicitante que no se han representado en ninguna lista de funciones de PCI asociada. La reanudación de testigo permanece válida durante un periodo indefinido de tiempo después de ser devuelta, pero puede ser inválida debido a una variedad de razones dependientes del modelo, incluyendo el tiempo transcurrido de carga del sistema;
- Datos Dependientes del Modelo 1160: Este campo incluye datos que dependen del sistema;
- Número de Funciones de PCI 1162: Este campo indica el número máximo de funciones de PCI soportadas por la facilidad; y
- Tamaño de Entrada 1164: Este campo indica el tamaño de cada entrada en la lista de funciones de PCI.
- Detalles adicionales con respecto a la lista de funciones de PCI se describen con referencia a la FIG. 11D. En un ejemplo, la lista de funciones de PCI incluye una pluralidad de entradas y cada entrada 1156 incluye la siguiente información, como ejemplo:
- ID de Dispositivo 1170: Este campo indica el adaptador de I/O asociado con la función de PCI correspondiente;
- ID de Proveedor 1172: Este campo identifica el fabricante del adaptador de I/O asociado con la función de PCI correspondiente;
- Identificador de Función 1174: Este campo incluye un identificador persistente de la función de PCI;
- Gestor de Función 1176: Este campo identifica una función de PCI. El gestor de función de PCI almacenado es un gestor general cuando un bit especificado del gestor es cero y es un gestor habilitado cuando ese bit es uno. Si la función de PCI está deshabilitada, se almacena un gestor de función de PCI general. Si la función de PCI está habilitada, se almacena un gestor de función de PCI habilitado. Un gestor de función de PCI, en un ejemplo, no es persistente más allá de un IPL, que difiere del ID de función de PCI, que es persistente y se fija durante la vida de la definición de configuración de I/O; y
- Estado de Configuración 1178: Este campo indica el estado de la función de PCI. Cuando este indicador es, por ejemplo, cero, el estado está en espera y cuando, por ejemplo, es uno, el estado está configurado. Cuando está en espera, el gestor de función de PCI es el gestor de función de PCI general y cuando está configurado, es o bien el gestor de función de PCI general o bien habilitado dependiendo de si está habilitada la función de PCI.
- Posterior a obtener la lista de funciones de adaptador, se puede obtener información con respecto a los atributos de una función seleccionada que se designa por un gestor de función de PCI especificado. Esta información se puede obtener emitiendo una instrucción CLP con un comando consultar función.
- Una realización del bloque de petición para un comando consultar función de PCI se describe con referencia a la FIG. 12A. En un ejemplo, el bloque de petición 1200 incluye, por ejemplo:
- Campo de Longitud 1202: Este campo indica la longitud del bloque de petición;
- Código de Comando 1204: Este campo indica el comando consultar función de PCI; y
- Gestor de Función 1206: Este campo incluye el gestor de función de PCI (por ejemplo, general o habilitado) que designa la función de PCI a ser consultada.
- En respuesta a emitir la instrucción Llamar al Procesador Lógico para el comando consultar función de PCI, se devuelve un bloque de respuesta. Una realización del bloque de respuesta se representa en la FIG. 12B. En un ejemplo, un bloque de respuesta 1250 incluye lo siguiente:
- Longitud 1252: Este campo indica la longitud del bloque de respuesta;
- Código de Respuesta 1254: Este campo indica un estado del comando;
- ID de Grupo de Funciones 1256: Este campo indica el identificador de grupo de funciones de PCI. Un identificador de grupo de funciones de PCI se usa para asociar un grupo de funciones de PCI con un conjunto de

atributos (también conocidos en la presente memoria como características). Cada función de PCI con el mismo identificador de grupo de funciones de PCI tiene el mismo conjunto de atributos;

ID de Función 1258: Este Id de función de PCI es un identificador persistente de la función de PCI originalmente especificada por el gestor de función de PCI y se fija durante la vida de la definición de configuración de I/O;

5 Adaptador de Canal Físico 1260: Este valor representa una identificación dependiente del modelo de la ubicación del adaptador de I/O físico que corresponde a la función de PCI;

10 Registros de Direcciones Base (BAR) 1...n 1262: Este campo incluye una pluralidad de enteros sin signo, designados como $BAR_0 - BAR_n$, que están asociados con la función de PCI especificada originalmente y cuyos valores también se almacenan en los registros de direcciones base asociados con la función de PCI. Cada BAR especifica la dirección de inicio de un espacio de memoria o espacio de I/O dentro del adaptador y también indica el tipo de espacio de direcciones, que es si es un espacio de memoria de 64 o 32 bits o un espacio de I/O de 32 bits, como ejemplos;

15 Tamaño 1...n 1264: Este campo incluye una pluralidad de enteros sin signo, designados como $TAMAÑO_0 - TAMAÑO_n$. El valor de un campo Tamaño, cuando no es cero, representa el tamaño de cada espacio de direcciones con cada entrada que corresponde a un BAR descrito previamente;

Inicio de DMA Disponible 1266: Este campo incluye una dirección que indica el comienzo de un intervalo de direcciones de PCI que están disponibles para operaciones de DMA;

Fin de DMA Disponible 1268: Este campo incluye un valor que indica el fin de un intervalo de direcciones de PCI que están disponibles para operaciones de DMA.

20 Además de obtener atributos con respecto a la función de adaptador específico, también se pueden obtener atributos obtenidos con respecto al grupo que incluye esta función. Estos atributos comunes se pueden obtener a partir de emitir una instrucción CLP con un comando consultar grupo de funciones de PCI. Este comando se usa para obtener un conjunto de características que se soportan para un grupo de una o más funciones de PCI designadas por el identificador de grupo de funciones de PCI especificado. Un identificador de grupo de funciones de PCI se usa para asociar un grupo de funciones de PCI con el mismo conjunto de características. Una realización de un bloque de petición para el comando consultar grupo de funciones de PCI se describe con referencia a la FIG. 13A. En un ejemplo, el bloque de petición 1300 incluye lo siguiente:

Campo de Longitud 1302: Este campo indica la longitud del bloque de petición;

Código de Comando 1304: Este campo indica el comando consultar grupo de funciones de PCI; e

30 ID de Grupo de Funciones 1306: Este campo especifica el identificador del grupo de funciones de PCI para el cual van a ser obtenidos atributos.

En respuesta a emitir y procesar la instrucción Llamar al Procesador Lógico con un comando consultar grupo de funciones de PCI, se devuelve un bloque de respuesta. Una realización del bloque de respuesta se representa en la FIG. 13B. En un ejemplo, un bloque de respuesta 1350 incluye:

35 Campo de Longitud 1352: Este campo indica la longitud del bloque de respuesta;

Código de Respuesta 1354: Este campo indica un estado del comando;

40 Número de Interrupciones 1356: Este campo indica el número máximo de números de vectores de MSI consecutivos (es decir, indicadores de evento de interrupción) que están soportados por la facilidad de PCI para cada función de PCI en el grupo de funciones de PCI especificado. Los valores válidos posibles del número de interrupciones están en el intervalo de cero a 2.048, en un ejemplo;

Versión 1358: Este campo indica la versión de la especificación de PCI que se soporta por la facilidad de PCI a la que está unido el grupo de funciones de PCI designado por el identificador del grupo de PCI especificado;

Trama 1362: Este campo indica los tamaños de trama (o página) soportados para la traducción de direcciones de I/O;

45 Intervalo de Actualización de Bloque de Medición 1364: Este es un valor que indica el intervalo de tiempo aproximado (por ejemplo, en milisegundos) en el que está actualizando el bloque de medición de la función de PCI;

Máscara de Espacio de Direcciones de DMA 1366: Este es un valor usado para indicar qué bits en una dirección de PCI se usan para identificar un espacio de direcciones de DMA; y

Dirección de MSI 1368: Este es un valor que va a ser usado para peticiones de interrupción de señal por mensaje.

5 Los comandos consultar lista y función descritos anteriormente recuperan información desde, por ejemplo, la tabla de funciones. En el momento de la inicialización o después de una conexión en caliente de un adaptador, el microprograma realiza un paseo por el canal principal para determinar la ubicación del adaptador y determina sus características básicas. Esta información se almacena por el microprograma en la entrada de tabla de función (FTE) para cada adaptador. La accesibilidad al adaptador se determina en base a la política fijada por un administrador del sistema y también se fija por el microprograma en la FTE. Los comandos consultar lista y función pueden recuperar entonces esta información y almacenarla en sus bloques de respuesta respectivos accesible para el sistema operativo.

10 Además, la información del grupo se basa en una infraestructura de I/O del sistema dada y las capacidades del microprograma y el centro de I/O. Esta se puede almacenar en la FTE o en cualquier otra ubicación conveniente para recuperación posterior durante el proceso de consulta. En particular, el comando consultar el grupo recupera la información y la almacena en su bloque de respuesta accesible para el sistema operativo.

15 Descrito en detalle anteriormente está una capacidad para convertir una interrupción de señal por mensaje de PCI en una notificación de evento de adaptador de I/O para un sistema operativo. Esto proporciona una petición de interrupción de baja latencia; una entrega de MSI a partir de un número relativamente grande de funciones de PCI al sistema operativo; y la retención del aroma de la designación de vector de MSI que encaja la MSI en la arquitectura de notificación de evento de adaptador. Se adapta para permitir al centro de I/O conectar con un número relativamente grande de funciones de PCI y elimina el problema de cada escritura a un vector de MSI que genera una interrupción única.

25 En las realizaciones descritas en la presente memoria, los adaptadores son adaptadores de PCI. PCI, como se usa en la presente memoria, se refiere a cualquier adaptador implementado según una especificación basada en PCI como se define por el Grupo de Interés Especial de Interconexión de Componentes Periféricos (PCI-SIG) (www.pcisig.com/home), que incluye pero no se limita a, PCI o PCIe. En un ejemplo particular, la Interconexión de Componentes Periféricos Rápida (PCIe) es un estándar de interconexión a nivel de componente que define un protocolo de comunicación bidireccional para transacciones entre sistemas de ordenador principal y adaptadores de I/O. Las comunicaciones PCIe se encapsulan en paquetes según el estándar PCIe para transmisión en un canal principal de PCIe. Las transacciones que se originan en los adaptadores de I/O y que terminan en sistemas de ordenador principal se conocen como transacciones de límite superior. Las transacciones que se originan en sistemas de ordenadores principales y que terminan en adaptadores de I/O se conocen como transacciones de límite inferior. La topología PCIe se basa en enlaces unidireccionales punto a punto que se emparejan (por ejemplo, un enlace de límite superior, un enlace de límite inferior) para formar el canal principal PCIe. El estándar PCIe se mantiene y publica por el PCI-SIG.

35 Como se apreciará por un experto en la técnica, aspectos de la presente invención se pueden realizar como un sistema, método o producto de programa de ordenador. Por consiguiente, aspectos de la presente invención pueden tomar la forma de una realización enteramente hardware, una realización enteramente software (incluyendo microprograma, software residente, microcódigo, etc.) o una realización que combina aspectos software y hardware que se pueden conocer todos de manera general en la presente memoria como un "circuito", "módulo" o "sistema". Por otra parte, aspectos de la presente invención pueden tomar la forma de un producto de programa de ordenador incorporado en uno o más medio(s) legible(s) por ordenador que tiene(n) código de programa legible por ordenador incorporado en el mismo.

45 Se puede utilizar cualquier combinación de uno o más medio(s) legible(s) por ordenador. El medio legible por ordenador puede ser un medio de almacenamiento legible por ordenador. Un medio de almacenamiento legible por ordenador puede ser, por ejemplo, pero no limitado a, un sistema, aparato o dispositivo electrónico, magnético, óptico, electromagnético, de infrarrojos o de semiconductores o cualquier combinación adecuada de los precedentes. Ejemplos más específicos (una lista no exhaustiva) del medio de almacenamiento legible por ordenador incluyen los siguientes: una conexión eléctrica que tiene uno o más hilos, un disquete de ordenador portátil, un disco duro, una memoria de acceso aleatorio (RAM), una memoria de solo lectura (ROM), una memoria de solo lectura programable borrrable (EPROM o memoria rápida), 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 precedentes. En el contexto de este documento, un medio de almacenamiento legible por ordenador puede ser cualquier medio tangible que puede contener o almacenar un programa para uso por o en conexión con un sistema, aparato o dispositivo de ejecución de instrucciones.

55 Con referencia ahora a la FIG. 14, en un ejemplo, un producto de programa de ordenador 1400 incluye, por ejemplo, uno o más medios de almacenamiento legibles por ordenador 1402 para almacenar medios o lógica de código de programa legible por ordenador 1404 en el mismo para proporcionar y facilitar uno o más aspectos de la presente invención.

El código de programa incorporado en un medio legible por ordenador se puede transmitir usando un medio adecuado, incluyendo pero no limitado a inalámbrico, línea cableada, cable de fibra óptica, RF, etc. o cualquier combinación adecuada de los precedentes.

5 El código de programa de ordenador para llevar a cabo operaciones para aspectos de la presente invención se puede escribir en cualquier combinación de uno o más lenguajes de programación, incluyendo un lenguaje de programación orientado a objetos, tal como Java, Smalltalk, C++ o similares y lenguajes de programación de proceso convencional, tales como el lenguaje de programación "C", ensamblador o lenguajes de programación similares. El código de programación puede ejecutarse enteramente en el ordenador del usuario, parcialmente en el ordenador del usuario, como un paquete software autónomo, parcialmente en el ordenador del usuario y
10 parcialmente en un ordenador remoto o enteramente en el ordenador o servidor remoto. En este último escenario, el ordenador remoto se puede conectar al ordenador del usuario a través de cualquier tipo de red, incluyendo una red de área local (LAN) o una red de área extensa (WAN) o la conexión se puede hacer a un ordenador externo (por ejemplo, a través de Internet usando un Proveedor de Servicios de Internet).

15 Aspectos de la presente invención se describen en la presente memoria con referencia a las ilustraciones del diagrama de flujo y/o diagramas de bloques de métodos, aparatos (sistemas) y productos de programa de ordenador según realizaciones de la invención. Se entenderá que cada bloque de las ilustraciones de diagrama de flujo y/o diagramas de bloques y combinaciones de bloques en las ilustraciones de diagrama de flujo y/o diagramas de bloques, se pueden implementar por instrucciones de programa de ordenador. Estas instrucciones de programa de ordenador se pueden proporcionar a un procesador de un ordenador de propósito general, ordenador de propósito
20 especial u 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, crean medios para implementar las funciones/actos especificados en el bloque o bloques de diagrama de flujo y/o diagrama de bloques.

25 Estas instrucciones de programa de ordenador también se pueden almacenar en un medio legible por ordenador que puede dirigir un ordenador, otro aparato de procesamiento de datos programable u otros dispositivos para funcionar de una manera particular, de manera que las instrucciones almacenadas en el medio legible por ordenador producen un artículo de fabricación que incluye instrucciones que implementan la función/acto especificado en el bloque o bloques del diagrama de flujo y/o diagrama de bloques.

30 Las instrucciones de programa de ordenador también se pueden cargar en un ordenador, otro aparato de procesamiento de datos programable u otros dispositivos para hacer que una serie de pasos de operación sean realizados en el ordenador, otro aparato programable u otros dispositivos para producir un proceso implementado por ordenador de manera que las instrucciones que se ejecutan en el ordenador u otro aparato programable proporcionan procesos para implementar las funciones/actos especificados en el bloque o bloques del diagrama de flujo y/o diagrama de bloques.

35 El diagrama de flujo y los diagramas de bloques en las figuras ilustran la arquitectura, funcionalidad y operación de implementaciones posibles de sistemas, métodos y productos de programa de ordenador según diversas realizaciones de la presente invención. A este respecto, cada bloque en el diagrama de flujo o diagramas de bloques puede representar un módulo, segmento o porción de código, que comprende una o más instrucciones ejecutables para implementar la(s) función(funciones) lógica(s) especificada(s). Se debería señalar también que, en algunas
40 implementaciones alternativas, las funciones señaladas en el bloque pueden ocurrir fuera del orden señalado en las figuras. Por ejemplo, dos bloques mostrados en sucesión se pueden, de hecho, ejecutar sustancialmente concurrentemente o los bloques se puede ejecutar algunas veces en el orden inverso, dependiendo de la funcionalidad implicada. También se señalará que cada bloque de los diagramas de bloques y/o ilustración de diagrama de flujo y combinaciones de bloques en los diagramas de bloques y/o ilustración de diagrama de flujo, se
45 pueden implementar por sistemas basados en hardware de propósito especial que realizan las funciones o actos especificados o combinaciones de hardware de propósito especial e instrucciones de ordenador.

Además de lo anterior, se pueden proporcionar, ofrecer, desplegar, gestionar, servir, etc. uno o más aspectos de la presente invención por un proveedor de servicios que ofrece gestión de entornos de cliente. Por ejemplo, el proveedor de servicios puede crear, mantener, soportar, etc. el código informático y/o la infraestructura informática que realiza uno o más aspectos de la presente invención para uno o más clientes. A cambio, el proveedor de servicios puede recibir un pago del cliente bajo una suscripción y/o acuerdo de pago, como ejemplos. Adicional o
50 alternativamente, el proveedor de servicios puede recibir un pago a partir de la venta de contenido de publicidad de una o más terceras partes.

55 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 infraestructura informática operable para realizar uno o más aspectos de la presente invención.

Como un aspecto adicional de la presente invención, se puede desplegar una infraestructura informática que comprende integrar código legible por ordenador en un sistema informático, en el que el código en combinación con el sistema informático es capaz de realizar uno o más aspectos de la presente invención.

Aún como un aspecto adicional de la presente invención, se puede proporcionar un proceso para integrar infraestructura informática que comprende integrar código legible por ordenador en un sistema informático. El sistema informático comprende un medio legible por ordenador, en el que el medio legible por ordenador comprende uno o más aspectos de la presente invención. El código en combinación con el sistema informático es capaz de realizar uno o más aspectos de la presente invención.

Aunque se describieron anteriormente diversas realizaciones, estas son solamente ejemplos. Por ejemplo, entornos informáticos de otras arquitecturas pueden incorporar y usar uno o más aspectos de la presente invención. Como ejemplos, servidores distintos de servidores System z[®], tales como servidores de Power Systems u otros servidores ofrecidos por International Business Machines Corporation o servidores de otras compañías pueden incluir, usar y/o beneficiarse de uno o más aspectos de la presente invención. Además, aunque en el ejemplo en la presente memoria, los adaptadores y centro de PCI se consideran una parte del servidor, en otras realizaciones, no tienen que ser necesariamente considerados una parte del servidor, sino que pueden ser considerados simplemente como que están acoplados a la memoria del sistema y/u otros componentes de un entorno informático. El entorno informático no necesita ser un servidor. Además, aunque los adaptadores están basados en PCI, uno o más aspectos de la presente memoria son utilizables con otros adaptadores y otros componentes de I/O. El adaptador y el adaptador PCI son sólo ejemplos. Además, uno o más aspectos de la presente invención son aplicables a esquemas de interrupción distintos de MSI de PCI. Aún más, aunque se describen ejemplos en los cuales se fijan bits, en otras realizaciones, se pueden fijar bytes u otro tipo de indicadores. Además, la DTE puede incluir más, menos o diferente información. Son posibles muchas otras variantes.

Además, otros tipos de entornos informáticos pueden beneficiarse de uno o más aspectos de la presente invención. Como ejemplo, un sistema de procesamiento de datos adecuado para almacenar y/o ejecutar código de programa es utilizable que incluye al menos dos procesadores acoplados directa o indirectamente a elementos de memoria a través de un canal principal del sistema. Los elementos de memoria incluyen, por ejemplo, una memoria local empleada durante la ejecución real del código de programa, almacenamiento masivo y memoria caché que proporcionan almacenamiento temporal de al menos algún código de programa a fin de reducir el número de veces que se debe recuperar el código desde el almacenamiento masivo durante la ejecución.

Los dispositivos de Entrada/Salida o I/O (incluyendo, pero no limitados a, teclados, visualizadores, dispositivos de apuntamiento, DASD, cinta, CD, DVD, memorias USB y otros medios de memoria, etc.) se pueden acoplar al sistema o bien directamente o bien a través de controladores de I/O intervinientes. Los adaptadores de red también se pueden acoplar al sistema para permitir al sistema de procesamiento de datos llegar a ser acoplado a otros sistemas de procesamiento de datos o impresoras remotas o dispositivos de almacenamiento a través de redes privadas o públicas intervinientes. Módem, cable módem y tarjetas Ethernet son sólo unos pocos de los tipos disponibles de adaptadores de red.

Con referencia a la FIG. 15, se representan componentes representativos de un sistema de Ordenador Principal 5000 para implementar uno o más aspectos de la presente invención. El ordenador principal 5000 representativo comprende una o más CPU 5001 en comunicación con la memoria del ordenador (es decir, almacenamiento central) 5002, así como interfaces de I/O a dispositivos de medios de almacenamiento 5011 y redes 5010 para comunicar con otros ordenadores o SAN y similares. La CPU 5001 es compatible con una arquitectura que tiene un conjunto de instrucciones de arquitectura y funcionalidad de arquitectura. La CPU 5001 puede tener traducción de direcciones dinámica (DAT) 5003 para transformar direcciones de programa (direcciones virtuales) en direcciones reales de memoria. Una DAT típicamente incluye un almacenador temporal de traducción anticipada (TLB) 5007 para capturar traducciones de manera que los accesos posteriores al bloque de memoria de ordenador 5002 no requieren el retardo de traducción de direcciones. Típicamente, se emplea una caché 5009 entre la memoria de ordenador 5002 y el procesador 5001. La caché 5009 puede ser jerárquica teniendo una caché grande disponible para más de una CPU y cachés más pequeñas, más rápidas (menor nivel) entre la caché grande y cada CPU. En algunas implementaciones, las cachés de menor nivel se dividen para proporcionar cachés de nivel bajo separadas para traer instrucciones y accesos de datos. En una realización, una instrucción se trae desde la memoria 5002 por una unidad de extracción de instrucciones 5004 a través de una caché 5009. La instrucción se decodifica en una unidad de decodificación de instrucciones 5006 y despacha (con otras instrucciones en algunas realizaciones) a la unidad o unidades de ejecución de instrucciones 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 instrucciones de ramal. La instrucción se ejecuta por la unidad de ejecución, accediendo a los operandos a partir de la memoria o los registros especificados de instrucción según se necesite. Si se va a acceder (cargar o almacenar) a un operando desde la memoria 5002, una unidad de carga/almacenamiento 5005 típicamente maneja el acceso bajo el control de la instrucción que se ejecuta. Las instrucciones se pueden ejecutar en circuitos hardware o en microcódigo (microprograma) interno o por una combinación de ambos.

Como se señala, un sistema informático incluye información en almacenamiento local (o principal), así como direccionamiento, protección y registro de referencia y cambio. Algunos aspectos de direccionamiento incluyen el formato de direcciones, el concepto de espacios de direcciones, los diversos tipos de direcciones y la manera en la que un tipo de dirección se traduce a otro tipo de dirección. Algo del almacenamiento principal incluye ubicaciones de almacenamiento asignadas permanentemente. El almacenamiento principal dota al sistema con almacenamiento

de datos de acceso rápido directamente direccionable. Tanto los datos como los programas van a ser cargados en el almacenamiento principal (desde los dispositivos de entrada) antes de que se puedan procesar.

El almacenamiento principal puede incluir uno o más almacenamientos de almacenador temporal menores, de acceso más rápido, algunas veces llamados cachés. Una caché está asociada típicamente de manera física con una CPU o un procesador de I/O. Los efectos, excepto sobre el rendimiento, de la construcción física y uso de medios de almacenamiento distintos no son generalmente observables por el programa.

Se pueden mantener cachés separadas para instrucciones y para operandos de datos. La información dentro una caché se mantiene en bytes contiguos en un límite integral llamado bloque de caché o línea de caché (o línea, para abreviar). Un modelo puede proporcionar una instrucción EXTRAER ATRIBUTO DE CACHÉ que devuelve el tamaño de una línea de caché en bytes. Un modelo también puede proporcionar instrucciones TRAER PREVIAMENTE DATOS y TRAER PREVIAMENTE DATOS RELATIVAMENTE LARGOS que efectúa la obtención previa de almacenamiento en la caché de datos o instrucción o la liberación de datos desde la caché.

El almacenamiento se ve como una cadena horizontal larga de bits. Para la mayoría de las operaciones, los accesos al almacenamiento proceden en una secuencia de izquierda a derecha. La cadena de bits se subdivide en unidades de ocho bits. Una unidad de ocho bits se llama byte, que es el bloque de construcción básico de todos los formatos de información. Cada ubicación de bytes en el almacenamiento se identifica por un entero no negativo único, que es la dirección de esa ubicación de byte o, simplemente, la dirección de byte. Las ubicaciones de bytes adyacentes tienen direcciones consecutivas, que comienzan con 0 en la izquierda y que proceden en una secuencia de izquierda a derecha. Las direcciones son enteros binarios sin signo y son de 24, 31 o 64 bits.

La información se transmite entre el almacenamiento y una CPU o un subsistema de canal de un byte o un grupo de bytes, a la vez. A menos que se especifique de otro modo, en, por ejemplo, la z/Architecture[®], un grupo de bytes en almacenamiento se direcciona por el byte de más a la izquierda del grupo. El número de bytes en el grupo está o bien implícito o bien especificado explícitamente por la operación a ser realizada. Cuando se usa en una operación de CPU, un grupo de bytes se llama campo. Dentro de cada grupo de bytes, en, por ejemplo, la z/Architecture[®], los bits se numeran en una secuencia de izquierda a derecha. En la z/Architecture[®], los bits de más a la izquierda se conocen algunas veces como los bits de "orden alto" y los bits de más a la derecha como los bits de "orden bajo". Los números de bits no son direcciones de almacenamiento, no obstante. Solamente se pueden direccionar los bytes. Para operar sobre bits individuales de un byte en almacenamiento, se accede al byte entero. Los bits en un byte se numeran 0 hasta 7, de izquierda a derecha (en, por ejemplo, la z/Architecture[®]). Los bits en una dirección se pueden numerar 8-31 o 40-63 para direcciones de 24 bits o 1-31 o 33-63 para direcciones de 31 bits; se numeran 0-63 para direcciones de 64 bits. Dentro de cualquier otro formato de longitud fija de múltiples bytes, los bits que componen el formato se numeran consecutivamente comenzando desde 0. Para propósitos de detección de errores y preferiblemente para corrección, se pueden transmitir uno o más bits de comprobación con cada byte o con un grupo de bytes. Tales bits de comprobación se generan automáticamente por la máquina y no pueden ser controlados directamente por el programa. Las capacidades de almacenamiento se expresan en número de bytes. Cuando la longitud de un campo de almacenamiento de operando está implícita por el código de operación de una instrucción, el campo se dice que tiene una longitud fija, que puede ser uno, dos, cuatro, ocho o dieciséis bytes. Se pueden implicar campos mayores para algunas instrucciones. Cuando la longitud de un campo de almacenamiento de operando no está implícito pero se indica explícitamente, el campo se dice que tiene una longitud variable. Los operandos de longitud variable pueden variar en longitud en incrementos de un byte (o con algunas instrucciones, en múltiplos de dos bytes u otros múltiplos). Cuando la información se coloca en el almacenamiento, se sustituyen los contenidos solamente de esas ubicaciones de byte que están incluidas en el campo designado, incluso aunque la anchura del camino físico al almacenamiento pueda ser mayor que la longitud del campo que se almacena.

Ciertas unidades de información van a estar en un límite integral en el almacenamiento. Un límite se llama 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 campos de 2, 4, 8 y 16 bytes en un límite integral. Una media palabra es un grupo de dos bytes consecutivos en un límite de dos bytes y es el bloque de construcción básico de las instrucciones. Una palabra es un grupo de cuatro bytes consecutivos en un límite de cuatro bytes. Una palabra doble es un grupo de ocho bytes consecutivos en un límite de ocho bytes. Una palabra cuádruple es un grupo de 16 bytes consecutivos en un límite de 16 bytes. Cuando las direcciones de almacenamiento designan medias palabras, palabras, palabras dobles y palabras cuádruples, la representación binaria de la dirección contiene uno, dos, tres o cuatro bits cero de más a la derecha, respectivamente. Las instrucciones van a estar en límites integrales de dos bytes. Los operandos de almacenamiento de la mayoría de instrucciones no tienen requisitos de límite de alineamiento.

En dispositivos que implementan cachés separadas para instrucciones y operandos de datos, se puede experimentar un retardo significativo si el programa almacena una línea de caché desde la cual se traen posteriormente instrucciones, con independencia de si el almacenamiento altera las instrucciones que se traen posteriormente.

En una realización, la invención se puede poner en práctica por software (algunas veces conocido como código interno con licencia, microprograma, microcódigo, milicódigo, picocódigo y similares, cualquiera de los cuales sería

coherente con la presente invención). Con referencia a la FIG. 15, se accede típicamente a un código de programa software que incorpora la presente invención por el procesador 5001 del sistema de ordenador principal 5000 desde dispositivos de medios de almacenamiento a largo plazo 5011, tales como una unidad de CD-ROM, unidad de cinta o unidad de disco duro. El código de programa software se puede incorporar en cualquiera de una variedad de medios conocidos para uso con un sistema de procesamiento de datos, tal como un disquete, disco duro o CD-ROM. El código se puede distribuir en tales medios o se puede distribuir a usuarios desde la memoria del ordenador 5002 o almacenamiento de un sistema informático sobre una red 5010 a otros sistemas informáticos para uso por los usuarios de tales otros sistemas.

El código de programa software incluye un sistema operativo que controla la función e interacción de los diversos componentes informáticos y uno o más programas de aplicaciones. El código de programa se pagina normalmente desde el dispositivo de medios de almacenamiento 5011 al almacenamiento informático de velocidad relativamente más alta 5002 donde está disponible para procesamiento por el procesador 5001. Las técnicas y métodos para incorporar código de programa software en memoria, en medios físicos y/o distribuir código software a través de redes son bien conocidos y no se tratarán más en la presente memoria. El código de programa, cuando se crea y almacena en un medio tangible (que incluye pero no se limita a módulos de memoria electrónicos (RAM), memoria rápida, Discos Compactos (CD), DVD, Cinta Magnética y similares se conoce a menudo como un "producto de programa de ordenador". El medio de producto de programa de ordenador es típicamente legible por un circuito de procesamiento preferiblemente un sistema informático para ejecución por el circuito de procesamiento.

La FIG. 16 ilustra un sistema hardware de estación de trabajo o servidor representativo en el cual se puede poner en práctica la presente invención. El sistema 5020 de la FIG. 16 comprende un sistema informático base representativo 5021, tal como un ordenador personal, una estación de trabajo o un servidor, incluyendo dispositivos periféricos opcionales. El sistema informático base 5021, incluye uno o más procesadores 5026 y un canal principal empleado para conectar y permitir comunicación entre el(los) procesador(es) 5026 y los otros componentes del sistema 5021 según técnicas conocidas. El canal principal conecta el procesador 5026 a la memoria 5025 y el almacenamiento de largo plazo 5027 que puede incluir un disco duro (incluyendo cualquier medio magnético, CD, DVD y Memoria Rápida por ejemplo) o una unidad de cinta por ejemplo. El sistema 5021 también podría incluir un adaptador de interfaz de usuario, que conecta el microprocesador 5026 a través del canal principal a uno o más dispositivos de interfaz, tales como un teclado 5024, un ratón 5023, una impresora/escáner 5030 y/u otros dispositivos de interfaz, que pueden ser cualquier dispositivo de interfaz de usuario, tales como una pantalla sensible al tacto, almohadilla de entrada digitalizada, etc. El canal principal también conecta un dispositivo de visualización 5022, tal como una pantalla o monitor LCD, al microprocesador 5026 a través de un adaptador de visualización.

El sistema 5021 puede comunicar con otros ordenadores o redes de ordenadores por medio de un adaptador de red capaz de comunicar 5028 con una red 5029. Adaptadores de red ejemplo son canales de comunicaciones, token ring, Ethernet o módem. Alternativamente, el sistema 5021 puede comunicar usando una interfaz inalámbrica, tal como una tarjeta CDPC (datos por paquetes digitales celulares). El sistema 5021 se puede asociar con tales otros ordenadores en una Red de Área Local (LAN) o una Red de Área Extensa (WAN) o el sistema 5021 puede ser un cliente en una disposición cliente/servidor con otro ordenador, etc. Todas estas configuraciones, así como el hardware y software de comunicaciones adecuado, son conocidas en la técnica.

La FIG. 17 ilustra una red de procesamiento de datos 5040 en la que se puede poner en 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 cableada, cada una de las cuales puede incluir una pluralidad de estaciones de trabajo individuales 5041, 5042, 5043, 5044. Adicionalmente, como los expertos en la técnica apreciarán, se pueden incluir una o más LAN, donde una LAN puede comprender una pluralidad de estaciones de trabajo inteligentes acopladas a un procesador del ordenador principal.

Aún con referencia a la FIG. 17, las redes también pueden incluir servidores u ordenadores centrales, tales como un ordenador pasarela (cliente servidor 5046) o servidor de aplicaciones (servidor remoto 5048 que puede acceder a un repositorio de datos y también se puede acceder directamente desde una estación de trabajo 5045). Un ordenador pasarela 5046 sirve como un punto de entrada en cada red individual. Se necesita una pasarela cuando se conecta un protocolo de red a otro. La pasarela 5046 se puede acoplar preferiblemente a otra red (Internet 5047 por ejemplo) por medio de un enlace de comunicaciones. La pasarela 5046 también se puede acoplar directamente a una o más estaciones de trabajo 5041, 5042, 5043, 5044 usando un enlace de comunicaciones. El ordenador pasarela se puede implementar utilizando un servidor eServer™ System z® IBM disponible en International Business Machines Corporation.

Con referencia concurrentemente a la FIG. 16 y la FIG. 17, se puede acceder a código de programación software que puede incorporar la presente invención mediante el procesador 5026 del sistema 5020 desde medios de almacenamiento a largo plazo 5027, tales como una unidad de CD-ROM o de disco duro. El código de programación software se puede incorporar en cualquiera de una variedad de medios conocidos para uso con un sistema de proceso de datos, tal como un disquete, un disco duro o CD-ROM. El código se puede distribuir en tales medios o se puede distribuir a los usuarios 5050, 5051 desde la memoria o almacenamiento de un sistema informático sobre una red para otros sistemas informáticos para uso por usuarios de tales otros sistemas.

Alternativamente, el código de programación se puede incorporar en la memoria 5025 y acceder por el procesador 5026 usando el canal principal del procesador. Tal código de programación incluye un sistema operativo que controla la función e interacción de los diversos componentes informáticos y uno o más programas de aplicaciones 5032. El código de programa normalmente es paginado desde los medios de almacenamiento 5027 a la memoria de alta velocidad 5025 donde está disponible para procesamiento por el procesador 5026. Las técnicas y métodos para incorporar un código de programación software en memoria, en medios físicos y/o distribuir un código software a través de redes son bien conocidos y no se tratarán más en la presente memoria. El código de programa, cuando se crea y almacena en un medio tangible (incluyendo pero no limitado a módulos de memoria electrónicos (RAM), memoria rápida, Discos Compactos (CD), DVD, Cinta Magnética y similares a menudo se conoce como un “producto de programa de ordenador”. El medio de producto de programa de ordenador típicamente es legible por un circuito de procesamiento preferiblemente en un sistema informático para ejecución por el circuito de procesamiento.

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

Con referencia a la FIG. 18, una realización de procesador ejemplar se representa para el procesador 5026. Típicamente se emplean uno o más niveles de caché 5053 para almacenar temporalmente bloques de memoria a fin de mejorar el rendimiento del procesador. La caché 5053 es un almacenador temporal de alta velocidad que contiene líneas de caché de datos de memoria que probablemente van a ser usados. Las líneas de caché típicas son de 64, 128 o 256 bytes de datos de memoria. Las cachés separadas se emplean a menudo para almacenar en caché instrucciones distintas de datos de caché. La coherencia de caché (sincronización de copias de líneas en memoria y las cachés) se proporciona a menudo por diversos algoritmos de “fisgoneo” bien conocidos en la técnica. El almacenamiento de memoria principal 5025 de un sistema procesador se conoce a menudo como una caché. En un sistema procesador que tiene 4 niveles de caché 5053, el almacenamiento principal 5025 algunas veces se conoce como la caché de nivel 5 (L5) dado que es típicamente más rápido y solamente contiene una porción del almacenamiento no volátil (DASD, cinta, etc.) que está disponible para un sistema informático. El almacenamiento principal 5025 “almacena en caché” páginas de datos paginadas dentro y fuera del almacenamiento principal 5025 por el sistema operativo.

Un contador de programa (contador de instrucciones) 5061 hace el seguimiento de la dirección de la instrucción actual a ser ejecutada. Un contador de programa en un procesador de z/Architecture[®] es de 64 bits y se puede trunca a 31 o 24 bits para soportar límites de direccionamiento anteriores. Un contador de programa se incorpora típicamente en una PSW (palabra de estado de programa) de un ordenador de manera que persiste durante la conmutación de contexto. De esta manera, un programa en curso, que tiene un valor de contador de programa, se puede interrumpir, por ejemplo, por el sistema operativo (conmutación de contexto desde el entorno de programa al entorno de sistema operativo). La PSW del programa contiene el valor de contador de programa mientras que el programa no está activo y el contador de programa (en la PSW) del sistema operativo se usa mientras que está ejecutándose el sistema operativo. Típicamente, el contador de programa se aumenta en una cantidad igual al número de bytes de la instrucción actual. Las instrucciones RISC (Cálculo con Juego de Instrucciones Reducido) son típicamente de longitud fija mientras que las instrucciones CISC (Cálculo con Juego de Instrucciones Complejo) son típicamente de longitud variable. Las instrucciones de la z/Architecture[®] de IBM son instrucciones CISC que tienen una longitud de 2, 4 o 6 bytes. El contador de Programa 5061 se modifica o bien por una operación de conmutación de contexto o bien por una operación de toma de ramal de una instrucción de ramal por ejemplo. En una operación de conmutación de contexto, el valor de contador de programa actual se guarda en la palabra de estado del programa junto con otra información de estado acerca del programa que se ejecuta (tal como códigos de condición) y un nuevo valor de contador de programa se carga apuntando a una instrucción de un nuevo módulo de programa a ser ejecutado. Una operación de toma de ramal se realiza a fin de permitir al programa tomar decisiones o hacer un bucle dentro del programa cargando el resultado de la instrucción de ramal en el contador de programa 5061.

Típicamente una unidad de extracción de instrucciones 5055 se emplea para traer instrucciones en nombre del procesador 5026. La unidad de extracción o bien trae las “siguientes instrucciones secuenciales”, las instrucciones objetivo de instrucciones de toma de ramal o bien las primeras instrucciones de un programa que siguen una conmutación de contexto. Las modernas unidades de extracción de Instrucciones a menudo emplean técnicas de extracción previa para traer previamente especulativamente instrucciones en base a la probabilidad de que se pudieran usar las instrucciones traídas previamente. Por ejemplo, una unidad de extracción puede traer 16 bytes de instrucción que incluye la siguiente instrucción secuencial y bytes adicionales de instrucciones secuenciales adicionales.

Las instrucciones traídas entonces se ejecutan por el procesador 5026. En una realización, la(s) instrucción(es) traída(s) se pasan a una unidad de despacho 5056 de la unidad de extracción. La unidad de despacho decodifica la(s) instrucción(es) y reenvía información acerca de la(s) instrucción(es) decodificada(s) a las unidades adecuadas 5057, 5058, 5060. Una unidad de ejecución 5057 recibirá típicamente información acerca de las instrucciones aritméticas decodificadas desde la unidad de

extracción de instrucciones 5055 y realizará operaciones aritméticas sobre operandos según el código de operación de la instrucción. Los operandos se proporcionan a la unidad de ejecución 5057 preferiblemente o bien desde la memoria 5025, los registros de arquitectura 5059 o bien desde un campo inmediato de la instrucción que se ejecuta. Los resultados de la ejecución, cuando se almacenan, se almacenan o bien en la memoria 5025, los registros 5059 o bien en otro hardware de máquina (tal como registros de control, registros de PSW y similares).

Un procesador 5026 típicamente tiene una o más unidades 5057, 5058, 5060 para ejecutar la función de la instrucción. Con referencia a la FIG. 19A, una unidad de ejecución 5057 puede comunicar con registros generales de arquitectura 5059, una unidad de decodificación/despacho 5056, una unidad de almacenamiento de carga 5060 y otras unidades procesadoras 5065 por medio una lógica de interfaz 5071. Una unidad de ejecución 5057 puede emplear varios circuitos de registro 5067, 5068, 5069 para mantener la información sobre la que operará la unidad de lógica aritmética (ALU) 5066. La ALU realiza operaciones aritméticas tales como sumar, restar, multiplicar y dividir así como funciones lógicas tales como y (and), o (or) y o-exclusiva (XOR), rotar y desplazar. Preferiblemente la ALU soporta operaciones especializadas que son dependientes del diseño. Otros circuitos pueden proporcionar otras facilidades de arquitectura 5072 incluyendo códigos de condición y recuperación de lógica de soporte por ejemplo. Típicamente el resultado de una operación de ALU se mantiene en un circuito de registro de salida 5070 que puede reenviar el resultado a una variedad de otras funciones de procesamiento. Hay muchas disposiciones de unidades procesadoras, la presente descripción solamente se destina a proporcionar una comprensión representativa de una realización.

Una instrucción SUMAR por ejemplo se ejecutaría en una unidad de ejecución 5057 que tiene funcionalidad aritmética y lógica mientras que una instrucción de punto flotante por ejemplo se ejecutaría en una ejecución de punto flotante que tiene capacidad de punto flotante especializada. Preferiblemente, una unidad de ejecución opera sobre operandos identificados por una instrucción realizando una función definida de código de operación sobre los operandos. Por ejemplo, una instrucción SUMAR se puede ejecutar por una unidad de ejecución 5057 sobre operandos encontrados en dos registros 5059 identificados por campos de registro de la instrucción.

La unidad de ejecución 5057 realiza la suma aritmética sobre dos operandos y almacena el resultado en un tercer operando donde el tercer operando puede ser un tercer registro o uno de los dos registros fuente. La unidad de ejecución utiliza preferiblemente una Unidad de Lógica Aritmética (ALU) 5066 que es capaz de realizar una variedad de funciones lógicas tales como Desplazar, Rotar, Y, O y XOR así como una variedad de funciones algebraicas incluyendo cualquiera de sumar, restar, multiplicar, dividir. Algunas ALU 5066 se diseñan para operaciones escalares y algunas para punto flotante. Los datos pueden ser Big Endian (donde el byte menos significativo está en la dirección de byte más alta) o Little Endian (donde el byte menos significativo está en la dirección de byte más baja) dependiendo de la arquitectura. La z/Architecture[®] de IBM es Big Endian. Los campos con signo pueden ser signo y magnitud, complemento de 1, complemento de 2 dependiendo de la arquitectura. Un número complemento de 2 es ventajoso por que la ALU no necesita diseñar una capacidad de resta dado que o bien un valor negativo o bien un valor positivo en complemento de 2 requiere solamente una suma dentro de la ALU. Los números se describen comúnmente en abreviatura, donde un campo de 12 bits define una dirección de un bloque de 4.096 bytes y se describe comúnmente como un bloque de 4Kbytes (Kilobytes), por ejemplo.

Con referencia a la FIG. 19B, información de instrucciones de ramal para ejecutar una instrucción de ramal se envía típicamente a una unidad de ramal 5058 que a menudo emplea un algoritmo de predicción de ramal tal como una tabla de historia de ramal 5082 para predecir el resultado del ramal antes de que se completen otras operaciones condicionales. El objetivo de la instrucción de ramal actual se traerá y ejecutará especulativamente antes de que se completen las operaciones condicionales. Cuando se completan las operaciones condicionales las instrucciones de ramal ejecutadas especulativamente o bien se completan o bien se descartan en base a las condiciones de la operación condicional y el resultado especulado. Una instrucción de ramal típica puede comprobar los códigos de condición y ramal para una dirección objetivo si los códigos de condición cumplen el requisito de ramal de la instrucción de ramal, una dirección objetivo se puede calcular en base a varios números que incluyen unos encontrados en los campos de registro o un campo inmediato de la instrucción por ejemplo. La unidad de ramal 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 ramal 5058 puede comunicar con registros generales 5059, la unidad de decodificación despacho 5056 u otros circuitos 5073, por ejemplo.

La ejecución de un grupo de instrucciones se puede interrumpir por una variedad de razones incluyendo una conmutación de contexto iniciada por un sistema operativo, una excepción o error de programa que causa una conmutación de contexto, una señal de interrupción de I/O que causa una conmutación de contexto o actividad de múltiples hilos de una pluralidad de programas (en un entorno de múltiples hilos), por ejemplo. Preferiblemente una acción de conmutación de contexto guarda información de estado acerca de un programa que se ejecuta actualmente y entonces carga la información de estado acerca de otro programa que se invoca. La información de estado se puede guardar en registros hardware o en memoria por ejemplo. La información de estado preferiblemente comprende un valor de contador de programa que apunta una siguiente instrucción a ser ejecutada, códigos de condición, información de traducción de memoria y contenido de registros de arquitectura. Una actividad de conmutación de contexto se puede ejercitar por circuitos hardware, programas de aplicaciones, programas de sistema operativo o código de microprograma (microcódigo, picocódigo o código interno con licencia (LIC)) sólo o en combinación.

Un procesador accede a los operandos según métodos definidos de instrucciones. La instrucción puede proporcionar un operando inmediato usando el valor de una parte de la instrucción, puede proporcionar uno o más campos de registro apuntando explícitamente o bien a registros de propósito general o bien a registros de propósito especial (registros de punto flotante por ejemplo). La instrucción puede utilizar registros implícitos identificados por un campo de código de operación como operandos. La instrucción puede utilizar ubicaciones de memoria para los operandos. Una ubicación de memoria de un operando se puede proporcionar por un registro, un campo inmediato o una combinación de registros y campo inmediato como se ejemplifica por la facilidad de desplazamiento largo de la z/Architecture® en donde la instrucción define un registro base, un registro de índice y un campo inmediato (campo de desplazamiento) que se suman juntos para proporcionar la dirección del operando en memoria por ejemplo. La ubicación en la presente memoria típicamente implica una ubicación en memoria principal (almacenamiento principal) a menos que se indique de otro modo.

Con referencia a la FIG. 19C, un procesador accede al almacenamiento usando 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 memoria 5053 y cargando el operando en un registro 5059 u otra ubicación de memoria 5053 o puede realizar una operación de almacenamiento obteniendo la dirección del operando objetivo en memoria 5053 y almacenando datos obtenidos de un registro 5059 u otra ubicación de memoria 5053 en la ubicación de operando objetivo en la memoria 5053. La unidad de almacenamiento/carga 5060 puede ser especulativa y puede acceder a memoria en una secuencia que está fuera de orden respecto a la secuencia de instrucciones, no obstante la unidad de carga/almacenamiento 5060 va a mantener la apariencia para los programas de que las instrucciones se ejecutaron en orden. Una unidad de carga/almacenamiento 5060 puede comunicar con registros generales 5059, unidad de decodificación/despacho 5056, interfaz de caché/memoria 5053 u otros elementos 5083 y comprende varios circuitos de registro, ALU 5085 y lógica de control 5090 para calcular direcciones de almacenamiento y para proporcionar secuenciamiento de canalización para mantener las operaciones en orden. Algunas operaciones pueden estar fuera de orden pero la unidad de carga/almacenamiento proporciona una funcionalidad para hacer a las operaciones fuera de orden parecer al programa como que han sido realizadas en orden, como es bien conocido en la técnica.

Preferiblemente las direcciones que “ve” un programa de aplicaciones se conocen a menudo como direcciones virtuales. Las direcciones virtuales se conocen algunas veces como “direcciones lógicas” y “direcciones eficaces”. Estas direcciones virtuales son virtuales por que se redirigen a una ubicación de memoria física por una variedad de tecnologías de traducción de direcciones dinámica (DAT) incluyendo, pero no limitado a, simplemente prefijar una dirección virtual con un valor de desplazamiento, traducir la dirección virtual a través de una o más tablas de traducción, las tablas de traducción preferiblemente que comprenden al menos una tabla de segmento y una tabla de página solas o en combinación, preferiblemente, la tabla de segmento que tiene una entrada que apunta a la tabla de página. En la z/Architecture®, se proporciona una jerarquía de traducción que incluye una primera tabla de región, una segunda tabla de región, una tercera tabla de región, una tabla de segmento y una tabla de página opcional. El rendimiento de la traducción de direcciones se mejora a menudo utilizando un almacenador temporal de traducción anticipada (TLB) que comprende entradas que correlacionan una dirección virtual a una ubicación de memoria física asociada. Las entradas se crean cuando la DAT traduce una dirección virtual que usa las tablas de traducción. El uso posterior de la dirección virtual entonces puede utilizar la entrada del TLB más que los accesos de tabla de traducción secuencial lenta. El contenido del TLB se puede gestionar por una variedad de algoritmos de sustitución que incluyen LRU (Menos Recientemente usado).

En el caso donde el procesador es un procesador de un sistema multiprocesador, cada procesador tiene la responsabilidad de mantener los recursos compartidos, tales como I/O, cachés, TLB y memoria, entrelazados por coherencia. Típicamente, se utilizarán tecnologías de “fisgoneo” en el mantenimiento de la coherencia de caché. En un entorno de fisgoneo, cada línea de caché se puede marcar como que está en cualquiera de un estado compartido, un estado exclusivo, un estado cambiado, un estado inválido y similares a fin de facilitar la compartición.

Las unidades de I/O 5054 (FIG. 18) dotan al procesador con medios para unión a dispositivos periféricos incluyendo cinta, disco, impresoras, visualizadores y redes por ejemplo. Las unidades de I/O a menudo se presentan al programa de ordenador por controladores software. En ordenadores centrales, tales como System z® de IBM®, los adaptadores de canal y adaptadores de sistema abierto son unidades de I/O del ordenador central que proporcionan las comunicaciones entre el sistema operativo y los dispositivos periféricos.

Además, otros tipos de entornos informáticos pueden beneficiarse de uno o más aspectos de la presente invención. Como ejemplo, un entorno puede incluir un emulador (por ejemplo, software u otros mecanismos de emulación), en el cual una arquitectura particular (incluyendo, por ejemplo, ejecución de instrucciones, funciones de arquitectura, tales como traducción de direcciones y registros de arquitectura) o un subconjunto de la misma se emula (por ejemplo, en un sistema informático 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 ejecuta el emulador pueda tener una arquitectura diferente de las capacidades que se emulan. Como ejemplo, en modo de emulación, se decodifica la instrucción u operación específica que se emula y se construye una función de emulación adecuada 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 extracción de instrucciones para traer instrucciones desde la memoria y para, opcionalmente, proporcionar almacenamiento temporal local para la instrucción traída; una unidad de decodificación de instrucciones para recibir las instrucciones traídas y para determinar el tipo de instrucciones que se han traído; y una

5 unidad de ejecución de instrucciones para ejecutar las instrucciones. La ejecución puede incluir cargar datos en un registro desde memoria; almacenar datos de nuevo a la memoria desde un registro; o realizar algún tipo de operación aritmética o lógica, según se determine por la unidad de decodificación. En un ejemplo, cada unidad se implementa en software. Por ejemplo, las operaciones que se realizan por las unidades se implementan como una o más subrutinas dentro del software emulador.

10 Más particularmente, en un ordenador central, se usan instrucciones de máquina de arquitectura por programadores, normalmente hoy en día programadores "C", a menudo por medio de una aplicación de compilador. Estas instrucciones almacenadas en el medio de almacenamiento se pueden ejecutar nativamente en un Servidor de z/Architecture[®] de IBM[®] o alternativamente en máquinas que ejecutan otras arquitecturas. Se pueden emular en servidores de ordenador central de IBM[®] existentes y en el futuro y en otras máquinas de IBM[®] (por ejemplo, Servidores de Power Systems y Servidores de System x[®]). Se pueden ejecutar en máquinas que ejecutan Linux en una amplia variedad de máquinas que usan hardware fabricado por IBM[®], Intel[®], AMD[™] y otros. Además de la

15 ejecución en ese hardware bajo una z/Architecture[®], se puede usar Linux así como máquinas que usan emulación mediante Hercules (ver www.hercules-390.org) o FSI (Fundamental Software, Inc) (ver www.funsoft.com), donde generalmente la ejecución es en un modo de emulación. En el modo de emulación, se ejecuta el software de emulación por un procesador nativo para emular la arquitectura de un procesador emulado.

20

El procesador nativo típicamente ejecuta software de emulación que comprende o bien un microprograma o bien un sistema operativo nativo para realizar la emulación del procesador emulado. El software de emulación es responsable para traer y ejecutar instrucciones de la arquitectura de procesador emulado. El software de emulación mantiene un contador de programa emulado para hacer el seguimiento de los límites de instrucción. El software de emulación puede traer una o más instrucciones de máquina emulada a la vez y convertir la una o más instrucciones de máquina emulada a un grupo correspondiente de instrucciones de máquina nativa para ejecución por el procesador nativo. Estas instrucciones convertidas se pueden almacenar en caché de manera que se puede consumir una conversión más rápida. No obstante, el software de emulación va a mantener las reglas de arquitectura de la arquitectura de procesador emulado para asegurar sistemas operativos y aplicaciones escritas para operar correctamente el procesador emulado. Por otra parte, el software emulado va a proporcionar recursos identificados por la arquitectura de procesador emulado incluyendo, pero no limitados a, registros de control, registros de propósito general, registros de punto flotante, función de traducción de direcciones dinámica incluyendo tablas de segmento y tablas de página por ejemplo, mecanismos de interrupción, mecanismos de conmutación de contexto, relojes de Hora del Día (TOD) e interfaces de arquitectura a subsistemas de I/O de manera que un sistema operativo o un programa de aplicaciones diseñado para ejecutarse en el procesador emulado, se puede ejecutar en el procesador nativo que tiene el software de emulación.

25

30

35

Una instrucción específica que se emula se decodifica y se llama a una subrutina para realizar la función de la instrucción individual. Una función de software de emulación que emula una función de un procesador emulado se implementa, por ejemplo, en un controlador o subrutina "C" o algún otro método de suministro de un controlador para el hardware específico que estará dentro de los conocimientos de los expertos en la técnica después de comprender la descripción de la realización preferida. Varias patentes de emulación software y hardware incluyendo, pero no limitadas a la Concesión de Patente de EE.UU. N° 5.551.013, titulada "Multiprocessor for Hardware Emulation", de Beausoleil et al.; y la Concesión de Patente de EE.UU. N° 6.009.261, titulada "Preprocessing of Stored Target Routines for Emulating Incompatible Instructions on a Target Processor", de Scalzi et al; y la Concesión de Patente de EE.UU. N° 5.574.873, titulada "Decoding Guest Instruction to Directly Access Emulation Routines that Emulate the Guest Instructions", de Davidian et al.; y la Concesión de Patente de EE.UU. N° 6.308.255, titulada "Symmetrical Multiprocessing Bus and Chipset Used for Coprocessor Support Allowing Non-Native Code to Run in a System", de Gorishhek et al; y la Concesión de Patente de EE.UU. N° 6.463.582, titulada "Dynamic Optimizing Object Code Translator for Architecture Emulation and Dynamic Optimizing Object Code Translation Method", de Lethin et al; y la Concesión de Patente de EE.UU. N° 5.790.825, titulada "Method for Emulating Guest Instructions on a Host Computer Through Dynamic Recompilation of Host Instructions", de Eric Traut; y muchas otras, ilustran una variedad de formas conocidas para lograr emulación de un formato de instrucciones de arquitectura para una máquina diferente para una máquina objetivo disponible para los expertos en la técnica.

40

45

50

En la FIG. 20, se proporciona un ejemplo de un sistema de ordenador principal emulado 5092 que emula un sistema de ordenador principal 5000' de una arquitectura de ordenador principal. En el sistema de ordenador principal emulado 5092, el procesador de ordenador principal (CPU) 5091 es un procesador de ordenador principal emulado (o procesador de ordenador principal virtual) y comprende un procesador de emulación 5093 que tiene una diferente arquitectura de conjunto de instrucciones nativas distintas de las del procesador 5091 del ordenador principal 5000'. El sistema de ordenador principal emulado 5092 tiene una memoria 5094 accesible al procesador de emulación 5093. En la realización ejemplo, la memoria 5094 está dividida en una porción de memoria de ordenador principal 5096 y una porción de rutinas de emulación 5097. La memoria de ordenador principal 5096 está disponible para programas del ordenador principal emulado 5092 según una arquitectura de ordenador principal. El procesador de emulación 5093 ejecuta instrucciones nativas de un conjunto de instrucciones de arquitectura de una arquitectura

55

60

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 de ordenador principal para ejecución desde un programa en la memoria de ordenador principal 5096 empleando una o más instrucción(instrucciones) obtenida(s) en una rutina de secuencia y acceso/decodificación que puede decodificar la(s) instrucción(instrucciones) de ordenador principal accedida(s) para determinar una rutina de ejecución de instrucciones nativas para emular la función de la instrucción de ordenador principal accedida. Otras facilidades que se definen para la arquitectura del sistema de ordenador principal 5000' se pueden emular por rutinas de facilidades de arquitectura, que incluyen tales facilidades como registros de propósito general, registros de control, traducción de direcciones dinámica y caché de procesador y soporte de subsistema de I/O, por ejemplo. Las rutinas de emulación también pueden tomar ventaja de las funciones disponibles en el procesador de emulación 5093 (tales como registros generales y traducción dinámica de direcciones virtuales) para mejorar el rendimiento de las rutinas de emulación. También se pueden proporcionar hardware especial y motores fuera de carga para ayudar al procesador 5093 en la emulación de la función del ordenador principal 5000'.

REIVINDICACIONES

1. Un método de gestión de peticiones de interrupción en un entorno informático, **caracterizado por** comprender los pasos de:
 - 5 en respuesta a ejecutar una operación de interrupciones de registro de instrucción Modificar los Controles de Función de PCI (MPFC) que especifica un gestor de función de un adaptador, especificar (601) en una tabla localizada en un centro de entrada/salida (I/O) acoplado al adaptador, una ubicación en la memoria del sistema de un vector de bit de interrupción de adaptador (AIBV) del adaptador, el AIBV incluido en una formación de uno o más AIBV y una ubicación en la memoria del sistema de un bit de resumen de interrupción de adaptador (AISB) de una formación de AISB;
 - 10 recibir (603) desde el adaptador una petición de interrupción; y
 - en respuesta a la petición recibida, fijar (605) por el centro de I/O un indicador en el AIBV que indica un tipo de evento desde el adaptador y fijar (606) el AISB que indica que un indicador está fijado en el AIBV.
2. El método de la reivindicación 1, en donde el método además comprende presentar una interrupción a un sistema operativo, la interrupción en respuesta a la petición de interrupción.
- 15 3. El método de la reivindicación 2, en donde la petición de interrupción representa una pluralidad de interrupciones señaladas por mensaje y la interrupción al sistema operativo es parte de una notificación de evento de adaptador de entrada/salida para el sistema operativo.
4. El método de la reivindicación 2, en donde el método además comprende obtener, en respuesta a la presentación, una o más indicaciones de AIBV para uno o más adaptadores especificando al menos una razón para la interrupción por adaptador.
- 20 5. El método de la reivindicación 4, en donde la obtención comprende obtener una pluralidad de indicaciones de AIBV que especifican una pluralidad de razones para la interrupción, la pluralidad de razones para la interrupción que corresponden a la pluralidad de peticiones de interrupción.
- 25 6. El método de la reivindicación 5, en donde la obtención comprende usar el AISB en la obtención de una o más indicaciones de AIBV en la una o más AIBV.
7. El método de la reivindicación 1, en donde el ajuste del indicador en el AIBV comprende:
 - usar un identificador de la petición desde el adaptador para obtener una entrada de tabla de dispositivo, la entrada de tabla de dispositivo que comprende un valor que especifica un número de interrupciones que se permiten para el adaptador;
 - 30 determinar si un número de vector proporcionado en la petición está dentro del número de interrupciones permitidas para el adaptador;
 - en respuesta a determinar que el número de vector está dentro del número permitido de interrupciones, usar uno o más parámetros de la entrada de tabla de dispositivo para localizar una posición de inicio de la AIBV; y
 - usar el número de vector para fijar el indicador en el AIBV.
- 35 8. Un sistema que comprende medios adaptados para llevar a cabo todos los pasos del método según cualquier reivindicación del método precedente.
9. Un programa de ordenador que comprende instrucciones para llevar a cabo todos los pasos del método según cualquier reivindicación del método precedente, cuando dicho programa de ordenador se ejecuta en un sistema informático.

40

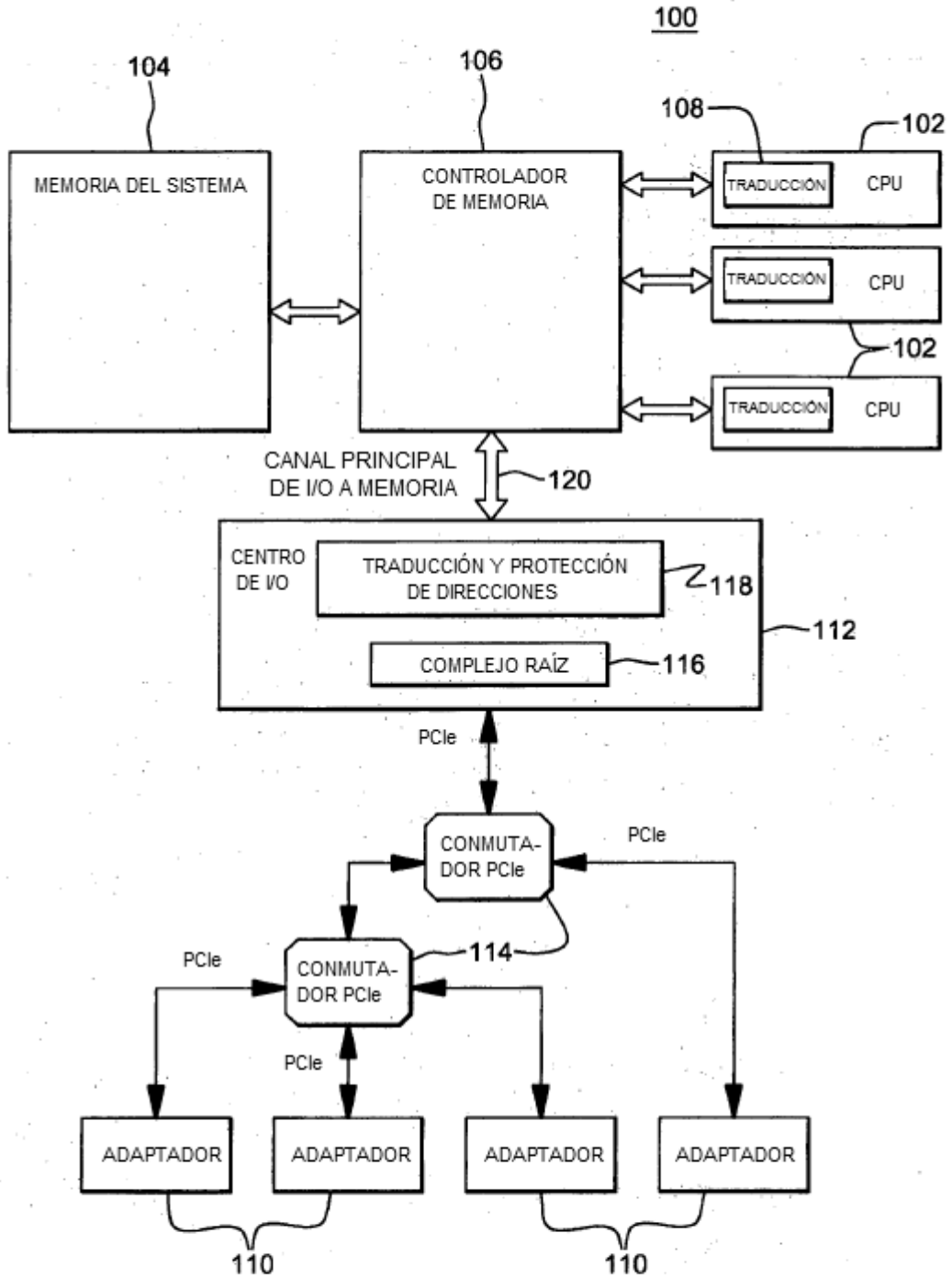


FIG. 1

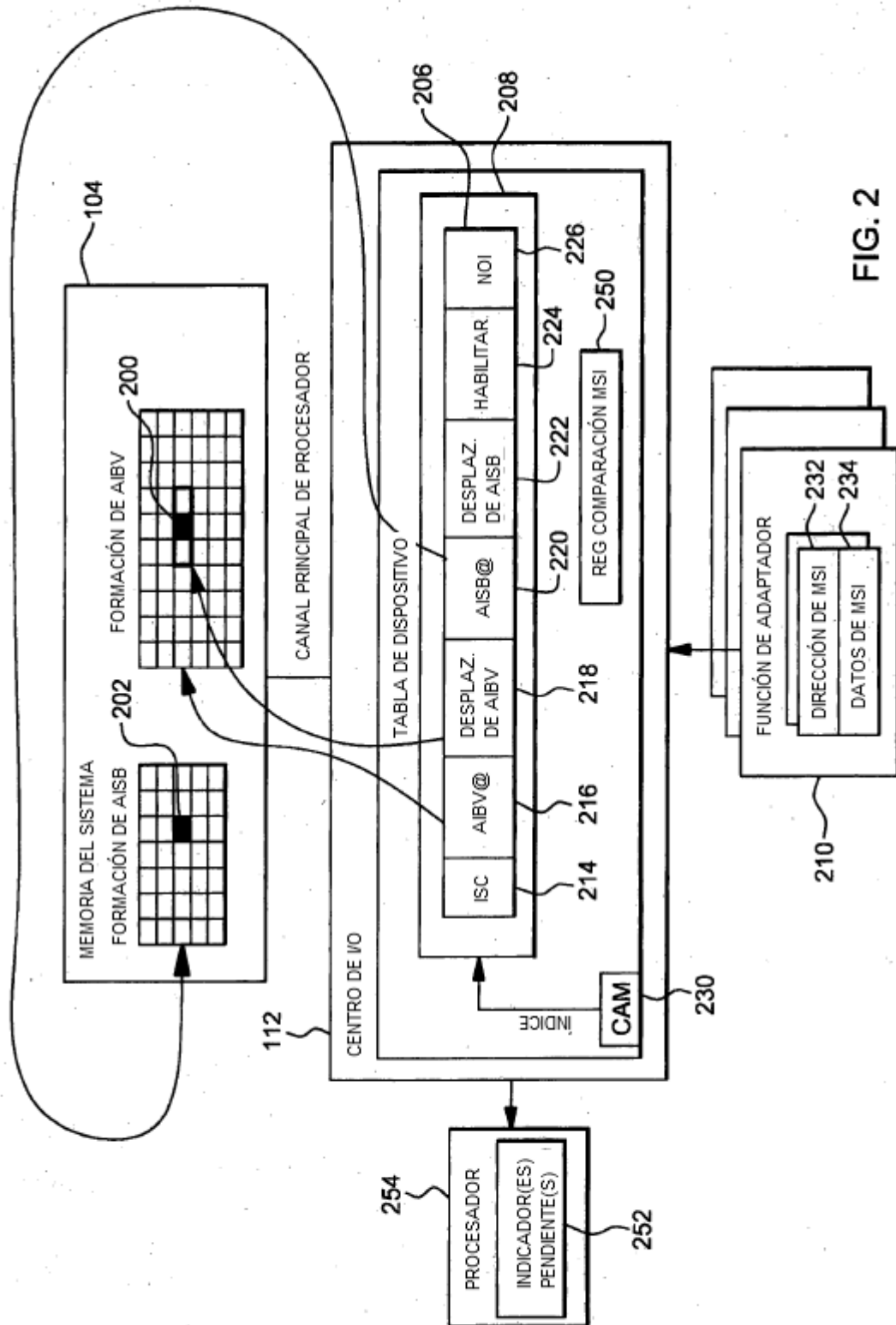


FIG. 2

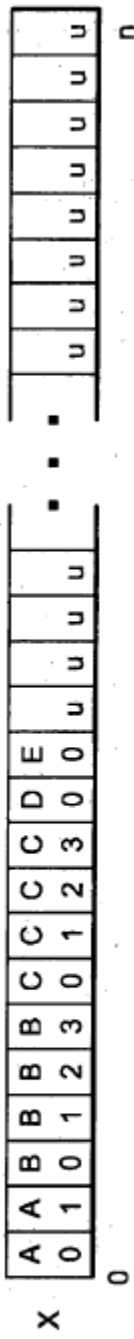


FIG. 3A

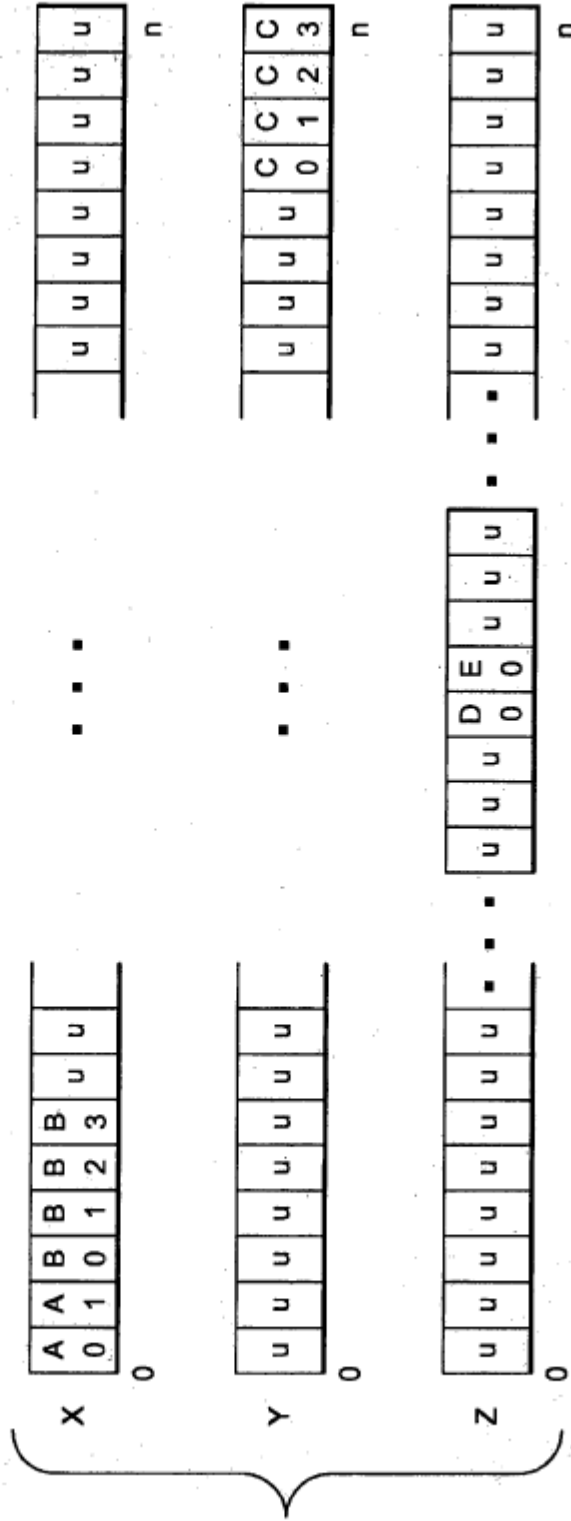


FIG. 3B

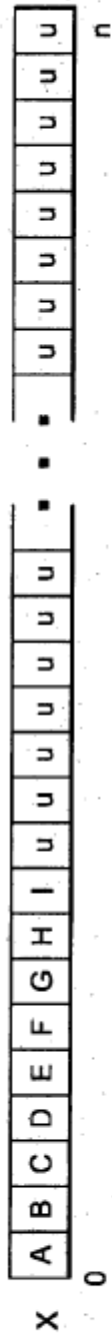


FIG. 3C

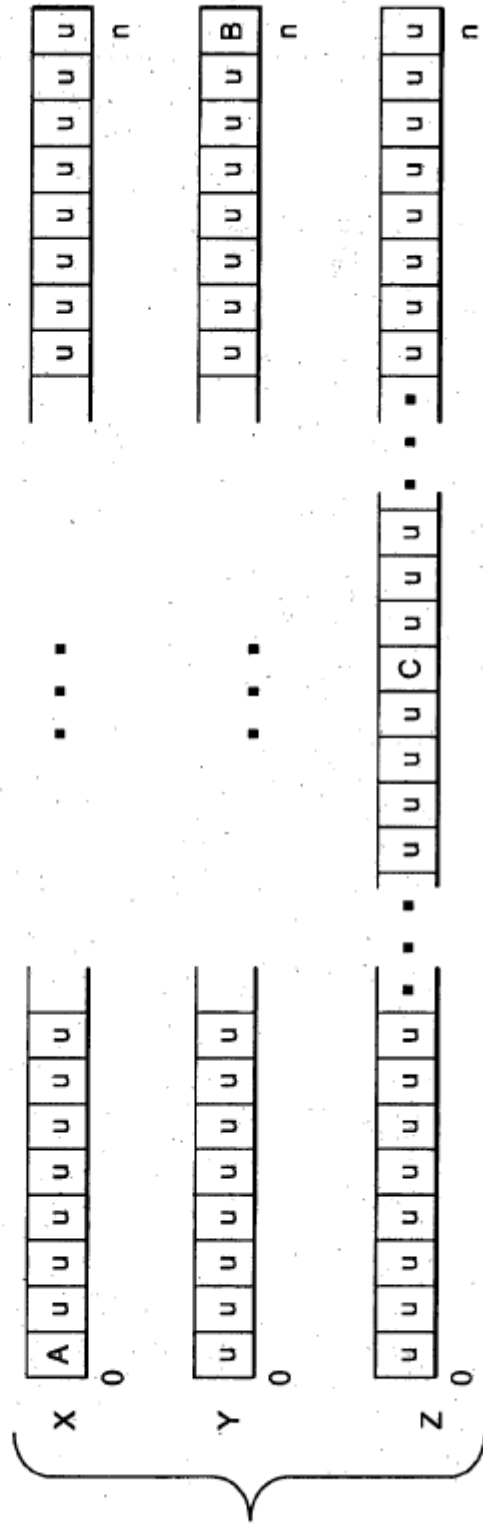


FIG. 3D

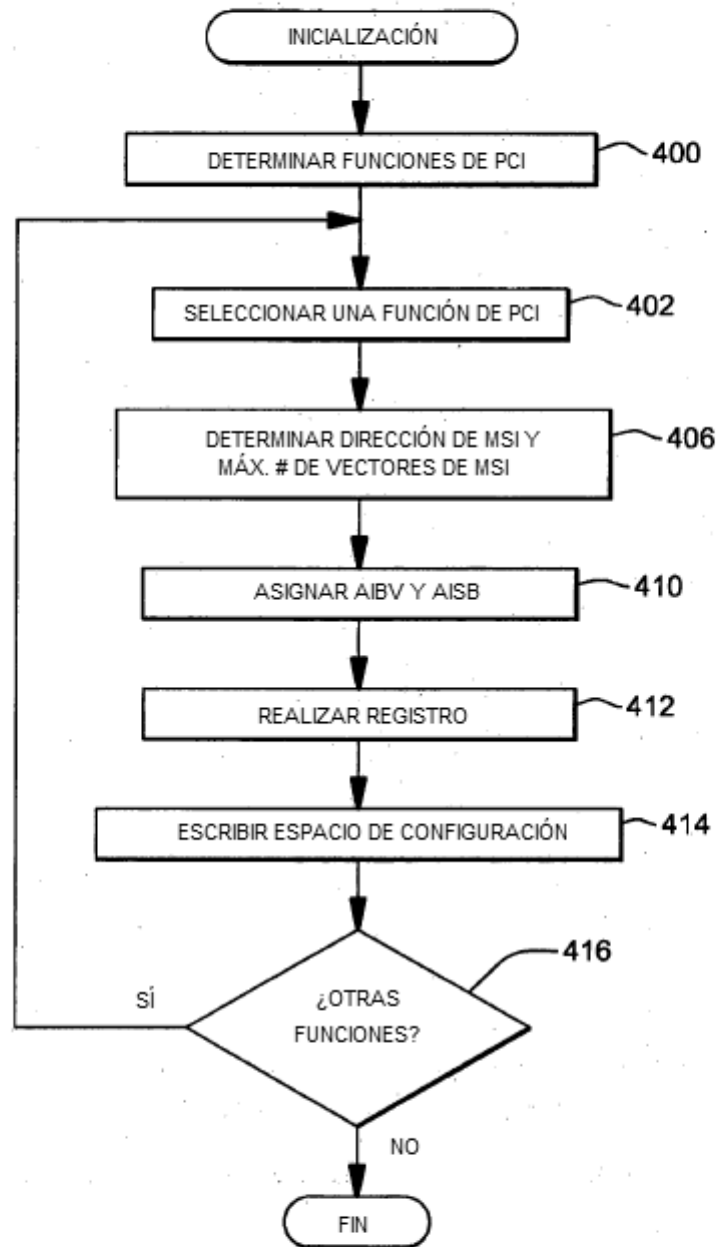


FIG. 4

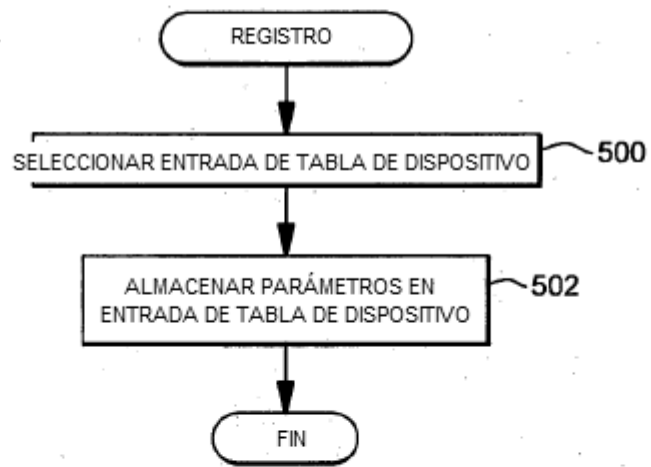


FIG. 5

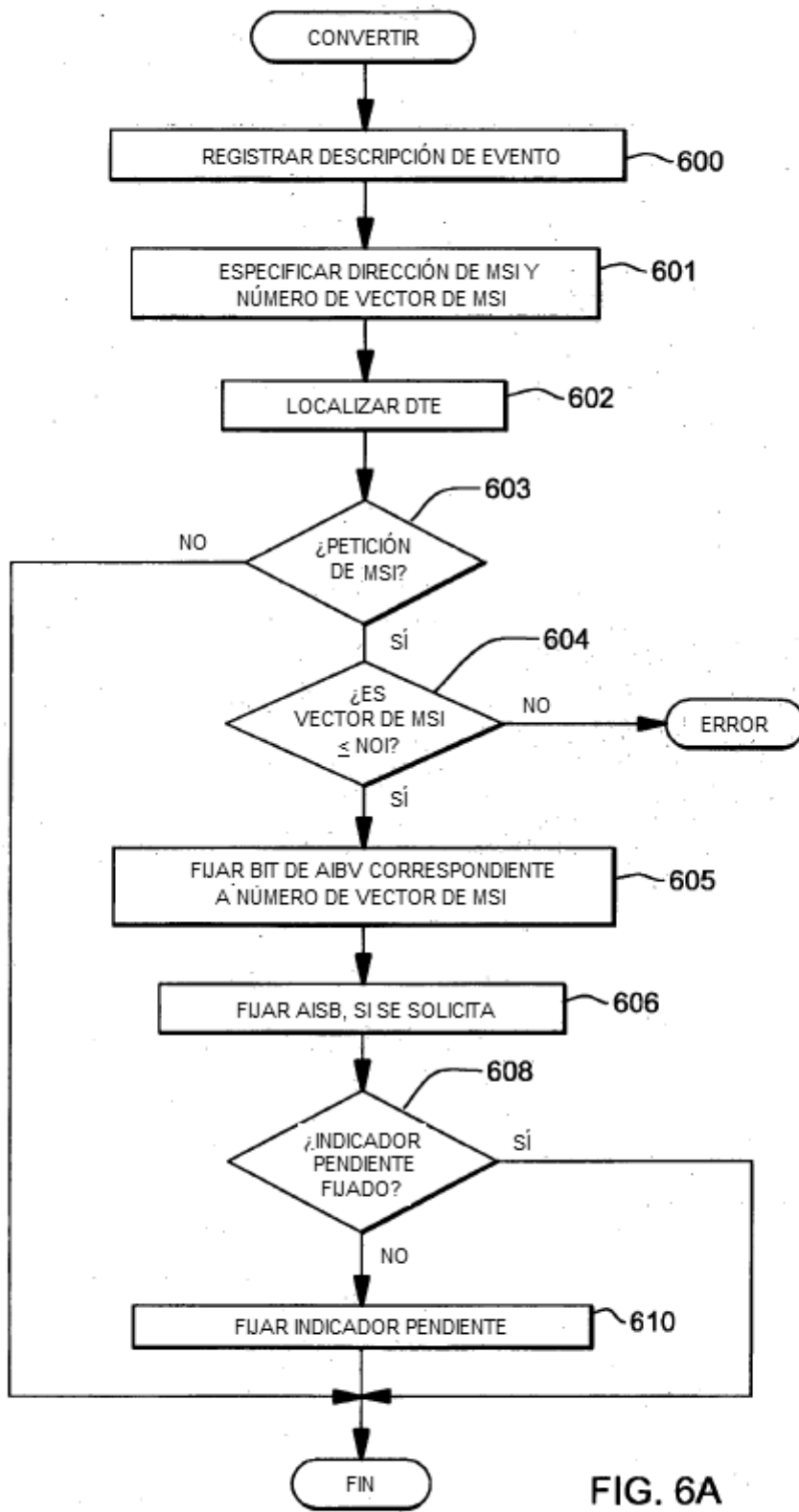


FIG. 6A

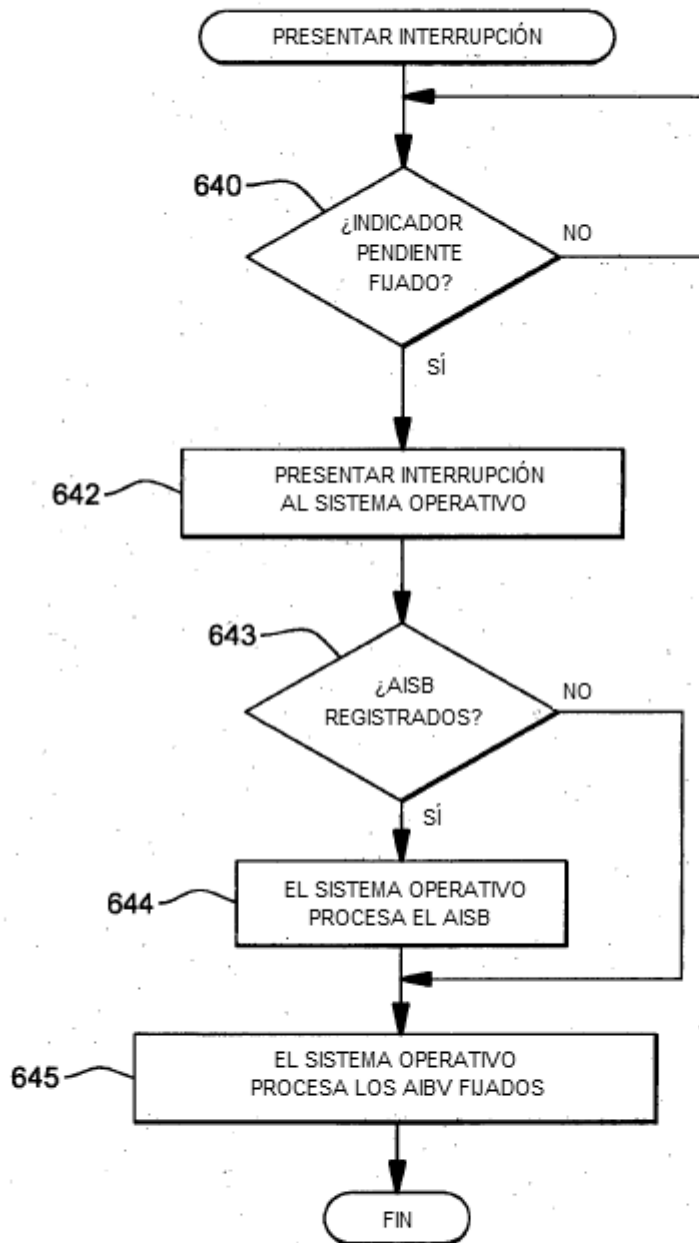


FIG. 6B

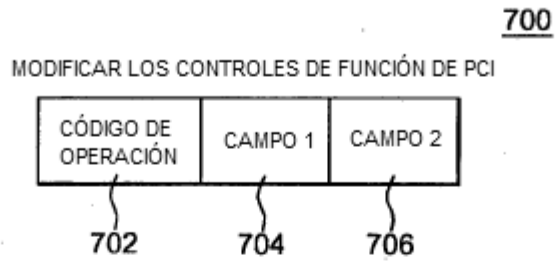


FIG. 7A

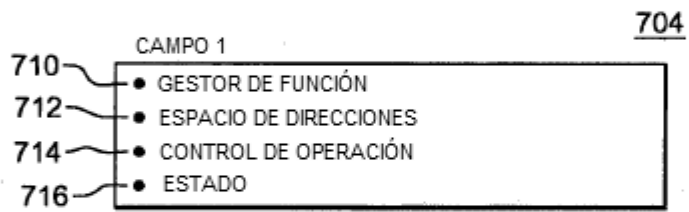


FIG. 7B

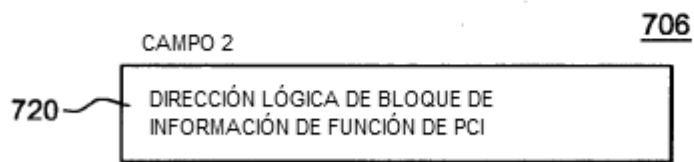


FIG. 7C

750

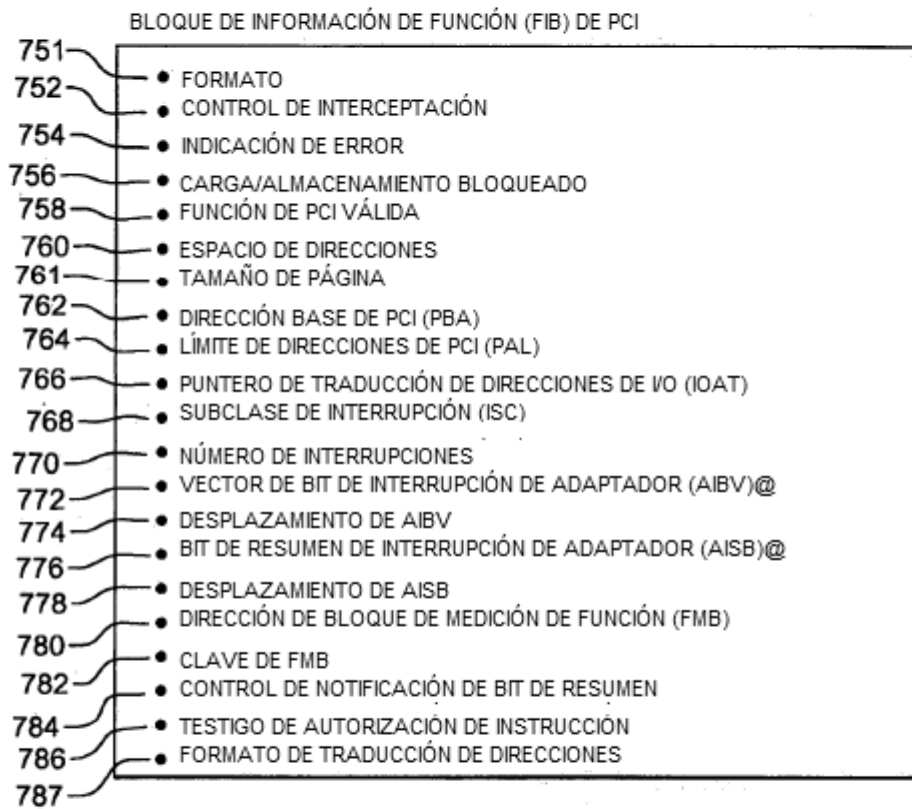


FIG. 7D

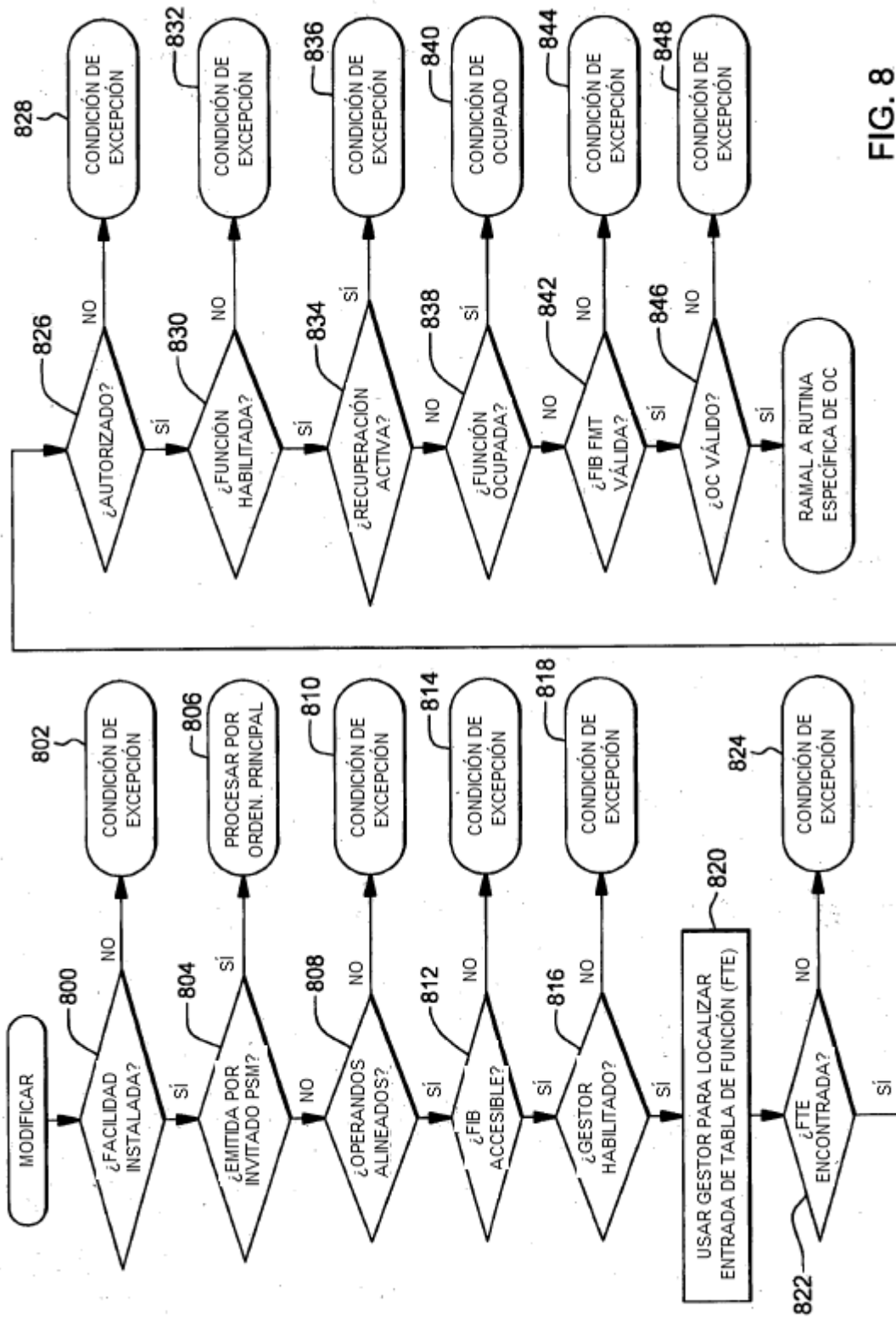


FIG. 8

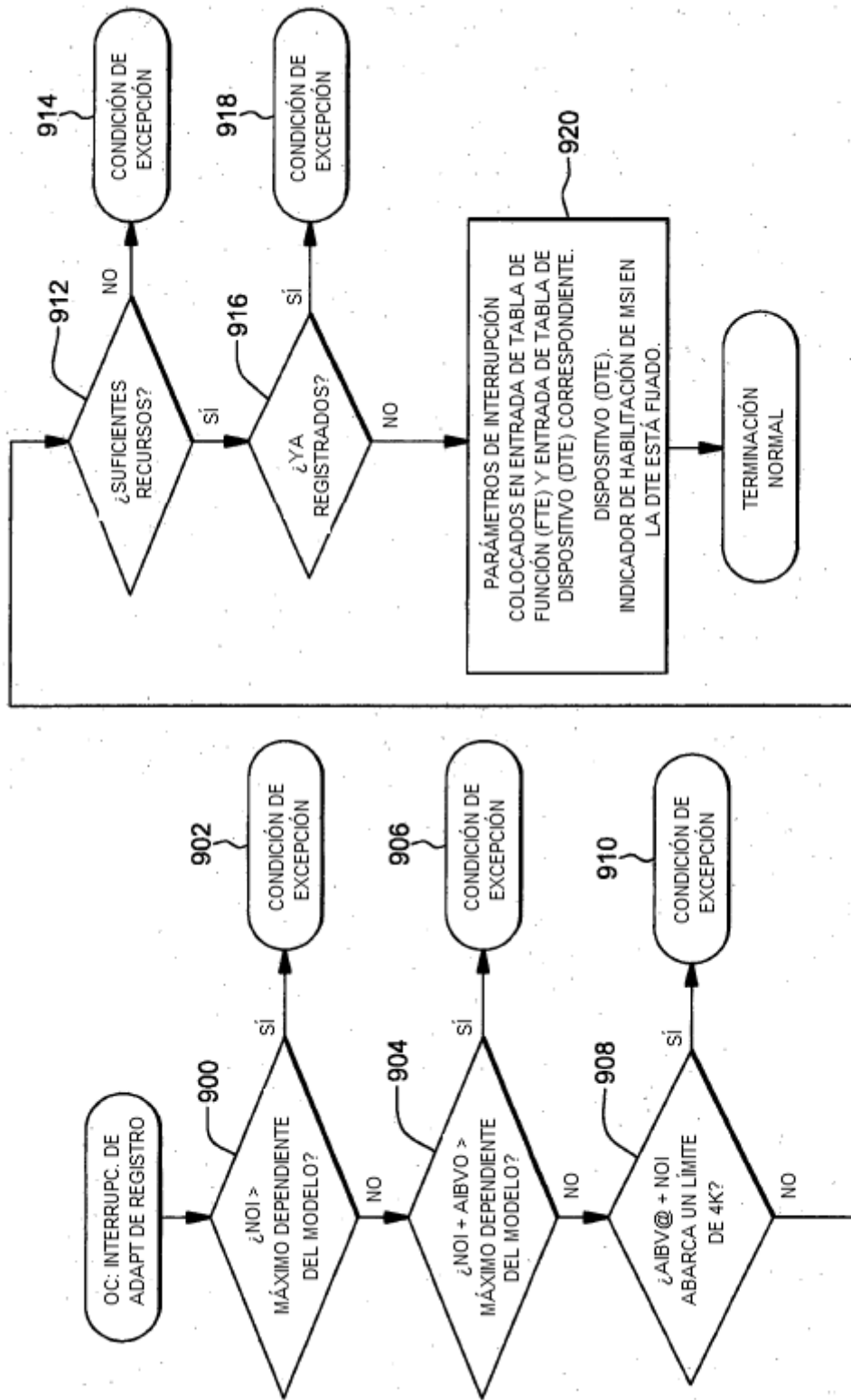


FIG. 9

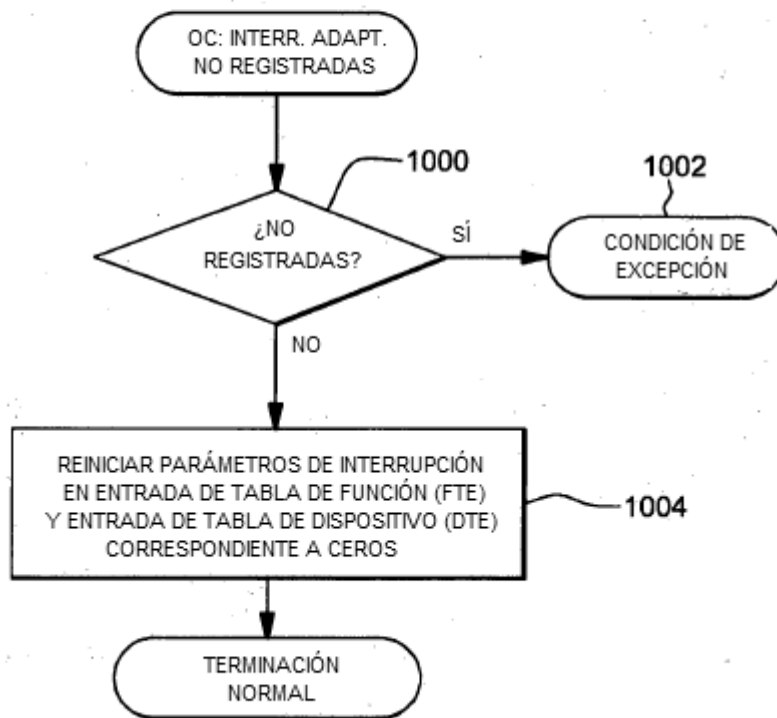
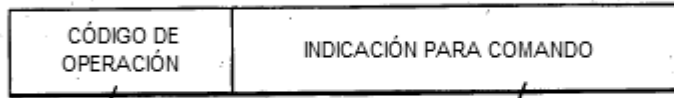


FIG. 10

INSTRUCCIÓN LLAMAR AL PROCESADOR LÓGICO (CLP)

1100



1102

1104

FIG. 11A

BLOQUE DE PETICIÓN PARA LISTA

1120

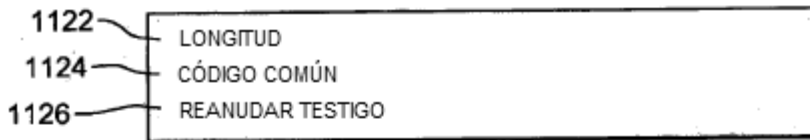


FIG. 11B

BLOQUE DE RESPUESTA PARA LISTA

1150

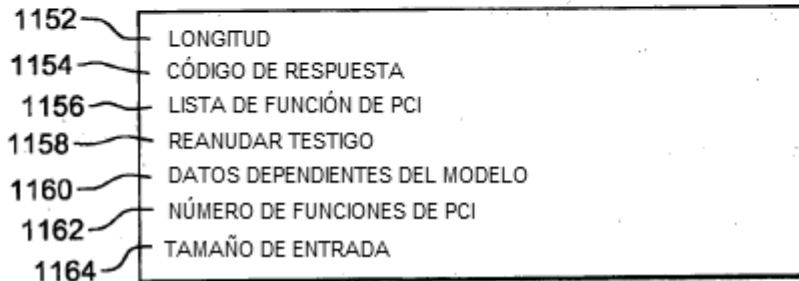


FIG. 11C

ENTRADA DE LISTA DE FUNCIÓN DE PCI

1156

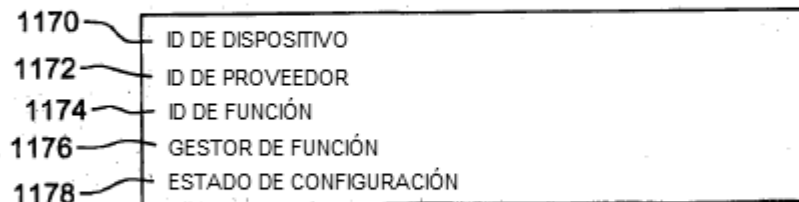


FIG. 11D

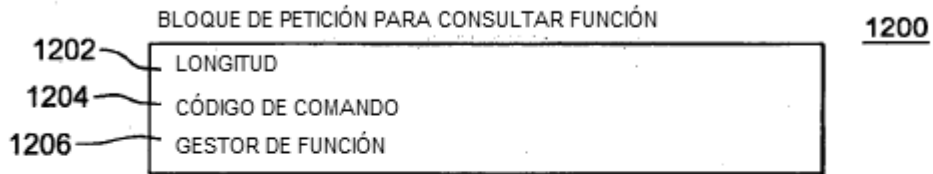


FIG. 12A

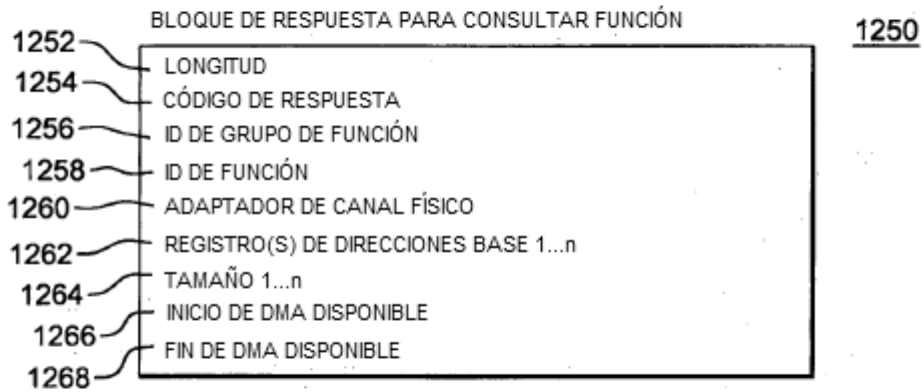


FIG. 12B

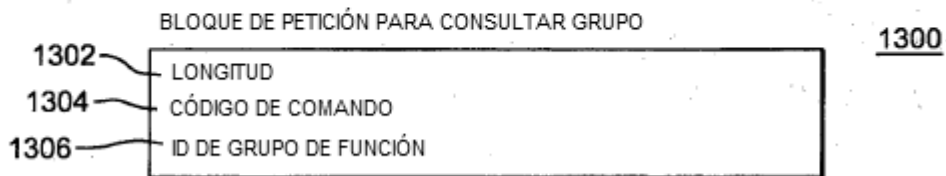


FIG. 13A

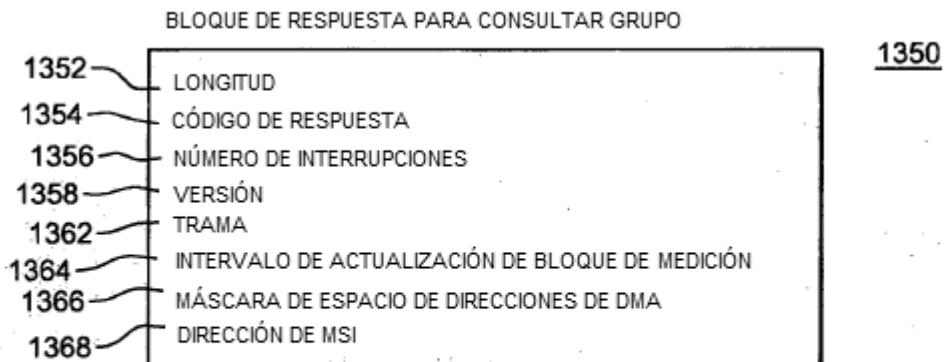


FIG. 13B

PRODUCTO DE
PROGRAMA DE
ORDENADOR
1400

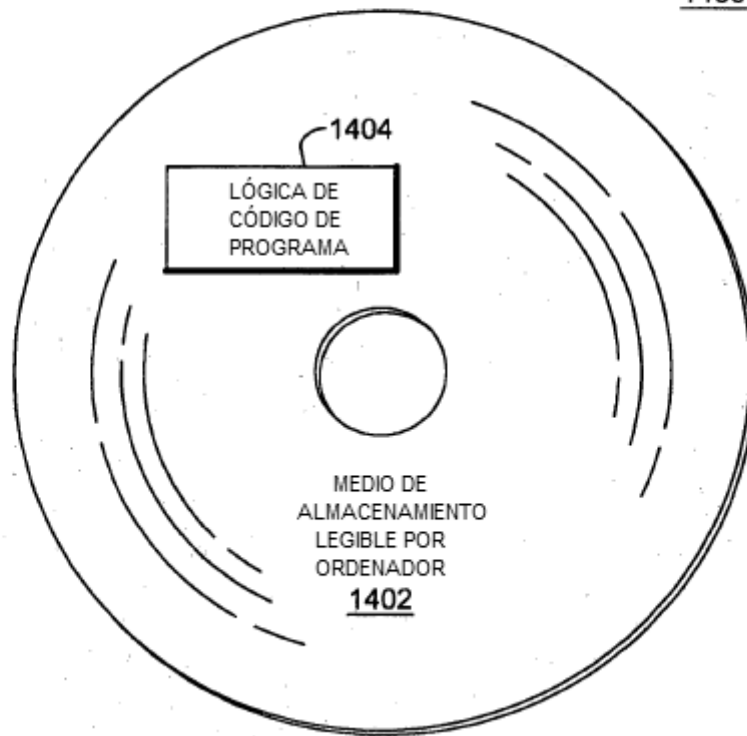


FIG. 14

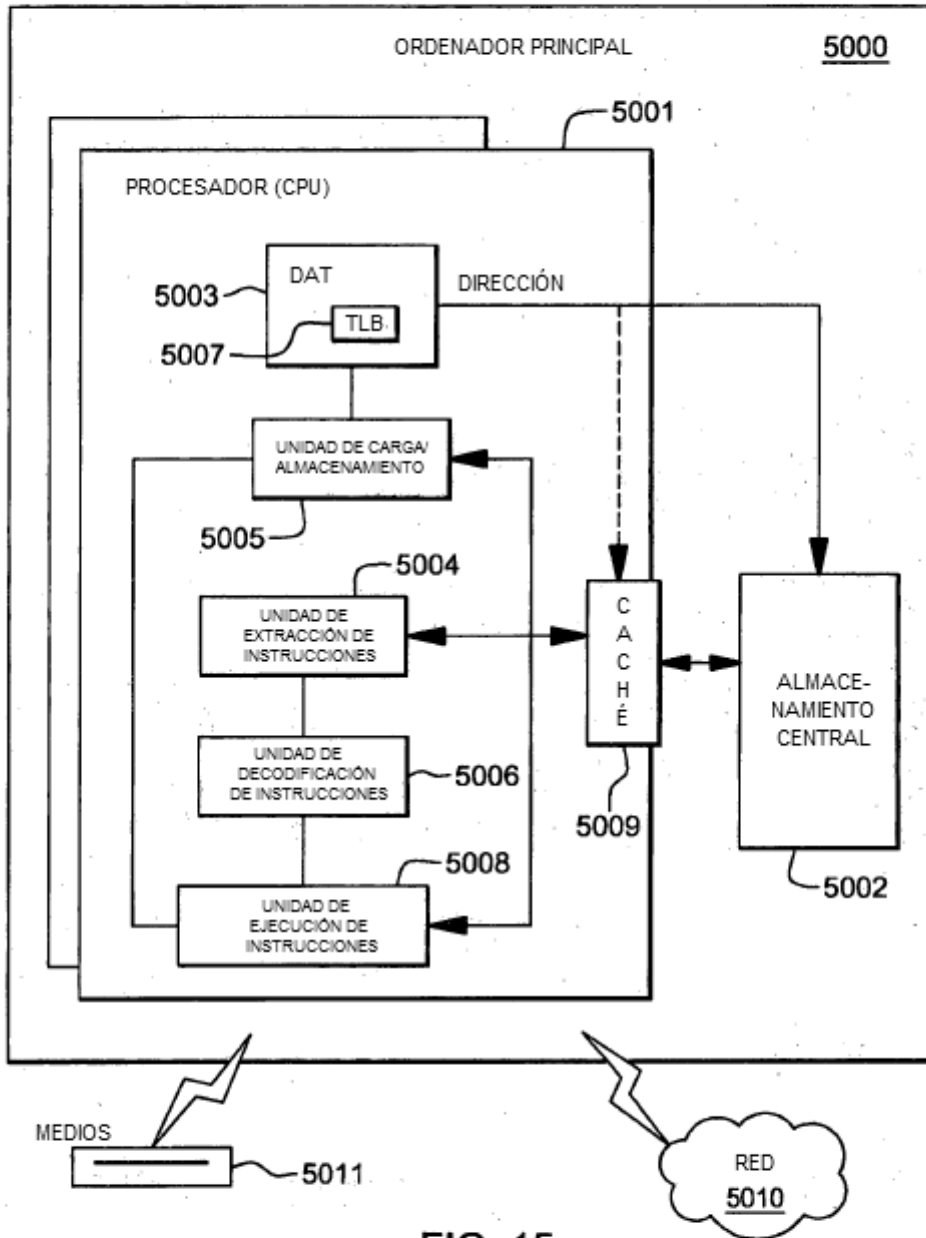


FIG. 15

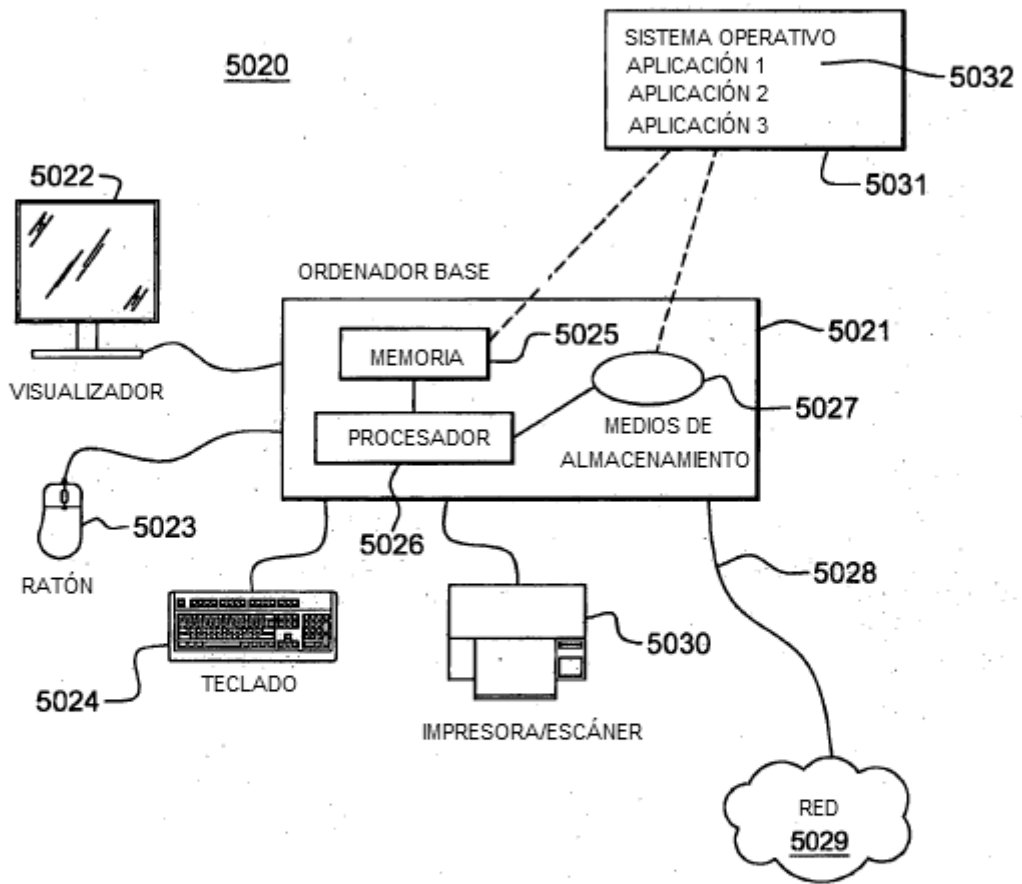


FIG. 16

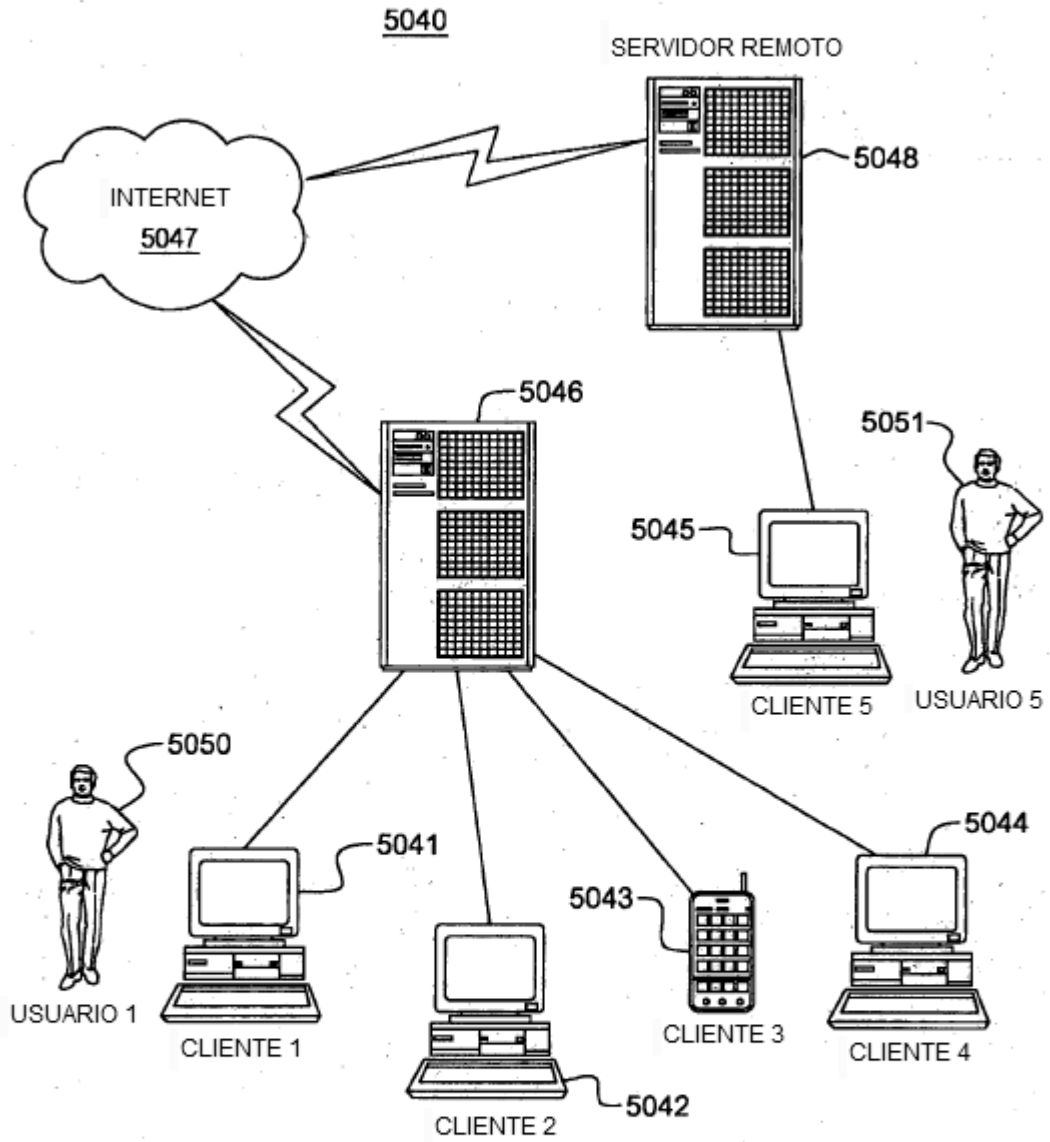


FIG. 17

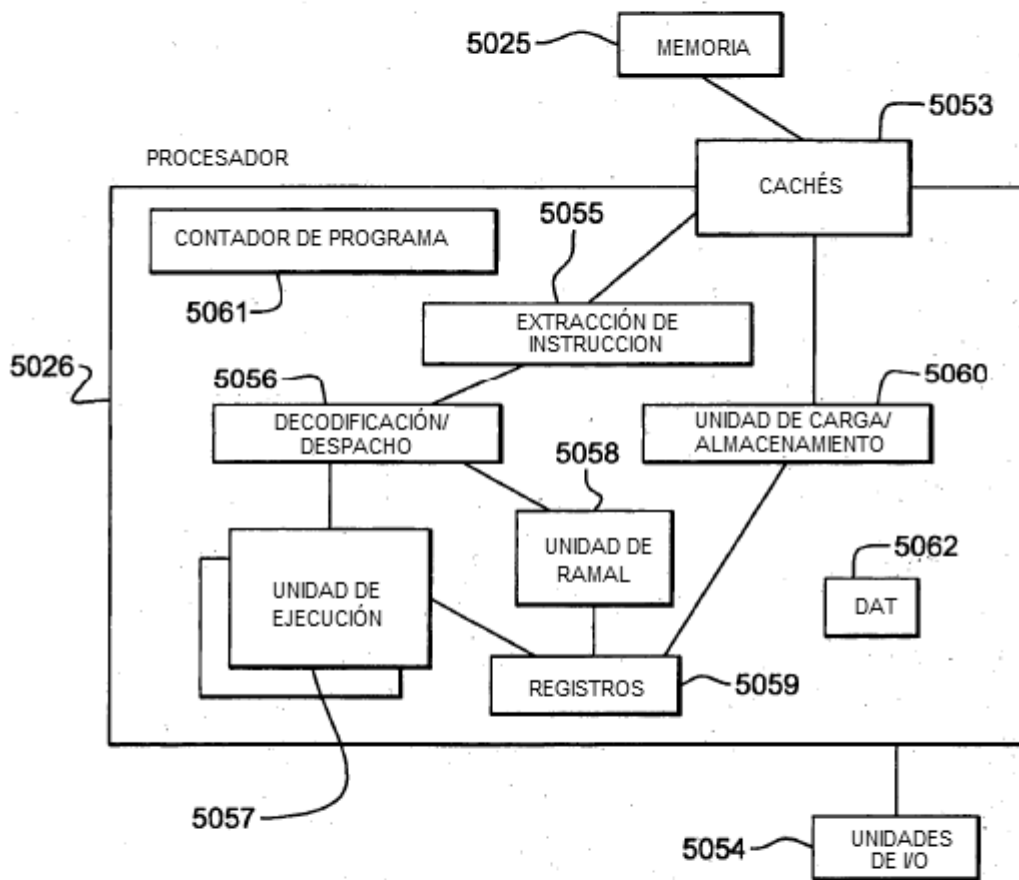


FIG. 18

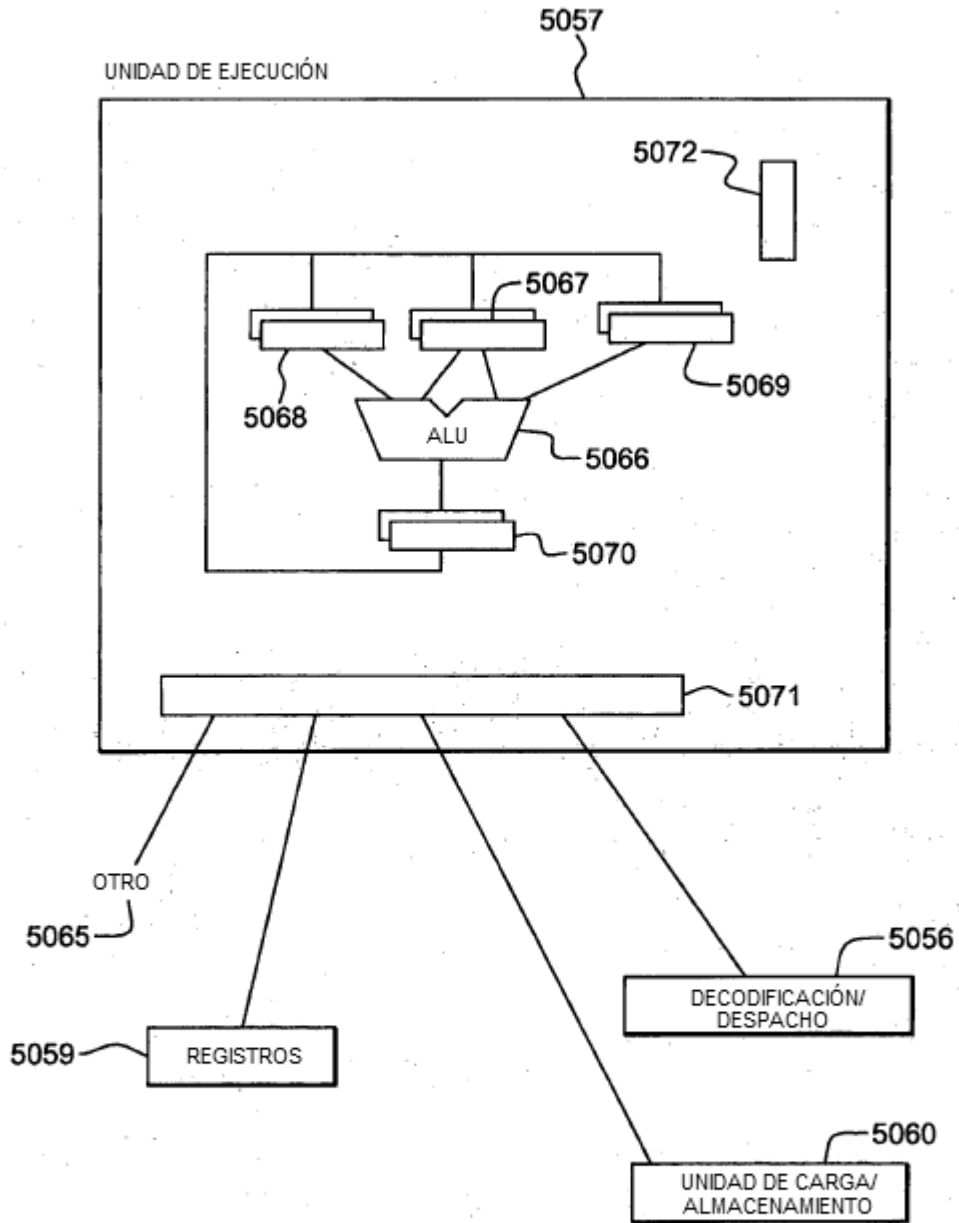


FIG. 19A

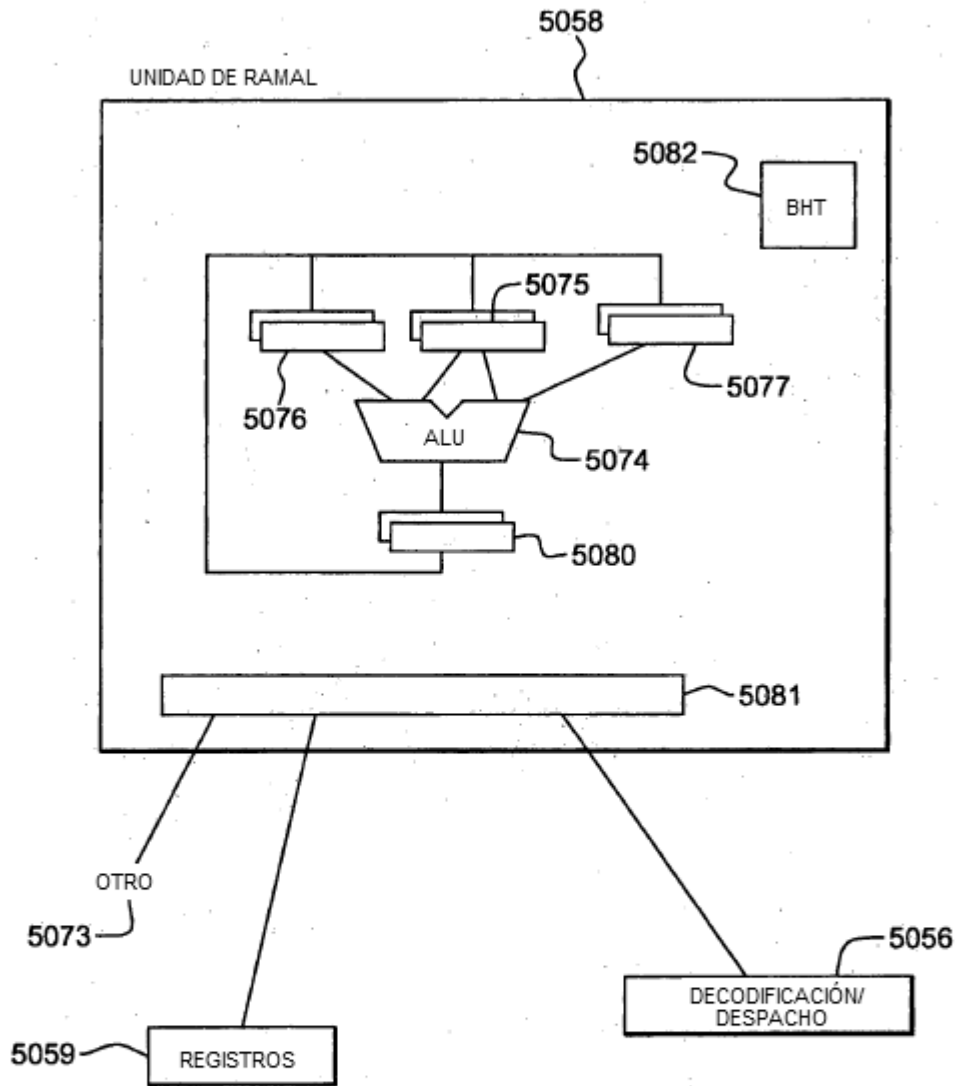


FIG. 19B

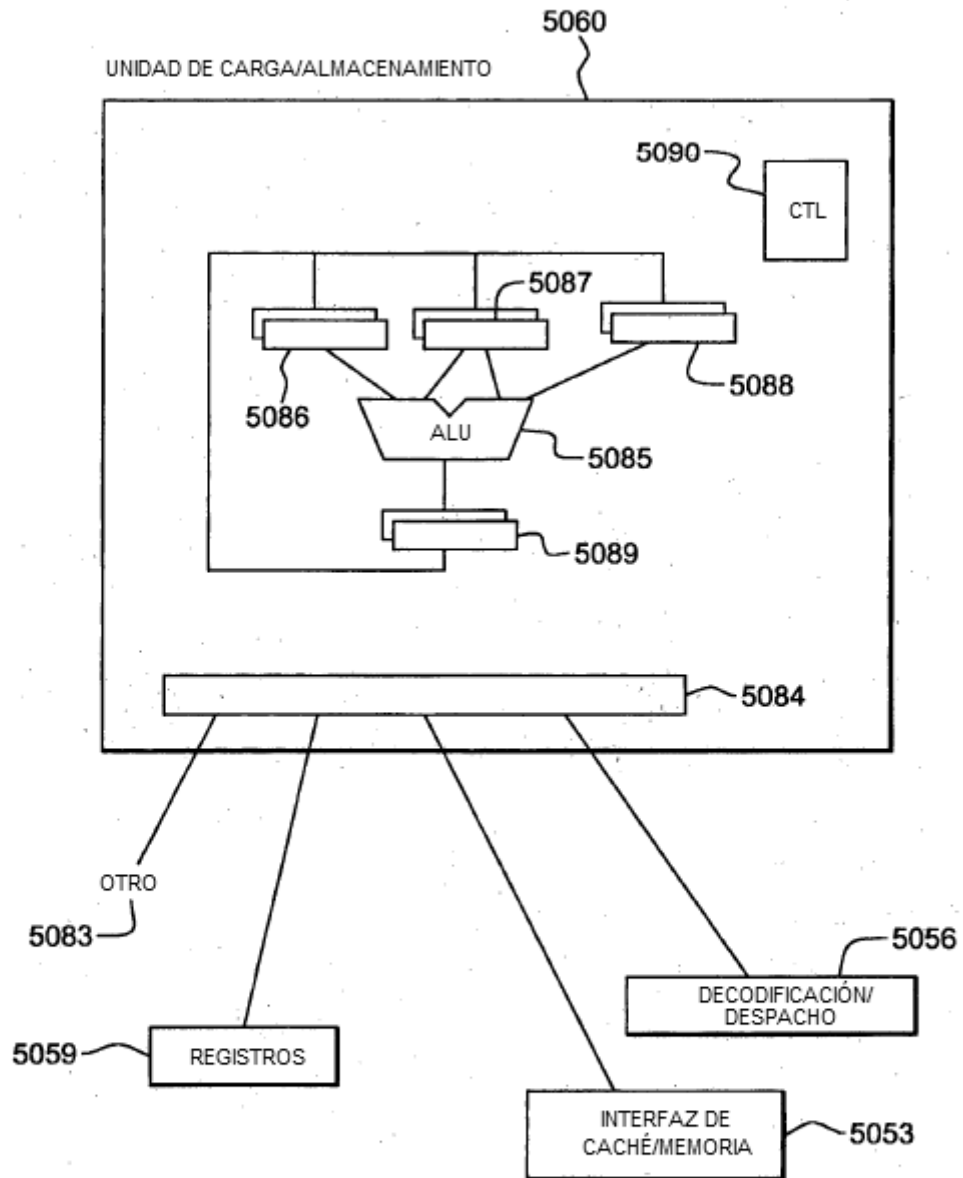


FIG. 19C

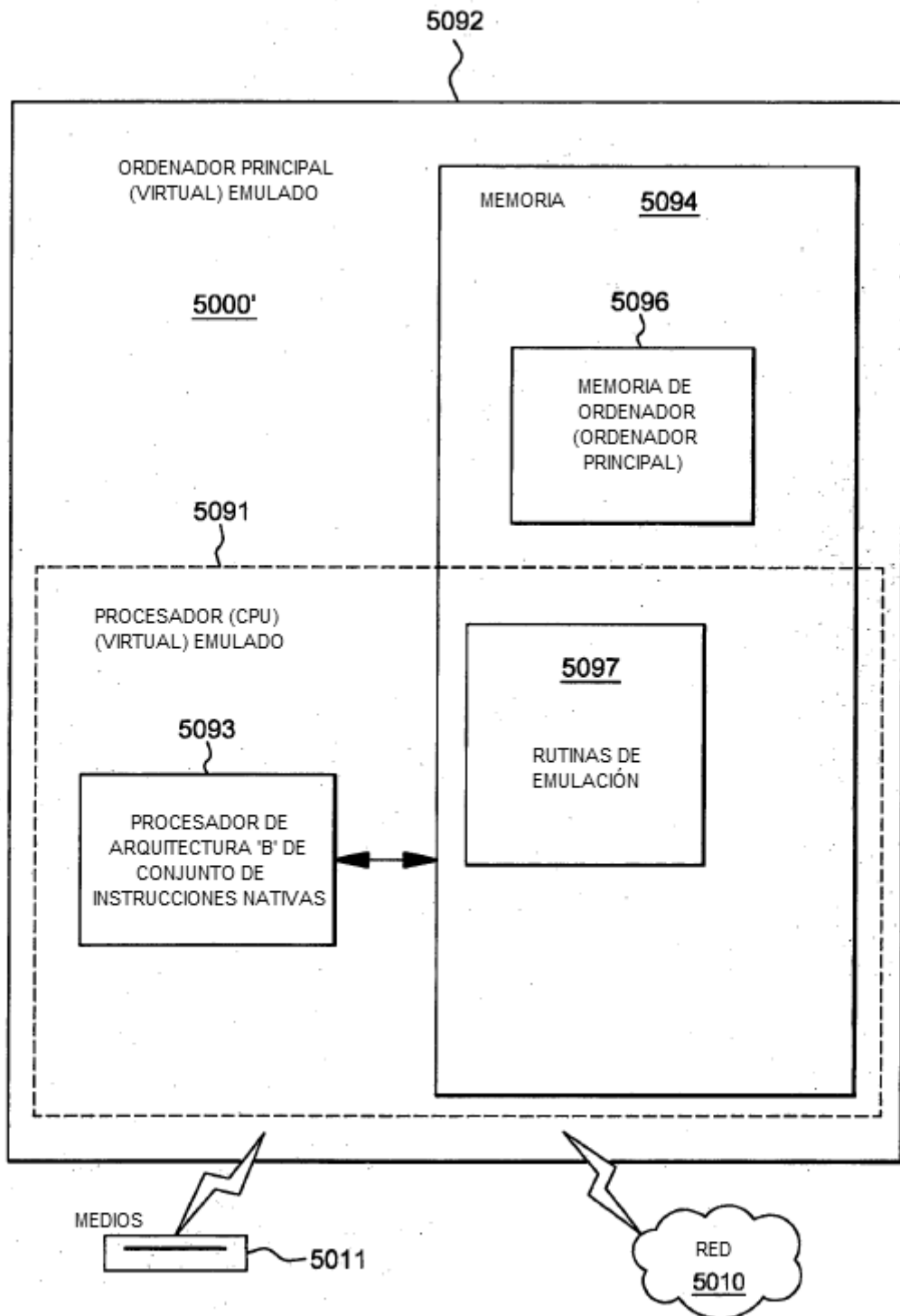


FIG. 20