

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 428 822**

51 Int. Cl.:

G06F 13/16 (2006.01)

G06F 12/10 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **08.11.2010 E 10776350 (0)**

97 Fecha y número de publicación de la concesión europea: **28.08.2013 EP 2430555**

54 Título: **Traducción de direcciones de entrada/salida a direcciones de memoria**

30 Prioridad:

23.06.2010 US 821170

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
11.11.2013

73 Titular/es:

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

72 Inventor/es:

**CRADDOCK, DAVID;
GREGG, THOMAS;
GREINER, DAN y
LAIS, ERIC NORMAN**

74 Agente/Representante:

DE ELZABURU MÁRQUEZ, Alberto

ES 2 428 822 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

Traducción de direcciones de entrada/salida a direcciones de memoria

Antecedentes

5 Esta invención se refiere, en general, a traducir direcciones en un entorno informático, y en particular, a traducir una dirección de entrada/salida a una dirección de memoria utilizable en el acceso a memoria del sistema del entorno.

10 La memoria del sistema es accesible mediante peticiones de lectura y escritura. Estas peticiones pueden llegar de diversos componentes de un entorno informático, incluyendo unidades centrales de proceso, así como adaptadores. Cada petición incluye una dirección que va a ser usada para acceder a la memoria del sistema. Esta dirección, no obstante, típicamente no tiene una correspondencia uno a uno con una ubicación física en la memoria del sistema. Por lo tanto, se realiza la traducción de dirección.

15 La traducción de dirección se usa para traducir una dirección que se proporciona en una forma no directamente utilizable en el acceso a memoria del sistema a otra forma que es directamente utilizable en el acceso a una ubicación física en una memoria del sistema. Por ejemplo, una dirección virtual incluida en una petición proporcionada por una unidad central de proceso se traduce a una dirección real o absoluta en la memoria del sistema. Como ejemplo adicional, se puede traducir una dirección de Interconexión de Componentes Periféricos (PCI) proporcionada en una petición desde un adaptador a una dirección absoluta en la memoria del sistema.

20 La Publicación de EE.UU. N° 2008/0168208 A1, publicada el 10 de julio de 2008, Gregg, "I/O Adapter LPAR Isolation In A Hypertransport Environment With Assigned Memory Space Indexing A TVT Via Unit Ids" describe un sistema de procesamiento de datos y método de aislamiento de una pluralidad de adaptadores de I/O en el sistema. El sistema de procesamiento de datos también comprende un conjunto de procesadores, un puente central, y un canal principal del sistema que conecta el conjunto de procesadores y el puente central. Cada uno de los adaptadores de I/O tiene un ID respectivo y envía comandos al puente central que incluyen uno o más de los ID de los adaptadores de I/O. En la realización preferida, estos ID son ID de Unidad definida de Hiper Transporte, y los comandos emitidos por los Adaptadores de I/O incluyen un campo de ID de Unidad que contiene uno o más de los ID de Unidad de los Adaptadores de I/O. Los ID de Unidad cada uno se usa para indexar un TVT para identificar espacios de memoria del sistema únicos e independientes.

30 La Publicación de EE.UU. N° 2009/0182966 A1, publicada el 16 de julio de 2009, Greiner et al., "Dynamic Address Translation with Frame Management" describe una función de gestión de tramas definida para una arquitectura de máquina de un sistema informático. En una realización, se obtiene una instrucción de gestión de tramas que identifica un primer y segundo registro general. El primer registro general contiene un campo de gestión de tramas que tiene un campo de clave con bits de protección de acceso y una indicación de tamaño de bloque. Si la indicación de tamaño de bloque indica un bloque grande entonces se obtiene una dirección de operando de un bloque grande de datos a partir del segundo registro general. El bloque grande de datos tiene una pluralidad de bloques pequeños cada uno de los cuales está asociado con una clave de almacenamiento correspondiente que tiene una pluralidad de bits de protección de acceso de clave de almacenamiento. Si la indicación de tamaño de bloque indica un bloque grande, los bits de protección de acceso de clave de almacenamiento de cada clave de almacenamiento correspondiente de cada bloque pequeño dentro del bloque grande se fijan con los bits de protección de acceso del campo de clave.

40 La US 6.658.521 describe un método y aparato para traducción de direcciones en un canal principal PCI sobre la red infiniband.

La US 2008/0114906 A1 describe eficientemente controlar accesos a memoria correlacionados con memoria especial.

Breve compendio

45 Los defectos de la técnica anterior se superan y se proporcionan ventajas adicionales a través del suministro de un método según la reivindicación 1, y un producto de programa informático y sistema correspondientes.

Se realizan rasgos y ventajas adicionales a través de las técnicas de la presente invención. Otras realizaciones y aspectos de la invención se describen en detalle en la presente memoria y se consideran una parte de la invención reivindicada.

Breve descripción de las diversas vistas de los dibujos

50 Uno o más aspectos de la presente invención se señalan particularmente y se reivindican distintivamente como ejemplos en las reivindicaciones en la conclusión de la especificación. El anteriormente mencionado y otros objetos, rasgos, y ventajas de la invención son evidentes a partir de la siguiente descripción detallada tomada en conjunto con los dibujos anexos en los que:

- 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 la memoria del sistema y centro de I/O de la FIG. 1, según un aspecto de la presente invención;
- 5 La FIG. 3A representa una realización de una visión de conjunto de la lógica para registrar un espacio de dirección DMA (Acceso a Memoria Directo) para un adaptador, según un aspecto de la presente invención;
- La FIG. 3B representa una realización de diversos detalles de registro del espacio de dirección DMA para el adaptador, según un aspecto de la presente invención;
- 10 La FIG. 4 representa una realización de la lógica para procesar una operación DMA, según un aspecto de la presente invención;
- La FIG. 5A representa un ejemplo de los niveles de traducción empleados cuando se usa una dirección entera para indexar en tablas de traducción de direcciones para traducir la dirección y acceder a la página;
- La FIG. 5B representa un ejemplo de niveles de traducción empleados cuando se ignora una parte de la dirección cuando se indexa en las tablas de traducción de direcciones, según un aspecto de la presente invención;
- 15 La FIG. 5C representa ejemplos de varios formatos compatibles de DAT de CPU utilizables según uno o más aspectos de la presente invención;
- La FIG. 5D representa ejemplos de diversos formatos de traducción de direcciones extendidas de I/O utilizables según uno o más aspectos de la presente invención;
- 20 La FIG. 6A representa una realización de una instrucción de Modificar Controles de Función PCI usada según un aspecto de la presente invención;
- La FIG. 6B representa una realización de un campo usado por la instrucción de Modificar Controles de Función PCI de la FIG. 6A, según un aspecto de la presente invención;
- La FIG. 6C representa una realización de otro campo usado por la instrucción de Modificar Controles de Función PCI de la FIG. 6A, según un aspecto de la presente invención;
- 25 La FIG. 6D representa una realización de los contenidos de un bloque de información de función (FIB) usado según un aspecto de la presente invención;
- La FIG. 7 representa una realización de una visión de conjunto de la lógica de la instrucción de Modificar Controles de Función PCI, según un aspecto de la presente invención;
- 30 La FIG. 8 representa una realización de la lógica asociada con un registro de operación de parámetros de traducción de direcciones de I/O que se puede especificar por la instrucción de Modificar Controles de Función PCI, según un aspecto de la presente invención;
- La FIG. 9 representa una realización de la lógica asociada con una baja de registro de operación de parámetros de traducción de direcciones de I/O que se puede especificar por la instrucción de Modificar Controles de Función PCI, según un aspecto de la presente invención;
- 35 La FIG. 10 representa una realización de un producto de programa informático que incorpora uno o más aspectos de la presente invención;
- La FIG. 11 representa una realización de un sistema de ordenador central para incorporar y usar uno o más aspectos de la presente invención;
- 40 La FIG. 12 representa un ejemplo adicional de un sistema informático para incorporar y usar uno o más aspectos de la presente invención;
- La FIG. 13 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. 14 representa una realización de varios elementos de un sistema informático para incorporar y usar uno o más aspectos de la presente invención;
- 45 La FIG. 15A representa una realización de la unidad de ejecución del sistema informático de la FIG. 14 para incorporar y usar uno o más aspectos de la presente invención;
- La FIG. 15B representa una realización de la unidad de ramificación del sistema informático de la FIG. 14 para

incorporar y usar uno o más aspectos de la presente invención;

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

5 La FIG. 16 representa una realización de un sistema informático central emulado para incorporar y usar uno o más aspectos de la presente invención.

Descripción detallada

10 Según un aspecto de la presente invención, se proporciona una capacidad para traducir direcciones en un entorno informático. En un ejemplo, las direcciones que se traducen son direcciones proporcionadas por un adaptador (conocidas en la presente memoria como direcciones de entrada/salida (I/O)), que van a ser traducidas a direcciones utilizables en el acceso a la memoria del sistema. Para realizar la traducción de direcciones, se usa un número de niveles de traducción de direcciones, y el número de niveles se basa en, por ejemplo, un tamaño del espacio de dirección de memoria asignado al adaptador, un tamaño de una o más tablas de traducción de direcciones usado en la traducción, y/o un tamaño de la página (u otra unidad de memoria) a ser accedida.

15 La dirección que se traduce incluye una pluralidad de bits, y en una realización, solamente se usan una parte de esos bits para indexar en las tablas de traducción de direcciones para obtener la dirección traducida. Los otros bits se ignoran para la traducción. Por ejemplo, la dirección incluye bits de orden alto y bits de orden bajo (basados en el tamaño del espacio de dirección asignado). En este ejemplo, los bits de orden bajo se usan para indexar en las tablas de traducción de direcciones, incluyendo la tabla de página, y para indexar en la página en sí misma. Los bits de orden alto (con independencia del valor; es decir, cero o no cero) se ignoran para la traducción y no se usan para indexar en las tablas de traducción de direcciones. Esto reduce el número de niveles de tablas de traducción de direcciones usadas para traducir. (La indicación de bits de orden bajo y orden alto es independiente de cómo se numeran los bits).

20 En un ejemplo, la traducción se realiza mientras que se proporciona protección adecuada para un servidor de clase empresa, tal como un servidor System z®. Como ejemplo, se usa una dirección completa (por ejemplo, la dirección de 64 bit entera) en el acceso a memoria; no obstante, solamente se usa una parte de la dirección para la traducción. Otra parte de la dirección, que se ignora para la traducción, se usa para validación (por ejemplo, una comprobación de intervalo), junto con al menos una parte de la dirección usada para la traducción. Usando solamente una parte de la dirección para la traducción, se minimizan las búsquedas de traducción de direcciones. Esto se consume realizando un número de niveles de traducción en base, por ejemplo, al tamaño del espacio de dirección DMA registrado para el acceso por el adaptador, en lugar del tamaño de la dirección en sí mismo.

25 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.

30 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 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 dinámica de direcciones (DAT).

35 La petición, que incluye la dirección traducida, se recibe por el controlador de memoria 106. En un ejemplo, el controlador de memoria 106 está compuesto de hardware y se usa para mediar para el acceso a la memoria del sistema y para mantener la consistencia de la memoria. Esta mediación 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.

40 En un ejemplo, el adaptador 110 es un adaptador de Interconexión de Componentes Periféricos (PCI) o PCI Expreso (PCIe) que incluye una o más funciones PCI. Una función PCI emite una petición que requiere acceso a la memoria del sistema. La petición se encamina a un centro de entrada/salida 112 (por ejemplo, un centro PCI) a través de uno o más conmutadores (por ejemplo, conmutadores PCIe) 114. En un ejemplo, el centro de entrada/salida se compone

de hardware, que incluye una o más máquinas de estado.

Como se usa en la presente memoria, el término adaptador incluye cualquier tipo de adaptador (por ejemplo, adaptador de almacenamiento, adaptador de red, adaptador de procesamiento, adaptador PCI, adaptador criptográfico, otro tipo de adaptadores de entrada/salida, etc.). En una realización, un adaptador incluye una función de adaptador. 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, adaptador se usa intercambiamente con función de adaptador (por ejemplo, función PCI) a menos que se señale de otro modo.

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 a ser traducida, y de esta manera, el complejo raíz proporciona la dirección a una unidad de protección y traducción de direcciones 118. Esta unidad es, por ejemplo, una unidad hardware que traduce la dirección de I/O a una dirección directamente utilizable para acceder a la memoria del sistema 104, como se describe en más detalle más adelante.

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 de I/O a memoria 120. El controlador de memoria realiza su mediación y reenvía la petición con la dirección traducida a la memoria del sistema en el momento apropiado.

Detalles adicionales con respecto a la memoria del sistema y el centro de entrada/salida se describen con referencia a la FIG. 2. En esta realización, el controlador de memoria no se muestra. No obstante, el centro de I/O se puede acoplar a la memoria del sistema directamente o a través de un controlador de memoria. En un ejemplo, la memoria del sistema 104 incluye uno o más espacios de dirección 200. Un espacio de dirección es una parte particular de la memoria del sistema que ha sido asignada a un componente particular del entorno informático, tal como un adaptador particular. En un ejemplo, el espacio de dirección es accesible mediante acceso directo a memoria (DMA) iniciado por el adaptador, y por lo tanto, el espacio de dirección se conoce en los ejemplos de la presente memoria como un espacio de dirección DMA. No obstante, en otros ejemplos, el acceso directo a memoria no se usa para acceder al espacio de dirección.

Además, en un ejemplo, la memoria del sistema 104 incluye tablas de traducción de direcciones 202 usadas para traducir una dirección de una que no es directamente utilizable para acceder a la memoria del sistema a una que es directamente utilizable. En una realización, hay una o más tablas de traducción de direcciones asignadas a un espacio de dirección DMA, y aquéllas una o más tablas de traducción de direcciones están configuradas en base a, por ejemplo, el tamaño del espacio de dirección al cual está asignado, el tamaño de las tablas de traducción de direcciones en sí mismo, y/o el tamaño de la página (u otra unidad de memoria).

En un ejemplo, hay una jerarquía de tablas de traducción de direcciones. Por ejemplo, como se muestra en la FIG. 2, hay una tabla de primer nivel 202a (por ejemplo, una tabla de segmento) apuntada por un puntero IOAT 218 (descrito más adelante), y una segunda tabla de nivel inferior 202b (por ejemplo, una tabla de página) apuntada por una entrada 206a de la tabla de primer nivel. Uno o más bits de una dirección recibida 204 se usan para indexar en la tabla 202a para localizar una entrada particular 206a, que indica una tabla de nivel inferior 202b particular. Entonces, uno o más de otros bits de una dirección 204 se usan para localizar una entrada particular 206b en esa tabla. En este ejemplo, esa entrada proporciona la dirección usada para localizar la página correcta y se usan bits adicionales en la dirección 204 para localizar una ubicación particular 208 en la página para realizar una transferencia de datos. Es decir, la dirección en la entrada 206b y los bits seleccionados de la dirección PCI recibida 204 se usan para proporcionar la dirección directamente utilizable para acceder a la memoria del sistema. Por ejemplo, la dirección directamente utilizable se forma a partir de una concatenación de bits de orden alto de la dirección en la entrada 206b (por ejemplo, los bits 63:12, en un ejemplo de página de 4k) y bits de orden bajo seleccionados a partir de la dirección PCI recibida (por ejemplo, los bits 11:0 para una página de 4k).

En un ejemplo, es un sistema operativo que asigna un espacio de dirección DMA a un adaptador particular. Esta asignación se realiza a través de un proceso de registro, que causa una iniciación (a través, por ejemplo, de software de confianza) de una entrada de tabla de dispositivo 210 para ese adaptador. La entrada de tabla de dispositivo se localiza en una tabla de dispositivo 211 situada en un centro de I/O 112. Por ejemplo, la tabla de dispositivo 211 se localiza dentro de la unidad de protección y traducción de direcciones del centro de I/O.

En un ejemplo, la entrada de tabla de dispositivo 210 incluye un número de campos, tales como los siguientes:

Formato 212: Este campo incluye una pluralidad de bits para indicar diversa información, incluyendo, por ejemplo, el formato de traducción de direcciones (incluyendo el nivel) de una tabla de nivel superior de las tablas de traducción de direcciones (por ejemplo, en el ejemplo anterior, la tabla de primer nivel);

Tamaño de página 213: Este campo indica un tamaño de una página (u otra unidad de memoria) a ser accedida;

Dirección base PCI 213 y límite PCI 216: Estos valores proporcionan un intervalo usado para definir un espacio

de dirección DMA y verificar que una dirección recibida (por ejemplo, una dirección PCI) es válida; y

Puntero de IOAT (Traducción de Direcciones de Entrada/Salida) 218: Este campo incluye un puntero al más alto nivel de la tabla de traducción de direcciones usada para el espacio de dirección DMA.

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

5 En una realización, la entrada de tabla de dispositivo a ser usada en una traducción particular se localiza usando un
 10 identificador solicitante (RID) localizado en una petición emitida por una función PCI 220 asociada con un adaptador
 (y/o por una parte de la dirección). El id solicitante (por ejemplo, un valor de 16 bit que especifica, por ejemplo, un
 número de canal principal, número de dispositivo y número de función) está incluido en la petición, así como la
 dirección de I/O (por ejemplo, una dirección PCIe de 64 bit) a ser usada para acceder a la memoria del sistema. La
 15 petición, incluyendo la dirección RID e I/O, se proporcionan a, por ejemplo, una memoria direccionable de
 contenidos (CAM) 230 a través de, por ejemplo, un conmutador 114, que se usa para proporcionar un valor de
 índice. Por ejemplo, la CAM incluye múltiples entradas, con cada entrada que corresponde a un índice en la tabla de
 dispositivo. Cada entrada CAM incluye el valor de un RID. Si, por ejemplo, el RID recibido encaja con el valor
 contenido en una entrada en la CAM, el índice de tabla de dispositivo correspondiente se usa para localizar la
 20 entrada de tabla de dispositivo. Es decir, la salida de la CAM se usa para indexar en la tabla de dispositivo 211 para
 localizar la entrada de tabla de dispositivo 210. Si no hay coincidencia, el paquete recibido se descarta sin que se
 realice acceso a la memoria del sistema. (En otras realizaciones, una CAM u otra búsqueda no se necesitan y el RID
 se usa como el índice.)

20 Posteriormente, los campos dentro de la entrada de tabla de dispositivo se usan para asegurar la validez de la
 dirección y la configuración de las tablas de traducción de direcciones. Por ejemplo, se comprueba la dirección
 entrante en la petición por el hardware del centro de I/O (por ejemplo, la unidad de protección y traducción de
 direcciones) para asegurar que está dentro de los límites definidos por la dirección base PCI 214 y el límite PCI 216
 almacenados en la entrada de tabla de dispositivo localizada usando el RID de la petición que proporcionó la
 25 dirección. Esto asegura que la dirección está dentro del intervalo registrado previamente y para que el que las tablas
 de traducción de direcciones están configuradas válidamente.

Según un aspecto de la presente invención, en una realización, para traducir una dirección de I/O (es decir, una
 dirección proporcionada por un adaptador u otro componente de un subsistema de I/O) a una dirección de memoria
 del sistema (es decir, una dirección utilizable directamente para acceder a la memoria del sistema), inicialmente, se
 realiza un proceso de registro. Este proceso de registro registra un espacio de dirección particular con un solicitante
 30 específico, tal como un adaptador específico o función de adaptador. Un ejemplo de una visión de conjunto de este
 proceso de registro se describe con referencia a la FIG. 3A.

Inicialmente, un sistema operativo que se ejecuta dentro de una de las unidades centrales de proceso acopladas a la
 memoria del sistema determina un tamaño y ubicación del espacio de dirección al que va a acceder el adaptador,
 PASO 300. En un ejemplo, el tamaño del espacio de dirección se determina por la dirección base PCI y el límite PCI
 35 fijados por el sistema operativo. El sistema operativo determina la base y límite usando uno o más criterios. Por
 ejemplo, si el sistema operativo desea tener direcciones PCI correlacionadas directamente con direcciones virtuales
 de CPU, entonces la base y el límite se fijan como tales. En un ejemplo adicional, si se desea aislamiento adicional
 entre imágenes del sistema operativo y/o adaptadores, entonces las direcciones que se usan se seleccionan para
 proporcionar espacios de direcciones no solapados y disjuntos. La ubicación también se especifica por el sistema
 40 operativo, y se basa, por ejemplo, en las características del adaptador.

A partir de entonces, se crean una o más tablas de traducción de direcciones para cubrir ese espacio de dirección
 DMA, PASO 302. Como ejemplos, las tablas pueden ser compatibles con las tablas de traducción de direcciones de
 CPU o se puede proporcionar un formato único que se soporta por el centro de entrada/salida. En un ejemplo, la
 45 creación incluye construir las tablas y colocar las direcciones apropiadas dentro de las entradas de tabla. Como
 ejemplo, una de las tablas de traducción es una tabla de página de 4k que tiene 512 entradas de 64 bits, y cada
 entrada incluye una dirección de página de 4k compatible con el espacio de dirección asignado.

A partir de entonces, se registra el espacio de dirección DMA para el adaptador, PASO 304, como se describe en
 más detalle con referencia a la FIG. 3B. En este ejemplo, se supone que hay una función PCI por adaptador, y por lo
 tanto, un ID solicitante por adaptador. Esta lógica se realiza, por ejemplo, por una unidad central de proceso
 50 acoplada a la memoria del sistema, en respuesta a una petición del sistema operativo.

Inicialmente, en una realización, se selecciona una entrada de tabla de dispositivo disponible que es para
 corresponder al ID solicitante del adaptador, PASO 310. Es decir, el ID solicitante se usará para localizar una
 entrada de tabla de dispositivo.

Adicionalmente, la dirección base PCI y el límite PCI se almacenan en la entrada de tabla de dispositivo, PASO 312.
 55 Además, el formato de la tabla de traducción de direcciones del nivel más alto está también almacenado en la
 entrada de tabla de dispositivo (por ejemplo, campo de formato), PASO 314, así como el puntero de traducción de
 direcciones de entrada/salida (IOAT) usado para apuntar a la tabla de traducción de direcciones del nivel más alto,

PASO 316. Esto completa el proceso de registro.

En respuesta a realizar el registro, un espacio de dirección DMA y las tablas de traducción de direcciones correspondientes están preparados para uso, así como una entrada de tabla de dispositivo. Detalles con respecto a procesar una petición emitida por un solicitante, tal como un adaptador, para acceder a la memoria del sistema se describen con referencia a la FIG. 4. El procesamiento descrito más adelante se realiza por el centro de I/O. En un ejemplo, es la unidad de protección y traducción de direcciones la que realiza la lógica. En una realización, inicialmente, se recibe una petición DMA en el centro de entrada/salida, PASO 400. Por ejemplo, una función PCI emite una petición que se reenvía al centro de PCI a través, por ejemplo, de un conmutador PCI. Usando el ID solicitante en la petición, se localiza la entrada de tabla de dispositivo apropiada, PASO 402. A partir de entonces, se hace una determinación en cuanto a si la entrada de la tabla de dispositivo es válida, PREGUNTA 404. En un ejemplo, se determina la validez comprobando un bit de validez en la entrada en sí misma. Este bit se fija, por ejemplo, en respuesta a la ejecución de una petición de función de habilitación por el sistema operativo. Si está habilitada, el bit se fija a, por ejemplo, uno (es decir, válida); de otro modo, permanece en cero (es decir, inválida). En un ejemplo adicional, el bit se puede fijar cuando el proceso de registro está completo.

Si la entrada de tabla de dispositivo es inválida, se presenta un error, PASO 405. De otro modo, se hace una determinación adicional en cuanto a si la dirección PCI proporcionada en la petición es menor que la dirección base PCI almacenada en la entrada de tabla de dispositivo, PREGUNTA 406. Si es, entonces la dirección está fuera de un intervalo válido y se proporciona un error, PASO 407. No obstante, si la dirección PCI es mayor o igual a la dirección base, entonces se hace otra determinación en cuanto a si la dirección PCI es mayor que el valor de límite PCI en la entrada de tabla de dispositivo, PREGUNTA 408. Si la dirección PCI es mayor que el límite, entonces una vez más, se presenta un error dado que la dirección está fuera del intervalo válido, PASO 409. No obstante, si la dirección está dentro de un intervalo válido, entonces continúa el procesamiento.

En un ejemplo, el formato proporcionado en la entrada de tabla de dispositivo se usa para determinar los bits de dirección PCI en la dirección a ser usada para traducción de direcciones, PASO 410. Por ejemplo, si el formato indica que la tabla del nivel superior es una tabla de primer nivel con páginas de 4k, entonces los bits 29:21 de la dirección se usan para indexar en la tabla de primer nivel; los bits 20:12 se usan para indexar en la tabla de página; y los bits 11:0 se usan para indexar en la página de 4k. Los bits usados dependen de cuántos bits se necesitan para indexar en la tabla o página de tamaño dado. Por ejemplo, para una página de 4k con direccionamiento a nivel de byte, se usan 12 bit para direccionar 4096 octetos; y para una tabla de página de 4k con 512 entradas, de 8 octetos cada una, se usan 9 bit para direccionar 512 entradas, etc.

A continuación, el centro PCI extrae la entrada de tabla de traducción de direcciones apropiada, PASO 412. Por ejemplo, inicialmente, la tabla de traducción del nivel más alto se localiza usando el puntero IOAT de la entrada de tabla de dispositivo. Entonces, los bits de la dirección (aquellos después de los bits de orden alto usados para validez o no traducción; por ejemplo, los bits 29:21 en el ejemplo anterior) se usan para localizar la entrada particular dentro de esa tabla.

Entonces se hace una determinación basada, por ejemplo, en el formato proporcionado en la entrada de tabla de dispositivo, en cuanto a si la entrada de traducción de direcciones localizada tiene un formato correcto, PREGUNTA 414. Por ejemplo, el formato en la entrada de tabla de dispositivo se compara con un formato indicado en la entrada de traducción de dirección. Si es igual, entonces el formato en la entrada de tabla de dispositivo es válido. Si no, se proporciona un error, PASO 415; de otro modo, el procesamiento continúa con una determinación en cuanto a si ésta es la última tabla a ser procesada, PREGUNTA 416. Es decir, se hace una determinación en cuanto a si hay otras tablas de traducción de direcciones necesarias para obtener la dirección real o absoluta o si la entrada de tabla del nivel más bajo se ha localizado. Esta determinación se hace en base al formato y tamaño proporcionados de las tablas ya procesadas. Si no es la última tabla, entonces el procesamiento continúa con el PASO 412. De otro modo, el centro de I/O continúa el procesamiento para permitir una extracción o almacenamiento de los datos en la dirección traducida, PASO 418. En un ejemplo, el centro de I/O reenvía la dirección traducida al controlador de memoria, que usa la dirección para extraer o almacenar datos en la ubicación DMA designada por la dirección traducida.

Como se describió anteriormente, según un aspecto de la presente invención, el número de niveles de traducción, y por lo tanto, el número de búsquedas requeridas para realizar traducción se reduce. Esto se consume, por ejemplo, ignorando los bits de orden alto de una dirección durante la traducción y usando solamente los bits de orden bajo para atravesar las tablas de traducción, que se basan, por ejemplo, en el tamaño del espacio de dirección DMA asignado al adaptador. El uso de una dirección parcial frente a la dirección completa se muestra además en los siguientes ejemplos.

Con referencia inicialmente a la FIG. 5A, se representa un ejemplo en el que la dirección entera se usa en la traducción de direcciones/acceso a memoria. Con esta técnica anterior, se necesitan seis niveles de tablas de traducción, incluyendo la tabla de página. El comienzo de la tabla del nivel más alto (por ejemplo, la tabla de 5º nivel en este ejemplo) se apunta a un puntero IOAT, y entonces se usan los bits de la dirección PCI para localizar una entrada en la tabla. Cada entrada de tabla de traducción apunta al inicio de una tabla de traducción de nivel inferior o a una página (por ejemplo, una entrada en la tabla de 5º nivel apunta al inicio de una tabla de 4º nivel, etc.)

En este ejemplo, el espacio de dirección DMA (DMAAS) es de 6M de tamaño, y cada tabla es de 4k bytes que tienen un máximo de 512 entradas de 8 bytes (excepto la tabla de 5º nivel que tiene 128 entradas basadas en el tamaño de la dirección). La dirección es, por ejemplo, de 64 bits: FFFF C000 0009 C600. El comienzo de la tabla de 5º nivel se apunta por el puntero IOAT y los bits 63:57 de la dirección PCI se usan para indexar en la tabla de 5º nivel para localizar el comienzo de la tabla de 4º nivel; los bits 56:48 de la dirección PCI se usan para indexar en la tabla de 4º nivel para localizar el comienzo de la tabla de 3º nivel; los bits 47:39 se usan para indexar en la tabla de 3º nivel para localizar el comienzo de la tabla de 2º nivel; los bits 38:30 se usan para indexar en la tabla de 2º nivel para localizar el comienzo de la tabla de 1º nivel; los bits 29:21 se usan para indexar en la tabla de 1º nivel para localizar el comienzo de la tabla de página; los bits 20:12 se usan para indexar la tabla de página para localizar el comienzo de la página; y los bits 11:0 se usan para localizar la entrada en la página de 4k. De esta manera, en este ejemplo, todos los bits de dirección se usan para traducción/acceso.

Esto se contrapone al ejemplo en la FIG. 5B, en el que el espacio de dirección es del mismo tamaño (por ejemplo, 6M) y la dirección es la misma, pero la técnica de traducción ignora algunos de los bit de dirección durante la traducción. En este ejemplo, se ignoran los bits 63:30 de la dirección para la traducción. El puntero IOAT apunta al comienzo de la tabla de 1º nivel y los bits 29:21 de la dirección PCI se usan para indexar en la tabla de 1º nivel para localizar el comienzo de la tabla de página; los bits 20:12 se usan para indexar en la tabla de página apropiada para localizar el comienzo de la página; y los bit 11:0 se usan para indexar en la página de 4k.

Como se muestra, la tabla de 1º nivel 500 incluye tres entradas 502, cada una que proporciona una dirección a una de las tres tablas de página 504. El número de tablas de página necesarias, y por lo tanto, el número de otras tablas de nivel, depende, por ejemplo, del tamaño del espacio de dirección DMA, el tamaño de las tablas de traducción, y/o el tamaño de las páginas. En este ejemplo, el espacio de dirección DMA es 6M, y cada tabla de página es de 4k, teniendo hasta 512 entradas. Por lo tanto, cada tabla de página puede correlacionar hasta 2M de memoria (4k x 512 entradas). De esta manera, se necesitan tres tablas de página para el espacio de dirección de 6M. La tabla de 1º nivel es capaz de mantener las tres entradas, una para cada tabla de página, y de esta manera, no se necesitan niveles adicionales de tablas de traducción de direcciones, en este ejemplo.

En una realización adicional, pueden haber diferentes formatos de tablas de traducción de direcciones usados para traducción de direcciones, y puede haber variaciones dentro de los formatos. De esta manera, hay variaciones en los bits usados para indexar en una tabla o página particular. Algunos de estos ejemplos se describen con referencia a las FIG. 5C y 5D.

Por ejemplo, un formato es un formato compatible de DAT de CPU en el que las tablas de traducción son compatibles con las tablas de traducción usadas para traducciones DAT de CPU. Puede haber diversos formatos compatibles de DAT de CPU, ejemplos de los cuales se describen con referencia a la FIG. 5C. Como se muestra un formato compatible de DAT de CPU es un formato compatible de DAT de CPU de página de 4k 550, y otro es un formato compatible de DAT de CPU de página de 1M 552, como ejemplos. El número de bits mostrados son el número de bit de dirección usado para indexar en esa página o tabla (o de otro modo localizar una entrada en esa página o tabla). Por ejemplo, se usan 12 bits 554 de una dirección PCI como un desplazamiento de byte en una página de 4k 556; se usan 8 bits 558 como un índice en una tabla de página 560; se usan 11 bits 562 como un índice en una tabla de segmento 564, etc. Localizado bajo la tabla de traducción de direcciones designada está el tamaño máximo del espacio de dirección soportado por esa tabla de traducción de direcciones. Por ejemplo, la tabla de página 560 soporta un espacio de dirección DMA de 1M; la tabla de segmento 564 soporta un espacio de dirección DMA de 2G, etc. En esta figura, así como en la FIG. 5D, K = kilobytes, M = megabytes, G = gigabytes, T = terabytes, P = petabytes, y E = exabytes.

Como se representa, según aumenta el tamaño de la página, disminuye el número de niveles de tablas de traducción. Por ejemplo, para la página de 4k 556, se necesita una tabla de página, pero no se necesita para la página de 1M. Son posibles otros ejemplos o variaciones.

Otro formato de traducción de direcciones es un formato de traducción de direcciones extendido de I/O en el que se usan tablas de traducción de direcciones extendidas. Diversos ejemplos de formatos de traducción de direcciones extendidas de I/O se representan en la FIG. 5D. Por ejemplo, se muestran los siguientes formatos: una tabla de traducción de direcciones de 4k con páginas de 4k 570; tablas de traducción de direcciones de 1M con páginas de 4k 572; y tablas de traducción de direcciones de 1M con páginas de 1M 574. Como con los formatos compatibles de DAT de CPU, el número de bits enumerados son aquellos bits usados para localizar una entrada en la tabla particular. Por ejemplo, en el número de referencia 576, los 12 bits son un desplazamiento en la página de 4k. De manera similar, en el número de referencia 578, los 9 bits se usan para indexar en una tabla de página de I/O. Esta tabla de página de I/O permite un espacio de dirección DMA que tiene un tamaño de 2M. Existen muchos otros ejemplos.

En una implementación particular, para realizar el registro de un espacio de dirección DMA al adaptador, una instrucción conocida como una instrucción de Modificar Controles de Función PCI (MPFC), se usa. Por ejemplo, el sistema operativo determina qué formato de traducción de direcciones desea usar, construye las tablas de traducción de direcciones para ese formato, y entonces emite la instrucción MPFC con ese formato incluido como un operando de la instrucción. En un ejemplo, el formato y otros operandos de la instrucción se incluyen en un bloque

de información de función (descrito más abajo), que es un operando de la instrucción. El bloque de información de función se usa entonces para actualizar la DTE y, en una realización, opcionalmente, una entrada de tabla de función (FTE) que incluye parámetros operacionales del adaptador.

5 Una realización de los detalles relacionados con esta instrucción, y particularmente el proceso de registro, se describen con referencia a las FIG. 6A-9. Con referencia a la FIG. 6ª, una instrucción de Modificar Controles de Función PCI 600 incluye, por ejemplo, un código de operación 602 que indica la instrucción de Modificar Controles de Función PCI; un primer campo 604 que especifica una ubicación en la cual se incluye diversa información con respecto a la función de adaptador para la cual están siendo establecidos parámetros operacionales; y un segundo campo 606 que especifica una ubicación a partir de la cual se extrae un bloque de información de función PCI (FIB).
10 Los contenidos de las ubicaciones designadas por los Campos 1 y 2 se describen además más adelante.

En una realización, el Campo 1 designa un registro general que incluye diversa información. Como se muestra en la FIG. 6B, los contenidos del registro incluyen, por ejemplo, un manejo de función 610 que identifica el manejo de la función de adaptador en nombre de la cual está siendo realizada la instrucción de modificar; un espacio de dirección 612 que designa un espacio de dirección en una memoria del sistema asociada con la función de adaptador designada por el manejo de función; un control de operación 614 que especifica la operación a ser realizada para la función de adaptador; y el estado 616 que proporciona un estado con respecto a la instrucción cuando la instrucción se completa con un código predefinido.
15

En una realización, el manejo de función incluye, por ejemplo, un indicador de habilitación que indica si el manejo está habilitado, 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 manejo de función. Hay un manejo 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 operacionales y/u otra información asociada con esta función de adaptador. Como ejemplo, una entrada de tabla de función incluye:
20

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

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 DMA, peticiones MSI) e información relativa a peticiones asociadas con la función de adaptador (por ejemplo, instrucciones PCI). Cada entrada de tabla de dispositivo está asociada con un espacio de dirección dentro de la memoria del sistema asignado a la función de adaptador. Una función de adaptador puede tener uno o más espacios de dirección dentro de la memoria del sistema asignados a la función de adaptador.
30
35

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

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

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

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

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

45 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.

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 dirección, incluyendo, 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) a la función de adaptador. Especificado en la instrucción está un desplazamiento en el espacio de configuración y un manejo de función usado para localizar la entrada de tabla de función apropiada que incluye el RID. Los microprogramas reciben la instrucción y determinan si 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 en el espacio de configuración de la función de adaptador.
50
55

Como se usa en la presente memoria, los microprogramas incluyen, por ejemplo, el micro código, mili código y/o macro código del procesador. Incluyen, por ejemplo, las instrucciones a nivel de hardware y/o estructuras de datos usadas en la implementación de código máquina de nivel más alto. En una realización, incluyen, por ejemplo, código propietario que se entrega típicamente como micro código que incluye software de confianza y micro código específico al hardware subyacente y controla el acceso del sistema operativo al hardware del sistema.

Registro de Direcciones Base (BAR) (1 a n): Este campo incluye una pluralidad de enteros sin signo, designados como BAR0 – BARn, que están asociados con la función de adaptador especificada originalmente, cuyos valores también se almacenan 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 dirección, es decir si es un espacio de memoria de 64 o 32 bits, o un espacio de I/O de 32 bits, como ejemplos;

En un ejemplo, se usa para accesos a espacio de memoria y/o espacio 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 direcciones base asociado con el espacio de dirección designado en la instrucción para obtener la dirección a ser usada para acceder a la función de adaptador. El identificador de espacio de dirección proporcionado en la instrucción identifica el espacio de dirección dentro de la función de adaptador a ser accedido y el BAR correspondiente a ser usado;

Tamaño 1...n: Este campo incluye una pluralidad de enteros sin signo, designados como TAMAÑO0 – TAMAÑOn. El valor de un campo Tamaño, cuando no es cero, representa el tamaño de cada espacio de dirección con cada entrada correspondiente a un BAR descrito previamente.

Detalles adicionales con respecto a un BAR y Tamaño se describen más adelante.

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

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

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

- a. El campo BARn representa los bits de dirección menos significativos.
- b. El próximo campo BARn+1 consecutivo representa los bits de dirección más significativos.
- c. El campo TAMAÑOn correspondiente no es cero y representa el tamaño del espacio de dirección.
- d. El campo TAMAÑOn+1 correspondiente no es significativo y se almacena como cero.

Información de Encaminamiento Interna: Esta información se usa para realizar encaminamiento particular al adaptador. Incluye, por ejemplo, información de nodo, fragmento de procesador, y direccionamiento de centro, como ejemplos.

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

En un ejemplo, el indicador de ocupado, indicador de estado de error permanente, e indicador de recuperación iniciada se fijan en base a la monitorización realizada por los microprogramas. 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 de canal principal por el procesador (por ejemplo, microprogramas del procesador). Otros campos se pueden fijar en base a la configuración, inicialización, y/o eventos. En otras realizaciones, la entrada de tabla de función puede incluir más, menos o diferente información. La información incluida puede depender de las operaciones soportadas por o habilitadas para la función de adaptador.

Con referencia a la FIG. 6C, en un ejemplo, el Campo 2 designa una dirección lógica 620 de un bloque de información de función (FIB) 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 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. 6D. En una realización, un bloque de información de función 650 incluye los siguientes campos:

Formato 651: Este campo especifica el formato del FIB.

Control de Intercepción 652: Este campo se usa para indicar si la ejecución invitada de instrucciones específicas mediante un invitado de modo paginable provoca intercepción de instrucciones;

5 Indicación de Error 654: Este campo incluye la indicación de estado de error para acceso directo a memoria e interrupciones de adaptador. Cuando el bit se fija (por ejemplo, a 1), uno o más errores han sido detectados mientras que realiza un acceso directo a memoria o interrupción de adaptador para la función de adaptador;

Cargar/Almacenar Bloqueado 656: Este campo indica si están bloqueadas las operaciones de cargar/almacenar;

Función PCI Válida 658; Este campo incluye un control de habilitación para la función de adaptador. Cuando el bit se fija (por ejemplo, a 1), la función de adaptador se considera que está habilitada para operaciones de I/O;

10 Espacio de Dirección Registrado 660: Este campo incluye un control de habilitación de acceso directo a memoria para una función de adaptador. Cuando el campo se fija (por ejemplo, a 1) el acceso directo a memoria está habilitado;

Tamaño de Página 661: Este campo indica el tamaño de la página u otra unidad de memoria a ser accedida por un acceso a memoria DMA;

15 Dirección Base PCI (PBA) 662: Este campo es una dirección base para un espacio de dirección en la memoria del sistema asignada 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 dirección DMA especificado;

Límite de Dirección PCI (PAL) 664: El campo representa la dirección virtual más alta que se permite a una función de adaptador para acceder dentro del espacio de dirección DMA especificado;

20 Puntero de Traducción de Direcciones de Entrada/Salida (IOAT) 666: El puntero de traducción de direcciones de entrada/salida designa la primera de cualesquiera tablas de traducción usadas por una traducción de direcciones virtuales PCI, o puede designar directamente la dirección absoluta de una trama de almacenamiento que es el resultado de la traducción;

25 Subclase de Interrupción (ISC) 668: Este campo incluye la subclase de interrupción usada para presentar interrupciones de adaptador para la función de adaptador;

Número de Interrupciones (NOI) 670: Este campo designa el número de códigos de interrupción distintos aceptados por 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 de desplazamiento de vector de bit de interrupción de adaptador;

30 Dirección de Vector de Bit de Interrupción de Adaptador (AIBV) 672: Este campo especifica una dirección del vector de bit de interrupción de adaptador para la función de adaptador. Este vector se usa en el procesamiento de interrupción;

Desplazamiento de Vector de Bit de Interrupción de Adaptador 674: Este campo especifica el desplazamiento del primer bit de vector de bit de interrupción de adaptador para la función de adaptador;

35 Dirección de Bit de Resumen de Interrupción de Adaptador (AISB) 676: Este campo proporciona una dirección que designa el bit de resumen de interrupción de adaptador, que se usa opcionalmente en el procesamiento de interrupción;

Desplazamiento de Bit de Resumen de Interrupción de Adaptador 678: Este campo proporciona el desplazamiento en el vector de bit de resumen de interrupción de adaptador;

40 Dirección de Bloque de Medición de Función (FMB) 680: Este campo proporciona una dirección de un bloque de medición de función usado para recoger mediciones con respecto a la función de adaptador;

Clave de Bloque de Medición de Función 682: Este campo incluye una clave de acceso para acceder al bloque de medición de función;

45 Control de Notificación de Bit de Resumen 684: Este campo indica si hay un vector de bit de resumen que se usa;

Testigo de Autorización de Instrucciones 686: Este campo se usa para determinar si un invitado de modo de almacenamiento paginable está autorizado para ejecutar instrucciones PCI sin intervención del ordenador principal.

En un ejemplo, en la z/Architecture®, un invitado paginable se ejecuta interpretativamente a través de la

instrucción de Ejecución Interpretativa de Inicio (SIE), a nivel 2 de interpretación. Por ejemplo, el hipervisor de partición lógica (LPAR) ejecuta las instrucciones SIE para comenzar la partición lógica en memoria física, fija. Si z/VM® es el sistema operativo en esa partición lógica, emite la instrucción 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 z/VM® usa SIE de nivel 2; y

Formato de traducción de Direcciones 687: Este campo indica un formato seleccionado para traducción de direcciones de la tabla de traducción del nivel más alto a ser usado en la traducción (por ejemplo, una indicación de la tabla del nivel más alto (por ejemplo tabla de segmento, 3ª región, etc.)).

El bloque de información de función designado en la instrucción de Modificar Controles de Función PCI se usa para modificar una entrada de tabla de dispositivo, una entrada de tabla de función y/u otros controles de microprogramas seleccionados 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 microprogramas, se proporcionan ciertos servicios para el adaptador. Estos servicios incluyen, por ejemplo, interrupciones de adaptador; traducciones de direcciones; reinicio de estado de error; reinicio de carga/almacén bloqueado; ajuste de parámetros de medición de función; y ajuste de control de intercepción.

Una realización de la lógica asociada con la instrucción de Modificar Controles de Función PCI se describe con referencia a la FIG. 7. En un ejemplo, la instrucción se emite por un sistema operativo (u otra configuración) y ejecuta por el procesador (por ejemplo, microprogramas) que ejecutan el sistema operativo. En los ejemplos de la presente memoria, la instrucción y las funciones de adaptador están basadas en PCI. No obstante, en otros ejemplos, se puede usar una arquitectura de adaptador diferente y las instrucciones correspondientes.

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 manejo de función PCI; el identificador de espacio de dirección DMA; un control de operación; y una dirección del bloque de información de función.

Con referencia a la FIG. 7, inicialmente, se hace una determinación en cuanto a si la facilidad que permite una instrucción de Modificar Controles de Función PCI está instalada, PREGUNTA 700. Esta determinación se hace, por ejemplo, comprobando un indicador almacenado, por ejemplo, en un bloque de control. Si la facilidad no está instalada, se proporciona una condición de excepción, PASO 702. De otro modo, se hace una determinación en cuanto a si la instrucción fue emitida por un invitado de modo de almacenamiento paginable (u otro invitado), PREGUNTA 704. En caso afirmativo, el sistema operativo central emulará la operación para ese invitado, PASO 706.

De otro modo, se hace una determinación en cuanto a si uno o más de los operandos están alineados, PREGUNTA 708. Por ejemplo, se hace una determinación en cuanto a si la dirección del bloque de información de función está sobre 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 710. De otro modo, se hace una determinación en cuanto a si es accesible el bloque de información de función, PREGUNTA 712. En caso negativo, entonces se proporciona una condición de excepción, PASO 714. De otro modo, se hace una determinación en cuanto a si está habilitado el manejo proporcionado en los operandos de la instrucción de Modificar Controles de Función PCI, PREGUNTA 716. En un ejemplo, se hace esta determinación comprobando un indicador de habilitación en el manejo. Si el manejo no está habilitado, entonces se proporciona una condición de excepción, PASO 718.

Si el manejo está habilitado, entonces el manejo se usa para localizar una entrada de tabla de función, PASO 720. Es decir, al menos una parte del manejo se usa como un índice en la tabla de función para localizar la entrada de tabla de función correspondiente a la función de adaptador para la cual van a ser establecidos parámetros operacionales.

Se hace una determinación en cuanto a si fue encontrada la entrada de tabla de función, PREGUNTA 722. En caso negativo, entonces se proporciona una condición de excepción, PASO 724. De otro modo, si la configuración que emite la instrucción es un invitado, PREGUNTA 726, entonces se proporciona una condición de excepción (por ejemplo, intercepción al ordenador principal), PASO 728. Esta pregunta se puede ignorar si la configuración no es un invitado o se pueden comprobar otras autorizaciones, si se designa.

Se hace entonces una determinación en cuanto a si la función está habilitada, PREGUNTA 730. En un ejemplo, se hace 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 732.

Si la función está habilitada, entonces se hace una determinación en cuanto a si está activa la recuperación, PREGUNTA 734. 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 736. No obstante, si la recuperación no está activa, entonces se hace una determinación adicional en cuanto a si la función está ocupada, PREGUNTA 738. Esta determinación se hace comprobando el indicador de ocupado en la entrada de tabla de función. Si la función está ocupada, entonces se proporciona una condición de ocupado, PASO 740. Con la

condición de ocupado, se puede volver a intentar la instrucción, en lugar de abandonar.

Si la función no está ocupada, entonces se hace una determinación adicional en cuanto a si el formato de bloque de información de función es válido, PREGUNTA 742. Por ejemplo, el campo de formato del FIB se comprueba para determinar si este formato se soporta por el sistema. Si es inválido, entonces se proporciona una condición de excepción, PASO 744. Si el formato de bloque de información de función es válido, entonces se hace una determinación adicional en cuanto a si el control de operación especificado en los operandos de la instrucción es válido, PREGUNTA 746. Es decir, es el control de operación uno de los controles de operación especificados para esta instrucción. Si es inválido, entonces se proporciona una condición de excepción, PASO 748. No obstante, si el control de operación es válido, entonces el procesamiento continúa con el control de operación específico que se especifica.

Un control de operación que se puede especificar es un registro operación de parámetros de traducción de direcciones de I/O usado en el control de traducciones de direcciones para un adaptador. Con esta operación, los parámetros de función PCI relevantes a la traducción de direcciones de I/O se fijan en la DTE, FTE y/u otra ubicación a partir de los parámetros apropiados del FIB, que es un operando para la instrucción. Estos parámetros incluyen, por ejemplo, la dirección base PCI; el límite de dirección PCI (también conocido como, límite PCI o límite); el formato de traducción de direcciones; el tamaño de página; y el puntero de traducción de direcciones de I/O, que son operandos para esta operación. Hay también operandos implicados, incluyendo una dirección DMA de inicio (SDMA) y una dirección DMA de terminación (EDMA), que están almacenadas en una ubicación accesible para el procesador que ejecuta la instrucción.

Una realización de la lógica para establecer los parámetros operacionales para traducción de direcciones de I/O se describe con referencia a la FIG. 8. Inicialmente, se hace una determinación en cuanto a si la dirección base PCI en el FIB es mayor que el límite PCI en el FIB, PREGUNTA 800. Si la comparación de la dirección base y el límite indican que la dirección base es mayor que el límite, entonces se reconoce una condición de excepción, PASO 802. No obstante, si la dirección base es menor o igual al límite, entonces se hace una determinación adicional en cuanto a si el formato de traducción de dirección y el tamaño de página son válidos, PREGUNTA 804. Si son inválidos, entonces se proporciona una condición de excepción, PASO 806. No obstante, si son válidos, entonces se hace una determinación adicional en cuanto a si el tamaño del espacio de dirección (basado en la dirección base y límite) excede la capacidad de traducción, PREGUNTA 808. En un ejemplo, el tamaño del espacio de dirección se compara con la máxima capacidad de traducción de dirección posible en base al formato de la tabla de nivel superior. Por ejemplo, si la tabla de nivel superior es una tabla de segmento compatible de DAT, la máxima capacidad de traducción es 2 Gbytes.

Si el tamaño del espacio de dirección excede la capacidad de traducción, entonces se proporciona una condición de excepción, PASO 810. De otro modo, se hace una determinación adicional en cuanto a si la dirección base es menor que la dirección DMA de inicio, PREGUNTA 812. Si es así, entonces se proporciona una condición de excepción, PASO 814. De otro modo, se hace otra determinación en cuanto a si el límite de dirección es mayor que la dirección de DMA de terminación, PREGUNTA 816. Si es así, entonces se proporciona una condición de excepción, PASO 818. En un ejemplo, la dirección DMA de inicio y la dirección DMA de terminación están basadas en una política de sistema amplia.

A partir de entonces, se hace una determinación en cuanto a si suficientes recursos, si es que hacen falta, están disponibles para realizar una traducción de direcciones de I/O, PREGUNTA 820. En caso negativo, entonces se proporciona una condición de excepción, PASO 822. De otro modo, se hace una determinación adicional en cuanto a si los parámetros de traducción de direcciones de I/O ya han sido registrados en la FTE y DTE, PREGUNTA 824. Esto se determina comprobando los valores de los parámetros en la FTE/DTE. Por ejemplo, si los valores en la FTE/DTE son cero u otro valor definido, entonces no se ha realizado el registro. Para localizar la FTE, se usa el manejo proporcionado en la instrucción, y para localizar la DTE, se usa un índice de dispositivo en la FTE.

Si la función de adaptador ya se ha registrado para traducción de direcciones, entonces se proporciona una condición de excepción, PASO 826. En caso negativo, entonces se hace una determinación en cuanto a si el espacio de dirección DMA que se especifica es válido (es decir, es un espacio de dirección para el que se ha habilitado una DTE), PREGUNTA 828. En caso negativo, entonces se proporciona una condición de excepción, PASO 830. Si todas las comprobaciones son exitosas, entonces los parámetros de traducción se colocan en la entrada de tabla de dispositivo y opcionalmente, en la entrada de tabla de función correspondiente (u otra ubicación designada), PASO 832. Por ejemplo, los parámetros de función PCI relevantes para la traducción de direcciones de I/O se copian desde el bloque de información de función y colocan en la DTE/FTE. Estos parámetros incluyen, por ejemplo, la dirección base PCI, el límite de dirección PCI, el formato de traducción, el tamaño de página, y el puntero de traducción de direcciones de I/O. Esta operación habilita accesos DMA al espacio de dirección DMA especificado. Ello permite la traducción de direcciones de I/O para la función de adaptador.

Otro control de operación que se puede especificar por la instrucción de Modificar Controles de Función PCI es una baja de registro de operación de parámetros de traducción de direcciones de I/O, un ejemplo de la cual se describe con referencia a la FIG. 9. Con esta operación, los parámetros de función relevantes para traducción de direcciones de I/O se reinician a cero. Esta operación deshabilita accesos DMA al espacio de dirección DMA especificado y hace

una purga de entradas de almacenador temporal de traducción anticipada de I/O para ese espacio de dirección DMA. Ello deshabilita la traducción de direcciones.

Con referencia a la FIG. 9, en una realización, se hace una determinación en cuanto a si los parámetros de traducción de dirección no están registrados, PREGUNTA 900. En un ejemplo, se hace esta determinación comprobando los valores de los parámetros apropiados en la FTE o DTE. Si esos campos son cero o algún valor especificado, no se registran. Por lo tanto, se proporciona una condición de excepción, PASO 902. Si están registrados, entonces se hace una determinación en cuanto a si el espacio de dirección DMA es válido, PREGUNTA 904. Si es inválido, entonces se proporciona una condición de excepción, PASO 906. Si el espacio de dirección DMA es válido, entonces los parámetros de traducción en la entrada de tabla de dispositivo y opcionalmente, en la entrada de tabla de función se borran, PASO 908.

Descrito en detalle anteriormente está un mecanismo eficiente para traducir una dirección de entrada/salida proporcionada por un adaptador a una dirección de memoria del sistema. En un ejemplo, la dirección PCI completa (por ejemplo, la dirección de 64 bits entera) se usa en el acceso a memoria; no obstante, las búsquedas de traducción de dirección se minimizan usando solamente una parte de la dirección para la traducción. Mientras que el uso de la dirección completa proporciona protección añadida, el uso de, por ejemplo, solamente los bits de orden inferior para traducción permite una traducción más eficiente por medio de menos niveles de búsqueda de traducción. Esto permite flexibilidad del sistema operativo en el uso de direcciones que pueden coexistir con o ser las mismas que las direcciones virtuales del sistema operativo. Además, permite que las tablas de traducción de direcciones de CPU sean compartidas por adaptadores, mientras que permite a la I/O reducir el número de tablas que necesitan ser exploradas. Además, se proporciona protección adicional en permitir a diferentes adaptadores y/o sistemas operativos usar intervalos de espacio de dirección disjuntos. En las realizaciones descritas en la presente memoria, los adaptadores son adaptadores PCI. PCI, como se usa en la presente memoria, se refiere a cualesquiera adaptadores implementados de acuerdo con una especificación basada en PCI como se define por el Grupo de Interés Especial de Interconexión de Componentes Periféricos (PCI-SIG), incluyendo pero no limitado a, PCI o PCIe. En un ejemplo particular, Interconexión de Componentes Periféricos Expreso (PCIe) es un estándar de interconexión a nivel de componentes que define un protocolo de comunicación bidireccional para transacciones entre adaptadores de I/O y sistemas centrales. Las comunicaciones PCIe están encapsuladas en paquetes según el estándar PCIe para transmisión en un canal principal PCIe. Las transacciones que se originan en adaptadores de I/O y finalizan en sistemas centrales se conocen como transacciones enlazadas ascendentes. Las transacciones que se originan en sistemas centrales y que terminan en adaptadores de I/O se conocen como transacciones enlazadas descendentes. La topología PCIe se basa en enlaces unidireccionales punto a punto que están emparejados (por ejemplo, un enlace enlazado ascendente, un enlace enlazado descendente) para formar el canal principal PCIe. El estándar PCIe se mantiene y publica por el PCI-SIG.

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 hardware enteramente, una realización software enteramente (incluyendo microprogramas, software residente, micro código, etc.) o una realización que combina aspectos software y hardware que todos pueden ser conocidos generalmente en la presente memoria como "circuito", "módulo" o "sistema". Adicionalmente, 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 al respecto.

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 semiconductor, o cualquier combinación adecuada de los anteriormente mencionados. Más ejemplos 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 solamente de lectura (ROM), una memoria solamente de lectura programable y borrable (EPROM o memoria Instantánea), una fibra óptica, una memoria solamente de lectura de disco compacto portátil (CD-ROM), un dispositivo de almacenamiento óptico, un dispositivo de almacenamiento magnético, o cualquier combinación adecuada de los anteriormente mencionados. 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.

Con referencia ahora a la FIG. 10, en un ejemplo, un producto de programa de ordenador 1000 incluye, por ejemplo, uno o más medios de almacenamiento legibles por ordenador 1002 para almacenar medios o lógica de código de programa legible por ordenador 1004 al respecto 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 apropiado, que incluye pero no se limita a inalámbrico, cableado, cable de fibra óptica, RF, etc., o cualquier

combinación adecuada de los anteriormente mencionados.

El código de programa de ordenador para llevar a cabo las operaciones para aspectos de la presente invención se pueden escribir en cualquier combinación de uno o más lenguajes de programación, incluyendo un lenguaje de programación orientado a objeto, tal como Java, Smalltalk, C++ o similares, y lenguajes de programación de procedimiento convencionales, tal como el lenguaje de programación "C", ensamblador o lenguajes de programación similares. El código de programa se puede ejecutar enteramente en el ordenador del usuario, parcialmente en el ordenador del usuario, como un paquete software autónomo, parcialmente en el ordenador del usuario y 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 se puede hacer la conexión a un ordenador externo (por ejemplo, a través de Internet usando un Proveedor de Servicios de Internet).

Aspectos de la presente invención se describirán en la presente memoria con referencia a ilustraciones de diagrama de flujo y/o diagramas de bloques de métodos, aparatos (sistemas) y productos de programa de ordenador según las 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 mediante 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 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.

Estas instrucciones de programa de ordenador se pueden almacenar también 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.

Las instrucciones de programa de ordenador se pueden cargar también en un ordenador, otro aparato de procesamiento de datos programable, u otros dispositivos para hacer que una serie de pasos operacionales 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 de diagrama de flujo y/o diagrama de bloques.

El diagrama de flujo y los diagramas de bloques en las figuras ilustran la arquitectura, funcionalidad, y operación de posibles implementaciones de sistemas, métodos y productos de programa de ordenador según diversas realizaciones de la presente invención. En este sentido, cada bloque en el diagrama de flujo o diagramas de bloques puede representar un módulo, segmento, o parte de código, que comprende una o más instrucciones ejecutables para implementar la(s) función(funciones) lógica(s) especificada(s). También se debería señalar que, en algunas 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 pueden ejecutar algunas veces en el orden inverso, dependiendo de la funcionalidad implicada. Se señalará también 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 puede implementar mediante sistemas basados en hardware de propósito general 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, dar servicio, etc. uno o más aspectos de la presente invención por un proveedor de servicios que ofrece gestión de entornos de clientes. Por ejemplo, el proveedor de servicios puede crear, mantener, soportar, etc. código informático y/o una 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 pago del cliente bajo una suscripción y/o acuerdo de pago, como ejemplos. Adicionalmente o alternativamente, el proveedor de servicios puede recibir pago de la venta de contenido de publicidad a una o más terceras partes.

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

Como 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 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

5 Aunque se describieron anteriormente diversas realizaciones, éstas son solamente ejemplos. Por ejemplo, los entornos informáticos de otras arquitecturas pueden incorporar y usar uno o más aspectos de la presente invención. Como ejemplos, servidores distintos de servidores System z®, tales como servidores Power Systems u otros servidores ofrecidos por International Business Machines Corporation, o servidores de otras compañías pueden incluir, usar y/o beneficiarse de uno o más aspectos de la presente invención. Además, aunque en el ejemplo de la
10 presente memoria, los adaptadores y centro de PCI se consideran una parte del servidor, en otras realizaciones, no tienen necesariamente que ser considerados una parte del servidor, sino que simplemente se pueden considerar 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 se describen tablas de traducción, se puede usar cualquier estructura de datos y la tabla de términos va a incluir todas las estructuras de datos tales. Aún además, aunque los
15 adaptadores están basados en PCI, uno o más aspectos de la presente invención son utilizables con otros adaptadores u otros componentes de I/O. El adaptador y adaptador PCI son sólo ejemplos. Además, se puede usar otro tamaño de espacios de dirección, tablas de direcciones y/o páginas si apartarse de la presente invención. Además, la DTE puede incluir más, menos o diferente información. Aún además, otros tipos de direcciones se pueden traducir usando uno o más aspectos de la presente invención. Muchas otras variaciones son posibles.

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

Dispositivos de Entrada/Salida o I/O (que incluyen, pero no se limitan a, teclados, visualizadores, dispositivos de puntero, 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 de intervención. También se pueden acoplar adaptadores
30 de red al sistema para permitir al sistema de procesamiento de datos llegar a estar acoplado a otros sistemas de procesamiento de datos o impresoras remotas o dispositivos de almacenamiento a través de redes privadas o públicas de intervención. Módems, cable módem, y tarjetas de Internet son sólo unos pocos de los tipos disponibles de adaptadores de red.

Con referencia a la FIG. 11, se representan componentes representativos de un sistema Informático Central 5000 para implementar uno o más aspectos de la presente invención. El ordenador central representativo 5000 comprende una o más CPU 5001 en comunicación con la memoria del sistema (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 o 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
40 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 almacenamiento en caché de traducciones de manera que accesos posteriores al bloque de memoria de ordenador 5002 no requieran 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 a más de una CPU y cachés más pequeñas, más rápidas (de nivel menor) entre la caché grande y cada CPU. En algunas implementaciones, las cachés de nivel inferior se dividen para proporcionar separación de cachés de bajo nivel para extraer instrucciones y acceso a datos. En una realización, una instrucción se extrae desde la memoria 5002 mediante 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 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 ramificación. La instrucción se ejecuta por la unidad de ejecución, accediendo a operandos desde los registros o memoria especificada de instrucciones según se necesite. Si un operado va ser accedido (cargado o almacenado) desde memoria 5002, una unidad de carga/almacén 5005 típicamente maneja el acceso bajo control de la instrucción que se ejecuta. Se pueden ejecutar instrucciones en circuitos hardware o en micro código (microprogramas) interno o mediante una combinación de ambos.
55

Como se señaló, un sistema informático incluye información en almacenamiento local (o principal), así como direccionamiento, protección, y grabación de cambio y referencia. Algunos aspectos de direccionamiento incluyen el formato de direcciones, el concepto de espacios de direcciones, los diversos tipos de direcciones, y la manera en 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
60

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

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

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

El almacenamiento se ve como una cadena horizontal larga de bits. Para la mayoría de operaciones, los accesos a almacenamiento pasan 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 un byte, que es el bloque de construcción básico de todos los formatos de información. Cada ubicación de byte en almacenamiento se identifica por un entero no negativo único, el cual es la dirección de esa ubicación de byte o, simplemente, la dirección de byte. Las ubicaciones de byte adyacentes tienen direcciones consecutivas, comenzando con 0 en la izquierda y procediendo 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 en 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 son algunas veces conocidos 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 bytes. Para operar en bits individuales de un byte en el almacenamiento, se accede al byte entero. Los bits en un byte se numeran de 0 hasta 7, de izquierda a derecha (en, por ejemplo, la z/Architecture®). Los bits en una dirección se puede numerar de 8-31 o de 40-63 para direcciones de 24 bits, o de 1-31 o de 33-63 para direcciones de 31 bits; se numeran de 0-63 para direcciones de 64 bits. Dentro de cualquier otro formato de longitud fija de bytes múltiples, los bits que componen el formato se numeran consecutivamente comenzando desde 0. Para propósitos de detección de error, 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 operando de almacenamiento está implicado 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. Los campos más grandes pueden ser implícitos para algunas instrucciones. Cuando la longitud de un campo de operando de almacenamiento no es implícita sino que se fija explícitamente, el campo se dice que tiene una longitud variable. Los operandos de longitud variable pueden variar en longitud mediante 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 de solamente aquellas ubicaciones de byte que están incluidas en el campo designado, incluso aunque la anchura de la ruta física 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 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 requerimientos de alineamiento de límite.

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

En una realización, la invención se puede poner en práctica mediante software (algunas veces conocido como

- código interno licenciado, microprogramas, micro código, mili código, pico código y similares, cualquiera de los cuales sería consistente con la presente invención). Con referencia a la FIG. 11, se accede típicamente al código de programa software que incorpora la presente invención por el procesador 5001 del sistema central 5000 desde los dispositivos de medios de almacenamiento de largo plazo 5011, tales como una unidad de CD-ROM, unida de cinta o disco duro. El código de programa software se puede realizar 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 de ordenador 5002 o almacenamiento de un sistema informático sobre una red 5010 a otros sistemas informáticos para uso por 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 procesar por el procesador 5001. Las técnicas y métodos para incorporar el código de programa software en memoria, en medios físicos, y/o código software de distribución 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 instantánea, 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 legible típicamente por un circuito de procesamiento preferiblemente en un sistema informático para ejecución por el circuito de procesamiento.
- La FIG. 12 ilustra una estación de trabajo representativa o sistema hardware servidor en el que se puede poner en práctica la presente invención. El sistema 5020 de la FIG. 12 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 (que incluye cualquiera de medios magnéticos, CD, DVD y Memoria Instantánea por ejemplo) o una unidad de cinta por ejemplo. El sistema 5021 podría incluir también un adaptador de interfaz de usuario, que conecta el microprocesador 5026 a través del canal principal a uno o más dispositivos de interfaz, tal como un teclado 5024, un ratón 5023, una impresora/explorador 5030 y/u otros dispositivos de interfaz, que pueden ser cualquier dispositivo de interfaz de usuario, tal 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. Ejemplo de adaptadores de red son canales de comunicaciones, token ring, Ethernet o módems. Alternativamente, el sistema 5021 puede comunicarse usando una interfaz inalámbrica, tal como una tarjeta CDPD (datos de 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 adaptación cliente/servidor con otro ordenador, etc. Todas estas configuraciones, así como el hardware y software de comunicaciones apropiado, son conocidas en la técnica.
- La FIG. 13 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 puede incluir una o más LAN, donde una LAN puede comprender una pluralidad de estaciones de trabajo inteligentes acopladas a un procesador central.
- Con referencia aún a la FIG. 13, las redes pueden incluir también ordenadores o servidores centrales, tales como un ordenador pasarela (servidor cliente 5046) o servidor de aplicaciones (servidor remoto 5048 al que puede acceder un repositorio de datos y también al que 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 interconexión de redes con 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 simultáneamente a la FIG. 12 y la FIG. 13, se puede acceder al código de programación software que puede incorporar la presente invención por el procesador 5026 del sistema 5020 a partir de los medios de almacenamiento de largo plazo 5027, tales como una unidad de CD-ROM o disco duro. El código de programación 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 los usuarios 5050, 5051 desde la memoria o almacenamiento de un sistema informático sobre

una red a 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 se pagina normalmente 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 código de programación 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 o almacena en un medio tangible (que incluye pero no se limita a módulos de memoria electrónicos (RAM), memoria instantánea, 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 en un sistema informático para ejecución por el circuito de procesamiento.

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

Con referencia a la FIG. 14, se representa una realización de procesador ejemplar para el procesador 5026. Típicamente uno o más niveles de caché 5053 se emplean para bloques de memoria de almacenamiento temporal a fin de mejorar el rendimiento del procesador. La caché 5053 es un almacenador temporal de alta velocidad que mantiene líneas de caché de datos de memoria que van a ser usados probablemente. Las líneas caché son de 64, 128 o 256 bytes de datos de memoria. Las cachés separadas se emplean a menudo para almacenamiento en caché de instrucciones que para 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 de procesador se conoce a menudo como una caché. En un sistema de procesador que tiene 4 niveles de caché 5053, el almacenamiento principal 5025 se conoce algunas veces como la caché de nivel 5 (L5) dado que es típicamente más rápida y solamente mantiene una parte 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 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 (conmutador de contexto desde el entorno de programa al entorno de sistema operativo). La PSW del programa mantiene el valor de contador de programa mientras que el programa no está activo, y el contador de programa (en la PSW) del sistema operativo se usa mientras que se ejecuta el sistema operativo. Típicamente, el contador de programa se incrementa en una cantidad igual al número de bytes de la instrucción actual. Las instrucciones RISC (Ordenador de Conjunto de Instrucciones Reducido) son típicamente de longitud fija mientras que las instrucciones CISC (Ordenador de Conjunto 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 conmutador de contexto o una operación de toma de ramificación de una instrucción de ramificación por ejemplo. En una operación de conmutador de contexto, el valor de contador de programa actual se guarda en la palabra de estado de 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 ramificación se realiza a fin de permitir al programa tomar decisiones o hacer bucle dentro del programa cargando el resultado de la instrucción de ramificación en el contador de programa 5061.

Típicamente se emplea una unidad de extracción de instrucciones 5055 para extraer instrucciones en nombre del procesador 5026. La unidad de extracción o bien extrae "próximas instrucciones secuenciales", instrucciones objetivo de instrucciones de toma de ramificación, o primeras instrucciones de un programa que sigue a un conmutador de contexto. Las unidades de extracción de Instrucciones modernas a menudo emplean técnicas de extracción previas para extraer previamente de manera especulativa instrucciones en base a la probabilidad de que las instrucciones de extracción previas pudieran ser usadas. Por ejemplo, una unidad de extracción puede extraer 16 bytes de instrucción que incluye la siguiente instrucción secuencial y bytes adicionales de instrucciones secuenciales adicionales.

Las instrucciones de extracción entonces se ejecutan por el procesador 5026. En una realización, la(s) instrucción(es) extraída(s) se pasa(n) 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 apropiadas 5057, 5058, 5060. Una unidad de ejecución 5057 recibirá típicamente información acerca de las instrucciones aritméticas decodificadas a partir de la unidad de extracción de instrucciones 5055 y realizará operaciones aritméticas sobre operandos según el código de operación de la instrucción. Se proporcionan operandos a la unidad de ejecución 5057 preferiblemente o bien desde la memoria 5025, registros de arquitectura 5059 o bien a partir de 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, 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. 15A, una unidad de ejecución 5057 puede comunicar con los registros generales de arquitectura 5059, una unidad de decodificación/despacho 5056, una unidad de almacén de carga 5060, y otras unidades de procesador 5065 por medio de la lógica de interfaz 5071. Una unidad de ejecución 5057 puede emplear varios circuitos de registro 5067, 5068, 5069 para mantener información en la que operará la unidad de lógica aritmética (ALU) 5066. La ALU realiza operaciones aritméticas tales como suma, resta, multiplicación y división así como funciones lógicas tales como y, o y o exclusiva (XOR), rotación y cambio. Preferiblemente la ALU soporta operaciones especializadas que son dependientes del diseño. Otros circuitos pueden proporcionar otras instalaciones de arquitectura 5072 que incluyen códigos de condición y lógica de soporte de recuperación 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 adaptaciones de unidades de procesador, la presente descripción prevé solamente proporcionar una comprensión representativa de una realización.

Una instrucción ADD 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, se puede ejecutar una instrucción ADD 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 preferiblemente utiliza una Unidad de Lógica Aritmética (ALU) 5066 que es capaz de realizar una variedad de funciones lógicas tales como Cambiar, Rotar, Y, O y XOR así como una variedad de funciones algebraicas que incluyen cualquier suma, resta, multiplicación, división. 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 marcados pueden ser signo y magnitud, complementos de 1 o complementos de 2 dependiendo de la arquitectura. Un número de complemento de 2 es ventajoso porque 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 bit define una dirección de un bloque de 4.096 bytes y se describe comúnmente como un bloque de 4 Kbyte (Kilobyte), por ejemplo.

Con referencia a la FIG. 15B, la información de instrucción de ramificación para ejecutar una instrucción de ramificación se envía típicamente a una unidad de ramificación 5058 que a menudo emplea un algoritmo de predicción de ramificación tal como una tabla de historia de ramificación 5082 para predecir la salida de la ramificación antes de que estén completas otras operaciones condicionales. El objetivo de la instrucción de ramificación actual se extraerá y especulativamente ejecutará antes de que estén completas las operaciones condicionales. Cuando las operaciones condicionales se completan las instrucciones de ramificación ejecutadas especulativamente o bien se completan o bien descartan en base a las condiciones de la operación condicional y la salida especulada. Una instrucción de ramificación típica puede probar los códigos de condición y se ramifica a una dirección objetivo si los códigos de condición cumplen el requerimiento de ramificación de la instrucción de ramificación, una dirección objetivo se puede calcular en base a varios números que incluyen unos encontrados en campos de registro o un campo inmediato de la instrucción por ejemplo. La unidad de ramificación 5058 puede emplear una ALU 5074 que tiene una pluralidad de circuitos de registro de entrada 5075, 5076, 5077 y un circuito de registro de salida 5080. La unidad de ramificación 5058 puede comunicar con los registros generales 5059, la unidad de despacho de decodificación 5056 u otros circuitos 5073, por ejemplo.

La ejecución de un grupo de instrucciones se puede interrumpir por una variedad de razones que incluyen un conmutador de contexto iniciado por un sistema operativo, una excepción de programa o error que causa un conmutador de contexto, una señal de interrupción de I/O que causa un conmutador de contexto o actividad de procesos múltiples de una pluralidad de programas (en un entorno de procesos múltiples), por ejemplo. Preferiblemente una acción de conmutador 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 a una próxima instrucción a ser ejecutada, códigos de condición, información de traducción de memoria y contenido de registro de arquitectura. Una actividad de conmutador de contexto se puede ejercer por circuitos hardware, programas de aplicaciones, programas de sistema operativo o código de microprogramas (micro código, pico código o código interno licenciado (LIC)) solo o en combinación.

Un procesador accede a operandos según métodos definidos de instrucción. La instrucción puede proporcionar un operando inmediato usando el valor de una parte de la instrucción, puede proporcionar uno o más campos de registro apuntando explícitamente o bien a registros de propósito general o bien 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 operandos. Una ubicación de memoria de un operando se puede proporcionar por un registro, un campo inmediato, o una combinación de registros de un campo inmediato como se ejemplifica por la facilidad de desplazamiento largo z/Architecture® en donde la instrucción define un registro base, un registro de índice y un campo inmediato (campo de desplazamiento) que se añaden 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. 15C, un procesador accede al almacenamiento usando una unidad de carga/almacén 5060. La unidad de carga/almacén 5060 puede realizar una operación de carga obteniendo la dirección del operando objetivo en la memoria 5053 y cargar 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 la memoria 5053 y almacenando los datos obtenidos a partir de un registro 5059 u otra ubicación de memoria 5053 en la ubicación del operando objetivo en la memoria 5053. La unidad de carga/almacén 5060 puede ser especulativa y puede acceder a memoria en una secuencia que está fuera de servicio con respecto a la secuencia de instrucciones, no obstante la unidad de carga/almacén 5060 va a mantener la apariencia para los programas en que las instrucciones fueron ejecutadas en orden. Una unidad de carga/almacén 5060 puede comunicar con registros generales 5059, unidad de decodificación/despacho 5056, interfaz de caché/memoria 5053 u otros elementos 5083 y comprende diversos circuitos de registro, ALU 5085 y la lógica de control 5090 para calcular direcciones de almacenamiento y proporcionar secuenciamiento de canalización para mantener las operaciones en orden. Algunas operaciones pueden estar fuera de servicio pero la unidad de carga/almacén proporciona funcionalidad para hacer que las operaciones fuera de servicio parezcan al programa como que se han realizado en orden, como es bien conocido en la técnica.

Preferiblemente las direcciones que “ve” un programa de aplicaciones son a menudo conocidas como direcciones virtuales. Las direcciones virtuales se conocen algunas veces como “direcciones lógicas” y “direcciones efectivas”. Estas direcciones virtuales son virtuales porque se redirigen a una ubicación de memoria física por una de una variedad de tecnologías de traducción de direcciones dinámicas (DAT) incluyendo, pero no limitadas 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 que comprenden preferiblemente al menos una tabla de segmentos y una tabla de páginas sola 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 con una ubicación de memoria física asociada. Las entradas se crean cuando la DAT traduce una dirección virtual usando las tablas de traducción. El uso posterior de la dirección virtual puede utilizar entonces la entrada del TLB rápido en lugar de los accesos a la tabla de traducción secuencial lenta. El contenido del TLB se puede gestionar por una variedad de algoritmos de sustitución incluyendo LRU (Menos Recientemente usados).

En el caso donde el procesador es un procesador de un sistema de procesador múltiple, cada procesador tiene la responsabilidad de mantener recursos compartidos, tal como I/O, cachés, TLB y memoria, bloqueados entre ellos por coherencia. Típicamente, las tecnologías de “fisgoneo” se utilizarán 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 compartición.

Las unidades de I/O 5054 (FIG. 14) dotan al procesador con medios para unir a dispositivos periféricos incluyendo cinta, disco, impresoras, visualizadores, y redes por ejemplo. Se presentan a menudo unidades de I/O al programa de ordenador por controladores software. En ordenadores centrales, tales como el 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 que 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 los mismos se emula (por

ejemplo, en un sistema informático nativo que tiene un procesador y 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 el modo de emulación, la instrucción u operación específica que se emula se decodifica, y se construye una función de emulación apropiada para implementar la instrucción u operación individual.

En un entorno de emulación, un ordenador central incluye, por ejemplo, una memoria para almacenar instrucciones y datos; una unidad de extracción de instrucciones para extraer instrucciones de la memoria y para proporcionar, opcionalmente almacenamiento temporal local para la instrucción extraída; una unidad de decodificación de instrucciones para recibir las instrucciones extraídas y determinar el tipo de instrucciones que se han extraído; y una unidad de ejecución de instrucciones para ejecutar las instrucciones. La ejecución puede incluir cargar datos en un registro desde la memoria; almacenar datos de vuelta a la memoria desde un registro; o realizar algún tipo de operación aritmética o lógica, como se determina 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.

Más particularmente, en un ordenador central, se usan instrucciones de máquina de arquitectura por programadores, normalmente hoy en día programadores en "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 IBM® de z/Architecture®, o alternativamente en máquinas que ejecutan otras arquitecturas. Se pueden emular en los servidores de ordenador central IBM® existentes y en el futuro y en otras máquinas de IBM® (por ejemplo, servidores Power Systems y Servidores System x®). Se pueden ejecutar en máquinas que ejecutan Linux sobre una amplia variedad de máquinas usando hardware fabricado por IBM®, Intel®, AMDTM, y otros. Junto con la ejecución en ese hardware bajo una z/Architecture®, se puede usar Linux así como máquinas que usan emulación mediante TurboHercules (www.turbohercules.com/), Hercules (www.hercules-390.org/) o FSI (Fundamental Software, Inc) (www.funsoft.com/), donde la ejecución generalmente está en un modo emulación. En modo emulación, se ejecuta software de emulación por un procesador nativo para emular la arquitectura de un procesador emulado.

El procesador nativo típicamente ejecuta software de emulación que comprende o bien microprogramas o bien un sistema operativo nativo para realizar emulación del procesador emulado. Este software de emulación es responsable de extraer y ejecutar instrucciones de la arquitectura de procesador emulada. El software de emulación mantiene un contador de programa emulado para hacer el seguimiento de límites de instrucciones. El software de emulación puede extraer 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. Sin embargo, el software de emulación va a mantener las reglas de arquitectura de la arquitectura de procesador emulada para asegurar los sistemas operativos y aplicaciones escritas para operar el procesador emulado correctamente. Adicionalmente, el software de emulación va a proporcionar recursos identificados por la arquitectura de procesador emulado incluyendo, pero no limitado a, registros de control, registros de propósito general, registros de punto flotante, función de traducción de direcciones dinámica incluyendo tablas de segmentos y tablas de páginas por ejemplo, mecanismos de interrupción, mecanismos de conmutador 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.

Se decodifica una instrucción específica que se emula, y se llama una subrutina para realizar la función de la instrucción individual. Se implementa una función de software de emulación que emula una función de un procesador emulado, por ejemplo, en un controlador o subrutina "C", o algún otro método de proporcionar un controlador para el hardware específico que estará dentro de la experiencia de aquéllos en la técnica después de comprender la descripción de la realización preferida. Diversas patentes de emulación software y hardware incluyendo, pero no limitadas a las Concesiones de Patente de EE.UU. N° 5.551.013, titulada "Mutiprocessor for Hardware Emulation", por Beausoleil et al.; y las Concesiones de Patente de EE.UU. N° 6.009.261, titulada "Preprocessing of Stored Target Routines for Emulating Incompatible Instructions on a Target Processor", por Scalzi et al.; y las Concesiones de Patente de EE.UU. 5.574.873, titulada "Decoding Guest Instruction to Directly Access Emulation Routines that Emulate the Guest Instructions", por Davidian et al.; y las Concesiones 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", por Gorishek et al; y las Concesiones 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", por Lethin et al; y las Concesiones de Patente de EE.UU. N° 5.790.825, titulada "Method for Emulating Guest Instructions on a Host Computer Through Dynamic Recompile of Host Instructions", por Eric Traut; y muchas otras, ilustran una variedad de formas conocidas para lograr emulación de un formato de instrucción de arquitectura para una máquina diferente para una máquina objetivo disponible para los expertos en la técnica.

En la FIG. 16, se proporciona un ejemplo de un sistema informático central emulado 5092 que emula un sistema informático central 5000' de una arquitectura central. En el sistema informático central emulado 5092, el procesador central (CPU) 5091 es un procesador central emulado (o procesador central virtual) y comprende un procesador de

5 emulación 5093 que tiene una arquitectura de conjunto de instrucciones nativo diferente de aquella del procesador 5091 del ordenador central 5000'. El sistema informático central emulado 5092 tiene una memoria 5094 accesible al procesador de emulación 5093. En la realización ejemplo, la memoria 5094 se divide en una parte de memoria de ordenador central 5096 y una parte de rutinas de emulación 5097. La memoria de ordenador central 5096 está disponible para programas del ordenador central emulado 5092 según la arquitectura de ordenador central. El procesador de emulación 5093 ejecuta instrucciones nativas de un conjunto de instrucciones de arquitectura de una arquitectura distinta de aquella del procesador emulado 5091, las instrucciones nativas obtenidas desde la memoria de rutinas de emulación 5097, y puede acceder a una instrucción central para ejecución desde un programa en la memoria de ordenador central 5096 empleando una o más instrucción(instrucciones) obtenida(s) en una secuencia y rutina de acceso/decodificación que puede decodificar la(s) instrucción(instrucciones) central(es) accedida(s) para determinar una rutina de ejecución de instrucciones nativas para emular la función de la instrucción central accedida. Otras facilidades que se definen para la arquitectura del sistema informático central 5000' se puede emular por rutinas de facilidades de arquitectura, incluyendo tales facilidades como registros de propósito general, registros de control, traducción de direcciones dinámica y soporte del subsistema de I/O y caché de procesador, por ejemplo. Las rutinas de emulación también pueden tomar ventaja de 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 puede proporcionar hardware especial y motores de carga para ayudar al procesador 5093 en la emulación de la función del ordenador central 5000'.

20 La terminología usada en la presente memoria es para el propósito de describir realizaciones particulares solamente y no se pretende que sea limitante de la invención. Como se usa en la presente memoria, las formas singulares "un", "una" y "el", "la" se pretende que incluyan las formas plurales también, a menos que el contexto claramente lo indique de otro modo. Se entenderá además que los términos "comprende" y/o "que comprende", cuando se usan en esta especificación, especifican la presencia de rasgos, enteros, pasos, operaciones, elementos, y/o componentes indicados, pero no excluyen la presencia o adición de uno o más de otros rasgos, enteros, pasos, operaciones, elementos, componentes y/o grupos de los mismos.

REIVINDICACIONES

1. Un método de traducción de direcciones en un entorno informático, dicho método que comprende:
 - 5 obtener una dirección a partir de un adaptador (110) a ser traducida a una dirección de memoria directamente utilizable en el acceso a la memoria del sistema del entorno informático, la dirección que comprende una pluralidad de bits, la pluralidad de bits que comprende una primera parte de bits y una segunda parte de bits;
 - recibir un valor de intervalo de direcciones que indica un intervalo de direcciones permitidas, en donde el intervalo se define por una dirección base (214) y un límite (216) situado en una entrada de tabla de dispositivo asociada con el adaptador, la entrada de tabla de dispositivo (210) situada mediante un identificador solicitante situado en una petición emitida por el adaptador;
 - 10 validar la dirección obtenida a partir del adaptador usando al menos la primera parte de bits y el intervalo de direcciones recibido; y
 - convertir la dirección obtenida a partir del adaptador a la dirección de memoria directamente utilizable en el acceso a la memoria del sistema del entorno informático, el método que se caracteriza por la conversión que ignora la primera parte de bits y que usa la segunda parte de bits para obtener información de dirección a partir de uno o más niveles de tablas de traducción de direcciones para realizar la conversión.
2. El método de la reivindicación 1, en donde la primera parte de bits comprende bits de orden alto de la dirección y la segunda parte de bits comprende bits de orden bajo de la dirección, los bits de orden bajo determinados en base a un tamaño de un espacio de dirección asignado que incluye la dirección de memoria.
3. El método de la reivindicación 1, en donde un número de niveles de tablas de traducción de direcciones está basado en a menos uno de un tamaño de un espacio de dirección asignado que incluye la dirección de memoria, un tamaño de una o más tablas de traducción de direcciones a ser usadas en la conversión, y un tamaño de la unidad de memoria accedida por la dirección de memoria.
4. El método de la reivindicación 1, en donde la conversión comprende seleccionar una tabla de traducción de direcciones a ser usada para convertir la dirección, la selección que usa un puntero en una entrada de tabla de dispositivo (210) usada en la conversión.
5. El método de la reivindicación 4, en donde el método además comprende localizar la entrada de la tabla del dispositivo (210), la ubicación que usa al menos uno de un identificador solicitante del adaptador que emite una petición que incluye la dirección a ser traducida o una parte de la dirección.
6. El método de la reivindicación 4, en donde el método además comprende determinar un formato de la tabla de traducción de direcciones seleccionada, la determinación usando un campo de formato de la entrada de la tabla de dispositivo (210).
7. Un producto de programa informático para traducir direcciones en un entorno informático, el producto de programa informático que comprende:
 - 35 un medio de almacenamiento legible por ordenador legible por un circuito de procesamiento e instrucciones de almacenamiento para ejecución por el circuito de procesamiento para realizar un método según cualquiera de las reivindicaciones 1 a 6.
8. Un sistema que comprende medios adaptados para llevar a cabo todos los pasos del método según cualquier reivindicación del método precedente.

40

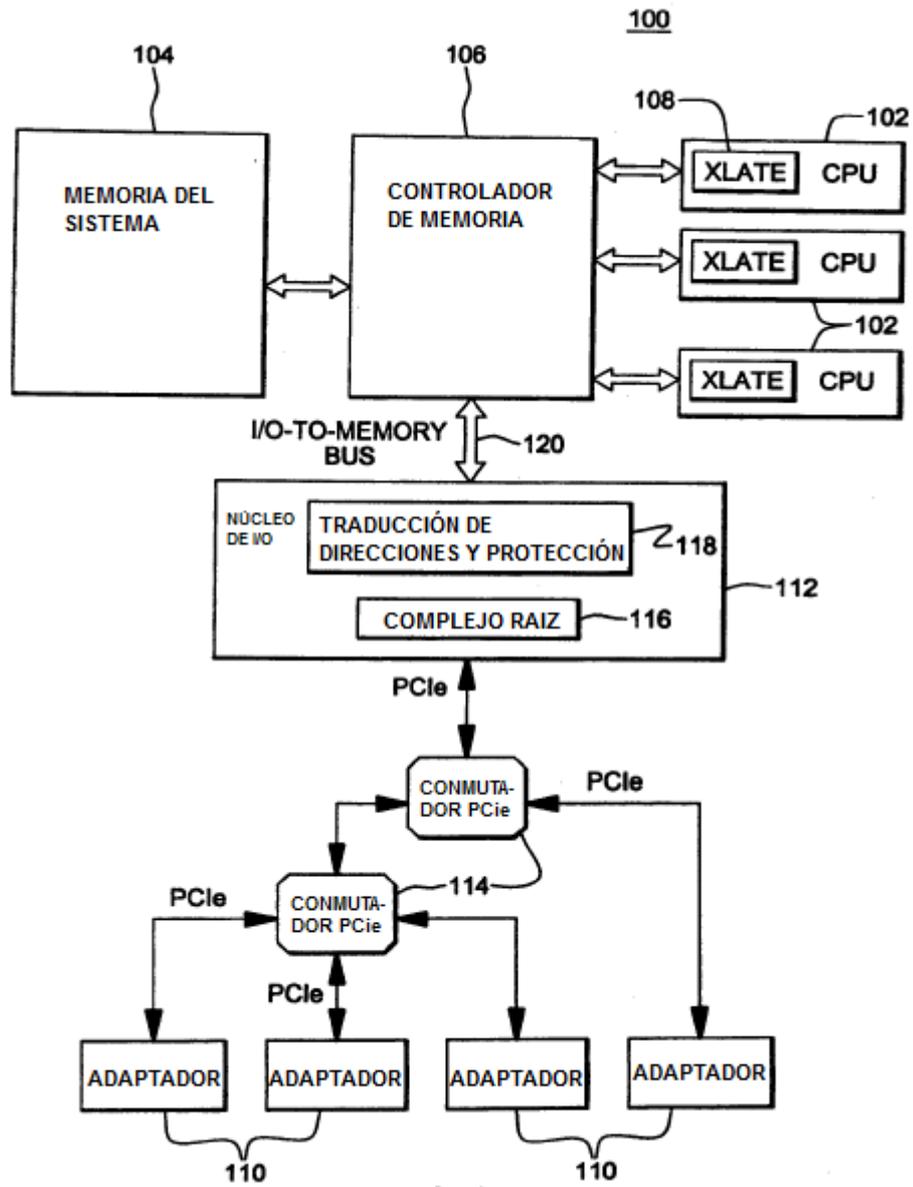


FIG. 1

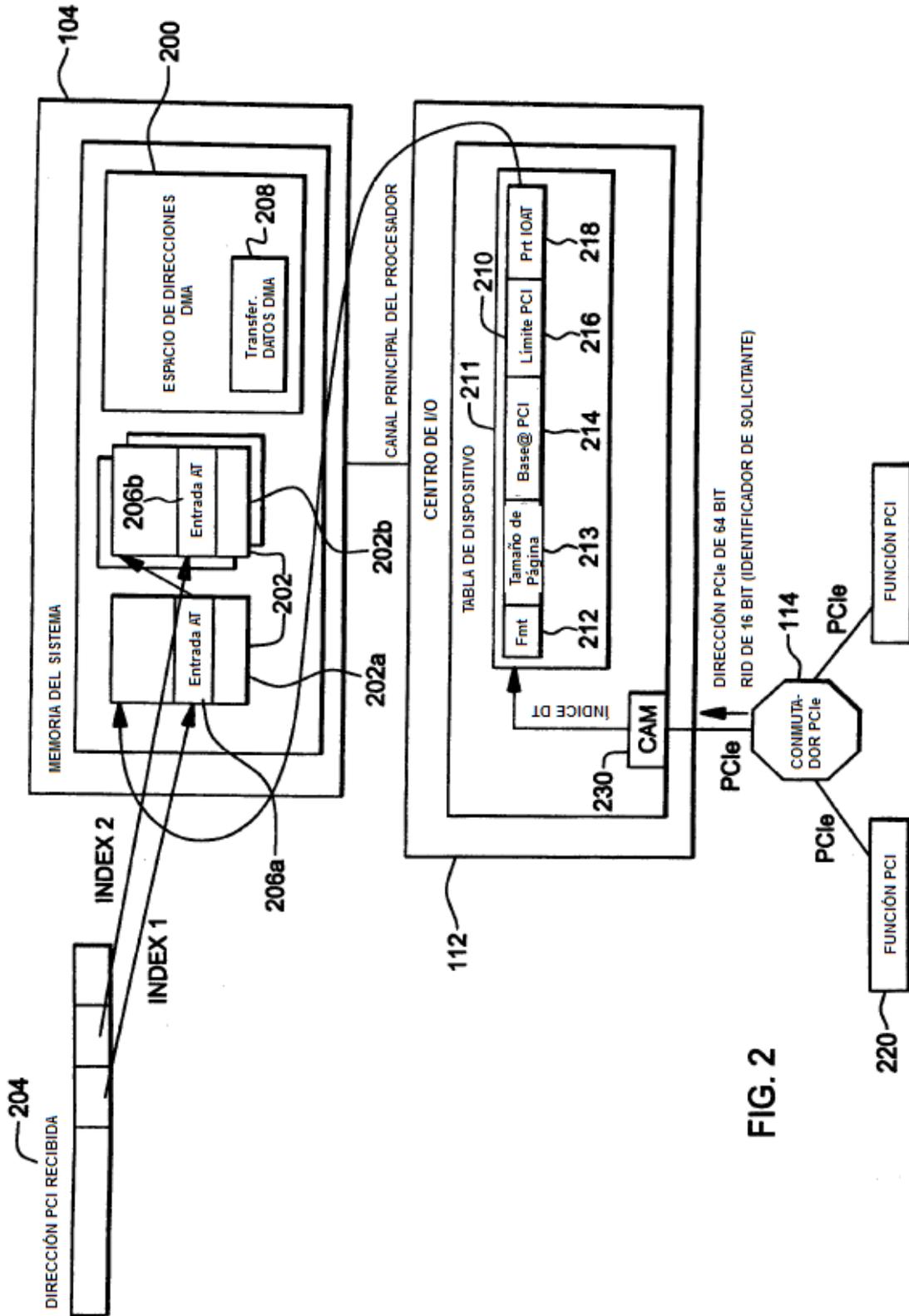


FIG. 2

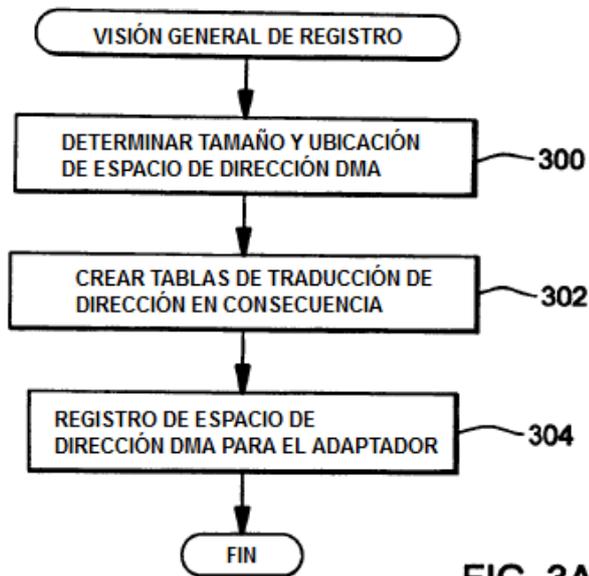


FIG. 3A

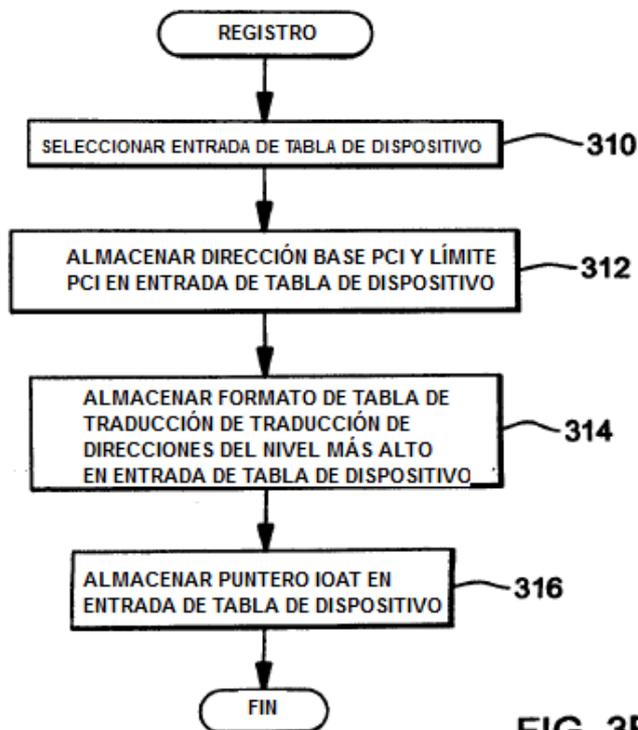


FIG. 3B

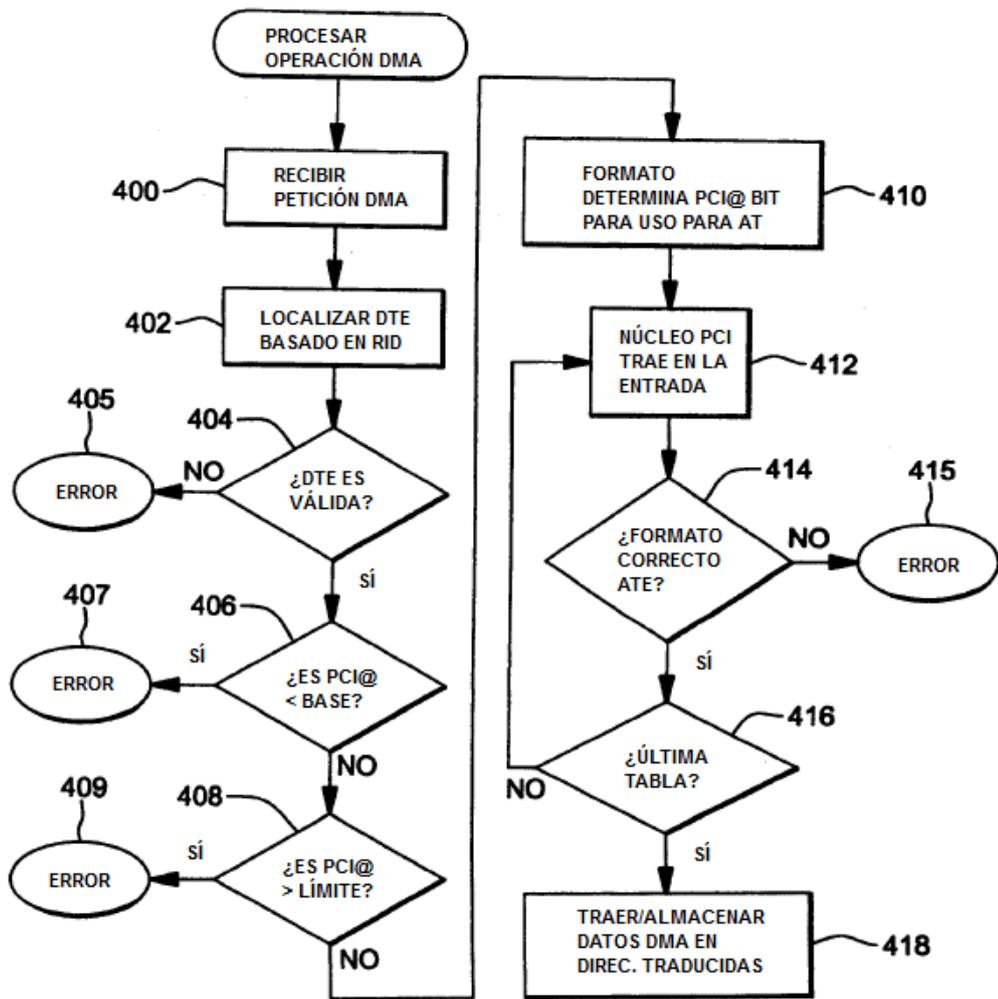


FIG. 4

DMAAS DE 6M DE TAMAÑO
REQUIERE 6 NIVELES DE TABLA INCLUYENDO TABLAS DE PÁGINA (CADA PÁGINA DE 4K CONTIENE 512 ENTRADAS DE 8 BYTES)
DIRECCIÓN PCI PARA LA OPERACIÓN DMA: FFF C000 0009 C600

DIRECCIÓN PCI PARA LA OPERACIÓN DMA: FFF C000 0009 C600

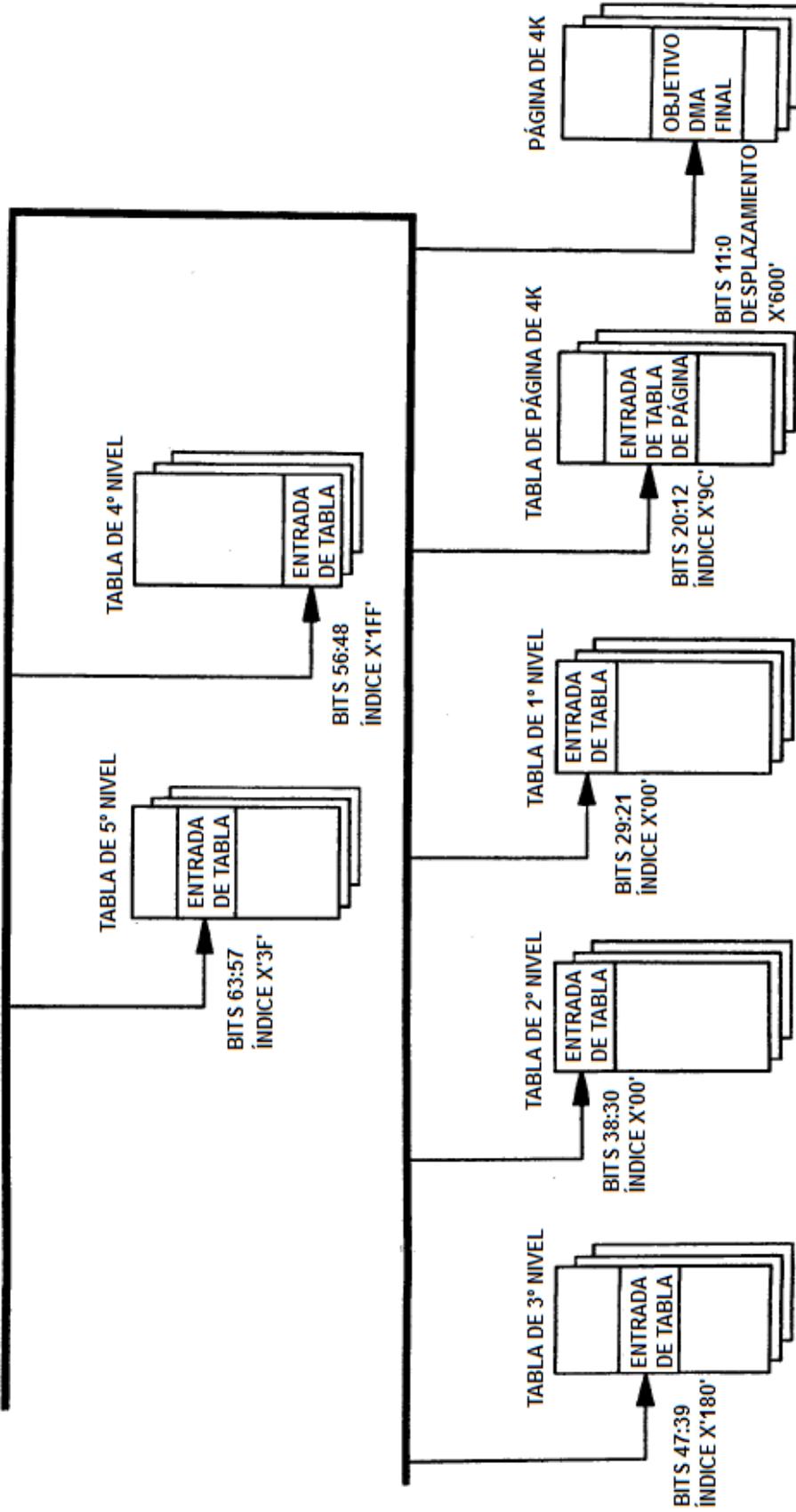


FIG. 5A

DMAAS DE 6M DE TAMAÑO
 REQUIERE 1 TABLA DE 1º NIVEL DE 4K Y 3 TABLAS DE PÁGINA DE 4K (CADA PÁGINA DE 4K CONTIENE 512 ENTRADAS DE 8 BYTES)
 DIRECCIÓN PCI BASE FFFF C000 0000 0000
 DIRECCIÓN DE LÍMITE DE PCI FFFF C000 005F FFFF
 DIRECCIÓN DE PCI PARA LA OPERACIÓN DMA: FFFF C000 0009 C600
 LA COMPROBACIÓN LÍMITE/BASE PCI PERMITE QUE LOS BITS DE DIRECCIÓN PCI 63:30 SEAN IGNORADOS

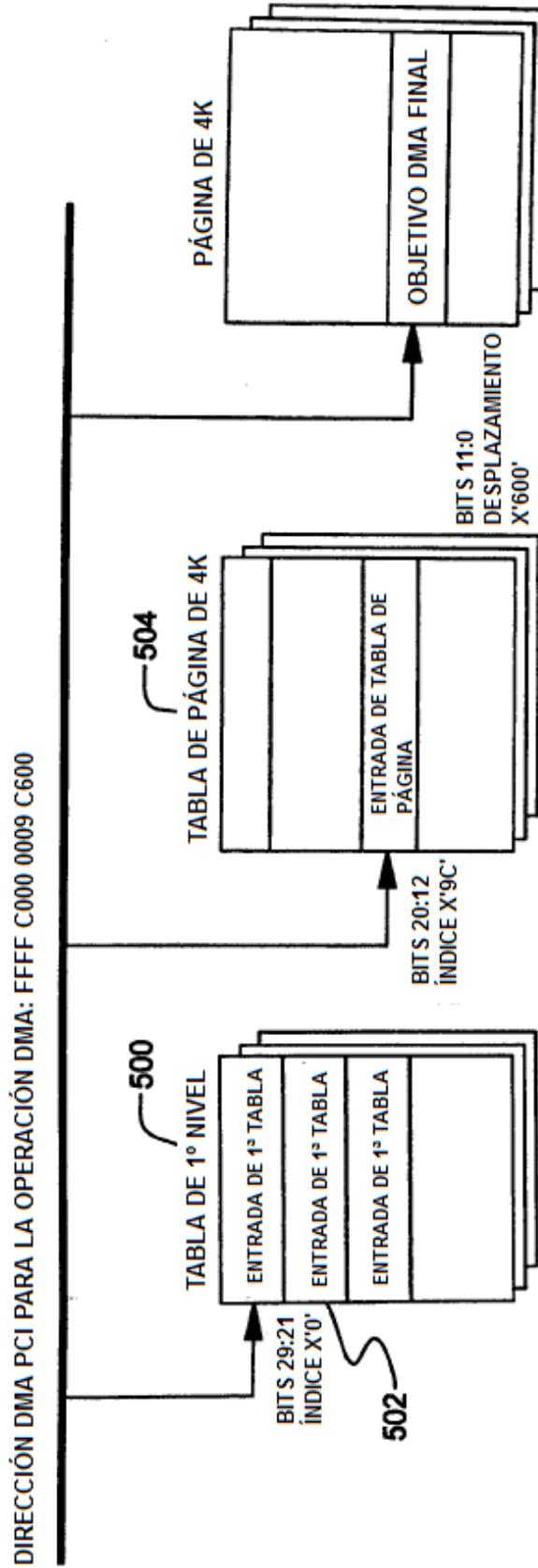


FIG. 5B

FORMATOS COMPATIBLES DE DAT DE CPU

FORMATO	11 BITS	11 BITS	11 BITS	11 BITS	11 BITS	564 11 BITS	562 8 BITS	558 12 BITS	556 12 BITS	554
550 COMP. DAT DE PÁGINA DE 1M	DMAAS DE PRIMERA REGIÓN 16E	DMAAS DE SEGUNDA REGIÓN 8P	DMAAS DE TERCERA REGIÓN 4T	DMAAS DE TERCERA REGIÓN 4T	DMAAS DE TABLA DE SEGMENTO DE 2G	DMAAS DE TABLA DE SEGMENTO DE 2G	DMAAS DE TABLA DE PÁGINA DE 1M	DESPLAZAMIENTO DE BYTE DE 4K		
552 COMP. DAT DE PÁGINA DE 1M	11 BITS	11 BITS	11 BITS	DMAAS DE TERCERA REGIÓN 4T	DMAAS DE TABLA DE SEGMENTO DE 2G	DMAAS DE TERCERA REGIÓN 4T	DESPLAZAMIENTO DE BYTE 1M	20 BITS		

FIG. 5C

FORMATOS DE TRADUCCIÓN DE DIRECCIONES EXTENDIDAS DE I/O

FORMATO	7 BITS	9 BITS	9 BITS	9 BITS	9 BITS	9 BITS	578 9 BITS	576 12 BITS
570 PÁGINA DE 4K 4K EN TABLAS	DMAAS DE 5º NIVEL 16E	DMAAS DE 4º NIVEL 128P	DMAAS DE 3º NIVEL 256T	DMAAS DE 2º NIVEL 512G	DMAAS DE 1º NIVEL 1G	DMAAS DE 1º NIVEL 1G	DMAAS DE IOPT DE 2M	DESPLAZAMIENTO DE BYTE DE 4K
572 PÁGINA DE 4K 1M EN TABLAS	DMAAS DE 2º NIVEL 8E	17 BITS		DMAAS DE 1º NIVEL 64T	DMAAS DE IOPT DE 512M	17 BITS		12 BITS
574 PÁGINA DE 1M 1M EN TABLAS	DMAAS DE 2º NIVEL 16E	DMAAS DE 1º NIVEL 16P	17 BITS		DMAAS DE IOPT DE 128G	17 BITS		20 BITS
	DMAAS DE 2º NIVEL 16E	DMAAS DE 1º NIVEL 16P	17 BITS		DMAAS DE IOPT DE 128G	17 BITS		DESPLAZAMIENTO DE BYTE 1M

FIG. 5D

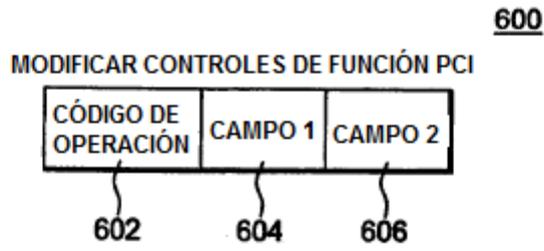


FIG. 6A

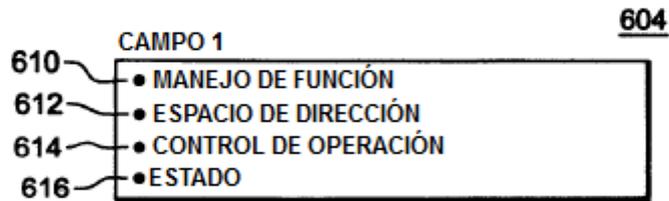


FIG. 6B

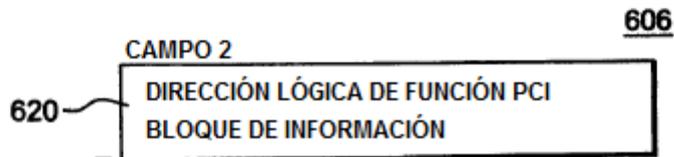


FIG. 6C

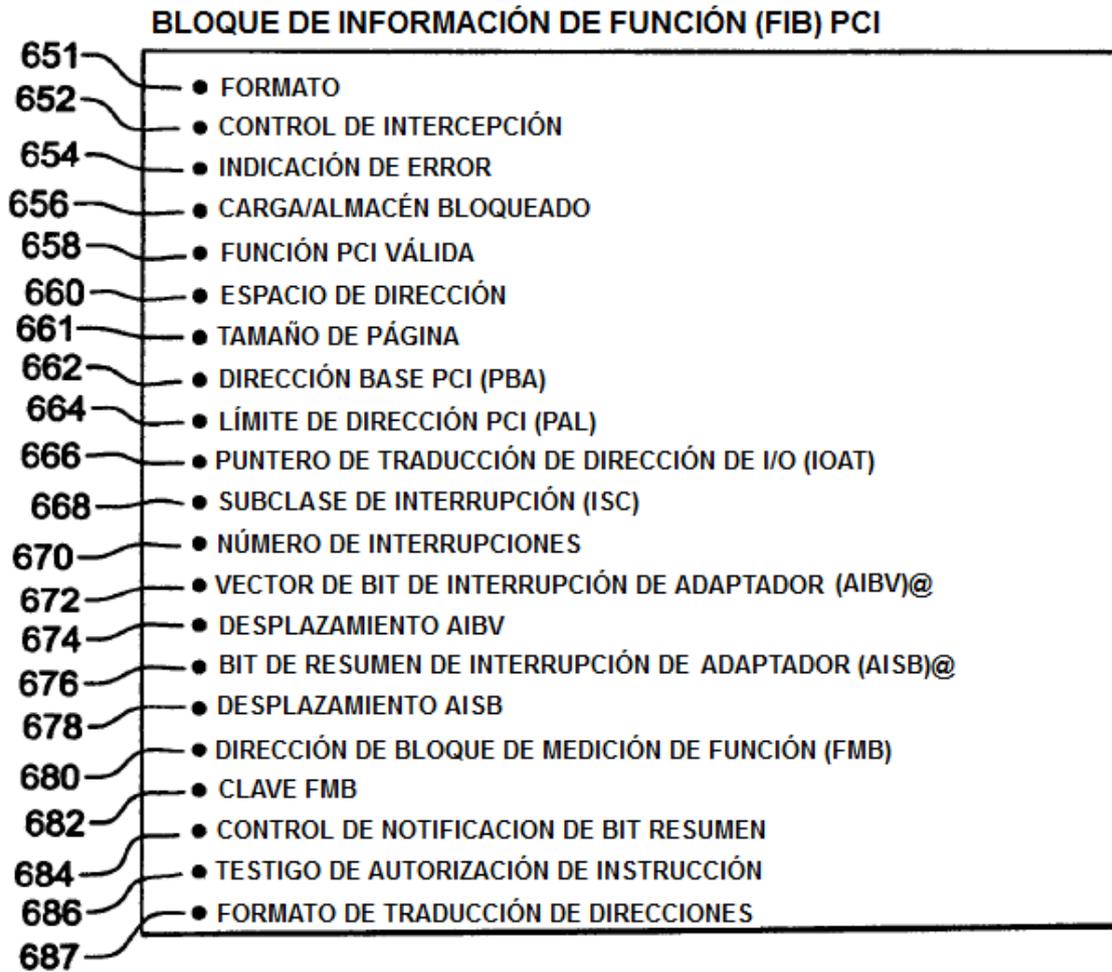


FIG. 6D

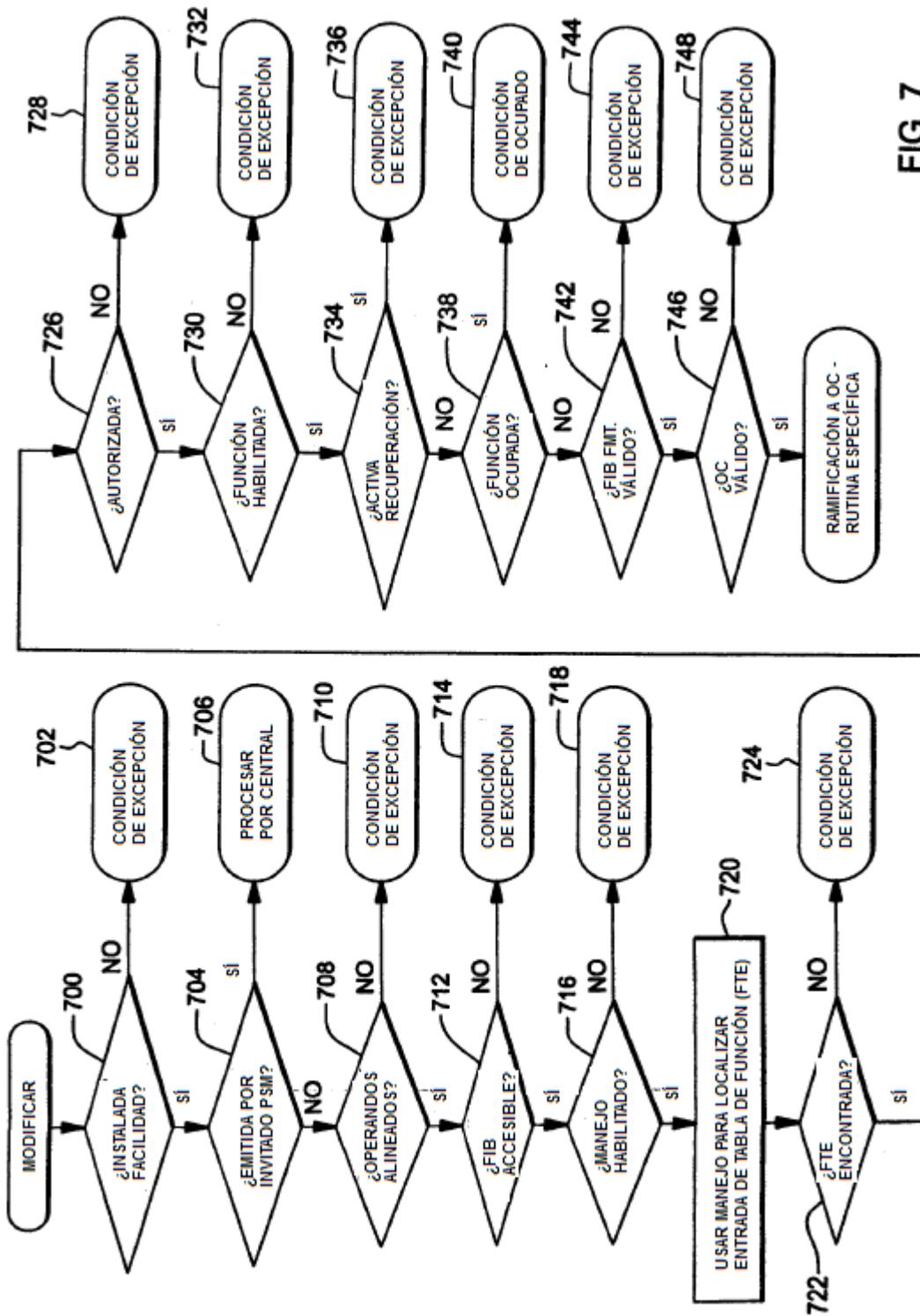


FIG. 7

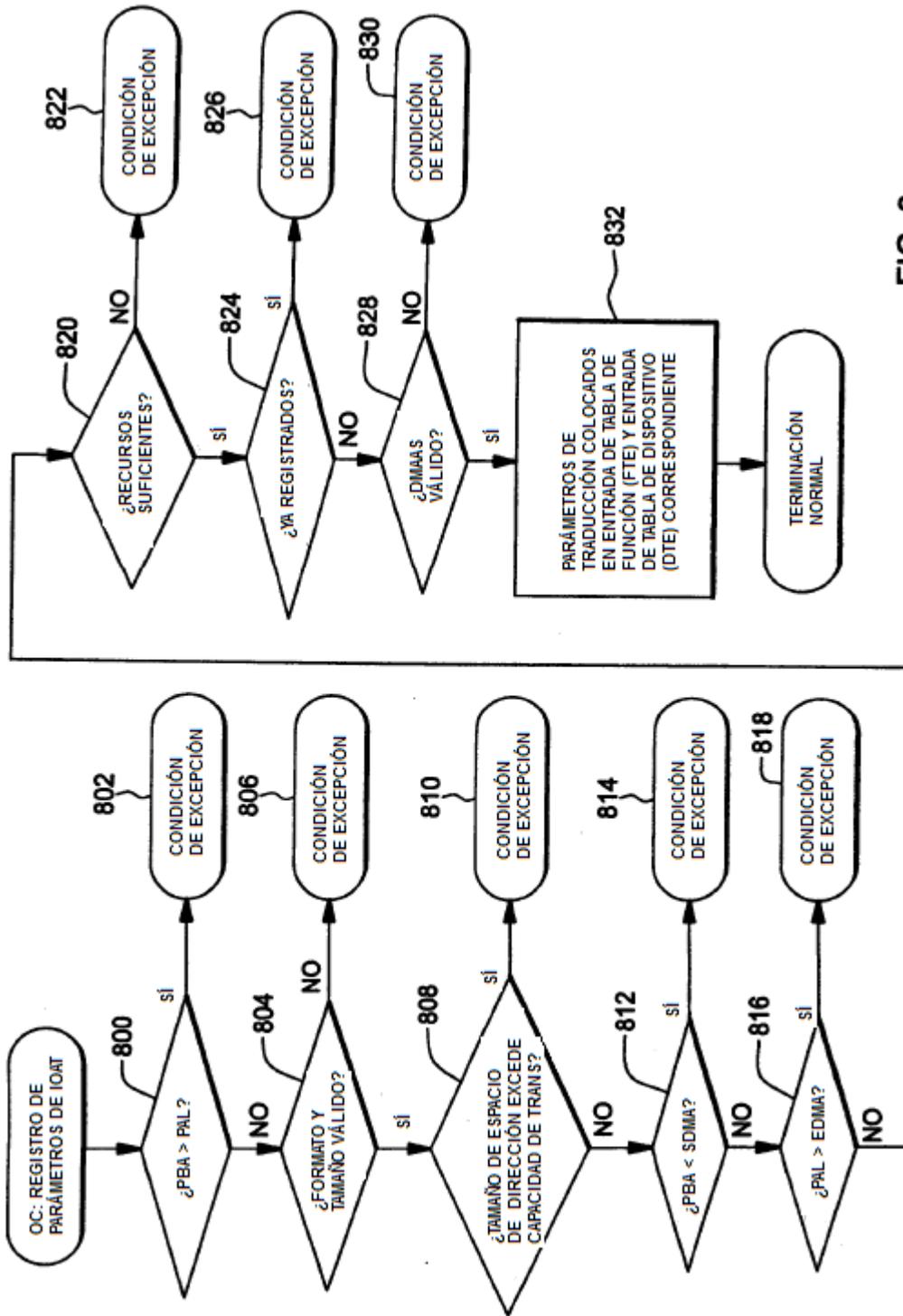


FIG. 8

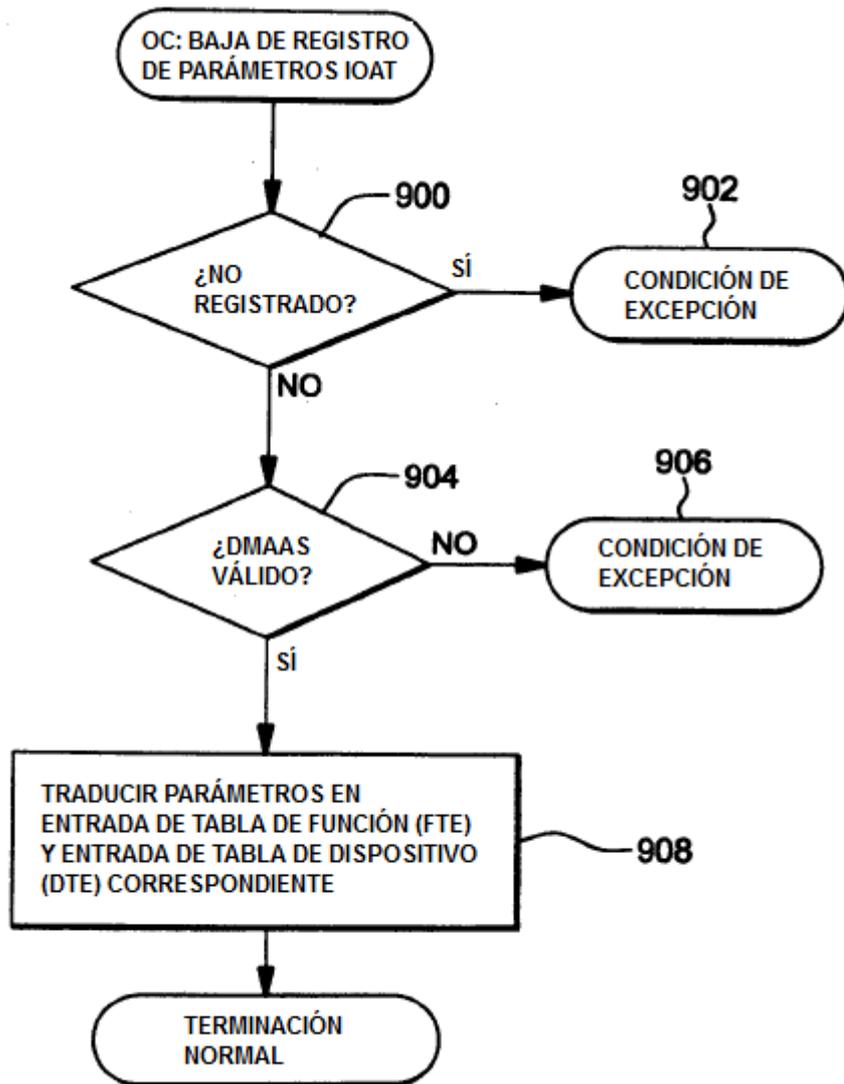


FIG. 9

PRODUCTO
DE PROGRAMA
DE ORDENADOR

1000

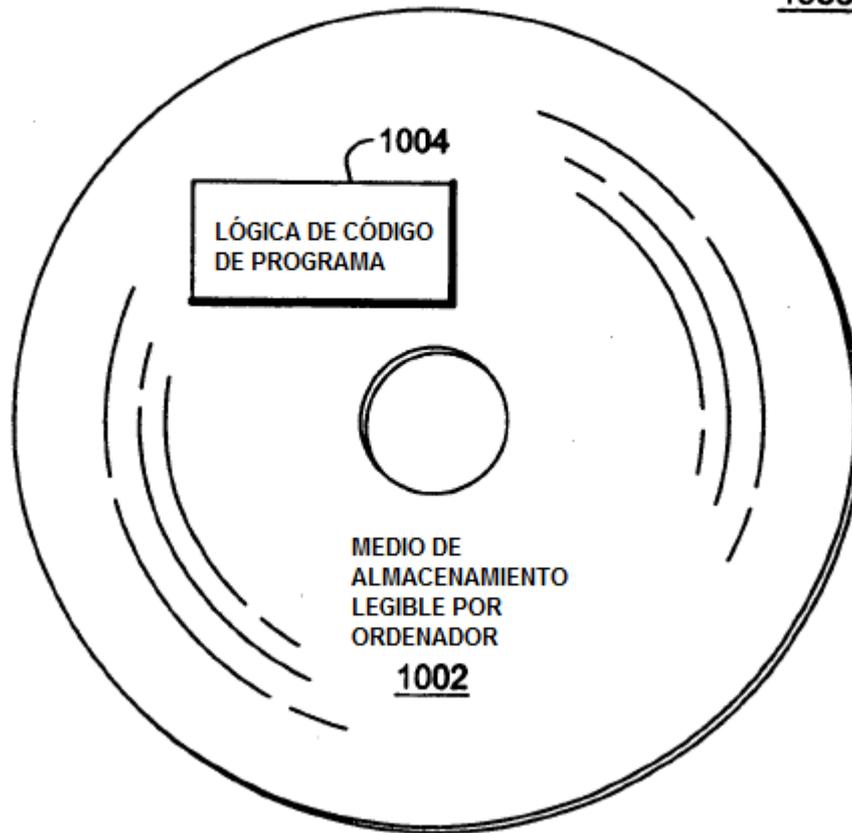


FIG. 10

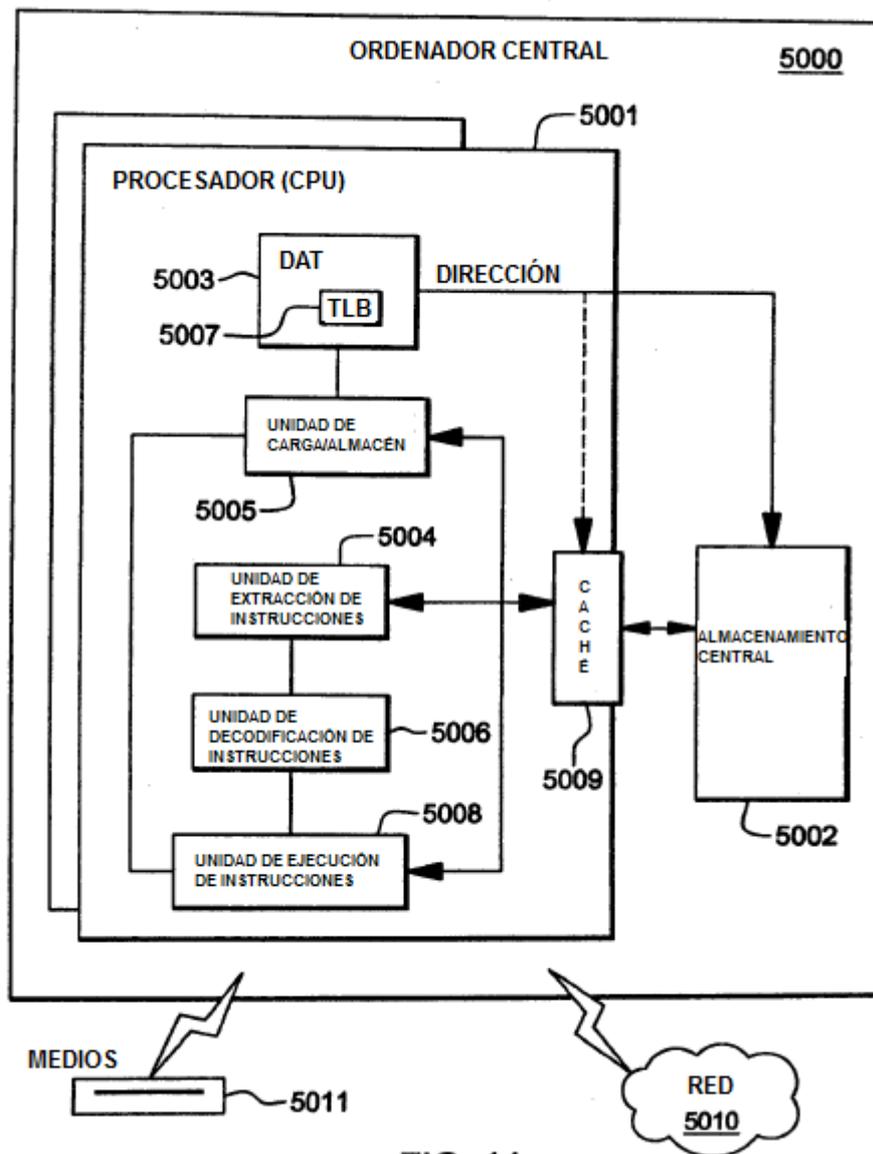


FIG. 11

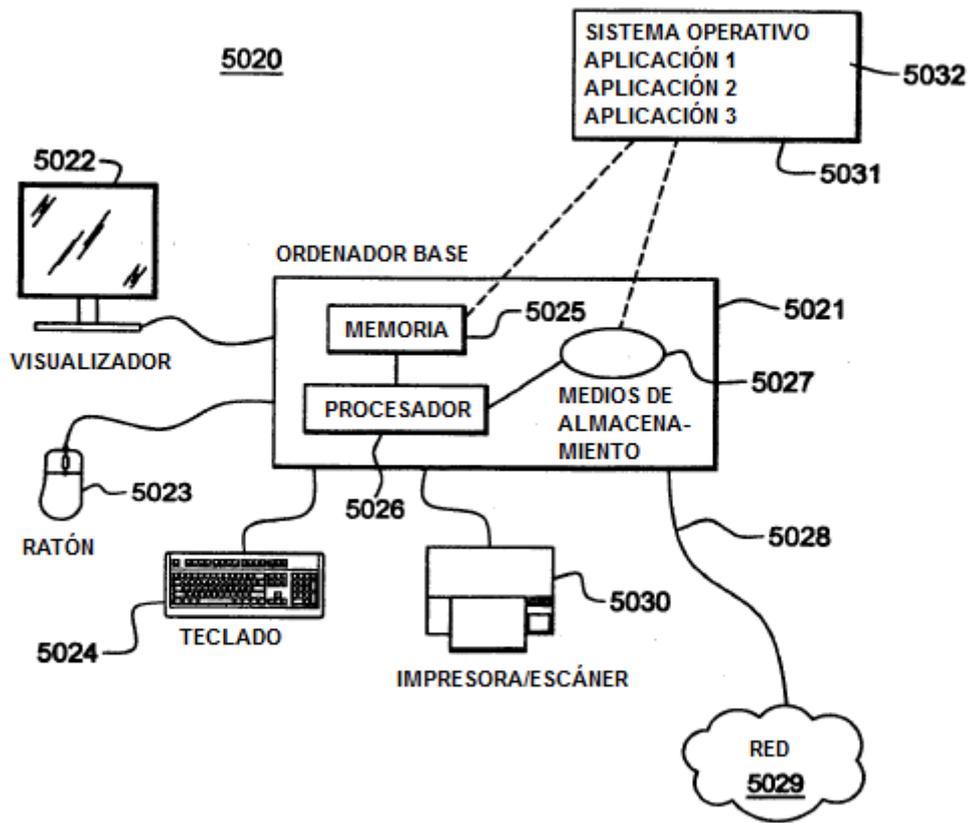


FIG. 12

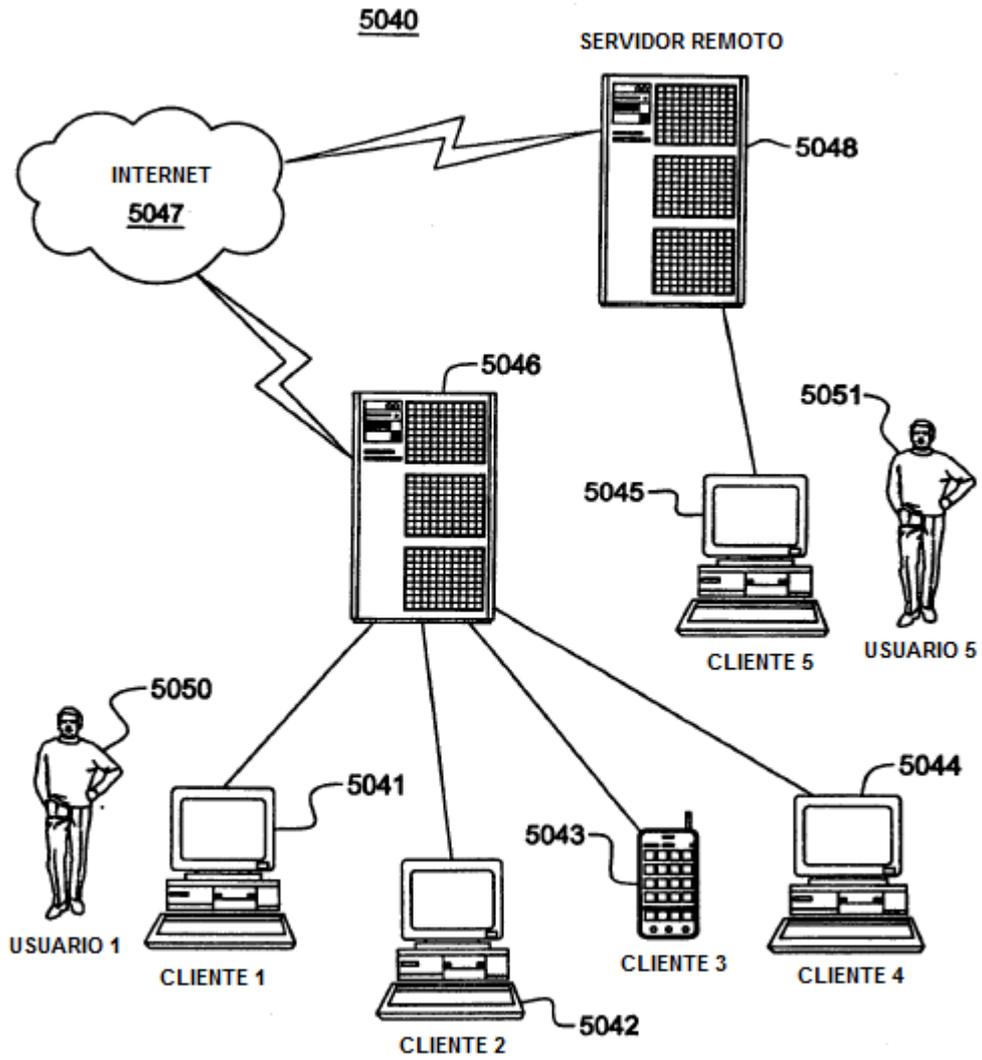


FIG. 13

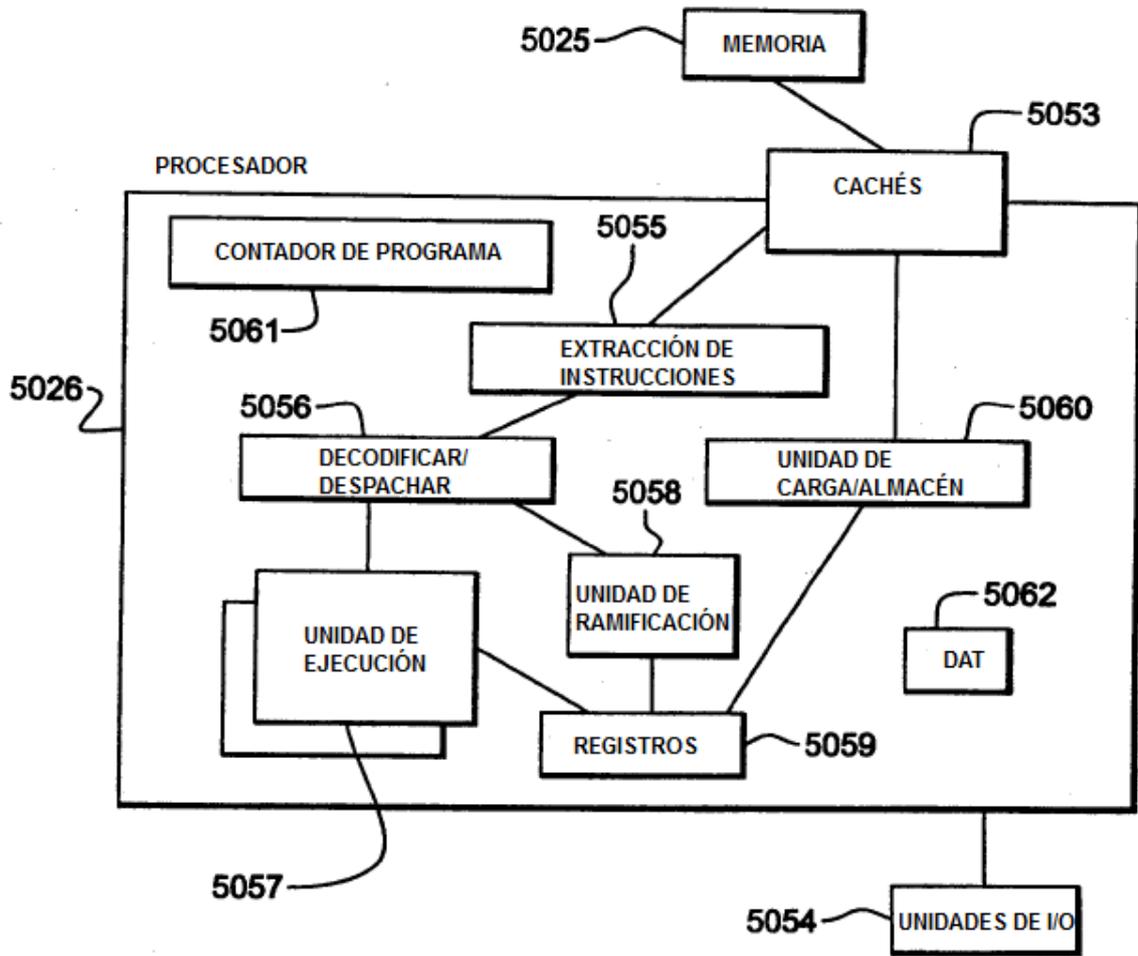


FIG. 14

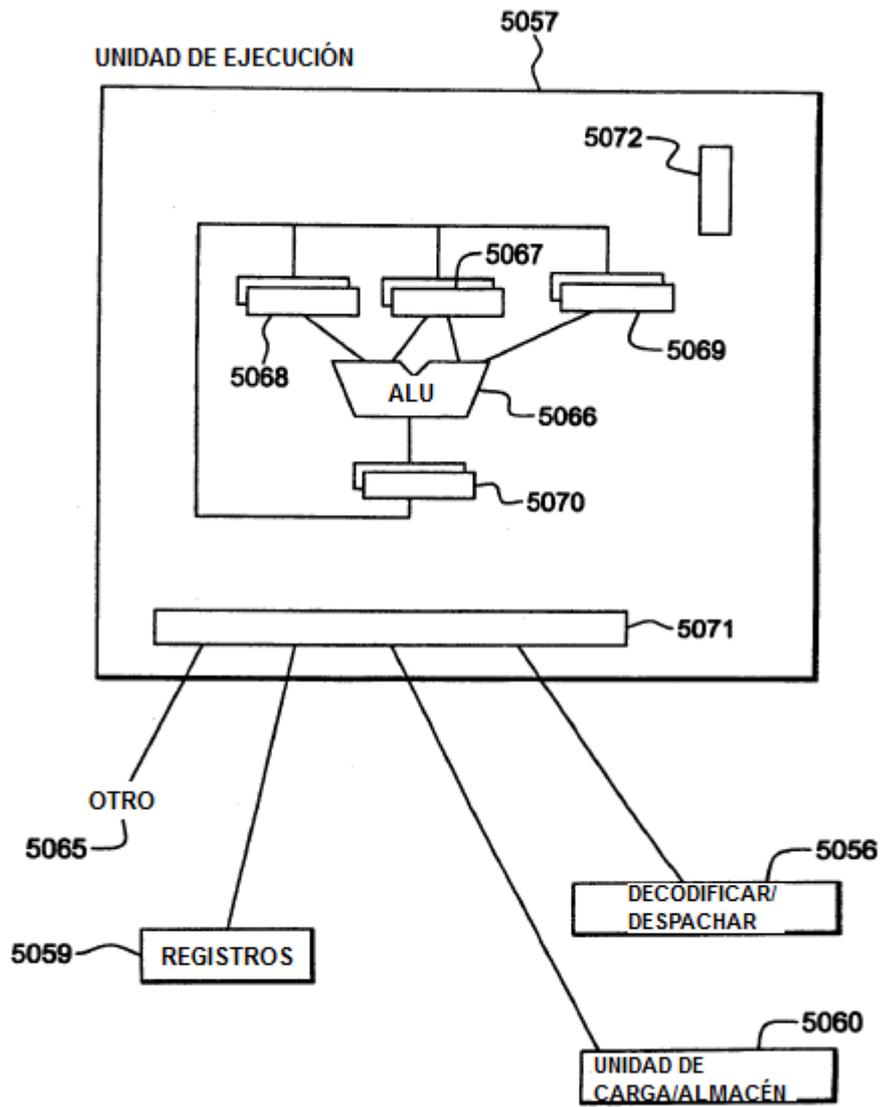


FIG. 15A

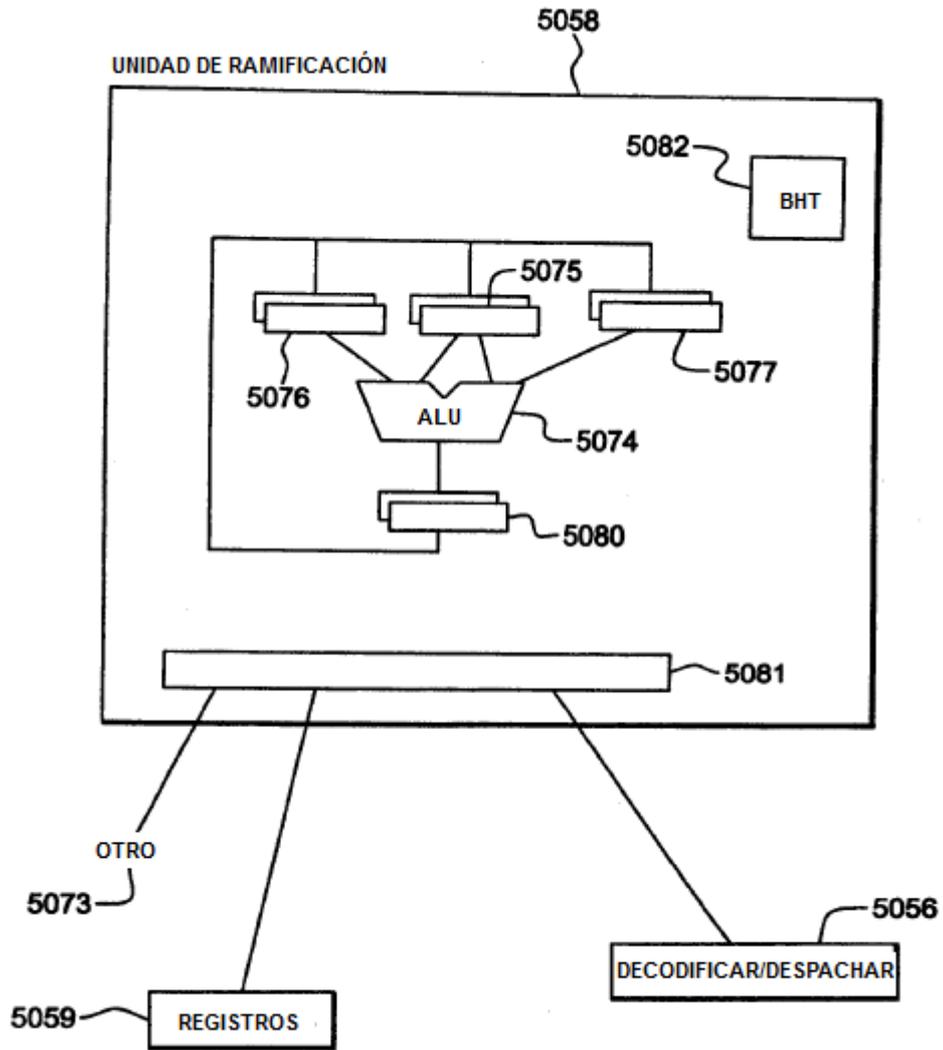


FIG. 15B

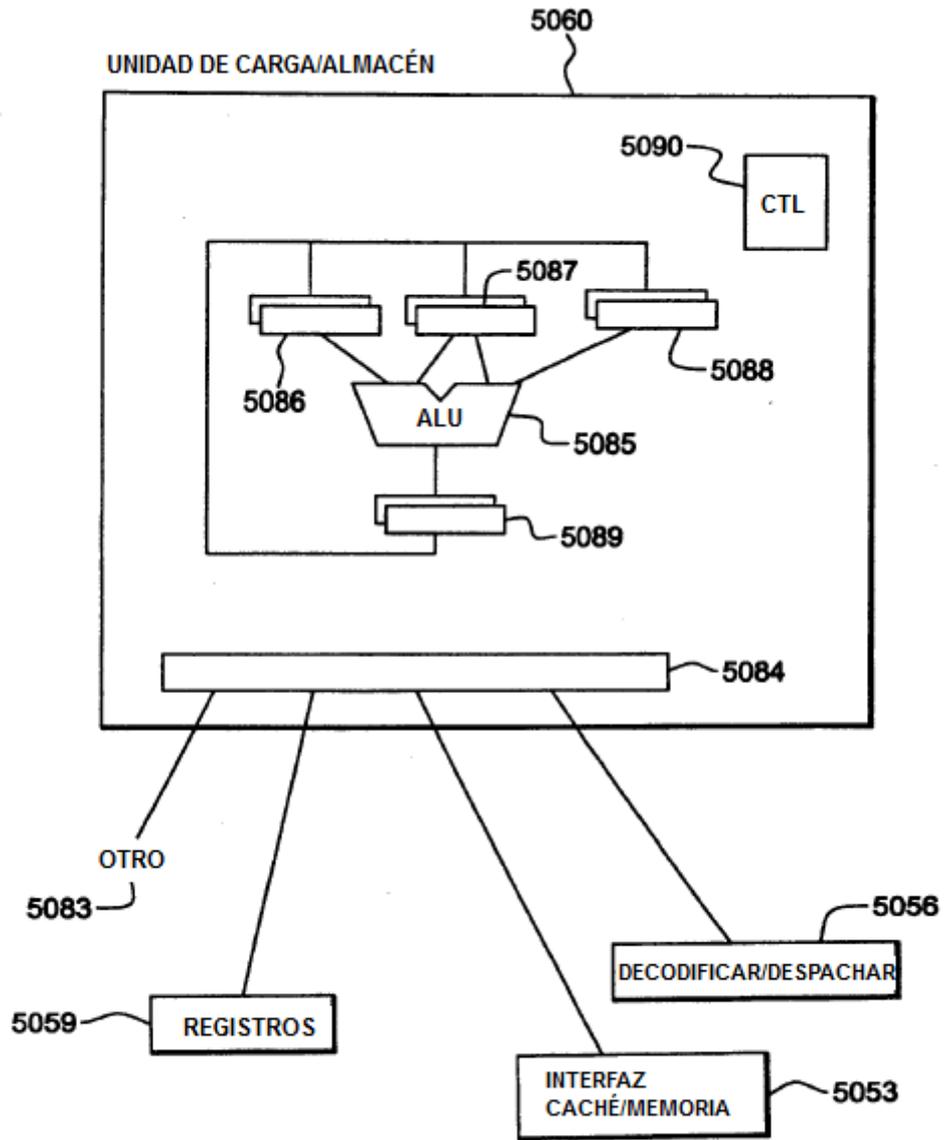


FIG. 15C

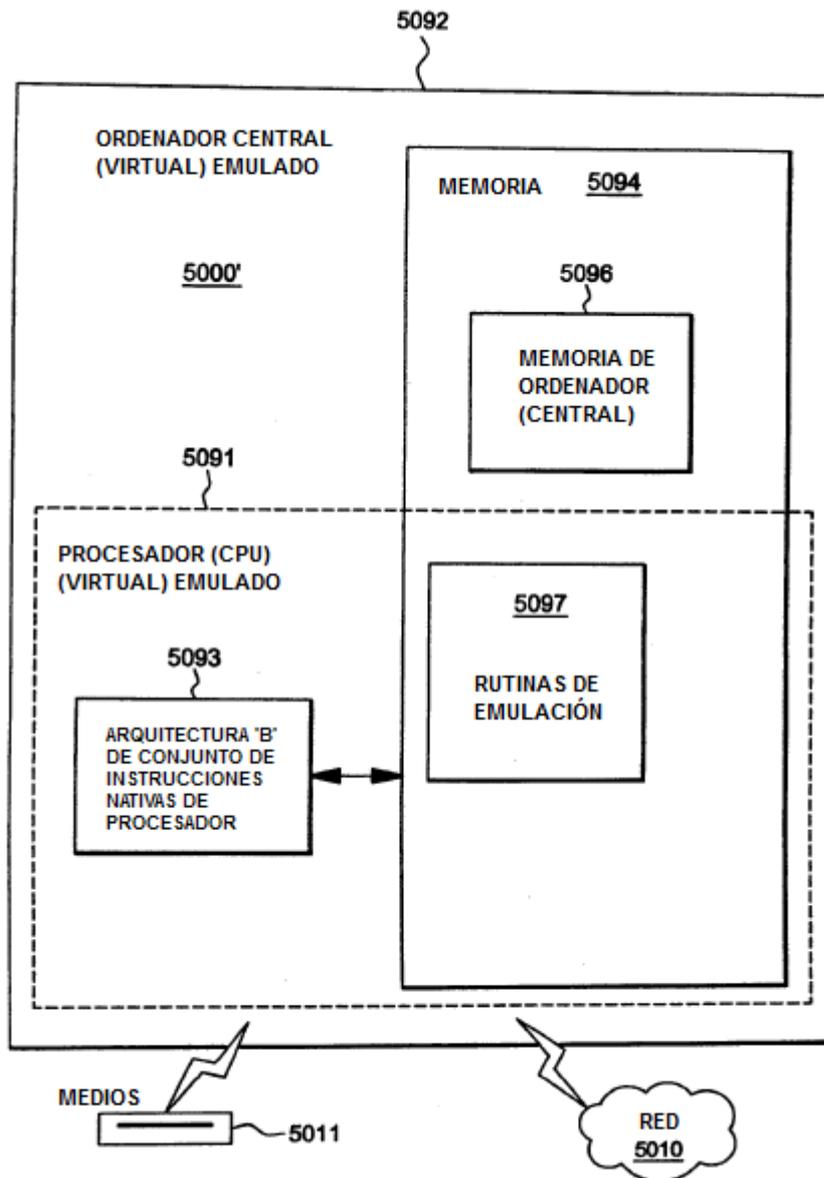


FIG. 16