

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 768 277**

51 Int. Cl.:

G06F 9/46 (2006.01)
G06F 9/48 (2006.01)
G06F 9/455 (2008.01)
G06F 9/50 (2006.01)
G06F 9/54 (2006.01)
G06F 13/24 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **13.11.2012** **PCT/IB2012/056371**
87 Fecha y número de publicación internacional: **25.07.2013** **WO13108091**
96 Fecha de presentación y número de la solicitud europea: **13.11.2012** **E 12865669 (1)**
97 Fecha y número de publicación de la concesión europea: **25.12.2019** **EP 2805235**

54 Título: **Función de interrupción de seguimiento de advertencias**

30 Prioridad:

18.01.2012 US 201213352509

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
22.06.2020

73 Titular/es:

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

72 Inventor/es:

**GAINEY JR, CHARLES;
MULDER, JAMES;
FARRELL, MARK;
BERNARD, PIERCE;
SCHMIDT, DONALD, WILLIAM;
KUBALA, JEFFREY, PAUL y
ROGERS, ROBERT**

74 Agente/Representante:

ISERN JARA, Jorge

ES 2 768 277 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

Función de interrupción de seguimiento de advertencias

5 Antecedentes

Un aspecto de esta invención se refiere, en general, a procesamiento en un entorno informático, y en particular, a facilitar el procesamiento asociado con recursos compartidos.

10 Un tipo de entorno en el que se comparten los recursos es un entorno virtual, que incluye una unidad de procesamiento central (CPU) de anfitrión y una o más unidades de procesamiento central invitadas. Una CPU invitada (también conocida como una CPU virtual) se aprovisiona por un programa anfitrión (por ejemplo, el sistema operativo anfitrión) que se ejecuta en una CPU de anfitrión. El programa de anfitrión realiza acciones para asignar recursos de una configuración de anfitrión subyacente y para asignar estos recursos a la CPU invitada.

15 En una realización particular, una CPU invitada existe cuando una CPU de anfitrión entra en el modo de ejecución interpretativa. En este punto, el sistema operativo de invitado (también denominado en el presente documento como el programa de invitado) comienza la ejecución en la CPU virtualizada, mientras el programa de anfitrión suspende la ejecución en la CPU de anfitrión. El programa de anfitrión reanuda la ejecución en la CPU cuando finaliza el modo de ejecución interpretativa. Existen técnicas de vinculación entre el anfitrión y el invitado mediante las cuales se graba y restaura el estado de anfitrión e invitado. Típicamente, cuando un programa de anfitrión inicia un programa de invitado, el programa de anfitrión se suspende hasta que vuelve el programa de invitado. Tanto la CPU invitada como la CPU de anfitrión son diferentes modos de una CPU de anfitrión.

25 La configuración de anfitrión normalmente incluye todos los recursos del sistema informático. Estos recursos incluyen, pero sin limitación, unidades de procesamiento central (CPU), memoria principal, y dispositivos de entrada/salida (E/S). En un sistema de este tipo, pueden soportarse múltiples CPU invitadas de una única CPU de anfitrión. Esto se consigue asignando a cada CPU invitada un periodo de tiempo para usar la CPU de anfitrión, denominado un segmento de tiempo, y a continuación moviendo la CPU de anfitrión a otra CPU invitada durante un segmento de tiempo y así sucesivamente. El número de CPU invitadas soportable por una CPU de anfitrión varía de acuerdo con las capacidades de la CPU de anfitrión y la capacidad deseada que va a asignarse a cada CPU invitada.

Una configuración de invitado típicamente está formada de dos o más CPU invitadas, y se denomina como una configuración de multiprocesamiento de invitado (MP). Cada CPU invitada puede aprovisionarse mediante una compartición de una CPU de anfitrión separada o incluso compartiendo una única CPU de anfitrión. Un atributo de tal compartición es que una CPU invitada puede operar durante un periodo de tiempo, denominado como un segmento de tiempo, y a continuación está inactiva durante algún periodo de tiempo arbitrario. El periodo inactivo varía basándose en políticas de prioridad establecidas por el sistema, el número global de las CPU invitadas que van a compartir las CPU de anfitrión, y la técnica de compartición particular que se está usando.

40 En un sistema de multiprocesamiento de invitado de este tipo, un programa, en ocasiones denominado una unidad despachable (DU), puede despacharse por el sistema operativo de invitado en una CPU invitada, y a continuación durante la ejecución de esa unidad despachable, el segmento de tiempo de anfitrión de esa unidad despachable se agota. Esto podría dejar la unidad despachable en una condición de este tipo que no pueda continuarse en cualquier otra CPU invitada de la configuración multiprocesamiento de invitado, independientemente de la disponibilidad de cualquier otra CPU invitada. En su lugar, debe esperar a que la una y única CPU invitada reciba su siguiente segmento de tiempo para continuar. Basándose en la técnica de compartición y la prioridad relativa de la configuración de invitado, el siguiente segmento de tiempo puede visualizarse durante un periodo de tiempo sustancial. Incluso si la configuración de invitado tiene otras CPU invitadas que pueden ejecutar la unidad despachable, la continuación de la unidad despachable no es posible debido al estado de la CPU invitada de la unidad despachable que se grabó cuando se agotó el segmento de tiempo anterior. Hasta que pueda usarse el estado preciso para continuar la CPU invitada, la unidad despachable está inactiva.

55 La Patente de Estados Unidos Número US 7536690 B2 presenta un método y sistema que prepara una tarea para que se intercambie de la utilización de procesador que se está ejecutando en un ordenador con múltiples procesadores que cada uno soporta múltiples flujos.

Breve resumen

60 Se superan las desventajas de la técnica anterior y se proporcionan ventajas a través de la provisión de un producto de programa informático para facilitar el procesamiento en un entorno informático.

De acuerdo con un aspecto de la invención, se proporciona un sistema informático para facilitar el procesamiento en un entorno informático de acuerdo con la reivindicación 1.

65 De acuerdo con otro aspecto de la invención, se proporciona un método de facilitación de procesamiento en un entorno

informático de acuerdo con la reivindicación 8.

Los métodos y sistemas relacionados con uno o más aspectos de la presente invención se describen y reivindican también en el presente documento. Además, los servicios relacionados con uno o más aspectos de la presente invención se describen también y pueden reivindicarse en el presente documento.

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

Breve descripción de las diversas vistas de los dibujos

Uno o más aspectos de la presente invención se señalan particularmente y se reivindican de manera distinta como ejemplos en las reivindicaciones en la conclusión de la memoria descriptiva. Los anteriores y objetos, características, y ventajas de la invención son evidentes a partir de la siguiente descripción detallada tomada en conjunto con los dibujos adjuntos en los que:

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

La Figura 2 representa otra realización de un entorno informático para incorporar y usar uno o más aspectos de la presente invención;

La Figura 3 representa otra realización más de un entorno informático para incorporar y usar uno o más aspectos de la presente invención;

La Figura 4 representa una realización de la lógica asociada con la observación de invitado de la función de interrupción de seguimiento de advertencias, de acuerdo con un aspecto de la presente invención;

La Figura 5 representa una realización de la lógica asociada con la observación de invitado de la función de interrupción de seguimiento de advertencias, de acuerdo con un aspecto de la presente invención;

La figura 6 representa una realización de la lógica asociada con un anfitrión que maneja una salida de invitado, de acuerdo con un aspecto de la presente invención;

La Figura 7 representa una realización de una vista general de la lógica de la función de interrupción de seguimiento de advertencias, de acuerdo con un aspecto de la presente invención;

Las Figuras 8A-8C representan realizaciones de la lógica asociada con el procesamiento de función de interrupción de seguimiento de advertencias, de acuerdo con un aspecto de la presente invención;

La Figura 9 representa una realización de la lógica asociada con recibir una interrupción de seguimiento de advertencias, de acuerdo con un aspecto de la presente invención;

La Figura 10 representa una realización de un formato de una instrucción de diagnóstico usada de acuerdo con un aspecto de la presente invención;

La Figura 11 representa una realización de un producto de programa informático que incorpora uno o más aspectos de la presente invención;

La Figura 12 representa una realización de un sistema de ordenador de anfitrión para incorporar y usar uno o más aspectos de la presente invención;

La Figura 13 representa un ejemplo adicional de un sistema informático para incorporar y usar uno o más aspectos de la presente invención;

La Figura 14 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 Figura 15 representa una realización de diversos elementos de un sistema informático para incorporar y usar uno o más aspectos de la presente invención;

La Figura 16A representa una realización de la unidad de ejecución del sistema informático de la Figura 15 para incorporar y usar uno o más aspectos de la presente invención;

La Figura 16B representa una realización de la unidad de ramal del sistema informático de la Figura 15 para

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

La Figura 16C representa una realización de la unidad de carga/almacén del sistema informático de la Figura 15 para incorporar y usar uno o más aspectos de la presente invención; y

La Figura 17 representa una realización de un sistema informático de anfitrión emulado para incorporar y usar uno o más aspectos de la presente invención.

Descripción detallada

De acuerdo con un aspecto de la presente invención, se proporciona una capacidad para advertir a un programa (por ejemplo, un sistema operativo) que tiene un periodo de gracia en el que realizar a función. Por ejemplo, se proporciona a un programa un periodo de gracia para realizar limpieza (por ejemplo, completar, detener y/o mover una unidad despachable).

De acuerdo con un aspecto adicional de la presente invención, se advierte a un programa y/o a un procesador que está a punto de perder el acceso a recursos (por ejemplo, recursos compartidos). Por ejemplo, se proporciona una advertencia a un procesador que comparte recursos con otros procesadores que el procesador está a punto de perder acceso a los recursos. Como un ejemplo adicional, se advierte a un programa, tal como un sistema operativo, que se ejecuta en un procesador compartido (es decir, el programa comparte el procesador con otros programas) que está a punto de perder sus recursos de procesador.

En una realización particular, se proporciona una capacidad en la que un programa de invitado que se ejecuta en una CPU invitada aprovisionada por una CPU de anfitrión una advertencia del agotamiento de un segmento de tiempo dado a la CPU invitada de la CPU de anfitrión o de preferencia por el anfitrión del segmento de tiempo del invitado. La advertencia proporciona un periodo de gracia que la CPU invitada puede usar para realizar una función particular, tal como ejecución completa de una unidad despachable, detener la unidad despachable en un punto en el que la unidad despachable es re-despachable, y/o mover la unidad despachable a otra CPU invitada.

Como se usa en el presente documento, un periodo de gracia incluye como ejemplos una cantidad de tiempo, un número de instrucciones, un número de ciclos, etc., tiene una duración predeterminada en la que puede realizarse una o más funciones.

Una realización de un entorno informático para incorporar y usar uno o más aspectos de la presente invención se describe con referencia a la Figura 1. En esta realización particular, un entorno 100 informático incluye una pluralidad de procesadores 102 que comparten recursos 104. A cada procesador (y/o un programa, tal como un sistema operativo, que se ejecuta en el procesador) se proporciona una cierta cantidad de tiempo, denominada como un segmento de tiempo, para compartir los recursos. Los recursos incluyen recursos de unidad de procesamiento central, memoria, dispositivos de entrada/salida o interfaces, y/u otros recursos, como ejemplos. El procesador (o un programa que se ejecuta en el mismo) que tiene acceso a los recursos se advierte de que su acceso está a punto de finalizar y por lo tanto, el procesador (o programa) es para realizar una acción particular, tal como limpieza, finalizando una unidad de trabajo, deteniendo una unidad de trabajo, moviendo una unidad de trabajo, etc.

Otra realización de un entorno 200 informático para incorporar y usar uno o más aspectos de la presente invención se describe con referencia a la Figura 2. El entorno 200 informático está basado, por ejemplo, en la z/Architecture® ofrecida por International Business Machines Corporation (IBM®), Armonk, Nueva York. La z/Architecture® se describe en una publicación de IBM® titulada, "z/Architecture Principles of Operation", IBM® Publicación N.º SA22-7832-08, novena edición, agosto de 2010. En un ejemplo, un entorno informático basado en la z/Architecture® incluye un sistema z® server, ofrecido por International Business Machines Corporation, Armonk, Nueva York. IBM®, z/Architecture® y zSeries®, así como z/VM® y z/OS® anteriormente mencionados son marcas comerciales registradas de International Business Machines Corporation, Armonk, Nueva York, Estados Unidos. Otros nombres usados en el presente documento pueden ser marcas comerciales registradas, marcas comerciales, o nombres de producto de International Business Machines Corporation u otras compañías.

Como un ejemplo, el entorno 200 informático incluye un complejo de procesador central (CPC) 202 que proporciona soporte de máquina virtual. El CPC 202 incluye, por ejemplo, una o más máquinas 204 virtuales (o en otra realización, particiones lógicas), uno o más procesadores 206, al menos un anfitrión 208 (por ejemplo, un programa de control, tal como un hipervisor), y un subsistema 210 de entrada/salida, cada uno de los cuales se describe a continuación. En este ejemplo, las máquinas virtuales y el anfitrión están incluidos en la memoria.

El soporte de la máquina virtual del CPC proporciona la capacidad para operar grandes números de máquinas virtuales, pudiendo alojar cada una un sistema 212 de operativo de invitado, tal como z/VM®, z/OS® o Linux, como ejemplos. Cada máquina 204 virtual puede funcionar como un sistema separado. Es decir, cada máquina virtual puede resetearse independientemente, alojar un sistema operativo de invitado, y operar con diferentes programas. Un sistema operativo o programa de aplicación que se ejecuta en una máquina virtual parece que tiene acceso a un sistema total y completo, pero, en realidad, únicamente está disponible una porción de él.

Los recursos físicos del CPC (por ejemplo, las CPU, memoria, dispositivos de E/S, etc.) son de propiedad del anfitrión 208, y los recursos físicos compartidos se despachan por el anfitrión a los sistemas operativos de invitado, según sea necesario, para cumplir sus demandas de procesamiento. Las interacciones entre los sistemas operativos de invitado y los recursos de máquina compartida física se controlan por el anfitrión, puesto que el gran número de invitados típicamente impide que el anfitrión simplemente particione y asigne los recursos de hardware a los invitados configurados.

Los procesadores 206 centrales son recursos de procesadores físicos que pueden asignarse a una máquina virtual. Por ejemplo, la máquina 204 virtual incluye uno o más procesadores lógicos, cada uno de los cuales representa todo o una partición de un recurso 206 de procesador físico que puede asignarse dinámicamente a la máquina virtual. Las máquinas 204 virtuales se gestionan por el anfitrión 208. Como ejemplos, el anfitrión puede implementarse en microcódigo que se ejecuta en procesadores 206 o puede ser una parte de un sistema operativo anfitrión que se ejecuta en la máquina. En un ejemplo, el anfitrión 208 es el Gestor de Recursos/Sistema de Procesador (PR/SM) ofrecido por International Business Machines Corporation, Armonk, Nueva York.

El subsistema 210 de entrada/salida dirige el flujo de información entre dispositivos y el almacenamiento principal. Está acoplado al complejo de procesamiento central, en que puede ser parte del complejo de procesamiento central o separado del mismo. El subsistema de E/S alivia a los procesadores centrales de la tarea de comunicar directamente con los dispositivos de E/S acoplados al CPC y permite que el procesamiento de datos proceda concurrentemente con el procesamiento de E/S.

En una realización, el hardware/firmware del anfitrión (por ejemplo, PR/SM) y del procesador (por ejemplo, System z®) interactúan entre sí de una manera cooperativa controlada para procesar las operaciones de sistema operativo de invitado sin requerir la transferencia de control desde/al sistema operativo de invitado y al anfitrión. Las operaciones de invitado pueden ejecutarse directamente sin intervención de anfitrión mediante una instalación que permite que se ejecuten de manera interpretativa las instrucciones para un invitado. Esta instalación proporciona una instrucción, Iniciar Ejecución Interpretativa (SIE), que el anfitrión puede emitir, designando un bloque de control denominado descripción de estado que mantiene el estado y controles del invitado (máquina virtual). La instrucción coloca la CPU en un modo de ejecución interpretativa en el que las instrucciones de invitado y las interrupciones se procesan directamente, hasta que surja una condición que requiera la atención del anfitrión. Cuando tiene lugar una condición de este tipo, se finaliza la ejecución interpretativa, y se presenta una interrupción de anfitrión, o la instrucción del SIE se completa almacenando detalles de la condición encontrada; esta última acción se denomina intercepción. Un ejemplo de ejecución interpretativa se describe en "System/370 Extended Architecture/Interpretive Execution", IBM Publicación N.º SA22-7095-01, septiembre de 1985.

Como se usa en el presente documento, el firmware incluye, por ejemplo, el microcódigo, el milicode y/o macrocódigo del procesador. Incluye, por ejemplo, las instrucciones de nivel de hardware y/o estructuras de datos usadas en la implementación de código máquina de nivel superior. En una realización, incluye, por ejemplo, código propietario que se entrega típicamente como microcódigo que incluye software confiable o microcódigo específico para el hardware subyacente y controla el acceso del sistema operativo al sistema hardware.

Otro ejemplo de un entorno informático para incorporar uno o más aspectos de la presente invención se representa en la Figura 3. En este ejemplo, se proporciona un sistema 300 informático de anfitrión emulado que emula un ordenador de anfitrión 302 de una arquitectura de anfitrión. En el sistema 300 informático de anfitrión emulado, un procesador de anfitrión (CPU) 304 es un procesador de anfitrión emulado (o procesador de anfitrión virtual) y se realiza a través de un procesador 306 de emulación que tiene una arquitectura de conjunto de instrucciones nativa diferente que la usada por los procesadores del ordenador 302 de anfitrión. El sistema 300 informático de anfitrión emulado tiene la memoria 308 accesible al procesador 306 de emulación. En la realización de ejemplo, la memoria 308 se particiona en una porción 310 de memoria de ordenador de anfitrión y una porción 312 de rutinas de emulación. La memoria 310 informática de anfitrión está disponible para programas del ordenador 302 de anfitrión emulado de acuerdo con la arquitectura informática de anfitrión, y puede incluir tanto un anfitrión como hipervisor 314 como una o más máquinas virtuales 316 que ejecutan sistemas 318 operativos invitados, análogo a los elementos de similar nombre en la Figura 2.

El procesador 306 de emulación ejecuta instrucciones nativas de un conjunto de instrucciones de arquitectura de una arquitectura distinta de la del procesador 304 emulado. Las instrucciones nativas se obtienen, por ejemplo, de la memoria 312 de rutinas de emulación. El procesador 306 de emulación puede acceder a una instrucción de anfitrión para la ejecución de un programa en la memoria 310 informática de anfitrión empleando una o más instrucción o instrucciones obtenidas en una rutina de secuencia y acceso/decodificación que puede decodificar la instrucción o instrucciones de anfitrión accedidas para determinar una rutina de ejecución de instrucción nativa para emular la función de la instrucción de anfitrión accedida. Una instrucción de anfitrión de este tipo puede ser, por ejemplo, una instrucción Iniciar Ejecución interpretativa (SIE), mediante la cual el anfitrión busca ejecutar un programa de invitado en una máquina virtual. Las rutinas 312 de emulación pueden incluir soporte para esta instrucción y para ejecutar una secuencia de instrucciones de invitado en una máquina 316 virtual de acuerdo con la definición de esta instrucción de SIE.

Otras funciones que se definen para la arquitectura del sistema 302 informático de anfitrión pueden emularse mediante rutinas de funciones de arquitectura, que incluyen tales funciones como registros de fin general, registros de control, traducción de dirección dinámica y caché de soporte y procesador de subsistema de E/S, como ejemplos. Las rutinas de emulación pueden aprovecharse también de funciones disponibles en el procesador 306 de emulación (tal como registros generales y traducción dinámica de direcciones virtuales) para mejorar el rendimiento de las rutinas de emulación. Pueden proporcionarse también motores de hardware y descarga especiales para ayudar al procesador 306 a emular las funciones del ordenador 302 anfitrión.

De acuerdo con un aspecto de la presente invención, se proporciona una función de interrupción de seguimiento de advertencias que puede usarse en muchos tipos de entornos informáticos. Aunque pueden usarse en muchos tipos de entornos, los aspectos de la función se describen en el presente documento con referencia a un sistema de multiprocesamiento invitado. Como se ha descrito anteriormente, en sistemas de procesamiento invitados, los sistemas operativos de invitado despachan unidades despachables (por ejemplo, programas, código, etc.) en unidades de procesamiento central invitadas que se alojan por al menos una unidad de procesamiento central de anfitrión. La CPU de anfitrión proporciona un segmento de tiempo (por ejemplo, una cantidad de tiempo u otro periodo, tal como el número de instrucciones, número de ciclos, etc.) a la CPU invitada, tiempo durante el cual la unidad despachable ejecuta. Si durante la ejecución de la unidad despachable, el segmento de tiempo se agota, la unidad despachable puede dejarse en una condición de este tipo que no pueda continuarse en cualquier otra CPU invitada de la configuración multiprocesamiento de invitado, independientemente de la disponibilidad de cualesquiera otras CPU invitadas. En su lugar, debe esperar a que la una y única CPU invitada reciba su siguiente segmento de tiempo para continuar. Basándose en la técnica de compartición particular empleada y la prioridad relativa de la configuración de invitado, el siguiente segmento de tiempo puede visualizarse durante un periodo de tiempo sustancial. Incluso si la configuración de invitado tiene otras CPU invitadas que pueden ejecutar la unidad despachable, la continuación de la unidad despachable no es posible debido al estado de la CPU invitada de la unidad despachable que se grabó cuando se agotó el segmento de tiempo anterior. Hasta que pueda usarse el estado preciso para continuar la CPU invitada, la unidad despachable está inactiva.

Es posible ampliar el segmento de tiempo concediendo tiempo adicional (u otro periodo adicional, tal como instrucciones adicionales, ciclos, etc.), pero incluso con este tiempo adicional, la CPU invitada podría ampliar la ejecución de la unidad despachable, y aún dejarla en la misma condición no despachable como estaría en el agotamiento del segmento de tiempo normal.

Puesto que un programa anfitrión no conoce los controles y estado usados por un programa invitado arbitrario que ejecuta una unidad despachable arbitraria, siempre se concede tiempo adicional para el sistema operativo de invitado para limpiar su unidad despachable hasta que no pueda hacerse sin establecer un protocolo entre el programa de anfitrión y el programa de invitado. Sin un protocolo, cualquier tiempo adicional concedido a la CPU invitada se consumiría en el procesamiento principal y aún posiblemente concluiría con la misma condición de unidad despachable bloqueada. Por lo tanto, de acuerdo con un aspecto de la presente invención, se proporciona un protocolo de este tipo.

De acuerdo con un aspecto de la presente invención, se proporciona un periodo de gracia o extensión que incluye una advertencia que indica al programa de invitado que debería tomarse una acción particular (por ejemplo, completar una unidad despachable o hacer la unidad despachable re-despachable). Como ejemplos, se proporciona el periodo de gracia en respuesta al agotamiento del segmento de tiempo, o en respuesta a que el anfitrión de preferencia al invitado, antes del agotamiento de su segmento de tiempo, para reclamar al procesador alguna otra prioridad superior, como se observa por el anfitrión, como se describe en detalle adicional en el presente documento.

Como un ejemplo, se proporciona el periodo de gracia en lugar de ampliar de manera incondicional el segmento de tiempo. Si el segmento de tiempo normal se ha agotado completamente, se proporciona un periodo de gracia, pero se carga contra el siguiente segmento de tiempo normal para permanecer equitativo a todos los otros invitados virtualizados que cada uno tiene una expectativa de segmento de tiempo. Si el segmento de tiempo normal no se ha agotado, el periodo de gracia se toma del tiempo normal restante. De cualquier manera, el periodo de gracia limita el tiempo restante (u otro periodo) dado a la CPU invitada y no es por sí mismo ampliable. Por lo tanto, la CPU invitada no puede continuar ejecutándose durante un periodo arbitrario y desconocido.

En el comienzo del periodo de gracia, se notifica al programa de invitado del fin de la limpieza de la unidad despachable (por ejemplo, completándola, deteniéndola y/o moviéndola). La aplicación del periodo de gracia asegura que la CPU invitada no supera el periodo adicional concedido. El protocolo mediante el cual se le proporciona al programa de invitado un periodo de gracia y se notifica que ese tiempo (u otro periodo) está casi agotado (el periodo de gracia ha comenzado) es un acuerdo entre el programa de invitado y el programa de anfitrión que el protocolo es entendido por el programa de invitado, haciendo de esta manera que valga la pena una notificación de este tipo. Es decir, el programa de invitado respetaría normalmente la notificación, cuando sea necesario, haciendo a la unidad despachable actual despachable en otra CPU invitada de la configuración de invitado (por ejemplo, moviéndola).

Se describen a continuación detalles adicionales con respecto al protocolo y a la función de interrupción de seguimiento

de advertencias (también denominado como seguimiento de advertencias o función de seguimiento de advertencias) con referencia a las Figuras 4-10. La realización descrita con referencia a estas figuras se refiere a un entorno virtual que tiene uno o más invitados aprovisionados por uno o más anfitriones. Sin embargo, uno o más aspectos de la presente invención se refieren a otros entornos también, que incluyen entornos no virtuales en los que múltiples procesadores y/o múltiples programas comparten recursos.

Con referencia a la Figura 4, se describen detalles con respecto al protocolo de función de interrupción de seguimiento de advertencias como se observa por un invitado. El protocolo de seguimiento de advertencias, como se observa por el invitado, incluye, por ejemplo, una indicación de función instalada, registro de invitado, notificación y salida voluntaria, cada una de las cuales se describe a continuación.

Haciendo referencia a la Figura 4, el programa de invitado entiende el protocolo de seguimiento de advertencias y busca una indicación de función instalada, etapa 400. En un ejemplo, esta indicación es un bit almacenado en un bloque de control (por ejemplo, un bloque de control de llamada de servicio (SCCB)), que se observa usando un comando de lectura, tal como un comando de información de SCP leída. Posterior a la determinación de que la función está instalada, el programa de invitado realiza el registro, etapa 402. El registro es un mecanismo en el que el programa de invitado comunica al programa de anfitrión que el programa de invitado entiende el protocolo de función de interrupción de seguimiento de advertencias. En un ejemplo, el registro se realiza usando una instrucción de diagnóstico, un ejemplo de la cual se describe adicionalmente a continuación.

El registro iniciado desde cualquier unidad de procesamiento central invitada cubre, en una realización, todas las unidades de procesamiento central invitadas de la configuración de multiprocesamiento, puesto que se desea el comportamiento consistente a través de CPU invitadas de la configuración multiprocesamiento de invitado. En una configuración de multiprocesamiento de invitado, las CPU invitadas usan la misma memoria principal y se supone que las CPU invitadas operan en lo que en ocasiones se denomina una única imagen. Por lo tanto, el registro de una CPU invitada es aplicable a las otras CPU invitadas del entorno de multiprocesamiento. El registro es irrevocable, en una realización, y esto ayuda a evitar ventanas de temporización, simplifica el desarrollo y permite capacidad de prueba mejorada. Incluso aunque el registro sea irrevocable, el programa de invitado puede determinar si continuará participando en el protocolo. Si elige esto, no necesita participar reseteando o teniendo que resetear uno o más indicadores de activación descritos a continuación.

Posterior a registrar el programa de invitado para el protocolo de seguimiento de advertencias, una CPU invitada puede notificarse de un periodo de gracia, etapa 404. Por ejemplo, puede advertirse por una CPU de anfitrión del agotamiento de un segmento de tiempo (o un agotamiento inminente, en un ejemplo adicional) o de dar preferencia a su segmento de tiempo. En un ejemplo particular, se notifica a una CPU de la configuración de invitado registrada, por ejemplo, del agotamiento de su segmento de tiempo normal y el comienzo de un periodo de gracia que proporciona un periodo adicional para, por ejemplo, limpieza.

Posterior a que se notifique, el invitado tiene una cantidad restringida de tiempo u otro periodo, el periodo de gracia (por ejemplo, 50 microsegundos en una realización particular), para hacer una unidad despachable re-despachable o para hacer cualesquiera otros ajustes apropiados. Si ya se ha finalizado el segmento de tiempo normal, entonces, por ejemplo, antes de devolver el control al anfitrión, se usa el periodo de gracia para hacer la unidad despachable re-despachable o para hacer cualesquiera otros ajustes apropiados. Si no ha finalizado el segmento de tiempo, entonces se usa el periodo de gracia y se renuncia cualquier porción restante del segmento de tiempo. Tiene lugar la contabilidad normal del tiempo real usado por una CPU invitada.

Posterior a hacer la notificación, la CPU invitada está en un periodo restringido (por ejemplo, cantidad limitada de tiempo) después de lo cual se finaliza involuntariamente la operación de la CPU invitada. Únicamente se realiza una notificación para cada periodo de segmento de tiempo normal. Por lo tanto, la CPU invitada aún está restringida por un control de tiempo final que asegura que la CPU compartida subyacente de anfitrión puede compartirse en cualquier lugar este punto, conservando de esta manera buen orden y disciplina en la virtualización global proporcionada por el programa de anfitrión.

La notificación puede conseguirse por cualquier mecanismo que provoque que se detecte un estado único por el programa de invitado. Ejemplos incluyen una interrupción de invitado única, una localización de memoria principal definida a nivel de arquitectura que puede establecerse, o un dispositivo de memoria externa de E/S, disponible para tanto anfitrión como cliente. El primero requiere una activación de invitado para permitir la interrupción. Los últimos dos requieren exámenes periódicos suficientemente a menudo para que el periodo de gracia no se desperdicie. En un ejemplo particular de la z/Architecture®, una interrupción de invitado, denominada como una interrupción de seguimiento de advertencias (WTI), se usa como una notificación.

Posterior a que se advierta, el invitado sale voluntariamente de su segmento de tiempo/periodo de gracia dado, etapa 406. El invitado envía el segmento de tiempo/periodo de gracia actual después de notificarse y hacer la unidad despachable re-despachable (por ejemplo, detener y mover la unidad despachable, o completarla). Esta salida señala al programa de anfitrión que el invitado de hecho está siguiendo el protocolo. Pueden tener lugar otras razones para que un invitado renuncie al control, volviendo de esta manera al programa de anfitrión. Normalmente, para que

el procesamiento restringido haga una unidad despachable re-despachable, no tendrían que tener lugar condiciones para cualquier salida extraña. Si la CPU invitada sale dentro del periodo de gracia mediante el protocolo de interrupción de seguimiento de advertencias, se proporciona una indicación de realimentación en el siguiente segmento de tiempo, cada vez que tenga lugar. Por lo tanto, el programa de invitado sabe que si cumple restricción de tiempo impuesta por el periodo de gracia.

Si el invitado llega tarde en la salida voluntaria, su ejecución se da preferencia por el agotamiento del periodo de gracia. La siguiente vez que se inicia la CPU invitada con un segmento de tiempo normal, se proporciona una indicación de realimentación de modo que el invitado sepa que ha llegado tarde. En general, esto puede usarse para determinación de problemas en el programa de invitado puesto que el periodo de gracia habitual permite suficiente tiempo para limpieza y salida voluntaria.

Si tiene lugar una salida extraña, la siguiente vez que se inicie la CPU invitada dentro de un segmento de tiempo normal, se espera que la salida voluntaria tenga lugar rápidamente. El mismo mecanismo de realimentación informaría al programa de invitado que ha intervenido una salida extraña y por lo tanto proporciona diferente información para informar la determinación de problema.

La salida voluntaria se consigue mediante cualquier mecanismo que provoque que el control pase a través del programa de anfitrión desde el programa de invitado, y que incluye el mecanismo de realimentación anteriormente mencionado. El mecanismo usado es para definirse en el protocolo de interrupción de seguimiento de advertencias de la arquitectura particular para que el programa de anfitrión reconozca la solicitud de invitado. En un ejemplo, este mecanismo incluye la instrucción de diagnóstico descrita a continuación.

Además de la observación de invitado del protocolo de interrupción de seguimiento de advertencias, en una realización, el anfitrión también observa la instalación, como se describe en detalle adicionalmente a continuación con referencia a la Figura 5.

Haciendo referencia a la Figura 5, el anfitrión reconoce la función instalada y la refleja a sus invitados, etapa 500. Por ejemplo, el anfitrión comprueba el bit instalado en el bloque de control (por ejemplo, SCCB) y reconoce el estado instalado del protocolo de seguimiento de advertencias (es decir, está activado) y sabe cómo el programa de anfitrión puede usarlo para aprovecharse del anfitrión. Por lo tanto, la indicación de instalación se refleja a su invitado. Por ejemplo, para reflejar la instalación a un invitado, el anfitrión activa un bit instalado en un bloque de control de invitado (por ejemplo, SCCB invitado) o un área de memoria accesible para el invitado. Si, por cualquier razón, el programa de anfitrión no desea que un invitado observe el estado instalado del protocolo de función de interrupción de seguimiento de advertencias y no permite que un invitado observe su estado instalado y lo use, el programa de anfitrión pasa una indicación de no instalado al invitado (por ejemplo, establece el bit observado por el invitado a cero). Adicionalmente, en una realización, el programa de anfitrión establece los controles de las CPU invitadas de manera que se desactiva el protocolo de seguimiento de advertencias (por ejemplo, desconecta uno o más bits designados en la descripción de estado de la CPU invitada).

A medida que se inicia el registro por un invitado, el programa de anfitrión recibe la solicitud de registro no solicitada y recuerda que el invitado se ha registrado, etapa 502. Una solicitud de registro iniciada por cualquier CPU invitada única es suficiente para registrar todas las CPU invitadas en una configuración de multiprocesamiento de invitado. Por lo tanto, el programa de anfitrión posibilita el protocolo de seguimiento de advertencias para todas las CPU de la configuración de invitado, etapa 504. Por ejemplo, el programa de anfitrión establece uno más bits designados en la descripción de estado de la CPU invitada para posibilitar la función de interrupción de seguimiento de advertencias para los invitados. La realimentación del registro no se devuelve necesariamente al invitado. Cuando una CPU invitada intente registro incluso aunque la instalación no esté instalada, el anfitrión ignoraría la solicitud y no podría activar las CPU invitadas para el protocolo de función de interrupción de seguimiento de advertencias.

Posterior a registrar y posibilitar los invitados para la función de seguimiento de advertencias, un invitado puede recibir la notificación de la invocación del protocolo, etapa 506. Esto puede conseguirse en un número de escenarios, como se describe a continuación.

Como un ejemplo, cuando se activa el protocolo de interrupción de seguimiento de advertencias para una CPU invitada que opera en el modo de ejecución interpretativa en una CPU de anfitrión, es decir la CPU de anfitrión X, el programa de anfitrión puede iniciar el protocolo de la CPU de anfitrión Y. Es decir, una CPU invitada se ha aprovisionado con una CPU de anfitrión X y esa CPU de anfitrión X está actualmente no disponible para el programa de anfitrión. Si el programa de anfitrión tiene razón para re-obtener la CPU X, en primer lugar provoca que la CPU X salga de modo de ejecución interpretativa. Es decir, se detiene la CPU invitada, saliendo de esta manera del modo de ejecución interpretativa de la CPU X. Detener la CPU invitada en cualquier punto arbitrario sin permitir que la CPU invitada se detenga por sí misma voluntariamente crea un riesgo de un problema del cual se pretende que lo resuelva el protocolo de interrupción de seguimiento de advertencias. El protocolo de interrupción de seguimiento de advertencias permite que la CPU de anfitrión Y solicite la notificación permitiendo que una acción de programa de anfitrión se transforme en una notificación en la CPU invitada X, etapa 506. Puesto que el programa de invitado se ha registrado previamente, el programa de anfitrión tiene una expectativa de que el programa de invitado reconozca la notificación y soporte un

manejo apropiado de la notificación que incluye la última etapa de finalizar voluntariamente la ejecución, devolviendo por lo tanto el control de la CPU de anfitrión X al programa de anfitrión. Una vez que esto tiene lugar, el programa de anfitrión puede continuar con cualquiera que sea el uso de la CPU de anfitrión X que pudiera haber provocado que se inicie el proceso.

El programa de anfitrión notifica a un invitado, por ejemplo, estableciendo cualquier estado, estableciendo un indicador (por ejemplo, un bit), o provocando que se envíe una señal asíncrona no solicitada (por ejemplo, una interrupción de seguimiento de advertencias) al invitado. Aunque está registrado, el tiempo de recepción de una notificación de este tipo es aún desconocido para el invitado. Al registrar, el invitado acaba de acordar adherirse al protocolo si y cuando se señalice.

En un sistema uni-procesador de anfitrión, si una CPU de anfitrión está en el modo de ejecución interpretativa de manera que la CPU invitada está operando, no existe otra CPU de anfitrión para invocar el protocolo de interrupción de seguimiento de advertencias. Sin embargo, incluso en este caso, el protocolo de interrupción de seguimiento de advertencias puede invocarse por la misma CPU cuando la CPU de anfitrión, mientras está en el modo de ejecución interpretativa, reconoce el agotamiento del segmento de tiempo, y puede a continuación conceder un periodo de gracia y realizar la notificación.

En un ejemplo adicional de notificación, la notificación tiene lugar cuando debido a un cambio de estado interno reconocido por la CPU de anfitrión cuando está en el modo de ejecución interpretativa, la CPU de anfitrión provoca la notificación definida en el protocolo de interrupción de seguimiento de advertencias para que se envíe a la CPU invitada. Un ejemplo de esto tiene lugar cuando la CPU invitada está activada para el protocolo de interrupción de seguimiento de advertencias y la CPU de anfitrión reconoce el fin del segmento de tiempo. Antes de señalar la CPU invitada para renunciar al control, la CPU concede internamente un periodo de gracia para permitir a la CPU invitada suficiente tiempo para recibir la señal, para tomar la acción apropiada (por ejemplo, completar la unidad despachable actual o hacer la unidad despachable actual re-despachable), y para finalizar voluntariamente. De manera interna, la CPU de anfitrión mantiene el estado para indicar que se ha notificado a la CPU invitada. Si el invitado no finaliza voluntariamente dentro del periodo de gracia, la CPU reconoce esto y finaliza la ejecución del invitado, devolviendo de esta manera el control al programa de anfitrión finalizando el modo de ejecución interpretativa. El invitado no tiene manera para determinar, en una realización, por qué se invocó el protocolo, sino únicamente que se notificó para limpiar y finalizar. Pueden existir otras razones del anfitrión para provocar un fin del modo de ejecución interpretativa, y por lo tanto, finalizar la ejecución de invitado. Por ejemplo, existen escenarios donde tiene que detenerse la ejecución de todas las CPU invitadas para hacer algún cambio coordinado a la configuración de invitado completa. Una configuración de invitado de multiprocesamiento no tiene que tener algunas CPU operando bajo diferentes reglas o suposiciones de otras CPU de la configuración de invitado. Tal asimetría podría crear resultados de invitado impredecibles.

La CPU de anfitrión recibe el efecto de la CPU invitada que ha realizado la salida voluntaria o salida por cualquier otra razón (por ejemplo, el anfitrión ha devuelto el recurso de la CPU), etapa 508. Si la salida es debido a lo que define el protocolo de interrupción de seguimiento de advertencias, entonces el programa de anfitrión recuerda proporcionar una realimentación a la CPU invitada la siguiente vez que inicie, por mucho tiempo que pueda ser. Esta realimentación es una indicación positiva ("buena"), suponiendo que el invitado salió voluntariamente antes del agotamiento del periodo de gracia. Si la salida es debido a cualquier otra razón, no tiene lugar realimentación de protocolo de interrupción de seguimiento de advertencias en el siguiente inicio de la CPU invitada.

Si el invitado llega tarde en la salida voluntaria; es decir, el invitado realiza una acción para salir voluntariamente, pero se ha agotado el periodo de gracia, la ejecución de la CPU invitada se le da preferencia por el agotamiento del periodo de gracia. La siguiente vez que se inicia la CPU invitada con un segmento de tiempo normal, se proporciona una indicación de realimentación de excepción de modo que el invitado sepa que ha llegado tarde en su salida voluntaria. En general, esto puede usarse para determinación de problemas en el programa de invitado puesto que el periodo de gracia habitual permite suficiente tiempo para limpieza y salida voluntaria.

Si tiene lugar una salida distinta de la salida voluntaria del protocolo de interrupción de seguimiento de advertencias, la siguiente vez que se inicia la CPU invitada con un segmento de tiempo normal, no se incluye realimentación por el protocolo de seguimiento de advertencias.

Los medios para salir voluntariamente se consiguen mediante cualquier mecanismo que provoque que el control pase al programa de anfitrión del programa de invitado, que se reconoce por el programa de anfitrión como la salida voluntaria del protocolo, y que incluye el mecanismo de realimentación anteriormente mencionado. En un ejemplo, se usa una instrucción de diagnóstico para la salida voluntaria. Es decir, la instrucción de diagnóstico con un parámetro particular se usa para indicar la finalización del segmento de tiempo. Después de que se emita la instrucción de diagnóstico por el programa de invitado y se ejecute, el programa de anfitrión determina si la salida fue a tiempo. A continuación, cuando el invitado se inicia de nuevo, que es en la siguiente instrucción secuencial después del diagnóstico, se proporciona un código de condición que indica si se proporcionó a tiempo. El código de condición se establece, por ejemplo, en la PSW invitada que se usa para iniciar el invitado en la siguiente instrucción secuencial. El invitado puede a continuación probar el código de condición.

El manejo de la salida voluntaria de invitado por el anfitrión se describe adicionalmente con referencia a la Figura 6. Inicialmente, el control vuelve a la CPU de anfitrión, cuando la CPU invitada se detiene, etapa 600. Se realiza una determinación en cuanto a si se devuelve el control dentro del periodo de gracia, INQUIRY 602. Si el control se devuelve dentro del periodo de gracia, entonces el programa de anfitrión observa una salida de invitado voluntaria por el protocolo de interrupción de seguimiento de advertencias y recuerda buena realimentación para el siguiente inicio de la CPU invitada, independientemente de cuál CPU de anfitrión pueda aprovisionar la CPU invitada en ese momento, etapa 604. Esto es suponiendo que la función de interrupción de seguimiento de advertencias está instalada. Si no, no se recuerda el estado de realimentación. Sin embargo, si el invitado realiza una acción para salir voluntariamente pero está fuera del periodo de gracia, INQUIRY 602, entonces el programa de anfitrión en la CPU de anfitrión observa una salida de invitado voluntaria por el protocolo de interrupción de seguimiento de advertencias (incluso aunque fuera de manera inoportuna y hubiera de salirse involuntariamente) y recuerda mala realimentación para el siguiente inicio de la CPU invitada, independientemente de qué CPU de anfitrión pueda aprovisionar la CPU invitada en ese momento, etapa 606. De nuevo, esto es suponiendo que la función de seguimiento de advertencias está instalada. Si no, no se recuerda el estado de realimentación.

Posteriormente, si el anfitrión está recordando buena o mala realimentación, el programa de anfitrión redirige la CPU de anfitrión a una asignación de preferencia, etapa 608. Es decir, el anfitrión se redirige para realizar una o más funciones ahora que tiene sus recursos (CPU) de vuelta.

Además, en el siguiente inicio secuencial de la CPU invitada, independientemente de qué CPU de anfitrión aprovisiona a la CPU invitada, si se recuerda el estado de realimentación, la indicación de estado de realimentación se establece antes de iniciar la CPU invitada, etapa 610. En un ejemplo, se establece en la descripción de estado de SIE, por ejemplo, en la PSW de la descripción de estado, que indica el inicio de la siguiente instrucción secuencial.

Se describen detalles adicionales con respecto al procesamiento asociado con la función de interrupción de seguimiento de advertencias con referencia a las Figuras 7-9. En particular, la Figura 7 representa una realización de la lógica asociada con una vista general del procesamiento de función de interrupción de seguimiento de advertencias; Las Figuras 8A-8C proporcionan detalles del procesamiento de función de interrupción de seguimiento de advertencias, de acuerdo con un aspecto de la presente invención; y la Figura 9 representa una realización de la lógica asociada con recibir una interrupción de seguimiento de advertencias.

Haciendo referencia a la Figura 7, inicialmente un programa de invitado (por ejemplo, el sistema operativo de invitado) reconoce que la función de interrupción de seguimiento de advertencias está instalada, etapa 700. En una realización, esto se consigue observando el programa de invitado un indicador de función instalada (por ejemplo, bit) que está localizado, por ejemplo, en un bloque de control designado. Si el sistema operativo de invitado tiene el soporte para participar en la función de interrupción de seguimiento de advertencias, reconoce el indicador de función de interrupción de seguimiento de advertencias instalado y a continuación indica su capacidad para participar en el protocolo. En un ejemplo, esto incluye registrar su intento para participar en el procesamiento de seguimiento de advertencias, etapa 702. Como se describe en el presente documento, en un ejemplo, se realiza el registro mediante una instrucción de diagnóstico. Cuando se registra, el sistema operativo de invitado indica tanto a la CPU de anfitrión como al programa de anfitrión que sabe cómo manejar una interrupción de seguimiento de advertencias (WTI), que es una interrupción no ambigua que proporciona una advertencia al invitado que está, por ejemplo, a punto de perder acceso a su recurso compartido (por ejemplo, CPU invitada) y que está, por ejemplo, para tomar la acción con respecto a su unidad despachable que se está ejecutando actualmente. El registro es un prerrequisito, en una realización, para la recepción de la WTI. Si el invitado no está registrado para la función de interrupción de seguimiento de advertencias, tras el agotamiento del segmento de tiempo del invitado, no se ofrece periodo de gracia, y la CPU invitada se lleva fuera del modo de ejecución interpretativa.

Incluso si está registrado, en una realización, el programa de invitado tiene dos mecanismos para desactivar la presentación de la WTI. Por ejemplo, puede establecerse un bit seleccionado a cero en, por ejemplo, la palabra de estado de programa (PSW) que desactiva la presentación de todas las interrupciones externas incluyendo WTI; o puede establecerse un bit en un registro de control especificado (por ejemplo, CR0) a cero para desactivar justo la WTI. Cuando ambos bits son unos, se activa la presentación de la WTI. Si la presentación de una WTI permanece desactivada para el periodo de gracia de WTI, la ejecución del invitado finaliza sin el beneficio de la WTI, que constituye una salida involuntaria.

Durante la ejecución interpretativa de la CPU invitada, si la CPU invitada reconoce internamente cualquiera de una condición de interrupción externa de temporizador de CPU de anfitrión (por ejemplo, un segmento de tiempo agotado) o una preferencia solicitada por el programa de anfitrión, INQUIRY 704, el procesamiento de CPU interno determina, antes de recibir el anfitrión el control, si ha de realizarse el procesamiento de interrupción de seguimiento de advertencias, INQUIRY 706. Es decir, el procesamiento de CPU interno comprueba que se posibilita que el invitado advierta el procesamiento de seguimiento y por lo tanto determine que el procesamiento de seguimiento de advertencias debería incluirse en el procesamiento a realizarse. Si no se realiza el procesamiento de interrupción de seguimiento de advertencias, a continuación finaliza la ejecución interpretativa del invitado, etapa 708, y el control vuelve al programa de anfitrión, etapa 710. Volviendo a INQUIRY 706, si ha de realizarse, sin embargo, el

procesamiento de interrupción de seguimiento de advertencias, a continuación se realiza ese procesamiento, como se describe en detalle adicional a continuación, etapa 712.

Se describen realizaciones de detalles adicionales del procesamiento de interrupción de seguimiento de advertencias con referencia a las Figuras 8A-8C. En este procesamiento, se emplea un número de indicadores de control que incluye lo siguiente:

Un control interno de periodo activo de gracia de función de interrupción de seguimiento de advertencias (por ejemplo, bit G), que no es a nivel de arquitectura visible, pero que se usa por la lógica de CPU interna;

Una interrupción de seguimiento de advertencias (WTI) presenta el control interno (por ejemplo, bit P), que indica, cuando es uno, que la WTI se ha presentado al invitado y cuando es cero indica que no se ha presentado. Como el control interno activo de periodo de gracia de la función de interrupción de seguimiento de advertencias, el control interno de la WTI presentada no es visible a nivel de arquitectura, pero se usa por la lógica de CPU interna;

Una preferencia de programa de anfitrión del control de invitado (por ejemplo, bit T), que es, por ejemplo, el indicador de solicitud de intervención de seguimiento de advertencias en la descripción de estado de CPU invitada; y

Existe la activación de las interrupciones externas, cuando un indicador E es uno. En un ejemplo, el indicador E es un bit dentro de la palabra de estado de programa actual (PSW).

Haciendo referencia a la Figura 8A, en un ejemplo, se reconoce una condición de interrupción de temporizador de CPU de anfitrión (por ejemplo, segmento de tiempo esperado) o se reconoce una solicitud de intervención de seguimiento de advertencias (por ejemplo, el anfitrión desea devolver de manera temprana los recursos de CPU; es decir, antes de la finalización del segmento de tiempo). Si se reconoce una condición de interrupción de temporizador de CPU de anfitrión, INQUIRY 800, se realiza una determinación en cuanto a si se activa el indicador de control activo de periodo de gracia (por ejemplo, es G igual a 1), INQUIRY 802. Si no se activa G, a continuación el indicador G se activa a, por ejemplo, 1, etapa 804, y el periodo de gracia de la función de interrupción de seguimiento de advertencias está a punto de comenzar. El valor actual del temporizador de CPU de anfitrión se graba a continuación (el valor grabado se denomina el valor original en el presente documento), etapa 806, y el temporizador de CPU de anfitrión se establece al periodo de gracia de seguimiento de advertencias (por ejemplo, 50 microsegundos), etapa 808.

Posteriormente, se realiza una determinación en cuanto a si se activa el invitado para una interrupción de seguimiento de advertencias, INQUIRY 810. En una realización, si el nivel 2 de invitado está activo que indica que un invitado ha iniciado otro invitado, a continuación el invitado 2 sale del modo de ejecución interpretativa como para una interrupción de invitado 1, y se desprecia la instrucción de ejecución interpretativa de inicio del invitado 1. Por lo tanto, el procesamiento es como un invitado 1 en este punto. Si el invitado 2 no está activo, a continuación el procesamiento simplemente continúa con el invitado 1. Si el invitado se activa para una WTI, a continuación se presenta la interrupción externa de seguimiento de advertencia (WTI) al invitado, etapa 812. En un ejemplo, esta interrupción incluye un código de interrupción particular que se presenta que indica que tiene un periodo de gracia para realizar una o más funciones (por ejemplo, limpieza), si se desea.

Además, se establece P a 1 que indica que se ha presentado la WTI, etapa 814. También, el bit T se establece a 1 usando una función de actualización interbloqueada (puede ya tener 1 si se hubiera usado originalmente una solicitud de intervención), etapa 816. El periodo de gracia en el temporizador de CPU de anfitrión continúa decrementándose, independientemente de si se presentó la WTI, etapa 818. A continuación se sale de este procesamiento, etapa 820. En un ejemplo, una designación de la salida de este proceso indica que la CPU ha completado el procesamiento actual de la función de interrupción de seguimiento de advertencias y está volviendo a otro procesamiento como se indica por el estado actual de la CPU.

Volviendo a INQUIRY 810, si el invitado no se posibilita para la interrupción de seguimiento de advertencias, el procesamiento continúa a la etapa 816. En este caso, el invitado no se posibilita para WTI, por lo que no puede presentarse al invitado. Sin embargo, el bit T se establece pendiente de modo que puede detectarse más tarde cuando el invitado no posibilita la WTI.

Volviendo a INQUIRY 800, si no es una condición de interrupción de temporizador de CPU de anfitrión, entonces se reconoce una solicitud de intervención de seguimiento de advertencias (es decir, preferencia de anfitrión). Es decir, el bit T en el campo de solicitud de intervención de la descripción de estado del invitado es 1. Por lo tanto, se realiza una determinación en cuanto a si el indicador G está activado, INQUIRY 850. Si no está activado (por ejemplo, 0), a continuación el procesamiento continúa con la ETAPA 804. En esta situación, la condición T igual a 1 es la razón inicial para iniciar el proceso de WTI. Sin embargo, si el bit G está activado, a continuación se realiza una determinación en cuanto a si se activa P, INQUIRY 852. Si P no está activado (por ejemplo, igual a 0), a continuación el procesamiento continúa con la ETAPA 810 en un intento para presentar la WTI. Sin embargo, si P está activado (por ejemplo, no es igual a 0), a continuación el descubrimiento de T equivale a 1 después de que no haya tenido efecto el periodo de gracia de función de seguimiento de advertencias que haya comenzado, y se sale del proceso, etapa 854.

Volviendo a INQUIRY 802, si está activada G (por ejemplo, igual a 1), la CPU invitada ya ha estado ejecutando en el periodo de gracia y el agotamiento del temporizador de CPU de anfitrión indica que se ha agotado el periodo de gracia. Por lo tanto, se había iniciado previamente un ciclo de WTI y se ha agotado el periodo de gracia. Por lo tanto, haciendo referencia a la Figura 8B, el valor de temporizador de CPU de anfitrión original previamente grabado se reduce por la cantidad de tiempo realmente usado durante el periodo de gracia y a continuación se carga en el temporizador de CPU de anfitrión, etapa 860. Se sale del modo de ejecución interpretativa, etapa 862, y la interrupción externa del temporizador de CPU de anfitrión se presenta al anfitrión, etapa 864 (esto es una forma de una salida de invitado involuntaria).

Además de lo anterior, el análisis de WTI puede iniciarse mediante ciertas instrucciones que pueden posibilitar la CPU para WTI. Por ejemplo, haciendo referencia a la Figura 8C, inicialmente, un número de instrucciones que puede activar la CPU para WTI, que incluye, por ejemplo, una instrucción de PSW de Carga (Ampliada) y una instrucción Almacenar Entonces o Máscara de Sistema que pueden establecerse al bit designado en la PSW, y el Control de Carga que puede establecer el bit seleccionado en el registro de control, realizar monitorización, como se describe en el presente documento. Por ejemplo, una instrucción que puede activar interrupciones comprueba el bit T para procesamiento de seguimiento de advertencias potencial. Si T=0, INQUIRY 880, entonces no hay WTI, y se sale de este proceso, etapa 884. Sin embargo, si T=1, entonces el procesamiento continúa con INQUIRY 822.

En INQUIRY 882, se realiza una determinación en cuanto a si se activa P (por ejemplo, igual a 1). En caso afirmativo, a continuación se sale de este proceso, etapa 884, puesto que se había detectado previamente la activación. Sin embargo, si P no está activada (por ejemplo, no igual a 1), a continuación se realiza una determinación adicional en cuanto a si está activada G (por ejemplo, igual a 1), INQUIRY 886. Si no, entonces el procesamiento continúa con la ETAPA 804 (Figura 8A). Sin embargo, si se activa G (por ejemplo, igual a 1), etapa 886 (Figura 8C), a continuación el procesamiento continúa con INQUIRY 810 en la Figura 8A, etapa 888, y se sale del procesamiento.

Se describen detalles adicionales del procesamiento de la interrupción de seguimiento de advertencias con referencia a la Figura 9. Cuando el programa de invitado recibe la interrupción de seguimiento de advertencias, realiza sean cual sean las funciones (por ejemplo, funciones de SO) que han de realizarse para, por ejemplo, hacer la unidad despachable de trabajo re-despachable, etapa 900. Por ejemplo, el invitado detiene la unidad despachable en un punto particular, graba su estado, y mueve a otra CPU invitada o posibilita que se mueva proporcionado información de estado, etc. El sistema operativo de invitado señala que ha finalizado emitiendo una señal completa de limpieza de seguimiento de advertencias al programa de anfitrión (también conocido como, una salida voluntaria), etapa 902. Esta señal puede ser cualquier mecanismo que provoca que se renuncie la operación invitada el segmento de tiempo restante. Sin embargo, ha de reconocerse por el programa de anfitrión como la parte de limpieza del protocolo. En un ejemplo, se usa una función de limpieza completada de la instrucción de diagnóstico.

Si el programa de invitado emite la señal de limpieza completada antes de que se agote el periodo de gracia, INQUIRY 904, el programa de anfitrión recuerda que la CPU invitada sale en el tiempo, etapa 906. Esto es una salida voluntaria. Cuando se inicia a continuación la CPU, la naturaleza a tiempo de la señal se indica de vuelta a la CPU invitada, etapa 908. En un ejemplo, se establece una PSW de reanudación de invitado para indicar un código de condición satisfactorio (por ejemplo, código de condición 0).

Volviendo a INQUIRY 904, si el programa de invitado, por cualquier razón, dura demasiado, el periodo de gracia se agota por medio del temporizador de CPU de anfitrión que ha decrementado el periodo de gracia a cero, presentando por lo tanto una condición de interrupción externa de temporizador de CPU de anfitrión a la CPU. En este caso, la CPU reconoce que el invitado ya estaba en el periodo de gracia y no concede otro periodo de gracia. En su lugar, la ejecución de invitado se detiene y el control se revierte al programa de anfitrión por la recepción de la interrupción externa. Un programa de anfitrión reconoce que esta finalización de la CPU invitada es una salida de invitado involuntaria.

En el siguiente inicio de la CPU invitada, el sistema operativo de invitado puede a continuación emitir una señal de limpieza completa, aunque ahora demasiado tarde. El programa de anfitrión ya no tiene una expectativa más de espera para la recepción de la señal de limpieza completada. Por lo que, la siguiente vez que se inicia la CPU invitada, la naturaleza demasiado tardía de la señal se indica de vuelta a la CPU invitada, etapa 912. En un ejemplo, la PSW de reanudación de invitado se marca para indicar una condición tardía, que se observará por el invitado en el siguiente inicio. La emisión de una instrucción de diagnóstico demasiado tardía se denomina en ocasiones una instrucción de diagnóstico obsoleta puesto que se ha perdido previamente saliendo dentro del periodo de gracia y a continuación se salió sin ningún motivo.

En un ejemplo, tras un nuevo inicio de la CPU invitada, el programa de invitado puede comprobar la parte de reanudación de señal del protocolo para si la señal se hubo emitido o no dentro del periodo de gracia. El programa de invitado puede usar esta información para investigar por qué puede haberse retardado y hacer mejoras para mejorar las estadísticas para ser más oportuno en el futuro.

En una realización, cuando se desactiva el invitado para todas las interrupciones externas, un número de instrucciones

que pueden activar interrupciones externas realizan monitorización. Cuando el invitado está activado para interrupciones externas, se examina una activación de WTI. En este punto, si se activa la WTI y el bit P es 0, la WTI se presenta a la CPU invitada.

Como se ha mencionado anteriormente, en una realización, se usa una función de diagnóstico para indicar que la limpieza está completada o para registrar la función de interrupción de seguimiento de advertencias. Para la limpieza completada, la función de diagnóstico, cuando se emite con un parámetro de limpieza y se ejecuta, señala que la CPU expedidora ha realizado cualquier procesamiento deseado asociado con la recepción de una interrupción externa de seguimiento de advertencias. Cuando la ejecución se completa, se establece un código de condición que indica si la finalización se emitió o no dentro del intervalo de tiempo dependiente del modelo permitido para la limpieza después de la interrupción de seguimiento de advertencias.

Como para la función de registro, la función de diagnóstico, cuando se emite con el parámetro de registro y se ejecuta, señala que la configuración de emisión entiende la interrupción de seguimiento de advertencias. Cuando se finaliza la ejecución, se establece un código de condición satisfactorio. El estado de registro se limpia por un reseteo de sistema.

Una realización de un formato de una instrucción de diagnóstico se describe con referencia a la Figura 10. En una realización, una instrucción de diagnóstico 1000 incluye un opcode 1002 que indica la función de diagnóstico; un primer campo 1004 de registro (R_1); un segundo campo 1006 de registro (R_3); un campo 1008 de registro general (B_2); y un campo 1010 de desplazamiento (D_2). En un ejemplo, los contenidos del campo D_2 se añaden a los contenidos del registro general B_2 . El resultado no se usa para direccionar datos, sino que en su lugar, se usan ciertos bits (por ejemplo, los bits 48-63) como una extensión de código de operación. Cuando la extensión de código de operación es un valor predeterminado, se especifica la limpieza de seguimiento de advertencias y se renuncia a un segmento de tiempo.

En un ejemplo, el campo R_3 no se usa y contiene ceros. Además, no se usan los bits especificados del registro general R_1 y han de contener ceros y un bit particular del registro general R_1 (por ejemplo, el bit 63) especifica la función de limpieza completada cuando es cero y la función de registro cuando es 1.

En una partición lógica que usa CPU físicas compartidas, esta función puede mejorar el rendimiento de sistema permitiendo que la CPU física en la que se está ejecutando la CPU lógica se asigne a otra CPU lógica.

Más allá de diagnóstico, cualquier otra salida de SIE mientras está en el intervalo de gracia de WTI, sea cual sea la causa, restaura de manera similar el valor original del temporizador de CPU de anfitrión decrementado por la cantidad de tiempo de periodo de gracia gastada.

Se describe en detalle en el presente documento una función de interrupción de seguimiento de advertencias que proporciona, en una realización, un mecanismo mediante el cual puede presentarse una interrupción externa de seguimiento de advertencias a una CPU en una configuración con recursos de CPU compartidos, tales como una partición lógica. El programa de control puede usar la interrupción externa de seguimiento de advertencias como la señal para hacer la unidad despachable que se ejecuta actualmente despachable en una CPU diferente en la configuración. En una realización, un procesador lógico (invitado) que se ejecuta en un segmento de tiempo en un procesador físico recibe una señal de advertencia que indica un periodo de gracia, por ejemplo, una cantidad de tiempo antes de que se interrumpirá el procesador lógico (desasignado del procesador físico que puede compartirse) que posibilita que se haga el trabajo por el procesador lógico para que se complete o mueva a otro procesador lógico. Como un ejemplo, se señala a la CPU invitada que su segmento de tiempo se ha agotado y que debería dar preferencia a la unidad despachable actual de trabajo (DU) o hacerla re-despachable en otra CPU invitada. En un ejemplo, la señal de advertencia es una interrupción que tiene un código de interrupción que indica que es una WTI. En un ejemplo adicional, el código de interrupción incluye información acerca de la cantidad del tiempo u otro periodo dado al periodo de gracia.

En una realización, la función de interrupción de seguimiento de advertencias puede usarse en entornos no virtuales, así como virtuales, en los que un programa y/o procesador comparte recursos (por ejemplo, recursos de CPU u otros recursos) con uno o más otros programas y/o procesadores.

En una realización, en la que el entorno es un entorno virtual, desde una perspectiva del invitado:

1. El programa invitado observa la condición instalada de la función de protocolo de interrupción de seguimiento de advertencias.
2. El programa invitado se registra para el protocolo de interrupción de seguimiento de advertencias.
3. La CPU invitada recibe una notificación de seguimiento de advertencias, de acuerdo con la arquitectura particular (por ejemplo, indicación de memoria compartida, indicación de dispositivo de E/S compartido, interrupción).

4. El programa invitado que se ejecuta en la CPU invitada realiza procesamiento aplicable de acuerdo con la naturaleza del programa de invitado que ha recibido la notificación (procesamiento de notificación esperada para que sea única por sistema operativo).

5. La CPU invitada renuncia al control de acuerdo con la técnica voluntaria del protocolo de seguimiento de advertencias.

6. En el siguiente inicio de la CPU invitada, el programa invitado puede observar la realimentación, de acuerdo con protocolo de seguimiento de advertencias.

Además, en una realización, desde la perspectiva del anfitrión:

A. El programa anfitrión observa la condición instalada de la función de protocolo de interrupción de seguimiento de advertencias.

1. El programa anfitrión obtiene la indicación de función instalada de protocolo de interrupción de seguimiento de advertencias.

2. El programa de anfitrión recuerda de manera persistente el estado instalado del protocolo de interrupción de seguimiento de advertencias.

3. El programa de anfitrión indica el estado de protocolo de seguimiento de advertencias instalado a cada configuración de invitado.

4. El programa de anfitrión desactiva el protocolo de seguimiento de advertencias en todas las CPU invitadas no registradas.

5. El programa de anfitrión se prepara para reconocer una solicitud de registro de seguimiento de advertencias de invitado de cada configuración de invitado.

B. El programa de anfitrión reconoce una solicitud de registro de seguimiento de advertencias de un invitado.

1. El programa de anfitrión recuerda de manera persistente que la configuración de invitado entiende el protocolo de seguimiento de advertencias.

2. El programa anfitrión posibilita al invitado el protocolo de seguimiento de advertencias.

C. Durante la operación normal de la CPU invitada X, se usa la preferencia de la CPU invitada X para volver a obtener la correspondiente CPU de anfitrión X.

1. El programa de anfitrión en la CPU Y señala la notificación a la CPU invitada X.

a. La CPU X propaga la notificación a la CPU invitada X mediante la actualización de la localización de memoria compartida, actualización del dispositivo de E/S compartido, o interrupción a la CPU invitada X, de acuerdo con el protocolo de seguimiento de advertencias.

D. La CPU X invitada se detiene, devolviendo el control a la CPU de anfitrión X.

1. Si está dentro del periodo de gracia, el programa de anfitrión en la CPU X observa la salida de invitado voluntaria por protocolo de seguimiento de advertencias y recuerda la buena realimentación para el siguiente inicio de la CPU invitada X, independientemente de qué CPU de anfitrión pueda aprovisionar la CPU invitada X en ese momento.

a. Si está dentro del periodo de gracia, pero la salida de la CPU invitada X no es de acuerdo con el protocolo de seguimiento de advertencias, no se recuerda el estado de realimentación.

2. Si está dentro del periodo de gracia, el programa de anfitrión en la CPU X observa la salida de invitado voluntaria por protocolo de seguimiento de advertencias y recuerda la mala realimentación para el siguiente inicio de la CPU invitada X, independientemente de qué CPU de anfitrión pueda aprovisionar la CPU invitada X en ese momento.

a. Si no está dentro del periodo de gracia, pero la salida de la CPU invitada X no es de acuerdo con el protocolo de seguimiento de advertencias, no se recuerda el estado de realimentación.

3. El programa de anfitrión en la CPU de anfitrión X redirige la CPU X para dar prioridad a la asignación.

E. El siguiente inicio secuencial de la CPU invitada X, independientemente de qué CPU de anfitrión aprovisiona la CPU invitada X, si se recuerda el estado de realimentación, establecer la indicación de realimentación, de acuerdo con el protocolo de seguimiento de advertencias antes de iniciar la CPU invitada X.

5 En una realización, un procesador invitado de la configuración de invitado recibe una interrupción única, definiéndose esta interrupción para una arquitectura informática, y siendo el significado de la interrupción una interrupción de seguimiento de advertencias. La interrupción indica un código específico que identifica la interrupción como una interrupción de seguimiento de advertencias. La interrupción implica un intervalo de tiempo relativamente corto, denominado como un periodo de gracia, que conduce a la finalización de una ejecución de procesador invitado.

10 Durante el periodo de gracia, en un ejemplo, el programa de invitado normalmente se espera que haga la unidad despachable actual de trabajo re-despachable en otro procesador invitado, evitando de esta manera que se bloquee en el procesador invitado actual, esperando su siguiente inicio de segmento de tiempo normal del anfitrión.

15 En un ejemplo, el intervalo de tiempo relativamente corto se concede únicamente una vez por inicio de programa de anfitrión del procesador invitado. El intervalo de tiempo se concede desde, por ejemplo, el intervalo de tiempo existente bajo el que se ejecuta el procesador invitado. Puesto que el intervalo de tiempo concedido se asigna desde el segmento de tiempo restante normal, no es tiempo de préstamo, per se, sino que se usa una cantidad restringida de tiempo del intervalo de tiempo actual para asegurar que el procesador invitado se le da de hecho preferencia en un periodo de tiempo relativamente corto.

20 En un ejemplo adicional en el que se ha agotado el segmento de tiempo actual, se concede el intervalo de tiempo como tiempo adicional además del intervalo de tiempo existente bajo el cual se está ejecutando el procesador invitado. El intervalo de tiempo concedido se contabiliza por el programa de anfitrión como contra el siguiente intervalo de tiempo normal secuencial esperado, que se consumirá por el procesador invitado bajo el cual se espera que el procesador invitado esté ejecutándose a continuación. Se pretende aún asegurar que el procesador invitado se le dé de hecho preferencia en un periodo de tiempo relativamente corto.

25 En un ejemplo, puede generarse una solicitud de interrupción para un evento de seguimiento de advertencia para informar al programa que está cerca del fin del intervalo de ejecución actual en una CPU compartida. La solicitud de interrupción es un tipo de condición pendiente que se genera cuando se registra la configuración y se activa para la función de interrupción de seguimiento de advertencias.

30 El procesamiento cooperativo entre los programas (por ejemplo, anfitrión e invitado) optimiza la compartición de recursos (por ejemplo, CPU) entre programas (por ejemplo, sistemas operativos de invitado). Uno o más aspectos proporcionan, por ejemplo, mejor tiempo de respuesta con la misma utilización de CPU. Además, la serialización del sistema se libera antes del no despacho por el hipervisor.

35 En una realización adicional, puede usarse uno o más aspectos de la invención con solicitudes de un sistema operativo para dejar que continúe un hilo de ejecución individual para mejorar el tiempo transcurrido de trabajo de tiempo sensible. Es decir, un hilo puede solicitar o puede proporcionarse tiempo adicional para realizar una función.

40 Como se apreciará por un experto en la materia, uno o más aspectos de la presente invención pueden realizarse como un sistema, método o producto de programa informático. Por consiguiente, uno o más aspectos de la presente invención pueden tomar la forma de una realización completamente de hardware, una realización completamente de software (que incluye firmware, software residente, micro-código, etc.) o una realización que combina aspectos de software y hardware que pueden todos denominarse de manera general en el presente documento como un "circuito", "módulo" o "sistema". Adicionalmente, uno o más aspectos de la presente invención pueden tomar la forma de un producto de programa informático realizado en uno o más medio o medios legibles por ordenador que tienen código de programa legible por ordenador incorporado en los mismos.

45 Puede utilizarse cualquier combinación de uno o más medio o medios legibles 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 sin limitación, uno electrónico, magnético, óptico, electromagnético, sistema de infrarrojos o de semiconductores, aparatos, o dispositivo, o cualquier combinación adecuada de lo anterior. Ejemplos más específicos (una lista no exhaustiva) del medio de almacenamiento legible por ordenador incluyen lo siguiente: una conexión eléctrica que tiene uno o más alambres, un disquete de ordenador portátil, un disco duro, una memoria de acceso aleatorio (RAM), una memoria de sólo lectura (ROM), una memoria de sólo lectura programable borrrable (EPROM o memoria flash), una fibra óptica, una memoria solo de lectura de disco compacto portátil (CD-ROM), un dispositivo de almacenamiento óptico, un dispositivo de almacenamiento magnético, o cualquier combinación adecuada de lo anterior. En el contexto de este documento, un medio de almacenamiento legible por ordenador puede ser cualquier medio tangible que pueda contener o almacenar un programa para su uso por o en relación con un sistema, aparato, o dispositivo de ejecución de instrucciones.

60 Haciendo referencia ahora a la Figura 11, en un ejemplo, un producto 1100 de programa informático incluye, por ejemplo, uno o más medios 1102 de almacenamiento legibles por ordenador no transitorios para almacenar medios o

lógica 1104 de código de programa legible por ordenador en los mismos para proporcionar y facilitar uno o más aspectos de la presente invención.

El código de programa incorporado en un medio legible por ordenador puede transmitirse usando un medio apropiado, incluyendo, pero sin limitación, inalámbrico, alámbrico, cable de fibra óptica, RF, etc., o cualquier combinación adecuada de lo anterior.

El código de programa informático para llevar a cabo operaciones para uno o más aspectos de la presente invención puede escribirse en cualquier combinación de uno o más lenguajes de programación, incluyendo un lenguaje de programación orientado a objetos, tal como Java, Smalltalk, C++ o similares, y lenguajes de programación procedurales convencionales, tales como el lenguaje de programación "C", ensamblador o lenguajes de programación similares. El código de programa puede ejecutarse completamente en el ordenador del usuario, parcialmente en el ordenador del usuario, como un paquete de software independiente, parcialmente en el ordenador del usuario y parcialmente en un ordenador remoto o completamente en el ordenador o servidor remoto. En el último escenario, el ordenador remoto puede estar conectado al ordenador del usuario a través de un tipo de red, que incluye una red de área local (LAN) o una red de área extensa (WAN), o la conexión puede hacerse a un ordenador externo (por ejemplo, a través de Internet usando un Proveedor de Servicio de Internet).

Uno o más aspectos de la presente invención se describen en el presente documento con referencia unas ilustraciones de diagrama de flujo y/o diagramas de bloques de los métodos, aparato (sistemas) y productos de programa informático de acuerdo con 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, pueden implementarse por instrucciones de programa informático. Estas instrucciones de programa informático pueden proporcionarse a un procesador de un ordenador de fin general, ordenador de fin especial, u otro aparato de procesamiento de datos programable para producir una máquina, de manera que las instrucciones, que se ejecutan mediante el procesador del ordenador u otro aparato de procesamiento de datos programable, crean medios para implementar las funciones/actos especificados en el diagrama de flujo y/o bloque o bloques del diagrama de bloques.

Estas instrucciones de programa informático pueden almacenarse 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 diagrama de flujo y/o bloque o bloques del diagrama de bloques.

Las instrucciones de programa informático pueden cargarse también en un ordenador, otro aparato de procesamiento de datos programable, u otros dispositivos para provocar que se realice una serie de etapas operacionales 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 diagrama de flujo y/o bloque o bloques del diagrama de bloques.

El diagrama de flujo y diagramas de bloques en las figuras ilustran la arquitectura, funcionalidad, y operación de posibles implementaciones de sistemas, métodos y productos de programa informático de acuerdo con diversas realizaciones de uno o más aspectos de la presente invención. En este sentido, cada bloque en el diagrama de flujo o diagramas de bloques puede representar un módulo, segmento, o porción de código, que comprende una o más instrucciones ejecutables para implementar la función o funciones lógicas especificadas. Debería observarse también que, en algunas implementaciones alternativas, las funciones indicadas en el bloque pueden tener lugar fuera del orden indicado en las figuras. Por ejemplo, dos bloques mostrados en serie, de hecho, pueden ejecutarse sustancialmente de manera concurrente, o los bloques pueden ejecutarse en ocasiones en el orden inverso, dependiendo de la funcionalidad implicada. Se entenderá 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, puede implementarse por sistemas basados en hardware de fin especial que realizan las funciones o actos especificados, o combinaciones de hardware de fin especial e instrucciones informáticas.

Además de lo anterior, pueden proporcionarse, ofrecerse, desplegarse, gestionarse, servirse, etc., uno o más aspectos de la presente invención por un proveedor de servicio que ofrece gestión de entornos de cliente. Por ejemplo, el proveedor de servicio 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 servicio puede recibir pago del cliente bajo una suscripción y/o acuerdo de cuota, como ejemplos. Adicionalmente o como alternativa, el proveedor de servicio puede recibir pago de la venta de contenido de anuncio a una o más terceras partes.

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

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

- 5 Como un aspecto adicional más de la presente invención, puede proporcionarse 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 informático comprende uno o más aspectos de la presente invención. El código en combinación con el sistema informático puede realizar uno o más aspectos de la presente invención.

10 Aunque se han descrito anteriormente diversas realizaciones, estas son únicamente ejemplos. Por ejemplo, los entornos informáticos de otras arquitecturas pueden incorporar y usar uno o más aspectos de la presente invención. Además, el periodo de gracia puede ser distinto de una cantidad de tiempo, tal como un número de instrucciones o ciclos o cualquier otro valor cuantificable. Pueden realizarse muchos cambios y/o adiciones sin alejarse del alcance de la presente invención.

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

20 Los dispositivos de entrada/salida o E/S (que incluyen, pero sin limitación, teclados, pantallas, dispositivos apuntadores, DASD, cinta, CD, DVD, unidades de memoria y otros medios de memoria, etc.) pueden acoplarse al sistema ya sea directamente o a través de controladores de E/S intermedios. Los adaptadores de red pueden acoplarse también al sistema para posibilitar que el sistema de procesamiento de datos se acople a otros sistemas de procesamiento de datos o impresoras remotas o dispositivos de almacenamiento a través de redes privadas o públicas intermedias. Los módems, módems de cable, y tarjetas de Ethernet son solamente unos pocos de los tipos disponibles de adaptadores de red.

Otros ejemplos de entornos informáticos que pueden incorporar y/o usar uno o más aspectos de la presente invención se describen a continuación.

35 Haciendo referencia a la Figura 12, se representan componentes representativos de un sistema 5000 de ordenador de anfitrión para implementar uno o más aspectos de la presente invención. El ordenador 5000 de anfitrión representativo comprende una o más CPU 5001 en comunicación con memoria informática (es decir, almacenamiento central) 5002, así como interfaces de E/S a dispositivos 5011 de medios de almacenamiento y redes 5010 para comunicar con otros ordenadores o SAN y similares. La CPU 5001 es compatible con una arquitectura que tiene un conjunto de instrucciones de arquitectura y funcionalidad de arquitectura. La CPU 5001 puede tener traducción de dirección dinámica (DAT) 5003 para transformar direcciones de programa (direcciones virtuales) en direcciones de memoria reales. Una DAT típicamente incluye una memoria intermedia de traducción adelantada (TLB) 5007 para almacenar en caché traducciones de modo que los últimos accesos al bloque de memoria 5002 informática no requieren el retardo de traducción de dirección. Típicamente, se emplea una caché 5009 entre la memoria 5002 informática y el procesador 5001. La caché 5009 puede tener jerárquicamente una gran caché disponible para más de una CPU y cachés más pequeñas, más rápidas (nivel inferior) entre la caché grande y cada CPU. En algunas implementaciones, las cachés de nivel inferior se dividen para proporcionar cachés de nivel bajo separadas para recuperación de instrucciones y accesos de datos. En una realización, se recupera una instrucción de la memoria 5002 por una unidad 5004 de recuperación de instrucción mediante una caché 5009. La instrucción se decodifica en una unidad 5006 de decodificación de instrucción y se despacha (con otras instrucciones en algunas realizaciones) a la unidad o unidades 5008 de ejecución de instrucciones. Típicamente se emplean varias unidades 5008 de ejecución, por ejemplo, una unidad de ejecución aritmética, una unidad de ejecución de punto flotante y una unidad de ejecución de instrucción de ramal. La instrucción se ejecuta por la unidad de ejecución, que accede a los operandos de los registros de instrucción especificados o memoria según sea necesario. Si ha de accederse a un operando (cargarse o almacenarse) de la memoria 5002, una unidad de carga/almacén 5005 típicamente maneja el acceso bajo el control de la instrucción que se está ejecutando. Las instrucciones pueden ejecutarse en circuitos de hardware o en microcódigo interno (firmware) o por una combinación de ambos.

60 Como se indica, un sistema informático incluye información en almacenamiento local (o principal), así como direccionamiento, protección y referencia y grabación de cambio. Algunos aspectos del direccionamiento incluyen el formato de direcciones, el concepto de espacios de dirección, los diversos tipos de direcciones, y la manera en la que un tipo de dirección se traduce a otro tipo de dirección. Alguno del almacenamiento principal incluye localizaciones de almacenamiento asignadas permanentemente. El almacenamiento principal proporciona el sistema con almacenamiento de datos de acceso rápido directamente direccionable. Han de cargarse tanto datos como programas en almacenamiento principal (de dispositivos de entrada) antes de que se procesen.

El almacenamiento principal puede incluir uno o más almacenamientos de memoria intermedia de acceso más rápido más pequeños, en ocasiones denominados cachés. Una caché está asociada físicamente de manera típica con una CPU o un procesador de E/S. Los efectos, excepto en rendimiento, de la construcción física y uso de medios de almacenamiento distintos en general no son observables por el programa.

Las cachés separadas pueden mantenerse para instrucciones y para operandos de datos. La información dentro de una caché se mantiene en bytes contiguos en un límite integral denominado un bloque de caché o línea de caché (o línea, por resumir). Un modelo puede proporcionar una instrucción EXTRAER ATRIBUTO DE CACHÉ que devuelve el tamaño de una línea de caché en bytes. Un modelo puede proporcionar también instrucciones PRE-RECUPERAR DATOS y PRE-RECUPERAR DATOS RELATIVOS LARGO que efectúan la pre recuperación de almacenamiento en la caché de datos o de instrucciones o la liberación de datos de la caché.

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

La información se transmite entre almacenamiento y una CPU o un subsistema de canal un byte, o un grupo de bytes, a la vez. A menos que se especifique de otra manera, en, por ejemplo, la z/Architecture®, un grupo de bytes en almacenamiento se direcciona por el byte más a la izquierda del grupo. El número de bytes en el grupo está implicado o se especifica explícitamente por la operación que va a realizarse. Cuando se usa en una operación de CPU, un grupo de bytes se denomina un campo. Dentro de cada grupo de bytes, en, por ejemplo, la z/Architecture®, se numeran los bits en una secuencia de izquierda a derecha. En la z/Architecture®, los bits más a la izquierda en ocasiones se denominan como los bits "de orden alto" y los bits más a la derecha como los bits "de orden bajo". Sin embargo, los números de bits no son direcciones de almacenamiento. Únicamente pueden direccionarse bytes. Para operar en bits individuales de un byte en almacenamiento, se accede al byte entero. Los bits en un byte se numeran 0 a 7, de izquierda a derecha (en, por ejemplo, la z/Architecture®). Los bits en una dirección pueden numerarse 8-31 o 40-63 para direcciones de 24 bits, o 1-31 o 33-63 para direcciones de 31 bits; se numeran 0-63 para direcciones de 64 bits. Dentro de cualquier otro formato de longitud fija de múltiples bytes, los bits que componen el formato se numeran de manera consecutiva empezando desde 0. Para los fines de detección de error, y preferentemente para su corrección, puede transmitirse 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 controlarse directamente por el programa. Las capacidades de almacenamiento se expresan en número de bytes. Cuando la longitud de un campo de almacenamiento-operando se ve implicada por el código de operación de una instrucción, se dice que el campo tiene una longitud fija, que puede ser uno, dos, cuatro, ocho o dieciséis bytes. Pueden verse implicados campos más grandes para algunas instrucciones. Cuando la longitud de un campo de almacenamiento-operando no se ve implicado sino que se establece explícitamente, el campo se dice que tiene una longitud variable. Los operandos de longitud variable pueden variar en longitud en incrementos de un byte (o con algunas instrucciones, en múltiplos de dos bytes u otros múltiplos). Cuando la información se coloca en almacenamiento, los contenidos de únicamente aquellas localizaciones de bytes que se sustituyen están incluidos en el campo designado, incluso aunque la anchura de la ruta física al almacenamiento pueda ser mayor que la longitud del campo que se está almacenando.

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

En dispositivos que implementan cachés separadas para instrucciones y operandos de datos, puede experimentarse un retardo significativo si el programa almacena en una línea de caché a partir de la cual se recuperan posteriormente instrucciones, independientemente de si el almacenamiento modifica las instrucciones que se recuperan posteriormente.

En una realización, la invención puede ponerse en práctica por software (en ocasiones denominado código interno con licencia, firmware, micro-código, milicode, pico-código y similares, cualquiera de los cuales puede ser consistente con uno o más aspectos la presente invención). Haciendo referencia a la Figura 12, el código de programa de software

que realiza uno o más aspectos de la presente invención puede accederse por el procesador 5001 del sistema anfitrión 5000 de los dispositivos 5011 de medios de almacenamiento a largo plazo, tales como una unidad de CD-ROM, unidad de cinta o unidad de disco. El código de programa de software puede realizarse en cualquiera de una diversidad de medios conocidos para su uso con un sistema de procesamiento de datos, tal como un disquete, disco duro, o CD ROM. El código puede distribuirse en tales medios, o distribuirse a usuarios desde la memoria 5002 informática o almacenamiento de un sistema informático a través de una red 5010 a otros sistemas informáticos para su uso por usuarios de otros sistemas de este tipo.

El código de programa de software incluye un sistema operativo que controla la función e interacción de los diversos componentes informáticos y uno o más programas de aplicación. El código de programa normalmente se pagina desde el dispositivo 5011 de medios de almacenamiento al almacenamiento 5002 informático de velocidad relativamente superior donde está disponible para su procesamiento por el procesador 5001. Las técnicas y métodos para realizar código de programa de software en memoria, en medios físicos y/o distribuir código de software mediante redes son bien conocidos y no se analizarán adicionalmente en el presente documento. El código de programa, cuando se crea y almacena en un medio de almacenamiento (que incluye, pero sin limitación, módulos de memoria electrónica (RAM), memoria flash, Discos Compactos (CD), DVD, Cinta Magnética y similares a menudo se denomina como un "producto de programa informático". El medio de producto de programa informático es típicamente legible por un circuito de procesamiento preferentemente en un sistema informático para su ejecución por el circuito de procesamiento.

La Figura 13 ilustra una estación de trabajo o servidor representativo en el que pueden ponerse en práctica uno o más aspectos de la presente invención. El sistema 5020 de la Figura 13 comprende un sistema 5021 informático base representativo, tal como un ordenador personal, una estación de trabajo o un servidor, que incluye dispositivos periféricos opcionales. El sistema 5021 informático base incluye uno o más procesadores 5026 y un bus empleado para conectar y posibilitar la comunicación entre el procesador o procesadores 5026 y los otros componentes del sistema 5021 de acuerdo con técnicas conocidas. El bus conecta el procesador 5026 a la memoria 5025 y al almacenamiento a largo plazo 5027 que puede incluir un disco duro (que incluye cualquiera de medios magnéticos, CD, DVD y Memoria Flash, por ejemplo) o una unidad de cinta por ejemplo. El sistema 5021 puede incluir también un adaptador de interfaz de usuario, que conecta el microprocesador 5026 mediante el bus a uno o más dispositivos de interfaz, tales como un teclado 5024, un ratón 5023, una impresora/escáner 5030 y/u otros dispositivos de interfaz, que puede ser cualquier dispositivo de interfaz de usuario, tal como una pantalla táctil, almohadilla de entrada digitalizada. El bus también conecta un dispositivo 5022 de visualización, tal como una pantalla de LCD o monitor, al microprocesador 5026 mediante un adaptador de pantalla.

El sistema 5021 puede comunicar con otros ordenadores o redes informáticas por medio de un adaptador de red que puede comunicar 5028 con una red 5029. Adaptadores de red ejemplares son canales de comunicación, anillo con paso de testigo, Ethernet o módems. Como alternativa, el sistema 5021 puede comunicar usando una interfaz inalámbrica, tal como una tarjeta de CDPD (datos de paquetes digitales celulares). El sistema 5021 puede asociarse con tales otros ordenadores en una red de área local (LAN) o una red de área extensa (WAN), o el sistema 5021 puede ser un cliente en una disposición de cliente/servidor con otro ordenador, etc. Todas estas configuraciones, así como el hardware y software de comunicaciones apropiadas, son conocidos en la técnica.

La Figura 14 ilustra una red 5040 de procesamiento de datos en la que puede ponerse en práctica uno o más aspectos de la presente invención. La red 5040 de procesamiento de datos puede incluir una pluralidad de redes individuales, tales como una red inalámbrica y una red alámbrica, cada una de las cuales puede incluir una pluralidad de estaciones de trabajo 5041, 5042, 5043, 5044 individuales. Adicionalmente, como apreciarán los expertos en la materia, puede incluirse una o más LAN, donde una LAN puede comprender una pluralidad de estaciones de trabajo inteligentes acopladas a un procesador de anfitrión.

Haciendo referencia aún a la Figura 14, las redes pueden incluir también ordenadores centrales o servidores, tales como un ordenador de pasarela (servidor cliente 5046) o servidor de aplicación (servidor remoto 5048 que puede acceder a un repositorio de datos y puede accederse también directamente desde una estación de trabajo 5045). Un ordenador 5046 de pasarela sirve como un punto de entrada en cada red individual. Es necesaria una pasarela cuando se conecta un protocolo de interconexión de red a otro. La pasarela 5046 puede acoplarse preferentemente a otra red (la Internet 5047 por ejemplo) por medio de un enlace de comunicaciones. La pasarela 5046 puede estar directamente acoplada a una o más estaciones de trabajo 5041, 5042, 5043, 5044 usando un enlace de comunicaciones. El ordenador de pasarela puede implementarse usando un servidor IBM eServer™ System z® disponible a partir de International Business Machines Corporation.

Haciendo referencia de manera concurrente a la Figura 13 y a la Figura 14, el código de programación de software que puede incorporar uno o más aspectos de la presente invención puede accederse por el procesador 5026 del sistema 5020 de los medios 5027 de almacenamiento a largo plazo, tales como una unidad de CD-ROM o unidad de disco. El código de programación de software puede realizarse en cualquiera de una diversidad de medios conocidos para su uso con un sistema de procesamiento de datos, tal como un disquete, disco duro, o CD ROM. El código puede distribuirse en tales medios, o puede distribuirse a los usuarios 5050, 5051 de la memoria o almacenamiento de un sistema informático a través de una red a otros sistemas informáticos para su uso por usuarios de tales otros sistemas.

Como alternativa, el código de programación puede realizarse en la memoria 5025, y accederse por el procesador 5026 usando el bus de 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 5032 de aplicación. El código de programa normalmente está paginado desde los medios 5027 de almacenamiento a memoria 5025 de alta velocidad donde está disponible para su procesamiento por el procesador 5026. Las técnicas y métodos para realizar código de programación de software en memoria, en medios físicos y/o distribuir código de software mediante redes son bien conocidos y no se analizarán adicionalmente en el presente documento. El código de programa, cuando se crea y almacena en un medio de almacenamiento (que incluye, pero sin limitación, módulos de memoria electrónica (RAM), memoria flash, Discos Compactos (CD), DVD, Cinta Magnética y similares a menudo se denomina como un "producto de programa informático". El medio de producto de programa informático es típicamente legible por un circuito de procesamiento preferentemente en un sistema informático para su ejecución por el circuito de procesamiento.

La caché está en su mayoría fácilmente disponible para el procesador (normalmente más rápidas y más pequeñas que otras cachés del procesador) es la caché más inferior (L1 o nivel uno) y el almacén principal (memoria principal) es la caché de nivel más alta (L3 si hay 3 niveles). La caché de nivel más baja a menudo se divide en una caché de instrucciones (I-caché) que mantiene instrucciones de máquina a ejecutarse y una caché de datos (D-caché) que maneja operandos de datos.

Haciendo referencia a la Figura 15, una realización de procesador ejemplar se representa para el procesador 5026. Típicamente se emplean uno o más niveles de caché 5053 para almacenar en memoria intermedia bloques para mejorar el rendimiento de procesador. La caché 5053 es una memoria intermedia de alta velocidad que mantiene líneas de caché de datos de memoria que es probable que se usen. Las líneas de caché típicas son 64, 128 o 256 bytes de datos de memoria. Las cachés de separadas a menudo se emplean para almacenar en caché instrucciones que para almacenar en caché datos. La coherencia de caché (sincronización de copias de líneas en memoria y las cachés) a menudo se proporciona por diversos algoritmos "sondeo" bien conocidos en la técnica. El almacén 5025 de memoria principal de un sistema de procesador a menudo se denomina como una caché. En un sistema de procesador que tiene 4 niveles de caché 5053, un almacén 5025 principal a menudo se denomina como la caché de nivel 5 (L5) puesto que típicamente es más rápido y únicamente mantiene una porción del almacenamiento no volátil (DASD, cinta, etc.) que está disponible para un sistema informático. El almacén 5025 principal "almacena en caché" páginas de datos paginados en y fuera del almacén 5025 principal por el sistema operativo. Un contador de programa (contador de instrucción) 5061 mantiene el seguimiento de la dirección de la instrucción actual que va a ejecutarse. Un contador de programa en un procesador z/Architecture® es de 64 bits y puede truncarse a 31 o 24 bits para soportar límites de direccionamiento anterior. Un contador de programa se realiza típicamente en una PSW (palabra de estado de programa) de un ordenador de manera que persiste durante la conmutación de contexto. Por lo tanto, un programa en progreso, que tiene un valor de contador de programa, puede interrumpirse por, por ejemplo, el sistema operativo (conmutación de contexto del entorno de programa al entorno del sistema operativo). La PSW del programa mantiene el valor de contador de programa mientras el programa no está activo, y se usa el contador de programa (en la PSW) del sistema operativo mientras se está ejecutando el sistema operativo. Típicamente, el contador de programa se incrementa por una cantidad igual al número de bytes de la instrucción actual. Las instrucciones RISC (Informática de Conjunto Reducido de Instrucciones) típicamente están fijadas en longitud mientras que las instrucciones CISC (Informática de Conjunto Complejo de Instrucciones) típicamente tienen longitud variable. Las instrucciones de IBM z/Architecture® son instrucciones CISC que tienen una longitud de 2, 4 o 6 bytes. El contador de programa 5061 se modifica por cualquiera de una operación de conmutación de contexto o una operación de ramal tomada de una instrucción de ramal, por ejemplo. En una operación de conmutación de contexto, se graba el valor de contador de programa actual en la palabra de estado de programa junto con otra información de estado acerca del programa que se está ejecutando (tal como códigos de condición), y se carga un nuevo valor de contador de programa que apunta a una instrucción de un nuevo módulo de programa para que se ejecute. Se realiza una operación de ramal tomado para permitir que el programa haga decisiones o realice bucles dentro del programa cargando el resultado de la instrucción de ramal en el contador de programa 5061.

Típicamente se emplea una unidad 5055 de recuperación de instrucción para recuperar instrucciones en nombre del procesador 5026. La unidad de recuperación recupera "siguientes instrucciones secuenciales", instrucciones objetivo de instrucciones de ramal tomado, o primeras instrucciones de un programa que sigue una conmutación de contexto. Las unidades de recuperación de instrucción modernas a menudo emplean técnicas de pre recuperación para pre recuperar de manera especulativa instrucciones basándose en la probabilidad de que las instrucciones pre recuperadas puedan usarse. Por ejemplo, una unidad de recuperación puede recuperar 16 bytes de instrucciones que incluyen la siguiente instrucción secuencial y bytes adicionales de instrucciones secuenciales adicionales.

Las instrucciones recuperadas se ejecutan a continuación por el procesador 5026. En una realización, la instrucción o instrucciones recuperadas se pasan a una unidad 5056 de despacho de la unidad de recuperación. La unidad de despacho decodifica la instrucción o instrucciones y reenvía información acerca de la instrucción o instrucciones decodificadas a las unidades 5057, 5058, 5060 apropiadas. Una unidad 5057 de ejecución recibirá típicamente información acerca de instrucciones aritméticas decodificadas de la unidad 5055 de recuperación de instrucción y realizará operaciones aritméticas en operandos de acuerdo con el opcode de la instrucción. Los operandos se proporcionan a la unidad 5057 de ejecución preferentemente desde la memoria 5025, los registros 5059 de arquitectura desde un campo inmediato de la instrucción que se está ejecutando. Los resultados de la ejecución,

cuando se almacenan, se almacenan en la memoria 5025, registros 5059 o 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. Haciendo referencia a la Figura 16A, una unidad 5057 de ejecución puede comunicar con registros 5059 generales de arquitectura, una unidad 5056 de decodificación/despacho, una unidad 5060 de almacén de carga, y otras 5065 unidades de procesador por medio de la lógica 5071 de interconexión. Una unidad 5057 de ejecución puede emplear varios circuitos 5067, 5068, 5069 de registro para mantener la información en la que operará la unidad aritmético lógica (ALU) 5066. La ALU realiza operaciones aritméticas tales como añadir, restar, multiplicar y dividir así como la función lógica tal como y, o y o exclusiva (XOR), rotar y desplazar. Preferentemente la ALU soporta operaciones especializadas que son dependientes del diseño. Otros circuitos pueden proporcionar otras funciones 5072 de arquitectura que incluyen códigos de condición y recuperar la lógica de soporte por ejemplo. Típicamente el resultado de una operación de ALU se mantiene en un circuito 5070 de registro de salida que puede reenviar el resultado a una diversidad de otras funciones de procesamiento. Hay muchas disposiciones de unidades de procesador, la presente descripción se pretende únicamente para proporcionar un entendimiento representativo de una realización.

Una instrucción ADD, por ejemplo, se ejecutaría en una unidad 5057 de ejecución 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. Preferentemente, una unidad de ejecución opera en operandos identificados por una instrucción realizando una función definida opcode en los operandos. Por ejemplo, una instrucción ADD puede ejecutarse por una unidad 5057 de ejecución en operandos hallando en dos registros 5059 identificados por campos de registro de la instrucción.

La unidad 5057 de ejecución realiza la adición aritmética en 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 de origen. La unidad de ejecución utiliza preferentemente una Unidad Aritmético Lógica (ALU) 5066 que puede realizar una diversidad de funciones lógicas tales como Desplazar, Rotar, y, o y XOR así como una diversidad de funciones algebraicas que incluyen cualquiera de añadir, restar, multiplicar y dividir. Algunas ALU 5066 están diseñadas para operaciones escalares y algunas para punto flotante. Los datos pueden ser Big Endian (donde el byte menos significativo es en la dirección de byte más alta) o Little Endian (donde el byte menos significativo es en la dirección de byte más baja) dependiendo de la arquitectura. IBM z/Architecture® es Big Endian. Los campos con signo pueden ser de signo y magnitud, en complemento de 1 o en complemento de 2 dependiendo de la arquitectura. Un número de complemento de 2 es ventajoso en que la ALU no necesita diseñar una capacidad de resta puesto que un valor negativo o un valor positivo en complemento de 2 requiere únicamente una adición dentro de la ALU. Los números se describen comúnmente en forma abreviada, donde un campo de 12 bits define una dirección de un bloque de 4096 bytes y se describe comúnmente como un bloque de 4 Kbytes (Kilobytes), por ejemplo.

Haciendo referencia a la Figura 16B, la información de instrucción de ramal para ejecutar una instrucción de ramal se envía típicamente a una unidad 5058 de ramal que a menudo emplea un algoritmo de predicción de ramal tal como una tabla 5082 de historial de ramal para predecir el resultado del ramal antes de que se completen las otras operaciones condicionales. El objetivo de la instrucción de ramal actual se recuperará y se ejecutará de manera especulativa antes de que se completen las operaciones condicionales. Cuando se completan las operaciones condicionales, las instrucciones de ramal ejecutadas de manera especulativa se completan o descartan basándose en las condiciones de la operación condicional y el resultado especulado. Una instrucción de ramal típica puede probar los códigos de condición y ramal a una dirección objetivo si los códigos de condición cumplen el requisito de ramal de la instrucción de ramal, puede calcularse una dirección objetivo basándose en varios números que incluyen unos hallados en campos de registro o un campo inmediato de la instrucción, por ejemplo. La unidad 5058 de ramal puede emplear una ALU 5074 que tiene una pluralidad de circuitos 5075, 5076, 5077 de registro de entrada y un circuito 5080 de registro de salida. La unidad 5058 de ramal puede comunicar con registros 5059 generales, unidad 5056 de despacho de decodificación u otros circuitos 5073, por ejemplo.

La ejecución de un grupo de instrucciones puede interrumpirse por una diversidad de razones que incluyen una conmutación de contexto iniciada por un sistema operativo, una excepción de programa o error que provoca una conmutación de contexto, una señal de interrupción de E/S que provoca una conmutación de contexto o actividad de múltiples hilos de una pluralidad de programas (en un entorno de múltiples hilos), por ejemplo. Preferentemente una acción de conmutación de contexto graba información de estado acerca de un programa actualmente en ejecución y a continuación carga información de estado acerca de otro programa que se está invocando. La información de estado puede grabarse en registros de hardware o en memoria, por ejemplo. La información de estado preferentemente comprende un valor de contador de programa que apunta a una siguiente instrucción a ejecutarse, códigos de condición, información de traducción de memoria y contenido de registro de arquitectura. Una actividad de conmutación de contexto puede ejercerse por circuitos de hardware, programas de aplicación, programas de sistema operativo o código de firmware (microcódigo, pico-código o código interno con licencia (LIC)) en solitario o en combinación.

Un procesador accede a operandos de acuerdo con métodos de instrucción definida. La instrucción puede proporcionar un operando inmediato que usa el valor de una porción de la instrucción, puede proporcionar uno o más

campos de registro que apuntan explícitamente a cualesquiera registros de fin general o registros de fin especial (registros de punto flotante, por ejemplo). La instrucción puede utilizar registros implicados identificados por un campo opcode como operandos. La instrucción puede utilizar localizaciones de memoria para operandos. Una localización de memoria de un operando puede proporcionarse por un registro, un campo inmediato, o una combinación de registros y campo inmediato como se ejemplifica por la función de desplazamiento largo de z/Architecture® en la que la instrucción define un registro de 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 localización en el presente documento típicamente implica una localización en memoria principal (almacenamiento principal) a menos que se indique de otra manera.

Haciendo referencia a la Figura 16C, un procesador accede a almacenamiento usando una unidad 5060 de carga/almacén. La unidad 5060 de carga/almacén puede realizar una operación de carga obteniendo la dirección del operando objetivo en memoria 5053 y cargar el operando en un registro 5059 u otra localización de memoria 5053, o puede realizar una operación de almacén obteniendo la dirección del operando objetivo en memoria 5053 y almacenar datos obtenidos de un registro 5059 u otra localización de memoria 5053 en la localización de operando objetivo en memoria 5053. La unidad 5060 de carga/almacén puede ser especulativa y puede acceder a memoria en una secuencia que está desordenada con relación a la secuencia de instrucción sin embargo, la unidad 5060 de carga/almacén es para mantener la apariencia a programas en los que se ejecutaron instrucciones en orden. Una unidad 5060 de carga/almacén puede comunicar con registros 5059 generales, unidad 5056 de decodificación/despacho, interfaz 5053 de caché/memoria u otros elementos 5083 y comprende diversos circuitos de registro, ALU 5085 y lógica 5090 de control para calcular direcciones de almacenamiento y para proporcionar la generación de secuencias en tuberías para mantener las operaciones en orden. Algunas operaciones pueden estar desordenadas pero la unidad de carga/almacén proporciona funcionalidad para hacer que las operaciones desordenadas aparezcan para el programa como que se han realizado en orden, como es bien conocido en la técnica.

Preferentemente las direcciones que un programa de aplicación "observa" a menudo se denominan como direcciones virtuales. Las direcciones virtuales se denominan en ocasiones como "direcciones lógicas" y "direcciones efectivas". Estas direcciones virtuales son virtuales en que se redirigen a la localización de memoria física por una de una diversidad de tecnologías de traducción de dirección dinámica (DAT) que incluyen, pero sin limitación, prefijar simplemente una dirección virtual con un valor de desplazamiento, traducir la dirección virtual mediante una o más tablas de traducción, las tablas de traducción preferentemente que comprenden al menos una tabla de segmentos y una tabla de página en solitario o en combinación, preferentemente, la tabla de segmentos que tiene una entrada que apunta a la tabla de páginas. En 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 dirección a menudo se mejora utilizando una memoria intermedia de traducción adelantada (TLB) que comprende entradas que mapean una dirección virtual a una localización de memoria física asociada. Las entradas se crean cuando la DAT traduce una dirección virtual usando las tablas de traducción. Posterior al uso de la dirección virtual pueden a continuación utilizar la entrada de la TLB rápida en lugar de los accesos de tabla de traducción secuencial lenta. El contenido de la TLB puede gestionarse por una diversidad de algoritmos de sustitución que incluyen LRU (Menos Recientemente usado).

En el caso donde el procesador es un procesador de un sistema multi-procesador, cada procesador tiene la responsabilidad de mantener recursos compartidos, tal como E/S, cachés, TLB y memoria, interbloqueado por coherencia. Típicamente, las tecnologías de "sondeo" se utilizarán al mantener la coherencia de caché. En un entorno de sondeo, cada línea de caché puede marcarse como que está en uno cualquiera de un estado compartido, un estado exclusivo, un estado cambiado, un estado inválido y similares para facilitar la compartición.

Las unidades 5054 de E/S (Figura 15) proporcionan al procesador con medios para fijarse a dispositivos periféricos que incluyen cinta, disco, impresoras, pantallas y redes, por ejemplo. Las unidades de E/S a menudo se presentan al programa informático por controladores de software. En ordenadores principales, tales como System z® de IBM®, los adaptadores de canal y adaptadores de sistema abiertos son unidades de E/S del ordenador principal que proporcionan las comunicaciones entre el sistema operativo y dispositivos periféricos.

Además, otros tipos de entornos informáticos pueden beneficiarse de uno o más aspectos de la presente invención. Como un ejemplo, como se menciona en el presente documento, un entorno puede incluir un emulador (por ejemplo, software u otros mecanismos de emulación), en el que una arquitectura particular (que incluye, por ejemplo, ejecución de instrucción, funciones de arquitectura, tal como traducción de dirección, y registros de arquitectura) o un subconjunto de las mismas se emula (por ejemplo, en un sistema informático nativo que tiene un procesador y memoria). En un entorno de este tipo, 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 puede tener una arquitectura diferente de las capacidades que se están emulando. Como un ejemplo, en modo de emulación, se decodifica la instrucción específica u operación que se está emulando, y se crea una función de emulación apropiada para implementar la instrucción u operación individual.

En un entorno de emulación, un ordenador de anfitrión incluye, por ejemplo, una memoria para almacenar instrucciones y datos; una unidad de recuperación de instrucción para recuperar instrucciones de memoria y para

proporcionar, opcionalmente, almacenamiento en memoria intermedia local para la instrucción recuperada; una unidad de decodificación de instrucción para recibir instrucciones recuperadas y para determinar el tipo de instrucciones que se han recuperado; y una unidad de ejecución de instrucción para ejecutar las instrucciones. La ejecución puede incluir datos de carga en un registro de la memoria; almacenar datos de vuelta a la memoria de 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 de emulador.

Más particularmente, en un ordenador central, las instrucciones de máquina de arquitectura se usan por programadores, normalmente hoy en día programadores de "C", a menudo por medio de una aplicación de compilador. Estas instrucciones almacenadas en el medio de almacenamiento pueden ejecutarse de manera nativa en un Servidor de z/Architecture® IBM®, o como alternativa en máquinas al ejecutar otras arquitecturas. Pueden emularse en los servidores de ordenadores centrales de IBM® existentes y futuros y en otras máquinas de IBM® (por ejemplo, servidores de Power Systems y servidores System x®). Pueden ejecutarse en máquinas que ejecutan Linux en una amplia diversidad de máquinas usando hardware fabricado por IBM®, Intel®, AMD™, y otras. Además de la ejecución en ese hardware bajo una z/Architecture®, puede usarse Linux así como máquinas que usan emulación por Hercules, UMX, o FSI (Fundamental Software, Inc), donde la ejecución en general es en un modo de emulación. En modo de 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 cualquiera de firmware o un sistema operativo nativo para realizar la emulación del procesador emulado. El software de emulación es responsable de recuperar y ejecutar instrucciones de la arquitectura de procesador emulado. El software de emulación mantiene un contador de programa emulado para mantener el seguimiento de límites de instrucción. El software de emulación puede recuperar una o más instrucciones de máquina emuladas a la vez y convertir la una o más instrucciones de máquina emuladas a un correspondiente grupo de instrucciones de máquina nativas para su ejecución por el procesador nativo. Estas instrucciones convertidas pueden almacenarse en caché de manera que puede conseguirse una conversión más rápida. No obstante, el software de emulación es para mantener las reglas de arquitectura de la arquitectura de procesador emulado para asegurar que los sistemas operativos y aplicaciones escritos para el procesador emulado operan correctamente. Adicionalmente, el software de emulación es para proporcionar recursos identificados por la arquitectura de procesador emulado que incluye, pero sin limitación, registros de control, registros de fin general, registros de punto flotante, función de traducción de dirección dinámica que incluye tablas de segmento y tablas de página por ejemplo, mecanismos de interrupción, mecanismos de conmutación de contexto, relojes de hora del día (TOD) e interfaces de arquitectura a subsistemas de E/S de manera que un sistema operativo o un programa de aplicación designado para ejecutarse en el procesador emulado, puede ejecutarse en el procesador nativo que tiene el software de emulación.

Se decodifica una instrucción específica que se está emulando, y se solicita 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 una subrutina o controlador de "C", o algún otro método de suministro de un controlador para el hardware específico como estará dentro de la habilidad de los expertos en la materia después de entender la descripción de la realización preferida. Diversas patentes de emulación de software y hardware que incluyen, pero sin limitación la Patente de Cartas de Estados Unidos N.º 5.551.013, titulada "Multiprocessor for Hardware Emulation", por Beausoleil et al.; y Patente de Cartas de Estados Unidos N.º 6.009.261, titulada "Preprocessing of Stored Target Routines for Emulating Incompatible Instructions on a Target Processor", por Scalzi et al; y Patente de Cartas de Estados Unidos N.º 5.574.873, titulada "Decoding Guest Instruction to Directly Access Emulation Routines that Emulate the Guest Instructions", por Davidian et al; y Patente de Cartas de Estados Unidos N.º 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 Patente de Cartas de Estados Unidos N.º 6.463.582, titulada "Dynamic Optimizing Object Code Translator for Architecture Emulation and Dynamic Optimizing Object Code Translation Method", por Lethin et al; y Patente de Cartas de Estados Unidos N.º 5.790.825, titulada "Method for Emulating Guest Instructions on a Host Computer Through Dynamic Recompilation of Host Instructions", por Eric Traut, y muchas otras, ilustran una diversidad de maneras conocidas para conseguir la emulación de un formato de instrucción de arquitectura para una máquina diferente para una máquina objetivo disponible para aquellos expertos en la materia.

En la Figura 17, se proporciona un ejemplo de un sistema 5092 informático de anfitrión emulado que emula un sistema 5000' ordenador de anfitrión de una arquitectura de anfitrión. En el sistema 5092 informático de anfitrión emulado, el procesador de anfitrión (CPU) 5091 es un procesador de anfitrión emulado (o procesador de anfitrión virtual) y comprende un procesador 5093 de emulación que tiene una arquitectura de conjunto de instrucciones nativas diferentes que las del procesador 5091 del ordenador 5000' anfitrión. El sistema 5092 informático de anfitrión emulado tiene la memoria 5094 accesible al procesador 5093 de emulación. En la realización de ejemplo, la memoria 5094 se particiona en una porción de memoria 5096 de ordenador de anfitrión y una porción de rutinas 5097 de emulación. La memoria 5096 ordenador anfitrión está disponible para programas del ordenador 5092 de anfitrión emulado de acuerdo con la arquitectura del ordenador de anfitrión. El procesador 5093 de emulación ejecuta instrucciones nativas de un conjunto de instrucciones de arquitectura de una arquitectura distinta de la del procesador 5091 emulado, las instrucciones nativas obtenidas de la memoria 5097 de rutinas de emulación, y puede acceder a una instrucción de

anfitrión para la ejecución de un programa en memoria 5096 informática de anfitrión empleando una o más instrucción o instrucciones obtenidas en una secuencia y rutina de acceso/decodificación que puede decodificar la instrucción o instrucciones de anfitrión accedidas para determinar una rutina de ejecución de instrucción nativa para emular la función de la instrucción de anfitrión accedida. Otras funciones que se definen para la arquitectura del sistema 5000' informático de anfitrión pueden emularse mediante rutinas de funciones de arquitectura, que incluyen tales funciones como registros de fin general, registros de control, traducción de dirección dinámica y caché de soporte y procesador de subsistema de E/S, por ejemplo. Las rutinas de emulación pueden aprovecharse también de funciones disponibles en el procesador 5093 de emulación (tal como registros generales y traducción dinámica de direcciones virtuales) para mejorar el rendimiento de las rutinas de emulación. Los motores de hardware especial y de descarga pueden proporcionarse también para ayudar al procesador 5093 a emular la función del ordenador 5000' de anfitrión.

La terminología usada en el presente documento es para el fin de describir realizaciones particulares únicamente y no se pretende que sea para limitar la invención. Como se usa en el presente documento, las formas singulares "un", "una" y "el", "la" se pretende que incluyan las formas plurales también, a menos que el contexto lo indique claramente de otra manera. Se entenderá adicionalmente que los términos "comprende" y/o "comprendiendo/que comprende", cuando se usan en esta memoria descriptiva, especifican la presencia de características indicadas, elementos integrantes, etapas, operaciones, elementos y/o componentes, pero no excluyen la presencia o adición de una o más otras características, elementos integrantes, etapas, operaciones, elementos, componentes y/o grupos de los mismos.

Las correspondientes estructuras, materiales, actos y equivalentes de todos los significados o elementos de etapa más función en las reivindicaciones a continuación, si las hubiera, se pretende que incluyan cualquier estructura, material o acto para realizar la función en combinación con otros elementos reivindicados como se reivindica específicamente. La descripción de uno o más aspectos de la presente invención se ha presentado para fines de ilustración y descripción, pero no se pretende que sea exhaustiva o limite a la invención en la forma desvelada. Serán evidentes muchas modificaciones y variaciones para los expertos en la materia sin alejarse del alcance de la invención. La realización se eligió y describió para explicar mejor los principios de la invención y la aplicación práctica, y para posibilitar a otros expertos en la materia entender la invención para diversas realizaciones con diversas modificaciones como que son adecuadas al uso particular contemplado. Las realizaciones que no caen dentro del alcance de las reivindicaciones han de interpretarse simplemente como ejemplos útiles para entender la invención.

REIVINDICACIONES

1. Un sistema informático para facilitar el procesamiento en un entorno informático, comprendiendo dicho sistema informático:

una memoria; y

un procesador de anfitrión y uno o más procesadores virtuales, siendo el uno o más procesadores virtuales un modo diferente del procesador de anfitrión, el procesador de anfitrión en comunicación con la memoria, en el que el sistema informático está configurado para realizar un método, dicho método caracterizado por comprender:

proporcionar por un sistema operativo anfitrión a un sistema operativo de invitado que se ejecuta en el uno o más procesadores virtuales una indicación de función de seguimiento de advertencias instalada que indica la instalación de una función de seguimiento de advertencias en el entorno informático, la indicación de instalación que va a obtenerse por el sistema operativo de invitado, y la función de seguimiento de advertencias para proporcionar al sistema operativo de invitado un periodo de gracia para realizar una primera función, siendo el periodo de gracia una duración predeterminada no ampliable, en el que la primera función es una de finalizar la ejecución de una unidad despachable, detener la unidad despachable en un punto en el que la unidad despachable es re-despachable, y/o mover la unidad despachable a otro procesador invitado; obtener, por el sistema operativo anfitrión basándose en el suministro de la indicación de que está instalada una función de seguimiento de advertencias, una indicación no solicitada de que el sistema operativo de invitado se ha registrado para la función de seguimiento de advertencias, comprendiendo el registro una indicación al sistema operativo anfitrión de que el sistema operativo de invitado entiende un protocolo de la función de seguimiento de advertencias y pretende participar en la función de seguimiento de advertencias de manera que el sistema operativo de invitado respetaría normalmente la notificación haciendo a la unidad despachable actual despachable en otra CPU invitada de la configuración de invitado cuando sea necesario; posibilitar, basándose en la obtención de la indicación de que el sistema operativo de invitado se ha registrado, la función de seguimiento de advertencias para el sistema operativo de invitado; proporcionar por el sistema operativo anfitrión al sistema operativo de invitado una notificación de seguimiento de advertencias; basándose en la notificación de seguimiento de advertencias, iniciar por el sistema operativo de invitado la primera función dentro del periodo de gracia; y realizar por el sistema operativo anfitrión una segunda función posterior al periodo de gracia.

2. El sistema informático de la reivindicación 1, en el que la notificación de seguimiento de advertencias comprende una interrupción en la que los recursos compartidos asignados al sistema operativo de invitado se liberan después de la terminación del periodo de gracia.

3. El sistema informático de la reivindicación 1, el sistema operativo de invitado tiene acceso a recursos compartidos del entorno informático durante un segmento de tiempo proporcionado a una unidad de procesamiento central de invitado en la que se ejecuta el sistema operativo de invitado, siendo distinguible el periodo de gracia del segmento de tiempo.

4. El sistema informático de la reivindicación 3, en el que el periodo de gracia termina prematuramente el segmento de tiempo.

5. El sistema informático de la reivindicación 3, en el que el periodo de gracia proporciona un periodo además del segmento de tiempo en el que realizar la primera función.

6. El sistema informático de la reivindicación 1, en el que la primera función comprende una de:

finalizar una unidad despachable que se ejecuta en un procesador en el que se ejecuta el sistema operativo de invitado; o
hacer la unidad despachable re-despachable en otro procesador del entorno informático.

7. El sistema informático de la reivindicación 1, en el que el método comprende adicionalmente:

indicar por el sistema operativo de invitado la finalización de la primera función durante el periodo de gracia; y proporcionar una indicación al sistema operativo de invitado una siguiente vez que el sistema operativo de invitado ejecuta esa primera función completada durante el periodo de gracia.

8. Un método de facilitación del procesamiento en un entorno informático, comprendiendo dicho método y caracterizado por:

proporcionar por un sistema operativo anfitrión a un sistema operativo de invitado que se ejecuta en el uno o más procesadores virtuales una indicación de función de seguimiento de advertencias instalada que indica la instalación de una función de seguimiento de advertencias en el entorno informático, la indicación de instalación que va a

obtenerse por el sistema operativo de invitado, y la función de seguimiento de advertencias para proporcionar al sistema operativo de invitado un periodo de gracia para realizar una primera función, siendo el periodo de gracia una duración predeterminada no ampliable, en el que la primera función es una de finalizar la ejecución de una unidad despachable, detener la unidad despachable en un punto en el que la unidad despachable es re-despachable, y/o mover la unidad despachable a otro procesador invitado;

obtener, por el sistema operativo anfitrión basándose en el suministro de la indicación de que está instalada una función de seguimiento de advertencias, una indicación no solicitada de que el sistema operativo de invitado se ha registrado para la función de seguimiento de advertencias, comprendiendo el registro una indicación al sistema operativo anfitrión de que el sistema operativo de invitado entiende un protocolo de la función de seguimiento de advertencias y pretende participar en la función de seguimiento de advertencias de manera que el sistema operativo de invitado respetaría normalmente la notificación haciendo a la unidad despachable actual despachable en otra CPU invitada de la configuración de invitado cuando sea necesario;

posibilitar, basándose en la obtención de la indicación de que el sistema operativo de invitado se ha registrado, la función de seguimiento de advertencias para el sistema operativo de invitado;

proporcionar por el sistema operativo anfitrión al sistema operativo de invitado una notificación de seguimiento de advertencias;

basándose en la notificación de seguimiento de advertencias, iniciar por el sistema operativo de invitado la primera función dentro del periodo de gracia; y

realizar por el sistema operativo anfitrión una segunda función posterior al periodo de gracia.

9. El método de la reivindicación 8, en el que el sistema operativo de invitado tiene acceso a recursos compartidos del entorno informático durante un segmento de tiempo proporcionado a una unidad de procesamiento central de invitado en la que se ejecuta el sistema operativo de invitado, siendo distinguible el periodo de gracia del segmento de tiempo.

10. El método de la reivindicación 8, que comprende adicionalmente:

indicar por el sistema operativo de invitado la finalización de la primera función durante el periodo de gracia; y proporcionar una indicación al sistema operativo de invitado una siguiente vez que el sistema operativo de invitado ejecuta esa primera función completada durante el periodo de gracia.

11. El método de la reivindicación 8, en el que el registro se realiza mediante una instrucción de diagnóstico, comprendiendo la instrucción de diagnóstico un parámetro de registro para señalar que el sistema operativo de invitado entiende una interrupción de seguimiento de advertencias de la función de seguimiento de advertencias.

12. El método de la reivindicación 8, en el que el periodo de gracia es distinguible de un segmento de tiempo proporcionado a un procesador en el que se ejecuta el sistema operativo de invitado, teniendo acceso el sistema operativo de invitado a recursos compartidos del entorno informático durante el segmento de tiempo, y en el que el periodo de gracia proporciona un periodo además del segmento de tiempo, en el que cualquier porción del periodo usado por el sistema operativo de invitado para realizar la primera función se carga contra un siguiente segmento de tiempo para el sistema operativo de invitado.

13. El método de la reivindicación 8, en el que el método comprende adicionalmente:

determinar si el sistema operativo de invitado sale voluntariamente antes del agotamiento del periodo de gracia; basándose en la determinación de que el sistema operativo de invitado salió voluntariamente antes del agotamiento del periodo de gracia, proporcionar al sistema operativo de invitado, una siguiente vez que se ejecuta el sistema operativo de invitado, realimentación positiva que indica la salida voluntaria antes del agotamiento del periodo de gracia; y basándose en la determinación de que el sistema operativo de invitado llega tarde en la salida voluntaria antes de la expiración del periodo de gracia, proporcionar al sistema operativo de invitado, una siguiente vez que se ejecuta el sistema operativo de invitado, una indicación de realimentación de excepción que indica tardanza en la salida voluntaria.

14. El método de la reivindicación 8, en el que el método comprende adicionalmente:

determinar, basándose en una condición de interrupción de temporizador de un temporizador de sistema operativo anfitrión o una solicitud de intervención de seguimiento de advertencia, si se establece un indicador de control activo de periodo de gracia; basándose en que no se establezca el indicador de control activo de periodo de gracia, establecer el indicador de control activo de periodo de gracia; grabar un valor actual del temporizador de sistema operativo anfitrión como un valor grabado; establecer el temporizador del sistema operativo anfitrión al periodo de gracia; determinar si el sistema operativo de invitado se posibilita para la función de seguimiento de advertencias; y basándose en determinar que se posibilita el sistema operativo de invitado, notificar al sistema operativo de invitado que ha comenzado el periodo de gracia.

15. Un programa informático que comprende código de programa informático para provocar, cuando se carga en un sistema informático y se ejecuta en el mismo, que dicho sistema informático realice las etapas del método de acuerdo con cualquiera de las reivindicaciones 8 a 14.

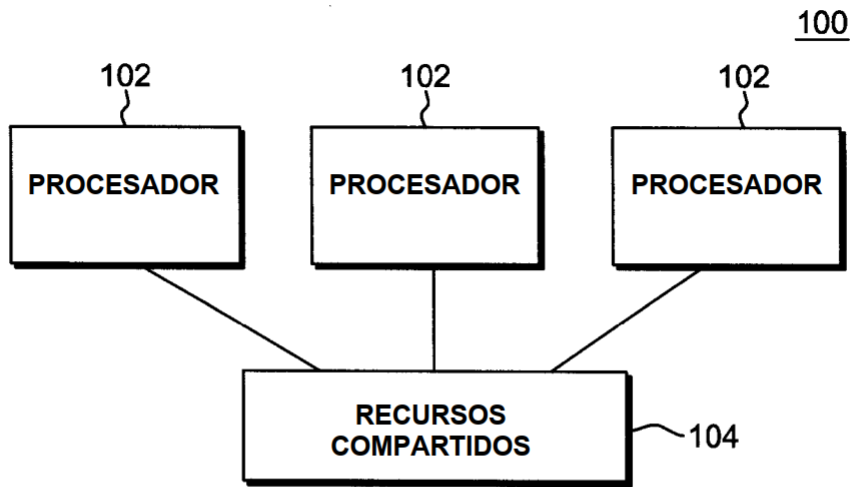


FIG. 1

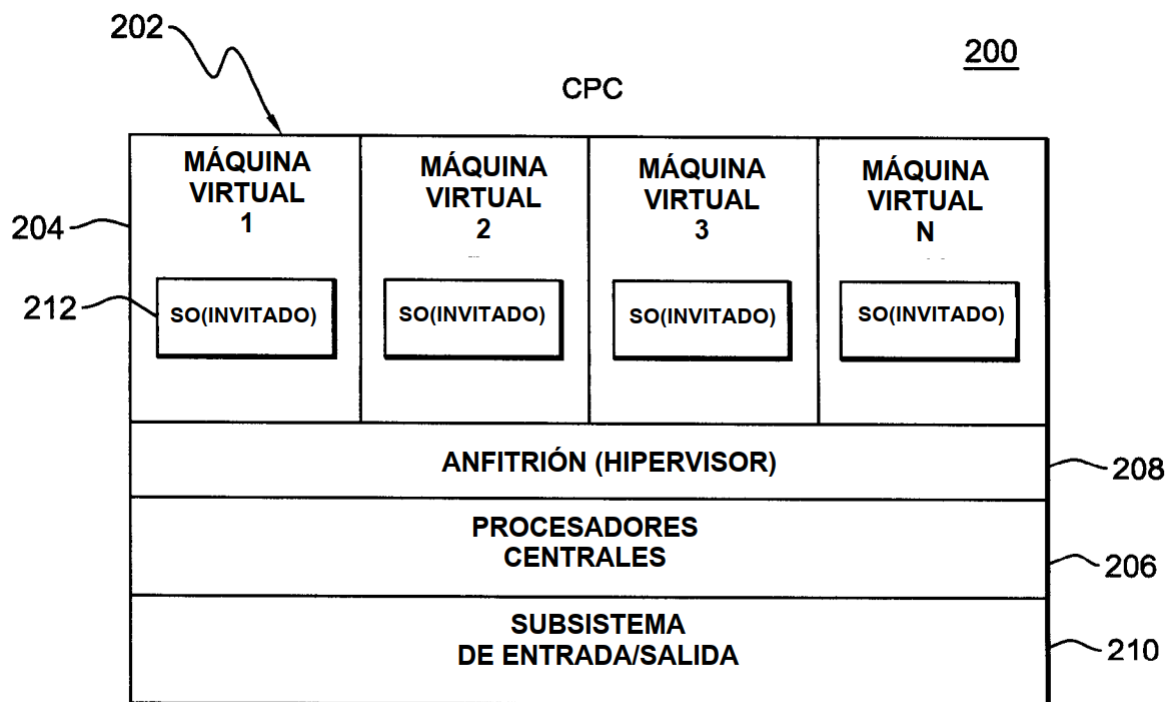


FIG. 2

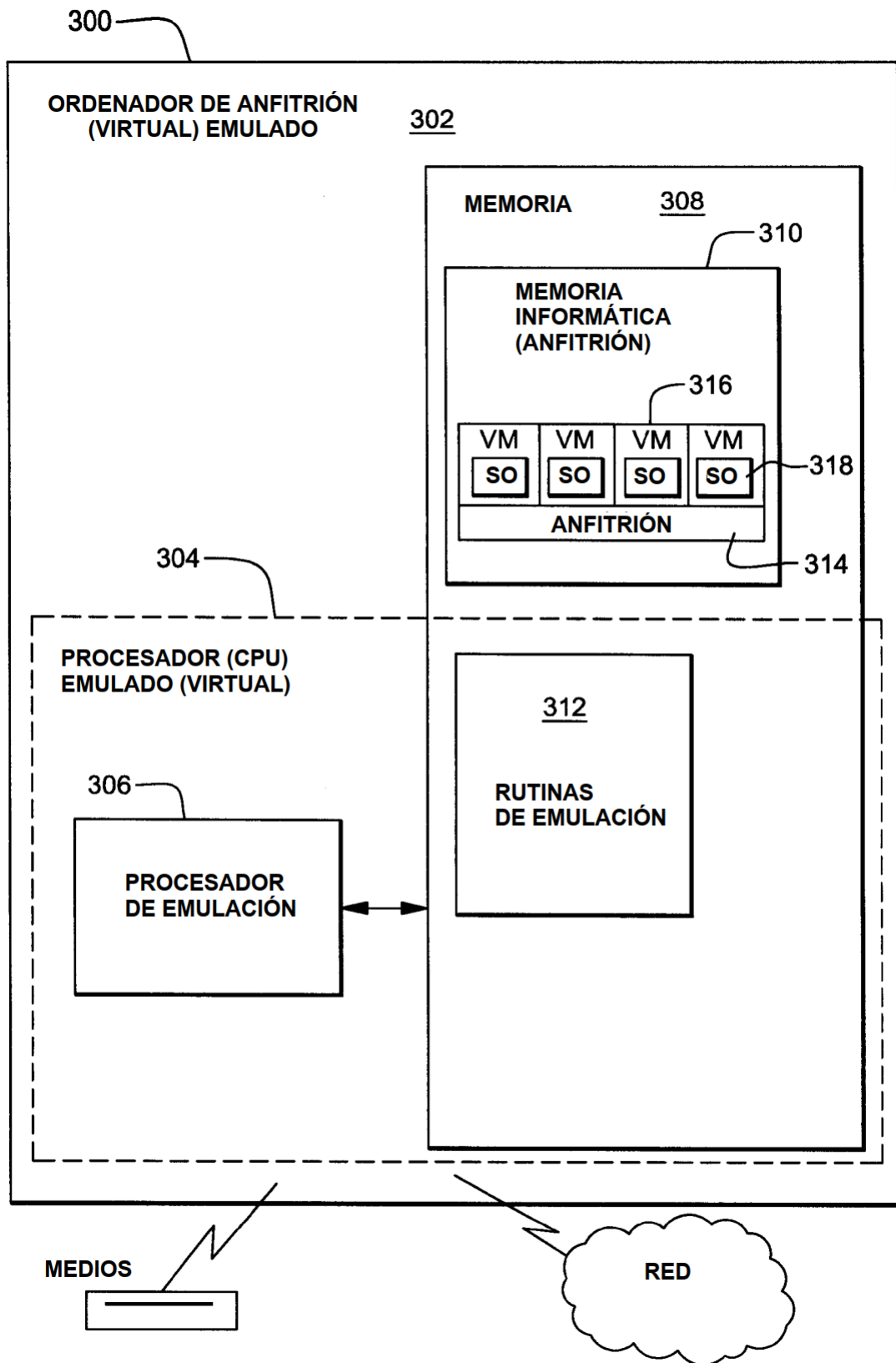


FIG. 3

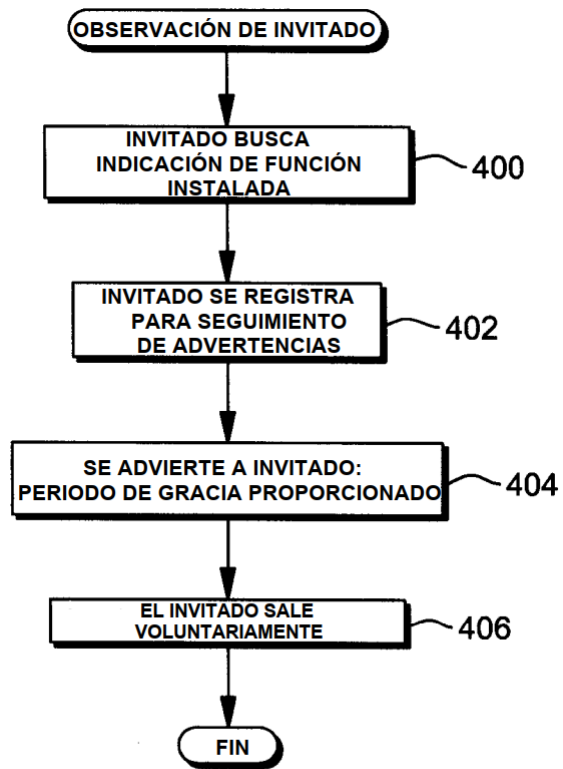


FIG. 4

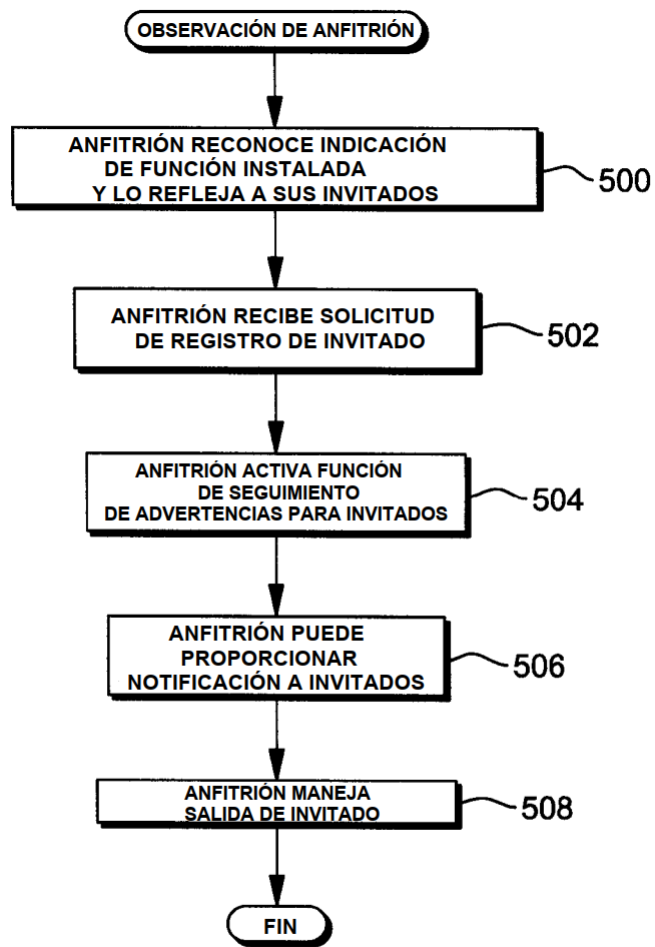


FIG. 5

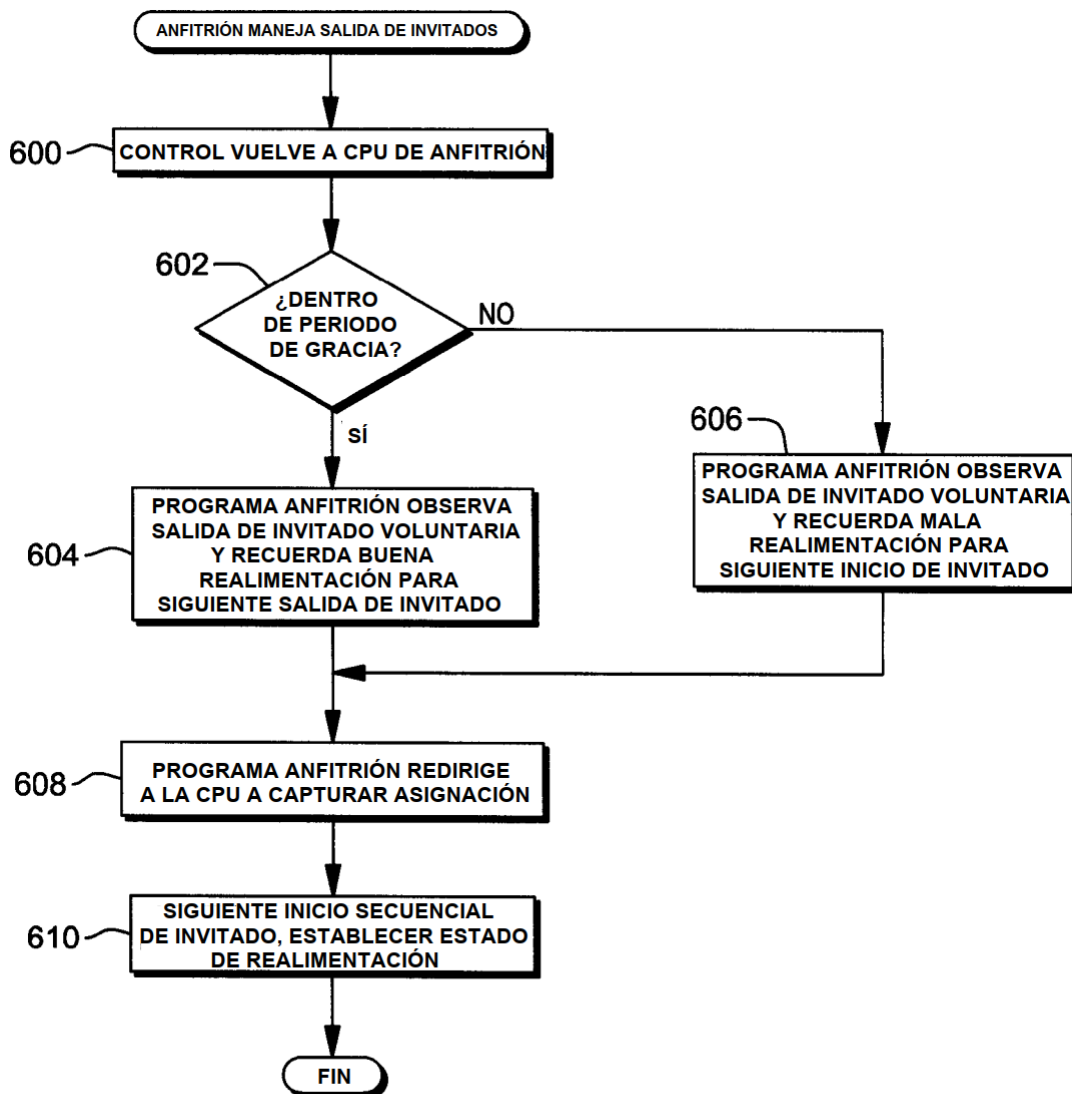


FIG. 6

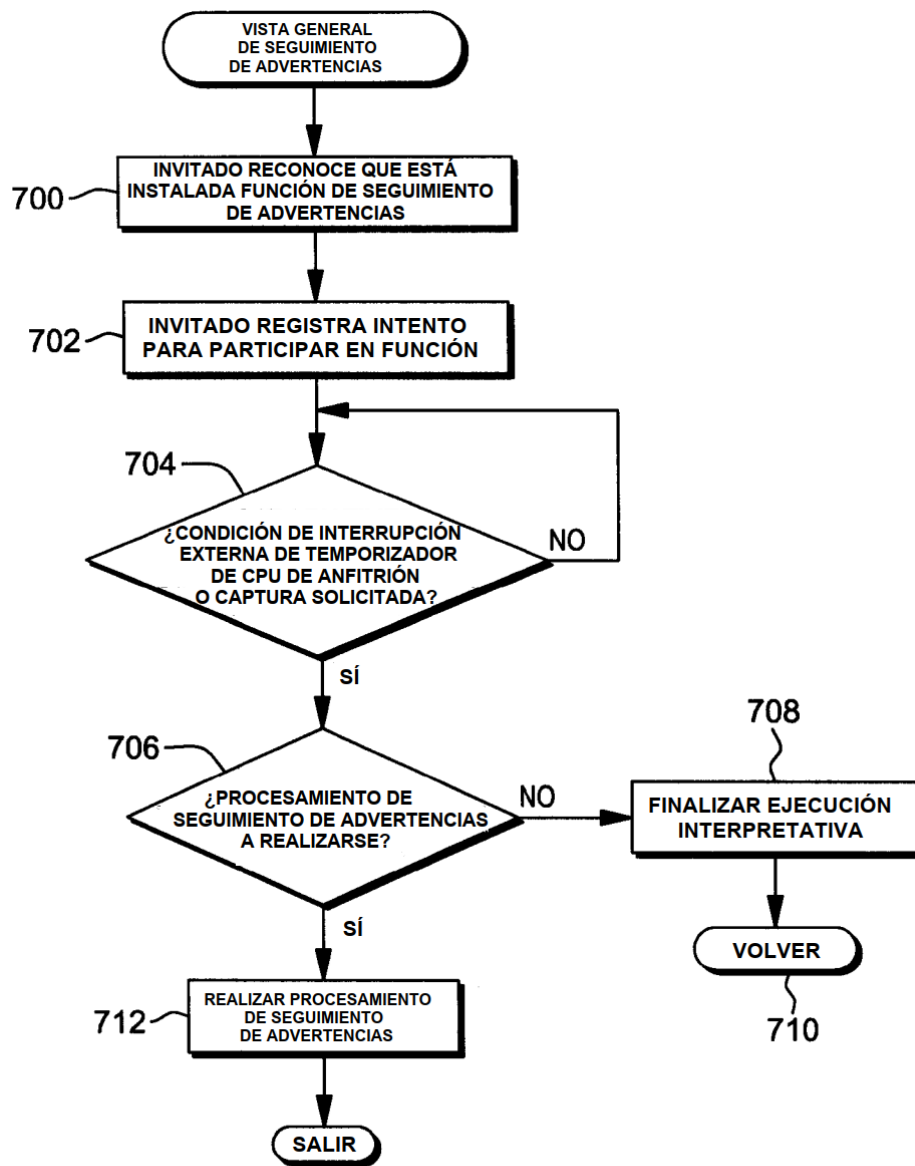
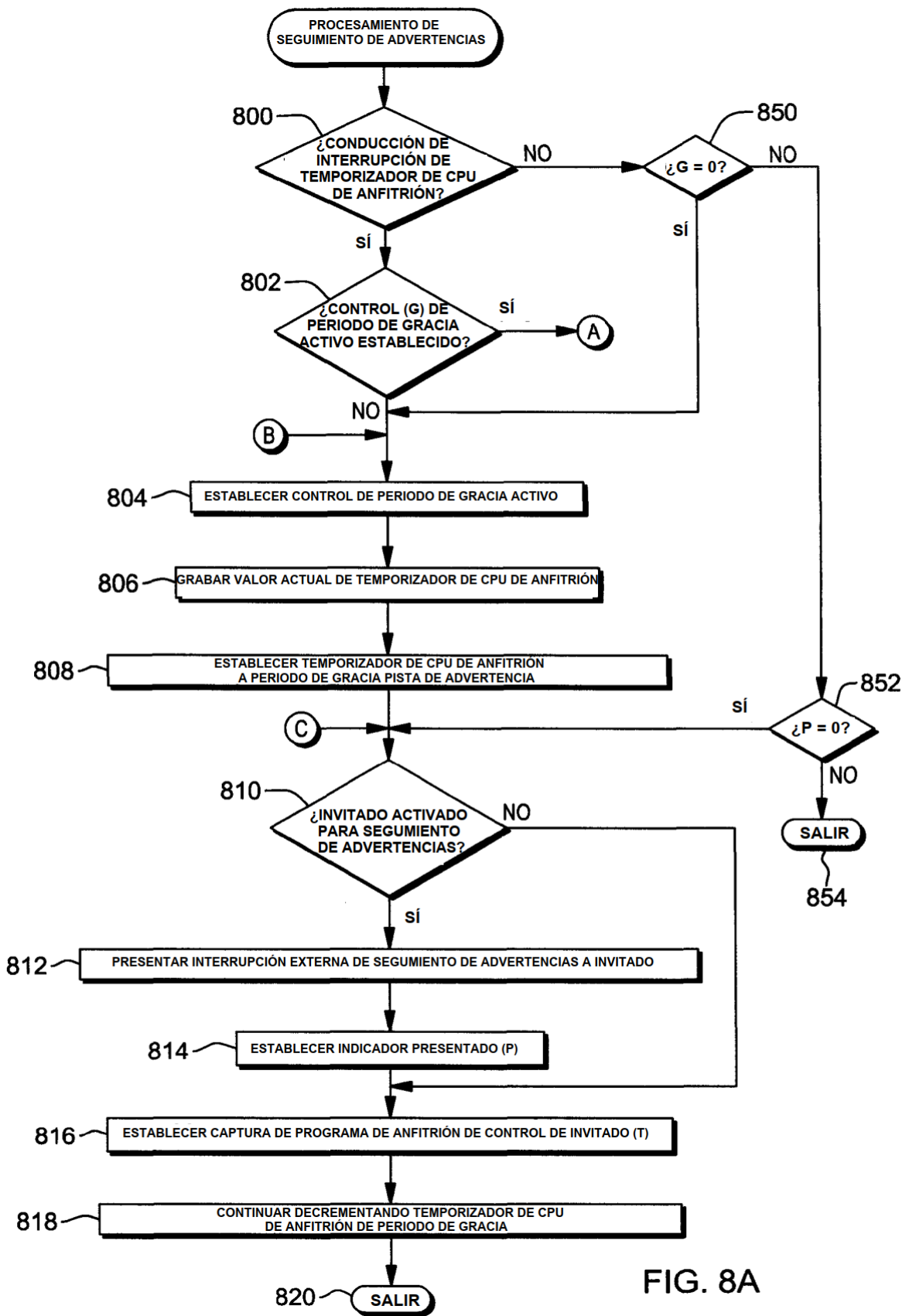


FIG. 7



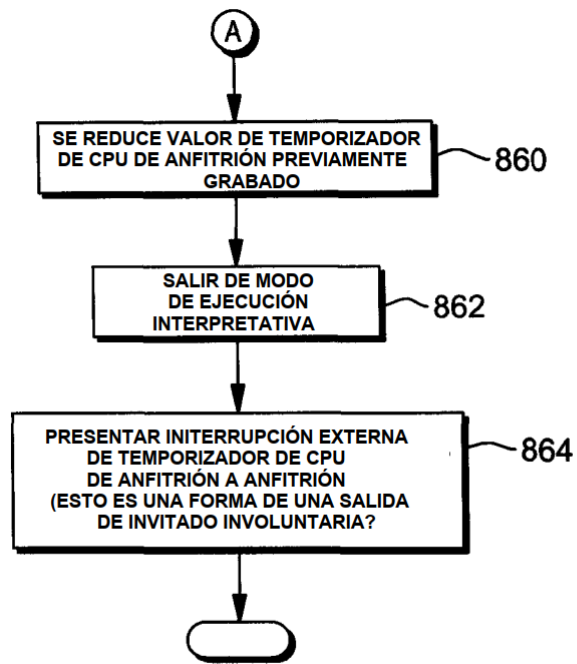


FIG. 8B

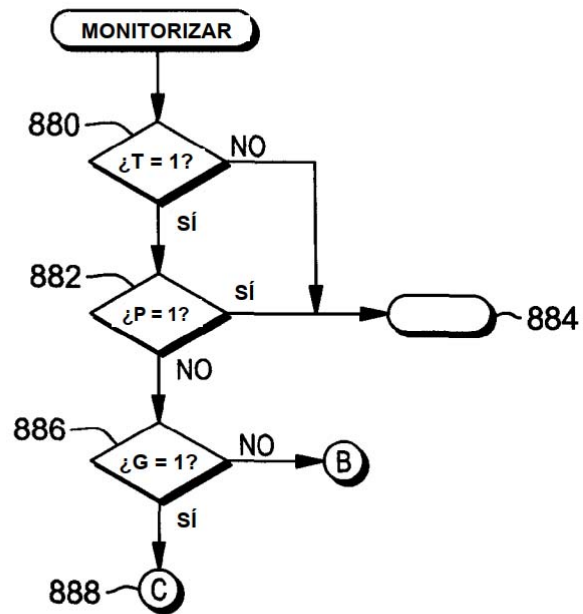


FIG. 8C

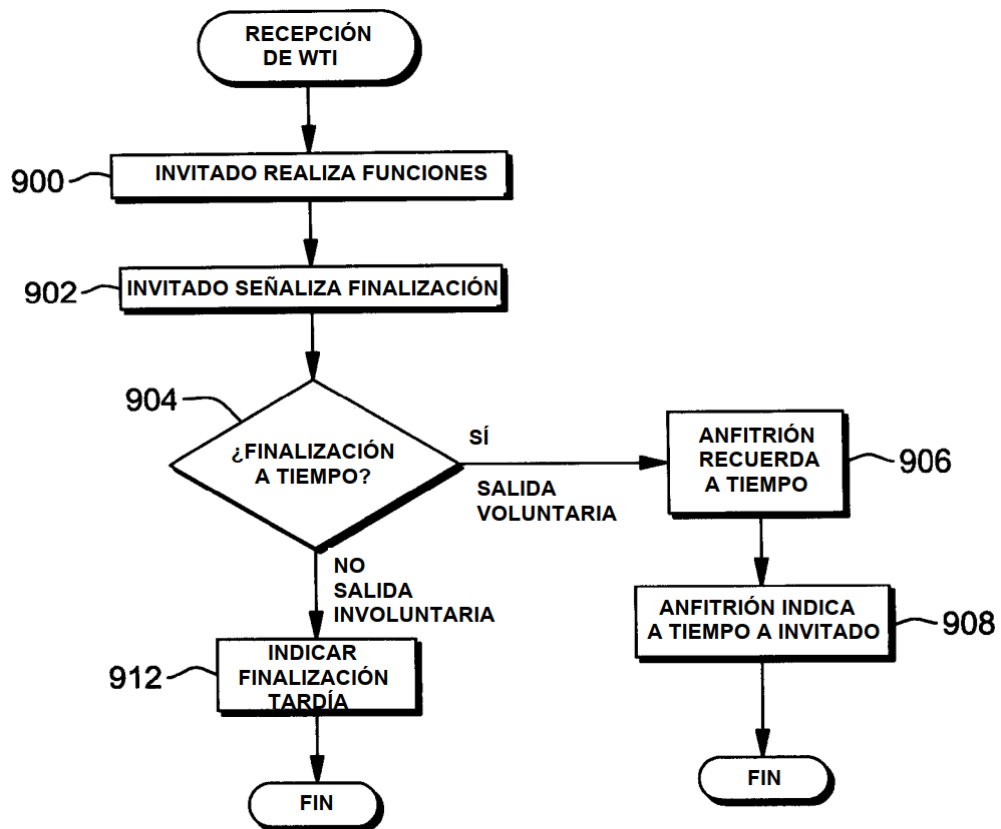


FIG. 9

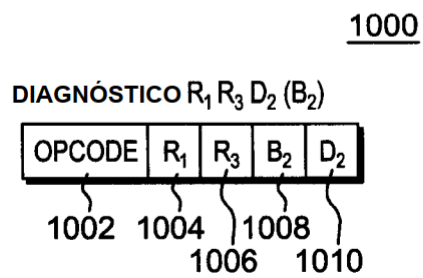


FIG. 10

**PRODUCTO
DE PROGRAMA
INFORMÁTICO**
1100



FIG. 11

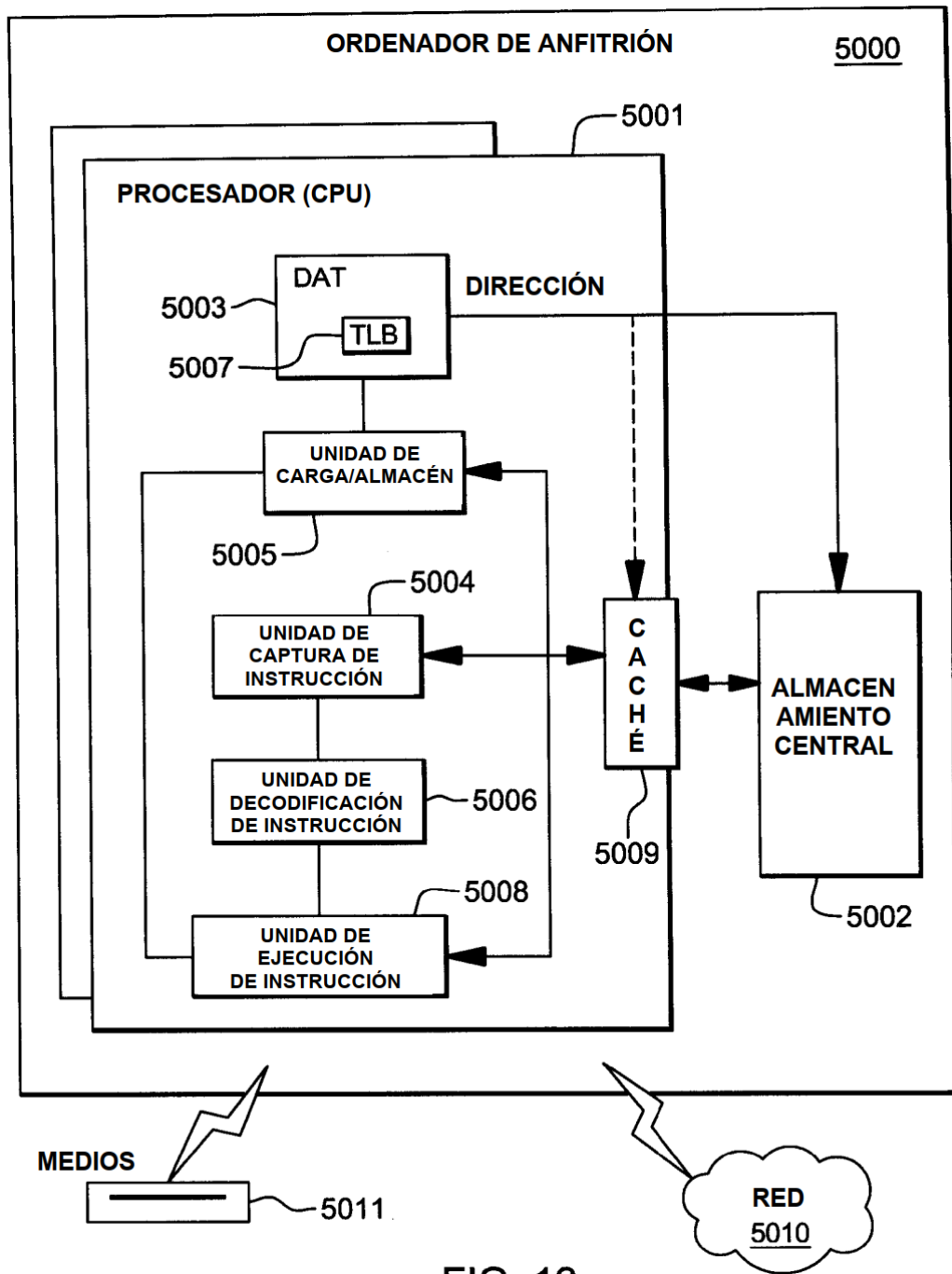


FIG. 12

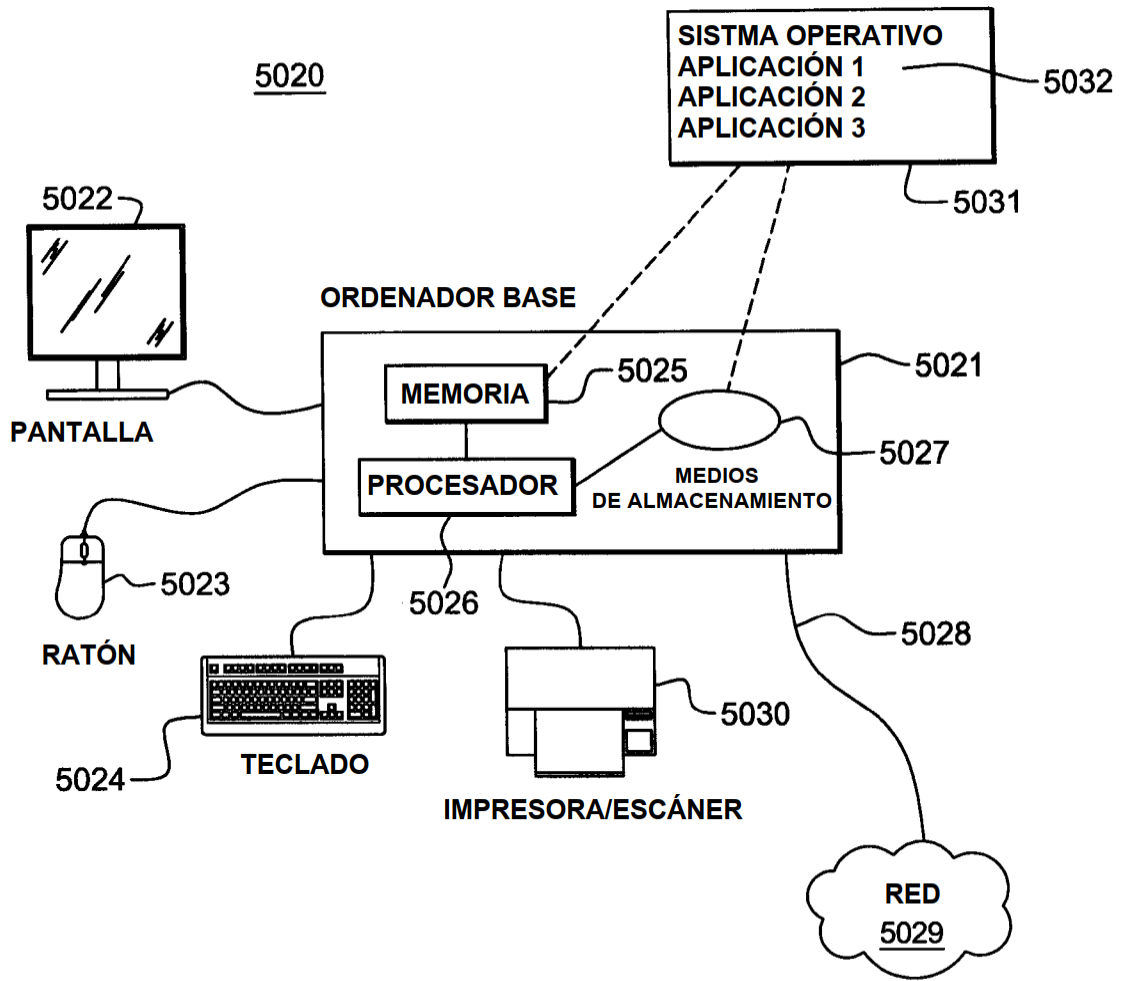


FIG. 13

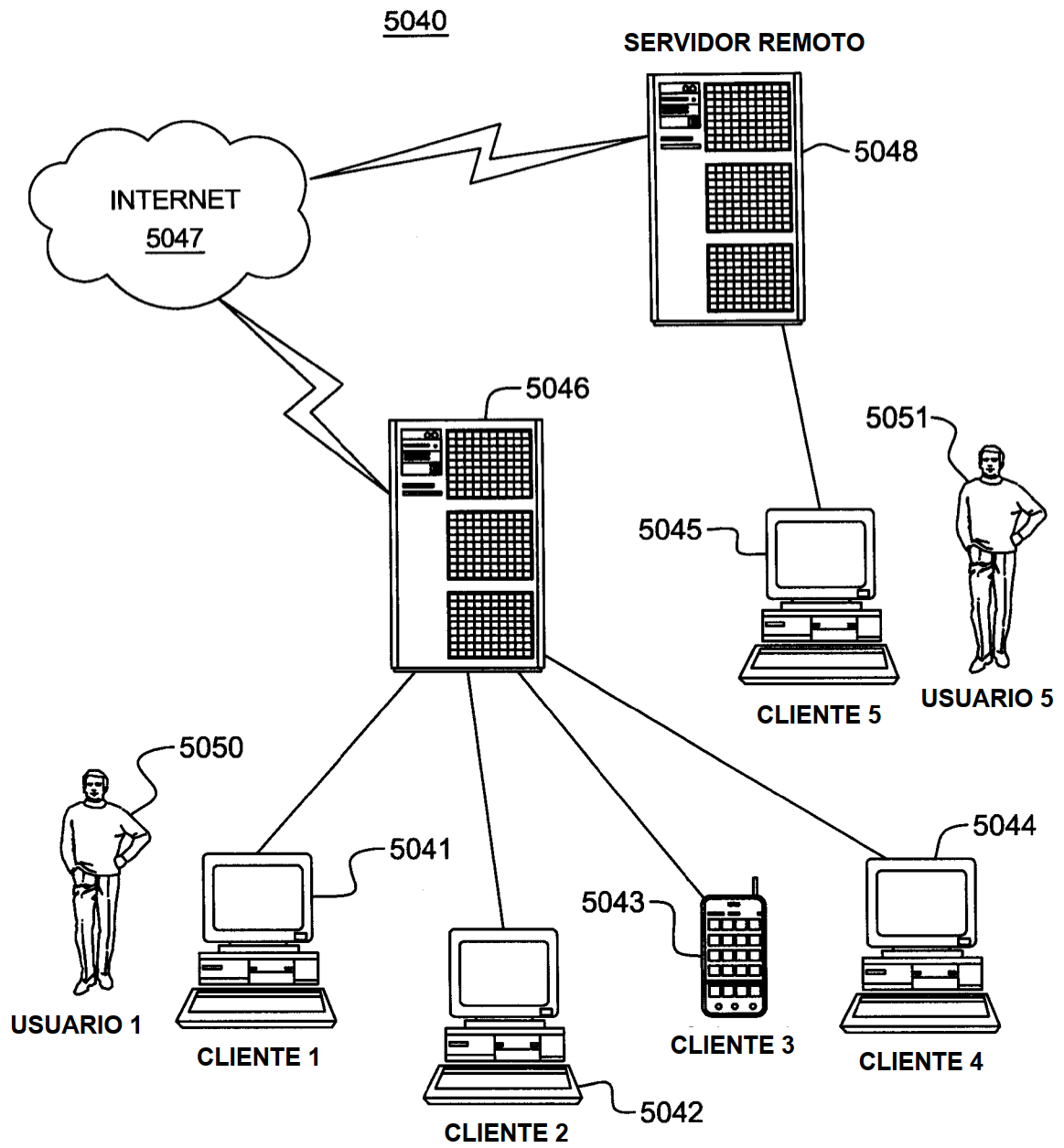


FIG. 14

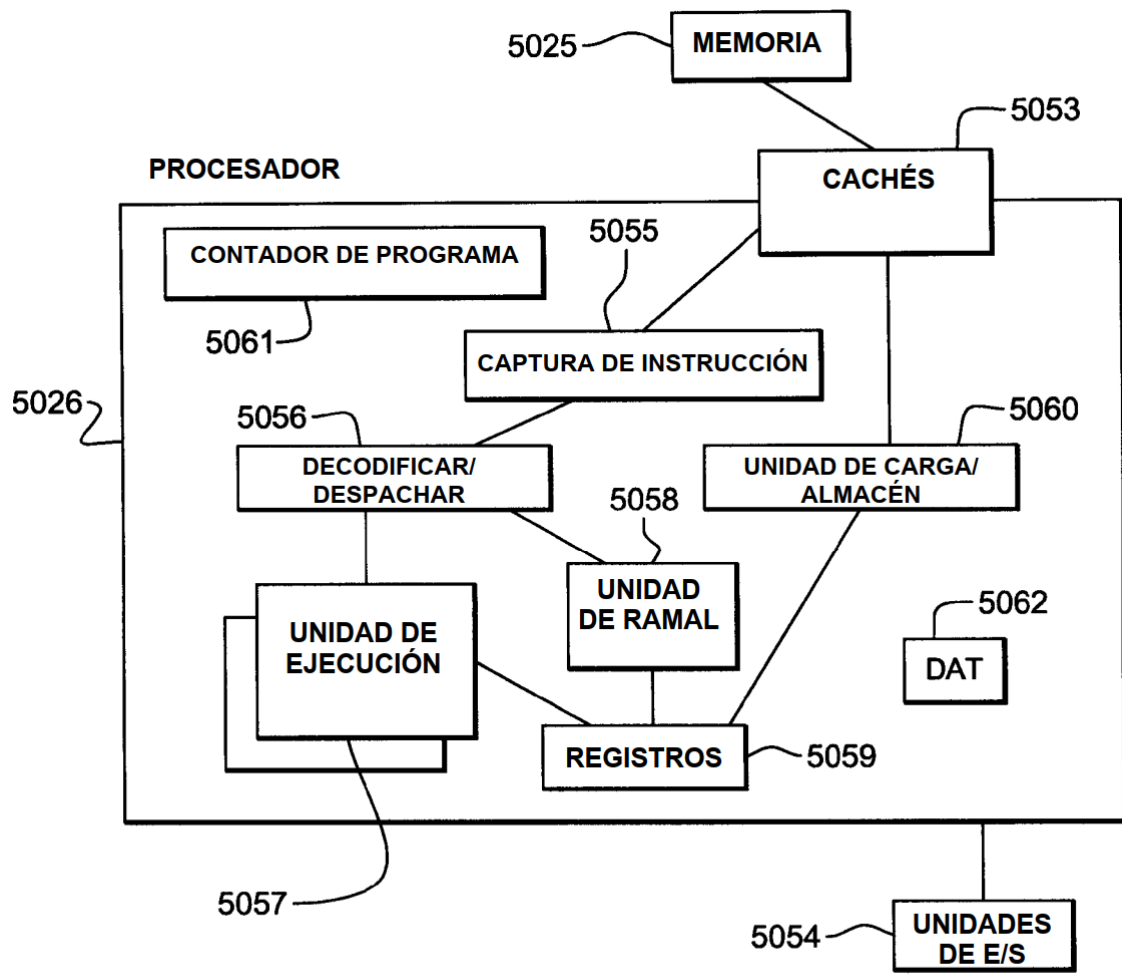


FIG. 15

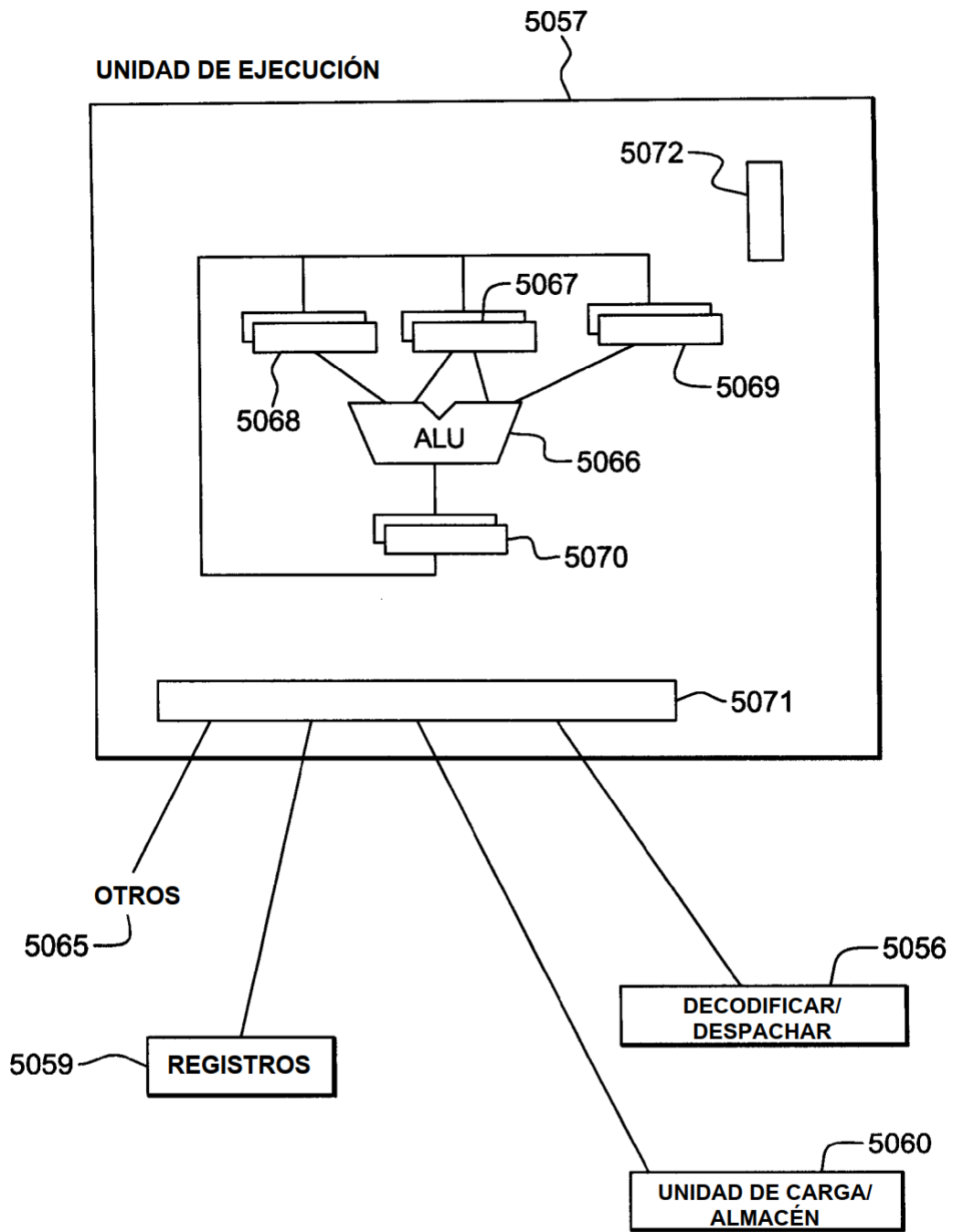


FIG. 16A

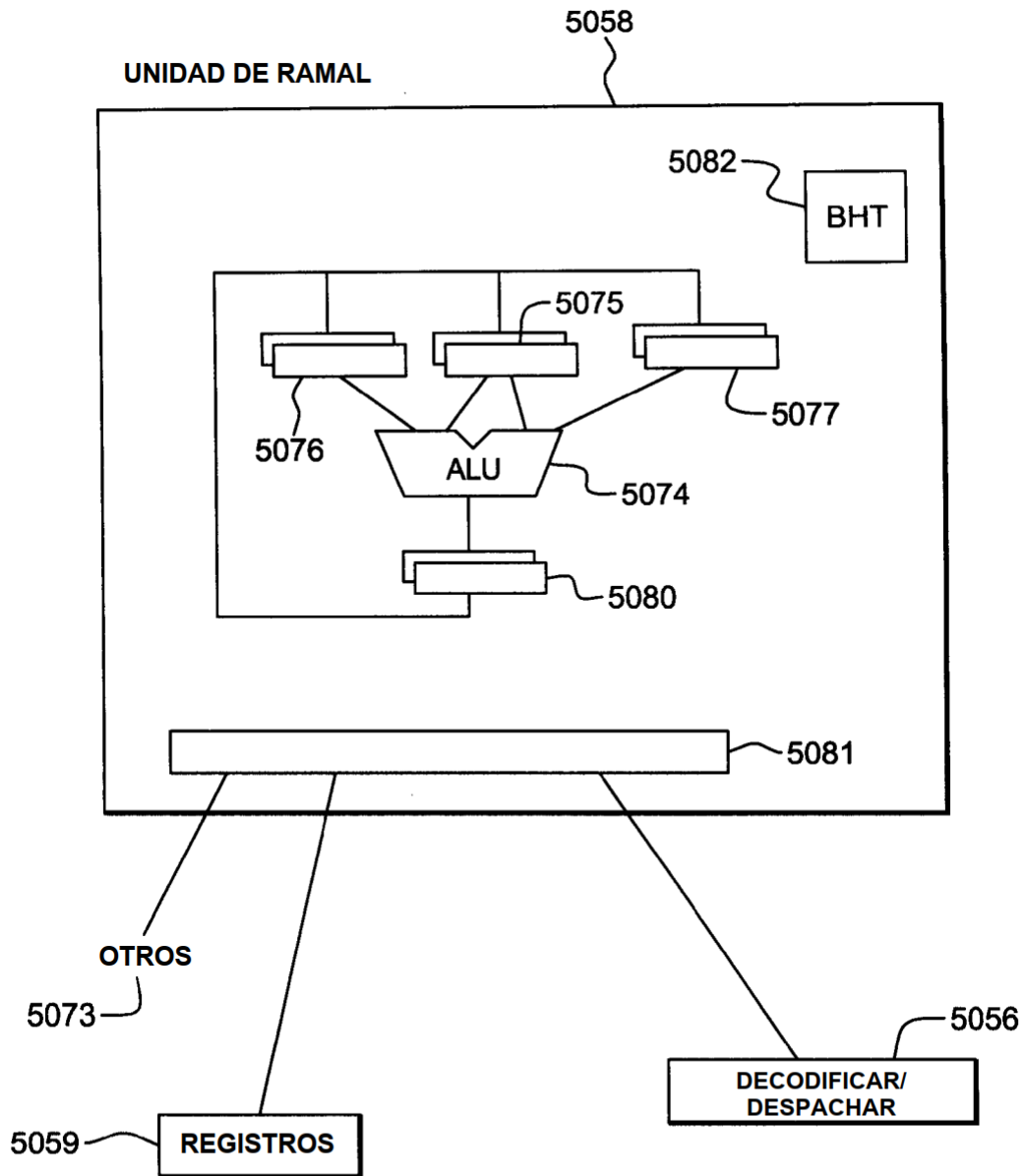


FIG. 16B

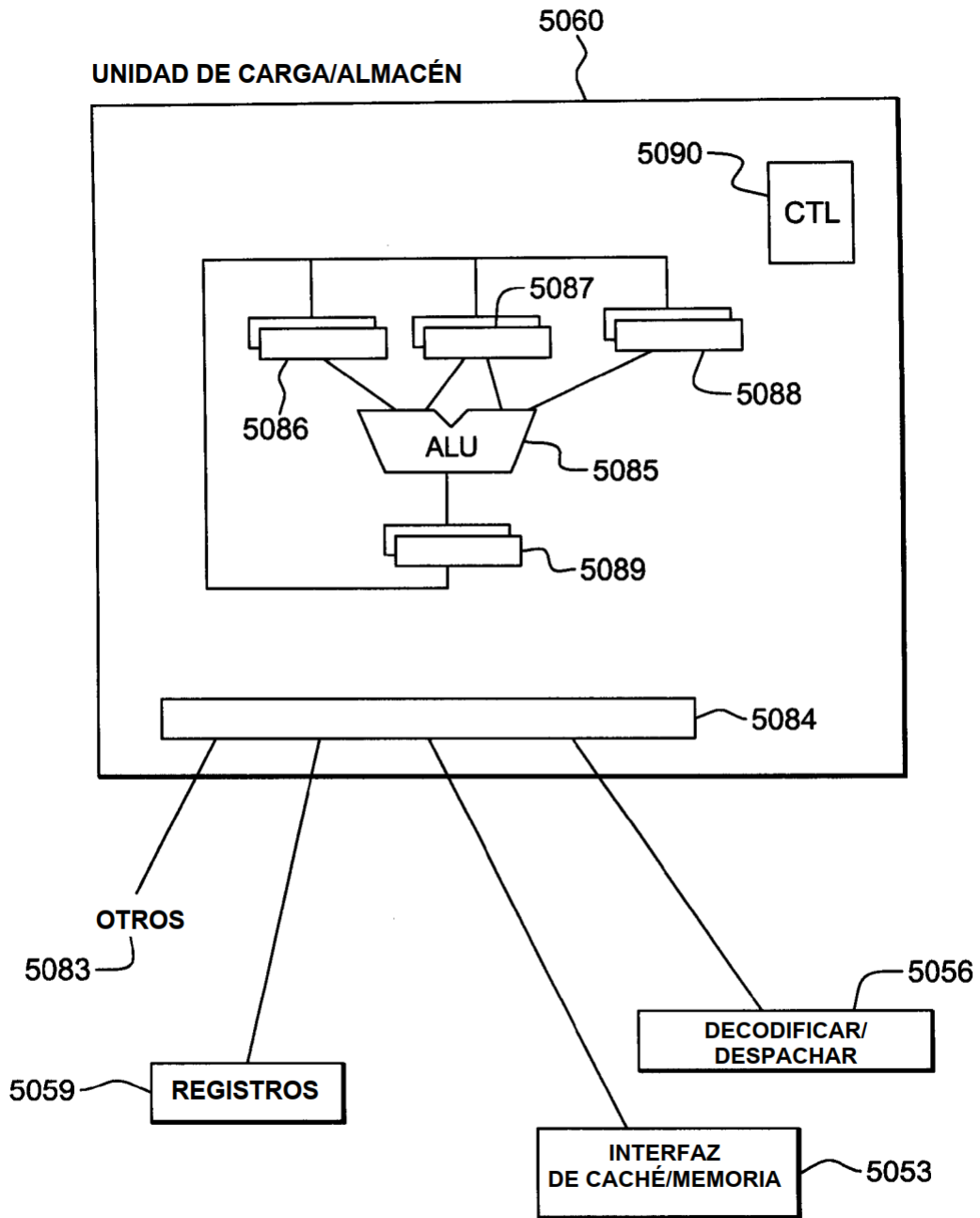


FIG. 16C

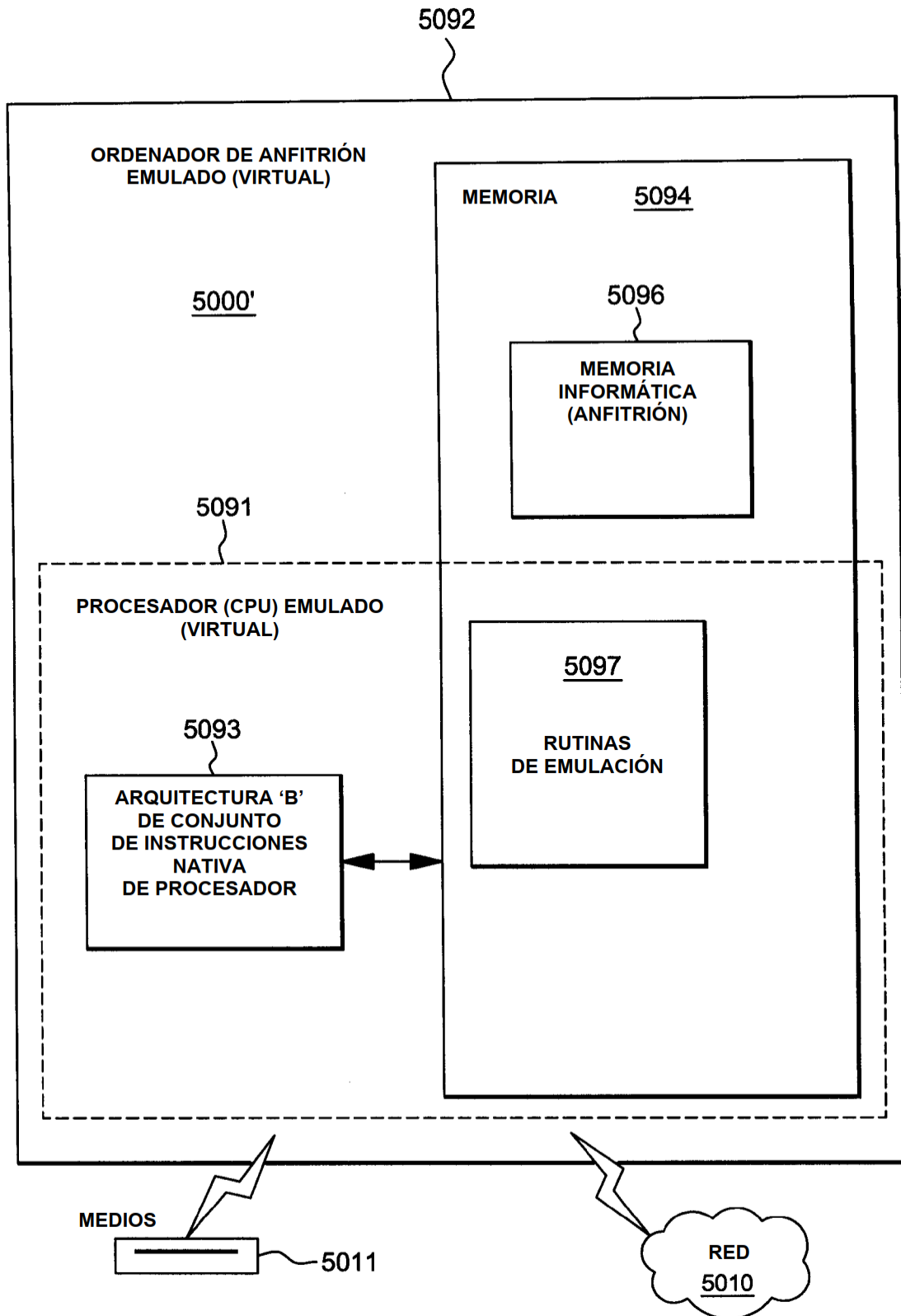


FIG. 17