

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 717 525**

51 Int. Cl.:

G06F 9/4401 (2008.01)

G06F 15/76 (2006.01)

G06F 15/78 (2006.01)

G06F 9/445 (2008.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **09.03.2015 PCT/EP2015/054850**

87 Fecha y número de publicación internacional: **24.09.2015 WO15139992**

96 Fecha de presentación y número de la solicitud europea: **09.03.2015 E 15711073 (5)**

97 Fecha y número de publicación de la concesión europea: **27.02.2019 EP 3114561**

54 Título: **Configuración de modo arquitectónico en un sistema informático**

30 Prioridad:

18.03.2014 US 201414217840
26.11.2014 US 201414554806

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
21.06.2019

73 Titular/es:

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

72 Inventor/es:

GSCHWIND, MICHAEL KARL y
GAINEY, CHARLES

74 Agente/Representante:

ELZABURU, S.L.P

ES 2 717 525 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

Configuración de modo arquitectónico en un sistema informático

Campo técnico

5 Uno o más aspectos se refieren, en general, a configuraciones de entornos informáticos y, en particular, a alterar las configuraciones de tales entornos.

Antecedentes de la técnica

Los entornos informáticos ofrecen una gama de capacidades y funciones que dependen de las configuraciones arquitectónicas de los entornos. Dos arquitecturas que han sido ofrecidas por International Business Machines Corporation, Armonk, Nueva York, incluyen ESA/390 y z/Architecture.

10 ESA/390 es una arquitectura predecesora de z/Architecture. No obstante, cuando se introdujo z/Architecture, ESA/390 continuó siendo soportada. Para soportar ambas arquitecturas en un entorno, se siguen ciertos procedimientos. Por ejemplo, en el encendido, se arranca ESA/390, y luego, se puede hacer una conmutación a z/Architecture, si se desea. Esto permitió que el software heredado continuase ejecutándose sin cambios. Se proporcionan otros de tales procedimientos con el fin de soportar ambas configuraciones arquitectónicas en un
15 entorno.

No obstante, las pruebas de memoria virtual son caras. A medida que la arquitectura se pone en suspenso, puede ser deseable proporcionar entornos heredados, por ejemplo, para sistemas que usan un soporte arquitectónico mínimo, tales como los sistemas operativos DOS (por ejemplo, tales como MS DOS o CMS), que funcionan principalmente como entornos de intérprete de líneas de comandos, o para entornos que se usan para ejecutar parte del BIOS (y que se pueden ejecutar sin las complejidades de memoria virtual). La Publicación de Solicitud de Patente de Estados Unidos US 2012/260064 A1 (Glenn, H.G. et al. "Heterogeneous ISA Microprocessor with Shared Hardware ISA Registers", 11 de octubre de 2012) describe un microprocesador capaz de ejecutar tanto programas de lenguaje máquina de Arquitectura de Conjunto de Instrucciones (ISA) x86 como programas de lenguaje máquina ISA de Máquinas RISC Avanzada (ARM) incluye un indicador de modo que indica si el microprocesador está obteniendo actualmente instrucciones de un programa de lenguaje máquina ISA o ISA de ARM x86 y una pluralidad de registros de hardware. Cuando el indicador de modo indica que el microprocesador está obteniendo actualmente instrucciones de programa de lenguaje máquina ISA x86, la pluralidad de registros de hardware almacenan el estado arquitectónico ISA x86; cuando el indicador de modo indica que el microprocesador está obteniendo actualmente instrucciones de programa en lenguaje máquina de ISA de ARM, la pluralidad de registros de hardware almacenan el estado arquitectónico de ISA ARM. IBM eServer zSeries 900 Technical Guide, Injey, et al. 1 de septiembre de 2002 (IBM Redbooks) proporciona una guía técnica para el IBM eServer zSeries 900.
20
25
30

Por lo tanto, hay una necesidad en la técnica de abordar el problema mencionado anteriormente.

Compendio

La invención se define por las reivindicaciones independientes.

35 Se superan las deficiencias de la técnica anterior y se proporcionan ventajas a través de la provisión de un producto de programa de ordenador para reconfigurar un entorno informático. El producto de programa de ordenador incluye, por ejemplo, un medio de almacenamiento legible por ordenador legible por un circuito de procesamiento y que almacena instrucciones para su ejecución por el circuito de procesamiento para realizar un método. El método incluye, por ejemplo, determinar, mediante un procesador, que una facilidad de modo arquitectónico de configuración está instalada en un entorno informático que está configurado para una pluralidad de modos arquitectónicos y tiene una secuencia de encendido definida que es para encender el entorno informático en un modo arquitectónico de la pluralidad de modos arquitectónicos, el modo arquitectónico único que comprende una primera arquitectura de conjunto de instrucciones y que tiene un primer conjunto de características soportadas; en base a la determinación de que la facilidad de modo arquitectónico de configuración está instalada, reconfigurar, mediante el procesador, el entorno informático para restringir el uso del modo arquitectónico único, en donde la reconfiguración incluye: seleccionar una secuencia de encendido diferente para encender el entorno informático en otro modo arquitectónico de la pluralidad de modos arquitectónicos, en donde el otro modo arquitectónico es diferente del modo arquitectónico único, y el otro modo arquitectónico comprende una segunda arquitectura de conjunto de instrucciones y que tiene un segundo conjunto de características soportadas; y ejecutar la secuencia de encendido diferente para encender el entorno informático en el otro modo arquitectónico en lugar del modo arquitectónico único que restringe el uso del modo arquitectónico único.
40
45
50

Visto desde un primer aspecto, la presente invención proporciona un método para reconfigurar un entorno informático, dicho método que comprende: determinar, mediante un procesador, que un indicador de instalación de facilidad se establece en un valor predeterminado, el valor predeterminado que indica la instalación de una facilidad de modo arquitectónico de configuración en un entorno informático que está configurado para una pluralidad de modos arquitectónicos y tiene una secuencia de encendido definida que es para encender el entorno informático en
55

un modo arquitectónico de la pluralidad de modos arquitectónicos, el modo arquitectónico único que comprende una primera arquitectura de conjunto de instrucciones y que tiene un primer conjunto de características soportadas; en base a la determinación de que el indicador de instalación de facilidad se establece en el valor predeterminado, reconfigurar, mediante el procesador, el entorno informático para restringir el uso del modo arquitectónico único, de manera que ya no se soportan más los aspectos del modo arquitectónico único, la capacidad de usar esos aspectos del modo arquitectónico único se elimina del entorno informático y se evita que el entorno informático se reconfigure de vuelta al modo arquitectónico único no soportado, en donde la reconfiguración es sensible a determinar que el indicador de instalación de facilidad se establece en el valor predeterminado y no en respuesta a una solicitud explícita para reconfigurar y en donde la reconfiguración comprende: seleccionar una secuencia de encendido diferente para encender el entorno informático en otro modo arquitectónico de la pluralidad de modos arquitectónicos, en donde el otro modo arquitectónico es diferente del modo arquitectónico único, y el otro modo arquitectónico comprende una segunda arquitectura de conjunto de instrucciones y que tiene un segundo conjunto de características soportadas; y ejecutar la secuencia de encendido diferente para encender el entorno informático en el otro modo arquitectónico en lugar del modo arquitectónico único que restringe el uso del modo arquitectónico único, en donde la ejecución de la secuencia de encendido diferente comprende crear una nueva palabra de estado de programa para controlar las operaciones del entorno informático en el otro modo arquitectónico, la creación de la nueva palabra de estado de programa que comprende formar la nueva palabra de estado de programa para tener un formato indicado por el otro modo arquitectónico, el formato que comprende la expansión de un campo de dirección de un primer tamaño a un segundo tamaño, e invertir, durante la formación de la nueva palabra de estado de programa, un indicador de modo arquitectónico en la nueva palabra de estado de programa para indicar el otro modo arquitectónico.

Visto desde un aspecto adicional, la presente invención proporciona un sistema informático para reconfigurar un entorno informático, dicho sistema informático que comprende: una memoria; y un procesador en comunicaciones con la memoria, en donde el sistema informático está configurado para realizar un método, dicho método que comprende determinar, mediante un procesador, que un indicador de instalación de facilidad se establece en un valor predeterminado, el valor predeterminado que indica la instalación de una facilidad de modo arquitectónico de configuración en un entorno informático que está configurado para una pluralidad de modos arquitectónicos y tiene una secuencia de encendido definida que es para encender el entorno informático en un modo arquitectónico único de la pluralidad de modos arquitectónicos, el modo arquitectónico único que comprende una primera arquitectura de conjunto de instrucciones y que tiene un primer conjunto de características soportadas; en base a determinar que el indicador de instalación de facilidad se establece al valor predeterminado, reconfigurar, mediante el procesador, el entorno informático para restringir el uso del modo arquitectónico único de manera que ya no se soporten más los aspectos del modo arquitectónico único, la capacidad de usar esos aspectos del modo arquitectónico único se elimina del entorno informático y se evita que el entorno informático se reconfigure de vuelta al modo arquitectónico único no soportado, en donde la reconfiguración es sensible a determinar que el indicador de instalación de facilidad se establece en el valor predeterminado y no en respuesta a una solicitud explícita para reconfigurar y en donde la reconfiguración comprende: seleccionar una secuencia de encendido diferente para encender el entorno informático en otro modo arquitectónico de la pluralidad de modos arquitectónicos, en donde el otro modo arquitectónico es diferente del modo arquitectónico único, y el otro modo arquitectónico comprende una segunda arquitectura de conjunto de instrucciones y que tiene un segundo conjunto de características soportadas; y ejecutar la secuencia de encendido diferente para encender el entorno informático en el otro modo arquitectónico en lugar del modo arquitectónico único que restringe el uso del modo arquitectónico único, en donde la ejecución de la secuencia de encendido diferente comprende crear una nueva palabra de estado de programa para controlar las operaciones del entorno informático en el otro modo arquitectónico, la creación de la nueva palabra de estado de programa que comprende formar la nueva palabra de estado de programa para tener un formato indicado por el otro modo arquitectónico, el formato que comprende expandir un campo de dirección de un primer tamaño a un segundo tamaño, e invertir, durante la formación de la nueva palabra de estado de programa, un indicador de modo arquitectónico en la nueva palabra de estado de programa para indicar el otro modo arquitectónico.

Visto desde un aspecto adicional, la presente invención proporciona un producto de programa de ordenador para reconfigurar un entorno informático, el producto de programa de ordenador que comprende un medio de almacenamiento legible por ordenador legible por un circuito de procesamiento y que almacena instrucciones para su ejecución por el circuito de procesamiento para realizar un método para realizar los pasos de la invención. Visto desde un aspecto adicional, la presente invención proporciona un producto de programa de ordenador para configurar un entorno informático, el producto de programa de ordenador que comprende un medio de almacenamiento legible por ordenador legible por un circuito de procesamiento y que almacena instrucciones para su ejecución por el circuito de procesamiento para realizar un método para realizar los pasos de la invención.

Visto desde un aspecto adicional, la presente invención proporciona un programa de ordenador almacenado en un medio legible por ordenador y que se puede cargar en la memoria interna de un ordenador digital, que comprende partes de código software, cuando dicho programa se ejecuta en un ordenador, para realizar los pasos de la invención.

Los métodos y sistemas relacionados con una o más realizaciones también se describen y reivindican en la presente memoria. Además, servicios relacionados con una o más realizaciones también se describen y se pueden reivindicar en la presente memoria.

Se realizan características y ventajas adicionales. Otras realizaciones y aspectos se describen en detalle en la presente memoria y se consideran una parte de la invención reivindicada.

Breve descripción de los dibujos

5 La presente invención se describirá ahora, a modo de ejemplo solamente, con referencia a realizaciones preferidas, como se ilustra en las siguientes figuras:

la FIG. 1A representa un ejemplo de un entorno informático para incorporar y usar uno o más aspectos de una facilidad de modo arquitectónico de configuración, según una realización preferida de la presente invención;

la FIG. 1B representa un ejemplo de un entorno informático virtual para incorporar y usar uno o más aspectos de una facilidad de modo arquitectónico de configuración, según una realización preferida de la presente invención;

10 la FIG. 2 representa otro ejemplo de un entorno informático para incorporar y usar uno o más aspectos de una facilidad de modo arquitectónico de configuración, según una realización preferida de la presente invención;

la FIG. 3A representa otro ejemplo más de un entorno informático para incorporar y usar uno o más aspectos de una facilidad de modo arquitectónico de configuración, según una realización preferida de la presente invención;

15 la FIG. 3B representa detalles adicionales de la memoria de la FIG. 3A, según una realización preferida de la presente invención;

la FIG. 4A representa una realización de la lógica para encender un entorno informático en un modo arquitectónico único, según una realización preferida de la presente invención;

la FIG. 4B representa una realización de un procesamiento adicional asociado con el proceso de encendido de la FIG. 4A, según una realización preferida de la presente invención;

20 la FIG. 5 representa una realización de un formato de una palabra de estado de programa, según una realización preferida de la presente invención;

la FIG. 6A representa una realización de la lógica para encender un entorno informático en un modo arquitectónico diferente del modo arquitectónico único encendido en la FIG. 4A, según una realización preferida de la presente invención;

25 la FIG. 6B representa una realización de un procesamiento adicional asociado con el proceso de encendido de la FIG. 6A, según una realización preferida de la presente invención;

la FIG. 7 representa un ejemplo de un formato de una instrucción Cargar Palabra de Estado de Programa, según una realización preferida de la presente invención;

30 la FIG. 8A representa un ejemplo de un formato de una instrucción Señalar Procesador, según una realización preferida de la presente invención;

la FIG. 8B representa una realización del procesamiento asociado con la instrucción Señalar Procesador de la FIG. 8A, según una realización preferida de la presente invención;

la FIG. 9 representa una realización de la lógica para encender un entorno informático en una configuración reconfigurada, según una realización preferida de la presente invención;

35 la FIG. 10 representa cambios adicionales a ser hechos en la reconfiguración de un entorno informático, según una realización preferida de la presente invención;

la FIG. 11 representa una realización de la lógica para restablecer un entorno informático;

la FIG. 12 representa una realización de una lógica para configurar un entorno informático, según una realización preferida de la presente invención;

40 la FIG. 13 representa una realización de un producto de programa de ordenador, según la técnica anterior, y en la que se puede implementar una realización preferida de la presente invención;

la FIG. 14 representa una realización de un sistema informático central, según la técnica anterior, y en la que se puede implementar una realización preferida de la presente invención;

45 la FIG. 15 representa un ejemplo adicional de un sistema informático, según la técnica anterior, y en el que se puede implementar una realización preferida de la presente invención;

la FIG. 16 representa otro ejemplo de un sistema informático que comprende una red informática, según la técnica anterior, y en el que se puede implementar una realización preferida de la presente invención;

la FIG. 17 representa una realización de varios elementos de un sistema informático, según la técnica anterior, y en la que se puede implementar una realización preferida de la presente invención;

la FIG. 18A representa una realización de la unidad de ejecución del sistema informático de la FIG. 17, según la técnica anterior, y en la que se puede implementar una realización preferida de la presente invención;

5 la FIG. 18B representa una realización de la unidad de ramificación del sistema informático de la FIG. 17, según la técnica anterior, y en la que se puede implementar una realización preferida de la presente invención;

la FIG. 18C representa una realización de la unidad de carga/almacenamiento del sistema informático de la FIG. 17, según la técnica anterior, y en la que se puede implementar una realización preferida de la presente invención;

10 la FIG. 19 representa una realización de un sistema informático central emulado, según la técnica anterior, y en el que se puede implementar una realización preferida de la presente invención;

la FIG. 20 representa una realización de un nodo informático en la nube, según la técnica anterior, y en la que se puede implementar una realización preferida de la presente invención;

15 la FIG. 21 representa una realización de un entorno informático en la nube, según la técnica anterior, y en la que se puede implementar una realización preferida de la presente invención; y

la FIG. 22 representa un ejemplo de capas de modelo de abstracción, según la técnica anterior, y en el que se puede implementar una realización preferida de la presente invención.

Descripción detallada

20 Según un aspecto, se proporciona una capacidad que restringe el uso de una configuración por un entorno informático configurado para soportar múltiples configuraciones, de manera que uno o más aspectos de la configuración restringida no están disponibles para su uso. Como ejemplo, un procesador se configura en un modo arquitectónico de configuración (CAM). En CAM, un entorno informático (por ejemplo, un procesador, una partición lógica, un invitado), que se configura originalmente para una pluralidad de arquitecturas, por ejemplo, una arquitectura heredada y una arquitectura mejorada, se reconfigura de manera que uno o más aspectos de al menos una de las arquitecturas, tal como la arquitectura heredada, ya no se soportan más. En tal configuración, los aspectos no soportados de la arquitectura no están disponibles.

30 Como ejemplo particular, se proporciona una facilidad de Modo Arquitectónico de Configuración de z/Architecture (CZAM) en entornos informáticos que soportan múltiples arquitecturas, tales como ESA/390 y z/Architecture, que elimina la capacidad de usar aspectos de ESA/390. En su lugar, se usa z/Architecture (y/u otras arquitecturas, en otras realizaciones distintas de ESA/390). CZAM puede aplicarse a una máquina nativa, una partición lógica y/o un invitado virtual, como ejemplos.

35 Un ejemplo de un entorno informático para incorporar y usar uno o más aspectos de la facilidad de modo arquitectónico de configuración se describe con referencia a la FIG. 1A. Con referencia a la FIG. 1A, en un ejemplo, un entorno informático 100 se basa en la z/Architecture, ofrecida por International Business Machines (IBM®) Corporation, Armonk, Nueva York. La z/Architecture se describe en una publicación de IBM titulada "z/Architecture - Principles of Operation", N° de Publicación SA22-7932-09, 10ª Edición, septiembre de 2012. Aunque el entorno informático se basa en la z/Architecture, en una realización preferida de la presente invención, también soporta una o más de otras configuraciones arquitectónicas, tales como ESA/390.

40 Como ejemplo, el entorno informático 100 incluye un complejo de procesador de ordenador central (CPC) 102 acoplado a uno o más dispositivos de entrada/salida (I/O) 106 a través de una o más unidades de control 108. El complejo de procesador de ordenador central 102 incluye, por ejemplo, una memoria de procesador 104 (también conocida como memoria principal, almacenamiento principal, almacenamiento central) acoplada a uno o más procesadores centrales (también conocidos como unidades centrales de procesamiento (CPU)) 110, y un subsistema de entrada/salida 111, cada uno de los cuales se describen a continuación.

45 La memoria de procesador 104 incluye, por ejemplo, una o más particiones 112 (por ejemplo, particiones lógicas), y el microprograma de procesador 113, que incluye un hipervisor de partición lógica 114 y otro microprograma de procesador 115. Un ejemplo del hipervisor de partición lógica 114 es el Recurso de Procesador/Administrador de Sistema (PR/SM), ofrecido por International Business Machines Corporation, Armonk, Nueva York.

50 Una partición lógica funciona como un sistema separado y tiene una o más aplicaciones 120 y, opcionalmente, un sistema operativo residente 122 dentro del mismo, que puede diferir para cada partición lógica. En una realización preferida de la presente invención, el sistema operativo es el sistema operativo z/OS, el sistema operativo z/VM, el sistema operativo z/Linux, o el sistema operativo TPF, ofrecido por International Business Machines Corporation, Armonk, Nueva York. Las particiones lógicas 112 se gestionan mediante el hipervisor de partición lógica 114, que se implementa mediante el microprograma que se ejecuta en los procesadores 110. Como se usa en la presente

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

Los procesadores centrales 110 son recursos de procesador físicos asignados a las particiones lógicas. En particular, cada partición lógica 112 tiene uno o más procesadores lógicos, cada uno de los cuales representa todo o una parte de un procesador físico 110 asignado a la partición. Los procesadores lógicos de una partición 112 particular pueden ser o bien dedicados a la partición, de modo que el recurso de procesador subyacente 110 se reserve para esa partición; o bien compartidos con otra partición, de modo que el recurso de procesador subyacente esté potencialmente disponible para otra partición. En un ejemplo, una o más de las CPU incluyen aspectos de una facilidad de modo arquitectónico de configuración 130 descrita en la presente memoria.

El subsistema de entrada/salida 111 dirige el flujo de información entre los dispositivos de entrada/salida 106 y el almacenamiento principal 104. Está acoplado al complejo de procesamiento central, en el sentido de que puede ser una parte del complejo de procesamiento central o estar separado del mismo. El subsistema de I/O alivia a los procesadores centrales de la tarea de comunicarse directamente con los dispositivos de entrada/salida y permite que el procesamiento de datos proceda concurrentemente con el procesamiento de entrada/salida. Para proporcionar comunicaciones, el subsistema de I/O emplea adaptadores de comunicaciones de I/O. Hay diversos tipos de adaptadores de comunicaciones que incluyen, por ejemplo, canales, adaptadores de I/O, tarjetas PCI, tarjetas Ethernet, tarjetas de Pequeña Interfaz de Almacenamiento de Ordenador (SCSI), etc. En el ejemplo particular descrito en la presente memoria, los adaptadores de comunicaciones de I/O son canales, y, por lo tanto, el subsistema de I/O se conoce en la presente memoria como subsistema de canal. No obstante, éste es solamente un ejemplo. Se pueden usar otros tipos de subsistemas de I/O.

El subsistema de I/O usa uno o más caminos de entrada/salida como enlaces de comunicación en la gestión del flujo de información hacia o desde los dispositivos de entrada/salida 106. En este ejemplo particular, estos caminos se denominan caminos de canal, dado que los adaptadores de comunicación son canales.

Otro ejemplo de un entorno informático para incorporar y usar uno o más aspectos de la facilidad de CAM se describe con referencia a la FIG. 1B. En este ejemplo, un entorno informático 150 incluye un complejo de procesador de ordenador central 152 que proporciona soporte de máquina virtual. El CPC 152 se acopla a uno o más dispositivos de entrada/salida (I/O) 106 a través de una o más unidades de control 108. El complejo de procesador de ordenador central 152 incluye, por ejemplo, una memoria de procesador 154 (también conocida como memoria principal, almacenamiento principal, almacenamiento central) acoplada a uno o más procesadores centrales (también conocidos como unidades de procesamiento central (CPU)) 110, y un subsistema de entrada/salida 111.

La memoria de procesador 154 incluye, por ejemplo, una o más máquinas virtuales 162, y el microprograma de procesador 163, que incluye un hipervisor de ordenador central 164 y otro microprograma de procesador 165. Un ejemplo de hipervisor de ordenador central 164 es z/VM[®], ofrecido por International Business Machines Corporation, Armonk, Nueva York.

El soporte de máquina virtual del CPC proporciona la capacidad de operar grandes números de máquinas virtuales 162, cada una capaz de alojar un sistema operativo invitado 172, tal como Linux[®]. Cada máquina virtual 162 es capaz de funcionar como un sistema separado. Es decir, cada máquina virtual se puede reiniciar independientemente, alojar un sistema operativo invitado y operar con diferentes programas 120. Un sistema operativo o programa de aplicaciones que se ejecuta en una máquina virtual parece tener acceso a un sistema pleno y completo, pero en realidad, solamente está disponible una parte de él. Linux es una marca registrada de Linus Torvalds en Estados Unidos, en otros países o en ambos.

En este ejemplo particular, el modelo de máquinas virtuales es un modelo V=V, en el que la memoria absoluta o real de una máquina virtual está respaldada por la memoria virtual del ordenador central, en lugar de la memoria real o absoluta. Cada máquina virtual tiene un espacio de memoria lineal virtual. Los recursos físicos son propiedad del ordenador central 164, y los recursos físicos compartidos se despachan por el ordenador central a los sistemas operativos invitados, según sea necesario, para satisfacer sus demandas de procesamiento. Este modelo de máquina virtual V=V (es decir, invitado paginable) asume que las interacciones entre los sistemas operativos invitados y los recursos de máquina compartida física están controlados por el ordenador central, dado que el gran número de invitados típicamente impide que el ordenador central simplemente divida y asigne los recursos de hardware a los invitados configurados. Uno o más aspectos de un modo V=V se describen además en una publicación de IBM[®] titulada "z/VM: Running Guest Operating Systems", Publicación de IBM[®] N° SC24-5997-02, octubre de 2001.

Los procesadores centrales 110 son recursos de procesador físicos que son asignables a una máquina virtual. Por ejemplo, la máquina virtual 162 incluye uno o más procesadores lógicos, cada uno de los cuales representa todo o una parte de un recurso de procesador físico 110 que se puede asignar dinámicamente a la máquina virtual. Las máquinas virtuales 162 se gestionan por el ordenador central 164.

En una realización preferida de la presente invención, el hardware/microprograma del ordenador central (por ejemplo, z/VM[®]) y del procesador (por ejemplo, System z) interactúan uno con otro de una manera cooperativa controlada con el fin de procesar operaciones de sistema operativo invitado V=V sin requerir transferencia de control desde/hacia el sistema operativo invitado y el ordenador central. Las operaciones de invitado se pueden ejecutar directamente sin la intervención del ordenador central a través de una facilidad que permite que las instrucciones se ejecuten de manera interpretativa para un invitado de modo de almacenamiento paginable. Esta facilidad proporciona una instrucción, Iniciar Ejecución Interpretativa (SIE), que el ordenador central puede emitir, designando un bloque de control denominado descripción de estado que contiene el estado y los controles del invitado (máquina virtual), tales como controles de ejecución y controles de modo. La instrucción pone la máquina en un modo de ejecución interpretativa en el que las instrucciones e interrupciones del invitado se procesan directamente, hasta que surge una condición que requiere la atención del ordenador central. Cuando ocurre tal condición, se finaliza la ejecución interpretativa y o bien se presenta una interrupción del ordenador central o bien la instrucción SIE completa los detalles de almacenamiento de la condición encontrada; esta última acción se llama interceptación. Un ejemplo de ejecución interpretativa se describe en System/370 Extended Architecture/Interpretive Execution, Publicación de IBM N° SA22-7095-01, septiembre de 1985.

En particular, en una realización preferida de la presente invención, la facilidad de ejecución interpretativa proporciona una instrucción para la ejecución de máquinas virtuales. Esta instrucción, llamada Iniciar Ejecución Interpretativa (SIE), se emite por un ordenador central que establece un entorno de ejecución invitado. El ordenador central es el programa de control que gestiona directamente la máquina real y un invitado se refiere a cualquier máquina virtual o interpretada. La máquina se pone en el modo de ejecución interpretativa por el ordenador central, que emite la instrucción SIE. De este modo, la máquina proporciona las funciones de una arquitectura seleccionada (por ejemplo, z/Architecture, ESA/390). Las funciones incluyen, por ejemplo, la ejecución de instrucciones de programa con problemas y privilegiadas, la traducción de direcciones, el manejo de interrupciones y la temporización, entre otras cosas. Se dice que la máquina interpreta las funciones que ejecuta en el contexto de la máquina virtual.

La instrucción SIE tiene un operando, llamado descripción de estado, que incluye información relevante al estado actual del invitado. Cuando finaliza la ejecución de SIE, la información que representa el estado del invitado, incluyendo la PSW invitada, se guarda en la descripción de estado antes de que se devuelva el control al ordenador central.

La arquitectura de ejecución interpretativa proporciona un modo de almacenamiento para el almacenamiento absoluto conocido como modo de almacenamiento paginable. En el modo de almacenamiento paginable, la traducción dinámica de direcciones a nivel de ordenador central se usa para mapear el almacenamiento principal del invitado. El ordenador central tiene la capacidad de dispersar el almacenamiento real de los invitados de modo de almacenamiento paginable a tramas utilizables en cualquier lugar en el almacenamiento real del ordenador central usando la DAT del ordenador central, y para paginar datos del invitado fuera del almacenamiento auxiliar. Esta técnica proporciona flexibilidad cuando se asignan recursos de máquinas reales al tiempo que conserva la apariencia esperada de un rango contiguo de almacenamiento absoluto para el invitado.

Un entorno de máquina virtual puede pedir una aplicación de DAT dos veces: primero a nivel de invitado, para traducir una dirección virtual de invitado a través de tablas de traducción gestionadas de invitado a una dirección real de invitado, y luego, para un invitado paginable, a nivel de ordenador central, para traducir la dirección virtual de ordenador central correspondiente a una dirección real de ordenador central.

En ciertos casos, el ordenador central ha de interceder en operaciones normalmente delegadas a la máquina. Con este propósito, la descripción del estado incluye controles ajustables por el ordenador central para "atrapar" o interceptar, condiciones específicas. Los bits de control de interceptación solicitan que la máquina devuelva el control para simulación del ordenador central cuando se encuentran instrucciones particulares de invitado. Los controles de intervención capturan la introducción de un estado habilitado en la PSW, de modo que el ordenador central pueda presentar una interrupción que tiene pendiente para el invitado. Los controles de intervención se pueden establecer de manera asíncrona por el ordenador central en otro procesador real mientras que procede con la interpretación. La máquina recupera periódicamente los controles de almacenamiento, de modo que se reconocerán los valores actualizados. Por ello, las interrupciones de invitado se pueden dejar pendientes sin una interpretación prematuramente perturbadora.

En una realización preferida de la presente invención, los controles de modo en la descripción de estado especifican si el invitado se ejecuta en el modo de ESA/390 o de z/Architecture y selecciona una de una pluralidad de formas para representar el almacenamiento principal invitado de una máquina virtual de invitado en el almacenamiento del ordenador central. Según una realización preferida de la presente invención, se proporciona un bit de control en un control de estado para seleccionar entre un invitado en un primer y un segundo modo arquitectónico (por ejemplo, z/Architecture y ESA/390, respectivamente). Según otra realización preferida de la presente invención otra realización preferida de la presente invención otra realización preferida de la presente invención, dos instrucciones distintas pueden dotar a un ordenador central con la capacidad de crear una primera y una segunda máquina virtual de invitado, por ejemplo, se pueden proporcionar instrucciones distintas SIEz y SIEe para iniciar máquinas de invitado en modo de z/Architecture y ESA/390, respectivamente.

La instrucción SIE ejecuta un servidor virtual despachado por el programa de control hasta que el segmento de tiempo del servidor se ha consumido o hasta que el servidor quiere realizar una operación que el hardware no puede virtualizar o para la cual el programa de control ha de recuperar el control. En ese punto, la instrucción SIE finaliza y el control se devuelve al programa de control, que o bien simula la instrucción o bien pone al servidor virtual en un estado de espera involuntaria. Cuando se completa, el programa de control de nuevo programa el servidor virtual para ejecutarse, y el ciclo comienza de nuevo. De esta forma, las capacidades completas y la velocidad de la CPU están disponibles para el servidor virtual. Solamente se interceptan aquellas instrucciones privilegiadas que requieren asistencia de o validación por el programa de control. Estas interceptaciones SIE, como se las conoce, también se usan por el programa de control para imponer límites a las operaciones que un servidor virtual puede realizar en un dispositivo real.

Se describen detalles adicionales con respecto a SIE en la arquitectura de ejecución interpretativa ESA/390, fundamento para VM/ESA, Osisek et al., IBM Systems Journal, vol. 30, N° 1, enero de 1991, páginas 34-51.

Otro ejemplo de un entorno informático para incorporar y usar uno o más aspectos de la facilidad de modo arquitectónico de configuración se describe con referencia a la FIG. 2. En este ejemplo, un entorno informático 200 incluye un entorno no particionado que se configura para una pluralidad de modos arquitectónicos, incluyendo la z/Architecture y ESA/390. Incluye, por ejemplo, un procesador (unidad central de procesamiento - CPU) 202 que incluye, por ejemplo, una facilidad de modo arquitectónico de configuración 204, y una o más memorias caché 206. El procesador 202 está acoplado comunicativamente a una parte de memoria 208 que tiene uno o más memorias caché 210, y a un subsistema de entrada/salida (I/O) 212. El subsistema de I/O 212 está acoplado comunicativamente a los dispositivos de I/O 214 externos que pueden incluir, por ejemplo, dispositivos de entrada de datos, sensores y/o dispositivos de salida, tales como visualizadores.

Otra realización preferida de la presente invención de un entorno informático para incorporar y usar uno o más aspectos de la facilidad de modo arquitectónico de configuración se describe con referencia a la FIG. 3A. En este ejemplo, un entorno informático 300 incluye, por ejemplo, una unidad central de procesamiento (CPU) nativa 302, una memoria 304 y uno o más dispositivos de entrada/salida y/o interfaces 306 acoplados entre sí a través de, por ejemplo, uno o más buses 308 y/u otras conexiones. Como ejemplos, el entorno informático 300 puede incluir un procesador PowerPC o un servidor Power Systems ofrecidos por International Business Machines Corporation, Armonk, Nueva York; un HP Superdome con procesadores Intel Itanium II ofrecidos por Hewlett Packard Co., Palo Alto, California; y/u otras máquinas basadas en arquitecturas ofrecidas por International Business Machines Corporation, Hewlett Packard, Intel, Oracle u otras. Intel e Itanium son marcas comerciales o marcas comerciales registradas de Intel Corporation o sus subsidiarias en Estados Unidos y otros países.

La unidad central de procesamiento nativa 302 incluye uno o más registros nativos 310, tales como uno o más registros de propósito general y/o uno o más registros de propósito especial usados durante el procesamiento dentro del entorno, así como una facilidad de modo arquitectónico de configuración 311. Estos registros incluyen información que representa el estado del entorno en cualquier punto particular en el tiempo.

Además, la unidad central de procesamiento nativa 302 ejecuta instrucciones y código que se almacenan en la memoria 304. En un ejemplo particular, la unidad central de procesamiento ejecuta el código de emulador 312 almacenado en la memoria 304. Este código permite que el entorno informático configurado en una arquitectura emule una o más de otras arquitecturas. Por ejemplo, el código de emulador 312 permite que máquinas basadas en arquitecturas distintas de z/Architecture, tales como procesadores PowerPC, servidores Power Systems, servidores HP Superdome u otros, emulen la z/Architecture (y/o ESA/390) y ejecuten software e instrucciones desarrolladas en base a la z/Architecture.

Detalles adicionales relacionados con el código de emulador 312 se describen con referencia a la FIG. 3B. Las instrucciones de invitado 350 almacenadas en la memoria 304 comprenden instrucciones de software (por ejemplo, correlacionando con las instrucciones de máquina) que se desarrollaron para ser ejecutadas en una arquitectura distinta a la de la CPU nativa 302. Por ejemplo, las instrucciones de invitado 350 pueden haber sido diseñadas para ejecutarse en un procesador de z/Architecture 202, pero en su lugar, se están emulando en la CPU nativa 302, que puede ser, por ejemplo, un procesador Intel Itanium II. En un ejemplo, el código de emulador 312 incluye una rutina de búsqueda de instrucciones 352 para obtener una o más instrucciones de invitado 350 de la memoria 304 y, opcionalmente, proporcionar almacenamiento temporal local para las instrucciones obtenidas. También incluye una rutina de traducción de instrucciones 354 para determinar el tipo de instrucción de invitado que se ha obtenido y para traducir la instrucción de invitado en una o más instrucciones nativas 356 correspondientes. Esta traducción incluye, por ejemplo, identificar la función a ser realizada por la instrucción de invitado y elegir la instrucción o las instrucciones nativas para realizar esa función.

Además, el código de emulador 312 incluye una rutina de control de emulación 360 para hacer que las instrucciones nativas se ejecuten. La rutina de control de emulación 360 puede hacer que la CPU nativa 302 ejecute una rutina de instrucciones nativas que emulan una o más instrucciones de invitado obtenidas previamente y, a la conclusión de tal ejecución, devolver el control de la rutina de búsqueda de instrucciones para emular la obtención de la siguiente instrucción de invitado o un grupo de instrucciones de invitado. La ejecución de las instrucciones nativas 356 puede

incluir cargar datos en un registro desde la memoria 304; almacenar datos de vuelta en la memoria desde un registro; o realizar algún tipo de operación aritmética o lógica, como se determina por la rutina de traducción.

5 Cada rutina, por ejemplo, se implementa en software, que se almacena en la memoria y se ejecuta por la unidad central de procesamiento nativa 302. En otros ejemplos, una o más de las rutinas u operaciones se implementan en microprograma, hardware, software o alguna combinación de los mismos. Los registros del procesador emulado se pueden emular usando registros 310 de la CPU nativa o usando ubicaciones en la memoria 304. En realizaciones, las instrucciones de invitado 350, las instrucciones nativas 356 y el código de emulador 312 pueden residir en la misma memoria o se pueden esparcir entre diferentes dispositivos de memoria.

10 Los entornos informáticos descritos anteriormente son solamente ejemplos de entornos informáticos que se pueden usar. Se pueden usar otros entornos, incluyendo, pero no limitados a, otros entornos no particionados, otros entornos particionados y/u otros entornos emulados; las realizaciones no se limitan a cualquier entorno.

15 Según uno o más aspectos, se instala una facilidad de modo arquitectónico de configuración (CAM) en uno o más procesadores (por ejemplo, unidades centrales de procesamiento) de un entorno informático para controlar la reconfiguración del entorno. Por ejemplo, cuando se instala una CAM en un entorno informático que soporta una pluralidad de modos arquitectónicos, el entorno informático se reconfigura de manera que se restringe el uso de uno o más aspectos de al menos uno de los modos arquitectónicos.

20 Un ejemplo particular de una facilidad de modo arquitectónico de configuración es la facilidad de Modo Arquitectónico de Configuración de z/Architecture (CZAM). La instalación de CZAM se indica, por ejemplo, mediante un indicador de instalación de facilidad, por ejemplo, el bit 138, establecido en, por ejemplo, uno. En un ejemplo particular, cuando el bit 138 se establece en uno, la facilidad CZAM está instalada, y cuando está instalada, un reinicio normal y un reinicio de borrado pone la configuración en el modo arquitectónico de z/Architecture. De este modo, el bit de facilidad, por ejemplo, el bit 2, que indica que el modo arquitectónico de z/Architecture está activo también se establece en uno, en un ejemplo.

25 En base a una instalación de CZAM, un entorno informático (por ejemplo, un único procesador, una partición lógica, un invitado virtual, etc.) se reconfigura de manera que uno o más aspectos de una arquitectura seleccionada, por ejemplo, ESA/390, ya no se soporta más. Se describen a continuación aquellos aspectos que ya no se soportan más y/o procesos afectados por la instalación de CZAM.

30 Aunque en las realizaciones descritas en la presente memoria, la pluralidad de modos arquitectónicos incluye una arquitectura heredada (por ejemplo, ESA/390) y una arquitectura mejorada (por ejemplo, z/Architecture) y aspectos de la arquitectura heredada, ESA/390, ya no se soportan más, otras realizaciones pueden incluir otras arquitecturas. ESA/390 y z/Architecture son solamente ejemplos.

35 Un proceso que se ve afectado por la instalación de CZAM es un proceso de encendido. Para describir cómo se ve afectado este proceso, inicialmente, un proceso de encendido para un entorno que soporta múltiples configuraciones arquitectónicas y no incluye la facilidad de CZAM se describe con referencia a las FIG. 4A-4B, y luego un proceso de encendido para un entorno que está configurado para múltiples configuraciones arquitectónicas y que incluye la facilidad de CZAM se describe con referencia a las FIG. 6A-6B. El encendido para un sistema incluye, por ejemplo, arrancar el sistema e iniciar una secuencia de arranque u otros medios de iniciación de operaciones en el sistema. Puede corresponder a un encendido físico, un reinicio de hardware y/o un encendido virtual (por ejemplo, en un sistema emulado, una máquina virtual o un entorno de invitado).

40 Con referencia inicialmente a la FIG. 4A, en base a un procesador del entorno informático que se enciende y una clave de operador, por ejemplo, una clave de carga normal o una clave de borrado de carga, que se activa, el procesador entra en un estado de carga y establece el entorno informático en un modo arquitectónico particular, por ejemplo, el modo de ESA/390, PASO 400. Por ejemplo, se realiza una carga de programa inicial (IPL), tal como una carga de programa inicial (IPL) de palabra de control de canal (CCW), PASO 402. La carga de programa inicial proporciona un medio manual para hacer que un programa se lea desde un dispositivo designado y para iniciar la ejecución de ese programa. Una IPL de tipo CCW se inicia manualmente estableciendo los controles de dirección de unidad de carga en un número de cuatro dígitos para designar un dispositivo de entrada y activando posteriormente la clave de borrado de carga o de carga normal para una CPU particular.

45 La activación de la clave de borrado de carga hace que se realice un reinicio de borrado en la configuración; y la activación de la clave de carga normal hace que se realice un reinicio inicial de la CPU en esta CPU (la CPU en la que se activó la clave), un reinicio de CPU se propague a todas las otras CPU en la configuración, y un reinicio del subsistema se realice en el resto de la configuración. La activación de la clave de borrado de carga o la clave de carga normal establece el modo arquitectónico (por ejemplo, ESA/390).

55 En la parte de carga de la operación, después de que se hayan realizado los reinicios, esta CPU entra en el estado de carga. Esta CPU no necesariamente entra en el estado de parada durante la ejecución de las operaciones de reinicio. El indicador de carga está encendido mientras la CPU está en el estado de carga.

Posteriormente, se inicia una operación de lectura de programa de canal desde el dispositivo de I/O designado por los controles de dirección de unidad de carga. El efecto de ejecutar el programa de canal es como si una CCW de formato 0 que comienza en la ubicación de almacenamiento absoluta 0 especificase un comando de lectura con los bits ceros modificadores, una dirección de datos de cero, un recuento de bytes de 24, el comando de cadena y las banderas de SLI unos, y todas las otras banderas ceros.

Cuando la operación de entrada/salida de IPL se completa con éxito, una palabra de identificación de subsistema para el dispositivo de IPL se almacena en ubicaciones de almacenamiento absoluto seleccionadas (por ejemplo, las ubicaciones 184-187), los ceros se almacenan en otras ubicaciones de almacenamiento absoluto seleccionadas (por ejemplo, las ubicaciones 188-191), y se carga una nueva palabra de estado de programa (PSW) desde ubicaciones de almacenamiento absolutas seleccionadas (por ejemplo, las ubicaciones 0-7), PASO 404. La palabra de estado de programa controla las operaciones del entorno informático.

Si la carga de PSW es un éxito y no se detectan malfuncionamientos en la máquina, esta CPU abandona el estado de carga, y se apaga el indicador de carga. Si el control de tasa se establece en la posición de proceso, la CPU entra en el estado operativo, y la operación del entorno informático pasa bajo el control de la nueva palabra de estado de programa (PSW), PASO 406. Se ejecuta entonces el entorno informático arrancado, PASO 408, como se describe además con referencia a la FIG. 4B.

Con referencia a la FIG. 4B, el entorno informático iniciado se inicia en el modo de ESA/390, PASO 420 y, de este modo, las operaciones se realizan en el modo de ESA/390, PASO 422. En algún punto, se puede hacer una solicitud para cambiar el modo arquitectónico de ESA/390 a z/Architecture. En particular, un programa envía un código de orden (por ejemplo, un código que designa Establecer Arquitectura) al procesador, que emite una instrucción Señalar Procesador (SIGP) con el código de orden para conmutar del modo de ESA/390 al modo de z/Architecture, PASO 424. Por ejemplo, se usa una facilidad de respuesta y señalización de CPU que incluye la instrucción Señalar Procesador (descrita a continuación) y un mecanismo para interpretar y actuar sobre los códigos de orden de servidor, incluyendo uno para Establecer Arquitectura. La facilidad proporciona comunicaciones entre las CPU, incluyendo transmitir, recibir y decodificar un conjunto de códigos de orden asignados; iniciar la operación especificada; y responder a una CPU de señalización. Usando Establecer Arquitectura, el modo arquitectónico se establece en la configuración deseada, por ejemplo, z/Architecture. Se describen además a continuación detalles adicionales de este procesamiento.

A partir de entonces, se hace una determinación de si se aceptó la operación SIGP, CONSULTA 426. En base al código de retorno, se pueden diagnosticar un número de condiciones de error, incluyendo una indicación de "parámetro no válido" cuando se ha hecho una determinación de que la CPU ya está en el modo arquitectónico especificado por el código (es decir, que el establecer arquitectura representa una conmutación al modo actual en sí mismo, o si es una conmutación de un modo a otro modo). Si se acepta el SIGP y el establecer arquitectura representa una operación de conmutación de modo legal, entonces todos los procesadores del entorno informático que recibieron la transición de operación de SIGP al modo de z/Architecture usando, por ejemplo, el procesamiento Establecer Arquitectura descrito en la presente memoria, PASO 428. No obstante, si la operación SIGP no es legal, se indica un error, PASO 430.

Como se ha descrito anteriormente, la operación de encendido carga una palabra de estado de programa. Una realización preferida de la presente invención de un formato de una palabra de estado de programa (PSW) se describe con referencia a la FIG. 5. Con referencia a la FIG. 5, en este ejemplo, el formato de la palabra de estado de programa es un formato de ESA/390, excepto que el bit 31 se muestre como EA, como se indica a continuación.

En una realización preferida de la presente invención, una palabra de estado de programa 500 incluye los siguientes campos, como un ejemplo:

Por Máscara (R) 502: El bit 1 controla si la CPU está habilitada para interrupciones asociadas con la grabación de eventos de programa (PER). Cuando el bit es cero, ningún evento PER puede causar una interrupción. Cuando el bit es uno, se permiten interrupciones, sujetas a los bits de máscara de evento PER en el registro de control 9;

Modo DAT (T) 504: El bit 5 controla si tiene lugar una traducción dinámica de direcciones (DAT) implícita de las direcciones lógicas y de instrucciones usadas para acceder al almacenamiento. Cuando el bit es cero, DAT está apagada y las direcciones lógicas y de instrucciones se tratan como direcciones reales. Cuando el bit es uno, DAT está encendida y se invoca el mecanismo de traducción dinámica de direcciones.

Máscara de I/O (IO) 506: El bit 6 controla si la CPU está habilitada para interrupciones de I/O. Cuando el bit es cero, no puede ocurrir una interrupción de I/O. Cuando el bit es uno, las interrupciones de I/O están sujetas a los bits de máscara de subclase de interrupción de I/O en el registro de control 6. Cuando un bit de máscara de subclase de interrupción de I/O es cero, no puede ocurrir una interrupción de I/O para esa subclase de interrupción de I/O; cuando el bit de máscara de subclase de interrupción de I/O es uno, puede ocurrir una interrupción de I/O para esa subclase de interrupción de I/O;

Máscara Externa (EX) 508: El bit 7 controla si la CPU está habilitada para la interrupción por las condiciones incluidas en la clase externa. Cuando el bit es cero, no puede ocurrir una interrupción externa. Cuando el bit es uno, una interrupción externa está sujeta a los bits de máscara de subclase externa correspondientes en el registro de control 0. Cuando el bit de máscara de subclase es cero, las condiciones asociadas con la subclase no pueden causar una interrupción. Cuando el bit de máscara de subclase es uno, puede ocurrir una interrupción en esa subclase.

Clave de PSW (Clave) 510: Los bits 9-11 forman la clave de acceso para las referencias de almacenamiento por la CPU. Si la referencia está sujeta a protección controlada por clave, la clave de PSW se hace coincidir con una clave de almacenamiento cuando se almacena información o cuando se obtiene información desde una ubicación que está protegida contra búsqueda. No obstante, para uno de los operandos de cada uno de Mover a Primario, Mover a Secundario, Mover con Clave, Mover con Clave Fuente y Mover con Clave Objetivo, y para cualquiera de o ambos operandos de Mover con Especificaciones Opcionales, se usa una clave de acceso especificada como un operando en lugar de la clave de PSW.

Bit 12 512: Este bit indica el modo arquitectónico actual. Se establece en uno para el formato de PSW de ESA/390. Para el formato de PSW de z/Architecture, este bit se define que es cero. Cuando está en modo de z/Architecture, se define una instrucción de carga de PSW extendida (LPSWE) para cargar una PSW de z/Architecture verdadera (que tiene un formato diferente del formato descrito en la presente memoria, incluyendo tener una dirección de instrucción en los bits 64-127). No obstante, todavía se soporta una carga de PSW de ESA/390 (LPSW) y se puede usar para cargar una PSW de formato de ESA/390. Cuando se ejecuta LPSW y el entorno informático está en el modo de z/Architecture, el procesador expande la PSW de formato de ESA/390 al formato de z/Architecture, incluyendo invertir el bit 12. Esto es lo contrario a colapsar el formato de PSW de z/Architecture que el sistema operativo realiza para crear la PSW de formato de ESA/390. Es decir, en entornos informáticos que soportan tanto ESA/390 como z/Architecture, cuando una copia de una PSW se pone en el almacenamiento, el sistema operativo colapsa la PSW de z/Architecture completa al tamaño y formato de una PSW de ESA/390. De este modo, otro software con dependencias de formato de PSW puede no ser consciente de la PSW de z/Architecture.

Máscara de Comprobación de Máquina (M) 514: El bit 13 controla si la CPU está habilitada para la interrupción por las condiciones de comprobación de máquina. Cuando el bit es cero, no puede ocurrir una interrupción de comprobación de máquina. Cuando el bit es uno, se permiten las interrupciones de comprobación de máquina debidas a daños en el sistema y daños en el procesamiento de instrucciones, pero las interrupciones debidas a otras condiciones de subclase de comprobación de máquina están sujetas a los bits de máscara de subclase en el registro de control 14.

Estado de Espera (W) 516: Cuando el bit 14 es uno, la CPU está esperando; es decir, no se procesan instrucciones por la CPU, pero pueden tener lugar interrupciones. Cuando el bit 14 es cero, ocurre la búsqueda y la ejecución de instrucciones de la manera normal. El indicador de espera es uno cuando el bit es uno.

Estado de Problema (P) 518: Cuando el bit 15 es uno, la CPU está en el estado de problema. Cuando el bit 15 es cero, la CPU está en el estado de supervisor. En el estado de supervisor, todas las instrucciones son válidas. En el estado de problema, solamente son válidas aquellas instrucciones que proporcionan información significativa al programa de problema y que no pueden afectar a la integridad del sistema; tales instrucciones se llaman instrucciones no privilegiadas. Las instrucciones que no son válidas en el estado de problema se denominan instrucciones privilegiadas. Cuando una CPU en el estado de problema intenta ejecutar una instrucción privilegiada, se reconoce una excepción de operación privilegiada. Otro grupo de instrucciones, llamadas instrucciones semiprivilegiadas, se ejecutan por una CPU en el estado de problema solamente si se cumplen las pruebas de autoridad específicas; de otro modo, se reconoce una excepción de operación privilegiada o alguna otra excepción de programa, dependiendo del requisito particular que se viole.

Control de Espacio de Direcciones (AS) 520: Los bits 16 y 17, junto con el bit 5 de la PSW, controlan el modo de traducción.

Código de Condición (CC) 522: Los bits 18 y 19 son los dos bits del código de condición. El código de condición se establece en 0, 1, 2 o 3 dependiendo del resultado obtenido en la ejecución de ciertas instrucciones. La mayoría de operaciones aritméticas y lógicas, así como algunas otras operaciones, establecen el código de condición. La instrucción RAMIFICAR BAJO UNA CONDICIÓN puede especificar cualquier selección de los valores de código de condición como criterio para la ramificación.

Máscara de Programa 524: Los bits 20-23 son los cuatro bits de máscara de programa. Cada bit está asociado con una excepción de programa, de la siguiente manera:

Bit de Máscara de Programa	Excepción de Programa
20	Desbordamiento de punto fijo

21	Desbordamiento decimal
22	Subdesbordamiento de exponente de HFP
23	Significación de HFP

Cuando el bit de máscara es uno, la excepción da como resultado una interrupción. Cuando el bit de máscara es cero, no ocurre ninguna interrupción. El ajuste del bit de máscara de subdesbordamiento de exponente de HFP del bit de máscara de significación de HFP también determina la manera en la que se completa la operación cuando ocurre la excepción correspondiente.

5 Modo de Direccionamiento Extendido (EA) 526: El bit 31 controla el tamaño de las direcciones eficaces y la generación de direcciones eficaz junto con el bit 32, el bit del modo de direccionamiento básico. Cuando el bit 31 es cero, el modo de direccionamiento está controlado por el bit 32. Cuando los bits 31 y 32 son ambos uno, se especifica el direccionamiento de 64 bits.

10 Modo de Direccionamiento Básico (BA) 528: Los bits 31 y 32 controlan el tamaño de las direcciones eficaces y la generación de direcciones eficaces. Cuando los bits 31 y 32 son ambos cero, se especifica un direccionamiento de 24 bits. Cuando el bit 31 es cero y el bit 32 es uno, se especifica un direccionamiento de 31 bits. Cuando los bits 31 y 32 son ambos uno, se especifica un direccionamiento de 64 bits. El bit 31 uno y el bit 32 cero es una combinación no válida que hace que se reconozca una excepción de especificación. El modo de direccionamiento no controla el tamaño de las direcciones PER o de las direcciones usadas para acceder a las tablas DAT, ASN, control de unidad despachable, vinculación, entrada y de rastreo, o listas de acceso o pila de vinculación. El control del modo de direccionamiento por los bits 31 y 32 de la PSW se resume de la siguiente manera:

PSW: 31	PSW: 32	Modo de Direccionamiento
0	0	24 bits
0	1	31 bits
1	1	64 bits

20 Dirección de Instrucción 530: Los bits 33-63 de la PSW son la dirección de instrucción. La dirección designa la ubicación del byte de más a la izquierda de la siguiente instrucción a ser ejecutada, a menos que la CPU esté en el estado de espera (el bit 14 de la PSW es uno).

Según un aspecto, cuando una facilidad de modo arquitectónico de configuración, tal como la facilidad de Modo Arquitectónico de Configuración de z/Architecture (CZAM), está instalada y activada en el entorno informático, se cambia el proceso de encendido. Una realización preferida de la presente invención de un proceso de encendido de CZAM se describe con referencia a la FIG. 6A.

25 Con referencia a la FIG. 6A, en base a un procesador del entorno informático que se enciende, el entorno informático se establece en el modo arquitectónico particular especificado por la facilidad de modo arquitectónico de configuración, por ejemplo, el modo de z/Architecture (también conocido como ESAME) cuando CZAM está instalado, PASO 600. Por ejemplo, una carga de programa inicial (IPL), tal como una carga de programa inicial (IPL) de palabra de control de canal (CCW), se realiza, como se ha descrito anteriormente, PASO 602, y cuando la operación de entrada/salida de la IPL se completa con éxito, una palabra de identificación de subsistema para el dispositivo de IPL se almacena en ubicaciones de almacenamiento absoluto seleccionadas (por ejemplo, las ubicaciones 184-187), se almacenan ceros en otras ubicaciones de almacenamiento absoluto seleccionadas (por ejemplo, las ubicaciones 188-191), y en esta realización, se crea una nueva palabra de estado de programa (PSW) de 16 bytes de las ubicaciones de almacenamiento absolutas seleccionadas (por ejemplo, las ubicaciones 0-7), PASO 604. La nueva PSW de 16 bytes se forma, por ejemplo, a partir de los contenidos de la palabra doble de almacenamiento seleccionada (por ejemplo, las ubicaciones 0-7). El bit 12 de la palabra doble ha de ser uno; de otro modo, se puede indicar un error. (El error puede ser una excepción de especificación que se reconoce, una comprobación de máquina o aún otra indicación de error). Los bits 0-32 de la PSW recién creada se establecen en los bits 0-32 de la palabra doble seleccionada, excepto con el bit 12 invertido. Los bits 33-96 de la PSW recién creada se configuran en ceros. Las posiciones de bits 97-127 de la PSW recién creada se inicializan a partir de los bits 33-63 de la palabra doble seleccionada.

45 En una realización preferida de la presente invención, los campos de PSW que han de ser cargados por la instrucción no se comprueban para su validez antes de que se carguen. En una realización preferida de la presente invención, el bit 12 de la PSW se comprueba para su validez. En otra realización preferida más de la presente invención, todos los campos se comprueban para su validez. En otra realización preferida de la presente invención, cualquier bit no comprobado antes de la carga de la PSW se comprueba para su validez después de que se haya inicializado la PSW, y el procesador puede indicar un error (por ejemplo, elevando una excepción de especificación que se reconoce, una comprobación de máquina, o otra indicación de error más).

El entorno informático entra en el estado operativo, y la operación del entorno informático pasa bajo el control de la nueva palabra de estado de programa (PSW), PASO 606. El entorno informático arrancado entonces se ejecuta, PASO 608, como se describe además con referencia a la FIG. 6B.

5 Con referencia a la FIG. 6B, el entorno informático arrancado se inicia en el modo de z/Architecture, PASO 620 y, de este modo, las operaciones se realizan en el modo de z/Architecture, PASO 622. No es necesaria una conmutación de modo, y el procesamiento continúa directamente con el procesamiento en el modo de z/Architecture. De este modo, en una realización preferida de la presente invención, los siguientes pasos no son necesarios: una operación Señalar Procesador (SIGP) para conmutar del modo de ESA/390 al modo de z/Architecture; una determinación en cuanto a si la operación SIGP es una operación aceptada; la transición a z/Architecture si es una operación aceptada; o la indicación de error, si no se acepta la operación SIGP.

10 Todos los procesadores del entorno informático (es decir, el entorno que se configura, por ejemplo, un único procesador, una partición lógica, un invitado VM) están en el modo de z/Architecture, sin realizar los pasos indicados anteriormente. De este modo, como se describe en la presente memoria, según un aspecto, la capacidad de arranque o encendido en el modo de ESA/390 se elimina del entorno informático que está configurado tanto para ESA/390 como para z/Architecture. En particular, aunque un entorno informático está configurado para soportar múltiples arquitecturas, se proporciona una capacidad para restringir ciertos aspectos de al menos una de las arquitecturas configuradas, uno de los aspectos que es la capacidad de encendido en esa arquitectura.

15 En una o más realizaciones, el encendido en modo de z/Architecture proporciona un mecanismo para especificar una de (1) una partición lógica (invitado 1), y (2) una partición lógica e invitado 2 han de ser arrancadas y reiniciadas en el modo de z/Architecture, sin la necesidad de arrancar en el modo de ESA/390. Esta característica se puede instalar incondicionalmente o bajo el control de un conmutador de configuración.

20 La secuencia de arranque con respecto a la inicialización de PSW se modifica. Por ejemplo, al final de la IPL, se carga la PSW de IPL en las ubicaciones absolutas 0-7. Como se hace actualmente cuando la condición de reinicio es ESA/390, el bit 12 es uno, que hace la PSW de IPL de ESA/390 válida, y el programa pasa a ejecutar instrucciones en el modo arquitectónico de ESA/390. Con CZAM instalado, la condición de reinicio es z/Architecture, el bit 12 es todavía uno, haciendo una PSW de IPL de ESA/390 válida, pero el bit 12 se invierte durante la formación de la PSW actual de z/Architecture de 16 byte, como se ha definido anteriormente.

25 Además del proceso de encendido, otros procesos, comportamientos y/u operaciones también se pueden cambiar o ver afectados por la instalación de una facilidad de modo arquitectónico de configuración. Estos procesos, comportamientos y/u operaciones afectados son específicos de los modos de ESA/390 y de z/Architecture. No obstante, procesos similares y/o diferentes se pueden ver afectados por otros tipos de arquitecturas. Los procesos, comportamientos y/u operaciones de ejemplo que se pueden ver afectados en una o más realizaciones incluyen, por ejemplo:

30 (1) Habilitar una conmutación de modo a auto (por ejemplo, de modo de z/Architecture a modo de z/Architecture) sin generar un error (o ignorar el error). Es decir, un procesador puede emitir una instrucción SIGP para conmutar al modo de z/Architecture y, si ya está en ese modo, no se generará ningún error. Previamente, intentar una conmutación al modo correspondiente al modo actual que generó un error.

35 (2) Deshabilitar una conmutación al modo de ESA/390. En base a la instalación y activación de CZAM, la conmutación a ESA/390 está deshabilitada y ahora genera un error. Se evita una conmutación de vuelta a ESA/390 comprobando el bit 12 de la PSW, y tomando una excepción, si el bit 12 no está establecido para indicar el modo de z/Architecture (representado por un bit 12 de "1" en el almacenamiento que se invierte al bit "0" para representar la z/Architecture en la PSW cuando una PSW de ESA/390 se convierte en una PSW de z/Architecture válida).

40 (3) Modificar la operación de Cargar PSW para restringir el manejo del bit 12. Si está instalada la facilidad de Modo Arquitectónico de Configuración de z/Architecture, Cargar PSW reconoce una excepción de especificación si el bit 12 de su segundo operando no es uno. Cargar PSW carga los bits 0-32 de su segundo operando, excepto con el bit 12 invertido, y los bits 33-63 del operando como los bits 0-32 y 97-127, respectivamente de la PSW actual, y establece los bits 33-96 de la PSW actual en ceros.

45 Los detalles adicionales con respecto a la instrucción Cargar PSW se describen con referencia a la FIG. 7. En una realización preferida de la presente invención, una instrucción Cargar PSW 700 incluye un campo de código de operación 702 que incluye un código de operación para indicar una operación de cargar PSW; un campo base (B₂) 704; y un campo de desplazamiento (D₂) 706. Los contenidos del registro general designado por el campo B₂ se añaden a los contenidos del campo D₂ para formar una dirección de un segundo operando en el almacenamiento (conocida como la segunda dirección de operando).

50 En la operación de la instrucción Cargar PSW, la PSW actual se sustituye por una PSW de 16 bytes formada a partir de los contenidos de la palabra doble en la ubicación designada por la segunda dirección de operando.

El bit 12 de la palabra doble ha de ser uno; de otro modo, se puede reconocer una excepción de especificación, dependiendo del modelo. Si la facilidad de Modo Arquitectónico de Configuración de z/Architecture está instalada, entonces se reconoce una excepción de especificación si el bit 12 de la palabra doble no es uno.

5 Los bits 0-32 de la palabra doble, excepto con el bit 12 invertido, se ponen en las posiciones 0-32 de la PSW actual. Los bits 33-63 de la palabra doble se ponen en las posiciones 97-127 de la PSW actual. Los bits 33-96 de la PSW actual se establecen en cero.

Se realiza una función de serialización y de sincronización de punto de control antes o después de que se obtenga el operando y de nuevo después de que se completa la operación.

10 El operando ha de ser designado en un límite de palabra doble; de otro modo, se reconoce una excepción de especificación. Se puede reconocer una excepción de especificación si el bit 12 del operando es cero, dependiendo del modelo.

15 Los campos de PSW que han de ser cargados por la instrucción no se comprueban para su validez antes de que se carguen, excepto para la comprobación del bit 12. No obstante, inmediatamente después de la carga, se reconoce una excepción de especificación, y ocurre una interrupción del programa, cuando cualquiera de lo siguiente es cierto para la PSW recién cargada:

- Cualquiera de los bits 0, 2-4, 12 o 24-30 es un uno.
- Los bits 31 y 32 son ambos cero, y los bits 97-103 no son todos ceros.
- Los bits 31 y 32 son uno y cero, respectivamente.

En estos casos, la operación se completa, y el código de longitud de instrucción resultante es 0.

20 La operación se suprime en todas las excepciones de direccionamiento y protección.

Código de condición resultante: El código se establece como se especifica en la nueva PSW cargada.

Excepciones de Programa:

- Acceso (buscar, operando 2)
- Operación privilegiada
- 25 • Especificación

Nota de Programación: El segundo operando debería tener el formato de una PSW de ESA/390. Se reconocerá una excepción de especificación durante o después de la ejecución de CARGAR PSW si el bit 12 del operando es cero.

Se describen detalles adicionales con respecto a la PSW en "Development and Attributes of z/Architecture", Plambeck et al., IBM J. Res. & Dev., Vol. 46, N° 4/5, julio/septiembre de 2002.

30 Además de los procesos, operaciones y/o comportamientos anteriores que se pueden cambiar debido a la instalación de una facilidad de modo arquitectónico de configuración, el modo de reinicio también se puede cambiar en una o más realizaciones, como se explica a continuación.

35 (4) Cambia el modo de reinicio (por ejemplo, para reiniciar, reinicio de borrado y otras acciones para reiniciar). Cuando la facilidad de CZAM está instalada, el reinicio de la CPU establece el modo arquitectónico en el modo de z/Architecture, si se causa por activación, por ejemplo, de la clave de carga normal.

Hay una serie de funciones de reinicio que se incluyen como parte de los modos de ESA/390 y de z/Architecture, incluyendo, por ejemplo, reinicio de la CPU, reinicio de la CPU inicial, reinicio del subsistema, reinicio del borrado y reinicio del encendido, cada uno de los cuales se describen a continuación.

Reinicio de la CPU

40 El reinicio de la CPU proporciona un medio de borrado de las indicaciones de comprobación del equipo y cualquier imprevisibilidad resultante en el estado de la CPU con la menor cantidad de información destruida. En particular, se usa para borrar las condiciones de comprobación cuando el estado de la CPU ha de ser conservado para el análisis o la reanudación de la operación. Si la facilidad de Modo Arquitectónico de Configuración de z/Architecture (CZAM) no está instalada, el reinicio de la CPU establece el modo arquitectónico en el modo de ESA/390 si se causa por la

45 activación de la clave de carga normal (una facilidad del operador). Cuando la facilidad de CZAM está instalada, el reinicio de la CPU establece el modo arquitectónico en el modo de z/Architecture si se causa por la activación de la clave de carga normal. Cuando el reinicio de la CPU establece el modo de ESA/390, guarda la PSW actual de modo

que se pueda restaurar la PSW mediante una orden de Establecer Arquitectura Señalar Procesador que cambia el modo arquitectónico de vuelta a z/Architecture.

El reinicio de la CPU causa las siguientes acciones, en una realización preferida de la presente invención:

- 5 1. Se termina la ejecución de la instrucción actual u otra secuencia de procesamiento, tal como una interrupción, y se borran todas las condiciones de interrupción de programa y de interrupción de llamada de supervisor.
2. Se borra cualquier condición de interrupción externa pendiente que es local para la CPU. Las condiciones de interrupción externa flotante no se borran.
- 10 3. Se borran cualquier condición de interrupción de comprobación de máquina pendiente e indicaciones de error que son locales para la CPU y cualquier estado de comprobación de parada. Las condiciones de interrupción de comprobación de máquina flotante no se borran. Cualquier condición de comprobación de máquina que se notifica a todas las CPU en la configuración y que se ha hecho pendiente para una CPU se dice que es local a la CPU.
- 15 4. Se borran todas las copias de las instrucciones o los operandos obtenidos previamente. Además, se borra cualquier resultado a ser almacenado debido a la ejecución de las instrucciones en el intervalo de punto de comprobación actual.
5. El almacenador temporal anticipado de ART (Traducción de Registros de Acceso) y el almacenador temporal de traducción anticipada se borran de entradas.
6. Si el reinicio está causado por activación de la clave de carga normal en cualquier CPU en la configuración, ocurren las siguientes acciones:
 - 20 a. Cuando la facilidad de CZAM no está instalada, el modo arquitectónico de la CPU (y de todas las otras CPU en la configuración debida al reinicio de la CPU inicial o al reinicio de la CPU realizado por ellos) se cambia del modo de z/Architecture al modo de ESA/390. Si la facilidad de CZAM está instalada, el modo arquitectónico de la CPU (y de todas las otras CPU en la configuración debida al reinicio de la CPU inicial o al reinicio de la CPU realizado por ellos) se establece en el modo de z/Architecture.
 - 25 b. Cuando la facilidad de CZAM no está instalada, la PSW actual se guarda para su uso posterior por una orden de Establecer Arquitectura Señalar Procesador que restaura el modo de z/Architecture.
 - 30 c. Cuando la facilidad de CZAM no está instalada, la PSW actual se cambia de 16 bytes a ocho bytes. Los bits de la PSW de ocho bytes se establecen de la siguiente manera: los bits 0-11 y 13-32 se establecen iguales a los mismos bits de la PSW de 16 bytes, el bit 12 se establece en uno, y los bits 33-63 se establecen iguales a los bits 97-127 de la PSW de 16 bytes.

Un reinicio de la CPU causado por la activación de la clave de reinicio normal del sistema o por la orden de Reinicio de la CPU del Señalar Procesador, y cualquier reinicio de la CPU en el modo de ESA/390, no afecta al registro de la PSW de z/Architecture capturada (es decir, una PSW guardada cuando la CPU fue por última vez del modo de z/Architecture al modo de ESA/390 debido a una orden de Establecer Arquitectura con código 0 o un reinicio de la CPU debido a la activación de la clave de carga normal).

- 35 7. La CPU se pone en estado de parada después de que se hayan completado las acciones 1-6. Cuando la secuencia de IPL de tipo CCW sigue a la función de reinicio en esa CPU, la CPU entra en el estado de carga a la terminación de la función de reinicio y no necesariamente entra en el estado parado durante la ejecución de la operación de reinicio. Cuando la secuencia de IPL dirigida por lista sigue a la función de reinicio en esa CPU, la CPU entra en el estado operativo y no necesariamente entra en el estado parado durante la ejecución de la operación de reinicio.

Los registros, los contenidos de almacenamiento y el estado de las condiciones externas para la CPU permanecen sin cambios por el reinicio de la CPU. No obstante, los contenidos posteriores del registro, la ubicación o el estado son impredecibles si está en curso una operación que cambia los contenidos en el momento del reinicio. Un bloqueo mantenido por la CPU cuando se ejecuta REALIZAR OPERACIÓN BLOQUEADA no se libera mediante el reinicio de la CPU.

Cuando la función de reinicio en la CPU se inicia en el momento en que la CPU está ejecutando una instrucción de I/O o está realizando una interrupción de I/O, la operación actual entre la CPU y el subsistema de canal se puede o no completar, y el estado resultante de la facilidad de subsistema de canal asociada puede ser impredecible.

50 Notas de programación:

1. La mayoría de las operaciones que cambiarían un estado, una condición o los contenidos de un campo no pueden ocurrir cuando la CPU está en el estado parado. No obstante, algunas funciones de señalar procesador y algunas

funciones del operador pueden cambiar estos campos. Para eliminar la posibilidad de perder un campo cuando se emite el reinicio de la CPU, la CPU se debería parar, y no debería estar en curso ninguna función de operador.

2. Si el modo arquitectónico se cambia al modo de ESA/390 y el bit 31 de la PSW actual es uno, la PSW no es válida.

5 Reinicio de la CPU inicial

El reinicio de la CPU inicial proporciona las funciones de reinicio de la CPU junto con la inicialización de la PSW actual, la PSW de z/Architecture capturada, el temporizador de la CPU, el comparador de reloj, el prefijo, el control de la dirección de evento de ruptura, el control de punto flotante y registros programables de la hora del día (TOD). Si la facilidad de CZAM no está instalada, el reinicio de la CPU inicial establece el modo arquitectónico en el modo de ESA/390 si se causa por la activación de la clave de carga normal. Cuando la facilidad de CZAM está instalada, el reinicio de la CPU inicial establece el modo arquitectónico en el modo de z/Architecture si se causa por la activación de la clave de carga normal.

El reinicio de la CPU inicial combina las funciones de reinicio de la CPU con las siguientes funciones de borrado e inicialización:

- 15 1. Cuando la facilidad de CZAM no está instalada, si el reinicio se causa por la activación de la clave de carga normal, el modo arquitectónico de la CPU (y de todas las otras CPU en la configuración) se establece en el modo de ESA/390. De otro modo, si la facilidad de CZAM está instalada, el modo arquitectónico de la CPU (y de todas las otras CPU en la configuración) se establece en el modo de z/Architecture.
- 20 2. Los contenidos de la PSW actual, PSW de z/Architecture capturada, prefijo, temporizador de la CPU, comparador de reloj y registro programable de TOD se establece en cero. Cuando la secuencia de IPL sigue la función de reinicio en esa CPU, los contenidos de la PSW no se establecen necesariamente en cero.
- 3. Los contenidos de los registros de control se establecen en sus valores de z/Architecture iniciales. Todos los 64 bits de los registros de control se configuran independientemente de si la CPU está en el modo arquitectónico de ESA/390 o de z/Architecture.
- 25 4. Los contenidos del registro de control de punto flotante se establecen en cero.
- 5. Los contenidos del registro de la dirección del evento de interrupción se inicializan a 0000000000000001 hex.

Estas funciones de borrado e inicialización incluyen validación.

El establecimiento de la PSW actual en cero cuando la CPU está en el modo arquitectónico de ESA/390 al final de la operación hace que la PSW no sea válida, dado que el bit 12 de la PSW ha de ser uno en ese modo. De este modo, en este caso, si la CPU se pone en el estado operativo después de un reinicio sin introducir primero una nueva PSW, se reconoce una excepción de especificación.

Reinicio del subsistema

El reinicio del subsistema proporciona un medio para borrar las condiciones de interrupción flotantes, así como para invocar el reinicio del sistema de I/O.

35 Reinicio del borrado

El reinicio del borrado hace que el reinicio de la CPU inicial y el reinicio del subsistema se realicen y, además, borra o inicializa todas las ubicaciones y registros de almacenamiento en todas las CPU en la configuración, con la excepción del reloj de TOD. Tal borrado es útil en la depuración de programas y en el aseguramiento de la privacidad del usuario. El reinicio del borrado también libera todos los bloqueos usados por la instrucción REALIZAR OPERACIÓN BLOQUEADA. Si la facilidad de CZAM no está instalada, el reinicio del borrado establece el modo arquitectónico en el modo de ESA/390. Cuando la facilidad de CZAM está instalada, el reinicio del borrado establece el modo arquitectónico en el modo de z/Architecture. El borrado no afecta el almacenamiento externo, tal como los dispositivos de almacenamiento de acceso directo usados por el programa de control para mantener los contenidos de las páginas no direccionables.

45 El reinicio del borrado combina la función de reinicio de la CPU inicial con una función de inicialización lo que causa las siguientes acciones:

- 1. Cuando la facilidad de CZAM no está instalada, el modo arquitectónico de todas las CPU en la configuración se establece en el modo de ESA/390. Si la facilidad de CZAM está instalada, el modo arquitectónico de todas las CPU en la configuración se establece en el modo de z/Architecture.

2. Los registros de acceso, generales y de punto flotante de todas las CPU en la configuración se establecen en cero. Todos los 64 bits de los registros generales se establecen en cero independientemente de si la CPU estaba en el modo arquitectónico de ESA/390 o de z/Architecture cuando se inició la función de reinicio del borrado.

5 3. Los contenidos del almacenamiento principal en la configuración y las claves de almacenamiento asociadas se establecen en cero con un código de bloque de comprobación válido.

4. Se liberan los bloqueos usados por cualquier CPU en la configuración cuando se ejecuta la instrucción REALIZAR OPERACIÓN BLOQUEADA.

5. Se realiza un reinicio del subsistema.

10 La validación se incluye en el establecimiento de registros y en el borrado del almacenamiento y de las claves de almacenamiento.

Notas de programación:

1. El modo arquitectónico no se cambia por la activación de la clave normal de reinicio del sistema o por la ejecución de una orden de Reinicio de CPU de Señalar Procesador o de Reinicio de la CPU inicial. Todas las CPU en la configuración están en el mismo modo arquitectónico.

15 2. Para que la operación de reinicio de la CPU no afecte a los contenidos de los campos que han de ser dejados sin cambios, la CPU no ha de estar ejecutando las instrucciones y ha de ser deshabilitada para todas las interrupciones en el momento del reinicio. Excepto por la operación del temporizador de la CPU y por la posibilidad de que ocurra una interrupción de comprobación de máquina, toda la actividad de la CPU se puede detener poniendo la CPU en el estado de espera y deshabilitándola para la I/O y las interrupciones externas. Para evitar la posibilidad de causar un
20 reinicio en el momento que el temporizador de la CPU está siendo actualizado u ocurra una interrupción de comprobación de máquina, la CPU ha de estar en el estado parado.

3. El reinicio de la CPU, el reinicio de la CPU inicial, el reinicio del subsistema y el reinicio del borrado no afectan al valor y al estado del reloj de TOD.

25 4. Las condiciones bajo las cuales la CPU entra en el estado de comprobación de parada son dependientes del modelo e incluyen malfuncionamientos que impiden la terminación de la operación actual. Por lo tanto, si el reinicio de la CPU o el reinicio de la CPU inicial se ejecuta mientras que la CPU está en el estado de comprobación de parada, los contenidos de la PSW, los registros y las ubicaciones de almacenamiento, incluyendo las claves de almacenamiento y la ubicación de almacenamiento a la que se accede en el momento del error, puede tener valores impredecibles y, en algunos casos, los contenidos aún puede estar en error después de que se borre el estado de
30 comprobación de parada mediante estos reinicios. En esta situación, se requiere un reinicio del borrado para borrar el error.

Reinicio del encendido

35 La función de reinicio del encendido para un componente de la máquina se realiza como parte de la secuencia de encendido para ese componente. Las secuencias de encendido para el reloj de TOD, el almacenamiento principal, el almacenamiento expandido y el subsistema de canal se pueden incluir como parte de la secuencia de encendido de la CPU, o se puede iniciar por separado la secuencia de encendido para estas unidades.

40 Reinicio del encendido de la CPU: El reinicio de encendido hace que el reinicio de la CPU inicial se realice y puede o no causar que el reinicio del sistema de I/O se realice en el subsistema de canal. Los contenidos de los registros generales, registros de acceso y registros de punto flotante se borran a ceros con un código de bloque de comprobación válido. Los bloqueos usados por REALIZAR OPERACIÓN BLOQUEADA y asociados con la CPU se liberan a menos que se mantengan por una CPU ya encendida. Si la facilidad de CZAM no está instalada y el reinicio está asociado con el establecimiento de una configuración, la CPU se pone en el modo de ESA/390; de otro modo, la CPU se pone en el modo arquitectónico de las CPU ya en la configuración. Si la facilidad de CZAM está instalada, la CPU se pone en el modo de z/Architecture.

45 El reinicio de la CPU, el reinicio de la CPU inicial, el reinicio del subsistema y el reinicio del borrado se pueden iniciar manualmente usando las facilidades del operador. El reinicio de la CPU inicial es parte de la función de carga del programa inicial. El reinicio del encendido se realiza como parte del encendido.

50 Cuando la facilidad de CZAM no está instalada, si el reinicio se inicia por la clave de reinicio del borrado del sistema, de carga normal, o borrado de carga o por un reinicio del encendido de la CPU que establece la configuración, se establece el modo arquitectónico al modo de ESA/390; de otro modo, el modo arquitectónico no se cambia, excepto que el reinicio de encendido establezca el modo al de las CPU que ya están en la configuración. Si la facilidad de CZAM está instalada, el modo arquitectónico se establece en el modo de z/Architecture.

Otros procesos, operaciones y/o comportamientos que se pueden cambiar debido a la instalación de una facilidad de modo arquitectónico de configuración se describen a continuación:

5 (5) Suprime otras acciones relacionadas con el reinicio que se toman para facilitar el cambio entre el modo de ESA/390 y de z/Architecture, cuando se realiza el reinicio. Cuando la facilidad de CZAM no está instalada, la PSW actual se guarda para su uso posterior mediante una orden de Establecer Arquitectura Señalar Procesador que restaura el modo de z/Architecture. Cuando la facilidad de CZAM no está instalada, la PSW actual se cambia de 16 bytes a ocho bytes. Los bits de la PSW de ocho bytes se establecen de la siguiente manera, en un ejemplo: los bits 0-11 y 13-32 se establecen iguales a los mismos bits de la PSW de 16 bytes, el bit 12 se establece en uno y los bits 33-63 se establecen iguales a los bits 97-127 de la PSW de 16 bytes. Cuando la facilidad de CZAM está instalada, la PSW no se guarda para un uso posterior por una orden de Establecer Arquitectura Señalar Procesador que restaura el modo de z/Architecture, y la PSW actual no se cambia de 16 bytes a 8 bytes.

15 (6) Cambia el proceso para configurar una CPU con un comando configurar SCLP (Procesador Lógico de Llamada de Servicio) de CPU, y cargar las operaciones clave. En lugar de configurar en ESA/390, configurar en el modo definido por el reinicio. El comando configurar SCLP de CPU pone la CPU objetivo en el modo arquitectónico de las CPU ya en el estado configurado. Al menos la primera CPU puesta en una configuración se pone allí junto con una potencia de CPU en el reinicio y, como parte de ese reinicio, se pone en el modo de arquitectura definido en el reinicio del encendido de la CPU. Un modelo puede establecer alternativamente el modo de las CPU que están en el estado de espera cuando establece el modo de las CPU configuradas.

La activación de la clave de borrado de carga o la clave de carga normal establece el modo arquitectónico como se define en el reinicio del borrado o el reinicio de la CPU inicial, respectivamente.

20 (7) Cambia el SIGP para no permitir que una orden de Establecer Arquitectura cambie el modo arquitectónico a ESA/390.

25 Una realización preferida de la presente invención de una instrucción Señalar Procesador (SIGP) se describe con referencia a la FIG. 8A. En una realización preferida de la presente invención, una instrucción Señalar Procesador 800 tiene una pluralidad de campos, incluyendo, por ejemplo, un campo de código de operación (código de operación) 802 que tiene un código de operación que indica una operación señalar procesador; un primer campo de registro (R₁) 804; un segundo campo de registro (R₃) 806; un campo base (B₂) 808; y un campo de desplazamiento (D₂) 810. R₁ designa un registro general, los contenidos del cual son el primer operando; R₃ designa un registro general, los contenidos del cual son el tercer operando; y los contenidos de un registro designado por R₂ se añaden al desplazamiento en D₂ para proporcionar una dirección de un segundo operando.

30 En operación, un código de orden de ocho bits y, si se pide, un parámetro de 32 bits se transmite a la CPU designada por la dirección de la CPU contenida en el tercer operando. El resultado se indica mediante el código de condición y se puede detallar mediante el estado ensamblado en las posiciones de bits 32-63 de la ubicación del primer operando.

35 La dirección del segundo operando no se usa para direccionar datos; en su lugar, los bits 56-63 de la dirección contienen el código de orden de ocho bits. Los bits 0-55 de la dirección del segundo operando se ignoran. El código de orden especifica la función a ser realizada por la CPU direccionada. La asignación y definición de códigos de orden incluyen, por ejemplo, los siguientes, en un ejemplo:

Código		
(Dec)	(Hex)	Orden
0	00	No asignado
1	01	Percibir
2	02	Llamada externa
3	03	Señal de emergencia
4	04	Iniciar
5	05	Parar
6	06	Reiniciar
7	07	No asignado
8	08	No asignado
9	09	Parar y almacenar estado
10	0A	No asignado

ES 2 717 525 T3

11	0B	Reinicio de la CPU inicial
12	0C	Reinicio de la CPU
13	0D	Establecer prefijo
14	0E	Almacenar estado en dirección
15-17	0F-11	No asignados
18	12	Establecer Arquitectura
19	13	Señal de emergencia condicional
14	14	No asignado
21	15	Detectar estado de ejecución
22-255	16-FF	No asignados

El número binario de 16 bits contenido en las posiciones de bits 48-63 del registro general R_3 forma la dirección de la CPU. Los bits 0-47 del registro se ignoran. Cuando la orden especificada es la orden Establecer Arquitectura, la dirección de la CPU se ignora; todas las otras CPU en la configuración se consideran que están direccionadas.

5 El registro general que contiene el parámetro de 32 bits en las posiciones de bits 32-63 es R_1 o R_1+1 , cualquiera que sea el registro numerado impar. Depende del código de orden si se proporciona un parámetro y para qué propósito se usa.

Los operandos que se acaban de describir tienen los siguientes formatos, en un ejemplo:

Registro general designado por R_1 : Los bits 0-31 no usados; los bits 32-63 incluyen el estado;

10 Registro general designado por R_1 o R_1+1 , cualquiera que sea el registro numerado impar: Los bits 0-31 no usados; los bits 32-63 incluyen el parámetro;

Registro general designado por R_3 : Los bits 0-48 no usados; los bits 49-63 incluyen la dirección de la CPU;

Dirección del segundo operando: Los bits 0-55 no usados; los bits 56-63 incluyen el código de orden.

Se realiza una función de serialización antes de que comience la operación y de nuevo una vez que se complete la operación.

15 Cuando se acepta el código de orden y no se devuelve ningún estado no cero, se establece el código de condición 0. Cuando una información de estado se genera por esta CPU (la CPU que realiza el SIGP) o se devuelve por la CPU direccionada, el estado se pone en las posiciones de bits 32-63 del registro general R_1 , los bits 0-31 del registro permanecen sin cambios, y se establece el código de condición 1.

20 Cuando el camino de acceso a la CPU direccionada está ocupado, o la CPU direccionada está operativa pero en un estado donde no puede responder al código de orden, se establece el código de condición 2.

Cuando la CPU direccionada no está operativa (es decir, no se proporciona en la instalación, no está en la configuración, está en cualquiera de ciertos modos de prueba de cliente ingeniero, o su alimentación está apagada), se establece el código de condición 3.

Código de condición resultante:

25 0 Código de orden aceptado

1 Estado almacenado

2 Ocupado

3 No operativo

Excepciones del programa:

- 30
- Operación privilegiada
 - Restricción transaccional

5 Cuando se especifica la orden Establecer Arquitectura Señalar Procesador en las posiciones de bits 56-63 de la dirección del segundo operando de la instrucción Señalar Procesador, los contenidos de las posiciones de bits 56-63 del registro de parámetros se usan como un código que especifica un modo arquitectónico al que han de ser establecidas todas las CPU en la configuración: el código 0 especifica el modo de ESA/390, y los códigos 1 y 2 especifican el modo de z/Architecture. El código 1 especifica que, para cada una de todas las CPU en la configuración, la PSW de ESA/390 actual ha de ser transformada a una PSW de z/Architecture. El Código 2 especifica que la PSW de la CPU que ejecuta Señalar Procesador ha de ser transformada a una PSW de z/Architecture y que, para cada una de todas las otras CPU en la configuración, la PSW se ha de establecer con el valor del registro de la PSW de z/Architecture capturada para esa CPU. El establecimiento de la PSW con el valor del registro de la PSW de z/Architecture capturada restaurará la PSW que existía cuando la CPU estuvo por última vez en el modo de z/Architecture, a condición de que no se haya establecido el registro de la PSW de z/Architecture capturada en todo ceros por un reinicio.

Se ignoran los bits 0-55 del registro de parámetros. Se ignoran los contenidos del registro de direcciones de la CPU de la instrucción Señalar Procesador; todas las otras CPU en la configuración se consideran que se direccionan.

15 Cuando la facilidad de CZAM no está instalada, la orden se acepta solamente si el código es 0, 1 o 2, la CPU no está ya en el modo especificado por el código, cada una de las otras CPU está o bien en el estado parado o bien en el estado de comprobación de parada, y ninguna otra condición impide aceptar la orden.

20 Cuando la facilidad de CZAM está instalada, no se acepta el código 0 debido a que no se permite una vuelta al modo de ESA/390, y dado que la CPU ya está en el modo arquitectónico de z/Architecture, la especificación de los códigos 1 y 2 da como resultado una terminación que indica parámetro inválido y código de condición 1. Las otras condiciones de requisito previo normalmente verificadas por la orden Establecer Arquitectura pueden o no ser comprobadas.

Si se acepta, la orden se completa por todas las CPU durante la ejecución de Señalar Procesador. En ningún caso, en esta realización, diferentes CPU pueden estar en diferentes modos arquitectónicos.

25 La orden Establecer Arquitectura se completa, de la siguiente manera, en un ejemplo:

- Si el código en el registro de parámetros no es 0, 1 o 2, o si la CPU ya está en el modo arquitectónico especificado por el código, no se acepta la orden. En su lugar, el bit 55 (parámetro no válido) del registro general designado por el campo R₁ de la instrucción Señalar Procesador se establece en uno, y se establece el código de condición 1.
- 30 • Si no es cierto que todas las otras CPU de la configuración están en estado parado o de comprobación de parada, no se acepta la orden. En su lugar, el bit 54 (estado incorrecto) del registro general designado por el campo R₁ de la instrucción Señalar Procesador se establece en uno, y se establece el código de condición 1.
- 35 • El modo arquitectónico de todas las CPU en la configuración se establece como se especifica por el código (por ejemplo, el bit 12 de la PSW para controlar las operaciones se establece en el modo arquitectónico especificado, y/o se establece otra indicación en el entorno informático que indica el modo arquitectónico especificado).
- 40 • Si la orden cambia el modo arquitectónico de ESA/390 a z/Architecture y el código es 1, entonces, para cada CPU en la configuración, la PSW actual de ocho bytes se cambia a una PSW de 16 bytes, y los bits de la PSW de 16 bytes se establecen de la siguiente manera: los bits 0-11 y 13-32 se establecen iguales a los mismos bits de la PSW de ocho bytes, el bit 12 y los bits 33-96 se establecen en ceros, y los bits 97-127 se establecen iguales a los bits 33-63 de la PSW de ocho bytes. Además, el bit 19 del prefijo de ESA/390, que llega a ser el bit 51 del prefijo de z/Architecture, se establece en cero.
- 45 • Si el código es 2, la PSW de la CPU que ejecuta Señalar Procesador y los valores de prefijo de todas las CPU se establecen como en el caso de código 1. Para cada una de todas las otras CPU en la configuración, la PSW se establece con el valor del registro de la PSW de z/Architecture capturada. No obstante, el registro de la PSW de z/Architecture capturada se ha establecido a todo ceros si la CPU realizó un reinicio, distinto del reinicio de la CPU, o bien en el momento de la transición del modo arquitectónico o bien posteriormente.
- 50 • Si la orden cambia el modo arquitectónico de z/Architecture a ESA/390, entonces, para cada CPU en la configuración, (1) la PSW actual, que es la PSW actualizada en el caso de la CPU que ejecuta Señalar Procesador, se guarda en el registro de la PSW de z/Architecture capturada, y (2) la PSW actual de 16 bytes se cambia a una PSW de ocho bytes estableciendo los bits de la PSW de ocho bytes de la siguiente manera: los bits 0-11 y 13-32 se establecen iguales a los mismos bits de la PSW de 16 bytes, el bit 12 se establece en uno, y los bits 33-63 se establecen iguales a los bits 97-127 de la PSW de 16 bytes. El bit 51 del prefijo de z/Architecture, que llega a ser el bit 19 del prefijo de ESA/390, permanece sin cambios.
- 55

- Los ALB y TLB de todas las CPU en la configuración se borran de sus contenidos.
- Se realiza una función de serialización y sincronización de punto de control en todas las CPU en la configuración.

5 Si la orden cambia el modo arquitectónico de z/Architecture a ESA/390 y la instrucción Señalar Procesador causa la aparición de un evento PER de búsqueda de instrucciones, solamente los 31 bits de más a la derecha de la dirección de la instrucción se almacenan en el campo de dirección PER de ESA/390.

10 En una realización preferida de la presente invención, con CZAM, el siguiente es un requisito previo: Cada una de todas las otras CPU está o bien en el estado de parado o bien de comprobación de parada, y ninguna otra condición excluye la aceptación de la orden. Cuando la facilidad de CZAM está instalada, no se acepta el código 0 debido a que no se permite una vuelta al modo de ESA/390, y dado que la CPU ya está en el modo arquitectónico de z/Architecture, la especificación de los códigos 1 y 2 da como resultado una indicación de terminación que indica un parámetro no válido y el código de condición 1. Se pueden comprobar o no otras condiciones de requisito previo normalmente verificadas por la orden Establecer Arquitectura. En otra realización preferida más de la presente invención, SIGP con los códigos 1 y 2 indica una terminación con éxito sin indicación adicional.

15 Una realización preferida de la presente invención de procesamiento asociado con la ejecución de una instrucción SIGP para un código de orden Establecer Arquitectura se describe con referencia a la FIG. 8B. Con referencia a la FIG. 8B, un procesador del entorno informático ejecuta una instrucción SIGP y obtiene un código de orden que indica una operación Establecer Arquitectura, PASO 850. En un ejemplo, el código de orden se incluye en la dirección del segundo operando de la instrucción SIGP.

20 Además, el modo arquitectónico solicitado a ser conmutado se obtiene a partir de, por ejemplo, el registro de parámetros especificado por la instrucción SIGP, PASO 852. Además, se hace una determinación en cuanto a si una facilidad de modo arquitectónico de configuración, tal como CZAM, está instalada, CONSULTA 854. Esto se determina, en un ejemplo, mediante un indicador de facilidad.

25 Si CZAM no está instalado, entonces se hace una determinación adicional en cuanto a si la CPU ya está en el modo arquitectónico solicitado, CONSULTA 856. Si es así, entonces se proporciona el estado, por ejemplo, en un registro designado por la instrucción SIGP, PASO 858, y el estado se trata como error, PASO 860. No obstante, si la CPU no está en el modo solicitado, CONSULTA 856, entonces se hace una determinación en cuanto a si se cumplen otras condiciones especificadas por la instrucción, tales como si las otras CPU del entorno informático que se configura están en un estado parado, etc., CONSULTA 862. Si no se cumplen las condiciones, entonces el procesamiento continúa al PASO 858. De otro modo, se acepta la orden, PASO 864, y el modo arquitectónico ha de ser cambiado. De este modo, la PSW se establece, como se ha descrito anteriormente, PASO 866, y el procesamiento para este aspecto de la instrucción finaliza, PASO 868.

35 Volviendo a CONSULTA 854, si CZAM está instalado, entonces se hace una determinación en cuanto a si la CPU está en el modo solicitado, CONSULTA 870. Si la CPU ya está en el modo solicitado, entonces, en un ejemplo, se proporciona un estado de que la CPU ya está en el modo arquitectónico solicitado (por ejemplo, z/Architecture), PASO 872. En esta realización, no obstante, este estado es aceptable y no se trata como error, PASO 874. O bien se ignora, o en otra realización preferida de la presente invención, se puede proporcionar un código de condición que sea un código sin error. En realización adicional más, el estado indica meramente una terminación con éxito. También existen otras posibilidades para indicar sin error aún cuando la CPU ya esté en el modo arquitectónico solicitado.

40 Volviendo a la CONSULTA 870, si, no obstante, la CPU no está en el modo solicitado, entonces no se acepta la orden, dado que es ilegal volver al modo arquitectónico único (por ejemplo, ESA/390), PASO 876. Se proporciona el estado, PASO 878, que se considera un error, PASO 880.

45 En una realización preferida de la presente invención, cuando CZAM está en un sistema como facilidad no seleccionable, entonces se puede omitir la CONSULTA 854 y el control puede pasar del PASO 852 directamente al PASO 870. En tal realización, los PASOS 854 a 868 pueden no ser implementados.

En otra realización preferida de la presente invención, cuando se recibe una orden para conmutar al modo de arquitectura actual, la orden puede no ser aceptado y se puede indicar un error en el PASO 874.

50 Otros comportamientos, procesos y/u operaciones que pueden cambiar en base a la instalación de un CAM incluyen:

- (8) Cambios en los bits de la facilidad: Se añade un nuevo bit, por ejemplo, el bit 138, a los bits de la facilidad para indicar la facilidad de Modo Arquitectónico de Configuración de z/Architecture, y el bit 2, que indica si el modo de arquitectónico de z/Architecture está activo, ha de ser establecido en uno (indicando activo).

55 En al menos una realización preferida de la presente invención, la facilidad de CZAM está instalada para LPAR e invitado 1 (invitados de primer nivel - invitados iniciados por un hipervisor (por ejemplo, emitiendo una instrucción

Iniciar Ejecución Interpretativa (SIE)), pero no para el invitado 2 (invitados de segundo nivel - un invitado iniciado por otro invitado (por ejemplo, emitiendo una instrucción SIE).

En al menos una realización preferida de la presente invención, cuando CZAM está instalado y se inicia un invitado 2 de z/Architecture, el invitado se inicia en el modo de z/Architecture según la técnica de la FIG. 6A. No obstante, cuando CZAM está instalado, y se inicia un invitado 2 de ESA/390, se inicia en el modo de ESA/390, según la técnica de la FIG. 4A, dado que no está afectado por CZAM, en esta realización. De este modo, el ordenador central y los invitados de primer nivel se controlan por CZAM, en el que se iniciarán/reiniciarán, etc. en z/Architecture, independientemente de la preferencia del modo arquitectónico (por ejemplo, forzado a estar en z/Architecture, dado que ESA/390 no se soporta), pero los invitados de ESA/390 de segundo nivel no se ven afectados por CZAM y continuarán siendo iniciados/reiniciados, etc. en ESA/390.

Como se describe en la presente memoria, en base a instalar una facilidad de modo arquitectónico de configuración, tal como una facilidad de Modo Arquitectónico de Configuración de z/Architecture, se cambian ciertos procesos, operaciones y/o comportamientos de un entorno informático que está configurado para múltiples modos arquitectónicos. Uno de tales procesos es el proceso de encendido. Aspectos adicionales del procesamiento asociado con un proceso de encendido cuando una facilidad de modo arquitectónico de configuración está instalada se describen con referencia a la FIG. 9.

Con referencia a la FIG. 9, inicialmente se hace una determinación en cuanto a si una facilidad de modo arquitectónico de configuración está instalada en un entorno informático configurado para una pluralidad de modos arquitectónicos y tiene una secuencia de encendido definida para encender el entorno informático en un modo arquitectónico (por ejemplo, un modo heredado, tal como ESA/390), PASO 900. El modo arquitectónico único que incluye una primera arquitectura de conjunto de instrucciones y que tiene un primer conjunto de características soportadas, tales como direccionamiento de 31 bits, uso de registros de propósito general de 32 bits y diversas características. Si se determina que la facilidad de modo arquitectónico de configuración no está instalada, CONSULTA 902, entonces se realiza la secuencia de encendido actual, PASO 904, como se describe con referencia a las FIG. 4A-4B. De otro modo, el entorno informático se reconfigura para restringir el uso del modo arquitectónico único (por ejemplo, el modo de ESA/390 heredado), PASO 906. La reconfiguración incluye, por ejemplo, seleccionar una secuencia de encendido diferente para encender el entorno informático en otro modo arquitectónico (por ejemplo, una versión posterior o mejorada del modo de arquitectura - por ejemplo, z/Architecture), PASO 908. Otro modo arquitectónico que incluye una segunda arquitectura de conjunto de instrucciones y que tiene un segundo conjunto de características soportadas, tales como direccionamiento de 64 bits, uso de registros de propósito general de 64 bits y diversas facilidades, tales como la traducción dinámica de direcciones, y/u otras facilidades. La secuencia de encendido se ejecuta entonces para encender el entorno informático en el otro modo arquitectónico que restringe el uso del modo arquitectónico único, PASO 910, como se describe, por ejemplo, con referencia a las FIG. 6A-6B. En un ejemplo, esta ejecución incluye cargar la PSW e invertir el bit 12. A partir de entonces, el entorno informático se ejecuta en el otro modo arquitectónico (por ejemplo, z/Architecture), PASO 912.

En una realización adicional, con referencia a la FIG. 10, la reconfiguración incluye la deshabilitación de una o más operaciones que soportan el modo arquitectónico único, incluyendo deshabilitar la operación del conmutador, PASO 1000. Por ejemplo, la instrucción Señalar Procesador se altera para proporcionar un error en base a una solicitud para conmutar de vuelta al modo arquitectónico único, por ejemplo, ESA/390.

Además, uno o más de otros procesos, operaciones y/o comportamientos se cambian para soportar el encendido en el otro modo arquitectónico, en lugar del modo arquitectónico único y se restringe el uso del modo arquitectónico único, PASO 1002. Estos uno o más procesos incluyen, por ejemplo, el comando configurar SCLP de CPU que pone la CPU en el modo arquitectónico de las CPU ya en el estado configurado 1004; la clave de borrado de carga y la clave de carga normal, que son facilidades del operador que establecen el modo de arquitectura como se define en reinicio del borrado o reinicio de la CPU inicial, respectivamente, 1006; la instrucción Señalar Procesador que se cambia para aceptar una conmutación de un modo arquitectónico al mismo modo arquitectónico, de manera que el estado se proporciona y no se trata como error 1008; y los bits de facilidad se añaden a los indicadores de facilidad para indicar la facilidad de modo arquitectónico de configuración 1010.

Como se describe en la presente memoria, otra operación que se ve afectada por la instalación de la facilidad de modo arquitectónico de configuración es la operación de reinicio. Una realización preferida de la presente invención de procesamiento asociado con el reinicio se describe con referencia a la FIG. 11. Inicialmente, un procesador obtiene (por ejemplo, recibe, se proporciona, o de otro modo obtiene) una operación de reinicio, PASO 1100, y la operación de reinicio se realiza para reiniciar el entorno informático al otro modo arquitectónico (por ejemplo, z/Architecture), PASO 1102, como se describe en la presente memoria. Esto incluye, por ejemplo, usar una PSW que esté en el formato apropiado para la arquitectura y establecer el bit 12 en la PSW en cero.

En la presente memoria se describe en detalle una facilidad de modo arquitectónico de configuración que restringe el uso de ciertos aspectos arquitectónicos de una arquitectura soportada por un entorno informático configurado para una pluralidad de arquitecturas. En un ejemplo, se instala una facilidad de modo arquitectónico de configuración, y un entorno informático que soporta múltiples configuraciones arquitectónicas se puede reconfigurar de manera que ya no se soportan más los aspectos de uno de los modos arquitectónicos (por ejemplo, el modo heredado), sino que

otro modo arquitectónico (por ejemplo, un modo arquitectónico mejorado) sigue siendo soportado. Cuando un entorno informático está así configurado, se evita que el entorno informático se reconfigure de nuevo al modo arquitectónico no soportado.

5 En una realización adicional, un entorno informático se configura dinámicamente en un modo arquitectónico seleccionado, tal como z/Architecture. En esta realización, no se puede hacer una comprobación en cuanto a si una facilidad de CZAM está instalada y/o no se puede realizar una orden explícita de Establecer Arquitectura SIGP. Una realización preferida de la presente invención de la lógica para realizar esta configuración se describe con referencia a la FIG. 12.

10 Con referencia a la FIG. 12, en una realización preferida de la presente invención, un procesador configura un entorno informático para realizar operaciones en un modo arquitectónico seleccionado (por ejemplo, z/Architecture), PASO 1200. La configuración incluye, por ejemplo, comenzar la inicialización del entorno informático usando una palabra de estado de programa almacenado, PASO 1202. En un ejemplo, la palabra de estado de programa almacenado tiene un formato de modo arquitectónico diferente del modo arquitectónico seleccionado. De este modo, se hace una determinación de que la palabra de estado de programa almacenado tiene el formato del modo arquitectónico diferente del modo arquitectónico seleccionado, PASO 1204. En base a esa determinación, la palabra de estado de programa almacenado se modifica automáticamente para tener un formato del modo arquitectónico seleccionado, PASO 1206. La modificación de manera automática se realiza en ausencia de una solicitud explícita para conmutar al modo arquitectónico seleccionado. La inicialización del entorno informático usando la palabra de estado de programa modificado se completa entonces para configurar el entorno informático en el modo arquitectónico seleccionado, PASO 1208.

15 En una realización preferida de la presente invención, la facilidad de CZAM se puede usar con una o más de otras facilidades incluyendo, por ejemplo, una facilidad No-DAT y/o una facilidad de arranque de utilidad de control, descritas en las siguientes solicitudes presentadas conjuntamente, comúnmente asignadas tituladas "Managing Processing Associated with Selected Architectural Facilities" de Gainey, et al., (Número de Legajo de IBM: POU920140020US1); y "Common Boot Sequence for Control Utility Able to be Initialized in Multiple Architectures", Michael K. Gschwind, (Número de Legajo de IBM: POU920140019US1), respectivamente, cada una de las cuales por este medio se incorpora por referencia en la presente memoria en su totalidad.

20 Con referencia a la FIG. 13, en un ejemplo, un producto de programa de ordenador 1300 incluye, por ejemplo, uno o más medios de almacenamiento legibles por ordenador no transitorios 1302 para almacenar medios de código de programa, lógica y/o instrucciones legibles por ordenador 1304 en los mismos para proporcionar y facilitar una o más realizaciones.

25 La presente invención puede ser un sistema, un método y/o un producto de programa de ordenador. El producto de programa de ordenador puede incluir un medio (o medios) de almacenamiento legible por ordenador que tiene instrucciones de programa legibles por ordenador en el mismo para hacer que un procesador lleve a cabo los aspectos de la presente invención.

30 El medio de almacenamiento legible por ordenador puede ser un dispositivo tangible que puede retener y almacenar instrucciones para su uso por un dispositivo de ejecución de instrucciones. El medio de almacenamiento legible por ordenador puede ser, por ejemplo, pero no está limitado a, un dispositivo de almacenamiento electrónico, un dispositivo de almacenamiento magnético, un dispositivo de almacenamiento óptico, un dispositivo de almacenamiento electromagnético, un dispositivo de almacenamiento de semiconductores, o cualquier combinación adecuada de los precedentes. Una lista no exhaustiva de ejemplos más específicos del medio de almacenamiento legible por ordenador incluye los siguientes: 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 borrable programable (EPROM o memoria rápida), una memoria estática de acceso aleatorio (SRAM), una memoria de sólo lectura de disco compacto (CD-ROM) portátil, un disco versátil digital (DVD), una tarjeta de memoria, un disco flexible, un dispositivo codificado mecánicamente tal como tarjetas perforadas o estructuras elevadas en un surco que tiene instrucciones grabadas en el mismo, y cualquier combinación adecuada de los precedentes. Un medio de almacenamiento legible por ordenador, como se usa en la presente memoria, no ha de ser interpretado como que son señales transitorias por sí mismo, tales como ondas de radio u otras ondas electromagnéticas de propagación libre, ondas electromagnéticas que se propagan a través de una guía de ondas u otros medios de transmisión (por ejemplo, pulsos de luz que pasan a través de un cable de fibra óptica), o señales eléctricas transmitidas a través de un cable.

35 Las instrucciones de programa legibles por ordenador descritas en la presente memoria se pueden descargar a los dispositivos informáticos/de procesamiento respectivos desde un medio de almacenamiento legible por ordenador o a un ordenador externo o dispositivo de almacenamiento externo a través de una red, por ejemplo, Internet, una red de área local, una red de área extensa y/o una red inalámbrica. La red puede comprender cables de transmisión de cobre, fibras ópticas de transmisión, transmisión inalámbrica, encaminadores, cortafuegos, conmutadores, ordenadores de pasarela y/o servidores de borde. Una tarjeta adaptadora de red o una interfaz de red en cada dispositivo informático/de procesamiento recibe instrucciones de programa legibles por ordenador de la red y envía

las instrucciones de programa legibles por ordenador para el almacenamiento en un medio de almacenamiento legible por ordenador dentro del dispositivo informático/de procesamiento respectivo.

5 Las instrucciones de programa legibles por ordenador para llevar a cabo las operaciones de la presente invención pueden ser instrucciones de ensamblador, instrucciones de Arquitectura de Conjunto de Instrucciones (ISA),
 10 instrucciones de máquina, instrucciones dependientes de máquina, microcódigo, instrucciones de microprograma, datos de establecimiento de estado, o bien código fuente o bien código de objeto escrito en cualquier combinación de uno o más lenguajes de programación, incluyendo un lenguaje de programación orientado a objetos tal como Smalltalk, C++ o similares, y lenguajes de programación de procedimientos convencionales, tales como el lenguaje de programación "C" o lenguajes de programación similares. Las instrucciones de programa legibles por ordenador
 15 pueden ejecutarse completamente en el ordenador del usuario, parcialmente en el ordenador del usuario, como un paquete de software autónomo, parcialmente en el ordenador del usuario y parcialmente en un ordenador remoto o completamente en el ordenador o servidor remoto. En este último escenario, el ordenador remoto se puede conectar al ordenador del usuario a través de cualquier tipo de red, incluyendo una red de área local (LAN) o una red de área extensa (WAN), o la conexión se puede hacer a un ordenador externo (por ejemplo, a través de Internet usando un Proveedor de Servicios de Internet). En algunas realizaciones, los circuitos electrónicos incluyendo, por ejemplo, circuitos lógicos programables, agrupaciones de puertas programables en campo (FPGA), o agrupaciones de lógica programable (PLA) pueden ejecutar las instrucciones de programa legibles por ordenador utilizando información de estado de las instrucciones de programa legibles por ordenador para personalizar los circuitos electrónicos, con el fin de realizar los aspectos de la presente invención.

20 Los aspectos de la presente invención se describen en la presente memoria con referencia a ilustraciones de diagramas de flujo y/o diagramas de bloques de métodos, aparatos (sistemas) y productos de programas de ordenador según las realizaciones de la invención. Se entenderá que cada bloque de las ilustraciones de diagrama de flujo y/o los diagramas de bloques, y las combinaciones de bloques en las ilustraciones del diagrama de flujo y/o los diagramas de bloques, se pueden implementar mediante instrucciones de programa legibles por ordenador.

25 Estas instrucciones de programa legibles por ordenador se pueden proporcionar a un procesador de un ordenador de propósito general, ordenador de propósito especial u otro aparato de procesamiento de datos programable para producir una máquina, de manera que las instrucciones, que se ejecutan a través del procesador del ordenador u otra aparatos 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 legibles por ordenador también se pueden almacenar en un medio de almacenamiento legible por ordenador que puede dirigir un ordenador, un aparato de procesamiento de datos programable y/u otros dispositivos a funcionar de una manera particular, de manera que el medio de almacenamiento legible por ordenador que tiene instrucciones almacenadas dentro del mismo comprende un artículo de fabricación que incluye instrucciones que implementan aspectos de la función/acto especificado en el diagrama de flujo y/o el bloque o bloques del diagrama de bloques.
 35

Las instrucciones de programa legibles por ordenador también se pueden cargar en un ordenador, otro aparato de procesamiento de datos programable u otro dispositivo para hacer que una serie de pasos operativos se realicen en el ordenador, otro aparato programable u otro dispositivo para producir un proceso implementado por ordenador, de manera que las instrucciones que se ejecutan en el ordenador, otros aparatos programables u otro dispositivo implementen las funciones/actos especificados en el diagrama de flujo y/o bloque o bloque de diagrama de bloques.
 40

El diagrama de flujo y los diagramas de bloques en las Figuras ilustran la arquitectura, la funcionalidad y operación de posibles implementaciones de sistemas, métodos y productos de programas de ordenador según diversas realizaciones de la presente invención. A este respecto, cada bloque en el diagrama de flujo o diagramas de bloques puede representar un módulo, segmento o parte de las instrucciones, que comprende una o más instrucciones ejecutables para implementar la función o las funciones lógicas especificadas. En algunas implementaciones alternativas, las funciones señaladas en el bloque pueden ocurrir fuera del orden señalado en las figuras. Por ejemplo, dos bloques mostrados en sucesión, de hecho, se pueden ejecutar de manera sustancialmente concurrente, o los bloques algunas veces se pueden ejecutar en el orden inverso, dependiendo de la funcionalidad implicada. También se observará que cada bloque de los diagramas de bloques y/o la ilustración del diagrama de flujo, y las combinaciones de bloques en los diagramas de bloques y/o la ilustración del diagrama de flujo, se pueden implementar por sistemas basados en hardware de propósito especial que realizan las funciones o actos especificados o llevar a cabo combinaciones de hardware de propósito especial e instrucciones de ordenador.
 45
 50

Además de lo anterior, uno o más aspectos se pueden proporcionar, ofrecer, desplegar, gestionar, atender, etc. por un proveedor de servicios que ofrece gestión de entornos de cliente. Por ejemplo, el proveedor de servicios puede crear, mantener, soportar, etc. un código de ordenador y/o una infraestructura de ordenador que realice uno o más aspectos para uno o más clientes. A cambio, el proveedor de servicios puede recibir el pago del cliente bajo una suscripción y/o un acuerdo de pago, como ejemplos. Además o alternativamente, el proveedor de servicios puede recibir el pago de la venta de contenido publicitario a uno o más terceros.
 55

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

5 Como aspecto adicional, se puede desplegar una infraestructura informática que comprenda integrar código legible por ordenador en un sistema informático, en el que el código en combinación con el sistema informático es capaz de realizar una o más realizaciones.

10 Como un aspecto adicional más, se puede proporcionar un proceso para integrar una 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 de ordenador comprende una o más realizaciones. El código en combinación con el sistema informático es capaz de realizar una o más realizaciones.

15 Aunque se han descrito anteriormente varias realizaciones, estas son solamente ejemplos. Por ejemplo, los entornos informáticos de otras arquitecturas se pueden usar para incorporar y usar una o más realizaciones. Además, se pueden usar diferentes instrucciones, formatos de instrucción, campos de instrucción y/o valores de instrucción. Aún más, otros tipos de procesos, operaciones y/o comportamientos se pueden ver afectados por la instalación de un CAM. Son posibles muchas variaciones.

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

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

30 Con referencia a la FIG. 14, se representan componentes representativos de un sistema de Ordenador Central 5000 para implementar una o más realizaciones. El ordenador central 5000 representativo comprende una o más CPU 5001 en comunicación con la memoria de ordenador (es decir, almacenamiento central) 5002, así como interfaces de I/O para almacenar dispositivos de medios 5011 y redes 5010 para comunicarse con otros ordenadores o SAN y similares. La CPU 5001 es compatible con una arquitectura que tiene un conjunto de instrucciones de arquitectura y una funcionalidad de arquitectura. La CPU 5001 puede tener una traducción de registros de acceso (ART) 5012, que incluye un almacenador temporal de ART anticipada (ALB) 5013, para seleccionar un espacio de direcciones a ser usado por la traducción dinámica de direcciones (DAT) 5003 para transformar direcciones de programas (direcciones virtuales) en direcciones reales de memoria. Una DAT típicamente incluye un almacenador temporal de traducción anticipada (TLB) 5007 para almacenar en caché las traducciones, de modo que los accesos posteriores al bloque de la memoria de ordenador 5002 no requieren el retraso de la traducción de direcciones. Típicamente, una memoria caché 5009 se emplea entre la memoria de ordenador 5002 y el procesador 5001. La memoria caché 5009 puede ser jerárquica teniendo una memoria caché grande disponible para más de una CPU y memorias caché más pequeñas y más rápidas (nivel inferior) entre la memoria caché grande y cada CPU. En algunas implementaciones, las memorias caché de nivel inferior se dividen para proporcionar memorias caché de nivel bajo separadas para la obtención de instrucciones y accesos de datos.

35 40 45 En una realización preferida de la presente invención, una instrucción se obtiene de la memoria 5002 mediante una unidad de búsqueda de instrucciones 5004 a través de una memoria caché 5009. La instrucción se decodifica en una unidad de decodificación de instrucciones 5006 y se despacha (con otras instrucciones en algunas realizaciones) a la unidad o las unidades de ejecución de instrucciones 5008. Típicamente, se emplean varias unidades de ejecución 5008, por ejemplo, una unidad de ejecución aritmética, una unidad de ejecución de punto flotante y una unidad de ejecución de instrucciones de ramificación. La instrucción se ejecuta por la unidad de ejecución, accediendo a los operandos desde los registros especificados por la instrucción o la memoria según sea necesario. Si un operando ha de ser accedido (cargado o almacenado) desde la memoria 5002, una unidad de carga/almacenamiento 5005 típicamente maneja el acceso bajo el control de la instrucción que se ejecuta. Las instrucciones se pueden ejecutar en circuitos de hardware o en microcódigo interno (microprograma) o mediante una combinación de ambos.

50 55 Como se ha señalado, un sistema informático incluye información en el almacenamiento local (o principal), así como direccionamiento, protección, y grabación de referencias y cambios. Algunos aspectos del direccionamiento incluyen el formato de direcciones, el concepto de espacios de direcciones, los diversos tipos de direcciones y la manera en

que un tipo de dirección se traduce a otro tipo de dirección. Algo del almacenamiento principal incluye ubicaciones de almacenamiento asignadas permanentemente. El almacenamiento principal dota al sistema con un almacenamiento de datos de acceso rápido directamente direccionable. Tanto los datos como los programas han de ser cargados en el almacenamiento principal (desde los dispositivos de entrada) antes de que se puedan procesar.

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

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

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

- 20 La información se transmite entre el almacenamiento y una CPU o un subsistema de canal de un byte, o un grupo de bytes, a la vez. A menos que se especifique de otro modo, por ejemplo, en la z/Architecture, un grupo de bytes en el almacenamiento se direcciona por el byte de más a la izquierda del grupo. El número de bytes en el grupo está o bien implícita o bien explícitamente especificado por la operación a ser realizada. Cuando se usa en una operación de CPU, un grupo de bytes se llama campo. Dentro de cada grupo de bytes, por ejemplo, en la z/Architecture, los bits se numeran en una secuencia de izquierda a derecha. En la z/Architecture, los bits de más a la izquierda algunas veces se conocen como los bits de "orden alto" y los bits de más a la derecha como los bits de "orden bajo".
- 25 No obstante, los números de bits no son direcciones de almacenamiento. Solamente se pueden direccionar los bytes. Para operar sobre bits individuales de un byte en el almacenamiento, se accede a todo el byte. Los bits en un byte se numeran de 0 hasta 7, de izquierda a derecha (por ejemplo, en la z/Architecture). Los bits en una dirección se pueden numerar 8-31 o 40-63 para direcciones de 24 bits, o 1-31 o 33-63 para direcciones de 31 bits; se numeran 0-63 para direcciones de 64 bits. En un ejemplo, los bits 8-31 y 1-31 se aplican a las direcciones que están
- 30 en una ubicación (por ejemplo, un registro) que tiene 32 bits de ancho, mientras que los bits 40-63 y 33-63 se aplican a direcciones que están en una ubicación amplia de 64 bits. Dentro de cualquier otro formato de longitud fija de múltiples bytes, los bits que componen el formato se numeran consecutivamente a partir de 0. Con los propósitos de detección de errores, y preferiblemente para corrección, se pueden transmitir uno o más bits de comprobación con cada byte o con un grupo de bytes. Tales bits de comprobación se generan automáticamente por la máquina y no se pueden controlar directamente por el programa. Las capacidades de almacenamiento se expresan en número de bytes. Cuando la longitud de un campo de operando de almacenamiento está implícita 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. Los campos más grandes pueden estar implícitos para algunas instrucciones. Cuando la longitud de un campo de operando de almacenamiento no está implícita sino que se expresa explícitamente, se dice que el campo
- 35 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 pone en el almacenamiento, se sustituyen los contenidos solamente de aquellas ubicaciones de bytes que se incluyen en el campo designado, aún cuando el ancho del camino físico al almacenamiento pueda ser mayor que la longitud del campo que se almacena.

- 40 Ciertas unidades de información han de estar en un límite integral en el almacenamiento. Un límite se llama integral para una unidad de información cuando su dirección de almacenamiento es un múltiplo de la longitud de la unidad en bytes. Se dan nombres especiales a los campos de 2, 4, 8, 16 y 32 bytes en un límite integral. Una media palabra es un grupo de dos bytes consecutivos en un límite de dos bytes y es el bloque de construcción básico de las instrucciones. Una palabra es un grupo de cuatro bytes consecutivos en un límite de cuatro bytes. Una palabra doble
- 45 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. Una palabra óctuple es un grupo de 32 bytes consecutivos en un límite de 32 bytes. Cuando las direcciones de almacenamiento designan medias palabras, palabras, palabras dobles, palabras cuádruples y palabras óctuples, la representación binaria de la dirección contiene uno, dos, tres, cuatro o cinco bits cero de más a la derecha, respectivamente. Las instrucciones han de estar en los límites integrales de dos bytes.
- 50 Los operandos de almacenamiento de la mayoría de las instrucciones no tienen requisitos de alineación de límites.

En los dispositivos que implementan memorias caché separadas para instrucciones y operandos de datos, se puede experimentar un retraso significativo si el programa se almacena en una línea de caché desde la cual las instrucciones se obtienen posteriormente, independientemente de si el almacén altera las instrucciones que se obtienen posteriormente.

5 En un ejemplo, la realización se puede poner en práctica mediante software (algunas veces conocido como código interno bajo licencia, microprograma, microcódigo, milicódigo, picocódigo y similares, cualquiera de los cuales sería coherente con una o más realizaciones). Con referencia a la FIG. 14, se puede acceder al código de programa de software que incorpora uno o más aspectos mediante el procesador 5001 del sistema ordenador central 5000 desde dispositivos de medios de almacenamiento a largo plazo 5011, tales como una unidad de CD-ROM, una unidad de cinta o una unidad de disco duro. El código de programa de software se puede incorporar en cualquiera de una variedad de medios conocidos para su uso con un sistema de procesamiento de datos, tales como un disquete, un disco duro o un CD-ROM. El código se puede distribuir en tales medios, o se puede distribuir a los usuarios desde la memoria de ordenador 5002 o el almacenamiento de un sistema informático sobre una red 5010 a otros sistemas informáticos para su uso por los usuarios de tales otros sistemas.

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

La FIG. 15 ilustra una estación de trabajo representativa o un sistema de hardware de servidor en el que se pueden poner en práctica una o más realizaciones. El sistema 5020 de la FIG. 15 comprende un sistema informático base 5021 representativo, tal como un ordenador personal, una estación de trabajo o un servidor, incluyendo dispositivos periféricos opcionales. El sistema informático base 5021 incluye uno o más procesadores 5026 y un bus empleado para conectar y habilitar la comunicación entre el procesador o los procesadores 5026 y los otros componentes del sistema 5021 según 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 (incluyendo cualquiera de medios magnéticos, CD, DVD y memoria rápida, por ejemplo) o una unidad de cinta, por ejemplo. El sistema 5021 también podría incluir un adaptador de interfaz de usuario, que conecta el microprocesador 5026 a través del bus a uno o más dispositivos de interfaz, tales como un teclado 5024, un ratón 5023, una impresora/escáner 5030 y/u otros dispositivos de interfaz, que pueden ser cualquier dispositivo de interfaz de usuario, tal como una pantalla sensible al tacto, almohadilla de entrada digitalizada, etc. El bus también conecta un dispositivo de visualización 5022, tal como una pantalla o monitor LCD, al microprocesador 5026 a través de un adaptador de visualizador.

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

La FIG. 16 ilustra una red de procesamiento de datos 5040 en la que se pueden poner en práctica una o más realizaciones. La red de procesamiento de datos 5040 puede incluir una pluralidad de redes individuales, tales como una red inalámbrica y una red cableada, cada una de las cuales puede incluir una pluralidad de estaciones de trabajo individuales 5041, 5042, 5043, 5044. Además, como apreciarán los expertos en la técnica, se pueden incluir una o más LAN, donde una LAN puede comprender una pluralidad de estaciones de trabajo inteligentes acopladas a un procesador de ordenador central.

Todavía con referencia a la FIG. 16, las redes también pueden incluir grandes ordenadores o servidores, tales como un ordenador pasarela (cliente servidor 5046) o un servidor de aplicaciones (servidor remoto 5048 que puede acceder a un repositorio de datos y también al que se puede acceder directamente desde una estación de trabajo 5045). Un ordenador pasarela 5046 sirve como punto de entrada a cada red individual. Se necesita una pasarela cuando se conecta un protocolo de interconexión de redes con otro. La pasarela 5046 se puede acoplar preferiblemente a otra red (por ejemplo, Internet 5047) por medio de un enlace de comunicaciones. La pasarela 5046 también se puede acoplar directamente a una o más estaciones de trabajo 5041, 5042, 5043, 5044 usando un enlace de comunicaciones. El ordenador pasarela se puede implementar utilizando un servidor IBM eServer System z disponible en International Business Machines Corporation.

Con referencia concurrentemente a la FIG. 15 y la FIG. 16, se puede acceder al código de programación de software 5031, que puede incorporar uno o más aspectos, por el procesador 5026 del sistema 5020, desde medios de almacenamiento a largo plazo 5027, tales como una unidad de CD-ROM o un disco duro. El código de programación de software se puede incorporar en cualquiera de una variedad de medios conocidos para su uso con un sistema de procesamiento de datos, tal como un disquete, un disco duro o un CD-ROM. El código se puede distribuir en tales medios, o se puede distribuir a los usuarios 5050, 5051 desde la memoria o el almacenamiento de un sistema informático sobre una red a otros sistemas informáticos para su uso por los usuarios de tales otros sistemas.

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

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

Con referencia a la FIG. 17, se representa una realización de procesador ejemplar para el procesador 5026. Típicamente, se emplean uno o más niveles de memoria caché 5053 para almacenar temporalmente bloques de memoria con el fin de mejorar el rendimiento del procesador. La memoria caché 5053 es un almacenador temporal de alta velocidad que contiene líneas de caché de datos de memoria que probablemente han de ser usadas. Las líneas de caché típicas son de 64, 128 o 256 bytes de datos de memoria. A menudo se emplean memorias caché separadas para almacenar en caché instrucciones distintas que para almacenar en caché datos. La coherencia de memoria caché (sincronización de copias de líneas en memoria y las memorias caché) a menudo se proporciona por diversos algoritmos de "fisgoneo" bien conocidos en la técnica. El almacenamiento de memoria principal 5025 de un sistema de procesador a menudo se conoce como memoria caché. En un sistema de procesador que tiene 4 niveles de memoria caché 5053, el almacenamiento principal 5025 algunas veces se conoce como memoria caché de nivel 5 (L5), dado que típicamente es más rápido y solamente contiene una parte del almacenamiento no volátil (DASD, cinta, etc.) que está disponible para un sistema informático. El almacenamiento principal 5025 "almacena en caché" páginas de datos paginados dentro y fuera del almacenamiento principal 5025 por el sistema operativo.

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

Típicamente se emplea una unidad de búsqueda de instrucciones 5055 para obtener instrucciones en nombre del procesador 5026. La unidad de búsqueda o bien obtiene "las siguientes instrucciones secuenciales", las instrucciones objetivo de las instrucciones tomadas de ramificación, o las primeras instrucciones de un programa que sigue a una conmutación de contexto. Las unidades de búsqueda de instrucciones modernas a menudo emplean técnicas de obtención previa para obtener previamente instrucciones de manera especulativa en base a la probabilidad de que se pudieran usar las instrucciones obtenidas previamente. Por ejemplo, una unidad de

búsqueda puede obtener 16 bytes de instrucción que incluye la siguiente instrucción secuencial y bytes adicionales de instrucciones secuenciales adicionales.

Las instrucciones obtenidas se ejecutan entonces por el procesador 5026. En una realización, la instrucción o las instrucciones obtenidas se pasan a una unidad de despacho 5056 de la unidad de búsqueda. La unidad de despacho decodifica la instrucción o las instrucciones y envía información acerca de la instrucción o las instrucciones decodificadas a las unidades 5057, 5058, 5060 apropiadas. Una unidad de ejecución 5057 típicamente recibirá información acerca de las instrucciones aritméticas decodificadas de la unidad de búsqueda de instrucciones 5055 y realizará operaciones aritméticas sobre operandos según el código de operación de la instrucción. Los operandos se proporcionan a la unidad de ejecución 5057, preferiblemente o bien desde la memoria 5025, los registros de arquitectura 5059 o bien desde un campo inmediato de la instrucción que se ejecuta. Los resultados de la ejecución, cuando se almacenan, se almacenan o bien en la memoria 5025, los registros 5059 o bien en otro hardware de máquina (tal como los registros de control, los registros de PSW y similares).

Las direcciones virtuales se transforman en direcciones reales usando la traducción dinámica de direcciones 5062 y, opcionalmente, usando la traducción de registros de acceso 5063.

Un procesador 5026 típicamente tiene una o más unidades 5057, 5058, 5060 para ejecutar la función de la instrucción. Con referencia a la FIG. 18A, una unidad de ejecución 5057 puede comunicarse 5071 con los registros generales de arquitectura 5059, una unidad de decodificación/despacho 5056, una unidad de almacenamiento de carga 5060 y otras unidades de procesador 5065 por medio de la lógica de interfaz 5071. Una unidad de ejecución 5057 puede emplear varios circuitos de registro 5067, 5068, 5069 para contener información sobre la que operará la unidad aritmético lógica (ALU) 5066. La ALU realiza operaciones aritméticas tales como sumar, restar, multiplicar y dividir, así como funciones lógicas tales como AND, OR y OR exclusiva (XOR), rotar y desplazar. Preferiblemente, la ALU soporta operaciones especializadas que son dependientes del diseño. Otros circuitos pueden proporcionar otras facilidades de arquitectura 5072 incluyendo códigos de condición y lógica de soporte de recuperación, por ejemplo. Típicamente, el resultado de una operación ALU se mantiene en un circuito de registro de salida 5070 que puede enviar el resultado a una variedad de otras funciones de procesamiento. Hay muchas disposiciones de unidades de procesador, la presente descripción solamente se pretende que proporcione una comprensión representativa de una realización preferida de la presente invención.

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

La unidad de ejecución 5057 realiza la suma aritmética sobre dos operandos y almacena el resultado en un tercer operando donde el tercer operando puede ser un tercer registro o uno de los dos registros fuente. La unidad de ejecución utiliza preferiblemente una Unidad Aritmético Lógica (ALU) 5066 que es capaz de realizar una variedad de funciones lógicas tales como Desplazar, Rotar, AND, OR y XOR, así como una variedad de funciones algebraicas incluyendo cualquiera de sumar, restar, multiplicar, dividir. Algunas ALU 5066 están diseñadas para operaciones escalares y otras para punto flotante. Los datos pueden ser Big Endian (donde el byte menos significativo está en la dirección de byte más alta) o Little Endian (donde el byte menos significativo está en la dirección de byte más baja) dependiendo de la arquitectura. La z/Architecture de IBM es Big Endian. Los campos de signo pueden ser signo y magnitud, complemento de 1 o complemento de 2 dependiendo de la arquitectura. Un número de complemento de 2 es ventajoso en el sentido que la ALU no necesita diseñar una capacidad de resta dado que o bien un valor negativo o bien un valor positivo en el complemento de 2 solamente requiere una adición dentro de la ALU. Los números se describen comúnmente en forma abreviada, donde un campo de 12 bits define una dirección de un bloque de 4096 bytes y se describe comúnmente como un bloque de 4 Kbytes (Kilobytes), por ejemplo.

Con referencia a la FIG. 18B, la información de instrucción de ramificación para ejecutar una instrucción de ramificación se envía típicamente a una unidad de ramificación 5058 que a menudo emplea un algoritmo de predicción de ramificación tal como una tabla de historial de ramificación 5082 para predecir el resultado de la ramificación antes de que se completen otras operaciones condicionales. El objetivo de la instrucción de ramificación actual se buscará y ejecutará especulativamente antes de que se completen las operaciones condicionales. Cuando se completan las operaciones condicionales, las instrucciones de ramificación ejecutadas especulativamente o bien se completan o bien se descartan en base a las condiciones de la operación condicional y al resultado especulado. Una instrucción de ramificación típica puede probar códigos de condición y ramificarse a una dirección objetivo si los códigos de condición cumplen con el requisito de ramificación de la instrucción de ramificación, una dirección objetivo se puede calcular en base a varios números, incluyendo los encontrados en los campos de registro o en un campo inmediato de la instrucción, por ejemplo. La unidad de ramificación 5058 puede emplear una ALU 5074 que tiene una pluralidad de circuitos de registro de entrada 5075, 5076, 5077 y un circuito de registro de salida 5080. La unidad de ramificación 5058 puede comunicarse 5081 con los registros generales 5059, la unidad de despacho de decodificación 5056 u otros circuitos 5073, por ejemplo.

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

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

Con referencia a la FIG. 18C, un procesador accede al almacenamiento usando una unidad de carga/almacenamiento 5060. La unidad de carga/almacenamiento 5060 puede realizar una operación de carga obteniendo la dirección del operando objetivo en la memoria 5053 y cargando el operando en un registro 5059 u otra ubicación de memoria 5053, o puede realizar una operación de almacén obteniendo la dirección del operando objetivo en la memoria 5053 y almacenando los datos obtenidos de un registro 5059 u otra ubicación de memoria 5053 en la ubicación de operando objetivo en la memoria 5053. La unidad de carga/almacenamiento 5060 puede ser especulativa y puede acceder a la memoria en una secuencia que está fuera de orden en relación con la secuencia de instrucciones, no obstante, la unidad de carga/almacenamiento 5060 ha de mantener la apariencia para programas en que las instrucciones se ejecutaron en orden. Una unidad de carga/almacenamiento 5060 puede comunicarse 5084 con los registros generales 5059, la unidad de decodificación/despacho 5056, la interfaz de memoria caché/memoria 5053 u otros elementos 5083 y comprende diversos circuitos de registro 5086, 5087, 5088 y 5089, las ALU 5085 y la lógica de control 5090 para calcular direcciones de almacenamiento y proporcionar una secuencia entubada para mantener las operaciones en orden. Algunas operaciones pueden estar fuera de orden, pero la unidad de carga/almacenamiento proporciona funcionalidad para hacer que las operaciones fuera de orden le parezcan al programa como si se hubieran realizado en orden, como es bien conocido en la técnica.

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

En el caso donde el procesador es un procesador de un sistema multiprocesador, cada procesador tiene la responsabilidad de mantener los recursos compartidos, tales como I/O, memorias caché, TLB y memoria, entrelazados por coherencia. Típicamente, las tecnologías de “fisgoneo” se utilizarán en el mantenimiento de la coherencia de la memoria caché. En un entorno de fisgoneo, cada línea de caché se puede marcar como que está en uno cualquiera de un estado compartido, un estado exclusivo, un estado cambiado, un estado no válido y similares con el fin de facilitar el intercambio.

Las unidades de I/O 5054 (FIG. 17) dotan al procesador con medios para unirse a dispositivos periféricos incluyendo cinta, discos, impresoras, visualizadores y redes, por ejemplo. Las unidades de I/O a menudo se presentan al programa de ordenador mediante controladores de software. En grandes ordenadores, tales como el System z de

IBM®, los adaptadores de canal y los adaptadores de sistema abierto son unidades de I/O del gran ordenador que proporcionan las comunicaciones entre el sistema operativo y los dispositivos periféricos.

Además, otros tipos de entornos informáticos pueden beneficiarse de uno o más aspectos. Como ejemplo, un entorno puede incluir un emulador (por ejemplo, software u otros mecanismos de emulación), en el que una arquitectura particular (incluyendo, por ejemplo, ejecución de instrucciones, funciones de arquitectura, tales como la traducción de direcciones, y registros de arquitectura) o un subconjunto de la misma se emula (por ejemplo, en un sistema informático nativo que tiene un procesador y una memoria). En tal entorno, una o más funciones de emulación del emulador pueden implementar una o más realizaciones, aún cuando un ordenador que ejecuta el emulador pueda tener una arquitectura diferente de las capacidades que se emulan. Como ejemplo, en el modo de emulación, se decodifica la instrucción u operación específica que se emula, y se construye una función de emulación apropiada para implementar la instrucción u operación individual.

En un entorno de emulación, un ordenador central incluye, por ejemplo, una memoria para almacenar instrucciones y datos; una unidad de búsqueda de instrucciones para obtener instrucciones de la memoria y, opcionalmente, para proporcionar almacenamiento temporal local para la instrucción de búsqueda; una unidad de decodificación de instrucciones para recibir las instrucciones de búsqueda y para determinar el tipo de instrucciones que se han obtenido; y una unidad de ejecución de instrucciones para ejecutar las instrucciones. La ejecución puede incluir cargar datos en un registro desde la memoria; almacenar datos de vuelta en la memoria desde un registro; o realizar algún tipo de operación aritmética o lógica, como se determina por la unidad de decodificación. En un ejemplo, cada unidad se implementa en software. Por ejemplo, las operaciones que se realizan por las unidades se implementan como una o más subrutinas dentro del software de emulador.

Más particularmente, en un gran ordenador, las instrucciones de máquina de arquitectura se usan por programadores, normalmente hoy en día programadores en "C", a menudo por medio de una aplicación de compilador. Estas instrucciones almacenadas en el medio de almacenamiento se pueden ejecutar de manera nativa en un Servidor de IBM® de z/Architecture, o alternativamente en máquinas que ejecutan otras arquitecturas. Se pueden emular en los servidores de grandes ordenadores de IBM® existentes y futuros y en otras máquinas de IBM® (por ejemplo, servidores Power Systems y Servidores System x). Se pueden ejecutar en máquinas que ejecutan Linux en una amplia variedad de máquinas que usan hardware fabricado por IBM®, Intel®, AMD y otros. Además de la ejecución en ese hardware bajo una z/Architecture, se puede usar Linux, así como máquinas que usan emulación de Hercules, UMX o FSI (Fundamental Software, Inc.), donde generalmente la ejecución está en un modo de emulación. En el modo de emulación, el software de emulación se ejecuta por un procesador nativo para emular la arquitectura de un procesador emulado.

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

Se decodifica una instrucción específica que se emula, y se llama a 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 "C", o algún otro método para proporcionar un controlador para el hardware específico como estará dentro de la habilidad de los expertos en la técnica después de entender la descripción de la realización preferida. Diversas patentes de emulación de software y hardware incluyendo, pero no limitadas a, la Patente de EE.UU. N° 5551013, titulada "Multiprocessor for Hardware Emulation", de Beausoleil et al.; y la patente de EE.UU. N° 6009261, titulada "Preprocessing of Stored Target Routines for Emulating Incompatible Instructions on a Target Processor", de Scalzi et al; y la Patente de EE.UU. N° 5574873, titulada "Decoding Guest Instruction to Directly Access Emulation Routines that Emulate the Guest Instructions", de Davidian et al.; y la Patente de EE.UU. N° 6308255, titulada "Symmetrical Multiprocessing Bus and Chipset Used for Coprocessor Support Allowing Non-Native Code to Run in a System", de Gorishek et al.; y la Patente de EE.UU. N° 6463582, titulada "Dynamic Optimizing Object Code Translator for Architecture Emulation and Dynamic Optimizing Object Code Translation Method", de Lethin et al; y la Patente de EE.UU. N° 5790825, titulada "Method for Emulating Guest Instructions on a Host Computer Through Dynamic Recompilation of Host Instructions", de Eric Traut; y

muchas otras, ilustran una variedad de formas conocidas para lograr la emulación de un formato de instrucción diseñado para una máquina diferente para una máquina objetivo disponible para los expertos en la técnica.

En la FIG. 19, se proporciona un ejemplo de un sistema de ordenador central emulado 5092 que emula un sistema de ordenador central 5000' de una arquitectura de ordenador central. En el sistema de ordenador central emulado 5092, el procesador de ordenador central (CPU) 5091 es un procesador de ordenador central emulado (o procesador de ordenador central virtual) y comprende un procesador de emulación 5093 que tiene una arquitectura de conjunto de instrucciones nativas diferente de la del procesador 5091 del ordenador central 5000'. El sistema de ordenador central emulado 5092 tiene una memoria 5094 accesible por el procesador de emulación 5093. En la realización de ejemplo, la memoria 5094 se divide en una parte de memoria de ordenador central 5096 y una parte de rutinas de emulación 5097. La memoria de ordenador central 5096 está disponible para los programas del ordenador central emulado 5092 según la arquitectura del ordenador central. El procesador de emulación 5093 ejecuta instrucciones nativas de un conjunto de instrucciones de arquitectura de una arquitectura distinta de la del procesador emulado 5091, las instrucciones nativas obtenidas de la memoria de rutinas de emulación 5097, y puede acceder a una instrucción de ordenador central para ejecución de un programa en la memoria de ordenador central 5096 empleando una o más instrucciones obtenidas en una secuencia y rutina de acceso/decodificación que puede decodificar la instrucción o las instrucciones del ordenador central a las que se accede para determinar una rutina de ejecución de instrucciones nativas para emular la función de la instrucción de ordenador central a la que se accede. Otras facilidades que se definen para la arquitectura del sistema de ordenador central 5000' se pueden emular por las rutinas de facilidades de arquitectura, incluyendo tales facilidades como registros de propósito general, registros de control, traducción dinámica de direcciones y soporte del subsistema de I/O y memoria caché del procesador, por ejemplo. Las rutinas de emulación también pueden aprovecharse de las funciones disponibles en el procesador de emulación 5093 (tales como registros generales y traducción dinámica de direcciones virtuales) para mejorar el rendimiento de las rutinas de emulación. También se pueden proporcionar hardware especial y motores sin carga para ayudar al procesador 5093 en la emulación de la función del ordenador central 5000'.

En una realización adicional, uno o más aspectos se relacionan con la informática en la nube. Se entiende de antemano que aunque esta descripción incluye una descripción detallada de la informática en la nube, la implementación de las enseñanzas mencionadas en la presente memoria no se limita a un entorno informático en la nube. Más bien, las realizaciones de la presente invención son capaces de ser implementadas junto con cualquier otro tipo de entorno informático conocido ahora o desarrollado más tarde.

La informática en la nube es un modelo de prestación de servicios para permitir un acceso de red cómodo y bajo demanda a un conjunto compartido de recursos informáticos configurables (por ejemplo, redes, ancho de banda de red, servidores, procesamiento, memoria, almacenamiento, aplicaciones, máquinas virtuales y servicios) que se pueden aprovisionar y lanzar rápidamente con un mínimo esfuerzo de gestión o interacción con un proveedor del servicio. Este modelo en la nube puede incluir al menos cinco características, al menos tres modelos de servicio y al menos cuatro modelos de implementación.

Las características son de la siguiente manera:

Autoservicio bajo demanda: un consumidor en la nube puede provisionar capacidades informáticas de manera unilateral, tales como la hora del servidor y el almacenamiento de red, según sea necesario de manera automática, sin requerir interacción humana con el proveedor del servicio.

Amplio acceso a la red: las capacidades están disponibles sobre una red y se accede a ellas a través de mecanismos estándar que fomentan el uso por plataformas heterogéneas de clientes ligeros o pesados (por ejemplo, teléfonos móviles, ordenadores portátiles y PDA).

Agrupación de recursos: los recursos informáticos del proveedor se agrupan para servir a múltiples consumidores usando un modelo de múltiples inquilinos, con diferentes recursos físicos y virtuales asignados dinámicamente y reasignados según la demanda. Hay un sentido de independencia de ubicación en el sentido de que el consumidor generalmente no tiene control ni conocimiento sobre la ubicación exacta de los recursos proporcionados, pero puede ser capaz de especificar la ubicación en un nivel más alto de abstracción (por ejemplo, país, estado o centro de datos).

Elasticidad rápida: las capacidades se pueden aprovisionar rápida y elásticamente, en algunos casos de manera automática, para poner a escala rápidamente y liberar rápidamente para escalar rápidamente. Para el consumidor, las capacidades disponibles para aprovisionamiento a menudo parecen ser ilimitadas y se pueden adquirir en cualquier cantidad en cualquier momento.

Servicio medido: los sistemas en la nube controlan y optimizan automáticamente el uso de recursos aprovechando una capacidad de medición en algún nivel de abstracción apropiado al tipo de servicio (por ejemplo, almacenamiento, procesamiento, ancho de banda y cuentas de usuario activas). El uso de recursos se puede monitorizar, controlar y notificar proporcionando transparencia tanto para el proveedor como para el consumidor del servicio utilizado.

Los modelos de servicio son de la siguiente manera:

Software como Servicio (SaaS): la capacidad que se proporciona al consumidor es para usar las aplicaciones del proveedor que se ejecutan en una infraestructura en la nube. Las aplicaciones son accesibles desde varios dispositivos cliente a través de una interfaz de cliente ligero tal como un navegador web (por ejemplo, correo electrónico basado en web). El consumidor no gestiona ni controla la infraestructura de la nube subyacente, incluyendo la red, los servidores, los sistemas operativos, el almacenamiento o incluso las capacidades de aplicaciones individuales, con la posible excepción de ajustes limitados de configuración de aplicaciones específicas del usuario.

Plataforma como Servicio (PaaS): la capacidad proporcionada al consumidor es para desplegar en la infraestructura en la nube, aplicaciones creadas o adquiridas por el consumidor, creadas usando lenguajes de programación y herramientas soportadas por el proveedor. El consumidor no gestiona o controla la infraestructura en la nube subyacente, incluyendo las redes, los servidores, los sistemas operativos o el almacenamiento, pero tiene control sobre las aplicaciones implementadas y posiblemente las configuraciones del entorno del alojamiento de aplicaciones.

Infraestructura como Servicio (IaaS): la capacidad proporcionada al consumidor es para aprovisionar procesamiento, almacenamiento, redes y otros recursos informáticos fundamentales donde el consumidor es capaz de desplegar y ejecutar software arbitrario, que puede incluir sistemas operativos y aplicaciones. El consumidor no gestiona ni controla la infraestructura en la nube subyacente, pero tiene control sobre los sistemas operativos, el almacenamiento, las aplicaciones desplegadas y, posiblemente, control limitado de los componentes de interconexión de redes seleccionados (por ejemplo, cortafuegos de ordenador central).

Los modelos de despliegue son de la siguiente manera:

Nube privada: la infraestructura en la nube se opera únicamente para una organización. Se puede gestionar por la organización o por un tercero y puede existir en las instalaciones o fuera de las instalaciones.

Nube en comunidad: la infraestructura en la nube es compartida por varias organizaciones y soporta una comunidad específica que tiene intereses compartidos (por ejemplo, misión, requisitos de seguridad, políticas y consideraciones de cumplimiento). Se puede gestionar por las organizaciones o un tercero y puede existir en las instalaciones o fuera de las instalaciones.

Nube pública: la infraestructura en la nube se pone a disposición del público en general o de un gran grupo industrial y es propiedad de una organización que vende servicios en la nube.

Nube híbrida: la infraestructura en la nube es una composición de dos o más nubes (privada, en comunidad o pública) que siguen siendo entidades únicas pero están unidas entre sí por una tecnología estandarizada o propietaria que permite la portabilidad de datos y aplicaciones (por ejemplo, el estallido en la nube para balanceo de carga entre nubes).

Un entorno informático en la nube está orientado a servicio con un enfoque en apatridia, bajo acoplamiento, modularidad e interoperabilidad semántica. En el corazón de la informática en la nube está una infraestructura que comprende una red de nodos interconectados.

Con referencia ahora a la FIG. 20, se muestra un esquema de un ejemplo de un nodo de informática en la nube. El nodo de informática en la nube 6010 es solamente un ejemplo de un nodo de informática en la nube adecuado y no se pretende que sugiera ninguna limitación en cuanto al alcance de uso o funcionalidad de las realizaciones de la invención descritas en la presente memoria. A pesar de todo, el nodo de informática en la nube 6010 es capaz de ser implementado y/o realizar cualquiera de las funcionalidades expuestas anteriormente.

En el nodo de informática en la nube 6010 hay un sistema/servidor informático 6012, que está operativo con otros numerosos entornos o configuraciones de sistemas informáticos de propósito general o de propósito especial. Ejemplos de sistemas, entornos y/o configuraciones informáticos bien conocidos que pueden ser adecuados para su uso con el sistema/servidor informático 6012 incluyen, pero no se limitan a, sistemas informáticos personales, sistemas informáticos de servidor, clientes ligeros, clientes pesados, dispositivos de mano o de ordenador portátil, sistemas multiprocesador, sistemas basados en microprocesadores, decodificadores, electrónica de consumo programable, PC de red, sistemas de miniordenadores, sistemas informáticos de grandes ordenadores y entornos informáticos en la nube distribuidos que incluyen cualquiera de los sistemas o dispositivos anteriores, y similares.

El sistema/servidor informático 6012 se puede describir en el contexto general de instrucciones ejecutables por el sistema informático, tales como módulos de programa, que se ejecutan por un sistema informático. En general, los módulos de programa pueden incluir rutinas, programas, objetos, componentes, lógica, estructuras de datos, etc. que realizan tareas particulares o implementan tipos de datos abstractos particulares. El sistema/servidor informático 6012 se puede poner en práctica en entornos informáticos en la nube distribuidos donde las tareas se realizan por dispositivos de procesamiento remoto que están enlazados a través de una red de comunicaciones. En un entorno informático en la nube distribuido, los módulos de programa se pueden situar tanto en medios de almacenamiento de sistema informático locales como remotos, incluyendo dispositivos de almacenamiento en memoria.

Como se muestra en la FIG. 20, el sistema/servidor informático 6012 en el nodo informático en la nube 6010 se muestra en forma de un dispositivo informático de propósito general. Los componentes del sistema/servidor informático 6012 pueden incluir, pero no se limitan a, uno o más procesadores o unidades de procesamiento 6016, una memoria de sistema 6028 y un bus 6018 que acopla varios componentes del sistema, incluyendo la memoria de sistema 6028, al procesador 6016.

El bus 6018 representa uno o más de cualquiera de varios tipos de estructuras de bus, incluyendo un bus de memoria o un controlador de memoria, un bus periférico, un puerto de gráficos acelerado y un procesador o bus local usando cualquiera de una variedad de arquitecturas de bus. A modo de ejemplo, y no de limitación, tales arquitecturas incluyen el bus de Arquitectura Estándar de la Industria (ISA), el bus de Arquitectura de Micro Canal (MCA), el bus de ISA Mejorado (EISA), el bus local de la Asociación de Estándares de Electrónica de Video (VESA) y el bus de Interconexión de Componentes Periféricos (PCI).

El sistema/servidor informático 6012 incluye típicamente una variedad de medios legibles por sistema informático. Tales medios pueden ser cualquier medio disponible que es accesible por el sistema/servidor informático 6012, e incluye tanto medios volátiles como no volátiles, medios extraíbles y no extraíbles.

La memoria del sistema 6028 puede incluir medios legibles por sistema informático en forma de memoria volátil, tal como la memoria de acceso aleatorio (RAM) 6030 y/o la memoria caché 6032. El sistema/servidor informático 6012 puede incluir además otros medios de almacenamiento de sistemas informáticos volátiles/no volátiles y extraíbles/no extraíbles. A modo de ejemplo solamente, se puede proporcionar el sistema de almacenamiento 6034 para leer y escribir en un medio magnético no volátil y no extraíble (no mostrado y típicamente denominado "disco duro"). Aunque no se muestra, se pueden proporcionar una unidad de disco magnético para leer de y escribir en un disco magnético no volátil y extraíble (por ejemplo, un "disco flexible"), y una unidad de disco óptico para leer de o escribir en un disco óptico no volátil y extraíble, tal como un CD-ROM, DVD-ROM u otros medios ópticos. En tales casos, cada uno se puede conectar al bus 6018 mediante una o más interfaces de medios de datos. Como se representará y describirá además a continuación, la memoria 6028 puede incluir al menos un producto de programa que tiene un conjunto (por ejemplo, al menos uno) de módulos de programa que están configurados para llevar a cabo las funciones de las realizaciones de la invención.

El programa/utilidad 6040, que tiene un conjunto (al menos uno) de módulos de programa 6042, se puede almacenar en la memoria 6028 a modo de ejemplo, y no de limitación, así como un sistema operativo, uno o más programas de aplicaciones, otros módulos de programa, y datos de programa. Cada uno del sistema operativo, uno o más programas de aplicaciones, otros módulos de programa y datos de programa o alguna combinación de los mismos, pueden incluir una implementación de un entorno de interconexión de redes. Los módulos de programa 6042 generalmente llevan a cabo las funciones y/o metodologías de las realizaciones de la invención como se describe en la presente memoria.

El sistema/servidor informático 6012 también puede comunicarse con uno o más dispositivos externos 6014, tales como un teclado, un dispositivo de apuntamiento, una visualizador 6024, etc.; uno o más dispositivos que permiten a un usuario interactuar con el sistema/servidor informático 6012; y/o cualquier dispositivo (por ejemplo, tarjeta de red, módem, etc.) que permita al sistema/servidor informático 6012 comunicarse con uno o más de otros dispositivos informáticos. Tal comunicación puede ocurrir a través de las interfaces de Entrada/Salida (I/O) 6022. Aún así, el sistema/servidor informático 6012 puede comunicarse con una o más redes, tales como una red de área local (LAN), una red de área extensa (WAN) general, y/o una red pública (por ejemplo, Internet) a través del adaptador de red 6020. Como se representa, el adaptador de red 6020 se comunica con los otros componentes del sistema/servidor informático 6012 a través del bus 6018. Se debería entender que, aunque no se muestra, otros componentes de hardware y/o de software se podrían usar junto con el sistema/servidor informático 6012. Ejemplos incluyen, pero no se limitan a: microcódigo, controladores de dispositivo, unidades de procesamiento redundantes, agrupaciones de discos duros externos, sistemas RAID, unidades de cinta y sistemas de almacenamiento de archivos de datos, etc.

Con referencia ahora a la FIG. 21, se representa el entorno informático en la nube 6050 ilustrativo. Como se muestra, el entorno informático en la nube 6050 comprende uno o más nodos informáticos en la nube 6010 con los cuales pueden comunicarse los dispositivos informáticos locales usados por los consumidores en la nube, tales como, por ejemplo, un asistente digital personal (PDA) o un teléfono celular 6054A, un ordenador de sobremesa 6054B, un ordenador portátil 6054C, y/o un sistema informático de automóvil 6054N. Los nodos 6010 pueden comunicarse unos con otros. Se pueden agrupar (no mostrado) física o virtualmente, en una o más redes, tales como nubes privadas, en comunidad, públicas o híbridas, como se ha descrito anteriormente, o una combinación de las mismas. Esto permite que el entorno informático en la nube 6050 ofrezca infraestructura, plataformas y/o software como servicios para los cuales un consumidor en la nube no necesita mantener los recursos en un dispositivo informático local. Se entiende que los tipos de dispositivos informáticos 6054A-N mostrados en la FIG. 21 se pretende que sean solamente ilustrativos y que los nodos informáticos 6010 y el entorno informático en la nube 6050 pueden comunicarse con cualquier tipo de dispositivo informatizado sobre cualquier tipo de red y/o conexión direccionable de red (por ejemplo, usando un navegador web).

Con referencia ahora a la FIG. 22, se muestra un conjunto de capas de abstracción funcional proporcionadas por el entorno informático en la nube 6050 (FIG. 21). Se debería entender por adelantado que los componentes, las capas

y las funciones mostrados en la FIG. 22 se pretende que sean solamente ilustrativos y las realizaciones de la invención no están limitadas a los mismos. Como se representa, se proporcionan las siguientes capas y funciones correspondientes:

5 La capa de hardware y de software 6060 incluye componentes de hardware y de software. Ejemplos de componentes de hardware incluyen grandes ordenadores, en un ejemplo, sistemas IBM® zSeries®, servidores basados en arquitectura RISC (Ordenador de Conjunto de Instrucciones Reducido), en un ejemplo, sistemas IBM pSeries®; sistemas IBM xSeries®; sistemas IBM BladeCenter®, dispositivos de almacenamiento; redes y componentes de interconexión de redes. Ejemplos de componentes de software incluyen software de servidor de aplicaciones de red, en un ejemplo, software de servidor de aplicaciones IBM WebSphere®, y software de base de datos, en un ejemplo, software de base de datos IBM DB2®. IBM, zSeries, pSeries, xSeries, BladeCenter, WebSphere y DB2, z/OS, z/VM, z/Architecture y Processor Resource/Systems Manager son marcas registradas de International Business Machines Corporation registradas en muchas jurisdicciones en todo el mundo. Otros nombres usados en la presente memoria pueden ser marcas comerciales registradas, marcas registradas o nombres de productos de International Business Machines Corporation u otras compañías.

15 La capa de virtualización 6062 proporciona una capa de abstracción a partir de la cual se pueden proporcionar los siguientes ejemplos de entidades virtuales: servidores virtuales; almacenamiento virtual; redes virtuales, incluyendo redes privadas virtuales; aplicaciones virtuales y sistemas operativos; y clientes virtuales.

20 En un ejemplo, la capa de gestión 6064 puede proporcionar las funciones descritas a continuación. El aprovisionamiento de recursos proporciona una obtención dinámica de recursos informáticos y otros recursos que se utilizan para realizar tareas dentro del entorno informático en la nube. Medición y Precios proporcionan un seguimiento de costes a medida que se utilizan recursos dentro del entorno informático en la nube, y la facturación del consumo de estos recursos. En un ejemplo, estos recursos pueden comprender licencias de software de aplicaciones. La seguridad proporciona verificación de identidad para los consumidores y tareas en la nube, así como protección de datos y otros recursos. El portal de usuario proporciona acceso al entorno informático en la nube para consumidores y administradores de sistemas. La gestión de nivel de servicio proporciona asignación y gestión de recursos informáticos en la nube de manera que se cumplan los niveles de servicio requeridos. La planificación y el cumplimiento del Acuerdo de Nivel de Servicio (SLA) proporcionan un acuerdo previo para, y la adquisición de, recursos informáticos en la nube para los cuales se anticipa un requisito futuro según un SLA.

30 La capa de cargas de trabajo 6066 proporciona ejemplos de funcionalidad para lo cual se puede utilizar el entorno informático en la nube. Ejemplos de cargas de trabajo y funciones que se pueden proporcionar desde esta capa incluyen: mapeado y navegación; desarrollo de software y gestión del ciclo de vida; entrega de educación en el aula virtual; procesamiento de analítica de datos; y procesamiento de transacciones.

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

40 Las estructuras, materiales, actos y equivalentes correspondientes de todos los medios o elementos de función más paso en las reivindicaciones a continuación, en su caso, 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 una o más realizaciones se ha presentado con propósitos de ilustración y descripción, pero no se pretende que sea exhaustiva o esté limitada a la forma descrita. Muchas modificaciones y variaciones serán evidentes para los expertos en la técnica. La realización se eligió y describió con el fin de explicar mejor diversos aspectos y la aplicación práctica, y para permitir que otros expertos en la técnica entiendan diversas realizaciones con diversas modificaciones que sean adecuadas para el uso particular contemplado.

REIVINDICACIONES

1. Un método para reconfigurar un entorno informático, dicho método que comprende:

5 determinar, mediante un procesador, que un indicador de instalación de facilidad se establece en un valor predeterminado, el valor predeterminado que indica la instalación de una facilidad de modo arquitectónico de configuración en un entorno informático que está configurado para una pluralidad de modos arquitectónicos y tiene una secuencia de encendido definida que es encender el entorno informático en un modo arquitectónico de la pluralidad de modos arquitectónicos, el modo arquitectónico único que comprende una primera arquitectura de conjunto de instrucciones y que tiene un primer conjunto de características soportadas;

10 en base a la determinación de que el indicador de instalación de facilidad se establece en el valor predeterminado, reconfigurar, mediante el procesador, el entorno informático para restringir el uso del modo arquitectónico único, de manera que ya no se soportan más los aspectos del modo arquitectónico único, la capacidad de usar esos aspectos del modo arquitectónico único se eliminan del entorno informático y se evita que el entorno informático se reconfigure de nuevo al modo arquitectónico único no soportado, en donde la reconfiguración es sensible a la determinación de que el indicador de instalación de facilidad se establece en el valor predeterminado y no en respuesta a una solicitud explícita para reconfigurar y en donde la reconfiguración comprende:

20 seleccionar una secuencia de encendido diferente para encender el entorno informático en otro modo arquitectónico de la pluralidad de modos arquitectónicos, en donde el otro modo arquitectónico es diferente del modo arquitectónico único, y el otro modo arquitectónico comprende una segunda arquitectura de conjunto de instrucciones y que tiene un segundo conjunto de características soportadas; y

25 ejecutar la secuencia de encendido diferente para encender el entorno informático en el otro modo arquitectónico en lugar del modo arquitectónico único que restringe el uso del modo arquitectónico único, en donde la ejecución de la secuencia de encendido diferente comprende crear una nueva palabra de estado de programa para controlar las operaciones del entorno informático en el otro modo arquitectónico, la creación de la nueva palabra de estado de programa que comprende formar la nueva palabra de estado de programa para tener un formato indicado por el otro modo arquitectónico, el formato que comprende expandir un campo de dirección desde un primer tamaño hasta un segundo tamaño, e invertir, durante la formación de la nueva palabra de estado de programa, un indicador de modo arquitectónico en la nueva palabra de estado de programa para indicar el otro modo arquitectónico.

30 2. El método de cualquiera de las reivindicaciones precedentes, el indicador de instalación de facilidad que se establece incondicionalmente o bajo el control de un indicador de configuración.

35 3. El método de cualquiera de las reivindicaciones precedentes, en donde la reconfiguración comprende además deshabilitar dentro del entorno informático una o más operaciones para soportar el modo arquitectónico único, la una o más operaciones que comprenden una operación de conmutación para conmutar desde el otro modo arquitectónico al modo arquitectónico único, en donde se deshabilita una conmutación de vuelta al modo arquitectónico único.

40 4. El método de la reivindicación 3, en donde la deshabilitación comprende alterar el procesamiento de una instrucción de señalar procesador para proporcionar un error en base a una solicitud de conmutar de vuelta al modo arquitectónico único.

5. El método de las reivindicaciones precedentes, en donde el método comprende además realizar un reinicio de al menos un procesador del entorno informático, en donde la realización del reinicio comprende:

reiniciar el entorno informático en el otro modo arquitectónico, el reinicio que comprende establecer un modo arquitectónico del entorno informático en el otro modo arquitectónico; e

45 invertir un indicador de modo arquitectónico en una palabra de estado de programa para indicar el otro modo arquitectónico, la palabra de estado de programa usada para controlar las operaciones del entorno informático.

6. El método de las reivindicaciones precedentes, en donde la reconfiguración comprende cambiar el procesamiento de una operación de señalar procesador, en donde una operación de señalar procesador para establecer un modo arquitectónico del entorno informático al modo arquitectónico en el que está actualmente da como resultado un estado de almacenamiento que indica que el entorno informático está actualmente en el modo arquitectónico, este estado que se trata como aceptable por un emisor de la operación de señalar procesador.

7. El método de las reivindicaciones precedentes, en donde el modo arquitectónico único es un modo heredado y el otro modo arquitectónico es un modo mejorado, y en donde el primer conjunto de características soportadas comprende direccionamiento de 31 bits y uso de registros de propósito general de 32 bits, y el segundo conjunto de características soportadas comprende direccionamiento de 64 bits y uso de registros de propósito general de 64 bits.

8. El método de las reivindicaciones precedentes, en donde el entorno informático es un entorno invitado virtual que tiene un procesador de ordenador central, una primera máquina virtual invitada en el primer nivel de virtualización, y una segunda máquina virtual invitada en el segundo nivel de virtualización, y en donde la reconfiguración se realiza para el procesador de ordenador central y la primera máquina virtual invitada, pero no para la segunda máquina virtual invitada, la segunda máquina virtual invitada que se inicia y se procesa en el modo arquitectónico único.
9. Un sistema informático para reconfigurar un entorno informático, dicho sistema informático que comprende:
una memoria; y
un procesador en comunicaciones con la memoria, en donde el sistema informático está configurado para realizar un método, dicho método que comprende:
determinar, mediante un procesador, que un indicador de instalación de facilidad se establece en un valor predeterminado, el valor predeterminado que indica la instalación de una facilidad de modo arquitectónico de configuración en un entorno informático que está configurado para una pluralidad de modos arquitectónicos y tiene una secuencia de encendido definida que es para encender el entorno informático en un modo arquitectónico único de la pluralidad de modos arquitectónicos, el modo arquitectónico único que comprende una primera arquitectura de conjunto de instrucciones y que tiene un primer conjunto de características soportadas;
en base a la determinación de que el indicador de instalación de facilidad se establece en el valor predeterminado, reconfigurar, mediante el procesador, el entorno informático para restringir el uso del modo arquitectónico único, de manera que ya no se soporten más los aspectos del modo arquitectónico único, la capacidad de usar esos aspectos del modo arquitectónico único se eliminan del entorno informático y se evita que el entorno informático se reconfigure de vuelta al modo arquitectónico único no soportado, en donde la reconfiguración es sensible a determinar que el indicador de instalación de facilidad se establece en el predeterminado y no en respuesta a una solicitud explícita para reconfigurar y en donde la reconfiguración comprende:
seleccionar una secuencia de encendido diferente para encender el entorno informático en otro modo arquitectónico de la pluralidad de modos arquitectónicos, en donde el otro modo arquitectónico es diferente del modo arquitectónico único, y el otro modo arquitectónico comprende una segunda arquitectura de conjunto de instrucciones y que tiene un segundo conjunto de características soportadas; y
ejecutar la secuencia de encendido diferente para encender el entorno informático en el otro modo arquitectónico en lugar del modo arquitectónico único que restringe el uso del modo arquitectónico único, en donde la ejecución de la secuencia de encendido diferente comprende crear una nueva palabra de estado de programa para controlar las operaciones del entorno informático en el otro modo arquitectónico, la creación de la nueva palabra de estado de programa que comprende formar la nueva palabra de estado de programa para tener un formato indicado por el otro modo arquitectónico, el formato que comprende expandir un campo de dirección de un primer tamaño a un segundo tamaño, e invertir, durante la formación de la nueva palabra de estado de programa, un indicador de modo arquitectónico en la nueva palabra de estado de programa para indicar el otro modo arquitectónico.
10. El sistema informático de la reivindicación 9, en donde la reconfiguración comprende además deshabilitar dentro del entorno informático una o más operaciones para soportar el modo arquitectónico único, la una o más operaciones que comprenden una operación de conmutación para conmutar del otro modo arquitectónico al modo arquitectónico único, en donde se deshabilita una conmutación de vuelta al modo arquitectónico único.
11. El sistema informático de la reivindicación 9, en donde la deshabilitación comprende alterar el procesamiento de una instrucción de señalar procesador para proporcionar un error en base a una solicitud para conmutar de vuelta al modo arquitectónico único.
12. El sistema informático de cualquiera de las reivindicaciones 9 a 11, en donde la reconfiguración comprende cambiar el procesamiento de una operación de señalar procesador, en donde una operación de señalar procesador para establecer un modo arquitectónico del entorno informático al modo arquitectónico en el que está actualmente da como resultado un estado de almacenamiento que indica que el entorno informático está actualmente en el modo arquitectónico, este estado que se trata como aceptable por un emisor de la operación de señalar procesador.
13. Un producto de programa de ordenador para reconfigurar un entorno informático, el producto de programa de ordenador que comprende: un medio de almacenamiento legible por ordenador legible por un circuito de procesamiento y almacenar instrucciones para su ejecución por el circuito de procesamiento para realizar un método según cualquiera de las reivindicaciones 1 a 8.
14. Un programa de ordenador almacenado en un medio legible por ordenador y cargable en la memoria interna de un ordenador digital, que comprende partes de código software, cuando dicho programa se ejecuta en un ordenador, para realizar el método de cualquiera de las reivindicaciones 1 a 8.

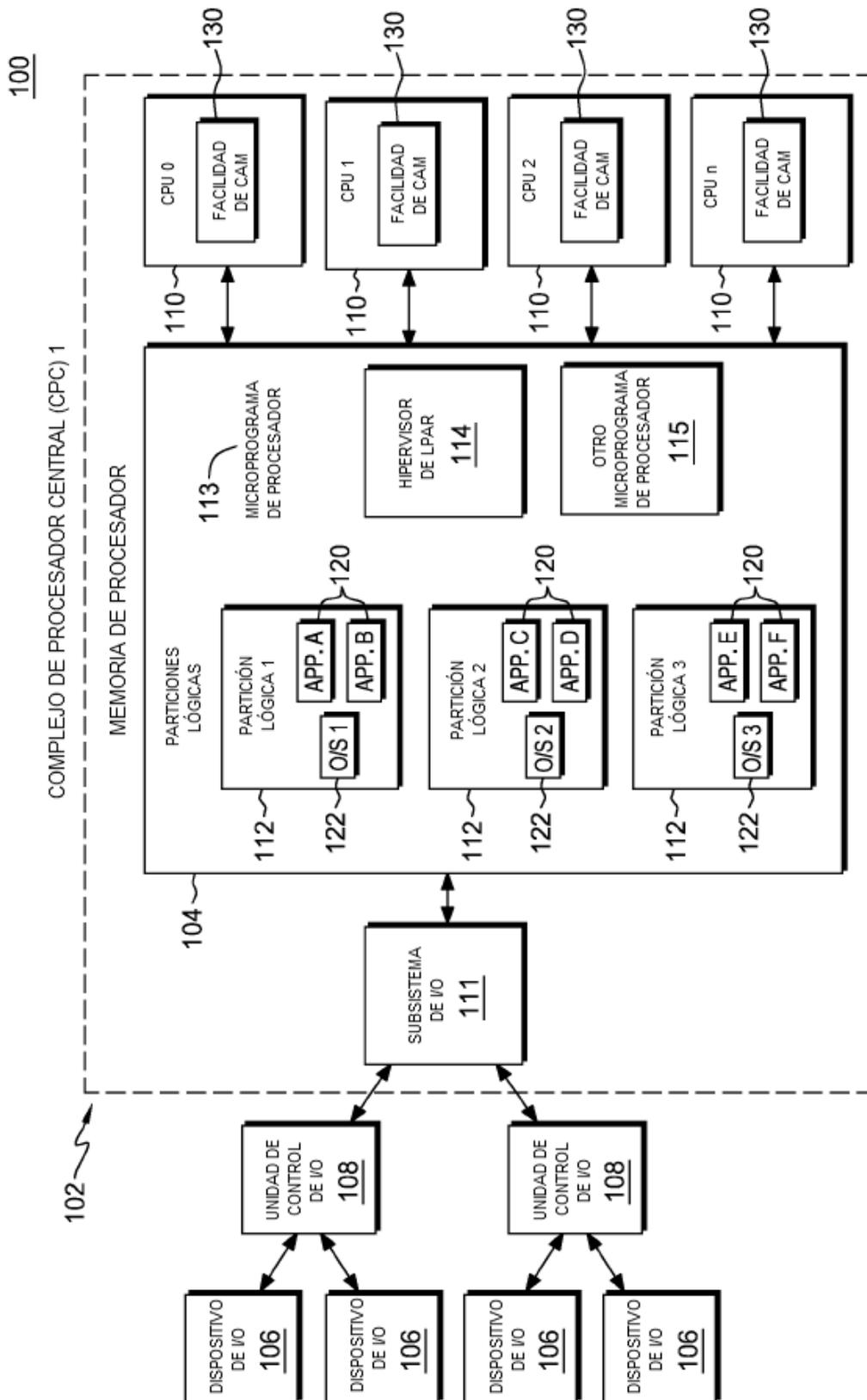


FIG. 1A

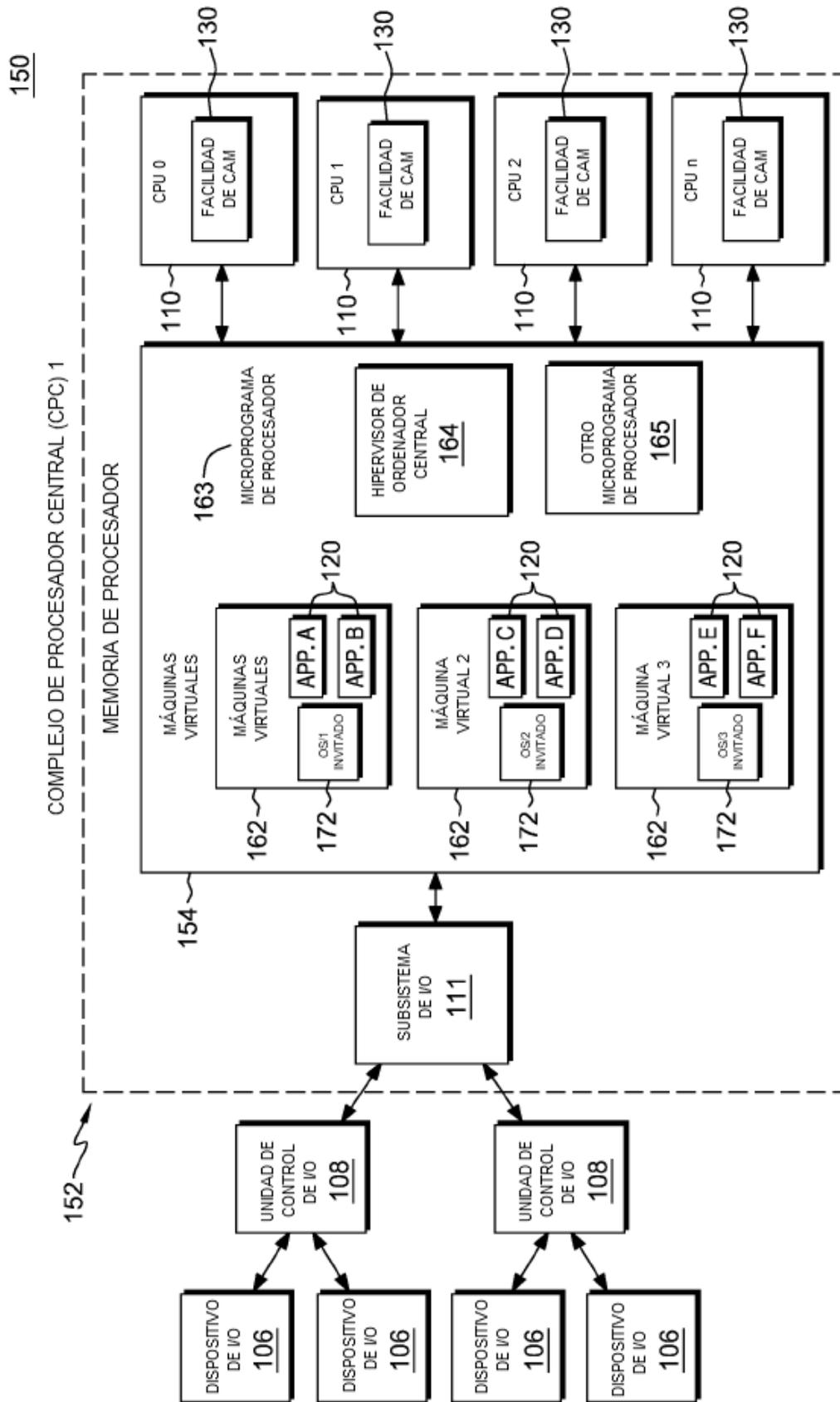


FIG. 1B

200

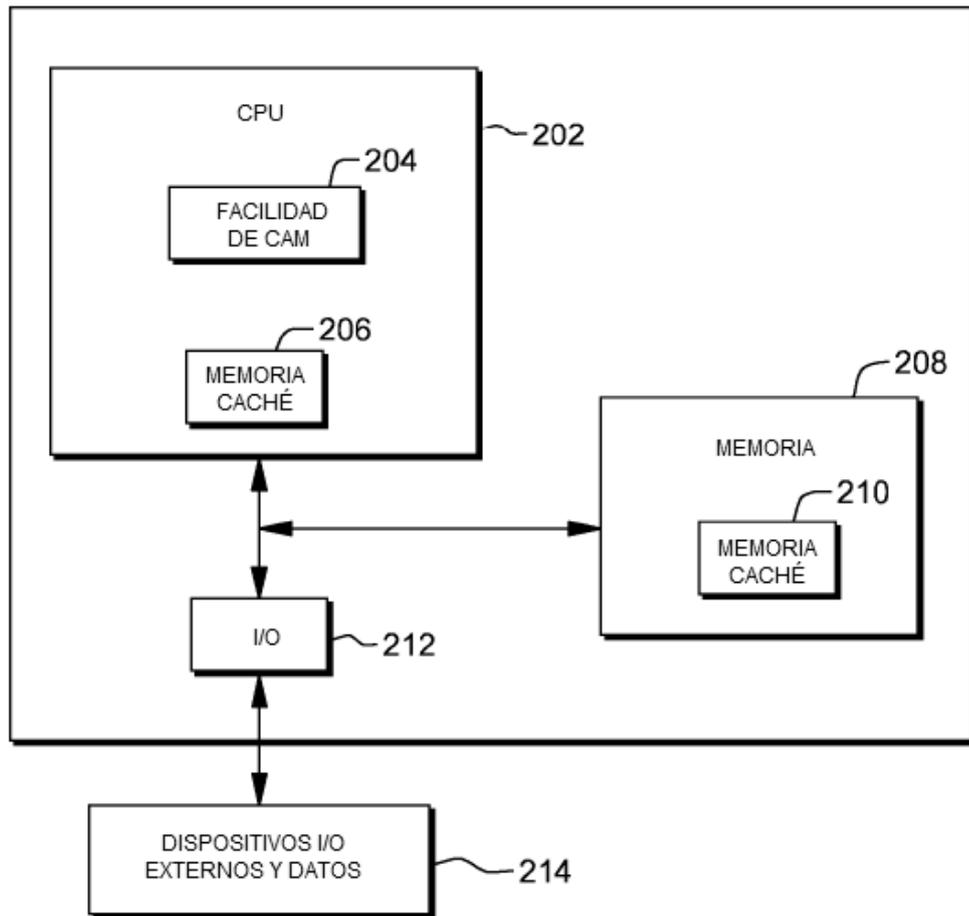


FIG. 2

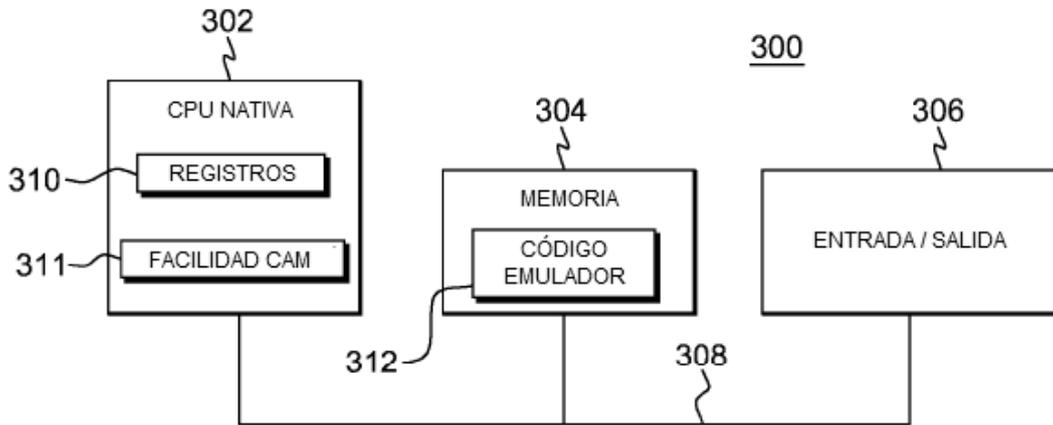


FIG. 3A

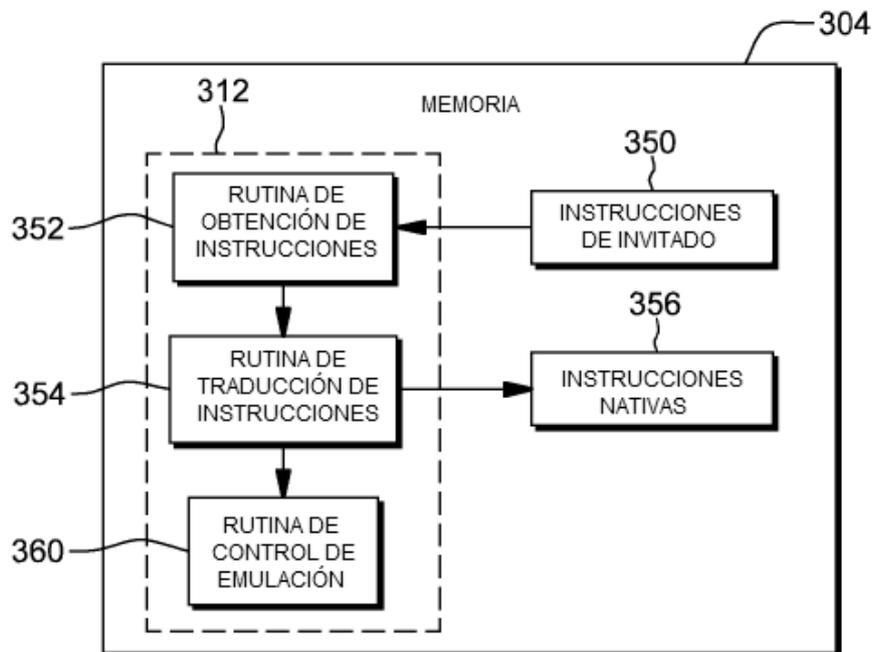


FIG. 3B

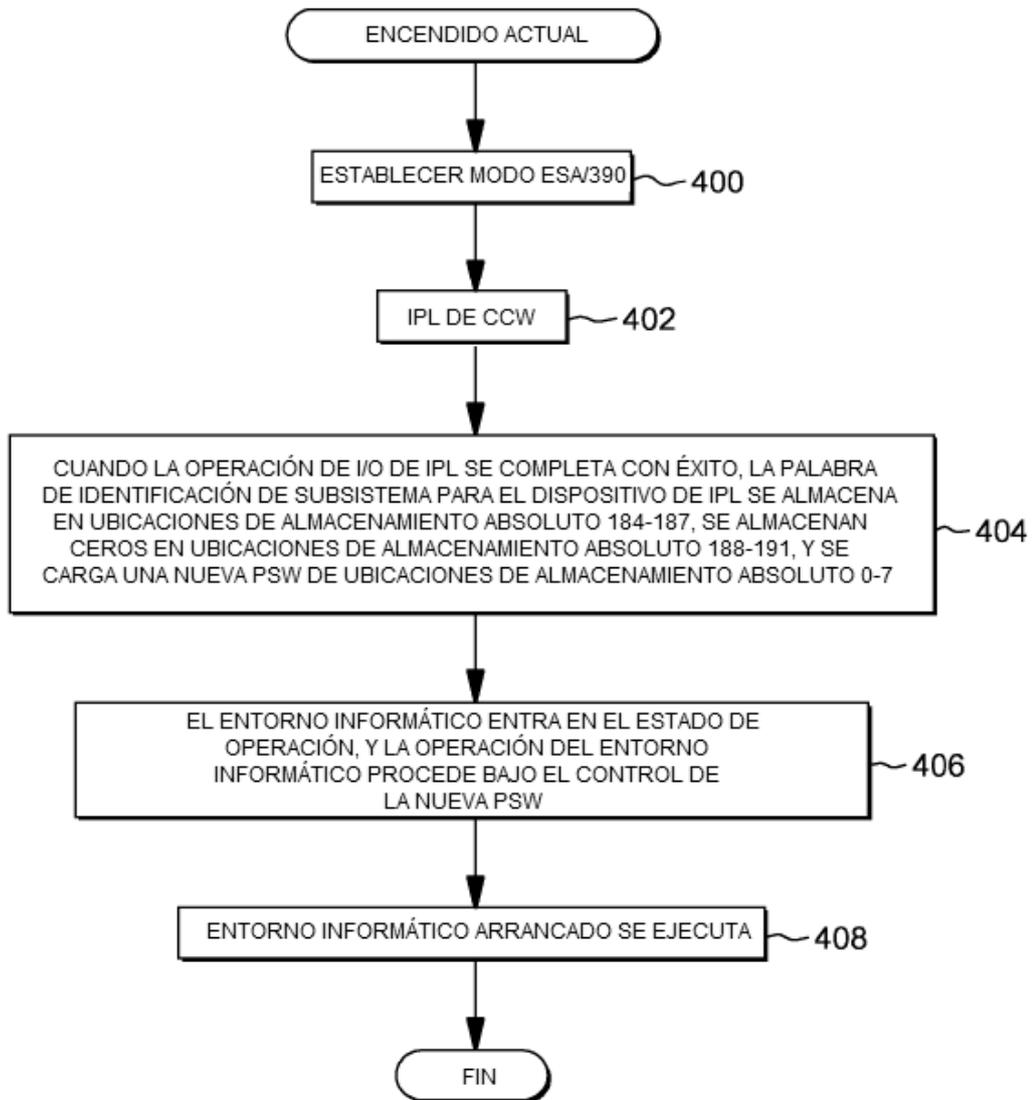


FIG. 4A

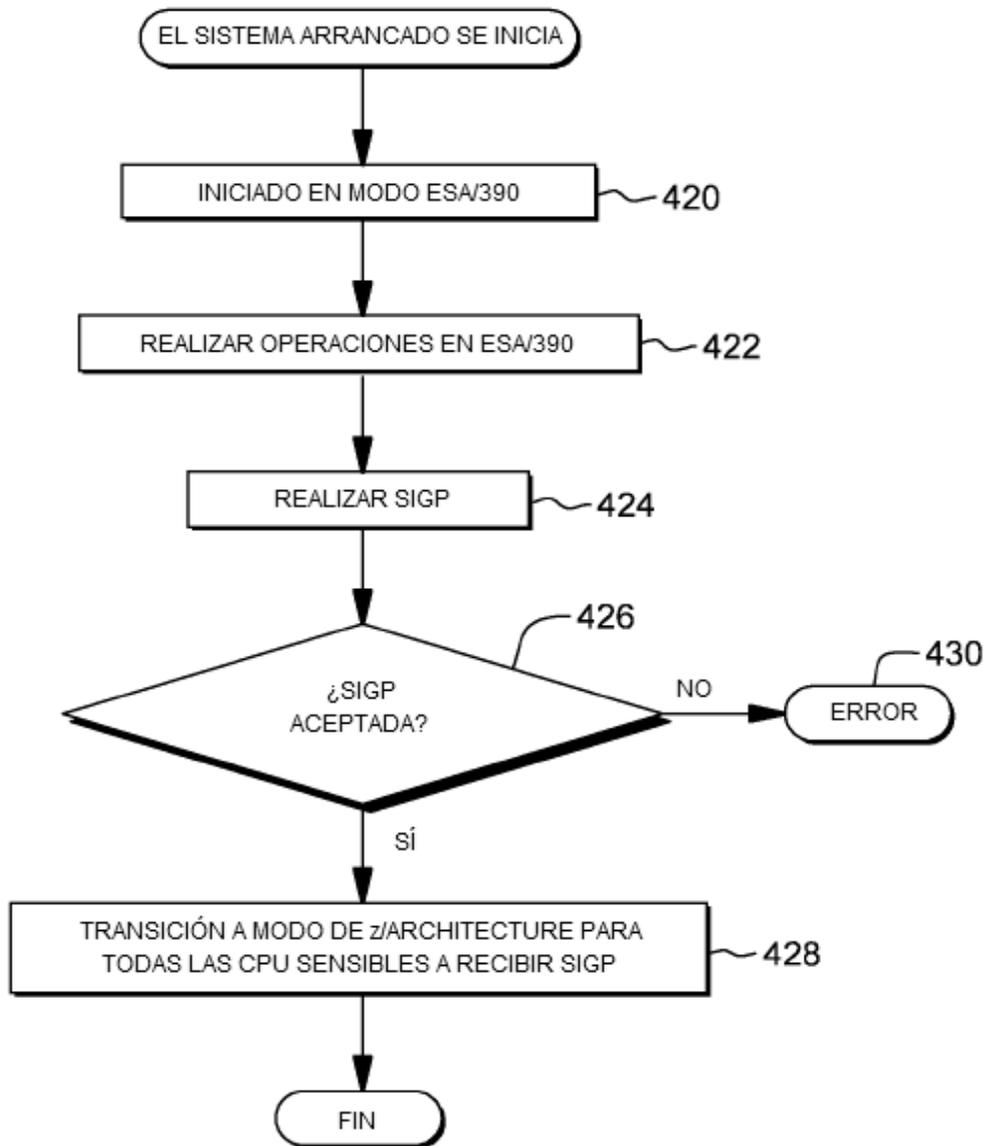


FIG. 4B

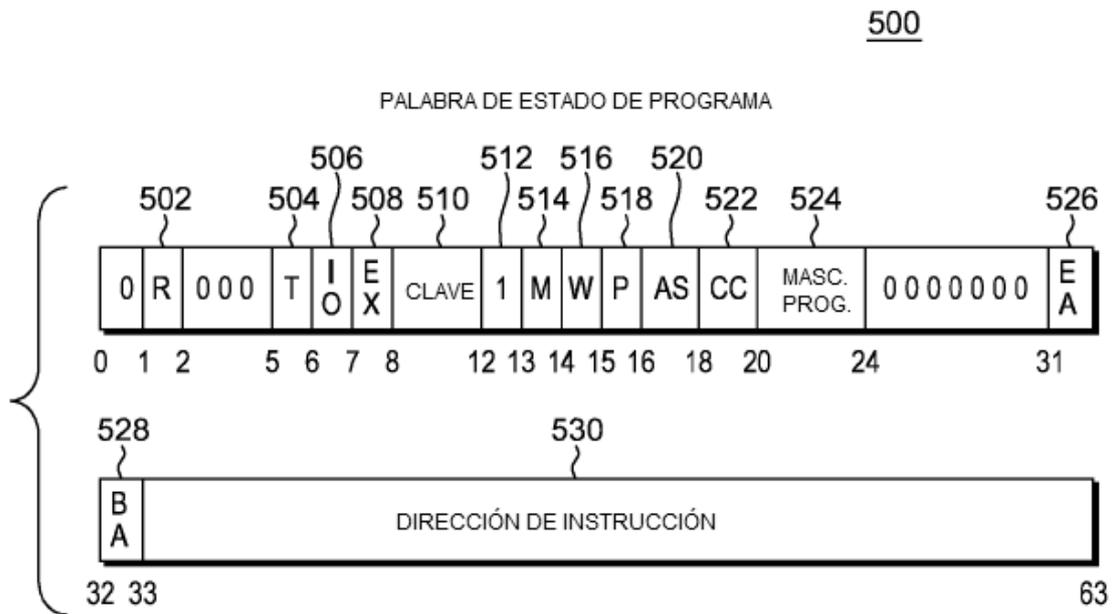


FIG. 5

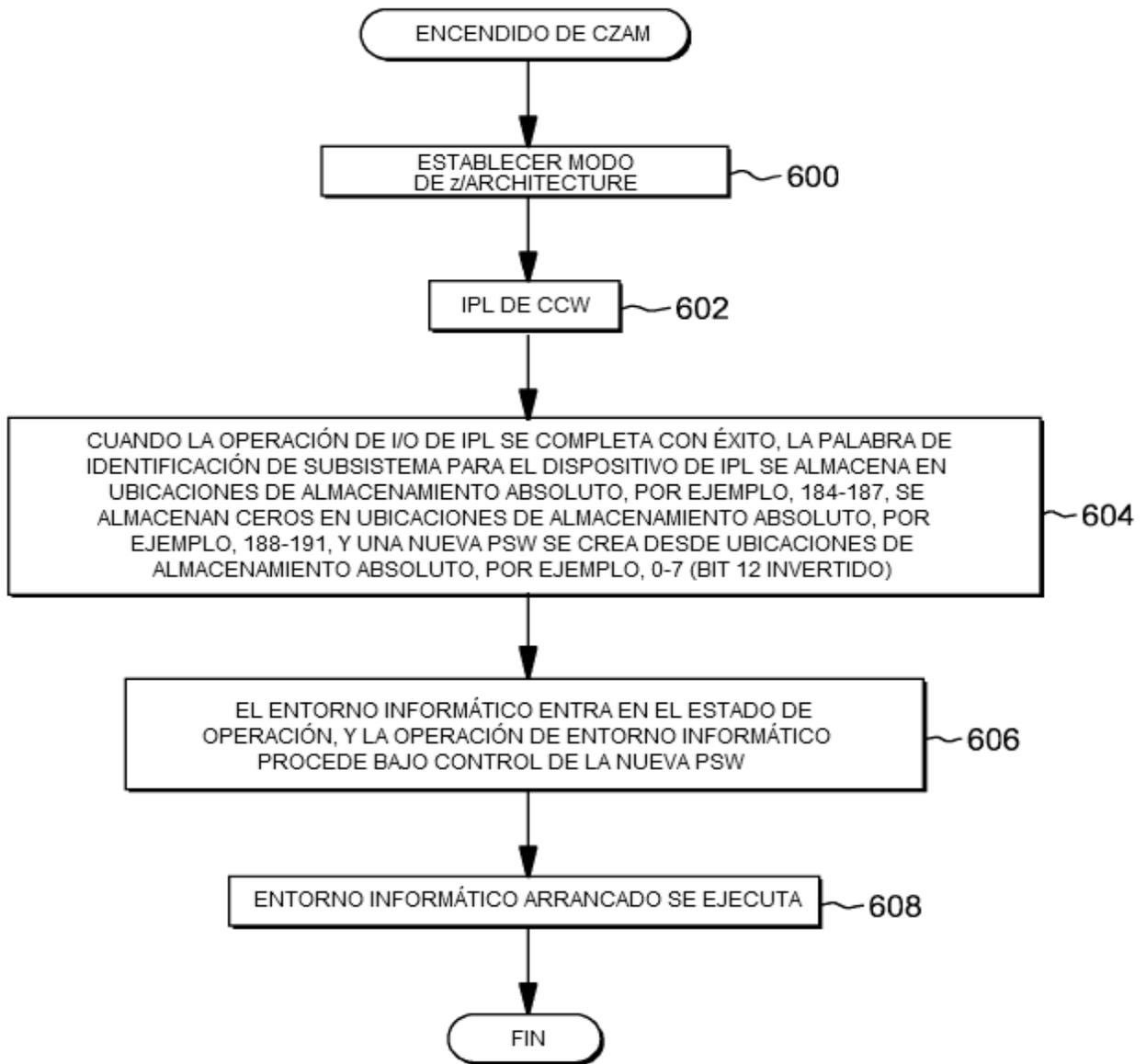


FIG. 6A

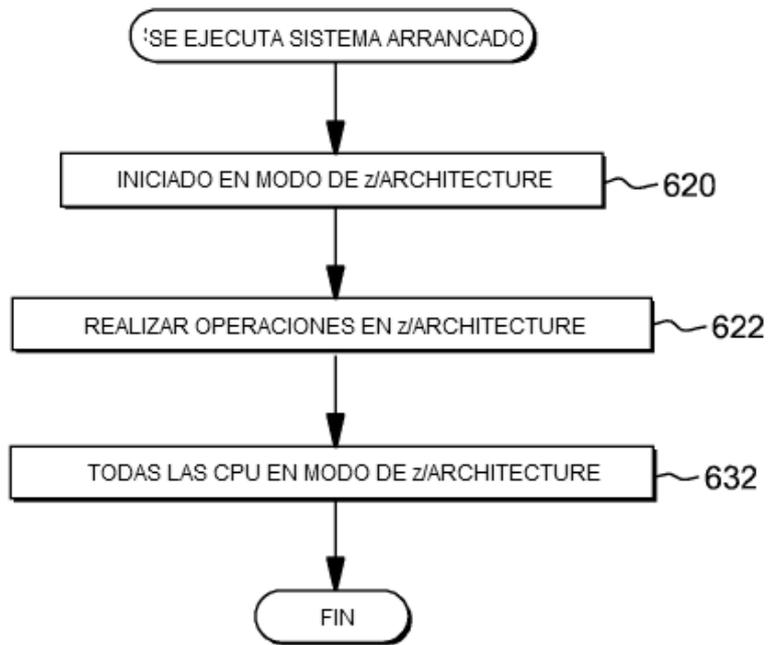


FIG. 6B

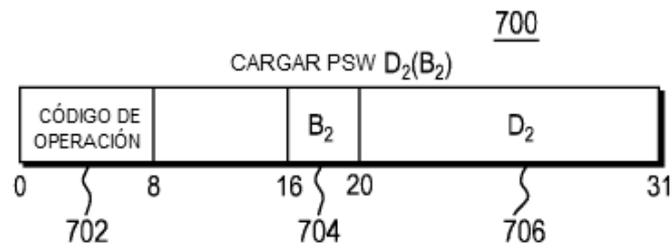


FIG. 7

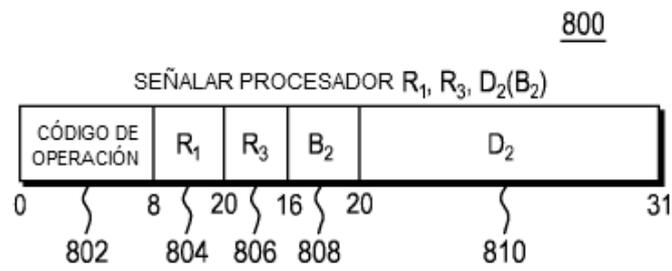


FIG. 8A

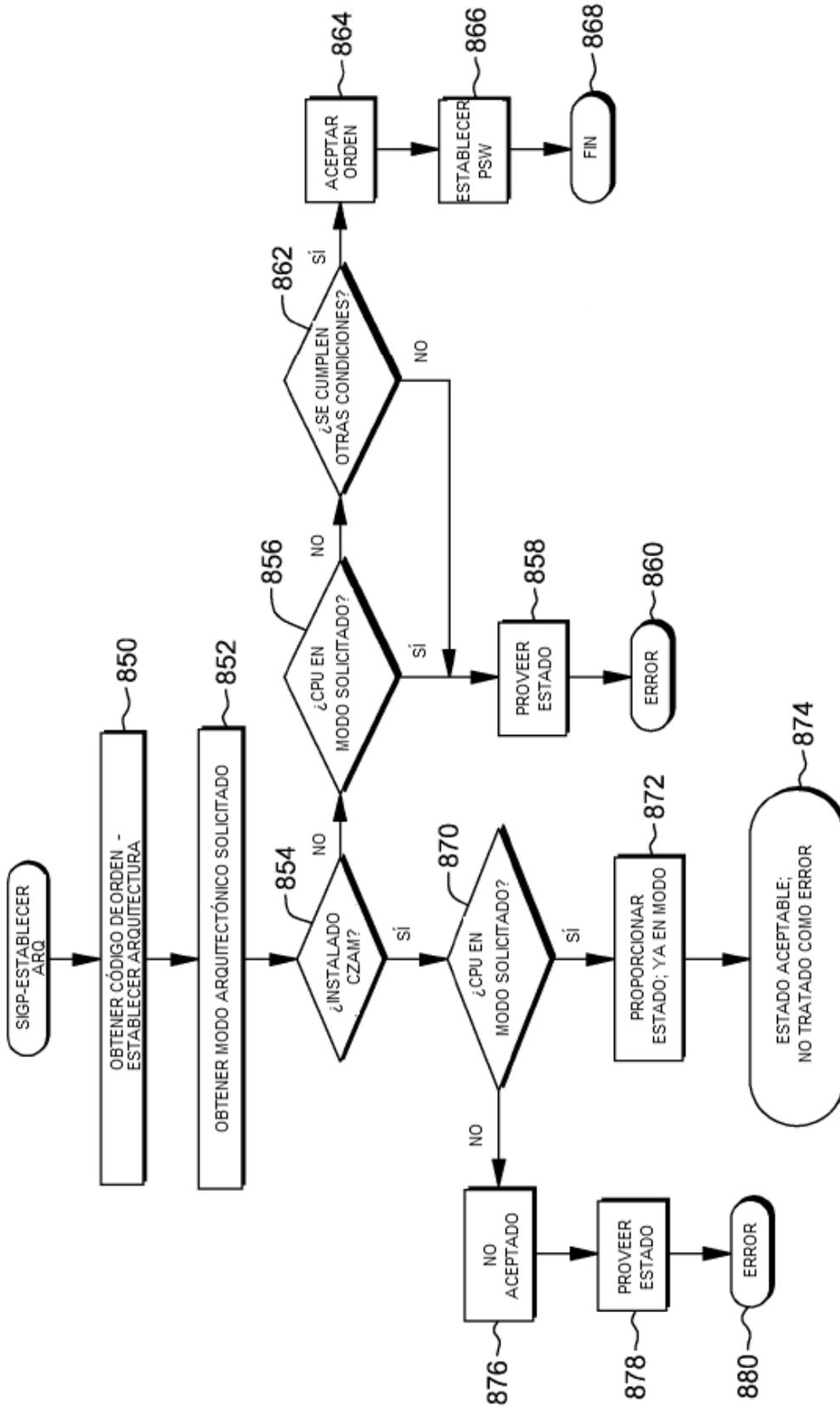


FIG. 8B

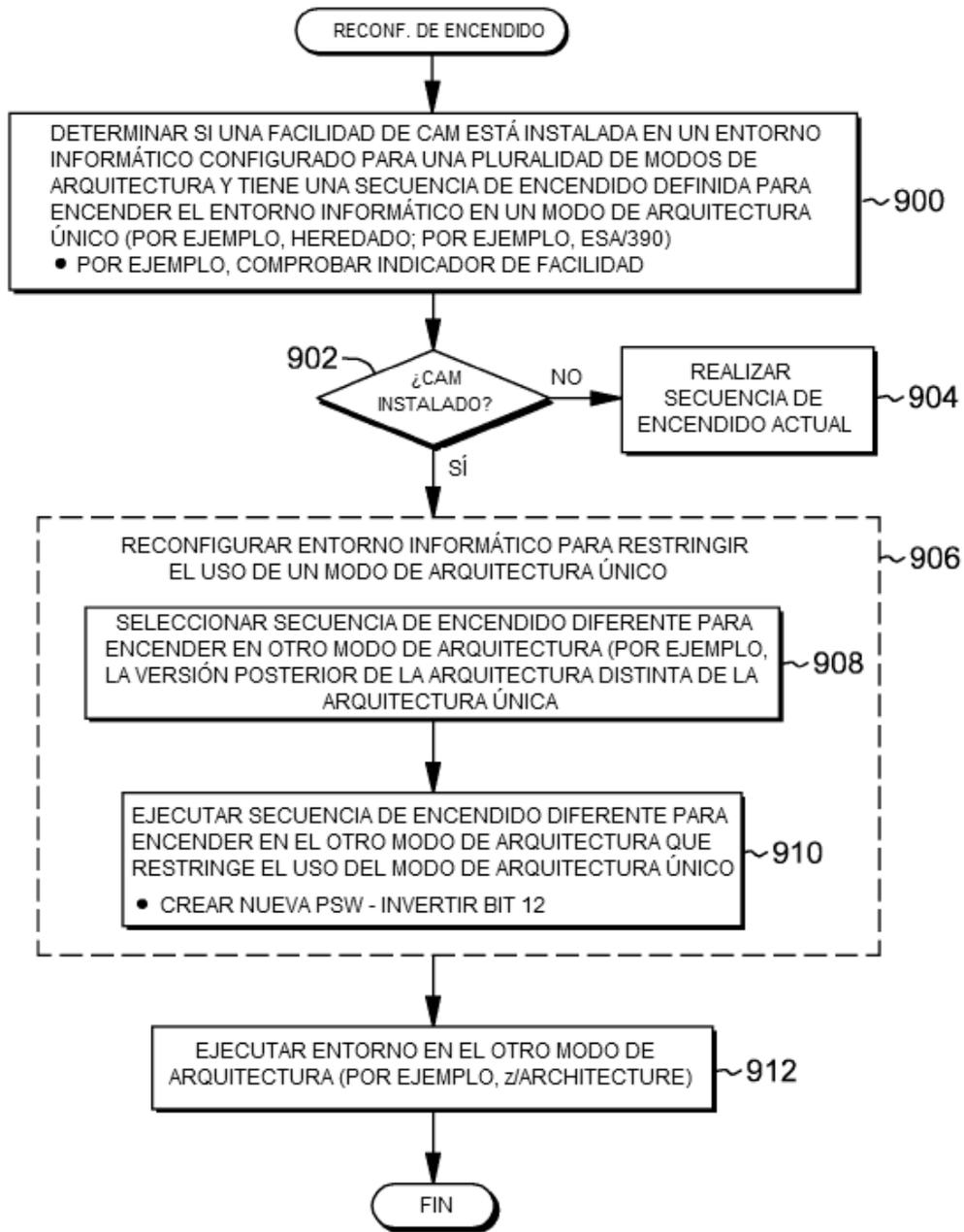


FIG. 9

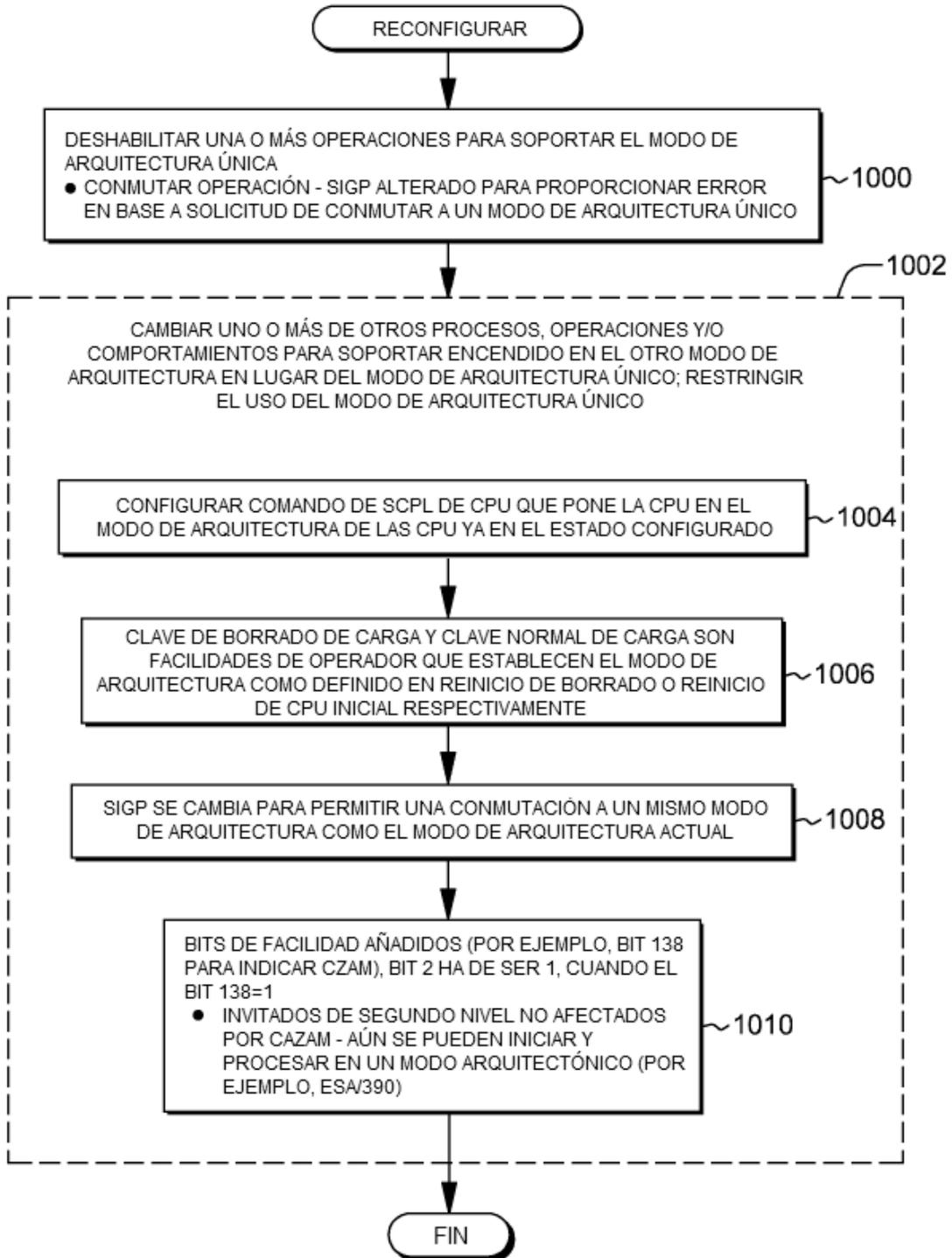


FIG. 10

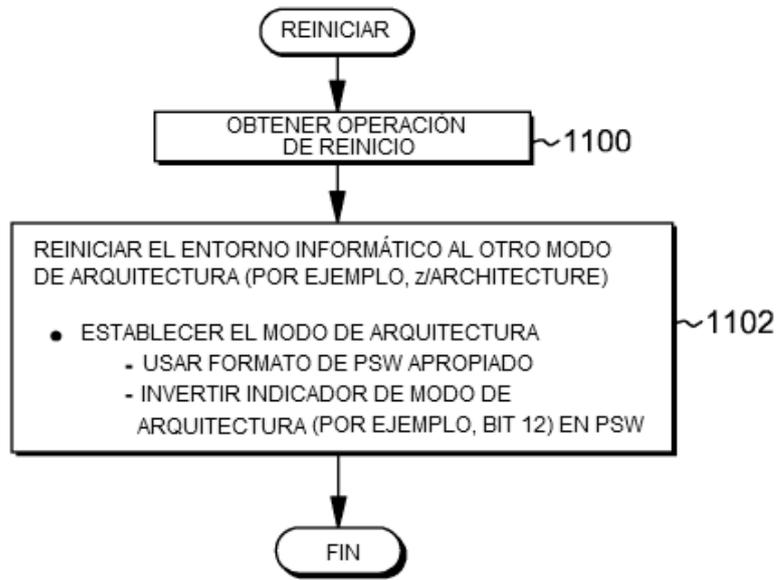


FIG. 11

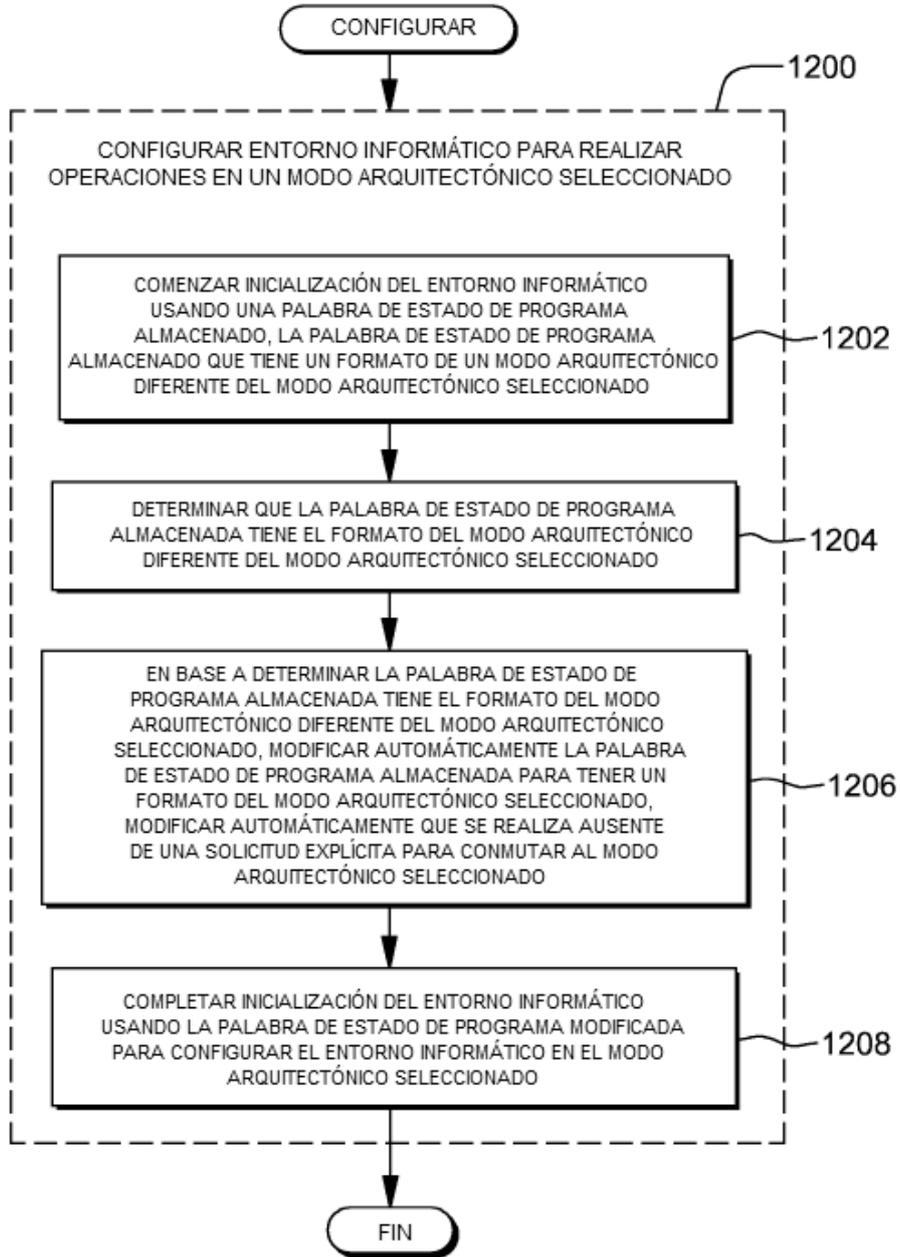


FIG. 12

PRODUCTO DE
PROGRAMA DE
ORDENADOR
1300

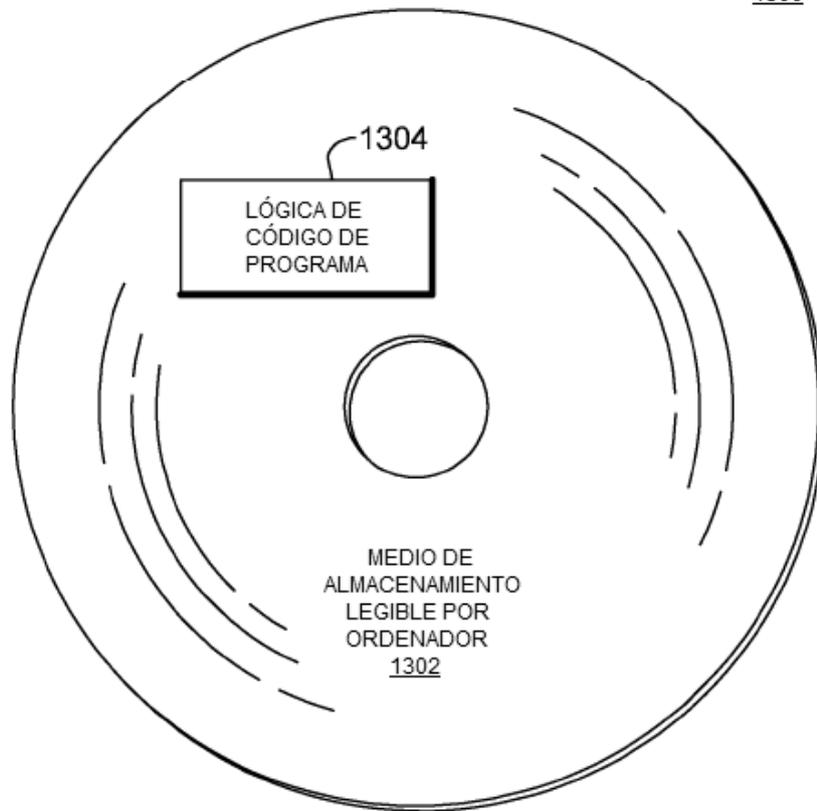


FIG. 13

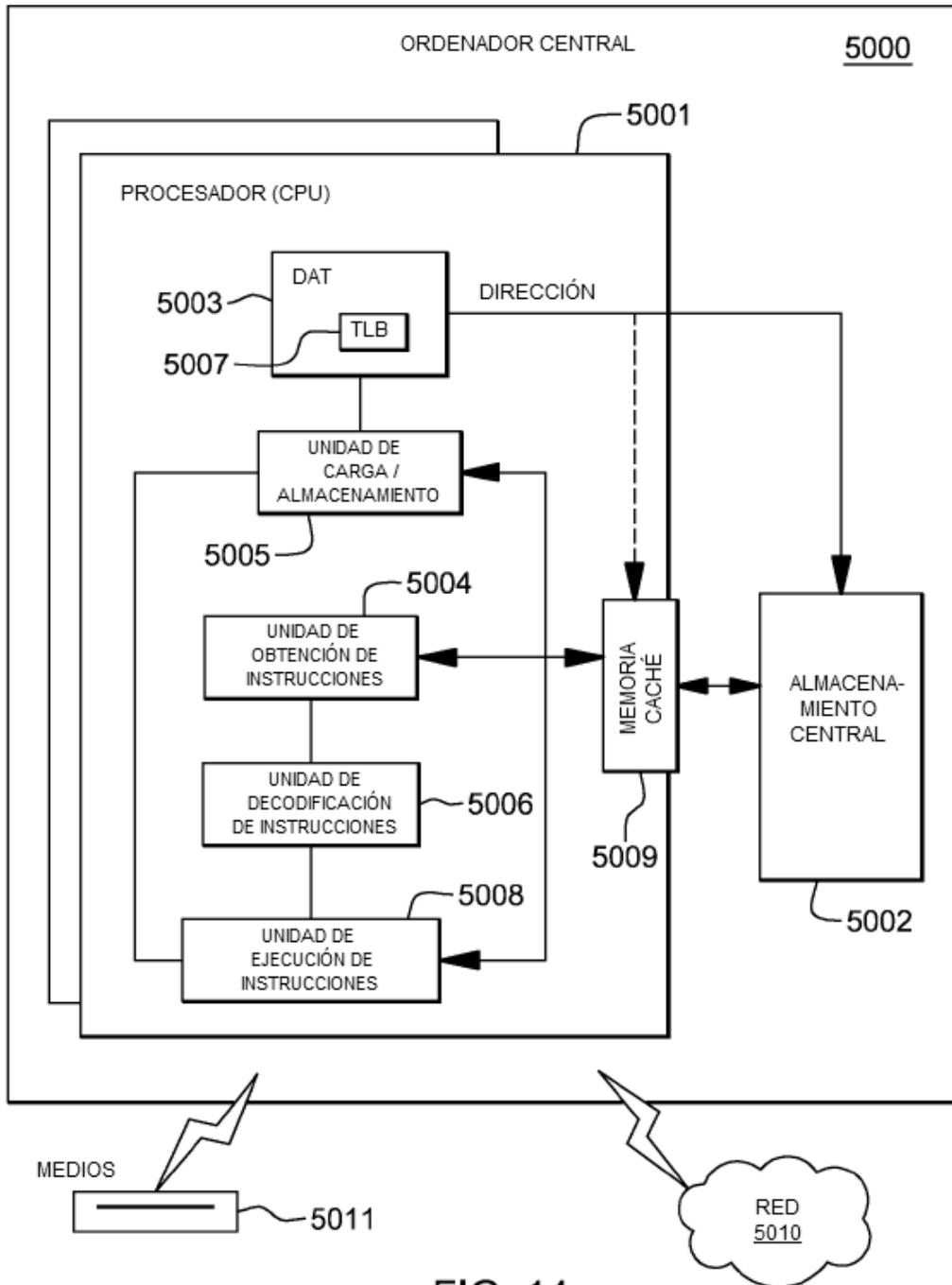


FIG. 14

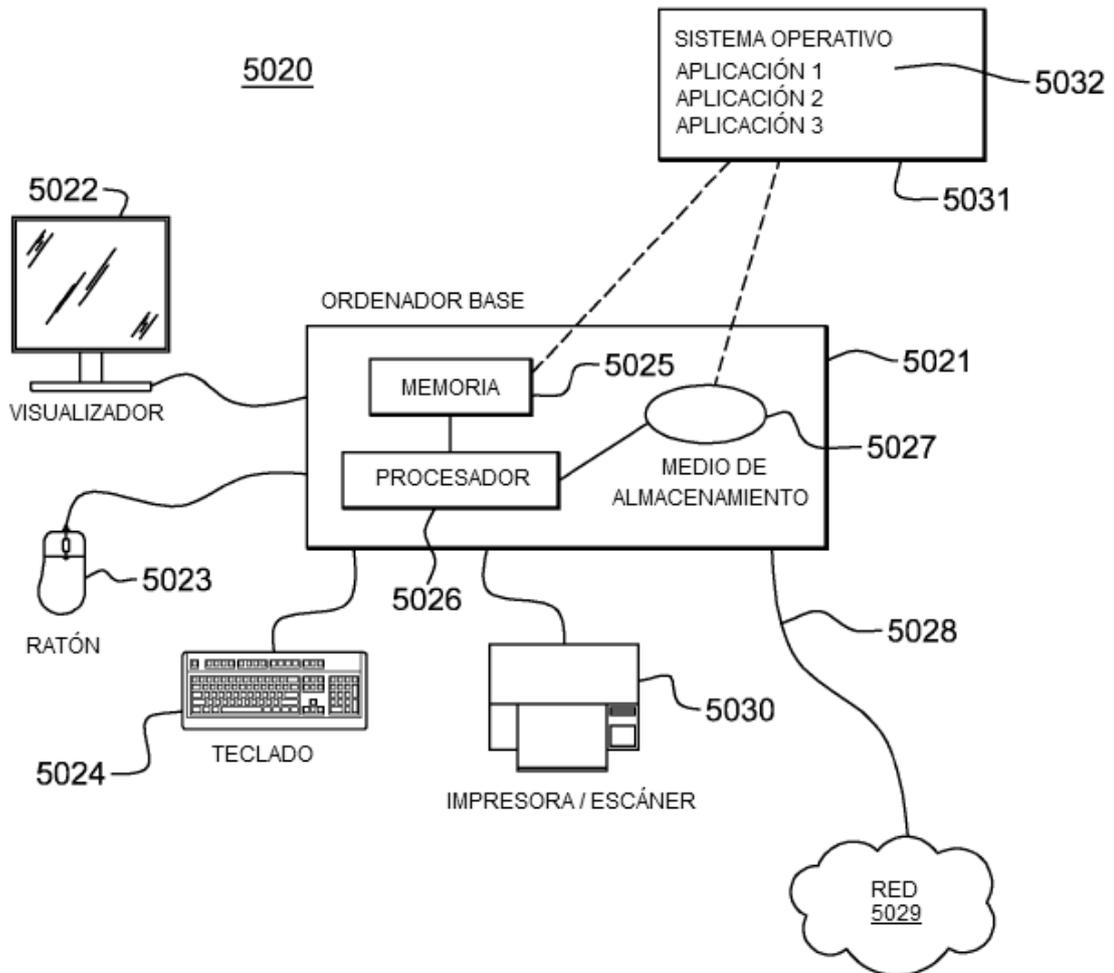


FIG. 15

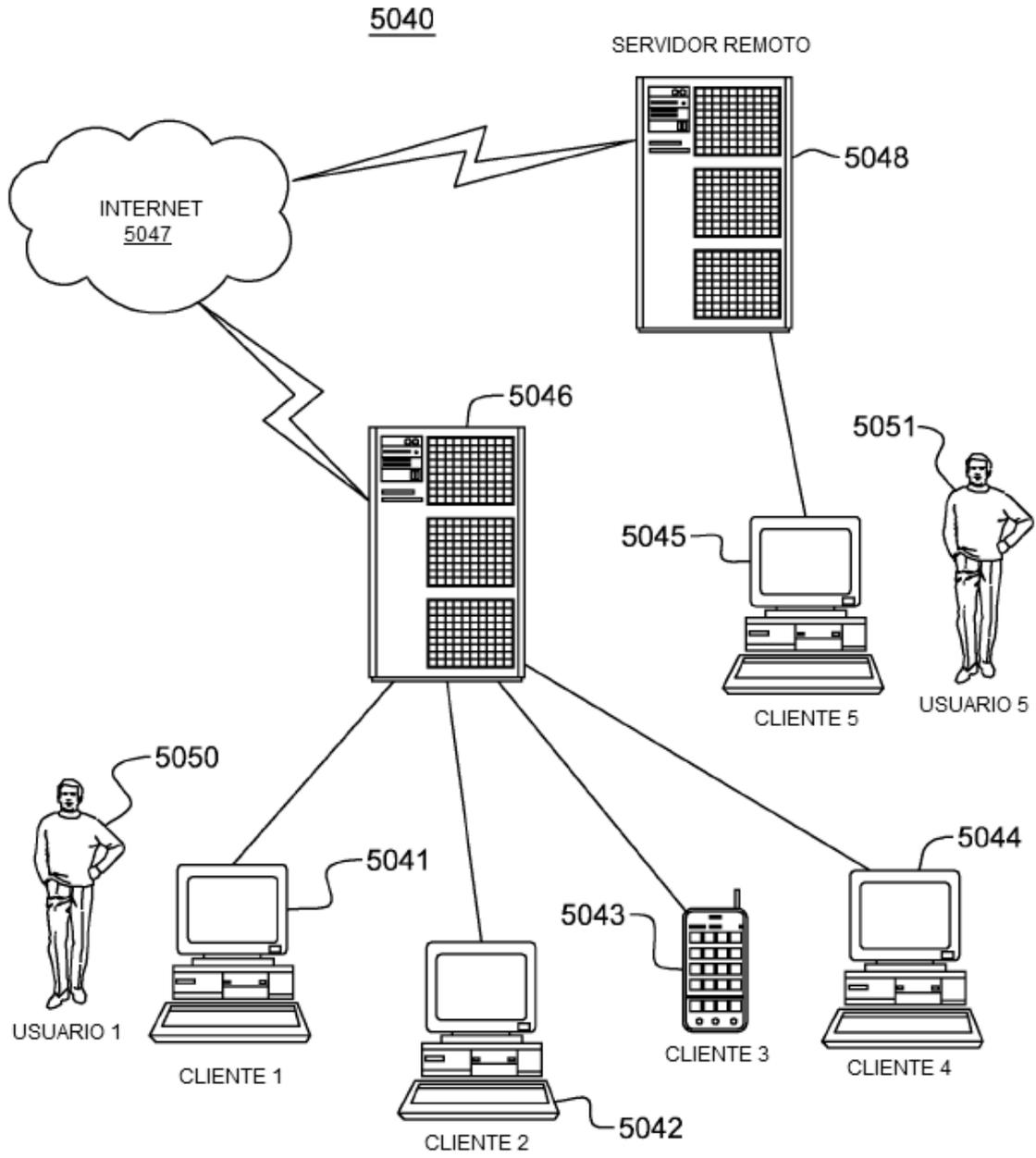


FIG. 16

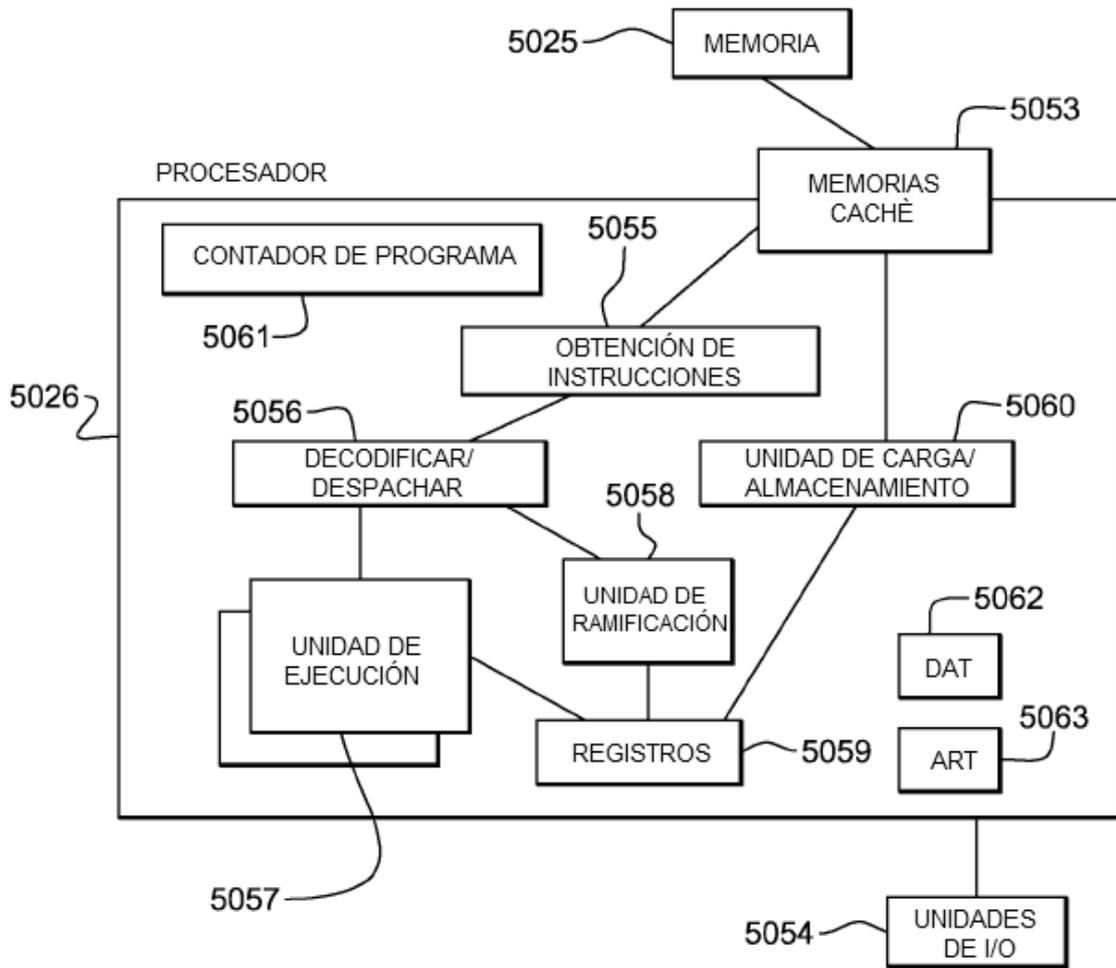


FIG. 17

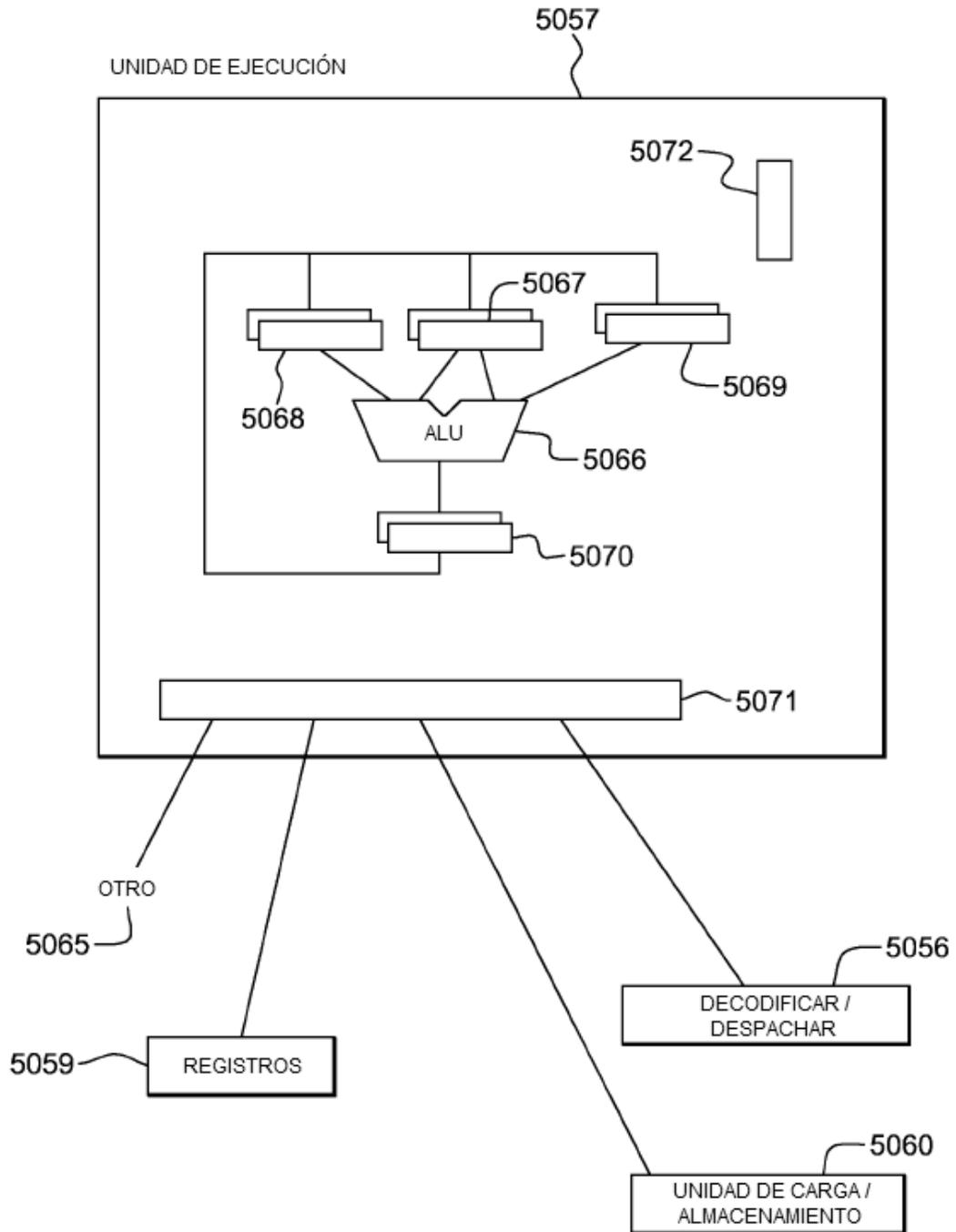


FIG. 18A

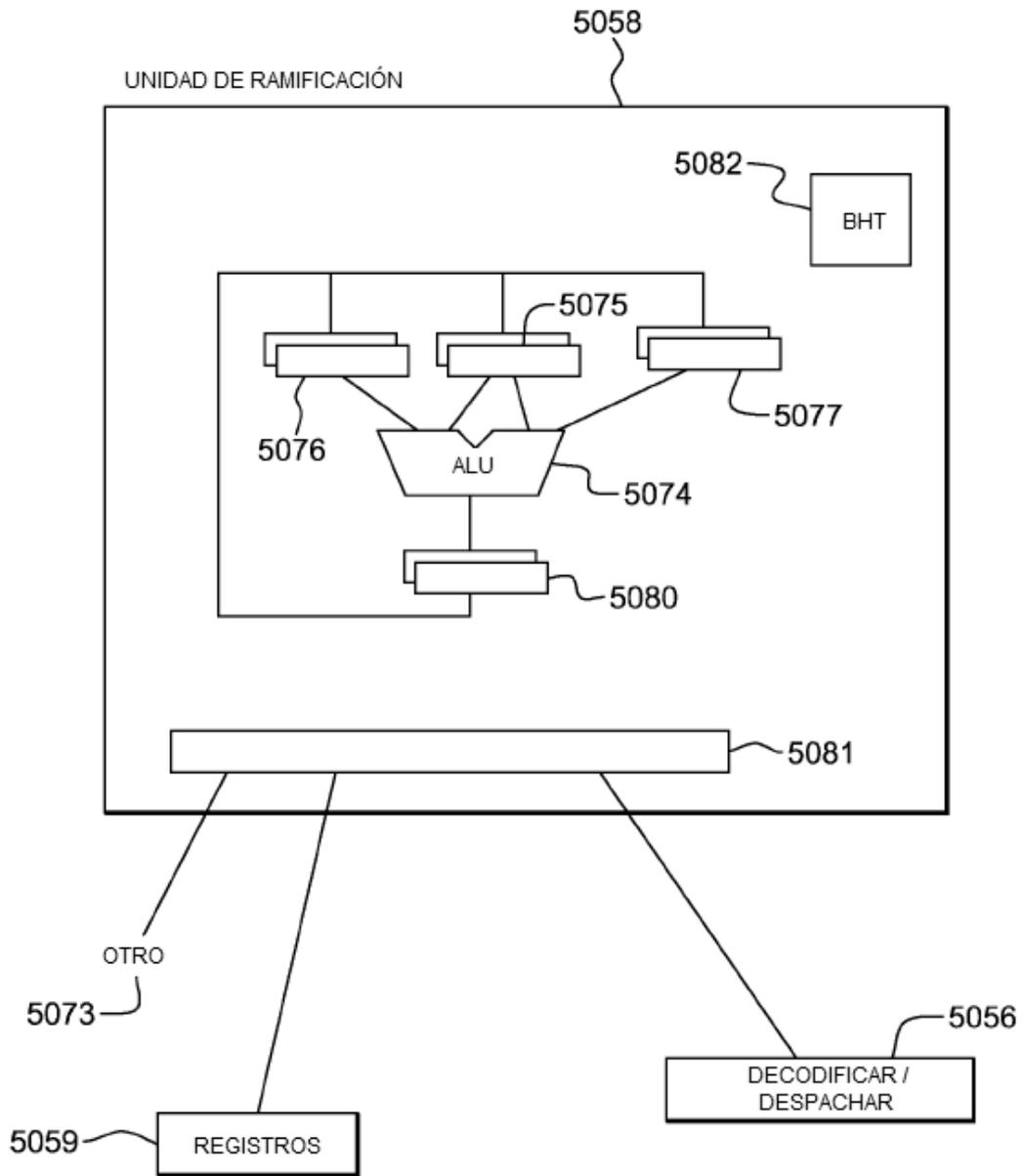


FIG. 18B

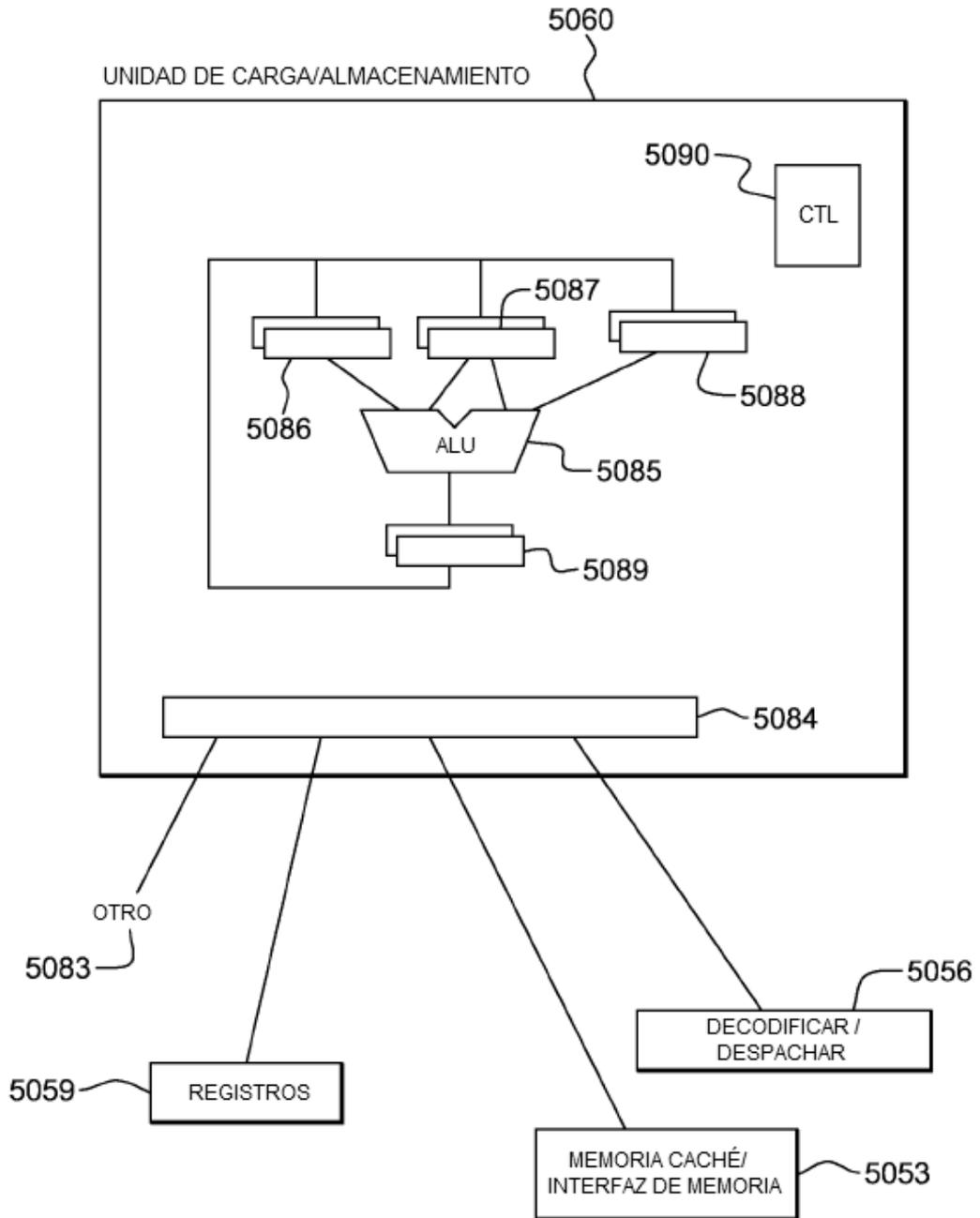


FIG. 18C

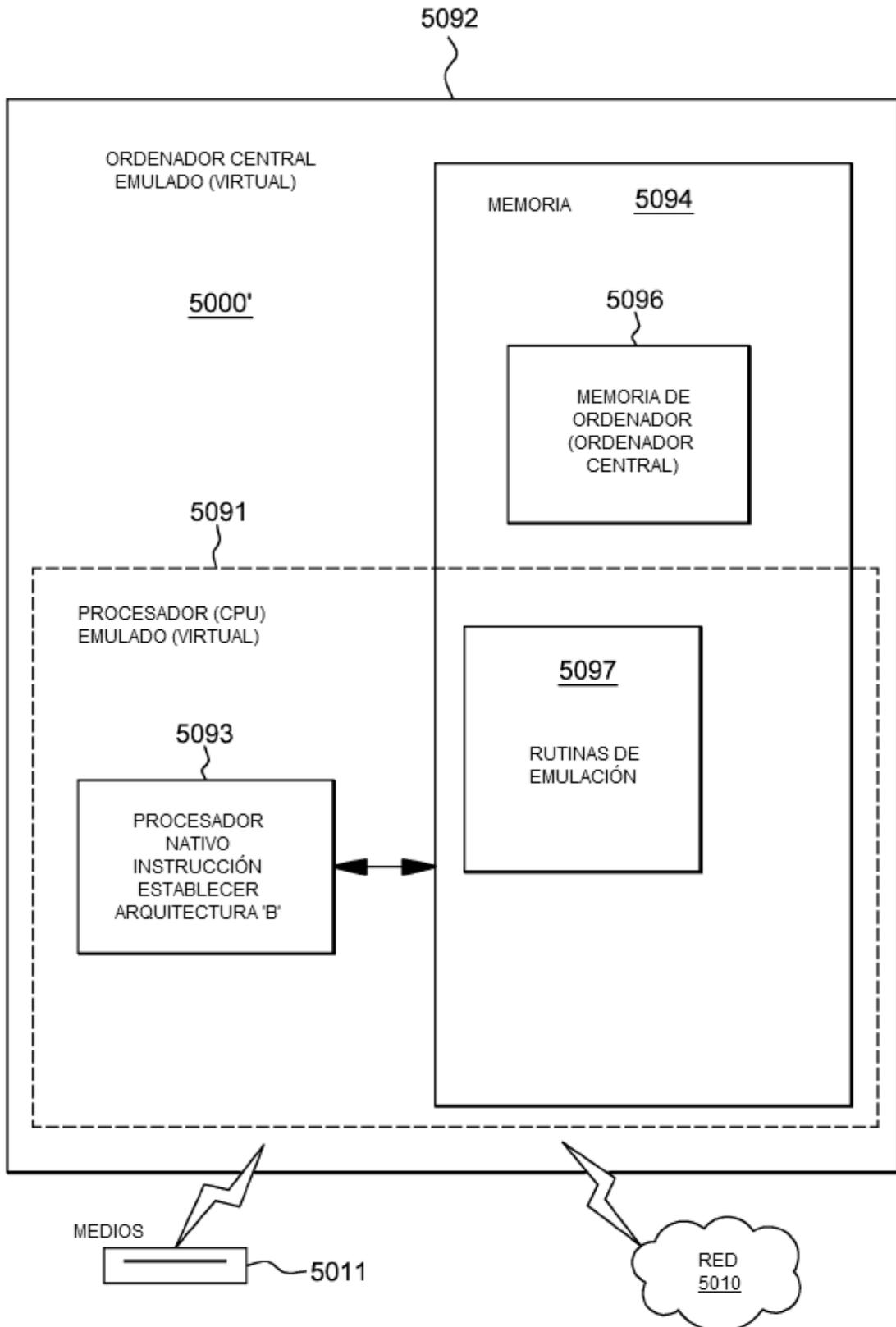


FIG. 19

6010

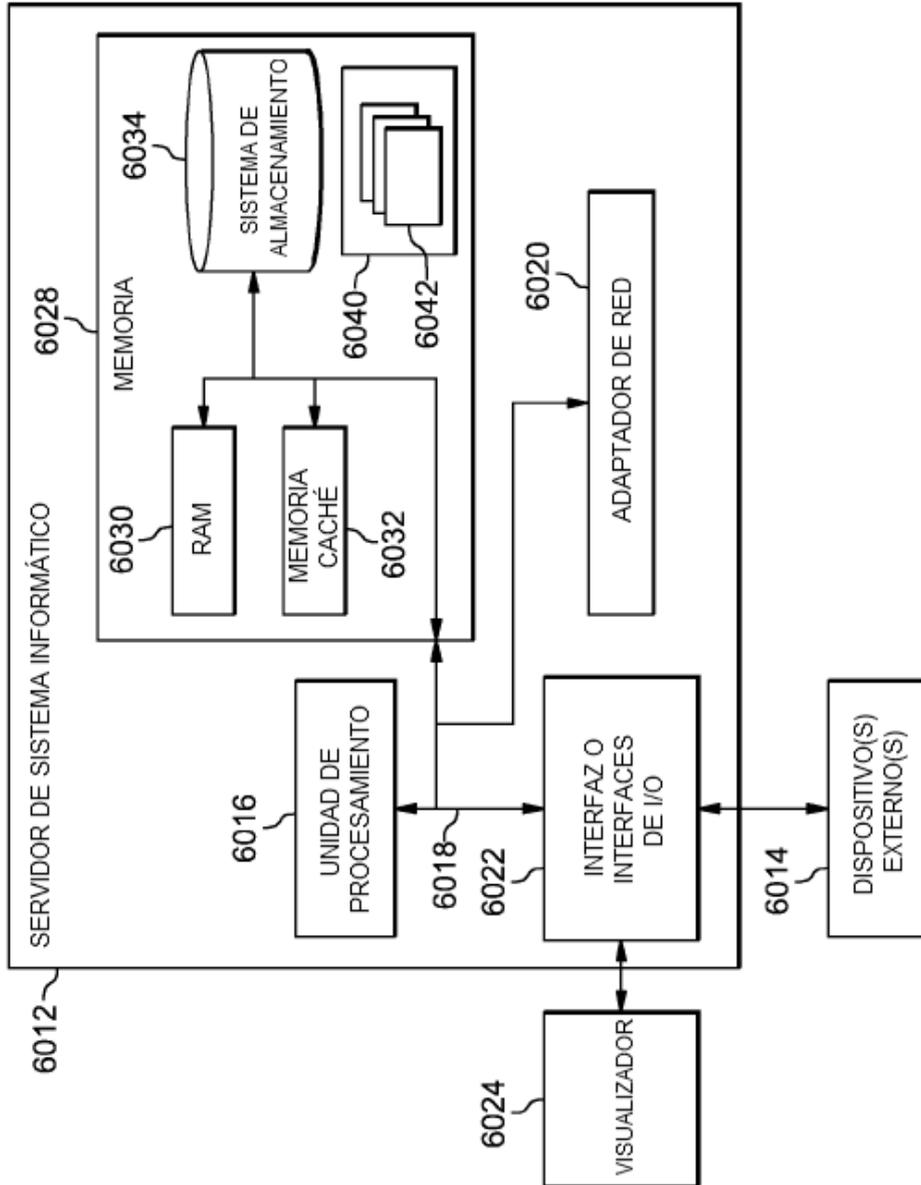


FIG. 20

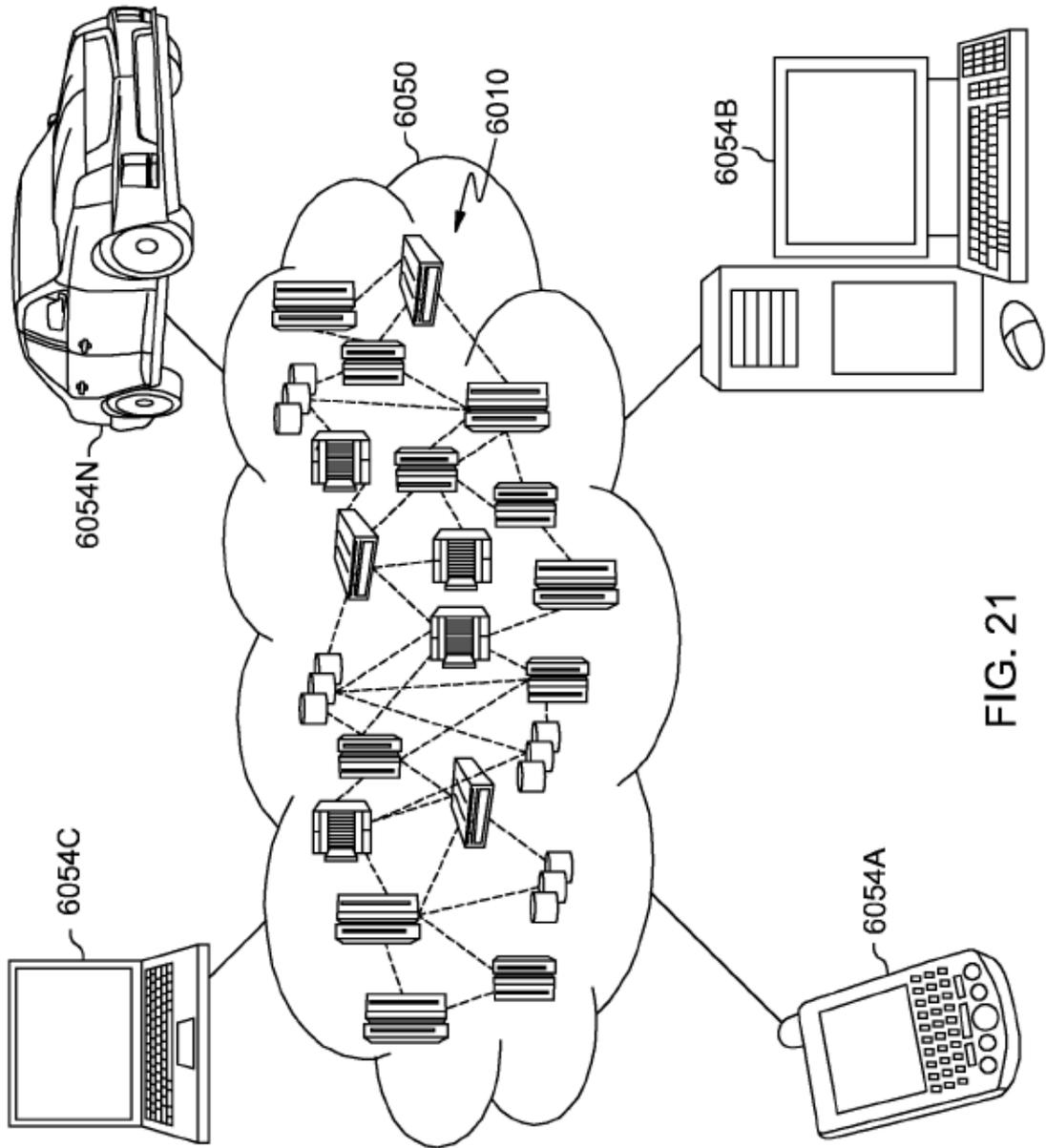


FIG. 21

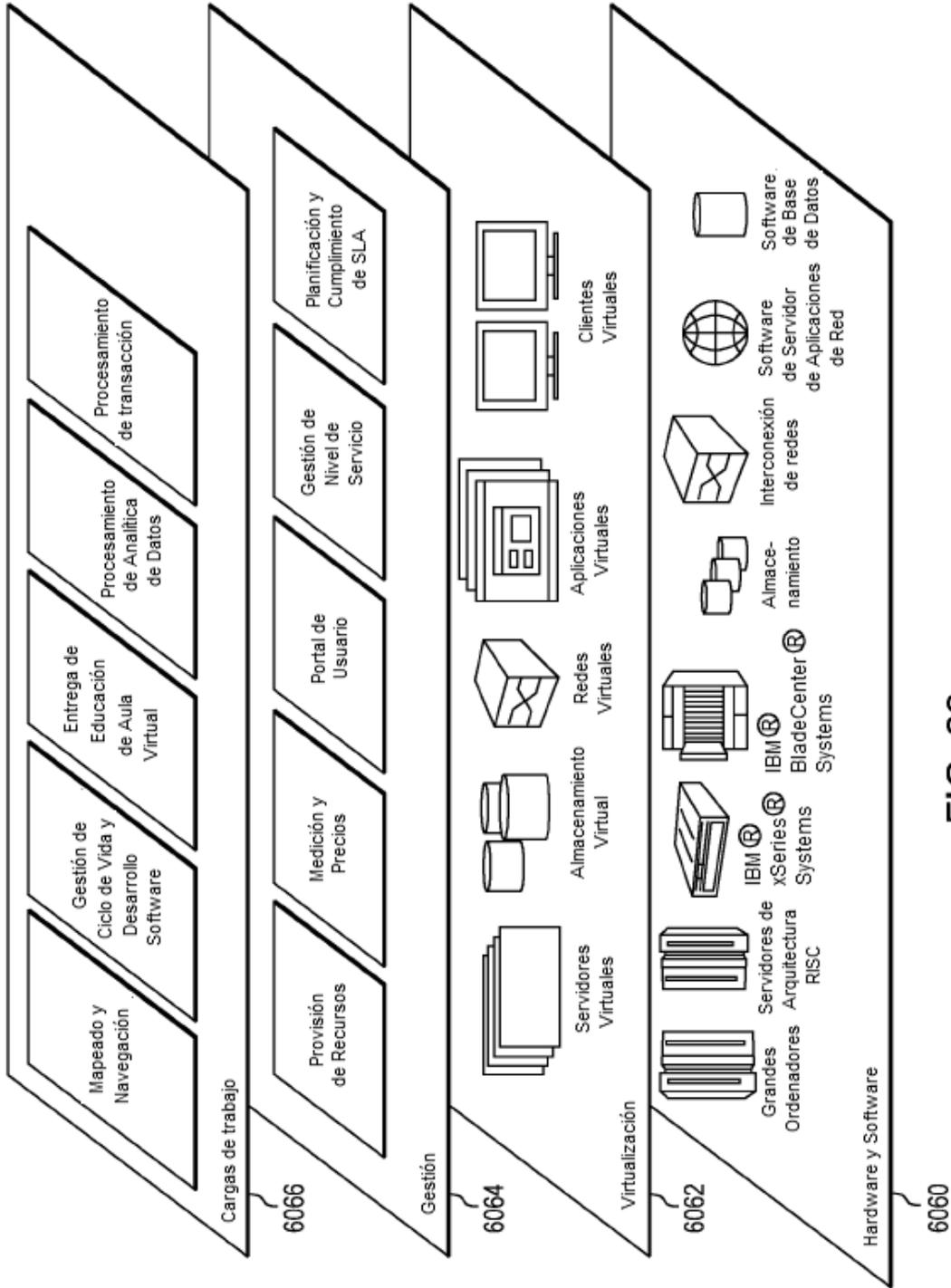


FIG. 22