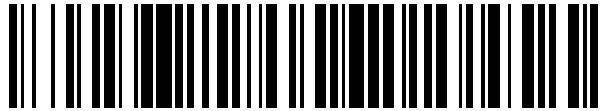


19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 746 479**

51 Int. Cl.:

**G06F 9/44**

(2008.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **20.02.2015 PCT/EP2015/053638**

87 Fecha y número de publicación internacional: **24.09.2015 WO15139917**

96 Fecha de presentación y número de la solicitud europea: **20.02.2015 E 15705333 (1)**

97 Fecha y número de publicación de la concesión europea: **21.08.2019 EP 3120239**

54 Título: **Secuencia de arranque común para el programa utilitario de control capaz de ser inicializado en múltiples arquitecturas**

30 Prioridad:

**18.03.2014 US 201414217800  
26.11.2014 US 201414554313**

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

**06.03.2020**

73 Titular/es:

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

72 Inventor/es:

**GSCHWIND, MICHAEL KARL**

74 Agente/Representante:

**ISERN JARA, Jorge**

**ES 2 746 479 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**

Secuencia de arranque común para el programa utilitario de control capaz de ser inicializado en múltiples arquitecturas

5 Campo  
 Uno o más aspectos se relacionan, en general, con configuraciones de entornos informáticos, y en particular, con el soporte de múltiples configuraciones.

10 Técnica antecedente  
 Los entornos informáticos ofrecen un rango de capacidades y funciones dependiendo de las configuraciones arquitectónicas de los entornos. Dos arquitecturas que se han ofrecido por IBM® incluyen ESA/390 y z/Architecture®. ESA/390 es una arquitectura predecesora de z/Architecture®. Sin embargo, cuando se introdujo la z/Architecture, ESA/390 continuó siendo soportada. Para soportar ambas arquitecturas en un entorno, se siguen ciertos procedimientos. Por ejemplo, con el encendido, se arranca ESA/390, y luego, se puede hacer una conmutación a la z/Architecture, si se desea. Esto permite que el software heredado continúe ejecutándose sin un cambio. Otros de dichos procedimientos se proporcionan con el fin de soportar ambas configuraciones arquitectónicas en un entorno.

20 El mantener una arquitectura anterior incluso cuando el software ha pasado a una arquitectura posterior agrega complejidad. El número de Publicación de Solicitud de Patente de los Estados Unidos US 2012/0297180 A1 Teng, C-Y, et al. "Method of Switching Between Multiple Operating Systems of Computer System", 22 de noviembre de 2012) divulga un método de conmutación entre múltiples sistemas operativos de un sistema informático que incluyen las siguientes etapas. En primer lugar, el sistema informático se encuentra en un entorno de un primer sistema operativo. Luego, se activa una interrupción de gestión del sistema para permitir que el sistema informático entre en un modo de gestión del sistema, y una autoridad de control del sistema informático se transfiere desde el primer sistema operativo hasta un sistema básico de entrada y salida. Luego se crea una copia de seguridad de un parámetro ambiental del primer sistema operativo. Si el segundo parámetro ambiental no está incluido en el sistema informático, se carga un segundo sistema operativo en modo normal. Por otro lado, si el segundo parámetro ambiental está incluido en el sistema informático, el segundo sistema operativo se arranca de acuerdo con el segundo parámetro ambiental.

25 Por lo tanto, subsiste la necesidad en la técnica de tratar el problema mencionado anteriormente.

35 Resumen

De acuerdo con la presente invención se proporcionan un método, un sistema, un producto de programa informático y un programa informático de acuerdo con las reivindicaciones independientes.

40 Se obtienen características y ventajas adicionales. Otras realizaciones y aspectos preferidos se describen en detalle en el presente documento y se consideran una parte de la invención reivindicada.

45 Breve descripción de los dibujos

Ahora se describirá la presente invención solo a modo de ejemplo, con referencia a las realizaciones preferidas, como se ilustra en las siguientes figuras:  
 La Figura 1 representa un ejemplo de un entorno informático para incorporar y utilizar uno o más aspectos de una funcionalidad de secuencia de arranque común, de acuerdo con una realización preferida de la presente invención;  
 50 La Figura 2 representa otro ejemplo de un entorno informático para incorporar y utilizar uno o más aspectos de una funcionalidad de secuencia de arranque común, de acuerdo con una realización preferida de la presente invención;  
 La Figura 3A representa aún otro ejemplo de un entorno informático para incorporar y utilizar uno o más aspectos de una funcionalidad de secuencia de arranque común, de acuerdo con una realización preferida de la presente invención;  
 55 La Figura 3B representa detalles adicionales de la memoria de la Figura 3A, de acuerdo con una realización preferida de la presente invención;  
 La Figura 4A representa una realización de la lógica para encender un entorno informático en un modo arquitectónico, de acuerdo con una realización preferida de la presente invención;  
 La Figura 4B representa una realización de procesamiento adicional asociado con el proceso de encendido de la Figura 4A, de acuerdo con una realización preferida de la presente invención;  
 60 La Figura 5 representa una realización de un formato de una palabra de estado de programa, de acuerdo con una realización preferida de la presente invención;  
 La Figura 6A representa una realización de la lógica para encender un entorno informático en un modo arquitectónico diferente del modo arquitectónico encendido en la Figura 4A, de acuerdo con una realización preferida de la presente invención;  
 65

La Figura 6B representa una realización de procesamiento adicional asociado con el proceso de encendido de la Figura 6A, de acuerdo con una realización preferida de la presente invención;

La Figura 7 representa un formato de ejemplo de una instrucción de Palabra de Estado de Programa de Carga, de acuerdo con una realización preferida de la presente invención;

5 La Figura 8A representa un formato de ejemplo de una instrucción de Procesador de Señal, de acuerdo con una realización preferida de la presente invención;

La Figura 8B representa una realización del procesamiento asociado con la instrucción de Procesador de Señal de la Figura 8A, de acuerdo con una realización preferida de la presente invención;

10 La Figura 9 representa una realización de lógica para realizar un arranque en el que un programa utilitario de control se inicializa en ESA/390, de acuerdo con una realización preferida de la presente invención;

La Figura 10 representa una realización de lógica para realizar un arranque en el que un programa utilitario de control se inicializa en z/Architecture;

La Figura 11 representa una realización de lógica para una secuencia de arranque común, de acuerdo con una realización preferida de la presente invención;

15 La Figura 12 representa otra realización de lógica para una secuencia de arranque común, de acuerdo con una realización preferida de la presente invención;

La Figura 13 representa un formato de ejemplo de una Palabra de Estado de Programa de Extracto, de acuerdo con una realización preferida de la presente invención;

20 La Figura 14 representa una realización adicional de lógica para una secuencia de arranque común, de acuerdo con una realización preferida de la presente invención;

La Figura 15 representa una realización de lógica para realizar un reinicio para volver a poner en marcha un programa utilitario de control, de acuerdo con una realización preferida de la presente invención;

La Figura 16 representa otra realización de lógica para realizar un reinicio para volver a poner en marcha un programa utilitario de control, de acuerdo con una realización preferida de la presente invención;

25 La Figura 17 representa todavía otra realización de lógica para realizar un reinicio para volver a poner en marcha un programa utilitario de control, de acuerdo con una realización preferida de la presente invención;

La Figura 18 representa una realización de lógica para detectar un programa utilitario de control incompatible que se relaciona con la secuencia de arranque, de acuerdo con una realización preferida de la presente invención;

30 La Figura 19 representa una realización del procesamiento asociado con realizar un arranque de un programa utilitario de control, de acuerdo con una realización preferida de la presente invención;

La Figura 20 representa una realización de un producto de programa informático, de acuerdo con la técnica anterior, y en el que se puede implementar una realización preferida de la presente invención;

La Figura 21 representa una realización de un sistema informático central, de acuerdo con la técnica anterior, y en el que se puede implementar una realización preferida de la presente invención;

35 La Figura 22 representa un ejemplo adicional de un sistema informático, de acuerdo con la técnica anterior, y en la que se puede implementar una realización preferida de la presente invención;

La Figura 23 representa otro ejemplo de un sistema informático que comprende una red informática, de acuerdo con la técnica anterior, y en la que se puede implementar una realización preferida de la presente invención;

40 La Figura 24 representa una realización de varios elementos de un sistema informático, de acuerdo con la técnica anterior, y en la que se puede implementar una realización preferida de la presente invención;

La Figura 25A representa una realización de la unidad de ejecución del sistema informático de La Figura 24, de acuerdo con la técnica anterior, y en la que se puede implementar una realización preferida de la presente invención;

La Figura 25B representa una realización de la unidad de derivación del sistema informático de la Figura 24, de acuerdo con la técnica anterior, y en la que se puede implementar una realización preferida de la presente invención;

45 La Figura 25C representa una realización de la unidad de carga/almacenamiento del sistema informático de la Figura 24, de acuerdo con la técnica anterior, y en la que se puede implementar una realización preferida de la presente invención;

La Figura 26 representa una realización de un sistema informático central emulado;

50 La Figura 27 representa una realización de un nodo informático en nube, de acuerdo con la técnica anterior, y en la que se puede implementar una realización preferida de la presente invención;

La Figura 28 representa una realización de un entorno informático en nube, de acuerdo con la técnica anterior, y en la que se puede implementar una realización preferida de la presente invención; y

La Figura 29 representa un ejemplo de capas del modelo de abstracción, de acuerdo con la técnica anterior, y en el que se puede implementar una realización preferida de la presente invención.

55 Descripción detallada

De acuerdo con uno o más aspectos, se proporciona una funcionalidad de secuencia de arranque común que habilita un programa utilitario de control (por ejemplo, sistema operativo, programa de control u otra herramienta independiente que se inicia sin necesidad del sistema operativo, tales como programas utilitarios de mantenimiento y gestión (por ejemplo, volcado, formato de disco), como ejemplos) para arrancar en una pluralidad de configuraciones sin cambiar la secuencia de arranque. Por ejemplo, un sistema operativo u otro programa utilitario de control utiliza la secuencia de arranque común para poder arrancar en una primera configuración de arquitectura que se inicializa en una arquitectura, por ejemplo, ESA/390, y luego conmuta a otra arquitectura para el procesamiento, tal como la z/Architecture; o en una segunda configuración arquitectónica que se inicializa y procesa en, por ejemplo, la otra arquitectura, por ejemplo, z/Architecture.

5 Cuando se introdujo por fases z/Architecture, los programas utilitarios de control continuaron arrancando en ESA/390, un predecesor de z/Architecture, y luego, durante la secuencia de arranque, el procesamiento conmutaría a z/Architecture. Esto permitió que el software heredado continuara ejecutándose sin cambios. Sin embargo, puede haber casos en los que sea deseable arrancar directamente en z/Architecture. Por lo tanto, se proporciona una secuencia de arranque común que permite que un programa utilitario de control arranque de forma libre en cualquier configuración. Por ejemplo, un sistema operativo que está escrito para z/Architecture utiliza la secuencia de arranque para arrancar en ESA/390 y luego conmutar a z/Architecture, o simplemente arrancar directamente en z/Architecture. Aunque las realizaciones en el presente documento se describen con referencia a ESA/390 y z/Architecture, uno o más aspectos también son aplicables a otras arquitecturas.

15 Un ejemplo de un entorno informático para incorporar y utilizar uno o más aspectos de una funcionalidad de secuencia de arranque común se describe con referencia a la Figura 1. Con referencia a la Figura 1, en un ejemplo, un entorno 100 informático se basa en la z/Architecture, ofrecida por IBM. La z/Architecture se describe en una publicación de IBM titulada "z/Architecture - Principles of Operation," Publicación No. SA22-7932-09, 10ª Edición, septiembre 2012. Aunque el entorno informático se basa en la z/Architecture, en una realización preferida, también es compatible con una o más configuraciones arquitectónicas, como ESA/390.

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

25 La memoria 104 del procesador incluye, por ejemplo, una o más particiones 112 (por ejemplo, particiones lógicas) y el firmware 113 del procesador, que incluye un hipervisor 114 de partición lógica y otro firmware 115 de procesador. Un ejemplo del hipervisor 114 de partición lógica es el Processor Resource/Systems Manager™ (PR/SM). IBM, z/OS, z/VM, z/Architecture y Processor Resource/Systems Manager son marcas comerciales de International Business Machines Corporation, registradas en muchas jurisdicciones de todo el mundo. Otros nombres utilizados en este documento pueden ser marcas registradas, marcas comerciales o nombres de productos de International Business Machines Corporation u otras compañías.

30 Una partición lógica funciona como un sistema separado y tiene una o más aplicaciones 120, y opcionalmente, un sistema 122 operativo residente en el mismo, que puede diferir para cada partición lógica. En una realización preferida, 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. Como ejemplos, el sistema operativo puede incluir una funcionalidad 124 de secuencia de arranque común o la funcionalidad de secuencia de arranque común se puede proporcionar por separado del sistema operativo, pero se utiliza para arrancar (o inicializar) el sistema operativo.

35 Las particiones 112 lógicas son gestionadas por el hipervisor 114 de particiones lógicas, que se implementa mediante firmware que se ejecuta en los procesadores 110. Como se utiliza en este documento, el firmware incluye, por ejemplo, el microcódigo y/o el milicódigo del procesador. Incluye, por ejemplo, las instrucciones de nivel de hardware y/o las estructuras de datos utilizadas en la implementación del código de máquina de nivel superior. En una realización preferida, incluye, por ejemplo, código privado que se entrega normalmente 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.

40 Los procesadores 110 centrales son recursos de procesador físicos asignados a las particiones lógicas. En particular, cada partición 112 lógica tiene uno o más procesadores lógicos, cada uno de los cuales representa todo o una parte de un procesador 110 físico asignado a la partición. Los procesadores lógicos de una partición 112 particular están dedicados a la partición, de modo que el recurso de procesador subyacente 110 está reservado para esa partición; o compartido 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 pueden incluir aspectos de una funcionalidad 130 de modo arquitectónico de configuración descrita en este documento.

45 El subsistema 111 de entrada/salida dirige el flujo de información entre los dispositivos 106 de entrada/salida y el almacenamiento 104 principal. Está acoplado al complejo de procesamiento central, ya que puede ser 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 simultáneamente con el procesamiento de entrada/salida. Para proporcionar comunicaciones, el subsistema de I/O emplea adaptadores de comunicaciones de I/O. Existen varios tipos de adaptadores de comunicaciones que incluyen, por ejemplo, canales, adaptadores de I/O, tarjetas PCI, tarjetas Ethernet, tarjetas de interfaz de almacenamiento de ordenador pequeño (SCSI), etc. En el ejemplo particular descrito en este documento, los adaptadores de comunicaciones de I/O son canales y, por lo tanto, el subsistema de I/O se

denomina en este documento un subsistema de canales. Sin embargo, este es solo un ejemplo. Se pueden utilizar otros tipos de subsistemas de I/O.

5 El subsistema de I/O utiliza una o más rutas de entrada/salida como enlaces de comunicación para gestionar el flujo de información hacia o desde los dispositivos 106 de entrada/salida. En este ejemplo particular, estas rutas se denominan rutas de canal, ya que los adaptadores de comunicación son canales.

10 Otro ejemplo de un entorno informático para incorporar y utilizar uno o más aspectos de una funcionalidad de secuencia de arranque común se describe con referencia a la Figura 2. En este ejemplo, un entorno 200 informático incluye un entorno no particionado que se puede configurar para una pluralidad de modos arquitectónicos, que incluyen z/Architecture y ESA/390. Incluye, por ejemplo, un procesador 202 (unidad central de procesamiento - CPU) que incluye, por ejemplo, una funcionalidad 204 de modo arquitectónico de configuración, y uno o más cachés 206. El procesador 202 está acoplado comunicativamente a una porción de memoria 208 que tiene uno o más cachés 210, y a un subsistema 212 de entrada/salida (I/O). El subsistema 212 de I/O está acoplado comunicativamente a dispositivos 214 de I/O externos que pueden incluir, por ejemplo, dispositivos de entrada de datos, sensores y/o dispositivos de salida, tales como pantallas.

20 En un ejemplo, la porción 208 de memoria también incluye al menos un programa 220 utilitario de control, tal como un sistema operativo, que utiliza una secuencia 222 de arranque común al arrancar.

25 Otra realización preferida de un entorno informático para incorporar y utilizar uno o más aspectos de una funcionalidad de secuencia de arranque común se describe con referencia a la Figura 3A. En este ejemplo, un entorno 300 informático incluye, por ejemplo, una unidad 302 central de procesamiento (CPU) nativa, una memoria 304 y uno o más dispositivos de entrada/salida y/o interfaces 306 acoplados entre sí mediante, por ejemplo, uno o más buses 308 y/u otras conexiones. Como ejemplos, el entorno 300 informático puede incluir un procesador PowerPC® o un servidor Power Systems™; un HP Superdome con procesadores Intel® Itanium® II ofrecido 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. PowerPC y Power Systems son marcas comerciales de International Business Machines Corporation, registradas en muchas jurisdicciones de todo el mundo. Intel e Itanium son marcas comerciales o marcas comerciales registradas de Intel Corporation o sus subsidiarias en los Estados Unidos y otros países.

35 La unidad 302 de procesamiento central nativa incluye uno o más registros 310 nativos, tales como uno o más registros de propósito general y/o uno o más registros de propósito especial utilizados durante el procesamiento dentro del entorno, y opcionalmente, una funcionalidad 311 de modo arquitectónico de configuración. Estos registros incluyen información que representa el estado del medio ambiente en un momento determinado.

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

45 Se describen detalles adicionales relacionados con el código 312 de emulador con referencia a la Figura 3B. Las instrucciones 350 para invitados almacenadas en la memoria 304 comprenden instrucciones de software (por ejemplo, correlacionadas con las instrucciones de la máquina) que se desarrollaron para ejecutarse en una arquitectura distinta de la CPU 302 nativa. Por ejemplo, las instrucciones 350 para invitados pueden haber sido diseñadas para ejecutarse en un procesador 202 z/Architecture, pero en su lugar, se están emulando en la CPU 302 nativa, que puede ser, por ejemplo, un procesador Intel Itanium II. En un ejemplo, las instrucciones para invitados incluyen una secuencia 351 de arranque común. En un ejemplo, el código 312 de emulador incluye una rutina 352 de extracción de instrucciones para obtener una o más instrucciones 350 para invitados de la memoria 304 y, opcionalmente, proporcionar almacenamiento intermedio local para las instrucciones obtenidas. También incluye una rutina 354 de traducción de instrucciones para determinar el tipo de instrucción de invitado que se ha obtenido y para traducir la instrucción de invitado en una o más instrucciones 356. Esta traducción incluye, por ejemplo, identificar la función que debe realizar la instrucción de invitado y elegir las instrucciones nativas para realizar esa función.

60 Adicionalmente, el código 312 de emulador incluye una rutina 360 de control de emulación para hacer que se ejecuten las instrucciones nativas. La rutina 360 de control de emulación puede hacer que la CPU 302 nativa ejecute una rutina de instrucciones nativas que emule una o más instrucciones de invitado obtenidas previamente y, al concluir dicha ejecución, devuelva el control a la rutina de extracción de instrucciones para emular la obtención de la siguiente instrucción de invitado o un grupo de instrucciones para invitados. La ejecución de las instrucciones 356 nativas puede incluir cargar datos en un registro desde la memoria 304; almacenar datos en la memoria desde un registro; o realizar algún tipo de operación aritmética o lógica, según lo determine la rutina de traducción.

Cada rutina se implementa, por ejemplo, en un software, que se almacena en la memoria y se ejecuta mediante la unidad 302 de procesamiento central nativa. En otros ejemplos, una o más de las rutinas u operaciones se implementan en firmware, hardware, software o alguna combinación de los mismos. Los registros del procesador emulado se pueden emular utilizando los registros 310 de la CPU nativa o utilizando ubicaciones en la memoria 304.

5 En realizaciones preferidas, las instrucciones 350 para invitados, las instrucciones 356 nativas y el código 312 de emulador pueden residir en la misma memoria o se pueden desembolsar entre diferentes dispositivos de memoria.

Los entornos informáticos descritos anteriormente son solo ejemplos de entornos informáticos que se pueden utilizar. Se pueden utilizar otros entornos, incluidos, entre otros, otros entornos no particionados, otros entornos particionados y/u otros entornos emulados ambientes; las realizaciones no están limitadas a ningún entorno.

10

Como se describe en este documento, se proporciona una secuencia de arranque común para iniciar un programa utilitario de control en una de una pluralidad de configuraciones arquitectónicas, que incluyen, por ejemplo, una primera configuración en la que el programa utilitario de control se inicializa en un primer modo arquitectónico (por ejemplo, un modo heredado, como ESA/390) y luego, el procesamiento cambia a un segundo modo arquitectónico (por ejemplo, un modo mejorado, tal como z/Architecture); o una segunda configuración en la que el programa utilitario de control se inicializa y procesa, por ejemplo, en el segundo modo arquitectónico. En una realización, el primer modo arquitectónico incluye una primera Establecer Arquitectura de instrucciones y tiene un primer conjunto de características compatibles, tales como direccionamiento de 31 bits, uso de registros de propósito general de 32 bits y diversas instalaciones. Adicionalmente, el segundo modo arquitectónico incluye una segunda Establecer Arquitectura de instrucciones y tiene un segundo conjunto de características compatibles, como el direccionamiento de 64 bits, el uso de registros de propósito general de 64 bits y diversas instalaciones, tales como la traducción dinámica de direcciones y/u otras instalaciones.

15

20

En una realización, la segunda configuración en la que se facilita la funcionalidad de arranque se inicializa y procesa directamente en un modo arquitectónico seleccionado, tal como z/Architecture, mediante la instalación de una funcionalidad de modo arquitectónico de configuración. Esta funcionalidad, como se describe en este documento, proporciona una secuencia de encendido o arranque para el entorno informático (por ejemplo, procesador, LPAR, invitado virtual) que arranca el entorno informático en el modo arquitectónico seleccionado. Por lo tanto, antes de proporcionar más detalles sobre la secuencia de arranque común, se describen aspectos de una funcionalidad de modo arquitectónico de configuración. Sin embargo, se observa que la capacidad de secuencia de arranque común también se puede utilizar en entornos que no incluyen una funcionalidad de modo arquitectónico de configuración.

25

30

En una realización, se instala una funcionalidad de modo arquitectónico de configuración (CAM) en uno o más procesadores (por ejemplo, unidades de procesamiento central) de un entorno informático para controlar la reconfiguración del entorno. Por ejemplo, cuando CAM se instala en un entorno informático que admite una pluralidad de modos arquitectónicos, el entorno informático se reconfigura de modo que se restringe el uso de uno o más aspectos de al menos uno de los modos arquitectónicos.

35

Un ejemplo particular de una funcionalidad de modo arquitectónico de configuración es la funcionalidad de Modo Arquitectónico de Configuración z/Architecture (CZAM). La instalación de CZAM se indica mediante, por ejemplo, un indicador de instalación de funcionalidad, por ejemplo, bit 138, establecido en, por ejemplo, uno. En un ejemplo particular, cuando el bit 138 se establece en uno, se instala la funcionalidad de CZAM, y cuando se instala, un reinicio normal y un reinicio de borrado colocan la configuración en el modo arquitectónico z/Architecture. Por lo tanto, el bit de funcionalidad, por ejemplo, el bit 2, que indica que el modo arquitectónico z/Architecture está activo, también se establece en uno, en un ejemplo.

40

45

Basado en la instalación de CZAM, un entorno informático (por ejemplo, un único procesador, una partición lógica, un invitado virtual, etc.) se reconfigura de modo que uno o más aspectos de una arquitectura seleccionada, por ejemplo, ESA/390, ya no es compatible. A continuación, se describen varios aspectos que ya no son compatibles y/o ciertos procesos afectados por la instalación de CZAM. Aunque en las realizaciones preferidas descritas en el presente documento, la pluralidad de modos arquitectónicos incluye una arquitectura heredada (por ejemplo, ESA/390) y una arquitectura mejorada (por ejemplo, z/Architecture) y los aspectos de la arquitectura heredada, ESA/390, ya no son compatibles, otras realizaciones preferidas pueden incluir otras arquitecturas. ESA/390 y z/Architecture son solo ejemplos.

50

55

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, se describe un proceso de encendido para un entorno que soporta múltiples configuraciones arquitectónicas y no incluye la funcionalidad de CZAM con referencia a las Figura 4A-4B, y luego se describe un proceso de encendido para un entorno que está configurado para múltiples configuraciones arquitectónicas y que incluye la funcionalidad de CZAM con referencia a las Figuras 6A-6B. El encendido de un sistema incluye, por ejemplo, comenzar el sistema e iniciar una secuencia de arranque u otros medios para iniciar 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 para invitado).

60

65

Con referencia inicialmente a la Figura 4A, con base en un procesador informático el entorno se enciende y se activa una tecla de operador, por ejemplo, una tecla de carga normal o una tecla de borrado de carga, el procesador entra en un estado de carga y establece el entorno informático en un modo arquitectónico particular, por ejemplo, modo ESA/390, ETAPA 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), ETAPA 402. El arranque del programa 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 al configurar los controles de dirección de la unidad de carga en un número de cuatro dígitos para designar un dispositivo de entrada y al activar posteriormente la tecla de borrado de carga o de carga normal para una CPU en particular.

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

En la parte de carga de la operación, una vez que se han realizado los reinicios, esta CPU ingresa al estado de carga. Esta CPU no entra necesariamente en el estado detenido durante la ejecución de las operaciones de reinicio. El indicador de carga está encendido mientras la CPU está en 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 un formato 0 CCW que comienza en la ubicación de almacenamiento absoluto 0 especificara un comando de lectura con los bits modificadores ceros, una dirección de datos de cero, un conteo de bytes de 24, el comando de cadena y los indicadores SLI unos, y todos los demás indicadores ceros.

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

Si la carga de PSW es exitosa y no se detectan fallas en la máquina, esta CPU deja el estado de carga y el indicador de carga se apaga. Si el control de velocidad se establece en la posición del proceso, la CPU entra en el estado operativo y la operación del entorno informático continúa bajo el control de la nueva palabra de estado del programa (PSW), ETAPA 406. Luego se ejecuta el entorno informático de arranque, ETAPA 408, como se describe adicionalmente con referencia a la Figura 4B.

En referencia a la Figura 4B, el entorno informático de arranque se inicia en modo ESA/390, ETAPA 420, y por lo tanto, las operaciones se realizan en modo ESA/390, ETAPA 422. En algún momento, se puede hacer una solicitud para cambiar el modo arquitectónico desde ESA/390 hasta 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 de Procesador de Señal (SIGP) con el código de orden para conmutar desde el modo ESA/390 hasta el modo z/Architecture, ETAPA 424. Por ejemplo, se utiliza una funcionalidad de señalización y respuesta de la CPU que incluye la instrucción del Procesador de Señales (descrita a continuación) y un mecanismo para interpretar y actuar sobre los códigos de orden, que incluye Establecer Arquitectura. La funcionalidad proporciona comunicaciones entre las CPU, que incluye la transmisión, recepción y decodificación de un conjunto de códigos de orden asignados; iniciar la operación especificada; y responder a una CPU de señalización. Al utilizar Establecer Arquitectura, el modo arquitectónico se establece en la configuración deseada, por ejemplo, z/Architecture. Detalles adicionales de este procesamiento se describen más adelante.

Después de esto, se hace una determinación de que, si se aceptó la operación SIGP, PREGUNTA 426. Con base en el código de retorno, se pueden diagnosticar una serie de condiciones de error, incluida una indicación de "parámetro no válido" cuando se ha determinado que la CPU ya está en el modo arquitectónico especificado por el código (es decir, que la arquitectura establecida representa una conmutación al modo actual, o si se trata de una conmutación de un modo a otro modo). Si se acepta el SIGP y la arquitectura del conjunto representa una operación de conmutación de modo legal, entonces todos los procesadores del entorno informático que recibieron la operación de SIGP pasan al modo z/Architecture utilizando, por ejemplo, el procesamiento de Establecer Arquitectura descrito en este documento, ETAPA 428. Sin embargo, si la operación SIGP no es legal, se indica un error, ETAPA 430.

Como se describió anteriormente, la operación de encendido carga una palabra de estado del programa. Una realización de un formato de una palabra de estado de programa (PSW) se describe con referencia a la Figura 5) Con referencia a la Figura 5, en este ejemplo, el formato de la palabra de estado del programa es un formato ESA/390, excepto que el bit 31 se muestra como EA, como se indica a continuación.

## ES 2 746 479 T3

En una realización, una palabra 500 de estado del programa incluye los siguientes campos, como un ejemplo:

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

Modo 504 DAT (T): el bit 5 controla si tiene lugar la traducción implícita de direcciones dinámicas (DAT) de direcciones lógicas y de instrucciones utilizadas para acceder al almacenamiento. Cuando el bit es cero, DAT está desactivado y las direcciones lógicas y de instrucciones se tratan como direcciones reales. Cuando el bit es uno, DAT está encendido y se invoca el mecanismo de traducción de dirección dinámica.

Máscara 506 de I/O (IO): el bit 6 controla si la CPU está habilitada para interrupciones I/O. Cuando el bit es cero, no se puede producir 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 6 de control. Cuando un bit de máscara de subclase de interrupción de I/O es cero, no se puede producir 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, se puede producir una interrupción de I/O para esa subclase de interrupción de I/O;

Máscara 508 externa (EX): el bit 7 controla si la CPU está habilitada para la interrupción por condiciones incluidas en la clase externa. Cuando el bit es cero, no se puede producir 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 provocar una interrupción. Cuando el bit de máscara de subclase es uno, se puede producir una interrupción en esa subclase.

Tecla 510 PSW (Tecla): los bits 9-11 forman la tecla de acceso para las referencias de almacenamiento de la CPU. Si la referencia está sujeta a protección controlada por clave, la tecla PSW se hace coincidir con una tecla de almacenamiento cuando se almacena la información o cuando la información se extrae de una ubicación que está protegida contra la extracción. Sin embargo, para uno de los operandos de cada uno de Mover a Primaria, Mover a Secundaria, Mover con Tecla, Mover con Tecla Fuente y Mover con Tecla de Destino, y para uno o ambos operandos de Mover con especificaciones Opcionales, se utiliza una tecla de acceso especificada como un operando en lugar de la tecla PSW.

Bit 12 512: este bit indica el modo arquitectónico actual. Está configurado a uno para el formato ESA/390 PSW. Para el formato z/Architecture PSW, este bit se define como cero. Cuando está en modo z/Architecture, se define una instrucción de carga PSW extendida (LPSWE) para cargar una verdadera PSW z/Architecture (que tiene un formato diferente al formato descrito en este documento, que incluye tener una dirección de instrucción en los bits 64-127). Sin embargo, un PSA de carga ESA/390 (LPSW) todavía es soportado y se puede utilizar para cargar un PSW de formato ESA/390. Cuando se ejecuta LPSW y el entorno informático está en modo z/Architecture, el procesador expande el formato ESA/390 PSW al formato z/Architecture, incluido el bit 12 de inversión. Esto es lo contrario de colapsar el formato z/Architecture PSW que realiza el sistema operativo para crear el formato ESA/390 PSW. Es decir, en entornos informáticos que soportan ESA/390 y z/Architecture, cuando se almacena una copia de una PSW, el sistema operativo colapsa la PSW z/Architecture completa al tamaño y formato de una PSA ESA/390. Por lo tanto, otro software con dependencias de formato PSW puede ignorar el z/Architecture PSW.

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

Estado 516 de espera (W): cuando el bit 14 es uno, la CPU está en espera; es decir, la CPU no procesa instrucciones, pero pueden producirse interrupciones. Cuando el bit 14 es cero, la extracción y ejecución de instrucciones se produce de la manera normal. El indicador de espera es uno cuando el bit es uno.

Estado 518 del problema (P): cuando el bit 15 es uno, la CPU está en el estado del 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 del problema, solo son válidas las instrucciones que proporcionan información significativa al programa del problema y que no pueden afectar la integridad del sistema; tales instrucciones se llaman instrucciones no privilegiadas. Las instrucciones que no son válidas en el estado del problema se denominan instrucciones privilegiadas. Cuando una CPU en el estado problemático 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 del problema solo si se cumplen pruebas de autoridad específicas; de lo contrario, se reconoce una excepción de operación privilegiada o alguna otra excepción de programa, dependiendo del requisito particular que se viole.



## ES 2 746 479 T3

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

5 Código 522 de condición (CC): 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 al ejecutar ciertas instrucciones. La mayoría de las operaciones aritméticas y lógicas, así como algunas otras operaciones, establecen el código de condición. La instrucción DERIVACIÓN EN CONDICIÓN puede especificar cualquier selección de los valores del código de condición como criterio para la derivación.

10 Máscara 524 de programa: los bits 20-23 son los cuatro bits de máscara de programa. Cada bit está asociado con una excepción de programa, como sigue:

| Bit de Máscara DE Programa | Excepción del programa    |
|----------------------------|---------------------------|
| 20                         | Sobreflujo de punto fijo  |
| 21                         | Sobreflujo de decimal     |
| 22                         | Subflujo de exponente HFP |
| 23                         | Significancia de HFP      |

15 Cuando el bit de máscara es uno, la excepción resulta en una interrupción. Cuando el bit de máscara es cero, no se produce ninguna interrupción. La configuración del bit de máscara de subflujo de exponente de HFP del bit de máscara de significancia de HFP también determina la manera en que se completa la operación cuando se produce la excepción correspondiente.

20 Modo 526 de direccionamiento extendido (EA): el bit 31 controla el tamaño de las direcciones efectivas y la generación efectiva de direcciones 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.

25 Modo 528 de direccionamiento básico (BA): los bits 31 y 32 controlan el tamaño de las direcciones efectivas y la generación efectiva de direcciones. Cuando los bits 31 y 32 son ambos cero, se especifica el direccionamiento de 24 bits. Cuando el bit 31 es cero y el bit 32 es uno, se especifica el direccionamiento de 31 bits. Cuando los bits 31 y 32 son ambos uno, se especifica el direccionamiento de 64 bits. El bit 31 uno y el bit 32 cero son 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 utilizadas para acceder a DAT, ASN, control de unidades de capacidad de suministro, enlaces, entradas y tablas de seguimiento o listas de acceso o la pila de enlaces. 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-bit                   |
| 0       | 1       | 31-bit                   |
| 1       | 1       | 64-bit                   |

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

40 De acuerdo con un aspecto, cuando una funcionalidad de modo arquitectónico de configuración, tal como la funcionalidad de Modo Arquitectónico z/Architecture de configuración (CZAM), se instala y activa en el entorno informático, se cambia el proceso de encendido. Una realización de un proceso de encendido CZAM se describe con referencia a la Figura 6A.

45 Con referencia a la Figura 6A, con base en un procesador del entorno informático que se está encendiendo, el entorno informático se establece en el modo arquitectónico particular especificado por la funcionalidad del modo arquitectónico de configuración, por ejemplo, el modo z/Architecture (también denominado ESAME) cuando se instala CZAM, ETAPA 600. 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), como se describió anteriormente, ETAPA 602, y cuando la operación de entrada/salida de IPL se completa con éxito, un la palabra de identificación del subsistema para el dispositivo IPL se almacena en ubicaciones de almacenamiento absoluto seleccionadas (por ejemplo, ubicaciones 184-187), los ceros se almacenan en otras ubicaciones de almacenamiento absoluto seleccionadas (por ejemplo, ubicaciones 188-191), y en esta realización, una nueva palabra de estado del programa (PSW) de 16 bytes se crea a partir de ubicaciones de almacenamiento absoluto seleccionadas (por ejemplo, ubicaciones 0-7), ETAPA 604. El nuevo PSW de 16 bytes se forma, por ejemplo, a partir del contenido de la palabra doble de almacenamiento seleccionada (por ejemplo, ubicaciones 0-7). El bit 12 de la palabra doble es uno; de lo contrario, se puede indicar un error. (El error puede ser una excepción de especificación reconocida, una verificación de la máquina u 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 establecen en ceros. Las

## ES 2 746 479 T3

posiciones de bit 97-127 del PSW recién creado se inicializan a partir de los bits 33-63 de la palabra doble seleccionada.

5 En una realización, no se verifica la validez de los campos PSW que deben cargarse mediante la instrucción antes de cargarlos. En una realización, se verifica la validez del bit 12 de la PSW. En aún otra realización, se verifica la validez de todos los campos. En otra realización, cualquier bit no verificado antes de la carga de la PSW se verifica para verificar su validez después de que la PSW se haya inicializado, y el procesador puede indicar un error (por ejemplo, al generar una excepción de especificación que se reconoce, una verificación de la máquina o aún otra indicación de error).

10 El entorno informático entra en el estado operativo, y la operación del entorno informático continúa bajo el control de la nueva palabra de estado del programa (PSW), ETAPA 606. Luego se ejecuta el entorno informático de arranque, ETAPA 608, como se describe adicionalmente con referencia a la Figura 6B.

15 Con referencia a la Figura 6B, el entorno informático de arranque se inicia en modo z/Architecture, ETAPA 620, y por lo tanto, las operaciones se realizan en modo z/Architecture, ETAPA 622. No es necesario conmutar de modo, y el procesamiento continúa directamente con el procesamiento en el modo z/Architecture. Por lo tanto, en una realización, las siguientes etapas no son necesarios: una operación de Procesador de Señal (SIGP) para conmutar desde el modo ESA/390 hasta el modo z/Architecture; una determinación sobre 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.

20 Todos los procesadores del entorno informático (es decir, el entorno que se está configurando, por ejemplo, procesador único, partición lógica, invitado VM) están en modo z/Architecture, sin realizar las etapas indicadas anteriormente. Por lo tanto, como se describe en este documento, de acuerdo con un aspecto, la capacidad de arrancar o encender en modo ESA/390 se elimina del entorno informático configurado para ESA/390 y z/Architecture. En particular, aunque un entorno informático está configurado para admitir múltiples arquitecturas, se proporciona una capacidad para restringir ciertos aspectos de al menos una de las arquitecturas configuradas, siendo uno de los aspectos la capacidad de encender esa arquitectura.

25 En una o más realizaciones preferidas, el encendido en modo z/Architecture proporciona un mecanismo para especificar una de (1) una partición lógica (invitado-1), y (2) una partición lógica y un invitado-2 se deben arrancar y restablecer en modo z/Architecture, sin necesidad de iniciar en modo ESA/390. Esta característica se puede instalar incondicionalmente o bajo el control de un conmutador de configuración.

30 La secuencia de arranque con respecto a la inicialización de PSW se modifica. Por ejemplo, al final de IPL, se carga el IPL PSW en ubicaciones absolutas 0-7. Como se hace actualmente cuando la condición de reinicio es ESA/390, el bit 12 es uno, que hace un PSA IPL PSW ESA/390 válido, y el programa procede a ejecutar instrucciones en el modo arquitectónico ESA/390. Con CZAM instalado, la condición de reinicio es z/Architecture, el bit 12 sigue siendo uno, lo que hace un PSA IPL PSW ESA/390 válido, pero el bit 12 se invierte durante la formación del PSW actual de 16 bytes z/Architecture, como se definió anteriormente.

35 Además del proceso de encendido, otros procesos, comportamientos y/u operaciones también pueden modificarse o verse afectados por la instalación de una funcionalidad de modo arquitectónico de configuración. Estos procesos, comportamientos y/u operaciones afectados son específicos de los modos ESA/390 y z/Architecture. Sin embargo, procesos similares y/o diferentes pueden verse afectados por otros tipos de arquitecturas. Procesos, comportamientos y/u operaciones de ejemplo que se pueden ver afectados en una o más realizaciones preferidas incluyen, por ejemplo:

40 (1) Habilitar un conmutador de modo a auto (por ejemplo, desde el modo z/Architecture hasta el modo z/Architecture) sin generar un error (o ignorar el error). Es decir, un procesador puede emitir una instrucción SIGP para conmutar al modo z/Architecture y, si ya está en ese modo, no se generará ningún error. Anteriormente, intentaba conmutar al modo correspondiente al modo actual generaba un error.

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

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

- 5 Se describen detalles adicionales con respecto a la instrucción de PSW de Carga con referencia a la Figura 7) En una realización, una instrucción 700 de PSW de carga incluye un campo 702 de código de operación que incluye un código de operación para indicar una operación de PSW de carga; un campo 704 base (B<sub>2</sub>); y un campo 706 de desplazamiento (D<sub>2</sub>). El contenido del registro general designado por el campo B<sub>2</sub> se agrega al contenido del campo D<sub>2</sub> para formar una dirección de un segundo operando en almacenamiento (denominado como la segunda dirección de operando).
- 10 En la operación de la instrucción de PSW de carga, la PSW actual se reemplaza por una PSW de 16 bytes formada a partir del contenido de la palabra doble en la ubicación designada por la segunda dirección de operando.
- 15 El bit 12 de la palabra doble es uno; de lo contrario, se puede reconocer una excepción de especificación, dependiendo del modelo. Si se instala la funcionalidad del Modo Arquitectónico z/Architecture de Configuración, se reconoce una excepción de especificación si el bit 12 de la palabra doble no es uno.
- 20 Los bits 0-32 de la palabra doble, excepto con el bit 12 invertido, se colocan en las posiciones 0-32 de la PSW actual. Los bits 33-63 de la palabra doble se colocan en las posiciones 97-127 de la PSW actual. Los bits 33-96 de la PSW actual se establecen en cero.
- 25 Se realiza una función de serialización y sincronización del punto de comprobación antes o después de extraer el operando y nuevamente después de que se completa la operación.
- El operando se designará en un límite de doble palabra; de lo contrario, se reconoce una excepción de especificación.  
Se puede reconocer una excepción de especificación si el bit 12 del operando es cero, dependiendo del modelo.
- 30 No se verifica la validez de los campos PSW que deben cargarse con la instrucción antes de cargarlos, excepto la verificación del bit 12. Sin embargo, inmediatamente después de la carga, se reconoce una excepción de especificación y se produce una interrupción del programa, cuando cualquiera de lo siguiente es cierto para el PSW recién cargado:
- Cualquiera de los bits 0, 2-4, 12 o 24-30 es 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.
- 35 En estos casos, la operación se completa y el código de longitud de la instrucción resultante es 0.
- La operación se suprime en todas las excepciones de direccionamiento y protección. Código de Condición Resultante: el código se establece como se especifica en el nuevo PSW cargado.  
Excepciones del programa:
- Acceso (extracción, operando 2)
  - Operación privilegiada
  - Especificación
- 45 Nota de programación: el segundo operando debe tener el formato de un ESA/390 PSW. Se reconocerá una excepción de especificación durante o después de la ejecución de PSW DE CARGA si el bit 12 del operando es cero.
- 50 Se describen detalles ADICIONALES sobre el PSW en "Development and Attributes of z/Architecture," Plambeck et al., IBM J. Res. & Dev., Vol. 46, No. 4/5, July/September 2002.
- 55 Además de los procesos, operaciones y/o comportamientos anteriores que se pueden cambiar debido a la instalación de una funcionalidad de modo arquitectónico de configuración, el modo de reinicio también se puede cambiar en una o más realizaciones preferidas, como se explica a continuación.
- (4) Cambia el modo de reinicio (por ejemplo, para reinicio, reinicio de borrado y otras acciones para reinicio). Cuando se instala la funcionalidad de CZAM, el reinicio de la CPU establece el modo arquitectónico en el modo z/Architecture, si es provocado por la activación de, por ejemplo, la tecla de carga normal.
- 60 Se presenta una serie de funciones de reinicio que se incluyen como parte de los modos ESA/390 y z/Architecture, que incluyen, por ejemplo, reinicio de la CPU, reinicio de la CPU inicial, reinicio del subsistema, reinicio de borrado y reinicio de encendido, cada uno de los cuales se describe a continuación.
- 65 Reinicio de la CPU

El reinicio de la CPU proporciona un medio para borrar las indicaciones de verificación del equipo y cualquier imprevisibilidad resultante en el estado de la CPU con la menor cantidad de información destruida. En particular, se utiliza para borrar las condiciones de verificación cuando se debe preservar el estado de la CPU para el análisis o la reanudación de la operación. Si la funcionalidad de Modo Arquitectónico z/Architecture de Configuración (CZAM) no está instalada, el reinicio de la CPU establece el modo arquitectónico en el modo ESA/390 si es provocado por la activación de la tecla de carga normal (una funcionalidad del operador). Cuando se instala la funcionalidad de CZAM, el reinicio de la CPU establece el modo arquitectónico en el modo z/Architecture si es provocado por la activación de la tecla de carga normal. Cuando el reinicio de la CPU establece el modo ESA/390, guarda el PSW actual de tal manera que PSW se pueda restaurar mediante una orden Establecer Arquitectura de Procesador de Señal que cambia el modo arquitectónico de nuevo a z/Architecture.

El reinicio de la CPU provoca las siguientes acciones, en una realización:

1. Se finaliza la ejecución de la instrucción actual u otra secuencia de procesamiento, como una interrupción y se borran todas las condiciones de interrupción del programa y de interrupción de la llamada del supervisor.

2. Se borra cualquier condición de interrupción externa pendiente que sea local para la CPU. Las condiciones de interrupción externa flotante no se borran.

3. Se borran cualquier condición pendiente de interrupción de verificación de máquina e indicaciones de error que sean locales para la CPU y cualquier estado de detención de verificación. Las condiciones de interrupción de verificación de máquina flotante no se borran. Se dice que cualquier condición de verificación de máquina que se reporta a todas las CPU en la configuración y que se ha hecho pendiente a una CPU es local a la CPU.

4. Se borran todas las copias de instrucciones u operandos extraídos previamente. Adicionalmente, se borran todos los resultados que se almacenarán debido a la ejecución de instrucciones en el intervalo de punto de verificación actual.

5. El búfer ART (Traducción del Registro de Acceso) –lateral y el búfer de traducción lateral se borran de las entradas.

6. Si el reinicio es provocado por la activación de la tecla de carga normal en cualquier CPU en la configuración, se producen las siguientes acciones:

a. Cuando la funcionalidad de CZAM no está instalada, el modo arquitectónico de la CPU (y de todas las otras CPU en la configuración debido al reinicio de la CPU inicial o los reinicios de la CPU realizados por ellos) se cambia desde el modo z/Architecture hasta el modo ESA/390. Si la funcionalidad de CZAM está instalada, el modo arquitectónico de la CPU (y de todas las otras CPU en la configuración debido al reinicio de la CPU inicial o los reinicios de la CPU realizados por ellos) se establece en el modo z/Architecture.

b. Cuando la funcionalidad de CZAM no está instalada, la PSW actual se guarda para su uso posterior por una orden Establecer Arquitectura de Procesador de Señal que restaura el modo z/Architecture.

c. Cuando la funcionalidad de CZAM no está instalada, la PSW actual se cambia desde 16 bytes hasta 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 a uno y los bits 33-63 se establecen iguales a los bits 97-127 del PSW de 16 bytes.

Un reinicio de la CPU provocado por la activación de la tecla de reinicio normal del sistema o por el orden de reinicio de la CPU del procesador de señal, y cualquier reinicio de la CPU en el modo ESA/390, no afecta el registro z/Architecture-PSW capturado (es decir, un PSW guardado la última vez que la CPU pasó del modo z/Architecture al modo ESA/390 debido a una orden Establecer Arquitectura con código 0 o un reinicio de la CPU debido a la activación de la tecla de carga normal).

7. La CPU se coloca en el estado detenido después de que se hayan completado las acciones 1-6. Cuando la secuencia IPL de tipo CCW sigue la función de reinicio en esa CPU, la CPU ingresa al estado de carga al finalizar la función de reinicio y no necesariamente ingresa al estado detenido durante la ejecución de la operación de reinicio. Cuando la secuencia de IPL dirigida por la lista sigue la función de reinicio en esa CPU, la CPU ingresa al estado operativo y no necesariamente ingresa al estado detenido durante la ejecución de la operación de reinicio.

Los registros, el contenido de almacenamiento y el estado de las condiciones externas a la CPU no se modifican con el reinicio de la CPU. Sin embargo, el contenido posterior del registro, la ubicación o el estado son impredecibles si hay una operación en curso que cambia el contenido en el momento del reinicio. Un bloqueo mantenido por la CPU cuando se ejecuta la OPERACIÓN DE REALIZACIÓN DE BLOQUEO no se libera por el reinicio de la CPU.

## ES 2 746 479 T3

Cuando se inicia la función de reinicio en la CPU 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 del canal puede o no completarse, y el estado resultante de la funcionalidad del subsistema de canal asociado puede ser impredecible.

### 5 Notas de programación:

1. La mayoría de las operaciones que cambiarían un estado, una condición o el contenido de un campo no pueden ocurrir cuando la CPU está en estado detenido. Sin embargo, algunas funciones del procesador de señal 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, se debe detener la CPU y no se deben ejecutar funciones del operador en progreso.

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

### 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, PSW de z/Architecture capturada, temporizador de la CPU, comparador de reloj, prefijo, control de dirección de evento de interrupción, control de punto de flotación y registros programables de hora del día (TOD) Si la funcionalidad de CZAM no está instalada, el reinicio de la CPU inicial establece el modo arquitectónico en el modo ESA/390 si es provocado por la activación de la tecla de carga normal. Cuando se instala la funcionalidad de CZAM, el reinicio de la CPU inicial establece el modo arquitectónico en el modo z/Architecture si es provocado por la activación de la tecla 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:

1. Cuando la funcionalidad de CZAM no está instalada, si el reinicio es provocado por la activación de la tecla de carga normal, el modo arquitectónico de la CPU (y de todas las otras CPU en la configuración) se establece en el modo ESA/390. De lo contrario, si la funcionalidad de CZAM está instalada, el modo arquitectónico de la CPU (y de todas las otras CPU en la configuración) se establece en el modo z/Architecture.

2. Los contenidos del PSW actual, z/Architecture-PSW capturado, prefijo, temporizador de CPU, comparador de reloj y registro programable TOD se establecen 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 iniciales de z/Architecture. Los 64 bits de los registros de control se establecen independientemente de si la CPU está en el modo arquitectónico ESA/390 o z/Architecture.

4. Los contenidos del registro de control de punto de flotación se establece en cero.

5. Los contenidos del registro de dirección de evento de ruptura se inicializa en 0000000000000001 hexadecimal.

Estas funciones de borrado e inicialización incluyen validación.

Establecer el PSW actual a cero cuando la CPU está en el modo arquitectónico ESA/390 al final de la operación hace que el PSW sea inválido, ya que el bit 12 de PSW será uno en ese modo. Por lo tanto, en este caso, si la CPU se coloca 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 flotante, así como para invocar el reinicio del sistema de I/O.

### Reinicio de borrado

El reinicio de borrado hace que se realice el reinicio de la CPU inicial y el reinicio del subsistema y, adicionalmente, borra o inicializa todas las ubicaciones de almacenamiento y registros en todas las CPU en la configuración, con la excepción del reloj TOD. Dicho borrado es útil para depurar programas y para garantizar la privacidad del usuario. El reinicio de borrado también libera todos los bloqueos utilizados por la instrucción OPERACIÓN DE REALIZACIÓN DE BLOQUEO. Si la funcionalidad de CZAM no está instalada, el reinicio de borrado establece el modo arquitectónico en el modo ESA/390. Cuando se instala la funcionalidad de CZAM, el reinicio de borrado establece el modo arquitectónico en el modo z/Architecture. El borrado no afecta el almacenamiento externo, tal como los dispositivos de almacenamiento de acceso directo utilizados por el programa de control para preservar el contenido de páginas no direccionables.

El reinicio de borrado combina la función de reinicio de la CPU inicial con una función de inicialización que provoca las siguientes acciones:

- 5 1. Cuando la funcionalidad de CZAM no está instalada, el modo arquitectónico de todas las CPU en la configuración se establece en el modo ESA/390. Si la funcionalidad de CZAM está instalada, el modo arquitectónico de todas las CPU en la configuración se establece en el modo z/Architecture.
- 10 2. Los registros de acceso, general y de punto de flotación 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 ESA/390 o z/Architecture cuando se inició la función de reinicio de borrado.
- 15 3. Los contenidos del almacenamiento principal en la configuración y las claves de almacenamiento asociadas se establecen en cero con un código válido de bloque de verificación.
- 20 4. Se liberan los bloqueos utilizados por cualquier CPU en la configuración al ejecutar la instrucción OPERACIÓN DE REALIZACIÓN DE BLOQUEO.
5. Se realiza un reinicio del subsistema.

La validación se incluye en la configuración de registros y en el borrado del almacenamiento y claves de almacenamiento.

Notas de programación:

- 25 1. El modo arquitectónico no se cambia mediante la activación de la tecla normal de reinicio del sistema o mediante la ejecución de una orden de reinicio de la CPU del Procesador de Señal o de reinicio inicial de CPU. Todas las CPU en la configuración están en el mismo modo arquitectónico.
- 30 2. Para que la operación de reinicio de la CPU no afecte los contenidos de los campos que se dejarán sin cambios, la CPU no debe ejecutar instrucciones y se debe deshabilitar para todas las interrupciones en el momento del reinicio. Excepto por el funcionamiento del temporizador de la CPU y la posibilidad de que ocurra una interrupción de la verificación de la máquina, toda la actividad de la CPU se puede detener al colocar la CPU en el estado de espera y deshabilitándola para I/O e interrupciones externas. Para evitar la posibilidad de provocar un reinicio en el momento en que se actualiza el temporizador de la CPU o se produce una interrupción de la verificación de la máquina, la CPU debe estar en el estado detenido.
- 35 3. El reinicio de la CPU, el reinicio de la CPU inicial, el reinicio del subsistema y el reinicio de borrado no afectan el valor y el estado del reloj TOD.
- 40 4. Las condiciones bajo las cuales la CPU ingresa al estado de detención de verificación dependen del modelo e incluyen fallas que impiden la finalización de la operación actual. Por lo tanto, si el reinicio de la CPU o el reinicio de la CPU inicial se ejecuta mientras la CPU se encuentra en el estado de verificación y detención, los contenidos de la PSW, los registros y las ubicaciones de almacenamiento, que incluyen las claves de almacenamiento y la ubicación de almacenamiento accedida en el momento del error, pueden tener valores impredecibles y, en algunos casos, los contenidos aún pueden estar en error después de que el estado de verificación y detención se borra por estos reinicios. En esta situación, se requiere un reinicio de borrado para borrar el error.

Reinicio de encendido

- 50 La función de reinicio de encendido de un componente de la máquina se realiza como parte de la secuencia de encendido de ese componente. Las secuencias de encendido para el reloj TOD, el almacenamiento principal, el almacenamiento expandido y el subsistema de canales pueden incluirse como parte de la secuencia de encendido de la CPU, o la secuencia de encendido para estas unidades puede iniciarse por separado.

- 55 Reinicio del encendido de la CPU: el reinicio del encendido provoca que se realice el reinicio de la CPU inicial y puede o no provocar el reinicio del sistema de I/O en el subsistema del canal. Los contenidos de los registros generales, los registros de acceso y los registros de punto de flotación se borran a ceros con un código válido de bloque de verificación. Los bloqueos utilizados por OPERACIÓN DE REALIZACIÓN DE BLOQUEO y asociados con la CPU se liberan a menos que sean mantenidos por una CPU ya encendida. Si la funcionalidad de CZAM no está instalada y el reinicio está asociado con el establecimiento de una configuración, la CPU se coloca en el modo ESA/390; de lo contrario, la CPU se coloca en el modo arquitectónico de las CPU que ya están en la configuración. Si la funcionalidad de CZAM está instalada, la CPU se coloca en el modo z/Architecture.
- 60

## ES 2 746 479 T3

El reinicio de la CPU, el reinicio de la CPU inicial, el reinicio del subsistema y el reinicio de borrado pueden iniciarse manualmente al utilizar las funcionalidades del operador. El reinicio de la CPU inicial es parte de la función de arranque del programa. El reinicio de encendido se realiza como parte del encendido.

5 Cuando la funcionalidad de CZAM no está instalada, si se inicia el reinicio mediante la tecla borrado de reinicio de sistema, normal de carga o borrado de carga o mediante un reinicio de encendido de CPU que establece la configuración, el modo arquitectónico se establece en Modo ESA/390; de lo contrario, el modo arquitectónico no cambia, excepto que el reinicio de encendido establece el modo al de las CPU que ya están en la configuración. Si la funcionalidad de CZAM está instalada, el modo arquitectónico se establece en el modo z/Architecture.

10 A continuación, se describen otros procesos, operaciones y/o comportamientos que pueden cambiar debido a la instalación de una funcionalidad de modo arquitectónico de configuración:

15 (5) Suprime otras acciones relacionadas con el reinicio que se toman para facilitar el cambio entre el modo ESA/390 y z/Architecture, cuando se realiza el reinicio. Cuando la funcionalidad de CZAM no está instalada, la PSW actual se guarda para su uso posterior por una orden Establecer Arquitectura de Procesador de Señal que restaura el modo z/Architecture. Cuando la funcionalidad 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 se instala la funcionalidad de CZAM, la PSW no se guarda para su posterior uso por una orden Establecer Arquitectura de Procesador de Señal que restaura el modo z/Architecture, y la PSW actual no se cambia de 16 bytes a 8 bytes.

20 (6) Cambia el proceso para configurar una CPU con un comando configure CPU SCLP (Procesador Lógico de Llamada de Servicio), y cargar las operaciones clave. Más bien que configurar en ESA/390, configurar en el modo definido por reset. El comando configure CPU SCLP coloca a la CPU en cuestión en el modo arquitectónico de las CPU que ya están en el estado configurado. Al menos la primera CPU colocada en una configuración se coloca allí junto con un reinicio de energía de la CPU y, como parte de ese reinicio, se coloca en el modo de arquitectura definido en el reinicio de encendido de la CPU. Un modelo puede establecer alternativamente el modo de las CPU que están en estado de espera cuando establece el modo de las CPU configuradas.

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

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

Una realización de una instrucción del Procesador de Señal (SIGP) se describe con referencia a la Figura 8A. En una realización, una instrucción 800 de Procesador de Señal tiene una pluralidad de campos, que incluyen, por ejemplo, un campo 802 de código de operación (código de operación) que tiene un código de operación que indica una operación de procesador de señal; un primer campo 804 de registro (R<sub>1</sub>); un segundo campo 806 de registro (R<sub>3</sub>); un campo 808 base (B<sub>2</sub>); y un campo 810 de desplazamiento (D<sub>2</sub>). R<sub>1</sub> designa un registro general, cuyo contenido es el primer operando; R<sub>3</sub> designa un registro general, cuyo contenido es el tercer operando; y el contenido de un registro designado por R<sub>2</sub> se agrega al desplazamiento en D<sub>2</sub> para proporcionar una dirección de un segundo operando.

40 En operación, un código de orden de ocho bits y, si se solicita, 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 puede detallarse mediante el estado ensamblado en las posiciones de bit 32-63 de la ubicación del primer operando.

45 La dirección del segundo operando no se utiliza para direccionar datos; en cambio, 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 que debe realizar la CPU direccionada. La asignación y definición de códigos de orden incluyen, por ejemplo, lo siguiente, en un ejemplo:

| Código (Dec) | (Hex) | Orden               |
|--------------|-------|---------------------|
| 0            | 00    | No asignado         |
| 1            | 01    | Sentido             |
| 2            | 02    | Llamada externa     |
| 3            | 03    | Señal de emergencia |
| 4            | 04    | Inicio              |
| 5            | 05    | Detención           |
| 6            | 06    | Reinicio            |
| 7            | 07    | No asignado         |

## ES 2 746 479 T3

| Código<br>(Dec) | (Hex) | Orden                                 |
|-----------------|-------|---------------------------------------|
| 8               | 08    | No asignado                           |
| 9               | 09    | Estado de detención y almacenamiento  |
| 10              | 0A    | No asignado                           |
| 11              | 0B    | reinicio de la CPU inicial            |
| 12              | 0C    | Reinicio de la CPU                    |
| 13              | 0D    | Prefijo de Conjunto                   |
| 14              | 0R    | Estado de almacenamiento en dirección |
| 15-17           | 0F-11 | No asignado                           |
| 18              | 12    | Establecer Arquitectura               |
| 19              | 13    | Señal de emergencia condicional       |
| 14              | 14    | No asignado                           |
| 21              | 15    | Estado de ejecución de detección      |
| 22-255          | 16-FF | No asignado                           |

5 El número binario de 16 bits contenido en las posiciones de bit 48-63 del registro general R<sub>3</sub> forma la dirección de la CPU. Los bits 0-47 del registro se ignoran. Cuando el orden especificado es la orden Establecer Arquitectura, la dirección de la CPU se ignora; todas las demás CPU en la configuración se consideran direccionadas.

El registro general que contiene el parámetro de 32 bits en las posiciones de bit 32-63 es R<sub>1</sub> o R<sub>1</sub> + 1, el que sea el registro impar. Depende del código de orden si se proporciona un parámetro y para qué se utiliza.

10 Los operandos recién descritos tienen los siguientes formatos, en un ejemplo:

Registro general designado por R<sub>1</sub>: bits 0-31 sin utilizar;  
los bits 32-63 incluyen estado;

15 Registro general designado por R<sub>1</sub> o R<sub>1</sub> + 1, cualquiera de los cuales es el registro impar: Bits 0-31 sin utilizar;  
los bits 32-63 incluyen el parámetro;

Registro general designado por R<sub>3</sub>: bits 0-48 sin utilizar;  
los bits 49-63 incluyen la dirección de la CPU;

Dirección del segundo operando: bits 0-55 sin utilizar; los bits 56-63 incluyen el código de orden.

20 Se realiza una función de serialización antes de que comience la operación y nuevamente después de que se completa la operación.

25 Cuando se acepta el código de orden y no se devuelve ningún estado distinto de cero, se establece el código de condición 0. Cuando se genera la información de estado por esta CPU (la CPU que realiza el SIGP) o la CPU direccionada la devuelve, el estado se coloca en las posiciones de bit 32-63 del registro general R<sub>1</sub>, los bits 0-31 del registro permanecen sin cambios y se establece el código 1 de condición.

30 Cuando la ruta de acceso a la CPU direccionada está ocupada, o la CPU direccionada está operativa, pero en un estado en el que no puede responder al código de orden, se establece el código 2 de condición.

35 Cuando la CPU direccionada no está operativa (es decir, no se proporciona en la instalación, no está en la configuración, se encuentra en alguno de los modos de prueba del ingeniero y cliente o está apagado), se establece el código 3 de condición.

Código de condición resultante:

- 0 Código de orden aceptado
- 1 Estado almacenado
- 2 Ocupado
- 3 No operacional

Excepciones del programa:

- Operación privilegiada
- Restricción transaccional

50 Cuando se especifica el orden De Procesador de Señal de Establecer Arquitectura en las posiciones de bits 56-63 de la segunda dirección de operando de la instrucción del Procesador de Señales, el contenido de las posiciones de bits 56-63 del registro de parámetros se utiliza como un código que especifica un modo arquitectónico en el que todos deben configurar las CPU en la configuración: el código 0 especifica el modo ESA/390, y los códigos 1 y 2 especifican el modo z/Architecture. El código 1 especifica que, para cada una de las CPU de la configuración, el



## ES 2 746 479 T3

ESA/390 PSW actual se transformará en un PSW z/Architecture. El código 2 especifica que la PSW de la CPU que ejecuta el procesador de señal se transformará en una PSW z/Architecture y que, para cada una de las otras CPU en la configuración, la PSW se debe establecer con el valor del registro z/Architecture-PSW capturado para esa CPU. La configuración de la PSW con el valor del registro capturado-z/Architecture- PSW restaurará la PSW que existía cuando la CPU estuvo por última vez en el modo z/Architecture, siempre que el registro capturado-z/Architecture-PSW no haya sido establecido en todos los ceros mediante un reinicio.

Los bits 0-55 del registro de parámetros se ignoran. Se ignoran los contenidos del registro de dirección de CPU de la instrucción del procesador de señal; todas las demás CPU en la configuración se consideran direccionadas.

Cuando la funcionalidad de CZAM no está instalada, la orden se acepta solo si el código es 0, 1 o 2, la CPU no está en el modo especificado por el código, cada una de las otras CPU está detenida o comprobada -stop, y ninguna otra condición impide aceptar la orden.

Cuando se instala la funcionalidad de CZAM, el código 0 no se acepta porque no se permite volver al modo ESA/390, y dado que la CPU ya está en el modo arquitectónico z/Architecture, la especificación de los códigos 1 y 2 da como resultado una finalización que indica parámetro inválido y código 1 de condición. Las otras condiciones de requisitos previos normalmente verificadas por la orden Establecer Arquitectura pueden o no verificarse.

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

La orden Establecer Arquitectura se completa, como sigue, 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 cambio, el bit 55 (parámetro no válido) del registro general designado por el campo R1 de la instrucción del Procesador de Señal se establece en uno, y se establece el código 1 de condición.

- Si no es cierto que todas las otras CPU de la configuración están en estado detenido o de verificación de detención, la orden no se acepta. En cambio, el bit 54 (estado incorrecto) del registro general designado por el campo R1 de la instrucción del Procesador de Señal se establece en uno, y se establece el código 1 de condición.

- El modo arquitectónico de todas las CPU en la configuración se establece según lo especificado por el código (por ejemplo, el bit 12 de la PSW para controlar las operaciones se establece en el modo arquitectónico especificado, y/u otra indicación en el entorno informático se establece indicando el modo arquitectónico especificado).

- Si el 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 los PSW de 16 bytes se establecen de la siguiente manera: los bits 0-11 y 13-32 se establecen iguales a los mismos bits del PSW de ocho bytes, el bit 12 y los bits 33-96 se establecen en ceros, y los bits 97-127 se establecen igual a los bits 33-63 de la PSW de ocho bytes. También, el bit 19 del prefijo ESA/390, que se convierte en el bit 51 del prefijo z/Architecture, se establece en cero.

Si el código es 2, la PSW de la CPU que ejecuta el procesador de señal y los valores de prefijo de todas las CPU se establecen como en el caso del código 1. Para cada una de las otras CPU en la configuración, la PSW se establece con el valor del registro z/Architecture-PSW capturado. Sin embargo, el registro z/Architecture-PSW capturado se ha establecido en todos los ceros si la CPU realizó un reinicio, que no sea el reinicio de la CPU, ya sea en el momento de la transición del modo arquitectónico o posteriormente.

- Si el orden cambia desde el modo arquitectónico de z/Architecture hasta el ESA/390, entonces, para cada CPU en la configuración, (1) se guarda la PSW actual, que es la PSW actualizada en el caso de la CPU que ejecuta el Procesador de Señal, en el registro z/Architecture-PSW capturado, y (2) la PSW actual de 16 bytes se cambia a una PSW de ocho bytes al establecer los bits de la PSW de ocho bytes de la siguiente manera: bits 0-11 y 13-32 se establecen iguales a los mismos bits del PSW de 16 bytes, el bit 12 se establece en uno y los bits 33-63 se establecen iguales a los bits 97-127 del PSW de 16 bytes. El bit 51 del prefijo z/Architecture, que se convierte en el bit 19 del prefijo ESA/390, permanece sin cambios.

- Los ALB y TLB de todas las CPU en la configuración se borran de su contenido.

- Se realiza una función de serialización y sincronización de punto de control en todas las CPU de la configuración.

Si el orden cambia el modo arquitectónico desde z/Architecture hasta ESA/390 y la instrucción del Procesador de Señal provoca la ocurrencia de un evento PER que extrae instrucciones, solo los 31 bits más a la derecha de la dirección de la instrucción se almacenan en el campo de direcciones PER de ESA/390.

- 5 En una realización, con CZAM, lo siguiente es un prerrequisito: cada una de las otras CPU está en el estado detenido o en el de verificación y parada, y ninguna otra condición impide aceptar la orden. Cuando se instala la funcionalidad de CZAM, el código 0 no se acepta porque no se permite volver al modo ESA/390, y dado que la CPU ya está en el modo arquitectónico z/Architecture, la especificación de los códigos 1 y 2 da como resultado una finalización que indica un parámetro no válido y un código 1 de condición. Las otras condiciones de requisitos previos normalmente verificadas por la orden Establecer Arquitectura pueden o no verificarse. En aún otra realización, SIGP con el código 1 y 2 indica la finalización exitosa sin indicación adicional.
- 10 Una realización del procesamiento asociado con la ejecución de una instrucción SIGP para un código de orden Establecer Arquitectura se describe con referencia a la Figura 8B. Con referencia a la Figura 8B, un procesador del entorno informático ejecuta una instrucción SIGP y obtiene un código de orden que indica una operación Establecer Arquitectura, ETAPA 850. En un ejemplo, el código de orden se incluye en la dirección del segundo operando de la instrucción SIGP.
- 15 Adicionalmente, el modo arquitectónico solicitado para conmutar se obtiene, por ejemplo, del registro de parámetros especificado por la instrucción SIGP, ETAPA 852. Adicionalmente, se determina si está instalada una funcionalidad de modo arquitectónico de configuración, tal como CZAM, PREGUNTA 854. Esto está determinado, en un ejemplo, por un indicador de funcionalidad.
- 20 Si CZAM no está instalado, entonces se realiza una determinación adicional en cuanto a si la CPU ya está en el modo arquitectónico solicitado, PREGUNTA 856. Si es así, entonces se proporciona el estado, por ejemplo, en un registro designado por la instrucción SIGP, ETAPA 858, y el estado se trata como un error, ETAPA 860. Sin embargo, si la CPU no está en el modo solicitado, PREGUNTA 856, entonces se realiza una determinación si se cumplen otras condiciones especificadas por la instrucción, tales como si las otras CPU del entorno informático están configuradas en un estado detenido, etc., PREGUNTA 862. Si no se cumplen las condiciones, el procesamiento continúa hasta la ETAPA 858. De lo contrario, se acepta la orden, ETAPA 864, y se debe cambiar el modo arquitectónico. Por lo tanto, el PSW se establece, como se describió anteriormente, ETAPA 866, y el procesamiento para este aspecto de la instrucción finaliza, ETAPA 868.
- 25 Si CZAM no está instalado, entonces se realiza una determinación adicional en cuanto a si la CPU ya está en el modo arquitectónico solicitado, PREGUNTA 856. Si es así, entonces se proporciona el estado, por ejemplo, en un registro designado por la instrucción SIGP, ETAPA 858, y el estado se trata como un error, ETAPA 860. Sin embargo, si la CPU no está en el modo solicitado, PREGUNTA 856, entonces se realiza una determinación si se cumplen otras condiciones especificadas por la instrucción, tales como si las otras CPU del entorno informático están configuradas en un estado detenido, etc., PREGUNTA 862. Si no se cumplen las condiciones, el procesamiento continúa hasta la ETAPA 858. De lo contrario, se acepta la orden, ETAPA 864, y se debe cambiar el modo arquitectónico. Por lo tanto, el PSW se establece, como se describió anteriormente, ETAPA 866, y el procesamiento para este aspecto de la instrucción finaliza, ETAPA 868.
- 30 Volviendo a la PREGUNTA 854, si CZAM está instalado, entonces se determina si la CPU está en el modo solicitado, la PREGUNTA 870. Si la CPU ya está en el modo solicitado, entonces, en un ejemplo, se proporciona el estado de que la CPU ya está en el modo arquitectónico solicitado (por ejemplo, z/Architecture), ETAPA 872. Sin embargo, en esta realización, este estado es aceptable y no se trata como un error, ETAPA 874. O bien se ignora, o en otra realización, se puede proporcionar un código de condición que sea un código sin error. En aún otra realización, el estado simplemente indica la finalización exitosa. También existen otras posibilidades para indicar que no hay errores a pesar de que la CPU ya está en el modo arquitectónico solicitado.
- 35 Volviendo a la PREGUNTA 870, si, sin embargo, la CPU no está en el modo solicitado, entonces la orden no se acepta, ya que es ilegal volver al modo arquitectónico único (por ejemplo, ESA/390), ETAPA 876. Se proporciona el estado, ETAPA 878, que se considera un error, ETAPA 880.
- 40 En una realización, cuando CZAM está en un sistema como una funcionalidad no seleccionable, entonces se puede omitir la PREGUNTA 854 y el control puede pasar desde la ETAPA 852 directamente hasta la ETAPA 870. En dicha realización, las etapas 854 a 868 no se pueden implementar.
- 45 En otra realización, cuando se recibe una orden para conmutar al modo de arquitectura actual, la orden puede no ser aceptada y se puede indicar un error en la ETAPA 874.
- 50 Otros comportamientos, procesos y/u operaciones que pueden cambiar con base en la instalación de un CAM incluyen:
- (8) Cambios en los bits de funcionalidad: se agrega un nuevo bit, por ejemplo, el bit 138, a los bits de funcionalidad para indicar la funcionalidad del Modo Arquitectónico z/Architecture de Configuración, y el bit 2, que indica si el modo arquitectónico z/Architecture está activo, se establece en uno (que indica activo).
- 55 Como se describió anteriormente, un entorno informático se puede arrancar en un modo arquitectónico (por ejemplo, ESA/390) y luego conmutarse a otro modo arquitectónico (por ejemplo, z/Architecture) para su procesamiento; o un entorno informático puede arrancarse directamente en el modo arquitectónico en el que se va a procesar (por ejemplo, z/Architecture). Por lo tanto, de acuerdo con uno o más aspectos, se proporcionan programas utilitarios de control, que incluyen, pero no se limitan a, el sistema operativo u otros programas utilitarios independientes que no necesitan el sistema operativo para arrancar, que se pueden arrancar en una primera configuración arquitectura en la que el programa utilitario de control se inicializa en un modo arquitectónico (por ejemplo, ESA/390) y luego se conmuta a otro modo arquitectónico (por ejemplo, z/Architecture); o una segunda configuración arquitectónica en la que el programa utilitario de control se inicializa y procesa en un modo arquitectónico seleccionado (por ejemplo, z/Architecture). En una realización, el modo arquitectónico seleccionado es la arquitectura en la que se escribe el programa utilitario de control.
- 60 Como se describió anteriormente, un entorno informático se puede arrancar en un modo arquitectónico (por ejemplo, ESA/390) y luego conmutarse a otro modo arquitectónico (por ejemplo, z/Architecture) para su procesamiento; o un entorno informático puede arrancarse directamente en el modo arquitectónico en el que se va a procesar (por ejemplo, z/Architecture). Por lo tanto, de acuerdo con uno o más aspectos, se proporcionan programas utilitarios de control, que incluyen, pero no se limitan a, el sistema operativo u otros programas utilitarios independientes que no necesitan el sistema operativo para arrancar, que se pueden arrancar en una primera configuración arquitectura en la que el programa utilitario de control se inicializa en un modo arquitectónico (por ejemplo, ESA/390) y luego se conmuta a otro modo arquitectónico (por ejemplo, z/Architecture); o una segunda configuración arquitectónica en la que el programa utilitario de control se inicializa y procesa en un modo arquitectónico seleccionado (por ejemplo, z/Architecture). En una realización, el modo arquitectónico seleccionado es la arquitectura en la que se escribe el programa utilitario de control.
- 65 Como se describió anteriormente, un entorno informático se puede arrancar en un modo arquitectónico (por ejemplo, ESA/390) y luego conmutarse a otro modo arquitectónico (por ejemplo, z/Architecture) para su procesamiento; o un entorno informático puede arrancarse directamente en el modo arquitectónico en el que se va a procesar (por ejemplo, z/Architecture). Por lo tanto, de acuerdo con uno o más aspectos, se proporcionan programas utilitarios de control, que incluyen, pero no se limitan a, el sistema operativo u otros programas utilitarios independientes que no necesitan el sistema operativo para arrancar, que se pueden arrancar en una primera configuración arquitectura en la que el programa utilitario de control se inicializa en un modo arquitectónico (por ejemplo, ESA/390) y luego se conmuta a otro modo arquitectónico (por ejemplo, z/Architecture); o una segunda configuración arquitectónica en la que el programa utilitario de control se inicializa y procesa en un modo arquitectónico seleccionado (por ejemplo, z/Architecture). En una realización, el modo arquitectónico seleccionado es la arquitectura en la que se escribe el programa utilitario de control.

Una realización de la lógica asociada con el arranque de un programa utilitario de control en la primera configuración arquitectónica se describe con referencia a la Figura 9) Inicialmente, un procesador (por ejemplo, hardware) establece una palabra de estado del programa (PSW) que se utilizará para controlar la ejecución, ETAPA 900. En este ejemplo, el hardware carga el PSW desde la memoria y el PSW tiene un formato del modo arquitectónico único, por ejemplo, ESA/390. El código de configuración luego se realiza en el modo arquitectónico, ETAPA 904. Por ejemplo, el programa utilitario de control comienza a ejecutar un primer conjunto de instrucciones especificadas por la palabra de estado del programa. Posteriormente, en una realización, se realiza la preparación para conmutar desde un modo arquitectónico, ESA/390, hasta otro modo arquitectónico, tal como z/Architecture, ETAPA 904. Por ejemplo, el programa utilitario de control se prepara para conmutar de modo al configurar un registro con un código de orden (por ejemplo, código de orden Establecer Arquitectura) y pasar ese registro a una instrucción conocida como la instrucción Procesador de Señal.

El procesador ejecuta la instrucción del Procesador de Señal para colocar el procesador y, por lo tanto, el programa utilitario de control, en modo z/Architecture, ETAPA 906. Posteriormente, en una realización, se hace una determinación de si la instrucción del Procesador de Señal se ejecutó con éxito, PREGUNTA 908. Si hubo una condición de error del procesador de señal, entonces se cancela el arranque del programa utilitario de control, ETAPA 910; de lo contrario, el código restante se realiza en modo z/Architecture, ETAPA 912.

En una realización adicional, el programa utilitario de control, que se escribe para una arquitectura seleccionada (por ejemplo, z/Architecture), se arranca directamente en esa arquitectura en lugar de arrancarse en una arquitectura y modos de conmutación. Una referencia de la lógica para arrancar un programa utilitario de control directamente en un sistema que no requiere un conmutador, tal como un sistema que incluye la funcionalidad de CZAM, se describe con referencia a la Figura 10. Con referencia a la Figura 10, inicialmente, el procesador (por ejemplo, hardware) establece la palabra de estado del programa desde el almacenamiento, ETAPA 1000. En un ejemplo, se crea una nueva palabra de estado de programa (PSW) de 16 bytes a partir de ubicaciones de almacenamiento absoluto seleccionadas (por ejemplo, ubicaciones 0-7). El nuevo PSW de 16 bytes se forma, por ejemplo, a partir de los contenidos de la palabra doble de almacenamiento seleccionada (por ejemplo, ubicaciones 0-7). El bit 12 de la palabra doble es uno; de lo contrario, se puede indicar un error. (El error puede ser una excepción de especificación reconocida, una verificación de la máquina u 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 establecen en ceros. Las posiciones de bit 97-127 del PSW recién creado se inicializan a partir de los bits 33-63 de la palabra doble seleccionada.

A partir de entonces, el código de configuración se realiza en modo z/Architecture, ya que el sistema ya está en ese modo, ETAPA 1002. Adicionalmente, el resto del código también se realiza en modo z/Architecture, ETAPA 1004.

Aunque las secuencias de arranque anteriores se pueden utilizar por separado dependiendo de la configuración arquitectónica (por ejemplo, la primera configuración o la segunda configuración), en una realización, se proporciona una secuencia de arranque que se puede utilizar independientemente del modo arquitectónico en el que se encuentre el programa utilitario de control que está siendo arrancado. Es decir, se proporciona una secuencia de arranque común que puede utilizarse en una pluralidad de configuraciones arquitectónicas, que incluyen, pero no se limitan a, la primera configuración arquitectónica y la segunda configuración arquitectónica.

Una realización de la lógica asociada con una secuencia de arranque común se describe con referencia a la Figura 11) Inicialmente, el procesador (por ejemplo, hardware) establece la palabra de estado del programa desde el almacenamiento, ETAPA 1100. En esta realización, el modo arquitectónico puede ser cualquiera de una pluralidad de modos, que incluye un modo heredado, tal como ESA/390, o un modo de arquitectura mejorada, tal como z/Architecture. Por lo tanto, la palabra de estado del programa refleja el modo arquitectónico inicial del procesador.

Posteriormente, se puede ejecutar un subconjunto común de instrucciones para realizar una cierta cantidad de configuración, la ETAPA 1102. Estas instrucciones son igualmente aplicables a los diferentes modos arquitectónicos, aunque pueden comportarse de manera diferente según el modo. Son comunes porque se definen en las diferentes arquitecturas y se pueden realizar independientemente del modo arquitectónico actual.

Además de y/o como parte de la ejecución del subconjunto común de instrucciones para realizar alguna configuración, se realiza una conmutación de modo, independientemente del modo arquitectónico actual, ETAPA 1104. Como se indicó anteriormente, se coloca un código de orden en un registro y se reenvía a la instrucción SIGP. La instrucción SIGP se ejecuta para colocar el entorno en el modo z/Architecture, ETAPA 1106. En particular, en una realización, si el entorno está en ESA/390, entonces la ejecución de la instrucción SIGP coloca el entorno en modo z/Architecture. Sin embargo, si el entorno ya está en modo z/Architecture, entonces la ejecución de la instrucción SIGP no realiza un cambio de modo y, en cambio, proporciona el estado de que el entorno ya está en modo z/Architecture. Este estado se acepta como no error.

El resto del código se realiza en modo z/Architecture, ETAPA 1108.

Un ejemplo de una secuencia de arranque simple puede incluir, por ejemplo, una o más instrucciones de control de carga (LCTL) para inicializar registros de control, cargar un registro con un registro de comando para realizar una instrucción SIGP y una instrucción SIGP. Estas instrucciones pueden emitirse en modo ESA/390 o z/Architecture con resultados "similares", de modo que después de completar SIGP el sistema se haya configurado inicialmente y esté en modo z/Architecture. En al menos una realización, el código puede verificar el código de condición después de completar la instrucción SIGP para determinar si el sistema se configuró inicialmente en modo ESA/390 (por ejemplo, mediante un valor de código de condición de 0 que indica una conmutación exitosa de ESA/390 se realizó) o z/Architecture (por ejemplo, mediante un código de condición de 1 que indica que no se realizó ningún cambio porque el sistema ya estaba en el modo arquitectónico solicitado).

Otra realización del uso de una secuencia de arranque común para arrancar un programa utilitario de control se describe con referencia a la Figura 12) En este ejemplo, el procesador (por ejemplo, hardware) establece el PSW del almacenamiento en el modo arquitectónico que el procesador está procesando inicialmente, como se describió anteriormente, ETAPA 1200. Luego, el bit 12 del PSW se verifica para determinar el modo arquitectónico del entorno informático, PREGUNTA 1202. En una realización, para verificar el bit PSW, se utiliza una instrucción PSW de Extracto. Un ejemplo de una instrucción PSW de Extracto se describe con referencia a la Figura 13)

Con referencia a la Figura 13, en una realización, una instrucción 1300 de PSW de Extracto incluye un código 1302 de operación que incluye un código de operación para indicar una operación PSW de Extracto; un primer campo 1304 de registro (R<sub>1</sub>); y un segundo campo 1306 de registro (R<sub>2</sub>). En operación, los bits 0-31 del PSW actual se colocan en las posiciones de bits 32-63 del primer operando (registro especificado por el primer campo de registro), y los bits 0-31 del operando permanecen sin cambios. Posteriormente, los bits 32-63 de la PSW actual se colocan en las posiciones de bits 32-63 del segundo operando (registro especificado por el segundo campo de registro), y los bits 0-31 del operando permanecen sin cambios. La acción asociada con el segundo operando no se realiza, en un ejemplo, si el campo R<sub>2</sub> es cero.

Volviendo a la Figura 12, si el modo de arquitectura es, por ejemplo, ESA/390 (por ejemplo, PSW bit 12 = 1), entonces el código de configuración se realiza en ESA/390, ETAPA 1204. El procesador se prepara para la conmutación de modo, ETAPA 1206, y la instrucción SIGP se utiliza para cambiar el modo al modo z/Architecture, ETAPA 1208. Adicionalmente, se hace una determinación de si la instrucción SIGP se completó con éxito, ETAPA 1210. Si no se completó con éxito, se presenta un error, ETAPA 1214. De lo contrario, el resto del código se realiza en modo de arquitectura z/Architecture, ETAPA 1216.

Volviendo a la PREGUNTA 1202, si la comprobación del bit 12 del PSW indica que el entorno informático está en modo z/Architecture (por ejemplo, PSW bit 12 = 0), entonces el código de configuración se realiza en modo z/Architecture, ETAPA 1215, y el procesamiento continúa con la ejecución del resto del código en z/Architecture, ETAPA 1216.

Una realización aún adicional del uso de una secuencia de arranque común se describe con referencia a la Figura 14). En este ejemplo, el procesador (por ejemplo, Hardware) establece el PSW del almacenamiento en función del modo arquitectónico en el que el procesador está procesando inicialmente, como se describió anteriormente, ETAPA 1400, y el código de configuración se realiza utilizando un subconjunto común de instrucciones (por ejemplo, una pequeña cantidad de instrucciones, tales como 1-5 instrucciones), ETAPA 1402. El procesador se prepara para una conmutación de modo, ETAPA 1404, y la instrucción SIGP se ejecuta para cambiar al modo z/Architecture, ETAPA 1406.

Se hace una determinación de si la instrucción SIGP se completó con éxito, SOLICITUD 1408. Si tuvo éxito, entonces se realiza parte del código de z/Architecture en respuesta al arranque en modo ESA/390, ETAPA 1410, y el resto del código se realiza en z/Architecture, ETAPA 1412. Por ejemplo, se puede guardar un indicador que indica que se debe realizar un reinicio en modo ESA/390, cuando se va a realizar un reinicio de software. En otro aspecto, los modelos que arrancan en el modo ESA/390 pueden tener algunos registros de control inicializados de manera diferente a los modelos que inician en el modo z/Architecture, y el código que responde a haber arrancado en el modo ESA/390 puede reconfigurar los controles para que correspondan a las configuraciones preferidas para z/Architecture.

Volviendo a la PREGUNTA 1408, si la instrucción SIGP no realiza una conmutación porque el entorno informático ya está en modo z/Architecture, entonces se ejecuta algún código z/Architecture que responde al inicio en el modo z/Architecture, ETAPA 1414, y el resto del código también se realiza en modo z/Architecture, ETAPA 1412. Por ejemplo, un programador puede guardar un indicador de que se debe realizar un reinicio al modo z/Architecture. En otro aspecto, los modelos que arrancan en modo z/Architecture pueden ofrecer registros de control adicionales que podrían inicializarse en la ETAPA 1414 que serían inapropiados para inicializar si el sistema arrancara en modo ESA/390. Nuevamente, en esta realización, el SIGP puede "fallar" porque no realiza una conmutación, pero no hay condición de error. Es un uso aceptable de SIGP.

Además del arranque anterior de los programas utilitarios de control, también se proporciona el arranque de los programas utilitarios de control debido a los reinicios. En una realización, hay un código que restablece una máquina

completa y emula un reinicio al restablecer el sistema a un modo particular, tal como un modo heredado (por ejemplo, ESA/390), y derivar a un cargador de arranque. Sin embargo, con la funcionalidad de CZAM, esto no ocurre. Por lo tanto, hay un par de otras opciones que incluyen, por ejemplo, determinar si el modelo actual se enciende en un modo heredado (por ejemplo, ESA/390) o un modo de arquitectura mejorada (por ejemplo, z/Architecture), configurando el modo de máquina en un modo correspondiente y derivando al cargador de arranque en los modos de arranque nativos dependiendo del modo de máquina; o volver a arrancar como el modelo actual como un modelo de arranque de z/Architecture independientemente del modo de arranque de hardware nativo, es decir, realizar una reinicialización iniciada por software como si fuera un re arranque de z/Architecture, incluso en sistemas ESA/390. Esto es posible ya que es un re arranque de software, también conocido como re arranque en caliente, en el que el procesador se recicla como si se reiniciara, pero sin salir al disco y volver a cargar un programa utilitario de control completo. Los registros y PSW se inicializan al mismo estado que si se realizara un arranque, y luego se derivan a la primera instrucción de la secuencia de arranque.

Una realización de reinicio se describe con referencia a la Figura 15. Inicialmente, el procesador comienza a salir el entorno, que incluye, por ejemplo, desconectar el I/O, inmovilizar la máquina y, en general, finalizar las operaciones, ETAPA 1500. Posteriormente, se leen los bits de la funcionalidad u otro indicador para determinar si CZAM está instalado, ETAPA 1502. Si CZAM no está instalado, PREGUNTA 1504, entonces el procesador se prepara para conmutar de modo al configurar el registro con el código de orden para la instrucción SIGP, ETAPA 1506, y la instrucción SIGP se realiza para conmutar al modo ESA/390, ETAPA 1508. Posteriormente, se realiza una derivación al comienzo de una secuencia de arranque del programa utilitario de control, cuyos ejemplos se describen anteriormente, ETAPA 1510. El código de arranque se realiza en modo ESA/390 con un cambio al modo z/Architecture.

Volviendo a la PREGUNTA 1504, si CZAM está instalado, entonces no se necesita un conmutador ya que los entornos informáticos ya están en modo z/Architecture y, por lo tanto, se realiza una derivación al inicio de la secuencia de arranque del programa utilitario de control como se describió anteriormente, ETAPA 1514. El código de arranque se realiza en modo z/Architecture, ETAPA 1512.

En una realización en la que no se proporciona un indicador de funcionalidad CZAM, un indicador guardado como parte de las etapas 1410, 1414 se puede utilizar en su lugar para indicar un modo de arquitectura de configuración de arranque.

Se describe una realización adicional de la lógica de reinicio con referencia a la Figura 16. Inicialmente, el procesador comienza a salir el entorno, incluyendo, por ejemplo, desconectar la I/O, detener la máquina y, en general, finalizar las operaciones, ETAPA 1600. Después de esto, se leen los bits de la funcionalidad u otro indicador para determinar si CZAM está instalado, ETAPA 1602. Si CZAM no está instalado, PREGUNTA 1604, entonces el procesador se prepara para conmutar de modo, ETAPA 1606, y la instrucción SIGP se realiza para conmutar al modo ESA/390, ETAPA 1608. Adicionalmente, el PSW se configura para restablecer el PSW, ETAPA 1610. Por ejemplo, el PSW se inicializa a un valor conocido en el reinicio, por ejemplo, al valor almacenado en ubicaciones absolutas 0 a 7 como se describe en este documento, o mediante un valor fijo definido por la especificación de arquitectura. Luego, se realiza una derivación al inicio del código de arranque del programa utilitario de control, como se describió anteriormente, ETAPA 1612. El código de arranque luego se realiza en el modo ESA/390 con una conmutación a z/Architecture, ETAPA 1614.

Volviendo a la PREGUNTA 1604, si CZAM está instalado, entonces no se necesita ninguna conmutación, la PSW se establece en la PSW de reinicio, ETAPA 1616, se realiza una derivación al inicio del código de inicio del programa utilitario de control, como se describió anteriormente, ETAPA 1618 y el código de arranque se realiza en modo z/Architecture, ETAPA 1614.

En aún otra realización de reinicio, el reinicio se realiza como si estuviera en el modo de z/Architecture, como se describe con referencia a la Figura 17. En esta realización, inicialmente el sistema comienza a salir el entorno, ETAPA 1700. Luego, el PSW se establece en el reinicio z/Architecture PSW, ETAPA 1702, y se realiza una derivación al inicio del código de arranque del programa utilitario de control, ETAPA 1704. El código de arranque se realiza en modo z/Architecture independientemente de si la máquina se inicia en ESA/390 o z/Architecture de forma nativa, ETAPA 1706.

En un aspecto adicional, se proporciona lógica de detección para determinar si el programa utilitario de control que está arrancando es compatible con la secuencia de arranque común. Por ejemplo, en los ejemplos de este documento, el programa utilitario de control está escrito para una arquitectura en particular, por ejemplo, la z/Architecture, y la secuencia de arranque se basa en esto. Por lo tanto, se produce una falta de coincidencia si resulta que el programa utilitario de control está escrito para otra arquitectura, tal como ESA/390, sin contemplar z/Architecture.

Por lo tanto, en una realización, se realiza una verificación para una instrucción SIGP al principio de la secuencia de arranque, que será una indicación, en un ejemplo, de si el programa utilitario de control está destinado a la z/Architecture. Esta lógica puede incluirse directamente en la lógica del programa que implementa la inicialización de

la secuencia de arranque, y la instrucción SIGP se puede incluir en la secuencia de arranque como se describe en la Figura 11.

Una realización de esta lógica de detección se describe con referencia a la Figura 18) Inicialmente, se ejecuta hasta un número predefinido de instrucciones en la secuencia de arranque, ETAPA 1800. Si se ha ejecutado un número predefinido de instrucciones, n, sin ejecutar una instrucción SIGP, PREGUNTA 1802, entonces se toman medidas correctivas, ya que se supone que el programa utilitario de control no es un programa utilitario de arranque común, ETAPA 1804. Esta acción incluye, por ejemplo, detener la ejecución, generar una excepción y/o proporcionar una notificación.

Volviendo a PREGUNTA 1802, si se ha ejecutado SIGP, entonces se supone que la secuencia de arranque puede continuar procesando, como se describió anteriormente, ETAPA 1806.

La lógica anterior supone que se emitirá un SIGP en la secuencia de arranque común. Si se utilizan otros tipos de secuencias de arranque en las que un SIGP podría no ejecutarse, entonces se puede utilizar otra lógica de detección. En otra realización, se utiliza la misma lógica de detección cuando un programa utilitario de control se adapta a la ejecución exclusiva en sistemas que se arrancan de forma nativa en modo z/Architecture, como se describe, por ejemplo, con referencia a la Figura 10. La secuencia de arranque de la Figura 10 se aumenta, por ejemplo, con Establecer Arquitectura SIGP (u otro indicador, tal como, que incluye, pero no se limita a, una "afirmación SIGP recién definida" z/Architecture "que indica que se está arrancando una aplicación de z/Architecture) para cumplir los requisitos de la lógica de control de la figura 18.

Se describió en detalle anteriormente una capacidad para utilizar una secuencia de arranque común para arrancar un programa utilitario de control en una pluralidad de configuraciones arquitectónicas. Con esta capacidad, el procesador puede arrancar el mismo sistema operativo y otros programas utilitarios de control en arranques de entornos informáticos que se arrancan en ESA/390 y conmutan a z/Architecture, y aquellos que arrancan en z/Architecture. El hardware se modifica, en un ejemplo, para no informar un error en SIGP, si SIGP intenta cambiar al mismo modo arquitectónico.

Otros aspectos con respecto a una secuencia de arranque común se describen con referencia a la Figura 19. Inicialmente, se inicia una secuencia de arranque para arrancar un programa utilitario de control, como un sistema operativo, en un modo arquitectónico seleccionado, ETAPA 1900. El modo arquitectónico seleccionado se basa en una arquitectura seleccionada para la cual se escribe el programa utilitario de control. El programa utilitario de control se configura para inicializarse en al menos una de una primera configuración arquitectónica que se inicializa en una arquitectura y conmuta a la arquitectura seleccionada para el procesamiento, o en una segunda configuración arquitectónica que se inicializa y procesa en la arquitectura seleccionada.

La secuencia de arranque se realiza e incluye, por ejemplo, establecer la palabra de estado del programa con base en la configuración arquitectónica actual del procesador, ETAPA 1902. Adicionalmente, una o más instrucciones se ejecutan independientemente de si el procesador debe realizar una inicialización basada en la primera configuración arquitectónica o la segunda configuración arquitectónica, ETAPA 1904. La una o más instrucciones pueden incluir una conmutación de modo de preparación, ETAPA 1906, en el que el procesador se prepara para realizar una operación de conmutación independientemente del modo arquitectónico del procesador. Por ejemplo, se proporciona un código de orden Establecer Arquitectura a una instrucción SIGP para ejecutar la operación de conmutación, ETAPA 1908. La operación de conmutación solicita una conmutación de modos arquitectónicos para el procesador, independientemente de si el procesador debe realizar la inicialización en la arquitectura única o en la arquitectura seleccionada. La operación de conmutación funcionará de manera diferente según la arquitectura, una arquitectura o la arquitectura seleccionada, el procesador realizará la inicialización. Por ejemplo, si el procesador está en la arquitectura única, entonces la operación de conmutación se conmuta desde la arquitectura única hasta la arquitectura seleccionada y el procesamiento del programa utilitario de control está en la arquitectura seleccionada. Sin embargo, si el procesador ya está en la arquitectura seleccionada, entonces la operación de conmutación solicita una conmutación del modo de arquitectura seleccionado al modo de arquitectura seleccionado. Esto da como resultado el estado de almacenamiento que indica que el procesador está actualmente en el modo arquitectónico seleccionado, pero ese estado se trata como aceptable para un emisor de la operación de conmutación. Es decir, no se considera un error. Después de realizar la operación del conmutador, el programa utilitario de control se inicia y el código se realiza en la arquitectura seleccionada, ETAPA 1910.

En los flujos descritos en este documento, la preparación para la operación de conmutación de modo y la instrucción SIGP se pueden mostrar por separado del subconjunto común de instrucciones. Sin embargo, pueden incluirse en el subconjunto común y/o separarse de él.

Los aspectos de la funcionalidad de utilidad de arranque común se pueden utilizar con una o más funcionalidades adicionales, que incluyen, por ejemplo, la funcionalidad de CZAM y una funcionalidad No-DAT, descrita en las siguientes aplicaciones compartidas, comúnmente asignadas "Architectural Mode Configuration", Michael K Gschwind, (Expediente IBM No.: POU920140021US1); y "Managing Processing Associated with Selected

Architectural Facilities”, Gainey et al, (Expediente IBM No.: POU920140020US1), respectivamente, cada uno de los cuales se incorpora en este documento como referencia en su totalidad.

Con referencia a la Figura 20, en un ejemplo, un producto 2000 de programa informático incluye, por ejemplo, uno o más medios 2002 de almacenamiento legibles por ordenador no transitorios para almacenar medios de código de programa legible por ordenador, lógica y/o instrucciones 2004 para proporcionar y facilitar una o más preferidas realizaciones

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

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 se limita 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 anteriores. Una lista no exhaustiva de ejemplos más específicos del medio de almacenamiento legible por ordenador incluye lo siguiente: un disquete de ordenador portátil, un disco duro, una memoria de acceso aleatorio (RAM), una memoria de solo lectura (ROM), una memoria de solo lectura programable borrrable (EPROM o memoria Flash), una memoria estática de acceso aleatorio (SRAM), una memoria portátil de solo lectura de disco compacto (CD-ROM), un disco digital versátil (DVD), un dispositivo de memoria, un disquete, un dispositivo mecánico dispositivo codificado tal como tarjetas perforadas o estructuras elevadas en una ranura que tiene instrucciones grabadas al respecto, y cualquier combinación adecuada de lo anterior. Un medio de almacenamiento legible por ordenador, como se utiliza en este documento, no debe interpretarse como señales transitorias per se, tales como ondas de radio u otras ondas electromagnéticas que se propagan libremente, 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.

Las instrucciones del programa legible por ordenador descritas en este documento se pueden descargar a los respectivos dispositivos informáticos/de procesamiento 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 amplia y/o una red inalámbrica. La red puede comprender cables de transmisión de cobre, fibras de transmisión óptica, transmisión inalámbrica, enrutadores, cortafuegos, conmutadores, ordenadores de compuerta de enlace y/o servidores de EDGE. Una tarjeta adaptadora de red o interfaz de red en cada dispositivo informático/de procesamiento recibe instrucciones de programa legibles por ordenador de la red y reenvía las instrucciones de programa legibles por ordenador para almacenamiento en un medio de almacenamiento legible por ordenador dentro del dispositivo informático/de procesamiento respectivo.

Las instrucciones de programa legibles por ordenador para llevar a cabo las operaciones de la presente invención pueden ser instrucciones de ensamblador, instrucciones de Establecer Arquitectura de instrucción (ISA), instrucciones de máquina, instrucciones dependientes de máquina, microcódigo, instrucciones de firmware, datos de configuración de estado o código fuente u objeto código escrito en cualquier combinación de uno o más lenguajes de programación, que incluyen un lenguaje de programación orientado a objetos tal como Smalltalk, C++ o similar, y lenguajes de programación de procedimientos convencionales, tales como el lenguaje de programación “C” o lenguajes de programación similares. Las instrucciones del programa legible por ordenador se pueden ejecutar completamente en el ordenador del usuario, en parte en el ordenador del usuario, como un paquete de software independiente, en parte en el ordenador del usuario y en parte en un ordenador remoto o completamente en el ordenador o servidor remoto. En el último escenario, el ordenador remoto se puede conectar al ordenador del usuario a través de cualquier tipo de red, que incluye una red de área local (LAN) o una red de área amplia (WAN), o la conexión se puede hacer a un ordenador externo (por ejemplo, a través de Internet utilizando un proveedor de servicios de Internet). En algunas realizaciones preferidas, los circuitos electrónicos que incluyen, por ejemplo, circuitos lógicos programables, matrices de compuerta programables en campo (FPGA) o arreglos lógicos programables (PLA) pueden ejecutar las matrices de programa legibles por ordenador al utilizar información de estado de las instrucciones de programa legibles por ordenador para personalizar los circuitos electrónicos para realizar aspectos de la presente invención.

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

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 modo que las instrucciones, que se ejecutan a través del procesador del ordenador u otro aparato de procesamiento de datos programable, cree medios para implementar las funciones/actos especificados en el bloque o bloques de diagrama de flujo y/o del diagrama de bloques. Estas instrucciones de programa legibles por ordenador también pueden almacenarse en un medio de almacenamiento legible por ordenador que puede dirigir un ordenador, un aparato de procesamiento de datos programable y/u otros dispositivos para que funcionen de una manera particular, de modo que el medio de almacenamiento legible por ordenador que tiene instrucciones almacenadas en el mismo comprende un artículo de fabricación que incluye instrucciones que implementan aspectos de la función/acto especificado en el bloque o bloques de diagrama de flujo y/o del diagrama de bloques.

Las instrucciones del programa legible por ordenador también se pueden cargar en un ordenador, otro aparato de procesamiento de datos programable u otro dispositivo para hacer que se realicen una serie de etapas operativas en el ordenador, otro aparato programable u otro dispositivo para producir un proceso implementado por ordenador, de tal manera que las instrucciones que se ejecutan en el ordenador, otro aparato programable u otro dispositivo implementan funciones/actos especificados en el bloque o bloques del diagrama de flujo y/o diagrama de bloque.

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 informáticos de acuerdo con diversas realizaciones preferidas de la presente invención. A este respecto, cada bloque en el diagrama de flujo o en los diagramas de bloques puede representar un módulo, segmento o porción de instrucciones, que comprende una o más instrucciones ejecutables para implementar las funciones lógicas especificadas. En algunas implementaciones alternativas, las funciones indicadas en el bloque pueden ocurrir fuera del orden indicado en las figuras. Por ejemplo, dos bloques mostrados en sucesión pueden, de hecho, ejecutarse de manera sustancialmente concurrente, o los bloques a veces pueden ejecutarse en el orden inverso, dependiendo de la funcionalidad involucrada. 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, pueden implementarse mediante sistemas basados en hardware de propósito especial que realizan las funciones o actos especificados o llevar a cabo combinaciones de hardware para fines especiales e instrucciones informáticas.

Además de lo anterior, uno o más aspectos pueden ser proporcionados, ofrecidos, desplegados, gestionados, atendidos, etc. por un proveedor de servicios que ofrece gestión de entornos de clientes. Por ejemplo, el proveedor de servicios puede crear, mantener, dar soporte, etc., código informático y/o una infraestructura informática que realice uno o más aspectos para uno o más clientes. A cambio, el proveedor de servicios puede recibir el pago del cliente bajo un acuerdo de suscripción y/o tarifa, como ejemplos. Adicional o alternativamente, el proveedor de servicios puede recibir el pago de la venta de contenido publicitario a uno o más terceros.

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

Como un 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 preferidas.

Como aún otro aspecto adicional, se puede proporcionar un proceso para integrar la infraestructura informática que comprende integrar código legible por ordenador en un sistema informático. El sistema informático comprende un medio legible por ordenador, en el que el medio informático comprende una o más realizaciones preferidas. El código en combinación con el sistema informático es capaz de realizar una o más realizaciones preferidas.

Aunque se describen varios anteriormente, estos son solo ejemplos. Por ejemplo, los entornos informáticos de otras arquitecturas se pueden utilizar para incorporar y utilizar una o más realizaciones preferidas. Adicionalmente, se pueden utilizar diferentes instrucciones, formatos de instrucción, campos de instrucción y/o valores de instrucción. También, se pueden utilizar otras secuencias de arranque y re arranque. Son posibles muchas variaciones.

Adicionalmente, otros tipos de entornos informáticos pueden beneficiarse y ser utilizados. Como ejemplo, se puede utilizar un sistema de procesamiento de datos adecuado para almacenar y/o ejecutar código de programa que incluye al menos dos procesadores acoplados directa o indirectamente a elementos de memoria a través de un bus del sistema. Los elementos de memoria incluyen, por ejemplo, la memoria local empleada durante la ejecución real del código del programa, el almacenamiento masivo y la memoria caché que proporcionan almacenamiento temporal de al menos parte del código del programa para reducir la cantidad de veces que el código debe recuperarse del almacenamiento masivo durante la ejecución

Los dispositivos de entrada/salida o de I/O (incluyen, pero no se limitan a, teclados, pantallas, dispositivos señaladores, DASD, cintas, CD, DVD, unidades de memoria USB y otros medios de memoria, etc.) se pueden acoplar al sistema directamente o 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 se acople a otros sistemas de procesamiento de datos o impresoras remotas o dispositivos de almacenamiento a través de redes públicas o privadas que intervienen. Los módems, los módems de cable y las tarjetas Ethernet son solo algunos de los tipos de adaptadores de red disponibles.

5 Con referencia a la Figura 21, se representan componentes representativos de un sistema 5000 de ordenador central para implementar una o más realizaciones preferidas. El ordenador 5000 central representativo comprende una o más CPU 5001 en comunicación con la memoria 5002 de ordenador (es decir, almacenamiento central), así como interfaces de I/O para dispositivos de medios 5011 de almacenamiento y redes 5010 para comunicarse con otros ordenadores o SAN y similares. La CPU 5001 cumple con una arquitectura que tiene un conjunto de instrucciones diseñado y una funcionalidad diseñada. La CPU 5001 puede tener traducción 5012 de registro de acceso (ART), que incluye un búfer 5013 lateral de ART (ALB), para seleccionar un espacio de direcciones para ser utilizado por la traducción 5003 dinámica de direcciones (DAT) para transformar direcciones de programa (direcciones virtuales) en reales direcciones de memoria. Un DAT normalmente incluye un búfer 5007 lateral de traducción (TLB) para el almacenamiento en caché de las traducciones, de modo que los accesos posteriores al bloque de memoria 5002 del ordenador no requieran el retraso de la traducción de la dirección. Normalmente, se emplea un caché 5009 entre la memoria 5002 del ordenador y el procesador 5001. El caché 5009 puede ser jerárquico y tiene un caché grande disponible para más de una CPU y cachés más pequeños y más rápidos (nivel inferior) entre el caché grande y cada CPU. En algunas implementaciones, los cachés de nivel inferior se dividen para proporcionar cachés separados de bajo nivel para la extracción de instrucciones y el acceso a datos.

En una realización, instrucción se extrae de una memoria 5002 mediante una unidad 5004 de extracción de instrucción a través de un caché 5009. La instrucción se decodifica en una unidad de 5006 decodificación de instrucciones y se envía (con otras instrucciones en algunas realizaciones preferidas) a la unidad o unidades 5008 de ejecución de instrucciones. Normalmente, se emplean varias unidades 5008 de ejecución, por ejemplo, una unidad de ejecución aritmética, una unidad de ejecución de punto de flotación y una unidad de ejecución de instrucción de derivación. La unidad de ejecución ejecuta la instrucción, accediendo a los operandos desde los registros especificados de la instrucción o la memoria según sea necesario. Si se va a acceder a un operando (cargado o almacenado) desde la memoria 5002, una unidad 5005 de carga/almacenamiento normalmente maneja el acceso bajo el control de la instrucción que se está ejecutando. Las instrucciones pueden ejecutarse en circuitos de hardware o en microcódigo interno (firmware) o mediante una combinación de ambos.

Como se señaló, un sistema informático incluye información en el almacenamiento local (o principal), así como direccionamiento, protección y registro de referencia y cambio. Algunos aspectos del direccionamiento incluyen el formato de las direcciones, el concepto de espacios de direcciones, los diversos tipos de direcciones y la forma en que un tipo de dirección se traduce a otro tipo de dirección. Parte del almacenamiento principal incluye ubicaciones de almacenamiento asignadas permanentemente. El almacenamiento principal proporciona al sistema un almacenamiento de datos de acceso rápido directamente direccionable. Tanto los datos como los programas deben cargarse en el almacenamiento principal (desde los dispositivos de entrada) antes de que se puedan procesar.

El almacenamiento principal puede incluir uno o más almacenamientos de búfer más pequeños y de acceso más rápido, a veces llamados cachés. Un caché generalmente está físicamente asociado con una CPU o un procesador de I/O. El programa generalmente no puede observar los efectos, excepto en el rendimiento, de la construcción física y el uso de distintos medios de almacenamiento.

Se pueden mantener cachés separados para instrucciones y para operandos de datos. La información dentro de un 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 de ATRIBUTO DE CACHÉ DE EXTRACCIÓN que devuelve el tamaño de una línea de caché en bytes. En otra realización, esta información puede obtenerse del firmware, por ejemplo, de acuerdo con las interfaces especificadas por la especificación de Referencia de Plataforma de Arquitectura de Potencia. Un modelo también puede proporcionar una o más instrucciones de DATOS PREEXTRAÍDOS y DATOS PREEXTRAÍDOS RELATIVOS LARGOS, táctiles de bloqueo de caché de datos (resto), que afectan la extracción previa de almacenamiento en el caché de datos o instrucciones o la liberación de datos del caché.

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

La información se transmite entre el almacenamiento y una CPU o un subsistema de canal de un byte, o un grupo de bytes, a la vez. A menos que se especifique lo contrario, en, por ejemplo, Power ISA y z/Architecture, un grupo de bytes en el almacenamiento se direcciona por el byte más a la izquierda del grupo. El número de bytes en el grupo

está implícito o explícitamente especificado por la operación a realizar. Cuando se utiliza en una operación de CPU, un grupo de bytes se denomina campo. Dentro de cada grupo de bytes, en, por ejemplo, Power ISA y z/Architecture, los bits se numeran en una secuencia de izquierda a derecha. En Power ISA y z/Architecture, los bits más a la izquierda a veces se denominan bits de "orden superior" y los bits más a la derecha como bits de "orden inferior". Sin embargo, los números de bit no son direcciones de almacenamiento. Solo se pueden direccionar los bytes. Para operar en bits individuales de un byte en almacenamiento, se accede a todo el byte. Los bits en un byte están numerados del 0 al 7, de izquierda a derecha (en, por ejemplo, z/Architecture). Los bits en una dirección pueden estar numerados 8-31 o 40-63 para direcciones de 24 bits, o 1-31 o 33-63 para direcciones de 31 bits; están numerados del 0 al 63 para direcciones de 64 bits. En un ejemplo, los bits 8-31 y 1-31 se aplican a las direcciones que se encuentran en una ubicación (por ejemplo, registro) que tiene 32 bits de ancho, mientras que los bits 40-63 y 33-63 se aplican a las direcciones que están en una ubicación amplia de bit 64. Dentro de cualquier otro formato de longitud fija de múltiples bytes, los bits que componen el formato se numeran consecutivamente a partir de 0. Para propósitos de detección de errores, y preferiblemente para corrección, se pueden transmitir uno o más bits de verificación con cada byte o con un grupo de bytes. Dichos bits de verificación son generados automáticamente por la máquina y no pueden ser controlados directamente por el programa. Las capacidades de almacenamiento se expresan en número de bytes. Cuando la longitud de un campo de operando de almacenamiento está implícita en 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 implicados para algunas instrucciones. Cuando la longitud de un campo de operando a almacenamiento no está implícita, pero se establece explícitamente, se dice que el campo tiene una longitud variable. Los operandos de longitud variable pueden variar en longitud en incrementos de un byte (o con algunas instrucciones, en múltiplos de dos bytes u otros múltiplos). Cuando la información se coloca en almacenamiento, solo se reemplaza el contenido de las ubicaciones de bytes que se incluyen en el campo designado, aunque el ancho de la ruta física al almacenamiento puede ser mayor que la longitud del campo que se está almacenando.

Ciertas unidades de información deben 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 componente básico de las instrucciones. Una palabra es un grupo de cuatro bytes consecutivos en un límite de cuatro bytes. Una palabra doble es un grupo de ocho bytes consecutivos en un límite de ocho bytes. Una palabra cuádruple es un grupo de 16 bytes consecutivos en un límite de 16 bytes. Una octopalabra 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 octopalabras, la representación binaria de la dirección contiene uno, dos, tres, cuatro o cinco bits cero a la derecha, respectivamente. Las instrucciones deben estar en límites integrales de dos bytes. Los operandos de almacenamiento de la mayoría de las instrucciones no tienen requisitos de alineación de límites.

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

En un ejemplo, la realización puede ser practicada por software (a veces referido a código interno con licencia, firmware, microcódigo, milicódigo, picocódigo y similares, cualquiera de los cuales sería consistente con una o más realizaciones preferidas). Con referencia a la Figura 21, el procesador 5001 del sistema 5000 central puede acceder al código del programa de software que incorpora uno o más aspectos desde los dispositivos 5011 de almacenamiento a largo plazo, tal como una unidad de CD-ROM, una unidad de cinta o un disco duro. El código del programa de software se puede incorporar en cualquiera de una variedad de medios conocidos para utilizar con un sistema de procesamiento de datos, tal como un disquete, disco duro o CD-ROM. El código puede distribuirse en dichos medios, o puede distribuirse a los usuarios desde la memoria 5002 del ordenador o el almacenamiento de un sistema informático a través de una red 5010 a otros sistemas informáticos para uso de los usuarios de dichos otros sistemas.

El código del programa de software incluye un sistema operativo que controla la función e interacción de los diversos componentes del ordenador y uno o más programas de aplicación. El código de programa normalmente se envía desde el dispositivo 5011 de medios de almacenamiento al almacenamiento 5002 de ordenador de velocidad relativamente más alta en el que está disponible para su procesamiento por el procesador 5001. Las técnicas y métodos para incorporar código de programa de software en la memoria, en medios físicos y/o distribuir código de software a través de redes son bien conocidos y no se discutirán más en este documento. El código del programa, cuando se crea y almacena en un medio tangible (que incluye, pero no se limita a módulos de memoria electrónica (RAM), memoria flash, discos compactos (CD), DVD, cinta magnética y similares, a menudo se denomina "producto de programa informático". El medio de producto del programa informático es normalmente legible por un circuito de procesamiento, preferiblemente en un sistema informático para su ejecución por el circuito de procesamiento.

La Figura 22 ilustra una estación de trabajo representativa o un sistema de hardware de servidor en el que se pueden practicar una o más realizaciones preferidas. El sistema 5020 de la Figura 22 comprende un sistema

informático 5021 base representativo, tal como un ordenador personal, una estación de trabajo o un servidor, que incluye dispositivos periféricos opcionales. El sistema 5021 i informático base incluye uno o más procesadores 5026 y un bus empleado para conectar y permitir la comunicación entre el procesador o procesadores 5026 y los otros componentes del sistema 5021 de acuerdo con técnicas conocidas. El bus conecta el procesador 5026 a la memoria 5025 y al almacenamiento 5027 a largo plazo que puede incluir un disco duro (que incluye cualquiera de los medios magnéticos, CD, DVD y memoria Flash, 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, tal 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 táctil, teclado de entrada digitalizado, etc. El bus también conecta un dispositivo de visualización 5022, tal como una pantalla LCD o monitor, al microprocesador 5026 a través de un adaptador de pantalla.

El sistema 5021 se puede comunicar con otros ordenadores o redes de ordenadores por medio de un adaptador de red capaz de comunicarse 5028 con una red 5029. Los adaptadores de red de ejemplo son canales de comunicaciones, anillos de credencial, Ethernet o módems. Alternativamente, el sistema 5021 puede comunicarse utilizando una interfaz inalámbrica, tal como una tarjeta CDPD (paquete de datos digitales celulares). El sistema 5021 puede estar asociado con otros ordenadores en una red de área local (LAN) o una red de área amplia (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 apropiados, se conocen en la técnica.

La Figura 23 ilustra una red de procesamiento de datos 5040 en la que se pueden practicar una o más realizaciones preferidas. La red 5040 de procesamiento de datos 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 5041, 5042, 5043, 5044 de trabajo individuales. Adicionalmente, como lo apreciarán aquellos expertos en la técnica, se pueden incluir una o más LAN, en las que una LAN puede comprender una pluralidad de estaciones de trabajo inteligentes acopladas a un procesador central.

Todavía en referencia a la Figura 23, las redes también pueden incluir ordenadores o servidores de mainframe, tales como un ordenador de compuerta de enlace (servidor 5046 de cliente) o un servidor de aplicaciones (servidor 5048 remoto que puede acceder a un repositorio de datos y también se puede acceder directamente desde una estación 5045 de trabajo). Un ordenador 5046 de compuerta de enlace sirve como punto de entrada en cada red individual. Se necesita una compuerta de enlace cuando se conecta un protocolo de red a otro. La compuerta 5046 de enlace puede estar preferiblemente acoplada a otra red (por ejemplo, Internet 5047) por medio de un enlace de comunicaciones. La compuerta 5046 de enlace también se puede acoplar directamente a una o más estaciones 5041, 5042, 5043, 5044 de trabajo utilizando un enlace de comunicaciones. El ordenador de compuerta de enlace se puede implementar utilizando uno de los servidores de IBM Power Systems y un servidor de IBM System z® disponible de International Business Machines Corporation. El System z es una marca comercial de International Business Machines Corporation, registrada en muchas jurisdicciones en todo el mundo.

Refiriéndose simultáneamente a la Figura 22 y la Figura 23, el código 5031 de programación de software, que puede incorporar uno o más aspectos, puede ser accedido por el procesador 5026 del sistema 5020 desde un medio 5027 de almacenamiento a largo plazo, tal como una unidad de CD-ROM o disco duro. El código de programación de software se puede incorporar en cualquiera de una variedad de medios conocidos para utilizar con un sistema de procesamiento de datos, como un disquete, disco duro o CD-ROM. El código se puede distribuir en dichos medios, o se puede distribuir a los usuarios 5050, 5051 desde la memoria o el almacenamiento de un sistema informático a través de una red a otros sistemas informáticos para uso de los usuarios de dichos otros sistemas.

Alternativamente, el código de programación puede estar incorporado en la memoria 5025, y el procesador 5026 puede acceder a él utilizando el bus del procesador. Dicho código de programación incluye un sistema operativo que controla la función e interacción de los diversos componentes informáticos y uno o más programas 5032 de aplicación. El código del programa normalmente se envía desde el medio 5027 de almacenamiento a la memoria 5025 de alta velocidad, donde está disponible para su procesamiento por el procesador 5026. Las técnicas y métodos para incorporar código de programación de software en la memoria, en medios físicos, y/o distribuir código de software a través de redes son bien conocidos y no se discutirán más en este documento. El código del programa, cuando se crea y almacena en un medio tangible (que incluye, entre otros, módulos de memoria electrónica (RAM), memoria flash, discos compactos (CD), DVD, cinta magnética y similares, a menudo se denomina "producto de programa informático". El medio de producto del programa informático es normalmente legible por un circuito de procesamiento, preferiblemente en un sistema informático para su ejecución por el circuito de procesamiento.

El caché que está más fácilmente disponible para el procesador (normalmente más rápido y más pequeño que otros cachés del procesador) es el caché más bajo (L1 o nivel uno) y el almacén principal (memoria principal) es el caché de nivel más alto (L3 si hay 3 niveles). El caché de nivel más bajo a menudo se divide en un caché de instrucciones (I-Caché) que contiene instrucciones de máquina para ejecutar y un caché de datos (D-Caché) que contiene operando de datos.

Con referencia a la Figura 24, se representa una realización de procesador de ejemplo para el procesador 5026. Normalmente, uno o más niveles de caché 5053 se emplean para almacenar bloques de memoria para mejorar el rendimiento del procesador. El caché 5053 es un búfer de alta velocidad que contiene líneas de caché de datos de memoria que es probable que se utilicen. Las líneas de caché típicas son 64, 128 o 256 bytes de datos de memoria. Las memorias caché separadas a menudo se emplean para almacenar en caché las instrucciones que para almacenar en caché los datos. La coherencia de la memoria caché (sincronización de copias de líneas en la memoria y las memorias caché) a menudo es proporcionada por varios algoritmos “de seguimiento” bien conocidos en la técnica. El almacenamiento 5025 de memoria principal de un sistema procesador a menudo se denomina caché. En un sistema de procesador que tiene 4 niveles de caché 5053, el almacenamiento 5025 principal a veces se denomina caché de nivel 5 (L5) ya que generalmente es más rápido y solo contiene una parte del almacenamiento no volátil (DASD, cinta, etc.) que está disponible para un sistema informático. Almacenamiento principal 5025 “almacena en caché” páginas de datos paginados dentro y fuera del almacenamiento 5025 principal por el sistema operativo.

Un contador 5061 de programa (contador de instrucciones) realiza un seguimiento de la dirección de la instrucción actual a ejecutar. Un contador de programa en un procesador z/Architecture tiene 64 bits y se puede truncar a 31 o 24 bits para admitir límites de direccionamiento anteriores. Un contador de programa en un procesador de Arquitectura de Potencia tiene 64 bits y se puede truncar a 32 bits para admitir límites de direccionamiento anteriores. Un contador de programa normalmente está incorporado en un PSW (palabra de estado del programa) de un ordenador de modo que persista durante la conmutación de contexto. Por lo tanto, un programa en progreso, que tiene un valor de contador de programa, puede ser interrumpido, por ejemplo, por el sistema operativo (cambio de contexto del entorno del programa al entorno del sistema operativo). El PSW del programa mantiene el valor del contador del programa mientras el programa no está activo, y el contador del programa (en el PSW) del sistema operativo se utiliza mientras el sistema operativo se está ejecutando. Normalmente, el contador del programa se incrementa en una cantidad igual al número de bytes de la instrucción actual. Las instrucciones de RISC (Cómputo de Conjunto de Instrucciones Complejas) son normalmente de longitud fija, mientras que las instrucciones CISC (Computación de conjunto de instrucciones complejas) son normalmente de longitud variable. Las instrucciones de IBM z/Architecture son instrucciones CISC que tienen una longitud de 2, 4 o 6 bytes. Las instrucciones de IBM Power ISA son instrucciones RISC que tienen una longitud de 4 bytes. El contador 5061 de programa se modifica por una operación de conmutación de contexto o por una operación tomada de derivación o una instrucción de derivación, por ejemplo. En una operación de conmutación de contexto, el valor del contador del programa actual se guarda en la palabra de estado del programa junto con otra información de estado sobre el programa que se está ejecutando (como los códigos de condición), y se carga un nuevo valor del contador del programa apuntando a una instrucción de un nuevo módulo del programa que se va a ejecutar. Se realiza una operación de derivación tomada para permitir que el programa tome decisiones o se repita dentro del programa al cargar el resultado de la instrucción de derivación en el contador 5061 del programa.

Normalmente, se utiliza una unidad 5055 de extracción de instrucciones para extraer instrucciones en nombre del procesador 5026. La unidad de extracción extrae “las siguientes instrucciones secuenciales”, instrucciones objetivo de las instrucciones tomadas de la derivación o las primeras instrucciones de un programa después de una conmutación de contexto. Las unidades de extracción de Instrucciones Modernas a menudo emplean técnicas de extracción previa para extraer previamente de forma especulativa instrucciones basadas en la probabilidad de que se puedan utilizar las instrucciones captadas previamente. Por ejemplo, una unidad de extracción puede extraer 16 bytes de instrucción que incluye la siguiente instrucción secuencial y bytes adicionales de instrucciones secuenciales adicionales.

Las instrucciones extraídas son ejecutadas por el procesador 5026. En una realización, las instrucciones extraídas se pasan a una unidad 5056 de envío de la unidad de extracción. La unidad de suministro decodifica las instrucciones y envía información sobre las instrucciones decodificadas a las unidades 5057, 5058, 5060 apropiadas. Una unidad 5057 de ejecución recibirá normalmente información sobre instrucciones aritméticas decodificadas de la unidad 5055 de extracción de instrucciones y realizará operaciones aritméticas en operandos de acuerdo con el código de operación de la instrucción. Los operandos se proporcionan a la unidad 5057 de ejecución preferiblemente desde la memoria 5025, los registros 5059 con arquitectura o desde un campo inmediato de la instrucción que se está ejecutando. Los resultados de la ejecución, cuando se almacenan, se almacenan en la memoria 5025, los registros 5059 o en otro hardware de la máquina (como registros de control, registros PSW y similares).

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

Un procesador 5026 tiene normalmente una o más unidades 5057, 5058, 5060 para ejecutar la función de la instrucción. Con referencia a la Figura 25A, una unidad 5057 de ejecución puede comunicarse 5071 con registros 5059 generales arquitectónicos, una unidad 5056 de decodificación/despacho, una unidad 5060 de almacenamiento de carga y otras unidades 5065 de procesador por medio de la lógica 5071 de interfaz. Una unidad 5057 de ejecución puede emplear varios circuitos 5067, 5068, 5069 de registro para contener información sobre la cual operará la unidad de lógica aritmética (ALU) 5066. La ALU realiza operaciones aritméticas como sumar, restar, multiplicar y dividir, así como funciones lógicas como y, o y o exclusivo (XOR), rotar y desplazar. Preferiblemente, la

ALU admite operaciones especializadas que dependen del diseño. Otros circuitos pueden proporcionar otras funcionalidades 5072 arquitectónicas que incluyen códigos de condición y lógica de soporte de recuperación, por ejemplo. Normalmente, el resultado de una operación de ALU se mantiene en un circuito 5070 de registro de salida que puede enviar el resultado a una variedad de otras funciones de procesamiento. Se presenta muchas disposiciones de unidades de procesador, la presente descripción solo pretende proporcionar una comprensión representativa de una realización.

Una instrucción ADD, por ejemplo, se ejecutaría en una unidad 5057 de ejecución con funcionalidad aritmética y lógica, mientras que una instrucción de punto de flotación, por ejemplo, se ejecutaría en una ejecución de punto de flotación con capacidad especializada de punto de flotación. Preferiblemente, una unidad de ejecución opera en operandos identificados por una instrucción al realizar una función definida de código de operación en los operandos. Por ejemplo, una instrucción ADD puede ser ejecutada por una unidad 5057 de ejecución en operandos encontrados en dos registros 5059 identificados por campos de registro de la instrucción.

La unidad 5057 de ejecución realiza la suma aritmética en dos operandos y almacena el resultado en un tercer operando donde el tercer operando puede ser un tercer registro o uno de los dos registros fuente. La unidad de ejecución utiliza preferiblemente una Unidad 5066 de Lógica Aritmética (ALU) que es capaz de realizar una variedad de funciones lógicas como Cambiar, Rotar, Y, O y XOR, así como una variedad de funciones algebraicas que incluyen cualquiera de sumar, restar, multiplicar, dividir. Algunas ALU 5066 están diseñadas para operaciones escalares y otras para punto de flotación. Los datos pueden ser Big Endian (en los que el byte menos significativo está en la dirección de byte más alta) o Little Endian (en los que el byte menos significativo está en la dirección de byte más baja) dependiendo de la arquitectura. La z/Architecture de IBM es Big Endian. IBM Power ISA soporta los modos de ejecución Big Endian y Little Endian. Los campos firmados pueden ser signos y magnitudes, complemento de 1's o complemento de 2's dependiendo de la arquitectura. Un número de complemento 2's es ventajoso porque la ALU no necesita diseñar una capacidad de sustracción ya que un valor negativo o un valor positivo en el complemento de 2's requiere solo una adición dentro de la ALU. Los números se describen comúnmente en forma abreviada, en los que un campo de 12 bits define una dirección de un bloque de 4.096 bytes y se describe comúnmente como un bloque de 4 Kbytes (Kilobyte), por ejemplo.

Con referencia a la Figura 25B, la información de instrucción de derivación para ejecutar una instrucción de derivación se envía normalmente a una unidad 5058 de derivación que a menudo emplea un algoritmo de predicción de derivación tal como una tabla 5082 de historial de derivación para predecir el resultado de la rama antes de que se completen otras operaciones condicionales. El objetivo de la instrucción de derivación actual se extraerá y se ejecutará especulativamente antes de que se completen las operaciones condicionales. Cuando se completan las operaciones condicionales, las instrucciones de derivación ejecutadas especulativamente se completan o se descartan según las condiciones de la operación condicional y el resultado especulado. Una instrucción de derivación típica puede probar códigos de condición y derivarse a una dirección de destino si los códigos de condición cumplen con el requisito de derivación de la instrucción de derivación, una dirección de destino puede calcularse en función de varios números, incluidos los que se encuentran en los campos de registro o un campo inmediato de la instrucción, por ejemplo. La unidad 5058 de derivación puede emplear una ALU 5074 que tiene una pluralidad de circuitos 5075, 5076, 5077 de registro de entrada y un circuito 5080 de registro de salida. La unidad 5058 de derivación puede comunicarse 5081 con registros 5059 generales, decodificar la unidad 5056 de despacho u otros circuitos 5073, por ejemplo.

La ejecución de un grupo de instrucciones puede interrumpirse por una variedad de razones, que incluye 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 provoca una conmutación de contexto o una actividad de subprocesamiento múltiple de una pluralidad de programas (en un entorno multiproceso), por ejemplo. Preferiblemente, una acción de conmutación de contexto guarda información de estado sobre un programa que se está ejecutando actualmente y luego carga información de estado sobre otro programa que se invoca. La información de estado puede guardarse en registros de hardware o en memoria, por ejemplo. La información de estado comprende preferiblemente un valor de contador de programa que apunta a una próxima instrucción a ejecutar, códigos de condición, información de traducción de memoria y contenido de registro arquitectónico. Una actividad de conmutación de contexto puede ser ejercida por circuitos de hardware, programas de aplicación, programas del sistema operativo o código de firmware (microcódigo, picocódigo o código interno con licencia (LIC)) solo o en combinación.

Un procesador accede a los operandos de acuerdo con los métodos definidos por la instrucción. La instrucción puede proporcionar un operando inmediato utilizando el valor de una porción de la instrucción, puede proporcionar uno o más campos de registro que apuntan explícitamente a registros de propósito general o registros de propósito especial (registros de punto de flotación, por ejemplo). La instrucción puede utilizar registros implícitos identificados por un campo de código de operación como operandos. La instrucción puede utilizar ubicaciones de memoria para operandos. Un registro, un campo inmediato o una combinación de registros y campo inmediato puede proporcionar una ubicación de memoria de un operando, como se ejemplifica en la funcionalidad de desplazamiento largo de z/Architecture en la que la instrucción define un registro base, un registro de índice y un campo inmediato (campo de desplazamiento) que se suman para proporcionar la dirección del operando en la memoria, por ejemplo; o los modos

de direccionamiento Power ISA donde las direcciones en forma de D definen un registro base y un campo inmediato (campo de desplazamiento) que se suman para proporcionar la dirección del operando en la memoria; y en el que las direcciones de Forma X definen un registro base y un registro de índice que se suman para proporcionar la dirección del operando en la memoria. La ubicación en este documento normalmente implica una ubicación en la memoria principal (almacenamiento principal) a menos que se indique lo contrario.

Con referencia a la Figura 25C, un procesador accede al almacenamiento utilizando una unidad 5060 de carga/almacenamiento. La unidad 5060 de carga/almacenamiento puede realizar una operación de carga al obtener la dirección del operando de destino en la memoria 5053 y cargando el operando en un registro 5059 u otra ubicación de memoria 5053, o puede realizar una operación de almacenamiento obteniendo la dirección del operando de destino en la memoria 5053 y almacenar datos obtenidos de un registro 5059 u otra ubicación de la memoria 5053 en la ubicación del operando objetivo en la memoria 5053. La unidad 5060 de carga/almacenamiento puede ser especulativa y puede acceder a la memoria en una secuencia que está fuera de orden en relación con la secuencia de instrucciones, sin embargo, la unidad 5060 de carga/almacenamiento debe mantener la apariencia de los programas cuyas instrucciones se ejecutaron en orden. Una unidad 5060 de carga/almacenamiento puede comunicarse 5084 con registros 5059 generales, unidad 5056 de decodificación/despacho, interfaz 5053 de caché/memoria u otros elementos 5083 y comprende varios circuitos 5086, 5087, 5088 y 5089 de registro, ALU 5085 y lógica 5090 de control para calcular direcciones de almacenamiento y para proporcionar secuencia de canalización para mantener las operaciones en orden. Algunas operaciones pueden estar fuera de servicio, pero la unidad de carga/almacenamiento proporciona funcionalidad para hacer que las operaciones fuera de servicio aparezcan en el programa como ejecutadas en orden, como es bien sabido en la técnica.

Preferiblemente, las direcciones que “ve” un programa de aplicación a menudo se denominan direcciones virtuales. Las direcciones virtuales a veces se denominan “direcciones lógicas” y “direcciones efectivas”. Estas direcciones virtuales son virtuales porque se redirigen a la ubicación de la memoria física mediante una de una variedad de tecnologías de traducción de direcciones dinámicas (DAT) que incluyen, pero no se limitan a, simplemente prefijar una dirección virtual con un valor de desplazamiento, traducir la dirección virtual a través de uno o más tablas de traducción, las tablas de traducción 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 primera tabla de regiones, una segunda tabla de regiones, una tercera tabla de regiones, una tabla de segmentos y una tabla de páginas opcional. El rendimiento de la traducción de la dirección a menudo se mejora mediante la utilización de un búfer lateral de traducción (TLB) que comprende entradas que asignan una dirección virtual a una ubicación de memoria física asociada. Las entradas se crean cuando el DAT traduce una dirección virtual utilizando las tablas de traducción. El uso posterior de la dirección virtual puede utilizar la entrada del TLB rápido en lugar de los accesos lentos a la tabla de traducción secuencial. El contenido de TLB puede ser administrado por una variedad de algoritmos de reemplazo que incluyen LRU (menos utilizado recientemente).

En el caso de que el procesador sea un procesador de un sistema multiprocesador, cada procesador tiene la responsabilidad de mantener los recursos compartidos, tal como I/O, cachés, TLB y memoria, enclavados por coherencia. Normalmente, las tecnologías “de seguimiento” se utilizarán para mantener la coherencia de caché. En un entorno de seguimiento, cada línea de caché se puede marcar como en cualquiera de un estado compartido, un estado exclusivo, un estado modificado, un estado no válido y similares para facilitar el uso compartido.

Las unidades 5054 de I/O (Figura 24) proporcionan al procesador medios para conectarse a dispositivos periféricos, que incluyen cintas, discos, impresoras, pantallas y redes, por ejemplo. Las unidades de I/O a menudo se presentan al programa informático mediante controladores de software. En los mainframes, como el System z de IBM®, los adaptadores de canal y los adaptadores de sistema abierto son unidades de I/O del mainframe que proporcionan las comunicaciones entre el sistema operativo y los dispositivos periféricos. En los servidores RISC, tales como Power Systems de IBM®, los adaptadores patentados y los adaptadores de sistema abierto son unidades de I/O que proporcionan las comunicaciones entre el sistema operativo y los dispositivos periféricos.

Adicionalmente, 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 (que incluye, por ejemplo, ejecución de instrucciones, funciones de arquitectura, tal como traducción de direcciones y registros de arquitectura) o un subconjunto de las mismas, se emula (por ejemplo, en un sistema informático nativo que tiene un procesador y una memoria). En dicho entorno, una o más funciones de emulación del emulador pueden implementar una o más realizaciones preferidas, aunque un ordenador que ejecuta el emulador puede tener una arquitectura diferente a las capacidades que se están emulando. Como un ejemplo, en el modo de emulación, la instrucción u operación específica que se emula se decodifica, y se construye una función de emulación apropiada para implementar la instrucción u operación individual.

En un entorno de emulación, un ordenador central incluye, por ejemplo, una memoria para almacenar instrucciones y datos; una unidad de extracción de instrucciones para extraer instrucciones de la memoria y, opcionalmente, proporcionar almacenamiento en búfer local para la instrucción extraída; una unidad de decodificación de instrucciones para recibir las instrucciones extraídas y para determinar el tipo de instrucciones que se han

recuperado; 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 en la memoria desde un registro; o realizar algún tipo de operación aritmética o lógica, según lo determine la unidad de decodificación. En un ejemplo, cada unidad se implementa en software. Por ejemplo, las operaciones que realizan las unidades se implementan como una o más subrutinas dentro del software emulador.

Más particularmente, en un mainframe, los programadores utilizan las instrucciones de a máquina con arquitectura, generalmente los programadores "C", a menudo a través de una aplicación compiladora. Estas instrucciones almacenadas en el medio de almacenamiento pueden ejecutarse de forma nativa en un servidor Power Systems o z/Architecture IBM®, o alternativamente en máquinas que ejecutan otras arquitecturas. Se pueden emular en los servidores mainframe IBM® existentes y futuros, en los servidores Power Systems y en otras máquinas de IBM® (por ejemplo, servidores IBM System x). Se pueden ejecutar en máquinas que ejecutan Linux en una amplia variedad de máquinas que utilizan hardware fabricado por IBM®, Intel®, AMD y otros. Además de la ejecución en ese hardware bajo Arquitectura de Potencia o z/Architecture, Linux® se puede utilizar, así como máquinas que utilizan la emulación de Hercules, UMX o FSI (Fundamental Software, Inc), donde generalmente la ejecución se realiza en modo de emulación. En modo de emulación, un procesador nativo ejecuta el software de emulación para emular la arquitectura de un procesador emulado. Linux es una marca registrada de Linus Torvalds en los Estados Unidos, en otros países o en ambos.

El procesador nativo normalmente ejecuta un software de emulación que comprende firmware o un sistema operativo nativo para realizar la emulación del procesador emulado. El software de emulación es responsable de extraer y ejecutar instrucciones de la arquitectura del procesador emulado. El software de emulación mantiene un contador de programa emulado para realizar un seguimiento de los límites de las instrucciones. El software de emulación puede extraer 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 que el procesador nativo las ejecute. Estas instrucciones convertidas pueden almacenarse en caché de modo que se pueda lograr una conversión más rápida. No obstante, el software de emulación debe mantener las reglas de arquitectura de la arquitectura del procesador emulado para garantizar que los sistemas operativos y las aplicaciones escritas para el procesador emulado operen correctamente. Adicionalmente, el software de emulación debe proporcionar recursos identificados por la arquitectura del procesador emulado, que incluyen, pero no se limitan a, registros de control, registros de propósito general, registros de punto de flotación, función de traducción de direcciones dinámicas, que incluyen tablas de segmentos y tablas de páginas, por ejemplo, mecanismos de interrupción, mecanismos de conmutación de contexto, relojes de la hora del día (TOD) y las interfaces arquitectónicas para los subsistemas de I/O, de modo que un sistema operativo o un programa de aplicación 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. Una función de software de emulación que emula una función de un procesador emulado se implementa, por ejemplo, en una subrutina o controlador "C", o algún otro método para proporcionar un controlador para el hardware específico, que estará dentro de la habilidad de aquellos expertos en la técnica después de comprender la descripción de la realización preferida de la presente invención. Diversas patentes de emulación de software y hardware que incluyen, pero no se limitan a, los Títulos de la Patente de Estados Unidos No. 5,551,013, titulada "Multiprocessor for Hardware Emulation", otorgada a Beausoleil et al.; y Títulos de la Patente de Estados Unidos No. 6,009,261, titulada "Preprocessing of Stored Target Routines for Emulating Incompatible Instructions on a Target Processor", otorgada a Scalzi et al; y Títulos de la Patente de Estados Unidos No. 5,574,873, titulada "Decoding Guest Instruction to Directly Access Emulation Routines that Emulate the Guest Instructions", otorgada a Davidian et al; y Títulos de la Patente de Estados Unidos No. 6,308,255, titulada "Symmetrical Multiprocessing Bus and Chipset Used for Coprocessor Support Allowing Non-Native Code to Run in a System", otorgada a Gorishek et al; y Títulos de la Patente de Estados Unidos No. 6,463,582, titulada "Dynamic Optimizing Object Code Translator for Architecture Emulation and Dynamic Optimizing Object Code Translation Method", otorgada a Lethin et al; y Títulos de la Patente de Estados Unidos No. 5,790,825, titulada "Method for Emulating Guest Instructions on a central Computer Through Dynamic Recompilation of central Instructions", otorgada a Eric Traut cada una de las cuales se incorpora en este documento como referencia en su totalidad; y muchos otros, ilustran una variedad de formas conocidas para lograr la emulación de un formato de instrucciones diseñado para una máquina diferente para una máquina de destino disponible para aquellos expertos en la técnica.

En la Figura 26, se proporciona un ejemplo de un sistema 5092 informático central emulado que emula un sistema 5000' informático central de una arquitectura central. En el sistema 5092 informático central emulado, el procesador 5091 central (CPU) es un procesador central emulado (o procesador central virtual) y comprende un procesador 5093 de emulación que tiene una arquitectura establecida de instrucciones nativas diferente a la del procesador 5091 del ordenador 5000' central. El sistema 5092 informático central emulado tiene memoria 5094 accesible para el procesador 5093 de emulación. En el ejemplo de realización preferida de la presente invención, la memoria 5094 se divide en una porción de memoria 5096 de ordenador central y una porción de rutinas de emulación 5097. La memoria 5096 del ordenador central está disponible para los programas del ordenador 5092 central emulado de acuerdo con la arquitectura del ordenador central. El procesador 5093 de emulación ejecuta instrucciones nativas de un conjunto de instrucciones arquitectónicas de una arquitectura diferente a la del procesador 5091 emulado, las

instrucciones nativas obtenidas de la memoria de rutinas 5097 de emulación, y puede acceder a una instrucción central para la ejecución desde un programa en la memoria 5096 del ordenador central mal emplear una o más instrucciones obtenidas en una secuencia y secuencia de acceso/decodificación que pueden decodificar las instrucciones centrales a las que se accede para determinar una rutina de ejecución de instrucciones nativas para emular la función de la instrucción central a la que se accede. Otras funcionalidades que se definen para la arquitectura del sistema 5000' informático central pueden ser emuladas por rutinas de funcionalidades arquitectónicas, incluidas dichas funcionalidades como registros de propósito general, registros de control, traducción de direcciones dinámicas y soporte del subsistema de I/O y caché del procesador, por ejemplo. Las rutinas de emulación también pueden aprovechar las funciones disponibles en el procesador 5093 de emulación (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 de descarga para ayudar al procesador 5093 a emular la función del ordenador 5000' central.

En una realización preferida adicional de la presente invención, uno o más aspectos se refieren a la informática en la nube. Se entiende de antemano que, aunque esta divulgación incluye una descripción detallada sobre la informática en la nube, la implementación de las enseñanzas en este documento mencionadas no se limita a un entorno informático en la nube. Por el contrario, las realizaciones preferidas de la presente invención pueden implementarse junto con cualquier otro tipo de entorno informático ahora conocido o desarrollado posteriormente.

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

Las características son las siguientes:

Autoservicio a pedido: un consumidor de la nube puede aprovisionar unilateralmente capacidades informáticas, tal como la hora del servidor y el almacenamiento en red, según sea necesario automáticamente sin requerir la interacción humana con el proveedor del servicio.

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

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

Elasticidad rápida: las capacidades se pueden aprovisionar rápida y elásticamente, en algunos casos automáticamente, para escalar rápidamente y liberarse rápidamente para escalar rápidamente. Para el consumidor, las capacidades disponibles para el aprovisionamiento a menudo parecen ser ilimitadas y se pueden comprar en cualquier cantidad en cualquier momento.

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

Los modelos de servicio son los siguientes:

Software como servicio (SaaS): la capacidad que se brinda al consumidor es utilizar las aplicaciones del proveedor que se ejecutan en una infraestructura en la nube. Se puede acceder a las aplicaciones desde varios dispositivos cliente a través de una interfaz de cliente ligero, como un navegador web (por ejemplo, correo electrónico basado en la web). El consumidor no gestiona ni controla la infraestructura de la nube subyacente, que incluye la red, servidores, sistemas operativos, almacenamiento o incluso las capacidades de las aplicaciones individuales, con la posible excepción de la configuración de la aplicación limitada específica del usuario.

Plataforma como servicio (PaaS): la capacidad proporcionada al consumidor es implementar en la infraestructura de la nube aplicaciones creadas o adquiridas por el consumidor creado utilizando lenguajes de programación y herramientas compatibles con el proveedor. El consumidor no administra ni controla la infraestructura de nube subyacente, incluidas las redes, los servidores, los sistemas operativos o el almacenamiento, pero tiene control sobre las aplicaciones implementadas y posiblemente las configuraciones del entorno de alojamiento de aplicaciones.



Infraestructura como servicio (IaaS): la capacidad que se brinda al consumidor es el aprovisionamiento de procesamiento, almacenamiento, redes y otros recursos informáticos fundamentales en los que el consumidor puede implementar y ejecutar software arbitrario, que puede incluir sistemas operativos y aplicaciones. El consumidor no administra ni controla la infraestructura de nube subyacente, pero tiene control sobre los sistemas operativos, el almacenamiento, las aplicaciones implementadas y posiblemente el control limitado de componentes de red seleccionados (por ejemplo, cortafuegos centrales).

Los modelos de implementación son los siguientes:

Nube privada: la infraestructura de la nube se opera únicamente para una organización. Puede ser administrado por la organización o por un tercero y puede existir dentro o fuera de las instalaciones.

Nube comunitaria: la infraestructura de la nube es compartida por varias organizaciones y soporta una comunidad específica que ha compartido preocupaciones (por ejemplo, misión, requisitos de seguridad, políticas y consideraciones de cumplimiento). Se puede administrar por las organizaciones o por un tercero y puede existir dentro o fuera de las instalaciones.

Nube pública: la infraestructura de 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 de la nube es una composición de dos o más nubes (privadas, comunitarias o públicas) que siguen siendo entidades únicas, pero están unidas por tecnología estandarizada o patentada que permite la portabilidad de datos y aplicaciones (por ejemplo, la explosión de la nube para el equilibrio de carga entre nubes)

Un entorno informático en la nube está orientado al servicio con un enfoque en la capacidad de no registrar interacciones, bajo acoplamiento, la modularidad y interoperabilidad semántica. En el núcleo de la informática en la nube hay una infraestructura que comprende una red de nodos interconectados.

Con referencia ahora a la Figura 27, se muestra un esquema de un ejemplo de un nodo informático en la nube. El nodo 6010 informático en la nube es solo un ejemplo de un nodo informático en la nube adecuado y no pretende sugerir ninguna limitación en cuanto al alcance de uso o funcionalidad de las realizaciones preferidas de la invención descritas en este documento. Independientemente, el nodo 6010 informático en la nube es capaz de implementarse y/o realizar cualquiera de las funciones expuestas anteriormente.

En el nodo 6010 informático en la nube hay un sistema/servidor 6012 informático, que funciona con numerosos otros entornos o configuraciones de sistemas informáticos de propósito general o de propósito especial. Ejemplos de sistemas informáticos conocidos, entornos y/o configuraciones que pueden ser adecuados para utilizar con el sistema/servidor 6012 informático incluye, entre otros, sistemas de ordenador personal, sistemas de ordenador servidor, clientes simples, clientes complejos, dispositivos portátiles o de mano, sistemas multiprocesador, sistemas basados en microprocesador, decodificadores, electrónica de consumo programable, PC en red, sistemas de miniordenador, sistemas informáticos de mainframe y entornos informático en la nube distribuidos que incluyen cualquiera de los sistemas o dispositivos anteriores, y similares.

El sistema/servidor 6012 informático puede describirse en el contexto general de instrucciones ejecutables del sistema informático, tales como módulos de programa, ejecutados 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 6012 informático se puede practicar en entornos informático en la nube distribuidos en el que las tareas son realizadas por dispositivos de procesamiento remoto que están vinculados a través de una red de comunicaciones. En un entorno informático en la nube distribuido, los módulos de programa pueden ubicarse en medios de almacenamiento de sistemas informáticos locales y remotos, que incluyen dispositivos de almacenamiento de memoria.

Como se muestra en la Figura 27, el sistema/servidor 6012 informático en el nodo 6010 informático en la nube se muestra en forma de un dispositivo informático de propósito general. Los componentes del sistema/servidor 6012 informático pueden incluir, entre otros, uno o más procesadores o unidades de procesamiento 6016, una memoria 6028 del sistema y un bus 6018 que acopla varios componentes del sistema, incluida la memoria 6028 del sistema al procesador 6016.

El bus 6018 representa uno o más de varios tipos de estructuras de bus, que incluye un bus de memoria o controlador de memoria, un bus periférico, un puerto de gráficos acelerado y un procesador o bus local que utiliza cualquiera de una variedad de arquitecturas de bus. A modo de ejemplo, y sin limitación, dichas arquitecturas incluyen el bus de Arquitectura Estándar de la Industria (ISA), el bus de Arquitectura de Microcanales (MCA), el bus ISA Mejorado (EISA), el bus local de la Asociación de Estándares de Video Electrónica (VESA) y el bus de Interconexión de Componentes Periféricos (PCI).

El sistema/servidor 6012 informático normalmente incluye una variedad de medios legibles del sistema de ordenador. Dichos medios pueden ser cualquier medio disponible al que pueda acceder el sistema/servidor 6012 informático, e incluye medios volátiles y no volátiles, medios extraíbles y no extraíbles.

5 La memoria 6028 del sistema puede incluir medios legibles del sistema informático en forma de memoria volátil, como la memoria 6030 de acceso aleatorio (RAM) y/o la memoria 6032 caché. El sistema/servidor 6012 informático puede incluir además otros medios de almacenamiento de sistemas informáticos extraíbles/no extraíbles, volátiles/no volátiles. Solo a modo de ejemplo, el sistema 6034 de almacenamiento se puede proporcionar para leer y escribir en un medio magnético no volátil y no extraíble (no mostrado y normalmente se denomina "disco duro"). Aunque no se muestra, una unidad de disco magnético para leer y escribir en un disco magnético no volátil extraíble (por ejemplo, un "disquete") y una unidad de disco óptico para leer o escribir en una unidad óptica no volátil extraíble se puede proporcionar un disco tal como un CD-ROM, DVD-ROM u otro medio óptico. En tales casos, cada uno puede conectarse al bus 6018 mediante una o más interfaces de medios de datos. Como se representa y describirá más adelante, 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 preferidas de la presente invención.

El programa/funcionalidad 6040, que tiene un conjunto (al menos uno) de módulos 6042 de programa, puede almacenarse en la memoria 6028 a modo de ejemplo, sin limitación, así como un sistema operativo, uno o más programas de aplicación, otros módulos de programa, y datos del programa. Cada uno de los sistemas operativos, uno o más programas de aplicación, otros módulos de programa y datos de programa o alguna combinación de los mismos, puede incluir una implementación de un entorno de red. Los módulos 6042 de programa generalmente llevan a cabo las funciones y/o metodologías de realizaciones preferidas de la presente invención como se describe en el presente documento.

El sistema/servidor 6012 informático también puede comunicarse con uno o más dispositivos 6014 externos tales como un teclado, un dispositivo indicador, una pantalla 6024, etc.; uno o más dispositivos que permiten a un usuario interactuar con el sistema/servidor 6012 informático; y/o cualquier dispositivo (por ejemplo, tarjeta de red, módem, etc.) que permita que el sistema/servidor 6012 informático se comunique con uno o más dispositivos informáticos. Dicha comunicación puede ocurrir a través de las interfaces 6022 de entrada/salida (I/O). Aún así, el sistema/servidor 6012 informático se puede comunicar con una o más redes, tales como una red de área local (LAN), una red de área amplia general (WAN) y/o una red pública (por ejemplo, Internet) a través del adaptador 6020 de red. Como se muestra, el adaptador 6020 de red se comunica con los otros componentes del sistema/servidor 6012 informático a través del bus 6018. Se debe entender que, aunque no se muestra, otros componentes de hardware y/o software se podrían utilizar junto con el sistema/servidor 6012 informático. Los ejemplos incluyen, pero no se limitan: microcódigo, controladores de dispositivos, unidades de procesamiento redundantes, matrices de unidades de disco externas, sistemas RAID, unidades de cinta y sistemas de almacenamiento de archivos de datos, etc.

Con referencia ahora a la Figura 28, se representa el entorno 6050 ilustrativo informático en la nube. Como se muestra, el entorno 6050 informático en la nube comprende uno o más nodos informáticos en la nube 6010 con los que los dispositivos informáticos locales utilizados por los consumidores de la nube, como, por ejemplo, el asistente digital personal (PDA) o el teléfono 6054A celular, el ordenador 6054B de escritorio, el ordenador 6054C portátil, y/o el sistema 6054N informático del automóvil se puede comunicar. Los nodos 6010 se pueden comunicar entre sí. Se pueden agrupar (no se muestran) física o virtualmente, en una o más redes, tales como nubes privadas, comunitarias, públicas o híbridas como se describe anteriormente, o una combinación de las mismas. Esto permite que el entorno 6050 informático en la nube ofrezca infraestructura, plataformas y/o software como servicios para los cuales un consumidor de la nube no necesita mantener recursos en un dispositivo informático local. Se entiende que los tipos de dispositivos informáticos 6054A-N mostrados en la Figura 28 están destinados a ser solo ilustrativos y que los nodos 6010 informáticos y el entorno 6050 informático en la nube se pueden comunicar con cualquier tipo de dispositivo computarizado a través de cualquier tipo de red y/o conexión direccionable de red (por ejemplo, utilizando un navegador web).

Con referencia ahora a la Figura 29, se muestra un conjunto de capas de abstracción funcionales proporcionadas por el entorno 6050 informático en la nube (Figura 28). Se debe entender de antemano que los componentes, capas y funciones mostradas en la Figura 29 están destinadas a ser solo ilustrativas y las realizaciones preferidas de la invención no están limitadas a los mismos. Como se muestra, se proporcionan las siguientes capas y funciones correspondientes:

La capa 6060 de hardware y software incluye componentes de hardware y software. Los ejemplos de componentes de hardware incluyen mainframes, en un ejemplo, sistemas IBM® zSeries®; Servidores basados en arquitectura RISC (Ordenador de Conjunto de Instrucciones Reducidas), en un ejemplo sistemas IBM pSeries®; Sistemas IBM xSeries®; Sistemas IBM BladeCenter®; dispositivos de almacenamiento; redes y componentes de redes. Los 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 el software de base de datos IBM DB2®. (IBM, zSeries, pSeries, xSeries, BladeCenter, WebSphere y DB2 son marcas comerciales de International Business Machines Corporation registradas en muchas jurisdicciones de todo el mundo).

La capa 6062 de virtualización 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, incluidas redes privadas virtuales; aplicaciones virtuales y sistemas operativos; y clientes virtuales.

5 En un ejemplo, la capa 6064 de gestión puede proporcionar las funciones que se describen a continuación. El aprovisionamiento de recursos proporciona una adquisición dinámica de recursos informáticos y otros recursos que se utilizan para realizar tareas dentro del entorno informático en la nube. La medición y la fijación de precios brindan seguimiento de costes a medida que los recursos se utilizan dentro del entorno informático en la nube, y la facturación por el consumo de estos recursos. En un ejemplo, estos recursos pueden comprender licencias de software de aplicación. La seguridad proporciona verificación de identidad para consumidores y tareas en la nube, así como protección para datos y otros recursos. El portal de usuarios proporciona acceso al entorno informático en la nube para consumidores y administradores de sistemas. La gestión de nivel de servicio proporciona la asignación y gestión de recursos informático en la nube de tal manera que se cumplan los niveles de servicio requeridos. La planificación y el cumplimiento del Acuerdo de Nivel de Servicio (SLA) proporcionan una predisposición y adquisición de recursos informático en la nube para los cuales se anticipa un requerimiento futuro de acuerdo con un SLA.

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

25 La terminología utilizada en este documento tiene el propósito de describir únicamente los preferidos particulares y no pretende ser limitante. Como se utiliza en este documento, las formas singulares “un”, “una” y “el” están destinadas a incluir también las formas plurales, a menos que el contexto indique claramente lo contrario. Se entenderá adicionalmente que los términos “comprende” y/o “que comprende”, cuando se utilizan en esta especificación, especifican la presencia de características, números enteros, etapas, operaciones, elementos y/o componentes establecidos, pero no excluyen la presencia o la adición de una o más características, números enteros, pasos, operaciones, elementos, componentes y/o grupos de los mismos.

30 Las estructuras, materiales, actos y equivalentes correspondientes de todos los medios o elementos de función de paso más en las reivindicaciones a continuación, si las hay, tienen la intención de incluir 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 preferidas se ha presentado con fines ilustrativos y descriptivos, pero no pretende ser exhaustiva o limitada en la forma divulgada. Muchas modificaciones y variaciones serán evidentes para aquellos expertos en la técnica. La realización se eligió y describió para explicar mejor varios aspectos y la aplicación práctica, y para permitir que otros expertos en la técnica entiendan varias realizaciones preferidas con diversas modificaciones que sean adecuadas para el uso particular contemplado.

40

**REIVINDICACIONES**

1. Un método para inicializar un sistema (122, 220) operativo de un programa utilitario de control en un entorno informático, el método comprende:

5 realizar, mediante un procesador (202), una secuencia (124, 222) de arranque para arrancar el sistema operativo en una de una primera configuración arquitectónica o una segunda configuración arquitectónica, en la que en la primera configuración arquitectónica el sistema operativo se inicializa (900) en un primer modo arquitectónico, el primer modo arquitectónico que comprende un primer conjunto de instrucciones, y luego procesar (912) conmutadores (904, 906) a un segundo modo arquitectónico, el segundo modo arquitectónico comprende un segundo conjunto de instrucciones diferente del primer conjunto de instrucciones arquitectónicas, en el que en la segunda configuración arquitectónica el sistema operativo se inicializa (1000, 1002) y se procesa (1004) en el segundo modo arquitectónico, y en el que la realización de la secuencia de arranque comprende:

15 ejecutar una o más instrucciones independientemente de si el procesador realiza la inicialización (1100, 1102) basada en la primera configuración arquitectónica o la segunda configuración arquitectónica, una o más instrucciones comunes para el primer modo arquitectónico y el segundo modo arquitectónico en el que la una o más instrucciones se definen en el primer modo arquitectónico y el segundo modo arquitectónico, y la ejecución comprende ejecutar una operación (1104) de conmutación para solicitar una conmutación de modos arquitectónicos para el procesador independientemente de si el procesador realiza la inicialización en el primer modo arquitectónico o en el segundo modo arquitectónico, en el que la operación de conmutación se realiza de forma diferente con base en cual arquitectura del primer modo arquitectónico o el segundo modo arquitectónico, el procesador realiza la inicialización, y con base en la ejecución de la operación de conmutación, el programa utilitario de control se arranca en el modo arquitectónico seleccionado;

20 en el que la conmutación comprende enviar una instrucción de conmutación al procesador para conmutar al modo arquitectónico seleccionado.

2. El método de cualquiera de las reivindicaciones precedentes, en el que la realización de la secuencia (124, 222) de arranque comprende:

30 establecer una palabra (500) de estado de programa en un formato del primer modo arquitectónico o el segundo modo arquitectónico, con base en si el procesador (202) realiza la inicialización en el primer modo arquitectónico o en el segundo modo arquitectónico; y preparar a modo conmutación para ejecutar la operación de conmutación.

3. El método de cualquiera de las reivindicaciones precedentes, en el que la ejecución de la operación de conmutación comprende la conmutación desde el primer modo arquitectónico hasta el segundo modo arquitectónico, en el que el procesamiento del sistema (122, 220) operativo está en el segundo modo arquitectónico.

4. El método de cualquiera de las reivindicaciones precedentes, en el que la ejecución de la operación de conmutación comprende solicitar la conmutación desde el segundo modo arquitectónico hasta el segundo modo arquitectónico, y en el que la ejecución de la operación de conmutación resulta en almacenar el estado que indica que el procesador (202) está actualmente en el segundo modo arquitectónico, el estado es tratado como aceptable para un emisor de la operación de conmutación.

5. El método de cualquiera de las reivindicaciones precedentes, en el que el primer modo arquitectónico utiliza direccionamiento de 31 bits y registros de propósito general de 32 bits, y el segundo modo arquitectónico utiliza direccionamiento de 64 bits y registros de propósito general de 64 bits.

6. El método de cualquiera de las reivindicaciones precedentes, en el que la realización de la secuencia (124, 222) de arranque comprende determinar si el sistema (122, 220) operativo se escribe para el segundo modo arquitectónico, y continuar con la realización de la secuencia de arranque con base en el sistema operativo que se escribe para el segundo modo arquitectónico.

7. Un sistema informático para inicializar un sistema operativo en un entorno informático, dicho sistema informático comprende:

una memoria; y un procesador en comunicación con la memoria, en el que el sistema informático se configura para realizar un método, dicho método comprende: realizar, mediante un procesador (202), una secuencia (124, 222) de arranque para arrancar el sistema operativo en una de una primera configuración arquitectónica o una segunda configuración arquitectónica, en la que en la primera configuración arquitectónica el sistema operativo se inicializa (900) en un primer modo arquitectónico, el primer modo arquitectónico comprende un primer conjunto de instrucciones, y luego procesar (912) los conmutadores (904, 906) a un segundo modo arquitectónico, el segundo modo arquitectónico comprende un segundo conjunto de instrucciones diferente del primer conjunto de instrucciones arquitectónicas, en el que en la segunda configuración arquitectónica el sistema operativo se inicializa (1000, 1002) y se procesa (1004) en el segundo modo arquitectónico y en el que la realización de la secuencia de arranque comprende:

- 5 ejecutar una o más instrucciones independientemente de si el procesador realiza la inicialización (1100, 1102) basada en la primera configuración arquitectónica o la segunda configuración arquitectónica, una o más instrucciones comunes para el primer modo arquitectónico y el segundo modo arquitectónico en el que la una o más instrucciones se definen en el primer modo arquitectónico y el segundo modo arquitectónico, y la ejecución comprende ejecutar una operación (1104) de conmutación para solicitar una conmutación de modos arquitectónicos para el procesador independientemente de si el procesador realiza la inicialización en el primer modo arquitectónico o en el segundo modo arquitectónico, en el que la operación de conmutación se realiza de forma diferente con base en cual arquitectura del primer modo arquitectónico o el segundo modo arquitectónico, el procesador realiza la inicialización, y con base en la ejecución de la operación de conmutación, el sistema operativo se arranca en el modo arquitectónico seleccionado;
- 10 en el que la conmutación comprende enviar una instrucción de conmutación al procesador para conmutarse al modo arquitectónico seleccionado.
- 15 8. El sistema informático de la reivindicación 7, en el que la realización de la secuencia (124, 222) de arranque comprende:
- establecer una palabra (500) de estado de programa en un formato del primer modo arquitectónico o el segundo modo arquitectónico, con base en si el procesador (202) realiza la inicialización en el primer modo arquitectónico o en el segundo modo arquitectónico; y
- 20 preparar a modo conmutación para ejecutar la operación de conmutación.
9. El sistema informático de cualquiera de las reivindicaciones 7 o 8, en el que la ejecución de la operación de conmutación comprende la conmutación desde el primer modo arquitectónico hasta el segundo modo arquitectónico, en el que procesamiento del sistema (122, 220) operativo está en el segundo modo arquitectónico.
- 25 10. El sistema informático de cualquiera de las reivindicaciones 7 a 9, en el que la ejecución de la operación de conmutación comprende solicitar la conmutación desde el segundo modo arquitectónico hasta el segundo modo arquitectónico, y en el que la ejecución de la operación de conmutación resulta en almacenar el estado que indica que el procesador (202) está actualmente en el segundo modo arquitectónico, el estado es tratado como aceptable para un emisor de la operación de conmutación.
- 30 11. El sistema informático de cualquiera de las reivindicaciones 7 a 10, en el que el primer modo arquitectónico utiliza direccionamiento de 31 bits y registros de propósito general de 32 bits, y el segundo modo arquitectónico utiliza direccionamiento de 64 bits y registros de propósito general de 64 bits.
- 35 12. El sistema informático de cualquiera de las reivindicaciones 7 a 11, en el que la realización de la secuencia (124, 222) de arranque comprende determinar si el sistema (122, 220) operativo se escribe para el segundo modo arquitectónico, y continuar con la realización de la secuencia de arranque con base en el programa utilitario de control que se escribe para el segundo modo arquitectónico.
- 40 13. Un producto de programa informático para inicializar un programa utilitario de control en un entorno informático, el producto de programa informático comprende:
- un medio de almacenamiento legible por ordenador legible por un circuito de procesamiento y almacenar las instrucciones para la ejecución por el circuito de procesamiento para realizar un método de acuerdo con cualquiera de las reivindicaciones 1 a 6.
- 45 14. Un programa informático almacenado en un medio legible por ordenador y que se puede cargar en la memoria interna de un ordenador digital de cualquiera de las reivindicaciones 8 a 12, que comprende porciones de código de software, cuando dicho programa se corre sobre el ordenador digital, para realizar el método de cualquiera de las reivindicaciones 1 a 6.
- 50

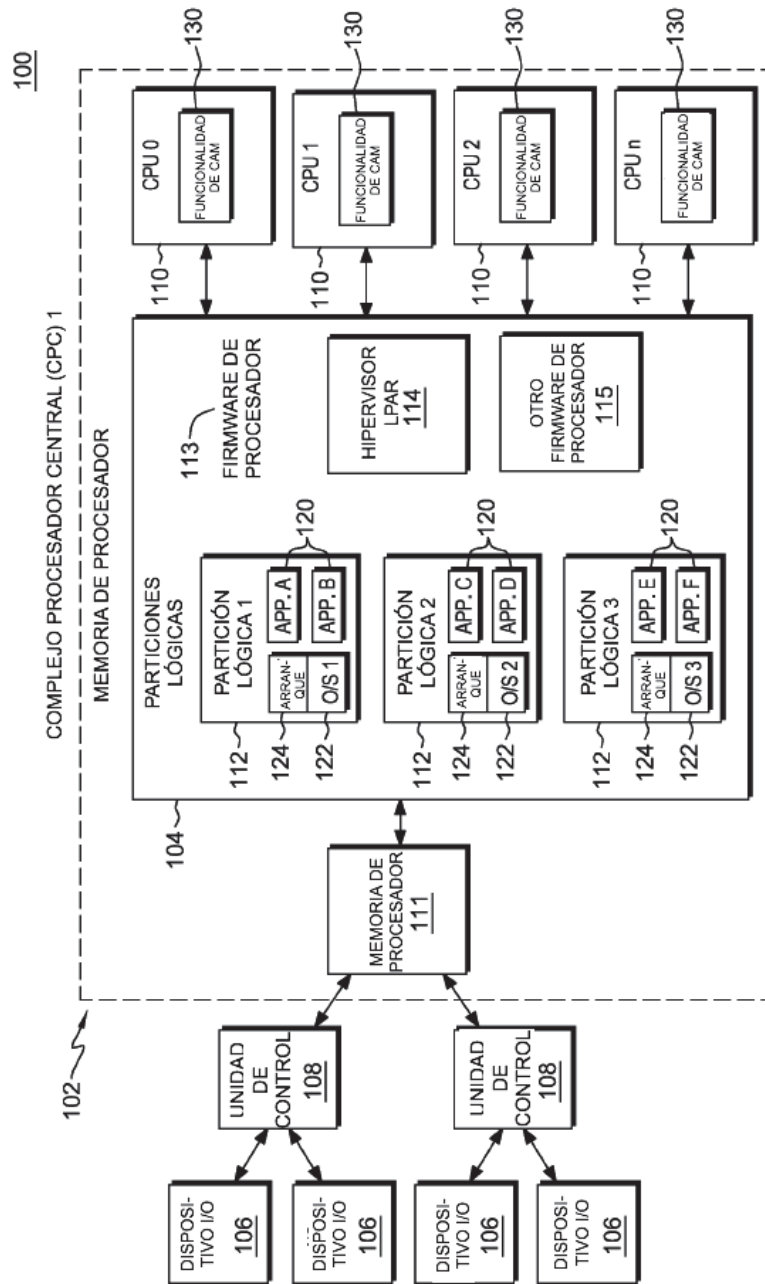


FIG. 1

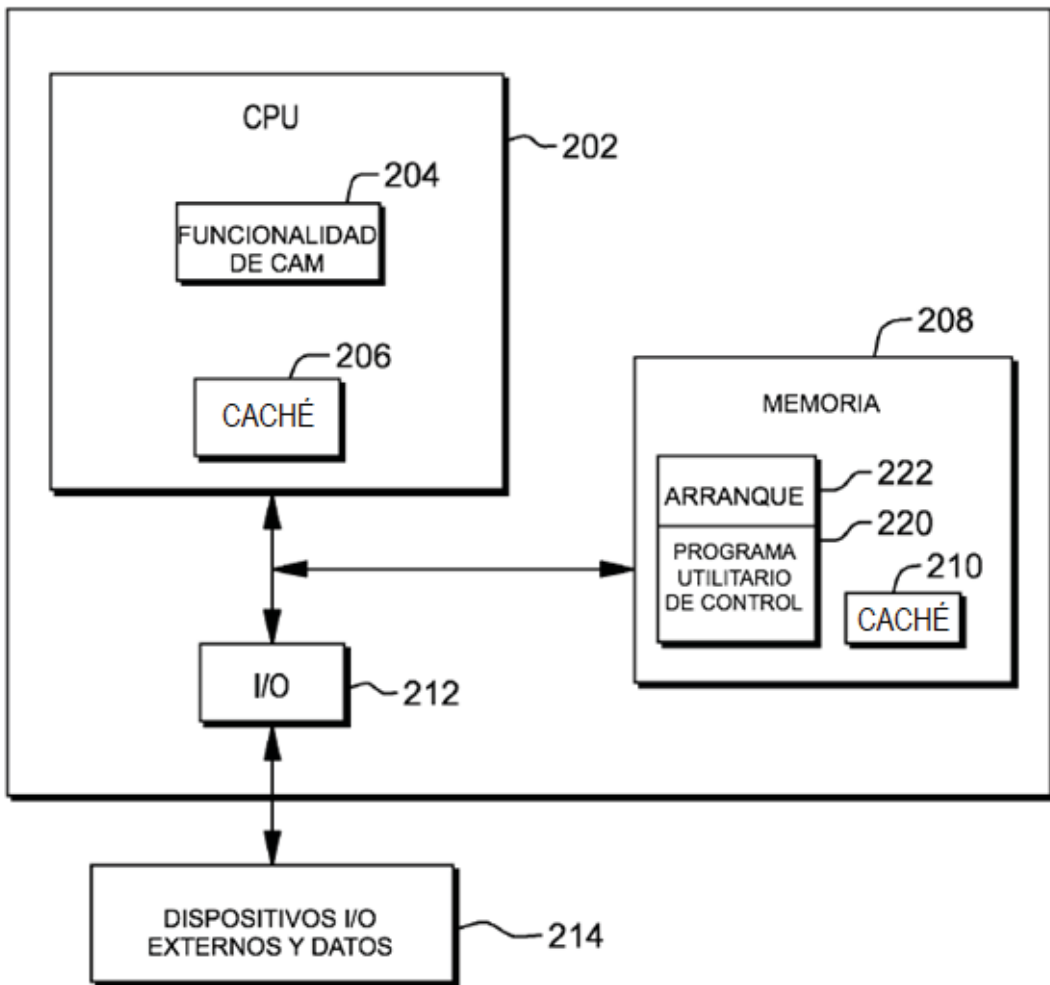


FIG. 2

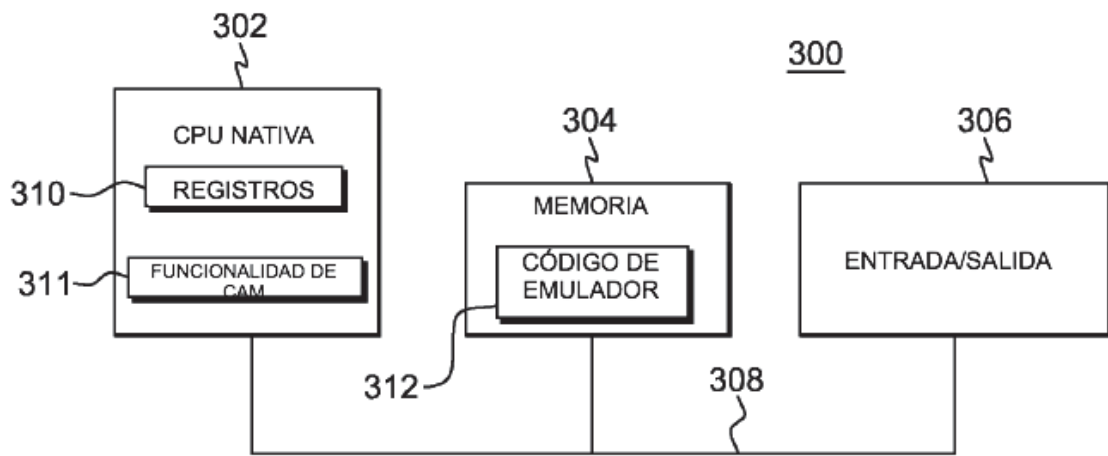


FIG. 3A

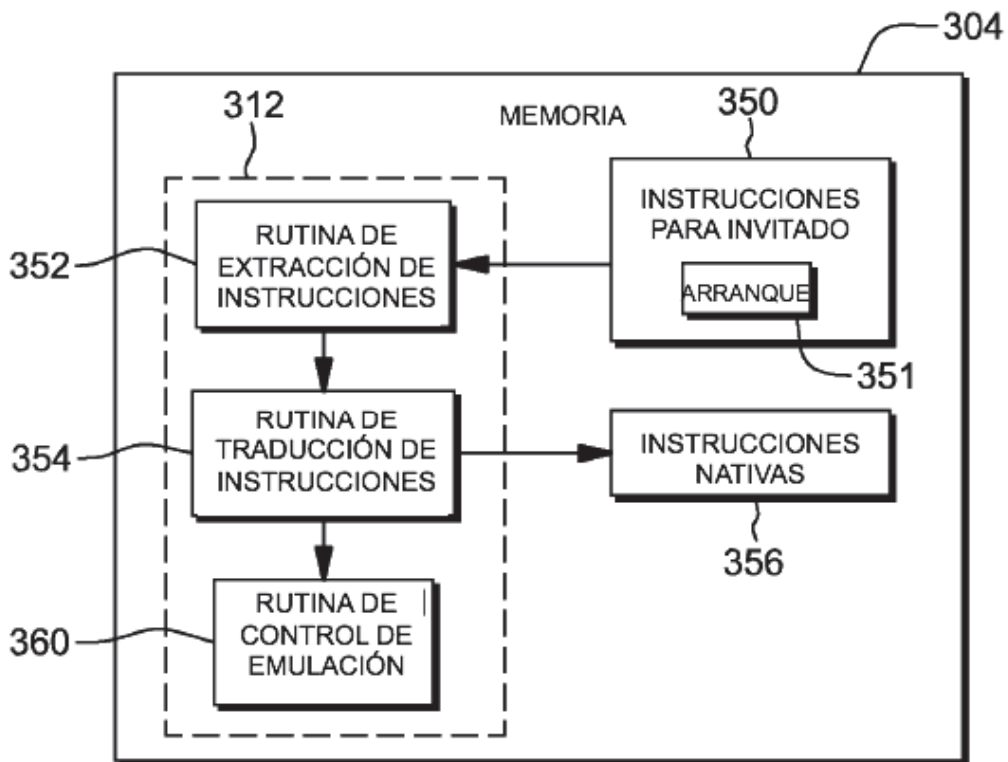


FIG. 3B



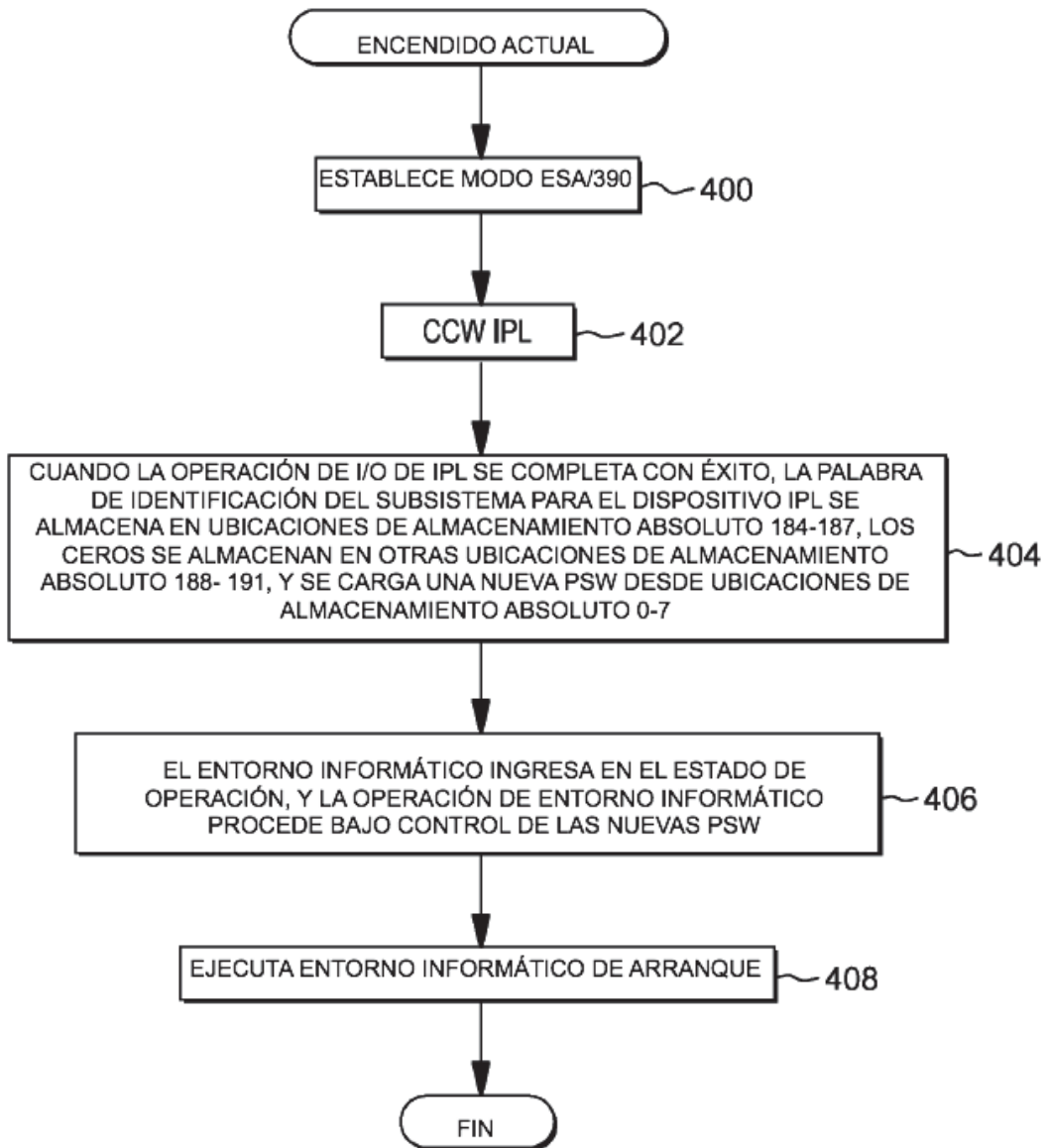


FIG. 4A

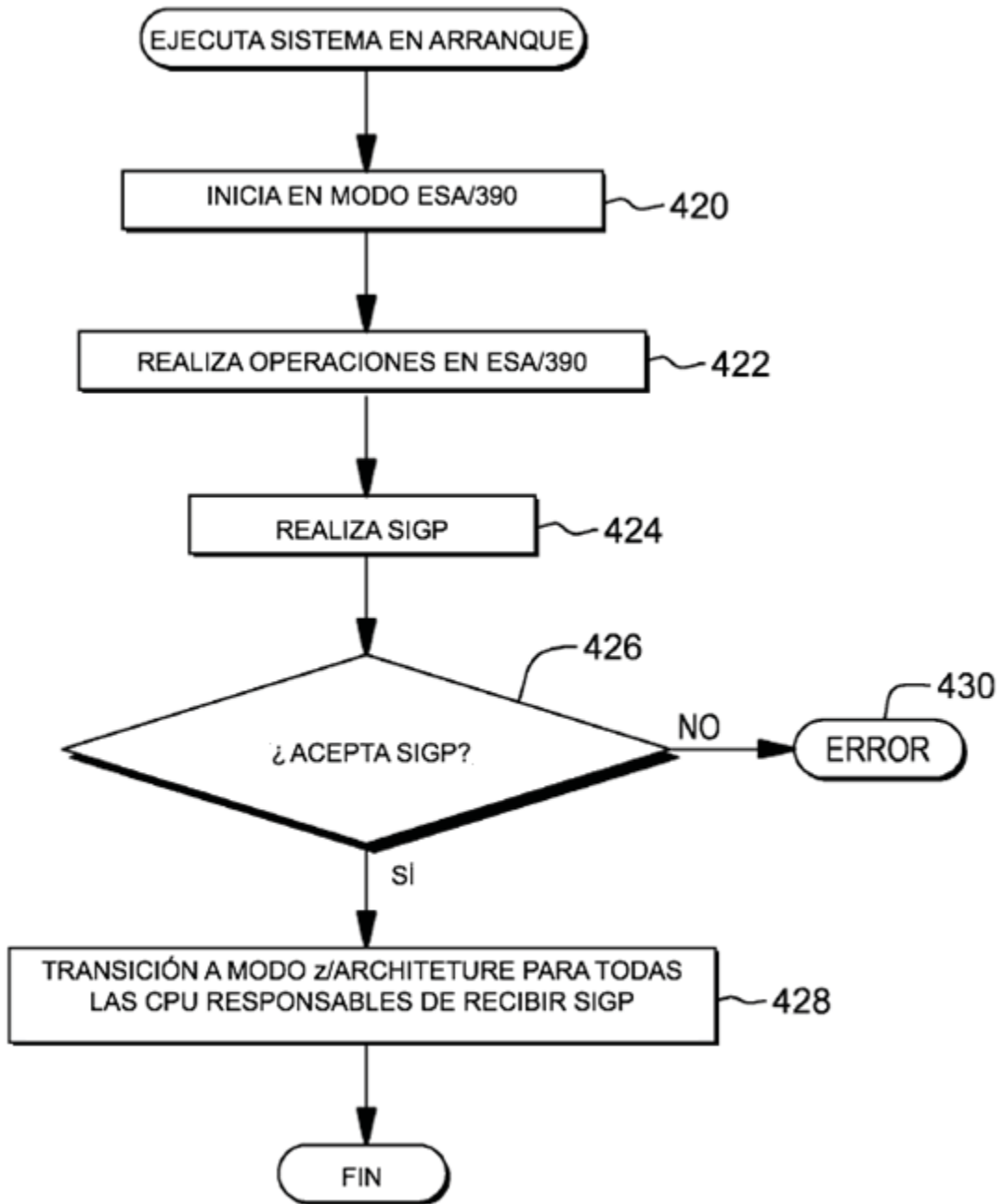


FIG. 4B

500

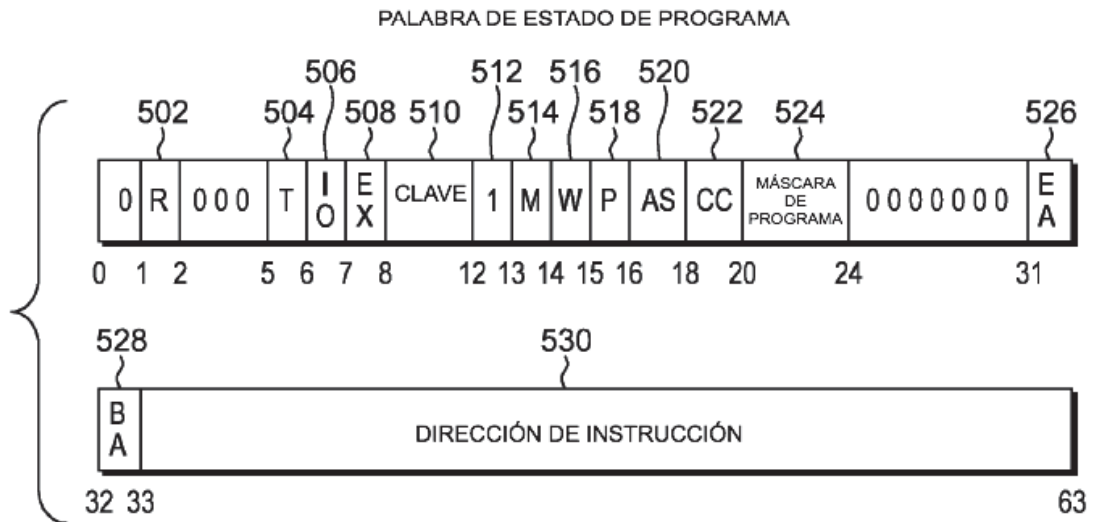


FIG. 5

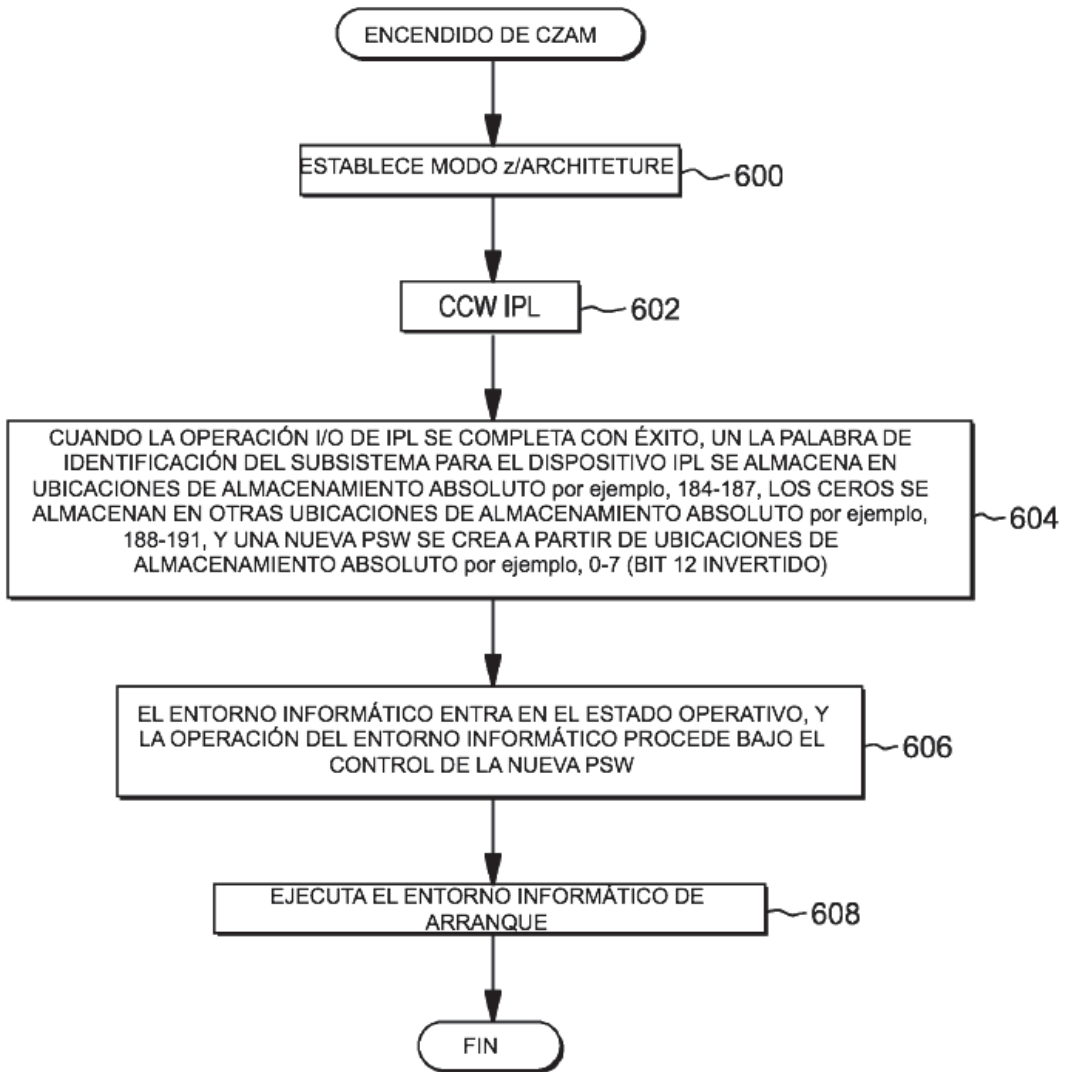


FIG. 6A

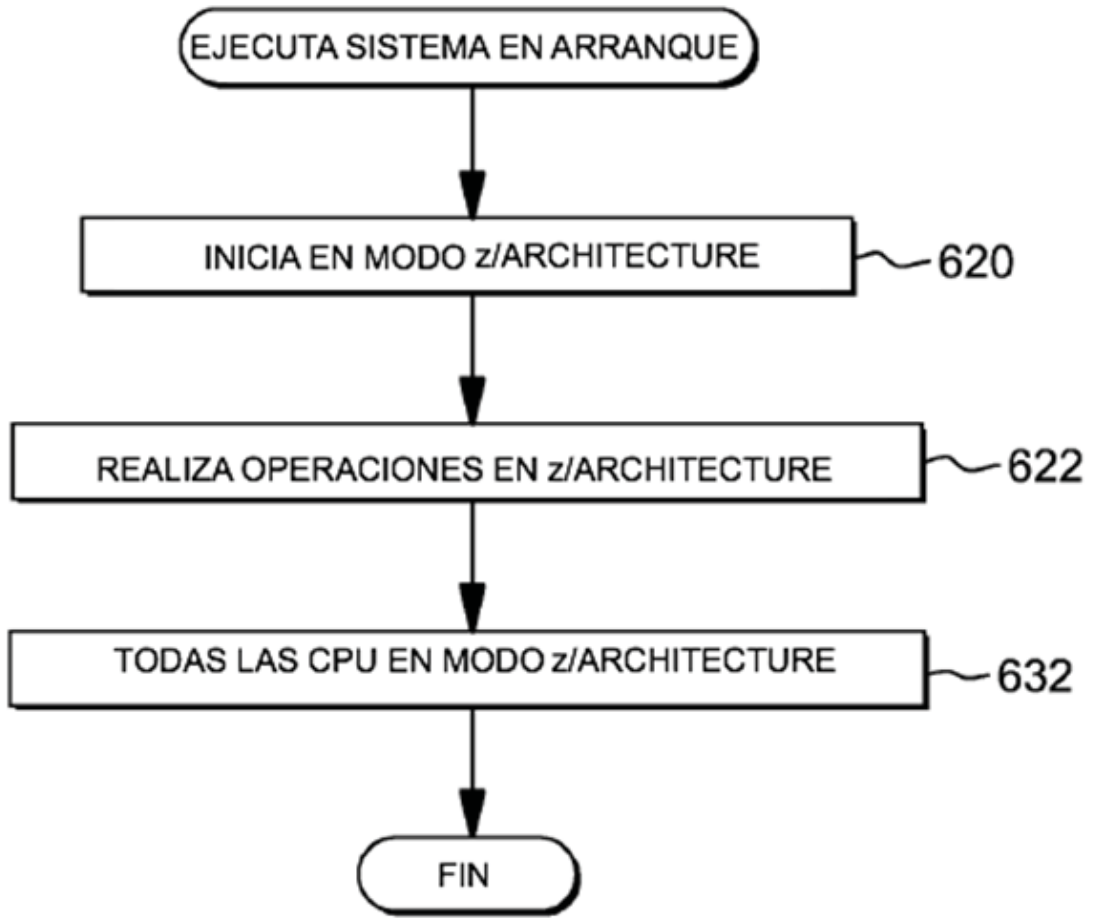
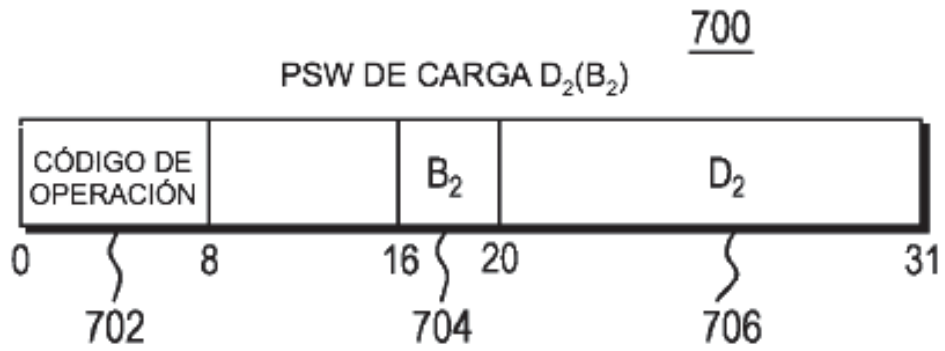
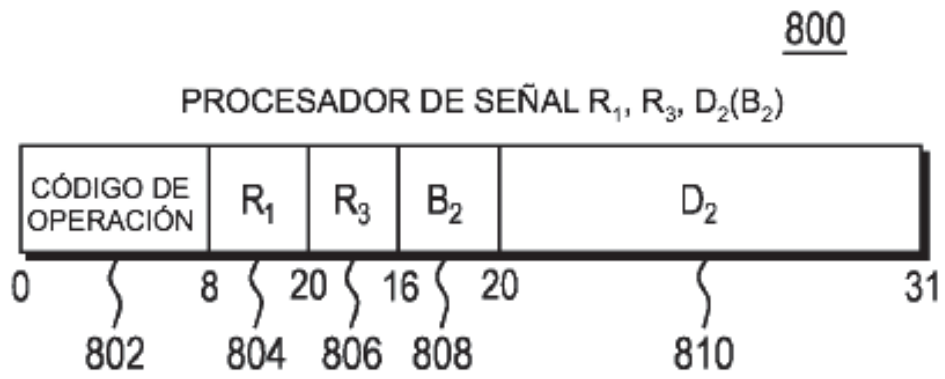


FIG. 6B



**FIG. 7**



**FIG. 8A**

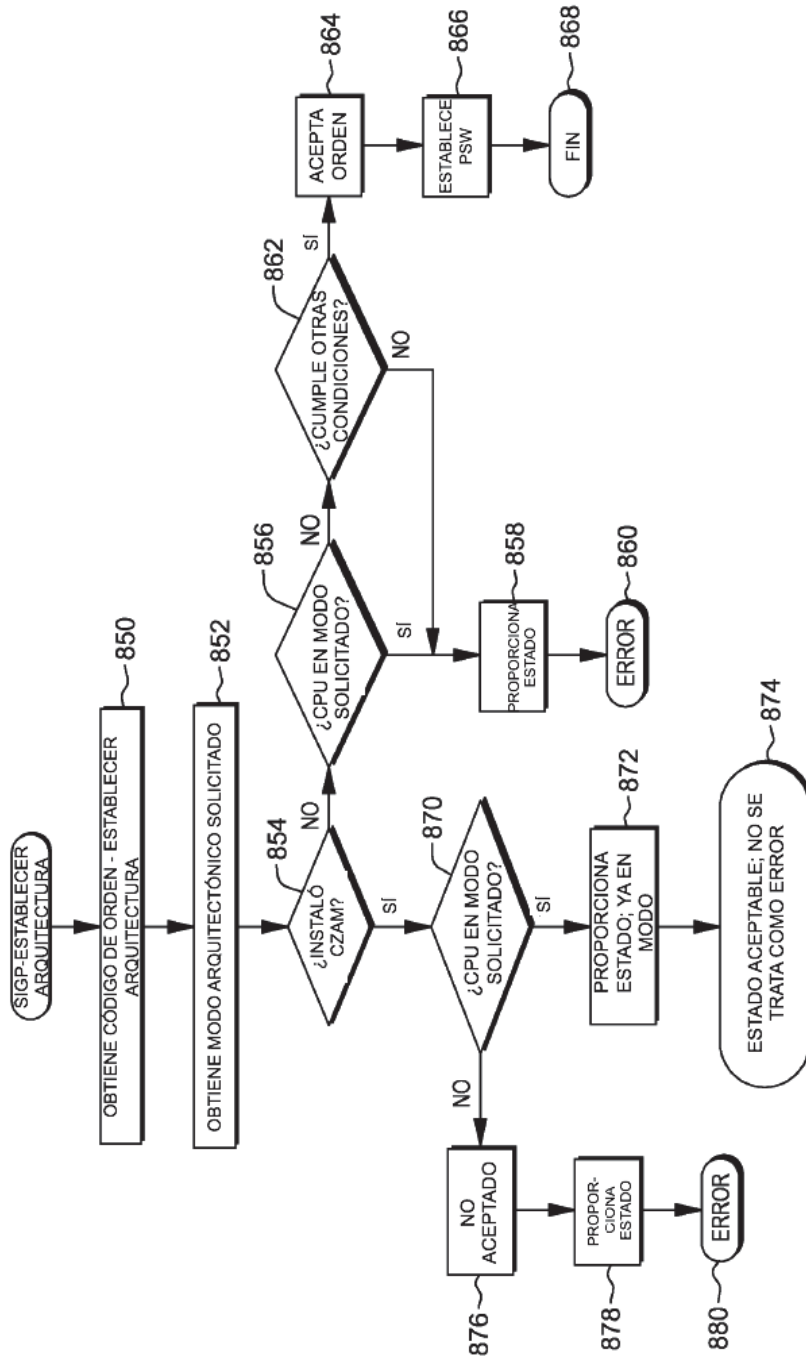


FIG. 8B

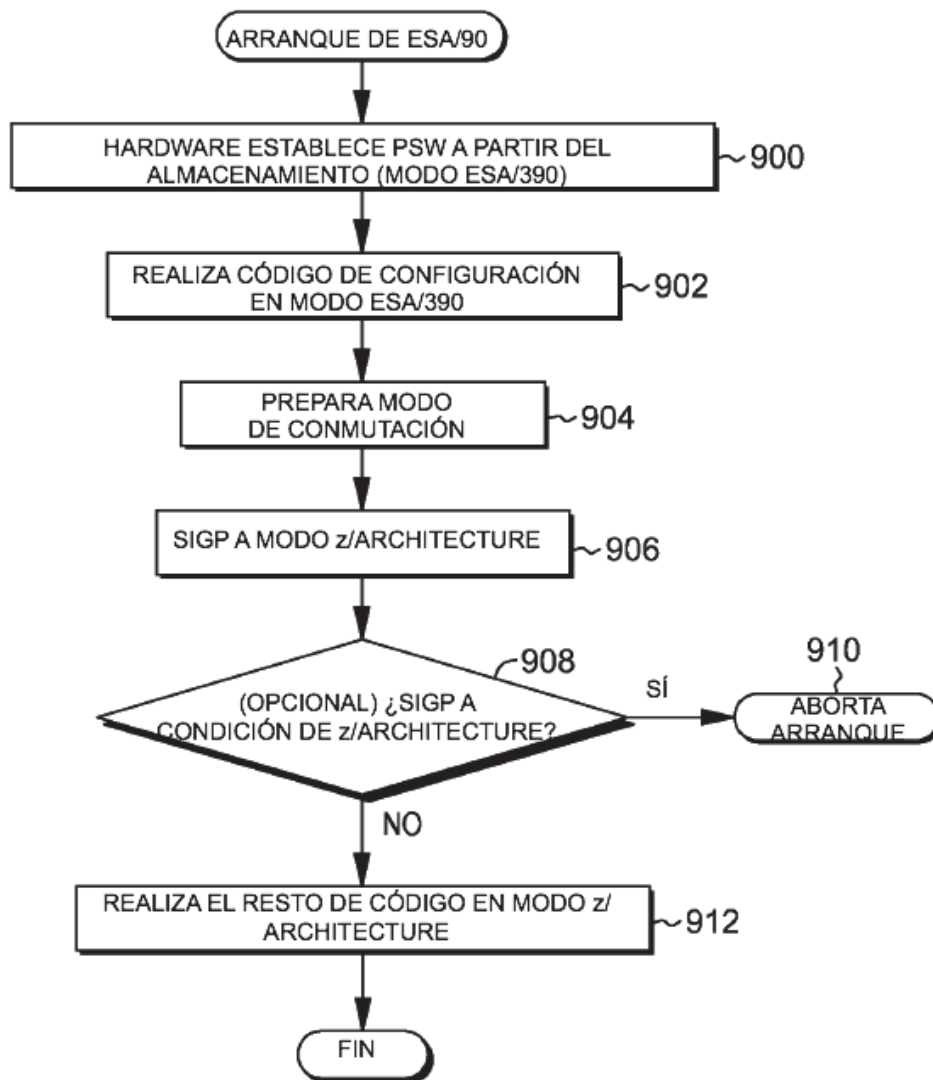


FIG. 9



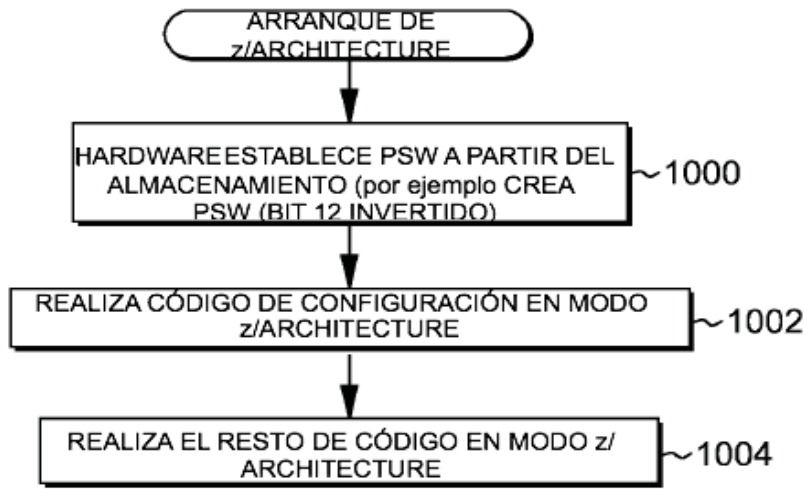


FIG. 10

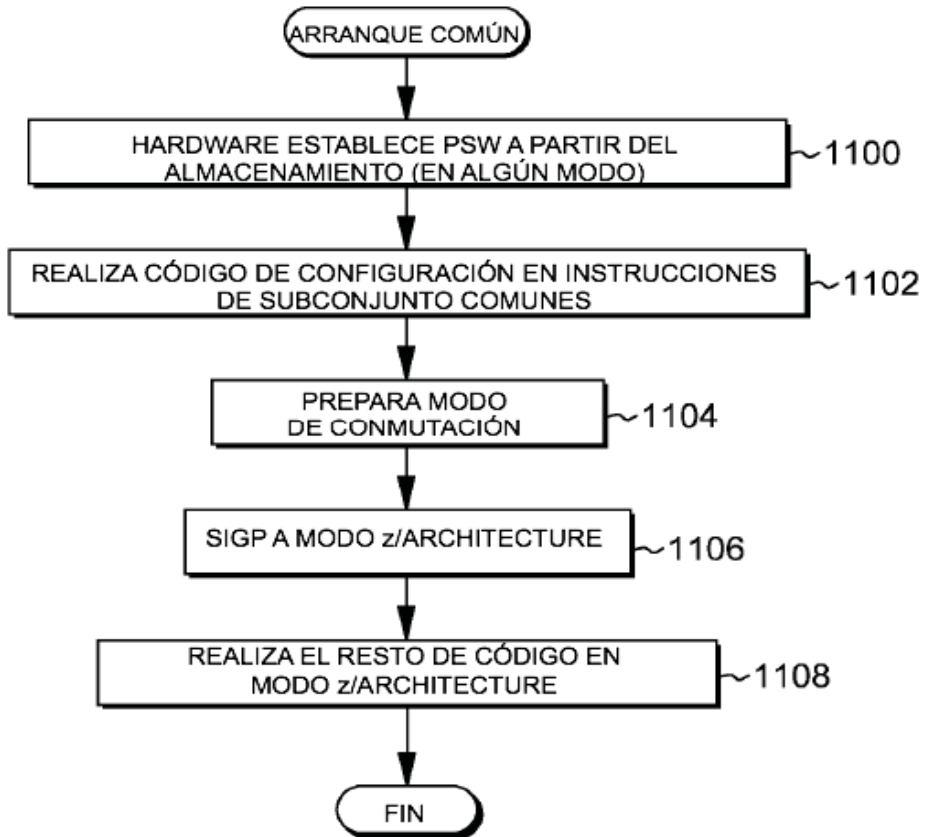


FIG. 11

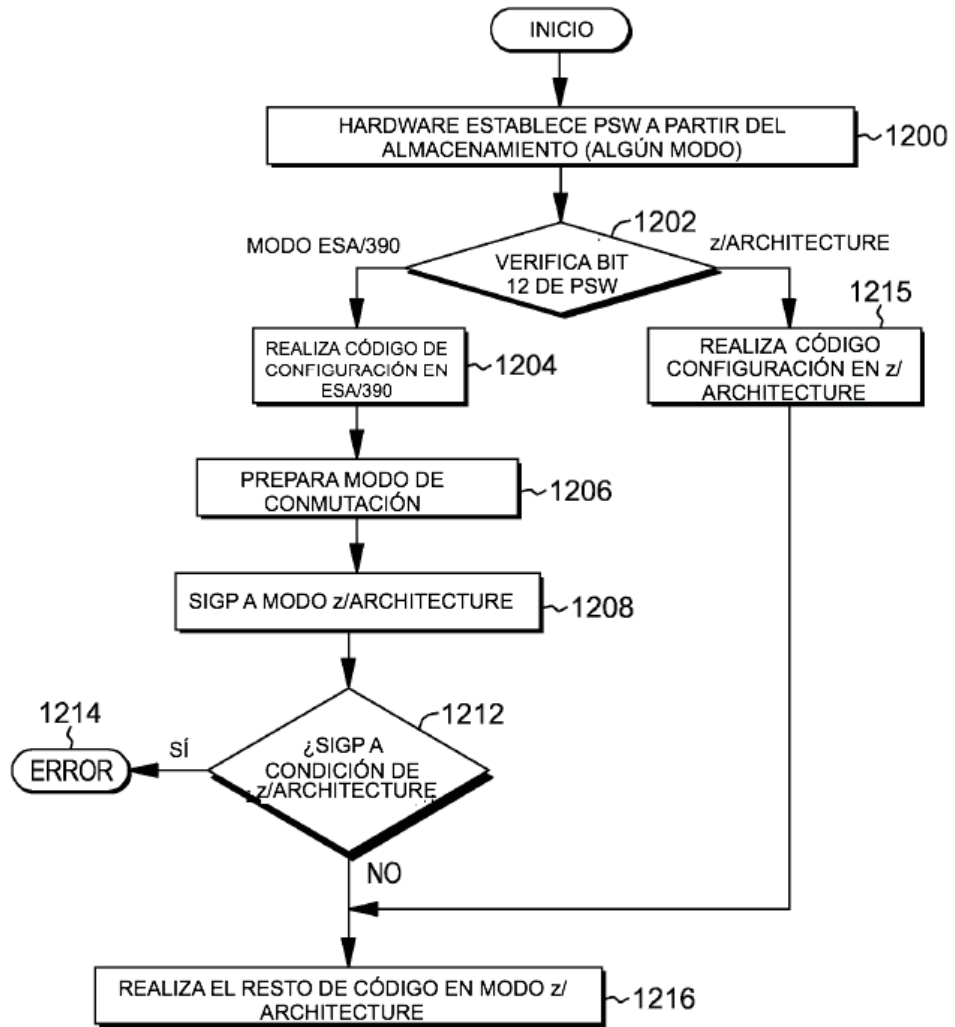


FIG. 12

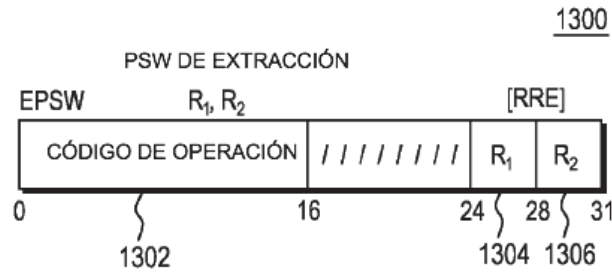


FIG. 13

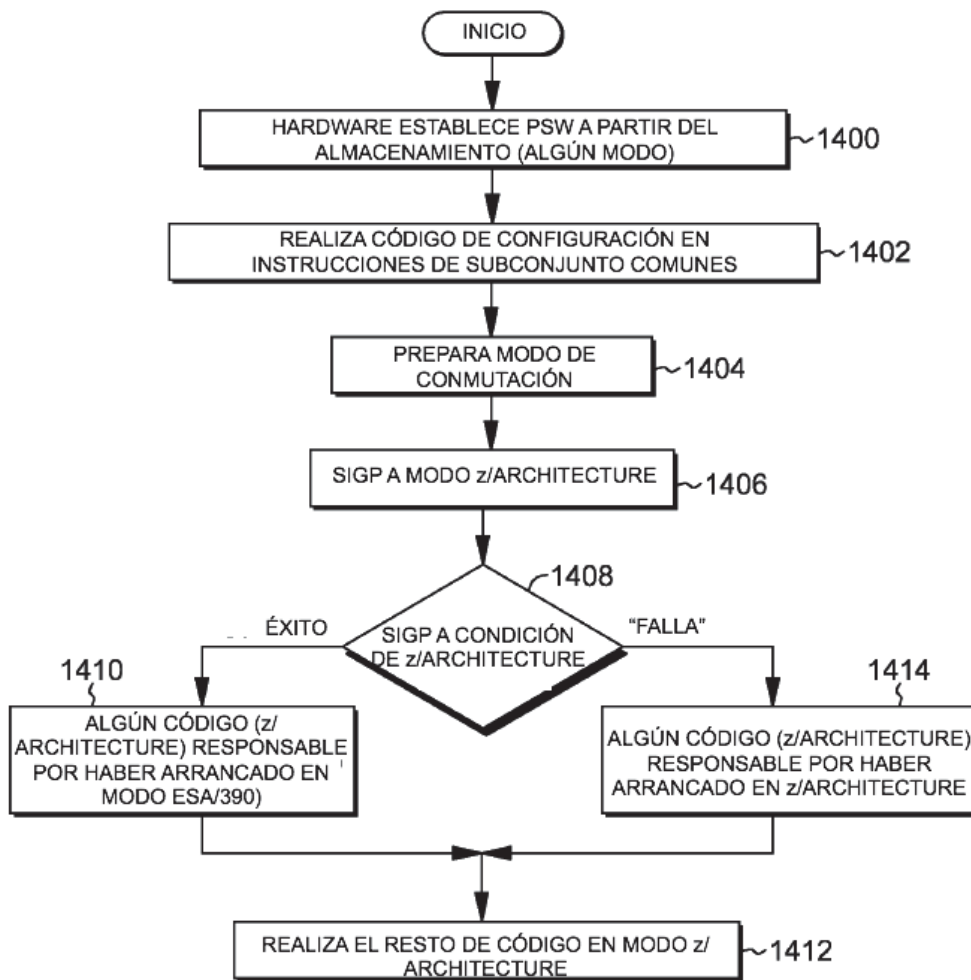


FIG. 14

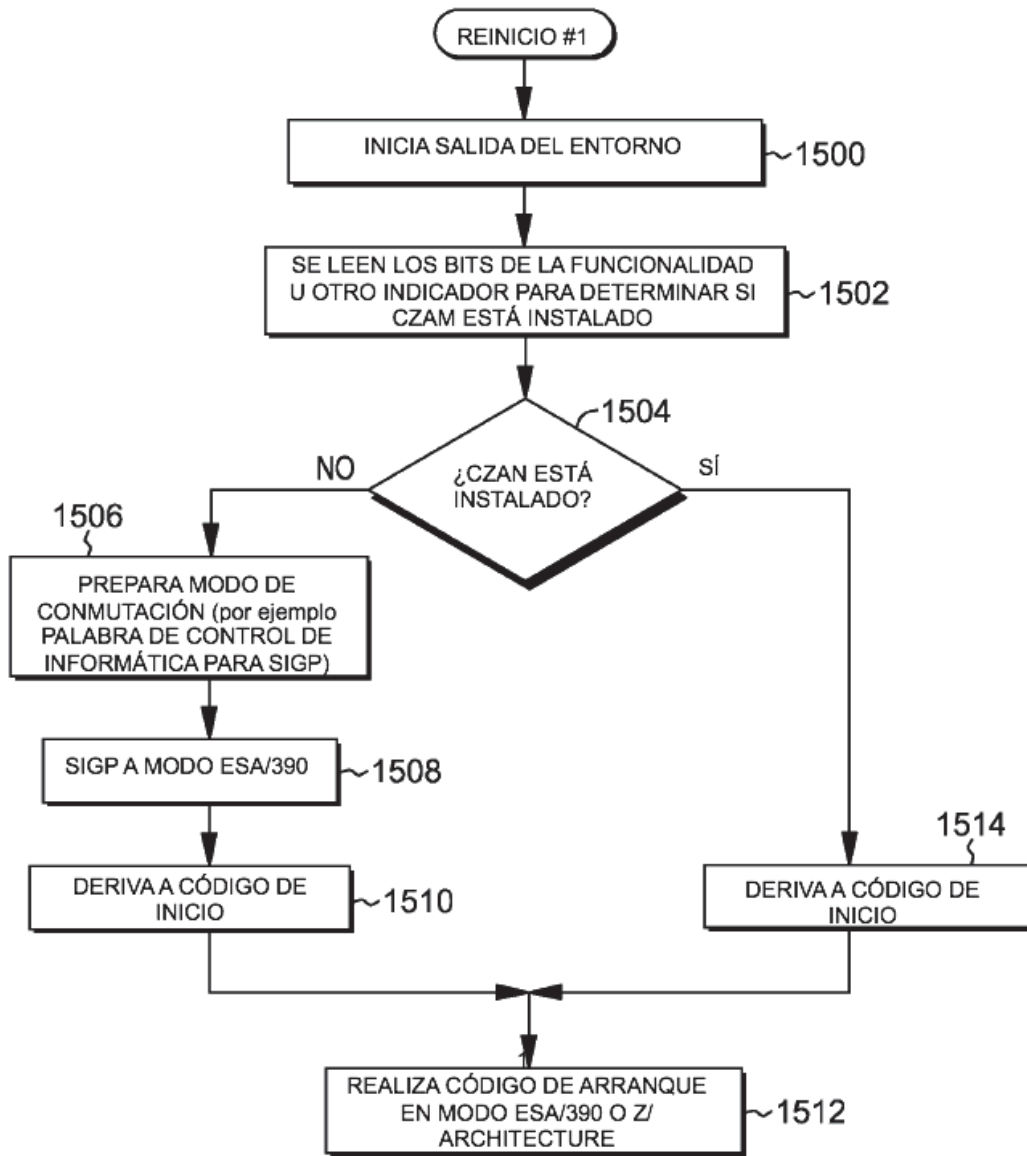


FIG. 15

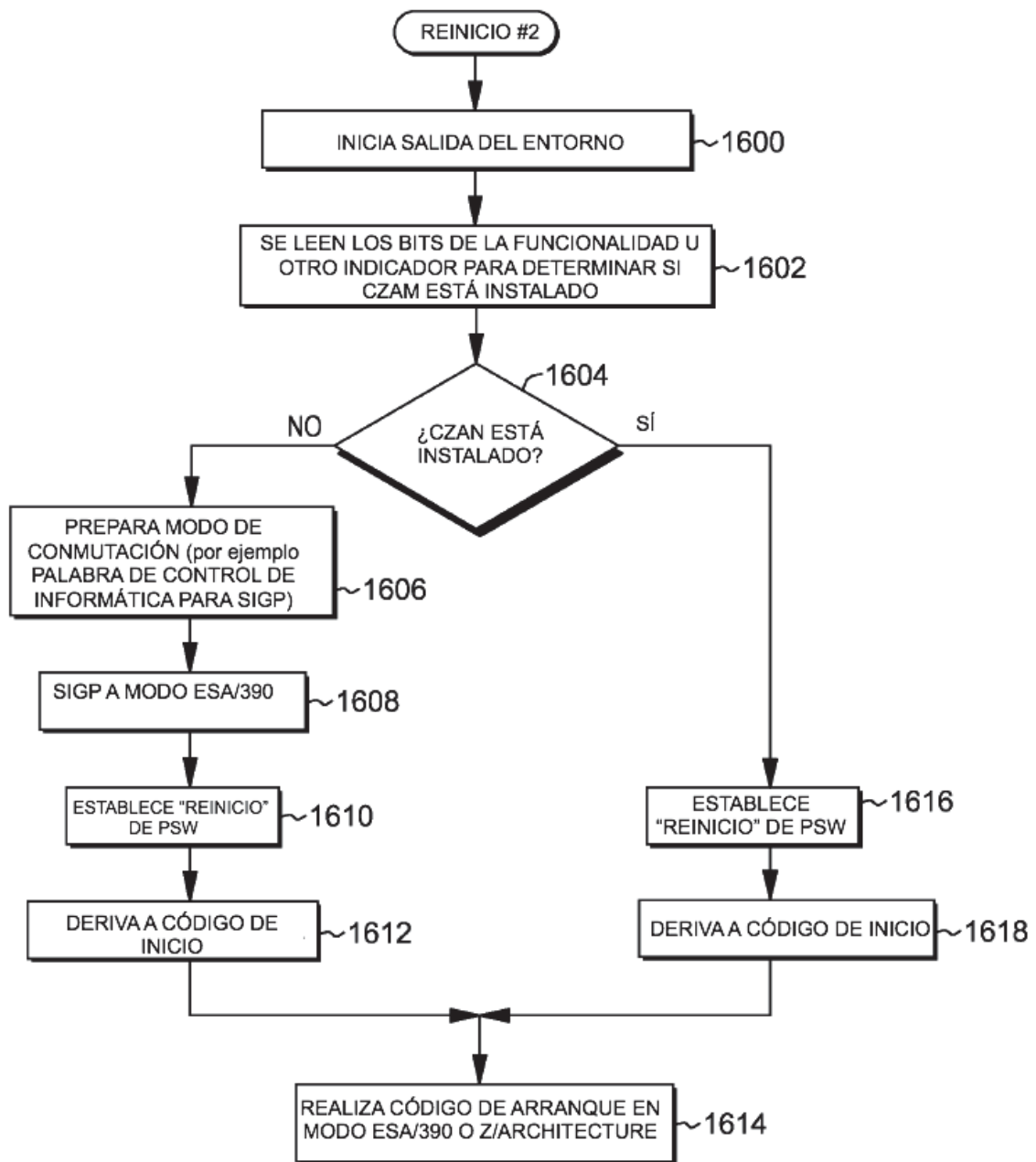


FIG. 16

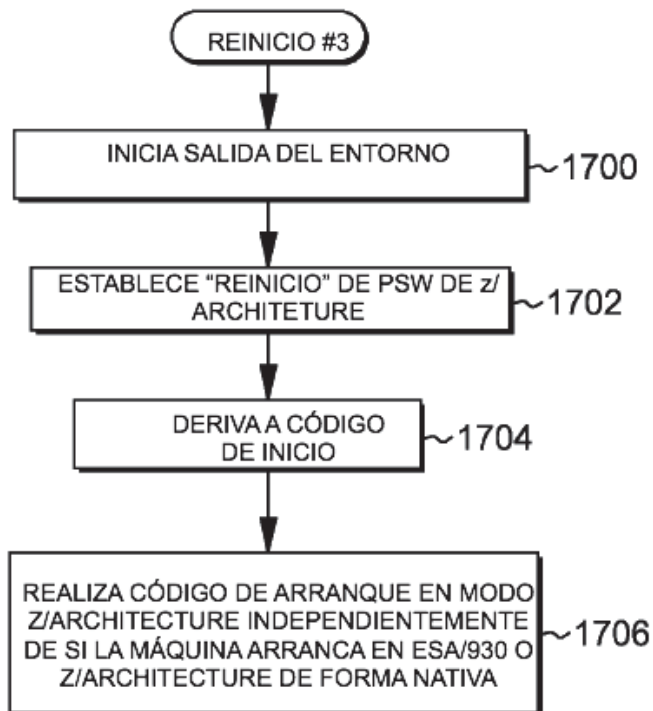


FIG. 17

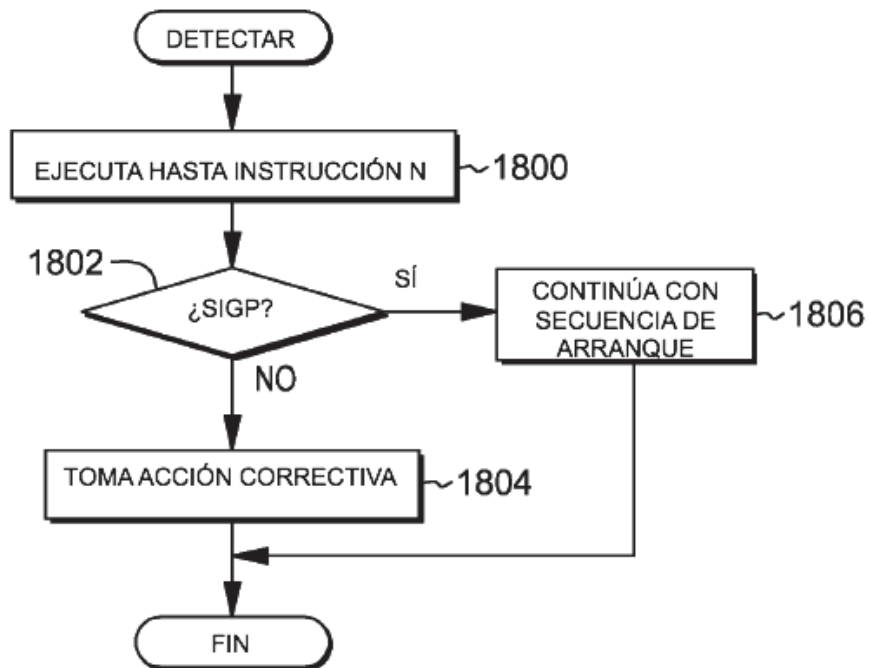


FIG. 18

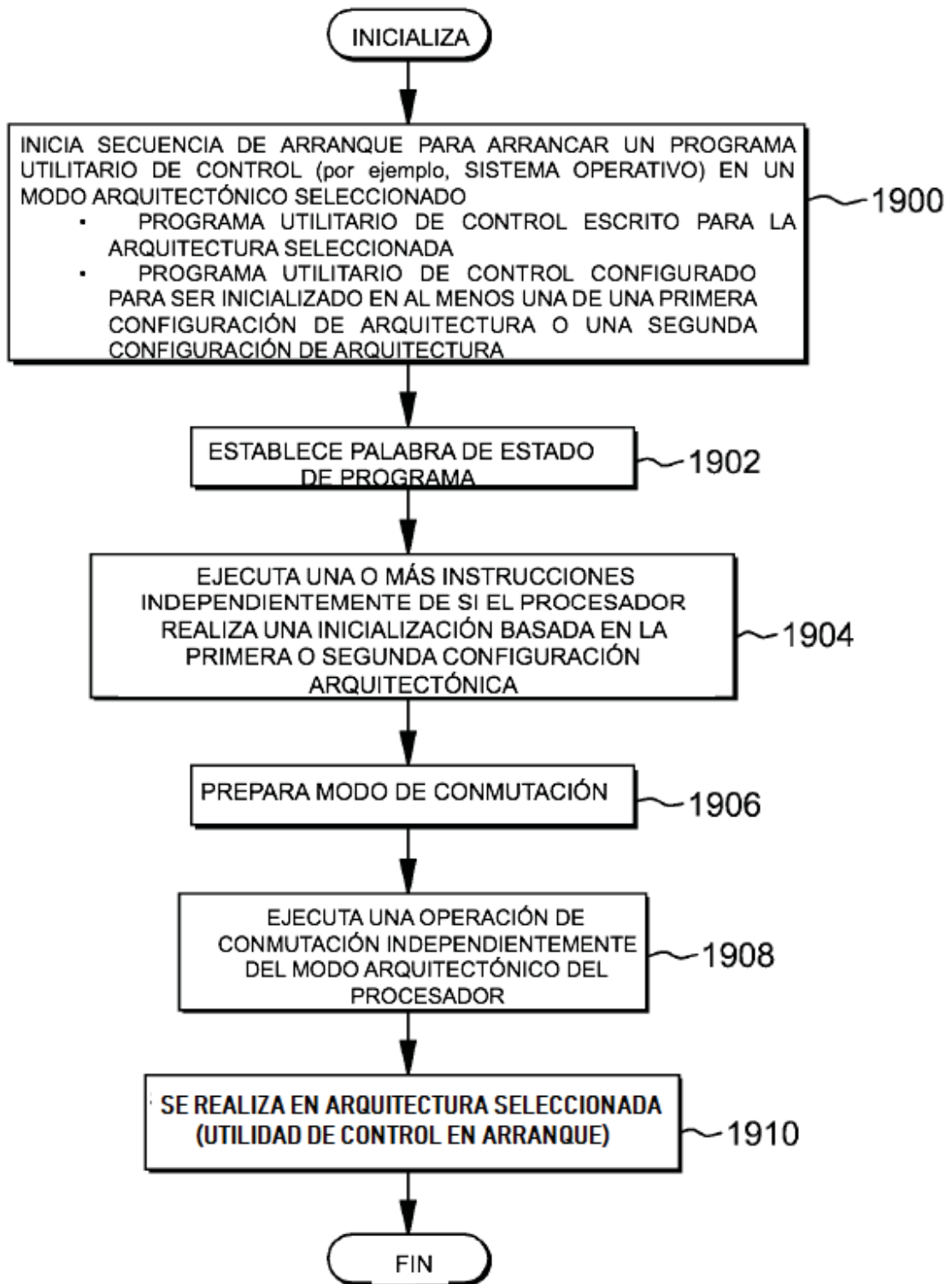


FIG. 19

PRODUCTO DE  
PROGRAMA  
INFORMÁTICO  
2000



FIG. 20



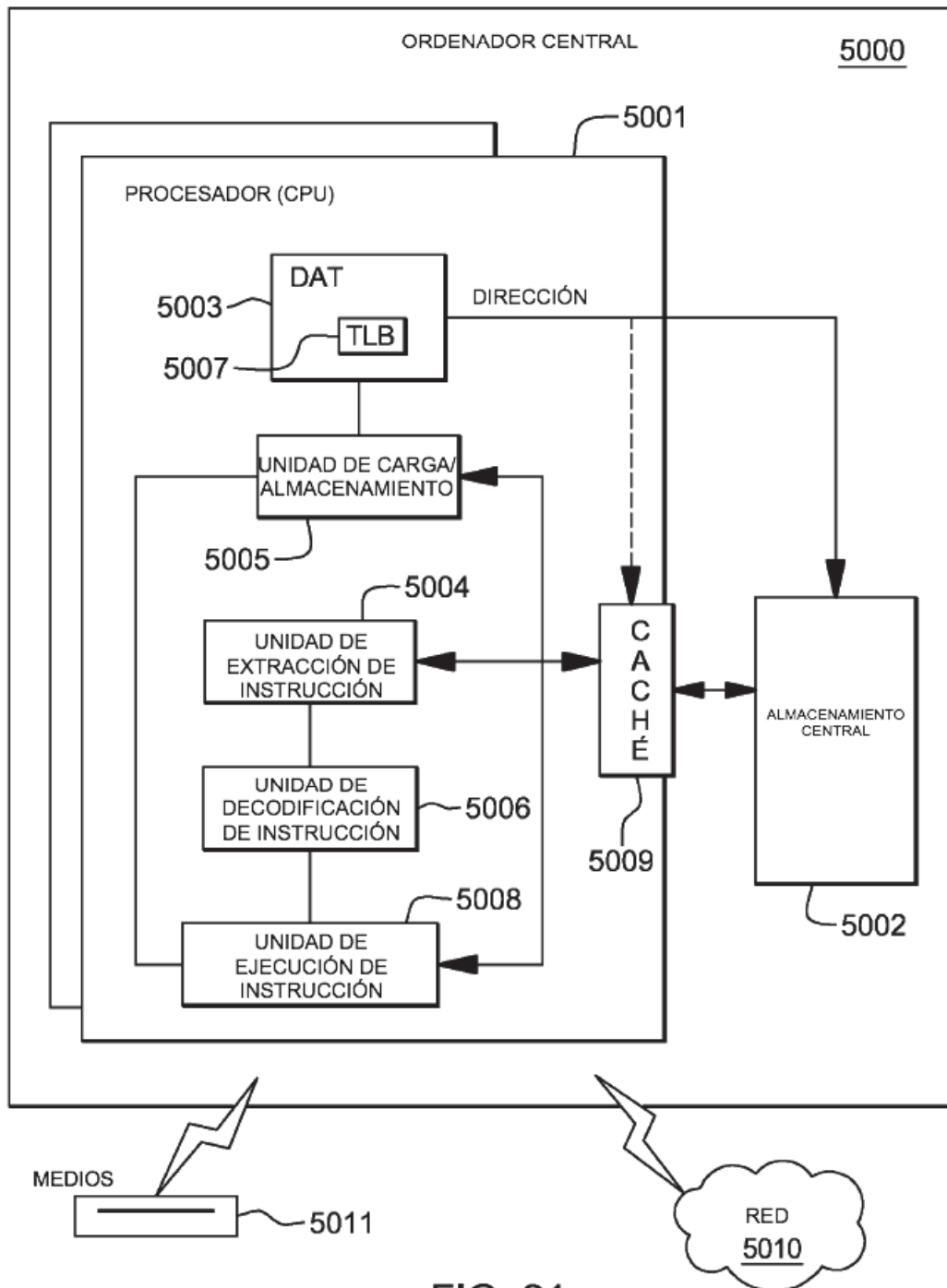


FIG. 21

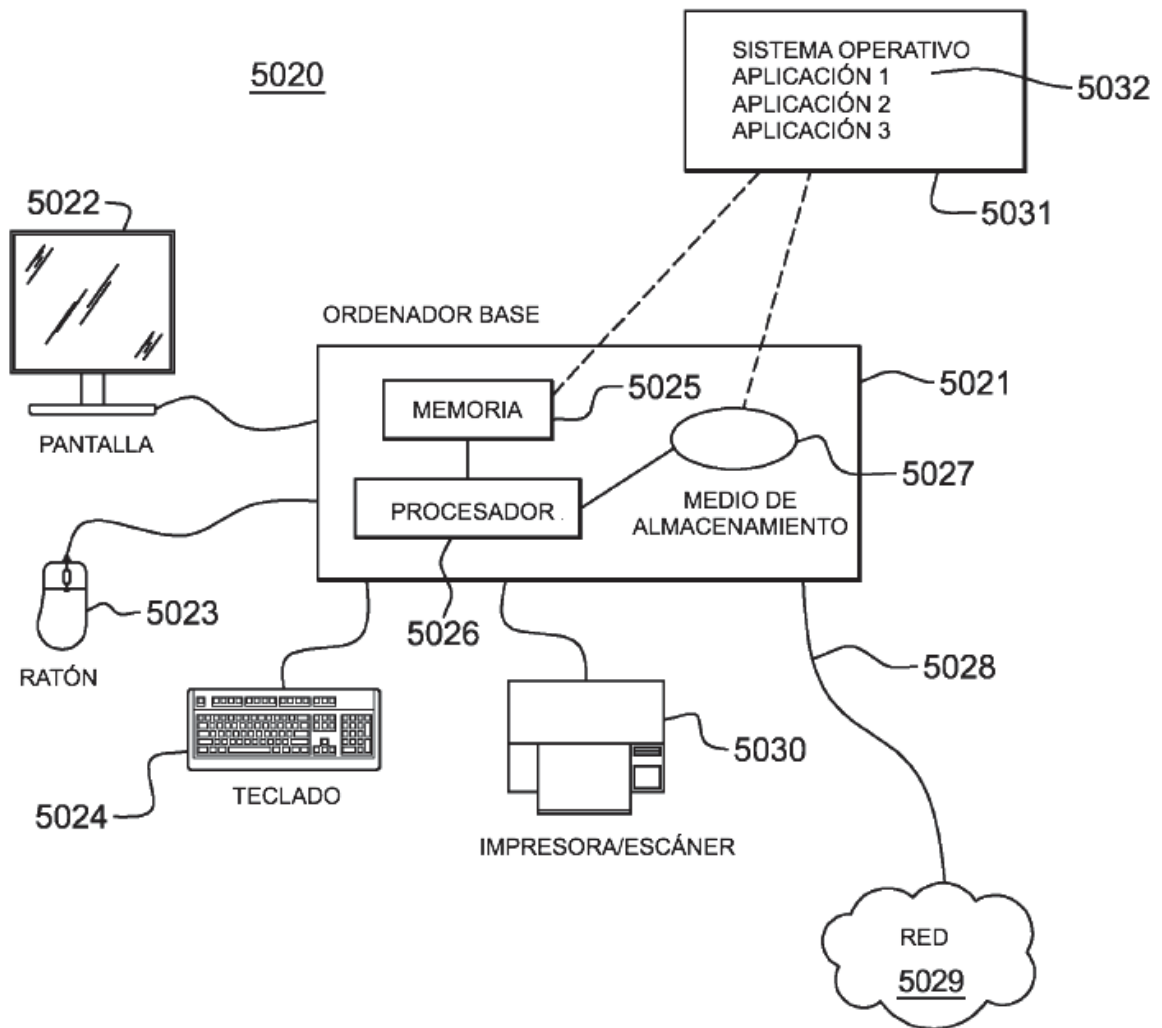


FIG. 22

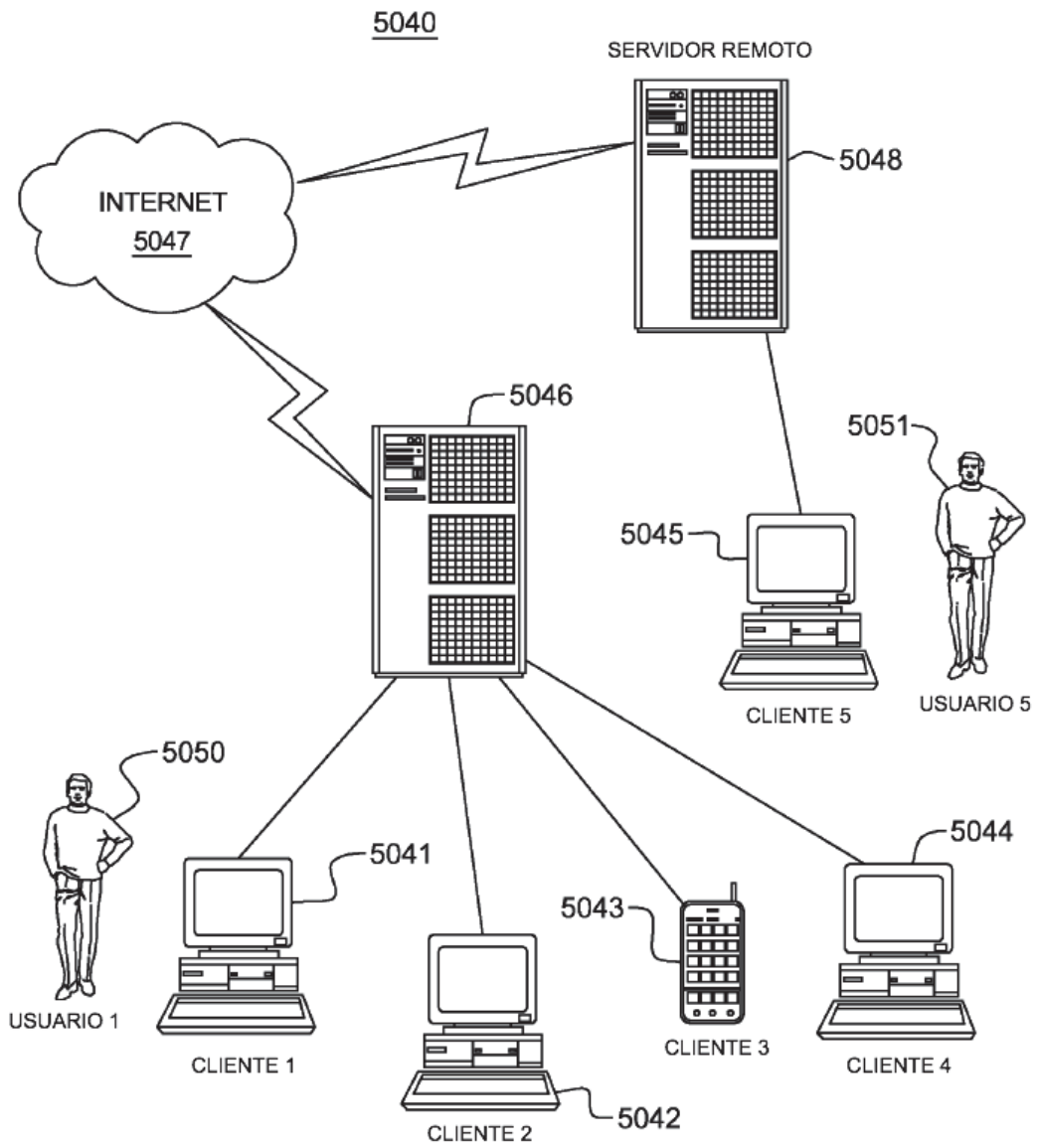


FIG. 23

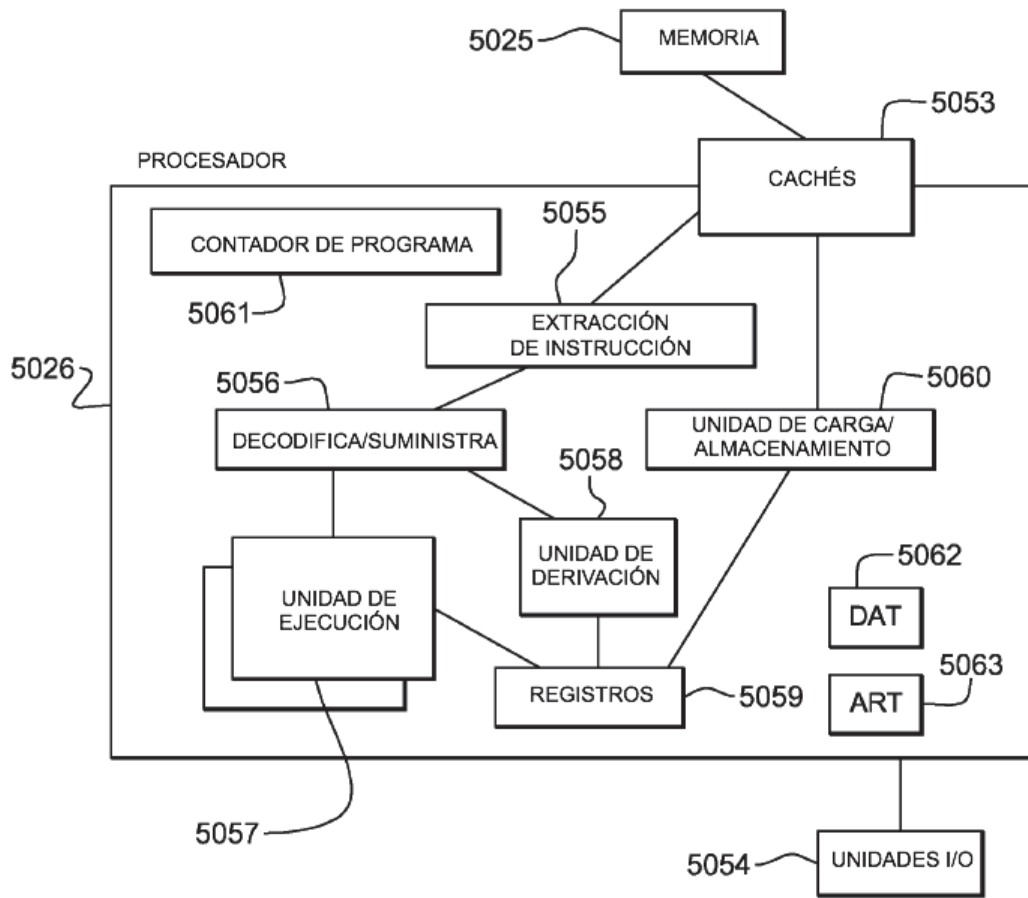


FIG. 24

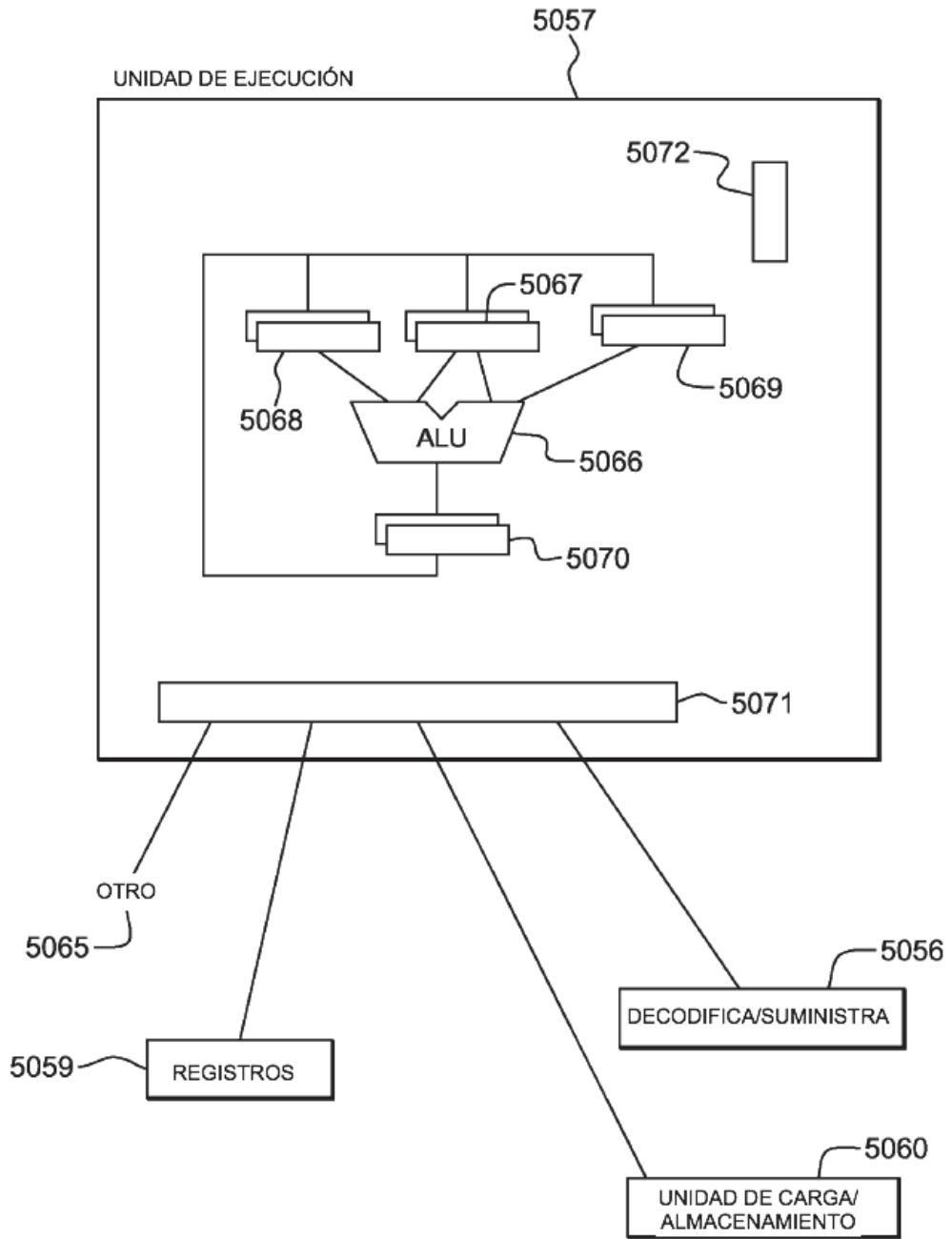


FIG. 25A

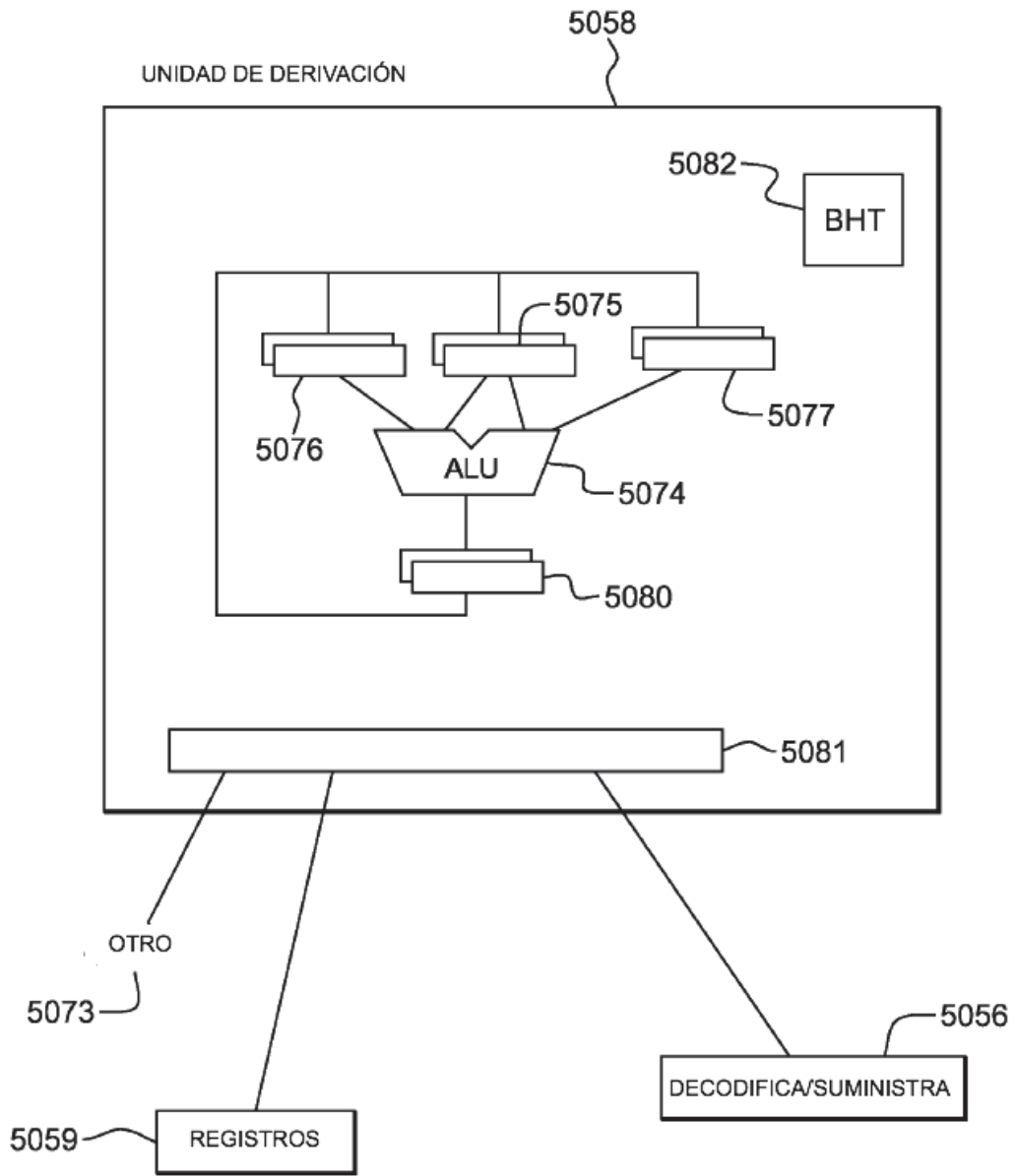


FIG. 25B

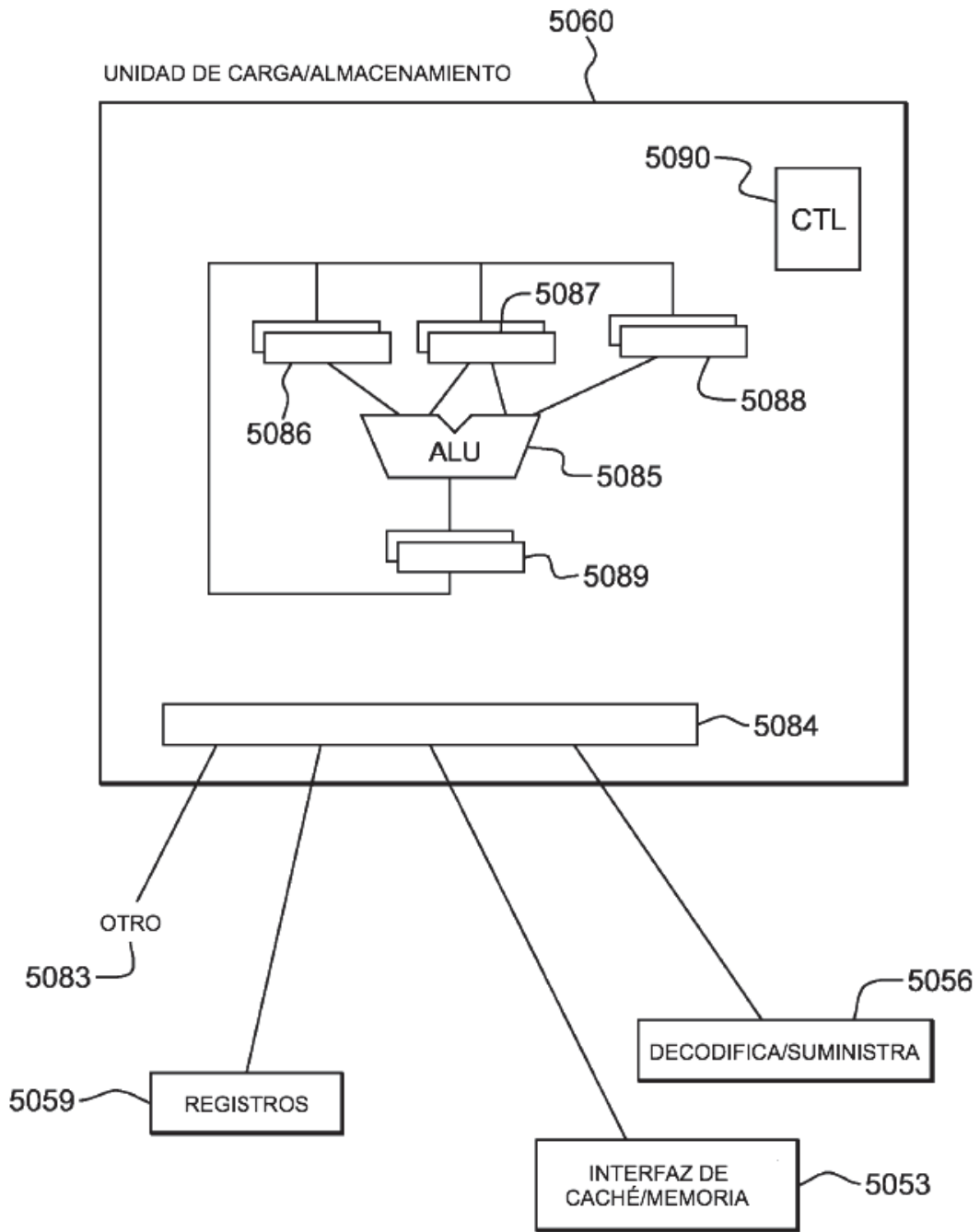


FIG. 25C

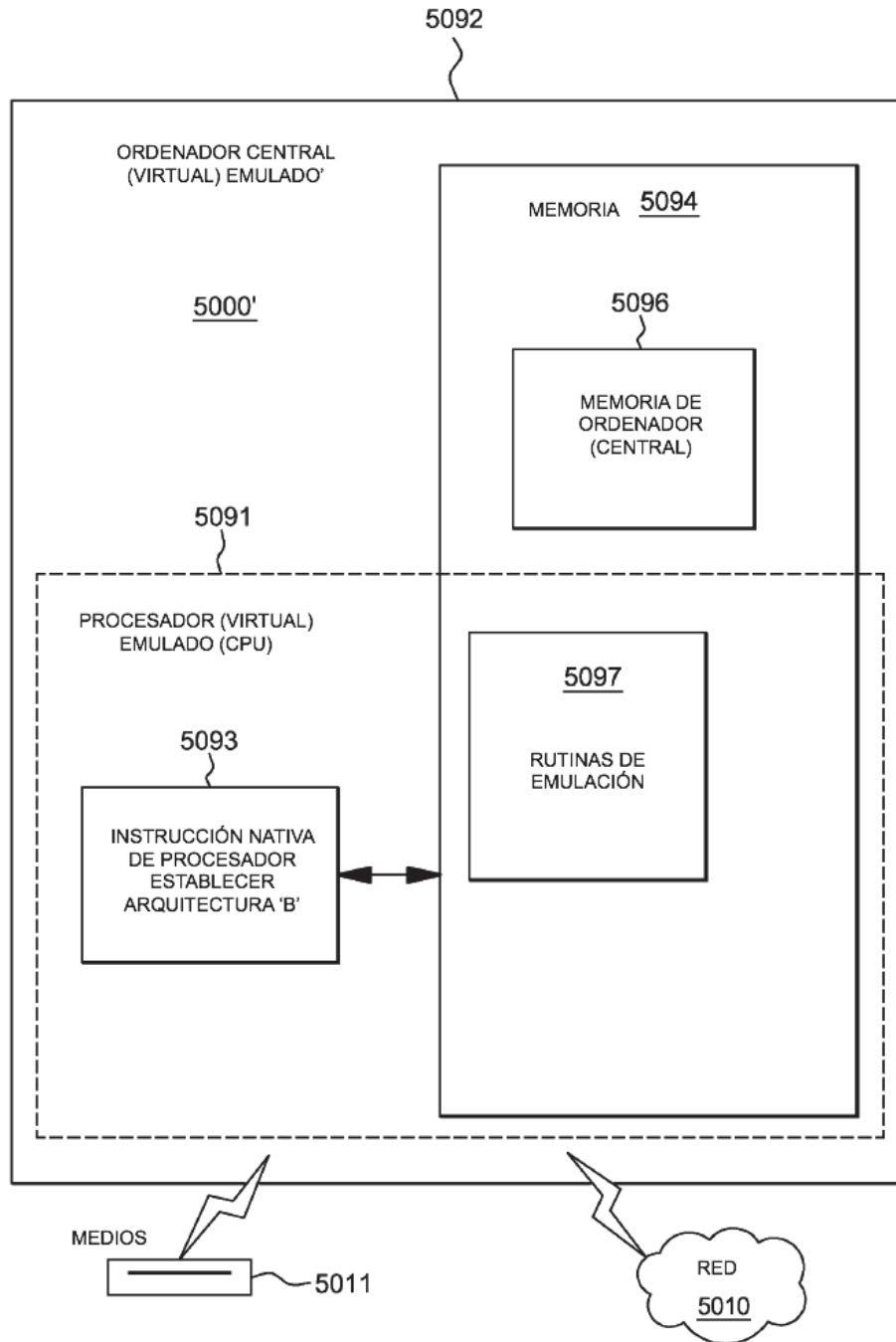


FIG. 26



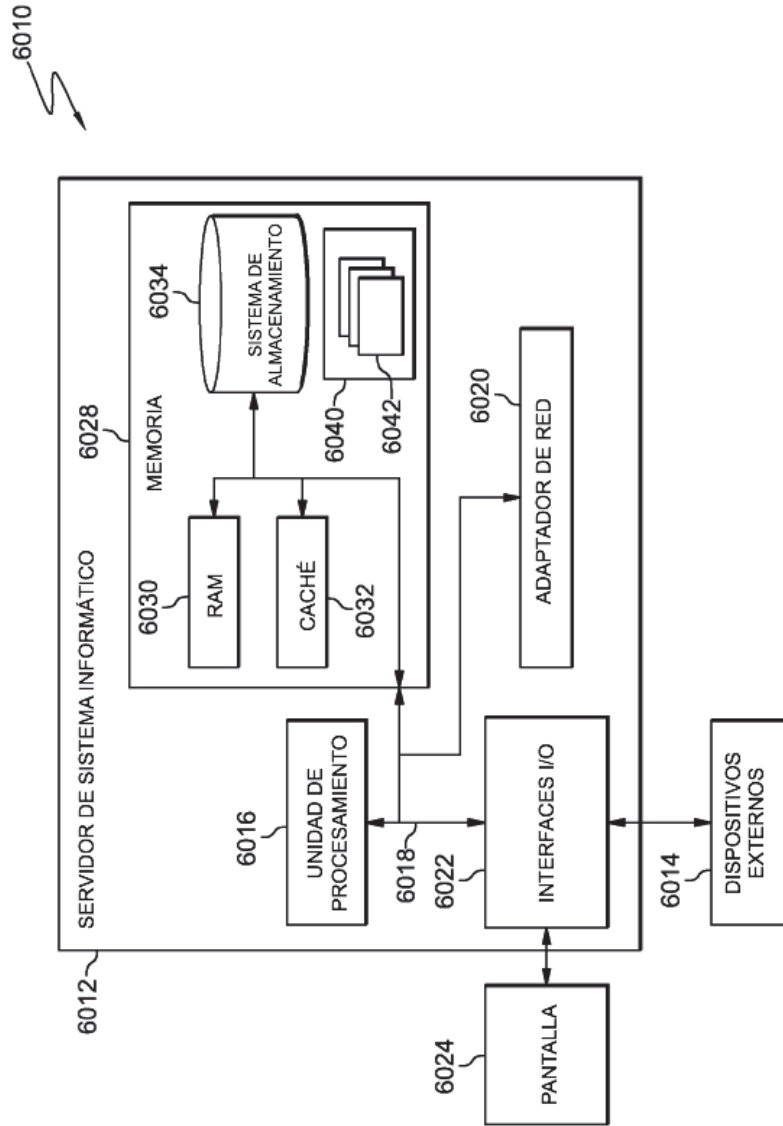


FIG. 27

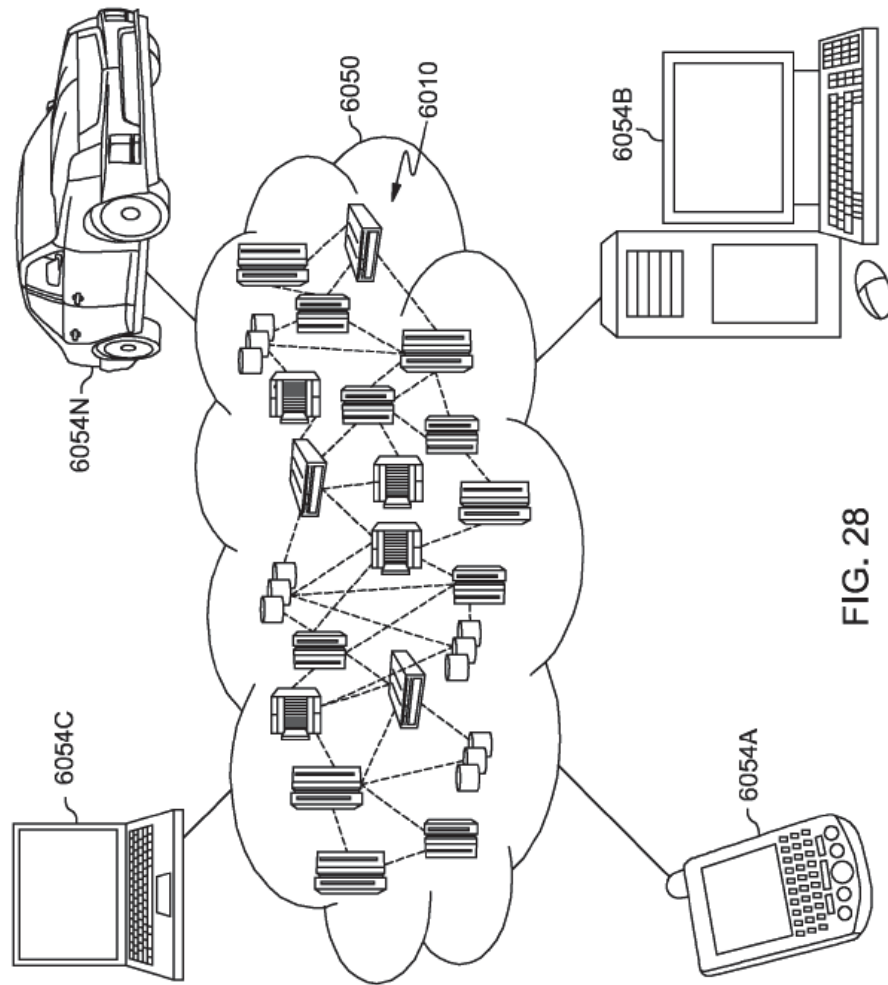


FIG. 28

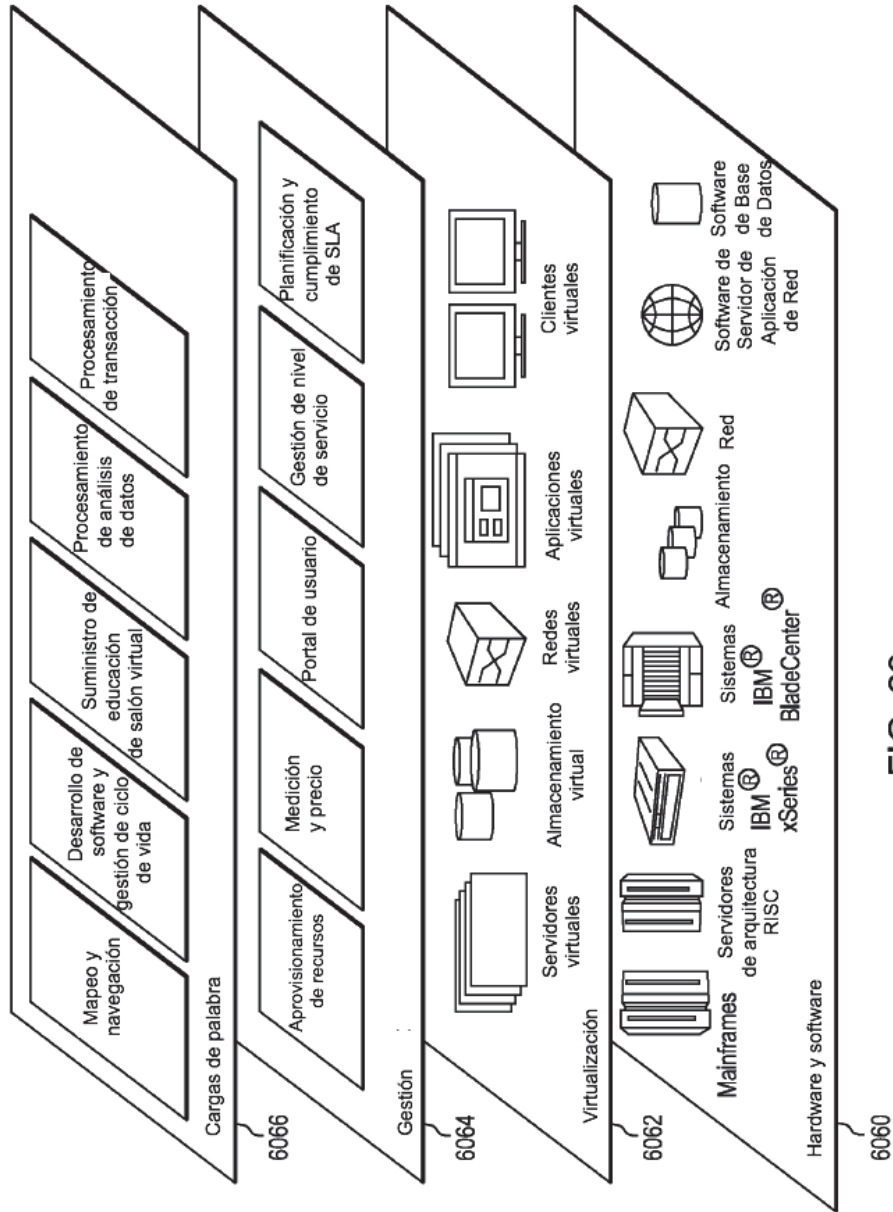


FIG. 29