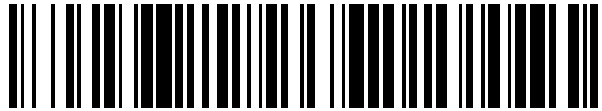


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 745 709**

51 Int. Cl.:

G06F 11/34 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **28.02.2013 PCT/JP2013/001224**

87 Fecha y número de publicación internacional: **19.09.2013 WO13136700**

96 Fecha de presentación y número de la solicitud europea: **28.02.2013 E 13761271 (9)**

97 Fecha y número de publicación de la concesión europea: **21.08.2019 EP 2825963**

54 Título: **Notificación de instrumentación en tiempo de ejecución**

30 Prioridad:

16.03.2012 US 201213422552

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

03.03.2020

73 Titular/es:

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

72 Inventor/es:

**FARRELL, MARK S.;
GAINEY JR., CHARLES W.;
MITRAN, MARCEL M.;
SHUM, CHUNG-LUNG KEVIN y
SMITH, BRIAN LEONARD**

74 Agente/Representante:

ISERN JARA, Jorge

ES 2 745 709 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

Notificación de instrumentación en tiempo de ejecución

5 CAMPO TÉCNICO

La presente invención se refiere en general al procesamiento dentro de un entorno informático y, más específicamente, a la notificación de instrumentación en tiempo de ejecución.

10 ANTECEDENTES DE LA TÉCNICA

Los procesadores de ordenadores ejecutan transacciones utilizando predicciones de bifurcación cada vez más complejas y lógica de almacenamiento en caché de instrucciones. Estos procedimientos se han introducido para aumentar la capacidad de proceso de instrucciones y, por lo tanto, el rendimiento del procesamiento. La introducción de la lógica para mejorar el rendimiento hace que sea difícil predecir con certeza cómo se ejecutará una aplicación de software en particular en el procesador de ordenador. Durante el procedimiento de desarrollo de software, a menudo hay un equilibrio entre funcionalidad y rendimiento. El software se ejecuta en uno o más niveles de abstracción del hardware subyacente que está ejecutando el software. Cuando el hardware se virtualiza, se introduce una capa adicional de abstracción. Con la introducción de la lógica de mejora del rendimiento y las diversas capas de abstracción, es difícil tener una comprensión exhaustiva de lo que realmente ocurre a nivel de hardware cuando se ejecuta un programa. Sin esta información, los desarrolladores de software utilizan procedimientos más abstractos, como la duración de la ejecución, la utilización de memoria, el número de hilos de ejecución, etc., para optimizar la aplicación de software. El número de publicación de la solicitud de patente de los Estados Unidos US 2010/0088771 A1 (Heller, L.C. y col. "Virtualization of a central processing unit measurement facility", 8 de abril de 2010) desvela una herramienta de medición de la unidad central de procesamiento, virtualizada para dar soporte al uso simultáneo de la herramienta por parte de múltiples invitados que ejecutan dentro de un entorno virtual. Cada invitado del entorno tiene un control independiente sobre la deshabilitación/habilitación de la herramienta para ese invitado. El número de publicación de solicitud de patente de los Estados Unidos US 2003/0154430 A1 (Allen, W. James, y col. "On-Chip Instrumentation", 14 de agosto de 2003) desvela una unidad de control electrónico que incluye una unidad central de procesamiento, un banco de memoria no volátil, un banco de memoria volátil y una máquina de estados. La máquina de estados está en comunicación con la CPU y funciona para capturar de forma selectiva la información disponible en un bus interno de una CPU ciclo por ciclo y almacenar la información capturada en la memoria volátil, que también está acoplada a la CPU.

35 RESUMEN DE LA INVENCION

PROBLEMA TÉCNICO

40 Cuando la información específica del hardware está disponible, típicamente se proporciona a un desarrollador después del hecho y se proporciona de forma agregada, a un alto nivel y/o intercalado con la actividad de otros programas y el sistema operativo, lo que dificulta la identificación problemas que pueden estar afectando la eficacia y la exactitud de la aplicación de software.

SOLUCIÓN AL PROBLEMA

45 Las realizaciones incluyen un producto de programa de ordenador, procedimiento y sistema para la notificación de instrumentación en tiempo de ejecución. Un flujo de instrucciones es ejecutado por un procesador. El procesador captura la información de instrumentación en tiempo de ejecución del flujo de instrucciones en ejecución. Los registros de instrumentación en tiempo de ejecución se crean en base a la información de instrumentación en tiempo de ejecución capturada. Se detecta un punto de muestra de instrumentación en tiempo de ejecución del flujo de instrucciones en ejecución en el procesador. Un grupo de notificación se almacena en una memoria intermedia del programa de instrumentación en tiempo de ejecución. El almacenamiento se basa en la detección y el almacenamiento incluye: determinar una dirección actual de la memoria intermedia del programa de instrumentación en tiempo de ejecución, la determinación basada en controles de instrumentación accesibles en tiempo de ejecución mediante instrucciones; y almacenar el grupo de notificación en la memoria intermedia del programa de instrumentación en tiempo de ejecución en base a una dirección de origen y la dirección actual de la memoria intermedia del programa de instrumentación en tiempo de ejecución, el grupo de notificación que incluye los registros de instrumentación en tiempo de ejecución creados.

60 Ventajas y características adicionales se realizan a través de las técnicas de la presente invención. Otras realizaciones y aspectos de la invención se describen en detalle en el presente documento y se consideran parte de la invención reivindicada. Para una mejor comprensión de la invención con ventajas y características, consulte la descripción y los dibujos.

65 BREVE DESCRIPCIÓN DE LOS DIBUJOS

La materia objeto, que se considera como la invención, se señala en particular y se reivindica claramente en las reivindicaciones al término de la memoria descriptiva. Las características anteriores y otras, y las ventajas de la invención son evidentes a partir de la siguiente descripción detallada, considerada junto con los dibujos adjuntos, en los que:

5 la FIG. 1A es un diagrama que representa un sistema de ordenador proveedor de alojamiento de ejemplo en una realización;

10 la FIG. 1B es un diagrama que representa un sistema de ordenador proveedor de alojamiento de emulación de ejemplo en una realización;

la FIG. 1C es un diagrama que representa un sistema de ordenador de ejemplo en una realización;

15 la FIG. 2 es un diagrama que representa una red de ordenador de ejemplo en una realización;

la FIG. 3 es un diagrama que representa elementos de un sistema de ordenador en una realización;

la FIG. 4A representa elementos detallados de un sistema de ordenador en una realización;

20 la FIG. 4B representa elementos detallados de un sistema de ordenador en una realización;

la FIG. 4C representa elementos detallados de un sistema de ordenador en una realización;

25 la FIG. 5 representa un diagrama esquemático de un sistema para la instrumentación en tiempo de ejecución de un procesador según una realización;

la FIG. 6 representa una porción de un bloque de control de instrumentación en tiempo de ejecución (RICCB) que incluye controles que son configurables mediante un estado privilegiado en una realización;

30 la FIG. 7 representa una porción de un bloque de control RICCB cuando el bit semiprivilegiado (K) se establece en 1 en una realización;

la FIG. 8 representa una memoria intermedia de recopilación según una realización;

35 la FIG. 9 representa un grupo de notificación según una realización;

la FIG. 10 representa una realización de un registro de inicio en un grupo de notificación;

40 la FIG. 11 representa una realización de un registro de marca de tiempo en un grupo de notificación;

la FIG. 12 representa realizaciones de registros de emisión en un grupo de notificación;

la FIG. 13 representa realizaciones de registros de aborto en un grupo de notificación;

45 la FIG. 14 representa realizaciones de registros de llamadas en un grupo de notificación;

la FIG. 15 representa una realización de un registro de relleno en un grupo de notificación;

50 la FIG. 16 representa realizaciones de registros de instrucciones en un grupo de notificación;

la FIG. 17 representa un procedimiento para la notificación de instrumentación en tiempo de ejecución según una realización; y

55 la FIG. 18 ilustra un producto de programa de ordenador según una realización.

DESCRIPCIÓN DE LAS REALIZACIONES

60 Las realizaciones se relacionan solo con combinaciones de características reivindicadas. A continuación, cuando el término "realización" se refiere a combinaciones de características no reivindicadas, dicho término debe entenderse como que se refiere a ejemplos de la presente invención.

65 Una realización de la presente invención es una herramienta de instrumentación en tiempo de ejecución basada en hardware para tiempos de ejecución gestionados. Como se usa en el presente documento, el término "tiempo de ejecución gestionado" se refiere a un entorno que encapsula un estado y gestiona los recursos utilizados para ejecutar un programa o aplicación (p. ej., máquina virtual Java(R) o "JVM", sistema operativo, middleware, etc.) Las realizaciones de la herramienta de instrumentación en tiempo de ejecución permiten que un programa recopile

información sobre la ejecución del programa, incluidos los datos de la unidad central de procesamiento (CPU). La información recopilada permite que el programa adquiera conocimiento sobre el programa del cual se recopila la información. Las realizaciones de la herramienta de instrumentación en tiempo de ejecución incluyen una herramienta de hardware para recopilar secuencias de eventos (p. ej., bifurcaciones tomadas, valores de registro, etc.) en una memoria intermedia de recopilación. El contenido de la memoria intermedia de recopilación (o un subconjunto de la memoria intermedia de recopilación que contiene los registros más recientes) se copia en una memoria intermedia del programa en el espacio de direcciones de la aplicación (por ejemplo, el espacio de direcciones de una JVM) en un conjunto programable de eventos desencadenantes de muestra tales como, pero no limitados a: una directiva de software en forma de una instrucción insertada en el flujo de instrucciones; un intervalo de instrucciones ejecutadas ha finalizado, un tiempo transcurrido dado desde que expiró la última muestra y/o se observa un evento de hardware dado, tal como un fallo de memoria caché de datos o instrucciones.

Los datos almacenados en cada punto de muestra se almacenan como un conjunto de registros relacionados, denominados en el presente documento como un grupo de notificación. Un grupo de notificación se almacena en la memoria intermedia del programa cuya ubicación se determina a partir de los controles asociados con la instrumentación en tiempo de ejecución. Por lo tanto, los datos instrumentados almacenados como grupos de notificación en la memoria intermedia del programa están disponibles para el programa para su análisis posterior (o incluso un análisis agresivo durante la instrumentación en tiempo de ejecución).

Los compiladores dinámicos pueden aprovechar la información en tiempo de ejecución, tal como la recopilada por la herramienta de instrumentación en tiempo de ejecución basada en hardware descrita en el presente documento para realizar optimizaciones dirigidas por los comentarios en línea. Por ejemplo, un compilador dinámico puede utilizar información sobre rutas de ejecución importantes, valores perfilados y direcciones de bifurcación preferidas para realizar optimizaciones que se especializan o codifican versiones, dan instrucciones en línea, reordenan rutas de ejecución y enderezan bifurcaciones. Las realizaciones descritas en el presente documento no se limitan al uso por parte de los compiladores y se pueden usar de varias maneras, tales como, entre otras, para dirigir otras herramientas gestionadas en tiempo de ejecución. Por ejemplo, los eventos de memoria caché de datos se pueden usar para ayudar a un procedimiento recolector de basura a tomar decisiones sobre la ubicación conjunta o el distanciamiento de objetos para obtener una mejor localidad de datos o evitar problemas falsos de compartición.

La FIG. 1A representa los componentes representativos de un sistema de ordenador proveedor de alojamiento 50 en una realización. También se pueden emplear otras disposiciones de componentes en un sistema de ordenador. El sistema de ordenador proveedor de alojamiento 50 representativo comprende uno o más procesadores 1 en comunicación con el almacenamiento principal (memoria de ordenador) 2, así como interfaces de E/S para dispositivos de almacenamiento 11 y redes 10 para comunicarse con otros ordenadores o SAN y similares. El procesador 1 cumple con una arquitectura que tiene un conjunto de instrucciones diseñado y una funcionalidad diseñada. El procesador 1 puede tener una traducción de dirección dinámica (DAT) 3 para transformar las direcciones del programa (direcciones virtuales) en una dirección real en la memoria. Una DAT 3 típicamente incluye una memoria intermedia de búsqueda precalculada de las traducciones (TLB) 7 para almacenar en caché las traducciones, de modo que los accesos posteriores al bloque de la memoria de ordenador 2 no requieran el retardo de la traducción de dirección. Típicamente, se utiliza una memoria caché 9 entre la memoria de ordenador 2 y el procesador 1. Una memoria caché 9 puede ser jerárquica con una memoria caché grande disponible para más de una CPU y memorias caché más pequeñas y más rápidas (nivel inferior) entre la memoria caché grande y cada CPU. En algunas realizaciones, las memorias caché de nivel inferior se dividen para proporcionar memorias caché de bajo nivel por separado para la búsqueda de instrucciones y el acceso a datos. En una realización, una unidad de búsqueda de instrucción 4 trae una instrucción de la memoria de ordenador 2 mediante la memoria caché 9. La instrucción se decodifica en una unidad de decodificación de instrucción 6 y se envía (con otras instrucciones en algunas realizaciones) a las unidades de ejecución de instrucción 8. Típicamente, se emplean varias unidades de ejecución de instrucción 8, por ejemplo, una unidad de ejecución aritmética, una unidad de ejecución de coma flotante y una unidad de ejecución de instrucciones de bifurcación. La unidad de ejecución de instrucción 8 ejecuta la instrucción, accediendo a los operandos desde los registros especificados en la instrucción o la memoria de ordenador 2, según sea necesario. Si se va a acceder a un operando (cargado o almacenado) desde la memoria de ordenador 2, la unidad de almacenamiento de carga 5 típicamente gestiona 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.

En la FIG. 1B, representa un sistema de ordenador proveedor de alojamiento emulado 21 se proporciona que emula un sistema de ordenador proveedor de alojamiento de una arquitectura proveedora de alojamiento, tal como el sistema de ordenador proveedor de alojamiento 50 de la FIG. 1. En el sistema de ordenador proveedor de alojamiento emulado 21, un procesador proveedor de alojamiento (CPU) 1 es un procesador proveedor de alojamiento emulado (o procesador proveedor de alojamiento virtual) 29, y comprende un procesador nativo 27 que tiene una arquitectura de conjunto de instrucciones nativas diferente a la del procesador 1 del sistema de ordenador proveedor de alojamiento 50. El sistema de ordenador proveedor de alojamiento emulado 21 tiene memoria 22 accesible para el procesador nativo 27. En una realización, la memoria 22 se divide en una porción de memoria de ordenador 2 y una porción de memoria de rutinas de emulación 23. La memoria de ordenador 2 está disponible para los programas del sistema de ordenador proveedor de alojamiento emulado 21 según la arquitectura de ordenador proveedor de alojamiento. El procesador nativo 27 ejecuta instrucciones nativas de un conjunto de instrucciones diseñadas de una arquitectura

diferente a la del procesador emulado 29, las instrucciones nativas obtenidas de la memoria de rutinas de emulación 23, y puede acceder a una instrucción de proveedor de alojamiento para la ejecución desde un programa en la memoria de ordenador 2 empleando una o más instrucciones obtenidas en una secuencia y rutina de acceso/decodificación que puede decodificar las instrucciones de proveedor de alojamiento a las que se accede para determinar una rutina de ejecución de instrucción nativa para emular la función de la instrucción de proveedor de alojamiento a la que se accede. Otras herramientas que se definen para la arquitectura del sistema de ordenador proveedor de alojamiento 50 pueden ser emuladas por rutinas de herramientas diseñadas, incluidas las herramientas como registros de propósito general, registros de control, traducción de direcciones dinámicas y soporte para el subsistema de entrada/salida (E/S) y memoria caché de procesador, por ejemplo. Las rutinas de emulación también pueden aprovechar funciones disponibles en el procesador nativo 27 (tales como registros generales y traducción dinámica de direcciones virtuales) para mejorar el rendimiento de las rutinas de emulación. También se pueden proporcionar hardware especial y motores de descarga para ayudar al procesador nativo 27 a emular la función del sistema de ordenador proveedor de alojamiento 50.

En un procesador central, las instrucciones máquina diseñadas son usadas por programadores, hoy en día normalmente programadores de "C" a menudo por medio de una aplicación de compilación. Estas instrucciones almacenadas en el medio de almacenamiento pueden ejecutarse de forma nativa en un servidor IBM basado en z/Architecture, o de forma alternativa en máquinas que ejecutan otras arquitecturas. Se pueden emular en los servidores de procesador central de IBM existentes y futuros y en otras máquinas de IBM (p. ej., servidores pSeries(R) y servidores xSeries(R)). Se pueden ejecutar en máquinas que ejecutan Linux en una amplia variedad de máquinas que utilizan hardware fabricado por IBM(R), Intel(R), AMD(TM), Sun Microsystems y otros. Además de la ejecución en ese hardware bajo Z/Architecture(R), Linux se puede usar así como también máquinas que usan la emulación de Hercules, UMX, Fundamental Software, Inc. (FSI) o Platform Solutions, Inc. (PSI), donde en general la ejecución está en modo de emulación. En modo de emulación, el software de emulación es ejecutado por un procesador nativo para emular la arquitectura de un procesador emulado.

Uno o más de los componentes del sistema de ordenador proveedor de alojamiento emulado 21 se describen más detalladamente en "IBM(R) z/Architecture Principles of Operation", N.º de publicación SA22-7832-08, 9ª edición, agosto de 2010. IBM es una marca registrada de International Business Machines Corporation, Armonk, Nueva York, EE. UU. Otros nombres utilizados en el presente documento pueden ser marcas registradas, marcas comerciales o nombres de productos de International Business Machines Corporation u otras compañías.

El procesador nativo 27 típicamente ejecuta software de emulación almacenado en la memoria de rutinas de emulación 23 que comprende firmware o un sistema operativo nativo para realizar la emulación del procesador emulado. El software de emulación es responsable de ir a buscar 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 ir a buscar una o más instrucciones máquina emuladas a la vez y convertir una o más instrucciones máquina emuladas en un grupo correspondiente de instrucciones máquina nativas para su ejecución por el procesador nativo 27. Estas instrucciones convertidas pueden almacenarse en caché de manera que se pueda conseguir una conversión más rápida. El software de emulación mantiene las reglas de arquitectura de la arquitectura del procesador emulado para asegurar que los sistemas operativos y las aplicaciones escritas para el procesador emulado funcionen correctamente. Además, el software de emulación proporciona recursos identificados por la arquitectura del procesador emulado, que incluyen, pero no se limitan a, registros de control, registros de propósito general, registros de coma flotante, función de traducción de direcciones dinámicas, incluidas tablas de segmentos y tablas de páginas, por ejemplo, mecanismos de interrupción, mecanismos de conmutación de contexto, relojes horarios (TOD) e interfaces diseñadas para subsistemas de E/S de manera que un sistema operativo o un programa de aplicación diseñado para ejecutarse en el procesador emulado 29, pueda ejecutarse en el procesador nativo 27 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 29 se implementa, por ejemplo, en una subrutina "C" o controlador, o algún otro procedimiento para proporcionar un controlador para el hardware específico como estará dentro de la habilidad de los expertos en la técnica después de comprender la descripción de la realización preferida.

En una realización, la invención se puede poner en práctica por software (a veces denominado código interno con licencia, firmware, microcódigo, milicódigo, picocódigo y similares, cualquiera de los cuales sería congruente con la presente invención). En referencia a la FIG. 1A, el código de programa de software que materializa la presente invención es accedido por el procesador también conocido como CPU (Unidad Central de Procesamiento) 1 del sistema de ordenador proveedor de alojamiento 50 desde el dispositivo de almacenamiento 11 tal como un medio de almacenamiento a largo plazo, una unidad de CD-ROM, unidad de cinta o disco duro. El código de programa de software se puede materializar en cualquiera de una variedad de medios conocidos para su uso con un sistema de procesamiento de datos, como un disquete, disco duro o CD-ROM. El código puede distribuirse en dichos medios, o puede distribuirse a los usuarios desde la memoria de ordenador 2 o el almacenamiento de un sistema de ordenador a través de una red 10 a otros sistemas de ordenador para su uso por los usuarios de dichos otros sistemas.

De forma alternativa, el código de programa se puede materializar en la memoria de ordenador 2, y ser accedido por el procesador 1 utilizando un bus de procesador (no se muestra). Dicho código de programa incluye un sistema operativo que controla la función e interacción de los diversos componentes de ordenador y uno o más programas de aplicación. El código de programa normalmente se pagina desde un medio denso tal como el dispositivo de almacenamiento 11 a la memoria de ordenador 2, donde está disponible para su procesamiento por el procesador 1. Las técnicas y procedimientos para materializar 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 analizarán más en el presente documento. El código de programa, cuando se crea y se 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 de ordenador". El medio del producto de programa de ordenador es típicamente legible por un circuito de procesamiento, preferentemente en un sistema de ordenador para su ejecución por el circuito de procesamiento.

La FIG. 1C ilustra una estación de trabajo representativa o un sistema de hardware de servidor en el que se puede poner en práctica la presente invención. El sistema 100 de la FIG. 1C comprende un sistema de ordenador base representativo 101, tal como un ordenador personal, una estación de trabajo o un servidor, que incluye dispositivos periféricos opcionales. El sistema de ordenador base 101 incluye uno o más procesadores 106 y un bus (no se muestra) empleado para conectar y permitir la comunicación entre los uno o más procesadores 106 y los otros componentes del sistema de ordenador base 101 según técnicas conocidas. El bus conecta el procesador 106 a la memoria 105 y al almacenamiento a largo plazo 107 que puede incluir un disco duro (incluido cualquiera de medios magnéticos, CD, DVD y memoria flash, por ejemplo) o una unidad de cinta, por ejemplo. El sistema de ordenador base 101 también puede incluir un adaptador de interfaz de usuario, que conecta los uno o más procesadores 106 a través del bus a uno o más dispositivos de interfaz, tal como un teclado 104, un ratón 103, una impresora/escáner 110 y/u otro dispositivos de interfaz, que pueden ser cualquier dispositivo de interfaz de usuario, tal como una pantalla táctil, tableta de entrada digitalizada, etc. El bus también conecta los uno o más procesadores a un dispositivo de visualización 102, tal como una pantalla LCD o monitor a través de un adaptador de pantalla.

El sistema de ordenador base 101 puede comunicarse con otros ordenadores o redes de ordenadores por medio de un adaptador de red capaz de comunicarse 108 con una red 109. Los adaptadores de red de ejemplo son canales de comunicaciones, anillo token, Ethernet o módems. De forma alternativa, el sistema de ordenador base 101 puede comunicarse usando una interfaz inalámbrica, tal como una tarjeta digital celular de datos en paquetes (CDPD). El sistema de ordenador base 101 puede estar asociado con dichos otros ordenadores en una red de área local (LAN) o una red de área amplia (WAN), o el sistema de ordenador base 101 puede ser un cliente en una disposición cliente/servidor con otro ordenador, etc.

La FIG. 2 ilustra una red de procesamiento de datos 200 en la que se puede poner en práctica la presente invención. La red de procesamiento de datos 200 puede incluir una pluralidad de redes individuales, tales como una red inalámbrica y una red cableada, cada una de las cuales puede incluir una pluralidad de estaciones de trabajo individuales 201, 202, 203, 204 y/o el sistema de ordenador base 101 de la FIG. 1C. Además, como apreciarán los expertos en la técnica, se pueden incluir una o más LAN, donde una LAN puede comprender una pluralidad de estaciones de trabajo inteligentes acopladas a un procesador proveedor de alojamiento.

El código de programación 111 se puede materializar en la memoria 105, y puede ser accedido por el procesador 106 utilizando el bus del procesador. Dicho código de programación incluye un sistema operativo que controla la función e interacción de los diversos componentes de ordenador y uno o más programas de aplicación 112. El código de programa normalmente se pagina desde el almacenamiento a largo plazo 107 a la memoria de alta velocidad 105 donde está disponible para su procesamiento por el procesador 106. Las técnicas y procedimientos para materializar el 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 analizarán más en el presente documento. El código de programa, cuando se crea y se 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 de ordenador". El medio del producto de programa de ordenador es típicamente legible por un circuito de procesamiento, preferentemente en un sistema de ordenador para su ejecución por el circuito de procesamiento.

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

Todavía en referencia a la FIG. 2, las redes también pueden incluir ordenadores o servidores de procesador central, tales como un ordenador de puerta de enlace (servidor cliente) 206 o un servidor de aplicaciones (servidor remoto) 208 que puede acceder a un repositorio y también pueden ser accedidos directamente desde una estación de trabajo 205. Un ordenador de puerta de enlace 206 sirve como punto de entrada en cada red 207. Se necesita una puerta de enlace cuando se conecta un protocolo de red a otro. El ordenador de puerta de enlace 206 puede estar preferentemente acoplado a otra red (Internet 207 por ejemplo) por medio de un enlace de comunicaciones. El

ordenador de puerta de enlace 206 también se puede acoplar directamente a las una o más estaciones de trabajo 101, 201, 202, 203 y 204 usando un enlace de comunicaciones. El ordenador de puerta de enlace puede implementarse utilizando un servidor IBM eServer(TM) zSeries(R) z9(R) disponible en International Business Machines Corporation.

5 En una realización, el código de programación de software que materializa la presente invención es accedido por el procesador 106 del sistema de ordenador base 101 desde medios de almacenamiento a largo plazo, tales como el almacenamiento a largo plazo 107 de la FIG. 1C. El código de programación de software se puede materializar en cualquiera de una variedad de medios conocidos para su uso con un sistema de procesamiento de datos, como un disquete, disco duro o CD-ROM. El código puede distribuirse en dichos medios, o puede distribuirse a los usuarios 210 y 211 desde la memoria o almacenamiento de un sistema de ordenador a través de una red a otros sistemas de ordenador para su uso por los usuarios de dichos otros sistemas.

15 En referencia a la FIG. 3, se representa una realización de procesador ejemplar para el procesador 106. Se emplean uno o más niveles de memoria caché 303 para almacenar en memoria intermedia bloques de memoria con el fin de mejorar el rendimiento del procesador 106. La memoria caché 303 es una memoria intermedia de alta velocidad que contiene líneas de memoria caché de datos de memoria que es probable que se usen. Las líneas de memoria caché típicas son 64, 128 o 256 bytes de datos de memoria. En una realización, se emplean memorias caché separadas más 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 en las memorias caché) a menudo es proporcionada por diversos algoritmos "snoop" bien conocidos en la técnica. El almacenamiento principal, tal como la memoria 105 de un sistema procesador, a menudo se denomina memoria caché. En un sistema de procesador que tiene 4 niveles de memoria caché 303 la memoria 105 se denomina a veces memoria caché de nivel 5 (L5), ya que típicamente es más rápida y solo contiene una porción del almacenamiento no volátil (DASD, cinta, etc.) que está disponible para un sistema de ordenador. La memoria 105 "almacena en caché" páginas de datos paginadas dentro y fuera de la memoria 105 por el sistema operativo.

25 Un contador de programa (contador de instrucciones) 311 realiza un seguimiento de la dirección de la instrucción actual a ejecutar. Un contador de programa en un procesador az/Architecture es de 64 bits y puede truncarse a 31 o 30 24 bits para admitir los límites de direccionamiento anteriores. Un contador de programa típicamente se materializa en una palabra de estado de programa (PSW) de un ordenador de manera que persiste durante la conmutación de contexto. Por lo tanto, un programa en progreso, que tiene un valor de contador de programa, puede ser interrumpido, por ejemplo, por el sistema operativo (es decir, el contexto actual conmuta del entorno de programa al entorno de sistema operativo). La PSW del programa mantiene el valor del contador de programa mientras el programa no está activo, y el contador de programa (en la PSW) del sistema operativo se usa mientras el sistema operativo se está ejecutando. En una realización, el contador de programa se incrementa en una cantidad igual al número de bytes de la instrucción actual. Las instrucciones de cálculo de conjuntos de instrucciones reducidas (RISC) son típicamente de longitud fija, mientras que las instrucciones de cálculo de conjuntos de instrucciones complejas (CISC) son típicamente de longitud variable. Las instrucciones de IBM z/Architecture son instrucciones CISC que tienen una longitud de 2, 4 40 o 6 bytes. El contador de programa 311 se modifica mediante una operación de conmutación de contexto o una operación de una bifurcación tomada de una instrucción de bifurcación, por ejemplo. En una operación de conmutación de contexto, el valor del contador de programa actual se guarda en la PSW junto con otra información de estado sobre el programa que se está ejecutando (tales como los códigos de condición), y se carga un nuevo valor del contador de programa apuntando a una instrucción de un nuevo módulo de programa a ejecutar. Se realiza una operación de una 45 bifurcación tomada con el fin de permitir que el programa tome decisiones o itere dentro del programa cargando el resultado de la instrucción de bifurcación en el contador de programa 311.

50 En una realización, se emplea una unidad de búsqueda de instrucción 305 para ir a buscar instrucciones en nombre del procesador 106. La unidad de búsqueda de instrucción 305 va a buscar o bien las "instrucciones secuenciales siguientes", las instrucciones de destino de las instrucciones tomadas de la bifurcación, o bien las primeras instrucciones de un programa después de una conmutación de contexto. En una realización, la unidad de búsqueda de instrucción 305 emplea técnicas de búsqueda previa para ir a buscar especulativamente instrucciones en base a la probabilidad de que puedan usarse las instrucciones traídas previamente. Por ejemplo, la unidad de búsqueda de instrucción 305 puede obtener 16 bytes de instrucciones que incluyen la siguiente instrucción secuencial y bytes 55 adicionales de instrucciones secuenciales adicionales.

Las instrucciones traídas son ejecutadas por el procesador 106. En una realización, las instrucciones traídas se pasan a una unidad de decodificación/envío 306 de la unidad de búsqueda de instrucción 305. La unidad de decodificación/envío

60 306 decodifica la(s) instrucción(es) y reenvía información sobre la(s) instrucción(es) decodificada(s) a las unidades de ejecución adecuadas 307, 308 y/o 310. Una unidad de ejecución 307 recibe información sobre instrucciones aritméticas decodificadas de la unidad de búsqueda de instrucción 305 y realizará operaciones aritméticas en operandos según el código de operación (códigoop) de la instrucción. Los operandos se proporcionan a la unidad de ejecución 307 desde la memoria 105, registros diseñados 309 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 105, los registros

diseñados 309 o en otro hardware de la máquina (tales como registros de control, registros PSW y similares).

Un procesador 106 típicamente tiene una o más unidades de ejecución 307, 308 y 310 para ejecutar la función de la instrucción. En referencia a la FIG. 4A, una unidad de ejecución 307 puede comunicarse con los registros diseñados 309, la unidad de decodificación/envío 306, la unidad de carga/almacenamiento 310 y otras unidades de procesador 401 por medio de la lógica de interfaz 407. La unidad de ejecución 307 puede emplear varios circuitos de registro 403, 404 y 405 para contener información sobre la que operará la unidad de lógica aritmética (ALU) 402. La ALU 402 realiza operaciones aritméticas tal como sumar, restar, multiplicar y dividir, así como funciones lógicas como AND, OR y OR exclusiva (XOR), rotación y desplazamiento. En una realización, la ALU admite operaciones especializadas que dependen del diseño. Otros circuitos pueden proporcionar otras herramientas diseñadas 408 que incluyen códigos de condición y lógica de soporte de recuperación, por ejemplo. Típicamente, el resultado de una operación de ALU se mantiene en un circuito de registro de salida 406 que puede reenviar el resultado a una variedad de otras funciones de procesamiento. En otras realizaciones, hay muchas disposiciones de unidades de procesador, la presente descripción solo está destinada a proporcionar una comprensión representativa de una realización.

Una instrucción ADD, por ejemplo, se ejecutaría en una unidad de ejecución 307 con funcionalidad aritmética y lógica, mientras que una instrucción de coma flotante, por ejemplo, se ejecutaría en una unidad de ejecución de coma flotante (no se muestra) con capacidad de coma flotante especializada. Preferentemente, una unidad de ejecución opera en operandos identificados por una instrucción realizando una función definida por el código op en los operandos. Por ejemplo, una instrucción ADD puede ser ejecutada por una unidad de ejecución 307 en operandos encontrados en dos registros 309 diseñados identificados por campos de registro de la instrucción.

La unidad de ejecución 307 realiza la adición aritmética en dos operandos y almacena el resultado en un tercer operando donde el tercer operando puede ser un tercer registro o uno de los dos registros de origen. La unidad de ejecución 307 utiliza preferentemente una unidad de lógica aritmética (ALU) 402 que es capaz de realizar una variedad de funciones lógicas tales como desplazamiento, rotación y/o XOR, así como una variedad de funciones algebraicas que incluyen cualquiera de sumar, restar, multiplicar, dividir. Algunas ALU 402 están diseñadas para operaciones escalares y otras para coma flotante. En realizaciones, el orden de los bytes de los datos puede ser grande-pequeño (donde el byte menos significativo está en la dirección de byte más alta) o pequeño-grande (donde el byte menos significativo está en la dirección de byte más baja) dependiendo de la arquitectura. IBM z/Architecture es un orden de los bytes grande-pequeño. Los campos firmados pueden ser signo y magnitud, complemento de 1 o complemento de 2 dependiendo de la arquitectura. El número de complemento de 2 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 solo requiere una suma dentro de la ALU. Los números se describen comúnmente en forma abreviada, donde un campo de 12 bits define una dirección de un bloque de 4.096 bytes y se describe comúnmente como un bloque de 4 Kbytes (Kilobyte), por ejemplo.

En referencia a la FIG. 4B, la información de instrucción de bifurcación para ejecutar una instrucción de bifurcación se envía típicamente a una unidad de bifurcación 308 que emplea un algoritmo de predicción de la bifurcación tal como una tabla de historial de bifurcación 432 para predecir el resultado de la bifurcación antes de que finalicen otras operaciones condicionales. El destino de la instrucción de bifurcación actual se irá a buscar y se ejecutará especulativamente antes de que finalicen las operaciones condicionales. Cuando finalizan las operaciones condicionales, las instrucciones de bifurcación ejecutadas especulativamente finalizan o se descartan en base a las condiciones de la operación condicional y el resultado especulado. Una instrucción de bifurcación típica puede probar códigos de condición y bifurcarse a una dirección de destino si los códigos de condición cumplen con el requisito de bifurcación de la instrucción de bifurcación, una dirección de destino puede calcularse en base a varios números, incluidos los que se encuentran en los campos de registro o un campo inmediato de la instrucción por ejemplo. En una realización, la unidad de bifurcación 308 puede emplear una ALU 426 que tiene una pluralidad de circuitos de registro de entrada 427, 428 y 429 y un circuito de registro de salida 430. La unidad de bifurcación 308 puede comunicarse con registros generales, la unidad de decodificación/envío 306 u otros circuitos 425, por ejemplo.

La ejecución de un grupo de instrucciones puede interrumpirse por una variedad de razones que incluyen una conmutación de contexto iniciado por un sistema operativo, una excepción o error del programa que causa una conmutación de contexto, una señal de interrupción de E/S que causa una conmutación de contexto o actividad de múltiples hilos de ejecución de una pluralidad de programas (en un entorno de múltiples hilos de ejecución) por ejemplo. En una realización, 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 incluye un valor de contador de programa que apunta a una instrucción siguiente a ejecutar, códigos de condición, información de traducción de memoria y contenido de registro diseñado. Una actividad de conmutación de contexto puede ser ejercida por circuitos de hardware, programas de aplicación, programas de sistema operativo o código de firmware (microcódigo, picocódigo o código interno con licencia (LIC) solo o en combinación).

Un procesador accede a los operandos según los procedimientos 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 o bien a registros de propósito general o registros de propósito especial

(registros de coma flotante, por ejemplo). La instrucción puede utilizar registros implícitos identificados por un campo códigoop 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 herramienta 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. La ubicación en el presente documento implica una ubicación en la memoria principal (almacenamiento principal) a menos que se indique lo contrario.

En referencia a la FIG. 4C, un procesador accede al almacenamiento utilizando una unidad de carga/almacenamiento 310. La unidad de carga/almacenamiento 310 puede realizar una operación de carga obteniendo la dirección del operando de destino en la memoria a través de la interfaz de la memoria caché/memoria y cargando el operando en un registro diseñado 309 u otra ubicación de memoria, o puede realizar una operación de almacenamiento obteniendo la dirección del operando de destino en la memoria y almacenando los datos obtenidos de un registro diseñado 309 u otra ubicación de memoria en la ubicación del operando de destino en la memoria. La unidad de carga/almacenamiento 310 puede ser especulativa y puede acceder a la memoria en una secuencia que está desordenada en relación con el flujo de instrucciones; sin embargo, la unidad de carga/almacenamiento 310 mantiene el aspecto para los programas cuyas instrucciones se ejecutaron en orden. Una unidad de carga/almacenamiento 310 puede comunicarse con registros diseñados 309, la unidad de decodificación/envío 306, la interfaz de la memoria caché/memoria u otros elementos 455 y comprende diversos circuitos de registro, ALU 458 y lógica de control 463 para calcular las direcciones de almacenamiento y proporcionar la secuencia de canalización para mantener las operaciones en orden. Algunas operaciones pueden estar desordenadas, pero la unidad de carga/almacenamiento proporciona funcionalidad para hacer que las operaciones desordenadas el programa las vea como que se han realizado en orden, como es bien conocido en la técnica.

Preferentemente, 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 en el sentido de que son redirigidas a la ubicación de memoria física por una de una variedad de tecnologías DAT como la DAT 312 de la FIG. 3, que incluyen, pero no se limitan a, prefijar una dirección virtual con un valor de desplazamiento, traducir la dirección virtual a través de una o más tablas de traducción, las tablas de traducción que incluyen al menos una tabla de segmentos y una tabla de páginas sola o en combinación, preferentemente, la tabla de segmentos que tiene una entrada que apunta a la tabla de páginas. En z/Architecture, se proporciona una jerarquía de traducciones 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 direcciones a menudo se mejora utilizando una memoria intermedia de búsqueda precalculada de las traducciones (TLB) que comprende entradas que asignan una dirección virtual a una ubicación de memoria física asociada. Las entradas se crean cuando la DAT 312 traduce una dirección virtual utilizando las tablas de traducción. El uso posterior de la dirección virtual puede utilizar la entrada de la TLB rápida en lugar de los accesos lentos secuenciales a la tabla de traducción. El contenido de TLB puede ser gestionado por una variedad de algoritmos de sustitución, incluido el usado menos recientemente (LRU).

En el caso de que el procesador 106 es un procesador de un sistema multiprocesador, cada procesador tiene la responsabilidad de mantener recursos compartidos tales como E/S, memorias caché, TLB y memoria entrelazables por coherencia. En una realización, se utilizarán tecnologías "snoop" para mantener la coherencia de la memoria caché. En un entorno snoop, cada línea de memoria caché se puede marcar como uno cualquiera de un estado compartido, un estado exclusivo, un estado modificado, un estado no válido y similares con el fin de facilitar la compartición.

Las unidades de E/S 304 de la FIG. 3 proporcionan al procesador 106 medios para conectarse a dispositivos periféricos, incluidos cintas, discos, impresoras, pantallas y redes, por ejemplo. Las unidades de E/S 304 a menudo se presentan al programa de ordenador mediante controladores de software. En procesadores centrales como la serie z/de IBM, los adaptadores de canal y los adaptadores de sistema abierto son unidades de E/S del procesador central que proporcionan las comunicaciones entre el sistema operativo y los dispositivos periféricos.

Los datos de instrumentación son datos relacionados con las operaciones del procesador 106. En una realización, el acceso a los datos de instrumentación y otras métricas a nivel de sistema puede estar restringido o no disponible. Un procesador de ordenador funciona bajo un estado privilegiado (o estado supervisor) y un estado menos privilegiado (o estado de problema). En el estado privilegiado, un programa puede tener acceso a todos los recursos del sistema a través de operaciones privilegiadas (p. ej., acceso a todos los registros de control y al espacio de memoria de supervisor). El estado privilegiado también se denomina modo privilegiado o modo supervisor. Un sistema operativo que se ejecuta en el procesador de ordenador puede estar funcionando en el estado privilegiado. El estado con menos privilegios es un estado sin privilegios donde el acceso a los recursos del sistema es limitado. Por ejemplo, los programas de aplicación que se ejecutan en un estado con menos privilegios pueden tener acceso limitado o nulo a los registros de control y pueden acceder solo al espacio de memoria de usuario asignado al programa de aplicación por el sistema operativo. El estado con menos privilegios típicamente se asigna a los programas de aplicación ejecutados bajo el control de un sistema operativo, y no se pueden realizar operaciones con privilegios en el estado con menos privilegios. El estado con menos privilegios también se conoce como estado de problema, modo de

problema o modo de usuario.

Uno de dichos recursos restringidos que no tiene acceso de escritura para un programa que se ejecuta en el estado con menos privilegios es la palabra de estado de programa (PSW). La PSW puede comprender un contador de programa de la instrucción siguiente que se ejecutará, un campo de código de condición utilizable por las instrucciones de bifurcación, un campo de control de instrumentación para indicar si la instrumentación está habilitada o deshabilitada y otra información utilizada para controlar la secuencia de instrucciones y determinar el estado del procesador de ordenador, incluido el estado de privilegio asignado al programa. En un entorno de procesamiento con múltiples hilos de ejecución, múltiples programas comparten, o fragmentan en el tiempo, la capacidad de procesador de ordenador disponible. Cada uno de los programas tiene información de contexto que incluye una PSW asociada, una dirección de origen de una tabla de traducción de direcciones para acceder al almacenamiento principal asignado al programa, un conjunto de valores actuales de registro de propósito general, registros de control, registros de coma flotante, etc. La PSW actualmente activa, o de control, se llama la PSW actual. Rige el programa que se está ejecutando actualmente. El procesador de ordenador tiene una capacidad de interrupción, lo que permite que el procesador de ordenador conmute el contexto rápidamente a otro programa en respuesta a condiciones de excepción y estímulos externos. Cuando se produce una interrupción, el procesador de ordenador coloca la PSW actual en una ubicación de almacenamiento asignada, llamada ubicación de PSW anterior, para la clase de interrupción en particular. El procesador de ordenador va a buscar una nueva PSW desde una segunda ubicación de almacenamiento asignada. Este nuevo contexto determina el siguiente programa que se ejecutará. En una realización, estas ubicaciones de almacenamiento están ubicadas en una ubicación de memoria accesible para el procesador de ordenador. Cuando el procesador de ordenador ha terminado de procesar la interrupción, el programa que gestiona la interrupción puede volver a cargar el contexto anterior, incluida la PSW anterior, convirtiéndola nuevamente en la PSW actual, de modo que el programa interrumpido pueda continuar.

Los campos de la PSW pueden ser referenciados explícitamente (p. ej., cuando la ejecución de la instrucción lee parte de los bits de la PSW) o implícitamente (p. ej., en la búsqueda de instrucciones, búsqueda de operandos, cálculos de generación de direcciones, origen de generación de direcciones, etc.). La referencia explícita se realiza en general en el tiempo de ejecución, mientras que la referencia implícita se realiza en general en diferentes etapas de la canalización durante la ejecución de la instrucción (es decir, búsqueda de instrucción, decodificación de instrucción, tiempo de ejecución y tiempo de finalización). Los campos individuales en la PSW pueden ser referenciados o actualizados de forma independiente uno del otro.

En una realización, al manipular el contexto, un sistema operativo controla los recursos de procesamiento de ordenador, incluida la habilitación de instrumentación en tiempo de ejecución por parte del procesador de ordenador. La instrumentación en tiempo de ejecución puede habilitarse o deshabilitarse durante la ejecución del sistema operativo, así como cualquier aplicación de software ejecutada por el sistema operativo. El estado habilitado/deshabilitado de la instrumentación en tiempo de ejecución se guarda como información de contexto en la PSW asociada con un programa.

Se puede incorporar una herramienta de instrumentación en tiempo de ejecución (RI) en los modelos que implementan z/Architecture. Cuando la herramienta RI está instalada y habilitada, los datos se recopilan durante la ejecución del programa en una o más memorias intermedias de recopilación dentro de la CPU y luego se notifican a una memoria intermedia del programa. Cada unidad de información almacenada se denomina grupo de notificación. El contenido de un grupo de notificación consiste en múltiples registros cuyo contenido representan eventos reconocidos por la CPU durante la ejecución del programa.

Cuando la herramienta de instrumentación en tiempo de ejecución se instala en una configuración, un campo PSW (bit RI) habilita la instrumentación en tiempo de ejecución. La validez de los controles de instrumentación en tiempo de ejecución determina la capacidad de activar el bit RI, pero cuando RI es uno, los controles de la CPU son válidos y la instrumentación en tiempo de ejecución está habilitada. La herramienta de instrumentación en tiempo de ejecución puede incluir las siguientes instrucciones: cargar controles de instrumentación en tiempo de ejecución, modificar controles de instrumentación en tiempo de ejecución, emitir instrumentación en tiempo de ejecución, siguiente instrumentación en tiempo de ejecución, desactivar instrumentación en tiempo de ejecución, activar instrumentación en tiempo de ejecución, almacenar controles de instrumentación en tiempo de ejecución y probar controles de instrumentación en tiempo de ejecución.

La instrucción de cargar controles de instrumentación en tiempo de ejecución (LRIC) inicializa los controles de instrumentación en tiempo de ejecución que rigen la instrumentación en tiempo de ejecución. La instrucción de modificar controles de instrumentación en tiempo de ejecución (MRIC) modifica todos o un subconjunto de los controles de instrumentación en tiempo de ejecución establecidos inicialmente por la LRIC. La instrucción de emitir instrumentación en tiempo de ejecución (RIEMIT) recopila el valor de un registro general almacenándolo en una memoria intermedia de recopilación. La instrucción de siguiente instrumentación en tiempo de ejecución (RINEXT) realiza un muestreo dirigido de la siguiente instrucción secuencial (NSI) después de RINEXT. La instrucción de desactivar instrumentación en tiempo de ejecución (RIOFF) deshabilita la instrumentación en tiempo de ejecución. La instrucción de activar instrumentación en tiempo de ejecución (RIOFF) habilita la instrumentación en tiempo de ejecución. La instrucción de almacenar controles de instrumentación en tiempo de ejecución (STRIC) coloca los

valores actuales de los controles de instrumentación en tiempo de ejecución en una ubicación de almacenamiento especificada. La instrucción de probar controles de instrumentación en tiempo de ejecución (TRIC) examina los controles de instrumentación en tiempo de ejecución. Si es válido, se establece el estado de un indicador con controles alterados.

5 La herramienta de instrumentación en tiempo de ejecución incluye la capacidad de hacer una alerta por medición externa pendiente de interrupción. Parte de la información recopilada por la instrumentación en tiempo de ejecución y notificada a una memoria intermedia del programa depende del modelo y, por lo tanto, no está definida. Las muestras y los datos proporcionados por la herramienta de instrumentación en tiempo de ejecución están destinados a la estimación estadística de las características de rendimiento, son sustancialmente exactos y pueden no ser repetibles. Por ejemplo, independientemente del modo de muestreo, es impredecible si una instrucción de muestra que ha provocado una excepción o está asociada con ciertas actividades internas del sistema dará lugar al almacenamiento de un grupo de notificación y, si se almacena, si los datos dependientes del modelo incluidos en la instrumentación en tiempo de ejecución están afectados.

15 Una memoria intermedia de recopilación se utiliza para capturar un conjunto de registros cuyo contenido notifica sobre eventos reconocidos por el procesador durante la ejecución del programa. Los ejemplos son: ejecución de una o más bifurcaciones tomadas, eventos de aborto de ejecución transaccional, errores de memoria caché en la búsqueda de instrucciones, errores de memoria caché o búsqueda de datos y un operando de la instrucción RIEMIT. La ejecución de la instrucción RIEMIT recopila el valor de un registro general almacenándolo en la memoria intermedia de recopilación. Se pueden recopilar y/o almacenar datos adicionales en otras memorias intermedias, tal como una memoria intermedia de datos de instrucciones.

25 La notificación está sujeta a controles de notificación. Cuando se identifica una instrucción de muestra, cada control de notificación permite la verificación de una condición correspondiente. Si existe una condición correspondiente, se forma y almacena un grupo de notificación. Un grupo de notificación no se almacena cuando no se habilita el control de notificación o la condición correspondiente no existe para un control de notificación habilitado. Los datos notificados sobre una instrucción de muestra pueden adquirirse a partir de la memoria intermedia de datos de instrucciones y otras fuentes dependientes del modelo, y luego usarse para crear el contenido de uno o más registros del grupo de notificación, uno de dichos registros que es un registro de instrucciones.

35 Los tipos de registros que pueden capturarse en el almacenamiento del grupo de notificación incluyen: relleno, extra, inicio, marca de tiempo, instrucción, emisión, aborto de TX, llamada, retorno y transferencia. Se utiliza un registro de relleno en un grupo de notificación cuando el número de registros válidos en la memoria intermedia de recopilación no es suficiente para llenar un grupo de notificación del tamaño actual del grupo de notificación. Se puede usar un registro adicional en la sección extra de un grupo de notificación. Un registro de inicio es el primer registro del primer grupo de notificación. Un registro de marca de tiempo se almacena como registro 0 de cada grupo de notificación que no sea el primer grupo de notificación. Se crea un registro de instrucción cuando se almacena un grupo de notificación para una instrucción de muestra como el último registro del grupo de notificación. Se crea un registro de emisión mediante la ejecución satisfactoria de RIEMIT. Se crea un registro de aborto del modo de ejecución de la transacción (TX) mediante un aborto implícito o mediante la ejecución de una instrucción de aborto de transacción. Se crea un registro de llamadas mediante la ejecución de una instrucción de bifurcación que se clasifica como una instrucción de bifurcación de tipo llamada. Se crea un registro de retorno mediante la ejecución de una instrucción de bifurcación de tipo retorno que se clasifica como una instrucción de retorno. Se crea un registro de transferencia mediante la ejecución de una instrucción de bifurcación que cumple con ciertos criterios del código de condición.

50 La FIG. 5 representa un diagrama esquemático de un sistema para instrumentación en tiempo de ejecución de un procesador que puede implementarse en una realización. En una realización, el sistema 500 incluye una unidad central de procesamiento (CPU) tal como el procesador 106 de la FIG. 1. En una realización, el procesador 106 es un procesador único. En una realización alternativa, el procesador 106 es un núcleo de procesamiento único de un procesador multinúcleo. En una realización, el procesador 106 es capaz de operar a velocidades variables.

55 En una realización, el procesador 106 incluye además un registro 510. El registro 510 es un registro de hardware capaz de almacenar palabras de datos para su uso por el procesador 106. El registro 510 incluye uno o más enclavamientos para almacenar bits de datos que son accesibles por el procesador 106. El registro 510 puede incluir registros de propósito general y registros de control, por ejemplo. El procesador 106 incluye adicionalmente un módulo de instrumentación 506 que está en comunicación con el registro 510. El módulo de instrumentación 506 es un circuito de procesamiento que controla la instrumentación del procesador 106. El módulo de instrumentación 506 está configurado para recopilar datos de instrumentación, tales como la ruta de ejecución de una o más bifurcaciones tomadas, eventos de aborto de ejecución transaccional, diversos operandos en tiempo de ejecución, información de la marca de tiempo, etc. directamente desde el procesador 106. El módulo de instrumentación 506 recopila los datos de instrumentación del procesador 106, y almacena los datos de instrumentación en una memoria intermedia de recopilación 508. En una realización, la memoria intermedia de recopilación 508 es una memoria intermedia circular que recopila los datos recibidos del módulo de instrumentación 506 y, cuando se llena la memoria intermedia circular, sobrescribe los datos más antiguos con nuevos datos.

El procesador 106 ejecuta uno o más sistemas operativos 516 y una o más aplicaciones 518. Los uno o más sistemas operativos 516 y una o más aplicaciones 518 se almacenan en un almacenamiento 520, tal como un disco duro, CD-ROM, memoria flash, etc. y se cargan en una memoria principal 514 en un área de memoria en tiempo de ejecución 504 reservada para almacenar una o más piezas activas del sistema operativo y/o aplicación que se ejecuta actualmente, llamadas páginas, que se cargan desde el almacenamiento 520 en la memoria en tiempo de ejecución 504 según sea necesario. En una realización, cada uno de los sistemas operativos se ejecuta como una máquina virtual gestionada por un hipervisor (no se muestra) y ejecutada por el procesador 106.

En una realización, el procesador 106 carga una PSW 512 en el registro 510 a partir de los datos de PSW 512 en la memoria principal 514 para el sistema operativo o aplicación actualmente en ejecución desde la memoria principal 514 y establece una o más configuraciones de procesador en, por ejemplo, el registro 510. En una realización, la PSW en el registro 510 incluye uno o más bits para habilitar y controlar el módulo de instrumentación 506.

Las una o más aplicaciones 518 incluyen aplicaciones de software compiladas para ejecutarse en un sistema operativo específico, código interpretado que se ejecuta en un intérprete (p. ej., Java) o hilos de soporte del sistema operativo (p. ej., gestión de procedimientos, demonios, etc.). Cada uno de los uno o más sistemas operativos 516 y/o las una o más aplicaciones 518 pueden ejecutar una instrucción para desencadenar el módulo de instrumentación 506 para iniciar o detener la recopilación de datos de instrumentación.

En una realización, una de las una o más aplicaciones 518 ejecuta una instrucción que se ha determinado que es una instrucción de muestra, creando así un punto de muestra al finalizar la ejecución de la instrucción de muestra y que luego hace que el módulo de instrumentación 506 mueva el los datos recopilados de la aplicación desde la memoria intermedia de recopilación 508, a una memoria intermedia del programa 522 en la memoria principal 514 que es accesible para la aplicación. La memoria principal 514 puede ser cualquier memoria direccionable conocida en la técnica. En una realización, la memoria principal 514 puede incluir un almacenamiento de memoria intermedia de acceso rápido, a veces denominado una memoria caché. Cada CPU puede tener una memoria caché asociada. En una realización adicional, la memoria principal 514 es la memoria dinámica de acceso aleatorio (DRAM). En otra realización más, la memoria principal es un dispositivo de almacenamiento, tal como un disco duro de ordenador, o memoria flash accesible por una aplicación.

Para configurar controles de instrumentación en tiempo de ejecución, el procesador 106 admite una instrucción de cargar controles de instrumentación en tiempo de ejecución (LRIC). Además de los campos LRIC específicos descritos más adelante en el presente documento, se entenderá que se pueden definir campos adicionales para dar soporte a otras funcionalidades. La instrucción LRIC se puede usar para cargar y configurar inicialmente la instrumentación en tiempo de ejecución y es admitida por el módulo de instrumentación 506 de la FIG. 5. En una realización, el módulo de instrumentación 506, también denominado módulo de instrumentación en tiempo de ejecución 506, implementa controles de instrumentación en tiempo de ejecución y controles de notificación. Se puede almacenar un estado actual de los controles de instrumentación en tiempo de ejecución desde el registro 510 de la FIG. 5 en la memoria principal 514 utilizando la instrucción de almacenar controles en tiempo de ejecución (STRIC). La definición de diversos campos de un bloque de control que se puede cargar como un operando de la instrucción LRIC también se usa en el presente documento para referirse al estado de los valores correspondientes de los controles de instrumentación en tiempo de ejecución.

La FIG. 6 representa una porción de un bloque de control de controles de instrumentación en tiempo de ejecución (RICCB) que incluye controles que son configurables mediante un estado privilegiado en una realización; La porción de bloque de control 600 puede incluir valores adicionales distintos de los descritos en referencia a la FIG. 6. La modificación de la porción de bloque de control 600 puede realizarse mediante una instrucción LRIC.

La porción del bloque de control incluye un bit de validez 602 (bit V). El bit de validez 602 indica la validez del conjunto de controles de instrumentación en tiempo de ejecución en el procesador, tal como fueron establecidos previamente por una instrucción LRIC.

El bloque de control también incluye un bit S 604, que se utiliza para determinar si el programa de estado con menos privilegios puede ejecutar una instrucción MRIC. El bit K 606 indica si el programa de estado con menos privilegios puede ejecutarse en un estado semiprivilegiado con respecto a los controles de instrumentación en tiempo de ejecución, tal como la dirección de origen y la dirección límite de los controles de instrumentación en tiempo de ejecución. El bit H 608 determina si los controles de dirección (es decir, la dirección de origen, la dirección límite y la dirección actual) se refieren a un espacio de dirección virtual primario o un espacio de dirección virtual local. El 0 bit 610 se ignora y se trata como un 0.

Se utiliza un bit de control de notificación de muestra de estado con menos privilegios 612 (bit Ps) junto con programas de estado con menos privilegios. Cuando en el estado con menos privilegios y el bit Ps 612 en los controles de instrumentación en tiempo de ejecución es cero, los controles de notificación de los controles de instrumentación en tiempo de ejecución se ignoran cuando la instrumentación en tiempo de ejecución está habilitada y, por lo tanto, no provocan que un grupo de notificación sea almacenado. Cuando está en el estado con menos privilegios y el bit Ps 612 en los controles de instrumentación en tiempo de ejecución es uno, los controles de notificación se verifican y

usan según su función definida.

- 5 Se utiliza un bit de control de notificación de muestra de estado supervisor 614 (bit Qs) junto con programas de estado supervisor. Cuando en el estado supervisor y el bit Qs 614 en los controles de instrumentación en tiempo de ejecución es cero, los controles de notificación de los controles de instrumentación en tiempo de ejecución se ignoran cuando la instrumentación en tiempo de ejecución está habilitada y, por lo tanto, no provocan que un grupo de notificación sea almacenado. Cuando está en el estado supervisor y el bit Qs 614 en los controles de instrumentación en tiempo de ejecución es uno, los controles de notificación se verifican y utilizan según su función definida.
- 10 El bit de control de la memoria intermedia de recopilación de estado con menos privilegios 616 (bit Pc) controla las actualizaciones en la memoria intermedia de recopilación 508 de la FIG. 5. Cuando está en un estado con menos privilegios y el bit Pc 616 en los controles de instrumentación en tiempo de ejecución es cero, los controles de la memoria intermedia de recopilación de los controles de instrumentación en tiempo de ejecución se ignoran cuando la instrumentación en tiempo de ejecución está habilitada y se impiden las actualizaciones de la memoria intermedia de recopilación 508. Cuando está en el estado con menos privilegios y el bit Pc 616 en los controles de instrumentación en tiempo de ejecución es uno, los controles de la memoria intermedia de recopilación se verifican y utilizan según su función definida.
- 15 El bit de control de la memoria intermedia de recopilación de estado supervisor 618 (bit Qc) controla las actualizaciones en la memoria intermedia de recopilación 508. Cuando está en estado supervisor y el bit Qc 618 en los controles de instrumentación en tiempo de ejecución es cero, los controles de la memoria intermedia de recopilación de los controles de instrumentación en tiempo de ejecución se ignoran cuando la instrumentación en tiempo de ejecución está habilitada y se impiden las actualizaciones de la memoria intermedia de recopilación 508. Cuando está en estado supervisor y el bit Qc 618 en los controles de instrumentación en tiempo de ejecución es uno, los controles de memoria intermedia de recopilación indicados se verifican y utilizan según su función definida.
- 20 El bit G 620 es el control pendiente de una interrupción detenida por instrumentación en tiempo de ejecución, también llamada interrupción detenida. Cuando el bit G 620 es cero, no hay una interrupción detenida pendiente. Cuando el bit G 602 es uno, hay una interrupción detenida pendiente. Cuando se escribe el primer grupo de notificación en una memoria intermedia del programa 522, el bit G 620 se establece en cero. Es decir, cuando la dirección de origen de la memoria intermedia del programa de instrumentación en tiempo de ejecución (ROA) 702 es igual a una dirección actual de memoria intermedia del programa de instrumentación en tiempo de ejecución (RCA) 706 de la FIG. 7, el bit G 620 se establece en cero. Cuando se realiza un intento de almacenar otro que no sea el primer grupo de notificación en la memoria intermedia del programa 522, el bit G 620 se establece en cero si no existe la condición de instrumentación detenida en tiempo de ejecución y el grupo de notificación se almacena. Cuando se realiza un intento de almacenar otro que no sea el primer grupo de notificación en la memoria intermedia del programa 522, el bit G 620 se establece en uno si existe la condición de instrumentación detenida en tiempo de ejecución y el grupo de notificación no se almacena.
- 30 El bit U 622 es el control de habilitación para una interrupción de memoria intermedia llena y una interrupción detenida. Cuando el bit U 622 es cero, la generación de una petición de interrupción se deshabilita y, si está pendiente, permanece pendiente.
- 35 El bit L 624 es el control pendiente de una interrupción de memoria intermedia llena. Cuando el bit L 624 es cero, una interrupción de memoria intermedia llena no está pendiente. Cuando el bit L 624 es uno, una interrupción de memoria intermedia llena está pendiente.
- 40 El campo clave 626 es un número entero sin signo de 4 bits cuyo valor se utiliza como clave de protección de almacenamiento para el almacenamiento de un grupo de notificación. Se permite almacenar un grupo de notificación solo cuando la clave de almacenamiento coincide con la clave de acceso asociada con la petición de acceso al almacenamiento, y se permite una búsqueda cuando la clave de almacenamiento coincide con la clave de acceso o cuando un bit de protección de búsqueda de la clave de almacenamiento es cero. Las claves coinciden cuando los cuatro bits de control de acceso de la clave de almacenamiento son iguales a la clave de acceso, o cuando la clave de acceso es cero.
- 45 La FIG. 7 representa una porción de un bloque de control RICCB cuando MRIC puede ejecutarse en modo semiprivilegiado (es decir, el bit K es uno). El bloque de control 700 también puede ser un operando de una instrucción LRIC para la inicialización de controles de instrumentación en tiempo de ejecución. El bloque de control 700 puede incluir valores adicionales distintos de los descritos en referencia a la FIG. 7. En una realización, las secciones del operando de instrucción MRIC que no están designadas de otra manera son inaccesibles por un programa de estado con menos privilegios. Cuando se permite el modo semiprivilegiado, una dirección de origen de memoria intermedia del programa de instrumentación en tiempo de ejecución (ROA) 702 y una dirección límite de memoria intermedia del programa de instrumentación en tiempo de ejecución (RLA) 704 se establecen con la instrucción MRIC por el programa de estado con menos privilegios. La ROA 702 es la ubicación del primer byte de la memoria intermedia del programa 522 de la FIG. 5. La RLA 704 indica la ubicación del último byte de la memoria intermedia del programa 522.
- 50
- 55
- 60
- 65

En una realización, la instrucción MRIC puede actualizar una dirección actual de memoria intermedia del programa de instrumentación en tiempo de ejecución (RCA) 706. La RCA 706 es la ubicación en la memoria intermedia del programa 522 de un próximo grupo de notificación que se almacenará. La RCA 706 examina el campo de tamaño del grupo de notificación 744 (campo RGS) y afecta el número de posiciones de bit significativas utilizadas para formar la dirección de la memoria intermedia del programa 522. La RCA 706 de 64 bits es la palabra 0, las posiciones de bit 0 a 26-RGS de la palabra 1 y los ceros binarios RGS+5 anexados a la derecha. Esta es la ubicación inicial en la memoria intermedia del programa 522 de la FIG. 5 de un grupo de notificación posterior que se almacenará en la memoria intermedia del programa 522. El grupo de notificación es una unidad de información creada por el módulo de instrumentación 506, y posteriormente almacenada en la memoria intermedia del programa 522. En una realización, cuando el campo RGS 744 especificado por la RCA 706 no es igual al tamaño del grupo de notificación actual del control de instrumentación en tiempo de ejecución (es decir, la RCA 706 cambiaría el campo RGS 744) entonces la RCA 706 se establece en la ROA 702.

El campo de recuento de intervalo de muestra restante 742 (campo RSIC) puede ser actualizado por el programa con menos privilegios utilizando la instrucción MRIC. El campo RSIC 742 incluye un número entero sin signo de 64 bits que indica un recuento de intervalo de muestra restante. Cuando el valor del campo RSIC 742 en los controles de instrumentación en tiempo de ejecución es cero o igual al valor en un campo de factor de escala 740 (campo SF), y la instrumentación en tiempo de ejecución está habilitada, entonces el siguiente intervalo de muestra es un intervalo completo basado en el modo de muestreo 708 (M) y los valores del campo SF 740. Cuando el campo RSIC 742 es distinto de cero y menor que el campo SF 740 y la instrumentación en tiempo de ejecución está habilitada, el siguiente intervalo de muestra es un intervalo parcial. Cuando el campo RSIC 742 es distinto de cero y mayor que el valor del campo SF 740 y la instrumentación en tiempo de ejecución está habilitada, el siguiente intervalo de muestra es un intervalo extendido. Cuando vence un intervalo extendido, el siguiente intervalo se basa en el valor del campo SF 740. Cuando el campo RSIC 742 se establece en un valor distinto de cero, está sujeto al mismo límite máximo dependiente del modelo al que también está sujeto el campo SF 740. Cuando el valor original del campo RSIC 742 es cero, el modo de muestreo determinará si el campo RSIC 742 se establece en el valor en el campo SF 740 durante la ejecución de las instrucciones LRIC y MRIC, o si continúa mostrándose como cero hasta se habilita la instrumentación en tiempo de ejecución.

El campo SF 740 contiene un número entero sin signo de 64 bits cuyo valor es un recuento de unidades de factor de escala. La dimensión de las unidades se determina a partir del campo de modo 708 (campo M). Cuando el valor en el campo RSIC 742 es cero, el campo SF 740 proporciona un valor inicial del campo RSIC 742 que se reduce a cero, en cuyo punto la instrucción actual se reconoce como una instrucción de muestra, y el recuento de intervalos se actualiza a partir del valor del campo SF 740. Un valor válido del campo SF 740 está en el intervalo de uno a $2^{64} - 1$. Si se especifica cero, se supone un valor de uno. Sin embargo, cada modelo puede tener un valor mínimo y máximo del campo SF 740. Los valores mínimo y máximo también pueden ser diferentes en base al campo de modo 708. Si se especifica un valor inferior al mínimo, se carga el valor mínimo dependiente del modelo. Si se especifica un valor mayor que el valor máximo, se carga el valor máximo dependiente del modelo.

El campo de control de DC 736 es un número entero sin signo de 4 bits cuyo valor designa un nivel de latencia de memoria caché asociado con un fallo de memoria caché al almacenar o buscar datos. Es decir, la instrucción de muestra ha encontrado un fallo de memoria caché en el acceso a datos. A menos que lo prohíba otro control de instrumentación en tiempo de ejecución, se intenta almacenar un grupo de notificación que represente la instrucción de muestra cuyo acceso a datos ha reconocido un fallo en un nivel de latencia en la memoria caché numéricamente mayor o igual al nivel designado por el valor del campo de control DC 736. La estructura de memoria caché y el nivel de latencia de memoria caché para el acceso a datos son dependientes del modelo. Para una instrucción con operandos múltiples o largos, es dependiente del modelo lo cual, si lo hay, el acceso a operandos se utiliza para control de notificación. El comportamiento dependiente del modelo puede ignorar el valor del campo de control DC 736 y, por lo tanto, no utilizarlo como un motivo para almacenar un grupo de notificación.

El campo IC 734 es un número entero sin signo de 4 bits cuyo valor designa un nivel de latencia de memoria caché asociado con un fallo de memoria caché en la búsqueda de instrucciones. Es decir, la búsqueda de la instrucción de muestra ha encontrado un fallo de memoria caché en la búsqueda de instrucciones. Tanto para el campo IC 734 como para el campo de control DC 736, un nivel de latencia de memoria caché es una abstracción de cuán lejos está un cierto acceso a nivel de memoria caché desde el procesador de observación. El nivel de latencia depende de la combinación de la cantidad de niveles de memoria caché anidados entre el procesador y el almacenamiento principal, y de cómo estos niveles de memoria caché se comparten entre múltiples procesadores. Un nivel de latencia mayor en general corresponde a un acceso que requiere más tiempo. Los valores en el campo IC 734 y el campo de control DC 736 pueden considerarse como identificación de origen cero de un nivel de latencia de memoria caché. Por ejemplo, un valor de cero corresponde a una memoria caché L1 (es decir, la memoria caché que está más cerca del procesador). Por lo tanto, un valor de uno es la siguiente capa de memoria caché que puede conocerse como una memoria caché L2, o incluso una memoria caché L1.5 en algunas máquinas. Los valores de 2-15 designan la progresión lógica de las capas de latencia de memoria caché adicionales hasta que se alcanza la memoria principal, pero sin incluir la propia memoria principal. En general, las estructuras de memoria caché no son tan profundas como quince capas. Por lo tanto, un valor de 15 en el campo IC 734 y el campo de control DC 736 se interpreta como un caso especial, lo que significa que un fallo de memoria caché en la búsqueda de instrucciones o el acceso a datos, respectivamente e

- independientemente del nivel de latencia de memoria caché, no se reconoce con el propósito de generar el almacenamiento de un grupo de notificación. A menos que lo prohíba otro control de instrumentación en tiempo de ejecución, se intenta almacenar un grupo de notificación que represente la instrucción de muestra cuya búsqueda ha reconocido un fallo en un nivel de latencia de memoria caché numéricamente mayor o igual al nivel designado por el valor del campo IC 734. La estructura de memoria caché y el nivel de latencia de memoria caché para la búsqueda de instrucciones son dependientes del modelo. El comportamiento dependiente del modelo puede ignorar el valor del campo IC 734 y, por lo tanto, no utilizarlo como un motivo para almacenar un grupo de notificación.
- El bit de control de notificación de anulación del nivel de latencia de memoria caché 732 (bit F) es para instrucciones sin bifurcación y para controles de predicción de la bifurcación. Cuando el bit F 732 en los controles de instrumentación en tiempo de ejecución es cero, los controles de notificación de memoria caché (campo IC 734 y campo de control DC 736) de los controles de instrumentación en tiempo de ejecución se verifican y se utilizan según su función definida. Los controles de predicción de la bifurcación (bits BPxn 722, BPxt 724, BPti 726, y BPni 728) de los controles de instrumentación en tiempo de ejecución se verifican y se utilizan según su función definida. Cuando el bit F 732 es uno, estos mismos controles se ignoran y se almacena un grupo de notificación a menos que lo prohíba otro control.
- El bit de control de fallo en la memoria caché de datos 730 (bit D) indica si se va a almacenar un grupo de notificación. Si el bit D 730 es uno, se puede colocar, o no, un registro de tipo extra en la sección extra del grupo de notificación que contiene datos dependientes del modelo sobre la instrucción de muestra.
- La instrucción MRIC incluye controles de notificación de predicción de la bifurcación (BP) (BPxn 722, BPxt 724, BPti 726 y BPni 728). Si un bit de control de notificación BP en los controles de instrumentación en tiempo de ejecución es cero, no se verifica la condición correspondiente. Si un bit de control de notificación BP es uno y existe la condición de predicción de la bifurcación correspondiente, y se almacena un grupo de notificación.
- El bit BPxn 722, cuando es uno, habilita la verificación de la información de predicción de la bifurcación. Por lo tanto, si se predice que la bifurcación de muestra se tomará incorrectamente pero no se toma, se almacena un grupo de notificación.
- El bit BPxt 724, cuando es uno, habilita la verificación de la información de predicción de la bifurcación. Por lo tanto, si se predice que la bifurcación de muestra no se tomará incorrectamente pero se toma, se almacena un grupo de notificación.
- El bit BPti 726, cuando es uno, habilita la verificación de la información de predicción de la bifurcación. Por lo tanto, si se predice que la bifurcación de muestra se tomará correctamente, y se toma, pero el destino de la bifurcación se predice incorrectamente, se almacena un grupo de notificación.
- El bit BPni 728, cuando es uno, habilita la verificación de la información de predicción de la bifurcación. Por lo tanto, si se predice correctamente que la bifurcación de muestra no se tomará, y no se toma, y el destino de la bifurcación se predice incorrectamente, se almacena un grupo de notificación.
- El control de habilitación del bit de registros del modo de ejecución transaccional 720 (bit X) controla la recopilación de registros de aborto del modo de ejecución transaccional. Cuando el bit X 720 en los controles de instrumentación en tiempo de ejecución es cero, no se recopilan los registros de aborto del modo de ejecución transaccional. Cuando el bit X 720 es uno, los registros de aborto del modo de ejecución transaccional se recopilan y se colocan en la memoria intermedia de recopilación 508 de la FIG. 5. Si un modelo no tiene una herramienta de ejecución transaccional instalada, se ignora el bit X 720.
- El bit de control de instrucción RIEMIT 718 (bit E) controla la ejecución de la instrucción RIEMIT. Cuando el bit E 718 en los controles de instrumentación en tiempo de ejecución es cero o se ignora y se trata como cero cuando la instrumentación en tiempo de ejecución está habilitada, RIEMIT ejecuta una no operación. Cuando el bit E 718 es uno, y no se ignora, RIEMIT está habilitado para ejecutar su función definida.
- El bit J 746 cuando es cero, especifica que la instrucción de condición de bifurcación (BC) está en la categoría de bifurcación de otro tipo, de forma independientemente del valor de la máscara. Si el bit J 746 es uno, la instrucción BC que especifica una máscara de 15 está en la categoría de bifurcación de tipo retorno. Cuando la instrucción BC especifica una máscara de 1-14, no se ve afectada por el bit J 746 y siempre está en la categoría de bifurcación de otro tipo. Cuando está en la categoría de bifurcación de tipo retorno, el bit R 716 controla la inclusión en la memoria intermedia de recopilación 508 de la FIG. 5. Cuando está en la categoría de bifurcación de otro tipo, el bit B 748 controla la inclusión en la memoria intermedia de recopilación 508. La categoría de bifurcación de otro tipo también puede indicarse como la categoría de bifurcación de tipo transferencia.
- El código de dirección de instrucción bit 714 (bit C) controla la habilitación de las bifurcaciones de tipo llamada. Si el bit C 714 en los controles de instrumentación en tiempo de ejecución es uno y la instrucción es una bifurcación de tipo llamada, la memoria intermedia de recopilación 508 se actualiza. Si se combina la detección dependiente del modelo de las bifurcaciones de tipo llamada y de tipo retorno, el bit C 714 opera en ambos tipos y el bit R 716 no es efectivo.

El bit R 716 es el control de habilitación de bifurcaciones de tipo retorno. Si el bit R 716 en los controles de instrumentación en tiempo de ejecución es uno y la instrucción es una bifurcación de tipo retorno, entonces la memoria intermedia de recopilación 508 se actualiza.

5 El bit B 748 es el control de habilitación de bifurcaciones distintas a las de tipo llamada y de tipo retorno. Si el bit B 748 en los controles de instrumentación en tiempo de ejecución es uno y la instrucción es una bifurcación de otro tipo reconocida por la instrumentación en tiempo de ejecución, entonces la memoria intermedia de recopilación 508 se actualiza.

10 El bit de dirección máxima excedida 712 (bit MAE), si se establece en 1, indica que se han almacenado uno o más grupos de notificación que tienen el bit C 714 establecido en uno. Una vez que el bit 712 de MAE se establece en uno, la ejecución continua de instrumentación en tiempo de ejecución no lo vuelve a establecer en cero. La ejecución de la instrucción LRIC o la instrucción MRIC que especifica el bit 712 de MAE como cero establecerá el bit 712 de MAE en cero.

15 El bit de control de la siguiente instrumentación en tiempo de ejecución (RINEXT) 710 (bit N) controla la habilitación de la siguiente instrucción de instrumentación en tiempo de ejecución, que controla la ejecución de una instrucción de muestra. Cuando el bit N 710 en los controles de instrumentación en tiempo de ejecución es cero o se ignora y se trata como cero, RINEXT ejecuta una no operación. Cuando el bit N 710 es uno, y en cualquier caso no se ignora, RINEXT está habilitado para ejecutar su función definida.

20 El campo de modo 708 (campo M) es un número entero sin signo de 4 bits cuyo valor en los controles de instrumentación en tiempo de ejecución especifica el modo de muestreo para los controles de instrumentación en tiempo de ejecución. Los modos de muestreo admitidos pueden incluir un muestreo basado en el recuento de ciclos de CPU, instrucciones de recuento o ser dirigido a una muestra en respuesta a una instrucción de muestra, tal como RINEXT.

25 El campo de tamaño del grupo de notificación 744 (RGS) es un número entero sin signo de 3 bits cuyo valor especifica el número de registros de un grupo de notificación (R_{RG}). El número de registros en un grupo de notificación puede variar desde dos registros, incluido un registro de inicio/marca de tiempo y un último registro de instrucciones, hasta doscientos cincuenta y seis registros. En una realización, el límite superior puede ser dependiente del modelo. El número de registros de 16 bytes colocados en un grupo de notificación es $2(RGS+1)$.

30 El bit de control de supresión de la capacidad de CPU primaria 738 (bit Y) y el bit de control de supresión de la capacidad de CPU secundaria 739 (bit Z) se denominan colectivamente control de supresión. La supresión del almacenamiento de un grupo de notificación significa que no se realiza un intento de almacenamiento. El control de supresión no es efectivo y no se produce supresión cuando la capacidad de la CPU de todas las CPU en la configuración es la misma. En una configuración, si la capacidad de la CPU de una CPU difiere de la capacidad de otra CPU, el control de supresión está en vigor y se dice que al menos una CPU está funcionando en la capacidad de la CPU o en la capacidad de la CPU primaria, mientras que al menos otra CPU se dice que está funcionando en la capacidad de CPU secundaria. Las capacidades de la CPU primaria y secundaria son diferentes velocidades de operación. Cuando el bit Y 738 y el bit Z 739 son ambos cero, no se produce la supresión. Cuando el bit Y 738 es cero y el bit Z 739 es uno, la supresión se produce si la CPU, p. ej., el procesador 106, está funcionando a la capacidad de la CPU secundaria. Cuando el bit Y 738 es uno y el bit Z 739 es cero, se produce la supresión si la CPU, p. ej., el procesador 106, está funcionando a la capacidad de la CPU primaria. Cuando el bit Y 738 y el bit Z 739 son uno, se produce la supresión.

35 Los campos y bits anteriores de la FIG. 7 son un ejemplo de la colocación y denominación de los campos y se proporcionan en el presente documento con fines de claridad. Se entenderá que en otras realizaciones se puede usar solo un subconjunto de los campos, los campos pueden estar en cualquier orden o posición, y/o pueden estar representados por diferentes nombres.

40 Cuando la instrumentación en tiempo de ejecución está instalada y habilitada, se pueden capturar varios eventos y datos en la memoria intermedia de recopilación 508. La memoria intermedia de recopilación 508 se utiliza para capturar un conjunto de registros cuyo contenido notifica sobre eventos reconocidos por el procesador 106 durante la ejecución del programa. Los ejemplos son: ejecución de una o más bifurcaciones tomadas, eventos de aborto de ejecución transaccional, fallos de memoria caché y un operando de una instrucción de emisión de instrumentación en tiempo de ejecución. Los campos de control IC y DC 734 y 736 establecen un nivel en el cual el programa estaría interesado en tomar alguna acción correctiva para mejorar la instrucción o el comportamiento de búsqueda previa de los datos. La ejecución de la instrucción RIEMIT recopila el valor de un registro general almacenándolo en la memoria intermedia de recopilación 508. Se pueden recopilar y/o almacenar datos adicionales en otras memorias intermedias, tal como una memoria intermedia de datos de instrucciones (IDB) (no se representa) utilizada para recopilar datos de instrucciones de muestra dependientes del modelo para construir un registro de instrucciones de instrumentación en tiempo de ejecución.

La información recopilada de la instrumentación en tiempo de ejecución se notifica sobre una base de muestreo. Las instrucciones del flujo de instrucciones se muestrean. La instrucción que se muestrea se llama instrucción de muestra. Un número de modos para determinar una instrucción de muestra se definen de la siguiente manera cuando la instrumentación en tiempo de ejecución está habilitada. En el modo de recuento de ciclos, un recuento es el número de ciclos de CPU especificados en SF 740 o RSIC 742, sea cual sea el que se use para proporcionar el recuento para el intervalo actual. El recuento se ajusta en respuesta a un evento asociado con el modo de muestreo. Por ejemplo, el recuento se puede disminuir cuando el procesador 106 está en el estado operativo. Cuando el recuento se reduce al valor umbral, tal como cero, la instrucción actual se reconoce como una instrucción de muestra, y el recuento se reinicializa al valor SF 740 y comienza a disminuir con el siguiente ciclo. Cuando finaliza la ejecución de la instrucción de muestra, se realiza una notificación, si corresponde.

En el modo de recuento de instrucciones, se especifica un recuento en SF 740 o RSIC 742, sea cual sea el que se use para proporcionar el recuento para el intervalo actual. Para una instrucción que consiste en una única unidad de operación, el recuento se reduce en la finalización de la instrucción como un evento utilizado para ajustar el recuento. La instrucción es una instrucción de muestra cuando el recuento se reduce a un valor umbral, tal como cero. Para una instrucción que consta de múltiples unidades de operación, el recuento se puede disminuir de una de las siguientes maneras: a. Para una instrucción interrumpible, todas las unidades de operación a través de la finalización parcial representan una unidad contada para la cual se disminuye la cuenta, b. Para una instrucción interrumpible, todas las unidades de operación desde la finalización parcial más reciente hasta la finalización final representan una unidad contada para la cual se disminuye el recuento, c. Para una instrucción que finaliza después de realizar una subporción determinada por la CPU del procesamiento especificado por los parámetros de la instrucción, la finalización representa una unidad contada para la cual se disminuye el recuento, d. Para una instrucción que finaliza después de realizar múltiples unidades de operación pero no en las categorías a-c anteriores, la finalización de la última unidad de operación representa una unidad contada para la cual se disminuye el recuento. Una instrucción es una instrucción de muestra cuando el recuento se reduce a cero para cualquier unidad contada de la instrucción. Cuando se alcanza un valor umbral, tal como cero, el recuento se reinicia al valor SF 740 y comienza a contar hacia atrás tal como se describe en los puntos a-d anteriores. En todos los casos de los modos de recuento, la notificación, si corresponde, se produce después de finalizar la última unidad de operación de la instrucción de muestra.

En el modo de muestreo dirigido, el muestreo dirigido se produce cuando el bit-N 710 es uno y la instrucción RINEXT se ejecuta satisfactoriamente. La instrucción de muestra es la siguiente instrucción secuencial (NSI) después de la instrucción RINEXT. Si la siguiente instrucción secuencial es una instrucción de tipo ejecución, la instrucción de muestra es la instrucción de destino de la instrucción de tipo ejecución. El muestreo dirigido puede producirse en el modo de recuento de ciclos o recuento de instrucciones. El muestreo de recuento continúa junto con el muestreo dirigido y cualquiera de sus acciones resultantes, y no se ve afectado de ninguna manera, excepto que si la instrucción de muestra determinada a partir del muestreo de recuento es la misma instrucción determinada por el muestreo dirigido, no se almacenan dos grupos de notificación.

Sea cual sea el modo de muestreo, cuando se identifica una instrucción de muestra mediante la ejecución de la instrucción RINEXT, se almacena un grupo de notificación. Sin embargo, los controles de instrumentación en tiempo de ejecución Y 738, Z 739, Qs 614 y Ps 612 continúan siendo efectivos.

El muestreo de recuento de ciclos y el recuento de instrucciones determina un intervalo aproximado que está sujeto a una cantidad de variabilidad en base a eventos internos del sistema y condiciones de excepción. La cuenta regresiva comienza cuando la instrumentación en tiempo de ejecución efectúa la transición de deshabilitada a habilitada. El muestreo dirigido está sujeto a una menor cantidad de variabilidad, dependiendo de cualquier evento que pueda interponerse entre la finalización de RINEXT y la NSI. Cabe destacar que una interrupción puede hacer que lo que se pensaba que era la NSI dejara de ser la NSI.

El muestreo, independientemente del modo, identifica una instrucción de muestra. Una vez que se identifica una instrucción de muestra, la recopilación se detiene tras finalizar la ejecución de la instrucción de muestra y comienza la notificación. Entonces se aplican los diversos controles de notificación que rigen la notificación. La recopilación se reanuda cuando el almacenamiento del grupo de notificación queda pendiente.

Cuando no está en el modo de ejecución transaccional, el almacenamiento de un grupo de notificación queda pendiente tras finalizar la ejecución de una instrucción de muestra. Cuando se encuentra en el modo de ejecución transaccional, tras finalizar la ejecución de una instrucción de muestra, el almacenamiento de un grupo de notificación se aplaza hasta que finaliza la transacción y luego queda pendiente. Cuando el almacenamiento de un grupo de notificación está aplazado o pendiente, se puede eliminar si se reconoce alguna de las siguientes interrupciones: 1) interrupción del programa; 2) interrupción urgente de verificación de la máquina; 3) reinicio de interrupción; y 4) interrupción de llamada del supervisor.

Cualquier interrupción de E/S pendiente, externa y de verificación de máquina reprimible permanece pendiente hasta que o bien el grupo de notificación se haya almacenado o bien los controles de instrumentación en tiempo de ejecución determinan que un grupo de notificación no se va a almacenar.

Cada modo puede, o no, permitir un conjunto diferente de controles de notificación. Cuando el modo de muestreo es recuento de instrucciones o recuento de ciclos, pero también se usa el muestreo dirigido, es posible que la misma instrucción de muestra se identifique mediante múltiples procedimientos de muestreo. Cuando esto ocurre, y los controles de notificación que se utilizarán difieren según el modo de muestreo, se aplican los controles de notificación asociados con el muestreo dirigido.

La determinación exacta de un intervalo destinado a muestrear una instrucción particular en general no es viable, debido a los eventos asincrónicos y no solicitados del sistema que se pueden producir. En cambio, la instrucción RINEXT se puede utilizar para designar más de cerca una instrucción de muestra.

Cuando se encuentra en el modo de recuento de ciclos o en el modo de recuento de instrucciones, la instrucción RINEXT puede emitirse en una estrecha proximidad de la instrucción de muestra identificada a partir del muestreo de recuento de instrucciones o recuento de ciclos. El contenido del grupo de notificación asociado es como si la instrucción de muestra se identificara como la NSI de la instrucción RINEXT y no como si se aplicara una identificación de recuento de ciclos o recuento de instrucciones de la instrucción de muestra.

La ejecución de RINEXT puede ejecutarse como una no operación si se cumple una o más de las siguientes condiciones de excepción: 1. Los controles de instrumentación en tiempo de ejecución no son válidos. 2. En el estado de problema, Ps 612 de los controles actuales de instrumentación en tiempo de ejecución es cero, lo que indica que la notificación en estado de problema no está permitido. 3. En el estado supervisor, Qs 614 de los controles actuales de instrumentación en tiempo de ejecución es cero, lo que indica que no se permite la notificación de estado supervisor. 4. El bit-N 710 de los controles actuales de instrumentación en tiempo de ejecución es cero, lo que indica que la instrucción RINEXT en sí misma no está permitida. 5. El almacenamiento está suprimido. 6. Un campo en la PSW actual indica que la instrumentación en tiempo de ejecución está deshabilitada. 7. Se excedería un umbral dependiente del modelo. El número de veces que se ha emitido RINEXT en un período de tiempo ha excedido un límite dependiente del modelo. 8. Existe una condición de memoria intermedia llena del programa. 9. Existe una condición de instrumentación detenida en tiempo de ejecución. 10. La siguiente instrucción secuencial es una instrucción de inicio de ejecución interpretativa. 11. La siguiente instrucción secuencial es una instrucción de llamada de supervisor.

Volviendo a la FIG. 8, en general se muestra una realización de memoria intermedia de recopilación 508. Como se describe previamente, cuando la instrumentación en tiempo de ejecución está habilitada durante la ejecución del programa, los datos de la instrumentación en tiempo de ejecución se recopilan dentro del procesador 106. En una realización, el lugar donde se recopilan los datos dentro del procesador 106 es la memoria intermedia de recopilación 508 y, opcionalmente, una memoria intermedia de datos de instrucciones. En una realización, la memoria intermedia de recopilación 508 es una memoria intermedia interna del procesador 106 que se usa para guardar los registros recopilados más recientes. Cuando se detecta un punto desencadenante de muestra, los registros se copian de la memoria intermedia de recopilación 508 a la memoria intermedia del programa 522 como parte de un grupo de notificación que se escribe en la memoria intermedia del programa 522. En una realización, los registros se copian de la memoria intermedia de recopilación de una forma no destructiva.

La memoria intermedia de recopilación 508 puede denominarse una "memoria intermedia de recopilación de hardware" puesto que la memoria intermedia de recopilación 508 está ubicada en el procesador y en una realización implementada como una matriz de pares de registros que representan la dirección de instrucción 802 y los metadatos de evento 804 de un evento dado. En una realización, la memoria intermedia de datos de instrucciones también se implementa mediante una matriz de pares de registros. Un ejemplo de un evento es una bifurcación tomada para la cual el par de registros puede contener la dirección de instrucción de la bifurcación, y los metadatos pueden contener el destino de la bifurcación, así como información sobre el comportamiento histórico de la bifurcación. En una realización, los pares de registros se ordenan y actualizan secuencialmente a medida que se producen eventos en el flujo de instrucciones. Se mantiene un contador para indicar el índice de la entrada actualizada más recientemente en la matriz. En una realización, la memoria intermedia de recopilación 508 es una memoria intermedia circular, y cuando la memoria intermedia de recopilación 508 está llena, el siguiente evento sobrescribe la primera entrada en la matriz, y la actualización secuencial de los pares de registros de la matriz se reinicia en eventos posteriores. De este modo, suponiendo una matriz CB[0] a CB[N-1] y un contador i que indica el último índice actualizado, el rastro de eventos capturados estaría representado por la secuencia CB[i], CB[i-1] ... CB[1], CB[0], CB[N-1], CB[N-2] ... CB[i+1]. En otra realización, se usan dos punteros: un puntero al encabezado que apunta a la entrada más antigua en la memoria intermedia, y un puntero posterior/actual que apunta a la entrada más nueva en la memoria intermedia.

Los eventos que representan un estado del procesador 106 en cualquier punto de ejecución dado se capturan secuencialmente en la memoria intermedia de recopilación 508. La memoria intermedia de recopilación 508 se usa para capturar un conjunto de registros cuyo contenido notifica sobre eventos reconocidos por el procesador 106 durante la ejecución del programa (p. ej., ejecución de una o más bifurcaciones tomadas, eventos de aborto de ejecución transaccional, el operando de una instrucción RIEMIT, etc.). En una realización, los eventos reconocidos dependen del contenido del RIC-CB que se muestra en la FIG. 7. Las entradas en la realización de la memoria intermedia de recopilación 508 que se muestra en la FIG. 8 incluyen una dirección de instrucción de evento 802 y otros metadatos de evento 804 pertinentes. Los ejemplos de metadatos de eventos 804 incluyen, pero no se limitan a: la dirección de instrucción de una bifurcación tomada y su destino, incluida cierta información sobre el comportamiento

histórico de la bifurcación; la dirección de instrucción de una instrucción RIEMIT y un valor de registro respectivo; y la dirección de una instrucción de aborto de transacción y un punto de entrada de recuperación de la transacción respectiva.

5 La realización de la memoria intermedia de recopilación 508 que se muestra en la FIG. 8 es capaz de almacenar hasta treinta y dos entradas (es decir, información sobre treinta y dos eventos), con cada dirección de instrucción 802 especificada por sesenta y cuatro bits (p. ej., bits 0: 63), y metadatos de evento 804 por sesenta y cuatro bits (p. ej., bits 64: 127). El tamaño de la memoria intermedia de recopilación. (R_{CB}) es un recuento dependiente del modelo, que representa un número de registros. En la realización de la memoria intermedia de recopilación 508 que se muestra en
10 la FIG. 8, el tamaño de bytes de la memoria intermedia de recopilación es un múltiplo del tamaño de registro de dieciséis bytes. En una realización, el tamaño de la memoria intermedia de recopilación es un número de registros mayor o igual a la diferencia entre el recuento del grupo de notificación más grande (R_{RG}) del modelo y el recuento de los registros en un grupo de notificación que no se adquieren de la memoria intermedia de recopilación (R_{NC}). Por lo tanto, en una realización, el tamaño de la memoria intermedia de recopilación se expresa como:

15

$$R_{CB} \geq (R_{RG} - R_{NC}).$$

En una realización, el contenido de la memoria intermedia de recopilación 508 y la memoria intermedia de datos de instrucciones (si se usa una) se elimina o se ve afectado de alguna manera por los siguientes eventos: (1) una interrupción; (2) el bit PSW que activa y desactiva la herramienta de instrumentación en tiempo de ejecución (p. ej., el bit 24) cambia de uno a cero; y (3) cuando se identifica una instrucción de muestra cuando la herramienta de instrumentación en tiempo de ejecución está en un modo de ejecución transaccional (en este caso, la actualización adicional de la memoria intermedia de datos de recopilación 508 y la memoria intermedia de datos de instrucciones se detiene y se reanuda cuando finaliza la transacción, en ese momento, hay un almacenamiento del grupo de notificación pendiente y se eliminan la memoria intermedia de recopilación 508 y las memorias intermedias de datos de instrucciones.
20
25

En una realización, tal como el sistema de ordenador proveedor de alojamiento emulado que se muestra en la FIG. 1B, el ordenador proveedor de alojamiento proporciona acceso compartido a la herramienta de instrumentación en tiempo de ejecución y un estado de guardar/restaurar instrumentación en tiempo de ejecución para cada usuario distinto. El estado de instrumentación en tiempo de ejecución guardado se almacena utilizando registros y/o memoria que pertenecen al proveedor de alojamiento.
30

En realizaciones, las capacidades adicionales pueden afectar la recopilación de datos y pueden considerarse como puntos de recopilación de datos adicionales sin perturbar sustancialmente el recuento de instrucciones regulares o el muestreo de recuento de ciclos descrito previamente. Estos incluyen la ejecución de una instrucción RIEMIT, que recopila el valor de un registro general almacenándolo en la memoria intermedia de recopilación 508. Además, los bits de control de recopilación de datos en los controles de instrumentación en tiempo de ejecución descritos previamente se pueden usar para personalizar los tipos de datos recopilados (p. ej., los bits de control E, C, R y B). De esta forma, el tipo de datos recopilados es programable.
35
40

En una realización, se implementa una memoria intermedia de datos de instrucciones para recopilar datos de instrucciones de muestra dependientes del modelo que se utilizan para construir un registro de instrucciones de instrumentación en tiempo de ejecución. La memoria intermedia de datos de instrucciones recopila datos de una instrucción antes de estar disponible cuando la instrucción se identifica como una instrucción de muestra. En una realización, la memoria intermedia de datos de instrucciones es una ubicación de almacenamiento/memoria intermedia de hardware en el procesador donde se guarda información sobre una instrucción que se convertirá en el desencadenante a medida que se guarda un punto de muestra, de modo que durante el procedimiento de cierre de sesión, se puede escribir juntos con datos de la memoria intermedia de recopilación 508. Similar al memoria intermedia de recopilación 508, incluye la dirección de la instrucción y los metadatos asociados con esa instrucción. Los metadatos en la memoria intermedia de datos de instrucciones a menudo dependen de la máquina y pueden incluir, entre otros: información relacionada con un fallo de memoria caché e información relacionada con la predicción de una bifurcación.
45
50

Según las realizaciones, otros datos recopilados pueden no ser de la memoria intermedia de recopilación 508 y tampoco de la memoria intermedia de datos de instrucciones. Los ejemplos incluyen datos utilizados para formar parte de los registros siguientes: (1) el primer registro de un grupo de notificación: registro de inicio o marca de tiempo; y (2) pueden crearse tipos adicionales de registros para cada grupo de notificación y, por lo tanto, no almacenarse en la memoria intermedia de recopilación 508, dichos registros, cuando están presentes, pueden colocarse en la sección extra o dependiente de la máquina de un grupo de notificación. Estos registros generados por el sistema se denominan en el presente documento "registros de información del sistema".
55
60

La FIG. 9 representa un ejemplo de alto nivel de un grupo de notificación 900 almacenado en la memoria intermedia del programa 522 en un punto de muestra. El número de registros válidos de la memoria intermedia de recopilación

508, representado por R_{VCB} , está en el intervalo de

$$0 \leq R_{VCB} \leq R_{CB}$$

5 (donde R_{CB} es el tamaño de la memoria intermedia de recopilación 508). El tamaño de un grupo de notificación 900 en registros, representado por R_{R9} , es igual a $2^{\langle R_{GS}+1 \rangle}$, donde R_{GS} es el tamaño del grupo de notificación como un exponente. Un número de registros dependiente del modelo (R_{NC}), tal como un registro de instrucciones, copiado desde una ubicación distinta de la memoria intermedia de recopilación 508, puede, o no, copiarse de forma no destructiva cuando se usa en un grupo de notificación 900. En una realización, si el R_{VCB} es menos que $R_{R9} - R_{NC}$, entonces el resto del grupo de notificación 900 se llena con registros de tipo relleno, de manera que el recuento requerido de R_{R9} registros se cumple en el grupo de notificación 900. En el ejemplo de la FIG. 9, $R_{R9} = 8$, $R_{GS} = 2$ y $R_{NC} = 4$. En una forma de realización R_{R9} está en el intervalo de

$$2 \leq R_{RG} \leq 2^8$$

15 registros. Según una realización, R_{GS} está limitado a un máximo dependiente del modelo que es inferior a siete, por lo que el tamaño del grupo de notificación más grande en un modelo puede ser inferior a 256, pero sigue siendo una potencia de dos.

20 El grupo de notificación 900 de ejemplo que se muestra en la FIG. 9 incluye una sección de encabezamiento 902, una sección de cuerpo 904, una sección de registros extra 906 y una sección de pie 908. La sección de encabezamiento 902 puede incluir un registro de inicio o un registro de marca de tiempo para contener información de estado, seguimiento y/o temporización. La sección de cuerpo 904 puede incluir una variedad de registros para eventos e información muestreada de la memoria intermedia de recopilación 508. Los eventos y la información pueden representar, por ejemplo, información de estado capturada por una instrucción de emisión, un aborto de ejecución transaccional, una llamada, un retorno, una bifurcación y un relleno. La sección de registros extra 906 puede incluir registros dependientes del modelo. La sección de pie 908 puede incluir un registro de instrucción que contiene información sobre la ejecución de una instrucción de muestra.

30 Según una realización, el primer registro de un grupo de notificación 900 está en la sección de encabezamiento 902 y es o bien un registro de inicio o bien un registro de marca de tiempo para contener información de estado, seguimiento y/o temporización. Se almacena un registro de inicio para el primer grupo de notificación 900 almacenado en una memoria intermedia del programa 522 (es decir, cuando la RCA 706 es igual a la ROA 702). Un registro de marca de tiempo se almacena como el primer registro de cada grupo de notificación posterior. Ni el registro de inicio ni el registro de marca de tiempo se adquieren de la memoria intermedia de recopilación 508 y, por lo tanto, son parte del recuento R_{NC} . Cuando R_{R9} es igual R_{NC} , el cuerpo es nulo y no se almacenan registros de la memoria intermedia de recopilación 508 en el grupo de notificación. Cuando R_{R9} es mayor que R_{NC} y R_{NC} es igual a dos, el contenido de los registros uno a $R_{R9}-R_{NC}$ de un grupo de notificación están en la sección de cuerpo de un grupo de notificación 900 y se adquieren de la memoria intermedia de recopilación 508.

40 Los registros de la memoria intermedia de recopilación almacenados en el cuerpo de un grupo de notificación 900 se organizan del más antiguo al más nuevo según una realización, y el registro más nuevo se almacena en el registro $R_{R9}-R_{NC}$ cuando la memoria intermedia de recopilación 508 tiene un número suficiente de registros válidos. Cuando se almacenan múltiples registros de la memoria intermedia de recopilación 508 como parte de un grupo de notificación 900, el registro uno es el registro de memoria intermedia de recopilación más antiguo notificado, aunque la memoria intermedia de recopilación 508 puede tener registros aún más antiguos que no caben en el cuerpo, puesto que el registro más nuevo en la memoria intermedia de recopilación 508 debe notificarse. Si el número de registros válidos en la memoria intermedia de recopilación 508 es insuficiente para llenar un grupo de notificación 900, se utilizan uno o más registros de relleno para llenar tantos registros de grupos de notificación como sea necesario, comenzando en el registro inmediatamente después del registro más reciente de la memoria intermedia de recopilación. Es decir, cuando sea necesario, los registros de tipo relleno se colocan en la parte inferior de la sección de cuerpo 904.

55 Cuando R_{R9} es mayor que R_{NC} , y R_{NC} es mayor que dos, el contenido de los registros ($R_{R9}-R_{NC}+1$) mediante ($R_{R9}-2$) son la sección extra 906 de un grupo de notificación 900 y contienen uno o más registros extra o dependientes del modelo. Cuando R_{NC} es igual a dos, la sección extra 906 es nula y no se almacenan registros extra. Los registros de tipo extra o dependientes del modelo en la sección extra 906 ocupan el espacio requerido y cualquier espacio restante se convierte en parte de la sección de cuerpo 904.

60 En una realización, el registro $R_{R9}-1$ de un grupo de notificación 900 es la sección de pie 908 y contiene un registro de instrucción que representa una instrucción de muestra. El registro de instrucción no se adquiere de la memoria intermedia de recopilación 508 y se adquiere parcialmente de la memoria intermedia de datos de instrucciones.

Según una realización, un grupo de notificación 900 se almacena en un límite integral del tamaño de bytes del grupo

de notificación en la memoria intermedia del programa 522.

5 Cuando se intenta almacenar un grupo de notificación 900, se almacenan todos los registros del grupo o se almacenan menos registros. La RCA 706 solo se avanza si se almacenan todos los registros en el grupo de notificación 900. En una realización, cuando se almacena un grupo de notificación 900, los registros en el grupo de notificación 900 no se almacenan en ningún orden particular. Sin embargo, después de que se hayan almacenado todos los registros del grupo de notificación 900, se actualiza el registro de inicio.

10 En una realización, la protección de baja dirección, la protección de traducción de dirección dinámica (DAT) y/o la protección controlada por clave se aplican a un almacenamiento de un grupo de notificación 900. La protección controlada por clave obtiene la clave de los controles de instrumentación en tiempo de ejecución (p. ej., la clave 626) en lugar de la PSW actual. El espacio de direcciones utilizado es local o primario según lo especificado mediante un bloque de control de instrumentación en tiempo de ejecución (p. ej., bit H 608). Si se reconoce una excepción de acceso en la ubicación del grupo de notificación o en la ubicación del registro de inicio, o en ambas, no se realiza ningún almacenamiento y el contenido del grupo de notificación no se almacena.

15 En una realización, la RCA 706 se incrementa en dieciséis para cada registro del grupo de notificación almacenado. La actualización real se realiza una vez por grupo de notificación 900, incrementándose por el tamaño agregado del grupo de notificación 900. Dicho incremento de la RCA 706 se denomina en el presente documento como un avance de la dirección actual. Por lo tanto, un almacenamiento satisfactorio de un grupo de notificación 900 está representado por si la RCA 706 ha avanzado o no y si un campo que almacena el número de grupos de notificación (NRG) en el registro de inicio se incrementa en uno. La RCA 706 no avanza cuando se reconoce una condición de instrumentación detenida en tiempo de ejecución. Existe una condición de memoria intermedia llena del programa cuando la RCA 706 es igual a la suma de uno más la RLA 704 ($RCA = 1 + RLA$). Cuando existe la condición de memoria intermedia llena del programa, según una realización, no se almacena un grupo de notificación 900. La condición de memoria intermedia llena del programa se reconoce cuando la RCA 706 ha avanzado y provoca que la condición exista. Cuando la RCA 706 ha avanzado y no se reconoce la condición de memoria intermedia llena del programa, si la dirección avanzada ha cruzado un límite de página, se puede reconocer una excepción de acceso. Dicha excepción de acceso se indica como un acceso al almacenamiento de efecto lateral. La interrupción externa de alerta de medición provocada por el reconocimiento de la condición de memoria intermedia llena del programa se denomina en el presente documento interrupción total de la memoria intermedia.

20 Cuando se reconoce la condición de memoria intermedia llena del programa, una realización puede realizar las siguientes acciones en el orden indicado:

35 1. Un bit en un campo de parada (S) en el registro de inicio se establece en uno para indicar la existencia de una condición de memoria intermedia llena del programa.

40 2. Una interrupción total de la memoria intermedia se hace pendiente al establecer un bit en un bloque de control de instrumentación en tiempo de ejecución (p. ej., el bit L 624) en uno. La interrupción total de la memoria intermedia permanece pendiente hasta que el bit L se pone a cero. El bit L puede ponerse a cero por cualquiera de los siguientes casos: ejecución satisfactoria de una instrucción LRIC que especifica L como cero; presentación de la interrupción total de la memoria intermedia; un reinicio de la CPU; o la ejecución satisfactoria de una instrucción MRIC que implícitamente establece L en cero si la condición de memoria intermedia llena del programa no se carga.

45 En una realización, mientras existe la condición de memoria intermedia llena del programa, se aplican las siguientes consecuencias: la ejecución de RINEXT no realiza ninguna operación; la ejecución de RIEMIT no realiza ninguna operación; depende del modelo si se produce una disminución adicional del intervalo, si ocurre, también depende del modelo si se reinicializa al valor del factor de escala (SF) en el bloque de control (p. ej., SF 740) tras disminuir a cero, si el intervalo disminuye a cero, no se reconoce una instrucción de muestra; y la ejecución de las otras instrucciones de instrumentación en tiempo de ejecución no se ven afectadas.

50 En una realización, una condición de instrumentación detenida en tiempo de ejecución existe primero cuando un código de versión en el registro de marca de tiempo del grupo de notificación 900 que se va a almacenar no es igual a un código de versión en el registro de inicio del grupo de notificación 900, y continúa hasta que la instrumentación en tiempo de ejecución se reanuda después de un cambio en los controles de instrumentación en tiempo de ejecución (p. ej., un bloque de control) que elimina la desigualdad potencial. La condición de instrumentación detenida en tiempo de ejecución se reconoce inicialmente en el punto donde se reconoce el registro de inicio. La interrupción externa de alerta de medición provocada por el reconocimiento de la condición de instrumentación detenida en tiempo de ejecución se denomina en el presente documento interrupción de instrumentación detenida en tiempo de ejecución. Cuando se reconoce la condición de instrumentación detenida en tiempo de ejecución, se aplican las siguientes etapas, en el orden indicado.

55 1. Un campo de detención (H) en el registro de inicio se establece en uno.

2. Una interrupción de instrumentación detenida en tiempo de ejecución queda pendiente al establecer el bit G (p. ej., el bit G 62) en un bloque de control a uno. La interrupción de instrumentación detenida en tiempo de ejecución permanece pendiente hasta que se borra. En una realización, se borra por cualquiera de las siguientes maneras: ejecución satisfactoria de una instrucción LRIC que especifica el bit G como cero; la presentación de la interrupción de instrumentación detenida en tiempo de ejecución; reinicio de la CPU; ejecución satisfactoria de una instrucción MRIC que implícitamente establece G en cero si se carga la RCA 706 que es igual a la ROA 702 (esto también es un borrado de la condición de instrumentación detenida en tiempo de ejecución).

Mientras existe la condición de instrumentación detenida en tiempo de ejecución, se aplican las siguientes consecuencias: 1. la ejecución de RINEXT no realiza ninguna operación; 2. la ejecución de RIEMIT no realiza ninguna operación; 3. depende del modelo si se produce una disminución adicional del intervalo (p. Ej., RSIC 742), si ocurre, también depende del modelo si se reinicia al valor de SF (p. ej., SF 740) tras disminuirlo a cero, si el intervalo disminuye a cero, no se reconoce una instrucción de muestra; 4. no se almacenan grupos de notificación adicionales 900; 5. la RCA 706 no se avanza; y 6. la ejecución de las otras instrucciones de instrumentación en tiempo de ejecución no se ve afectada. Si el bit G se carga como uno durante la ejecución de una instrucción LRIC o MRIC, la interrupción de instrumentación detenida en tiempo de ejecución queda pendiente. El bit G se establece implícitamente en cero cuando una instrucción MRIC especifica que la ROA 702 es igual a la RCA 706. Si el bit U 622 en los controles de instrumentación en tiempo de ejecución es igual a cero, se deshabilita una interrupción de instrumentación detenida en tiempo de ejecución.

Volviendo ahora a la FIG. 10, en general se muestra una realización de un registro de inicio 1000. En una realización, el registro de inicio 1000 se almacena en la sección de encabezamiento 902 para el primer grupo de notificación 900 almacenado en una memoria intermedia del programa 522 (es decir, cuando la RCA 706 es igual a la ROA 702), y el registro de inicio 1000 se actualiza después de que el primer grupo de notificación 900 se almacena en la memoria intermedia del programa 522. El registro de inicio 1000 que se muestra en la FIG. 10 incluye un campo de tipo de registro 1002, un número de campos de grupos de notificación (NRG) 1004, un campo de versión 1014, un campo RGS 1006, un campo de parada (S) 1008, un campo de capacidad (T) 1016, un campo de detención (H) 1010 y un campo de reloj horario (TOD) 1012. El registro de inicio 1000 se inicializa utilizando valores de control de instrumentación en tiempo de ejecución (p. ej., del RICCB u otro bloque de control). El campo de tipo de registro 1002 se establece en "02" para indicar que el registro es un registro de inicio 1000. El campo NRG 1004 se establece en uno para indicar que hay un grupo de notificación 900 almacenado en la memoria intermedia del programa 522. El campo NRG 1004 se actualiza cada vez que se almacena un nuevo grupo de notificación en la memoria intermedia del programa 522. El campo de versión 1014 almacena un número entero cuyo valor indica un nivel de cambio de ingeniería funcional (p. ej., del módulo de instrumentación 506). Durante la inicialización del registro de inicio 1000, el campo RGS 1006 se actualiza con el valor actual del RGS 1006 (p. ej., del RICCB) para indicar el tamaño de todos los grupos de notificación 900 en la memoria intermedia del programa 522. El campo S 1008 se establece en cero para indicar que la memoria intermedia del programa 522 no está llena. Si la memoria intermedia del programa 522 se llena, el campo S 1008 se cambia a uno lo cual indica que existe una condición de memoria intermedia llena del programa. El campo T 1016 se establece en cero durante la inicialización cuando el grupo de notificación 900 representa una muestra capturada en una CPU que funciona con su capacidad de CPU, y se establece en uno cuando el grupo de notificación 900 representa una muestra capturada en una CPU que opera en una capacidad de CPU secundaria. El campo H 1010 se establece en cero durante la inicialización y luego se establece en uno si la instrumentación en tiempo de ejecución se ha detenido debido a razones distintas a la condición de memoria intermedia llena. Cuando el campo H 1010 es igual a uno, el bit G 620 en el RICCB también se establece en uno. El campo de reloj TOD 1012 está configurado para indicar la hora del día en que el grupo de notificación que contiene el registro de inicio 1000 se almacenó inicialmente. Las porciones reservadas del registro de inicio 1000 (y cualquier porción reservada en el resto de los registros del grupo de notificación) no tienen valores predecibles.

La FIG. 11 es una realización de un registro de marca de tiempo 1100 en la sección de encabezamiento 902 de un grupo de notificación 900. Un registro de marca de tiempo 1100 se almacena como registro cero (es decir, el primer registro) de cada grupo de notificación 900 que no sea el primer grupo de notificación 900 en la memoria intermedia del programa 522. Se crea un registro de marca de tiempo 1100 cuando un grupo de notificación 900 se almacena en la memoria intermedia del programa 522 y la ROA 702 no es igual a la RCA 706. La realización del registro de marca de tiempo 1100 que se muestra en la FIG. 11 incluye un tipo de registro 1102, un campo de versión 1106, un campo de capacidad (T) 1108 y un campo de reloj TOD 1104. El tipo de registro 1102 se establece en "03" para indicar que el registro es un registro de marca de tiempo 1100. El campo de versión 1106 almacena un número entero cuyo valor indica un nivel de cambio de ingeniería funcional (p. ej., del módulo de instrumentación 506). El campo T 1108 se establece en cero durante la inicialización cuando el grupo de notificación 900 representa una muestra capturada en una CPU que funciona con su capacidad de CPU, y se establece en uno cuando el grupo de notificación 900 representa una muestra capturada en una CPU que opera en una capacidad de CPU secundaria. El campo de reloj TOD 1104 está configurado para indicar la hora del día en que el grupo de notificación 900 que contiene el registro de marca de tiempo 1100 se almacenó inicialmente.

La sección de cuerpo 904 del grupo de notificación 900 puede incluir una variedad de eventos e información muestreados de la memoria intermedia de recopilación 508. Los eventos y la información pueden representar, por ejemplo, información de estado capturada por una instrucción de emisión, un aborto de modo de ejecución

transaccional, una llamada, un retorno, una bifurcación y un relleno.

Un registro de emisión se crea y se almacena en la sección de cuerpo 904 de un grupo de notificación 900 en la memoria intermedia del programa 522 tras una ejecución satisfactoria de una instrucción RIEMIT. La FIG. 12 ilustra tres realizaciones diferentes de un registro de emisión 1202, 1204 y 1206 que varían en base a si se está implementando un modo de direccionamiento de 64, 31 o 24 bits. Los registros de emisión 1202, 1204 y 1206 incluyen todos un campo de tipo de registro 1208, un campo de código de dirección de instrucción (C) 1210, un campo de dirección de instrucción 1212 y un campo de datos de emisión 1214. El campo de tipo de registro 1208 se establece en "10" para indicar que el registro es un registro de emisión 1202, 1204 o 1206. En una realización, el campo C 1210 es un número entero sin signo cuyo valor es un código que describe cómo se representan las posiciones de bit de dirección de instrucción en la PSW actual en el registro de emisión 1202, 1204 y 1206. Cuando el campo C 1210 es igual a cero, las posiciones de bit 22-62 de la dirección de instrucción (IA) de la PSW actual se colocan en las posiciones de bit 22-62 del registro y las posiciones de bit 0-21 de la dirección de instrucción son todos ceros. Cuando el campo C 1210 es igual a uno, las posiciones de bit 22-62 de la dirección de instrucción de la PSW actual se colocan en las posiciones de bit 22-62 del registro y las posiciones de bit 0-21 de la dirección de instrucción no son todos ceros. Cuando el campo C es igual a uno, un bit de dirección máxima excedida (p. ej., MAE 712) en el bloque de control de instrumentación en tiempo de ejecución se establece en uno si el registro se almacena como parte de un grupo de notificación 900. El campo de dirección de instrucción 1212 contiene la dirección de instrucción de la instrucción RIEMIT o la instrucción de tipo ejecución si el RIEMIT era el destino de una instrucción de tipo ejecución. Como se muestra en la FIG. 12, el campo de dirección de instrucción 1212 varía en función del modo de direccionamiento (p. ej., 64, 31 o 24 bits). El campo de emisión de datos 1214 incluye los datos del registro general especificado por la instrucción RIEMIT.

Un registro de aborto del modo de ejecución transaccional se crea mediante un aborto implícito o mediante la ejecución del aborto de transacción y se almacena en la sección de cuerpo 904 de un grupo de notificación 900 en la memoria intermedia del programa 522. La FIG. 13 ilustra tres realizaciones diferentes de un registro de aborto 1302, 1304 y 1306 que varían en base a si se está implementando un modo de direccionamiento de 64, 31 o 24 bits. Los registros de aborto 1302, 1304 y 1306 incluyen un campo de tipo de registro 1308, un campo de código de dirección de instrucción (C) 1310, un campo de dirección de instrucción 1312 y un campo dependiente del modelo 1314. El campo de tipo de registro 1308 se establece en "11" para indicar que el registro es un registro de aborto 1302, 1304 o 1306. En una realización, el campo C 1310 es un número entero sin signo cuyo valor es un código que describe cómo se representan las posiciones del bit de dirección de instrucción en la PSW actual en el registro de aborto 1302, 1304 y 1306, y tiene propiedades similares al campo C 1210 descrito anteriormente con respecto a la FIG. 12. El campo de dirección de instrucción 1312 contiene la dirección de instrucción de la instrucción abortada o la instrucción de tipo de ejecución si la instrucción abortada era el destino de una instrucción de tipo ejecución. Como se muestra en la FIG. 13, el campo de dirección de instrucción 1312 varía en función del modo de direccionamiento (p. ej., 64, 31 o 24 bits). El campo de datos dependiente del modelo 1314 incluye cualquier dato dependiente del modelo asociado con el aborto.

Se crea un registro de llamadas mediante la ejecución de una instrucción de bifurcación que se clasifica como una instrucción de bifurcación de tipo llamada. El registro de llamadas se almacena en la sección de cuerpo 904 de un grupo de notificación 900 en la memoria intermedia del programa 522. En una realización, las bifurcaciones de tipo llamada incluyen: BIFURCAR Y GUARDAR (BASR) cuando el campo R2 es distinto de cero, BIFURCAR Y GUARDAR (BAS), BIFURCACIÓN RELATIVA Y GUARDAR LARGO, BIFURCACIÓN RELATIVA Y GUARDAR POR MUCHO TIEMPO, BIFURCAR Y ENLAZAR (BALR) cuando el campo R2 es distinto de cero, BIFURCAR Y ENLAZAR (BAL), y BIFURCAR Y GUARDAR Y ESTABLECER MODO cuando el campo R2 no es cero. La FIG. 14 ilustra tres realizaciones diferentes de un registro de llamadas 1402, 1404 y 1406 que varían en base a si se está implementando un modo de direccionamiento de 64, 31 o 24 bits. Los registros de llamadas 1402, 1404 y 1406 incluyen un campo de tipo de registro 1408, un campo de código de dirección de instrucción (C) 1410, un campo de dirección de instrucción 1412, un campo de buen comportamiento (W) 1416 y un campo de dirección de destino 1414. El campo de tipo de registro 1408 se establece en "12" para indicar que el registro es un registro de llamadas 1402, 1404 o 1406. El campo C 1410 es un número entero sin signo cuyo valor es un código que describe cómo se representan las posiciones del bit de dirección de instrucción en la PSW actual en el registro de llamadas 1402, 1404 y 1406, y tiene propiedades similares al campo C 1210 descrito anteriormente con respecto a la FIG. 12. El campo de dirección de instrucción 1412 contiene la dirección de la instrucción de bifurcación o la instrucción de tipo ejecución si la bifurcación era el destino de una instrucción de tipo ejecución. Un campo de AW 1414 que tiene un valor de uno indica que la bifurcación se predijo correctamente, y un valor de cero indica que la bifurcación no se predijo correctamente. El campo de dirección de destino 1414 contiene la dirección de destino de la bifurcación (también denominada "ubicación llamada").

Los registros de retorno y los registros de transferencia en la sección de cuerpo 904 de un grupo de notificación 900 en la memoria intermedia del programa 522 tienen el mismo formato que los registros de llamadas 1402 1404 y 1406. En una realización, un registro de retorno tiene un campo de tipo de registro de "13" y se crea mediante la ejecución de una instrucción de bifurcación de tipo retorno, como una CONDICIÓN DE BIFURCACIÓN (BCR) cuando el campo R2 no es cero y la máscara es 15, CONDICIÓN DE BIFURCACIÓN (BC) cuando el bit J es uno y la máscara es 15, y BIFURCACIÓN Y ESTABLECER MODO cuando el campo R2 no es cero. Para el registro de retorno, el campo de dirección de instrucción contiene la dirección de la instrucción de bifurcación o instrucción de tipo ejecución si la

bifurcación es el destino de una instrucción de tipo ejecución, y el campo de dirección de destino contiene la ubicación de retorno.

5 En una realización, un registro de transferencia tiene un campo de tipo de registro de "14" y se crea mediante la ejecución de una instrucción de bifurcación de tipo retorno tal como: a. CONDICIÓN DE BIFURCACIÓN (BCR) cuando el campo R2 no es cero y la máscara está en el intervalo 1-14; b. CONDICIÓN DE BIFURCACIÓN (BC) cuando el bit J es cero o la máscara está en el intervalo 1-14; c. BIFURCACIÓN EN RECUENTO (BCT, BCTR, BCTG, BCT-GR); d. BIFURCACIÓN EN ÍNDICE ALTO (BXH, BXHG); e. BIFURCACIÓN EN ÍNDICE BAJO O IGUAL (BXLE, BXLEG); f. CONDICIÓN DE BIFURCACIÓN RELATIVA (BRC); g. CONDICIÓN LARGA DE BIFURCACIÓN RELATIVA (BRCL); h. BIFURCACIÓN RELATIVA EN RECUENTO (BRCT, BRCTG); i. BIFURCACIÓN RELATIVA EN RECUENTO ALTO (BRCTH); j. BIFURCACIÓN RELATIVA EN RECUENTO ALTO (BRCTH); j. BIFURCACIÓN RELATIVA EN ÍNDICE ALTO (BRXH, BRXHG); k. BIFURCACIÓN RELATIVA EN ÍNDICE BAJO O IGUAL (BRXLE, BRXLG); l. COMPARAR Y BIFURCAR (CRB, CGRB); m. COMPARACIÓN Y BIFURCACIÓN RELATIVA (CRJ, CGRJ); n. COMPARACIÓN INMEDIATA Y BIFURCACIÓN (CIB, CGIB); o. COMPARACIÓN RELATIVA INMEDIATA Y BIFURCACIÓN RELATIVA (CIJ, CGIJ); p. COMPARACIÓN LÓGICA Y BIFURCACIÓN (CLRB, CLGRB); q. COMPARACIÓN LÓGICA Y BIFURCACIÓN RELATIVA (CLRJ, CLGRJ); r. COMPARACIÓN LÓGICA INMEDIATA Y BIFURCACIÓN (CLIB, CLGIB); y s. COMPARACIÓN LÓGICA INMEDIATA Y BIFURCACIÓN RELATIVA (CLIJ, CLGIJ). El registro de transferencia se crea cuando se toma la bifurcación. Para el registro de transferencia, el campo de dirección de instrucción contiene la dirección de la instrucción de bifurcación o instrucción de tipo de ejecución si la bifurcación es el destino de una instrucción de tipo de ejecución, y el campo de dirección de destino contiene la ubicación de retorno.

25 Se utiliza un registro de relleno en la sección de cuerpo 904 de un grupo de notificación 900 cuando el número de registros válidos en la memoria intermedia de recopilación 508 no es suficiente para llenar un grupo de notificación 900 del RGS actual. La FIG. 15 es una realización de un registro de relleno 1500. La realización que se muestra en la FIG. 15 incluye un tipo de registro 1502 establecido en "00" para indicar que el registro es un registro de relleno 1500 y que los bytes restantes no están definidos.

30 La sección de registros extra 906 en el grupo de notificación 900, cuando está presente, puede contener registros dependientes del modelo. En una realización, el formato de un registro extra es similar al registro de relleno 1500, excepto que el tipo de registro se establece en "01" para indicar que el registro es un registro extra y los bytes restantes del registro extra contienen datos dependientes del modelo.

35 En una realización, la sección de pie 908 del grupo de notificación 900 incluye un registro de instrucción que contiene información sobre la ejecución de una instrucción de muestra. Se crea un registro de instrucción cuando se almacena un grupo de notificación 900 para una instrucción de muestra. La FIG. 16 ilustra tres realizaciones diferentes de un registro de instrucción 1602, 1604 y 1606 que varían en base a si se está implementando un modo de direccionamiento de 64, 31 o 24 bits. Los registros de instrucciones 1602, 1604 y 1606 incluyen todos un campo de tipo de registro 1608, un campo de código de dirección de instrucción (C) 1610, un campo de dirección de instrucción 1612 y un campo de memoria intermedia de datos de instrucciones (IDB) 1614. El campo de tipo de registro 1608 se establece en "04" para indicar que el registro es un registro de instrucción 1602, 1604 o 1606. El campo C 1610 es un número entero sin signo cuyo valor es un código que describe cómo se representan las posiciones del bit de dirección de instrucción en la PSW actual en el registro de instrucción 1602, 1604 y 1606, y tiene propiedades similares al campo C 1210 descrito anteriormente con respecto a la FIG. 12. El campo de dirección de instrucción 1612 contiene la dirección de la instrucción de muestra o la instrucción de tipo ejecución si la instrucción de muestra era el destino de una instrucción de tipo ejecución. Como se muestra en la FIG. 16, el campo de dirección de instrucción 1612 varía en función del modo de direccionamiento (p. ej., 64, 31 o 24 bits). El campo IDB 1614 incluye datos recopilados de la memoria intermedia de los datos de instrucciones y el contenido son dependientes del modelo.

50 La FIG. 17 representa un flujo de procedimiento para la notificación de instrumentación en tiempo de ejecución según una realización. En una realización, el procesamiento que se muestra en la FIG. 17 se realiza mediante el módulo de instrumentación 506. En el bloque 1702, se ejecuta un flujo de instrucciones en un procesador, tal como el procesador 106 o el procesador emulado 29. En el bloque 1704, se captura la información de instrumentación en tiempo de ejecución del flujo de instrucciones en ejecución, y en el bloque 1706 se crean registros de instrumentación en tiempo de ejecución (tales como los que se muestran en la memoria intermedia de recopilación 508) en base a la información de instrumentación capturada en tiempo de ejecución. Un punto de muestra de instrumentación en tiempo de ejecución de la instrucción de ejecución se detecta en el bloque 1708. En el bloque 1710, un grupo de notificación, tal como el grupo de notificación 900, se almacena en una memoria intermedia del programa de instrumentación en tiempo de ejecución, tal como la memoria intermedia del programa 522. El almacenamiento incluye determinar una dirección actual de la memoria intermedia del programa de instrumentación en tiempo de ejecución en base a los controles de instrumentación en tiempo de ejecución. En una realización, la dirección actual de los controles de instrumentación en tiempo de ejecución es la RCA 706 almacenada en el RIC-CB 700. El almacenamiento se basa en la dirección de origen (p. ej., ROA 702) y la dirección actual de la memoria intermedia del programa de instrumentación en tiempo de ejecución.

65 Como se describe anteriormente, las realizaciones pueden realizarse en forma de procedimientos y aparatos implementados por ordenador para poner en práctica esos procedimientos. Una realización puede incluir un producto

de programa de ordenador 1800 como se representa en la FIG. 18 en un medio 1802 legible/utilizable por ordenador con lógica de código de programa de ordenador 1804 que contiene instrucciones materializadas en medios tangibles como un artículo de fabricación. Los artículos de fabricación ejemplares para el medio 1802 legible/utilizable por ordenador pueden incluir discos flexibles, CD-ROM, discos duros, unidades de memoria flash de bus serie universal (USB) o cualquier otro medio de almacenamiento legible por ordenador, en el que, cuando la lógica de código de programa de ordenador 1804 es cargada y ejecutada por un ordenador, el ordenador se convierte en un aparato para poner en práctica la invención. Las realizaciones incluyen la lógica de código de programa de ordenador 1804, por ejemplo, ya sea almacenada en un medio de almacenamiento, cargada y/o ejecutada por un ordenador, o transmitida a través de algún medio de transmisión, tal como a través de cableado o cables eléctricos, a través de fibra óptica o por medio de radiación electromagnética, en el que, cuando la lógica de código de programa de ordenador 1804 es cargada y ejecutada por un ordenador, el ordenador se convierte en un aparato para poner en práctica la invención. Cuando se implementa en un microprocesador de propósito general, los segmentos de lógica de código de programa de ordenador 1804 configuran el microprocesador para crear circuitos lógicos específicos.

Los efectos y beneficios técnicos incluyen la capacidad de extraer datos de instrumentación en tiempo de ejecución en un entorno de laboratorio, para su análisis fuera de línea y en entornos de software en tiempo real dentro de los programas en tiempo de ejecución y bajo el control del programa. Un grupo de notificación se almacena en la memoria intermedia del programa cuya ubicación se determina a partir de los controles asociados con la instrumentación en tiempo de ejecución. Por lo tanto, los datos instrumentados almacenados como grupos de notificación en la memoria intermedia del programa están disponibles para el programa para su análisis posterior (o incluso un análisis agresivo durante la instrumentación en tiempo de ejecución).

La terminología utilizada en el presente documento tiene el propósito de describir realizaciones particulares solamente y no pretende ser limitante de la invención. Tal como se utilizan en este documento, las formas en singular "un", "una", "el" y "la" están destinadas a incluir también las formas plurales, a menos que el contexto indique claramente lo contrario. Se entenderá además que los términos "comprende" y/o "que comprende", cuando se usan en esta memoria descriptiva, especifican la presencia de características, números enteros, etapas, operaciones, elementos y/o componentes indicados, pero no descartan la presencia o la adición de una o más características, números enteros, etapas, operaciones, elementos, componentes y/o grupos de los mismos.

Las estructuras, materiales, acciones y equivalentes correspondientes de todos los medios o etapas más elementos de función en las reivindicaciones a continuación están destinadas a incluir cualquier estructura, material o acción para realizar la función en combinación con otros elementos reivindicados tal como se reivindica específicamente. La descripción de la presente invención se ha presentado con fines ilustrativos y descriptivos, pero no pretende ser exhaustiva o ni limitada a la invención en la forma desvelada. Muchas modificaciones y variaciones serán evidentes para los expertos en la técnica sin apartarse del alcance de la invención. La realización se ha elegido y descrito con el fin de explicar mejor los principios de la invención y la aplicación práctica, y para permitir que otros con experiencia ordinaria en la técnica entiendan la invención para diversas realizaciones con diversas modificaciones según se adapten al uso particular contemplado.

Como apreciará un experto en la técnica, los aspectos de la presente invención se pueden materializar como un sistema, procedimiento o producto de programa de ordenador. Por consiguiente, los aspectos de la presente invención pueden adoptar la forma de una realización de hardware completamente, una realización de software completamente (incluido firmware, software residente, microcódigo, etc.) o una realización que combina aspectos de software y hardware que en general se pueden denominar en el presente documento como "circuito", "módulo" o "sistema". Además, aspectos de la presente invención pueden adoptar la forma de un producto de programa de ordenador materializado en uno o más medios legibles por ordenador que tienen un código de programa legible por ordenador materializado en el mismo.

Se puede utilizar cualquier combinación de uno o más medios legibles por ordenador. El medio legible por ordenador puede ser un medio de señal legible por ordenador o un medio de almacenamiento legible por ordenador. Un medio de almacenamiento legible por ordenador puede ser, por ejemplo, entre otros, un sistema, aparato o dispositivo electrónico, magnético, óptico, electromagnético, infrarrojo o semiconductor, o cualquier combinación adecuada de los anteriores. Ejemplos más específicos (una lista no exhaustiva) del medio de almacenamiento legible por ordenador incluirían los siguientes: una conexión eléctrica que tiene uno o más cables, un disco flexible de ordenador portátil, un disco duro, una memoria de acceso aleatorio (RAM), una memoria de solo lectura (ROM), una memoria de solo lectura programable y borrable (EPROM o memoria flash), una fibra óptica, una memoria de solo lectura de disco compacto portátil (CD-ROM), un dispositivo de almacenamiento óptico, un dispositivo de almacenamiento magnético o cualquier combinación adecuada de los anteriores. En el contexto de este documento, un medio de almacenamiento legible por ordenador puede ser cualquier medio tangible que pueda contener o almacenar un programa para su uso por, o en conexión con, un sistema, aparato o dispositivo de ejecución de instrucciones.

Un medio de señal legible por ordenador puede incluir una señal de datos propagada con código de programa legible por ordenador materializado, por ejemplo, en banda base o como parte de una onda portadora. Dicha señal propagada puede adoptar cualquiera de una variedad de formas, que incluyen, pero no se limitan a, electromagnética, óptica o cualquier combinación adecuada de las mismas. Un medio de señal legible por ordenador puede ser cualquier medio

legible por ordenador que no sea un medio de almacenamiento legible por ordenador y que pueda comunicar, propagar o transportar un programa para su uso por, o en conexión con, un sistema, aparato o dispositivo de ejecución de instrucciones.

- 5 El código de programa materializado en un medio legible por ordenador puede transmitirse usando cualquier medio adecuado, incluidos, entre otros, inalámbrico, alámbrico, cable de fibra óptica, RF, etc., o cualquier combinación adecuada de los anteriores.

10 El código de programa de ordenador para llevar a cabo operaciones de aspectos de la presente invención se puede escribir en cualquier combinación de uno o más lenguajes de programación, incluido un lenguaje de programación orientado a objetos como Java, Smalltalk, C++ o similares y lenguajes de programación de procedimientos convencionales, tales como el lenguaje de programación "C" o lenguajes de programación similares. El código de programa puede ejecutarse completamente en el ordenador del usuario, parcialmente en el ordenador del usuario, como un paquete de software autónomo, parcialmente en el ordenador del usuario y parcialmente en un ordenador remoto o completamente en el ordenador o servidor remoto. En el último escenario, el ordenador remoto puede estar conectado al ordenador del usuario a través de cualquier tipo de red, incluida una red de área local (LAN) o una red de área extensa (WAN), o la conexión puede efectuarse a un ordenador externo (por ejemplo, a través de Internet usando un proveedor de servicios de Internet).

20 Los aspectos de la presente invención se describen anteriormente en referencia a ilustraciones de diagramas de flujo y/o diagramas esquemáticos de procedimientos, aparatos (sistemas) y productos de programas de ordenador según realizaciones de la invención. Se comprenderá que cada bloque de las ilustraciones de diagramas de flujo y/o diagramas de bloques, y las combinaciones de bloques en las ilustraciones de diagramas de flujo y/o diagramas de bloques, pueden implementarse mediante instrucciones de un programa de ordenador. Estas instrucciones de ordenador pueden proporcionarse a un procesador de un ordenador de propósito general, un ordenador de propósito especial, u otro aparato de procesamiento de datos programable para producir una máquina, de manera que las instrucciones, que se ejecutan por medio del procesador de ordenador u otro aparato de procesamiento de datos programable, crean medios para implementar las funciones/acciones especificadas en el bloque o bloques de diagramas de flujo y/o diagramas de bloques.

30 Estas instrucciones del programa de ordenador también pueden almacenarse en un medio legible por ordenador que puede dar instrucciones a un ordenador, otro aparato de procesamiento de datos programable u otros dispositivos para que funcionen de una manera particular, de manera que las instrucciones almacenadas en el medio legible por ordenador produzcan un artículo de fabricación incluidas las instrucciones que implementan la función/acción especificada en el bloque o bloques de diagramas de flujo y/o diagramas de bloques.

40 Las instrucciones del programa de ordenador también pueden cargarse en un ordenador, otro aparato de procesamiento de datos programable u otros dispositivos para hacer que se realicen una serie de etapas operativas en el ordenador, otro aparato programable u otros dispositivos para producir un procedimiento implementado por ordenador de manera que las instrucciones que se ejecutan en el ordenador u otro aparato programable proporcionan procedimientos para implementar las funciones/acciones especificadas en el bloque o bloques de diagramas de flujo y/o diagramas de bloques.

45 Como se describe anteriormente, las realizaciones pueden realizarse en forma de procedimientos y aparatos implementados por ordenador para poner en práctica esos procedimientos. En realizaciones, la invención se materializa en código de programa de ordenador ejecutado por uno o más elementos de red. Las realizaciones incluyen un producto de programa de ordenador en un medio utilizable por ordenador con lógica de código de programa de ordenador que contiene instrucciones materializadas en medios tangibles como un artículo de fabricación. Los artículos de fabricación ejemplares para el medio legible por ordenador pueden incluir discos flexibles, CD-ROM, discos duros, unidades de memoria flash de bus serie universal (USB) o cualquier otro medio de almacenamiento legible por ordenador, en el que, cuando la lógica del código de programa de ordenador es cargada y ejecutada por un ordenador, el ordenador se convierte en un aparato para poner en práctica la invención. Las realizaciones incluyen la lógica de código de programa de ordenador, por ejemplo, ya sea almacenada en un medio de almacenamiento, cargada y/o ejecutada por un ordenador, o transmitida a través de algún medio de transmisión, tal como a través de cableado o cables eléctricos, a través de fibra óptica o por medio de radiación electromagnética, en el que, cuando la lógica de código de programa de ordenador es cargada y ejecutada por un ordenador, el ordenador se convierte en un aparato para poner en práctica la invención. Cuando se implementa en un microprocesador de propósito general, los segmentos lógicos de código de programa de ordenador configuran el microprocesador para crear circuitos lógicos específicos.

60 El diagrama de flujo y los diagramas de bloques de las figuras ilustran la arquitectura, la funcionalidad y el funcionamiento de posibles implementaciones de sistemas procedimientos, y productos de programa de ordenador según diversas realizaciones de la presente invención. En este sentido, cada bloque del diagrama de flujo o los diagramas de bloques puede representar un módulo, un segmento o una porción de código, que comprende una o más instrucciones ejecutables para implementar la(s) función(es) lógica(s) especificada(s). Obsérvese que también que, en algunas implementaciones alternativas, las funciones señaladas en el bloque pueden producirse fuera del

orden indicado en las figuras. Por ejemplo, dos bloques que se muestran en sucesión pueden, de hecho, ejecutarse de manera sustancialmente simultánea, o los bloques a veces pueden ejecutarse en el orden inverso,

- 5 dependiendo de la funcionalidad implicada. También se observará que cada bloque de los diagramas de bloques y/o la ilustración de diagrama de flujo, y combinaciones de bloques en los diagramas de bloques y/o la ilustración de diagrama de flujo, pueden implementarse mediante sistemas basados en hardware de propósito especial que realizan las funciones o acciones especificadas, o combinaciones de hardware de propósito especial e instrucciones de ordenador.

REIVINDICACIONES

1. Un procedimiento implementado por ordenador para la notificación de instrumentación en tiempo de ejecución, el procedimiento que comprende:

5 ejecutar (1702) un flujo de instrucciones de un primer programa de ordenador, la ejecución mediante un procesador (106);
 determinar, en base al contenido de un campo de control de instrumentación en una palabra de estado de programa, PSW (512) del primer programa de ordenador, que debe capturarse la información de instrumentación
 10 en tiempo de ejecución de dicho flujo de instrucciones en ejecución, en el que el contenido de la PSW del primer programa de ordenador persiste durante las conmutaciones de contexto por parte del procesador;
 en respuesta a la determinación de que el campo de control de instrumentación indica que la instrumentación está habilitada, capturar (1704), mediante el procesador, la información de instrumentación en tiempo de ejecución de dicho flujo de instrucciones en ejecución;
 15 en base a dicha información de instrumentación en tiempo de ejecución capturada, crear (1706) registros de instrumentación en tiempo de ejecución;
 detectar (1708) un punto de muestra de instrumentación en tiempo de ejecución del flujo de instrucciones en ejecución en el procesador; y
 almacenar (1710) un grupo de notificación en una memoria intermedia del programa de instrumentación en tiempo
 20 de ejecución (522), el almacenamiento se basa en la detección de un punto de muestra de instrumentación en tiempo de ejecución, el almacenamiento que comprende:

25 determinar una dirección actual (706) de la memoria intermedia del programa de instrumentación en tiempo de ejecución, la determinación basada en controles de instrumentación accesibles en tiempo de ejecución mediante instrucciones; y almacenar el grupo de notificación (900) en la memoria intermedia del programa de instrumentación en tiempo de ejecución en base a una dirección de origen (702) y la dirección actual de la memoria intermedia del programa de instrumentación en tiempo de ejecución, el grupo de notificación que comprende dichos registros de instrumentación en tiempo de ejecución creados.

30 2. El procedimiento de la reivindicación 1, en el que el grupo de notificación (900) incluye un número predeterminado de registros de instrumentación en tiempo de ejecución especificados por los controles de instrumentación en tiempo de ejecución.

35 3. El procedimiento de cualquiera de las reivindicaciones 1 o 2, que comprende además ejecutar un programa en un estado supervisor para establecer una cualquiera de las direcciones de origen (702) de la memoria intermedia del programa de instrumentación en tiempo de ejecución (522) o la dirección actual de la memoria intermedia del programa de instrumentación en tiempo de ejecución en los controles de instrumentación en tiempo de ejecución.

40 4. El procedimiento de la reivindicación 3, en el que el programa incluye una instrucción de los controles de instrumentación en tiempo de ejecución de carga, LRIC, y una instrucción de los controles de instrumentación en tiempo de ejecución modificados, MRIC, en el que una instrucción LRIC inicializa los controles de instrumentación en tiempo de ejecución que rigen la instrumentación en tiempo de ejecución, y una instrucción MRIC modifica todos o un subconjunto de los controles de instrumentación en tiempo de ejecución establecidos por la LRIC.

45 5. El procedimiento de cualquiera de las reivindicaciones 1 a 4, que comprende además ejecutar un programa en un estado de problema para establecer la dirección de origen (702) de la memoria intermedia del programa de instrumentación en tiempo de ejecución (522) en los controles de instrumentación en tiempo de ejecución, en el que el programa incluye una instrucción de los controles de instrumentación en tiempo de ejecución modificados, MRIC, en la que una instrucción LRIC inicializa controles de instrumentación en tiempo de ejecución que rigen la instrumentación en tiempo de ejecución, y una instrucción MRIC modifica todos o un subconjunto de los controles de instrumentación en tiempo de ejecución establecidos por la LRIC.

50 6. El procedimiento de cualquiera de las reivindicaciones 1 a 5, en el que los controles de instrumentación en tiempo de ejecución especifican un número de registros en cada grupo de notificación (900) en la memoria intermedia del programa de instrumentación en tiempo de ejecución (522).

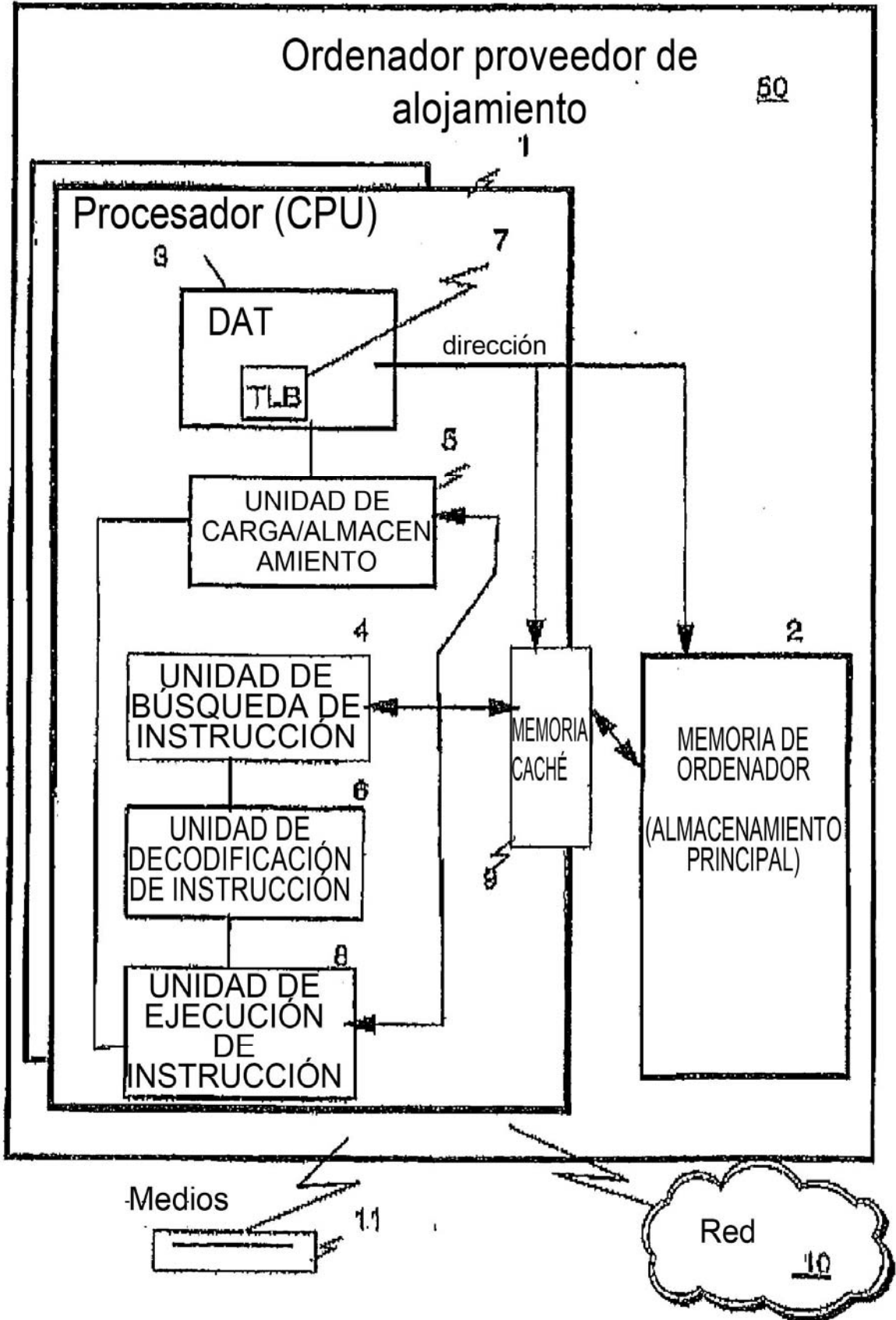
55 7. El procedimiento de cualquiera de las reivindicaciones 1 a 6, en el que la memoria intermedia del programa de instrumentación en tiempo de ejecución (522) está ubicada en un espacio de direcciones que es accesible por un programa de aplicación.

60 8. Un producto de programa de ordenador para la notificación de instrumentación en tiempo de ejecución, el producto de programa de ordenador que comprende:

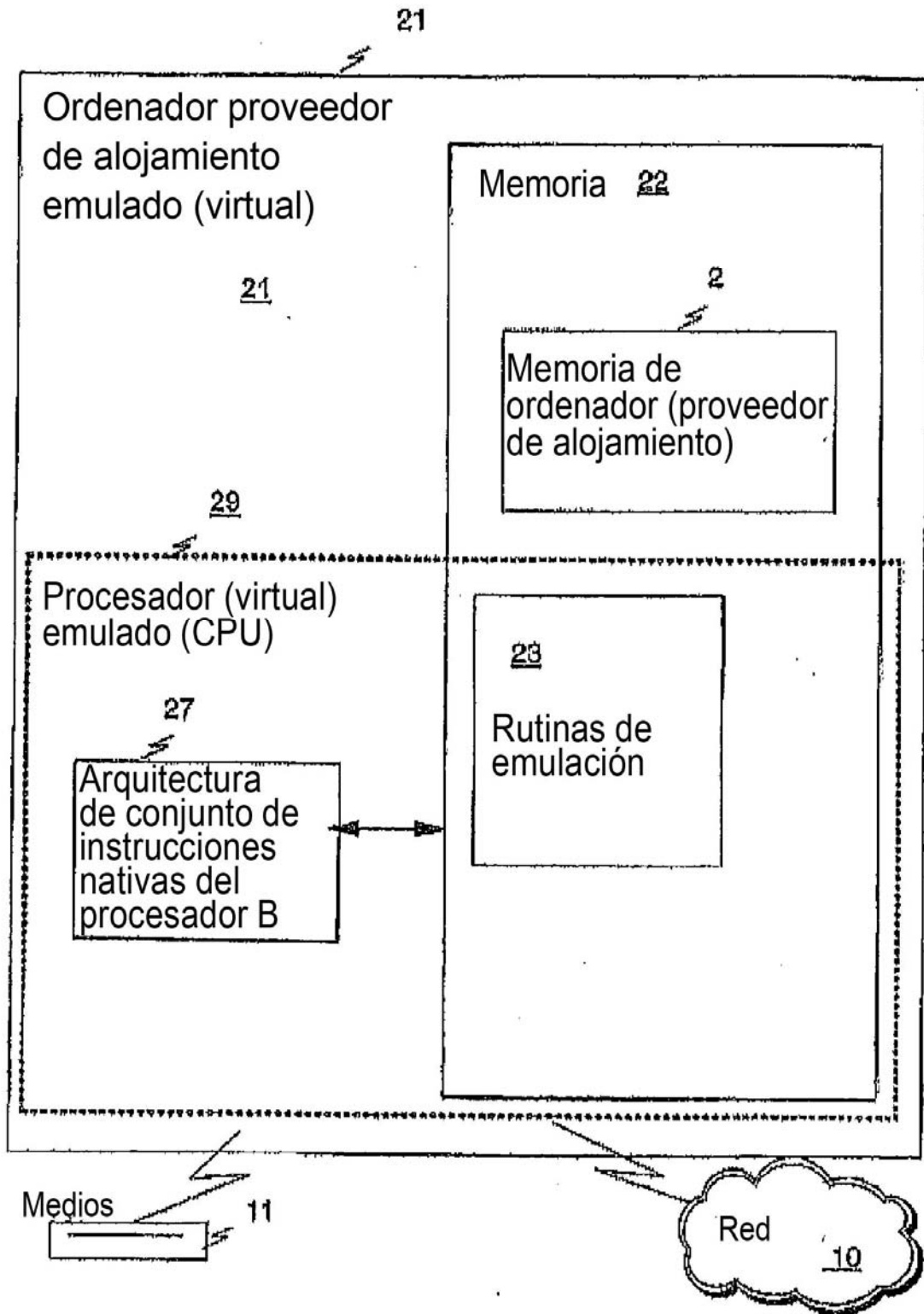
65 un medio de almacenamiento legible por ordenador legible por un circuito de procesamiento y que almacena instrucciones para su ejecución por el circuito de procesamiento para realizar un procedimiento según cualquiera de las reivindicaciones 1 a 7.

- 5 9. Un programa de ordenador almacenado en un medio legible por ordenador y que puede cargarse en la memoria interna de un ordenador digital, que comprende porciones de código de software, cuando dicho programa se ejecuta en un ordenador, para realizar el procedimiento de cualquiera de las reivindicaciones 1 a 7.
10. Un sistema para la notificación de instrumentación en tiempo de ejecución, el sistema que comprende: un procesador, el sistema configurado para realizar un procedimiento que comprende:
- 10 ejecutar (1702) un flujo de instrucciones de un programa de ordenador, la ejecución mediante un procesador (106); determinar, en base al contenido de un campo de control de instrumentación en una palabra de estado de programa, PSW (512) del primer programa de ordenador, que debe capturarse la información de instrumentación en tiempo de ejecución de dicho flujo de instrucciones en ejecución, en el que el contenido de la PSW del primer programa de ordenador persiste durante las conmutaciones de contexto por parte del procesador;
- 15 en respuesta a la determinación de que el campo de control de instrumentación indica que la instrumentación está habilitada, capturar (1704), mediante el procesador, la información de instrumentación en tiempo de ejecución de dicho flujo de instrucciones en ejecución;
- en base a dicha información de instrumentación en tiempo de ejecución capturada, crear (1706) registros de instrumentación en tiempo de ejecución;
- 20 detectar (1708) un punto de muestra de instrumentación en tiempo de ejecución del flujo de instrucciones en ejecución en el procesador; y almacenar (1710) un grupo de notificación en una memoria intermedia del programa de instrumentación en tiempo de ejecución (522), el almacenamiento se basa en la detección de un punto de muestra de instrumentación en tiempo de ejecución, el almacenamiento que comprende:
- 25 determinar una dirección actual (706) de la memoria intermedia del programa de instrumentación en tiempo de ejecución, la determinación basada en controles de instrumentación accesibles en tiempo de ejecución mediante instrucciones; y almacenar el grupo de notificación (900) en la memoria intermedia del programa de instrumentación en tiempo de ejecución en base a una dirección de origen (702) y la dirección actual de la memoria intermedia del programa de instrumentación en tiempo de ejecución, el grupo de notificación que
- 30 comprende dichos registros de instrumentación en tiempo de ejecución creados.
11. El sistema de la reivindicación 10, en el que el grupo de notificación (900) incluye un número predeterminado de registros de instrumentación en tiempo de ejecución especificados por los controles de instrumentación en tiempo de ejecución.
- 35 12. El sistema de cualquiera de las reivindicaciones 10 o 11, que comprende además ejecutar un programa en un estado supervisor para establecer una cualquiera de las direcciones de origen (702) de la memoria intermedia del programa de instrumentación en tiempo de ejecución (522) o la dirección actual de la memoria intermedia del programa de instrumentación en tiempo de ejecución en los controles de instrumentación en tiempo de ejecución.
- 40 13. El sistema de la reivindicación 12, en el que el programa incluye una instrucción de los controles de instrumentación en tiempo de ejecución de carga, LRIC, y una instrucción de los controles de instrumentación en tiempo de ejecución modificados, MRIC, en el que una instrucción LRIC inicializa los controles de instrumentación en tiempo de ejecución que rigen la instrumentación en tiempo de ejecución, y una instrucción MRIC modifica todos o un subconjunto de los controles de instrumentación en tiempo de ejecución establecidos por la LRIC.
- 45 14. El sistema de cualquiera de las reivindicaciones 10 a 13, que comprende además ejecutar un programa en un estado de problema para establecer la dirección de origen (702) de la memoria intermedia del programa de instrumentación en tiempo de ejecución (522) en los controles de instrumentación en tiempo de ejecución, en el que el programa incluye una instrucción de los controles de instrumentación en tiempo de ejecución modificados, MRIC, en la que una instrucción LRIC inicializa controles de instrumentación en tiempo de ejecución que rigen la instrumentación en tiempo de ejecución, y una instrucción MRIC modifica todos o un subconjunto de los controles de instrumentación en tiempo de ejecución establecidos por la LRIC.
- 50 15. El sistema de cualquiera de las reivindicaciones 10 a 14, en el que los controles de instrumentación en tiempo de ejecución especifican un número de registros en cada grupo de notificación (900) en la memoria intermedia del programa de instrumentación en tiempo de ejecución (522).
- 55

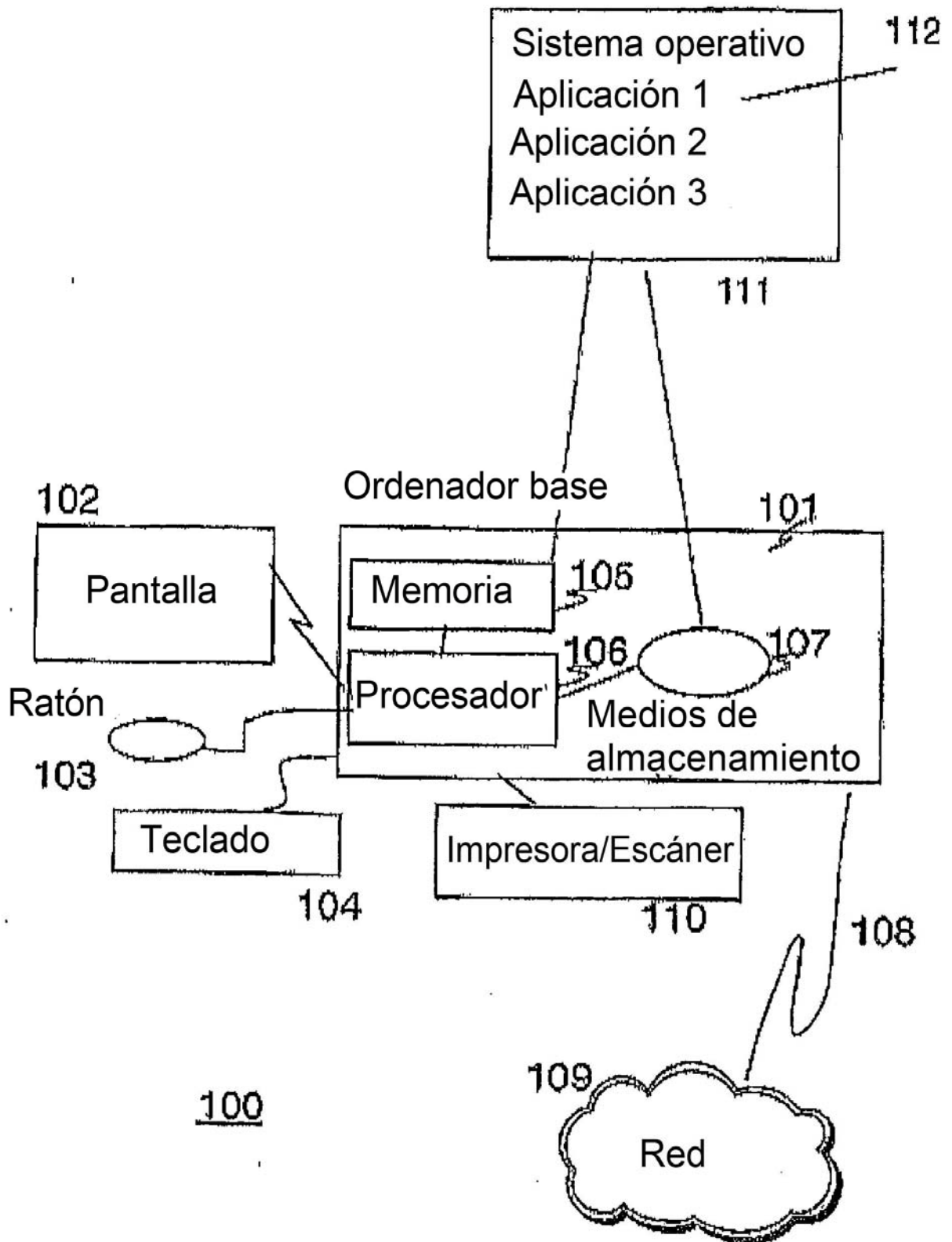
[Fig. 1A]



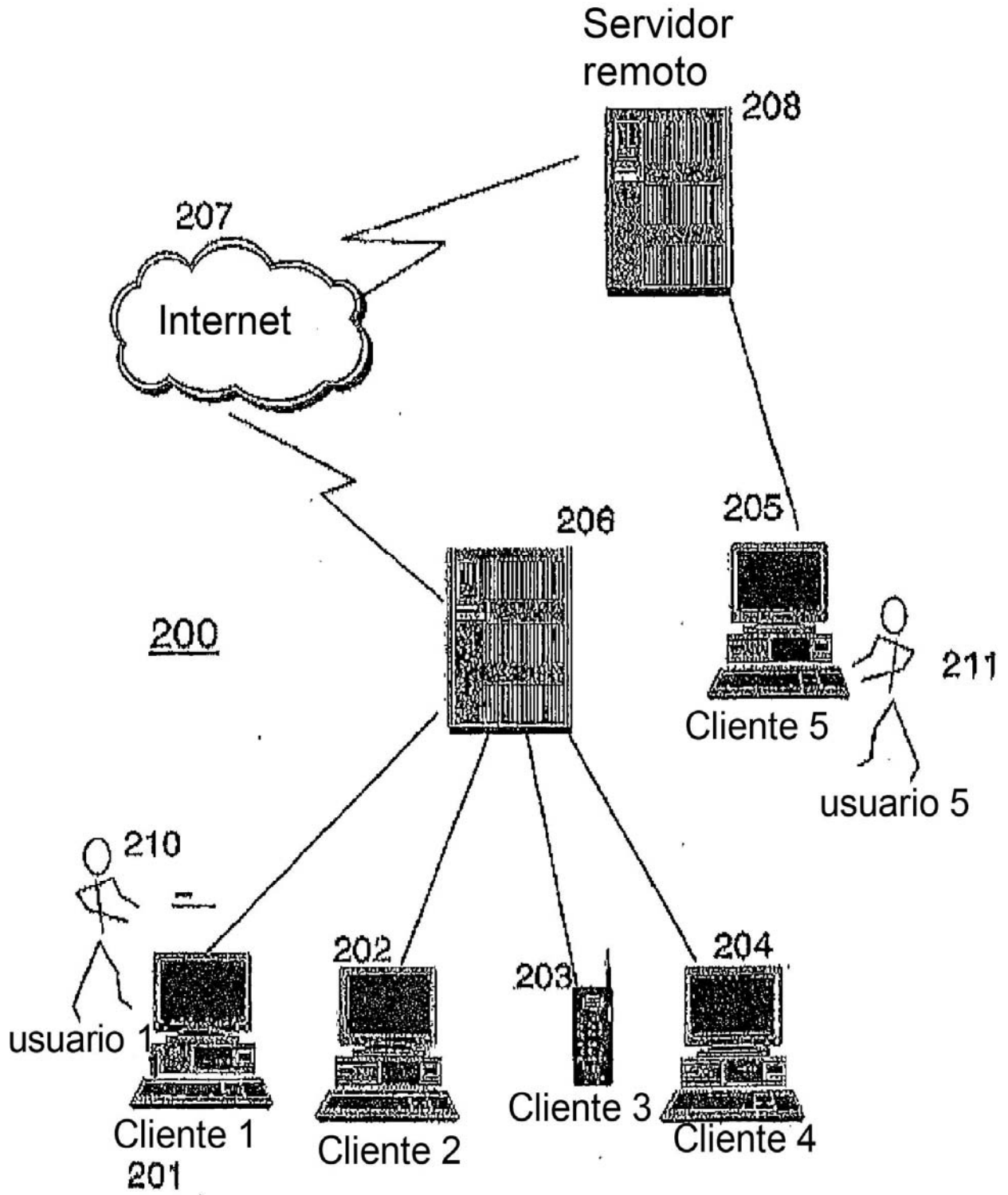
[Fig. 1B]



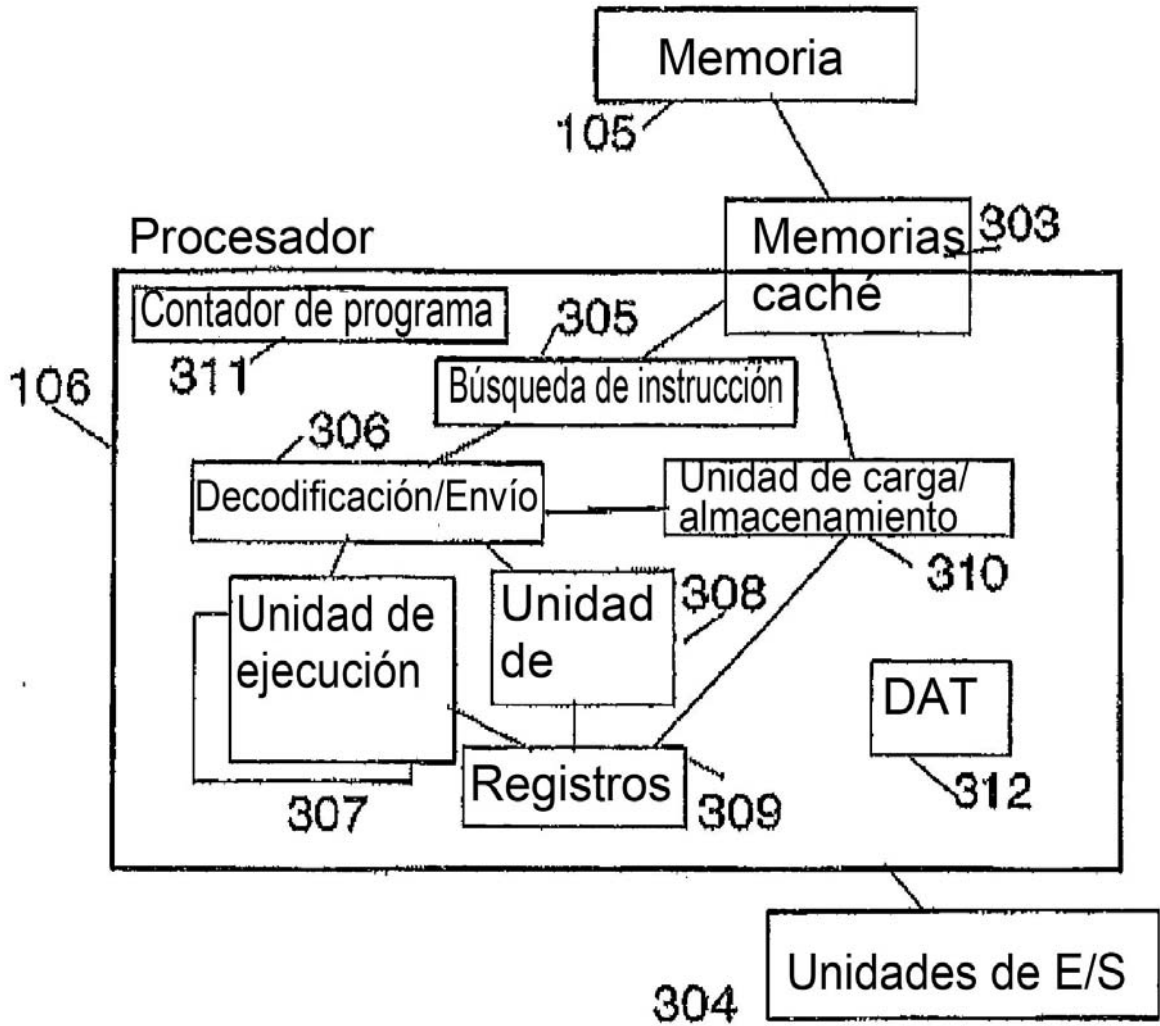
[Fig. 1C]



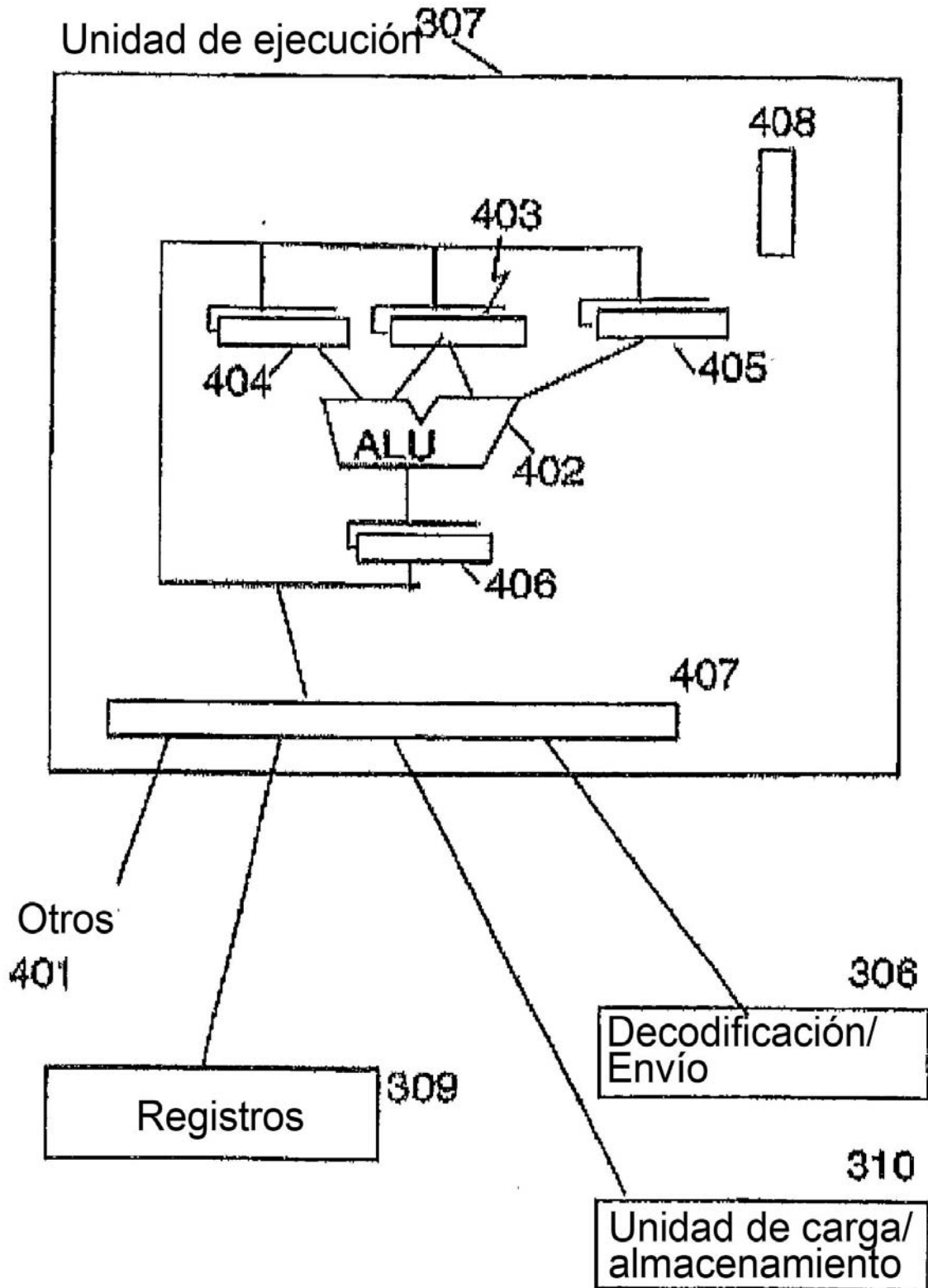
[Fig. 2]



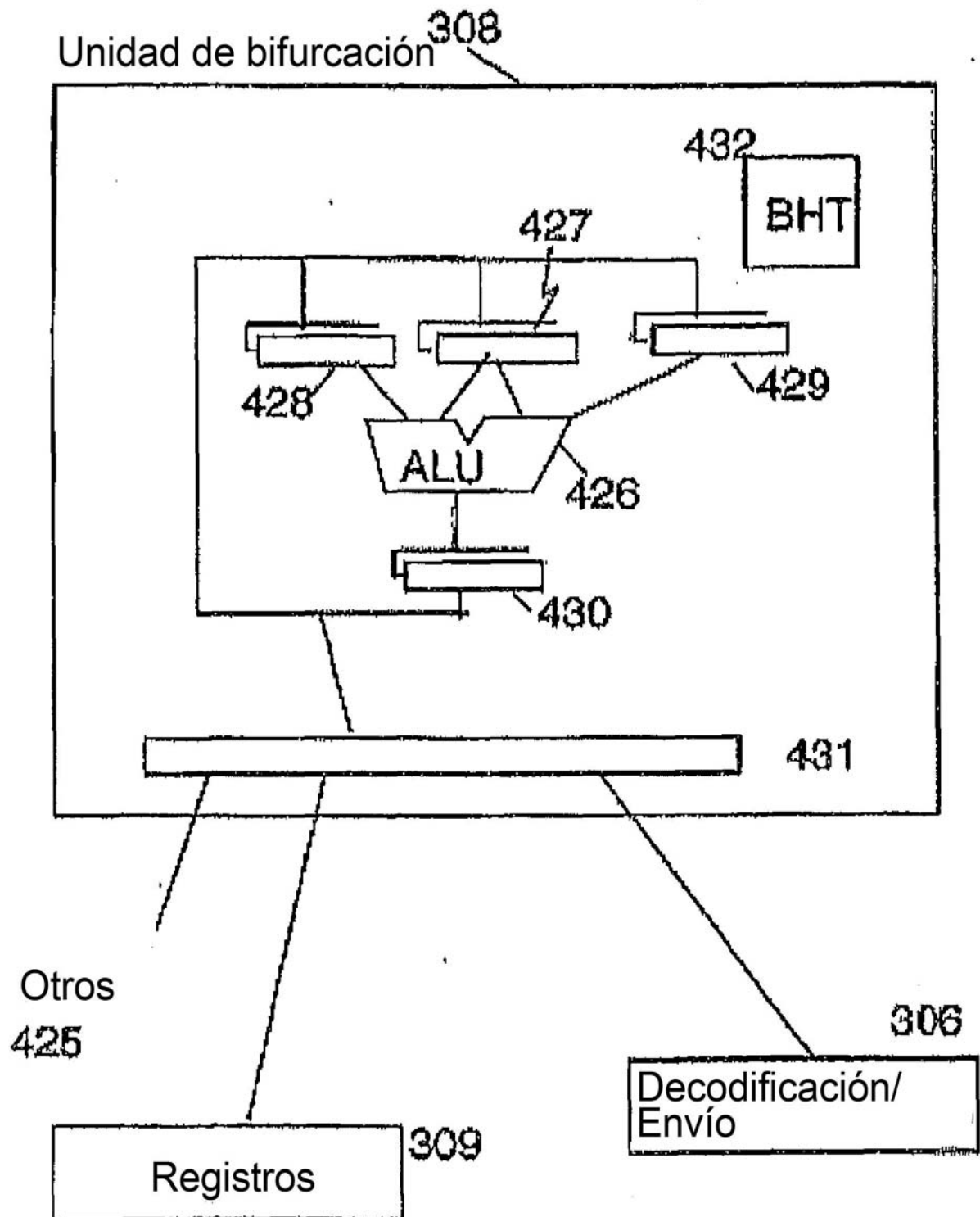
[Fig. 3]



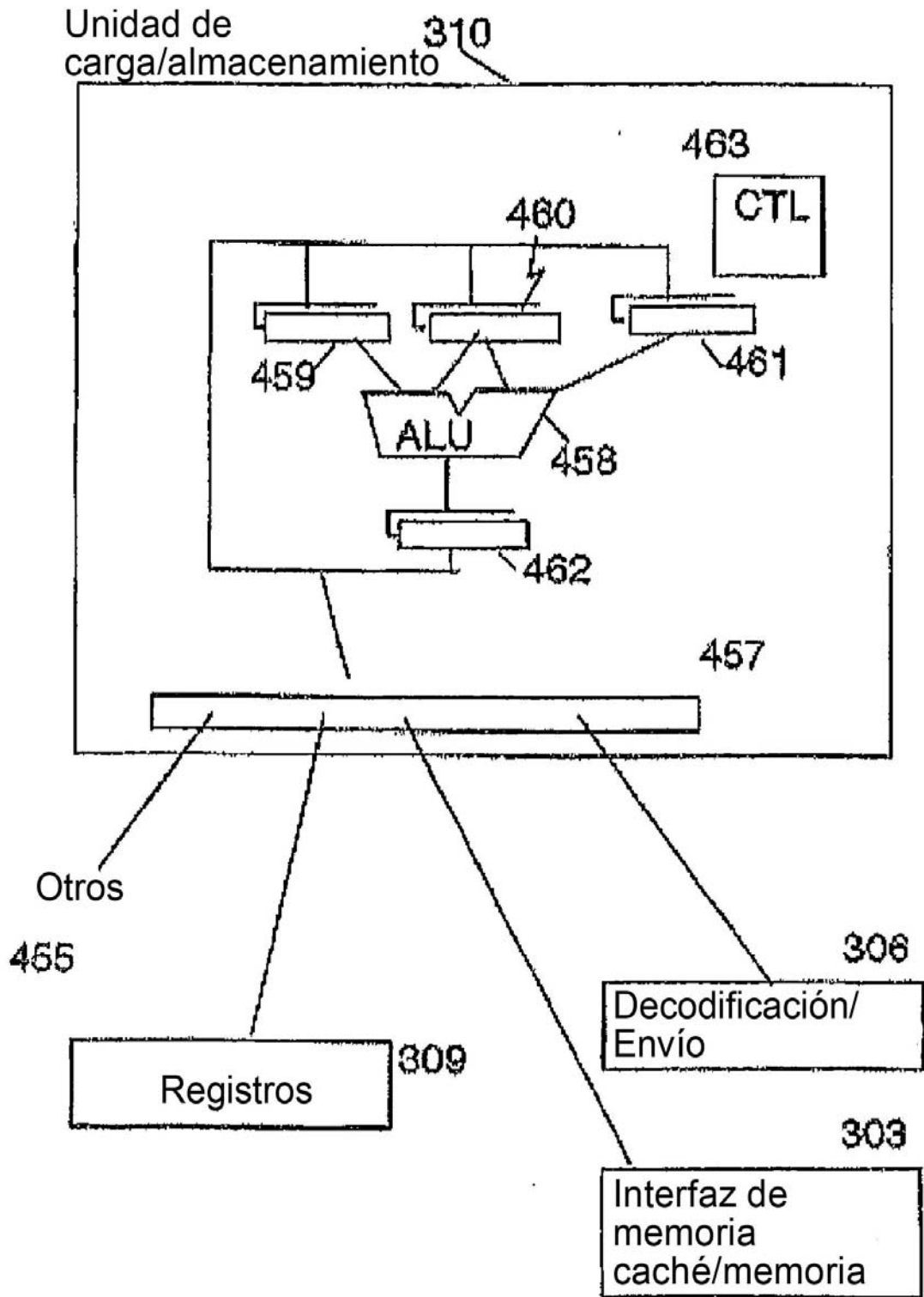
[Fig. 4A]



[Fig. 4B]

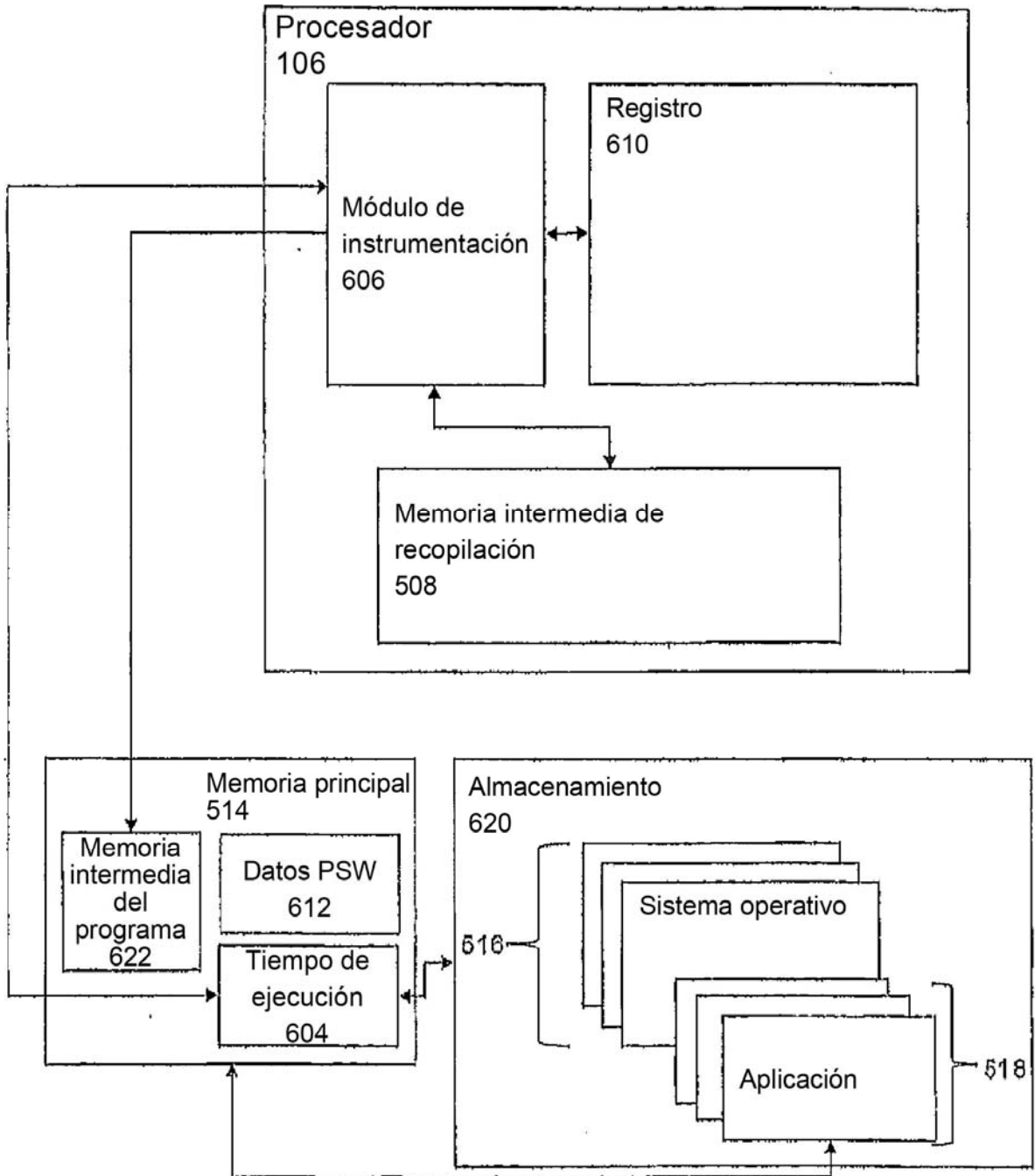


[Fig. 4C]



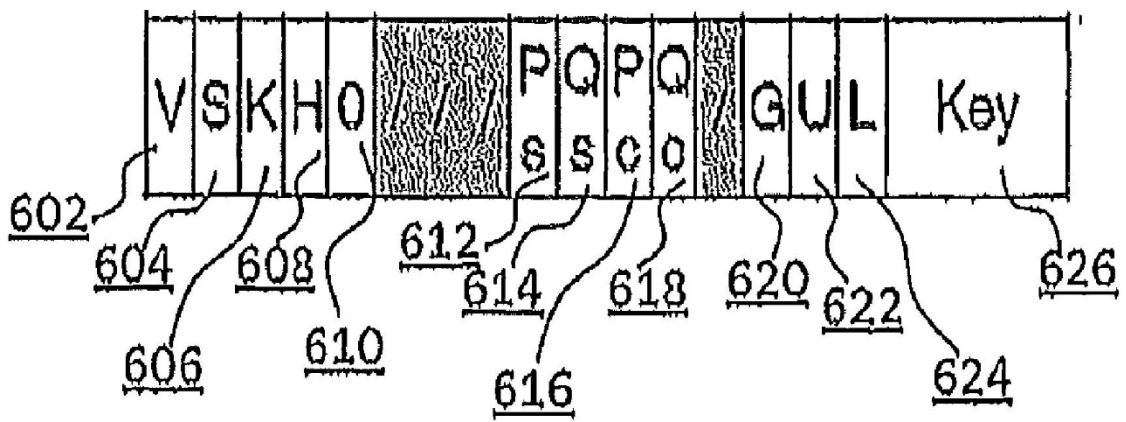
[Fig. 5]

500

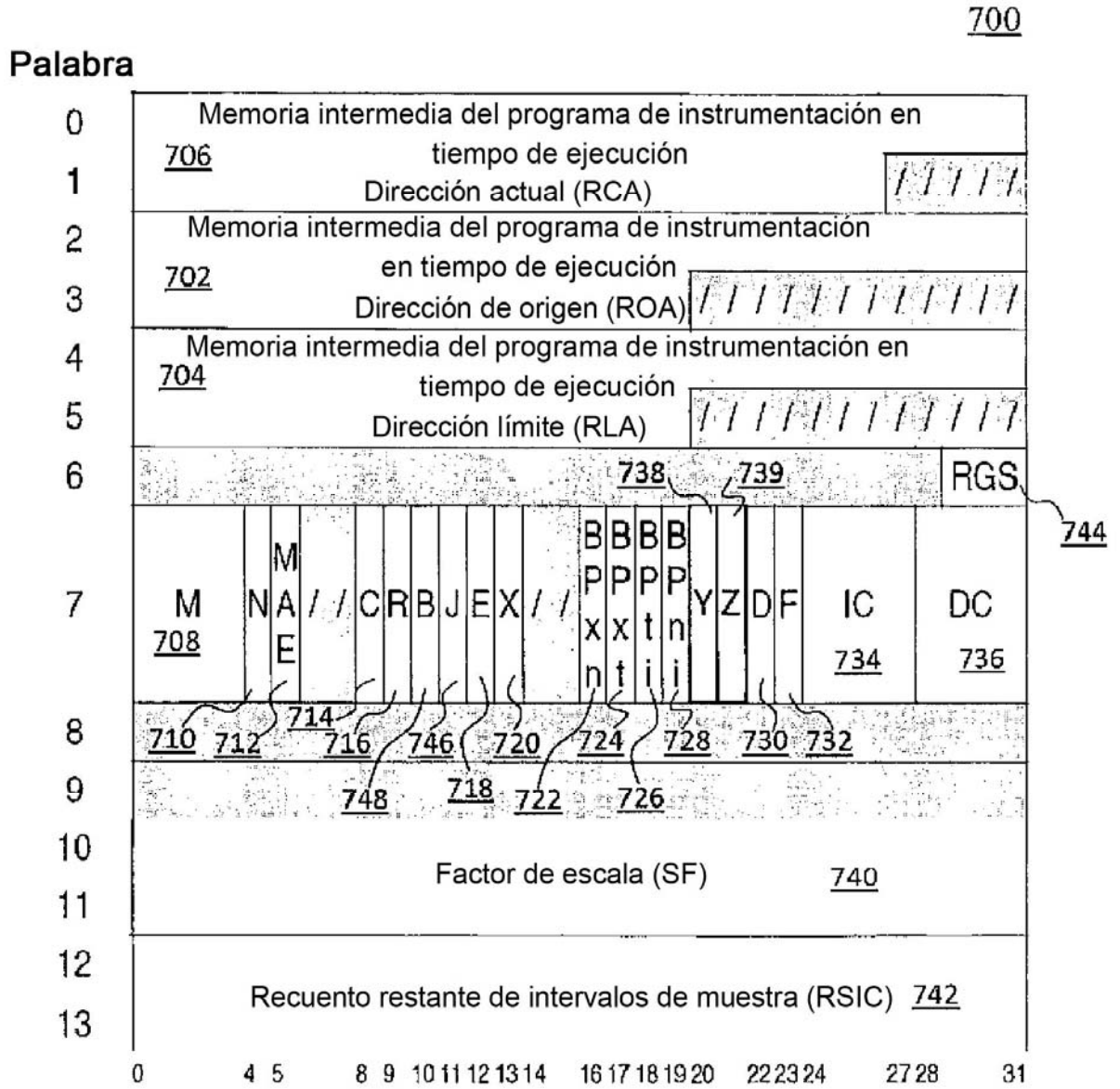


[Fig. 6]

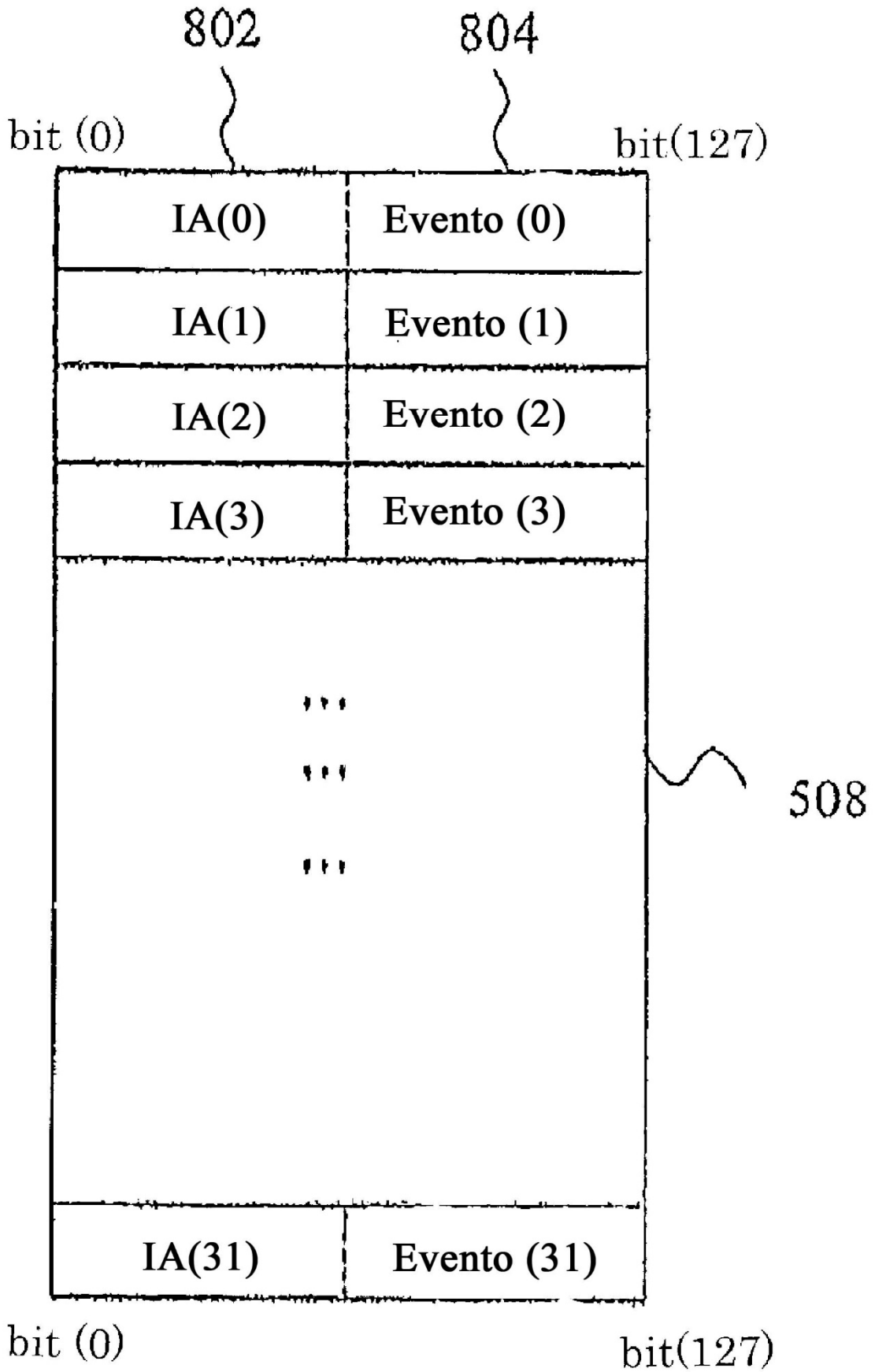
600



[Fig. 7]



[Fig. 8]

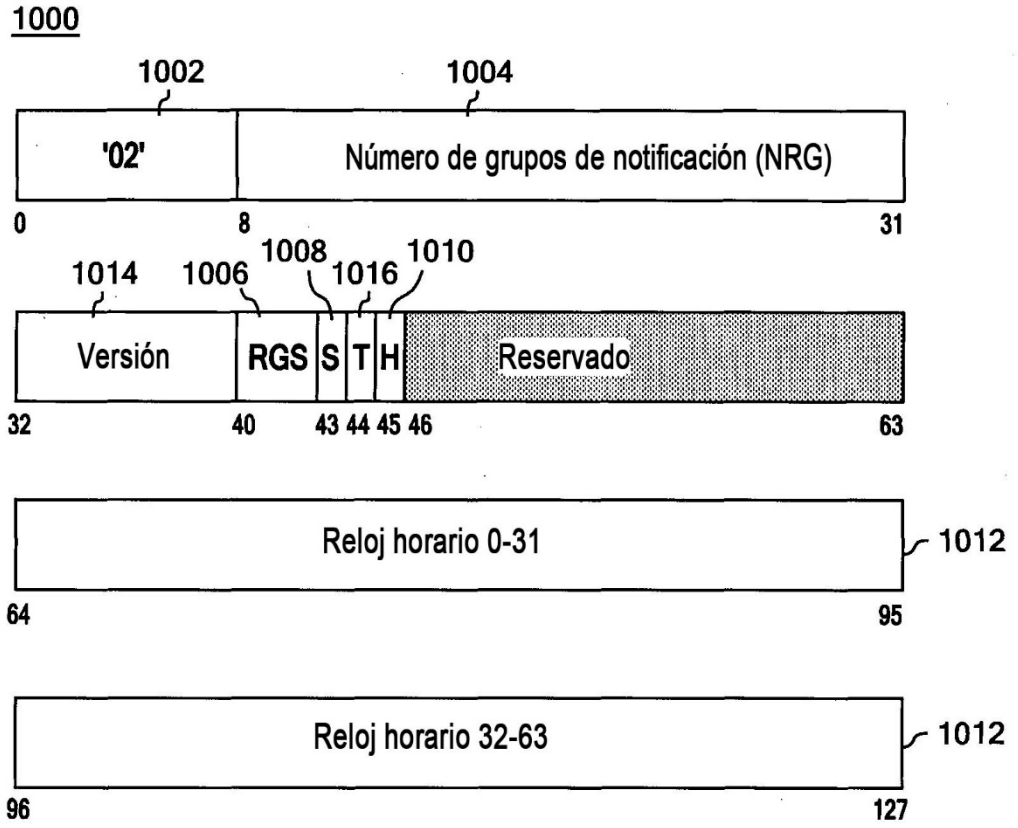


[Fig. 9]

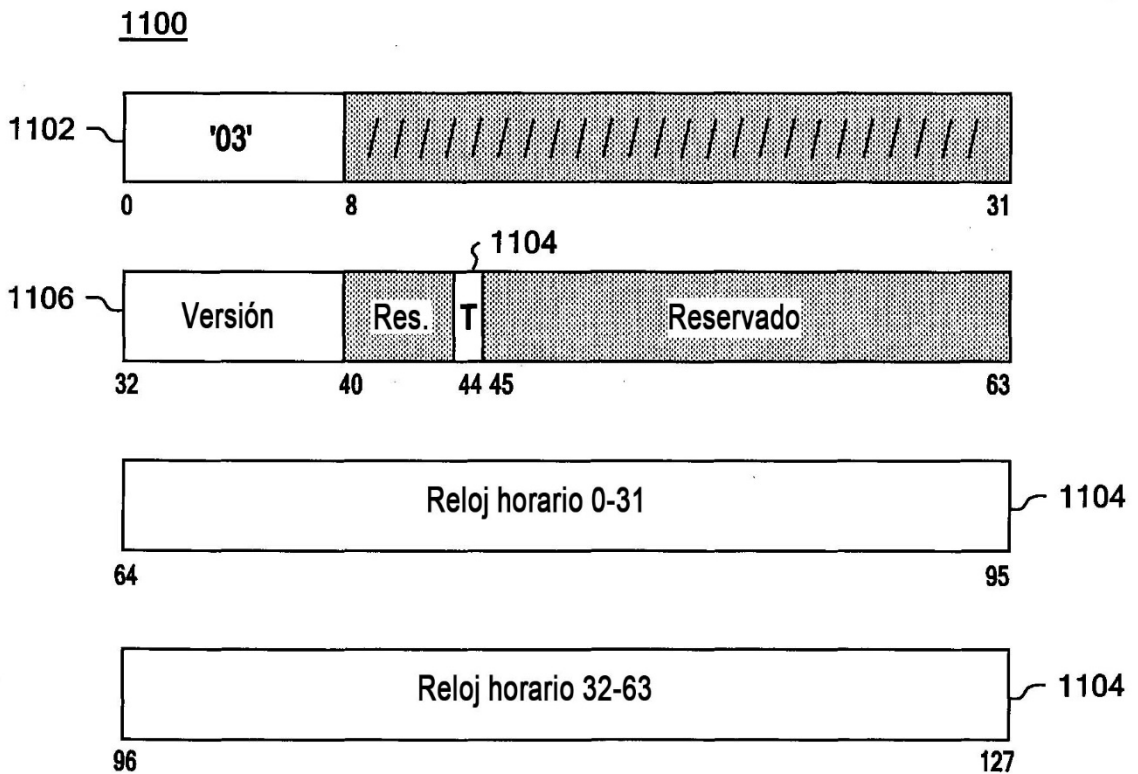
900

Núm. Registro	Tipo(s) de registro	Sección
0	Inicio, Marca de tiempo	Encabezamiento 902
1	Emitir, Abortar TX, Llamar, Retorno, Bifurcación, Relleno	Cuerpo 904
2		
3		
$R_{RG}-R_{NO} (=4)$		
$R_{RG}-R_{NO}+1 (=5)$	Extra, dependiente del modelo	Extra 906
$R_{RG}-2 (=6)$		
$R_{RG}-1$	Instrucción	Pie 908

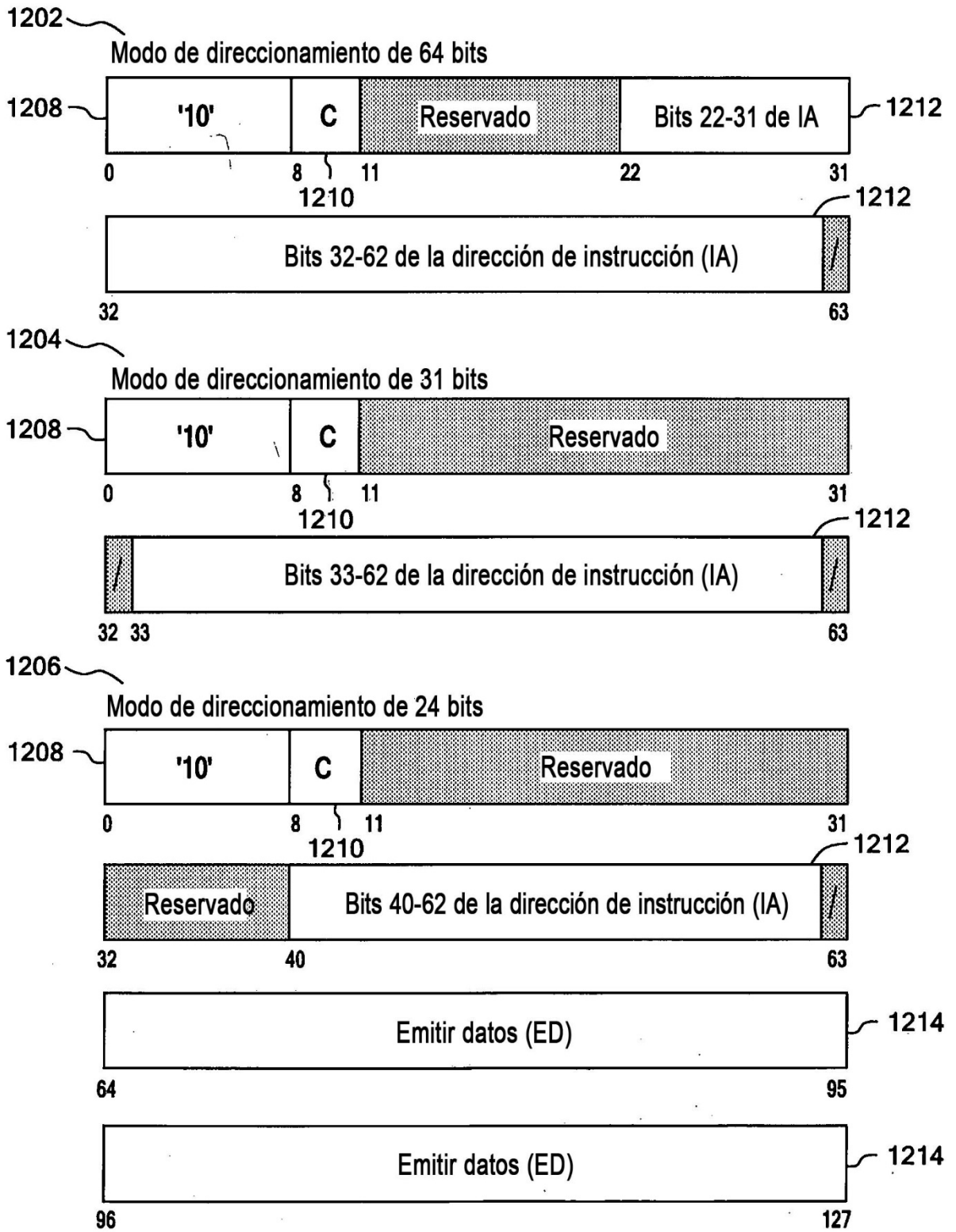
[Fig. 10]



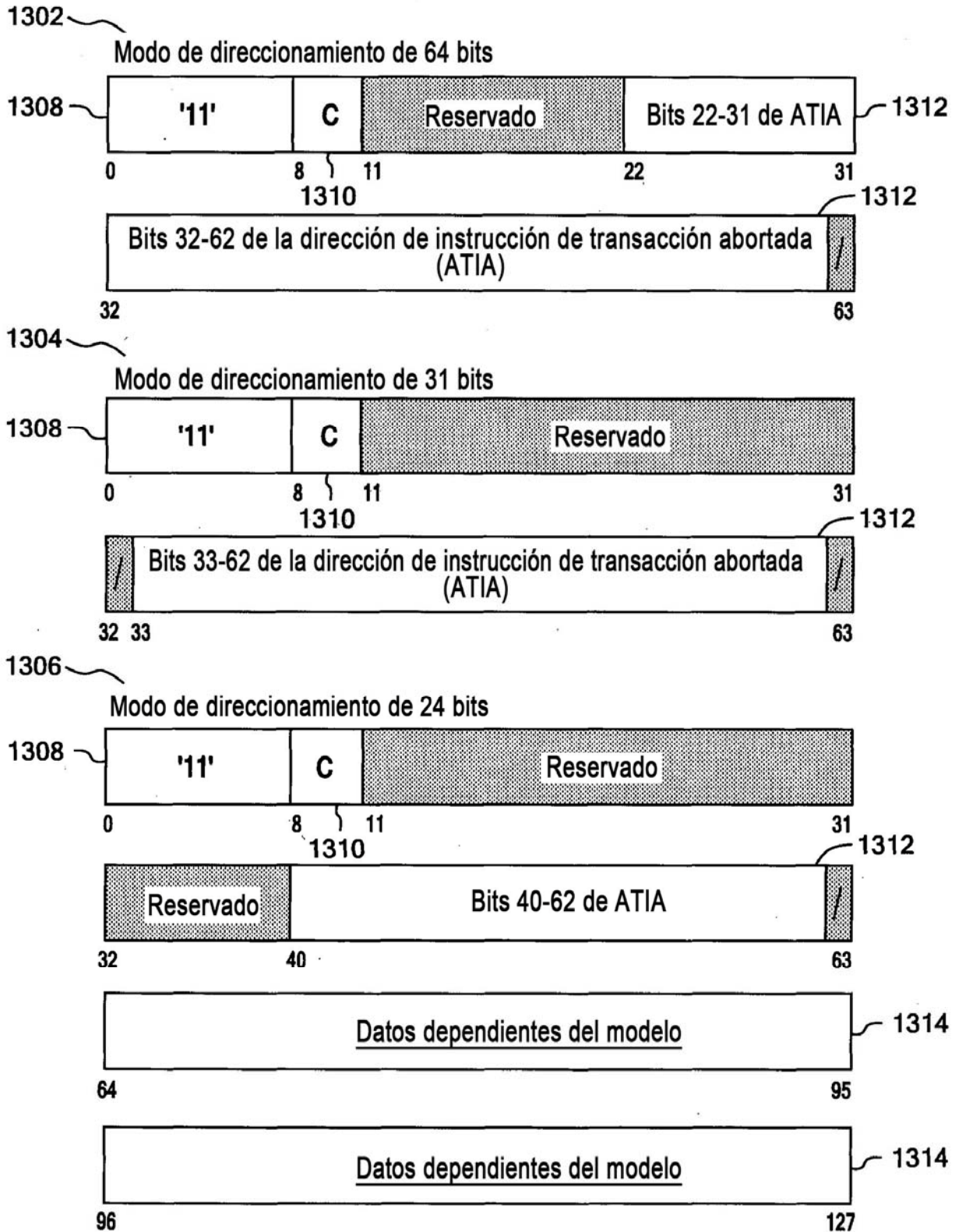
[Fig. 11]



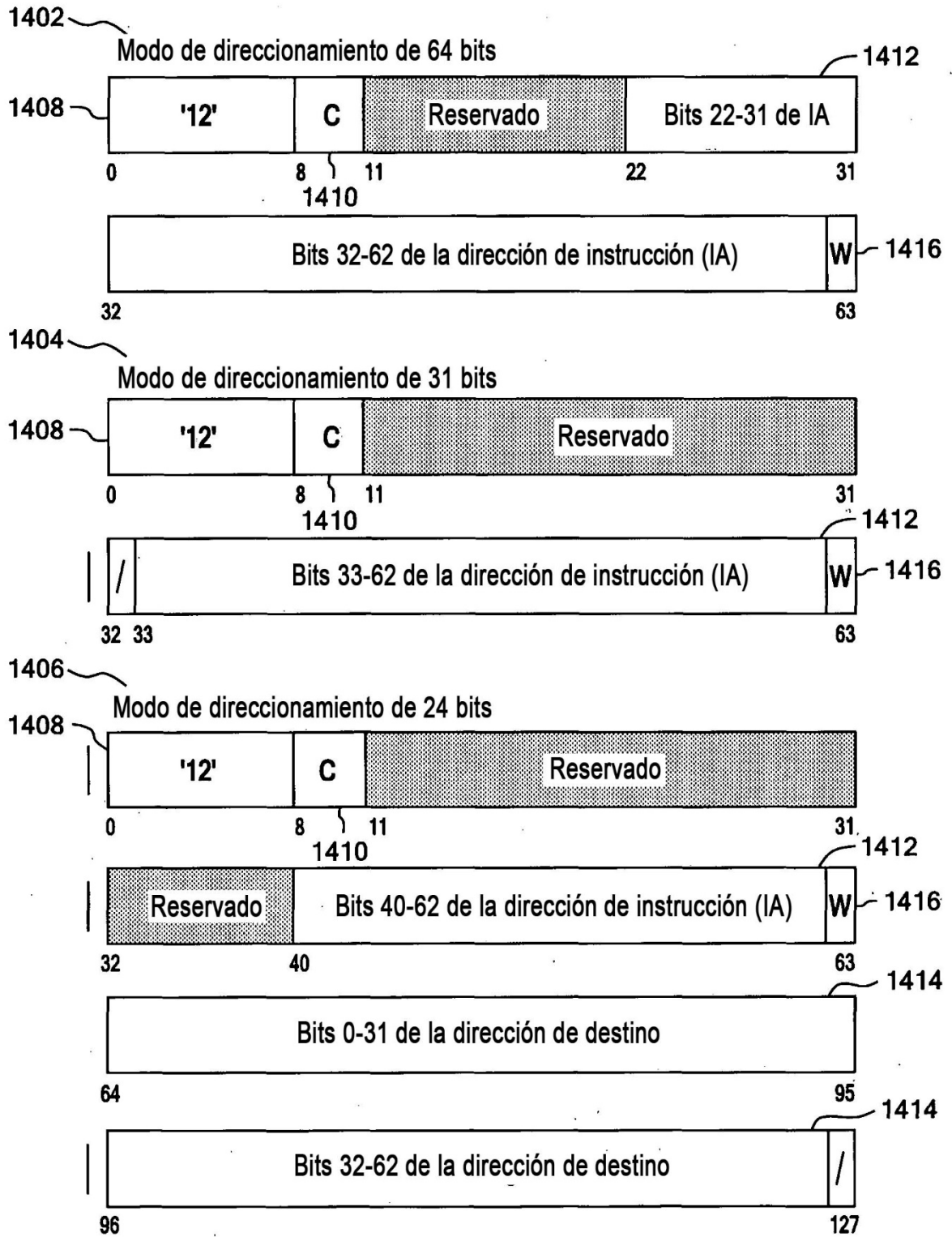
[Fig. 12]



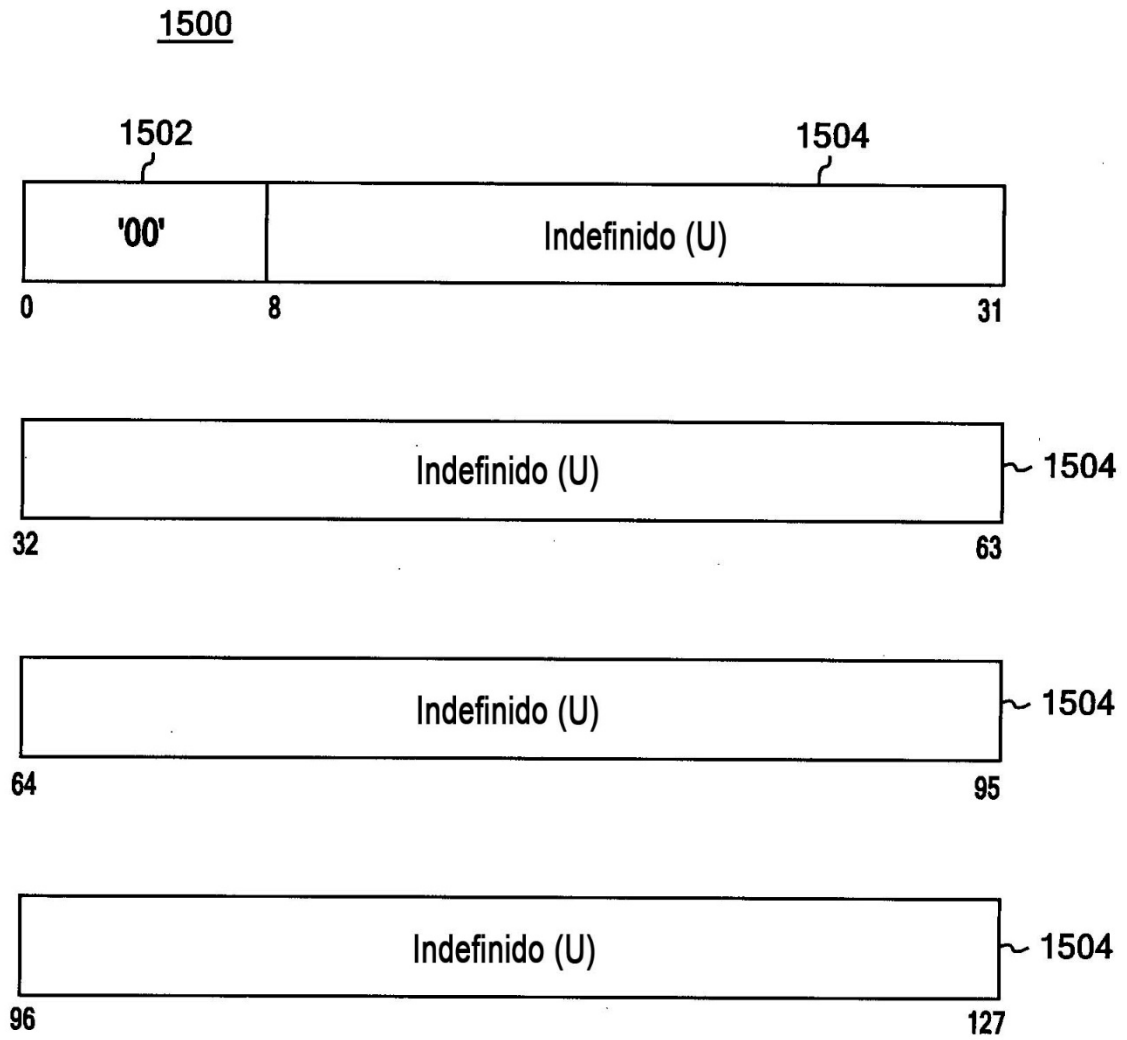
[Fig. 13]



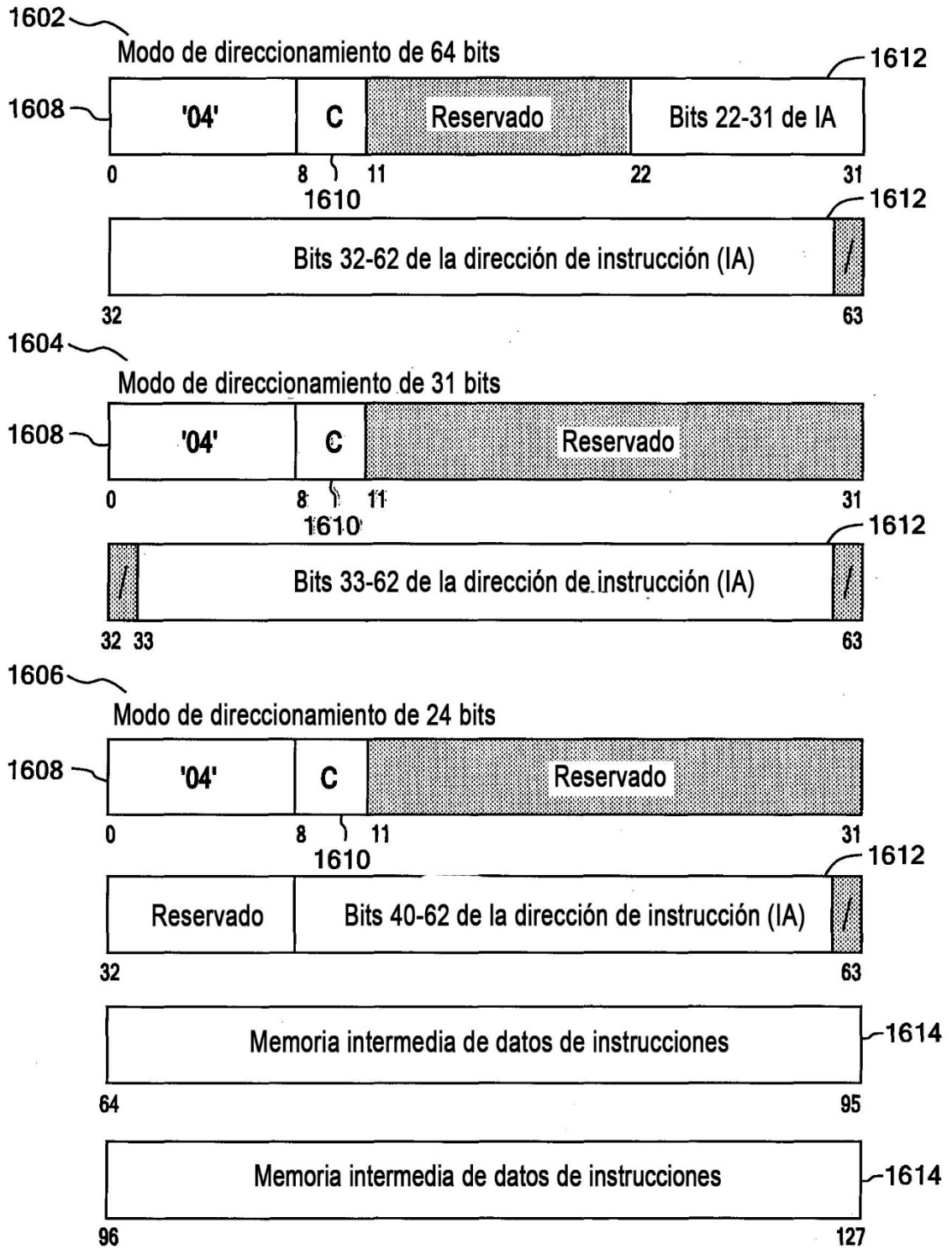
[Fig. 14]



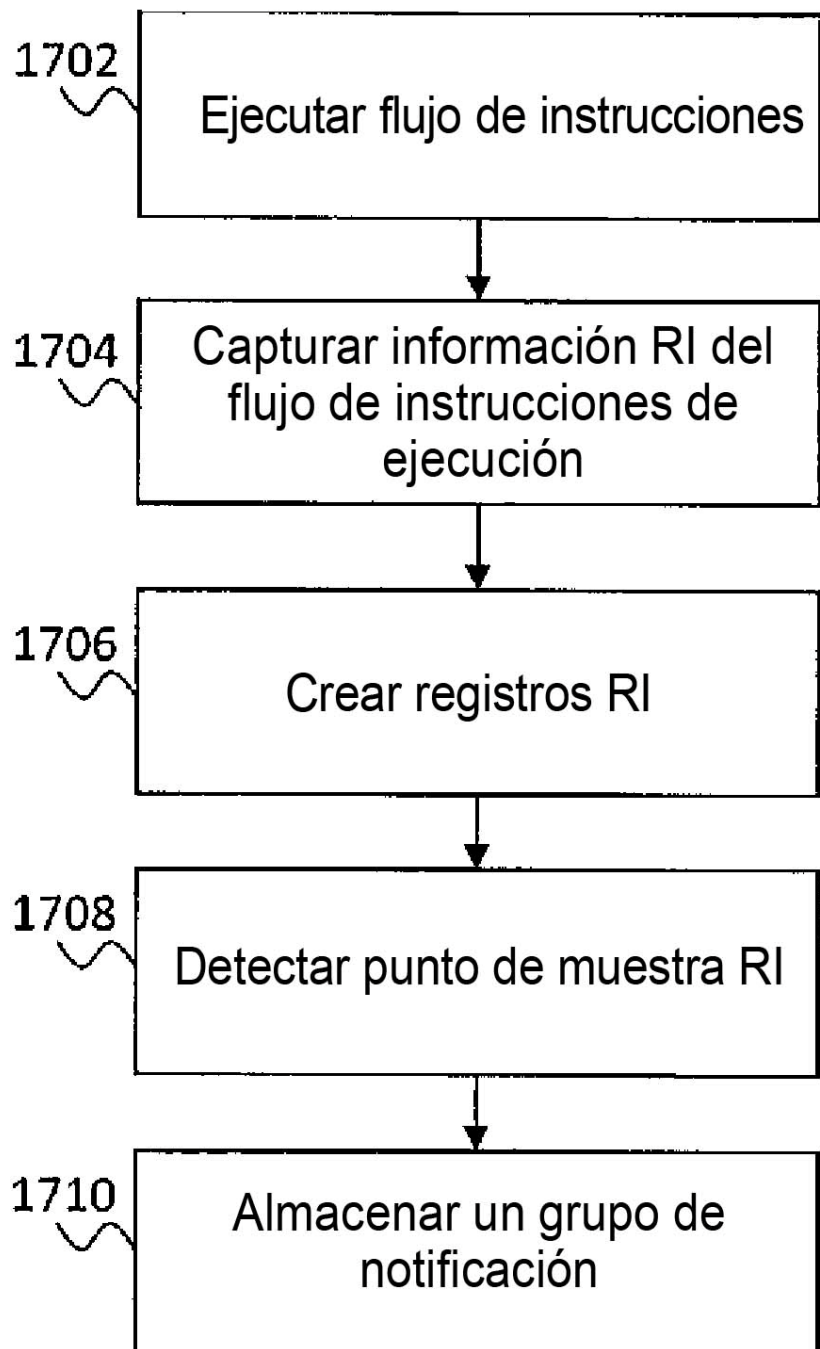
[Fig. 15]



[Fig. 16]



[Fig. 17]



[Fig. 18]

