

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **3 021 260**

51 Int. Cl.:

G06F 9/30

(2008.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **18.02.2022 PCT/EP2022/054049**

87 Fecha y número de publicación internacional: **01.09.2022 WO22179942**

96 Fecha de presentación y número de la solicitud europea: **18.02.2022 E 22710975 (8)**

97 Fecha y número de publicación de la concesión europea: **26.03.2025 EP 4298507**

54 Título: **Instrucción de escalado decimal y conversión y división a coma flotante hexadecimal**

30 Prioridad:

26.02.2021 US 202117186302

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

26.05.2025

73 Titular/es:

**INTERNATIONAL BUSINESS MACHINES CORPORATION (100.00%)
New Orchard Road
Armonk, New York 10504, US**

72 Inventor/es:

**SCHWARZ, ERIC;
LEBER, PETRA;
SCHELM, KERSTIN;
MUELLER, SILVIA;
COPELAND, REID;
GUO, XIN y
LICHTENAU, CEDRIC**

74 Agente/Representante:

ELZABURU, S.L.P

ES 3 021 260 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

Instrucción de escalado decimal y conversión y división a coma flotante hexadecimal

5 Antecedentes

Uno o más aspectos se refieren, en general, a facilitar el procesamiento dentro de un entorno informático y, en particular, a mejorar dicho procesamiento.

10 Las aplicaciones que se ejecutan en un entorno informático proporcionan muchas operaciones usadas por numerosos tipos de tecnologías, que incluyen, pero no se limitan a, la ingeniería, la fabricación, las tecnologías médicas, las tecnologías del automóvil, el procesamiento informático, etc. Estas aplicaciones, escritas en un lenguaje de programación, tal como COBOL, suelen realizar cálculos complejos al realizar las operaciones. Los cálculos incluyen, por ejemplo, funciones de potencia y/o exponenciación, que a menudo requieren la conversión de datos de un formato (por ejemplo, decimal codificado en binario) a otro formato (por ejemplo, coma flotante hexadecimal) y viceversa.

20 Para que una aplicación realice la conversión de un formato a otro formato, se ejecutan varias etapas. Por ejemplo, para convertir un decimal codificado en binario en coma flotante hexadecimal, una aplicación incluye etapas para convertir un número decimal codificado en binario a un número entero y, a continuación, el número entero se convierte en coma flotante hexadecimal. Además, cada una de esas etapas puede incluir subetapas. Esto lleva mucho tiempo, afecta al rendimiento del entorno informático y afecta a la disponibilidad de los recursos informáticos. El documento US2007022152 describe un procesador que ejecuta una instrucción que convierte un valor de coma flotante hexadecimal en coma flotante binaria. El documento US6591361 enseña una instrucción que escala un primer número según un factor de escala y, posteriormente, convierte el número escalado en un tipo diferente.

Compendio

30 Se superan los inconvenientes de la técnica anterior y se proporcionan ventajas adicionales a través de la provisión de un producto de programa informático para facilitar el procesamiento en un entorno informático. El producto de programa informático incluye uno o más medios de almacenamiento legibles por ordenador e instrucciones de programa almacenadas conjuntamente en el uno o más medios de almacenamiento legibles por ordenador, para realizar un procedimiento. El procedimiento incluye ejecutar una instrucción para realizar operaciones de escalado, conversión y división. La ejecución de la instrucción incluye escalar un valor de entrada de un formato para proporcionar un resultado escalado, convertir el resultado escalado de un formato para proporcionar un resultado convertido en otro formato, dividir el resultado convertido en múltiples partes y colocar una o más partes de las múltiples partes en una ubicación seleccionada.

40 Al usar una única instrucción para realizar las operaciones de escalado, conversión y división, se mejora el rendimiento y se reduce la utilización de recursos. Al usar una única instrucción diseñada para realizar las operaciones de escalado, conversión y división, se pueden realizar ciertas tareas, tales como las operaciones de escalado, conversión y división, de manera mucho más eficiente que cuando se usa un paradigma de software. Las operaciones de escalado, conversión y división se realizan mucho más rápido, lo que reduce el tiempo de ejecución y mejora el rendimiento del procesador y/o del sistema en general.

45 En un ejemplo, un formato es un formato decimal y el otro formato es un formato de coma flotante hexadecimal. Como ejemplo, el formato decimal es un formato decimal codificado en binario. El formato de coma flotante hexadecimal proporciona una precisión mejorada, lo que es beneficioso para varias tecnologías, ya que mejora la precisión y/o el rendimiento.

50 En un ejemplo, el escalado incluye determinar un factor de escalado y usar el factor de escalado para escalar el valor de entrada para proporcionar el resultado escalado. La determinación del factor de escalado incluye, por ejemplo, obtener un valor de escalado usando un operando de la instrucción y usar el valor de escalado para determinar el factor de escalado. El uso del factor de escalado incluye multiplicar el valor de entrada por el factor de escalado para obtener el resultado escalado.

55 El escalado facilita la conversión de un valor de un formato (por ejemplo, decimal) a otro formato (por ejemplo, coma flotante hexadecimal) aislando ciertos dígitos de un número para indicar una ubicación seleccionada en el número para truncar o redondear, como ejemplos.

60 En un ejemplo, la división incluye normalizar el resultado convertido para obtener un primer resultado normalizado y truncar el primer resultado normalizado para obtener un resultado en un formato corto del otro formato. En un ejemplo, la división incluye además restar el resultado del resultado convertido para proporcionar una diferencia, normalizar la diferencia para proporcionar un segundo resultado normalizado y truncar el segundo resultado normalizado para obtener otro resultado en un formato largo del otro formato.

65

Al dividir el número de coma flotante hexadecimal en partes más pequeñas, las partes más pequeñas se pueden usar de forma independiente, y/o las partes más pequeñas se pueden sumar para proporcionar un resultado de mayor precisión.

5 Como ejemplo, la colocación incluye colocar un resultado en una parte de la ubicación seleccionada y colocar el otro resultado en otra parte de la ubicación seleccionada. La parte de la ubicación seleccionada incluye, por ejemplo, los primeros bits seleccionados de un registro especificado por la instrucción y la otra parte de la ubicación seleccionada incluye los segundos bits seleccionados del registro especificado por la instrucción.

10 En un ejemplo, se determina un signo de un resultado, así como un signo del otro resultado. El signo de un resultado y el signo del otro resultado se colocan en la ubicación seleccionada.

También se describen y se reivindican en esta invención procedimientos y sistemas implementados por ordenador relacionados con uno o más aspectos. Además, también se describen y pueden estar reivindicados en esta invención servicios relacionados con uno o más aspectos.

15 Se obtienen características y ventajas adicionales a través de las técnicas descritas en esta invención. Otras realizaciones y aspectos se describen en detalle en esta invención y se consideran parte de los aspectos reivindicados.

20 **Breve descripción de los dibujos**

Uno o más aspectos se señalan particularmente y se reivindican claramente como ejemplos en las reivindicaciones al final de la memoria descriptiva. Lo anterior y los objetos, características y ventajas de uno o más aspectos son evidentes a partir de la siguiente descripción detallada, tomada junto con los dibujos adjuntos, donde:

25 La FIG. 1A representa un ejemplo de un entorno informático para incorporar y usar uno o más aspectos de la presente invención;

30 La FIG. 1B representa detalles adicionales de un procesador de la FIG. 1A, según uno o más aspectos de la presente invención;

35 La FIG. 2 representa un ejemplo de un formato de una instrucción de escalado decimal y conversión y división a coma flotante hexadecimal, según uno o más aspectos de la presente invención;

La FIG. 3 representa un ejemplo de un procesamiento asociado con la ejecución de una instrucción de escalado decimal y conversión y división a coma flotante hexadecimal, según uno o más aspectos de la presente invención;

40 La FIG. 4 representa un ejemplo de un procesamiento asociado con una operación de escalado de una instrucción de escalado decimal y conversión y división a coma flotante hexadecimal, según uno o más aspectos de la presente invención;

45 La FIG. 5 representa un ejemplo de un procesamiento lógico para realizar una operación de conversión de la instrucción de escalado decimal y conversión y división a coma flotante hexadecimal, según uno o más aspectos de la presente invención;

50 La FIG. 6 representa un ejemplo de un procesamiento asociado con una operación de división de la instrucción de escalado decimal y conversión y división a coma flotante hexadecimal, según uno o más aspectos de la presente invención;

55 La FIG. 7 representa un ejemplo de un procesamiento asociado con una operación de colocación de la instrucción de escalado decimal y conversión y división a coma flotante hexadecimal, según uno o más aspectos de la presente invención;

Las FIG. 8A-8B representan un ejemplo para facilitar el procesamiento dentro de un entorno informático, según uno o más aspectos de la presente invención;

60 La FIG. 9A representa otro ejemplo de un entorno informático para incorporar y utilizar uno o más aspectos de la presente invención;

La FIG. 9B representa un ejemplo de detalles adicionales de una memoria de la FIG. 9A, según uno o más aspectos de la presente invención;

65 La FIG. 9C representa otro ejemplo de detalles adicionales de una memoria de la FIG. 9A, según uno o más aspectos de la presente invención;

La FIG. 10A representa otro ejemplo más de un entorno informático para incorporar y usar uno o más aspectos de la presente invención;

5 La FIG. 10B representa detalles adicionales de la memoria de la FIG. 10A, según uno o más aspectos de la presente invención;

La FIG. 11 representa una realización de un entorno informático en la nube, según uno más aspectos de la presente invención; y

10 La FIG. 12 representa un ejemplo de capas de modelo de abstracción, según uno o más aspectos de la presente invención.

Descripción detallada

15 Según un aspecto de la presente invención se proporciona una capacidad para facilitar el procesamiento dentro de un entorno informático. Como ejemplo, se proporciona una única instrucción (por ejemplo, una única instrucción de máquina de hardware diseñada en la interfaz de hardware/software) para realizar operaciones de escalado, conversión y división. La instrucción, denominada en la presente memoria instrucción de escalado decimal y conversión y división a coma flotante hexadecimal, forma parte de una arquitectura de conjunto de instrucciones de procesador (ISA) de uso general, que se distribuye mediante un programa en un procesador, tal como un procesador de uso general. (En otro ejemplo, la instrucción puede formar parte de un procesador de propósito especial, tal como un coprocesador configurado para ciertas funciones).

20 Como parte de la ejecución de la única instrucción (por ejemplo, la instrucción de escalado decimal y conversión y división a coma flotante hexadecimal), se realizan varias operaciones que incluyen el escalado de los datos de entrada usando un factor de escalado para proporcionar datos escalados y la conversión de los datos escalados de un formato (por ejemplo, decimal codificado en binario) a otro formato (por ejemplo, coma flotante hexadecimal), y la división del resultado (por ejemplo, el número de coma flotante hexadecimal) en múltiples partes. Cada parte del resultado de la división puede usarse independientemente de otra parte y/o las partes seleccionadas pueden combinarse para proporcionar un resultado con mayor precisión. Cada una de estas operaciones se realiza como parte de la ejecución de la única instrucción, lo que mejora el rendimiento del sistema y reduce el uso de recursos del sistema.

25 En un ejemplo, como se indica, la conversión es de decimal codificado en binario a coma flotante hexadecimal. El decimal codificado en binario es una codificación binaria de un número decimal, donde cada dígito decimal está representado por un número fijo de bits (por ejemplo, 4 u 8 bits). El coma flotante hexadecimal es un formato para codificar números de coma flotante. En un ejemplo, un número de coma flotante hexadecimal incluye un bit de signo, una característica (por ejemplo, 7 bits) y una fracción (por ejemplo, 6, 14 o 28 dígitos). La característica representa un exponente con signo y se obtiene sumando, por ejemplo, 64 al valor del exponente. El intervalo de la característica es de 0 a 127, lo que corresponde a un intervalo del exponente de, por ejemplo, -64 a +63. La magnitud de un número de coma flotante hexadecimal es el producto de su fracción y el número 16 elevado a la potencia del exponente que se representa por su característica. El número es positivo o negativo dependiendo de si el bit de signo es, por ejemplo, cero o uno, respectivamente.

30 Un número de coma flotante hexadecimal puede representarse en varios formatos diferentes, incluyendo un formato corto (por ejemplo, 32 bits), un formato largo (por ejemplo, 64 bits) y un formato extendido (por ejemplo, 128 bits). En cada formato, el primer bit (por ejemplo, el primer bit situado más a la izquierda, el bit 0) es el bit de signo; el siguiente número de bits seleccionado (por ejemplo, siete bits) es la característica, y en los formatos corto y largo, los bits restantes son la fracción, que incluye, por ejemplo, seis o catorce dígitos hexadecimales, respectivamente. En el formato extendido, la fracción es, por ejemplo, una fracción de 28 dígitos, y el número de coma flotante hexadecimal extendido consiste en dos números de formato largo que se denominan partes de orden superior y de orden inferior. La parte de orden superior es cualquier número de coma flotante hexadecimal largo. La fracción de la parte de orden superior contiene, por ejemplo, los 14 dígitos hexadecimales situados más a la izquierda de la fracción de 28 dígitos, y la fracción de la parte de orden inferior contiene, por ejemplo, los 14 dígitos hexadecimales situados más a la derecha de la fracción de 28 dígitos. La característica y el signo de la parte de orden superior son la característica y el signo del número de coma flotante hexadecimal extendido, y el signo y la característica de la parte de orden inferior de un operando extendido se ignoran.

35 Una realización de un entorno informático para incorporar y usar uno o más aspectos de la presente invención se describe con referencia a la FIG. 1A. En un ejemplo, el entorno informático se basa en arquitectura hardware z/Architecture®, ofrecida por International Business Machines Corporation, Amonk, Nueva York. Una realización de la arquitectura de hardware z/Architecture se describe en una publicación titulada "z/Architecture Principles of Operation", publicación de IBM no. SA22-7832-12, decimotercera edición, septiembre de 2019.

60 No obstante, la arquitectura de hardware z/Architecture es solo un ejemplo de arquitectura; otras arquitecturas y/u

otros tipos de entornos informáticos de International Business Machines Corporation y/o de otras entidades pueden incluir y/o usar uno o más aspectos de la presente invención. z/Architecture e IBM, son marcas comerciales o marcas comerciales registradas de International Business Machines Corporation en al menos una jurisdicción.

5 Haciendo referencia a la FIG. 1A, un entorno 100 informático incluye, por ejemplo, un sistema 102 informático mostrado, por ejemplo, en forma de un dispositivo informático de propósito general. El sistema 102 informático puede incluir, pero no se limita a, uno o más procesadores o unidades 104 de procesamiento (por ejemplo, unidades de procesamiento central (CPU)), una memoria 106 (también conocida como memoria del sistema, memoria principal, almacenamiento principal, almacenamiento o almacenamiento central, como ejemplos) y una o más interfaces 108 de entrada/salida (E/S), acopladas entre sí a través de uno o más buses y/u otras conexiones 110.

15 El bus 110 representa uno o más de cualquiera de varios tipos de estructuras de bus, incluyendo un bus de memoria o controlador de memoria, un puerto de gráficos acelerados, y un procesador o bus local que usa cualquiera de una variedad de arquitecturas de buses. A modo de ejemplo, y no de limitación, tales arquitecturas incluyen la Arquitectura estándar de la industria (Industry Standard Architecture, ISA), la Arquitectura de microcanales (Micro Channel Architecture, MCA), la ISA mejorada (Enhanced ISA, EISA), el bus local de la Asociación de estándares de electrónicos de vídeo (Video Electronics Standards Association, VESA) y la Interconexión de componentes periféricos (Peripheral Component Interconnect, PCI).

20 La memoria 106 puede incluir, por ejemplo, una caché 112, tal como una caché compartida, que puede estar acoplada a las cachés 114 locales de los procesadores 104. Además, la memoria 106 puede incluir uno o más programas o aplicaciones 116 y al menos un sistema operativo 118. Un ejemplo de sistema operativo incluye un sistema operativo z/OS®, ofrecido por International Business Machines Corporation, Armonk, Nueva York. z/OS es una marca comercial o marca comercial registrada de International Business Machines Corporation en al menos una jurisdicción. También se pueden usar otros sistemas operativos ofrecidos por International Business Machines Corporation y/u otras entidades. La memoria 106 también puede incluir una o más instrucciones 120 de programa legibles por ordenador, que pueden configurarse para llevar a cabo funciones de realizaciones de aspectos de la invención.

30 El sistema 102 informático puede comunicarse, por ejemplo, a través de las interfaces 108 de E/S con uno o más dispositivos 130 externos, tales como un terminal de usuario, una unidad de cinta, un dispositivo señalador, una pantalla y uno o más dispositivos 134 de almacenamiento de datos, etc. El dispositivo 134 de almacenamiento de datos puede almacenar uno o más programas 136, una o más instrucciones 138 de programa legibles por ordenador y/o datos, etc. Las instrucciones de programa legibles por ordenador pueden ser configuradas para llevar a cabo funciones de realizaciones de aspectos de la invención.

35 El sistema 102 informático también puede comunicarse, por ejemplo, a través de las interfaces 108 de E/S con la interfaz 132 de red, lo que permite que el sistema 102 informático se comuniquen con una o más redes, tales como una red de área local (LAN), una red de área amplia general (WAN) y/o una red pública (por ejemplo, Internet), proporcionando comunicación con otros dispositivos o sistemas informáticos.

40 El sistema informático 102 puede incluir y/o estar acoplado a medios de almacenamiento del sistema informático, volátiles/no volátiles, extraíbles/no extraíbles. Por ejemplo, puede incluir y/o estar acoplado a un medio magnético no extraíble y no volátil (habitualmente denominado "disco duro"), una unidad de disco magnético para leer y escribir en un disco magnético extraíble y no volátil (por ejemplo, un "disquete") y/o una unidad de disco óptico para leer o escribir en un disco óptico extraíble y no volátil, tal como un CD-ROM, un DVD-ROM u otros medios ópticos. Se debe comprender que otros componentes de hardware y/o software podrían ser utilizados junto con el sistema informático 102. Los ejemplos incluyen, pero no se limitan a: microcódigo, controladores de dispositivo, unidades de procesamiento redundantes, matrices de unidades de disco externo, sistemas RAID, unidades de cinta y sistemas de almacenamiento de archivado de datos, etc.

50 El sistema 102 informático puede funcionar con otros numerosos entornos o configuraciones de sistemas informáticos de propósito general o de propósito especial. Ejemplos de sistemas informáticos, entornos y/o configuraciones conocidos que pueden ser adecuados para ser utilizados con el sistema informático 102 incluyen, entre otros, sistemas de ordenadores personales (Personal Computers, PC), sistemas informáticos de servidor, clientes ligeros, clientes pesados, dispositivos portátiles o de mano, sistemas basados en multiprocesador, sistemas basados en microprocesador, decodificadores de TV, productos electrónicos programables para el consumidor, ordenadores de red, sistemas de miniordenador, sistemas informáticos de ordenador central y entornos informáticos distribuidos en la nube que incluyen cualquiera de los sistemas o dispositivos anteriores, y similares.

60 En un ejemplo, un procesador (por ejemplo, el procesador 104) incluye una pluralidad de componentes funcionales usados para ejecutar instrucciones. Tal como se representa en la FIG. 1B, estos componentes funcionales incluyen, por ejemplo, un componente 150 de búsqueda de instrucciones, para buscar instrucciones a ejecutar; una unidad 152 de decodificación de instrucciones, para decodificar las instrucciones buscadas y obtener operandos de las instrucciones decodificadas; uno o más componentes 154 de ejecución de instrucciones para ejecutar las instrucciones decodificadas; un componente 156 de acceso a memoria, para acceder a la memoria para la ejecución de instrucciones, si es necesario; y un componente 158 de reescritura para proporcionar el

5 resultado de las instrucciones ejecutadas. Uno o más de los componentes pueden acceder y/o usar uno o más registros 160 en el procesamiento de instrucciones. Además, uno o más de los componentes pueden, según uno o más aspectos de la presente invención, incluir al menos una parte o tener acceso a uno o más de los otros componentes usados para realizar operaciones de escalado, conversión y/o división de, por ejemplo, una instrucción de escalado decimal, conversión y división a coma flotante hexadecimal (u otro procesamiento que pueda usar uno o más aspectos de la presente invención), como se describe en la presente memoria. El uno o más componentes adicionales incluyen, por ejemplo, un componente de escalado, conversión y división (o uno o más de otros componentes) 170.

10 Según un aspecto de la presente invención, se proporciona una instrucción, denominada en la presente memoria instrucción de escalado decimal y conversión y división a coma flotante hexadecimal, para realizar, como parte de la ejecución de una instrucción de operaciones de escalado, conversión y división para convertir un número de un formato (por ejemplo, decimal), a otro formato (por ejemplo, coma flotante hexadecimal).

15 Una realización de una instrucción de escalado decimal y conversión y división a coma flotante hexadecimal usada para realizar operaciones de escalado, conversión y división se describe con referencia a la FIG. 2. La instrucción se ejecuta, en un ejemplo, usando un procesador de propósito general (por ejemplo, el procesador 104). En la descripción de la presente memoria, se indican ubicaciones específicas, campos específicos y/o tamaños específicos de los campos (por ejemplo, bytes y/o bits específicos). No obstante, se pueden proporcionar otras ubicaciones, campos y/o tamaños. Además, aunque puede especificarse el establecimiento de un bit a un valor particular, por ejemplo, uno o cero, esto es solo un ejemplo. El bit puede establecerse a un valor diferente, tal como el valor opuesto o a otro valor, en otros ejemplos. Son posibles muchas variaciones.

20 En un ejemplo, la instrucción de escalado decimal y conversión y división a coma flotante hexadecimal tiene un formato VRR-b que denota una operación de registro de vector y registro con un código de operación extendido (opcode). En una realización, la instrucción de escalado decimal y conversión y división a coma flotante hexadecimal es parte de una función de vector que proporciona, por ejemplo, vectores de tamaño fijo que van de uno a dieciséis elementos. Cada vector incluye datos que son operados por instrucciones de vector definidas en la función. En una realización, si un vector está formado por múltiples elementos, entonces cada elemento se procesa en paralelo con los otros elementos. La finalización de la instrucción no se produce, en un ejemplo, hasta que se completa el procesamiento de todos los elementos. En otras realizaciones, los elementos se procesan parcialmente en paralelo y/o secuencialmente.

25 En una realización, hay 32 registros de vector y se pueden mapear otros tipos de registros a un cuadrante de los registros de vector. Por ejemplo, un archivo de registro, que es una matriz de registros de procesador en una unidad de procesamiento central (por ejemplo, el procesador 104), puede incluir 32 registros de vectores y cada registro tiene una longitud de 128 bits. Dieciséis registros de coma flotante, que son de 64 bits de longitud, pueden solapar los registros de vectores. Por lo tanto, como ejemplo, cuando se modifica un registro 2 de coma flotante, entonces también se modifica el registro 2 de vector. También son posibles otros mapeos para otros tipos de registros.

30 Los datos de vector aparecen en el almacenamiento, por ejemplo, en la misma secuencia de izquierda a derecha que otros formatos de datos. Los bits de un formato de datos que se numeran del 0-7 constituyen el byte en la ubicación de bytes más a la izquierda (numerado más bajo) en el almacenamiento, los bits 8-15 forman el byte en la siguiente ubicación secuencial, y así sucesivamente. En un ejemplo adicional, los datos de vector pueden aparecer en el almacenamiento en otra secuencia, tal como de derecha a izquierda.

35 Como se muestra en la FIG. 2, en un ejemplo, una instrucción 200 de escala decimal y conversión y división a coma flotante hexadecimal tiene una pluralidad de campos, y un campo puede tener un número de subíndice asociado al mismo. El número de subíndice asociado con un campo de la instrucción denota el operando al que se aplica el campo. Por ejemplo, el número 1 de subíndice asociado al registro de vector V_1 denota que el registro especificado que usa V_1 incluye el primer operando, y así sucesivamente. Un operando de registro es un registro de longitud, que es, por ejemplo, de 128 bits.

40 En una realización ejemplo, la instrucción de escalado y conversión y división a coma flotante hexadecimal 200 incluye los campos 202a, 202b de código de operación (opcode) (por ejemplo, los bits 0-7 y 40-47) que indican operaciones de escalado y conversión y división, donde los datos de entrada son, por ejemplo, un número decimal (por ejemplo, un número decimal codificado en binario) y la salida es, por ejemplo un valor de coma flotante hexadecimal, un campo 204 de primer registro de vector (V_1), (por ejemplo bits 8-11) para designar el registro de un primer vector; un campo 206 de segundo registro de vector (V_2), (por ejemplo, bits 12-15) para designar un segundo registro de vector; un campo 208 de tercer registro de vector (V_3) (por ejemplo, bits 16-19) para designar un tercer registro de vector; y un campo 210 de registro de extensión de bit (RXB) (por ejemplo, bits 36-39), cada uno de los cuales se describe a continuación. En una realización, los campos están separados y son independientes entre sí; sin embargo, en otras realizaciones se puede combinar más de un campo. A continuación se describe información adicional sobre estos campos.

45 Como se muestra en la FIG. 2, en un ejemplo, una instrucción 200 de escala decimal y conversión y división a coma flotante hexadecimal tiene una pluralidad de campos, y un campo puede tener un número de subíndice asociado al mismo. El número de subíndice asociado con un campo de la instrucción denota el operando al que se aplica el campo. Por ejemplo, el número 1 de subíndice asociado al registro de vector V_1 denota que el registro especificado que usa V_1 incluye el primer operando, y así sucesivamente. Un operando de registro es un registro de longitud, que es, por ejemplo, de 128 bits.

50 En una realización ejemplo, la instrucción de escalado y conversión y división a coma flotante hexadecimal 200 incluye los campos 202a, 202b de código de operación (opcode) (por ejemplo, los bits 0-7 y 40-47) que indican operaciones de escalado y conversión y división, donde los datos de entrada son, por ejemplo, un número decimal (por ejemplo, un número decimal codificado en binario) y la salida es, por ejemplo un valor de coma flotante hexadecimal, un campo 204 de primer registro de vector (V_1), (por ejemplo bits 8-11) para designar el registro de un primer vector; un campo 206 de segundo registro de vector (V_2), (por ejemplo, bits 12-15) para designar un segundo registro de vector; un campo 208 de tercer registro de vector (V_3) (por ejemplo, bits 16-19) para designar un tercer registro de vector; y un campo 210 de registro de extensión de bit (RXB) (por ejemplo, bits 36-39), cada uno de los cuales se describe a continuación. En una realización, los campos están separados y son independientes entre sí; sin embargo, en otras realizaciones se puede combinar más de un campo. A continuación se describe información adicional sobre estos campos.

55 En una realización ejemplo, la instrucción de escalado y conversión y división a coma flotante hexadecimal 200 incluye los campos 202a, 202b de código de operación (opcode) (por ejemplo, los bits 0-7 y 40-47) que indican operaciones de escalado y conversión y división, donde los datos de entrada son, por ejemplo, un número decimal (por ejemplo, un número decimal codificado en binario) y la salida es, por ejemplo un valor de coma flotante hexadecimal, un campo 204 de primer registro de vector (V_1), (por ejemplo bits 8-11) para designar el registro de un primer vector; un campo 206 de segundo registro de vector (V_2), (por ejemplo, bits 12-15) para designar un segundo registro de vector; un campo 208 de tercer registro de vector (V_3) (por ejemplo, bits 16-19) para designar un tercer registro de vector; y un campo 210 de registro de extensión de bit (RXB) (por ejemplo, bits 36-39), cada uno de los cuales se describe a continuación. En una realización, los campos están separados y son independientes entre sí; sin embargo, en otras realizaciones se puede combinar más de un campo. A continuación se describe información adicional sobre estos campos.

60 En una realización ejemplo, la instrucción de escalado y conversión y división a coma flotante hexadecimal 200 incluye los campos 202a, 202b de código de operación (opcode) (por ejemplo, los bits 0-7 y 40-47) que indican operaciones de escalado y conversión y división, donde los datos de entrada son, por ejemplo, un número decimal (por ejemplo, un número decimal codificado en binario) y la salida es, por ejemplo un valor de coma flotante hexadecimal, un campo 204 de primer registro de vector (V_1), (por ejemplo bits 8-11) para designar el registro de un primer vector; un campo 206 de segundo registro de vector (V_2), (por ejemplo, bits 12-15) para designar un segundo registro de vector; un campo 208 de tercer registro de vector (V_3) (por ejemplo, bits 16-19) para designar un tercer registro de vector; y un campo 210 de registro de extensión de bit (RXB) (por ejemplo, bits 36-39), cada uno de los cuales se describe a continuación. En una realización, los campos están separados y son independientes entre sí; sin embargo, en otras realizaciones se puede combinar más de un campo. A continuación se describe información adicional sobre estos campos.

65 En un ejemplo, el campo 204 del registro de vector (V_1) se usa para indicar un registro de vector que va a almacenar

el primer operando. El primer operando es el resultado de escalar y convertir un valor decimal en un valor de punto flotante hexadecimal y de dividir el valor de punto flotante hexadecimal en varios operandos (por ejemplo, dos). El segundo operando está contenido en el registro de vector especificado usando el campo 206 del registro de vector (V_2) y es, por ejemplo, un número decimal empaquetado con signo (por ejemplo, un decimal codificado en binario que tiene, por ejemplo, 31 dígitos más un signo) que se escala usando un entero sin signo incluido en el tercer operando, que está contenido en el registro de vector especificado usando el campo 208 del registro de vector (V_3).

En un ejemplo, cada uno de los campos 204, 206, 208 de registro de vector se usa con el campo 210 de RXB para designar el registro de vector. Por ejemplo, el campo 210 de RBX incluye el bit más significativo para un operando designado de registro de vector. Los bits para designaciones de registro no especificados por la instrucción deben reservarse y establecerse a cero. El bit más significativo se concatena, por ejemplo, a la izquierda de la designación del registro de cuatro bits del campo de registro de vector para crear una designación de registro de vector de cinco bits.

En un ejemplo, el campo de RXB incluye cuatro bits (por ejemplo, los bits 0-3), y los bits se definen como sigue:

0 - Bit más significativo para la designación del primer registro de vector (por ejemplo, en los bits 8-11) de la instrucción.

1 - Bit más significativo para la designación del segundo de registro de vector (por ejemplo, en los bits 12-15) de la instrucción, si lo hubiera.

2 - Bit más significativo para la designación del tercer registro de vector (por ejemplo, en los bits 16-19) de la instrucción, si lo hubiera.

3 - Bit más significativo para la designación del cuarto registro de vector (por ejemplo, en los bits 32-35) de la instrucción, si lo hubiera.

Cada bit se establece a cero o uno, por ejemplo, mediante el ensamblador dependiendo del número de registro. Por ejemplo, para los registros 0-15, el bit se establece a 0; para los registros 16-31, el bit se establece a 1, etc.

En una realización, cada bit RXB es un bit de extensión para una ubicación particular en una instrucción que incluye uno o más registros de vector. Por ejemplo, el bit 0 de RXB es un bit de extensión para la ubicación 8-11, que se asigna, por ejemplo, a V_1 y así sucesivamente. En particular, para los registros de vector, el registro que contiene el operando se especifica usando, por ejemplo, un campo de cuatro bits del campo de registro con la adición de su bit de extensión de registro (RXB) correspondiente como el bit más significativo. Por ejemplo, si el campo de cuatro bits es 0110 y el bit de extensión es 0, entonces el campo de cinco bits 00110 indica el número 6 de registro. En una realización adicional, el campo de RXB incluye bits adicionales, y se usa más de un bit como una extensión para cada vector o ubicación.

En la ejecución de un ejemplo de la instrucción escala decimal y conversión y división a coma flotante hexadecimal, el segundo operando se escala usando un entero sin signo en una ubicación seleccionada (por ejemplo, el elemento de byte siete) del tercer operando y se convierte en un número de coma flotante hexadecimal. El segundo operando se multiplica por un factor de escalado que es igual, por ejemplo, a 10 elevado al elemento de byte siete del tercer operando. El resultado escalado se convierte, por ejemplo, en un valor de punto flotante hexadecimal, que se divide en, por ejemplo, dos operandos de punto flotante hexadecimales, utilizando la normalización y el truncamiento. Los dos operandos hexadecimales de coma flotante, por ejemplo, se colocan en el primer registro de operandos.

Se describen con referencia a las FIGS. 3-7, los detalles adicionales de una realización de procesamiento basándose en la ejecución de una instrucción de escalado decimal y conversión y división a coma flotante hexadecimal, según uno o más aspectos de la presente invención. En un ejemplo, se usa un procesador, tal como un procesador 104 general, para ejecutar la instrucción. Como ejemplo, el hardware del procesador se usa para ejecutar la instrucción. El hardware puede estar dentro del procesador o acoplado al mismo con el propósito de recibir la instrucción del procesador, que, por ejemplo, obtiene, decodifica y configura la instrucción para que se ejecute en el hardware. Son posibles otras variaciones.

Haciendo referencia a la FIG. 3, inicialmente, se obtiene una instrucción, tal como una instrucción de escalado decimal y conversión y división a coma flotante hexadecimal (por ejemplo, se busca, se recibe, se proporciona, etc.) 300 y se ejecuta 310. La ejecución incluye, por ejemplo, obtener el segundo y tercer operandos de la instrucción 312. El segundo operando es, por ejemplo, un número decimal empaquetado con signo obtenido de una ubicación (por ejemplo, un registro de vector) especificada por la instrucción (por ejemplo, usando el campo 206 de V_2), y el tercer operando es, por ejemplo, un entero sin signo obtenido de una ubicación (por ejemplo, un registro de vector) especificada por la instrucción (por ejemplo, usando el campo 208 de V_3). En un ejemplo, el entero sin signo está ubicado en el elemento de byte siete del tercer operando.

El segundo operando (por ejemplo, el número decimal empaquetado con signo obtenido usando V_2) se escala usando el entero sin signo en, por ejemplo, el elemento de byte siete del tercer operando (obtenido usando, por ejemplo, V_3) para obtener un resultado escalado 314.

5 El resultado escalado, que está en un formato (por ejemplo, decimal, tal como decimal codificado en binario), se convierte a un resultado convertido en otro formato 316. Por ejemplo, el número decimal escalado se convierte a un número de coma flotante hexadecimal. El resultado convertido se divide en varias partes. Por ejemplo, el número hexadecimal de coma flotante se divide en múltiples (por ejemplo, dos) operandos de punto flotante hexadecimales 318, y uno o más de esos operandos (por ejemplo, los dos operandos) se colocan en la ubicación del primer operando (por ejemplo, un registro especificado usando V_1) 320. Los detalles adicionales relacionados con el escalado, la conversión, la división y la colocación se describen con referencia a las FIGS. 4-6.

15 Haciendo referencia inicialmente a la FIG. 4, se describe una realización de la realización del escalado del segundo operando (314 de la FIG. 3). En un ejemplo, se obtiene (400) un valor, denominado valor de escalado, de una parte seleccionada (por ejemplo, el elemento de byte siete) del tercer operando (almacenado en un registro de vector designado usando V_3). Se determina si el valor es válido (410). Por ejemplo, se determina si el valor tiene una relación predeterminada con un valor preseleccionado, por ejemplo, ¿es el valor menor que un valor preseleccionado, tal como 8, por ejemplo? Si el valor no es válido, el procesamiento finaliza, por ejemplo, con un error. Sin embargo, si el valor es válido, el procesamiento continúa con el uso del valor para determinar un factor de escalado. Por ejemplo, el factor de escalado es igual a 10 a la potencia del valor (415). El segundo operando se multiplica por el factor de escalado para obtener un resultado escalado 420. En un ejemplo, dado que el segundo operando es un número decimal empaquetado con signo, el escalado en una potencia de 10 equivale a desplazar los dígitos a la izquierda. La escala facilita la conversión del número decimal empaquetado firmado en un número hexadecimal de coma flotante.

25 El resultado escalado, que está en un formato (por ejemplo, un formato decimal), se convierte entonces en un resultado convertido, que está en otro formato (por ejemplo, un formato de coma flotante hexadecimal) (316 de la FIG. 3). La conversión se puede realizar usando varias técnicas. En un ejemplo, para convertir un número decimal a un número hexadecimal:

30 El número decimal se divide entre 16 en un cociente y un resto;

El resto multiplicado por 16 es un dígito del número hexadecimal, empezando por el dígito situado más a la derecha;

35 El cociente se divide por 16 para proporcionar otro cociente y un resto; y

El proceso se repite comenzando con el resto multiplicado por 16 hasta que el cociente sea 0.

40 Aunque la técnica anterior se puede usar para convertir un número decimal a un número hexadecimal, también se pueden usar otras técnicas. En un ejemplo, la lógica del hardware se usa para facilitar la conversión, mejorando la velocidad a la que se puede realizar el procesamiento.

45 Un ejemplo de lógica de hardware usada para realizar la conversión se describe con referencia a la FIG. 5. Haciendo referencia a la FIG. 5, en un ejemplo, se introduce en la lógica un número decimal codificado en binario 500. Inicialmente, se seleccionan 510 hasta 4 dígitos del número decimal codificado en binario, comenzando por los dígitos más a la izquierda del número decimal codificado en binario. Los dígitos seleccionados se introducen en un árbol contador 520, que usa un formato redundante de los dígitos y una ecuación para multiplicar cada dígito. En un ejemplo, la multiplicación se realiza desplazando el número, donde cada multiplicación de potencia de 2 es un desplazamiento del número.

50 Un ejemplo de una ecuación usada por el árbol contador es el siguiente, donde $(X' + Y')$ se establece inicialmente en 0 y representa un valor resultante de un bucle anterior en el árbol contador, y A, B, C, D son dígitos del número decimal codificado en binario (BCD).

55 Nueva suma = $X + Y = (X' + Y') * 10000 + A * 1000 + B * 100 + C * 10 + D$

= $(X' + Y') * (2^{13} + 2^{11} - 2^8 + 2^4)$

60 + $A * (2^9 + 2^8 + 2^7 + 2^6 + 2^5 + 2^3)$

+ $B * (2^6 + 2^5 + 2^2)$

+ $C * (2^3 + 2^1)$

65 + D.

ES 3 021 260 T3

El procesamiento se repite en el árbol contador hasta que se hayan procesado todos los dígitos del número BCD. La salida del árbol contador 520 se introduce en un sumador 2:1 530, que proporciona un resultado convertido intermedio (por ejemplo, un número de coma flotante hexadecimal intermedio) en un formato no redundante. La salida del sumador 530 se introduce en la lógica de normalización hexadecimal y actualización de exponente 540, que se usa para normalizar la salida del sumador y actualizar el exponente para proporcionar un número de coma flotante hexadecimal. Por ejemplo, la normalización desplaza hacia la izquierda cualquier dígito cero inicial que esté a la derecha del punto base para eliminar esos dígitos y disminuir el exponente en uno para cada desplazamiento.

En la presente memoria se describe un ejemplo particular del procesamiento anterior para un número BCD de 8 dígitos. En el siguiente ejemplo, dado que los valores reales de X e Y son específicos de la implementación (por ejemplo, X e Y pueden ser valores diferentes pero seguir teniendo la misma suma), se usa Z, que es independiente de la implementación. Por lo tanto, $Z = X + Y$; $Z' = X' + Y'$; y Z' inicializa a 0. Además, en este ejemplo, BCD = 32610423 y, por lo tanto, inicialmente, A = 3; B = 2; C = 6; D = 1.

	Fórmula	Valor Hexadecimal
	$Z = 0 * (2^{13} + 2^{11} - 2^8 + 2^4)$	Z = x0
20	$+ 3 * (2^9 + 2^8 + 2^7 + 2^6 + 2^5 + 2^3)$	+ xBB8
	$+ 2 * (2^6 + 2^5 + 2^2)$	+xC8
	$+ 6 * (2^3 + 2^1)$	+x3C
25	+ 1	+ x1
	$Z = 0 + 3000 + 200 + 60 + 1 = 3261$	Z = xCBD
30	A continuación, A = 0; B = 4; C = 2; D = 3	
	$Z = 3261 * (2^{13} + 2^{11} - 2^8 + 2^4)$	Z=1F96D0
35	$+ 0 * (2^9 + 2^8 + 2^7 + 2^6 + 2^5 + 2^3)$	+ x0
	$+ 4 * (2^6 + 2^5 + 2^2)$	+x190
	$+ 2 * (2^3 + 2^1)$	+x14
40	+ 3	+x3
	$Z = 32610000 + 400 + 20 + 3 = 32610423$	Z = x1F19877

El número hexadecimal de coma flotante (también denominado en el presente documento resultado convertido) se divide en múltiples partes, tales como dos operandos de punto flotante hexadecimales (318 de la figura 3). Por ejemplo, el número de coma flotante hexadecimal se divide usando la normalización y el truncamiento, como se describe más adelante con referencia a la figura 6. En un ejemplo, el número de coma flotante hexadecimal (también conocido como resultado convertido) se normaliza para obtener un primer resultado normalizado (600). El primer resultado normalizado se trunca a un formato corto hexadecimal de coma flotante, denominado resultado truncado corto o resultado alto (610). Como ejemplo, la parte fraccionaria del número de coma flotante hexadecimal (es decir, los dígitos después de la característica) se normaliza (por ejemplo, se eliminan los ceros iniciales) para proporcionar el primer resultado normalizado y, a continuación, se trunca cualquier dígito después de, por ejemplo, los seis dígitos, dado que es el formato corto.

El resultado truncado de forma corta se resta del resultado convertido para obtener una diferencia (620). La diferencia se normaliza, como se describe en el presente documento, para obtener un segundo resultado normalizado (630), y el segundo resultado normalizado se trunca a un formato hexadecimal largo (por ejemplo, trunca los dígitos después de, por ejemplo, 14 dígitos, ya que es el formato largo), denominado resultado bajo (640). Los resultados altos y bajos son los dos operandos hexadecimales de coma flotante resultantes de la conversión y la división. En un ejemplo, si el resultado bajo es igual a cero, se fuerza a un cero verdadero.

Uno o más de los resultados se colocan en la ubicación del primer operando. En un ejemplo, los resultados altos y bajos se colocan en la ubicación del primer operando (320 de la figura 3), como se describe con más detalle con referencia a la figura 7. En un ejemplo, el resultado de formato corto de coma flotante de alto hexadecimal se convierte a un formato hexadecimal largo. Por ejemplo, el resultado corto se coloca en los bits 0 a 31 de la ubicación del primer operando (por ejemplo, el registro vectorial designado por V₁) y los bits 32 a 63 se establecen en ceros

en la ubicación del primer operando 700. El resultado bajo en el formato largo de coma flotante hexadecimal se coloca, por ejemplo, en los bits 64 a 127 de la misma ubicación (por ejemplo, el mismo registro vectorial) 710.

En un ejemplo, se determinan los signos de los resultados 720 altos y bajos. Por ejemplo, el signo del resultado alto es igual al código de signo del segundo operando, excepto cuando el segundo operando es un cero negativo, entonces el resultado alto se establece a un cero con signo positivo. El signo del resultado bajo es igual al signo del resultado alto, en un ejemplo, excepto cuando el resultado bajo se fuerza a un cero verdadero y el resultado alto es distinto de cero y negativo. Los signos determinados se colocan en la ubicación 730 del primer operando. Por ejemplo, el bit 0 se establece en el signo del resultado alto y el bit 64 se establece en el signo del resultado bajo.

En un ejemplo, se verifica la validez de los dígitos y el signo del segundo operando. Si la comprobación de validez falla, se reconoce una excepción de datos de operando general.

Aunque se describen varios campos y registros de la instrucción de escalado decimal y conversión y división a coma flotante hexadecimal, uno o más aspectos de la presente invención pueden usar otros campos y/o registros adicionales y/o menos campos y/o registros, y/u otros tamaños de campos y/o registros, etc. Son posibles muchas variaciones. Por ejemplo, pueden usarse registros implícitos en lugar de registros y/o campos de la instrucción especificados explícitamente y/o pueden usarse registros y/o campos especificados explícitamente en lugar de registros y/o campos implícitos. También son posibles otras variaciones.

Como se describe en la presente memoria, en un aspecto, se proporciona una única instrucción (por ejemplo, una única instrucción de máquina diseñada en la interfaz de hardware/software, por ejemplo, una instrucción de escalado decimal y conversión y división a coma flotante hexadecimal) para realizar un escalado de un número decimal para proporcionar un número decimal escalado, convertir el número decimal escalado a un número de coma flotante hexadecimal, y dividir el número hexadecimal de coma flotante a números múltiples de coma flotante hexadecimal. Esta instrucción es, por ejemplo, una instrucción de hardware definida en una arquitectura de conjunto de instrucciones (ISA) que convierte directamente un valor en un formato, por ejemplo, un número decimal, a un valor en otro formato, por ejemplo, un número de coma flotante hexadecimal. La conversión es directa de, por ejemplo, un número decimal a un número de coma flotante hexadecimal, en lugar de, por ejemplo, de decimal a entero y de entero a coma flotante hexadecimal. El procesamiento es más rápido y eficiente que un programa que realiza, por ejemplo, multiplicar o desplazar un decimal codificado en binario, convertir el decimal codificado en binario a entero y el entero a coma flotante hexadecimal, y retener suficientes dígitos para obtener un número de coma flotante hexadecimal alto de corta precisión y un número de coma flotante hexadecimal de larga precisión. Se reduce la complejidad de un programa relacionado con la realización de operaciones de escalado, conversión y división. Además, se mejora el rendimiento de las operaciones y, por lo tanto, del procesador. La ejecución de instrucciones de hardware reduce los tiempos de ejecución y mejora el rendimiento.

Usando una única instrucción para realizar, por ejemplo, el escalado, la conversión y la división, en lugar de múltiples instrucciones, se mejora el rendimiento no requiriendo múltiples pasadas a través de la interfaz de hardware/software. Además, realizando el procesamiento como parte de una instrucción, el procesamiento permanece en la unidad de procesamiento que realiza las operaciones (por ejemplo, una unidad de procesamiento de coma flotante), sin requerir antes de completar el procesamiento, la actualización de los registros de un archivo de registro del procesador (es decir, una matriz de registros de procesador usados para almacenar datos entre la memoria y las unidades funcionales, por ejemplo, una unidad de procesamiento de coma flotante). Esto mejora el tiempo de ejecución y reduce el uso de los recursos del procesador.

En un ejemplo, al dividir el número de coma flotante hexadecimal en partes más pequeñas (por ejemplo, un número alto corto en coma flotante y un número bajo largo en coma flotante hexadecimal), las partes más pequeñas se pueden usar de forma independiente y/o se pueden sumar varias partes más pequeñas para proporcionar un resultado de mayor precisión.

Uno o más aspectos de la presente invención están inextricablemente ligados a la tecnología informática y facilitan el procesamiento dentro de un ordenador, mejorando el rendimiento del mismo. El uso de una única instrucción de máquina diseñada para realizar un escalado de un número decimal (por ejemplo, un número decimal codificado en binario) para obtener un número decimal escalado, convertir el número decimal escalado en un número de coma flotante hexadecimal y dividir el número de coma flotante hexadecimal en múltiples partes mejora el rendimiento en el entorno informático reduciendo la complejidad, reduciendo el uso de recursos y aumentando la velocidad de procesamiento. Los datos y/o la instrucción se pueden usar en muchos campos técnicos, tal como el procesamiento informático, el procesamiento médico, la ingeniería, las tecnologías del automóvil, la fabricación, etc. Proporcionando optimizaciones en la conversión de datos, estos campos técnicos se mejoran, mediante la reducción del tiempo de ejecución.

Con referencia a las FIGS. 8A-8B, se describen detalles adicionales de una o más realizaciones, en lo que se refiere a uno o más aspectos de la presente invención.

Con referencia a la FIG. 8A, en una realización, se ejecuta una instrucción para realizar operaciones de escalado,

conversión y división (800). La ejecución de la instrucción incluye, por ejemplo, escalar un valor de entrada de un formato para proporcionar un resultado escalado 802, convertir el resultado escalado de un formato para proporcionar un resultado convertido en otro formato 804, dividir el resultado convertido en múltiples partes 806 y colocar una o más partes de las múltiples partes en una ubicación seleccionada 808. Al usar una única instrucción para realizar, al menos, las operaciones de escalado, conversión y división, se mejora el rendimiento y se reduce la utilización de recursos.

En un ejemplo, un formato es un formato decimal y el otro formato es un formato de coma flotante hexadecimal 820. Como ejemplo, el formato decimal es un formato decimal codificado en binario 822. El formato hexadecimal de punto flotante proporciona una precisión mejorada, lo que es beneficioso para varias tecnologías, ya que mejora la precisión y/o el rendimiento.

En un ejemplo, el escalado incluye determinar un factor de escalado y usar el factor de escalado 830 para escalar el valor de entrada para proporcionar el resultado escalado 832. La determinación del factor de escalado incluye, por ejemplo, obtener un valor de escalado usando un operando de la instrucción 834, y usar el valor de escalado para determinar el factor de escalado 836. El valor de entrada se multiplica por el factor de escalado para obtener el resultado escalado 838.

El escalado facilita la conversión de un valor de un formato (por ejemplo, decimal) a otro formato (por ejemplo, coma flotante hexadecimal) aislando ciertos dígitos de un número para indicar una ubicación seleccionada en el número para truncar o redondear, como ejemplos.

Con referencia a la FIG. 8B, en un ejemplo la división incluye normalizar el resultado convertido para obtener un primer resultado normalizado y truncar el primer resultado normalizado 850 para obtener un resultado en un formato corto del otro formato 852. En un ejemplo, la división incluye además restar el resultado del resultado convertido para proporcionar una diferencia 854, normalizar la diferencia para proporcionar un segundo resultado normalizado 856 y truncar el segundo resultado normalizado para obtener otro resultado en un formato largo del otro formato 858.

Al dividir el número de coma flotante hexadecimal en partes más pequeñas, las partes más pequeñas se pueden usar de forma independiente, y/o las partes más pequeñas se pueden sumar para proporcionar un resultado de mayor precisión.

Como ejemplo, la colocación incluye colocar un resultado en una parte de la ubicación seleccionada 860 y colocar el otro resultado en otra parte de la ubicación seleccionada 862. La parte de la ubicación seleccionada incluye, por ejemplo, los primeros bits seleccionados de un registro especificado por la instrucción 864, y la otra parte de la ubicación seleccionada incluye los segundos bits seleccionados del registro especificado por la instrucción 866.

En un ejemplo, un signo de un resultado se determina 867, un signo del otro resultado se determina 868, y el signo de un resultado y el signo del otro resultado se colocan en la ubicación seleccionada 870.

Son posibles otras variaciones y realizaciones.

Los aspectos de la presente invención pueden utilizarse por muchos tipos de entornos informáticos. Una realización de un entorno informático para incorporar y usar uno o más aspectos de la presente invención se describe con referencia a la FIG. 9A. Como un ejemplo, el entorno informático de la FIG. 9A se basa en arquitectura hardware z/Architecture®, ofrecida por International Business Machines Corporation, Amonk, Nueva York. Sin embargo, la arquitectura de hardware z/Architecture es solo un ejemplo de arquitectura. De nuevo, el entorno informático puede estar basado en otras arquitecturas, que incluyen, entre otras, las arquitecturas Intel® x86, otras arquitecturas comercializadas por la firma International Business Machines Corporation y/o arquitecturas comercializadas por otras firmas. Intel es una marca comercial o marca comercial registrada de Intel Corporation o sus subsidiarias en los Estados Unidos y otros países.

En un ejemplo, un entorno 10 informático incluye un complejo electrónico central (CEC) 11. El complejo 11 electrónico central incluye una pluralidad de componentes, tales como, por ejemplo, una memoria 12 (también denominada memoria del sistema, memoria principal, almacenamiento principal, almacenamiento central, almacenamiento) acoplada a uno o más procesadores (también denominados unidades centrales de procesamiento (CPU)) 13 y a un subsistema 14 de entrada/salida (E/S).

El subsistema de E/S 14 puede formar parte del complejo electrónico central o estar separado del mismo. Dirige el flujo de información entre el almacenamiento 12 principal y las unidades 15 de control de entrada/salida y los dispositivos 16 de entrada/salida (E/S) acoplados al complejo electrónico central.

Se pueden utilizar muchos tipos de dispositivos de E/S. Un tipo particular es un dispositivo de almacenamiento de datos 17. El dispositivo 17 de almacenamiento de datos puede almacenar uno o más programas 18, una o más instrucciones 19 de programa legibles por ordenador y/o datos, etc. Las instrucciones de programa legibles por

ordenador pueden ser configuradas para llevar a cabo funciones de realizaciones de aspectos de la invención.

El complejo electrónico central 11 puede incluir y/o estar acoplado a medios de almacenamiento del sistema informático, volátiles/no volátiles, extraíbles/no extraíbles. Por ejemplo, puede incluir y/o estar acoplado a un medio magnético no extraíble y no volátil (habitualmente denominado "disco duro"), una unidad de disco magnético para leer y escribir en un disco magnético extraíble y no volátil (por ejemplo, un "disquete") y/o una unidad de disco óptico para leer o escribir en un disco óptico extraíble y no volátil, tal como un CD-ROM, un DVD-ROM u otros medios ópticos. Se debe comprender que se podrían utilizar otros componentes de hardware y/o software junto con el complejo electrónico central 11. Los ejemplos incluyen, pero no se limitan a: microcódigo, controladores de dispositivo, unidades de procesamiento redundantes, matrices de unidades de disco externo, sistemas RAID, unidades de cinta y sistemas de almacenamiento de archivado de datos, etc.

Además, el complejo electrónico central 11 puede funcionar con muchos otros entornos o configuraciones de sistema informático de propósito general o de propósito especial. Ejemplos de sistemas informáticos, entornos y/o configuraciones bien conocidos que pueden ser adecuados para su uso con el complejo electrónico central 11 incluyen, entre otros, sistemas de ordenadores personales (PC), sistemas informáticos de servidores, clientes ligeros, clientes pesados, dispositivos de mano o portátiles, sistemas de multiprocesadores, sistemas basados en microprocesadores, decodificadores, productos electrónicos de consumo programables, PC en red, sistemas de miniordenadores, sistemas informáticos centralizados y entornos informáticos distribuidos en la nube que incluyen cualquiera de los sistemas o dispositivos anteriores, y similares.

El complejo electrónico central 11 proporciona, en una o más realizaciones, particionamiento lógico y/o soporte de virtualización. En una realización, como se muestra en la FIG. 9B, la memoria 12 incluye, por ejemplo, una o más particiones 20 lógicas, un hipervisor 21 que gestiona las particiones lógicas, y el firmware 22 del procesador. Un ejemplo de hipervisor 21 es el Processor Resource/System Manager (PR/SM™), ofrecido por la firma International Business Machines Corporation, Armonk, Nueva York. Tal como se usa en la presente memoria el firmware incluye, por ejemplo, el microcódigo del procesador. Incluye, por ejemplo, las instrucciones a nivel de hardware y/o las estructuras de datos usadas en la implementación del código máquina de nivel superior. En una realización, incluye, por ejemplo, código patentado que generalmente se suministra como microcódigo que incluye software fiable o microcódigo específico para el hardware subyacente, y controla el acceso del sistema operativo al hardware del sistema. PR/SM es una marca comercial o marca comercial registrada de International Business Machines Corporation en al menos una jurisdicción.

Cada partición lógica 20 es capaz de funcionar como un sistema independiente. Es decir, cada partición lógica se puede reiniciar de forma independiente, ejecutar un sistema operativo 23 huésped, tal como el sistema operativo z/OS®, ofrecido por International Business Machines Corporation, Armonk, Nueva York, u otro código 24 de control, tal como el código de control de funcionalidad de acoplamiento (CFCC), y funcionar con diferentes programas 25. Un sistema operativo o un programa de aplicación que se ejecuta en una partición lógica parece tener acceso a un sistema entero y completo, pero en realidad solo una parte del mismo está disponible. Aunque el sistema operativo z/OS se ofrece como ejemplo, se pueden usar otros sistemas operativos ofrecidos por International Business Machines Corporation y/u otras empresas según uno o más aspectos de la presente invención.

La memoria 12 está acoplada a las CPU 13 (FIG. 9A), que son recursos de procesadores físicos que pueden ser asignados a las particiones lógicas. Por ejemplo, una partición 20 lógica incluye uno o más procesadores lógicos, cada uno de los cuales representa la totalidad o una parte de un recurso 13 de procesador físico que puede ser asignado dinámicamente a la partición lógica.

En otra realización más, el complejo electrónico central proporciona soporte para máquinas virtuales (con o sin soporte de particionamiento lógico). Como se muestra en la FIG. 9C, la memoria 12 del complejo 11 electrónico central incluye, por ejemplo, una o más máquinas 26 virtuales, un gestor de máquinas virtuales, tal como un hipervisor 27, que gestiona las máquinas virtuales, y el firmware 28 del procesador. Un ejemplo de hipervisor 27 es el hipervisor z/VM®, comercializado por la firma International Business Machines Corporation, Armonk, Nueva York. El hipervisor algunas veces se denomina anfitrión. z/VM es una marca comercial o marca comercial registrada de International Business Machines Corporation en al menos una jurisdicción.

El soporte de la máquina virtual del complejo electrónico central proporciona la capacidad de operar grandes números de máquinas 26 virtuales, cada una capaz de operar con diferentes programas 29 y ejecutar un sistema operativo 30 huésped, tal como el sistema operativo Linux®. Cada máquina virtual 26 es capaz de funcionar como un sistema separado. Es decir, cada máquina virtual se puede reiniciar de forma independiente, ejecutar un sistema operativo huésped y operar con diferentes programas. Un sistema operativo o un programa de aplicación que se ejecuta en una máquina virtual parece tener acceso a un sistema total y completo, pero en realidad solo una parte de él está disponible. Aunque z/VM y Linux se ofrecen como ejemplos, se pueden usar otros gestores de máquinas virtuales y/o sistemas operativos según uno o más aspectos de la presente invención. La marca registrada Linux® se usa de conformidad con una sublicencia de la Fundación Linux®, la licenciataria exclusiva de Linus Torvalds, propietario de la marca en todo el mundo.

Otra realización de un entorno informático para incorporar y usar uno o más aspectos de la presente invención se describe con referencia a la FIG. 10A. En este ejemplo, un entorno 36 informático incluye, por ejemplo, una unidad central de procesamiento (CPU) 37 nativa, una memoria 38 y uno o más dispositivos y/o interfaces 39 de entrada/salida acoplados entre sí a través de, por ejemplo, uno o más buses 40 y/u otras conexiones. Como ejemplos, el entorno 36 informático puede incluir un procesador PowerPC® ofrecido por la firma International Business Machines Corporation, Armonk, Nueva York; un Superdome HP con procesadores Intel® Itanium® II ofrecidos por Hewlett Packard Co., Palo Alto, California; y/u otras máquinas basadas en arquitecturas ofrecidas por International Business Machines Corporation, Hewlett Packard, Intel Corporation, Oracle y/u otros. PowerPC es una marca comercial o marca comercial registrada de International Business Machines Corporation en al menos una jurisdicción. Itanium es una marca comercial o marca comercial registrada de Intel Corporation o sus subsidiarias en los Estados Unidos y otros países.

La unidad 37 central de procesamiento nativa incluye uno o más registros 41 nativos, tales como uno o más registros de propósito general y/o uno o más registros de propósito especial usados durante el procesamiento dentro del entorno. Estos registros incluyen información que representa el estado del entorno en cualquier punto particular del tiempo.

Además, la unidad 37 central de procesamiento nativa ejecuta instrucciones y código que están almacenados en la memoria 38. En un ejemplo particular, la unidad central de procesamiento ejecuta el código 42 de emulador almacenado en la memoria 38. Este código permite que el entorno informático configurado en una arquitectura emule otra arquitectura. Por ejemplo, el código de emulador 42 permite a máquinas basadas en arquitecturas diferentes de la arquitectura de hardware z/Architecture, tales como los procesadores PowerPC, los servidores Superdome HP u otros, emular la arquitectura de hardware z/Architecture y ejecutar el software y las instrucciones desarrolladas basadas en la arquitectura de hardware z/Architecture.

Se describen detalles adicionales relacionados con el código 42 del emulador con referencia a la FIG. 10B. Las instrucciones 43 de huésped almacenadas en la memoria 38 comprenden instrucciones de software (por ejemplo, correlacionadas con instrucciones de máquina) que fueron desarrolladas para ser ejecutadas en una arquitectura distinta a la de la CPU 37 nativa. Por ejemplo, las instrucciones 43 de huésped pueden haber sido diseñadas para ser ejecutadas en un procesador basado en la arquitectura de hardware z/Architecture, pero en cambio, se están emulando en una CPU 37 nativa, que puede ser, por ejemplo, un procesador Intel Itanium II. En un ejemplo, el código 42 de emulador incluye una rutina 44 de búsqueda de instrucciones para obtener una o más instrucciones 43 de huésped de la memoria 38 y, opcionalmente, proporcionar almacenamiento en memoria intermedia local para las instrucciones obtenidas. También incluye una rutina 45 de traducción de instrucciones para determinar el tipo de instrucción de huésped que se ha obtenido y para traducir la instrucción de huésped en una o más instrucciones 46 nativas correspondientes. Esta traducción incluye, por ejemplo, identificar la función que realizará la instrucción de huésped y elegir la instrucción o instrucciones nativas para realizar esa función.

Además, el código 42 de emulador incluye una rutina 47 de control de emulación para hacer que se ejecuten las instrucciones nativas. La rutina 47 de control de emulación puede hacer que la CPU 37 nativa ejecute una rutina de instrucciones nativas que emulen una o más instrucciones de huésped obtenidas previamente y, a la conclusión de dicha ejecución, devuelva el control a la rutina de búsqueda de instrucciones para emular la obtención de la siguiente instrucción de huésped o un grupo de instrucciones de huésped. La ejecución de las instrucciones 46 nativas puede incluir cargar datos en un registro desde la memoria 38; almacenar datos de vuelta a la memoria desde un registro; o realizar algún tipo de operación aritmética o lógica, según lo determinado por la rutina de traducción.

Cada rutina se implementa, por ejemplo, en un software, que se almacena en la memoria y se ejecuta mediante la unidad central de procesamiento nativa 37. En otros ejemplos, una o más de las rutinas u operaciones se implementan en firmware, hardware, software o alguna combinación de los mismos. Los registros del procesador emulado pueden ser emulados usando los registros 41 de la CPU nativa o usando ubicaciones en la memoria 38. En realizaciones, las instrucciones 43 de huésped, las instrucciones 46 nativas y el código 42 de emulador pueden residir en la misma memoria o pueden estar distribuidos entre diferentes dispositivos de memoria.

Una instrucción que se puede emular es la instrucción de escala decimal y conversión y división descrita en el presente documento, según un aspecto de la presente invención.

Los entornos informáticos descritos anteriormente son solo ejemplos de entornos informáticos que se pueden utilizar. Otros entornos, que incluyen, entre otros, entornos no particionados, entornos particionados, entornos informáticos y/o entornos emulados; las realizaciones no están limitadas a ningún entorno. Aunque en la presente memoria se describen diversos ejemplos de entornos informáticos, uno o más aspectos de la presente invención se pueden utilizar con muchos tipos de entornos. Los entornos informáticos proporcionados en la presente memoria son solo ejemplos.

Cada entorno informático puede ser configurado para incluir uno o más aspectos de la presente invención.

Uno o más aspectos pueden estar relacionados con la informática en la nube.

5 Debe entenderse que, aunque esta descripción incluye una descripción detallada de la informática en la nube, la implementación de las enseñanzas mencionadas en el presente documento no se limita a un entorno informático en la nube. Más bien, las realizaciones de la presente invención son capaces de ser implementadas junto con cualquier otro tipo de entorno informático conocido ahora o desarrollado posteriormente.

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

15 Las características son como sigue:

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

20 Acceso a red amplia: las capacidades están disponibles sobre una red y se accede a ellas a través de mecanismos estándar que favorecen la utilización por parte de plataformas heterogéneas de clientes ligeros o pesados (por ejemplo, teléfonos móviles, ordenadores portátiles y PDA).

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

30 Elasticidad rápida: las capacidades se pueden aprovisionar rápida y elásticamente, en algunos casos de manera automática, para ampliar y liberar rápidamente para reducir rápidamente. Para el consumidor, las capacidades disponibles para el aprovisionamiento a menudo parecen ilimitadas y pueden adquirirse en cualquier cantidad en cualquier momento.

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

40 Los modelos de servicio son como sigue:

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

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

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

60 Los modelos de implementación son como sigue:

Nube privada: La infraestructura en la nube opera solamente para una organización. La pueden gestionar la organización o un tercero y puede existir en las instalaciones o fuera de las instalaciones.

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

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

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

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

25 Haciendo referencia ahora a la FIG. 11, se representa el entorno 50 de informática en la nube ilustrativo. Como se muestra, el entorno informático 50 en la nube incluye uno o más nodos informáticos 52 en la nube con los que se pueden comunicar los dispositivos informáticos locales utilizados por los consumidores de la nube, tales como, por ejemplo, un asistente digital personal (Personal Digital Assistant, PDA) o un teléfono celular 54A, un ordenador de sobremesa 54B, un ordenador portátil 54C y/o el sistema informático de automóvil 54N. Los nodos 52 pueden comunicarse entre sí. Se pueden agrupar (no mostrado) física o virtualmente, en una o más redes, tales como nubes privadas, comunitarias, públicas o híbridas, como se ha descrito anteriormente, o en una combinación de las mismas. Esto permite que el entorno 50 de informática en la nube ofrezca infraestructura, plataformas y/o software como servicios para los que un consumidor de la nube no necesita mantener recursos en un dispositivo informático local. Se entiende que se pretende que los tipos de dispositivos 54A-N informáticos mostrados en la FIG. 11 sean solamente ilustrativos y que los nodos 52 informáticos y el entorno 50 informático en la nube puedan comunicarse con cualquier tipo de dispositivo informatizado sobre cualquier tipo de red y/o conexión direccionable de red (por ejemplo, utilizando un navegador web).

35 Haciendo referencia ahora a la FIG. 12, se muestra un conjunto de capas de abstracción funcionales proporcionadas por el entorno 50 de informática en la nube (FIG. 11). Debe entenderse de antemano que los componentes, capas y funciones mostrados en la FIG. 12 se pretende que sean únicamente ilustrativos y las realizaciones de la invención no se limitan a los mismos. Como se representa, están dispuestas las siguientes capas y funciones correspondientes:

40 La capa 60 de hardware y software incluye componentes de hardware y software. Ejemplos de componentes de hardware incluyen: ordenadores centrales 61; servidores basados en arquitectura RISC (Reduced Instruction Set Computer) 62; servidores blade 64; dispositivos de almacenamiento 65; y redes y componentes de red 66. En algunas realizaciones, los componentes de software incluyen software de servidor de aplicaciones de red 67 y el software de base de datos 68.

50 La capa de virtualización 70 proporciona una capa de abstracción a partir de la que se pueden proporcionar los siguientes ejemplos de entidades virtuales: servidores 71 virtuales; almacenamiento 72 virtual; redes 73 virtuales, que incluyen redes privadas virtuales; aplicaciones virtuales y sistemas 74 operativos; y clientes 75 virtuales.

55 En un ejemplo, la capa 80 de gestión puede proporcionar las funciones que se describen a continuación. El aprovisionamiento 81 de recursos proporciona una adquisición dinámica de recursos informáticos y otros recursos que se utilizan para realizar tareas dentro del entorno informático en la nube. La Medición y Fijación de Precios 82 proporcionan un seguimiento de los costes a medida que se utilizan los recursos dentro del entorno informático en la nube, y el cobro o facturación por el consumo de estos recursos. En un ejemplo, estos recursos pueden incluir licencias de software de aplicaciones. La seguridad proporciona comprobación de identidad para los consumidores y las tareas en la nube, así como protección para los datos y otros recursos. El portal de usuario 83 proporciona acceso al entorno informático en la nube para consumidores y administradores de sistemas. La gestión de nivel de servicio 84 proporciona la asignación y gestión de recursos informáticos en la nube, de tal manera que se cumplan los niveles de servicio requeridos. La planificación y el cumplimiento del Acuerdo de Nivel de Servicio (SLA) 85 proporcionan disposición previa para, y adquisición de, recursos informáticos en la nube para los que se anticipa un requisito futuro según un SLA.

65 La capa de cargas de trabajo 90 proporciona ejemplos de funcionalidad para lo cual se puede utilizar el entorno informático en la nube. Ejemplos de cargas de trabajo y funciones que se pueden proporcionar desde esta capa incluyen: mapeo y navegación 91; desarrollo de software y gestión del ciclo de vida 92; distribución de educación

en el aula virtual 93; procesamiento de analíticas de datos 94; procesamiento de transacciones 95; y procesamiento de escalado, conversión y división 96.

5 Aspectos de la presente invención puede ser un sistema, un procedimiento y/o un producto de programa informático en cualquier posible nivel de detalle técnico de integración. El producto de programa informático puede incluir un medio (o medios) de almacenamiento legible por ordenador que tiene instrucciones de programa legibles por ordenador en el mismo para hacer que un procesador lleve a cabo aspectos de la presente invención.

10 El medio de almacenamiento legible por computadora puede ser un dispositivo tangible que puede retener y almacenar instrucciones para su uso por un dispositivo de ejecución de instrucciones. El medio de almacenamiento legible por ordenador puede ser, por ejemplo, entre otros, un dispositivo de almacenamiento electrónico, un dispositivo de almacenamiento magnético, un dispositivo de almacenamiento óptico, un dispositivo de almacenamiento electromagnético, un dispositivo de almacenamiento de semiconductores o cualquier combinación adecuada de los anteriores. Una lista no exhaustiva de ejemplos más específicos del medio de
15 almacenamiento legible por ordenador incluye lo siguiente: un disquete de ordenador portátil, un disco duro, una memoria de acceso aleatorio (Random Access Memory, RAM), una memoria de solo lectura (Read Only Memory, ROM), una memoria de solo lectura programable y borrable (Erasable Programmable Read-Only Memory, EPROM o memoria Flash), una memoria de acceso aleatorio estática (Static RAM, SRAM), una memoria de solo lectura de disco compacto portátil (Compact Disc-ROM, CD-ROM), un disco versátil digital (Digital Versatile Disk, DVD), una memoria USB, un disquete, un dispositivo codificado mecánicamente tal como tarjetas perforadas o estructuras elevadas en una ranura que tienen instrucciones grabadas en ellas, y cualquier combinación adecuada de los anteriores. Un medio de almacenamiento legible por ordenador, como se usa en el presente documento, no debe interpretarse como señales transitorias en sí, tales como ondas de radio u otras ondas electromagnéticas que se propagan libremente, ondas electromagnéticas que se propagan a través de una guía de ondas u otros medios de
20 transmisión (por ejemplo, pulsos de luz que pasan a través de un cable de fibra óptica) o señales eléctricas transmitidas a través de un cable.

Las instrucciones del programa legibles por ordenador descritas en el presente documento pueden descargarse a los respectivos dispositivos informáticos/de procesamiento desde un medio de almacenamiento legible por
30 ordenador o a un ordenador externo o dispositivo de almacenamiento externo a través de una red, por ejemplo, Internet, una red de área local, una red de área amplia y/o una red inalámbrica. La red puede comprender cables de transmisión de cobre, fibras de transmisión óptica, transmisión inalámbrica, enrutadores, cortafuegos, conmutadores, ordenadores de puerta de enlace y/o servidores periféricos. Una tarjeta de adaptador de red o interfaz de red en cada dispositivo informático/de procesamiento recibe instrucciones de programa legibles por ordenador desde la red y reenvía las instrucciones de programa legibles por ordenador para su almacenamiento en un medio de
35 almacenamiento legible por ordenador dentro del dispositivo informático/de procesamiento respectivo.

Las instrucciones de programa legibles por ordenador para llevar a cabo las operaciones de la presente invención pueden ser instrucciones de ensamblador, instrucciones de arquitectura de conjunto de instrucciones (ISA),
40 instrucciones de máquina, instrucciones dependientes de la máquina, microcódigo, instrucciones de firmware, datos de ajuste de estado, datos de configuración para circuito integrado o código fuente o código objeto escritos en cualquier combinación de uno o más lenguajes de programación, incluyendo un lenguaje de programación orientado a objetos tal como Smalltalk, C++ o similares, y lenguajes de programación procedimentales, tal como el lenguaje de programación "C" o lenguajes de programación similares. Las instrucciones de programa legibles
45 por ordenador pueden ser ejecutadas completamente en el ordenador del usuario, parcialmente en el ordenador del usuario, como un paquete de software independiente, parcialmente en el ordenador del usuario y parcialmente en un ordenador remoto o completamente en el ordenador o servidor remoto. En este último escenario, el ordenador remoto puede conectarse al ordenador del usuario a través de cualquier tipo de red, incluyendo una red de área local (Local Area Network, LAN) o una red de área amplia (Wide Area Network, WAN), o la conexión puede realizarse a un ordenador externo (por ejemplo, a través de Internet utilizando un Proveedor de Servicios de Internet). En algunas realizaciones, los circuitos electrónicos incluyen, por ejemplo, circuitos lógicos programables, matrices de puertas programables en campo (FPGA) o matrices lógicas programables (PLA) que pueden ejecutar
50 instrucciones de programa legibles por ordenador utilizando información de estado de las instrucciones de programa legibles por ordenador para personalizar los circuitos electrónicos, con el fin de realizar aspectos de la presente invención.

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

Estas instrucciones de programa legibles por ordenador pueden ser proporcionadas a un procesador de un ordenador, u otro aparato de procesamiento de datos programable para producir una máquina, de manera que las
65 instrucciones, que se ejecutan a través del procesador del ordenador u otro aparato programable de procesamiento de datos, creen medios para implementar las funciones/acciones especificadas en el bloque o bloques del

diagrama de flujo y/o diagrama de bloques. Estas instrucciones de programa legibles por ordenador también pueden almacenarse en un medio de almacenamiento legible por ordenador que puede dirigir un ordenador, un aparato de procesamiento de datos programable y/u otros dispositivos para que funcionen de una manera particular, de tal manera que el medio de almacenamiento legible por ordenador que tiene instrucciones almacenadas en el mismo comprende un artículo de fabricación, que incluye instrucciones que implementen aspectos de la función/acciones especificadas en el bloque o bloques en el diagrama de flujo y/o diagrama de bloques.

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

El diagrama de flujo y los diagramas de bloques de las Figuras ilustran la arquitectura, funcionalidad y operación de posibles implementaciones de sistemas, procedimientos y productos de programas informáticos según varias realizaciones de la presente invención. A este respecto, cada bloque en el diagrama de flujo o los diagramas de bloques puede representar un módulo, un segmento o una parte de instrucciones, que comprende una o más instrucciones ejecutables para implementar la función o funciones lógicas especificadas. En algunas implementaciones alternativas, las funciones indicadas en los bloques pueden ocurrir fuera del orden indicado en las figuras. Por ejemplo, dos bloques mostrados en sucesión pueden, de hecho, ser realizados como una sola etapa, ejecutados concurrentemente, sustancialmente concurrentes, de manera parcial o totalmente superpuesta temporalmente, o los bloques pueden a veces ser ejecutados en el orden inverso, dependiendo de la funcionalidad implicada. También se observará que cada bloque de los diagramas de bloques y/o la ilustración del diagrama de flujo, y las combinaciones de bloques en los diagramas de bloques y/o la ilustración del diagrama de flujo, pueden implementarse mediante sistemas basados en hardware de propósito especial que realizan las funciones o actos especificados o llevan a cabo combinaciones de hardware de propósito especial e instrucciones de ordenador.

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

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

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

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

Aunque anteriormente se han descrito varias realizaciones, estas son solamente ejemplos. Por ejemplo, se pueden usar entornos informáticos de otras arquitecturas para incorporar y usar uno o más aspectos. Además, se pueden usar diferentes instrucciones u operaciones. Además, se pueden usar diferentes tipos de registros y/o diferentes registros. Son posibles muchas variaciones.

En la presente memoria se describen diversos aspectos. Cabe señalar que, salvo incoherencia, cada aspecto o característica descrita en la presente memoria, y variantes de los mismos, pueden ser combinados con cualquier otro aspecto o característica.

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

Los dispositivos de entrada/salida o E/S (que incluyen, pero no se limita a, teclados, pantallas, dispositivos

5 señaladores, DASD, cintas, CD, DVD, memorias USB y otros medios de memoria, etc.) se pueden acoplar al sistema o bien directamente o a través de controladores de E/S intermedios. Los adaptadores de red también se pueden acoplar al sistema para permitir que el sistema de procesamiento de datos llegue a acoplarse a otros sistemas de procesamiento de datos o impresoras remotas o dispositivos de almacenamiento a través de redes privadas o públicas intermedias. Los módems, módems de cable y tarjetas Ethernet son tan solo algunos de los tipos de adaptadores de red disponibles.

10 La terminología utilizada en el presente documento tiene el propósito de describir las realizaciones particulares solamente y no se pretende que sea limitante. Como se usa en el presente documento, las formas singulares "un", "una" y "el", "la" se pretende que incluyan también las formas plurales, a menos que el contexto lo indique claramente de otro modo. Se entenderá además que los términos "comprende" y/o "que comprende", cuando se usan en esta memoria descriptiva, especifican la presencia de características, números enteros, etapas, operaciones, elementos y/o componentes indicados, pero no excluyen la presencia o adición de una o más características, números enteros, etapas, operaciones, elementos, componentes y/o grupos de los mismos.

15 Las correspondientes estructuras, materiales, actos y equivalentes de todos los medios o etapas más elementos de función en las reivindicaciones siguientes, si los hay, se pretende que incluyan cualquier estructura, material o acto para realizar la función en combinación con otros elementos reivindicados, como se reivindica específicamente. La descripción de una o más realizaciones se ha presentado con fines ilustrativos y descriptivos, pero no se pretende que sea exhaustiva ni se limite a la forma divulgada. Muchas modificaciones y variaciones resultarán evidentes para los expertos en la técnica. La realización se eligió y describió con el fin de explicar mejor diversos aspectos y la aplicación práctica, y para permitir que otros expertos en la materia comprendan diversas realizaciones con diversas modificaciones que sean adecuadas para el uso particular contemplado.

25

REIVINDICACIONES

1. Un producto de programa informático para facilitar el procesamiento dentro de un entorno informático, comprendiendo el producto de programa informático:

uno o más medios de almacenamiento legibles por ordenador e instrucciones de programa almacenadas colectivamente en el uno o más medios de almacenamiento legibles por ordenador para realizar un procedimiento que comprende:

ejecutar una instrucción única (200) para realizar las operaciones de escalado y conversión, y división, comprendiendo la ejecución de la instrucción:

escalar un valor de entrada en un formato para proporcionar un resultado escalado;

convertir el resultado escalado de un formato para proporcionar un resultado convertido en otro formato;

dividir, mediante la normalización y el truncamiento, el resultado convertido en varias partes;

colocar una o más partes de las múltiples partes en una ubicación seleccionada; y

en donde el escalado, la conversión, la división y la colocación se realizan como parte de la ejecución de la instrucción única.

2. El producto de programa informático de la reivindicación 1, en donde un formato es un formato decimal y el otro formato es un formato de coma flotante hexadecimal.

3. El producto de programa informático de la reivindicación 2, en donde el formato decimal es un formato decimal codificado en binario.

4. El producto de programa informático de una cualquiera de las reivindicaciones anteriores, en donde el escalado comprende:

determinar un factor de escalado; y

usar el factor de escalado para escalar el valor de entrada para proporcionar el resultado escalado.

5. El producto de programa informático de la reivindicación 4, en donde la determinación del factor de escalado comprende:

obtener un valor de escalado usando un operando de la instrucción;

usar el valor de escalado para determinar el factor de escalado, y

en donde el uso del factor de escalado comprende multiplicar el valor de entrada por el factor de escalado para obtener el resultado escalado.

6. El producto de programa informático de una cualquiera de las reivindicaciones anteriores, en donde la división comprende:

normalizar el resultado convertido para obtener un primer resultado normalizado; y

truncar el primer resultado normalizado para obtener un resultado en un formato corto del otro formato.

7. El producto de programa informático de la reivindicación 6, en donde la división además comprende:

restar un resultado del resultado convertido para obtener una diferencia;

normalizar la diferencia para proporcionar un segundo resultado normalizado; y

truncar el segundo resultado normalizado para obtener otro resultado en un formato largo del otro formato.

8. El producto de programa informático de la reivindicación 7, en donde la colocación comprende:

colocar un resultado en una parte de la ubicación seleccionada; y

colocar el otro resultado en una parte de la ubicación seleccionada.

5 9. El producto de programa informático de la reivindicación 8, en donde parte de la ubicación seleccionada incluye, por ejemplo, los primeros bits seleccionados de un registro especificado por la instrucción y la otra parte de la ubicación seleccionada incluye los segundos bits seleccionados del registro especificado por la instrucción.

10 10. El producto de programa informático de la reivindicación 8, en donde la ejecución de la instrucción comprende además:

10 determinar un signo de un resultado;

determinar un signo del otro resultado; y

15 colocar el signo de un resultado y el signo del otro resultado en la ubicación seleccionada.

11. Un sistema informático para facilitar el procesamiento seguro dentro de un entorno informático, comprendiendo el sistema informático:

20 una memoria; y

un procesador en comunicación con la memoria, en donde el sistema informático está configurado para realizar un procedimiento, comprendiendo dicho procedimiento:

25 ejecutar una instrucción única (200) para realizar las operaciones de escalado y conversión, y

división, comprendiendo la ejecución de la instrucción:

30 escalar un valor de entrada en un formato para proporcionar un resultado escalado;

convertir el resultado escalado de un formato para proporcionar un resultado convertido en otro formato;

35 dividir, mediante la normalización y el truncamiento, el resultado convertido en varias partes;

colocar una o más partes de las múltiples partes en una ubicación seleccionada; y

40 en donde el escalado, la conversión, la división y la colocación se realizan como parte de la ejecución de la instrucción única.

12. El sistema informático de la reivindicación 11, en donde el escalado comprende:

40 determinar un factor de escalado; y

usar el factor de escalado para escalar el valor de entrada para proporcionar el resultado escalado.

45 13. El sistema informático de una cualquiera de las reivindicaciones 11 a 12, en donde la división comprende:

normalizar el resultado convertido para obtener un primer resultado normalizado; y

50 truncar el primer resultado normalizado para obtener un resultado en un formato corto del otro formato.

14. El sistema informático de la reivindicación 13, en donde la división comprende además:

55 restar un resultado del resultado convertido para obtener una diferencia;

normalizar la diferencia para proporcionar un segundo resultado normalizado; y

truncar el segundo resultado normalizado para obtener otro resultado en un formato largo del otro formato.

60 15. El sistema informático de la reivindicación 14, en donde la colocación comprende:

colocar un resultado en una parte de la ubicación seleccionada; y

colocar el otro resultado en una parte de la ubicación seleccionada.

65 16. Un procedimiento implementado por ordenador para facilitar el procesamiento dentro de un entorno informático, comprendiendo el procedimiento implementado por ordenador:

ES 3 021 260 T3

ejecutar una única instrucción (200) para realizar operaciones de escalado, conversión y división, comprendiendo la ejecución de la instrucción:

- 5 escalar un valor de entrada en un formato para proporcionar un resultado escalado; convertir el resultado escalado de un formato para proporcionar un resultado convertido en otro formato;
- dividir, mediante la normalización y el truncamiento, el resultado convertido en varias partes;
- 10 colocar una o más partes de las múltiples partes en una ubicación seleccionada; y
- en donde el escalado, la conversión, la división y la colocación se realizan como parte de la ejecución de la instrucción única.
- 15 17. El procedimiento implementado por ordenador de la reivindicación 16, en donde el escalado comprende:
- determinar un factor de escalado; y
- usar el factor de escalado para escalar el valor de entrada para proporcionar el resultado escalado.
- 20 18. El procedimiento implementado por ordenador de una cualquiera de las reivindicaciones 16 a 17, en donde la división comprende:
- normalizar el resultado convertido para obtener un primer resultado normalizado; y
- 25 truncar el primer resultado normalizado para obtener un resultado en un formato corto del otro formato.
19. El procedimiento implementado por ordenador de la reivindicación 18, en donde la división además comprende:
- 30 restar un resultado del resultado convertido para obtener una diferencia;
- normalizar la diferencia para proporcionar un segundo resultado normalizado; y
- truncar el segundo resultado normalizado para obtener otro resultado en un formato largo del otro formato.
- 35 20. El procedimiento implementado por ordenador de la reivindicación 19, en donde la colocación comprende:
- colocar un resultado en una parte de la ubicación seleccionada; y
- 40 colocar el otro resultado en una parte de la ubicación seleccionada.

DIBUJOS

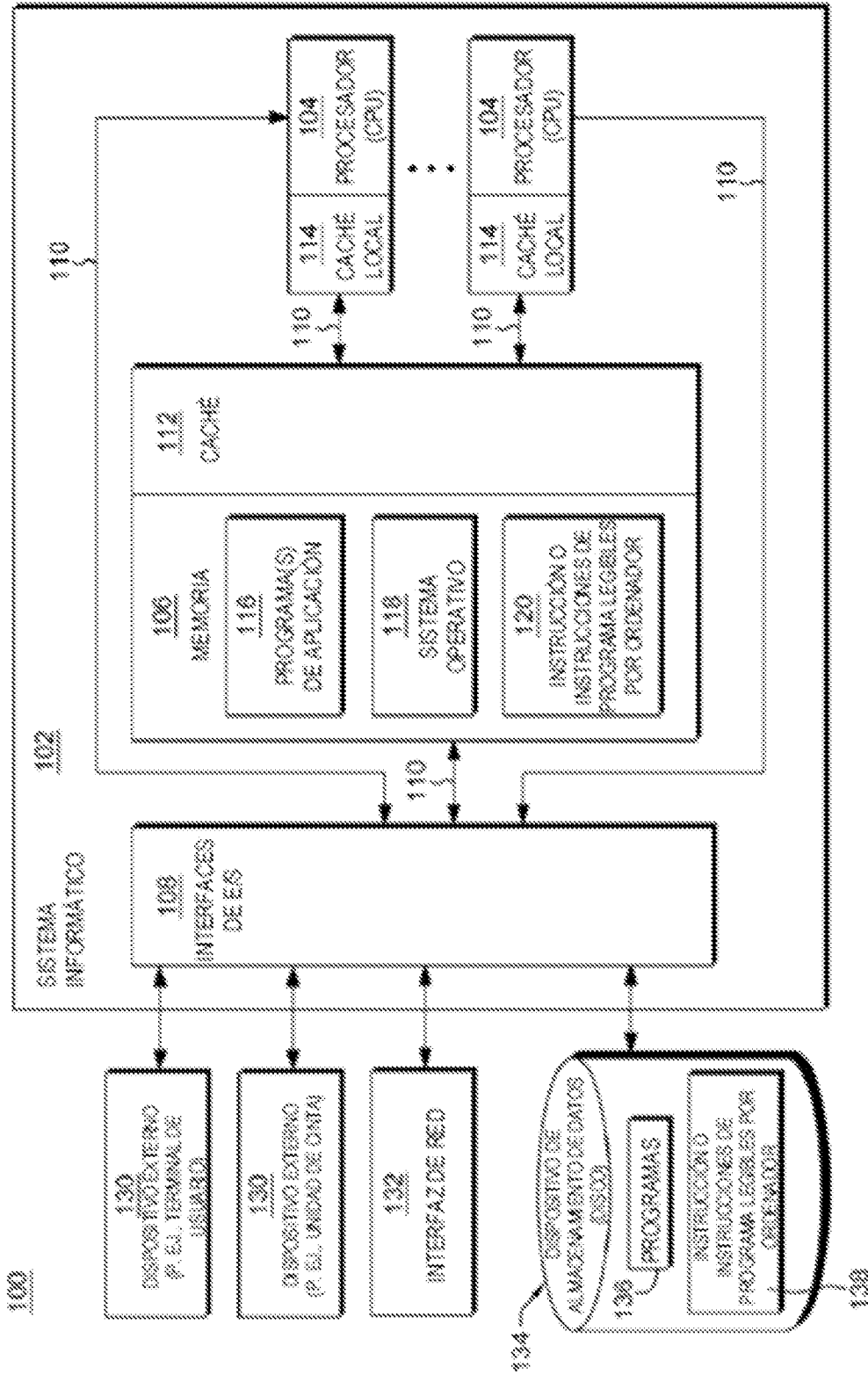


FIG. 1A

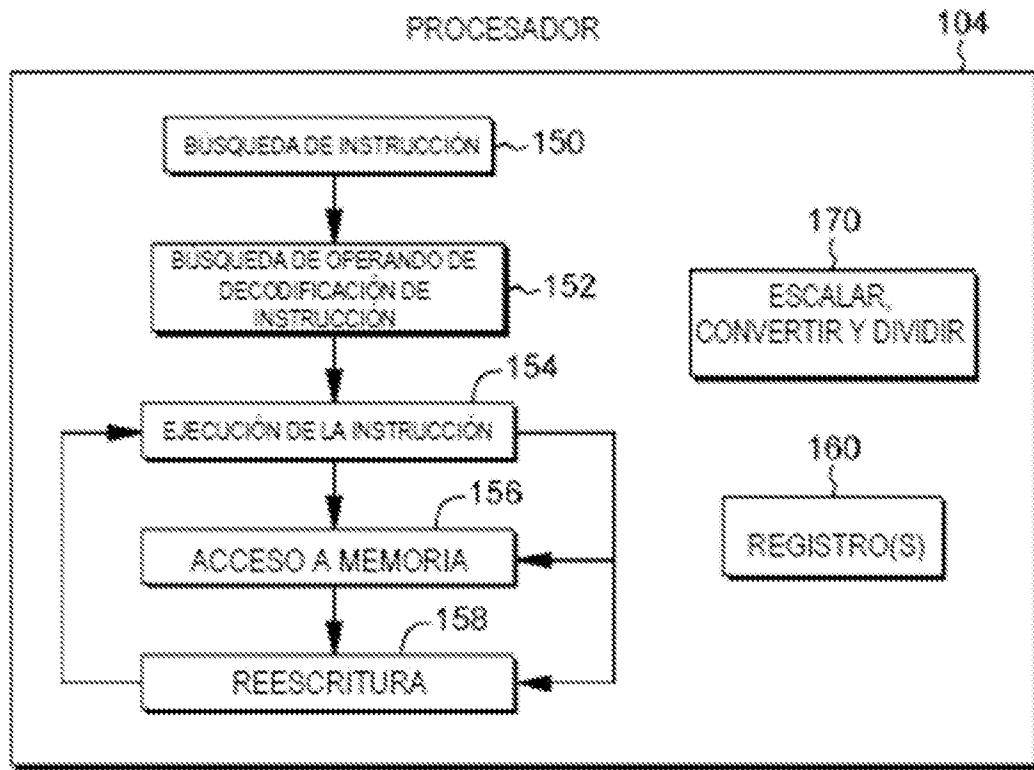


FIG. 1B

ESCALADO DECIMAL Y CONVERSIÓN Y
DIVISIÓN A COMA FLOTANTE HEXADECIMAL

200

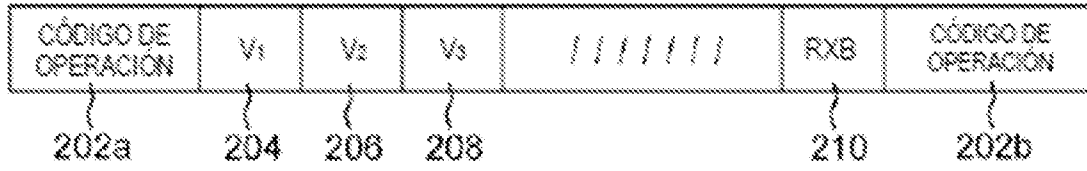


FIG. 2

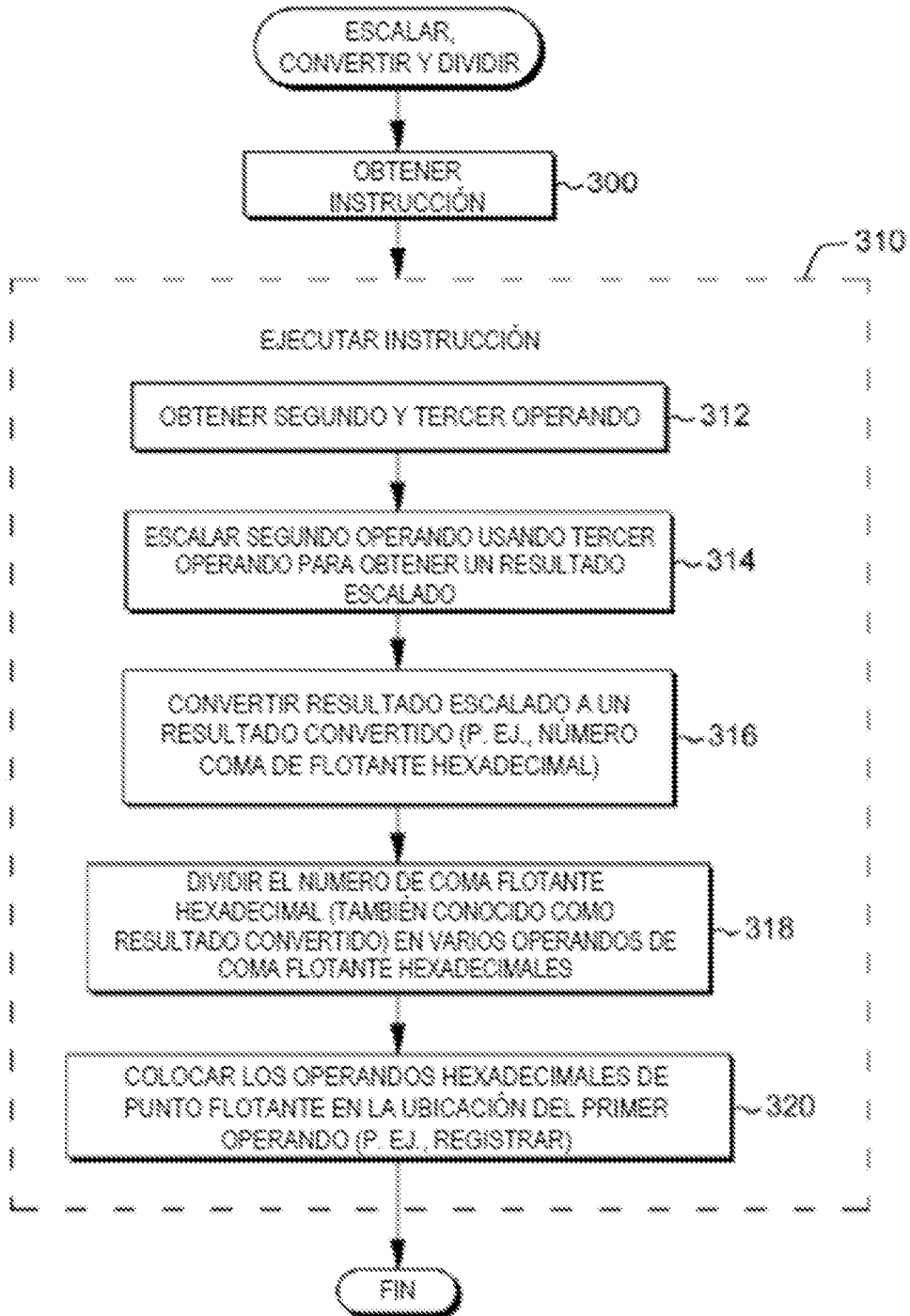


FIG. 3

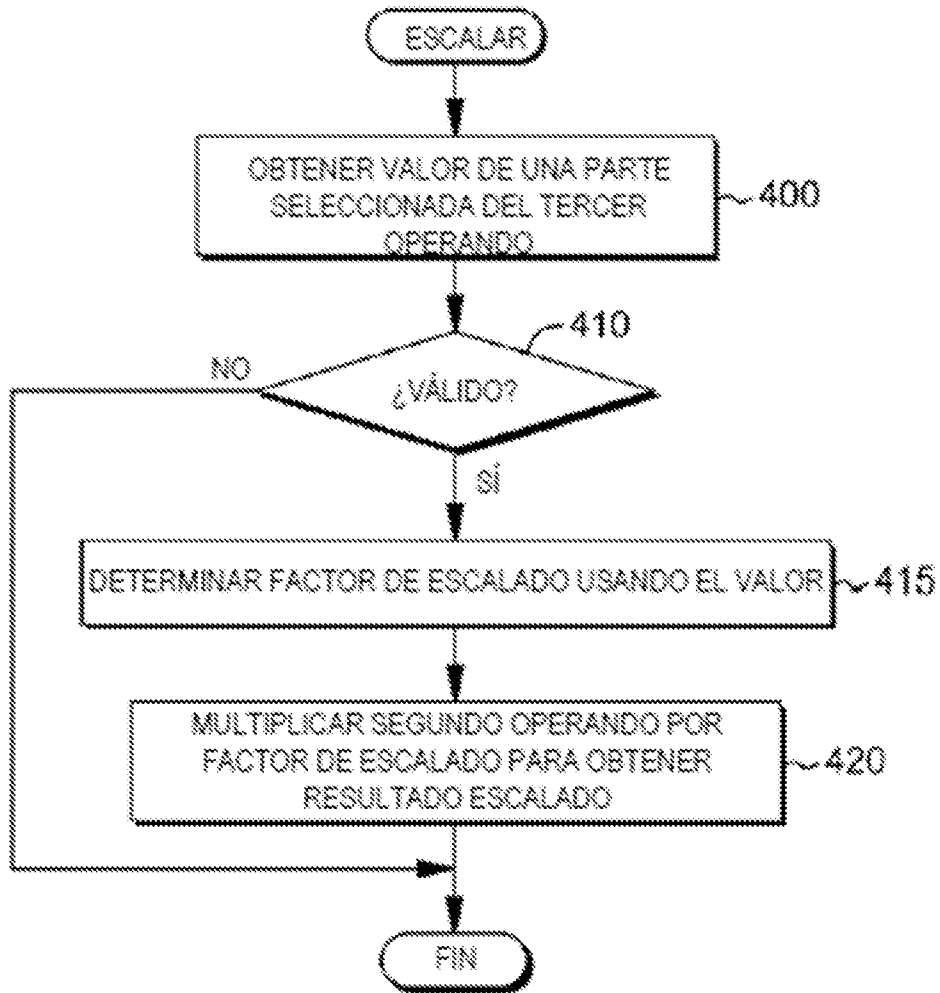


FIG. 4

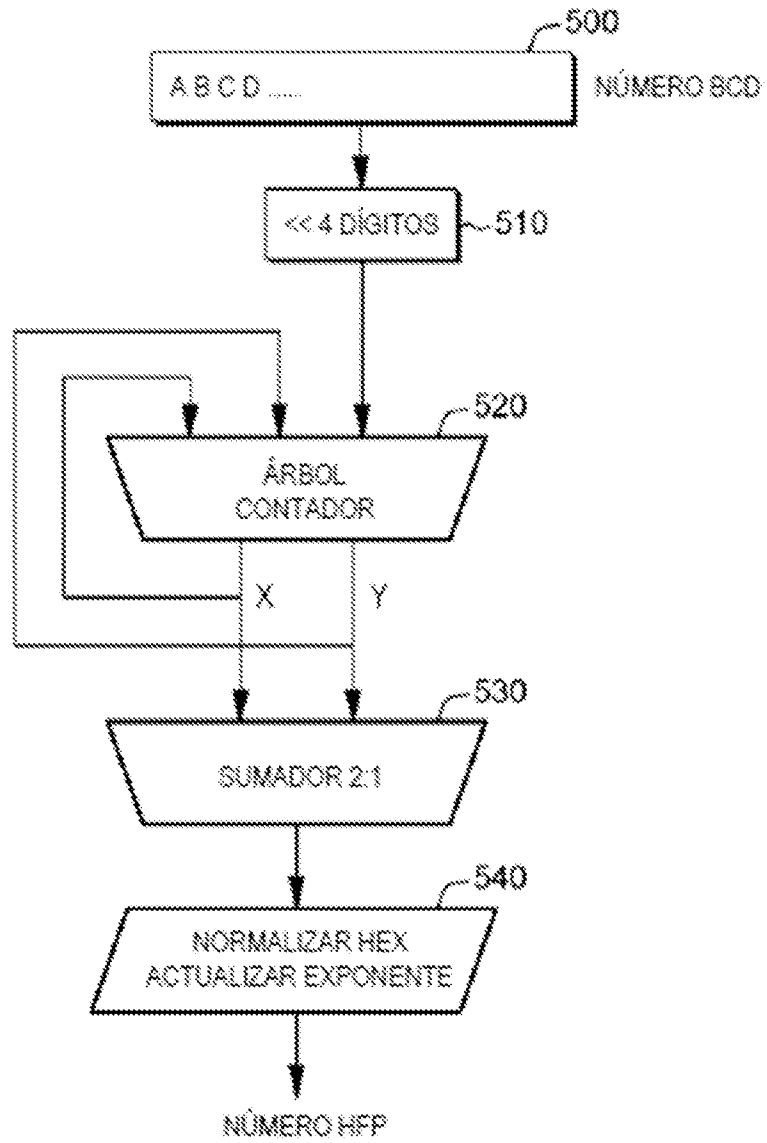


FIG. 5

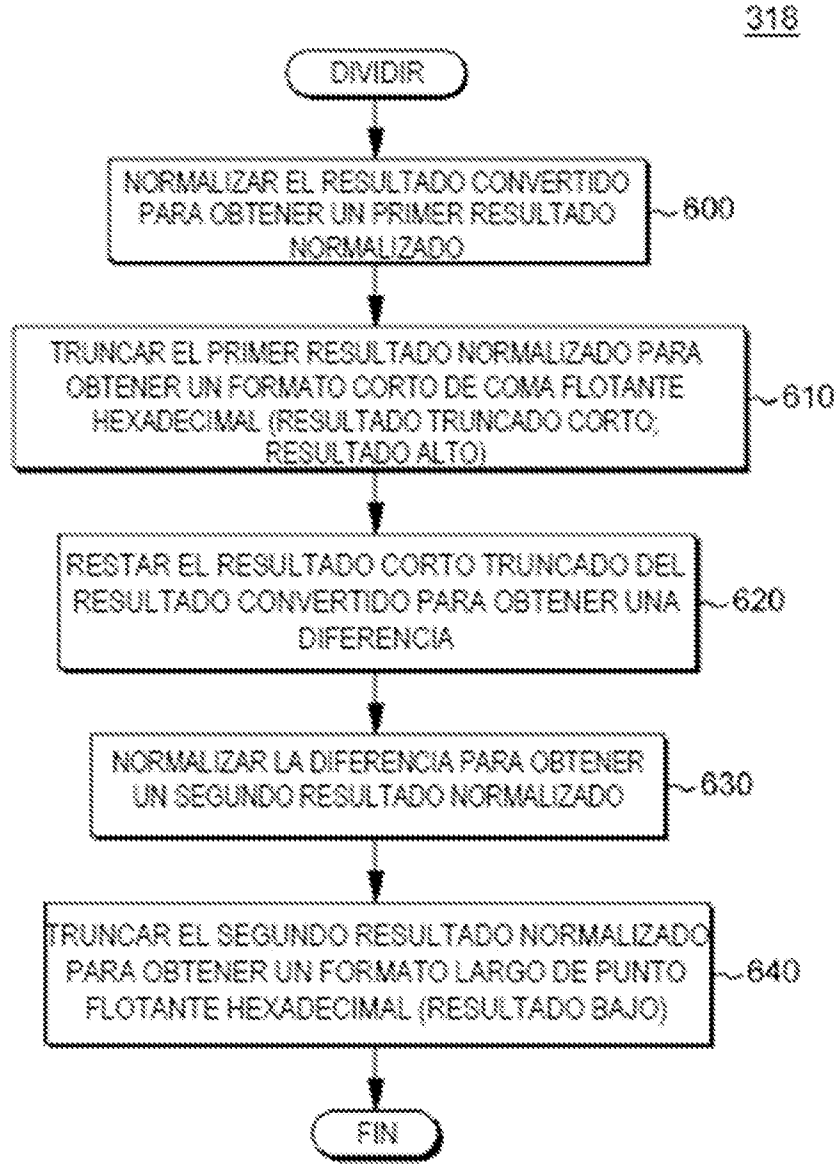


FIG. 6

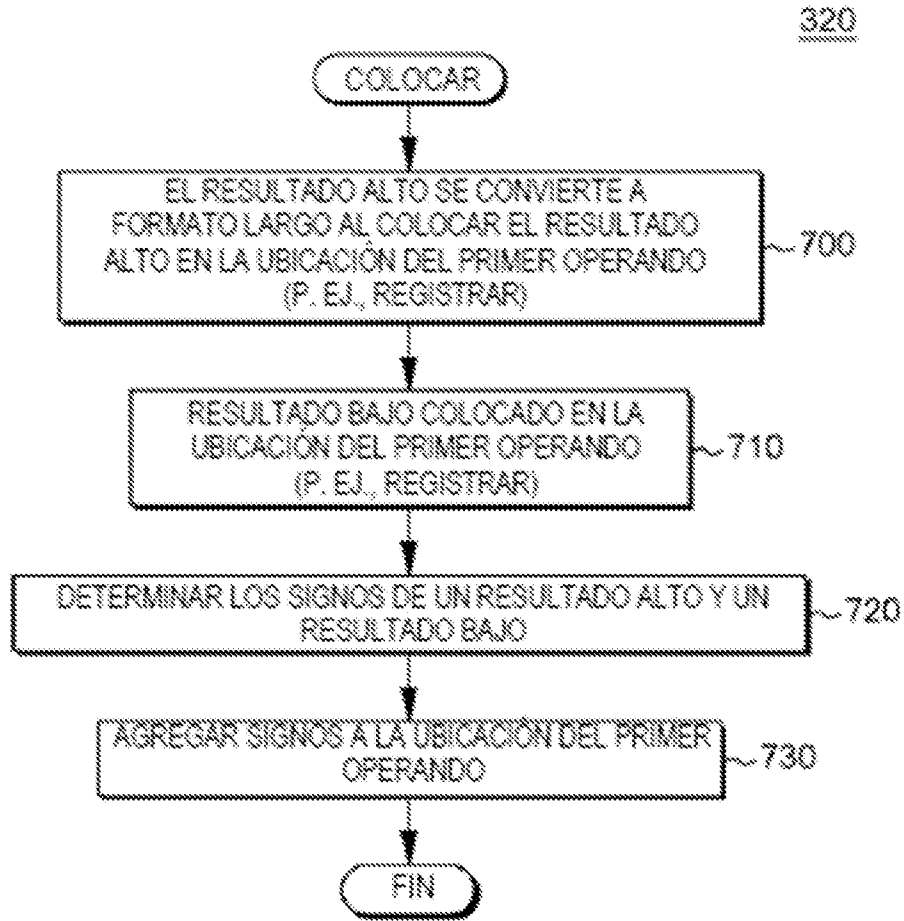


FIG. 7

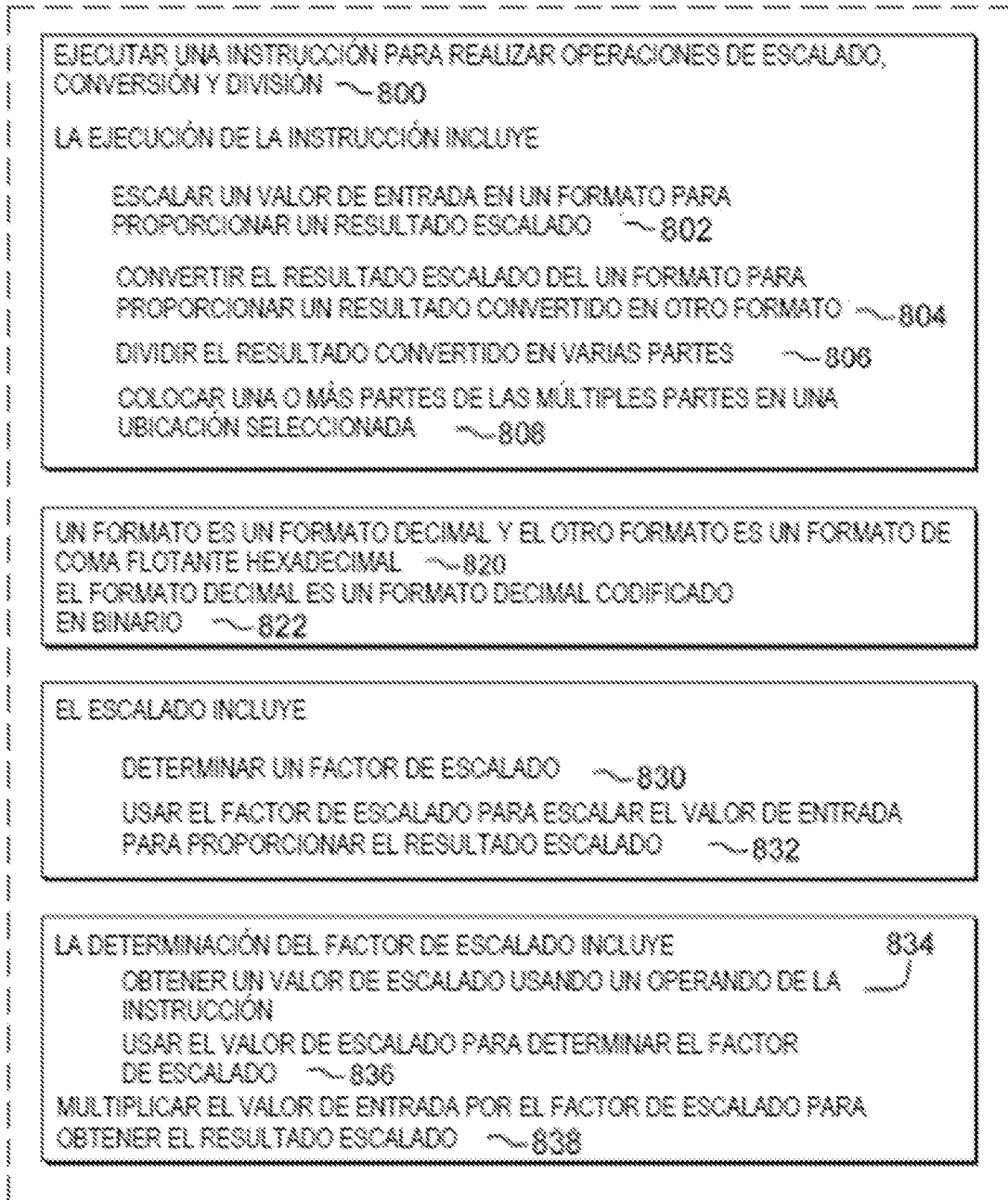


FIG. 8A

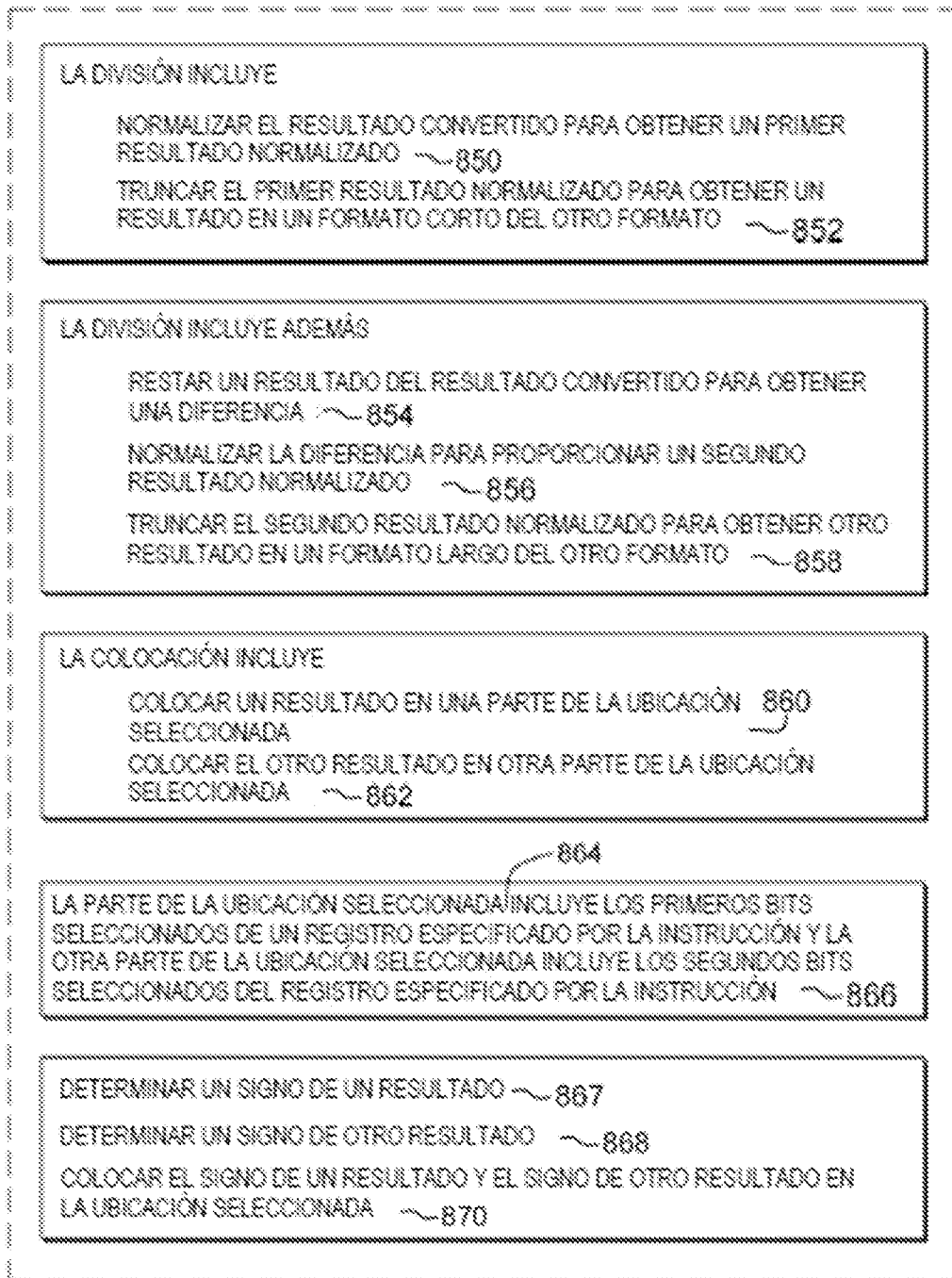


FIG. 8B

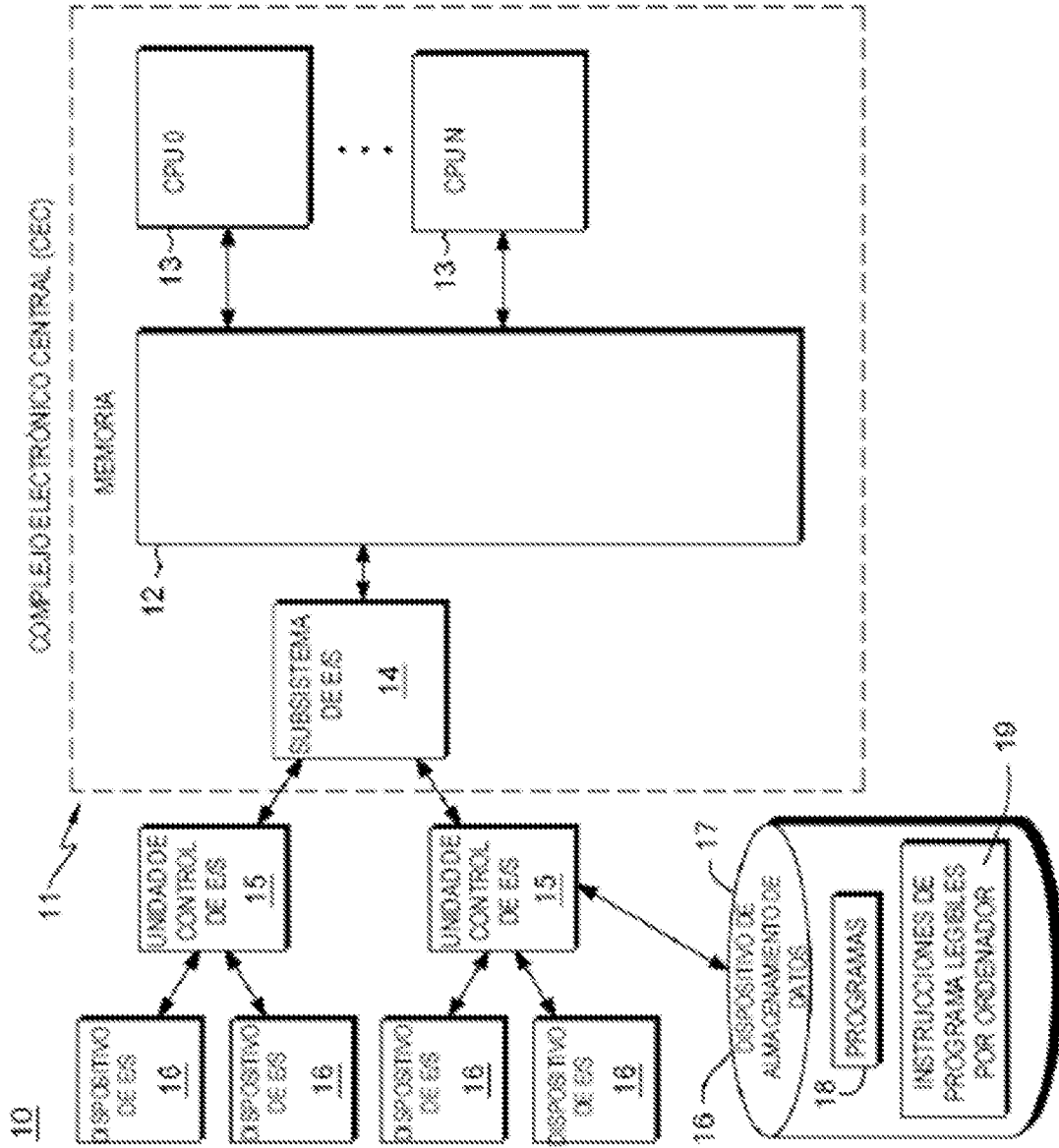


FIG. 9A

11

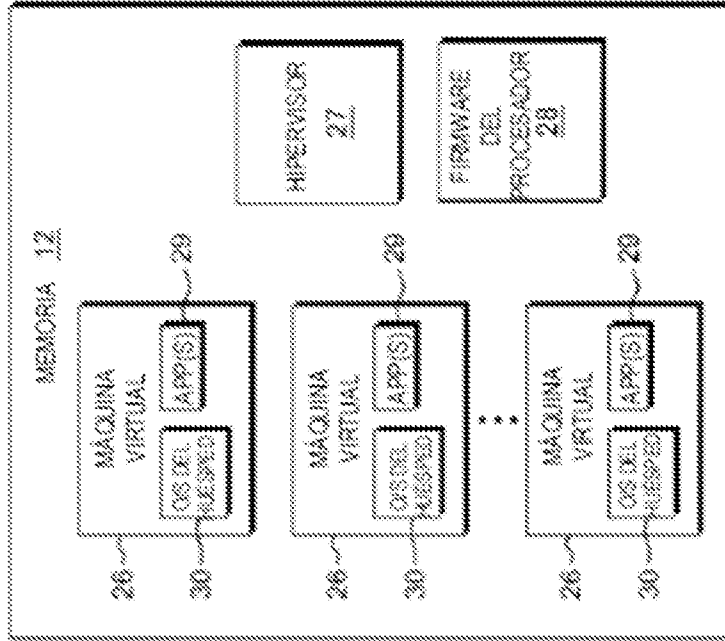


FIG. 9C

11

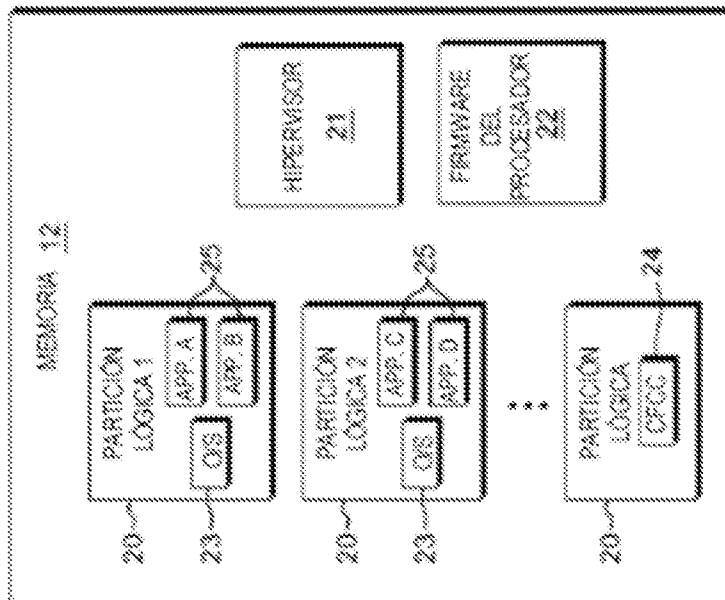


FIG. 9B

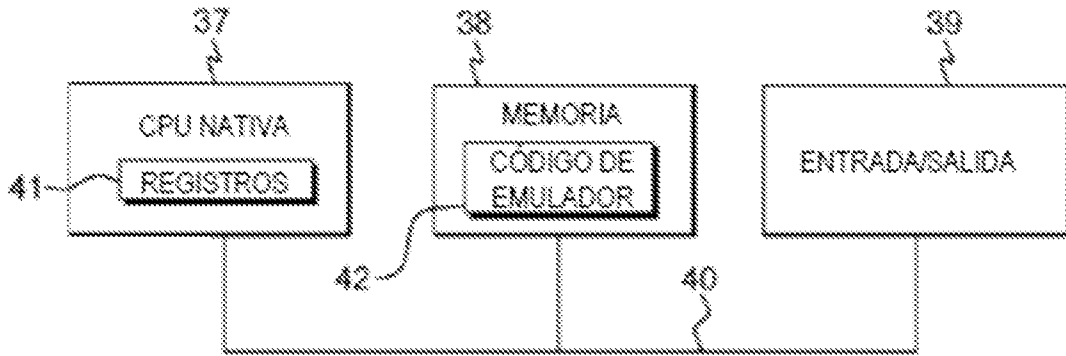


FIG. 10A

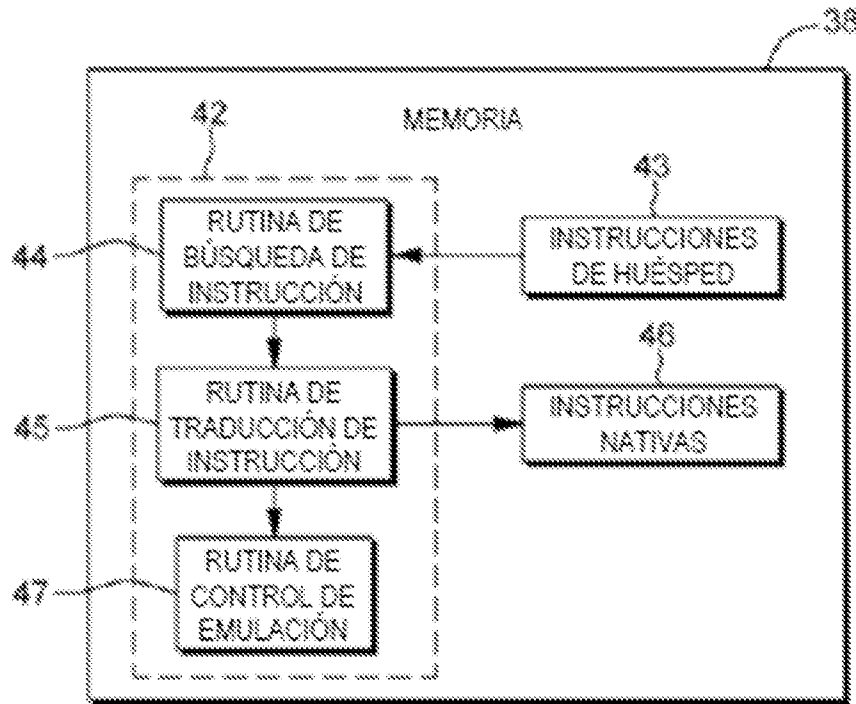


FIG. 10B

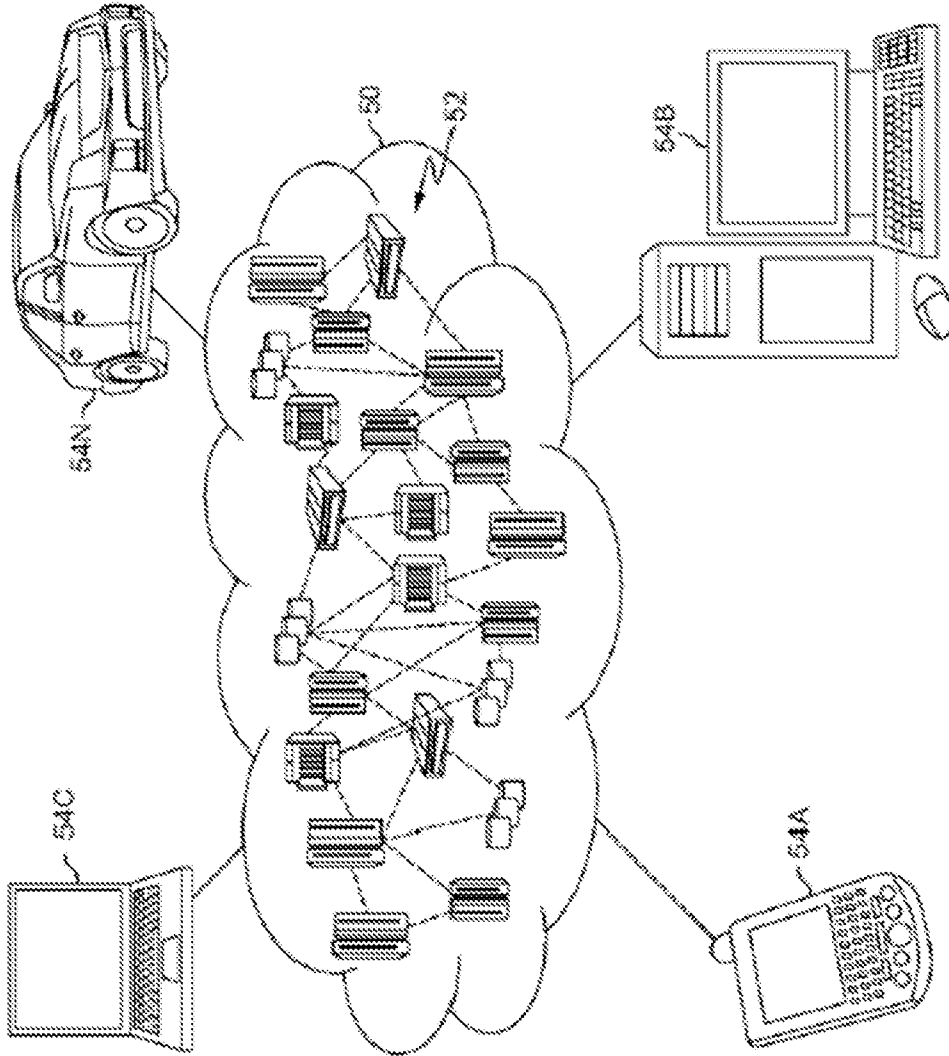


FIG. 11

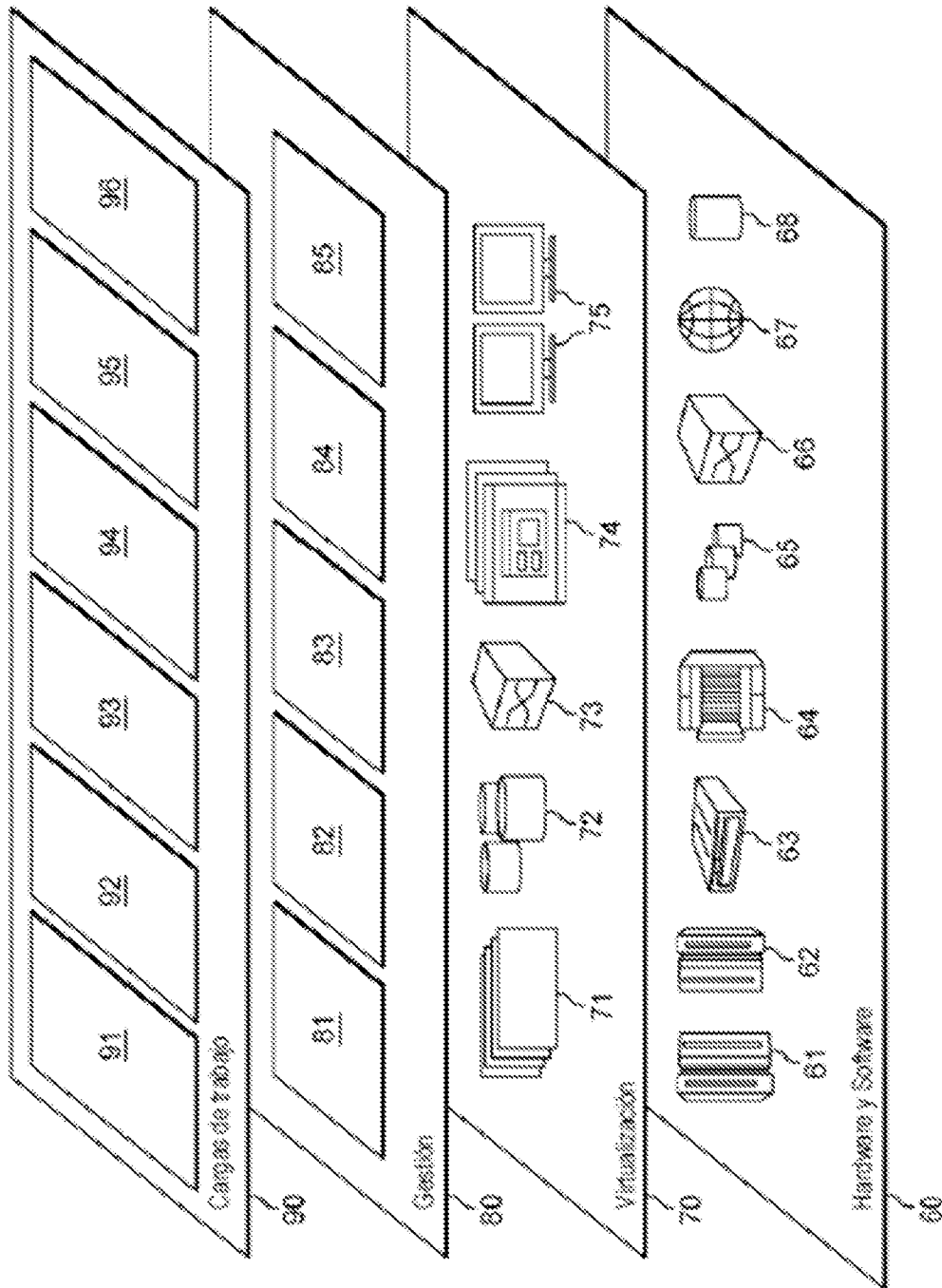


FIG. 12