

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 757 173**

51 Int. Cl.:

G06F 1/3234 (2009.01)

G06F 9/4401 (2008.01)

G06F 9/48 (2006.01)

G06F 9/50 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **18.09.2013 PCT/US2013/060242**

87 Fecha y número de publicación internacional: **18.12.2014 WO14200520**

96 Fecha de presentación y número de la solicitud europea: **18.09.2013 E 13779651 (2)**

97 Fecha y número de publicación de la concesión europea: **23.10.2019 EP 3008544**

54 Título: **Temporizadores virtuales por procesador para sistemas multiprocesador**

30 Prioridad:

13.06.2013 US 201313916621

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

28.04.2020

73 Titular/es:

**MICROSOFT TECHNOLOGY LICENSING, LLC
(100.0%)**

**One Microsoft Way
Redmond, WA 98052, US**

72 Inventor/es:

**HARTWIG, CODY;
SHERWIN, BRUCE y
WOHLGEMUTH, JASON**

74 Agente/Representante:

CARPINTERO LÓPEZ, Mario

ES 2 757 173 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

Temporizadores virtuales por procesador para sistemas multiprocesador

Antecedentes

5 En un sistema informático, un procesador usa generalmente uno o más temporizadores para gestionar los progresos de avance sobre el trabajo realizado por el procesador. El procesador establece un temporizador de modo que el temporizador genere una interrupción en un punto deseado del tiempo en el futuro.

10 En sistemas informáticos multiprocesador, que incluyen dispositivos de procesamiento con múltiples núcleos, hay generalmente dos tipos de temporizadores: temporizadores por-procesador y temporizadores de plataforma. Un temporizador de plataforma representa una instancia de un temporizador que se comparte entre procesadores y al que puede accederse desde cualquier procesador. Un temporizador por-procesador es una instancia de un temporizador que se manipula principalmente por su procesador respectivo.

15 Desde un punto de vista de la gestión de la energía hay también dos tipos de temporizadores: temporizadores que pueden desconectarse y temporizadores que siempre están conectados. Un temporizador que puede desconectarse generalmente pierde el contexto cuando hay un transitorio de la alimentación. Un temporizador que está siempre conectado mantiene al contexto a través de transitorios de la alimentación.

20 Cuando un procesador se desconecta, tal como cuando el procesador entra en un estado de inactividad de pérdida de contexto, un temporizador por-procesador que no esté siempre conectado se detendrá. Para que el procesador continúe supervisando el estado del sistema en el estado de inactividad, de modo que identifique una condición que dé como resultado pasar a activo de nuevo, se usa por ese procesador un temporizador de plataforma, siempre activo, para mantener el contexto. Sin embargo, un temporizador de plataforma puede controlarse generalmente solo por un procesador a la vez. Por ello, en el sistema multiprocesador, cuando todos los procesadores han entrado en estados de inactividad de pérdida de contexto, un procesador programa el temporizador de plataforma siempre conectado para identificar cualquier condición que dé como resultado que uno o más procesadores pasen de nuevo a activos.

25 Antecedentes adicionales pueden hallarse en:

30 Siddha Suresh ET AL, "Getting maximum mileage out of tickless", (20071231), URL: <http://www.cs.columbia.edu/~nahum/w6998/papers/ols2007v2-tickless.pdf>, que enseña que un temporizador de plataforma tal como un temporizador de difusión es compartido por un conjunto de procesadores y puede usarse para activar cualquier procesador en el conjunto usando una interrupción de temporizador APIC;

35 Mark Russinovich ET AL, "Windows Internals Part 1 6th edition", ISBN 978-0-73-567131-7, (20120321), páginas i - 276, URL: http://rogunix.com/docs/Reversing&Exploiting/WindowsInternals_6thED_part2.pdf, que enseña designar un procesador como procesador de mantenimiento del tiempo que actualiza el tiempo global del sistema y permite que otros procesadores permanezcan en estado durmiente y que el kernel use un temporizador variable que indique qué procesadores necesitan recibir un intervalo de reloj para un temporizador manual dado; o

40 Gang Wei ET AL, "The On-going Evolutions of Power Management in Xen", (20091231), URL: http://www.archive.org/files/xensummit_oracle09/xensummit_intel.pdf que enseña un mecanismo de difusión por temporizadores de plataforma de programación con un límite de tiempo esperado más próximo de todas las CPU en un estado C profundo y envía unos IPI para activar las CPU inactivas con el tiempo esperado en el temporizador ISR de plataforma.

Sumario

El presente sumario introduce conceptos seleccionados en forma simplificada que se describen adicionalmente a continuación en la descripción detallada.

La invención se define en las reivindicaciones independientes.

45 Un sistema proporciona temporizadores por-procesador virtuales mediante la virtualización de un temporizador de plataforma. Para virtualizar un temporizador de plataforma para su uso por cada procesador de forma independiente, se mantiene en la memoria una estructura de datos. La estructura de datos tiene una entrada para cada procesador que usa el temporizador de plataforma, especifica el procesador y el plazo previsto, con las entradas clasificadas por plazo previsto. En una implementación, cada procesador tiene una única entrada en la estructura de datos. Esta

50 entrada puede marcarse como válida o inválida, dependiendo del estado del procesador y de si el procesador está usando el temporizador por-procesador virtual. Cuando un procesador añade una entrada a la estructura de datos que virtualiza el temporizador de plataforma, en esta implementación, cualquier entrada previa se cancela y la nueva entrada se coloca en la estructura de datos en el orden basado en el plazo previsto. La estructura de datos puede protegerse frente a accesos simultáneos por múltiples procesadores, tal como mediante el uso de un bloqueo.

55 El temporizador de plataforma puede ser un temporizador siempre activo, y puede por ello usarse para mantener el

contexto para uno o más procesadores a través de un transitorio de la alimentación. Un procesador que use cualquier otro temporizador por-procesador puede conmutar al temporizador por-procesador virtual (derivado mediante la virtualización del temporizador de plataforma) en un transitorio de la alimentación. Cuando un procesador tiene un transitorio de alimentación, cualquier otro temporizador por-procesador que no esté siempre activo se desconecta y el procesador comienza a usar el temporizador por-procesador virtual, permitiendo a cada procesador tener transitorios de la alimentación independientemente de los otros procesadores.

En la descripción que sigue, se hace referencia a los dibujos adjuntos que forman parte de la misma, y en los que se muestran, a modo de ilustración, implementaciones de ejemplo específicas de la presente técnica. Se entiende que pueden utilizarse otras realizaciones y pueden realizarse cambios estructurales sin apartarse del ámbito de la divulgación.

Descripción de los dibujos

La Figura 1 es un diagrama de bloques de un ordenador de ejemplo en el que puede implementarse un sistema con temporizadores virtuales por-procesador.

La Figura 2 es un diagrama de bloques de una implementación de ejemplo de un sistema que usa temporizadores virtuales por-procesador.

La Figura 3 ilustra una implementación de ejemplo de una estructura de datos usada en la virtualización de un temporizador de plataforma.

La Figura 4 es un diagrama de flujo de una implementación de ejemplo de la transición a un temporizador virtual por-procesador cuando un procesador está perdiendo su estado y simultáneamente está usando un temporizador por-procesador que no está siempre conectado.

La Figura 5 es un diagrama de flujo de una implementación de ejemplo del armado de un temporizador.

La Figura 6 es un diagrama de flujo de una implementación de ejemplo de la detención de un temporizador.

La Figura 7 es un diagrama de flujo de una implementación de ejemplo del acuse de recibo de una interrupción desde un temporizador.

Descripción detallada

La sección siguiente proporciona un entorno operativo de ejemplo en el que pueden implementarse temporizadores virtuales por-procesador.

Con referencia a la Figura 1, la descripción siguiente está dirigida a proporcionar una descripción breve, general de un ordenador de finalidad general que puede usar dichos temporizadores virtuales. El ordenador puede ser cualquiera de una variedad de configuraciones de hardware informático de finalidad general o finalidad especial. Ejemplos de ordenadores bien conocidos que pueden ser adecuados incluyen, pero sin limitación, ordenadores personales, ordenadores servidores, dispositivos portátiles o manuales (por ejemplo, reproductores de medios, ordenadores notebook, teléfonos celulares, asistentes de datos personales, grabadoras de voz, tabletas), sistemas multiprocesador, sistemas basados en microprocesador, decodificadores, consolas de juego, electrónica de consumo programable, PC de red, miniordenadores, ordenadores principales, entornos informáticos distribuidos que incluyen cualquiera de los sistemas anteriores o dispositivos, y otros similares.

La Figura 1 ilustra un ejemplo de un ordenador adecuado. Esto es solamente un ejemplo de un ordenador adecuado y no se pretende que sugiera cualquier limitación como el alcance de utilización o funcionalidad de dicho ordenador.

Con referencia a la Figura 1, un ordenador 100 de ejemplo, en una configuración básica, incluye al menos una unidad 102 de procesamiento y una memoria 104. El ordenador puede incluir múltiples unidades de procesamiento y/o unidades de coprocesamiento adicionales tales como la unidad 120 de procesamiento gráfico. Dependiendo de la configuración y tipo exacto de ordenador, la memoria 104 puede ser volátil (tal como RAM), no volátil (tal como memoria ROM, flash, etc.) o una combinación de las dos. Esta configuración se ilustra en la Figura 1 por una línea 106 discontinua.

Adicionalmente, el ordenador 100 puede tener también características/funcionalidades adicionales. Por ejemplo, el ordenador 100 puede incluir también almacenamiento adicional (extraíble y/o no extraíble) que incluye, pero sin limitación, discos o cintas magnéticas u ópticas. Dicho almacenamiento adicional se ilustra en la Figura 1 mediante el almacenamiento 108 extraíble y el almacenamiento 110 no extraíble. Los medios de almacenamiento informático incluyen medios volátiles y no volátiles, extraíbles y no extraíbles implementados en cualquier procedimiento o tecnología de almacenamiento de información tales como instrucciones de programa informático, estructuras de datos, módulos de programa u otros datos. La memoria 104, almacenamiento 108 extraíble y almacenamiento 110 no extraíble son todos ellos ejemplos de un medio de almacenamiento informático. Los medios de almacenamiento informático incluyen, pero sin limitación, memoria RAM, ROM, EEPROM, flash u otra tecnología de memoria, un CD-ROM, discos versátiles digitales (DVD) u otros de almacenamiento óptico, casetes magnéticas, cinta magnética, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, o cualquier otro medio que pueda usarse para almacenar la información deseada y al que pueda accederse por parte del ordenador 100. Cualquiera de dichos medios de almacenamiento informático puede ser parte del ordenador 100.

El ordenador 100 puede contener también conexión o conexiones de comunicación 112 que permiten que el

dispositivo comunique con otros dispositivos a través de un medio de comunicación. El medio de comunicación transporta típicamente instrucciones de programa informático, estructuras de datos, módulos de programa u otros datos en una señal de datos modulada tal como una onda portadora u otro mecanismo de transporte e incluye cualquier medio de suministro de información. El término "señal de datos modulada" indica una señal que tiene una o más de sus características fijadas o cambiadas de tal manera que codifique información en la señal, cambiando de ese modo la configuración o estado del dispositivo receptor de la señal. A modo de ejemplo, y no de limitación, el medio de comunicación incluye medios cableados tales como una red por cable o conexión por cable directa, y medios inalámbricos tales como acústicos, de RF, infrarrojos y otros medios inalámbricos. Las conexiones 112 de comunicaciones son dispositivos que se interrelacionan con el medio de comunicación para transmitir datos a través y recibir datos desde medios de comunicación, tales como una interfaz de red.

El ordenador 100 puede tener varios dispositivos 114 de entrada tales como un teclado, ratón, lápiz, cámara, dispositivo de entrada táctil y otros similares. Pueden incluirse también dispositivos 116 de salida tales como una pantalla, altavoces, una impresora y otros. Todos estos dispositivos son bien conocidos en la técnica y no se comentarán extensamente en el presente documento. Diversos dispositivos de entrada y salida pueden implementar una interfaz de usuario natural (NUI), que es cualquier tecnología de interfaz que permita a un usuario interactuar con un dispositivo en una forma "natural", libre de restricciones artificiales impuestas por los dispositivos de entrada tales como ratón, teclados, mandos a distancia y similares.

Ejemplos de procedimientos NUI son aquellos que se basan en el reconocimiento de la voz, reconocimiento táctil y de escritura, reconocimiento de gestos tanto sobre pantalla como adyacentes a la pantalla, gestos en el aire, seguimiento de cabeza y ojos, voz y habla, visión, toques, gestos e inteligencia de máquina, y pueden incluir el uso de pantallas táctiles, reconocimiento de la voz y habla, comprensión de intención y objetivo, detección de movimientos de gestos usando cámaras de profundidad (tales como sistemas de cámaras estereoscópicas, sistemas de cámaras infrarrojas y otros sistemas de cámaras y combinaciones de estos), detección del movimientos de gesto usando acelerómetros o giróscopos, reconocimiento facial, pantallas tridimensionales, seguimiento de cabeza, ojo y visión, realidad aumentada inmersiva y sistemas de realidad virtual, todos los cuales proporcionan una interfaz más natural, así como tecnologías para detectar la actividad cerebral usando electrodos de detección del campo eléctrico (EEG y procedimientos relacionados).

Cada componente en este sistema que funciona en un ordenador se implementa generalmente mediante software, tal como uno o más programas informáticos, que incluyen instrucciones ejecutables por ordenador y/o instrucciones interpretadas por ordenador, tal como módulos de programa, que se procesan por un ordenador. En general, los módulos de programa incluyen rutinas, programas, objetos, componentes, estructuras de datos y otros similares que, cuando se procesan por una unidad de procesamiento, dan instrucciones a la unidad de procesamiento para realizar tareas particulares o implementar tipos de datos abstractos particulares. Este sistema informático puede ponerse en práctica en entornos informáticos distribuidos en donde las tareas se realizan por dispositivos de procesamiento remotos que están enlazados a través de una red de comunicaciones. En un entorno informático distribuido, los módulos de programa pueden localizarse tanto en medios de almacenamiento informático locales como remotos que incluyen dispositivos de almacenamiento de memoria.

Alternativamente o además, la funcionalidad descrita en el presente documento puede implementarse, al menos en parte, por uno o más componentes lógicos en hardware. Por ejemplo, y sin limitación, tipos ilustrativos de componentes lógicos en hardware que pueden usarse incluyen matrices de puertas programables en campo (FPGA), circuitos integrados de programación específica (ASIC), productos estándar de programación específica (ASSP), equipos de sistema-en-un-chip (SOC), dispositivos lógicos complejos programables (CPLD), etc.

Dado un ordenador tal como se ha descrito anteriormente, el uso de temporizadores virtuales por-procesador supone que el ordenador incluye múltiples procesadores, tanto múltiples dispositivos de procesamiento diferenciados o múltiples núcleos y similares, y sus temporizadores. El uso del término "procesador" en el presente documento está dirigido a incluir cualquier procesador lógico, incluyendo, pero sin limitación, una única instancia de un procesador en hardware, uno de múltiples procesadores, un "núcleo" en un procesador de núcleo múltiple o cualquier otra unidad de procesamiento que pueda gestionarse independientemente de otras unidades de procesamiento.

Los temporizadores virtuales por-procesador descritos en el presente documento se gestionan en general por el sistema operativo, el firmware u otro software de bajo nivel en el ordenador. En una implementación, el sistema operativo incluye un kernel que gestiona el acceso a recursos de hardware específicos a través de una capa de abstracción de hardware. Los temporizadores virtuales descritos a continuación pueden implementarse en dicha capa de abstracción de hardware. También pueden implementarse en un módulo de extensión de la capa de abstracción de hardware. Alternativamente, otros componentes del sistema operativo, firmware de un ordenador y similares pueden implementar dicha funcionalidad. El programa informático que gestiona los temporizadores virtuales por-procesador puede asociarse con un subconjunto de todos los procesadores disponibles en el ordenador.

Con referencia a la Figura 2, se describirá ahora un ejemplo de implementación de dicho sistema en el contexto de un ejemplo ilustrativo. En la Figura 2, hay cuatro procesadores 200, 202, 204 y 206 y cada uno puede tener un

temporizador 210, 212, 214 y 216 por-procesador respectivo. Cada temporizador tiene un estado (por ejemplo activado/desactivado) que puede depender, por ejemplo, del estado de la alimentación de su procesador respectivo. Si el temporizador está activado, también puede programarse para generar una interrupción en un plazo previsto especificado.

- 5 Se proporciona también un temporizador 208 de plataforma. Este temporizador de plataforma puede ser un temporizador siempre activo, es decir, un temporizador que mantiene el contexto en el caso de transitorios de la alimentación. Este temporizador de plataforma se virtualiza usando una estructura 218 de datos que se mantiene en la memoria. La estructura de datos tiene una entrada 220 para cada procesador que usa el temporizador de plataforma. Cada entrada 220 especifica el procesador y el plazo previsto. Las entradas pueden almacenarse en una
10 forma de lista enlazada que se clasifica en la lista por plazo previsto. Una implementación de ejemplo de dicha estructura de datos se describe con más detalle a continuación. Pueden usarse otras estructuras, tal como una matriz ordenada u otra estructura de datos que pueda clasificarse fácilmente.

- 15 En un plazo previsto designado para cada entrada de un procesador, el temporizador de plataforma genera una interrupción para el procesador asociado con esa entrada, y no para los otros procesadores. Dada la lista clasificada de entradas, el temporizador de plataforma se arma sucesivamente para el plazo previsto válido más próximo correspondiente a cada entrada y genera una interrupción para el procesador especificado, cuando se alcanza cada plazo previsto válido sucesivo. Cada interrupción puede encaminarse a su procesador a través de un controlador de interrupciones o puede etiquetarse para el procesador usando una interrupción basada en mensajes o puede dirigirse al procesador usando otras técnicas conocidas de encaminamiento de interrupciones.

- 20 Una implementación de ejemplo de la estructura de datos para la virtualización del temporizador de plataforma se muestra en la Figura 3 y se llama, en el presente documento, una lista 300 de temporizador. En esta implementación, cada procesador que usa este temporizador virtualizado de plataforma tiene una única entrada 302 que representa su temporizador virtual por-procesador. En la Figura 3, se muestran dos de dichas entradas, indicando que dos procesadores están usando el temporizador virtualizado de plataforma. La entrada indica un plazo
25 previsto 304 y el procesador 306 asociado. Esta entrada puede marcarse como válida o inválida, como se indica en 308, dependiendo del estado del procesador y de si el procesador está usando el temporizador virtual por-procesador. La lista 300 de temporizadores es una lista clasificada de estas entradas 302.

- Dicha lista clasificada puede implementarse, por ejemplo, como una lista enlazada, en el que cada entrada 302 también almacena un puntero 310 delantero y un puntero 312 posterior a las entradas previa y siguiente en la lista.
30 Dichos punteros no son válidos, por ejemplo, nulos cuando la entrada es inválida. Un objeto 314 de cabeza de lista identifica la entrada en el frente de la lista con un puntero delantero y el final de la lista con su puntero posterior, la entrada en la parte frontal de la lista tiene un puntero posterior que apunta al objeto de cabeza de lista. Si solo hay una entrada en la lista, ambos punteros delantero y posterior del objeto de cabeza apuntan a la única entrada. Cuando un procesador añade una entrada, en esta implementación, se elimina cualquier entrada previa de la lista clasificada. Cualquier nueva entrada se coloca en la lista en el orden basado en el plazo previsto, mediante el ajuste de los punteros delantero, posterior y de cabeza de lista en correspondencia. Puede usarse en lugar de una lista enlazada cualquier otra estructura de datos, tal como una matriz, que sea adecuada para una lista clasificada corta.

- 35 La lista 300 de temporizadores puede protegerse frente a acceso simultáneo por múltiples procesadores, tal como mediante el uso de un bloqueo 316 o similar técnica de programación. El bloqueo 316 como parte de la estructura de datos es solo una implementación posible de dicho bloqueo.
40

Usando la implementación de ejemplo mostrada en las Figuras 2 y 3, pueden realizarse los procesos siguientes durante el funcionamiento de un procesador para un transitorio entre el uso de un temporizador virtual por-procesador y el uso de cualquier otro temporizador por-procesador.

- 45 Primero, cuando un procesador está usando uno o más temporizadores por-procesador y el procesador está perdiendo el estado (tal como transitando a un estado inactivo de pérdida de contexto), se realizan las siguientes etapas para cada temporizador por-procesador antes de transitar al uso del temporizador virtual por-procesador. Si el temporizador está armado (es decir, fijado con un plazo previsto para la generación de una interrupción) como se determina en 400, entonces el temporizador virtual por-procesador se arma para el mismo plazo previsto. En particular, el plazo previsto del temporizador se registra 402 en la estructura de datos para el temporizador virtual
50 por-procesador y se marca como válido. El temporizador virtual por-procesador puede armarse 404 entonces para ese plazo previsto. El otro temporizador por-procesador se detiene 406 a continuación. Si el otro temporizador por-procesador no cumple los criterios de la etapa 400, es decir, no está armado, también puede detenerse 406 dado que no se transfiere el plazo previsto al temporizador virtual por-procesador.

- 55 En la práctica es probable que solo un temporizador por-procesador, es decir, el temporizador del sistema actual, cumpla con los criterios para ser transferido al temporizador virtual por-procesador. El soporte para múltiples temporizadores por-procesador en la estructura 300 de datos puede proporcionarse añadiendo datos que asocien cada temporizador por-procesador de cada procesador con su correspondiente plazo previsto, tal como en la entrada 302 para el procesador.

Si un procesador transita a un estado en el que está usando su propio temporizador por-procesador después del uso del temporizador virtual por-procesador, hay una variedad de posibles implementaciones. Por ejemplo, puede cancelarse el plazo previsto pendiente en el temporizador virtual por-procesador. Como otro ejemplo, el procesador puede esperar a que expire su plazo previsto con el temporizador virtual por-procesador y transitar a continuación al uso del otro temporizador por-procesador.

La Figura 5 describe el proceso de armar el temporizador virtual por-procesador para un procesador. El procesador adquiere 500 un bloqueo en la lista de temporizadores. Si la entrada actual para el procesador es válida, tal como se determina en 502, entonces se elimina 504 la entrada actual para el procesador de la lista clasificada de temporizadores. La entrada para el procesador actual se marca 506 como válida. El plazo previsto en esta entrada se fija 508 al plazo previsto registrado. La entrada para el procesador actual se coloca 510 entonces en la lista clasificada de entradas, en el orden de su plazo previsto. El temporizador físico se redirige 512 entonces para apuntar a su interrupción hacia el procesador con la entrada en la cabeza de la lista de temporizadores. El temporizador físico arma 514 con el plazo previsto indicado por la entrada en la cabeza de la lista de temporizadores. El bloqueo en la lista de temporizadores se libera 516 a continuación.

La Figura 6 describe el proceso de detener el temporizador virtual por-procesador para un procesador. El procesador consigue 600 un bloqueo en la lista de temporizadores. La entrada para este procesador se elimina 602 de la lista de entradas clasificadas para la lista de temporizadores. La entrada para este procesador se establece 604 como inválida. Si las entradas permanecen en la lista de temporizadores, como se determina en 605, entonces el temporizador físico se establece 606 para dirigir la interrupción hacia el procesador con la entrada en la cabeza de la lista de temporizadores y el temporizador físico se arma para el plazo previsto en esa entrada. Si no hay entradas en la lista de temporizadores, tal como se determina en 605, el temporizador de plataforma se detiene 609 para evitar que tengan lugar interrupciones espurias. Después de detener o armar el temporizador físico con la entrada en la cabeza de la lista, se libera 610 a continuación el bloqueo en la lista de temporizadores.

La Figura 7 describe el proceso de acuse de recibo de una interrupción desde el temporizador por-procesador virtual por parte de un procesador. El procesador consigue 700 el bloqueo en la lista de temporizadores. La entrada para el procesador se establece entonces 704 como inválida, y se elimina la primera entrada en la lista clasificada. Se acusa recibo 706 entonces de la interrupción desde el temporizador físico. Si la lista de entradas clasificadas en la lista de temporizadores no está vacía, como se determina en 708, entonces el temporizador físico se redirige 710 al procesador que tiene la entrada en la cabeza de la lista clasificada y el temporizador físico se arma para el plazo previsto de esa entrada. Si no hay entradas restantes en la lista de temporizadores, como se determina en 708, el temporizador de plataforma se detiene 709 para evitar que tengan lugar interrupciones espurias. Después de detener o armar el temporizador de plataforma, se libera 714 el bloqueo en la lista de temporizadores.

Además de las variaciones que permiten múltiples temporizadores para cada procesador, hay varias variaciones de la estructura de datos y procesos usados para virtualizar el temporizador de plataforma para proporcionar un temporizador virtual por-procesador. En general, la estructura de datos identifica un procesador y un plazo previsto para que se genere cada interrupción. Los detalles de cómo manipular esta estructura de datos para armar, detener y acusar recibo y otras operaciones, dependen de la implementación.

Además de lo anterior, la selección del temporizador virtual de plataforma para dar soporte a un temporizador virtual por-procesador puede realizarse en el tiempo de arranque del sistema. En general, la virtualización de un temporizador de plataforma depende de la detección de la presencia de un temporizador de plataforma. Dada la presencia de uno o más procesadores que usen temporizadores virtuales por-procesador, pueden hacerse entradas para estos procesadores en la estructura de datos asociada con el temporizador de plataforma para definir los temporizadores virtuales por-procesador.

La invención tampoco está limitada a la virtualización de un temporizador de plataforma, un temporizador por-procesador puede virtualizarse por su procesador para realizar múltiples temporizadores virtuales por-procesador.

La invención tampoco está limitada a un sistema en el que los temporizadores por-procesador están en otra forma disponibles. Los temporizadores virtuales por-procesador pueden generarse en un sistema que tenga solo temporizadores de plataforma, para permitir que diferentes procesadores y el software de alto nivel los apile para acceder a un temporizador por-procesador basado en un temporizador de plataforma.

Si el temporizador de plataforma es un temporizador siempre activo y otros temporizadores por-procesador no están siempre activos, entonces puede usarse un temporizador virtual por-procesador basado en el temporizador de plataforma para gestionar los transitorios de la alimentación. En particular, cuando un procesador entra en un estado de inactividad de pérdida de contexto, puede transitar de cualquier temporizador por-procesador a un temporizador virtual por-procesador.

En algunas implementaciones, se consulta la hora actual para calcular un valor con el que armar el temporizador físico. Por ello, cada operación de armado puede incurrir en una sobrecarga asociada con la consulta de la hora actual. Para gestionar el número de consultas, la adición de entradas a la cola en la lista de temporizadores puede controlarse de una diversidad de formas. En la implementación arriba observada, solo se mantiene una entrada por

procesador en la estructura de datos y se mantiene la entrada por procesador más recientemente añadida. Es posible tener múltiples entradas por procesador. También, la selección entre las entradas, si se descarta una o más entradas, puede ser una función del plazo previsto, tal como que se mantenga el plazo previsto más temprano o más tardío. La estrategia de control particular usada para actualizar la cola puede ser un ajuste del sistema.

- 5 Teniendo una implementación de temporizadores virtuales por-procesador, ejemplos de los cuales se han descrito anteriormente, múltiples procesadores pueden gestionar transitorios de la alimentación independientemente entre sí aun compartiendo un temporizador de plataforma.

Cualquiera o todas las realizaciones alternativas anteriormente mencionadas descritas en el presente documento pueden usarse en cualquier combinación deseada para formar realizaciones híbridas adicionales. Debería entenderse que la materia objeto definida en las reivindicaciones adjuntas no está necesariamente limitada a las implementaciones específicas descritas anteriormente. Las implementaciones específicas descritas anteriormente se divulgan solamente como ejemplos.

10

REIVINDICACIONES

1. Un ordenador que comprende:

una pluralidad de procesadores (202-206); y

un temporizador de plataforma (208) que mantiene el contexto durante transitorios de la alimentación;

incluyendo el ordenador instrucciones de programa informático que cuando se ejecutan por uno de los procesadores (202-206) el ordenador se configura de modo que se virtualiza el temporizador (208) de plataforma para definir, para cada procesador (202-206) de la pluralidad de procesadores (202-206), un temporizador virtual por-procesador para el procesador (202-206) teniendo una estructura (218) de datos disponible para la pluralidad de procesadores (202-206), especificando la estructura (218) de datos un plazo previsto para una interrupción para el procesador (202-206) desde el temporizador (208) de plataforma y recibiendo el procesador (202-206) la interrupción,

en el que, cuando uno de la pluralidad de procesadores (202-206), que usa un primer temporizador (210-216) por-procesador, se somete a un transitorio de alimentación de pérdida de contexto, dicho procesador (202-206) se configura para armar su temporizador virtual por-procesador asociado para proporcionar interrupciones a dicho procesador (202-206) después del transitorio de la alimentación, y

en el que el temporizador virtual por-procesador asociado se arma para un plazo previsto del primer temporizador (210-216) por-procesador mediante la especificación del plazo previsto para una interrupción desde el temporizador (208) de plataforma y el primer temporizador (210-216) por-procesador se detiene después de que el temporizador virtual por-procesador asociado se arme para el plazo previsto.

2. El ordenador de la reivindicación 1, incluyendo la estructura (218) de datos una o más entradas (220), especificando cada entrada (220) un plazo previsto para una interrupción y un procesador (202-206) para recibir la interrupción, estando las entradas (220) clasificadas por plazo previsto.

3. El ordenador de la reivindicación 2, en el que la estructura (218) de datos es una lista de temporizadores que comprende una entrada para cada procesador (202-206), en el que una entrada (220) es válida si el procesador (202-206) está usando el temporizador virtual por-procesador y no es válida si el procesador (202-206) no está usando el temporizador virtual por-procesador, y una cola de entradas (220) válidas clasificadas por plazo previsto.

4. El ordenador de la reivindicación 3, en el que las instrucciones de programa informático configuran el ordenador para armar el temporizador (208) de plataforma mediante el establecimiento como válida de la entrada para un procesador (202-206), y el establecimiento del plazo previsto para la entrada (220).

5. El ordenador de la reivindicación 3, en el que las instrucciones de programa informático configuran el ordenador para detener el temporizador virtual por-procesador para un procesador (202-206) dado, mediante el establecimiento como no válida de la entrada para el procesador (202-206) y para rearmar el temporizador (208) de plataforma si quedan entradas (220) válidas en la lista de temporizadores.

6. El ordenador de la reivindicación 3, en el que las instrucciones de programa informático configuran el ordenador para acusar recibo de la interrupción desde el temporizador (208) de plataforma.

7. Un artículo de fabricación que comprende:

un medio de almacenamiento informático;

instrucciones de programa informático almacenadas en el medio de almacenamiento informático que, cuando se procesan por un procesador (202-206) en el ordenador, teniendo el ordenador una pluralidad de procesadores (202-206), configura el ordenador para que comprenda:

un temporizador (208) de plataforma que mantiene el contexto durante transitorios de la alimentación y se virtualiza para definir, para cada procesador (202-206) de la pluralidad de procesadores (202-206), un temporizador por-procesador virtual para el procesador (202-206) teniendo una estructura (218) de datos disponible para la pluralidad de procesadores (202-206), especificando la estructura (218) de datos un plazo previsto de una interrupción para el procesador (202-206) desde el temporizador (208) de plataforma y recibiendo el procesador (202-206) la interrupción,

en el que, cuando uno de la pluralidad de procesadores (202-206), que usa un primer temporizador (210-216) por-procesador, se somete a un transitorio de alimentación con pérdida de contexto, dicho procesador (202-206) se configura para armar su temporizador virtual por-procesador asociado para proporcionar interrupciones a dicho procesador (202-206) después del transitorio de la alimentación, y

en el que el temporizador virtual por-procesador se arma para un plazo previsto del primer temporizador (210-216) por-procesador mediante la especificación del plazo previsto para una interrupción desde el temporizador (208) de plataforma y el primer temporizador (210-216) por-procesador se detiene después de que el temporizador virtual por-procesador se arme para el plazo previsto.

8. Un procedimiento implementado en ordenador para un ordenador que tiene una pluralidad de procesadores (202-206) y un temporizador (208) de plataforma que mantiene el contexto durante transitorios de la alimentación, comprendiendo el procedimiento:

- definir una estructura (218) de datos, especificando la estructura (218) de datos un plazo previsto de una interrupción para un procesador (202-206) desde el temporizador (208) de plataforma y recibiendo el procesador (202-206) la interrupción;
- 5 realizar, para cada procesador (202-206) de la pluralidad de procesadores (202-206), la estructura (218) de datos disponible como un temporizador virtual por-procesador;
- uno de la pluralidad de procesadores (202-206), que usa un primer temporizador (210-216) por-procesador y que se somete a un transitorio de la alimentación con pérdida de contexto arma su temporizador (208) virtual por-procesador asociado mediante la especificación del plazo previsto para una interrupción desde el temporizador (208) de plataforma, siendo el plazo previsto un plazo previsto del primer temporizador (210-216) por-procesador,
- 10 en el que el primer temporizador (210-216) por-procesador se detiene después de que el temporizador (218) virtual por-procesador se arme con el plazo previsto; y
- generando el temporizador (208) de plataforma la interrupción para dicho procesador (202-206) en el plazo previsto.
- 15 9. El proceso de la reivindicación 8, incluyendo la estructura (218) de datos una o más entradas (220), especificando cada entrada (220) un plazo previsto para una interrupción y un procesador (202-206) para recibir la interrupción, estando clasificadas las entradas (220) por plazo previsto.

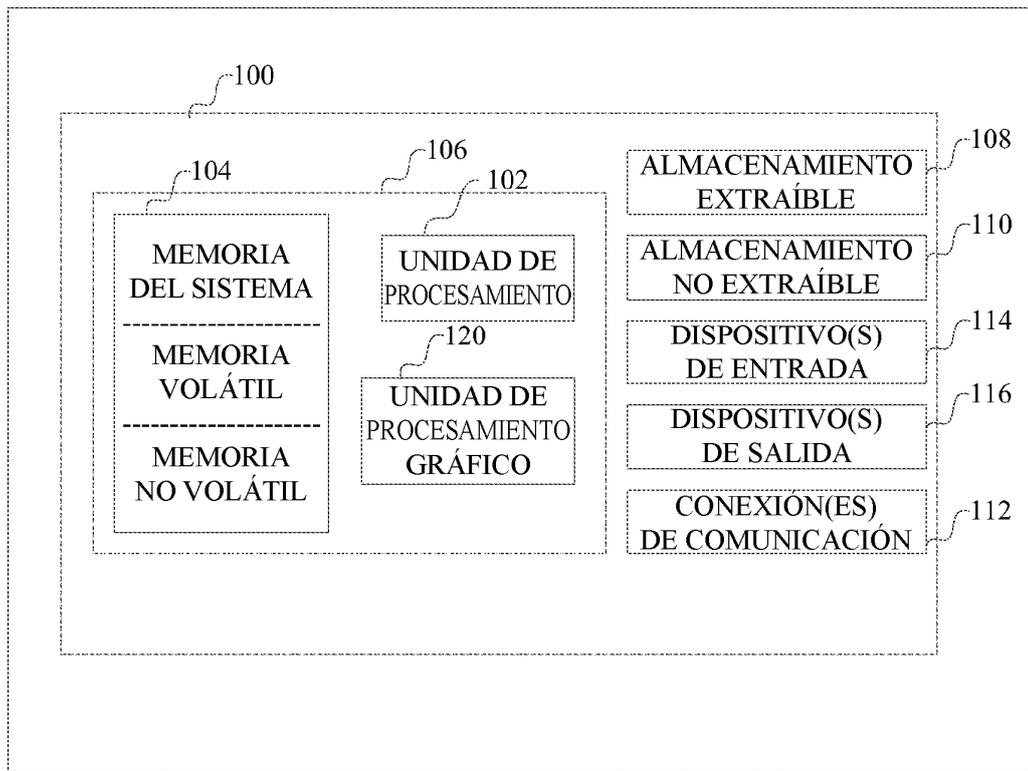


FIG. 1

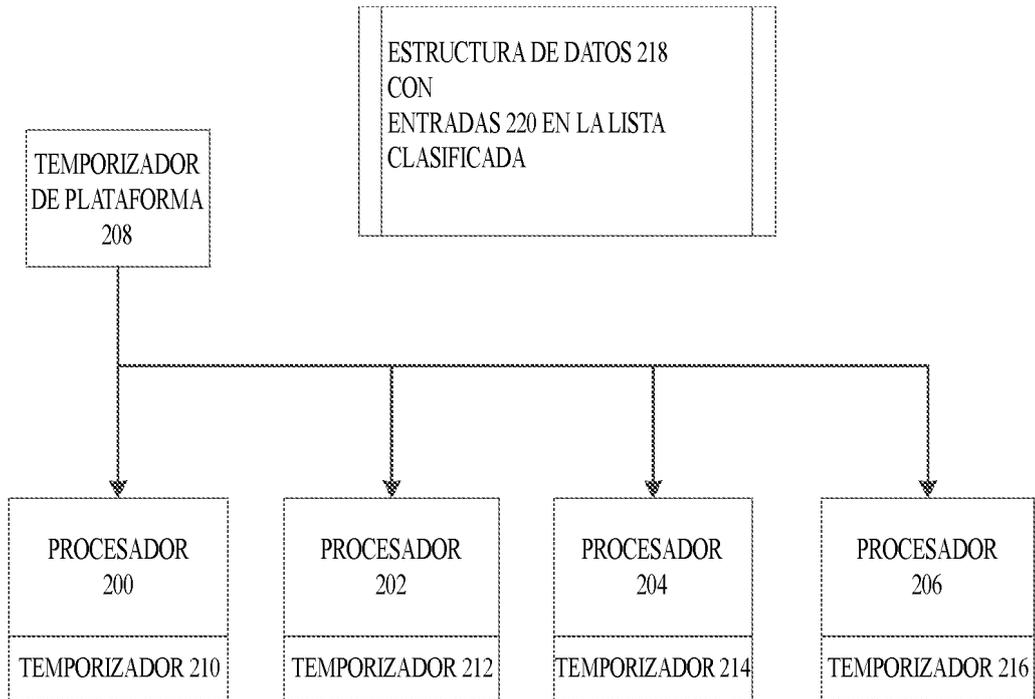


FIG. 2

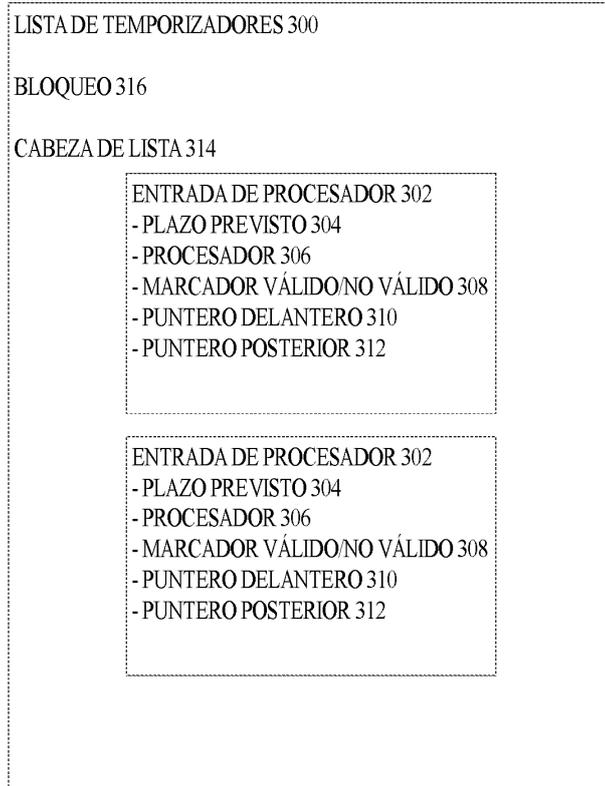


FIG. 3

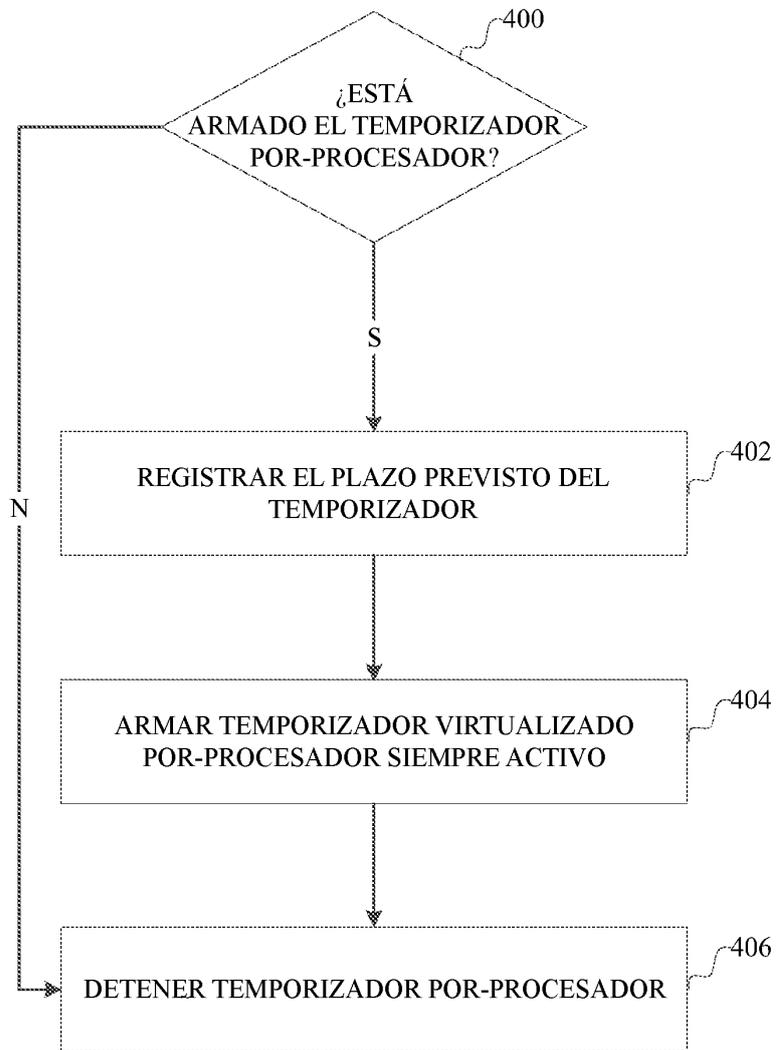


FIG. 4

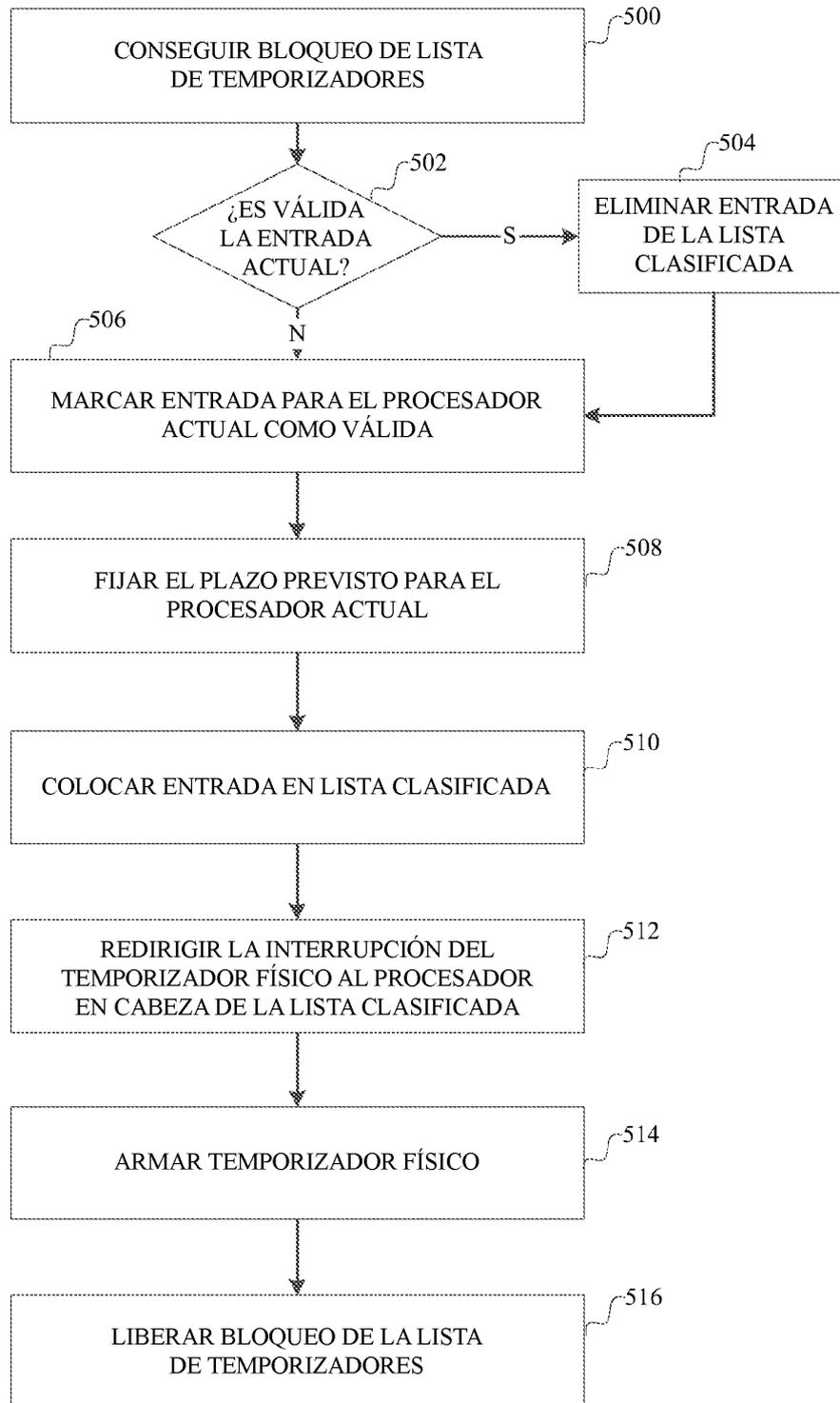


FIG. 5

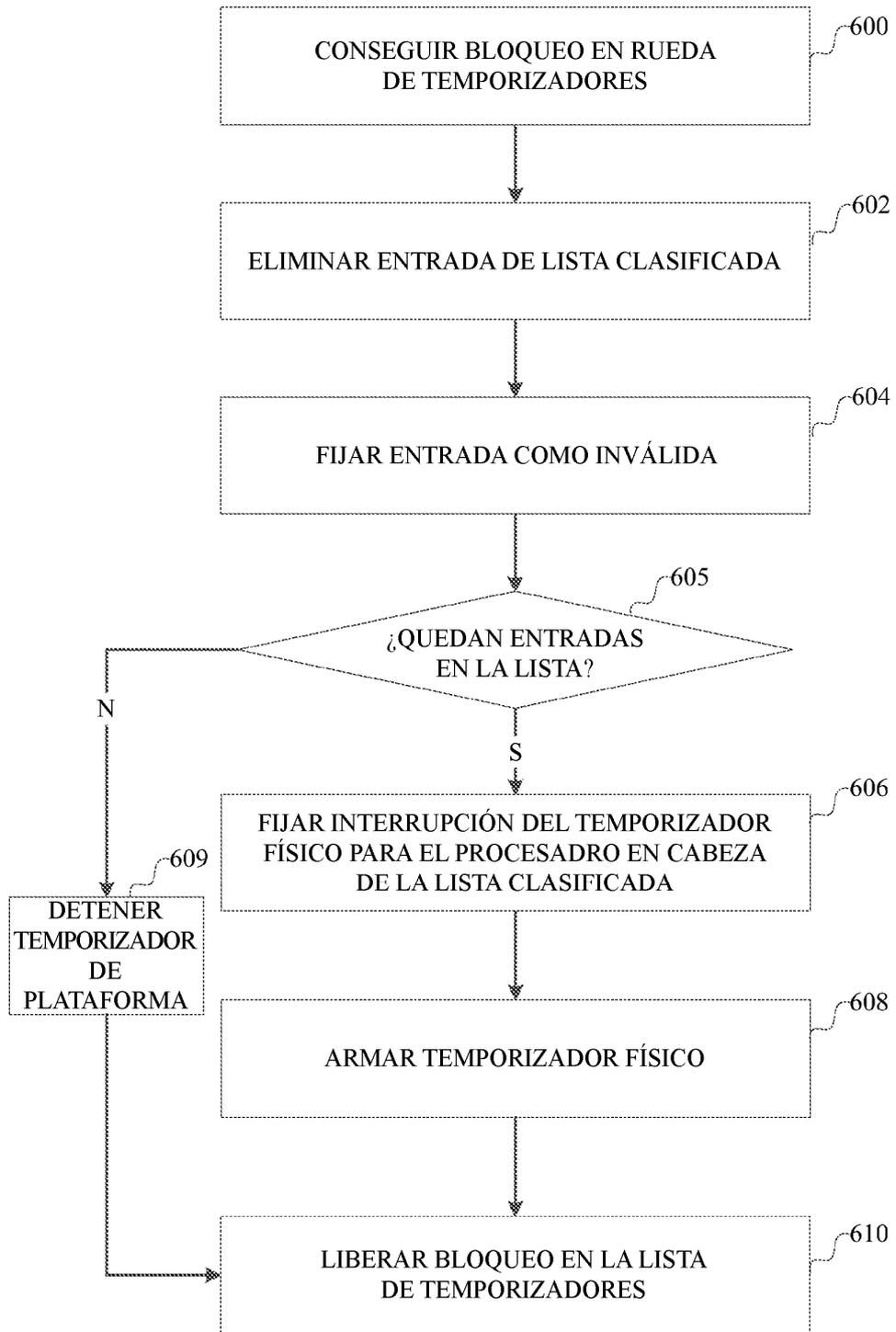


FIG. 6

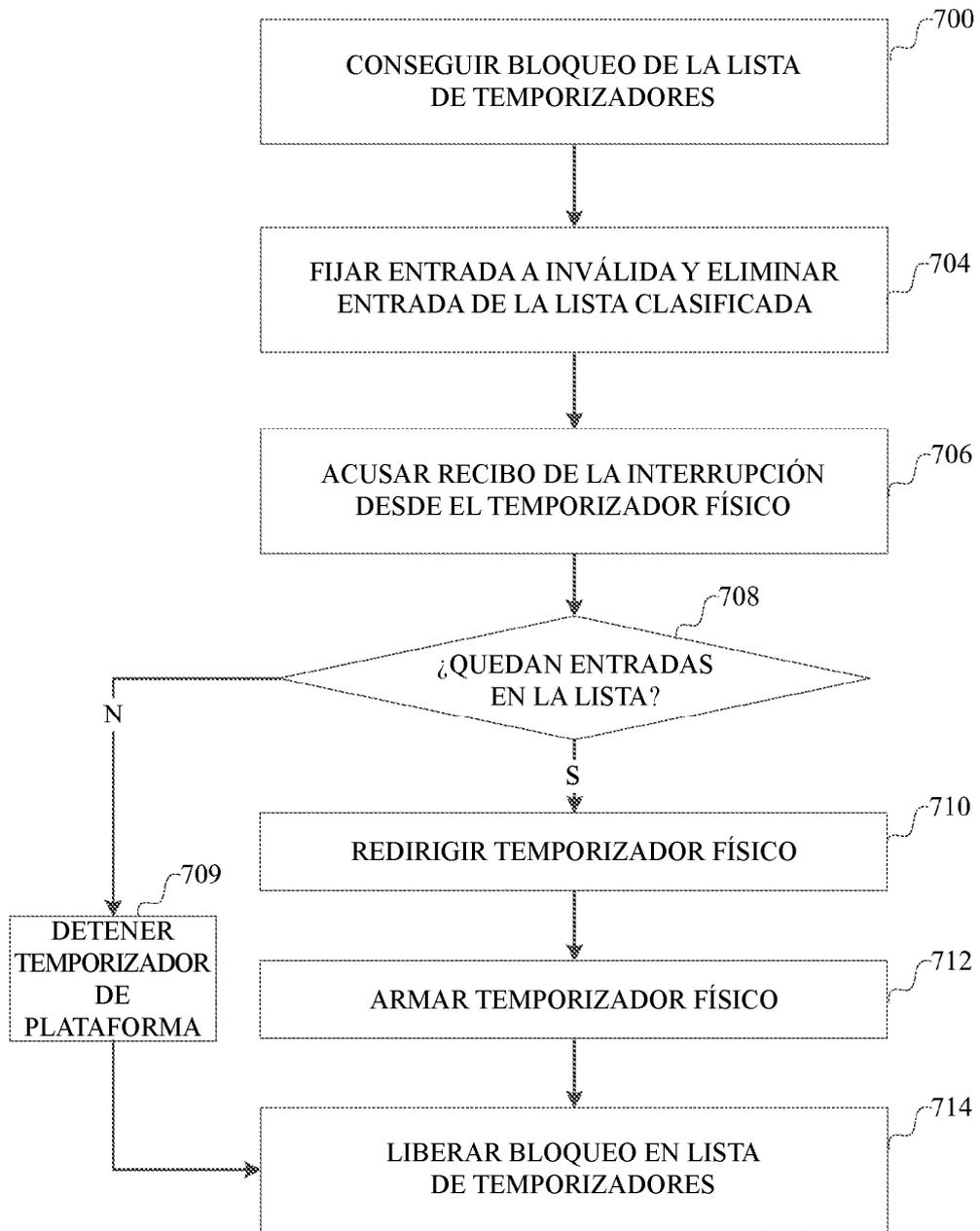


FIG. 7