

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 701 739**

51 Int. Cl.:

G06F 9/50 (2006.01)

G06F 1/32 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **30.11.2009** **E 09252700 (1)**

97 Fecha y número de publicación de la concesión europea: **26.09.2018** **EP 2207092**

54 Título: **Reasignación de hilos basada en software para el ahorro de energía**

30 Prioridad:

09.12.2008 US 316014

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

25.02.2019

73 Titular/es:

**INTEL CORPORATION (100.0%)
2200 Mission College Boulevard
Santa Clara, CA 95054, US**

72 Inventor/es:

SONG, JUSTIN J.

74 Agente/Representante:

LEHMANN NOVO, María Isabel

ES 2 701 739 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

Reasignación de hilos basada en software para el ahorro de energía

Antecedentes

5 La gestión térmica y energética se está volviendo más compleja que nunca en todos los segmentos de los sistemas basados en ordenadores. Mientras que en el dominio de los servidores es el coste de la electricidad el que impulsa la necesidad de sistemas de energía baja, en sistemas móviles la vida de la batería y las limitaciones térmicas hacen estos problemas relevantes. La gestión de un sistema basado en ordenadores para un rendimiento máximo con un consumo de potencia mínimo se puede lograr reduciendo la energía de todo o parte del sistema informático cuando esté inactivo o, por lo contrario, no sea necesario.

10 Un estándar de gestión energética para ordenadores es el estándar de Interfaz Avanzada de Configuración y Energía (ACPI), p. ej., Rev. 3.0b, publicado el 10 de octubre de 2006, que define una interfaz que permite al sistema operativo (SO) controlar elementos de hardware. Muchos sistemas operativos modernos utilizan el estándar ACPI para llevar a cabo la gestión térmica y energética de los sistemas informáticos. Una implementación ACPI permite a un núcleo estar en diferentes estados de ahorro de energía (también denominados estados de energía baja o estados de inactividad) denominados en general como los estados así llamados C1 a Cn.

15 Cuando el núcleo está activo, este ejecuta un estado así llamado C0, pero cuando el núcleo está inactivo, el SO intenta mantener un equilibrio entre la cantidad de energía que puede ahorrar y la sobrecarga de entrar y salir a/de un estado dado. Por tanto, C1 representa el estado de energía baja que tiene el mínimo ahorro de energía pero que se puede activar y desactivar casi de manera inmediata (por tanto, se denomina estado de “energía baja superficial” o de “inactividad superficial”), mientras que los estados de energía baja profundos (p. ej., C3, C6 o C7) representan un estado de energía donde el consumo de potencia estacionario puede ser despreciable, dependiendo de la implementación de silicio, aunque el instante para entrar en este estado y responder a una actividad (es decir, de vuelta a activo C0) es relativamente largo. Cabe destacar que diferentes procesadores pueden incluir diferentes números de estados C de núcleo, cada uno asignado a un estado C de ACPI. Es decir, los estados C de múltiples núcleos se pueden asignar al mismo estado C de ACPI.

20 La política de estados C de los SO actuales puede no proporcionar los resultados de rendimiento más eficientes debido a que la política de estados C de los SO actuales puede no considerar las actividades de otros núcleos en el mismo paquete. En particular, la política de estados C de los SO actuales puede no beneficiarse de las eficiencias que se podría obtener mediante una gestión y un seguimiento más cercanos de los estados de energía de los diversos hilos que se ejecutan en diferentes núcleos del mismo paquete. Es decir, un hilo de hardware de un núcleo puede estar en un estado de energía baja profundo, mientras que otro hilo de hardware del núcleo puede estar activo. De acuerdo con la política de estados C de los SO actuales, un núcleo no puede entrar en un estado de energía baja profundo a menos que todos los hilos en el núcleo estén en un estado de energía baja profundo. Si múltiples núcleos experimentan esta condición, entonces ninguno de los núcleos puede entrar en un estado de energía baja profundo (incluso si múltiples hilos de hardware están inactivos).

25 El documento US 2008/0250260 A1 analiza un aparato de procesamiento de información que incluye una CPU, que incluye una pluralidad de procesadores de instrucciones, una unidad de monitorización que monitoriza un entorno operativo de suministro de energía y una unidad de ahorro de energía que controla el número de procesadores de instrucciones operativos dispuestos en la CPU, de acuerdo con el entorno operativo de suministro de energía obtenido mediante la monitorización con la unidad de monitorización.

Descripción breve de los dibujos

La figura 1 es un diagrama de bloques que ilustra al menos un ejemplo de un sistema para llevar a cabo las técnicas expuestas.

45 La figura 2 es un diagrama de bloques de flujo de datos que representa unas vistas anterior y posterior de un ejemplo de reasignación de acuerdo con al menos un ejemplo.

La figura 3 es un diagrama de flujo de control y datos que ilustra al menos un ejemplo de un método para llevar a cabo el salto de hilos basado en software para el ahorro de energía en un sistema de muestra que incluye dos núcleos de hilos dobles.

50 La figura 4 es un diagrama de flujo que ilustra al menos una realización de un método para llevar a cabo el salto de hilos basado en software para el ahorro de energía.

La figura 5 es un diagrama de bloques de un sistema de acuerdo con al menos una realización de la presente invención.

La figura 6 es un diagrama de bloques de un sistema de acuerdo con al menos otra realización de la presente invención.

La figura 7 es un diagrama de bloques de un sistema de acuerdo con al menos otra realización de la presente invención.

Descripción detallada

5 La invención se define de acuerdo con el método de la reivindicación independiente 1, el sistema de la reivindicación independiente 8 y las instrucciones legibles por máquina de la reivindicación 15, a las cuales el lector debería hacer referencia ahora. Las realizaciones específicas se definen en las reivindicaciones dependientes.

10 Los ejemplos llevan a cabo con precisión y en tiempo real una reasignación lógica basada en software de hilos a núcleos, lo que "intercambia" de manera efectiva los hilos inactivos al mismo núcleo, de modo que el núcleo pueda entrar en un estado de energía baja. De manera más específica, un sistema operativo puede llevar a cabo una reasignación lógica y extracción de tareas para extraer tareas desde un contexto de hilo de hardware en un núcleo hacia otro contexto de hilo de hardware en un núcleo diferente. Si un contexto de hilo de hardware está listo para realizar la transición a un estado de energía baja, el sistema operativo puede entonces determinar si cualesquiera otros contextos de hilos en el paquete están también en un estado de energía baja. A continuación, el sistema operativo puede llevar a cabo una reasignación lógica basada en software para intercambiar uno o más de los contextos de hilos de energía baja al mismo núcleo que el primer hilo de energía baja y, si es conveniente, intercambiar un hilo activo desde ese núcleo a otro núcleo. Cuando se ha completado el intercambio de modo que todos los contextos de hilos asignados a un núcleo particular estén en un estado de energía baja, el propio núcleo se puede poner en un estado de energía baja. Tal como se utiliza en la presente, el término "hilo" se refiere a un hilo de software. Las expresiones "CPU lógica", "procesador lógico", "contexto de hilo de hardware" y "unidad de hilos" se utilizan de manera intercambiable en al presente para hacer referencia a un contexto de hilo de hardware en el que se puede ejecutar un hilo de software. Al menos una realización de un contexto de hilo de hardware se describe con más detalle a continuación en relación con la figura 2 (véase el análisis de LP₀ y LP₁). Para dicha realización, una unidad de hilos puede comprender una parte de un núcleo, por ejemplo, un procesador lógico SMT (de ejecución multihilo simultánea) de un núcleo multihilo SMT. No obstante, en otras realizaciones, una unidad de hilos puede comprender, por ejemplo, un núcleo completo de un solo hilo.

25 En al menos una realización, las realizaciones del sistema de intercambio analizado en la presente se pueden utilizar juntamente con mecanismos del SO existente, con el fin de lograr la planificación de tareas en aquellos núcleos para los que se producirá el coste mínimo (en términos de energía y/o tiempo). Las realizaciones se pueden desarrollar en el código del kernel del SO juntamente con la política de estados C y planificación del SO. Como alternativa, una o más realizaciones se pueden desarrollar en el firmware de la plataforma con una interfaz a los mecanismos de política de estados C y de planificación del SO.

30 Cabe destacar que los estados C de núcleo del procesador descritos en la presente son para un procesador a modo de ejemplo, tal como aquellos basados en la arquitectura IA-32 y la arquitectura IA-64, comercializados por Intel Corporation, Santa Clara, CA, aunque las realizaciones se pueden utilizar igualmente con otros procesadores. Mostrado a continuación en la Tabla 1 hay una designación a modo de ejemplo de los estados C de núcleo disponibles en una realización, y la tabla 2 asigna estos estados C de núcleo a los estados de ACPI correspondientes. No obstante, se debe sobreentender que el alcance de la presente invención no está limitado por este respecto.

35 Haciendo referencia ahora a la figura 1, se muestra un diagrama de bloques de un sistema 10 que emplea un mecanismo de intercambio para migrar hilos desde un procesador lógico en un núcleo hasta un procesador lógico diferente en un núcleo diferente, basado en la información del estado de energía, de acuerdo con al menos una realización. Tal como se muestra en la figura 1, el sistema 10 incluye un paquete de procesadores 20 que tiene una pluralidad de núcleos de procesadores 25₀ – 25_{n-1} (de manera genérica núcleo 25). El número de núcleos puede variar en diferentes implementaciones, desde paquetes de doble núcleo a paquetes de muchos núcleos que incluyen un número potencialmente grande de núcleos. La naturaleza opcional de los núcleos adicionales se indica en la figura 1 mediante líneas discontinuas. Cada núcleo 25 puede incluir diversas estructuras lógicas y de control para llevar a cabo operaciones sobre los datos en respuesta a instrucciones. Aunque únicamente se ilustra un paquete 20, los métodos y mecanismos descritos se pueden emplear en sistemas informáticos que incluyan también múltiples paquetes.

40 En al menos una realización, uno o más de los núcleos 25 pueden soportar múltiples contextos de hilos de hardware por núcleo (véase, p. ej., el sistema 250 de la figura 2, en el que cada núcleo 252 soporta dos contextos de hilos de hardware por núcleo). Dicha realización no se debería tomar como limitante, ya que alguien experto en la técnica sobreentenderá que cada núcleo puede soportar más de dos contextos de hilos de hardware.

45 La figura 1 ilustra que un sistema informático 10 puede incluir elementos adicionales. Por ejemplo, además del hardware en paquete 20, el sistema 10 también puede incluir una capa de firmware 30, que puede incluir una BIOS (sistema básico de entrada y salida). El sistema informático 10 también puede incluir una interfaz térmica y de energía 40. En al menos una realización, la interfaz térmica y de energía 40 es una interfaz de hardware/software tal como la que se define mediante el estándar de Interfaz Avanzada de Configuración y Energía (ACPI), p. ej., Rev. 3.0b, publicado el 10 de octubre de 2006, mencionado anteriormente. La especificación de ACPI describe los

registros de plataformas, las tablas ACPI, p. ej., 42, y el funcionamiento de una BIOS de ACPI. La figura 1 muestra estos componentes de ACPI colectivamente de manera lógica como una capa entre el hardware en paquete 20 y el firmware 30, por un lado, y un sistema operativo (“SO”) 50 por otro.

5 El sistema operativo 50 de la figura 1 se puede configurar de modo que interactúe con la interfaz térmica y de energía 40, con el fin de dirigir la gestión energética del paquete 20. En consecuencia, la figura 1 ilustra un sistema 10 capaz de utilizar una interfaz ACPI 40 para llevar a cabo la gestión de energía y configuración dirigida por el sistema operativo (OSPM).

Por tanto, el sistema operativo 50 puede incluir la lógica (software, firmware, hardware o una combinación) para llevar a cabo la función de OSPM. En al menos una realización, un módulo lógico de OSPM puede ser código del sistema que es parte del kernel del SO 51. El kernel del SO 51 también puede incluir un módulo lógico de planificación (no se muestra).

El SO 50 también puede incluir un controlador de ACPI (no se muestra) que establece el enlace entre el sistema operativo o la aplicación y el hardware del PC. El controlador puede facilitar las llamadas a ciertas funciones de ACPI de la BIOS, acceso a los registros de ACPI y la lectura de las tablas de ACPI 42.

15 A modo de ejemplo, la Tabla 1 a continuación muestra los estados C de núcleo y sus descripciones, junto con el consumo de potencia estimado y las latencias de salida para estos estados, haciendo referencia a un procesador ejemplar que tiene una potencia térmica de diseño (TDP) de 95 vatios (W). Obviamente, se debe sobrentender que esto es únicamente un ejemplo, y que las realizaciones no están limitadas por este respecto. La Tabla 1 también muestra los estados C de paquete y sus descripciones, la latencia de salida estimada y el consumo de potencia estimado.

Tabla 1

	Descripción	Latencia de salida estimada	Consumo de potencia estimado
Núcleo C0	Todas las lógicas del núcleo activas	N/A	4.9 W
Núcleo C1	Conmutación periódica del reloj del núcleo	2 µs	2.4 W
Núcleo C3	Memoria caché multinivel del núcleo (MLC) borrada e invalidada	10-20 µs	1.7W
Núcleo C6	Conmutación periódica de la energía del núcleo	20-40 µs	0 W
Núcleo C7	Conmutación periódica de la energía del núcleo y las señales “OK para reducir la memoria caché de último nivel (LLC) del paquete (pkg)”	20-40 µs	0 W
Paquete. C0	Todas las lógicas fuera del núcleo y en el núcleo activas	N/A	95 W
Paquete. C1	Todos los núcleos inactivos, conmutación periódica del reloj del paquete	2-5 µs	29 W
Paquete C3	Paquete C1 + todos los enlaces externos pasan a estados inactivos de latencia larga + se pone la memoria en un estado inactivo de latencia corta	~50 µs	21 W
Paquete C6	Paquete C3 + voltaje reducido para el plano de energía (únicamente permanece un voltaje de retención muy bajo) + se pone la memoria en un estado inactivo de latencia larga	~80 µs	6 W
Paquete C7	Paquete C6 + reducción de la LLC	~100 µs	4 W

La Tabla 1 ilustra que los estados C de núcleo C0 y núcleo C1 son estados de energía de latencia relativamente baja, mientras que los estados C profundos (p. ej., núcleo C3, núcleo C6 y núcleo C7) son estados de latencia alta. Cada unidad de hilos SMT (o “procesador lógico”) de un núcleo también puede estar asociado con uno de los estados C ilustrados en la Tabla 1. El estado C asociado con un procesador lógico se puede denominar en la presente como “estado C de hilo”.

La Tabla 2 muestra una asignación, a modo de ejemplo, de los estados C de núcleo de un procesador, a modo de ejemplo, a los estados C de ACPI. Cabe destacar de nuevo que esta asignación es únicamente a modo de ejemplo y que las realizaciones no están limitadas por este respecto.

Tabla 2

Núcleo C0→ACPI C0
Núcleo C1→ACPI C1
Núcleo C3→ACPI C1 o C2
Núcleo C6→ACPI C2 o C3
Núcleo C7→ACPI C3

Cabe destacar que los estados C de paquete no están soportados por ACPI; por lo tanto, no se proporcionan asignaciones ACPI en la Tabla 2 para estados C de paquete citados anteriormente en la Tabla 1.

5 Volviendo ahora a la figura 2 para un análisis breve con el fin de ilustrar las ineficiencias de energía que se pueden producir cuando el kernel del SO (véase 51 de la figura 1) lleva a cabo las técnicas conocidas que no proporcionan la migración de los hilos de energía baja al mismo núcleo.

10 La figura 2 ilustra un sistema 250 con un paquete 20 que incluye dos núcleos, 252₀ y 252₁. Obviamente, mientras el paquete 20 ilustra únicamente dos núcleos, esta simplificación tiene solo fines ilustrativos. Alguien experto en la técnica reconocerá que un paquete 20 puede incluir cualquier número de núcleos sin alejarse del alcance de las realizaciones descritas y reivindicadas en la presente.

15 Los núcleos 252₀ y 252₁ mostrados en la figura 2 son núcleos multihilo. Es decir, la figura 2 ilustra que cada núcleo 252 es un núcleo de ejecución multihilo simultánea (“SMT”) de doble hilo, donde cada núcleo 252 mantiene un estado de la arquitectura independiente (T₀, T₁) para cada uno de los dos contextos de hilos de hardware LP₀, LP₁, pero donde ciertos recursos diferentes 220, 222, 224 son compartidos por los dos contextos de hilos de hardware LP₀, LP₁. Tal como se ha mencionado anteriormente, para dicho ejemplo cada contexto de hilo de hardware LP (o “CPU lógica” o “procesador lógico”) puede tener un estado C independiente.

20 Si se permite a un contexto de hilo de hardware realizar una transición a un estado C de hilo profundo sin tomar en consideración el hilo total y el estado C de núcleo para el paquete 20, se pueden producir ineficiencias tanto de energía como de rendimiento. Este inconveniente se ilustra en el ejemplo “Antes” del sistema 250A de la figura 2. A modo de ejemplo, se supone que el hilo de hardware LP₀ del núcleo 0, 252₀, está en un estado C activo (p. ej., C0) pero el hilo de hardware LP₁ del núcleo 0, 252₀, está en un estado C de núcleo profundo (p. ej., C6). De acuerdo con la política actual de estados C, el núcleo C0 252₀ no puede entrar en un estado C de núcleo profundo debido a que uno de sus hilos de hardware LP₀ no está en un estado C de hilo profundo. Tal como se ha mencionado anteriormente la política actual de estados C dicta que un núcleo únicamente puede entrar en un estado C de núcleo profundo si todos los contextos de hardware de ese núcleo están en un estado C de hilo profundo.

30 La figura 2 ilustra una situación similar con el núcleo 1, 252₁ en el ejemplo “Antes”, se supone que un hilo de hardware LP₂ del núcleo 1, 252₁, está en un estado C de hilo superficial (p. ej., C1) y que el otro hilo de hardware LP₃ del núcleo 1, 252₁, está en un estado C de hilo profundo (p. ej., C6). En consecuencia, el núcleo 1 252₁ no puede entrar en un estado C de núcleo profundo debido a que uno de sus hilos de hardware LP₀ no está en un estado C de hilo profundo.

En el ejemplo “Antes” ilustrado en la figura 2, cada núcleo tiene n contextos de hilo y hay un total de n contextos de hilo en un estado C de hilo profundo. No obstante, ninguno de los núcleos puede entrar en un estado C de núcleo profundo debido al requisito de la política de que ningún núcleo puede entrar en un estado C de núcleo profundo a menos que **todos** los contextos de hilo SMT en el núcleo estén en un estado C de hilo profundo.

35 Consultando la Tabla 1, uno puede observar que el ejemplo mostrado para el sistema 250A en la figura 2 conduce a un consumo de potencia innecesario. La Tabla 1 ilustra que un núcleo en un estado de energía C-6 utiliza aproximadamente cero vatios (0 W). Por tanto, una utilización eficiente del estado de energía C-6 de núcleo puede proporcionar unos beneficios energéticos significativos. No obstante, tal como se ilustra en la figura 2, hay dos contextos de hilo SMT, LP1 y LP3, en un estado C de hilo profundo para el paquete 20. Debido a que estos contextos de hilo están en núcleos diferentes, ninguno de los núcleos, el núcleo 0, 252₀, o el núcleo 1, 252₁, puede entrar en un estado C de núcleo profundo. Es decir, el núcleo 0 252₀ está en un estado de núcleo C0 debido a que uno de sus contextos de hilos de hardware, LP₀, está en un estado C de hilo activo (p. ej., C0). Por tanto, el consumo de potencia del núcleo 0 252₀ es 4.9 W. De manera similar, el núcleo 1 252₁ no puede entrar en un estado C de núcleo profundo debido a que uno de sus contextos de hilos de hardware, LP₀, está en un estado C de hilo de inactividad superficial (C1). El núcleo 1 252 está en el estado de núcleo C1. Por tanto, el consumo de potencia del núcleo 1 252₁ es de 2.4 W. El consumo de potencia total del sistema 250A en el ejemplo “Antes” de la figura 2 es 4.9 + 2.4 = 7.3 W.

El ejemplo “Después” del sistema 250B de la figura 2 ilustra que estas ineficiencias de energía se pueden evitar mediante una reasignación basada en software de los hilos de software a contextos de hilos de hardware, de

acuerdo con al menos una realización de la presente invención. Esta reasignación aprovecha el beneficio energético del estado de núcleo C-6 mediante la asignación de n hilos de hardware (en este caso n=2) al mismo núcleo de ejecución SMT de n vías, donde todos los n hilos están en el estado de hilo C-6.

5 El ejemplo “Después” de la figura 2 ilustra que se puede llevar a cabo una reasignación, de acuerdo con al menos una realización de la invención, con el fin de lograr un intercambio de hilos de software entre los contextos de hilos de hardware. La figura 2 muestra que, en el sistema 250B, el hilo de software de LP₁ se ha reasignado a LP₂, y viceversa. Por tanto, en el ejemplo “Después”, ambos contextos de hilos de hardware LP₀ y LP₁ del núcleo 0 del sistema 250B están en el estado C6. En consecuencia, el núcleo 0 252₀ se ha puesto en el estado C de núcleo C6 mucho más eficiente energéticamente. Por el contrario, los hilos que están en estados superficiales se han asignado a los contextos de hilos de hardware del núcleo 1 252₁. Por tanto, el núcleo 252₁ está en el estado C C0 activo. La utilización de potencia de los dos núcleos 252₀ y 252₁ del sistema 250B es: 4.9 W + 0 W = 4.9 W. En consecuencia, la diferencia en la utilización de potencia entre los núcleos del sistema “Antes” 250A y el sistema “Después” 250B es de 7.3 W – 4.9 W = 2.4 W. Esto representa un ahorro de energía de un 33% debido al intercambio.

15 Cabe destacar que el intercambio ilustrado en la figura 2 se lleva a cabo en el software y permite un intercambio de un subconjunto de hilos de un núcleo SMT multihilo. Por el contrario, los sistemas de salto de núcleo basado en hardware actuales copian todos los estados de un primer núcleo a otro núcleo. Esto se hace, en algunos casos, para equilibrar la carga o uniformizar los puntos calientes en el núcleo. Dichos planteamientos de salto de núcleo basado en hardware se pueden utilizar para conmutar trabajo desde un núcleo de un solo hilo a otro, o para conmutar todo el trabajo para todos los hilos de un núcleo multihilo a otro núcleo multihilo. No obstante, dichos planteamientos de salto de núcleo basado en hardware son mecanismos relativamente gruesos que no permiten una asignación de grano más fino de un subconjunto de hilos de un núcleo a otro. Un problema que se debe resolver cuando se intercambia un subconjunto de hilos de un núcleo a otro es la tarea muy compleja de “desenredar” los recursos compartidos (véase, p. ej., los recursos de ejecución compartidos 220 y las memorias cachés compartidas 222 de la figura 2) de modo que se pueda transferir el estado del hardware específico del hilo al nuevo núcleo. Dicha operación puede ser extremadamente compleja en sistemas de salto de núcleo basado en hardware conocidos. Otro inconveniente del salto de núcleo basado en hardware es que los planteamientos conocidos transfieren la ejecución de los hilos de un núcleo a otro únicamente cuando el hardware pasa a estar inactivo. El tiempo que se tarda en esperar que los recursos de hardware pasen a estar inactivos puede dar como resultado una oportunidad perdida para ahorrar energía.

20 Para resolver estas y otras dificultades con el salto de núcleo basado en hardware, el inventor ha ideado un planteamiento de software, denominado en la presente como “salto de hilo”. El planteamiento es un planteamiento de reasignación basada en software que reasigna los hilos entre los procesadores lógicos de diferentes núcleos en un paquete, con el fin de lograr un ahorro de energía.

25 Las realizaciones de salto de hilo descritas en la presente se pueden llevar a cabo en el software (tal como, p. ej., el código del kernel de un sistema operativo) sin que se requiera ningún cambio de hardware subyacente. El salto de hilo resuelve la dificultad de desenredar recursos por hilo, cuando se transfiere solo un subconjunto de hilos de un núcleo a otro núcleo. Una razón para esto es que, en lugar de esperar para que los hilos de hardware pasen a estar inactivos, las realizaciones del mecanismo de salto de hilo (tal como, p. ej., la descrita a continuación en relación con las figuras 3 y 4) utilizan un planteamiento de software para suspender la ejecución del hilo de software que se debe expulsar a otro núcleo. Una vez suspendido, el hilo de software se asigna a un nuevo procesador lógico en un *núcleo de procesador diferente*, y el planificador del SO extrae una tarea para el hilo receptor, la tarea a realizar en el procesador lógico recién asignado. Este procesamiento se contrapone a la planificación de hilos del SO tradicional, que no se activa mediante un cambio en el estado C de hilo y no conlleva una reasignación lógica del hilo de software a un contexto de hilo de hardware en un núcleo diferente.

30 La figura 3 es un diagrama de flujo de datos que ilustra con más detalle las operaciones de un mecanismo de reasignación basado en software de acuerdo con al menos una realización. La figura 3 ilustra que, en la operación 1, la OSPM ha determinado que un contexto de hilo de hardware (“X”) está listo para entrar en un estado de inactividad profundo. Antes de que el kernel del SO ponga el contexto de hilo de hardware en un estado inactivo, este lleva a cabo un procesamiento (analizado con más detalle a continuación en relación con la figura 4) para determinar si en su lugar sería más eficiente una extracción de tarea basada en software con el fin de extraer la tarea de algún otro contexto de hilo en el contexto de hilo X. Este procesamiento incluye la operación 2, donde se determina si el(los) contexto(s) de hilo de hardware hermano en el mismo núcleo 352 que el X está/n en un estado de inactividad profundo. Si no lo están, podría no ser eficiente, desde un punto de vista de la energía, poner X en un estado de inactividad profundo, ya que el núcleo 352 no puede entrar en el estado de inactividad profundo de acuerdo con la política de estados C en curso, a menos que todos sus contextos de hardware estén inactivos. Tal como se utiliza en la presente, un contexto de hilo de hardware “hermano” hace referencia a los demás contextos de hilo de hardware, además del contexto particular que está listo para entrar en un estado de inactividad profundo, en un único núcleo. Para las realizaciones que tienen núcleos de doble hilo, el número de hermanos para cualquier contexto de hilo de hardware es uno. Para realizaciones que tienen núcleos con más de 2 contextos de hilo de hardware por núcleo, el número de hermanos puede ser mayor de uno.

Si se determina en la operación 2 que uno o más del (de los) contexto(s) de hilo hermano para X no están en un estado de inactividad profundo, entonces el sistema operativo determina si cualquier otro núcleo incluye un contexto de hilo de hardware que ya está en un estado de inactividad profundo. Por tanto, en las operaciones 3 y 4, se evalúa el estado de energía de los contextos de hilo en otros núcleos.

5 Para mayor facilidad de ilustración y análisis, la figura 3 ilustra dos núcleos 352, 353, con dos contextos de hilo por núcleo: W, X para el núcleo 0 e Y, Z para el núcleo 1, respectivamente. No obstante, alguien experto en la técnica reconocerá que las realizaciones del mecanismo de salto de hilo descrito en la presente se pueden aplicar a cualquier número de núcleos en los que cada uno tenga cualquier número de contextos de hilo de hardware.

10 En la operación 3, se determina si cualquier otro contexto de hardware en un núcleo activo está en un estado de inactividad profundo. Con una finalidad ejemplar, la figura 3 ilustra que, en la operación 3, se determina que el contexto de hilo de hardware Y está en un estado de inactividad profundo. Tras descubrir un contexto de hilo de inactividad profundo Y para un núcleo activo 353, se evalúa el estado de energía de uno o más de los otros contextos de hilo en el núcleo 353. En la figura 3, se determina en la operación 4 que el contexto de hilo Z está activo. Por tanto, con una finalidad de ahorro de energía sería más eficiente poner el contexto de hilo Z en el estado de inactividad profundo y extraer la tarea en curso de Z al contexto de hilo de hardware X.

15 En consecuencia, en la operación 5 se lleva a cabo una reasignación. Esta reasignación se lleva a cabo en el software. El kernel del SO ejecuta una operación de suspensión para suspender la tarea que se ejecuta en curso en el procesador lógico Z. Los procesadores lógicos se reasignan de modo que el hilo asociado con el contexto de hilo de hardware Z se reasigne al procesador lógico Y. Esta reasignación incluye la transferencia de los datos de contexto de la tarea que se ejecuta mediante el procesador lógico Z en el núcleo 1 353 al procesador lógico Y en el núcleo 0 252. La ejecución de la tarea se reinicia posteriormente, pero en el contexto de hardware Y en lugar de Z.

20 En la operación 6, el procesador lógico Z se pone mediante el kernel del SO en un estado de inactividad profundo. Como todos los contextos de hilo de hardware del núcleo 1 están ahora en el estado de inactividad profundo, el núcleo 353 entra en el estado de inactividad profundo en la operación 7.

25 La figura 4 es un diagrama de flujo que muestra al menos una realización de un método 400 para llevar a cabo un intercambio tal como se ilustra en la figura 3 y tal como se ilustra en el ejemplo "Después" del sistema 250B de la figura 2. En al menos una realización, el método 400 ilustrado en la figura 4 se puede llevar a cabo mediante un kernel del SO (véase, p. ej., 51 de la figura 1). La figura 4 ilustra que el método 400 proporciona el salto de hilo para intercambiar una tarea de software a un contexto de hilo de hardware en un núcleo en respuesta a una transición inminente del estado de energía de un contexto de hilo de hardware en otro núcleo.

30 La figura 4 ilustra que el método 400 comienza en el bloque de inicio 402. El bloque de inicio 402 se puede activar en respuesta a una determinación por parte de la lógica OSPM (no se muestra) del kernel (p. ej., 51 de la figura 1) que un procesador lógico (LP X) en uno de los núcleos (núcleo 0) de un sistema debería ponerse en un estado C de hilo profundo.

35 Desde el bloque de inicio 402, el procesamiento continúa en el bloque 404. En el bloque 404, se determina si el otro (los otros) contexto(s) de hilo de hardware en el núcleo 0 están en un estado C de hilo profundo. Para una realización que tiene dos contextos de hilo de hardware por núcleo, se determina en el bloque 404 si el otro núcleo está en un estado C de hilo profundo. En todas las realizaciones que incluyen n contextos de hilo de hardware por núcleo, donde $n > 2$, el procesamiento en el bloque 404 determina si todos los demás contextos de hilo de hardware además de LP X en el núcleo 0 están en un estado C de hilo profundo. Si esto es así, no es necesario intercambio alguno y es adecuado poner todo el núcleo en un estado C de núcleo profundo, después de poner X en un estado C de hilo profundo, y el procesamiento continúa en el bloque 418. En caso contrario, el procesamiento continúa en el bloque 406.

45 En el bloque 406, se ha determinado por medio del procesamiento en el bloque 404 que no se dan las condiciones para poner el núcleo local de X en un estado C de núcleo profundo. Debido a que aún se podría lograr un beneficio de ahorro de energía si cualquier otro núcleo es un contexto de hilo casi en condiciones de estado C profundo, en el bloque 406 se determina si otro núcleo además del núcleo 0 tiene un contexto de hilo de hardware en un estado C de hilo profundo. Obviamente, otros núcleos que ya están en un estado C de núcleo profundo tendrán contextos de hilo en un estado C de hilo profundo, de modo que en el bloque 406 se lleva a cabo una evaluación únicamente en los núcleos que no están ya en un estado C de núcleo profundo.

50 Si no se identifican contextos de hilo en otro núcleo en el bloque 406, entonces el procesamiento continúa al bloque opcional 415 o al bloque 416, dependiendo de la realización. El procesamiento continúa en el bloque opcional 415 para las realizaciones que incluyen más de dos núcleos en un paquete. La naturaleza opcional del bloque 415 se indica mediante las líneas discontinuas en la figura 4. En el bloque 415, se determina si hay núcleos adicionales que se deben evaluar para buscar hilos potenciales a intercambiar a LP X. Para una realización de doble núcleo, el bloque 415 es opcional no es necesario llevarlo a cabo. Si en el bloque 415 se determina que se deben evaluar núcleos adicionales, el procesamiento vuelve al bloque 406. En caso contrario, el procesamiento continúa en el

bloque 416. En el bloque 416, se produce el procesamiento del estado C normal, no hay una ganancia de eficiencias adicionales mediante salto de hilo.

5 Si en lugar de esto, en el bloque 406 se determina que un contexto de hilo de hardware (LP Y) en otro núcleo (núcleo 1) está en un estado C de hilo profundo, entonces el procesamiento continúa al bloque opcional 407 o al bloque 408, dependiendo de la realización. El procesamiento continúa en el bloque opcional 407 para las realizaciones que incluyen más de dos procesadores lógicos por núcleo y continúa en el bloque 408 para realizaciones que incluyen únicamente dos procesadores lógicos por núcleo. La naturaleza opcional del bloque 407 se indica en la figura 4 con líneas discontinuas.

10 En el bloque 407, se determina si un intercambio de trabajo entre los dos núcleos daría como resultado que el nuevo núcleo tuviera más procesadores lógicos en un estado de hilo de inactividad profundo de los que tendría el núcleo 0 si se pusiera LP X en un estado de hilo de inactividad profundo en el núcleo 0. Si esto es así, el procesamiento continúa en el bloque 408 con el fin de proseguir con el procesamiento del intercambio. Si esto no es así, no hay ganancia de eficiencias debido al intercambio, así que por el contrario el procesamiento continúa en el bloque, el bloque opcional 415 (parar realizaciones con más de dos procesadores lógicos por núcleo) o en el bloque 416.

15 En el bloque 408, se determina si cualquier contexto de hilo de hardware (LP Z) en el otro núcleo (núcleo 1) está en un estado activo. Si esto es así, entonces se produce la situación de que un núcleo, diferente del núcleo local de LP X, tiene al menos un contexto de hilo de hardware en un estado C profundo y al menos otro contexto de hilo de hardware en un estado activo. Por tanto, esto sería conveniente, desde un punto de vista del ahorro de energía, para consolidar los contextos de hilo de hardware de inactividad profundo en el mismo núcleo, aunque también es verdad que el trabajo en curso del LP Z activo se debería extraer a otro contexto de hilo de hardware antes de que el LP Z realice la transición al estado C de hilo de inactividad profundo. En consecuencia, el procesamiento continúa en el bloque 410. No obstante, si en el bloque 408 se determina que ningún hermano en el otro núcleo está en un estado activo, entonces no se debería producir la extracción de la tarea, y el procesamiento continúa en el bloque 414. En este caso, el núcleo hermano ya está en un estado C de hilo profundo o en algún otro estado no activo (tal como, p. ej., un estado de inactividad superficial C1).

20 En el bloque 410, se lleva a cabo la extracción de tareas de software. Es decir, la tarea que se ejecuta en ese momento en LP Z se suspende temporalmente y se mueve a LP X. Para hacerlo, una entidad de software, tal como el sistema operativo (y, más en particular, el planificador de un SO) provoca la suspensión de la tarea activa de LP Z. El trabajo asociado con la tarea activa de LP Z se asigna a continuación al núcleo local de LP X. El estado necesario de LP X para comenzar a ejecutar la tarea de LP Z se mueve al núcleo de LP X. La manera de transferir el estado del contexto entre núcleos en el bloque 410 depende de la implementación y se puede llevar a cabo de diversas maneras. En al menos una realización, por ejemplo, los datos de contexto de LP Z se copian en una sección adecuada de memoria o jerarquía de memoria (p. ej., en una memoria caché) a la cual tiene acceso LP X. El puntero a instrucciones es parte del estado que se transfiere, de modo que LP X, cuando reinicia la ejecución, comenzará la ejecución en la siguiente instrucción que habría ejecutado LP Z. Después de que se lleve a cabo la extracción de tarea en el bloque 410 la operación continúa en el bloque 414.

30 En el bloque 414, la reasignación lógica de LP X y LP Z se intercambia por software. Es decir, la ID de la CPU lógica asociada con LP X se intercambia por LP Z y, de manera similar, la ID de CPU lógica asociada con LP Z se intercambia por LP X. La operación del hilo activo que estaba corriendo anteriormente en LP Z se reinicia posteriormente en LP X. A continuación, el procesamiento continúa en el bloque 417, donde LP Z entra en el estado de inactividad profundo.

Desde el bloque 417, el procesamiento continúa al bloque 420. En el bloque 420 se determina si todos los procesadores lógicos en el núcleo local de LP Y están ahora en estados de suspensión profundos. Si no lo están, el procesamiento finaliza en el bloque 422.

45 Si se determina en el bloque 420 que todos los procesadores lógicos del núcleo local de LP Y están ahora en un estado C de hilo profundo, en ese caso el núcleo entra en un estado C de núcleo profundo. A continuación, el procesamiento finaliza en el bloque 422.

50 Las realizaciones se pueden implementar en muchos tipos diferentes de sistemas. Haciendo referencia ahora a la figura 5, se muestra un diagrama de bloques de un sistema 500 de acuerdo con una realización de la presente invención. Tal como se muestra en la figura 5, el sistema 500 puede incluir uno o más elementos de procesamiento 510, 515, que están acoplados a un concentrador de controladora de la memoria gráfica (GMCH) 520. La naturaleza opcional de los elementos de procesamiento adicionales 515 se indica en la figura 5 con líneas discontinuas.

55 Cada elemento de procesamiento puede ser un único núcleo o, como alternativa, puede incluir múltiples núcleos. De manera opcional, los elementos de procesamiento pueden incluir otros elementos en el chip además de los núcleos de procesamiento, tal como el controlador de memoria integrado y/o la lógica de control de E/S integrada. Además, en al menos una realización, el(los) núcleo(s) de los elementos de procesamiento pueden ser multihilo ya que estos pueden incluir más de un contexto de hilo de hardware por núcleo.

La figura 5 ilustra que el GMCH 520 puede estar acoplado a una memoria 540 que, por ejemplo, puede ser una memoria de acceso aleatorio dinámica (DRAM). En al menos una realización, la memoria 540 puede incluir las instrucciones o el código que comprende un sistema operativo (p. ej., 50 de la figura 1).

5 El GMCH 520 puede ser un circuito integrado auxiliar, o una parte de un circuito integrado auxiliar. El GMCH 520 se puede comunicar con el(los) procesador(es) 510, 515 y controlar la interacción entre el(los) procesador(es) 510, 515 y la memoria 540. El GMCH 520 también puede actuar como una interfaz de bus acelerado entre el(los) procesador(es) 510, 515 y otros elementos del sistema 500. En al menos una realización, el GMCH 520 se comunica con el(los) procesador(es) 510, 515 a través de un bus multipunto, tal como un bus frontal (FSB) 595.

10 Asimismo, el GMCH 520 está acoplado a una pantalla 540 (tal como una pantalla de panel plano). El GMCH 520 puede incluir un acelerador gráfico integrado. El GMCH 520 está acoplado además a un concentrador de controladora (ICH) de entrada/salida (E/S) 550, el cual se puede utilizar para acoplar diversos dispositivos periféricos al sistema 500. En la realización de la figura 5 se muestra, por ejemplo, un dispositivo gráfico externo 560, que puede ser un dispositivo gráfico discreto acoplado al ICH 550, junto con otro dispositivo periférico 570.

15 Como alternativa, en el sistema 500 también pueden estar presentes elementos de procesamiento adicionales o diferentes. Por ejemplo, el(los) elemento(s) de procesamiento adicional(es) 515 puede(n) incluir procesador(es) adicional(es) que son iguales al procesador 510, procesador(es) adicional(es) que son heterogéneos o asimétricos al procesador 510, aceleradores (tal como, p. ej., aceleradores gráficos o unidades de procesamiento digital de señales (DSP)), matrices de puertas programables o cualquier otro elemento de procesamiento. Puede haber diversas diferencias entre los recursos físicos 510, 515 en términos de un espectro de métricas de calidad que incluyen características de arquitectura, microarquitectura, térmicas, de consumo de potencia y similares. Estas diferencias se pueden manifestar de manera efectiva por sí mismas como una asimetría y heterogeneidad entre los elementos de procesamiento 510, 515. En al menos una realización, los diversos elementos de procesamiento 510, 515 pueden residir en el mismo encapsulado del chip.

25 Haciendo referencia ahora a la figura 6, se muestra un diagrama de bloques de una segunda realización del sistema 600 de acuerdo con una realización de la presente invención. Tal como se muestra en la figura 6, el sistema multiprocesador 600 es un sistema de interconexión de punto a punto, e incluye un primer elemento de procesamiento 670 y un segundo elemento de procesamiento 680 acoplados a través de una interconexión de punto a punto 650. Tal como se muestra en la figura 6, cada uno de los elementos de procesamiento 670 y 680 pueden ser procesadores multinúcleo, que incluyen un primer y segundo núcleo de procesador (es decir, los núcleos de procesador 674a y 674b y los núcleos de procesador 684a y 684b).

30 Como alternativa, uno o más elementos de procesamiento 670, 680 pueden ser un elemento distinto a un procesador, tal como un acelerador o una matriz de puertas programables.

35 Aunque se muestra con únicamente dos elementos de procesamiento 670, 680, se debe sobreentender que el alcance de la presente invención no está limitado por ello. En otras realizaciones, pueden estar presentes uno o más elementos de procesamiento adicionales en un procesador dado.

40 El primer elemento de procesamiento 670 puede incluir además un concentrador de controladora de la memoria (MCH) 672 y unas interfaces de punto a punto (P-P) 676 y 678. De manera similar, el segundo elemento de procesamiento 680 puede incluir un MCH 682 y unas interfaces P-P 686 y 688. Tal como se muestra en la figura 6, los MCH 672 y 682 acoplan los procesadores a las memorias respectivas, en concreto a una memoria 642 y una memoria 644, que pueden ser partes de una memoria principal conectada localmente a los procesadores respectivos.

45 El primer elemento de procesamiento 670 y el segundo elemento de procesamiento 680 pueden estar acoplados a un circuito integrado auxiliar 690 a través de las interconexiones P-P 676, 686 y 684, respectivamente. Tal como se muestra en la figura 6, el circuito integrado auxiliar 690 incluye unas interfaces P-P 694 y 698. Asimismo, el circuito integrado auxiliar 690 incluye una interfaz 692 para acoplar el circuito integrado auxiliar 690 con un motor gráfico de alto rendimiento 648. En una realización, el bus 649 se puede utilizar para acoplar el motor gráfico 648 al circuito integrado auxiliar 690. Como alternativa, una interconexión de punto a punto 649 puede acoplar estos componentes.

50 A su vez, el circuito integrado auxiliar 690 puede estar acoplado a un primer bus 616 a través de una interfaz 696. En una realización, el primer bus 616 puede ser un bus de interconexión de componentes periféricos (PCI), o un bus tal como un bus PCI Express u otro bus de interconexión de E/S de tercera generación, aunque el alcance de la presente invención no está limitado por esto.

55 Tal como se muestra en la figura 6, diversos dispositivos de E/S 614 pueden estar acoplados al primer bus 616, junto con un puente de bus 618 que acopla el primer bus 616 a un segundo bus 620. En una realización, el segundo bus 620 puede ser un bus de pocos pines (LPC). Diversos dispositivos pueden estar acoplados al segundo bus 620 que incluyen, por ejemplo, un teclado/ratón 622, dispositivos de comunicación 626 y una unidad de almacenamiento de datos 628, tal como una unidad de disco u otro dispositivo de almacenamiento masivo que puede incluir el código 630, en una realización. El código 630 puede incluir instrucciones para llevar a cabo realizaciones de uno o más de los métodos descritos anteriormente. Además, una E/S de audio puede estar acoplada al segundo bus 620. Cabe

destacar que son posibles otras arquitecturas. Por ejemplo, en lugar de la arquitectura de punto a punto de la figura 6, un sistema puede implementar un bus multipunto u otra arquitectura de este tipo.

5 Haciendo referencia ahora a la figura 7, se muestra un diagrama de bloques de una tercera realización del sistema 700 de acuerdo con una realización de la presente invención. Elementos iguales en las figuras 6 y 7 tienen los mismos números de referencia, y se han omitido ciertos aspectos de la figura 6 en la figura 7 con el fin de evitar complicar otros aspectos de la figura 7.

10 La figura 7 ilustra que los elementos de procesamiento 670, 680 pueden incluir una memoria y una lógica de control de E/S ("CL") integradas 672 y 682, respectivamente. En al menos una realización, la CL 672, 682 puede incluir la lógica del concentrador de controladora de la memoria (MCH), tal como el descrito anteriormente en relación con las figuras 5 y 6. Además, la CL 672, 682 también puede incluir la lógica de control de E/S. La figura 7 ilustra que no solo las memorias 642, 644 están acopladas a la CL 672, 682, sino también que los dispositivos de E/S 714 también están acoplados a la lógica de control 672, 682. Los dispositivos de E/S antiguos 715 están acoplados al circuito integrado auxiliar 690.

15 Las realizaciones de los mecanismos expuestos en la presente se pueden implementar en el hardware, software, firmware o en una combinación de dichos planteamientos de implementación. Las realizaciones de la invención se pueden implementar como programas informáticos que se ejecutan en sistemas programables que comprenden al menos un procesador, un sistema de almacenamiento de datos (que incluye una memoria volátil y no volátil y/o unos elementos de almacenamiento), al menos un dispositivo de entrada y al menos un dispositivo de salida.

20 El código de programa, tal como el código 630 ilustrado en la figura 6, se puede aplicar para introducir datos con el fin de llevar a cabo las funciones descritas en la presente y generar una información de salida. Por ejemplo, el código de programa 630 puede incluir un sistema operativo que está codificado para llevar a cabo las realizaciones de los métodos ilustrados en las figuras 2, 3 y 4. En consecuencia, las realizaciones de la invención también incluyen los medios que son accesibles por máquina y utilizables por el ordenador, los medios que contienen las instrucciones para llevar a cabo las operaciones de un método o que contienen los datos de diseño, tal como la HDL, que define las características de estructuras, circuitos, aparatos, procesadores y/o sistema descritos en la presente. Dichas realizaciones también se pueden denominar productos de programas informáticos.

25 Dichos medios de almacenamiento accesibles por máquina y utilizables por un ordenador pueden incluir, sin carácter limitante, unas disposiciones tangibles de partículas fabricadas o formadas mediante una máquina o dispositivo, que incluyen medios de almacenamiento tales como discos duros, cualquier otro tipo de disco que incluye los discos flexibles, discos ópticos, discos compactos con memorias de solo lectura (CD-ROM), discos compactos regrabables (CD-RW) y discos magnéticos y ópticos, dispositivos semiconductores tales como memorias de solo lectura (ROM), memorias de acceso aleatorio (RAM) tales como memorias de acceso aleatorio dinámicas (DRAM), memorias de acceso aleatorio estáticas (SRAM), memorias de solo lectura programables y borrables (EPROM), memorias flash, memorias de solo lectura programable y borrable eléctricamente (EEPROM), tarjetas magnéticas u ópticas, o cualquier otro tipo de medios utilizables por un ordenador adecuados para almacenar instrucciones electrónicas.

30 La información de salida se puede aplicar a uno o más dispositivos de salida, de una manera conocida. Para la finalidad de esta aplicación, un sistema de procesamiento incluye cualquier sistema que tenga un procesador, tal como, por ejemplo, un procesador digital de señales (DSP), un microcontrolador, un circuito integrado de aplicación específica (ASIC) o un microprocesador.

35 Los programas se pueden implementar en un lenguaje de programación orientado a objetos o procedimental de alto nivel para comunicarse con un sistema de procesamiento. Los programas también se pueden implementar, si se desea, en lenguaje ensamblador o máquina. De hecho, los mecanismos descritos en la presente no están limitados en su alcance por ningún lenguaje de programación particular. En cualquier caso, el lenguaje puede ser un lenguaje compilado o interpretado.

40 En la presente se ofrecen realizaciones de métodos, aparatos y sistemas para reasignar unidades de hilos y redistribuir el trabajo entre los hilos en núcleos diferentes para consolidar los hilos inactivos en el mismo núcleo. Aunque se han mostrado y descrito realizaciones particulares de la presente invención, será obvio para aquellos que son expertos en la técnica que se pueden realizar numerosos cambios, variaciones y modificaciones sin alejarse del alcance de las reivindicaciones adjuntas. Las reivindicaciones adjuntas engloban dentro de su alcance todos de dichos cambios, variaciones y modificaciones que están dentro del alcance de la presente invención.

REIVINDICACIONES

1. Un método que comprende:
 - 5 en base a la información del estado de energía en un primer procesador lógico de un primer núcleo (510, 670) y un primer procesador lógico de un segundo núcleo (515, 680), reasignar el trabajo del primer procesador lógico del primer núcleo al primer procesador lógico del segundo núcleo;
 - poner el primer procesador lógico del primer núcleo en un estado de suspensión profundo;
 - poner el primer núcleo en el estado de suspensión profundo; y
 - poner el primer procesador lógico del segundo núcleo en un segundo estado de energía.
- 10 2. El método de la reivindicación 1, donde dicha reasignación se basa además en la información del estado de energía de un segundo procesador lógico del primer núcleo (510, 670).
- 15 3. El método de la reivindicación 1, donde dicha reasignación se basa además en la información del estado de energía de un segundo procesador lógico del segundo núcleo (515, 680); y/o donde dicha información del estado de energía de dicho primer procesador lógico de dicho primer núcleo (510, 670) comprende además si el primer procesador lógico del primer núcleo está listo para entrar en el estado de suspensión profundo.
4. El método de la reivindicación 2, donde dicha información del estado de energía de dicho segundo procesador lógico de dicho primer núcleo (510, 670) comprende, además:
 - si el segundo procesador lógico del primer núcleo está en el estado de suspensión profundo.
- 20 5. El método de la reivindicación 3, donde dicha información del estado de energía de dicho segundo procesador lógico de dicho segundo núcleo (515, 680) comprende, además:
 - si el segundo procesador lógico del segundo núcleo (515, 680) están en el estado de suspensión profundo.
6. El método de la reivindicación 1, donde dicha reasignación se lleva a cabo en el software; y de manera opcional, donde dicha reasignación se lleva a cabo mediante un sistema operativo (51); y de manera opcional, donde dicha reasignación comprende además llevar a cabo una conmutación de contexto basada en software.
- 25 7. El método de la reivindicación 1, donde los procesadores lógicos residen en el mismo encapsulado de chip (20); y/o donde la reasignación comprende además la reasignación de un identificador de hilo del primer procesador lógico del primer núcleo (510, 670) a un primer procesador lógico del segundo núcleo (515, 680), y reasignar un identificador de hilo de un segundo procesador lógico del segundo núcleo al primer procesador lógico del primer núcleo.
- 30 8. Un sistema (500, 600) que comprende:
 - un encapsulado de chip (20) que incluye un primer procesador lógico de un primer núcleo (510, 670) y un primer procesador lógico de un segundo núcleo (515, 680);
 - un módulo planificador (51) acoplado con el encapsulado de chip, donde el módulo planificador reasigna, basado en la información del estado de energía relacionada con el primer procesador lógico del primer núcleo y el primer procesador lógico del segundo núcleo, el trabajo del primer procesador lógico del primer núcleo al primer procesador lógico del segundo núcleo; y
 - 35 un módulo de energía acoplado con el módulo planificador, donde el módulo de energía pone el primer procesador lógico del primer núcleo en un estado de suspensión profundo, pone el primer núcleo en el estado de suspensión profundo y pone el primer procesador lógico del segundo núcleo en un segundo estado de energía.
- 40 9. Un sistema (500, 600) según se reivindica en la reivindicación 8, donde dicho módulo planificador (51) y dicho módulo de energía son módulos de software; y/o los núcleos residen en un mismo encapsulado de chip (20); y/o dicho primer núcleo comprende además un controlador de memoria integrado.
10. El sistema (500, 600) de la reivindicación 8, donde dicha reasignación está basada además en la información del estado de energía de un segundo procesador lógico del primer núcleo (510, 670).
- 45 11. El sistema (500, 600) de la reivindicación 8, donde dicha reasignación está basada además en la información del estado de energía de un segundo procesador lógico del segundo núcleo (515, 680); y/o donde dicha información del estado de energía de dicho primer procesador lógico de dicho primer núcleo (510, 670) comprende además si el primer procesador lógico del primer núcleo está listo para entrar en el estado de suspensión profundo.

12. El sistema (500, 600) de la reivindicación 8, donde dicha información del estado de energía de dicho segundo procesador lógico de dicho primer núcleo (510, 670) comprende, además:

si el segundo procesador lógico del primer núcleo está en el estado de suspensión profundo.

5 13. El sistema (500, 600) de la reivindicación 8, donde dicha información del estado de energía de dicho segundo procesador lógico de dicho segundo núcleo (515, 680) comprende, además:

si el segundo procesador lógico del segundo núcleo está en el estado de suspensión profundo.

10 14. El sistema (250, 500, 600) de la reivindicación 8, donde el módulo planificador (51) se configura además de modo que reasigne un identificador de hilo del primer procesador lógico del primer núcleo al primer procesador lógico del segundo núcleo (515, 680), y reasigne un identificador de hilo de un segundo procesador lógico del segundo núcleo (515, 680) al primer procesador lógico del primer núcleo (510, 670).

15. Unas instrucciones legibles por máquina dispuestas en un soporte legibles por ordenador que, cuando se ejecutan mediante un ordenador, hacen que el ordenador lleve a cabo el método según se reivindica en una de las reivindicaciones 1 a 7.

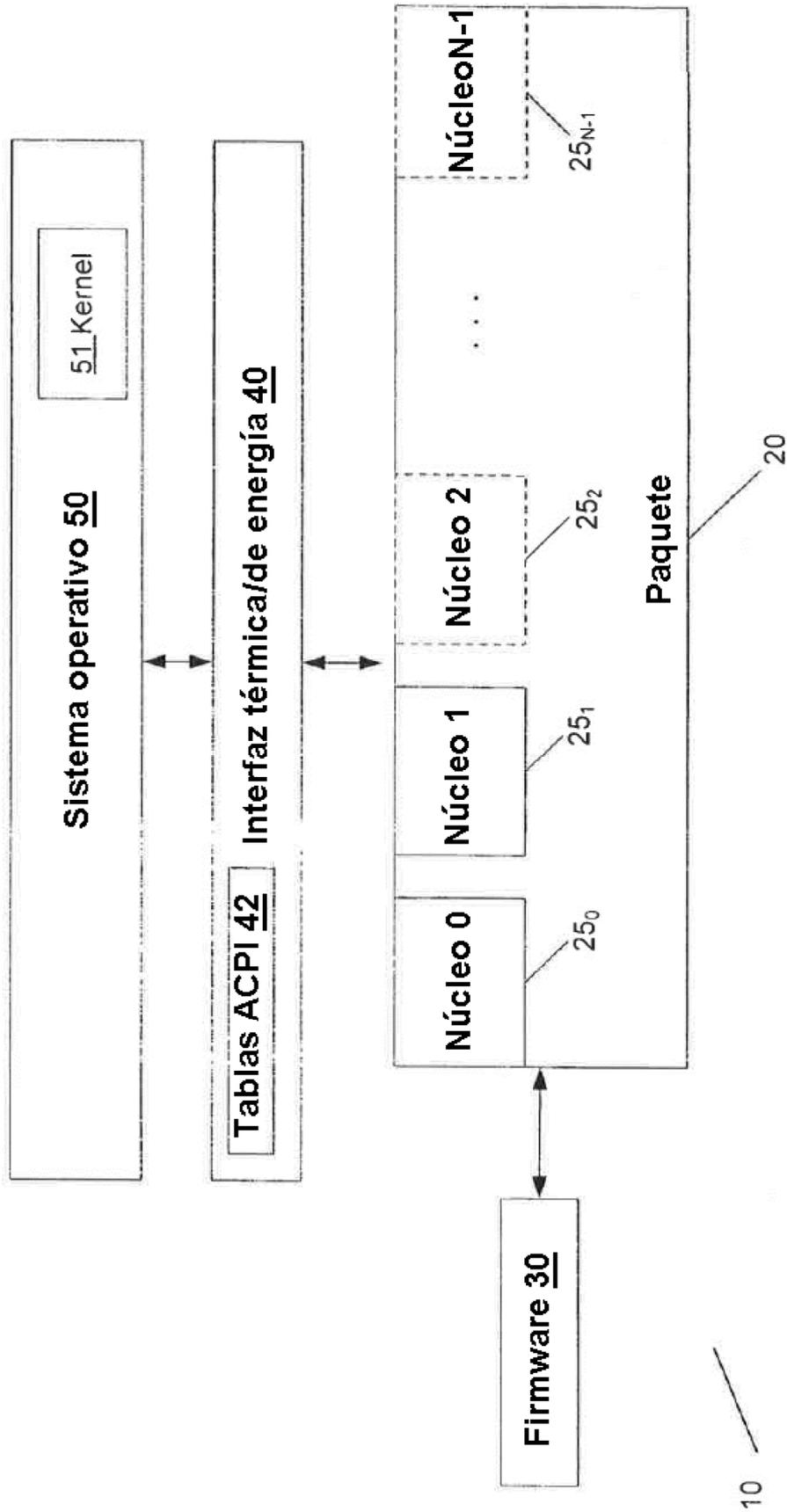
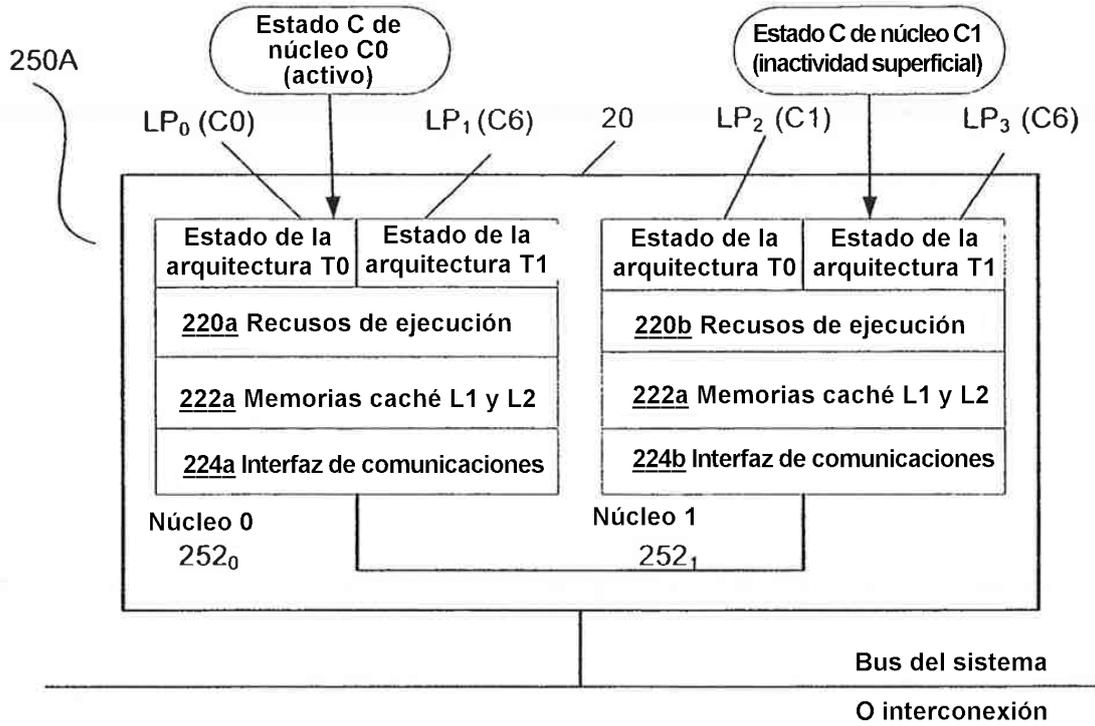


FIG. 1



ANTES

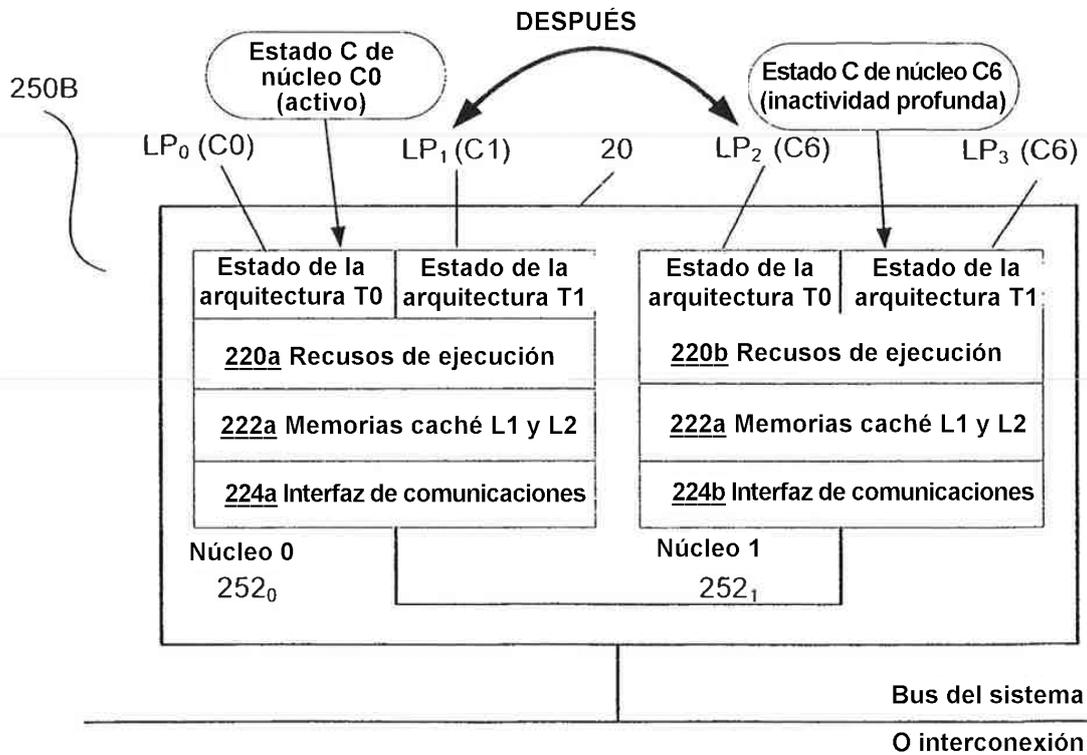


FIG. 2

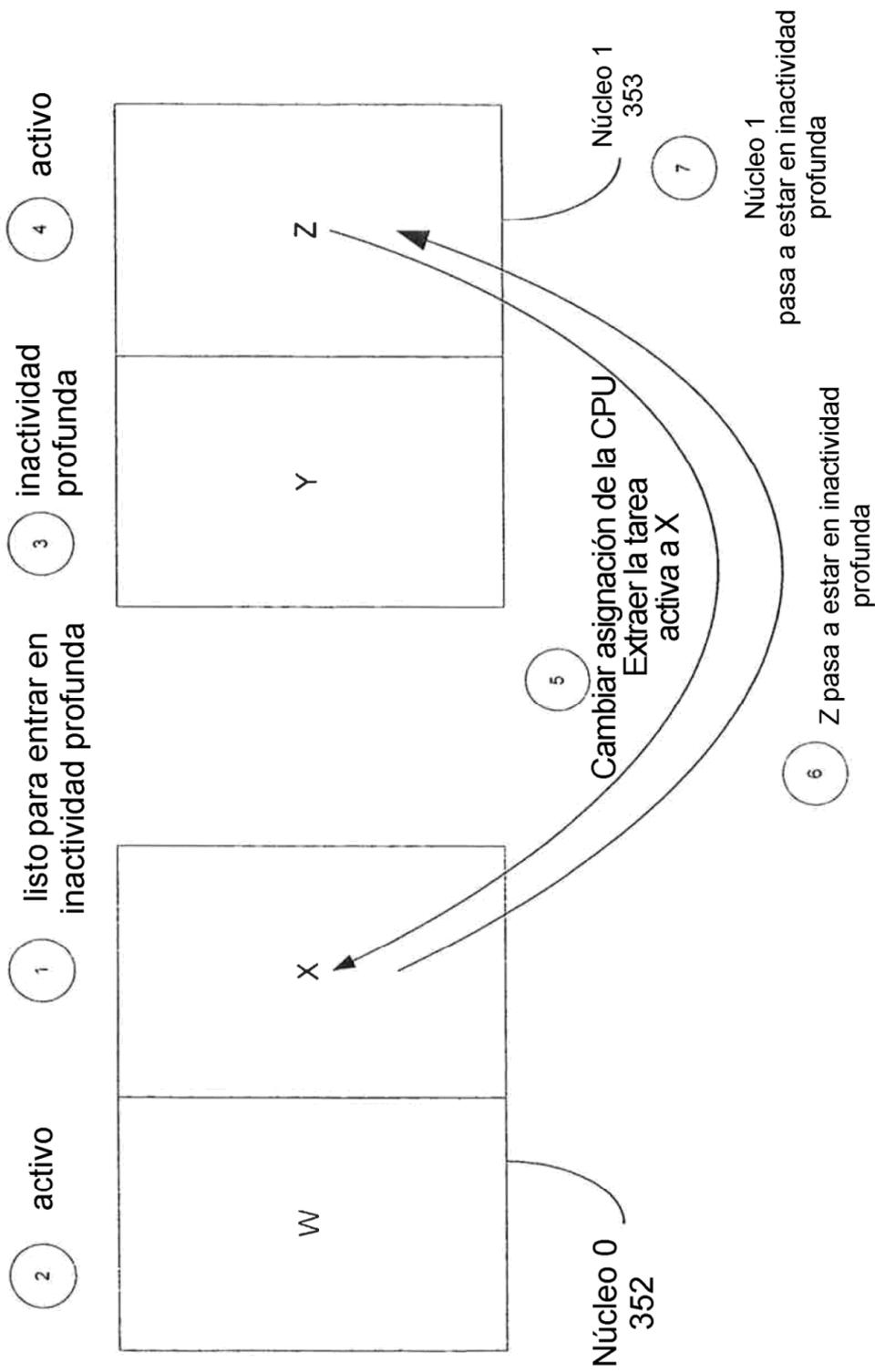


FIG. 3

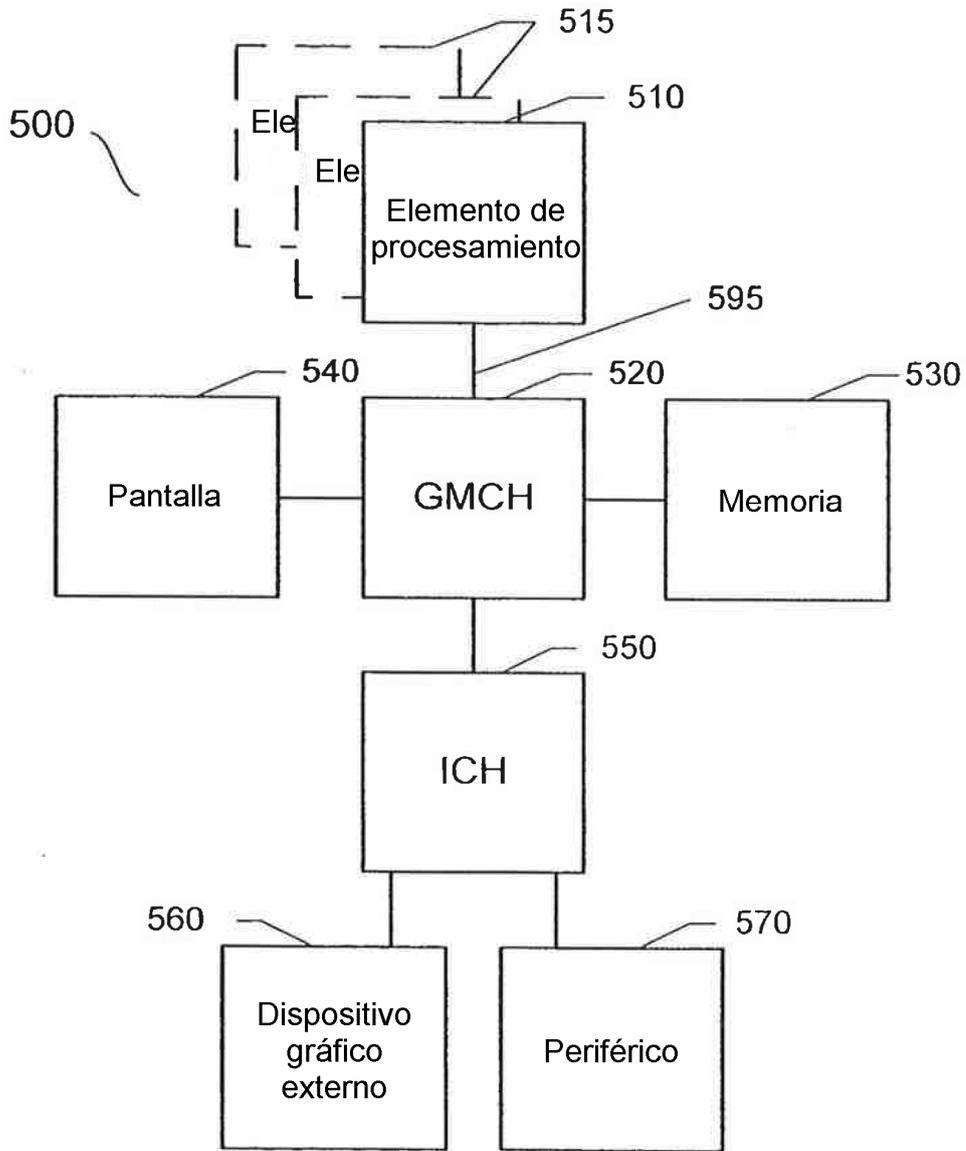


FIG. 5

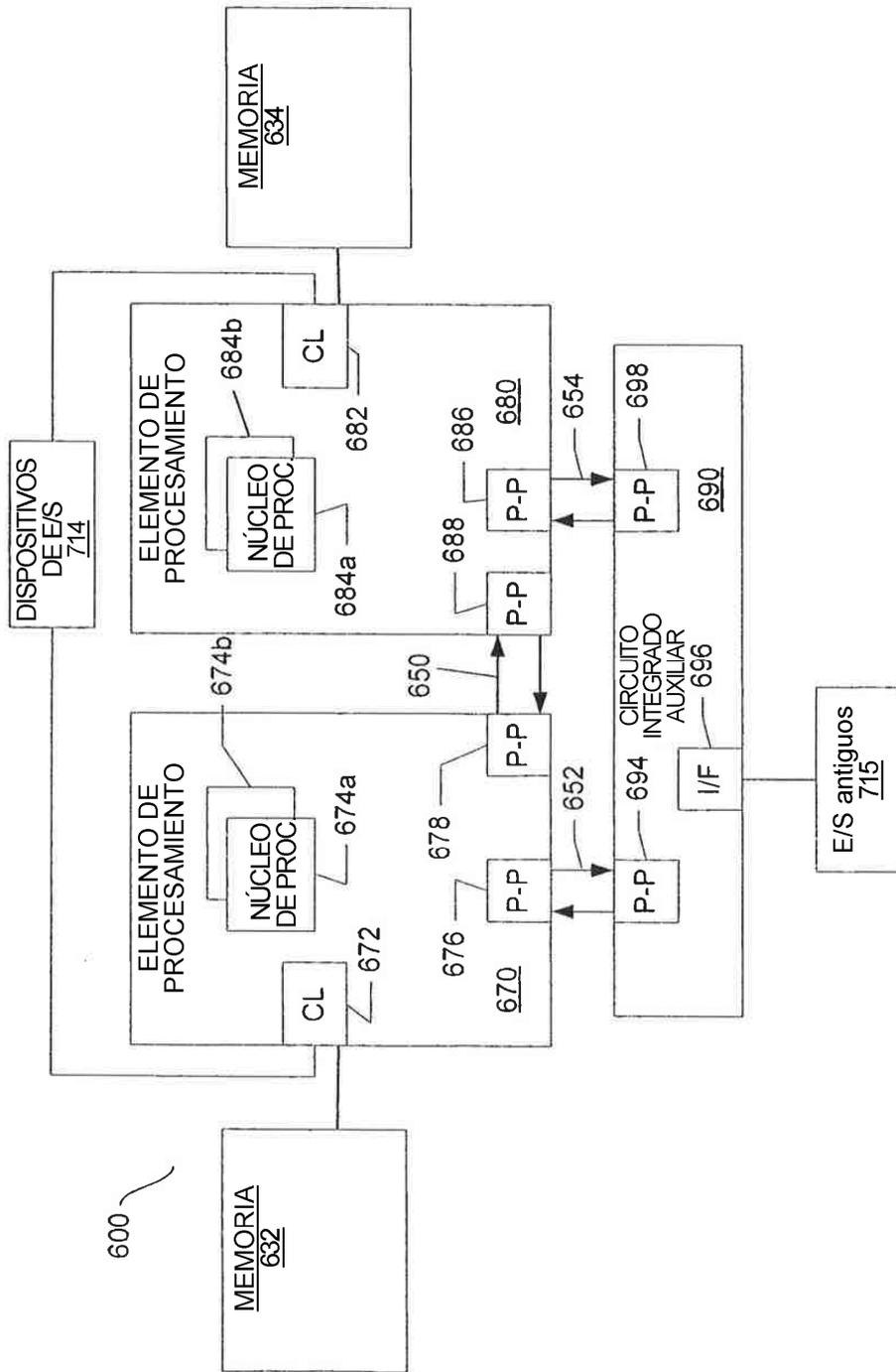


FIG. 7