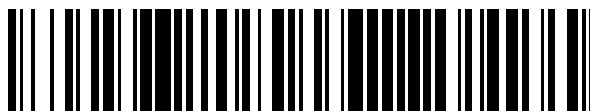


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 547 377**

51 Int. Cl.:

G06F 1/32

(2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **08.05.2006** **E 06759390 (5)**

97 Fecha y número de publicación de la concesión europea: **17.06.2015** **EP 1889140**

54 Título: **Circuitería para la predicción de elementos en reposo y lógica anti-atascamiento**

30 Prioridad:

10.05.2005 US 126442

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

05.10.2015

73 Titular/es:

QUALCOMM INCORPORATED (100.0%)
5775 Morehouse Drive
San Diego, CA 92121, US

72 Inventor/es:

DOCKSER, KENNETH ALAN

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 547 377 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

Circuitería para la predicción de elementos en reposo y lógica anti-atascamiento

5 **Campo técnico**

Las presentes enseñanzas se refieren a técnicas y arquitecturas de procesador para proporcionar de manera eficiente lógica de procesamiento complejo con menor consumo de energía cuando las funciones de procesamiento requieren menos que la capacidad completa de procesamiento.

10

Antecedentes

Muchos dispositivos utilizan procesadores integrados, tales como microprocesadores y procesadores de señales digitales, con arreglos complejos de lógica para la realización de funciones de procesamiento de datos de acuerdo con instrucciones del programa. Muchas aplicaciones de estos procesadores, por ejemplo en dispositivos portátiles tienen fuentes de alimentación por batería, necesitan un cuidadoso control del consumo de energía, por lo general, para prolongar la vida de una carga de la fuente de alimentación por batería. Muchas de las funciones o aplicaciones del procesador no requieren la capacidad completa de procesamiento del dispositivo procesador, o requieren la capacidad completa de procesamiento solamente durante un tiempo muy limitado. Si está totalmente alimentado de forma continua, sin embargo, la lógica no utilizada está consumiendo energía innecesariamente.

Tradicionalmente las puertas lógicas Complementary Metal Oxide Semiconductor (CMOS) han sido muy buenos a la hora de minimizar la energía consumida por los elementos en reposo, ya que sólo consumen energía dinámica cuando un circuito está conmutando activamente. Sin embargo, tienen también un componente estático, o de fuga, relacionado a su consumo de energía. Comúnmente, el componente dinámico ha superado ampliamente al estático. Sin embargo, en las tecnologías más nuevas submicronas-profundas, el componente estático está comenzando a contribuir con una cantidad significativa al consumo total de energía. Para poder llegar a ser energéticamente eficiente, los dispositivos modernos pueden necesitar limitar activamente la cantidad de energía estática disipada.

Para lograr reducción de energía, para aplicaciones embebidas alimentadas a pilas, como los teléfonos celulares, se han hecho intentos de apagar las partes no utilizadas de la lógica, por tanto eliminando la pérdida de energía de fuga. Algunos se han basado en el software de control para desactivar componentes lógicos no utilizados. Con este fin, las instrucciones en el programa pueden activar ciertos elementos del procesador cuando van a ser necesarios y desactivar cuando no sean necesarios. Si bien esto se traduce en ahorro de energía, impone una sobrecarga adicional sobre el software y requiere que el programador controle activamente diversas funciones del procesador.

Se pueden utilizar sistemas automatizados para apagar un componente no usado después de una determinada cantidad de tiempo. Sin embargo, surgen casos en los que el apagado de un elemento lógico en base al tiempo es potencialmente problemático. El despertar o activar el componente puede resultar en un consumo de energía adicional. En algunos casos, se necesita más para encender el elemento que el mantener el elemento encendido durante algún periodo de tiempo corto. Además, se necesita tiempo para volver a activar un componente. El retraso o latencia resultante en reiniciar el componente puede resultar en interrupciones de procesador, que degrada el rendimiento.

Por lo tanto, es conveniente asegurarse de que el componente no se apaga repetidamente sólo para volver a encenderse un poco más tarde. El apagado y encendido repetido se puede denominar "atasco". El documento US 2004/153263 A1 divulga aparatos controladores dinámicos de potencia, sistemas y procedimientos para controlar la energía a los componentes. Se emplean valores de tiempo de inactividad dinámicamente controlados basados en el uso histórico del componente. En un ejemplo, se ordena a un controlador de disco apagar un disco duro cuando su tiempo de inactividad equivale a un tiempo de reposo de control de potencia calculado dinámicamente. El documento WO 00/26756 A1 divulga un aparato para la gestión de energía para un circuito en un sistema basado en procesador. Se determina un tiempo de acceso al sistema del circuito y se determina si el tiempo de acceso al sistema es menor que un primer valor predeterminado. Si es así, se incrementa un período de accesibilidad del circuito. Existe la necesidad de una técnica para controlar selectivamente la energía a un elemento de un procesador, a fin de reducir efectivamente el consumo de energía a la vez que evitando el atasco indebido.

Resumen

En un aspecto, las enseñanzas de la presente divulgación se refieren a un procedimiento para reducir el consumo de energía de un procesador programable. El procedimiento implica monitorizar el tiempo desde una última petición para una operación de una de las unidades funcionales del procesador programable. El tiempo transcurrido desde una última parada anterior de la unidad también se controla. Cuando el tiempo desde la última petición supera un umbral actual, este procedimiento apaga la una unidad funcional, con el fin de reducir la energía consumida por la una unidad funcional. La unidad se reactiva en respuesta a una petición posterior para la operación de la una unidad funcional. El procedimiento también incluye el ajuste del umbral. En el momento de la reactivación de la una unidad

funcional, si el tiempo desde el apagado es inferior a un valor establecido, el procedimiento aumenta el umbral actual a un nuevo umbral más alto.

Por ejemplo, el valor establecido puede ser igual al valor de intervalo actual que se utiliza como el umbral. Si el tiempo de apagado a la siguiente petición es corto, por ejemplo, menos del umbral antiguo, el incremento en el umbral ayuda a reducir futuros atascos. El umbral puede también ajustarse de modo incremental, típicamente como una disminución en el umbral de tiempo, en respuesta al intervalo desde la última petición anterior. La gradual disminución ayuda a optimizar la conservación de energía.

Otros aspectos de las presentes enseñanzas se refieren a un control de energía y / o a un procesador de control de energía utilizando ajuste de umbral.

El control de energía puede incluir circuitería acoplada a una de las unidades funcionales, para activar y apagar selectivamente la una unidad funcional. El control incluye medios para el seguimiento de tiempo. En concreto, el tiempo se controla tanto desde una última petición previa para la operación de la una unidad funcional y desde el último apagado previo de la unidad. Medios de control apagan la una unidad funcional si el tiempo desde la última petición para la operación de la unidad funcional supera un valor actual de umbral. Los medios de control también sirven para aumentar el valor del umbral por encima de su valor actual, si una petición para la una unidad funcional se produce en un tiempo desde el último apagado previo de menos de un valor establecido.

Un aspecto de procesador de las presentes enseñanzas podría incluir unidades funcionales para el procesamiento de datos de acuerdo con instrucciones y circuitos acoplados a una de esas unidades, para activar y apagar selectivamente la una unidad funcional. Un controlador de energía supervisa peticiones de operación de la una unidad funcional durante el procesamiento de instrucciones y controla el circuito para apagar la unidad. El controlador apaga la una unidad funcional a medida que pasa cada un intervalo de tiempo desde una de las últimas peticiones previas igual a un umbral ajustable. El controlador también reactiva la unidad funcional en respuesta a cada nueva petición de operación de la una unidad funcional después de un apagado. También se proporcionan medios para ajustar el umbral.

El ejemplo divulgado incluye lógica de control que supervisa el uso de un elemento funcional particular (por ejemplo, un divisor, o multiplicador, o similares) y apaga la unidad cuando no se ha utilizado durante un período de tiempo especificado. Un sello de tiempo se almacena cada vez que hay una petición de operación de la unidad. La lógica emplea un contador (local o central) para determinar cuándo ha transcurrido un período establecido sin el uso del elemento. La lógica también almacena un sello de tiempo para cada apagado y supervisa cuan pronto se despierta la lógica de nuevo para determinar si el control de energía está causando atascamiento. Tras la determinación de tal atascamiento, la unidad ajusta automáticamente su periodo de umbral, para minimizar el atascamiento. Del mismo modo, cuando la lógica determina que está siendo demasiado conservador, disminuye gradualmente el umbral. Existen bits de modo para permitir al programador sobrescribir la lógica de apagado para o bien mantener la lógica siempre encendido, o siempre apagado.

Por lo tanto, la técnica a modo de ejemplo permite que el hardware reduzca dinámicamente el consumo de energía en base de fugas basado en el tiempo de actividad de las unidades funcionales específicas. La lógica anti-atascamiento asegura que los esfuerzos para reducir la energía no causan consumos indebidos de energía o interrupciones de procesador debidos a reciclamientos excesivos. Sin embargo, cuando no está atascado, el ajuste incremental ayuda a optimizar la conservación de energía.

Las ventajas adicionales y características nuevas se expondrán en parte en la descripción que sigue, y serán evidentes para los expertos en la técnica tras el examen de lo siguiente y los dibujos adjuntos, o pueden aprenderse mediante la producción u operación de los ejemplos. Las ventajas de las presentes enseñanzas pueden ser realizadas y conseguidas por la práctica o el uso de las metodologías, instrumentos y combinaciones particularmente señaladas en las reivindicaciones adjuntas.

Breve descripción de los dibujos

Las figuras de los dibujos muestran una o más implementaciones de acuerdo con los presentes conceptos, a modo de ejemplo solamente, no a modo de limitación. En las figuras, números de referencia iguales se refieren a los mismos o similares elementos.

La FIG. 1 es un diagrama de bloques simplificado de un procesador segmentado con control ajustable de una unidad funcional del procesador.

La FIG. 2 es un diagrama de bloques funcional de un ejemplo de un control de energía, para controlar la activación de un divisor, basado en un umbral de tiempo ajustable.

La FIG. 3 es un diagrama de flujo de un ejemplo sencillo del procesamiento para activar y desactivar una unidad funcional del procesador, basado en la sincronización con respecto a la actividad de la unidad y un umbral de tiempo ajustable.

- 5 La FIG. 4 es un diagrama de flujo de un ejemplo del procesamiento que se puede utilizar para ajustar el umbral, para optimizar el ahorro de energía mientras se controla el atascamiento.

Descripción detallada

- 10 En la siguiente descripción detallada, numerosos detalles específicos se exponen a modo de ejemplo con el fin de proporcionar una comprensión completa de las enseñanzas pertinentes. Sin embargo, debería ser aparente para los expertos en la técnica que las presentes enseñanzas pueden ponerse en práctica sin tales detalles. En otros casos, conocidos métodos, procedimientos, componentes y circuitos se han descrito relativamente con un alto nivel, sin detalle, a fin de evitar oscurecer innecesariamente aspectos de las presentes enseñanzas.

- 15 Los diversos ejemplos de la arquitectura de procesador y el procesamiento de flujos descritos a continuación controlan la operación de una unidad funcional particular (por ejemplo, un divisor, o multiplicador, o similar) sobre la base de las peticiones monitorizadas para la operación de dicha unidad, por ejemplo, para apagar la unidad funcional cuando no ha sido utilizada (por ejemplo, sin ninguna nueva petición) para un cierto período. El establecimiento del período o umbral de tiempo, sin embargo, se ajusta en función del uso del elemento. El ajuste del umbral puede disminuir el umbral gradualmente, para mejorar aún más el consumo de energía, ya que el elemento controlado se utiliza con menos frecuencia. Para evitar atascamiento, si una instrucción re-petición a la unidad brevemente seguido a un apagado, por ejemplo, en un período inferior al umbral actual establecido, entonces el procesador ajusta el umbral para aumentar el período de inactividad antes de su corte posterior.

- 25 Se hace ahora referencia en detalle a los ejemplos ilustrados en los dibujos adjuntos y discutidos a continuación. Los principios de control de energía descritos en este documento son adaptables a una variedad de diferentes arquitecturas de procesador. Sin embargo, para facilitar la comprensión, puede ser útil considerar un procesador en cadena como un ejemplo. La FIG. 1 es un diagrama de bloques simplificado de un procesador en cadena 10. El procesador 10 está diseñado para aplicaciones de baja energía, por ejemplo, en las estaciones de telefonía móvil u otros dispositivos electrónicos portátiles que utilizan energía de la batería. En tales aplicaciones, la técnica es aplicable al procesador principal o a un coprocesador numérico, como un procesador de instrucción única – datos múltiples (SIMD) para el procesamiento multimedia.

- 35 Para facilitar la discusión, el ejemplo de una cadena de procesamiento 10 es un diseño escalonado, esencialmente implementando una sola cadena. Los expertos en la técnica entenderán, sin embargo, que el procesamiento discutido en este documento también es aplicable a diseños super-escalares y otras arquitecturas implementando múltiples cadenas. Además, la profundidad de la cadena (por ejemplo, el número de etapas) es sólo ilustrativa. Una cadena de procesamiento real puede tener menos etapas o más etapas que la cadena 10 en el ejemplo.

- 40 La cadena simplificada 10 incluye cinco categorías principales de etapas de procesamiento de cadena: Búsqueda 11, Decodificación 13, Lectura 15, Ejecución 17 y Sobre-escritura 19. Las flechas en el diagrama representan flujos de datos lógicos, no necesariamente conexiones físicas. Los expertos en la técnica reconocerán que cualquiera de estas etapas puede ser descompuesta en múltiples etapas que realizan porciones de la función correspondiente, o que la cadena puede incluir etapas adicionales para proporcionar funcionalidad adicional. Aunque no se muestra por separado en la FIG. 1, cada etapa de la cadena 10 comprende típicamente una máquina de estado o similar implementando las funciones lógicas pertinentes y un registro asociado para pasar la instrucción y / o cualquier resultados de procesamiento a la siguiente etapa o para otros recursos del procesador.

- 50 En la cadena 10 a modo de ejemplo, la primera etapa es la etapa de Búsqueda 11. La etapa de Búsqueda 11 obtiene instrucciones para el procesamiento por las etapas posteriores. La etapa de Búsqueda 11 obtiene las instrucciones de una jerarquía de memorias (no mostrada), que típicamente incluyen un caché de instrucción o nivel 1 (L1), un caché de nivel 2 (L2) y una memoria principal. La etapa de Búsqueda 11 suministra cada instrucción a una etapa de Decodificación 13. La lógica de la etapa de Decodificación de instrucciones 13 decodifica los bytes de instrucción recibidos y suministra el resultado a la siguiente etapa de la cadena.

- La siguiente categoría importante de etapas proporciona acceso a datos o Lectura. La lógica de la etapa de Lectura 15 accede a datos de operando en los registros especificados en un registro de propósito general o archivo "GPR" (no mostrado). Cada instrucción y sus datos operandos requeridos se pasan a una o más etapas 17 que proporciona la función de Ejecución. La(s) etapa (s) de Ejecución 17 esencialmente ejecutan la función particular de cada instrucción en los datos de operando recuperados y producen un resultado. La etapa o etapas 17 que proporcionan la función de Ejecución pueden, por ejemplo, implementar una unidad de lógica aritmética (ALU). La etapa de Ejecución 17 suministra el resultado o resultados de la ejecución de cada instrucción a la etapa de Sobre-escritura 19. La etapa 19 sobre-escribe los resultados a un registro o en memoria.

- 65 Los elementos del procesador 10 consumen energía incluso cuando no están conmutando. En muchas aplicaciones,

por ejemplo, utilizando fuentes de energía por batería, la vida de la fuente de energía es limitada. Algunos elementos del procesador 10 no se utilizan continuamente. Algunas unidades funcionales, en la etapa de Ejecución 17 del ejemplo, pueden no ser necesarios durante períodos sustanciales de tiempo. Para reducir el consumo de energía, cada unidad funcional de este tipo se puede apagar cuando no se necesita. Para propósitos de discusión, la unidad funcional que va a ser el sujeto del control de energía es un divisor 21, aunque los expertos en la técnica reconocerán que un control similar se puede aplicar a un multiplicador o cualquiera de una variedad de otros elementos funcionales o recursos de la etapa de Ejecución 17 o en otro lugar en el procesador 10.

El ejemplo incluye un control de energía 23, que controla las peticiones y operación de una unidad funcional particular (por ejemplo, el divisor 21). El Control 23 apaga la unidad cuando no ha sido peticionado por un período de tiempo definido por un umbral ajustable en el registro 25. El control de energía 23 determina cuando ha transcurrido un período establecido sin petición para usar por el divisor 21, es decir, cuando el período de inactividad del elemento 21 equivale o supera el umbral 25. Como se discutirá más tarde, el control de energía 23 también monitoriza cuan pronto la unidad funcional, en este caso el divisor 21, se enciende de nuevo después de cada apagado para determinar si el control de energía está causando atascamiento. Tras la determinación de tal atascamiento, el control de energía 23 ajusta automáticamente hacia arriba su intervalo umbral, para minimizar el atascamiento. Del mismo modo, cuando el control de energía 23 determina que está siendo demasiado conservador, disminuye su umbral 25. Existen bits de modo para permitir que el programador anule la lógica de apagado para o bien mantener la unidad siempre encendido, o siempre apagado.

El estado activo de la unidad funcional controlada, es decir, el divisor 21 en el ejemplo de la FIG. 1, puede ser controlado en respuesta a señales de selección del control de energía 23 en cualquiera de una variedad de formas conocidas. Para el propósito de discusión general, la FIG. 1 muestra una señal (S) necesaria para la operación del divisor 21, y un circuito de puerta genérico 27 para acoplar selectivamente la señal al divisor en respuesta a unas señales de control de energía desde el control de energía 23. El circuito mostrado genéricamente en 27 en la FIG. 1 puede ser una puerta lógica, un conmutador, una combinación de los mismos, o cualquier otro circuito configurado para suministrar el tipo apropiado de señal a la unidad funcional en respuesta a las señales de selección apropiadas desde el control 23. Suministrar la señal S a la unidad 21 activa o enciende la unidad; mientras que la interrumpir o retirar la señal S apaga la unidad 21.

Por ejemplo, el divisor 21 se puede activar y desactivar de forma selectiva mediante conmutación controlada del suministro de energía al divisor 21, de tal manera que la operación de la puerta 27 enciende y apaga la energía de acuerdo con las peticiones de uso. En tal implementación, la señal S representaría uno de los terminales de suministro de energía o voltajes. Cuando el control de energía 23 desactiva el divisor 21, el control 23 activa la puerta 27 para cortar una conexión con uno de los terminales de energía S (por ejemplo, suministro o tierra) con respecto a la circuitería del divisor 21. El corte elimina el consumo dinámico de energía y las fugas a través de la circuitería de esa unidad funcional.

El ejemplo de la FIG. 1 muestra una única unidad funcional, el divisor 21, controlado basado en peticiones para su uso. Los expertos en la técnica reconocerán que un procesador dado puede incluir varios controles similares para un número de elementos que consumen energía de forma sustancial, pero que pueden estar inactivos durante períodos de tiempo considerables.

La FIG. 2 es un diagrama de bloques funcional de una implementación del control de energía 23, para controlar la activación del divisor 21 sobre la base de un umbral ajustable en el registro 25. En este ejemplo, el control de energía 23 incluye un contador 31 que responde a una señal de reloj. El contador 31 puede utilizar el reloj de sistema u otra señal de reloj generada en o para el procesador 10. El contador 31 puede ser compartido entre varias unidades funcionales, o dedicado a la unidad funcional en cuestión. El conteo es en unidades de tiempo razonables para la unidad funcional. Puede variar desde ciclos a segundos (o más allá). El control de energía 23 incluye también dos registros de sellos de tiempo (TS) 33 y 35. Cuando activado, un registro 33 o 35 almacena el conteo actual del contador 31, lo que representa una medida o valor relacionado con la hora actual.

El registro 33 se activa para almacenar el valor actual de recuento de tiempo desde el contador 31, cada vez que la unidad funcional (divisor 21 en nuestro ejemplo) se peticiona por una instrucción que fluye a través de la cadena de procesador 10. El registro 33 mantiene la cuenta hasta que la unidad se peticiona de nuevo. De esta manera, el valor TS(LC) en el registro 33 representa un sello de tiempo para la petición última (más reciente) para la operación del divisor 21.

El registro 35 se activa para almacenar el valor actual de recuento de tiempo del contador 31, cada vez que la unidad funcional (divisor de 21 en nuestro ejemplo) se apaga. El registro mantiene ese cuento hasta que la unidad se apaga de nuevo. De esta manera, el valor TS(LPD) en el registro 33 representa un sello de tiempo para el último apagado (la desactivación más reciente) del divisor 21.

El contador 31 suministra también recuentos relacionados con el tiempo de operación a dos circuitos de sustracción 37 y 39. En operación, el circuito de sustracción 37 también recibe el último valor del sello de tiempo de petición TS(LC) del registro 33. La diferencia I_c entre el contador (relacionado a la hora actual) y el valor de conteo

almacenado TS(LC) representa el intervalo de inter-retardo desde la última petición previa para el divisor 21. Un comparador 41 compara el intervalo de inter-retardo desde la última petición I_C con el valor actual del umbral ajustable 25. De esta manera, el valor del contador se resta de los datos en el registro 33 y se compara con el umbral actual registrado en 25. Alternativamente, esto se puede hacer restando el sello de tiempo antiguo en el registro 33 y el umbral 25 del valor actual del contador 31, en cuyo caso, un resultado cero o negativo significa un período igual al umbral ha pasado desde la última petición para el divisor 21.

Si I_C alcanza (es decir, equivale o excede) el valor actual del umbral 25, entonces la lógica de control 45 desactiva la puerta 27, para cortar el divisor 21. Cuando la lógica de control desactiva el divisor 35, también causa que el registro 35 almacene un nuevo recuento del marcador 31 como el valor TS(LPD) de sello de tiempo del último apagado. La lógica de control enciende el divisor 21 en respuesta a una petición posterior para el divisor 21.

El segundo circuito de sustracción 39 recibe el valor TS(LPD) de sello de tiempo del último apagado (LPD) del registro 35. La diferencia I_P entre el valor en el contador 31 (relacionado a la hora actual) y el valor de conteo almacenado TS(LPD) representa el intervalo de inter-retardo desde el último apagado del divisor 21. Un comparador 43 compara el intervalo desde el último apagado I_P a un valor establecido, en este ejemplo el valor actual del umbral ajustable 25, y suministra el resultado a la lógica de control 45. La lógica de control 45 utiliza este resultado de la comparación en su algoritmo interno para ajustar el umbral 25 para evitar atascamiento. Alternativamente, esta comparación se puede hacer restando el sello de tiempo antiguo en el registro 35 y el umbral 25 del valor actual del contador 31, en cuyo caso, un resultado positivo significa que un período menor que el umbral ha pasado desde el último de apagado. El circuito de sustracción 37 y el comparador 41 se activan cuando el divisor se peticiona o se activa, como parte del procesamiento para ajustar el umbral 25.

El algoritmo implementado por la lógica de control 45 para ajustar el período utilizado para el apagado basado en el uso está destinado a optimizar el consumo de energía pero a la vez minimizar atascamiento. Por ejemplo, si el control 23 no esperó un tiempo suficiente antes de apagar el divisor de 21 la última vez, la lógica 45 amplía el umbral utilizado como medida del periodo de inactividad para el siguiente ciclo de energía con respecto a la unidad funcional particular. Sin embargo, el algoritmo también disminuye el umbral un poquito (afinar), por ejemplo, cada vez que peticiona la función cuando el retardo entre peticiones fue mayor que el umbral.

En primer lugar considerar las operaciones que puedan estar involucradas en afinar el umbral hacia abajo, para reducir aún más el consumo de energía.

Un umbral mínimo de apagado está diseñado o cargado por software en el registro de Umbral Mínimo 47. El valor de umbral inicial (es decir, Th_0) en el registro 47 es el valor mínimo leído del registro 25. Típicamente, el umbral mínimo se establece al valor en el que el ahorro de energía de apagado (fuga) es igual al costo de encendido, aunque el mínimo se puede establecer inferior a este valor. También se podría establecer mayor, para tener en cuenta el costo del retardo (por ejemplo, interrupciones, vaciados, y búsquedas repetidas) inducido en el flujo de instrucciones a la espera del inicio de la unidad funcional controlada.

Cuando se peticiona o activa la unidad funcional (por ejemplo, el divisor 21), el valor en el valor actual del contador 31 es capturado en el registro correspondiente 33 o 35 como un sello de tiempo. Además, cuando la unidad funcional se peticiona o la unidad se enciende de nuevo, el valor actual del contador 31 se resta de los datos en el registro correspondiente y se compara con el umbral actual, por ejemplo, utilizando los circuitos comparadores y de sustracción discutidos anteriormente. Alternativamente, esto se puede hacer restando el respectivo sello de tiempo y el umbral del valor actual del contador - un valor negativo significa que el evento ocurrió antes de un intervalo de umbral más allá del último evento.

Esencialmente, cada vez que el control de energía 23 detecta una nueva petición para la unidad funcional, el control de energía ejecuta dos algoritmos para ajustar el valor umbral en el registro 25. Un algoritmo proporciona pequeños ajustes o afinaciones, para optimizar el ahorro de energía mediante el control 23. El otro algoritmo proporciona ajustes más grandes, para minimizar el atascamiento.

Para la afinación, cada vez que hay una petición para la función de la unidad, por ejemplo para la división por la unidad 21, se determina cuánto tiempo ha pasado desde la última petición mediante la comparación de los sellos de tiempo (la salida del contador) para las peticiones. Este valor I_C , para inter-retardo entre peticiones, se resta del antiguo umbral Th_n , y la diferencia (Δ) se desplaza hacia la derecha por un valor constante, por ejemplo, 3 lugares, esencialmente como si dividiendo por el correspondiente poder de dos. Este resultado se resta del valor antiguo de umbral Th_n , para producir un nuevo umbral Th_{n+1} . Un ejemplo de tal fórmula para ajustar el umbral sería la siguiente:

$$Th_n - \frac{Th_n - I_C}{2^x} = Th_{n+1} .$$

(1)

Sobre la base de un cálculo de este tipo, la lógica de control 45 ajusta el valor de umbral en el registro 25. Este procesamiento en respuesta al intervalo entre las peticiones produce un ajuste gradual del umbral. Puesto que un inter-retardo I_C muy grande podría provocar grandes cambios en el umbral, se podría saturar la diferencia (Δ) a un valor máximo (por ejemplo, el umbral anterior Th_n) para que ninguna sola lectura pudiera afectar indebidamente el umbral. Así, por ejemplo, el nuevo umbral podría siempre ser mantenido dentro del rango de:

$$Th_n - \frac{Th_n}{2^x} \leq Th_{n+1} \leq Th_n + \frac{Th_n}{2^x}.$$

Como valores I_C muy grandes son propicios al ahorro de energía, en realidad se podría reducir el umbral (en lugar de aumentarlo) por

$$\frac{Th_n}{2^x}$$

en tales casos; para los propósitos de este ejemplo, un valor muy grande de I_C se podría definir como $> Th_n 2^x$. Además, el umbral mínimo podría ser utilizado como un límite en la disminución del umbral.

Asumir como un primer ejemplo que la unidad funcional estaba encendida, y el tiempo desde la última petición no ha pasado el umbral actual Th_n cuando se recibe otra petición. En este ejemplo, la lógica de control 45 ha mantenido la unidad funcional 21 encendida. Cuando se recibe la nueva petición para la unidad, el inter-retardo I_C es menor que el umbral actual Th_n . La diferencia entre Th_n e I_C sería un valor Δ positivo. Por lo tanto, el umbral Th_{n+1} nuevo se reduciría o menor que el umbral Th_n antiguo por la cantidad de Δ desplazada por x lugares (dividido por 2^x). Dicho de otra manera, el control de energía no ha sido lo suficientemente agresivo, por lo que reduce gradualmente el umbral en un esfuerzo por apagar la unidad (ahorrando de energía) antes en el futuro.

Asumir como un segundo ejemplo, que el tiempo desde que la última petición ha pasado el umbral actual, y la lógica de control 45 ha apagado la unidad funcional 21. Cuando se peticona seguidamente a la unidad, el inter-retardo I_C es mayor que el umbral actual Th_n . Si no hay atascamiento, I_P es mayor que o igual a Th_n , por lo que I_C es al menos el doble del valor umbral actual Th_n . La diferencia sería un valor Δ negativo. Por lo tanto, el umbral Th_{n+1} nuevo en la fórmula (1) se incrementaría o más grande que el umbral Th_n antiguo, por la cantidad de Δ desplazado por x lugares (dividido por 2^x). Sin embargo, si el inter-retardo I_C fuera particularmente grande, el control de energía reduciría el nuevo umbral en el ejemplo de Th_{n+1} a $Th_n(1 - 2^x)$.

La lógica de control 45 también ajusta el valor umbral en el registro 25 para minimizar el atascamiento. El atascamiento se produce cuando, después de un apagado, se recibe una nueva petición para la operación de la unidad funcional demasiado pronto, por ejemplo, de manera que el encendido consume más energía que fue ahorrado por el apagado y / o interrupciones agregado a retrasos al encender afectando significativamente al rendimiento. Si la función de la unidad en cuestión ha sido apagada, cuando se le peticona otra vez y se vuelve a encender, el tiempo de activación se compara con el sello de tiempo de apagado $TS(LPD)$.

En el ejemplo de la FIG. 2, cuando la lógica de control 45 enciende el divisor 21 en respuesta a una nueva petición para su operación, la lógica lee la comparación del valor de conteo en el contador 31 a la combinación del sello de tiempo en el registro 35 y el valor umbral en el registro 25. Se utiliza aquí por conveniencia el mismo umbral, aunque un valor de tiempo establecido diferente podría ser utilizado en este algoritmo de ajuste. Si el intervalo de tiempo medido desde el último apagado es demasiado pequeño (por ejemplo, el período de retardo entre encendido y apagado es menor o igual que el umbral), el umbral se ajusta de una forma destinada a reducir el atascamiento potencial futuro. Este ajuste aumenta el umbral para uso futuro, por ejemplo a dos veces el umbral antiguo. Por lo tanto, el componente anti-atascamiento tiene un efecto más directo sobre el umbral, mientras que el componente de afinación tiene un efecto más suavizado.

El procesamiento para controlar la activación y desactivación de un elemento funcional de un procesador basado en la inactividad y para ajustar el umbral de tiempo de inactividad puede ser implementado en una variedad de maneras. Sin embargo, puede ser útil considerar un ejemplo de un flujo lógico de procesamiento de acuerdo con las operaciones descritas anteriormente. La FIG. 3 es un diagrama de flujo que ilustra un ejemplo del flujo de procesos que podrían ser implementados por el control de energía 23.

Cuando el divisor 21 ya está encendido (en S1), se calcula en la etapa S2 el retraso I_C entre peticiones desde la última petición previa para la operación del divisor 21. Utilizando el registro 33 y el circuito de sustracción 37, por ejemplo, la etapa S2 calcula $I_C = \text{Count} - TS(LC)$. Una etapa S3 implica comparación con el valor umbral actual Th_n (por ejemplo, como en el comparador 41). Si la lógica de control 45 determina que el retardo I_C de inactividad desde la última petición no cumple o excede el valor umbral actual Th_n , el procesamiento en la etapa S3 sigue a la etapa S4.

En la etapa S4, la lógica de control 45 comprueba para una petición para la operación del divisor 21. Si no hay ninguna petición, entonces el procesamiento en la etapa S4 fluye a la etapa S5. En la etapa S5, el contador se

incrementa y el procesamiento vuelve a la etapa S2 para actualizar el valor del inter-retardo desde la última petición y para comprobar ese retraso de nuevo contra el valor umbral actual en S3.

Asumir con fines de discusión en este punto que el bucle de las etapas S2 a S5 continúa hasta que el control de energía detecta una petición para la operación del divisor 21. Por lo tanto, en la etapa S4, la lógica de control 45 detecta la petición para el divisor 21, y el proceso fluye desde la etapa S4 a la etapa S6. En la etapa S6, la lógica de control implementa una rutina para ajustar el valor umbral, como se discutirá más adelante, en relación con el diagrama de flujo de la FIG. 4. Es de destacar que, puesto que la unidad no se ha apagado, en este punto del ejemplo el ajuste disminuye gradualmente el umbral para uso futuro. Después del ajuste del umbral, el procesamiento implica la actualización del sello de tiempo TS(LC) en el registro 33 (etapa S7) al valor actual en el contador 31 en el ciclo en el que el control 23 recibió o detectó la petición para el divisor 21. La etapa 5 de nuevo incrementa el contador, y el procesamiento vuelve a la etapa S2 para reanudar el análisis basado en el inter-retardo I_C desde la última petición para la función del divisor 21.

Los bucles de los pasos S2 a S7 continuarán siempre y cuando la lógica de control 45 reciba una nueva petición para la función del divisor 21 antes de la expiración de un período del inter-retardo I_C desde la última petición para la función del divisor 21 igual o mayor que un valor actual Th_n del umbral. Sin embargo, supongamos ahora que el retraso desde la última petición alcanza el valor umbral actual. Por lo tanto, en la etapa S3, la lógica de control 45 detecta que $I_C \geq Th_n$, y hace que el procesamiento fluya desde la etapa S3 a S8. En la etapa S8, la lógica de control 45 desactiva el divisor 21, por ejemplo, cortando la alimentación al divisor 21 como se discutió anteriormente. La lógica de control 45 a continuación (en S9) activa el registro 35 para capturar el valor de la cuenta actual como el nuevo valor de sello de tiempo TS(LPD) para el último apagado del divisor 21.

En la etapa S10, la lógica de control verifica si se ha recibido o se detecta una nueva petición para la función del divisor 21. Si no, el procesamiento fluye a la etapa S11 en el que se incrementa el contador, y el procesamiento fluye de vuelta a la etapa S10. Las etapas S10 y S11 se repiten hasta que la lógica de control 45 detecta una nueva petición para la función del divisor 21. Hasta que se detecta una nueva petición, el divisor 21 permanece desactivado y por lo tanto no consume energía.

Sin embargo, cuando la lógica de control 45 detecta una nueva petición para la función del divisor 21, el procesamiento fluye desde S10 a S12. En S12, la lógica de control 45 inicia el procedimiento para el encendido de la unidad funcional, es decir, el divisor 21 en nuestro ejemplo. El procesamiento fluye entonces a la etapa S6 para ajustar el umbral, ya sea para afinar el umbral o para aumentar el umbral para evitar atascamiento, dependiendo de la longitud de tiempo entre las peticiones. En la etapa S7, la lógica 45 establece el sello de tiempo para la última petición TS(LC) al valor actual del contador 31. El procesamiento entonces fluye de vuelta a través de la etapa S5 a la etapa S2 para reanudar el procesamiento discutido anteriormente.

Como se muestra en la FIG. 3 y la descripción de los mismos en los párrafos anteriores, cada vez que hay una petición para la operación de la unidad funcional, la lógica de control 45 iniciará un procedimiento en la etapa S6 para ajustar el umbral. El procesamiento en S6 para ajustar el umbral de acuerdo con los principios expuestos anteriormente puede implementarse de una variedad de maneras. Sin embargo, puede ser útil considerar un ejemplo de un flujo lógico de procesamiento de acuerdo con las operaciones mencionadas anteriormente, con respecto al diagrama de flujo de la FIG. 4.

En la etapa S61, se calcula el intervalo I_P desde el último apagado del divisor 21. Utilizando el registro 35 y el circuito de sustracción 39, por ejemplo, la etapa S61 calcula $I_P = \text{Count} - \text{TS(LPD)}$. En la etapa S62 la lógica 45 determina si esta operación de ajuste siguió un encendido del divisor 21. Si es así, el procesamiento fluye desde S62 a S63. La etapa S63 implica la comparación con un valor establecido utilizado para medir atascamiento, por ejemplo, al valor umbral Th_n antiguo (por ejemplo, como en el comparador 43). Si la lógica de control 45 determina que el intervalo I_P desde el último apagado es menor que el valor umbral Th_n antiguo, entonces hay un problema de atascamiento, por lo que el procesamiento en la etapa S63 pasa a la etapa S64.

En la etapa S64, el umbral se incrementa para reducir atascamiento. Este ajuste en S64 puede producir un aumento del umbral igual a, por ejemplo, el doble del umbral antiguo ($Th_{n+1} = 2(Th_n)$). Por lo tanto, al detectar una condición de atascamiento, el umbral de apagado puede duplicarse, para evitar apagados precoces del divisor en ciclos futuros.

Volviendo a la etapa S62, si la unidad funcional no se encendió en respuesta a la petición nueva (el intervalo de umbral desde la última petición previa no había expirado) el procesamiento fluiría a S65. El procesamiento también puede llegar a S65 si hay un encendido en respuesta a la nueva petición, pero el intervalo I_P desde el último apagado era mayor que o igual al valor umbral antiguo. Esto significa que el período transcurrido desde la última petición previa es también mayor que el umbral (ya que la última petición previa precedió al último apagado) y la unidad funcional se había encendido en la etapa S12 en este paso a través del flujo de proceso de la FIG. 3. En cualquier caso, no hay ningún problema de atascamiento, pero el umbral puede necesitar ser afinado para optimizar la conservación de energía.

Como se ha señalado, valores muy grandes de I_C son propicios para el ahorro de energía. En este ejemplo, un valor muy grande de I_C se define como $> Th_n 2^x$, y la lógica comprueba el inter-retardo contra este valor definido en la etapa S65. Si el inter-retardo $I_C > Th_n 2^x$, el procesamiento fluye a la etapa S66 en el que el umbral se reduce por

$$\frac{Th_n}{2^x}$$

(el antiguo valor binario desplazado a la izquierda por x lugares, por ejemplo, uno, dos, tres o más lugares). A continuación, el procesamiento fluye a la etapa S67, para comprobar el valor de umbral reducido con respecto al valor de umbral mínimo. Si el nuevo valor de umbral calculado Th_{n+1} fuera menor que el umbral mínimo Th_{min} , entonces la etapa S68 sirve para establecer el nuevo umbral para el valor mínimo. Si no, entonces el procesamiento aborta la rutina de ajuste S6 con el nuevo umbral establecido en el valor calculado en la etapa S66.

Regresando ahora a la consideración de la etapa S65, donde se determinó si el intervalo inter-retardo I_C era relativamente grande. Si no, entonces el procesamiento pasa a la etapa S69. Allí, el valor actual I_C para el inter-retardo entre las peticiones de operación de la unidad divisora 21, se resta del antiguo umbral Th_n , la diferencia se desplaza a la derecha por un valor constante, por ejemplo, 3 lugares ($x = 3$), esencialmente como si dividiendo por el correspondiente poder de 2. Este resultado se resta del valor umbral Th_n antiguo, para producir un nuevo umbral Th_{n+1} . Como se discutió anteriormente, un ejemplo de esta fórmula para ajustar el umbral sería la siguiente:

$$Th_n - \frac{Th_n - I_C}{2^x} = Th_{n+1}.$$

(1)

Puesto que un inter-retardo I_C muy grande podría provocar grandes cambios en el umbral, se podría saturar la diferencia (Δ) entre el umbral y el inter-retardo a un valor máximo (por ejemplo, el umbral anterior Th_n) para que ninguna sola lectura pueda afectar indebidamente al umbral. Así, por ejemplo, se podría mantener el nuevo umbral siempre dentro del rango de:

$$Th_n - \frac{Th_n}{2^x} \leq Th_{n+1} \leq Th_n + \frac{Th_n}{2^x},$$

como se representa por las etapas limitantes S70 y S71.

El flujo de procesamiento de la FIG. 3 y la rutina de ajuste de la FIG. 4 se dan a modo de ejemplo, solamente. Los expertos en la técnica reconocerán que otras rutinas y / u otros algoritmos pueden ser utilizados para implementar las presentes enseñanzas para la conservación de energía, evitando atascamiento indebido.

Mientras que lo anterior ha descrito lo que se considera ser el mejor modo y / u otros ejemplos, se entiende que varias modificaciones pueden hacerse en ella y que el objeto descrito en este documento puede implementarse en diversas formas y ejemplos, y que las enseñanzas pueden implementarse en numerosas aplicaciones, sólo algunas de las cuales se han descrito en el presente documento. Se pretende mediante las siguientes reivindicaciones reivindicar cualquiera y todas las aplicaciones, modificaciones y variaciones que caen dentro del verdadero alcance de las presentes enseñanzas.

REIVINDICACIONES

1. Un procedimiento para reducir el consumo de energía de un procesador programable que comprende:

monitorizar un tiempo (S2) desde una última petición para una operación (I_C) de una de una pluralidad de unidades funcionales (21) del procesador programable;
 monitorizar un tiempo transcurrido desde un último apagado de la una unidad funcional (21);
 cuando la una unidad funcional (21) está encendida y el tiempo desde la última petición (I_C) supera un umbral ($S3, Th_n$) que tiene un valor igual a un valor actual, apagar (S8) la una unidad funcional (21) con el fin de reducir la energía consumida por la una unidad funcional (21);
 después de apagar la una unidad funcional (21), reactivar (S12) la una unidad funcional (21) en respuesta a una petición posterior (S10) para la operación de la una unidad funcional (21);
 si al momento de la reactivación de la una unidad funcional (21) el tiempo (I_P) transcurrido desde el cierre es menor que un valor establecido (S63), aumentar el umbral (Th_{n+1}) para tener un valor mayor que el valor actual (S64); pero
 si en el momento de la reactivación de la una unidad funcional (21) el tiempo (I_P) transcurrido desde el cierre no es menor que el valor establecido (S63), reducir el umbral (S69, S66, Th_{n+1}) para tener un valor menor que el valor actual; y
 cuando la una unidad funcional está encendida y el tiempo desde la última petición (I_C) no supera el umbral ($S3, Th_n$), a continuación, en respuesta a la petición posterior para la operación de la una unidad funcional (21) reducir el umbral (S69, S66, Th_{n+1}) para tener el valor menor que el valor actual.

2. El procedimiento de la reivindicación 1, en el que el valor establecido es igual al valor actual del umbral (Th_n).

3. El procedimiento de la reivindicación 1, en el que el valor mayor que el valor actual es dos veces el valor actual.

4. El procedimiento de la reivindicación 1, en el que la reducción del umbral (S69, S66, Th_{n+1}) para tener el valor menor que el valor actual comprende además:

si el tiempo desde la última petición (I_C) supera un segundo valor establecido (S65),
 reducir el umbral (Th_{n+1}) para tener un valor igual al mayor de un valor mínimo (S68, S67) o una primera cantidad restada del valor actual (S66); pero si el tiempo transcurrido desde la última petición (I_C) no supera el segundo valor establecido (S65), reducir el umbral (S69, Th_{n+1}) para tener un valor igual a una segunda cantidad proporcional a la diferencia entre el valor actual y el tiempo desde la última petición (I_C) restado del valor actual, pero no menor que la primera cantidad restada del valor actual (S70) y no mayor que la primera cantidad añadida al valor actual (S71).

5. El procedimiento de la reivindicación 1, en el que la unidad funcional (21) comprende un elemento de lógica para la ejecución de una función aritmética.

6. El procedimiento de la reivindicación 5, en el que la función aritmética comprende una función de multiplicación o de división.

7. Control de energía (23) para un procesador programable que comprende una pluralidad de unidades funcionales (21), el control de energía (23) comprendiendo:

circuitería acoplada a una de las unidades funcionales (21), para activar o apagar selectivamente la una unidad funcional (21);
 medios (31) para monitorizar un tiempo (S2) desde una última petición para una operación (I_C) de la una unidad funcional (21) y un tiempo (S61) transcurrido desde un último apagado (I_P) de la una unidad funcional (21);
 medios (41, 43, 45) para controlar la circuitería para que cuando la una unidad funcional (21) esté encendida y el tiempo desde la última petición (I_C) supera un umbral ($S3, Th_n$) que tiene un valor igual a un valor actual, apagar (S8) la una unidad funcional (21) con el fin de reducir la energía consumida por la una unidad funcional (21);
 después de apagar la una unidad funcional (21), los medios para controlar la circuitería reactivando (S12) la una unidad funcional (21) en respuesta a una petición posterior (S10) para la operación de la una unidad funcional (21);
 si al momento de la reactivación de la una unidad funcional (21) el tiempo (I_P) transcurrido desde el apagado es menor que un valor establecido (S63), los medios para controlar la circuitería aumentando el umbral (Th_{n+1}) para tener un valor mayor que el valor actual (S64); pero si en el momento de la reactivación de la una unidad funcional (21) el tiempo (I_P) transcurrido desde el cierre no es menor que el valor establecido (S63), los medios para controlar la circuitería reduciendo el umbral (S69, S66, Th_{n+1}) para tener un valor menor que el valor actual; y
 cuando la una unidad funcional está encendida y el tiempo (I_C) desde la última petición no supera el umbral ($S3, Th_n$), a continuación, en respuesta a la petición posterior para la operación de la una unidad funcional (21),

los medios para controlar la circuitería reduciendo el umbral (S_{69} , S_{66} , Th_{n+1}) para tener el valor menor que el valor actual.

- 5 **8.** El control de energía (23) según la reivindicación 7, en el que el valor establecido es igual al valor actual del umbral.
- 9.** El control de energía (23) según la reivindicación 8, en el que el aumento del umbral comprende el establecimiento de un nuevo valor de umbral (Th_{n+1}) igual al doble del valor actual del umbral (Th_n).
- 10 **10.** El control de energía (23) de la reivindicación 7, en el que las unidades funcionales (21) forman una pluralidad de etapas de procesamiento de una cadena de procesamiento, las etapas incluyen una etapa de ejecución, y la una unidad funcional (21) comprende un elemento de la etapa de ejecución.

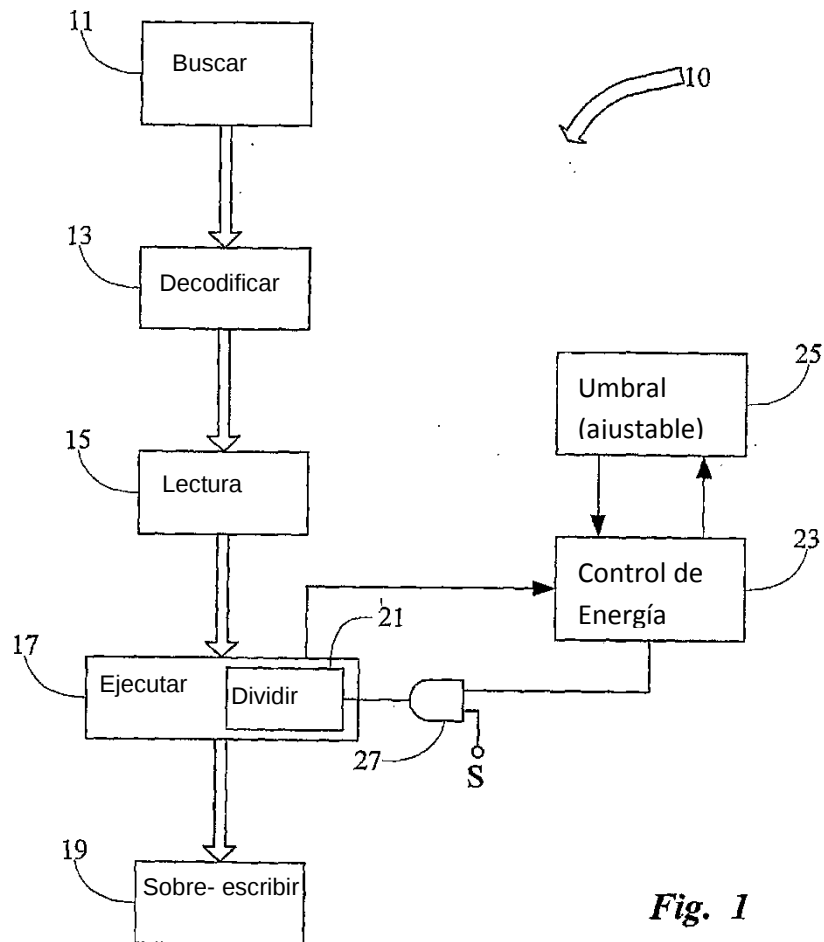


Fig. 1

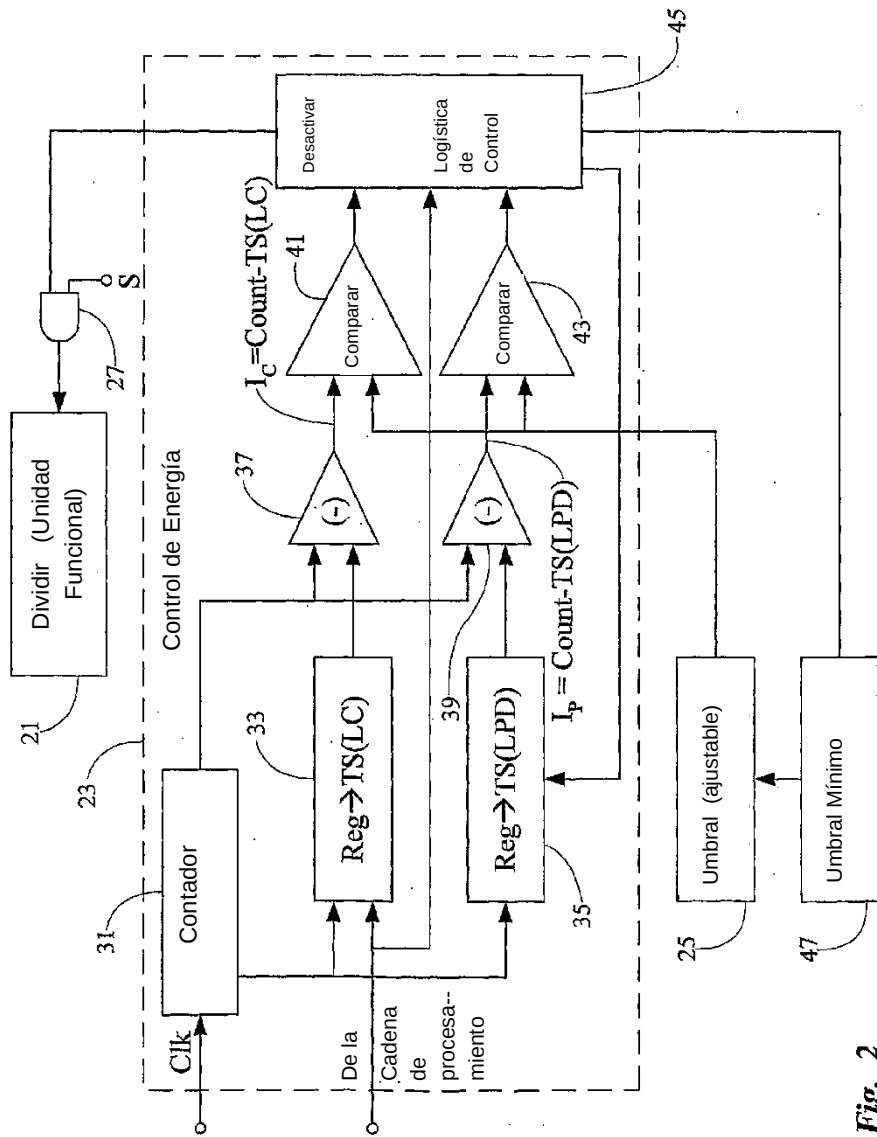
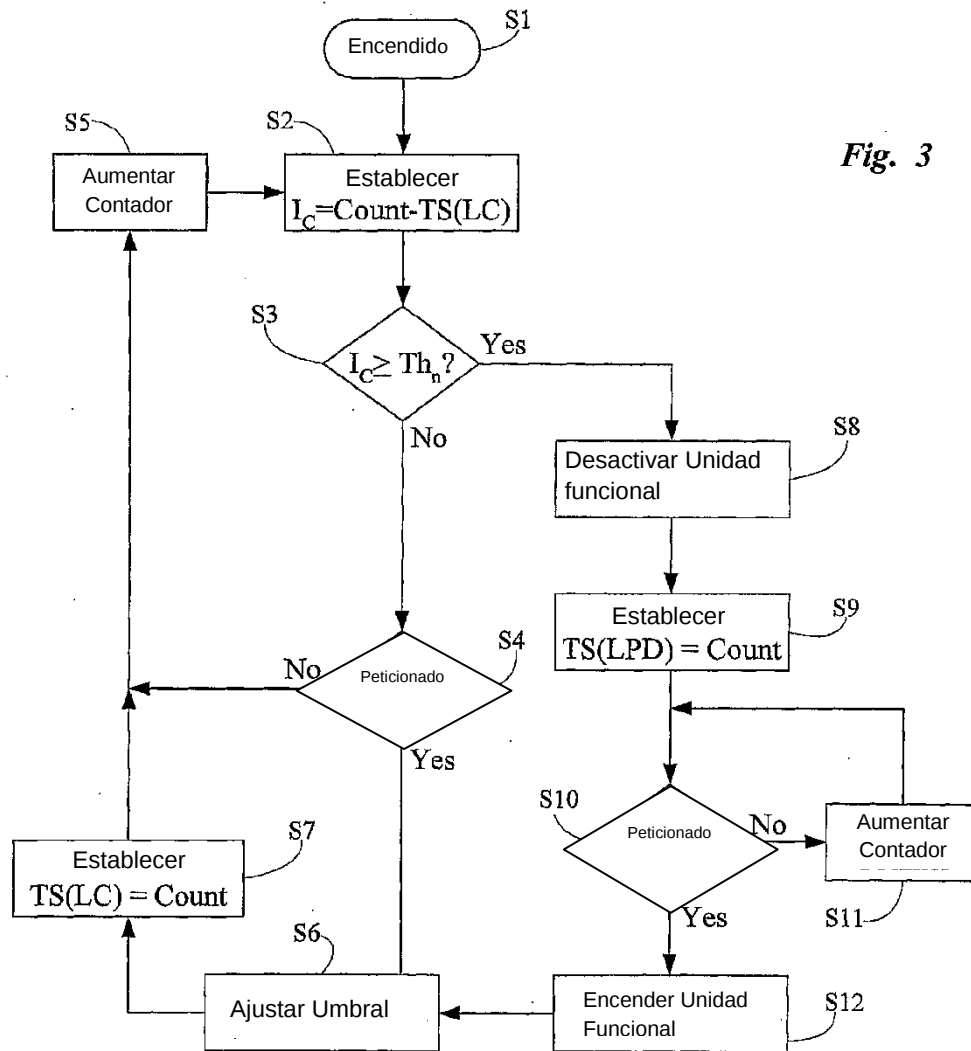


Fig. 2



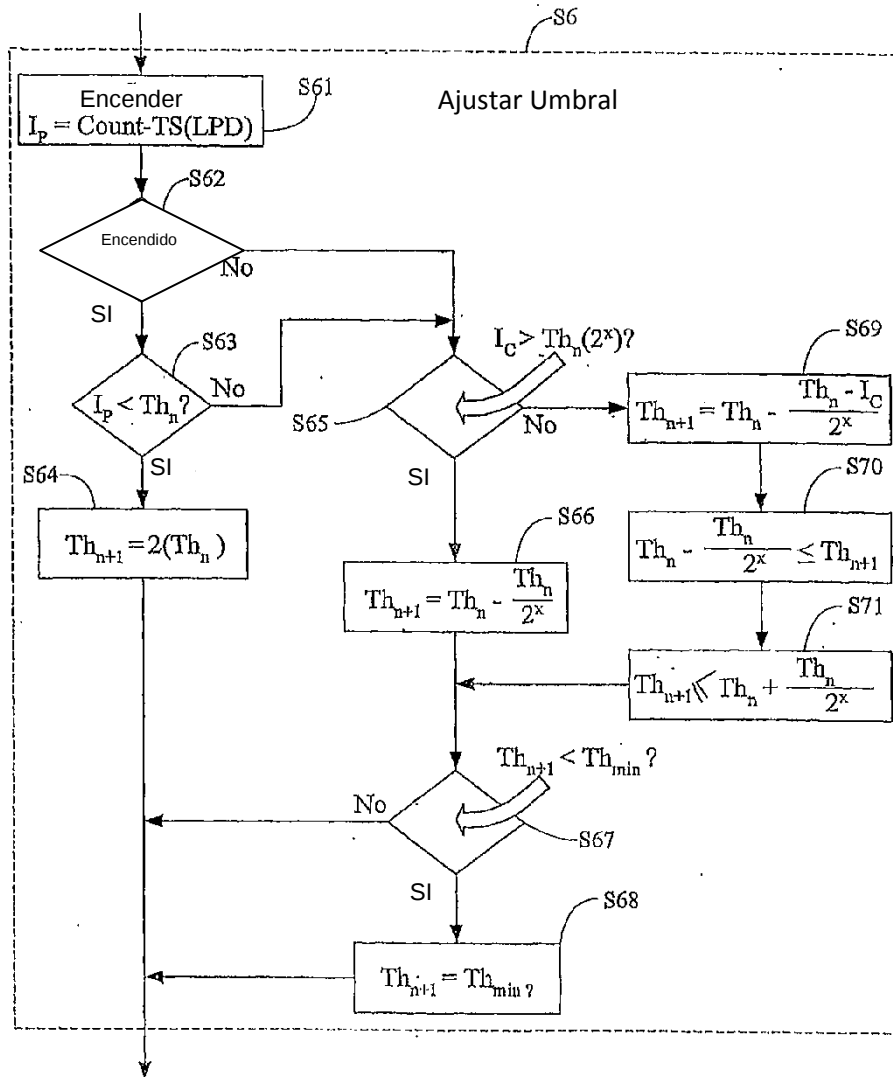


Fig. 4