



19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

11 Número de publicación: **2 360 816**

51 Int. Cl.:
G06F 9/38 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Número de solicitud europea: **06771403 .0**

96 Fecha de presentación : **25.05.2006**

97 Número de publicación de la solicitud: **1894091**

97 Fecha de publicación de la solicitud: **05.03.2008**

54 Título: **Microprocesador con selección automática de paralelismo SIMD.**

30 Prioridad: **09.06.2005 US 150729**

45 Fecha de publicación de la mención BOPI:
09.06.2011

45 Fecha de la publicación del folleto de la patente:
09.06.2011

73 Titular/es: **QUALCOMM Incorporated**
5775 Morehouse Drive
San Diego, California 92121, US

72 Inventor/es: **Dockser, Kenneth, Alan**

74 Agente: **Carpintero López, Mario**

ES 2 360 816 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

Microprocesador con selección automática de paralelismo SIMD.

Campo de la técnica

5 El presente objeto se refiere a técnicas y equipos para controlar automáticamente el estatus activo (activo e inactivo) de uno, o diversos, elementos procesadores de datos en paralelo, p. ej., de un procesador de Instrucción Única y Datos Múltiples (SIMD) o similar, para igualar el grado de paralelismo operativo con una condición en relación con las operaciones de procesamiento del procesador.

Antecedentes

10 Muchos dispositivos utilizan procesadores integrados, tales como microprocesadores y procesadores de señales digitales, con disposiciones complejas de lógica para llevar a cabo funciones de procesamiento de datos de acuerdo con instrucciones de programa. Las aplicaciones que precisan procesamiento digital de datos multimedia, tales como video, audio o gráficos, se están volviendo cada vez más populares entre los consumidores. Sin embargo, el procesamiento de tal información es intensivo y ha resultado en arquitecturas de procesadores que son particularmente adecuadas para tales datos.

15 Los datos multimedia típicamente incluyen una cantidad considerable de datos "en paralelo". Los datos son "en paralelo" cuando las unidades individuales de datos no son dependientes entre sí. Por lo tanto, el procesamiento de una unidad de datos es independiente del procesamiento de otra unidad, esto es que no es necesario esperar a que el procesamiento de otra unidad sea completado. Como resultado, es posible efectuar al tiempo diversas de las mencionadas operaciones independientes de datos en paralelo. Esta característica de ciertos tipos de datos, particularmente las formas comunes de datos multimedia, ha llevado a la creación de procesadores en paralelo, que pueden manipular simultáneamente unidades de datos en paralelo. El procesamiento en paralelo de datos multimedia, por ejemplo, a menudo ayuda a aumentar sustancialmente la velocidad total de procesamiento.

25 Se han desarrollado diversas arquitecturas y tipos de instrucciones para el procesamiento de datos en paralelo, particularmente para aplicaciones multimedia. Por ejemplo, los procesadores de Instrucción Única y Datos Múltiples (SIMD) procesan datos en paralelo. El procesamiento multimedia usando instrucciones SIMD reduce el número total de instrucciones requeridas para ejecutar una tarea de programa particular y acelera la ejecución al operar sobre múltiples elementos de datos en paralelo. Aunque el procesador puede ejecutar una única corriente de instrucciones, la ejecución SIMD de tales instrucciones procesa concurrentemente múltiples corrientes de datos en paralelo.

30 Muchas aplicaciones de procesadores, incluyendo los dispositivos del tipo de elevado procesamiento de datos en paralelo como los procesadores SIMD, producen severas restricciones en la potencia y la energía que los circuitos del procesador pueden consumir. Por ejemplo, los dispositivos portátiles, como los teléfonos móviles, PDAs (asistentes digitales portátiles) y consolas de videojuegos de mano, utilizan baterías como suministro de energía. Sin embargo, estos dispositivos incluyen microprocesadores sofisticados y en algunos casos usan coprocesadores para el procesamiento relacionado con multimedia. Los diseños de procesadores para tales aplicaciones aseguran un cuidadoso control del consumo de potencia y energía, típicamente, para alargar la vida de carga en la batería de suministro de energía así como la vida del chip incluido.

40 La arquitectura de un procesador establece un "ancho" de la ruta de datos a través del procesador, esto es el tamaño máximo de los datos que pueden ser procesados. Los diseños de procesamiento en paralelo, tales como las arquitecturas de procesadores SIMD, son típicamente escaladas para proporcionar un ancho de ruta de datos que se corresponda con la cantidad máxima de datos paralelos que el dispositivo puede procesar durante un ciclo dado. Los procesadores SIMD actuales disponibles pueden procesar hasta 128 bits de datos a un mismo tiempo, lo que significa que el ancho total de la ruta de datos es 128 bits. Sin embargo, en cualquier momento dado, las porciones paralelas del procesador pueden estar procesando unidades más pequeñas de datos.

45 Aunque se conocen otros tamaños, hoy en día los procesadores en paralelo comunes ofrecen una ruta de datos de 64 bits o un ancho de ruta de datos de 128 bits. La ruta de datos está construida por elementos de procesamiento en paralelo, aunque las rutas pueden estar configuradas para manejar datos de diferentes anchos. Por ejemplo, una ruta de datos de 128 bits puede ser fragmentada en pequeños tamaños, esto es que el procesador puede procesar secciones de los datos de 128 bits que tengan una longitud de 8 bits, 16 bits, 32 bits o 64 bits, según lo especificado por las instrucciones de SIMD escritas para la aplicación particular. Usando instrucciones de 8 bits, por ejemplo, un procesador con una ruta de datos con un ancho de 128 bits puede procesar dieciséis unidades de datos de 8 bits, en paralelo. Por el contrario, en la ruta de datos de 64 bits, si una instrucción requiere 128 bits, entonces los datos pueden ser divididos en dos secciones de 64 bits, y la instrucción es ejecutada secuencialmente en ambas secciones de 64 bits. Por supuesto, el procesamiento de cada sección de 64 bits puede entrañar

procesamiento en paralelo, p. ej., de unidades de datos de 8 bits. Dividiendo el procesamiento de la instrucción de 128 bits, la ruta de datos con un ancho de 64 bits puede manejar la instrucción de 128 bits, aunque el tiempo para el procesamiento dividido es mayor.

5 Estas operaciones permiten una utilización óptima de los recursos en paralelo del procesador. Aún así, se presentan ocasiones en las que no son necesarios todos los recursos disponibles para aplicaciones de procesamiento particulares u ocasiones en las que puede ser deseable un funcionamiento completo. Por ejemplo, muchas funciones o aplicaciones del procesador simplemente no requieren la capacidad de procesamiento completa del dispositivo procesador o requieren la capacidad de procesamiento completa sólo por un periodo de tiempo muy limitado. En el ejemplo de procesador con ruta de datos de 128 bits, una aplicación o una porción de la misma puede requerir sólo un procesamiento de datos de 64 bits durante un/os periodo/s de tiempo, por ejemplo, 10 porque existe una cantidad de paralelismo de datos limitada, la cantidad de datos a procesar es baja, o no existe una gran necesidad de velocidad. Sin embargo, si los elementos que proporcionan la ruta de datos de 128 bits son energizados por completo de manera continua, los elementos en paralelo no usados están consumiendo energía de manera innecesaria.

15 A modo de otro ejemplo, las operaciones de procesamiento completamente en paralelo implican una actividad intensa de todos los elementos en paralelo. Por lo tanto, todos los elementos en paralelo generan calor. Algunos diseños de procesador, por ejemplo los que están encapsulados en un paquete plástico para aplicaciones de bajo coste, pueden no ser capaces de soportar el calor por encima de determinadas temperaturas. Las operaciones a altas velocidades de manera continua de todo el conjunto de elementos en paralelo durante un periodo de tiempo 20 extendido pueden generar demasiado calor.

Podría diseñarse un procesador en paralelo con un nivel de paralelismo más bajo de lo requerido para algunas aplicaciones, con el fin de que sea más eficiente para aplicaciones que no requieran un nivel elevado de paralelismo. Aunque estos compromisos pueden reducir el consumo de potencia y, por lo tanto, la generación de calor, resultan en energía gastada y un rendimiento bajo cuando se requiere más paralelismo.

25 Por lo tanto, existe la necesidad de una técnica para ajustar automáticamente el paralelismo de tal procesador en base a la tarea de procesamiento y/o a las condiciones ambientales.

El documento US2003/0088799 A1 da a conocer un sistema para la regulación de la temperatura de los componentes eléctricos y la tasa de consumo de potencia a través de la reconfiguración entre diferentes anchos de bus de interconexión.

30 El documento US2004/254965 A1 da a conocer una unidad computacional que comprende un procesador que tiene una pluralidad de elementos de procesamiento, cada uno de los mismos con una unidad de lógica aritmética, y un controlador para controlar los elementos de procesador. El procesador puede proporcionar un bit respectivo de una palabra de bits múltiples a cada uno de los elementos de procesador y permite transmitir señales entre las unidades lógicas aritméticas para permitir a las unidades ejecutar una operación en paralelo sobre los bits de la palabra de bits múltiples. Se proporcionan circuitos de extensión para acoplar selectivamente una o más unidades 35 computacionales entre sí para combinar sus capacidades de procesamiento en paralelo.

Sumario

Las enseñanzas del presente documento, con referencia a las reivindicaciones adjuntas, proporcionan la 40 activación/desactivación automática de uno o más elementos de un procesador de datos programable en paralelo, en base a una condición de procesamiento detectada. Esencialmente, el control ajusta el grado de paralelismo operativo a los requerimientos de una tarea a ejecutar por parte del procesador de datos en paralelo y/o a la condición ambiental del procesador. Por ejemplo, puede apagarse un elemento de procesamiento en paralelo cuando no es necesario, para conservar energía, o cuando la temperatura de un procesador es demasiado elevada, para permitir que el procesador se enfríe. Los aspectos de estas enseñanzas abarcan diversos procedimientos de 45 operación así como dispositivos de procesamiento en paralelo.

Por ejemplo, un procedimiento para controlar el paralelismo de las operaciones de un procesador de datos en paralelo implica monitorizar una o más condiciones relacionadas con el procesamiento ejecutado a través del procesador de datos en paralelo. Cuando la condición, o condiciones, monitorizadas relacionadas con el procesamiento ejecutado a través del procesador de datos en paralelo se correlacionan con un primer estado del 50 procesamiento, una o más instrucciones son ejecutadas en paralelo en dos elementos de procesamiento en paralelo del procesador de datos, proporcionando una ruta de datos con un primer ancho. Cuando la condición, o condiciones, monitorizadas se correlacionan con un segundo estado de procesamiento, se ejecutan una o más instrucciones en un primero de los dos elementos de procesamiento en paralelo. Bajo estas circunstancias, se procesan los datos de un segundo y menor ancho a través del primer elemento, mientras que un segundo de los 55 elementos de procesamiento en paralelo está inactivo. En una implementación típica, desactivar el segundo

elemento conserva energía y/o reduce la generación de calor (es decir, el consumo de energía) por parte del procesador.

Se dan a conocer ejemplos que controlan automáticamente el paralelismo (activar o desactivar al menos un elemento de procesamiento en paralelo), en base a los requerimientos o la historia de procesamiento detectados.

5 Esta aproximación puede ajustar el grado de paralelismo a los requerimientos de las tareas, y también puede ser indicado por la frecuencia de los requerimientos para procesar los datos del ancho mayor en un nuevo conjunto de instrucciones, o en unas instrucciones procesadas recientemente. Otros ejemplos controlan el paralelismo automáticamente, en base a una condición ambiental detectada, tal como la temperatura del dispositivo.

10 En un ejemplo de coprocesador en paralelo de tipo de Instrucción Única y Datos Múltiples (SIMD) de 128 bits, los elementos de procesamiento en paralelo pueden ser dos unidades lógicas aritméticas (ALUs) SIMD de 64 bits. Cuando ambas unidades están operativas, las ALUs proporcionan una ruta de datos con un ancho de 128 bits, y el coprocesador opera en un modo de procesamiento de datos de 128 bits. La lógica de control monitoriza una condición de procesamiento y detecta un estado de la misma, garantizando un cambio a una operación de 64 bits. En respuesta, una de las ALUs es apagada automáticamente y la otra ALU ejecuta subsiguientemente instrucciones para un procesamiento de datos con un ancho de 64 bits. Sin embargo, incluso en el modo de 64 bits el procesador puede manejar instrucciones para procesar los datos de 128 bits. Por ejemplo, la metodología puede implicar adicionalmente recibir una instrucción SIMD solicitando procesar los datos de 128 bits y expandir esa instrucción SIMD en dos instrucciones que solicitan procesar los datos del ancho de datos de 64 bits. Luego el procedimiento implica ejecutar en secuencia las instrucciones resultantes de la expansión a través de la ALU operativa.

20 Por lo tanto, otro procedimiento para controlar el paralelismo de las operaciones en un procesador de datos en paralelo puede implicar ejecutar una o más instrucciones en paralelo en dos elementos de procesamiento en paralelo del procesador de datos, de manera que se procesan los datos de un primer ancho, detectando una condición relacionada con procesamiento a través del procesador de datos en paralelo, y desactivando el segundo elemento de procesamiento en paralelo al detectar un estado de la condición detectada. En este procedimiento, mientras el segundo elemento de procesamiento en paralelo está desactivado, una instrucción que solicita el procesamiento de datos en paralelo de los datos del ancho mayor es expandida a una pluralidad de instrucciones. Por ejemplo, dos de tales instrucciones solicitan el procesamiento de datos en paralelo de los datos del segundo ancho menor. El primer elemento de procesamiento en paralelo ejecuta las dos instrucciones secuencialmente, mientras que el segundo elemento de procesamiento en paralelo es desactivado.

30 Como puede observarse, las presentes enseñanzas también abarcan procesadores de datos en paralelo adaptados para controlar el grado de paralelismo en respuesta a una o más condiciones monitorizadas. Un ejemplo de tal dispositivo comprende un primer elemento de procesamiento que responde a un programa de instrucciones, para procesar datos de un primer ancho, p. ej. 64 bits en una implementación. El procesador de datos también incluye un segundo elemento de procesamiento que responde a programas de instrucciones, conectado para operar en paralelo con la primera unidad de procesamiento. La operación en paralelo de los dos elementos de procesamiento proporciona un procesamiento en paralelo de los datos de un segundo ancho mayor (p. ej. 128 bits). El procesador también incluye una lógica de control para monitorizar una condición relacionada con operaciones de procesamiento del procesador de datos. Tal como se ha comentado anteriormente, algunos ejemplos de las condiciones monitorizadas incluyen condiciones ambientales tales como la temperatura del procesador, así como condiciones relacionadas con tareas tales como la frecuencia con la que el procesamiento se relaciona con el segundo ancho (p. ej. 128 bits de datos). La lógica activa y desactiva selectivamente el segundo elemento de procesamiento, en base a la relación de la condición monitorizada con un umbral.

45 La lógica puede ser adaptada para controlar la activación y la desactivación del segundo elemento de procesamiento para mitigar la potencial hiperpaginación. Se presentan ejemplos en los que se usan diferentes umbrales (en lo que se refiere a la temperatura y/o a la frecuencia de las instrucciones de 128 bits), para activar y desactivar la segunda ALU, con el fin de proporcionar histéresis. En otro ejemplo, uno o más umbrales pueden ser ajustados dinámicamente, p. ej. en respuesta a una medición de temporización que puede indicar problemas potenciales de hiperpaginación. Por ejemplo, si el umbral de la condición relacionada con la tarea es demasiado sensible, y la segunda ALU está siendo reiniciada demasiado pronto después del apagado más reciente, la lógica puede aumentar el valor del umbral relevante.

55 En la siguiente descripción se expondrán parcialmente algunos objetos y ventajas adicionales, y características novedosas, y se harán en parte aparentes para los expertos en la técnica al examinar lo siguiente y los dibujos adjuntos, o podrán ser aprendidas al producir los ejemplos o poner en funcionamiento los mismos. Los objetivos y ventajas de las presentes enseñanzas pueden ser realizados y obtenidos mediante la práctica o el uso de las metodologías, instrumentalizaciones y combinaciones señaladas en particular en las reivindicaciones adjuntas.

Breve Descripción de los Dibujos

Las figuras de los dibujos representan una o más implementaciones de acuerdo con las presentes enseñanzas, a modo de ejemplo únicamente y no a modo de limitación. En las figuras, los mismos números de referencia se refieren a los mismos elementos, o similares.

- 5 La Fig. 1 es un diagrama de bloques funcional, útil para comprender el control automático de potencia y energía de un elemento de procesamiento en paralelo, por ejemplo en un coprocesador SIMD.

La Fig. 2 es un diagrama de flujo simplificado útil para comprender un ejemplo de las operaciones de control implicadas en una conmutación automática entre dos niveles diferentes de paralelismo en el coprocesador.

Descripción Detallada

- 10 En la siguiente descripción detallada se exponen numerosos detalles específicos a modo de ejemplos para proporcionar una comprensión profunda de las enseñanzas relevantes. Sin embargo, será aparente para los expertos en la técnica que las presentes enseñanzas pueden ser puestas en práctica sin tales detalles. En otros casos, algunos métodos, procedimientos, componentes, y circuitos conocidos han sido descritos a un nivel relativamente elevado, sin detalle, para evitar oscurecer innecesariamente aspectos de las presentes enseñanzas.

- 15 Las diversas técnicas dadas a conocer en el presente documento se refieren al control automático selectivo de potencia y energía de uno o más elementos de procesamiento, en un procesador de datos programable altamente en paralelo. Tal como se observará con mayor detalle más adelante, la lógica del procesador en paralelo detecta cuando las operaciones del programa (p. ej. para una tarea particular o debido a una temperatura detectada) no requieren todo el ancho de la ruta de datos. En respuesta, la lógica de control establece un modo de operación que requiere un subconjunto de la capacidad de procesamiento en paralelo. Puede apagarse al menos un elemento de
20 procesamiento en paralelo, que no sea necesario, para conservar energía y/o reducir el consumo de potencia. Más tarde, cuando la operación de la capacidad añadida sea apropiada, la lógica detecta el cambio en las condiciones de procesamiento y establece automáticamente el modo de operación a una ruta de datos más ancha, típicamente al ancho completo. El cambio de modo reactiva el elemento de procesamiento previamente apagado.

- 25 Las presentes enseñanzas son aplicables a arquitecturas de procesador que tengan elementos de procesamiento de datos en paralelo. Se presentan ejemplos con respecto a la arquitectura de procesadores en paralelo de tipo SIMD. A continuación se hace referencia en detalle a los ejemplos ilustrados en los dibujos adjuntos y comentados más adelante. Tal como puede observarse, los diseños SIMD disponibles comercialmente en la actualidad proporcionan típicamente rutas de datos con un ancho de 64 bits y 128 bits. Sin embargo, las presentes
30 enseñanzas son aplicables a procesadores en paralelo que tengan rutas de datos más estrechas o más anchas. El diagrama de bloques funcionales de la Fig. 1 ilustra componentes de un dispositivo SIMD que son útiles para explicar el control automático de potencia y energía. Los expertos en la técnica reconocerán que un procesador SIMD real incluiría más elementos. En el ejemplo, el dispositivo de procesamiento en paralelo está implementado como un coprocesador SIMD 11, p. ej. para ejecutar operaciones de procesamiento de datos multimedia bajo el control de un núcleo 13 del procesador principal asociado.

- Aunque no se muestra, puede proporcionarse un control para apagar completamente el coprocesador 11 cuando no está siendo usado por la aplicación que se está ejecutando en el núcleo 13 del procesador principal, p. ej. cuando no existe la necesidad de un proceso multimedia. El presente análisis se centrará sin embargo en los casos en los que se precisa cierto procesamiento a través del dispositivo SIMD 11, aunque los diferentes modos operativos utilizarán diferentes niveles de la capacidad de procesamiento en paralelo del dispositivo 11.
40

- En funcionamiento, el núcleo 13 del procesador principal ejecutará uno o más programas, que precisarán de procesamiento multimedia para al menos algunos datos. El conjunto de instrucciones ejecutado incluirá una extensión SIMD, es decir, un número de instrucciones serán instrucciones de procesamiento de tipo SIMD. El núcleo 13 del procesador principal envía estas instrucciones SIMD al coprocesador 11, en donde son inicialmente
45 puestas en una cola de envío (IQ) 15. La cola de envío es esencialmente un dispositivo de memoria tampón de registro en serie para almacenar una cantidad de instrucciones SIMD secuencialmente, antes de la ejecución.

- La etapa 15 de IQ suministra instrucciones, en secuencia, a una etapa 17 de expansión de instrucciones. El coprocesador SIMD 11 puede proporcionar procesamiento en paralelo en diversos modos diferentes de ancho de datos. Aunque puede haber más modos o variaciones en los anchos de datos soportados en cada modo, el ejemplo muestra una configuración del coprocesador 11 que soporta una operación de 64 bits y una operación de 128 bits.
50

Un bit de bandera (M) 19 indica el modo de operación actual. El valor del bit puede mantenerse en un biestable o como un bit de un registro mayor, por ejemplo, un registro de condición. Un ejemplo típico del procesador 11 incluirá un registro de control de 32 bits (no representado por separado), y la bandera 19 de modo puede ser un bit en una posición designada en ese registro. Tal como se comenta más adelante en mayor profundidad, la bandera

19 es establecida automáticamente en respuesta a una condición detectada del procesador, p. ej. el nivel requerido de paralelismo para una tarea de procesamiento particular o una temperatura detectada del dispositivo. La etapa 17 de expansión es sensible al estado de la bandera 19 de modo.

5 El coprocesador SIMD también incluye un archivo 21 de registros, para guardar los datos operantes antes del procesamiento, y una etapa de ejecución. En este sencillo ejemplo, la etapa de ejecución consiste en dos unidades lógicas aritméticas (ALU) 23, 25 de tipo SIMD. En el modo de 128 bits, ambas ALUs 23 y 25 están activas, mientras que en el modo de 64 bits, únicamente la primera (#1) ALU 23 está activa.

10 En el modo de 64 bits, el coprocesador 11 aún puede manejar una instrucción SIMD de 128 bits. Con este propósito, la etapa 17 “expande” la instrucción de 128 bits fraccionándola en dos instrucciones, cada una estructurada para el procesamiento de datos de 64 bits. En el modo de 64 bits, la etapa 17 despacha todas las instrucciones a la ALU 23, incluyendo las instrucciones que originalmente solicitaban el procesamiento de datos con un ancho de 64 bits, así como parejas de instrucciones de 64 bits derivadas de la expansión de cualquier instrucción de 128 bits en la corriente de instrucciones SIMD de la etapa 15 de IQ. En el modo de 128 bits, la etapa 17 despacha instrucciones a ambas primera y segunda ALUs 23, 25, de manera que las unidades 23 y 25 operan en paralelo para proporcionar una ruta de procesamiento de datos SIMD con un ancho de 128 bits.

15 Aunque el procesador 11 ejecuta una única corriente de instrucciones, la ejecución SIMD de esas instrucciones procesa simultáneamente fragmentos de datos en paralelo. Cada ALU 23 ó 25, por ejemplo, puede operar simultáneamente con dos palabras de datos de 32 bits o con cuatro palabras de datos de 16 bits. Cuando ambas unidades 23 y 25 están operando en paralelo, la capacidad de procesamiento de datos combinada puede manejar cuatro palabras de datos de 32 bits u ocho palabras de datos de 16 bits. También son posibles otras combinaciones de procesamiento en paralelo.

20 En un sencillo ejemplo, el archivo 21 de registros puede comprender dos registros con un ancho de 128 bits para datos, aunque pueden proporcionarse registros adicionales. Los expertos en la técnica comprenderán que el archivo de registros podría comprender registros adicionales, por ejemplo, dieciséis registros, cada uno con un ancho de 128 bits. El control del archivo de registros divide cada 128 bits de datos y envía una cantidad apropiada a cada una de las ALUs 23 y 25. El puerto D del archivo 21 es un puerto de escritura. A través del puerto D, los datos con un ancho de 128 bits pueden ser escritos en un registro del archivo 21, p. ej. desde una fuente (no representada) o desde los resultados emitidos por las ALUs. El puerto S del archivo 21 es un puerto de lectura. A través del puerto S, los datos que tienen un ancho de 128 bits pueden ser leídos desde un registro del archivo 21 y llevados, p. ej. hasta un sumidero (no representado) tal como una memoria. Los puertos A y B del archivo 21 de registros son puertos de lectura, para suministrar datos divididos (de 64 bits cada uno) a las ALUs 23 y 25.

30 Para las instrucciones de procesamiento de datos de 128 bits, el archivo 21 de registros suministra la mitad más baja (la menos significativa) de los datos de 128 bits a la segunda ALU 25, y envía la mitad más elevada (la más significativa) de los datos a la primera ALU 23. Para una instrucción de 64 bits, el procesador puede seleccionar tanto la mitad más baja como la mitad más elevada de los 128 bits de datos en el archivo de registros para suministrarla a la primera ALU 23. En el modo de 128 bits, cualquier instrucción de 64 bits llega a la primera (#1) ALU 23, y las instrucciones de 128 bits llegan a ambas ALUs SIMD 23 y 25. En el modo de 64 bits, todas las instrucciones de 64 bits llegan a la primera ALU SIMD 23. Cualquier instrucción de 128 bits es convertida en dos instrucciones de 64 bits, que son suministradas secuencialmente a la primera ALU SIMD 23.

40 Las técnicas dadas a conocer en el presente documento controlan automáticamente el paralelismo de las operaciones de un procesador de datos en paralelo, en base a una o más condiciones detectadas en relación al procesamiento ejecutado a través del procesador de datos en paralelo. En el ejemplo, el control automático activa y desactiva selectivamente la segunda (#2) ALU 25.

45 El estatus operativo del elemento de procesamiento en paralelo controlado, es decir la segunda (#2) ALU 25 en el ejemplo de la Fig. 1, puede ser controlado por medio de una o más puertas o circuitos de conmutación, representados genéricamente por las puertas lógicas 27 y 29. Tal puerta o conmutador suministra y retira selectivamente una señal de capacitación precisada por el elemento particular. Las puertas 27 y 29 aparecen como puertas AND en el dibujo, pero pretenden ser representaciones genéricas de circuitos para acoplar selectivamente señales a los elementos de la ALU 25. Tal circuito 27 ó 29 puede ser una puerta lógica de cualquier tipo, un conmutador, una combinación de los mismos, u otro circuito cualquiera configurado para suministrar el tipo de señal apropiado a los elementos funcionales de la ALU 25 en respuesta a la selección de señales apropiada desde el control 31 de modo.

55 De esta manera, el control 31 de modo controla el estatus activo de la segunda (#2) ALU 25, en el ejemplo de la Fig. 1. La propia bandera de modo podría proporcionar un control directo de la activación y la desactivación selectivas de la/s puerta/s 27, 29 y por lo tanto de la ALU 25. Sin embargo, en tal caso la ALU se encenderá y apagará inmediatamente ante cada transición del modo de operación según indiquen los cambios en el estado de la

bandera 19 de modo. En la mayoría de las implementaciones, las ALUs 23 y 25 comprenderán unas unidades pipeline multietapa, y puede haber un número de instrucciones en vuelo en la ALU 25 cuando la bandera 19 cambia. La lógica de control de potencia y energía es sensible al bandera 19 de modo, pero puede proporcionar un retardo de tiempo tras un cambio de estado, para permitir una transición de procesamiento suave. Por ejemplo, la lógica del control 31 puede monitorizar las operaciones de la ALU 25, de manera que al detectar una transición a 1 bit (un cambio del modo de 128 bits al modo de 64 bits), el control 31 retardará la desactivación de la ALU 25, hasta que cualquier instrucción de vuelo de 128 bits restante haya sido procesada y pasada por dicha ALU. Aunque no se muestra, la lógica del control 31 de modo también puede proporcionar señales a otros elementos del procesador para controlar la aplicación inicial de instrucciones de 128 bits tras una transición al modo de 128 bits, p. ej. para permitir el tiempo preciso para encender la ALU 25.

El control 31 de modo puede ser implementado para controlar selectivamente el estatus de la segunda (#2) ALU 25 en cualquiera de las diversas maneras conocidas, habilitando y deshabilitando selectivamente cualquier señal precisada para la operación de la unidad por medio de la operación de una puerta o circuito de conmutación 27 ó 29 apropiados. Por ejemplo, la ALU 25 puede ser habilitada y deshabilitada selectivamente por medio de la activación periódica controlada de la señal de reloj (CLK) que se usa para controlar las funciones de conmutación del circuito dentro de la unidad 25. En tal implementación, la ALU 25 es deshabilitada cortando el flujo de la señal de reloj (CLK) que atraviesa la puerta 27 hasta la ALU 25, de manera que los circuitos de la unidad 25 no conmuten en absoluto en respuesta a las señales de reloj. Esto reduce el consumo de potencia dinámica. Para reducir las fugas en tal implementación, los transistores de la ALU 25 pueden estar diseñados con unos voltajes de umbral de puerta relativamente elevados. Sin embargo, el control basado únicamente en la activación periódica de la señal de reloj puede permitir unos reinicios de la ALU relativamente rápidos.

Alternativamente, o adicionalmente (tal como se muestra), el estado de la ALU puede estar controlado por la aplicación o la retirada selectiva de una conexión a un terminal del suministro de potencia. Aunque la conexión efectuada podría ser a tierra o a un voltaje negativo de suministro, para el propósito del ejemplo, la puerta 29 controla el suministro de voltaje V a la segunda (#2) ALU 25 de SIMD. La operación de la puerta 29 por medio del control 31 de modo activa y desactiva la ALU 25 de acuerdo con el modo de operación actual del coprocesador 11. En tal implementación, cuando la bandera de modo cambia a 1 (funcionamiento de 64 bits), el control 31 de modo deshabilita la ALU 25 una vez que cualquier instrucción de 128 bits restante haya sido procesada. En ese momento, el control 31 dispara la puerta 29 para cortar la conexión al terminal de potencia del voltaje V de suministro, con respecto a los circuitos de la ALU 25. Este tipo de corte elimina el consumo de potencia dinámica y las fugas a través de los circuitos de dicha unidad.

La primera (#1) ALU 23 de SIMD está activa en ambos modos. Por lo tanto, tal ALU se muestra conectada directamente a la señal de reloj (CLK) y al voltaje (V) de suministro, sin un control de puerta. Sin embargo, las señales de potencia y/o de reloj a ese elemento de procesamiento 23 también pueden ser controladas, p. ej. para permitir el apagado del mismo cuando el coprocesador 11 no sea necesario.

El ejemplo de la Fig. 1 muestra un elemento de procesamiento en paralelo individual, ALU 25, controlado en base al modo de operación activo. Los expertos en la técnica reconocerán que un procesador dado puede incluir varios controles similares para una cantidad de elementos en paralelo que pueden estar inactivos mientras el procesador 11 opera sobre datos más estrechos durante periodos considerables de tiempo. Asumiendo un ancho de banda máximo de 128 bits, en otro ejemplo, la primera ALU puede ser implementada como dos ALUs de 32 bits. En ese caso, un sistema de control adicional similar al 27, 29 y 31 podría controlar la segunda ALU de 32 bits para proporcionar una desactivación selectiva adicional de una ALU SIMD de 32 bits, dejando únicamente una unidad activa para el funcionamiento único de 32 bits.

La operación del control 31 de modo controla el paralelismo del coprocesador 11 en respuesta a un estado de al menos una condición detectada. Cuando la condición monitorizada está en un primer estado, se ejecutan instrucciones SIMD en paralelo en dos elementos de procesamiento en paralelo 23 y 25 del procesador de datos, proporcionando una ruta de datos con un ancho de 128 bits. Cuando la condición monitorizada está en un segundo estado, se ejecutan las instrucciones en el primer (#1) elemento de procesamiento en paralelo 23. Bajo esta circunstancia, se procesan datos con un ancho de 64 bits a través del primer elemento, mientras que el segundo elemento de procesamiento en paralelo 25 está inactivo. El control 31 de modo también establece la bandera 19 de modo para controlar el funcionamiento de la etapa 17 de expansión de instrucciones.

El estatus de modo indicado por la bandera 19 controla las operaciones selectivas de las funciones de despacho y expansión de la etapa 17. Por ejemplo, el procesador podría estar configurado para procesar todas las instrucciones en la forma de instrucciones de 64 bits, siempre y cuando el bit de la bandera 19 sea un 1. Si ocasionalmente la corriente de instrucciones SIMD incluye una instrucción de 128 bits, la etapa 17 de expansión de instrucciones la dividirá en dos instrucciones de 64 bits y las enviará una tras otra a la ALU 23. Cuando el bit de la bandera 19 es un 0, la etapa 17 conmuta al modo de 128 bits, en el que dirige las instrucciones para las

operaciones con un ancho de 128 bits a ambas ALUs 23 y 25 sin expandirlas. En este modo, si existe una instrucción ocasional de 64 bits en la corriente del procesador principal 13, la etapa 17 puede despachar la instrucción de 64 bits a la primera ALU 23.

5 El estatus de modo del coprocesador 11 establecido por el control 31 de modo controla la activación y desactivación selectivas de al menos un elemento de procesamiento en paralelo, en el ejemplo, la segunda ALU 25. La ALU 25 está activa (energizada) en el modo de 128 bits, mientras que no resulta necesaria en el modo de 64 bits y está desenergizada (desactivada) en ese modo.

10 En la implementación ilustrada, el control 31 de modo es sensible a una condición relacionada con una tarea de procesamiento, tan frecuentemente como el procesador maneje instrucciones de un tipo particular. En esta implementación, el control 31 de modo también es sensible a una condición ambiental, la temperatura. Por ejemplo, la lógica de la máquina de estado o el dispositivo programable que sirve como control 31 de modo identifica el número de instrucciones de 128 bits entre las instrucciones de la etapa 15 de espera. Si la tasa o frecuencia de instrucciones de 128 bits está en, o por encima de, un nivel predefinido, el control 31 de modo establece la bandera 19 de modo a 0 (modo de 128 bits) y activa la segunda ALU 25. Por el contrario, si la tasa o frecuencia de instrucciones de 128 bits está en, o por debajo de, un nivel predefinido, el control 31 de modo establece la bandera 19 de modo a 1 (modo de 64 bits) e inicia un procedimiento para desactivar la segunda ALU 25. Los expertos en la técnica reconocerán que esta función de control sensible a las tareas podría usar otros parámetros monitorizados, tales como la frecuencia o la tasa de instrucciones de 64 bits, o la frecuencia de cualquier tipo de instrucción de la historia reciente (en base a algún número de instrucciones existentes en el dispositivo 11 o recién procesadas a través del mismo).

20 Para la monitorización ambiental, el procesador ejemplar 11 incluye un sensor de temperatura 33. El sensor 33 proporciona una señal al control 31 de modo. El control 31 de modo activa y desactiva la ALU 25 y establece la bandera de modo 19, en base al nivel de la señal del sensor de temperatura, esencialmente en base a la relación de la temperatura con uno o más valores de umbral. Si la temperatura detectada se vuelve demasiado elevada mientras se opera en el modo de 128 bits, el control 31 puede desactivar la ALU 25 para permitir que el dispositivo 11 se enfríe. Más adelante (p. ej. cuando exista una lectura de temperatura más baja), si la tarea solicita un procesamiento de datos con un ancho de 128 bits, el control 31 puede reactivar la ALU 25 y establecer la bandera 19 de vuelta al modo de 128 bits. Los expertos en la técnica reconocerán que hay otras condiciones ambientales que pueden ser detectadas en lugar de, o adicionalmente a, la temperatura.

30 Además, la ejecución de una instrucción puede invalidar esencialmente la operación automática del control 31 de modo, para permitir al programador establecer el modo a un nivel deseado de paralelismo. El control de modo recibe un comando de invalidación desde una de las ALUs, 25, o desde el núcleo 13 del procesador principal. En respuesta, el control 31 de modo establecerá la bandera 19 de modo, para indicar si la etapa 17 deberá operar o no en el modo de 64 bits, y proporcionará una configuración correspondiente del estatus operativo (APAGADO o ENCENDIDO) de la ALU 25.

40 Si se usa la invalidación, el programa está escrito para establecer el modo de operación en puntos apropiados del flujo de proceso. Las instrucciones de configuración de modo pueden ser grabadas por el programador, o un compilador puede insertar las instrucciones de configuración de modo cuando el programa está compilado en el código de lenguaje de la máquina. El coprocesador 11 puede estar diseñado para establecer un modo como invalidación, en respuesta a las instrucciones de modo ejecutadas por una de las ALUs 23, 25, o ambas, en cuyo caso las instrucciones están escritas de tal manera que el núcleo envíe tales instrucciones al coprocesador 11 para su ejecución. Alternativamente, el coprocesador 11 podría estar acoplado al procesador principal 13, de manera que el núcleo 13 establezca el modo, en respuesta a las instrucciones de modo ejecutadas por el núcleo 13 del procesador principal. También es posible que tanto el procesador 11 como el 13 puedan establecer el modo en base a dicha instrucción de invalidación.

45 También son posibles otras variaciones en la invalidación. Por ejemplo, el comando de invalidación desde el núcleo 13 del procesador principal o las ALUs 23, 25, en respuesta a una instrucción específica, puede invalidar la configuración automática en base a la condición relacionada con la tarea, pero no invalidar el control en base a una condición ambiental crítica, p. ej. de manera que el control 31 de modo pueda reducir el paralelismo en respuesta a una temperatura excesiva incluso si esto es contrario a la configuración proporcionada por la invalidación en base a una instrucción explícita.

50 A partir de lo anteriormente mencionado debería ser aparente que el control de modo puede responder automáticamente a los estados de una variedad de condiciones monitorizadas en relación con el procesador o sus operaciones de procesamiento. Además, pueden diseñarse una variedad de diferentes algoritmos para implementar la función de control de paralelismo deseada. Para apreciar del todo las ventajas de las presentes enseñanzas, puede resultar útil considerar un ejemplo.

La Fig. 2 es un diagrama de flujo que muestra un posible flujo de la ejecución implicada en la conmutación automática entre dos diferentes niveles de paralelismo en el coprocesador 11 de la Fig. 1, en base a la frecuencia (o tasa) de las instrucciones en cola y a la temperatura del procesador detectada. A los efectos de esta discusión, asumamos que inicialmente (en la etapa S1) el procesador está ejecutando instrucciones en paralelo en las dos unidades lógicas aritméticas en paralelo 23 y 25. En este modo, los datos procesados pueden tener un ancho de hasta 128 bits, es decir tan ancho como toda la ruta de datos proporcionada por las unidades 23 y 25 trabajando juntas. En este estado, el control 31 de modo compara el valor T de la señal de temperatura con un valor T_h de umbral de alta temperatura (etapa S2). Si la temperatura es suficientemente baja (el valor T no es mayor que el valor T_h de umbral de alta temperatura), el procesamiento fluye desde S2 a S3.

La etapa S3 representa una determinación de una condición de procesamiento relacionada con una tarea, en este caso el número de instrucciones de 128 bits en cola. Si la espera puede contener hasta ocho instrucciones, por ejemplo, cuando la etapa 15 de IQ está llena, la frecuencia o la tasa f sería el número de instrucciones de 128 bits dividido por 8. Por supuesto, pueden usarse otras medidas de una condición relacionada con una tarea para determinar cuándo una tarea solicita predominantemente un procesamiento de datos del ancho de 128 bits. Por ejemplo, si la longitud de la espera es constante, el número de instrucciones de 128 bits en la espera puede ser usado en lugar de la frecuencia. Pueden desarrollarse medidas históricas en base a análisis similares de cierta cantidad de instrucciones ya en vuelo y/o procesadas a través de las ALUs, o tal medida histórica puede ser combinada con los análisis de las instrucciones en la etapa 15 de IQ.

A los efectos de esta discusión, asumamos que la lógica monitoriza la frecuencia (f) de las instrucciones de 128 bits en la etapa 15 de JQ. En la etapa S3, si el valor f actual está en, o por encima de, un valor f_1 de umbral bajo, el procesamiento fluye de vuelta desde S3 a S1, de manera que el coprocesador 11 continúa en el modo de operación de 128 bits. Asumiendo una longitud de espera máxima de ocho instrucciones, el umbral f_1 de frecuencia baja puede ser establecido alrededor de $1/8$, de manera que se mantenga el coprocesador en el modo de 128 bits en curso siempre y cuando 1 de cada 8 instrucciones sea una instrucción de 128 bits. El umbral para permanecer en el modo de 128 bits sería típicamente bajo. Incluso $1/16$ sería un criterio eficiente para permanecer en el modo de 128 bits. Por supuesto el umbral también podría ser dependiente del tiempo, p. ej. requerir 2 o más instrucciones de 128 bits por cada 8 instrucciones para cada número de ciclos de procesamiento. En el flujo ilustrado, el procesamiento representado por las etapas S1 a S3 continúa, es decir con el coprocesador funcionando en el modo de procesamiento de datos con un ancho de 128 bits, siempre y cuando la temperatura permanezca en, o por debajo de, el umbral T_h y la frecuencia de instrucciones de 128 bits permanezca en, o por encima de, el umbral f_1 .

Si la lógica que forma el control 31 de modo detecta que a la temperatura T se excede el umbral T_h , entonces el procesamiento fluye desde la etapa S2 a la etapa de procesamiento S4. De manera similar, si la lógica que forma el control 31 de modo detecta que la frecuencia f de las instrucciones de 128 bits está en, o por debajo de, el umbral f_1 , el procesamiento fluye desde la etapa S3 a la etapa de procesamiento S4. En ambos casos, en S4, el bloque 17 de expansión de instrucciones comienza la conversión de las instrucciones de 128 bits subsiguientes, de haberlas, en parejas de instrucciones de 64 bits. En la etapa S5, el control 31 de modo comprueba la segunda unidad de lógica aritmética (ALU #2) 25 para determinar si queda alguna instrucción de 128 bits en vuelo que necesite ser procesada y sacada desde esa unidad de lógica aritmética 25. De ser así, la lógica espera (S6) y comprueba nuevamente. Esta monitorización (S5-S6) continúa hasta que no quede ninguna instrucción de 128 bits en vuelo para ser procesada por la segunda unidad de lógica aritmética (ALU #2) 25, en cuyo momento el procesamiento fluye hasta la etapa S7.

En la etapa S7, el control 31 de modo desactiva o apaga el segundo elemento de procesamiento 25 de tipo ALU, para conservar la energía y/o reducir la generación de calor. El procesamiento adicional se ejecuta en el modo de 64 bits (S8), tal como se ha mencionado anteriormente. Por ejemplo, mientras que la segunda unidad de lógica aritmética 25 está inactiva, el procesador 11 ejecuta una o más instrucciones en la primera unidad de lógica aritmética 23, para procesar 64 bits de datos. Cualquier instrucción de 128 bits recibida en este modo es expandida y procesada secuencialmente como dos instrucciones de 64 bits (tal como se comenzó en S4).

Mientras el coprocesador opera en el modo de 64 bits, el control 31 de modo continúa comprobando la temperatura y la frecuencia con la que las instrucciones de 128 bits están siendo recibidas en la etapa 15 de IQ. La temperatura medida es comparada con un umbral en la etapa S9, y la frecuencia detectada de instrucciones de 128 bits es comparada con un umbral en la etapa S10. El mismo umbral que fue usado en S2 podría ser usado en S8; y/o el mismo umbral que fue usado en S3 podría ser usado en S10. Sin embargo, el uso de los mismos umbrales puede llevar a la conmutación entre encendido y apagado de la ALU 25 en respuesta a cada pequeño cambio en uno o ambos de los parámetros monitorizados, lo que resultaría en conmutaciones excesivas, a menudo de duraciones cortas.

Activar o energizar la ALU 25 desde un estado de apagado lleva tiempo y consume energía. En algunos casos, energizar el elemento 25 podría costar más energía que simplemente dejarlo encendido durante un corto periodo de

tiempo. Además, lleva tiempo energizar nuevamente un componente. El retardo o latencia resultante al reinicializar la ALU podría resultar en un atasco del procesador, lo que degrada el rendimiento. Por lo tanto, es deseable asegurar que el elemento de procesamiento en paralelo controlado, en nuestro ejemplo la ALU, no sea apagado repetidamente únicamente para ser encendido poco tiempo después. Desenergizar y energizar repetidamente puede ser denominado “hiperpaginación”. Los ejemplos ilustrados usan diferentes umbrales para las comparaciones en los diferentes modos de operación, con el fin de proporcionar cierta histéresis en las operaciones de conmutación, para reducir los eventos de conmutación y la potencial hiperpaginación. Tal como se comentará con mayor detalle más adelante, la lógica del control 31 de modo puede implementar estrategias anti-hiperpaginación diferentes, o adicionales.

Volviendo al ejemplo específico, en la etapa S9, el control 31 de modo compara la medición T de la temperatura actual con el umbral T_1 , que es ligeramente menor que el umbral T_h . Si la temperatura medida T es aún superior al umbral inferior frío T_1 , el procesamiento continúa en el modo de 64 bits (S8). Típicamente, la temperatura continuará cayendo en este modo. Cuando el coprocesador se ha enfriado lo suficiente, el control 31 de modo determina que la temperatura T medida está en, o por debajo de (ya no es superior a), el umbral inferior T_1 , y el procesamiento fluye desde S9 hasta S10.

Dado que ahora el procesador está lo suficientemente frío como para permitir la vuelta al funcionamiento de 128 bits, acto seguido el control de modo determina (en S10) si la tarea de procesamiento garantiza las operaciones de 128 bits. De esta manera, en S10 en nuestro ejemplo, el control de modo compara la frecuencia o tasa (f) de las instrucciones de tipo de 128 bits en la etapa 15 de IQ con el valor de umbral más alto f_h . Asumiendo nuevamente una longitud de espera con un máximo de ocho instrucciones, el umbral de frecuencia alta puede ser establecido alrededor de $\frac{3}{8}$, de manera que se mantenga el coprocesador en el modo de 64 bits en curso siempre y cuando no hay más de tres instrucciones de 128 bits por cada 8 instrucciones. Por supuesto el umbral también podría ser dependiente del tiempo, p. ej. requerir 4 o más instrucciones de 128 bits por cada 8 instrucciones durante un número de ciclos de procesamiento para disparar la transición a la operación de 128 bits.

Si f es inferior o igual (aún no superior a) al valor f_h de umbral, entonces no hay suficiente demanda como para justificar operaciones de 128 bits, incluso aunque la temperatura del procesador sea segura. Por lo tanto, el procesamiento fluye desde S10 hasta S 8, de manera que el coprocesador 11 continúe en el modo de 64 bits de operación. El procesamiento representado por las etapas S8 a S10 continuará, es decir con el procesador operando en el modo de procesamiento con un ancho de 64 bits, siempre y cuando no se alcance ninguno de los umbrales, es decir hasta que la temperatura sea suficientemente baja y la frecuencia de las instrucciones de 128 bits sea lo suficientemente alta para garantizar la transición al modo de operación de 128 bits.

Cuando la temperatura está en, o por debajo de, el umbral T_1 y la frecuencia de 128 bits excede el umbral f_h , el procesamiento fluye a través de S9 y S10 hasta la etapa S11. En la etapa S11, la lógica del control 31 activa la segunda unidad de lógica aritmética (ALU #2) 25. En este momento (S 12), las instrucciones de 128 bits pasarán hasta las ALUs sin modificación (la recodificación adicional en dos instrucciones de 64 bits ya no es necesaria). En este estado, el procesamiento regresa a la etapa S1 en donde operaciones adicionales utilizan el modo de instrucciones de 128 bits.

Tal como se ha mencionado anteriormente, la lógica del control 31 de modo puede implementar otras estrategias anti-hiperpaginación en vez de, o en combinación con, la histéresis proporcionada por las diferencias en los valores de umbral. Por ejemplo, los valores de umbral puede ser ajustados en respuesta a medidas que pueden indicar hiperpaginación, p. ej. el intervalo de tiempo durante el que la ALU está apagada en respuesta a una baja tasa de ocurrencia de instrucciones de 128 bits.

Las presentes enseñanzas tienen un amplio rango de aplicaciones. Por ejemplo, el control de la potencia y la energía puede ser implementado en otros procesadores en paralelo y en procesadores con unos anchos de ruta de datos diferentes. Además, el ejemplo del procesador en paralelo mencionado anteriormente proporcionaba un coprocesador asociado a un núcleo del procesador principal. Los expertos en la técnica comprenderán, sin embargo, que las técnicas de control de paralelismo descritas en el presente documento no están limitadas a implementaciones con coprocesadores.

Aunque lo anterior ha descrito el que está considerado como el mejor modo y/u otros ejemplos, debe comprenderse que pueden hacerse diversas modificaciones en el mismo y que el asunto descrito en el presente documento puede ser implementado en diversas formas y ejemplos, y que las enseñanzas pueden ser aplicadas en numerosas aplicaciones, de las cuales sólo algunas han sido descritas en el presente documento. Las siguientes reivindicaciones pretenden reivindicar cualquiera y todas las aplicaciones, modificaciones y variaciones que estén dentro del verdadero alcance de las presentes enseñanzas.

REIVINDICACIONES

- 5 1.- Un procedimiento para controlar el paralelismo de operaciones en un procesador de datos en paralelo, que comprende:
- monitorizar una o más condiciones en relación con el procesamiento llevado a cabo por el procesador de datos en paralelo;
- 10 cuando la una o más condiciones monitorizadas se correspondan con un primer estado, ejecutar una o más instrucciones en paralelo en dos elementos de procesamiento en paralelo del procesador de datos en paralelo, proporcionando una ruta de datos con un primer ancho: y
- cuando la una o más condiciones monitorizadas se correspondan con un segundo estado, ejecutar una o más instrucciones en paralelo en el primero de los dos elementos de procesamiento en paralelo, de manera que se procesen los datos de un segundo ancho de banda menor que el primero, mientras que el segundo de los dos elementos de procesamiento en paralelo está inactivo;
- 15 en el cual la monitorización comprende:
- monitorizar la temperatura del procesador de datos en paralelo, en el cual el primer estado está asociado con una temperatura monitorizada en, o por debajo de, un umbral de temperatura y el segundo estado está asociado con una temperatura por encima del umbral; y
- 20 monitorizar una tarea llevada a cabo mediante la ejecución de la una o más instrucciones, en el cual en el primer estado la tarea solicita el procesamiento de datos del primer ancho a una frecuencia superior a un umbral de frecuencia establecido.
- 2.- El procedimiento de la reivindicación 1, en el cual la ejecución de una o más instrucciones del primer elemento de procesamiento en paralelo, mientras el segundo de los dos elementos de procesamiento en paralelo está inactivo, comprende:
- 25 expandir una instrucción que solicita el procesamiento de datos en paralelo de los datos del primer ancho en una pluralidad de instrucciones que comprenden dos instrucciones que solicitan el procesamiento de datos en paralelo de los datos del segundo ancho; y
- ejecutar las dos instrucciones secuencialmente en el primer elemento de procesamiento en paralelo.
- 3.- El procedimiento de la reivindicación 2, en el cual:
- 30 ejecutar la una o más instrucciones de procesamiento en el elemento de procesamiento en paralelo proporciona un procesamiento de datos con un ancho de 64 bits en base a unas instrucciones ejecutadas; y
- ejecutar la una o más instrucciones de procesamiento en el primer y el segundo elementos de procesamiento en paralelo proporciona el procesamiento de datos con un ancho de 128 bits en base a las instrucciones ejecutadas.
- 35 4.- El procedimiento de la reivindicación 1, en el cual cada uno del primer y el segundo elementos de procesamiento en paralelo comprende una unidad de lógica aritmética.
- 5.- El procedimiento de la reivindicación 4, en el cual la una o más instrucciones ejecutadas en el primero y segundo elementos de procesamiento son instrucciones de tipo Instrucción Única y Datos Múltiples (SIMD).
- 6.- El procedimiento de la reivindicación 5, en el cual el primero y segundo elementos de procesamiento son elementos de un coprocesador SIMD.
- 40 7.- El procedimiento de la reivindicación 1, en el cual ejecutar la una o más instrucciones en el elemento de procesamiento en paralelo comprende el procesamiento de datos multimedia.
- 8.- El procedimiento de la reivindicación 1, en el cual
- la potencia al segundo elemento de procesamiento en paralelo es cortada cuando el segundo elemento de procesamiento en paralelo está inactivo; o
- 45 una señal de reloj procedente del segundo elemento de procesamiento en paralelo es cortada cuando el segundo

elemento de procesamiento en paralelo está inactivo.

9.- El procedimiento de la reivindicación 1 en el cual la monitorización de una tarea comprende determinar la frecuencia con la que las instrucciones solicitan el procesamiento de datos del primer ancho.

10.- El procedimiento de la reivindicación 1, que comprende adicionalmente:

5 detectar una transición desde el primer estado hasta el segundo estado a partir de la monitorización de la condición, desactivar el segundo elemento de procesamiento en paralelo, en respuesta a la transición detectada desde el primer estado al segundo estado;

detectar una transición desde el primer estado hasta el segundo estado a partir de la monitorización de la una o más condiciones; y

10 activar el segundo elemento de procesamiento en paralelo, en respuesta a la transición detectada desde el segundo estado al primer estado.

11.- El procedimiento de la reivindicación 10, en el cual las etapas de activación y desactivación están adaptadas para mitigar la hiperpaginación potencial.

12.- Un procesador de datos en paralelo (11), que comprende:

15 una lógica (27, 29, 31) configurada para monitorizar una o más condiciones en relación con el procesamiento llevado a cabo por el procesador de datos en paralelo;

un primer elemento de procesamiento (23) y un segundo elemento de procesamiento (25), estando el segundo elemento de procesamiento en paralelo conectado para operar en paralelo con el primer elemento de procesamiento, proporcionando la operación en paralelo del primer y segundo elementos de procesamiento un procesamiento de datos en paralelo con un primer ancho cuando la una o más condiciones monitorizadas se correspondan con un primer estado, en el cual el primer ancho es mayor que el segundo ancho;

20 un primer procesador en paralelo configurado para ejecutar una o más instrucciones con el fin de procesar datos de un segundo ancho menor que el primer ancho, mientras el segundo elemento de procesamiento en paralelo está inactivo, cuando la una o más condiciones monitorizadas se correspondan con un segundo estado; en el cual la lógica configurada para monitorizar comprende

un sensor de temperatura configurado para monitorizar la temperatura del procesador de datos en paralelo, en el cual el primer estado está asociado con una temperatura monitorizada que está en, o por debajo de, un umbral de temperatura y el segundo estado está asociado con una temperatura que está en, o por encima de, un umbral; y

30 una lógica configurada para monitorizar una tarea efectuada por ejecución de la una o más instrucciones, en el cual en el primer estado la tarea solicita un procesamiento de datos con el primer ancho a una frecuencia superior a una frecuencia diana establecida.

13.- El procesador de datos en paralelo de la reivindicación 12, en el cual el primer y segundo elementos de procesamiento están configurados para procesar datos en respuesta a instrucciones del tipo Instrucción Única y Datos Múltiples (SIMD).

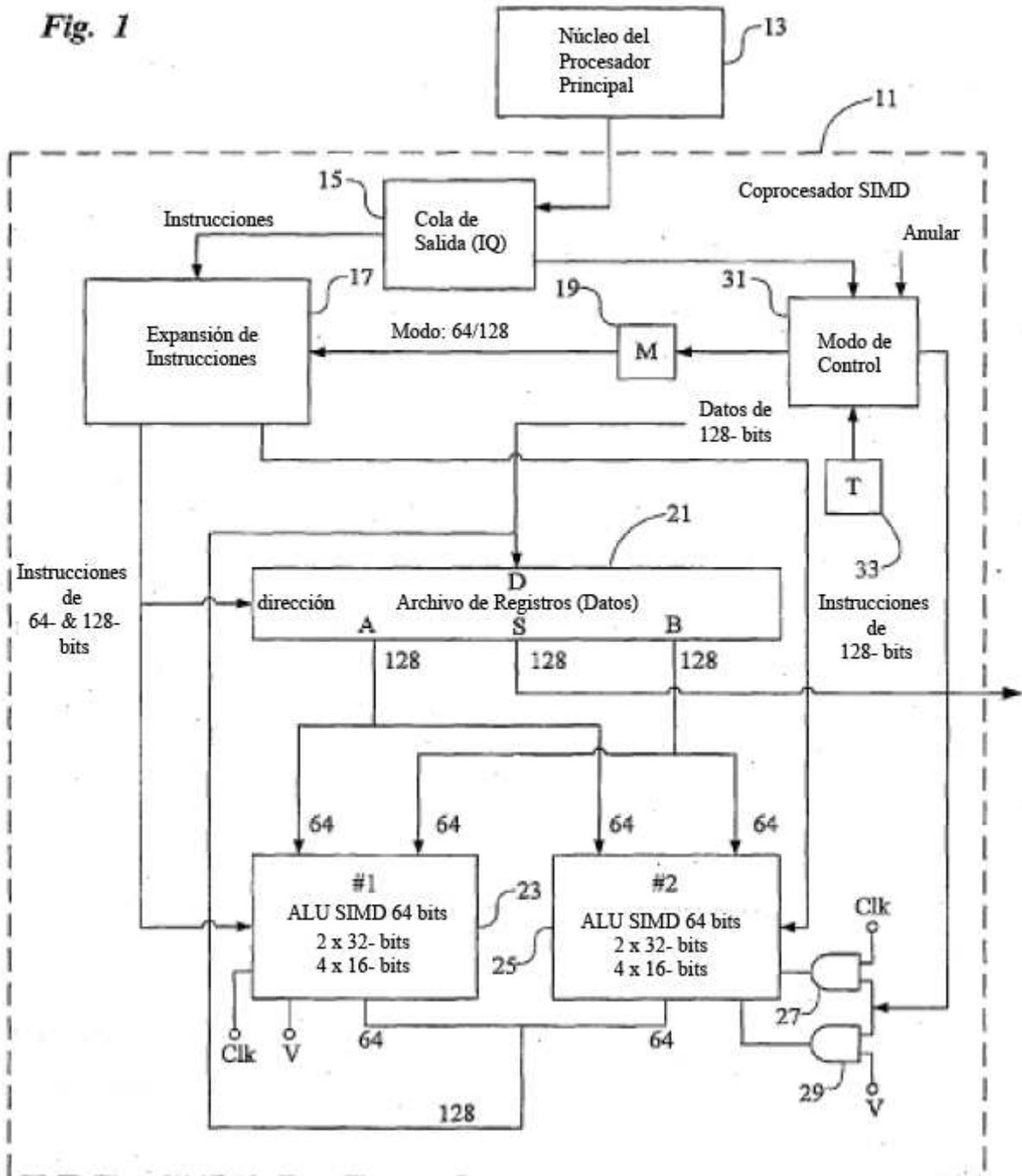
35 14.- El procesador de datos en paralelo de la reivindicación 13, en el cual cada uno del primer y segundo elementos de procesamiento comprende una unidad de lógica aritmética de tipo SIMD.

40 15.- El procesador de datos en paralelo de la reivindicación 12 ó 13, que comprende adicionalmente una etapa (17) de expansión de instrucciones, para convertir una instrucción SIMD que solicita que los datos del segundo ancho sean procesados para crear una pluralidad expandida de instrucciones SIMD que solicitan el procesamiento de los datos del primer ancho, estando la etapa de expansión de instrucciones configurada adicionalmente para suministrar secuencialmente la pluralidad expandida de instrucciones SIMD al primer elemento de procesamiento para su ejecución.

16.- El procesador de datos en paralelo de la reivindicación 12, en el cual la lógica está adaptada para controlar la activación y desactivación del segundo elemento de procesamiento para mitigar la potencial hiperpaginación.

45 17.- El procesador de datos en paralelo de la reivindicación 12, en el cual la lógica está adaptada para determinar si ya no queda por procesar ninguna instrucción de programa con el segundo ancho que esté en vuelo y, en respuesta, desactivar selectivamente el segundo elemento de procesamiento.

Fig. 1



2/2

Fig. 2

