(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 749 883**

(51) Int. CI.:

G06F 9/44 (2008.01)
G06F 15/78 (2006.01)
G06F 15/82 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **13.01.2012 PCT/US2012/021344**

(87) Fecha y número de publicación internacional: **19.07.2012 WO12097316**

(96) Fecha de presentación y número de la solicitud europea: **13.01.2012 E 12704456 (8)**

(97) Fecha y número de publicación de la concesión europea: **24.07.2019 EP 2663921**

(54) Título: **Canalización de recursos de cálculo en unidad de procesamiento de gráficos de propósito general**

(30) Prioridad:

14.01.2011 US 201113007333

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
24.03.2020

(73) Titular/es:

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

(72) Inventor/es:

BOURD, ALEXEI V.;
GRUBER, ANDREW;
KRSTIC, ALEKSANDRA L.;
SIMPSON, ROBERT J.;
SHARP, COLIN y
YU, CHUN

(74) Agente/Representante:

FORTEA LAGUNA, Juan José

DESCRIPCIÓN

Canalización de recursos de cálculo en unidad de procesamiento de gráficos de propósito general

CAMPO TÉCNICO

[0001] La presente divulgación se refiere al procesamiento de datos y, más en particular, al procesamiento de datos usando una unidad de procesamiento de gráficos de propósito general (GPGPU)

ANTECEDENTES

[0002] Las unidades de procesamiento de gráficos de propósito general (GPGPU) son versiones generalizadas de las unidades de procesamiento de gráficos diseñadas originalmente para procesar gráficos 2D y 3D. Las GPGPU extienden el procesamiento en paralelo de alta potencia de las GPU a las aplicaciones de procesamiento de datos de propósito general más allá del procesamiento de gráficos. Como ejemplo, una GPU puede estar configurada para procesar datos de acuerdo con la especificación OpenCL que da acceso a determinadas aplicaciones a la unidad de procesamiento de gráficos para cálculo no gráfico. La "especificación OpenCL, versión 1.1", se publicó en junio de 2010 y está disponible públicamente.

[0003] Las GPGPU incluyen unidades de procesamiento programables dispuestas en una estructura altamente paralela que no permite compartir ni sincronizar datos entre las unidades de procesamiento. En cambio, las unidades de procesamiento individuales solo intercambian conjuntos de datos con una memoria externa. Debido a esta estructura, las aplicaciones para las GPGPU están limitadas a las que son inherentemente paralelas. Las arquitecturas GPGPU pueden estar tan altamente paralelizadas que impiden la implementación eficaz de los cálculos basados en canalización. Esta limitación se extiende al procesamiento de gráficos 2D y 3D que usa el procesamiento en paralelo en cada fase de procesamiento, pero requiere la canalización de recursos de cálculo entre las fases.

[0004] El documento WO 02/46917 describe un aparato de procesamiento de señales digitales. El documento US 4807183 describe un chip de interconexión programable para módulos de funciones de sistema informático. El artículo "The GeForce 6800" de J. Montrym y H. Moreton, publicado en IEEE Micro, marzo de 2005, páginas 41-51 describe la arquitectura de la unidad de procesamiento de gráficos GeForce 6800.

[0005] El documento US 2010/0079454 describe un procesador de gráficos que divide los recursos de procesamiento dentro de una GPU en conjuntos para realizar diferentes operaciones de teselación con parámetros de vértice y teselación que se encaminan directamente de un recurso de procesamiento a otro en lugar de almacenarse en memoria.

[0006] El documento US 2007/0113232 describe la sincronización de subprocesos de programa.

[0007] El documento US 5.355.508 describe un sistema de procesamiento de datos en paralelo que combina una unidad SIMD con una unidad MIMD y comparte un bus común, memoria y controlador de sistema.

SUMARIO

[0008] La invención se define en las reivindicaciones independientes. Las características preferentes se definen en mayor detalle en las reivindicaciones dependientes. La presente divulgación describe técnicas para extender la arquitectura de una unidad de procesamiento de gráficos de propósito general (GPGPU) con unidades de procesamiento en paralelo para permitir un procesamiento eficaz de aplicaciones basadas en canalización. Configurando memorias intermedias locales conectadas a unidades de procesamiento en paralelo que funcionan como fases de una canalización de procesamiento para contener datos para transferencia entre las unidades de procesamiento en paralelo, las memorias intermedias locales permiten una transferencia directa de datos en chip de baja potencia entre las unidades de procesamiento en paralelo. Las memorias intermedias locales pueden incluir mecanismos de control de flujo de datos basados en hardware para permitir una transferencia de datos entre las unidades de procesamiento en paralelo. De esta forma, los datos se pueden pasar directamente de una unidad de procesamiento en paralelo a la siguiente unidad de procesamiento en paralelo en la canalización de procesamiento por medio de las memorias intermedias locales, transformando de hecho las unidades de procesamiento en paralelo en una serie de fases de canalización. Las memorias intermedias locales pueden reducir significativamente el uso de ancho de banda de memoria reduciendo o eliminando la necesidad de que cada una de las unidades de procesamiento en paralelo de la canalización de procesamiento realice llamadas a la memoria del sistema para recuperar y/o almacenar datos.

[0009] Las memorias intermedias locales pueden ser memorias intermedias basadas en hardware que son configurables para intercambiar profundidad por anchura.

[0010] Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y en la siguiente descripción. Otras características, objetivos y ventajas resultarán evidentes a partir de la descripción y de los dibujos, y a partir de las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0011]

La FIG. 1 es un diagrama de bloques que ilustra un dispositivo que incluye una unidad de procesamiento de gráficos de propósito general (GPGPU) que es configurable para implementar una canalización de procesamiento.

La FIG. 2 es un diagrama de bloques que ilustra una GPGPU convencional que incluye unidades de procesamiento en paralelo configuradas para realizar procesamiento en paralelo.

La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de la GPGPU de la FIG. 1 que incluye unidades de procesamiento en paralelo y memorias intermedias locales configuradas para implementar una canalización de procesamiento.

La FIG. 4 es un diagrama de flujo que ilustra un ejemplo de funcionamiento de la GPGPU que incluye memorias intermedias locales conectadas a unidades de procesamiento en paralelo para transferir datos entre las unidades de procesamiento en paralelo como fases de una canalización de procesamiento.

La FIG. 5 es un diagrama de flujo que ilustra un ejemplo de funcionamiento para mantener una secuencia de datos dentro de una canalización de procesamiento implementada por unidades de procesamiento en paralelo y memorias intermedias locales de la GPGPU.

DESCRIPCIÓN DETALLADA

[0012] La presente divulgación describe técnicas para extender la arquitectura de una unidad de procesamiento de gráficos de propósito general (GPGPU) con unidades de procesamiento en paralelo para permitir un procesamiento eficaz de aplicaciones basadas en canalización. Específicamente, las técnicas incluyen configurar memorias intermedias locales conectadas a unidades de procesamiento en paralelo que funcionan como fases de una canalización de procesamiento para contener datos para transferencia entre las unidades de procesamiento en paralelo. Las memorias intermedias locales permiten una transferencia directa de datos en chip de baja potencia entre las unidades de procesamiento en paralelo. Las memorias intermedias locales pueden incluir mecanismos de control de flujo de datos basados en hardware para permitir una transferencia de datos entre las unidades de procesamiento en paralelo. De esta forma, los datos se pueden pasar directamente de una unidad de procesamiento en paralelo a la siguiente unidad de procesamiento en paralelo en la canalización de procesamiento por medio de las memorias intermedias locales, transformando de hecho las unidades de procesamiento en paralelo en una serie de fases de canalización. Las memorias intermedias locales pueden reducir significativamente el uso de ancho de banda de memoria reduciendo o eliminando la necesidad de que cada una de las unidades de procesamiento en paralelo de la canalización de procesamiento realice llamadas a la memoria del sistema para recuperar y/o almacenar datos.

[0013] La FIG. 1 es un diagrama de bloques que ilustra un dispositivo 2 que incluye una unidad de procesamiento de gráficos de propósito general (GPGPU) 6 que es configurable para implementar una canalización de procesamiento 10. Como se describe en más detalle a continuación, la canalización de procesamiento 10 de la GPGPU 6 incluye dos o más unidades de procesamiento en paralelo configuradas para funcionar como fases de la canalización de procesamiento 10 y una o más memorias intermedias locales configuradas para contener datos para transferir entre las unidades de procesamiento en paralelo para implementar la canalización de procesamiento 10.

[0014] El dispositivo 2 es capaz de transmitir y recibir datos, admitiendo una variedad de aplicaciones de procesamiento de datos y facilitando datos procesados para su presentación a un usuario. Los ejemplos del dispositivo 2 incluyen, pero no se limitan a, teléfonos móviles inalámbricos, asistentes digitales personales (PDA), dispositivos de videojuegos, consolas de videojuegos, unidades de videoconferencia, ordenadores portátiles, ordenadores de escritorio, tabletas, descodificadores de televisión, dispositivos de grabación digital, reproductores de medios digitales y similares.

[0015] En el ejemplo ilustrado en la FIG. 1, el dispositivo 2 incluye un procesador principal 4, una GPGPU 6 con una canalización de procesamiento 10, una pantalla 8, un altavoz 10, una memoria de dispositivo 12, un módulo transceptor 14 y un dispositivo de entrada de usuario 16. En otros casos, por ejemplo, cuando el dispositivo 2 es un ordenador de escritorio, la pantalla 8, el altavoz 10 y/o la interfaz de usuario 16 pueden ser externos al dispositivo 2. El procesador principal 4 y la GPGPU 6 pueden comprender procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables *in situ* (FPGA) u otros circuitos lógicos integrados o discretos equivalentes.

[0016] El procesador principal 4 puede ejecutar una o más aplicaciones. Los ejemplos de las aplicaciones incluyen navegadores web, aplicaciones de correo electrónico, hojas de cálculo, videojuegos, aplicaciones de edición de audio y vídeo, u otras aplicaciones que generan una salida visual y/o de audio para presentar a un usuario por medio de la pantalla 8 y/o el altavoz 10. La GPGPU 6 también puede ejecutar una o más aplicaciones. La GPGPU 6 puede ejecutar aplicaciones en apoyo de las aplicaciones ejecutadas por el procesador principal 4. Específicamente, la GPGPU 6 puede ejecutar aplicaciones para preparar datos para la presentación a un usuario por medio de la pantalla 8 y/o el altavoz 10.

[0017] La GPGPU 6 es una versión generalizada de una unidad de procesamiento de gráficos (GPU) que extiende el procesamiento en paralelo de alta potencia de la GPU a aplicaciones de procesamiento de datos de propósito general más allá del procesamiento de gráficos. Como ejemplo, la GPGPU 6 puede estar configurada para procesar datos de acuerdo con la especificación OpenCL que da a determinadas aplicaciones acceso a una GPU para cálculo no gráfico. Las GPGPU convencionales, descritas con más detalle a continuación con respecto a la FIG. 2, incluyen unidades de procesamiento programables dispuestas en una estructura altamente paralela que impide la implementación eficaz de aplicaciones basadas en canalización. Esta limitación se extiende a las aplicaciones de procesamiento de gráficos 2D y 3D que usan procesamiento en paralelo en cada fase de procesamiento, pero requieren canalización de recursos de cálculo entre las fases.

[0018] Las aplicaciones basadas en canalización requieren que se procese un conjunto de datos en fases, de modo que una primera fase está configurada para procesar un conjunto de datos original, una segunda fase está configurada para procesar la salida de la primera fase, una tercera etapa está configurada para procesar la salida de la tercera fase, y así sucesivamente para el número de fases requeridas para la aplicación. La implementación más eficaz de las aplicaciones basadas en canalización consiste en pasar los conjuntos de datos directamente de una fase a la siguiente fase en la canalización de procesamiento. Una implementación menos eficaz de las aplicaciones basadas en canalización consiste en que cada fase de la canalización de procesamiento recupere los datos procesados por una fase anterior desde una memoria fuera de chip y a continuación almacene nuevamente los datos procesados en la memoria fuera de chip para la siguiente fase. Esta implementación menos eficaz sigue requiriendo mecanismos de secuenciación para asegurar que cada fase de la canalización de procesamiento procesa el conjunto de datos en la secuencia correcta. Las GPGPU convencionales no se pueden configurar para implementar canalizaciones de procesamiento o incluso los mecanismos de secuenciación necesarios para realizar aplicaciones basadas en canalización.

[0019] De acuerdo con las técnicas de la presente divulgación, y a diferencia de las GPGPU convencionales, en algunos ejemplos, la GPGPU 6 es configurable para implementar la canalización de procesamiento 10 para ejecutar aplicaciones basadas en canalización, que incluyen las aplicaciones de procesamiento de gráficos 2D y 3D. Como se describe en más detalle a continuación con respecto a la FIG. 3, la canalización de procesamiento 10 de la GPGPU 6 incluye dos o más unidades de procesamiento en paralelo configuradas para funcionar como fases de la canalización de procesamiento 10 y una o más memorias intermedias locales configuradas para contener datos para la transferencia entre las unidades de procesamiento en paralelo para implementar una canalización de procesamiento 10. Las memorias intermedias locales incluidas en la canalización de procesamiento 10 permiten la transferencia directa de datos en chip de baja potencia entre las unidades de procesamiento en paralelo. De esta manera, los datos se pueden pasar directamente desde una unidad de procesamiento en paralelo a la siguiente unidad de procesamiento en paralelo de la canalización de procesamiento 10 por medio de las memorias intermedias locales, transformando de hecho las unidades de procesamiento en paralelo en una serie de fases de canalización. La implementación de la canalización de procesamiento 10 puede reducir significativamente el uso del ancho de banda de memoria reduciendo o eliminando la necesidad de que cada una de las unidades de procesamiento en paralelo de la canalización de procesamiento 10 realice llamadas a la memoria del dispositivo 12, localizada fuera del chip de la GPGPU 6, para recuperar y/o almacenar datos.

[0020] Las técnicas de la presente divulgación pueden incluir configurar cada una de las memorias intermedias locales dentro de la canalización de procesamiento 10 para que tenga una anchura requerida para que la memoria intermedia contenga datos facilitados desde una unidad de procesamiento en paralelo anterior. Por ejemplo, las memorias intermedias locales pueden ser memorias intermedias basadas en hardware que son configurables para intercambiar profundidad por anchura. Además, las técnicas incluyen la ejecución de barreras de secuenciación para mantener una secuencia de datos dentro de la canalización de procesamiento 10. Por ejemplo, una secuencia de los subprocesos de datos de un conjunto de datos se puede registrar tras la entrada del conjunto de datos en una unidad de procesamiento en paralelo dentro de la canalización de procesamiento 10 y, después de que el conjunto de datos se haya procesado, los subprocesos de datos del conjunto de datos se pueden liberar de la unidad de procesamiento en paralelo en la misma secuencia que la registrada.

[0021] Por ejemplo, cuando la GPGPU 6 está configurada para implementar la canalización de procesamiento 10, la GPGPU 6 puede ejecutar aplicaciones de procesamiento de gráficos 2D y 3D basadas en canalización en apoyo del navegador web, correo electrónico, videojuegos y aplicaciones de edición de vídeo ejecutadas por el procesador principal 4. Como otro ejemplo, cuando la GPGPU 6 no está configurada para implementar la canalización de procesamiento 10, la GPGPU 6 puede ejecutar aplicaciones que funcionan eficazmente en una estructura altamente paralela, tales como aplicaciones de búsqueda basadas en imágenes, generación/extracción de descriptor de

imágenes, ajustes radiométricos de imágenes, procesamiento de audio y otras operaciones realizadas típicamente por el procesador principal 4.

[0022] En algunos casos, la GPGPU 6 puede ejecutar aplicaciones en apoyo de aplicaciones de procesamiento de gráficos basadas en canalización. La propia GPGPU 6 puede ejecutar aplicaciones de procesamiento de gráficos basadas en canalización usando la canalización de procesamiento 10 o una GPU separada incluida en el dispositivo 2. Por ejemplo, la GPGPU 6 puede ejecutar aplicaciones de efectos especiales de imágenes, generación de vértices para una canalización de GPU y aplicaciones de posprocesamiento de gráficos usando una memoria intermedia de color de una canalización de GPU.

[0023] La pantalla 8 y el altavoz 10 comprenden dispositivos de salida para el dispositivo 2. En algunos casos, la pantalla 8 y el altavoz 10 se pueden usar conjuntamente para presentar una salida tanto visual como de audio a un usuario. En otros casos, la pantalla 8 y el altavoz 10 se pueden usar por separado para presentar la salida al usuario. Como ejemplo, la pantalla 8 puede comprender una pantalla de cristal líquido (LCD), una pantalla de tubo de rayos catódicos (CRT), una pantalla de plasma u otro tipo de dispositivo de visualización.

[0024] El dispositivo de entrada de usuario 16 comprende uno o más dispositivos de entrada de usuario para el dispositivo 2. Por ejemplo, el dispositivo de entrada de usuario 16 puede incluir una bola de seguimiento, un ratón, un teclado, un micrófono y/u otros tipos de dispositivos de entrada. En algunos ejemplos, el dispositivo de entrada de usuario 16 puede comprender una pantalla táctil y se puede incorporar como parte de la pantalla 8. Un usuario puede seleccionar la una o más aplicaciones que el procesador principal 4 y/o la GPGPU 6 van a ejecutar, por medio del dispositivo de entrada de usuario 16.

[0025] El procesador principal 4 puede descargar datos que el procesador principal 4 y/o la GPGPU 6 van a procesar, por medio del módulo transceptor 14. El procesador principal 4 también puede descargar la una o más aplicaciones ejecutadas por el procesador principal 4 y/o la GPGPU 6 por medio del módulo transceptor 14. El módulo de transceptor 14 puede incluir circuitos para permitir la comunicación inalámbrica o alámbrica entre el dispositivo 2 y otro dispositivo o una red. El módulo de transceptor 14 puede incluir moduladores, desmoduladores, amplificadores y otros de dichos circuitos para comunicación alámbrica o inalámbrica.

[0026] La memoria de dispositivo 12 puede almacenar datos que el procesador principal 4 y/o la GPGPU 6 van a procesar, y también puede almacenar datos procesados recibidos desde el procesador principal 4 y/o la GPGPU 6. Además, la memoria de dispositivo 12 puede almacenar la una o más aplicaciones ejecutadas por el procesador principal 4 y/o la GPGPU 6. La memoria de dispositivo 12 puede comprender uno o más medios de almacenamiento legibles por ordenador. Los ejemplos de memoria de dispositivo 12 incluyen, pero no se limitan a, una memoria de acceso aleatorio (RAM), una memoria de solo lectura (ROM), una memoria de solo lectura programable y borrable eléctricamente (EEPROM), una CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, memoria *flash* u otro medio que se pueda usar para transportar o almacenar un código de programa deseado en forma de instrucciones o estructuras de datos y al que se pueda acceder mediante un ordenador o un procesador.

[0027] La FIG. 2 es un diagrama de bloques que ilustra una GPGPU 18 convencional que incluye unidades de procesamiento en paralelo 22A-22D configuradas para realizar un procesamiento en paralelo. En algunos ejemplos, la GPGPU 18 puede estar incluida dentro de un dispositivo sustancialmente similar al dispositivo 2 descrito anteriormente con referencia a la FIG. 1. La GPGPU 18 incluye una unidad de distribución de datos 20, unas unidades de procesamiento en paralelo 22A-22D ("unidades de procesamiento en paralelo 22") y un bus 24 para conectar las unidades de procesamiento en paralelo 22 a la memoria de dispositivo 26 externa a la GPGPU 18.

[0028] La GPGPU convencional 18 es una versión generalizada de una GPU diseñada originalmente para procesar gráficos 2D y 3D. La GPGPU 18 es capaz de extender el procesamiento en paralelo de alta potencia de una GPU a aplicaciones de procesamiento de propósito general más allá del procesamiento de gráficos. Como ejemplo, la GPGPU 18 puede estar configurada para procesar datos de acuerdo con la especificación OpenCL. La especificación OpenCL da acceso a determinadas aplicaciones a una GPU para cálculo no gráfico. En terminología OpenCL, los subprocesos de datos se denominan elementos de trabajo, los conjuntos de datos se denominan grupos de trabajo, las unidades de procesamiento se denominan unidades de cálculo y una colección de unidades de procesamiento se denomina grupo de cálculo.

[0029] Una tarea típica de GPU es altamente paralela y no requiere intercambio de información entre subprocesos de datos de un conjunto de datos que se procesa dentro de una unidad de procesamiento dada. Por ejemplo, los valores calculados para un vértice son independientes de los valores calculados para un vértice diferente, y los valores calculados para un píxel son independientes de los valores calculados para un píxel diferente. A fin de imitar la naturaleza paralela de una GPU, la GPGPU 18 está diseñada para incluir unidades de procesamiento en paralelo 22 dispuestas en una estructura altamente paralela.

[0030] La arquitectura de la GPGPU 18 es tan altamente paralela que no permite compartir ni sincronizar datos entre unidades de procesamiento en paralelo 22. En funcionamiento, la unidad de distribución de datos 20 asigna un

conjunto de datos almacenado en la memoria de dispositivo 26 a cada una de las unidades de procesamiento en paralelo 22. Durante el procesamiento, se pueden compartir y sincronizar, dentro de cada una de las unidades de procesamiento en paralelo 22, subprocesos de datos de un conjunto de datos asignado. Sin embargo, no se pueden compartir ni sincronizar subprocesos de datos de diferentes conjuntos de datos entre unidades de procesamiento en paralelo 22. En su lugar, cada una de las unidades de procesamiento en paralelo 22 solo intercambia los conjuntos de datos asignados con la memoria de dispositivo 26 por medio del bus 24. Más específicamente, cada una de las unidades de procesamiento en paralelo 22 recupera los conjuntos de datos asignados para el procesamiento desde la memoria de dispositivo 26 por medio del bus 24 y, después de procesar los conjuntos de datos, vuelve a almacenar los conjuntos de datos procesados en la memoria de dispositivo 26 por medio del bus 24.

[0031] La arquitectura paralela de la GPGPU 18 impide la implementación eficaz de aplicaciones basadas en canalización entre unidades de procesamiento en paralelo 22. En aplicaciones basadas en canalización, las unidades de procesamiento están conectadas como fases en una canalización para permitir que los datos se desplacen de una fase a otra fase para diferentes tareas de procesamiento. La limitación contra las aplicaciones basadas en canalización en la GPGPU 18 se extiende a las aplicaciones de procesamiento de gráficos 2D y 3D, que usan el procesamiento en paralelo en cada fase de procesamiento, pero requieren canalización entre las fases.

[0032] Las aplicaciones para la GPGPU 18 están, por lo tanto, limitadas a las que son inherentemente paralelas. Cada una de las unidades de procesamiento en paralelo 22 puede comprender una agrupación de unidades aritméticas lógicas (ALU) u otros elementos lógicos configurables. Las unidades de procesamiento en paralelo 22 son, por lo tanto, programables o configurables para realizar diferentes operaciones dependiendo de la aplicación ejecutada por la GPGPU 18. Las aplicaciones que funcionan eficazmente en la estructura altamente paralela de la GPGPU 18 pueden incluir aplicaciones de búsqueda basadas en imágenes, generación/extracción de descriptor de imagen, ajustes radiométricos de imágenes, procesamiento de audio, otras operaciones típicamente realizadas por un procesador de señales digitales (DSP) y similares. Además, las aplicaciones ejecutadas por la GPGPU 18 pueden requerir interacción con aplicaciones de procesamiento de gráficos basadas en canalización, tales como la generación de efectos especiales de imágenes, la generación de vértices para una canalización de GPU y operaciones de posprocesamiento de gráficos usando una memoria intermedia de color de una canalización de GPU.

[0033] La FIG. 3 es un diagrama de bloques que ilustra una GPGPU 6 ejemplar de la FIG. 1, que incluye las unidades de procesamiento en paralelo 42A-42D y las memorias intermedias locales 44A-44C configuradas para implementar una canalización de procesamiento 10. En otros ejemplos, la GPGPU 6 puede incluir más o menos unidades de procesamiento en paralelo y memorias intermedias locales.

[0034] En el ejemplo de la FIG. 3, la GPGPU 6 incluye una unidad de distribución de datos 40, unas unidades de procesamiento en paralelo 42A-42D ("unidades de procesamiento en paralelo 42") y un bus 46 para conectar las unidades de procesamiento en paralelo 42 a la memoria de dispositivo 12 (de la FIG. 1) externa a la GPGPU 6. A diferencia de una GPGPU convencional, como la GPGPU 18 de la FIG. 3, la GPGPU 6 también incluye memorias intermedias locales 44A-44C ("memorias intermedias locales 44") conectadas entre unidades de procesamiento en paralelo 42. La combinación de unidades de procesamiento en paralelo 42 y memorias intermedias locales 44 conectadas entre unidades de procesamiento en paralelo 42 se puede denominar canalización de procesamiento 10. La GPGPU 6 también incluye una unidad de control 30 y una memoria local 38. La memoria local 38 puede comprender una memoria intermedia similar a las memorias intermedias locales 44, un registro o una memoria caché que almacena temporalmente datos para la GPGPU 6. La unidad de control 30 incluye interfaces de programación de aplicaciones (API) 32, un gestor de memoria intermedia 34 y gestor de secuencia 36.

[0035] Las memorias intermedias locales 44 pueden incluir mecanismos de control de flujo de datos basados en hardware para permitir la transferencia de datos entre unidades de procesamiento en paralelo 42. Por ejemplo, las memorias intermedias locales 44 pueden comprender memorias intermedias de primera entrada, primera salida (FIFO) basadas en hardware u otros tipos de memorias intermedias basadas en hardware, tales como memorias intermedias de última entrada, primera salida (LIFO) o memorias intermedias indexadas. En el caso en que la memoria intermedia local 44A comprende una memoria FIFO basada en hardware, por ejemplo, la memoria intermedia local 44A incluye mecanismos de control de flujo de datos que permiten que la unidad de procesamiento en paralelo 42A envíe datos a la memoria intermedia local 44A cuando hay espacio para escribir datos en la memoria intermedia, y de lo contrario detenga la petición de escritura. En ese caso, la memoria intermedia local 44A también incluye mecanismos de control de flujo de datos que permiten que la unidad de procesamiento en paralelo 42B reciba datos desde la memoria intermedia local 44A cuando hay datos disponibles para leer desde la memoria intermedia, y de lo contrario detenga la petición de lectura. Cuando las memorias intermedias locales 44 incluyen mecanismos de control de flujo de datos basados en hardware, no son necesarios controles de flujo de datos basados en software menos eficaces para permitir la transferencia de datos entre unidades de procesamiento en paralelo 42.

[0036] Las memorias intermedias locales 44 permiten la transferencia directa de datos en chip de baja potencia entre unidades de procesamiento en paralelo 42. Las memorias intermedias locales 44 son "locales" porque están localizadas dentro de la GPGU 6 y en el mismo chip que las unidades de procesamiento 42. De esta manera, los datos se pueden pasar directamente desde una de las unidades de procesamiento en paralelo 42 a otra de las unidades de procesamiento en paralelo 42 en la canalización de procesamiento 10 por medio de memorias intermedias locales 44.

No se requiere que las unidades de procesamiento en paralelo 42 recuperen y almacenen datos repetidamente con la memoria del dispositivo 12, que es externa a o se encuentra fuera del chip de la GPGPU 6. Las memorias intermedias locales 44, por lo tanto, transforman las unidades de procesamiento en paralelo 42 en una serie de fases de canalización e implementan la canalización de procesamiento 10 dentro de la GPGPU 6.

[0037] En el ejemplo ilustrado, cada una de las memorias intermedias locales 44 está conectada directamente entre dos de las unidades de procesamiento en paralelo 42 en orden sucesivo, de modo que la canalización de procesamiento 10 es una canalización puramente en serie. Las memorias intermedias locales 44 están conectadas "directamente" en la medida en que solo son accesibles por las dos unidades de procesamiento en paralelo 42 a las que están conectadas y no son direccionables mediante bus por ninguna de las unidades de procesamiento en paralelo 42. Por ejemplo, la memoria intermedia local 44A está conectada directamente entre las unidades de procesamiento en paralelo 42A y 42B, la memoria intermedia local 44B está directamente conectada entre las unidades de procesamiento en paralelo 42B y 42C, y la memoria intermedia local 44C está directamente conectada entre las unidades de procesamiento en paralelo 42C y 42D.

[0038] En otros ejemplos, cada una de las memorias intermedias locales 44 también puede estar directamente conectada a una o más de las unidades de procesamiento en paralelo 42 que no están en orden sucesivo. En este caso, cada una de las memorias intermedias locales 44 puede estar directamente conectada a cualquiera de las unidades de procesamiento en paralelo 42 por medio de conexiones de barras cruzadas. Por ejemplo, la memoria intermedia local 44A puede estar directamente conectada a cada una de las unidades de procesamiento en paralelo 42 por medio de una conexión de barras cruzadas, de modo que la unidad de procesamiento en paralelo 42A puede transferir datos a cualquiera de las unidades de procesamiento en paralelo 42B-42D por medio de la memoria intermedia local 44A. El uso de conexiones de barras cruzadas hace que las memorias intermedias locales 44 sean más ampliamente accesibles para las unidades de procesamiento en paralelo 42 y permite la implementación de canalización de procesamiento que no son puramente en serie.

[0039] En el ejemplo ilustrado en el que la canalización de procesamiento 10 comprende una canalización puramente en serie, las unidades de procesamiento en paralelo 42 solo pueden tener permiso para escribir datos en una sucesiva de las memorias intermedias locales 44, y pueden tener permiso solo para leer datos de una anterior de las memorias intermedias locales 44. Por ejemplo, la unidad de procesamiento en paralelo 42B puede ser solo capaz de leer datos de la memoria intermedia local 44A y solo ser capaz de escribir datos en la memoria intermedia local 44B. En los casos en que la canalización de procesamiento puede incluir conexiones de barras cruzadas, la unidad de procesamiento en paralelo 42 puede tener permiso tanto para leer como para escribir en cualquiera de las memorias intermedias locales 44. Por ejemplo, la unidad de procesamiento en paralelo 42B puede ser capaz de leer y escribir datos con la memoria intermedia local 44A y con la memoria intermedia local 44B.

[0040] Como se describe anteriormente, las memorias intermedias locales 44 pueden comprender al menos una de unas memorias intermedias FIFO, memorias intermedias LIFO o memorias intermedias indexadas. El tipo de memoria intermedia usada para las memorias intermedias locales 44 puede depender del tipo de mecanismos de control de flujo de datos basados en hardware requeridos en la canalización de procesamiento 10. El tipo de memoria intermedia usada para las memorias intermedias locales 44 también puede depender de si las memorias intermedias locales 44 están conectadas a unidades de procesamiento en paralelo 42 por medio de conexiones de uno a uno o conexiones de barras cruzadas. Además, cuando se usan conexiones de barras cruzadas, el gestor de memoria intermedia 34 de la unidad de control 30 puede necesitar realizar determinado control de memoria para gestionar qué unidad de procesamiento en paralelo 42 tiene acceso y a qué a memoria intermedia local 44 lo tiene en un momento dado.

[0041] Como se describe anteriormente, las memorias intermedias locales 44 pueden estar directamente conectadas entre al menos dos de las unidades de procesamiento en paralelo 42 por medio de conexiones de uno a uno o de barras cruzadas. Sin embargo, las memorias intermedias locales 44 pueden no ser direccionables mediante bus por las unidades de procesamiento en paralelo 42. De esta manera, un controlador de memoria designado para las memorias intermedias locales 44 puede no ser necesario. Específicamente, no es necesario un controlador de memoria para procesar mandatos de lectura y escritura en las memorias intermedias locales 44 por medio de un bus.

[0042] Las memorias intermedias locales 44 pueden reducir significativamente el uso de ancho de banda de memoria reduciendo o eliminando la necesidad de que cada una de las unidades de procesamiento en paralelo 42 realice llamadas a la memoria de dispositivo 12 por medio del bus 46 para recuperar y/o almacenar datos. En funcionamiento, la unidad de procesamiento en paralelo 42A, como primera unidad de procesamiento de la canalización de procesamiento 10, recupera un conjunto de datos original de la memoria de dispositivo 12 por medio del bus 46. La unidad de distribución de datos 40 puede asignar el conjunto de datos a la unidad de procesamiento en paralelo 42A. Además, la unidad de procesamiento en paralelo 42D, como unidad de procesamiento final de la canalización de procesamiento 10, almacena un conjunto de datos postcanalización en la memoria de dispositivo 12 por medio del bus 46. Las unidades de procesamiento en paralelo 42B y 42C, como unidades de procesamiento intermedias de la canalización de procesamiento 10, reciben el conjunto de datos de una anterior de las unidades de procesamiento en paralelo 42 por medio de una de las memorias intermedias locales 44, y envían el conjunto de datos a una posterior de las unidades procesamiento en paralelo 42 por medio de una de las memorias intermedias locales 44. Por lo tanto, no se requiere que las unidades de procesamiento intermedias interactúen con la memoria de dispositivo 12 para

recuperar y/o almacenar datos. En algunos casos, las unidades de procesamiento intermedias pueden recuperar datos complementarios de la memoria de dispositivo a fin de realizar la fase particular de la canalización de procesamiento 10. El conjunto de datos principal para procesamiento, sin embargo, se pasa directamente a lo largo de la canalización de procesamiento 10 por medio de las memorias intermedias locales 44.

[0043] Como se describe anteriormente, la GPGPU 6 es una versión generalizada de una GPU que extiende el procesamiento en paralelo de alta potencia de la GPU a aplicaciones de procesamiento de datos de propósito general más allá del procesamiento de gráficos. Como ejemplo, la GPGPU 6 puede estar configurada para procesar datos de acuerdo con la especificación OpenCL que da acceso a determinadas aplicaciones a una unidad de procesamiento de gráficos para cálculo no gráfico. En terminología OpenCL, los subprocesos de datos se denominan elementos de trabajo, los conjuntos de datos se denominan grupos de trabajo, las unidades de procesamiento se denominan unidades de cálculo y una colección de unidades de procesamiento se denomina grupo de cálculo.

[0044] De acuerdo con las técnicas de la presente divulgación, la GPGPU 6 es configurable para implementar la canalización de procesamiento 10 para ejecutar aplicaciones basadas en canalización, que incluyen las aplicaciones de procesamiento de gráficos 2D y 3D. Más específicamente, la unidad de control 30 de la GPGPU 6 configura las unidades de procesamiento en paralelo 42 para que funcionen como fases de una canalización de procesamiento. La unidad de control 30 también configura las memorias intermedias locales 44 conectadas entre las unidades de procesamiento en paralelo 42 para que almacenen datos para la transferencia entre las unidades de procesamiento en paralelo 42.

[0045] Las unidades de procesamiento en paralelo 42 pueden ser programables o configurables para realizar diferentes operaciones dependiendo de la aplicación ejecutada por la GPGPU 6. La unidad de control 30 puede configurar cada una de las unidades de procesamiento en paralelo 42 para que funcione de acuerdo con la aplicación. Por ejemplo, cada una de las unidades de procesamiento en paralelo 22 puede comprender una agrupación de unidades aritméticas lógicas (ALU) u otros elementos lógicos configurables.

[0046] Las memorias intermedias locales 44 también pueden ser programables o configurables para contener diferentes tipos de datos facilitados por las unidades de procesamiento en paralelo 42 dependiendo de la aplicación ejecutada por la GPGPU 6. Por ejemplo, las memorias intermedias locales 44 pueden comprender memorias intermedias basadas en hardware, pero incluyen un conjunto de aspectos configurables. Uno de los aspectos configurables puede ser la anchura de las memorias intermedias locales 44 a fin de adaptarse a los diferentes tipos de datos facilitados por las unidades de procesamiento en paralelo 42. Por ejemplo, las memorias intermedias locales 44 pueden ser configurables para intercambiar profundidad por anchura. El gestor de memoria intermedia 34 de la unidad de control 30 puede determinar una anchura requerida para cada una de las memorias intermedias locales 44 para que almacene los datos facilitados por una anterior de las unidades de procesamiento en paralelo 42. El gestor de memoria intermedia 34 puede conocer el tipo de datos facilitado por cada una de las unidades de procesamiento en paralelo 42 y, por lo tanto, conocer las anchuras requeridas por cada una de las memorias intermedias locales 44 para contener los datos. A continuación, el gestor de memoria intermedia 34 puede configurar cada una de las memorias intermedias locales 44 para tener la anchura determinada.

[0047] Una vez que las unidades de procesamiento en paralelo 42 y las memorias intermedias locales 44 están configuradas para implementar la canalización de procesamiento 10 dentro de la GPGPU 6, las unidades de procesamiento en paralelo 42 pueden transferir datos por medio de las memorias intermedias locales 44. La unidad de control 30 puede configurar una o más de las unidades de procesamiento en paralelo 42 para enviar datos a las memorias intermedias locales 44, y configurar una o más de las unidades de procesamiento en paralelo 44 para recibir datos desde las memorias intermedias locales 44. Por ejemplo, la unidad de control 30 puede configurar las unidades de procesamiento en paralelo 42A, 42B y 42C para que envíen datos a las memorias intermedias locales 44A, 44B y 44C, respectivamente. La unidad de control 30 también puede configurar las unidades de procesamiento en paralelo 42B, 42C y 42D para que reciban datos desde las memorias intermedias locales 44A, 44B y 44C, respectivamente.

[0048] Las memorias intermedias locales 44 con mecanismos de control de flujo basados en hardware se pueden exponer usando un estándar GPGPU, como el estándar OpenCL, introduciendo nuevas API 32. Por ejemplo, la unidad de control 30 puede ejecutar una o más de las API 32 para determinar la anchura requerida para cada una de las memorias intermedias locales 44, configurar cada una de las memorias intermedias locales 44 con la anchura determinada y determinar una profundidad de cada una de las memorias intermedias locales 44. Además, la unidad de control 30 puede ejecutar una o más de las API 32 para configurar las unidades de procesamiento en paralelo 42 para que envíen datos a las memorias intermedias locales 44 y recibir datos desde las memorias intermedias locales 44. Los mecanismos de control de flujo de datos basados en hardware incluidos en las memorias intermedias locales 44 permiten que las unidades de procesamiento en paralelo 42 envíen datos a y reciban datos desde las memorias intermedias locales 44 sin ningún control adicional de flujo de datos basado en software.

[0049] Además, la unidad de control 30 de la GPGPU 6 puede mantener la secuencia de datos dentro de la canalización de procesamiento 10 manteniendo la secuencia de datos dentro de una o más de las unidades de procesamiento en paralelo 42. Las aplicaciones basadas en canalización ejecutadas por la GPGPU 6, específicamente las aplicaciones de gráficos 3D, pueden requerir que los datos se procesen en una secuencia determinada dentro de

la canalización de procesamiento 10. Cuando los datos se procesan en cada fase de la canalización de procesamiento, los datos pueden cambiar la secuencia debido a problemas de ejecución, tales como condicionantes, resultados positivos o negativos de memoria caché y similares. El gestor de secuencia 36 de la unidad de control 30 puede ejecutar barreras de secuenciación para mantener la secuencia de datos dentro de al menos algunas de las unidades de procesamiento en paralelo 42. Las barreras de secuenciación pueden reducir la velocidad de procesamiento dentro de la canalización de procesamiento 10, por lo que el gestor de secuencia 36 solo podría ejecutar las barreras de secuenciación en las unidades de procesamiento en paralelo 42 que requieran que se mantenga la secuencia de datos para un procesamiento exacto.

[0050] Las barreras de secuenciación ejecutadas por el gestor de secuencia 36 pueden incluir un contador de determinación de secuencia (SDC) y una barrera de imposición de secuencia (SEB). Por ejemplo, las barreras de secuenciación se pueden exponer usando un estándar de GPGPU, tal como el estándar Open CL, añadiendo nuevas llamadas de función al lenguaje OpenCL C para el SDC y la SEB.

[0051] El gestor de secuencia 36 puede ejecutar el SDC tras la entrada de un conjunto de datos en cualquiera de las unidades de procesamiento en paralelo 42. A continuación, el gestor de secuencia 36 realiza la operación SDC registrando una secuencia de subprocesos de datos del conjunto de datos recibido dentro de la memoria local 38. Por ejemplo, el gestor de secuencia 36 puede registrar un índice de cada subproceso de datos del conjunto de datos en el orden en que se reciben los subprocesos de datos desde la memoria de dispositivo 12.

[0052] El gestor de secuencia 36 puede ejecutar la SEB tras la salida del conjunto de datos de la una de las unidades de procesamiento en paralelo 42. El gestor de secuencia 36 realiza a continuación la operación SEB liberando los subprocesos de datos del conjunto de datos de la una de las unidades de procesamiento en paralelo 42 en la misma secuencia que la registrada por el SDC. Por ejemplo, el gestor de secuencia 36 puede acceder a los índices de subproceso de datos registrados en la memoria local 38, y liberar cada subproceso de datos de acuerdo con el orden en que se ha registrado su índice. De esta manera, los subprocesos de datos del conjunto de datos entrarán en una posterior de las unidades de procesamiento en paralelo 42 en el mismo orden en que los subprocesos de datos del conjunto de datos entraron en la actual de las unidades de procesamiento en paralelo 42.

[0053] En un ejemplo, la unidad de control 30 puede configurar la GPGPU 6 para que ejecute una aplicación de procesamiento de gráficos 3D basada en canalización. En ese caso, la unidad de control 30 puede configurar las unidades de procesamiento en paralelo 42 para que funcionen como fases de una canalización de procesamiento de gráficos 3D. Por ejemplo, la unidad de control 30 puede configurar la unidad de procesamiento en paralelo 42A para que funcione como un sombreador de vértices, la unidad de procesamiento en paralelo 42B para que funcione como un rasterizador triangular, la unidad de procesamiento en paralelo 42C para que funcione como un sombreador de fragmentos, y la unidad de procesamiento en paralelo 42D para que funcione como un mezclador de píxeles.

[0054] La unidad de control 30 también puede configurar las memorias intermedias locales 44 con mecanismos de control de flujo de datos basados en hardware para contener datos para la transferencia entre las unidades de procesamiento en paralelo 42 para implementar la canalización de procesamiento de gráficos 3D 10. Por ejemplo, la unidad de control 30 puede configurar la memoria intermedia local 44A para que contenga datos de vértice del sombreador posvértice para la transferencia entre la unidad de procesamiento en paralelo 42A que funciona como sombreador de vértice y la unidad de procesamiento en paralelo 42B que funciona como rasterizador triangular. La unidad de control 30 puede configurar la memoria intermedia local 44B para que contenga datos de píxeles del sombreador de prefragmentos para la transferencia entre la unidad de procesamiento en paralelo 42B que funciona como rasterizador triangular y la unidad de procesamiento en paralelo 42C que funciona como sombreador de fragmentos. Por último, la unidad de control 30 puede configurar la memoria intermedia local 44C para contener valores de píxeles del sombreador de posfragmentos para la transferencia entre la unidad de procesamiento en paralelo 42C que funciona como sombreador de fragmentos y la unidad de procesamiento en paralelo 42D que funciona como mezclador de píxeles.

[0055] Tras ejecutar las aplicaciones de procesamiento de gráficos 3D, la unidad de distribución de datos 40 puede asignar un conjunto de datos de vértice original a la unidad de procesamiento en paralelo 42A que funciona como sombreador de vértices. La unidad de procesamiento en paralelo 42A recupera el conjunto de datos de vértice original asignado de la memoria de dispositivo 12 por medio del bus 46. Tras la entrada del conjunto de datos, el gestor de secuencia 36 ejecuta el SDC para registrar una secuencia de los datos de vértice. La unidad de procesamiento en paralelo 42A realiza una operación de sombreado de vértices y envía los datos de vértice del sombreador de posvértices a la memoria intermedia local 44A. Tras la salida del conjunto de datos de la unidad de procesamiento en paralelo 42A, el gestor de secuencia 36 ejecuta la SEB para liberar los datos de vértice en la misma secuencia que la registrada por el SDC. De esta manera, los datos de vértice llegarán a la unidad de procesamiento en paralelo 42B, que funciona como rasterizador triangular, en el mismo orden en que los datos de vértice entraron en la unidad de procesamiento en paralelo 42A, que funciona como sombreador de vértices.

[0056] La unidad de procesamiento en paralelo 42B, que funciona como rasterizador triangular, recibe los datos de vértice del sombreador de posvértices de la memoria intermedia local 44A. En algunos casos, la unidad de procesamiento en paralelo 42B también puede recuperar datos complementarios de la memoria de dispositivo 12 por

medio del bus 46 para realizar una operación de rasterización triangular. La unidad de procesamiento en paralelo 42B realiza a continuación la operación de rasterización triangular y envía los datos de píxel del sombreador de prefragmentos a la memoria intermedia local 44B. En algunos ejemplos, el gestor de secuencia 36 puede ejecutar el SDC tras la entrada de los datos de vértice en la unidad de procesamiento en paralelo 42B, y ejecutar la SEB tras la salida de los datos de píxeles desde la unidad de procesamiento en paralelo 42B para mantener la secuencia de datos. En otros ejemplos, las barreras de secuenciación pueden no ser necesarias y, por lo tanto, no ejecutarse para la unidad de procesamiento en paralelo 42B.

[0057] La unidad de procesamiento en paralelo 42C, que hace funcionar el sombreador de fragmentos, recibe los datos de píxeles del sombreador de prefragmentos desde la memoria intermedia local 44B. Tras la entrada del conjunto de datos, el gestor de secuencia 36 ejecuta el SDC para registrar una secuencia de los datos de píxeles. En algunos casos, la unidad de procesamiento en paralelo 42C también puede recuperar datos complementarios de la memoria de dispositivo 12 por medio del bus 46 a fin de realizar una operación de sombreado de fragmentos. La unidad de procesamiento en paralelo 42C realiza a continuación la operación de sombreado de fragmentos y envía los valores de píxeles del sombreador de posfragmentos a la memoria intermedia local 44C. Tras la salida del conjunto de datos de la unidad de procesamiento en paralelo 42C, el gestor de secuencia 36 ejecuta la SEB para liberar los datos de píxeles en la misma secuencia que la registrada por el SDC. De esta manera, los datos de píxeles llegarán a la unidad de procesamiento en paralelo 42D, que funciona como mezclador de píxeles, en el mismo orden en que los datos de píxeles entraron en la unidad de procesamiento en paralelo 42C, que funciona como sombreador de fragmentos.

[0058] La unidad de procesamiento en paralelo 42D, que funciona como mezclador de píxeles, recibe los valores de píxeles del sombreador de posfragmentos desde la memoria intermedia local 44C. La unidad de procesamiento en paralelo 44D realiza a continuación una operación de mezcla de píxeles y almacena el conjunto de datos postcanalización en la memoria de dispositivo 12 por medio del bus 46. En algunos ejemplos, el gestor de secuencia 36 puede ejecutar el SDC tras la entrada de los datos de píxeles en la unidad de procesamiento en paralelo 42D, y ejecutar la SEB tras la salida de los datos de imagen desde la unidad de procesamiento en paralelo 42D para mantener la secuencia de datos. En otros ejemplos, las barreras de secuenciación pueden no ser necesarias y, por lo tanto, no ejecutarse para la unidad de procesamiento en paralelo 42D. El ejemplo descrito anteriormente de una aplicación de procesamiento de gráficos 3D es meramente ejemplar, y las técnicas divulgadas se pueden usar para ejecutar una variedad de aplicaciones basadas en canalización en la GPGPU 6.

[0059] La FIG. 4 es un diagrama de flujo que ilustra un ejemplo de operación de la GPGPU 6 que incluye memorias intermedias locales 44 conectadas a unidades de procesamiento en paralelo 42 para transferir datos entre las unidades de procesamiento en paralelo como fases de la canalización de procesamiento 10. La operación ilustrada se describe con referencia a la GPGPU 6 de la FIG. 3.

[0060] La unidad de control 30 de la GPGPU 6 configura las unidades de procesamiento en paralelo 42 para que funcionen como fases de la canalización de procesamiento 10 (50). Por ejemplo, la unidad de control 30 puede configurar las unidades de procesamiento en paralelo 42 para que funcionen como una fase de una canalización de procesamiento de gráficos 3D. En ese ejemplo, la unidad de control 30 puede configurar la unidad de procesamiento en paralelo 42A para que funcione como un sombreador de vértices, la unidad de procesamiento en paralelo 42B para que funcione como un rasterizador triangular, la unidad de procesamiento en paralelo 42C para que funcione como un sombreador de fragmentos, y la unidad de procesamiento en paralelo 42D para que funcione como un mezclador de píxeles.

[0061] La unidad de control 30 también configura las memorias intermedias locales 44 para que contengan datos para la transferencia entre las unidades de procesamiento en paralelo 42, transformando de hecho las unidades de procesamiento en paralelo 42 en la canalización de procesamiento 10 (52). Las memorias intermedias locales 44 pueden incluir mecanismos de control de flujo de datos basados en hardware para permitir la transferencia de datos entre unidades de procesamiento en paralelo 42. Por ejemplo, las memorias intermedias locales 44 pueden comprender memorias intermedias FIFO, LIFO o indexadas basadas en hardware. Las memorias intermedias locales 44 pueden estar directamente conectadas entre al menos dos de las unidades de procesamiento en paralelo 42. Por ejemplo, en el caso de una canalización de procesamiento de gráficos 3D, la memoria intermedia local 44A puede estar directamente conectada entre la unidad de procesamiento en paralelo 42A que funciona como sombreador de vértices y la unidad de procesamiento en paralelo 42B que funciona como rasterizador triangular, y configurada para contener datos de vértice del sombreador de posvértices. La memoria intermedia local 44B puede estar directamente conectada entre la unidad de procesamiento en paralelo 32B que funciona como rasterizador triangular y la unidad de procesamiento en paralelo 42C que funciona como sombreador de fragmentos, y configurada para contener datos de píxeles del sombreador de prefragmentos. Por último, la memoria intermedia local 44C puede estar directamente conectada entre la unidad de procesamiento en paralelo 42C que funciona como sombreador de fragmentos y la unidad de procesamiento en paralelo 42D que funciona como mezclador de píxeles, y configurada para contener valores de píxeles del sombreador de posfragmentos.

[0062] Además, el gestor de memoria intermedia 34 de la unidad de control 30 puede determinar una anchura requerida para cada una de las memorias intermedias locales 44 para contener los datos facilitados por una anterior de las unidades de procesamiento en paralelo 42 (54). El gestor de memoria intermedia 34 puede conocer el tipo de

datos facilitado por cada una de las unidades de procesamiento en paralelo 42 y, por lo tanto, conocer las anchuras requeridas por cada una de las memorias intermedias locales 44 para contener los datos. El gestor de memoria intermedia 34 puede configurar a continuación cada una de las memorias intermedias locales 44 para que tengan la anchura determinada (56). En algunos casos, las memorias intermedias locales 44 pueden estar basadas en hardware, pero incluir un conjunto de aspectos configurables. Por ejemplo, las memorias intermedias locales 44 pueden ser configurables para intercambiar profundidad por anchura.

[0063] Por ejemplo, el gestor de memoria intermedia 34 puede saber que la unidad de procesamiento en paralelo 42A que funciona como sombreador de vértices facilita datos de vértice del sombreador de posvértices, y configurar la memoria intermedia local 44A para que tenga la anchura requerida para contener los datos de vértice del sombreador de posvértices. El gestor de memoria intermedia 34 también puede saber que la unidad de procesamiento en paralelo 42B que funciona como rasterizador triangular facilita datos de píxeles del sombreador de prefragmentos, y configurar la memoria intermedia local 44B para que tenga la anchura requerida para contener los datos de píxeles del sombreador de prefragmentos. Además, el gestor de memoria intermedia 34 puede saber que la unidad de procesamiento en paralelo 42C que funciona como sombreador de fragmentos facilita valores de píxeles del sombreador de posfragmentos, y configurar la memoria intermedia local 44C para que tenga la anchura requerida para almacenar los valores de píxeles de sombreador de posfragmentos.

[0064] Una vez que las unidades de procesamiento en paralelo 42 y las memorias intermedias locales 44 están configuradas para implementar la canalización de procesamiento 10 dentro de la GPGPU 6, las unidades de procesamiento en paralelo 42 pueden transferir datos entre sí por medio de memorias intermedias locales 44 (58). Más específicamente, la unidad de control 30 puede configurar una o más unidades de procesamiento en paralelo 42 para enviar datos a las memorias intermedias locales 44, y configurar una o más de las unidades de procesamiento en paralelo 44 para recibir datos de las memorias intermedias locales 44. Por ejemplo, la unidad de control 30 puede configurar las unidades de procesamiento en paralelo 42A, 42B y 42C para que envíen datos a las memorias intermedias locales 44A, 44B y 44C, respectivamente. La unidad de control 30 también puede configurar las unidades de procesamiento en paralelo 42B, 42C y 42D para que reciban datos desde las memorias intermedias locales 44A, 44B y 44C, respectivamente.

[0065] La FIG. 5 es un diagrama de flujo que ilustra un ejemplo de la operación de mantener una secuencia de datos dentro de la canalización de procesamiento implementada por las unidades de procesamiento en paralelo 42 y las memorias intermedias locales 44 de la GPGPU 6. La unidad de control 30 de la GPGPU 6 puede mantener la secuencia de datos dentro de la canalización de procesamiento manteniendo la secuencia de datos dentro de una o más de las unidades de procesamiento en paralelo 42. La operación ilustrada se describe con referencia a la unidad de procesamiento en paralelo 42A de la GPGPU 6 de la FIG. 3. Se puede realizar una operación similar para cualquiera de las otras unidades de procesamiento en paralelo 42.

[0066] Como ejemplo, las unidades de procesamiento en paralelo 42 y las memorias intermedias locales 44 pueden estar configuradas para implementar una canalización de procesamiento de gráficos 3D. En ese ejemplo, la unidad de procesamiento en paralelo 42A puede estar configurada para funcionar como un sombreador de vértices, la unidad de procesamiento en paralelo 42B puede estar configurada para funcionar como un rasterizador triangular, la unidad de procesamiento en paralelo 42C puede estar configurada para funcionar como un sombreador de fragmentos, y la unidad de procesamiento en paralelo 42D puede estar configurada para funcionar como un mezclador de píxeles.

[0067] La unidad de procesamiento en paralelo 42A configurada para funcionar como una fase de la canalización de procesamiento 10, por ejemplo, el sombreador de vértice, recibe un conjunto de datos para procesar (62). Por ejemplo, la unidad de distribución de datos 40 puede asignar un conjunto de datos de datos de vértice a la unidad de procesamiento en paralelo 42A, y la unidad de procesamiento en paralelo 42A puede recibir el conjunto de datos asignado desde la memoria de dispositivo 12 por medio del bus 46. Tras la entrada del conjunto de datos en la unidad de procesamiento en paralelo 42A, el gestor de secuencia 36 de la unidad de control 30 ejecuta un contador de determinación de secuencia (SDC) (64). De acuerdo con el SDC, el gestor de secuencia 36 registra una secuencia de subprocesos de datos del conjunto de datos recibido dentro de la memoria local 38 (66). Por ejemplo, el gestor de secuencia 36 puede registrar un índice de cada subproceso de datos del conjunto de datos en el orden en que se reciben los subprocesos de datos desde la memoria de dispositivo 12.

[0068] La unidad de procesamiento en paralelo 42A configurada para funcionar como sombreador de vértices a continuación procesa el conjunto de datos para generar datos de vértice del sombreador de posvértices (68). Como se describe anteriormente, la unidad de procesamiento en paralelo 42A puede estar configurada para enviar los datos de vértice del sombreador de posvértices a la memoria intermedia local 44A a fin de transferir el conjunto de datos a la unidad de procesamiento en paralelo 42B configurada para funcionar como rasterizador triangular. Tras la salida del conjunto de datos de la unidad de procesamiento en paralelo 42A, el gestor de secuencia 36 ejecuta una barrera de imposición de secuencia (SEB) (70). De acuerdo con la SEB, el gestor de secuencia 36 libera los subprocesos de datos del conjunto de datos de la unidad de procesamiento en paralelo 42A en la misma secuencia que la registrada por el SDC (72). Por ejemplo, el gestor de secuencia 36 puede acceder a los índices de subproceso de datos registrados en la memoria local 38, y liberar cada subproceso de datos de acuerdo con el orden en que se ha registrado su índice. De esta manera, los vértices entrarán en la unidad de procesamiento paralela 42B configurada para

funcionar como rasterizador triangular en el mismo orden en que los vértices entraron en la unidad de procesamiento en paralelo 42A configurada para funcionar como sombreador de vértices.

[0069] En uno o más ejemplos, las funciones descritas se pueden implementar en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones u operaciones se pueden almacenar como una o más instrucciones o código en un medio no transitorio legible por ordenador, y ejecutar mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como unos medios de almacenamiento de datos, o medios de comunicación que incluyen cualquier medio que facilita la transferencia de un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder en general a (1) medios de almacenamiento tangibles legibles por ordenador que son no transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser unos medios disponibles cualesquiera a los que se puede acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en la presente divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

[0070] A modo de ejemplo, y no de limitación, dichos medios legibles por ordenador pueden comprender medios no transitorios tales como RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria *flash* o cualquier otro medio que se pueda usar para transportar o almacenar un código de programa deseado en forma de instrucciones o estructuras de datos y al que se pueda acceder mediante un ordenador. Además, cualquier conexión recibe adecuadamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otro origen remoto usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o unas tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas están incluidas en la definición de medio. Sin embargo, debería entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales ni otros medios transitorios, sino que, en cambio, se orientan a medios de almacenamiento tangibles no transitorios. Los discos, como se usan en el presente documento, incluyen el disco compacto (CD), el disco láser, el disco óptico, el disco versátil digital (DVD), el disco flexible y el disco Blu-ray, de los cuales los discos flexibles habitualmente reproducen datos magnéticamente, mientras que los demás discos reproducen datos ópticamente con láseres. Las combinaciones de los anteriores deberían estar también incluidas dentro del alcance de los medios legibles por ordenador.

[0071] Las instrucciones se pueden ejecutar mediante uno o más procesadores, tales como uno o más DSP, microprocesadores de propósito general, ASIC, FPGA u otros circuitos lógicos integrados o discretos equivalentes. En consecuencia, el término "procesador", como se usa en el presente documento, se puede referir a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar dentro de módulos de hardware y/o software dedicados, configurados para codificar y descodificar, o incorporados en un códec combinado. Además, las técnicas se podrían implementar por completo en uno o más circuitos o elementos lógicos.

[0072] Las técnicas de la presente divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, que incluyen un microteléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). Diversos componentes, módulos o unidades se describen en la presente divulgación para resaltar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En cambio, como se describe anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionar mediante una colección de unidades de hardware interoperativas, que incluyen uno o más procesadores como se describe anteriormente, conjuntamente con software y/o firmware adecuados.

[0073] Se han descrito diversos ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Una unidad de procesamiento de gráficos de propósito general, GPGPU, (6) que comprende:

una pluralidad de unidades de procesamiento en paralelo programables (42A - 42D) de la GPGPU configuradas para funcionar selectivamente como fases de una canalización de procesamiento, comprendiendo cada unidad de procesamiento en paralelo programable una agrupación de elementos lógicos configurables;

una pluralidad de memorias intermedias locales (44A - 44C), en la que cada memoria intermedia local está conectada directamente entre al menos dos de las unidades de procesamiento en paralelo;

medios (30) para ejecutar una o más interfaces de programación de aplicaciones, API, para configurar dos o más de la pluralidad de unidades de procesamiento en paralelo programables de la GPGPU para que funcionen como fases de una canalización de procesamiento para enviar datos a las memorias intermedias locales directamente conectadas y recibir datos de las memorias intermedias locales directamente conectadas;

medios (30) para ejecutar la una o más API para configurar cada una de la una o más memorias intermedias locales (44A - 44C) de la GPGPU para que tenga una anchura necesaria para contener datos para transferir entre las unidades de procesamiento en paralelo directamente conectadas; y

medios (36) para mantener una secuencia de datos dentro de la canalización de procesamiento, comprendiendo los medios:

medios para ejecutar un contador de determinación de secuencia tras una entrada de un conjunto de datos en al menos una de las unidades de procesamiento en paralelo para registrar una secuencia de subprocesos de datos del conjunto de datos; y

medios para ejecutar una barrera de imposición de secuencia tras una salida del conjunto de datos de la al menos una de las unidades de procesamiento en paralelo para liberar los subprocesos de datos del conjunto de datos de la unidad de procesamiento en paralelo en la misma secuencia registrada por el contador de determinación de secuencia.

2. La GPGU de la reivindicación 1, en la que la una o más memorias intermedias locales incluyen mecanismos de control de flujo de datos basados en hardware para permitir una transferencia de los datos entre las unidades de procesamiento en paralelo.

3. La GPGPU de la reivindicación 1, que comprende además ejecutar una o más interfaces de programación de aplicaciones, API (32) para determinar una profundidad de cada una de las memorias intermedias locales.

4. La GPGPU de la reivindicación 3, en la que cada una de las memorias intermedias locales es configurable para intercambiar una profundidad por una anchura.

5. Un procedimiento de procesamiento de datos con una unidad de procesamiento de gráficos de propósito general (GPGPU) que comprende una pluralidad de unidades de procesamiento en paralelo programables configuradas para funcionar selectivamente como fases de una canalización de procesamiento y una pluralidad de memorias intermedias locales, en el que cada una de la pluralidad de unidades de procesamiento en paralelo comprende una agrupación de elementos lógicos configurables, en el que cada memoria intermedia local está conectada directamente entre al menos dos de las unidades de procesamiento en paralelo; comprendiendo el procedimiento:

configurar dos o más de la pluralidad de unidades de procesamiento en paralelo programables de la GPGPU para que funcionen como fases de una canalización de procesamiento para enviar datos a las memorias intermedias locales directamente conectadas y recibir datos desde las memorias intermedias locales directamente conectadas;

configurar una o más de las memorias intermedias locales de la GPGPU para que tengan una anchura necesaria para contener datos para una transferencia entre las unidades de procesamiento en paralelo directamente conectadas; y

mantener una secuencia de datos dentro de la canalización de procesamiento ejecutando un contador de determinación de secuencia tras una entrada de un conjunto de datos en al menos una de las unidades de procesamiento en paralelo para registrar una secuencia de subprocesos de datos del conjunto de datos, y ejecutando una barrera de imposición de secuencia tras una salida del conjunto de datos desde la al menos una de las unidades de procesamiento en paralelo para liberar los subprocesos de datos del conjunto de

datos desde la unidad de procesamiento en paralelo en la misma secuencia registrada por el contador de determinación de secuencia.

6. El procedimiento de la reivindicación 5, en el que configurar dos o más unidades de procesamiento en paralelo comprende al menos uno de:

(i) configurar una de las unidades de procesamiento en paralelo para que funcione como una primera fase de la canalización de procesamiento y recuperar un conjunto de datos original de una memoria de dispositivo;

(ii) configurar una de las unidades de procesamiento en paralelo para que funcione como una fase final de la canalización de procesamiento y almacenar un conjunto de datos de canalización procesados en una memoria de dispositivo; y/o

(iii) configurar al menos una de las unidades de procesamiento en paralelo para que funcione como una fase intermedia de la canalización de procesamiento, recibir un conjunto de datos de una anterior de las unidades de procesamiento en paralelo en la canalización de procesamiento por medio de una de las memorias intermedias locales, y enviar el conjunto de datos a una posterior de las unidades de procesamiento en paralelo en la canalización de procesamiento por medio de otra de las memorias intermedias locales.

7. El procedimiento de la reivindicación 5, en el que configurar al menos una de las unidades de procesamiento en paralelo comprende configurar la al menos una de las unidades de procesamiento en paralelo para recuperar datos complementarios de una memoria de dispositivo para procesar el conjunto de datos.

8. Un medio legible por ordenador que comprende instrucciones para procesar datos con una unidad de procesamiento de gráficos de propósito general (GPGPU) que, cuando se ejecuta, hace que un procesador programable lleve a cabo el procedimiento de cualquiera de las reivindicaciones 5 a 7.

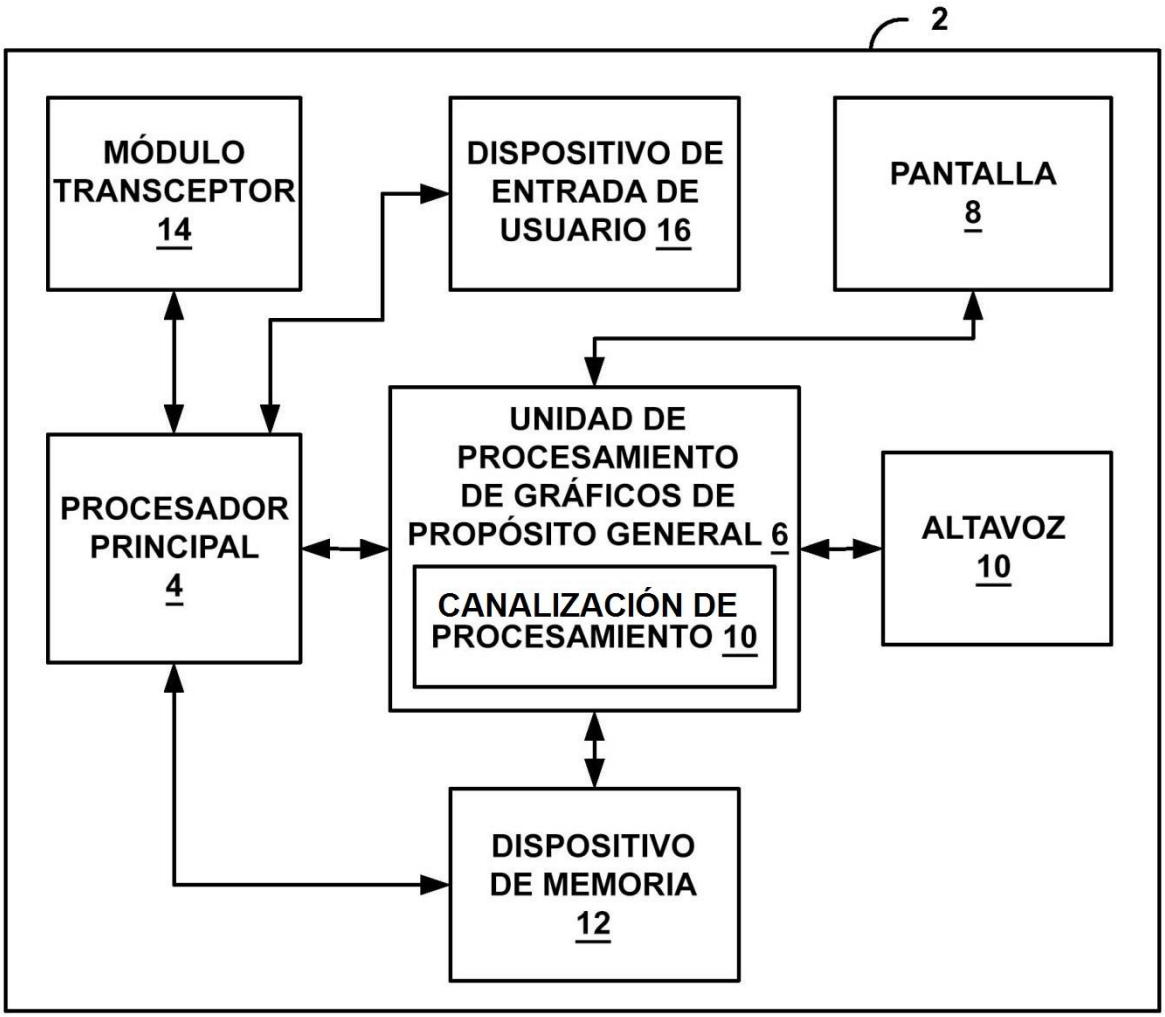
2
MÓDULO TRANSCEPTOR 14
DISPOSITIVO DE ENTRADA DE USUARIO 16
PANTALLA 8
UNIDAD DE PROCESAMIENTO DE GRÁFICOS DE PROPÓSITO GENERAL 6
CANALIZACIÓN DE PROCESAMIENTO 10
PROCESADOR PRINCIPAL 4
ALTAVOZ 10
DISPOSITIVO DE MEMORIA 12

FIG. 1

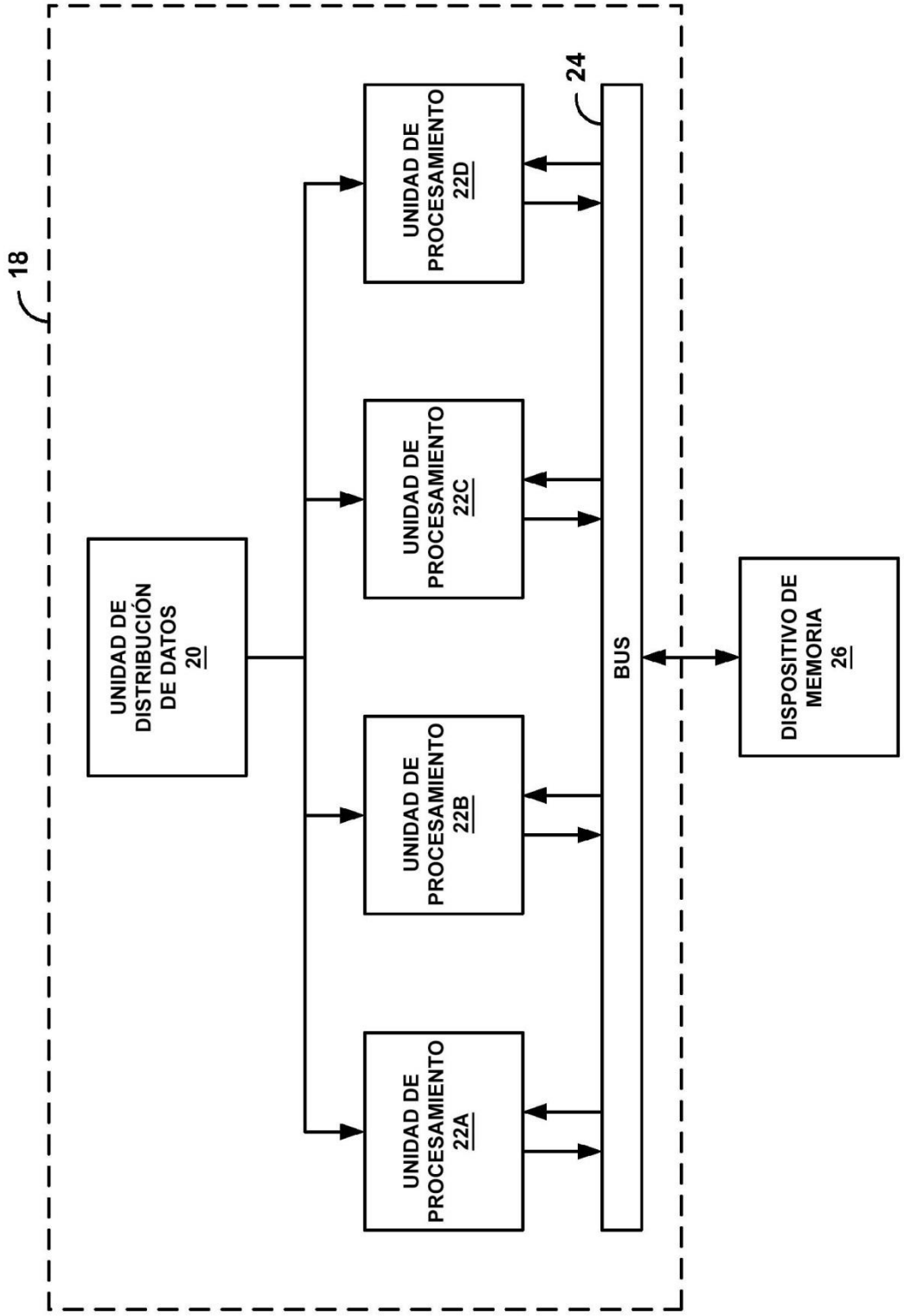
18
24
UNIDAD DE DISTRIBUCIÓN DE DATOS 20
UNIDAD DE PROCESAMIENTO 22A
UNIDAD DE PROCESAMIENTO 22B
UNIDAD DE PROCESAMIENTO 22C
UNIDAD DE PROCESAMIENTO 22D
BUS
DISPOSITIVO DE MEMORIA 26

FIG. 2

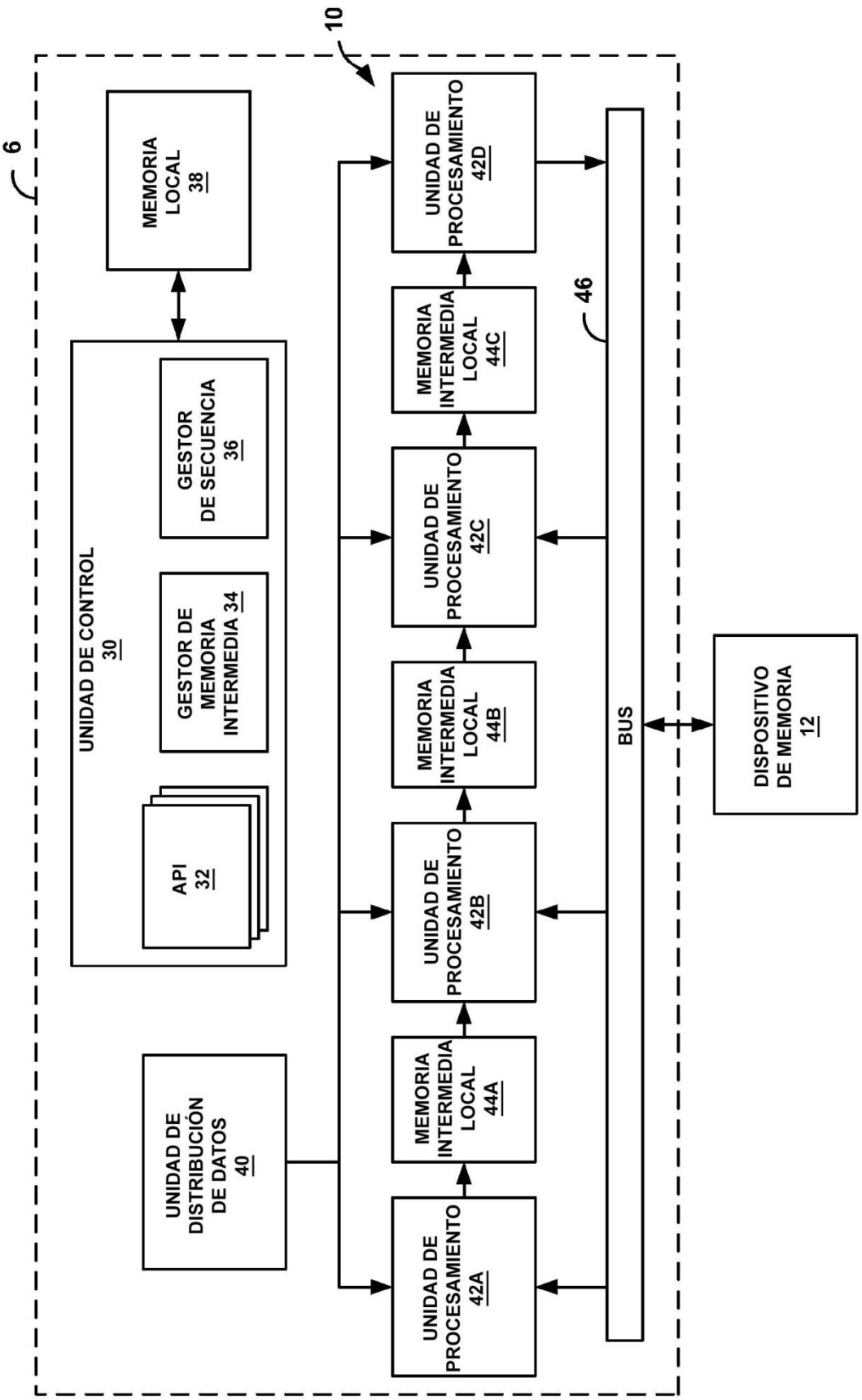
10
6
MEMORIA LOCAL 38
UNIDAD DE CONTROL 30
GESTOR DE SECUENCIA 36
GESTOR DE MEMORIA INTERMEDIA 34
API 32
UNIDAD DE DISTRIBUCIÓN DE DATOS 40
UNIDAD DE PROCESAMIENTO 42A
MEMORIA INTERMEDIA LOCAL 44A
UNIDAD DE PROCESAMIENTO 42B
MEMORIA INTERMEDIA LOCAL 44B
UNIDAD DE PROCESAMIENTO 42C
MEMORIA INTERMEDIA LOCAL 44C
UNIDAD DE PROCESAMIENTO 42D
46
BUS
DISPOSITIVO DE MEMORIA 12

FIG. 3

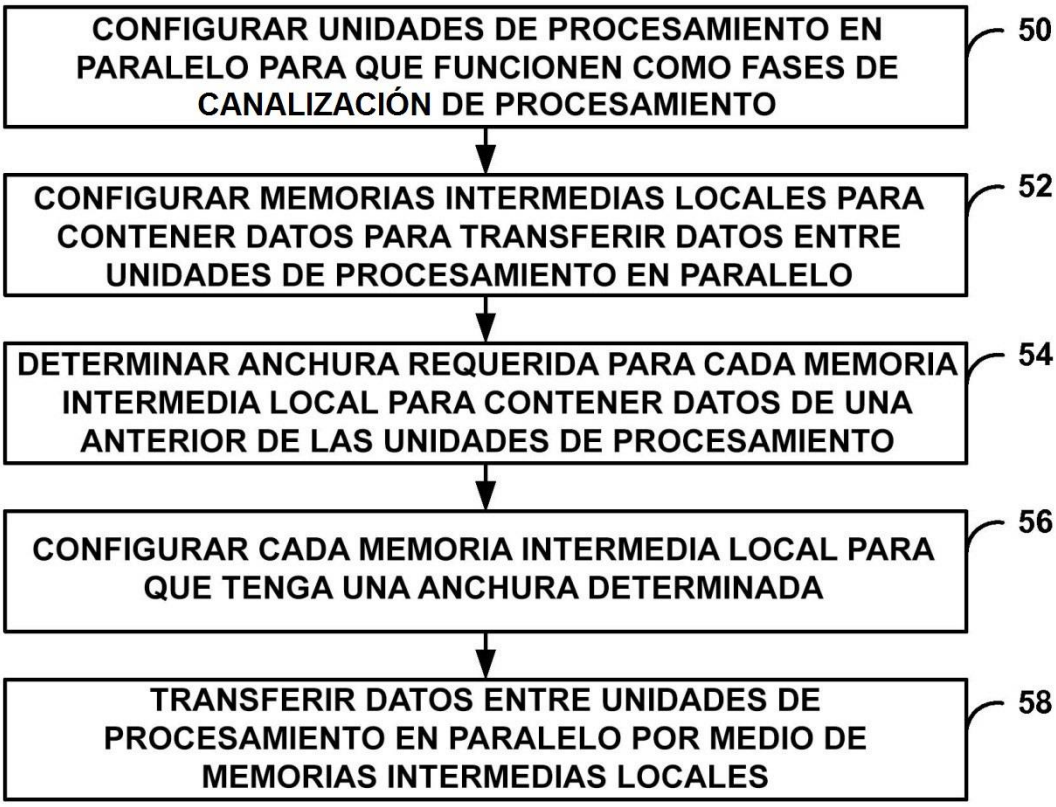
CONFIGURAR UNIDADES DE PROCESAMIENTO EN PARALELO PARA QUE FUNCIONEN COMO FASES DE CANALIZACIÓN DE PROCESAMIENTO
50
CONFIGURAR MEMORIAS INTERMEDIAS LOCALES PARA CONTENER DATOS PARA TRANSFERIR DATOS ENTRE UNIDADES DE PROCESAMIENTO EN PARALELO
52
DETERMINAR ANCHURA REQUERIDA PARA CADA MEMORIA INTERMEDIA LOCAL PARA CONTENER DATOS DE UNA ANTERIOR DE LAS UNIDADES DE PROCESAMIENTO
54
CONFIGURAR CADA MEMORIA INTERMEDIA LOCAL PARA QUE TENGA UNA ANCHURA DETERMINADA
56
TRANSFERIR DATOS ENTRE UNIDADES DE PROCESAMIENTO EN PARALELO POR MEDIO DE MEMORIAS INTERMEDIAS LOCALES
58

FIG. 4

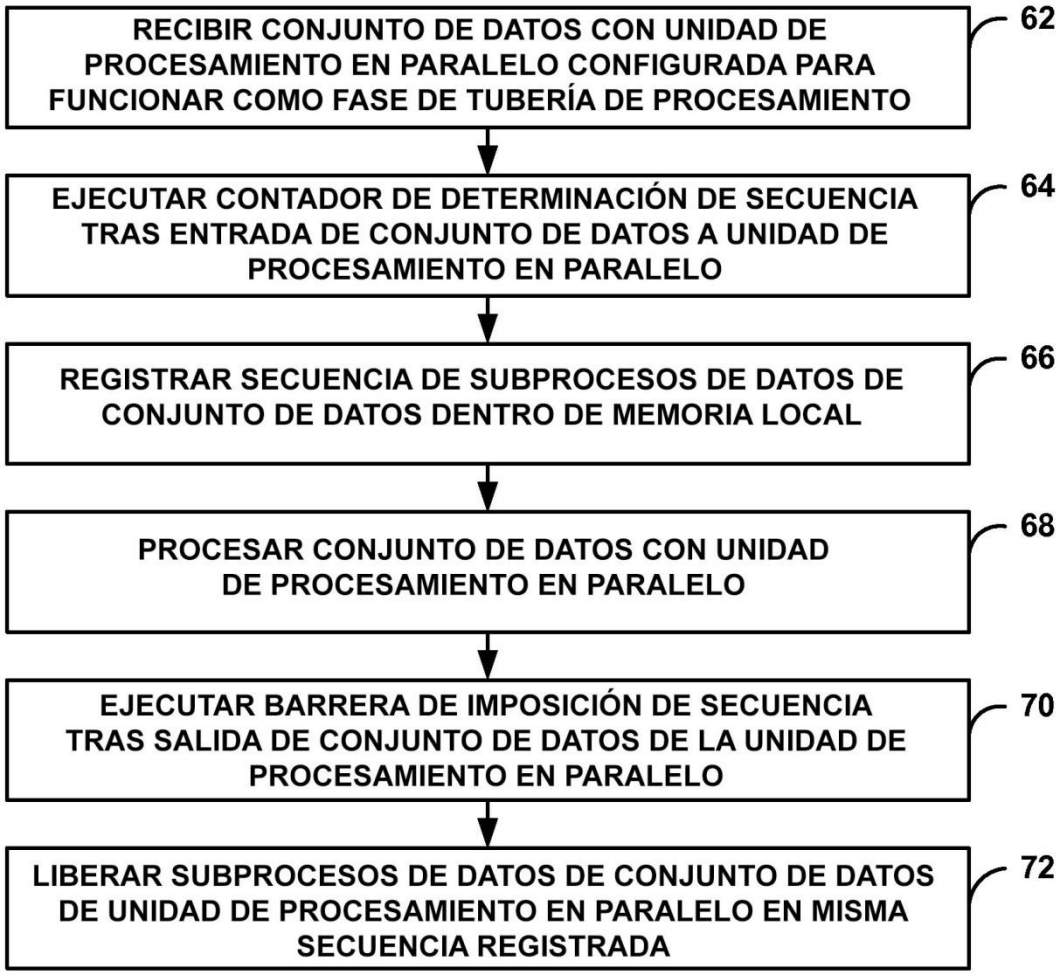
RECIBIR CONJUNTO DE DATOS CON UNIDAD DE PROCESAMIENTO EN PARALELO CONFIGURADA PARA FUNCIONAR COMO FASE DE TUBERÍA DE PROCESAMIENTO
62
EJECUTAR CONTADOR DE DETERMINACIÓN DE SECUENCIA TRAS ENTRADA DE CONJUNTO DE DATOS A UNIDAD DE PROCESAMIENTO EN PARALELO
64
REGISTRAR SECUENCIA DE SUBPROCESOS DE DATOS DE CONJUNTO DE DATOS DENTRO DE MEMORIA LOCAL
66
PROCESAR CONJUNTO DE DATOS CON UNIDAD DE PROCESAMIENTO EN PARALELO
68
EJECUTAR BARRERA DE IMPOSICIÓN DE SECUENCIA TRAS SALIDA DE CONJUNTO DE DATOS DE LA UNIDAD DE PROCESAMIENTO EN PARALELO
70
LIBERAR SUBPROCESOS DE DATOS DE CONJUNTO DE DATOS DE UNIDAD DE PROCESAMIENTO EN PARALELO EN MISMA SECUENCIA REGISTRADA
72

FIG. 5