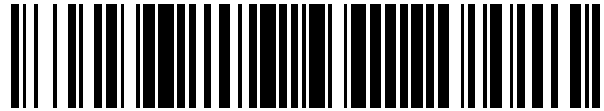


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 691 957**

51 Int. Cl.:

G06F 9/38

(2008.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **04.08.2015 PCT/US2015/043649**

87 Fecha y número de publicación internacional: **25.02.2016 WO16028487**

96 Fecha de presentación y número de la solicitud europea: **04.08.2015 E 15751211 (2)**

97 Fecha y número de publicación de la concesión europea: **25.07.2018 EP 3183646**

54 Título: **Salto de almacenamiento de datos**

30 Prioridad:

19.08.2014 US 201414462932

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

29.11.2018

73 Titular/es:

**QUALCOMM INCORPORATED (100.0%)
5775 Morehouse Drive
San Diego, CA 92121-1714, US**

72 Inventor/es:

**DU, YUN;
CHEN, LIN;
GRUBER, ANDREW EVAN;
ZHANG, CHIHONG y
YU, CHUN**

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 691 957 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

Salto de almacenamiento de datos

5 **CAMPO TÉCNICO**

[0001] Esta divulgación se refiere al almacenamiento de datos, y más particularmente, a técnicas de almacenamiento de datos en registros de uso general (GPR).

10 **ANTECEDENTES**

15 [0002] Las unidades de procesamiento, tales como las unidades de procesamiento de gráficos (GPU), incluyen núcleos de procesamiento que tienen registros de uso general (GPR) que almacenan datos que resultan de la ejecución de una instrucción o de datos necesarios para la ejecución de una instrucción. Por ejemplo, un elemento de procesamiento del núcleo de procesamiento incluye una unidad de lógica aritmética (ALU) que realiza una pluralidad de operaciones aritméticas. El GPR puede almacenar los resultados de las operaciones aritméticas y/o almacenar operandos de las operaciones aritméticas.

20 [0003] El GPR puede ser local al núcleo de procesador de la unidad de procesamiento. En consecuencia, acceder a un GPR puede requerir menos energía que acceder a la memoria caché local o a la memoria de sistema externo. Aunque es más eficiente que la memoria caché local o que la memoria de sistema externo, el almacenamiento de datos en el GPR consume energía. El documento EP 1 119 629 divulga un procesador enrutado que comprende una señal de inhibición de escritura para que los operandos con longitud de vida reducida puedan recuperarse solamente desde una red de reenvío y no tengan que escribirse en el archivo de registro.

25 **RESUMEN**

30 [0004] En general, la divulgación describe técnicas para saltar el acceso a un registro de uso general (GPR) para que los datos a los que no haya que accederse nuevamente no se almacenen en el GPR. Un compilador puede determinar si los datos no son necesarios para ejecutar instrucciones posteriores de un programa. Si los datos no son necesarios para ejecutar instrucciones posteriores, el compilador puede indicar que los datos no son necesarios para ejecutar instrucciones posteriores. Una unidad de procesamiento puede determinar que los datos no son necesarios para ejecutar instrucciones posteriores en base a la indicación y, en respuesta, evitar almacenar los datos en el GPR (es decir, saltar el almacenamiento de los datos en el GPR). De esta manera, el consumo de energía puede reducirse al evitar el acceso innecesario al GPR para almacenar datos que posteriormente no se necesiten.

40 [0005] En un ejemplo, la divulgación describe un procedimiento para procesar valores en un sistema de procesamiento de datos, comprendiendo el procedimiento recibir una pluralidad de sub-instrucciones de una instrucción de un programa, en el que la instrucción define una pluralidad de operaciones y en el que cada una de las sub-instrucciones define una operación constitutiva de la pluralidad de operaciones, recibir una indicación de al menos un valor intermedio generado por la ejecución de una de las sub-instrucciones que indica si el valor intermedio debe almacenarse en un registro de uso general (GPR), determinar si el valor intermedio debe almacenarse en el GPR en base al menos en parte a la indicación recibida y evitar el almacenamiento del valor intermedio en el GPR si se determina que el valor intermedio no debe almacenarse en el GPR.

50 [0006] En un ejemplo, la divulgación describe un dispositivo para procesar datos, comprendiendo el dispositivo una unidad de procesamiento que comprende un núcleo, comprendiendo el núcleo una memoria de instrucciones, un registro de uso general (GPR) y al menos un elemento de procesamiento, comprendiendo el al menos un elemento de procesamiento un controlador y una estructura de canal que tiene una pluralidad de etapas de ejecución, en el que el controlador está configurado para recibir una pluralidad de sub-instrucciones de una instrucción de un programa desde la memoria de instrucciones del núcleo, en el que la instrucción define una pluralidad de operaciones, en el que cada una de las sub-instrucciones define una operación constitutiva de la pluralidad de operaciones, y en el que las sub-instrucciones se ejecutan por la pluralidad de etapas de ejecución de la estructura de canal, reciben una indicación de al menos un valor intermedio generado por la ejecución de una de las sub-instrucciones en una de la pluralidad de etapas de ejecución que indica si el valor intermedio debe almacenarse en el GPR, determinan si el valor intermedio debe almacenarse en el GPR en base al menos en parte a la indicación recibida y evitan el almacenamiento del valor intermedio en el GPR si se determina que el valor intermedio no debe almacenarse en el GPR.

60 [0007] En un ejemplo, la divulgación describe un medio de almacenamiento legible por ordenador que tiene instrucciones almacenadas en el mismo que, cuando se ejecuta por una o más unidades de procesamiento, causa que la una o más unidades de procesamiento reciban una pluralidad de sub-instrucciones de una instrucción de un programa, en el que la instrucción define una pluralidad de operaciones y en el que cada una de las sub-instrucciones define una operación constitutiva de la pluralidad de operaciones, reciban una indicación de al menos un valor intermedio generado por la ejecución de una de las sub-instrucciones que indique si el valor intermedio

debe almacenarse en un registro de uso general (GPR), determinen si el valor intermedio debe almacenarse en el GPR en base al menos en parte a la indicación recibida y eviten el almacenamiento del valor intermedio en el GPR si se determina que el valor intermedio no debe almacenarse en el GPR.

5 **[0008]** En un ejemplo, la divulgación describe un dispositivo para procesar datos, comprendiendo el dispositivo medios para recibir una pluralidad de sub-instrucciones de una instrucción de un programa, en el que la instrucción define una pluralidad de operaciones y en el que cada una de las sub-instrucciones define una operación constitutiva de la pluralidad de operaciones, medios para recibir una indicación de al menos un valor intermedio generado por la ejecución de una de las sub-instrucciones que indique si el valor intermedio debe almacenarse en un registro de uso
10 general (GPR), medios para determinar si el valor intermedio debe almacenarse en el GPR en base al menos en parte a la indicación recibida y medios para evitar el almacenamiento del valor intermedio en el GPR si se determina que el valor intermedio no debe almacenarse en el GPR.

15 **[0009]** En un ejemplo, la divulgación describe un procedimiento de compilación, comprendiendo el procedimiento determinar una pluralidad de sub-instrucciones a partir de una instrucción de un programa, en el que la instrucción define una pluralidad de operaciones y en el que cada una de las sub-instrucciones define una operación constitutiva de la pluralidad de operaciones, determinar un último uso para al menos un valor intermedio que se generará mediante la ejecución de una de las sub-instrucciones, en el que determinar el último uso comprende determinar que el valor intermedio ya no es necesario por la pluralidad de sub-instrucciones para generar un valor final a partir de la pluralidad de operaciones definidas por la instrucción, e indicar el último uso del al menos un valor intermedio en base a la determinación.

20 **[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, de los dibujos y de las reivindicaciones.
25

BREVE DESCRIPCIÓN DE LOS DIBUJOS

30 **[0011]**
La FIG. 1 es un diagrama de bloques que ilustra un dispositivo de ejemplo para procesar datos de acuerdo con una o más técnicas de ejemplo descritas en esta divulgación.
35 La FIG. 2 es un diagrama de bloques que ilustra los componentes del dispositivo ilustrado en la FIG. 1 en mayor detalle.
La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de un núcleo de sombreado de una unidad de procesamiento de gráficos (GPU) de la FIG. 2 en mayor detalle.
40 La FIG. 4 es un diagrama de bloques que ilustra un ejemplo de un elemento de procesamiento de un núcleo de sombreado de la FIG. 3 en mayor detalle.
La FIG. 5 es un diagrama de flujo que ilustra una técnica de ejemplo de valores de procesamiento.
45 La FIG. 6 es un diagrama de flujo que ilustra una técnica de ejemplo de compilación.

DESCRIPCIÓN DETALLADA

50 **[0012]** Las unidades de procesamiento paralelo, tales como la unidad de procesamiento de gráficos (GPU), que están configuradas para realizar muchas operaciones en paralelo (por ejemplo, al mismo tiempo), incluyen uno o más núcleos de procesador (por ejemplo, núcleos de sombreado para una GPU) que ejecutan instrucciones de uno o más programas. Para facilitar la descripción, las técnicas descritas en la divulgación se describen con respecto a una GPU o a una GPU de uso general (GPGPU). Sin embargo, las técnicas descritas en esta divulgación se pueden extender a unidades de procesamiento paralelo que no sean necesariamente GPU ni GPGPU, así como unidades de procesamiento no paralelo (por ejemplo, las que no estén configuradas específicamente para el procesamiento paralelo).
55

60 **[0013]** La GPU puede diseñarse con una estructura de una sola instrucción, de datos múltiples (SIMD). En la estructura SIMD, un núcleo de sombreado incluye una pluralidad de elementos de procesamiento SIMD, donde cada elemento de procesamiento SIMD ejecuta instrucciones del mismo programa, pero en datos diferentes. Una instrucción particular que se ejecuta en un elemento de procesamiento SIMD en particular se denomina subproceso. Se puede considerar que cada elemento de procesamiento SIMD ejecuta un subproceso diferente porque los datos para un subproceso dado pueden ser diferentes; sin embargo, el subproceso que se ejecuta en un elemento de procesamiento es la misma instrucción, del mismo programa, que la instrucción que se ejecuta en los otros elementos de procesamiento. De esta manera, la estructura SIMD permite que la GPU realice muchas tareas en paralelo (por ejemplo, al mismo tiempo).
65

[0014] Como ejemplo, para el procesamiento de vértices, la GPU puede realizar las mismas tareas para cada vértice, pero los datos de vértice para cada vértice pueden ser diferentes (por ejemplo, diferentes coordenadas, diferentes valores de color, etc.). La estructura SIMD permite que la GPU procese muchos vértices en paralelo ejecutando la misma instancia de un sombreador de vértices en diferentes elementos de procesamiento SIMD. En este ejemplo, cada elemento de procesamiento SIMD puede ejecutar un subproceso del sombreador de vértices, donde cada subproceso es una instancia del sombreador de vértices.

[0015] Un elemento de procesamiento SIMD incluye una o más unidades lógicas aritméticas (ALU), y el elemento de procesamiento SIMD almacena el resultado en un GPR. Un GPR es un registro dedicado para almacenar los datos resultantes de uno o más elementos de procesamiento SIMD. El GPR solamente puede ser accesible por el núcleo de procesamiento que incluye los elementos de procesamiento, y no por otras unidades de la GPU, tales como unidades de hardware de un canal de gráficos. En este sentido, un GPR no debería confundirse con el caché local o con la memoria local de la GPU, que está disponible para todas las unidades de la GPU.

[0016] Las ALU realizan funciones aritméticas tales como sumar, restar, multiplicar, dividir, etc., en base a la operación aritmética definida en el subproceso. En algunos ejemplos, para realizar las operaciones aritméticas definidas en el subproceso, las ALU producen valores intermedios que pueden usarse para realizar las operaciones aritméticas. Por ejemplo, el subproceso puede definir que se deben sumar tres valores. En este caso, una ALU puede añadir los dos primeros valores para generar un valor intermedio y agregar el valor intermedio al tercer valor. La ALU puede almacenar entonces el valor final resultante en el GPR.

[0017] En algunas técnicas, las ALU siempre almacenan los valores intermedios en el GPR y entonces sobrescriben los valores intermedios en el GPR con el valor final. Sin embargo, dicho almacenamiento de los valores intermedios en el GPR no siempre es necesario. Al almacenar innecesariamente valores intermedios en el GPR, la GPU consume energía innecesariamente.

[0018] Como se describe con más detalle a continuación, las técnicas descritas en esta divulgación determinan si ya no se necesitan valores intermedios para completar la operación aritmética definida en el subproceso. Si los valores intermedios ya no son necesarios, las técnicas se indican como tales, y la GPU puede evitar de ese modo el almacenamiento (por ejemplo, saltar el almacenamiento) de los valores intermedios en el GPR en base a dicha indicación. De esta manera, las técnicas descritas en esta divulgación reducen el número de veces que se necesita acceder al GPR, ahorrando de ese modo energía.

[0019] Por ejemplo, un procesador, tal como una unidad de procesamiento central (CPU), compila el programa que se va a ejecutar en la GPU a través de un compilador. Como parte de la compilación, el compilador puede dividir una operación aritmética definida por un subproceso en operaciones constituyentes. El compilador puede escanear a través de las operaciones constituyentes para determinar si ya no es necesario que alguno de los valores intermedios, generados por las operaciones constituyentes, complete la operación aritmética definida por el subproceso. Para cualquiera de los valores intermedios que se determine que ya no son necesarios, el compilador puede indicar a la GPU que estos valores intermedios ya no son necesarios. En respuesta, la GPU puede saltar el almacenamiento de valores intermedios en el GPR que se indican como que ya no se necesitan para la operación aritmética definida por el subproceso.

[0020] La FIG. 1 es un diagrama de bloques que ilustra un dispositivo de ejemplo para procesar datos de acuerdo con una o más técnicas de ejemplo descritas en esta divulgación. La FIG. 1 ilustra el dispositivo 10, cuyos ejemplos incluyen, pero no se limitan a, dispositivos de video tales como reproductores de medios, decodificadores, teléfonos inalámbricos tales como teléfonos móviles, asistentes digitales personales (PDA), ordenadores de escritorio, ordenadores portátiles, consolas de videojuegos, unidades de videoconferencia, dispositivos informáticos de tablets y similares.

[0021] En el ejemplo de la FIG. 1, el dispositivo 10 incluye el procesador 12, la unidad de procesamiento de gráficos (GPU) 14 y la memoria de sistema 16. En algunos ejemplos, tales como ejemplos en los que el dispositivo 10 sea un dispositivo móvil, el procesador 12 y la GPU 14 pueden formarse como un circuito integrado (IC). Por ejemplo, el IC puede considerarse como un chip de procesamiento dentro de un paquete de chips. En algunos ejemplos, el procesador 12 y la GPU 14 pueden estar alojados en diferentes circuitos integrados (es decir, diferentes paquetes de chips) tales como ejemplos en los que el dispositivo 10 sea un ordenador de escritorio o portátil. Sin embargo, puede ser posible que el procesador 12 y la GPU 14 estén alojados en diferentes circuitos integrados en ejemplos en los que el dispositivo 10 sea un dispositivo móvil.

[0022] Los ejemplos del procesador 12 y del GPU 14 incluyen, pero no se limitan a, uno o más procesadores de señales digitales (DSP), microprocesadores de uso general, circuitos integrados específicos de la aplicación (ASIC), matrices lógicas programables de campo (FPGA) u otra circuitería lógica integrada equivalente o lógica discreta. El procesador 12 puede ser la unidad de procesamiento central (CPU) del dispositivo 10. En algunos ejemplos, la GPU 14 puede ser hardware especializado que incluya circuitería lógica integrada y/o discreta que proporcione la GPU 14 con capacidades de procesamiento paralelo masivas adecuadas para el procesamiento de gráficos. En algunos

casos, la GPU 14 también puede incluir capacidades de procesamiento de uso general, y se puede referir a una GPU de uso general (GPGPU) cuando se implementen tareas de procesamiento de uso general (es decir, tareas no relacionadas con gráficos).

5 **[0023]** Para los propósitos de ilustración, las técnicas descritas en esta divulgación se describen con la GPU 14. Sin embargo, las técnicas descritas en esta divulgación no están limitadas a esto. Las técnicas descritas en esta divulgación pueden extenderse a otros tipos de unidades de procesamiento en paralelo (por ejemplo, unidades de procesamiento que proporcionen capacidades de procesamiento paralelo masivo, incluso si no es para el procesamiento de gráficos). Además, las técnicas descritas en esta divulgación pueden extenderse a unidades de procesamiento no configuradas específicamente para el procesamiento paralelo.

15 **[0024]** El procesador 12 puede ejecutar diversos tipos de aplicaciones. Los ejemplos de las aplicaciones incluyen navegadores web, aplicaciones de correo electrónico, hojas de cálculo, videojuegos u otras aplicaciones que generan objetos visibles para su visualización. La memoria de sistema 16 puede almacenar instrucciones para la ejecución de una o más aplicaciones. La ejecución de una aplicación en el procesador 12 causa que el procesador 12 produzca datos gráficos para el contenido de la imagen que se vaya a visualizar. El procesador 12 puede transmitir datos gráficos del contenido de la imagen a la GPU 14 para un procesamiento posterior.

20 **[0025]** En algunos ejemplos no relacionados con gráficos, el procesador 12 puede generar datos que sean adecuados para procesarse por la GPU 14. Dichos datos no necesitan necesariamente ser necesarios para los propósitos de gráficos o de visualización. Por ejemplo, el procesador 12 puede emitir datos sobre qué operaciones de matriz deben realizarse en la GPU 14, y la GPU 14 puede a su vez realizar las operaciones de la matriz.

25 **[0026]** En general, el procesador 12 puede descargar tareas de procesamiento a la GPU 14, tales como las tareas que requieran operaciones paralelas masivas. Como ejemplo, el procesamiento de gráficos requiere operaciones paralelas masivas, y el procesador 12 puede descargar dichas tareas de procesamiento de gráficos a la GPU 14. Sin embargo, otras operaciones tales como las operaciones de matriz también pueden beneficiarse de las capacidades de procesamiento paralelo de la GPU 14. En estos ejemplos, el procesador 12 puede aprovechar las capacidades de procesamiento paralelo de la GPU 14 para causar que la GPU 14 realice operaciones no relacionadas con los gráficos.

35 **[0027]** El procesador 12 puede comunicarse con la GPU 14 de acuerdo con una interfaz de procesamiento de aplicaciones (API) particular. Entre los ejemplos de dichas API se incluyen DirectX® API de Microsoft®, OpenGL® u OpenGL ES® del grupo Khronos y OpenCL™; sin embargo, los aspectos de esta divulgación no están limitados a las API de DirectX, OpenGL o OpenCL, y pueden extenderse a otros tipos de API. Además, las técnicas descritas en esta divulgación no requieren funcionar de acuerdo con una API, y el procesador 12 y la GPU 14 pueden utilizar cualquier técnica de comunicación.

40 **[0028]** La memoria de sistema 16 puede ser la memoria para el dispositivo 10. La memoria de sistema 16 puede comprender uno o más medios de almacenamiento legibles por ordenador. Los ejemplos de memoria de sistema 16 incluyen, pero no están limitados a, una memoria de acceso aleatorio (RAM), una memoria de solo lectura programable y borrable eléctricamente (EEPROM), una memoria flash u otro medio que pueda usarse para transportar o almacenar el código de programa deseado en forma de instrucciones y/o estructuras de datos y al que se pueda acceder por un ordenador o por un procesador.

45 **[0029]** En algunos aspectos, la memoria de sistema 16 puede incluir instrucciones que causen que el procesador 12 y/o la GPU 14 realicen las funciones atribuidas en esta divulgación al procesador 12 y a la GPU 14. En consecuencia, la memoria de sistema 16 puede ser un medio de almacenamiento legible por ordenador que tenga instrucciones almacenadas en el mismo que, cuando se ejecute, cause que uno o más procesadores (por ejemplo, el procesador 12 y la GPU 14) realicen diversas funciones.

55 **[0030]** Por ejemplo, como se describe con más detalle en otra parte de esta divulgación, un compilador que se ejecute en el procesador 12 puede determinar si los valores intermedios generados por instrucciones de un programa de sombreado que se ejecuta en la GPU 14 no se almacenan en un registro de uso general (GPR) de un núcleo de sombreado de GPU 14 para que el elemento de procesamiento no necesite innecesariamente almacenar valores en el GPR. La memoria de sistema 16 puede almacenar el código de compilador y el código del programa de sombreado que se ejecuta en la GPU 14. Como otro ejemplo, una aplicación que se ejecuta en el procesador 12 causa que la GPU 14 ejecute el programa de sombreado, y la memoria de sistema 16 puede almacenar las instrucciones de la aplicación que se ejecute en el procesador 12.

60 **[0031]** En algunos ejemplos, la memoria de sistema 16 puede ser un medio de almacenamiento no transitorio. La expresión «no transitorio» puede indicar que el medio de almacenamiento no está incorporado ni en una onda portadora ni en una señal propagada. Sin embargo, el término "no transitorio" no debería interpretarse en el sentido de que la memoria de sistema 16 es no móvil o que sus contenidos son estáticos. Como ejemplo, la memoria de sistema 16 puede retirarse del dispositivo 10 y trasladarse a otro dispositivo. Como otro ejemplo, la memoria, sustancialmente similar a la memoria de sistema 16, puede insertarse en el dispositivo 10. En ciertos ejemplos, un

medio de almacenamiento no transitorio puede almacenar datos que, con el tiempo, pueden cambiar (por ejemplo, en una RAM).

5 **[0032]** El dispositivo 10 también puede incluir la pantalla 18, la interfaz de usuario 20 y el módulo transceptor 22. El dispositivo 10 puede incluir módulos o unidades adicionales que no se muestran en la FIG. 1 para los propósitos de claridad. Por ejemplo, el dispositivo 10 puede incluir un altavoz y un micrófono, ninguno de los cuales se muestra en la FIG. 1, para efectuar comunicaciones telefónicas en ejemplos donde el dispositivo 10 es un teléfono inalámbrico móvil. Además, los diversos módulos y unidades mostrados en el dispositivo 10 pueden no ser necesarios en cada ejemplo del dispositivo 10. Por ejemplo, la interfaz de usuario 20 y la pantalla 18 pueden ser externas al dispositivo 10 en ejemplos donde el dispositivo 10 sea un ordenador de escritorio. Como otro ejemplo, la interfaz de usuario 20 puede ser parte de la pantalla 18 en ejemplos donde la pantalla 18 sea una pantalla sensible al tacto o sensible a la presencia de un dispositivo móvil.

15 **[0033]** La pantalla 18 puede comprender una pantalla de cristal líquido (LCD), una pantalla de tubo de rayos catódicos (CRT), una pantalla de plasma, una pantalla sensible al tacto, una pantalla sensible a la presencia u otro tipo de dispositivo de visualización. Los ejemplos de la interfaz de usuario 20 incluyen, sin limitarse a, una rueda de desplazamiento, un ratón, un teclado y otros tipos de dispositivos de entrada. La interfaz de usuario 20 también puede ser una pantalla táctil y puede incorporarse como parte de la pantalla 18. El módulo de transceptor 22 puede incluir circuitería para permitir la comunicación inalámbrica o alámbrica entre el dispositivo 10 y otro dispositivo o una red. El módulo de transceptor 22 puede incluir moduladores, demoduladores, amplificadores y otra dicha circuitería para comunicación alámbrica o inalámbrica.

25 **[0034]** Como se describió anteriormente, la GPU 14 proporciona capacidades masivas de procesamiento paralelo. Una forma en que la GPU 14 proporciona dichas capacidades de procesamiento paralelo es a través de una estructura de una sola instrucción, de datos múltiples (SIMD). En la estructura SIMD, la GPU 14 ejecuta una pluralidad de instancias del mismo programa. Por ejemplo, el procesamiento de gráficos y algunos procesos no relacionados con gráficos requieren que se realicen las mismas operaciones, pero con datos diferentes.

30 **[0035]** Por ejemplo, la GPU 14 puede ejecutar programas de sombreado (a los que se hace referencia simplemente como sombreadores) que realicen tareas relacionadas con gráficos o tareas no relacionadas con gráficos. La GPU 14 incluye al menos un núcleo de sombreado, y los programas de sombreado se ejecutan en el núcleo de sombreado. Para mayor brevedad y facilidad de descripción, la GPU 14 se describe como la realización de tareas relacionadas con gráficos, pero puede realizar de manera similar tareas no relacionadas con gráficos. La GPU 14 puede ejecutar múltiples instancias del programa de sombreado para procesar múltiples valores de datos de gráficos a la vez. Un ejemplo de un programa de sombreado es un sombreador de vértices. La GPU 14 puede ejecutar múltiples instancias del sombreador de vértices para procesar múltiples vértices a la vez. En general, las operaciones que se deben realizar para un vértice son las mismas que las operaciones que se deben realizar para otros vértices; sin embargo, los valores de vértice específicos de cada uno de los vértices pueden ser diferentes.

40 **[0036]** Para ejecutar las múltiples instancias de un programa de sombreado, el núcleo de sombreado de la GPU 14 incluye una pluralidad de elementos de procesamiento, y cada elemento de procesamiento puede ejecutar una instrucción de una instancia del programa de sombreado. Por ejemplo, cada elemento de procesamiento puede ejecutar una primera instrucción del programa de sombreado. Aunque la instrucción que ejecutan los elementos de procesamiento sea la misma, los valores sobre los que funciona la instrucción en cada elemento de procesamiento pueden ser diferentes. En este sentido, cada elemento de procesamiento puede considerarse como la ejecución de un subproceso del programa de sombreado, donde el subproceso es una instrucción del programa de sombreado para un dato dado. En consecuencia, cada elemento de procesamiento puede ejecutar un subproceso diferente porque los datos para cada subproceso pueden ser diferentes, pero la instrucción subyacente para cada subproceso puede ser la misma.

50 **[0037]** Como ejemplo básico, una instrucción de un programa de sombreado puede definir una operación de suma de tres valores. En este ejemplo, cada uno de los elementos de procesamiento puede sumar tres valores, pero los valores específicos que se suman por cada elemento de procesamiento pueden ser diferentes. En consecuencia, cada elemento de procesamiento puede ejecutar un subproceso del programa de sombreado, donde el subproceso defina una operación de suma de tres valores. Por lo tanto, la instrucción puede ser la misma (es decir, agregar tres valores), pero el uno o más de los tres valores para cada elemento de procesamiento puede ser diferente. Al ejecutar la instrucción para agregar tres valores en cada uno de los elementos de procesamiento al mismo tiempo, la GPU 14 puede proporcionar el cálculo paralelo de la suma de tres valores, en este ejemplo.

60 **[0038]** En algunos casos, como parte de la realización de las operaciones definidas por una instrucción, el elemento de procesamiento puede producir un valor intermedio. Por ejemplo, la instrucción puede definir una pluralidad de operaciones (por ejemplo, una combinación de sumas, restas, multiplicaciones y/o divisiones). En este caso, cada operación de la pluralidad de operaciones puede ser una operación constituyente. El resultado de una operación constitutiva puede ser un valor intermedio. Por ejemplo, el valor intermedio no es el valor final de la pluralidad de operaciones, sino el resultado de una de las operaciones constituyentes de la pluralidad de operaciones.

- 5 **[0039]** Cada uno de los elementos de procesamiento puede almacenar el valor final resultante de las operaciones realizadas por el elemento de procesamiento en un registro de uso general (GPR) de un núcleo de sombreado. En algunos ejemplos, el almacenamiento de los valores intermedios generados a partir del elemento de procesamiento que realiza una operación constituyente en el GPR puede no ser necesario. Por ejemplo, el elemento de procesamiento puede retroalimentar un valor intermedio para el procesamiento adicional (por ejemplo, el valor intermedio se usa como un valor de entrada para realizar operaciones constituyentes adicionales definidas en la instrucción). Sin embargo, este valor intermedio puede no ser necesario nuevamente y, por lo tanto, puede que no haya necesidad de almacenar el valor intermedio en el GPR.
- 10 **[0040]** Si un valor intermedio se retroalimenta y ya no es necesario, el ciclo de vida del valor intermedio puede considerarse como final (por ejemplo, último uso del valor intermedio). Para dichos valores intermedios (por ejemplo, aquellos cuyo ciclo de vida ha finalizado), la eficiencia de energía puede obtenerse al no almacenar (por ejemplo, saltar el almacenamiento) los valores intermedios en el GPR.
- 15 **[0041]** Por ejemplo, en un sistema de sombreado con la estructura SIMD (por ejemplo, dicha GPU 14 que incluye un núcleo de sombreado en la estructura SIMD), la estructura GPR es normalmente profunda y ancha para una mejor densidad para que el coste por bit sea pequeño. En otras palabras, el GPR de un elemento de procesamiento puede configurarse para almacenar más de un solo valor. Como ejemplo, supongamos que los valores producidos por el elemento de procesamiento son 8 bits. En este ejemplo, el GPR puede almacenar muchos valores de 32 bits (por ejemplo, más ancho porque 32 bits es más de 8 bits y más profundo porque se puede almacenar más de un valor en el GPR).
- 20 **[0042]** Aunque tener GPR profundos y amplios puede ser beneficioso, también puede haber algunos inconvenientes potenciales. Por ejemplo, para un GPR de mayor tamaño, el tiempo necesario para acceder a los valores (por ejemplo, datos) puede ser mayor, causando canales de lectura de datos adicionales en el elemento de procesamiento (por ejemplo, el elemento de procesamiento está inactivo mientras se leen los datos). Además, acceder al GPR requiere que el elemento de procesamiento y, por lo tanto, la GPU 14 consuman energía.
- 25 **[0043]** Por ejemplo, puede haber diversas unidades en las que la GPU 14 puede almacenar datos (por ejemplo, valores). La GPU 14 puede almacenar datos en la memoria de sistema 16 o puede almacenar datos en la memoria local (por ejemplo, caché). El GPR de un núcleo de sombreado es distinto tanto de la memoria de sistema 16 como de la memoria local de la GPU 14. Por ejemplo, la memoria de sistema 16 es accesible por diversos componentes del dispositivo 10, y estos componentes usan la memoria de sistema 16 para almacenar datos. La memoria local de la GPU 14 es accesible por diversos componentes de la GPU 14, y estos componentes usan la memoria local de la GPU 14 para almacenar datos. Sin embargo, un GPR solamente puede ser accesible por los elementos de procesamiento del núcleo de sombreado o por otras unidades del sombreador, y puede que no sea accesible por unidades externas al núcleo de sombreado.
- 30 **[0044]** Como se describió anteriormente, si es el final del ciclo de vida para un valor intermedio, dicho valor intermedio no puede almacenarse en el GPR. Dicha técnica de ejemplo se describió anteriormente con respecto a una estructura SIMD de la GPU 14. Sin embargo, las técnicas descritas en esta divulgación no están limitadas a una estructura SIMD. En general, las técnicas descritas en esta divulgación pueden ser aplicables a ejemplos de un elemento de procesamiento y de un combo GPR, donde los elementos de procesamiento implementan un esquema de ejecución canalizado para las operaciones aritméticas, como se describe con más detalle. Para facilitar la descripción, las técnicas se describen con respecto a una estructura SIMD, y también porque la estructura SIMD tiende a usar un tamaño de GPR más grande.
- 35 **[0045]** De acuerdo con las técnicas descritas en esta divulgación, si es el último uso de un valor intermedio, el elemento de procesamiento puede evitar almacenar el valor intermedio en el GPR. Por ejemplo, el elemento de procesamiento puede saltar el almacenamiento del valor intermedio en el GPR. En algunos ejemplos, el elemento de procesamiento puede retroalimentar solamente el valor intermedio como un operando para las otras operaciones constituyentes de las operaciones definidas por la instrucción. Si no es el último uso del valor intermedio o si el valor intermedio no está recibiendo retroalimentación, incluso si es el último uso del valor intermedio, el elemento de procesamiento puede almacenar el valor intermedio en el GPR.
- 40 **[0046]** Puede haber diversas formas de determinar si un valor intermedio está al final de su ciclo de vida. Por ejemplo, el procesador 12 puede ejecutar un compilador que compile el programa de sombreado (por ejemplo, un programa de sombreado de gráficos) que se ejecutará en el núcleo de sombreado de la GPU 14. Como parte de la compilación, el compilador puede separar la pluralidad de operaciones definidas en la instrucción del programa de sombreado en operaciones constituyentes. El compilador puede determinar entonces cuándo un valor intermedio está al final de su vida útil (es decir, cuándo se produce el último uso del valor intermedio) al explorar las operaciones constituyentes.
- 45 **[0047]** Cuando el compilador, que se ejecuta en el procesador 12, determina el último uso de un valor intermedio particular, el compilador puede incluir una indicación que indique el último uso de un valor intermedio particular. Por
- 50
- 55
- 60
- 65

ejemplo, el compilador puede incluir un indicador asociado con el valor intermedio que indique si es el último uso del valor intermedio (por ejemplo, un valor de 1 para el indicador indica que es el último uso y un valor de 0 para el indicador indica que no es el último uso, o viceversa).

5 **[0048]** En dichos ejemplos, un controlador de un elemento de procesamiento puede leer el valor del indicador asociado con el valor intermedio y, en base al valor del indicador, determinar si el valor intermedio debe almacenarse o no almacenarse (por ejemplo, evitar o saltar el almacenamiento del valor intermedio). Si no se va a almacenar el valor intermedio, el controlador del elemento de procesamiento puede saltar el almacenamiento del valor intermedio (o bloquear el almacenamiento del valor intermedio) en el GPR y, si el valor intermedio debe almacenarse, el controlador del elemento de procesamiento almacena el valor intermedio en el GPR.

15 **[0049]** De esta manera, las técnicas pueden reducir la cantidad de veces que los datos se almacenan en el GPR, promoviendo de ese modo el ahorro de energía. Por ejemplo, cada almacenamiento en el GPR puede consumir relativamente poca energía. Sin embargo, dado que la GPU 14 proporciona capacidades masivas de procesamiento paralelo, hay muchas escrituras en cada GPR. Al usar un esquema que evite (por ejemplo, salte) almacenar datos en el GPR si no se necesita almacenamiento, puede haber una reducción general en el uso de energía por la GPU 14 en comparación con algunas otras técnicas.

20 **[0050]** En algunas otras técnicas, el elemento de procesamiento siempre almacena un valor intermedio en el GPR. Por ejemplo, estas otras técnicas no lograron reconocer si un valor intermedio está al final de su ciclo de vida, que no puede haber un beneficio apreciable de almacenar el valor intermedio en el GPR y que puede haber un impacto negativo en el almacenamiento del valor intermedio que está al final de su ciclo de vida. Además, estas otras técnicas fallaron al reconocer que un compilador puede aprovecharse para determinar el final del ciclo de vida de un valor intermedio y que el compilador puede usarse para identificar cuándo fue el último uso del valor intermedio (por ejemplo, identificar cuando el final del ciclo de vida del valor intermedio se producirá tal como con un indicador, como ejemplo). Estas otras técnicas no pueden usar un controlador de un elemento de procesamiento en un núcleo de sombreado de la GPU 14 para determinar cuándo se identifica el último uso de un valor intermedio y evitar el almacenamiento del valor intermedio en dichos casos.

30 **[0051]** La FIG. 2 es un diagrama de bloques que ilustra los componentes del dispositivo ilustrado en la FIG. 1 en mayor detalle. Como se ilustra en la FIG. 2, la GPU 14 incluye el núcleo de sombreado 28 y el canal de función fija 30. El núcleo de sombreado 28 y el canal de función fija 30 pueden formar juntos un canal de procesamiento usado para realizar funciones relacionadas con gráficos o no relacionadas con gráficos. El canal de procesamiento realiza las funciones definidas por el software o el firmware que se ejecutan en la GPU 14 y realiza funciones mediante unidades de función fija que están cableadas para realizar funciones específicas.

40 **[0052]** Como se describió anteriormente, el software o el firmware que se ejecuta en la GPU 14 se puede denominar programas de sombreado (o simplemente sombreadores), y los programas de sombreado se pueden ejecutar en el núcleo de sombreado 28 de la GPU 14. Aunque se ilustra solamente un núcleo de sombreado 28, en algunos ejemplos, la GPU 14 puede incluir uno o más núcleos de sombreado similares al núcleo de sombreado 28. El canal de función fija 30 incluye las unidades de función fija. El núcleo de sombreado 28 y el canal de función fija 30 pueden transmitir y recibir datos entre sí. Por ejemplo, el canal de procesamiento puede incluir programas de sombreado que se ejecuten en el núcleo de sombreado 28 que reciban datos de una unidad de función fija del canal de función fija 30 y emitan datos procesados a otra unidad de función fija del canal de función fija 30.

50 **[0053]** Los programas de sombreado proporcionan a los usuarios flexibilidad funcional porque un usuario puede diseñar el programa de sombreado para realizar las tareas deseadas de cualquier manera concebible. Sin embargo, las unidades de función fija están cableadas por la manera en que las unidades de función fija realizan tareas. En consecuencia, las unidades de función fija pueden no proporcionar mucha flexibilidad funcional.

55 **[0054]** Los ejemplos de los programas de sombreado incluyen el sombreador de vértices 32, el sombreador de fragmentos 34 y el sombreador de cálculo 36. El sombreador de vértices 32 y el sombreador de fragmentos 34 pueden ser programas de sombreado para tareas relacionadas con gráficos, y el sombreador de cálculo 36 puede ser un programa de sombreado para una tarea no relacionada con gráficos. En algunos ejemplos, solo se pueden usar programas de sombreado relacionados con gráficos como el sombreador de vértices 32 y el sombreador de fragmentos 34. En algunos ejemplos, solo se pueden usar programas de sombreado no relacionados con gráficos como el sombreador de cálculo 36. Hay ejemplos adicionales de programas de sombreado tales como sombreadores de geometría, que no se describen para los propósitos de brevedad.

60 **[0055]** Como se describe a continuación, el controlador de gráficos 26 que se ejecuta en el procesador 12 puede configurarse para implementar una interfaz de programación de aplicaciones (API). En dichos ejemplos, los programas de sombreado (por ejemplo, el sombreador de vértices 32, el sombreador de fragmentos 34 y el sombreador de cálculo 36) se pueden configurar de acuerdo con la misma API que el controlador de gráficos 26. Aunque no se ilustra, la memoria de sistema 16 puede almacenar el código para el controlador de gráficos 26 que el procesador 12 recupera de la memoria de sistema 16 para su ejecución. El controlador de gráficos 26 se ilustra en una casilla discontinua para indicar que el controlador de gráficos 26 es un software, que se ejecuta en hardware

(por ejemplo, el procesador 12), en este ejemplo. Sin embargo, parte o la totalidad de la funcionalidad del controlador de gráficos 26 puede implementarse como hardware en el procesador 12.

[0056] En algunos ejemplos, la memoria de sistema 16 puede almacenar el código fuente para uno o más del sombreador de vértices 32, del sombreador de fragmentos 34 y del sombreador de cálculo 36. En estos ejemplos, el compilador 24 que se ejecuta en el procesador 12 puede compilar el código fuente de estos programas de sombreado para crear un código objeto o intermedio ejecutable por el núcleo de sombreado 28 de la GPU 14 durante el tiempo de ejecución (por ejemplo, en el momento en que estos programas de sombreado necesiten ejecutarse en el núcleo de sombreado 28). En algunos ejemplos, el compilador 24 puede compilar previamente los programas de sombreado y almacenar el objeto o código intermedio de los programas de sombreado en la memoria de sistema 16.

[0057] De manera similar al controlador de gráficos 26, aunque no se ilustra, la memoria de sistema 16 puede almacenar el código para el compilador 24 que el procesador 12 recupere de la memoria de sistema 16 para su ejecución. El compilador 24 se ilustra en una casilla discontinua para indicar que el compilador 24 es un software, que se ejecuta en hardware (por ejemplo, el procesador 12), en este ejemplo. Sin embargo, alguna funcionalidad del compilador 24 puede implementarse como hardware en el procesador 12, en algunos ejemplos.

[0058] El controlador de gráficos 26 puede configurarse para permitir que el procesador 12 y la GPU 14 se comuniquen entre sí. Por ejemplo, cuando el procesador 12 descarga tareas de procesamiento de gráficos o sin gráficos a la GPU 14, el procesador 12 descarga dichas tareas de procesamiento a la GPU 14 a través del controlador de gráficos 26.

[0059] Como ejemplo, el procesador 12 puede ejecutar una aplicación de juegos que produzca datos gráficos, y el procesador 12 puede descargar el procesamiento de estos datos gráficos a la GPU 14. En este ejemplo, el procesador 12 puede almacenar los datos gráficos en la memoria de sistema 16, y el controlador de gráficos 26 puede indicar a la GPU 14 cuándo recuperar los datos gráficos, desde dónde recuperar los datos gráficos en la memoria de sistema 16 y cuándo procesar los gráficos datos. Además, la aplicación de juegos puede requerir que la GPU 14 ejecute uno o más programas de sombreado. Por ejemplo, la aplicación de juegos puede requerir que el núcleo de sombreado 28 ejecute el sombreador de vértices 32 y el sombreador de fragmentos 34 para generar imágenes que se vayan a visualizar (por ejemplo, en la pantalla 18 de la FIG. 1). El controlador de gráficos 26 puede ordenar a la GPU 14 cuándo ejecutar los programas de sombreado y ordenar a la GPU 14 dónde recuperar los datos de gráficos necesarios para los programas de sombreado. De esta forma, el controlador de gráficos 26 puede formar un enlace entre el procesador 12 y la GPU 14.

[0060] El controlador de gráficos 26 se puede configurar de acuerdo con una API; aunque el controlador de gráficos 26 no necesite limitarse a configurarse de acuerdo con una API particular. En un ejemplo en el que el dispositivo 10 es un dispositivo móvil, el controlador de gráficos 26 puede configurarse de acuerdo con la API de OpenGL ES. La API de OpenGL ES está específicamente diseñada para dispositivos móviles. En un ejemplo en el que el dispositivo 10 es un dispositivo no móvil, el controlador de gráficos 26 puede configurarse de acuerdo con la API de OpenGL.

[0061] En las técnicas descritas en esta divulgación, el núcleo de sombreado 28 puede configurarse para ejecutar muchas instancias de la misma instrucción del mismo programa de sombreado en paralelo. Por ejemplo, el controlador de gráficos 26 puede ordenar a la GPU 14 que recupere valores de vértice para una pluralidad de vértices y ordenar a la GPU 14 que ejecute el sombreador de vértice 32 para procesar los valores de vértices de los vértices. En este ejemplo, el núcleo de sombreado 28 puede ejecutar múltiples instancias del sombreador de vértices 32 y hacerlo ejecutando una instancia del sombreador de vértices 32 en un elemento de procesamiento del núcleo de sombreado 28 para cada uno de los vértices.

[0062] Cada elemento de procesamiento del núcleo de sombreado 28 puede ejecutar la misma instrucción del sombreador de vértices 32 en la misma instancia; sin embargo, los valores de vértice particulares pueden ser diferentes porque cada elemento de procesamiento procesa un vértice diferente. Como se describió anteriormente, cada elemento de procesamiento puede considerarse como la ejecución de un subproceso del sombreador de vértices 32, donde un subproceso se refiere a una instrucción del sombreador de vértices 32 que está procesando un vértice particular. De esta manera, el núcleo de sombreado 28 puede ejecutar muchas instancias del sombreador de vértices 32 para procesar valores de vértice de una pluralidad de vértices en paralelo (por ejemplo, al mismo tiempo).

[0063] El núcleo de sombreado 28 puede ejecutar de manera similar muchas instancias del sombreador de fragmentos 34 para procesar valores de píxeles de una pluralidad de píxeles en paralelo o ejecutar muchas instancias del sombreador de cálculo 36 para procesar muchos datos no relacionados con gráficos en paralelo. De esta manera, el núcleo de sombreado 28 puede configurarse en una estructura de una sola instrucción, de datos múltiples (SIMD). Para facilitar la descripción, se describe lo siguiente con respecto a un programa de sombreado genérico, cuyos ejemplos incluyen el sombreador de vértices 32, el sombreador de fragmentos 34, el sombreador de cálculo 36 y otros tipos de sombreadores tales como los sombreadores de geometría.

5 **[0064]** Como también se describió anteriormente, el núcleo de sombreado 28 incluye un registro de uso general (GPR) para almacenar datos (por ejemplo, valores) generados por los elementos de procesamiento. Por ejemplo, un elemento de procesamiento puede incluir una pluralidad de etapas de ejecución. Estas etapas de ejecución pueden formarse en forma de canal (por ejemplo, una etapa de ejecución alimenta datos a la siguiente etapa de ejecución para el procesamiento). El resultado de la última etapa de ejecución puede enviarse al GPR para el almacenamiento de datos.

10 **[0065]** En algunos ejemplos, al procesar una instrucción de un programa de sombreado, las etapas de ejecución pueden producir un valor intermedio. Este valor intermedio puede ser necesario solamente temporalmente y, por lo tanto, puede que no sea necesario almacenar dicho valor intermedio en el GPR. Sin embargo, debido a la configuración de canal del elemento de procesamiento, algunas técnicas requieren el almacenamiento de un valor intermedio en el GPR, incluso si dicho almacenamiento no es necesario. Dicho almacenamiento puede consumir energía innecesariamente al almacenar valores en el GPR, cuando dicho almacenamiento no sea necesario.

15 **[0066]** En las técnicas descritas en esta divulgación, el procesador 12, a través del compilador 24, puede determinar si un valor intermedio necesita almacenarse en el GPR o no necesita almacenarse en el GPR. Si no es necesario almacenar el valor intermedio en el GPR, el procesador 12, a través del compilador 24, puede indicarlo como tal. Como ejemplo, el procesador 12, a través del compilador 24, puede incluir un indicador asociado con el valor intermedio. El indicador indica si el valor intermedio debe almacenarse en el GPR o no almacenarse. Un controlador dentro de un elemento de procesamiento del núcleo de sombreado 28 puede determinar si el valor intermedio debe almacenarse o no en base al indicador y si puede almacenar o no almacenar el valor intermedio en el GPR en base al indicador. En otras palabras, el controlador puede determinar si el valor intermedio debe almacenarse en el GPR en base, al menos en parte, a una indicación recibida (por ejemplo, el indicador). El controlador puede evitar el almacenamiento del valor intermedio en el GPR si se determina que el valor intermedio no se almacena en el GPR. El controlador puede almacenar el valor intermedio en el GPR si se determina que el valor intermedio debe almacenarse en el GPR.

20 **[0067]** Por ejemplo, una instrucción de un programa de sombreado puede definir una pluralidad de operaciones (por ejemplo, variables que se multipliquen, se dividan, se sumen, se resten, o una combinación). Durante la compilación del programa de sombreado, el compilador 24 puede separar la pluralidad de operaciones en operaciones constituyentes y crear sub-instrucciones para procesar cada una de las operaciones constituyentes.

25 **[0068]** Como ejemplo, una instrucción de un programa de sombreado (denominada instrucción DP4) puede definir las siguientes operaciones: $DP4 = X0 * X1 + Y0 * Y1 + Z0 * Z1 + W0 * W1$. En este ejemplo, $X0 * X1$, $Y0 * Y1$, $Z0 * Z1$ y $W0 * W1$ pueden considerarse operaciones constitutivas para la pluralidad de operaciones definidas por la instrucción DP4 porque cada una de estas operaciones forma parte de la pluralidad de operaciones definidas por la instrucción DP4.

30 **[0069]** En este ejemplo ilustrativo, el compilador 24 puede separar la instrucción DP4 en las siguientes sub-instrucciones:

35
40
45
Sub-instrucción 1: $R0 = X0 * X1$;
Sub-instrucción 2: $R1 = Y0 * Y1$;
Sub-instrucción 3: $R2 = Z0 * Z1 + R0$;
Sub-instrucción 4: $R3 = W0 * W1 + R1$;
Sub-instrucción 5: $R4 = R2 + R3$.

50 **[0070]** En este ejemplo, R4 es igual al valor final de la instrucción DP4, que se produce ejecutando una pluralidad de sub-instrucciones. Por ejemplo, las etapas de ejecución de un elemento de procesamiento pueden ejecutar las sub-instrucciones de ejemplo anteriores en forma de canal, donde cada sub-instrucción produzca un valor intermedio a partir de una operación constituyente que se use para generar el valor final. Por ejemplo, en el ejemplo anterior, R0, R1, R2 y R3 son todos valores intermedios que se calculan como parte de la determinación del valor final R4, pero no son el valor final en sí mismos. En el ejemplo anterior, el GPR puede almacenar el valor R4 como resultado de la instrucción DP4.

55 **[0071]** Como se puede ver en las sub-instrucciones anteriores, después de la sub-instrucción 3, el valor intermedio R0 ya no es necesario, después de la sub-instrucción 4, el valor intermedio R1 ya no es necesario y, después de la sub-instrucción 5, los valores intermedios R2 y R3 ya no son necesarios. Sin embargo, en algunas técnicas, debido a la estructura de canal de los elementos de procesamiento del núcleo de sombreado 28, los valores intermedios R0, R1, R2 y R3 se almacenan en el GPR incluso después de que ya no se necesiten.

60 **[0072]** En las técnicas descritas en la divulgación, el compilador 24 puede escanear las sub-instrucciones para determinar si algún valor intermedio no necesita almacenarse en el GPR y puede indicar como tal cualquier valor intermedio que no sea necesario para el almacenamiento en el GPR. Por ejemplo, en el ejemplo anterior, el compilador 24 puede determinar que el valor intermedio R0 ya no es necesario después de la sub-instrucción 3, que

el valor intermedio R1 ya no es necesario después de la sub-instrucción 4 y que los valores intermedios R2 y R3 ya no son necesarios después de la sub-instrucción 5.

[0073] En algunos ejemplos, el procesador 12, a través del compilador 24, puede incluir una indicación con un valor intermedio que indique que el valor intermedio ya no es necesario para realizar las operaciones definidas por la instrucción del programa de sombreado. Por ejemplo, cuando ya no se necesite un valor intermedio, el compilador 24 puede indicar que es el último uso (LU) del valor intermedio, lo que significa el final del ciclo de vida del valor intermedio. Como ejemplo, el compilador 24 puede incluir un indicador LU que indique el último uso del valor intermedio en las sub-instrucciones. Por ejemplo, las siguientes sub-instrucciones incluyen el indicador LU que indica el final del ciclo de vida de los valores intermedios.

Sub-instrucción 1': $R0 = X0 * X1$;
 Sub-instrucción 2': $R1 = Y0 * Y1$;
 Sub-instrucción 3': $R2 = Z0 * Z1 + (LU) R0$;
 Sub-instrucción 4': $R3 = W0 * W1 + (LU) R1$;
 Sub-instrucción 5': $R4 = (LU) R2 + (LU) R3$.

[0074] En las técnicas descritas en esta divulgación, un controlador dentro de un elemento de procesamiento puede leer los indicadores de LU para determinar que un valor intermedio no debe almacenarse en el GPR. En estos ejemplos, el controlador puede saltar el almacenamiento (por ejemplo, evitar el almacenamiento) de los valores intermedios en el GPR, lo que a su vez puede reducir la cantidad de energía consumida por la GPU 14. Sin embargo, si el indicador LU no se afirma para ciertos valores intermedios, el controlador puede determinar que el valor intermedio debe almacenarse en el GPR, y puede almacenar el valor intermedio en el GPR.

[0075] La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de un núcleo de sombreado de una GPU de la FIG. 2 en mayor detalle. Por ejemplo, la FIG. 3 ilustra un ejemplo del núcleo de sombreado 28 de la GPU 14. El núcleo de sombreado 28 puede incluir la unidad de control 38, la memoria de instrucciones 40, uno o más elementos de procesamiento 42A-42H (denominados colectivamente como "elementos de procesamiento 42") y el registro de uso general (GPR) 44. El GPR 44 puede almacenar datos y, por lo tanto, también puede denominarse memoria de datos. Aunque la FIG. 3 ilustre ocho elementos de procesamiento 42, puede haber más o menos de ocho elementos de procesamiento 42 en otros ejemplos. Los elementos de procesamiento 42 son ejemplos de elementos de procesamiento en cuyas instancias del programa de sombreado se ejecutan en paralelo.

[0076] La unidad de control 38 puede controlar la funcionalidad del sombreador 28. Por ejemplo, la unidad de control 38 puede recuperar las instrucciones que se vayan a ejecutar procesando los elementos 42 y almacenar las instrucciones en la memoria de instrucciones 40. Además, la unidad de control 38 puede recuperar los valores (por ejemplo, datos) de los elementos de procesamiento 42 para procesar y almacenar los valores en la memoria de datos 44.

[0077] La memoria de instrucciones 40 puede ser cualquier tipo de memoria capaz de almacenar instrucciones tales como, pero no limitadas a, memoria volátil, memoria no volátil, memoria caché, memoria de acceso aleatorio (RAM), RAM estática (SRAM), RAM dinámica (DRAM), etc. La unidad de control 38 puede ser una circuitería cableada del núcleo de sombreado 28 que controle los componentes del núcleo de sombreado 28. Sin embargo, puede ser posible que la unidad de control 38 sea software o firmware, que se ejecute en hardware, del núcleo de sombreado 28.

[0078] Los elementos de procesamiento 42 están configurados para ejecutar subprocesos de un programa de sombreado. Cada uno de los elementos de procesamiento 42 puede ejecutar un subproceso diferente. Por ejemplo, cada uno de los elementos de procesamiento 42 puede ejecutar una instancia de una instrucción de un programa de sombreado con respecto a elementos de datos potencialmente diferentes. Los elementos de procesamiento 42 pueden ser elementos de procesamiento de una sola instrucción, de datos múltiples (SIMD). Los elementos de procesamiento SIMD se refieren a elementos de procesamiento que, cuando se activan, están configurados para ejecutar la misma instrucción al mismo tiempo con respecto a datos diferentes. Esto puede permitir que los elementos de procesamiento 42 ejecuten una pluralidad de subprocesos de un programa de sombreado en paralelo con respecto a diferentes elementos de datos. En algunos casos, cada uno de los elementos de procesamiento 42 puede ejecutar instrucciones de un programa de sombreado en base a un contador de programa común que apunte a una instrucción contenida en la memoria de instrucciones 40.

[0079] Si uno o más de los elementos de procesamiento 42 se desactivan por la unidad de control 38, dichos elementos de procesamiento 42 no ejecutan una instrucción de programa para un ciclo de instrucción dado. En algunos casos, la unidad de control 38 puede desactivar uno o más de los elementos de procesamiento 42 para implementar instrucciones de ramificación condicional donde la condición de ramificación se satisfaga para algunos subprocesos y no se satisfaga para otros subprocesos.

[0080] En algunos ejemplos, cada uno de los elementos de procesamiento 42 puede incluir y/o corresponder a una o más unidades lógicas aritméticas (ALU). En ejemplos adicionales, cada uno de los elementos de procesamiento 42

puede implementar la funcionalidad ALU. La funcionalidad ALU puede incluir suma, resta, multiplicación, etc.

[0081] De acuerdo con las técnicas descritas en esta divulgación, el compilador 24 puede separar una instrucción que defina una pluralidad de operaciones en sub-operaciones e indicar qué valores intermedios no son necesarios para almacenarse en el GPR de los respectivos elementos de procesamiento 42. Por ejemplo, la memoria de instrucciones 40 puede almacenar una sub-instrucción con un valor de indicador asociado con un valor intermedio que indique si es el último uso del valor intermedio, donde la sub-instrucción se derive de la instrucción que define una pluralidad de operaciones y que realiza una de las operaciones constituyentes. Un controlador dentro de los respectivos elementos de procesamiento 42 puede determinar si el valor intermedio debe almacenarse en el GPR o no, y saltar el almacenamiento (por ejemplo, evitar el almacenamiento) si no es necesario el almacenamiento del valor intermedio, como lo indica el valor de indicador, o almacenar si se necesita el almacenamiento.

[0082] La FIG. 4 es un diagrama de bloques que ilustra un ejemplo de un elemento de procesamiento de un núcleo de sombreado de la FIG. 3 en mayor detalle. Por ejemplo, la FIG. 4 ilustra el elemento de procesamiento 42A. Los elementos de procesamiento 42B-42H pueden incluir componentes similares a los descritos con respecto al elemento de procesamiento 42A. Como se ilustra, el elemento de procesamiento 42A incluye la estructura de canal 46, el controlador 48, la lógica de reenvío (FW) 54A y 54B y el reloj 56. La estructura de canal 46 incluye las etapas de ejecución 50A-50C (denominadas colectivamente etapas de ejecución 50). Como se ilustra, la estructura de canal 46 incluye tres etapas de ejecución 50. Sin embargo, la estructura de canal 46 puede incluir más o menos de tres etapas de ejecución 50 en diferentes ejemplos.

[0083] Cada una de las etapas de ejecución 50 realiza una función por ciclo de reloj desde el reloj 56. Por ejemplo, cada una de las etapas de ejecución 50 puede configurarse para realizar una operación aritmética tal como una operación constituyente de la pluralidad de operaciones definidas por una instrucción de un programa de sombreado. Cada una de las etapas de ejecución 50 puede transmitir el resultado de la operación aritmética a la siguiente etapa de ejecución 50 en un borde ascendente o descendente del reloj 56. De esta manera, las etapas de ejecución 50 forman la estructura de canal 46 en la que las operaciones constituyentes de la pluralidad de operaciones definidas por una instrucción se llevan a cabo en forma de canal.

[0084] El controlador 48 puede estar configurado para transmitir las sub-instrucciones que deba realizar cada una de las etapas de ejecución 50. Por ejemplo, como se describió anteriormente, la unidad de control 38 del núcleo de sombreado 28 puede cargar las instrucciones, incluidas las sub-instrucciones según lo determine el compilador 24, de un programa de sombreado en la memoria de instrucciones 40. El controlador 48 del elemento de procesamiento 42A puede recuperar las sub-instrucciones de la memoria de instrucciones 40 y los valores de datos que formen los operandos para las operaciones constituyentes. El controlador 48 puede alimentar los operandos (por ejemplo, los valores de datos) a las respectivas etapas de ejecución 50. Además, si una cualquiera de las etapas de ejecución 50 es procesar los datos resultantes de otra de las etapas de ejecución 50, entonces los datos de la otra de las etapas de ejecución 50 también pueden ser un operando en la operación constitutiva que se vaya a realizar.

[0085] Como también se ilustra, la estructura de canal 46 puede incluir lógica de reenvío tal como la lógica de reenvío 54A y la lógica de reenvío 54B. La lógica de reenvío 54A y 54B se puede considerar como lógica interna de reenvío ALU. Por ejemplo, en la estructura de canal 46, puede haber algún beneficio de poder transmitir resultados de retorno de una de las etapas de ejecución 50 a una anterior de las etapas de ejecución 50. Por ejemplo, una de las etapas de ejecución 50 puede generar un valor intermedio que se forme como un operando para otra operación constitutiva que esté más temprano en la estructura de canal 46. Con la lógica de reenvío 54A y 54B, una de las etapas de ejecución 50 puede ser capaz de reenviar el valor a una etapa anterior de ejecución 50 en la estructura de canal 46.

[0086] Además, aunque no se ilustra en la FIG. 4, en algunos ejemplos, el elemento de procesamiento 42A puede incluir una lógica de reenvío adicional para compensar la latencia asociada al acceso al GPR 44. Por ejemplo, dichas unidades lógicas de reenvío de compensación de latencia pueden permitir que la salida de la etapa de ejecución 50C retroalimente a la etapa de ejecución 50A.

[0087] En algunas otras técnicas, incluso en aquellas que incluyan la lógica de reenvío de compensación de latencia, el GPR 44 almacenaba siempre la salida de la etapa de ejecución 50C, incluso si dicho almacenamiento no era necesario. Por ejemplo, en la estructura de canal 46, todos los valores de una etapa anterior de ejecución 50 pasan a la siguiente etapa de ejecución 50. Esto puede ser cierto incluso si el valor se reenvía. Por ejemplo, la salida de la etapa de ejecución 50C puede retroalimentarse a la lógica de reenvío 54B y puede transmitirse a la GPR 44. Por lo tanto, todos los valores intermedios pueden desplazarse a través de las etapas de ejecución 50 y, en estas otras técnicas, todos los valores intermedios se almacenarán en el GPR 44.

[0088] Como ilustración, un ejemplo anterior de una instrucción fue $X0 * X1 + Y0 * Y1 + Z0 * Z1 + W0 * W1$, con las siguientes sub-instrucciones:

Sub-instrucción 1: $R0 = X0 * X1$;
Sub-instrucción 2: $R1 = Y0 * Y1$;

ES 2 691 957 T3

Sub-instrucción 3: $R2 = Z0 * Z1 + R0$;
 Sub-instrucción 4: $R3 = W0 * W1 + R1$;
 Sub-instrucción 5: $R4 = R2 + R3$.

5 **[0089]** En el pseudocódigo, lo anterior se puede procesar de la siguiente manera mediante etapas de ejecución 50, vistas desde la perspectiva de la etapa de ejecución 50A:

10 Ciclo 0: $R0 = X0 * X1$;
 Ciclo 1: $R1 = Y0 * Y1$;
 Ciclo 2: $R2 = Z0 * Z1 + R0$; //R0 se reenvía en FW 54B
 Ciclo 3: $R3 = W0 * W1 + R1$; //R1 se reenvía en FW54B
 Ciclo 4: NOP; //R0 escribir al GPR 44
 Ciclo 5: NOP; //R1 escribir al GPR 44, R2 se reenvía en FW54A
 Ciclo 6: $R4 = R2 + R3$; // R2 escribir al GPR 44, R3 se reenvía en FW54B
 15 Ciclo 7: NOP; //R3 escribir al GPR 44

20 **[0090]** En el ejemplo anterior, escribir R0, R1, R2 y R3 al GPR 44 en ciclos 4, 5, 6 y 7, respectivamente, es innecesario. Por ejemplo, después del ciclo 2, R0 ya no es necesario, después del ciclo 3, R1 ya no es necesario y, después del ciclo 6, R2 y R3 ya no son necesarios. Sin embargo, en estas otras técnicas, R0, R1 y R2 se escriben en el GPR 44 en los ciclos 4, 5 y 6, respectivamente, y después de que estos valores ya no sean necesarios para determinar el valor final de la instrucción DP4, y se escribe R3 al GPR 44 en el ciclo 7, aunque ya no se necesite R3 para determinar el valor final de la instrucción DP4.

25 **[0091]** La siguiente tabla es un ejemplo que ilustra las funciones realizadas por cada una de las etapas de ejecución 50 para el ejemplo anterior.

Tabla 1. Momento de las operaciones a través de las etapas de ejecución

	T1	T2	T3	T4	T5	T6	T7	T8
Etapa 50A	X0, X1	Y0, Y1	Z0, Z1	W0, W1		R2		
Etapa 50B		$R0 = X0 * X1$	$R1 = Y0 * Y1$	$R2 = Z0 * Z1 + R0$	$R3 = W0 * W1 + R1$		$R4 = R2 + R3$	
Etapa 50C			R0	R1	R2	R3		R4

30 **[0092]** Como se ilustra en la Tabla 1, en el tiempo T1, la etapa de ejecución 50A recibe los valores de X0 y X1 de la memoria de datos 40 a través del controlador 48. En el tiempo T2, el controlador 48 provee a la etapa de ejecución 50B con la sub-instrucción que causa que la etapa de ejecución 50B determine el valor intermedio R0 multiplicando X0 y X1 y, en el tiempo T2, la etapa de ejecución 50A recibe los valores de Y0 e Y1 de la memoria de datos 40 a través del controlador 48.

35 **[0093]** En el tiempo T3, la etapa de ejecución 50C recibe el valor intermedio R0, y el controlador 48 provee a la etapa de ejecución 50B con la sub-instrucción que causa que la etapa de ejecución 50B determine el valor intermedio R1 multiplicando Y0 e Y1. Además, en el tiempo T3, la etapa de ejecución 50A recibe los valores de Z0 y Z1 de la memoria de datos 40 a través del controlador 48. Además, el valor intermedio R0 se reenvía a la etapa de ejecución 50B a través de FW 54B. En este ejemplo, la etapa de ejecución 50C escribe el valor intermedio R0 (innecesariamente) en el GPR 44.

40 **[0094]** En el tiempo T4, la etapa de ejecución 50C recibe el valor intermedio R1, y la etapa de ejecución 50C reenvía R1 a la etapa de ejecución 50B a través de FW 54B. También, en el tiempo T4, el controlador 48 provee a la etapa de ejecución 50B con la sub-instrucción que causa que la etapa de ejecución 50B determine el valor intermedio R2 multiplicando Z0 y Z1 y sumando el resultado a R0. Además, en el tiempo T4, la etapa de ejecución 50A recibe los valores de W0 y W1 de la memoria de datos 40 a través del controlador 48. En este ejemplo, la etapa de ejecución 50C escribe el valor intermedio R1 (innecesariamente) en el GPR 44.

45 **[0095]** En el tiempo T5, la etapa de ejecución 50C recibe el valor intermedio R2 y, en este ejemplo, la etapa de ejecución 50C escribe el valor intermedio R2 (innecesariamente) en el GPR 44. El controlador 48 provee a la etapa de ejecución 50B con la sub-instrucción que causa que la etapa de ejecución 50B determine el valor intermedio R3 multiplicando W0 y W1 y sumando el resultado a R3.

50 **[0096]** En el tiempo T6, la etapa de ejecución 50C escribe el valor intermedio R3 (innecesariamente) al GPR 44, reenvía el valor intermedio R3 a la etapa de ejecución 50B y la etapa de ejecución 50A recibe el valor intermedio R2. En el tiempo T7, la etapa de ejecución 50B calcula R4 sumando el valor intermedio R2 y el valor intermedio R3. En el tiempo T8, la etapa de ejecución 50C almacena el valor R4 al GPR 44.

[0097] En el ejemplo anterior, el almacenamiento de los valores intermedios R0, R1, R2 y R3 al GPR 44 es innecesario porque estos valores no son necesarios después del último uso. Sin embargo, estos valores intermedios aún pasan a través de las diversas etapas de ejecución 50 para el almacenamiento eventual en el GPR 44. En las técnicas descritas en esta divulgación, el compilador 24 puede determinar el final del ciclo de vida (por ejemplo, el último uso) de un valor intermedio e incluir una indicación (por ejemplo, un indicador) asociada con el valor intermedio. En estos ejemplos, el valor intermedio, incluso si no se va a almacenar en el GPR 44, todavía se desplaza por las etapas de ejecución 50. Sin embargo, el controlador 48 lee el indicador asociado con el valor intermedio y, en base al valor del indicador, no almacena el valor intermedio en el GPR 44 (por ejemplo, salta el almacenamiento del valor intermedio).

[0098] Por ejemplo, el compilador 24 puede escanear a través de las sub-instrucciones 1-5 de una instrucción de un programa de sombreado. Si se genera un operando (por ejemplo, entrada a una sub-instrucción) en una sub-instrucción anterior (por ejemplo, es un valor intermedio), el compilador 24 puede determinar cuándo ya no se necesita ese valor intermedio para determinar el valor final de la instrucción. Por ejemplo, el compilador 24 puede determinar el último uso del valor intermedio en las sub-instrucciones. En base al último uso determinado, el compilador 24 puede incluir un indicador de último uso (LU) con el valor intermedio. En otras palabras, el indicador LU indica que el ciclo de vida de los datos se finaliza después que se usa (por ejemplo, se lee), independientemente de si se lee desde los canales hacia adelante, la lógica de compensación de latencia o el GPR 44.

[0099] El controlador 48 puede configurarse para leer el valor indicador asociado con el valor intermedio y determinar si el valor intermedio debe almacenarse en el GPR 44 o no. Por ejemplo, como se describió anteriormente, diversas etapas de ejecución 50 pueden enviar datos a través de la lógica de reenvío 54A o 54B y/o a través de las unidades lógicas de compensación de latencia. En algunos ejemplos, el controlador 48 puede determinar que se están reenviando valores intermedios. Si el controlador 48 determina que el indicador asociado con el valor intermedio indica el último uso del valor intermedio y el controlador 48 determina que el valor intermedio se está reenviando (por ejemplo, la lógica de reenvío 54A o 54B y la una o más unidades lógicas de compensación de latencia están "activas"), entonces el controlador 48 puede determinar que el valor intermedio no necesita almacenarse en el GPR 44. En estos ejemplos, el controlador 48 puede saltar el almacenamiento (por ejemplo, evitar el almacenamiento) del valor intermedio en el GPR 44.

[0100] Como se describió anteriormente, en algunos ejemplos, además de determinar si se afirma el indicador LU (por ejemplo, uno digital), el controlador 48 puede determinar si el valor intermedio asociado con el indicador LU se está reenviando. En algunos ejemplos, el controlador 48 puede evitar almacenar el valor intermedio solamente si se afirma el indicador LU asociado y se reenvía el valor intermedio. Por ejemplo, incluso si se afirma el indicador LU, pero el valor intermedio no se reenvía, el controlador 48 aún puede almacenar el valor intermedio en el GPR 44. En otras palabras, si el controlador 38 determina que el valor intermedio no se reenvía, entonces el controlador 48 puede almacenar el valor intermedio en el GPR incluso si se indica que el valor intermedio no se almacena en el GPR. Por ejemplo, en algunos ejemplos, debido a la latencia de búsqueda de instrucciones o si el elemento de procesamiento 42A pierde el arbitraje a otro subproceso, entonces puede ser necesario almacenar el valor intermedio en el GPR 44, pero no reenviarlo. Por lo tanto, en algunos ejemplos, el controlador 48 puede evitar almacenar un valor intermedio tanto si se afirma el indicador de LU como si se determina que el valor intermedio se reenvía.

[0101] Como ilustración, la divulgación describe un pseudocódigo para los valores que se procesan a través de las etapas de ejecución 50 en el ejemplo en el que los valores intermedios se almacenan innecesariamente en el GPR 44. El siguiente pseudocódigo ilustra el último indicador de uso (LU), así como también indica que el valor intermedio se está reenviando para que el controlador 48 determine que no es necesario almacenar el valor intermedio. El siguiente pseudocódigo se basa en las sub-instrucciones que incluyen el indicador LU, que como se describió anteriormente fueron:

Sub-instrucción 1': $R0 = X0 * X1$;
 Sub-instrucción 2': $R1 = Y0 * Y1$;
 Sub-instrucción 3': $R2 = Z0 * Z1 + (LU) R0$;
 Sub-instrucción 4': $R3 = W0 * W1 + (LU) R1$;
 Sub-instrucción 5': $R4 = (LU)R2 + (LU) R3$.

[0102] El pseudo-código con el indicador LU es el siguiente:

Ciclo 0: $R0 = X0 * X1$;
 Ciclo 1: $R1 = Y0 * Y1$;
 Ciclo 2: $R2 = Z0 * Z1 + (LU) R0$; //R0 se reenvía en FW 54B
 Ciclo 3: $R3 = W0 * W1 + (LU) R1$; //R1 se reenvía en FW54B
 Ciclo 4: NOP; //saltar R0 y escribir al GPR 44 porque se produjo el reenvío en FW 54B
 Ciclo 5: NOP; //saltar R1 y escribir al GPR 44 porque se produjo un reenvío en FW 54B
 Ciclo 6: $R4 = R2 + R3$; //saltar R2 y escribir al GPR 44 porque la unidad lógica de compensación de latencia se producirá a continuación

Ciclo 7: NOP; //saltar R3 y escribir al GPR 44, porque se reenvía la unidad lógica de compensación de latencia, R2 se reenvía a través de una primera de las unidades lógicas de compensación de latencia, R3 se reenvía a través de una segunda de las unidades lógicas de compensación de latencia.

5 **[0103]** Como se describió anteriormente, en algunos ejemplos, el controlador 48 puede determinar si se produce el reenvío de un valor intermedio, como parte de la determinación de si se debe evitar el almacenamiento del valor intermedio en el GPR 44. Como ejemplo, supongamos que, en el pseudocódigo anterior, hubo un retraso de tres ciclos al recuperar la parte de sub-instrucción de Z0*Z1 para el ciclo 2 de la memoria de instrucciones 40. Este retardo podría deberse a la latencia de búsqueda de instrucciones o la unidad de control 38 determina que se debería ejecutar un subproceso de mayor prioridad en el elemento de procesamiento 42A, ya que son posibles dos ejemplos y otras razones para el retardo.

15 **[0104]** En este caso, aún se puede necesitar el valor intermedio R0, pero se afirma el indicador LU. Debido a que R0 no se habría reenviado, el controlador 48 puede determinar que el valor intermedio R0 se debe almacenar en el GPR 44, incluso aunque se afirme el indicador LU. En otras palabras, en este caso, incluso aunque se afirme el indicador LU, el controlador 48 puede no evitar el salto del almacenamiento del valor intermedio R0. Sin embargo, si no hubiera retardo, entonces se reenviaría el valor intermedio R0 y se afirmaría su indicador LU asociado (por ejemplo, se establece en 1 o 0, según sea el caso, para indicar si es el último uso de los datos); por lo tanto, el controlador 48 evitaría almacenar el valor intermedio R0 en el GPR 44.

20 **[0105]** En la FIG. 4, se ilustran tres etapas de ejecución 50 y dos conjuntos de lógica de reenvío 54A y 54B. En algunos ejemplos, las etapas de ejecución 50 adicionales y la lógica de reenvío adicional pueden reducir además la cantidad de datos que deben escribirse en el GPR 44. Por ejemplo, el compilador 24 puede realizar un escaneo más profundo de las sub-instrucciones (por ejemplo, escanear más sub-instrucciones) porque hay más posibilidades de reenviar valores intermedios. Como hay más posibilidades de reenviar valores intermedios, el compilador 24 puede identificar más casos en los que el controlador 48 pueda saltar el almacenamiento de valores intermedios en el GPR 44. Por ejemplo, el compilador 24 puede identificar más valores intermedios con un ciclo de vida corto (que tiende a ser el caso para muchos programas de sombreado), o el compilador 24 puede optimizar el programa de sombreado para que haya más casos con valores intermedios con ciclos de vida cortos.

30 **[0106]** Además, debido a que hay menos lecturas y escrituras en el GPR 44, las técnicas no solo reducen el consumo de energía, sino que las técnicas descritas en esta divulgación pueden mejorar el rendimiento de otras maneras. Por ejemplo, el elemento de procesamiento 42A puede estar disponible antes que en otras técnicas para propósitos tales como lecturas de textura, carga/almacenamiento de memoria, etc. Por ejemplo, en otras técnicas, el elemento de procesamiento 42A puede almacenar datos innecesariamente en el GPR 44, lo que causa la cantidad de tiempo que el elemento de procesamiento 42A se asigna a un subproceso particular que se vaya a aumentar. Con las técnicas descritas en esta divulgación, el elemento de procesamiento 42A está disponible antes para procesar el siguiente subproceso, tal como un subproceso para leer textura o cargar/almacenar memoria, antes porque el elemento de procesamiento 42A puede no almacenar datos que posteriormente no se necesiten. De esta manera, el aumento en el rendimiento también permite que el GPR 44 esté disponible antes para escribir y leer en otros clientes de escritura/lectura para el acceso, ya que ocurren menos escrituras y lecturas y el GPR 44 se vuelve libre para dicho procesamiento antes.

45 **[0107]** La FIG. 5 es un diagrama de flujo que ilustra una técnica de ejemplo de valores de procesamiento. En el ejemplo ilustrado en la FIG. 5, un elemento de procesamiento recibe una pluralidad de sub-instrucciones de una instrucción de un programa (por ejemplo, un programa de sombreado) (100). La instrucción define una pluralidad de operaciones, y cada una de las sub-instrucciones define una operación constituyente de la pluralidad de operaciones. Por ejemplo, una unidad de procesamiento (por ejemplo, la GPU 14) incluye un núcleo (por ejemplo, el núcleo de sombreado 28). El núcleo incluye una memoria de instrucciones (por ejemplo, la memoria de instrucciones 40), un GPR (por ejemplo, el GPR 44) y al menos un elemento de procesamiento (por ejemplo, al menos uno de los elementos de procesamiento 42), y el al menos un elemento de procesamiento incluye un controlador (por ejemplo, el controlador 48) y una estructura de canal (por ejemplo, la estructura de canal 46) que tiene una pluralidad de etapas de ejecución (por ejemplo, las etapas de ejecución 50).

55 **[0108]** Como ejemplo, el controlador 48 está configurado para recibir la pluralidad de sub-instrucciones (por ejemplo, las sub-instrucciones 1'-5') de la instrucción (por ejemplo, la instrucción DP4) de un programa de sombreado. Las sub-instrucciones 1'-5' definen cada una operación constituyente de la pluralidad de operaciones definidas por la instrucción DP4. Además, en este ejemplo, cada uno de los elementos de procesamiento 42 puede configurarse para ejecutar las mismas sub-instrucciones 1'-5' de la misma instrucción al mismo tiempo (por ejemplo, en paralelo).

65 **[0109]** El elemento de procesamiento también puede recibir una indicación de al menos un valor intermedio generado por la ejecución de una de las sub-instrucciones que indica si el valor intermedio debe almacenarse en un GPR (102). Por ejemplo, la sub-instrucción 1' generó el valor intermedio R0. En este ejemplo, el controlador 48 recibió una indicación (por ejemplo, el indicador LU) para el valor intermedio R0 en la sub-instrucción 3' que indica si el valor intermedio R0 debe almacenarse en el GPR 44. Lo mismo ocurre con los valores intermedios R1, R2 y R3,

pero con respecto a diferentes sub-instrucciones.

[0110] El controlador puede determinar si el valor intermedio debe almacenarse en el GPR en base al menos en parte a la indicación recibida (104). Por ejemplo, cuando se esté ejecutando la sub-instrucción 3' en una de las etapas de ejecución 50, el controlador 48 puede determinar si el valor intermedio R0 debe almacenarse en el GPR 44 en base al indicador LU asociado con el valor intermedio R0. El controlador 48 puede determinar de manera similar si los valores intermedios R1, R2 y R3 deben almacenarse en el GPR 44. Por ejemplo, cuando la indicación de que el controlador 48 indica que el valor intermedio no debe almacenarse en el GPR 44, la indicación (por ejemplo, el indicador LU) también indica el último uso del valor intermedio en la pluralidad de sub-instrucciones. Después del último uso, el valor intermedio ya no es necesario por ninguna de las otras sub-instrucciones para determinar el valor final de la pluralidad de operaciones definidas por la instrucción.

[0111] En algunos ejemplos, el controlador 48 también puede determinar si el valor intermedio se reenvía a una etapa de ejecución anterior desde una etapa de ejecución que ejecutó la sub-instrucción que generó el valor intermedio. Por ejemplo, las etapas de ejecución 50 están dispuestas como un canal en la estructura de canal 46. El controlador 48 puede determinar que el valor intermedio R0 se reenvía a través de la lógica de reenvío 54B a una etapa de ejecución anterior. Como otro ejemplo, el controlador 48 puede determinar que los valores intermedios R2 y R3 se están reenviando a través de unidades lógicas de retardo de compensación de latencia. En dichos ejemplos, el controlador 48 puede determinar si el valor intermedio debe almacenarse en el GPR 44 en base, al menos en parte, a la indicación recibida y a la determinación de si el valor intermedio se reenvía. Por ejemplo, si se determina que el valor intermedio no se reenvía a la etapa de ejecución anterior, el controlador 48 puede almacenar el valor intermedio en el GPR 44 incluso si se indica que el valor intermedio no se debe almacenar en el GPR 44.

[0112] El controlador puede evitar el almacenamiento del valor intermedio si se determina que el valor intermedio no debe almacenarse en el GPR (106). Por ejemplo, el controlador 48 puede saltar el almacenamiento del valor intermedio en el GPR 44, en base a la determinación de que el valor intermedio no debe almacenarse en el GPR en base a la indicación, para evitar el almacenamiento innecesario de datos en el GPR 44. Sin embargo, si se determina que el valor intermedio debe almacenarse en el GPR 44 (por ejemplo, sin indicador LU), el controlador 48 puede almacenar el valor intermedio en el GPR 44. En las técnicas descritas en la divulgación, las etapas de ejecución 50 ejecutan las sub-instrucciones para generar un valor final a partir de la pluralidad de operaciones definidas por la instrucción, donde el valor final se genera a partir del valor intermedio que funciona como un operando a una de las sub-instrucciones. Si bien el controlador 48 puede no almacenar valores intermedios en el GPR 44 que no necesitan almacenarse, el controlador 48 puede almacenar el valor final en el GPR 44 como se muestra en la etapa de ejecución 50C al final de las sub-instrucciones.

[0113] La FIG. 6 es un diagrama de flujo que ilustra una técnica de ejemplo de compilación. En el ejemplo ilustrado en la FIG. 6, el compilador 24, que se ejecuta en el procesador 12, puede determinar una pluralidad de sub-instrucciones a partir de una instrucción de un programa (110). La instrucción define una pluralidad de operaciones, y cada sub-instrucción define una operación constituyente de la pluralidad de operaciones. Por ejemplo, el compilador 24, que se ejecuta en el procesador 12, puede determinar las sub-instrucciones 1-5 de la instrucción DP4.

[0114] El compilador 24, que se ejecuta en el procesador 12, puede determinar un último uso para al menos un valor intermedio que se generará mediante la ejecución de una de las sub-instrucciones (112). Por ejemplo, el compilador 24 puede determinar que el valor intermedio ya no es necesario por la pluralidad de sub-instrucciones para generar un valor final a partir de la pluralidad de operaciones definidas por la instrucción. Por ejemplo, en la sub-instrucción 3, el compilador 24 puede determinar que el valor intermedio R0 ya no es necesario por la pluralidad de sub-instrucciones (por ejemplo, ningún ciclo necesitará el valor intermedio R0 de nuevo) para generar el valor final a partir de la pluralidad de operaciones definido por la instrucción DP4. Por ejemplo, para generar R4, ya no se necesita el valor intermedio R0 porque el valor intermedio R2 ya incluye el valor del valor intermedio R0 (por ejemplo, R2 se genera a partir del valor intermedio R0).

[0115] El compilador 24, a través del procesador 12, puede indicar el último uso del al menos un valor intermedio en base a la determinación (114). Por ejemplo, el compilador 24 puede incluir un indicador que indique el último uso del al menos un valor intermedio en una de las sub-instrucciones que usa el valor intermedio como un operando. Por ejemplo, en la sub-instrucción 3', el valor intermedio R0 es un operando. En este ejemplo, el compilador 24 puede incluir el indicador LU con el valor intermedio R0 para producir la sub-instrucción 3'. El indicador LU con el valor intermedio R0 en la sub-instrucción 3' indica el último uso del valor intermedio R0. En otras palabras, la sub-instrucción 3' es la última instrucción para usar el valor intermedio R0 como un operando.

[0116] Además, como se describió anteriormente, indicar el último uso del al menos un valor intermedio también indica que el al menos un valor intermedio no necesita almacenarse en un GPR. Por ejemplo, el controlador 48 puede leer el indicador LU y determinar que el valor intermedio asociado con el indicador LU no necesita almacenarse en el GPR 44 para procesar el elemento 42A, que ejecuta las sub-instrucciones, del núcleo de sombreado 28 de la GPU 14.

5 [0117] El procesador 12 puede almacenar las sub-instrucciones compiladas (por ejemplo, las sub-instrucciones 1'-5') en la memoria de sistema 16 como el código de programa de sombreado. La GPU 14 puede recuperar las instrucciones del programa de sombreado, incluidas las sub-instrucciones 1'-5', desde la memoria de sistema 16, cuando así lo ordene el controlador de gráficos 28. Cada uno de los elementos de procesamiento 42 puede ejecutar las sub-instrucciones 1'-5' al mismo tiempo (por ejemplo, en paralelo).

10 [0118] En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o en cualquier combinación de estos. Si se implementan en software, las funciones pueden almacenarse en o transmitirse a través de, como una o más instrucciones o código, un medio legible por ordenador o ejecutarse mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que correspondan a un medio tangible tal como medios de almacenamiento de datos. De esta manera, los medios legibles por ordenador pueden corresponder en general a medios de almacenamiento legibles por ordenador tangibles que no sean transitorios. Los medios de almacenamiento de datos pueden ser cualquier medio disponible 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 esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

20 [0119] A modo de ejemplo, y no de manera limitativa, dichos medios de almacenamiento legibles por ordenador pueden comprender 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 pueda usarse para almacenar el código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda accederse mediante un ordenador. Debería entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen ondas portadoras, señales u otros medios transitorios, sino que, en cambio, se orientan a medios de almacenamiento tangibles no transitorios. El término disco, como se usa en el presente documento, incluye un disco compacto (CD), un disco láser, un disco óptico, un disco versátil digital (DVD), un disco flexible y un disco Blu-ray, donde algunos discos habitualmente reproducen datos magnéticamente, mientras que otros discos emiten datos ópticamente con láseres. Las combinaciones de lo anterior también deben incluirse dentro del alcance de los medios legibles por ordenador.

30 [0120] Las instrucciones pueden ejecutarse por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de uso general, circuitos integrados específicos de la aplicación (ASIC), matrices lógicas programables de campo (FPGA) u otros circuitos lógicos integrados o discretos equivalentes. En consecuencia, el término «procesador», como se usa en el presente documento, puede referirse 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 puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para la codificación y la decodificación, o incorporarse en un códec combinado. Además, las técnicas podrían implementarse por completo en uno o más circuitos o elementos lógicos.

40 [0121] Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluidos un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). Diversos componentes, módulos o unidades se describen en esta divulgación para enfatizar 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 ha descrito anteriormente, diversas unidades pueden combinarse en una unidad de hardware de códec o proporcionarse por medio de un grupo de unidades de hardware interoperativas, que incluyan uno o más procesadores como los descritos anteriormente, conjuntamente con software y/o firmware adecuados.

50 [0122] Se han descrito diversos ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Un procedimiento para procesar valores en un sistema de procesamiento de datos, comprendiendo el procedimiento:

5 recibir una pluralidad de sub-instrucciones de una instrucción de un programa, en la que la instrucción define una pluralidad de operaciones y en el que cada una de las sub-instrucciones define una operación constituyente de la pluralidad de operaciones;

10 recibir una indicación para al menos un valor intermedio generado por la ejecución de una de las sub-instrucciones que indica si el valor intermedio debe almacenarse en un registro de uso general (GPR);

15 determinar si el valor intermedio se reenvía a una etapa de ejecución anterior desde otra etapa de ejecución que ejecutó la sub-instrucción que generó el valor intermedio, en el que la etapa de ejecución anterior y la otra etapa de ejecución que generó el valor intermedio están dispuestas en una estructura de canal;

20 determinar si el valor intermedio debe almacenarse en el GPR en base al menos en parte a la indicación recibida y a la determinación de si el valor intermedio se reenvía a la etapa de ejecución anterior; y uno de:

25 evitar el almacenamiento del valor intermedio en el GPR si la indicación recibida indica que el valor intermedio no debe almacenarse en el GPR y se determina que el valor intermedio se reenvía a la etapa de ejecución anterior,

almacenar el valor intermedio en el GPR si la indicación recibida indica que el valor intermedio no debe almacenarse en el GPR y se determina que el valor intermedio no se reenvía a la etapa de ejecución anterior, o

almacenar el valor intermedio en el GPR si la indicación recibida indica que el valor intermedio debe almacenarse en el GPR.

2. El procedimiento según la reivindicación 1, que comprende además:

35 ejecutar las sub-instrucciones para generar un valor final, en el que el valor final se genera a partir del valor intermedio; y

almacenar el valor final en el GPR.

3. El procedimiento según la reivindicación 1, en el que el programa comprende un programa de sombreado, en el que recibir la pluralidad de sub-instrucciones comprende recibir, con un controlador de un elemento de procesamiento de un núcleo de sombreado de una unidad de procesamiento de gráficos (GPU), la pluralidad de sub-instrucciones, en el que recibir la indicación comprende recibir, con el controlador del elemento de procesamiento, la indicación de al menos un valor intermedio generado por la ejecución de una de las sub-instrucciones que indica si el valor intermedio debe almacenarse en el GPR;

50 en el que determinar si el valor intermedio se reenvía a la etapa de ejecución anterior comprende determinar, con el controlador del elemento de procesamiento, si el valor intermedio se reenvía a la etapa de ejecución anterior; y

en el que determinar si el valor intermedio debe almacenarse comprende determinar, con el controlador del elemento de procesamiento, si el valor intermedio debe almacenarse en el GPR.

4. El procedimiento según la reivindicación 1, en el que recibir la indicación comprende recibir un indicador asociado con el valor intermedio que indica si el valor intermedio debe almacenarse en el GPR.

5. El procedimiento según la reivindicación 1, en el que, cuando la indicación indica que el valor intermedio no debe almacenarse en el GPR, la indicación indica el último uso del valor intermedio en la pluralidad de sub-instrucciones.

6. Un dispositivo para procesar datos, comprendiendo el dispositivo:

65 una unidad de procesamiento que comprende un núcleo, comprendiendo el núcleo una memoria de instrucciones, un registro de uso general (GPR) y al menos un elemento de procesamiento, comprendiendo el al menos un elemento de procesamiento un controlador y una estructura de canal que tiene una pluralidad de etapas de ejecución, en el que el controlador está configurado para:

- 5 recibir una pluralidad de sub-instrucciones de una instrucción de un programa desde la memoria de instrucciones del núcleo, en la que la instrucción define una pluralidad de operaciones, en la que cada una de las sub-instrucciones define una operación constituyente de la pluralidad de operaciones y en la que las sub-instrucciones se ejecutan mediante la pluralidad de etapas de ejecución de la estructura de canal;
- 10 recibir una indicación de al menos un valor intermedio generado por la ejecución de una de las sub-instrucciones en una de la pluralidad de etapas de ejecución que indica si el valor intermedio debe almacenarse en el GPR;
- 15 determinar si el valor intermedio se reenvía a una etapa de ejecución anterior desde otra etapa de ejecución que ejecutó la sub-instrucción que generó el valor intermedio, en el que la etapa de ejecución anterior y la otra etapa de ejecución que generó el valor intermedio están dispuestas en la estructura de canal;
- 20 determinar si el valor intermedio debe almacenarse en el GPR en base al menos en parte a la indicación recibida y a la determinación de si el valor intermedio se reenvía a la etapa de ejecución anterior; y uno de:
- 25 evitar el almacenamiento del valor intermedio en el GPR si la indicación recibida indica que el valor intermedio no debe almacenarse en el GPR y se determina que el valor intermedio se reenvía a la etapa de ejecución anterior, almacenar el valor intermedio en el GPR si la indicación recibida indica que el valor intermedio no debe almacenarse en el GPR y se determina que el valor intermedio no se reenvía a la etapa de ejecución anterior, o
- 30 almacenar el valor intermedio en el GPR si la indicación recibida indica que el valor intermedio debe almacenarse en el GPR.
- 35 **7.** El dispositivo según la reivindicación 6, en el que la unidad de procesamiento comprende una unidad de procesamiento de gráficos, en el que el núcleo comprende un núcleo de sombreado y en el que el programa comprende un programa de sombreado.
- 8.** El dispositivo según la reivindicación 6, en el que el núcleo incluye una pluralidad de elementos de procesamiento, que incluyen el al menos un elemento de procesamiento, y en el que cada uno de los elementos de procesamiento está configurado para ejecutar las mismas sub-instrucciones de la misma instrucción al mismo tiempo.
- 40 **9.** El dispositivo según la reivindicación 6, en el que la pluralidad de etapas de ejecución está configurada para ejecutar las sub-instrucciones para generar un valor final, en el que el valor final se genera a partir del valor intermedio y en el que el controlador está configurado para almacenar el valor final en el GPR.
- 10.** El dispositivo según la reivindicación 6, en el que, para recibir la indicación, el controlador está configurado para recibir un indicador asociado con el valor intermedio que indica si el valor intermedio debe almacenarse en el GPR.
- 45 **11.** El dispositivo según la reivindicación 6, en el que, cuando la indicación indica que el valor intermedio no se almacena en el GPR, la indicación indica el último uso del valor intermedio en la pluralidad de sub-instrucciones.
- 50 **12.** El dispositivo según la reivindicación 6, en el que el dispositivo comprende un dispositivo de comunicación inalámbrica.
- 13.** Un medio de almacenamiento legible por ordenador que tiene instrucciones almacenadas en el mismo que, cuando se ejecutan por el dispositivo de cualquiera de las reivindicaciones 6 a 12, hacen que el dispositivo realice el procedimiento de acuerdo con una cualquiera de las reivindicaciones 1 a 5.
- 55

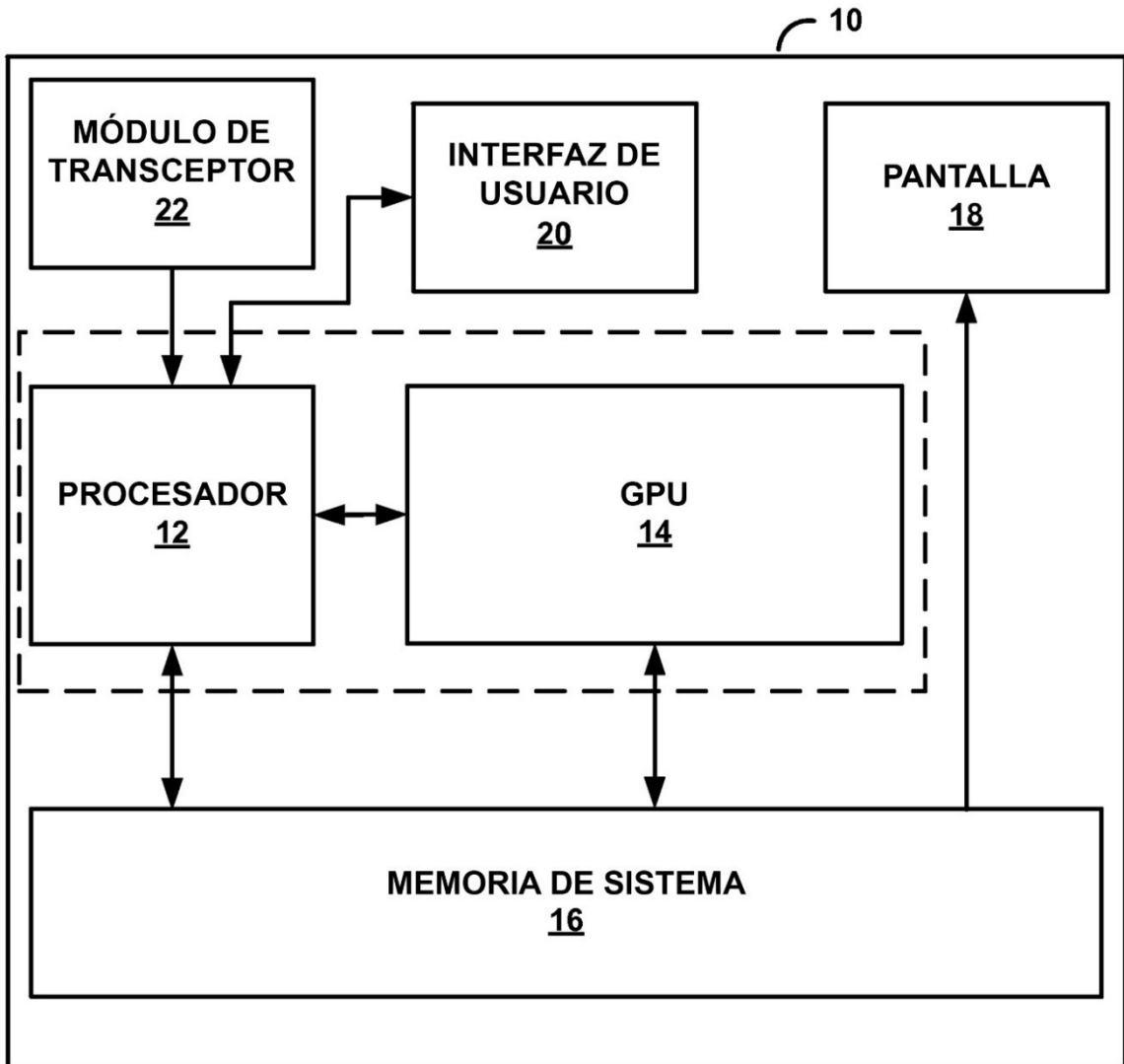


FIG. 1

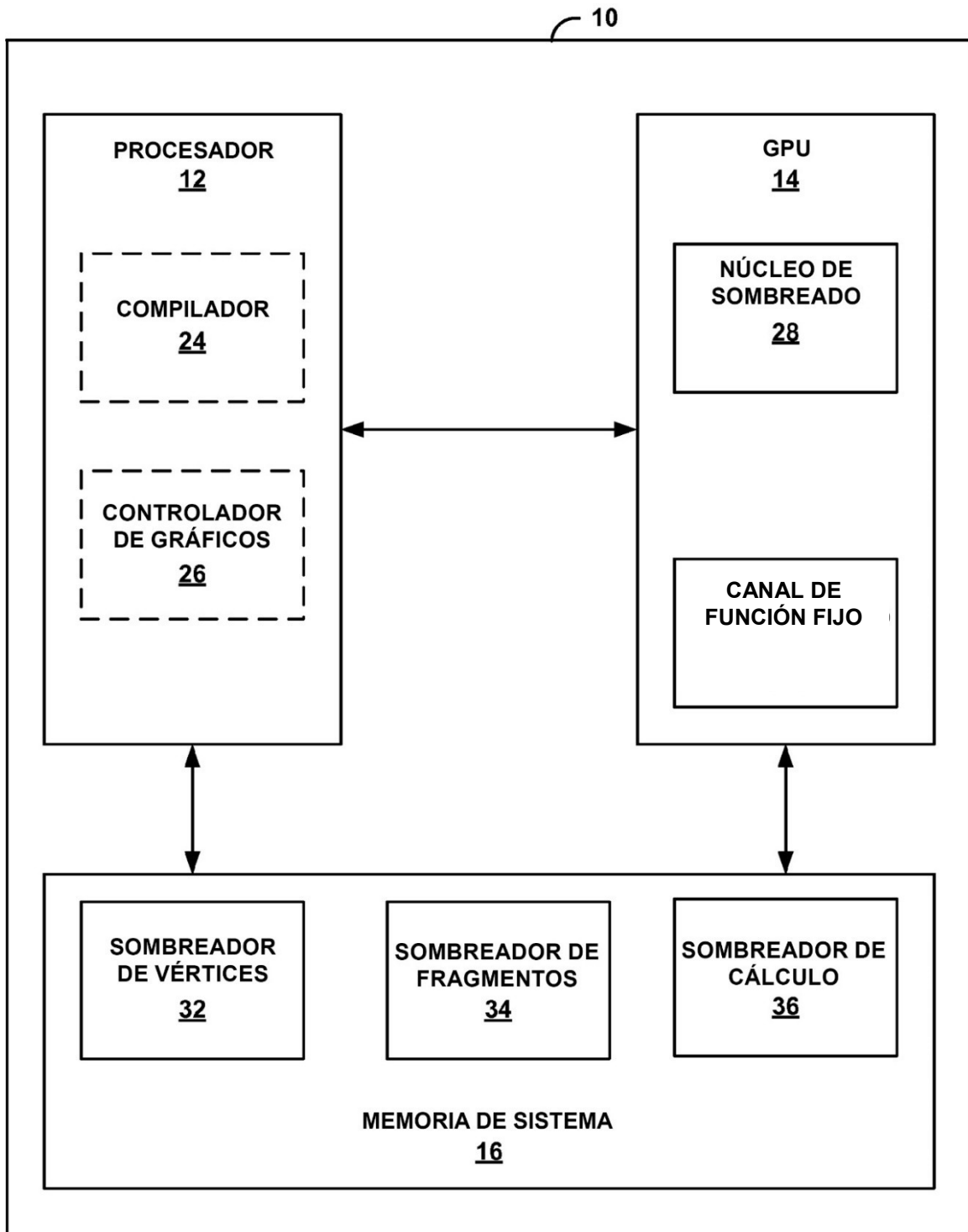


FIG. 2

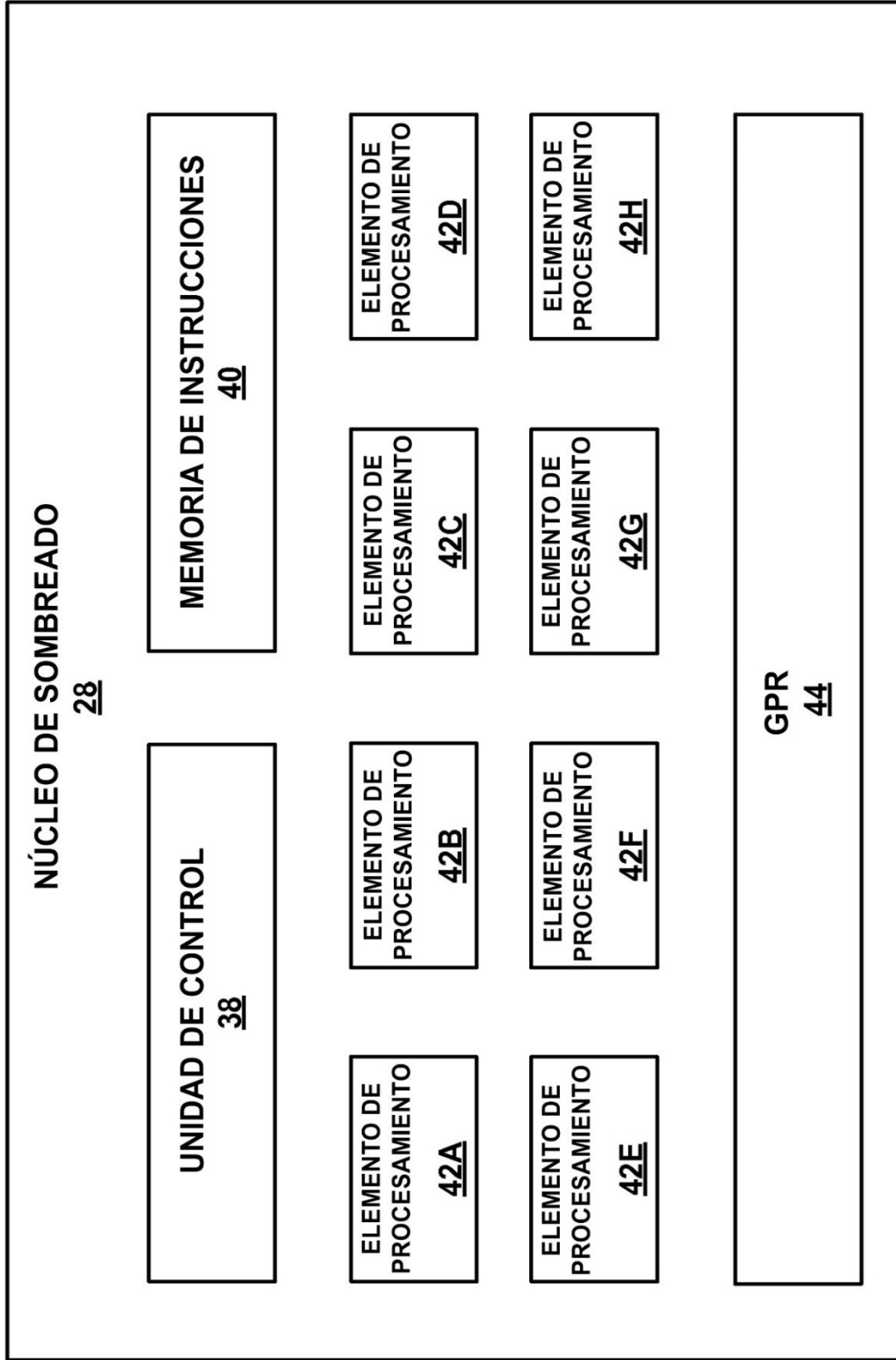


FIG. 3

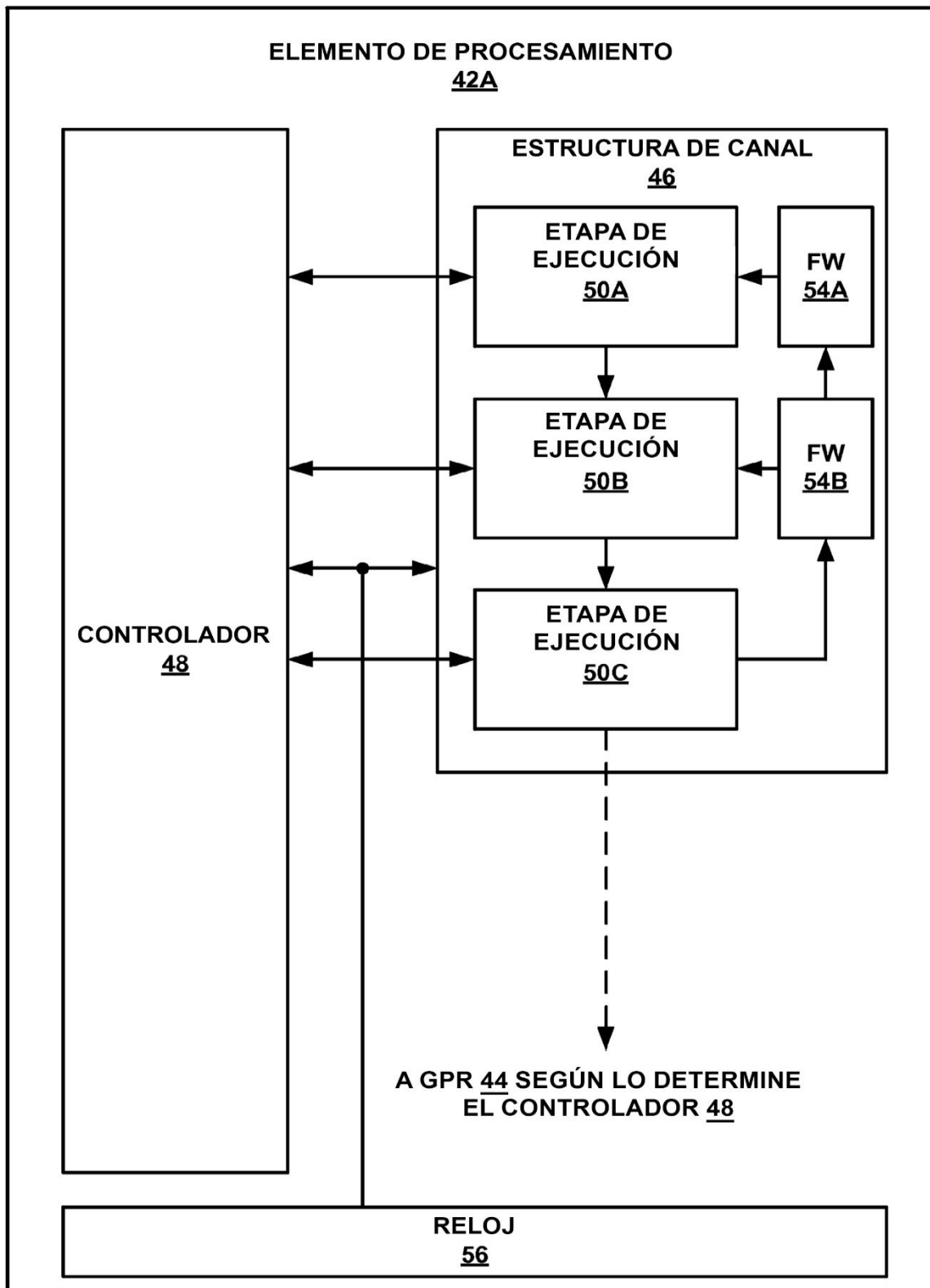


FIG. 4

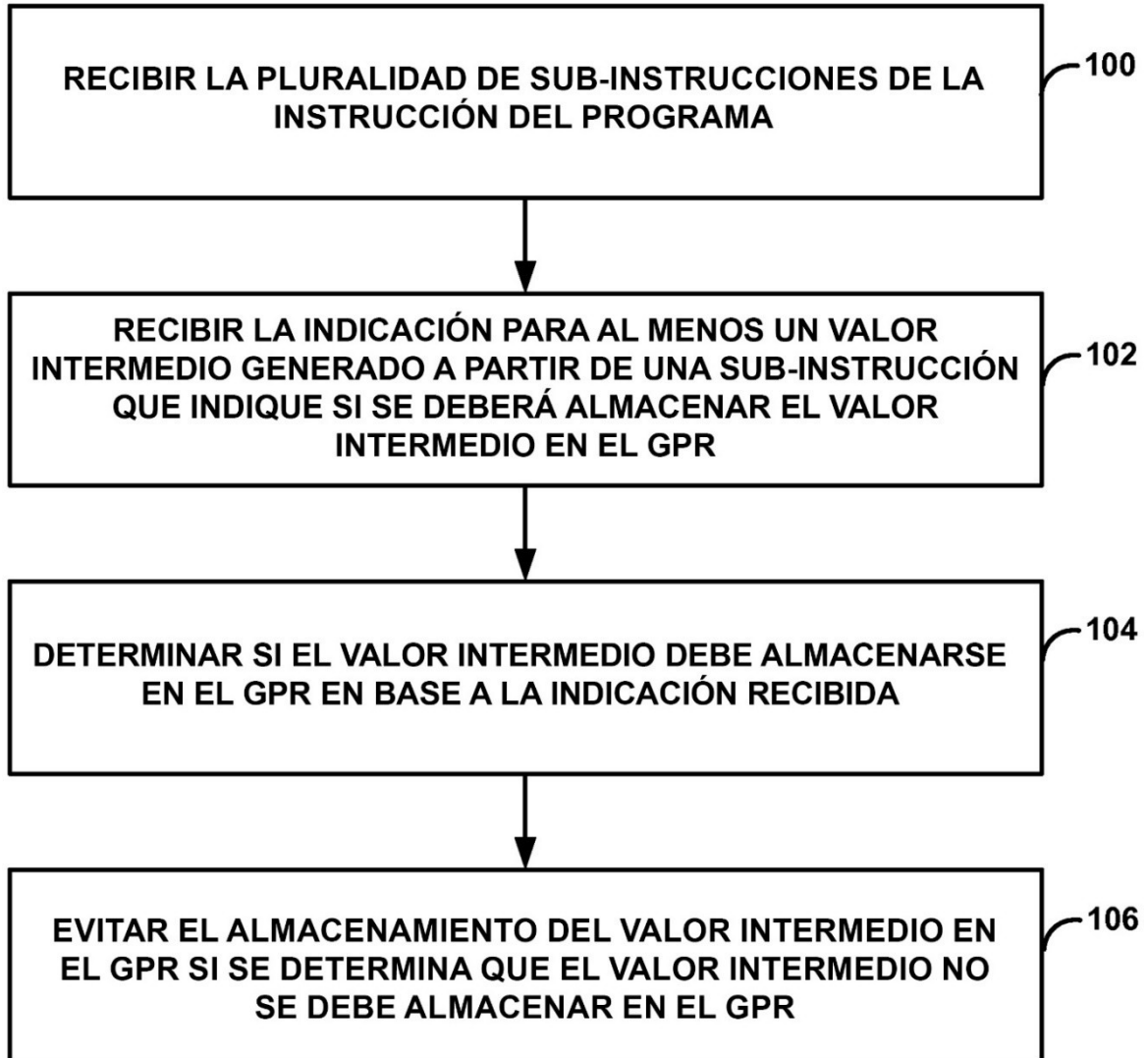


FIG. 5

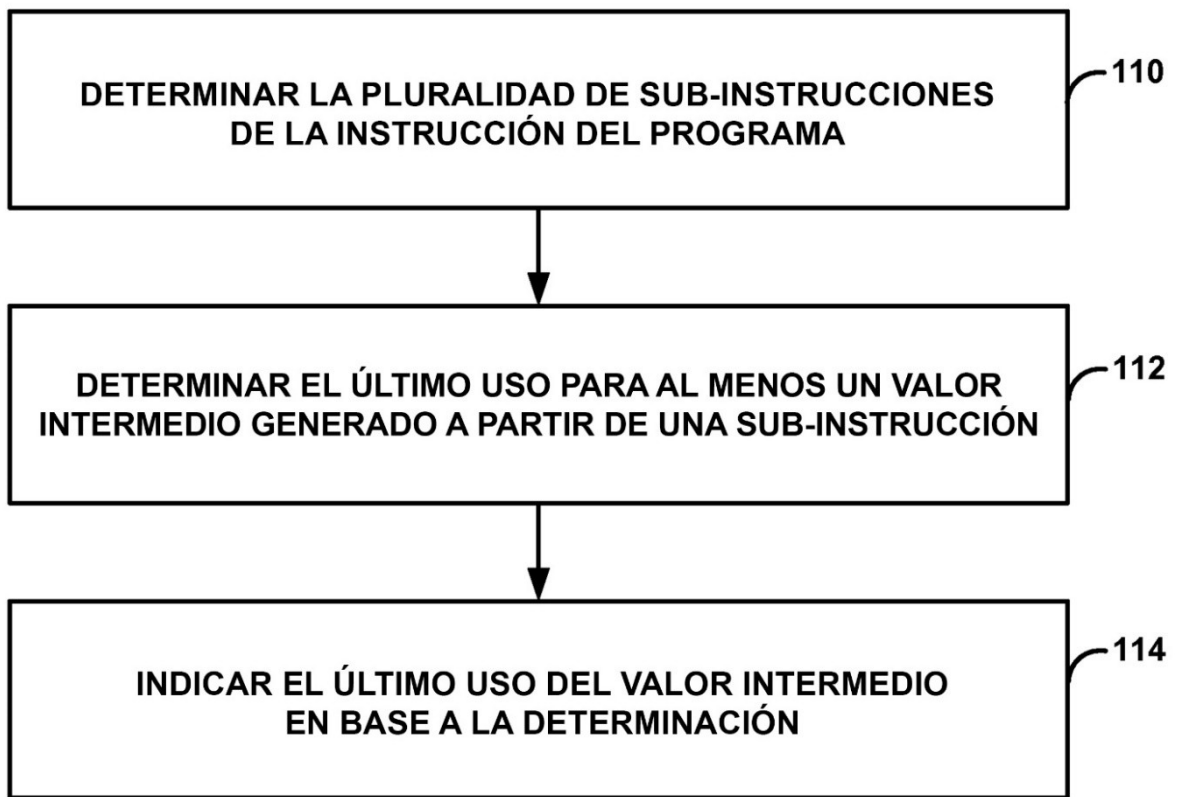


FIG. 6