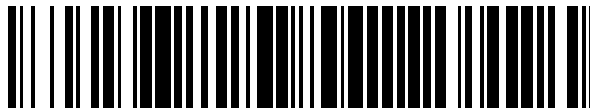


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 777 827**

51 Int. Cl.:

G06F 9/30 (2008.01)

G06F 9/38 (2008.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **09.01.2017 PCT/US2017/012748**

87 Fecha y número de publicación internacional: **24.08.2017 WO17142642**

96 Fecha de presentación y número de la solicitud europea: **09.01.2017 E 17701772 (0)**

97 Fecha y número de publicación de la concesión europea: **11.12.2019 EP 3417369**

54 Título: **Predicados uniformes en sombreadores para unidades de procesamiento de gráficos**

30 Prioridad:

19.02.2016 US 201615048599

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

06.08.2020

73 Titular/es:

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

72 Inventor/es:

**GRUBER, ANDREW EVAN;
ARGADE, PRAMOD VASANT y
WU, JING**

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 777 827 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

Predicados uniformes en sombreadores para unidades de procesamiento de gráficos

5 **CAMPO TÉCNICO**

[0001] La divulgación se refiere a procesamiento de múltiples subprocesos y, más particularmente, a la ejecución de técnicas en un único sistema de procesamiento de datos de múltiples instrucciones (SIMD).

10 **ANTECEDENTES**

[0002] Un sistema de procesamiento de una única instrucción, múltiples datos (SIMD) es una clase de sistemas de informáticos en paralelo que incluye múltiples elementos de procesamiento que ejecutan la misma instrucción en múltiples conjuntos de datos. Un sistema SIMD puede ser un ordenador o un subsistema independiente de un sistema informático. Por ejemplo, se pueden usar una o más unidades de ejecución SIMD en una unidad de procesamiento de gráficos (GPU) para implementar una unidad de sombreado programable que soporte sombreado programable. Un sistema de procesamiento SIMD permite múltiples subprocesos de ejecución para que un programa se ejecute sincronamente en los múltiples elementos de procesamiento de manera paralela, lo cual aumenta el rendimiento de los programas en los que el mismo conjunto de operaciones debe realizarse en múltiples conjuntos de datos. Una instrucción particular que se ejecuta en un elemento de procesamiento SIMD particular se conoce como subproceso o fibra. Un grupo de subprocesos se puede denominar onda o urdimbre.

[0003] Las unidades de procesamiento, como las GPU, incluyen elementos de procesamiento y un registro de propósito general (GPR) que almacena datos para la ejecución de una instrucción. En algunos ejemplos, un elemento de procesamiento ejecuta instrucciones para procesar un elemento de datos, y los elementos de procesamiento respectivos almacenan los datos del elemento o los datos resultantes del elemento del procesamiento en el GPR. Un elemento de datos puede ser la unidad base en la que se produce el procesamiento. Por ejemplo, en el procesamiento de gráficos, un vértice de una primitiva es un ejemplo de un elemento, y un píxel es otro ejemplo de un elemento. Hay datos gráficos asociados con cada vértice y píxel (por ejemplo, coordenadas, valores de color, etc.).

[0004] Puede haber múltiples elementos de procesamiento dentro de un núcleo de procesador del elemento de procesamiento, lo cual permite la ejecución paralela de una instrucción (por ejemplo, múltiples elementos de procesamiento ejecutan la misma instrucción al mismo tiempo). En algunos casos, cada uno de los elementos de procesamiento almacena datos de un elemento en el GPR y lee los datos del elemento desde el GPR incluso si los datos son los mismos para varios elementos. Us2011072248 divulga una instrucción de ramificación unánime, en la que cada subproceso de un grupo de subprocesos acepta tomar la ramificación basándose en el predicado o en un código de condición.

40 **SUMARIO**

[0005] En general, la divulgación describe técnicas para la determinación de los bits precedentes para un núcleo de sombreado y/o núcleo de procesamiento SIMD de una GPU. Los bits de predicado indican qué parte(s) de un conjunto de instrucciones ejecutará o no un subproceso de un núcleo de procesamiento SIMD al alcanzar una condición de ramificación en el conjunto de instrucciones. En particular, esta divulgación describe técnicas mediante las cuales un único conjunto de bits de predicado puede ser almacenado y utilizado por todos los subprocesos de una urdimbre en un núcleo de procesamiento SIMD cuando todos los subprocesos se ramifican de la misma manera. Dicha técnica permite la simplificación y el ahorro de energía durante el uso de bits de predicado por subproceso.

[0006] En un ejemplo, la divulgación describe un procedimiento de procesamiento de datos en una unidad de procesamiento de gráficos, comprendiendo el procedimiento recibir una indicación de que todos los subprocesos de una urdimbre en una unidad de procesamiento de gráficos (GPU) están para ejecutar una misma ramificación en un primer conjunto de instrucciones, almacenar uno o más bits de predicado en una memoria como un único conjunto de bits de predicado, en el que el único conjunto de bits de predicado se aplica a todos los subprocesos en la urdimbre, y ejecutar una parte del primer conjunto de instrucciones de acuerdo con el único conjunto de bits de predicado.

[0007] En otro ejemplo, la divulgación describe un aparato para el procesamiento de datos, comprendiendo el aparato una memoria configurada para almacenar un primer conjunto de instrucciones, y un primer procesador configurado para recibir una indicación de que todos los subprocesos de una urdimbre en el primer procesador deben ejecutar una misma ramificación en un primer conjunto de instrucciones, almacenar uno o más bits de predicado en un registro como un único conjunto de bits de predicado, en el que el único conjunto de bits de predicado se aplica a todos los subprocesos en la urdimbre, y ejecutar una parte del primer conjunto de instrucciones de acuerdo con el único conjunto de bits de predicado.

[0008] En otro ejemplo, la divulgación describe un aparato para procesamiento de datos, comprendiendo el aparato medios para recibir una indicación de que todos los subprocesos de una urdimbre en una unidad de procesamiento de gráficos (GPU) deben ejecutar una misma ramificación en un primer conjunto de instrucciones, medios para

almacenar uno o más bits de predicado en una memoria como un único conjunto de bits de predicado, en el que el único conjunto de bits de predicado se aplica a todos los subprocesos en la urdimbre, y medios para ejecutar una parte del primer conjunto de instrucciones en de acuerdo con el único conjunto de bits de predicado.

5 **[0009]** En otro ejemplo, la divulgación describe un medio de almacenamiento legible por ordenador que tiene instrucciones almacenadas en el mismo que cuando se ejecutan hacen que uno o más procesadores reciban una indicación de que todos los subprocesos de una urdimbre en una unidad de procesamiento de gráficos (GPU) deben ejecutar una misma ramificación en un primer conjunto de instrucciones, almacenar uno o más bits de predicado en una memoria como un único conjunto de bits de predicado, en el que el único conjunto de bits de predicado se aplica
10 a todos los subprocesos en la urdimbre, y ejecutar una parte del primer conjunto de instrucciones de acuerdo con el único conjunto de bits de predicado.

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

15 **BREVE DESCRIPCIÓN DE LOS DIBUJOS**

[0011]

20 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. 2 es un diagrama de bloques que ilustra los componentes del dispositivo ilustrado en la FIG. 1 en mayor detalle.

25 La FIG. 3 es un diagrama conceptual que ilustra un ejemplo de un conjunto de instrucciones.

La FIG. 4 es un diagrama conceptual que ilustra un ejemplo de almacenamiento de datos en un registro de propósito general (GPR) de un núcleo de sombreado de una GPU.

30 La FIG. 5 es un diagrama de bloques de un núcleo de sombreado de ejemplo de la GPU de la FIG. 1 en mayor detalle.

35 La FIG. 6 es un diagrama de bloques de otro núcleo de sombreado de ejemplo de la GPU de la FIG. 1 en mayor detalle.

La FIG. 7 es un diagrama de flujo que ilustra una técnica de ejemplo de datos de procesamiento en una GPU.

40 **DESCRIPCIÓN DETALLADA**

[0012] Entre 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), se 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).

50 **[0013]** La GPU puede diseñarse con una estructura de una única instrucción, múltiples datos (SIMD). En la estructura SIMD, un núcleo de sombreado (o más en general, un núcleo de sombreado SIMD) 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 un subproceso o una fibra. Un grupo de subprocesos se puede denominar onda o urdimbre.
55 Todos los elementos de procesamiento juntos que ejecutan una urdimbre pueden denominarse una unidad de procesamiento de vectores, donde cada línea (por ejemplo, elemento de procesamiento) del vector ejecuta un 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
60 elementos de procesamiento. De esta manera, la estructura SIMD permite que la GPU realice muchas tareas en paralelo (por ejemplo, al mismo tiempo).

[0014] Un núcleo de procesamiento SIMD puede incluir múltiples elementos de procesamiento, con cada elemento de procesamiento ejecutando un subproceso. Cada elemento de procesamiento ejecuta el mismo conjunto de instrucciones en paralelo. En algunas situaciones, un conjunto de instrucciones puede incluir una instrucción de ramificación condicional (por ejemplo, una instrucción if-then-else). A medida que cada elemento de procesamiento

opera en diferentes datos, diferentes subprocesos en la misma urdimbre pueden ramificarse de diferentes maneras. A eso se le dan los diferentes datos de entrada para cada subproceso, la condición de si tomar la ramificación "then" de la instrucción condicional o la ramificación "else" del condicional puede ser diferente para cada subproceso.

5 **[0015]** En algunos ejemplos, el flujo de control para el núcleo de procesamiento SIMD se lleva a cabo en base "por subproceso" utilizando bits de predicado que indican cómo cada subproceso (línea) del vector SIMD (por ejemplo, cómo cada elemento de procesamiento) debe ramificar (es decir, en el caso el único conjunto de instrucciones incluye una instrucción de ramificación). Debido a que los subprocesos tienen datos independientes, estos predicados en general no tienen correlación entre sí. Sin embargo, eso no siempre es cierto, y para los casos en que el mismo predicado se aplica a todos los subprocesos en una urdimbre, el uso de predicados independientes es un desperdicio.

10 **[0016]** Esta divulgación describe procedimientos, técnicas y dispositivos que utilizan un conjunto de predicados uniformes en los que un único conjunto de bits de predicado se aplica a toda una urdimbre. Tales predicados uniformes pueden usarse cuando se determina que todos los subprocesos de una urdimbre se ramifican de la misma manera. De esta forma, se puede lograr un ahorro de energía ya que solo se realiza una única prueba cuando se utilizan predicados uniformes, en lugar de asignar bits de predicado por subproceso.

15 **[0017]** Además, esta descripción también describe la adición de una unidad de tratamiento escalar en la GPU junto a un núcleo SIMD de la GPU. La unidad de procesamiento escalar se puede usar para ejecutar un conjunto de instrucciones que se ha determinado que se ramifican de la misma manera para todos los subprocesos de una urdimbre. De esta manera, una unidad de procesamiento escalar puede usar el conjunto de predicados uniforme para ejecutar el conjunto de instrucciones con ramificación uniforme, mientras que la unidad de procesamiento de vectores del núcleo SIMD puede usarse para procesar otro conjunto de instrucciones en paralelo con la unidad de procesamiento escalar.

20 **[0018]** 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 para la ejecución de subprocesos en un núcleo de sombreado con predicados uniformes. La FIG. 1 ilustra el dispositivo 10, entre cuyos ejemplos se incluyen, entre otros, dispositivos de vídeo tales como reproductores de medios, descodificadores, dispositivos de comunicación inalámbricos, tales como teléfonos móviles, asistentes digitales personales (PDA), ordenadores de escritorio, ordenadores portátiles, consolas de videojuegos, unidades de videoconferencia, dispositivos informáticos tipo tablet y similares.

25 **[0019]** 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 del 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, como un sistema en chip (SoC). En algunos ejemplos, el procesador 12 y la GPU 14 pueden estar alojados en diferentes circuitos integrados (por ejemplo, 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.

30 **[0020]** Entre los ejemplos del procesador 12 y de la GPU 14 se incluyen, entre otros, uno o más procesadores de señales digitales (DSP), microprocesadores de uso general, circuitos integrados específicos de la aplicación (ASIC), matrices lógicas programables in situ (FPGA) u otra circuitería lógica integrada o discreta equivalente. 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).

35 **[0021]** Con fines ilustrativos, las técnicas descritas en esta divulgación se describen con la GPU 14. Sin embargo, las técnicas descritas en esta divulgación no están limitadas de ese modo. 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.

40 **[0022]** El procesador 12 puede ejecutar diversos tipos de aplicaciones. Entre los ejemplos de las aplicaciones se incluyen sistemas operativos, navegadores de Internet, aplicaciones de correo electrónico, hojas de cálculo, videojuegos u otras aplicaciones que generan objetos visibles para su visualización. La memoria del sistema 16 puede almacenar instrucciones para la ejecución de una o más aplicaciones. La ejecución de una aplicación en el procesador 12 hace 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.

[0023] A modo de ejemplo, la ejecución de una aplicación en el procesador 12 hace que el procesador 12 produzca vértices de primitivas, en el que la interconexión de primitivas en los respectivos vértices forma un objeto gráfico. En este ejemplo, los datos gráficos que produce el procesador 12 son los datos de los atributos de los vértices. Por ejemplo, la aplicación que se ejecuta en el procesador 12 puede generar valores de color, valores de opacidad, coordenadas, etc. para los vértices, que son todos ejemplos de atributos de los vértices. Puede haber también atributos adicionales, y en algunos ejemplos, la aplicación no necesita producir todos los atributos de ejemplo. En general, las técnicas son extensibles a tipos de datos (por ejemplo, contadores) que no sean datos de atributos, y las técnicas no deben considerarse limitadas a datos de atributos ni limitadas a ejemplos de datos de atributos como valores de color, valores de opacidad, coordenadas, etc.

[0024] En algunos ejemplos no relacionados con gráficos, el procesador 12 puede generar datos que sean más adecuados para procesarse mediante la GPU 14. Dichos datos no es necesario que sean para fines de gráficos o de visualización. Por ejemplo, el procesador 12 puede enviar datos sobre qué operaciones de matriz deben ser realizadas por la GPU 14, y la GPU 14 puede a su vez realizar las operaciones de matriz.

[0025] 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 hacer que la GPU 14 realice operaciones no relacionadas con los gráficos.

[0026] 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 u 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.

[0027] El dispositivo 10 también puede incluir la pantalla 18, la interfaz de usuario 20 y el módulo de transceptor 22. El dispositivo 10 puede incluir módulos o unidades adicionales que no se muestran en la FIG. 1 para una mayor 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.

[0028] 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. Entre los ejemplos de interfaz de usuario 20 se incluyen, entre otros, 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 se puede incorporar como parte de la pantalla 18. El módulo de transceptor 22 puede incluir circuitos 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, desmoduladores, amplificadores y otros circuitos de este tipo para la comunicación alámbrica o inalámbrica.

[0029] La memoria del sistema 16 puede ser la memoria para el dispositivo 10. La memoria del sistema 16 puede comprender uno o más medios de almacenamiento legibles por ordenador. Entre los ejemplos de memoria del sistema 16 se incluyen, entre otros, una memoria de acceso aleatorio (RAM), una memoria de solo lectura programable y borrrable 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.

[0030] En algunos aspectos, la memoria del sistema 16 puede incluir instrucciones que hagan 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 del sistema 16 puede ser un medio de almacenamiento legible por ordenador que tenga instrucciones almacenadas en el mismo que, cuando se ejecuten, hagan que uno o más procesadores (por ejemplo, el procesador 12 y la GPU 14) realicen diversas funciones.

[0031] En algunos ejemplos, la memoria del sistema 16 puede ser un medio de almacenamiento no transitorio. La expresión «no transitorio» puede indicar que el medio de almacenamiento no está realizado 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 del sistema 16 es no móvil o que sus contenidos son estáticos. Como ejemplo, la memoria del sistema 16 se

puede retirar del dispositivo 10 y trasladarse a otro dispositivo. Como otro ejemplo, la memoria, sustancialmente similar a la memoria del 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]** Por ejemplo, como se describe en más detalle en otra parte en esta divulgación, la memoria del sistema 16 puede almacenar el código de un compilador que se ejecuta en el procesador 12 que lleva a cabo técnicas de uno o más ejemplos descritos en la presente divulgación. La memoria del sistema 16 también puede almacenar código para programas de sombreado (por ejemplo, un sombreador de vértices, un sombreador de píxeles o fragmentos, un sombreador de cómputo, etc.) que se ejecutan en un núcleo de sombreado de la GPU 14.

10 **[0033]** El elemento gráfico de plazo se utiliza en esta descripción para referirse a una unidad de base sobre la que GPU 14 realiza el procesamiento en paralelo. La GPU 14 puede procesar una pluralidad de elementos gráficos en paralelo (por ejemplo, al mismo tiempo). Por ejemplo, un sombreador de vértices puede procesar un vértice, y la GPU 14 puede ejecutar una pluralidad de instancias del sombreador de vértices en paralelo para procesar una pluralidad de vértices al mismo tiempo. De manera similar, un sombreador de píxeles o fragmentos puede procesar un píxel de una pantalla, y la GPU 14 puede ejecutar una pluralidad de instancias del sombreador de píxeles en paralelo para procesar una pluralidad de píxeles de la pantalla al mismo tiempo. Un vértice y un píxel son ejemplos de un elemento gráfico.

15 **[0034]** Para aplicaciones no-gráficas relacionadas, el elemento gráfico de término también se refiere a la unidad más pequeña en la que GPU 14 realiza el procesamiento. Sin embargo, dicho procesamiento puede no estar relacionado con los gráficos. En consecuencia, el término "elemento de gráficos" se refiere a elementos que una unidad de procesamiento de gráficos (por ejemplo, GPU 14) o una unidad de procesamiento de gráficos de uso general (por ejemplo, ejemplos en los que GPU 14 funciona como una GPGPU) debe procesar en paralelo. El elemento gráfico se puede usar para fines gráficos o no gráficos.

20 **[0035]** Como se explicará en más detalle a continuación, de acuerdo con diversos ejemplos de la divulgación, la GPU 14 puede estar configurada para recibir una indicación de que todos los subprocesos de una urdimbre en la GPU deben ejecutar una misma ramificación en un primer conjunto de instrucciones, almacenar uno o más bits de predicado en una memoria como un único conjunto de bits de predicado, en el que el único conjunto de bits de predicado se aplica a todos los subprocesos en la urdimbre, y ejecutar una parte del primer conjunto de instrucciones de acuerdo con el único conjunto de bits de predicado.

25 **[0036]** 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 24, que incluye el registro de propósito general (GPR) 26, un registro de predicados uniforme (UPR) 92 y el canal de función fija 30. El núcleo de sombreado 24 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.

30 **[0037]** El software y/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 24 de la GPU 14. Aunque se ilustra solamente un núcleo de sombreado 24, en algunos ejemplos, la GPU 14 puede incluir uno o más núcleos de sombreado similares al núcleo de sombreado 24. El canal de función fija 30 incluye las unidades de función fija. El núcleo de sombreado 24 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 24 que reciban datos de una unidad de función fija del canal de función fija 30 y envíen datos procesados a otra unidad de función fija del canal de función fija 30.

35 **[0038]** Los programas de sombreado proporcionan a los usuarios y/o desarrolladores 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.

40 **[0039]** Entre los ejemplos de los programas de sombreado se incluyen el programa de sombreado de vértices 32, el programa de sombreado de fragmentos 34 y el programa de sombreado de cálculo 36. El programa de sombreado de vértices 32 y el programa de sombreado de fragmentos 34 pueden ser programas de sombreado para tareas relacionadas con gráficos, y el programa de sombreado 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 programa de sombreado de vértices 32 y el programa de sombreado de fragmentos 34. En algunos ejemplos, solo se pueden usar programas de sombreado no relacionados con gráficos como el programa de sombreado de cálculo 36. Hay ejemplos adicionales de programas de sombreado tales como sombreadores de geometría, que no se describen en aras de la brevedad.

65

[0040] El controlador de gráficos 40 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 programa de sombreado de vértices 32, el programa de sombreado de fragmentos 34 y el programa de sombreado de cálculo 36) se pueden configurar de acuerdo con la misma API que el controlador de gráficos 40. Aunque no se ilustra, la memoria del sistema 16 puede almacenar el código para el controlador de gráficos 40 que el procesador 12 recupera de la memoria del sistema 16 para su ejecución. El controlador de gráficos 40 se ilustra en una casilla discontinua para indicar que el controlador de gráficos 40 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 40 puede implementarse como hardware en el procesador 12.

[0041] El controlador de gráficos 40 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 40.

[0042] 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 del sistema 16, y el controlador de gráficos 40 puede indicar a la GPU 14 cuándo recuperar los datos gráficos, desde dónde recuperar los datos gráficos en la memoria del 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 24 ejecute el programa de sombreado de vértices 32 y el programa de sombreado 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 40 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 40 puede formar el enlace entre el procesador 12 y la GPU 14.

[0043] El controlador de gráficos 40 se puede configurar de acuerdo con una API; aunque el controlador de gráficos 40 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 40 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 40 puede configurarse de acuerdo con la API de OpenGL. Otros ejemplos de API incluyen la familia DirectX de API de Microsoft Corporation.

[0044] En algunos ejemplos, la memoria del sistema 16 puede almacenar el código fuente para uno o más del programa de sombreado de vértices 32, el programa de sombreado de fragmentos 34 y el programa de sombreado de cálculo 36. En estos ejemplos, el compilador 38 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 mediante el núcleo de sombreado 24 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 24). En algunos ejemplos, el compilador 38 puede compilar previamente los programas de sombreado y almacenar el objeto o código intermedio de los programas de sombreado en la memoria del sistema 16.

[0045] El núcleo de sombreado 24 puede configurarse para ejecutar muchas instancias de la misma instrucción del mismo programa de sombreado en paralelo. Por ejemplo, el controlador de gráficos 40 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 programa de sombreado de vértices 32 para procesar los valores de vértice de los vértices. En este ejemplo, el núcleo de sombreado 24 puede ejecutar múltiples instancias del programa de sombreado de vértices 32 y hacerlo ejecutando una instancia del programa de sombreado de vértices 32 en un elemento de procesamiento del núcleo de sombreado 24 para cada uno de los vértices.

[0046] Cada elemento de procesamiento del núcleo de sombreado 24 puede ejecutar la misma instrucción del programa de sombreado 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 programa de sombreado de vértices 32, donde un subproceso se refiere a una instrucción del programa de sombreado de vértices 32 que está procesando un vértice particular. De esta manera, el núcleo de sombreado 24 puede ejecutar muchas instancias del programa de sombreado de vértices 32 para procesar valores de vértice de una pluralidad de vértices en paralelo (por ejemplo, al mismo tiempo).

[0047] El núcleo de sombreado 24 puede ejecutar de manera similar muchas instancias del programa de sombreado 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 24 puede configurarse en una estructura de una única instrucción, múltiples datos (SIMD). Para facilitar la descripción, se describe lo siguiente con respecto a un programa de sombreado genérico, entre cuyos ejemplos se incluyen el programa de sombreado de vértices 32, el programa de sombreado de

fragmentos 34, el programa de sombreado de cálculo 36 y otros tipos de sombreadores tales como los sombreadores de geometría.

5 **[0048]** La memoria del sistema 16 puede almacenar el código para el compilador 38 que el procesador 12 recupera de la memoria del sistema 16 para su ejecución. En el ejemplo de la FIG. 2, el compilador 38 se ilustra en una casilla discontinua para indicar que el compilador 38 es un software, que se ejecuta en hardware (por ejemplo, el procesador 12), en este ejemplo. Sin embargo, alguna funcionalidad del compilador 38 puede implementarse como hardware en el procesador 12, en algunos ejemplos.

10 **[0049]** De acuerdo con las técnicas de la presente divulgación, el procesador 12 puede estar configurado para determinar si un conjunto particular de instrucciones de un programa de sombreado (por ejemplo, el programa de sombreado de vértices 32, el programa de sombreado de fragmentos 34, el programa de sombreado de cálculo 36, o cualquier otro programa de sombreado) incluye instrucciones de ramificación. En este ejemplo, el conjunto de instrucciones puede ser un conjunto particular de instrucciones para ejecutarse como una urdimbre (por ejemplo, usando múltiples subprocesos en paralelo) usando el núcleo de sombreado 24 (por ejemplo, un núcleo de procesamiento SIMD). El compilador 38 que se ejecuta en el procesador 12 puede configurarse para determinar que el conjunto de instrucciones para la urdimbre incluye la instrucción de ramificación.

20 **[0050]** La FIG. 3 es un diagrama conceptual que muestra un conjunto de instrucciones 37 entre las que se incluye la instrucción de ramificación 39. En este ejemplo, la instrucción de ramificación 39 es una instrucción de ramificación if-then-else. En términos generales, el término "if (condición)" determina si una condición particular es verdadera (por ejemplo, comparando un valor con otro valor). Independientemente de cómo se determine la condición, si la condición es verdadera, un elemento de procesamiento en el núcleo de sombreado 24 ejecuta una primera parte 41 del conjunto de instrucciones 37 (por ejemplo, la parte de las instrucciones asociadas con la ramificación "entonces"). Si la condición no es verdadera, un elemento de procesamiento ejecuta una segunda parte 43 del conjunto de instrucciones 37 (por ejemplo, la parte de las instrucciones asociadas con la ramificación 'else').

30 **[0051]** Con referencia de nuevo a la FIG. 2, la GPU 14 y/o el compilador 38 pueden configurarse para determinar cómo cada subproceso de la urdimbre tomará la instrucción de ramificación. Basándose en esta determinación, la GPU 14 y/o el compilador 38 pueden asignar bits de predicado para cada subproceso de la urdimbre para indicar cómo tomará el subproceso la instrucción de ramificación. En efecto, los bits de predicado indican qué partes de un conjunto de instrucciones serán ejecutadas por el subproceso particular. Por ejemplo, los bits de predicado pueden indicar si la primera parte 41 de la FIG. 3 o segunda parte 43 de la FIG. 3 deben ser ejecutadas.

35 **[0052]** En algunos ejemplos, los bits de predicado pueden especificar qué partes de un conjunto de instrucciones ejecutar. En otros ejemplos, los bits de predicado indican qué partes del conjunto de instrucciones no ejecutar. En cualquier caso, cada subproceso de una urdimbre ejecuta el conjunto de instrucciones en paralelo. Si un subproceso particular no ejecuta una parte particular del conjunto de instrucciones como lo indican los bits de predicado, ese subproceso particular no realiza ninguna acción (por ejemplo, ejecuta instrucciones sin operación) mientras que otros subprocesos en la urdimbre ejecutan esa parte particular del conjunto de instrucciones.

45 **[0053]** De acuerdo con las técnicas de la presente divulgación, el procesador 12 (por ejemplo, a través de la ejecución de compilador 38) puede estar configurado además para determinar si todos los subprocesos de una urdimbre tomarán la misma ramificación de una instrucción de ramificación. Si se realiza tal determinación, el compilador 38 puede indicar a la GPU 14 que establezca un único conjunto de bits de predicado uniformes que se pueden usar para cada subproceso en la urdimbre. Es decir, en lugar de tener bits de predicado separados para cada subproceso de una urdimbre, cuando todos los subprocesos toman la misma ramificación de una instrucción de ramificación condicional, solo se almacena un único conjunto de bits de predicado.

50 **[0054]** Por ejemplo, el compilador 38 puede configurarse para almacenar, o hacer que la GPU 14 almacene, el único conjunto de bits de predicado uniformes en UPR 92. El UPR 92 puede ser un registro o cualquier otro tipo de memoria al que pueda acceder cada elemento de procesamiento en el núcleo de sombreado 24 que esté ejecutando un subproceso particular de una urdimbre. El compilador 38 puede determinar que todos los subprocesos de una urdimbre toman la misma ramificación de una instrucción de ramificación condicional de cualquier manera. Por ejemplo, el compilador 38 puede hacer tal determinación en el caso de que los valores de entrada para la urdimbre estén basados en constantes o se acceda desde un registro uniforme de propósito general (uGPR).

60 **[0055]** Como un ejemplo, durante la compilación de un programa de sombreado (por ejemplo, programa de sombreado de vértices 32, programa de sombreado de fragmentos 34, o programa de sombreado de cómputo 36), el compilador 38 puede determinar si un conjunto de instrucciones del programa de sombreado que se ejecutan como una urdimbre requiere acceder a datos uniformes. En este contexto, un programa de sombreado que requiere acceso a datos uniformes significa que cada subproceso del programa de sombreado que se ejecuta en elementos de procesamiento respectivos de núcleo de sombreado 24 solicita los mismos datos. En este ejemplo, el núcleo de sombreado 24 puede recuperar estos mismos datos del uGPR.

65

[0056] Como otro ejemplo, el programa de sombreado puede incluir instrucciones de ramificación (por ejemplo, instrucciones if-then-else). Para alguna ramificación en particular, el programa de sombreado puede incluir una instrucción que establece una o más variables iguales a un valor constante (por ejemplo, una instrucción que establece todas las coordenadas x, y, z y w iguales a 0). En este ejemplo, el compilador 38 puede determinar que el valor constante son datos para una variable de los elementos gráficos que son uniformes en todos los subprocesos de la urdimbre.

[0057] Si el compilador 38 determina que el programa de sombreado requiere acceder a datos uniformes a través de la onda gráfica, el compilador 38 puede incluir una instrucción explícita que indica al núcleo de sombreado 24 que los datos son uniformes a través de la onda gráfica. En respuesta, la GPU 14 puede establecer bits de predicado uniformes en UPR 92. En algunos ejemplos, los datos uniformes pueden almacenarse en una sección de GPR 26 que está asignada para almacenar datos uniformes (es decir, datos a los que todos los subprocesos de urdimbre tienen acceso uniforme). Tal sección de GPR 26 asignada para datos uniformes puede denominarse uGPR.

[0058] Considérese un ejemplo general de código que incluye instrucciones de ramificación. Por ejemplo, en un lenguaje de alto nivel, un ejemplo de una instrucción de ramificación podría ser:

```
If (A = B)
{ <perform action 1> }
Else
{ <perform action 2> }
```

[0059] En un ejemplo, dicho código puede ser compilado por el compilador 38 a:

```
CMP.EQ      PRO <- R0,R1 //R0 is a register that holds 'A', R1 is a register that holds
'B', P0 is the single bit predicate that holds the compare result.
BRA        !PRO, ELSE //Branch instruction to label 'ELSE' if PRO is '0'
          <action 1 >
          BRA        CONTINUE //unconditional branch around 'else' clause
ELSE:      <action 2>
CONTINUE   <rest of program>
```

[0060] En otro ejemplo, podría utilizarse "predicación" en lugar de las instrucciones de "ramificación":

```
CMP.EQ PRO <- R0,R1 //R0 holds 'A', R1 holds 'B', P0 is the single bit predicate
that holds the compare result
(PRO)      <action 1> //only execute <action 1> if PRO == 1
(!PRO)     <action 2> //only execute <action 2> if PRO == 0
<rest of program>
```

[0061] Si el compilador 38 no pudo determinar que los valores de R0 y R1 son "uniformes" entonces se utilizarían los bits de predicado por subproceso. Sin embargo, si el compilador 38 determina que los valores R0 y R1 son uniformes (por ejemplo, si los valores usados provienen de registros constantes, entonces podría usarse el predicado "uniforme" (UP0) en su lugar:

```
CMP.EQ      UPR0 <-C0,C1 //C0 holds 'A', C1 holds 'B', UPR0 is the single bit shared
predicate that holds the compare result.
(UPR0)      <action 1> //only execute <action 1> if UPR0 == 1
(!UPR0)     <action 2> //only execute <action 2> if UPR0 == 0
<rest of program>
```

[0062] La FIG. 4 es un diagrama conceptual que ilustra un ejemplo de almacenamiento de datos en un GPR de un núcleo de sombreado de una GPU. Como se ilustra, la GPU 14 incluye el núcleo de sombreado 24 y el núcleo de sombreado 24 incluye GPR 26. Aunque no se ilustra en la FIG. 4, el núcleo de sombreado 24 también incluye una pluralidad de elementos de procesamiento, donde cada uno ejecuta una instancia del programa de sombreado.

[0063] Como un ejemplo, el núcleo de sombreado 24 puede incluir treinta y dos elementos de procesamiento y cada uno puede ejecutar una instancia de un programa de sombreado para procesar un elemento gráfico. El GPR 26 puede almacenar datos para los elementos gráficos. Por ejemplo, el GPR 26 puede almacenar datos de atributos para nueve atributos para treinta y dos elementos gráficos. Sin embargo, el GPR 26 puede almacenar datos para más o menos de nueve atributos para los treinta y dos elementos gráficos. Además, el GPR 26 puede almacenar datos que no están asociados con un atributo de los elementos gráficos, pero son los datos para una variable necesaria para procesar los elementos gráficos.

[0064] En el ejemplo ilustrado en la FIG. 4, los elementos gráficos se identifican como P0-P31, que pueden ser vértices. El atributo se identifica mediante la variable que sigue al identificador del elemento gráfico. Por ejemplo, P0.X se refiere

a la coordenada x para el elemento gráfico P0, P0.Y se refiere a la coordenada y para el elemento gráfico P0, etc. P0.R, P0.G, P0.B y P0.A se refieren al componente rojo, el componente verde, el componente azul y la opacidad del elemento gráfico P0, respectivamente. Los otros elementos gráficos (por ejemplo, P1-P31) se identifican de manera similar.

5
 [0065] En otras palabras, en la FIG. 4, cada uno de los vértices P0-P31 está asociado con una pluralidad de variables. Como ejemplo, cada uno de los vértices P0-P31 está asociado con una variable que identifica la coordenada x (P0.X a P31.X). Cada uno de los vértices P0-P31 está asociado con una variable que identifica la coordenada y (P0.Y a P31.Y), etc. Cada una de estas variables es necesaria para procesar cada uno de la pluralidad de elementos gráficos. Por ejemplo, la variable que identifica la coordenada x es necesaria para procesar cada uno de los vértices P0-P31.

10
 [0066] Como también se ilustra en la FIG. 4, cada uno de los elementos gráficos también incluye un atributo PRJ. El atributo PRJ es una matriz de proyección que puede utilizar un sombreador de vértices que se ejecuta en elementos de procesamiento del núcleo de sombreado 24. En este ejemplo, el atributo PRJ es otra variable que se necesita para procesar cada uno de los vértices P0-P31. Por ejemplo, el sombreador de vértices puede multiplicar la matriz de proyección con las coordenadas respectivas (por ejemplo, multiplicar P0.PRJ con P0.X, P0.Y, P0.Z y P0.W).

15
 [0067] Debe comprenderse que 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 del sistema 16 o puede almacenar datos en la memoria local (por ejemplo, memoria caché). El GPR 26 del núcleo de sombreado 24 es distinto tanto de la memoria del sistema 16 como de la memoria local de la GPU 14. Por ejemplo, la memoria del sistema 16 es accesible por diversos componentes del dispositivo 10, y estos componentes usan la memoria del 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, GPR 26 solo puede ser accesible por los componentes del núcleo de sombreado 24, y solo puede almacenar datos para los elementos de procesamiento del núcleo de sombreado 24.

20
 [0068] En algunos ejemplos, una o más variables de elementos gráficos en una onda gráfica son uniformes en toda la onda gráfica. En tales ejemplos, en lugar de almacenar los datos uniformes para una o más variables en entradas separadas para cada subproceso en GPR 26, la GPU 14 puede almacenar los datos uniformes una sola vez en uGPR 27. Como se muestra en la FIG. 4, el uGPR 27 es parte del GPR 26. En otros ejemplos, el uGPR 27 puede ser un registro separado.

25
 [0069] En un ejemplo, uGPR 27 puede incluir una pluralidad de ubicaciones de almacenamiento, donde cada ubicación de almacenamiento está asociada con un atributo de la pluralidad de atributos de los elementos gráficos. Por ejemplo, como se ilustra en la FIG. 4, cada elemento gráfico P0-P31 incluye nueve atributos (PRJ, x, y, z, w, R, G, B y A). En este ejemplo, uGPR 27 puede incluir nueve ubicaciones de almacenamiento, donde la primera ubicación de uGPR 27 está asociada con el atributo PRJ, la segunda ubicación de uGPR 27 está asociada con la coordenada x, etc. De nuevo, los datos en uGPR 27 pueden ser utilizados por cada subproceso de una urdimbre.

30
 [0070] La FIG. 5 es un diagrama de bloques que ilustra una configuración de ejemplo de GPU 14 que puede usarse para implementar las técnicas para predicados uniformes de esta divulgación. La GPU 14 está configurada para ejecutar instrucciones para un programa de manera paralela. La GPU 14 incluye un núcleo de sombreado 24 que incluye una unidad de control 72, elementos de procesamiento 74A-74D (colectivamente "elementos de procesamiento 74"), un almacén de instrucciones 76, un almacén de datos 78, rutas de comunicación 80, 82, 84, 86A-86D, registros de predicados (PR) 75A-75D (colectivamente "registros de predicados 75 o PR 75), y registro de predicados uniforme (UPR) 92. Las rutas de comunicación 86A-86D pueden denominarse colectivamente "rutas de comunicación 86." En algunos ejemplos, la GPU 14 puede configurarse como un sistema de procesamiento de una única instrucción, múltiples datos (SIMD) que está configurado para ejecutar una pluralidad de subprocesos de ejecución para una urdimbre de un programa (por ejemplo, sombreador) utilizando elementos de procesamiento 74. En dicho sistema SIMD, los elementos de procesamiento 74 pueden procesar juntos una única instrucción a la vez con respecto a diferentes elementos de datos. El programa puede retirarse después de que todos los subprocesos asociados con el programa completen la ejecución.

35
 [0071] La unidad de control 72 está acoplada comunicativamente al almacén de instrucciones 76 a través de la ruta de comunicación 80, a los elementos de procesamiento 74 a través de la ruta de comunicación 82, y al almacén de datos 78 a través de la ruta de comunicación 84. La unidad de control 72 puede usar la ruta de comunicación 80 para enviar instrucciones de lectura al almacén de instrucciones 76. Una instrucción de lectura puede especificar una dirección de instrucción en el almacén de instrucciones 76 desde la que se debe recuperar una instrucción. La unidad de control 72 puede recibir una o más instrucciones de programa del almacén de instrucciones 76 en respuesta al envío de la instrucción de lectura. La unidad de control 72 puede usar la ruta de comunicación 82 para proporcionar instrucciones a los elementos de procesamiento 74, y en algunos ejemplos, para recibir datos de los elementos de procesamiento 74, por ejemplo, el resultado de una instrucción de comparación para evaluar una condición de ramificación. En algunos ejemplos, la unidad de control 72 puede usar la ruta de comunicación 84 para recuperar valores de elementos de datos del almacén de datos 78, por ejemplo, para determinar una condición de ramificación.

Aunque la FIG. 4 ilustra la GPU 14 que incluye una ruta de comunicación 84, en otros ejemplos, la GPU 14 puede no incluir una ruta de comunicación 84.

[0072] Cada uno de los elementos de procesamiento 74 puede estar configurado para instrucciones de proceso para el programa almacenado en el almacén de instrucciones 76. En algunos ejemplos, cada uno de los elementos de procesamiento 74 puede configurarse para realizar el mismo conjunto de operaciones. Por ejemplo, cada uno de los elementos de procesamiento 74 puede implementar la misma arquitectura de conjunto de instrucciones (ISA). En ejemplos adicionales, cada uno de los elementos de procesamiento 74 puede ser una unidad lógica aritmética (ALU). En otros ejemplos, la GPU 14 puede configurarse como un procesador de vectores, y cada uno de los elementos de procesamiento 74 puede ser un elemento de procesamiento dentro del procesador de vectores. En ejemplos adicionales, la GPU 14 puede ser una unidad de ejecución SIMD, y cada uno de los elementos de procesamiento 74 puede ser un elemento de procesamiento SIMD dentro de la unidad de ejecución SIMD.

[0073] Entre las operaciones realizadas por los elementos de procesamiento 74 pueden incluirse operaciones aritméticas, operaciones lógicas, operaciones de comparación, etc. Entre las operaciones aritméticas pueden incluirse operaciones tales como, por ejemplo, una operación de suma, una operación de resta, una operación de multiplicación, etc. Entre las operaciones aritméticas también se pueden incluir, por ejemplo, operaciones aritméticas de enteros y/u operaciones aritméticas de coma flotante. Entre las operaciones lógicas pueden incluirse operaciones, como, por ejemplo, una operación AND a nivel de bit, una operación OR a nivel de bit, una operación XOR a nivel de bit, etc. Entre las operaciones de comparación pueden incluirse operaciones, como, por ejemplo, una operación mayor que, una operación menor que, una operación igual a cero, una operación no igual a cero, etc. Las operaciones mayor que y menor que pueden determinar si un primer elemento de datos es mayor o menor que un segundo elemento de datos. Las operaciones igual a cero y no igual a cero pueden determinar si un elemento de datos es igual a cero o no igual a cero. Los operandos utilizados para las operaciones pueden almacenarse en registros contenidos en el almacén de datos 78.

[0074] Cada uno de los elementos de procesamiento 74 puede estar configurado para realizar una operación en respuesta a recibir una instrucción desde la unidad de control 72 a través de la ruta de comunicación 82. En algunos ejemplos, cada uno de los elementos de procesamiento 74 puede configurarse para activarse y/o desactivarse independientemente de los otros elementos de procesamiento 74. En tales ejemplos, cada uno de los elementos de procesamiento 74 puede configurarse para realizar una operación en respuesta a recibir una instrucción de la unidad de control 72 cuando el elemento de procesamiento respectivo 74A-74D está activado, y no realizar la operación en respuesta a recibir la instrucción de unidad de control 72 cuando el elemento de procesamiento respectivo 74A-74D está desactivado, es decir, no activado.

[0075] Cada uno de los elementos de procesamiento 74A-74D puede estar acoplado en comunicación al almacén de datos 78 a través de una respectiva ruta de comunicación 86A-86D. Los elementos de procesamiento 74 pueden configurarse para recuperar datos del almacén de datos 78 y almacenar datos en el almacén de datos 78 a través de rutas de comunicación 86. Los datos recuperados del almacén de datos 78 pueden, en algunos ejemplos, ser operandos para las operaciones realizadas por los elementos de procesamiento 74. Los datos almacenados en el almacén de datos 78 pueden, en algunos ejemplos, ser el resultado de una operación realizada por elementos de procesamiento 74.

[0076] El almacén de instrucciones 76 está configurado para almacenar un programa para su ejecución mediante la GPU 14. El programa puede almacenarse como una secuencia de instrucciones. En algunos ejemplos, cada instrucción puede ser dirigida por un valor de dirección de instrucción único. En tales ejemplos, los valores de dirección de instrucción para instrucciones posteriores en la secuencia de instrucciones son mayores que los valores de dirección de instrucción para instrucciones anteriores en la secuencia de instrucciones. Las instrucciones del programa, en algunos ejemplos, pueden ser instrucciones a nivel de máquina. Es decir, en tales ejemplos, las instrucciones pueden estar en un formato que corresponda a la ISA de la GPU 14. El almacén de instrucciones 76 está configurado para recibir una instrucción de lectura desde la unidad de control 72 a través de la ruta de comunicación 80. La instrucción de lectura puede especificar una dirección de instrucción desde la cual se debe recuperar una instrucción. En respuesta a la recepción de la instrucción de lectura, el almacén de instrucciones 76 puede proporcionar una instrucción correspondiente a la dirección de instrucción especificada en la instrucción de lectura a la unidad de control 72 a través de la ruta de comunicación 80.

[0077] El almacén de instrucciones 76 puede ser cualquier tipo de memoria, memoria caché o combinación de los mismos. Cuando el almacén de instrucciones 76 es una memoria caché, el almacén de instrucciones 76 puede almacenar en memoria caché un programa almacenado en una memoria de programa externa a la GPU 14. Aunque el almacén de instrucciones 76 se ilustra dentro de la GPU 14, en otros ejemplos, el almacén de instrucciones 76 puede ser externo a la GPU 14.

[0078] El almacén de datos 78 está configurado para almacenar elementos de datos utilizados por los elementos de procesamiento 74. En algunos ejemplos, el almacén de datos 78 puede comprender una pluralidad de registros, estando cada registro configurado para almacenar un elemento de datos respectivo dentro de una pluralidad de elementos de datos utilizados en la GPU 14. El almacén de datos 78 puede estar acoplado a una o más rutas de

comunicación (no mostradas) que están configuradas para transferir datos entre los registros en el almacén de datos 78 y una memoria o memoria caché (no mostradas).

5 **[0079]** Aunque la FIG. 4 ilustra un único almacén de datos 78 para almacenar datos utilizados por los elementos de procesamiento 74, en otros ejemplos, la GPU 14 puede incluir almacenes de datos separados y dedicados para cada uno de los elementos de procesamiento 74. La GPU 14 ilustra cuatro elementos de procesamiento 74 a modo de ejemplo. En otros ejemplos, la GPU 14 puede tener muchos más elementos de procesamiento en la misma configuración o en una diferente.

10 **[0080]** La unidad de control 72 está configurada para controlar la GPU 14 para ejecutar instrucciones para un programa almacenado en el almacén de instrucciones 76. Para cada instrucción o conjunto de instrucciones del programa, la unidad de control 72 puede recuperar la instrucción del almacén de instrucciones 76 a través de la ruta de comunicación 80, y procesar la instrucción. En algunos ejemplos, la unidad de control 72 puede procesar la instrucción haciendo que una operación asociada con la instrucción se ejecute en uno o más de los elementos de procesamiento 74. Por ejemplo, la instrucción recuperada por la unidad de control 72 puede ser una instrucción aritmética que indica a la GPU 14 que realice una operación aritmética con respecto a los elementos de datos especificados por la instrucción, y la unidad de control 72 puede hacer que uno o más de los elementos de procesamiento 74 realicen el operación aritmética en los elementos de datos especificados. En otros ejemplos, la unidad de control 72 puede procesar la instrucción sin hacer que se realice una operación en los elementos de procesamiento 74.

20 **[0081]** La unidad de control 72 puede hacer que se realice una operación en uno o más de los elementos de procesamiento 74 proporcionando una instrucción a los elementos de procesamiento 74 a través de la ruta de comunicación 82. La instrucción puede especificar la operación a realizar mediante los elementos de procesamiento 74. La instrucción proporcionada al uno o más de los elementos de procesamiento 74 puede ser igual o diferente a la instrucción recuperada del almacén de instrucciones 76. En algunos ejemplos, la unidad de control 72 puede hacer que la operación se realice en un subconjunto particular de elementos de procesamiento 74 (incluso mediante un único elemento de procesamiento) mediante uno o ambos de la activación de un subconjunto particular de elementos de procesamiento 74 sobre el cual se debe realizar la operación y la desactivación de otro subconjunto de elementos de procesamiento 74 sobre el cual no se debe realizar la operación. La unidad de control 72 puede activar y/o desactivar los elementos de procesamiento 74 proporcionando señales de activación y/o desactivación respectivas a cada uno de los elementos de procesamiento 74 a través de la ruta de comunicación 82. En algunos ejemplos, la unidad de control 72 puede activar y/o desactivar los elementos de procesamiento 74 proporcionando señales de activación y/o desactivación a los elementos de procesamiento 74 junto con proporcionar una instrucción a los elementos de procesamiento 74. En otros ejemplos, la unidad de control 72 puede activar y/o desactivar los elementos de procesamiento 74 antes de proporcionar una instrucción a los elementos de procesamiento 74.

40 **[0082]** La unidad de control 72 puede ejecutar una pluralidad de subprocesos de ejecución para un programa usando elementos de procesamiento 74. Una pluralidad de subprocesos que se ejecutarán en paralelo a veces se denomina urdimbre. Cada uno de los elementos de procesamiento 74 puede configurarse para procesar instrucciones del programa para un subproceso respectivo de la pluralidad de subprocesos. Por ejemplo, la unidad de control 72 puede asignar cada subproceso de ejecución a uno individual de elementos de procesamiento 74 para su procesamiento. Los subprocesos de ejecución para el programa pueden ejecutar el mismo conjunto de instrucciones con respecto a diferentes elementos de datos en un conjunto de elementos de datos. Por ejemplo, el elemento de procesamiento 74A puede ejecutar un primer subproceso de ejecución para un programa almacenado en el almacén de instrucciones 76 con respecto a un primer subconjunto de elementos de datos en una pluralidad de elementos de datos, y el elemento de procesamiento 74B puede ejecutar un segundo subproceso de ejecución para el programa almacenado en el almacén de instrucciones 76 con respecto a un segundo subconjunto de elementos de datos en la pluralidad de elementos de datos. El primer subproceso de ejecución puede incluir las mismas instrucciones que el segundo subproceso de ejecución, pero el primer subconjunto de elementos de datos puede ser diferente del segundo subconjunto de elementos de datos.

55 **[0083]** En algunos ejemplos, la unidad de control 72 puede activar y desactivar subprocesos individuales en la pluralidad de subprocesos de ejecución. Cuando la unidad de control 72 desactiva un subproceso, la unidad de control 72 también puede desactivar y/o deshabilitar el elemento de procesamiento 74A-74D que está asignado para ejecutar el subproceso. Tales subprocesos desactivados pueden denominarse subprocesos inactivos. De manera similar, cuando la unidad de control 72 activa un subproceso, la unidad de control 72 también puede activar el elemento de procesamiento 74A-74D que está asignado para ejecutar el subproceso. Tales subprocesos activados pueden denominarse subprocesos activos. Como se explicará con más detalle a continuación, la unidad de control 72 puede configurarse para seleccionar un subproceso de activación para ejecutar una operación divergente (por ejemplo, una operación en serie), sin necesidad de considerar otros subprocesos activos o inactivos.

65 **[0084]** Como se usa en el presente documento, un subproceso activo puede referirse a un subproceso que se activa, y un subproceso inactivo puede referirse a un subproceso que está desactivado. Para una pluralidad de subprocesos que se ejecutan en la GPU 14 durante un ciclo de procesamiento dado, cada uno de los subprocesos activos puede configurarse para procesar una instrucción del programa identificado por un registro de contador de programa global para los subprocesos de pluralidad durante el ciclo de procesamiento. Por ejemplo, la unidad de control 72 puede

activar los elementos de procesamiento 74 que están asignados a subprocesos activos para configurar dichos elementos de procesamiento 74 para procesar la instrucción del programa durante el ciclo de procesamiento. Por otro lado, para una pluralidad de subprocesos que se ejecutan en la GPU 14 durante un ciclo de procesamiento dado, cada uno de los subprocesos inactivos puede configurarse para no procesar las instrucciones del programa durante el ciclo de procesamiento. Por ejemplo, la unidad de control 72 puede desactivar los elementos de procesamiento 74 que están asignados a subprocesos inactivos para configurar dichos elementos de procesamiento 74 para que no procesen la instrucción del programa durante el ciclo de procesamiento.

[0085] En algunos ejemplos, un ciclo de procesamiento puede referirse al intervalo de tiempo entre las cargas sucesivas de contador de programa. Por ejemplo, un ciclo de procesamiento puede referirse al tiempo entre cuando el contador del programa se carga con un primer valor y cuando el contador del programa se carga con un segundo valor. Los valores primero y segundo pueden ser iguales o diferentes. En los ejemplos en los que el contador del programa se carga de manera asíncrona debido a las técnicas de comprobación de reanudación, tales cargas asíncronas pueden, en algunos ejemplos, no servir para diferenciar los ciclos de procesamiento. En otras palabras, en tales ejemplos, un ciclo de procesamiento puede referirse al intervalo de tiempo entre cargas síncronas sucesivas del contador del programa. Una carga síncrona del contador del programa puede, en algunos ejemplos, referirse a una carga que se activa mediante una señal de reloj.

[0086] En algún momento antes de la recuperación de la siguiente instrucción, la unidad de control 72 determina una siguiente instrucción a ser procesada por la GPU 14. La manera en que la unidad de control 72 determina la siguiente instrucción a procesar es diferente dependiendo de si la instrucción recuperada previamente por la GPU 14 es una instrucción de flujo de control. Si la instrucción recuperada previamente por la GPU 14 no es una instrucción de flujo de control, entonces la unidad de control 72 puede determinar que la siguiente instrucción que procesará la GPU 14 corresponde a una siguiente instrucción secuencial almacenada en el almacén de instrucciones 76. Por ejemplo, el almacén de instrucciones 76 puede almacenar las instrucciones para un programa en una secuencia ordenada, y la siguiente instrucción secuencial puede ser una instrucción que ocurre inmediatamente después de la instrucción recuperada previamente.

[0087] Si la instrucción recuperada previamente por la GPU 14 es una instrucción de flujo de control, entonces la unidad de control 72 puede determinar la siguiente instrucción que será procesada por la GPU 14 basándose en la información especificada en la instrucción de flujo de control. Por ejemplo, la instrucción de flujo de control puede ser una instrucción de flujo de control incondicional, por ejemplo, una instrucción de ramificación incondicional o una instrucción de salto, en cuyo caso la unidad de control 72 puede determinar que la próxima instrucción a procesar por la GPU 14 es una instrucción objetivo identificada por las instrucciones de flujo de control. Como otro ejemplo, la instrucción de flujo de control puede ser una instrucción de flujo de control condicional, por ejemplo, una instrucción de ramificación condicional, en cuyo caso la unidad de control 72 puede seleccionar una de una instrucción objetivo identificada por la instrucción de flujo de control o una próxima instrucción secuencial almacenada en el almacén de instrucciones 76 como la próxima instrucción para procesar desde el almacén de instrucciones 76.

[0088] Como se usa en el presente documento, una instrucción de flujo de control puede referirse a una instrucción que incluye información que identifica una instrucción objetivo en el almacén de instrucciones 76. Por ejemplo, la instrucción de flujo de control puede incluir un valor objetivo indicativo de un valor de contador de programa objetivo para la instrucción de flujo de control. El valor del contador del programa objetivo puede ser indicativo de una dirección objetivo para la instrucción objetivo. La instrucción objetivo puede, en algunos ejemplos, ser diferente de la siguiente instrucción secuencial almacenada en el almacén de instrucciones 76. El código de programa de alto nivel puede incluir declaraciones de flujo de control como, por ejemplo, if, switch, do, for, while, continue, break y goto. El compilador 38 puede traducir las declaraciones de flujo de control de alto nivel en instrucciones de flujo de control de bajo nivel, por ejemplo, nivel de máquina. Una instrucción que no es una instrucción de flujo de control puede denominarse en el presente documento como una instrucción secuencial. Por ejemplo, una instrucción secuencial puede no incluir información que identifique una instrucción objetivo.

[0089] Para obtener instrucciones de flujo de control, la información que identifica la instrucción de objetivo puede ser un indicativo de valor de una instrucción de objetivo almacenada en el almacén de instrucciones 76. En algunos ejemplos, el valor indicativo de la instrucción objetivo en el almacén de instrucciones 76 puede ser un valor indicativo de la dirección de instrucción de la instrucción objetivo en el almacén de instrucciones 76. El valor indicativo de la dirección de instrucción de la instrucción objetivo puede, en algunos casos, ser la dirección de la instrucción objetivo en el almacén de instrucciones 76. El valor indicativo de la dirección de instrucción de la instrucción objetivo puede, en casos adicionales, ser un valor utilizado para calcular la dirección de la instrucción objetivo. En otros ejemplos, el valor indicativo de la dirección de instrucción de la instrucción objetivo puede ser un valor indicativo de un valor de contador de programa objetivo que corresponde a la instrucción objetivo. El valor indicativo del valor del contador del programa objetivo puede ser, en algunos casos, el valor del contador del programa objetivo que corresponde a la instrucción objetivo. El valor indicativo del valor del contador del programa objetivo puede, en casos adicionales, ser un valor utilizado para calcular el valor del contador del programa objetivo. El valor del contador del programa objetivo que corresponde a la instrucción objetivo puede, en algunos ejemplos, ser igual a la dirección de la instrucción objetivo.

[0090] Una instrucción de flujo de control puede ser una instrucción de flujo de control hacia adelante o una instrucción de flujo de control hacia atrás. Una instrucción de flujo de control directo puede ser una instrucción de flujo de control donde la instrucción objetivo ocurre después de la instrucción de flujo de control en la secuencia ordenada de instrucciones almacenadas en el almacén de instrucciones 76. Una instrucción de flujo de control hacia atrás puede ser una instrucción de flujo de control donde la instrucción objetivo ocurre antes de la siguiente instrucción secuencial en la secuencia ordenada de instrucciones almacenadas en el almacén de instrucciones 76. La siguiente instrucción secuencial puede ocurrir inmediatamente después de la instrucción de flujo de control en la secuencia ordenada de instrucciones.

[0091] Una instrucción de flujo de control puede ser una instrucción de flujo de control condicional o una instrucción de flujo de control incondicional. Una instrucción de flujo de control condicional incluye información que especifica una condición para saltar a la instrucción objetivo asociada con la instrucción de flujo de control. Al procesar una instrucción de flujo de control condicional, si la unidad de control 72 determina que se cumple la condición, entonces la unidad de control 72 puede determinar que la siguiente instrucción a procesar es la instrucción objetivo. Por otro lado, si la unidad de control 72 determina que la condición no se cumple, entonces la unidad de control 72 puede determinar que la siguiente instrucción a procesar es la siguiente instrucción secuencial almacenada en el almacén de instrucciones 76. Una instrucción de flujo de control incondicional no incluye información que especifique una condición para saltar a la instrucción objetivo asociada con la instrucción de flujo de control. Al procesar una instrucción de flujo de control incondicional, la unidad de control 72 puede determinar incondicionalmente que la siguiente instrucción para procesar es la instrucción objetivo identificada por la instrucción de flujo de control. En otras palabras, la determinación en tal caso no está condicionada a ninguna condición especificada en la instrucción de flujo de control incondicional. Como se usa en el presente documento, una instrucción de flujo de control de condición puede denominarse en el presente documento como una instrucción de bifurcación a menos que la instrucción de bifurcación se designe como una instrucción de bifurcación incondicional. Además, una instrucción de flujo de control incondicional se puede denominar en el presente documento como una instrucción de salto.

[0092] Una instrucción de ramificación condicional puede incluir condiciones que se especifican con respecto a uno o más valores de elemento de datos. Por ejemplo, un tipo de condición puede ser una condición de comparación que compara un primer valor de elemento de datos con un segundo valor de elemento de datos para cada subproceso activo que se ejecuta en la GPU 14. La comparación de los valores del elemento de datos puede incluir, por ejemplo, determinar si el primer valor del elemento de datos es mayor, menor que, no mayor, no menor, igual o no igual al segundo valor del elemento de datos. Otro tipo de condición puede ser una condición de comprobación cero que determina si el valor de un elemento de datos para cada subproceso activo que se ejecuta en la GPU 14 es igual o no igual a cero. Debido a que cada uno de los elementos de procesamiento 74 opera en diferentes elementos de datos, el resultado de evaluar la condición puede ser diferente para cada subproceso activo que se ejecuta en la GPU 14.

[0093] Si todos los subprocesos activos que se ejecutan en la GPU 14 satisfacen la condición de ramificación o todos los subprocesos activos que se ejecutan en la GPU 14 no satisfacen la condición de ramificación, entonces se produce una condición de ramificación uniforme y se dice que la divergencia de ramificación para los subprocesos es uniforme. Por otro lado, si al menos uno de los subprocesos activos que se ejecutan en la GPU 14 satisface la condición de ramificación y al menos uno de los subprocesos activos que se ejecutan en la GPU 14 no cumple la condición de ramificación, se produce una condición de ramificación divergente y se dice que la divergencia de ramificación para los subprocesos es divergente.

[0094] Un ejemplo de una instrucción divergente es una instrucción if-then-else. Como se analizó anteriormente, cuando se procesa una instrucción if-then-else, el término "if (condición)" determina si una condición particular es verdadera (por ejemplo, al comparar un valor con otro valor). Independientemente de cómo se determine la condición, si la condición es verdadera, un elemento de procesamiento 74 en el núcleo de sombreado 24 ejecuta una primera parte del conjunto de instrucciones (por ejemplo, la parte 41). Si la condición no es verdadera, un elemento de procesamiento 74 ejecuta una segunda parte del conjunto de instrucciones (por ejemplo, la parte 43). En algunos ejemplos, la parte de las instrucciones ejecutadas de acuerdo con la condición de ramificación puede ser una instrucción sin operación (por ejemplo, las instrucciones de ramificación pueden incluir instrucciones para no realizar ninguna operación).

[0095] Para indicar cómo cada uno de los elementos 74 de procesamiento tiene que manejar una instrucción de ramificación, la unidad de control 72 puede almacenar los bits de predicado en cada uno de los registros de predicados respectivos 75. Cada elemento de procesamiento 74 puede acceder a un PR 75 dedicado asociado con el elemento de procesamiento particular. Esto se debe a que cada uno de los elementos de procesamiento 74 típicamente opera en datos diferentes. De este modo, la condición para las instrucciones de ramificación puede ser diferente para cada uno de los elementos de procesamiento 74 (es decir, diferente para cada subproceso de una ramificación).

[0096] Como se analizó anteriormente, los bits de predicado indican qué partes de un conjunto de instrucciones serán ejecutadas por el subproceso en particular, y de este modo, un elemento de procesamiento en particular 74. En algunos ejemplos, los bits de predicado pueden especificar qué partes de un conjunto de instrucciones ejecutar. En otros ejemplos, los bits de predicado indican qué partes del conjunto de instrucciones no ejecutar. Independientemente, cada elemento de procesamiento 74 de una urdimbre ejecuta el conjunto de instrucciones en

paralelo. Si un elemento de procesamiento particular 74 no ejecuta una parte particular del conjunto de instrucciones como lo indican los bits de predicado, ese elemento de procesamiento particular 74 no realiza ninguna acción (por ejemplo, ejecuta instrucciones de ninguna operación) mientras que otros elementos de procesamiento 74 en la urdimbre ejecutan esa parte particular del conjunto de instrucciones.

[0097] De acuerdo con las técnicas de la presente divulgación, el procesador 12 (por ejemplo, a través de la ejecución de compilador 38) puede estar configurado además para determinar si todos los subprocesos de una urdimbre tomarán la misma ramificación de una instrucción de ramificación. Si se realiza tal determinación, el compilador 38 puede indicar a la GPU 14 que establezca un único conjunto de bits de predicado uniformes que se pueden usar para cada subproceso en la urdimbre. Es decir, en lugar de tener bits de predicado separados para cada subproceso de una urdimbre, cuando todos los subprocesos toman la misma ramificación de una instrucción de ramificación condicional, solo se almacena un único conjunto de bits de predicado.

[0098] Por ejemplo, la unidad de control 72 puede estar configurada para almacenar un único conjunto de bits de predicado uniformes 93 en UPR 92 en respuesta a una determinación de que todos los subprocesos de una urdimbre para las primeras instrucciones 90 deben ejecutar la misma ramificación de instrucciones de ramificación. Las primeras instrucciones 90 pueden ser cualquier conjunto de instrucciones de un programa de sombreado que incluya una instrucción de ramificación. Los bits de predicado 93 almacenados en UPR 92 son accesibles por cada uno de los elementos de procesamiento 74 que ejecutan la urdimbre. Al incluir un único conjunto de bits de predicado 93 para todos los elementos de procesamiento 74, se puede ahorrar energía porque solo se realiza una única prueba para el conjunto de predicados (en lugar de una prueba del predicado para cada subproceso).

[0099] La FIG. 6 es un diagrama de bloques de otro núcleo de sombreado de ejemplo de la GPU de la FIG. 1 en mayor detalle. En el ejemplo de la FIG. 6, el núcleo de sombreado 24 es el mismo que se muestra en la FIG. 5, excepto con la adición de la unidad de procesamiento escalar 94. En este ejemplo, la unidad de procesamiento escalar 94 puede usarse para ejecutar las primeras instrucciones 90 en el caso de que los bits de predicado en UPR 92 indiquen que todos los subprocesos de una ramificación de urdimbre en la misma dirección y los datos a procesar son uniformes. De esta manera, en lugar de usar el procesador de vectores que incluye elementos de procesamiento 74, se puede usar una única unidad de procesamiento escalar 94 para procesar las primeras instrucciones 90 de ramificación uniformemente que se ejecutan en datos uniformes. En algunos ejemplos, los elementos de procesamiento 74 pueden usarse para procesar las segundas instrucciones 91 (que pueden no estar ramificadas uniformemente) en paralelo con la unidad de procesamiento escalar 94.

[0100] El uso de la unidad de procesamiento escalar 94 para ejecutar instrucciones de ramificación uniformemente, como lo indican los bits de predicado 93 en UPR 92, proporciona algunos beneficios de implementación. Esto se debe a que la unidad de procesamiento escalar 94 no necesita acceder a los PR 75 para determinar cómo se ramifica un conjunto de instrucciones. De este modo, no hay "colisiones" ni estados ambiguos entre las segundas instrucciones 91 ejecutadas por los elementos de procesamiento 74 y las primeras instrucciones 90 ejecutadas por la unidad de procesamiento escalar 94. Esto permite el procesamiento de las primeras instrucciones 90 y las segundas instrucciones 91 en paralelo.

[0101] La unidad de procesamiento escalar 94 puede ser cualquier tipo de procesador que esté configurado para funcionar en un elemento de datos al mismo tiempo. Al igual que los elementos de procesamiento 74, la unidad de procesamiento escalar 94 puede incluir una ALU. Las operaciones realizadas por la unidad de procesamiento escalar 94 pueden incluir operaciones aritméticas, operaciones lógicas, operaciones de comparación, etc. Las operaciones aritméticas pueden incluir operaciones tales como, por ejemplo, una operación de suma, una operación de resta, una operación de multiplicación, una operación de división, etc. Las operaciones aritméticas también pueden incluir, por ejemplo, operaciones aritméticas de enteros y/u operaciones aritméticas de coma flotante. Entre las operaciones lógicas pueden incluirse operaciones, como, por ejemplo, una operación AND a nivel de bit, una operación OR a nivel de bit, una operación XOR a nivel de bit, etc. Entre las operaciones de comparación pueden incluirse operaciones, como, por ejemplo, una operación mayor que, una operación menor que, una operación igual a cero, una operación no igual a cero, etc. Las operaciones mayor que y menor que pueden determinar si un primer elemento de datos es mayor o menor que un segundo elemento de datos. Las operaciones igual a cero y no igual a cero pueden determinar si un elemento de datos es igual a cero o no igual a cero. Los operandos utilizados para las operaciones pueden almacenarse en registros contenidos en el almacén de datos 78.

[0102] En algunas situaciones, cada subproceso de las primeras instrucciones 90 se ramifica de la misma manera, porque cada elemento de datos o elemento gráfico que debe hacer funcionar la urdimbre tiene el mismo valor. En este ejemplo, la unidad de procesamiento escalar 94 solo necesita realizar una operación y el resultado de esa operación puede almacenarse para todos los elementos de datos que debían procesarse.

[0103] La FIG. 7 es un diagrama de flujo que ilustra un procedimiento de ejemplo de acuerdo con las técnicas de esta divulgación. Las técnicas de la FIG. 7 puede ser implementadas por uno o más de la GPU 14 y/o el procesador 12 (véase la FIG. 1).

[0104] En un ejemplo de la divulgación, la GPU 14 puede configurarse para recibir una indicación de que todos los subprocesos de una urdimbre en la GPU 14 deben ejecutar una misma ramificación en un primer conjunto de instrucciones (100). La GPU 14 puede configurarse además para almacenar uno o más bits de predicado en una memoria como un único conjunto de bits de predicado, en el que el único conjunto de bits de predicado se aplica a todos los subprocesos en la urdimbre (102). La GPU 14 puede configurarse adicionalmente para ejecutar una parte del primer conjunto de instrucciones de acuerdo con el único conjunto de bits de predicado (104).

[0105] En un ejemplo de la divulgación, el único conjunto de bits de predicado indica la parte del primer conjunto de instrucciones a realizar por cada subproceso de la urdimbre, la parte del conjunto de instrucciones relacionadas con la misma ramificación en el primer conjunto de instrucciones. En otro ejemplo de la divulgación, el único conjunto de bits de predicado indica la parte del primer conjunto de instrucciones que no debe realizar cada subproceso de la urdimbre, la parte del conjunto de instrucciones relacionadas con la misma ramificación en el primer conjunto de instrucciones.

[0106] En otro ejemplo de la divulgación, la GPU 14 puede configurarse para ejecutar el primer conjunto de instrucciones de acuerdo con el único conjunto de bits de predicado usando un núcleo de procesamiento SIMD (por ejemplo, elementos de procesamiento 74 de la FIG. 5). En otro ejemplo de la divulgación, la GPU 14 puede configurarse para ejecutar el primer conjunto de instrucciones de acuerdo con el único conjunto de bits de predicado usando una unidad de procesamiento escalar (por ejemplo, la unidad de procesamiento escalar 94 de la FIG. 6). En un ejemplo, la GPU 14 puede configurarse para ejecutar un segundo conjunto de instrucciones usando un núcleo de procesamiento SIMD en paralelo con la ejecución del primer conjunto de instrucciones de acuerdo con el único conjunto de bits de predicado que usa la unidad de procesamiento escalar.

[0107] En otro ejemplo de la divulgación, el procesador 12 puede configurarse para determinar que todos los subprocesos de la urdimbre en la GPU deben ejecutar la misma ramificación del primer conjunto de instrucciones, y la indicación en respuesta a la determinación. En otro ejemplo, el procesador 12 puede configurarse para determinar que todos los subprocesos de la urdimbre en la GPU 14 deben ejecutar la misma ramificación del primer conjunto de instrucciones determinando que toda la urdimbre para el primer conjunto de instrucciones utiliza el mismo conjunto de constantes o determinando que toda la urdimbre para el primer conjunto de instrucciones utiliza datos de un registro uniforme de propósito general (uGPR) que se aplica a todos los subprocesos de la urdimbre.

[0108] En uno o más ejemplos, las funciones descritas se pueden implementar en hardware, programa informático, firmware o cualquier combinación de los mismos. Si se implementan en programa informático, las funciones se pueden almacenar en o transmitir a través de un medio legible por ordenador como una o más instrucciones o código, y 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 al que se pueda acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en la presente divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

[0109] A modo de ejemplo, y no de limitación, 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 se pueda usar para almacenar un código de programa deseado en forma de instrucciones o estructuras de datos y al que se pueda acceder 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 el disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disco flexible y disco Blu-ray, donde algunos discos habitualmente reproducen datos magnéticamente, mientras que otros discos reproducen datos ópticamente con láseres. Las combinaciones de los anteriores también se deben incluir dentro del alcance de los medios legibles por ordenador.

[0110] Las instrucciones se pueden ejecutar por uno o más procesadores, tales como uno o más procesadores digitales de señales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), matrices lógicas programables in situ (FPGA) u otros circuitos lógicos discretos o integrados equivalentes. En consecuencia, el término "procesador", como se usa en el presente documento, se puede referir a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar en módulos de hardware y/o de software dedicados configurados para la codificación y la decodificación, o incorporarse en un códec combinado. Además, las técnicas se podrían implementar por completo en uno o más circuitos o elementos lógicos.

- 5 **[0111]** Las técnicas de la presente divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para destacar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no se requiere necesariamente su realización por diferentes unidades de hardware. En su lugar, como se ha descrito anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionar mediante un grupo de unidades de hardware interoperativas, incluidos uno o más procesadores como los descritos anteriormente, junto con software y/o firmware adecuado.
- 10 **[0112]** Se han descrito diversos ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Un procedimiento para procesar datos, comprendiendo el procedimiento:
- 5 recibir una indicación de que todos los subprocesos de una urdimbre en una unidad de procesamiento de gráficos (GPU) deben ejecutar una misma ramificación en un primer conjunto de instrucciones;
- almacenar uno o más bits de predicado en una memoria como un único conjunto de bits de predicado, en el que el único conjunto de bits de predicado se aplica a todos los subprocesos en la urdimbre; y
- 10 ejecutar una parte del primer conjunto de instrucciones de acuerdo con el único conjunto de bits de predicado.
2. El procedimiento según la reivindicación 1, en el que el único conjunto de bits de predicado indica la parte del primer conjunto de instrucciones que debe realizar cada subproceso de la urdimbre, la parte del conjunto de instrucciones relacionadas con la misma ramificación en el primer conjunto de instrucciones.
- 15 3. El procedimiento según la reivindicación 1, en el que el único conjunto de bits de predicado indica la parte del primer conjunto de instrucciones que no debe realizar cada subproceso de la urdimbre, la parte del conjunto de instrucciones relacionadas con la misma ramificación en el primer conjunto de instrucciones.
- 20 4. El procedimiento según la reivindicación 1, en el que ejecutar la parte del primer conjunto de instrucciones comprende:
- ejecutar la parte del primer conjunto de instrucciones de acuerdo con el único conjunto de bits de predicado utilizando un núcleo de procesamiento de una única instrucción, múltiples datos (SIMD).
- 25 5. El procedimiento según la reivindicación 1, en el que ejecutar la parte del primer conjunto de instrucciones comprende:
- 30 ejecutar la parte del primer conjunto de instrucciones de acuerdo con el único conjunto de bits de predicado utilizando una unidad de procesamiento escalar.
6. El procedimiento según la reivindicación 5, que comprende además:
- 35 ejecutar un segundo conjunto de instrucciones utilizando un núcleo de procesamiento de una única instrucción, múltiples datos (SIMD) en paralelo con la ejecución del primer conjunto de instrucciones de acuerdo con el único conjunto de bits de predicado utilizando la unidad de procesamiento escalar.
7. El procedimiento según la reivindicación 1, que comprende además:
- 40 determinar que todos los subprocesos de la urdimbre en la GPU deben ejecutar la misma ramificación del primer conjunto de instrucciones; y
- enviar la indicación en respuesta a la determinación.
- 45 8. El procedimiento según la reivindicación 7, que comprende además:
- determinar que todos los subprocesos de la urdimbre en la GPU deben ejecutar la misma ramificación del primer conjunto de instrucciones determinando que toda la urdimbre para el primer conjunto de instrucciones usa el mismo conjunto de constantes o determinando que toda la urdimbre para el primer conjunto de instrucciones utiliza datos de un registro de propósito general uniforme (uGPR) que se aplica a todos los subprocesos de la urdimbre.
- 50 9. Un aparato para procesar datos, comprendiendo el aparato:
- 55 una memoria configurada para almacenar un primer conjunto de instrucciones; y
- un primer procesador configurado para:
- 60 recibir una indicación de que todos los subprocesos de una urdimbre en el primer procesador deben ejecutar una misma ramificación en un primer conjunto de instrucciones;
- almacenar uno o más bits de predicado en un registro como un único conjunto de bits de predicado, en el que el único conjunto de bits de predicado se aplica a todos los subprocesos en la urdimbre; y
- 65

ejecutar una parte del primer conjunto de instrucciones de acuerdo con el único conjunto de bits de predicado.

- 5
10. El aparato según la reivindicación 9, en el que el primer procesador es una unidad de procesamiento de gráficos (GPU).
11. El aparato reivindicación 9, que comprende además:
- 10 un segundo procesador en comunicación con el primer procesador, con el segundo procesador configurado para:
- determinar que todos los subprocesos de la urdimbre en el primer procesador deben ejecutar la misma ramificación del primer conjunto de instrucciones; y
- 15 enviar la indicación en respuesta a la determinación al primer procesador.
12. El aparato según la reivindicación 11, en el que el segundo procesador es una unidad central de procesamiento (CPU).
- 20 13. El aparato según la reivindicación 11, en el que el segundo procesador está configurado además para determinar que todos los subprocesos de la urdimbre en el primer procesador deben ejecutar la misma ramificación del primer conjunto de instrucciones determinando que toda la urdimbre para el primer conjunto de instrucciones usa el mismo conjunto de constantes o determinando que toda la urdimbre para el primer conjunto de instrucciones utiliza datos de un registro uniforme de propósito general (uGPR) que se aplica a todos los subprocesos de la
- 25 urdimbre.
14. El aparato de la reivindicación 9, en el que el aparato es un dispositivo de comunicación inalámbrica.
- 30 15. Un medio de almacenamiento legible por ordenador que almacena instrucciones que, al ejecutarse, hacen que uno o más aparatos de la reivindicación 9 emprendan el procedimiento de acuerdo con cualquiera de las reivindicaciones 1 a 8.

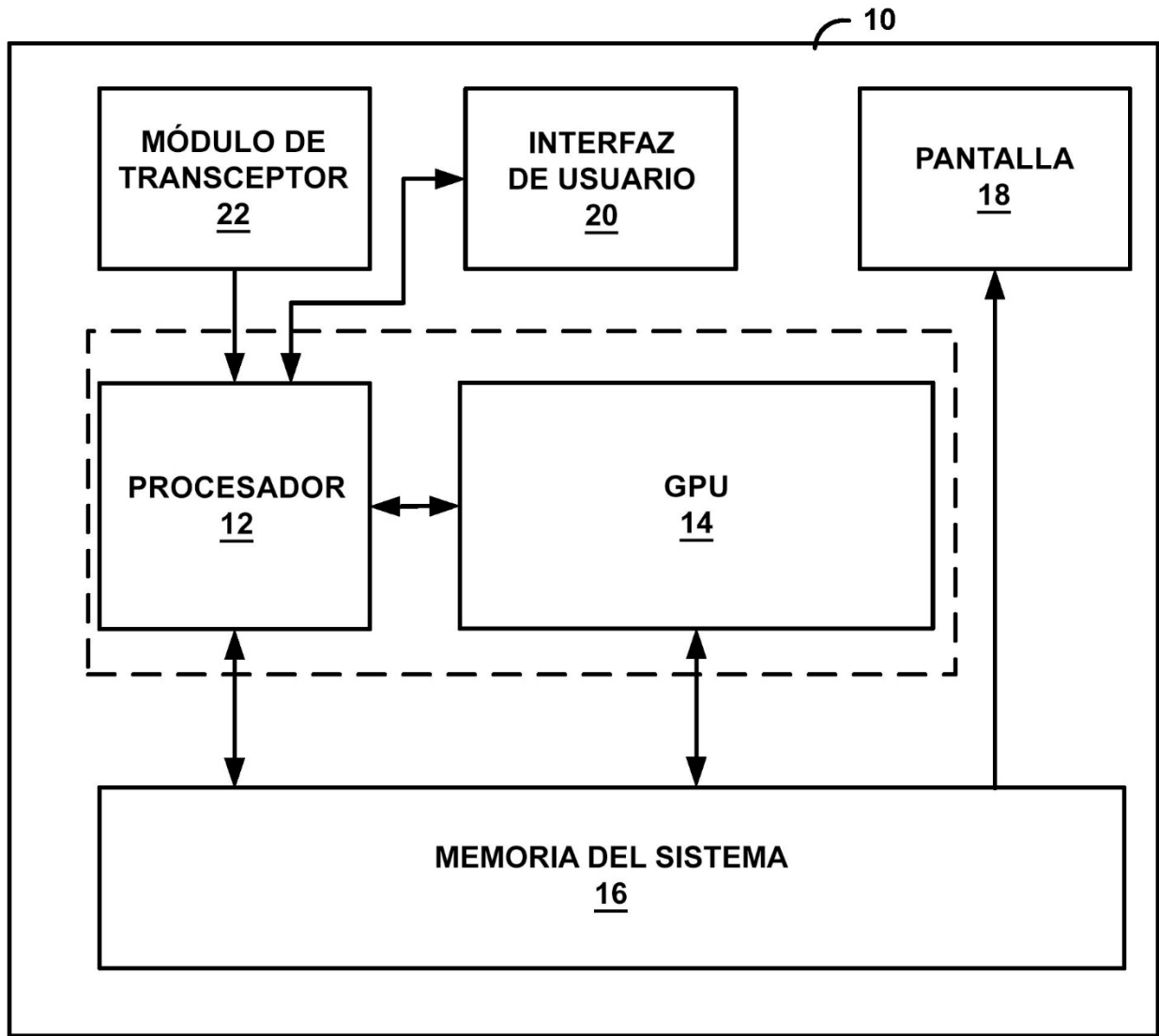


FIG. 1

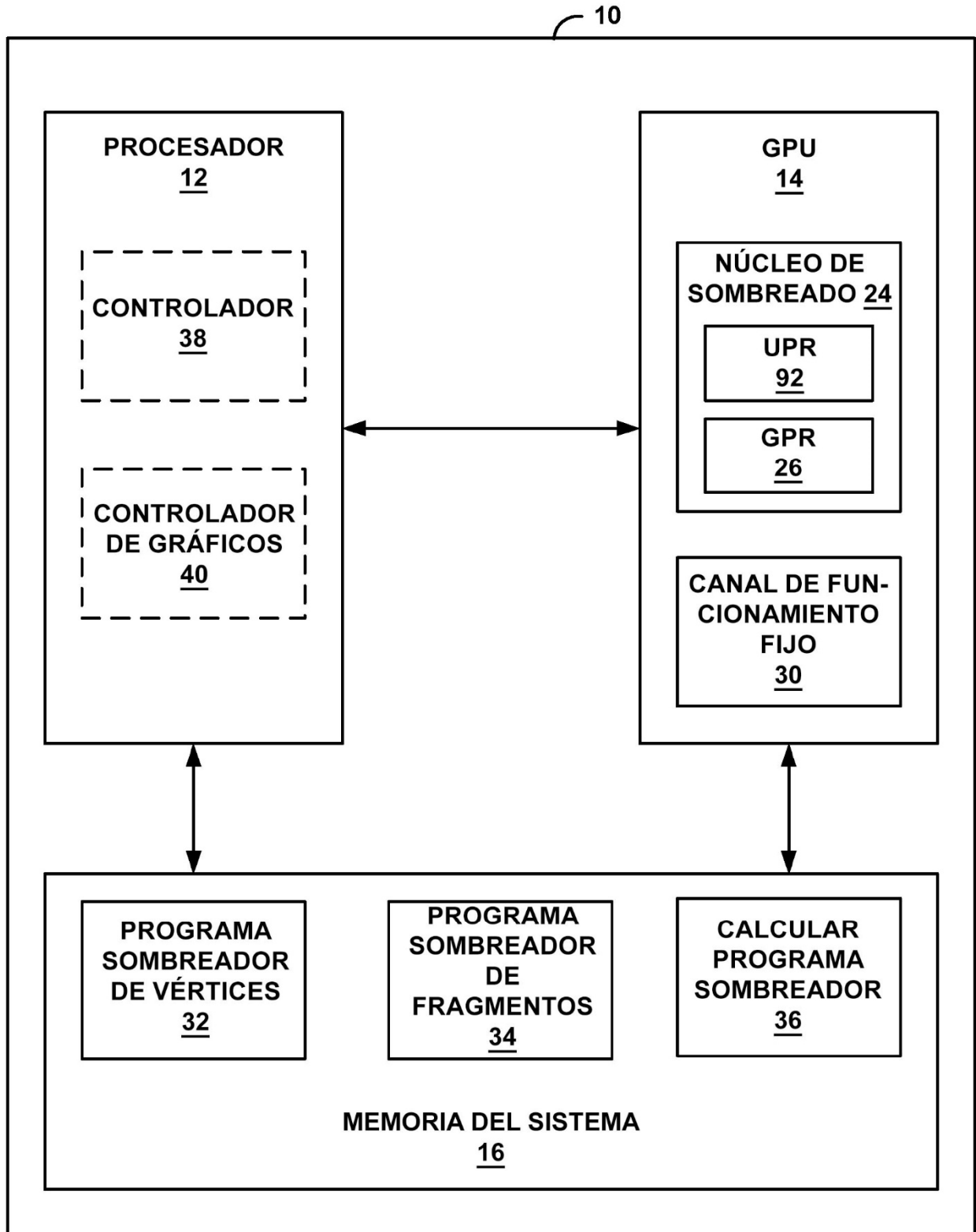


FIG. 2

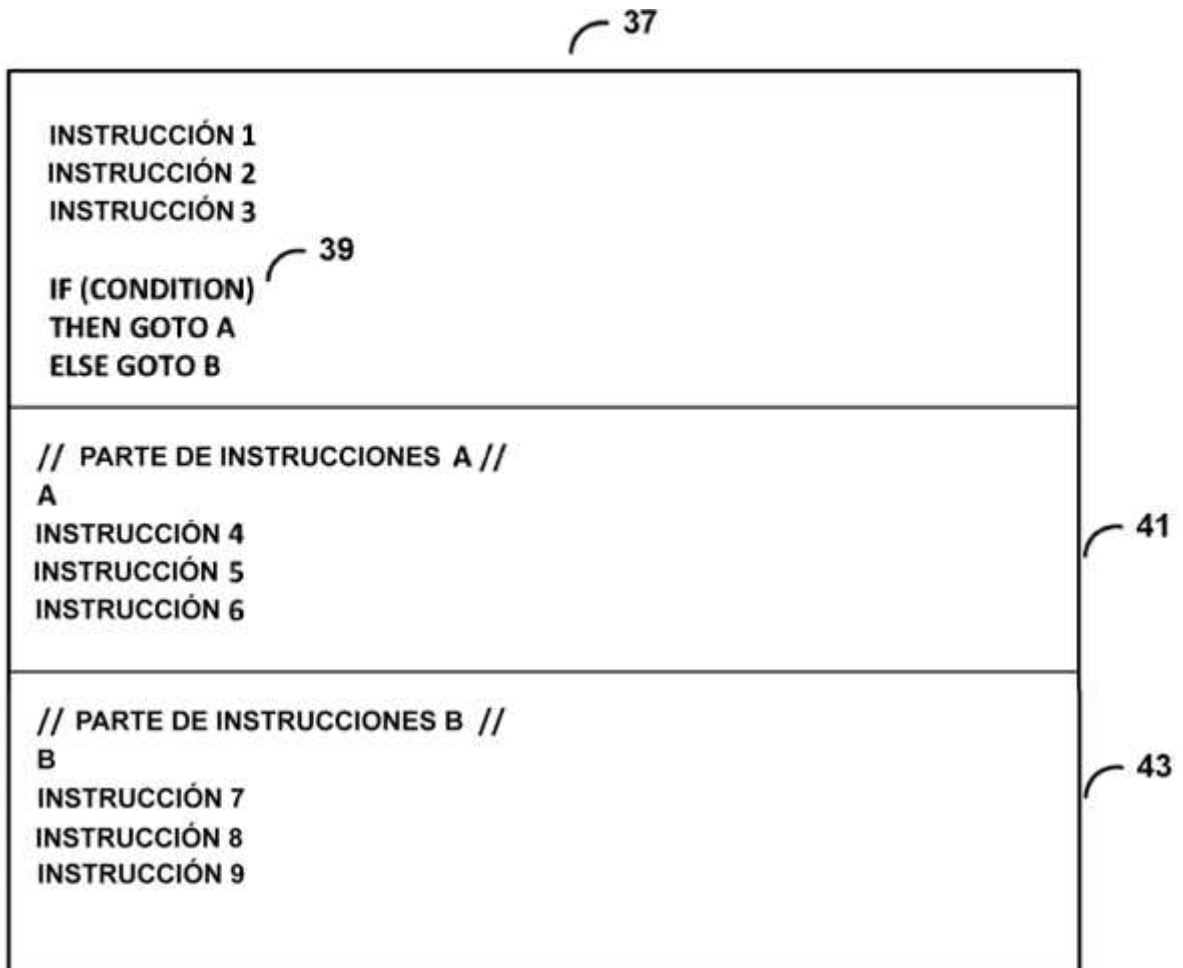


FIG. 3

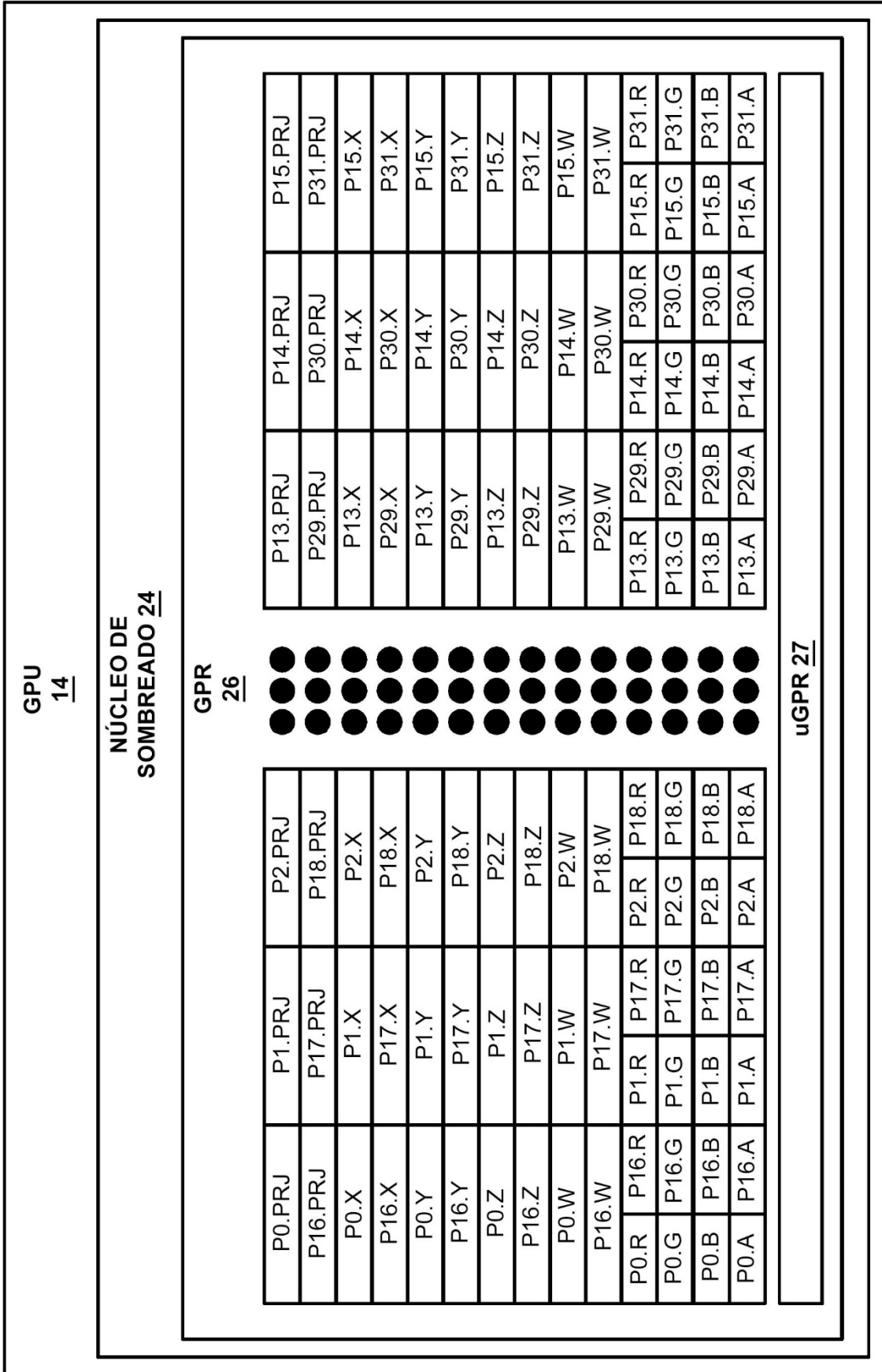


FIG. 4

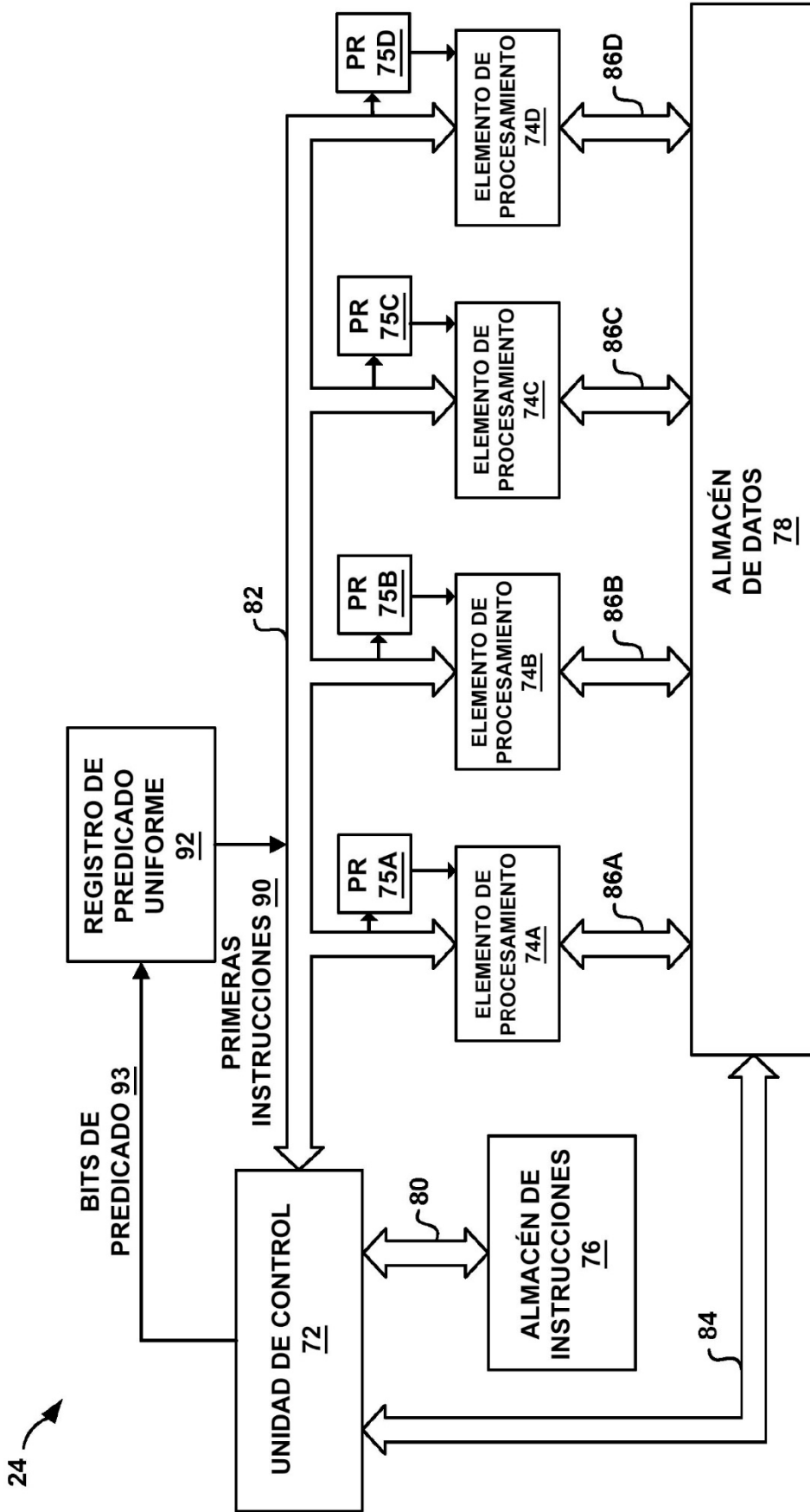


FIG. 5

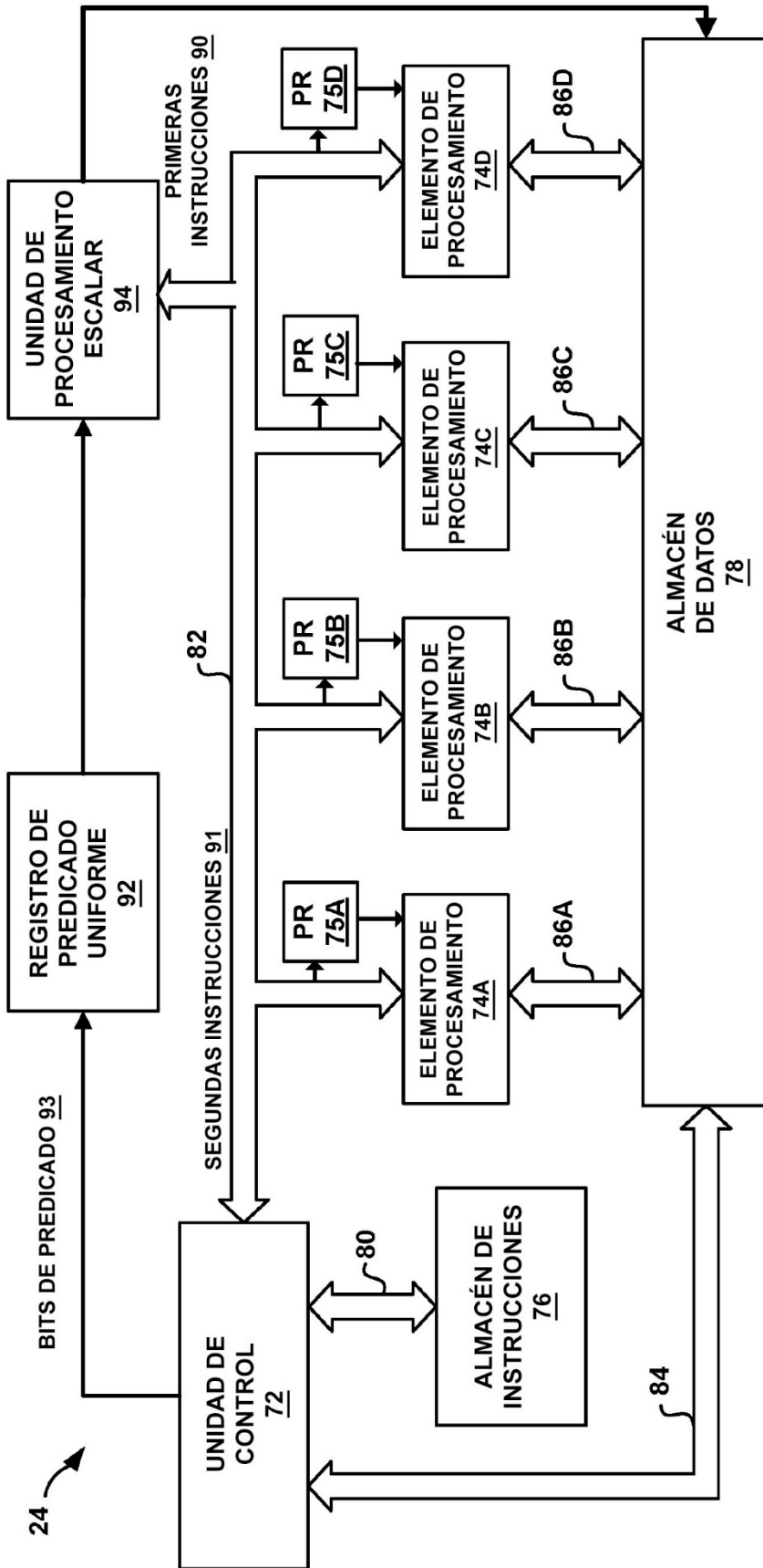


FIG. 6

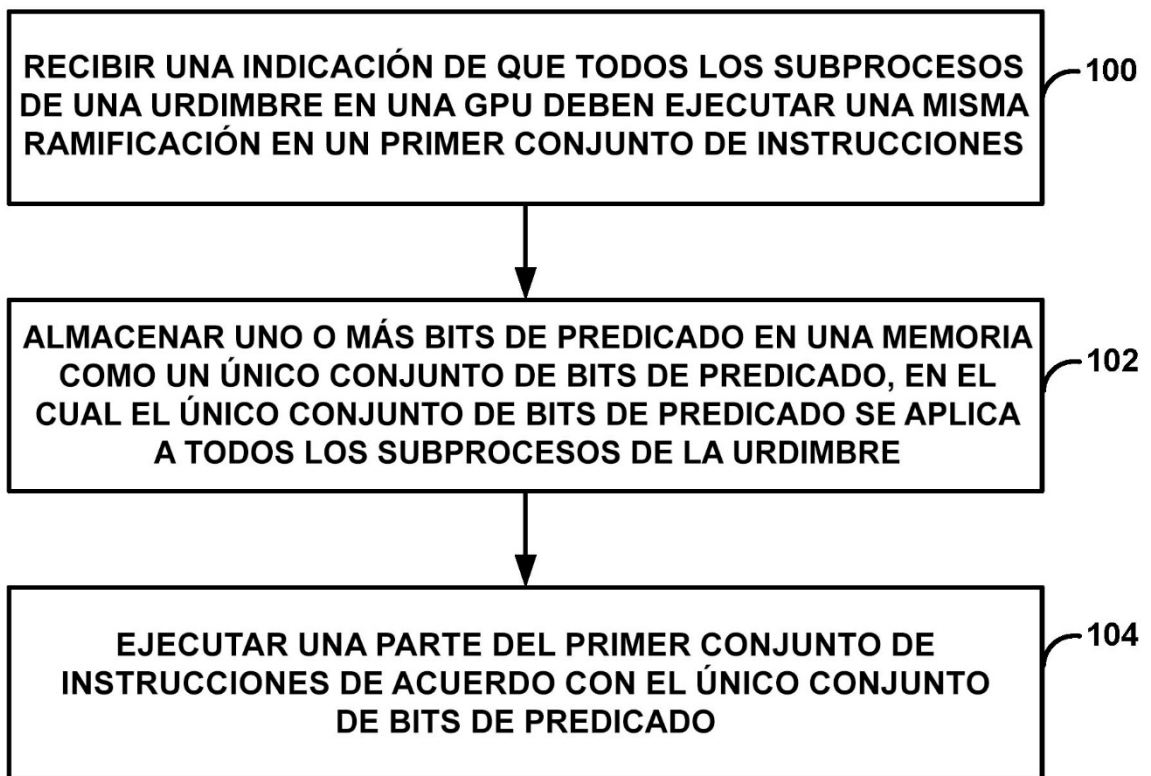


FIG. 7