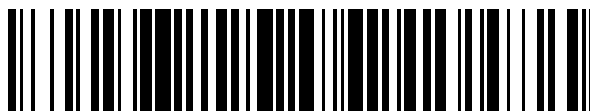


19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 791 883**

51 Int. Cl.:

**G09G 5/36** (2006.01)

**G06T 1/60** (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **11.06.2015 PCT/US2015/035332**

87 Fecha y número de publicación internacional: **30.12.2015 WO15199997**

96 Fecha de presentación y número de la solicitud europea: **11.06.2015 E 15739043 (6)**

97 Fecha y número de publicación de la concesión europea: **19.02.2020 EP 3161817**

54 Título: **Esquema de carga para registro compartido en GPU**

30 Prioridad:

**26.06.2014 US 201414316391**

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

**06.11.2020**

73 Titular/es:

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

72 Inventor/es:

**DU, YUN;  
GRUBER, ANDREW EVAN;  
CHEN, LIN;  
JIAO, GUOFANG y  
YU, CHUN**

74 Agente/Representante:

**FORTEA LAGUNA, Juan José**

ES 2 791 883 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

Esquema de carga para registro compartido en GPU

### 5 CAMPO TÉCNICO

[0001] La presente divulgación se refiere al almacenamiento de datos y, más en particular, a técnicas de almacenamiento de datos en registros.

### 10 ANTECEDENTES

[0002] Las unidades de procesamiento, tales como las unidades de procesamiento gráfico (GPU), incluyen elementos de procesamiento y un registro de uso general (GPR) que almacena datos para la ejecución de una instrucción. En algunos ejemplos, un elemento de procesamiento ejecuta instrucciones para procesar un ítem de datos, y los elementos de procesamiento respectivos almacenan los datos del ítem o los datos resultantes del ítem del procesamiento en el GPR. Un ítem de datos puede ser la unidad base en la cual se produzca el procesamiento. Por ejemplo, en el procesamiento gráfico, un vértice de una primitiva es un ejemplo de un ítem, y un píxel es otro ejemplo de un ítem. Hay datos gráficos asociados con cada vértice y píxel (por ejemplo, coordenadas, valores de color, etc.).

[0003] Puede haber múltiples elementos de procesamiento dentro de un núcleo de procesador del elemento de procesamiento que permitan 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 ítem en el GPR y lee los datos del ítem desde el GPR aunque los datos sean los mismos para múltiples ítems.

[0004] El documento US 2012/320070 divulga un procedimiento de procesamiento de datos con una unidad de procesamiento gráfico (GPU). El procedimiento incluye recibir entradas que definan órdenes de ejecución para un procesador de sombreador (SP), en las que las órdenes de ejecución comprenden una pluralidad de designaciones de kernels y una pluralidad de designaciones de grupos de trabajo. Los datos de entrada pueden permanecer en la memoria SP local y pueden estar disponibles para grupos de trabajo de múltiples kernels mediante la generación de órdenes de ejecución de kernels y grupos de trabajo, por tanto se puede reducir el consumo de ancho de banda de memoria.

### 35 BREVE EXPLICACIÓN

[0005] En general, la divulgación describe técnicas para almacenar datos uniformes en un registro de uso general compartido (SGPR) de una unidad de procesamiento gráfico (GPU). Cada unidad de procesamiento de un núcleo de procesamiento puede leer o escribir datos de un ítem, también denominado fibra, que es una unidad base en la cual se produce el procesamiento. Cada ítem puede estar asociado con múltiples variables, y estas variables de un ítem forman juntas los datos de una fibra.

[0006] En algunos ejemplos, una pluralidad de ítems, denominada onda, puede tener algunos datos que sean los mismos para cada uno de los ítems en la onda. Las técnicas descritas en la presente divulgación pueden determinar si al menos algunos datos de los ítems en una onda son los mismos. En dichos casos, el núcleo de procesamiento puede almacenar los datos, que son los mismos para todos los ítems en una onda, en el SGPR, y las instrucciones que se ejecutan en el núcleo de procesamiento pueden leer los datos del SGPR. De esta manera, en lugar de leer y escribir datos en los GPR de cada uno de los ítems, las técnicas pueden leer y escribir datos del SGPR, lo que puede reducir la cantidad de veces que se accede al GPR y, de este modo, el consumo de energía.

[0007] Los modos de realización se exponen en el conjunto adjunto de reivindicaciones. Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y en la descripción siguiente. Otros rasgos característicos, objetivos y ventajas resultarán evidentes a partir de la descripción, los dibujos y las reivindicaciones.

### 55 BREVE DESCRIPCIÓN DE LOS DIBUJOS

#### [0008]

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 la presente divulgación.

La FIG. 2 es un diagrama conceptual que ilustra un ejemplo de almacenamiento de datos en un registro de uso general (GPR) de un núcleo de sombreador de una GPU.

La FIG. 3 es un diagrama de bloques que ilustra los componentes del dispositivo ilustrado en la FIG. 1 en mayor detalle.

La FIG. 4 es un diagrama de bloques que ilustra un ejemplo de un núcleo de sombreador de una GPU de la FIG. 3 en mayor detalle.

5 Las FIGS. 5A y 5B son diagramas conceptuales que ilustran ejemplos de almacenamiento de datos de acuerdo con las técnicas descritas en la presente divulgación.

La FIG. 6 es un diagrama de flujo que ilustra una técnica de ejemplo para almacenar datos.

**DESCRIPCIÓN DETALLADA**

10 **[0009]** Las unidades de procesamiento paralelo, tales como la unidad de procesamiento gráfico (GPU), que están configuradas para realizar muchas operaciones en paralelo (por ejemplo, al mismo tiempo), incluyen uno o más núcleos de procesador (por ejemplo, núcleos de sombreador 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 la presente divulgación se pueden extender a unidades de procesamiento paralelo que no sean necesariamente GPU ni GPGPU, así como a unidades de procesamiento no paralelo (por ejemplo, las que no estén configuradas específicamente para el procesamiento paralelo).

20 **[0010]** La GPU se puede diseñar con una estructura de instrucción única, múltiples datos (SIMD). En la estructura SIMD, un núcleo de sombreador 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 particular se denomina subproceso. Se puede considerar que cada elemento de procesamiento SIMD ejecuta un subproceso diferente porque los datos para un subproceso dado pueden ser diferentes; sin embargo, el subproceso que se ejecuta en un elemento de procesamiento es la misma instrucción, del mismo programa, que la instrucción que se ejecuta en los otros elementos de procesamiento. De esta manera, la estructura SIMD permite que la GPU realice muchas tareas en paralelo (por ejemplo, al mismo tiempo).

30 **[0011]** En la estructura SIMD, el núcleo de sombreador procesa múltiples fibras simultáneamente usando la misma operación definida por la única instrucción. Una fibra se refiere a un ítem básico para el cual se produce el procesamiento. Por ejemplo, para un sombreador de vértices que se ejecuta en el núcleo de sombreador, una fibra sería un vértice. Para un sombreador de píxeles o fragmentos que se ejecuta en el núcleo de sombreador, una fibra sería un píxel.

35 **[0012]** Parte del procesamiento SIMD incluye la lectura y escritura de datos necesarios para procesar las múltiples fibras. Un ejemplo de datos necesarios para procesar múltiples fibras son los datos de atributos de las fibras (por ejemplo, datos de atributos de un vértice o datos de atributos de un píxel). Para facilitar la ilustración, las técnicas se describen con respecto a los datos de atributos, pero las técnicas no son tan limitadas. Las técnicas se pueden extender a diversos otros tipos de datos y, en general, a los datos necesarios para procesar una fibra. Estos datos también pueden incluir valores, tales como valores de contador u otros tipos de valores, que son necesarios para procesar una fibra.

45 **[0013]** Por ejemplo, para procesar una fibra, el núcleo de sombreador puede procesar una pluralidad de variables asociadas con la fibra. Un ejemplo de las variables asociadas con una fibra son los atributos asociados con una fibra. Un ejemplo de los datos de una variable son los datos de atributos de un atributo.

50 **[0014]** En algunos diseños de núcleos de sombreador, el núcleo de sombreador lee datos que son los mismos para la mayoría de las fibras (es decir, datos uniformes) una y otra vez. Por ejemplo, no son divergentes los datos de una o más variables (por ejemplo, los datos de atributos de uno o más atributos) que tengan que leerse para las fibras. El núcleo de sombreador almacena entonces estos datos en localizaciones de almacenamiento asignadas a la fibra dentro de un registro de uso general (GPR), donde el GPR está localizado dentro del núcleo de sombreador.

55 **[0015]** Como ejemplo, cada fibra está asociada con una pluralidad de datos de atributos. Para un vértice o un píxel, los datos de atributos incluyen coordenadas, datos de color, opacidad, etc., como algunos ejemplos. Sin embargo, también hay otros tipos de datos necesarios para procesar la fibra, y las técnicas son aplicables a dichos tipos de datos. Si bien todos los datos de atributos pueden no ser los mismos para una pluralidad de vértices o píxeles, en algunos casos, algunos de los datos de atributos pueden ser los mismos para la pluralidad de vértices o píxeles. Los datos (por ejemplo, los datos de atributos en este ejemplo) que son los mismos para la pluralidad de vértices o píxeles se denominan datos uniformes. Los datos que son diferentes para al menos dos de los vértices o píxeles se denominan datos divergentes (es decir, los datos para un vértice son diferentes a los datos para al menos otro vértice).

60 **[0016]** En lugar de leer los datos uniformes una y otra vez para almacenarlos en los GPR y leer repetidamente los GPR para los datos uniformes, las técnicas descritas en la presente divulgación determinan si no son divergentes los datos que se van a leer para una pluralidad de fibras (denominándose la pluralidad de fibras onda). Si los datos que se van a leer para la onda no son divergentes, las técnicas descritas en la divulgación leen los datos uniformes una vez (por ejemplo, una sola vez) para almacenarlos en un registro de uso general compartido (SGPR) especializado.

El núcleo de sombreador lee los datos uniformes del SGPR en lugar de acceder repetidamente al GPR. El SGPR es un GPR compartido en el sentido de que el SGPR almacena datos que son necesarios para todas las fibras de la onda (por ejemplo, almacena datos de atributos que comparten todas las fibras de una onda).

5 **[0017]** El número de fibras que forman una onda puede ser una función de las capacidades de procesamiento paralelo del núcleo de sombreador. Por ejemplo, si el núcleo de sombreador está configurado para procesar treinta y dos fibras (como un simple ejemplo) simultáneamente, entonces el número de fibras en una onda es igual a treinta y dos.

10 **[0018]** Puede haber diversas formas de determinar si algunos datos de las fibras son uniformes a través de una onda. Como ejemplo, el compilador que se ejecuta en un procesador puede determinar si una instrucción particular de un programa de sombreador requerirá acceso a datos uniformes para una onda (por ejemplo, un atributo que sea el mismo para las fibras en la onda). En este caso, el compilador puede codificar una instrucción que indique el acceso uniforme a los datos. El hardware de la GPU puede leer esta instrucción y almacenar los datos uniformes en el SGPR para la onda (por ejemplo, almacenar los datos de atributos que sean los mismos para todas las fibras en la onda en el SGPR).

15 **[0019]** Es posible que el compilador no pueda siempre determinar si el acceso a los datos es uniforme a través de una onda para una instrucción particular. Sin embargo, el compilador puede determinar que existe una probabilidad de que una instrucción particular requiera acceso a datos uniformes a través de una onda. En dichos ejemplos, el compilador puede codificar una instrucción que identifique una instrucción particular como candidato para usar el SGPR. Para la instrucción identificada, el hardware de la GPU puede determinar si las direcciones de memoria para acceder a los datos de las fibras de una onda son las mismas. Si las direcciones de memoria son las mismas, el hardware puede establecer un valor de indicador de SGPR como 1, recuperar los datos una vez y almacenar los datos uniformes en el SGPR. Cuando los datos son necesarios para el procesamiento, el hardware puede leer el valor de indicador y determinar si los datos deberían leerse desde el SGPR o el GPR.

20 **[0020]** De esta manera, las técnicas descritas en la presente divulgación reducen el número de veces que se necesita acceder al GPR. Esto a su vez puede aumentar la energía y la eficiencia del procesamiento. Por ejemplo, el GPR puede ser una unidad de almacenamiento de memoria relativamente grande que almacene los datos (por ejemplo, valores) para todas las variables (por ejemplo, atributos) necesarios para una pluralidad de fibras diferentes. Un SGPR, por otro lado, puede almacenar los datos para una sola variable (por ejemplo, un solo atributo) que sean los mismos para una onda de fibras. En consecuencia, el SGPR puede ser de menor tamaño que el GPR. En algunos casos, acceder al SGPR puede requerir menos energía que acceder al GPR. Al reducir la cantidad de veces que se debe acceder al GPR, al acceder al SGPR en su lugar, las técnicas pueden reducir la cantidad de energía que consume la GPU.

30 **[0021]** Como se describe anteriormente, una fibra se refiere a una unidad base en la cual se produce el procesamiento. Para evitar confusiones, la presente divulgación se refiere a una fibra como ítem gráfico para indicar un ítem base en el cual una GPU realiza el procesamiento. En consecuencia, una pluralidad de ítems gráficos puede formar una onda gráfica. Sin embargo, en algunos ejemplos, la GPU puede no realizar necesariamente operaciones gráficas. Para dichos casos, el término ítem gráfico se refiere a una unidad base en la cual la GPU realiza el procesamiento, incluso aunque dicho ítem gráfico no sea para procesamiento gráfico. En otras palabras, un ítem gráfico es una unidad base en la cual la GPU realiza el procesamiento, y los ejemplos de un ítem gráfico incluyen un ítem para el procesamiento relacionado con gráficos o para el procesamiento no relacionado con gráficos.

35 **[0022]** 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 la presente divulgación. La FIG. 1 ilustra el dispositivo 10, cuyos ejemplos incluyen, pero no se limitan a, dispositivos de vídeo tales como reproductores de medios, decodificadores, auriculares inalámbricos tales como teléfonos móviles, asistentes digitales personales (PDA), ordenadores de escritorio, ordenadores portátiles, consolas de videojuegos, unidades de videoconferencia, dispositivos informáticos de *tablets* y similares.

40 **[0023]** En el ejemplo de la FIG. 1, el dispositivo 10 incluye el procesador 12, la unidad de procesamiento gráfico (GPU) 14 y la memoria de sistema 16. En algunos ejemplos, tales como ejemplos en los que el dispositivo 10 sea un dispositivo móvil, el procesador 12 y la GPU 14 se pueden formar como un circuito integrado (IC). Por ejemplo, el IC se puede considerar como un chip de procesamiento dentro de un paquete de chips, tal como un sistema en chip (SoC). En algunos ejemplos, el procesador 12 y la GPU 14 se pueden alojar en diferentes circuitos integrados (es decir, diferentes paquetes de chips) tales como ejemplos en los que el dispositivo 10 sea un ordenador de escritorio o un 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.

45 **[0024]** Los ejemplos del procesador 12 y de la GPU 14 incluyen, pero no se limitan a, uno o más procesadores de señales digitales (DSP), microprocesadores de uso general, circuitos integrados específicos de la aplicación (ASIC), matrices lógicas programables por campo (FPGA) u otros circuitos lógicos integrados equivalentes o lógica discreta. El procesador 12 puede ser la unidad central de procesamiento (CPU) del dispositivo 10. En algunos ejemplos, la GPU

14 puede ser hardware especializado que incluya circuitos lógicos integrados y/o discretos que proporcionen a la GPU 14 capacidades de procesamiento paralelo masivas adecuadas para el procesamiento gráfico. En algunos casos, la GPU 14 también puede incluir capacidades de procesamiento de uso general, y se puede denominar GPU de uso general (GPGPU) cuando se implementen tareas de procesamiento de uso general (es decir, tareas no relacionadas con gráficos).

**[0025]** Para los propósitos de ilustración, las técnicas descritas en la presente divulgación se describen con la GPU 14. Sin embargo, las técnicas descritas en la presente divulgación no están limitadas así. Las técnicas descritas en la presente divulgación se pueden extender a otros tipos de unidades de procesamiento en paralelo (por ejemplo, unidades de procesamiento que proporcionen capacidades de procesamiento paralelo masivo, incluso aunque no sea para el procesamiento gráfico). Además, las técnicas descritas en la presente divulgación se pueden extender a unidades de procesamiento no configuradas específicamente para el procesamiento paralelo.

**[0026]** El procesador 12 puede ejecutar diversos tipos de aplicaciones. Los ejemplos de las aplicaciones incluyen navegadores web, aplicaciones de correo electrónico, hojas de cálculo, videojuegos u otras aplicaciones que generen objetos visibles para su visualización. La memoria de sistema 16 puede almacenar instrucciones para la ejecución de la una o más aplicaciones. La ejecución de una aplicación en el procesador 12 causa que el procesador 12 produzca datos gráficos para el contenido de imagen que se vaya a visualizar. El procesador 12 puede transmitir datos gráficos del contenido de imagen a la GPU 14 para un procesamiento adicional.

**[0027]** A modo de ejemplo, la ejecución de una aplicación en el procesador 12 causa que el procesador 12 produzca vértices de primitivas, donde la interconexión de primitivas en los vértices respectivos forma un objeto gráfico. En este ejemplo, los datos gráficos que produce el procesador 12 son los datos de 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.

**[0028]** En algunos ejemplos no relacionados con gráficos, el procesador 12 puede generar datos que sean adecuados para procesarse por la GPU 14. No es necesario que dichos datos sean para propósitos de gráficos o de visualización. Por ejemplo, el procesador 12 puede emitir datos sobre qué operaciones de matriz deben realizarse por la GPU 14, y la GPU 14 puede realizar a su vez las operaciones de matriz.

**[0029]** 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 gráfico requiere operaciones paralelas masivas, y el procesador 12 puede descargar dichas tareas de procesamiento gráfico a la GPU 14. Sin embargo, otras operaciones tales como las operaciones de matriz también se pueden beneficiar de las capacidades de procesamiento paralelo de la GPU 14. En estos ejemplos, el procesador 12 puede aprovechar las capacidades de procesamiento paralelo de la GPU 14 para causar que la GPU 14 realice operaciones no relacionadas con los gráficos.

**[0030]** El procesador 12 se puede comunicar con la GPU 14 de acuerdo con una interfaz de procesamiento de aplicaciones (API) particular. Ejemplos de dichas API incluyen DirectX® API de Microsoft®, OpenGL® u OpenGL ES® del grupo Khronos y OpenCL™; sin embargo, los aspectos de la presente divulgación no están limitados a las API de DirectX, OpenGL u OpenCL, y se pueden extender a otros tipos de API. Además, las técnicas descritas en la presente divulgación no requieren funcionar de acuerdo con una API, y el procesador 12 y la GPU 14 pueden usar cualquier técnica de comunicación.

**[0031]** El dispositivo 10 también puede incluir la pantalla 18, la interfaz de usuario 20 y el módulo transceptor 22. El dispositivo 10 puede incluir módulos o unidades adicionales no mostrados en la FIG. 1 para 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 sea 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 formar 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.

**[0032]** 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 transceptor 22 puede incluir circuitos para permitir la comunicación inalámbrica o alámbrica entre el dispositivo 10 y otro dispositivo o una red. El

módulo transceptor 22 puede incluir moduladores, demoduladores, amplificadores y otros circuitos de este tipo para la comunicación alámbrica o inalámbrica.

5 **[0033]** La memoria de sistema 16 puede ser la memoria para el dispositivo 10. La memoria de sistema 16 puede comprender uno o más medios de almacenamiento legibles por ordenador. Los ejemplos de la memoria de sistema 16 incluyen, entre otros, una memoria de acceso aleatorio (RAM), una memoria de solo lectura programable y borrable eléctricamente (EEPROM), una memoria flash u otro medio que se pueda usar 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.

10 **[0034]** En algunos aspectos, la memoria de sistema 16 puede incluir instrucciones que causen que el procesador 12 y/o la GPU 14 realicen las funciones atribuidas en la presente divulgación al procesador 12 y a la GPU 14. En consecuencia, la memoria de sistema 16 puede ser un medio de almacenamiento legible por ordenador que tenga instrucciones almacenadas en el mismo que, cuando se ejecuten, causen que uno o más procesadores (por ejemplo, el procesador 12 y la GPU 14) realicen diversas funciones.

15 **[0035]** En algunos ejemplos, la memoria de sistema 16 puede ser un medio de almacenamiento no transitorio. El término "no transitorio" puede indicar que el medio de almacenamiento no se realiza ni en una onda portadora ni en una señal propagada. Sin embargo, el término "no transitorio" no debería interpretarse en el sentido de que la memoria de sistema 16 es no móvil o que sus contenidos son estáticos. Como ejemplo, la memoria de sistema 16 se puede retirar del dispositivo 10 y trasladar a otro dispositivo. Como otro ejemplo, la memoria, sustancialmente similar a la memoria de sistema 16, se puede insertar en el dispositivo 10. En determinados ejemplos, un medio de almacenamiento no transitorio puede almacenar datos que, con el tiempo, puedan cambiar (por ejemplo, en una RAM).

20 **[0036]** Por ejemplo, como se describe en más detalle en otra parte en la presente divulgación, la memoria de sistema 16 puede almacenar el código de un compilador que se ejecute en el procesador 12 que lleve a cabo técnicas de uno o más ejemplos descritos en la presente divulgación. La memoria de sistema 16 también puede almacenar código para programas de sombreador (por ejemplo, un sombreador de vértices, un sombreador de píxeles o fragmentos, un sombreador de cálculo, etc.) que se ejecuten en un núcleo de sombreador de la GPU 14.

25 **[0037]** Por ejemplo, un compilador que se ejecute en el procesador 12 puede determinar si los datos para una variable (por ejemplo, la fecha de atributo para un atributo) de una pluralidad de ítems gráficos son los mismos o tienen la probabilidad de ser los mismos. El compilador puede indicar si los datos de la pluralidad de ítems gráficos son los mismos o si tiene probabilidad de ser los mismos. La GPU 14 puede recibir la indicación y un núcleo de sombreador de la GPU 14 puede leer y escribir los datos que sean los mismos para la pluralidad de ítems gráficos de un registro de uso general compartido (SGPR) especializado, en lugar de un GPR del núcleo de sombreador.

30 **[0038]** El término ítem gráfico se usa en la presente divulgación para referirse a una unidad de base en la cual la GPU 14 realiza el procesamiento paralelo. La GPU 14 puede procesar una pluralidad de ítems 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 forma similar, un sombreador de píxeles o de 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 ítem gráfico.

35 **[0039]** Para aplicaciones no relacionadas con gráficos, el término ítem gráfico 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 "ítem gráfico" se refiere a ítems que una unidad de procesamiento gráfico (por ejemplo, GPU 14) o una unidad de procesamiento gráfico de uso general (por ejemplo, ejemplos donde la GPU 14 funciona como una GPGPU) debe procesar en paralelo. El ítem gráfico se puede usar para propósitos gráficos o no gráficos.

40 **[0040]** En algunos casos, el término "fibra" se denomina unidad base en la cual la GPU 14 realiza el procesamiento. El término "fibra" es sinónimo de "ítem gráfico". Las capacidades de procesamiento paralelo masivo de la GPU 14 le permiten a la GPU 14 procesar una pluralidad de fibras al mismo tiempo. Una pluralidad de fibras que la GPU 14 procesa al mismo tiempo se denomina onda. En la presente divulgación, una "onda gráfica" es sinónimo de una onda de fibras. En otras palabras, una pluralidad de ítems gráficos forma una onda gráfica.

45 **[0041]** Como se describe anteriormente, la GPU 14 proporciona capacidades masivas de procesamiento paralelo. Una forma en la cual la GPU 14 proporciona dichas capacidades de procesamiento paralelo es por medio de una estructura de instrucción única, múltiples datos (SIMD). En la estructura SIMD, la GPU 14 ejecuta una pluralidad de instancias del mismo programa. Por ejemplo, el procesamiento gráfico y algunos procesos no relacionados con gráficos requieren que se realicen las mismas operaciones, pero sobre datos diferentes.

50 **[0042]** Por ejemplo, la GPU 14 puede ejecutar programas de sombreador (denominados simplemente sombreadores) que realicen tareas relacionadas con gráficos o tareas no relacionadas con gráficos. La GPU 14 incluye

al menos un núcleo de sombreador, y los programas de sombreador se ejecutan en el núcleo de sombreador. Para mayor brevedad y facilidad de descripción, la GPU 14 se describe realizando tareas relacionadas con gráficos, pero puede realizar de forma similar tareas no relacionadas con gráficos. La GPU 14 puede ejecutar múltiples instancias del programa de sombreador para procesar múltiples ítems gráficos a la vez. Un ejemplo de un programa de sombreador es un sombreador de vértices. La GPU 14 puede ejecutar múltiples instancias del sombreador de vértices para procesar múltiples vértices a la vez (en paralelo). En general, las operaciones que se deben realizar para un vértice son las mismas que las operaciones que se deben realizar para otros vértices.

**[0043]** Aunque las operaciones que deben realizarse para el vértice son las mismas, uno o más datos de atributos de cada uno de los vértices pueden ser diferentes. Por ejemplo, cada ítem gráfico (por ejemplo, un vértice o un píxel) está asociado con una pluralidad de atributos. Cada atributo incluye datos de atributos. Como ejemplo, un vértice de una primitiva o un píxel de una pantalla se puede definir por coordenadas (por ejemplo, x, y, z, w). Cada uno de los vértices y píxeles también se puede definir por colores (por ejemplo, un componente rojo, un componente verde, un componente azul (RGB)). También puede haber un valor de opacidad para cada vértice o píxel (por ejemplo, un valor alfa). Las coordenadas, el color y la opacidad son ejemplos de atributos de un ítem gráfico (por ejemplo, fibra) y puede haber más o menos atributos que los ejemplos.

**[0044]** Como se describe anteriormente, los atributos no deben considerarse limitados a coordenadas, color y opacidad. Las coordenadas, el color y la opacidad se proporcionan simplemente para ayudar con la comprensión. Sin embargo, en algunos casos, los atributos de un ítem gráfico pueden estar limitados a tipos de datos específicos (por ejemplo, debido a la manera en que se programa un programa de sombreador o debido a la manera en la cual se diseña la GPU 14).

**[0045]** En general, cada ítem gráfico puede estar asociado con una pluralidad de variables, y la pluralidad de variables es necesaria para procesar cada uno de los ítems gráficos. Un ejemplo de la pluralidad de variables son los atributos asociados con un vértice o píxel. Para cada variable, puede haber datos (por ejemplo, valores) que definen la variable. Un ejemplo de los datos son los datos de atributos asociados con cada atributo. En este ejemplo, los atributos asociados con los vértices o píxeles son necesarios para procesar cada uno de los vértices y píxeles. Para procesar los atributos, la GPU 14 procesa los datos de atributos de los atributos.

**[0046]** Para facilitar la descripción, las técnicas se describen con referencia a los datos de atributos asociados con los atributos necesarios para procesar un vértice o un píxel. Sin embargo, las técnicas descritas en la presente divulgación son aplicables en general a los datos asociados con las variables necesarias para procesar un ítem gráfico, y los datos de atributos asociados con los atributos para un vértice o un píxel se proporcionan para facilitar la comprensión.

**[0047]** Por ejemplo, una de las funciones del sombreador de vértices es multiplicar las coordenadas con una matriz de la matriz de proyección (PRJ). Esta matriz PRJ puede ser un ejemplo del atributo de los vértices. Para muchos vértices en un objeto gráfico, la matriz PRJ puede ser la misma. Por ejemplo, los valores de color y las coordenadas pueden ser diferentes, pero la matriz PRJ puede ser la misma. Esta matriz PRJ se puede considerar como un ejemplo de datos de instancia de vértice OpenGL.

**[0048]** En las técnicas descritas en la presente divulgación, se puede configurar un núcleo de sombreador de la GPU 14 para determinar si los datos de atributos para uno o más atributos son los mismos para una pluralidad de ítems gráficos (por ejemplo, los ítems gráficos de una onda gráfica). Si los datos de atributos son los mismos, el núcleo de sombreador de la GPU 14 puede almacenar los datos de atributos en un registro de uso general compartido (SGPR) que se comparta por elementos de procesamiento del núcleo de sombreador de la GPU 14 de modo que los datos de atributos se puedan leer desde el SGPR, en lugar de un GPR de los elementos de procesamiento.

**[0049]** Para ejecutar las múltiples instancias de un programa de sombreador, el núcleo de sombreador de la GPU 14 incluye una pluralidad de elementos de procesamiento, y cada elemento de procesamiento puede ejecutar una instrucción de una instancia del programa de sombreador. Por ejemplo, cada elemento de procesamiento puede ejecutar una primera instrucción del programa de sombreador al mismo tiempo. En este sentido, cada elemento de procesamiento se puede considerar como la ejecución de un subproceso del programa de sombreador, donde el subproceso es una instrucción del programa de sombreador para un ítem gráfico dado.

**[0050]** En algunos ejemplos, un registro de uso general (GPR) almacena datos para las variables de los ítems gráficos que deben procesar los elementos de procesamiento. Por ejemplo, cada elemento de procesamiento puede ejecutar instrucciones para procesar un ítem gráfico, y el GPR almacena los datos de atributos para los atributos de los ítems gráficos que deben procesarse por los elementos de procesamiento respectivos. Como se describe con más detalle, si los datos de atributos de un atributo son los mismos para todos los ítems gráficos en una onda gráfica, y los datos atribuidos se almacenan en el SGPR, puede haber ganancias computacionales y de eficiencia energética al acceder a los datos del SGPR, en lugar del GPR. Para ayudar a comprender, seguidamente se describe un ejemplo de almacenamiento en un GPR.

**[0051]** La FIG. 2 es un diagrama conceptual que ilustra un ejemplo de almacenamiento de datos en un registro de uso general (GPR) de un núcleo de sombreador de una GPU. Como se ilustra, la GPU 14 incluye el núcleo de sombreador 24, y el núcleo de sombreador 24 incluye el GPR 26. Aunque no se ilustra en la FIG. 2, el núcleo de sombreador 24 también incluye una pluralidad de elementos de procesamiento, donde cada uno ejecuta una instancia del programa de sombreador.

**[0052]** Como ejemplo, el núcleo de sombreador 24 puede incluir treinta y dos elementos de procesamiento y cada uno puede ejecutar una instancia de un programa de sombreador para procesar un ítem gráfico. El GPR 26 puede almacenar datos para los ítems gráficos. Por ejemplo, el GPR 26 puede almacenar datos de atributos para nueve atributos para treinta y dos ítems gráficos. Sin embargo, el GPR 26 puede almacenar datos para más o menos de nueve atributos para los treinta y dos ítems gráficos. Además, el GPR 26 puede almacenar datos que no estén asociados con un atributo de los ítems gráficos, pero son los datos para una variable necesaria para procesar los ítems gráficos.

**[0053]** En el ejemplo ilustrado en la FIG. 2, los ítems gráficos (por ejemplo, las fibras) se identifican como PO-P31, que pueden ser vértices. El atributo se identifica mediante la variable que sigue al identificador de ítem gráfico. Por ejemplo, P0.X se refiere a la coordenada x para el ítem gráfico P0, P0.Y se refiere a la coordenada y para el ítem gráfico P0, etc. P0.R, P0.G, P0.B, y P0.A se refieren al componente rojo, al componente verde, al componente azul y a la opacidad del ítem gráfico P0, respectivamente. Los otros ítems gráficos (por ejemplo, P1-P31) se identifican de forma similar.

**[0054]** En otras palabras, en la FIG. 2, cada uno de los vértices PO-P31 está asociado con una pluralidad de variables. Como ejemplo, cada uno de los vértices PO-P31 está asociado con una variable que identifica la coordenada x (P0.X a P31.X). Cada uno de los vértices PO-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 ítems gráficos. Por ejemplo, la variable que identifica la coordenada x es necesaria para procesar cada uno de los vértices PO-P31.

**[0055]** Como también se ilustra en la FIG. 2, cada uno de los ítems gráficos también incluye un atributo PRJ. El atributo PRJ es una matriz de proyección que puede usar un sombreador de vértices que se ejecute en elementos de procesamiento del núcleo de sombreador 24. En este ejemplo, el atributo PRJ es otra variable que se necesita para procesar cada uno de los vértices PO-P31. Por ejemplo, el sombreador de vértices puede multiplicar la matriz de proyección con las coordenadas respectivas (por ejemplo, multiplicar PO.PRJ con P0.X, P0.Y, P0.Z y P0.W).

**[0056]** En este sentido, se puede considerar que el GPR 26 es tan profundo como ancho. Por ejemplo, GPR 26 puede almacenar datos de atributos para muchos atributos (por ejemplo, profundo) y puede almacenar la fecha de atributo para muchos ítems gráficos (por ejemplo, ancho). Aunque tener GPR profundos y amplios puede ser beneficioso, también puede haber algunos inconvenientes potenciales. Por ejemplo, para un GPR de mayor tamaño, el tiempo necesario para acceder a los valores (por ejemplo, datos) puede ser mayor, causando canales de lectura de datos adicionales en el elemento de procesamiento (por ejemplo, estando el elemento de procesamiento inactivo mientras se leen los datos). Además, acceder al GPR requiere que el elemento de procesamiento, y por lo tanto la GPU 14, consuma energía, y la cantidad de consumo de energía puede ser proporcional al tamaño del GPR.

**[0057]** 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 de sistema 16 o puede almacenar datos en la memoria local (por ejemplo, memoria caché). El GPR 26 del núcleo de sombreador 24 es distinto tanto de la memoria del sistema 16 como de la memoria local de la GPU 14. Por ejemplo, la memoria de sistema 16 es accesible por diversos componentes del dispositivo 10, y estos componentes usan la memoria 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, el GPR 26 solo puede ser accesible por los componentes del núcleo de sombreador 24, y solo puede almacenar datos para los elementos de procesamiento del núcleo de sombreador 24.

**[0058]** Por ejemplo, la memoria caché local de la GPU 14 puede requerir que la GPU 14 determine los aciertos de la memoria caché, y la memoria caché puede estar asociada de diversas formas a la memoria de sistema 16. Esto puede requerir que la GPU 14 realice diversas pruebas de aciertos/errores de la memoria caché, use etiquetas para mantener las direcciones y diversas otras tareas de gestión de almacenamiento con la memoria caché. El GPR 26, por otro lado, puede ser local, interno y dedicado al núcleo de sombreador 24, y puede no necesitar una gestión de memoria especial como la memoria caché local o la memoria de sistema 16.

**[0059]** En algunos casos, los datos de atributos para uno o más atributos de los ítems gráficos almacenados en el GPR 26 pueden ser los mismos. Por ejemplo, cada uno de los ítems gráficos PO-P31 se puede considerar como una fibra, y los ítems gráficos PO-P31 se pueden considerar como una onda (también denominada onda gráfica). En algunos ejemplos, los datos de atributos para uno o más atributos en una onda gráfica pueden ser los mismos.



- 5 **[0060]** Por ejemplo, los ítems gráficos PO-P31 pueden ser vértices y pueden formar una onda de vértice de treinta y dos vértices. En algunos casos, una matriz de proyección (matriz PRJ) puede ser la misma para todos los treinta y dos vértices. Como ejemplo, la aplicación que se ejecuta en el procesador 12 puede determinar que la GPU 14 debe hacer que un tazón gire 90° de modo que el lado abierto del tazón esté orientado hacia el espectador. En este ejemplo, la proyección para muchos de los vértices dentro del tazón puede ser la misma, pero la proyección para algunos de los vértices en el borde (por ejemplo, cerca o en el borde del tazón) puede ser diferente. En este ejemplo, si los ítems gráficos PO-P31 están dentro del bol, entonces la matriz PRJ para PO-P31 es la misma (por ejemplo, PO.PRJ es igual que P1.PRJ-P31.PRJ).
- 10 **[0061]** En algunas técnicas, independientemente de si los datos de atributos para un atributo son los mismos en la onda gráfica, el núcleo de sombreador 24 pide los datos de atributos y almacena los datos de atributos en el GPR 26. Por ejemplo, el núcleo de sombreador 24 ejecuta instrucciones para recuperar los datos de atributos por ítem gráfico para el almacenamiento en el GPR 26. El núcleo de sombreador 24 puede emitir peticiones de memoria para los datos de atributos para cada ítem gráfico (por ejemplo, cada fibra) en serie y almacenar los datos de atributos por ítem gráfico (por ejemplo, por fibra) en el GPR 26. Sin embargo, para una estructura SIMD, tal como la del núcleo de sombreador 24, puede que no sea eficaz en el consumo de energía almacenar datos uniformes por onda en cada ítem gráfico (por ejemplo, por fibra) en el GPR 26 (por ejemplo, almacenar datos de atributos que sean los mismos para los ítems gráficos de una onda gráfica).
- 15 **[0062]** Por ejemplo, para los datos de atributos PO.PRJ a P31.PRJ, en algunas técnicas, el núcleo de sombreador 24 puede emitir peticiones de memoria en serie para PO.PRJ a P31.PRJ. Como ejemplo, una unidad de almacenamiento de memoria externa al núcleo de sombreador 24 puede almacenar la matriz de proyección. Esta unidad de almacenamiento de memoria, externa al núcleo de sombreador 24, puede ser una memoria local de la GPU 14 o incluso la memoria de sistema 16. El núcleo de sombreador 24 puede emitir peticiones de memoria a esta unidad de almacenamiento de memoria para cada ítem gráfico P0-P31 para recibir la matriz de proyección para cada ítem gráfico PO-P31.
- 20 **[0063]** Por ejemplo, el núcleo de sombreador 24 ejecuta una primera instrucción para pedir la matriz de proyección para el ítem gráfico P0 en base a la dirección de memoria donde se almacena la matriz de proyección para el ítem gráfico P0. El núcleo de sombreador 24 ejecuta una segunda instrucción para pedir la matriz de proyección para el ítem gráfico P1 en base a la dirección de memoria donde se almacena la matriz de proyección para el ítem gráfico P1, etc. Si la matriz de proyección es la misma para los ítems gráficos PO-P31, entonces es probable que la dirección de memoria sea la misma, en las instrucciones respectivas que piden la matriz de proyección para cada uno de los ítems gráficos PO-P31.
- 25 **[0064]** El núcleo de sombreador 24 puede recibir entonces cada uno de los datos de atributos PO.PRJ a P31.PRJ y almacenar datos de atributos PO.PRJ a P31.PRJ en GPR 26, como se ilustra en la FIG. 2. En este ejemplo, la matriz PRJ se puede considerar como una carga constante (por ejemplo, datos que son uniformes para los ítems gráficos de una onda gráfica). En muchos casos, la carga constante para una carga de memoria intermedia constante (por ejemplo, para cargar datos de atributos en GPR 26) no es divergente (por ejemplo, PO.PRJ a P31.PRJ son uniformes, lo que significa lo mismo). Sin embargo, el núcleo de sombreador 24 puede pedir y leer no obstante la matriz de proyección múltiples veces y almacenar la misma matriz de proyección que PO.PRJ a P31.PRJ en GPR 26.
- 30 **[0065]** De acuerdo con las técnicas descritas en la presente divulgación, la GPU 14 se puede configurar para determinar si los datos para una o más variables de ítems gráficos en una onda gráfica son uniformes a través de la onda gráfica. En dichos ejemplos, en lugar de almacenar los datos uniformes para una o más variables en el GPR 26, la GPU 14 puede almacenar los datos uniformes en un GPR compartido (SGPR) especializado.
- 35 **[0066]** Por ejemplo, el núcleo de sombreador 24 puede incluir un SGPR. En un ejemplo, el GPR 27 puede incluir una pluralidad de localizaciones de almacenamiento, donde cada localización de almacenamiento está asociada con un atributo de la pluralidad de atributos de los ítems gráficos. Por ejemplo, como se ilustra en la FIG. 2, cada ítem gráfico PO-P31 incluye nueve atributos (PRJ, x, y, z, w, R, G, B y A). En este ejemplo, el SGPR 27 puede incluir nueve localizaciones de almacenamiento, donde la primera localización del SGPR 27 está asociada con el atributo PRJ, la segunda localización del SGPR 27 está asociada con la coordenada x, etc.
- 40 **[0067]** Como ejemplo ilustrativo, la primera localización de almacenamiento del SGPR puede estar asociada con las dos primeras filas del GPR 26 para PO.PRJ a P31.PRJ. La segunda localización de almacenamiento del SGPR puede estar asociada con las dos filas siguientes para el GPR 26 para P0.X a P31.X, etc. Debe entenderse que las dos primeras localizaciones de almacenamiento del SGPR que están asociadas con los dos primeros grupos de dos filas de GPR 26 se proporcionan con propósitos ilustrativos y no deben considerarse limitantes.
- 45 **[0068]** En algunos ejemplos, la primera localización de almacenamiento del SGPR puede estar asociada con la enésima agrupación de dos filas del GPR 26. Como otro ejemplo, en lugar de treinta y dos ítems gráficos en una onda gráfica, puede haber solo dieciséis ítems gráficos en una onda gráfica (por ejemplo, las fibras PO-P15 se procesan en paralelo y las fibras 16-31 no se procesan en paralelo con P0-P15). En este ejemplo, la primera localización de almacenamiento del SGPR puede estar asociada con la primera fila del GPR 26, y así sucesivamente; sin embargo,
- 50 **[0069]** En algunos ejemplos, el núcleo de sombreador 24 puede emitir peticiones de memoria para los datos de atributos para cada ítem gráfico (por ejemplo, cada fibra) en serie y almacenar los datos de atributos por ítem gráfico (por ejemplo, por fibra) en el GPR 26. Sin embargo, para una estructura SIMD, tal como la del núcleo de sombreador 24, puede que no sea eficaz en el consumo de energía almacenar datos uniformes por onda en cada ítem gráfico (por ejemplo, por fibra) en el GPR 26 (por ejemplo, almacenar datos de atributos que sean los mismos para los ítems gráficos de una onda gráfica).
- 55 **[0070]** Por ejemplo, para los datos de atributos PO.PRJ a P31.PRJ, en algunas técnicas, el núcleo de sombreador 24 puede emitir peticiones de memoria en serie para PO.PRJ a P31.PRJ. Como ejemplo, una unidad de almacenamiento de memoria externa al núcleo de sombreador 24 puede almacenar la matriz de proyección. Esta unidad de almacenamiento de memoria, externa al núcleo de sombreador 24, puede ser una memoria local de la GPU 14 o incluso la memoria de sistema 16. El núcleo de sombreador 24 puede emitir peticiones de memoria a esta unidad de almacenamiento de memoria para cada ítem gráfico P0-P31 para recibir la matriz de proyección para cada ítem gráfico PO-P31.
- 60 **[0071]** Por ejemplo, el núcleo de sombreador 24 ejecuta una primera instrucción para pedir la matriz de proyección para el ítem gráfico P0 en base a la dirección de memoria donde se almacena la matriz de proyección para el ítem gráfico P0. El núcleo de sombreador 24 ejecuta una segunda instrucción para pedir la matriz de proyección para el ítem gráfico P1 en base a la dirección de memoria donde se almacena la matriz de proyección para el ítem gráfico P1, etc. Si la matriz de proyección es la misma para los ítems gráficos PO-P31, entonces es probable que la dirección de memoria sea la misma, en las instrucciones respectivas que piden la matriz de proyección para cada uno de los ítems gráficos PO-P31.
- 65 **[0072]** El núcleo de sombreador 24 puede recibir entonces cada uno de los datos de atributos PO.PRJ a P31.PRJ y almacenar datos de atributos PO.PRJ a P31.PRJ en GPR 26, como se ilustra en la FIG. 2. En este ejemplo, la matriz PRJ se puede considerar como una carga constante (por ejemplo, datos que son uniformes para los ítems gráficos de una onda gráfica). En muchos casos, la carga constante para una carga de memoria intermedia constante (por ejemplo, para cargar datos de atributos en GPR 26) no es divergente (por ejemplo, PO.PRJ a P31.PRJ son uniformes, lo que significa lo mismo). Sin embargo, el núcleo de sombreador 24 puede pedir y leer no obstante la matriz de proyección múltiples veces y almacenar la misma matriz de proyección que PO.PRJ a P31.PRJ en GPR 26.
- [0073]** De acuerdo con las técnicas descritas en la presente divulgación, la GPU 14 se puede configurar para determinar si los datos para una o más variables de ítems gráficos en una onda gráfica son uniformes a través de la onda gráfica. En dichos ejemplos, en lugar de almacenar los datos uniformes para una o más variables en el GPR 26, la GPU 14 puede almacenar los datos uniformes en un GPR compartido (SGPR) especializado.
- [0074]** Por ejemplo, el núcleo de sombreador 24 puede incluir un SGPR. En un ejemplo, el GPR 27 puede incluir una pluralidad de localizaciones de almacenamiento, donde cada localización de almacenamiento está asociada con un atributo de la pluralidad de atributos de los ítems gráficos. Por ejemplo, como se ilustra en la FIG. 2, cada ítem gráfico PO-P31 incluye nueve atributos (PRJ, x, y, z, w, R, G, B y A). En este ejemplo, el SGPR 27 puede incluir nueve localizaciones de almacenamiento, donde la primera localización del SGPR 27 está asociada con el atributo PRJ, la segunda localización del SGPR 27 está asociada con la coordenada x, etc.
- [0075]** Como ejemplo ilustrativo, la primera localización de almacenamiento del SGPR puede estar asociada con las dos primeras filas del GPR 26 para PO.PRJ a P31.PRJ. La segunda localización de almacenamiento del SGPR puede estar asociada con las dos filas siguientes para el GPR 26 para P0.X a P31.X, etc. Debe entenderse que las dos primeras localizaciones de almacenamiento del SGPR que están asociadas con los dos primeros grupos de dos filas de GPR 26 se proporcionan con propósitos ilustrativos y no deben considerarse limitantes.
- [0076]** En algunos ejemplos, la primera localización de almacenamiento del SGPR puede estar asociada con la enésima agrupación de dos filas del GPR 26. Como otro ejemplo, en lugar de treinta y dos ítems gráficos en una onda gráfica, puede haber solo dieciséis ítems gráficos en una onda gráfica (por ejemplo, las fibras PO-P15 se procesan en paralelo y las fibras 16-31 no se procesan en paralelo con P0-P15). En este ejemplo, la primera localización de almacenamiento del SGPR puede estar asociada con la primera fila del GPR 26, y así sucesivamente; sin embargo,

como antes, cualquier localización de almacenamiento en el SGPR puede estar asociada con cualquier agrupación de atributos. En otras palabras, una localización de almacenamiento en el SGPR está asociada con una variable para los ítems gráficos en una onda gráfica.

5 **[0069]** De esta manera, el SGPR puede ser más pequeño que el GPR. Por ejemplo, el SGPR puede ser tan profundo como el GPR, pero puede no ser tan amplio. En el ejemplo anterior, el SGPR incluía localizaciones de almacenamiento para cada uno de los atributos y, por lo tanto, el SGPR puede ser tan profundo como el GPR. Sin embargo, cada fila del SGPR se puede configurar para almacenar solo un dato de atributos para un atributo. Por ejemplo, en la FIG. 2, dos filas de GPR 26 pueden almacenar hasta treinta y dos coordenadas x (por ejemplo, treinta y dos datos escalares por onda), si cada coordenada x requiere ocho bits, entonces las dos filas del GPR 26 almacenan 256 bits (832), o 128 bits por fila. Si las coordenadas x son las mismas para cada uno de los ítems gráficos PO-P31, entonces el SGPR puede almacenar un único valor de 8 bits. En consecuencia, donde cada fila del GPR 26 puede tener 128 bits de ancho, cada fila del SGPR puede tener solo 8 bits de ancho.

15 **[0070]** Como se describe anteriormente, si los datos para una variable de los ítems gráficos en una onda gráfica son uniformes, la GPU 14 almacena los datos en el SGPR. Por ejemplo, supongamos que la matriz PRJ es la misma para la onda gráfica. En este ejemplo, un núcleo de sombreador de la GPU 14 puede almacenar los datos de atributos (por ejemplo, la matriz PRJ) en una localización de almacenamiento del SGPR asociado con el atributo (por ejemplo, PRJ) si los datos de atributos del atributo de la pluralidad de ítems gráficos son los mismos. En algunos ejemplos, puede ser posible que la GPU 14 almacene los datos de atributos uniformes en el SGPR y en el GPR 26. Cuando la GPU 14 deba leer los datos de atributos, la GPU 14 puede leer los datos de atributos (por ejemplo, la matriz PRJ) del SGPR, en lugar del GPR.

25 **[0071]** Anteriormente se describió un ejemplo en el que la matriz PRJ se puede almacenar en el SGPR. La matriz PRJ está asociada en general con funciones gráficas. Sin embargo, las técnicas descritas en la presente divulgación no están limitadas así solo a funciones gráficas.

30 **[0072]** Por ejemplo, en algunos ejemplos, las técnicas también se pueden usar para OpenCL por datos constantes de grupo de trabajo (WG) tal como el identificador de grupo de trabajo (ID WG). Por ejemplo, en OpenCL, un programa que se ejecuta en la GPU 14 se denomina kernel, y un kernel se puede dividir en múltiples grupos de trabajo, donde un grupo de trabajo se ejecuta cada vez por elemento de procesamiento de un núcleo de sombreador de la GPU 14 (descrito con más detalle a continuación), como lo indica la ID de WG. El kernel puede usar la ID de WG para calcular cada ID y dirección de ítem de trabajo para cargar/almacenar los datos necesarios para el ítem de trabajo. La ID de WG puede ser constante (por ejemplo, uniforme o igual) para todos los ítems de trabajo en un grupo de trabajo. En algunos ejemplos, la ID de WG se puede cargar en el SGPR para guardar la huella del archivo de GPR y ahorrar energía de acceso.

40 **[0073]** Además, las técnicas descritas en la presente divulgación se pueden extender para propósitos en aplicaciones gráficas distintas al almacenamiento de la matriz PRJ. Por ejemplo, las técnicas descritas en la presente divulgación se pueden extender para la carga del identificador de instancia gráfica (ID) y de los datos de instancia. Por ejemplo, en OpenGL, una función drawInstancePrimitives se puede dividir en múltiples instancias cuando se ejecute (por ejemplo, un compilador, como se describe en la FIG. 3, puede dividir esta función en múltiples instancias). Para cada instancia de drawInstancePrimitives, hay una ID de instancia y algunos datos de atributos (por ejemplo, podrían ser una variedad de tipos de datos de atributos de vértices) que son comunes (por ejemplo, iguales o uniformes) a todos los objetos en la instancia. Un sombreador de vértices puede aplicar estos atributos comunes e ID de instancia y, por lo tanto, puede ser posible almacenar estos atributos comunes e ID de instancia en el SGPR, en lugar del GPR para lograr ahorros de energía, así como una reducción de la cantidad de datos almacenados en el GPR.

50 **[0074]** De esta manera, con el SGPR, las técnicas pueden reducir el número de veces que se debe acceder al GPR 26 tanto para escribir datos como para leerlos. Más bien, cuando esté disponible, las técnicas pueden leer los datos uniformes para una onda gráfica del SGPR. El acceso al SGPR puede requerir menos energía que el acceso al GPR 26 debido al tamaño más pequeño del SGPR. Además, dado que la GPU 14 puede determinar si los datos son uniformes (por ejemplo, los mismos) para una variable de los ítems gráficos en una onda gráfica cuando se pidan los datos, las técnicas pueden reducir el número de veces que deben ejecutarse las instrucciones de carga (por ejemplo, las instrucciones para recuperar datos de atributos).

60 **[0075]** Por ejemplo, las técnicas descritas en la presente divulgación promueven el ahorro de energía al reducir el acceso por fibra para datos uniformes al acceso por onda para datos uniformes. Además, las técnicas descritas en la presente divulgación promueven la mejora del rendimiento al reducir el procesamiento de instrucciones de carga constante por fibra redundante (por ejemplo, reducir la instrucción de carga para datos uniformes), así como reducir los ciclos de emisión y escritura que se desperdician por el almacenamiento repetitivo de datos que es el mismo para toda la onda gráfica.

65 **[0076]** La FIG. 3 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. 3, la GPU 14 incluye el núcleo de sombreador 24, que incluye el registro de uso general (GPR) 26, un GPR compartido (SGPR) 28 y la tubería de función fija 30. El núcleo de sombreador 24 y la

tubería de función fija 30 pueden formar juntos una tubería de procesamiento usada para realizar funciones relacionadas con gráficos o no relacionadas con gráficos. La tubería de procesamiento realiza funciones como se define 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.

5 **[0077]** Como se describe anteriormente, el software o el firmware que se ejecuta en la GPU 14 se puede denominar programas de sombreador (o simplemente sombreadores), y los programas de sombreador se pueden ejecutar en el núcleo de sombreador 24 de la GPU 14. Aunque se ilustra solamente un núcleo de sombreador 24, en algunos ejemplos, la GPU 14 puede incluir uno o más núcleos de sombreador similares al núcleo de sombreador 24. La tubería de función fija 30 incluye las unidades de función fija. El núcleo de sombreador 24 y la tubería de función fija 30 pueden transmitir y recibir datos entre sí. Por ejemplo, la tubería de procesamiento puede incluir programas de sombreador que se ejecuten en el núcleo de sombreador 24 que reciban datos de una unidad de función fija de la tubería de función fija 30 y envíen datos procesados a otra unidad de función fija de la tubería de función fija 30.

15 **[0078]** Los programas de sombreador proporcionan a los usuarios flexibilidad funcional porque un usuario puede diseñar el programa de sombreador para realizar las tareas deseadas de cualquier manera concebible. Sin embargo, las unidades de función fija están cableadas por la manera en cual las unidades de función fija realizan tareas. En consecuencia, las unidades de función fija pueden no proporcionar mucha flexibilidad funcional.

20 **[0079]** Los ejemplos de los programas de sombreador incluyen el sombreador de vértices 32, el sombreador de fragmentos 34 y el sombreador de cálculo 36. El sombreador de vértices 32 y el sombreador de fragmentos 34 pueden ser programas de sombreador para tareas relacionadas con gráficos, y el sombreador de cálculo 36 puede ser un programa de sombreador para una tarea no relacionada con gráficos. En algunos ejemplos, solo se pueden usar programas de sombreador relacionados con gráficos como el sombreador de vértices 32 y el sombreador de fragmentos 34. En algunos ejemplos, solo se pueden usar programas de sombreador no relacionados con gráficos como el sombreador de cálculo 36. Hay ejemplos adicionales de programas de sombreador tales como sombreadores de geometría, que no se describen para propósitos de brevedad.

30 **[0080]** El controlador gráfico 40 que se ejecuta en el procesador 12 se puede configurar para implementar una interfaz de programación de aplicaciones (API). En dichos ejemplos, los programas de sombreador (por ejemplo, el sombreador de vértices 32, el sombreador de fragmentos 34 y el sombreador de cálculo 36) se pueden configurar de acuerdo con la misma API que el controlador gráfico 40. Aunque no se ilustra, la memoria de sistema 16 puede almacenar el código para el controlador gráfico 40 que el procesador 12 recupera de la memoria de sistema 16 para su ejecución. El controlador gráfico 40 se ilustra en una casilla discontinua para indicar que el controlador gráfico 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 gráfico 40 se puede implementar como hardware en el procesador 12.

35 **[0081]** En algunos ejemplos, la memoria de sistema 16 puede almacenar el código fuente para uno o más del sombreador de vértices 32, del sombreador de fragmentos 34 y del sombreador de cálculo 36. En estos ejemplos, el compilador 38 que se ejecuta en el procesador 12 puede compilar el código fuente de estos programas de sombreador para crear un código objeto o intermedio ejecutable por el núcleo de sombreador 24 de la GPU 14 durante el tiempo de ejecución (por ejemplo, en el momento en que estos programas de sombreador necesiten ejecutarse en el núcleo de sombreador 24). En algunos ejemplos, el compilador 38 puede compilar previamente los programas de sombreador y almacenar el objeto o código intermedio de los programas de sombreador en la memoria de sistema 16.

45 **[0082]** La memoria de sistema 16 puede almacenar el código para el compilador 38 que el procesador 12 recupera de la memoria de sistema 16 para su ejecución. En el ejemplo de la FIG. 3, 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 se puede implementar como hardware en el procesador 12, en algunos ejemplos.

50 **[0083]** El controlador gráfico 40 se puede configurar para permitir que el procesador 12 y la GPU 14 se comuniquen entre sí. Por ejemplo, cuando el procesador 12 descarga tareas de procesamiento gráfico o sin gráficos a la GPU 14, el procesador 12 descarga dichas tareas de procesamiento a la GPU 14 por medio del controlador gráfico 40.

55 **[0084]** Como ejemplo, el procesador 12 puede ejecutar una aplicación de juegos que produzca datos gráficos, y el procesador 12 puede descargar el procesamiento de estos datos gráficos a la GPU 14. En este ejemplo, el procesador 12 puede almacenar los datos gráficos en la memoria de sistema 16, y el controlador gráfico 40 puede indicar a la GPU 14 cuándo recuperar los datos gráficos, desde dónde recuperar los datos gráficos en la memoria de sistema 16 y cuándo procesar los datos gráficos. Además, la aplicación de juegos puede requerir que la GPU 14 ejecute uno o más programas de sombreador. Por ejemplo, la aplicación de juegos puede requerir que el núcleo de sombreador 24 ejecute el sombreador de vértices 32 y el sombreador de fragmentos 34 para generar imágenes que se vayan a visualizar (por ejemplo, en la pantalla 18 de la FIG. 1). El controlador gráfico 40 puede indicar a la GPU 14 cuándo ejecutar los programas de sombreador e indicar a la GPU 14 dónde recuperar los datos gráficos necesarios para los programas de sombreador. De esta forma, el controlador gráfico 40 puede formar el enlace entre el procesador 12 y la GPU 14.

**[0085]** El controlador gráfico 40 se puede configurar de acuerdo con una API; aunque el controlador gráfico 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 gráfico 40 se puede configurar de acuerdo con la API de OpenGL ES. La API de OpenGL ES está diseñada específicamente para dispositivos móviles. En un ejemplo en el que el dispositivo 10 es un dispositivo no móvil, el controlador gráfico 40 se puede configurar de acuerdo con la API de OpenGL.

**[0086]** El núcleo de sombreador 24 se puede configurar para ejecutar muchas instancias de la misma instrucción del mismo programa de sombreador en paralelo. Por ejemplo, el controlador gráfico 40 puede indicar a la GPU 14 que recupere valores de vértice para una pluralidad de vértices e indicar a la GPU 14 que ejecute el sombreador de vértices 32 para procesar los valores de vértices de los vértices. En este ejemplo, el núcleo de sombreador 24 puede ejecutar múltiples instancias del sombreador de vértices 32 y hacerlo ejecutando una instancia del sombreador de vértices 32 en un elemento de procesamiento del núcleo de sombreador 24 para cada uno de los vértices.

**[0087]** Cada elemento de procesamiento del núcleo de sombreador 24 puede ejecutar la misma instrucción del sombreador de vértices 32 en la misma instancia; sin embargo, los valores de vértice particulares pueden ser diferentes porque cada elemento de procesamiento procesa un vértice diferente. Como se describe anteriormente, cada elemento de procesamiento se puede considerar como la ejecución de un subproceso del sombreador de vértices 32, donde un subproceso se refiere a una instrucción del sombreador de vértices 32 que está procesando un vértice particular. De esta manera, el núcleo de sombreador 24 puede ejecutar muchas instancias del sombreador de vértices 32 para procesar valores de vértice de una pluralidad de vértices en paralelo (por ejemplo, al mismo tiempo).

**[0088]** El núcleo de sombreador 24 puede ejecutar de forma similar muchas instancias del sombreador de fragmentos 34 para procesar valores de píxeles de una pluralidad de píxeles en paralelo o ejecutar muchas instancias del sombreador de cálculo 36 para procesar muchos datos no relacionados con gráficos en paralelo. De esta manera, el núcleo de sombreador 24 se puede configurar en una estructura de instrucción única, múltiples datos (SIMD). Para facilitar la descripción, se describe lo siguiente con respecto a un programa de sombreador genérico, cuyos ejemplos incluyen el sombreador de vértices 32, el sombreador de fragmentos 34, el sombreador de cálculo 36 y otros tipos de sombreadores tales como los sombreadores de geometría.

**[0089]** En las técnicas descritas en la presente divulgación, el SGPR 28 puede almacenar datos (por ejemplo, datos de atributos) para una variable (por ejemplo, un atributo) para ítems gráficos en una onda gráfica que sean uniformes. Puede haber varias formas de determinar si los datos son uniformes en la onda gráfica.

**[0090]** Por ejemplo, el compilador 38 puede conocer el SGPR 28 (es decir, el SGPR 28 es visible para el compilador 38). En otras palabras, el compilador 38 está diseñado con funcionalidad que usa el SGPR 28. En algunos ejemplos, el compilador 38 puede garantizar que, en lugar de almacenar datos uniformes por onda en el GPR 26 por fibra, el núcleo de sombreador 24 almacena datos uniformes por onda en SGPR 28 (por ejemplo, treinta y dos datos escalares por onda). El compilador 38 puede usar diferentes técnicas de ejemplo para permitir una carga constante de datos no divergentes en el SGPR 28.

**[0091]** Como ejemplo, durante la compilación de un programa de sombreador (por ejemplo, el programa de sombreador de vértices 32, el programa de sombreador de fragmentos 34 o el programa de sombreador de cálculo 36), el compilador 38 puede determinar si el programa de sombreador requiere acceder a datos uniformes. En este contexto, un programa de sombreador que requiere acceso a datos uniformes significa que cada subproceso del programa de sombreador que se ejecuta en elementos de procesamiento respectivos del núcleo de sombreador 24 pide cada uno los mismos datos para al menos una variable de los ítems gráficos.

**[0092]** Por ejemplo, el compilador 38 puede generar datos uniformes tal como un contador de bucle for. Este contador de bucle for puede ser el mismo para todas las instancias de ejecución del programa de sombreador, lo que significa que el contador de bucle for es el mismo para procesar cada ítem gráfico. Este contador de bucle for puede ser un ejemplo de datos de una variable para los ítems gráficos. En este ejemplo, el compilador 38 puede determinar que el contador de bucle for son datos para una variable de los ítems gráficos que son uniformes en todos los ítems gráficos en la onda gráfica.

**[0093]** Como otro ejemplo, el programa de sombreador puede incluir instrucciones de ramificación (por ejemplo, instrucciones si-entonces-si no). Para alguna ramificación particular, el programa de sombreador puede incluir una instrucción que establezca una o más variables iguales a un valor constante (por ejemplo, una instrucción que establezca 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 ítems gráficos que son uniformes en todos los ítems gráficos en la onda gráfica.

**[0094]** Si el compilador 38 determina que el programa de sombreador requiere acceder a datos uniformes a través de la onda gráfica, el compilador 38 puede incluir una instrucción explícita que indique al núcleo de sombreador 24 que los datos son uniformes a través de la onda gráfica. En este ejemplo, el núcleo de sombreador 24 puede emitir una petición de datos uniformes desde la memoria (por ejemplo, memoria local o memoria de sistema 16) para la

variable para un primer ítem gráfico, y puede almacenar los datos en el SGPR 28 en lugar del GPR 26. Sin embargo, debido a que los datos para la variable son los mismos para la onda gráfica, el núcleo de sombreador 24 puede no emitir una petición para otros ítems gráficos. Por ejemplo, el núcleo de sombreador 24 puede convertir la carga constante por ítem gráfico (por ejemplo, por fibra) en una carga constante por onda, y almacenar los datos en el SGPR 28. De esta manera, el núcleo de sombreador 24 puede no necesitar emitir peticiones de memoria innecesarias, lo que ahorra ciclos de procesamiento de instrucciones por fibra, ciclos de emisión de peticiones de memoria y ciclos de retorno de escritura de datos.

**[0095]** Como se describe anteriormente, el SGPR 28 puede incluir una pluralidad de localizaciones de almacenamiento, donde cada localización de almacenamiento está asociada con una sola variable (por ejemplo, atributo) de las variables (por ejemplo, atributos) de los ítems gráficos que forman la onda gráfica. Como también se describe anteriormente, el compilador 38 puede ver el SGPR 28. Por ejemplo, el compilador 38 puede asignar a cada una de las localizaciones de almacenamiento del SGPR 28 con un identificador único (ID de SGPR). El compilador 38 puede transmitir la ID de SGPR para la localización de almacenamiento asociada con el atributo que tenga los datos de atributos uniformes. El núcleo de sombreador 24 puede almacenar entonces los datos de atributos uniformes para la onda gráfica en la localización de almacenamiento del SGPR 28 identificada por la ID de SGPR.

**[0096]** En estos ejemplos, para las instrucciones que usan los datos, los elementos de procesamiento del núcleo de sombreador 24 pueden recuperar los datos de la ID de SGPR asociada con la localización de almacenamiento para el atributo. En dichos ejemplos, las técnicas pueden reducir aún más el consumo de energía al no acceder al GPR 26 y acceder al SGPR 28 en su lugar.

**[0097]** En los ejemplos donde el compilador 38 determina que los datos de los ítems gráficos de una onda gráfica son divergentes (por ejemplo, no uniformes), el compilador 38 puede no incluir las instrucciones que indiquen el acceso uniforme a los datos. En estos ejemplos, el núcleo de sombreador 24 puede pedir los datos para cada uno de los ítems gráficos y almacenar los datos en el GPR 26. Por ejemplo, el núcleo de sombreador 24 puede pedir a la memoria los datos de atributos para el primer atributo de un primer ítem gráfico, pedir a la memoria los datos de atributos para el primer atributo de un segundo ítem gráfico, etc. En el escenario de datos uniformes de atributos, el núcleo de sombreador 24 puede pedir datos de atributos solo una vez para los ítems gráficos de la onda gráfica y almacenar los datos de atributos en el SGPR 28.

**[0098]** En algunos casos, el compilador 38 puede no determinar que los datos para una variable sean uniformes a través de la onda gráfica, pero también puede no determinar que los datos para una variable son necesariamente divergentes en la onda gráfica. En otras palabras, el compilador 38 puede determinar que existe la probabilidad de que los datos de atributos para un atributo sean uniformes (por ejemplo, "QUIZÁS" haya un acceso uniforme para los datos de atributos de un atributo particular).

**[0099]** En dichos ejemplos, el compilador 38 puede solapar las localizaciones de almacenamiento del SGPR 28 con los atributos de los ítems gráficos almacenados en el GPR 26, e indicar dicho solapamiento al núcleo de sombreador 24. Como ejemplo, supongamos que cada ítem gráfico está asociado con ocho atributos. En este ejemplo, el compilador 38 puede solapar los identificadores (ID) R48-R55 con ID de GPR 0-7 (por ejemplo, con los atributos 0-7 del ítem gráfico). En este ejemplo, las ID R48-R55 pueden ser identificadores reservados, y el núcleo de sombreador 24 se puede configurar para determinar que, cuando se usen los ID R48-R55, el compilador 38 determinó que existe la probabilidad de que los datos de atributos para al menos uno de los atributos de los ítems gráficos sean los mismos.

**[0100]** El solapamiento de las localizaciones de almacenamiento del SGPR 28 y del GPR 26 puede significar que el compilador 38 puede acceder y asignar tanto el espacio de SGPR como el espacio de GPR para el destino. En otras palabras, el compilador 38 puede permitir que los datos se almacenen en el SGPR 28 o en el GPR 26. Por ejemplo, el compilador 38 puede incluir instrucciones en el programa de sombreador que indiquen que los datos de atributos deben almacenarse en uno de estos identificadores. Si el núcleo de sombreador 24 determina que los datos de atributos son uniformes, el núcleo de sombreador 24 puede almacenar los datos de atributos en la localización de almacenamiento del SGPR 28 identificado por el identificador de solapamiento. Si el núcleo de sombreador 24 determina que los datos de atributos son divergentes, el núcleo de sombreador 24 puede almacenar los datos de atributos en el GPR 26 en base al identificador de solapamiento. Un ejemplo del uso del identificador de solapamiento se describe con más detalle con respecto a las FIGS. 5A y 5B.

**[0101]** Como se describe anteriormente, el núcleo de sombreador 24 se puede configurar para determinar si los datos son uniformes o divergentes. Puede haber diversas formas en que el núcleo de sombreador 24 puede determinar si los datos son uniformes o divergentes. Como ejemplo, el núcleo de sombreador 24 puede monitorear las direcciones de memoria pedidas por cada uno de los elementos de procesamiento para cada una de las variables. Estas direcciones de memoria pueden ser para una dirección de memoria de la memoria local de la GPU 14 o para una dirección de memoria de la memoria de sistema 16.

**[0102]** Si las direcciones de memoria son las mismas, el núcleo de sombreador 24 puede determinar que los datos para una variable de cada uno de los ítems gráficos, que se procesan por los elementos de procesamiento respectivos del núcleo de sombreador 24, son uniformes. Por ejemplo, si la dirección de memoria de donde se van a recuperar los

datos es la misma para cada uno de los ítems gráficos, entonces es probable que sean los mismos los datos que se pidan por los elementos de procesamiento que procesan los ítems gráficos.

5 **[0103]** En algunos ejemplos, si el núcleo de sombreador 24 determina que las direcciones de memoria para los datos de una variable de los ítems gráficos de una onda gráfica es el mismo, el núcleo de sombreador 24 puede afirmar un indicador válido uniforme (por ejemplo, establecer un indicador válido uniforme igual a 1) que indique que son uniformes los datos de una variable. Luego, al escribir los datos pedidos, el núcleo de sombreador 24 puede determinar primero el valor del indicador válido uniforme y almacenar los datos uniformes (por ejemplo, una instancia de los datos) en la localización de almacenamiento del SGPR 28 identificada por el identificador de solapamiento si se afirma un  
10 indicador válido uniforme.

15 **[0104]** El indicador válido uniforme está asociado con la localización de almacenamiento del SGPR 28 asociada con el identificador de solapamiento. Por ejemplo, cada localización de almacenamiento del SGPR 28 también se puede asociar con un indicador válido uniforme. Si los datos de atributos para un atributo asociado con una localización de almacenamiento particular del SGPR 28 son uniformes, según lo determinado por el núcleo de sombreador 24, el núcleo de sombreador 24 puede afirmar (por ejemplo, establecer el valor del indicador en 1) el indicador válido uniforme asociado con esa localización de almacenamiento. Si los datos de atributos para un atributo asociado con una localización de almacenamiento particular del SGPR 28 son divergentes, según lo determinado por el núcleo de sombreador 24, el núcleo de sombreador 24 puede no afirmar el indicador válido uniforme (por ejemplo, establecer un  
20 indicador válido uniforme igual a 0) asociado con esa localización de almacenamiento.

25 **[0105]** Por ejemplo, si el núcleo de sombreador 24 determina que al menos una dirección de memoria para los datos de atributos de un atributo de los ítems gráficos no es la misma que las otras, el núcleo de sombreador 24 puede no afirmar el valor uniforme. Luego, al escribir los datos de atributos pedidos, el núcleo de sombreador 24 puede determinar primero el valor del indicador válido uniforme y almacenar los datos de atributos (por ejemplo, los datos de atributos divergentes de un atributo de cada uno de los ítems gráficos) en las localizaciones de almacenamiento del GPR 26 asociado con el identificador de solapamiento si no se afirma el indicador válido uniforme.

30 **[0106]** En algunos ejemplos, para las instrucciones del programa de sombreador que luego usan los datos de atributos, el compilador 38 puede incluir como operando el identificador de solapamiento de la localización de almacenamiento del SGPR 28 asociado con los datos de atributos en estas instrucciones. El núcleo de sombreador 24 lee luego el valor del indicador válido uniforme y determina si los datos de atributos se almacenan en el SGPR 28 o en el GPR 26. Por ejemplo, si se afirma el indicador válido uniforme, el núcleo de sombreador 24 puede determinar que los datos de atributos se almacenan en el SGPR 28, y, si el indicador válido uniforme no se afirma, el núcleo de sombreador 24 puede determinar que los datos de atributos se almacenan en el GPR 26.  
35

40 **[0107]** La FIG. 4 es un diagrama de bloques que ilustra un ejemplo de un núcleo de sombreador de una GPU de la FIG. 3 en mayor detalle. Por ejemplo, la FIG. 4 ilustra un ejemplo del núcleo de sombreador 24 de la GPU 14. El núcleo de sombreador 24 puede incluir la unidad de control 42, lo que incluye la unidad de detección de lógica uniforme 44, el almacenamiento de instrucciones 46, uno o más elementos de procesamiento 48A-48H (denominados colectivamente como "elementos de procesamiento 48"), el GPR 26 y el SGPR 28. Aunque la FIG. 4 ilustra ocho elementos de procesamiento 48, puede haber más o menos de ocho elementos de procesamiento 48 en otros ejemplos. Los elementos de procesamiento 48 son ejemplos de elementos de procesamiento en los cuales se ejecutan en paralelo instancias del programa de sombreador.  
45

50 **[0108]** La unidad de control 42 puede controlar la funcionalidad del núcleo de sombreador 24. Por ejemplo, la unidad de control 42 puede recuperar las instrucciones que se vayan a ejecutar por los elementos de procesamiento 48 y almacenar las instrucciones en el almacenamiento de instrucciones 46. Además, la unidad de control 42 puede recuperar los datos de atributos que los elementos de procesamiento 48 deben procesar y almacenar los datos de atributos en el GPR 26 y/o en el SGPR 28.

55 **[0109]** Por ejemplo, como se ilustra, la unidad de control 42 incluye la unidad de detección lógica uniforme 44. La unidad de detección lógica uniforme 44 puede ser hardware del núcleo de sombreador 24, software que se ejecute en el núcleo de sombreador 24, o una combinación de ambos. Aunque la unidad de detección lógica uniforme 44 se ilustra como parte de la unidad de control 42, las técnicas descritas en la presente divulgación no están tan limitadas. En general, la unidad de detección lógica uniforme 44 se ilustra para ayudar a comprender, y puede ser una unidad separada de la unidad de control 42, una unidad especializada dentro de la unidad de control 42, o integrada con la unidad de control 42.

60 **[0110]** En las técnicas descritas en la divulgación, la unidad de detección lógica uniforme 44 es un ejemplo de la unidad configurada para implementar las técnicas de ejemplo. Por ejemplo, cada uno de los elementos de procesamiento 48 puede procesar un ítem gráfico. Para procesar los ítems gráficos respectivos, cada uno de los elementos de procesamiento 48 puede ejecutar instancias del programa de sombreador.

65 **[0111]** Por ejemplo, cada uno de los elementos de procesamiento 48 puede ejecutar una instrucción que pida datos (por ejemplo, datos de atributos) para una variable (por ejemplo, atributo) de cada uno de los ítems gráficos procesados

por los elementos de procesamiento 48 respectivos. La unidad de detección lógica uniforme 44 puede determinar si las instrucciones que piden los datos están pidiendo los datos de la misma dirección de memoria, donde la dirección de memoria es para la memoria local de la GPU 14 o la memoria de sistema 16. Si la unidad de detección lógica uniforme 44 determina que la dirección de memoria para los datos para la variable de cada uno de los ítems gráficos es la misma, la unidad de detección lógica uniforme 44 puede emitir la petición de los datos una vez y recuperar los datos una vez, en lugar de recuperar datos múltiples veces para cada petición múltiple de los datos.

**[0112]** La unidad de detección lógica uniforme 44 también puede afirmar el indicador válido uniforme para la localización de almacenamiento dentro del SGPR 28 asociado con la variable y escribir los datos recuperados en la localización de almacenamiento del SGPR 28. En algunos ejemplos, la unidad de detección lógica uniforme 44 puede no escribir los datos en el GPR 26 (aunque puede ser posible escribir todavía en el GPR 26). Entonces, cuando cada uno de los elementos de procesamiento 48 ejecute una instrucción que requiera los datos de la variable, la unidad de control 42 puede recuperar los datos de la localización de almacenamiento del SGPR 28 asociada con la variable. Por ejemplo, solo se puede necesitar una petición al SGPR 28 para recuperar los datos de todos los elementos de procesamiento 48 a la vez.

**[0113]** Si la unidad de detección lógica uniforme 44 determina que la dirección de memoria para al menos uno de los datos pedidos es diferente (por ejemplo, al menos dos de las direcciones de memoria son diferentes entre sí), la unidad de detección lógica uniforme 44 puede emitir cada una de las peticiones para los datos y puede no almacenar los datos en el SGPR 28, y en su lugar puede almacenar los datos en el GPR 26. Entonces, cuando cada uno de los elementos de procesamiento 48 ejecute una instrucción que requiere los datos de la variable, la unidad de control 42 puede recuperar los datos de las localizaciones de almacenamiento del GPR 26. Por ejemplo, pueden ser necesarias múltiples peticiones (por ejemplo, una para cada elemento de procesamiento 48) para recuperar los datos para cada uno de los elementos de procesamiento 48.

**[0114]** El almacenamiento de instrucciones 46 puede ser cualquier tipo de memoria capaz de almacenar instrucciones tales como, pero sin limitarse a, memoria volátil, memoria no volátil, memoria caché, memoria de acceso aleatorio (RAM), RAM estática (SRAM), RAM dinámica (DRAM), etc. La unidad de control 42 puede ser circuitos cableados del núcleo de sombreador 24 que controle los componentes del núcleo de sombreador 24. Sin embargo, puede ser posible que la unidad de control 42 sea software o firmware, que se ejecute en hardware, del núcleo de sombreador 24.

**[0115]** Los elementos de procesamiento 48 están configurados para ejecutar subprocesos de un programa de sombreador. Cada uno de los elementos de procesamiento 48 puede ejecutar un subproceso diferente. Por ejemplo, cada uno de los elementos de procesamiento 48 puede ejecutar una instancia de una instrucción de un programa de sombreador con respecto a ítems gráficos potencialmente diferentes. Los elementos de procesamiento 48 pueden ser elementos de procesamiento de instrucción única, múltiples datos (SIMD). Los elementos de procesamiento SIMD se refieren a elementos de procesamiento que, cuando se activan, están configurados para ejecutar la misma instrucción al mismo tiempo con respecto a datos diferentes. Esto puede permitir que los elementos de procesamiento 48 ejecuten una pluralidad de subprocesos de un programa de sombreador en paralelo con respecto a diferentes ítems de datos. En algunos casos, cada uno de los elementos de procesamiento 48 puede ejecutar instrucciones de un programa de sombreador en base a un contador de programa común que apunte a una instrucción contenida en el almacenamiento de instrucciones 46.

**[0116]** Si uno o más de los elementos de procesamiento 48 se desactivan por la unidad de control 42, dichos elementos de procesamiento 48 no ejecutan una instrucción de programa para un ciclo de instrucción dado. En algunos casos, la unidad de control 42 puede desactivar uno o más de los elementos de procesamiento 48 para implementar instrucciones de ramificación condicional donde la condición de ramificación se satisfaga para algunos subprocesos y no se satisfaga para otros subprocesos.

**[0117]** En algunos ejemplos, cada uno de los elementos de procesamiento 48 puede incluir y/o corresponder a una o más unidades lógicas aritméticas (ALU). En ejemplos adicionales, cada uno de los elementos de procesamiento 48 puede implementar la funcionalidad ALU. La funcionalidad ALU puede incluir suma, resta, multiplicación, etc.

**[0118]** Las FIGS. 5A y 5B son diagramas conceptuales que ilustran ejemplos de almacenamiento de datos de acuerdo con las técnicas descritas en la presente divulgación. En el ejemplo de las FIGS. 5A y 5B, cada ítem gráfico está asociado con ocho atributos. Por ejemplo, la FIG. 5A ilustra un ejemplo en el que cuatro ítems gráficos P0-P3 forman una primera onda gráfica, y la FIG. 5B ilustra un ejemplo en el que cuatro ítems gráficos P4-P7 forman una segunda onda gráfica.

**[0119]** Debe entenderse que las FIGS. 5A y 5B ilustran solo una porción del GPR 26, y en particular solo la porción del GPR 26 que se asigna al SGPR 28. En general, el GPR 26 está configurado para almacenar datos adicionales que no se asignen a una localización de almacenamiento del SGPR 28. Por ejemplo, el GPR es un registro general usado para una variedad de propósitos de almacenamiento, mientras que el SGPR 28 se puede usar, en algunos ejemplos no limitantes, para almacenar solo datos que sean uniformes o que tengan la probabilidad de ser uniformes en una onda gráfica. En otras palabras, de todos los datos que se almacenan en el GPR 26, puede ser posible que una parte

de dichos datos se almacene en el SGPR 28. Para facilitar la ilustración, las FIGS. 5A y 5B ilustran los datos del GPR 26 que se pueden almacenar en el SGPR 28 (donde, en la FIG. 5A, los datos son uniformes para el almacenamiento en el SGPR 28, y donde, en la FIG. 5B, los datos no son uniformes y no se pueden almacenar en el SGPR 28).

5 **[0120]** En estos ejemplos, cada uno de los ítems gráficos P0-P3 está asociado con las variables R0-R7 y, de forma similar, cada uno de los ítems gráficos P4-P7 está asociado con las variables R0-R7. En los ejemplos ilustrados en las FIGS. 5A y 5B, los ID R48-R55 están asociados con las variables R0-R7, respectivamente. Debe entenderse que las ID R48-R55 asociadas con las variables R0-R7 son simplemente un ejemplo, y las ID R48-R55 pueden estar asociadas con cualquiera de las variables R0-R7. En la FIG. 5A, W0 se refiere a la onda gráfica que incluye ítems gráficos P0-P3 y, en la FIG. 5B, W1 se refiere a la onda gráfica que incluye ítems gráficos P4-P7.

**[0121]** En las FIGS. 5A y 5B, los elementos de procesamiento 48A-48D (solo pueden necesitarse cuatro elementos de procesamiento 48) del núcleo de sombreador 24 pueden implementar las siguientes instrucciones:

15 LDC R48, m[x]; // localización de memoria constante de carga x a R48;  
Añadir R1, R48, 0x1; // R1 = R48 +1

**[0122]** En el ejemplo de la FIG. 5A, la primera localización de almacenamiento del SGPR 28 está asociada con la variable R0 y tiene la ID de SGPR de R48. En el ejemplo de la FIG. 5B, la primera localización de almacenamiento del SGPR 28 está asociada con la variable R0 y tiene la ID de SGPR de R48. En la FIG. 5A, cuatro elementos de procesamiento 48 del núcleo de sombreador 24 están procesando cada uno los ítems gráficos P0-P3 respectivos al mismo tiempo y, en la FIG. 5B, cuatro elementos de procesamiento 48 del núcleo de sombreador 24 están procesando cada uno ítems gráficos P4-P7 al mismo tiempo.

25 **[0123]** En los ejemplos de las FIGS. 5A y 5B, cada uno de los cuatro elementos de procesamiento 48 del núcleo de sombreador 24 ejecuta la instrucción LDC R48, m[x] para emitir una petición para cargar el valor almacenado en la localización de memoria x. La dirección de memoria para la localización de memoria x es para los datos de la variable R0 para los ítems gráficos P0-P3 en la FIG. 5A y para los datos de la variable R0 para los ítems gráficos P4-P7. La unidad de detección lógica uniforme 44 puede determinar si la dirección para la localización de memoria x es la misma para todas las peticiones de memoria de los cuatro elementos de procesamiento 48.

35 **[0124]** Supongamos que, en el ejemplo de la FIG. 5A, la dirección para la localización de memoria x es la misma para los cuatro elementos de procesamiento 48 y, en el ejemplo de la FIG. 5B, la dirección para la localización de memoria x es diferente para al menos uno de los cuatro elementos de procesamiento 48 (por ejemplo, hay al menos dos direcciones que son diferentes entre sí). Por consiguiente, en el ejemplo de la FIG. 5A, el núcleo de sombreador 24 (por ejemplo, la unidad de detección lógica uniforme 44 de la unidad de control 42) puede determinar que la dirección de memoria para los datos de la variable R0 es la misma para los ítems gráficos P0-P3 y, en el ejemplo de la FIG. 5B, el núcleo de sombreador 24 (por ejemplo, la unidad de detección lógica uniforme 44 de la unidad de control 42) puede determinar que la dirección de memoria para los datos de al menos una variable R0 es diferente para los ítems gráficos P4-P7 (por ejemplo, la dirección de memoria para los datos de al menos dos variables R0 es diferente entre sí).

45 **[0125]** En el ejemplo ilustrado en la FIG. 5A, el núcleo de sombreador 24 (por ejemplo, por medio de la unidad de control 42) puede cargar los datos de la variable R0 en la primera localización de almacenamiento del SGPR 28, que se identifica como R48, en lugar de R0 en localizaciones de almacenamiento del GPR 26 reservadas para los ítems gráficos P0 -P3. En este ejemplo, el núcleo de sombreador 24 (por ejemplo, por medio de la unidad de control 42) puede emitir una petición de lectura, y no emitir una petición de lectura para cada uno de los elementos de procesamiento que procesen los respectivos ítems gráficos P0-P3. Además, en este ejemplo, el núcleo de sombreador 24 (por ejemplo, por medio de la unidad de control 42) puede escribir los datos para la variable R0 una vez, y no escribir los datos para la variable R0 cuatro veces. Luego, para la siguiente instrucción (R1 = R48 + 1), los elementos de procesamiento 48A-48D del núcleo de sombreador 24 pueden leer el valor de la variable R0 del SGPR 28, en lugar de leer los datos cuatro veces del GPR 26. De esta manera, las técnicas pueden ahorrar ciclos de emisión de peticiones de memoria y devolver datos escribiendo ciclos si la carga constante en una onda es uniforme.

55 **[0126]** En el ejemplo ilustrado en la FIG. 5B, el núcleo de sombreador 24 (por ejemplo, por medio de la unidad de control 42) puede cargar los datos de cada variable R0 identificada por la dirección de memoria de la localización de memoria x en localizaciones de almacenamiento del GPR 26. Por ejemplo, el núcleo de sombreador 24 (por ejemplo, por medio de la unidad de control 42) puede ejecutar cada una de las cuatro peticiones de los datos de atributos para la variable R0 para los ítems gráficos P4-P7 porque la localización de memoria x es diferente para al menos uno de los cuatro ítems gráficos P4-P7. Luego, para la siguiente instrucción (R1 = R48 + 1), los elementos de procesamiento 48A-48D del núcleo de sombreador 24 pueden leer los datos respectivos de la variable R0 del GPR 26 para cada localización de almacenamiento respectiva para los ítems gráficos P4-P7.

65 **[0127]** Como se describe anteriormente, el SGPR 28 es más pequeño que el GPR 26. Como ejemplo, el SGPR 28 puede incluir una columna y el mismo número de filas que el GPR 26. De esta manera, el SGPR 28 puede ser tan profundo como el GPR 26, pero puede no ser tan ancho. Como se ilustra en las FIGS. 5A y 5B, una fila del SGPR 28



puede almacenar treinta y dos bits, pero una fila del GPR 26 puede almacenar 128 bits. Como otro ejemplo, puede ser posible que el SGPR 28 incluya una pluralidad de columnas, donde el número de columnas se basa en el número de atributos, e incluye una fila. En general, cada localización de almacenamiento del SGPR 28 está asociada con una variable y se puede configurar para almacenar los datos para la variable respectiva de los ítems gráficos respectivos si los datos para las variables respectivas son los mismos.

**[0128]** La FIG. 6 es un diagrama de flujo que ilustra una técnica de ejemplo para almacenar datos. Como se ilustra en la FIG. 6, el núcleo de sombreador 24 (por ejemplo, por medio de la unidad de detección lógica uniforme 44 de la unidad de control 42) puede determinar si los datos de una variable de una pluralidad de ítems gráficos son los mismos (100). En este ejemplo, la pluralidad de ítems gráficos se procesa con elementos de procesamiento 48 respectivos del núcleo de sombreador 24 de la GPU 14 durante la ejecución de un programa de sombreador (por ejemplo, el sombreador de vértices 32, el sombreador de fragmentos 34 o el sombreador de cálculo 36).

**[0129]** En general, la variable de una pluralidad de ítems gráficos se refiere a una variable que se necesita para procesar cada una de la pluralidad de ítems gráficos. Como ejemplo, cada uno de la pluralidad de ítems gráficos puede estar asociado con una coordenada x, y, z y w, un valor de componente rojo, un valor de componente verde, un valor de componente azul y un valor de opacidad. En este ejemplo, las coordenadas x, y, z y w son cada una un ejemplo de una variable que se necesita para procesar cada uno de la pluralidad de ítems gráficos, los componentes rojo, verde y azul y los valores de opacidad son también un ejemplo de una variable que se necesita para procesar cada uno de la pluralidad de ítems gráficos. Como otro ejemplo, un valor de contador puede ser una variable necesaria para procesar cada uno de la pluralidad de ítems gráficos.

**[0130]** En algunos ejemplos, los datos de la variable (por ejemplo, el valor de la variable) pueden ser diferentes para la pluralidad de ítems gráficos. Por ejemplo, uno de los valores de los componentes rojo, verde o azul para al menos uno de los ítems gráficos puede ser diferente de los valores de los componentes rojo, verde o azul para los otros ítems gráficos de la pluralidad de ítems gráficos. Sin embargo, en algunos ejemplos, los datos de la variable pueden ser los mismos para la pluralidad de ítems gráficos. Por ejemplo, los valores de una matriz PRJ de una matriz PRJ pueden ser los mismos para toda la pluralidad de ítems gráficos. En las técnicas descritas en la presente divulgación, si los datos de la variable son los mismos para una pluralidad de ítems gráficos, los datos se pueden almacenar en el SGPR, lo que reduce los tiempos de acceso y promueve ganancias de eficiencia energética.

**[0131]** En algunos ejemplos, la unidad de control 42 del núcleo de sombreador 24 puede determinar si una instrucción en un programa de sombreador, incluida por el compilador 38, indica que los datos de la variable de la pluralidad de ítems gráficos son los mismos para determinar si los datos de la variable de la pluralidad de ítems gráficos son los mismos. Como otro ejemplo, la unidad de control 42 (por medio de la unidad de detección lógica uniforme 44) puede determinar si son las mismas las direcciones de memoria pedidas por los elementos de procesamiento 48 para los datos de la variable, para procesar los ítems gráficos respectivos. En estos ejemplos, la unidad de control 42 puede determinar que los datos de la variable son los mismos para todos los ítems gráficos si todas las direcciones de memoria son las mismas, y la unidad de control 42 puede determinar que los datos de la variable no son los mismos para al menos uno de los ítems gráficos si todas las direcciones de memoria no son las mismas (por ejemplo, hay al menos dos ítems gráficos con datos diferentes).

**[0132]** El núcleo de sombreador 24 (por ejemplo, por medio de la unidad de control 42) puede almacenar los datos en una localización de almacenamiento del SGPR 28 asociada con la variable si los datos de la variable de la pluralidad de ítems gráficos (102) son los mismos. Por ejemplo, en las FIGS. 5A, la primera localización de almacenamiento del SGPR 28, identificada como R48, está asociada con la variable R0 para los ítems gráficos P0-P3. Además, R48 está asociado con las localizaciones de almacenamiento del GPR 26 que almacenan los datos de la variable R0 para los ítems gráficos P0-P3. De esta manera, si los datos se almacenan en el SGPR 28, R48 se refiere a la primera localización de almacenamiento del SGPR 28. Si los datos se almacenan en el GPR 26, R48 se refiere a las localizaciones de almacenamiento del GPR 26 que almacenan datos para la variable R0 para los ítems gráficos P0-P3 respectivos.

**[0133]** En algunos ejemplos, si los datos de la variable de la pluralidad de ítems gráficos son los mismos, la unidad de control 42 puede emitir solo una petición de datos desde una dirección de memoria que sea la misma para la variable para todos los ítems gráficos. La unidad de control 42 puede recuperar los datos solo una vez para almacenarlos en la localización de almacenamiento del SGPR 28 asociada con la variable. En algunos ejemplos, la unidad de control 42 puede afirmar un indicador válido uniforme si se determina que los datos de la variable de la pluralidad de gráficos son los mismos. En algunos de estos ejemplos, la unidad de control 42 puede almacenar los datos en la localización de almacenamiento del SGPR si se afirma el indicador válido uniforme.

**[0134]** El núcleo de sombreador 24 (por ejemplo, por medio de la unidad de control 42) puede almacenar datos para la variable de cada uno de la pluralidad de ítems gráficos en múltiples localizaciones de almacenamiento de GPR 26 si los datos de la variable de al menos dos de la pluralidad de ítems gráficos no son los mismos (104) (por ejemplo, hay al menos dos ítems gráficos cuyos datos son diferentes entre sí). Por ejemplo, como se ilustra en la FIG. 5B, la unidad de control 42 puede almacenar los datos para la variable para cada uno de los elementos de procesamiento 48 en localizaciones de almacenamiento del GPR 26.

5 **[0135]** La unidad de control 42 también puede recibir una petición de los datos almacenados de la variable de cada uno de los elementos de procesamiento 48 para procesar los ítems gráficos respectivos. Si los datos son los mismos, la unidad de control 42 puede emitir solo una petición de los datos del SGPR 28. Si los datos no son los mismos, la unidad de control 42 puede emitir peticiones de todos los elementos de procesamiento 48 para los datos del GPR 26.

10 **[0136]** En uno o más ejemplos, las funciones descritas se pueden implementar en hardware, software, firmware o en cualquier combinación de los mismos. Si se implementan en software, las funciones se pueden almacenar en o transmitir sobre un medio legible por ordenador como una o más instrucciones o código, y ejecutarse por 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 tangibles legibles por ordenador 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.

20 **[0137]** 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 en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que se pueda usar para almacenar 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 disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disco flexible y disco Blu-ray, donde unos discos reproducen normalmente los datos magnéticamente, mientras que otros discos reproducen datos ópticamente con láseres. Las combinaciones de lo anterior se deberían incluir también dentro del alcance de los medios legibles por ordenador.

30 **[0138]** Uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de uso general, circuitos integrados específicos de la aplicación (ASIC), matrices lógicas programables por campo (FPGA) u otros circuitos lógicos integrados o discretos equivalentes pueden ejecutar las instrucciones. En consecuencia, el término "procesador", como se usa en el presente documento, se puede referir a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar dentro de módulos de hardware y/o de software dedicados configurados para codificar y decodificar, o incorporar en un códec combinado. Además, las técnicas se podrían implementar por completo en uno o más circuitos o elementos lógicos.

40 **[0139]** 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 la presente 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 mediante diferentes unidades de hardware. En su lugar, como se describe anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionar mediante un grupo de unidades de hardware interoperativas, que incluya uno o más procesadores como se describe anteriormente, junto con software y/o firmware adecuados.

50 **[0140]** Se han descrito diversos ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

**REIVINDICACIONES**

1. Un procedimiento para almacenar datos, comprendiendo el procedimiento:

5 determinar si los datos de una variable para una pluralidad de ítems gráficos son los mismos, en los que la variable es necesaria para procesar cada uno de la pluralidad de ítems gráficos, en los que la pluralidad de ítems gráficos se procesan con elementos de procesamiento respectivos de un núcleo de sombreador de gráficos de una unidad de procesamiento (GPU) durante la ejecución de un programa de sombreador, y en los que cada uno de los ítems gráficos es una unidad base en la cual se produce el procesamiento;

10 **caracterizado por:**  
 almacenar los datos en una localización de almacenamiento de un registro de uso general compartido (SGPR) asociado con la variable si los datos de la variable de la pluralidad de ítems gráficos son los mismos;  
 y

15 almacenar datos para la variable de cada uno de la pluralidad de ítems gráficos en múltiples localizaciones de almacenamiento de un registro de uso general (GPR), separado del SGPR, si los datos de la variable de al menos dos de la pluralidad de ítems gráficos no son los mismos.

20 **2.** El procedimiento de la reivindicación 1, en el que los datos comprenden datos de atributos de un atributo, y la variable comprende el atributo para la pluralidad de ítems gráficos.

25 **3.** El procedimiento de la reivindicación 1, en el que determinar si los datos de la variable de la pluralidad de ítems gráficos son los mismos comprende determinar si una instrucción en un programa de sombreador, generada por un compilador, indica que los datos de la variable de la pluralidad de ítems gráficos son los mismos.

30 **4.** El procedimiento de la reivindicación 1, en el que determinar si los datos de la variable de la pluralidad de ítems gráficos son los mismos comprende:

determinar si las direcciones de memoria pedidas por los elementos de procesamiento para los datos, para procesar los ítems gráficos respectivos, son las mismas;

35 determinar que los datos de la variable son los mismos para todos los ítems gráficos si todas las direcciones de memoria son las mismas; y

determinar que los datos de la variable no son los mismos para al menos dos de los ítems gráficos si todas las direcciones de memoria no son las mismas.

40 **5.** El procedimiento de la reivindicación 1, que comprende además:

recibir una petición de los datos almacenados de la variable de cada uno de los elementos de procesamiento para procesar los ítems gráficos respectivos; y

45 si los datos son los mismos, emitir solo una petición de los datos del SGPR.

**6.** El procedimiento de la reivindicación 1, que comprende además:

50 recibir una petición de los datos almacenados de la variable de cada uno de los elementos de procesamiento para procesar los ítems gráficos respectivos; y

si los datos no son los mismos, emitir peticiones de todos los elementos de procesamiento para los datos del GPR.

55 **7.** El procedimiento de la reivindicación 1, en el que almacenar los datos en la localización de almacenamiento del SGPR asociado con la variable si los datos de la variable de la pluralidad de ítems gráficos son los mismos comprende:

emitir solo una petición de los datos desde una dirección de memoria que sea la misma para la variable para todos los ítems gráficos; y

60 recuperar los datos solo una vez para almacenarlos en la localización de almacenamiento del SGPR asociado con la variable.

**8.** El procedimiento de la reivindicación 1, que comprende además:

65

afirmar un indicador válido uniforme si se determina que los datos de la variable de la pluralidad de ítems gráficos son los mismos,

5 en el que almacenar los datos en la localización de almacenamiento del SGPR comprende almacenar los datos en la localización de almacenamiento del SGPR si se afirma el indicador válido uniforme.

9. El procedimiento de la reivindicación 1, en el que los ítems gráficos comprenden una de una pluralidad de vértices o de una pluralidad de píxeles.

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

15 una unidad de procesamiento gráfico (GPU), comprendiendo la GPU un núcleo de sombreador, comprendiendo el núcleo de sombreador una unidad de control, una pluralidad de elementos de procesamiento, un registro de uso general compartido (SGPR) y un registro de uso general (GPR), en el que la unidad de control está configurada para:

20 determinar si los datos de una variable para una pluralidad de ítems gráficos son los mismos, en el que la variable es necesaria para procesar cada uno de la pluralidad de ítems gráficos, en el que la pluralidad de ítems gráficos se procesa con elementos de procesamiento respectivos del núcleo de sombreador de la GPU durante la ejecución de un programa de sombreador, y en el que cada uno de los ítems gráficos es una unidad base en la cual se produce el procesamiento;

**caracterizado por:**

25 almacenar los datos en una localización de almacenamiento del SGPR asociado con la variable si los datos de la variable de la pluralidad de ítems gráficos son los mismos; y

30 almacenar datos para la variable de cada uno de la pluralidad de ítems gráficos en múltiples localizaciones de almacenamiento del GPR, separado del SGPR, si los datos de la variable de al menos dos de la pluralidad de ítems gráficos no son los mismos.

11. El dispositivo de la reivindicación 10, que comprende además un procesador, en el que el procesador está configurado para, al menos, uno de:

35 codificar una instrucción que indique que los datos de la variable para la pluralidad de ítems gráficos son los mismos a partir de los cuales la unidad de control determina si los datos son los mismos; o

40 generar instrucciones en el programa de sombreador que solapa las localizaciones de almacenamiento de la SGPR con variables de los ítems gráficos.

12. El dispositivo de la reivindicación 10, en el que, para determinar si los datos de la variable de la pluralidad de ítems gráficos son los mismos, la unidad de control está configurada para:

45 determinar si las direcciones de memoria pedidas por los elementos de procesamiento para los datos de la variable, para procesar los ítems gráficos respectivos, son las mismas;

determinar que los datos de la variable son los mismos para todos los ítems gráficos si todas las direcciones de memoria son las mismas; y

50 determinar que los datos de la variable no son los mismos para al menos dos de los ítems gráficos si todas las direcciones de memoria no son las mismas.

13. El dispositivo de la reivindicación 10, en el que el codificador de vídeo está configurado para:

55 recibir una petición de los datos almacenados de la variable de cada uno de los elementos de procesamiento para procesar los ítems gráficos respectivos; y

si los datos son los mismos, emitir solo una petición de los datos del SGPR, o

60 si los datos no son los mismos, emitir las peticiones de todos los elementos de procesamiento para los datos del GPR.

14. El dispositivo de acuerdo con la reivindicación 10, en el que, para almacenar los datos en la localización de almacenamiento del SGPR asociados con la variable si los datos de la variable de la pluralidad de ítems gráficos son los mismos, la unidad de control está configurada para:

65

emitir solo una petición de datos desde una dirección de memoria que sea la misma para la variable para todos los ítems gráficos; y

5 recuperar los datos solo una vez para almacenarlos en la localización de almacenamiento del SGPR asociado con la variable.

10 **15.** Un medio de almacenamiento de datos legible por ordenador que tiene instrucciones almacenadas en el mismo que, cuando se ejecutan, causan que uno o más procesadores realicen el procedimiento de acuerdo con una cualquiera de las reivindicaciones 1 a 9.

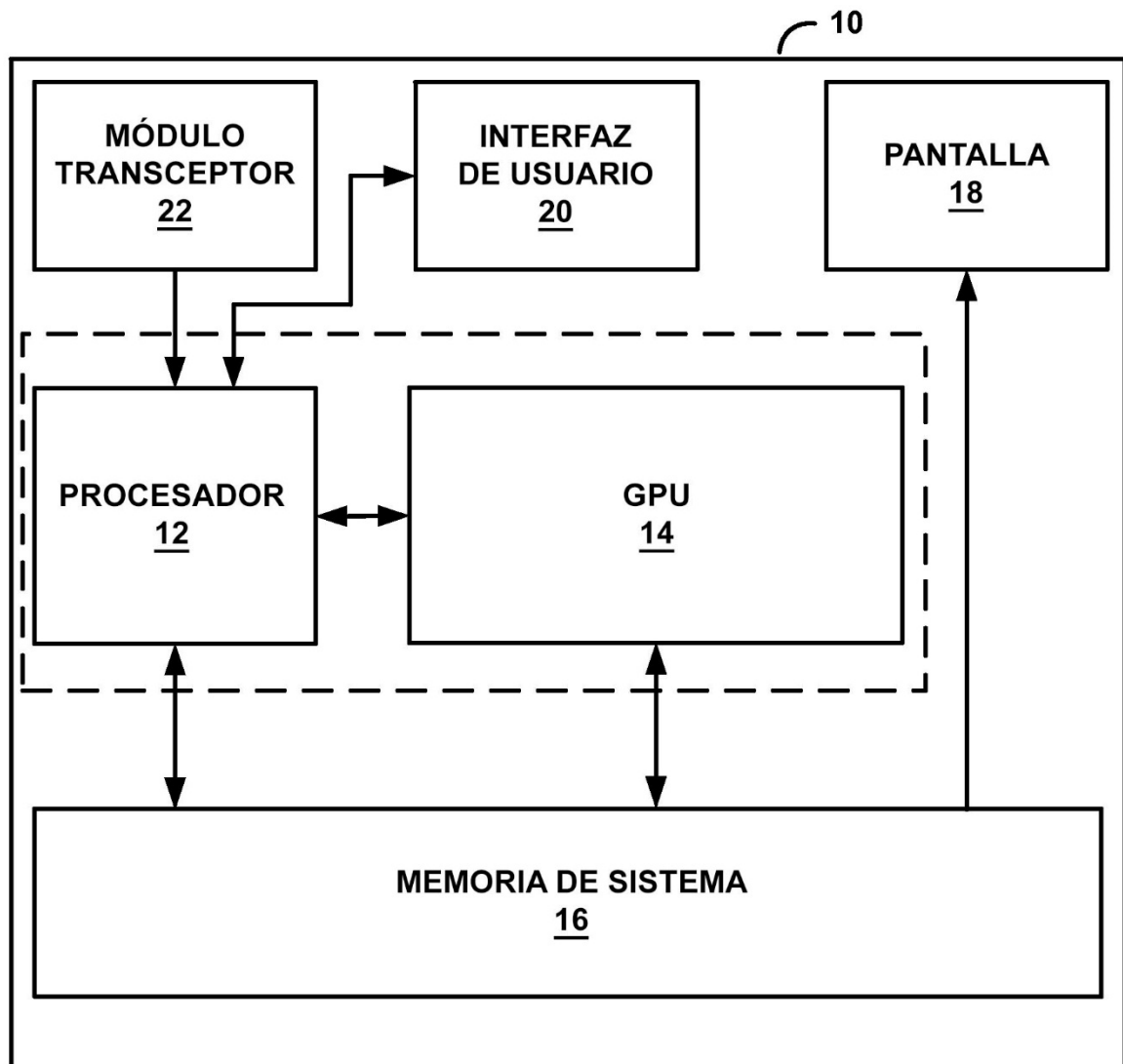
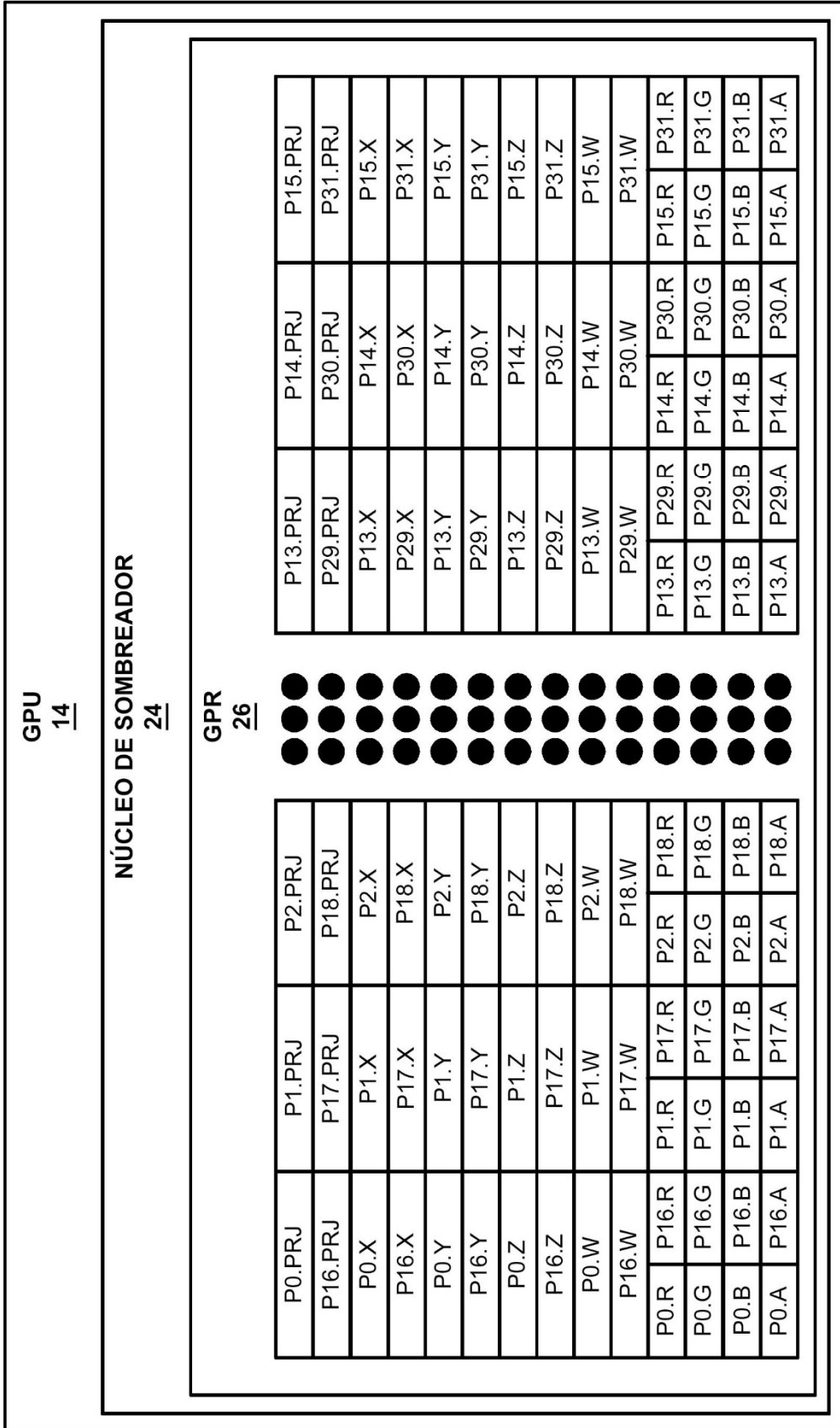


FIG. 1



**FIG. 2**

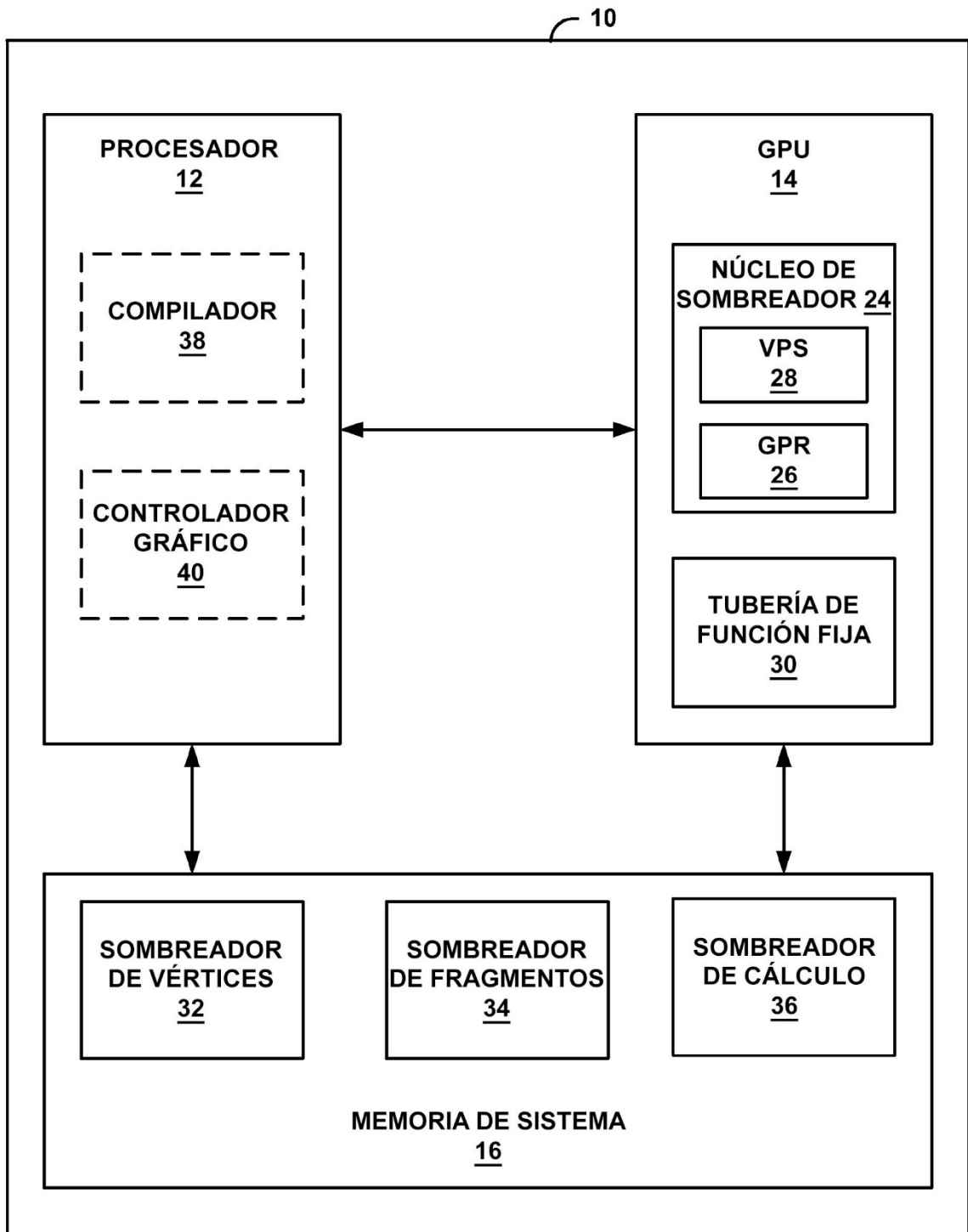


FIG. 3



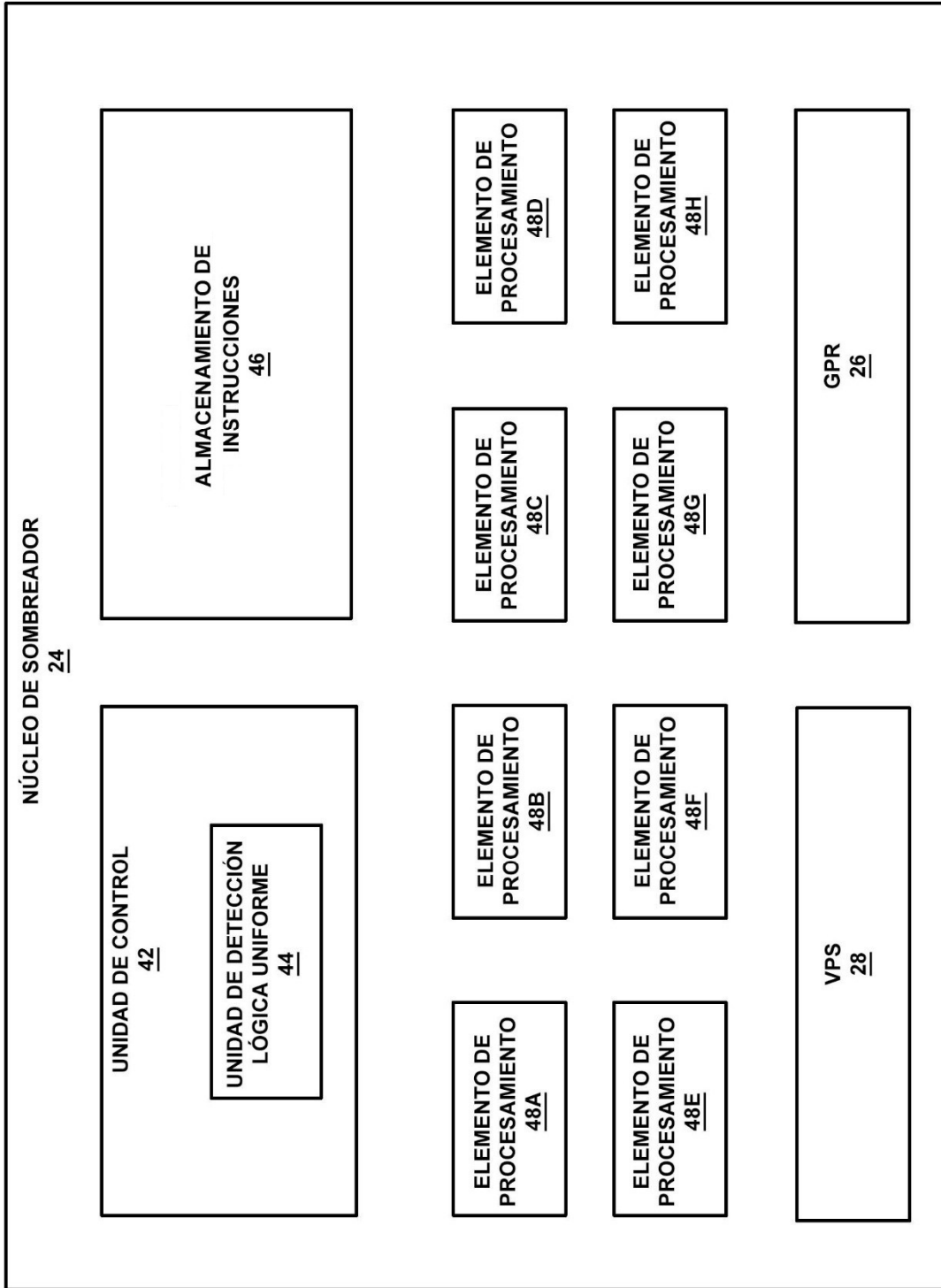


FIG. 4

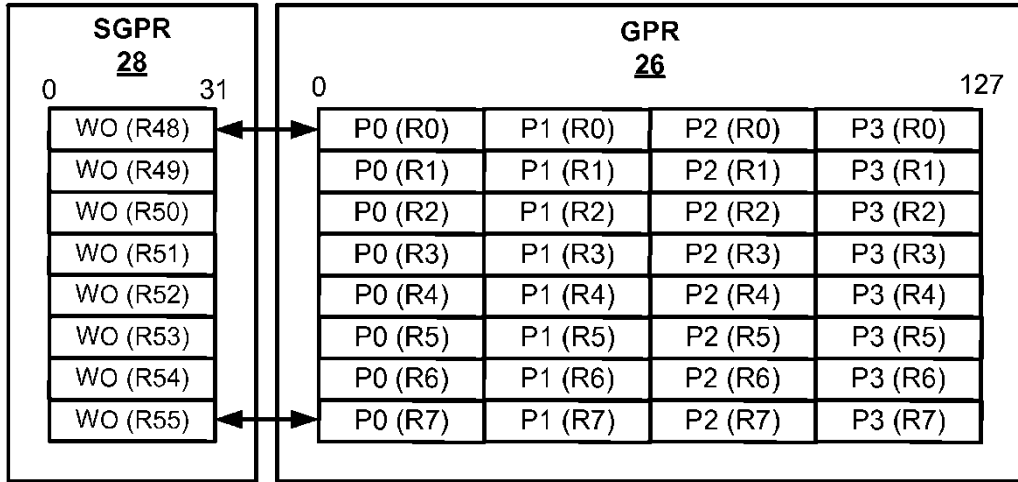


FIG. 5A

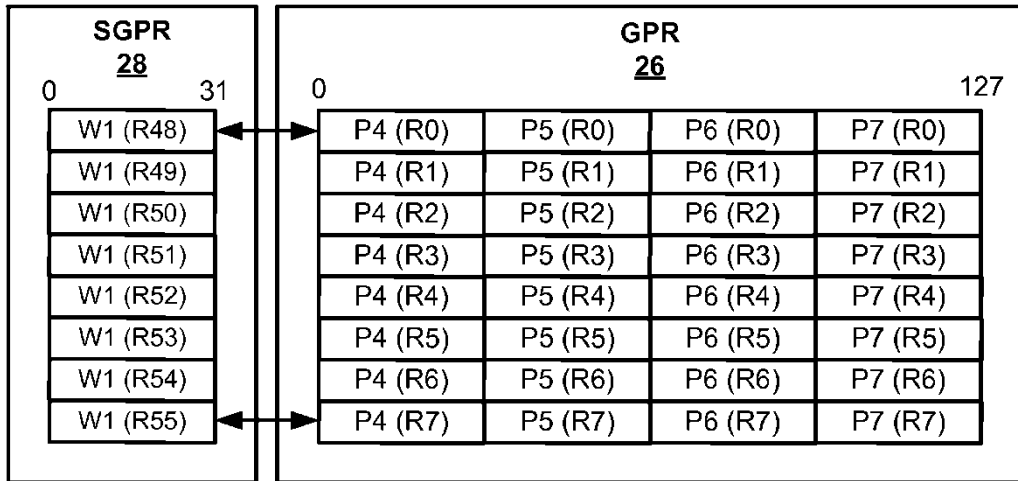
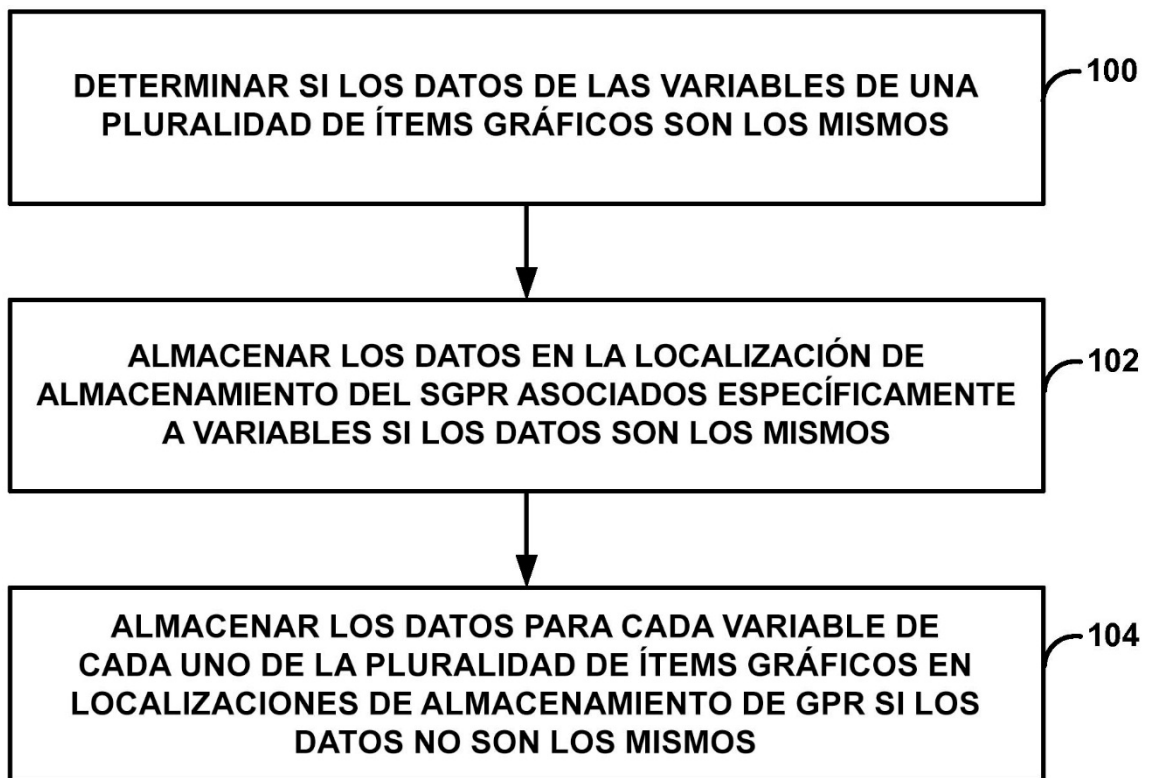


FIG. 5B



**FIG. 6**