(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 732 056**

(51) Int. CI.:

G06T 15/00 (2011.01)
G06T 1/20 (2006.01)
G06F 9/50 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **07.08.2015 PCT/US2015/044279**

(87) Fecha y número de publicación internacional: **25.02.2016 WO16028519**

(96) Fecha de presentación y número de la solicitud europea: **07.08.2015 E 15753276 (3)**

(97) Fecha y número de publicación de la concesión europea: **20.03.2019 EP 3183714**

(54) Título: **Técnicas de ejecución del programa sombreador para uso en procesamiento de gráficos**

(30) Prioridad:

22.08.2014 US 201414466554

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
20.11.2019

(73) Titular/es:

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

(72) Inventor/es:

GOEL, VINEET;
KIM, DONGHYUN y
ZHONG, GANG

(74) Agente/Representante:

FORTEA LAGUNA, Juan José

DESCRIPCIÓN

Técnicas de ejecución del programa sombreador para uso en procesamiento de gráficos

CAMPO TÉCNICO

[0001] Esta divulgación se refiere a sistemas de procesamiento de gráficos, y más particularmente, a la ejecución de programas sombreadores en los sistemas de procesamiento de gráficos.

ANTECEDENTES

[0002] Los dispositivos informáticos a menudo utilizan una unidad de procesamiento de gráficos (GPU) para acelerar la renderización de datos gráficos para su visualización. Dichos dispositivos informáticos pueden incluir, por ejemplo, estaciones de trabajo con ordenador, teléfonos móviles (por ejemplo, los llamados teléfonos inteligentes), sistemas integrados, ordenadores personales, tablets y consolas de videojuegos. Las GPU típicamente implementan una línea de procesamiento de gráficos que incluye una pluralidad de etapas de procesamiento que funcionan juntas para ejecutar comandos de procesamiento de gráficos. Tradicionalmente, las GPU incluían una línea de procesamiento de gráficos de función fija donde cada etapa de procesamiento en la línea se implementaba con hardware de función fija (por ejemplo, hardware que está cableado para realizar un determinado conjunto de funciones especializadas y no es capaz de ejecutar un programa descargable por usuario).

[0003] Más recientemente, las líneas de procesamiento de gráficos han desplazado a una arquitectura programable donde una o más etapas de procesamiento en la línea son etapas de procesamiento programables y se implementan con una o más unidades de sombreador programables. Cada una de las unidades de sombreador programables puede configurarse para ejecutar un programa sombreador. Una aplicación de usuario puede especificar el programa sombreador que se ejecutará en las etapas de procesamiento programables en una línea de gráficos programable, lo cual proporcionará un alto grado de flexibilidad en el uso de las GPU de hoy en día.

El documento US2014204080 A1 divulga "INDEXED STREAMOUT BUFFERS FOR GRAPHICS PROCESSING [MEMORIAS INTERMEDIAS DE FLUJO INDEXADAS PARA EL PROCESAMIENTO DE GRÁFICOS]"; el documento US2009237401 A1 divulga "MULTI-STAGE TESSELLATION FOR GRAPHICS RENDERING [TESELACIÓN MULTI-ETAPA PARA RENDERIZACIÓN DE GRÁFICOS]"; el documento US2013265307 A1 divulga "PATCHED SHADING IN GRAPHICS PROCESSING [SOMBREADO DE PARCHES EN EL PROCESAMIENTO DE GRÁFICOS]"; el documento US2008094408 A1 divulga "System and Method for Geometry Graphics Processing [Sistema y procedimiento para el procesamiento de gráficos de geometría]".

[0004] A medida que se desarrolla la tecnología de procesamiento de gráficos, las líneas de procesamiento de gráficos se están volviendo más sofisticados y se está añadiendo un número creciente de diferentes tipos de etapas de procesamiento programable a las líneas de procesamiento de gráficos estándar que se especifican mediante los principales interfaces de programación de aplicación de gráficos (API). La implementación de estos diferentes tipos de etapas de procesamiento programables con los recursos limitados en una GPU puede presentar desafíos significativos.

SUMARIO

[0005] La invención está definida en las reivindicaciones adjuntas. Esta divulgación describe técnicas para ejecutar programas sombreadores en una unidad de procesamiento de gráficos (GPU). Un programa sombreador puede referirse a un programa que se carga en una GPU y es ejecutado por la GPU con una o más unidades de sombreador que están incluidas en la GPU. Una GPU puede ejecutar varias instancias de un programa sombreador donde cada una de las instancias del programa sombreador ejecuta las mismas instrucciones del programa con respecto a diferentes elementos de datos. Los elementos de datos de ejemplo pueden incluir vértices, primitivas y píxeles. Los programas sombreadores que procesan vértices se configuran típicamente para generar un único vértice de salida para cada uno de los vértices de entrada que recibe el programa sombreador. Sin embargo, las técnicas de esta divulgación pueden, en algunos ejemplos, ejecutar un programa sombreador que realiza el procesamiento de sombreador de vértices y que genera múltiples vértices de salida para cada vértice de entrada que recibe el programa sombreador.

[0006] La ejecución de un programa sombreador que realiza el procesamiento de sombreador de vértices y que genera múltiples vértices de salida para cada uno de los vértices de entrada que recibe el programa sombreador puede reducir el número de subprocesos que se necesitan para procesar un conjunto particular de vértices de entrada en relación con la cantidad de subprocesos que se necesitan cuando se usa un programa sombreador que simplemente genera un único vértice de salida para cada vértice de entrada. Reducir el número de subprocesos que se utilizan para procesar vértices puede reducir los recursos de procesamiento utilizados por una GPU y/o reducir la potencia consumida por una GPU. Además, permitir que un programa sombreador que realice el procesamiento de sombreador de vértices genere múltiples vértices de salida para cada vértice de entrada puede mejorar la flexibilidad de

programación de una GPU. De esta manera, se puede mejorar el rendimiento, el consumo de energía y/o la flexibilidad de programación de una GPU que realiza el procesamiento de vértices programable.

[0007] En un ejemplo, esta divulgación describe un procedimiento que incluye la ejecución, con una unidad de sombreador de un procesador de gráficos, un programa sombreador que realiza el procesamiento de sombreador de vértices y que genera múltiples vértices de salida para cada vértice de entrada que es recibido por el programa sombreador.

[0008] En otro ejemplo, esta divulgación describe un dispositivo que incluye una unidad de procesamiento de gráficos (GPU) que comprende una unidad de sombreador configurado para ejecutar un programa sombreador que realiza el procesamiento de sombreador de vértices y que genera múltiples vértices de salida para cada vértice de entrada que es recibido por el programa sombreador.

[0009] En otro ejemplo, esta divulgación describe un aparato que incluye un procesador de gráficos que comprende una unidad de sombreador. El aparato incluye además medios para ejecutar, con la unidad de sombreador del procesador de gráficos, un programa sombreador que realiza el procesamiento de sombreador de vértices y que genera múltiples vértices de salida para cada vértice de entrada que recibe el programa sombreador.

[0010] En otro ejemplo, esta divulgación describe un medio de almacenamiento legible por ordenador no transitorio que almacena instrucciones que después de la ejecución por uno o más procesadores hacen que los uno o más procesadores ejecuten, con una unidad de sombreador de un procesador de gráficos, un programa sombreador que realiza el procesamiento de sombreador de vértices y eso genera múltiples vértices de salida para cada vértice de entrada que recibe el programa sombreador.

[0011] La información de uno o más ejemplos de la divulgación se exponen en los dibujos adjuntos y la descripción siguiente. Otras características, objetos y ventajas de la divulgación serán evidentes a partir de la descripción y dibujos, y a partir de las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0012]

La FIG. 1 es un diagrama conceptual que ilustra un ejemplo de flujo de gráficos que puede implementarse utilizando las técnicas de ejecución del programa sombreador de esta divulgación.

La FIG. 2 es un diagrama de bloques de una GPU de ejemplo que se puede usar para implementar las técnicas de ejecución del programa sombreador de esta divulgación.

La FIG. 3 es un diagrama de bloques que ilustra una unidad de sombreador de ejemplo que se puede usar en la GPU de la FIG. 2.

La FIG. 4 es un diagrama conceptual que ilustra una tira de triángulo de ejemplo que puede procesarse utilizando las técnicas de ejecución de programa sombreador de ejemplo de esta divulgación.

La FIG. 5 es un diagrama conceptual que ilustra configuraciones de subprocesos de ejemplo que pueden usarse para ejecutar una pluralidad de instancias de un programa sombreador de vértices/geometría fusionado para procesar la tira de triángulo que se muestra en la FIG. 4 de acuerdo con esta divulgación.

La FIG. 6 es un diagrama conceptual que ilustra el flujo de procesamiento de ejemplo asociado con la ejecución de un programa sombreador de vértices/geometría fusionado de acuerdo con esta divulgación.

La FIG. 7 ilustra el pseudo código asociado con la ejecución de un programa sombreador de vértices/geometría fusionado de acuerdo con esta divulgación.

La FIG. 8 es un diagrama de bloques que ilustra un ejemplo de dispositivo informático que puede usarse para implementar las técnicas de ejecución de programa sombreador de esta divulgación.

La FIG. 9 es un diagrama de flujo que ilustra una técnica de ejemplo para ejecutar un programa sombreador de acuerdo con esta divulgación.

La FIG. 10 es un diagrama de flujo que ilustra una técnica de ejemplo para ejecutar un programa sombreador de vértices/geometría fusionado de acuerdo con un modo replicado y un modo no replicado de acuerdo con esta divulgación.

La FIG. 11 es un diagrama de flujo, una técnica de ejemplo para seleccionar uno de un modo replicado y un modo no replicado para ejecutar programas sombreadores de vértices/geometría fusionados de acuerdo con esta divulgación.

DESCRIPCIÓN DETALLADA

[0013] Esta divulgación describe técnicas para la ejecución de programas sombreadores en una unidad de procesamiento de gráficos (GPU). Un programa sombreador puede referirse a un programa que se carga en una GPU y es ejecutado por la GPU con una o más unidades de sombreador que están incluidas en la GPU. Una GPU puede ejecutar varias instancias de un programa sombreador donde cada una de las instancias del programa sombreador ejecuta las mismas instrucciones del programa con respecto a diferentes elementos de datos. Los elementos de datos de ejemplo pueden incluir vértices, primitivas y píxeles. Los programas sombreadores que procesan vértices se configuran típicamente para generar un único vértice de salida para cada uno de los vértices de entrada que recibe el programa sombreador. Sin embargo, las técnicas de esta divulgación pueden, en algunos ejemplos, ejecutar un programa sombreador que realiza el procesamiento de sombreador de vértices y que genera múltiples vértices de salida para cada vértice de entrada que recibe el programa sombreador.

[0014] La ejecución de un programa sombreador que realiza el procesamiento de sombreador de vértices y que genera múltiples vértices de salida para cada uno de los vértices de entrada que recibe el programa sombreador puede reducir el número de subprocesos que se necesitan para procesar un conjunto particular de vértices de entrada en relación con la cantidad de subprocesos que se necesitan cuando se usa un programa sombreador que simplemente genera un único vértice de salida para cada vértice de entrada. Reducir el número de subprocesos que se utilizan para procesar vértices puede reducir los recursos de procesamiento utilizados por una GPU y/o reducir la potencia consumida por una GPU. De esta manera, se puede mejorar el rendimiento y/o el consumo de energía de una GPU que realiza el procesamiento de vértices programable.

[0015] Por otra parte, permitir a un programa sombreador que realice el procesamiento de sombreador de vértices para generar múltiples vértices de salida para cada vértice de entrada puede mejorar la flexibilidad de programación de una GPU. Típicamente, los modelos de programación de sombreador de vértices especifican que un programa sombreador de vértices debe invocarse una vez para cada vértice de entrada y que el programa sombreador de vértices debe generar un vértice de salida individual para cada invocación del programa sombreador de vértices. Las técnicas de esta divulgación se pueden usar para implementar un modelo de programación de sombreador de vértices que permita generar múltiples vértices de salida para cada invocación de un programa sombreador de vértices. De esta manera, se puede mejorar la flexibilidad del procesamiento de vértices programable realizado por una GPU.

[0016] En algunos ejemplos, el programa sombreador que realiza el procesamiento de sombreador de vértices y que genera múltiples vértices de salida para cada uno de los vértices de entrada que son recibidos por el programa sombreador puede ser un programa sombreador de vértices/geometría fusionado. Un programa sombreador de vértices/geometría fusionado puede referirse a un programa sombreador que se puede configurar para realizar el procesamiento de sombreador de vértices con respecto a un procesamiento de sombreador de vértices y geometría con respecto a una primitiva. Los programas sombreadores de geometría se configuran típicamente para recibir una primitiva de entrada y emitir cero o más primitivas de salida en respuesta a la recepción de la primitiva de entrada. Debido a que cada una de las primitivas de salida generadas por un programa sombreador de geometría puede incluir más de un vértice, si un programa sombreador de vértices/geometría fusionado genera solo un vértice de salida individual para cada vértice de entrada, entonces tal vez sea necesario ejecutar múltiples instancias del programa sombreador de vértices/geometría fusionado un para cada una de las primitivas para realizar el procesamiento de sombreador de geometría para las primitivas. La ejecución de un programa sombreador de vértices/geometría fusionado que genera múltiples vértices de salida para cada vértice recibido por el programa sombreador de acuerdo con las técnicas de esta divulgación, sin embargo, puede permitir, en algunos ejemplos, el procesamiento de sombreador de geometría para cada una de las primitivas a realizar con una sola instancia del programa sombreador de vértices/geometría fusionado por primitiva.

[0017] Permitir que el procesamiento de sombreador de geometría se realice con una sola instancia del programa sombreador de vértices/geometría fusionado por primitiva puede reducir el número de instancias del programa sombreador de vértices/geometría fusionado que se necesitan para procesar un conjunto particular de primitivas relativas a técnicas que requieren múltiples instancias de un programa sombreador de vértices/geometría fusionado para ser ejecutadas para cada primitiva. Reducir el número de instancias de un programa sombreador de vértices/geometría fusionado que se utilizan para procesar primitivas puede reducir los recursos de procesamiento utilizados por una GPU, reducir el número de asignaciones de recursos realizadas por una GPU y/o reducir la potencia consumida por una GPU. De esta manera, se puede mejorar el rendimiento y/o el consumo de energía de una GPU que realiza sombreado de vértice programable y sombreado de geometría programable con programas sombreadores de vértices/geometría fusionados.

[0018] En algunos ejemplos, las técnicas para la ejecución de programas sombreadores pueden incluir la ejecución de un programa sombreador de vértices/geometría fusionado utilizando un modo de ejecución no replicado. La ejecución de un programa sombreador de vértices/geometría fusionado usando un modo de ejecución no replicado

puede implicar asignar cada una de una pluralidad de primitivas a una instancia respectiva del programa sombreador de vértices/geometría fusionado para el procesamiento de sombreador de geometría, y hacer que cada una de las instancias del programa sombreador de vértices/geometría fusionado genere M vértices donde M corresponde al número de vértices que generados para cada primitiva mediante un programa sombreador de geometría que corresponde al programa sombreador de vértices/geometría fusionado. Se puede usar un programa sombreador de vértices/geometría fusionado que genera múltiples vértices de salida para cada vértice de entrada recibido por el programa sombreador de acuerdo con su divulgación cuando se ejecuta el programa sombreador de vértices/geometría fusionado de acuerdo con el modo de ejecución no replicado.

[0019] En contraste, un modo replicado de ejecución para la ejecución de un programa sombreador de vértices/geometría fusionado puede implicar la asignación de cada una de las primitivas a procesar a N instancias del programa sombreador de vértices/geometría resultante de la fusión, y hacer que cada una de las instancias del programa sombreador de vértices/geometría genere un solo vértice. En algunos ejemplos, N puede ser igual a un valor de recuento de vértices de salida máximo especificado por un programa sombreador de geometría que corresponde al programa sombreador de vértices/geometría fusionado.

[0020] El modo no replicado para la ejecución de programas sombreadores de vértices/geometría fusionado puede utilizar una instancia del programa sombreador de vértices/geometría fusionado por primitiva para realizar el procesamiento de sombreador de geometría, mientras que el modo replicado para la ejecución de programas sombreadores de vértices/geometría fusionados puede utilizar N instancias del programa sombreador de vértices/geometría fusionado por primitiva para realizar el procesamiento de sombreador de geometría. Por lo tanto, el uso del modo no replicado para ejecutar programas sombreadores de vértices/geometría fusionado puede reducir el número de instancias de un programa sombreador de vértices/geometría fusionado que son necesarios para procesar un conjunto particular de primitivas en relación con el modo replicado. Como ya se analizó anteriormente, reducir el número de instancias del programa sombreador de vértices/geometría fusionado que se utilizan para procesar primitivas puede reducir los recursos de procesamiento utilizados por una GPU, reducir la cantidad de asignaciones de recursos realizadas por una GPU y/o reducir la potencia consumida por una GPU. De esta manera, se puede mejorar el rendimiento y/o el consumo de energía de una GPU que realiza sombreado de vértice programable y sombreado de geometría programable con programas sombreadores de vértices/geometría fusionados.

[0021] En ejemplos adicionales, las técnicas para la ejecución de programas sombreadores en una GPU pueden incluir técnicas para permitir que una unidad de sombreador cambie entre un modo no replicado para la ejecución de programas sombreadores de vértices/geometría fusionados y un modo replicado para la ejecución de programas sombreadores de vértices/geometría fusionados. Permitir que una unidad de sombreador cambie entre un modo no replicado y un modo replicado para ejecutar programas sombreadores de vértices/geometría fusionado puede proporcionar un control adicional y/o flexibilidad a los usuarios del procesador de gráficos para elegir modos de ejecución particulares que se adapten a requisitos de procesamiento particulares, como, por ejemplo, requisitos de rendimiento, requisitos de consumo de energía, etc.

[0022] En ejemplos adicionales, las técnicas para la ejecución de programas sombreadores pueden incluir técnicas para seleccionar entre el modo no replicado y el modo replicado para ejecutar un programa sombreador de vértices/geometría fusionado y hacer que una unidad de sombreador ejecute el programa sombreador de vértices/geometría fusionado de acuerdo con el modo de ejecución del programa sombreador seleccionado. En algunos ejemplos, las técnicas para seleccionar entre el modo no replicado y el modo replicado pueden seleccionar entre el modo no replicado y el modo replicado basándose en la cantidad total de espacio de almacenamiento requerido para almacenar los vértices de salida asociados con una invocación de interfaz de programación de aplicaciones (API) de un programa sombreador de geometría que corresponde al programa sombreador de vértices/geometría fusionado y/o basándose en la cantidad de amplificación de vértice realizada por el programa sombreador de geometría que corresponde al programa sombreador de vértices/geometría fusionado.

[0023] En general, la cantidad de espacio de almacenamiento (por ejemplo, registros de propósito general (GPRS)) para almacenar vértices de salida en una unidad de sombreador puede ser limitada. Debido a que el modo no replicado para ejecutar programas sombreadores de vértices/geometría fusionado permite que se generen múltiples vértices por cada una de las instancias del programa sombreador de vértices/geometría fusionado, la cantidad de espacio de almacenamiento necesario para almacenar vértices de salida para el modo no replicado puede ser mayor que lo que se necesita para el modo replicado. Si la cantidad de espacio de almacenamiento incluido en una unidad de sombreador no es suficiente para almacenar los vértices de salida para un conjunto dado de instancias de un programa sombreador de vértices/geometría fusionado que tienen que ejecutarse en paralelo, entonces es posible que se deban realizar accesos a la memoria externa, lo cual puede reducir significativamente el rendimiento de la unidad de sombreador.

[0024] Como se analizó anteriormente, el modo no replicado para la ejecución de programas sombreadores de vértices/geometría fusionados puede proporcionar un mejor rendimiento y/o un consumo de energía reducido para la ejecución de un conjunto dado de instancias de un programa sombreador de vértices/geometría fusionado en paralelo. Sin embargo, si los requisitos de espacio de almacenamiento de vértices de salida asociados con la ejecución del conjunto de instancias del programa sombreador de vértices/geometría fusionado son mayores que la cantidad de

espacio de almacenamiento de vértices de salida disponible en la unidad de sombreador, entonces las mejoras de rendimiento y/o potencia logradas por no la replicación del procesamiento del sombreador de geometría puede ser superada por la reducción en el rendimiento ocasionada por los accesos a la memoria externa.

[0025] Por lo tanto, el modo no replicado para la ejecución de programas sombreadores de vértices/geometría fusionados puede proporcionar un mejor rendimiento y/o consumo de energía reducido si los requisitos de espacio de almacenamiento para almacenar los vértices de salida son relativamente pequeños (por ejemplo, si los requisitos de espacio de almacenamiento de vértices de salida son menores o iguales a la cantidad de espacio de almacenamiento de vértices de salida contenido en la unidad de sombreador). Por otro lado, el modo replicado para ejecutar programas sombreadores de vértices/geometría fusionados puede proporcionar un mejor rendimiento si los requisitos de espacio de almacenamiento para almacenar vértices de salida son relativamente altos (por ejemplo, si los requisitos de espacio de almacenamiento de vértices de salida son mayores que la cantidad de espacio de almacenamiento de vértices de salida contenido en la unidad de sombreador).

[0026] La selección entre el modo no replicado y el modo replicado basado en una cantidad total de espacio de almacenamiento requerida para almacenar los vértices de salida asociados con una invocación API de un programa sombreador de geometría y/o basado en la cantidad de amplificación de vértice realizada por el sombreador de geometría puede permitir que un sistema de gráficos use el modo no replicado para ejecutar programas sombreadores de vértices/geometría fusionados cuando la cantidad de espacio de almacenamiento necesario para almacenar los vértices de salida es relativamente pequeño, y para usar el modo replicado para ejecutar los programas sombreadores de vértices/geometría fusionado programas cuando la cantidad de espacio necesario para almacenar vértices de salida es relativamente grande. De esta manera, los beneficios de usar el modo no replicado se pueden obtener para programas sombreadores con requisitos de almacenamiento de vértices de salida relativamente pequeños, al tiempo que se evitan los inconvenientes de rendimiento asociados con los accesos a la memoria externa en los casos en que el espacio de almacenamiento de la unidad de sombreador no es suficiente para almacenar los vértices de salida asociados con un programa sombreador.

[0027] En ejemplos adicionales, las técnicas para la ejecución de programas sombreadores puede incluir técnicas para la generación de código compilado para un programa sombreador de vértices/geometría fusionado, donde el código compilado incluye instrucciones que hacen que una unidad de sombreador ejecute selectivamente el programa sombreador de vértices/geometría fusionado de acuerdo con el modo no replicado o el modo replicado basándose en información indicativa de un modo que se utilizará para la ejecución del programa sombreador. Poner instrucciones en el código compilado para el programa sombreador de vértices/geometría fusionado que sean capaces de ejecutar selectivamente cualquiera de los modos puede permitir cambiar el modo de procesamiento de la unidad de sombreador sin tener que volver a cargar un nuevo programa sombreador en la unidad de sombreador. Además, la colocación de instrucciones en el código compilado para el programa sombreador de vértices/geometría fusionado que son capaces de ejecutar selectivamente cualquiera de los modos también puede simplificar la compilación del programa sombreador de vértices/geometría fusionado.

[0028] La FIG. 1 es un diagrama conceptual que ilustra una línea de gráficos de ejemplo 10 que puede implementarse utilizando las técnicas de ejecución de programa sombreador de esta divulgación. En algunos ejemplos, la línea de gráficos 10 puede corresponder a una línea de gráficos Microsoft® DirectX (DX) 10. En otros ejemplos, la línea de gráficos 10 puede corresponder a una línea de gráficos de Microsoft® DX 11 con teselación deshabilitada.

[0029] La línea de gráficos 10 está configurada para renderizar una o más primitivas de gráficos en un destino de renderización. La línea de gráficos 10 incluye un bloque de recursos 12, un ensamblador de entrada 14, un sombreado de vértice 16, un sombreador de geometría 18, un rasterizador 20, un sombreador de píxeles 22 y una fusión de salida 24.

[0030] El bloque de recursos 12 puede corresponder a uno o más recursos de memoria utilizados por las etapas de línea en la línea de gráficos 10, como, por ejemplo, una o más texturas y/o una o más memorias intermedias (por ejemplo, memorias intermedias de vértice, memorias intermedias de trama, etc.). Las etapas de procesamiento representadas en la FIG. 1 con esquinas rectas representan etapas de procesamiento de función fija, y las etapas de procesamiento representadas en la FIG. 1 con esquinas redondeadas representan etapas de procesamiento programables. Por ejemplo, como se muestra en la FIG. 1, el ensamblador de entrada 14, el rasterizador 20 y la fusión de salida 24 son etapas de procesamiento de función fija, y el sombreado de vértice 16, el sombreador de geometría 18 y el sombreador de píxeles 22 son etapas de procesamiento programables.

[0031] Una etapa de procesamiento programable puede referirse a una etapa de procesamiento que está configurada para ejecutar un programa (por ejemplo, un programa sombreador) que se define mediante, compila mediante, y/o carga en una GPU que implementa la línea de gráficos 10 mediante un dispositivo principal que utiliza la GPU. En algunos casos, el programa puede definirse mediante una aplicación de gráficos a nivel de usuario que se ejecuta en un dispositivo principal y se carga en la GPU mediante un controlador de GPU que se ejecuta en el dispositivo principal. Una etapa de procesamiento de función fija puede incluir hardware que no está configurado para recibir y ejecutar programas desde un dispositivo principal. El hardware incluido en una etapa de procesamiento de función fija puede estar cableado para realizar ciertas funciones. Aunque el hardware incluido en una etapa de procesamiento de función

fija puede ser configurable, la configurabilidad del hardware se basa en una o más señales de control en lugar de basarse en un programa (por ejemplo, un programa sombreador).

[0032] Cada una de las etapas programables que se muestran en línea de gráficos 10 pueden estar configurados para ejecutar un programa sombreador de un tipo particular. Por ejemplo, el sombreador de vértices 16 puede configurarse para ejecutar un programa sombreador de vértices, el sombreador de geometría 18 puede configurarse para ejecutar un programa sombreador de geometría, y el sombreador de píxeles 22 puede configurarse para ejecutar un programa sombreador de píxeles.

[0033] Una GPU que implementa el canal de gráficos 10 puede incluir una o más unidades de sombreador que están configuradas para ejecutar los diferentes tipos de programas sombreadores. Cada uno de los diferentes tipos de programas sombreadores puede ejecutarse en una unidad de sombreador común de una GPU que implementa la línea de gráficos 10 y/o en una o más unidades de sombreador dedicadas que están dedicadas a ejecutar programas sombreadores de uno o más tipos particulares.

[0034] En algunos ejemplos, un programa sombreador de vértices y un programa sombreador de geometría se pueden fusionar en un programa sombreador de vértices/geometría fusionado y una unidad de sombreador en una GPU que implementa la línea de gráficos 10 puede ejecutar el programa sombreador de vértices/geometría fusionado como se describe con más detalle más adelante en esta divulgación. En tales ejemplos, la unidad de sombreador puede, en algunos ejemplos, configurarse adicionalmente para ejecutar un programa sombreador de píxeles como un programa separado en momentos en que el programa sombreador de vértices/geometría fusionado no se ejecuta en la unidad de sombreador.

[0035] Ahora se describirá el funcionamiento general de línea de gráficos 10. La línea de gráficos 10 comienza a renderizar un conjunto de primitivas en respuesta a la recepción de un comando de llamada de extracción y datos indicativos de una o más primitivas a renderizar. Los datos indicativos de las primitivas a renderizar pueden incluir, por ejemplo, uno o más memorias intermedias de vértice, una o más memorias intermedias de índice, y/o una o más configuraciones de estado indicativas del tipo de primitiva a representar. Las memorias intermedias de vértice y/o memorias intermedias de índice pueden, en algunos ejemplos, almacenarse en el bloque de recursos 12.

[0036] El ensamblador de entrada 14 puede recuperar uno o más vértices del bloque de recursos 12, formar la geometría (por ejemplo, las primitivas) basándose en los vértices, y emitir los vértices al sombreador de vértices 16 para procesamiento adicional. El ensamblador de entrada 14 también puede generar uno o más valores generados por el sistema para cada uno de los vértices y suministrar los valores generados por el sistema al sombreador de vértices 16 y/o al sombreador de geometría 18. Por ejemplo, el ensamblador de entrada 14 puede generar valores de identificación de vértices que identifican de manera única cada uno de los vértices en una llamada de extracción particular y suministrar los valores de identificación de vértices al sombreador de vértices 16 y/o al sombreador de geometría 18. Como otro ejemplo, el ensamblador de entrada 14 puede generar valores de identificación de primitiva que identifican de manera única cada una de las primitivas en una llamada de extracción particular, y suministrar los valores de identificación de primitiva al sombreador de geometría 18.

[0037] El sombreador de vértices 16 puede generar vértices de salida basándose en los vértices recibidos desde el ensamblador de entrada 14 y basándose en un programa sombreador de vértices. Desde una perspectiva de programación, para generar los vértices de salida, el sombreado de vértice 16 puede ejecutar una instancia respectiva del programa sombreador de vértices para cada uno de los vértices que se reciben desde el ensamblador de entrada 14. En algunos ejemplos, el programa sombreador de vértices puede realizar un procesamiento por vértice en los vértices de entrada para generar los vértices de salida. El procesamiento por vértice puede referirse al procesamiento que se realiza independientemente para cada uno de los vértices que se procesan. El procesamiento por vértice puede incluir, por ejemplo, realizar transformaciones de vértices, realizar operaciones de iluminación, realizar operaciones de niebla, realizar sombreado de vértice, etc.

[0038] El sombreador de geometría 18 puede generar primitivas de salida basadas en primitivas de entrada que son recibidas por el sombreador de geometría 18 y basadas en un programa sombreador de geometría. Las primitivas de entrada que recibe el sombreador de geometría 18 pueden formarse basándose en los vértices de salida generados por el sombreado de vértice 16. Desde una perspectiva de programación, para generar las primitivas de salida, el sombreador de geometría 18 puede ejecutar una instancia respectiva del programa sombreador de geometría para cada una de las primitivas que recibe el sombreador de geometría 18. En algunos ejemplos, el programa sombreador de geometría puede realizar un procesamiento por primitiva en las primitivas de entrada para generar las primitivas de salida. El procesamiento por primitiva puede referirse al procesamiento que se realiza independientemente para cada una de las primitivas que se procesan. El procesamiento por primitiva puede incluir, por ejemplo, agregar o eliminar vértices, agregar o eliminar el número de primitivas que genera el sombreador de geometría 18 para cada primitiva de entrada, etc.

[0039] El rasterizador 20 puede generar píxeles de origen basándose en las primitivas recibidas del sombreador de geometría 18. Por ejemplo, para cada una de las primitivas recibidas desde el sombreador de geometría 18, el rasterizador 20 puede rasterizar la primitiva para generar una pluralidad de píxeles de origen que corresponden a la

primitiva. Rasterizar una primitiva puede implicar, por ejemplo, realizar una conversión de escaneo en la primitiva para determinar qué píxeles corresponden a la primitiva y/o interpolar atributos para los píxeles que corresponden a una primitiva basándose en los atributos de los vértices de la primitiva.

[0040] El sombreador de píxeles 22 puede generar píxeles de la fuente de salida basándose en los píxeles de la fuente de entrada recibidos desde el rasterizador 20 y basándose en un programa sombreador de píxeles. Desde una perspectiva de programación, para generar los píxeles de la fuente de salida, el sombreador de píxeles 22 puede ejecutar una instancia respectiva del programa sombreador de píxeles para cada uno de los píxeles que se reciben desde el rasterizador 20. En algunos ejemplos, el programa sombreador de píxeles puede realizar un procesamiento por píxel en los píxeles de la fuente de entrada para generar los píxeles de la fuente de salida. El procesamiento por píxel puede referirse al procesamiento que se realiza independientemente para cada uno de los píxeles que se procesan. El procesamiento por píxel puede incluir, por ejemplo, realizar sombreado de píxeles, realizar asignación de texturas, etc.

[0041] La fusión de salida 24 pueden generar píxeles de destino basándose en los píxeles de origen recibidos de sombreador de píxeles 22. En algunos ejemplos, la fusión de salida 24 puede combinar cada uno de los píxeles de origen recibidos desde el sombreador de píxeles 22 con un píxel de destino correspondiente almacenado en un objetivo de procesamiento para generar una versión actualizada del píxel de destino correspondiente. Un píxel de destino puede corresponder a un píxel de origen si el píxel de destino tiene la misma ubicación de píxeles en el destino de renderización que la ubicación de píxeles del píxel de origen. Para combinar los píxeles de origen con los píxeles de destino, la fusión de salida 24 puede realizar una o más de una operación de fusión, una operación de composición y una operación de rasterización con respecto a los píxeles de origen y destino que se fusionarán.

[0042] Los píxeles de destino resultantes se almacenan en un objetivo de renderización, que en algunos ejemplos, puede ser una memoria intermedia de trama. El objetivo de renderización puede formar parte del bloque de recursos 12. Los datos almacenados en el objetivo de renderización pueden corresponder a una versión rasterizada y compuesta de las primitivas recibidas por la línea de gráficos 10.

[0043] Como se analizó anteriormente, a partir de una perspectiva de programación (por ejemplo, desde la perspectiva de la API), los programas sombreadores de vértices son típicamente invocados por una línea de gráficos una vez para cada vértice de entrada y están configurados para generar un vértice de salida para cada invocación. Los programas sombreadores de píxeles se invocan típicamente una vez para cada píxel entrante y se configuran para generar un píxel de salida para cada invocación. Los programas sombreadores de geometría típicamente se invocan una vez para cada primitiva entrante (por ejemplo, punto, línea, triángulo) y se configuran para generar cero, una, dos o más primitivas de salida para cada invocación.

[0044] Las etapas de sombreado programables de línea de gráficos 10 se implementan típicamente en una GPU con una o más unidades de sombreador. Cada una de las unidades de sombreador puede incluir una pluralidad de elementos de procesamiento (por ejemplo, unidades lógicas aritméticas (ALU)) que ejecutan una pluralidad de subprocesos para un programa sombreador particular en paralelo. En algunos casos, las unidades de sombreador pueden ser unidades de sombreador de instrucción única, datos múltiples (SIMD) donde cada uno de los elementos de procesamiento en la unidad de sombreador ejecuta la misma instrucción de un programa sombreador al mismo tiempo con respecto a datos diferentes.

[0045] A menudo, el mismo conjunto de unidades de sombreador puede implementar múltiples tipos diferentes de etapas de sombreado que se incluyen en la línea de gráficos 10. Antes del desarrollo de los sombreadores de geometría, las únicas etapas de procesamiento programables en las líneas de procesamiento de gráficos eran típicamente sombreadores de vértices y sombreadores de píxeles. Los sombreadores de vértices y los sombreadores de píxeles funcionan bajo una interfaz de programación de entrada única/salida única donde se recibe un vértice o píxel de entrada única para cada invocación de sombreado y se genera un vértice o píxel de salida única para cada invocación de sombreador. La interfaz de programación de entrada única/salida única para los sombreadores de vértice y píxel permite que ambos tipos de sombreadores se ejecuten en una unidad de sombreador de hardware común con una interfaz de hardware de entrada única/salida única.

[0046] Una desventaja de la interfaz de hardware de entrada única/salida única para unidades de sombreador, sin embargo, es que tal interfaz no permite que un programa sombreador de vértices genere múltiples vértices por invocación. Esto limita la flexibilidad de los modelos de programación de sombreador de vértices que se pueden implementar para una API de renderización gráfica.

[0047] Otro inconveniente de la interfaz de hardware de entrada/salida única para unidades de sombreador es que los sombreadores de geometría no se ajustan a dicha interfaz. Más específicamente, los sombreadores de geometría están configurados para generar cualquier número de primitivas (dentro de los límites especificados) para cada primitiva de entrada recibida por el sombreador de geometría, y cada una de las primitivas puede incluir cualquier número de vértices. Por lo tanto, los sombreadores de geometría no se ajustan a la interfaz de programación de entrada única/salida única. Esto dificulta la ejecución de sombreadores de geometría con sombreadores de vértice y/o

sombreadores de píxeles en una unidad de sombreador de hardware común que implementa una interfaz de hardware de entrada única/salida única.

[0048] Una solución para hacer frente a la dificultad de interfaces de programación de sombreador de geometría que no se ajusten a una interfaz de hardware de entrada única/salida única es fusionar los programas sombreadores de vértices y geometría en un único programa sombreador de vértices/geometría fusionado y ejecutar el programa sombreador de vértices/geometría fusionado como parte de un subproceso de sombreado común. El programa sombreador de vértices/geometría fusionado puede incluir funciones de sombreador de vértices que están especificadas por un programa sombreador de vértices seguidas de funciones de sombreador de geometría que están especificadas por un programa sombreador de geometría. El programa sombreador de vértices/geometría fusionado puede incluir además un código de parche que se inserta entre las funciones de sombreador de vértices y las funciones de sombreador de geometría para administrar adecuadamente los elementos de datos de salida generados por las funciones de sombreador de vértices y los elementos de datos de entrada recibidos por las funciones de sombreador de geometría.

[0049] Para permitir que el programa sombreador de vértices/geometría de fusionado implemente una interfaz de entrada única/salida única en el ejemplo mencionado anteriormente, múltiples instancias del programa sombreador de vértices/geometría fusionado pueden ser instanciadas para cada una de las primitivas para las cuales deba realizarse la sombreador de geometría el procesamiento, y cada una de las instancias del programa sombreador de vértices/geometría fusionado puede configurarse para recibir un solo vértice y generar un solo vértice. El conjunto de vértices de salida generados por las múltiples instancias del sombreador de vértices/geometría fusionado pueden corresponder colectivamente a los vértices de una primitiva de salida generada por una instancia de API de una etapa de sombreador de geometría que se implementa mediante el programa sombreador de vértices/geometría fusionado.

[0050] Por ejemplo, para cada una de las primitivas de entrada que son procesadas por una línea de procesamiento de gráficos, el programa sombreador de vértices/geometría fusionado puede ser instanciado N veces, donde N es igual al número máximo de vértices de salida por primitiva especificado por una geometría programa sombreador de geometría que corresponde al programa sombreador de vértices/geometría fusionado. Cada uno de los vértices de salida especificados por el programa sombreador de geometría puede ser emitido por cada una de las diferentes instancias, de manera que, colectivamente, las N instancias del programa sombreador de vértices/geometría fusionado generan todos los vértices de salida definidos por un programa sombreador de geometría. que corresponde al programa sombreador de vértices/geometría fusionado. Debido a que cada una de las instancias del programa sombreador de vértices/geometría fusionado en este ejemplo recibe un solo vértice de entrada y genera un solo vértice de salida, el sombreador de vértices/geometría fusionado puede ejecutarse en una unidad de sombreador que implementa una interfaz de hardware de entrada única/salida única.

[0051] Las API de gráficos (como, por ejemplo, DX 10 y DX11) definen la etapa del sombreador de geometría como una etapa que se ejecuta una vez para cada primitiva entrante. Sin embargo, para las técnicas de ejecución del programa sombreador de vértices/geometría fusionado descritas anteriormente, se pueden ejecutar N instancias diferentes de un programa sombreador de vértices/geometría fusionado para cada primitiva entrante. En otras palabras, el procesamiento del sombreador de geometría se replica efectivamente N veces para cada invocación API del sombreador de geometría. Por lo tanto, la técnica descrita anteriormente para ejecutar sombreadores de vértices/geometría fusionados puede denominarse un modo replicado para ejecutar sombreadores de vértices/geometría fusionados.

[0052] Un inconveniente del modo replicado para la ejecución de sombreadores de vértices/geometría fusionados es que los cálculos realizados por el procesamiento de sombreador de geometría se pueden repetir para cada instancia del programa sombreador de vértices/geometría fusionado. Por ejemplo, en algunos casos, el procesamiento de sombreador de geometría para un programa sombreador de vértices/geometría fusionado puede contener un bucle de control de programa que hace que los vértices de salida se calculen en un orden particular, y el procesamiento de sombreador de geometría para el programa sombreador de vértices/geometría fusionado puede, en algunos ejemplos, funcionar usando un mecanismo de cascada. Cuando se usa el mecanismo de cascada, para una instancia dada del programa sombreador de vértices/geometría fusionado, el bucle de control de programa para las funciones de sombreador de geometría se puede realizar para cada uno de los vértices de salida hasta que se calcule el vértice particular que tiene que ser emitido por la instancia particular. Una vez calculado el vértice que debe emitir la instancia particular, el flujo de control para el procesamiento del sombreador de geometría puede dejar de ejecutar el bucle de control.

[0053] En otras palabras, cuando se utiliza un mecanismo de cascada, una primera instancia del programa sombreador de vértices/geometría fusionado puede ejecutar un bucle de control de programa para las funciones de sombreador de geometría una vez y generar un vértice que corresponde a la única iteración del bucle de control, una segunda instancia del programa sombreador de vértices/geometría fusionado puede ejecutar el bucle de control del programa para las funciones de sombreador de geometría dos veces y generar un vértice que corresponda a la segunda iteración del bucle de control, una tercera instancia del programa sombreador de vértices/geometría fusionado puede ejecutar el bucle de control de programa para las funciones de sombreador de geometría tres veces y generar un vértice que corresponda a la tercera iteración del bucle de control, etc.

[0054] El uso de un mecanismo de cascada puede reducir algunos de los cálculos repetidos que ocurren cuando se instancian múltiples instancias del sombreador de vértices/geometría fusionado para cada primitiva, pero tal mecanismo todavía da como resultado cálculos repetidos. Por ejemplo, el primer vértice a ser calculado por las funciones de sombreador de geometría se calculará N veces, el segundo vértice se calculará N-1 veces, etc. Tales cálculos repetidos de vértices pueden reducir la eficiencia de la unidad de sombreador y/o aumentar el consumo de energía de la unidad de sombreador.

[0055] Otro inconveniente del modo replicado para la ejecución de sombreadores de vértices/geometría fusionados es que los recursos (por ejemplo, registros de propósitos generales (GPRS)) deben ser asignados para cada invocación del sombreador de geometría que da como resultado la asignación de recursos repetidos. Las asignaciones de recursos repetidos también pueden reducir la eficiencia de la unidad de sombreador y/o aumentar el consumo de energía de la unidad de sombreador.

[0056] Una desventaja adicional del modo replicado para la ejecución de sombreadores de vértices/geometría fusionados es que un modo tal no es capaz de reutilizar vértices que son compartidos por múltiples primitivas con el fin de reducir los requisitos de procesamiento para sombreadores de vértices. No permitir la reutilización de vértices puede aumentar el consumo de energía y/o aumentar el uso del ancho de banda de la memoria.

[0057] De acuerdo con algunos aspectos de la presente divulgación, se describen técnicas para la ejecución de un programa sombreador que realiza el procesamiento de sombreador de vértices y que genera múltiples vértices de salida para cada vértice de entrada que es recibido por el programa sombreador. La pluralidad de vértices de salida se puede generar basándose en las instrucciones del programa contenidas en el programa sombreador. La ejecución de un programa sombreador que realiza el procesamiento de sombreador de vértices y que genera múltiples vértices de salida para cada uno de los vértices de entrada que recibe un programa sombreador puede, por ejemplo, permitir que el procesamiento de sombreador de geometría se realice con un programa sombreador de vértices/geometría fusionado sin necesidad de replicación del procesamiento del sombreador de geometría en varios subprocesos de ejecución para cada invocación API del sombreador de geometría.

[0058] Por ejemplo, una única instancia de un programa sombreador de vértices/geometría fusionado puede ser invocada para realizar el procesamiento de sombreador de geometría para cada invocación API del sombreador de geometría en lugar de N instancias de sombreador de vértices/geometría como se ha analizado anteriormente con respecto al modo replicado. La instancia única del programa sombreador de vértices/geometría fusionado puede emitir N vértices donde N es el número de vértices de salida definidos por un programa sombreador de geometría que corresponde al programa sombreador de vértices/geometría fusionado. Los N vértices de salida generados por una sola instancia del programa sombreador de vértices/geometría fusionado pueden corresponder colectivamente a los vértices de una primitiva de salida generada por una instancia API de una etapa de sombreador de geometría que se implementa mediante el programa sombreador de vértices/geometría fusionado. De esta manera, un sombreador de vértices/geometría fusionado puede ser ejecutado por una unidad de sombreador sin requerir la replicación del procesamiento de sombreador de geometría en múltiples subprocesos de ejecución. Debido a que se puede invocar una única instancia de sombreador de vértices/geometría para realizar el procesamiento de sombreador de geometría para cada invocación API del sombreador de geometría, este modo de ejecutar un sombreado de vértices/geometría fusionado puede denominarse un modo no replicado para ejecutar un sombreador de vértices/geometría.

[0059] Permitir que un sombreador de vértices/geometría fusionado sea ejecutado por una unidad de sombreador sin requerir la replicación del procesamiento sombreador de geometría puede reducir el número de cálculos ALU necesarios por invocación API del sombreador de geometría, reducir el número de recursos consumidos por una GPU por invocación API del sombreador de geometría, y/o reducir el número de asignaciones de recursos que deben ocurrir para cada invocación API del sombreador de geometría. De esta manera, el rendimiento de la GPU puede mejorarse y/o el consumo de energía de la GPU puede reducirse.

[0060] Además, permitir que un sombreador de vértices/geometría fusionado sea ejecutado por una unidad de sombreador sin requerir la replicación de las funciones de sombreador de geometría puede permitir que se produzca la reutilización de vértice entre las primitivas que comparten uno o más vértices. Permitir que los vértices compartidos se reutilicen puede ahorrar ancho de banda de memoria y/o reducir el consumo de energía.

[0061] Por otra parte, la ejecución de un programa sombreador que genera múltiples vértices de salida para cada uno de los vértices de entrada que son recibidas por el programa sombreador puede aumentar la flexibilidad de los modelos de programación de sombreador de vértices que se utilizan en las API de gráficos. Típicamente, los modelos de programación de sombreador de vértices especifican que un programa sombreador de vértices debe invocarse una vez para cada vértice de entrada y que el programa sombreador de vértices debe generar un vértice de salida individual para cada invocación del programa sombreador de vértices. Sin embargo, la ejecución de un programa sombreador que realiza el procesamiento de sombreador de vértices y que genera múltiples vértices de salida para cada vértice de entrada se puede usar para implementar un modelo de programación de sombreador de vértices que permite generar múltiples vértices de salida para cada invocación de un programa sombreador de vértices. De esta manera, se puede mejorar la flexibilidad del procesamiento de vértices programable realizado por una GPU.

[0062] La FIG. 2 es un diagrama de bloques de un ejemplo de unidad de procesamiento de gráficos (GPU) 30 que puede usarse para implementar las técnicas de ejecución del programa sombreador de esta divulgación. La GPU 30 está configurada para realizar el procesamiento de gráficos basándose en los comandos de gráficos, la información de estado y los programas sombreadores recibidos desde un dispositivo principal (por ejemplo, una unidad central de procesamiento (CPU)). La GPU 30 puede ser un ejemplo de, y de forma alternativa denominarse procesador de gráficos. La GPU 30 incluye un motor de comandos 32 y unidades de procesamiento 34.

[0063] En algunos ejemplos, una o más unidades de procesamiento 34 puede ser configurables para formar una línea de renderización de gráficos tridimensional (3D). Por ejemplo, una o más de las unidades de procesamiento 34 pueden ser configurables para implementar la línea de gráficos 10 ilustrada en la FIG. 1.

[0064] El motor de comandos 32 y las unidades de procesamiento 34 pueden incluir cualquier combinación de unidades de hardware dedicadas, firmware, software, unidades de sombreador, procesadores y elementos de procesamiento que están configurados para realizar las funciones atribuidas a tales componentes. En algunos ejemplos, la GPU 30 puede configurarse para ejecutar instrucciones que hacen que uno o más procesadores de la GPU 30 realicen la totalidad o parte de cualquiera de las técnicas descritas en esta divulgación.

[0065] El motor de comandos 32 puede recibir comandos y datos gráficos de un dispositivo principal, administrar el estado de la GPU 30 basándose en los comandos y controlar el funcionamiento de las unidades de procesamiento 34 para renderizar los datos gráficos basándose en los comandos. Por ejemplo, en respuesta a la recepción de un comando de extracción, el motor de comandos 32 puede controlar el funcionamiento de las unidades de procesamiento 34, de manera que las unidades de procesamiento 34 implementen una línea de renderización de gráficos (por ejemplo, la línea de gráficos 10 en la FIG. 1), y hagan que los datos gráficos se rendericen en un destino de procesamiento a través de la línea de renderización de gráficos. Como otro ejemplo, el motor de comandos 32 puede recibir uno o más programas sombreadores desde un dispositivo principal, recibir uno o más comandos que le indiquen a la GPU 30 que cargue los programas sombreadores en la GPU 30, y hacer que los programas sombreadores se carguen y/o almacenen en las memorias caché de instrucciones asociadas con una o más unidades de sombreador en GPU 30.

[0066] Cada una de las unidades de procesamiento 34 puede ser una unidad de procesamiento programable o una unidad de procesamiento de función fija. Una unidad de procesamiento programable puede referirse a una unidad de procesamiento que está configurada para ejecutar un programa (por ejemplo, un programa sombreador) que se define, compila y/o carga en la GPU 30 mediante un dispositivo principal que usa la GPU 30. Una unidad de procesamiento de función fija puede incluir hardware que no está configurado para recibir y ejecutar programas desde un dispositivo principal. Aunque el hardware incluido en una etapa de procesamiento de función fija puede ser configurable, la configurabilidad del hardware se basa en una o más señales de control en lugar de basarse en un programa (por ejemplo, un programa sombreador).

[0067] Como se muestra en la FIG. 2, las unidades de procesamiento 34 incluyen el programador de subprocesos 36 y las unidades de sombreador 38. En algunos ejemplos, el programador de subprocesos 36 puede ser una unidad de procesamiento de función fija. Aunque no se muestra específicamente en la FIG. 2, las unidades de procesamiento 34 pueden incluir otras unidades de procesamiento de función fija y/o programables que se pueden usar para implementar todo o parte de una línea de renderización de gráficos. Por ejemplo, las unidades de procesamiento 34 pueden incluir unidades de procesamiento de función fija que están configuradas para implementar el ensamblador de entrada 14, el rasterizador 20 y la fusión de salida 24 en la línea de gráficos 10 de la FIG. 1.

[0068] Cada una de unidades de sombreador 38 puede ser una unidad de procesamiento programable. En algunos ejemplos, las unidades de sombreador 38 pueden implementar una o más de las etapas de sombreado de una línea de renderización de gráficos. Por ejemplo, las unidades de sombreador 38 pueden implementar el sombreador de vértices 16, el sombreador de geometría 18 y/o el sombreador de píxeles 22 de la línea de gráficos 10 mostrada en la FIG. 1. En algunos ejemplos, todas o un subconjunto de unidades de sombreador 38 pueden ser unidades de sombreador dedicadas que están configuradas para ejecutar solo tipos específicos de programas sombreadores que se implementan mediante una línea de renderización gráfica. En otros ejemplos, todas o un subconjunto de unidades de sombreador 38 pueden ser unidades de sombreador generales (por ejemplo, unidades de sombreador unificadas) que están configuradas para ejecutar todos o un subconjunto de los tipos de programas sombreadores que se implementan mediante una línea de renderización de gráficos.

[0069] El programador de subprocesos 36 está configurado para controlar cómo se ejecutan los subprocesos del programa sombreador en las unidades de sombreador 38. Cada una de las unidades de sombreador 38 puede configurarse para ejecutar un programa sombreador que se carga en la GPU 30 desde un dispositivo principal. Un programa sombreador puede referirse a un programa que es ejecutado por una etapa de procesamiento programable de GPU 30. En algunos casos, un programa sombreador se puede definir mediante una aplicación de gráficos a nivel de usuario que se ejecuta en un dispositivo principal y que el dispositivo principal lo carga en la GPU 30. En casos adicionales, un programa sombreador puede ser una versión compilada de un programa sombreador de código fuente

que está escrito en un lenguaje de sombreado de alto nivel, como, por ejemplo, un lenguaje de sombreado de alto nivel (HLSL), un lenguaje de sombreado OpenGL (GLSL), un lenguaje de sombreado de C para gráficos (Cg), etc.

[0070] El programador de subprocesos 36 puede recibir información indicativa de uno o más elementos de datos para ser procesados por uno o más programas sombreadores, determinar una configuración de subprocesos para procesar los elementos de datos y hacer que las unidades de sombreador 38 ejecuten uno o más subprocesos para procesar los elementos de datos basados en la configuración del subproceso. Los elementos de datos pueden incluir, por ejemplo, uno o más vértices para ser procesados por un programa sombreador (por ejemplo, un programa sombreador de vértices), una o más primitivas para ser procesadas por un programa sombreador (por ejemplo, un programa sombreador de geometría), y/o uno o más vértices y una o más primitivas para ser procesados por un programa sombreador (por ejemplo, un programa sombreador de vértices/geometría fusionado).

[0071] Un subproceso puede referirse a una instancia de un programa sombreador que es ejecutado por una de las unidades de sombreador 38 (por ejemplo, ejecutada por uno de una pluralidad de elementos de procesamiento incluidos en una unidad de sombreador). Cada una de las instancias de un programa sombreador en particular puede ejecutar las mismas instrucciones con respecto a elementos de datos potencialmente diferentes. Por ejemplo, cada una de las instancias de un programa sombreador de vértices puede ejecutar las mismas instrucciones con respecto a una respectiva de una pluralidad de vértices. Como otro ejemplo, cada una de las instancias de un programa sombreador de geometría puede ejecutar las mismas instrucciones con respecto a una respectiva de una pluralidad de primitivas. Como un ejemplo adicional, cada una de las instancias de un programa sombreador de vértices/geometría fusionado puede ejecutar las mismas instrucciones con respecto a uno o ambos de uno respectivo de una pluralidad de vértices y uno respectivo de una pluralidad de primitivas. En algunos casos, los vértices y/o primitivas individuales ejecutados por subconjuntos de las instancias del programa sombreador de vértices/geometría fusionado pueden ser los mismos.

[0072] Una configuración de subprocesos puede incluir información que asigna los elementos de datos para ser procesados por un programa sombreador particular, a los subprocesos respectivos (por ejemplo, instancias) del programa sombreador que han de ser ejecutados por una o más unidades de sombreador 38. Por ejemplo, para un programa sombreador de vértices, la configuración del subproceso puede incluir información que asigna uno respectivo de una pluralidad de vértices a cada uno de los subprocesos que deben ser ejecutados por una o más unidades de sombreador 38. Como otro ejemplo, para un programa sombreador de geometría, la configuración de subprocesos puede incluir información que asigna uno respectivo de una pluralidad de primitivas a cada uno de los subprocesos que deben ser ejecutados por una o más unidades de sombreador 38. Como un ejemplo adicional, para un programa sombreador de vértices/geometría fusionado, la configuración del subproceso puede incluir información que asigna uno respectivo de una pluralidad de vértices y/o uno respectivo de una pluralidad de primitivas a cada uno de los subprocesos que van a ser ejecutado por una o más de las unidades de sombreador 38. Un programa sombreador de vértices/geometría fusionado puede, por ejemplo, asignarse tanto a un vértice como a una primitiva, a solo un vértice, o solo a una primitiva.

[0073] Para hacer que una de las unidades de sombreador 38 para ejecutar los subprocesos basándose en la configuración de subprocesos, programador de subprocesos 36 puede proporcionar información indicativa de una configuración de subprocesos a la unidad de sombreador. La información indicativa de la configuración del subproceso puede incluir información indicativa de las asignaciones de elementos de datos a ser procesados por un programa sombreador en particular a los subprocesos respectivos del programa sombreador que deben ser ejecutados por la unidad de sombreador. Los elementos de datos pueden incluir vértices, primitivas y/o píxeles.

[0074] Cada una de las unidades de sombreador 38 puede estar configurada para recibir información indicativa de una configuración de subprocesos para el procesamiento de uno o más elementos de datos, y para ejecutar una pluralidad de instancias de un programa sombreador con respecto a los elementos de datos basándose en la información indicativa de La configuración del subproceso recibida desde el programador de subprocesos 36. En algunos ejemplos, cada una de las unidades de sombreador 38 puede configurarse adicionalmente para asignar los registros en la unidad de sombreador respectiva basándose en el programa sombreador que debe ejecutar la unidad de sombreador respectiva y/o basándose en la configuración del subproceso.

[0075] Una onda puede referirse a un grupo de subprocesos que se sometió a una de las unidades de sombreador 38 a ser ejecutadas en paralelo por la unidad de sombreador. En otras palabras, los subprocesos incluidos en una onda pueden ejecutarse simultáneamente en una unidad de sombreador. Cada uno de los subprocesos en una onda puede corresponder a uno respectivo de una pluralidad de instancias del mismo programa sombreador. En algunos ejemplos, cada uno de los subprocesos en una onda puede denominarse fibra. El tamaño de una onda puede referirse al número de fibras que se incluyen en una onda.

[0076] En ejemplos en los que una de las unidades de sombreador 38 incluye una pluralidad de una sola instrucción, elementos de procesamiento-datos múltiples (SIMD), la unidad de sombreador puede ejecutar la onda de tal manera que cada una de las fibras en la onda se ejecuta en una respectiva pluralidad de elementos procesadores SIMD. En tales ejemplos, el número de elementos de procesamiento SIMD en la unidad de sombreador puede ser mayor o igual al tamaño de las ondas (es decir, el número de fibras en las ondas) que ejecuta la unidad de sombreador.

[0077] En algunos ejemplos, para determinar la configuración del subproceso para el procesamiento de un conjunto particular de elementos de datos con un programa sombreador, el programador de subprocesos 36 puede determinar una configuración de onda para procesar el conjunto de elementos de datos. Una configuración de onda puede incluir información que asigna elementos de datos para ser procesados por un programa sombreador en particular a los subprocesos respectivos que se incluyen en una onda de subprocesos que se ejecutarán en paralelo en una de las unidades de sombreador 38. En tales ejemplos, los datos indicativos de la configuración de subprocesos proporcionados por el programador de subprocesos 36 a la unidad de sombreador pueden incluir datos indicativos de una configuración de onda para procesar el conjunto de elementos de datos.

[0078] En algunos ejemplos, las unidades de planificador de subprocesos 36 y/o sombreador 38 pueden configurarse para realizar cualquiera o todas las técnicas descritas en esta divulgación. Por ejemplo, el programador de subprocesos 36 puede configurarse para determinar las configuraciones de subprocesos y/o las configuraciones de onda de acuerdo con las técnicas de esta divulgación. Como otro ejemplo, cada una de las unidades de sombreador 38 puede configurarse para ejecutar uno o más subprocesos y/u ondas que se envían a la unidad de sombreador respectiva de acuerdo con las técnicas de esta divulgación. Más adelante, en esta divulgación, se describirán más detalles sobre cómo funcionan las unidades de programador de subprocesos 36 y sombreador 38, después de describir un ejemplo de las unidades de sombreador 38 con respecto a la FIG. 3.

[0079] La FIG. 3 es un diagrama de bloques que ilustra una unidad de sombreador de ejemplo 40 que se puede usar en la GPU de la FIG. 2. En algunos ejemplos, la unidad de sombreador 40 puede corresponder a una de las unidades de sombreador 38 mostradas en la FIG. 2 La unidad de sombreador 40 está configurada para ejecutar uno o más tipos diferentes de programas sombreadores basándose en la información indicativa de una configuración de subprocesos. La unidad de sombreador 40 incluye una unidad de control 42, un almacén de instrucciones 44, elementos de procesamiento 46A-46H (colectivamente "elementos de procesamiento 46"), registros 48 y una memoria local 50.

[0080] La unidad de control 42 puede recibir información indicativa de una configuración de subprocesos para procesar elementos de datos, y hacer que uno o más subprocesos se ejecuten en elementos de procesamiento 46 basados en un programa sombreador y basados en la información indicativa de la configuración de subprocesos. El programa sombreador se puede almacenar en el almacén de instrucciones 44. Los elementos de datos pueden almacenarse, por ejemplo, en registros 48, memoria local 50 y/o una memoria externa. En algunos ejemplos, la unidad de control 42 puede configurarse adicionalmente para asignar registros 48 a los elementos de procesamiento 46A-46H basándose en el programa sombreador que será ejecutado por la unidad de sombreador 40 y/o basándose en la configuración del subproceso.

[0081] El almacén de instrucciones 44 está configurado para almacenar la totalidad o parte de uno o más programas sombreadores (por ejemplo, instrucciones de programa sombreador) que deben ejecutarse mediante la unidad de sombreador 40. El almacén de instrucciones 44 puede ser cualquier tipo de unidad de almacenamiento que incluya, por ejemplo, memoria volátil, memoria no volátil, memoria caché, memoria de acceso aleatorio (RAM), RAM estática (SRAM), RAM dinámica (DRAM), etc. Cuando el almacén de instrucciones 44 es una memoria caché, el almacén de instrucciones 44 puede almacenar en memoria caché uno o más programas sombreadores que se almacenan en una memoria externa a la unidad de sombreador 40. Aunque el almacén de instrucciones 44 se ilustra dentro de la unidad de sombreador 40, en otros ejemplos, el almacén de instrucciones 44 puede ser externo a la unidad de sombreador 40.

[0082] Los elementos de procesamiento 46 están configurados para ejecutar subprocesos de un programa sombreador. Cada uno de los elementos de procesamiento 46 puede ejecutar un subproceso diferente. Un subproceso puede referirse a una instancia de un programa sombreador que se ejecuta con respecto a un elemento de datos que es particular al subproceso. De este modo, cada uno de los elementos de procesamiento 46 puede recibir órdenes de ejecutar una instancia de una instrucción de un programa sombreador con respecto a elementos de datos potencialmente diferentes. La colección de subprocesos que se ejecutan en paralelo en el procesamiento de elementos 46A-46H en un punto común en el tiempo puede denominarse una onda de subprocesos.

[0083] En la unidad de sombreador 40 de la FIG. 3, los elementos de procesamiento 46 pueden ser elementos de procesamiento de instrucción única, datos múltiples (SIMD). Los elementos de procesamiento SIMD se refieren a elementos de procesamiento que, cuando se activan, están configurados para ejecutar la misma instrucción al mismo tiempo con respecto a datos diferentes. Esto puede permitir que los elementos de procesamiento 46 ejecuten una pluralidad de subprocesos de un programa sombreador en paralelo con respecto a diferentes elementos de datos. En algunos casos, cada uno de los elementos de procesamiento 46 puede ejecutar instrucciones de un programa sombreador basándose en un contador de programa común que apunte a una instrucción contenida en la memoria de instrucciones 44.

[0084] Si uno o más de los elementos de procesamiento 46 se desactivan, entonces dichos elementos de procesamiento 46 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 46 para implementar

instrucciones de ramificación condicional donde la condición de ramificación se satisfaga para algunos subprocesos y no se satisfaga para otros subprocesos.

[0085] En algunos ejemplos, cada uno de los elementos de procesamiento 46 puede incluir y/o corresponder a una unidad lógica aritmética (ALU). En ejemplos adicionales, cada uno de los elementos de procesamiento 46 puede implementar la funcionalidad ALU. La funcionalidad ALU puede incluir suma, resta, multiplicación, etc. En ejemplos adicionales, cada uno de los elementos de procesamiento 46 puede ser una ALU escalar o una ALU vectorial. Una ALU escalar puede funcionar en elementos de datos escalares, y una ALU vectorial puede funcionar en elementos de datos vectoriales. Un elemento de datos escalares puede incluir un solo valor correspondiente a un solo componente para un escalar. Un elemento de datos vectoriales puede incluir múltiples valores correspondientes a los múltiples componentes de un vector. En los ejemplos donde los elementos de procesamiento 46 son ALU escalares, si los elementos de datos vectoriales son procesados por la unidad de sombreador 40, cada uno de los componentes del vector puede, en algunos ejemplos, procesarse en paralelo mediante un subconjunto de elementos de procesamiento 46. Por ejemplo, los elementos de procesamiento 46A, 46B, 46C y 46D pueden procesar un vector de cuatro componentes en paralelo.

[0086] Cada uno de los elementos de procesamiento 46 puede leer las instrucciones del almacén de instrucciones 44 y/o leer los elementos de datos de uno o más de los registros 48, la memoria local 50 y una memoria externa. Cada uno de los elementos de procesamiento 46 puede escribir datos de salida en uno o más de los registros 48, la memoria local 50 y una memoria externa.

[0087] Los registros 48 pueden asignarse dinámicamente a diversos elementos de procesamiento 46. En algunos casos, algunos o todos los registros 48 pueden servir como registros de entrada y/o registros de salida para los diversos subprocesos que se ejecutan en la unidad de sombreador 40. Un registro de entrada puede referirse a un registro que almacena elementos de datos de entrada (por ejemplo, vértices de entrada, primitivas de entrada) para un programa sombreador, y un registro de salida puede referirse a un registro que almacena elementos de datos de salida (por ejemplo, vértices de salida, primitivas de salida) para un programa sombreador.

[0088] La memoria local 50 puede ser cualquier tipo de memoria que incluye, por ejemplo, memoria volátil, memoria de acceso aleatorio (RAM), memoria RAM estática (SRAM), RAM dinámica (DRAM), etc. En algunos ejemplos, el espacio de direcciones para la memoria local 50 puede ser local a los elementos de procesamiento 46 incluidos en la unidad de sombreador 40. En otras palabras, es posible que otras unidades de sombreador y/u otras partes de la GPU 30 no puedan acceder directamente a la memoria local 50. De manera similar, es posible que un dispositivo principal no pueda acceder directamente a la memoria local 50. En algunos ejemplos, la memoria local 50 puede implementarse en el mismo chip que la unidad de sombreador 40 y/o la GPU 30.

[0089] Ahora se describirá el funcionamiento general del programador de subprocesos 36 y la unidad de sombreado 40. La GPU 30 (por ejemplo, el motor de comandos 32) carga un programa sombreador en el almacén de instrucciones 44 o en un espacio de memoria al que puede acceder el almacén de instrucciones 44. El programador de subprocesos 36 recibe información indicativa de uno o más elementos de datos para procesar con un programa sombreador, determina una configuración de subprocesos para ejecutar el programa sombreador con respecto a los elementos de datos, y proporciona información indicativa de la configuración de subprocesos a la unidad 40 de sombreado. La configuración del subproceso puede especificar asignaciones de cada uno de los elementos de datos a una o más instancias de un programa sombreador para ser ejecutadas por la unidad de sombreador 40. Cada una de las instancias del programa sombreador puede configurarse para realizar el procesamiento del sombreador (por ejemplo, ejecutar las instrucciones del programa sombreador) con respecto a los elementos de datos que se asignan a la instancia respectiva del programa sombreador.

[0090] La unidad de control 42 recibe la información indicativa de la configuración del subproceso, y hace que los elementos de procesamiento 46A-46H ejecuten una o más instancias del programa sombreador basándose en la configuración del subproceso. Para hacer que los elementos de procesamiento 46A-46H ejecuten una o más instancias del programa sombreador basándose en la configuración del subproceso, la unidad de control 42 puede cargar los registros de entrada de los elementos de procesamiento 46A-46H con los elementos de datos de entrada (por ejemplo, vértices de entrada) asociados con los subprocesos que se ejecutarán antes de ejecutar las instancias del programa sombreador. Por ejemplo, se pueden asignar uno o más registros de entrada a cada uno de los elementos de procesamiento 46A-46H, y para cada una de las instancias del programa sombreador a ejecutar, la unidad de control 42 puede cargar los registros de entrada que corresponden a la instancia de sombreador respectiva con los elementos de datos de entrada a ser ejecutados por la instancia de sombreador respectiva. De esta manera, una unidad de sombreador 40 puede ejecutar una pluralidad de instancias de un programa sombreador.

[0091] De acuerdo con esta divulgación, la unidad de sombreador 40 puede estar configurada para ejecutar un programa sombreador que realiza el procesamiento de sombreador de vértices y que genera múltiples vértices de salida para cada vértice de entrada que es recibido por el programa sombreador. Por ejemplo, la unidad de sombreador 40 puede ejecutar una pluralidad de instancias del programa sombreador de tal manera que cada una de las instancias del programa sombreador reciba uno respectivo de una pluralidad de vértices de entrada y genere múltiples vértices de salida en respuesta a la recepción del uno respectivo de la pluralidad de vértices de entrada. Cada una de las

instancias del programa sombreador puede ser ejecutada por uno respectivo de los elementos de procesamiento 46A-46H.

[0092] La ejecución de un programa sombreador que genera múltiples vértices de salida para cada uno de los vértices de entrada que recibe un programa sombreador puede, en algunos ejemplos, permitir que el número de elementos de procesamiento 46A-46H que se utilizan para ejecutar el programa sombreador para un conjunto particular de vértices de entrada sea reducido. La reducción del número de elementos de procesamiento 46A-46H que se utilizan para ejecutar un programa sombreador puede reducir los recursos de procesamiento utilizados por la GPU 30 y/o la potencia consumida por la GPU 30. De esta manera, se puede mejorar el rendimiento y/o el consumo de energía de la GPU 30.

[0093] Por otra parte, la ejecución de un programa sombreador que genera múltiples vértices de salida para cada uno de los vértices de entrada que son recibidos por un programa sombreador puede, en otros ejemplos, ser utilizado para implementar un modelo de programación sombreador de vértices que permite que se generen múltiples vértices de salida para cada invocación de un programa sombreador de vértices. Típicamente, los modelos de programación de sombreador de vértices especifican que un programa sombreador de vértices debe invocarse una vez para cada vértice de entrada y que el programa sombreador de vértices debe generar un vértice de salida individual para cada invocación del programa sombreador de vértices. Implementar un modelo de programación de sombreador de vértices que permita generar múltiples vértices de salida para cada invocación de un programa sombreador de vértices puede aumentar la flexibilidad de la programación de sombreador de vértices que puede ser utilizada por un programador de gráficos.

[0094] Además, la ejecución de un programa sombreador que genera múltiples vértices de salida para cada uno de los vértices de entrada que son recibidos por un programa sombreador puede, en ejemplos adicionales, permitir que el procesamiento de sombreador de geometría para cada una de las primitivas se realice con una sola instancia del programa sombreador de vértices/geometría fusionado por primitiva. Permitir que el procesamiento del sombreador de geometría se realice con una sola instancia del programa sombreador de vértices/geometría fusionado por primitiva puede reducir el número de instancias del programa sombreador de vértices/geometría fusionado que son necesarias para procesar un conjunto particular de primitivas en relación con las técnicas que requiere que se ejecuten múltiples instancias de un programa sombreador de vértices/geometría fusionado para cada primitiva. La reducción del número de instancias de un programa sombreador de vértices/geometría fusionado que se utilizan para procesar primitivas puede reducir los recursos de procesamiento utilizados por la GPU 30, reducir el número de asignaciones de recursos realizadas por la GPU 30 y/o reducir la energía consumida por la GPU 30. De esta manera, se puede mejorar el rendimiento y/o el consumo de energía de la GPU 30 que realiza el sombreado de vértice programable y el sombreado de geometría programable con programas sombreadores de vértices/geometría fusionados.

[0095] La FIG. 4 es un diagrama conceptual que ilustra una tira de triángulos de ejemplo que puede procesarse utilizando las técnicas de ejecución de programa sombreador de ejemplo de esta divulgación. Como se muestra en la FIG. 4, la tira de triángulos de ejemplo incluye ocho vértices (v0, v1, v2, v3, v4, v5, v6, v7) y seis primitivas (p0, p1, p2, p3, p4, p5). Varias de las primitivas en la tira de triángulos de la FIG. 4 comparten vértices. En otras palabras, el mismo vértice puede formar parte de dos o más primitivas.

[0096] La FIG. 5 es un diagrama conceptual que ilustra configuraciones de subprocesos de ejemplo 52, 54 que se pueden usar para ejecutar una pluralidad de instancias de un programa sombreador de vértices/geometría fusionado para procesar la tira de triángulos mostrada en la FIG. 4 de acuerdo con esta divulgación. La configuración de subprocesos 52 es un ejemplo de una configuración de subprocesos que se puede usar para ejecutar una pluralidad de instancias de un programa sombreador de vértices/geometría fusionado de acuerdo con un modo de ejecución de programa sombreador replicado. La configuración de subprocesos 54 es un ejemplo de una configuración de subprocesos que se puede usar para ejecutar una pluralidad de instancias de un programa sombreador de vértices/geometría fusionado de acuerdo con un modo de ejecución de programa sombreador no replicado.

[0097] La configuración de subprocesos 52 incluye configuraciones de onda 56, 58, 60. La configuración de subprocesos 54 incluye la configuración de onda 62. Cada una de las configuraciones de onda 56, 58, 60, 62 puede corresponder a una onda de instancias de un programa sombreador de vértices/geometría fusionado que se procesan en paralelo mediante la unidad de sombreador 40 (por ejemplo, los elementos de procesamiento 46A-46H de la unidad de sombreador 40). Cada una de las configuraciones de onda 56, 58, 60, 62 puede ser indicativa de la asignación de elementos de datos a las instancias del programa sombreador de vértices/geometría fusionado incluido en la onda respectiva. Cada una de las instancias del programa sombreador de vértices/geometría fusionado en una sola onda se puede ejecutar en una unidad separada de elementos de procesamiento 46A-46H en la unidad de sombreador 40. Cada una de las instancias del programa sombreador de vértices/geometría fusionado puede denominarse de forma alternativa un subproceso y/o una fibra.

[0098] Como se muestra en la FIG. 5, la primera fila de cada una de las configuraciones de onda 56, 58, 60, 62 especifica una identificación de fibra (ID de fibra), la segunda fila de cada una de las configuraciones de onda 56, 58, 60, 62 especifica una identificación de vértice (ID de vértice), y la tercera fila de cada una de las configuraciones de onda 56, 58, 60, 62 especifica una identificación primitiva (ID primitiva). Cada una de las instancias del programa

sombreador de vértices/geometría fusionado en una sola onda puede corresponder a un valor de ID de fibra único y respectivo. Cada uno de los valores de ID de vértice corresponde al vértice numérico similar en la tira de triángulos de la FIG. 4. De manera similar, cada uno de los valores de ID primitiva corresponde a la primitiva de números similares en la tira de triángulos de la FIG. 4.

[0099] Cada una de las columnas de configuraciones de onda 56, 58, 60, 62 representa una asignación de uno o ambos de un vértice y una primitiva a una fibra particular. Por ejemplo, la primera columna de la configuración de onda 56 representa una asignación del vértice v0 y la primitiva p0 a la fibra 0, y la sexta columna de la configuración de onda 56 representa una asignación del vértice v3 y la primitiva p1 de la fibra 5.

[0100] Los cuadros en las configuraciones de onda 56, 58, 60, 62 que no incluyen ningún valor indican que no se asigna ningún elemento de datos de ese tipo particular a la fibra. Por ejemplo, la octava columna de la configuración de onda 56 representa una asignación de primitiva p1 a la fibra 7 y una asignación de ningún vértice a la fibra 7, y la octava columna de la configuración de onda 62 representa una asignación del vértice v7 a la fibra 7 y una asignación de ninguna primitiva a la fibra 7.

[0101] Cuando un vértice y una primitiva se asignan a una fibra, la fibra puede realizar un procesamiento de sombreador de vértices con respecto al vértice asignado a la fibra y el procesamiento de sombreador de geometría con respecto a la primitiva asignada a la fibra. Por ejemplo, la fibra 0 en la configuración de onda 56 puede realizar el procesamiento de sombreador de vértices con respecto al vértice v0 y el procesamiento de sombreador de geometría con respecto a la primitiva p0.

[0102] Cuando se asigna una primitiva a la fibra y un vértice no se asigna a la fibra, entonces la fibra puede realizar un procesamiento de sombreador de geometría con respecto a la primitiva y no necesariamente puede realizar ningún procesamiento de sombreador de vértices. Por ejemplo, la fibra 7 en la configuración de onda 56 puede no realizar ningún procesamiento de sombreador de vértices y puede realizar el procesamiento de sombreador de geometría con respecto a la primitiva p1.

[0103] Cuando se asigna una primitiva a una fibra y un vértice no se asigna a la fibra, entonces la fibra puede realizar un procesamiento de sombreador de geometría con respecto a la primitiva y no necesariamente puede realizar ningún procesamiento de sombreador de vértices. Por ejemplo, la fibra 7 en la configuración de onda 56 puede no realizar ningún procesamiento de sombreador de vértices y puede realizar el procesamiento de sombreador de geometría con respecto a la primitiva p1.

[0104] Cuando se asigna un vértice a la fibra y no se asigna una primitiva a la fibra, la fibra puede realizar un procesamiento de sombreador de vértices con respecto al vértice y no necesariamente puede realizar ningún procesamiento de sombreador de geometría. Por ejemplo, la fibra 7 en la configuración de onda 62 puede realizar un procesamiento de sombreador de vértices con respecto al vértice v7 y puede no realizar ningún procesamiento de sombreador de geometría.

[0105] Para el modo de ejecución del programa sombreador replicado (es decir, configuraciones de onda 56, 58, 60), cada una de las fibras recibe un solo vértice y genera un solo vértice. Por lo tanto, varias fibras realizan el procesamiento de sombreador de geometría para cada una de las primitivas. En el ejemplo específico de la FIG. 5, se ejecutan cuatro fibras (es decir, cuatro instancias del sombreador de vértices/geometría fusionado) para cada primitiva.

[0106] Tenga en cuenta que este ejemplo podría haber usado tres fibras (es decir, el número de vértices incluidos en cada una de las primitivas de salida para la etapa de sombreador de geometría) en lugar de cuatro fibras. Sin embargo, el programa sombreador de geometría definió que el recuento máximo de vértices de salida en este ejemplo es cuatro, aunque no se utilice uno de los vértices de salida. En algunos ejemplos, un sombreador de geometría puede definir un recuento máximo de vértices de salida entre 1 y 1024 vértices. El recuento máximo de vértices de salida puede no ser necesariamente igual al número de vértices del tipo de primitiva de entrada.

[0107] Para el modo no replicado (es decir, la configuración de onda 62), cada una de las fibras recibe un solo vértice y puede generar múltiples vértices. Por lo tanto, para cada una de las primitivas, se utiliza una sola fibra para realizar el procesamiento de sombreador de geometría para la primitiva. En el ejemplo de la FIG. 5, cada fibra (por ejemplo, la instancia del programa sombreador de vértices/geometría fusionado) puede generar tres vértices. Cada conjunto de tres vértices emitidos por cada una de las fibras puede corresponder a una de las primitivas de triángulos mostradas en la FIG. 4.

[0108] Como se muestra en la FIG. 5, para el modo replicado (es decir, las configuraciones de onda 56, 58, 60), la ejecución repetida del procesamiento de sombreador de geometría para cada una de las primitivas hace que se repitan las operaciones de ALU y las asignaciones de recursos repetidos. Esto puede reducir el rendimiento de la GPU 30 y/o aumentar el consumo de energía de la GPU 30. Además, la ejecución repetida del procesamiento del sombreador de geometría consume varias fibras/elementos de procesamiento por primitiva, lo cual puede reducir aún más el rendimiento de la GPU 30. Además, los vértices no se reutilizan en el modo replicado, lo cual evita que la GPU 30 aproveche las mejoras de potencia, rendimiento y ancho de banda de memoria de la reutilización de vértice.

[0109] Por el contrario, el modo no replicado (es decir, la configuración de onda 62) ejecuta una única instancia de las funciones del sombreador de geometría para cada una de las primitivas procesadas por el sombreador de geometría, lo cual puede reducir el número de operaciones ALU realizadas por primitiva, reducir el número de asignaciones de recursos realizadas por primitiva, y/o reducir el número de fibras/elementos de procesamiento consumidos por primitiva. La reducción de las operaciones de ALU por primitiva, las asignaciones de recursos por primitiva y/o las fibras/elementos de procesamiento consumidos por primitiva pueden mejorar el rendimiento de la GPU y/o reducir el consumo de energía de la GPU. Además, la reutilización de vértices puede ocurrir en el modo no replicado, reduciendo así la cantidad de procesamiento de vértices que se realiza por primitiva. Reducir la cantidad de procesamiento de vértices puede mejorar el rendimiento, ahorrar energía y/o reducir el ancho de banda de la memoria en un sistema de procesamiento de gráficos.

[0110] Aunque el modo no replicado puede proporcionar varias ventajas en términos de mejorar el rendimiento y el consumo de energía del procesamiento del sombreador de geometría y en cuanto a permitir que se produzca la reutilización de vértices, puede haber desventajas en el modo no replicado. Por ejemplo, debido a que cada fibra puede generar múltiples vértices, la cantidad total de espacio de almacenamiento en el chip (por ejemplo, los registros 48 de la unidad de sombreador 40) necesaria para almacenar los vértices de salida para una sola onda de fibras de sombreado de vértices/geometría puede ser significativa, particularmente en casos donde el sombreador de geometría realiza una cantidad relativamente grande de amplificación de vértice. Aunque la memoria fuera del chip podría usarse para almacenar vértices, el uso de dicha memoria puede degradar significativamente el rendimiento de la unidad de sombreador 40.

[0111] En el ejemplo de la FIG. 5, cada fibra es capaz de generar hasta cuatro vértices, lo cual da como resultado que el espacio de almacenamiento en el chip deba asignarse a 32 vértices por onda. En contraste, el modo replicado genera un solo vértice por fibra. Por lo tanto, para una sola onda, el espacio de almacenamiento puede limitarse en este ejemplo a ocho vértices.

[0112] En general, la cantidad de espacio de almacenamiento (por ejemplo, registros de 48) para almacenar los vértices de salida en la unidad de sombreador 40 puede ser limitada. Debido a que el modo no replicado para ejecutar programas sombreadores de vértices/geometría fusionados permite que cada una de las fibras genere múltiples vértices, la cantidad de espacio de almacenamiento necesario para almacenar vértices de salida para el modo no replicado puede ser mayor que la que se necesita para el modo replicado. Si la cantidad de espacio de almacenamiento incluido en la unidad de sombreador 40 no es suficiente para almacenar los vértices de salida para un conjunto dado de fibras en una onda, es posible que se deban realizar accesos a la memoria externa, lo cual puede reducir significativamente el rendimiento de la unidad de sombreador 40.

[0113] Como se analizó anteriormente, el modo no replicado para la ejecución de programas sombreadores de vértices/geometría fusionados puede proporcionar un mejor rendimiento y/o la reducción de consumo de energía para la ejecución de un conjunto dado de fibras en una onda. Sin embargo, si los requisitos de espacio de almacenamiento de vértices de salida asociados con la ejecución de las fibras en la onda son mayores que la cantidad de espacio de almacenamiento de vértices de salida disponible en la unidad de sombreador 40, entonces las mejoras de rendimiento y/o potencia logradas al no replicar el procesamiento de sombreador de geometría pueden ser superadas por la reducción en el rendimiento ocasionada por los accesos a la memoria externa.

[0114] Por lo tanto, el modo no replicado para la ejecución de programas sombreadores de vértices/geometría fusionados puede proporcionar un mejor rendimiento y/o consumo de energía reducido si los requisitos de espacio de almacenamiento para almacenar los vértices de salida son relativamente pequeños (por ejemplo, si los requisitos de espacio de almacenamiento de vértices de salida son menores o iguales a la cantidad de espacio de almacenamiento de vértices de salida contenido en la unidad de sombreador 40). Por otro lado, el modo replicado para ejecutar programas sombreadores de vértices/geometría fusionados puede proporcionar un mejor rendimiento si los requisitos de espacio de almacenamiento para almacenar vértices de salida son relativamente altos (por ejemplo, si los requisitos de espacio de almacenamiento de vértices de salida son mayores que la cantidad de espacio de almacenamiento de vértices de salida contenido en la unidad de sombreador 40).

[0115] De acuerdo con esta divulgación, la unidad de sombreador 40 puede configurare para cambiar selectivamente entre un modo no replicado para la ejecución de programas sombreadores de vértices/geometría fusionados y un modo replicado para la ejecución de programas sombreadores de vértices/geometría fusionados. Permitir que la unidad de sombreador 40 cambie entre un modo no replicado y un modo replicado para ejecutar programas sombreadores de vértices/geometría fusionados puede proporcionar un control y/o flexibilidad adicional a los usuarios de la GPU 30 para elegir modos de ejecución particulares que se adapten a los requisitos de procesamiento particulares, tales como, por ejemplo, requisitos de rendimiento, requisitos de consumo de energía, etc.

[0116] En algunos ejemplos, para cambiar entre el modo no replicado y el modo replicado para ejecutar un programa sombreador de vértices/geometría fusionado, la unidad de sombreador 40 puede configurarse para recibir información indicativa de un modo de ejecución de programa sombreador para ser utilizado para ejecutar un programa sombreador de vértices/geometría fusionado. La información indicativa de un modo de ejecución del programa sombreador que se

utilizará para ejecutar un programa sombreador puede incluir, por ejemplo, información indicativa de si se debe usar un modo no replicado para ejecutar el programa sombreador de vértices/geometría fusionado y/o información indicativa de si se va a usar un modo replicado para ejecutar el programa sombreador de vértices/geometría fusionado. En respuesta a la recepción de información indicativa del modo de ejecución de un programa sombreador que se utilizará para ejecutar un programa sombreador de vértices/geometría fusionado, la unidad de sombreador 40 (por ejemplo, la unidad de control 42) puede configurar uno o más componentes en la unidad de sombreador 40 basándose en el modo de ejecución del programa sombreador para ejecutar un programa sombreador de vértices/geometría fusionado basado en el modo de ejecución del programa sombreador.

[0117] En ejemplos adicionales, para cambiar entre el modo no replicado y el modo replicado para ejecutar un programa sombreador de vértices/geometría fusionado, el programador de subprocesos 36 puede configurarse para determinar y/o generar una configuración de subprocesos basada en el modo de ejecución del programa sombreador seleccionado (por ejemplo, el modo no replicado o el modo replicado). Por ejemplo, el programador de subprocesos 36 puede asignar cada una de una pluralidad de primitivas y una pluralidad de vértices a las respectivas fibras, y generar una configuración de subprocesos basada en las asignaciones. El programador de subprocesos 36 puede enviar la configuración de subprocesos a las unidades de sombreador 38 para hacer que las unidades de sombreador 38 ejecuten un programa sombreador de vértices/geometría fusionado basado en la configuración del subproceso.

[0118] En ejemplos adicionales, para cambiar entre el modo no replicado y el modo replicado para ejecutar un programa sombreador de vértices/geometría fusionado, el programa sombreador de vértices fusionado puede incluir código que realiza selectivamente varias operaciones basadas en el modo de ejecución de programa sombreador seleccionado (por ejemplo, modo replicado respecto a modo no replicado). Por ejemplo, el código del sombreado puede hacer selectivamente que las instancias del sombreador de vértices/geometría fusionado emitan un vértice para el modo no replicado y varios vértices para el modo replicado. Como otro ejemplo, el código del sombreado puede hacer selectivamente que las instancias del sombreador de vértices/geometría fusionado calculen las ubicaciones de la memoria locales para almacenar y/o recuperar vértices sombreados de vértices de acuerdo con diferentes técnicas, dependiendo del modo de ejecución del programa sombreador seleccionado.

[0119] Por ejemplo, si se selecciona el modo no replicado, el programador de subprocesos 36 puede generar la configuración de subprocesos de manera que cada una de una pluralidad de primitivas se asigna a N fibras (por primitiva) para el procesamiento del sombreador de geometría. Por otro lado, si se selecciona el modo replicado, el programador de subprocesos 36 puede generar la configuración de subprocesos de manera que cada una de la pluralidad de primitivas se asigne a una fibra (por primitiva) para el procesamiento del sombreador de geometría.

[0120] En algunos ejemplos, N puede ser un número entero mayor que o igual a 2. En otros ejemplos, N puede ser igual a un valor de recuento de vértices de salida máximo que se especifica mediante un programa sombreador de geometría que se implementa mediante el programa sombreador de vértices/geometría fusionado. El valor máximo del recuento de vértices de salida puede ser indicativo de un número máximo de vértices de salida que debe generar el programa sombreador de geometría para cada primitiva que sea procesada por el programa sombreador de geometría.

[0121] Como otro ejemplo, si se selecciona el modo no replicado, el programador de subprocesos 36 puede generar la configuración de subprocesos de tal manera que cada una de las fibras emite y/o genera un vértice de salida (por fibra). Por otro lado, si se selecciona el modo replicado, el programador de subprocesos 36 puede generar la configuración de subprocesos de manera que cada una de las fibras emita y/o genere M vértices de salida (por fibra).

[0122] En algunos ejemplos, M puede ser un número entero mayor que o igual a 2. En otros ejemplos, M puede ser igual a la cantidad de vértices que se generan para cada una de las primitivas procesadas por una etapa de sombreador de geometría que se implementa mediante el programa sombreador de vértices/geometría fusionado. En ejemplos adicionales, M puede ser menor o igual a N.

[0123] Como un ejemplo adicional, si se selecciona el modo no replicado, el programador de subprocesos 36 puede generar el configuración de subprocesos de tal manera que cada uno de una pluralidad de vértices se asigna a una fibra (por ejemplo, una fibra por onda) para el procesamiento de sombreador de vértices. Por otro lado, si se selecciona el modo replicado, el programador de subprocesos 36 puede generar la configuración de subprocesos de manera que cada uno de una pluralidad de vértices se asigne a K fibras para el procesamiento de sombreador de vértices, donde K es un número entero igual a la cantidad de primitivas que contienen vértice respectivo.

[0124] En otras palabras, si se selecciona el modo no replicado, el programador de subprocesos 36 puede generar la configuración de subprocesos para permitir la reutilización de vértices, y si se selecciona el modo replicado, el programador de subprocesos 36 puede generar la configuración de subprocesos para no permitir la reutilización de vértices. La reutilización de vértices puede referirse a técnicas de procesamiento que permiten que las instancias de un programa sombreador que realiza el procesamiento de sombreador de geometría con respecto a diferentes primitivas utilicen un vértice sombreado de vértice generado por una única instancia de un programa sombreador que realiza el procesamiento de sombreador de vértices.

[0125] Por ejemplo, la reutilización de vértices puede ocurrir cuando una primera fibra realiza un procesamiento de sombreador de vértices con respecto a uno de una pluralidad de vértices para generar un vértice sombreado de vértice, una segunda fibra realiza un procesamiento de sombreador de geometría con respecto a una primera primitiva basada en el vértice sombreado de vértice generado por la primera fibra, y una tercera fibra realiza el procesamiento de sombreador de geometría con respecto a una segunda primitiva basada en el vértice sombreado de vértice generado por la primera fibra. Como se muestra en la FIG. 4, el vértice v1 se incluye en las primitivas p0 y p1, y como se muestra en la FIG. 5, el procesamiento de sombreador de geometría para las fibras 0 y 1 puede compartir el vértice sombreado de vértice v1, que es generado por la fibra 1.

[0126] De acuerdo con algunos aspectos de esta divulgación, las técnicas para ejecutar programas sombreadores pueden incluir técnicas para seleccionar entre el modo no replicado y el modo replicado para ejecutar un programa sombreador de vértices/geometría fusionado y hacer que la unidad de sombreador 40 ejecute el programa sombreador de vértices/geometría de acuerdo con el modo de ejecución del programa sombreador seleccionado. En algunos ejemplos, las técnicas para seleccionar entre el modo no replicado y el modo replicado pueden seleccionar entre el modo no replicado y el modo replicado basándose en la información indicativa de una cantidad total de espacio de almacenamiento requerido para almacenar los vértices de salida asociados con uno o más invocaciones API de un programa sombreador de geometría que se implementa mediante el programa sombreador de vértices/geometría fusionado. Se puede implementar un programa sombreador de geometría mediante un programa sombreador de vértices/geometría fusionado si el código compilado para el programa sombreador de vértices/geometría fusionado se genera basándose en el programa sombreador de geometría.

[0127] En estos ejemplos, si la cantidad total de espacio de almacenamiento requerido para almacenar vértices de salida es relativamente pequeña (por ejemplo, menor que o igual a un umbral), entonces el técnicas para seleccionar entre el modo no replicado y el modo replicado pueden seleccionar el modo no replicado como el modo de ejecución del programa sombreador seleccionado para ejecutar un programa sombreador de vértices/geometría fusionado. Esto puede permitir que un sistema de procesamiento de gráficos obtenga las ventajas de un mayor rendimiento y/o un consumo de energía reducido del modo no replicado sin requerir acceso a la memoria fuera del chip para los vértices sombreados de vértice. Por otro lado, si la cantidad total de espacio de almacenamiento requerido para almacenar los vértices de salida es relativamente grande (por ejemplo, mayor que un umbral), entonces las técnicas para seleccionar entre el modo no replicado y el modo replicado pueden seleccionar el modo replicado como el modo de ejecución del programa sombreador seleccionado para ejecutar el programa sombreador de vértices/geometría fusionado. Esto puede permitir que un sistema de procesamiento de gráficos evite los accesos a la memoria fuera del chip que pueden ser necesarios si se usara el modo no replicado en tales casos y evitar los inconvenientes de rendimiento asociados con dichos accesos a la memoria fuera del chip.

[0128] En algunos ejemplos, la cantidad total de espacio de almacenamiento requerido para almacenar los vértices de salida se puede determinar basado en el número total de vértices de salida generados por una o más invocaciones de la API del programa sombreador de geometría y la cantidad total de espacio de almacenamiento requerido para almacenar cada uno de los vértices de salida. Por ejemplo, la cantidad total de espacio de almacenamiento requerido para almacenar los vértices de salida puede ser igual al producto de la cantidad total de vértices generados por una o más invocaciones API del programa sombreador de geometría y la cantidad total de espacio de almacenamiento requerido para almacenar cada de los vértices de salida. En algunos casos, la cantidad total de vértices generados por una o más invocaciones de API del programa sombreador de geometría puede corresponder al número total de vértices incluidos en el conjunto de primitivas de salida generadas por una o más invocaciones de API del programa sombreador de geometría.

[0129] En ejemplos adicionales, para seleccionar entre el modo no replicado y el modo replicado basado en una cantidad total de espacio de almacenamiento requerido para almacenar los vértices de salida asociados con una o más invocaciones API de un programa sombreador de geometría, las técnicas para la selección entre el modo no replicado y el modo replicado pueden determinar la cantidad total de espacio de almacenamiento requerido para almacenar los vértices de salida para una onda de ejecución del programa sombreador de vértices/geometría fusionado. Una onda de ejecución del programa sombreador de vértices/geometría fusionado puede corresponder a las instancias de L del programa sombreador de vértices/geometría fusionado que son ejecutados en paralelo por la unidad de sombreador 40, donde L es igual al número de elementos de procesamiento en la unidad de sombreador 40 que ejecutan un programa sombreador en paralelo.

[0130] En algunos ejemplos, la información indicativa de la cantidad total de espacio de almacenamiento requerido para almacenar los vértices de salida para uno o más invocaciones de la API del programa sombreador de geometría puede corresponder a la cantidad total de espacio de almacenamiento requerido para almacenar los vértices de salida para una onda de ejecución del programa sombreador de vértices/geometría fusionado. Una onda de ejecución del programa sombreador de vértices/geometría fusionado puede corresponder a las instancias de L del programa sombreador de vértices/geometría fusionado que se ejecutan en paralelo mediante una unidad de sombreador 40, donde L es igual al número de elementos de procesamiento en la unidad de sombreador 40 que ejecutan un programa sombreador en paralelo.

[0131] En estos ejemplos, si la cantidad total de espacio de almacenamiento requerido para almacenar vértices de salida es menor que o igual a un umbral, entonces las técnicas para seleccionar entre el modo no replicado y el modo replicado pueden seleccionar el modo no replicado como el modo de ejecución del programa sombreador seleccionado para ejecutar un programa sombreador de vértices/geometría fusionado. Por otro lado, si la cantidad total de espacio de almacenamiento requerido para almacenar los vértices de salida es mayor que un umbral, entonces las técnicas para seleccionar entre el modo no replicado y el modo replicado pueden seleccionar el modo replicado como el modo de ejecución del programa sombreador seleccionado. para ejecutar el programa sombreador de vértices/geometría fusionado. En algunos casos, el umbral en tales ejemplos puede corresponder a la cantidad total de espacio de almacenamiento disponible en una unidad de sombreador 40 (por ejemplo, los registros 48) para almacenar los vértices de salida asociados con una onda de ejecución del programa sombreador de vértices/geometría fusionado.

[0132] En ejemplos adicionales, la información indicativa de una cantidad total de espacio de almacenamiento requerido para almacenar vértices de salida asociados con una o más invocaciones API de un programa sombreador de geometría puede corresponder a una cantidad de amplificación de vértice realizada por una invocación API de un programa sombreador de geometría que se implementa mediante el programa sombreador de vértices/geometría fusionado. La cantidad de amplificación de vértice puede referirse a la relación de vértices de salida a vértices de entrada asociados con una invocación API de un programa sombreador de geometría que se implementa mediante el programa sombreador de vértices/geometría fusionado.

[0133] En estos ejemplos, si la cantidad de amplificación de vértice es menor que o igual a un umbral, entonces las técnicas para seleccionar entre el modo no replicado y el modo replicado pueden seleccionar el modo no replicado como el modo de ejecución del programa sombreador seleccionado para ejecutar un programa sombreador de vértices/geometría fusionado. Por otro lado, si la cantidad de amplificación de vértice es mayor que un umbral, entonces las técnicas para seleccionar entre el modo no replicado y el modo replicado pueden seleccionar el modo replicado como el modo de ejecución del programa sombreador seleccionado para ejecutar el programa sombreador de vértices/geometría fusionado.

[0134] En algunos ejemplos, una o más de las técnicas descritas anteriormente para seleccionar entre el modo no replicado y el modo replicado para ejecutar un programa sombreador de vértices/geometría fusionado pueden ser implementadas por un dispositivo principal (por ejemplo, una unidad central de procesamiento central (CPU)) que es externa a la GPU 30. Por ejemplo, una o más de las técnicas para seleccionar entre el modo no replicado y el modo replicado puede implementarse mediante un controlador de gráficos que se ejecuta en un dispositivo principal y/o implementa mediante un compilador que se ejecuta en un dispositivo principal. En ejemplos adicionales, GPU 30 puede implementar una o más de las técnicas para seleccionar entre el modo no replicado y el modo replicado. Por ejemplo, una o más unidades de procesamiento de función fija y/o programables pueden implementar una o más de las técnicas de selección descritas anteriormente.

[0135] En ejemplos en los que las técnicas de selección son implementadas por un dispositivo principal. Las técnicas de esta divulgación pueden proporcionar a la GPU 30 información indicativa del modo de ejecución del programa sombreador seleccionado para la GPU 30. Por ejemplo, la GPU 30 puede incluir uno o más registros que se utilizan para almacenar información (por ejemplo, uno o más bits y/o uno o más valores) indicativos del modo de ejecución del programa sombreador seleccionado, y el dispositivo principal (por ejemplo, una CPU) puede proporcionar la información indicativa del modo de ejecución del programa sombreador seleccionado a la GPU 30 escribiendo uno o más valores indicativos del modo de ejecución del programa sombreador seleccionado en los registros.

[0136] La FIG. 6 es un diagrama conceptual que ilustra el flujo de procesamiento de ejemplo asociado con la ejecución de un programa sombreador de vértices/geometría fusionado de acuerdo con esta divulgación. La FIG. 7 ilustra el pseudo código asociado con la ejecución de un programa sombreador de vértices/geometría fusionado de acuerdo con esta divulgación. En general, el flujo de operaciones representado en la FIG. 6 puede corresponder a las instrucciones ilustradas en la FIG. 7.

[0137] En el ejemplo mostrado en la FIG. 6, la unidad de sombreador 40 escribe valores del sistema tales como atributos de vértice, id de vértice, id de instancia, id de primitiva, misc en una serie de registros R0, R1 y R2 (70). En algunos ejemplos, los registros pueden corresponder a los registros 48 de la unidad de sombreador 40. Típicamente, los valores del sistema se pueden almacenar en cualquier memoria no asignada de otra forma de la GPU 30. Al almacenar los valores generados por el sistema en una serie de registros en una ubicación predeterminada, la unidad de sombreador 40 puede acceder a los valores generados por el sistema para cada una de la etapa de procesamiento de sombreador de vértices y la etapa de procesamiento de sombreador de geometría. Por lo tanto, el programa sombreador de geometría especificado por el usuario no necesita compilarse basándose en el programa sombreador de vértices especificado por el usuario para determinar dónde se han almacenado los valores generados por el sistema. Más bien, la GPU 30 puede acceder a ubicaciones de memoria predeterminadas al realizar cada uno del procesamiento de sombreado de vértice y el procesamiento de sombreador de geometría para acceder a los valores generados por el sistema.

[0138] La unidad de sombreador 40 realiza operaciones de sombreado de vértice (72). Después de las operaciones de sombreado de vértice, la unidad de sombreador 40 escribe el contenido de los registros de propósito general (GPR)

(por ejemplo, los registros 48) (por ejemplo, un vértice de salida del procesamiento de sombreador de vértices) en la memoria local 50 (74). La unidad de sombreador 40 cambia a textura de sombreador de geometría y desviaciones constantes (76) y un contador de programa sombreador de geometría (78).

[0139] La unidad de sombreador 40 lee el contenido de la memoria local 50 (por ejemplo, vértices sombreados de vértice generados por el procesamiento de sombreador de vértices de una o más instancias del programa sombreador de vértices/geometría fusionado), y realiza operaciones de sombreado de geometría (80). Cuando se funciona de acuerdo con el modo no replicado, para cada emisión incluida en una fibra (por ejemplo, una instancia del programa sombreador de vértices/geometría fusionado), la unidad de sombreador 40 puede enviar un atributo de vértice a un caché de parámetros de vértice (VPC), un stream_id, y cualquier indicación de corte. Cuando funciona de acuerdo con el modo replicado, la unidad de sombreador 40 puede enviar un atributo de vértice a un caché de parámetros de vértice (VPC), así como una indicación de la posición de los vértices sombreados de geometría, un stream_id, cualquier indicación de corte y cualquier valor interpretado a una memoria caché de posición.

[0140] Con referencia a la FIG. 7, un programa sombreador de vértices/geometría fusionado puede incluir código de procesamiento de sombreador de vértices 82, código de procesamiento de sombreador de geometría 84 y código de parche 86. Un compilador (por ejemplo, ejecutándose en una CPU principal) puede generar un programa sombreador de vértices/geometría fusionado basado en un programa sombreador de vértices y un programa sombreador de geometría que se especifican por una aplicación de gráficos. En algunos ejemplos, el programa sombreador de vértices y el programa sombreador de geometría pueden compilarse de manera independiente antes de generar el programa sombreador de vértices/geometría fusionado, y el compilador puede agregar el código de procesamiento de sombreador de geometría compilado 84 al código de procesamiento de sombreador de vértices compilado 82 para generar el código compilado para el programa sombreador de vértices/geometría fusionado.

[0141] Como se muestra en la FIG. 7, el compilador puede insertar el código de parche 86 entre el código de procesamiento de sombreador de vértices 82 y el código de procesamiento de sombreador de geometría 84. El código de parche 86 puede hacer que la unidad de sombreador 40 almacene el vértice de salida generado por el código de procesamiento de sombreador de vértices 82 en los registros 48 de la unidad de sombreador 40, y realizar operaciones asociadas con el cambio de la unidad de sombreador 40 desde un modo de procesamiento de sombreador de vértices a un modo de procesamiento de sombreador de geometría. Las operaciones asociadas con cambiar la unidad de sombreador 40 de un modo de procesamiento de sombreador de vértices a un modo de procesamiento de sombreador de geometría pueden incluir una operación CHMSK y una operación CHSH. La operación CHMSK puede cambiar el puntero de recursos en la unidad de sombreador 40 a una desviación de recursos de sombreador de geometría. La operación CHSH puede cambiar el contador de programa en la unidad de sombreador 40 a un valor de contador de programa asociado con el código de procesamiento de sombreador de geometría 84.

[0142] De acuerdo con aspectos adicionales de esta divulgación, un compilador puede generar código compilado para un programa sombreador de vértices/geometría fusionado de tal manera que los códigos compilados incluyen instrucciones que hacen que la unidad de sombreador 40 ejecute selectivamente el programa sombreador de vértices/geometría fusionado de acuerdo con el modo no replicado o el modo replicado basado en información indicativa de un modo que se utilizará para la ejecución del programa sombreador de vértices/geometría fusionado. Por ejemplo, como se muestra en la FIG. 7, un compilador puede generar el código compilado para el sombreado de vértices/geometría fusionado de tal manera que el código compilado incluya los segmentos de código 88, 90 y 92. Cada uno de los segmentos de código 88, 90, 92 puede realizar varias operaciones de forma selectiva basándose en el modo de ejecución del programa sombreador seleccionado (por ejemplo, modo replicado vs. modo no replicado) para hacer que la unidad de sombreador 40 funcione de acuerdo con el modo de ejecución de programa sombreador seleccionado.

[0143] El segmento de código 88 puede incluir instrucciones que hacen que la unidad de sombreador 40 utilice selectivamente una de una primera fórmulas de cálculo de la dirección de memoria local y una segunda fórmula de cálculo de la dirección de la memoria local establecida basándose en el modo de ejecución del programa sombreador seleccionado. La fórmula de cálculo de la dirección de la memoria local seleccionada se puede usar para determinar qué ubicación en la memoria local 50 para almacenar la salida del procesamiento del sombreador de vértices (por ejemplo, un vértice sombreado de vértice) realizado por el programa sombreador de vértices/geometría fusionado. Por ejemplo, si se selecciona el modo no replicado como el modo de ejecución del programa sombreador, el código compilado puede hacer que la unidad de sombreador 40 use una primera fórmula de cálculo de dirección de memoria local que calcula la dirección de memoria local basándose en el valor de identificación de fibra (es decir,. *fiber_id_vertex*) para la fibra que se está ejecutando actualmente y el tamaño del vértice de salida (es decir, VERT_SIZE). Por otro lado, si se selecciona el modo replicado como modo de ejecución del programa sombreador, el código compilado puede hacer que la unidad de sombreador 40 use una segunda fórmula de cálculo de la dirección de la memoria local que calcula la dirección de la memoria local basándose en el valor de identificación de primitiva de la primitiva asignado a la fibra que se está ejecutando actualmente (es decir, *rel_primID*), el tamaño de la primitiva (es decir, PRIM_SIZE), el valor de identificación del vértice asignado a la fibra que se está ejecutando actualmente (es decir, *rel_vertex*) y el tamaño del vértice de salida (es decir, VERT_SIZE).

[0144] El segmento de código 90 puede incluir instrucciones que hacen que la unidad de sombreador 40 utilice selectivamente una de una primera fórmulas de cálculo de la dirección de memoria local y una segunda fórmula de cálculo de la dirección de la memoria local establecida basándose en el modo de ejecución del programa sombreador seleccionado. La fórmula de cálculo de la dirección de la memoria local seleccionada se puede usar para determinar desde qué ubicación en la memoria local 50 cargar los vértices de entrada (por ejemplo, vértices sombreados de vértice) asociados con el procesamiento del sombreador de geometría realizado por el programa sombreador de vértices/geometría fusionado. Por ejemplo, si se selecciona el modo no replicado como el modo de ejecución del programa sombreador, el código compilado puede hacer que la unidad de sombreador 40 use una primera fórmula de cálculo de la dirección de memoria local que calcula la dirección de la memoria local basándose en un valor de identificación de vértice (es decir,. *vertex_id*) asociado con el vértice a cargar y el tamaño de los vértices a cargar (es decir, VERT_SIZE). Por otro lado, si se selecciona el modo replicado como modo de ejecución del programa sombreador, el código compilado puede hacer que la unidad de sombreador 40 use una segunda fórmula de cálculo de la dirección de la memoria local que calcula la dirección de la memoria local basándose en la primitiva asignada a la fibra de ejecución actual (es decir, *rel_primID*), el tamaño de la primitiva (es decir, PRIM_SIZE), el valor de identificación del vértice (es decir, *rel_vertex*) asociado con el vértice a cargar, y el tamaño del vértice (es decir, VERT_SIZE).

[0145] El segmento de código 92 puede incluir instrucciones que hacen que la unidad de sombreador 40 produzca un vértice o múltiples vértices basándose en el modo de ejecución del programa sombreador seleccionado. Por ejemplo, si se selecciona el modo no replicado como el modo de ejecución del programa sombreador, el código compilado puede hacer que la unidad de sombreador 40 emita múltiples vértices para cada instancia del programa sombreador de geometría fusionado. Por otro lado, si se selecciona el modo replicado como el modo de ejecución del programa sombreador, el código compilado puede hacer que la unidad de sombreador 40 emita un solo vértice para cada instancia del programa sombreador de geometría fusionado. El único vértice que se emite puede corresponder al vértice donde *Gsoutcount==GsoutvertID.*

[0146] Como se muestra en la FIG. 7, los segmentos de código 88 y 90 pueden recibir un parámetro "misc-> reuse", que puede corresponder a la información indicativa de un modo que se utilizará para la ejecución del programa sombreador. De manera similar, el segmento de código 92 puede recibir un parámetro "optimized_mode", que puede corresponder a la información indicativa de un modo que se utilizará para la ejecución del programa sombreador de vértices/geometría fusionado. En algunos ejemplos, el parámetro "misc-> reuse" puede ser generado por la GPU 30, y el parámetro "optimized_mode" puede ser generado por un controlador de GPU, y proporcionado a la GPU 30. En algunos ejemplos, el parámetro "misc-> reuse" y el parámetro "optimized_mode" pueden implementarse como y/o generarse mediante un solo parámetro que es indicativo de un modo que se utilizará para la ejecución del programa sombreador de vértices/geometría fusionado.

[0147] La colocación de instrucciones en el código compilado para el programa sombreador de vértices/geometría fusionado que son capaces de ejecutar selectivamente cualquiera de los modos puede permitir el cambio del modo de procesamiento de la unidad de sombreador 40 sin tener que volver a cargar un nuevo programa sombreador en la unidad de sombreador. Además, la colocación de instrucciones en el código compilado para el programa sombreador de vértices/geometría fusionado que son capaces de ejecutar selectivamente cualquiera de los modos también puede simplificar la compilación del programa sombreador de vértices/geometría fusionado.

[0148] La FIG. 8 es un diagrama de bloques que ilustra un dispositivo informático de ejemplo 100 que puede usarse para implementar las técnicas de ejecución de programa sombreador de esta divulgación. El dispositivo informático 100 puede comprender un ordenador personal, un ordenador de escritorio, un ordenador portátil, una estación de trabajo de ordenador, una plataforma o consola de videojuegos, un dispositivo de comunicación inalámbrica (como, por ejemplo, un teléfono móvil, un teléfono celular, un teléfono satelital, y/o un terminal de teléfono móvil), un teléfono fijo, un teléfono de Internet, un dispositivo de mano como un dispositivo portátil de videojuegos o un asistente digital personal (PDA), un reproductor de música personal, un reproductor de vídeo, un dispositivo de pantalla, un televisión, un descodificador de televisión, un servidor, un dispositivo de red intermedio, un ordenador central o cualquier otro tipo de dispositivo que procesa y/o muestra datos gráficos.

[0149] Como se ilustra en el ejemplo de la FIG. 8, el dispositivo informático 100 incluye una interfaz de usuario 102, una CPU 104, un controlador de memoria 106, una memoria 108, una GPU 30, una interfaz de pantalla 110, una pantalla 112 y un bus 114. La interfaz de usuario 102, la CPU 104, el controlador de memoria 106, la GPU 30 y la interfaz de pantalla 110 pueden comunicarse entre sí utilizando el bus 114. En algunos ejemplos, la GPU 30 puede corresponder a la GPU 30 ilustrada en la FIG. 2. Cabe señalar que la configuración específica de los buses y las interfaces de comunicación entre los diferentes componentes mostrados en la FIG. 8 es meramente a modo de ejemplo, y se pueden usar otras configuraciones de dispositivos informáticos y/u otros sistemas de procesamiento de gráficos con los mismos o diferentes componentes para implementar las técnicas de esta divulgación.

[0150] La CPU 104 puede comprender un procesador de propósito general o de propósito especial que controla el funcionamiento del dispositivo informático 100. Un usuario puede proporcionar entrada al dispositivo informático 100 para hacer que la CPU 104 ejecute una o más aplicaciones de software. Entre las aplicaciones de software que se ejecutan en la CPU 104 pueden incluirse, por ejemplo, una aplicación de gráficos, una aplicación de procesador de

textos, una aplicación de correo electrónico, una aplicación de hoja de cálculo, una aplicación de reproductor de medios, una aplicación de videojuegos, una aplicación de interfaz gráfica de usuario, un sistema operativo, o cualquier otro tipo de programa. El usuario puede proporcionar entrada al dispositivo informático 100 a través de uno o más dispositivos de entrada (no mostrados) como un teclado, un ratón, un micrófono, un panel táctil u otro dispositivo de entrada que esté acoplado al dispositivo informático 100 a través de la interfaz de usuario 102.

[0151] Entre las aplicaciones de software que se ejecutan en la CPU 104 pueden incluirse una o más instrucciones de renderización de gráficos que le indican a la GPU 30 que renderice datos gráficos a una memoria intermedia de tramas para mostrarlos en la pantalla 112. En algunos ejemplos, las instrucciones de renderización gráfica pueden ajustarse a una interfaz de programación de aplicaciones gráficas (API), como, por ejemplo, una API de Open Graphics Library (OpenGL®), una API de Open Graphics Library Embedded Systems (OpenGL ES), una API de Direct3D, una API X3D, una API RenderMan, una API WebGL o cualquier otra API de gráficos estándar pública o patentada. Para procesar las instrucciones de renderización gráfica, la CPU 104 puede emitir uno o más comandos de renderización gráfica a la GPU 30 para que la GPU 30 realice una parte o la totalidad de la renderización de los datos gráficos. En algunos ejemplos, entre los datos gráficos que se van a renderizarse pueden incluirse una lista de primitivas de gráficos, por ejemplo, puntos, líneas, triángulos, cuadriláteros, tiras de triángulos, etc.

[0152] Como se muestra en la FIG. 8, la CPU 104 incluye un controlador de GPU 116 y un compilador 118. El controlador de GPU 116 puede recibir instrucciones de una aplicación de software (por ejemplo, una aplicación de gráficos) y controlar el funcionamiento de la GPU 30 para dar servicio a las instrucciones. Por ejemplo, el controlador de GPU 116 puede formular uno o más comandos, colocar los comandos en la memoria 108 e instruir a la GPU 30 para que ejecute los comandos.

[0153] El compilador 118 puede recibir código fuente para uno o más tipos diferentes de programas sombreadores y generar código fuente compilado para los programas sombreadores. Por ejemplo, el compilador 118 puede recibir código fuente para un programa sombreador de vértices y código fuente para un programa sombreador de geometría, generar código compilado para el programa sombreador de vértices basado en el código fuente para el programa sombreador de vértices, y generar código compilado para el sombreador de geometría basado en el código fuente para el programa sombreador de geometría. El compilador 118 también puede generar un programa sombreador de vértices/geometría fusionado basado en el código compilado para el programa sombreador de vértices y el programa sombreador de geometría. El controlador de GPU 116 puede cargar uno o más de los programas sombreadores compilados en la GPU 30 (por ejemplo, el almacén de instrucciones 44 de la unidad de sombreador 40) para su ejecución por la unidad de sombreador 40 de la GPU 30.

[0154] El controlador de memoria 106 facilita la transferencia de los datos que entran y salen de la memoria 108. Por ejemplo, el controlador de memoria 106 puede recibir comandos de lectura y escritura de memoria, y atender dichos comandos con respecto a la memoria 108 con el fin de proporcionar servicios de memoria para los componentes en el dispositivo informático 100. El controlador de memoria 106 está acoplado comunicativamente a la memoria 108. Aunque el controlador de memoria 106 se ilustra en el dispositivo informático de ejemplo 100 de la FIG. 8 como un módulo de procesamiento que está separado tanto de la CPU 104 como de la memoria 108, en otros ejemplos, parte o toda la funcionalidad del controlador de memoria 106 puede implementarse en una o ambas de CPU 104 y memoria 108.

[0155] La memoria 108 puede almacenar módulos de programa y/o instrucciones que son accesibles para ser ejecutados por la CPU 104 y/o datos para uso por los programas que se ejecutan en la CPU 104. Por ejemplo, la memoria 108 puede almacenar el código del programa y los datos gráficos asociados con las aplicaciones que se ejecutan en la CPU 104. La memoria 108 puede almacenar adicionalmente información para su uso por y/o generada por otros componentes del dispositivo informático 100. Por ejemplo, la memoria 108 puede actuar como una memoria de dispositivo para la GPU 30 y puede almacenar datos para ser operados por la GPU 30, así como datos resultantes de operaciones realizadas por la GPU 30. Por ejemplo, la memoria 108 puede almacenar cualquier combinación de memorias intermedias de textura, memorias intermedias de profundidad, memorias intermedias de plantilla, memorias intermedias de vértice, memorias intermedias de trama, objetivos de renderización, o similares. Además, la memoria 108 puede almacenar secuencias de comandos para su procesamiento por la GPU 30. La memoria 108 puede incluir una o más memorias o dispositivos de almacenamiento volátiles o no volátiles, como, por ejemplo, memoria de acceso aleatorio (RAM), RAM estática (SRAM), RAM dinámica (DRAM), memoria de solo lectura (ROM), ROM programable borrable (EPROM), ROM programable borrable eléctricamente (EEPROM), memoria Flash, un medio de datos magnéticos o un medio de almacenamiento óptico.

[0156] La GPU 30 puede configurarse para ejecutar comandos que son emitidos a la GPU 30 por la CPU 104. Los comandos ejecutados por la GPU 30 pueden incluir comandos de gráficos, comandos de llamadas de extracción, comandos de programación de estado de la GPU, solicitudes de marca de tiempo, comandos de transferencia de memoria, comandos informáticos de propósito general, comandos de ejecución del kernel, etc.

[0157] En algunos ejemplos, la GPU 30 puede configurarse para realizar operaciones gráficas para renderizar una o más primitivas gráficas para mostrar 112. En tales ejemplos, cuando una de las aplicaciones de software que se ejecuta en la CPU 104 requiere procesamiento de gráficos, la CPU 104 puede proporcionar datos gráficos a la GPU

30 y emitir uno o más comandos de gráficos a la GPU 30. Los comandos de gráficos pueden incluir, por ejemplo, comandos de llamadas de extracción, comandos de programación de estado de GPU, comandos de transferencia de memoria, comandos de transferencia, etc. Los datos gráficos pueden incluir memorias intermedias de vértice, datos de textura, datos de superficie, etc. En algunos ejemplos, la CPU 104 puede proporcionar los comandos y los datos gráficos a la GPU 30 escribiendo los comandos y los datos gráficos en la memoria 108, a la que se puede acceder mediante GPU 30.

[0158] En ejemplos adicionales, GPU 30 puede estar configurada para llevar a cabo la computación de propósito general para aplicaciones que se ejecutan en la CPU 104. En tales ejemplos, cuando una de las aplicaciones de software que se ejecutan en la CPU 104 decide descargar una tarea computacional a la GPU 30, la CPU 104 puede proporcionar datos informáticos de propósito general a la GPU 30, y emitir uno o más comandos informáticos de propósito general para la GPU 30. Los comandos informáticos de propósito general pueden incluir, por ejemplo, comandos de ejecución del kernel, comandos de transferencia de memoria, etc. En algunos ejemplos, la CPU 104 puede proporcionar los comandos y los datos informáticos de propósito general a la GPU 30 escribiendo los comandos y los datos en la memoria 108, a la que se puede acceder mediante la GPU 30.

[0159] La GPU 30 puede, en algunos casos, construirse con una estructura altamente paralela que proporciona un procesamiento más eficiente de las operaciones vectoriales que la CPU 104. Por ejemplo, la GPU 30 puede incluir una pluralidad de elementos de procesamiento que están configurados para funcionar en múltiples vértices, puntos de control, píxeles y/u otros datos de manera paralela. La naturaleza altamente paralela de la GPU 30 puede, en algunos casos, permitir que la GPU 30 reproduzca imágenes gráficas (por ejemplo, GUI y escenas gráficas bidimensionales (2D) y/o tridimensionales (3D)) en la pantalla 112 más rápidamente que la renderización de las imágenes utilizando la CPU 104. Además, la naturaleza altamente paralela de la GPU 30 puede permitir que la GPU 30 procese ciertos tipos de operaciones vectoriales y matriciales para aplicaciones informáticas de propósito general más rápidamente que la CPU 104.

[0160] La GPU 30, en algunos casos, puede integrarse en una placa base del dispositivo informático 100. En otros casos, la GPU 30 puede estar presente en una tarjeta gráfica que está instalada en un puerto en la placa base del dispositivo informático 100 o puede incorporarse de otro modo dentro de un dispositivo periférico configurado para interoperar con el dispositivo informático 100. En otros casos, la GPU 30 puede estar ubicada en el mismo microchip que la CPU 104 que forma un sistema en un chip (SoC). La GPU 30 puede incluir uno o más procesadores, como uno o más microprocesadores, circuitos integrados específicos de aplicación (ASIC), matrices de puertas programables en sobre el terreno (FPGA), procesadores de señales digitales (DSP) u otro circuito lógico integrado o discreto equivalente.

[0161] En algunos ejemplos, la GPU 30 puede incluir una memoria caché de GPU, que puede proporcionar servicios de almacenamiento en memoria caché para la totalidad o una parte de memoria 108. En tales ejemplos, la GPU 30 puede usar la memoria caché para procesar datos localmente utilizando un almacenamiento local, en lugar de la memoria fuera del chip. Esto permite que la GPU 30 funcione de una manera más eficiente al reducir la necesidad de que la GPU 30 acceda a la memoria 108 a través del bus 114, que puede experimentar un gran tráfico de bus, durante cada comando de lectura y escritura. Sin embargo, en algunos ejemplos, la GPU 30 puede no incluir una memoria caché separada, sino utilizar en su lugar la memoria 108 a través del bus 114. La memoria caché de GPU puede incluir una o más memorias o dispositivos de almacenamiento volátiles o no volátiles, como, por ejemplo, memoria de acceso aleatorio (RAM), RAM estática (SRAM), RAM dinámica (DRAM), etc.

[0162] La CPU 104 y/o la GPU 30 pueden almacenar datos de imagen de rasterizador en una memoria intermedia de tramas que está asignada dentro de la memoria 108. La interfaz de pantalla 110 puede recuperar los datos de la memoria intermedia de tramas y configurar la pantalla 112 para mostrar la imagen representada por los datos de imagen de rasterizador. En algunos ejemplos, la interfaz de pantalla 110 puede incluir un convertidor digital a analógico (DAC) que está configurado para convertir los valores digitales recuperados de la memoria intermedia de tramas en una señal analógica consumible por la pantalla 112. En otros ejemplos, la interfaz de pantalla 110 puede pasar los valores digitales directamente a la pantalla 112 para su procesamiento.

[0163] La pantalla 112 puede incluir un monitor, un televisor, un dispositivo de proyección, una pantalla de cristal líquido (LCD), un panel de pantalla de plasma, una matriz de diodos emisores de luz (LED), una pantalla de tubo de rayos catódicos (CRT), papel electrónico., una pantalla de emisión de electrones de conducción de superficie (SED), una pantalla de televisión láser, una pantalla de nanocristales u otro tipo de unidad de pantalla. La pantalla 112 puede estar integrada dentro del dispositivo informático 100. Por ejemplo, la pantalla 112 puede ser una pantalla de un teléfono móvil o una tablet. De forma alternativa, la pantalla 112 puede ser un dispositivo independiente acoplado al dispositivo informático 2 a través de un enlace de comunicaciones alámbrico o inalámbrico. Por ejemplo, la pantalla 112 puede ser un monitor de ordenador o una pantalla plana conectada a un ordenador personal a través de un cable o un enlace inalámbrico.

[0164] El bus 114 puede implementarse utilizando cualquier combinación de estructuras de bus y protocolos de bus, incluidas las estructuras y protocolos de bus de primera, segunda y tercera generación, estructuras y protocolos de bus compartido, estructuras y protocolos de bus punto a punto, estructuras y protocolos de bus unidireccional, y

estructuras y protocolos de bus bidireccionales. Los ejemplos de diferentes estructuras y protocolos de bus que se pueden usar para implementar el bus 114 incluyen, por ejemplo, un bus HyperTransport, un bus InfiniBand, un bus Advanced Graphics Port, un bus de interconexión de componentes periféricos (PCI), un bus PCI Express, un bus avanzado de alto rendimiento (AHB) de Arquitectura de bus de microcontrolador avanzado (AMBA), un bus periférico avanzado AMBA (APB) y un bus de interfaz avanzada eXentisible (AXI) AMBA. También se pueden usar otros tipos de estructuras de bus y protocolos.

[0165] De acuerdo con esta divulgación, el dispositivo informático 100 (por ejemplo, CPU 104 y/o GPU 30) puede estar configurada para realizar cualquiera de las técnicas de la ejecución del programa sombreador descritas en esta divulgación. Por ejemplo, la GPU 30 (por ejemplo, la unidad de sombreador 40 de la GPU 30) puede configurarse para ejecutar un programa sombreador que realiza el procesamiento del sombreador de vértices y que genera múltiples vértices de salida para cada vértice de entrada que recibe el programa sombreador de acuerdo con uno o más de las técnicas descritas en esta divulgación. Como otro ejemplo, la GPU 30 puede configurarse para ejecutar un programa sombreador de vértices/geometría fusionado de acuerdo con uno o ambos del modo de ejecución replicado y el modo de ejecución no replicado como se describe en esta divulgación. Como un ejemplo adicional, la GPU 30 puede configurarse para cambiar selectivamente entre usar el modo replicado y el modo no replicado para ejecutar un programa sombreador de vértices/geometría fusionado de acuerdo con una o más de las técnicas descritas en esta divulgación.

[0166] En ejemplos adicionales, la CPU 104, el compilador 118 y/o la GPU 30 pueden configurarse para seleccionar entre el modo no replicado y el modo replicado para ejecutar un programa sombreador de vértices/geometría fusionado y hacer que una unidad de sombreador en la GPU 30 ejecute el programa sombreador de vértices/geometría fusionado de acuerdo con el modo de ejecución del programa sombreador seleccionado de acuerdo con una o más de las técnicas descritas en esta divulgación. En ejemplos adicionales, la CPU 104, el compilador 118 y/o la GPU 30 pueden configurarse para generar código compilado para un programa sombreador de vértices/geometría fusionado de acuerdo con una o más de las técnicas descritas en esta divulgación. Por ejemplo, la CPU 104, el compilador 118 y/o la GPU 30 pueden generar el código compilado de manera que el código compilado incluya instrucciones que hagan que una unidad de sombreador ejecute selectivamente un programa sombreador de vértices/geometría fusionado de acuerdo con el modo no replicado o el modo no replicado basándose en información indicativa de un modo que se utilizará para la ejecución del programa sombreador.

[0167] La FIG. 9 es un diagrama de flujo que ilustra una técnica de ejemplo para ejecutar un programa sombreador de acuerdo con esta divulgación. La CPU 104 (por ejemplo, el controlador de GPU 116) puede cargar un programa sombreador en la GPU 30 (140). El programa sombreador puede realizar el procesamiento de sombreador de vértices y puede generar múltiples vértices de salida para cada vértice de entrada que recibe el programa sombreador. La GPU 30 (por ejemplo, la unidad de sombreador 40) puede ejecutar el programa sombreador que realiza el procesamiento de sombreador de vértices y que genera múltiples vértices de salida para cada vértice de entrada que recibe el programa sombreador (142).

[0168] En algunos ejemplos, la GPU 30 (por ejemplo, la unidad de sombreador 40) puede ejecutar una pluralidad de instancias del programa sombreador de tal manera que cada una de las instancias del programa sombreador recibe una respectiva de una pluralidad de vértices de entrada y genera múltiples vértices de salida en respuesta a recibir el vértice respectivo de la pluralidad de vértices de entrada. En algunos casos, la GPU 30 (por ejemplo, la unidad de sombreador 40) puede ejecutar la pluralidad de instancias del programa sombreador en paralelo y/o como parte de una onda de ejecución. En algunos ejemplos, ejecutar las instancias en paralelo puede incluir ejecutar las instancias de tal manera que cada una de las instancias se ejecute en uno respectivo de una pluralidad de elementos de procesamiento en una unidad de sombreador al mismo tiempo con respecto a diferentes elementos de datos.

[0169] En ejemplos adicionales, el programa sombreador que realiza el procesamiento de sombreador de vértices y que genera múltiples vértices de salida para cada vértice de entrada que es recibido por el programa sombreador puede ser un programa sombreador de vértices/geometría fusionado. El programa sombreador de vértices/geometría fusionado puede ser configurable para realizar el procesamiento de sombreador de vértices y el procesamiento de sombreador de geometría. El procesamiento del sombreador de vértices puede ser especificado por un programa sombreador de vértices y el procesamiento del sombreador de geometría puede ser especificado por un programa sombreador de geometría.

[0170] En algunos ejemplos, la GPU 30 (por ejemplo, la unidad de sombreador 40) puede ejecutar una pluralidad de instancias del programa sombreador de vértices/geometría fusionado de tal manera que cada una de las instancias del programa sombreador de vértices/geometría fusionado genere M vértices de salida, donde M es un número entero mayor o igual a dos. En algunos ejemplos, M puede ser igual a un valor de recuento de vértices de salida máximo que se especifica mediante un programa sombreador de geometría que se implementa mediante el programa sombreador de vértices/geometría fusionado. El valor máximo del recuento de vértices de salida puede ser indicativo de un número máximo de vértices de salida que debe generar el programa sombreador de geometría para cada primitiva que sea procesada por el programa sombreador de geometría. Se puede implementar un programa sombreador de geometría mediante un programa sombreador de vértices/geometría fusionado si el código compilado para el programa sombreador de vértices/geometría fusionado se genera basándose en el programa sombreador de geometría.

[0171] En ejemplos adicionales, cada una de las instancias del programa sombreador de vértices fusionado puede ser configurable para realizar el procesamiento de sombreador de geometría con respecto a una primitiva que se asigna a la instancia respectiva del programa sombreador de vértices fusionado. En tales ejemplos, cada una de una pluralidad de primitivas puede asignarse a una instancia de sombreador de vértices/geometría fusionado para su procesamiento (por ejemplo, una instancia de sombreador de vértices/geometría fusionado por primitiva y/o una instancia de sombreador de vértices/geometría fusionado por onda).

[0172] En algunos ejemplos, cada una de las instancias del programa sombreador de vértices fusionado puede ser configurable para realizar el procesamiento del sombreador de geometría con respecto a una primitiva que se asigna a la instancia respectiva del programa sombreador de vértices fusionado y realizar el procesamiento de sombreador de vértices con respecto a un vértice que se asigna a la instancia respectiva del programa sombreador de vértices fusionado. En tales ejemplos, cada una de una pluralidad de primitivas puede asignarse a una instancia de sombreador de vértices/geometría fusionado para su procesamiento (por ejemplo, una instancia de sombreador de vértices/geometría fusionado por primitiva y/o una instancia de sombreador de vértices/geometría fusionado por onda), y cada uno de una pluralidad de vértices puede asignarse a una instancia de sombreador de vértices/geometría fusionado para su procesamiento (por ejemplo, una instancia de sombreador de vértices/geometría fusionado por vértice y/o una instancia de sombreador de vértices/geometría fusionado por onda). En algunos casos, al menos uno de los casos del programa sombreador de vértices/geometría fusionado puede configurarse para realizar el procesamiento de sombreador de vértices con respecto a uno de la pluralidad de vértices y el procesamiento de sombreador de geometría con respecto a una de la pluralidad de primitivas.

[0173] En algunos ejemplos, la unidad de sombreador 40 puede permitir la reutilización de los vértices que son compartidos entre primitivas. Por ejemplo, la unidad de sombreador 40 puede ejecutar una primera instancia, una segunda instancia y una tercera instancia del programa sombreador de vértices/geometría fusionado. La primera instancia del programa sombreador de vértices/geometría fusionado puede realizar el procesamiento de sombreador de vértices con respecto a uno de una pluralidad de vértices para generar un vértice sombreado de vértice. La segunda instancia del programa sombreador de vértices/geometría fusionado puede realizar el procesamiento del sombreador de geometría con respecto a una primera primitiva de la pluralidad de primitivas basadas en el vértice sombreado de vértice generado por la primera instancia del programa sombreador de vértices/geometría fusionado para generar uno o más vértices sombreados de geometría que corresponden a la primera primitiva. La tercera instancia del programa sombreador de vértices/geometría fusionado puede realizar el procesamiento del sombreador de geometría con respecto a una segunda primitiva de la pluralidad de primitivas basada en el vértice sombreado de vértice generado por la primera instancia del programa sombreador de vértices/geometría fusionado para generar uno o más vértices sombreados de geometría que corresponden a la segunda primitiva.

[0174] La FIG. 10 es un diagrama de flujo que ilustra una técnica de ejemplo para ejecutar un programa sombreador de vértices/geometría fusionado de acuerdo con un modo replicado y un modo no replicado de acuerdo con esta divulgación. En general, la GPU 30 (por ejemplo, una o más de las unidades de sombreador 38) puede configurarse para funcionar en un modo de ejecución de programa sombreador no replicado y configurarse para funcionar en un modo de ejecución de programa sombreador replicado.

[0175] La GPU 30 recibe información indicativa de un modo de ejecución del programa sombreador seleccionado para ser utilizado para ejecutar un programa sombreador de vértices/geometría fusionado (144). La GPU 30 determina si la información indicativa del modo de ejecución del programa sombreador seleccionado indica que el modo replicado se utilizará para ejecutar un programa sombreador de vértices/geometría fusionado (146).

[0176] Si la información indicativa del modo de ejecución del programa sombreador seleccionado indica que el modo replicado no se debe usar para ejecutar un programa sombreador de vértices/geometría fusionado, entonces la GPU 30 puede funcionar en el modo no replicado. Cuando se funciona en el modo no replicado, la unidad de sombreador 40 puede funcionar basándose en una configuración de subprocesos donde cada una de una pluralidad de primitivas se asigna a una instancia de sombreador de vértices/geometría fusionado para el procesamiento de sombreador de geometría (148), cada uno de una pluralidad de vértices se asigna a una instancia de sombreador de vértices/geometría fusionado para el procesamiento (150), y cada una de las instancias del programa sombreador de vértices/geometría fusionado genera M vértices de salida (152). En algunos ejemplos, el programador de subprocesos 36 puede asignar cada una de una pluralidad de primitivas a una instancia de sombreador de vértices/geometría fusionado para el procesamiento de sombreador de geometría (148), y asignar cada uno de una pluralidad de vértices a una instancia de sombreador de vértices/geometría fusionado para procesamiento (150).

[0177] En algunos ejemplos, M puede ser un número entero mayor que o igual a dos. En otros ejemplos, M puede ser igual a la cantidad de vértices que se generan para cada una de las primitivas procesadas por una etapa de sombreador de geometría que se implementa mediante el programa sombreador de vértices/geometría fusionado.

[0178] Si la información indicativa del modo de ejecución del programa sombreador seleccionado indica que el modo replicado se usará para ejecutar un programa sombreador de vértices/geometría fusionado, entonces la GPU 30 puede funcionar en el modo replicado. Cuando se funciona en el modo replicado, la unidad de sombreador 40 puede funcionar

basándose en una configuración de subprocesos donde cada una de una pluralidad de primitivas se asigna a N instancias de sombreador de vértices/geometría fusionado para el procesamiento de sombreador de geometría (154), cada una de una pluralidad de vértices se asigna a K instancias de sombreador de vértices/geometría fusionado para procesamiento (156), y cada una de las instancias del programa sombreador de vértices/geometría fusionado genera un vértice de salida (158). En algunos ejemplos, el programador de subprocesos 36 puede asignar cada una de una pluralidad de primitivas a N instancias de sombreador de vértices/geometría fusionado para el procesamiento de sombreador de geometría (154), y asignar cada uno de una pluralidad de vértices a K instancias de sombreador de vértices/geometría fusionado para procesar (156).

[0179] En algunos ejemplos, N puede ser un número entero mayor que o igual a dos. En otros ejemplos, N puede ser igual a un valor de recuento de vértices de salida máximo que se especifica mediante un programa sombreador de geometría que se implementa mediante el programa sombreador de vértices/geometría fusionado. El valor máximo del recuento de vértices de salida es indicativo de un número máximo de vértices de salida que debe generar el programa sombreador de geometría para cada primitiva procesada por el programa sombreador de geometría. En ejemplos adicionales, M puede ser menor o igual a N. En algunos ejemplos, K puede ser un número entero igual a la cantidad de primitivas que incluyen el vértice respectivo.

[0180] La FIG. 11 es un diagrama de flujo, una técnica de ejemplo para seleccionar uno de un modo replicado y un modo no replicado para ejecutar programas sombreadores de vértices/geometría fusionados de acuerdo con esta divulgación. La CPU 104 y/o el controlador de GPU 116 determinan la información indicativa de una cantidad total de espacio de almacenamiento requerido para almacenar los vértices de salida asociados con un programa sombreador de geometría implementado por el programa sombreador de vértices/geometría fusionado (160). La CPU 104 y/o el controlador de GPU 116 determinan si la información indicativa de la cantidad total de espacio de almacenamiento es mayor que un umbral (162).

[0181] La CPU 104 y/o el controlador de GPU 116 seleccionan el modo de ejecución del programa sombreador no replicado como el modo de ejecución del programa sombreador seleccionado en respuesta a la determinación de que la información indicativa de la cantidad total de espacio de almacenamiento no es mayor que el umbral (164). La CPU 104 y/o el controlador de GPU 116 seleccionan el modo de ejecución del programa sombreador replicado como el modo de ejecución del programa sombreador seleccionado en respuesta a la determinación de que la información indicativa de la cantidad total de espacio de almacenamiento es mayor que el umbral (166).

[0182] En general, la CPU 104 y/o el controlador de GPU 116 pueden seleccionar uno de un modo de ejecución de programa sombreador no replicado y un modo de ejecución de programa sombreador replicado como modo de ejecución de programa sombreador seleccionado para usar para ejecutar el programa sombreador de vértices/geometría fusionado basado en información indicativa de una cantidad total de espacio de almacenamiento requerido para almacenar vértices de salida asociados con un programa sombreador de geometría implementado por el programa sombreador de vértices/geometría fusionado. La CPU 104 y/o el controlador de GPU 116 pueden hacer que la unidad de sombreador 40 de la GPU 30 ejecute el programa sombreador de vértices/geometría fusionado basado en el modo de ejecución del programa sombreador seleccionado.

[0183] En algunos ejemplos, el modo no replicado puede utilizar la funcionalidad de sombreador de geometría que está optimizada para la pequeña amplificación pero puede soportar todas las amplificaciones. La funcionalidad de sombreador de geometría se puede usar con varios tipos diferentes de primitivas que incluyen, por ejemplo, un grupo de puntos (un punto hacia adentro, 4 vértices hacia afuera), un mapa de cubos (triángulo hacia adentro, 6 triángulos hacia fuera, es decir, 18 vértices hacia afuera), un volumen de sombra (triángulo adentro, 15 vértices afuera).

[0184] En algunos ejemplos, el modo no replicado puede soportar una fibra por instancia de sombreador de geometría. En otros ejemplos, el modo no replicado puede, en algunos ejemplos, evitar cálculos de ALU repetidos, evitar asignaciones de recursos repetidos por emisión de vértice (por ejemplo, GPR) y/o evitar la necesidad de implementar un manejo especializado de la vista de acceso no ordenado (UAV). En ejemplos adicionales, el modo no replicado puede soportar la reutilización de vértices para primitivas de entrada que tengan menos de 4 vértices.

[0185] En algunos ejemplos, el modo no replicado puede proporcionar mejor rendimiento que el modo replicado en términos del número de primitivas que pueden procesarse por onda como se ilustra en los siguientes ejemplos. De acuerdo con un primer ejemplo, se puede procesar una pluralidad de primitivas de grupo de puntos. La topología de entrada puede ser una lista de puntos (es decir, PointList), y el recuento máximo de vértices de salida para el sombreador de geometría puede ser igual a cuatro (es decir, MaxGSOutputVertex = 4). En este ejemplo, el modo replicado puede procesar 8 primitivas por onda. Sin embargo, el modo no replicado puede procesar 32 primitivas por onda, proporcionando así una mejora 4 veces mayor en el rendimiento de primitiva por onda.

[0186] De acuerdo con un segundo ejemplo, la topología de entrada es una tira de triángulos (es decir, TriangleStrip), y MaxGSOutputVertex = 4. En este ejemplo, el modo replicado puede procesar 8 primitivas por onda. Sin embargo, el modo no replicado puede procesar 30 primitivas por onda, proporcionando así una mejora 3,75 veces mayor en el rendimiento de primitiva por onda.

[0187] De acuerdo con un tercer ejemplo, la topología de entrada es una lista de triángulos (es decir, TriangleStrip) y MaxGSOutputVertex = 18 (por ejemplo, un mapa de cubos). En este ejemplo, el modo replicado puede procesar 2 primitivas por onda. Por ejemplo, el modo replicado puede realizar 32 emisiones por onda y, por lo tanto, comenzar a procesarlas tan pronto como se haga la onda. Sin embargo, el modo no replicado puede procesar 30 primitivas por onda, proporcionando así una mejora de 3,75 veces en el rendimiento de primitiva. Por ejemplo, el modo no replicado puede realizar 18 emisiones por fibra (es decir, 576 vértices por onda) y, por lo tanto, procesarlas cuando se completa la onda. En algunos casos, es posible que se pueda iniciarse un número limitado de ondas de sombreador de geometría. Sin embargo, un usuario puede ser capaz de programar para limitar el número de primitivas de sombreador de geometría de entrada por onda.

[0188] De acuerdo con un cuarto ejemplo, la topología de entrada es un parche de 10 puntos de control (es decir, Patch_10), y MaxGSOutputVertex = 4. En este ejemplo, el modo replicado puede procesar 3 primitivas por onda, y el modo no replicado puede procesar 3 primitivas por onda (por ejemplo, sin reutilización de vértice). Como tal, puede que no haya una mejora en el rendimiento en términos de rendimiento de primitiva por onda. Sin embargo, el dispositivo no replicado puede proporcionar ahorros de energía porque otros lotes de fibras en onda no pueden ejecutar el sombreador de geometría (es decir, 3 fibras pueden estar ejecutando código de sombreador de geometría).

[0189] En algunos ejemplos, cada fibra de sombreador de vértices/sombreador de geometría (VS | GS) fusionado puede soportar una primitiva dentro, multi-vértice hacia fuera. En algunos ejemplos, una GPU puede generar múltiples ondas de vértices con un indicador de máscara que indica qué fibra tiene vértices válidos (por ejemplo, una onda por emisión). En algunos ejemplos, se puede realizar una comprobación de reutilización de vértice para primitivas de entrada para generar sombreadores de vértice para vértices únicos para primitivas de entrada en una onda VSIGS dada para primitivas con menos de 4 vértices.

[0190] Las técnicas descritas en esta divulgación pueden implementarse, al menos en parte, en hardware, software, firmware o cualquier combinación de los mismos. Por ejemplo, varios aspectos de las técnicas descritas pueden implementarse en uno o más procesadores, incluidos uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados de aplicación específica (ASIC), matrices de puertas programables en campo (FPGA), o en cualquier otro sistema de circuitos de lógica integrada o discreta equivalente, así como en cualquier combinación de dichos componentes. El término "procesador" o "circuitos de procesamiento" puede referirse en general a cualquiera de los circuitos lógicos anteriores, solo o en combinación con otros circuitos lógicos, o cualquier otro circuito equivalente, como el hardware discreto que realiza el procesamiento.

[0191] Tal hardware, software y firmware pueden implementarse dentro del mismo dispositivo o dentro de dispositivos separados para soportar las diversas operaciones y funciones descritas en esta divulgación. Además, cualquiera de las unidades, módulos o componentes descritos pueden implementarse juntos o por separado como dispositivos lógicos discretos pero interoperables. La descripción de diferentes características como módulos o unidades está destinada a resaltar diferentes aspectos funcionales y no implica necesariamente que dichos módulos o unidades deban realizarse mediante componentes de software o hardware separados. Más bien, la funcionalidad asociada con uno o más módulos o unidades puede realizarse mediante hardware, firmware y/o componentes de software separados, o integrarse dentro de componentes de software o hardware comunes o separados.

[0192] Las técnicas descritas en esta divulgación también se pueden almacenar, realizar o codificar en un medio legible por ordenador, tal como un medio de almacenamiento legible por ordenador que almacena instrucciones. Las instrucciones integradas o codificadas en un medio legible por ordenador pueden hacer que uno o más procesadores realicen las técnicas descritas en el presente documento, por ejemplo, cuando las instrucciones son ejecutadas por uno o más procesadores. En algunos ejemplos, el medio legible por ordenador puede ser un medio de almacenamiento no transitorio legible por ordenador. Los medios de almacenamiento legibles por ordenador pueden incluir memoria de acceso aleatorio (RAM), memoria de solo lectura (ROM), memoria de solo lectura programable (PROM), memoria de solo lectura programable y borrable (EPROM), memoria de solo lectura programable y borrable electrónicamente (EEPROM), memoria flash, un disco duro, un CD-ROM, un disquete, un casete, medios magnéticos, medios ópticos u otros medios de almacenamiento legibles por ordenador que sean tangibles.

[0193] Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que correspondan a un medio de almacenamiento tangible, tales como los medios relacionados anteriormente. Los medios legibles por ordenador también pueden comprender medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa de ordenador de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, la frase "medios legibles por ordenador" en general pueden corresponder a (1) medios de almacenamiento legibles por ordenador tangibles que sean no transitorios o (2) un medio de comunicación legible por ordenador no tangible como una señal o una onda portadora transitoria.

[0194] Se han descrito varios aspectos y ejemplos. Sin embargo, se pueden realizar modificaciones a la estructura o técnicas de esta divulgación sin apartarse del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Un procedimiento que comprende:

ejecutar, con una unidad de sombreador (38) de un procesador gráfico (30), un programa sombreador de vértices/geometría fusionado que realiza el procesamiento del sombreador de vértices y el procesamiento del sombreador de geometría, y que genera múltiples vértices de salida para cada vértice de entrada que recibe el programa sombreador en el que la ejecución, con la unidad de sombreador del procesador gráfico, el programa sombreador de vértices/geometría fusionado comprende:

ejecutar, con la unidad de sombreador, una pluralidad de instancias del programa sombreador de vértices/geometría fusionado de tal manera que cada una de las instancias del programa sombreador de vértices/geometría fusionado genere M vértices de salida, donde M es un número entero mayor que o igual a dos;

cada una de las instancias del programa sombreador de vértices/geometría fusionado es configurable para realizar el procesamiento del sombreador de geometría con respecto a una primitiva que se asigna a la instancia respectiva del programa sombreador de vértices/geometría fusionado, en el que se asigna cada una de una pluralidad de primitivas a una instancia de sombreador de vértices/geometría fusionado para su procesamiento; y

cada una de las instancias del programa sombreador de vértices/geometría fusionado se puede configurar para realizar el procesamiento del sombreador de vértices con respecto a un vértice que se asigna a la instancia respectiva del programa sombreador de vértices/geometría fusionado, y en el que la ejecución, con la unidad de sombreador, de la pluralidad de instancias del programa sombreador de vértices/geometría fusionado comprende además:

realizar, con una primera instancia del programa sombreador de vértices/geometría fusionado, el procesamiento de sombreador de vértices con respecto a uno de una pluralidad de vértices para generar un vértice sombreado de vértice;

realizar, con una segunda instancia del programa sombreador de vértices/geometría fusionado, el procesamiento de sombreador de geometría con respecto a una primera primitiva de la pluralidad de primitivas basadas en el vértice sombreado de vértice generado por la primera instancia del programa sombreador de vértices/geometría fusionado para generar uno o más vértices sombreados de geometría que corresponden a la primera primitiva; y

realizar, con una tercera instancia del programa sombreador de vértices/geometría fusionado, el procesamiento del sombreador de geometría con respecto a una segunda primitiva de la pluralidad de primitivas basadas en el vértice sombreado de vértice generado por la primera instancia del programa sombreador de vértices/geometría fusionado para generar uno o más vértices sombreados de geometría que corresponden a la segunda primitiva.

2. El procedimiento según la reivindicación 1, en el que ejecutar, con la unidad de sombreador del procesador de gráficos, el programa sombreador de vértices/geometría fusionado comprende:

ejecutar, con la unidad de sombreador, una pluralidad de instancias del programa sombreador de vértices/geometría fusionado de tal manera que cada una de las instancias del programa sombreador de vértices/geometría fusionado recibe uno respectivo de una pluralidad de vértices de entrada y genera múltiples vértices de salida en respuesta a recibir el respectivo de la pluralidad de vértices de entrada.

3. El procedimiento según la reivindicación 1, en el que M es igual a un valor de recuento de vértices de salida máximo que se especifica mediante el programa sombreador de geometría que se implementa mediante el programa sombreador de vértices/geometría fusionado, siendo el valor de recuento de vértices de salida máximo un indicador de un número máximo de vértices de salida que debe generar el programa sombreador de geometría para cada primitiva procesada por el programa sombreador de geometría.

4. El procedimiento según la reivindicación 1,

en el que cada una de las instancias del programa sombreador de vértices/geometría fusionado se puede configurar para realizar el procesamiento del sombreador de vértices con respecto a un vértice que se asigna a la instancia respectiva del programa sombreador de vértices/geometría fusionado,

en el que cada uno de una pluralidad de vértices se asigna a una instancia de sombreador de vértices/geometría fusionado para su procesamiento, y

en el que al menos uno de los casos del programa sombreador de vértices/geometría fusionado está configurado para realizar el procesamiento de sombreador de vértices con respecto a uno de la pluralidad de vértices y el procesamiento del sombreador de geometría con respecto a una de la pluralidad de primitivas.

5. Un dispositivo que comprende:

una unidad de procesamiento de gráficos (GPU, 30) que comprende una unidad de sombreador (38) configurada para ejecutar un programa sombreador de vértices/geometría fusionado que realiza el procesamiento de sombreador de vértices y el procesamiento de sombreador de geometría, y que genera múltiples vértices de salida para cada vértice de entrada que recibe el programa sombreador, en el que ejecutar, con la unidad de sombreador de la unidad de procesamiento de gráficos, el programa sombreador de vértices/geometría fusionado comprende:

ejecutar, con la unidad de sombreador, una pluralidad de instancias del programa sombreador de vértices/geometría fusionado de tal manera que cada una de las instancias del programa sombreador de vértices/geometría fusionado genere M vértices de salida, donde M es un número entero mayor que o igual a dos;

cada una de las instancias del programa sombreador de vértices/geometría fusionado es configurable para realizar el procesamiento del sombreador de geometría con respecto a una primitiva que se asigna a la instancia respectiva del programa sombreador de vértices/geometría fusionado, en el que se asigna cada una de una pluralidad de primitivas a una instancia de sombreador de vértices/geometría fusionado para su procesamiento; y

cada una de las instancias del programa sombreador de vértices/geometría fusionado se puede configurar para realizar el procesamiento del sombreador de vértices con respecto a un vértice que se asigna a la instancia respectiva del programa sombreador de vértices/geometría fusionado, y en el que la ejecución, con la unidad de sombreador, de la pluralidad de instancias del programa sombreador de vértices/geometría fusionado comprende además:

realizar, con una primera instancia del programa sombreador de vértices/geometría fusionado, el procesamiento de sombreador de vértices con respecto a uno de una pluralidad de vértices para generar un vértice sombreado de vértice;

realizar, con una segunda instancia del programa sombreador de vértices/geometría fusionado, el procesamiento de sombreador de geometría con respecto a una primera primitiva de la pluralidad de primitivas basadas en el vértice sombreado de vértice generado por la primera instancia del programa sombreador de vértices/geometría fusionado para generar uno o más vértices sombreados de geometría que corresponden a la primera primitiva; y

realizar, con una tercera instancia del programa sombreador de vértices/geometría fusionado, el procesamiento del sombreador de geometría con respecto a una segunda primitiva de la pluralidad de primitivas basadas en el vértice sombreado de vértice generado por la primera instancia del programa sombreador de vértices/geometría fusionado para generar uno o más vértices sombreados de geometría que corresponden a la segunda primitiva.

6. El dispositivo de la reivindicación 7, en el que la unidad de sombreador está configurada también para:

ejecutar una pluralidad de instancias del programa sombreador de vértices/geometría fusionado de tal manera que cada una de las instancias del programa sombreador de vértices/geometría fusionado reciba uno respectivo de una pluralidad de vértices de entrada y genere múltiples vértices de salida en respuesta a la recepción del respectivo vértice de la pluralidad de vértices de entrada.

7. Un medio de almacenamiento legible por ordenador no transitorio que almacena instrucciones que, tras su ejecución mediante uno o más procesadores, hacen que el uno o más procesadores realicen el procedimiento de una cualquiera de las reivindicaciones 1 a 4.

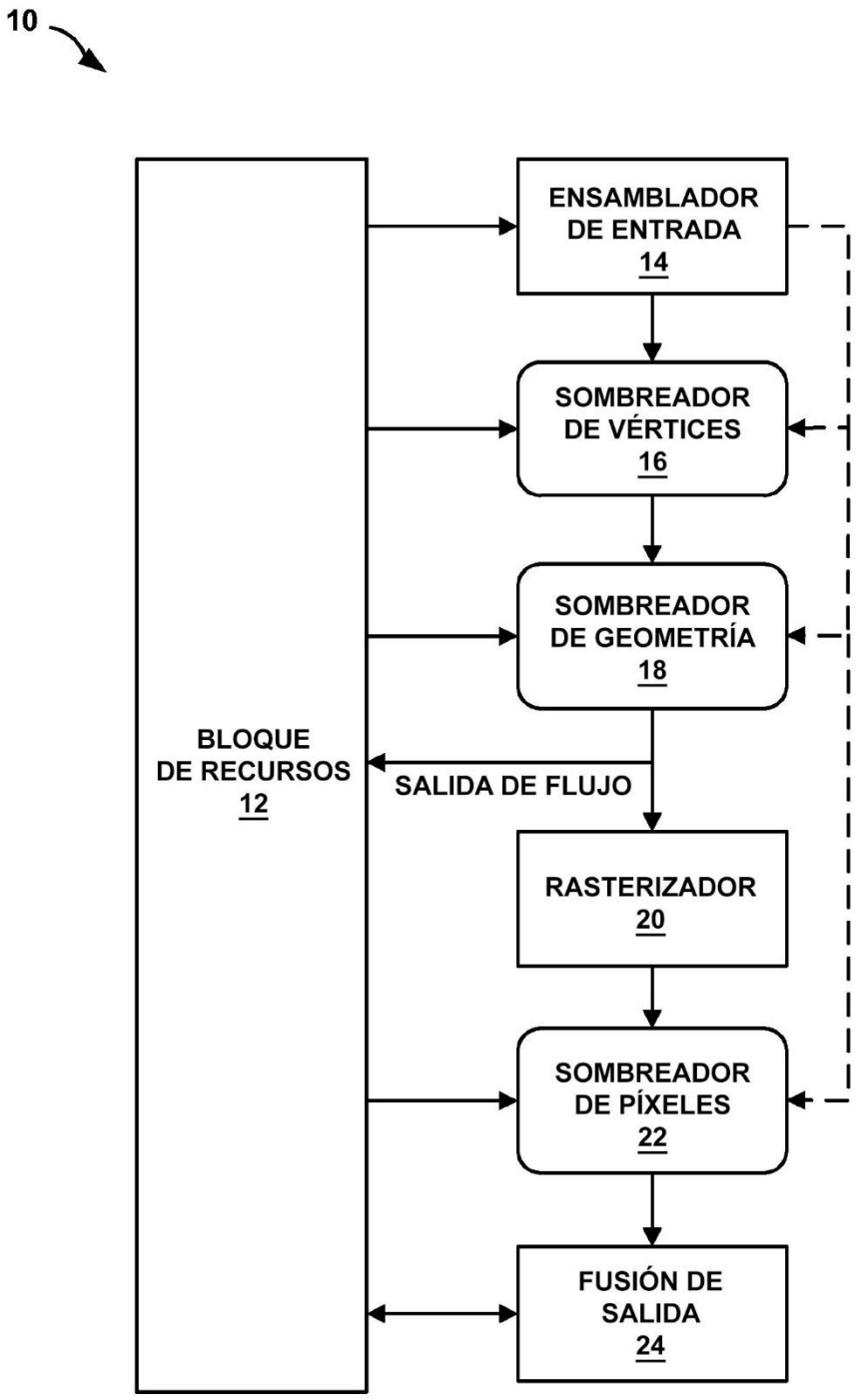
10
ENSAMBLADOR DE ENTRADA 14
SOMBREADOR DE VÉRTICES 16
SOMBREADOR DE GEOMETRÍA 18
BLOQUE DE RECURSOS 12
SALIDA DE FLUJO
RASTERIZADOR 20
SOMBREADOR DE PÍXELES 22
FUSIÓN DE SALIDA 24

FIG. 1

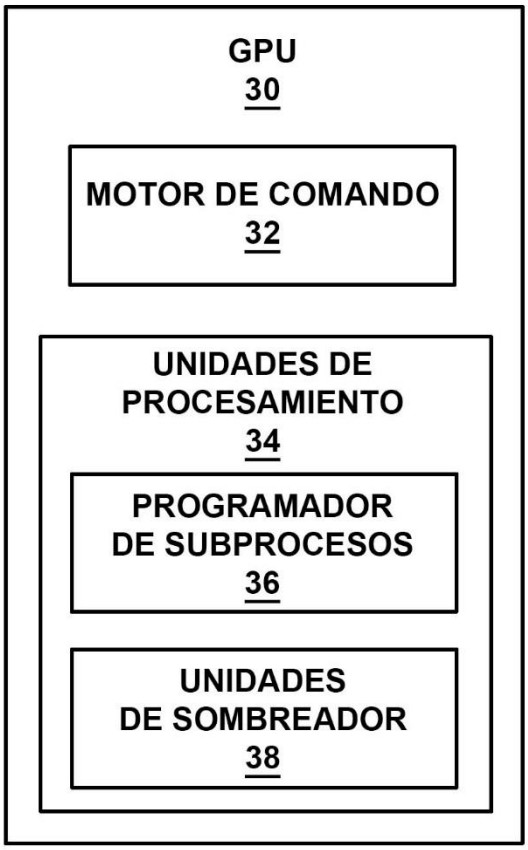
GPU
30
MOTOR DE COMANDO
32
UNIDADES DE
PROCESAMIENTO
34
PROGRAMADOR
DE SUBPROCESOS
36
UNIDADES
DE SOMBREADOR
38

FIG. 2

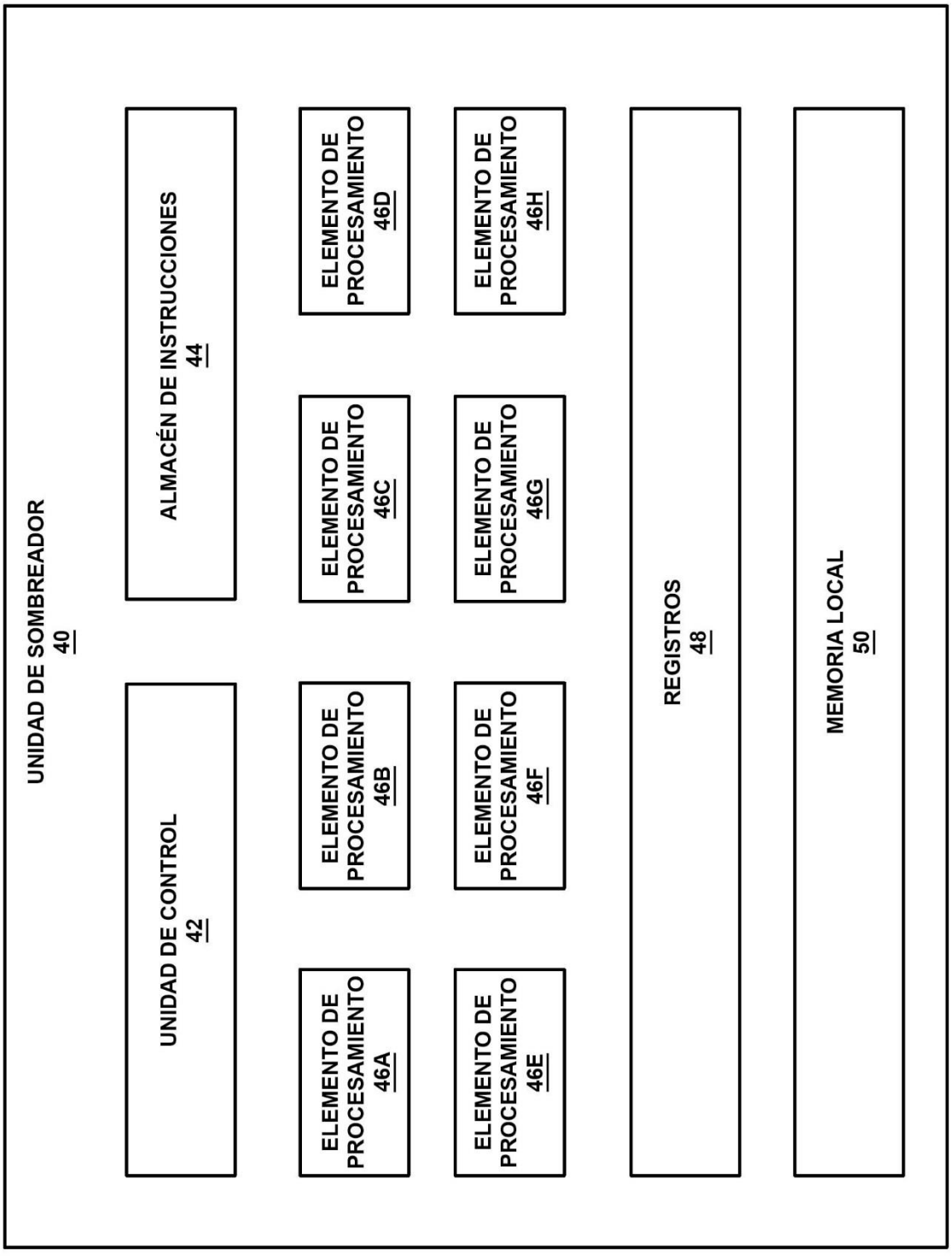
UNIDAD DE SOMBREADOR
40
UNIDAD DE CONTROL
42
ALMACÉN DE INSTRUCCIONES
44
ELEMENTO DE PROCESAMIENTO
46A
ELEMENTO DE PROCESAMIENTO
46B
ELEMENTO DE PROCESAMIENTO
46C
ELEMENTO DE PROCESAMIENTO
46D
ELEMENTO DE PROCESAMIENTO
46E
ELEMENTO DE PROCESAMIENTO
46F
ELEMENTO DE PROCESAMIENTO
46G
ELEMENTO DE PROCESAMIENTO
46H
REGISTROS
48
MEMORIA LOCAL
50

FIG. 3

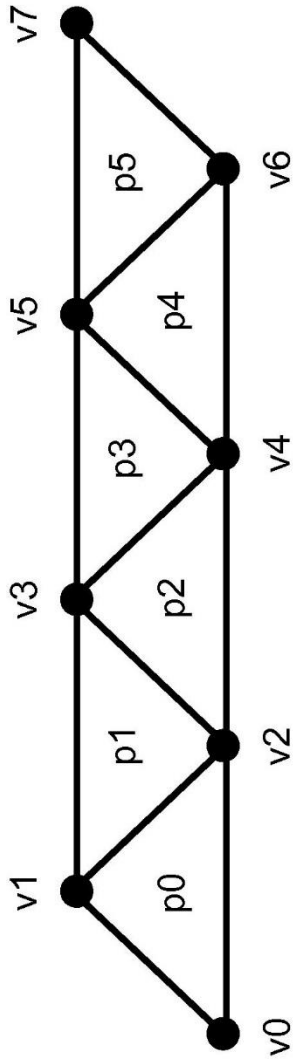

FIG. 4

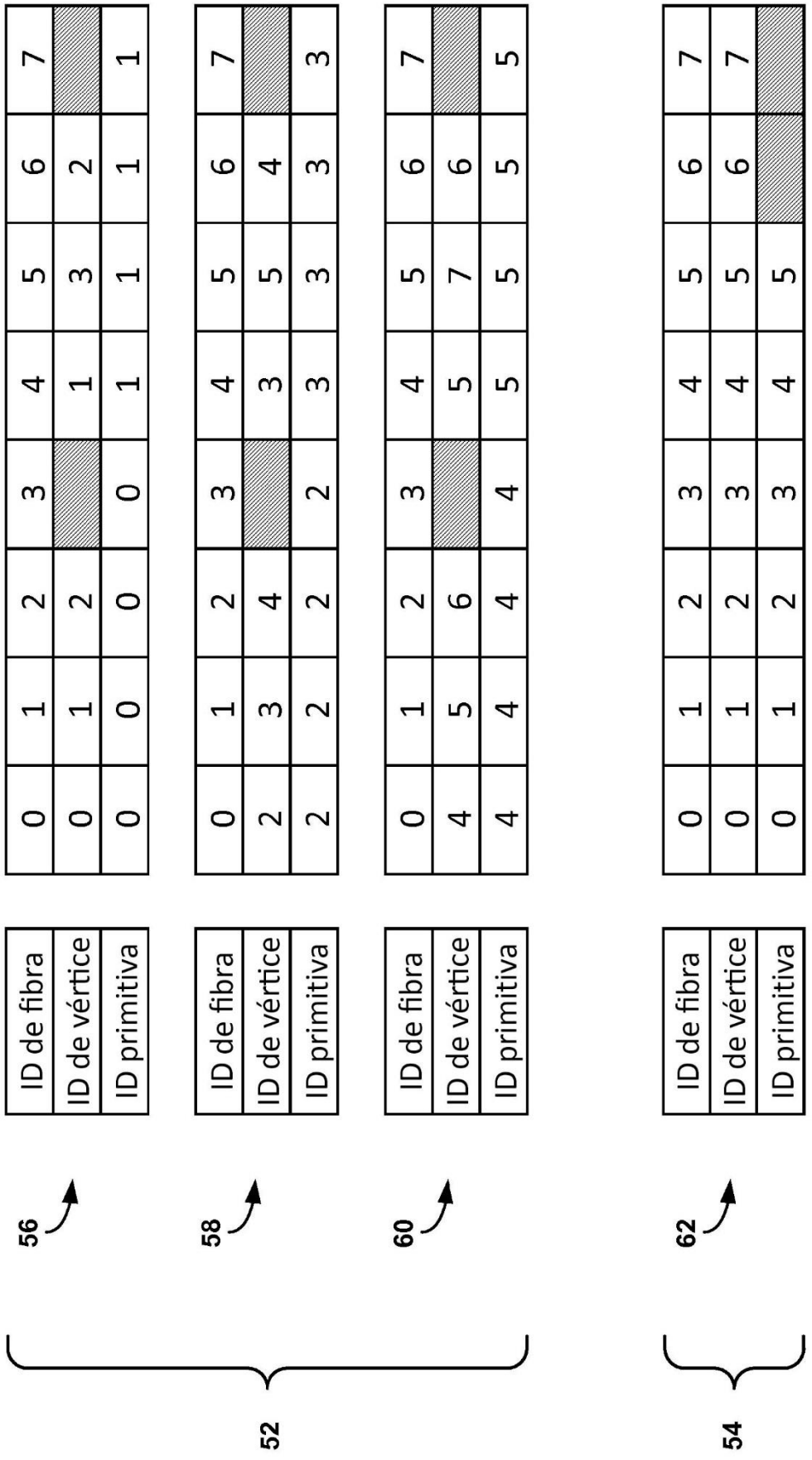
ID de fibra
ID de vértice
ID primitiva
0 1 2 3 4 5 6 7
0 1 2 1 3 2
0 0 0 0 1 1 1 1
56
58
60
62
52
54

FIG. 5

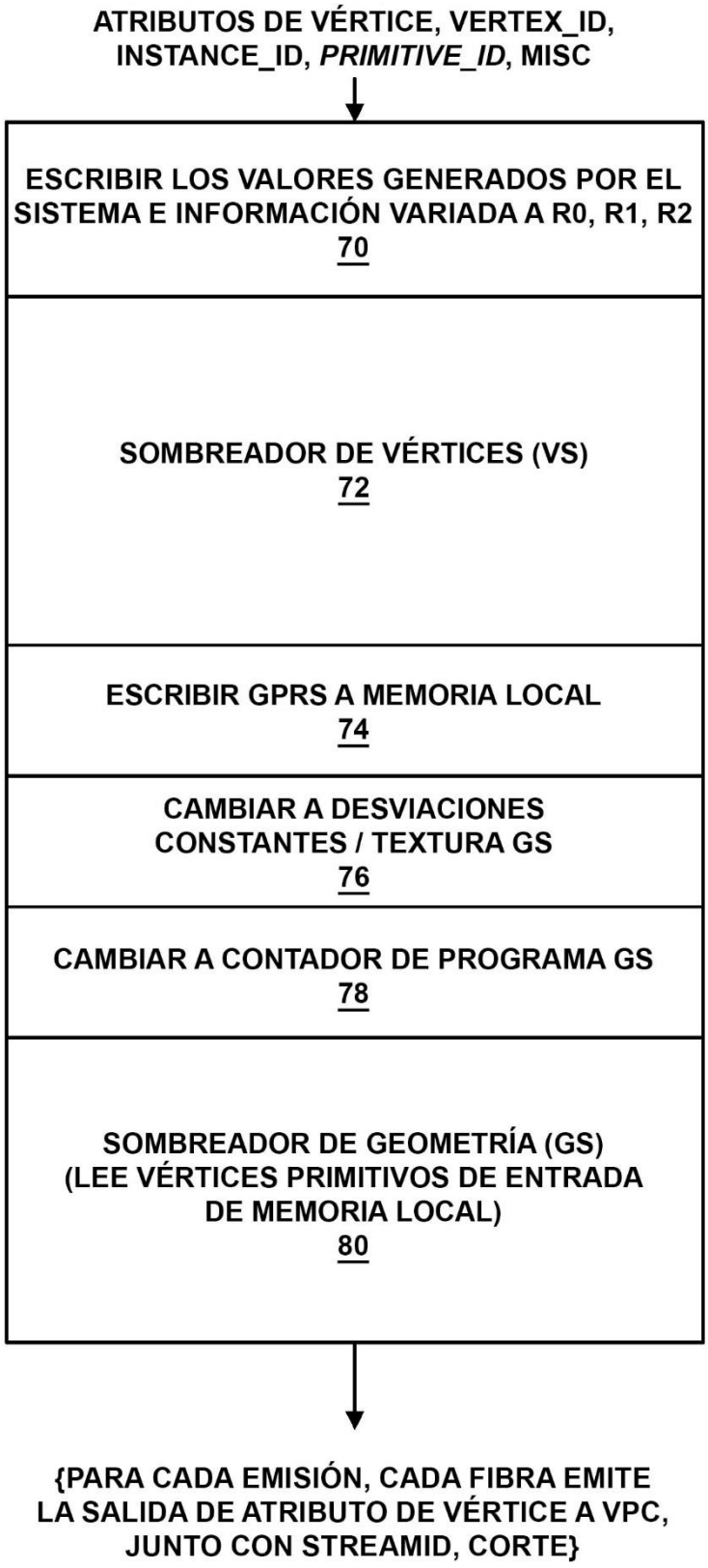
ATRIBUTOS DE VÉRTICE, VERTEX_ID, INSTANCE_ID, *PRIMITIVE_ID*, MISC
ESCRIBIR LOS VALORES GENERADOS POR EL SISTEMA E INFORMACIÓN VARIADA A R0, R1, R2
70
SOMBREADOR DE VÉRTICES (VS)
72
ESCRIBIR GPRS A MEMORIA LOCAL
74
CAMBIAR A DESVIACIONES CONSTANTES / TEXTURA GS
76
CAMBIAR A CONTADOR DE PROGRAMA GS
78
SOMBREADOR DE GEOMETRÍA (GS) (LEE VÉRTICES PRIMITIVOS DE ENTRADA DE MEMORIA LOCAL)
80
{PARA CADA EMISIÓN, CADA FIBRA EMITE LA SALIDA DE ATRIBUTO DE VÉRTICE A VPC, JUNTO CON STREAMID, CORTE}

FIG. 6

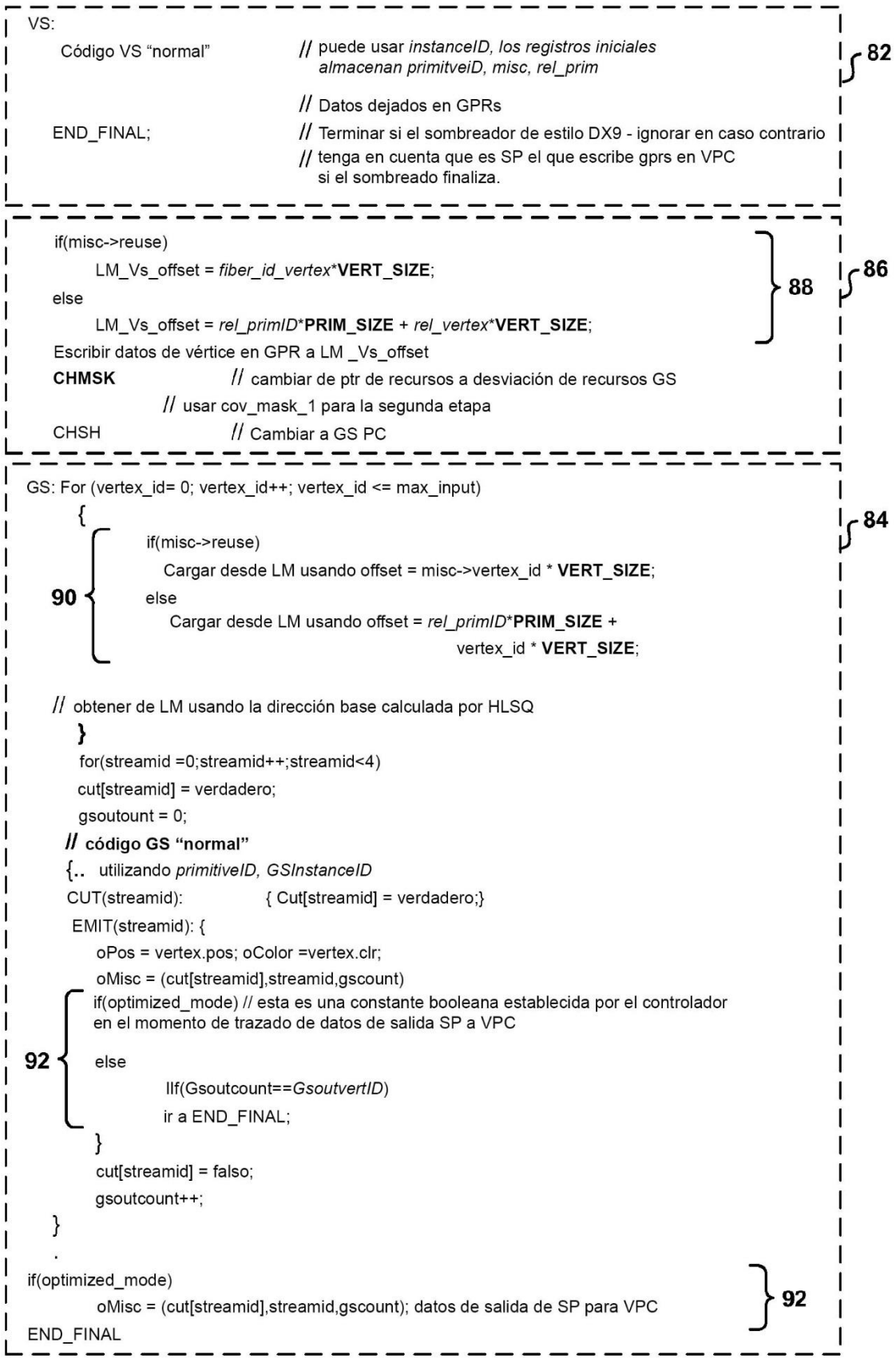
VS:
Código VS "normal" // puede usar instanceID, los registros iniciales almacenan primitveiD, misc, rel_prim
// Datos dejados en GPRs
END_FINAL; // Terminar si el sombreador de estilo DX9 - ignorar en caso contrario
// tenga en cuenta que es SP el que escribe gprs en VPC si el sombreado finaliza.
82
if(misc->reuse)
LM_Vs_offset = fiber_id_vertex*VERT_SIZE;
else
LM_Vs_offset = rel_primID*PRIM_SIZE + rel_vertex*VERT_SIZE;
88
86
Escribir datos de vértice en GPR a LM _Vs_offset
CHMSK // cambiar de ptr de recursos a desviación de recursos GS
// usar cov_mask_1 para la segunda etapa
CHSH // Cambiar a GS PC
GS: For (vertex_id= 0; vertex_id++; vertex_id <= max_input)
{
84
90
if(misc->reuse)
Cargar desde LM usando offset = misc->vertex_id * VERT_SIZE;
else
Cargar desde LM usando offset = rel_primID*PRIM_SIZE + vertex_id * VERT_SIZE;
// obtener de LM usando la dirección base calculada por HLSQ
}
for(streamid =0;streamid++;streamid<4)
cut[streamid] = verdadero;
gsoutount = 0;
// código GS "normal"
{.. utilizando primitiveID, GSInstanceID
CUT(streamid): { Cut[streamid] = verdadero;}
EMIT(streamid): {
oPos = vertex.pos; oColor =vertex.clr;
oMisc = (cut[streamid],streamid,gscount)
92
if(optimized_mode) // esta es una constante booleana establecida por el controlador en el momento de trazado de datos de salida SP a VPC
else
IIf(Gsoutcount==GsoutvertID)
ir a END_FINAL;
}
cut[streamid] = falso;
gsoutcount++;
}
.
if(optimized_mode)
oMisc = (cut[streamid],streamid,gscount); datos de salida de SP para VPC
92
END_FINAL

FIG. 7

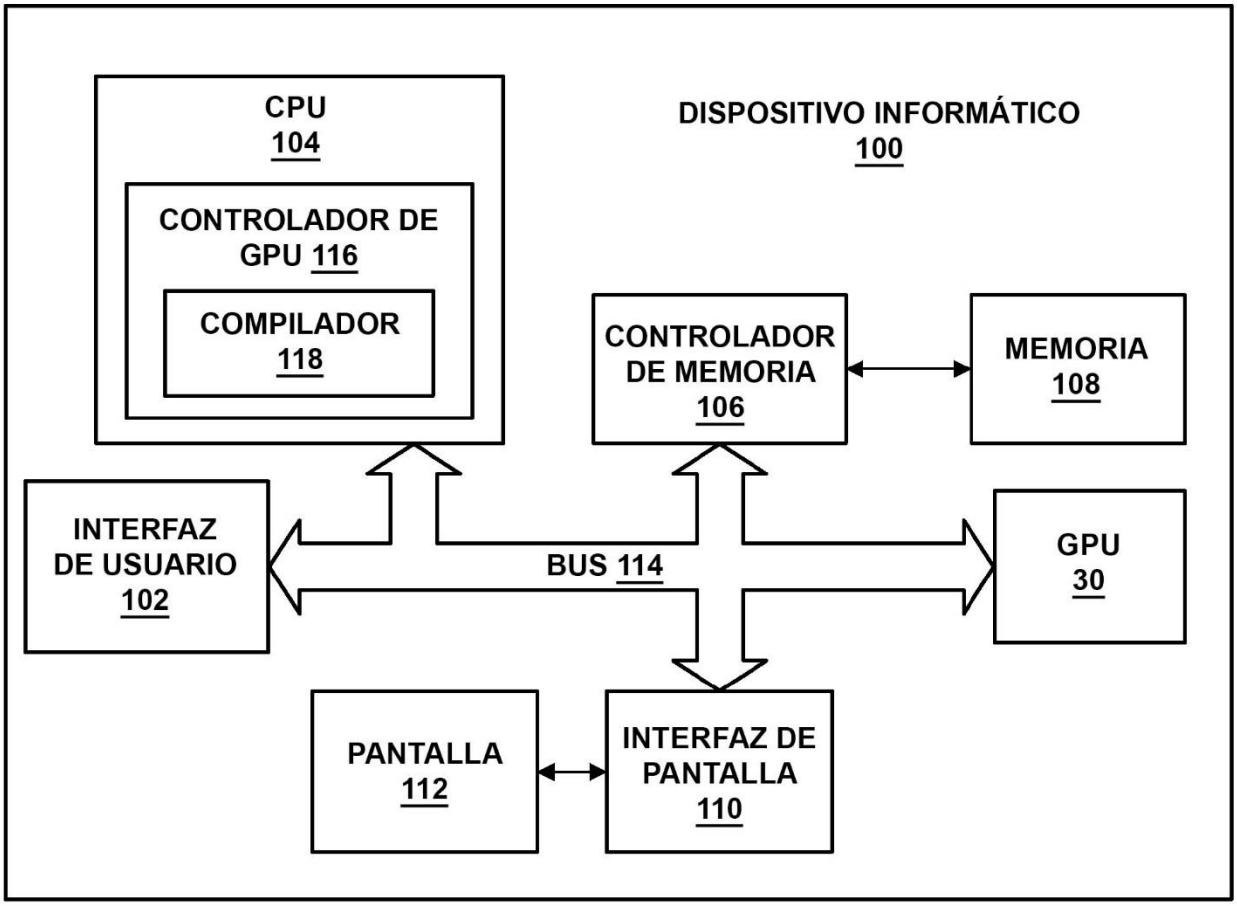
CPU
104
CONTROLADOR DE GPU 116
COMPILADOR
118
DISPOSITIVO INFORMÁTICO
100
CONTROLADOR DE MEMORIA
106
MEMORIA
108
INTERFAZ DE USUARIO
102
BUS 114
GPU
30
PANTALLA
112
INTERFAZ DE PANTALLA
110

FIG. 8

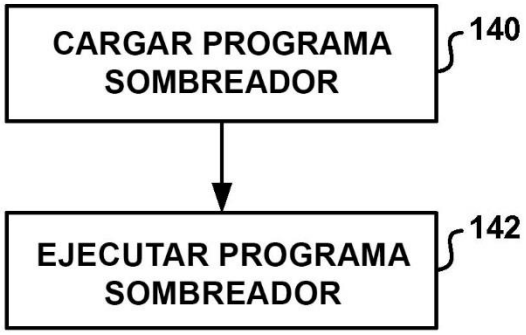
CARGAR PROGRAMA SOMBREADOR
140
EJECUTAR PROGRAMA SOMBREADOR
142

FIG. 9

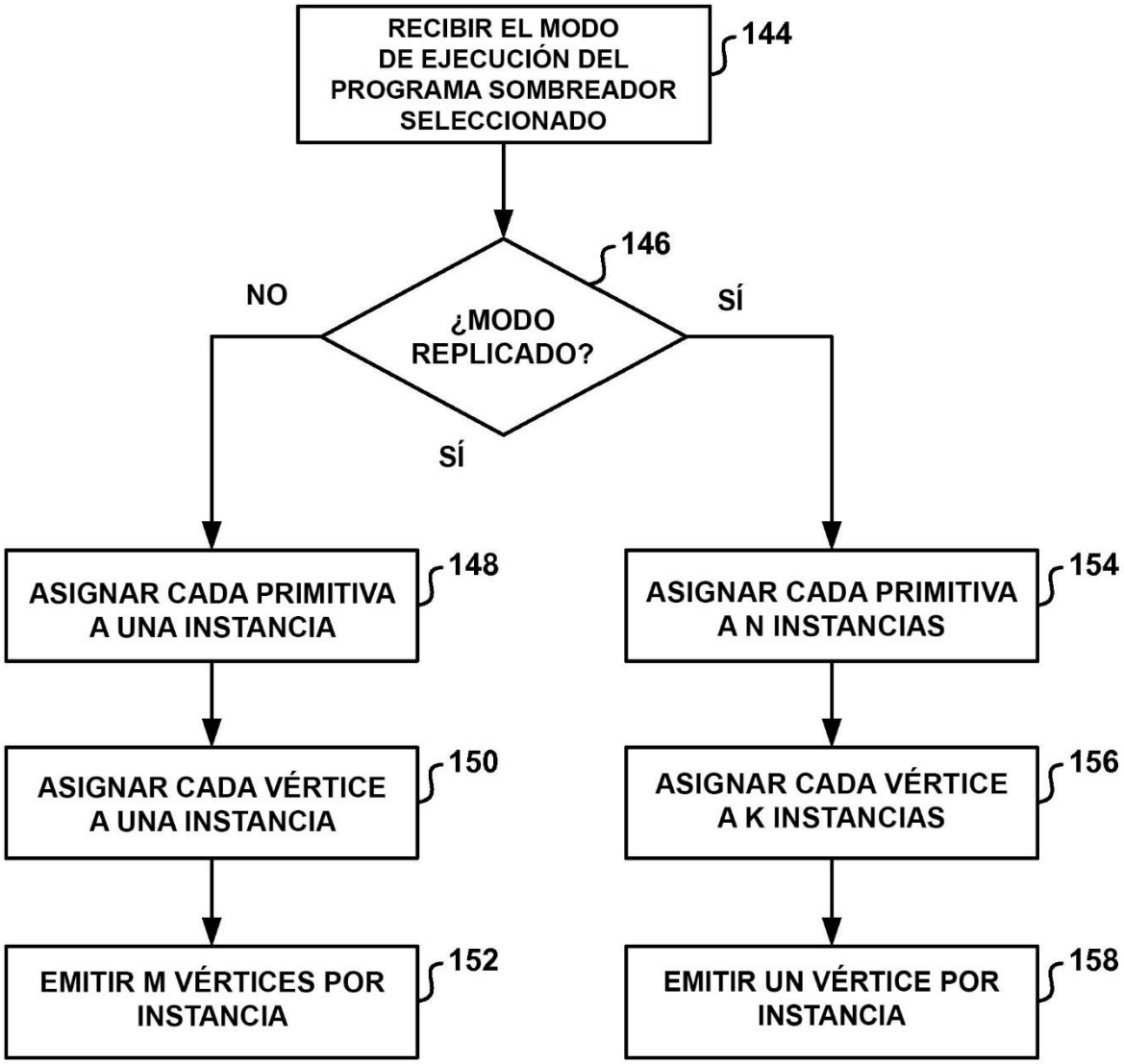
RECIBIR EL MODO DE EJECUCIÓN DEL PROGRAMA SOMBREADOR SELECCIONADO
144
¿MODO REPLICADO?
146
NO
SÍ
SÍ
ASIGNAR CADA PRIMITIVA A UNA INSTANCIA
148
ASIGNAR CADA VÉRTICE A UNA INSTANCIA
150
EMITIR M VÉRTICES POR INSTANCIA
152
ASIGNAR CADA PRIMITIVA A N INSTANCIAS
154
ASIGNAR CADA VÉRTICE A K INSTANCIAS
156
EMITIR UN VÉRTICE POR INSTANCIA
158

FIG. 10

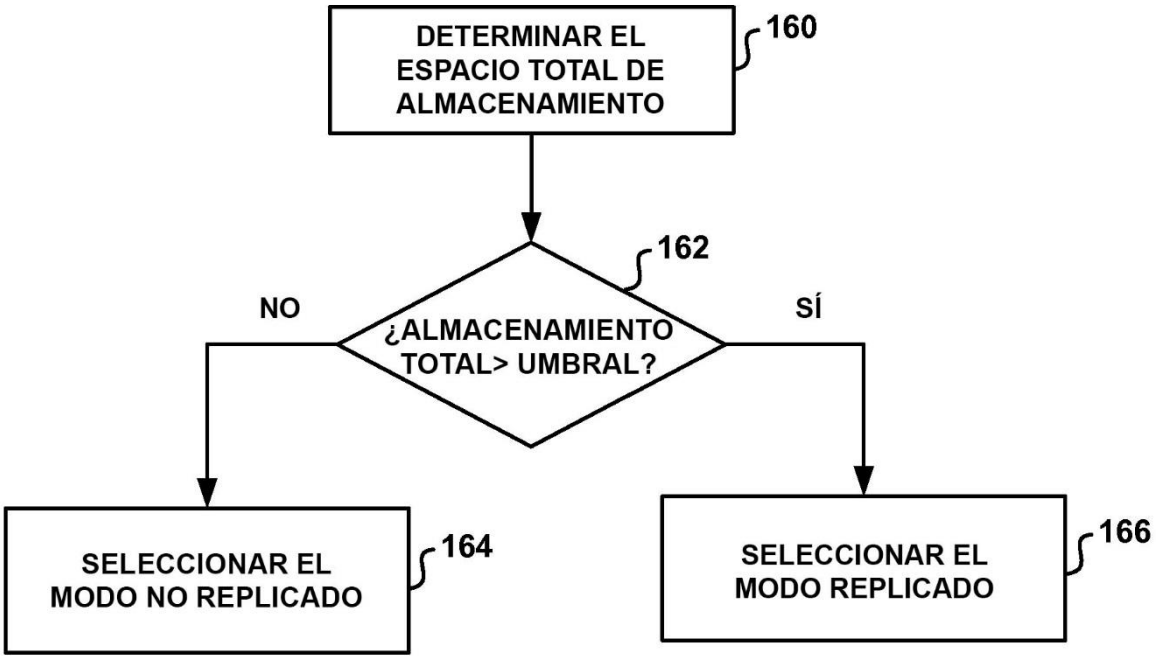
DETERMINAR EL ESPACIO TOTAL DE ALMACENAMIENTO
160
162
¿ALMACENAMIENTO TOTAL> UMBRAL?
NO
SÍ
SELECCIONAR EL MODO NO REPLICADO
164
SELECCIONAR EL MODO REPLICADO
166

FIG. 11