

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 614 716**

51 Int. Cl.:

G06T 15/00 (2011.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **20.11.2012 PCT/US2012/066079**

87 Fecha y número de publicación internacional: **20.06.2013 WO13089989**

96 Fecha de presentación y número de la solicitud europea: **20.11.2012 E 12805832 (8)**

97 Fecha y número de publicación de la concesión europea: **09.11.2016 EP 2791910**

54 Título: **Unidad de procesamiento de gráficos con procesador de comandos**

30 Prioridad:

15.12.2011 US 201113327069

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

01.06.2017

73 Titular/es:

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

72 Inventor/es:

**NORLUND, PETRI OLAVI;
ARVO, JUKKA-PEKKA y
SIMPSON, ROBERT J.**

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 614 716 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Unidad de procesamiento de gráficos con procesador de comandos

5 **CAMPO TÉCNICO**

Los aspectos de la divulgación se refieren al procesamiento de gráficos.

10 **ANTECEDENTES**

15 Las unidades de procesamiento de gráficos (GPU) se implementan generalmente para representar contenido de gráficos tridimensionales (3D) para su presentación en una pantalla visual bidimensional. Por ejemplo, una GPU puede implementarse para representar píxeles que son representativos de contenido tridimensional en una pantalla. La GPU puede realizar el procesamiento de gráficos para generar valores de píxel para cada píxel de la pantalla cuando se representa una escena.

20 Algunas GPU pueden representar una escena completa de una sola vez. Alternativamente, una GPU puede representar gráficos en partes más pequeñas de una escena, a menudo denominadas "placas" de una escena. Por ejemplo, una GPU puede subdividir una escena en una pluralidad de placas, y representar individualmente cada una de las placas. La GPU puede a continuación reconstruir la escena mediante la combinación de cada una de las placas representadas. Las placas pueden ser de un tamaño tal que los datos asociados a una placa en particular puedan ser almacenados en los recursos locales de memoria de la GPU durante la representación. El nuevo artículo, titulado "Arquitectura de Cálculo CUDA de la Próxima Generación de NVIDIA", XP002693218, proporciona una visión general del cálculo de unidades de procesamiento de gráficos.

25 **RESUMEN**

30 En general, esta divulgación describe técnicas para el control de una unidad de procesamiento de gráficos (GPU). Más específicamente, esta divulgación describe técnicas para la gestión de recursos de una GPU usando una unidad de procesamiento integrada, según las reivindicaciones.

35 Es decir, por ejemplo, en lugar de recibir un flujo de comandos desde una CPU que dicta cómo se van a utilizar los recursos de una GPU, la unidad de procesamiento integrada de la GPU puede recibir tareas de cálculo que, en general, definen el trabajo que debe ser realizado por la GPU. La unidad de procesamiento integrada de la GPU puede a continuación determinar cómo son ejecutadas las tareas por los recursos de la GPU. Por ejemplo, la unidad de procesamiento integrada de la GPU puede recibir una tarea y planificar de forma independiente la tarea para un recurso de GPU específico, controlando de ese modo el recurso de GPU que ejecuta la tarea.

40 En un ejemplo, una unidad de procesamiento de gráficos comprende un procesador sombreador, uno o más unidades de hardware de función fija y un procesador de comandos. El procesador sombreador está configurado para ejecutar una pluralidad de instrucciones en paralelo. Las una o más unidades de hardware de función fija están configuradas para representar datos de gráficos. La unidad procesadora de comandos está configurada para recibir una o más tareas desde un procesador principal y planificar de forma independiente las una o más tareas a ejecutar selectivamente mediante el procesador sombreador y las una o más unidades de hardware de función fija.

45 En otro ejemplo, un procedimiento para controlar una unidad de procesamiento de gráficos comprende la recepción de una o más tareas desde un procesador anfitrión, y la planificación, independientemente del procesador anfitrión, de las una o más tareas a ejecutar selectivamente mediante un procesador sombreador y una o más unidades de hardware de función fija, en el que el procesador sombreador está configurado para ejecutar una pluralidad de instrucciones en paralelo, y las una o más unidades de hardware de función fija están configuradas para representar datos de gráficos.

50 En otro ejemplo, un aparato para controlar una unidad de procesamiento de gráficos comprende medios para la recepción de una o más tareas desde un procesador principal, y medios para la planificación, independientemente del procesador anfitrión, de las una o más tareas a ejecutar selectivamente mediante un procesador sombreador y una o más unidades de hardware de función fija, en el que el procesador sombreador está configurado para ejecutar una pluralidad de instrucciones en paralelo, y las una o más unidades de hardware de función fija están configuradas para representar datos de gráficos.

55 En otro ejemplo, un producto de programa informático que comprende un medio legible por ordenador que comprende instrucciones almacenadas en el mismo que, al ejecutarse, hacen que uno o más procesadores reciban una o más tareas desde un procesador anfitrión y planifiquen, de manera independiente del procesador anfitrión, las una o más tareas a ejecutar selectivamente mediante un procesador sombreador y una o más unidades de hardware de función fija, en el que el procesador sombreador está configurado para ejecutar una pluralidad de instrucciones en paralelo, y las una o más unidades de hardware de función fija están configuradas para representar datos de gráficos.

Los detalles de uno o más aspectos de la divulgación se exponen en los dibujos adjuntos y la descripción a continuación. Otras características, objetos y ventajas de la divulgación serán evidentes a partir de la descripción y los dibujos, y a partir de las reivindicaciones.

5 **BREVE DESCRIPCIÓN DE LOS DIBUJOS**

La FIG. 1 es un diagrama de bloques que ilustra un dispositivo informático que tiene una unidad de procesamiento de gráficos que puede implementar las técnicas de esta divulgación.

10 La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de canal de representación de gráficos.

La FIG. 3 es un diagrama de bloques que ilustra la unidad de procesamiento de gráficos del ejemplo mostrado en la FIG. 1 en mayor detalle.

15 La FIG. 4 es un diagrama de flujo que ilustra un funcionamiento ejemplar de un procesador de comandos, según aspectos de la divulgación.

20 La FIG. 5 es un diagrama de flujo que ilustra otro funcionamiento ejemplar de un procesador de comandos, según aspectos de la divulgación.

DESCRIPCIÓN DETALLADA

En general, esta divulgación describe técnicas para el control de una unidad de procesamiento de gráficos (GPU). Más específicamente, esta divulgación describe técnicas para la gestión de recursos de una GPU usando una unidad de procesamiento integrada. Es decir, por ejemplo, en lugar de recibir un flujo de comandos desde una CPU que dicta cómo se van a utilizar los recursos de una GPU, la unidad de procesamiento integrada de la GPU puede recibir tareas de cálculo que, en general, definen el trabajo que debe ser realizado por la GPU. La unidad de procesamiento integrada de la GPU puede a continuación determinar cómo son ejecutadas las tareas por los recursos de la GPU. Por ejemplo, la unidad de procesamiento integrada de la GPU puede recibir una tarea y planificar de forma independiente la tarea para un recurso específico de la GPU, controlando de ese modo el recurso de la GPU que ejecuta la tarea.

La FIG. 1 es un diagrama de bloques que ilustra un dispositivo informático 20 que puede implementar técnicas para la gestión de recursos de una GPU, según aspectos de la divulgación. Como se muestra en la FIG. 1, el dispositivo informático 20 incluye un procesador anfitrión 24, un dispositivo de almacenamiento 28, una memoria 32, un módulo de red 36, una interfaz de usuario 40 y una pantalla 44. El dispositivo informático 20 también incluye una unidad de procesamiento de gráficos (GPU) 48 que tiene uno o más procesadores sombreadores 52, una o más unidades de función fija 54 y un procesador de comandos 56. El dispositivo informático 20 puede, en algunos ejemplos, incluir o ser parte de un dispositivo informático portátil o móvil (por ejemplo, teléfono móvil, ordenador plegable, ordenador portátil, tableta, reproductor de medios digitales, dispositivo de juegos u otro dispositivo informático portátil). Como alternativa, el dispositivo informático 20 puede configurarse como un dispositivo informático fijo, tal como un ordenador de escritorio, un servidor, un televisor, un decodificador de televisión, una consola de juegos o similares.

El dispositivo informático 20 de la FIG. 1 es simplemente un ejemplo. Las técnicas para la gestión de recursos de una GPU, como la GPU 48, pueden llevarse a cabo mediante otros múltiples dispositivos informáticos que tienen otros componentes. En algunos ejemplos, el dispositivo informático 20 puede incluir componentes adicionales que no se muestran en la FIG. 1, con fines de claridad. Por ejemplo, el dispositivo informático 20 puede incluir uno o más puentes de comunicación para la transferencia de datos entre los componentes del dispositivo informático 20. Además, los componentes del dispositivo informático 20 mostrados en la FIG. 1 tal vez no sean necesarios en todos los ejemplos del dispositivo informático 20. Por ejemplo, la interfaz de usuario 40 y la pantalla 44 pueden ser externas al dispositivo informático 20 en ejemplos en los que el dispositivo informático 20 es un ordenador de sobremesa.

El procesador anfitrión 24 puede incluir uno cualquiera o más de los siguientes: un microprocesador, un controlador, un procesador de señales digitales (DSP), un circuito integrado específico de la aplicación (ASIC), una formación de compuertas programables en el terreno (FPGA), o circuitos equivalentes de lógica discreta o integrada. Además, las funciones atribuidas al procesador anfitrión 24, en esta divulgación, pueden realizarse como software, firmware, hardware o cualquier combinación de los mismos.

El procesador anfitrión 24 procesa las instrucciones para su ejecución dentro del dispositivo informático 20. El procesador anfitrión 24 puede ser capaz de procesar instrucciones almacenadas en el dispositivo de almacenamiento 28 o instrucciones almacenadas en la memoria 32. El procesador anfitrión 24 puede generar un flujo de comandos utilizando un controlador (por ejemplo, que puede implementarse en software ejecutado por el procesador anfitrión 24) para su ejecución mediante la GPU 48. Es decir, el procesador anfitrión 24 puede generar un flujo de comandos que define las operaciones a realizar por la GPU 48.

El procesador anfitrión 24 puede generar un flujo de comandos a ejecutar mediante la GPU 48, que hace que el contenido visible se exhiba en la pantalla 44. Por ejemplo, el procesador anfitrión 24 puede generar un flujo de comandos que proporciona instrucciones para la GPU 48, para representar datos de gráficos. En este ejemplo, el procesador anfitrión 24 puede generar un flujo de comandos que es ejecutado por un canal de representación de gráficos, tal como el canal mostrado y descrito con respecto a la FIG. 2.

Además, o alternativamente, el procesador anfitrión 24 puede generar un flujo de comandos a ejecutar por la GPU 48, que hace que la GPU 48 realice otras operaciones. Por ejemplo, en algunos casos, el procesador anfitrión 24 puede generar un flujo de comandos para el uso de la GPU 48 como una unidad de procesamiento de gráficos de propósito general (GPGPU). Por ejemplo, la GPU 48 puede llevar a cabo múltiples funciones de cálculo de propósito general, realizadas tradicionalmente por el procesador anfitrión 24. Entre los ejemplos se incluyen múltiples funciones de procesamiento de imágenes, incluyendo la decodificación de vídeo y el post-procesamiento (por ejemplo, desbloqueo, reducción de ruido, corrección de color y similares) y otras funciones de procesamiento de imágenes específicas de la aplicación (por ejemplo, detección / reconocimiento facial, reconocimiento de patrones, transformaciones de ondículas y similares). En algunos ejemplos, la GPU 48 puede colaborar con el procesador anfitrión 24 para ejecutar tales aplicaciones de una GPGPU. Por ejemplo, el procesador anfitrión 24 puede descargar ciertas funciones a la GPU 48, proporcionando a la GPU 48 un flujo de comandos para su ejecución mediante la GPU 48.

El dispositivo de almacenamiento 28 puede incluir uno o más medios de almacenamiento legibles por ordenador. El dispositivo de almacenamiento 28 puede estar configurado para el almacenamiento a largo plazo de la información. En algunos ejemplos, el dispositivo de almacenamiento 28 puede incluir elementos de almacenamiento no volátiles. Los ejemplos de tales elementos de almacenamiento no volátiles pueden incluir discos duros magnéticos, discos ópticos, disquetes, memorias flash o formas de memorias programables eléctricamente (EPROM) o memorias borrables y programables eléctricamente (EEPROM). En algunos ejemplos, el dispositivo de almacenamiento 28 puede considerarse como un medio de almacenamiento no transitorio. La expresión "no transitorio" puede indicar que el medio de almacenamiento no está integrado en una onda portadora o una señal propagada. Sin embargo, no debería interpretarse que la expresión "no transitorio" significa que el dispositivo de almacenamiento 28 es no móvil. Como un ejemplo, el dispositivo de almacenamiento 28 puede retirarse del dispositivo informático 20 y trasladarse a otro dispositivo. Como otro ejemplo, un dispositivo de almacenamiento, esencialmente similar al dispositivo de almacenamiento 28, puede insertarse en el dispositivo informático 20. El dispositivo de almacenamiento 28 puede almacenar instrucciones para la ejecución de una o más aplicaciones mediante el procesador anfitrión 24 o la GPU 48.

La memoria 32 puede estar configurada para almacenar información dentro del dispositivo informático 20 durante el funcionamiento. En algunos ejemplos, la memoria 32 es una memoria temporal, lo cual significa que un propósito principal de la memoria 32 no es el almacenamiento a largo plazo. En algunos ejemplos, la memoria 32 puede describirse como un medio de almacenamiento legible por ordenador. En consecuencia, la memoria 32 también puede considerarse "no transitoria", a pesar de almacenar datos que pueden cambiar con el tiempo. La memoria 32 puede también, en algunos ejemplos, describirse como una memoria volátil, lo cual significa que la memoria 32 no mantiene contenidos almacenados cuando el ordenador está apagado. Los ejemplos de memorias volátiles incluyen memorias de acceso aleatorio (RAM), memorias dinámicas de acceso aleatorio (DRAM), memorias estáticas de acceso aleatorio (SRAM) y otras formas de memorias volátiles conocidas en la técnica.

En algunos ejemplos, la memoria 32 puede utilizarse para almacenar instrucciones de programa para su ejecución mediante el procesador anfitrión 24 o la GPU 48. La memoria 32 puede ser utilizada por software o aplicaciones que se ejecuten en el dispositivo informático 20 para almacenar temporalmente la información durante la ejecución de programas. Como tal, puede accederse a la memoria 32 desde otros componentes del dispositivo informático 20, tales como el procesador anfitrión 24 y la GPU 48.

Según algunos aspectos de la divulgación, la memoria 32 puede implementarse como una memoria intermedia de visualización que almacena datos de píxeles (por ejemplo, una trama de datos de vídeo, una imagen de gráficos generados por ordenador, una imagen fija y similares) antes de que los datos se exhiban mediante la pantalla 44. Por ejemplo, la GPU 48 puede determinar los datos de píxeles en la memoria 32 antes de que los datos de píxeles se exhiban mediante la pantalla 44.

El dispositivo informático 20 puede utilizar el módulo de red 36 para comunicarse con dispositivos externos a través de una o más redes, tales como una o más redes inalámbricas. El módulo de red 36 puede ser una tarjeta de interfaz de red, tal como una tarjeta de Ethernet, un transceptor óptico, un transceptor de frecuencia de radio o cualquier otro tipo de dispositivo que pueda enviar y recibir información. En algunos ejemplos, el dispositivo informático 20 puede utilizar el módulo de red 36 para comunicarse de forma inalámbrica con un dispositivo externo, tal como un servidor, un teléfono móvil u otro dispositivo informático en red.

El dispositivo informático 20 también incluye la interfaz de usuario 40. Los ejemplos de interfaz de usuario 40 incluyen, pero sin limitarse a, una bola de rastro, un ratón, un teclado y otros tipos de dispositivos de entrada. La

interfaz de usuario 40 también puede incluir una pantalla táctil que se incorpora como parte de la pantalla 44. La pantalla 44 puede comprender una pantalla de cristal líquido (LCD), una pantalla orgánica de diodos emisores de luz (OLED), una pantalla de plasma u otro tipo de dispositivo de pantalla.

5 La GPU 48 del dispositivo informático 20 puede ser una unidad de hardware específica que tiene una función fija y componentes programables para la representación de gráficos y la ejecución de aplicaciones de una GPGPU. Por ejemplo, según el ejemplo mostrado en la FIG. 1, la GPU 48 incluye un procesador sombreador programable 52, unidades de función fija 54 (que pueden incluir componentes de hardware de función fija) y un procesador de comandos 56. La GPU 48 también puede incluir un DSP, un microprocesador de propósito general, un ASIC, una
10 FPGA u otros circuitos lógicos equivalentes, discretos o integrados. La GPU 48 también puede incluir otros componentes, tales como una memoria específica de la GPU, como se describe con mayor detalle con respecto a las FIGs. 3 a 4

15 Además, aunque se muestran como componentes separados en la FIG. 1, en algunos ejemplos, la GPU 48 puede estar formada como parte del procesador anfitrión 24. La GPU 48 puede estar configurada para utilizar técnicas de procesamiento según múltiples interfaces de programación de aplicaciones (API). Por ejemplo, un usuario puede programar una aplicación para ser ejecutada por la GPU 48 usando una interfaz de software estándar que puede ejecutarse en múltiples plataformas, sistemas operativos y hardware.

20 En algunos ejemplos, la GPU 48 puede estar configurada para utilizar aplicaciones generadas utilizando la plataforma OpenGL, incluyendo OpenGL para sistemas Integrados ("OpenGL ES", publicada en marzo de 2007 y disponible para el público). Otros ejemplos de las API incluyen la Arquitectura de Dispositivos Unificados en Cálculo ("CUDA" desarrollada por NVIDIA Corporation, versión 3.2 publicada el 17 de septiembre de 2010) y DirectX (desarrollada por Microsoft, Inc., versión 11, publicada el 27 de octubre de 2009). En general, una API incluye un
25 conjunto predeterminado y estandarizado de comandos que se ejecutan mediante el hardware asociado. Los comandos de API permiten a un usuario dar instrucciones a componentes de hardware de una GPU para ejecutar comandos sin el conocimiento del usuario en cuanto a los detalles de los componentes de hardware.

30 Si bien el ejemplo de la FIG. 1 muestra el procesador sombreador de 52 como un solo bloque, el procesador sombreador 52 puede incluir una o más unidades de procesamiento sombreador y puede denominarse, en general, un "procesador sombreador unificado". Es decir, por ejemplo, el procesador sombreador 52 puede llevar a cabo operaciones de geometría, vértices, píxeles u otras operaciones de sombreado (tales como las descritas con respecto a las etapas de sombreador que se muestran en el ejemplo de la FIG. 2) para representar gráficos. En otro ejemplo, el procesador sombreador 52 puede realizar cálculos de propósito general. Es decir, el procesador sombreador 52 puede ejecutar instrucciones asociadas a una aplicación de una GPGPU. Una aplicación de una GPGPU normalmente incluye uno o más núcleos, que definen las funciones que pueden utilizarse para analizar o modificar múltiples datos de entrada. Los ejemplos incluyen funciones para el procesamiento de conjuntos de datos numéricos relativamente grandes en paralelo. En un contexto de procesamiento de imágenes, las funciones pueden incluir, por ejemplo, algoritmos de corrección de color, algoritmos de detección de rostros o funciones para la
35 realización de aplicaciones de realidad aumentada. Otros ejemplos incluyen funciones de transformación, funciones de trazado de rayos u otras múltiples funciones.

40 El procesador sombreador 52 generalmente puede configurarse para ejecutar una pluralidad de instrucciones en paralelo. Por ejemplo, el procesador sombreador 52 puede incluir una formación de unidades de procesamiento (por ejemplo, tales como unidades de lógica y aritmética ("ALU")) que ejecutan las instrucciones en paralelo. Según ello, el procesador sombreador 52 puede incluir uno o más componentes que no se muestran específicamente en la FIG. 3, tales como componentes para búsqueda y decodificación de instrucciones, componentes para el equilibrio de cargas y una o más ALU, u otras unidades de cálculo, para la realización de cálculos. El procesador sombreador 52 también puede incluir una o más memorias, memorias caché o registros (por ejemplo, tales como la memoria del procesador sombreador 60, mostrada y descrita con respecto al ejemplo de la FIG. 3).
45

50 Las unidades de función fija 54 pueden incluir una o más unidades, tales como los componentes de hardware de función fija, para la representación de datos de gráficos. Por ejemplo, las unidades de función fija 54 pueden incluir unidades para la realización de un ensamblaje inicial de entrada de datos de gráficos para preparar los datos de gráficos (triángulos, líneas y puntos) para su representación. En otro ejemplo, las unidades de función fija 54 pueden incluir unidades para la realización de la pixelación. Es decir, las unidades de función fija 54 pueden preparar primitivas para el sombreado durante la pixelación. Las unidades de función fija 54 también pueden realizar otras múltiples operaciones asociadas a la representación de datos de gráficos y/o realizar otras operaciones.
55

60 Como se ha señalado anteriormente, la GPU 48 puede representar gráficos para su visualización en la pantalla 44. Por ejemplo, la GPU 48 puede utilizar el procesador sombreador 52 y/o las unidades de función fija 54 para realizar las operaciones asociadas a las etapas de un canal de representación de gráficos, tal como el canal ejemplar mostrado en la FIG. 2. Según algunos aspectos de la divulgación, la GPU 48 puede implementar una arquitectura de representación basada en placas. Por ejemplo, en lugar de representar una escena completa de datos de píxeles (por ejemplo, una trama de datos de vídeo, una imagen de gráficos generados por ordenador, una imagen fija y similares), la GPU 48 pueden dividir una escena en varios trozos (por ejemplo, "placas") y representar de forma
65

individual las placas. La GPU 48 puede escribir cada una de las placas representadas en una memoria intermedia, tal como una memoria intermedia de visualización incluida en la memoria 32. Tras escribir todas las placas de la escena en la memoria 32, el dispositivo informático 20 puede ensamblar las placas representadas y exhibir la escena en la pantalla 44.

5 Cuando se implementa como una GPGPU, la GPU 48 puede ejecutar programas sombreadores, denominados núcleos en el presente documento. Por ejemplo, como se describe en mayor detalle con respecto a la FIG. 3, el procesador sombreador 52 puede ejecutar núcleos para realizar múltiples funciones de cálculo de propósito general, tales como las funciones de procesamiento de imágenes y otras funciones. Un núcleo puede ser definido por un usuario utilizando una API, tal como las API ejemplares descritas anteriormente. Los núcleos pueden comprender elementos de trabajo individuales (por ejemplo, una unidad básica de trabajo en una GPU) que están agrupados en grupos de trabajo. En algunos ejemplos, la GPU 48 puede ejecutar núcleos en una secuencia particular al ejecutar un flujo de comandos de una GPGPU. Es decir, el procesador sombreador 52 puede ejecutar múltiples instancias del mismo núcleo en paralelo antes de pasar al siguiente núcleo. En otros ejemplos, la GPU 48 puede ejecutar múltiples núcleos de forma simultánea.

En general, como se ha señalado anteriormente, la GPU 48 recibe un flujo de comandos desde un procesador anfitrión, tal como el procesador anfitrión 24. El flujo de comandos se genera normalmente mediante un controlador de software ejecutado por el procesador anfitrión 24, y controla el funcionamiento del procesador sombreador 52 y las unidades de función fija 54. Por ejemplo, el flujo de comandos controla habitualmente qué componentes de la GPU 48 realizan las operaciones definidas en el flujo de comandos.

Debido a las limitaciones impuestas por la configuración de algunas GPU y las limitaciones de un flujo de comandos, una GPU, tal como la GPU 48, normalmente puede ejecutar una sola tarea a la vez. Por ejemplo, la GPU 48 puede ejecutar en serie cada operación incluida en un flujo de comandos. Por lo tanto, ciertos recursos de la GPU 48 pueden estar inactivos mientras otros están procesando instrucciones según el flujo de comandos. Es decir, el procesador sombreador 52 de la GPU 48 puede ejecutar una operación según un flujo de comandos, mientras las unidades de función fija 54 están inactivas.

En algunos ejemplos, la multitarea puede ser emulada por el procesador anfitrión 24 mediante la conmutación entre más de un flujo de comandos en los momentos adecuados. La conmutación del flujo de comandos, sin embargo, puede ser relativamente intensa en términos de tiempo y puede dificultar compartir eficazmente los recursos de la GPU para realizar múltiples tareas. Por ejemplo, puede haber una latencia relativamente alta asociada a la conmutación desde un flujo de comandos a otro. Por consiguiente, tal vez no sea eficaz conmutar entre diversas operaciones de representación y otras operaciones, tales como operaciones de propósito general (por ejemplo, procesamiento de imágenes, trazado de rayos o similares). Además, tal vez no sea eficaz asignar tareas de alta prioridad, críticas en el tiempo (por ejemplo, tareas de procesamiento de multimedios, tales como el procesamiento de audio), a la GPU 48 mediante un flujo de comandos, debido a la incertidumbre respecto a cuándo se ejecutará la tarea. Por ejemplo, el tiempo de ejecución de las operaciones incluidas en un flujo de comandos puede ser difícil de determinar y puede variar significativamente en función de los comandos existentes que son ejecutados por la GPU 48.

Según aspectos de la divulgación, el procesador de comandos 56 puede controlar localmente los recursos de la GPU sin la intervención de un procesador anfitrión, tal como el procesador anfitrión 24 u otra unidad de procesamiento del anfitrión (por ejemplo, una unidad de procesamiento central (CPU)). Por ejemplo, según los aspectos de esta divulgación, el procesador de comandos 56 de la GPU 48 puede recibir una o más "tareas" desde el procesador anfitrión 24. El procesador de comandos 56 puede planificar de forma independiente las tareas a ejecutar por los recursos de la GPU 48, incluyendo, por ejemplo, el procesador sombreador 52 y las unidades de función fija 54. Es decir, en lugar de recibir un flujo de comandos desde el procesador anfitrión 24 que dicta cómo se utilizan los recursos de la GPU, el procesador de comandos 56 puede recibir una o más tareas de nivel superior que generalmente definen el trabajo que va a ser realizado por la GPU. Estas tareas pueden definir operaciones que han de ser realizadas por la GPU 48 sin dictar qué recursos de la GPU 48 se van a utilizar para realizar las operaciones.

El procesador de comandos 56 puede determinar de forma independiente cuándo ejecutar las tareas y/o qué recursos son para ejecutar las tareas. Es decir, los aspectos de esta divulgación se pueden referir al procesador de comandos 56 determinando de forma independiente cuándo ejecutar las tareas y/o qué recursos son para ejecutar las tareas, lo cual, por lo general, puede hacer referencia a la capacidad del procesador de comandos 56 para controlar la ejecución de las tareas sin la intervención del procesador anfitrión 24 (por ejemplo, sin la intervención proporcionada, por ejemplo, desde un flujo de comandos generado por el procesador anfitrión 24). Como se describe en mayor detalle más adelante, tales determinaciones se pueden hacer sobre la base de la disponibilidad de recursos y/o la prioridad de tareas, entre otras consideraciones potenciales. El procesador de comandos 56 también puede controlar independientemente el flujo de datos entre los distintos recursos de la GPU 48.

El procesador de comandos 56 puede controlar simultáneamente múltiples tareas, incluyendo la distribución de forma independiente y selectiva de tareas a los diferentes recursos de la GPU 48, tales como el procesador sombreador 52 y/o las unidades de función fija 54. Es decir, el procesador de comandos 56 puede controlar la

ejecución de las tareas sin la intervención del procesador anfitrión 24, y seleccionar qué recursos de la GPU 48 ejecutan cada tarea, sin estar todos los recursos de la GPU 48 necesariamente dedicados a la ejecución de una sola tarea. En el ejemplo, el procesador de comandos 56 puede planificar una tarea de representación de gráficos, a ser ejecutada por las unidades de función fija 54 de la GPU 48, planificando también al mismo tiempo selectivamente una tarea de cálculo a ser ejecutada simultáneamente por el procesador sombreador 52 de la GPU 48. De esta manera, el procesador de comandos 56 puede habilitar la GPU 48 para ejecutar varias tareas en paralelo, sin tener que esperar a que el procesador anfitrión 24 conmute los flujos de comandos. Más bien, el procesador anfitrión 24 puede enviar una pluralidad de tareas al procesador de comandos 56, y el procesador de comandos 56 puede controlar la ejecución de las tareas.

La capacidad de la GPU 48 para controlar localmente los recursos de la GPU sin la intervención del procesador anfitrión 24 puede proporcionar una mayor flexibilidad y promover una gestión eficaz de los recursos de la GPU. Por ejemplo, como se ha descrito anteriormente, el procesador anfitrión 24 transmite tradicionalmente un flujo de comandos a la GPU 48, que la GPU 48 ejecuta secuencialmente utilizando los recursos de la GPU 48 a modo de canal. Una sola operación es realizada normalmente por la GPU 48 en un momento dado, de manera que algunos de los recursos de la GPU pueden estar inactivos mientras se espera a que otros recursos finalicen la ejecución de un flujo particular de comandos. Es decir, en un ejemplo con fines de ilustración, ciertas unidades de función fija 54 pueden estar realizando operaciones de representación de gráficos mientras el procesador sombreador 52 se encuentra inactivo.

Según aspectos de esta divulgación, el procesador de comandos 56 puede aumentar la eficacia mediante la reducción de la cantidad de tiempo en que los recursos de la GPU 48 están inactivos. Por ejemplo, en lugar de tratar recursos de la GPU como una serie interconectada de componentes que solo pueden procesar un flujo de comandos a la vez, el procesador de comandos 56 puede controlar de forma individual y selectiva los recursos de la GPU 48. En el ejemplo anteriormente proporcionado, el procesador de comandos 56 puede planificar una tarea de representación de gráficos para las unidades de función fija 54 de la GPU 48, mientras también planifica una tarea de cálculo diferente para el procesador sombreador 52 de la GPU 48. En consecuencia, el procesador de comandos 56 reduce potencialmente la cantidad de tiempo en que los recursos de la GPU están inactivos, mediante la planificación de tareas de forma selectiva para los diferentes recursos de la GPU 48, en lugar de hacer que todos los recursos de la GPU 48 ejecuten una tarea a la vez.

La FIG. 2 ilustra un canal de representación de gráficos 80. En general, el canal de representación de gráficos 80 se puede utilizar para la representación de gráficos con una GPU, tal como la GPU 48 (FIG. 1). Se debería entender que el canal 80 se proporciona meramente con fines de explicación y que pueden incluirse menos etapas, etapas adicionales o etapas alternativas en el canal 80.

El canal de representación de gráficos 80 generalmente incluye etapas programables (por ejemplo, ilustradas con esquinas redondeadas) y etapas de función fija (por ejemplo, ilustradas con esquinas cuadradas). Por ejemplo, las operaciones de representación de gráficos, asociadas a ciertas etapas del canal de representación de gráficos 80, se realizan generalmente mediante un procesador sombreador programable, tal como el procesador sombreador 52, mientras que otras operaciones de representación de gráficos, asociadas a otras etapas del canal de representación de gráficos 80, se realizan generalmente mediante unidades de hardware de función fija, no programables, tales como las unidades de función fija 54. Las etapas de representación de gráficos realizadas por el procesador sombreador 52 generalmente pueden denominarse etapas "programables", mientras que las etapas llevadas a cabo por las unidades de función fija 54 generalmente pueden denominarse etapas de función fija.

La etapa ensambladora de entrada 82 se muestra en el ejemplo de la FIG. 2 como una de las etapas de función fija y generalmente es responsable del suministro de datos de gráficos (triángulos, líneas y puntos) al canal de representación de gráficos 80. Por ejemplo, la etapa ensambladora de entrada 82 puede recopilar datos de vértices para las superficies de orden alto, primitivas y similares, y enviar datos de vértices y atributos a la etapa sombreadora de vértices 84.

La etapa sombreadora de vértices 84 puede procesar los datos y los atributos de vértices recibidos. Por ejemplo, la etapa sombreadora de vértices 84 puede realizar un procesamiento por vértice, tal como transformaciones, revestimiento, desplazamiento de vértices y cálculo de los atributos materiales por vértice. En algunos ejemplos, la etapa sombreadora de vértices 84 puede generar coordenadas de textura, color de vértices, iluminación de vértices, factores de niebla y similares. La etapa sombreadora de vértices 84 generalmente toma un solo vértice de entrada y emite un solo vértice de salida procesado.

El proceso de cuadrículado puede generalmente ser realizado por la etapa sombreadora de recubrimiento 86, la etapa de cuadrículado 88, y la etapa sombreadora de dominio 90. Por ejemplo, la etapa sombreadora de recubrimiento 86 puede generar factores de cuadrículado para trasladar a la etapa de cuadrículado 88. En un ejemplo, la etapa sombreadora de recubrimiento 86 puede transformar los datos de entrada que definen una superficie de orden bajo en puntos de control que constituyen un parche. Un parche puede incluir datos para cada uno entre una pluralidad de nodos que especifican juntos ciertas propiedades (por ejemplo, tal como la geometría) para una parte relativamente pequeña de una superficie de un objeto. La etapa de cuadrículado 88 puede ser una

etapa de función fija que utiliza los factores de cuadrículado de la etapa sombreadora de recubrimiento 86 para cuadricular (o subdividir) un parche en múltiples primitivas triangulares o cuádruples. Cada vértice resultante de la etapa de cuadrículado 88 puede emitirse a la etapa sombreadora de dominio 90. La etapa sombreadora de dominio 90 puede evaluar la representación de la superficie en cada vértice. La etapa sombreadora de dominio 90 puede enviar los datos completos para cada vértice (por ejemplo, posición, coordenadas de texturas, etc.) al sombreador de geometría 92.

La etapa sombreadora de geometría 92 puede recibir una primitiva definida por los datos de vértices (por ejemplo, tres vértices para un triángulo, dos vértices para una línea o un solo vértice para un punto) y procesar adicionalmente la primitiva. Por ejemplo, la etapa sombreadora de geometría 92 puede llevar a cabo el procesamiento por cada primitiva, tal como la detección de bordes de silueta y la extrusión de volumen de sombra, entre otras posibles operaciones de procesamiento.

La etapa pixeladora 94 es habitualmente una etapa de función fija que es responsable de recortar primitivas y preparar primitivas para la etapa sombreadora de píxeles 96. Por ejemplo, la etapa pixeladora 94 puede generar una serie de fragmentos para el sombreado mediante el sombreador de píxeles 96. La etapa sombreadora de píxeles 96 recibe fragmentos desde la etapa pixeladora 94 y genera datos por píxel, tales como el color. La etapa sombreadora de píxeles 96 también puede realizar un procesamiento por píxel, tal como la mezcla de texturas y el cálculo del modelo de iluminación. La etapa de fusión de salida 98 es generalmente responsable de la combinación de varios tipos de datos de salida (tales como valores del sombreador de píxeles, información de profundidad y de plantilla) para generar un resultado final.

Como se ha señalado anteriormente, el canal de representación de gráficos 80 incluye generalmente etapas programables (por ejemplo, ilustradas con esquinas redondeadas) y etapas de función fija (por ejemplo, ilustradas con esquinas cuadradas). En consecuencia, algunas de las etapas del canal de representación de gráficos 80 se llevan a cabo normalmente mediante componentes programables, tales como un procesador sombreador (por ejemplo, el procesador sombreador 52 mostrado en el ejemplo de la FIG. 1), mientras que otras etapas del canal de representación de gráficos 80 se realizan habitualmente mediante unidades de hardware, no programables, de función fija (por ejemplo, las unidades de función fija 54 mostradas en el ejemplo de la FIG. 1).

Según aspectos de esta divulgación, un procesador de comandos (por ejemplo, el procesador de comandos 56 mostrado en el ejemplo de la FIG. 1) se puede utilizar para controlar de forma independiente cada una de las etapas del canal de representación de gráficos 80. Por ejemplo, en lugar de que los datos sean procesados de forma secuencial por cada etapa del canal 80, según aspectos de esta divulgación, el procesador de comandos 56 puede controlar independientemente las etapas del canal 80. Es decir, el procesador de comandos 56 puede distribuir una primera tarea de cálculo al ensamblador de entrada 82, y distribuir una segunda tarea de cálculo a una de las etapas sombreadoras (que usa el procesador sombreador 52). En este ejemplo, la segunda tarea de cálculo es encaminada o ejecutada por el ensamblador de entrada 82, como ocurriría normalmente en el procesamiento de canal. En consecuencia, el ensamblador de entrada 82 puede estar realizando una función de representación de gráficos, mientras el procesador sombreador 52 está ejecutando otra tarea (tal como una operación de una GPGPU).

La FIG. 3 es un diagrama de bloques que ilustra la GPU 48 con mayor detalle, que puede implementarse para llevar a cabo las técnicas de esta divulgación. En el ejemplo mostrado en la FIG. 3, la GPU 48 incluye el procesador sombreador 52, las unidades de función fija 54 y el procesador de comandos 56. Además, en el ejemplo mostrado en la FIG. 3, la GPU incluye la memoria del procesador sombreador 60 (que se muestra como incorporada con el procesador sombreador 52) y un motor de copia de memoria 62. Debería entenderse que ciertos módulos y/o unidades de la FIG. 3 pueden estar sumamente integrados, pero se ilustran por separado con fines conceptuales. Por otra parte, los ciertos módulos pueden describirse con respecto a un solo módulo con fines conceptuales, pero pueden incluir una o más unidades funcionales y/o estructurales.

También debería entenderse que la FIG. 3 se proporciona meramente como un ejemplo de una GPU que puede utilizar las técnicas descritas en esta divulgación. Las técnicas descritas con respecto a esta divulgación pueden ser realizadas por otras múltiples GPU que tienen otros componentes. Es decir, en otros ejemplos, la GPU 48 también puede incluir otros múltiples componentes y módulos relacionados con la representación de imágenes, el análisis de imágenes y/o la realización de otros cálculos. Por ejemplo, la GPU 48 puede incluir unidades de textura, unidades de planificación, unidades lógicas y aritméticas (ALU) u otros componentes de la GPU que no se muestran en la FIG. 3, con fines de claridad.

Como se ha señalado con respecto al ejemplo de la FIG. 1 anteriormente, el procesador sombreador 52 puede incluir una o más unidades de procesamiento de sombreador que pueden realizar representación de gráficos y/u otras operaciones de propósito general. Es decir, por ejemplo, el procesador sombreador 52 puede llevar a cabo operaciones de geometría, vértice, píxel u otras operaciones de sombreado para representar gráficos. En otros ejemplos, el procesador sombreador 52 puede realizar cálculos de propósito general.

Las unidades de función fija 54 pueden incluir generalmente una o más unidades, tales como los componentes de hardware de función fija, para la representación de datos de gráficos. Por ejemplo, las unidades de función fija 54

pueden incluir unidades para la realización de un ensamblaje de entrada inicial de datos de gráficos para preparar los datos de gráficos (triángulos, líneas y puntos) para su representación. En otro ejemplo, las unidades de función fija 54 pueden incluir unidades para la realización de la pixelación. Es decir, las unidades de función fija 54 pueden preparar primitivas para el sombreado durante la pixelación. Las unidades de función fija 54 también pueden realizar otras múltiples operaciones asociadas a la representación de datos de gráficos y/o a la realización de otras operaciones.

El procesador de comandos 56 puede incluir uno cualquiera o más de los siguientes: un microprocesador, un controlador, un procesador de señales digitales (DSP), un circuito integrado específico de la aplicación (ASIC), una formación de compuertas programables en el terreno (FPGA) o circuitos lógicos equivalentes, discretos o integrados. Además, las funciones atribuidas al procesador de comandos 56 pueden, en algunos ejemplos, ser realizadas como software, firmware, hardware o cualquier combinación de los mismos. Mientras que el procesador de comandos 56 mostrado se muestra como una unidad por separado del procesador sombreador 52, en algunos ejemplos, el procesador de comandos 56 puede estar integrado con el procesador sombreador 52.

Como se ha indicado anteriormente, con respecto a la FIG. 1, el procesador de comandos 56 puede controlar localmente los recursos de la GPU sin la intervención del procesador anfitrión 24 u otra unidad de procesamiento. Por ejemplo, según los aspectos de esta divulgación, el procesador de comandos 56 de la GPU 48 puede recibir una o más "tareas" desde el procesador anfitrión 24. El procesador de comandos 56 puede planificar de forma independiente las tareas a ejecutar mediante los recursos de la GPU 48, incluyendo, por ejemplo, el procesador sombreador 52 y las unidades de función fija 54. Es decir, en lugar de recibir un flujo de comandos desde el procesador anfitrión 24 que dicta cómo se utilizan los recursos de la GPU, el procesador de comandos 56 puede recibir una o más tareas de nivel superior que generalmente definen el trabajo que va a ser realizado por la GPU. Tales tareas pueden definir operaciones que han de ser realizadas por la GPU 48 sin dictar qué recursos de la GPU 48 se van a utilizar para realizar las operaciones. La GPU 48 puede determinar de forma independiente cuándo ejecutar las tareas y/o qué recursos son para ejecutar las tareas.

La memoria de la GPU 58 puede ser similar a la memoria 32 mostrada en la FIG. 1. Es decir, en algunos ejemplos, la memoria de la GPU 58 puede ser un medio temporal de almacenamiento legible por ordenador. Los ejemplos de memoria de la GPU 58 incluyen memorias de acceso aleatorio (RAM), memorias dinámicas de acceso aleatorio (DRAM), memorias estáticas de acceso aleatorio (SRAM) y otras formas de registros y memorias conocidos en la técnica. En ejemplos donde la GPU 48 se forma como parte de otro procesador, tal como el procesador anfitrión 24, se puede acceder a la memoria de la GPU 58 mediante componentes distintos a la GPU 48. Normalmente, la memoria de la GPU 58 almacena datos que son utilizados en las operaciones realizadas por la GPU 48. En algunos ejemplos, la memoria de la GPU 58 puede recibir datos desde otra unidad de memoria, tal como la memoria 32 del dispositivo informático 20. Es decir, el dispositivo informático 20 puede transferir datos desde la memoria 32 a la memoria de la GPU 58 de modo que los datos sean accesibles para la GPU 48.

En el ejemplo mostrado en la FIG. 3, el procesador sombreador 52 incluye la memoria del procesador sombreador 60 ("memoria de SP"). Como se describe en mayor detalle a continuación con respecto al motor de copia de memoria 62, los datos pueden intercambiarse entre la memoria de SP 60 y la memoria de la GPU 58. Por ejemplo, la memoria de SP 60 puede recibir y almacenar datos sin procesar para su ejecución por el procesador sombreador 52, desde la memoria de la GPU 58. Es decir, la memoria de SP 60 puede almacenar los datos antes de que los datos sean procesados, por ejemplo, por las ALU del procesador sombreador 52. Además, la memoria de SP puede almacenar datos procesados antes de transferir los datos procesados a la memoria de la GPU 58. Se puede acceder a la memoria de SP 60 de forma relativamente rápida mediante las unidades dentro del procesador sombreador 52. Sin embargo, la transferencia de datos entre la memoria de SP 60 y la memoria de la GPU 58 puede consumir uno o más ciclos de reloj durante los cuales el procesador sombreador 52 tal vez no procese otros datos.

Según aspectos de esta divulgación, la GPU 48 también incluye el motor de copia de memoria 62, que está en comunicación con una o más unidades de memoria de la GPU 48. Por ejemplo, como se muestra en la FIG. 3, el motor de copia de memoria 62 puede estar en comunicación con la memoria 32, la memoria de la GPU 58 y la memoria de SP 60. En algunos ejemplos, el motor de copia de memoria 62 puede facilitar la transferencia de datos entre la memoria 32 (por ejemplo, una memoria de sistema) y la memoria de SP 60. Por ejemplo, según los aspectos de esta divulgación, el motor de copia de memoria 62 puede recibir instrucciones desde el procesador de comandos 56 que identifican los datos a transferir entre la memoria 32 y la memoria de SP 60 (y viceversa). Además, el motor de copia de memoria 62 puede recibir instrucciones desde el procesador de comandos 56 que identifican cuándo transferir datos entre la memoria 32 y la memoria de SP 60 (y viceversa). Tras la recepción de las instrucciones desde el procesador de comandos 56, el motor de copia de memoria 62 puede llevar a cabo la transferencia de datos entre la memoria 32 y la memoria de SP 60. En otros ejemplos, el motor de copia de memoria 62 también puede ser responsable de la transferencia de datos entre otras unidades de memoria de la GPU 48, tal como entre la memoria de la GPU 58 y la memoria de SP 60.

En algunos ejemplos, según aspectos de esta divulgación, ciertas unidades de la GPU 48 (por ejemplo, tales como el procesador sombreador 52, la memoria de SP 60, la memoria 32 y/o la memoria de la GPU 58) pueden "desconocer" las operaciones del motor de copia de memoria 62. Es decir, el motor de copia de memoria 62 puede

funcionar independientemente de las unidades a y desde las que se están copiando datos. En un ejemplo con fines de ilustración, en lugar de copiar, el procesador sombreador 52, datos desde la memoria 32 a la memoria de SP 60 (y usar los recursos y ciclos de reloj asociados para facilitar la transferencia de datos), dicha transmisión de datos puede ser gestionada por el motor de copia de memoria 62. Es decir, el motor de copia de memoria 62 puede copiar los datos de la memoria 32 a la memoria de SP 60, para su uso por el procesador sombreador 52. Según ello, el procesador sombreador 52 puede utilizar los datos almacenados en la memoria de SP 60 sin esperar a que los datos sean transferidos entre la memoria de SP 60 y la memoria 32. De esta manera, el motor de copia de memoria 62 puede proporcionar al procesador sombreador 52 los datos según se necesiten, y el procesador sombreador 52 puede seguir realizando operaciones en los datos sin utilizar ciclos de reloj para la transferencia de datos. Por ejemplo, el procesador de comandos 56 puede sincronizar el motor de copia de memoria 62 con las tareas planificadas por el procesador de comandos 56, de tal manera que la memoria de SP 60 esté dotada de los datos adecuados al ejecutar una tarea en particular. Como se ha señalado anteriormente, el motor de copia de memoria 62 también puede ser responsable de la transferencia de datos procesados entre la memoria de la GPU 58 y la memoria de SP 60. Además, o alternativamente, el motor de copia de memoria 62 puede transferir datos hacia y desde las unidades de función fija 54 y/u otros componentes de la GPU 48.

En funcionamiento, según aspectos de esta divulgación, el procesador anfitrión 24 puede generar un flujo de comandos que incluye una o más tareas a realizar por la GPU 48. Las tareas pueden exponer múltiples trabajos que han de ser realizados por la GPU 48, tales como la realización de la representación de gráficos y/u otras operaciones de propósito general. El procesador de comandos 56 de la GPU 48 puede recibir el flujo de comandos y facilitar la ejecución de cada una de las tareas incluidas en el flujo de comandos.

Según aspectos de esta divulgación, el procesador de comandos 56 puede analizar inicialmente el flujo de comandos recibido e identificar cada tarea que ha de ser realizada por la GPU 48. Además de analizar las tareas a partir de la secuencia de comandos, el procesador de comandos 56 puede mantener una cola de comandos para la organización de cada una de las tareas a ejecutar por los componentes de la GPU 48. Por ejemplo, el procesador de comandos 56 puede planificar tareas a ejecutar por los componentes de la GPU 48 (tales como el procesador sombreador 52 y/o las unidades de función fija 54) utilizando la cola de comandos. En algunos ejemplos, las colas de comandos pueden ser unidades de hardware de función fija (por ejemplo, registros del tipo 'primero en entrar, primero en salir' (FIFO), o similares). En otros ejemplos, las colas de comandos pueden ser unidades generales de memoria o de registro.

El procesador de comandos 56 también puede mantener colas de comandos para controlar otras funciones asociadas a la GPU 48. Por ejemplo, el procesador de comandos 56 puede mantener una cola de comandos para la organización del intercambio de datos con un procesador anfitrión (por ejemplo, una unidad de procesamiento central (CPU)). En otros ejemplos, el procesador de comandos 56 puede mantener una cola de comandos para la organización del intercambio de datos con un procesador de señales digitales (DSP) u otros componentes informáticos, tales como unidades de multimedia.

En algunos ejemplos, el procesador de comandos 56 puede planificar tareas en base a una disponibilidad de recursos de la GPU 48. Por ejemplo, el procesador de comandos 56 puede planificar tareas en base a interfaces de entrada / salida que estén disponibles. En otro ejemplo, el procesador de comandos 56 puede planificar tareas en base a si los datos que están siendo ejecutados están o no listos para su procesamiento. En otro ejemplo, el procesador de comandos 56 puede planificar tareas en base a si hay o no espacio disponible en la memoria (por ejemplo, la memoria de la GPU 58) para almacenar el resultado de las tareas.

Según algunos aspectos de esta divulgación, el procesador de comandos 56 puede priorizar las tareas en las colas de comandos. En algunos ejemplos, el componente responsable de la generación de la tarea (por ejemplo, el procesador anfitrión 24) puede ser responsable de indicar una prioridad para la tarea. En otros ejemplos, el procesador de comandos 56 puede determinar la prioridad en base a las tareas que se incluyen en la cola de comandos. Por ejemplo, el procesador de comandos 56 puede identificar una tarea que es más sensible al tiempo que otras tareas en la cola de comandos, y puede dar prioridad a la tarea de alta prioridad en la cola de comandos de manera que la tarea sea planificada y ejecutada antes que las otras tareas.

En algunos ejemplos, el procesador de comandos 56 también puede mantener una lista de tareas activas que están siendo ejecutadas por los componentes de la GPU 48, así como realizar el seguimiento del componente responsable de la ejecución de las tareas. La lista de tareas activas se puede utilizar, por ejemplo, para realizar el seguimiento del estado de las tareas. Mediante el seguimiento del estado de las tareas, el procesador de comandos 56 puede determinar qué recursos de la GPU 48 están disponibles para la planificación de tareas. Es decir, si una tarea está actualmente siendo ejecutada por un recurso de la GPU (por ejemplo, la tarea está "activa"), ese recurso tal vez no esté disponible hasta que se complete la tarea activa. En algunos ejemplos, el procesador de comandos 56 puede ser responsable de identificar qué componentes de la GPU 48 están ejecutando tareas de forma activa (por ejemplo, mediante el sondeo o la determinación, de otra manera, de que un componente está ocupado). En otros ejemplos, el procesador de comandos 56 puede recibir una indicación de que una determinada tarea ha sido completada por el componente responsable de la ejecución de la tarea, o mediante la monitorización de los datos que se escriben en la memoria de la GPU 58.

En un ejemplo, el procesador de comandos 52 puede iniciar la ejecución de una tarea por el procesador sombreador 52. A continuación, el procesador de comandos 56 puede agregar la tarea a la lista de tareas activas que están actualmente siendo ejecutadas por el procesador sombreador 52. Utilizando la lista de tareas activas, el procesador de comandos 56 puede realizar un seguimiento de los recursos que en la actualidad están ejecutando tareas, identificando también al mismo tiempo los recursos disponibles que no están actualmente ejecutando tareas.

Usando la información de prioridad y/o la lista de tareas activas, el procesador de comandos 56 puede suplantar una tarea que está siendo ejecutada por un componente de la GPU 48, si se añade una tarea de mayor prioridad a la cola de comandos, y esa tarea de mayor prioridad está lista para ser ejecutada (por ejemplo, los datos asociados a la tarea están listos para la ejecución). Es decir, el procesador de comandos 56 puede detener la ejecución de una tarea de prioridad más baja a fin de ejecutar una tarea de mayor prioridad, si la tarea de mayor prioridad está lista para ser ejecutada. El procesador de comandos 56 puede reanudar la tarea de prioridad más baja después de completar la ejecución de la tarea de mayor prioridad.

De esta manera, el procesador de comandos 56 puede distribuir de forma independiente tareas a diferentes recursos de la GPU 48, tales como un procesador sombreador 52 y/o las unidades de función fija 54, sin la intervención de un procesador anfitrión. En un ejemplo con fines de ilustración, el dispositivo informático 20 que incluye la GPU 48 puede ser una cámara digital (por ejemplo, un dispositivo móvil que incluye una cámara digital). En este ejemplo, la GPU 48 de la cámara digital puede estar representando imágenes para su exhibición en un visor. El usuario de la cámara digital puede seleccionar una función de procesamiento de imágenes para llevar a cabo sobre las imágenes exhibidas (por ejemplo, nitidez de imagen, reducción de ruido y similares). La GPU 48 puede conmutar desde la representación de las imágenes a la realización de las funciones de procesamiento de imágenes (de propósito general). En este ejemplo, la conmutación entre las funciones de la GPU 48 puede ser facilitada de forma independiente por el procesador de comandos 56.

La capacidad de la GPU 48 para controlar localmente los recursos de la GPU sin la intervención del procesador anfitrión 24 puede proporcionar una mayor flexibilidad y promover una gestión eficaz de los recursos de la GPU. Por ejemplo, según aspectos de esta divulgación, el procesador de comandos 56 puede reducir la cantidad de tiempo en que los recursos de la GPU 48 están inactivos. En el ejemplo anteriormente proporcionado, el procesador de comandos 56 puede planificar una tarea de representación de gráficos para las unidades de función fija 54 de la GPU 48, planificando también al mismo tiempo una tarea de cálculo para el procesador sombreador 52 de la GPU 48. Es decir, por ejemplo, con respecto al ejemplo de cámara digital descrito anteriormente, el procesador de comandos 56 puede planificar la tarea de representación de gráficos (por ejemplo, la representación de imágenes para la exhibición en el visor) a la vez que la planificación de una tarea de cálculo (por ejemplo, procesamiento de imágenes). En consecuencia, el procesador de comandos 56 reduce potencialmente la cantidad de tiempo en que los recursos de la GPU están inactivos.

Según aspectos de esta divulgación, las "tareas" del procesador anfitrión 24 pueden ser definidas por un usuario (por ejemplo, un desarrollador de aplicaciones) utilizando una API. Por ejemplo, como se ha señalado anteriormente, las API, tales como DirectX y OpenGL, permiten a un usuario desarrollar una aplicación para la representación de gráficos o realizar otras operaciones con la GPU 48. Las API tradicionales, sin embargo, pueden estar restringidas a un flujo de datos fijo (por ejemplo, por canal), que puede ser adecuado para un tipo de representación de gráficos, pero tal vez no se adapte de manera eficaz a otras tareas (por ejemplo, tales como el trazado de rayos o la difusión de rayos). Es decir, las aplicaciones desarrolladas utilizando las API tradicionales pueden encaminar datos de forma secuencial a través de cada etapa de un canal de representación (tal como el canal de representación de gráficos 80 que se muestra en el ejemplo de la FIG. 2), independientemente de si se necesita o no cada etapa para llevar a cabo una operación en particular.

Los aspectos de esta divulgación se refieren a una API que un usuario puede aplicar para generar una aplicación que tenga tareas de cálculo de un nivel relativamente alto. Por ejemplo, la API puede permitir a un usuario definir tareas que generalmente describen el trabajo que va a ser realizado por la GPU. Es decir, las tareas pueden definir operaciones que han de ser realizadas por la GPU 48 sin dictar qué recursos de la GPU 48 se van a utilizar para realizar las operaciones. En algunos ejemplos, múltiples tareas pueden incluirse en una API y exponerse a un desarrollador de aplicaciones. Utilizando la API, el desarrollador de aplicaciones puede generar una aplicación que, cuando se ejecute, utilice un canal personalizado (por ejemplo, en relación con canales de gráficos de las API tradicionales). En un ejemplo, un desarrollador de aplicaciones puede generar una aplicación que utilice un canal de gráficos personalizados que esté optimizado para realizar operaciones de representación particulares, tales como el trazado de rayos o la difusión de rayos.

Tal vez no sea necesaria una API, sin embargo, para llevar a cabo las técnicas de esta divulgación. Por ejemplo, según algunos aspectos, un compilador (por ejemplo, un programa compilador, tal como un compilador de C/C++) puede correlacionar partes de código de programa para ser ejecutadas por los componentes de la GPU 48. Es decir, el compilador puede identificar los componentes adecuados de la GPU 48 para ejecutar una parte determinada del código de programa, así como especificar una tarea que incluya la parte dada de código. El procesador de comandos 56 puede a continuación recibir las partes correlacionadas de código y planificar las tareas en

consecuencia.

La FIG. 4 es un diagrama de flujo que ilustra un procedimiento ejemplar 100 que puede ser realizado por un procesador de comandos, según aspectos de la divulgación. Es decir, por ejemplo, el procedimiento que se muestra en la FIG. 4 puede ser realizado por el procesador de comandos 56 mostrado en las FIGs. 1 y 3. Sin embargo, mientras que el procedimiento en el ejemplo de la FIG. 4 se describe con respecto al procesador de comandos 56, se debería entender que otras unidades de procesamiento de gráficos de otros dispositivos informáticos pueden realizar algunas de, o todas, las funciones atribuidas al procesador de comandos 56.

En el procedimiento ejemplar 100 de la FIG. 4, el procesador de comandos 56 recibe inicialmente una o más tareas, por ejemplo, desde un procesador anfitrión, tal como el procesador anfitrión 24 (102). Por ejemplo, el procesador de comandos 56 puede recibir un procesador de comandos que incluye una o más tareas que definen el trabajo a ser realizado por los componentes de la GPU 48. El procesador de comandos 56 puede analizar el flujo de comandos para identificar cada tarea incluida en el flujo de comandos (104).

El procesador de comandos 56 también puede determinar la prioridad de tareas (106). Por ejemplo, según algunos aspectos de esta divulgación, el procesador de comandos 56 puede mantener una o más colas de comandos que pueden determinar cuándo cada una de las tareas recibidas ha de ser ejecutada por los componentes de la GPU 48. El procesador de comandos 56 puede organizar las una o más colas de comandos en base a la prioridad de las tareas, de manera que las tareas más críticas y/o sensibles al tiempo se ejecuten antes de otras tareas. En algunos ejemplos, el componente responsable de la generación de la tarea (por ejemplo, el procesador anfitrión 24) puede ser responsable de indicar una prioridad para la tarea. En otros ejemplos, el procesador de comandos 56 puede determinar la prioridad en base a las tareas que se incluyen en la cola de comandos.

El procesador de comandos 56 también puede determinar la disponibilidad de recursos (108). Por ejemplo, el procesador de comandos 56 puede determinar si el (los) componente(s) responsable(s) de la ejecución de la tarea está(n) o no disponible(s) (por ejemplo, no ejecutando otra tarea). Por otra parte, el procesador de comandos 56 puede determinar si los datos asociados a la tarea están o no disponibles.

Además, o en lugar, de la organización de las una o más colas de comandos en base a la prioridad de las tareas, el procesador de comandos 56 puede organizar las una o más colas de comandos en base a la disponibilidad de recursos. Es decir, por ejemplo, el procesador de comandos 56 tal vez no pueda planificar una tarea para ser ejecutada por un componente de la GPU 48 a menos que el componente esté disponible para ejecutar la tarea. Por otra parte, el procesador de comandos 56 tal vez no pueda planificar una tarea a ser ejecutada por un componente de la GPU 48 a menos que los datos asociados a la tarea estén disponibles. Es decir, el procesador de comandos 56 puede esperar hasta que los datos asociados a una tarea en particular hayan sido desplazados por el motor de copia de memoria 62 a la memoria de SP 60, antes de planificar la tarea para su ejecución. Por otra parte, el procesador de comandos 56 puede asegurarse de que los datos asociados a una determinada tarea no estén siendo modificados por cualquier otro componente de la GPU 48 antes de planificar la tarea.

El procesador de comandos 56 puede entonces planificar una tarea para su ejecución por uno o más de los componentes de la GPU 48 (110). En algunos ejemplos, según aspectos de esta divulgación, el procesador de comandos 56 puede administrar de forma independiente las tareas de tal manera que los componentes de la GPU 48 puedan estar ejecutando diferentes tareas en paralelo. Por ejemplo, el procesador de comandos 56 puede planificar una tarea para ser ejecutada por el procesador sombreador 52, en paralelo con una tarea a ser ejecutada por las unidades de función fija 54.

Al planificar una tarea, el procesador de comandos 56 puede actualizar una lista de tareas activas (112), es decir, una lista de tareas activas. Por ejemplo, el procesador de comandos 56 puede mantener una lista de tareas activas que están siendo ejecutadas por los componentes de la GPU 48, así como realizar el seguimiento del componente responsable de la ejecución de la tarea. La lista de tareas activas se puede utilizar, por ejemplo, para realizar el seguimiento del estado de las tareas. Mediante el seguimiento del estado de las tareas, el procesador de comandos 56 puede determinar qué recursos de la GPU 48 están disponibles para la planificación de tareas, y qué recursos de la GPU 48 están ocupados ejecutando tareas. Es decir, si una tarea está siendo ejecutada actualmente por un recurso de la GPU (por ejemplo, la tarea está "activa"), ese recurso tal vez no esté disponible hasta que se complete la tarea activa. En un ejemplo, después de iniciar la ejecución de una tarea mediante el procesador sombreador 52, el procesador de comandos 56 puede agregar la tarea a la lista de tareas activas que están actualmente siendo ejecutadas por el procesador sombreador 52. En algunos ejemplos, el procesador de comandos 56 puede ser responsable de identificar qué componentes de la GPU 48 están ejecutando tareas de forma activa (por ejemplo, mediante el sondeo, o determinando de otra manera que un componente está ocupado). En otros ejemplos, el procesador de comandos 56 puede recibir una indicación de que una determinada tarea ha sido completada por el componente responsable de la ejecución de la tarea, o mediante la monitorización de los datos que se escriben en la memoria de la GPU 58.

También debería entenderse que las etapas mostradas y descritas con respecto a la FIG. 4 se proporcionan meramente como un ejemplo. Es decir, las etapas del procedimiento de la FIG. 4 no tienen que realizarse

necesariamente en el orden que se muestra en la FIG. 4, y se pueden realizar menos etapas, etapas adicionales o etapas alternativas. Por ejemplo, la FIG. 4 muestra la prioridad de la tarea como determinada antes de determinar la disponibilidad de recursos. En otro ejemplo, la determinación de la disponibilidad de recursos puede llevarse a cabo antes de determinar la prioridad de la tarea.

5 La FIG. 5 es un diagrama de flujo que ilustra un procedimiento ejemplar 120 que puede ser realizado por un procesador de comandos, tal como el procesador de comandos 56, según aspectos de la divulgación. Es decir, por ejemplo, el procedimiento que se muestra en la FIG. 5 puede ser realizado por el procesador de comandos 56
10 mostrado en las FIGs. 1 y 3. Sin embargo, mientras que el procedimiento que se muestra en el ejemplo de la FIG. 5 se describe con respecto al procesador de comandos 56, se debería entender que otras unidades de procesamiento de gráficos de otros dispositivos informáticos pueden realizar algunas de, o todas, las funciones atribuidas al procesador de comandos 56.

15 En algunos ejemplos, el procedimiento 120 puede llevarse a cabo en lugar de, o conjuntamente con, el procedimiento 100 que se muestra en el ejemplo de la FIG. 4. Por ejemplo, el procedimiento mostrado en el ejemplo de la FIG. 5 se puede realizar al planificar una tarea para su ejecución (por ejemplo, la etapa 110 que se muestra en la FIG. 4).

20 El procesador de comandos 56 inicialmente determina si uno o más recursos (por ejemplo, incluyendo datos, interfaces de entrada / salida, memoria y/o unidades de procesamiento) que se requieren para ejecutar una tarea en curso están o no disponibles (122). Si los recursos están disponibles (la rama "sí" de la etapa 122), el procesador de comandos 56 puede dar instrucciones al motor de copia de datos 62 para copiar los datos adecuados desde la memoria de la GPU 58 a la memoria de SP 60, si es necesario (124). Además, el procesador de comandos 56 puede iniciar la ejecución de la tarea (126). Es decir, el procesador de comandos 56 puede planificar la tarea para
25 ser ejecutada por el componente adecuado de la GPU 48.

Si los recursos para la ejecución de la tarea actual no están disponibles (la rama "no" de la etapa 122), el procesador de comandos 56 puede determinar si la tarea actual es o no la tarea de mayor prioridad para los recursos necesarios para ejecutar la tarea (128). Si la tarea actual es la tarea de prioridad más alta (la rama "sí" de la etapa 128), el
30 procesador de comandos 56 puede suplantar la tarea actualmente ejecutada por los recursos (130). Es decir, el procesador de comandos 56 puede interrumpir el funcionamiento de los recursos necesarios para ejecutar la tarea, para que la tarea actual se pueda ejecutar (por ejemplo, los recursos de la GPU no requeridos para ejecutar la tarea pueden continuar ejecutando otras tareas). Después de suplantar la tarea que se está ejecutando actualmente, el procesador de comandos 56 puede realizar las etapas 124 y 126 para ejecutar la tarea actual, como se ha descrito
35 anteriormente. En algunos ejemplos, el estado de la tarea suplantada puede almacenarse (por ejemplo, almacenarse en la memoria de la GPU 58) para permitir que la tarea se reanude después de que la tarea actual se haya ejecutado.

40 Si la tarea actual no es la tarea de mayor prioridad para los recursos necesarios para ejecutar la tarea (la rama "no" de la etapa 128), el procesador de comandos 56 puede esperar a que la tarea que está siendo ejecutada actualmente por los recursos termine la ejecución (132). Después de que los recursos completen la tarea (la rama "sí" de la etapa 134), el procesador de comandos puede ejecutar la tarea actual, por ejemplo, mediante la realización de las etapas 124 y 126, como se ha descrito anteriormente.

45 También debería entenderse que las etapas mostradas y descritas con respecto a la FIG. 5 se proporcionan meramente como un ejemplo. Es decir, las etapas del procedimiento de la FIG. 5 no tienen que realizarse necesariamente en el orden que se muestra en la FIG. 5, y se pueden realizar menos etapas, etapas adicionales o etapas alternativas.

50 Además, ha de reconocerse que, según el ejemplo, ciertos actos o sucesos de cualquiera de los procedimientos descritos en el presente documento pueden realizarse en una secuencia distinta, pueden añadirse, fusionarse u omitirse por completo (por ejemplo, no todos los actos o sucesos descritos son necesarios para la puesta en práctica del procedimiento). Además, en ciertos ejemplos, los actos o sucesos pueden realizarse simultáneamente, por ejemplo, mediante el procesamiento de múltiples hebras, el procesamiento de interrupciones o múltiples
55 procesadores, en lugar de secuencialmente.

Además, en uno o más ejemplos, las funciones descritas en el presente documento pueden implementarse en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones, como una o más instrucciones o código, pueden almacenarse en, o transmitirse por, un medio legible por ordenador,
60 y ejecutarse mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como medios de almacenamiento de datos o medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa informático desde un lugar a otro, por ejemplo, según un protocolo de comunicación.

65 De esta manera, los medios legibles por ordenador pueden corresponder, generalmente, a (1) medios de almacenamiento tangibles y legibles por ordenador, que sean no transitorios, o (2) un medio de comunicación tal

como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser medios disponibles cualesquiera, a los que se pueda acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

5 A modo de ejemplo, y no de manera limitativa, tales medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda usarse para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que
 10 pueda accederse mediante un ordenador. Además, cualquier conexión puede denominarse adecuadamente un medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde una sede de la Red, un servidor u otro origen remoto, usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se
 15 incluyen en la definición de medio.

Sin embargo, debería entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en cambio, se orientan a medios de almacenamiento tangibles no transitorios. Los discos, como se usan en el
 20 presente documento, incluyen el disco compacto (CD), el disco de láser, el disco óptico, el disco versátil digital (DVD), el disco flexible y el disco Blu-ray, donde algunos discos normalmente reproducen datos de manera magnética, mientras que otros discos reproducen los datos de manera óptica con láser. Las combinaciones de lo anterior también deberían incluirse dentro del alcance de los medios legibles por ordenador.

25 Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), formaciones lógicas programables sobre el terreno (FPGA) u otros circuitos lógicos equivalentes, integrados o discretos. Por consiguiente, el término "procesador", como se usa en el presente documento, puede referirse a cualquier estructura anterior o a cualquier otra estructura adecuada para la implementación de las técnicas descritas
 30 en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento puede proporcionarse dentro de módulos de hardware y/o software específicos configurados para la codificación y la decodificación, o incorporarse en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

35 Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluyendo un equipo de mano inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). Varios componentes, módulos o unidades se describen en esta divulgación para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente la realización mediante diferentes unidades de hardware. Más bien, como se ha descrito anteriormente, pueden
 40 combinarse diversas unidades en una unidad de hardware de códec, o ser proporcionadas por una colección de unidades de hardware inter-operativas, incluyendo uno o más procesadores, como se ha descrito anteriormente, conjuntamente con el software y/o firmware adecuado.

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

REIVINDICACIONES

1. Una unidad de procesamiento de gráficos que comprende un procesador sombreador (52), una o más unidades de hardware de función fija (54) y un procesador de comandos (56), en donde:
 - 5 el procesador sombreador (52) configurado para ejecutar una pluralidad de instrucciones en paralelo;

10 las una o más unidades de hardware de función fija (54), separadas del procesador sombreador, en el que las una o más unidades de hardware de función fija están configuradas para realizar una operación de representación de gráficos no realizada por el procesador sombreador; y

15 la unidad procesadora de comandos (56) configurada para recibir una o más tareas desde un procesador anfitrión y planificar de forma independiente una o más tareas, determinando de forma independiente cuándo ejecutar las tareas y/o qué recursos de la GPU son para ejecutar las tareas, en base a una operación de cada una de las una o más tareas y a una prioridad respectiva de las una o más tareas a ejecutar selectivamente mediante el procesador sombreador y las una o más unidades de hardware de función fija.
 2. La unidad de procesamiento de gráficos de la reivindicación 1, en la que el procesador de comandos está configurado para mantener una cola de comandos de las una o más tareas que se reciben, en la que la cola de comandos comprende una lista de tareas a ejecutar por la unidad de procesamiento de gráficos.
 3. La unidad de procesamiento de gráficos de la reivindicación 1, en la que el procesador de comandos está configurado para determinar la prioridad respectiva de las una o más tareas y planificar las una o más tareas en base a la prioridad respectiva de las una o más tareas.
 4. La unidad de procesamiento de gráficos de la reivindicación 1, en la que el procesador de comandos está configurado para recibir una indicación de la prioridad respectiva de las una o más tareas, desde un procesador anfitrión, y planificar las una o más tareas en base a la prioridad respectiva de las una o más tareas.
 5. La unidad de procesamiento de gráficos de la reivindicación 1, en la que el procesador de comandos está configurado para mantener una lista de tareas activas, en la que una tarea activa es una tarea que actualmente está siendo ejecutada por al menos uno entre el procesador sombreador y las una o más unidades de hardware de función fija.
 6. La unidad de procesamiento de gráficos de la reivindicación 1, en la que el procesador de comandos está configurado para detener la ejecución de una primera tarea de las una o más tareas y comenzar la ejecución de una segunda tarea de las una o más tareas.
 7. La unidad de procesamiento de gráficos de la reivindicación 1, que comprende además un motor de copia de memoria, configurado para copiar datos desde una memoria de la unidad de procesamiento de gráficos a una memoria del procesador sombreador.
 8. La unidad de procesamiento de gráficos de la reivindicación 7, en la que el procesador de comandos controla el funcionamiento del motor de copia de memoria.
 9. Un procedimiento para controlar una unidad de procesamiento de gráficos que comprende un procesador sombreador (52), una o más unidades de hardware de función fija (54) y un procesador de comandos (56), procedimiento que comprende:
 - 50 la recepción, mediante el procesador de comandos, de una o más tareas desde un procesador anfitrión; y
 - 55 la planificación, mediante el procesador de comandos, y de forma independiente del procesador anfitrión, de las una o más tareas a ejecutar selectivamente mediante un procesador sombreador, estando dicho procesador sombreador configurado para ejecutar una pluralidad de instrucciones en paralelo, y una o más unidades de función fija, determinando independientemente cuándo ejecutar las tareas y/o qué recursos de la GPU son para ejecutar las tareas, en base a una operación de cada una de las una o más tareas y a una prioridad respectiva de las una o más tareas, para ejecutarse de forma selectiva mediante el procesador sombreador y las una o más unidades de hardware de función fija, y en donde las una o más unidades de hardware de función fija, que están separadas del procesador sombreador y configuradas para realizar una operación de representación de gráficos no realizada por el procesador sombreador.
 - 65 10. El procedimiento de la reivindicación 9, que comprende además el mantenimiento de una cola de comandos de las una o más tareas que se reciben, en el que la cola de comandos comprende una lista de tareas a

ejecutar mediante la unidad de procesamiento de gráficos.

- 5
11. El procedimiento de la reivindicación 9, que comprende además el mantenimiento de una lista de tareas activas, en el que una tarea activa es una tarea que actualmente está siendo ejecutada por al menos uno de los procesadores sombreadores y las una o más unidades de hardware de función fija.
- 10
12. El procedimiento de la reivindicación 9, que comprende además la interrupción de la ejecución de una primera tarea de las una o más tareas y el inicio de la ejecución de una segunda tarea de las una o más tareas.
13. El procedimiento de la reivindicación 9, que comprende además la copia de datos desde una memoria de la unidad de procesamiento de gráficos hasta una memoria del procesador sombreador antes de la ejecución de las una o más tareas.
- 15
14. Un producto de programa informático que comprende un medio legible por ordenador que comprende instrucciones almacenadas en el mismo que, cuando se ejecutan, hacen que uno o más procesadores lleven a cabo el procedimiento de cualquiera de las reivindicaciones 9 a 13.

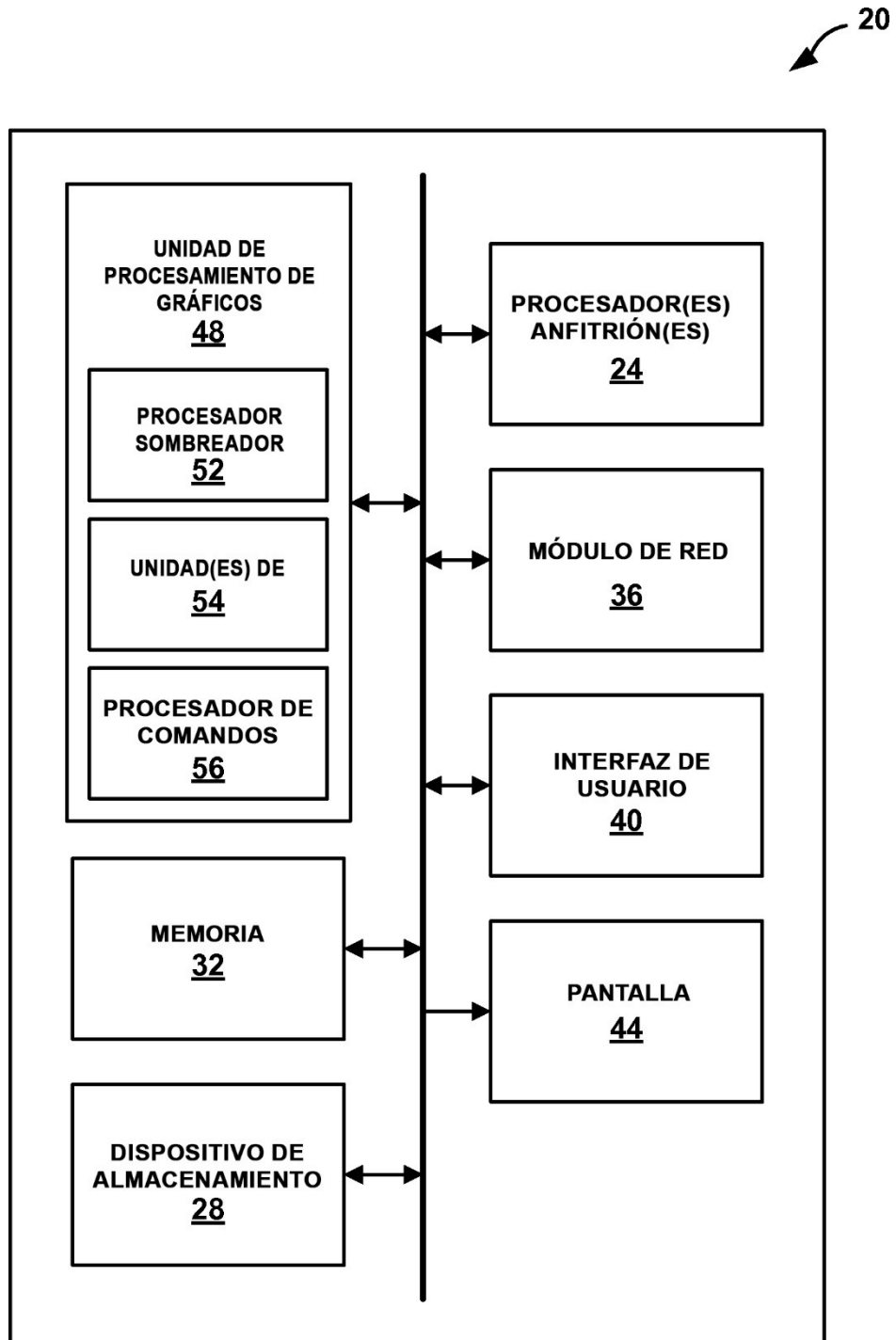


FIG. 1

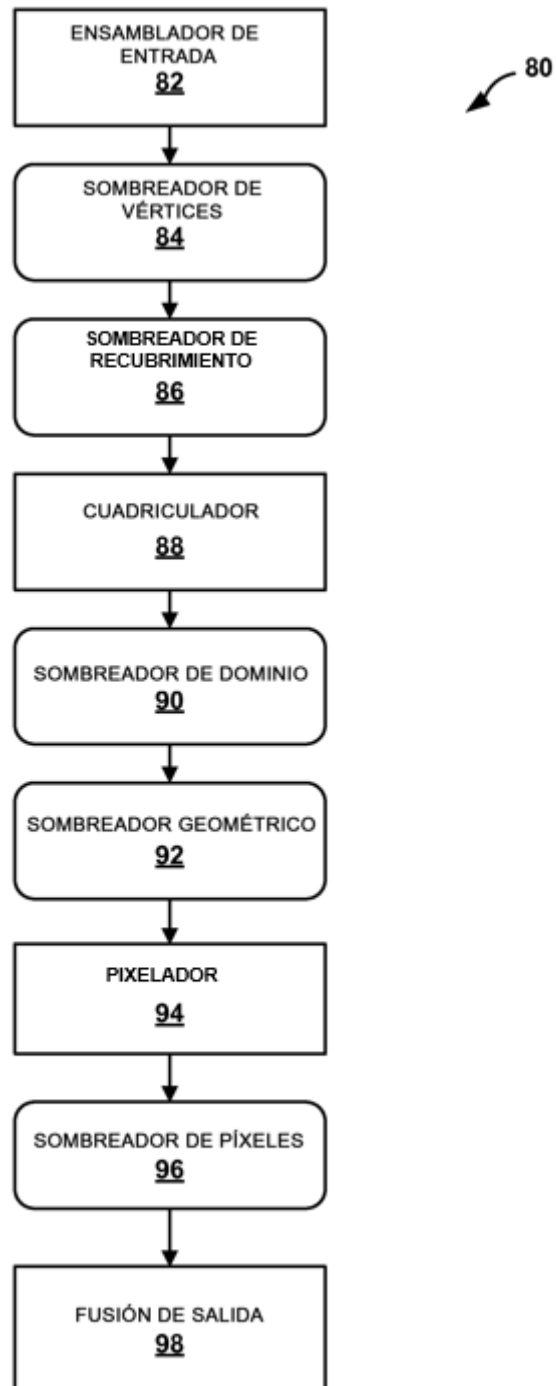


FIG. 2

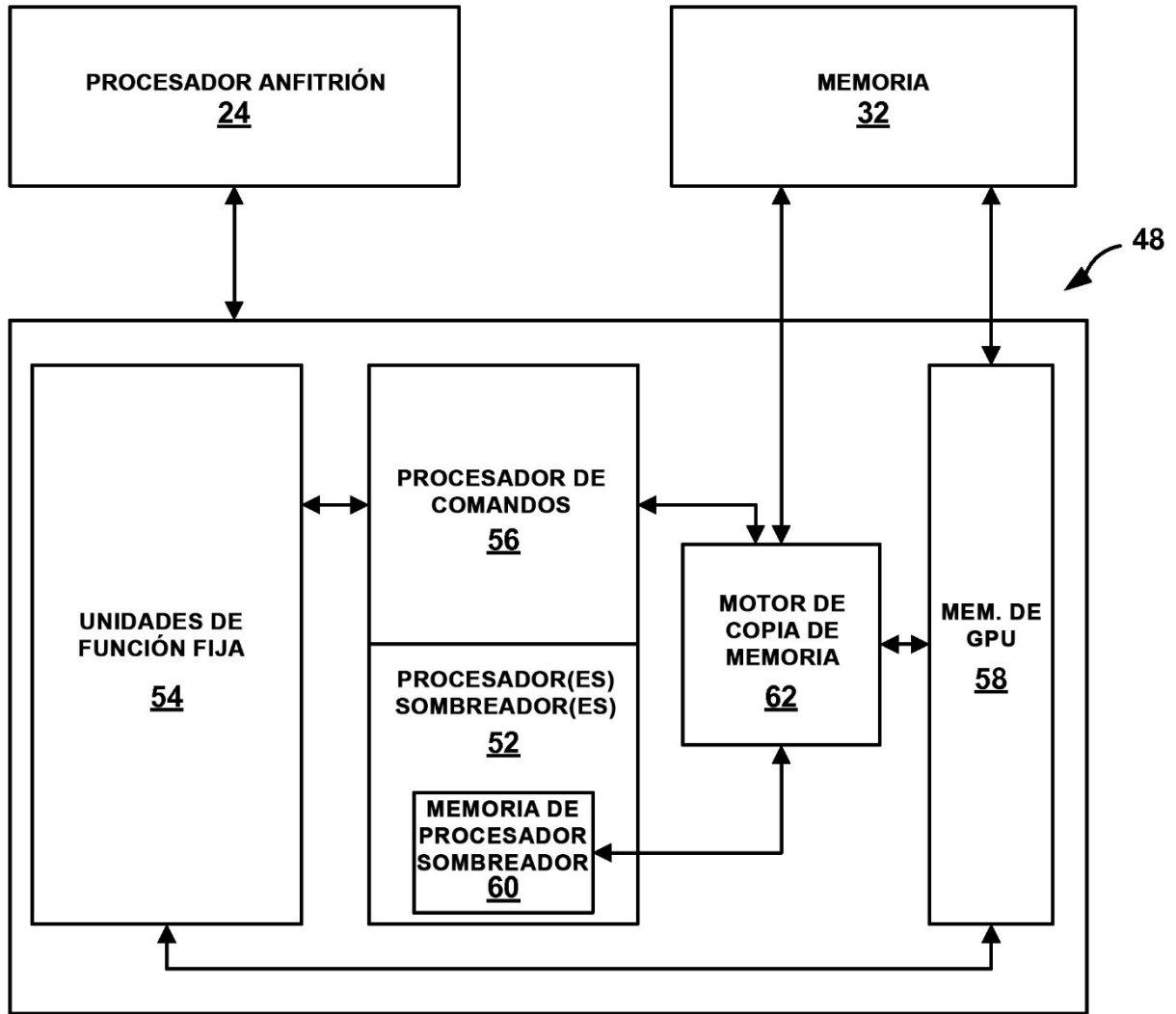


FIG. 3

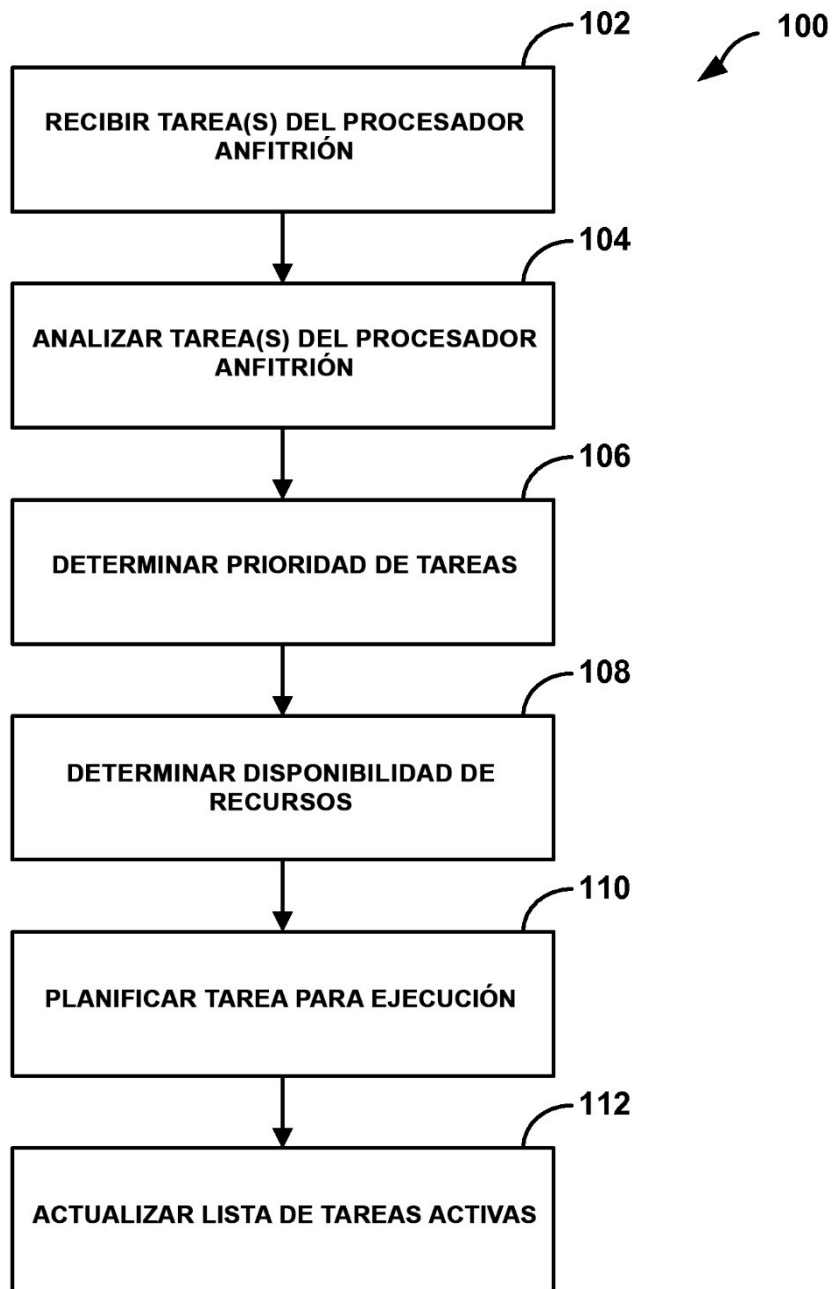


FIG. 4

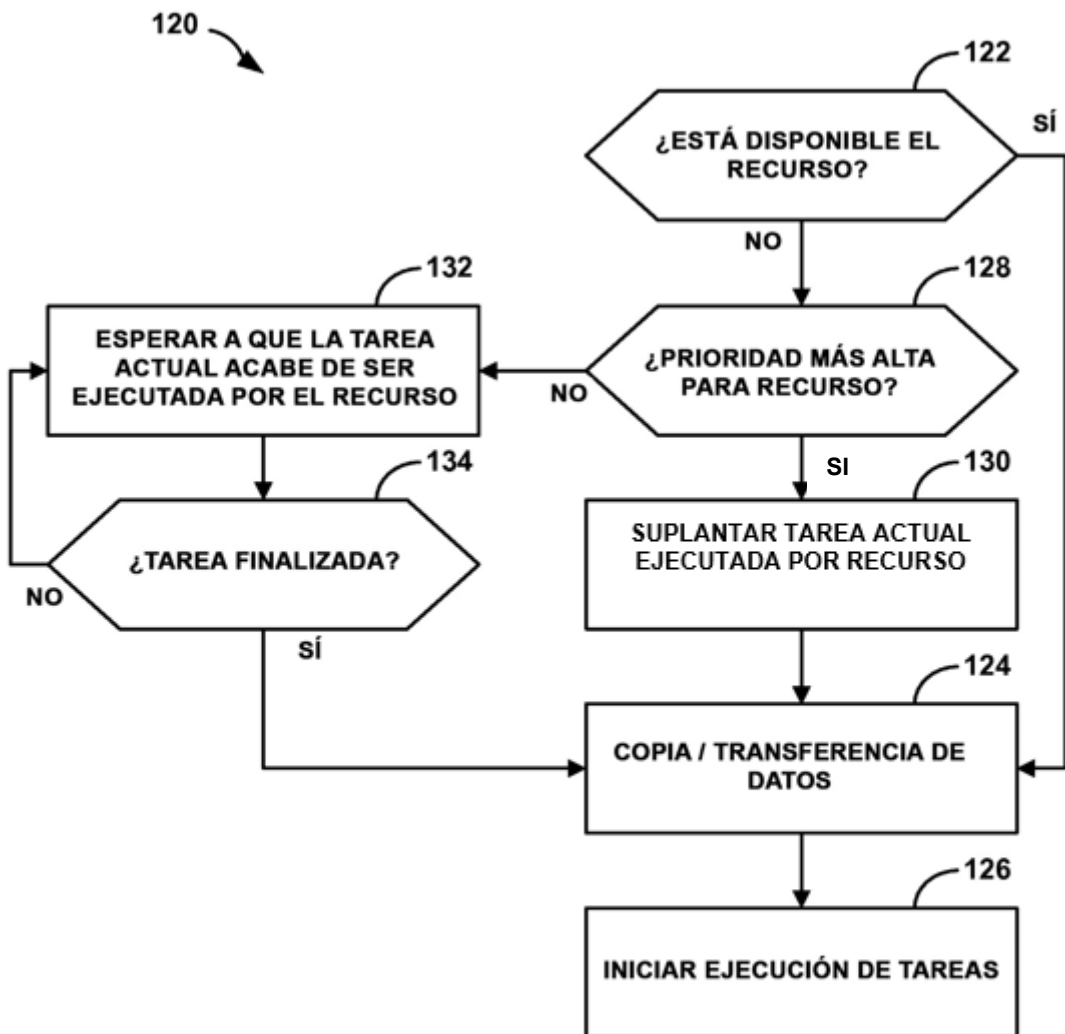


FIG. 5