

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 770 874**

51 Int. Cl.:

G06T 1/20 (2006.01)

G06T 1/60 (2006.01)

G06F 12/02 (2006.01)

G06T 11/40 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **01.08.2016 PCT/US2016/044975**

87 Fecha y número de publicación internacional: **23.03.2017 WO17048381**

96 Fecha de presentación y número de la solicitud europea: **01.08.2016 E 16753760 (4)**

97 Fecha y número de publicación de la concesión europea: **06.11.2019 EP 3350766**

54 Título: **Almacenamiento de datos gráficos comprimidos en ancho de banda**

30 Prioridad:

17.09.2015 US 201514857303

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

03.07.2020

73 Titular/es:

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

72 Inventor/es:

**GRUBER, ANDREW EVAN;
HILL, REXFORD ALAN y
KHANDELWAL, SHAMBHOO**

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 770 874 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

Almacenamiento de datos gráficos comprimidos en ancho de banda

5 CAMPO TÉCNICO

[0001] Esta divulgación se refiere al almacenamiento de datos, y más específicamente, al almacenamiento en memoria de datos gráficos comprimidos en ancho de banda.

10 ANTECEDENTES

[0002] Un dispositivo que proporciona contenido para la presentación visual en un dispositivo de visualización electrónico incluye, en general, una unidad de procesamiento gráfico (GPU). La GPU renderiza píxeles que son representativos del contenido en un dispositivo de visualización. La GPU genera uno o más valores de píxel para cada píxel en el dispositivo de visualización y realiza un procesamiento de gráficos en los valores de píxel para cada píxel en el dispositivo de visualización para renderizar cada píxel para su presentación, que realiza el sombreado de fragmentos de los fragmentos generados por la etapa de rasterización.

[0003] El documento de Akenine-Moller *et al.* "Graphics Processing Units for Handhelds", Actas del IEEE, Nueva York, EE. UU., vol. 96, n.º 5, mayo de 2008, páginas 779-789, ISSN: 0018-9219), una revisión de las arquitecturas de renderización modernas para dispositivos portátiles, analiza la renderización en dispositivos móviles y los procedimientos y arquitecturas más comunes usados por razones de eficacia, incluida la renderización basada en mosaicos y la compresión de memoria intermedia. El documento US 8 378 859 B2 divulga un procedimiento para la compresión de imágenes de propósito general que divide la imagen en mosaicos comprimidos, sin pérdidas y de manera individual. Los mosaicos se almacenan en bloques de memoria de tamaño uniforme y las partes de bloque no usadas se usan para aquellos mosaicos que requieran más memoria de la asignada.

SUMARIO

[0004] Las técnicas de esta divulgación se refieren, en general, a técnicas para almacenar en memoria una pluralidad de datos gráficos comprimidos en ancho de banda junto con datos adicionales que están asociados a la pluralidad de datos gráficos comprimidos en ancho de banda. La pluralidad de datos gráficos comprimidos en ancho de banda puede variar en tamaño, y la pluralidad de datos gráficos comprimidos en ancho de banda se almacena en bloques de tamaño uniforme en la memoria que pueden incorporar los datos gráficos más grandes comprimidos en ancho de banda de la pluralidad de gráficos comprimidos en ancho de banda datos. Por lo tanto, almacenar la pluralidad de datos gráficos comprimidos en ancho de banda en los bloques de tamaño uniforme de memoria puede dar como resultado un espacio restante no usado en algunos de los bloques de memoria que almacenan la pluralidad de datos gráficos comprimidos en ancho de banda. Dicho espacio no usado en algunos de los bloques de memoria se utiliza para almacenar datos adicionales que están asociados a la pluralidad de datos gráficos comprimidos en ancho de banda, tales como los datos de profundidad asociados a la pluralidad de datos gráficos comprimidos en ancho de banda o códigos *hash* que identifican cada uno de la pluralidad de datos gráficos comprimidos en ancho de banda.

[0005] En un ejemplo de la divulgación, un procedimiento para el procesamiento de gráficos puede incluir almacenar, por al menos un procesador, una pluralidad de datos gráficos comprimidos en ancho de banda en una pluralidad respectiva de bloques en memoria, donde cada bloque de la pluralidad de bloques tiene un tamaño fijo uniforme en la memoria, y donde uno o más de la pluralidad de datos gráficos comprimidos en ancho de banda tiene un tamaño que es menor que el tamaño fijo. El proceso incluye además almacenar, mediante el al menos un procesador, datos asociados a la pluralidad de datos gráficos comprimidos en ancho de banda en el espacio no usado de uno o más de la pluralidad de bloques que contiene el uno o más datos respectivos de la pluralidad de datos gráficos comprimidos en ancho de banda.

[0006] En otro ejemplo de la divulgación, un aparato configurado para procesar datos gráficos puede incluir memoria. El aparato puede incluir además al menos un procesador configurado para: almacenar una pluralidad de datos gráficos comprimidos en ancho de banda en una pluralidad respectiva de bloques en la memoria, donde cada bloque de la pluralidad de bloques tiene un tamaño fijo uniforme en la memoria, y donde uno o más de la pluralidad de datos gráficos comprimidos en ancho de banda tiene un tamaño que es menor que el tamaño fijo; y almacenar datos asociados a la pluralidad de datos gráficos comprimidos en ancho de banda en el espacio no usado de uno o más de la pluralidad de bloques que contiene el uno o más datos respectivos de la pluralidad de datos gráficos comprimidos en ancho de banda.

[0007] En otro ejemplo de la divulgación, un aparato puede incluir medios para almacenar una pluralidad de datos gráficos comprimidos en ancho de banda en una pluralidad respectiva de bloques en memoria, donde cada bloque de la pluralidad de bloques tiene un tamaño fijo uniforme en la memoria, y donde uno o más de la pluralidad de datos gráficos comprimidos en ancho de banda tiene un tamaño que es menor que el tamaño fijo. El aparato puede incluir además medios para almacenar datos asociados a la pluralidad de datos gráficos comprimidos en ancho de

banda en el espacio no usado de uno o más de la pluralidad de bloques que contiene el uno o más datos respectivos de la pluralidad de datos gráficos comprimidos en ancho de banda.

5 [0008] Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y en la descripción siguiente. Otras características, objetos y ventajas resultarán evidentes a partir de la descripción y de los dibujos, y a partir de las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

10 [0009]

15 La FIG. 1 es un diagrama de bloques que ilustra un dispositivo informático de ejemplo que se puede configurar para implementar uno o más aspectos de esta divulgación para almacenar en memoria datos gráficos comprimidos en ancho de banda.

La FIG. 2 es un diagrama de bloques que ilustra con más detalle implementaciones de ejemplo de la CPU, de la GPU y de la memoria de sistema de la FIG. 1.

20 Las FIGS. 3A-3F son diagramas conceptuales que ilustran técnicas de ejemplo para almacenar en memoria datos gráficos comprimidos en ancho de banda.

25 La FIG. 4 es un diagrama de flujo que ilustra un proceso de ejemplo para almacenar en memoria datos gráficos comprimidos en ancho de banda.

DESCRIPCIÓN DETALLADA

30 [0010] Los datos gráficos comprimidos en ancho de banda son datos gráficos que se comprimen para que puedan transferirse más rápidamente a través de los buses de un dispositivo informático. Puesto que una unidad de procesamiento gráfico (GPU) de un dispositivo informático realiza operaciones de procesamiento gráfico en datos gráficos, como una superficie, el dispositivo informático puede transferir la superficie a través de un bus entre la GPU y la memoria o entre diferentes memorias. Por ejemplo, el dispositivo informático puede realizar una operación de composición que combina dos superficies diferentes transfiriendo esas dos superficies desde la memoria a la GPU para realizar la operación de composición, y transfiriendo la superficie compuesta resultante desde la GPU a la memoria. Por lo tanto, al reducir el tamaño de la superficie mediante compresión, el dispositivo informático puede transferir la superficie más rápidamente entre los componentes del dispositivo informático, mejorando así el rendimiento del dispositivo informático.

40 [0011] El dispositivo informático puede realizar la compresión del ancho de banda de una superficie dividiendo la superficie en subregiones y comprimiendo cada una de las subregiones de la superficie para generar una pluralidad de datos gráficos comprimidos en ancho de banda. La pluralidad de datos gráficos comprimidos en ancho de banda puede variar de tamaño debido a diferencias en el contenido entre subregiones de la superficie. Por ejemplo, un dispositivo informático puede comprimir una subregión de la superficie que contiene de manera uniforme píxeles de un solo color en un tamaño relativamente más pequeño que otra subregión de la superficie que contiene píxeles de muchos colores diferentes.

50 [0012] El dispositivo informático puede almacenar la pluralidad de datos gráficos comprimidos en ancho de banda en una pluralidad de bloques de tamaño uniforme que el dispositivo informático asigna en memoria. Cada uno de los bloques es lo suficientemente grande como para contener el dato más grande de la pluralidad de datos gráficos comprimidos en ancho de banda. Debido a que cada bloque de la pluralidad de bloques tiene el mismo tamaño, mientras que la pluralidad de datos gráficos comprimidos en ancho de banda puede variar de tamaño, el almacenamiento de la pluralidad de datos gráficos comprimidos en ancho de banda en la pluralidad de bloques puede dar como resultado que uno o más de los bloques tenga un espacio no usado que no esté ocupado por los respectivos datos gráficos comprimidos en ancho de banda almacenados en el bloque.

55 [0013] De acuerdo con la invención reivindicada, el dispositivo informático almacena otros datos asociados a la pluralidad de datos gráficos comprimidos en ancho de banda en el espacio no usado del uno o más de los bloques. Por ejemplo, en lugar de almacenar datos de profundidad asociados a la pluralidad de datos gráficos comprimidos en ancho de banda en un área separada (por ejemplo, bloque) en la memoria, el dispositivo informático puede almacenar dichos datos de profundidad en el espacio no usado del uno o más de los bloques. De manera similar, el dispositivo informático puede almacenar códigos *hash* que identifican cada dato de la pluralidad de datos gráficos comprimidos en ancho de banda en el espacio no usado del uno o más de los bloques. De esta manera, el dispositivo informático puede utilizar el espacio no usado de la pluralidad de bloques para almacenar datos adicionales asociados a la pluralidad de datos gráficos comprimidos en ancho de banda, aumentando así la eficiencia de utilización de memoria del dispositivo informático.

[0014] Los otros datos que el dispositivo informático puede almacenar en el espacio no usado del uno o más bloques pueden ser superficies de optimización, ya que el dispositivo informático puede usar dichos datos para optimizar el rendimiento de las operaciones gráficas en los datos gráficos. Por ejemplo, el dispositivo informático puede utilizar los datos de profundidad para aumentar su rendimiento al renderizar los datos gráficos asociados, mientras que el dispositivo informático puede utilizar los códigos *hash* para aumentar su rendimiento en determinadas operaciones gráficas en los datos gráficos. De este modo, el dispositivo informático puede almacenar cualquier cantidad de datos adicionales que no sean datos de profundidad o códigos *hash* en el espacio no usado del uno o más bloques, incluso almacenar superficies de optimización adicionales que se pueden usar para optimizar la renderización de los datos gráficos.

[0015] La FIG. 1 es un diagrama de bloques que ilustra un dispositivo informático de ejemplo que se puede configurar para implementar uno o más aspectos de esta divulgación para almacenar en memoria datos gráficos comprimidos en ancho de banda. Como se muestra en la FIG. 1, el dispositivo 2 puede ser un dispositivo informático que incluye, pero sin limitarse a, dispositivos de vídeo, reproductores multimedia, decodificadores, teléfonos inalámbricos tales como teléfonos móviles y los denominados teléfonos inteligentes, asistentes digitales personales (PDA), ordenadores de escritorio, ordenadores portátiles, consolas de videojuegos, unidades de videoconferencia, dispositivos informáticos de tipo tableta y similares. En el ejemplo de la FIG. 1, el dispositivo 2 puede incluir una unidad central de procesamiento (CPU) 6, una memoria de sistema 10 y una GPU 12. El dispositivo 2 también puede incluir un procesador de visualización 14, un módulo transceptor 3, una interfaz de usuario 4 y un dispositivo de visualización 8. Tanto el módulo transceptor 3 como el procesador de visualización 14 pueden formar parte del mismo circuito integrado (IC) que la CPU 6 y/o la GPU 12, pueden ser externos al IC o a los IC que incluyen la CPU 6 y/o la GPU 12, o se pueden formar en un IC que sea externo al IC que incluye la CPU 6 y/o la GPU 12.

[0016] El dispositivo 2 puede incluir módulos o unidades adicionales que no se muestran en la FIG. 1 para una mayor claridad. Por ejemplo, el dispositivo 2 puede incluir un altavoz y un micrófono, ninguno de los cuales se muestra en la FIG. 1, para efectuar comunicaciones telefónicas en ejemplos en los que el dispositivo 2 es un teléfono móvil inalámbrico, o un altavoz donde el dispositivo 2 es un reproductor multimedia. El dispositivo 2 también puede incluir una cámara de vídeo. Además, los diversos módulos y unidades mostrados en el dispositivo 2 pueden no ser necesarios en cada ejemplo del dispositivo 2. Por ejemplo, la interfaz de usuario 4 y el dispositivo de visualización 8 pueden ser externos al dispositivo 2 en ejemplos en los que el dispositivo 2 es un ordenador de escritorio u otro dispositivo que esté equipado para interconectarse con una interfaz de usuario o dispositivo de visualización externos.

[0017] Ejemplos de interfaz de usuario 4 incluyen, pero no se limitan a, una rueda de desplazamiento, un ratón, un teclado y otros tipos de dispositivos de entrada. La interfaz de usuario 4 también puede ser una pantalla táctil y se puede incorporar como parte de un dispositivo de visualización 8. El módulo transceptor 3 puede incluir circuitos para permitir la comunicación inalámbrica o alámbrica entre el dispositivo informático 2 y otro dispositivo o una red. El módulo transceptor 3 puede incluir moduladores, desmoduladores, amplificadores y otros circuitos de este tipo para la comunicación alámbrica o inalámbrica.

[0018] La CPU 6 puede ser un microprocesador, tal como una unidad central de procesamiento (CPU) configurada para procesar instrucciones de un programa informático para su ejecución. La CPU 6 puede comprender un procesador de propósito general o de propósito especial que controle el funcionamiento del dispositivo informático 2. Un usuario puede proporcionar datos de entrada al dispositivo informático 2 para hacer que la CPU 6 ejecute una o más aplicaciones de software. Las aplicaciones de software que se ejecutan en la CPU 6 pueden incluir, por ejemplo, un sistema operativo, una aplicación de procesador de textos, una aplicación de correo electrónico, una aplicación de hoja de cálculo, una aplicación de reproductor multimedia, una aplicación de videojuegos, una aplicación de interfaz gráfica de usuario u otro programa. Adicionalmente, la CPU 6 puede ejecutar el controlador de GPU 22 para controlar el funcionamiento de la GPU 12. El usuario puede proporcionar datos de entrada al dispositivo informático 2 por medio de uno o más dispositivos de entrada (no mostrados) tales como un teclado, un ratón, un micrófono, un panel táctil u otro dispositivo de entrada que esté acoplado al dispositivo informático 2 por medio de la interfaz de usuario 4.

[0019] Las aplicaciones de software que se ejecutan en la CPU 6 pueden incluir una o más instrucciones de renderización de gráficos que indiquen al procesador 6 que inicie la renderización de datos de gráficos en el dispositivo de visualización 8. En algunos ejemplos, las instrucciones de software se pueden ajustar a una interfaz de programación de aplicaciones gráficas (API), tal como, por ejemplo, una API de Librería Abierta de Gráficos (OpenGL®), una API de Sistemas Integrados de Librería Abierta de Gráficos (OpenGL ES), una API Direct3D, una API X3D, una API RenderMan, una API WebGL o cualquier otra API gráfica estándar pública o propietaria.

[0020] Para procesar las instrucciones de renderización gráfica de las aplicaciones de software, la CPU 6 puede emitir uno o más comandos de renderización gráfica a la GPU 12 (por ejemplo, a través del controlador de GPU 22) para hacer que la GPU 12 realice una parte o la totalidad de la renderización de los datos gráficos. En algunos ejemplos, los datos gráficos a renderizar pueden incluir una lista de primitivas gráficas, por ejemplo, puntos, líneas, triángulos, cuadriláteros, tiras de triángulos, etc.

[0021] La GPU 12 se puede configurar para realizar operaciones gráficas para renderizar una o más primitivas gráficas en el dispositivo de visualización 8. Por tanto, cuando una de las aplicaciones de software que se ejecuta en la CPU 6 requiere procesamiento gráfico, la CPU 6 puede proporcionar comandos gráficos y datos gráficos a la GPU 12 para que su renderización en el dispositivo de visualización 8. Los datos gráficos pueden incluir, por ejemplo, comandos de representación, información de estado, información de primitivas, información de textura, etc. La GPU 12 se puede construir, en algunos casos, con una estructura altamente paralela que proporcione un procesamiento más eficaz de operaciones complejas relacionadas con gráficos que la CPU 6. Por ejemplo, la GPU 12 puede incluir una pluralidad de elementos de procesamiento, tales como unidades de sombreado, que están configurados para funcionar en múltiples vértices o píxeles de manera paralela. La naturaleza altamente paralela de la GPU 12 puede, en algunos casos, permitir que la GPU 12 represente imágenes gráficas (por ejemplo, GUI y escenas gráficas bidimensionales (2D) y/o tridimensionales (3D)) en el dispositivo de visualización 8 más rápidamente que representando las escenas directamente en el dispositivo de visualización 8 usando la CPU 6.

[0022] La GPU 12, en algunos casos, se puede integrar en una placa base del dispositivo informático 2. En otros casos, la GPU 12 puede estar presente en una tarjeta gráfica que esté instalada en un puerto en la placa base del dispositivo informático 2 o se puede incorporar de otro modo dentro de un dispositivo periférico configurado para interfundar con el dispositivo informático 2. La GPU 12 puede incluir uno o más procesadores, tales como uno o más microprocesadores, circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables *in situ* (FPGA), procesadores de señales digitales (DSP) u otro circuito lógico integrado o discreto equivalente. La GPU 12 también puede incluir uno o más núcleos de procesador, de modo que la GPU 12 se puede denominar procesador multinúcleo.

[0023] La GPU 12 se puede acoplar directamente a la memoria gráfica 40. Por tanto, la GPU 12 puede leer datos de y escribir datos en la memoria gráfica 40 sin usar un bus. En otras palabras, la GPU 12 puede procesar datos usando localmente un almacenamiento local, en lugar de una memoria fuera de chip. Dicha memoria gráfica 40 se puede denominar memoria en chip. Esto permite que la GPU 12 funcione de una manera más eficaz al eliminar la necesidad de que la GPU 12 lea y escriba datos por medio de un bus, que puede experimentar un tráfico pesado. Sin embargo, en algunos casos, la GPU 12 puede no incluir una memoria separada, sino que utiliza la memoria de sistema 10 por medio de un bus. La memoria gráfica 40 puede incluir una o más memorias volátiles o no volátiles o dispositivos de almacenamiento, tales como, por ejemplo, memoria de acceso aleatorio (RAM), RAM estática (SRAM), RAM dinámica (DRAM), ROM programable y borrable (EPROM), ROM programable y borrable eléctricamente (EEPROM), memoria Flash, un medio de datos magnético o un medio de almacenamiento óptico.

[0024] En algunos ejemplos, la GPU 12 puede almacenar una imagen completamente formada en la memoria de sistema 10, donde la imagen puede ser una o más superficies. Una superficie, en algunos ejemplos, puede ser un bloque bidimensional de píxeles, donde cada uno de los píxeles puede tener un valor de color. A lo largo de esta divulgación, el término datos gráficos puede, en un ejemplo no limitativo, incluir superficies o partes de superficies. El procesador de visualización 14 puede recuperar la imagen de la memoria de sistema 10 y proporcionar valores que hacen que los píxeles del dispositivo de visualización 8 se iluminen para visualizar la imagen. El dispositivo de visualización 8 puede ser el dispositivo de visualización del dispositivo informático 2 que visualiza el contenido de imagen generado por la GPU 12. El dispositivo de visualización 8 puede ser una pantalla de cristal líquido (LCD), una pantalla de diodos orgánicos emisores de luz (OLED), una pantalla de tubo de rayos catódicos (CRT), una pantalla de plasma u otro tipo de dispositivo de visualización.

[0025] De acuerdo con aspectos de la presente divulgación, el dispositivo informático 2 puede asignar una pluralidad de bloques en memoria, tal como la memoria de sistema 10 o la memoria gráfica 40, donde cada bloque de la pluralidad de bloques tiene un tamaño fijo uniforme en la memoria. El dispositivo informático 2 puede almacenar además una pluralidad de datos gráficos comprimidos en ancho de banda en la pluralidad respectiva de bloques en la memoria, donde uno o más datos de la pluralidad de datos gráficos comprimidos en ancho de banda tiene un tamaño que es menor que el tamaño fijo. El dispositivo informático 2 puede almacenar además datos asociados a la pluralidad de datos gráficos comprimidos en ancho de banda en el espacio no usado de uno o más bloques de la pluralidad de bloques que contiene el uno o más datos respectivos de la pluralidad de datos gráficos comprimidos en ancho de banda.

[0026] La FIG. 2 es un diagrama de bloques que ilustra con mayor detalle implementaciones de ejemplo de la CPU 6, de la GPU 12 y de la memoria de sistema 10 de la FIG. 1. Como se muestra en la FIG. 2, la CPU 6 puede incluir al menos una aplicación de software 18, una API gráfica 20 y un controlador de GPU 22, cada uno de los cuales puede ser una o más aplicaciones de software o servicios que se ejecutan en la CPU 6.

[0027] La memoria disponible para la CPU 6 y la GPU 12 puede incluir una memoria de sistema 10, una memoria intermedia de cuadros 16 y objetivos de renderización 24. La memoria intermedia de cuadros 16 puede formar parte de la memoria de sistema 10 o puede estar separada de la memoria de sistema 10 y puede almacenar datos de imágenes renderizadas. La GPU 12 también puede renderizar datos de imagen para su almacenamiento en objetivos de renderización 24. De manera similar a la memoria intermedia de cuadros 16, los objetivos de renderización 24 pueden formar parte de la memoria de sistema 10 o pueden estar separados de la memoria de sistema 10.

- 5 **[0028]** La aplicación de software 18 puede ser cualquier aplicación que utilice la funcionalidad de la GPU 12. Por ejemplo, la aplicación de software 18 puede ser una aplicación de GUI, un sistema operativo, una aplicación de correlación portátil, un programa de diseño asistido por ordenador para aplicaciones de ingeniería o artísticas, una aplicación de videojuegos u otro tipo de aplicación de software que use gráficos 2D o 3D.
- 10 **[0029]** La aplicación de software 18 puede incluir una o más instrucciones de representación que indiquen a la GPU 12 que renderice una interfaz gráfica de usuario (GUI) y/o una escena gráfica. Por ejemplo, las instrucciones de representación pueden incluir instrucciones que definan un conjunto de una o más primitivas gráficas a renderizar por la GPU 12. En algunos ejemplos, las instrucciones de representación pueden, conjuntamente, definir todo o parte de una pluralidad de superficies de ventanas usadas en una GUI. En ejemplos adicionales, las instrucciones de representación pueden, conjuntamente, definir todo o parte de una escena gráfica que incluya uno o más objetos gráficos dentro de un espacio modelo o espacio mundial definido por la aplicación.
- 15 **[0030]** La aplicación de software 18 puede invocar al controlador de GPU 22, por medio de la API gráfica 20, para emitir uno o más comandos a la GPU 12 para renderizar una o más primitivas gráficas en imágenes gráficas visualizables. Por ejemplo, la aplicación de software 18 puede invocar al controlador de GPU 22, por medio de la API gráfica 20, para proporcionar definiciones de primitivas a la GPU 12. En algunos casos, las definiciones de primitivas se pueden proporcionar a la GPU 12 en forma de una lista de primitivas de representación, por ejemplo, triángulos, rectángulos, abanicos de triángulos, tiras de triángulos, etc. Las definiciones de primitivas pueden incluir especificaciones de vértices que especifiquen uno o más vértices asociados a las primitivas que se vayan a renderizar. Las especificaciones de vértices pueden incluir coordenadas de posición para cada vértice y, en algunos casos, otros atributos asociados al vértice, tales como, por ejemplo, coordenadas de color, vectores normales y coordenadas de textura. Las definiciones de primitivas también pueden incluir información de tipo de primitiva (por ejemplo, triángulo, rectángulo, abanico de triángulos, tira de triángulos, etc.), información de escalamiento, información de rotación y similares. En base a las instrucciones emitidas por la aplicación de software 18 para el controlador de GPU 22, el controlador de GPU 22 puede formular uno o más comandos que especifiquen una o más operaciones para que la GPU 12 la(s) realice para renderizar la primitiva. Cuando la GPU 12 recibe un comando desde la CPU 6, un grupo de procesadores 46 puede ejecutar una cadena de procesamiento gráfico para descodificar el comando y puede configurar la cadena de procesamiento gráfico para realizar la operación especificada en el comando. Por ejemplo, un motor de comandos de la cadena de procesamiento gráfico puede leer datos de primitivas y ensamblar los datos en primitivas para su uso por las otras fases de la cadena de gráficos de la cadena de procesamiento gráfico. Después de realizar las operaciones especificadas, la GPU 12 proporciona los datos renderizados a la memoria intermedia de cuadros 16 asociado a un dispositivo de visualización o a uno de los objetivos de renderización 24.
- 20 **[0031]** La memoria intermedia de cuadros 16 almacena píxeles de destino para la GPU 12. Cada píxel de destino puede estar asociado a una ubicación única de píxel en pantalla. En algunos ejemplos, la memoria intermedia de cuadros 16 puede almacenar componentes de color y un valor alfa de destino para cada píxel de destino. Por ejemplo, la memoria intermedia de cuadros 16 puede almacenar componentes Rojo, Verde, Azul, Alfa (RGBA) para cada píxel, donde los componentes "RGB" corresponden a valores de color y el componente "A" corresponde a un valor alfa de destino. La memoria intermedia de cuadros 16 también puede almacenar valores de profundidad para cada píxel de destino. De esta manera, se puede decir que la memoria intermedia de cuadros 16 almacena datos gráficos (por ejemplo, una superficie). Aunque la memoria intermedia de cuadros 16 y la memoria de sistema 10 se ilustran como unidades de memoria separadas, en otros ejemplos, la memoria intermedia de cuadros 16 puede formar parte de la memoria de sistema 10. Una vez que la GPU 12 ha renderizado todos los píxeles de un cuadro en la memoria intermedia de cuadros 16, la memoria intermedia de cuadros puede proporcionar el cuadro terminado al dispositivo de visualización 8 para su visualización.
- 25 **[0032]** De manera similar a la memoria intermedia de cuadros 16, cada uno de los objetivos de renderización 24 también puede almacenar píxeles de destino para la GPU 12, incluidos valores de color y/o valores de profundidad para los píxeles. Cada uno de los objetivos de renderización 24 puede almacenar información para el mismo número de ubicaciones únicas de píxel que la memoria intermedia de cuadros 16 o puede almacenar un subconjunto del número de ubicaciones únicas de píxel que la memoria intermedia de cuadros 16.
- 30 **[0033]** El grupo de procesadores 46 puede incluir una o más unidades de procesamiento programables 42 y/o una o más unidades de procesamiento de función fija 44. La unidad de procesamiento programable 42 puede incluir, por ejemplo, unidades de sombreado programables que estén configuradas para ejecutar uno o más programas de sombreado que se descarguen en la GPU 12 desde la CPU 6. En algunos ejemplos, las unidades de procesamiento programables 42 se pueden denominar "procesadores de sombreado" o "sombreadores unificados", y pueden realizar operaciones de geometría, vértice, píxel u otras operaciones de sombreado para renderizar gráficos. Cada unidad de sombreado puede incluir uno o más componentes para buscar y descodificar operaciones, una o más ALU para llevar a cabo cálculos aritméticos, una o más memorias, memorias caché y registros.
- 35 **[0034]** La GPU 12 puede designar unidades de procesamiento programables 42 para realizar una variedad de operaciones de sombreado, tales como sombreado de vértices, sombreado de casco, sombreado de dominio,

sombreado de geometría, sombreado de fragmentos y similares, enviando comandos a las unidades de procesamiento programables 42 para ejecutar una o más de una fase de sombreado de vértice, una fase de sombreado de teselado, una fase de sombreado de geometría, una fase de rasterización y una fase de sombreado de fragmentos en una cadena de procesamiento gráfico. En algunos ejemplos, el controlador de GPU 22 puede hacer que un compilador que se ejecute en un CPU 6 compile uno o más programas de sombreado, y descargar los programas de sombreado compilados en unidades de procesamiento programables 42 contenidas en la GPU 12. Los programas de sombreado se pueden escribir en un lenguaje de sombreado de alto nivel, tal como, por ejemplo, un Lenguaje de Sombreado OpenGL (GLSL), un Lenguaje de Sombreado de Alto Nivel (HLSL), un lenguaje de sombreado C para Gráficos (Cg), un kernel C OpenCL etc. Los programas de sombreado compilados pueden incluir una o más instrucciones que controlen el funcionamiento de las unidades de procesamiento programables 42 dentro de la GPU 12. Por ejemplo, los programas de sombreado pueden incluir programas de sombreado de vértices que se pueden ejecutar por unidades de procesamiento programables 42 para realizar las funciones de la fase de sombreado de vértices, programas de sombreado de teselado que se pueden ejecutar por las unidades de procesamiento programables 42 para realizar las funciones de las fases de teselado, programas de sombreado de geometría que se pueden ejecutar por unidades de procesamiento programables 42 para realizar las funciones de la fase de sombreado de geometría y/o programas de sombreado de fragmentos que se pueden ejecutar por unidades de procesamiento programables 42 para realizar las funciones de la fase de sombreado de fragmentos. Un programa de sombreado de vértices puede controlar la ejecución de una unidad de sombreado de vértices programable o de una unidad de sombreado unificada e incluir instrucciones que especifiquen una o más operaciones por vértice.

[0035] El grupo de procesadores 46 también puede incluir unidades de procesamiento de función fija 44. Las unidades de procesamiento de función fija 44 pueden incluir hardware que está cableado para realizar determinadas funciones. Aunque las unidades de procesamiento de función fija 44 pueden configurarse, por medio de una o más señales de control, por ejemplo, para realizar diferentes funciones, el hardware de función fija no incluye típicamente una memoria de programa que sea capaz de recibir programas compilados por el usuario. En algunos ejemplos, las unidades de procesamiento de función fija 44 en el grupo de procesadores 46 pueden incluir, por ejemplo, unidades de procesamiento que realizan operaciones de rasterización, tales como, por ejemplo, prueba de profundidad, prueba de tijera, mezcla alfa, prueba de profundidad de baja resolución, etc. para realizar las funciones de la fase de rasterización de la cadena de procesamiento gráfico.

[0036] La memoria gráfica 40 es almacenamiento o memoria en chip que se integró físicamente en el circuito integrado de la GPU 12. En algunos casos, debido a que la memoria gráfica 40 está en chip, la GPU 12 puede leer o escribir valores en la memoria gráfica 40 más rápidamente que leer o escribir valores en la memoria de sistema 10 a través de un bus de sistema.

[0037] En algunos ejemplos, la GPU 12 puede funcionar de acuerdo con un modo de renderización diferida (también denominada renderización por agrupación o renderización basada en mosaicos) para renderizar datos gráficos. Durante un funcionamiento de acuerdo con el modo de renderización diferida, el grupo de procesadores 46 dentro de la GPU 12 realiza primero una fase de agrupación (también conocida como fase de mosaico) para dividir un cuadro en una pluralidad de mosaicos y determinar qué primitivas hay dentro de cada mosaico. En algunos ejemplos, la fase de agrupación puede indicar si una primitiva está o no dentro de un mosaico. En otros ejemplos, la fase de agrupación también puede incluir una prueba de profundidad e indicar si una primitiva en particular es visible o no en un mosaico renderizado. Para cada mosaico de la pluralidad de mosaicos, el grupo de procesadores 46 renderiza datos gráficos (valores de color de los píxeles) del mosaico en la memoria gráfica 40 ubicada localmente en la GPU 12, incluyendo realizar la cadena procesamiento gráfico para renderizar cada mosaico y, cuando se completa, introduce los datos gráficos renderizados de la memoria gráfica 40 en la memoria intermedia de cuadros 16 o en uno de los objetivos de renderización 24. En algunos ejemplos, debido a que cada mosaico renderizado incluye los valores de color de los píxeles de un bloque bidimensional de píxeles, un mosaico puede considerarse una superficie, o puede considerarse una parte de una superficie que es la imagen finalmente renderizada formada por una pluralidad de mosaicos.

[0038] La GPU 12 puede dividir cada mosaico en una pluralidad de bloques de píxeles. El tamaño de los bloques de píxeles puede ser similar al tamaño de los bloques de píxeles en el dispositivo de visualización 8 que corresponden a una ubicación de almacenamiento en la memoria intermedia de baja resolución. La GPU 12 puede transformar primitivas de cada mosaico en espacio de pantalla, y puede ordenar las primitivas entre sí de adelante hacia atrás, probando submosaicos del mosaico actual para: 1) determinar si cada primitiva está incluida dentro del submosaico dado; y 2) si está incluida en el submosaico dado, determinar si los píxeles de la primitiva están ocluidos por píxeles de cualquier otra primitiva en el submosaico particular.

[0039] En algunos ejemplos, durante la fase de agrupación, la GPU 12 también puede generar datos z de baja resolución (LRZ) para bloques de píxeles de cada uno de la pluralidad de mosaicos y puede almacenar dichos datos LRZ en una memoria intermedia de baja resolución en memoria, tal como la memoria de sistema 10. z de baja resolución se refiere al hecho de que la memoria intermedia de baja resolución almacena datos de profundidad asociados a un bloque de píxeles en lugar de para cada píxel de cada una de la pluralidad de mosaicos. La memoria intermedia de baja resolución puede ser una memoria intermedia bidimensional con una pluralidad de ubicaciones

de almacenamiento. Cada ubicación de almacenamiento en la memoria intermedia de baja resolución puede corresponder a un bloque de píxeles representado en el dispositivo de visualización 8. En algunos ejemplos, el número de ubicaciones de almacenamiento dentro de la memoria intermedia de baja resolución puede ser menor que el número de píxeles que se representarán en el dispositivo de visualización 8. Los datos LRZ pueden ser datos de profundidad para un bloque de píxeles (por ejemplo, un bloque 2x2 de píxeles) que contiene el valor de profundidad situado más atrás para el bloque de píxeles dado. Un mosaico puede estar asociado a uno o más datos LRZ. Por ejemplo, dado un mosaico que es un bloque 8x8 de píxeles, el mosaico puede incluir 16 datos LRZ que están asociados a un bloque 2x2 de píxeles dado del mosaico, y cada uno de los datos de 16 LRZ puede contener el valor de profundidad situado más atrás para el bloque 2x2 de píxeles asociado del mosaico.

[0040] La GPU 12 puede determinar los datos LRZ basándose en la determinación de los valores de profundidad de píxeles de primitivas que ocupan el bloque de píxeles asociado a los datos LRZ. Debido a que los datos LRZ son datos de profundidad para un bloque de píxeles en lugar de para un píxel individual, la GPU 12 puede ser conservadora para determinar los datos LRZ para cada bloque de píxeles. Por ejemplo, si los datos LRZ son un bloque 2x2 de píxeles (p00, p01, p10 y p11), la GPU 12 puede establecer que los datos LRZ correspondientes sean los datos de profundidad del píxel situado más atrás (es decir, el píxel más alejado de la cámara). Si los píxeles p00, p01, p10 y p11 tienen valores de profundidad correspondientes de 0,1, 0,1, 0,2 y 0,15, respectivamente, donde un valor más bajo representa una profundidad que está más alejada de la cámara que un valor más alto, la GPU 12 puede establecer los datos de LRZ para ese bloque de píxeles como 0,1.

[0041] Después de actualizar la memoria intermedia de baja resolución con información de profundidad de los píxeles que componen la superficie renderizada, la GPU 12 puede, mosaico por mosaico, renderizar una imagen en la memoria gráfica 40 en base a los valores de profundidad almacenados en la memoria intermedia de baja resolución. Para renderizar píxeles, para cada píxel en el dispositivo de visualización, la GPU 12 puede determinar qué píxeles renderizar a partir de qué primitivas en el mosaico en base a los valores de profundidad almacenados dentro de la memoria intermedia de baja resolución. Si la GPU 12 determina, en base a los valores de profundidad almacenados dentro de la memoria intermedia de baja resolución, que los píxeles de una primitiva están ocluidos en la escena final, la GPU 12 puede determinar no realizar más operaciones de sombreado de píxeles o de sombreado de fragmentos en esos píxeles ocluidos, mejorando así el rendimiento de la GPU 12. Después de que cada mosaico se renderice en la memoria gráfica 40, la GPU 12 puede transferir el mosaico renderizado desde la memoria gráfica 40 a la memoria 26. De esta manera, la memoria intermedia de cuadros 16 o uno de los objetivos de renderización 24 puede llenarse mosaico por mosaico con mosaicos renderizados de la GPU 12 y transferir cada uno de los mosaicos renderizados desde la memoria gráfica a la memoria intermedia de cuadros 16 o uno de los objetivos de renderización 24, renderizando de este modo una superficie en la memoria intermedia de cuadros 16 o uno de los objetivos de renderización 24.

[0042] Cuando la GPU 12 intenta renderizar primitivas adicionales en la superficie renderizada, la GPU 12 puede utilizar los datos LRZ generados para la superficie para optimizar la renderización de esas primitivas. La GPU 12 puede rasterizar esas primitivas en píxeles a través de las técnicas de esta divulgación y puede realizar pruebas de profundidad de baja resolución para descartar píxeles que la GPU 12 determina que están ocluidos. La GPU 12 puede, para cada píxel, comparar el valor de profundidad del píxel con el valor de profundidad de los datos LRZ asociados (es decir, los datos LRZ asociados a la ubicación de píxel del píxel que se está probando), y puede descartar el píxel si el valor de profundidad del píxel es más pequeño (por ejemplo, más alejado de la cámara) que el valor de profundidad de los datos LRZ asociados. Al descartar estos píxeles ocluidos, la GPU 12 puede omitir el rendimiento de cualquier operación de renderización de gráficos adicional para esos píxeles, tales como las operaciones de sombreado de píxeles y similares, mejorando de este modo el rendimiento del procesamiento gráfico de la GPU 12.

[0043] En algunas situaciones, la GPU 12 puede no rechazar píxeles, ya que están ocluidos necesariamente por otros píxeles cuando la GPU 12 realiza pruebas de baja resolución de esos píxeles usando datos LRZ, incluso si esos píxeles pueden rechazarse durante la prueba de profundidad a nivel de píxel de píxeles individuales. Por ejemplo, dado un dato LRZ que representa un bloque 2x2 de píxeles (p00, p01, p10 y p11), los datos LRZ pueden tener un valor de profundidad de 0,1, donde un valor más bajo representa una profundidad que está más alejada de la cámara que un valor más alto, aunque el píxel p01 puede tener un valor de profundidad real de 0,2. Posteriormente, la GPU 12 puede determinar si se debe renderizar una primitiva que tiene un nuevo píxel p01 'con un valor de profundidad de 0,15 en la misma ubicación de píxel que el píxel p01. Debido a que los datos LRZ tienen un valor de profundidad de 0,1, la GPU 12 puede, no obstante, basándose en los datos LRZ, determinar que la primitiva asociada al nuevo píxel p01 'será visible en la superficie finalmente renderizada ya que el píxel p01' tiene un valor de profundidad de 0,15 que es mayor que el valor de profundidad 0,1 de los datos LRZ, aunque el valor de profundidad real del píxel p01 es 0,2. Debido a la determinación de la GPU 12 de que el píxel p01' es visible en base a los datos LRZ, la GPU 12 puede realizar operaciones de renderización gráfica para el píxel (por ejemplo, operaciones de sombreado de fragmentos) antes de que la GPU 12 realice pruebas de profundidad a nivel de píxel en el píxel p01' para determinar que el píxel p01' no es realmente visible en la escena finalmente renderizada y descarta el píxel p01', evitando de este modo que los valores de color del píxel p01' se escriban en la memoria intermedia de cuadros 16 o en uno de los objetivos de renderización 24.

[0044] Debido a que la GPU 12 realiza pruebas de profundidad a nivel de píxel de cada píxel después de una prueba de profundidad de baja resolución usando datos LRZ, el uso de datos LRZ puede considerarse opcional. Si bien las pruebas de profundidad de baja resolución pueden descartar píxeles antes de que la GPU 12 realice operaciones de sombreado de píxeles en esos píxeles, la GPU 12 aún puede realizar en última instancia pruebas de profundidad por píxel de cada píxel no descartado después de que la GPU 12 realice operaciones de sombreado de píxeles en esos píxeles. Por lo tanto, las pruebas de profundidad de baja resolución usando datos LRZ pueden considerarse una optimización del procesamiento de la GPU 12 que evita que la GPU 12 emplee su procesamiento en realizar sombreado de píxeles en determinados píxeles que se descartan como resultado de las pruebas de profundidad de baja resolución. De este modo, la GPU 12 aún puede funcionar correctamente para representar datos gráficos, incluso si la GPU 12 no realiza pruebas de profundidad de baja resolución como parte de su procesamiento gráfico.

[0045] La GPU 12 también puede determinar un código *hash* basado en mosaico para cada mosaico renderizado en base a los datos de color del bloque de píxeles incluido en cada mosaico renderizado, de modo que un código *hash* basado en mosaico identifica de forma única mosaicos que tienen diferentes datos de color para su bloque de píxeles. Como se analiza anteriormente, cada mosaico renderizado es un bloque (por ejemplo, 8x8) de píxeles, donde cada píxel tiene un valor de color. La GPU 12 puede asociar mosaicos que contienen diferentes patrones de valores de píxel (por ejemplo, un mosaico completamente lleno de píxeles rojos y un mosaico completamente lleno de píxeles verdes) con diferentes códigos *hash* basados en mosaico, y puede asociar mosaicos que contienen el mismo patrón de valores de píxel (por ejemplo, dos mosaicos que están completamente llenos de píxeles rojos) con el mismo código *hash* basado en mosaicos.

[0046] Tales códigos *hash* basados en mosaico pueden ser útiles cuando la GPU 12 determina si se debe realizar una transferencia de bloques de bits de datos de color correspondientes a un mosaico desde un primer mosaico a un segundo mosaico. Si el primer mosaico y el segundo mosaico están asociados al mismo código *hash* basado en mosaicos, la GPU 12 puede determinar que no es necesario que se realicen transferencias de datos de color porque el primer y el segundo mosaico contienen el mismo conjunto de datos de color para sus respectivos bloques de píxeles, mejorando de este modo el rendimiento del dispositivo informático 2. En algunos ejemplos, la GPU 12 puede determinar un código *hash* basado en mosaico para bloques de píxeles que son más pequeños que el tamaño de un mosaico. Por ejemplo, si un mosaico comprende un bloque 8x8 de píxeles, la GPU 12 puede determinar un código *hash* basado en mosaico para cada bloque 4x4 de píxeles de una superficie. En este caso, cada mosaico puede estar asociado a cuatro códigos *hash* basados en mosaico para cada bloque 4x4 de píxeles que contiene.

[0047] A medida que cada mosaico renderizado se transfiere desde de la memoria gráfica 40 para su almacenamiento en la memoria intermedia de cuadros 16 o uno de los objetivos de renderización 24, la GPU 12 puede comprimir, por medio de cualquier algoritmo de compresión adecuado, cada mosaico para mover el mosaico de manera más eficiente a través del bus a la memoria intermedia de cuadros 16 o uno de los objetivos de renderización 24. El tamaño resultante de los mosaicos comprimidos puede diferir según la variabilidad del contenido de cada mosaico. Mientras que algunos mosaicos comprimidos pueden tener una fracción del tamaño de un mosaico sin comprimir, otros mosaicos comprimidos pueden ser ligeramente más pequeños o tener el mismo tamaño que un mosaico sin comprimir o no se pueden comprimir en absoluto. Por lo tanto, una pluralidad de mosaicos comprimidos en ancho de banda puede incluir uno o más mosaicos sin comprimir entre otros mosaicos comprimidos.

[0048] En algunos ejemplos, la GPU 12 puede determinar un código *hash* basado en mosaico para cada mosaico comprimido. Por lo tanto, en lugar de generar códigos *hash* basados en mosaico para los valores de color de superficie subyacentes del mosaico sin comprimir, la GPU 12 puede generar códigos *hash* basados en mosaico basados en los datos de cada mosaico después de la compresión, actuando de este modo como sumas de verificación para la pluralidad de mosaicos comprimidos. En este ejemplo, dos códigos *hash* basados en mosaico pueden ser iguales si los dos mosaicos comprimidos asociados, después de la compresión, son iguales.

[0049] Debido a que los mosaicos sin comprimir de una imagen renderizada dada tienen el mismo tamaño, la memoria intermedia de cuadros 16 o uno de los objetivos de renderización 24 están configurados para tener suficiente espacio para almacenar todos los mosaicos sin comprimir de una superficie en bloques de tamaño fijo, cada uno del mismo tamaño, como un mosaico sin comprimir. Además, debido a que la compresión de mosaicos que forman una superficie puede dar como resultado mosaicos de diferentes tamaños que varían en los valores de color de cada mosaico específico, la GPU 12 puede no ser capaz de asignar bloques personalizados de diferentes tamaños en la memoria 26 específicamente para almacenar los mosaicos comprimidos. Por lo tanto, la GPU 12 puede utilizar la misma pluralidad de bloques asignados para almacenar mosaicos sin comprimir de una imagen renderizada almacenando la pluralidad de mosaicos comprimidos en la pluralidad de bloques, de modo que cada mosaico comprimido se almacene en uno de los bloques.

[0050] Debido a que la memoria 26 almacena los mosaicos comprimidos en bloques que son cada uno del mismo tamaño que un mosaico sin comprimir, la memoria 26 en realidad no conserva ningún espacio al almacenar la pluralidad de mosaicos comprimidos en lugar de mosaicos sin comprimir. Aunque la pluralidad de mosaicos

comprimidos puede ocupar menos espacio en la memoria 26 que los mosaicos sin comprimir, se reserva la misma cantidad de espacio en la memoria 26 para la pluralidad de bloques, independientemente de si los mosaicos comprimidos o sin comprimir se almacenan en la pluralidad de bloques.

5 **[0051]** Por lo tanto, cuando la GPU 12 almacena los mosaicos comprimidos en la pluralidad de bloques, la pluralidad de bloques puede incluir espacio no usado que no se ocupa con el almacenamiento de la pluralidad de mosaicos comprimidos. Para cada mosaico comprimido que ocupa menos del espacio completo del bloque correspondiente en el que se almacena el mosaico comprimido, el bloque correspondiente puede tener espacio no usado. De este modo, de acuerdo con las técnicas de esta divulgación, la GPU 12 está configurada para utilizar el
10 espacio no usado para almacenar datos adicionales que están asociados a la superficie renderizada que está compuesta por la pluralidad de mosaicos comprimidos. Por ejemplo, en lugar de almacenar datos LRZ y códigos *hash* basados en mosaico para la pluralidad de mosaicos comprimidos en memorias intermedias dedicadas en la memoria 26, la GPU 12 puede almacenar dichos datos en el espacio no usado de la pluralidad de bloques.

15 **[0052]** Debido a que el espacio no usado en un bloque que almacena un mosaico comprimido no está garantizado, la GPU 12 puede almacenar datos asociados a un mosaico comprimido particular, como datos LRZ y códigos *hash* basados en mosaicos, solo si el bloque que almacena el mosaico comprimido particular tiene un espacio sin usar. Sin embargo, si un mosaico comprimido ocupa completamente un bloque, la GPU 12 puede no ser capaz de almacenar datos asociados al mosaico comprimido particular en el bloque. Por lo tanto, la GPU 12 puede
20 almacenar datos que son opcionales para cada parte de la superficie asociada a un mosaico comprimido correspondiente en los espacios no usados de los bloques.

[0053] La GPU 12 puede determinar el espacio no usado disponible en cada bloque de la pluralidad de bloques que resulta de la pluralidad de bloques que almacenan los mosaicos comprimidos. Por ejemplo, la GPU 12 puede
25 determinar el tamaño de un bloque de la pluralidad de bloques, y puede determinar el tamaño de cada uno de los mosaicos comprimidos. Si la GPU 12 determina que el tamaño de un mosaico comprimido particular es menor que el tamaño de un bloque de la pluralidad de bloques, la GPU 12 puede determinar que el bloque que almacena el mosaico comprimido particular puede tener espacio no usado.

30 **[0054]** En respuesta a la GPU 12 que determina que uno o más de la pluralidad de bloques incluyen espacio no usado, la GPU 12 puede almacenar superficies de optimización que la GPU 12 puede utilizar para mejorar su rendimiento en el espacio no usado del uno o más de la pluralidad de bloques. Por ejemplo, los datos LRZ son útiles al indicar primitivas que no son visibles en la superficie finalmente renderizada al permitir que la GPU 12 no realice la rasterización de esas primitivas. Sin embargo, sin los datos LRZ, la GPU 12 aún puede renderizar correctamente
35 una superficie dada al realizar la rasterización de primitivas independientemente de si esas primitivas son visibles en la superficie finalmente renderizada. De este modo, si bien los datos LRZ pueden mejorar el rendimiento de la GPU 12 al renderizar una superficie, no es información crítica para que la GPU 12 renderice correctamente una superficie.

40 **[0055]** Los códigos *hash* basados en mosaico son similares a los datos LRZ en que son útiles para mejorar el rendimiento de la GPU 12, pero no son críticos para que la GPU 12 realice correctamente operaciones gráficas. Sin códigos *hash* basados en mosaico, la GPU 12 aún puede realizar correctamente funciones tales como transferencias de datos de color en bloques de bits, y también puede realizar transferencias redundantes de datos de color entre partes de la superficie que tienen el mismo bloque de datos de color.
45

[0056] Las FIGS. 3A-3F son diagramas conceptuales que ilustran técnicas de ejemplo para almacenar en memoria datos gráficos comprimidos en ancho de banda. Como se muestra en la FIG. 3A, la GPU 12 puede almacenar datos gráficos comprimidos en ancho de banda 56A-56N ("datos gráficos comprimidos en ancho de banda 56") en bloques 54A-54N ("bloques 54") en memoria 26, tal como la memoria de sistema 10, la memoria intermedia de cuadros 16,
50 uno o más de los objetivos de renderización 24 y similares. Los datos gráficos comprimidos en ancho de banda 56, en algunos ejemplos, pueden ser un mosaico (por ejemplo, una parte de una superficie de imagen) que forma una escena o superficie renderizada comprimida por la GPU 12 para mover de manera más eficiente los datos gráficos a través de buses y entre componentes del dispositivo informático 2 (por ejemplo, entre la GPU 12 y la memoria 26).

55 **[0057]** Los bloques 54 pueden estar contiguos en la memoria 26 y cada uno puede tener el mismo tamaño fijo uniforme para almacenar cada uno de los datos gráficos comprimidos en ancho de banda 56. En algunos ejemplos, si cada uno de los datos gráficos comprimidos en ancho de banda 56 es un mosaico comprimido en ancho de banda, la GPU 12 puede asignar, en la memoria 26, el mismo número de bloques 54 que el número de mosaicos que componen una superficie renderizada, de modo que cada uno de los bloques 54 puede almacenar un dato correspondiente de los datos gráficos comprimidos en ancho de banda 56.
60

[0058] Debido a que cada uno de los bloques 54 es lo suficientemente grande como para almacenar datos gráficos sin comprimir de una superficie renderizada, el almacenamiento de datos gráficos comprimidos en ancho de banda 56 en los bloques 54 puede dar como resultado un espacio no usado restante en los bloques 54. En el ejemplo de la FIG. 3A, el espacio no usado 58A, 58B, 58C, 58D y 58E ("espacio no usado 58") puede permanecer
65

en los bloques 54A, 54B, 54C, 54E y 54N, respectivamente, cuando los bloques 54A, 54B, 54C, 54E y 54N almacenan datos gráficos respectivos comprimidos en ancho de banda 56A, 56B, 56C, 56E y 56N.

5 **[0059]** Como se analiza anteriormente, la GPU 12 puede determinar si cada bloque de bloques 54 tiene un espacio no usado 58 comparando el tamaño de cada dato de gráficos comprimidos en ancho de banda 56 con el tamaño de un bloque de bloques 54. La GPU 12 puede crear y almacenar superficies bandera 52A-52N ("superficies bandera 52") en la memoria 26, donde cada una de las superficies bandera 52 está asociada a uno de los bloques 54, y puede indicar la cantidad de espacio no usado en un bloque de bloques 54 correspondiente.

10 **[0060]** En el ejemplo de la FIG. 3A, las superficies bandera 52 pueden almacenar la fracción, sobre cuatro, de la cantidad de espacio no usado en un bloque correspondiente de bloques 54. La superficie bandera 52A puede indicar que el espacio no usado ocupa $\frac{3}{4}$ del bloque 54A. La superficie bandera 52B puede indicar que el espacio no usado ocupa $\frac{1}{2}$ del bloque 54B. La superficie bandera 52C puede indicar que el espacio no usado ocupa $\frac{1}{4}$ del bloque 54C. La superficie bandera 52D puede indicar que el bloque 54D no tiene espacio no usado. La superficie bandera 52E puede indicar que el espacio no usado ocupa $\frac{1}{4}$ del bloque 54E. La superficie bandera 52F puede indicar que el bloque 54F no tiene espacio no usado. La superficie bandera 52N puede indicar que el espacio no usado ocupa $\frac{1}{4}$ del bloque 54N. Debido a que las superficies bandera 52 también se almacenan en la memoria 26, almacenar datos de gráficos comprimidos en ancho de banda 56 en memoria puede ocupar más espacio en la memoria 26 que almacenar datos gráficos sin comprimir comparables 56.

20 **[0061]** Como se ha analizado anteriormente, la GPU 12 almacena datos asociados a datos gráficos comprimidos en ancho de banda 56 en el espacio no usado 58. Como se muestra en la FIG. 3B, la GPU 12 puede determinar, en base a la superficie bandera 52, los bloques de bloques 54 que tienen espacio no usado 58, y puede almacenar datos LRZ 60A-60E en el espacio no usado 58A-58E de bloques 54. Cada uno de los datos LRZ 60A-60E puede ser de un tamaño fijo. Debido a que solo los bloques 54A, 54B, 54C, 54E y 54N tienen un espacio no usado respectivo 58A-58E, la GPU 12 puede, en el ejemplo de la FIG. 3B, solo almacenar datos LRZ 60A-60E que incluyen información de profundidad para datos gráficos respectivos comprimidos en ancho de banda 56A, 56B, 56C, 56E y 56N en el espacio no usado 58 de los bloques 54. Por lo tanto, la información de profundidad para los datos gráficos comprimidos en ancho de banda 56D y 56F no se almacenan en el espacio no usado 58 de los bloques 54.

30 **[0062]** Los datos LRZ 60A pueden estar asociados a los datos gráficos comprimidos en ancho de banda 56A ya que los datos LRZ 60A pueden incluir datos LRZ para los píxeles que forman la parte de la superficie que corresponde a los datos gráficos comprimidos en ancho de banda 56A. Por ejemplo, si los datos gráficos comprimidos en ancho de banda 56A incluyen datos gráficos con respecto a un bloque 8x8 particular de píxeles, los datos LRZ 60A, en un ejemplo, pueden incluir una pluralidad correspondiente de datos LRZ para cada bloque 2x2 de píxeles del bloque 8x8 de píxeles. De manera similar, los datos LRZ 60B pueden incluir datos LRZ para los píxeles que forman la parte de la superficie que corresponde a datos gráficos comprimidos en ancho de banda 56B, los datos LRZ 60C pueden incluir datos LRZ para los píxeles que forman la parte de la superficie que corresponde a los datos gráficos comprimidos en ancho de banda 56C, los datos LRZ 60D pueden incluir datos LRZ para los píxeles que forman la parte de la superficie que corresponde a los datos gráficos comprimidos en ancho de banda 56E, y los datos LRZ 60E pueden incluir datos LRZ para los píxeles que forman la parte de la superficie que corresponde a los datos gráficos comprimidos en ancho de banda 56N.

45 **[0063]** Para los datos gráficos comprimidos en ancho de banda 56D y 56F que no tienen datos LRZ asociados almacenados en los bloques asociados 54D y 54F, la GPU 12 puede asociar un valor de profundidad predeterminado a cada uno de los datos gráficos comprimidos en ancho de banda 56D y 56F que ocupan completamente sus bloques 54D y 54F respectivos. El valor de profundidad predeterminado puede ser el valor de profundidad situado más atrás que indica que píxeles adicionales que se renderizarán en las partes de la superficie asociadas a los datos gráficos comprimidos en ancho de banda 56D y 56F están delante de los píxeles de las partes de la superficie asociadas a los datos gráficos comprimidos en ancho de banda 56D y 56F y, por lo tanto, serán visibles, independientemente de si esos píxeles adicionales son realmente visibles en la escena finalmente renderizada.

55 **[0064]** Para incorporar la información de profundidad para cada uno de los datos gráficos comprimidos en ancho de banda 56, la GPU 12 puede almacenar información de profundidad para múltiples datos gráficos comprimidos en ancho de banda 56 en el espacio no usado 58 de un solo bloque de bloques 54. Como se muestra en la FIG. 3C, la GPU 12 puede almacenar datos LRZ 60F en el espacio no usado 58A del bloque 54A que incluye datos LRZ para múltiples datos gráficos consecutivos comprimidos en ancho de banda 56. Los datos LRZ almacenados en el espacio no usado 58 de un solo bloque del bloque 54 pueden incluir información de profundidad para los datos gráficos asociados comprimidos en ancho de banda de datos gráficos comprimidos en ancho de banda 56, así como información de profundidad para el siguiente número específico consecutivo de datos gráficos comprimidos en ancho de banda 56. Por ejemplo, si los datos LRZ almacenados en el espacio no usado 58 de un solo bloque del bloque 54 pueden incluir datos LRZ para seis de los datos gráficos comprimidos en ancho de banda 56, los datos LRZ 60F pueden incluir datos de profundidad para datos gráficos comprimidos en ancho de banda 56A-56F. De manera similar, los datos LRZ 60G pueden incluir datos de profundidad para datos gráficos comprimidos en ancho de banda 56N, así como los siguientes cinco datos gráficos posteriores comprimidos en ancho de banda 56. De esta

manera, los bloques 54 pueden almacenar datos de profundidad para cada dato gráfico comprimido en ancho de banda 56.

[0065] Como se muestra en la FIG. 3D, la GPU 12 también puede almacenar códigos *hash* basados en mosaico 62A-62E en el espacio no usado 58A-58E de los bloques 54. Cada uno de los códigos *hash* basados en mosaico 62A-62E puede ser del mismo tamaño. Debido a que solo los bloques 54A, 54B, 54C, 54E y 54N tienen un espacio no usado respectivo 58A-58E, la GPU 12 solo puede, en el ejemplo de la FIG. 3B, almacenar códigos *hash* basados en mosaico 62A-62E que identifiquen los valores de color para los respectivos datos gráficos comprimidos en ancho de banda 56A, 56B, 56C, 56E y 56N en el espacio no usado 58 de los bloques 54. Por lo tanto, los códigos *hash* basados en mosaico para los datos gráficos comprimidos en ancho de banda 56D y 56F no se almacenan en el espacio no usado 58 de los bloques 54.

[0066] Para incorporar códigos *hash* basados en mosaico para cada uno de los datos gráficos comprimidos en ancho de banda 56, la GPU 12 puede almacenar códigos *hash* basados en mosaico para múltiples datos gráficos comprimidos en ancho de banda 56 en el espacio no usado 58 de un solo bloque de bloques 54. Como se muestra en la FIG. 3E, la GPU 12 puede almacenar código *hash* basado en mosaico 62F en el espacio no usado 58C del bloque 54C que incluye códigos *hash* basados en mosaico para múltiples datos gráficos consecutivos comprimidos en ancho de banda 56. Los códigos *hash* basados en mosaicos almacenados en el espacio no usado 58 de un solo bloque del bloque 54 pueden incluir códigos *hash* basados en mosaicos para los datos gráficos asociados comprimidos en ancho de banda de los datos gráficos comprimidos en ancho de banda 56, así como códigos *hash* basados en mosaicos para el siguiente número especificado consecutivo de datos gráficos comprimidos en ancho de banda 56 o un número consecutivo previo de datos gráficos comprimidos en ancho de banda 56. Por ejemplo, si los códigos *hash* basados en mosaicos almacenados en el espacio no usado 58 de un solo bloque del bloque 54 pueden incluir datos LRZ para tres de los datos gráficos comprimidos en ancho de banda 56, el código *hash* basado en mosaico 62F puede incluir códigos *hash* basados en mosaico para cada uno de los datos gráficos comprimidos en ancho de banda 56A-56C. De manera similar, el código *hash* basado en mosaico 62G puede incluir códigos *hash* basados en mosaico para datos gráficos comprimidos en ancho de banda 56N, así como los dos datos gráficos anteriores comprimidos en ancho de banda 56. De esta manera, los bloques 54 pueden almacenar códigos *hash* basados en mosaico para cada dato gráfico comprimido en ancho de banda 56.

[0067] En algunos ejemplos, la GPU 12 puede almacenar múltiples tipos de datos asociados a datos gráficos comprimidos en ancho de banda 56 en el espacio no usado 58 de los bloques 54 al mismo tiempo. Por ejemplo, el espacio no usado 58 de los bloques 54 puede almacenar tanto datos de profundidad como códigos *hash* basados en mosaico para cada uno de los datos gráficos comprimidos en ancho de banda 56. Como se muestra en la FIG. 3F, la GPU 12 puede almacenar datos LRZ 60H en el espacio no usado 58A del bloque 54A y datos LRZ 60I en el espacio no usado 58D del bloque 54E. La GPU 12 también puede almacenar el código *hash* basado en mosaico 62H en el espacio no usado 58C del bloque 54A y el código *hash* basado en mosaico 62I en el espacio no usado 58E del bloque 54N. De este modo, el espacio no usado 58 de los bloques 54 puede almacenar tanto datos LRZ como códigos *hash* basados en mosaico para datos gráficos comprimidos en ancho de banda 56 al mismo tiempo.

[0068] Si bien las FIGS. 3A-3F ilustran que la GPU 12 puede almacenar datos LRZ y códigos *hash* basados en mosaico en el espacio no usado 58 de los bloques 54, esta divulgación no se limita necesariamente al almacenamiento de solo datos LRZ y códigos *hash* basados en mosaico en el espacio no usado 58 de bloques 54. Por el contrario, la GPU 12 puede almacenar cualquier otro dato relacionado con datos gráficos comprimidos en ancho de banda 56 en el espacio no usado 58 de bloques 54.

[0069] La FIG. 4 es un diagrama de flujo que ilustra un proceso de ejemplo para almacenar en memoria datos gráficos comprimidos en ancho de banda. Como se muestra en la FIG. 4, el proceso puede incluir el almacenamiento, mediante la GPU 12, de una pluralidad de datos gráficos comprimidos en ancho de banda 56 en una pluralidad respectiva de bloques 54 en la memoria 26, donde cada bloque de la pluralidad de bloques 54 tiene un tamaño fijo uniforme en la memoria 26, y donde uno o más datos de la pluralidad de datos gráficos comprimidos en ancho de banda 56 tiene un tamaño que es más pequeño que el tamaño fijo (102). El proceso puede incluir además el almacenamiento, mediante la GPU 12, de datos asociados a la pluralidad de datos gráficos comprimidos en ancho de banda 56 en el espacio no usado 58 de uno o más bloques de la pluralidad de bloques 54 que contiene el uno o más datos respectivos de la pluralidad de datos gráficos comprimidos en ancho de banda 56 (104).

[0070] En algunos ejemplos, los datos asociados a la pluralidad de datos gráficos comprimidos en ancho de banda 56 comprenden datos de profundidad para el uno o más datos de la pluralidad de datos gráficos comprimidos en ancho de banda 56 almacenados en el uno o más bloques de la pluralidad de bloques 54. En algunos ejemplos, uno o más segundos datos de la pluralidad de datos gráficos comprimidos en ancho de banda 56 puede ocupar completamente uno o más segundos bloques de la pluralidad de bloques 54, y el proceso puede incluir además asociar, mediante la GPU 12, un valor de profundidad predeterminado para cada uno del uno o más segundos datos de la pluralidad de datos gráficos comprimidos en ancho de banda 56. En algunos ejemplos, los datos asociados a la pluralidad de datos gráficos comprimidos en ancho de banda 56 comprenden datos de profundidad para cada uno de los datos de la pluralidad de datos gráficos comprimidos en ancho de banda 56 almacenados en el uno o más bloques de la pluralidad de bloques 54.

- 5 **[0071]** En algunos ejemplos, los datos asociados a la pluralidad de datos gráficos comprimidos en ancho de banda 56 comprenden uno o más códigos *hash* que identifican cada dato del uno o más datos de la pluralidad de datos gráficos comprimidos en ancho de banda 56 almacenados en el uno o más bloques de la pluralidad de bloques 54.
- 10 **[0072]** En algunos ejemplos, los datos asociados a la pluralidad de datos gráficos comprimidos en ancho de banda 56 comprenden códigos *hash* que identifican cada uno de los datos de la pluralidad de datos gráficos comprimidos en ancho de banda 56 almacenados en el uno o más bloques de la pluralidad de bloques 54 y datos de profundidad para cada uno de los datos de la pluralidad de datos gráficos comprimidos en ancho de banda 56 almacenados en el uno o más bloques de la pluralidad de bloques 54.
- 15 **[0073]** En algunos ejemplos, los datos asociados a la pluralidad de datos gráficos comprimidos en ancho de banda 56 comprenden superficies de optimización asociadas a la pluralidad de datos gráficos comprimidos en ancho de banda 56. En algunos ejemplos, la pluralidad de datos gráficos comprimidos en ancho de banda 56 puede comprender partes comprimidas en ancho de banda de una superficie de imagen.
- 20 **[0074]** En algunos ejemplos, el almacenamiento, mediante la GPU 12, de los datos asociados a la pluralidad de datos gráficos comprimidos en ancho de banda 56 en el espacio no usado del uno o más bloques de la pluralidad de bloques 54 que contiene el uno o más datos respectivos de la pluralidad de datos gráficos comprimidos en ancho de banda 56 puede incluir además determinar, mediante la GPU 12, que el uno o más bloques de la pluralidad de bloques 54 incluyan el espacio no usado y, en respuesta a la determinación de que uno o más bloques de la pluralidad de bloques 54 incluyan el espacio no usado, almacenar, mediante la GPU 12, los datos asociados a la pluralidad de datos gráficos comprimidos en ancho de banda 56 en el espacio no usado del uno o más bloques de la pluralidad de bloques 54 que contiene el uno o más datos respectivos de la pluralidad de datos gráficos comprimidos en ancho de banda 56.
- 25 **[0075]** En uno o más ejemplos, las funciones descritas se pueden implementar en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones se pueden almacenar en, o transmitir por, un medio legible por ordenador como una o más instrucciones o código. Los medios legibles por ordenador pueden incluir medios de almacenamiento de datos informáticos o medios de comunicación, incluido cualquier medio que facilite la transferencia de un programa informático desde un lugar a otro. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se pueda acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en la presente divulgación. A modo de ejemplo y no de limitación, dichos medios legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otros dispositivos de almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, o cualquier otro medio que se pueda usar para transportar o almacenar el código de programa deseado en forma de instrucciones o estructuras de datos y al que se pueda acceder mediante un ordenador. Además, cualquier conexión recibe apropiadamente la denominación de medio legible por ordenador. Por ejemplo, si el software se transmite desde un sitio web, un servidor u otra fuente remota usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea digital de abonado (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 incluyen en la definición de medio. Los discos, como se usan en el presente documento, incluyen el disco compacto (CD), el disco láser, el disco óptico, el disco versátil digital (DVD), el disco flexible y el disco Blu-ray, donde algunos discos reproducen normalmente los datos magnéticamente, mientras que otros discos reproducen los datos ópticamente con láseres. Combinaciones de lo anterior también deben incluirse dentro del alcance de los medios legibles por ordenador.
- 30 **[0076]** El código se puede ejecutar 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), matrices lógicas programables *in situ* (FPGA) u otros circuitos lógicos, integrados o discretos equivalentes. En consecuencia, el término "procesador" o "unidad de procesamiento", como se usa en el presente documento, se puede referir a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar en módulos de hardware y/o de software dedicados configurados para la codificación y la decodificación, o incorporarse en un códec combinado. Además, las técnicas se podrían implementar por completo en uno o más circuitos o elementos lógicos.
- 35 **[0077]** Las técnicas de esta divulgación se pueden implementar en una gran diversidad de dispositivos o aparatos, incluido un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (es decir, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para destacar los aspectos funcionales de los dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En su lugar, como se ha descrito anteriormente, diversas
- 40 **[0078]** Las técnicas de esta divulgación se pueden implementar en una gran diversidad de dispositivos o aparatos, incluido un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (es decir, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para destacar los aspectos funcionales de los dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En su lugar, como se ha descrito anteriormente, diversas
- 45 **[0079]** Las técnicas de esta divulgación se pueden implementar en una gran diversidad de dispositivos o aparatos, incluido un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (es decir, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para destacar los aspectos funcionales de los dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En su lugar, como se ha descrito anteriormente, diversas
- 50 **[0080]** Las técnicas de esta divulgación se pueden implementar en una gran diversidad de dispositivos o aparatos, incluido un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (es decir, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para destacar los aspectos funcionales de los dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En su lugar, como se ha descrito anteriormente, diversas
- 55 **[0081]** Las técnicas de esta divulgación se pueden implementar en una gran diversidad de dispositivos o aparatos, incluido un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (es decir, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para destacar los aspectos funcionales de los dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En su lugar, como se ha descrito anteriormente, diversas
- 60 **[0082]** Las técnicas de esta divulgación se pueden implementar en una gran diversidad de dispositivos o aparatos, incluido un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (es decir, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para destacar los aspectos funcionales de los dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En su lugar, como se ha descrito anteriormente, diversas
- 65 **[0083]** Las técnicas de esta divulgación se pueden implementar en una gran diversidad de dispositivos o aparatos, incluido un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (es decir, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para destacar los aspectos funcionales de los dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En su lugar, como se ha descrito anteriormente, diversas

ES 2 770 874 T3

unidades se pueden combinar en una unidad de hardware de códec o proporcionar mediante un grupo de unidades de hardware interoperativas, incluidos uno o más procesadores como los descritos anteriormente, junto con software y/o firmware adecuados.

REIVINDICACIONES

1. Un procedimiento, que comprende:

5 almacenar, mediante al menos un procesador, una pluralidad de datos gráficos comprimidos en ancho de banda en una pluralidad respectiva de bloques en memoria, incluyendo almacenar uno o más primeros datos de la pluralidad de datos gráficos comprimidos en ancho de banda en uno o más primeros bloques de la pluralidad de bloques, y almacenar uno o más segundos datos de la pluralidad de datos gráficos comprimidos en ancho de banda en uno o más segundos bloques de la pluralidad de bloques, donde cada bloque de la pluralidad de bloques tiene un tamaño fijo uniforme en la memoria que puede contener completamente un dato más grande de la pluralidad de datos gráficos comprimidos en ancho de banda, donde cada uno de los uno o más primeros datos de la pluralidad de datos gráficos comprimidos en ancho de banda tiene un tamaño que es menor que el tamaño fijo, y donde cada uno del uno o más segundos datos de la pluralidad de datos gráficos comprimidos en ancho de banda se almacena completamente en cada uno de los uno o más segundos bloques respectivos de la pluralidad de bloques; y

15 almacenar, mediante el al menos un procesador, datos asociados a la pluralidad de datos gráficos comprimidos en ancho de banda en el espacio no usado del uno o más primeros bloques de la pluralidad de bloques que contiene el uno o más primeros datos respectivos de la pluralidad de datos gráficos comprimidos en ancho de banda en función de, al menos en parte, una determinación de que el uno o más primeros bloques de la pluralidad de bloques incluye el espacio no usado de manera que el procesador usa los datos para optimizar el rendimiento de las operaciones gráficas en los datos gráficos.

20 2. El procedimiento según la reivindicación 1, en el que los datos asociados a la pluralidad de datos gráficos comprimidos en ancho de banda comprenden datos de profundidad para el uno o más primeros datos de la pluralidad de datos gráficos comprimidos en ancho de banda almacenados en el uno o más primeros bloques de la pluralidad de bloques.

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

30 asociar, mediante el al menos un procesador, un valor de profundidad predeterminado para cada uno de los uno o más segundos datos de la pluralidad de datos gráficos comprimidos en ancho de banda que ocupan completamente el uno o más segundos bloques de la pluralidad de bloques.

35 4. El procedimiento según la reivindicación 1, en el que los datos asociados a la pluralidad de datos gráficos comprimidos en ancho de banda comprenden uno o más códigos *hash* que identifican cada uno del uno o más primeros datos de la pluralidad de datos gráficos comprimidos en ancho de banda almacenados en el uno o más primeros bloques de la pluralidad de bloques, o en el que

40 los datos asociados a la pluralidad de datos gráficos comprimidos en ancho de banda comprenden códigos *hash* que identifican cada uno de los datos de la pluralidad de datos gráficos comprimidos en ancho de banda almacenados en la pluralidad de bloques.

45 5. El procedimiento según la reivindicación 1, en el que los datos asociados a la pluralidad de datos gráficos comprimidos en ancho de banda comprenden códigos *hash* que identifican cada uno de los datos de la pluralidad de datos gráficos comprimidos en ancho de banda almacenados en la pluralidad de bloques y datos de profundidad para cada uno de los datos de la pluralidad de datos gráficos comprimidos en ancho de banda almacenados en la pluralidad de bloques.

50 6. El procedimiento según la reivindicación 1, en el que los datos asociados a la pluralidad de datos gráficos comprimidos en ancho de banda comprenden superficies de optimización asociadas a la pluralidad de datos gráficos comprimidos en ancho de banda que se usa para optimizar la renderización de los datos gráficos.

55 7. El procedimiento según la reivindicación 1, en el que la pluralidad de datos gráficos comprimidos en ancho de banda comprende partes comprimidas en ancho de banda de una superficie de imagen.

8. El procedimiento según la reivindicación 1, en el que el al menos un procesador incluye una unidad de procesamiento gráfico.

60 9. Un aparato, que comprende:

65 medios para almacenar una pluralidad de datos gráficos comprimidos en ancho de banda en una pluralidad respectiva de bloques en memoria, incluyendo medios para almacenar uno o más primeros datos de la pluralidad de datos gráficos comprimidos en ancho de banda en uno o más primeros bloques de la pluralidad de bloques, y medios para almacenar uno o más segundos datos de la pluralidad de datos gráficos comprimidos en ancho de banda en uno o más segundos bloques de la pluralidad de bloques, donde cada

5 uno de los bloques de la pluralidad de bloques tiene un tamaño fijo uniforme en la memoria, donde cada uno de los uno o más primeros datos de la pluralidad de datos gráficos comprimidos en ancho de banda tiene un tamaño que es menor que el tamaño fijo, y donde cada uno del uno o más segundos datos de la pluralidad de datos gráficos comprimidos en ancho de banda se almacena completamente en cada uno de los uno o más segundos bloques respectivos de la pluralidad de bloques; y

10 medios para almacenar datos asociados a la pluralidad de datos gráficos comprimidos en ancho de banda en el espacio no usado de uno o más bloques de la pluralidad de bloques que contiene el uno o más datos respectivos de la pluralidad de datos gráficos comprimidos en ancho de banda de modo que los datos almacenados se usen para optimizar el rendimiento de las operaciones gráficas en los datos gráficos.

15 **10.** El aparato según la reivindicación 9, en el que los datos asociados a la pluralidad de datos gráficos comprimidos en ancho de banda comprenden datos de profundidad para el uno o más primeros datos de la pluralidad de datos gráficos comprimidos en ancho de banda almacenados en el uno o más primeros bloques de la pluralidad de bloques.

11. El aparato según la reivindicación 10, que comprende además:

20 medios para asociar un valor de profundidad predeterminado para cada uno de los uno o más segundos datos de la pluralidad de datos gráficos comprimidos en ancho de banda que ocupan completamente el uno o más segundos bloques de la pluralidad de bloques.

25 **12.** El aparato según la reivindicación 9, en el que los datos asociados a la pluralidad de datos gráficos comprimidos en ancho de banda comprenden datos de profundidad para cada uno de los datos de la pluralidad de datos gráficos comprimidos en ancho de banda almacenados en la pluralidad de bloques.

30 **13.** El aparato según la reivindicación 9, en el que los datos asociados a la pluralidad de datos gráficos comprimidos en ancho de banda comprenden uno o más códigos *hash* que identifican cada uno del uno o más primeros datos de la pluralidad de datos gráficos comprimidos en ancho de banda almacenados en el uno o más primeros bloques de la pluralidad de bloques.

35 **14.** El aparato según la reivindicación 9, en el que los datos asociados a la pluralidad de datos gráficos comprimidos en ancho de banda comprenden códigos *hash* que identifican cada uno de los datos de la pluralidad de datos gráficos comprimidos en ancho de banda almacenados en la pluralidad de bloques.

40 **15.** El aparato según la reivindicación 9, en el que los datos asociados a la pluralidad de datos gráficos comprimidos en ancho de banda comprenden códigos *hash* que identifican cada uno de los datos de la pluralidad de datos gráficos comprimidos en ancho de banda almacenados en la pluralidad de bloques y datos de profundidad para cada uno de los datos de la pluralidad de datos gráficos comprimidos en ancho de banda almacenados en la pluralidad de bloques.

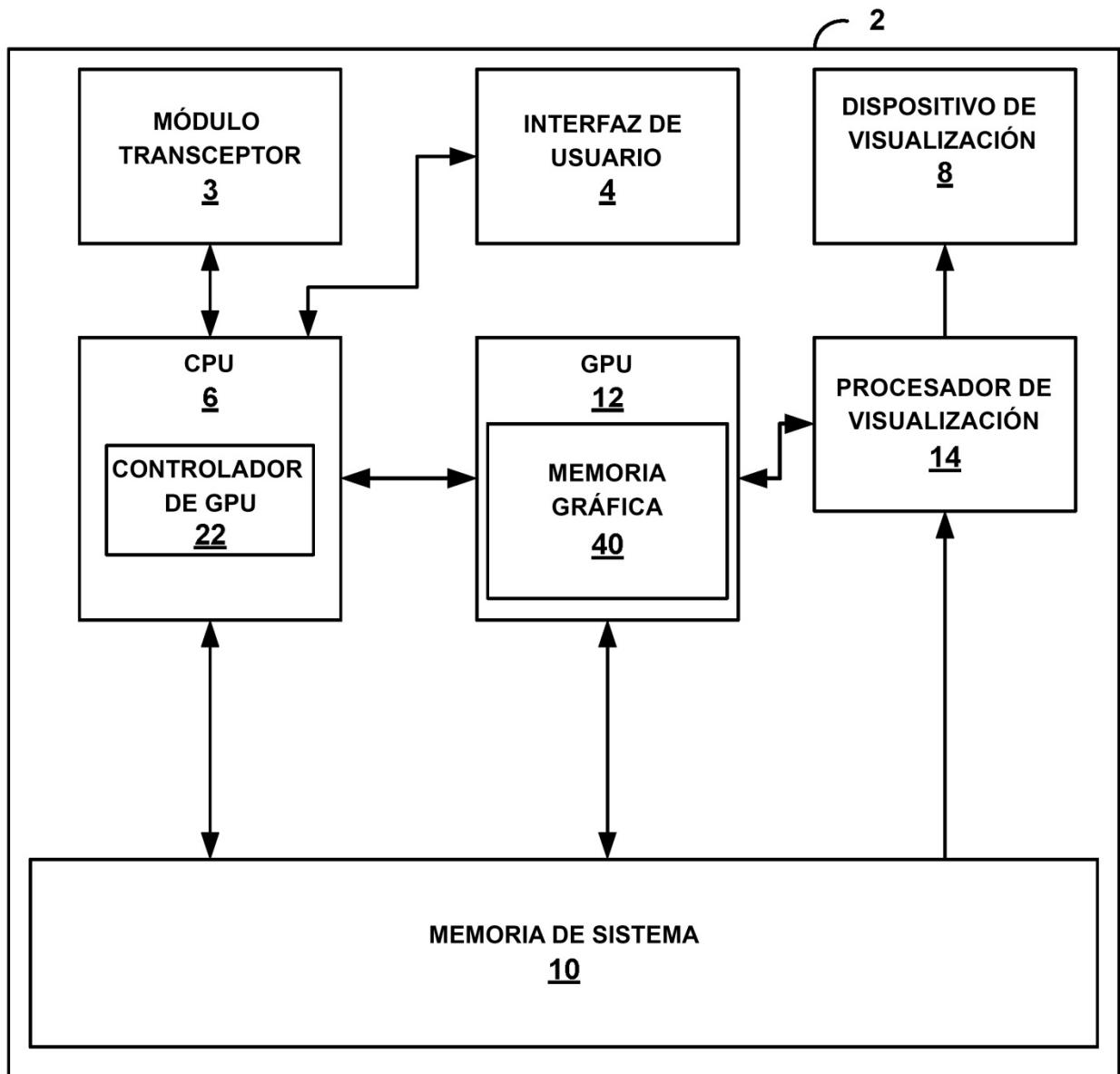


FIG. 1

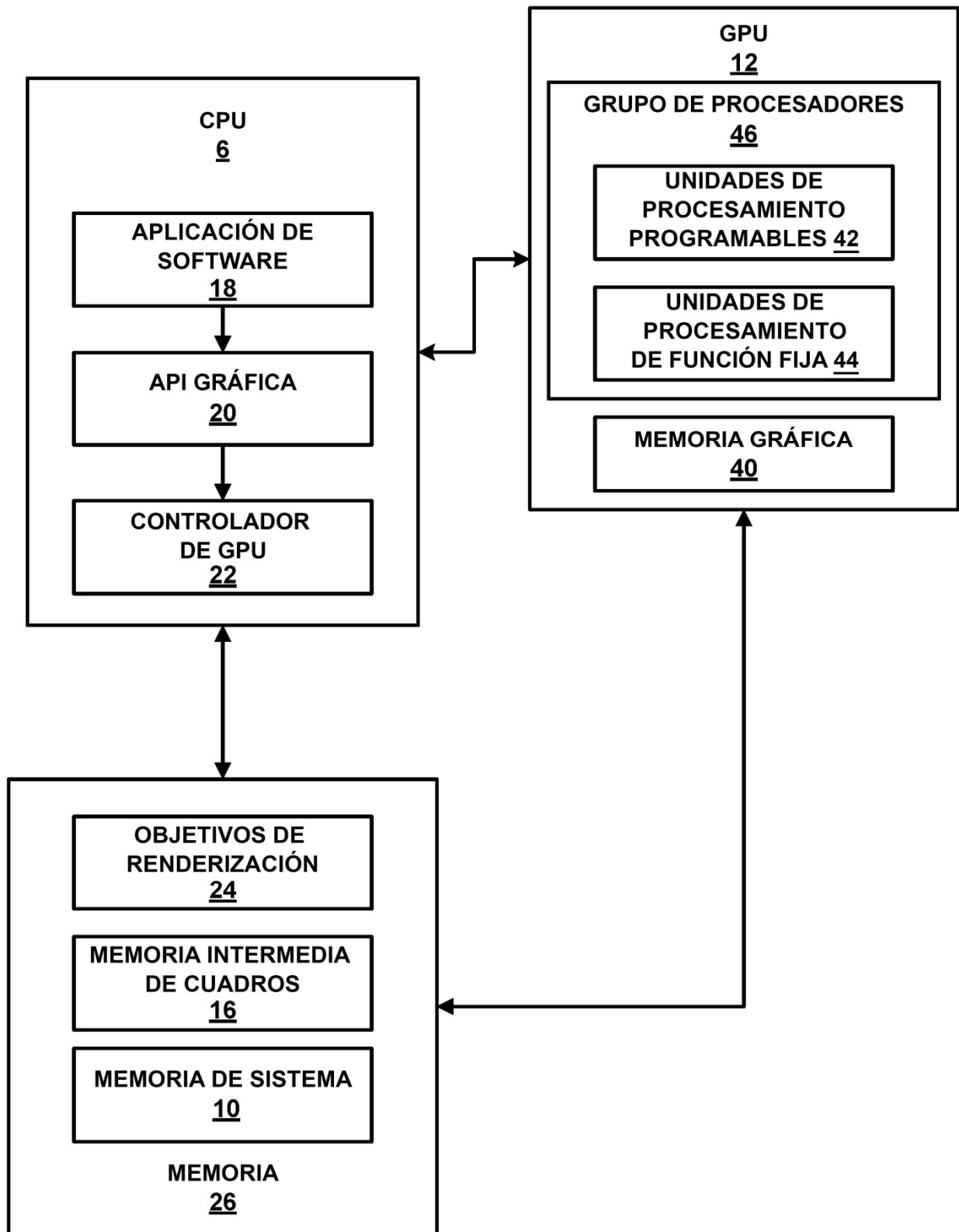


FIG. 2

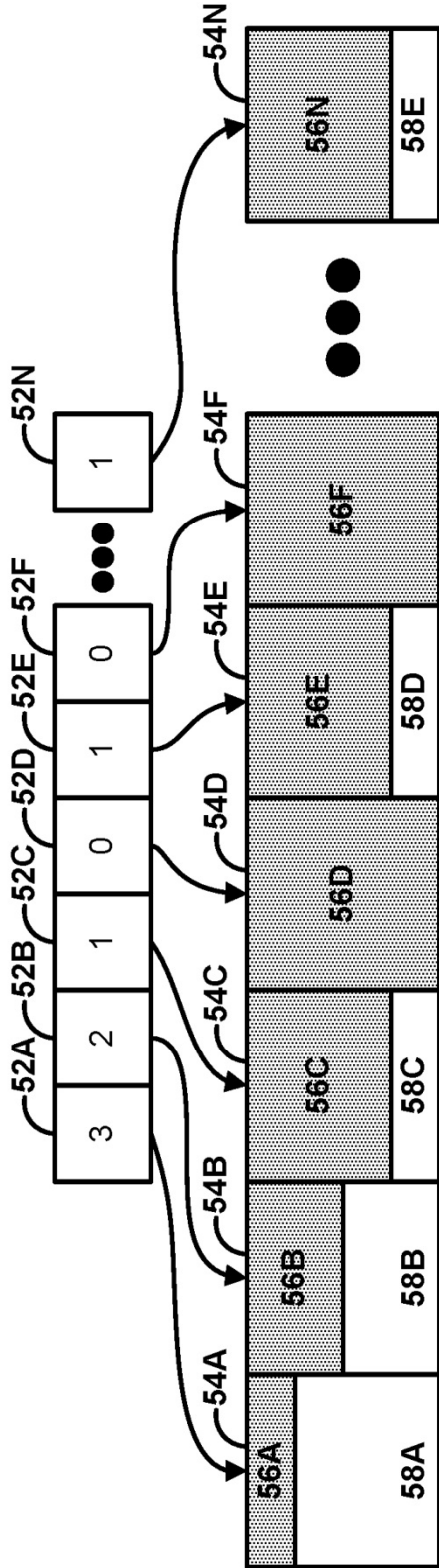


FIG. 3A

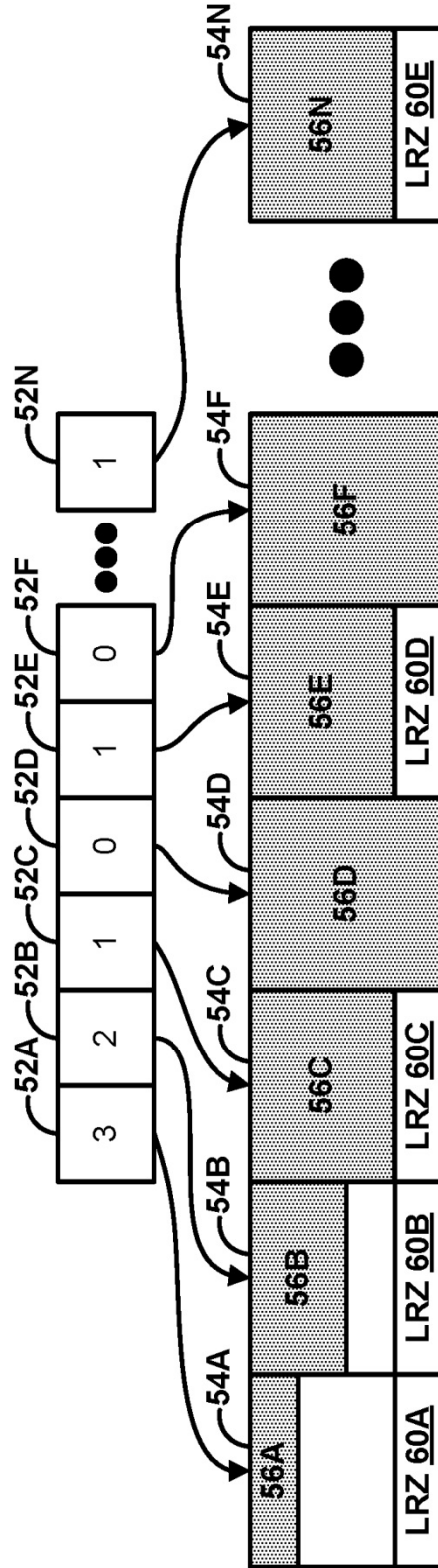


FIG. 3B

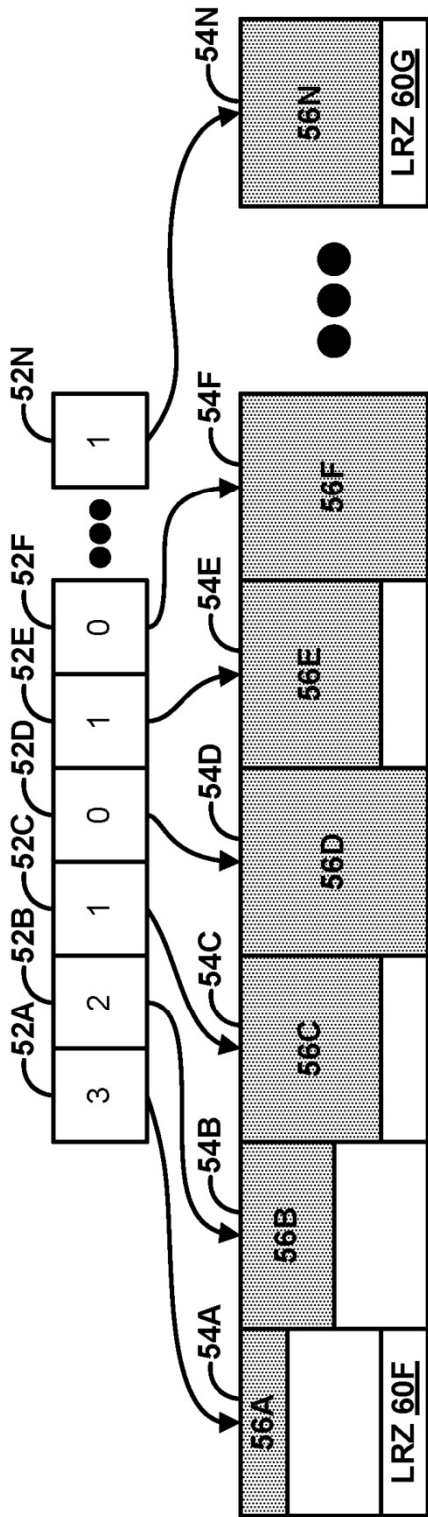


FIG. 3C

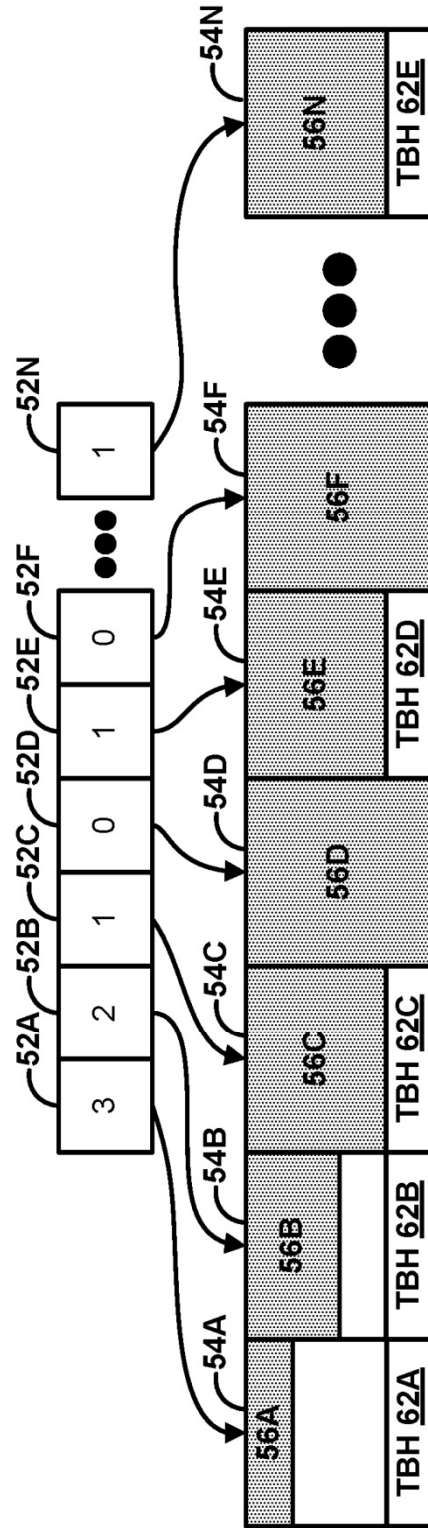


FIG. 3D

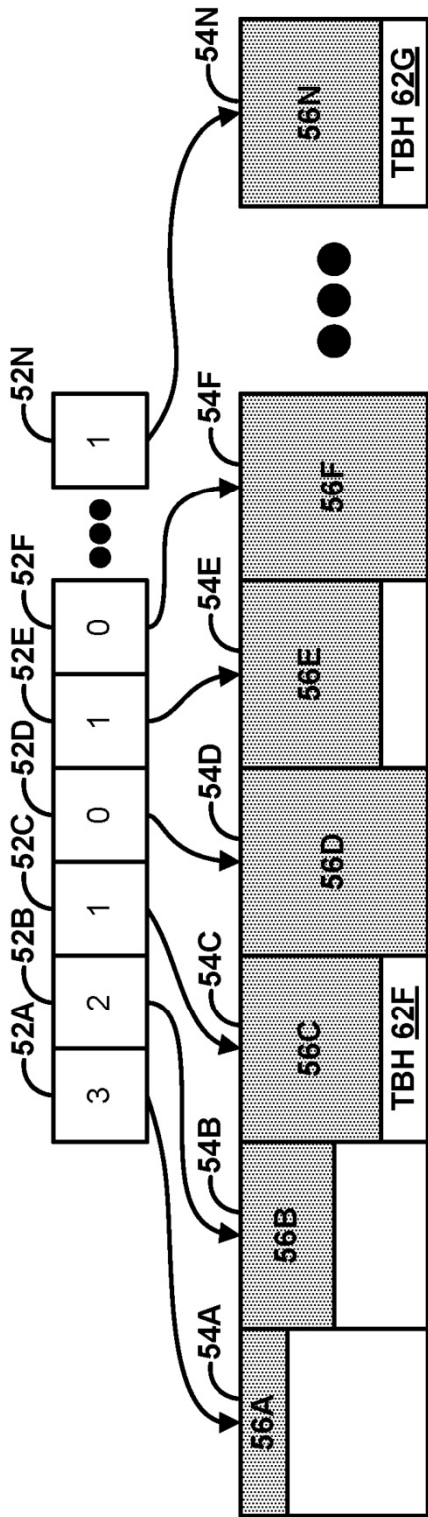


FIG. 3E

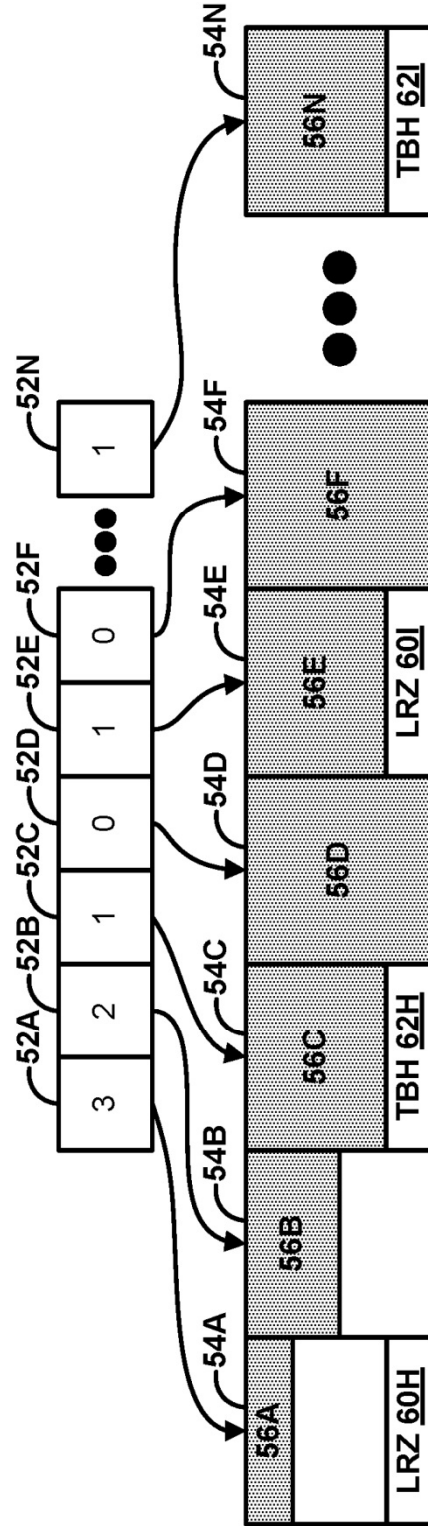


FIG. 3F

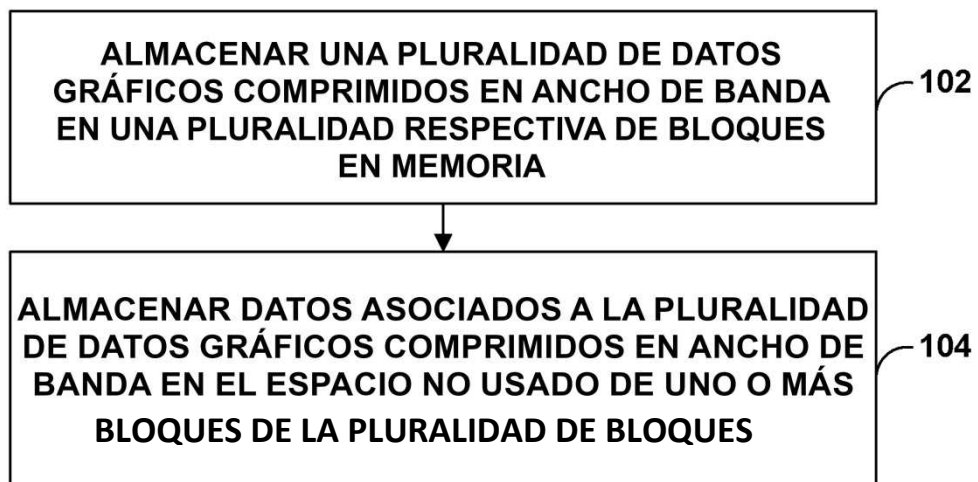


FIG. 4