

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 750 977**

51 Int. Cl.:

**G06F 7/552** (2006.01)

**G09C 1/00** (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **22.06.2015 PCT/US2015/036960**

87 Fecha y número de publicación internacional: **21.01.2016 WO16010681**

96 Fecha de presentación y número de la solicitud europea: **22.06.2015 E 15736718 (6)**

97 Fecha y número de publicación de la concesión europea: **24.07.2019 EP 3170069**

54 Título: **Instrucciones de escalado de vectores para su uso en una unidad aritmética lógica**

30 Prioridad:

**15.07.2014 US 201414331991**

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

**30.03.2020**

73 Titular/es:

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

72 Inventor/es:

**CHEN, LIN;  
GRUBER, ANDREW EVAN;  
JIAO, GUOFANG;  
HO, CHIEN y  
ARGADE, PRAMOD VASANT**

74 Agente/Representante:

**FORTEA LAGUNA, Juan José**

ES 2 750 977 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**

Instrucciones de escalado de vectores para su uso en una unidad aritmética lógica

5 **CAMPO TÉCNICO**

[0001] Esta divulgación se refiere al escalado de vectores en el procesamiento informático.

10 **ANTECEDENTES**

[0002] La normalización vectorial es una operación en un vector que requiere calcular la longitud del vector y dividir a continuación cada componente del vector por la longitud calculada del vector. Si la longitud de un vector tridimensional (x, y, z) se calcula como la raíz cuadrada de  $(x^2+y^2+z^2)$ , dicho cálculo puede desbordar los registros que almacenan los resultados intermedios del cálculo si los valores (x, y, z) del vector son grandes.

15 **SUMARIO**

[0003] Esta divulgación presenta técnicas para el escalado de vectores en el procesamiento informático. De acuerdo con las técnicas de esta divulgación, antes de que se normalice un vector, el vector se puede escalar de modo que calcular la longitud del vector durante la normalización no desborde los registros que almacenan los resultados intermedios de calcular la longitud del vector. Una unidad aritmética lógica (ALU) de una unidad de procesamiento gráfico (GPU) se puede configurar para ejecutar una instrucción de escala de tres ciclos para realizar la reducción de escala de vectores. La instrucción para realizar la reducción de escala de vectores proporcionada por la ALU puede realizar potencialmente la reducción de escala de vectores relativamente más eficazmente que la reducción de escala de vectores basada en software.

[0004] En un ejemplo de la divulgación, un procedimiento para escalar un vector puede incluir recibir, por al menos un procesador, componentes de un vector, en el que cada uno de los componentes del vector comprende al menos un exponente. El procedimiento puede incluir además determinar, por el al menos un procesador, un exponente máximo a partir de exponentes respectivos de los componentes del vector. El procedimiento puede incluir además determinar, por al menos un procesador, un valor de escala en base al menos en parte al exponente máximo. El procedimiento puede incluir además el escalado, por una unidad aritmética lógica (ALU) del al menos un procesador, del vector restando el valor de escala de cada uno de los exponentes respectivos de los componentes del vector.

[0005] En otro ejemplo de la divulgación, un aparato para escalar un vector puede incluir una memoria configurada para almacenar componentes de un vector, en el que cada uno de los componentes del vector comprende al menos un exponente. El aparato puede incluir además al menos un procesador configurado para determinar un exponente máximo a partir de exponentes respectivos de los componentes del vector, y determinar un valor de escala en base al menos en parte al exponente máximo. El aparato puede incluir además una unidad aritmética lógica (ALU) configurada para escalar el vector restando el valor de escala de cada uno de los exponentes respectivos de los componentes del vector.

[0006] En otro ejemplo de la divulgación, un aparato para escalar un vector puede incluir medios para recibir componentes de un vector, en el que cada uno de los componentes del vector comprende al menos un exponente. El aparato puede incluir además medios para determinar un exponente máximo a partir de exponentes respectivos de los componentes del vector. El aparato puede incluir además medios para determinar un valor de escala en base al menos en parte al exponente máximo. El aparato puede incluir además medios para escalar el vector restando el valor de escala de cada uno de los exponentes respectivos de los componentes del vector.

[0007] En otro ejemplo de la divulgación, un medio de almacenamiento legible por ordenador puede almacenar instrucciones que, cuando se ejecuten, provoquen que uno o más procesadores programables: reciban componentes de un vector, en el que cada uno de los componentes del vector comprenda al menos un exponente; determinen un exponente máximo a partir de exponentes respectivos de los componentes del vector; determinen un valor de escala en base al menos en parte al exponente máximo; y escalen el vector, restando el exponente máximo de cada uno de los exponentes respectivos de los componentes del vector.

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

60 **BREVE DESCRIPCIÓN DE LOS DIBUJOS**

[0009]

65 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.

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

5 La FIG. 3 es un diagrama conceptual que ilustra un vector tridimensional de ejemplo que se puede escalar de acuerdo con las técnicas divulgadas en la divulgación.

La FIG. 4 es un diagrama conceptual que ilustra un formato de punto flotante de ejemplo para representar cada componente de un vector.

10

La FIG. 5 es un diagrama de flujo que ilustra un procedimiento de ejemplo para escalar un vector.

## DESCRIPCIÓN DETALLADA

15 **[0010]** En general, esta divulgación describe técnicas para escalar un vector por medio de hardware de modo que una operación de normalización vectorial no desborde los registros que almacenan los resultados intermedios de la operación. En un ejemplo, un procesador puede ejecutar un código de software para escalar un vector. Para un vector tridimensional, el código de software puede incluir un código para determinar el componente más grande del vector tridimensional y dividir cada componente del vector tridimensional por el componente más grande. Sin embargo, el código de software para escalar un vector puede ser más lento que las técnicas basadas en hardware para escalar un vector. Como tal, un enfoque basado en hardware para escalar un vector puede incrementar el rendimiento.

20 **[0011]** Los procesadores, tal como una unidad central de procesamiento (CPU) o una unidad de procesamiento gráfico (GPU), pueden incluir una unidad aritmética lógica (ALU) de hardware. La ALU puede ser un circuito digital que pueda realizar rápidamente operaciones aritméticas y lógicas enteras. Como tal, la ALU puede ser una parte ideal de hardware para escalar vectores más eficazmente. Sin embargo, debido a que la ALU se puede diseñar a menudo para realizar solo operaciones simples tales como suma, resta y operaciones AND y OR, la ALU puede no soportar las operaciones de multiplicación o división necesarias para implementar las técnicas descritas anteriormente para escalar un vector, que incluyen típicamente dividir componentes de un vector por el componente más grande.

30

**[0012]** De acuerdo con los aspectos de la presente divulgación, al menos un procesador, tal como una GPU o una CPU, puede recibir componentes de un vector, en el que cada uno de los componentes del vector comprende al menos un exponente y puede determinar un exponente máximo a partir de exponentes respectivos de los componentes del vector. El al menos un procesador puede determinar además un valor de escala en base al menos en parte al exponente máximo. Una ALU de la CPU o de la GPU puede escalar el vector, restando el factor de escala de cada uno de los exponentes respectivos de los componentes del vector.

35

40 **[0013]** 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. Como se muestra en la FIG. 1, el dispositivo informático 2 puede ser un dispositivo informático que incluya, pero sin limitarse a, dispositivos de vídeo, reproductores multimedia, decodificadores, teléfonos inalámbricos tales como teléfonos móviles y los llamados teléfonos inteligentes, asistentes digitales personales (PDA), ordenadores de escritorio, ordenadores portátiles, consolas de videojuegos, unidades de videoconferencia, dispositivos informáticos de tablets, y similares. En el ejemplo de la FIG. 1, el dispositivo informático 2 puede incluir la unidad central de procesamiento (CPU) 6, la memoria de sistema 10 y la GPU 12. El dispositivo informático 2 también puede incluir el procesador de pantalla 14, el módulo transceptor 3, la interfaz de usuario 4 y la pantalla 8. El módulo transceptor 3 y el procesador de pantalla 14 pueden formar parte ambos del mismo circuito integrado (IC) que la CPU 6 y/o la GPU 12, pueden ser externos ambos al IC o a los IC que incluyen la CPU 6 y/o la GPU 12, o se pueden formar en el IC que es externo al IC que incluye la CPU 6 y/o la GPU 12.

45

50 **[0014]** El dispositivo informático 2 puede incluir módulos o unidades adicionales no mostrados en la FIG. 1 para los propósitos de 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 donde el dispositivo informático 2 es un teléfono móvil inalámbrico, o un altavoz donde el dispositivo informático 2 es un reproductor multimedia. El dispositivo informático 2 también puede incluir una cámara de vídeo. Además, los diversos módulos y unidades mostrados en el dispositivo informático 2 pueden no ser necesarios en cada ejemplo del dispositivo informático 2. Por ejemplo, la interfaz de usuario 4 y la pantalla 8 pueden ser externas al dispositivo informático 2 en ejemplos donde el dispositivo informático 2 sea un ordenador de escritorio u otro dispositivo que esté equipado para interconectarse con una interfaz de usuario o pantalla externos.

55

60 **[0015]** Los 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 la pantalla 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, demoduladores, amplificadores y otros dichos circuitos para la comunicación alámbrica o inalámbrica.

65

**[0016]** El procesador 6 puede ser un microprocesador, tal como una unidad central de procesamiento (CPU) configurada para procesar instrucciones de un programa informático para su ejecución. El procesador 6 puede comprender un procesador de uso general o de uso especial que controle el funcionamiento del dispositivo informático 2. Un usuario puede proporcionar entrada al dispositivo informático 2 para provocar que el procesador 6 ejecute una o más aplicaciones de software. Entre las aplicaciones de software que se ejecutan en el procesador 6 se 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 o cualquier otro tipo de programa. Adicionalmente, el procesador 6 puede ejecutar el accionador de GPU 22 para controlar la operación de la GPU 12. El usuario puede proporcionar entrada al dispositivo informático 2 por medio de uno o más dispositivos de entrada (no mostrados) tal 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.

**[0017]** Las aplicaciones de software que se ejecutan en el procesador 6 pueden incluir una o más instrucciones de representación de gráficos que indiquen al procesador 6 que provoque la representación de datos de gráficos en la pantalla 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 Incrustados de Librería Abierta de Gráficos (OpenGL ES), una API de Direct3D, una API X3D, una API RenderMan, una API WebGL o cualquier otra API gráfica estándar pública o patentada. Para procesar las instrucciones de representación gráfica, el procesador 6 puede emitir uno o más comandos de representación gráfica a la GPU 12 (por ejemplo, a través del accionador de GPU 22) para provocar que la GPU 12 realice una parte o la totalidad de la renderización de los datos gráficos. En algunos ejemplos, entre los datos gráficos que se vayan a renderizar, se puede incluir una lista de primitivas gráficas, por ejemplo, puntos, líneas, triángulos, cuadriláteros, tiras de triángulos, etc.

**[0018]** La GPU 12 se puede configurar para realizar operaciones gráficas para renderizar una o más primitivas gráficas en la pantalla 8. Por tanto, cuando una de las aplicaciones de software que se ejecuta en el procesador 6 requiere procesamiento gráfico, el procesador 6 puede proporcionar comandos gráficos y datos gráficos a la GPU 12 para que se renderice en la pantalla 8. Los datos gráficos pueden incluir, por ejemplo, comandos de representación, información de estado, información primitiva, información de textura, etc. La GPU 12 se puede construir, en algunos casos, con una estructura altamente paralela que proporcione un procesamiento más eficiente de operaciones complejas relacionadas con gráficos que el procesador 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 la pantalla 8 más rápidamente que la representación de las escenas directamente en la pantalla 8 usando el procesador 6.

**[0019]** 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 interoperar 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 por campo (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.

**[0020]** La GPU 12 se puede acoplar directamente a la memoria gráfica 40. Por tanto, la GPU 12 puede leer y escribir datos desde 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 sin 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 borrable (EPROM), ROM programable borrable eléctricamente (EEPROM), memoria Flash, un medio de datos magnéticos o un medio de almacenamiento óptico.

**[0021]** En algunos ejemplos, la GPU 12 puede almacenar una imagen completamente formada en la memoria de sistema 10. El procesador de pantalla 14 puede recuperar la imagen de la memoria de sistema 10 y generar valores que provoquen que los píxeles de la pantalla 8 se iluminen para visualizar la imagen. La pantalla 8 puede visualizar el dispositivo informático 2 que visualiza el contenido de imagen generado por la GPU 12. La pantalla 8 puede comprender una pantalla de cristal líquido (LCD), una pantalla de diodo orgánico emisor de luz (OLED), una pantalla de tubo de rayos catódicos (CRT), una pantalla de plasma, u otro tipo de dispositivo de pantalla.

**[0022]** Como se analiza anteriormente, la GPU 12 puede incluir una ALU 24, que puede ser un circuito digital que realice operaciones aritméticas enteras, de punto flotante y lógicas. Las operaciones que puede realizar la ALU 24 pueden incluir operaciones de suma, resta y bit a bit. En algunos ejemplos, la ALU 24 puede no ser capaz de realizar operaciones tales como la multiplicación y la división. En algunos ejemplos, el procesador 6 también puede incluir una ALU que puede funcionar de forma similar a la ALU 24 en que puede ser un circuito digital que realiza operaciones aritméticas y lógicas.

**[0023]** La FIG. 2 es un diagrama de bloques que ilustra con mayor detalle implementaciones de ejemplo del procesador 6, de la GPU 12 y de la memoria de sistema 10 de la FIG. 1. Como se muestra en la FIG. 2, el procesador 6 puede incluir al menos una aplicación de software 18, una API gráfica 20 y un accionador de GPU 22, cada uno de los cuales puede ser una o más aplicaciones de software o servicios que se ejecutan en el procesador 6.

**[0024]** La memoria disponible para el procesador 6 y la GPU 12 puede incluir la memoria de sistema 10 y la memoria intermedia de tramas 16. La memoria intermedia de tramas 16 puede formar parte de la memoria de sistema 10 o puede estar separada de la memoria de sistema 10. La memoria intermedia de tramas 16 puede almacenar datos de imagen representados.

**[0025]** 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 de funcionamiento, una aplicación de mapeo 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.

**[0026]** La aplicación de software 18 puede incluir una o más instrucciones de representación que indiquen a la GPU 12 que represente una interfaz de usuario gráfica (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 para representarlas 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.

**[0027]** La aplicación de software 18 puede invocar el accionador de GPU 22, por medio de la API gráfica 20, para emitir uno o más comandos a la GPU 12 para representar una o más primitivas gráficas en imágenes gráficas visualizables. Por ejemplo, la aplicación de software 18 puede invocar el accionador de GPU 22, por medio de la API gráfica 20, para proporcionar definiciones primitivas a la GPU 12. En algunos casos, las definiciones primitivas se pueden proporcionar a la GPU 12 en forma de una lista de primitivas de representación, por ejemplo, triángulos, rectángulos, abanicos triangulares, tiras triangulares, etc. Las definiciones primitivas pueden incluir especificaciones de vértices que especifiquen uno o más vértices asociados con las primitivas que se rendericen. Las especificaciones de los vértices pueden incluir coordenadas de posición para cada vértice y, en algunos casos, otros atributos asociados con el vértice, tales como, por ejemplo, coordenadas de color, vectores normales y coordenadas de textura. Las definiciones primitivas también pueden incluir información de tipo primitiva (por ejemplo, triángulo, rectángulo, abanico de triángulo, tira de triángulo, etc.), información de escala, información de rotación, y similares.

**[0028]** En base a las instrucciones emitidas por la aplicación de software 18 para el accionador de GPU 22, el accionador 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 de la CPU 6, una tubería de procesamiento gráfico decodifica el comando y configura la tubería de procesamiento gráfico para realizar la operación especificada en el comando. Por ejemplo, un ensamblador de entrada en la tubería de procesamiento gráfico puede leer datos primitivos y ensamblar los datos en primitivas para su uso por las otras etapas de la tubería de gráficos en una tubería de procesamiento gráfico. Después de realizar las operaciones especificadas, la tubería de procesamiento gráfico genera los datos renderizados a la memoria intermedia de tramas 16 asociada con un dispositivo de visualización.

**[0029]** La memoria intermedia de tramas 16 almacena píxeles de destino para la GPU 12. Cada píxel de destino puede estar asociado con una localización única de píxel de pantalla. En algunos ejemplos, la memoria intermedia de tramas 16 puede almacenar componentes de color y un valor alfa de destino para cada píxel de destino. Por ejemplo, la memoria intermedia de tramas 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. Aunque la memoria intermedia de tramas 16 y la memoria de sistema 10 se ilustran como unidades de memoria separadas, en otros ejemplos, la memoria intermedia de tramas 16 puede formar parte de la memoria de sistema 10.

**[0030]** En algunos ejemplos, una tubería de procesamiento gráfico puede incluir una o más de una etapa de sombreado de vértices, de una etapa de sombreado de casco, de una etapa de sombreado de dominio, de una etapa de sombreado de geometría y de una etapa de sombreado de píxeles. Estas etapas de la tubería de procesamiento gráfico se pueden considerar etapas de sombreado. Estas etapas de sombreado se pueden implementar como uno o más programas de sombreado que se ejecutan en las unidades de sombreado 46 en la GPU 12. Las unidades de

sombreado 46 pueden comprender una o más unidades de sombreado configuradas como una tubería programable de componentes de procesamiento. En algunos ejemplos, las unidades de sombreado 46 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.

5 **[0031]** La GPU 12 puede designar unidades de sombreado 46 para realizar una variedad de operaciones de sombreado, tales como sombreado de vértices, sombreado de casco, sombreado de dominio, sombreado de geometría, sombreado de píxeles, y similares, enviando comandos a las unidades de sombreado 46 para ejecutar una o más de una etapa de sombreado de vértice, de una etapa de sombreado de casco, de una etapa de sombreado de dominio, de una etapa de sombreado de geometría y de una etapa de sombreado de píxeles en una tubería de procesamiento gráfico. En algunos ejemplos, el accionador de GPU 22 se puede configurar para compilar uno o más programas de sombreado y para descargar los programas de sombreado compilados en una o más unidades de sombreado programables contenidas dentro de 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 C para el lenguaje de sombreado de Gráficos (Cg), etc. Los programas de sombreado compilados pueden incluir una o más instrucciones que controlen el funcionamiento de las unidades de sombreado 46 dentro de la GPU 12. Por ejemplo, los programas de sombreado pueden incluir programas de sombreado de vértices que se pueden ejecutar por las unidades de sombreado 46 para realizar las funciones de una etapa de sombreado de vértices, programas de sombreado de casco que se pueden ejecutar por las unidades de sombreado 46 para realizar las funciones de una etapa de sombreado de casco, programas de sombreado de dominio que se pueden ejecutar por la unidad de sombreado 46 para realizar las funciones de una etapa de sombreado de dominio, programas de sombreado de geometría que se pueden ejecutar por la unidad de sombreado 46 para realizar las funciones de una etapa de sombreado de geometría y/o programas de sombreado de píxeles que se pueden ejecutar por las unidades de sombreado 46 para realizar las funciones de un sombreado de píxeles. 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.

30 **[0032]** Las unidades de sombreado 46 pueden incluir núcleos de procesador 48, cada uno de los cuales puede incluir uno o más componentes para buscar y decodificar operaciones, una o más unidades lógicas aritméticas para llevar a cabo cálculos aritméticos, una o más memorias, memorias caché y registros. En algunos ejemplos, los núcleos de procesador 48 también se pueden denominar elementos de procesamiento escalar. Cada uno de los núcleos de procesador 48 puede incluir registros de uso general 25. Los registros de uso general 25 pueden almacenar datos a los que la ALU 24 puede acceder directamente en los núcleos de procesador 48. Por ejemplo, los registros de uso general 25 pueden almacenar los componentes de vector que se vayan a escalar por la ALU 24 y también pueden almacenar los componentes de vector escalados emitidos por la ALU 24.

40 **[0033]** Cada uno de los núcleos de procesador 48 puede incluir una ALU escalar, tal como la ALU 24. Como se analiza anteriormente, la ALU 24 puede ser un circuito digital que realice operaciones aritméticas enteras, de punto flotante y lógicas. Las operaciones que puede realizar la ALU 24 pueden incluir operaciones de suma, resta y bit a bit. En algunos ejemplos, la ALU 24 puede no ser capaz de realizar operaciones tales como la multiplicación y la división. De acuerdo con los aspectos de la presente divulgación, la ALU 24 puede escalar un vector escalando los componentes del vector. La ALU 24 también puede generar los componentes escalados del vector a la memoria gráfica 40 o registros de uso general 25.

45 **[0034]** La memoria gráfica 40 es almacenamiento en chip o memoria que se integró físicamente en el circuito integrado de la GPU 12. Debido a que la memoria gráfica 40 está en chip, la GPU 12 puede leer valores de o escribir valores en la memoria gráfica 40 más rápidamente que leer valores de o escribir valores en la memoria de sistema 10 por medio de un bus de sistema. La memoria gráfica 40 puede almacenar componentes de un vector y también puede almacenar componentes escalados de un vector después de escalar por la ALU 24. La memoria gráfica 40 puede almacenar los componentes del vector en formato de punto flotante, de modo que cada componente se puede almacenar en la memoria gráfica 40 como un bit con signo, un significando y un exponente.

55 **[0035]** La FIG. 3 es un diagrama de bloques que ilustra un vector tridimensional de ejemplo que se puede escalar por la CPU 6 o la GPU 12. Como se muestra en la FIG. 3, el vector 50 en el sistema cartesiano de coordenadas tridimensionales 52 se puede representar por medio de una tupla (x, y, z) que indique el valor de los componentes respectivos 54A-54C ("componentes 54") del vector 50. Los componentes 54 del vector 50 pueden incluir el componente x 54A, el componente y 54B y el componente z 54C.

60 **[0036]** Como se analiza anteriormente, cada componente de los componentes 54 del vector 50 puede ser un valor de punto flotante que se almacene en la memoria gráfica 40 o registros de uso general 25 como un bit con signo, un significando y un exponente. Por ejemplo, un valor de punto flotante 1.2345 de ejemplo puede ser igual a  $12345 \times 10^{-4}$ , de modo que 12345 puede ser el significando o mantisa y -4 puede ser el exponente de base 10. En otros ejemplos, el exponente puede ser un exponente de base 2. Para representar valores de exponente negativos, el exponente se puede desviar o desplazar, de modo que los valores de exponente se conviertan en valores positivos. Por ejemplo, el valor 15 se puede agregar a un exponente de modo que un valor exponente de -4 se pueda almacenar en la memoria como 11.

**[0037]** La FIG. 4 es un diagrama conceptual que ilustra un formato de punto flotante de ejemplo para representar cada componente de los componentes 54 del vector 50. Como se analiza anteriormente, cada componente de los componentes 54 del vector 50 puede ser un valor de punto flotante. Como se muestra en la FIG. 4, cada componente de los componentes 54 se puede representar en formato de punto flotante 60. El formato de punto flotante 60 puede incluir el bit de signo 62 que indica el signo del valor de punto flotante representado por el formato de punto flotante 60. El bit de signo 62 puede ser 1 si el signo del valor de punto flotante es negativo y puede ser 0 si el signo del valor de punto flotante es positivo. El formato de punto flotante 60 puede incluir además el exponente 64 y el significando 66. En un ejemplo, para un formato de punto flotante de IEEE de 32 bits 60, el bit de signo 62 puede ser de un bit, el exponente 64 puede ser de 8 bits con una desviación de 127, y el significando 66 puede ser de 23 bits con el número entero oculto. Por ejemplo, el valor de punto flotante -82.3125 puede ser igual a  $-1.0100100101_2 \cdot 2^6$ . En este ejemplo, el bit de signo 62 se puede establecer en 1. El exponente 64 puede ser de 10000101, que es 133,10 debido a una desviación de 127, y el significando 66 puede ser de 0100100101000000000000 porque el bit entero puede estar oculto.

**[0038]** De acuerdo con los aspectos de la presente divulgación, el procesador 6 o la GPU 12 pueden usar la ALU 24 para escalar el vector 50 de modo que el procesador 6 o la GPU 12 puedan realizar la normalización vectorial del vector 50 sin desbordar los registros que almacenan los resultados intermedios de la operación de normalización vectorial. Debido a que la ALU 24 puede ser un circuito de hardware, dicha escala del vector 50 se puede realizar en hardware en lugar de realizarse en software que se ejecuta por, por ejemplo, unidades de sombreado 46. Además, debido a que la ALU 24 puede incluir funcionalidad para realizar operaciones de suma y resta, pero puede no incluir funcionalidad para realizar operaciones de multiplicación y/o división, la ALU 24 puede escalar el vector 50 sin realizar operaciones de multiplicación o división. Para escalar el vector 50, la GPU 12 puede recibir los componentes 54 del vector 50. Por ejemplo, si el vector 50 es un vector tridimensional, la GPU 12 puede recibir el componente x 54A, el componente y 54B y el componente z 54C del vector 50. Los componentes 54 del vector 50 se pueden almacenar en la memoria, tales como la memoria de sistema 10, la memoria gráfica 40, la memoria de las unidades de sombreado 46, los registros de uso general 25, y similares.

**[0039]** Como se analiza anteriormente, los componentes 54 del vector 50 pueden ser cada uno un valor de punto flotante que incluya al menos un significando y un exponente. La GPU 12 puede determinar el exponente máximo a partir de los exponentes de los componentes 54. Por ejemplo, si los exponentes de los componentes 54 son -1, 2 y 5, la GPU 12 puede determinar que el exponente máximo de los exponentes a partir de los componentes 54 es 5. En algunos ejemplos, determinar el exponente máximo a partir de los exponentes de los componentes 54 puede incluir determinar el exponente de valor máximo a partir de los exponentes de los componentes 54. Por tanto, por ejemplo, si los exponentes de los componentes 54 son -1, 2 y 5, la GPU 12 puede determinar que el exponente de valor máximo a partir de los componentes 54 es de 5, porque 5 es mayor que 2 o -1.

**[0040]** En respuesta a la GPU 12 que determina el exponente máximo a partir de los exponentes del componente 54, la GPU 12 puede determinar un valor de escala para escalar cada uno de los exponentes del componente 54. En un ejemplo, el valor de escala puede ser igual al exponente máximo, de modo que el valor de escala puede ser 5 para los exponentes de los componentes 54. En otro ejemplo, la GPU 12 puede determinar el valor de escala para evitar el subflujo y/o el desbordamiento de los exponentes escalados después de la escala. En este caso, la GPU 12 puede determinar el valor de escala en base al menos en parte al exponente máximo. Por ejemplo, el valor de escala puede ser el exponente máximo + una constante. Por ejemplo, el valor de escala puede ser el exponente máximo - (maximum\_representable\_exponent - 1)/2 + 1. El maximum\_representable\_exponent puede ser un exponente representable máximo que sea una constante derivada del formato de punto flotante de los componentes 54. Por ejemplo, para números de punto flotante del IEEE de 32 bits, el exponente representable máximo es 128. El intervalo de exponente es [-127, 128] (es decir, de -127 a 128 incluidos) porque el exponente en un número de punto flotante de 32 bits se representa con 8 bits. En otro ejemplo, la GPU 12 puede determinar el valor de escala para ser (exponente máximo - 1)/2 - 2. Por lo tanto, dado un exponente máximo de 15, el valor de escala puede ser (15-1)/2-2, que puede ser 5.

**[0041]** En respuesta a la GPU 12 que determina el valor de escala, la ALU 24 se puede configurar para escalar cada componente de los componentes 54 restando el valor de escala de cada exponente de los componentes 54. Por ejemplo, dados los valores de exponente de -1, 2, un 5 para el componente x 54A, el componente y 54B y el componente z 54C del vector 50, dado que la GPU 12 determina que 5 es el exponente máximo a partir de los exponentes de los componentes 54, y dado que la GPU 12 determina que el valor de escala es el valor del exponente máximo (es decir, establece el valor de escala en 5), la ALU 24 puede restar 5 del valor de exponente de -1 para el componente x 54A, la ALU 24 puede restar 5 del valor de exponente de 2 para el componente y 54B, y la ALU 24 puede restar 5 del valor de exponente de 5 para el componente z 54C, dando como resultado que los componentes escalados 54 tengan valores de exponente de -6, -3 y 0, respectivamente para el componente x 54A, el componente y 54B y el componente z 54C. Los componentes escalados resultantes que incluyen los exponentes emitidos por la ALU 24 se pueden almacenar en la memoria, tal como la memoria gráfica 40, la memoria de sistema 10, los registros de uso general 25 y similares.

**[0042]** Como se analiza anteriormente, en algunos ejemplos, los exponentes de los componentes 54 pueden ser exponentes desviados. LA GPU 12 y la ALU 24 pueden manejar componentes desviados de forma similar a los exponentes no desviados. Por ejemplo, si los valores de los exponentes de los componentes 54 de -1, 2 y 5 se desvían por 15, de modo que se agrega 15 a cada exponente de los componentes 54, los valores de los exponentes desviados del componente 54 pueden ser de 14, 17 y 20. En consecuencia, la GPU 12 puede determinar que el exponente máximo a partir de los exponentes desviados de los componentes 54 sea 20. En respuesta a la GPU 12 que determina el exponente máximo a partir de los exponentes desviados del componente 54, la GPU 12 puede determinar un valor de escala en base al menos en parte al exponente máximo. En este ejemplo, la GPU 12 puede establecer el valor de escala al valor del exponente máximo. En respuesta a la GPU 12 que determina el valor de escala, la ALU 24 puede escalar cada componente de los componentes 54 restando el valor de escala de cada exponente de los componentes 54. Por ejemplo, valores de exponente desviados dados de 14, 17 y 20 para el componente x 54A, el componente y 54B y el componente z 54C del vector 50, y dado que la GPU 12 determina que 20 es el exponente máximo a partir de los exponentes de los componentes 54 y que el valor de escala se establece en el valor del exponente máximo, la ALU 24 puede restar 20 del valor de exponente de 14 para el componente x 54A, la ALU 24 puede restar 20 del valor de exponente de 17 para el componente y 54B, y la ALU 24 puede restar 20 del valor de exponente de 20 para el componente z 54C. La ALU 24 puede añadir una desviación de 15 a cada exponente de los componentes 54, dando como resultado que los componentes escalados 54 tengan valores de exponente desviados de 9, 12 y 15, respectivamente para el componente x 54A, el componente y 54B y el componente z 54C.

**[0043]** La ALU 24 se puede configurar para generar un componente escalado por ciclo de reloj, de modo que la ALU 24 pueda generar el componente x escalado en un primer ciclo de reloj, el componente y escalado en un segundo ciclo de reloj y el componente z escalado en un tercer ciclo de reloj. El pseudocódigo de ejemplo para realizar el escalado del vector 50 se puede expresar como sigue:

```

25     if (src0, src1 o src2 es INF o NaN) { // output = input
        dst = src0;
        dst+1 = src1;
30     dst+2 = src2;
    }
    else maxexp = max (src0,exp, src1,exp, src2,exp);
35     if (src0 is 0 or denorm)
        dst = 0 preserving src0 sign;
40     else dst.exp = src0,exp - maxexp;
        if (src1 is 0 or denorm)
45     dst = 0 preserving src1 sign;
        else (dst + 1).exp = src1,exp - maxexp;
        if (src2 is 0 or denorm)
50     dst = 0 preserving src2 sign;
        else (dst + 2).exp = src2,exp - maxexp;

```

**[0044]** Como se muestra en el pseudocódigo anteriormente, src0, src1 y src2 pueden ser las localizaciones de memoria de origen de los componentes de un vector tridimensional, dst puede ser la localización de destino en la memoria para un primer componente escalado, dst+1 puede ser la próxima localización de destino consecutiva en memoria para un segundo componente escalado, y dst+2 puede ser la siguiente localización de destino consecutiva en memoria para un tercer componente escalado. Como se puede observar, los componentes escalados se pueden almacenar en localizaciones de memoria consecutivas en la memoria.

**[0045]** Como se muestra anteriormente, la GPU 12 puede determinar si alguno de los componentes es infinito o no es un número. Un componente puede no ser un número si el componente es un valor indefinido o no representable. Un componente que no sea un número puede tener el exponente 64 lleno de 1s y el significando 66 puede ser un valor distinto de cero. Si es así, los componentes del vector no están escalados. De otro modo, la GPU 12 puede determinar un valor de escala en base al menos en parte al exponente máximo a partir de los exponentes de los componentes. Para cada uno de los componentes en src0, src1 y src2, si el exponente es cero o un número denormal,

entonces el componente no se escala. Un número denormal puede ser cualquier número distinto de cero con una magnitud que sea menor que el número normal más pequeño. De otro modo, la ALU 24 puede escalar el componente restando el exponente del componente por el valor de escala. En algunos ejemplos, cualquiera de las localizaciones de memoria de destino dst, dst+1 o dst+2 se puede superponer con las localizaciones de memoria de origen src0, src1 o src2.

**[0046]** La GPU 12 puede proporcionar una función de escala para generar componentes escalados de un vector como (rpt2) scale.x, (r)x, (x, y, z);. (rpt2) puede indicar que la instrucción de escala se repetirá dos veces después de la ejecución inicial, de modo que puede ejecutar un total de tres veces para generar componentes escalados x, y y z de un vector tridimensional, scale.x puede ser el nombre de función para la función de escalado, donde la x en scale.x puede ser la localización inicial de las localizaciones de memoria de origen que almacenan los componentes sin escala del vector. (r)x puede indicar que la instrucción de escala se repetirá, y la x en (r)x puede ser la localización inicial de las localizaciones de memoria de destino para almacenar los componentes escalados del vector, (x, y, z) pueden ser los componentes del vector que se vayan a escalar por la función de escala. El rendimiento para la ALU 24, que puede ser el número de instrucciones de escala que se pueden emitir a la tubería de la ALU 24 en un período de tiempo determinado, puede ser de 3 ciclos por instrucción de escala. La latencia de la ALU 24 en la realización de la instrucción de escala, que puede ser la cantidad total de tiempo que se tarda en ejecutar la instrucción de escala desde su emisión hasta su finalización, puede depender de la implementación de la instrucción de escala.

**[0047]** La FIG. 5 es un diagrama de flujo que ilustra un procedimiento de ejemplo para escalar un vector. Como se muestra en la FIG. 5, el proceso puede incluir recibir, por el procesador 6 o la GPU 12, componentes de un vector, en el que cada uno de los componentes del vector comprende al menos un exponente; (502). El proceso puede incluir además determinar, por el procesador 6 o la GPU 12, un exponente máximo a partir de exponentes respectivos de los componentes del vector (504). El proceso puede incluir además determinar, por el procesador 6 o la GPU 12, un valor de escala en base al menos en parte al componente máximo (506). El proceso puede incluir además escalar, por la ALU 24 del procesador 6 o la GPU 12, el vector restando el valor de escalado de cada uno de los exponentes respectivos de los componentes del vector (508).

**[0048]** En algunos ejemplos, escalar el vector puede incluir además escalar, por la ALU 24, el vector sin realizar una operación de multiplicación ni una operación de división. En algunos ejemplos, cada uno de los componentes del vector puede ser un número de punto flotante, y en el que el número de punto flotante se puede representar como un bit de signo, un significando y el exponente. En algunos ejemplos, el vector puede comprender un vector tridimensional, y los componentes del vector pueden comprender un componente x, un componente y y un componente z. En algunos ejemplos, escalar el vector puede incluir escalar, por la ALU 24, el componente x del vector restando el valor de escala de un primer exponente del componente x del vector en un primer ciclo de reloj, escalando, por la ALU 24, el componente y del vector restando el valor de escala de un segundo exponente del componente y del vector en un segundo ciclo de reloj, y escalando, por la ALU 24, el componente z del vector restando el valor de escala de un tercer exponente del componente z del vector en un tercer ciclo de reloj. En algunos ejemplos, el proceso puede incluir además generar el componente x escalado, el componente y escalado y el componente z escalado en localizaciones de almacenamiento consecutivas en la memoria. En algunos ejemplos, la ALU 24 puede ser un circuito digital de hardware.

**[0049]** En algunos ejemplos, determinar un valor de escala en base al menos en parte al exponente máximo puede incluir determinar el valor de escala como el exponente máximo. En algunos otros ejemplos, la determinación de un valor de escala en base al menos en parte al exponente máximo puede incluir la determinación del valor de escala en base al menos en parte al exponente máximo y a un exponente máximo representativo.

**[0050]** 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 esta 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 adecuadamente la denominación de medio legible por ordenador. Por ejemplo, si el software se transmite desde una página 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. Las combinaciones de lo anterior se deberían incluir también dentro del alcance de los medios legibles por ordenador.

5 **[0051]** El código se puede ejecutar por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de uso general, circuitos integrados específicos de la aplicación (ASIC), matrices lógicas programables por campo (FPGA) u otros circuitos lógicos, integrados o discretos, equivalentes. 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 dentro de módulos de hardware y/o programa informático dedicados configurados para codificar y decodificar, o incorporarse en un códec combinado. También, las técnicas se podrían implementar por completo en uno o más circuitos o elementos lógicos.

15 **[0052]** Las técnicas de esta divulgación se pueden implementar en una gran diversidad de dispositivos o aparatos, incluyendo 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 requiere necesariamente la realización mediante diferentes unidades de hardware. En su lugar, como se describe anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionar por un grupo de unidades de hardware interoperativas, incluyendo uno o más procesadores, como se describe anteriormente, conjuntamente con un programa informático y/o firmware adecuados.

20 **[0053]** Se han descrito diversos ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

25

**REIVINDICACIONES**

1. Un procedimiento para escalar un vector, comprendiendo el procedimiento:
  - 5 recibir, por al menos un procesador, componentes de un vector, en el que cada uno de los componentes del vector comprende al menos un exponente;
  - determinar, por el al menos un procesador, un exponente máximo a partir de exponentes respectivos de los componentes del vector; y
  - 10 escalar, por una unidad aritmética lógica (ALU) del al menos un procesador, el vector, incluyendo para cada exponente de los exponentes respectivos de los componentes del vector:
    - 15 establecer el componente respectivo en cero si el componente respectivo es cero o un número denormal; y restar el exponente máximo del exponente del componente respectivo si el componente respectivo no es cero y no es un número denormal.
2. El procedimiento de la reivindicación 1, en el que cada uno de los componentes del vector comprende un número de punto flotante, y en el que el número de punto flotante se representa como un bit de signo, un significando y el exponente.
3. El procedimiento de la reivindicación 1, en el que:
  - 25 el vector comprende un vector tridimensional; y
  - los componentes del vector comprenden un componente x, un componente y y un componente z.
4. El procedimiento de la reivindicación 3, en el que escalar el vector comprende además:
  - 30 escalar, por la ALU, el componente x del vector restando el exponente máximo de un primer exponente del componente x del vector en un primer ciclo de reloj;
  - escalar, por la ALU, el componente y del vector restando el exponente máximo de un segundo exponente del componente y del vector en un segundo ciclo de reloj; y
  - 35 escalar, por la ALU, el componente z del vector restando el exponente máximo de un tercer exponente del componente z del vector en un tercer ciclo de reloj.
5. El procedimiento de la reivindicación 4, que comprende además:
  - 40 generar el componente x escalado, el componente y escalado y el componente z escalado en localizaciones de almacenamiento consecutivas en la memoria.
6. El procedimiento de la reivindicación 1, en el que la ALU comprende un circuito digital de hardware.
7. Un aparato para escalar un vector, comprendiendo el aparato:
  - 50 una memoria configurada para almacenar componentes de un vector, en la que cada uno de los componentes del vector comprende al menos un exponente;
  - al menos un procesador configurado para:
    - determinar un exponente máximo a partir de exponentes respectivos de los componentes del vector, y
    - 55 una unidad aritmética lógica (ALU) configurada para escalar el vector, incluyendo para cada exponente de los exponentes respectivos de los componentes del vector:
      - 60 establecer el componente respectivo en cero si el componente respectivo es cero o un número denormal; y
      - restar el exponente máximo del exponente del componente respectivo si el componente respectivo no es cero y no es un número denormal.
8. El aparato de la reivindicación 7, en el que cada uno de los componentes del vector comprende un número de punto flotante, y en el que el número de punto flotante se representa como un bit de signo, un significando y el exponente.

9. El aparato de la reivindicación 7, en el que:  
el vector comprende un vector tridimensional; y
- 5 los componentes del vector comprenden un componente x, un componente y y un componente z.
10. El aparato de la reivindicación 9, en el que la ALU está configurada para:  
escalar el componente x del vector restando el exponente máximo de un primer exponente del componente x del vector en un primer ciclo de reloj;
- 10 escalar el componente y del vector restando el exponente máximo de un segundo exponente del componente y del vector en un segundo ciclo de reloj; y
- 15 escalar el componente z del vector restando el exponente máximo de un tercer exponente del componente z del vector en un tercer ciclo de reloj.
11. El aparato de la reivindicación 10, en el que la ALU está configurada para:  
generar el componente x escalado, el componente y escalado y el componente z escalado en localizaciones de almacenamiento consecutivas en la memoria.
- 20 12. El aparato de la reivindicación 7, en el que la ALU comprende un circuito digital de hardware.
- 25 13. Un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando se ejecutan por uno o más procesadores, provocan que uno o más procesadores realicen el procedimiento de acuerdo con una cualquiera de las reivindicaciones 1 a 6.

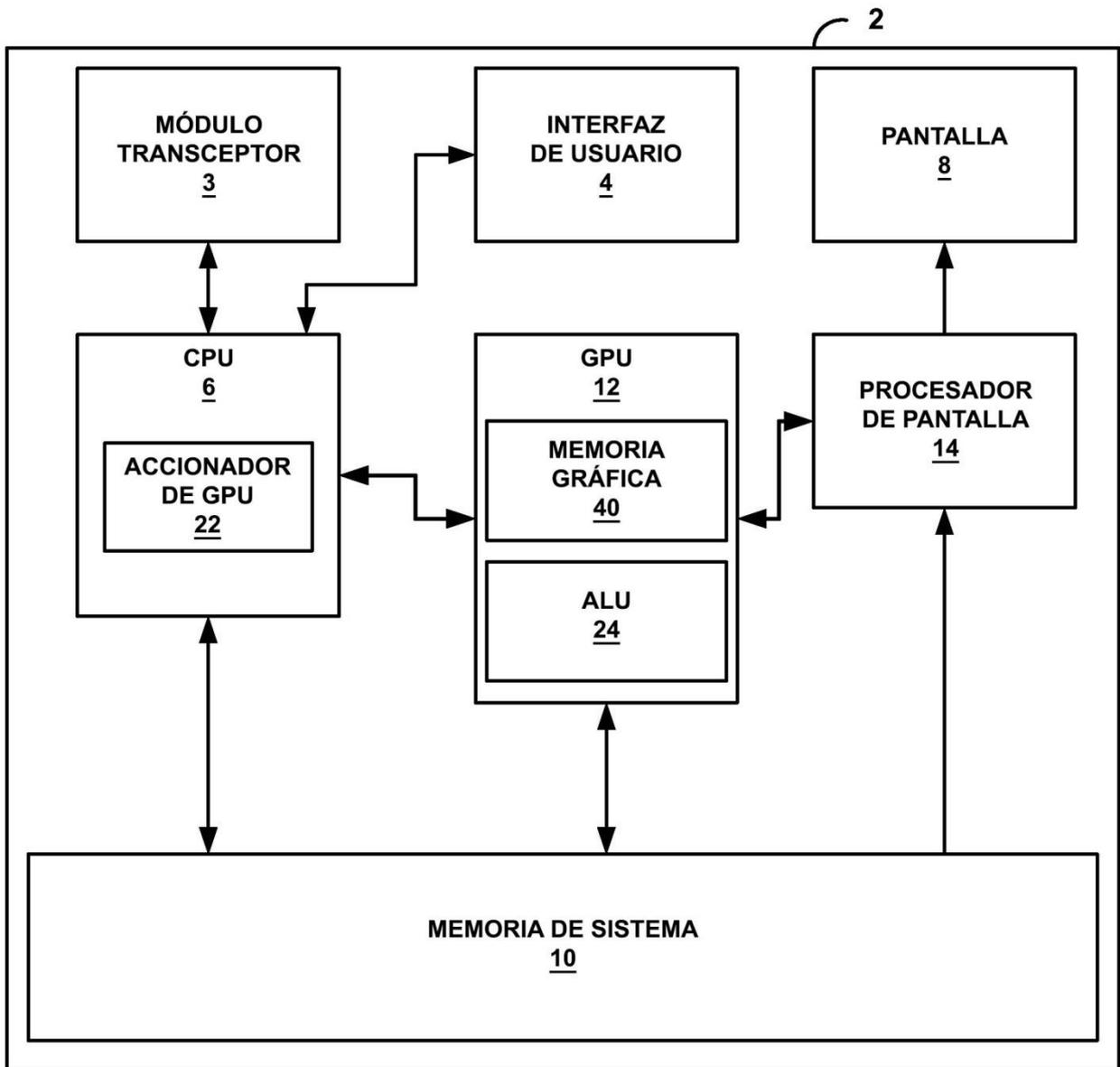


FIG. 1

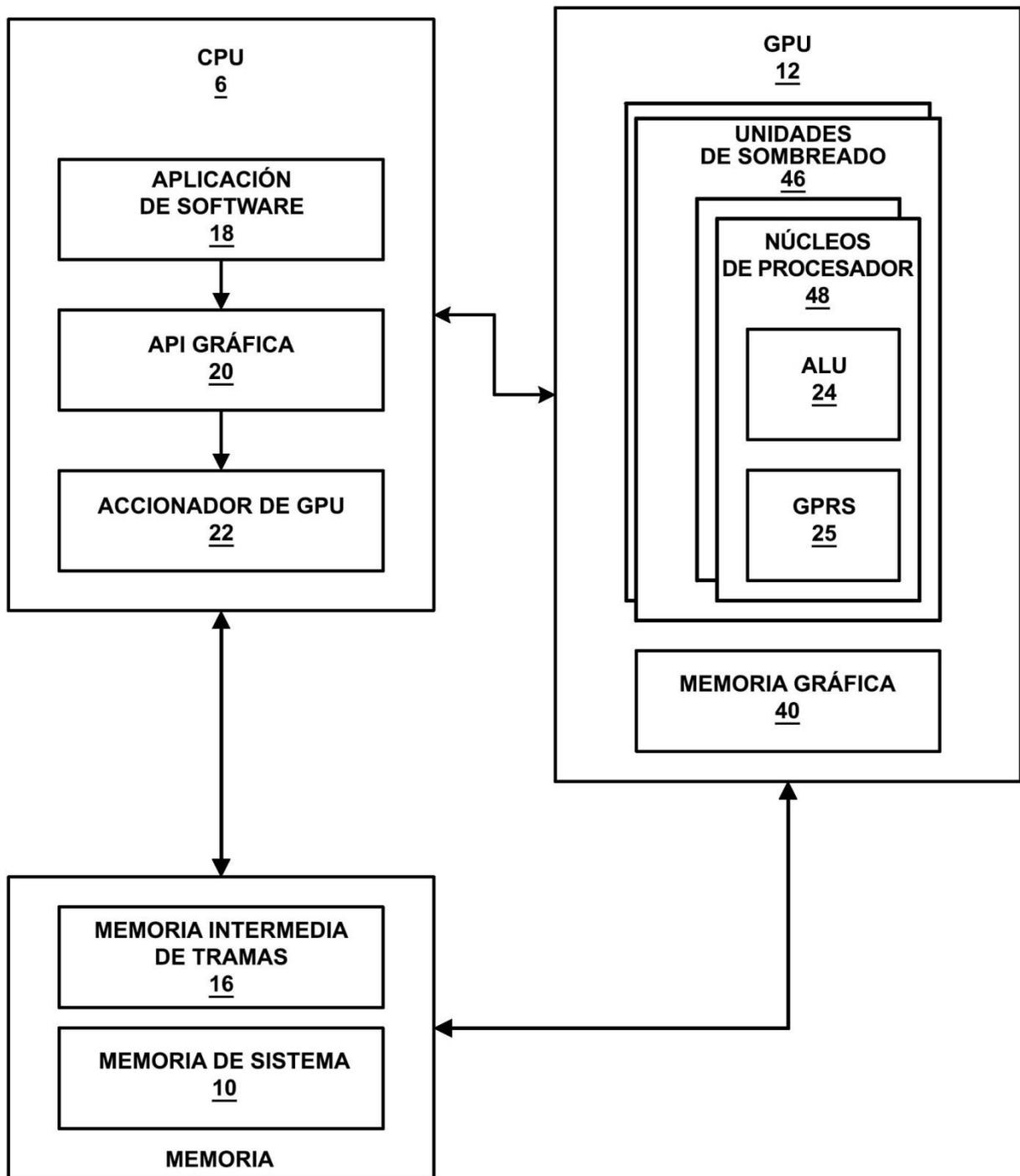


FIG. 2

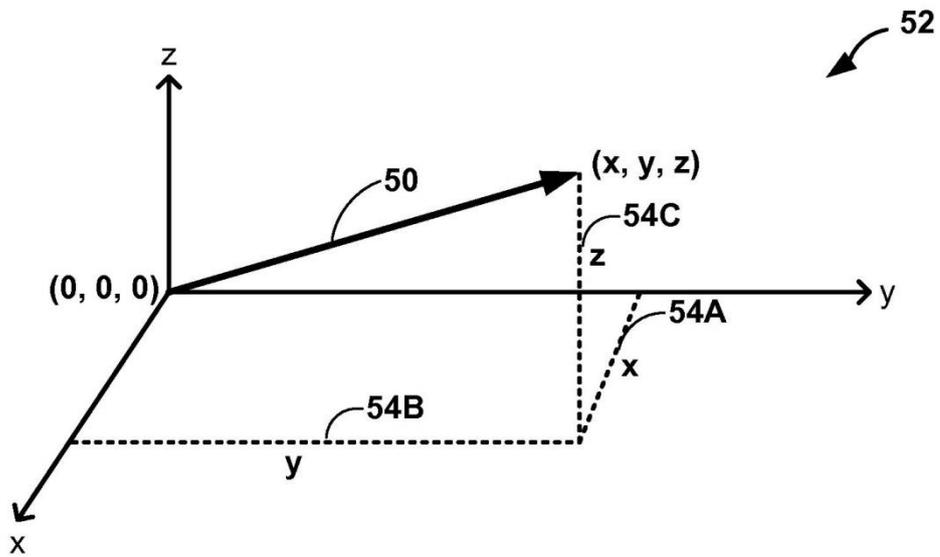


FIG. 3

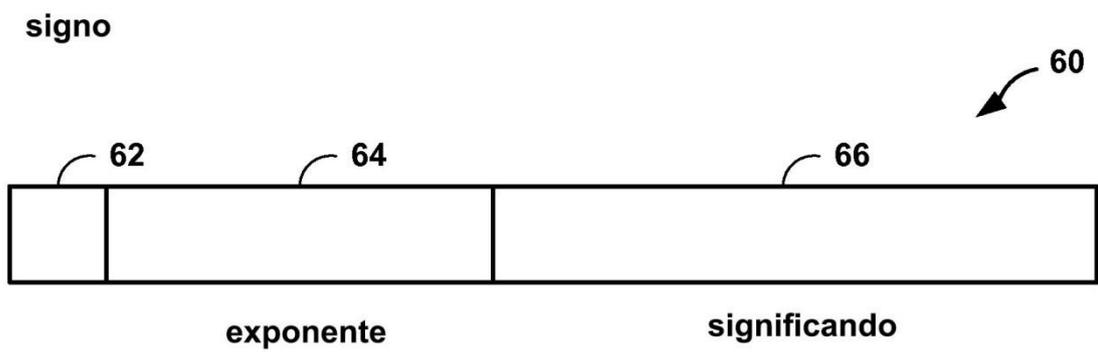
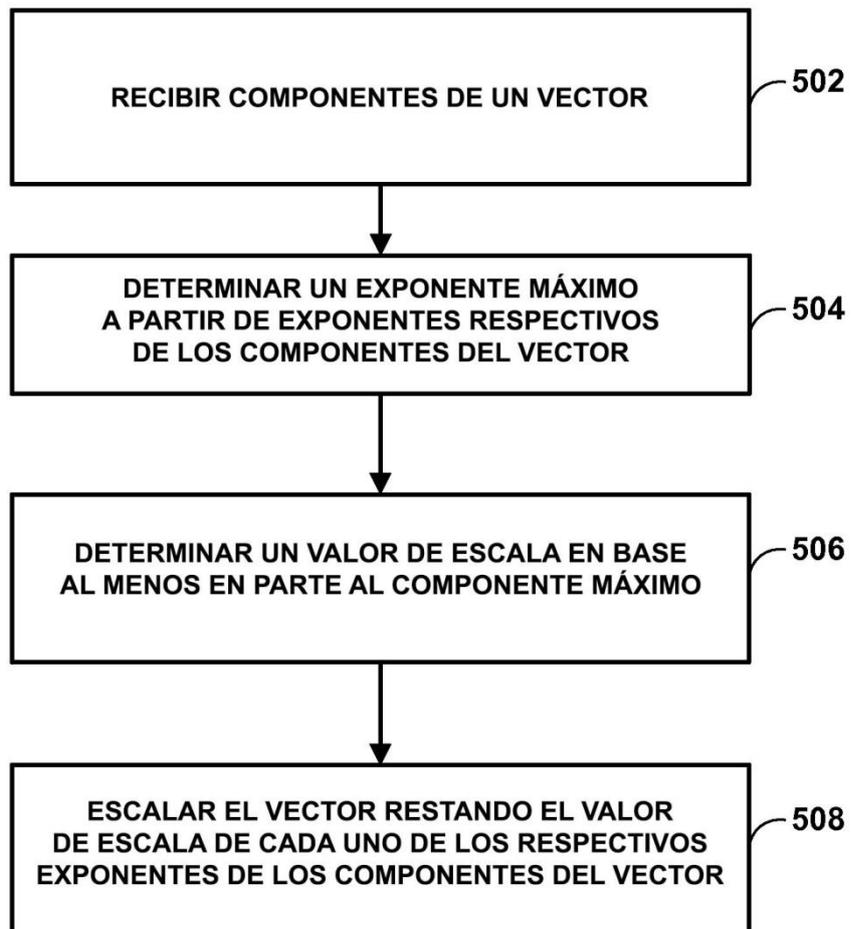


FIG. 4



**FIG. 5**