

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 959 308**

51 Int. Cl.:

G06F 3/01

(2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **23.03.2018** **E 18163724 (0)**

97 Fecha y número de publicación de la concesión europea: **19.07.2023** **EP 3396496**

54 Título: **Reducción del ancho de banda de pantalla con múltiples resoluciones**

30 Prioridad:

24.04.2017 US 201715494935

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

23.02.2024

73 Titular/es:

**INTEL CORPORATION (100.0%)
2200 Mission College Blvd.
Santa Clara, CA 95054, US**

72 Inventor/es:

**HICKS, RICHMOND;
RUNYAN, ARTHUR J.;
ANSARI, NAUSHEEN y
BISWAL, NARAYAN**

74 Agente/Representante:

LEHMANN NOVO, María Isabel

ES 2 959 308 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

Reducción del ancho de banda de pantalla con múltiples resoluciones

5 **CAMPO TÉCNICO**

Las realizaciones generalmente se refieren a pantallas y, más particularmente, a pantallas que tienen regiones de menor resolución para ahorrar ancho de banda y energía.

10 La invención está dirigida al aparato de la reivindicación 1, al método de la reivindicación 5 y al medio legible por máquina de la reivindicación 8. En las reivindicaciones dependientes se exponen realizaciones ventajosas.

ANTECEDENTES DE LA DESCRIPCIÓN

15 Las pantallas aumentan de tamaño y consumen más energía y ancho de banda. Las pantallas montadas en la cabeza (HMD) están mejorando, pero pueden requerir más ancho de banda del disponible en algunos sistemas.

El documento US 2009/316795 A1 divulga un método para visualizar vídeo que incluye recibir un flujo de bits de vídeo correspondiente a un vídeo. El vídeo comprende una secuencia de fotogramas que tienen un nivel de resolución de vídeo asociado. Se decodifica al menos una parte del flujo de bits de vídeo. Se muestra una región de fondo del vídeo en un primer nivel de resolución de vídeo y una región de ventana del vídeo se muestra en un segundo nivel de resolución de vídeo. El segundo nivel de resolución de vídeo es mayor que el primer nivel de resolución de vídeo.

25 El documento EP 2 919 098 A1 divulga métodos y sistemas para determinar una asignación de recursos de imagen para mostrar contenido dentro de un área de visualización. Un dispositivo de captura de imágenes o datos asociado con un dispositivo de visualización puede capturar una imagen de un espacio asociado con el usuario o capturar datos relacionados con otros objetos en el espacio. La distancia de visualización entre el usuario y el área de visualización (p. ej., el dispositivo de visualización) puede supervisarse y procesarse para determinar y/o ajustar la asignación de recursos de imagen para el contenido mostrado dentro del área de visualización. El movimiento del usuario, incluido el movimiento ocular, también puede supervisarse y procesarse para determinar y/o ajustar la asignación de recursos de imagen para el contenido mostrado dentro del área de visualización.

35 El documento US 2012/154277 A1 divulga un método y sistema que mejora la experiencia de un usuario cuando usa un dispositivo de visualización de cerca, para un dispositivo de visualización transparente o un dispositivo de visualización montado en la cabeza. Se crea una imagen optimizada para su visualización en relación con el campo de visión de un usuario en una escena. Se rastrea la posición y el movimiento de la cabeza y los ojos del usuario para determinar una región focal para el usuario. Una parte de la imagen optimizada se acopla a la región focal del usuario en la posición actual de los ojos, se predice una siguiente posición de la cabeza y los ojos, y una parte de la imagen optimizada se acopla a la región focal del usuario en la siguiente posición.

40 El documento WO 2017/015162 A1 divulga un sistema y un método de generación de imágenes virtuales. Se renderiza una pluralidad de fotogramas de imágenes sintéticas de la escena tridimensional y se muestran secuencialmente a un usuario final. Cada uno de los fotogramas de imagen mostrados tiene una distribución de resolución no uniforme.

45 El documento US 2014/247277 A1 divulga un método y un sistema para la representación de imágenes foveadas. El método incluye rastrear un punto de mirada de un usuario en un dispositivo de visualización y generar un número específico de capas de excentricidad en función del punto de mirada del usuario. El método también incluye suavizar las capas de excentricidad para eliminar artefactos, representar una imagen foveada basada en las capas de excentricidad y mostrar la imagen foveada al usuario a través del dispositivo de visualización.

50 **BREVE DESCRIPCIÓN DE LOS DIBUJOS**

Las diversas ventajas de las realizaciones se harán evidentes para los expertos en la materia al leer la siguiente memoria descriptiva y las reivindicaciones adjuntas, y al hacer referencia a los siguientes dibujos, en los que:

55 la Figura 1 es un diagrama de bloques que ilustra un sistema informático configurado para implementar uno o más aspectos de las realizaciones descritas en el presente documento;

las Figuras 2A-2D ilustran unos componentes de procesador paralelo, de acuerdo con una realización;

60 las Figuras 3A-3B son diagramas de bloques de multiprocesadores de gráficos, de acuerdo con las realizaciones;

65 las Figuras 4A-4F ilustran una arquitectura ilustrativa en la que una pluralidad de GPU están comunicativamente acopladas a una pluralidad de procesadores de múltiples núcleos;

la Figura 5 ilustra una canalización de procesamiento de gráficos, de acuerdo con una realización;

las Figuras 6A-6D son diagramas de bloques del sistema de un ejemplo de un sistema de procesamiento electrónico del tipo que puede ser útil para el rastreo de mirada;

la Figura 6E es una ilustración de un ejemplo de una solución de rastreo de mirada según una realización;

las Figuras 7A-7B son diagramas de bloques de ejemplo que muestran una solución de dos planos para mostrar ventanas de diferente resolución en la misma pantalla;

la Figura 8 es un diagrama de bloques de ejemplo que muestra una forma posible de componer dos planos de diferente resolución o calidad en una pantalla;

las Figuras 9A-9B son diagramas de bloques que muestran una forma ilustrativa de reducir las necesidades de ancho de banda de una HMD para hacerla inalámbrica;

las Figuras 10A-10B son diagramas de bloques de ejemplo que muestran dividir una imagen fija muy grande en porciones que pueden codificarse como fotogramas I de vídeo para retransmisión;

la Figura 10C es un diagrama de flujo que ilustra el método de las Figuras 10A-10B;

la Figura 11 es un diagrama de bloques de un ejemplo de una pantalla con capacidad de retroiluminación localizada según una realización;

la Figura 12A es un diagrama de bloques de un ejemplo de un dispositivo de procesamiento de datos según una realización;

la Figura 12B es una ilustración de un ejemplo de una determinación de distancia según una realización;

la Figura 13 es un diagrama de bloques de un ejemplo de una arquitectura de visualización en capas según una realización;

la Figura 14 es un diagrama de bloques de un ejemplo de una arquitectura de visualización que incluye múltiples unidades de visualización según una realización; y

la Figura 15 es un diagrama de bloques de un ejemplo de una arquitectura de suministro de medios asistidos por la nube según una realización;

las Figuras 16-18 son diagramas de bloques de un ejemplo de una visión general de un sistema de procesamiento de datos de acuerdo con una realización;

la Figura 19 es un diagrama de bloques de un ejemplo de un motor de procesamiento de gráficos de acuerdo con una realización;

las Figuras 20-22 son diagramas de bloques de ejemplos de unidades de ejecución de acuerdo con una realización;

la Figura 23 es un diagrama de bloques de un ejemplo de canalización de gráficos de acuerdo con una realización;

las Figuras 24A-24B son diagramas de bloques de ejemplos de programación de canalización de gráficos de acuerdo con una realización;

la Figura 25 es un diagrama de bloques de un ejemplo de una arquitectura de software de gráficos de acuerdo con una realización;

la Figura 26 es un diagrama de bloques de un ejemplo de un sistema de desarrollo central de propiedad intelectual (IP) de acuerdo con una realización; y

la Figura 27 es un diagrama de bloques de un ejemplo de un sistema en un circuito integrado de chip de acuerdo con una realización.

DESCRIPCIÓN DETALLADA

En la siguiente descripción, se exponen numerosos detalles específicos para proporcionar un entendimiento más minucioso de la presente invención. Sin embargo, será evidente para un experto en la materia que la presente

invención puede ponerse en práctica sin uno o más de estos detalles específicos. En otros casos, no se han descrito características bien conocidas para evitar oscurecer la presente invención.

Vista general del sistema

La Figura 1 es un diagrama de bloques que ilustra un sistema informático 100 configurado para implementar uno o más aspectos de las realizaciones descritas en el presente documento. El sistema informático 100 incluye un subsistema de procesamiento 101 que tiene uno o más procesadores 102 y una memoria de sistema 104 que se comunica mediante una ruta de interconexión que puede incluir un concentrador de memoria 105. El concentrador de memoria 105 puede ser un componente separado dentro de un componente de conjunto de chips o puede estar integrado dentro del uno o más procesador o procesadores 102. El concentrador de memoria 105 se acopla con un subsistema de E/S 111 mediante un enlace de comunicación 106. El subsistema de E/S 111 incluye un concentrador de E/S 107 que puede permitir que el sistema informático 100 reciba una entrada desde uno o más dispositivos de entrada 108. Adicionalmente, el concentrador de E/S 107 puede habilitar a un controlador de visualización, que puede incluirse en los uno o más procesadores 102, para que proporcione salidas a uno o más dispositivos de visualización 110A. En una realización, el uno o más dispositivos de visualización 110A acoplados con el concentrador de E/S 107 pueden incluir un dispositivo de visualización local, interno o embebido.

En una realización, el subsistema de procesamiento 101 incluye uno o más procesadores paralelos 112 acoplados al concentrador de memoria 105 mediante un bus u otro enlace de comunicación 113. El enlace de comunicación 113 puede ser uno de cualquier número de tecnologías o protocolos de enlace de comunicación basados en normas, tales como, pero sin limitación, PCI Express, o puede ser una interfaz de comunicaciones o tejido de comunicaciones específico del proveedor. En una realización, el uno o más procesadores paralelos 112 forman un sistema de procesamiento paralelo o vectorial computacionalmente enfocado que incluye un gran número de núcleos de procesamiento y/o agrupaciones de procesamiento, tal como un procesador de muchos núcleos integrados (MIC). En una realización, el uno o más procesadores paralelos 112 forman un subsistema de procesamiento de gráficos que puede emitir píxeles a uno del uno o más dispositivos de visualización 110A acoplados mediante el concentrador de E/S 107. El uno o más procesadores paralelos 112 pueden incluir también un controlador de visualización e interfaz de visualización (no mostrados) para permitir una conexión directa a uno o más dispositivos de visualización 110B.

Dentro del subsistema de E/S 111, una unidad de almacenamiento de sistema 114 puede conectarse al concentrador de E/S 107 para proporcionar un mecanismo de almacenamiento para el sistema informático 100. Puede usarse un conmutador de E/S 116 para proporcionar un mecanismo de interfaz para permitir conexiones entre el concentrador de E/S 107 y otros componentes, tales como un adaptador de red 118 y/o un adaptador de red inalámbrico 119 que pueden integrarse en la plataforma, y diversos otros dispositivos que pueden añadirse mediante uno o más dispositivos de adición 120. El adaptador de red 118 puede ser un adaptador de Ethernet u otro adaptador de red alámbrica. El adaptador de red inalámbrica 119 puede incluir uno o más de un dispositivo de Wi-Fi, Bluetooth, de comunicación de campo cercano (NFC) u otro dispositivo de red que incluye una o más radios inalámbricas.

El sistema informático 100 puede incluir otros componentes no explícitamente mostrados, incluyendo USB u otras conexiones de puerto, unidades de almacenamiento óptico, dispositivos de captura de vídeo, y similares, que también puede conectarse al concentrador de E/S 107. Las rutas de comunicación que interconectan los diversos componentes en la Figura 1 pueden implementarse usando cualquier protocolo adecuado, tal como protocolos (p. ej., PCI-Express) basados en PCI (Interconexión de Componentes Periféricos), o cualesquiera otras interfaces de comunicación de bus o de punto a punto y/o protocolo o protocolos, tal como la interconexión de alta velocidad NVLink, o protocolos de interconexión conocidos en la técnica.

En una realización, el uno o más procesadores paralelos 112 incorporan circuitería optimizada para procesamiento de gráficos y vídeo, que incluye, por ejemplo, circuitería de salida de vídeo y constituye una unidad de procesamiento de gráficos (GPU). En otra realización, los uno o más procesadores paralelos 112 incorporan circuitería optimizada para procesamiento de propósito general, mientras conservan la arquitectura computacional subyacente, descrita en mayor detalle en el presente documento. En otra realización más, los componentes del sistema informático 100 pueden estar integrados con uno o más otros elementos de sistema en un único circuito integrado. Por ejemplo, los uno o más procesadores paralelos 112, el concentrador de memoria 105, el procesador o procesadores 102 y el concentrador de E/S 107 pueden integrarse en un circuito integrado de sistema en chip (SoC). Como alternativa, los componentes del sistema informático 100 pueden estar integrados en un único paquete para formar una configuración de sistema en paquete (SIP). En una realización, al menos una porción de los componentes del sistema informático 100 puede estar integrada en un módulo de múltiples chips (MCM), que puede estar interconectado con otros módulos de múltiples chips en un sistema informático modular.

Se apreciará que el sistema informático 100 mostrado en el presente documento es ilustrativo y que son posibles variaciones y modificaciones. La topología de conexión, que incluye el número y disposición de puentes, el número de procesador o procesadores 102, y el número de procesador o procesadores paralelos 112, puede modificarse como se desee. Por ejemplo, en algunas realizaciones, la memoria de sistema 104 está conectada al procesador o procesadores 102 directamente en lugar de a través de un puente, mientras que otros dispositivos se comunican con la memoria de sistema 104 mediante el concentrador de memoria 105 y el procesador o procesadores 102. En otras

topologías alternativas, el procesador o procesadores paralelos 112 están conectados al concentrador de E/S 107 o directamente a uno del uno o más procesador o procesadores 102, en lugar de al concentrador de memoria 105. En otras realizaciones, el concentrador de E/S 107 y el concentrador de memoria 105 pueden estar integrados en un único chip. Algunas realizaciones pueden incluir dos o más conjuntos del procesador o procesadores 102 adjuntos mediante múltiples zócalos, que pueden acoplarse con dos o más instancias del procesador o procesadores paralelos 112.

Alguno de los componentes particulares mostrados en el presente documento es opcional y puede no estar incluido en todas las implementaciones del sistema informático 100. Por ejemplo, puede soportarse cualquier número de tarjetas o periféricos de adición, o pueden eliminarse algunos componentes. Adicionalmente, algunas arquitecturas pueden usar diferente terminología para componentes similares a aquellos ilustrados en la Figura 1. Por ejemplo, el concentrador de memoria 105 puede denominarse un puente norte en algunas arquitecturas, mientras que el concentrador de E/S 107 puede denominarse un puente sur.

La Figura 2A ilustra un procesador paralelo 200, de acuerdo con una realización. Los diversos componentes del procesador paralelo 200 pueden implementarse usando uno o más dispositivos de circuito integrado, tal como procesadores programables, circuitos integrados específicos de la aplicación (ASIC) o campos de matrices de puertas programables (FPGA). El procesador paralelo 200 ilustrado es una variante del uno o más procesador o procesadores paralelos 112 mostrados en la Figura 1, de acuerdo con una realización.

En una realización, el procesador paralelo 200 incluye una unidad de procesamiento paralelo 202. La unidad de procesamiento paralelo incluye una unidad de E/S 204 que posibilita la comunicación con otros dispositivos, que incluyen otras instancias de la unidad de procesamiento paralelo 202. La unidad de E/S 204 puede estar conectada directamente a otros dispositivos. En una realización, la unidad de E/S 204 se conecta con otros dispositivos mediante el uso de una interfaz de concentrador o de conmutador, tal como un concentrador de memoria 105. Las conexiones entre el concentrador de memoria 105 y la unidad de E/S 204 forman un enlace de comunicación 113. Dentro de la unidad de procesamiento paralelo 202, la unidad de E/S 204 se conecta con una interfaz de anfitrión 206 y una barra transversal de memoria 216, donde la interfaz de anfitrión 206 recibe comandos dirigidos a realizar las operaciones de procesamiento y la barra transversal de memoria 216 recibe comandos dirigidos a realizar operaciones de memoria.

Cuando la interfaz de anfitrión 206 recibe una memoria intermedia de comando mediante la unidad de E/S 204, la interfaz de anfitrión 206 puede dirigir operaciones de trabajo para realizar aquellos comandos a un extremo delantero 208. En una realización, el extremo delantero 208 se acopla con un planificador 210, que está configurado para distribuir comandos u otros elementos de trabajo a una matriz de agrupación de procesamiento 212. En una realización, el planificador 210 garantiza que la matriz de agrupación de procesamiento 212 está configurada apropiadamente y en un estado válido antes de que se distribuyan las tareas a las agrupaciones de procesamiento de la matriz de agrupación de procesamiento 212. En una realización, el planificador 210 se implementa mediante lógica de firmware que se ejecuta en un microcontrolador. El planificador implementado por microcontrolador 210 puede configurarse para realizar operaciones de planificación compleja y distribución de trabajo en granularidad basta y precisa, lo que permite la anticipación rápida y la conmutación de contexto de hilos que se ejecutan en la matriz de procesamiento 212. En una realización, el software de anfitrión puede probar cargas de trabajo para su planificación en la matriz de procesamiento 212 mediante uno de múltiples llamadores de procesamiento de gráficos. Las cargas de trabajo, a continuación, pueden distribuirse automáticamente a través de la matriz de procesamiento 212 por la lógica del planificador 210 dentro del microcontrolador planificador.

La matriz de agrupación de procesamiento 212 puede incluir hasta "N" agrupaciones de procesamiento (p. ej., de la agrupación 214A, la agrupación 214B a la agrupación 214N). Cada agrupación 214A-214N de la matriz de agrupación de procesamiento 212 puede ejecutar un gran número de hilos concurrentes. El planificador 210 puede asignar trabajo a las agrupaciones 214A-214N de la matriz de agrupación de procesamiento 212 usando diversos algoritmos de planificación y/o distribución de trabajo, que pueden variar dependiendo de la carga de trabajo que surge para cada tipo de programa o cálculo. La planificación puede manejarse dinámicamente por el planificador 210, o puede ser ayudada, en parte, por lógica de compilador durante la compilación de la lógica de programa configurada para la ejecución por la matriz de agrupación de procesamiento 212. En una realización, pueden asignarse diferentes agrupaciones 214A-214N de la matriz de agrupación de procesamiento 212 para procesar diferentes tipos de programas o para realizar diferentes tipos de cálculos.

La matriz de agrupación de procesamiento 212 puede configurarse para realizar diversos tipos de operaciones de procesamiento paralelo. En una realización, la matriz de agrupación de procesamiento 212 está configurada para realizar operaciones de cálculo paralelo de fin general. Por ejemplo, la matriz de agrupación de procesamiento 212 puede incluir lógica para ejecutar tareas de procesamiento que incluye filtración de datos de vídeo y/o de audio, realización de operaciones de modelado, que incluye operaciones físicas y realización de transformaciones de datos.

En una realización, la matriz de agrupación de procesamiento 212 está configurada para realizar operaciones de procesamiento de gráficos en paralelo. En realizaciones en las que el procesador paralelo 200 está configurado para realizar operaciones de procesamiento de gráficos, la matriz de agrupación de procesamiento 212 puede incluir una lógica adicional para soportar la ejecución de tales operaciones de procesamiento de gráficos, incluyendo, pero sin

limitación, una lógica de muestreo de textura para realizar operaciones de textura, así como una lógica de teselación y otra lógica de procesamiento de vértices. Adicionalmente, la matriz de agrupación de procesamiento 212 puede configurarse para ejecutar programas sombreadores relacionados con el procesamiento de gráficos tales como, pero sin limitación, sombreadores de vértices, sombreadores de teselación, sombreadores de geometría y sombreadores de píxeles. La unidad de procesamiento paralelo 202 puede transferir datos desde la memoria de sistema mediante la unidad de E/S 204 para su procesamiento. Durante el procesamiento, los datos transferidos pueden almacenarse en memoria en chip (p. ej., memoria de procesador paralelo 222) durante el procesamiento y, a continuación, escribirse de vuelta en memoria de sistema.

En una realización, cuando se usa la unidad de procesamiento paralelo 202 para realizar el procesamiento de gráficos, el planificador 210 puede estar configurado para dividir la carga de trabajo de procesamiento en tareas de tamaño aproximadamente igual, para permitir mejor la distribución de las operaciones de procesamiento de gráficos a múltiples agrupaciones 214A-214N de la matriz de agrupación de procesamiento 212. En algunas realizaciones, las porciones de la matriz de agrupación de procesamiento 212 pueden estar configuradas para realizar diferentes tipos de procesamiento. Por ejemplo, una primera porción puede estar configurada para realizar sombreado de vértices y generación de topología, una segunda porción puede estar configurada para realizar teselación y sombreado de geometría, y una tercera porción puede estar configurada para realizar sombreado de píxeles u otras operaciones de espacio de pantalla, para producir una imagen representada para su visualización. Los datos intermedios producidos por una o más de las agrupaciones 214A-214N pueden almacenarse en memorias intermedias para permitir que se transmitan los datos intermedios entre las agrupaciones 214A-214N para su procesamiento adicional.

Durante la operación, la matriz de agrupación de procesamiento 212 puede recibir tareas de procesamiento para que se ejecuten mediante el planificador 210, que recibe comandos que definen tareas de procesamiento desde el extremo delantero 208. Para operaciones de procesamiento de gráficos, las tareas de procesamiento pueden incluir índices de datos que van a procesarse, p. ej., datos de superficie (parche), datos de primitivas, datos de vértices y/o datos de píxeles, así como parámetros de estado y comandos que definen cómo han de procesarse los datos (p. ej., qué programa ha de ejecutarse). El planificador 210 puede estar configurado para extraer los índices que corresponden a las tareas o puede recibir los índices desde el extremo delantero 208. El extremo delantero 208 puede estar configurado para garantizar que la matriz de agrupación de procesamiento 212 está configurada en un estado válido antes de que se inicie la carga de trabajo especificada por las memorias intermedias de comando de entrada (p. ej., memorias intermedias de lote, memorias intermedias de inserción, etc.).

Cada una de la una o más instancias de la unidad de procesamiento paralelo 202 puede acoplarse con la memoria de procesador paralelo 222. Puede accederse a la memoria de procesador paralelo 222 mediante la barra transversal de memoria 216, que puede recibir solicitudes de memoria desde la matriz de agrupación de procesamiento 212, así como de la unidad de E/S 204. La barra transversal de memoria 216 puede acceder a la memoria de procesador paralelo 222 mediante una interfaz de memoria 218. La interfaz de memoria 218 puede incluir múltiples unidades de subdivisión (p. ej., la unidad de subdivisión 220A, la unidad de subdivisión 220B a la unidad de subdivisión 220N), cada una acoplable a una porción (p. ej., la unidad de memoria) de la memoria de procesador paralelo 222. En una implementación, el número de unidades de subdivisión 220A-220N está configurado para que sea igual al número de unidades de memoria, de manera que una primera unidad de subdivisión 220A tiene una correspondiente primera unidad de memoria 224A, una segunda unidad de subdivisión 220B tiene una correspondiente unidad de memoria 224B y una unidad de subdivisión de orden N 220N tiene una correspondiente unidad de memoria de orden N 224N. En otras realizaciones, el número de unidades de subdivisión 220A-220N puede no ser igual al número de dispositivos de memoria.

En diversas realizaciones, las unidades de memoria 224A-224N pueden incluir diversos tipos de dispositivos de memoria, que incluyen memoria de acceso aleatorio dinámica (DRAM) o memoria de acceso aleatorio de gráficos, tal como la memoria de acceso aleatorio de gráficos síncrona (SGRAM), que incluye la memoria de tasa de datos doble de gráficos (GDDR). En una realización, las unidades de memoria 224A-224N pueden incluir también memoria 3D apilada, que incluye, pero sin limitación, memoria de ancho de banda alto (HBM). Los expertos en la materia apreciarán que la implementación específica de las unidades de memoria 224A-224N puede variar, y puede seleccionarse de uno de diversos diseños convencionales. Los objetivos de representación, tales como las memorias intermedias de tramas o los mapas de textura pueden almacenarse a través de las unidades de memoria 224A-224N, permitiendo que las unidades de subdivisión 220A-220N escriban porciones de cada objetivo de representación en paralelo para usar de manera efectiva el ancho de banda disponible de la memoria de procesador paralelo 222. En algunas realizaciones, puede excluirse una instancia local de la memoria de procesador paralelo 222 en favor de un diseño de memoria unificado que utiliza memoria de sistema en conjunto con memoria caché local.

En una realización, una cualquiera de las agrupaciones 214A-214N de la matriz de agrupación de procesamiento 212 puede procesar datos que se escribirán en cualquiera de las unidades de memoria 224A-224N dentro de la memoria de procesador paralelo 222. La barra transversal de memoria 216 puede estar configurada para transferir la salida de cada agrupación 214A-214N en cualquier unidad de subdivisión 220A-220N o en otra agrupación 214A-214N, que puede realizar operaciones de procesamiento adicionales en la salida. Cada agrupación 214A-214N puede comunicarse con la interfaz de memoria 218 a través de la barra transversal de memoria 216 para leer desde o escribir en diversos dispositivos de memoria externos. En una realización, la barra transversal de memoria 216 tiene una

conexión a la interfaz de memoria 218 para comunicarse con la unidad de E/S 204, así como una conexión a una instancia local de la memoria de procesador paralelo 222, lo que posibilita que las unidades de procesamiento dentro de las diferentes agrupaciones de procesamiento 214A-214N se comuniquen con la memoria de sistema u otra memoria que no sea local a la unidad de procesamiento paralelo 202. En una realización, la barra transversal de memoria 216 puede usar canales virtuales para separar flujos de tráfico entre las agrupaciones 214A-214N y las unidades de subdivisión 220A-220N.

Aunque se ilustra una única instancia de la unidad de procesamiento paralelo 202 dentro del procesador paralelo 200, puede incluirse cualquier número de instancias de la unidad de procesamiento paralelo 202. Por ejemplo, pueden proporcionarse múltiples instancias de la unidad de procesamiento paralelo 202 en una única tarjeta de adición, o pueden interconectarse múltiples tarjetas de adición. Las diferentes instancias de la unidad de procesamiento paralelo 202 pueden estar configuradas para interoperar incluso si las diferentes instancias tienen diferentes números de núcleos de procesamiento, diferentes cantidades de memoria de procesador paralelo local y/u otras diferencias de configuración. Por ejemplo, y en una realización, algunas instancias de la unidad de procesamiento paralelo 202 pueden incluir unidades de coma flotante de precisión más alta con relación a otras instancias. Los sistemas que incorporan una o más instancias de la unidad de procesamiento paralelo 202 o el procesador paralelo 200 pueden implementarse en una diversidad de configuraciones y factores de forma, incluyendo, pero sin limitación, ordenadores personales de sobremesa, portátiles o de mano, servidores, estaciones de trabajo, consolas de juegos y/o sistemas integrados.

La Figura 2B es un diagrama de bloques de una unidad de subdivisión 220, de acuerdo con una realización. En una realización, la unidad de subdivisión 220 es una instancia de una de las unidades de subdivisión 220A-220N de la Figura 2A. Como se ilustra, la unidad de subdivisión 220 incluye una caché L2 221, una interfaz de memoria intermedia de trama 225 y una ROP 226 (unidad de operaciones de rasterización). La caché L2 221 es una caché de lectura/escritura que está configurada para realizar operaciones de carga y almacenamiento recibidas desde la barra transversal de memoria 216 y la ROP 226. Los fallos de lectura y las solicitudes de escritura urgentes se emiten por la caché L2 221 a la interfaz de memoria intermedia de trama 225 para su procesamiento. Pueden enviarse también las actualizaciones a la memoria intermedia de trama mediante la interfaz de memoria intermedia de trama 225 para su procesamiento. En una realización, la interfaz de memoria intermedia de trama 225 interconecta con una de las unidades de memoria en la memoria de procesador paralelo, tal como las unidades de memoria 224A-224N de la Figura 2 (por ejemplo, dentro de la memoria de procesador paralelo 222).

En las aplicaciones de gráficos, la ROP 226 es una unidad de procesamiento que realiza operaciones de rasterización tales como estarcido, prueba z, mezcla y similares. La ROP 226 a continuación emite los datos de gráficos procesados que se almacenan en la memoria de gráficos. En algunas realizaciones, la ROP 226 incluye lógica de compresión para comprimir datos de profundidad o de color que se escriben en memoria y descomprimir datos de profundidad o de color que se leen desde la memoria. La lógica de compresión puede ser lógica de compresión sin pérdidas, que hace uso de uno o más de múltiples algoritmos de compresión. El tipo de compresión que se realiza por la ROP 226 puede variar basándose en las características estadísticas de los datos que van a comprimirse. Por ejemplo, en una realización, se realiza compresión de color delta en datos de profundidad y de color en una base por pieza.

En algunas realizaciones, la ROP 226 está incluida dentro de cada agrupación de procesamiento (p. ej., la agrupación 214A-214N de la Figura 2) en lugar de dentro de la unidad de subdivisión 220. En tal realización, las solicitudes de lectura y escritura para datos de píxeles se transmiten a través de la barra transversal de memoria 216 en lugar de los datos de fragmento de píxel. Los datos de gráficos procesados pueden visualizarse en un dispositivo de visualización, tal como uno del uno o más dispositivos de visualización 110 de la Figura 1, encaminarse para su procesamiento adicional por el procesador o procesadores 102, o encaminarse para su procesamiento adicional por una de las entidades de procesamiento dentro del procesador paralelo 200 de la Figura 2A.

La Figura 2C es un diagrama de bloques de una agrupación de procesamiento 214 dentro de una unidad de procesamiento paralelo, de acuerdo con una realización. En una realización, la agrupación de procesamiento es una instancia de una de las agrupaciones de procesamiento 214A-214N de la Figura 2. La agrupación de procesamiento 214 puede estar configurada para ejecutar muchos hilos en paralelo, donde el término "hilo" se refiere a una instancia de un programa particular que se ejecuta en un conjunto particular de datos de entrada. En algunas realizaciones, se usan técnicas de emisión de instrucciones de única instrucción de múltiples datos (SIMD) para soportar la ejecución paralela de un gran número de hilos sin proporcionar múltiples unidades de instrucción independientes. En otras realizaciones, se usan técnicas de única instrucción de múltiples hilos (SIMT) para soportar la ejecución paralela de un gran número de hilos generalmente sincronizados, usando una unidad de instrucción común configurada para emitir instrucciones en un conjunto de motores de procesamiento dentro de cada una de las agrupaciones de procesamiento. A diferencia del régimen de ejecución de SIMD, donde todos los motores de procesamiento ejecutan habitualmente instrucciones idénticas, la ejecución de SIMT permite que diferentes hilos sigan más fácilmente rutas de ejecución divergentes a través de un programa de hilos dado. Los expertos en la materia entenderán que un régimen de procesamiento de SIMD representa un subconjunto funcional de un régimen de procesamiento de SIMT.

La operación de la agrupación de procesamiento 214 puede controlarse mediante un gestor de canalizaciones 232 que distribuye las tareas de procesamiento a procesadores paralelos de SIMT. El gestor de canalizaciones 232 recibe

instrucciones desde el planificador 210 de la Figura 2 y gestiona la ejecución de esas instrucciones mediante un multiprocesador de gráficos 234 y/o una unidad de texturas 236. El multiprocesador de gráficos 234 ilustrado es una instancia ilustrativa de un procesador paralelo de SIMT. Sin embargo, pueden incluirse diversos tipos de procesadores paralelos de SIMT de arquitecturas diferentes dentro de la agrupación de procesamiento 214. Puede incluirse una o más instancias del multiprocesador de gráficos 234 dentro de una agrupación de procesamiento 214. El multiprocesador de gráficos 234 puede procesar datos y puede usarse una barra transversal de datos 240 para distribuir los datos procesados a uno de múltiples posibles destinos, que incluyen otras unidades sombreadoras. El gestor de canalizaciones 232 puede facilitar la distribución de datos procesados especificando destinos para que se distribuyan datos procesados mediante la barra transversal de datos 240.

Cada multiprocesador de gráficos 234 dentro de la agrupación de procesamiento 214 puede incluir un conjunto idéntico de lógica de ejecución funcional (p. ej., unidades aritmético-lógicas, unidades de carga-almacenamiento, etc.). La lógica de ejecución funcional puede configurarse de una manera canalizada en la que pueden emitirse instrucciones nuevas antes de que se hayan completado instrucciones previas. La lógica de ejecución funcional soporta una diversidad de operaciones que incluyen aritmética de números enteros y de coma flotante, operaciones de comparación, operaciones booleanas, desplazamiento de bits y del cálculo de diversas funciones algebraicas. En una realización, puede aprovecharse el mismo hardware funcional-unitario para realizar diferentes operaciones y puede estar presente cualquier combinación de unidades funcionales.

Las instrucciones transmitidas a la agrupación de procesamiento 214 constituyen un hilo. Un conjunto de hilos que se ejecutan a través del conjunto de motores de procesamiento paralelo es un grupo de hilos. Un grupo de hilos ejecuta el mismo programa en diferentes datos de entrada. Cada hilo dentro de un grupo de hilos puede asignarse a un motor de procesamiento diferente dentro de un multiprocesador de gráficos 234. Un grupo de hilos puede incluir menos hilos que el número de motores de procesamiento dentro del multiprocesador de gráficos 234. Cuando un grupo de hilos incluye menos hilos que el número de motores de procesamiento, uno o más de los motores de procesamiento pueden encontrarse en espera durante ciclos en los que se está procesando ese grupo de hilos. Un grupo de hilos puede incluir también más hilos que el número de motores de procesamiento dentro del multiprocesador de gráficos 234. Cuando el grupo de hilos incluye más hilos que el número de motores de procesamiento dentro del multiprocesador de gráficos 234, puede realizarse el procesamiento a través de ciclos de reloj consecutivos. En una realización, pueden ejecutarse múltiples grupos de hilos concurrentemente en un multiprocesador de gráficos 234.

En una realización, el multiprocesador de gráficos 234 incluye una memoria caché interna para realizar operaciones de carga y almacenamiento. En una realización, el multiprocesador de gráficos 234 puede prescindir de una caché interna y usar una memoria caché (p. ej., la caché L1 308) dentro de la agrupación de procesamiento 214. Cada multiprocesador de gráficos 234 también tiene acceso a cachés de nivel L2 dentro de las unidades de subdivisión (p. ej., las unidades de subdivisión 220A-220N de la Figura 2) que se comparten entre todas las agrupaciones de procesamiento 214 y pueden usarse para transferir datos entre hilos. El multiprocesador de gráficos 234 puede acceder también a memoria global fuera de chip, que puede incluir uno o más de memoria de procesador paralelo local y/o memoria de sistema. Puede usarse cualquier memoria externa a la unidad de procesamiento paralelo 202 como memoria global. Las realizaciones en las que la agrupación de procesamiento 214 incluye múltiples instancias del multiprocesador de gráficos 234 pueden compartir instrucciones y datos comunes, que pueden almacenarse en la caché L1 308.

Cada agrupación de procesamiento 214 puede incluir una MMU 245 (unidad de gestión de memoria) que está configurada para mapear direcciones virtuales en direcciones físicas. En otras realizaciones, una o más instancias de la MMU 245 pueden residir dentro de la interfaz de memoria 218 de la Figura 2. La MMU 245 incluye un conjunto de entradas de tabla de página (PTE) usadas para mapear una dirección virtual a una dirección física de una pieza (más información sobre la aplicación de piezas) y, opcionalmente, un índice de línea de caché. La MMU 245 puede incluir memorias intermedias de traducción adelantada (TLB) de dirección o cachés que pueden residir dentro del multiprocesador de gráficos 234 o la caché L1 o la agrupación de procesamiento 214. La dirección física se procesa para distribuir la localidad de acceso de datos de superficie para permitir una intercalación de solicitud eficaz entre unidades de subdivisión. El índice de línea de caché puede usarse para determinar si una solicitud para una línea de caché es un acierto o un fallo.

En aplicaciones de gráficos e informáticas, puede configurarse una agrupación de procesamiento 214 de manera que cada multiprocesador de gráficos 234 esté acoplado a una unidad de textura 236 para realizar operaciones de mapeo de textura, p. ej., determinar posiciones de muestra de textura, leer datos de textura y filtrar los datos de textura. Los datos de textura se leen desde una caché L1 de textura interna (no mostrada) o, en algunas realizaciones, desde la caché L1 dentro del multiprocesador de gráficos 234 y se extraen desde una caché L2, memoria de procesador paralelo local o memoria de sistema, según sea necesario. Cada multiprocesador de gráficos 234 emite tareas procesadas a la barra transversal de datos 240 para proporcionar la tarea procesada a otra agrupación de procesamiento 214 para su procesamiento adicional o para almacenar la tarea procesada en una caché L2, memoria de procesador paralelo local o memoria de sistema mediante la barra transversal de memoria 216. Una preROP 242 (unidad de operaciones previas a la rasterización) está configurada para recibir datos desde el multiprocesador de gráficos 234, dirigir datos a las unidades de ROP, que pueden estar ubicadas con unidades de subdivisión como se describe en el presente documento (p. ej., las unidades de subdivisión 220A-220N de la Figura 2). La unidad preROP

242 puede realizar optimizaciones para la mezcla de color, organizar datos de color de píxel y realizar traducciones de dirección.

Se apreciará que la arquitectura de núcleo descrita en el presente documento es ilustrativa y que son posibles variaciones y modificaciones. Puede incluirse cualquier número de unidades de procesamiento, p. ej., el multiprocesador de gráficos 234, las unidades de textura 236, las preROP 242, etc., dentro de una agrupación de procesamiento 214. Además, aunque únicamente se muestra una agrupación de procesamiento 214, la unidad de procesamiento paralelo, como se describe en el presente documento, puede incluir cualquier número de instancias de la agrupación de procesamiento 214. En una realización, cada agrupación de procesamiento 214 puede estar configurada para operar independientemente de otras agrupaciones de procesamiento 214 usando unidades de procesamiento separadas y distintas, cachés L1, etc.

La Figura 2D muestra un multiprocesador de gráficos 234, de acuerdo con una realización. En tal realización, el multiprocesador de gráficos 234 se acopla con el gestor de canalizaciones 232 de la agrupación de procesamiento 214. El multiprocesador de gráficos 234 tiene una canalización de ejecución que incluye, pero sin limitación, una caché de instrucciones 252, una unidad de instrucciones 254, una unidad de mapeo de direcciones 256, un archivo de registro 258, uno o más núcleos de unidad de procesamiento de gráficos de fin general (GPGPU) 262 y una o más unidades de carga/almacenamiento 266. Los núcleos de GPGPU 262 y las unidades de carga/almacenamiento 266 están acoplados con la memoria caché 272 y la memoria compartida 270 mediante una interconexión de memoria y caché 268.

En una realización, la caché de instrucciones 252 recibe un flujo de instrucciones para ejecutarse desde el gestor de canalizaciones 232. Las instrucciones se almacenan en caché en la caché de instrucciones 252 y se despachan para su ejecución por la unidad de instrucciones 254. La unidad de instrucciones 254 puede despachar instrucciones como grupos de hilos (p. ej., envoltorios), con cada hilo del grupo de hilos asignado a una unidad de ejecución diferente dentro del núcleo de GPGPU 262. Una instrucción puede acceder a cualquiera del espacio de direcciones local, compartido o global, especificando una dirección dentro de un espacio de direcciones unificado. La unidad de mapeo de direcciones 256 puede usarse para traducir direcciones en el espacio de direcciones unificado en una dirección de memoria distinta a la que puede accederse por las unidades de carga/almacenamiento 266.

El archivo de registro 258 proporciona un conjunto de registros para las unidades funcionales del multiprocesador de gráficos 324. El archivo de registro 258 proporciona almacenamiento temporal para operandos conectados a las rutas de datos de las unidades funcionales (p. ej., los núcleos de GPGPU 262, las unidades de carga/almacenamiento 266) del multiprocesador de gráficos 324. En una realización, el archivo de registro 258 se divide entre cada una de las unidades funcionales de manera que cada unidad funcional está asignada a una porción especializada del archivo de registro 258. En una realización, el archivo de registro 258 se divide entre las diferentes envoltorios que se ejecutan por el multiprocesador de gráficos 324.

Cada uno de los núcleos de GPGPU 262 puede incluir unidades de coma flotante (FPU) y/o unidades aritméticas lógicas (ALU) de números enteros que se usan para ejecutar instrucciones del multiprocesador de gráficos 324. Los núcleos de GPGPU 262 pueden ser similares en arquitectura o pueden diferir en arquitectura, de acuerdo con las realizaciones. Por ejemplo, y en una realización, una primera porción de los núcleos de GPGPU 262 incluye una FPU de precisión sencilla y una ALU de números enteros, mientras que una segunda porción de los núcleos de GPGPU incluye una FPU de precisión doble. En una realización, las FPU pueden implementar la norma IEEE 754-2008 para aritmética de coma flotante o posibilitar aritmética de coma flotante de precisión variable. El multiprocesador de gráficos 324 puede incluir adicionalmente una o más unidades de función fija o de función especial para realizar funciones específicas, tales como operaciones de copia de rectángulo o de mezcla de píxeles. En una realización, uno o más de los núcleos de GPGPU puede incluir también lógica de función fija o especial.

En una realización, los núcleos de GPGPU 262 incluyen lógica de SIMD que puede realizar una instrucción sencilla en múltiples conjuntos de datos. En una realización, los núcleos de GPGPU 262 pueden ejecutar físicamente instrucciones SIMD4, SIMD8 y SIMD16 y ejecutar lógicamente instrucciones SIMD1, SIMD2 y SIMD32. Las instrucciones SIMD para los núcleos de GPGPU pueden generarse en el momento de compilación por un compilador de sombreador o generarse automáticamente cuando se ejecutan programas escritos y compilados para arquitecturas de datos de múltiples programas sencillos (SPMD) o SIMT. Pueden ejecutarse múltiples hilos de un programa configurado para el modelo de ejecución de SIMT mediante una instrucción de SIMD sencilla. Por ejemplo, y en una realización, ocho hilos SIMT pueden realizar las mismas operaciones o similares que pueden ejecutarse en paralelo mediante una unidad lógica SIMD8 sencilla.

La interconexión de memoria y caché 268 es una red de interconexión que conecta cada una de las unidades funcionales del multiprocesador de gráficos 324 al archivo de registro 258 y a la memoria compartida 270. En una realización, la interconexión de memoria y caché 268 es una interconexión de barra transversal que permite que la unidad de carga/almacenamiento 266 implemente operaciones de carga y almacenamiento entre la memoria compartida 270 y el archivo de registro 258. El archivo de registro 258 puede operar a la misma frecuencia que los núcleos de GPGPU 262, por lo tanto, la transferencia de datos entre los núcleos de GPGPU 262 y el archivo de registro 258 es de muy baja latencia. La memoria compartida 270 puede usarse para permitir la comunicación entre hilos que

se ejecutan en las unidades funcionales dentro del multiprocesador de gráficos 234. La memoria caché 272 puede usarse como una caché de datos, por ejemplo, para almacenar en caché datos de textura comunicados entre las unidades funcionales y la unidad de textura 236. La memoria compartida 270 puede usarse también como un programa gestionado almacenado en caché. Los hilos que se ejecutan en los núcleos de GPGPU 262 pueden almacenar datos mediante programación dentro de la memoria compartida además de los datos almacenados automáticamente en caché que se almacenan dentro de la memoria caché 272.

Las Figuras 3A-3B ilustran multiprocesadores de gráficos adicionales, de acuerdo con las realizaciones. Los multiprocesadores de gráficos 325, 350 ilustrados son variantes del multiprocesador de gráficos 234 de la Figura 2C. Los multiprocesadores de gráficos 325, 350 ilustrados pueden estar configurados como un multiprocesador de envío por flujo continuo (SM) que puede realizar la ejecución simultánea de un gran número de hilos de ejecución.

La Figura 3A muestra un multiprocesador de gráficos 325 de acuerdo con una realización adicional. El multiprocesador de gráficos 325 incluye múltiples instancias adicionales de unidades de recurso de ejecución relativas al multiprocesador de gráficos 234 de la Figura 2D. Por ejemplo, el multiprocesador de gráficos 325 puede incluir múltiples instancias de la unidad de instrucciones 332A-332B, el archivo de registro 334A-334B y la unidad o unidades de textura 344A-344B. El multiprocesador de gráficos 325 también incluye múltiples conjuntos de unidades de ejecución de gráficos o de cálculo (p. ej., el núcleo de GPGPU 336A-336B, el núcleo de GPGPU 337A-337B, el núcleo de GPGPU 338A-338B) y múltiples conjuntos de unidades de carga/almacenamiento 340A-340B. En una realización, las unidades de recurso de ejecución tienen una caché de instrucciones común 330, memoria caché de textura y/o de datos 342 y memoria compartida 346.

Los diversos componentes pueden comunicarse mediante un tejido de interconexión 327. En una realización, el tejido de interconexión 327 incluye uno o más conmutadores de barra transversal para posibilitar la comunicación entre los diversos componentes del multiprocesador de gráficos 325. En una realización, el tejido de interconexión 327 es una capa de tejido de red de alta velocidad separada en la que se apila cada componente del multiprocesador de gráficos 325. Los componentes del multiprocesador de gráficos 325 se comunican con componentes remotos mediante el tejido de interconexión 327. Por ejemplo, cada uno de los núcleos de GPGPU 336A-336B, 337A-337B y 3378A-338B puede comunicarse con la memoria compartida 346 mediante el tejido de interconexión 327. El tejido de interconexión 327 puede arbitrar la comunicación dentro del multiprocesador de gráficos 325 para garantizar una asignación de ancho de banda equitativa entre componentes.

La Figura 3B muestra un multiprocesador de gráficos 350 de acuerdo con una realización adicional. El procesador de gráficos incluye múltiples conjuntos de recursos de ejecución 356A-356D, donde cada conjunto de recursos de ejecución incluye múltiples unidades de instrucciones, archivos de registro, núcleos de GPGPU y unidades de carga-almacenamiento, como se ilustra en la Figura 2D y en la Figura 3A. Los recursos de ejecución 356A-356D pueden funcionar en conjunto con la unidad o unidades de textura 360A-360D para operaciones de textura, mientras que comparten una caché de instrucciones 354 y la memoria compartida 362. En una realización, los recursos de ejecución 356A-356D pueden compartir una caché de instrucciones 354 y una memoria compartida 362, así como múltiples instancias de una memoria de textura y/o de caché de datos 358A-358B. Los diversos componentes pueden comunicarse mediante un tejido de interconexión 352 similar al tejido de interconexión 327 de la Figura 3A.

Los expertos en la materia entenderán que la arquitectura descrita en las Figuras 1, 2A-2D y 3A-3B es descriptiva y no limitante en cuanto al alcance de las presentes realizaciones. Por tanto, las técnicas descritas en el presente documento pueden implementarse en cualquier unidad de procesamiento configurada apropiadamente, que incluye, sin limitación, uno o más procesadores de aplicación móvil, una o más unidades de procesamiento central (CPU) de sobremesa o servidor que incluyen CPU de múltiples núcleos, una o más unidades de procesamiento paralelo, tal como la unidad de procesamiento paralelo 202 de la Figura 2, así como uno o más procesadores de gráficos o unidades de procesamiento de propósito especial, sin alejarse del alcance de las realizaciones descritas en el presente documento.

En algunas realizaciones, un procesador paralelo o GPGPU como se describe en el presente documento está acoplado de manera comunicativa a núcleos de anfitrión/procesador para acelerar operaciones de gráficos, operaciones de aprendizaje automático, operaciones de análisis de patrones y diversas funciones de GPU de propósito general (GPGPU). La GPU puede estar acoplada de manera comunicativa al procesador de anfitrión/núcleos a través de un bus u otra interconexión (p. ej., una interconexión de alta velocidad tal como PCIe o NVLink). En otras realizaciones, la GPU puede estar integrada en el mismo paquete o chip que los núcleos y estar acoplada de manera comunicativa a los núcleos a través de un bus/interconexión de procesador interno (es decir, interno al paquete o chip). Independientemente de la manera en la que esté conectada la GPU, los núcleos de procesador pueden asignar trabajo a la GPU en forma de secuencias de comandos/instrucciones contenidas en un descriptor de trabajo. La GPU, a continuación, usa circuitería/lógica especializada para el procesamiento de manera eficaz de estos comandos/instrucciones.

Técnicas para interconexión de GPU a procesador de anfitrión

La Figura 4A ilustra una arquitectura ilustrativa en la que varias GPU 410-413 están acopladas de manera comunicativa a una pluralidad de procesadores de múltiples núcleos 405-406 a través de enlaces de alta velocidad 440-443 (p. ej., buses, interconexiones de punto a punto, etc.). En una realización, los enlaces de alta velocidad 440-443 soportan un caudal de comunicación de 4 GB/s, 30 GB/s, 80 GB/s o mayor, dependiendo de la implementación. Pueden usarse diversos protocolos de interconexión que incluyen, pero sin limitación, PCIe 4.0 o 5.0 y NVLink 2.0. Sin embargo, los principios subyacentes de la invención no están limitados a ningún protocolo o caudal de comunicación particular.

Además, en una realización, dos o más de las GPU 410-413 están interconectadas a través de enlaces de alta velocidad 444-445, que pueden implementarse usando los mismos protocolos/enlaces o diferentes que aquellos usados para enlaces de alta velocidad 440-443. De manera similar, dos o más de los procesadores de múltiples núcleos 405-406 pueden conectarse a través del enlace de alta velocidad 433 que pueden ser buses de múltiples procesadores simétricos (SMP) que operan a 20 GB/s, 30 GB/s, 120 GB/s o mayor. Como alternativa, toda la comunicación entre los diversos componentes de sistema mostrados en la Figura 4A puede conseguirse usando los mismos protocolos/enlaces (p. ej., a través de un tejido de interconexión común). Sin embargo, como se ha mencionado, los principios subyacentes de la invención no están limitados a ningún tipo particular de tecnología de interconexión.

En una realización, cada procesador de múltiples núcleos 405-406 está acoplado de manera comunicativa a una memoria de procesador 401-402, mediante las interconexiones de memoria 430-431, respectivamente, y cada GPU 410-413 está acoplada de manera comunicativa a la memoria de GPU 420-423 a través de las interconexiones de memoria de GPU 450-453, respectivamente. Las interconexiones de memoria 430-431 y 450-453 pueden utilizar las mismas tecnologías de acceso de memoria, o unas diferentes. A modo de ejemplo, y no como limitación, las memorias de procesador 401-402 y las memorias de GPU 420-423 pueden ser memorias volátiles, tales como memorias de acceso aleatorio dinámicas (DRAM) (que incluyen DRAM apiladas), SDRAM DDR de gráficos (GDDR) (p. ej., GDDR5, GDDR6), o Memoria de Ancho de Banda Alto (HBM) y/o pueden ser memorias no volátiles, tales como 3D XPoint o Nano-RAM. En una realización, alguna porción de las memorias puede ser memoria volátil y otra porción puede ser memoria no volátil (p. ej., usando una jerarquía de memoria de dos niveles (2LM)).

Como se describe a continuación, aunque los diversos procesadores 405-406 y las GPU 410-413 pueden estar físicamente acoplados a una memoria particular 401-402, 420-423, respectivamente, puede implementarse una arquitectura de memoria unificada en la que el mismo espacio de direcciones de sistema virtual (también denominado espacio "de direcciones eficaces") está distribuido entre todas las diversas memorias físicas. Por ejemplo, cada una de las memorias de procesador 401-402 puede comprender 64 GB del espacio de direcciones de memoria de sistema y cada una de las memorias de GPU 420-423 puede comprender 32 GB del espacio de direcciones de memoria de sistema (dando como resultado un total de memoria direccionable de 256 GB en este ejemplo).

La Figura 4B ilustra detalles adicionales para una interconexión entre un procesador de múltiples núcleos 407 y un módulo de aceleración de gráficos 446 de acuerdo con una realización. El módulo de aceleración de gráficos 446 puede incluir uno o más chips de GPU integrados en una tarjeta de línea que está acoplada al procesador 407 mediante el enlace de alta velocidad 440. Como alternativa, el módulo de aceleración de gráficos 446 puede estar integrado en el mismo paquete o chip que el procesador 407.

El procesador ilustrado 407 incluye una pluralidad de núcleos 460A-460D, cada uno con una memoria intermedia de traducción adelantada 461A-461D y una o más cachés 462A-462D. Los núcleos pueden incluir diversos otros componentes para ejecutar instrucciones y procesar datos que no se ilustran para evitar oscurecer los principios subyacentes de la invención (p. ej., unidades de extracción de instrucción, unidades de predicción de ramal, decodificadores, unidades de ejecución, memorias intermedias de reordenación, etc.). Las cachés 462A-462D pueden comprender cachés de nivel 1 (L1) y de nivel 2 (L2). Además, pueden incluirse una o más cachés compartidas 426 en la jerarquía de almacenamiento en caché y compartirse por conjuntos de los núcleos 460A-460D. Por ejemplo, una realización del procesador 407 incluye 24 núcleos, cada uno con su propia caché L1, doce cachés L2 compartidas y doce cachés L3 compartidas. En esta realización, una de las cachés L2 y L3 está compartida por dos núcleos adyacentes. El procesador 407 y el módulo de integración de acelerador de gráficos 446 se conectan con la memoria de sistema 441, que puede incluir las memorias de procesador 401-402

Se mantiene la coherencia para los datos e instrucciones almacenados en las diversas cachés 462A-462D, 456 y en la memoria de sistema 441 mediante la comunicación internúcleo a través de un bus de coherencia 464. Por ejemplo, cada caché puede tener una lógica/circuitería de coherencia de caché asociada con la misma para comunicarse a través del bus de coherencia 464 en respuesta a lecturas o escrituras detectadas en líneas de caché particulares. En una implementación, se implementa un protocolo de monitorización de caché a través del bus de coherencia 464 para monitorizar los accesos de caché. Las técnicas de monitorización/coherencia de caché son bien entendidas por los expertos en la materia y no se describirán en detalle en este punto para evitar oscurecer los principios subyacentes de la invención.

En una realización, un circuito de intermediario 425 acopla de manera comunicativa el módulo de aceleración de gráficos 446 al bus de coherencia 464, permitiendo que el módulo de aceleración de gráficos 446 participe en el protocolo de coherencia de caché como un homólogo de los núcleos. En particular, una interfaz 435 proporciona

conectividad al circuito de intermediario 425 a través del enlace de alta velocidad 440 (p. ej., un bus PCIe, NVLink, etc.) y una interfaz 437 conecta el módulo de aceleración de gráficos 446 al enlace 440.

En una implementación, un circuito de integración de acelerador 436 proporciona servicios de gestión de caché, de acceso de memoria, de gestión de contexto y de gestión de interrupciones en beneficio de una pluralidad de motores de procesamiento de gráficos 431, 432, N del módulo de aceleración de gráficos 446. Cada uno de los motores de procesamiento de gráficos 431, 432, N puede comprender una unidad de procesamiento de gráficos (GPU) separada. Como alternativa, los motores de procesamiento de gráficos 431, 432, N pueden comprender diferentes tipos de motores de procesamiento de gráficos dentro de una GPU, tales como unidades de ejecución de gráficos, motores de procesamiento de medios (p. ej., codificadores/decodificadores de vídeo), muestreadores y motores de BLIT. En otras palabras, el módulo de aceleración de gráficos puede ser una GPU con una pluralidad de motores de procesamiento de gráficos 431-432, N, o los motores de procesamiento de gráficos 431-432, N pueden ser unas GPU individuales integradas en un paquete, tarjeta de línea o chip común.

En una realización, el circuito de integración de acelerador 436 incluye una unidad de gestión de memoria (MMU) 439 para realizar diversas funciones de gestión de memoria tales como traducciones de memoria virtual a física (también denominadas traducciones de memoria eficaz a real) y protocolos de acceso de memoria para acceder a la memoria de sistema 441. La MMU 439 puede incluir también una memoria intermedia de traducción adelantada (TLB) (no mostrada) para almacenar en caché las traducciones de dirección virtual/eficaz a física/real. En una implementación, una caché 438 almacena comandos y datos para un acceso eficiente por los motores de procesamiento de gráficos 431-432, N. En una realización, los datos almacenados en la caché 438 y en las memorias de gráficos 433-434, N se mantienen coherentes con las cachés de núcleo 462A-462D, 456 y la memoria de sistema 411. Como se ha mencionado, esto puede conseguirse mediante el circuito de intermediario 425 que toma parte en el mecanismo de coherencia de caché en nombre de la caché 438 y las memorias 433-434, N (p. ej., enviando actualizaciones a la caché 438 relacionadas con las modificaciones/accesos de líneas de caché en las cachés del procesador 462A-462D, 456 y recibiendo actualizaciones de la caché 438).

Un conjunto de registros 445 almacenan datos de contexto para hilos ejecutados por los motores de procesamiento de gráficos 431-432, N y un circuito de gestión de contexto 448 gestiona los contextos de hilo. Por ejemplo, el circuito de gestión de contexto 448 puede realizar operaciones de grabación y restauración para grabar y restaurar contextos de los diversos hilos durante cambios de contexto (p. ej., cuando se graba un primer hilo y se almacena un segundo hilo de modo que el segundo hilo puede ejecutarse por un motor de procesamiento de gráficos). Por ejemplo, en un cambio de contexto, el circuito de gestión de contexto 448 puede almacenar valores de registro actuales en una región designada en memoria (p. ej., identificada por un puntero de contexto). A continuación, puede restablecer los valores de registro cuando se vuelve al contexto. En una realización, un circuito de gestión de interrupciones 447 recibe y procesa interrupciones recibidas desde los dispositivos de sistema.

En una implementación, direcciones virtuales/eficaces desde un motor de procesamiento de gráficos 431 son traducidas, por la MMU 439, a direcciones reales/físicas en la memoria de sistema 411. Una realización del circuito de integración de acelerador 436 soporta múltiples (p. ej., 4, 8, 16) módulos de acelerador de gráficos 446 y/u otros dispositivos aceleradores. El módulo de acelerador de gráficos 446 puede estar especializado en una única aplicación ejecutada en el procesador 407 o puede compartirse entre múltiples aplicaciones. En una realización, se presenta un entorno de ejecución de gráficos virtualizado en el que los recursos de los motores de procesamiento de gráficos 431-432, N se comparten con múltiples aplicaciones o máquinas virtuales (VM). Los recursos pueden subdividirse en "cortes" que se asignan a diferentes VM y/o aplicaciones basándose en los requisitos de procesamiento y las propiedades asociadas con las VM y/o las aplicaciones.

Por lo tanto, el circuito de integración de acelerador actúa como un puente al sistema para el módulo de aceleración de gráficos 446 y proporciona servicios de traducción de direcciones y de caché de memoria de sistema. Además, el circuito de integración de acelerador 436 puede proporcionar instalaciones de virtualización para que el procesador de anfitrión gestione la virtualización de los motores de procesamiento de gráficos, las interrupciones y la gestión de memoria.

Debido a que los recursos de hardware de los motores de procesamiento de gráficos 431-432, N se mapean explícitamente al espacio de direcciones real observado por el procesador de anfitrión 407, cualquier procesador de anfitrión puede dirigir estos recursos directamente usando un valor de dirección eficaz. Una función del circuito de integración de acelerador 436, en una realización, es la separación física de los motores de procesamiento de gráficos 431-432, N de modo que aparecen al sistema como unidades independientes.

Como se ha mencionado, en la realización ilustrada, una o más memorias de gráficos 433-434, M están acopladas a cada uno de los motores de procesamiento de gráficos 431-432, N, respectivamente. Las memorias de gráficos 433-434, M almacenan instrucciones y datos que se procesan por cada uno de los motores de procesamiento de gráficos 431-432, N. Las memorias de gráficos 433-434, M pueden ser memorias volátiles, tales como DRAM (incluyendo DRAM apiladas), memoria de GDDR (p. ej., GDDR5, GDDR6), o HBM, y/o pueden ser memorias no volátiles, tales como 3D XPoint o Nano-Ram.

En una realización, para reducir el tráfico de datos a través del enlace 440, se usan técnicas de desvío para garantizar que los datos almacenados en las memorias de gráficos 433-434, M son datos que serán usados de la manera más frecuente por los motores de procesamiento de gráficos 431-432, N y preferentemente no serán usados por los núcleos 460A-460D (al menos, no con frecuencia). De manera similar, el mecanismo de desvío intenta mantener datos que necesitan los núcleos (y, preferentemente, no los motores de procesamiento de gráficos 431-432, N) dentro de las cachés 462A-462D, 456 de los núcleos y la memoria de sistema 411.

La Figura 4C ilustra otra realización en la que el circuito de integración de acelerador 436 está integrado dentro del procesador 407. En esta realización, los motores de procesamiento de gráficos 431-432, N se comunican directamente a través del enlace de alta velocidad 440 al circuito de integración de acelerador 436 mediante la interfaz 437 y la interfaz 435 (que, de nuevo, puede utilizar cualquier forma de bus o protocolo de interfaz). El circuito de integración de acelerador 436 puede realizar las mismas operaciones que las descritas con respecto a la Figura 4B, pero potencialmente a un caudal superior dada su proximidad estrecha al bus de coherencia 462 y a las cachés 462A-462D, 426.

Una realización soporta diferentes modelos de programación que incluyen un modelo de programación de proceso especializado (sin virtualización de módulo de aceleración de gráficos) y modelos de programación compartida (con virtualización). El último puede incluir modelos de programación que se controlan por el circuito de integración de acelerador 436 y modelos de programación que se controlan por el módulo de aceleración de gráficos 446.

En una realización del modelo de proceso especializado, los motores de procesamiento de gráficos 431-432, N están especializados en una única aplicación o proceso bajo un único sistema operativo. La única aplicación puede canalizar otras solicitudes de aplicación a los motores de gráficos 431-432, N, lo que proporciona virtualización dentro de una VM/subdivisión.

En los modelos de programación de proceso especializado, los motores de procesamiento de gráficos 431-432, N, pueden estar compartidos por múltiples subdivisiones de aplicación/VM. Los modelos compartidos requieren que un hipervisor de sistema virtualice los motores de procesamiento de gráficos 431-432, N para permitir el acceso de cada sistema operativo. Para sistemas de subdivisión única sin un hipervisor, los motores de procesamiento de gráficos 431-432, N son propiedad del sistema operativo. En ambos casos, el sistema operativo puede virtualizar los motores de procesamiento de gráficos 431-432, N para proporcionar acceso a cada proceso o aplicación.

Para el modelo de programación compartida, el módulo de aceleración de gráficos 446 o un motor de procesamiento de gráficos individual 431-432, N selecciona un elemento de proceso usando un manejador de proceso. En una realización, los elementos de proceso se almacenan en memoria de sistema 411 y son direccionables usando las técnicas de traducción de dirección eficaz a dirección real descritas en el presente documento. El manejador de proceso puede ser un valor específico de la implementación proporcionado al proceso de anfitrión cuando se registra su contexto con el motor de procesamiento de gráficos 431-432, N (es decir, llamando al software de sistema para añadir el elemento de proceso a la lista vinculada de elementos de proceso). Los 16 bits inferiores del manejador de proceso pueden ser el desplazamiento del elemento de proceso dentro de la lista vinculada de elementos de proceso.

La Figura 4D ilustra un corte de integración de acelerador 490 ilustrativo. Como se usa en el presente documento, un "corte" comprende una porción especificada de los recursos de procesamiento del circuito de integración de acelerador 436. El espacio de direcciones eficaces de aplicación 482 dentro de la memoria de sistema 411 almacena los elementos de proceso 483. En una realización, los elementos de proceso 483 se almacenan en respuesta a invocaciones de GPU 481 desde las aplicaciones 480 ejecutadas en el procesador 407. Un elemento de proceso 483 contiene el estado de proceso para la correspondiente aplicación 480. Un descriptor de trabajo (WD) 484 contenido en el elemento de proceso 483 puede ser un único trabajo solicitado por una aplicación o puede contener un puntero a una cola de trabajos. En el último caso, el WD 484 es un puntero a la cola de solicitudes de trabajo en el espacio de direcciones de la aplicación 482.

El módulo de aceleración de gráficos 446 y/o los motores de procesamiento de gráficos individuales 431-432, N pueden compartirse por todos o un subconjunto de los procesos en el sistema. Las realizaciones de la invención incluyen una infraestructura para configurar el estado de proceso y enviar un WD 484 a un módulo de aceleración de gráficos 446 para empezar un trabajo en un entorno virtualizado.

En una implementación, el modelo de programación de proceso especializado es específico de la implementación. En este modelo, un único proceso posee el módulo de aceleración de gráficos 446 o un motor de procesamiento de gráficos individual 431. Debido a que el módulo de aceleración de gráficos 446 es de propiedad de un único proceso, el hipervisor inicializa el circuito de integración de acelerador 436 para la subdivisión de propiedad y el sistema operativo inicializa el circuito de integración de acelerador 436 para el proceso de propiedad en el momento cuando se asigna el módulo de aceleración de gráficos 446.

En la operación, una unidad de extracción de WD 491 en el corte de integración de acelerador 490 extrae el siguiente WD 484 que incluye una indicación del trabajo que va a hacerse por uno de los motores de procesamiento de gráficos del módulo de aceleración de gráficos 446. Los datos del WD 484 pueden almacenarse en los registros 445 y usarse

por la MMU 439, el circuito de gestión de interrupciones 447 y/o el circuito de gestión de contexto 446 como se ilustra. Por ejemplo, una realización de la MMU 439 incluye circuitería de paso de segmento/página para acceder a tablas de segmentos/páginas 486 dentro del espacio de direcciones virtual del SO 485. El circuito de gestión de interrupciones 447 puede procesar eventos de interrupción 492 recibidos del módulo de aceleración de gráficos 446. Cuando se realizan operaciones de gráficos, se traduce una dirección eficaz 493 generada por un motor de procesamiento de gráficos 431-432, N a una dirección real por la MMU 439.

En una realización, se duplica el mismo conjunto de registros 445 para cada motor de procesamiento de gráficos 431-432, N y/o módulo de aceleración de gráficos 446 y puede inicializarse por el hipervisor o el sistema operativo. Cada uno de estos registros duplicados puede incluirse en un corte de integración de acelerador 490. Se muestran los registros ilustrativos que pueden inicializarse por el hipervisor en la Tabla 1.

Tabla 1 - Registros inicializados por el hipervisor

1	Registro de control de corte
2	Puntero de área de procesos planificados de dirección real (RA)
3	Registro de anulación de máscara de autoridad
4	Desplazamiento de entrada de tabla de vector de interrupción
5	Límite de entrada de tabla de vector de interrupción
6	Registro de estado
7	ID de subdivisión lógica
8	Puntero de registro de utilización de acelerador de hipervisor de dirección real (RA)
9	Registro de descripción de almacenamiento

Se muestran los registros ilustrativos que pueden inicializarse por el sistema operativo en la Tabla 2.

Tabla 2 - Registros inicializados por el sistema operativo

1	Identificación de proceso y de hilo
2	Puntero de grabación/restauración de contexto de dirección eficaz (EA)
3	Puntero de registro de utilización de acelerador de dirección virtual (VA)
4	Puntero de tabla de segmentos de almacenamiento de dirección virtual (VA)
5	Máscara de autoridad
6	Descriptor de trabajo

En una realización, cada WD 484 es específico de un módulo de aceleración de gráficos 446 y/o de un motor de procesamiento de gráficos 431-432, N particular. Este contiene toda la información que requiere un motor de procesamiento de gráficos 431-432, N para hacer su trabajo, o puede ser un puntero a una ubicación de memoria en la que la aplicación ha establecido una cola de comandos de trabajo que hay que completar.

La Figura 4E ilustra detalles adicionales para una realización de un modelo compartido. Esta realización incluye un espacio de direcciones real de hipervisor 498 en el que se almacena una lista de elementos de proceso 499. El espacio de direcciones real de hipervisor 498 es accesible mediante un hipervisor 496 que virtualiza los motores de módulo de aceleración de gráficos para el sistema operativo 495.

Los modelos de programación compartidos permiten que todos o un subconjunto de procesos de todas o un subconjunto de las subdivisiones en el sistema usen un módulo de aceleración de gráficos 446. Hay dos modelos de programación donde el módulo de aceleración de gráficos 446 se comparte por múltiples procesos y subdivisiones: compartido en cortes de tiempo y compartido dirigido a gráficos.

En este modelo, el hipervisor de sistema 496 tiene propiedad del módulo de aceleración de gráficos 446 y hace que su función esté disponible para todos los sistemas operativos 495. Para que un módulo de aceleración de gráficos 446 soporte virtualización por el hipervisor de sistema 496, el módulo de aceleración de gráficos 446 puede adherirse a los siguientes requisitos: 1) Una solicitud de trabajo de la aplicación debe ser autónoma (es decir, el estado no necesita mantenerse entre trabajos), o el módulo de aceleración de gráficos 446 debe proporcionar un mecanismo de grabación y restauración de contexto. 2) Se garantiza una solicitud de trabajo de la aplicación por el módulo de aceleración de gráficos 446 para completar en una cantidad especificada de tiempo, que incluye cualquier fallo de traducción, o el módulo de aceleración de gráficos 446 proporciona la capacidad de anticiparse al procesamiento del trabajo. 3) El

módulo de aceleración de gráficos 446 debe garantizar equidad entre procesos cuando opera en el modelo de programación compartido dirigido.

En una realización, para el modelo compartido, se requiere que la aplicación 480 haga una llamada de sistema a un sistema operativo 495 con un tipo de módulo de aceleración de gráficos 446, un descriptor de trabajo (WD), un valor de registro de máscara de autoridad (AMR) y un puntero de área de grabación/restauración de contexto (CSRP). El tipo de módulo de aceleración de gráficos 446 describe la función de aceleración dirigida para la llamada de sistema. El tipo de módulo de aceleración de gráficos 446 puede ser un valor específico de sistema. El WD se formatea específicamente para el módulo de aceleración de gráficos 446 y puede estar en forma de un comando de módulo de aceleración de gráficos 446, un puntero de dirección eficaz a una estructura definida por el usuario, un puntero de dirección eficaz a una cola de comandos o cualquier otra estructura de datos para describir el trabajo que va a hacerse por el módulo de aceleración de gráficos 446. En una realización, el valor de AMR es el estado de AMR para su uso para el proceso actual. El valor pasado al sistema operativo es similar a una aplicación que configura el AMR. Si las implementaciones del circuito de integración de acelerador 436 y del módulo de aceleración de gráficos 446 no soportan un registro de anulación de máscara de autoridad de usuario (UAMOR), el sistema operativo puede aplicar el valor de UAMOR actual al valor de AMR antes de pasar el AMR en la llamada de hipervisor. El hipervisor 496 puede aplicar opcionalmente el valor de registro de anulación de máscara de autoridad (AMOR) actual antes de colocar el AMR en el elemento de proceso 483. En una realización, el CSRP es uno de los registros 445 que contiene la dirección eficaz de un área en el espacio de direcciones de la aplicación 482 para que el módulo de aceleración de gráficos 446 grabe y restaure el estado de contexto. Este puntero es opcional si no se requiere que se grabe estado entre trabajos o cuando se anticipa un trabajo. El área de grabación/restauración de contexto puede estar fijada en la memoria de sistema.

Después de recibir la llamada de sistema, el sistema operativo 495 puede verificar que se ha registrado la aplicación 480 y que se le ha dado la autoridad para usar el módulo de aceleración de gráficos 446. El sistema operativo 495, a continuación, llama al hipervisor 496 con la información mostrada en la Tabla 3.

Tabla 3 - Parámetros de llamada de SO al hipervisor

1	Un descriptor de trabajo (WD)
2	Un valor (potencialmente enmascarado) de registro de máscara de autoridad (AMR).
3	Un puntero de área de grabación/restauración de contexto (CSRP) de dirección eficaz (EA)
4	Un ID de proceso (PID) e ID de hilo opcional (TID)
5	Un puntero de registro de utilización de acelerador (AURP) de dirección virtual (VA)
6	La dirección virtual del puntero de tabla de segmentos de almacenamiento (SSTP)
7	Un número de servicio de interrupción lógica (LISN)

Después de recibir la llamada del hipervisor, el hipervisor 496 verifica que se ha registrado el sistema operativo 495 y se le ha dado la autoridad para usar el módulo de aceleración de gráficos 446. El hipervisor 496, a continuación, pone el elemento de proceso 483 en la lista de elementos de proceso vinculados para el correspondiente tipo de módulo de aceleración de gráficos 446. El elemento de proceso puede incluir la información mostrada en la Tabla 4.

Tabla 4 - Información de elemento de proceso

1	Un descriptor de trabajo (WD)
2	Un valor (potencialmente enmascarado) de registro de máscara de autoridad (AMR).
3	Un puntero de área de grabación/restauración de contexto (CSRP) de dirección eficaz (EA)
4	Un ID de proceso (PID) e ID de hilo opcional (TID)
5	Un puntero de registro de utilización de acelerador (AURP) de dirección virtual (VA)
6	La dirección virtual del puntero de tabla de segmentos de almacenamiento (SSTP)
7	Un número de servicio de interrupción lógica (LISN)
8	Tabla de vectores de interrupción, derivada de los parámetros de llamada de hipervisor.
9	Un valor de registro de estado (SR)
10	Un ID de subdivisión lógica (LPID)
11	Un puntero de registro de utilización de acelerador de hipervisor de dirección real (RA)
12	El registro de descriptor de almacenamiento (SDR)

En una realización, el hipervisor inicializa una pluralidad de registros 445 de corte de integración de acelerador 490.

5 Como se ha ilustrado en la Figura 4F, una realización de la invención emplea una memoria unificada direccionable mediante un espacio de direcciones de memoria virtual común usado para acceder a las memorias de procesador físico 401-402 y a las memorias de GPU 420-423. En esta implementación, las operaciones ejecutadas en las GPU 410-413 utilizan el mismo espacio de direcciones de memoria virtual/eficaz para acceder a las memorias de procesadores 401-402, y viceversa, simplificando de esta manera la programabilidad. En una realización, una primera
10 porción del espacio de direcciones virtual/eficaz está asignada a la memoria de procesador 401, una segunda porción a la segunda memoria de procesador 402, una tercera porción a la memoria de GPU 420, y así sucesivamente. El espacio de memoria virtual/eficaz total (denominado, en ocasiones, el espacio de direcciones eficaz) está distribuido, de esta manera, a lo largo de cada una de las memorias de procesador 401-402 y de las memorias de GPU 420-423, permitiendo que cualquier procesador o GPU acceda a cualquier memoria física con una dirección virtual mapeada a esa memoria.
15

En una realización, la circuitería de gestión de desvío/coherencia 494A-494E dentro de una o más de las MMU 439A-439E garantiza la coherencia de caché entre las cachés de los procesadores de anfitrión (p. ej., 405) y las GPU 410-413 e implementa técnicas de desvío que indican las memorias físicas en las que deberían almacenarse ciertos tipos
20 de datos. Aunque se ilustran múltiples instancias de la circuitería de gestión de desvío/coherencia 494A-494E en la Figura 4F, la circuitería de desvío/coherencia puede implementarse dentro de la MMU de uno o más procesadores de anfitrión 405 y/o dentro del circuito de integración de acelerador 436.

Una realización permite que la memoria adjunta a la GPU 420-423 se mapee como parte de memoria de sistema, y que se acceda a ella usando tecnología de memoria virtual compartida (SVM), pero sin sufrir las desventajas de rendimiento típicas asociadas con la coherencia de caché de sistema completa. La capacidad de que se acceda a la memoria adjunta a la GPU 420-423 como memoria de sistema sin sobrecarga de coherencia de caché onerosa proporciona un entorno de operación beneficioso para la descarga de la GPU. Esta disposición permite que el software del procesador de anfitrión 405 establezca operandos y acceda a resultados de cálculo, sin la sobrecarga de las copias de datos de DMA de E/S tradicionales. Tales copias tradicionales implican llamadas de controlador, interrupciones y accesos de E/S mapeados con memoria (MMIO) que son, todos ellos, ineficientes en relación con los accesos de memoria sencillos. Al mismo tiempo, la capacidad de acceder a la memoria adjunta a la GPU 420-423 sin sobrecargas de coherencia de caché puede ser crítica para el tiempo de ejecución de un cálculo descargado. En casos con tráfico de memoria de escritura de envío por flujo continuo sustancial, por ejemplo, la sobrecarga de coherencia de caché puede reducir significativamente el ancho de banda de escritura eficaz observado por una GPU 410-413. La eficacia de la configuración del operando, la eficacia del acceso a los resultados y la eficacia del cálculo de GPU, todos desempeñan un papel al determinar la eficacia de la descarga de la GPU.
35

En una implementación, la selección entre el desvío de GPU y el desvío de procesador de anfitrión se controla por una estructura de datos del rastreador de desvío. Puede usarse una tabla de desvío, por ejemplo, que puede ser una estructura de página granular (es decir, controlada a la granularidad de una página de memoria) que incluye 1 o 2 bits por página de memoria adjunta a la GPU. La tabla de desvío puede implementarse en un rango de memoria robado de una o más memorias adjuntas a la GPU 420-423, con o sin una caché de desvío en la GPU 410-413 (p. ej., para entradas usadas de manera frecuente/reciente de caché de la tabla de desvío). Como alternativa, toda la tabla de desvíos puede mantenerse dentro de la GPU.
45

En una implementación, se accede a la entrada de tabla de desvío asociada con cada acceso a la memoria adjunta a la GPU 420-423 antes del acceso real a la memoria de GPU, lo que provoca las siguientes operaciones. En primer lugar, las solicitudes locales de la GPU 410-413 que encuentran su página en el desvío de GPU se reenvían directamente a una correspondiente memoria de GPU 420-423. Las solicitudes locales de la GPU que encuentran su página en el desvío de anfitrión se reenvían al procesador 405 (p. ej., a través de un enlace de alta velocidad como se ha analizado anteriormente). En una realización, las solicitudes del procesador 405 que encuentran la página solicitada en el desvío de procesador de anfitrión completan la solicitud como una lectura de memoria normal. Como alternativa, las solicitudes dirigidas a una página con desvío de GPU pueden reenviarse a la GPU 410-413. La GPU puede pasar, a continuación, la página a un desvío de procesador de anfitrión si no está usando actualmente la página.

El estado de desvío de una página puede cambiarse mediante un mecanismo basado en software, un mecanismo basado en software asistido por hardware o, para un conjunto de casos limitado, un mecanismo basado puramente en hardware.

Un mecanismo para cambiar el estado de desvío emplea una llamada de API (p. ej., OpenCL), que, a su vez, llama al controlador del dispositivo de la GPU que, a su vez, envía un mensaje (o pone en cola un descriptor de comando) a la GPU que la dirige para cambiar el estado de desvío y, para algunas transiciones, realiza una operación de vaciado de caché en el anfitrión. Se requiere la operación de vaciado de caché para una transición desde el procesador de anfitrión 405 a un desvío de GPU, pero no se requiere para la transacción opuesta.

En una realización, la coherencia de caché se mantiene haciendo temporalmente que las páginas con desvío de GPU no puedan ser almacenadas en caché por el procesador de anfitrión 405. Para acceder a estas páginas, el procesador 405 puede solicitar acceso desde la GPU 410 que puede conceder o no el acceso de inmediato, dependiendo de la implementación. Por lo tanto, para reducir la comunicación entre el procesador 405 y la GPU 410, es beneficioso garantizar que las páginas con desvío de GPU son aquellas que son requeridas por la GPU, pero no por el procesador de anfitrión 405, y viceversa.

Canalización de procesamiento de gráficos

La Figura 5 ilustra una canalización de procesamiento de gráficos 500, de acuerdo con una realización. En una realización, un procesador de gráficos puede implementar la canalización de procesamiento de gráficos 500 ilustrada. El procesador de gráficos puede estar incluido dentro de los subsistemas de procesamiento paralelo como se describe en el presente documento, tal como el procesador paralelo 200 de la Figura 2, que, en una realización, es una variante del procesador o procesadores paralelos 112 de la Figura 1. Los diversos sistemas de procesamiento paralelo pueden implementar la canalización de procesamiento de gráficos 500 mediante una o más instancias de la unidad de procesamiento paralelo (p. ej., la unidad de procesamiento paralelo 202 de la Figura 2) como se describe en el presente documento. Por ejemplo, una unidad sombreadora (p. ej., el multiprocesador de gráficos 234 de la Figura 3) puede estar configurada para realizar las funciones de una o más de una unidad de procesamiento de vértices 504, una unidad de control de proceso de teselación 508, una unidad de procesamiento de evaluación de teselación 512, una unidad de procesamiento de geometría 516 y una unidad de procesamiento de fragmentos/píxeles 524. Las funciones del ensamblador de datos 502, los ensambladores de primitivas 506, 514, 518, la unidad de teselación 510, el rasterizador 522, y la unidad de operaciones del rasterizador 526 pueden realizarse también por otros motores de procesamiento dentro de una agrupación de procesamiento (p. ej., la agrupación de procesamiento 214 de la Figura 3) y una correspondiente unidad de subdivisión (p. ej., la unidad de subdivisión 220A-220N de la Figura 2). La canalización de procesamiento de gráficos 500 puede implementarse también usando unidades de procesamiento especializadas para una o más funciones. En una realización, pueden realizarse una o más porciones de la canalización de procesamiento de gráficos 500 mediante lógica de procesamiento paralelo dentro de un procesador de fin general (p. ej., la CPU). En una realización, una o más porciones de la canalización de procesamiento de gráficos 500 pueden acceder a una memoria en chip (p. ej., la memoria de procesador paralelo 222 como en la Figura 2) mediante una interfaz de memoria 528, que puede ser una instancia de la interfaz de memoria 218 de la Figura 2.

En una realización, el ensamblador de datos 502 es una unidad de procesamiento que recopila datos de vértices para superficies y primitivas. El ensamblador de datos 502, a continuación, emite los datos de vértices, que incluyen los atributos de vértices, a la unidad de procesamiento de vértices 504. La unidad de procesamiento de vértices 504 es una unidad de ejecución programable que ejecuta programas de sombreador de vértices, datos de vértices de iluminación y transformación como se especifica por los programas de sombreador de vértices. La unidad de procesamiento de vértices 504 lee datos que se almacenan en memoria caché, local o de sistema para su uso en el procesamiento de los datos de vértices y puede programarse para transformar los datos de vértices desde una representación de coordenadas basada en objetos hasta un espacio de coordenadas de espacio mundial o un espacio de coordenadas de dispositivo normalizado.

Una primera instancia de un ensamblador de primitivas 506 recibe atributos de vértice desde la unidad de procesamiento de vértices 504. El ensamblador de primitivas 506 lee atributos de vértice almacenados según sea necesario y construye primitivas de gráficos para su procesamiento por la unidad de procesamiento de control de teselación 508. Las primitivas de gráficos incluyen triángulos, segmentos de línea, puntos, parches y así

sucesivamente, según son soportados por diversas interfaces de programación de aplicación (API) de procesamiento de gráficos.

La unidad de procesamiento de control de teselación 508 trata los vértices de entrada como puntos de control para un parche geométrico. Los puntos de control se transforman de una representación de entrada a partir del parche (p. ej., las bases del parche) a una representación que es adecuada para su uso en una evaluación superficial por la unidad de procesamiento de evaluación de teselación 512. La unidad de procesamiento de control de teselación 508 también puede calcular factores de teselación para bordes de parches geométricos. Se aplica un factor de teselación a un único borde y cuantifica un nivel dependiente de la vista del detalle asociado con el borde. Una unidad de teselación 510 está configurada para recibir los factores de teselación para bordes de un parche y para teselar el parche en múltiples primitivas geométricas, tales como una línea, triángulo o primitivas cuadrilaterales, que se transmiten a una unidad de procesamiento de evaluación de teselación 512. La unidad de procesamiento de evaluación de teselación 512 opera en coordenadas parametrizadas del parche subdividido para generar una representación superficial y atributos de vértices para cada vértice asociado con las primitivas geométricas.

Una segunda instancia de un ensamblador de primitivas 514 recibe atributos de vértices desde la unidad de procesamiento de evaluación de teselación 512, que lee los atributos de vértices almacenados según sea necesario y construye primitivas de gráficos para su procesamiento por la unidad de procesamiento de geometría 516. La unidad de procesamiento de geometría 516 es una unidad de ejecución programable que ejecuta programas de sombreador de geometría para transformar primitivas de gráficos recibidas desde el ensamblador de primitivas 514 como se especifica por los programas de sombreador de geometría. En una realización, la unidad de procesamiento de geometría 516 está programada para subdividir las primitivas de gráficos en una o más primitivas de gráficos nuevas y calcular parámetros usados para rasterizar las nuevas primitivas de gráficos.

En algunas realizaciones, la unidad de procesamiento de geometría 516 puede añadir o borrar elementos en el flujo de geometría. La unidad de procesamiento de geometría 516 emite los parámetros y vértices que especifican primitivas de gráficos nuevas al ensamblador de primitivas 518. El ensamblador de primitivas 518 recibe los parámetros y vértices desde la unidad de procesamiento de geometría 516 y construye primitivas de gráficos para su procesamiento por una unidad de escalado, selección y recorte de ventana gráfica 520. La unidad de procesamiento de geometría 516 lee datos que están almacenados en la memoria de procesador paralelo o en la memoria de sistema para su uso en el procesamiento de los datos de geometría. La unidad de escalado, selección y recorte de ventana gráfica 520 realiza el recorte, selección y escalado de ventana gráfica y emite las primitivas de gráficos procesados a un rasterizador 522.

El rasterizador 522 puede realizar optimizaciones de selección de profundidad y otras basadas en la profundidad. El rasterizador 522 también realiza la conversión de exploración en las nuevas primitivas de gráficos para generar fragmentos y emitir aquellos fragmentos y datos de cobertura asociados a la unidad de procesamiento de fragmentos/píxeles 524. La unidad de procesamiento de fragmentos/píxeles 524 es una unidad de ejecución programable que está configurada para ejecutar programas de sombreador de fragmentos o programas de sombreador de píxeles. Transformando la unidad de procesamiento de fragmentos/píxeles 524 fragmentos o píxeles recibidos desde el rasterizador 522, como se especifica por los programas de sombreador de fragmentos o de píxeles. Por ejemplo, la unidad de procesamiento de fragmentos/píxeles 524 puede programarse para realizar operaciones que incluyen, pero sin limitación, mapeo de textura, sombreado, mezcla, corrección de textura y corrección de perspectiva para producir fragmentos o píxeles sombreados que se emiten a una unidad de operaciones de rasterización 526. La unidad de procesamiento de fragmentos/píxeles 524 puede leer datos que se almacenan en cualquiera de la memoria de procesador paralelo o la memoria de sistema para su uso cuando se procesan los datos de fragmento. Los programas de sombreador de fragmentos o de píxeles pueden estar configurados para sombrear a granularidad de muestra, de píxel, de pieza u otras dependiendo de las tasas de muestreo configuradas para las unidades de procesamiento.

La unidad de operaciones de rasterización 526 es una unidad de procesamiento que realiza operaciones de rasterización que incluyen, pero sin limitación, estarcido, prueba z, mezcla y similares, y emite datos de píxeles como datos de gráficos procesados para que se almacenen en la memoria de gráficos (p. ej., la memoria de procesador paralelo 222 como en la Figura 2, y/o la memoria de sistema 104 como en la Figura 1, para que se visualicen en el uno o más dispositivos de visualización 110 o para su procesamiento adicional por uno del uno o más procesadores 102 o procesadores paralelos 112. En algunas realizaciones, la unidad de operaciones de rasterización 526 está configurada para comprimir datos z o de color que se escriben en memoria y descomprimir datos z o de color que se leen desde la memoria.

Ejemplos de renderizado foveado con desvío de movimiento

Volviendo ahora a la Figura 6A, una realización de un sistema de procesamiento electrónico 600 puede incluir un procesador de aplicaciones 601, un medio de almacenamiento persistente 602 acoplado comunicativamente al procesador de aplicaciones 601 y un subsistema de gráficos 603 acoplado comunicativamente al procesador de aplicaciones 601. El sistema 600 puede incluir además un motor de detección 604 acoplado comunicativamente al subsistema de gráficos 603 para proporcionar información detectada, un motor de enfoque 605 acoplado comunicativamente al motor de detección 604 y al subsistema de gráficos 603 para proporcionar información de

enfoque, un motor de movimiento 606 acoplado comunicativamente al motor de detección 604, al motor de enfoque 605 y al subsistema de gráficos 603 para proporcionar información de movimiento, y un renderizador foveado 607 con desvío de movimiento acoplado comunicativamente al motor de movimiento 606, el motor de enfoque 605 y el motor de detección 604 para ajustar uno o más parámetros del subsistema de gráficos 603 en función de uno o más de la información de detección, la información de enfoque y la información de movimiento.

En algunas realizaciones del sistema 600, el motor de movimiento 606 puede incluir además un predictor de movimiento de aprendizaje automático. Algunas realizaciones del sistema 600 pueden incluir además un generador de mapa de sombras foveado acoplado comunicativamente al subsistema de gráficos 603 para generar un mapa de sombras foveado basado en la información de enfoque. Por ejemplo, el renderizador foveado 607 con desvío de movimiento también puede configurarse para sobredibujar un objetivo de renderizado en función de la información de movimiento. Según la invención, el renderizador foveado 607 con desvío de movimiento está configurado para identificar un área de enfoque futura basándose en la información de enfoque y la información de movimiento. El renderizador foveado 607 con desvío de movimiento también está configurado para rotar un área de enfoque en función de la información de movimiento.

Las realizaciones de cada uno de los procesadores de aplicaciones 601, medios de almacenamiento persistente 602, subsistema de gráficos 603, motor de detección 604, motor de enfoque 605, motor de movimiento 606, renderizador foveado 607 con desvío de movimiento y otros componentes del sistema anteriores pueden implementarse en hardware, software o cualquier otro sistema adecuado. Por ejemplo, las implementaciones de hardware pueden incluir lógica configurable tal como, por ejemplo, matrices lógicas programables (PLA), matrices de puertas programables en campo (FPGA), dispositivos lógicos programables complejos (CPLD), o en hardware lógico de funcionalidad fija usando tecnología de circuitos tal como, por ejemplo, ASIC, semiconductor de óxido metálico complementario (CMOS) o tecnología lógica de transistor-transistor (TTL), o cualquier combinación de las mismas. Como alternativa, o adicionalmente, estos componentes pueden implementarse en uno o más módulos como un conjunto de instrucciones lógicas almacenadas en un medio de almacenamiento legible por máquina u ordenador, tal como memoria de acceso aleatorio (RAM), memoria de solo lectura (ROM), ROM programable (PROM), firmware, memoria flash, etc., para ejecutarse por un procesador o dispositivo informático. Por ejemplo, el código de programa informático para llevar a cabo las operaciones de los componentes se puede escribir en cualquier combinación de uno o más lenguajes de programación apropiados/aplicables al sistema operativo, incluyendo un lenguaje de programación orientado a objetos tal como PYTHON, PERL, JAVA, SMALLTALK, C++, C# o similares y lenguajes de programación de procedurales convencionales, tales como el lenguaje de programación "C" o lenguajes de programación similares.

Por ejemplo, el sistema 600 puede incluir componentes y/o características similares a las del sistema 100, configurado además para desviar con movimiento una operación de representación foveada. Por ejemplo, el subsistema de gráficos 603 puede incluir componentes y/o características similares a las del sistema de procesamiento paralelo 200, configurado además con un renderizador foveado con desvío de movimiento como se describe en el presente documento. El sistema 600 también puede adaptarse para funcionar con un estéreo montado en cabeza.

Ejemplos de motor de detección

Volviendo ahora a la Figura 6B, un motor de detección 612 puede obtener información de sensores, contenido, servicios y/u otras fuentes para proporcionar información detectada. La información detectada puede incluir, por ejemplo, información de imagen, información de audio, información de movimiento, información de profundidad, información de temperatura, información biométrica, información de GPU, etc. En un alto nivel, algunas realizaciones pueden usar información detectada para ajustar los parámetros de rendimiento del sistema de gráficos.

Por ejemplo, un motor de detección puede incluir un concentrador de sensor acoplado comunicativamente a cámaras bidimensionales (2D), cámaras tridimensionales (3D), cámaras de profundidad, giroscopios, acelerómetros, unidades de medición inercial (IMU), servicios de ubicación, micrófonos, sensores de proximidad, termómetros, sensores biométricos, etc., y/o una combinación de múltiples fuentes que proporcionan información a los motores de enfoque y/o de movimiento. El concentrador de sensor se puede distribuir entre múltiples dispositivos. La información del concentrador de sensor puede incluir o combinarse con datos de entrada de los dispositivos de usuario (p. ej., datos táctiles).

Por ejemplo, el dispositivo o dispositivos de usuario pueden incluir una o más cámaras 2D, 3D y/o de profundidad. El dispositivo o dispositivos de usuario también pueden incluir giroscopios, acelerómetros, IMU, servicios de ubicación, termómetros, sensores biométricos, etc. Por ejemplo, el usuario puede llevar un teléfono inteligente (p. ej., en el bolsillo del usuario) y/o puede llevar un dispositivo llevable (p. ej., tal como un reloj inteligente, un monitor de actividad y/o un rastreador de actividad física). El dispositivo o dispositivos de usuario también pueden incluir un micrófono que puede utilizarse para detectar si el usuario está hablando, por teléfono, hablando con otra persona cercana, etc. El concentrador de sensor puede incluir algunos o todos los diversos dispositivos de usuario que pueden capturar información relacionada con las acciones o la actividad del usuario (p. ej., incluyendo una interfaz de entrada/salida (E/S) de los dispositivos de usuario que puede capturar la actividad del teclado/ratón/táctil). El concentrador de sensor puede obtener información directamente de los dispositivos de captura de los dispositivos de usuario (p. ej., de forma alámbrica o inalámbrica) o el concentrador de sensor puede integrar información de los dispositivos desde un servidor

o un servicio (p. ej., la información puede cargarse desde un rastreador de actividad física a un servicio en la nube, que el concentrador de sensor puede descargar).

Ejemplos de motor de enfoque

Volviendo ahora a la Figura 6C, un motor de enfoque 614 puede obtener información del motor de detección y/o del motor de movimiento y otras fuentes para proporcionar información de enfoque. La información de enfoque incluye un punto focal, el área de enfoque, la posición del ojo, el movimiento del ojo, el tamaño de la pupila, la dilatación de la pupila, la profundidad de enfoque (DOF), un punto focal de contenido, un objeto de enfoque de contenido, un área de enfoque de contenido etc. La información de enfoque también puede incluir información de enfoque anterior, información de enfoque futuro determinada y/o información de enfoque prevista (p. ej., un punto focal previsto, un área de enfoque prevista, una posición ocular prevista, movimiento ocular previsto, tamaño de pupila previsto, dilatación pupilar prevista, DOF prevista, punto focal de contenido futuro determinado, objeto de enfoque de contenido futuro determinado, área de enfoque de contenido futuro determinada, punto focal de contenido previsto, objeto de enfoque de contenido previsto, área de enfoque de contenido prevista, etc.).

En un alto nivel, algunas realizaciones pueden usar información de enfoque para ajustar los parámetros de rendimiento del sistema de gráficos basándose en 1) dónde se supone que mira el usuario, 2) dónde se determina que mira el usuario, 3) dónde una aplicación desea que mire el usuario, y/o 4) dónde se predice que el usuario mirará en el futuro. Algunas señales de enfoque pueden ser más intensas en la región donde mira el usuario. Si el usuario está mirando al frente, es posible que vea las cosas con un enfoque nítido. Con escenas u objetos hacia la periferia, el usuario puede notar movimiento, pero no detalles en un enfoque nítido.

Por ejemplo, la información de enfoque puede ser estática y/o basada en suposiciones (p. ej., se supone que el usuario está mirando el centro de la pantalla con la posición fija del ojo, DOF, etc.), si la información detectada o la capacidad de procesamiento del sistema de gráficos son limitadas (p. ej., una pantalla montada en la cabeza (HMD) adjunta o un anfitrión no puede proporcionar o hacer uso de la información). La información de enfoque también puede cambiar dinámicamente basándose en factores tales como la información de movimiento (p. ej., de un casco de realidad virtual (VR)), información de predicción de movimiento, información de contenido (p. ej., movimiento en la escena), etc. Más preferentemente, puede proporcionarse una mejor experiencia de usuario con un amplio conjunto de sensores que incluye el rastreo ocular (p. ej., a veces también denominado rastreo de la mirada) para identificar la región de enfoque y proporcionar la información de enfoque. Algunas realizaciones, por ejemplo, pueden incluir un rastreador ocular o recibir información ocular de un rastreador ocular para rastrear los ojos del usuario. La información ocular puede incluir la posición del ojo, el movimiento del ojo, el tamaño/dilatación de la pupila, la profundidad de enfoque, etc. Un rastreador ocular puede capturar una imagen del ojo del usuario, incluyendo la pupila. El punto focal y/o la DOF del usuario se pueden determinar, inferir y/o estimar basándose en la posición del ojo y la dilatación de la pupila. El usuario puede pasar por un proceso de calibración que puede ayudar al rastreador ocular a proporcionar información más precisa sobre el enfoque y/o DOF.

Cuando un usuario lleva un casco de realidad virtual, por ejemplo, una cámara puede capturar una imagen de una pupila y el sistema puede determinar hacia dónde mira el usuario (p. ej., un área de enfoque, profundidad y/o dirección). La cámara puede capturar información sobre la dilatación de la pupila y el sistema puede inferir dónde está el área de enfoque del usuario basándose en esa información. Por ejemplo, un ojo humano tiene una DOF determinada, de modo que, si la persona está enfocando algo cercano, las cosas más lejanas pueden verse borrosas. La información de enfoque puede incluir un punto focal a una distancia de enfoque X e información de DOF de delta (X), por lo que el área de enfoque puede corresponder a $X \pm \Delta X$ situado alrededor del punto focal del usuario. El tamaño de la DOF puede variar con la distancia X (p. ej., un delta diferente a diferentes distancias de enfoque). Por ejemplo, la DOF del usuario puede calibrarse y puede variar en cada dirección (p. ej., x , y y z) de modo que la función ΔX puede no ser necesariamente esférica.

En algunas realizaciones, la información de enfoque puede incluir información de enfoque basada en contenido. Por ejemplo, en un entorno 3D, VR y/o de realidad aumentada (AR), la información de profundidad y/o distancia se puede proporcionar desde una aplicación (p. ej., dónde se encuentra el usuario en el entorno virtual, dónde están los objetos y/o cómo de lejos están los objetos del usuario, etc.). La información de enfoque basada en contenido también puede incluir un punto, objeto o área en el contenido donde la aplicación desea que el usuario se concentre, como algo más interesante que sucede y la aplicación desea la atención del usuario. La aplicación también puede proporcionar información de enfoque de contenido futuro porque la aplicación puede conocer información de movimiento para el contenido y/o qué objetos/áreas en un fotograma o escena siguiente pueden ser de mayor interés para el usuario (p. ej., un objeto a punto de entrar en la escena desde un borde de la pantalla).

Ejemplos de motor de movimiento

Volviendo ahora a la Figura 6D, un motor de movimiento 616 puede obtener información del motor de detección y/o del motor de enfoque y otras fuentes para proporcionar información de movimiento. La información de movimiento puede incluir, por ejemplo, la posición de la cabeza, la velocidad de la cabeza, la aceleración de la cabeza, la dirección del movimiento de la cabeza, la velocidad del ojo, la aceleración del ojo, la dirección de movimiento del ojo, la posición

del objeto, la velocidad del objeto, la aceleración del objeto, la dirección de movimiento del objeto, etc. La información de movimiento también puede incluir información de movimiento anterior, información de movimiento futuro determinado y/o información de movimiento previsto (p. ej., una velocidad de la cabeza prevista, una aceleración de la cabeza prevista, una posición de la cabeza prevista, una dirección de movimiento de la cabeza prevista, una velocidad del ojo prevista, una aceleración del ojo prevista, una dirección de movimiento del ojo prevista, posición de contenido futuro determinada, velocidad de objeto de contenido futuro determinada, aceleración de objeto de contenido futuro determinada, posición de objeto prevista, velocidad de objeto prevista, aceleración de objeto prevista, etc.).

En un nivel alto, algunas realizaciones pueden usar información de movimiento para ajustar los parámetros de rendimiento del sistema de gráficos basándose en 1) el usuario mueve la cabeza, 2) el usuario mueve los ojos. Según la invención, la información de movimiento se usa para ajustar los parámetros de rendimiento del sistema de gráficos basándose en al menos uno de:

3) el usuario mueve el cuerpo, 4) cuando una aplicación quiere que el usuario gire la cabeza, los ojos y/o el cuerpo, y/o 4) cuando se predice que el usuario girará la cabeza, los ojos y/o el cuerpo en el futuro. Algo de información de movimiento puede determinarse fácilmente a partir de la información detectada. Por ejemplo, la posición de la cabeza, la velocidad, la aceleración, la dirección del movimiento, etc., pueden determinarse a partir de un acelerómetro. La información de movimiento de los ojos puede determinarse rastreando la información de la posición de los ojos a lo largo del tiempo (p. ej., si el rastreador de ojos únicamente proporciona información de la posición de los ojos).

Algo de la información de movimiento puede estar basada en el contenido. En un juego o contenido 3D sobre la marcha, por ejemplo, la aplicación puede saber cómo de rápido y hacia dónde se mueven los objetos. La aplicación puede proporcionar la información al motor de movimiento (p. ej., a través de una llamada de API). La información de movimiento de objeto basada en contenido futura para un fotograma/escena siguiente también se puede alimentar en el motor de movimiento para la toma de decisiones. Algo de la información de movimiento basada en contenido puede determinarse mediante el procesamiento de imágenes o el procesamiento de visión artificial del contenido.

Algunas realizaciones de un sistema de visión artificial, por ejemplo, pueden analizar y/o realizar reconocimiento de características/objetos en imágenes capturadas por una cámara. Por ejemplo, la visión artificial y/o el procesamiento de imágenes pueden identificar y/o reconocer objetos en una escena (p. ej., que un borde pertenece al frente de una silla). El sistema de visión artificial también se puede configurar para realizar reconocimiento facial, rastreo de la mirada, reconocimiento de expresiones faciales y/o reconocimiento de gestos, incluyendo gestos a nivel del cuerpo, gestos a nivel de brazos/piernas, gestos a nivel de manos y/o gestos a nivel de dedos. El sistema de visión artificial puede configurarse para clasificar una acción del usuario. En algunas realizaciones, un sistema de visión artificial configurado adecuadamente puede determinar si el usuario está presente en un ordenador, escribiendo en un teclado, usando el ratón, usando el panel táctil, usando la pantalla táctil, usando un HMD, usando un sistema VR, sentado, parado y/o realizando alguna otra acción o actividad.

Por ejemplo, el motor de movimiento puede recuperar datos de la cámara relacionados con un objeto real en una escena y puede usar esa información para identificar el movimiento y la orientación del objeto real. El motor de movimiento puede recuperar información de latencia del procesador de gráficos. A continuación, el motor de movimiento puede predecir la siguiente orientación del fotograma de ese objeto real. La cantidad de latencia puede basarse en uno o más del tiempo para representar una escena, el número de objetos virtuales en la escena y la complejidad de la escena, etc. Por ejemplo, un motor de detección puede incluir una o más cámaras para capturar una escena real. Por ejemplo, la una o más cámaras pueden incluir una o más cámaras 2D, cámaras 3D, cámaras de profundidad, cámaras de alta velocidad u otros dispositivos de captura de imágenes. La escena real puede incluir un objeto que se mueve en la escena. Las cámaras pueden estar acopladas a un procesador de imágenes para procesar datos de las cámaras para identificar objetos en la escena (p. ej., incluyendo el objeto en movimiento) y para identificar el movimiento del objeto (p. ej., incluyendo información de orientación). El motor de movimiento puede determinar la información de movimiento prevista basándose en el rastreo del movimiento del objeto y predecir una posición futura del objeto basándose en una latencia medida o estimada (p. ej., desde el momento de la captura hasta el momento de la representación). De acuerdo con algunas realizaciones, diversas técnicas de rastreo de movimiento y/o predicción de movimiento pueden aumentarse con flujo óptico y otras técnicas de estimación de movimiento real para determinar la siguiente ubicación de los objetos reales. Por ejemplo, algunas realizaciones pueden usar filtrado común ampliado y/o procesamiento de perspectiva (p. ej., de aplicaciones de conducción autónoma) para predecir el movimiento de objetos.

Ejemplos de superposición de motores

En referencia ahora a la Figura 6E, los expertos en la materia apreciarán que los aspectos de diversos motores descritos en el presente documento pueden superponerse con otros motores y que porciones de cada motor pueden implementarse o distribuirse a través de diversas porciones de un sistema de procesamiento electrónico. Por ejemplo, el motor de enfoque puede usar información de movimiento para proporcionar un área de enfoque futura prevista y el motor de movimiento puede usar información de enfoque para predecir un movimiento futuro. La información de movimiento del ojo puede provenir directamente del motor de detección, puede determinarse/predecirse mediante el motor de enfoque y/o puede determinarse/predecirse mediante el motor de movimiento.

La Figura 6E ilustra un ejemplo de detección de mirada. Se puede usar una pantalla 650 para mostrar vídeo u otro contenido visual. Pueden estar presentes uno o más rastreadores oculares 652, ya sea integrales con la pantalla 650 como se muestra o como dispositivos periféricos. Los rastreadores 652 pueden comprender microproyectores de infrarrojo cercano (IR) y sensores/cámaras ópticas. El rastreador 652 puede crear patrones de reflexión 653 en los ojos de un espectador 654. Se puede usar procesamiento de imágenes para identificar los patrones y características de los ojos para calcular la posición de los ojos y el punto de mirada 660 en tiempo real. Si bien este ejemplo ilustra una pantalla plana 650, los rastreadores de mirada también suelen incorporarse en pantallas montadas en la cabeza (HMD), generalmente usadas como gafas. Los ejemplos del presente documento deben considerarse ilustrativos y no limitativos en términos de implementaciones específicas.

Composición en el panel para reducción de ancho de banda

Volviendo ahora a las Figuras 7A y 7B, se muestra un diagrama de bloques de ejemplo de un sistema para reducir el ancho de banda y así reducir la energía a una pantalla 704. Un problema en los sistemas actuales es que muy a menudo una región pequeña dentro de una pantalla más grande tiene requisitos de mayor resolución y fidelidad. Esto puede basarse en los resultados del motor de enfoque 614 (Figura 6C). La Figura 7B muestra una pantalla 704 con la pantalla completa etiquetada como Plano 1 706 y la más pequeña o "ventana de interés" como Plano 2 708. Los datos enviados a la pantalla generalmente tienen una resolución y fidelidad fijas, lo que implica que todos los datos para la pantalla final 704 tienen calidad suficiente para cumplir con los requisitos del Plano 2 708.

La Figura 7A muestra un sistema en chip (SOC) 700, que puede contener todo lo que pueda ser necesario para transmitir vídeo a los paneles de memoria intermedia de visualización 702₁ a 702_N. El SOC 700 puede contener una o más CPU/GPU de uno o varios núcleos, memoria del sistema, canalización de gráficos, canalización de medios, etc., todo lo cual se analiza con mayor detalle anteriormente y a continuación.

Los paneles 702₁ a 702_N pueden comprender parte de la propia pantalla 704. En algunas realizaciones, puede haber solo un plano 702₁. Además, los paneles pueden comprender almacenamiento para un plano de fondo 706 y un plano de interés 708. Según una realización, el SOC 700 puede procesar la transmisión de vídeo al plano de fondo 706 y al plano de interés 708 de manera diferente. Es decir, el SOC 700 podrá desacoplar los esquemas de compresión. El plano de fondo 706 puede comprender una compresión con pérdidas, lo que da como resultado un ancho de banda mucho menor. El plano de interés 708, que normalmente es más pequeño, puede comprender una compresión de calidad de medios tal que los efectos de cualquier pérdida de datos sean imperceptibles para el ojo humano en vídeos y fotografías. Este enfoque deja entonces que sea la pantalla 704 que tiene un circuito compositor 705 la que componga el plano de fondo 706 y el plano de interés 708 juntos. El SOC 700 puede enviar además una coordenada x-y para el plano de interés 708 para indicar a la pantalla 704 dónde colocar el plano de interés, así como un comando para hacer que el plano de interés 708 sea opaco (llevado al frente, bloqueando el plano de fondo).

Composición foveada en motor de visualización

Haciendo referencia ahora a la Figura 8, se muestra un diagrama de bloques de ejemplo que ilustra una forma en que el SOC 700 puede desacoplar los esquemas de compresión para el plano de fondo 706 y el plano de interés 708. La representación foveada produce superficies planas separadas para el campo de visión cubierto por la fovea del usuario (por lo que requiere mayor densidad y calidad de píxeles) y para el fondo (menor densidad y calidad de píxeles). La pantalla compone los planos juntos. Se ahorra energía y ancho de banda al tener menor densidad y calidad en la región de fondo.

Aquí, el plano de fondo 706 puede comprender una resolución menor, tal como cuatro o dieciséis veces menor que el plano de interés. Por ejemplo, la pantalla puede tener una resolución de 500 píxeles por pulgada (PPI) y el plano de interés se representaría y transmitiría en esta resolución, mientras que el plano de fondo 706 podría representarse y transmitirse a 250 o 125 PPI, ahorrando tanto ancho de banda de transmisión como de renderizado. La coordenada x-y que informa a la pantalla 704 dónde colocar el plano de interés puede controlarse mediante un rastreador de mirada discutido previamente con referencia a la Figura 6B. Por tanto, cuando un espectador mira un vídeo, la única parte que se procesa y envía a la pantalla 704 con mayor calidad es la parte de la pantalla donde el espectador mira cualquier momento dado en el tiempo. Una combinación alfa se puede realizar en los bordes del plano de interés 708 para realizar una transición suave de los bordes. Se puede realizar un muestreo ascendente mediante interpolación en el plano de fondo durante el proceso compuesto para evitar artefactos que distraigan.

Transmisión de vídeo dual para una representación foveada simplificada

Haciendo referencia ahora a la Figura 9A, se muestra un diagrama de bloques de ejemplo donde un cliente de pantalla montada en la cabeza (HMD) 900 está conectado a un cliente 902 con un cable físico 904. El cliente 902 puede comprender un SOC que puede contener una o más CPU/GPU de uno o varios núcleos, memoria del sistema, canalización de gráficos, canalización de medios, etc., como se analizó anteriormente para enviar vídeo de alta resolución a la HMD 900. La HMD 900 puede ser un visor de realidad virtual como se analizó anteriormente. La HMD 900 ilustrada está equipada con un rastreador de mirada 906, como se analizó anteriormente, que identifica qué parte

de la imagen está mirando el espectador en un momento dado. Esta información de mirada puede ser devuelta al cliente 902. Además, la HMD 900 retroalimenta información relacionada con el movimiento y la orientación de la cabeza, ya que estas entradas también pueden afectar hacia dónde mira el espectador actualmente o ayudar a predecir hacia dónde mirará a continuación.

El cable físico 904 que une la HMD 900 y el cliente 902 puede usarse debido a consideraciones de ancho de banda con todo el vídeo de alta resolución enviado a la HMD 900 y toda la retroalimentación de mirada y movimiento enviada al cliente 902.

La resolución del ojo de un espectador depende de en qué parte del campo de visión esté mirando. A unos pocos grados se pueden percibir muy bien los pequeños detalles. Fuera de eso, sin embargo, la resolución ocular de unos pocos grados cae rápidamente y las cosas tienden a no verse tan nítidas como en el centro. La representación foveada se refiere a la resolución de la imagen, o cantidad de detalle, que varía en la imagen según uno o más puntos de fijación. El rastreador de mirada 906 puede identificar un punto de fijación e indica dónde debe estar la región de mayor resolución de la imagen que corresponde al centro de la retina del ojo, la fovea.

Haciendo referencia a la Figura 9B, se muestra un diagrama de bloques de ejemplo según una realización para eliminar el cable 904 y pasar a ser inalámbrico. Como antes, se podrán enviar dos planos. Se puede enviar de forma inalámbrica un plano de fondo 920 con una vista completa en una resolución más baja y discutido anteriormente, y un plano de interés 922 correspondiente a la región de mirada actual se puede enviar con una resolución más alta.

Esta pantalla 900 de ejemplo de la Figura 9A muestra un vídeo de un esquiador esquiando por las montañas con toda la pantalla en alta resolución. En la Figura 9B, solo una pequeña porción (es decir, el esquiador 924) de la pantalla 900 se muestra en una resolución más alta y el plano de fondo, que comprende las montañas y los árboles que constituyen la mayor parte del vídeo, se muestra en una resolución más baja. Como tal, el ancho de banda entre el cliente 902 y la HMD 900 puede reducirse en gran medida, permitiendo una conexión inalámbrica.

Esquema de codificación eficiente de imagen general

Volviendo ahora a la Figura 10A, se muestra una imagen fija 1000 del horizonte de una ciudad. La imagen fija 1000 puede ser una imagen digital extremadamente grande (p. ej., "imagen grande" de veinte gigapíxeles) que proporciona una vista panorámica de una escena relativamente grande. Más particularmente, la imagen fija 1000 podría unirse a partir de múltiples imágenes de alta resolución (p. ej., de origen público) que han sido capturadas de la misma área general. En consecuencia, la imagen fija 1000 puede ser ancha en términos de distancia física y densa en términos de píxeles. La escena que se muestra en 1000 es una vista amplia que muestra la ciudad.

Se pueden distinguir edificios altos, ventanas, carreteras y puentes. Por supuesto, esto es en blanco y negro, pero en realidad la escena sería a todo color en alta resolución. Normalmente, si uno hiciera zoom en una imagen grande, esta se pixelaría y los objetos eventualmente serían imperceptibles. Sin embargo, según realizaciones esta imagen está codificada de tal manera que no se produce pixelación. La escena es extremadamente detallada. Si un usuario hiciera zoom en un nivel, el usuario podría ver un edificio en particular, contar las ventanas y leer un letrero publicado en el edificio. Si un usuario hiciera zoom en un segundo nivel, es posible que el usuario esté mirando por la ventana y viendo claramente a la gente cenando en un restaurante. Si el usuario hiciera zoom en un tercer nivel, podría ver a los clientes comiendo y a uno con una mancha de comida en la camisa. En consecuencia, puede estar implicada una gran cantidad de datos.

Los fotogramas I suelen estar asociados con la compresión de vídeo. A modo de resumen rápido, en la compresión de vídeo existen fotogramas I, fotogramas B y fotogramas P. Un fotograma I es una "imagen intracodificada", es decir, una imagen completamente especificada. Los fotogramas P y B contienen solo una parte de la información de la imagen y, por lo tanto, usan menos espacio de almacenamiento que un fotograma I. Como resultado, los fotogramas P y B generalmente pueden mejorar las tasas de compresión de vídeo.

Según una realización, y como se muestra en la Figura 10B, la imagen fija 1000 puede subdividirse en una pluralidad de "cortes" de imagen (p. ej., corte I00, corte I01, corte I11, corte I12, etc.) y almacenarse en una infraestructura informática en la nube (p. ej., servidor) para su recuperación/descarga por uno o más dispositivos cliente (p. ej., dispositivos fuente, no mostrados). Cada corte puede incluir metadatos que describen cómo el corte se mapea a la imagen fija 1000 y cómo se realizará el zoom en el corte.

Debido a que el ancho de banda de comunicación puede ser limitado, la recuperación de la imagen fija 1000 completa mediante un dispositivo cliente determinado puede llevar mucho tiempo o ser imposible. Además, se puede aprovechar una interfaz existente para transportar los datos de manera eficiente. En consecuencia, la solución ilustrada determina una primera vista en perspectiva (p. ej., inicial) de la imagen fija 1000 y recupera solo un subconjunto 1002 de la pluralidad de cortes de imagen correspondientes a la primera vista en perspectiva, por ejemplo, el subconjunto 1002 puede comprender I11, I12, I21 y I22. Debido a que el subconjunto 1002 puede recuperarse a través de un enlace de transporte sin consumir una cantidad sustancial de ancho de banda de transmisión, el subconjunto puede decodificarse

y representarse en el otro extremo. Por lo tanto, la solución ilustrada logra una visualización de contenido adaptable de una manera que reduce el consumo de ancho de banda, disminuye el consumo de energía y mejora el rendimiento.

Volviendo ahora a la Figura 10C, se muestra un diagrama de flujo de ejemplo según una realización. En el bloque 1050 una imagen fija grande y muy detallada se corta en una pluralidad de cortes pequeños. En el bloque 1052, a cada corte se le asigna una coordenada en relación con la imagen general. La coordenada se adjunta como metadatos al corte. En el bloque 1054, cada corte de la imagen fija se codifica como un fotograma I usando técnicas de codificación de vídeo. En el bloque 1056, los fotogramas I o subconjuntos de fotogramas I pueden transmitirse a un dispositivo cliente solicitante como si fuera un vídeo. En el bloque 1058, los metadatos pueden usarse para volver a ensamblar la imagen fija o una parte de la imagen comprendida en el subconjunto de fotogramas I.

El flujo ilustrado puede implementarse como uno o más módulos en un conjunto de instrucciones lógicas almacenadas en un medio de almacenamiento no transitorio legible por máquina u ordenador tal como memoria de acceso aleatorio (RAM), memoria de solo lectura (ROM), ROM programable (PROM), firmware, memoria flash, etc., en lógica configurable como, por ejemplo, matrices lógicas programables (PLA), matrices de puertas programables en campo (FPGA), dispositivos lógicos programables complejos (CPLD), en lógica de hardware de funcionalidad fija que usa tecnología de circuito tal como, por ejemplo, circuito integrado de aplicación específica (ASIC), semiconductor complementario de óxido metálico (CMOS) o tecnología de lógica transistor-transistor (TTL), o cualquier combinación de las mismas.

Tecnología de visualización

Volviendo ahora a la Figura 11, se muestra un sistema informático 1100 de rendimiento mejorado. En el ejemplo ilustrado, un procesador 1110 está acoplado a una pantalla 1120. El procesador 1110 generalmente puede generar imágenes para mostrarlas en un panel LCD 1150 de la pantalla 1120. En un ejemplo, el procesador 1110 incluye una interfaz de comunicación tal como, por ejemplo, una matriz de gráficos de vídeo (VGA), una interfaz DisplayPort (DP), una interfaz DisplayPort integrada (eDP), una interfaz multimedia de alta definición (HDMI), una interfaz visual digital (DVI), etc. El procesador 1110 puede ser un procesador de gráficos (p. ej., unidad de procesamiento de gráficos/GPU) que procesa datos de gráficos y genera las imágenes (p. ej., fotogramas de vídeo, imágenes fijas) mostradas en el panel LCD 1150. Además, el procesador 1110 puede incluir una o más canalizaciones de procesamiento de imágenes que generan datos de píxeles. Las canalizaciones de procesamiento de imágenes pueden cumplir con la arquitectura OPENGGL u otra arquitectura adecuada. Además, el procesador 1110 puede estar conectado a un procesador de anfitrión (p. ej., unidad central de procesamiento/CPU), en donde el procesador de anfitrión ejecuta uno o más controladores de dispositivo que controlan y/o interactúan con el procesador 1110.

La pantalla 1120 ilustrada incluye un controlador de sincronización (TCON) 1130, que puede abordar individualmente diferentes píxeles en el panel LCD 1150 y actualizar cada píxel individual en el panel LCD 1150 por ciclo de actualización. A este respecto, el panel LCD 1150 puede incluir una pluralidad de elementos de cristal líquido tales como, por ejemplo, un cristal líquido y un filtro de color integrado. Cada píxel del panel LCD 1150 puede incluir un trío de elementos de cristal líquido con filtros de color rojo, verde y azul, respectivamente. El panel LCD 1150 puede disponer los píxeles en una matriz bidimensional (2D) que se controla mediante controladores de fila 1152 y controladores de columna 1154 para actualizar la imagen que se muestra en el panel LCD 1150. Por tanto, el TCON 1130 puede controlar los controladores de fila 1152 y los controladores de columna 1154 para abordar píxeles específicos del panel LCD 1150. El TCON 1130 también puede ajustar la tensión proporcionada a los elementos de cristal líquido en el píxel para cambiar la intensidad de la luz que pasa a través de cada uno de los tres elementos de cristal líquido y, por lo tanto, cambiar el color del píxel que se muestra en la superficie del panel LCD 1150.

Una retroiluminación 1160 puede incluir una pluralidad de elementos emisores de luz tales como, por ejemplo, diodos emisores de luz (LED), que están dispuestos en un borde del panel LCD 1150. En consecuencia, la luz generada por los LED puede dispersarse a través del panel LCD 1150 mediante un difusor (no mostrado). En otro ejemplo, los LED están dispuestos en una matriz 2D directamente detrás del panel LCD 1150 en una configuración a veces denominada retroiluminación directa porque cada LED dispersa la luz a través de uno o más píxeles correspondientes del panel LCD 1150 colocado frente al LED. Los elementos emisores de luz también pueden incluir lámparas fluorescentes compactas (CFL) dispuestas a lo largo de uno o más bordes del panel LCD 1150. Para eliminar múltiples bordes, la combinación de bordes puede alterarse para lograr una iluminación selectiva de una región, en donde se usa menos del conjunto total de elementos de iluminación con menos potencia.

Los elementos emisores de luz también pueden incluir una o más láminas de material electroluminiscente colocadas detrás del panel LCD 1150. En tal caso, la luz de la superficie de la lámina puede dispersarse a través de los píxeles del panel LCD 1150. Además, la lámina puede dividirse en una pluralidad de regiones tales como, por ejemplo, cuadrantes. En un ejemplo, cada región se controla individualmente para iluminar solo una parte del panel LCD 1150. También se pueden usar otras soluciones de retroiluminación.

La pantalla 1120 ilustrada también incluye un controlador de retroiluminación (BLC) 1140 que proporciona una tensión a los elementos emisores de luz de la retroiluminación 1160. Por ejemplo, el BLC 1140 puede incluir un controlador de modulación de ancho de pulso (PWM) (no mostrado) para generar una señal PWM que activa al menos una porción

de los elementos emisores de luz de la retroiluminación 1160. El ciclo de trabajo y la frecuencia de la señal PWM pueden hacer que la luz generada por los elementos emisores de luz se atenúe. Por ejemplo, un ciclo de trabajo del 100 % puede corresponder a que los elementos emisores de luz estén completamente encendidos y un ciclo de trabajo del 0 % puede corresponder a que los elementos emisores de luz estén completamente apagados. Por tanto, los ciclos de trabajo intermedios (p. ej., 25 %, 50 %) normalmente hacen que los elementos emisores de luz se enciendan durante una parte de un período de ciclo que es proporcional al porcentaje del ciclo de trabajo. El período de ciclo puede ser lo suficientemente rápido como para que el ojo humano no note el parpadeo de los elementos emisores de luz. Además, el efecto para el usuario puede ser que el nivel de luz emitida por la retroiluminación 1160 sea menor que si la retroiluminación 1160 estuviera completamente activada. El BLC 1140 puede estar separado o incorporado al TCON 1130.

Alternativamente, se puede usar un sistema de visualización emisivo donde el panel LCD 1150 sería reemplazado por un panel de visualización emisivo (p. ej., diodo emisor de luz orgánico/OLED), se omitiría la retroiluminación 1160 y los controladores de fila y columna 1152 y 1154, respectivamente, se pueden usar para modular directamente el color y el brillo de los píxeles.

Resolución de visualización basada en distancia

La Figura 12A muestra un escenario en el que un usuario 1218 interactúa con un dispositivo de procesamiento de datos 1200 que contiene una unidad de visualización 1228. El dispositivo de procesamiento de visualización 1200 puede incluir, por ejemplo, un ordenador portátil, un ordenador de sobremesa, una tableta, una tableta convertible, un dispositivo de Internet móvil (MID), un asistente digital personal (PDA), un dispositivo portátil (p. ej., pantalla montada en la cabeza/HMD), un reproductor multimedia, etc., o cualquier combinación de los mismos. El dispositivo de procesamiento de datos 1200 ilustrado incluye un procesador 1224 (p. ej., controlador integrado, microcontrolador, procesador de anfitrión, procesador de gráficos) acoplado a una memoria 1222, que puede incluir ubicaciones de almacenamiento que son direccionables a través del procesador 1224. Como se explicará con mayor detalle, un sensor de distancia 1210 puede permitir una resolución de visualización basada en la distancia con respecto a las unidades de visualización 1228.

La memoria 1222 ilustrada incluye datos de visualización 1226 que se representarán en la unidad de visualización 1228. En un ejemplo, el procesador 1224 realiza la conversión de datos en los datos de visualización 1226 antes de presentar los datos de visualización 1226 en la unidad de visualización 1228. Un motor de postprocesamiento 1214 puede ejecutarse en el procesador 1224 para recibir los datos de visualización 1226 y una salida del sensor de distancia 1210. El motor de postprocesamiento 1214 puede modificar los datos de visualización 1226 para mejorar la legibilidad del contenido de la pantalla en la unidad de visualización 1228, reducir el consumo de energía en el dispositivo de procesamiento de datos 1200, etc., o cualquier combinación de los mismos.

La memoria ilustrada 1222 almacena una configuración de resolución de visualización 1216, además de un sistema operativo 1212 y una aplicación 1220. La configuración de resolución de visualización 1216 puede especificar un número de píxeles de los datos de visualización 1226 que se presentarán en la unidad de visualización 1228 a lo largo de una dimensión de longitud y una dimensión de ancho. Si los datos de visualización 1226 generados por la aplicación 1220 son incompatibles con el formato de la unidad de visualización 1228, el procesador 1224 puede configurar la escala de los datos de visualización 1226 para que coincida con el formato de las unidades de visualización 1228. A este respecto, la configuración de resolución de visualización 1216 puede asociarse y/o incorporarse a datos de configuración que definen otras configuraciones para la unidad de visualización 1228. Además, la configuración de resolución de visualización 1216 puede definirse en términos de unidad de distancia o área (p. ej., píxeles por pulgada/PPI), u otro parámetro adecuado.

La aplicación 1220 puede generar una interfaz de usuario, en donde el usuario 1218 puede interactuar con la interfaz de usuario para seleccionar la configuración de resolución de visualización 1216 de una o más opciones proporcionadas a través de la interfaz de usuario, ingresar la configuración de resolución de visualización 1216 como un valor solicitado, y así sucesivamente. Por tanto, se puede cambiar el tamaño de los datos de visualización 1226 para que encajen en la configuración de resolución de visualización 1216 antes de ser representados en la unidad de visualización 1228.

El sensor de distancia 1210 puede rastrear la distancia entre el usuario 1218 y la unidad de visualización 1228, en donde la detección de distancia puede activarse a través de un botón físico asociado con el dispositivo de procesamiento de datos 1200/unidad de visualización 1228, a través de la interfaz de usuario proporcionada por la aplicación 1220 y /o carga del sistema operativo 1220, y así sucesivamente. Por ejemplo, durante un arranque del dispositivo de procesamiento de datos 1200, el sistema operativo 1212 puede realizar un proceso automático para activar la detección de distancia en segundo plano o en primer plano. La detección de distancias puede realizarse de forma periódica o continua.

La Figura 12B muestra un ejemplo de un escenario de detección de distancia. En el ejemplo ilustrado, el sensor de distancia 1210 usa un transceptor 1208 para emitir un haz electromagnético 1202 en la dirección del usuario 1218. Por tanto, el transceptor 1202 podría colocarse en una superficie orientada hacia el frente del dispositivo de

procesamiento de datos 1200 (Figura 12A). El haz electromagnético 1202 puede impactar al usuario 1218 y ser reflejado/dispersado desde el usuario 1218 como un haz electromagnético de retorno 1204. El haz electromagnético de retorno 1204 puede ser analizado, por ejemplo, por el procesador 1224 (Figura 12A) y/o el motor de postprocesamiento 1214 (Figura 12A) para determinar la distancia 1206 entre el usuario 1218 y la unidad de visualización 1228 (Figura 12A). La distancia 1206 se puede usar para ajustar la configuración de resolución de visualización 1216.

Capas de visualización

Volviendo ahora a la Figura 13, se muestra un sistema de visualización 1300 en el que se usan capas de visualización en cascada 1361, 1362 y 1363 para lograr una superresolución espacial/temporal en un conjunto de visualización 1360. En el ejemplo ilustrado, un procesador 1310 proporciona datos gráficos originales 1334 (p. ej., fotogramas de vídeo, imágenes fijas) al sistema 1300 a través de un bus 1320. Un programa de visualización en cascada 1331 puede almacenarse en una memoria 1330, en donde el programa de visualización en cascada 1331 puede ser parte de un controlador de visualización asociado con el conjunto de visualización 1360. La memoria ilustrada 1330 también incluye los datos de gráficos originales 1334 y los datos de gráficos factorizados 1335. En un ejemplo, el programa de visualización en cascada 1331 incluye un componente de factorización temporal 1332 y un componente de factorización espacial 1333. El componente de factorización temporal 1332 puede realizar un cálculo de factorización temporal y el componente de factorización espacial puede realizar un cálculo de factorización espacial. El programa de visualización en cascada 1331 puede derivar los datos gráficos factorizados 1335 para su presentación en cada capa de visualización 1361, 1362 y 1363 basándose en las configuraciones del usuario y los datos gráficos originales 1334.

El conjunto de visualización 1360 puede implementarse como una LCD (pantalla de cristal líquido) usada, por ejemplo, en una aplicación de pantalla montada en la cabeza (HMD). Más particularmente, el conjunto de visualización 1360 puede incluir una pila de placas de interfaz de paneles LCD, un accesorio de lente, etc. Cada panel puede funcionar con una resolución nativa de, por ejemplo, 1280x800 píxeles y con una frecuencia de actualización de 60 Hz. Se pueden usar otras resoluciones nativas, frecuencias de actualización, tecnología de panel de visualización y/o configuraciones de capas.

Múltiples unidades de visualización

La Figura 14 muestra un sistema de visualización de gráficos 1400 que incluye un conjunto de unidades de visualización 1430 (1430a-1430n) que generalmente se pueden usar para generar una presentación de pantalla ancha (p. ej., panorámica) 1440 que incluye contenido coordinado en una forma topológica cohesiva y estructurada. En el ejemplo ilustrado, un dispositivo de procesamiento de datos 1418 incluye un procesador 1415 que aplica una función lógica 1424 a los datos de perfil de hardware 1402 recibidos del conjunto de unidades de visualización 1430 a través de una red 1420. La aplicación de la función lógica 1424 a los datos del perfil de hardware 1402 puede crear un conjunto de configuraciones de topología automática 1406 cuando no se encuentra una coincidencia de los datos de perfil de hardware con un conjunto de configuraciones en una tabla de búsqueda de perfil de hardware 1412. El conjunto ilustrado de configuraciones de topología automática 1406 se transmite desde el dispositivo de procesamiento de visualización 1418 a las unidades de visualización 1430 a través de la red 1420.

El procesador 1415 puede realizar y ejecutar la función lógica 1424 al recibir la función lógica 1424 desde un controlador de visualización 1410. A este respecto, el controlador de visualización 1410 puede incluir un módulo de topología automática 1408 que configura y estructura automáticamente las topologías de las unidades de visualización 1432 para crear la presentación 1440. En un ejemplo, el controlador de visualización 1410 es un conjunto de instrucciones que, cuando las ejecuta el procesador 1415, hacen que el dispositivo de procesamiento de datos 1418 se comuniquen con las unidades de visualización 1430, tarjetas de vídeo, etc., y realice operaciones automáticas de generación de topología.

El dispositivo de procesamiento de datos 1418 puede incluir, por ejemplo, un servidor, un ordenador de escritorio, un ordenador portátil, una tableta, una tableta convertible, MID, PDA, un dispositivo portátil, un reproductor multimedia, etc. Por tanto, el dispositivo de procesamiento de visualización 1418 puede incluir un módulo de control de hardware 1416, un dispositivo de almacenamiento 1414, memoria de acceso aleatorio (RAM, no mostrada), tarjetas controladoras que incluyen una o más tarjetas controladoras de vídeo, etc. En un ejemplo, las unidades de visualización 1430 son pantallas de panel plano (p. ej., cristal líquido, matriz activa, plasma, etc.), HMD, dispositivos de proyección de vídeo, etc., que se coordinan entre sí para producir la presentación 1440. Además, la presentación 1440 puede generarse basándose en un archivo multimedia almacenado en el dispositivo de almacenamiento 1414, en donde el archivo multimedia podría incluir, por ejemplo, una película, un videoclip, una animación, un anuncio, etc., o cualquier combinación de los mismos.

El término "topología" puede considerarse el número, escala, forma y/u otro parámetro de configuración de una primera unidad de visualización 1430a, una segunda unidad de visualización 1430b, una tercera unidad de visualización 1430n, etc. En consecuencia, la topología de las unidades de visualización 1430 puede permitir que la presentación 1440 se presente visualmente en concierto de manera que las secciones individuales de la presentación 1440 sean

proporcionales y compatibles con las dimensiones originales y el alcance de los medios que se reproducen a través de las unidades de visualización 1430. Por tanto, la topología puede constituir relaciones espaciales y/o propiedades geométricas que no se ven afectadas por el cambio continuo de forma o tamaño del contenido representado en la presentación 1440. En un ejemplo, el módulo de topología automática 1408 incluye un módulo de temporización 1426, un módulo de control 1428, un módulo de monitorización de señales 1432 y un módulo de visualización de señales 1434. El módulo de sincronización 1426 puede designar una unidad de visualización particular en el conjunto de unidades de visualización 1430 como una unidad de visualización de muestra. En tal caso, el módulo de sincronización 1426 puede designar las unidades de visualización restantes 1430 como unidades de visualización adicionales. En un ejemplo, el módulo de sincronización 1426 establece automáticamente un factor de configuración para que sea compatible con los datos de perfil de hardware 1402, en donde la presentación 1440 se inicia automáticamente mediante una secuencia de señales gráficas 1422.

En un ejemplo, el módulo de control 1428 modifica el conjunto de configuraciones de topología automática 1406. Además, el módulo de monitorización de señales 1432 puede monitorear automáticamente la secuencia de señales gráficas 1422 y activar el dispositivo de almacenamiento 1414 para asociar el conjunto de configuraciones de topología automática 1406 con la tabla de búsqueda de perfil de hardware 1412. Además, el módulo de monitorización de señales 1432 puede detectar automáticamente cambios en el conjunto de unidades de visualización 1430 de acuerdo con un conjunto de criterios de cambio y generar automáticamente un nuevo perfil de topología correspondiente al cambio en el conjunto de unidades de visualización 1430. Por tanto, el nuevo perfil de topología puede aplicarse al conjunto de unidades de visualización 1430. El módulo de monitorización de señales 1432 también puede activar el módulo de visualización de señales 1434 para volver a aplicar el conjunto de configuraciones de disculpa automática 1406 si la secuencia de señales gráficas 1422 no cumple con un conjunto de criterios. Si los datos de perfil de hardware 1402 no soportan la visualización automática de topología de la secuencia de señales gráficas 1422, el dispositivo de procesamiento de datos 1418 puede informar de un error y registrar el error en un registro de errores 1413.

Suministro de medios asistido por la nube

Volviendo ahora a la Figura 15, un sistema de juego en la nube 1500 incluye un cliente 1540 que está acoplado a un servidor 1520 a través de una red 1510. El cliente 1540 generalmente puede ser un consumidor de contenido de gráficos (p. ej., juegos, realidad virtual/VR, realidad aumentada/AR) que se aloja, procesa y representa en el servidor 1520. El servidor 1520 ilustrado, que puede ser escalable, tiene la capacidad de proporcionar el contenido de gráficos a múltiples clientes simultáneamente (p. ej., aprovechando recursos de procesamiento y representación paralelos y distribuidos). En un ejemplo, la escalabilidad del servidor 1520 está limitada por la capacidad de la red 1510. En consecuencia, puede haber un número umbral de clientes por encima del cual el servicio a todos los clientes se degrada.

En un ejemplo, el servidor 1520 incluye un procesador de gráficos (p. ej., GPU) 1530, un procesador de anfitrión (p. ej., CPU) 1524 y una tarjeta de interfaz de red (NIC) 1522. La NIC 1522 puede recibir una solicitud del cliente 1540 para contenido gráfico. La solicitud del cliente 1540 puede hacer que el contenido de gráficos se recupere de la memoria a través de una aplicación que se ejecuta en el procesador de anfitrión 1524. El procesador de anfitrión 1524 puede llevar a cabo operaciones de alto nivel tales como, por ejemplo, determinar la posición, colisión y movimiento de objetos en una escena determinada. En función de las operaciones de alto nivel, el procesador de anfitrión 1524 puede generar comandos de representación que se combinan con los datos de la escena y se ejecutan mediante el procesador de gráficos 1530. Los comandos de representación pueden hacer que el procesador de gráficos 1530 defina la geometría de la escena, el sombreado, la iluminación, el movimiento, la textura, los parámetros de la cámara, etc., para las escenas que se presentarán a través del cliente 1540.

Más particularmente, el procesador de gráficos 1530 ilustrado incluye un renderizador de gráficos 1532 que ejecuta procedimientos de renderizado de acuerdo con los comandos de renderizado generados por el procesador de anfitrión 1524. La salida del procesador de gráficos 1532 puede ser un flujo de fotogramas de vídeo sin procesar que se proporcionan a un capturador de fotogramas 1534. El capturador de fotogramas 1534 ilustrado está acoplado a un codificador 1536, que puede comprimir/formatear el flujo de vídeo sin procesar para su transmisión a través de la red 1510. El codificador 1536 puede usar una amplia variedad de algoritmos de compresión de vídeo tales como, por ejemplo, el estándar H.264 del Sector de Normalización de Telecomunicaciones de la Unión Internacional de Telecomunicaciones (ITU-T), el estándar de codificación de vídeo avanzada (AVC) MPEG4 de la Organización Internacional para la Normalización/Comisión Electrotécnica Internacional (ISO/IEC), etc.

El cliente 1540 ilustrado, que puede ser un ordenador de sobremesa, un ordenador portátil, una tableta, una tableta convertible, un dispositivo portátil, MID, PDA, reproductor multimedia, etc., incluye una NIC 1542 para recibir el flujo de vídeo transmitido desde el servidor 1520. La NIC 1522 puede incluir la capa física y la base para la capa de software de la interfaz de red en el cliente 1540 para facilitar las comunicaciones a través de la red 1510. El cliente 1540 también puede incluir un decodificador 1544 que emplea el mismo esquema de formateo/compresión del codificador 1536. Por tanto, el flujo de vídeo descomprimido puede proporcionarse desde el decodificador 1544 a un renderizador de vídeo 1546. El renderizador de vídeo 1546 ilustrado está acoplado a una pantalla 1548 que presenta visualmente el contenido gráfico.

Como ya se señaló, el contenido gráfico puede incluir contenido de juegos. A este respecto, el cliente 1540 puede realizar una transmisión interactiva en tiempo real que implica la recopilación de la entrada de usuario desde un dispositivo de entrada 1550 y la entrega de la entrada de usuario al servidor 1520 a través de la red 1510. Este componente interactivo en tiempo real de los juegos en la nube puede plantear desafíos con respecto a la latencia.

Ejemplo de vista general del sistema adicional

La Figura 16 es un diagrama de bloques de un sistema de procesamiento 1600, de acuerdo con una realización. En diversas realizaciones, el sistema 1600 incluye uno o más procesadores 1602 y uno o más procesadores de gráficos 1608, y puede ser un sistema de sobremesa de procesador único, un sistema de estación de trabajo de multiprocesador, o un sistema de servidor que tiene un gran número de procesadores 1602 o núcleos de procesador 1607. En una realización, el sistema 1600 es una plataforma de procesamiento incorporada dentro de un circuito integrado de sistema en un chip (SoC) para su uso en dispositivos móviles, portátiles o integrados.

Una realización del sistema 1600 puede incluir, o estar incorporada dentro de una plataforma de juegos basada en servidor, una consola de juegos, que incluye una consola de juegos y medios, una consola de juegos móvil, una consola de juegos portátil o una consola de juegos en línea. En algunas realizaciones, el sistema 1600 es un teléfono móvil, un teléfono inteligente, un dispositivo informático de tipo tableta o un dispositivo de Internet móvil. El sistema de procesamiento de datos 1600 también puede incluir, acoplarse con o integrarse dentro de un dispositivo ponible, tal como un dispositivo ponible de reloj inteligente, un dispositivo de gafas inteligentes, un dispositivo de realidad aumentada o un dispositivo de realidad virtual. En algunas realizaciones, el sistema de procesamiento de datos 1600 es un dispositivo de televisión o de descodificador de salón que tiene uno o más procesadores 1602 y una interfaz gráfica generada por uno o más procesadores de gráficos 1608.

En algunas realizaciones, cada uno de los uno o más procesadores 1602 incluye uno o más núcleos de procesador 1607 para procesar instrucciones que, cuando se ejecutan, realizan operaciones para software de usuario y sistema. En algunas realizaciones, cada uno de los uno o más núcleos de procesador 1607 está configurado para procesar un conjunto de instrucciones 1609 específico. En algunas realizaciones, el conjunto de instrucciones 1609 puede facilitar el cálculo de conjunto de instrucciones complejo (CISC), el cálculo de conjunto de instrucciones reducido (RISC) o el cálculo mediante una palabra de instrucción muy larga (VLIW). Cada uno de múltiples núcleos de procesador 1607 puede procesar un conjunto de instrucciones diferente 1609, que puede incluir instrucciones para facilitar la emulación de otros conjuntos de instrucciones. El núcleo de procesador 1607 también puede incluir otros dispositivos de procesamiento, tales como un procesador de señales digitales (DSP).

En algunas realizaciones, el procesador 1602 incluye la memoria caché 1604. Dependiendo de la arquitectura, el procesador 1602 puede tener una única caché interna o múltiples niveles de caché interna. En algunas realizaciones, la memoria caché se comparte entre diversos componentes del procesador 1602. En algunas realizaciones, el procesador 1602 también usa una caché externa (p. ej., una caché de nivel 3 (L3) o una caché de último nivel (LLC)) (no mostrada), que puede compartirse entre los núcleos de procesador 1607 usando técnicas de coherencia de caché conocidas. Se incluye adicionalmente, en el procesador 1602, un archivo de registro 1606 que puede incluir diferentes tipos de registros para almacenar diferentes tipos de datos (p. ej., registros de número entero, registros de coma flotante, registros de estado y un registro de puntero de instrucción). Algunos registros pueden ser registros de propósito general, mientras que otros registros pueden ser específicos del diseño del procesador 1602.

En algunas realizaciones, el procesador 1602 está acoplado con un bus de procesador 1610 para transmitir señales de comunicación tales como señales de dirección, de datos o de control entre el procesador 1602 y otros componentes en el sistema 1600. En una realización, el sistema 1600 usa una arquitectura de sistema de 'concentrador' ilustrativa, incluyendo un concentrador de controlador de memoria 1616 y un concentrador de controlador de entrada-salida (E/S) 1630. Un concentrador de controlador de memoria 1616 facilita la comunicación entre un dispositivo de memoria y otros componentes del sistema 1600, mientras que un concentrador de controlador de E/S (ICH) 1630 proporciona conexiones a dispositivos de E/S mediante un bus de E/S local. En una realización, la lógica del concentrador de controlador de memoria 1616 está integrada dentro del procesador.

El dispositivo de memoria 1620 puede ser un dispositivo de memoria de acceso aleatorio dinámica (DRAM), un dispositivo de memoria de acceso aleatorio estática (SRAM), dispositivo de memoria flash, dispositivo de memoria de cambio de fase o algún otro dispositivo de memoria que tiene un rendimiento adecuado para servir como una memoria de proceso. En una realización, el dispositivo de memoria 1620 puede operar como memoria de sistema para el sistema 1600, para almacenar datos 1622 e instrucciones 1621 para su uso cuando el uno o más procesadores 1602 ejecutan una aplicación o proceso. El concentrador de controlador de memoria 1616 también se acopla con un procesador de gráficos externo opcional 1612, que puede comunicarse con el uno o más procesadores de gráficos 1608 en los procesadores 1602 para realizar operaciones de gráficos y de medios.

En algunas realizaciones, el ICH 1630 posibilita que los periféricos se conecten al dispositivo de memoria 1620 y al procesador 1602 mediante un bus de E/S de alta velocidad. Los periféricos de E/S incluyen, pero sin limitación, un controlador de audio 1646, una interfaz de firmware 1628, un transceptor inalámbrico 1626 (p. ej., Wi-Fi, Bluetooth), un dispositivo de almacenamiento de datos 1624 (p. ej., unidad de disco duro, memoria flash, etc.), y un controlador

de E/S heredado 1640 para acoplar dispositivos heredados (p. ej., dispositivos de sistema personal 2 (PS/2)) al sistema. Uno o más controladores de bus serie universal (USB) 1642 conectan dispositivos de entrada, tales como las combinaciones de teclado y ratón 1644. Un controlador de red 1634 puede acoplarse también con el ICH 1630. En algunas realizaciones, un controlador de red de alto rendimiento (no mostrado) se acopla con el bus de procesador 1610. Se apreciará que el sistema 1600 mostrado es ilustrativo y no limitante, ya que pueden usarse otros tipos de sistemas de procesamiento de datos que están configurados de manera diferente. Por ejemplo, el concentrador de controlador de E/S 1630 puede integrarse dentro de los uno o más procesadores 1602, o el concentrador de controlador de memoria 1616 y el concentrador de controlador de E/S 1630 pueden integrarse en un procesador de gráficos externo discreto, tal como el procesador de gráficos externo 1612.

La Figura 17 es un diagrama de bloques de una realización de un procesador 1700 que tiene uno o más núcleos de procesador 1702A-1702N, un controlador de memoria integrado 1714 y un procesador de gráficos integrado 1708. Aquellos elementos de la Figura 17 que tienen los mismos números (o nombres) de referencia que los elementos de cualquier otra figura en el presente documento pueden operar o funcionar de cualquier manera similar a la descrita en cualquier otra parte en el presente documento, pero no se limitan a tal cosa. El procesador 1700 puede incluir núcleos adicionales hasta e incluyendo el núcleo adicional 1702N representado por los recuadros en líneas discontinuas. Cada uno de los núcleos de procesador 1702A-1702N incluye una o más unidades de caché internas 1704A-1704N. En algunas realizaciones, cada núcleo de procesador también tiene acceso a una o más unidades almacenadas en caché compartidas 1706.

Las unidades de caché internas 1704A-1704N y las unidades de caché compartidas 1706 representan una jerarquía de memoria caché dentro del procesador 1700. La jerarquía de memoria caché puede incluir al menos un nivel de caché de instrucciones y de datos dentro de cada núcleo de procesador y uno o más niveles de caché de nivel medio compartida, tal como una caché de Nivel 2 (L2), de Nivel 3 (L3), de Nivel 4 (L4) o de otros niveles, donde el nivel más alto de caché antes de la memoria externa se clasifica como LLC. En algunas realizaciones, la lógica de coherencia de caché mantiene la coherencia entre las diversas unidades de caché 1706 y 1704A-1704N.

En algunas realizaciones, el procesador 1700 también puede incluir un conjunto de una o más unidades de controlador de bus 1716 y un núcleo de agente de sistema 1710. La una o más unidades de controlador de bus 1716 gestionan un conjunto de buses periféricos, tal como uno o más buses de interconexión de componentes periféricos (p. ej., PCI, PCI Express). El núcleo de agente de sistema 1710 proporciona funcionalidad de gestión para los diversos componentes de procesador. En algunas realizaciones, el núcleo de agente de sistema 1710 incluye uno o más controladores de memoria integrados 1714 para gestionar el acceso a diversos dispositivos de memoria externos (no mostrados).

En algunas realizaciones, uno o más de los núcleos de procesador 1702A-1702N incluyen soporte para múltiples hilos simultáneos. En una realización de este tipo, el núcleo de agente de sistema 1710 incluye componentes para coordinar y operar los núcleos 1702A-1702N durante el procesamiento de múltiples hilos. El núcleo de agente de sistema 1710 puede incluir adicionalmente una unidad de control de potencia (PCU), que incluye lógica y componentes para regular el estado de potencia de los núcleos de procesador 1702A-1702N y el procesador de gráficos 1708.

En algunas realizaciones, el procesador 1700 incluye adicionalmente un procesador de gráficos 1708 para ejecutar operaciones de procesamiento de gráficos. En algunas realizaciones, el procesador de gráficos 1708 se acopla con el conjunto de unidades de caché compartidas 1706 y el núcleo de agente de sistema 1710, incluyendo los uno o más controladores de memoria integrados 1714. En algunas realizaciones, un controlador de visualización 1711 está acoplado con el procesador de gráficos 1708 para controlar una salida del procesador de gráficos a una o más pantallas acopladas. En algunas realizaciones, el controlador de visualización 1711 puede ser un módulo separado acoplado con el procesador de gráficos mediante al menos una interconexión, o puede estar integrado dentro del procesador de gráficos 1708 o del núcleo de agente de sistema 1710.

En algunas realizaciones, se usa una unidad de interconexión basada en anillo 1712 para acoplar los componentes internos del procesador 1700. Sin embargo, se puede usar una unidad de interconexión alternativa, tal como una interconexión de punto a punto, una interconexión conmutada u otras técnicas, incluyendo técnicas bien conocidas en la técnica. En algunas realizaciones, el procesador de gráficos 1708 se acopla con la interconexión en anillo 1712 mediante un enlace de E/S 1713.

El enlace de E/S 1713 ilustrativo representa al menos una de múltiples variedades de interconexiones de E/S, que incluyen una interconexión de E/S de paquete, que facilita la comunicación entre diversos componentes de procesador y un módulo de memoria integrado de alto rendimiento 1718, tal como un módulo de eDRAM. En algunas realizaciones, cada uno de los núcleos de procesador 1702-1702N y del procesador de gráficos 1708 usa módulos de memoria integrados 1718 tal como una caché de último nivel compartida.

En algunas realizaciones, los núcleos de procesador 1702A-1702N son núcleos homogéneos que ejecutan la misma arquitectura de conjunto de instrucciones. En otra realización, los núcleos de procesador 1702A-1702N son heterogéneos en términos de arquitectura de conjunto de instrucciones (ISA), donde uno o más de los núcleos de procesador 1702A-N ejecutan un primer conjunto de instrucciones, mientras que al menos uno de los otros núcleos

ejecuta un subconjunto del primer conjunto de instrucciones o un conjunto de instrucciones diferente. En una realización, los núcleos de procesador 1702A-1702N son heterogéneos en términos de microarquitectura, donde uno o más núcleos que tienen un consumo de potencia relativamente más alto se acoplan con uno o más núcleos de potencia que tienen un consumo de potencia más bajo. Adicionalmente, el procesador 1700 puede implementarse en uno o más chips o como un circuito de SoC integrado que tiene los componentes ilustrados, además de otros componentes.

La Figura 18 es un diagrama de bloques de un procesador de gráficos 1800, que puede ser una unidad de procesamiento de gráficos discreta, o puede ser un procesador de gráficos integrado con una pluralidad de núcleos de procesamiento. En algunas realizaciones, el procesador de gráficos se comunica mediante una interfaz de E/S de memoria mapeada a registros en el procesador de gráficos y con comandos colocados en la memoria de procesador. En algunas realizaciones, el procesador de gráficos 1800 incluye una interfaz de memoria 1814 para acceder a memoria. La interfaz de memoria 1814 puede ser una interfaz a una memoria local, una o más cachés internas, una o más cachés externas compartidas y/o a una memoria de sistema.

En algunas realizaciones, el procesador de gráficos 1800 también incluye un controlador de visualización 1802 para controlar los datos de salida de visualización a un dispositivo de visualización 1820. El controlador de visualización 1802 incluye hardware para uno o más planos de superposición para la visualización y composición de múltiples capas de elementos de interfaz de usuario o de vídeo. En algunas realizaciones, el procesador de gráficos 1800 incluye un motor de códec de vídeo 1806 para codificar, decodificar o transcodificar medios a, desde o entre uno o más formatos de codificación de medios, incluyendo, pero sin limitación, formatos del Grupo de Expertos en Imágenes en Movimiento (MPEG) tales como MPEG-2, formatos de Codificación de Vídeo Avanzada (AVC) tales como H.264/MPEG-4 AVC, así como de la Sociedad de Ingenieros de Imágenes en Movimiento y de Televisión (SMPTE) 421M/VC-1 y formatos del Grupo Conjunto de Expertos en Fotografía (JPEG) tales como los formatos JPEG y Motion JPEG (MJPEG).

En algunas realizaciones, el procesador de gráficos 1800 incluye un motor de transferencia de imágenes en bloque (BLIT) 1804 para realizar operaciones de rasterizador bidimensionales (2D), incluyendo, por ejemplo, transferencias de bloque de frontera de bits. Sin embargo, en una realización, se realizan operaciones de gráficos 2D usando uno o más componentes del motor de procesamiento de gráficos (GPE) 1810. En algunas realizaciones, el motor de procesamiento de gráficos 1810 es un motor de cálculo para realizar operaciones de gráficos, que incluyen operaciones de gráficos tridimensionales (3D) y operaciones de medios.

En algunas realizaciones, el GPE 1810 incluye una canalización 3D 1812 para realizar operaciones 3D, tal como representar imágenes y escenas tridimensionales usando funciones de procesamiento que actúan en formas de primitivas 3D (p. ej., rectángulo, triángulo, etc.). La canalización 3D 1812 incluye elementos de función programable y fija que realizan diversas tareas dentro del elemento y/o abarcan hilos de ejecución en un subsistema 3D/de medios 1815. Aunque puede usarse la canalización 3D 1812 para realizar operaciones de medios, una realización del GPE 1810 también incluye una canalización de medios 1816 que se usa específicamente para realizar operaciones de medios, tales como postprocesamiento de vídeo y mejora de imagen.

En algunas realizaciones, la canalización de medios 1816 incluye unidades de lógica de función fija o programable para realizar una o más operaciones de medios especializadas, tales como aceleración de decodificación de vídeo, desentrelazado de vídeo y aceleración de codificación de vídeo en lugar de, o en beneficio del motor de códec de vídeo 1806. En algunas realizaciones, la canalización de medios 1816 incluye adicionalmente una unidad de generación de hilos para generar hilos para su ejecución en el subsistema 3D/de medios 1815. Los hilos generados realizan cálculos para las operaciones de medios en una o más unidades de ejecución de gráficos incluidas en el subsistema 3D/de medios 1815.

En algunas realizaciones, el subsistema 3D/de medios 1815 incluye una lógica para ejecutar hilos generados por la canalización 3D 1812 y la canalización de medios 1816. En una realización, las canalizaciones envían solicitudes de ejecución de hilos al subsistema 3D/de medios 1815, incluyendo una lógica de despacho de hilos para arbitrar y despachar las diversas solicitudes a recursos de ejecución de hilos disponibles. Los recursos de ejecución incluyen una matriz de unidades de ejecución de gráficos para procesar los hilos 3D y de medios. En algunas realizaciones, el subsistema 3D/de medios 1815 incluye una o más cachés internas para datos e instrucciones de hilo. En algunas realizaciones, el subsistema también incluye memoria compartida, incluyendo registros y memoria direccionable, para compartir datos entre hilos y para almacenar datos de salida.

Procesamiento 3D/de medios

La Figura 19 es un diagrama de bloques de un motor de procesamiento de gráficos 1910 de un procesador de gráficos de acuerdo con algunas realizaciones. En una realización, el GPE 1910 es una versión del GPE 1810 mostrado en la Figura 18. Los elementos de la Figura 19 que tienen los mismos números (o nombres) de referencia que los elementos de cualquier otra figura en el presente documento pueden operar o funcionar de cualquier manera similar a la descrita en cualquier otra parte en el presente documento, pero sin limitación a esto.

En algunas realizaciones, el GPE 1910 se acopla con un emisor por flujo continuo de comandos 1903, que proporciona un flujo de comandos al GPE 3D y canalizaciones de medios 1912, 1916. En algunas realizaciones, el emisor por flujo continuo de comandos 1903 está acoplado con memoria, que puede ser memoria de sistema, o una o más de memoria de caché interna y memoria de caché compartida. En algunas realizaciones, el emisor por flujo continuo de comandos 1903 recibe comandos desde la memoria y envía los comandos a la canalización 3D 1912 y/o a la canalización de medios 1916. Los comandos son directivas extraídas desde una memoria intermedia en anillo, que almacena comandos para las canalizaciones 3D y de medios 1912, 1916. En una realización, la memoria intermedia en anillo puede incluir adicionalmente unas memorias intermedias de comandos en lotes que almacenan lotes de múltiples comandos. Las canalizaciones 3D y de medios 1912, 1916 procesan los comandos realizando operaciones a través de la lógica dentro de las respectivas canalizaciones o despachando uno o más hilos de ejecución a una matriz de unidades de ejecución 1914. En algunas realizaciones, la matriz de unidades de ejecución 1914 es escalable, de modo que la matriz incluye un número variable de unidades de ejecución basándose en la potencia objetivo y el nivel de rendimiento de GPE 1910.

En algunas realizaciones, un motor de muestreo 1930 se acopla con la memoria (p. ej., la memoria caché o memoria de sistema) y la matriz de unidades de ejecución 1914. En algunas realizaciones, el motor de muestreo 1930 proporciona un mecanismo de acceso a memoria para la matriz de unidades de ejecución 1914 que permite que la matriz de ejecución 1914 lea gráficos y datos de medios de la memoria. En algunas realizaciones, el motor de muestreo 1930 incluye lógica para realizar operaciones de muestreo de imágenes especializadas para medios.

En algunas realizaciones, la lógica de muestreo de medios especializada en el motor de muestreo 1930 incluye un módulo de anulación de ruido/desentrelazado 1932, un módulo de estimación de movimiento 1934 y un módulo de escalado y filtrado de imágenes 1936. En algunas realizaciones, el módulo de anulación de ruido/desentrelazado 1932 incluye lógica para realizar uno o más algoritmos de anulación de ruido o desentrelazado en datos de vídeo decodificados. La lógica de desentrelazado combina campos alternos de contenido de vídeo entrelazado en un único fotograma de vídeo. La lógica de anulación de ruido reduce o elimina el ruido de los datos de vídeo e imagen. En algunas realizaciones, la lógica de anulación de ruido y la lógica de desentrelazado se adaptan al movimiento y usan filtrado espacial o temporal basándose en la cantidad de movimiento detectado en los datos de vídeo. En algunas realizaciones, el módulo de anulación de ruido/desentrelazado 1932 incluye lógica de detección de movimiento especializada (p. ej., dentro del motor de estimación de movimiento 1934).

En algunas realizaciones, el motor de estimación de movimiento 1934 proporciona aceleración de hardware para operaciones de vídeo realizando funciones de aceleración de vídeo tales como estimación y predicción de vector de movimiento en datos de vídeo. El motor de estimación de movimiento determina los vectores de movimiento que describen la transformación de los datos de imagen entre fotogramas de vídeo sucesivos. En algunas realizaciones, un códec de medios de procesador de gráficos usa el motor de estimación de movimiento de vídeo 1934 para realizar operaciones en vídeo a nivel de macrobloque que, de lo contrario, pueden ser demasiado intensivas a nivel de computación para realizarlas con un procesador de propósito general. En algunas realizaciones, el motor de estimación de movimiento 1934 está generalmente disponible para los componentes de procesador de gráficos para ayudar con las funciones de procesamiento y decodificación de vídeo que son sensibles o adaptables a la dirección o magnitud del movimiento dentro de los datos de vídeo.

En algunas realizaciones, el módulo de escalado y filtrado de imágenes 1936 realiza operaciones de procesamiento de imágenes para mejorar la calidad visual de las imágenes y de vídeo generados. En algunas realizaciones, el módulo de escalado y filtrado 1936 procesa datos de imagen y vídeo durante la operación de muestreo antes de proporcionar los datos a la matriz de unidades de ejecución 1914.

En algunas realizaciones, el GPE 1910 incluye un puerto de datos 1944, que proporciona un mecanismo adicional para que los subsistemas de gráficos accedan a memoria. En algunas realizaciones, el puerto de datos 1944 facilita el acceso a memoria para operaciones que incluyen escrituras objetivo de representación, lecturas de memoria intermedia constantes, lecturas/escrituras de espacio de memoria de borrador y accesos a la superficie de medios. En algunas realizaciones, el puerto de datos 1944 incluye espacio de memoria caché para almacenar en caché los accesos a memoria. La memoria caché puede ser una única caché de datos o estar separada en múltiples cachés para los múltiples subsistemas que acceden a la memoria a través del puerto de datos (p. ej., una caché de memoria intermedia de representación, una caché de memoria intermedia constante, etc.). En algunas realizaciones, los hilos que se ejecutan en una unidad de ejecución en la matriz de unidades de ejecución 1914 se comunican con el puerto de datos intercambiando mensajes a través de una interconexión de distribución de datos que acopla cada uno de los subsistemas de GPE 1910.

Unidades de ejecución

La Figura 20 es un diagrama de bloques de otra realización de un procesador de gráficos 2000. Los elementos de la Figura 20 que tienen los mismos números (o nombres) de referencia que los elementos de cualquier otra figura en el presente documento pueden operar o funcionar de cualquier manera similar a la descrita en cualquier otra parte en el presente documento, pero sin limitación a esto.

En algunas realizaciones, el procesador de gráficos 2000 incluye una interconexión en anillo 2002, un extremo frontal de canalización 2004, un motor de medios 2037 y núcleos de gráficos 2080A-2080N. En algunas realizaciones, la interconexión en anillo 2002 acopla el procesador de gráficos a otras unidades de procesamiento, que incluyen otros procesadores de gráficos o uno o más núcleos de procesadores de fin general. En algunas realizaciones, el procesador de gráficos es uno de muchos procesadores integrados dentro de un sistema de procesamiento de múltiples núcleos.

En algunas realizaciones, el procesador de gráficos 2000 recibe lotes de comandos mediante la interconexión en anillo 2002. Los comandos de entrada se interpretan por un emisor por flujo continuo de comandos 2003 en el extremo frontal de canalización 2004. En algunas realizaciones, el procesador de gráficos 2000 incluye una lógica de ejecución escalable para realizar procesamiento de geometría 3D y procesamiento de medios mediante el núcleo o núcleos de gráficos 2080A-2080N. Para los comandos de procesamiento de geometría 3D, el emisor por flujo continuo de comandos 2003 suministra comandos a la canalización de geometría 2036. Para al menos algunos comandos de procesamiento de medios, el emisor por flujo continuo de comandos 2003 suministra los comandos a un extremo frontal de vídeo 2034, que se acopla con un motor de medios 2037. En algunas realizaciones, el motor de medios 2037 incluye un motor de calidad de vídeo (VQE) 2030 para postprocesamiento de vídeo y de imagen y un motor de codificación/decodificación de múltiples formatos (MFX) 2033 para proporcionar codificación y decodificación de datos de medios acelerados por hardware. En algunas realizaciones, cada uno de la canalización de geometría 2036 y el motor de medios 2037 genera hilos de ejecución para los recursos de ejecución de hilos proporcionados por al menos un núcleo de gráficos 2080A.

En algunas realizaciones, el procesador de gráficos 2000 incluye recursos de ejecución de hilos ajustables a escala que cuentan con los núcleos modulares 2080A-2080N (denominados, en ocasiones, cortes de núcleo), teniendo cada uno múltiples subnúcleos 2050A-2050N, 2060A-2060N (denominados, en ocasiones, subcortes de núcleo). En algunas realizaciones, el procesador de gráficos 2000 puede tener cualquier número de núcleos de gráficos 2080A a 2080N. En algunas realizaciones, el procesador de gráficos 2000 incluye un núcleo de gráficos 2080A que tiene al menos un primer subnúcleo 2050A y un segundo subnúcleo de núcleo 2060A. En otras realizaciones, el procesador de gráficos es un procesador de baja potencia con un único subnúcleo (p. ej., 2050A). En algunas realizaciones, el procesador de gráficos 2000 incluye múltiples núcleos de gráficos 2080A-2080N, incluyendo cada uno un conjunto de primeros subnúcleos 2050A-2050N y un conjunto de segundos subnúcleos 2060A-2060N. Cada subnúcleo del conjunto de primeros subnúcleos 2050A-2050N incluye al menos un primer conjunto de unidades de ejecución 2052A-2052N y muestreadores de medios/texturas 2054A-2054N. Cada subnúcleo del conjunto de segundos subnúcleos 2060A-2060N incluye al menos un segundo conjunto de unidades de ejecución 2062A-2062N y muestreadores 2064A-2064N. En algunas realizaciones, cada subnúcleo 2050A-2050N, 2060A-2060N comparte un conjunto de recursos compartidos 2070A-2070N. En algunas realizaciones, los recursos compartidos incluyen memoria de caché compartida y lógica de operación de píxel. Pueden incluirse también otros recursos compartidos en las diversas realizaciones del procesador de gráficos.

La Figura 21 ilustra lógica de ejecución de hilos 2100 que incluye una matriz de elementos de procesamiento empleados en algunas realizaciones de un GPE. Los elementos de la Figura 21 que tienen los mismos números (o nombres) de referencia que los elementos de cualquier otra figura en el presente documento pueden operar o funcionar de cualquier manera similar a la descrita en cualquier otra parte en el presente documento, pero sin limitación a esto.

En algunas realizaciones, la lógica de ejecución de hilos 2100 incluye un sombreador de píxeles 2102, un despachador de hilos 2104, una caché de instrucciones 2106, una matriz de unidades de ejecución escalable que incluye una pluralidad de unidades de ejecución 2108A-2108N, un muestreador 2110, una caché de datos 2112 y un puerto de datos 2114. En una realización, los componentes incluidos están interconectados mediante un tejido de interconexión que se enlaza a cada uno de los componentes. En algunas realizaciones, la lógica de ejecución de hilos 2100 incluye una o más conexiones a memoria, tal como la memoria de sistema o memoria caché, a través de una o más de la caché de instrucciones 2106, el puerto de datos 2114, el muestreador 2110 y la matriz de unidades de ejecución 2108A-2108N. En algunas realizaciones, cada unidad de ejecución (p. ej., 2108A) es un procesador de vector individual que puede ejecutar múltiples hilos simultáneos y procesar múltiples elementos de datos en paralelo para cada hilo. En algunas realizaciones, la matriz de unidades de ejecución 2108A-2108N incluye cualquier número de unidades de ejecución individuales.

En algunas realizaciones, la matriz de unidades de ejecución 2108A-2108N se usa principalmente para ejecutar programas de "sombreador". En algunas realizaciones, las unidades de ejecución en la matriz 2108A-2108N ejecutan un conjunto de instrucciones que incluye el soporte nativo para muchas instrucciones de sombreador de gráficos 3D convencional, de manera que se ejecutan los programas de sombreador de las bibliotecas de gráficos (p. ej., Direct 3D y OpenGL) con una traducción mínima. Las unidades de ejecución soportan procesamiento de vértices y de geometría (p. ej., programas de vértices, programas de geometría, sombreadores de vértices), procesamiento de píxeles (p. ej., sombreadores de píxeles, sombreadores de fragmentos) y procesamiento de fin general (p. ej., sombreadores de cálculo y de medios).

Cada unidad de ejecución en la matriz de unidades de ejecución 2108A-2108N opera en matrices de elementos de datos. El número de elementos de datos es el "tamaño de ejecución", o el número de canales para la instrucción. Un canal de ejecución es una unidad lógica de ejecución para el acceso de elemento de datos, el enmascaramiento y el

control de flujo dentro de las instrucciones. El número de canales puede ser independiente del número de Unidades Aritméticas Lógicas (ALU) o Unidades de Coma Flotante (FPU) físicas para un procesador de gráficos particular. En algunas realizaciones, las unidades de ejecución 2108A-2108N soportan tipos de datos de números enteros y de coma flotante.

El conjunto de instrucciones de la unidad de ejecución incluye instrucciones de datos múltiples de una sola instrucción (SIMD). Los diversos elementos de datos pueden almacenarse como un tipo de datos empaquetado en un registro y la unidad de ejecución procesará los diversos elementos basándose en el tamaño de datos de los elementos. Por ejemplo, cuando se opera en un vector de 256 bits de ancho, los 256 bits del vector se almacenan en un registro y la unidad de ejecución opera en el vector como cuatro elementos de datos empaquetados de 64 bits separados (elementos de datos de tamaño de palabra cuádruple (QW)), ocho elementos de datos empaquetados de 32 bits separados (elementos de datos de tamaño de palabra doble (DW)), dieciséis elementos de datos empaquetados de 16 bits separados (elementos de datos de tamaño de palabra (W)), o treinta y dos elementos de datos de 8 bits separados (elementos de datos de tamaño de byte (B)). Sin embargo, son posibles diferentes tamaños de anchuras de vector y registros.

Una o más cachés de instrucción internas (p. ej., 2106) están incluidas en la lógica de ejecución de hilo 2100 a las instrucciones de hilo de caché para las unidades de ejecución. En algunas realizaciones, se incluyen una o más cachés de datos (p. ej., 2112) para almacenar en caché datos de hilo durante la ejecución de hilo. En algunas realizaciones, se incluye un muestreador 2110 para proporcionar un muestreo de textura para operaciones 3D y muestreo de medios para operaciones de medios. En algunas realizaciones, el muestreador 2110 incluye funcionalidad de textura especializada o muestreo de medios para procesar los datos de textura o de medios durante el proceso de muestreo antes de proporcionar los datos muestreados a una unidad de ejecución.

Durante la ejecución, las canalizaciones de gráficos y de medios envían solicitudes de iniciación de hilo a la lógica de ejecución de hilos 2100 mediante una lógica de generación y de despacho de hilos. En algunas realizaciones, la lógica de ejecución de hilos 2100 incluye un despachador de hilos local 2104 que arbitra las solicitudes de inicio de hilos de las canalizaciones de gráficos y medios y genera instancias a los hilos solicitados en una o más unidades de ejecución 2108A-2108N. Por ejemplo, la canalización de geometría (p. ej., 2036 de la Figura 20) despacha hilos de procesamiento de vértices, teselación o procesamiento de geometría a la lógica de ejecución de hilos 2100 (Figura 21). En algunas realizaciones, el despachador de hilos 2104 puede procesar también hilos en tiempo de ejecución que abarcan solicitudes desde los programas de sombreador de ejecución.

Una vez que un grupo de objetos geométricos ha sido procesado y rasterizado en datos de píxeles, se invoca el sombreador de píxeles 2102 para calcular además la información de salida y hacer que los resultados se escriban en las superficies de salida (p. ej., memorias intermedias de color, memorias intermedias de profundidad, memorias intermedias de estarcido, etc.). En algunas realizaciones, el sombreador de píxeles 2102 calcula los valores de los diversos atributos de vértice que han de interpolarse a través del objeto rasterizado. En algunas realizaciones, el sombreador de píxeles 2102 a continuación ejecuta un programa de sombreador de píxeles suministrado por la interfaz de programación de aplicaciones (API). Para ejecutar el programa de sombreador de píxeles, el sombreador de píxeles 2102 despacha hilos a una unidad de ejecución (p. ej., 2108A) mediante el despachador de hilos 2104. En algunas realizaciones, el sombreador de píxeles 2102 usa la lógica de muestreo de textura en el muestreador 2110 para acceder a datos de textura en mapas de textura almacenados en memoria. Las operaciones aritméticas en los datos de textura y los datos de geometría de entrada calculan datos de color de píxel para cada fragmento geométrico, o descartan uno o más píxeles de su procesamiento adicional.

En algunas realizaciones, el puerto de datos 2114 proporciona un mecanismo de acceso a memoria para que la lógica de ejecución de hilos 2100 emita datos procesados a memoria para su procesamiento en una canalización de salida de procesador de gráficos. En algunas realizaciones, el puerto de datos 2114 incluye o se acopla a una o más memorias de caché (p. ej., la caché de datos 2112) para almacenar en caché datos para el acceso a memoria mediante el puerto de datos.

La Figura 22 es un diagrama de bloques que ilustra unos formatos de instrucción de procesador de gráficos 2200 de acuerdo con algunas realizaciones. En una o más realizaciones, las unidades de ejecución de procesador de gráficos soportan un conjunto de instrucciones que tiene instrucciones en múltiples formatos. Los recuadros con línea continua ilustran los componentes que se incluyen en general en una instrucción de unidad de ejecución, mientras que las líneas discontinuas incluyen componentes que son opcionales o que únicamente están incluidos en un subconjunto de las instrucciones. En algunas realizaciones, el formato de instrucción 2200 descrito e ilustrado son macroinstrucciones, en el sentido de que las mismas son instrucciones suministradas a la unidad de ejecución, en contraposición a microoperaciones resultantes de la decodificación de instrucciones una vez que se ha procesado la instrucción.

En algunas realizaciones, las unidades de ejecución de procesador de gráficos soportan de manera nativa las instrucciones en un formato de 128 bits 2210. Un formato de instrucción de 64 bits compacto 2230 está disponible para algunas instrucciones basándose en la instrucción seleccionada, las opciones de instrucción y el número de operandos. El formato de 128 bits nativo 2210 proporciona acceso a todas las opciones de instrucción, mientras que

algunas opciones y operaciones están restringidas en el formato de 64 bits 2230. Las instrucciones nativas disponibles en el formato de 64 bits 2230 varían por realización. En algunas realizaciones, la instrucción se compacta en parte usando un conjunto de valores de índice en un campo de índice 2213. El hardware de la unidad de ejecución hace referencia a un conjunto de tablas de compactación basándose en los valores de índice y usa las salidas de tabla de compactación para reconstruir una instrucción nativa en el formato de 128 bits 2210.

Para cada formato, el código de operación de instrucción 2212 define la operación que ha de realizar la unidad de ejecución. Las unidades de ejecución ejecutan cada instrucción en paralelo a lo largo de los múltiples elementos de datos de cada operando. Por ejemplo, en respuesta a una instrucción de adición, la unidad de ejecución realiza una operación de adición simultánea a través de cada canal de color que representa un elemento de textura o elemento de imagen. Por defecto, la unidad de ejecución realiza cada instrucción a través de todos los canales de datos de los operandos. En algunas realizaciones, el campo de control de instrucción 2214 posibilita el control a través de ciertas opciones de ejecución, tal como la selección de canales (p. ej., predicación) y orden de canal de datos (p. ej., mezcla). Para las instrucciones de 128 bits 2210, un campo de tamaño de ejecución 2216 limita el número de canales de datos que se ejecutarán en paralelo. En algunas realizaciones, el campo de tamaño de ejecución 2216 no está disponible para su uso en el formato de instrucción compacto de 64 bits 2230.

Algunas instrucciones de la unidad de ejecución tienen hasta tres operandos, incluyendo dos operandos de origen, src0 2220, src1 2222 y un destino 2218. En algunas realizaciones, las unidades de ejecución soportan instrucciones de destino dual, donde uno de los destinos está implícito. Las instrucciones de manipulación de datos pueden tener un tercer operando de origen (p. ej., SRC2 2224), donde el código de operación de instrucción 2212 determina el número de operandos de origen. El último operando de origen de una instrucción puede ser un valor inmediato (p. ej., codificado de manera rígida) pasado con la instrucción.

En algunas realizaciones, el formato de instrucción de 128 bits 2210 incluye una información de modo de acceso/dirección 2226 que especifica, por ejemplo, si se usa el modo de direccionamiento de registro directo o el modo de direccionamiento de registro indirecto. Cuando se usa el modo de direccionamiento de registro directo, la dirección de registro de uno o más operandos se proporciona directamente por los bits en la instrucción 2210.

En algunas realizaciones, el formato de instrucción de 128 bits 2210 incluye un campo de modo de acceso/dirección 2226, que especifica un modo de dirección y/o un modo de acceso para la instrucción. En una realización, el modo de acceso para definir una alineación de acceso de datos para la instrucción. Algunas realizaciones soportan modos de acceso que incluyen un modo de acceso alineado de 16 bytes y un modo de acceso alineado de 1 byte, donde la alineación de bytes del modo de acceso determina la alineación de acceso de los operandos de instrucción. Por ejemplo, cuando está en un primer modo, la instrucción 2210 puede usar un direccionamiento alineado en bytes para operandos de origen y destino y, cuando está en un segundo modo, la instrucción 2210 puede usar direccionamiento alineado de 16 bytes para todos los operandos de origen y destino.

En una realización, la porción de modo de dirección del campo de modo de acceso/dirección 2226 determina si la instrucción es para usar el direccionamiento directo o indirecto. Cuando se usa el modo de direccionamiento de registro directo, los bits en la instrucción 2210 proporcionan directamente la dirección de registro de uno o más operandos. Cuando se usa el modo de direccionamiento de registro indirecto, puede calcularse la dirección de registro de uno o más operandos basándose en un valor de registro de dirección y un campo de dirección inmediata en la instrucción.

En algunas realizaciones, las instrucciones se agrupan basándose en los campos de bits del código de operación 2212 para simplificar la decodificación del código de operación 2240. Para un código de operación de 8 bits, los bits 4, 5 y 6 permiten que la unidad de ejecución determine el tipo de código de operación. La agrupación de código de operación precisa mostrada es simplemente un ejemplo. En algunas realizaciones, un grupo de código de operación de movimiento y de lógica 2242 incluye instrucciones de movimiento y de lógica de datos (p. ej., mover (mov), comparar (cmp)). En algunas realizaciones, el grupo de movimiento y lógica 2242 comparte los cinco bits más significativos (MSB), donde las instrucciones mover (mov) están en forma de 0000xxxxb y las instrucciones de lógica están en forma de 0001xxxxb. Un grupo de instrucciones de control de flujo 2244 (p. ej., llamada, salto (jmp)) incluye instrucciones en forma de 0010xxxxb (p. ej., 0x20). Un grupo de instrucciones misceláneas 2246 incluye una mezcla de instrucciones, incluyendo instrucciones de sincronización (p. ej., esperar, enviar) en forma de 0011xxxxb (p. ej., 0x30). Un grupo de instrucciones de cálculo matemático paralelo 2248 incluye instrucciones aritméticas a nivel de componente (p. ej., sumar, multiplicar (mult)) en forma de 0100xxxxb (p. ej., 0x40). El grupo de cálculo matemático paralelo 2248 realiza las operaciones aritméticas en paralelo a lo largo de canales de datos. El grupo de cálculo matemático vectorial 2250 incluye instrucciones aritméticas (p. ej., dp4) en forma de 0101xxxxb (p. ej., 0x50). El grupo de cálculo matemático vectorial realiza la aritmética tal como los cálculos de producto escalar en operandos vectoriales.

Canalización de gráficos

La Figura 23 es un diagrama de bloques de otra realización de un procesador de gráficos 2300. Los elementos de la Figura 23 que tienen los mismos números (o nombres) de referencia que los elementos de cualquier otra figura en el presente documento pueden operar o funcionar de cualquier manera similar a la descrita en cualquier otra parte en el presente documento, pero sin limitación a esto.

En algunas realizaciones, el procesador de gráficos 2300 incluye una canalización de gráficos 2320, una canalización de medios 2330, un motor de visualización 2340, lógica de ejecución de hilos 2350 y una canalización de salida de representación 2370. En algunas realizaciones, el procesador de gráficos 2300 es un procesador de gráficos dentro de un sistema de procesamiento de múltiples núcleos que incluye uno o más núcleos de procesamiento de fin general. El procesador de gráficos es controlado por escrituras de registro en uno o más registros de control (no mostrados) o mediante comandos emitidos al procesador de gráficos 2300 mediante una interconexión en anillo 2302. En algunas realizaciones, la interconexión en anillo 2302 acopla el procesador de gráficos 2300 a otros componentes de procesamiento, tales como otros procesadores de gráficos o procesadores de fin general. Los comandos desde la interconexión en anillo 2302 son interpretados por un emisor por flujo continuo de comandos 2303, que suministra instrucciones a componentes individuales de la canalización de gráficos 2320 o la canalización de medios 2330.

En algunas realizaciones, el emisor por flujo continuo de comandos 2303 dirige la operación de un extractor de vértices 2305 que lee datos de vértices desde memoria y ejecuta comandos de procesamiento de vértices proporcionados por el emisor por flujo continuo de comandos 2303. En algunas realizaciones, el extractor de vértices 2305 proporciona datos de vértices a un sombreador de vértices 2307, que realiza operaciones de transformación espacial de coordenadas y de iluminación en cada vértice. En algunas realizaciones, el extractor de vértices 2305 y el sombreador de vértices 2307 ejecutan instrucciones de procesamiento de vértices despachando hilos de ejecución a unidades de ejecución 2352A, 2352B mediante un despachador de hilos 2331.

En algunas realizaciones, las unidades de ejecución 2352A, 2352B son una matriz de procesadores vectoriales que tienen un conjunto de instrucciones para realizar operaciones de gráficos y de medios. En algunas realizaciones, las unidades de ejecución 2352A, 2352B tienen una caché L1 adjunta 2351 que es específica para cada matriz o está compartida entre las matrices. La caché puede estar configurada como una caché de datos, una caché de instrucciones o una única caché que está subdividida para contener datos e instrucciones en diferentes subdivisiones.

En algunas realizaciones, la canalización de gráficos 2320 incluye componentes de teselación para realizar teselación acelerada por hardware de objetos 3D. En algunas realizaciones, un sombreador de casco programable 2311 configura las operaciones de teselación. Un sombreador de domino programable 2317 proporciona una evaluación de extremo trasero de la salida de teselación. Un teselador 2313 opera en la dirección del sombreador de casco 2311 y contiene lógica de fin especial para generar un conjunto de objetos geométricos detallados basándose en un modelo geométrico aproximado que se proporciona como entrada a la canalización de gráficos 2320. En algunas realizaciones, si no se usa la teselación, pueden omitirse los componentes de teselación 2311, 2313, 2317.

En algunas realizaciones, pueden procesarse objetos geométricos completos por un sombreador de geometría 2319 mediante uno o más hilos despachados a unidades de ejecución 2352A, 2352B, o pueden continuar directamente al recortador 2329. En algunas realizaciones, el sombreador de geometría opera en objetos geométricos enteros, en lugar de en vértices o parches de vértices como en etapas anteriores de la canalización de gráficos. Si se desactiva la teselación, el sombreador de geometría 2319 recibe entrada desde el sombreador de vértices 2307. En algunas realizaciones, el sombreador de geometría 2319 se puede programar mediante un programa sombreador de geometría para realizar una teselación de geometría si las unidades de teselación están deshabilitadas.

Antes de la rasterización, un recortador 2329 procesa datos de vértices. El recortador 2329 puede ser un recortador de función fija o un recortador programable que tiene funciones de recorte y de sombreador de geometría. En algunas realizaciones, un rasterizador 2373 (p. ej., el componente de prueba de profundidad) en la canalización de salida de representación 2370 despacha sombreadores de píxeles para convertir los objetos geométricos en sus representaciones por píxeles. En algunas realizaciones, la lógica de sombreador de píxeles está incluida en la lógica de ejecución de hilos 2350. En algunas realizaciones, una aplicación puede omitir el rasterizador 2373 y acceder a datos de vértices no rasterizados mediante una unidad de salida de flujo 2323.

El procesador de gráficos 2300 tiene un bus de interconexión, tejido de interconexión o algún otro mecanismo de interconexión que permite que los datos y los mensajes pasen entre los componentes principales del procesador. En algunas realizaciones, las unidades de ejecución 2352A, 2352B y la caché o cachés asociadas 2351, el muestreador de textura y de medios 2354 y la caché de textura/muestreador 2358 se interconectan mediante un puerto de datos 2356 para realizar el acceso a memoria y comunicarse con los componentes de canalización de salida de representación del procesador. En algunas realizaciones, cada uno del muestreador 2354, las cachés 2351, 2358 y las unidades de ejecución 2352A, 2352B tienen rutas de acceso a memoria separadas.

En algunas realizaciones, la canalización de salida de representación 2370 contiene un rasterizador 2373 que convierte objetos basados en vértices en una representación basada en píxeles asociada. En algunas realizaciones, la lógica del rasterizador incluye una unidad generadora de ventanas/enmascaradora para realizar rasterización de triángulo y de línea de función fija. Una caché de representación asociada 2378 y caché de profundidad 2379 también están disponibles en algunas realizaciones. Un componente de operaciones de píxel 2377 realiza operaciones basadas en píxeles sobre los datos, aunque, en algunas instancias, las operaciones de píxel asociadas con operaciones 2D (p. ej., transferencias de imagen de bloque de bits con mezcla) son realizadas por el motor 2D 2341, o son sustituidas en el momento de la visualización por el controlador de visualización 2343 usando planos de

visualización de superposición. En algunas realizaciones, está disponible una caché L3 compartida 2375 para todos los componentes de gráficos, permitiendo la compartición de datos sin el uso de memoria de sistema principal.

En algunas realizaciones, la canalización de medios del procesador de gráficos 2330 incluye un motor de medios 2337 y un extremo frontal de vídeo 2334. En algunas realizaciones, el extremo frontal de vídeo 2334 recibe comandos de canalización desde el emisor por flujo continuo de comandos 2303. En algunas realizaciones, la canalización de medios 2330 incluye un emisor por flujo continuo de comandos separado. En algunas realizaciones, el extremo frontal de vídeo 2334 procesa comandos de medios antes de enviar el comando al motor de medios 2337. En algunas realizaciones, el motor de medios 2337 incluye una funcionalidad de generación de hilos para generar hilos para despacharlos a la lógica de ejecución de hilos 2350 mediante el despachador de hilos 2331.

En algunas realizaciones, el procesador de gráficos 2300 incluye un motor de visualización 2340. En algunas realizaciones, el motor de visualización 2340 es externo al procesador 2300 y se acopla con el procesador de gráficos mediante la interconexión en anillo 2302, o algún otro bus o tejido de interconexión. En algunas realizaciones, el motor de visualización 2340 incluye un motor 2D 2341 y un controlador de visualización 2343. En algunas realizaciones, el motor de visualización 2340 contiene lógica de fin especial que puede operar independientemente de la canalización 3D. En algunas realizaciones, el controlador de visualización 2343 se acopla con un dispositivo de visualización (no mostrado), que puede ser un dispositivo de visualización integrado en sistema, como en un ordenador portátil, o un dispositivo de visualización externo adjunto mediante un conector de dispositivo de visualización.

En algunas realizaciones, la canalización de gráficos 2320 y la canalización de medios 2330 se pueden configurar para realizar operaciones basándose en múltiples interfaces de programación de gráficos y de medios y no son específicas de ninguna interfaz de programación de aplicaciones (API) concreta. En algunas realizaciones, el software del controlador para el procesador de gráficos traduce llamadas API que son específicas a gráficos o a bibliotecas de medios particulares en comandos que pueden procesarse por el procesador de gráficos. En algunas realizaciones, se proporciona soporte para la biblioteca Open Graphics (OpenGL) y Open Computing Language (OpenCL) de Khronos Group, la biblioteca Direct3D de Microsoft Corporation, o se puede proporcionar soporte tanto para OpenGL como para D3D. También se puede proporcionar soporte para la Biblioteca de Visión Informática de Código Abierto (OpenCV). También se soportaría una API futura con una canalización 3D compatible si pudiera hacerse un mapeo de la canalización de la API futura a la canalización del procesador de gráficos.

Programación de canalización de gráficos

La Figura 24A es un diagrama de bloques que ilustra un formato de comando de procesador de gráficos 2400 de acuerdo con algunas realizaciones. La Figura 24B es un diagrama de bloques que ilustra una secuencia de comandos de procesador de gráficos 2410 de acuerdo con una realización. Los recuadros de línea continua en la Figura 24A ilustran los componentes que están incluidos en general en un comando de gráficos, mientras que las líneas discontinuas incluyen componentes que son opcionales o que están incluidos únicamente en un subconjunto del comando de gráficos. El formato de comando de procesador de gráficos 2400 ilustrativo de la Figura 24A incluye campos de datos para identificar un cliente objetivo 2402 del comando, un código de operación del comando (código de operación) 2404 y los datos relevantes 2406 para el comando. También se incluye un subcódigo de operación 2405 y un tamaño de comando 2408 en algunos comandos.

En algunas realizaciones, el cliente 2402 especifica la unidad de cliente del dispositivo de gráficos que procesa los datos de comando. En algunas realizaciones, un analizador de comando de procesador de gráficos examina el campo de cliente de cada comando para acondicionar el procesamiento adicional del comando y encaminar los datos de comando a la unidad de cliente apropiada. En algunas realizaciones, las unidades de cliente de procesador de gráficos incluyen una unidad de interfaz de memoria, una unidad de representación, una unidad 2D, una unidad 3D y una unidad de medios. Cada unidad de cliente tiene una canalización de procesamiento correspondiente que procesa los comandos. Una vez que el comando ha sido recibido por la unidad de cliente, la unidad de cliente lee el código de operación 2404 y, si está presente, el subcódigo de operación 2405 para determinar la operación a realizar. La unidad de cliente realiza el comando usando información en el campo de datos 2406. Para algunos comandos, se espera que un tamaño de comando explícito 2408 especifique el tamaño del comando. En algunas realizaciones, el analizador de comandos determina automáticamente el tamaño de al menos algunos de los comandos basándose en el código de operación de comando. En algunas realizaciones, los comandos se alinean mediante múltiplos de una palabra doble.

El diagrama de flujo en la Figura 24B muestra una secuencia de comandos de procesador de gráficos 2410 ilustrativa. En algunas realizaciones, el software o firmware de un sistema de procesamiento de datos que presenta una realización de un procesador de gráficos usa una versión de la secuencia de comandos mostrada para establecer, ejecutar y terminar un conjunto de operaciones de gráficos. Se muestra una secuencia de comandos de muestra y se describe para los fines de ejemplo únicamente ya que las realizaciones no están limitadas a estos comandos específicos o para esta secuencia de comandos. Además, pueden emitirse los comandos como un lote de comandos en una secuencia de comandos, de manera que el procesador de gráficos procesará la secuencia de comandos en concurrencia al menos parcialmente.

- En algunas realizaciones, la secuencia de comandos de procesador de gráficos 2410 puede comenzar con un comando de vaciado de canalización 2412 para hacer que cualquier canalización de gráficos activa complete los comandos actualmente pendientes para la canalización. En algunas realizaciones, la canalización 3D 2422 y la canalización de medios 2424 no operan concurrentemente. Se realiza el vaciado de la canalización para hacer que la canalización de gráficos activa complete algún comando pendiente. En respuesta a un vaciado de canalización, el analizador de comando para el procesador de gráficos pausará el procesamiento de comandos hasta que los motores de dibujo activos completen las operaciones pendientes y se invaliden las cachés de lectura relevantes. Opcionalmente, cualquier dato en la caché de representación que se marca como 'sucio' puede vaciarse a memoria. En algunas realizaciones, puede usarse el comando de vaciado de canalización 2412 para la sincronización de canalización o antes de colocar el procesador de gráficos en un estado de baja potencia.
- En algunas realizaciones, se usa un comando de selección de canalización 2413 cuando una secuencia de comandos requiere el procesador de gráficos para conmutar explícitamente entre canalizaciones. En algunas realizaciones, se requiere únicamente un comando de selección de canalización 2413 una vez dentro de un contexto de ejecución antes de emitir comandos de canalización a menos que el contexto sea emitir comandos para ambas canalizaciones. En algunas realizaciones, se requiere un comando de vaciado de canalización 2412 inmediatamente antes de un conmutador de canalización mediante el comando de selección de canalización 2413.
- En algunas realizaciones, un comando de control de canalización 2414 configura una canalización de gráficos para la operación y se usa para programar la canalización 3D 2422 y la canalización de medios 2424. En algunas realizaciones, el comando de control de canalización 2414 configura el estado de canalización para la canalización activa. En una realización, se usa el comando de control de canalización 2414 para sincronización de canalización y para limpiar datos de una o más memorias de caché dentro de la canalización activa antes de procesar un lote de comandos.
- En algunas realizaciones, se usan comandos de estado de memoria intermedia de retorno 2416 para configurar un conjunto de memorias intermedias de retorno para que las respectivas canalizaciones escriban datos. Algunas operaciones de canalización requieren la asignación, selección o configuración de una o más memorias intermedias de retorno en las que las operaciones escriben datos intermedios durante el procesamiento. En algunas realizaciones, el procesador de gráficos también usa una o más memorias intermedias de retorno para almacenar datos de salida y para realizar una comunicación de hilos cruzada. En algunas realizaciones, el estado de memoria intermedia de retorno 2416 incluye seleccionar el tamaño y el número de memorias intermedias de retorno que hay que usar para un conjunto de operaciones de canalización.
- Los comandos restantes en la secuencia de comandos difieren basándose en la canalización activa para las operaciones. Basándose en una determinación de la canalización 2420, la secuencia de comandos se adapta a la canalización 3D 2422 que comienza con el estado de canalización 3D 2430, o a la canalización de medios 2424 que comienza en el estado de canalización de medios 2440.
- Los comandos para el estado de canalización 3D 2430 incluyen los comandos de ajuste de estado 3D para el estado de memoria intermedia de vértice, estado de elemento de vértice, estado de color constante, estado de memoria intermedia de profundidad y otras variables de estado que han de configurarse antes de que se procesen los comandos de primitiva 3D. Los valores de estos comandos se determinan, al menos en parte, basándose en la API 3D particular en uso. En algunas realizaciones, los comandos de estado de canalización 3D 2430 también pueden desactivar o desviar selectivamente ciertos elementos de canalización si no se van a usar estos elementos.
- En algunas realizaciones, se usa el comando de primitiva 3D 2432 para enviar primitivas 3D a su procesamiento por la canalización 3D. Los comandos y parámetros asociados que se pasan al procesador de gráficos mediante el comando de primitiva 3D 2432 se reenvían a la función de extracción de vértice en la canalización de gráficos. La función de extracción de vértice usa los datos de comando de primitiva 3D 2432 para generar estructuras de datos de vértices. Las estructuras de datos de vértices se almacenan en una o más memorias intermedias de retorno. En algunas realizaciones, se usa el comando de primitiva 3D 2432 para realizar operaciones de vértice en primitivas 3D mediante sombreadores de vértice. Para procesar sombreadores de vértices, la canalización 3D 2422 despacha hilos de ejecución de sombreador a unidades de ejecución de procesador de gráficos.
- En algunas realizaciones, la canalización 3D 2422 se desencadena mediante un comando o evento de ejecución 2434. En algunas realizaciones, una escritura de registro desencadena una ejecución de comando. En algunas realizaciones, la ejecución se desencadena mediante un comando 'ir' o 'poner en marcha' en la secuencia de comandos. En una realización se activa la ejecución de comando usando un comando de sincronización de canalización para vaciar la secuencia de comandos a través de la canalización de gráficos. La canalización 3D realizará un procesamiento de geometría para las primitivas 3D. Una vez que están completadas las operaciones, se rasterizan los objetos geométricos resultantes y los colores de motor de píxel y los píxeles resultantes. Pueden incluirse también comandos adicionales para controlar el sombreado de píxeles y las operaciones de extremo trasero de píxeles para estas operaciones.
- En algunas realizaciones, la secuencia de comandos de procesador de gráficos 2410 sigue la ruta de canalización de medios 2424 cuando se realizan operaciones de medios. En general, el uso y manera específicos de la programación

para la canalización de medios 2424 depende de las operaciones de medios o de cálculo que van a realizarse. Las operaciones de decodificación de medios específicas pueden descargarse en la canalización de medios durante la decodificación de medios. En algunas realizaciones, puede desviarse también la canalización de medios y puede realizarse la decodificación de medios, en su totalidad o en parte, usando recursos proporcionados por uno o más núcleos de procesamiento de fin general. En una realización, la canalización de medios también incluye elementos para las operaciones de la unidad de procesador de gráficos de fin general (GPGPU), donde se usa el procesador de gráficos para realizar operaciones vectoriales SIMD usando programas de sombreador computacionales que no están relacionados explícitamente con la representación de primitivas de gráficos.

En algunas realizaciones, se configura la canalización de medios 2424 de una manera similar que la canalización 3D 2422. Se despacha o se coloca un conjunto de comandos de estado de canalización de medios 2440 en una cola de comandos antes de los comandos de objeto de medios 2442. En algunas realizaciones, los comandos de estado de canalización de medios 2440 incluyen datos para configurar los elementos de canalización de medios que se usarán para procesar los objetos de medios. Esto incluye datos para configurar la lógica de decodificación de vídeo y de codificación de vídeo dentro de la canalización de medios, tal como el formato de codificación o de decodificación. En algunas realizaciones, los comandos de estado de canalización de medios 2440 también soportan el uso de uno o más punteros a elementos de estado "indirecto" que contienen un lote de ajustes de estado.

En algunas realizaciones, los comandos de objeto de medios 2442 suministran punteros a objetos de medios para su procesamiento por la canalización de medios. Los objetos de medios incluyen memorias intermedias de memoria que contienen datos de vídeo que hay que procesar. En algunas realizaciones, todos los estados de canalización de medios han de ser válidos antes de emitir un comando de objeto de medios 2442. Una vez que se ha configurado el estado de canalización y los comandos de objeto de medios 2442 se han puesto en cola, la canalización de medios 2424 se desencadena mediante un comando de ejecución 2444 o un evento de ejecución equivalente (p. ej., una escritura de registro). La salida desde la canalización de medios 2424 puede postprocesarse entonces mediante operaciones proporcionadas por la canalización 3D 2422 o la canalización de medios 2424. En algunas realizaciones, las operaciones de GPGPU se configuran y se ejecutan de una manera similar a la de las operaciones de medios.

Arquitectura de software de gráficos

La Figura 25 ilustra una arquitectura de software de gráficos ilustrativa para un sistema de procesamiento de datos 2500 de acuerdo con algunas realizaciones. En algunas realizaciones, la arquitectura de software incluye una aplicación de gráficos 3D 2510, un sistema operativo 2520 y al menos un procesador 2530. En algunas realizaciones, el procesador 2530 incluye un procesador de gráficos 2532 y uno o más núcleos de procesador de fin general 2534. Cada uno de la aplicación de gráficos 2510 y el sistema operativo 2520 se ejecuta en la memoria de sistema 2550 del sistema de procesamiento de datos.

En algunas realizaciones, la aplicación de gráficos 3D 2510 contiene uno o más programas de sombreador que incluyen las instrucciones de sombreador 2512. Las instrucciones de lenguaje de sombreador pueden estar en un lenguaje de sombreador de alto nivel, tal como el Lenguaje de Sombreador de Alto Nivel (HLSL) o el Lenguaje de Sombreador OpenGL (GLSL). La aplicación también incluye instrucciones ejecutables 2514 en un lenguaje máquina adecuado para su ejecución por el núcleo de procesador de fin general 2534. La aplicación también incluye los objetos de gráficos 2516 definidos por los datos de vértices.

En algunas realizaciones, el sistema operativo 2520 es un sistema operativo Microsoft® Windows® de Microsoft Corporation, un sistema operativo similar a UNIX propietario o un sistema operativo similar a UNIX de código abierto que usa una variante del núcleo Linux. Cuando está en uso la API Direct3D, el sistema operativo 2520 usa un compilador de sombreador de extremo frontal 2524 para compilar cualquier instrucción de sombreador 2512 en HLSL en un lenguaje de sombreador de nivel inferior. La compilación puede ser una compilación justo a tiempo (JIT) o la aplicación puede realizar una precompilación de sombreador. En algunas realizaciones, los sombreadores de alto nivel se compilan a sombreadores de bajo nivel durante la compilación de la aplicación de gráficos 3D 2510.

En algunas realizaciones, el controlador de gráficos de modo de usuario 2526 contiene un compilador de sombreador de extremo trasero 2527 para convertir las instrucciones de sombreador 2512 en una representación específica de hardware. Cuando está en uso la API de OpenGL, las instrucciones de sombreador 2512 en el lenguaje de alto nivel GLSL se pasan a un controlador de gráficos de modo de usuario 2526 para su compilación. En algunas realizaciones, el controlador de gráficos de modo de usuario 2526 usa las funciones de modo de núcleo de sistema operativo 2528 para comunicarse con un controlador de gráficos de modo de núcleo 2529. En algunas realizaciones, el controlador de gráficos de modo de núcleo 2529 se comunica con el procesador de gráficos 2532 para despachar comandos e instrucciones.

Implementaciones de núcleo de IP

Uno o más aspectos de al menos una realización pueden implementarse mediante un código representativo almacenado en un medio legible por máquina que representa y/o define una lógica dentro de un circuito integrado tal como un procesador. Por ejemplo, el medio legible por máquina puede incluir instrucciones que representan una lógica

diversa dentro del procesador. Cuando son leídas por una máquina, las instrucciones pueden hacer que la máquina fabrique la lógica para realizar las técnicas descritas en el presente documento. Tales representaciones, conocidas como "núcleos de IP", son unidades reutilizables de lógica para un circuito integrado que pueden almacenarse en un medio legible por máquina tangible como un modelo de hardware que describe la estructura del circuito integrado. El modelo de hardware puede suministrarse a diversos clientes o instalaciones de fabricación, que cargan el modelo de hardware en máquinas de fabricación que fabrican el circuito integrado. El circuito integrado se puede fabricar de manera que el circuito realiza operaciones descritas en asociación con cualquiera de las realizaciones descritas en el presente documento.

La Figura 26 es un diagrama de bloques que ilustra un sistema de desarrollo de núcleo de IP 2600 que puede usarse para fabricar un circuito integrado para realizar las operaciones de acuerdo con una realización. El sistema de desarrollo de núcleo de IP 2600 puede usarse para generar diseños reutilizables modulares que pueden incorporarse en un diseño más grande o usarse para construir un circuito integrado entero (p. ej., un circuito integrado de SOC). Una instalación de diseño 2630 puede generar una simulación de software 2610 de un diseño de núcleo de IP en un lenguaje de programación de alto nivel (p. ej., C/C++). La simulación de software 2610 puede usarse para diseñar, probar y verificar el comportamiento del núcleo de IP. A continuación, puede crearse o sintetizarse un diseño de nivel de transferencia de registro (RTL) a partir del modelo de simulación 2600. El diseño de RTL 2615 es una abstracción del comportamiento del circuito integrado que modela el flujo de señales digitales entre registros de hardware, que incluyen la lógica asociada realizada usando las señales digitales modeladas. Además de un diseño de RTL 2615, los diseños de nivel inferior en el nivel de lógica o en el nivel de transistores también pueden crearse, diseñarse o sintetizarse. Por lo tanto, los detalles particulares del diseño y simulación inicial pueden variar.

El diseño de RTL 2615, o un equivalente, puede ser sintetizado adicionalmente por la instalación de diseño para obtener un modelo de hardware 2620, que puede estar en un lenguaje de descripción de hardware (HDL) o alguna otra representación de datos de diseño físico. El HDL puede simularse o probarse además para verificar el diseño de núcleo de IP. El diseño de núcleo de IP puede almacenarse para su entrega a una instalación de fabricación de 3^{os} 2665 usando memoria no volátil 2640 (p. ej., disco duro, memoria flash o cualquier medio de almacenamiento no volátil). Como alternativa, el diseño de núcleo de IP puede transmitirse (p. ej., mediante Internet) a través de una conexión alámbrica 2650 o conexión inalámbrica 2660. La instalación de fabricación 2665 puede fabricar entonces un circuito integrado que se basa, al menos en parte, en el diseño de núcleo de IP. El circuito integrado fabricado puede configurarse para realizar operaciones de acuerdo con al menos una realización descrita en el presente documento.

La Figura 27 es un diagrama de bloques que ilustra un circuito integrado de sistema en un chip 2700 ilustrativo que puede fabricarse usando uno o más núcleos de IP, de acuerdo con una realización. El circuito integrado ilustrativo incluye uno o más procesadores de aplicación 2705 (p. ej., las CPU), al menos un procesador de gráficos 2710, y puede incluir adicionalmente un procesador de imágenes 2715 y/o un procesador de vídeo 2720, cualquiera de los cuales puede ser un núcleo de IP modular desde las mismas o múltiples diferentes instalaciones de diseño. El circuito integrado incluye lógica de periférico o de bus que incluye un controlador de USB 2725, controlador de UART 2730, un controlador de SPI/SDIO 2735 y un controlador de I²S/I²C 2740. Adicionalmente, el circuito integrado puede incluir un dispositivo de visualización 2745 acoplado a uno o más de un controlador de interfaz multimedia de alta definición (HDMI) 2750 y una interfaz de visualización de interfaz de procesador de industria móvil (MIPI) 2755. El almacenamiento puede proporcionarse por un subsistema de memoria flash 2760 que incluye la memoria flash y un controlador de memoria flash. La interfaz de memoria se puede proporcionar mediante un controlador de memoria 2765 para el acceso a dispositivos de memoria SDRAM o SRAM. Algunos circuitos integrados incluyen adicionalmente un motor de seguridad integrado 2770.

Además, se pueden incluir otras lógicas y circuitos en el procesador del circuito integrado 2700, incluyendo procesadores/núcleos de gráficos adicionales, controladores de interfaz periféricos o núcleos de procesador de fin general.

El término "acoplado" puede usarse en el presente documento para referirse a cualquier tipo de relación, directa o indirecta, entre los componentes en cuestión, y puede aplicarse a conexiones eléctricas, mecánicas, de fluidos, ópticas, electromagnéticas, electromecánicas u otras. Además, los términos "primero", "segundo", etc., pueden usarse en el presente documento únicamente para facilitar el análisis y no tienen un significado temporal o cronológico particular a menos que se indique lo contrario. Adicionalmente, se entiende que los artículos indefinidos "un" o "una" conllevan el significado de "uno o más" o "al menos uno".

Como se usa en esta solicitud y en las reivindicaciones, una lista de elementos unidos por la expresión "uno o más de" puede significar cualquier combinación de los términos enumerados. Por ejemplo, las expresiones "uno o más de A, B o C" pueden significar A, B, C; A y B; A y C; B y C; o A, B y C.

Las realizaciones se han descrito anteriormente con referencia a realizaciones específicas. Sin embargo, los expertos en la técnica entenderán que se pueden realizar diversas modificaciones y cambios sin apartarse del alcance de las realizaciones tal como se establece en las reivindicaciones adjuntas. Por consiguiente, la descripción y los dibujos anteriores deben considerarse en un sentido ilustrativo más que restrictivo.

REIVINDICACIONES

1. Un aparato para mostrar (704, 900) una pluralidad de resoluciones, que comprende:
una pantalla (704, 900) para presentar vídeo;
5 al menos un panel acoplado comunicativamente a la pantalla (704, 900), que comprende:
una memoria intermedia de fondo para almacenar un plano de fondo (706, 920) en una primera resolución;
una memoria intermedia del plano (702) de interés (708, 922) para almacenar un plano (702) de interés en una
segunda resolución; y
un compositor para componer el plano (702) de interés (708, 922) encima del plano de fondo (706, 920),
10 el aparato comprende además un renderizador foveado (607) con desvío de movimiento para identificar un área de
enfoque futura basándose en la información de enfoque e información de movimiento de un usuario (1218) y para rotar
un área de enfoque basándose en la información de movimiento, en donde el renderizador foveado (607) produce
superficies planas (702) separadas para el campo de visión cubierto por la fovea del usuario y para el fondo, en donde
la información de enfoque incluye uno o más de un punto focal, el área de enfoque, la posición del ojo, el movimiento
15 del ojo, el tamaño de la pupila, la dilatación de la pupila, la profundidad de enfoque, la DOF, un punto focal de contenido,
un objeto de enfoque de contenido y un área de enfoque de contenido, y la información de movimiento se usa para
ajustar los parámetros de rendimiento del sistema de gráficos basándose en al menos uno de
el usuario (1218) que mueve su cuerpo,
donde una aplicación quiere que el usuario (1218) gire la cabeza, los ojos y/o el cuerpo, y/o
20 donde se predice que el usuario (1218) girará la cabeza, los ojos y/o el cuerpo en el futuro.
2. El aparato según la reivindicación 1, en donde la segunda resolución es mayor que la primera resolución.
3. El aparato según la reivindicación 1, que comprende además:
25 un rastreador de mirada (906) para proporcionar una coordenada (x, y) al compositor para localizar el plano (702) de
interés (708, 922) con respecto a la pantalla (704, 900).
4. El aparato según la reivindicación 3, en donde la pantalla (704, 900) comprende una pantalla montada en la cabeza
(704, 900), HMD (900).
5. Un método para mostrar (704, 900) una pluralidad de resoluciones, que comprende:
almacenar en una memoria intermedia de fondo un plano de fondo (706, 920) en una primera resolución;
almacenar en una memoria intermedia de un plano (702) de interés (708, 922) un plano (702) de interés en una
segunda resolución;
35 componer el plano (702) de interés (708, 922) encima del plano de fondo (706, 920) para crear medios compuestos;
y
mostrar los medios compuestos,
el método comprende además identificar un área de enfoque futura basándose en información de enfoque e
información de movimiento de un usuario (1218) y rotar un área de enfoque basándose en la información de
40 movimiento, en donde la información de enfoque incluye uno o más de un punto focal, el área de enfoque, la posición
del ojo, el movimiento del ojo, el tamaño de la pupila, la dilatación de la pupila, la profundidad de enfoque, DOF, un
punto focal de contenido, un objeto de enfoque de contenido y un área de enfoque de contenido, y la información de
movimiento se usa para ajustar los parámetros de rendimiento del sistema de gráficos basándose en al menos uno de
el usuario (1218) que mueve su cuerpo,
45 donde una aplicación quiere que el usuario (1218) gire la cabeza, los ojos y/o el cuerpo, y/o
donde se predice que el usuario (1218) girará la cabeza, los ojos y/o el cuerpo en el futuro.
6. El método según la reivindicación 5, en donde la segunda resolución es mayor que la primera resolución.
7. El método según la reivindicación 5, que comprende, además:
50 rastrear la mirada de un espectador con respecto a una pantalla (704, 900) para proporcionar una coordenada (x, y)
al compositor para localizar el plano (702) de interés (708, 922) con respecto a la pantalla (704, 900).
8. Al menos un medio legible por máquina que comprende una pluralidad de instrucciones que, en respuesta a su
55 ejecución en un dispositivo informático, hacen que el dispositivo informático lleve a cabo un método según una
cualquiera de las reivindicaciones 5 a 7.
9. Un sistema (100, 495, 600, 1100, 1212, 1300, 1400, 1500, 1600, 2500, 2520, 2600) para mostrar (704, 900) una
pluralidad de resoluciones, comprendiendo el sistema:
60 un aparato según la reivindicación 1, en donde la pantalla (704, 900) del aparato es
una pantalla montada en la cabeza (704, 900), HMD (900),
un rastreador de mirada (906) para rastrear una mirada para identificar una región de interés, y
un cliente (902) acoplado comunicativamente a la HMD (900), y
en donde el cliente (902) debe enviar a la HMD (900) una vista de pantalla completa a baja resolución y una vista de
65 pantalla parcial a alta resolución que comprende la región de interés.

10. El sistema (100, 495, 600, 1100, 1212, 1300, 1400, 1500, 1600, 2500, 2520, 2600) según la reivindicación 9, en donde el cliente (902) y la HMD (900) están acoplados de forma inalámbrica.

5 11. El sistema (100, 495, 600, 1100, 1212, 1300, 1400, 1500, 1600, 2500, 2520, 2600) según la reivindicación 9, en donde la resolución baja se muestrea mediante interpolación para igualar una resolución de visualización, mientras que la segunda resolución coincide con la resolución de visualización.

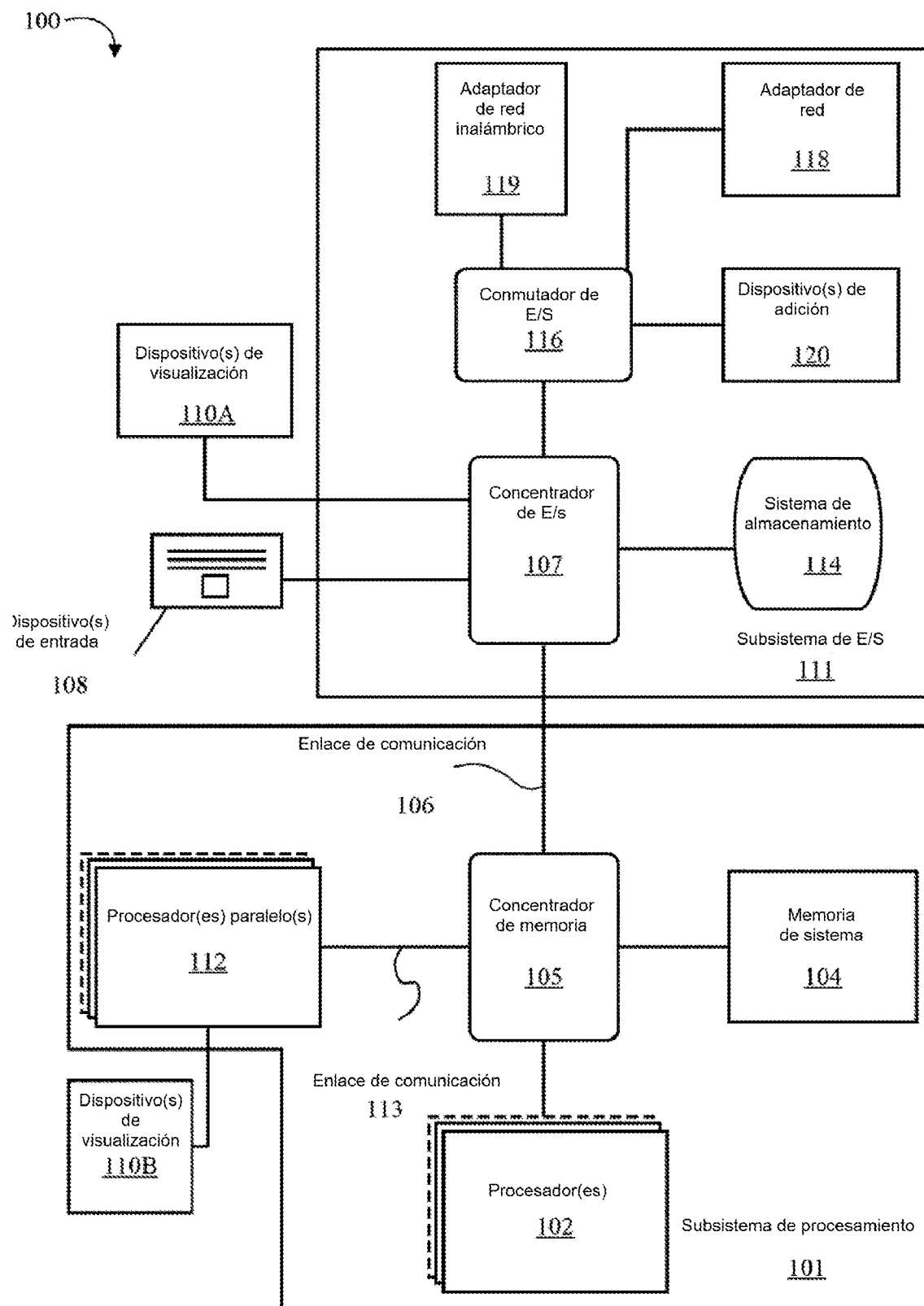


FIG. 1

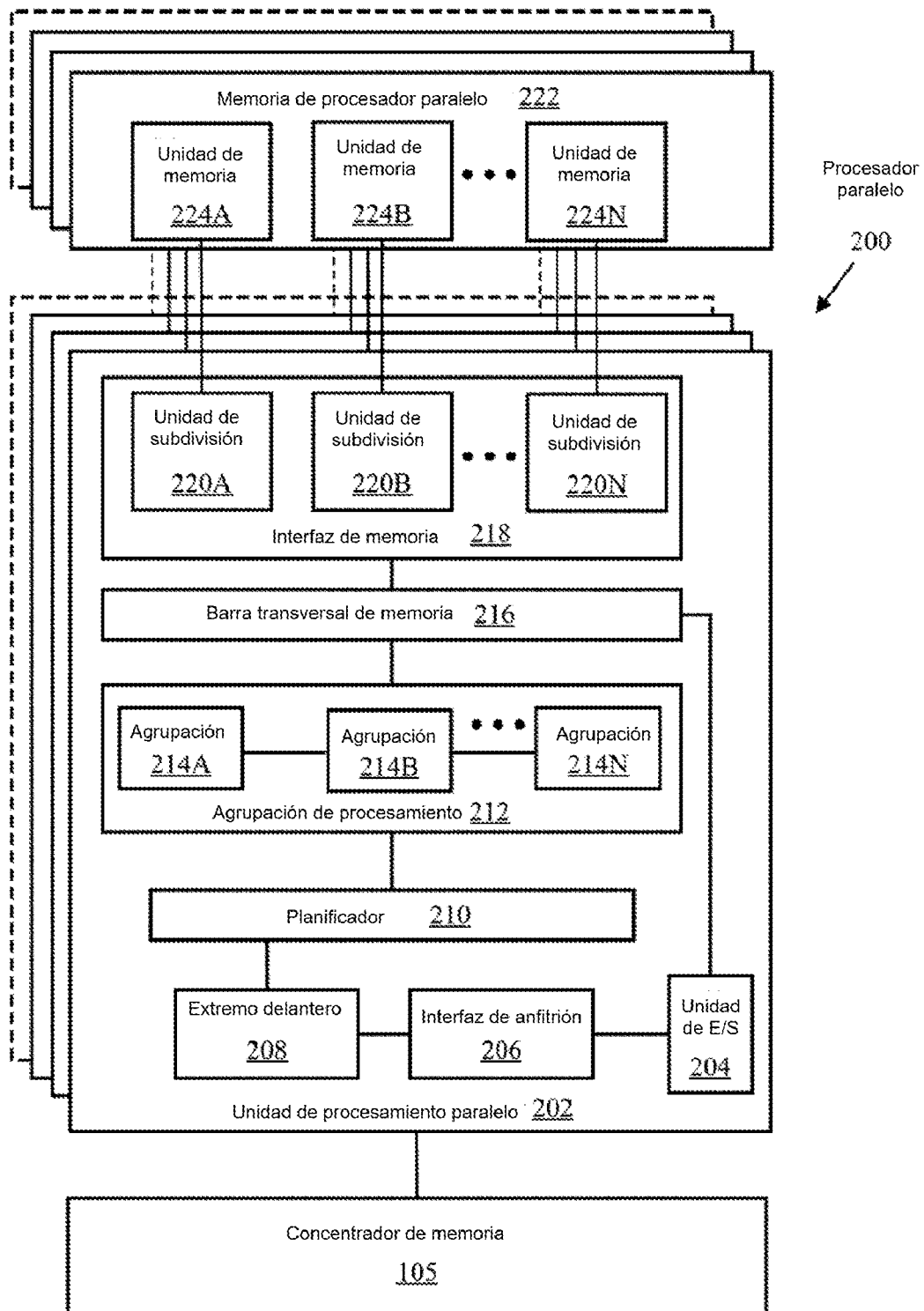


FIG. 2A

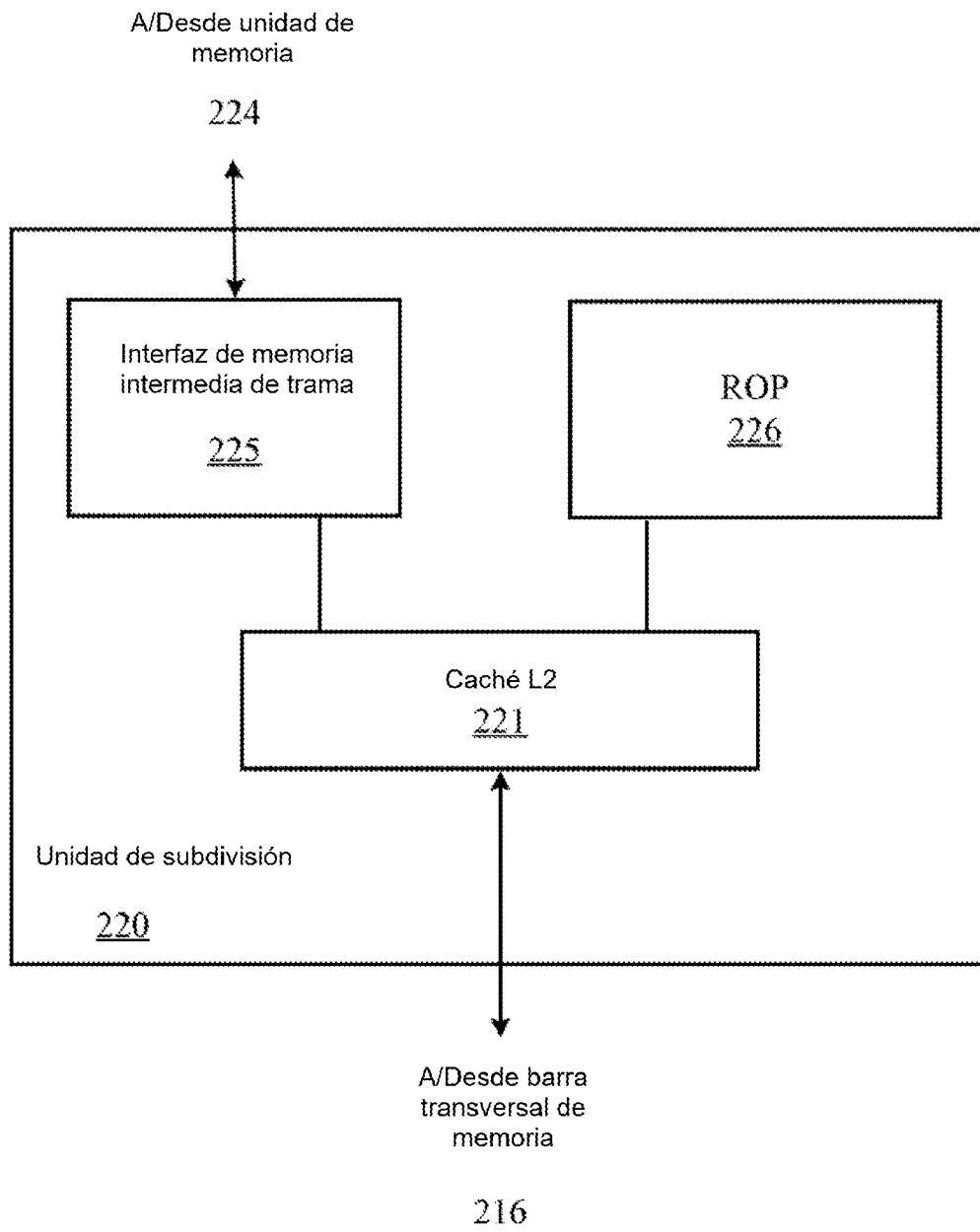


FIG. 2B

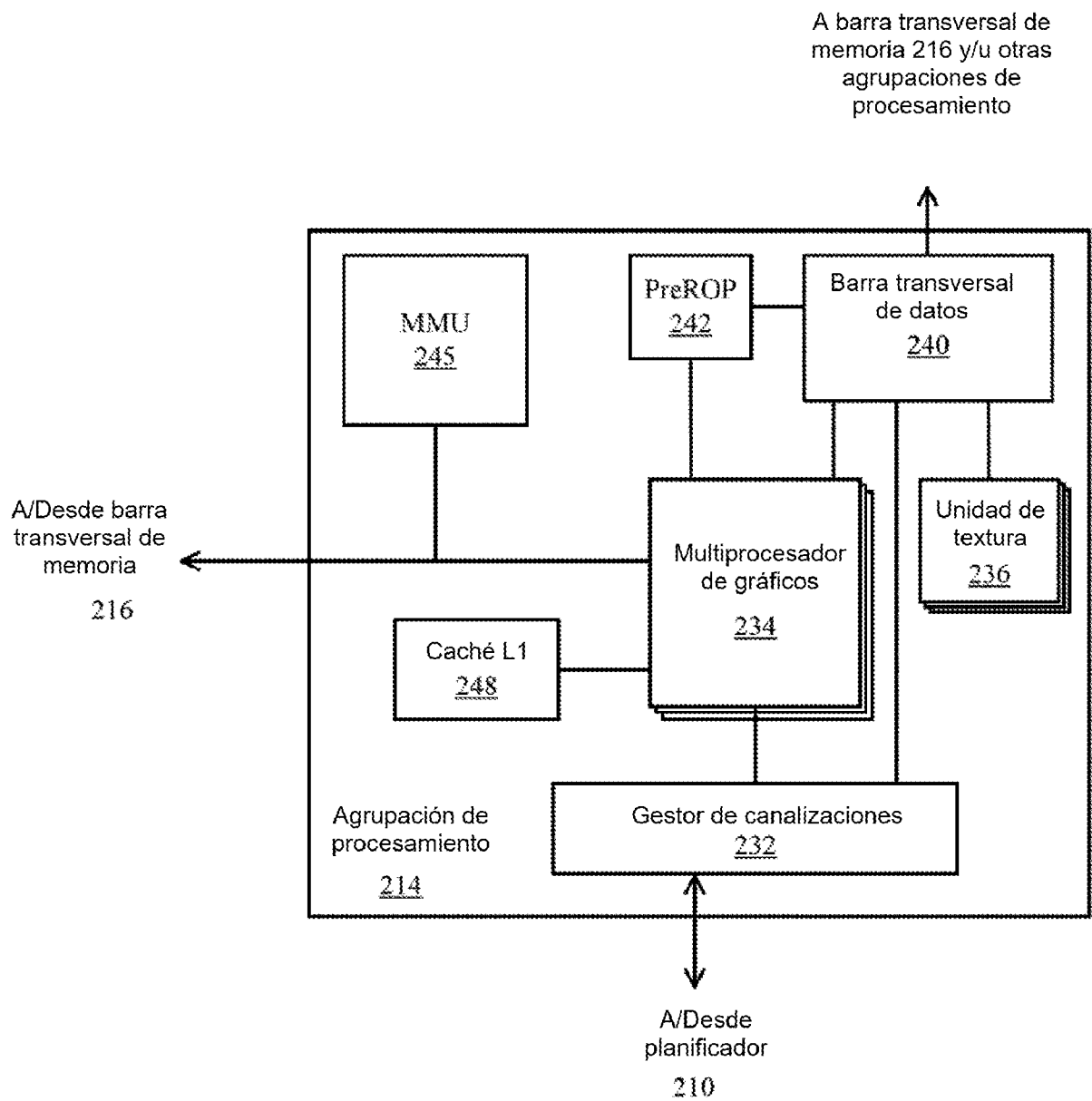


FIG. 2C

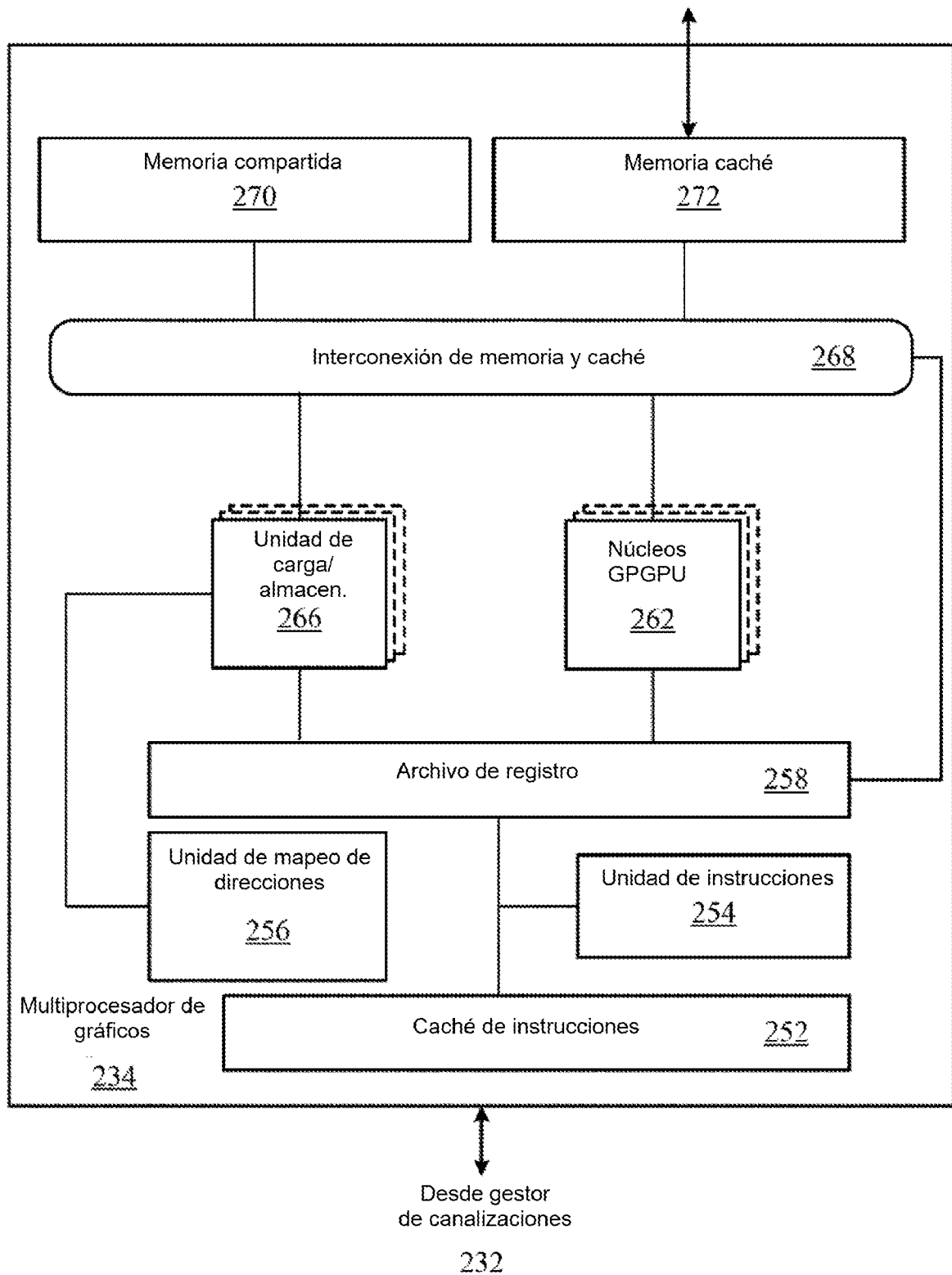


FIG. 2D

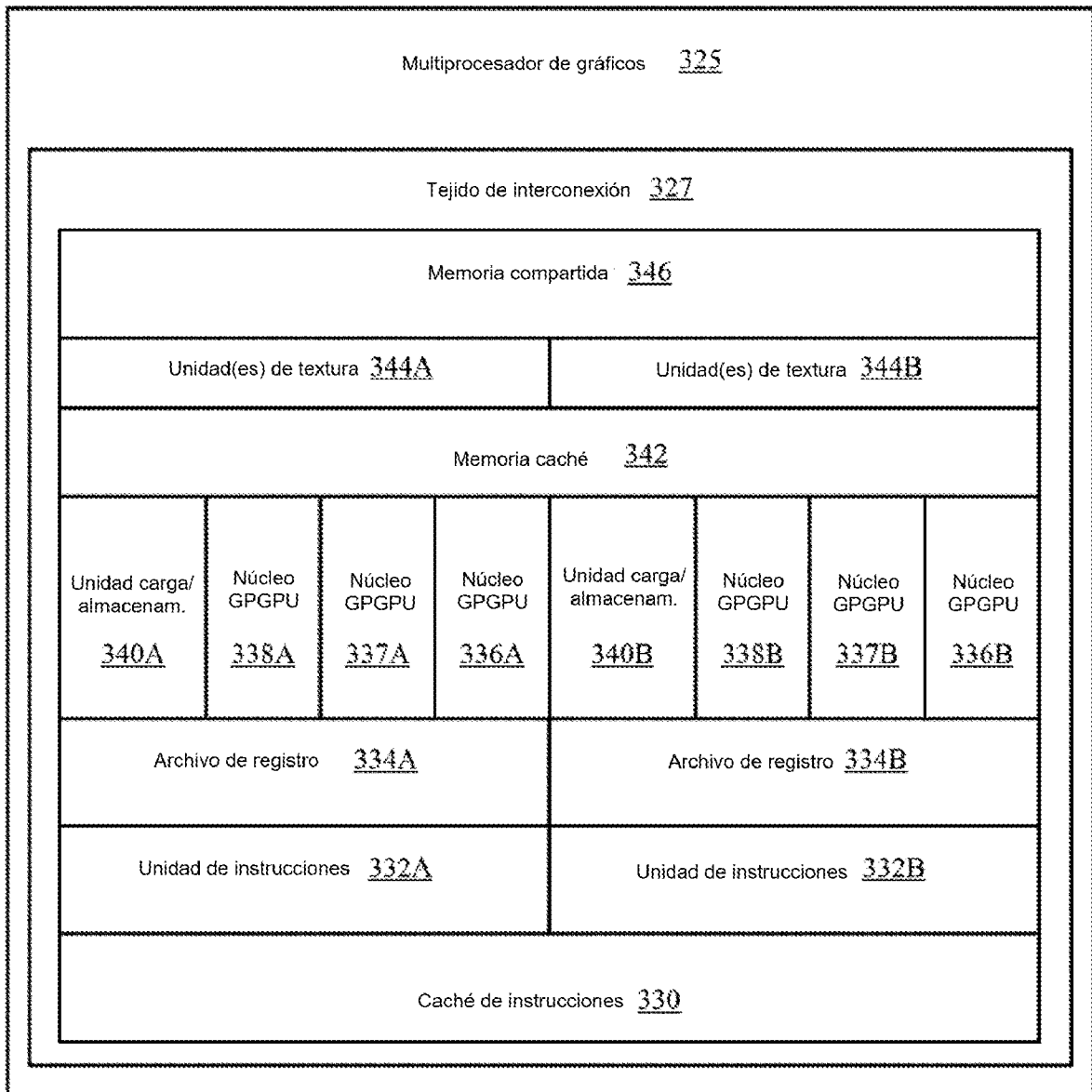


FIG. 3A

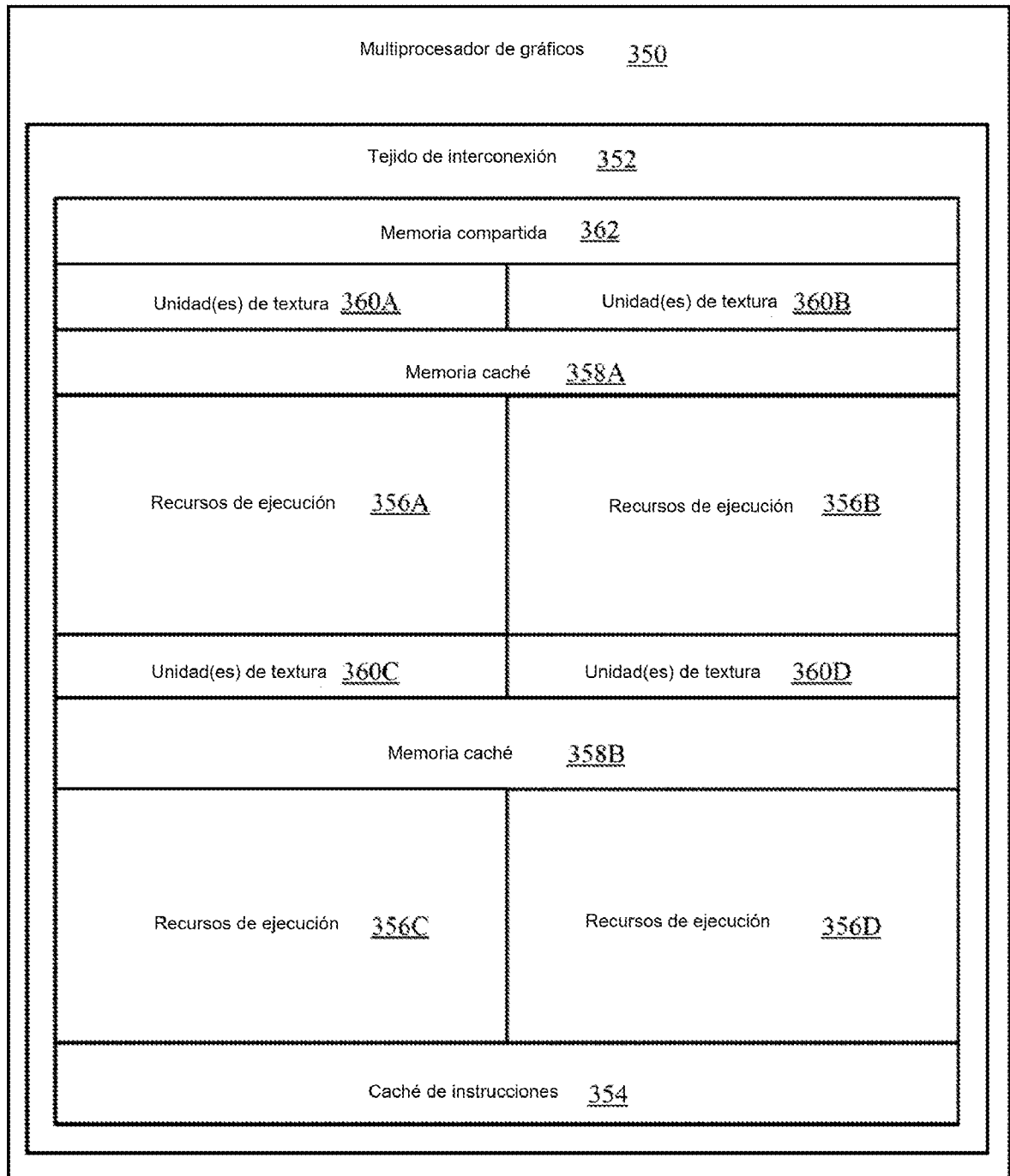


FIG. 3B

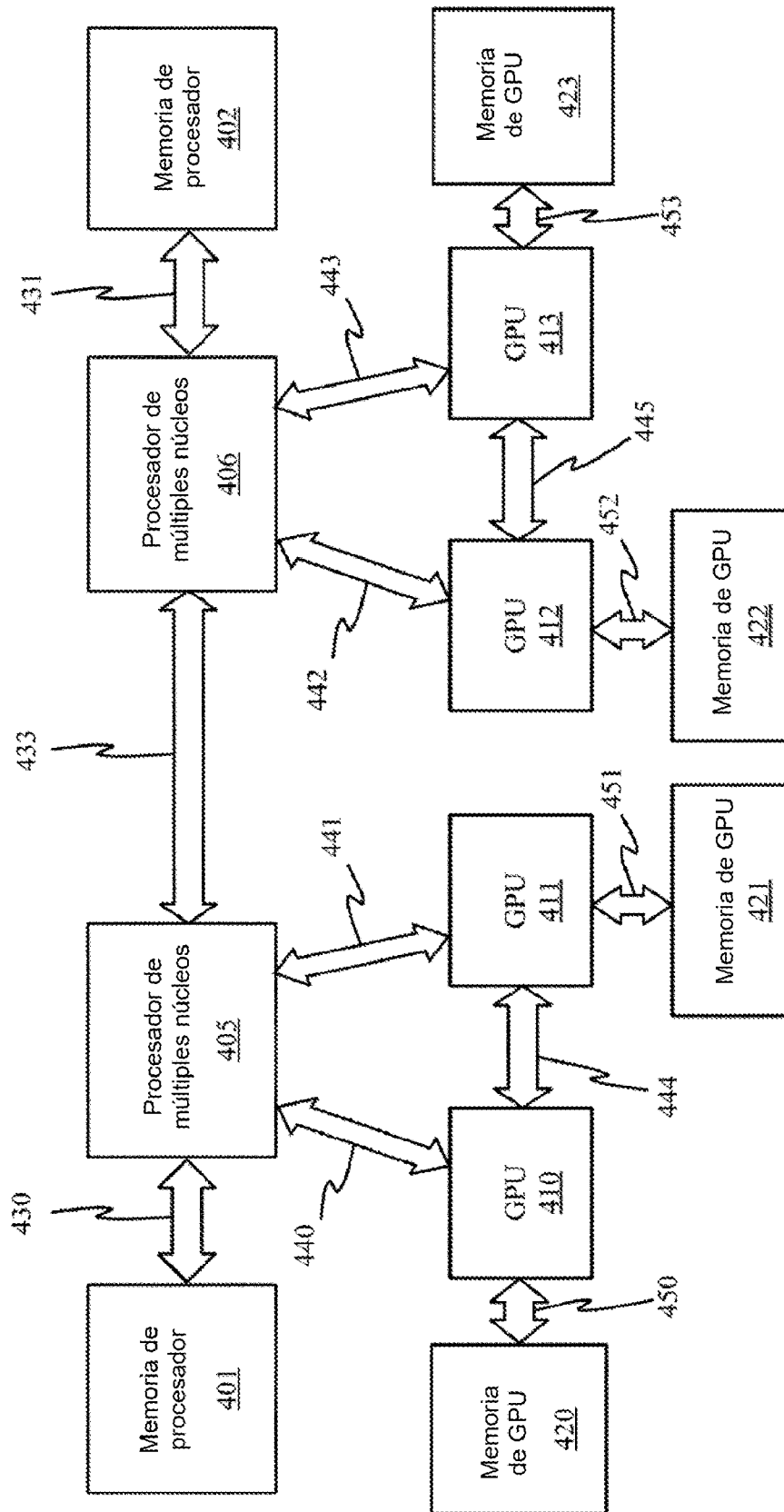


FIG. 4A

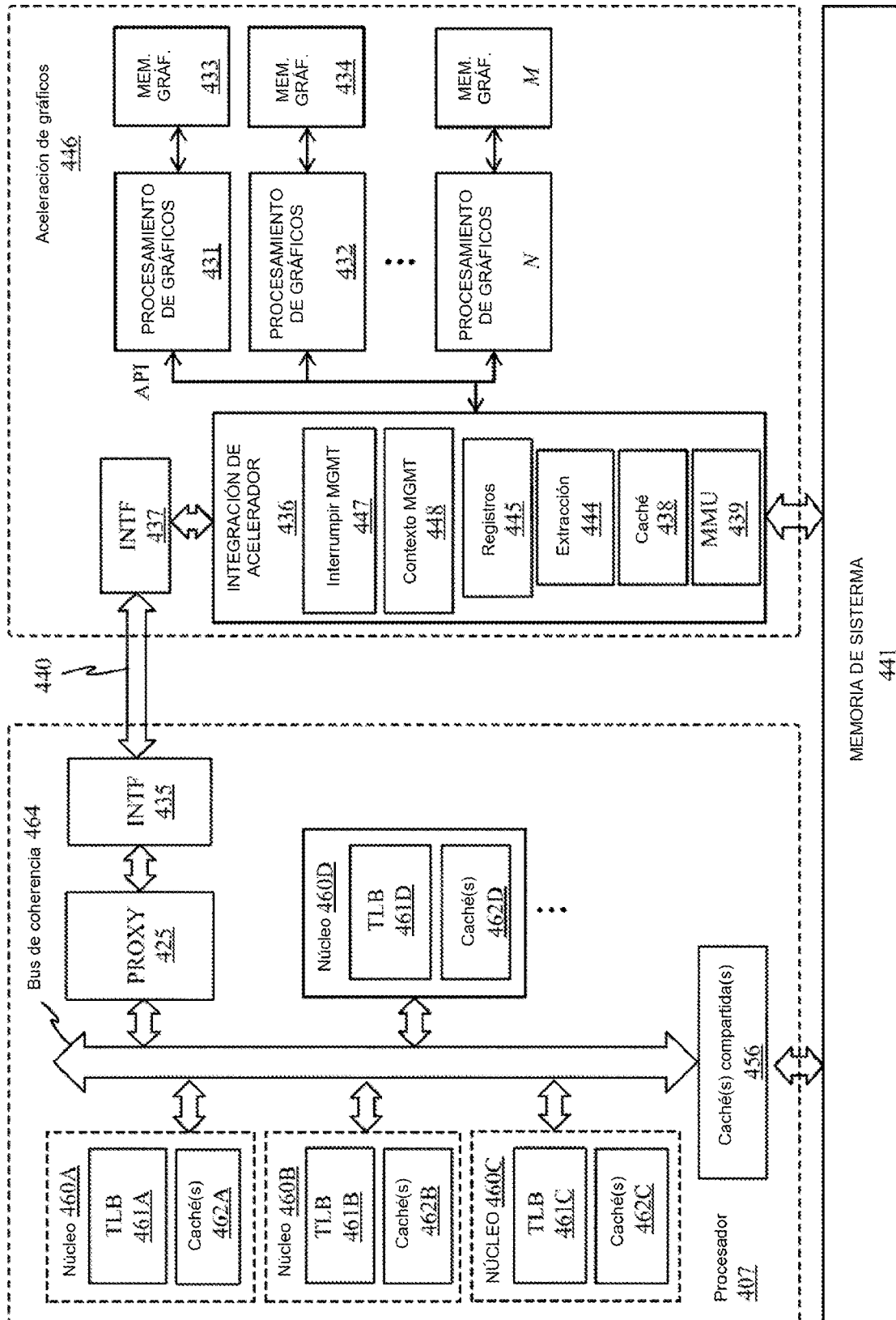


FIG. 4B

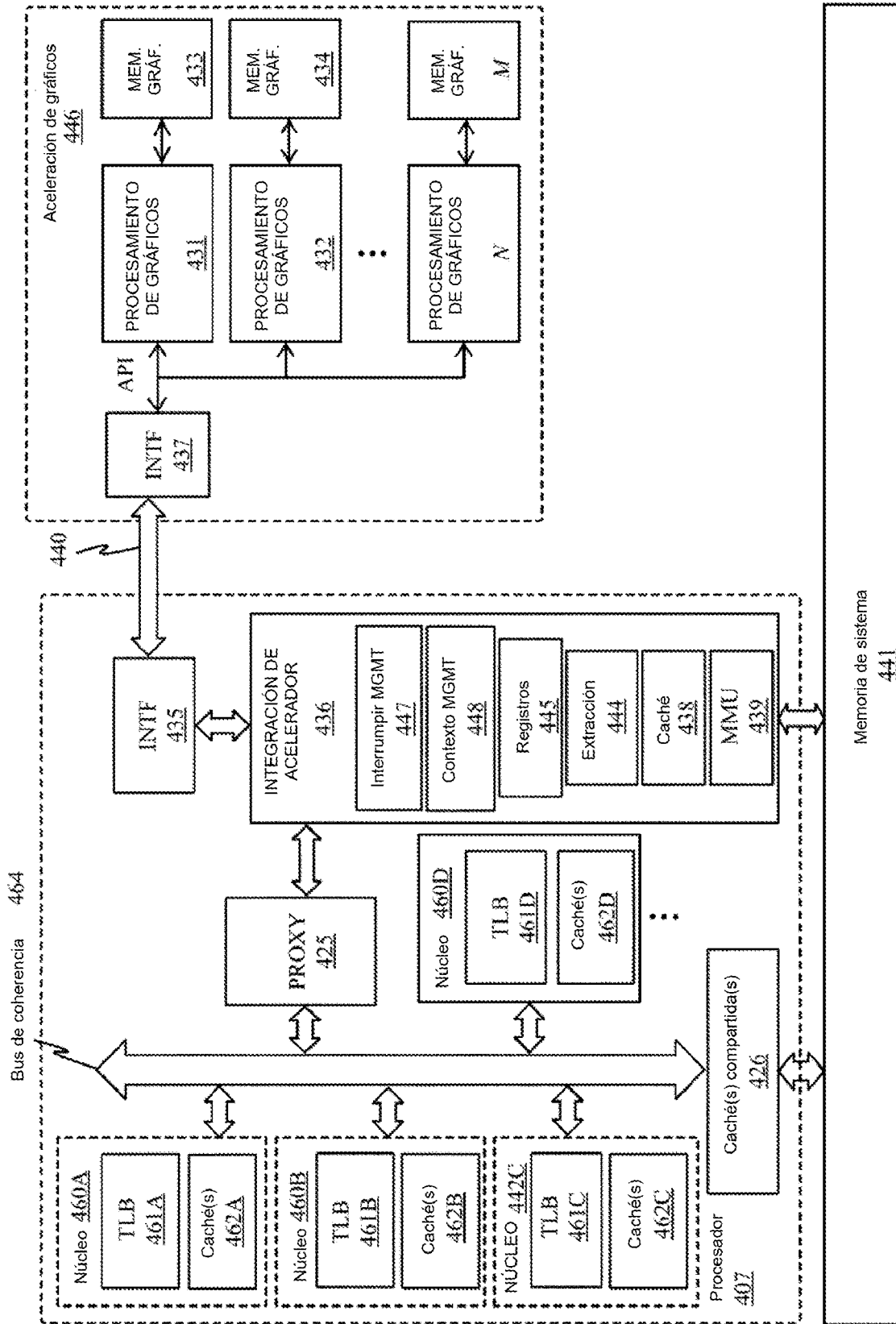


FIG. 4C

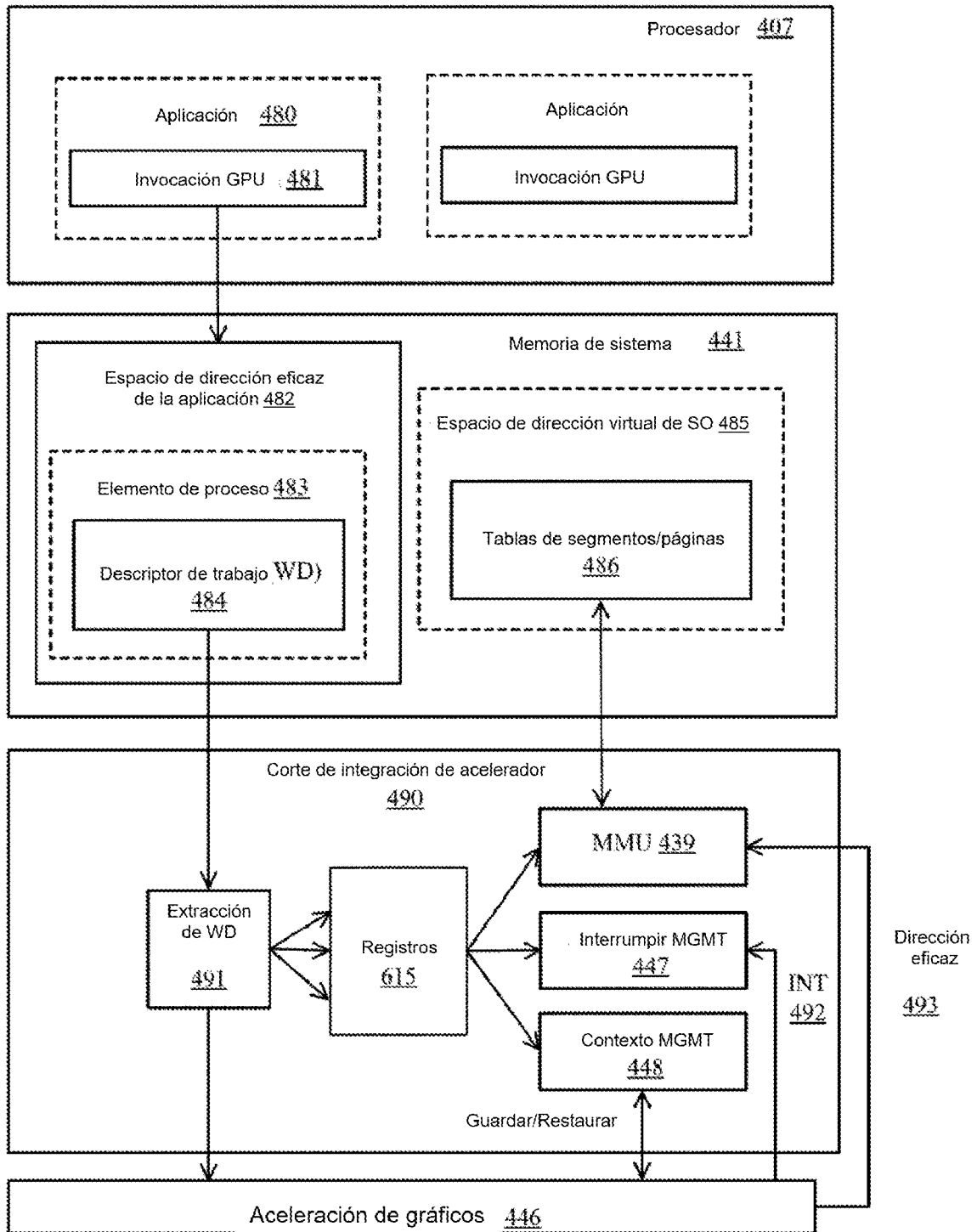


FIG. 4D

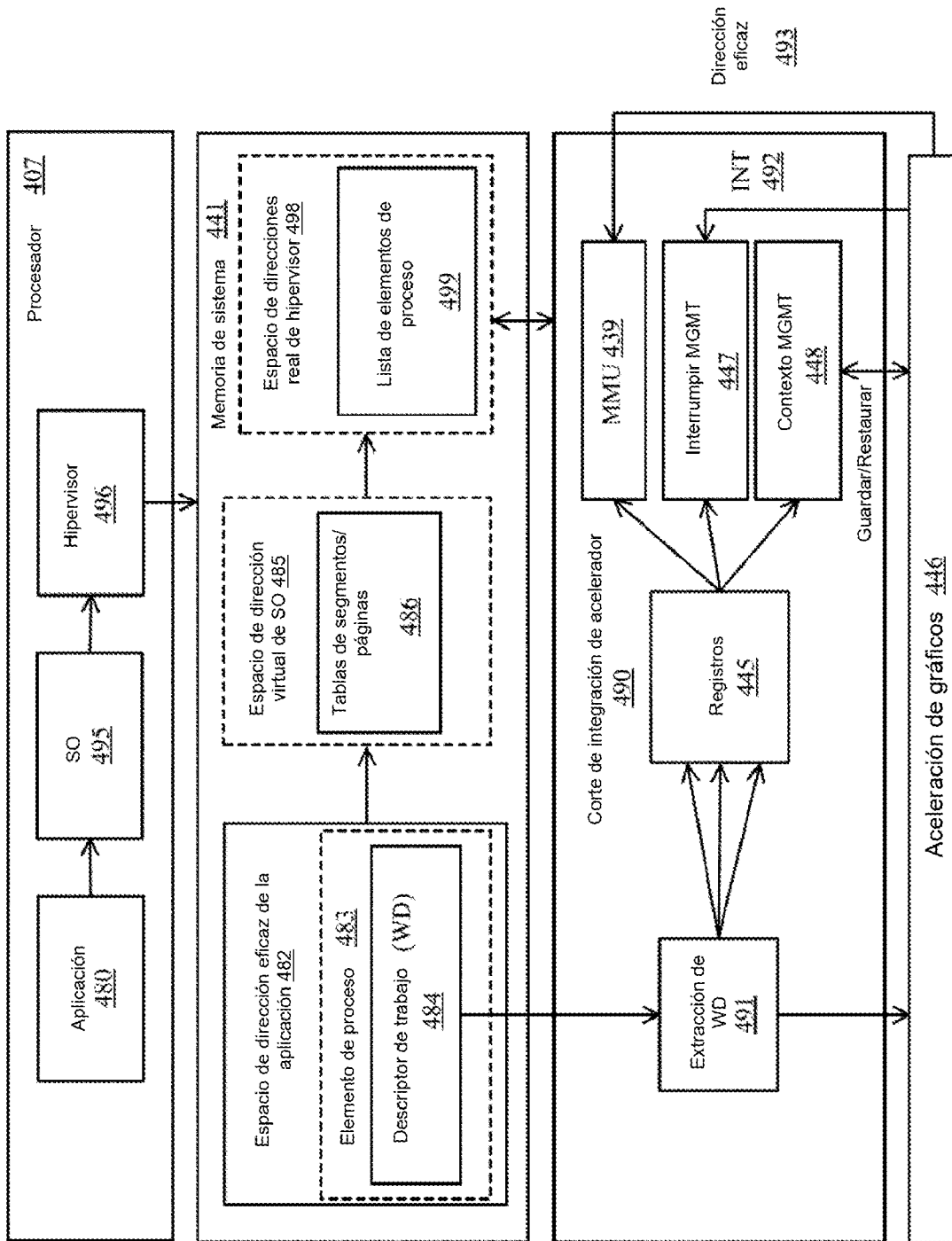


FIG. 4E

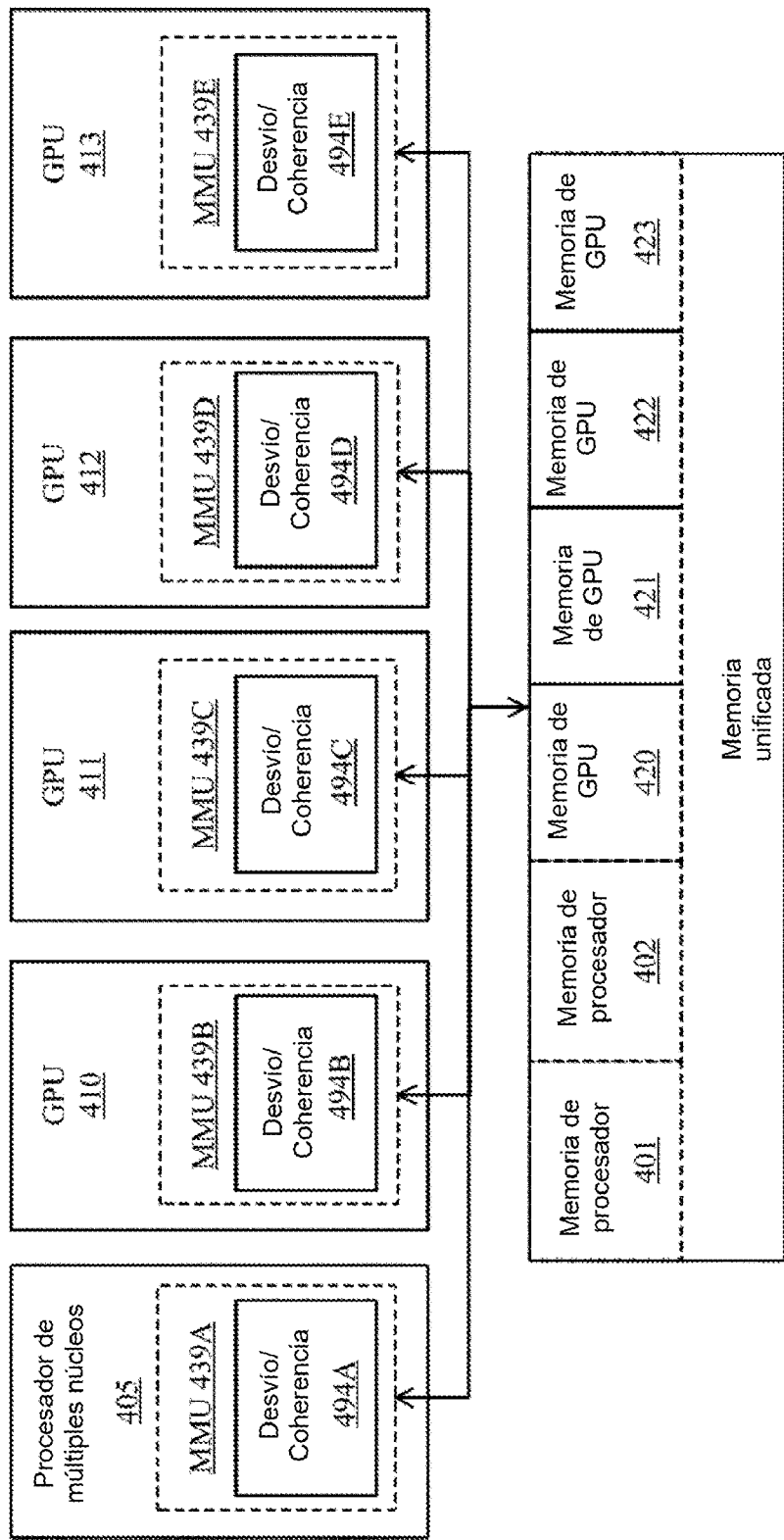


FIG. 4F

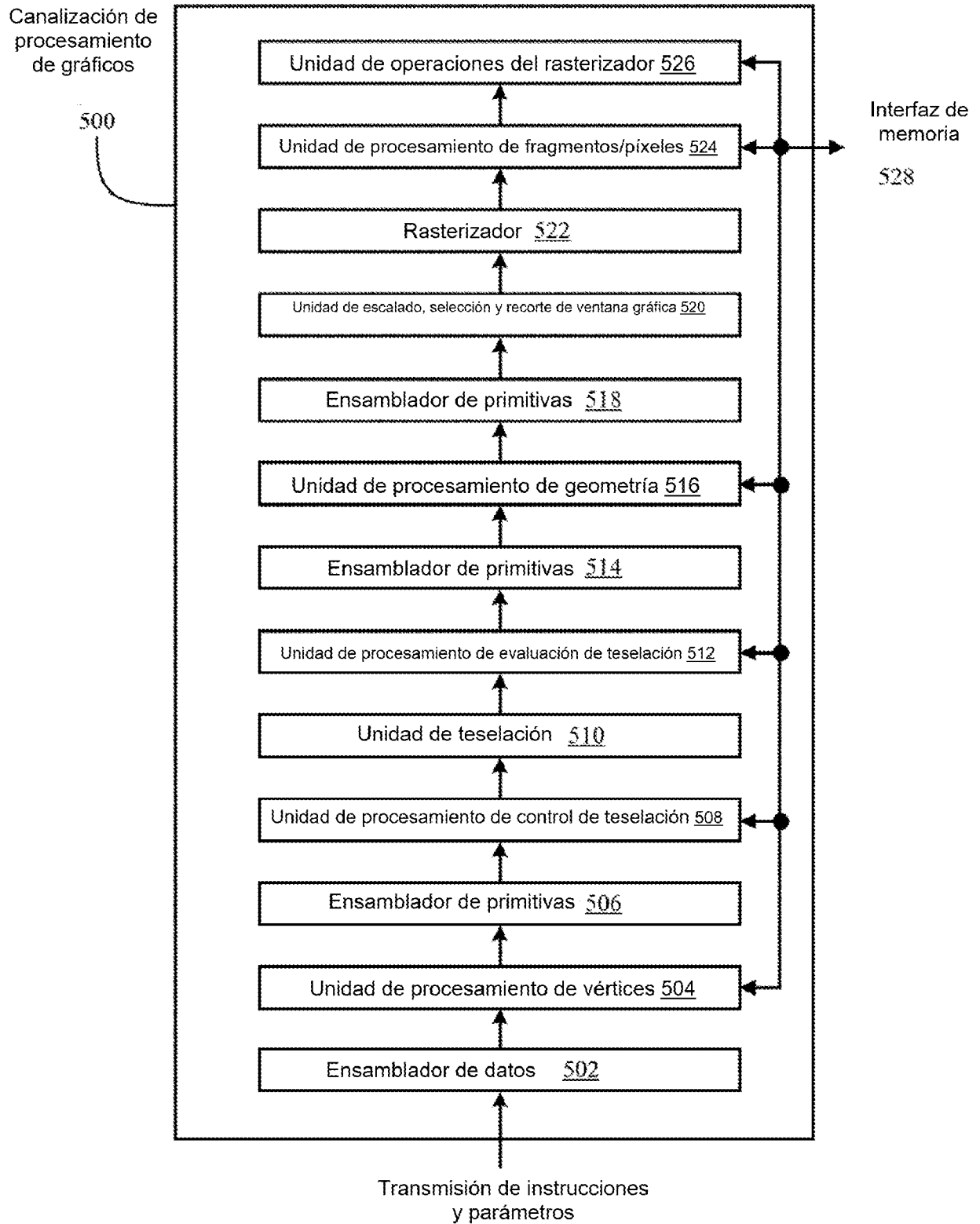


FIG. 5

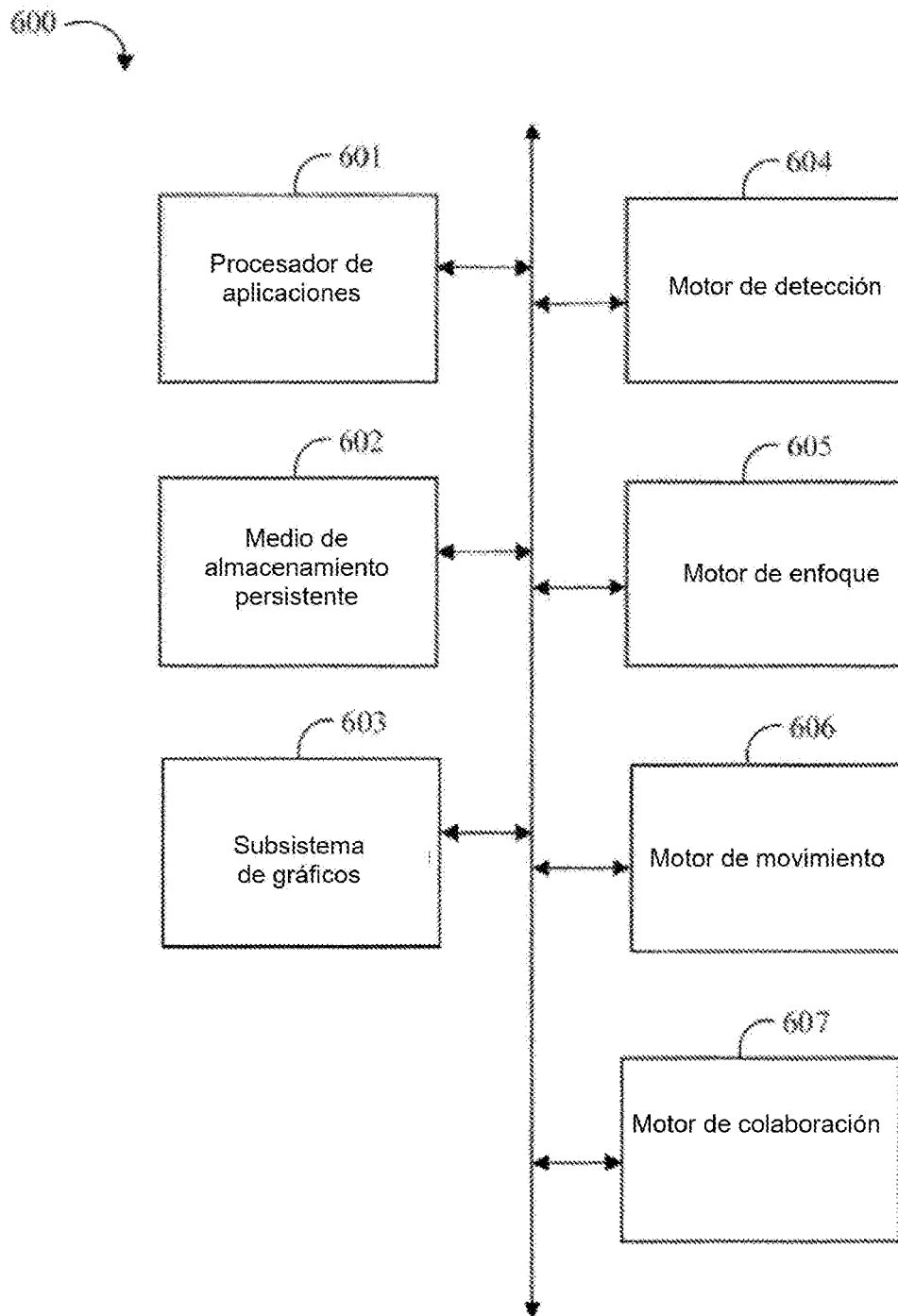


FIG. 6A

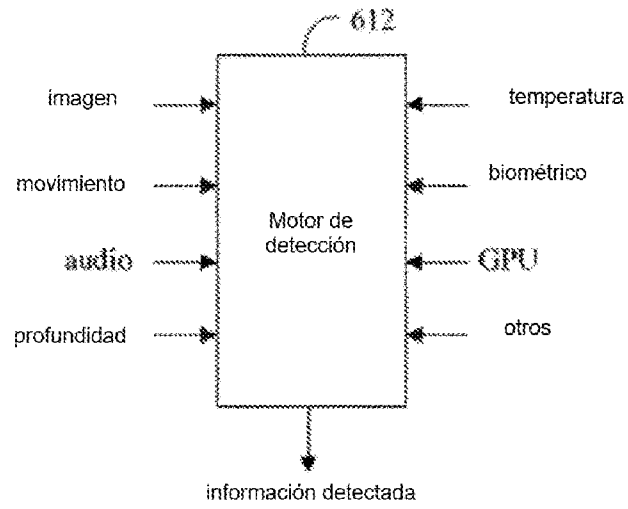


FIG. 6B

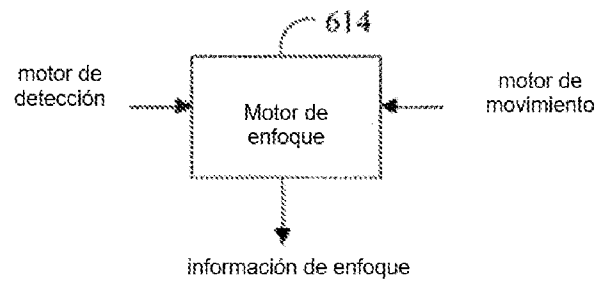


FIG. 6C

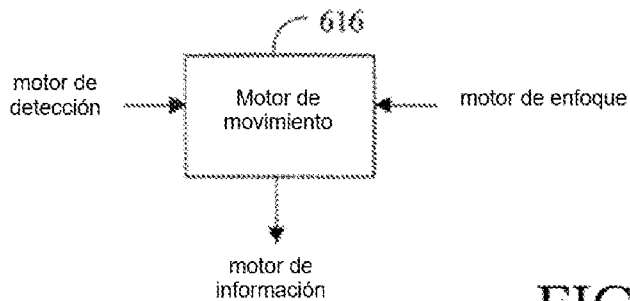


FIG. 6D

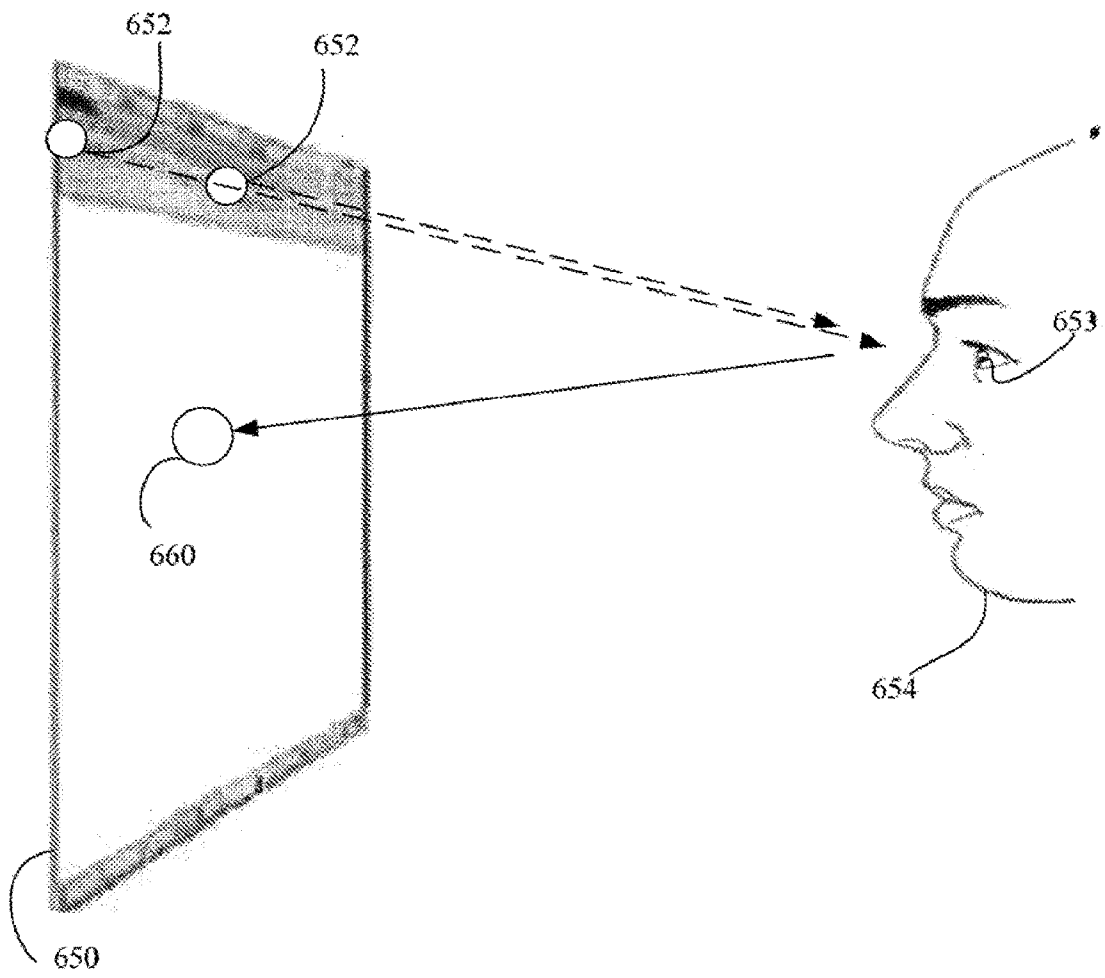


FIG. 6E

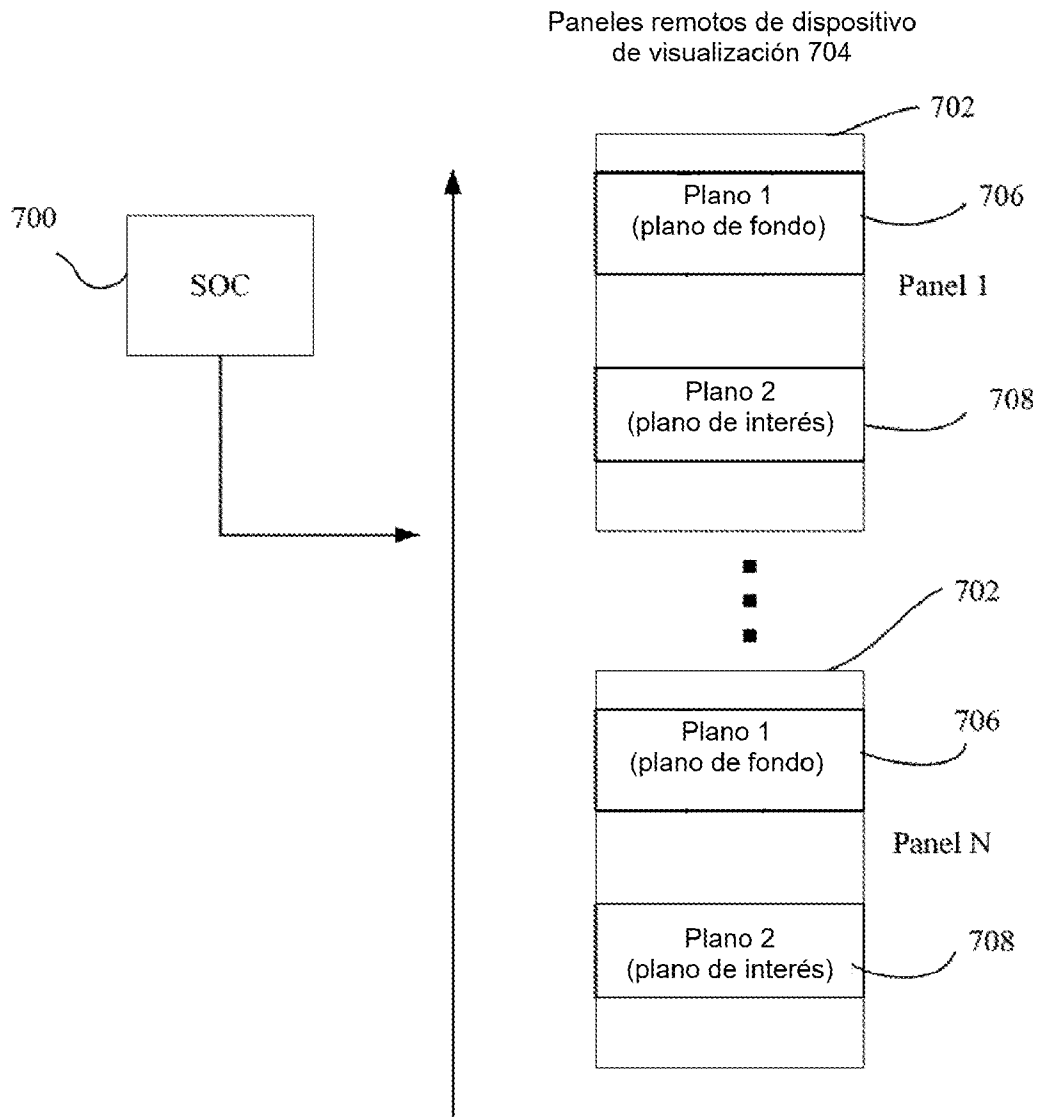


FIG. 7A

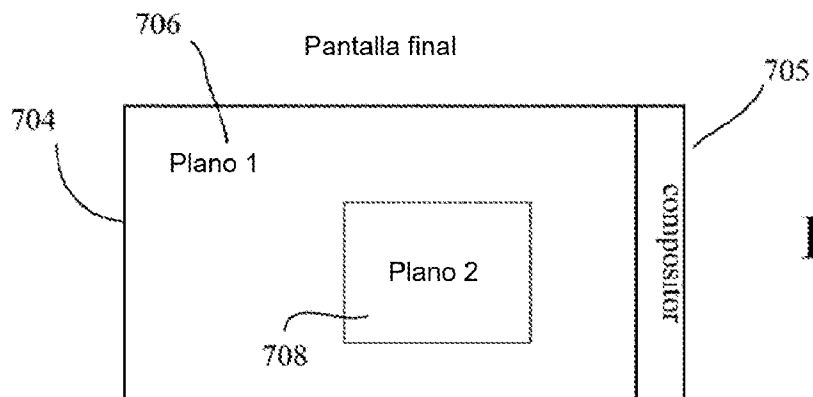


FIG. 7B

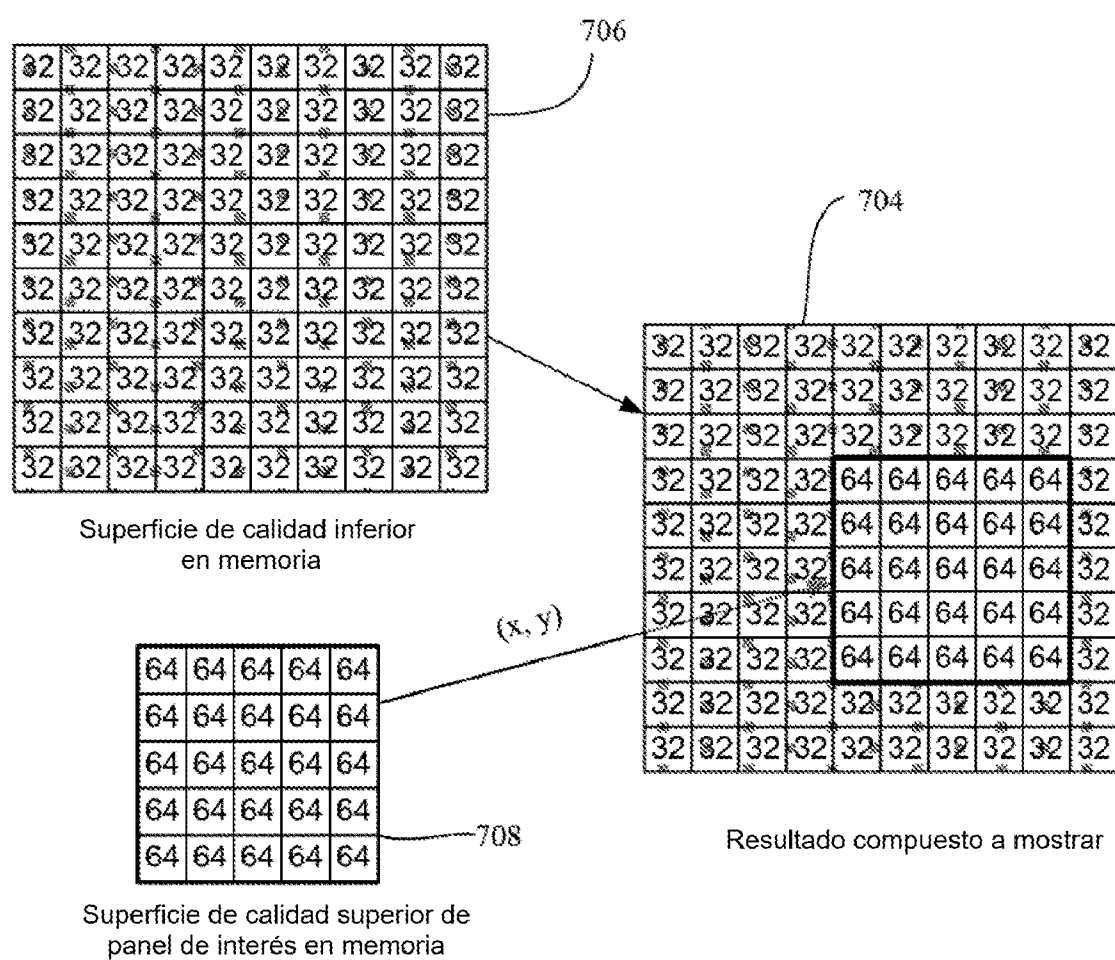
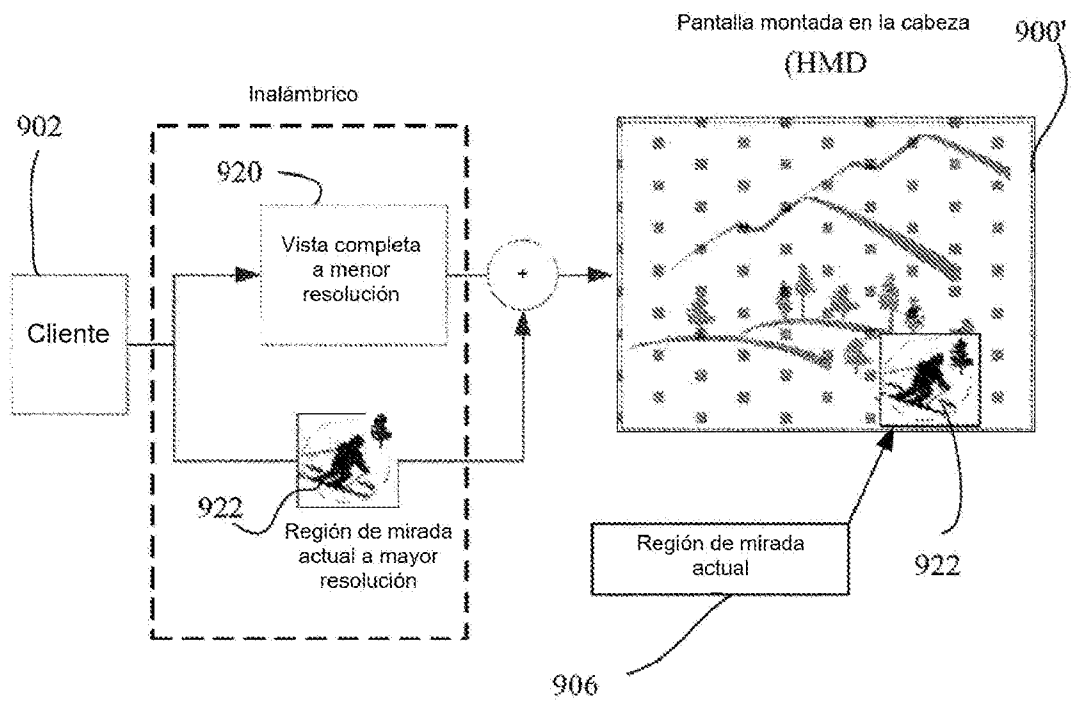
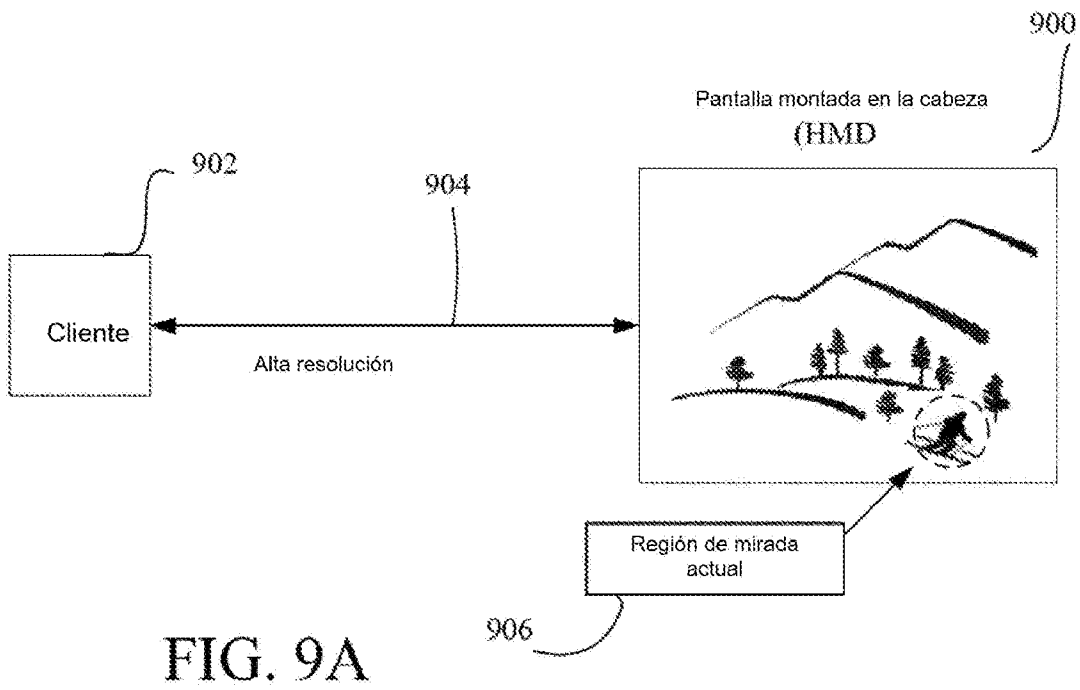


FIG. 8

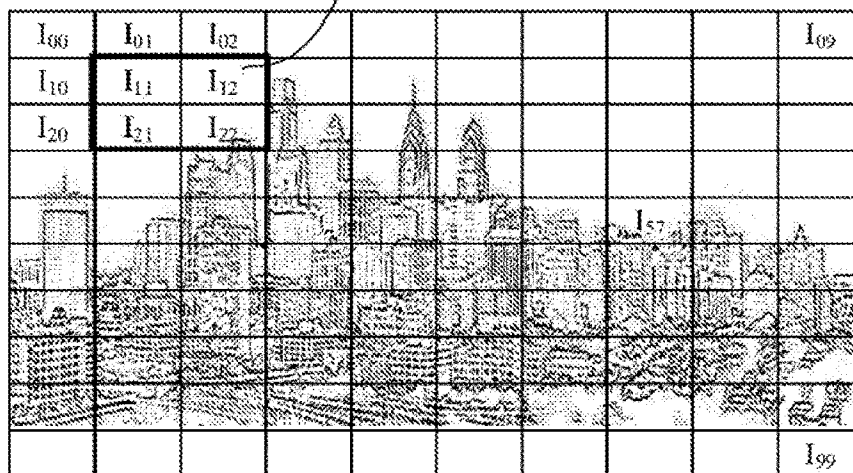




1000

FIG. 10A

1002



1000

FIG. 10B

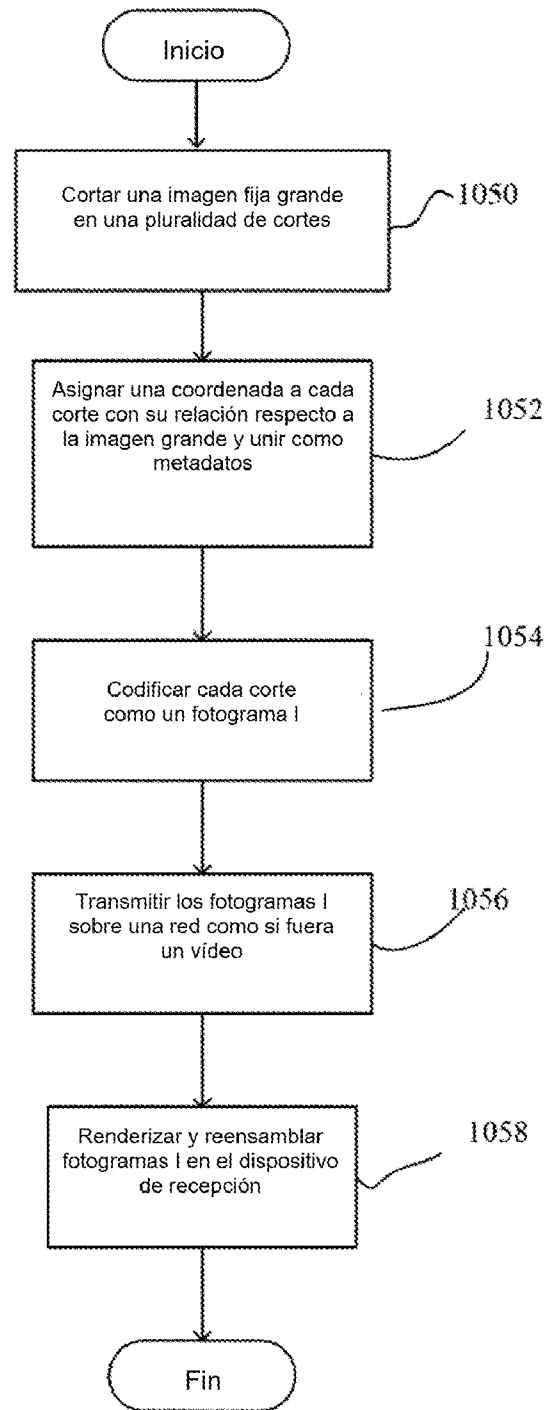


FIG. 10C

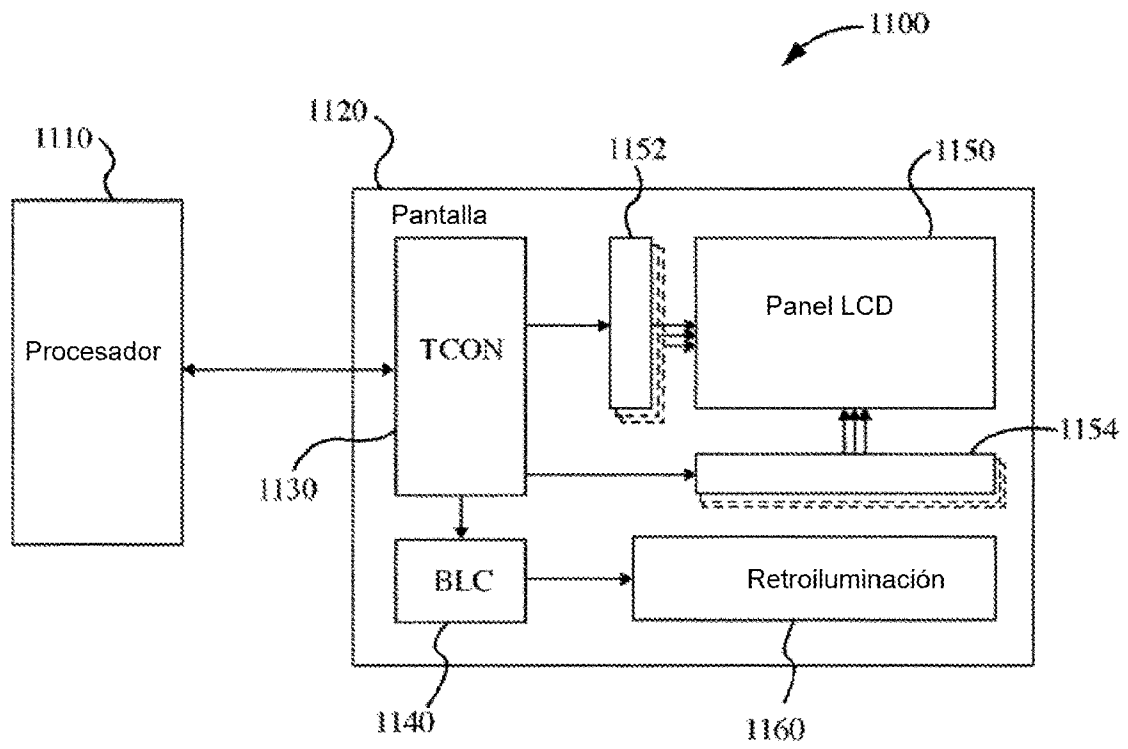


FIG. 11

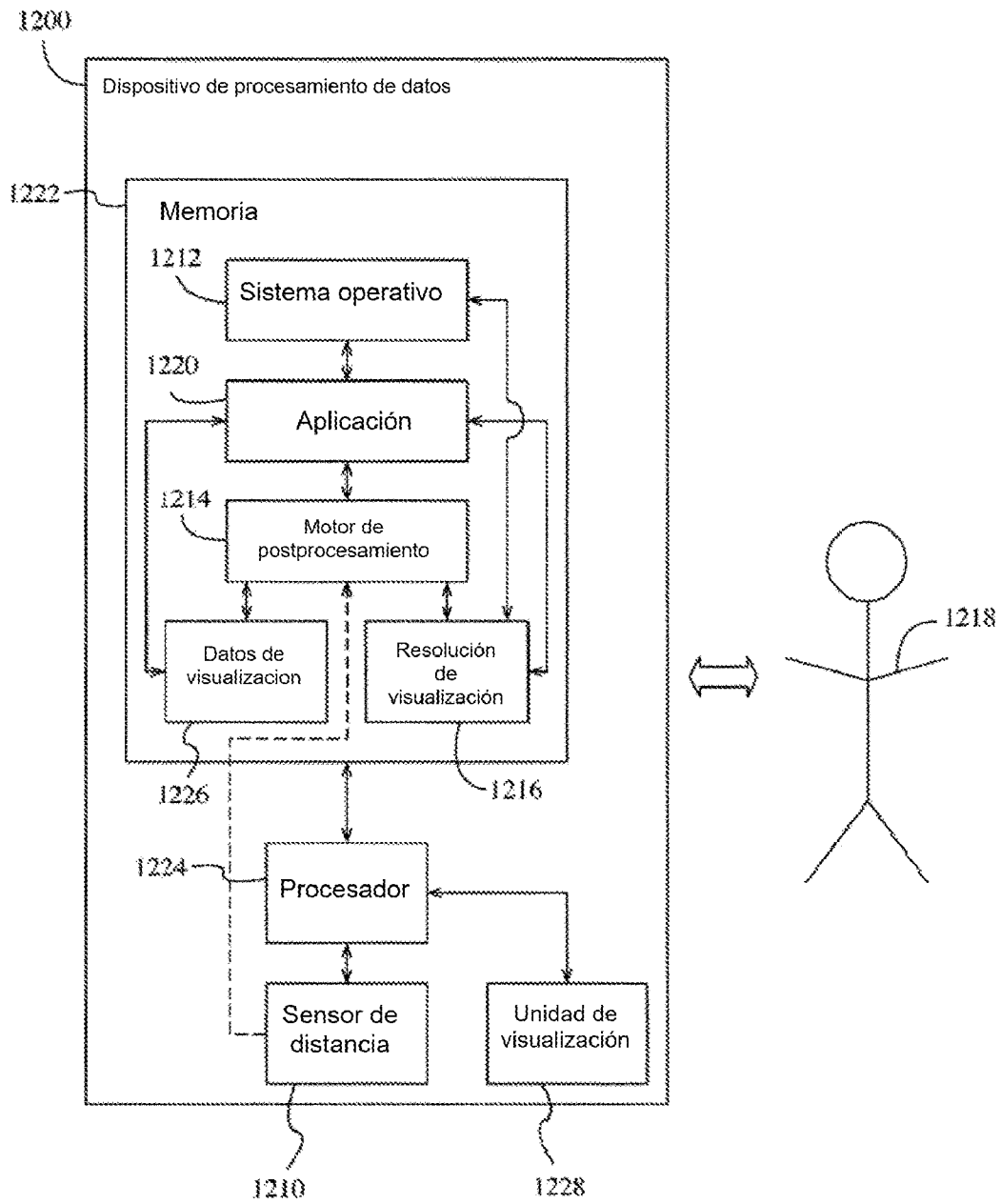


FIG. 12A

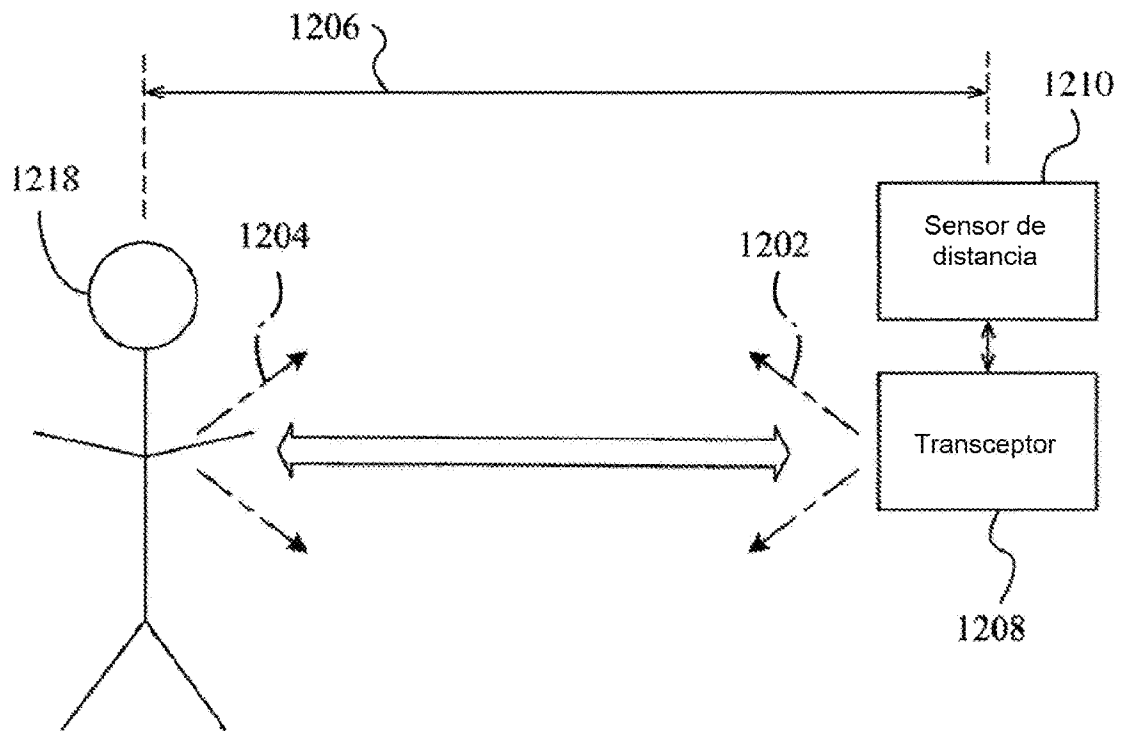


FIG. 12B

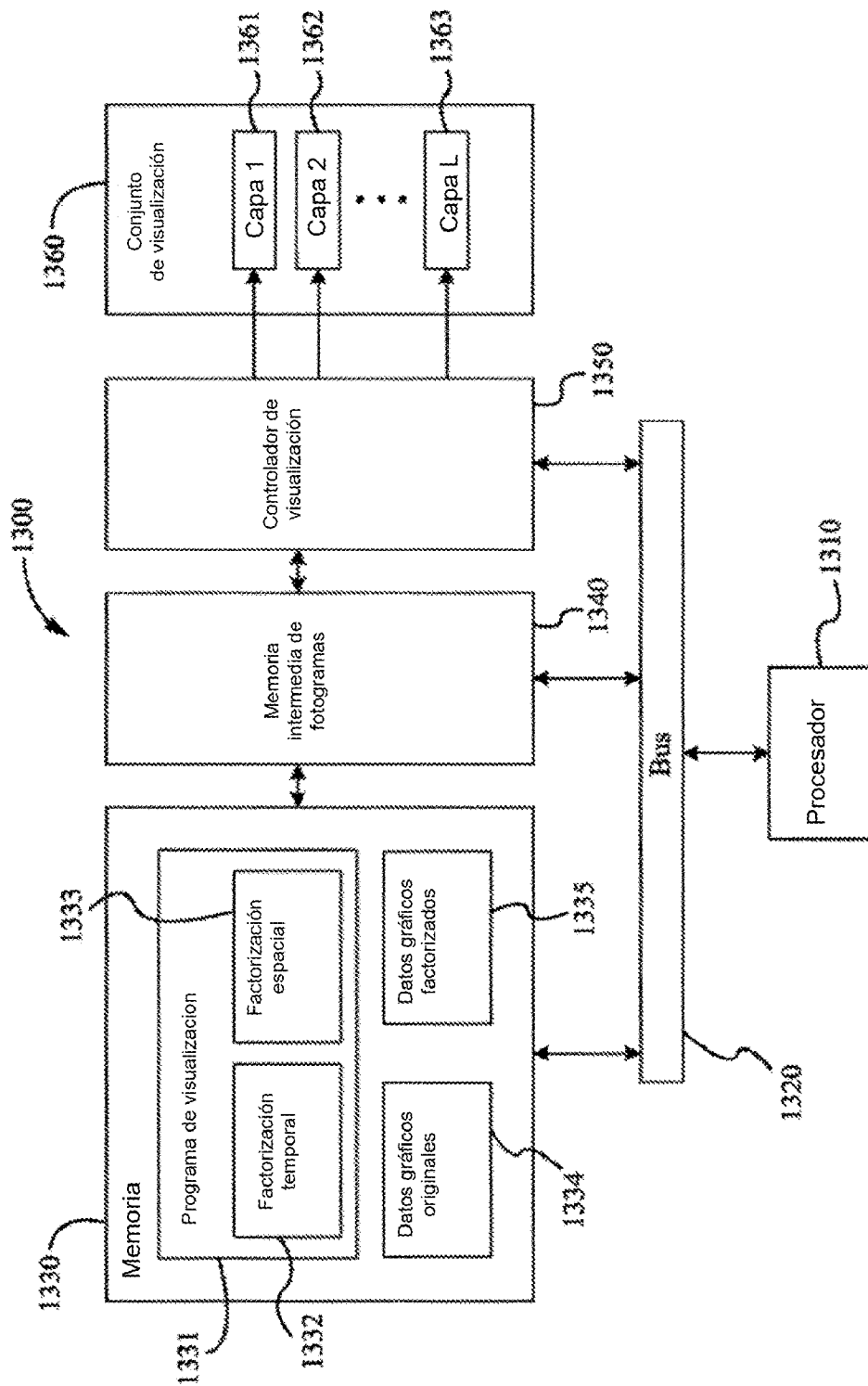


FIG. 13

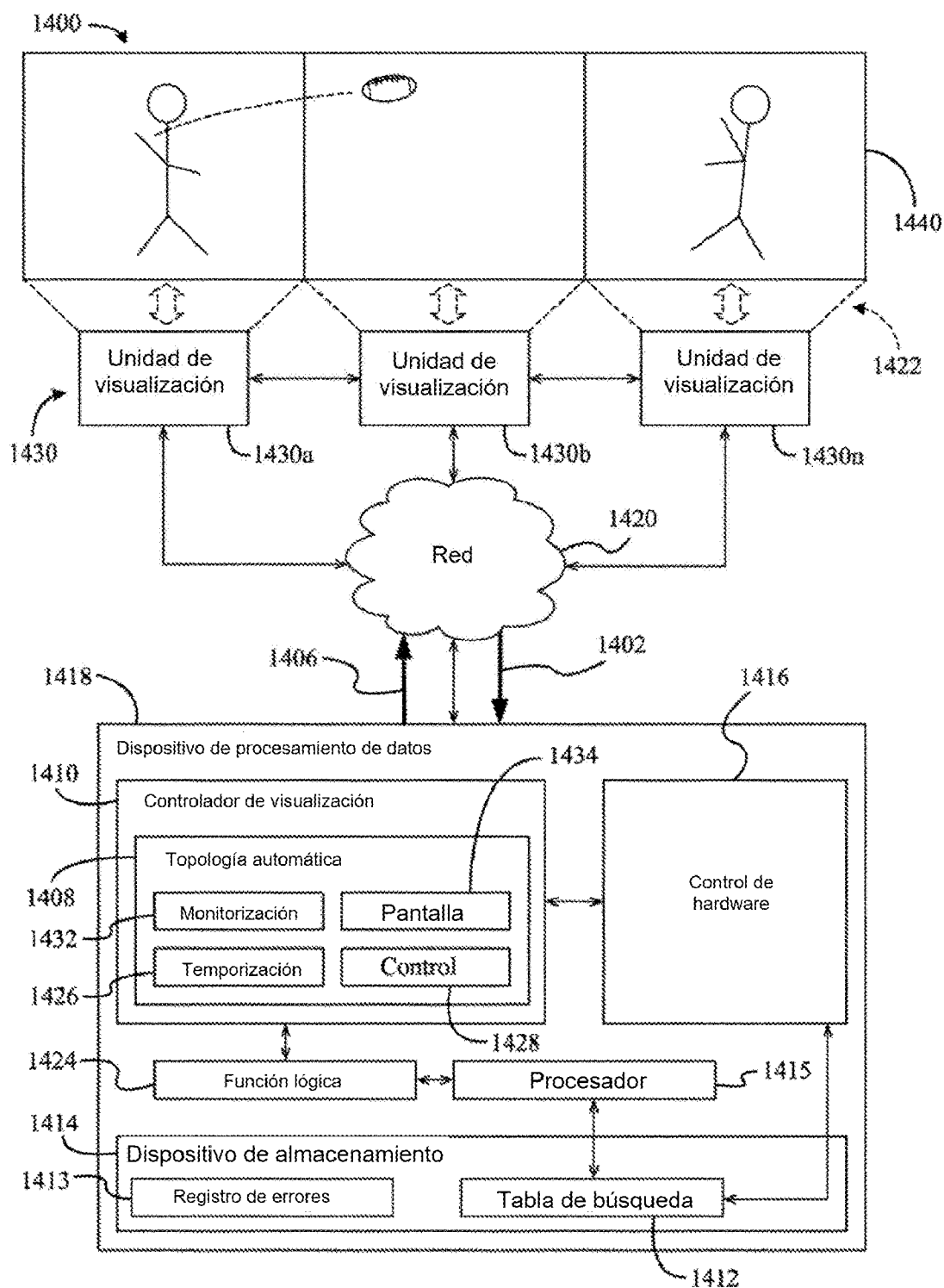


FIG. 14

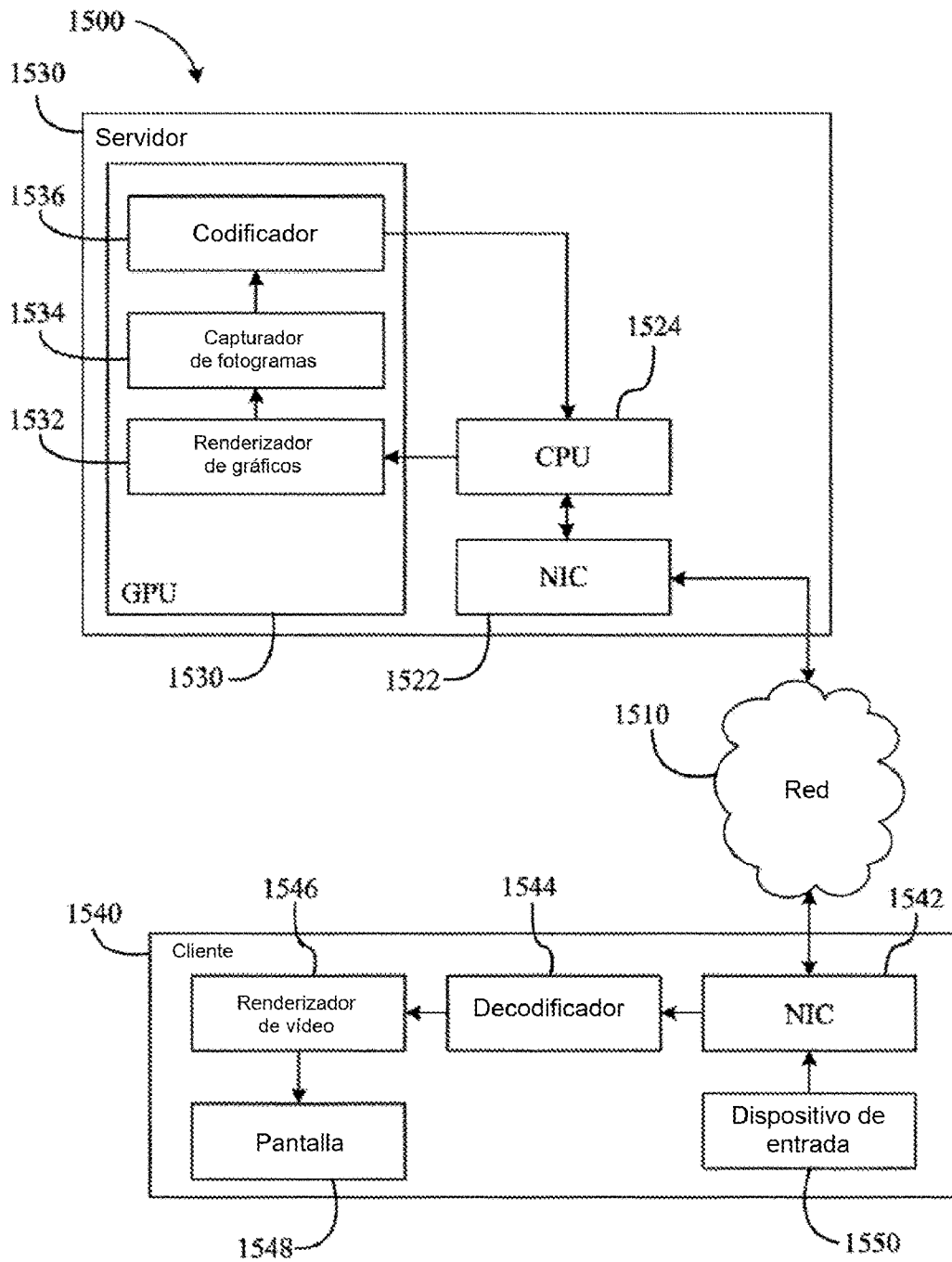


FIG. 15

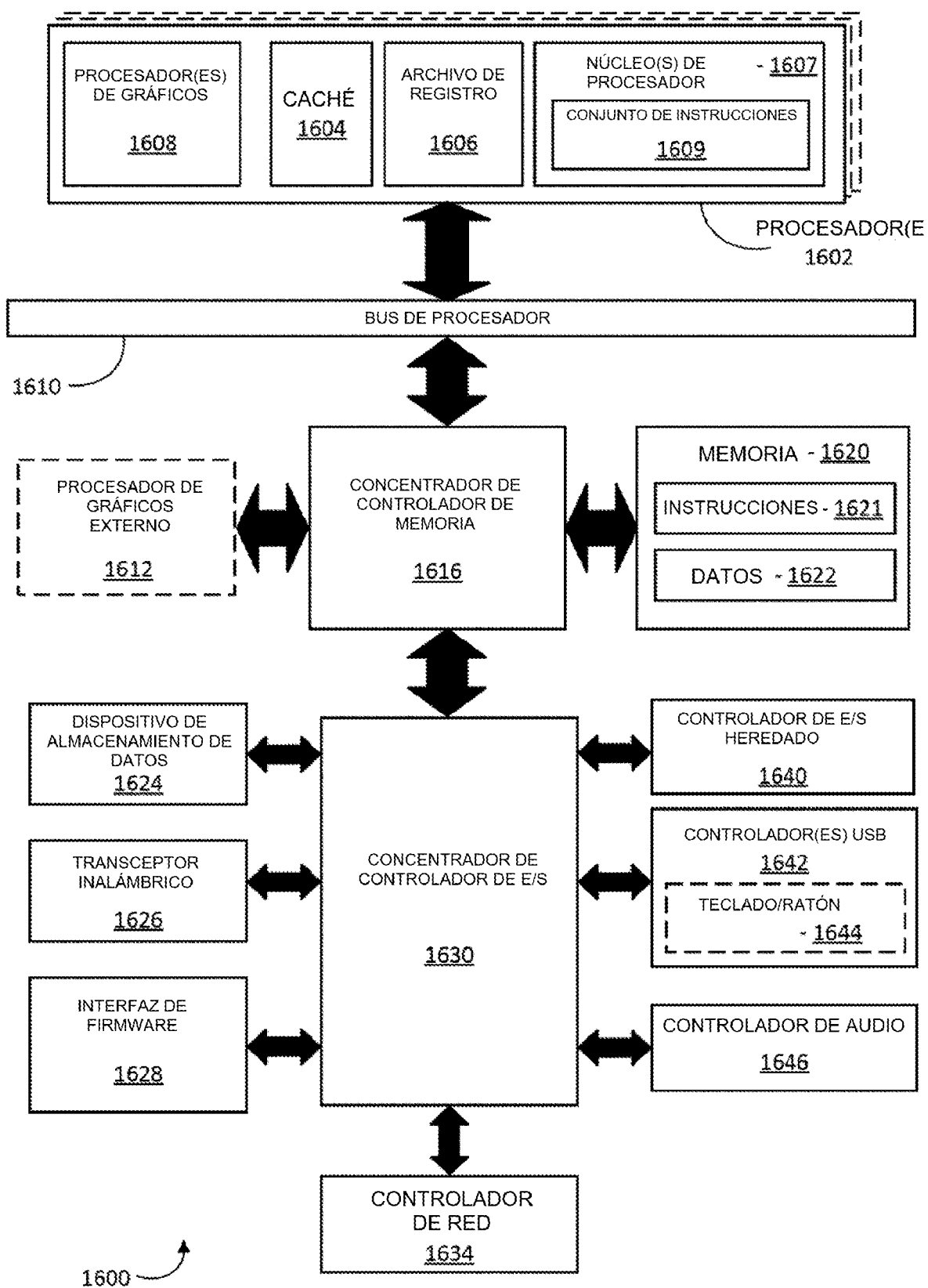


FIG. 16

PROCESADOR 1700 →

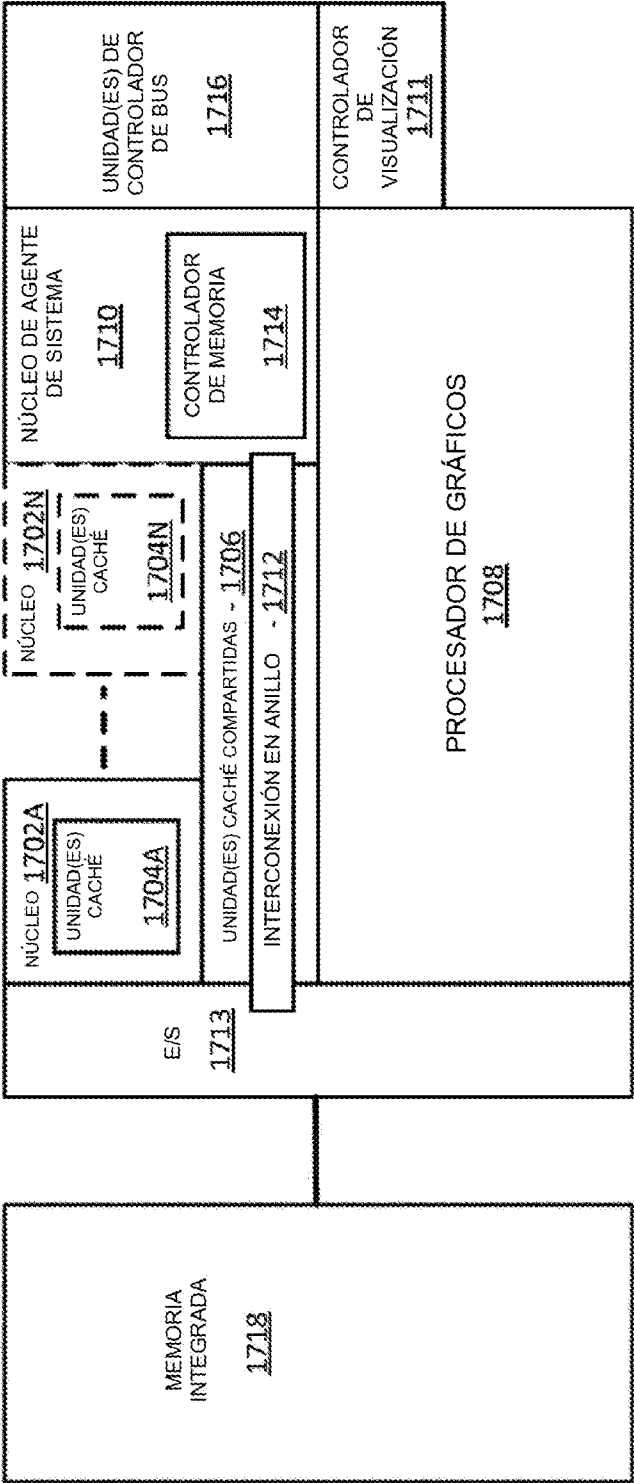


FIG. 17

PROCESADOR DE GRAFICOS
1800

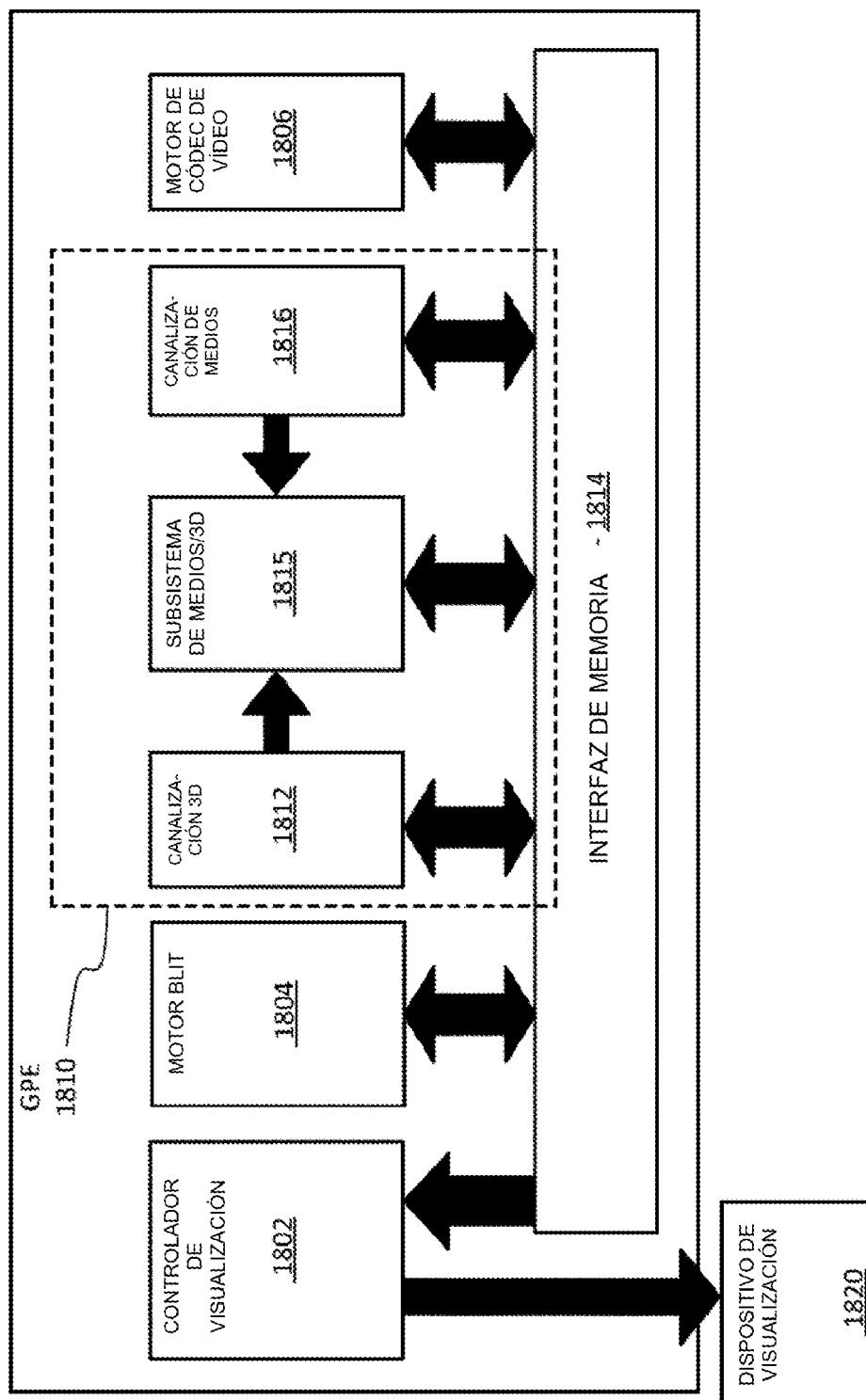
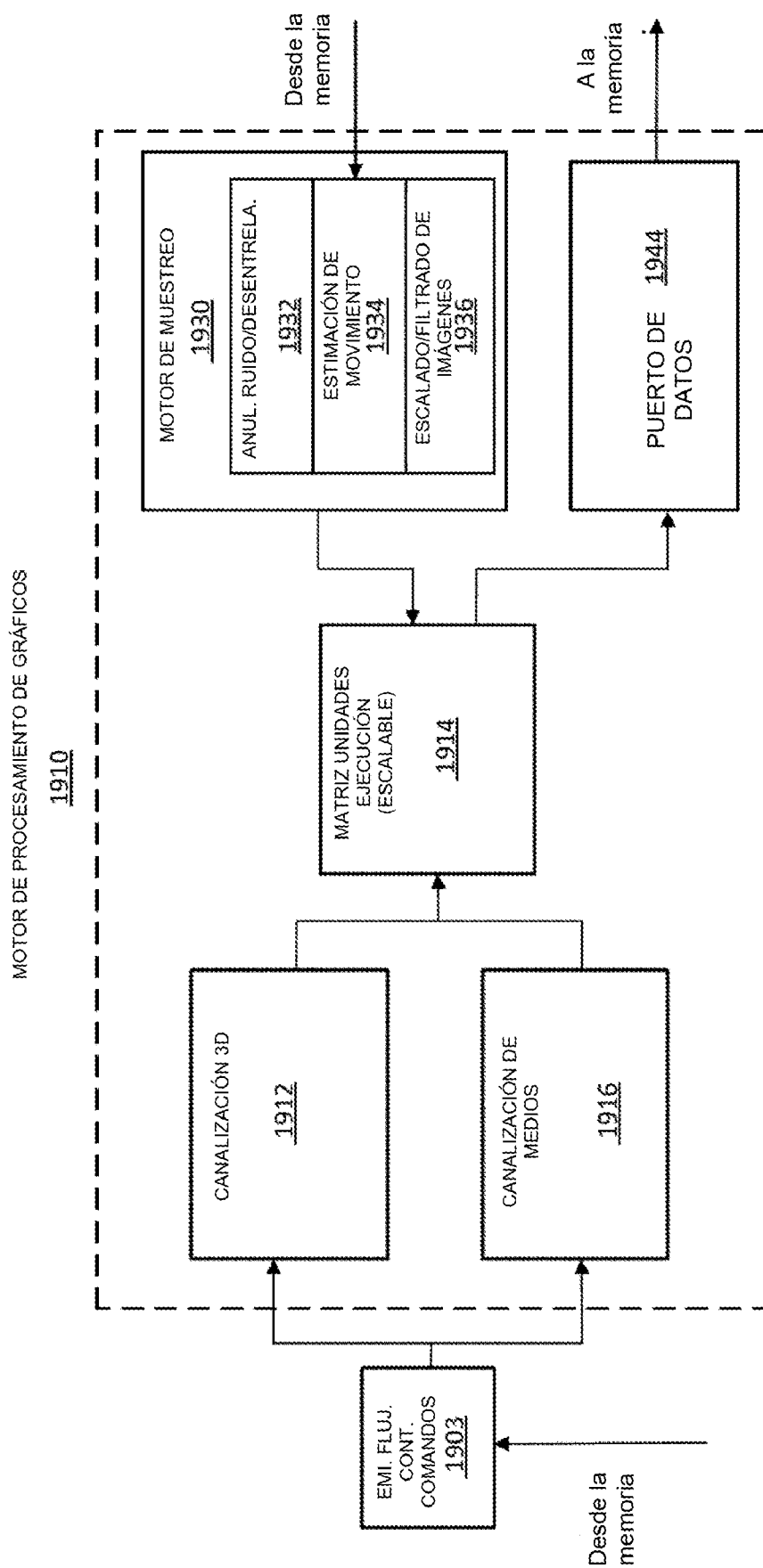


FIG. 18



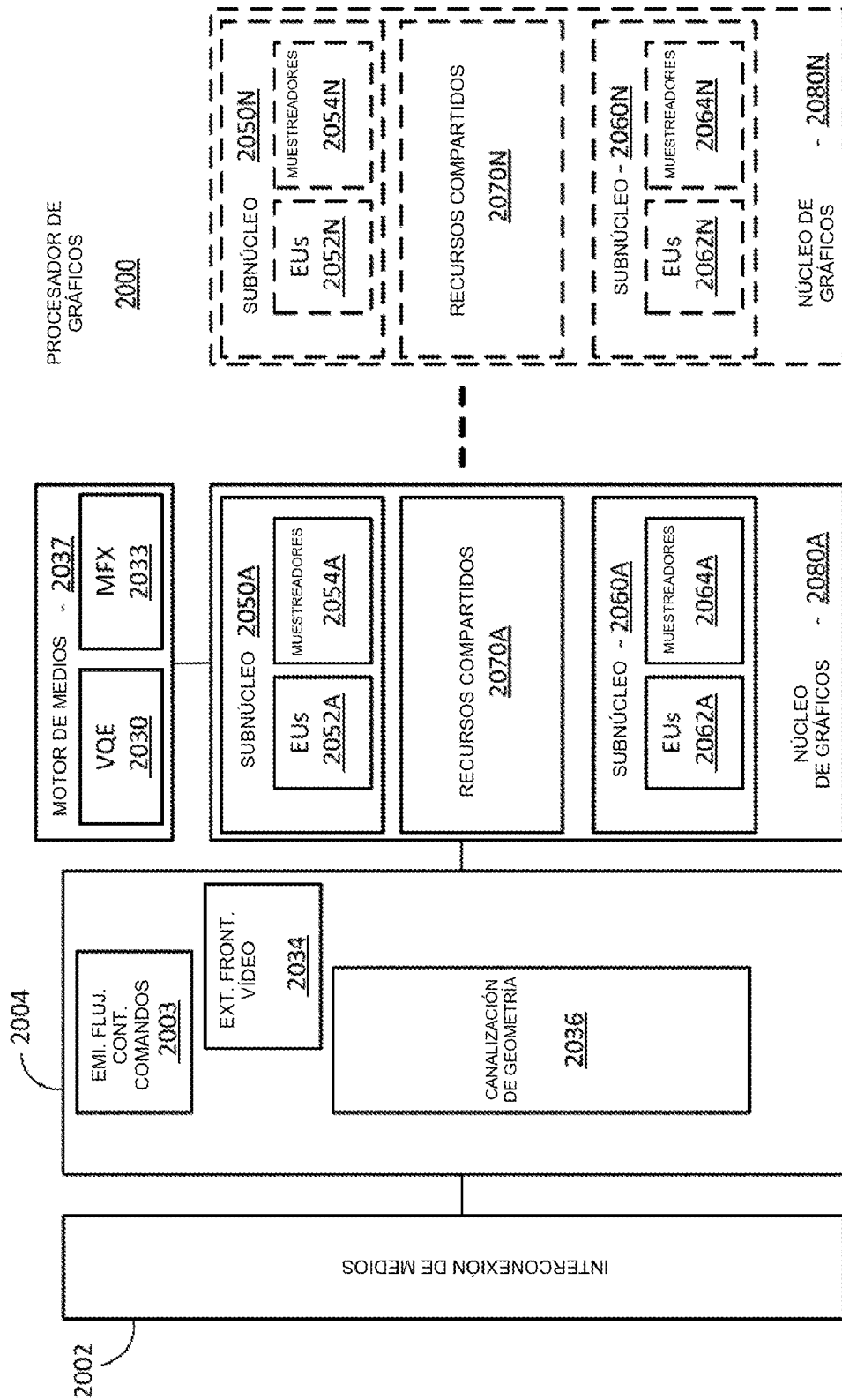


FIG. 20

2100 →

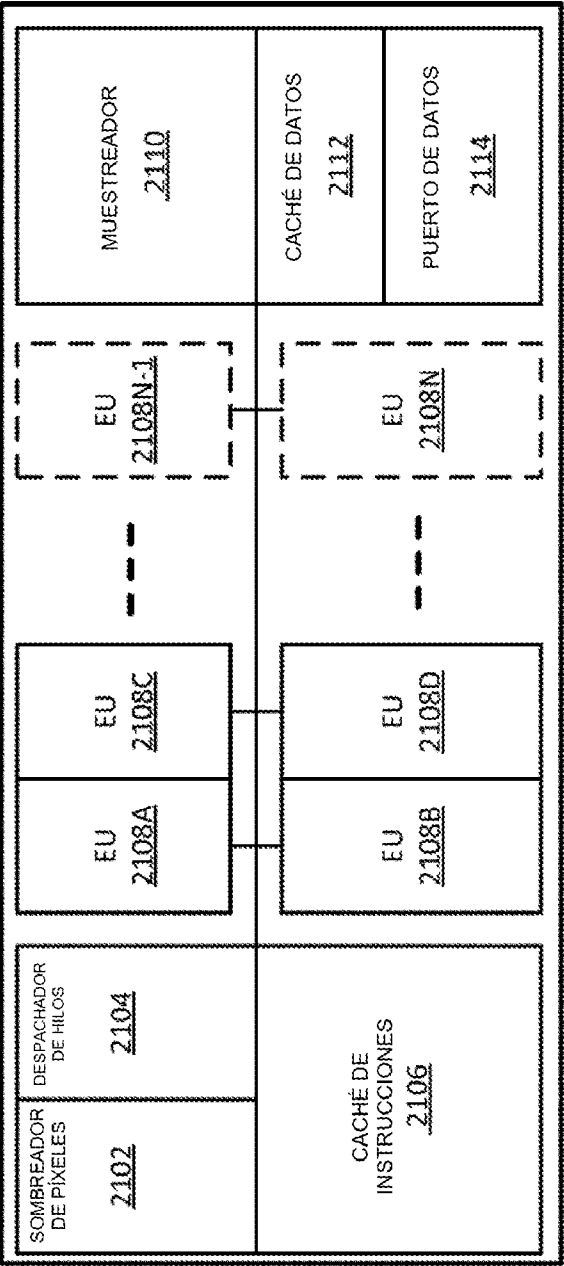
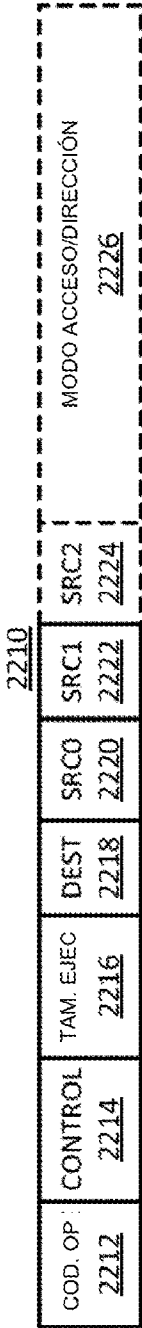


FIG. 21

FORMATOS DE INSTRUCCIÓN DE NÚCLEO DE GRÁFICOS

2200

INSTRUCCIÓN DE 128 BITS



INSTRUCCIÓN COMPACTA DE 64 BITS

2230



DECODIFICAR CÓDIGO DE OPERACIÓN

2240

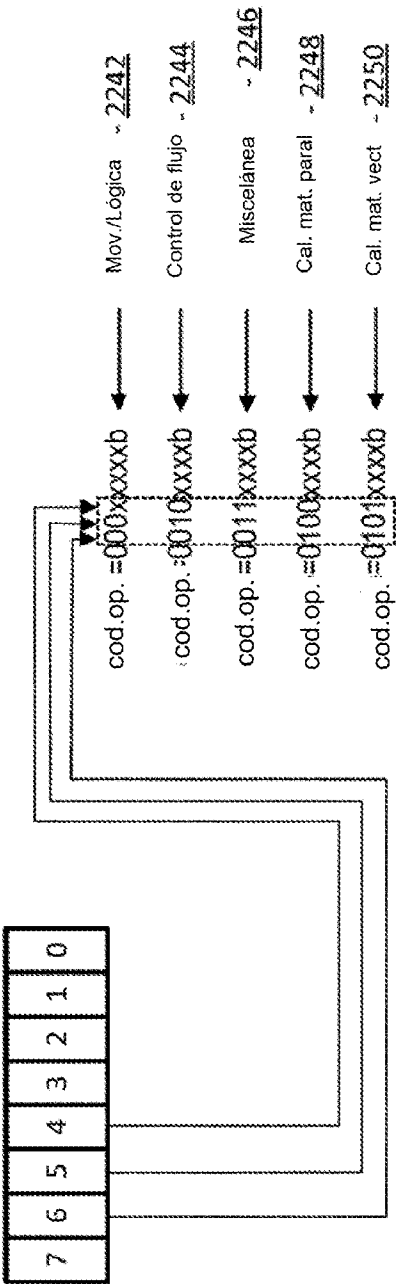


FIG. 22

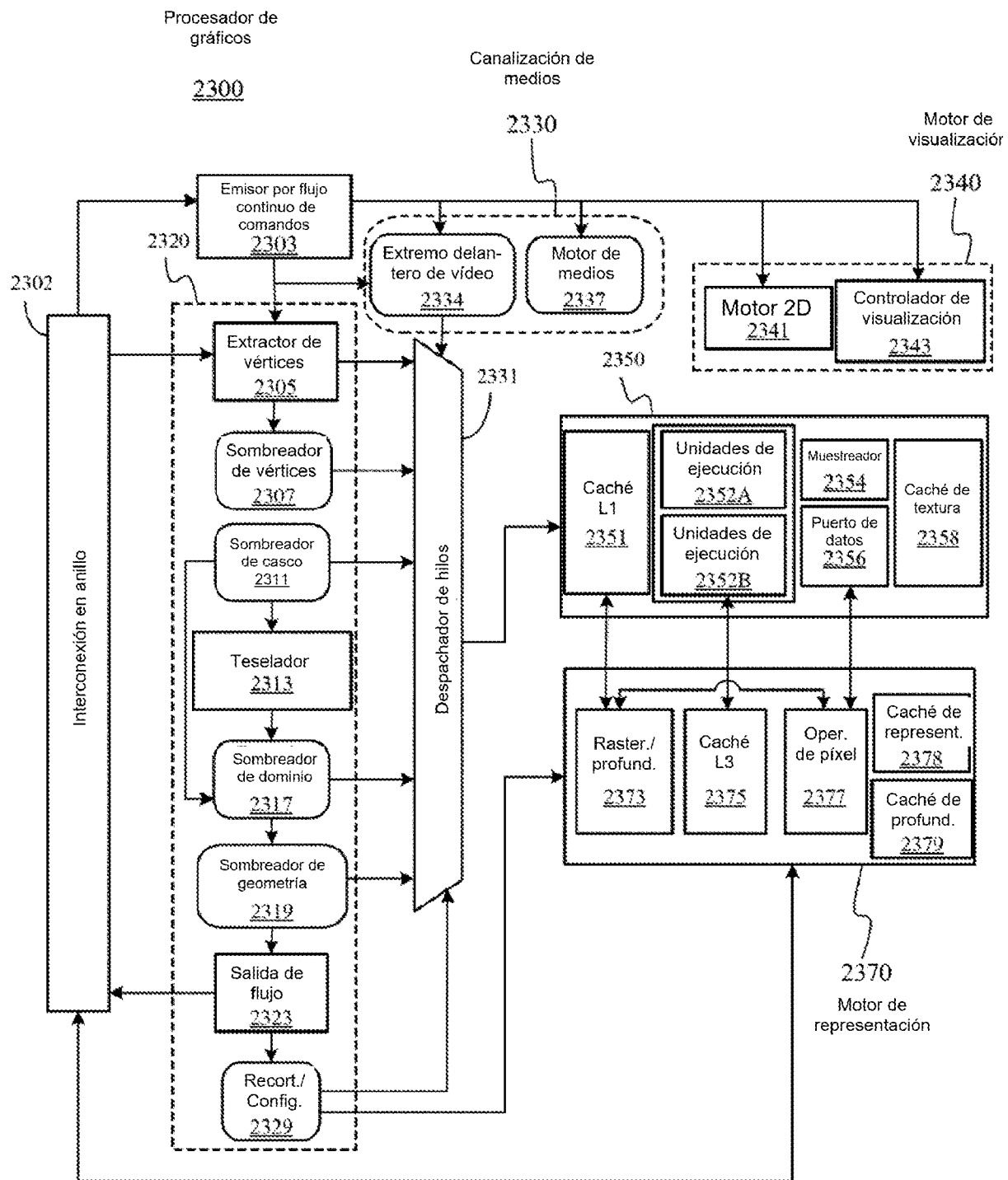


FIG. 23

Formato de comando de procesador de gráficos

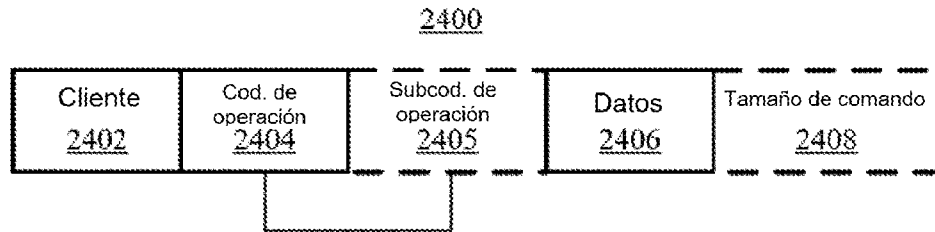


FIG. 24A

Secuencia de comando de procesador de gráficos

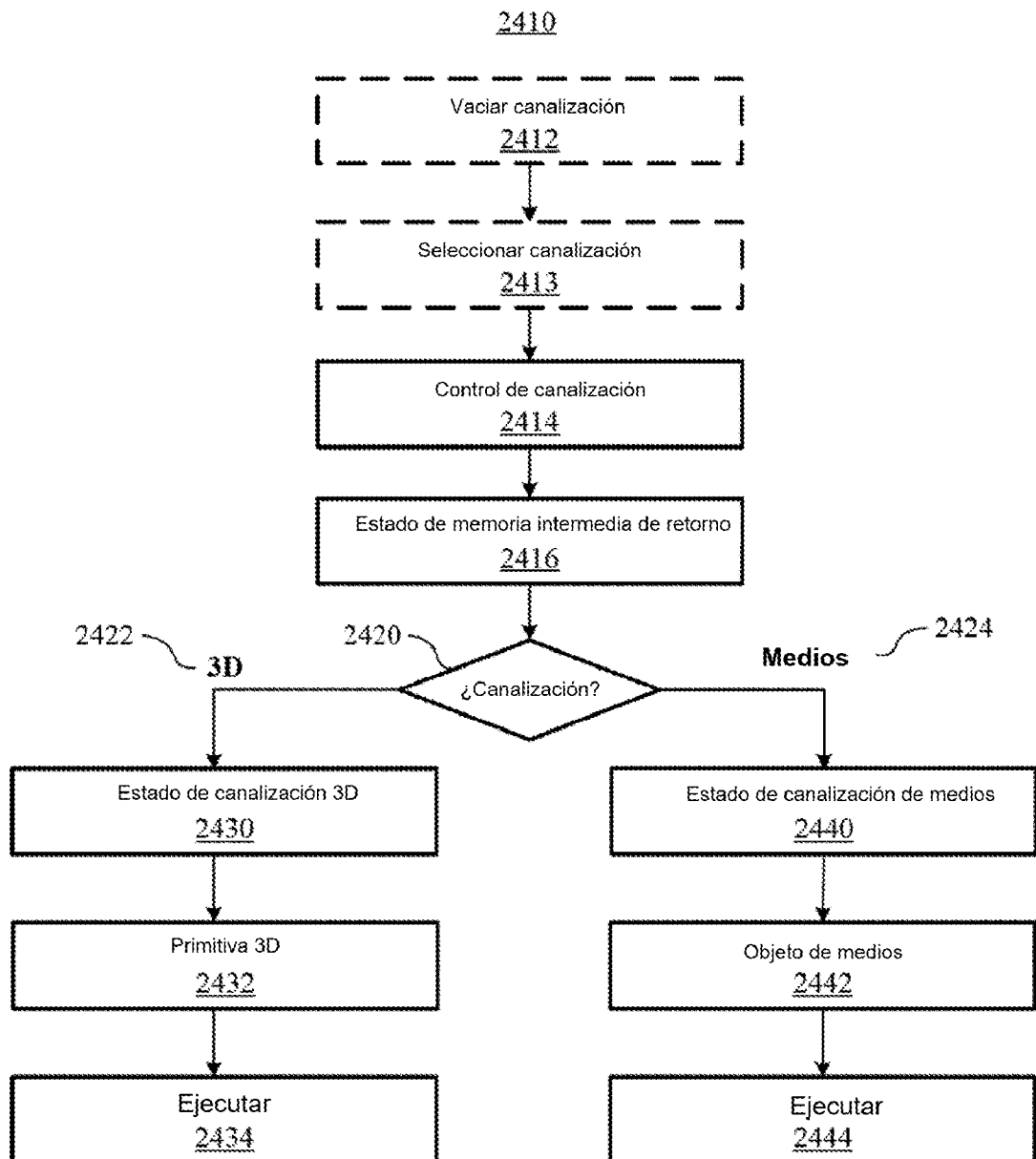


FIG. 24B

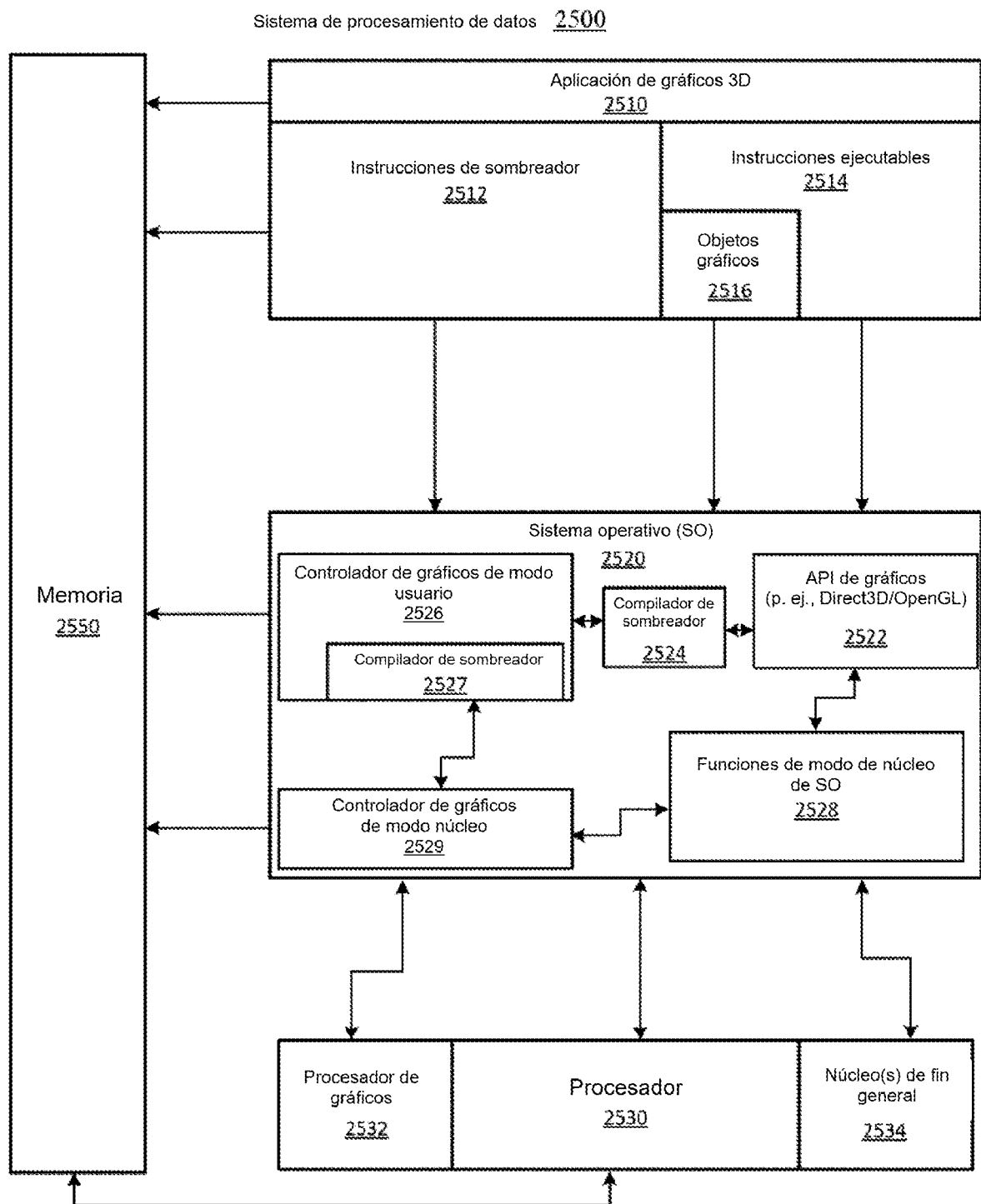


FIG. 25

DESARROLLO DE NÚCLEO IP - 2600

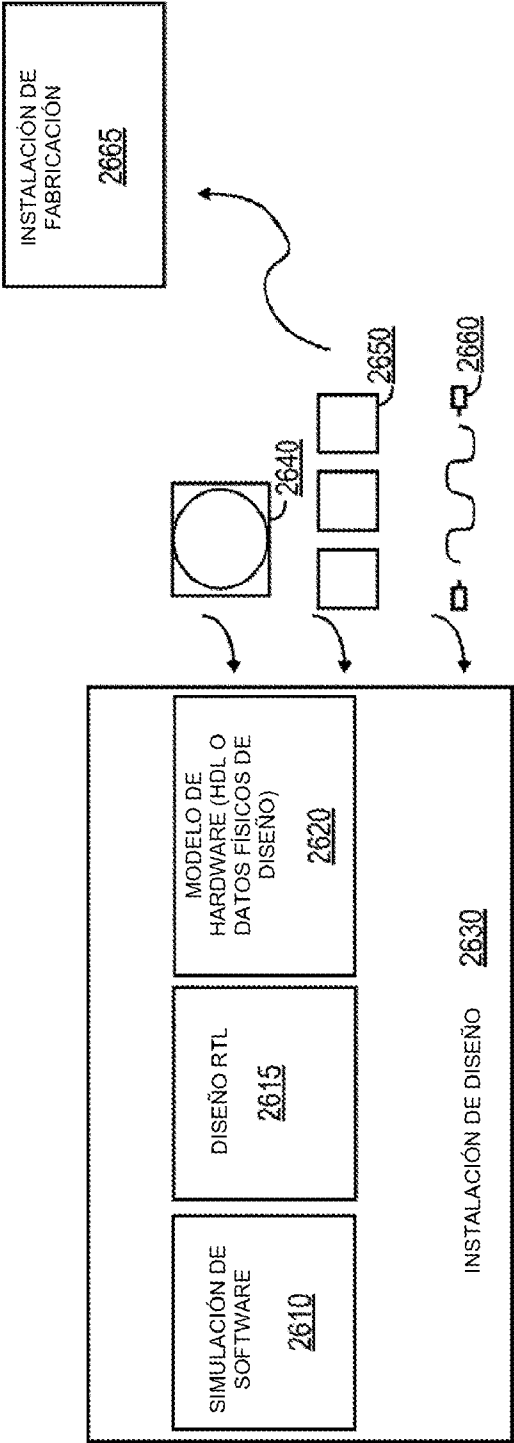


FIG. 26

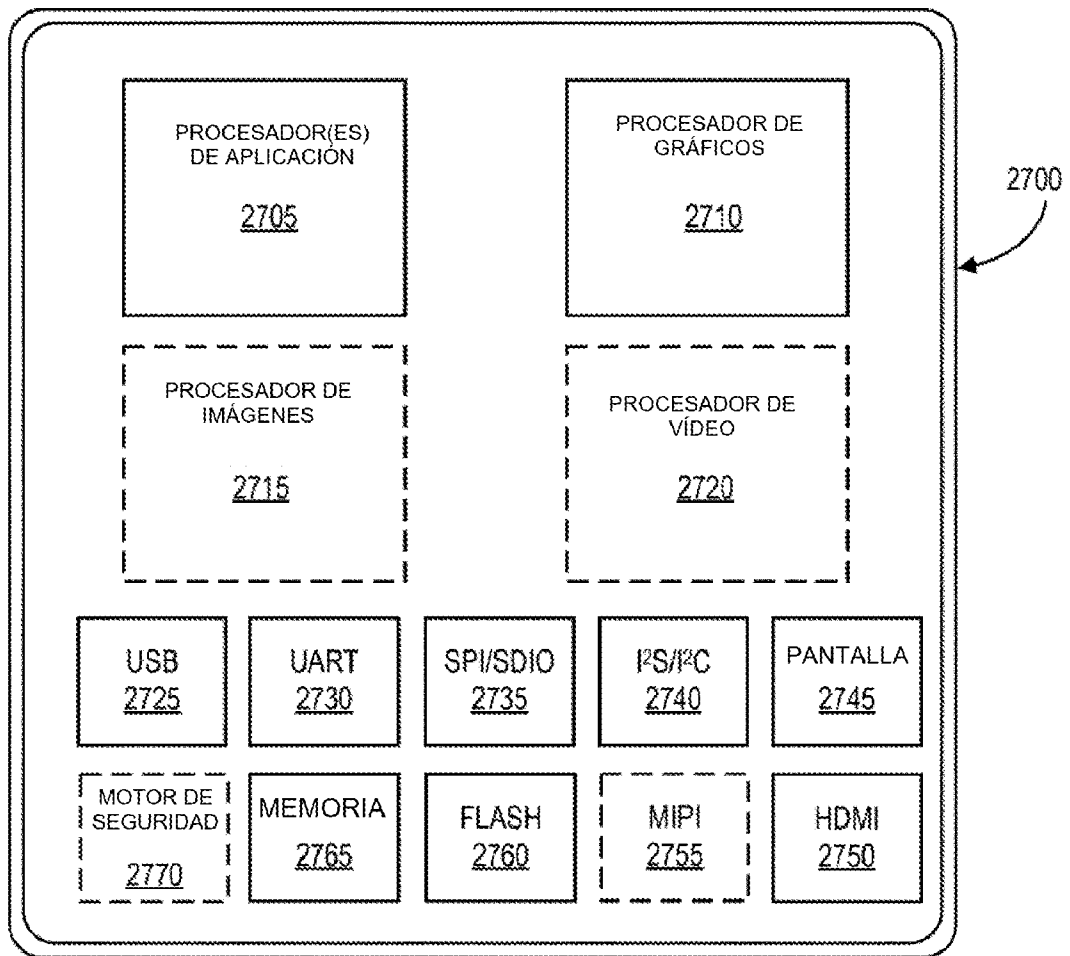


FIG. 27